

**UNIVERSITETI POLITEKNIK I TIRANËS  
FAKULTETI I TEKNOLOGJISË SË INFORMACIONIT  
DEPARTAMENTI I ELEKTRONIKËS DHE TELEKOMUNIKACIONIT**

**LEDION LIÇO**

**PËR MARRJEN E GRADËS**

**“DOKTOR”**

**NË “TEKNOLOGJITË E INFORMACIONIT DHE KOMUNIKIMIT”**

**DREJTIMI “TELEKOMUNIKACION DHE INXHINIERI INFORMACIONI”**

**DISERTACION**

**IMPLEMENTIMI DHE OPTIMIZIMI I TEKNIKAVE TË  
DATA MINING NË SISTEMET E INTELIGJENCËS SË  
BIZNESIT RETAIL NË SHQIPËRI**

**Udhëheqës Shkencor**

**Prof. Asoc. INDRIT ENESI**

**TIRANË, 2021**

**IMPLEMENTIMI DHE OPTIMIZIMI I TEKNIKAVE TË DATA MINING NË  
SISTEMET E INTELIGJENCËS SË BIZNESIT RETAIL NË SHQIPËRI**

**Disertacioni**

i paraqitur në Universitetin Politeknik të Tiranës

për marrjen e gradës

“Doktor”

në

“Teknologjitë e Informacionit dhe Komunikimit”

Drejtimi Telekomunikacion dhe Inxhinieri Informacioni

nga

z. Ledion Liço

TIRANË, 2021

JURIA PËR VLERËSIMIN E DISERTACIONIT PËR FITIMIN E GRADËS SHKENCORE  
“DOKTOR”

Miratuar

Me vendimin e Këshillit të Profesorëve të FTI-së Nr: 13, datë: 09.07.2021.

Kryetari i Jurisë:	Prof. Dr. Elma ZANAJ
Anëtar i Jurisë:	Prof. Asoc. Aleksandër BIBERAJ
Anëtar i Jurisë (Oponent) :	Prof. Asoc. Algenti LALA
Anëtar i Jurisë:	Prof. Dr. Fatmir HOXHA
Anëtar i Jurisë (Oponent) :	Prof. Asoc. Ana KTONA

Dekan i Fakultetit të Teknologjisë së Informacionit

Prof. Dr. Elinda MEÇE

Dedikuar nënës time që më mbështeti në çdo  
hap të rrugëtimit tim!

## ***FALENDERIME!***

Së pari, dua të shpreh mirënjohjen time më të sinqertë, për udhëheqësin tim, Prof. Asoc Indrit Enesi i cili me këmbënguljen dhe kontributin maksimal, bëri të mundur që ky punim të përfundohej me sukses. Puna e vazhdueshme me të, ka qënë tejmase e vlefshme për zhvillimin tim akademik.

Së dyti, dua të falenderoj të gjithë pedagogët e nderuar të Departamentit të Elektronikës dhe Telekomunikacionit në Fakultetin e Teknologjisë së Informacionit, për besimin e treguar ndaj meje dhe ndihmën shumë të vlefshme, që në kohën kur isha student e deri më sot. Eksperienca e marrë nga ky staf shumë i kualifikuar dhe profesional ka qënë thelbesore në rrugëtimin tim akademik dhe profesional.

Së fundmi, falenderoj familjen time që ka qënë pranë meje në çdo hap dhe më kanë mbështetur gjithmonë.

## ABSTRAKTI

Teknikat e Data Mining (DM) dhe Machine Learning (ML) janë zhvilluar shumë në vitet e fundit. Shumica e kërkimeve në fushën e shkencave kompjuterike janë fokusuar në këto 2 fusha. Njëkohësisht, biznesi retail dhe veçanërisht e-commerce është zhvilluar me hapa galopantë dhe për rrjedhojë është rritur nevoja për analiza shumë-dimensionale të të dhënave të klientëve dhe për parashikim e sjelljes së tyre. Sistemet e inteligjencës në biznes (BI) i ofrojnë biznesit analiza multidimensionale të të dhënave të tyre. Kombinimi i këtyre sistemeve me teknikat e DM dhe ML, i jep biznesit, gjithashtu mundësinë të parashikojë sjelljen konsumatore. Megjithatë krijimi i një modeli të përshtatshëm që nxjerr informacion të vlefshëm dhe të saktë nga bazat e mëdha të të dhënave të biznesit ngelet një problem. Një tjetër sfidë është të krijosh një model parashikues, duke qënë se atributet e të dhënave janë në rendin e qindrave apo mijërave dhe zgjedhja e attributeve më të vlefshëm, që na japin saktësinë maksimale në parashikim është e vështirë. Përzgjedhja e algoritmit më të mirë parashikues, është gjithashtu një detyrë tepër komplekse.

Objektivi kryesor i këtij punimi është zgjidhja e problemeve të sipër përmendura. Ne do të ndërtojmë modele që ndërthurin teknikat DM dhe ML me sistemet BI për ti ofruar biznesit një mekanizëm të plotë për të analizuar dhe parashikuar sjelljen e klientëve të tyre.

Studimi ynë do të ndahet në tre pjesë kryesore:

Ne pjesën e parë, ne do të agregojmë në shumë dimesione, të dhënat reale të marra nga një qendër tregtare në Tiranë, Shqipëri. Më pas do të ndërtojmë disa modele klasterizimi dhe do të analizojmë performancën e tyre në bazën tonë të të dhënave. Si përfundim, do të rekomandojmë modelin më të mirë që mund të përdoret për segmentimin e të dhënave të klientit, në qendrat tregtare në varesi të karakteristikave të tyre.

Ne pjesën e dytë, do të propozojmë 3 modele klasifikues me qëllimin e parashikimit të klasës së klientit në fund të vitit duke u nisur nga të dhënat e tre-mujorit të parë. Klasa e klientit është një numër që i alokohet klientit, bazuar në shpenzimet e tij vjetore. Modelet do të bazohen në algoritmat, Support Vector Machines (SVM), K-Nearest Neighbours (KNN) dhe në rrjetat neurale. Modelet do të trajnohen në të dhënat e tre-mujorit të parë të 2020 dhe do të testohen duke krahasuar klasën e parashikuar me klasën reale. Performanca e modeleve do të vlerësohet bazuar në metrikën e përqindjes së saktësisë në parashikim.

Ne pjesën e tretë të studimit tonë, do të analizojmë modelet e serive kohore dhe do të propozojmë një version të modifikuar të modelit Prophet për të parashikuar shitjet ditore në qendrën tregtare. Modeli bazë dhe modeli i modifikuar Prophet do të trajnohen dhe testohen në të dhënat e shitjeve nga 2011-2019, të qendrës tregtare dhe performanca e tyre do të vlerësohet në bazë të mesatares së gabimit absolut në parashikim (MAE). Modeli i modifikuar do të aplikohet në bazën e të dhënave dhe do të kryhet parashikimi për 6 mujorin në vazhdim të shitjeve. Performanca e modelit do të vlerësohet duke u bazuar në mesataren e gabimit absolut në përqindje (MAPE).

## ABSTRACT

Data Mining (DM) and Machine Learning (ML) have been evolving rapidly in the last years. Most of the Computer Science Communities have been focusing their research on these topics. Along with them, Retail and E-commerce is growing at the same pace and the need for customer analytics and behaviour prediction has become critical. Business Intelligence BI systems are particularly used to supply the retail business with multidimensional analysis of their data. Combining BI systems with DM and ML techniques, enables them to also predict their customer's behaviour. Although, building the right model that extracts valuable and accurate information from large customers' transactional databases remains a problem. Another challenge is creating a prediction model as the attributes of the data are in hundreds or thousands and finding the most valuable predictors is not easy. Utilizing the best segmentation and prediction algorithms is also a complex task.

The main purpose of our study is addressing these problems. We will create models based on DM and ML, in conjunction with BI to offer a complete mechanism for the businesses operating in retail, to analyse and predict their customer behaviour. Our study can be divided in three main sections:

In the first, we will first aggregate the data from a real department store in Albania, in multiple dimensions. Then we will build several clustering models and analyse their performance on this real data. We will conclude, in what is the best model that can be used for department stores based on their characteristics.

In the second part of the study, we will propose 3 classification models with the aim to predict the customer label at the end of the year based on quarterly data. The models will be based on Support Vector Machines (SVM), K-Nearest Neighbours (KNN) and neural networks algorithms. The customer label is a number allocated to each customer based on their annual expenses. Or models will be trained and tested in our dataset. Their performance will be evaluated on accuracy percentage metric.

In the third part of our study, we will analyse time series models and particularly propose a modified Prophet model to predict daily sales in the department stores. The base prophet model and the modified one will be evaluated based on prediction MAE (Mean Absolute Error). The modified model will be trained and tested in the sales of the department store from 2011-2019 and its performance will be evaluated bases on prediction MAPE (Mean Absolute Percentage Error).

## PËRMBAJTJA

<b>Abstrakti</b>	<b>vi</b>
<b>Abstract</b>	<b>vii</b>
<b>LISTA E TABELAVE</b>	<b>xi</b>
<b>LISTA E FIGURAVE</b>	<b>xii</b>
<b>LISTA E SIMBOLEVE DHE SHKURTIMEVE</b>	<b>xiv</b>
<b>HYRJE</b>	<b>1</b>
<b>MOTIVIMI</b>	<b>5</b>
<b>KAPITULLI 1</b>	<b>6</b>
<b>1. SISTEMET E INTELIGJENCËS NË BIZNES</b>	<b>6</b>
1.1 Hyrje	6
1.2 Evolimi i sistemeve BI	6
1.3 Përkufizimi i një sistemi BI	6
1.4 Sistemet BI	7
1.5 Arkitektura BI	7
1.5.1 Komponentet e një sistemi BI	9
1.6 Karakteristikat kryesore të një sistemi BI	11
1.7 Drejtuesit e një implementimi BI	11
1.8 Faktorët përcaktues në implementimin e sistemeve BI	14
1.9 Lidhjet midis sistemeve të inteligjencës së biznesit dhe magazinave të të dhënave	15
1.10 Benefitet e implemetimit të një sistemi BI	16
1.11 Konkluzione	17
<b>KAPITULLI 2</b>	<b>18</b>
<b>2. MAGAZINAT E TË DHËNAVE DHE TEKNOLOGJIA E KUBAVE TË TË DHËNAVE</b>	<b>18</b>
2.1 Hyrje	18
2.2 Përcaktimi i një Data Warehouse	18
2.3 Një krahasim midis bazave të të dhënave për procesim transaksionesh dhe ato që ndihmojnë në vendim-marrje	19
2.4 Teknologjia e kubave të të dhënave	19
2.5 Serveri OLAP	20
2.6 Metadata	20
2.7 Drill-down	20
2.8 Data mining	21
2.9 OLAP dhe Data Warehousing	21
2.10 Arkitektura e kubave	25
2.11 Kubat relacionar të të dhënave	26
2.12 Performanca	29



2.12.1	<i>Grupimet</i>	29
2.12.2	<i>Indeksi bitmap</i>	29
2.12.3	<i>Copëzimi</i>	31
<b>2.13</b>	<b>Konkluzione</b>	<b>32</b>
<b>KAPITULLI 3</b>		<b>33</b>
<b>3.</b>	<b>TEKNIKAT E DATA MINING</b>	<b>33</b>
<b>3.1</b>	<b>Hyrje</b>	<b>33</b>
<b>3.2</b>	<b>Modeli përshkrues</b>	<b>35</b>
3.2.1	<i>Ndarja në grupe</i>	35
3.2.2	<i>Asocimi</i>	35
3.2.3	<i>Ekstraktimi i veçorive</i>	36
<b>3.3</b>	<b>Modeli parashikues</b>	<b>37</b>
3.3.1	<i>Klasifikimi</i>	37
3.3.2	<i>Regresi</i>	38
3.3.3	<i>Seritë kohore</i>	39
3.3.4	<i>Identifikimi i anomalive të të dhënave</i>	39
<b>3.4</b>	<b>Teknikat data mining që përdoren në sistemet e inteligjencës së biznesit BI</b>	<b>40</b>
<b>3.5</b>	<b>Hapat e DATA-MINING</b>	<b>42</b>
3.5.1	<i>Cilësia e të dhënave</i>	42
3.5.2	<i>Hapat kryesore për paraprosesimin e të dhënave</i>	43
<b>3.6</b>	<b>Konkluzione</b>	<b>46</b>
<b>KAPITULLI 4</b>		<b>47</b>
<b>4.</b>	<b>TEKNIKAT DM TË MARRA NË SHQYRTIM</b>	<b>47</b>
<b>4.1</b>	<b>Metoda e Ndarjes/Particionimit në Klastera</b>	<b>47</b>
4.1.1	<i>Algoritmi K-Means</i>	47
4.1.2	<i>Algoritmi k-Medoids</i>	50
<b>4.2</b>	<b>Metoda hierkike e klasterizimit</b>	<b>53</b>
4.2.1	<i>Metoda hierkike grumbulluese vs ndarëse</i>	54
4.2.2	<i>Përlllogaritja e distances në metodat algoritmike</i>	55
4.2.3	<i>Algoritmi BIRCH</i>	57
4.2.4	<i>Metoda Chameleon</i>	60
<b>4.3</b>	<b>Teknikat e Klasifikimit</b>	<b>61</b>
4.3.1	<i>Konceptet Themelore</i>	62
<b>4.4</b>	<b>Rrjetat neurale</b>	<b>63</b>
4.4.1	<i>Neuroni artificial: Funksionet e aktivizimit</i>	63
4.4.2	<i>Derivatet e funksioneve të aktivizimit</i>	67
4.4.3	<i>Regresi</i>	68
4.4.4	<i>Klasifikimi</i>	69
4.4.5	<i>Funksionet e gabimit</i>	71
4.4.6	<i>Trajnimi i rrjetave neurale me një shtresë të ndërmjetme</i>	72
4.4.7	<i>Trajnimi i MLPs të thella</i>	81
<b>4.5</b>	<b>Support Vector Machines SVM</b>	<b>84</b>
4.5.1	<i>SVM për më shumë se 2 klasa</i>	84
4.5.2	<i>SVM me funksionin Bërthamë: Rasti jo-linear</i>	85
<b>4.6</b>	<b>Klasifikuesi i K fqinjëve më të afërt (KNN)</b>	<b>88</b>
<b>4.7</b>	<b>Algoritmi Prophet</b>	<b>89</b>

4.7.1	<i>Modeli i Trendit</i>	89
4.7.2	<i>Modeli i sezonalitetit</i>	90
4.7.3	<i>Ditët Festive</i>	91
4.7.4	<i>Metrikat e gabimit</i>	91
<b>4.1</b>	<b>Konkluzione</b>	<b>92</b>
<b>KAPITULLI 5</b>		<b>93</b>
<b>5.</b>	<b>PUNA EKSPERIMENTALE</b>	<b>93</b>
<b>5.1</b>	<b>Teknikat e Klasterizimit</b>	<b>93</b>
5.1.1	<i>Eksplorimi i bazës së të dhënave</i>	93
5.1.2	<i>Para-proçesimi i të dhënave</i>	94
5.1.3	<i>Korrelimi midis attributeve</i>	95
5.1.4	<i>Klasterizimi me K-Means</i>	96
5.1.5	<i>Klasterizimi me K-Medoids</i>	97
5.1.6	<i>Klasterizimi me DBSCAN</i>	98
5.1.7	<i>Algoritmi Hierarkik Aglomerativ</i>	99
5.1.8	<i>Rezultate të Klasterizimit</i>	100
<b>5.2</b>	<b>Modelet e klasifikimit</b>	<b>101</b>
5.2.1	<i>Eksplorimi i bazës së të dhënave</i>	101
5.2.2	<i>Para-proçesimi i të dhënave</i>	102
5.2.3	<i>Modeli KNN</i>	103
5.2.4	<i>Support Vector Machines</i>	104
5.2.5	<i>Eksplorimi i të dhënave dhe para-proçesimi</i>	105
5.2.6	<i>Modeli SVM</i>	105
5.2.7	<i>Rrjetat Neurale</i>	107
<b>5.3</b>	<b>Analiza e serive kohore</b>	<b>110</b>
5.3.1	<i>Modeli Prophet</i>	110
5.3.2	<i>Eksplorimi i të dhënave</i>	110
5.3.3	<i>Aplikimi i modelit standart dhe të modifikuar</i>	112
<b>5.4</b>	<b>Konkluzione</b>	<b>116</b>
<b>KAPITULLI 6</b>		<b>117</b>
<b>6.</b>	<b>REZULTATET E PUNIMIT DHE PUNA NË TË ARDHMEN</b>	<b>117</b>
<b>6.1</b>	<b>Rezultatet e punimit</b>	<b>117</b>
<b>6.2</b>	<b>Puna në të ardhmen</b>	<b>119</b>
<b>Referencat</b>		<b>120</b>
<b>Aneks 1</b>	<b>122</b>	

**LISTA E TABELAVE**

Tabela 2-1 Shembulli i tabelës së shitjeve për disa produkte të kompanisë .....	26
Tabela 2-2 Shembull i tabelës pivot.....	27
Tabela 2-3 Rezultatet e operatorit të kubit në agregimin e operatorit shumë, kur aplikohet në Tabelën 2.1.....	27
Tabela 2-4 Rezultatet e operatorit rollup kur aplikohet në Tabelën 2.1. ....	28
Tabela 2-5 Shembull i indeksimit bitmap. ....	30
Tabela 2-6 Shembull i indeksimit bit-sliced. ....	31
Tabela 3-1 Algoritme të Klasterizimit .....	35
Tabela 3-2 Shembull transaksionesh .....	36
Table 5-1 Përmbledhëse e datasetit dhe 5 rreshtat e parë .....	94
Tabela 5-2 Performanca në kohë e modeleve të klasterizimit .....	100
Tabela 5-3: Përmbledhëse e datasetit dhe 5 rreshtat e parë .....	102
Tabela 5-4 Saktësia pas Cross Validation.....	104
Tabela 5-5 Saktësia në varësi të normalizimit të të dhënave .....	105
Tabela 5-6 Cross-Validation në SVM.....	105
Tabela 5-7 Parametrat e modelit NN .....	107
Tabela 5-8 5 rreshtat e parë te bazës së të dhënave .....	111
Tabela 5-9 5 rreshtat e parë të databazës së festave.....	111
Tabela 5-10 Modeli i modifikuar Prophet .....	112

## LISTA E FIGURAVE

Figura 1-1 Arkitektura klasike e BI .....	8
Figura 1-2 Arkitektura teknike e një BI.....	9
Figura 1-3 Data warehouse të ngarkuara nga sistemet burim.....	10
Figura 1-4 Arsyet që të çojnë drejt inteligjencës në biznes .....	13
Figura 1-5 Vlerat e BI kundrejt kohës dhe kostos .....	14
Figura 2-1 Drill-down .....	21
Figura 2-2 Shembull i skemës yll. Çelësat e jashtëm janë shënuar me <fk> .....	22
Figura 2-3 Shëmbulli se si Figura 2.2 mund të organizohet si skema snowflake. Çelësat e jashtëm janë shënuar me <fk> .....	23
Figura 2-4 Grafi përshkruan se si operacioni group by mund të llogaritet duke përdorur operacionin group by të mëparshëm në vend të burimit të të dhënave.....	29
Figura 3-1 Modeli DM përshkruet (i pasupervizuar) dhe modeli DM parashikues (i supervizuar).....	34
Figura 3-2 Arkitektura OLAM.....	41
Figura 3-3 Hapat e paraprocesimit të të dhënave.....	45
Figura 4-1 Klasterizimi i një seti objektësh që përdorin algoritmin k-means; (b) azhurnimi e qendrave të klasterave dhe rivendosja e objekteve në përputhje me gjendjet (mesatarja e secilit klaster shënohet me +).....	48
Figura 4-2 Katër rastet e funksionit të kostos për klasterizimin k-medoids .....	51
Figura 4-3 Klasterizimi agglomerative dhe hierarkik në objektet {a, b, c, d, e} .....	54
Figura 4-4 Paraqitja Dendogram për klaster-izimin hierarkik për objektet data {a, b, c, d, e} .....	55
Figura 4-5 Klasterizimi linkage .....	57
Figura 4-6 Struktura CF – pemë .....	59
Figura 4-7 Kameleon: Klasterizim hierarkik i bazuar në k-fqinjët më të afërt dhe modelimi dinamik .....	60
Figura 4-8 Neuron artificial: Mbledhja dhe Aktivizimi.....	64
Figura 4-9 Funksionet e aktivizimit të neuroneve; ilustrimi i efektit të bias-it. ....	66
Figura 4-10 Softmax (netkversus netj) .....	67
Figura 4-11 Regresi linear dhe logjistik përmes rrjetave neurale.....	69
Figura 4-12 Rrjetet neurale për regresin logjistik multiklasor: Të dhënat e përbërësve kryesorë të Iris. Pikat e multiklasifikimit janë treguar në ngjyrë gri të errët. Pikat në klasën $c_1$ dhe $c_2$ tregohen të zhvendosura në lidhje me klasën bazë $c_3$ vetëm për ilustrim. ....	71
Figura 4-13 Perceptron i thellë me shumë shtresa, me h shtresa të fshehura. ....	74
Figura 4-14 Plani ndarës në 2D .....	84
Figura 5-1 Korrelimi midis attributeve .....	95
Figura 5-2 Metoda Elbow .....	96
Figura 5-3 Klasterat e klienteve K-Means .....	97
Figura 5-4 Klasterat e klienteve K-Medeoids.....	98
Figura 5-5: DBSCAN Min_pts=7 eps=0.3 .....	98
Figura 5-6 DBSCAN Min_pts=7 ,eps=2 .....	99
Figura 5-7 DBSCAN Min_pts=7 ,eps=0.05 .....	99
Figura 5-8 Dendogrami i të dhënave .....	100
Figura 5-9 Rezultatet e ndarjes në klastera.....	101
Figura 5-10 Ndarja e klientëve në klasa .....	101
Figura 5-11: Saktësia për numer te ndryshem fqinjesh .....	103
Figura 5-12 Matrica e Saktësisë.....	104
Figura 5-13 Matrica e saktësisë për SVM.....	106

Figura 5-14 Saktësia për tipe të ndryshme bërthamash .....	106
Figura 5-15 Humbja Cross Entropy për numër të ndryshëm nyjesh në shtresën e ndërmjetme .....	108
Figura 5-16 Saktësia për numër të ndryshëm nyjesh në shtresën e ndërmjetme .....	108
Figura 5-17 Humbja për funksione të ndryshme aktivizimi .....	109
Figure 5-18 Saktësia për funksione të ndryshme aktivizimi.....	109
Figure 5-19 Ecuria e shitjeve në vitet 2011-2019 .....	112
Figura 5-20 Parashikimi i shitjeve me anë të modelit të modifikuar Prophet .....	113
Figure 5-21 Krahasimi i modeleve për 6 mujorin e dytë të vitit 2019.....	113
Figura 5-22 Vlerat e MAPE pas Cross Validation.....	114
Figura 5-23 Trendet në shitje për qendrën tregtare.....	115

## LISTA E SIMBOLEVE DHE SHKURTIMEVE

**RDBMS** – Sistemet e menaxhimit të bazave të të dhënave relacionale (Relational Database Management Systems)

**KDD** - Zbulimi i Informacionit në Bazat e të Dhënave (Knowledge Discovery in Databases)

**DM** - Data Mining

**ERP**- Sistmet e Planifikimit të Burimeve të Ndërmarrjeve (Enterprise Resource Planning)

**ML** - Machine Learning

**BI** - Sistemet e Inteligjencës në biznes

**ETL** - Mjetet e nxjerrjes, transformimit dhe ngarkimit të të dhënave (Extract, Transform, and Load)

**CRM** - Sistemet e menaxhimit të marrëdhënies me klientin (Customer relationship management)

**NN** - Rrjetat Neurale

**KNN** - K Nearest Neighbours

**SVM** - Support Vector Machines

**DW** - Magazinë të dhënash (Data Warehouse)

**OLAP** - Sistemet e përpunimit analitik në kohë reale (Online Analytical Processing)

**OLTP** - Sistemet e procesimit transaksional në kohë reale (Online Transactional Processing)

**MAE** - Gabimi Absolut Mesatar (Mean Absloute Error)

**MAPE** - Gabimi Absolut Mesatar në përqindje (Mean Absolute Percentage Error)

**ROLAP** – Sistemet relacionare të përpunimit analitik në kohë reale (Relational Online Analytical Processing)

**MOLAP** - Sistemet multidimensionalë të përpunimit analitik në kohë reale (Multidimensional Online Analytical Processing)

**HOLAP** - Sistemet hibride të përpunimit analitik në kohë reale (Hybrid Online Analytical Processing)

**TID** - Numri i transaksionit (Transacation ID)

**POS** - Pikë Shitje (Point of Sale)

**PAM** - Particionim rreth të mesmes (Partition Around Medoids)

**BIRCH** - Klasterizimi hierarkik multifazë duke përdorur pemët e klasterizimit (Multiphase Hierarchical Clustering Using Clustering Feature Trees)

**ReLU** - Njësia Lineare e Korrigjuar (Rectified Linear Unit)

**MLP** - Algoritmi Perceptron më shumë shtresa (Multilayer Perceptron)

**SSE** – Shuma e gabimeve në katror (Sum of Squared Errors)

## HYRJJE

Vitet e fundit mund të themi me bindje të plotë që kryefjala e çdo biznesi dhe industrie janë të dhënat. Me zhvillimin dhe përparimet e teknologjisë sasia e të dhënave që prodhohen çdo ditë nga kompanitë është shumë e madhe. Problemi i vërtetë qëndron në tërheqjen e informacionit sa më të saktë dhe të lexueshëm në gjuhën e biznesit, nga këto sasi të mëdha të dhënash. Një tjetër sfidë është parashikimi i ecurisë së biznesit bazuar në këtë informacion. Teknikat kryesore në realizimin e këtij procesi janë Data mining (DM) dhe Machine Learning ML.

Data mining (DM) është pikërisht procesi i identifikimit dhe ekstraktimit të informacionit të nevojshëm. Kjo teknikë ka për qëllim të zbulojë automatikisht njohuritë që mund të përftohen nga këto depozitorë të mëdhenj të dhënash dhe që nuk janë lehtësisht të dallueshme përmes teknikave të tjera rutinë të eksplorimit ose përpunimit të të dhënave.

Nga ana tjetër Machine Learning (ML) e përdor këtë informacion për të trajnuar makinën për të kryer parashikime për të ardhmen. Një përkufizim i ML do të ishte: Një program kompjuterik mëson nga ekperiencia E, në lidhje me një detyrë T dhe disa indikatorë performance P, nëse performanca e tij për detyrën T, e matur nga P rritet me rritjen e E (Tom M. Mitchell).

Machine Learning dhe Data Mining janë fusha të cilat ndërlidhen me njëra tjetrën dhe nuk është shumë e thjeshtë të bësh një ndarje të qartë. Data Mining përdor teknikat ML për të analizuar të dhënat dhe nga ana tjetër ML përdor informacionin e ekstraktuar nga DM për të trajnuar makinën. Nje ndarje intuitive qëndron në faktin që DM është më së shumti një proces i pambikëqyrrur (përshkrimor), ndërsa ML është një proces i mbikëqyrrur (parashikues). Për thjeshtësi gjatë këtij studimi do ta quajmë DM si kategorinë e madhe ku bëjnë pjesë teknikat përshkrimore dhe parashikuese. Duhet të kemi të qartë se teknikat parashikuese janë në thelb teknika ML.

Teknikat më të përdorura të DM dhe ML në aplikimet e ndryshme në biznes janë:

- Klasifikimi
- Asocimi
- Ndarja në klastera
- Teknikat e Regresit
- Rrjetat neurale
- Support Vector Machines (SVM)
- Algoritmet gjenetikë
- Detektimi i anomalive
- Ekstraktimi i karakteristikave
- Analizat e serive kohore



Biznesi Retail është një ndër bizneset me përhapjen më të madhe në Shqipëri. Në vitet e fundit, blerjet janë përqendruar me së shumti nëpër qendra tregtare ku shumë dyqane të kategorive të ndryshme operojnë në të njëjtin ambient. Kjo menyrë tregtie e kombinuar me e-commerce është aktualisht predominuese në Shqipëri edhe në botë. Ky ndryshim në menyrën e operimit në retail ka sjellë edhe rritjen e përdorimit të kartave të klientit, si menyrë për të krijuar një marrëdhënie më të ngushtë midis biznesit dhe klientit. Në këtë mënyrë, biznesi ka në dorë një informacion të detajuar të klientit, gjë që mungonte në mënyrën tradicionale të tregtisë. Nga ana tjetër, një biznes ka shumë sisteme të tjera nga të cilat merr të dhëna, siç janë sistemet transakcionale të shitjeve, sistemet e menaxhimit të stokut dhe sistemet ERP, sistemet e menaxhimit të promocioneve, sistemet CRM, sistemet e detektimit të prezencës të cilat mund të integrohen me anë mjeteve të inteligjencës në biznes (BI) dhe ti ofrojnë biznesit një sasi tepër të madhe të dhënash.

Me këtë volum kaq të madh të dhënash pyetja është një: Si mund të përdoren këto të dhëna në menyrë që të rritet cilësia e shërbimit ndaj klientit dhe të maksimizohet fitimi?

Pergjigja e kësaj pyetjeje, na çon para dy sfidave të mëdha. Si mund të ekstraktujmë informacion të vlefshëm nga kjo sasi e madhe të dhënash? A mund ti përdorim këto të dhëna që të parashikojmë ecurinë e biznesit?

Teknikat DM dhe ML i japin përgjigje këtyre dy pyetjeve. Por krijimi i modeleve të duhur nuk është i thjeshtë dhe saktësia e modeleve varet nga shumë faktorë duke u nisur që nga zgjedhja e attributeve të duhur që japin informacionin më të vlefshëm e deri të përdorimi i algoritmeve të duhur. Ndërtimi i modeleve DM dhe ML në Retail është një detyrë që ka nevojë për njohuri të thella të fushës së biznesit në mënyrë që të kuptojmë atributet më të vlefshme të të dhënave por në të njëjtën kohë ka nevojë për njohuri të thella teknike mbi shkencën e të dhënave në menyrë që të implementojmë algoritme me performancë të lartë.

Ky është qëllimi primar i punës tonë, të ofrojmë modele të përmirësuara DM dhe ML me performancë të lartë që mund të përdoren në biznesin retail për të analizuar dhe parashikuar ecurinë e biznesit. Këto modele mund të përdoren në veçanti nga departamenti i shërbimit ndaj klientit për të ofruar një cilësi më të lartë shërbimi, nga departamenti i marketingut për të krijuar fushata të targetuara në baze të profileve të klientëve, nga departamenti i blerjeve për të parashikuar ecurinë dhe shitmërinë e stokut dhe nga menaxhimi për të analizuar në shumë dimensionet ecurinë e biznesit dhe për të parashikuar të ardhmen e tij.

Në këtë punim, do të trajtohen disa prej mënyrave se si DM dhe ML (Machine Learning) integrohet në aplikime të tipit BI (Business Intelligence) dhe do të propozojmë modele të DM dhe ML të cilat do të vlerësohen në një dataset real Retail të një qendre tregtare në Tiranë. Qëllimi parësor i studimit mbetet analiza e aplikimit të sistemeve BI dhe teknikave të Data Mining në sistemet Retail në Shqipëri. Do të merret në konsiderantë në mënyrë të veçantë përdorimi i këtyre sistemeve për bizneset e vogla dhe të mesme që nuk janë pjesë e grupeve të mëdha ndërkombëtare.

Në punim integrohen të dhëna nga sisteme të ndryshme në qendrën tregëtare si: sistemi transakcional i shitjeve, një sistem e-commerce, sisteme CRM dhe ERP për menaxhimin e klientit. Pjesa eksperimentale do të ndahet në 3 pjesë kryesore. Në pjesën e parë do të propozohen dhe analizohen modelet grupuese në klastera dhe do të grupohen klientët në klasterat kryesorë. Në pjesën e dytë do të propozojmë dhe analizojmë modelet parashikuese të klasifikimit. Në pjesën e tretë do të propozojmë dhe analizojmë modelet e serive kohore me fokus në algoritmin Prophet dhe një modifikim të tij.

Punimi përbëhet nga pesë kapituj.

**Në kapitullin e parë** përkufizohen dhe studiohen sistemet e inteligjencës në biznes, historiku i tyre dhe karakteristikat kryesore. Arkitekturat e ndryshme të përdorura në sistemet BI dhe komponentet e tyre pasqyrohen dhe shpjegohen me detaje. Jepet lidhja ndërmjet sistemeve BI dhe magazinave të të dhënave si element kritik i tyre. Gjithashtu përshkruhen faktorët kryesorë që duhen marrë parasysh në implementimet e sistemeve BI.

**Në kapitullin e dytë**, analizohen magazinat e të dhënave DW dhe arkitekturat kryesore të tyre, përkatësisht Star dhe Snowflake. Analizohen avantazhet dhe disavantazhet e secilës arkitekturë. Në punim shpjegohet koncepti i kubave të të dhënave dhe arsyet e përdorimit të tyre në sistemet BI. Përshkruhen tre teknologjitë kryesore OLAP (ROLAP, MOLAP, HOLAP) dhe karakteristikat e tyre. Bëhet studimi teorik i performacës së secilës teknikë dhe gjithashtu jepen rekomandime në implementimin e sistemeve OLAP. Diferencat kryesore të sistemeve DW dhe relacionale trajtohen në detaje.

**Në kapitullin e tretë** kryhet një studim i teknikave të Data Mining (DM). Fillimisht analizohen modelet kryesore të DM, përkatesisht: modelit përshkrues dhe modelit parashikues. Algoritmat kryesore të secilit model trajtohen në detaje bashkë me analizat përkatëse. Shpjegohet se si kryhet integrimi i DM me sistemet e inteligjencës në biznes dhe statusi aktual i këtyre integrimiëve. Së fundmi jepen hapat kryesorë që duhet të ndërmerren gjatë implementimit të DM me fokus në paraprocesimin e të dhënave, i cili ka rëndësi kritike.

**Në kapitullin e katërt** analizohen teknikat dhe algoritmat që implementohen dhe modifikohen gjatë pjesës eksperimentale. Fillimisht paraqiten algoritmat e klasterizimit dhe veçanërisht algoritmat K-Means, K-Medoids dhe algoritmat hierarkike. Aparati matematikor i secilit algoritëm analizohet duke u ilustruar me shëmbuj dhe skema. Më pas paraqiten teknikat e klasifikimit. Në këtë pjesë trajtohen nga ana teorike algoritmat KNN (Klasifikuesi i K-fqinjëve më të afërt), algoritmi SVM dhe rrjetat neurale.

Aparati matematikor i secilit algoritëm trajtohet në detaje.

Vlen për tu theksuar punimi fokusohet në rrjetat neurale (NN) duke qënë se besojmë se janë teknikat më premtuese të klasifikimit dhe regresit. Arkitekturat kryesore të rrjetave NN paraqiten në punim dhe fazat e përparme dhe të pas-shpërndarjes analizohen si dhe funksionet e ndryshme të aktivizimit dhe të gabimit që përdoren. Të gjitha këto funksione implementohen në modelin e krijuar në pjesën eksperimentale. Duhet të theksojmë se analiza teorike e teknikave të klasifikimit dhe e rrjetave neurale bazohet thelbësisht në librin *“Data Mining and Machine Learning Fundamental Concepts and Algorithms by Mohammed J. Zaki, Wagner Meira Jr, Cambridge University Press, 2020”* të referuar me numrin 33 duke i besuar cilesisë shumë të lartë të analizës që kryhet në të.

**Në kapitullin e pestë** puna eksperimentale u ndahet në 3 seksione kryesore. Në seksionin e parë implementohen modelet më të rëndësishme të klasterizimit për të kuptuar cili performon më mirë. Rezulton se për bazën tonë të të dhënave, modeli K-Medoids ishte më i përshtatshëm për shkak të pranisë së të dhënave periferike (outliers) me dizavantazhin e kompleksitetit në kohë. Në pjesën e dytë të studimit analizohen modelet e klasifikimit dhe propozohen tre modele të ndryshme të bazuar në KNN (K-Nearest Neighbours), SVM (Support Vector Machines) dhe rrjetat Neurale (NN). Vihet re se rezultatet më të mira i meren në rastin e modelit me SVM, 80.79% saktësi në parashikim por duhet të theksojmë se rezultatet janë të ngjashme. Në pjesën e tretë të studimit analizohen dhe propozohen modele që lidhen me seritë kohore me fokus kryesor në algoritmin Prophet. Propozohen disa ndryshime nga modeli bazë të cilat na dhanë

një përmirësim me 1.4 % të gabimit në parashikim dhe një MAPE<25% për parashikime 3-mujore të shitjeve të qendrës tregtare.

Punimi mbyllet me konkluzionet dhe puna në të ardhmen.

## MOTIVIMI

Përdorimi i mjeteve të inteligjencës në biznes dhe të data mining në Shqipëri është i ulët dhe avantazhet e këtyre teknikave njihen pak. Në veçanti në biznesin e vogël dhe të mesëm retail, këto teknika nuk përdoren ose përdoren shumë pak. Këto teknika do të ishin shumë të dobishme për të analizuar dhe parashikuar sjelljen e klientit në retail, fakt që rrjedhimisht do të rezultonte në rritjen e të ardhurave të biznesit dhe të cilësisë së shërbimit të klientit. Një tjetër sektor që do të përfitonte nga kjo analizë është marketing i cili mund të kryejë fushtata të drejtuara, duke u nisur nga modelet e zbuluara nga DM. Nga ana tjetër parashikimi i shitjeve ndihmon në një menaxhim dhe planifikim më të mirë të stokut.

Një tjetër shtysë për këtë punim ka qënë rritja galopante e sasisë së studimeve në Data Mining dhe Machine Learning në vitet e fundit. Një pjesë e mirë e punimit bazohet edhe te rrjetat neurale të cilat janë në fokus kryesor të studimeve në fushën e ML apo inteligjencës artificiale.

Në studimin tonë do të merret në konsiderantë, në mënyrë të veçantë, përdorimi i këtyre sistemeve në biznesin Retail në Shqipëri për bizneset e vogla dhe të mesme që nuk janë pjesë e grupeve të mëdha ndërkombëtare.

Analiza do të kryhet me të dhëna reale të marra nga sistemet e një qëndre tregtare në Shqipëri. Do të studiohen në veçanti teknikat e ndarjes në klastera, teknika të klasifikimit dhe të serive kohore. Do të vlerësohet efikasiteti dhe faktorët që e ndikojnë atë. Do të propozohen modele të ndryshme të optimizuara për të arritur saktësinë maksimale në parashikim.

## KAPITULLI 1

### 1. SISTEMET E INTELIGJENCËS NË BIZNES

#### 1.1 Hyrje

Në këtë kapitull do të japim një pamje të përgjithshme të sistemeve të inteligjencës në biznes të vazhduar nga një diskutim se nga çfarë përbëhet një sistem BI, karakteristikat e tij kryesore dhe rëndësinë e tij në biznes, lidhja e tij me magazinimin e të dhënave dhe raportimin, arkitekturat tipike të një sistemi BI të ndjekura nga cikli i jetës së një projekti për një sistem BI.

#### 1.2 Evolimi i sistemeve BI

Shumë njerëz në industri mendojnë se termi “Inteligjencë në biznes” u përdor nga “Howard Dresner” i kërkimeve Gartner në 1989, ku ai e quajti BI “një kategori e gjerë software dhe zgjidhjeve për mbledhjen, konsolidimin, analizimin dhe bërja e mundur e aksesit ndaj të dhënave si një mënyre për t’i lënë përdoruesit e një kompanie të marrin vendime më të mirë biznesi” (Gibson *et al*, 2004; Yeoh, 2008).

Gjithsesi sipas Yeoh(2008)[2] termi është përdorur për herë të parë nga “H.P.Luhn” në një editorial të IBM të titulluar “Një sistem inteligjencë në biznes”(Luhn,tetor 1958).

Më poshtë jepet përcaktimi origjinal i Luhn siç është përcaktuar në 1958:

“Biznesi është një grumbull aktiviteteve që kryhen për çdo lloj arsyeje, në shkencë, teknologji, tregti, drejtësi, qeveri, mbrojtje e të tjera. Fasiliteti i komunikimit që shërben në drejtimin e një biznesi (në kuptimin e gjerë) mund të përcaktohet si një sistem inteligjence. Nocioni i inteligjencës, gjithashtu përcaktohet këtu, në një kuptim më të përgjithshëm si “aftësia për të kuptuar lidhjet e fakteve të prezantuara në mënyrë të tillë që të drejtojnë veprimet drejt një qëllimi të caktuar”.

Megjithatë është pranuar gjerësisht se Howard Dresner ishte i pari që e familjarizoi botën me termin “Inteligjence në Biznes” kur ai e përdori BI si një mënyrë për të përshkruar “konceptet dhe metodat për të përmirësuar vendim-marrjen në biznes duke përdorur në sistemet e suportit që bazohen në fakte”.

#### 1.3 Përkufizimi i një sistemi BI

Inteligjenca në biznes ose BI si dhe terma të tjerë në këtë fushë nuk kanë një përkufizim standard siç kanë shumë terma shkencore. Megjithatë, shumica e literaturës BI kanë ardhur nga bota e biznesit, industria IT dhe shitës të ndryshëm (Jagielska *et al* 2003; Yeoh, 2008).

Më poshtë po rendisim disa përcaktime [1]:

Davenport (2007): “BI është një set teknologjish dhe procesesh që përdorin të dhënat për të kuptuar dhe analizuar performancën e një organizate”.

Platon A/S (2009) [7] termi BI mbulon “përdorimi i informacionit për të thelluar njohuritë në biznes. Pra shërben për të themeluar një bazë më të mirë për vendim-marrësit duke bërë të mundur informacionin në formën e duhur, në cilësinë e duhur dhe në kohën e duhur”

Elizabeth Vitt (2002): termi BI “përdoret nga shumë shitës të produkteve software për të karakterizuar një gamë të gjerë teknologjish, platformash software, aplikacione specifike dhe procese”. Megjithëse është pothuajse e qartë shumë njerëz e kanë parë dhe shprehur BI në mënyra të ndryshme dhe kuptime të ndryshme.

## 1.4 Sistemet BI

Sipas Olszak & Ziembra (2007)[1], sistemet e inteligjencës në biznes ofrojnë një propozim që mbulon nevojat e organizatave kontemporane. Çështjet kryesore që mbulojnë këto sisteme përfshijnë eksplorimin inteligjent, integrimin, agregimin dhe analizën multidimensionale të të dhënave që vijnë nga burime të ndryshme informacioni. Sistemet standarde BI kombinojnë të dhëna nga sistemet e brendshme të informacionit të një organizate dhe integrojnë të dhënat që vijnë nga sisteme specifike për shëmbull statistika financiare dhe baza të dhënash të ndryshme. Sisteme të tilla shërbejnë për të ofruar informacion të përshtatshëm, të besueshëm dhe të azhornuar për aspekte të ndryshme të aktiviteteve të një sipërmarrjeje.

Siç e thekson NAV(2007), kur implementohet një sistem BI të jep mundësinë për të aksesuar, përdorur dhe ndarë të dhënat në një mënyrë eficiente dhe në këtë mënyrë ndihmon në rritjen e performancës së një biznesi.

Kapacitet e sistemeve BI i bëjnë të mundur një punonjësi:

- Të rendisë operacionet e përditshme me strategjitë dhe objektivat e përgjithshme të kompanisë
- Të identifikojë dhe kuptojë lidhjen midis proceseve të biznesit dhe impaktit të tyre në performancë
- Aksesin ndaj informacionit sipas roleve të ndryshme të përdoruesve dhe përgjegjësive
- Analiza e të dhënave nga dokumentat në një mënyrë të thjeshtë
- Përfton një vështrim kontekstual në mekanizmat e biznesit
- Monitoron indikatorët kryesorë në një biznes që nevojiten për ta zhvilluar një kompani që janë:
  1. Statusi aktual dhe trendi i propocioneve kryesore financiare
  2. Efektiviteti dhe përfutueshmëria e kanaleve të shitjes
  3. Metrikat e operacioneve kryesore

Pra inteligjenca në biznes ndihmon kompanite të kenë një pamje të kuptueshme dhe të integruar të biznesit të tyre dhe ta bëjnë më të lehtë dhe më të thjeshtë vendim-marrjen.

## 1.5 Arkitektura BI

Për David Loshin (2008) [3], objektivi i analizës së informacionit në biznes ishte dhe akoma është aftësia për të vlerësuar nëse kompania operon në mënyrë eficiente dhe rrjedhimisht kërkimi i mundësive për shfrytëzuar njohuritë. Ky nocion i menaxhimit të performancës dhe

përmirësimit nëpërmjet raportimit dhe analizave ka evoluar në atë që shpesh quhet inteligjenca në biznes. Megjithatë shumë prezantime, raportime dhe produkte virtualizimi ndihmojnë në komunikimin e rezultatit të kësaj analize, inteligjenca në biznes mbetet në mënyrë të pandarë i lidhur me infrastrukturën teknike dhe DW [2].

Sfidat taktike për të përcaktuar një plan pune për një sistem të inteligjencës së biznesit përfshijnë koleksionimin e të dhënave nga sisteme të ndryshme të shpërndara dhe duke i konsoliduar ata në një model të centralizuar dhe organizimi i të dhënave për të furnizuar aplikacionet fundore për të drejtuar analizat e biznesit dhe raportet.

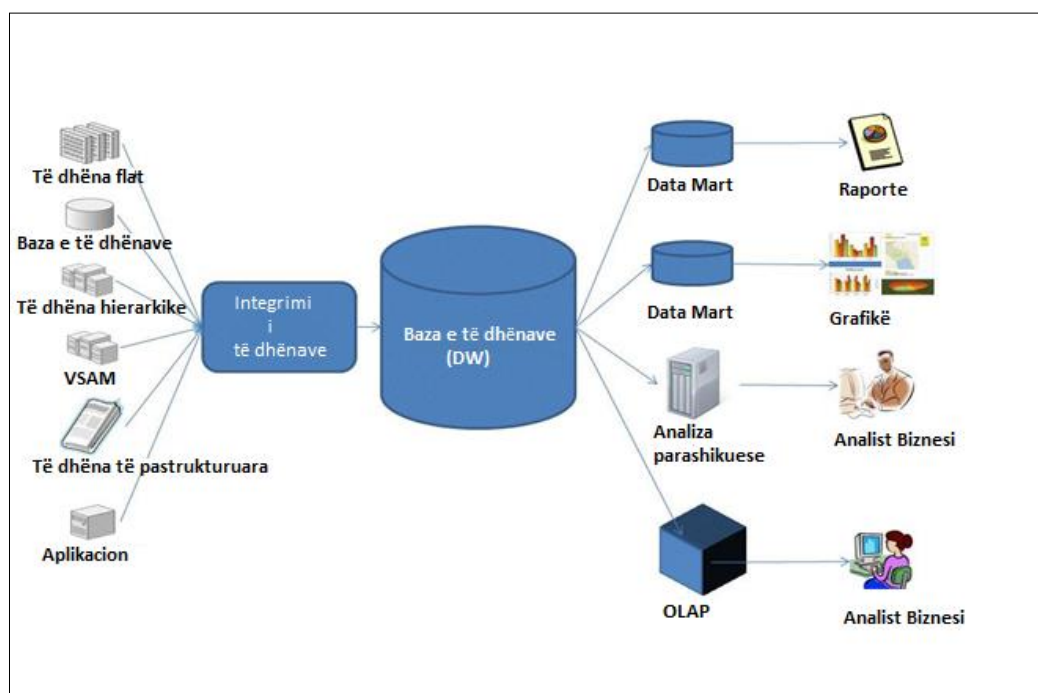


Figura 1-1 Arkitektura klasike e BI

Siç shihet në figurën 1-1 më BI ka nevojë për konektorë të dhënash për të marrë të dhënat nga burimet, mjete integrimi për të asimiluar dhe konsoliduar të dhënat, servera dhe memorie për të hostuar magazinën e të dhënave dhe sistemin e saj të meaxhimit, të gjitha aplikacionet që furnizojnë me të dhëna dhe ato fundore që përdoren për raportim dhe analiza dimensionale (David Loshin,2008) [3].

Në figurën 1.2 paraqiten të gjitha komponentet teknike të një sistemi BI që nga burimet e informacionit, mjetet për ekstraktimin, transformimin dhe hedhjen e të dhënave, llojet e memorieve që përdoren për të ruajtur të dhënat, shtresa semantike dhe llojet e raporteve dhe aplikimeve që i përdorin këto të dhëna.

Megjithëse burimet e informacionit mund të jenë të shumëllojshme, pjesa e përdoruesve përbëhet nga përdoruesit fundore, analistët e biznesit dhe administratorët.

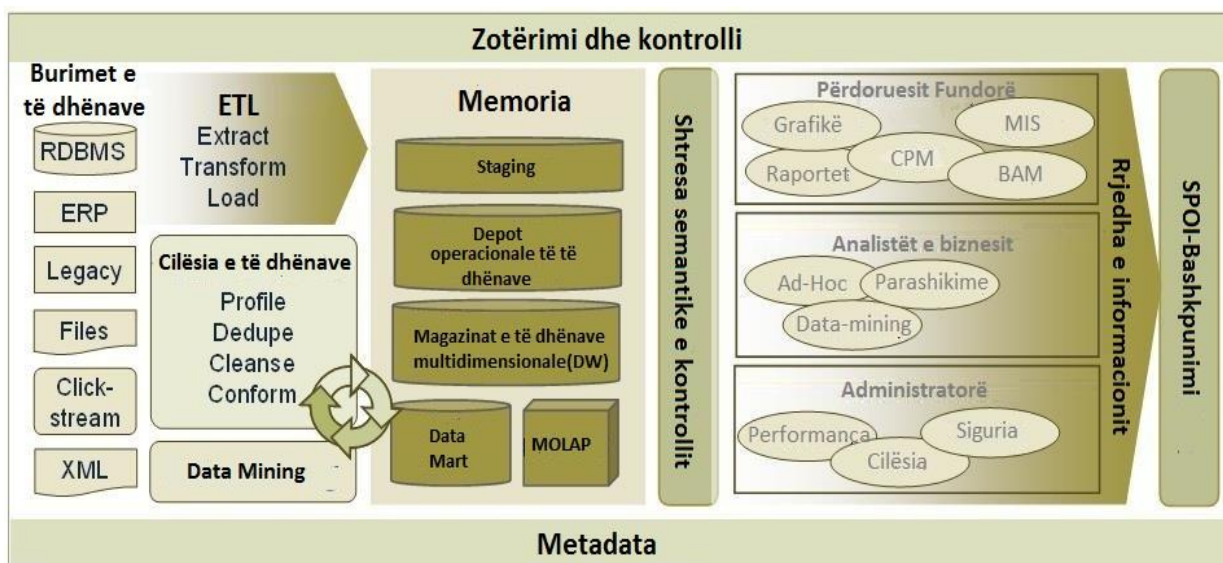


Figura 1-2 Arkitektura teknike e një BI

### 1.5.1 Komponentet e një sistemi BI

1. **Sistemet burim:** Konsiston në të gjitha të dhënat që mund t'i nevojiten një kompanie për analizat. Këto të dhëna mund të jenë të dhëna nga një bazë të dhënash burim, të dhëna të jashtme nga partnerët në format xml, të dhëna të marra nga website i kompanisë, një lidhje nga një server aplikacioni, të dhëna nga dokumenta në excel dhe burime të ndryshme (kJube 2002).
2. **Mjetet ETL:** ETL është procesi i marrjes së të dhënave, më së shumti nga lloje të ndryshme sistemesh, transformimi i tyre në një strukturë që është më e përshtatshme për raportim dhe analiza dhe futja e tyre në bazën e të dhënave. Figura 1.3 paraqet hapat e përfshira në këtë proces.

Në figurën 1.3 mund të shohim disa burime të ndryshme si sistemet e planifikimit të burimeve në një ndërmarrje ERP (Enterprise resource planning), sistemet e menaxhimit të klientëve CRM (Customer relationship management) dhe sistemet e linjave të biznesit LOB (Line of business).



Gjithashtu mund të ketë dhe lloje të tjera sistemesh ose dhe dokumenta të formateve të ndryshme që mund të integrohen nëpërmjet proceseve ETL në një magazinë të dhënash DW.

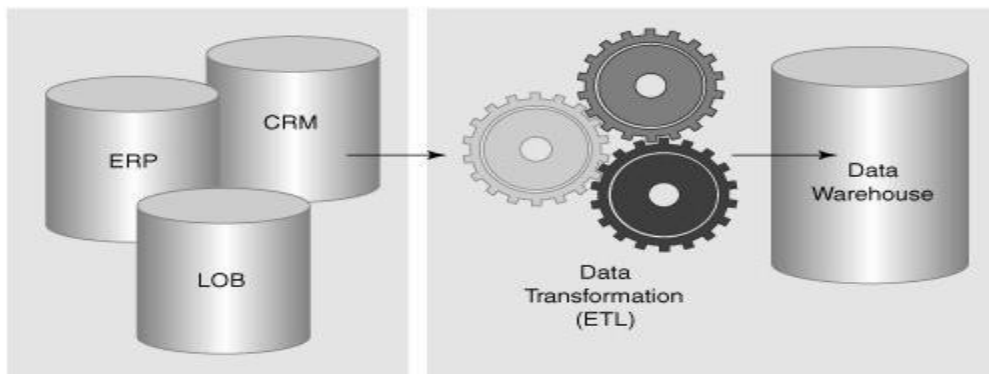


Figura 1-3 Data warehouse të ngarkuara nga sistemet burim

3. **Cilesia e të dhënave:** Sipas Passiones Nederland B.V (2009), sot ETL bën më shumë se për sa njihet. Ajo mbulon gjithashtu profilizimin e të dhënave, kontrolli i cilësisë së të dhënave, monitorimin dhe fshirjen, integrimin e të dhënave në kohë reale në një arkitekturë të orientuar nga shërbimi (SOA) dhe menaxhimi i Metadata-ve.
4. **Data Mining:** Për Kurt Thearling (2009), Data Mining, ekstraktimi i informacionit të fshehur parashikues nga baza të dhënash të mëdha është një teknologji e re me potencial të lartë për ti ndihmuar kompanitë të fokusohen në informacionin më të rëndësishëm në magazinat e tyre të të dhënave. Mjetet e data mining parashikojnë trendet dhe sjelljet e të ardhmes që e lejojnë një biznes të marrë vendime proaktive, të drejtuara nga njohuria. Analizat prospektive të automatizuara të ofruara nga data mining shkojnë përtej analizës së të dhënave historike që janë tipike në sistemet DSS. Mjetet data mining mund t'u përgjigjen pyetjeve të biznesit që mund të konsumojnë shumë kohë për tu zgjidhur.
5. **Organizimi i të dhënave:** Në procesin e magazinimit të të dhënave, pjesa e organizimit të të dhënave përbëhet nga një server aplikacioni dhe nga magazina e të dhënave (depo) e rezultateve të aktiviteteve të përthithjes, transformimit dhe futjes të të dhënave. Serveri i aplikacionit të organizimit të të dhënave ruan dhe transformon në mënyrë provizore të dhënat e marra nga burimet e të dhënave OLTP.
6. **Depot e të dhënave operacionale (ODS):** Sipas Immon (1998), një ODS është një strukturë e integruar, e orientuar drejt subjektit, jo e qëndrueshme (duke përfshirë përditësimet) e ndërtuar për tu shërbyer përdoruesve operacionale kur ata bëjnë procesime të integruara me performancë të lartë. Esenca e një ODS është bërja e mundur e procesimit të integruar dhe kolektiv në kohë reale. Një ODS suporton azhurnimin online. Një ODS është e integruar nëpërmjet shumë aplikacioneve. Në të njëjtën kohë një ODS suporton procesimin në ndihmë të vendim-marrjes.
7. **Magazina e të dhënave multidimensionale:** Siç e thekson kJube (2002), magazina e të dhënave multidimensionale është zemra e një mjedisi të inteligjencës në biznes. Në

themel ajo është një bazë të dhënash e madhe që mban të gjitha të dhënat që duhen për menaxhimin e performancës. Teknikat e modelimit që përdoren për të ndërtuar këtë bazë të dhënash janë kritike për funksionimin e një zgjidhjeje BI. Karakteristikë tipike e një magazine të dhënash multidimensionale është përmbajtja e të dhënave jo të ndryshueshme, të integruara dhe atomike. Duhet një njohuri të mira mbi teknikat e modelimit multidimensional për të kuptuar më mirë këto magazina të dhënash.

8. **Data Mart:** Një data Mart është një version më i vogël i një DW që kryesisht mban të dhëna të lidhura me një fushë funksionale të kompanisë me një shtrirje të limituar në një mënyrë tjetër. Mund të jetë një hap i nevojshëm drejt një magazine të dhënash.
9. **OLAP:** Sipas Rai&Storey (2001), procesimi analitik në kohë reale OLAP siç njihet është një teknologji për aksesin ad hoc në kohë reale të të dhënave dhe analizave.
10. **Shtresa Semantike:** Sipas Vezzosi (2009) [9], shtresa Semantike redukton kohën për sistemet e inteligjencës në biznes. Ajo ofron një akses të sigurtë të të dhënave me terma të thjeshtë biznesi. Mund të hapë akses për çdo klient ose përdorues apo çdo lloj të dhëne. Shtresat semantike shërbejnë si ndërmjetëse midis shtresës së prezantimit, e cila normalisht është ndërfaqja e përdoruesit për të gjithë përdoruesit fundor, analistët e biznesit dhe administratorët dhe zona e ruajtjes të të dhënave.
11. **Pikë e vetme e aksesit të informacionit:** Një zgjidhje BI mund të gjenerojë lloje të ndryshme outputesh. Këto outpute mund të përmbajnë tage, komente, interpretime që janë në formë të shumëfishta dhe duhet të ruhen në një hapsirë ku mund të lidhet me raportet dhe konsumojnë hapsira memorie të mëdha. Prandaj këshillohet të përdoret një pikë e vetme aksesi nëpërmjet një portali, ku është e mundur të lidhësh një përdorues biznesi, qëllimi i raportit dhe opsionet e aksesit.

## 1.6 Karakteristikat kryesore të një sistemi BI

Përgjithësisht karakteristikat i referohen tipareve të një entiteti të caktuar. Karakteristikat kryesore të një sistemi BI i referohen avantazheve që përfitohen kur implementohet një sistem BI.

Siç referon Dr.Saadia Asif (2009)[4], më poshtë renditen karakteristikat e një sistemi BI:

- Të dhëna të integruara nga burime të ndryshme
- Të dhëna të qëndrueshme përgjatë sistemeve të korporatës
- Të dhëna aktuale dhe historike
- Të dhëna të anlizuara përgjatë dimensioneve dhe hierarkive
- Dërgesa të shpejta të analizave

## 1.7 Drejtuesit e një implementimi BI

Sot menaxherët nuk bazohen më në instiktet e tyre për të marrë vendime të rëndësishme. Sistemet e inteligjencës së biznesit kanë filluar të përhapen dhe po luajnë një rol kyç në vendim-marrje.

“Sistemet BI kanë krijuar një reputacion dhe njihen si burime që dërgojnë raporte që askush nuk i lexon. Nuk duhet të jetë kështu. Dhe nuk duhet të lejohet që të jetë kështu.” (Grunman 2007)[5].

Grunman gjithashtu thekson se shumë organizata, duke e konsideruar BI si një set teknologjish ndërtojnë sisteme shumë komplekse që dështojnë në plotësimin e kërkesave të përdoruesve të tyre, ndërkohë që ajo që vërtetë është e nevojshme është një konceptim më i mirë i të dhënave dhe kërkesave të biznesit.

Si për Grunman(2007), Biznesi i përdor sistemet BI:

- Në 74 % të kompanive, menaxherët e konsiderojnë BI si një prioritet kyç. Ndër këta 48% drejtohen nga niveli ekzekutiv dhe 30% nga menaxhimi në linjën e biznesit.
- 74 % të kompanive e shohin BI si një fushë shumë prioritare në 3 deri në 5 vitet e ardhshme dhe 66% në vitin e ardhshëm.
- 61% e të pyeturve thonë se analistët e biznesit po luajnë një rol më strategjik në korporatat e tyre, 39% e këtyre korporatave planifikojnë të punësojnë më shumë analistë biznesi vitin e ardhshëm.

Gruman (2007) gjithashtu përmend si përdoret BI:

- Analizat parashikuese kanë një rol thelbësor në sistemet BI me 62 % të të anketuarve që i përdorin këto sisteme në këtë mënyrë ose planifikojnë ta bëjnë brenda një viti. Departamentet e shitjeve (56%), financat (54%), marketing (54%), logjistika (35%) dhe shërbimet ndaj klientit (35%) i përdorin më shumë sistemet e inteligjencës në biznes.
- Kapacitet më të rëndësishme të sistemeve BI janë renditja, filtrimi (60%) vlerësimi dhe optimizimi i vendim-marrjes (45%).
- 36% e të anketuarve kanë besim në saktësinë dhe rëndësinë e raporteve. Vetëm 10% nuk kanë besim në këto raporte.
- 54% e të anketuarve do të merrnin në konsideratë zëvendësimin e raporteve të gjeneruar nga BI me teknologjinë e kërkimit që i lejon përdoruesit të marrin rezultatet e tyre.

Por 43% e të anketuarve thonë se kompanitë e tyre nuk e kanë ekspertizën e duhur për ta bërë këtë.

Pse bota e biznesit është orientuar drejt sistemeve BI dhe kush janë forcat drejtuese të tyre, pse janë këto sisteme të nevojshëm dhe për çfarë qëllimesh.

Të gjitha këto ilustrohen në figurën 1.4.

Në këtë figurë tregohen disa nga nevojat e departamenteve të ndryshme të një ndërmarrje për një sistem BI.

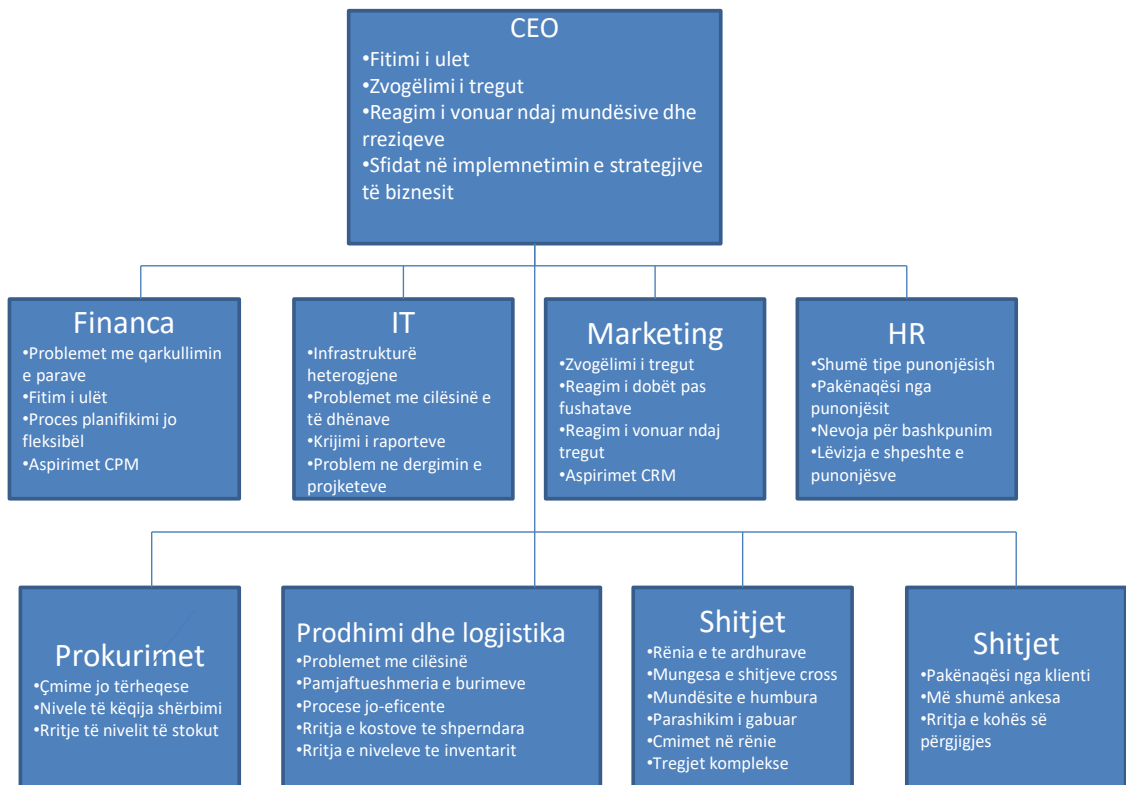


Figura 1-4 Arsyet që të çojnë drejt inteligjencës në biznes

Për NAV (2007) [6], e ndihmon në drejtimin e biznesit në një performancë më të mirë duke ju mundësuar të gjithë vendim-marrësve, kryesisht duke fuqizuar të gjithë punonjësit brenda një organizate të marrin vendime më të mira. Ka tre lloj vendimesh që një korporate merr:

- Vendime strategjike - këto janë vendimet më të rëndësishme që korporatat marrin (psh a duhet ta blejmë një partner, a duhet të hyjmë në një treg të ri). Vlera e këtyre vendimeve është e madhe por sasia e vogël.
- Vendime teknike - kjo është pjesa ku janë implementuar gjithnjë sistemet BI. Këtu një menaxher produkti vendos çfarë politikash uljesh duhet të ndjekë ose të vendosë mbi çmimin e një produkti të ri.
- Vendime operative - këto janë vendime biznesi të marra në vazhdimësi (zakonisht të bëra nga njerëz që nuk kanë dëgjuar ndonjëherë për sistemet BI) dhe kanë një impakt më të vogël në biznes. Gjithsesi në përgjithësi, shumë vendime operacionale marrin një vlerë shumë të madhe dhe mund të çojnë në një biznes më të mirë.

Për Platon A/S (2009)[7], drejtimet kryesore për një korporatë që të vendosë për implementimin e një sistemi BI duhet të jenë kërkesat ligjore dhe jo teknologjia ose iniciativa e IT.

Implementimi i sistemeve BI duhet të bëhet në bashkpunim të ngushtë me fushat e biznesit. Figura 1.5 ilustron vlerën e biznesit të sistemeve BI kundrejt kostos dhe kohës.

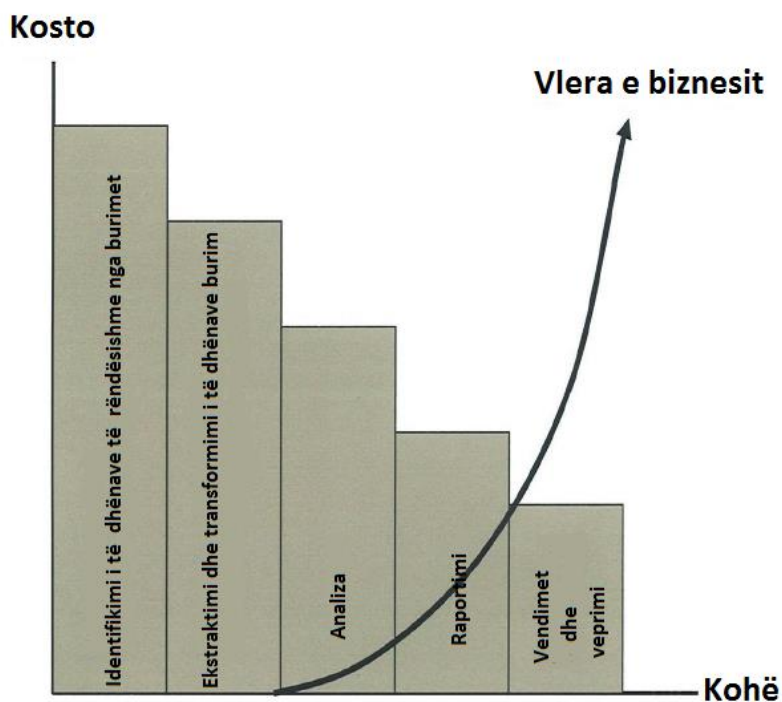


Figura 1-5 Vlerat e BI kundrejt kohës dhe kostos

## 1.8 Faktorët përcaktues në implementimin e sistemeve BI

Ka një numër faktorësh të cilët mund të ndikojnë në procesin e vendim-marrjes gjatë implementimit të një sistemi BI. Më poshtë kemi një listë të kategorizuar të faktorëve të cilët afektojnë procesin e vendim-marrjes të ndjekur nga aspektet e mëposhtëm:

### 1. Mjetet e raportimit dhe analizës:

- Veçoritë dhe funksionaliteti
- Përshkallëzimi dhe zbatimi
- Përdorimi dhe menaxhimi
- Aftësia për ta përshtatur

### 2. Bazat e të dhënave

- Përshkallëzimi dhe performanca
- Vlefshmëria dhe menaxhimi
- Siguria dhe përshtatje

- Aftësia për të rishkruar

### 3.Mjetet ETL

- Aftësia për të lexuar çdo burim
- Efiçenca dhe prodhueshmëria
- Suporti për platformat e përziera

### 4.Kostot e përfshira

- Kostot e hardware
- Kostot e software
- Kostot e brëndshme të implementimit
- Kostot e jashtme të implementimit
- Trajnimet e brëndshme
- Mirëmbajtja në vazhdimësi

### 5.Benefitet

- Të kursen kohë dhe efiçenca operacionale
- Kosto më të ulëta operacionale
- Shërbim më i mirë për përdoruesin
- Kosto më e ulët operimi
- Vendim-marrja operacionale dhe strategjike përmirësohet
- Një komunikim më i mirë midis punonjësve
- Një ndarje më e mirë e njohurive

## 1.9 Lidhjet midis sistemeve të inteligjencës së biznesit dhe magazinave të të dhënave

Shumë njerëz e ngatërrojnë fjalën BI (Business Intelligence) me atë të DW (DataWarehouse). Le të bëjmë një ekzaminim të lidhjes midis këtyre dy termave.

Sipas Platon A/S (2009) [7] termi Magazine Data (DW) mbulon menaxhimin e të dhënave ndërsa termi Inteligjencë në Biznes (BI) fokusohet në përdorimin e të dhënave. Qëllimi i një DW është të bëjë të mundur të dhëna të integruara, të qëndrueshme, të strukturuar, korrekte dhe në kohë për çdo sistem ose përdorues që kërkon informacion. Të dhënat ekstrahohen nga sistemet operacionale dhe integrohen në mjedisin e DW në mënyrë që ti ofrojë një ndërmarrjeje një perspektive më të gjerë dhe një version të të vërtetës. Ndërsa termi inteligjencë në biznes mbulon përdorimin e informacionit për të drejtuar vendim-marrjen e tij. Praktikisht është një mënyrë për të ofruar një bazë më të mirë për vendim-marrësit duke ofruar informacion në formën e duhur, në cilësinë e duhur dhe në kohën e duhur.

Siç dikton Dr.Saadia Asif (2009)[4] ”BI përdoret për shfrytëzimin e të dhënave ndërsa magazinat e të dhënave përdoren për integrimin e tyre” dhe thekson se ato nuk mund të përdoren në vend të njëra-tjetrës.

Më poshtë po paraqesim disa pikëvështrime të Moss (2003):

Sipas Larissa Moss, BI i referohet mundësisë për të siguruar një pamje 360° të biznesit dhe është një strukturë e disiplinave midis organizatave dhe një arkitekturë për ndërtimin dhe menaxhimin e një sërë aplikacionesh dhe bazash të dhënash operationale që suportojnë vendim-marrjen dhe i sigurojnë komunitet të biznesit akses të thjeshtë si dhe vendim-marrje të kujdesshme. Një mjet për të prodhuar inteligjencën në biznes është magazinimi i të dhënave; një tjetër mjet është CRM e kështu me radhë. Me pak fjalë magazinimi i të dhënave është një komponent dhe mjet për prodhimin e inteligjencës në biznes.

Adrienne Tannenbaumi referohet BI si një mënyrë për përdorimin e të dhënave ekzistuese/informacionit/njohurive në një korporatë. Në përgjithësi, shumica e të dhënave janë të ndryshme dhe intergohen në mënyrë sporadike (me një magazinë të vetme të dhënash). Në një strukturë BI e gjitha kjo bëhet pjesë e një sistemi më të madh dhe bëhet e mundur me anë të një kërkimi të vetëm. Dokumentimi, rregullat e biznesit, kriteret e kërkimit dhe raportet ekzistues janë shëmbuj të gjërave që mund të aksesohen nga një vënd i vetëm, në mënyrë tipike nëpërmjet termave të kërkimit të lidhura dhe të organizuara.

Sipas Scott Howard, BI i referohet sistemeve dhe teknologjive që i ofrojnë vendim-marrësve mënyrat për të ekstraktuar informacion të personalizuar dhe kuptimplotë për biznesin dhe industrinë e tyre, jo në mënyrë tipike vetëm nga sistemet e brendshme. Kjo përfshin mjetet për suportin e avancuar ndaj vendim-marrjes dhe baza të dhënash në të cilat mbështeten këto mjete. DW janë këto baza të dhënash. Nëse kombinohen me mjetet e suportit që duhen për ndërtimin dhe ruajtjen e bazës së të dhënave, sic është pastrimi, ekstraktimi, transformimi dhe ngarkimi i të dhënave, marrim atë që shumë e quajnë magazinimi i të dhënave.

Mendoni për magazinën e të dhënave si zyra e pasme dhe inteligjencën në biznes si të gjithë biznesin duke përfshirë dhe zyrën e pasme.

Clay Rehn thekson se magazinimi i të dhënave erdhi nga termat “suport në vendim-marrje” dhe “raportim menaxherial” shumë vite më parë. BI kërkoi të përmblihte më shumë procese që përfshinin magazinimin e të dhënave. Shumë shitës përdorin termin BI për të përshkruar shërbimet e tyre, për të treguar se ata ofrojnë më shumë shërbime dhe jo vetëm magazinim të dhënash.

## 1.10 Benefitet e implemetimit të një sistemi BI

Sipas Dr.Saadia Asif (2009)[4] implementimi i një sistemi BI mund të ofrojë:

### 1.Shtimi i të ardhurave

- Npërmjet përftimit të klientëve të rinj
- Rritja e shitjeve ndaj klientëve ekzistues

### 2.Reduktimi i kostove operative

- Rishpërndarja e burimeve në kanalet me kosto të ulët
- Kosto më të ulëta operative
- Shtimi i ofertave për produkte dhe përmirësimi i shërbimit ndaj klientit

### 3.Avantazhe kompetitive

- Një pjesë më e madhe e mendimeve dhe portofolit të klientëve duke identifikuar nevojat kryesore dhe mundësitë për shërbime.

Sipas Schiff (2009)[8] implementimi i një sistemi BI në një organizatë ndihmon për:

- Rritjen e efijensës dhe efikasitetit të organizatës
- Ndihmon organizatën në vendim-marrje
- Lejon përdoruesit e biznesit të analizojnë dhe kuptojnë më mirë planet dhe rezultatet e organizatës
- Paralajmëron organizatën për çështje potenciale kur ndodhin rethana të jashtëzakonshme-siç janë rënia e shitjeve me 20% nën parashikimet ose inventari që bie nën një vlerë kufi.
- 

### **1.11 Konkluzione**

Ne këtë kapitull u përkufizuan dhe studiuuan sistemet e inteligjencës në biznes, historiku i tyre dhe karakteristikat kryesore. Arkitekturat e ndryshme të përdorura në sistemet BI dhe komponentet e tyre u pasqyruan dhe shpjeguan në detaje. U shpjegua lidhja midis sistemeve BI dhe magazinave të të dhënave si element kritik i tyre. Gjithashtu u përshkruan faktorët kryesorë që duhen marrë parasysh në implementimet e sistemeve BI.



## KAPITULLI 2

### 2. MAGAZINAT E TË DHËNAVE DHE TEKNOLOGJIA E KUBAVE TË TË DHËNAVE

#### 2.1 Hyrje

Në një sistem bazë të dhënash të një korporate të gjerë është e vështirë analizimi i të dhënave sepse ato janë të ruajtura në formate dhe vënde të ndryshme. Për të zgjidhur problemin e integritimit të sistemeve të sipërmarrjes, data warehouse përdoren si një sistem i integruar.

#### 2.2 Përcaktimi i një Data Warehouse

Gjatë viteve të fundit institucionet po përdorin magazinimin e të dhënave për të përmirësuar fluksin e informacionit dhe suportin në vendim-marrje. Një DW mund të jetë një aset i vlefshëm që ofron akses të thjeshtë të të dhënave për analiza dhe raporte. Fatkeqësisht, ndërtimi dhe mirëmbajtja e një DW efektiv përmban disa sfida. Sot shumë organizata po mundohen të ndërtojnë një DW që përmbush kërkesat e biznesit të tyre.

Han dhe Kamber e përcaktojnë DW si një magazinë informacioni të marrë nga disa burime, të ruajtura në një skemë të unifikuar që zakonisht është e vendosur në një site të vetëm. Pra dhe në sistemet retail të gjitha të dhënat për shitjen, stokun, klientët, shitëset, artikujt, furnitorët do të ruhen në një vënd të vetëm.

Sipas Inmom “një DW është një koleksion të dhënash i integruar, i orientuar drejt subjektit, që ndryshon në kohë e cila vjen në ndihmë të procesit të marrjes së vendimeve të menaxhimit [9].

“Një DW është një magazinë e ndarë e të dhënave të marra nga një ose më shumë baza të dhënash produkti për të dhënë një burim autoritar për support ndaj vendim-marrjes. Ajo mund të konsiderohet një sistem që ndihmon në vendim-marrje. Raportimi i të dhënave është një tjetër aspekt i rëndësishëm i magazinimit të të dhënave sepse output kryesor i sistemeve DW janë query-it me formatim minimal ose raportet formale.

Veçoritë e DW mund të përmbledhen si më poshtë [10-12].

**I orientuar drejt subjektit:** DW projektohen për të ndihmuar dikë të analizojë të dhënat. Zakonisht këto lloj të dhënash nuk janë të përshtatshme për vendim-marrësit për tu përdorur. Vendim-marrësit kanë nevojë për të dhëna të orientuara drejt subjektit. Nëse do të duam të dimë për shitjet e një kategorie artikujsh të caktuar, do të ndërtohet një DW që do të përqëndrohet te shitjet. Duke përdorur këtë DW ne mund ti përgjigjemi pyetjes: Sa shitje ka pasur kjo kategori artikujsh vitin që shkoi? Aftësia për të përcaktuar një DW sipas subjektit (psh shitjet) e bën DW në këtë rast të orientuar drejt subjektit.

**E integruar:** Integrimi lidhet me orientimin drejt subjektit. DW duhet të vendosë të dhëna nga burime të ndara (siç janë dokumentat excel, bazat e të dhënave relacionale) në një format konsistent. Ato duhet të zgjidhin disa probleme siç janë konfliktet e emrit dhe paqëndrueshmërisë midis njësive të matjes. Kur arrihet ky qëllim thuhet se janë të integruara.

**Të qëndrueshme:** Të qëndrueshme do të thotë se të dhënat që futen në DW nuk duhet të ndryshojnë. Nuk ka azhornim të të dhënave si në një bazë të dhënash (OLTP) dhe nuk nevojiten mekanizmat e procesimit të transaksioneve dhe të rikuperimit. DW i duhen vetëm dy veprime në aksesimin e të dhënave: futja e të dhënave dhe aksesimi i tyre. Kjo është logjike sepse qëllimi i një magazine të dhënash është që të bëjë të mundur analizimin e veprimeve që kanë ndodhur. Kjo është arsyeja pse nuk janë të nevojshme mekanizmat e procesimit të transaksioneve, të rikuperimit dhe kontrollit të të dhënave.

**Variabël në kohë:** Në mënyrë që të zbulohen trendet në një biznes, analizuesve u nevojiten sasi të mëdha të dhënash. Kjo është e kundërt me sistemet OLTP ku për shkak të performancës kërkohet që të dhënat historike të kalohen në një arkiv. Një DW fokusohet në ndryshimin përgjatë elementit të kohës.

### 2.3 Një krahasim midis bazave të të dhënave për procesim transaksionesh dhe ato që ndihmojnë në vendim-marrje

Ka dy lloje kryesore të sistemeve të bazave të të dhënave: Sistemet operacionale dhe sistemet analitike që përfshijnë DW. Këto sisteme ndryshojnë në mënyrën se si i përdorin të dhënat. Sistemet operacionale quhen sisteme të procesimit të transaksioneve në kohë reale dhe janë të ndërtuara për futjen e të dhënave në kohë reale dhe modifikimin e tyre. Një transaksion hyrje produktesh, një kartë klienti mund të modifikohet dhe fshihet në çdo kohë. Zakonisht transaksionet përcaktohen më parë dhe kanë nevojë për një bazë të dhënash për të bërë të mundur aksesimin e shpejtë. Sistemet analitike janë projektuar për të ndihmuar korporatën në vendim-marrje. Një DW azhornohe periodikisht dhe zgjidhjet e saj ofrojnë mundësinë për të bërë query, analiza trendi dhe raporte.

### 2.4 Teknologjia e kubave të të dhënave

Principi i kubave të të dhënave shpjegohet jo vetëm nga ana se si ata janë dizenuar por edhe se për çfarë ata përdoren. Gjithashtu shpjegohet marrëdhënia ndërmjet kubave të të dhënave, DW dhe bazës së të dhënave relacionale.

Një nga pikat më të forta të bazës së të dhënave OLTP është se ata mund të kryejnë sasi të mëdha transaksionesh të vogla, duke e mbajtur bazën e të dhënave në dispozicion dhe të dhënat janë të qëndrueshme gjatë gjitha kohës.

Normalizimi ndihmon në ruajtjen e të dhënave të qëndrueshme, por ajo gjithashtu paraqet një shkallë më të lartë të kompleksitetit të bazës së të dhënave, e cila shkakton baza të dhënash të mëdha për të kryer operacionet e përbëra të bashkimit. Në kontekstin e biznesit është e dëshirueshme që të ketë historik të të dhënave që mbulojnë transaksionet e kryera prej vitesh, e cila rezulton në një sasi të madhe rekordesh të bazës së të dhënave për t'u analizuar.

Nuk është shumë e vështirë të kuptojmë se problemet e performancës do të lindin kur të kryhet përpunimi analitik që kërkon bashkimin e bazave të të dhënave komplekse. Një çështje tjetër që ka të bëjë me analizën OLTP është se ajo kërkon queries mjaft komplekse, ato të bëra posaçërisht për çdo kërkesë, në mënyrë që të marrim rezultatet e dëshiruara.

## 2.5 Serveri OLAP

Një server OLAP është një mjet me kapacitete të larta përpunuese dhe përdoruesish të dhënash i dizenuar veçanërisht për të suportuar dhe operuar në struktura të dhënash multidimensionale. Një strukturë multidimensionale është caktuar në këtë mënyrë që çdo artikull të dhënash të vendoset dhe bashkangjitet bazuar në ndërprerjen e përbërësve të dimensioneve që e përcaktojnë atë artikull. Dizenjimi i serverit dhe struktura janë optimizuar për marrjen e informacionit në veprim të shpejtë në çdo drejtim, gjithashtu edhe për kalkulime dhe transformime të shpejta dhe fleksibël të të dhënave të papërpunuara bazuar në marrëdhënien e formuluar. Serveri OLAP mund të organizojë gjithashtu dhe informacionet multidimensionale të procesuar për të shpërndarë konsistencë dhe përgjigje të shpejta të vendosura nga përdoruesit fundorë, ose ai mund të popullojë strukturën e të dhënave në kohë reale nga databaza relacionale ose të tjera, ose një zgjedhje e të dyjave.

## 2.6 Metadata

Metadata është ai lloj informacioni që përshkruan të dhënat e ruajtura në një databazë dhe përfshin informacione si:

- Një përshkrim i tabelave dhe fushave në një data warehouse, duke përfshirë tipet e të dhënave dhe rangun e vlerave të lejuara.
- Një përshkrim të njëjtë të tabelave dhe fushave në databazën burim, me një planifikim të fushave nga burimi në warehouse.
- Një përshkrim se si të dhënat janë transformuar, duke përfshirë formulimin, formatimin, shndërrimin korrent dhe grumbullimin në kohë.
- Çdo informacion tjetër që nevojitet për të suportuar dhe menaxhuar operacionet e data warehouse.

## 2.7 Drill-down

Drill-down mund të përcaktohet si kapaciteti për të shfletuar në informacion, duke ndjekuar një strukturë hierarkike. Në Figurën 2.1 tregohet një e tillë e thjeshtë.

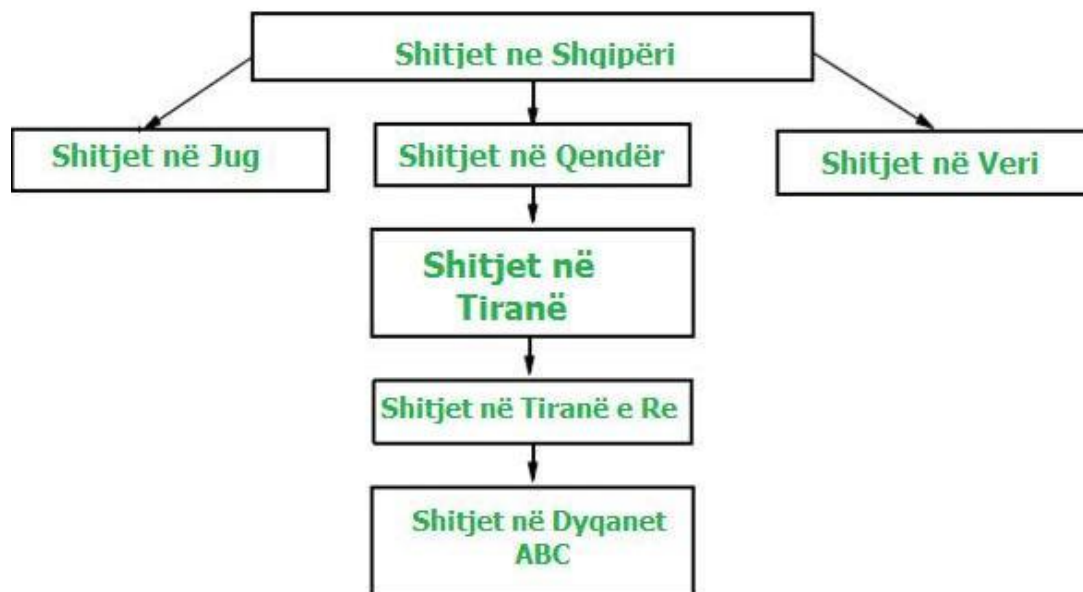


Figura 2-1 Drill-down

## 2.8 Data mining

Data mining është procesi i ekstraktimit të informacionit nga të dhëna të vlefshme, të përdorshme, të panjohur më parë e të krahasueshme dhe përdorimi i tyre për të hartuar vendime biznesi [10].

## 2.9 OLAP dhe Data Warehousing

Në mënyrë që të zgjidhen çështjet e mësipërme, koncepti i përpunimit analitik Online (OLAP) u propozua dhe u diskutua gjerësisht ndër vite dhe shumë artikuj janë shkruar mbi këtë temë. OLTP është përdorur shpesh për të trajtuar sasi të mëdha të transaksioneve të shkurtra dhe të përsëritura në një rrjedhë të vazhdueshme, të tilla si transaksione bankare apo rekordet. Sistemet e bazave të të dhënave janë të projektuar për të mbajtur të dhëna të qëndrueshme dhe për të maksimizuar shpejtesinë e transaksioneve [13].

Bazat e të dhënave OLAP nga ana tjetër përdoren për të ruajtur historikun e të dhënave për një periudhë të gjatë të kohe, shpesh të mbledhura nga burime të dhënave të ndryshme dhe me madhësinë e një baze të dhënash tipike OLAP është shpesh herë më e madhe në amplitudë se ajo e një baze të dhënash të zakonshme OLTP.

Bazat e të dhënave OLAP nuk azhurnohen vazhdimisht, por mund të ngarkohen në një bazë të rregullt çdo natë, çdo fundjavë, ose dhe në fund të muajit. Kjo çon në transaksione më të pakta dhe koha e përgjigjes së query është më e rëndësishme se transaksioni pasi querying është përdorimi kryesor i një baze të dhënash OLAP.

Thelbi i teknologjisë OLAP janë kubat e të dhënave, të cilët janë modele bazash të dhënash shumë dimensionale. Modeli përbëhet nga dimensione dhe parametra numerike të cilave u referohemi si masa. Masat janë të dhëna numerike të tilla si të ardhurat, shpenzimet, shitjet and buxheti. Ata varen nga dimensionet, të cilat përdoren për të grupuar të dhëna të ngjashme me

grupin nga operatori në bazat e të dhënave relacionale. Dimensione tipike janë koha, vendi dhe produkti dhe janë të organizuara shpesh në hierarki. Një hierarki është një strukturë që përcakton nivelet e përshkallëzimit të një dimensionit dhe lidhjet midis këtyre niveleve. Dimensionit kohë për shembull mund të ketë orën si nivelin e parë më lart, hierarkia mund të përmbajë ditë, muaj dhe vite. Kur një kub kërkohet për një matje, e cila shtrihet nga një ose shumë dimensione mund të zgjidhet për të filtruar të dhënat. Kubat e të dhënave bazohen në data warehouse, i cili është një qendër e të dhënave që ngarkohen me të dhëna nga burime të shumta. Data warehouse prirën të jenë shumë të mëdha në përmasa dhe procesi i projektimit është një detyrë mjaft komplekse dhe kërkon kohë. Disa kompani mund të vendosin të kenë një bazë të dhënash në një vend, i cili është një depo e të dhënave e kufizuar në departamente. Data warehouse implementohen zakonisht si një bazë e të dhënave relacionale me tabela, të cilat grupohen në dy kategori: tabela dimensionit dhe tabela fakti. Një tabelë dimension është një tabelë që përmban të dhëna që përcaktojnë një dimension. Dimensionit kohë për shembull mund të përmbajë datat, emrat e ditëve të javës, numrat e javëve, emrat e muajve, numrat e muajit dhe vitit.

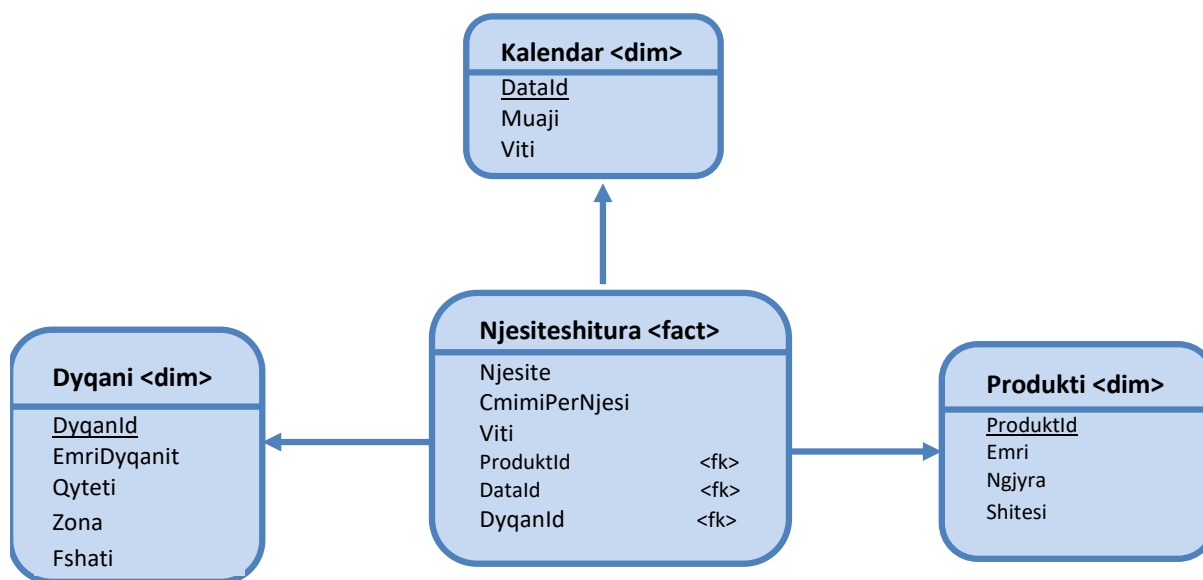


Figura 2-2 Shembull i skemës yll. Çelësat e jashtëm janë shënuar me <fk>

Një tabelë fakt përmban masat, që janë të dhënat të agregueshme që mund të numërohen, mbliidhen, shumëzohen, etj. Tabelat fakt gjithashtu përmbajnë referencat (çelësat e huaj) për tabelat dimension në kuba kështu që faktet mund të grupohen nga të dhënat dimensionale. Data warehouse në përgjithësi është i strukturuar si një skemë yll apo një skemë flokë dëbore (snowflake). Figura 2.2 ilustron një shembull të një data warehouse me strukturën e skemës yll. Siç shihet në figurë, një skemë yll ka një tabelë fakt në mes dhe të gjitha tabelat dimension i janë referuar kësaj table.

Me pak imagjinatë setup-i mund të mendohet në formë ylli. Në një skemë yll, tabelat dimension nuk kanë referenca në tabelat e tjera dimension. Nëse ata do ta bëjnë këtë, struktura do të quhej një skemë flokë dëbore. Një skemë yll përgjithësisht shkel 3NF duke patur tabela dimension të

bashkuara së bashku, e cila është shkak të humbjes së performancës që 3NF shkakton kur të dhënat janë shumë të mëdha. Në qoftë se për disa arsye është e dëshirueshme të mbahet data warehouse në 3NF mund të përdoret skema e flokë dëbore. Figura 2.3 ilustron një shembull se si Figura 2.2 mund të duket nëse do të jetë e strukturuar si një skemë flokë dëbore

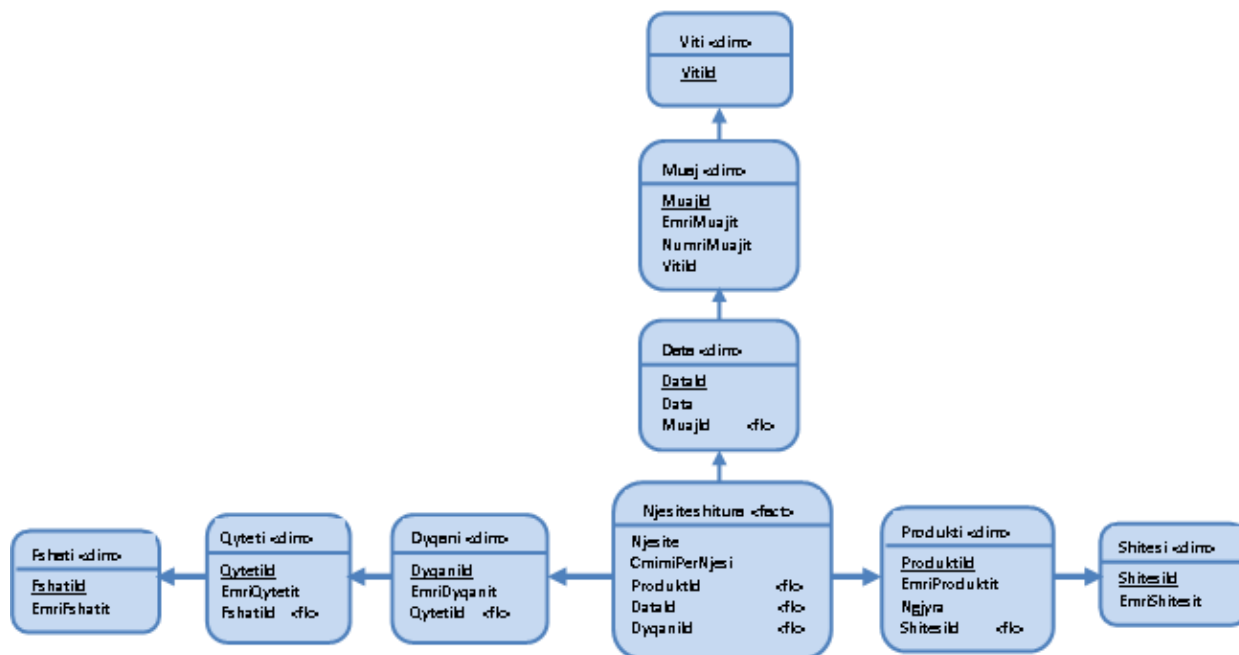


Figura 2-3 Shëmbulli se si Figura 2.2 mund të organizohet si skema snowflake. Çelësat e jashtëm janë shënuar me <fk>

Në 1993, Codd publikoi një artikull për nevojën e shërbimeve OLAP dhe kërkesat për OLAP të përmbledhur në dymbëdhjetë rregulla si më poshtë:

### 1. Pamja konceptuale shumë përmasore

Pamja e përdoruesve fundorë në modelin OLAP duhet të jetë shumë përmasore meqenëse analiza që do të bëhet nga natyra është shumë përmasore. Pamja shumë përmasore jo domosdoshmërisht nënkupton që të dhënat nuk ruhen në shumë përmasa.

### 2. Transparenca

Arkitektura OLAP duhet të jetë transparente, p.sh.faktet si, nëse arkitektura klient-server është pas aplikacionit, ose në cilën shtresë shtrihet funksionaliteti OLAP-it, duhet të jetë transparente për përdoruesin.

### 3. Aksesimi

Mjetet e OLAP duhet të jenë në gjendje të hartojnë skemën e tyre logjike në mbledhjen e të dhënave nga burime të ndryshme, heterogjene, relacionare dhe jo-relacionare të të dhënave. Përdoruesit fundorë duhet të jenë të lidhur me atë nga ku vijnë të dhënat ose se si do të ruhen fizikisht të dhënat, të dhënat duhet të shfaqen me pamje të vetme dhe konsekuente.

#### 4. Raportimi konsekuent i performancës

Kur numri i dimensioneve dhe madhësia e bazës së të dhënave rritet, përdoruesi nuk duhet të vërejë ndonjë raportim të rëndësishëm për degradimin e performancës. Në rastet kur performanca bëhet një çështje, mjetet e OLAP mund të propozojnë strategji alternative të tilla si paraqitjen e informacionit në ndonjë mënyrë tjetër.

#### 5. Arkitektura klient-server

Është e detyrueshme për mjetet e OLAP të jenë në gjendje të veprojnë në një mjedis klient-server pasi shpesh kubat e të dhënave janë depo qendrore të cilat aksesohen në distancë nga aplikimet e klientit.

#### 6. Dimensionaliteti i përgjithshëm

Çdo dimension i të dhënave duhet të jetë ekuivalent me kapacitetet si në strukturë ashtu edhe në operim.

Mund të shtohen karakteristika të tjera për dimensione të veçanta, por struktura themelore duhet të jetë e njëjta për të gjitha dimensionet dhe kështu karakteristikat e tjera duhet të jenë të mundura për çdo dimension.

#### 7. Dynamic Sparse Matrix Handling

Një kub tipik i të dhënave është shumë i rrallë. Prandaj, mjetet e OLAP duhet të trajtojnë matricat në mënyrë dinamike për të shmangur faktin që madhësia e kubit të rritet pa qenë e nevojshme. Nuk është e nevojshme të llogaritet agregimi për çdo qelizë të mundshme në kub në qoftë se vetëm një pjesë e vogël e qelizave në të vërtetë përmbajnë të dhëna.

#### 8. Suporti shumë-përdorues

Mjetet e OLAP duhet të sigurojnë akses të njëkohshëm për të dhënat dhe modelin OLAP, në mënyrë që të ruhet integriteti dhe siguria.

#### 9. Operacionet e pakufizuara ndër-dimensionale

Një mjet i OLAP duhet të jetë në gjendje të bëjë llogaritje dhe operacione të tjera në të gjithë dimensionet pa i kërkuar përdoruesit të përcaktojë në mënyrë eksplicite llogaritjen aktuale. Mjetet duhet të sigurojnë disa gjuhë për përdoruesit për t'i shfrytëzuar ato në mënyrë që të shprehin operacionet e dëshiruara.

#### 10. Manipulimi i të dhënave

Operacionet Drill-down, roll-up dhe të tjera që shtrihen në natyrën hierarkike të dimensionit duhet të kenë akses të lehtë me anë të manipulimit të drejtpërdrejtë të shfaqjes së të dhënave dhe nuk duhet të kërkojnë operacione të panevojshme të ndërfaqes së përdoruesit të tilla si menuja e navigimit.

#### 11. Raportimi fleksibël

Raportimi duhet të jetë fleksibël në kuptimin që rreshtat shtyllat dhe header-at e faqeve në raportin përfundimtar duhet të jenë në gjendje të përmbajnë numrin e dimensioneve nga modeli i të dhënave dhe çdo dimension i zgjedhur duhet të jetë në gjendje të shfaqë anëtarët e saj dhe lidhjet midis tyre.

#### 12. Dimensionet e pakufizuara dhe nivelet e agregimit

Në artikull Codd shprehet se një mjet i rëndësishëm i OLAP duhet të jetë në gjendje të trajtojë së paku pesëmbëdhjetë dhe mundësisht njëzet dimensione brënda të njëjtit model. Ndoshta ky rregull duhet të shtrihet për të lejuar një sasi të pakufizuar dimensionesh, ashtu siç nënkupton emri i rregullit. Në të dyja rastet, është deklaruar se çdo dimension duhet të lejojë për një sasi të pakufizuar përdoruesish nivele të përcaktuara agregimi brenda hierarkisë dimension.

Është e qartë se OLAP nuk duhet të implementohet si një teknologji e re e bazave të të dhënave meqenëse modeli relacionar i bazës së të dhënave është "teknologjia më e përshtatshme për bazat e të dhënave për një ndërmarrje". Gjithashtu theksohet se modeli relacionar kurrë nuk ka për qëllim të ofrojë shërbimet e mjeteve OLAP, por që këto shërbime duhet të shtohen nga mjetet e përdoruesve fundorë që të plotësojnë sistemet RDBMS me analizën e të dhënave në kohë reale.

## 2.10 Arkitektura e kubave

Edhe pse parimet themelore të kubave janë të njëjta, kubat mund të zbatohen në mënyra të ndryshme. Shitës të ndryshëm përkrahin arkitektura të ndryshme dhe disa ofrojnë mundësinë për të zgjedhur një arkitekturë për çdo kub që krijohet. Ka dy arkitektura kryesore që tradicionalisht janë diskutuar në fushën e bazës së të dhënave; *Multidimensionale* OLAP (MOLAP) dhe *Relacionare* OLAP (ROLAP).

MOLAP bazohet mbi filozofinë që kubat janë shumëdimensional në natyrën e tyre, të dhënat duhet të ruhen në mënyrë shumë dimensionale. Kështu, të dhënat kopjohen nga data warehouse në ruajtjen e kubave dhe agregohen sipas kombinimeve të ndryshme të dimensioneve të cilat janë të përlllogaritura dhe të depozituara në kuba sipas një strukture rrjet e bazës së të dhënave. Kjo do të thotë se koha e përgjigjes për query është shumë e shkurtër pasi nuk është e nevojshme të bëhen llogaritje në kohën kur bëhet query. Nga ana tjetër, ngarkimi i kubave është një proces i shtrenjtë për shkak të të gjitha llogaritjeve që duhen bërë dhe për këtë arsye të dhënat e kubave planifikohen të ngarkohen kur ato nuk kanë gjasa që të aksesohen, në intervale të rregullta si një herë në javë, në fundjavë apo çdo natë.

Një problem që duhet të merret parasysh kur punojmë me MOLAP është shpërthimi i të dhënave. Kjo është një dukuri që ndodh kur agregimi i të gjitha kombinimeve e dimensioneve duhet të llogariten dhe ruhen fizikisht. Për çdo dimension që shtohet në kub, numri i agregimeve që do të llogaritet rritet në mënyrë eksponenciale. ROLAP, ashtu siç dhe sugjeron emri, bazohet në modelin relacionar. Ideja kryesore është se është më mirë të lexohen direkt nga data warehouse, sesa të përdorin një tjetër lloj ruajtje për kubat. Ashtu si në rastin me MOLAP, të dhënat mund të agregohen dhe të pre-llogariten në ROLAP gjithashtu, duke përdorur pamjet e materializuara, pra ruajtja e agregimeve fizikisht në tabelat e bazës së të dhënave. Një arkitekturë ROLAP është më fleksibël pasi ajo mund të pre-llogarisë disa nga agregimet, por lë të tjerët të llogariten kur kërkohen. Gjatë viteve, ka pasur një debat të madh në fushën kërkimore nëse OLAP duhet të implementohet si MOLAP apo ROLAP. Debat i është venitur dhe në dekadën e fundit kërkuesit argumentojnë se ROLAP është më superior ndaj MOLAP. Argumentet janë se ROLAP performon pothuajse aq mirë sa MOLAP kur ka pak dimensione dhe kur ka shumë dimensione MOLAP nuk mund t'i trajtojë ato për shkak të shpërthimit të dhënave [17]. Tashmë për rreth 10-20 dimensione numri i llogaritjeve bëhet jashtëzakonisht i madh si pasojë e rritjes eksponenciale. Një tjetër aspekt i rëndësishëm është fleksibiliteti më i madh i ROLAP; të gjitha agregimet nuk kanë nevojë të llogariten paraprakisht, pasi ata mund të llogariten sipas kërkesave.



Vitet e fundit është propozuar dhe teknologjia HOLAP (Hybrid Online Transaction Processing) e cila është një kombinim i ROLAP dhe MOLAP.

HOLAP lejon ruajtjen e një pjese të të dhënave në një memorie MOLAP dhe një pjesë tjetër të dhënash në një memorie ROLAP duke lejuar një tradeoff të avantazheve të secilës [14].

## 2.11 Kubat relacionar të të dhënave

Një mënyrë për të implementuar kubat relacionare të të dhënave u propozua nga Gray më 1997. Operatori kubit - i cili është një përgjithësim n-dimensional i të njohurit *group by* - dhe operatorët e lidhur ngushtë me *rollup* përshkruhen në detaje. Kubat dhe operatori rollup janë implementuar në disa motorë bazash të të dhënave relacionare për të siguruar përdorimin e të dhënave të kubave. Marrim në konsideratë shembullin në Tabelën 2. Një kompani ka shitur xhinse dhe doreza me dy ngjyra, të zeza dhe blu. Shuma e përgjithshme e artikujve të shitur gjatë vitit 2017 dhe 2018 është përmbledhur për çdo vit si dhe ngjyra gjithashtu. Një menaxher ndoshta do të dëshironte të shikonte shifrat në trajtën e shitjeve totale për një vit, numrin total të xhinseve të shitura gjatë 2017, ose numrin e dorezave blu të shitura gjatë 2018.

Operatori kubit shpreh vendosjen e të dhënave në tabelën pivot në kontekstin e bazës së të dhënave relacionare. Ky operator në thelb është një set i *group by* të vendosura së bashku me një operacion bashkimi.

Tabela 2-1 Shembulli i tabelës së shitjeve për disa produkte të kompanisë

Produkti	Viti	Ngjyra	Shitjet
Xhinsa	2017	Të zeza	231
Xhinsa	2017	Blu	193
Xhinsa	2018	Të zeza	205
Xhinsa	2018	Blu	236
Doreza	2017	Të zeza	198
Doreza	2017	Blu	262
Doreza	2018	Të zeza	168
Doreza	2018	Blu	154

Një mënyrë intuitive për të arranzhuar të dhënat për këtë qëllim është përdorimi i tabelave pivot. Në këto tabela, të dhënat strukturohen sipas etiketave të rreshtave dhe kolonave dhe përmbledhet sipas dimensioneve në mënyrë që të kemi një pamje të qartë tyre.

Tabela 2.2 është shembulli se si Tabela 2.1 duhet të arranxhohet si një tabelë pivot  
Tabela 2-2 Shembull i tabelës pivot.

	2017		2018		2018		Totali përgjithshëm	i
	Të zeza	Blu	Total	Të zeza	Blu	Total		
Xhinsa	231	193	424	205	236	441	865	
Doreza	198	262	460	168	154	322	782	
Totali përgjithshëm	429	455	884	373	390	763	1647	

Ai grupon të dhënat sipas dimensioneve (atributeve) të dhënave, në këtë rast Produkti, Viti dhe Ngjyra dhe agregimet bëhen për të gjitha kombinimet e mundshme të dimensioneve. Tabela 2.3 ilustron të dhënat si rezultat kur operatori kub përdoret në kombinim me operatorin SHUMË.

Tabela 2-3 Rezultatet e operatorit të kubit në agregimin e operatorit shumë, kur aplikohet në Tabelën 2.1.

Produkti	Viti	Ngjyra	Shitjet	Produkti	Viti	Ngjyra	Shitjet
Xhinsa	2017	Të zeza	231	Doreza	2018	ALL	322
Xhinsa	2017	Blu	193	Doreza	2018	Të zeza	366
Xhinsa	2017	ALL	424	Doreza	2018	Blu	416
Xhinsa	2018	Të zeza	205	Doreza	2018	ALL	782
Xhinsa	2018	Blu	236	ALL	2017	Të zeza	429
Xhinsa	2018	ALL	441	ALL	2017	Blu	455
Xhinsa	ALL	Të zeza	436	ALL	2017	ALL	884
Xhinsa	ALL	Blu	429	ALL	2018	Të zeza	373
Xhinsa	ALL	ALL	865	ALL	2018	Blu	390
Doreza	2017	Të zeza	198	ALL	2018	ALL	763
Doreza	2017	Blu	262	ALL	2018	Të zeza	802
Doreza	2017	ALL	460	ALL	2018	Blu	845
Doreza	2018	Të zeza	168	ALL	2018	ALL	1647
Doreza	2018	Blu	154				

Vlerat speciale ALL janë futur për të treguar që rekorde të caktuara përmbajnë agregimin mbi atributet që kanë të njëjtën vlerë. Vlera NULL mund të përdoret në vend të vlerës ALL në mënyrë që të mos manipulohet gjuhën SQL. Operatori kub ka si rezultat shumë agregime, por të gjitha kombinimet e agreguara janë rallë të dëshiruara dhe prandaj u propozua dhe operatori rollup. Ky operator punon në të njëjtën mënyrë si operatori kub por me një ndryshim, agregimet e reja llogariten vetëm nga agregimet e llogaritura deri në atë moment.

Tabela 2-4 ilustron rezultatet që marrim nga operatori rollup. Algoritme të ndryshme janë zhvilluar në mënyrë që të përmirësohet performanca e operatorit të kubit. Të gjithë algoritmet shfrytëzojnë faktin që një agregim përdor operacionin group by (më sipër e quajtmë thjesht group by) në përgjithësi nuk ka nevojë të llogaritet nga relacioni aktual, por mund të llogaritet nga rezultati i një “group by” tjetër. Kështu që vetëm një “group by” duhet të llogaritet nga relacioni, ku njëra quhet granulariteti me i mirë dhe gjithë të tjerat mund të llogariten nga ky rezultat.

Tabela 2-4 Rezultatet e operatorit rollup kur aplikohet në Tabelën 2.1.

Produkti	Viti	Ngjyra	Shitjet	Produkti	Viti	Ngjyra	Shitjet
Xhinsa	2017	Të zeza	231	Doreza	2017	Të zeza	198
Xhinsa	2017	Blu	193	Doreza	2017	Blu	262
Xhinsa	2017	ALL	424	Doreza	2017	ALL	460
Xhinsa	2018	Të zeza	205	Doreza	2018	Të zeza	168
Xhinsa	2018	Blu	236	Doreza	2018	Blu	154
Xhinsa	2018	ALL	441	Doreza	2018	ALL	322
Xhinsa	2018	ALL	865	Doreza	2018	ALL	782
ALL	2018	ALL	1647				

Figura 2.4 ilustron se si rezultatet e “group by” mund të përdoret për të llogaritur agregimet e tjera. A, B, C dhe D janë atributet në relacion dhe çdo nyje në graf përfaqëson setin resultant të group by në atributet e nyjes. Atributi ALL në nyjen e pare tregon që nuk ka group by. Nyjet në fund përmbajnë atributet {A, B, C, D}, që tregon se nyja përfaqëson rezultatet e group by të katërtributeve. Duke përdorur këto rezultate, operacioni group by në {A,B,C}, {A,B,D}, {A,C,D} dhe {B,C,D} mund të llogaritet. Duke u ngjitur më lart në graf, shohim se rezultatet e “group by” në {A,B,C} mund të përdoren për të llogaritur rezultatet e operacionit “group by” në {A,B}, {A,C} dhe {B,C}, etj.

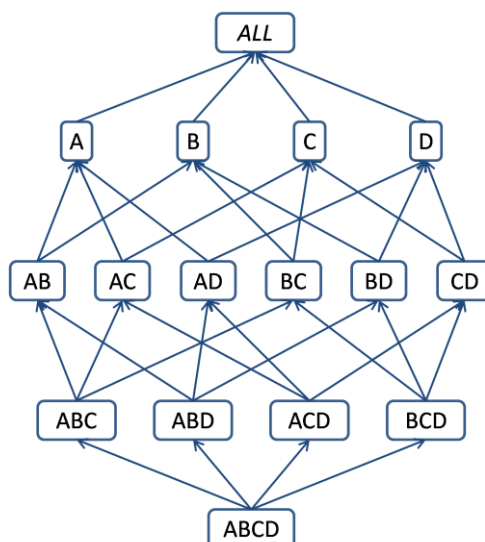


Figura 2-4 Grafi përshkruan se si operacioni group by mund të llogaritet duke përdorur operacionin group by të mëparshëm në vend të burimit të të dhënave

Kjo teknikë përmirëson performancën duke minimizuar diskun I/O por gjithashtu edhe duke minimizuar numrin e agregimeve duke mos agreguar të gjitha kombinimet e mundshme që janë rezultat i operatorit kub. Diferencat ndërmjet algoritmeve qëndrojnë në faktin se përdorin strategji të ndryshme për të zgjedhur se cili rezultat do të përdoret për llogaritjen e rezultateve të rejave, ose se si zgjedhin rrugë të ndryshme në grafën si në Figurën 2.4.

## 2.12 Performanca

Siç e përmendëm dhe më sipër, performanca është një çështje e madhe kur bëhet fjalë për një sasi të madhe të dhënash. Ka metoda që realizojnë përmirësimin e performancës së kubave të të dhënave, si për MOLAP dhe ROLAP.

### 2.12.1 Grupimet

Kur flasim për performancën e bazës së të dhënave, disku I/O është një problem meqenëse aksesimi i diskut shkakton vonesa më të mëdha se sa aksesimi i memories. Një zgjidhje e cila përdoret gjerësisht për minimizimin e aksesimit të diskut është grupimi (clustering). Principi bazë i klastering është organizimi i strukturës së diskut të të dhënave në mënyrë që të bëhet e mundur që rekordet të bëhen query dhe të ruhen fizikisht së bashku. Kjo do të thotë se blloqe të mëdha të dhënash të lexohen njëkohësisht, në vend të aksesimit për çdo rekord ,të cilët mund të jenë të shpërndarë në disk.

### 2.12.2 Indeksimi bitmap

Një mënyrë për të indeksuar bazën e të dhënave relacionare është përdorimi i pemës B+- për kërkimin dhe përdorimin e Row Identifiers (RIDs) në nyjet gjethe, për të specifikuar adresat fizike të diskut ku rreshtat e të dhënave shkruhen. Nëse të dhënat janë vetëm të lexueshme, si në bazën e të dhënave OLAP, performanca e query-it mund të përmirësohet duke përdorur

indeksin bitmap Index. Nëse të dhënat nuk janë vetëm të lexueshme si në rastin tipik të bazës së të dhënave OLTP, indeksi bitmap nuk është një ide e mirë për t’u përdorur pasi kërkon kosto të lartë për operacionet e futjes dhe nxjerrjes [15-16].

Ideja bazë e indeksit bitmap është se për çdo vlerë të mundshme në kolonën që do të indeksohet, bitmap krijohet me një bit për çdo rresht në tabelë. Në një bitmap të caktuar, të gjitha bitet që korrespondojnë me vlerën e përfaqësuar bëhet “1” dhe gjithë bitet e tjera bëhen “0”.

Tabela 2-5 Shembull i indeksimit bitmap.

Produkti	Ngjyra	B <sub>1</sub>	B <sub>2</sub>	B <sub>3</sub>
Xhinsa	Blu	1	0	0
Corape	Të zeza	0	0	1
T-shirt	Të zeza	0	0	1
T-shirt	Blu	1	0	0
Doreza	Të bardha	0	1	1

Tabela 2-5 ilustron idenë e indeksit bitmap; përmban pak produkte dhe ngjyrat e tyre, ku kolona e ngjyrës do të indeksohet. Për çdo ngjyrë të mundshme, krijohet një bitmap B<sub>i</sub> që shënon të gjitha njësitë në relacionin që ka këtë ngjyrë si vlerë, p.sh. të gjitha njësitë me vlerë atributi ‘Blu’ përfaqësohen nga B<sub>1</sub>=‘10010’. Duke patur parasysh se kardinaliteti i kolonës nuk është shumë i madh, p.sh. numri i vlerave të mundshme për atë kolonë nuk është shumë e madhe, mund të kemi dy tipe të ndryshme indeksimi bitmap. E para është që madhësia e kërkuar e diskut për të ruajtur indekset të reduktohet krahasuar me përdorimin e pemës B+- . Në shembujt e mësipërm, tre bitet që kërkohen për çdo vlerë, krahasohen me madhësinë e adresave të diskut kur përdoren RIDs. Përmirësimi i dytë është performanca. Operatorët AND, OR dhe NOT janë shumë të shpejtë dhe përdorimi i tyre bën të mundur ekzekutimin e query-it duke përdorur kushtin WHERE në mënyrë shumë efiçiente. Operacioni i agregimit si SHUMA dhe AVG afektohen gjithashtu nga zgjedhja e indeksit. Pra indeksimi bitmap nuk është mjaft i përdorshëm nëse vlera që do të indeksohet është një integer i cili mund të ketë qindra vlera të mundshme.

Në vend të tij përdorim një variant të quajtur indeksi bit-sliced që mund të përdoret në rastet kur duhet të ruajmë përfitimet nga indekset bitmap. Supozojmë se një kolonë objekt popullohet nga vlera integer të përfaqësuara nga N bite. Indeksi bit-slice përcakton një bitmap për çdo N bite, në vend të një bitmap për çdo vlerë të mundshme.

Tabela 2-6 ilustron idenë; sasia e artikujve të shitur për çdo produkt përfaqësohet nga kodimi binar me kolonën B<sub>3</sub> që përfaqëson bitin me më pak peshë dhe kolona B<sub>1</sub> përfaqëson bitin me më shumë peshë.

Duke lexuar këto kode binare me anë të kolonave të dhëna japin bitmap në indeks, p.sh. B<sub>1</sub> = ‘1100’.

Tabela 2-6 Shembull i indeksimit bit-sliced.

Produkti	Sasia e shitur	B <sub>1</sub>	B <sub>2</sub>	B <sub>3</sub>
Xhinsa	5	1	0	1
Corape	7	1	1	1
T-shirt	3	0	1	1
Doreza	2	0	1	0

Duke përdorur metodën e indeksimit, performanca e query rritet kur bëhen agregimit. Nëse skema relacionare ka kardinalitet të lartë, madhësia e indeksit të bitmap do të rritet më shumë sepse çdo vlerë e mundshme do të sjellë rritje në bitmapin tjetër, me aq bite sa njësi janë në relacion. Në të tilla raste ka teknika kompresimi që reduktojnë madhësinë e bitmap.

Dy skema të zakonshme kompresimi janë kodi byte-aligned bitmap (BBC) dhe kodi word-aligned hybrid (WAH). Kodi BBC bazohet në operacionet në byte dhe kodi WAH është një variant i BBC që bazohet në adresim fjalë. Në thelb, të dyja metodat punojnë duke gjetur sekuenca të vijueshme bitesh që kanë të njëjtën vlerë dhe i zëvendësojnë ato me vlera bitesh dhe një numërues që numëron se sa bite janë në sekuençë. Këto teknika kompresimi në rastin më të keq rezultojnë në indekse po aq sa ato të pemës B+- por shpesh edhe më të vogla. Megjithatë, shumë operacione logjike mund të kryhen pa dekompresuar të dhënat [15-16].

### 2.12.3 Copëzimi

Një teknikë tjetër që rrit performancën e query në OLAP, është vendosja në cache e rezultateve të query të kërkuar por në copëza në vënd të të gjithë rezultatit. Në një mjedis shumëdimensional, kërkesa për query të ndryshme rezulton në dimensione të ndryshme dhe shpesh rezultatet ndërpriten, duke rezultuar në të njëjtin subset. Duke përdorur cache-t e query-it, këto nënsete zakonisht nuk mund të vendosen në cache dhe ripërdoren nëse i gjithë query bëhet përsëri. Përdorimi i copëzave, pjesë më të vogla të secilit query vendoset në cache dhe kur kërkohet një query i ri mjeti i cache-së gjen se cila copëz ekziston në cache dhe i dërgon ato të query bashkë me copëzat e reja të llogaritura që nuk ekzistojnë në cache. Me anë të kësaj teknike, rezultatet e query mund të ripërdoren në mënyrë më eficiente, që do të thotë se të dhënat që kërkohen më shpesh duhet të ruhen në copeza në memorie dhe nuk kanë pse të lexohen nga disku për çdo query.

Optimizimi i mëtejshëm realizohet duke organizuar sistemin me file fundore me anë të copëzave, si dhe duke implementuar suportin për copezat e file-ve në DBMS ose duke lejuar RDBMS ekzistues të indeksojë file-t nga atributet e copëzave që u shtohen tabelave. Kjo është një formë e klastering që tregoi se ishte shumë efiçiente për të dhënat shumëdimensionale.

### 2.13 Konkluzione

Në këtë kapitull, u analizuan magazinat e të dhënave DW dhe arkitekturat kryesore të tyre, përkatësisht Star dhe Snowflake. U listuan avantazhet dhe disavantazhet e secilës arkitekturë. Më pas u shpjegua koncepti i kubave të të dhënave dhe arsyet e përdorimit të tyre në sistemet BI. Pamë dhe 3 teknologjitë kryesore OLAP (ROLAP, MOLAP, HOLAP) dhe karakteristikat e tyre. Kryem një studim teorik të performacës së secilës teknikë dhe gjithashtu dhamë rekomandime në implementimin e sistemeve OLAP. Diferencat kryesore të sistemeve DW dhe relacionale u listuan dhe shpjeguan.

## KAPITULLI 3

### 3. TEKNIKAT E DATA MINING

#### 3.1 Hyrje

Data mining është procesi i nxjerrjes në pah të informacionit të dobishëm dhe më të vlefshëm që mund të përftohet nga analiza e një sasive të madhe të dhënash. Ky proces përdor teknika të ndryshme të përpunimit të të dhënave, teknika statistikore, matematikore, ato të inteligjencës artificiale si dhe të mësimin automatik. Karakteristika kryesore e kësaj teknike është fuqia e saj parashikuese. Kjo veçori vjen si fryt i dizajnit unik duke kombinuar teknika nga mësimi automatik, njohja e sjelljes dhe statistika, me qëllim marrjen e informacionit rreth sjelljeve, koncepteve ose korelacionit midis elementeve (variablove) të ndryshëm nga depozitorët e të dhënave.

Biznesi përmes mënyrave më të zakonshme të raportimit arrin fillimisht të perceptojë disa sjellje të fshehura në indikatorë ose variabla kyç në të dhënat e tij. Pikërisht këto lidhje ose sjellje të fshehura shërbejnë si input për ndërtimin e modeleve DM, të cilët arrijnë të bëjnë më pas parashikime [20] të një saktësie të lartë në lidhje me performancën ose sjelljen e indikatorëve të ekstraktuar fillimisht nga baza e të dhënave.

DM përshkruese dhe ajo e parashikimit janë qasjet më të përdorura në zbulimin e informacionit të fshehur në sasi të mëdha të dhënash [19].

Data Mining është aktualisht një disiplinë e konsoliduar në shkencat kompjuterike. Origjina e kësaj discipline mund të përcaktohet në fundin e viteve 80 kur filloi të lakohet si terminologji fillimisht në rrethin e shkencëtarëve që merreshin me kërkime në këtë fushë.

Ashtu sikurse në fillimet e saj por edhe në ditët e sotme termi data mining lë shumë për të diskutuar. Në terma të përgjithshëm ai mund të shihet si një set mekanizmesh dhe teknikash, të jetësuar në një software, me qëllim ekstraktimin e informacionit të fshehur në një grumbullim të madh të dhënash.

Termi ‘i fshehur’ në këtë përkufizim është shumë i rëndësishëm; SQL, pavarësisht se është një teknikë e sofistikuar e marrjes së të dhënave, nuk është data mining.

Në fillimin e viteve 1990 data mining njihet më së gjeri si një nën proces në një zinxhir të madh procesesh të quajtur, zbulimi i informacionit në bazat e të dhënave (Knowledge Discovery in Databases) ose KDD. Përkufizimi më i përdorur për KDD është ai i cituar nga (Fayyad et al. 1996) si më poshtë:

KDD është procesi jo më pak i rëndësishëm i identifikimit të informacionit të vlefshëm, të përdorshëm dhe të kuptueshëm në lidhje me sjelljen e të dhënave.

Në tërë ciklin e KDD, data mining është një prej nën proceseve. Të tjerë nën procese që formojnë KDD janë përgatitja e të dhënave (ruajtja, kontrolli/pastrimi, para-proçesimi, etj) si dhe analiza/vizualizimi i rezultateve.



Shpesh KDD dhe DM shihen si sinonime të njëri tjetrit, por teknikisht njëri është nën proces i tjetrit. Ekzistojnë dy modele të rëndësishëm në Data Mining: modeli përshkrues dhe modeli parashikues.

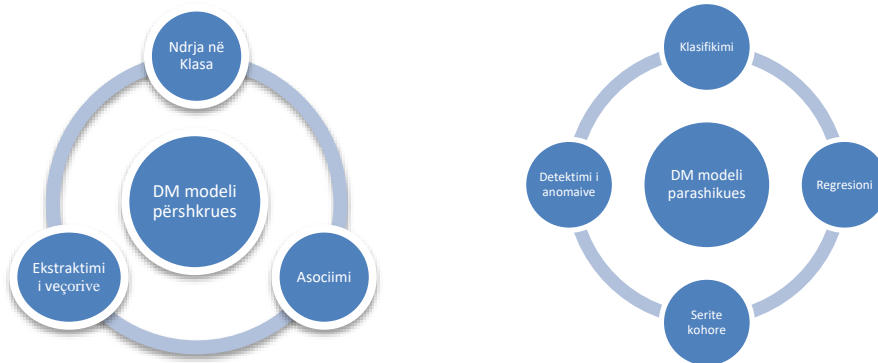


Figura 3-1 Modeli DM përshkrues (i pasupervizuar) dhe modeli DM parashikues (i supervizuar)

Modeli përshkrues analizon evente të ndodhura në të shkuarën me qëllim kuptimin e situatës ose pozicionit aktual, për të zgjedhur më pas mënyrën më të mirë për t'ju qasur të ardhmes. Kjo teknikë analizon sjelljen e të dhënave në të shkuarën për të identifikuar faktorët që mund të kenë çuar në sukses ose dështim. Pothuajse të gjithë departamentet që kanë një linjë raportimi direkte pranë menaxhimit, si shitje, marketing, operacione dhe finance, përdorin të gjithë këtë lloj analize pas ndodhjes së faktit.

Modelet përshkruese japin informacion sasior, ato kuantifikojnë relacionet mes të dhënave në mënyrë të atillë që të arrihet të bëhet një klasifikim i klienteve ose produkteve në grupe të caktuara. Ndryshe nga modelet parashikuese të cilat fokusohen në parashikimin e sjelljes së një klienti ose në renditjen e klientëve në varësi të probabilitetit që shfaq secili prej tyre në ndërmarrjen e një veprimi, modelet përshkruese gjejnë marrëdhënie të ndryshme mes klientëve dhe produkteve. Teknika të përdorura nga këto modele janë: Ndarja në klasa, Rregullat e asociimit dhe Ekstraktimi i veçorive.

Modelet parashikuese i kthejnë të dhënat në informacion të dobishëm. Analizat që bëhen përmes këtyre modeleve arrijnë të parashikojnë një ngjarje të mundshme në të ardhmen ose ecurinë e një situatë aktuale. Kjo teknikë njihet ndryshe si të mësuarit e supervizuar. Teknikat e modeleve parashikuese janë të përshtatshme për t'u përdorur kur lind nevoja e parashikimit të një vlere x të vendosur si objektiv. Në fushën e biznesit, këto modele eksplorojnë sjelljet në të dhënat historike dhe ato transaksionale mes faktorëve/variablave të ndryshëm, me qëllim identifikimin e risqeve dhe oportuniteteve të së ardhmes, duke ndikuar kështu në marrjen e vendimeve mbi transaksione ose evente opsional që mund t'i paraqiten menaxhimit.

Modelet parashikuese përfshijnë: Ndarjen në klasa, Identifikimin e anomalive, Regresin dhe Analizën e serive kohore.

## 3.2 Modeli përshkrues

Janë 3 teknikat më të përdorura në modelin përshkrues DM: Ndarja në klasa, Asocimi dhe Ekstraktimi i veçorive.

### 3.2.1 Ndarja në grupe

Ndarja në klasa është një teknikë e rëndësishme në data mining, ajo përfshin procesin e grupimit të të dhënave në një set grupesh në mënyrë që objektet e çdo grupi të kenë të njëjtat karakteristika me objektet e tjerë në grupin në të cilin bëjnë pjesë, por gjithashtu të jenë të dallueshëm nga objektet përbërës që i përkasin një grupi tjetër. Ngjashmëritë dhe veçoritë mes objekteve përcaktohen bazuar në vlerat e attributeve që përshkruajnë objektet. Ky proces grupimi realizohet pa patur asnjë të dhënë fillestare mbi emërtimin ose etiketimin e klasave/grupeve. Një grupim i mirë do ishte ai që do të reflektonte një shkallë të lartë ngjashmërie mes objekteve të një grupi si dhe një shkallë të lartë diversiteti mes dy grupeve. Superioriteti i algoritmit të përdorur për grupimin varet si nga matjet e ngjashmërisë të përdorura nga metoda ashtu dhe nga implementimi. Gjithashtu shkallë e lartë superioriteti tregon dhe aftësia për të zbuluar sa më shumë sjellje të fshehura nga të dhënat në analizë. Metodat e grupimit mund të krahasohen në mënyra të ndryshme duke marrë në shqyrtim kriteret e ndarjes, ndarjen e grupeve, matjen e ngjashmërive dhe hapësirën e grupit. Algoritmat e këtij tipi mund të kategorizohen në algoritme të bazuar në particionim, në ndarjen hierarkike, në algoritmet e bazuar në densitet si dhe ato të bazuar në shpërndarjen e të dhënave.

Tabela 3-1 Algoritme të Klasterizimit

Algoritme të bazuar në particionim	Algoritme të bazuar në ndarjen hierarkike	Algoritme të bazuar në densitet	Algoritme të bazuar në shpërndarjen e të dhënave
K-Means	Agglomerative (BIRCH, CHAMALEON)	DBSCAN	STING
K-Medoids (PAM,CLARA)	Divisive	DENCLUE	CLIQUE

Këto metoda ndryshojnë në (I) procedurat e përdorura për matjen e ngjashmërisë (brenda të njëjtit grup dhe mes grupeve), (II) në përdorimin e vlerave kufi në definimin e grupeve, (III) mënyrën e grupimit, duke i lejuar objektet të jenë në mënyrë strikte pjesë vetëm e një grupi apo mund të bëhen pjesë e grupeve të ndryshme në shkallë të ndryshme dhe në strukturën e algoritmit.

### 3.2.2 Asocimi

Një tjetër teknikë e rëndësishme në data mining është ai i rregullave të asocimit. Kjo teknikë kërkon për lidhjet që mund të ekzistojnë mes variablave.

Një shembull nga biznesi do ishte rasti i një dyqani që mbledh të dhëna mbi shitjet, me ndihmën e kësaj teknike mund të kuptohen cilët janë produktet që blihen në mënyrë frekvente me njëri tjetrin. Ky informacion i përfutur mund të përdoret më tej për arsye marketingu, kjo njihet shpesh si analiza e shportës së tregut.

Sjelljet që mund të zbulohen përmes kësaj teknike të data mining, mund të paraqiten në formën e rregullave të shoqërimit. Tipikisht këto rregulla quhen interesante nëse plotësojnë njëkohësisht një vlerë kufi minimum suporti dhe një minimum vlere kufi konfidence. Këto vlera kufi mund të përcaktohen paraprakisht nga përdoruesi ose nën një gjykim eksperti.

Përkufizim: Le të jetë  $I = \{I_1, I_2, \dots, I_m\}$  një set artikujsh. Le të jetë  $D$ , një set transaksionesh, objekt i studimit, ku çdo transaksion  $T$  është një set elementesh në mënyrë të tillë që:  $T \subseteq I$ . Çdo transaksion është i lidhur me një çelës identifikues të quajtur TID. Le të jetë  $A$ , një set elementesh. Një transaksion  $T$  mund të përmbajë një  $A$  vetëm nëse  $A \subseteq T$ . Një rregull shoqërimi do ishte një implikim i formës  $A \rightarrow B$ , ku  $A \subset I$ ,  $B \subset I$  dhe  $A \cap B = \emptyset$ . Rregulli  $A \rightarrow B$  bazohet në setin e transaksioneve  $D$  me suport  $s$ , ku  $s$  është përqindja e transaksioneve në  $D$  që përmbajnë  $A \cup B$ . Rregulli  $A \rightarrow B$  ka një konfidencë  $c$  në setin e transaksioneve  $D$  nëse  $c$  përcakton frekuencën e elementeve të  $B$ -së që shfaqen në transaksionet që përmbajnë  $A$ . Pra,  $\text{support}(A \rightarrow B) = \text{Prob}\{A \cup B\}$  dhe  $\text{konfidence}(A \rightarrow B) = \text{Prob}\{B/A\}$  [22].

Tabela 3-2 Shembull transaksionesh

TID	Artikuj
1	{Xhinse, T-Shirt, Këpucë, Çokollatë}
2	{Xhinse, Këpucë, Pallto, Syze}
3	{Orë, Çantë, Xhinse, T-Shirt}
4	{Rrip, Xhinse, Këmishë, Këpucë}

Supozojmë se  $A$  përfshin një listë artikujsh si: {Xhinse, T-Shirt} dhe  $B$  një listë artikujsh që përfshin {Këpucë}. Rregulli  $\{Xhinse, T-Shirt\} \rightarrow \{Këpucë\}$  ka një vlerë suport prej  $2/4=0.5$  dhe një vlerë konfidence prej  $2/3=0.67$ . Rregullat që plotësojnë njëkohësisht vlerën minimum të suportit dhe të konfidencës (minconf) quhen rregulla të forta. Disa prej algoritmeve më të njohur të teknikës së asociimit janë: AIS, SETM, Apriori, Aprioritid, Apriorihybrid, FP-Growth. Nga studimet më të fundit është vënë re që algoritmi FP-growth performon më mirë në aspektin e shpejtësisë dhe saktësisë se sa algoritma të tjerë më të vjetër si: AIS, SETM, Apriori, Aprioritid, Apriorihybrid [23].

### 3.2.3 Ekstraktimi i veçorive

Ekstraktimi i veçorive është teknika që krijon karakteristika të reja duke përdorur atributet aktuale të të dhënave. Këto karakteristika të reja paraqesin kombinime të sjelljeve të

rëndësishme që mund të identifikohen nga vlerat e attributeve. Modele të ndërtuara mbi bazën e këtyre karakteristikave të reja mund të jenë të një cilësie të lartë pasi mund ti përshkruajnë në mënyrë më konçize dhe të kuptueshme atributet.

Kjo teknikë në dallim nga selektimi i veçorive, që bën një renditje të attributeve ekzistuese në varësi të fuqisë parashikuese, realizon transformim të këtyre attributeve. Atributet e transformuara ose karakteristikat e reja janë një kombinim linear i attributeve origjinale. Paraqitja e të dhënave në këtë format të ri, përmes karakteristikave të reja, ndihmojnë në kompresimin e datasetit (duke zëvendësuar disa attribute me vlerën e një karakteristike të përfutur), lejon bërjen e parashikimeve (të dhënat me një karakteristike  $x$  kanë gjithashtu edhe vlerat e attributeve nga të cilat ka derivuar kjo karakteristikë) si dhe ndihmon në identifikimin e sjelljeve të fshehura të të dhënave.

Gjithashtu karakteristikat e reja të krijuara mund të shërbejnë si attribute të rinj, duke rritur kështu efikasitetin dhe saktësinë e teknikave të tipit të supervizuar si: klasifikimi, regresi, detektimi anomalive etj.).

Disa prej teknikave më të përdorura të ekstraktimit të veçorive janë: Principal Component Analysis (PCA) dhe Linear Discriminant Analysis (LDA).

PCA është metoda statistikore më e përdorur. Kjo metodë ekstraktton një hapësirë dimensionale më të ulët duke analizuar kovariancën strukturore të observimeve statistikore multivariate.

LDA kryesisht tenton të projektojë datasetet multidimensionale në të dhëna të një hapsire dimensionale më të vogël. Kjo teknikë synon të maksimizojë distancën mes klasave dhe të minimizojë atë midis elementeve të së njëjtës klasë [24].

### 3.3 Modeli parashikues

Janë 4 teknikat më të rëndësishme që përdoren nga modelet DM parashikuese. Konkretisht: Klasifikimi, Regresioni, Serite Kohore dhe Identifikimi i anomalive.

#### 3.3.1 Klasifikimi

Teknikat e klasifikimit në data mining janë në gjendje të procesojnë sasi të mëdha të dhënash. Ato mund të shërbejnë për përcaktimin e klasave kategorike mbi një training dataset dhe bëjnë gjithashtu klasifikimin e të dhënave të reja me të cilat mund të furnizohet seti i të dhënave. Procedura e klasifikimit njihet si metoda e marrjes së vendimeve në situata të reja. Krijimi i një procedure klasifikimi nga një set të dhënash për të cilat njihen paraprakisht klasat kryesore njihet si njohja e sjelljes. Kontekste në të cilat mund të përdoret kjo teknike janë psh. kategorizimi i klientëve individë në një institucion financiar sipas statusit, duke u bazuar në të dhënat e tyre financiare dhe ato personale. Ose diagnostikimi fillestar i një pacienti bazuar në simptomat madhore me qëllim fillimin e trajtimit urgjent gjatë kohës që priten rezultatet e tjera të testimeve. Në fushën e shkencës, në industri ose biznes, shfaqen shpesh situata kritike që kanë nevojë të klasifikohen dhe të merret një vendim në afat të shkurtër kohe. Klasifikimi bazohet në parashikimin e vlerave të mundshme që mund të marrë një variabël i nevojshëm për t'u parashikuar, bazuar në një input fillestar që i jepet modelit. Ky proces realizohet nga algoritmi i klasifikimit duke analizuar vlerat e attributeve të variablave input si dhe ato të variablit të target-uar që duhet të parashikohet. Algoritmi duhet të zbulojë lidhjet mes

atributeve kyç që mund të bëjnë më së miri parashikimin. Me të kuptuar këto relacione, në një dataset të ri ku variabli kyç që nevojitet të parashikohet mungon, me aplikimin e algoritmit në variablat e tjerë, mund të marrim një set të ri të dhënash me variablin target të plotësuar me vlera [25]. Shembull mund të jetë rasti i një set-i të dhënash nga një spital, i cili përmban rekordet e të gjithë klientëve të regjistruar ndonjëherë, duke patur informacion mbi sëmundjet dhe simptomat përkatëse. Në këtë rast sëmundja është variabli që kemi nevojë të parashikojmë ndërsa simptomat janë variablat inputues. Në momentin që inputojmë një set të ri të dhënash mbi pacientët e rinj të shtruar, algoritmi i klasifikimit duhet të jetë në gjendje të bëjë parashikimin rreth sëmundjes nga e cila mund të vuajnë këto pacientë.

Saktësia e parashikimit është elementi që përcakton se sa i mirë ose jo është algoritmi. Raporti i parashikimeve të sakta mbi totalin numrin e parashikimeve jep përqindjen se sa mirë janë bërë parashikimet. Një algoritëm i mirë parashikimi është ai që ka një përqindje saktësie më të lartë se sa vetë observimet finale të variablit parashikues. Në rastin e diagnozës së pacientëve, nëse rreth 80% e pacientëve që kanë vlera të rrahjeve të zemrës mbi 70 dhe janë të moshës mbi 65 vjeç ose kanë vlera të presionit të gjakut mbi 140/70 vuajnë nga sëmundjet kardiovaskulare, algoritmi i parashikimit duhet të ketë një % saktësie minimalisht 80%.

Metodat më të përdorura në ditët e sotme për këtë lloj algoritmi mund të grupohen si më poshtë [8] :

- Pema e marrjes së vendimeve (DT's)
- Support Vector Machine (SVM)
- Algoritmat Gjenetikë (GAs) / Evolutionary Programming (EP)
- Fuzzy Sets
- Rrjetat Neurale
- Rough Sets

### 3.3.2 Regresi

Regresi është një teknike data mining të të mësuarit automatik, që përpiket të përshtasë një ekuacion me një set të dhënash. Është më shumë i përdorur në rastet ku variabli target që duhet të parashikohet është i tipit numerik. Përmes regresit duhet të ndërtojmë një model duke u bazuar në lidhjet mes variablave të paravarur dhe variablit target që duhet të parashikohet, në një situatë ku vlerat e variablave të paravarur disponohen. Tipet kryesore të regresit janë:

- Regresi linear
- Regresi linear me shumë variabla
- Regresi jo linear
- Regresi jo linear me shumë variabla

Forma më e thjeshtë është regresi linear, i cili përdor formulën:  $(y = mx + b)$  dhe përcakton vlerat e duhura për  $m$  dhe  $b$ , në mënyrë që të zbulohen vlerat që mund të marrë  $y$ , bazuar në vlerat input të variablit  $x$ . Koeficientët  $m$  dhe  $b$  quhen koeficientët e regresit,  $m$  përcakton pjerrësinë e kurbës ndërsa  $b$  prerjen. Regresi linear me shumë variabla është një vazhdim i regresit të thjeshtë linear, i cili mundëson parashikimin e vlerave të një variabli target duke u bazuar jo në një por në disa variabla inputues [26]. Zakonisht marrëdhënia mes 2 variablave  $x$  dhe  $y$  nuk mund të përafrohet me një kurbë lineare të drejtë. Në raste si këto përdoret teknika e regresit jo linear e cila në thelb ka një ekuacion më të komplikuar për të përcaktuar funksionin që lidh  $y$  me  $x$ . Ndërsa regresi jo linear me shumë variabla i referohet regresit jo linear kur

kemi më shumë se sa një variabël të pavarur që ndikon në parashikim ( $x_1, x_2, \dots, x_n$ ). Në raste si këto lidhja jo lineare nuk mund të paraqitet në një hapësirë dy dimensionale. Fatkeqësisht, të dhëna të shumta nga jeta reale nuk janë të thjeshta të parashikohen, si shembull përmendim volumin e shitjeve, çmimet e stokut ose % e dështimit të produkteve, kjo për faktin se mund të varen nga lidhjet komplekse mes variablave të ndryshëm parashikues. Në rastet me kompleksitet të kësaj natyre përdoren teknika si: regresi logjistik, pema e marrjes së vendimeve ose rrjetat neurale. Pra të njëjtat teknika mund të përdoren edhe për modelet data mining të tipit regres dhe për ato që bazohen në klasifikim. Algoritmi CART (Classification and Regression Trees) mund të përdoret për të ndërtuar pema klasifikimi që kanë për qëllim klasifikimin e variablave target të tipit kategorik por njëkohësisht dhe për ndërtimin e regreseve që të bëjnë parashikimin e variablave target të tipit të vazhdueshëm. Edhe rrjetat neurale mund të përdoren për ndërtimin e modeleve të të dy llojeve, ato të klasifikimit dhe regresit.

### 3.3.3 Seritë kohore

Seri kohore quhet bashkësia e të dhënave të grumbulluara nga observime të bëra në mënyrë sequenciale në një afat të caktuar kohor. Në çdo moment në kohë mund të monitorohen matjet për një ose disa variabla që mund të jenë subjekte të studimit. Seria kohore rezultat quhet univariate ose multivariate në varësi të variablave të inkuorporuar. Qëllimet kryesore të përdorimit të metodave të serive kohore janë: indeksimi, ndarja në klasa, klasifikimi, gjetja e risive si dhe gjetja e rregullave. Më poshtë jepet një përshkrim i përgjithshëm për secilin prej tyre [27].

*Indeksimi:* Gjetja e serive kohore më të ngjashme në një bazë të dhënash.

*Ndarja në klasa:* Gjetja e grupeve të serive kohore në një dataset në mënyrë të tillë që seritë kohore të një grupi të jenë të ngjashme në karakteristika me njëra tjetrën dhe të ndryshme nga ato të një grupi tjetër.

*Klasifikimi:* Etiketimi i një grupi serish kohore në mënyrë të tillë që të jetë më i ngjashëm me seritë kohore të të njëjtit grup se sa me ato të serive kohore në grupet e tjera.

*Gjetja e risive:* Proçesi i gjetjes së seksioneve brënda një serie kohore që shfaqin sjellje të ndryshme ose të reja nga ato të përgjithshmet që mund të evidentohen nga një model bazë.

*Zbulimi motivit:* Zbulimi i sjelljeve të përsëritura dhe të panjohura më parë në një bazë të dhënash të serive kohore.

*Zbulimi i rregullave:* Gjetja e rregullave që mund të përshkruajnë sjellje të mundshme në një moment specifik në kohë, nga një ose më shumë seri kohore.

Kjo metodë DM është mjaft komplekse dhe problematikat kryesore vijnë nga dimensionet e mëdha të serive kohore si dhe nga vështirësia për të definuar një formë matjeje të ngjashmërisë.

### 3.3.4 Identifikimi i anomalive të të dhënave

Vlera e një atributi të caktuar në një bazë të dhënash do të quhet një outlier nëse ajo ka një devijim të konsiderueshëm nga sjella e rregullt ose vlerat më të zakonshme që gjenden në dataset për këtë atribut. Proçesi i identifikimit të këtyre objekteve në fazat e para të analizës së

të dhënave është shumë i rëndësishëm, në mënyrë që në vazhdim të vendoset edhe mënyra se si këto objekte do të merren në konsideratë ose jo gjatë modelimeve.

Proçesi i identifikimit të anomalive është proçesi i gjetjes së vlerave outlier në një set observimesh. Ky proçes sa vjen dhe kthehet në hallkë të një rëndësie të lartë në tërë zinxhirin e modelimit të të dhënave, duke qenë se sasia e të dhënave është në rritje. Identifikimi i këtyre anomalive mund të jetë tregues i një aksesit të paautorizuar në një sistem, një vjedhje në një kartë krediti ose diagnostikimin e një sëmundjeje. Anomalitë ose parregullsitë në të dhëna mund të jenë të tipit kontekstuale ose kolektive. Anomalitë e tipit kontekstual janë ato ku rekorde të veçantë të set-it të të dhënave devijojnë nga pjesa tjetër e tyre. Ky është edhe modeli më i thjeshtë dhe që përdoret nga shumica e algoritmeve ekzistues. Anomalitë kontekstuale janë ato ku rekordi ka sjellje dhe attribute kontekstual. Të njëjtët attribute mund të konsiderohen normal për një kontekst dhe jo normal për një kontekst tjetër. Ndërsa anomalitë kolektive janë ato ku një grup rekordesh të ngjashëm mes tyre devijojnë nga pjesa tjetër e datasetit. Kjo mund të ndodhë vetëm në set të dhënash ku rekordet janë të lidhur mes tyre. Ndërkaq anomalitë kontekstuale mund të konvertohen në vlera fikse duke i agreguar ato sipas kontekstit.

Kategoritë kryesore të realizimit të procesit të identifikimit të anomalive në bazat e të dhënave grupohen sipas natyrës, në ato të bazuara në klasifikim, ato të bazuar në vlerat e observimeve më të afërta (nearest-neighbor), ato që bazohen në ndarjen në klasa si dhe kategoritë e bazuara në analiza statistikore. Algoritmet e bazuar në klasifikim quhen ndryshe algoritme të mbikqyrur dhe pretendojnë se dallimi mes vlerave jo normale dhe normale mund të bëhet për një pjesë të caktuar të atributit. Algoritmet nearest-neighbor pretendojnë që anomalitë janë të shpërndara në dataset dhe larg vlerave më të afërta të një observimi. Këto algoritme quhen ndryshe algoritme të pa survejuar. Ato të bazuar në ndarjen e observimeve në klasa funksionojnë duke grupuar rekorde të ngjashme në klasa dhe duke pretenduar që anomalitë nuk do jenë pjesë e së njëjtës klasë ose do ndodhen larg qendrave të klasave, ose që u përkasin klasave më të vogla dhe më të shpërndara. Ndërsa algoritmet e bazuar në analizat statistikore i detektojnë anomalitë duke i parë ato të devijuara nga modeli klasik.

Metodat e përmendura më lart të detektimit të anomalive janë mjaft të përdorura për gjetjen e transaksioneve të dyshimta që mund të lidhen me mashtrime në institucione financiare.

### **3.4 Teknikat data mining që përdoren në sistemet e inteligjencës së biznesit BI**

BI (Business Intelligence) i referohet një set-i mjetesh ose programesh që përdoren për analiza multidimensionale të bazës së të dhënave, me qëllim lehtësimin e marrjes së vendimeve në biznes. Një prej komponenteve më të rëndësishëm në BI është OLAP. Baza e OLAP është data kubi, që përbën një model multidimensional të bazës së të dhënave, ky i fundit i kombinuar me teknika të ndryshme ka shfaqur një performancë shumë të lartë në shpejtësinë e analizës dhe përpunimit të sasive të mëdha të të dhënave.

Implementime të ndryshme të OLAP janë ROLAP, MOLAP dhe HOLAP të cilët kanë karakteristika dalluese nga njëri tjetri në lidhje me performancën dhe saktësinë. Në krahasime dhe simulime të ndryshme të teknologjisë ROLAP që bazohet në ekzekutimin e disa kërkesave në një depositor të madh të dhënash me HOLAP që përdor kubat inteligjent, kjo e fundit spikat në performancën totale të sistemit, koha e përgjigjes reduktohet në mënyrë të ndjeshme. Gjithashtu zënë një hapësirë më të vogël të memories për sa kohë kubat ruhen në server. Benefit tjetër i kësaj teknologjie është gjithashtu dhe rritja e efijencës dhe performancës në raportistikë.

Konkludojmë duke nënvizuar faktin që OLAP është një mjet i fuqishëm në ekstraktimin e të dhënave nga sistemet BI duke performuar mirë në kohë dhe në performancë në ekzekutimin e kërkesave në datase të shumë të mëdha [18]. Kjo teknologji e pajis përdoruesin me një informacion multidimensional duke e bërë raportimin një proces më fleksibël dhe më eficient, pavarësisht se për organizata më të vogla me sasi të kufizuara të dhënash, kapacitetet e kësaj teknologjie mund të jenë të tepërta, pra aplikimi i tyre është i përshtatshëm dhe më i frytshëm në sasi shumë të mëdha të dhënash. Aplikimet OLAP janë mjaft të përdorura nga teknikat e Data Mining. Në bazat e të dhënave OLAP janë ruajtura dhe agreguara të dhëna historike, në formën e skemave multi-dimensionale (kryesisht të tipit yll). Skemat yll në datase të mëdha rrisin performancën si rezultat i numrit të vogël të lidhjeve që duhet të bëjnë për të marrë një raport. Punime të ndryshme të mëtejshme në këtë fushë provuan se ishte e një interesi dhe përfitueshmërie të veçantë kombinimi i këtyre teknologjive bashkë OLAP dhe data mining duke krijuar kështu një degëzim të ri të quajtur Online Analytical Mining (OLAM). Termi OLAM u prezantua si fillim nga Han dhe u analizua më tej nga HAN, Chee and Chinag. Qëllimi i integritimit të OLAP me data mining ishte cilësia e mirë e bazës së të dhënave në depozitorët e mëdhenj të të dhënave, infrastruktura në përpunim që ofron një data ëarehouse, analizat në eksplorim që ofron OLAP si dhe selektimi on-line i funksioneve të ndryshme data mining. Një arkitekturë që integron OLAP dhe OLAM është propozuar në këtë studim.

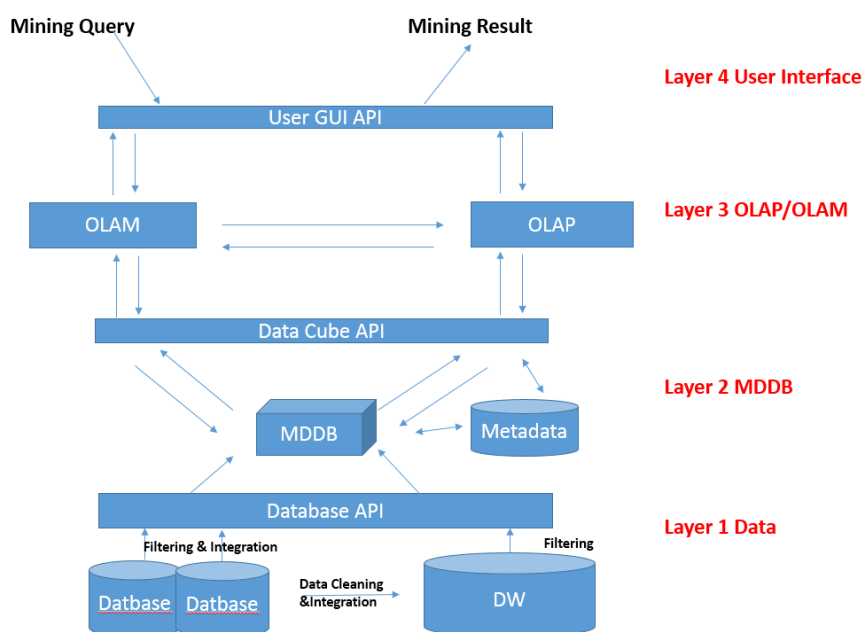


Figura 3-2 Arkitektura OLAM

Hua [29] propozoi dhe zhvilloi një integrim interesant mes teknologjisë OLAP dhe teknikës së asociimit në data mining që çoi në krijimin e një rregulli asociimi në mining shumë fleksibël dhe multidimensional. Dzeroski [30] kombinoi OLAP dhe data mining në një mënyrë tjetër me qëllim zbulimin e sjelljeve të ndryshme në një databazë pacientësh. Ai përdori teknikën e ndarjes në klasa si dhe atë të pemës së vendimeve.



### 3.5 Hapat e DATA-MINING

#### 3.5.1 Cilësia e të dhënave

Të dhënat kanë kualitet nëse përmbushin kërkesat e paravendosura për përdorimin e tyre. Ka shumë faktorë që ndikojnë në cilësinë e të dhënave, përfshirë saktësinë, plotësinë, konsistencën, koherencën, besueshmërinë dhe interpretueshmëria. Marrim rastin se kemi një menaxher dege në Coin dhe është ngarkuar me analizimin e të dhënave të kompanisë, duke u nisur nga shitjet e degës që kanë në ngarkim. Që në momentin e parë do të analizonte bazën e të dhënave dhe magazinën e të dhënave dhe do të zgjidhte atributet apo dimensionet (psh. artikull, cmim dhe njësi të shitura) që do të përfshihen në analizën e tij. Do të vërë re se një pjesë e attributeve për të dhëna të ndryshme, nuk kanë vlerë të regjistruar. Për analizën e tij, do të donte të përfshinte informacionin nëse të gjithë artikujt e shitur janë marketuar dhe etiketuar si “në shitje” por ky informacion nuk është regjistruar. Për më tepër, përdoruesit e bazës së të dhënave kanë raportuar gabime, vlera te pazakonta dhe inkonsistenca në të dhënat e regjistruara për disa transaksione. Në fjalë të tjera, të dhënat që do të analizojë me teknikat e data mining janë të paplota (mungojnë vlera te attributeve ose mungojnë attribute që i interesojnë, ose përmbajnë vetëm vlera të agreguara); të pasakta ose me zhurma (përmbajnë gabime ose vlera që devijojnë nga pritshmëria); dhe të paqëndrueshme (përmbajnë mos përputhje në kodet e departamenteve që përdoren për kategorizimin e artikujve).

Ky skenar ilustron tre elementë që përcaktojnë cilësinë e të dhënave: saktësia, plotësia dhe konsistenca. Të dhënat e pasakta, të paplota dhe të paqëndrueshme janë vecori të zakonshme të bazave të mëdha të të dhënave apo magazinave të mëdha të të dhënave. Ka shumë arsye të mundshme për pasaktësinë e të dhënave (psh. të pasurit e attributeve të gabuara). Instrumentat që përdoren për mbledhjen e të dhënave mund të jenë të gabuar. Mund të kenë ndodhur gabime njerëzore apo procesimi në hedhjen e të dhënave. Përdoruesit mund të vendosin qëllimisht të dhëna të gabuara për fusha të detyrueshme, kur nuk duan të përfshijnë informacion personal (psh duke vendosur një datë cfarëdo në vend të ditës së lindjes apo një adresë te pasaktë). Ky fenomen njihet si fshehje e qëllimshme e informacionit. Mund të ndodhin edhe gabime në transmetimin e të dhënave. Mund të ketë limitime të teknologjisë si hapë sirë e limituar e bufferit, që shërben për koordinimin dhe sinkronizimin e transmetimit të të dhënave. Gjithashtu, mund të kemi të dhëna të pasakta nga përdorimi i konvencioneve të ndryshme në emertim apo kodim për të njëjtat të dhëna ose nga përdorimi i formateve të gabuara për fushat (psh numër aty ku pritet datë). Një tjetër fenomen mund të jetë duplikimi i të dhënave, i cili ka nevojë për pastrim.

Të dhënat e paplota mund të ndodhin për një sërë arsyesh. Atributet e interesit mund të mos jenë disponibël në çdo kohë, si rasti i kodit të klientit në transaksionet e shitjes. Të dhënat e tjera nuk përshihen pasi nuk janë konsideruar të rëndësishme në momentin e hedhjes. Të dhëna të tjera të rëndësishme mund të mungojnë për shkak të keqkuptimeve apo keqfunksionimeve të pajisjeve. Përdoruesit mund të kenë fshirë të dhëna që kanë qënë jo konsistente me të dhënat e tjera të regjistruara. Për më tepër, regjistrimi i logeve për ndryshimet e të dhënave mund të mos jetë aktivizuar. Mungesa e të dhënave, veçanërisht për të dhënat që kanë mungesa për disa attribute mund të ketë nevojë të përcaktohet.

Duhet marrë parasysh se cilësia e të dhënave varet nga qëllimi i përdorimit të tyre. Dy përdorues të ndryshëm mund të kenë dy vlerësime të ndryshme për cilësinë e një baze të dhënash. Për shëmbull, një analist marketingu mund të dojë të aksesojë bazën e të dhënave që analizuam më sipër për të marrë adresat e klientëve. Disa nga adresat mund të jenë të pasakta

apo të vjetra por në total, rreth 80% e tyre janë të sakta. Analisti i marketingut është i kënaqur me cilësinë e të dhënave të cilat do ti përdorë për marketing të targetuar, pavarësisht se menaxheri i shitjeve që morrëm në shqyrtim më sipër i konsideron të dhënat e pasakta.

Koherenca afekton, gjithashtu cilësinë e të dhënave. Supozojmë se menaxheri i shitjeve po kontrollon shpërndarjen e bonuseve mbi shitjet për asistentët më të suksesshëm në shitjen e Coin. Një problem mund të jetë se disa asistentë në shitje nuk i deklarojnë shitjet e tyre në kohë në fund të muajit. Gjithashtu, mund të aplikohen saktësime dhe sisteme pas përfundimit të muajit. Për një periudhë kohe pas çdo fund-muaji, të dhënat e ruajtura në bazën e të dhënave janë te paplota. Megjithatë, kur të gjithë të dhënat hidhen në bazën e të dhënave, ato bëhen korrekte. Fakti që të dhënat e fund-muajit nuk hidhen në kohe reale, ka një impakt negativ në cilësinë e të dhënave.

Dy faktorë të tjerë që ndikojnë në cilësinë e të dhënave janë besueshmëria dhe interpretueshmëria. Besueshmëria reflekton se sa besohen të dhënat nga përdoruesit, ndërsa interpretueshmëria reflekton se sa e vështirë është të kuptohen të dhënat. Supozojmë se një bazë të dhënash në një moment të caktuar, ka disa gabime por që janë korrigjuar më vonë. Megjithatë, të dhënat e gabuara u kanë shkaktuar shumë probleme përdoruesve të departamentit të shitjeve dhe si rrjedhim ata nuk i besojnë më të dhënat. Të dhënat gjithashtu përdorin shumë kode kontabiliteti, të cilat departamenti i shitjeve nuk di si ti interpretojë. Edhe pse baza e të dhënave, tashmë është e saktë, konsistente dhe koherente, përdoruesit e departamentit të shitjeve mund ta cilësojnë atë me cilësi të ulët për shkak të besueshmërisë së ulët dhe interpretueshmërisë së dobët.

### 3.5.2 Hapat kryesore për paraproçesimin e të dhënave

Në këtë paragraf do të shohim hapat kryesore të procesit të paraproçesimit të të dhënave, përkatësisht: pastrimi i të dhënave, integrimi i të dhënave, reduktimi i të dhënave dhe transformimi i të dhënave [31].

Pastrimi i të dhënave konsiston në punën që kryhet për pastrimin e të dhënave duke vendosur të dhënat që mungojnë, sistemimit të të dhënave të paqarta, identifikimi dhe heqja e anomalive dhe zgjidhja e paqëndrueshmërisë së të dhënave. Nëse përdoruesit besojnë se të dhënat janë të papastra, ata nuk do ti besojnë rezultatet e proceseve të data mining që janë aplikuar në to. Për më tepër, të dhënat e papastra mund të shkaktojnë konfuzion për procedurën e data mining, duke rezultuar në të dhëna në dalje të cilat janë jo të besueshme. Pavarësisht se shumica e proceseve DM kanë procedurë për të trajtuar të dhënat e paplota apo të paqarta, ato nuk janë gjithmonë robuste. Përkundrazi këto procedurë mund të përqendrohen në mënjanimin e vendosjes së të dhënave në funksionin që po modelohet. Si rrjedhim, një hap preprocesimi i nevojshëm është kalimi i të dhënave në disa procedurë pastrimi.

Duke ju rikthyer shëmbullit të Coin më sipër, në të cilin përfshihen të dhëna nga burime të ndryshme në analizën tonë, do të na duhet të integrojmë shumë baza të dhënash, kuba të dhënash ose skedarë. Megjithatë, disa attribute që përfaqësojnë një koncept të caktuar mund të kenë emra të ndryshëm në baza të ndryshme të dhënash duke na shkaktuar paqëndrueshmëri apo teprica. Për shëmbull, një atribut për identifikimin e klientit mund të referohet si “*client*” në bazën e të dhënave të sistemit CRM dhe si “*codclient*” në bazën e të dhënave të sistemit POS. Mund të kemi paqëndrueshmëri në emërtim dhe në vlerat e attributeve. Përshembull, i njëjti emër mund të regjistrohët si Genti në një bazë të dhënash, Gent në një tjetër dhe G në një të tretë. Gjithashtu disa të dhëna mund të vijjnë si rrjedhojë e të dhënave të tjera (psh stoku

mujor). Të pasurit e një sasive të madhe të dhënash të tepërta mund të ngadalesojë ose konfuzojë procesin e zbulimit të njohurive (KDD). Qartësisht, përveç pastrimit të të dhënave duhet të merren masa për parandalimin e tepricave gjatë integritit të të dhënave. Zakonisht pastrimi dhe integrimi i të dhënave janë një hap preprocesimi gjatë përgatitjes të të dhënave për tu kaluar në magazinën e të dhënave. Një proces suplementar i pastrimit të të dhënave mund të kryhet për të hequr tepricat që rezultojnë nga integrimi i të dhënave.

Një tjetër faktor që duhet marrë parasysh është sasia e të dhënave. Nëse sasia e të dhënave është shumë e madhe atëherë procesi i data mining mund të ngadalësohet. Duhet të gjejmë një mënyrë për të reduktuar sasinë e të dhënave pa rrezikuar rezultatin e teknikave DM.

Reduktimi i të dhënave na jep një prezantim më të reduktuar të setit të të dhënave që është më i vogël në volum, por përsëri prodhon të njëjtin (pothuajse të njëjtin) rezultat analitik. Teknikat e reduktimit të të dhënave përfshijnë reduktimin e dimensioneve dhe reduktimin e numerik.

Në teknikën e reduktimit të dimensioneve, aplikohen skema enkodimi të të dhënave për të marrë një version të reduktuar apo të kompresuar të të dhënave fillestare. Një shëmbull do të ishte transformimet wavelet dhe analiza e komponenteve principale, zgjedhja e nënseteve të attributeve (psh heqja e attributeve të panevojshëm), dhe ndërtimi i attributeve ku një set më i vogël dhe më i dobishëm i attributeve derivohet nga seti original [31].

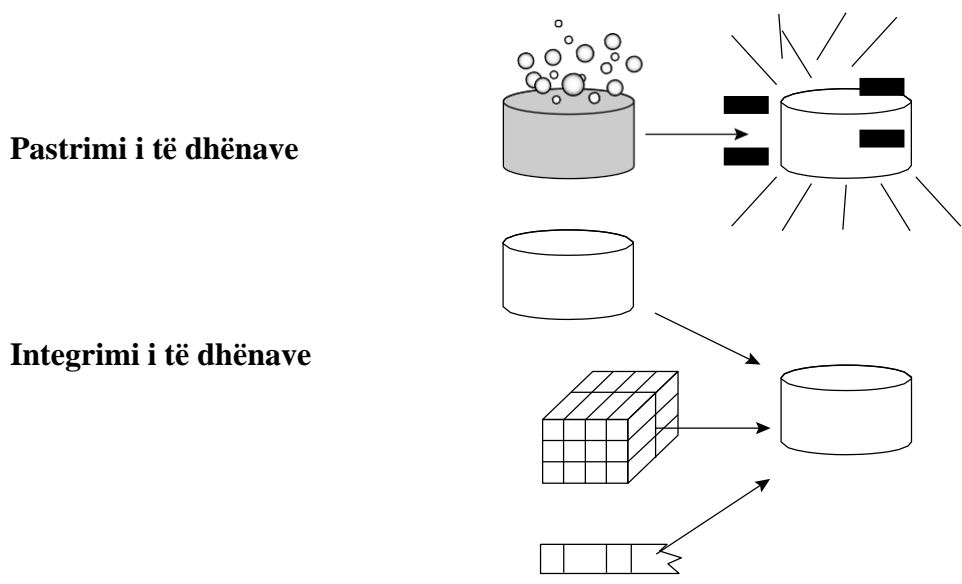
Në reduktimin numerik, të dhënat zëvendësohen nga prezantime më të vogla duke përdorur modelet parametrike (psh modeli i regresit dhe modeli log linear) ose modele jo parametrike (psh. histogramat, klasterat, kampionimi apo agregimi i të dhënave). Nëse marrim në konsideratë shëmbullin më sipër dhe duam të përdorim një algoritëm DM të bazuar te distanca siç janë: rrjetat neurale, klasifikuesit sipas fqinjët më të afërt apo Klastering. Këto algoritma na japin rezultate më të mira nëse të dhënat që do të analizohen normalizohen, që do të thotë përshkallëzohen në diapazone më të vegjël si [0. 0. 1. 0]. Marrim në konsideratë një klient për të cilin ruajmë 2 të dhëna: moshën dhe vlerën totale të blerjeve vjetore. Në përgjithësi vlera e ruajtur e blerjeve vjetore do të jetë më e madhe se vlera e ruajtur në fushën e moshës. Rrjedhimisht, nëse atributet nuk normalizohen, matjet e distancës që do të kryhen në fushën e blerjeve vjetore, përgjithësisht, do të kenë peshë më të madhe se matjet e distancës në fushën e moshës. Në rastet kur vlerat e të dhënave për atributet zëvendësohen nga diapazone ose nivele konceptuale, është e nevojshme teknika e diskretizimit ose gjenerimi i hierarkisë konceptuale. Për shembull, vlerat e vendosura në fushën e moshës mund të zëvendësohen me koncepte të nivelit më të lartë si "adoleshent", "i ri", "i moshuar".

Konceptet e diskretizimit dhe të gjenerimit të hierarkisë janë mjete të fuqishme për data mining pasi lejojnë Data Mining në nivele të shumta abstragimi. Normalizimi, diskretizimi i të dhënave dhe gjenerimi i hierarkisë së koncepteve janë forma të ndryshme të transformimit të të dhënave. Duhet të kuptojmë se këto operacione të transformimit të të dhënave bëjnë pjesë të procedurat e paraprocesimit të të dhënave të cilat kontribuojnë në suksesin e proceseve DM.

Figura 3.5 paraqet një përmbledhje të hapave të paraprocesimit të të dhënave të shpjeguara më sipër. Duhet të theksojmë se kategorizimi i mësipërm nuk është ekskluziv. Përshembull, heqja e tepricave të të dhënave, mund të shihet si një formë e pastrimit të të dhënave por edhe si një formë e reduktimit të të dhënave [31].

Si përfundim, të dhënat reale, përgjithësisht janë të papastra, të paplota dhe të paqëndrueshme. Teknikat e paraprocesimit të të dhënave mund të rrisin cilësinë e të dhënave dhe në këtë mënyrë ndihmojnë në rritjen e saktësisë dhe efikasitetit të proceseve DM në vazhdim. Paraprocesimi i

të dhënave është një hap i rëndësishëm në procesin KDD, pasi vendimet cilësore duhet të bazohen në të dhëna cilësore. Detektimi i anomalive të të dhënave, sistemimi i tyre shpejt dhe reduktimi i të dhënave që duhet të analizohen mund të sjellin shumë avantazhe në procesin e vendimmarrjes.



**Tranformimi i të dhënave**

-2, 32, 100, 59, 48      →      -0.02, 0.32, 1.00, 0.59, 0.48

**Reduktimi i të dhënave**

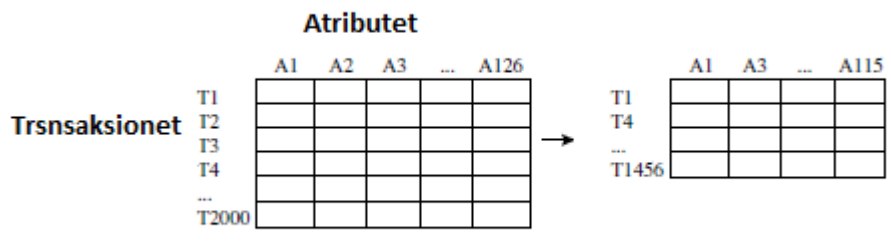


Figura 3-3 Hapat e paraproçesimit të të dhënave

### 3.6 Konkluzione

Ne këtë kapitull u krye një studim i teknikave të Data Mining (DM). Fillimisht u krye një analizë e modeleve kryesore të DM, përkatesisht: modelit përshkrues dhe modelit parashikues. Algoritmat kryesore të secilit model u listuan dhe analizuan. Gjithashtu u shpjegua sesi kryhet integrimi i DM me sistemet e inteligjencës në biznes dhe statusi aktual i këtyre integrimeve. Së fundmi hapat kryesorë që duhet të ndërmerren gjatë implementimit të DM me fokus në paraproçesimin e të dhënave u shpjeguan dhe analizuan. Rëndësia kritike e këtij hapi u ilustrua dhe shpjegua.

## KAPITULLI 4

### 4. TEKNIKAT DM TË MARRA NË SHQYRTIM

#### 4.1 Metoda e Ndarjes/Particionimit në Klastera

Versioni më i thjeshtë dhe më themelor i analizës së klaster-ave është ndarja, e cila organizon objektet e një grupi në disa grupe specifike. Për të mbajtur sa më të saktë specifikimin e problemit, mund të supozojmë se numri i klastera-ve është dhënë si një element i njohur. Ky parametër është pikënisja fillestare për metodat e ndarjes. Duke pasur parasysh një grup të dhënash  $D$ , të  $n$  objekteve dhe  $k$  numrin e klasterave për t'u formuar, algoritmi i ndarjes organizon objektet në  $k$  particione ( $k \leq n$ ), ku secila ndarje paraqet një klaster.

Grupimet formohen për të optimizuar një ndarje me një kriter sa më objektiv, të tillë si një funksion i pangjashmërisë bazuar në distancë, kështu që objektet brenda një grupimi mund të jenë "të ngjashme" me njëra-tjetrën dhe "jo të ngjashme" me objektet në grupe të tjera për sa i përket termave të attributeve të bashkësisë së të dhënave.

Në këtë seksion do të trajtohet si metoda  $k$ -means (në seksionin 4.1.1) dhe  $k$ -medoids (në seksionin 4.1.2). Ne gjithashtu do të shpjegojmë disa variacione të këtyre ndarjeve dhe se si ato mund të shkallëzohen për të trajtuar grupe të mëdha të të dhënave.

##### 4.1.1 Algoritmi K-Means

Supozojmë se kemi një grup të të dhënave të cilën e shënojmë me  $D$ , i cili përmban  $n$  objekte në hapësirën euklidiane. Metodat e ndarjes shpërndajnë objektet në  $D$  në  $k$  klastera, nga  $C_1$  deri në  $C_k$ , për çdo  $C_i$  që përfshihet në  $D$  dhe prerja e  $C_j$  me  $D$  jep një bashkësi boshe. Një funksion objektiv përdoret për të vlerësuar cilësinë e ndarjes në mënyrë që objektet brenda një klasteri janë të ngjashëm me njëri-tjetrin por jo të ngjashëm me objektet në klastera të tjera. Ky funksion objektiv synon ngjashmëri të lartë në tipologjitë intra-klaster dhe ngjashmëri më të ulët në tipologjitë inter-klaster [32].

Një teknikë ndarëse e bazuar në centroid përdor centroidin e një grupi, për të përfaqësuar atë klaster. Për ta kuptuar, centroidi i një klasteri është pika e tij qendrore. Centroidi mundet të përcaktohet në mënyra të ndryshme si p.sh. nga mesatarja ose medoidi i objekteve (ose pointerave) të caktuar në grup. Distanca midis një objekti  $p \in C_i$  dhe  $c_i$ , përfaqësuesit të klaster-it, matet me  $\text{dist}(p, c_i)$ , ku  $\text{dist}(x, y)$  është distanca euklidiane midis dy pikave  $x$  dhe  $y$ . Cilësia e klasterit  $C_i$  mund të matet me variacionin, e cila është shuma e gabimit në katror midis të gjitha objekteve në  $C_i$  dhe centroidit  $c_i$ , e përcaktuar si:

$$E = \sum_{i=1}^k \sum_{p \in C_i} \text{dist}(p, c_i)^2 \quad (4-1)$$

ku  $E$  është shuma e gabimit në katror për të gjitha objektet në grupin e të dhënave;  $p$  është pika në hapësirë që përfaqëson një objekt të caktuar; dhe  $c_i$  është centroidi i klasterit  $C_i$  ( $p$  dhe  $c_i$  janë shumëdimensionale). Domethënë, për secilin objekt në secilin klaster, distanca nga objekti me qendrën e grupit (klasterit) ngrihen në katror dhe distancat mblidhen.

Ky funksion objektiv përpiqet të bëjë që k-klasterat që rezultojnë, të jenë sa më kompakte dhe sa më të ndara. Optimizimi i variacionit brenda klasterit është sfidues nga pikëpamja kompjuterike. Në rastin më të keq, do të na duhet të numërojmë një numër ndarjesh të mundshme që janë eksponenciale në lidhje me numrin e grupeve.

Nëse numri i klasterave  $k$  dhe dimensionin e hapësirës arrijnë të fiksohen, problemi mund të zgjidhet në kohën  $O(n^{dk+1} + \log n)$ , ku  $n$  është numri i objekteve. Për të kapërcyer koston frenuese të llogaritjes për zgjidhjen e saktë, qasje të ndryshme shpesh përdoren në praktikë. Një shembull kryesor është algoritmi k-means, i cili është i thjeshtë dhe është i përdorshëm në ditët e sotme.

"Si funksionon algoritmi k-means?" Algoritmi k-means përcakton centroidin e një klasteri si vlerë mesatare të pikave brenda klasterit. Së pari, zgjedh rastësisht  $k$  nga objektet në  $D$ , secili prej të cilëve fillimisht përfaqëson një mesatare të klasterit. Secili nga objektet e mbetura, i caktohet klaster-it në të cilin është më afër, bazuar në distancën euklidiane midis objektit dhe qendrës së klasterit. Ky algoritëm përmirëson në mënyrë iterative ndryshimin brenda klasterit. Për secilin klaster, ai llogarit mesataren e re duke përdorur objektet e caktuara në klasterin në iteracionet e mëparshme. Përsëritjet vazhdojnë derisa caktimi të jetë i qëndrueshëm, domethënë klasterat e formuara në raundin aktual janë të njëjta me ato të formuara në raundin e mëparshëm. Metoda k-means është përmbledhur në figurën 4.1 [32].

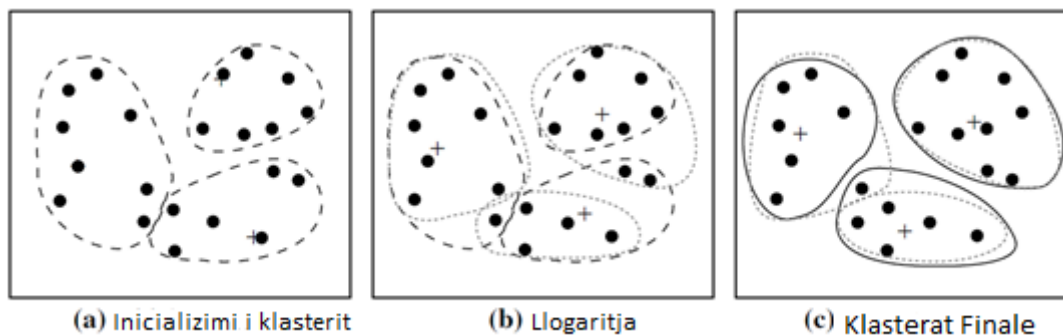


Figura 4-1 Klasterizimi i një seti objektësh që përdorin algoritmin k-means; (b) azhurnimi e qendrave të klasterave dhe rivendosja e objekteve në përputhje me gjendjet (mesatarja e secilit klaster shënohet me +).

**Algoritmi:  $k$ -means:** Algoritmi  $k$ -means, është algoritmi i ndarjes ku çdo qendër e klasterave përfaqësohet nga vlera mesatare e objekteve në klaster.

**Input:**

- $k$ : numri i klasterave,
- $D$ : seti i të dhënave i cili përmban  $n$  objekte.

**Output:** Një set me  $k$ -klastera.

**Metoda:**

- (1) në mënyrë arbitrare zgjidhen  $k$  objekte nga  $D$  si qendrat fillestare të klasterave;
- (2) përsëritet;
- (3) (ri)caktimi i secilit objekt në klasterin me të cilin objekti është më i ngjashëm, bazuar në vlerën mesatare të objekteve në grup;
- (4) update-im i klaster-means, përlogaritja e vlerës të objektit për secilin klaster
- (5) derisa të mos ndryshojë;

Ilustrojmë me shembullim në vijim. Proçesi i klasterizimit me anë të algoritmit  $k$ -means. Konsiderojmë një grup objektesh të vendosura në hapësirë 2-D, siç përshkruhet në Figurën 4.1 (a). Le të jetë  $k=3$ , domethënë, përdoruesi do të dëshironte që objektet të ishin të ndara në tre grupe. Bazuar në algoritmin e sipërmendur, zgjedhim në mënyrë arbitrare tre objekte si tre qendrat fillestare të klasterave, ku qendrat e grupeve shënohen nga një C. Secili objekt i është caktuar një grupi bazuar në qendrën e grupeve në të cilën është më e afërta.

Qendrat e grupeve azhurnohen. Kjo është, vlera mesatare e secilës klaster ku është rillogaritur dhe bazuar në objektet aktuale në set. Duke përdorur qendrat e reja të klasterave, objektet rishpërndahen në grupe bazuar te qendra më e afërt e grupeve. Një rishpërndarje e tillë formon silueta të reja të rrethuara nga kthesa të ndërprera, siç tregohet në Figura 4.1(b). Ky proçes përsëritet, duke çuar në figurën 4.1(c). Caktimi i proçeseve iterative për objektet e grupeve është për të përmirësuar ndarjen. Proçesi vazhdon derisa nuk ndodh asnjë rivendosje e objekteve në ndonjë klaster.

Rezultatet e algoritmit  $k$ -means mund të varen nga zgjedhja fillestare e rastësishme të qendrave të klasterave. Për të marrë rezultate të mira në praktikë, është e rekomandueshme të ekzekutohet shumë herë me qendra të ndryshme fillestare të klasterave [32].

Kompleksiteti kohor i algoritmit  $k$ -means është  $O(nkt)$ , ku  $n$  është numri i përgjithshëm i objekteve,  $k$  është numri i grupeve, dhe  $t$  është numri i përsëritjeve. Normalisht,  $k \ll n$  dhe  $t \ll n$ . Prandaj, metoda është relativisht e shkallëzuar dhe efikase në përpunimin e të dhënave të mëdha (big data). Ekzistojnë shumë variante të algoritmit  $k$ -means. Këto mund të ndryshojnë në përzgjedhjen dhe inicializimin e  $k$  dhe strategjitë për llogaritjen e klasterit.



Algoritmi k-means mund të zbatohet vetëm kur përcaktohet mesatarja e një grupi objektsh. Ky mund të mos jetë rasti në disa aplikime si p.sh. kur të dhëna me attribute nominale janë përfshirë. Algoritmi k-modes është një variant i k-means. Ky algoritëm përdor masa të reja të ngjashmërisë për t'u marrë me objektet nominale dhe një metodë e bazuar në frekuencë për të azhurnuar mënyrat e grupeve. Algoritmat k-means dhe k-modes mund të integrohen për të grumbulluar të dhëna me vlera të përziera numerike dhe nominale. Nevoja që përdoruesit të specifikojnë k, numrin e grupeve, paraprakisht mund të shihet si një disavantazh. Ka pasur studime se si të kapërcehet kjo vështirësi, të tilla si duke siguruar një gamë të përafërt të vlerave k dhe pastaj duke përdorur një teknikë analitike për të përcaktuar k-në më të mire, duke krahasuar rezultatet e klasterizimit të marra për vlerat k-të e ndryshme. Në studimin tonë do të përdorim metodën e bërrylit (Elbows) në të cilën ndërtojmë një grafik të gabimit në katror kundrejt numrit të klasterave. Në pikën ku kemi thyerjen e grafikut (berrylin) mund të themi se kemi numrin optimal të klasterave. Metoda k-means është e ndjeshme ndaj zhurmës dhe të dhënave të jashtme sepse një numër i vogël i të dhënave të tilla mund të ndikojnë thelbësisht në vlerën mesatare.

Një qasje për të bërë metodën k-means më efikase në grupe të mëdha të të dhënave është përdorimi i një grupi me të madh të elementetëve në klaster. Një qasje tjetër, është ideja e mikroklasterit, e cila së pari grupon objektet afër në "mikroklastera" dhe më pas kryhet klasterizimi nëpërmjet algoritmit.

#### 4.1.2 Algoritmi k-Medoids

Algoritmi k-means është një algoritëm i ndjeshëm ndaj skajeve, sepse objektet që janë shumë larg nga shumica e të dhënave, kur i caktohen një klasteri, mund të ndryshojnë në mënyrë drastike vlerën mesatare të tij. Kjo padashur ndikon në caktimin e objekteve të tjera të grupeve. Ky efekt përkeqësohet veçanërisht për shkak të përdorimit të gabimit në katror në formulën (4.1). Në vijim të punimit ne do të shohim shembullin e mësipërm [32].

"Si mund ta modifikojmë algoritmin k-means për të zvogëluar ndjeshmërinë e tillë ndaj skajeve?" Në vend që të marrim vlerën mesatare të objekteve në një grup si pikë referimi, ne kemi mundësi të zgjedhim objektet aktuale për të përfaqësuar klasterat, duke përdorur një objekt përfaqësues për secilin klaster. Metoda e ndarjes kryhet më pas bazuar në parimin e minimizimit të shumës së dallimeve midis secilit objekt p dhe objektit përfaqësues. Ky quhet kriteri i gabimit absolut dhe përcaktohet si:

$$E = \sum_{i=1}^k \sum_{p \in C_i} \text{dist}(p, o_i)^2 \quad (4-2)$$

ku E është shuma e gabimit absolut për të gjitha objektet p, dhe  $o_i$  është objekti përfaqësues i  $C_i$ . Kjo është baza e algoritmit K-Medoids, i cili i grupon objektet në klastera duke minimizuar gabimin absolut (Formula 4-2)

Kur  $k=1$ , mund të gjejmë te mesmen e saktë në  $O(n^2)$  (ku kompleksiteti i algoritmit shprehet në kohë). Sidoqoftë, kur k është një numër përgjithshëm pozitiv, problemi k-medoid është i vështirë.

Algoritmi i ndarjes rreth Medoids (PAM) është një realizim i njohur i klasterizimit të k-medoidsave. Ai trajton problemin në një mënyrë përsëritëse. Ashtu si algoritmi i k-means, objektet fillestare përfaqësuese (të njohura ndryshe si seeds) zgjidhen në mënyrë arbitrare. Të gjitha zëvendësimet e mundshme janë provuar. Procesi përsëritës i zëvendësimit të objekteve

përfaqësuese nga objekte të tjera vazhdon derisa cilësia e klasterit që rezulton nuk mund të përmirësohet nga ndonjë zëvendësim. Kjo cilësi matet me një funksion të kostos së pangjashmërisë mesatare midis një objekti dhe objektit përfaqësues të grupit të tij.

Specifikisht le të kemi  $o_1, \dots, o_k$  setin aktual të objekteve (medoid-save). Për të përcaktuar nëse një objekt  $O_{\text{random}}$ , është një zëvendësim i mirë për një medoid  $o_j$  ( $1 < j \leq k$ ), llogarisim distancën nga secili objekt  $p$  në objektin më të afërt në set-in  $\{o_1, \dots, o_{j-1}, O_{\text{random}}, o_{j+1}, \dots, o_k\}$  dhe përdorim distancën për të azhurnuar funksionin e kostos. Rivendosjet e objekteve në  $\{o_1, \dots, o_{j-1}, O_{\text{random}}, o_{j+1}, \dots, o_k\}$  janë të thjeshta. Supozojmë se objekti  $p$  i është caktuar aktualisht një klaster i përfaqësuar nga  $o_j$  medoid (Figura 4.2 a ose b). Nëse  $o_j$  zëvendësohet nga  $O_{\text{random}}$ , objekti  $p$  duhet të caktohet te  $O_{\text{random}}$  ose ndonjë klaster tjetër i përfaqësuar nga  $o_i$  ( $i \neq j$ ), cilado që është më e afërta. Për shëmbull, në Figurën 4.2 (a),  $p$  është më afër  $o_i$  dhe prandaj është rivendosur në  $o_i$ . Në figurën 4.2(b), megjithatë,  $p$  është më afër  $O_{\text{random}}$  dhe kështu është rivendosur në  $O_{\text{random}}$ . [32].

Po nëse, në vend të kësaj,  $p$  i është caktuar aktualisht një klasteri të përfaqësuar nga ndonjë objekt tjetër  $o_i$ , ( $i \neq j$ )?

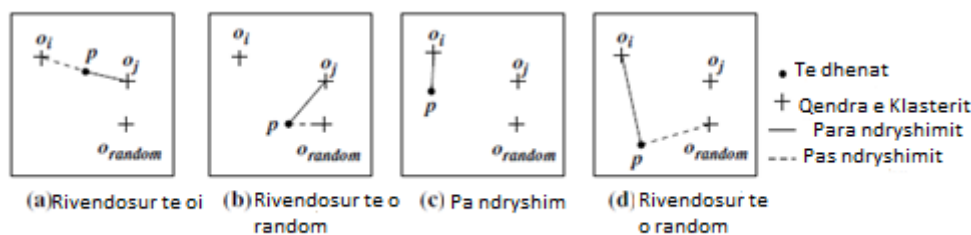


Figura 4-2 Katër rastet e funksionit të kostos për klasterizimin k-medoids

Objekti  $o_i$  mbetet grupit të përfaqësuar nga  $o_i$  për sa kohë që  $o$  është akoma më afër  $o_i$  sesa  $O_{\text{random}}$  (Figura 4.2(c)). Përndryshe,  $o$  është rivendosur në  $O_{\text{random}}$  (Figura 4.2. (d)). Sa herë që ndodh një rivendosje, kontribuohet në një ndryshim në gabimin absolut.

Prandaj, funksioni i kostos llogarit diferencën në vlerën absolute të gabimit nëse një objekt përfaqësues aktual zëvendësohet nga një objekt jopërfaqësues. Kostoja totale e ndërrimit është shuma e kostove të bëra nga të gjitha objektet jopërfaqësuese. Nëse kostoja totale është negative, atëherë  $o_j$  zëvendësohet me  $O_{\text{random}}$  sepse zvogëlohet gabimi aktual absolut  $E$ . Nëse kostoja totale është pozitive, objekti aktual përfaqësues,  $o_j$ , konsiderohet i pranueshëm dhe asgjë nuk ndryshon në përsëritje. Pyetja lind: "Cila metodë është më e mirë - k-means ose k-medoide?" Metoda k-medoids ka një performancë më të mirë se k-means në prani të zhurmës dhe skajeve sepse një medoid është më pak i ndikuar nga periferitë ose vlera të tjera ekstreme sesa një mesatare. Kompleksiteti i secilit iteracion në algoritmin k-medoids është  $O(k(n-k)^2)$ . Për vlerat e mëdha të  $n$  dhe  $k$ , një llogaritje e tillë bëhet shumë e kushtueshme dhe shumë e më shumë më e sesa k-means. Të dyja metodat kërkojnë nga përdoruesi të specifikojë  $k$ , numrin e klasterave. "Si mund ta rrisim performancën e algoritmit k-medoids?" Një algoritëm tipik i ndarjes së k-medoids si PAM (Figura 4.2) punon në mënyrë efektive për grupe të vogla të të dhënave, por nuk shkallëzohet mirë për grupe të mëdha të të dhënave. Për t'u marrë me grupe më të mëdha të të dhënave, një metodë e bazuar në marrjen e tyre mund të përdoret CLARA

(Clustering Large Applications). Në vend që të marrë në konsideratë tërësinë e të dhënave, CLARA përdor një shembull të rendit të të dhënave. Algoritmi PAM zbatohet më pas për të llogaritur medoidet më të mira nga data. Në shumë raste, një e dhënë e madhe funksionon mirë nëse krijohet në mënyrë që secili objekt të ketë probabilitet të barabartë për t'u zgjedhur. Objektet e përfaqsuara (medoidet) do të jenë të ngjashme me ato që do të ishin zgjedhur nga i gjithë grupi i të dhënave. CLARA ndërton grupe nga shembuj të shumtë të rastësishëm dhe kthen grupimin më të mirë. Kompleksiteti i llogaritjes së medoideve në një mostër të rastit është  $O(k^2 + k(n-k))$ , ku  $s$  është madhësia e kampionit,  $k$  është numri i grupeve dhe  $n$  është numri i përgjithshëm i objekteve. CLARA mund të merret me grupe më të mëdha të të dhënave sesa PAM. Efektiviteti i CLARA varet nga madhësia e kampionit [32].

**Algoritmi *k-medoids*:** PAM, një algoritëm *k-medoids* për ndarjen bazuar në medoid ose nënobjektet qëndrore.

**Input:**

- $k$ : numri i klasterave,
- $D$ : një grup të dhënash që përmbajnë  $n$  objekte.

**Output:** Një set prej  $k$  klasterash

**Metoda:**

- (1) në mënyrë arbitrare zgjedhim  $k$  objekte në  $D$  si objektet fillestare përfaqësuese ose seeds;
- (2) përsëritet;
- (3) caktojmë secilin objekt të mbetur në grup me objektin më të afërt përfaqësues;
- (4) zgjedhim rastësisht një objekt jo përfaqësues,  $O_{\text{random}}$ ;
- (5) llogarit koston totale,  $S$ , të ndërrimit të objektit përfaqësues,  $O_j$ , me  $O_{\text{random}}$ ;
- (6) Nëse  $S < 0$ , ndrysho  $O_j$  me  $O_{\text{random}}$
- (7) Përsërit derisa të mos ketë ndryshime

"Si mund ta përmirësojmë cilësinë dhe shkallëzimin e CLARA?" Gjithmonë duhet të kujtojmë se kur kërkojmë për medoide më të mira, PAM ekzaminon çdo objekt në grupin e të dhënave kundrejt çdo medoidi aktual, ndërsa CLARA kufizon medoidët kandidatë vetëm në një shembull të rastësishëm të grupit të të dhënave. Një algoritëm i rastësishëm i quajtur CLARANS (Klustering Large Applications based upon Randomized Search) paraqet një trade off mes koston dhe efektivitetit për përdorimin e kampioneve për të marrë grupimin (klasterizimin).

Së pari, ajo zgjedh rastësisht  $k$  objekte në klasterin e të dhënave si medoide aktuale. Pastaj zgjedh rastësisht një medoid  $x$  aktual dhe një objekt  $y$  që nuk është një nga medoidet aktuale[32].

## 4.2 Metoda hierarkike e klasterizimit

Ndërsa metodat e ndarjes plotësojnë kërkesat themelore të klasterave për të organizuar një sërë objektësh në një numër grupesh ekskluzive, në disa situata mund të dëshirojmë të ndajmë të dhënat tona në grupe në nivele të ndryshme si p.sh. në një hierarki të caktuar. Një metodë e një klasterizimi hierarkik funksionon duke grupuar objektet e të dhënave në një hierarki ose "tree" të klasterave. Përfaqësimi i objekteve të të dhënave në formën e një hierarkie është i dobishëm për përmbledhjen dhe vizualizimin e të dhënave.

Një shembull mund të ishte ndarja e blerjeve të klientëve sipas kategorive kryesore të mallrave në një qëndër tregtare. Në vazhdim mund të bëjmë nëndarjen e këtyre klasterave në nënklastera e me radhe deri në nivelin më të ulët të klasifikimit të artikujve.

Në disa aplikacione mund të besojmë se të dhënat mbajnë një strukturë hierarkike themelore që duam të zbulojmë. Në studimin e evolucionit, klasterizimi hierarkik mund të grupojë kafshët sipas veçorive të tyre biologjike për të zbuluar rrugët evolucionare, të cilat janë një hierarki e specieve. Si një shembull tjetër, grupimi i konfigurimeve të një loje strategjike (p.sh. shahu ose damat) në një mënyrë hierarkike mund të ndihmojë në zhvillimin e strategjive të lojës që mund të përdoren për të trajnuar lojtarët. Do të studiojmë metodat hierarkike të klasterizimit. Analiza fillon me një diskutim të klasterizimit hierarkik grumbullues kundrejt ndarjes, i cili organizon objektet në një hierarki duke përdorur përkatësisht një strategji nga poshtë-lart ose nga lart-poshtë. Këto metoda fillojnë me objekte individuale si grupe, të cilat bashkohen në mënyrë përsëritëse për të formuar grupe më të mëdha. Në të kundërt, metodat përçarëse fillimisht i lënë të gjitha objektet e dhëna të formojnë një grup, të cilin në mënyrë të përsëritur e ndajnë në grupe më të vegjël.

Metodat hierarkike të klasterizimit mund të hasin vështirësi në lidhje me zgjedhjen e pikave të bashkimit ose ndarjes. Një vendim i tillë është kritik, sepse sapo një grup objektësh mund të bashkohet ose të ndahet, procesi në hapin tjetër do të funksionojë në grupe të krijuara rishtas.

Vendimet e bashkimit ose ndarjes, nëse nuk zgjidhen mirë, mund të çojnë në grupe me cilësi të ulët. Për më tepër, metodat nuk shkallëzohen mirë, sepse çdo vendim i bashkimit ose ndarjes ka nevojë për të shqyrtuar dhe vlerësuar shumë objekte ose grupe.

Një drejtim për përmirësimin e cilësisë së klasterizimit të metodave hierarkike është integrimi i klasterizimit hierarkik me teknikat e tjera të klasterizimit, duke rezultuar në klasterizim me shumë faza (ose shumëfazore). Në këtë punim do të prezantojmë dy metoda të tilla, përkatësisht metoda BIRCH dhe Chamaleon. BIRCH fillon me ndarjen e objekteve në mënyrë hierarkike duke përdorur strukturat e pemëve, ku fletët ose nyjet jofletë të nivelit të ulët mund të shihen si "micro -klastera" në varësi të shkallës së rezolucionit.

Metoda Chamaleon eksploron modelimin dinamik në grupimin hierarkik. Ekzistojnë disa mënyra ortogonale për të kategorizuar metodat hierarkike të klasterizimit. Për shembull, ato mund të kategorizohen në metoda algoritmike, metodat Bayesian etj. Metodave aglomerative, përçarëse dhe shumëfazore janë algoritmike, domethënë ato i konsiderojnë objektet e të dhënave si grupe përcaktuese dhe llogaritin në përputhje me distancat përcaktuese midis objekteve. Metodave probabiliste përdorin modele probabiliteti për të kapur grupe dhe për të matur cilësinë e grupeve sipas përshtatjes së modeleve [32].

#### 4.2.1 Metoda hierarkike grumbulluese vs ndarëse

Një metodë hierarkike mund të jetë grumbulluese ose ndarëse, në varësi të kësaj nëse ndarja hierarkike është formuar në një nga poshtë-lart (bashkimi) ose topdown (ndarje). Le të shohim nga afër këto strategji.

**Një metodë klasterizmi hierarkik** përdor një strategji nga poshtë-lart. Zakonisht fillon duke e lënë secilin objekt të formojë klasterin e vet dhe bashkon në mënyrë të përsëritur klastera në grupe më të mëdha dhe më të mëdha, derisa të gjitha objektet të jenë në një klaster të vetëm ose përfundim të caktuar kushtet janë të kënaqura. Klasteri i vetëm bëhet rrënja e hierarkisë. Për hapin e bashkimit, gjenden dy klastera që janë më afër njëri-tjetrit (sipas ndonjë kategorie ngjashmërie) dhe i kombinon të dy për të formuar një. Për shkak se dy klastera bashkohen për përsëritje, ku secili klaster përmban të paktën një objekt, një metodë grumbulluese kërkon më së shumti përsëritje.

**Një metodë ndarjeje hierarkike e klasterave** përdor një strategji nga lart-poshtë. Fillon duke vendosur të gjitha objektet në një klaster, që është rrënja e hierarkisë. Pastaj ndan rrënjën grumbullohen në disa sub-klaster (nëngrupe) më të vegjël dhe në mënyrë rekursive i ndan ato klastera në ato më të vogla. Proçesi i klasterizimit (ndarjes) vazhdon derisa secili grup në nivelin më të ulët është mjaft koherent ose përmban vetëm një objekt, ose objektet brënda një grupi janë mjaft të ngjashëm me njëri-tjetrin. Në grupimet hierarkike grumbulluese ose ndarëse, një përdorues mund të specifikojë numrin e dëshiruar të klasterave si kusht përfundimi.

Analizojmë shembullim në vijim: Një metodë klasterizmi hierarkik vs një metodë ndarjeje hierarkike e klasterave. Figura 4.3 tregon zbatimin e AGNES (Agglomerative Nesting), një metodë klasterizimi hierarkike grumbulluese dhe DIANA (Divisive Analysis), një metodë ndarje hierarkike e klasterizimit, në një grup të dhënash prej pesë objektsh, *a*, *b*, *c*, *d*, *e*. Fillimisht, AGNES, metoda grumbulluese, e vendos secilin objekt në një grup të vetin. Më pas grupimet bashkohen hap pas hapi sipas deri në ndonjë kriter. Për shembull, klasterat C1 dhe C2 mund të bashkohen nëse një objekt në C1 dhe një objekt në C2 formojnë distancën minimale euklidiane midis dy objekteve nga grupet e diferencës [32].

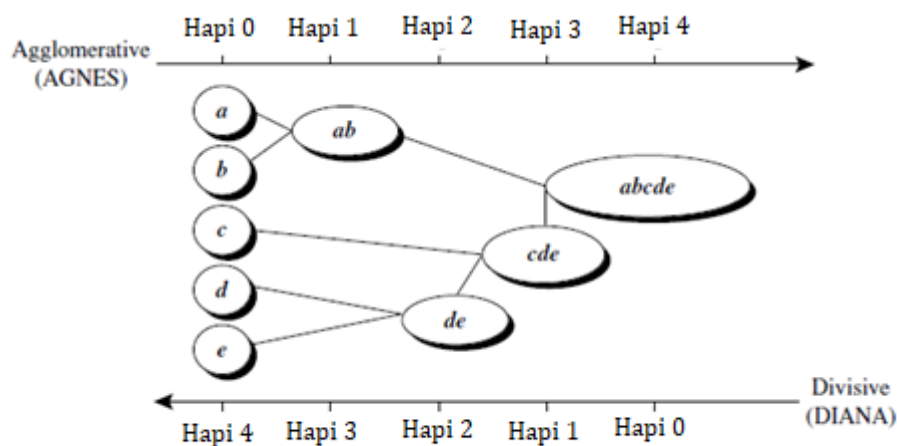


Figura 4-3 Klasterizimi agglomerative dhe hierarkik në objektet {*a*, *b*, *c*, *d*, *e*}.

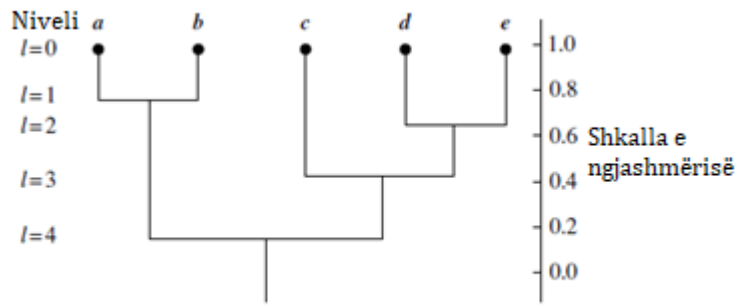


Figura 4-4 Paraqitja Dendogram për klaster-izimin hierarkik për objektet data {a, b, c, d, e}.

Kjo është një qasje me një lidhje të vetme në atë që përfaqësohet secili klaster nga të gjitha objektet në klaster dhe ngjashmëria midis dy klasterave matet me ngjashmërinë e çiftit më të afërt të pikave të të dhënave që u përkasin grupeve të ndryshme. Proçesi i bashkimit të klasterave përsëritet derisa të gjithë objektet përfundimisht të bashkohen për të formuar një klaster të vetëm.

DIANA, metoda përçarëse, vazhdon në mënyrë të kundërt. Të gjitha objektet përdoren për të formuar një klaster fillestar. Klasteri ndahet sipas disa parimeve të tilla si distanca maksimale euklidiane ndërmjet objekteve më të afërta fqinje në tufë. Proçesi i ndarjes së klasterave përsëritet derisa, përfundimisht, secili grup i ri përmban vetëm një objekt të vetëm.

Një strukturë peme e quajtur dendrogram përdoret zakonisht për të përfaqësuar procesin e klasterizimit hierarkik. Ajo tregon se si objektet grupohen së bashku (në një metodë grumbulluese) ose ndahen (në një metodë ndarëse) hap pas hapi. Figura 4-4 tregon një dendrogram për pesë objektet e paraqitura në Figurën 4.3, ku  $l=0$  tregon pesë objektet si grupe unike në nivelin 0. Në  $l=1$ , objektet a dhe b janë grupuar së bashku për të formuar klasterin e pare dhe qëndrojnë sëbashku në të gjitha nivelet pasuese.

Ne gjithashtu mund të përdorim një bosht vertikal për të treguar shkallën e ngjashmërisë midis klasterave. Për shembull, kur ngjashmëria e dy grupeve të objekteve, {a, b} dhe {c, d, e} p.sh. është afërsisht 0.16, ato bashkohen së bashku për të formuar një klaster i vetëm. Një sfidë me metodat përçarëse është si të ndash një grup të madh në disa më të vegjël. Për shembull, ekzistojnë mënyra të mundshme  $2^n - 1$  për të ndarë një klaster n objekteve në dy nënbashkësi ekskluzive, ku n është numri i objekteve. Kur n është i madh, nga ana kompjuterike është e pamundur të shqyrtohen të gjitha mundësitë.

Për hir të efikasitetit, metodat e përçarjes zakonisht nuk i rivleresojnë vendimet e marra të ndarjes. Sapo të ndahet një klaster, çdo ndarje alternative e kësaj grumbulli nuk do të konsiderohet përsëri. Për shkak të sfidave në metodat përçarëse, ka shumë më tepër metoda grumbulluese sesa metoda përçarëse [32].

#### 4.2.2 Përlllogaritja e distances në metodat algoritmike

Kur jemi duke përdorur një metodë grumbulluese ose një metodë përçarëse, një nevojë thelbësore është të masim distancën midis dy klasterave, ku secili klaster në përgjithësi është një grup objekteve. Katër masa të përdorura gjerësisht për distancën midis klasterave janë si më poshtë, ku  $|p-p'|$  është distanca midis dy objekteve ose pikave (points), p dhe p';  $m_i$  është

mesatarja e klasterit,  $C_i$ ; dhe  $n_i$  është numri i objekteve në  $C_i$ . Ato njihen gjithashtu si masa të lidhjes (linkage measures) [32].

$$\text{Distanca minimale: } dist_{min}(C_i, C_j) = \min_{p \in C_i, p' \in C_j} \{|p - p'|\} \quad (4.3)$$

$$\text{Distanca maksimale: } dist_{max}(C_i, C_j) = \max_{p \in C_i, p' \in C_j} \{|p - p'|\} \quad (4.4)$$

$$\text{Diferenca e distancave: } dist_{mean}(C_i, C_j) = |m_i - m_j| \quad (4.5)$$

$$\text{Distanca mesatare: } dist_{mesatare}(C_i, C_j) = \frac{1}{n_i n_j} \sum_{p \in C_i, p' \in C_j} |p - p'| \quad (4.6)$$

Kur algoritmat përdorin distancat minimale  $d_{min}(C_i, C_j)$ , për të llogaritur distancat midis klasterave, ato quhet algoritëm i klasterizimit të fqinjët më të afërt (nearest-neighbor clustering algorithm). Për më tepër, nëse procesi i klasterizimit përfundon kur distanca midis grupeve më të afërt tejkalon një prag të përcaktuar nga përdoruesi, quhet algoritëm me një lidhje të vetme (single-linkage algorithm). Nëse i shohim pikat e të dhënave si nyje të një grafiku, me skajet që formojnë një shteg midis nyjeve në një klaster, atëherë bashkimi i dy klasterave,  $C_i$  dhe  $C_j$ , korrespondon me shtimin e një lidhjeje midis çiftit më të afërt të nyjeve në  $C_i$  dhe  $C_j$ . Për shkak se skajet që lidhin klastera gjithmonë shkojnë midis grupeve të dallueshëm, grafiku që rezulton do të gjenerojë një pemë.

Kështu, një algoritëm klasterizimi hierarkik që përdor masën e distancës minimale quhet gjithashtu një algoritëm i pemës së shtrirjes minimale, ku një pemë shtrirje e një grafi është një pemë që lidh të gjitha kulmet, dhe një pemë minimale e shtrirjes është ajo me shumën më të vogël të peshat e skajit.

Kur algoritmat përdorin distancat maksimale  $d_{max}(C_i, C_j)$ , për të llogaritur distancat midis klasterave, ato quhet ndryshe dhe si algoritmi i klasterizimit të fqinjët më të largët (farthest-neighbor clustering algorithm). Nëse procesi i klasterizimit përfundon kur distanca maksimale midis klasterave më të afërt tejkalon një prag të përcaktuar nga përdoruesi, ai quhet një algoritëm i lidhjes së plotë. Duke parë pikat e të dhënave si nyje të një grafiku, me skajet që lidhin nyjet, ne mund të mendojmë për secilin grup si një nëngraf i plotë, domethënë, me skajet që lidhin të gjitha nyjet në grupe. Distanca midis dy grupeve përcaktohet nga nyjet më të largëta në to [32].

Algoritmet e fqinjët më të largët kanë tendencë të minimizojnë rritjen e diametrit të grupeve në secilën përsëritje. Nëse grupimet e vërteta janë njaft kompakte dhe me madhësi afërsisht të barabartë, metoda do të prodhojë grupe me cilësi të lartë. Përndryshe, klasterat e prodhuara mund të jenë të pakuptimta. Përdorimi i distancës mesatare është një kompromis midis minimumit dhe distancat maksimale dhe kapërcen problemin e ndjeshmërisë ndaj zhurmave. Ndërsa distanca mesatare është më e thjeshta për t'u llogaritur, llogaritja e vektorit mesatar për të dhënat kategorike mund të jetë e vështirë ose e pamundur për t'u kryer.

Le të zbatojmë klasterizimin hierarkik në grupin e të dhënave të Figurës 4.5 (a). Figura 4.5 (b) tregon dendrogramin duke përdorur një lidhje të vetme. Figura 4.5 (c) tregon rastin duke përdorur lidhje të plotë, ku skajet midis grupeve {A,B,J,H} dhe {C,D,G,F,E} hiqen për lehtësinë e paraqitjes. Ky shembull tregon se duke përdorur lidhje të vetme mund të gjejmë grupe hierarkike të përcaktuara nga afërsia lokale, ndërsa lidhja e plotë tenton të gjejë grupe që zgjedhin afërsinë globale. Ekzistojnë variacione të katër masave thelbësore të lidhjes të sapo diskutuara. Për shembull, ne mund të masim distancën midis dy grupeve me distancën midis objektet qendrore të klasterave [32].

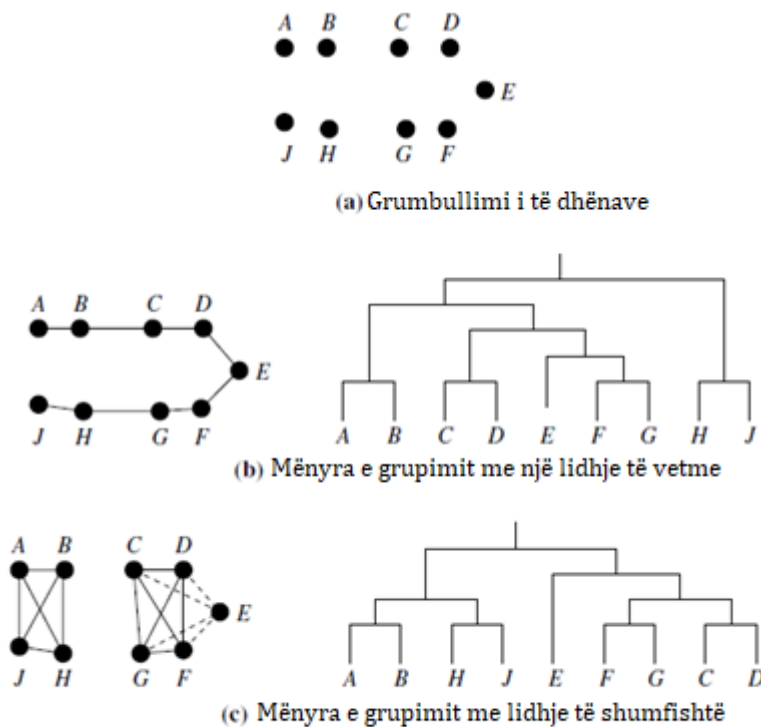


Figura 4-5 Klasterizimi linkage

### 4.2.3 Algoritmi BIRCH

BIRCH (Multiphase Hierarchical Clustering Using Clustering Feature Trees) është krijuar për klasterizimin e një sasive të madhe të të dhënave numerike duke integruar klasterizimin hierarkik (në fazën fillestare të mikroklasifikimit) dhe metoda të tjera të klasterizimit siç është ndarja përsëritëse (në fazën e mëvonshme të makro-klasifikimit). Ai kapërcen dy vështirësitë në metodat grumbulluese: (1) shkallëzimi dhe (2) paaftësia për të zhbërë atë që është bërë në hapin e mëparshëm. BIRCH përdor nocionet e tiparit të klasterizimit për të përmbledhur një klaster dhe pemën CF për të përfaqësuar një hierarki të grupeve. Këto struktura ndihmojnë metoda e klasterizimit të arrijnë shpejtësi dhe shkallëzim të mirë në transmetim të madh apo edhe bazat e të dhënave [32].

Konsiderojmë një klaster me  $n$ -dimensione ose pika. Karakteristika e grupimit (CF) e klasterit është një vektor 3-D që përmbledh informacionin rreth grupeve të objekteve. Ajo është përcaktuar si më poshtë:



$$CF = \langle n, LS, SS \rangle \quad (4.7)$$

Ku LS është shuma lineare e  $n$  pikave (points) (p. sh. ,  $\sum_{i=1}^n x_i$ ), dhe SS është shuma në katror e të gjitha data point (p. sh. ,  $\sum_{i=1}^n x_i^2$ )

Një tipar i klasterit është në thelb një përmbledhje e statistikave për klasterin e caktuar. Duke përdorur një veçori të grupimit, ne lehtë mund të nxjerrim shumë statistika të dobishme të një klasteri. Për shembull, centroidi i grupeve,  $x_0$ , rrezja,  $R$  dhe diametri,  $D$ , janë:

$$x_0 = \frac{\sum_{i=1}^n x_i}{n} = \frac{LS}{n} \quad (4.8)$$

$$R = \sqrt{\frac{\sum_{i=1}^n (x_i - x_0)^2}{n}} = \sqrt{\frac{nSS - 2LS^2 + nLS^2}{n^2}} \quad (4.9)$$

$$D = \sqrt{\frac{\sum_{i=1}^n \sum_{j=1}^n (x_i - x_j)^2}{n(n-1)}} = \sqrt{\frac{2nSS - 2LS^2}{n(n-1)}}. \quad (4.10)$$

Këtu,  $R$  është distanca mesatare nga objektet anëtare në centroid dhe  $D$  është distanca mesatare në çift brënda një klasteri. Të dy  $R$  dhe  $D$  pasqyrojnë ngushtësinë e klasterit. Përfundimisht një klaster duke përdorur një tipar të tij të shmangë ruajtjen e informacionit të detajuar në lidhje me objekte ose pika individuale. Në vend të kësaj na duhet vetëm një madhësi konstante e hapësirës për të ruajtur tiparin e klasterizimit. Ky është çelësi për efikasitetin e BIRCH në hapësirë. Për dy grupe të ndara,  $C1$  dhe  $C2$ , me karakteristikat e grupeve përkatësisht  $CF_1 = \langle n_1, LS_1, SS_1 \rangle$  dhe  $CF_2 = \langle n_2, LS_2, SS_2 \rangle$  tipari i bashkimit për grupin që formohet nga bashkimi i  $C1$  dhe  $C2$  është:  $CF_1 + CF_2 = \langle n_1 + n_2, LS_1 + LS_2, SS_1 + SS_2 \rangle$ .

Një pemë CF është një pemë e ekuilibruar në lartësi që ruan tiparet e klasterizimit hierarkik. Një shembull është treguar në Figurën 4-6. Sipas përkufizimit, një nyje jo gjethe në një pemë ka pasardhës ose "fëmijë". Nyjet pa gjethe ruajnë shumën e CF-ve të fëmijëve të tyre dhe kështu përmbledhin informacionet e grupeve për fëmijët e tyre. Një pemë CF ka dy parametra: faktorin e degëzimit,  $B$  dhe pragun,  $T$ . Faktori i degëzimit specifikon numrin maksimal të fëmijëve për nyjen pa gjethe. Parametri i pragut specifikon diametri maksimal i nëngrupeve të ruajtura në nyjet gjethe të pemës. Këto dy parametra në mënyrë kontrollojnë madhësinë e pemës që rezulton.

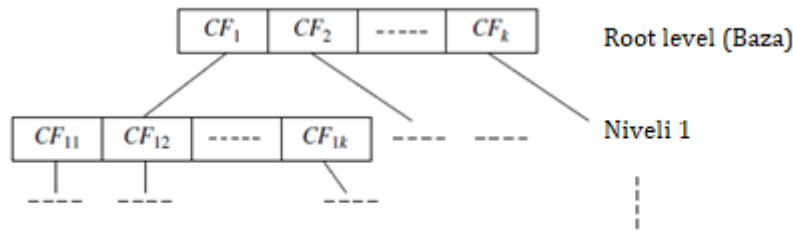


Figura 4-6 Struktura CF – pemë

Duke pasur parasysh një sasi të kufizuar të memories kryesore, në BIRCH është të minimizohet koha e nevojshme për hyrjen / daljen (I/O). BIRCH zbaton një teknikë klasterizimi shumëfazor. Një skanim i vetëm i grupit të të dhënave sjell një grupim bazë, të mirë dhe një ose më shumë skanime shtesë mund të përdoren opsionale për të përmirësuar më tej cilësinë. Dy fazat primare janë:

- Faza 1: BIRCH skanon databazën të krijojë një pemë fillestare me memorje CF, e cila mund të shihet si një fluks me shumë nivele të të dhënave që përpiqet të ruajë ato në strukturat e klasterizimit.
- Faza 2: BIRCH aplikon një algoritëm klasterizimi (të zgjedhur) për nyjet, gjethe të pemës CF, e cila heq grupe të rralla si skajet dhe grupon në klastera të dendura në ato më të mëdha [32].

Për Fazën 1, pema CF ndërtohet në mënyrë dinamike kur futen objektet. Kështu, metoda është inkrementuese. Një objekt futet në hyrjen më të afërt të gjetheve (subklaster). Nëse diametri e subklasterit së ruajtur në nyjen e gjetheve pas hyrjes është më e madhe se vlera e pragut, atëherë nyja e gjetheve dhe ndoshta nyjet e tjera ndahen. Pas futjes së objektit të ri, informacioni rreth objektit kalon drejt rrënjës së pemës. Përmasa e pemës CF mund të ndryshohet duke modifikuar pragun. Nëse madhësia e kujtesës që nevojitet për ruajtjen e pemës CF është më e madhe se madhësia e memories kryesore, atëherë mund të specifikohet vlera më e madhe e pragut dhe pema CF është rindërtuar [32].

Proçesi i rindërtimit kryhet duke ndërtuar një pemë të re nga nyjet e gjetheve të së pemës së vjetër. Kështu, proçesi i rindërtimit të pemës bëhet pa nevojën e rileximit të të gjitha objekteve. Për ndërtimin e pemës, të dhënat duhet të lexohen vetëm një herë. Sapo të ndërtohet pema CF, çdo tip algoritëm klasterizimi, siç është një algoritëm tipik i ndarjes, mund të përdoret me pemën CF në fazën 2.

"Sa efektive është BIRCH?" Kompleksiteti kohor i algoritmit është  $O(n)$ , ku  $n$  është numri i objekteve që do të grumbullohen. Eksperimentet kanë treguar shkallëzimin linear të algoritmit në lidhje me numrin e objekteve dhe cilësinë e mirë të klasterizimit të të dhënave. Sidoqoftë, meqenëse secila nyje në një pemë CF mund të mbajë vetëm një numër të kufizuar të shënimeve për shkak të madhësisë së saj, një nyje e pemës CF jo gjithmonë korrespondon me atë që një përdorues mund ta konsiderojë një klaster natyral. Për më tepër, nëse klasterat nuk janë në formë sferike, BIRCH nuk performon mirë sepse përdor nocionin e rrezes ose diametrit për të kontrolluar kufirin e një grupi.

Idetë e veçorive të grupeve dhe pemëve CF janë zbatuar përtej nocionit të BIRCH. Idetë janë huazuar nga shumë të tjerë për të trajtuar problemet e bashkimit të grupeve dhe të dhëna dinamike.

#### 4.2.4 Metoda Chameleon

Chameleon (Multiphase Hierarchical Clustering Using Dynamic Modeling) është një algoritëm hierarkik i klasterizimit që përdor modelimin dinamik për të përcaktuar ngjashmërinë midis çifteve të klasterave. Në Chameleon, vlerësohet ngjashmëria e klasterave bazuar në (1) sa mirë janë objektet e lidhura brenda një klasteri dhe (2) afërsia e klasterave. Kjo do të thotë, dy grupe bashkohen nëse ndërlidhja e tyre është e lartë dhe ato janë afër së bashku. Kështu, Chameleon nuk varet nga një model statik, i furnizuar nga përdoruesit dhe automatikisht mund të përshtaten me karakteristikat e brendshme të klasterave që bashkohen. Procesi i bashkimit lehtëson zbulimin e grupeve natyrore dhe homogjene dhe zbatohet te të gjitha llojet e të dhënave për sa kohë që mund të specifikohet një funksion i ngjashmërisë [32].

Figura 4.7 ilustron se si metoda Chamaleon funksionon. Kjo metodë përdor grafën e fqinjët k më të afërt (k-nearest-neighbor graph) ku secili kulm i grafikut përfaqëson një objekt të të dhënave dhe ekziston një skaj midis dy kulmeve (objekteve). Skajet peshohen për të pasqyruar ngjashmërinë midis objekteve. Metoda Chamaleon përdor një algoritëm të ndarjes së grafikëve për të ndarë në nën-grupe relativisht të vogla, të tilla që minimizon prerjen e skajit. Kjo do të thotë, një tufë C ndahet në nëngrupet  $C_i$  dhe  $C_j$  në mënyrë që të minimizohet pesha e skajeve që do të priten nëse C ndahet në  $C_i$  dhe  $C_j$ . Kjo metodë vlerëson ndërlidhshmërinë absolute midis grupeve  $C_i$  dhe  $C_j$ .

Metoda Chamaleon më pas përdor një algoritëm grumbullues hierarkik që bashkon në mënyrë të përsëritur subklasterave bazuar në ngjashmërinë e tyre. Për të përcaktuar çiftet e subklasterave më të ngjashëm, ajo merr parasysh si ndërlidhshmërinë ashtu edhe afërsinë e klasterave. Në mënyrë të veçantë, kjo metodë përcakton ngjashmërinë midis secilës palë klasteri  $C_i$  dhe  $C_j$  sipas ndërlidhjes së tyre relative,  $RI(C_i, C_j)$  dhe afërsinë e tyre relative  $RC(C_i, C_j)$ .

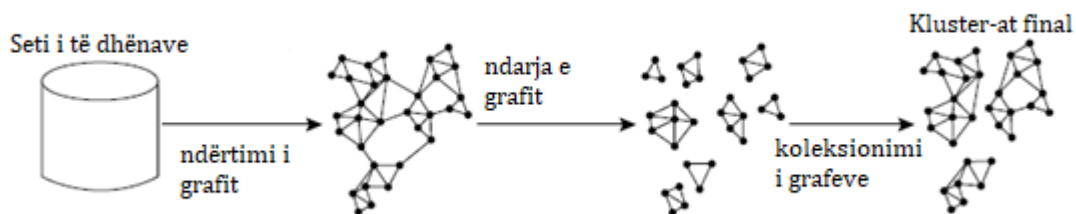


Figura 4-7 Kameleon: Klasterizim hierarkik i bazuar në k-fqinjët më të afërt dhe modelimi dinamik

- Ndërlidhja relative  $RI(C_i, C_j)$ , ndërmjet dy klasterave,  $C_i$  dhe  $C_j$  mund të përcaktohet si ndërlidhshmëri absolute midis  $C_i$  dhe  $C_j$ , e normalizuar në lidhje me ndërlidhjen e brendshme të dy klasterave,  $C_i$  dhe  $C_j$ . Domethënë:

Ku  $EC_{\{C_i, C_j\}}$  është prerja e skajit siç është përcaktuar më parë për një grup që përmban të  $C_i$  dhe  $C_j$ . Në mënyrë të ngjashme  $EC_{C_i}$  është shuma minimale e skajeve të prera që ndahen  $C_i$  (ose  $C_j$ ) në dy pjesë afërsisht të barabarta.

- Afërsia relative RC ( $C_i, C_j$ ), midis një çifti të klasterave  $C_i$  dhe  $C_j$  është afërsia absolute midis  $C_i$  dhe  $C_j$ , e normalizuar në lidhje me afërsinë e brendshme të dy klasterave,  $C_i$  dhe  $C_j$ . Ajo është e përcaktuar ndryshe dhe si:

$$RI(C_i, C_j) = \frac{|EC_{\{C_i, C_j\}}|}{\frac{1}{2}(|EC_{C_i}| + |EC_{C_j}|)} \quad (4.11)$$

Ku  $S_{EC_{\{C_i, C_j\}}}$  është pesha mesatare e skajeve që lidhin kulmet në  $C_i$  me kulmet në  $C_j$ , dhe  $S_{EC_i}$  është pesha mesatare e skajeve që i përkasin përgjysmimit min të klasterit  $C_i$ .

Metoda Chameleon është treguar se ka fuqi më të madhe në zbulimin e grupeve me cilësi të lartë në mënyrë arbitrare sesa disa algoritme të njohur si BIRCH dhe DBSCAN me bazë densitetin. Sidoqoftë, kostoja e përpunimit për të dhëna me dimensione të larta mund të kërkojë një kompleksitet  $O(n^2)$  për  $n$  objekte në rastin më të keq. [32].

$$RC(C_i, C_j) = \frac{\overline{S_{EC_{\{C_i, C_j\}}}}}{\frac{|C_i|}{|C_i| + |C_j|} \overline{S_{EC_{C_i}}} + \frac{|C_j|}{|C_i| + |C_j|} \overline{S_{EC_{C_j}}}} \quad (4.12)$$

### 4.3 Teknikat e Klasifikimit

Klasifikimi është një formë e analizës së të dhënave që synon ndërtimin ose gjetjen e modelit që përshkruan dhe është në gjendje të bëjë parashikime për një sasi input të dhënash. Lista e këtyre të fundit është në pjesën më të madhe të rasteve një listë variablash ose atributesh të tipit diskret ose jo numerik. Shembull i një modeli klasifikimi është ai i kategorizimit të aplikimeve për kredi pranë një banke, në aplikime të sigurta dhe në aplikime me rrezik të lartë. Duke njohur modelin i cili përfaqëson më së miri aplikimet e deritanishme pranë bankës, për çdo aplikim të ri ne mund të vlerësojmë në cilën fashë të portofolit të klientëve bën pjesë klienti dhe aplikimi i tij.

Ne rastin e studimit tone, do të parashikojmë klasën e klientit të biznesit retail, bazuar në atributet e tij. Ky aplikim është shumë i rëndësishëm për këto biznese pasi i ndihmon në marketingun e targetuar, rritjen e cilesisë së shërbimit, mire-manxhimin e stokut dhe rrjedhimisht në rritje të të ardhurave.

Klasifikimi ka aplikime të shumta në fusha të ndryshme të biznesit si retail, marketing, mjekësi, prodhim ose banking.

#### 4.3.1 Konceptet Themelore

Në analizën e konceptit të klasifikimit do të përshkruhen qasjet e përgjithshme të klasifikimit si një proces me dy hapa. Në hapin e parë, do të ndërtohet një model klasifikimi bazuar në të dhënat e mëparshme. Në hapin e dytë, ne do të përcaktojmë nëse saktësia e modelit është e pranueshme dhe nëse po, do përdoret ky model për të klasifikuar të dhënat e reja. Një punonjës i kredive bankare ka nevojë për analiza të të dhënave të saj për të mësuar se cilët aplikantë kredie janë "të sigurt" dhe cilët janë "me rrezik" për bankën. Një menaxher shitjesh ka nevojë për të dhëna që ta ndihmojnë të parashikojë nëse një klient me një profil të caktuar do të blejë një x artikull. Një studiues mjekësor dëshiron të analizojë për shembull të dhënat e kancerit të gjirit për të parashikuar se cilin nga tre trajtimet specifike një pacient duhet të marrë. Në secilin prej këtyre shembujve, detyra e analizës së të dhënave është klasifikimi, ku një model ose një klasifikues ndërtohet për të parashikuar etiketime të klasave (kategorive), të tilla si "i sigurt" ose "i rrezikshëm" për të dhënat e aplikimit të kredisë; "Po" ose "jo" për të dhënat e marketingut; ose "trajtimi A", "trajtimi B", ose "trajtimi C" për të dhënat mjekësore. Këto kategori mund të përfaqësohen nga vlera diskrete, ku renditja ndërmjet vlerave nuk ka kuptim. Për shembull, vlerat 1, 2 dhe 3 mund të përdoren për të përfaqësuar klasat A, B dhe C, ku nuk ka asnjë renditje të nënkuptuar midis këtij grupi të regjimeve të trajtimit. Supozoni se menaxheri i marketingut dëshiron të parashikojë se sa do shpenzojë një klient i caktuar gjatë një blerjeje. Kjo detyrë e analizës së të dhënave është një shembull parashikimi numerik, ku modeli i ndërtuar parashikon një funksion me vlerë të vazhdueshme, ose vlera e renditur, në vend të një etiketimi të klasës. Ky është një model parashikues. Analiza e regresit është një metodologji statistikore që përdoret shpesh për parashikimin numerik; prandaj këto dy terma përdoren si sinonime, megjithëse ekzistojnë metoda të tjera për parashikimin numerik. Klasifikimi dhe parashikimi numerik janë dy llojet kryesore të problemeve të parashikimit. Ky paragraf përqëndrohet në klasifikimin [32].

Megjithatë njihet emërtimi i klasës për secilin set trajnimi, ky hap njihet gjithashtu si të mësuarit e mbikëqyrur (d.m.th., mësimi i klasifikuesit "mbikëqyrjet" e i tregohet se cilës klasë i përket çdo set trajnimi). Ndryshe nga të mësuarit pa mbikëqyrje (ose klasterizimi) në të cilin nuk dihet paraprakisht emërtimi i klasës për secilin set trajnimi. Për shembull, nëse nuk e dimë klasën e klientëve në një biznes Retail paraprakisht, ne mund të përdorim klasterizimin për t'u përpjekur të përcaktojmë grupe të ngjashme, të cilat mund të korrespondojnë me grupet e klientëve të cilët shpenzojnë në mënyrë të ngjashme ose blejnë në kategori të ngjashme. Ky hap i parë i procesit të klasifikimit mund të shihet gjithashtu si mësimi i një harte ose funksioni,  $y = f(X)$ , që mund të parashikojë emërtimin e klasës shoqëruese të një seti  $X$  të dhënash. Në këtë këndvështrim, ne dëshirojmë të mësojmë një hartëzim ose funksion që ndan klasat e të dhënave. Tipikisht, ky hartëzim përfaqësohet në formën e rregullave të klasifikimit, pemëve të vendimit, ose formula matematikore. Në shembullin tonë, hartëzimi paraqitet si rregulla klasifikimit që identifikojnë klientet e tipit 1 apo klientet e tipit 2. Rregullat mund të përdoren për të kategorizuar të dhënat e ardhshme si dhe për të siguruar një pasqyrë më të thellë në përmbajtjen e të dhënave. Ato gjithashtu ofrojnë një paraqitje të të dhënave të kompresuara.

Hap i rëndësishëm para validimit të një modeli klasifikimi është dhe testimi i tij, duhet të provohet nëse saktësia e tij është e pranueshme ose jo. Kjo realizohet përmes aplikimit të modelit në një set të dhënash, të papërdorur gjatë fazës së zhvillimit, por për të cilat ne kemi dijeni në realitet që në cilën kategori bëjnë pjesë. Saktësia e modelit do të vlerësohet përmes raportit të kategorizimeve të sakta kundrejt total setit të elementëve të setit të të dhënave test. Nëse saktësia e klasifikuesit konsiderohet e pranueshme, klasifikuesi mund të përdoret për të klasifikuar setet e të dhënave të ardhshme për të cilat emërtimet e klasës nuk dihen.

## 4.4 Rrjetat neurale

Rrjetat neurale artificiale ose thjeshtë rrjetat neurale janë krijuar nisur nga rrjetat neurale biologjike. Një neuron real biologjik ose një qelizë neurale përbëhet nga dendritet, trupi i qelizës dhe një akson i cili shkon drejt terminaleve sinaptike. Një neuron transmeton informacion nëpërmjet sinjaleve elektrokimike. Në momentin që ka një përqëndrim të mjaftueshëm jonesh në dendritet e neuronit, ai gjeneron një puls elektrik përgjatë aksonit, i cili quhet një potencial veprimi. Ky potencial veprimi aktivizon terminalet sinaptike, duke çliruar më shumë jone dhe duke shkaktuar rrjedhjen e informacionit në dendritet e neuroneve të tjera. Truri i njeriut ka një numër neuronesh në rendin e 100 miliardëve, ku secili neuron ka 1.000 deri në 10.000 lidhje me neuronet e tjera. Kështu që truri i njeriut është një rrjet neural me 100 trilion deri në një kuadrilion ( $10^{15}$ ) ndërlidhje! Mesa dihet deri tani, të mësuarit realizohet duke përshtatur forcën sinaptike, meqënëse sinjalet sinaptike mund të jenë ngacmuese ose penguese, duke rritur ose duke ulur respektivisht probabilitetin që neuroni sinaptik të gjenerojë një potencial veprimi [33].

Rrjetat neurale artificiale janë të përbëra nga neurone abstrakte që përpiqen të veprojnë njësoj si neuronet reale në një nivel shumë të lartë. Ato mund të përshkruhen nëpërmjet një grafi të drejtuar, të peshuar  $G = (V, E)$ , ku çdo nyje  $v_i \in V$  përfaqëson një neuron dhe çdo hark i drejtuar  $(v_i, v_j) \in E$  përfaqëson një lidhje nga sinaptike në dendritike nga  $v_i$  në  $v_j$ . Peshat e harkut  $w_{ij}$  përcakton forcën sinaptike. Rrjetat neurale karakterizohen nga tipi i funksionit të aktivizimit, i cili përdoret për gjenerimin e një output-i dhe nga arkitektura e rrjetit, në këndvështrimin e mënyrës se si nyjet janë të ndërlidhura. Për shembull, nëse grafi është një graf i drejtuar pa cikle ose me cikle, nëse grafi është hierarkik ose jo etj. Është e rëndësishme të theksohet se një rrjet neural krijohet me qëllim që të përfaqësojë dhe të mësojë informacion duke përshtatur peshën sinaptike [33].

Në rrjetat neurale dhe në funksionet e saj është e nevojshme të përmendim termin *bias*, kuptimin dhe funksionin mjaft të rëndësishëm që ka ky element nëpër funksione.

Bias është elementi i cili nëpër funksione mundëson zhvendosjen e kurbës së funksionit si të ishte një konstante në funksionet linear, ku kurba mund të transpozohet majtas ose djathtas përmes vlerës së një konstanteje.

Në një skenar pa bias, input-i për shembull në një funksion aktivizimi në rrjetat neurale do ishte thjeshtë:  $\mathbf{x} * \mathbf{w}_0$ . Në një skenar me bias, inputi në funksionin e aktivizimit do ishte:  $\mathbf{x} * \mathbf{w}_0 + \mathbf{b} * \mathbf{w}_1$  ku  $\mathbf{b}$  është bias dhe  $\mathbf{w}_1$  është peshat e konstantes bias. Kështu jemi në gjendje të bëjmë një spostim të kurbës së funksionit në përdorim përmes një vlere që vjen nga prodhimi i konstantes bias me peshën e saj.

### 4.4.1 Neuron artificial: Funksionet e aktivizimit

Një neuron artificial vepron një njësi përpunimi, e cila fillimisht mbledh sinjalet hyrëse nëpërmjet një shume të peshuar dhe më pas aplikon një funksion për të gjeneruar një output. Për shembull, një neuron binar do të gjenerojë një output '1' sa herë që sinjali i kombinuar tejkalon një prag dhe gjeneron një '0' në të kundërt.

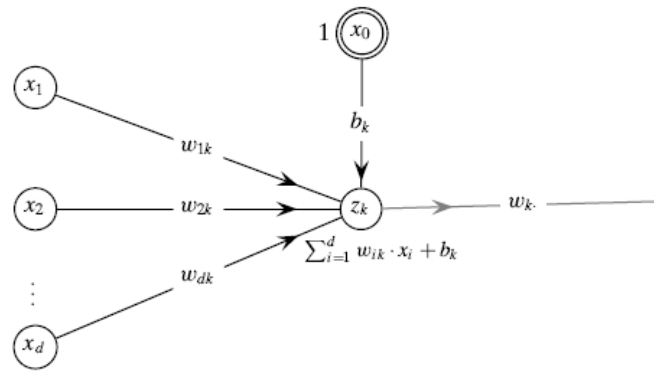


Figura 4-8 Neuroni artificial: Mbledhja dhe Aktivizimi

Figura 4-8 tregon skemën e një neuroni  $z_k$ , i cili ka harqe hyrëse nga neuronet  $x_1 \dots x_d$ . Për thjeshtësi, si emri edhe vlera e daljes së një neuroni paraqiten me një njëjtin simbol. Kështu që  $x_i$  paraqet neuronin  $i$ , si dhe vlerën e atij neuroni. Input-i neto në  $z_k$ , e paraqitur si  $net_k$  jepet si shuma e peshuar[33]:

$$net_k = b_k + \sum_{i=1}^d w_{ik} \cdot x_i = b_k + \mathbf{w}^T \mathbf{x} \quad (4.13)$$

ku  $w_k = (w_{1k}, w_{2k} \dots w_{dk})^T \in R^d$  dhe  $x = (x_1, x_2 \dots x_d)^T \in R^d$  është një pikë hyrjeje.

Vihet re se  $x_0$  është një neuron i veçantë bias-i, vlera e të cilit është gjithmonë 1 dhe pesha nga  $x_0$  në  $z_k$  është  $b_k$ , e cila specifikon termin e bias-it për neuronin. Si përfundim vlera e daljes  $z_k$  jepet si një funksion aktivizimi  $f(\cdot)$ , i aplikuar mbi input-in neto në  $z_k$ .

$$z_k = f(net_k)$$

Më pas vlera  $z_k$  transmetohet përmes harqeve dalëse nga  $z_k$  drejt neuroneve të tjera. Neronet ndryshojnë bazuar në tipin e funksionit të aktivizimit të përdorur. Disa funksione aktivizimi të përdorur zakonisht, të ilustruar në figurën 4-9, janë:

**Funksioni linear/ Identiteti:** Identiteti është funksioni më i thjeshtë i aktivizimit; ai thjeshtë kthen vlerën e argumentit të tij dhe mund të quhet ndryshe një funksion aktivizimi linear:

$$f(net_k) = net_k \quad (4.14)$$

Figura 4-9(a) jep paraqitjen grafike të funksionit të identitetit. Për të ekzaminuar rolin e termin të bias-it, vihet re se  $net_k > 0$  është ekuivalente me  $w^T x > -b_k$ . Kjo do të thotë se output-i kalon nga negativ në pozitiv kur shuma e peshuar e inpute-ve e kalon vlerën e  $-b_k$ , siç tregohet në figurën 4-9(b) [33].

**Funksioni shkallë:** Ky funksion aktivizimi është një funksion binar, ku neuroni jep daljen 0 nëse vlera neto është negative (ose zero) dhe jep daljen 1 nëse vlera neto është pozitive. (shiko figurën 4-9(c)).

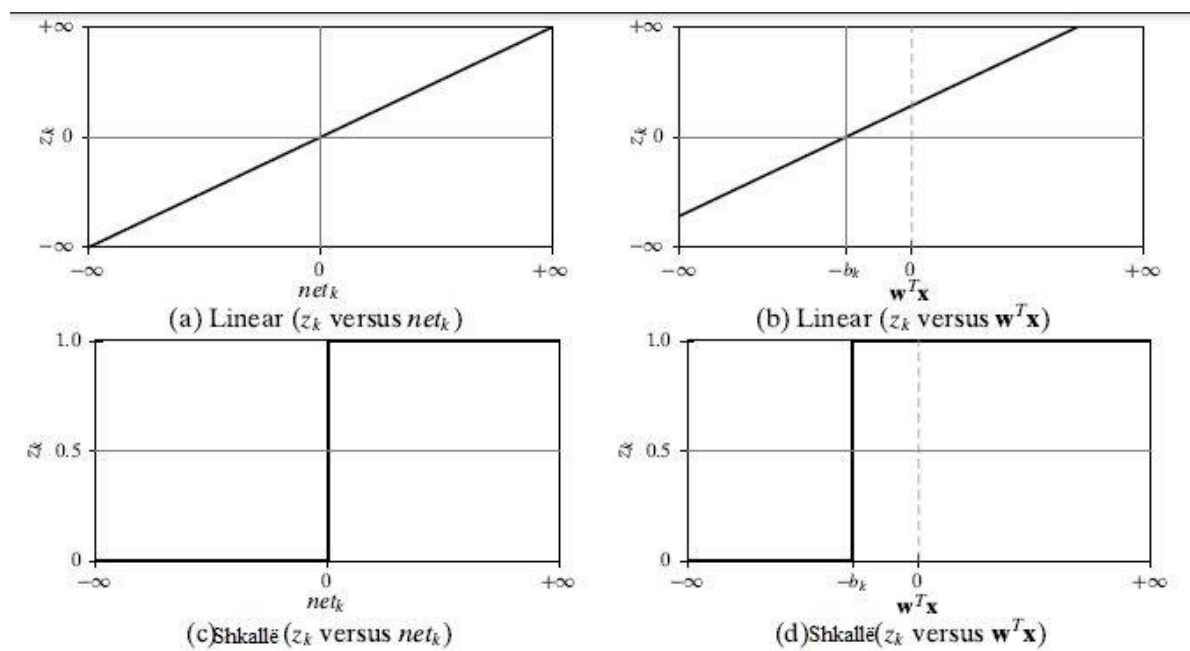
$$f(net_k) = \begin{cases} 0 & \text{nëse } net_k \leq 0 \\ 1 & \text{nëse } net_k > 0 \end{cases} \quad (4.15)$$

Është me interes të theksohet se kalimi nga 0 në 1 ndodh kur shuma e peshuar e input-eve e kalon  $-b_k$ , siç tregohet në figurën 4-9(d).

**Njësia Lineare e Korrigjuar (Rectified Linear Unit, ReLU):** Këtu neuroni qëndron inaktiv nëse hyrja neto është më e vogël ose e barabartë me zero dhe më pas rritet në mënyrë lineare me  $net_k$ , siç tregohet në figurën 4-9(e).

$$f(net_k) = \begin{cases} 0 & \text{nëse } net_k \leq 0 \\ net_k & \text{nëse } net_k > 0 \end{cases} \quad (4.16)$$

Një shprehje alternative për aktivizimin ReLU jepet si  $f(net_k) = \max\{0, net_k\}$ . Kalimi nga zero në dalje lineare ndodh kur shuma e peshuar e hyrjeve kalon vlerën  $-b_k$  (Shiko figurën 4-9(f)).





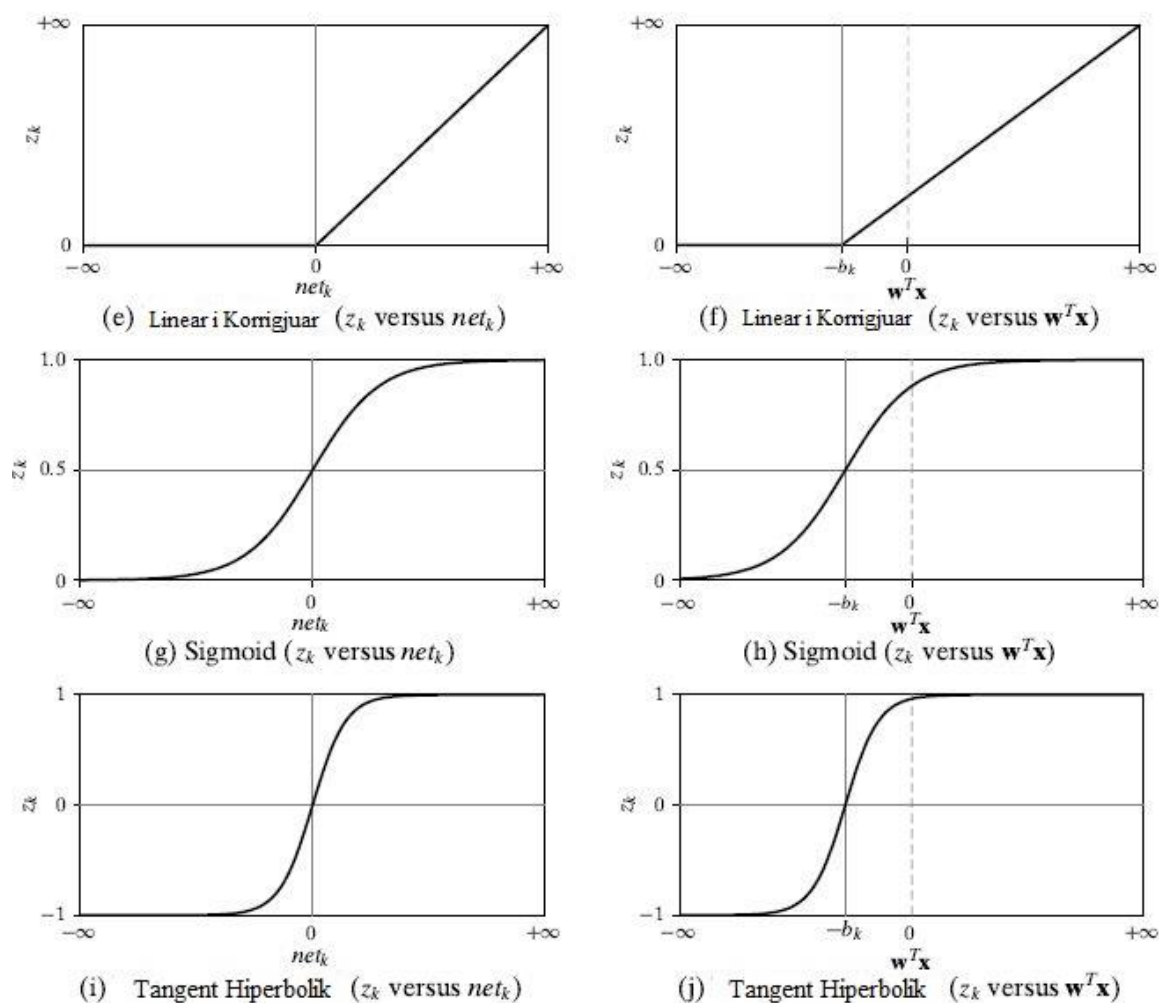


Figura 4-9 Funksonet e aktivizimit të neuroneve; ilustrimi i efektit të bias-it.

**Sigmoid:** Funksoni sigmoid, i ilustruar në figurën 4-9(g) e kufizon input-in e tij me qëllim që output-i të variojë midis 0 dhe 1.

$$f(net_k) = \frac{1}{1 + \exp\{-net_k\}} \quad (4.17)$$

Kur  $net_k = 0$ , atëherë  $f(net_k) = 0.5$ , që nënkupton se pika e kalimit ku output-i kalon vlerën 0.5 ndodh kur shuma e peshuar e input-eve kalon vlerën  $-b_k$ . (Shiko figurën 4-9(h)) [33].

**Tangenti Hiperbolik (tanh):** Tangenti hiperbolik ose funksioni tanh është i ngjashëm me funksionin sigmoid, por dalja e tij ndodhet në intervalin midis -1 dhe +1. (shiko figurën 4-9(i)).

$$f(net_k) = \frac{\exp\{net_k\} - \exp\{-net_k\}}{\exp\{net_k\} + \exp\{-net_k\}} = \frac{\exp\{2 \cdot net_k\} - 1}{\exp\{2 \cdot net_k\} + 1} \quad (4.18)$$

Kur  $net_k = 0$ , atëherë  $f(net_k) = 0$ , që do të thotë se output-i kalon nga negativ në pozitiv kur shuma e peshuar e input-eve kalon vlerën e  $-b_k$ , siç tregohet në figurën 4-9(j).

**Softmax:** Softmax është një përgjithësim i funksionit të aktivizimit sigmoid ose logjistik. Softmax përdoret kryesisht në shtresën e output-it në një rrjet neural dhe ndryshe nga funksionet e tjera, varet jo vetëm nga input-i neto në neuronin  $k$ , por edhe nga sinjali neto në të gjitha neuronet e tjera në shtresën e output-it. Kështu që, nëse jepet vektori i hyrjeve neto  $net = (net_1, net_2, \dots, net_p)^T$ , për të gjitha  $p$  neuronet dalëse, dalja e funksionit softmax për neuronin e  $k$ -të jepet si:

$$f(net_k | \mathbf{net}) = \frac{\exp\{net_k\}}{\sum_{i=1}^p \exp\{net_i\}} \quad (4.19)$$

Figura 4-10 jep paraqitjen grafike të aktivizimit softmax për  $net_k$ , kundrejt sinjalit neto  $net_j$ , me të gjitha vlerat e tjera neto të vendosura zero. Output-i sillet në mënyrë të ngjashme me një kurbë sigmoide për çdo vlerë të dhënë të  $net_j$ .

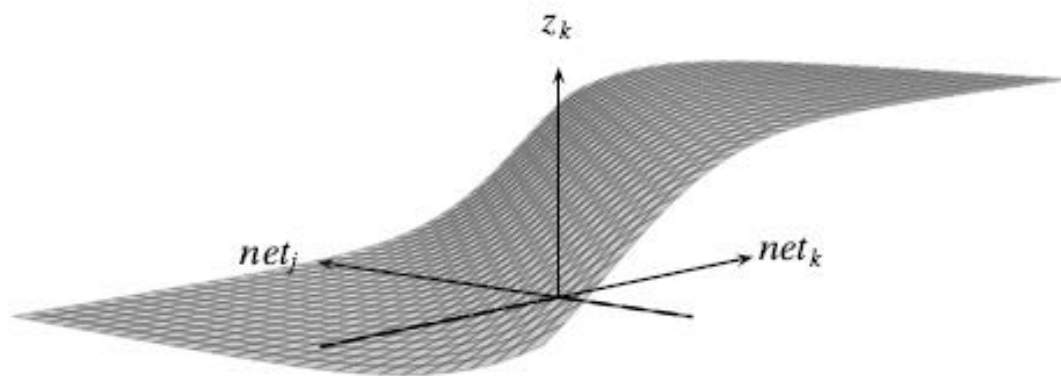


Figura 4-10 Softmax ( $net_k$  versus  $net_j$ )

#### 4.4.2 Derivatet e funksioneve të aktivizimit

Për procesin e të mësuarit ose të parashikimit duke përdorur një rrjet neural, duhet të konsiderohet derivati i funksionit të aktivizimit në lidhje me argumentin e tij. Derivatet e funksioneve të aktivizimit jepen si në vazhdim[33]:

**Identiteti/Linear:** Identiteti ose funksioni linear i aktivizimit ka derivatin 1 në lidhje me argumentin e tij duke dhënë:

$$\frac{\partial f(net_j)}{\partial net_j} = 1 \quad (4.20)$$

**Shkallë:** Funksioni shkallë ka derivatin 0 kudo, përveç pikës së këputjes në zero, ku derivati është  $\infty$ .

**ReLU:** Funksioni ReLU është i padiferencueshëm në 0, por pavarësisht kësaj për vlerat e tjera derivati i tij është 0 nëse  $net_j < 0$  dhe 1 nëse  $net_j > 0$ . Në 0, derivati mund të vendoset në çdo vlerë brenda intervalit  $[0,1]$ , ku një zgjidhje e thjeshtë mund të jetë 0. Duke i mbledhur të gjitha, derivati jepet si:

$$\frac{\partial f(net_j)}{\partial net_j} = \begin{cases} 0 & \text{nëse } net_j \leq 0 \\ 1 & \text{nëse } net_j > 0 \end{cases} \quad (4.21)$$

Megjithëse ReLU ka një pikë këputje në 0, është një zgjidhje e përhapur për trajnimin e rrjetave të thella neurale.

**Sigmoid:** Derivati i funksionit sigmoid jepet si:

$$\frac{\partial f(net_j)}{\partial net_j} = f(net_j) * (1 - f(net_j)) \quad (4.22)$$

**Tangenti Hiperbolik:** Derivati i funksionit tani jepet si:

$$\frac{\partial f(net_j)}{\partial net_j} = 1 - f(net_j)^2 \quad (4.23)$$

**Softmax:** Funksioni i aktivizimit softmax është një funksion vektorial, i cili kalon një vektor hyrjeje  $net = (net_1, net_2, \dots, net_p)^T$ , në një vektor me vlera probabilitetesh. Funksioni softmax përdoret zakonisht vetëm për shtresën e output-it. Derivati i pjesëshëm i  $f(net_j)$  në lidhje me  $net_j$  jepet si:

$$\frac{\partial f(net_j | \mathbf{net})}{\partial net_j} = f(net_j) * (1 - f(net_j))$$

Ndërsa derivati i pjesëshëm i  $f(net_j)$  në lidhje me  $net_k$ , për  $k \neq j$  jepet si:

$$\frac{\partial f(net_j | \mathbf{net})}{\partial net_k} = -f(net_k) * f(net_k)$$

Meqënëse funksioni softmax përdoret në shtresën e output-it, nëse përcaktojmë neuronin dalës të  $i$ -të si  $o_i$ , atëherë  $f(net_i) = o_i$  dhe derivati mund të shkruhet si:

$$\frac{\partial f(net_j | \mathbf{net})}{\partial net_k} = \frac{\partial o_j}{\partial net_k} = \begin{cases} o_j * (1 - o_j) & \text{nëse } k = j \\ -o_k * o_j & \text{nëse } k \neq j \end{cases} \quad (4.24)$$

#### 4.4.3 Regresi

Marrim parasysh problemin e regresit të shumëfishtë (linear), ku jepet një input  $x_i \in \mathbb{R}^d$ , qëllimi është të parashikojmë përgjigjen si më poshtë[33]:

$$\hat{y}_i = b + w_1 x_{i1} + w_2 x_{i2} + \dots + w_d x_{id} \quad (4.25)$$

Këtu,  $b$  është termi i bias dhe  $w_j$  është koeficienti ose pesha e regresit për atributin  $X_j$ . Jepet një e dhënë e trajnimit  $D$  që përfshin  $n$  pika  $x_i$  në një hapësirë  $d$ -dimensionale, së bashku me vlerat e tyre përkatëse të vërteta  $y_i$ , bias dhe peshat për regresin linear janë zgjedhur në mënyrë që të minimizojnë shumën e gabimeve në katror midis vlerës së vërtetë dhe vlerës së parashikuar mbi të gjitha pikat e të dhënave.

$$SSE = \sum_{i=1}^n (y_i - \hat{y}_i)^2 \quad (4.26)$$

Siç tregohet në Figurën 4-11 (a), një rrjet neural (neural) me  $d + 1$  neurone hyrëse  $x_0, x_1, \dots, x_d$ , përfshirë neuronin bias  $x_0 = 1$  dhe një neuron të vetëm dalës  $o$ , të gjithë me funksionet e aktivizimit të ngjashmërisë dhe me  $\hat{y} = o$ , paraqet ekzaktësisht të njëjtin model si një regres linear i shumëfishtë. Ndërsa problemi i regresit të shumëfishtë ka një zgjidhje të mbyllur, rrjetet neurale mësojnë bias dhe peshat përmes një gradienti zbritës që minimizon gabimin në katror. Rrjetet neurale mund të modelojnë po aq lehtë detyrën e regresit linear, ku kemi një vektor vlerash output  $p$ -dimensionale  $y_i \in \mathbb{R}_p$  në vend të një vlere të vetme  $y_i$ . Pra, të dhënat e trajnimit  $D$  përbëjnë  $n$  pika  $x_i \in \mathbb{R}^d$  dhe vektorët e outputit real  $y_i \in \mathbb{R}^d$ . Siç tregohet në Figurën 4-11 (b), regresi shumëvariabel mund të modelohet nga një rrjet neural me  $d + 1$  neurone hyrëse dhe  $P$  neurone dalëse  $o_1, o_2, \dots, o_p$ , me të gjitha hyrjet dhe neuronet dalëse duke përdorur funksionin e aktivizimit të ngjashmërisë.

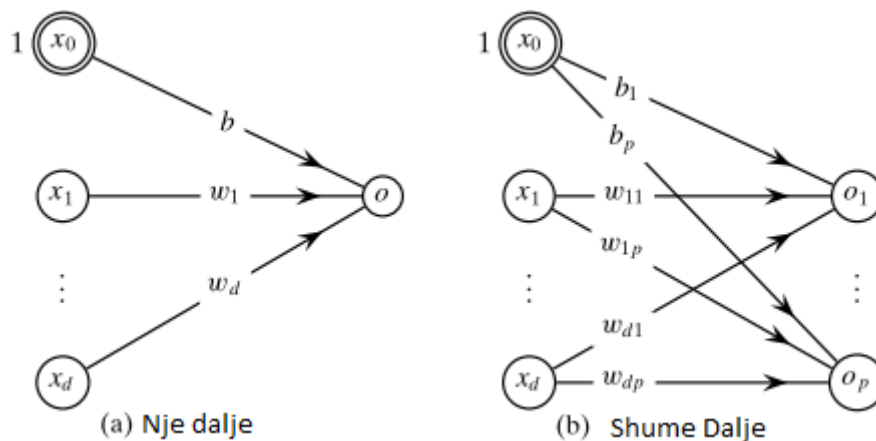


Figura 4-11 Regresi linear dhe logjistik përmes rrjetave neurale.

Një rrjet neural mëson peshat duke krahasuar outputin e tij të parashikuar  $\hat{y} = o = (o_1, o_2, \dots, o_p)^T$  me vektorin e vlerave reale  $y = (y_1, y_2, \dots, y_p)^T$ . Pra, trajnimi ndodh duke llogaritur së pari funksionin e gabimit ose funksionin e humbjes ndërmjet  $o$  dhe  $y$ . Kujtojmë që një funksion humbje cakton rezultatin ose ndëshkimin për parashikimin e daljes që të jetë  $o$  kur rezultati i dëshiruar është  $y$ . Kur parashikimi përputhet me prodhimin e vërtetë humbja duhet të jetë zero. Funksioni më i zakonshëm i humbjes për regres është funksioni i gabimit katror ku  $\epsilon_x$  tregon gabimin në hyrjen  $x$ . Në të gjitha pikat në një set të të dhënave, shuma totale e gabimeve në katror është

$$\mathcal{E} = \sum_{i=1}^n \epsilon_{x_i} = \frac{1}{2} \cdot \sum_{i=1}^n \|\mathbf{y}_i - \mathbf{o}_i\|^2 \quad (4.27)$$

#### 4.4.4 Klasifikimi

Rrjetet e neuroneve artificiale gjithashtu mund të mësojnë të klasifikojnë hyrjet. Merrni parasysh problemin e klasifikimit binar, ku  $y = 1$  tregon se pika i përket klasës pozitive dhe  $y = 0$  do të thotë që i përket klasës negative. Kujtojmë që në regresin logjistik, ne modelojmë probabilitetin e klasës pozitive përmes funksionit logjistik (ose sigmoid) [33]:

$$\pi(\mathbf{x}) = P(y = 1 | \mathbf{x}) = \frac{1}{1 + \exp\{-(b + \mathbf{w}^T \mathbf{x})\}} \quad (4.28)$$

ku  $b$  është termi bias dhe  $w = (w_1, w_2, \dots, w_d)^T$  është vektori i peshave të vlerësuara ose koeficientët e regresit.

Në anën tjetër

$$P(y=0|x=0) = 1 - P(y=0|x) = 1 - \pi(x) \quad (4.29)$$

Një ndryshim i thjeshtë në rrjetin neural të treguar në Figurën 4-11 (a) e lejon atë të zgjidhet problemi i regresit logjistik. Ajo çfarë duhet të bëjmë është të përdorim një funksion të aktivizimit sigmoid në neuronin dalës  $o$  dhe përdorimin e gabimit cross-entropisë në vend të gabimit katror. Jepet hyrja  $x$ , vlera reale  $y$  dhe vlera e parashikuar  $o$ , kujtojmë se gabimi i cross-entropisë përcaktohet si

$$\mathcal{E}_x = -(y \cdot \ln(o) + (1 - y) \cdot \ln(1 - o)) \quad (4.30)$$

Kështu, me aktivizimin sigmoid, dalja e rrjetit neural në Figurën 4-11 (a) është dhënë si

$$o = f(\text{net}_o) = \text{sigmoid}(b + \mathbf{w}^T \mathbf{x}) = \frac{1}{1 + \exp\{-(b + \mathbf{w}^T \mathbf{x})\}} = \pi(\mathbf{x}) \quad (4.31)$$

i cili është i njëjtë me modelin e regresit logjistik.

**Regresi logjistik shumëklasor**[33]: Në një mënyrë të ngjashme, arkitektura e rrjetit neural dalës të shumëfishtë e treguar në Figurën 4-11. (b) mund të përdoret për regres logjistik shumëklasor ose nominal. Për problemin e klasifikimit të përgjithshëm me klasat  $K \{c_1, c_2, \dots, c_K\}$ , vlera reale  $y$  është kodifikuar si një vektor “one-hot”. Kështu, klasa  $c_1$  kodohet si  $e_1 = (1, 0, \dots, 0)^T$ , klasa  $c_2$  kodohet si  $e_2 = (0, 1, \dots, 0)^T$ , dhe kështu me radhë, me  $e_i \in \{0, 1\}^K$  për  $i = 1, 2, \dots, K$ . Kështu, ne e kodifikojmë  $y$ -n si një vektor me shumë-ndryshore  $y \in \{e_1, e_2, \dots, e_K\}$ . Kujtojmë që në regresin logjistik shumëklasor detyra është të vlerësohet vektori bias  $b_i$  dhe pesha me klasë  $w_i \in \mathbb{R}^d$ , me klasën e fundit  $c_K$  të përdorur si klasë bazë me bias fiks  $b_K = 0$  dhe vektor të peshës fikse  $w_K = (0, 0, \dots, 0)^T \in \mathbb{R}^d$ . Vektori probabilitar në të gjitha klasat  $K$  modelohet përmes funksionit softmax.

$$\pi_i(\mathbf{x}) = \frac{\exp\{b_i + \mathbf{w}_i^T \mathbf{x}\}}{\sum_{j=1}^K \exp\{b_j + \mathbf{w}_j^T \mathbf{x}\}} \quad (4.32)$$

për të gjithë  $i = 1, 2, \dots, K$

Prandaj, rrjeti neural i treguar në Figurën 4-11 (b) (me  $p = K$ ) mund të zgjidhë detyrë e regresit logjistik shumëklasor, me kusht që të përdorim një aktivizim softmax në dalje dhe të përdoret gabimi i ndër-entropisë në rrugën  $K$  të përcaktuar si

$$\mathcal{E}_x = -(y_1 \cdot \ln(o_1) + \dots + y_K \cdot \ln(o_K)) \quad (4.33)$$

ku  $x$  është një vektor hyrës,  $o = (o_1, o_2, \dots, o_K)^T$  është vektori i përgjigjes së parashikuar, dhe  $y = (y_1, y_2, \dots, y_K)^T$  është vektori i vërtetë i përgjigjes. Vini re se vetëm një element i  $y$  është 1, dhe pjesa tjetër janë 0, për shkak të kodimit “one-hot”. Me aktivizimin e softmax, dalja e rrjetit neural në Figurën 4-11 (b) (me  $p = K$ ) është dhënë si

$$o_i = P(\mathbf{y} = \mathbf{e}_i | \mathbf{x}) = f(\text{net}_i | \text{net}) = \frac{\exp\{\text{net}_i\}}{\sum_{j=1}^p \exp\{\text{net}_j\}} = \pi_i(\mathbf{x}) \quad (4.34)$$

që përputhet me detyrën e regresit logjistik shumëklasëshe. I vetmi kufizim që duhet të bëjmë në rrjetin neural është se peshat në skajet në neuronin e fundit të daljes duhet të jetë zero për të modeluar peshat e klasës bazë  $w_K$ . Sidoqoftë, në praktikë, ne mund të reduktojmë këtë kufizim, dhe thjesht të mësohet një vektor i rregullt i peshës për klasën  $c_K$ .

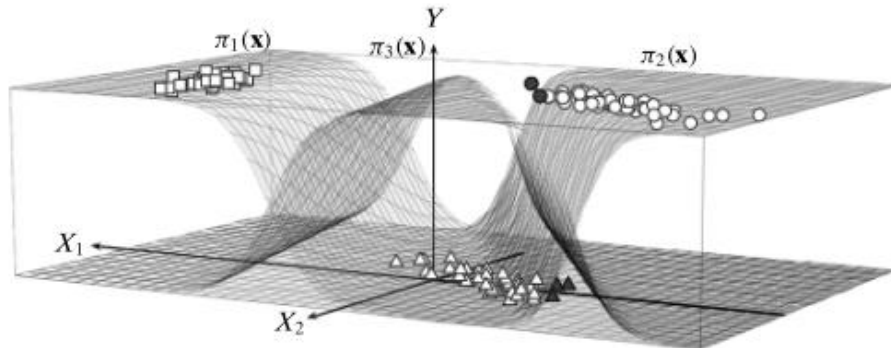


Figura 4-12 Rrjetet neurale për regresin logjistik multiklasor: Të dhënat e përbërësve kryesorë të Iris. Pikat e multiklasifikimit janë treguar në ngjyrë gri të errët. Pikat në klasën  $c_1$  dhe  $c_2$  tregohen të zhvendosura në lidhje me klasën bazë  $c_3$  vetëm për ilustrim.

#### 4.4.5 Funkzionet e gabimit

Zakonisht, për një detyrë regresi, ne përdorim gabimin në katror si funksionin e humbjes, ndërsa për klasifikim, ne përdorim funksionin e humbjes ndër-entropik. Për më tepër, nga përdorimi i rrjetave neurale, ne do të kërkojmë derivatet e pjesshëm të funksionit të gabimit përkundrejt neuroneve dalëse. Kështu, funksionet e gabimit të përdorura zakonisht dhe të derivateve të tyre renditen më poshtë[33]:

**Gabim në katror:** Jepet një vektor hyrës  $x \in \mathbb{R}^d$ , funksioni i humbjes në katror të gabimit mat devijimin në katror midis vektorit të parashikuar të daljes  $o \in \mathbb{R}^p$  dhe së vlerave reale  $y \in \mathbb{R}^p$ , të përcaktuar si më poshtë:

$$\mathcal{E}_x = \frac{1}{2} \| \mathbf{y} - \mathbf{o} \|^2 = \frac{1}{2} \sum_{j=1}^p (y_j - o_j)^2 \quad (4.35)$$

Ku  $\mathcal{E}_x$  tregon gabimin në hyrjen  $x$ . Derivati i pjesshëm i funksionit të gabimit në katror në lidhje me një neuron të veçantë të prodhimit të është

$$\frac{\partial \mathcal{E}_x}{\partial o_j} = \frac{1}{2} \cdot 2 \cdot (y_j - o_j) \cdot -1 = o_j - y_j \quad (4.36)$$

Në të gjithë neuronet e daljes, ne mund ta shkruajmë këtë si

$$\partial \mathcal{E}_x / \partial \mathbf{o} = \mathbf{o} - \mathbf{y} \quad (4.37)$$

#### Gabimi cross-entropy:

Për detyrat e klasifikimit, me  $K$  klasat  $\{c_1, c_2, \dots, c_K\}$ , ne zakonisht vendosim numrin e neuroneve dalëse  $p = K$ , me një neuron dalës për klasë. Për më tepër, secila prej klasave është e koduar si një vector "one-hot", me klasën e koduar si vektori bazik standard  $\mathbf{e}_i = (e_{i1}, e_{i2}, \dots,$

$e_{iK})^T \in \{0, 1\}^K$ , me  $e_{ii} = 1$  dhe  $e_{ij} = 0$  për të gjithë  $j$  e ndryshme nga  $i$ . Kështu, jepet hyrja  $x \in \mathbb{R}^d$ , me përgjigjen e vërtetë  $y = (y_1, y_2, \dots, y_K)^T$ , ku  $y \in \{e_1, e_2, \dots, e_K\}$ , humbja ndër-entropike përcaktohet si

$$\mathcal{E}_x = -\sum_{i=1}^K y_i \cdot \ln(o_i) = -(y_1 \cdot \ln(o_1) + \dots + y_K \cdot \ln(o_K)) \quad (4.38)$$

Mbajmë në konsideratë që vetëm një element i  $y$  është 1 dhe pjesa tjetër janë 0 për shkak të kodimit "one-hot". Pra, nëse  $y = e_i$ , atëherë vetëm  $y_i = 1$ , dhe elementët e tjerë  $y_j = 0$  për  $j$  të ndryshme nga  $i$ . Derivati i pjesshëm i funksionit të gabimit ndër-entropik në lidhje me një neuron të vetëm dalës  $o_j$  është

$$\partial \mathcal{E}_x / \partial o_j = -y_j / o_j$$

Vektori i derivateve të pjesshëm të funksionit të gabimit në lidhje me neuronet dalëse jepet si

$$\frac{\partial \mathcal{E}_x}{\partial o} = \left( \frac{\partial \mathcal{E}_x}{\partial o_1}, \frac{\partial \mathcal{E}_x}{\partial o_2}, \dots, \frac{\partial \mathcal{E}_x}{\partial o_K} \right)^T = \left( -\frac{y_1}{o_1}, -\frac{y_2}{o_2}, \dots, -\frac{y_K}{o_K} \right)^T \quad (4.39)$$

**Gabimi cross-entropy binar:** Për detyrat e klasifikimit me klasa binare, është tipike për të koduar klasën pozitive si 1 dhe klasën negative si 0, në krahasim me përdorimin e një kodimi "one-hot" si në rastin e përgjithshëm të klasës  $K$ . Jepet një hyrje  $x \in \mathbb{R}^d$ , me vlerën reale  $y \in \{0, 1\}$ , ekziston vetëm një neuron dalës  $o$ . Prandaj, gabimi ndër-entropik binar përcaktohet si [33]:

$$\mathcal{E}_x = -(y \cdot \ln(o) + (1 - y) \cdot \ln(1 - o)) \quad (4.40)$$

Këtu  $y$  është ose 1 ose 0. Derivati i pjesshëm i funksionit të gabimit binar të ndër-entropisë në lidhje me neuronin dalës  $o$  është si më poshtë

$$\begin{aligned} \frac{\partial \mathcal{E}_x}{\partial o} &= \frac{\partial}{\partial o} \{-y \cdot \ln(o) - (1 - y) \cdot \ln(1 - o)\} \\ &= -\left( \frac{y}{o} + \frac{1-y}{1-o} \cdot -1 \right) = \frac{-y \cdot (1-o) + (1-y) \cdot o}{o \cdot (1-o)} \\ &= \frac{o-y}{o \cdot (1-o)} \end{aligned} \quad (4.41)$$

#### 4.4.6 Trajnimi i rrjetave neurale me një shtresë të ndërmjetme

Algoritmi 1 [33]: tregon pseudo-kodin për të mësuar peshat duke marrë parasysh të gjitha pikat hyrëse në  $D$  përmes zbritjes së gradientit stokastik. Kodi tregohet për një MLP me një shtresë të vetme të fshehur, duke përdorur një funksion gabimi në katror dhe aktivizimet sigmoidale për të gjitha neuronet e fshehura dhe dalëse. Qasja quhet zbritje e gradientit stokastik pasi që ne llogarisim gradientët e peshës dhe bias pasi kemi vëzhguar secilën pikë trajnimi (në mënyrë të rastësishme).

```

MLP-TRAINING (D,  $m$ ,  $\eta$ , maxiter):
// Initialize bias vectors
1  $\mathbf{b}_h \leftarrow$  random  $m$ -dimensional vector with small values
2  $\mathbf{b}_o \leftarrow$  random  $p$ -dimensional vector with small values
// Initialize weight matrices
3  $\mathbf{W}_h \leftarrow$  random  $d \times m$  matrix with small values
4  $\mathbf{W}_o \leftarrow$  random  $m \times p$  matrix with small values
5  $t \leftarrow 0$  // iteration counter
6 repeat
7   foreach  $(\mathbf{x}_i, \mathbf{y}_i) \in \mathbf{D}$  in random order do
// Feed-forward phase
8      $\mathbf{z}_i \leftarrow f(\mathbf{b}_h + \mathbf{W}_h^T \mathbf{x}_i)$ 
9      $\mathbf{o}_i \leftarrow f(\mathbf{b}_o + \mathbf{W}_o^T \mathbf{z}_i)$ 
// Backpropagation phase: net gradients
10     $\delta_o \leftarrow \mathbf{o}_i \odot (\mathbf{1} - \mathbf{o}_i) \odot (\mathbf{o}_i - \mathbf{y}_i)$ 
11     $\delta_h \leftarrow \mathbf{z}_i \odot (\mathbf{1} - \mathbf{z}_i) \odot (\mathbf{W}_o \cdot \delta_o)$ 
// Gradient descent for bias vectors
12     $\nabla_{\mathbf{b}_o} \leftarrow \delta_o$ ;  $\mathbf{b}_o \leftarrow \mathbf{b}_o - \eta \cdot \nabla_{\mathbf{b}_o}$ 
13     $\nabla_{\mathbf{b}_h} \leftarrow \delta_h$ ;  $\mathbf{b}_h \leftarrow \mathbf{b}_h - \eta \cdot \nabla_{\mathbf{b}_h}$ 
// Gradient descent for weight matrices
14     $\nabla_{\mathbf{W}_o} \leftarrow \mathbf{z}_i \cdot \delta_o^T$ ;  $\mathbf{W}_o \leftarrow \mathbf{W}_o - \eta \cdot \nabla_{\mathbf{W}_o}$ 
15     $\nabla_{\mathbf{W}_h} \leftarrow \mathbf{x}_i \cdot \delta_h^T$ ;  $\mathbf{W}_h \leftarrow \mathbf{W}_h - \eta \cdot \nabla_{\mathbf{W}_h}$ 
16     $t \leftarrow t + 1$ 
17 until  $t \geq \text{maxiter}$ 

```

Algoritmi 1: Trajnimi i rrjetave neurale me një shtresë të ndërmjetme [33]

Algoritmi MLP merr si input databazën  $\mathbf{D}$  (me pikat  $\mathbf{x}_i$  dhe përgjigjet e dëshiruara  $\mathbf{y}_i$  për  $i = 1, 2, \dots, n$ ), numrin e neuroneve të shtresës së fshehur  $\mathbf{m}$ , shkallën e të mësuarit  $\eta$ , dhe një maksimum të pragut të plotë që specifikon numrin maksimal të përsëritjeve. Madhësia e shtresave të hyrjes ( $d$ ) dhe daljes ( $p$ ) përcaktohet automatikisht nga  $\mathbf{D}$ . MLP fillimisht inicializon hyrjen  $d \times m$  në matricën e peshës së fshehur të shtresës  $\mathbf{W}_h$ , dhe  $m \times p$  të fshehur në matricën e shtresës së prodhimit  $\mathbf{W}_o$  në vlera të vogla, për shembull, në mënyrë të njëtrajtshme random në intervalin  $[-0.01, 0.01]$ . Është e rëndësishme të theksohet se peshat nuk duhet të vendosen në 0, përndryshe, të gjithë neuronet e fshehura do të jenë identike në vlerat e tyre, dhe kështu do të jenë edhe neuronet dalëse.

Trajnimi i MLP-së merr iteracione të shumta mbi pikat hyrëse. Për secilin input  $\mathbf{x}_i$ , MLP llogarit vektorin e daljes  $\mathbf{o}_i$  përmes hapit feed-forward. Në fazën backpropagation, ne llogarisim vektorin e gradientit të gabimit  $\delta_o$  në lidhje me rrjetën në neuronet e daljes, të ndjekur nga dh për neuronet e fshehura. Në hapin e zbritjes së gradientit stokastik, ne llogarisim gradientët e gabimit në lidhje me peshat dhe bias, të cilat përdoren për të përditësuar matricat e peshës dhe vektorët bias. Kështu, për çdo vektor hyrës  $\mathbf{x}_i$ , të gjitha peshat dhe bias përditësohen bazuar në gabimin e ndodhur midis daljes së parashikuar të  $\mathbf{o}_i$  dhe përgjigjes së vërtetë  $\mathbf{y}_i$ . Pasi të jetë përpunuar çdo input, ai përfundon një përsëritje të trajnimit, të quajtur një epokë. Trajnimi ndalet kur të arrihet numri maksimal i përsëritjeve, maksimumi. Nga ana tjetër, gjatë testimit, për çdo hyrje  $\mathbf{x}$ , ne aplikojmë hapat feed-forward dhe shtypim output-in e parashikuar  $\mathbf{o}$ .

Për sa i përket kompleksitetit të llogaritjes, çdo përsëritje e algoritmit të trajnimit MLP merr kohë  $O(dm + mp)$  për fazën feed-forward,  $p + mp + m = O(mp)$  kohë për përhapjen backpropagation të gradientëve të gabimit dhe  $O(dm + mp)$  kohë për përditësimin e matricave të peshave dhe vektorëve bias. Kështu, koha totale e trajnimit për përsëritje është  $O(dm + mp)$ .

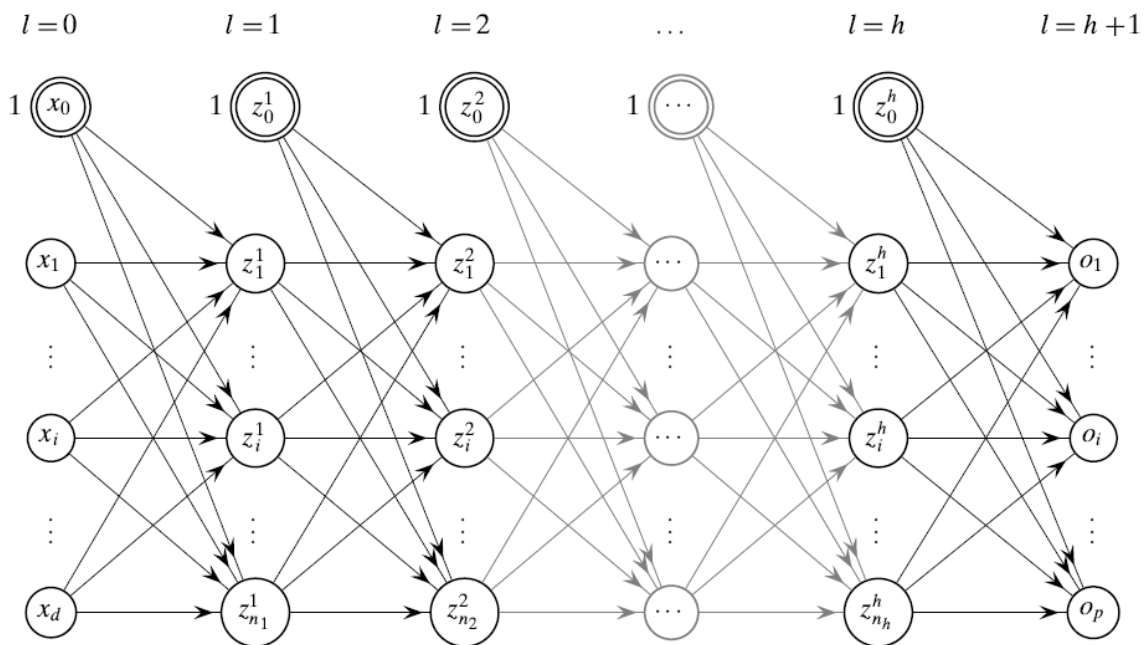
Përgjithësojmë kështu hapat feed-forward dhe backpropagation për shumë shtresa të fshehura, si dhe gabimet arbitrare dhe funksionet e aktivizimit të neuroneve.



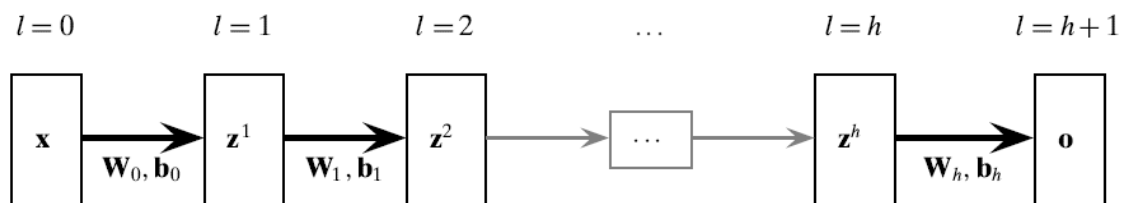
Marrim në konsideratë një MLP me  $h$  shtresa të fshehura siç tregohet në Figurën 4-11. Supozojmë se hyrja në MLP përfshin  $n$  pika  $x_i \in \mathbb{R}^d$  me vektorin përkatës të vlerave reale  $y_i \in \mathbb{R}^p$ . Ne shënojmë neuronet hyrëse si shtresa  $l = 0$ , shtresa e parë e fshehur si  $l = 1$ , shtresa e fundit e fshehur si  $l = h$ , dhe së fundmi shtresa e daljes si shtresa  $l = h + 1$ . Përdorim  $n_l$  për të treguar numrin e neuroneve në shtresën  $l$ . Meqenëse pikat hyrëse janë  $d$ -dimensionale, kjo nënkupton  $n_0 = d$  dhe meqenëse vektori i përgjigjes së vërtetë është  $p$ -dimensional, kemi  $n_{h+1} = p$ . Shtresat e fshehura kanë  $n_1$  neurone për shtresën e parë të fshehur,  $n_2$  për shtresën e dytë dhe  $n_h$  për shtresën e fundit të fshehur. Vektori i vlerave të neuroneve për shtresën  $l$  (për  $l = 0, \dots, h + 1$ ) shënohet si:

$$\mathbf{z}^l = (z_1^l, \dots, z_{n_l}^l)^T$$

Secila shtresë përveç shtresës së daljes ka një neuron shtesë të bias, i cili është neuroni në indeksin 0. Kështu, neuroni bias për shtresën  $l$  shënohet  $z_0^l$  dhe vlera e tij fiksohet në  $z_0^l = 1$ .



(a) Pamja e detajeve



(a) Pamja e shtresës

Figura 4-13 Perceptron i thellë me shumë shtresa, me  $h$  shtresa të fshehura.

Figura 4-13 (a) tregon një pamje të hollësishme të një MLP me h shtresa të fshehura, duke treguar neuronet individuale në secilën shtresë duke përfshirë edhe neuronin bias. Vini re se vektori i vlerave të neuroneve hyrëse është shkruar gjithashtu si:

$$x = (x_1, x_2, \dots, x_d)^T = (z_1^0, z_2^0, \dots, z_d^0)^T = z^0$$

dhe vektori i vlerave të neuroneve dalëse shënohet gjithashtu si:

$$o = (o_1, o_2, \dots, o_p)^T = (z_1^{h+1}, z_2^{h+1}, \dots, z_d^{h+1})^T = z^{h+1}$$

Matrica e peshës midis neuroneve të shtresës  $l$  dhe shtresës  $l + 1$  shënohet  $W_l \in \mathbb{R}^{n_l \times n_{l+1}}$  dhe vektori i termave të bias nga neuroni i bias  $z_0^l$  tek neuronet në shtresën  $l + 1$  shënohet  $b_l \in \mathbb{R}^{n_{l+1}}$ , për  $l = 0, 1, \dots, h$ . Kështu,  $W_0 \in \mathbb{R}^{d \times n_1}$  është matrica e peshës ndërmjet hyrjes dhe shtresës së parë të fshehur,  $W_l \in \mathbb{R}^{n_l \times n_{l+1}}$  është matrica e peshës midis shtresës së parë dhe të dytë të fshehur, e kështu me radhë; së fundmi,  $W_h \in \mathbb{R}^{n_h \times p}$  është matrica e peshës midis shtresës së fundit të fshehur dhe shtresës së daljes. Për vektorët e bias,  $b_0 \in \mathbb{R}^{n_1}$  specifikon bias për neuronet në shtresën e parë të fshehur,  $b_1 \in \mathbb{R}^{n_2}$  bias për neuronet në shtresën e dytë të fshehur, etj. Kështu,  $b_h \in \mathbb{R}^p$  specifikon bias për neuronet e daljes. Figura 4-11 (b) tregon një pamje të shtresave për një MLP me h shtresa të fshehura. Ky është një paraqitje më kompakte që specifikon qartë arkitekturën ose topologjinë e MLP. Secila shtresë  $l$  përfaqësohet si një nyje drejtkëndëshe dhe shënohet me vektorin e vlerave të neuroneve,  $z^l$ . Neuronet bias nuk tregohen, por supozohet se janë të pranishme në secilën shtresë përveç daljes. Një skaj midis shtresës  $l$  dhe  $l + 1$  është etiketuar me matricën e peshës  $W_l$  dhe vektorin e bias  $b_l$  që specifikojnë parametrat midis atyre shtresave. Për trajnimin e MLP-ve të thella, ne do t'i referohemi disa vektorëve të pjesshëm të derivateve, të përshkruara më poshtë. Përcaktoni  $\delta_i^l$  si gradient neto, d.m.th., derivati i pjesshëm i funksionit të gabimit në lidhje me vlerën neto në  $z_i^l$ .

$$\delta_i^l = \frac{\partial \mathcal{E}_x}{\partial \text{net}_i} \quad (4.42)$$

dhe me  $\delta^l$  shënojmë vektorin e gradientit neto në shtresën  $l$ , për  $l = 1, 2, \dots, h + 1$ .

$$\delta^l = (\delta_1^l, \dots, \delta_{n_l}^l)^T \quad (4.43)$$

Le të tregojë  $f^l$  funksionin e aktivizimit për shtresën  $l$ , për  $l = 0, 1, \dots, h + 1$ , dhe më tej le të tregojë  $\partial f^l$  vektorin e derivateve të funksionit të aktivizimit në lidhje me  $\text{net}_i$  për të gjitha neuronet  $z_i^l$  në shtresa  $l$ :

$$\partial f_l = \left( \frac{\partial f^l(\text{net}_1)}{\partial \text{net}_1}, \dots, \frac{\partial f^l(\text{net}_{n_1})}{\partial \text{net}_{n_1}} \right)^T \quad (4.44)$$

Le të shënojë  $\partial \mathcal{E}_x$  vektorin e derivateve të pjesshëm të funksionit të gabimit në lidhje me vlerat  $o_i$  për të gjitha neuronet dalëse:

$$\partial \mathcal{E}_x = \left( \frac{\partial \mathcal{E}_x}{\partial o_1}, \frac{\partial \mathcal{E}_x}{\partial o_2}, \dots, \frac{\partial \mathcal{E}_x}{\partial o_p} \right)^T \quad (4.45)$$

#### 4.4.6.1 Faza Feed-Forward

Në mënyrë tipike në një MLP të thellë, i njëjti funksion aktivizimi  $f^l$  përdoret për të gjitha neuronet në një shtresë të caktuar  $l$ . Shtresa e hyrjes gjithmonë përdor aktivizimin e identitetit, kështu që  $f^0$  është funksioni i identitetit [33]. Gjithashtu, të gjithë neuronet e njëanshmërisë përdorin funksionin e identitetit me një vlerë fikse prej 1. Shtresa e daljes zakonisht përdor aktivizime sigmoid ose softmax për detyrat e klasifikimit, ose aktivizime të identitetit për detyrat e regresit. Shtresat e fshehura zakonisht përdorin aktivizime sigmoid, tanh ose ReLU. Në diskutimin tonë, ne supozojmë funksionin fiks të aktivizimit  $f^l$  për të gjithë neuronet në një shtresë të caktuar. Sidoqoftë, është e lehtë të përgjithësohet në një funksion specifik të aktivizimit të neuroneve  $f^l$  i për neuronin  $z_i^l$  në shtresën  $l$ .

Për një palë të dhënë hyrëse  $(x, y) \in D$ , MLP e thellë llogarit vektorin e daljes përmes procesit feed-forward:

$$\begin{aligned} o &= f^{h+1}(b_h + W_h^T \cdot z^h) = f^{h+1}\left(b_h + W_h^T \cdot f^h(b_{h-1} + W_{h-1}^T \cdot z^{h-1})\right) \\ &=: \\ &= f^{h+1}\left(b_h + W_h^T \cdot f^h\left(b_{h-1} + W_{h-1}^T \cdot f^{h-1}\left(\dots f^2(b_1 + W_1^T \cdot f^1(b_0 + W_0^T \cdot x))\right)\right)\right) \end{aligned} \quad (4.46)$$

Vini re se çdo  $f^l$  shpërndan mbi argumentin e tij. Kjo është:

$$f^l(b_{l-1} + W_{l-1}^T \cdot x) = \left(f^l(net_1), f^l(net_2), \dots, f^l(net_{n_1})\right)^T \quad (4.47)$$

#### 4.4.6.2 Faza Backpropagation

Konsiderojmë përditësimin e peshës midis një shtrese të caktuar dhe një tjetre, duke përfshirë midis hyrjes dhe shtresës së fshehur, ose midis dy shtresave të fshehura, ose midis shtresës së fundit të fshehur dhe shtresës së daljes. Le të jetë  $z_i^l$  një neuron në shtresën  $l$ , dhe  $z_j^{l+1}$ , një neuron në shtresën tjetër  $l + 1$ . Le të jetë  $w_{ij}^l$ , pesha midis  $z_i^l$ , dhe  $z_j^{l+1}$ , dhe  $b_j^l$  le të tregojë termin bias midis  $z_0^l$ , dhe  $z_j^{l+1}$ . Pesha dhe bias përditësohen duke përdorur qasjen e zbritjes gradiente[33]:

$$w_{ij}^l = w_{ij}^l - \eta \nabla_{w_{ij}^l} \quad b_j^l = b_j^l - \eta \nabla_{b_j^l}$$

Ku,  $\nabla_{w_{ij}^l}$  është gradienti i peshës dhe  $\nabla_{b_j^l}$  është gradienti i bias, d.m.th., derivati i pjesshëm i funksionit të gabimit në lidhje me peshën dhe bias, përkatësisht:

$$\nabla_{w_{ij}^l} = \frac{\partial \mathcal{E}_x}{\partial w_{ij}^l} \quad \nabla_{b_j^l} = \frac{\partial \mathcal{E}_x}{\partial b_j^l}$$

Siç u përmend më herët, ne mund të përdorim rregullin e zinxhirit për të shkruar gradientin e peshës dhe bias, si më poshtë:

$$\nabla_{w_{ij}^l} = \frac{\partial \mathcal{E}_x}{\partial w_{ij}^l} = \frac{\partial \mathcal{E}_x}{\partial net_j} \cdot \frac{\partial net_j}{\partial w_{ij}^l} = \delta_j^{l+1} z_i^l = z_i^l \cdot \delta_j^{l+1}$$

$$\nabla_{b_j^l} = \frac{\partial \mathcal{E}_x}{\partial b_j^l} = \frac{\partial \mathcal{E}_x}{\partial net_j} \cdot \frac{\partial net_j}{\partial b_j^l} = \delta_j^{l+1}$$

ku  $\delta_j^{l+1}$  është gradienti neto .d.m.th., derivati i pjesshëm i funksionit të gabimit në lidhje me vlerën neto në  $z_i^{l+1}$ , dhe ne kemi

$$\frac{\partial net_j}{\partial w_{ij}^l} = \frac{\partial}{\partial w_{ij}^l} \left\{ b_j^l + \sum_{k=0}^{n_l} w_{ij}^l \cdot z_k^l \right\} = z_i^l \quad \frac{\partial net_j}{\partial b_j^l} = \frac{\partial}{\partial b_j^l} \left\{ b_j^l + \sum_{k=0}^{n_l} w_{ij}^l \cdot z_k^l \right\} = 1$$

Duke pasur parasysh vektorin e vlerave të neuroneve në shtresën l, përkatësisht  $z^l = (z_1^l, \dots, z_{n_l}^l)^T$ , ne mund të llogarisim të gjithë matricën e gradientit të peshës përmes një operacioni të produktit të jashtëm:

$$\nabla w_l = z^l \cdot (\delta^{l+1})^T$$

dhe vektori i gradientit bias si:

$$\nabla b_l = \delta^{l+1}$$

me  $l = 0, 1, \dots, h$ .  $\delta^{l+1}$  është vektori gradient neto në shtresën l + 1. Kjo gjithashtu na lejon të përditësojmë të gjitha peshat dhe biaset si më poshtë:

$$W_l = W_l - \eta \cdot \nabla W_l \quad (4.48)$$

$$b_l = b_l - \eta \cdot \nabla b_l$$

ku  $\eta$  është madhësia e hapit. Sidoqoftë, vërejmë se për të llogaritur gradientët e peshës dhe anësisë për shtresën l duhet të llogarisim gradientët neto  $\delta_{l+1}$  në shtresën l + 1.

#### 4.4.6.3 Gradientët në shtresën e daljes

Le të shqyrtojmë mënyrën e llogaritjes së gradientëve neto në shtresën e daljes h + 1. Nëse të gjithë neuronet dalëse janë të pavarur (për shembull, kur përdorni aktivizime lineare ose sigmoidale), gradienti neto merret duke diferencuar funksionin e gabimit në lidhje me sinjalin neto në neuronet dalëse[33].

$$\delta_j^{h+1} = \frac{\partial \mathcal{E}_x}{\partial net_j} = \frac{\partial \mathcal{E}_x}{\partial f^{h+1}(net_j)} \cdot \frac{\partial f^{h+1}(net_j)}{\partial net_j} = \frac{\partial \mathcal{E}_x}{\partial o_j} \frac{\partial f^{h+1}(net_j)}{\partial net_j}$$

Kështu, gradienti varet nga dy terma, derivati i pjesshëm i funksionit të gabimit në lidhje me vlerën e neuronit dalës, dhe derivati i funksionit të aktivizimit në lidhje me argumentin e tij. Vektori i gradientit neto në të gjithë neuronet dalëse është dhënë si :

$$\delta^{h+1} = \partial f^{h+1} \odot \partial \mathcal{E}_x \quad (4.49)$$

ku  $\odot$  është elementi ose produkti Hadamard,  $\partial f^{h+1}$  është vektori i derivateve të funksionit të aktivizimit në lidhje me argumentin e tij në shtresën e daljes  $l = h + 1$  dhe  $\partial \mathcal{E}_x$  është vektori i derivateve të gabimit në lidhje me vlerat e neuronit dalës .

Nga ana tjetër, nëse neuronet e daljes nuk janë të pavarur (për shembull, kur përdorni një aktivizim të softmax), atëherë duhet të modifikojmë llogaritjen e gradientit neto në secilin neuron dalës si më poshtë:

$$\delta_j^{h+1} = \frac{\partial \mathcal{E}_x}{\partial net_j} = \sum_{i=1}^p \frac{\partial \mathcal{E}_x}{\partial f^{h+1}(net_j)} \cdot \frac{\partial f^{h+1}(net_j)}{\partial net_j}$$

Në të gjithë neuronet dalëse, ne mund ta shkruajmë këtë në mënyrë kompakte si më poshtë:

$$\delta^{h+1} = \partial F^{h+1} \cdot \partial \mathcal{E}_x \quad (4.50)$$

ku  $\partial F^{h+1}$  është matrica e derivateve të  $o_i = f^{h+1}(net_i)$  në lidhje me  $net_j$  për të gjithë  $i, j = 1, 2, \dots, p$ , dhënë si:

$$\partial F^{h+1} = \begin{pmatrix} \frac{\partial o_1}{\partial net_1} & \frac{\partial o_1}{\partial net_2} & \dots & \frac{\partial o_1}{\partial net_p} \\ \frac{\partial o_2}{\partial net_1} & \frac{\partial o_2}{\partial net_2} & \dots & \frac{\partial o_2}{\partial net_p} \\ \vdots & \vdots & \dots & \vdots \\ \frac{\partial o_p}{\partial net_1} & \frac{\partial o_p}{\partial net_2} & \dots & \frac{\partial o_p}{\partial net_p} \end{pmatrix}$$

Në mënyrë tipike, për detyrat e regresit, ne përdorim funksionin e gabimit në katror me funksionin linear të aktivizimit në neuronet e daljes, ndërsa për regresin logjistik dhe klasifikimin, ne përdorim funksionin e gabimit ndër-entropi me një aktivizim sigmoid për klasat binare dhe aktivizimin e softmax për problemet me shumë klasa. Për këto raste të zakonshme, vektori i gradientit neto në shtresën e daljes jepet si më poshtë:

Gabim në katror: Nga Eq. (4.26), gradienti i gabimit është dhënë si:

$$\partial \mathcal{E}_x = \frac{\partial \mathcal{E}_x}{\partial o} = o - y$$

Gradienti neto në shtresën e daljes është dhënë si:

$$\delta^{h+1} = \partial f^{h+1} \odot \partial \mathcal{E}_x$$

Ku  $\partial f^{h+1}$  varet nga funksioni i aktivizimit në dalje. Në mënyrë tipike, për detyrat e regresit, ne përdorim një aktivizim linear në neuronet e daljes. Në atë rast, ne kemi  $\partial f^{h+1} = 1$ .

**Cross-Entropy Error (dalje binare, aktivizim sigmoid):** Së pari konsideroni rastin binar, me një neuron dalës të vetëm  $o$  me aktivizim sigmoid. Kujtojmë se gabimi binar i ndër-entropisë [Eq. (4.27)] jepet:

$$\mathcal{E}_x = -(y \cdot \ln(o) + (1 - y) \cdot \ln(1 - o))$$

Kemi:

$$\partial \mathcal{E}_x = \frac{\partial \mathcal{E}_x}{\partial o} = \frac{o - y}{o \cdot (1 - o)}$$

Më tej, për aktivizimin sigmoid, kemi:

$$\partial f^{h+1} = \frac{\partial f(\text{net}_o)}{\partial (\text{net}_o)} = o \cdot (1 - o)$$

Prandaj, gradienti neto në neuronin e daljes është:

$$\delta^{h+1} = \partial \mathcal{E}_x \cdot \partial f^{h+1} = \frac{o - y}{o \cdot (1 - o)} \cdot o \cdot (1 - o) = o - y$$

**Cross-Entropy Error (rezultatet K, aktivizimi i softmax):** Kujtojmë që funksioni i Cross-Entropy Error është dhënë si:

$$\mathcal{E}_x = - \sum_{i=1}^K y_i \cdot \ln(o_i) = -(y_1 \cdot \ln(o_1) + \dots + y_K \cdot \ln(o_K))$$

Përdorimi i Eq. (4.35), vektori i derivateve të gabimit në lidhje me neuronet dalëse është dhënë si:

$$\partial \mathcal{E}_x = \left( \frac{\partial \mathcal{E}_x}{\partial o_1}, \frac{\partial \mathcal{E}_x}{\partial o_2}, \dots, \frac{\partial \mathcal{E}_x}{\partial o_K} \right)^T = \left( -\frac{y_1}{o_1}, -\frac{y_2}{o_2}, \dots, -\frac{y_K}{o_K} \right)^T$$

ku  $p = K$  është numri i neuroneve dalëse.

Cross-Entropy Error përdoret zakonisht me aktivizimin e softmax në mënyrë që të marrim një vlerë të probabilitetit (të normalizuar) për secilën klasë. Pra:

$$o_j = \text{softmax}(\text{net}_j) = \frac{\exp\{\text{net}_j\}}{\sum_{i=1}^K \exp\{\text{net}_i\}}$$

në mënyrë që vlerat e neuronit dalës të japin shumën 1,  $\sum_{j=1}^K o_j = 1$ . Meqenëse një neuron dalje varet nga të gjithë neuronet e tjera dalëse, ne duhet të llogarisim matricën e derivateve të secilës dalje në lidhje me secilin prej sinjaleve neto në neuronet e daljes:

$$\begin{aligned} \partial F^{h+1} &= \begin{pmatrix} \frac{\partial o_1}{\partial net_1} & \frac{\partial o_1}{\partial net_2} & \dots & \frac{\partial o_1}{\partial net_K} \\ \frac{\partial o_2}{\partial net_1} & \frac{\partial o_2}{\partial net_2} & \dots & \frac{\partial o_2}{\partial net_K} \\ \vdots & \vdots & \dots & \vdots \\ \frac{\partial o_K}{\partial net_1} & \frac{\partial o_K}{\partial net_2} & \dots & \frac{\partial o_K}{\partial net_K} \end{pmatrix} \\ &= \begin{pmatrix} o_1 \cdot (1 - o_1) & -o_1 \cdot o_2 & \dots & -o_1 \cdot o_K \\ -o_1 \cdot o_2 & o_2 \cdot (1 - o) & \dots & -o_2 \cdot o_K \\ \vdots & \vdots & \dots & \vdots \\ -o_1 \cdot o_K & -o_2 \cdot o_K & \dots & o_K \cdot (1 - o) \end{pmatrix} \end{aligned}$$

Prandaj, vektori i gradientit neto në shtresën e daljes është:

$$\delta^{h+1} = \partial F^{h+1} \cdot \partial \mathcal{E}_x \quad (4.51)$$

$$\begin{aligned} &= \begin{pmatrix} \frac{\partial o_1}{\partial net_1} & \frac{\partial o_1}{\partial net_2} & \dots & \frac{\partial o_1}{\partial net_K} \\ \frac{\partial o_2}{\partial net_1} & \frac{\partial o_2}{\partial net_2} & \dots & \frac{\partial o_2}{\partial net_K} \\ \vdots & \vdots & \dots & \vdots \\ \frac{\partial o_K}{\partial net_1} & \frac{\partial o_K}{\partial net_2} & \dots & \frac{\partial o_K}{\partial net_K} \end{pmatrix} \cdot \begin{pmatrix} -\frac{y_1}{o_1} \\ -\frac{y_2}{o_2} \\ \vdots \\ -\frac{y_K}{o_K} \end{pmatrix} = \begin{pmatrix} -y_1 + y_1 \cdot o_1 + \sum_{i \neq 1}^K y_i \cdot o_1 \\ -y_2 + y_2 \cdot o_2 + \sum_{i \neq 2}^K y_i \cdot o_2 \\ \vdots \\ -y_K + y_K \cdot o_K + \sum_{i \neq K}^K y_i \cdot o_K \end{pmatrix} \\ &= \begin{pmatrix} -y_1 + o_1 \cdot \sum_{i=1}^K y_i \\ -y_2 + o_2 \cdot \sum_{i=1}^K y_i \\ \vdots \\ -y_K + o_K \cdot \sum_{i=1}^K y_i \end{pmatrix} = \begin{pmatrix} -y_1 + o_1 \\ -y_2 + o_2 \\ \vdots \\ -y_K + o_K \end{pmatrix} \left[ \text{meqënëse } \sum_{i=1}^K y_i = 1 \right] \\ &= \mathbf{o} - \mathbf{y} \end{aligned}$$

#### 4.4.6.4 Gradientët në shtresën e ndërmjetme

Le të supozojmë që tani i kemi llogaritur gradientët e rrjetit në shtresën l+1, përkatësisht  $\delta^{l+1}$ . Meqënëse neuroni  $z_j^l$  në shtresën l është e lidhur me të gjitha neuronet në shtresën l+1 (përveç neuronit të bias  $z_0^{l+1}$ ), për të llogaritur gradientin e rrjetit në  $z_j^l$ , ne duhet të llogarisim për gabimin nga secili neuron në shtresën l+1, si më poshtë [33]:

$$\delta_j^l = \frac{\partial \mathcal{E}_x}{\partial net_j} = \sum_{k=1}^{n_{l+1}} \frac{\partial \mathcal{E}_x}{\partial net_k} * \frac{\partial net_k}{\partial f^l(net_j)} * \frac{\partial f^l(net_j)}{\partial (net_j)} = \frac{\partial f^l(net_j)}{\partial (net_j)} * \sum_{k=1}^{n_{l+1}} \delta_k^{l+1} * \omega_{jk}^l$$

Pra, gradienti i rrjetit në  $z_j^l$  në shtresën  $l$  varet nga derivati i funksionit të aktivizimit në lidhje me  $net_j$  e saj, dhe shumë e ponderuar të gradientëve të rrjetit nga të gjitha neuronet  $z_k^{l+1}$  në shtresën tjetër  $l+1$ . Ne mund të llogarisim gradientët e rrjetit për të gjithë neuronet në nivelin  $l$  në një hap, si më poshtë:

$$\partial^l = \partial f^l \odot (W_1 * \delta^{l+1}) \quad (4.52)$$

ku  $\odot$  është produkt element-wise,  $\partial f^l$  është vektori i derivative të funksionit të aktivizimit në lidhje me argumentin e tij në shtresën  $l$ . Për funksionet e aktivizimit që përdoren zakonisht në hidden layer, ne kemi

$$\partial f^l = \begin{cases} 1 & \text{për linear} \\ z^l(1 - z^l) & \text{për sigmoid} \\ (1 - z^l) \odot z^l & \text{për tanh} \end{cases}$$

Për ReLU, duhet të aplikojmë Ek. (4.22) për secilë neuron. Vini re softmax është përgjithësisht jo i përdorur për shtresën e fshehur. Gradientët e rrjetit llogariten në mënyrë rekursive, duke filluar nga shtresa e output-it  $h+1$ , pastaj shtresa e fshehur  $h$ , e kështu me radhë, derisa më në fund të llogarisim gradientët e rrjetit tek shtresa e parë e fshehur  $l=1$ . Kjo është,

$$\delta^h = \partial f^h \odot (W_h * \delta^{h+1})$$

$$\delta^{h-1} = \partial f^{h-1} \odot (W_{h-1} * \delta^h) = \partial f^{h-1} \odot (W_{h-1} * (\partial f^h \odot (W_h * \delta^{h+1})))$$

$$\delta^1 = \partial f^1 \odot (W_1 * (\partial f^2 \odot (W_2 * (\partial f^h \odot (W_h * \delta^{h+1}))))))$$

#### 4.4.7 Trajnimi i MLPs të thella

Algoritmi 2 tregon pseudokodin për të mësuar peshat dhe biaset për deep MLP. Inputet përfshijnë databazën  $D$ , numrin e shtresave të fshehura  $h$ , madhësinë e hapit ose shkallë e të mësuarit për gradientin e zbritjes  $\eta$ , një maksimum integer të thresholdit që tregon numrin e iteracioneve për trajnim, parametrat  $n_1, n_2, \dots, n_h$  që tregojnë numrin e neuroneve (duke përjashtuar biasin, i cili do të shtohet automatikisht) për secilën nga shtresat e fshehura  $l=1, 2, \dots, h$ , dhe llojin e funksioneve të aktivizimit  $f^1, f^2, f^{h+1}$  për secilën nga shtresat (përveç shtresës së inputit që përdor aktivizimin e identitetit). Madhësia e inputit ( $d$ ) dhe e outputit ( $p$ ) përcaktohet direkt nga  $D$ . MLP fillimisht inicializon matricat e peshës ( $n_1 \times n_{l+1}$ )  $W_l$  midis shtresave  $l$  dhe  $l+1$  me vlera të vogla të zgjedhura në mënyrë uniforme në mënyrë të rastësishme, p.sh., në intervalin  $[-0,01, 0,01]$ . MLP konsideron secilën pjesë hyrëse  $(x_i, y_i) \in D$  dhe llogarit përgjigjen e parashikuar  $o_i$  përmes procesit feed-forward. Faza e



backpropagation fillon duke llogaritur gabim midis oi dhe përgjigjes së vërtetë  $y_i$  dhe llogaritjen e vektorit gradient të rrjetit  $\delta^{h+1}$  në shtresën e outputit. Këto gradient të rrjetit backpropagated nga shtresa  $h + 1$  në shtresën  $h$ , nga  $h$  në  $h - 1$ , e kështu me radhë derisa të marrim gradientët e rrjetit në shtresën e parë të fshehur  $l = 1$ . Këto gradientë të rrjetit përdoren për të llogaritur matricën e gradientit të peshës  $\nabla W_l$  në shtresën  $l$ , e cila nga ana tjetër mund të përdoret për të updatuar matricën e peshës  $W_l$ . Po kështu, gradientët e rrjetit specifikojnë vektorin e gradientit bias  $\nabla b_l$  në shtresën  $l$ , i cili përdoret për të updatuar  $b_l$ . Pas secilës pikë është përdorur për të përditësuar peshat, që plotësojnë një iterenar ose epokë të trajnimit. Trajnimi ndalet kur të jenë arritur epokat maksimale. Në anën tjetër gjatë testimit, për çdo hyrje  $x$ , ne zbatojmë hapat e feed-forward dhe printojmë outputin e parashikuar  $o$ . Është e rëndësishme të theksohet se Algoritmi 2 ndjek një qasje në zbritje të gradientit stokastik, pasi që pikat merren në mënyrë të rastësishme dhe gradientët e peshës dhe biasit llogariten pasi vëzhgojnë secilën pike të trajnimit. Në praktikë, është e zakonshme për të përditësuar gradientët duke konsideruar një nëngrup me madhësi fikse të pikave të trajnimit të quajtura minibatch në vend që të përdorë pikat e vetme. Të dhënat e trajnimit ndahen në minibatches duke përdorur një parametër shtesë të quajtur batch size dhe hapi i zbritjes së gradientit kryhet pas llogaritjes së biasit dhe peshës së gradientit nga secili minibatch. Kjo ndihmon në vlerësimin më të mirë të gradientëve dhe gjithashtu lejon veprime të matricës së vektorizuar mbi pikat e minibatch, të cilat mund të çojnë në konvergjencë më të shpejtë dhe thelbësore shpejtësi në të mësuar. Një paralajmërim gjatë trajnimit të MLP-ve shumë të thella është problemi i zhdukjes dhe gradientët që shpërthejnë. Në problemin e gradientit që zhduket, norma e gradientit të rrjetit mund të prishet në mënyrë eksponenciale me distancën nga shtresa e output-it, domethënë siç në backpropagate gradientët nga shtresa e output-it në shtresën e input-it. Në këtë rast rrjeti do të mësojë jashtëzakonisht ngadalë, që nga metoda e zbritjes gradientit do të bëjë ndryshime të vogla në peshat dhe biaset. Nga ana tjetër, problemi i shpërthimit të gradientit, norma e gradientit të rrjetit mund të rritet në mënyrë eksponenciale me distancën nga shtresa e output-it. Në këtë rast, peshat dhe biaset do të bëhen në mënyrë eksponenciale të mëdha, duke rezultuar në një dështim. Problemi i shpërthimit të gradient mund të zbutet në një farë mase nga pragu i gradientit, domethënë duke rivendosur vlerën nëse tejkalon një kufi të sipërm. Problemi i gradienteve në zhdukje është më i vështirë për tu adresuar. Në mënyrë tipike aktivizimet sigmoidale janë më të ndjeshme ndaj këtij problem dhe një zgjidhje është përdorimi i funksioneve alternative të aktivizimit siç është ReLU. Në përgjithësi, rrjetet neurale të përsëritura, të cilat janë rrjete të thella neurale me lidhje kthyesë, janë më të prirura në zhdukjen dhe shpërthimin e gradientëve [33].

---

```

DEEP-MLP-TRAINING ( $\mathbf{D}, h, \eta, \text{maxiter}, n_1, n_2, \dots, n_h, f^1, f^2, \dots, f^{h+1}$ ):
1  $n_0 \leftarrow d$  // input layer size
2  $n_{h+1} \leftarrow p$  // output layer size
   // Initialize weight matrices and bias vectors
3 for  $l = 0, 1, 2, \dots, h$  do
4    $\mathbf{b}_l \leftarrow$  random  $n_{l+1}$  vector with small values
5    $\mathbf{W}_l \leftarrow$  random  $n_l \times n_{l+1}$  matrix with small values
6  $t \leftarrow 0$  // iteration counter
7 repeat
8   foreach  $(\mathbf{x}_i, \mathbf{y}_i) \in \mathbf{D}$  in random order do
9     // Feed-Forward Phase
10     $\mathbf{z}^0 \leftarrow \mathbf{x}_i$ 
11    for  $l = 0, 1, 2, \dots, h$  do
12       $\mathbf{z}^{l+1} \leftarrow f^{l+1}(\mathbf{b}_l + \mathbf{W}_l^T \cdot \mathbf{z}^l)$ 
13     $\mathbf{o}_i \leftarrow \mathbf{z}^{h+1}$ 
14    // Backpropagation Phase
15    if independent outputs then
16       $\delta^{h+1} \leftarrow \partial \mathbf{f}^{h+1} \odot \partial \mathcal{E}_{\mathbf{x}_i}$  // net gradients at output
17    else
18       $\delta^{h+1} \leftarrow \partial \mathbf{F}^{h+1} \cdot \partial \mathcal{E}_{\mathbf{x}_i}$  // net gradients at output
19    for  $l = h, h-1, \dots, 1$  do
20       $\delta^l \leftarrow \partial \mathbf{f}^l \odot (\mathbf{W}_l \cdot \delta^{l+1})$  // net gradients at layer  $l$ 
21    // Gradient Descent Step
22    for  $l = 0, 1, \dots, h$  do
23       $\nabla_{\mathbf{W}_l} \leftarrow \mathbf{z}^l \cdot (\delta^{l+1})^T$  // weight gradient matrix at layer  $l$ 
24       $\nabla_{\mathbf{b}_l} \leftarrow \delta^{l+1}$  // bias gradient vector at layer  $l$ 
25    for  $l = 0, 1, \dots, h$  do
26       $\mathbf{W}_l \leftarrow \mathbf{W}_l - \eta \cdot \nabla_{\mathbf{W}_l}$  // update  $\mathbf{W}_l$ 
27       $\mathbf{b}_l \leftarrow \mathbf{b}_l - \eta \cdot \nabla_{\mathbf{b}_l}$  // update  $\mathbf{b}_l$ 
28     $t \leftarrow t + 1$ 
29 until  $t \geq \text{maxiter}$ 

```

---

**Algoritmi 2.** Trajnimi për rrjetat e thella neurale [33]

## 4.5 Support Vector Machines SVM

Support Vector Machines (SVMs) është një metodë klasifikimi e bazuar në maksimumin marxhinal të diskriminuesve linear, që ka si qëllim kryesor gjetjen e planit maksimal që maksimizon largësinë ose marzhin mes dy klasave. Një hiperplan në  $d$  dimensione është dhënë si seti i pikave  $x \in \mathbb{R}^d$  që plotësojnë ekuacionin  $h(x) = 0$ , ku  $h(x)$  është funksioni i hiperplanit, përcaktuar si më poshtë:  $h(x) = wTx + b = w_1x_1 + w_2x_2 + \dots + w_dx_d + b$ . Ku,  $w$  është një vektor peshë  $d$  dimensional dhe  $b$  është një vektor scalar i quajtur bias. Për të gjitha pikat që bëhen pjesë e këtij hiperplani kemi:  $h(x) = wTx + b = 0$ . Një hiperplan e ndan një hapësirë  $d$ -dimensionale në 2 gjysmë hapësira. Një dataset quhet linear nëse çdo gjysmë e tij ka pikë vetëm nga një klasë. Nëse dataseti input është linearisht i ndashëm, atëherë ne mund të gjejmë një hiperplan ndarës  $h(x) = 0$ , i tillë që për çdo pikë të klasifikuar si  $y_i = -1$ , ne kemi  $h(x_i) < 0$ , dhe për të gjitha pikat e klasifikuara si  $y_i = +1$ , ne kemi  $h(x_i) > 0$ . Në fakt, funksioni hiperplan  $h(x)$  shërben si një klasifikues linear ose si një diskriminues linear, që parashikon klasën  $y$  për çdo pikë  $x$ , sipas rregullit:  $y = +1$  nëse  $h(x) > 0$  dhe  $-1$  nëse  $h(x) < 0$ . Distanca direkte e një pike nga hiperplani mund të shkruhet si më poshtë:  $r = h(x) / \|w\|$  [33].

### 4.5.1 SVM për më shumë se 2 klasa

Vetëm se deri në këtë moment kemi trajtuar rastin kur kemi vetëm 2 klasa. Metoda Support vector machine është thelbësisht një metodë 2 klasifikuese [34].

Ndërkohë në praktikë janë të shumta rastet kur nevojitet të klasifikojmë në më shumë se 2 klasa,  $K > 2$ . Janë të ndryshme metodat e propozuara për kombinimin e më shumë SVM-ve me qëllim ndërtimin e një klasifikuesi multiklasor.

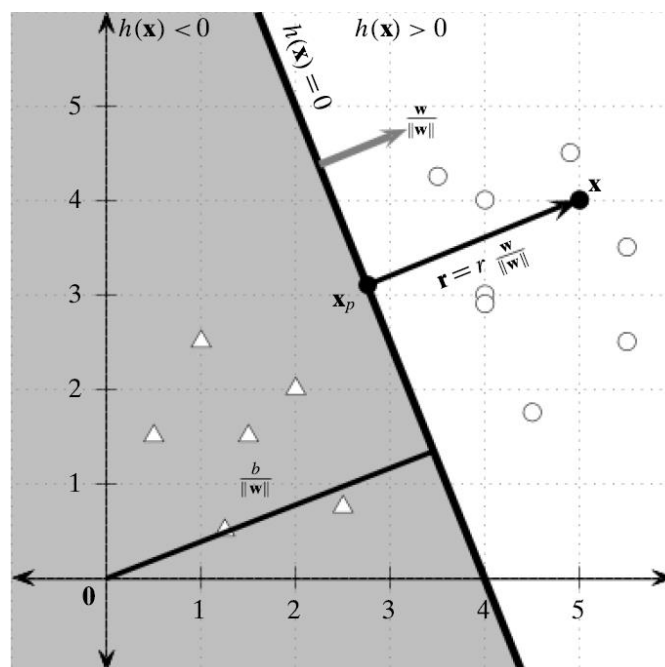


Figure 4-14 Plani ndarës në 2D

Një metodë shumë e njohur është ajo e (Vapnik, 1998), për të ndërtuar K SVM-më të veçanta, në të cilën modeli  $yk(x)$  ndërtohet duke përdorur të dhënat nga klasa Ck si shembujt pozitivë, ndërsa të dhënat nga pjesa e mbetur e klases K-1 si shembujt negativ. Kjo njihet ndryshe si qasja një kundër pjesës së mbetur (one-versus-the-rest).

Një përfaqje tjetër ndaj klasifikimit në më shumë se 2 klasa, u zhvillua nga Dietterich dhe Bakiri (1995), bazuar në korrigjimin e errësive në kodet output. Kjo metodë u përdor në support vector machines nga Allwein et al. (2000). Kjo mund të shihet si një përgjithësim i skemës së votimit të qasjes një-kundër-një në të cilën përdoren ndarje më të përgjithshme të klasave për të trajnuar klasifikuesit individualë. Vetë klasat K përfaqësohen si grupe të veçanta nga klasifikuesit me dy klasa të zgjedhura dhe së bashku me një skemë të përshtatshme dekodimi, kjo u jep qëndrueshmëri gabimeve dhe paqartësisë në rezultatet e klasifikuesve individualë. Megjithatë zbatimi i SVM-ve në problemet e klasifikimit me shumë klasa mbetet një çështje e hapur, në praktikë qasja një-kundër-pjesës së mbetur (one-versus-the-rest) është më e përdorura pavarësisht formulimit të saj ad-hoc dhe kufizimeve të saj në praktikë. [34]

#### 4.5.2 SVM me funksionin Bërthamë: Rasti jo-linear

Algoritmet SVM përdorin një set funksionesh matematikore të njohura ndryshe si kernel (bërthama). Funksioni i një bërthame është marrja si input i setit të të dhënave dhe transformimi i tij në një formë të caktuar. Tipe të ndryshëm algoritmesh SVM përdorin tipe të ndryshme funksionesh bërthame. Këto funksione mund të jenë lineare, jolineare, polynomial, radial basis (RBF) dhe sigmoid. Funksionet lineare të SVM përdoren në sete të dhënash me problematikë në dimension dhe në linearitet. Zgjidhja më e mirë në këto raste është përdorimi i bërthamës, kjo e fundit është metoda e cila në një dataset që nuk arrin të ndahet me dimensionet aktuale, i nënshtrohet shtimit të një dimensionit të ri, që mund ta bëjë më të ndashëm datasetin. Në parim ideja është të mapojmë çdo pikë  $\mathbf{x}_i$  të datasetit input **d-dimensional**, me një pikë  $\phi(\mathbf{x}_i)$  të datasetit në një dimension më të lartë, përmes disa transformimeve jolineare  $\phi$ . Përmes një fleksibiliteti më të madh, pikat  $\phi(\mathbf{x}_i)$  kanë më shumë shanse që të jenë linearisht të ndashme, duke bërë kështu që një dataset linear i përfutur nga transformimet jolineare  $\phi$ , t'i korespondojë një dataseti jo-linear në input. Kështu funksioni bërthame mundëson të gjitha operacionet e nevojshme për ta transformuar një dataset input në një dataset me një dimension të ndryshuar por me një shkallë lineariteti dhe fleksibiliteti më të lartë. Për të përdorur funksionin bërthame për klasifikimin SVM jo-linear, duhet të tregojmë që të gjitha operacionet kërkojnë vetëm funksionin bërthame [33]:

$$K(\mathbf{x}_i, \mathbf{x}_j) = \phi(\mathbf{x}_i)^T \phi(\mathbf{x}_j) \quad (4.53)$$

Ku  $\mathbf{d}$  - është një dataset input me n pika  $\mathbf{x}_i$  dhe labels  $\mathbf{y}_i$ . Duke aplikuar  $\phi$  ndaj çdo pike, ne përfutur një dataset të ri  $\mathbf{D}_\phi$  në një hapësirë të re, e përbërë nga pika të transformuara  $\phi(\mathbf{x}_i)$  me labels përkatës:  $\mathbf{y}_i$ , për  $\mathbf{i} = 1, 2, \dots, \mathbf{n}$ .

Funksioni SVM në hapësirën e re është si më poshtë:

$$\text{Funksioni Objektivi: } \min_{w, b, \varepsilon} \left\{ \frac{\|w\|^2}{2} + C \sum_{i=1}^n \varepsilon_i^k \right\} \quad (4.54)$$

**Kushtet Lineare:**  $y_i(w^T \phi(\mathbf{x}_i) + b) \geq 1 - \varepsilon_i$  dhe  $\varepsilon_i \geq 0$  për çdo  $\mathbf{x}_i \in \mathbf{D}$

Ku  $w$  është vektori peshë,  $b$  është bias dhe  $\xi_i$  janë variablat e tipit **slack variables**, të cilët bëjnë të mundur kthimin e një funksioni nga i pabarabartë në të barabartë. Për shembull variabli slack

$\mathbf{y} \geq \mathbf{0}$  bën të mundur që nga  $\mathbf{Ax} \leq \mathbf{b}$  të kalojmë në:  $\mathbf{Ax} + \mathbf{y} = \mathbf{b}$ . Në pemët e vendimarrjes si dhe në optimizimet matematikore funksionet e humbjes (humbja functions) janë funksionet që mapojnë një event ose vlerat e disa variablave me një numër ose event real, duke mbartur kështu edhe një kosto. Një problem optimizimi, synon të minimizojë në maksimum humbjet. Në statistikë, një funksion humbjeje përdoret kryesisht në vlerësimin e parametrizimit, eventi në këtë rast është një funksion i diferencës mes vlerave të estimuara dhe atyre reale. Më poshtë disa prej funksioneve kryesore të humbjes që përdoren në SVM.

## Humbja Hinge

Humbja Hinge, në ML është një funksion që përdoret për trajnimin e klasifikuesve, në gjetjen e klasifikimit "maximum-margin".

Funksioni dual Lagrangian për gjetjen e maksimumit në datasetin e përftuar nga transformimet, është si më poshtë:

$$\begin{aligned} \max_{\alpha} L_{dual} &= \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j y_i y_j \phi(\mathbf{x}_i)^T \phi(\mathbf{x}_j) \\ &= \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j y_i y_j K(\mathbf{x}_i, \mathbf{x}_j) \end{aligned} \quad (4.55)$$

Duke marrë parasysh që  $0 \leq \alpha_i \leq C$ , dhe  $\sum_{i=1}^n \alpha_i y_i = 0$ .

Dual Lagranzh varet vetëm nga produkti i pikave ndërmjet dy vektorëve në hapësirën e transformuar  $\phi(\mathbf{x}_i)^T \phi(\mathbf{x}_j) = K(\mathbf{x}_i, \mathbf{x}_j)$ , kështu ne mund të zgjidhim problemin e optimizimit duke përdorur matricën e bërthamës  $\mathbf{K} = \{K(\mathbf{x}_i, \mathbf{x}_j)\}$   $i, j = 1, \dots, n$  [33].

## Humbja Kuadratike

Funksioni i humbjes kuadratike është matematikisht më i gjurmueshëm se sa funksionet e tjera të humbjes për shkak të vetive të mospërputhjeve, si dhe faktit që është simetrik: një gabim mbi target shkakton të njëjtën humbje me të njëjtën madhësi gabimi nën target. Nëse objektivi është  $t$ , atëherë një funksion i humbjes kuadratike është  $\lambda(x) = C(t-x)^2$ , për një konstante  $C$ , ku vlera e konstantes nuk bën shumë diferencë dhe mund ta caktojmë edhe 1. Për humbjen kuadratike, dual Lagrangian i korrespondon një ndryshimi në bërthamë. Përcaktojmë një funksion të ri të bërthames  $K_q$ , si më poshtë:

$$K_q(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{x}_i^T \mathbf{x}_j + 1/2C \lambda_{ij} = K(\mathbf{x}_i, \mathbf{x}_j) + 1/2C \delta_{ij}$$

e cila prek vetëm hyrjet diagonale të matricës së bërthamës  $\mathbf{K}$ , si  $\delta_{ij} = 1$ , nëse  $i = j$  dhe në rast të kundërt është  $i = 0$ .

Kështu, funksioni i dual Lagrangian është si më poshtë:

$$\max_{\alpha} L_{dual} = \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j y_i y_j K_q(\mathbf{x}_i, \mathbf{x}_j) \quad (4.56)$$

Duke respektuar kufizimet që  $\alpha_i \geq 0$ , dhe  $\sum \alpha_i y_i = 0$ .

Optimizimi i mësipërm mund të zgjidhet duke përdorur të njëjtën qasje edhe me funksionin Hinge Humbja, me një ndryshim të thjeshtë të bërthamës [33].

### Vektori i peshës dhe Bias

Peshat dhe bias janë koncepti më i rëndësishëm në rrjetat neurale. Kur inputet transmetohen përmes neuroneve, janë peshat ato që i atashohen inputit dhe i kalohen funksionit final të aktivizimit bashkë me bias.

Ne mund të përcaktojmë në hapësirën e re të përfutur peshën  $\mathbf{w}$  si më poshtë:

$$\mathbf{w} = \sum_{\alpha_i > 0} \alpha_i y_i \phi(\mathbf{x}_i)$$

Duke qënë që  $\mathbf{w}$  përdor direkt  $\phi(\mathbf{x}_i)$ , përgjithësisht është e vështirë të paracaktohet qartësisht peshat  $\mathbf{w}$ , pavarësisht se në vijim do shohim që nuk është edhe shumë e nevojshme përcaktimi i qartë i peshës  $\mathbf{w}$ . Le të shohim se si të llogarisim **bias** duke përdorur operacionet bërthame për hinge humbja. Ajo llogaritet si mesatare mbi support vectors që janë në marzhin, ku  $\mathbf{0} < \alpha_i < C$ , dhe  $\xi_i = \mathbf{0}$ :

$$b = \text{avg}_{0 < \alpha_i < C} \{b_i\} = \text{avg}_{0 < \alpha_i < C} \{y_i - \mathbf{w}^T \phi(\mathbf{x}_i)\}$$

Në humbjen kuadratike, bias është mesatarja mbi të gjithë support vectors me  $\alpha_i > 0$ .

Më tej duke zëvendësuar  $\mathbf{w}$ , përftojme një shprehje të re për  $b_i$  si:

$$\begin{aligned} b_i &= y_i - \sum_{\alpha_j > 0} \alpha_j y_j \phi(\mathbf{x}_j)^T \phi(\mathbf{x}_i) \\ &= y_i - \sum_{\alpha_j > 0} \alpha_j y_j K(\mathbf{x}_j, \mathbf{x}_i) \end{aligned} \quad (4.57)$$

Ku  $b_i$  është një funksion prodhimi i pikave mes dy vektorëve në hapësirën e re të transformuar dhe mund të llogaritet kështu përmes funksionit bërthame në datasetin input.

### Klasifikuesi SVM Bërthame

Ne mund të parashikojmë klasën për një pikë të re  $\mathbf{z}$  si më poshtë:

$$\begin{aligned} \hat{y} &= \text{sign}(\mathbf{w}^T \phi(\mathbf{z}) + b) = \text{sign}\left(\sum_{\alpha_i > 0} \alpha_i y_i \phi(\mathbf{x}_i)^T \phi(\mathbf{z}) + b\right) = \\ &= \text{sign}\left(\sum_{\alpha_i > 0} \alpha_i y_i K(\mathbf{x}_i, \mathbf{z}) + b\right) \end{aligned} \quad (4.58)$$

Sërisht shikojmë që  $y$  përdor vetëm prodhimin e pikave në hapësirën e re të transformuar.

Bazuar në derivativët më lart, shikojmë që për të ndërtuar dhe për të testuar klasifikuesin SVM, pikat e mapuara  $\phi(\mathbf{x}_i)$  nuk nevojiten asnjëherë që të izoloohen. Si zëvendësim të gjitha operacionet mund të mbarten nga funksioni bërthame më poshtë:  $\mathbf{K}(\mathbf{x}_i, \mathbf{x}_j) = \phi(\mathbf{x}_i)^T \phi(\mathbf{x}_j)$ .

Kështu çdo funksion bërthame jo-linear mund të përdoret për të bërë klasifikime jo-lineare në hapësirën input. Shembuj të bërthamave të këtilla jo-lineare janë bërthama polinomiale dhe bërthama gaussiane[33].

#### 4.6 Klasifikuesi i K fqinjëve më të afërt (KNN)

Në këtë seksion do të analizohet një teknikë jo-parametrike, e cila nuk bën supozim rreth funksionit të densitetit të probabilitetit të përbashkët. Metoda përdor në mënyrë direkte kampionet e të dhënave për të vlerësuar densitetin.

Shënojmë me  $\mathbf{D}$  datasetin e trajnuar i cili përfshin  $n$  pikat  $\mathbf{x}_i \in R^d$  dhe me  $\mathbf{D}_i$  shënojmë nënbashkësinë e pikave të  $\mathbf{D}$  të etiketuara me klasën  $c_i$ , ku  $n_i = |\mathbf{D}_i|$ . Për një pikë test  $\mathbf{x} \in R^d$  dhe  $K$ , numri i fqinjëve të tij, shënojmë me  $r$  distancën nga pika  $\mathbf{x}$  në  $K$  fqinjët më të afërt në  $\mathbf{D}$ . Duke marrë në konsideratë hiper-sferën me rreze  $r$  rreth pikës së testimit  $\mathbf{x}$ , si në formulën:

$$B_d(\mathbf{x}, r) = \{\mathbf{x}_i \in \mathbf{D} \mid \|\mathbf{x} - \mathbf{x}_i\| \leq r\}$$

Ku  $\|\mathbf{x} - \mathbf{x}_i\|$  është distanca Euklidiane ndërmjet pikave  $\mathbf{x}$  dhe  $\mathbf{x}_i$ . Supozojmë se  $|B_d(\mathbf{x}, r)| = K$ .

Shënojmë me  $K_i$  numrim e pikave përgjatë  $K$  fqinjëve më të afërt të  $\mathbf{x}$  të etiketuara me klasën  $c_i$ , që plotësojnë relacionin:

$$K_i = \{\mathbf{x}_j \in B_d(\mathbf{x}, r) \mid y_j = c_i\}$$

Densiteti i probabilitetit të kushtëzuar në pikën  $\mathbf{x}$  mund të vlerësohet si fraksioni i pikës ng klasa  $c_i$ , e cila ndodhe brenda hipersferës pjesëtuar në volumnin e saj, si në formulë [33]:

$$\hat{f}(\mathbf{x} | c_i) = \frac{K_i / n_i}{V} = \frac{K_i}{n_i V}$$

Ku  $V = \text{vol}(B_d(\mathbf{x}, r))$  është volumni i hipersferës. Probabiliteti posterior mund të vlerësohet si:

$$P(c_i | \mathbf{x}) = \frac{\hat{f}(\mathbf{x} | c_i) \hat{P}(c_i)}{\sum_{j=1}^k \hat{f}(\mathbf{x} | c_j) \hat{P}(c_j)}$$

Meqënëse  $\hat{P}(c_i) = \frac{n_i}{n}$  përftojme:

$$\hat{f}(x|c_i) \hat{P}(c_i) = \frac{K_i}{n_i V} \cdot \frac{n_i}{n} = \frac{K_i}{nV}$$

Probabiliteti posterior jepet në formulën:

$$P(c_i|x) = \frac{\frac{K_i}{nV}}{\sum_{j=1}^K \frac{K_j}{nV}} = \frac{K_i}{K} \quad (4.59)$$

Klasa e parashikuar për pikën  $\mathbf{x}$  është:

$$\hat{y} = \operatorname{argmax}_{c_i} \{P(c_i|x)\} = \operatorname{argmax}_{c_i} \left\{ \frac{K_i}{K} \right\} = \operatorname{argmax}_{c_i} \{K_i\} \quad (4.60)$$

Meqënëse  $K$  është numër fiks, klasifikuesi KNN parashikon klasën e  $x$  si klasa mazhorance ndërmjet  $K$  fqinjëve më të afërt.

## 4.7 Algoritmi Prophet

Prophet [38] është një model parashikues i cili punon me karakteristikat kryesore të serive kohore të biznesit duke i rregulluar parametrat pa informacion paraprak të modelit [36]. Tre komponente të modelit implementohen: trendi, sezonaliteti dhe ditët festive si në ekuacionin më poshtë:

$$y(t) = g(t) + s(t) + h(t) + \epsilon_t \quad (4.61)$$

ku  $g(t)$  implementon funksionin e trendit si ndryshime jo periodike të vlerave të serive kohore,  $s(t)$  implementon ndryshimet periodike dhe  $h(t)$  implementon efektin e ditëve festive dhe termi i gabimit përfaqson ndryshimet idiosinkratike, zakonisht si një vlerë me një shpërndarje normale [37].

Parashikimi konsiderohet si një problem përshtatjes së kurbës duke marrë parasysh strukturën e varësisë kohore të të dhënave. Sistemi është fleksibël, çka do të thotë që periudha të shumta të sezonalitetit përfshihen lehtësisht, matjet nuk kanë nevojë të behen rregullisht, aplikimi është shumë i shpejtë dhe përdoruesit mund të manipulojnë në mënyrë interaktive shumë parametra të modelit ose të përfshijnë komponentë të rinj [36]

### 4.7.1 Modeli i Trendit

Përdorimet më të zakonshme të modelit të trendit janë 2, modeli i rritjes së saturuar dhe modeli linear [36]. Modeli i rritjes së saturuar jepet në ekuacionin më poshtë:

$$g(t) = \frac{c}{1 + \exp(-k(t-m))} \quad (4.62)$$

ku  $c$  është kapaciteti i mbartur,  $k$  është parametri i rritjes dhe  $m$  është parametri i offset.



Në bazë të analizës së ekuacionit më sipër, mund të themi se  $C$  dhe  $k$ , nuk janë zakonisht parametra konstante. I pari zëvendësohet nga  $C(t)$ . Në të dytin përfshihen pikat e ndryshimit ku rendi i rritjes lejohet të ndryshojë. Duke shënuar pikat e ndryshimit me  $s_j$ , ku  $j=1\dots S$ , përlllogaritet një vektor sistemimesh  $\delta \in R^S$  nga parametri bazë  $k$  dhe sistemimet deri në këtë pikë si në ekuacionin më poshtë:

$$t = k + \mathbf{a}(t)^T \boldsymbol{\delta}$$

ku:

$$a_j(t) = \begin{cases} 1, & \text{if } t \geq s_j \\ 0, & \text{ne te kundert} \end{cases}$$

Parametri i offset  $m$ , duhet të rregullohet sipas parametrin  $k$  në mënyrë që të lidhen pikat fundore të segmenteve [39]. Sistemimet në pikat fundore tregohen më poshtë:

$$\gamma_j = \left( s_j - m - \sum_{l < j} \gamma_l \right) \left( 1 - \frac{k + \sum_{l < j} \delta_l}{k + \sum_{l \leq j} \delta_l} \right)$$

Modeli i rritjes së saturuar si më poshtë:

$$g(t) = \frac{c(t)}{1 + \exp\left(-\left(k + \mathbf{a}(t)^T \boldsymbol{\delta}\right)\left(t - \left(m + \mathbf{a}(t)^T \boldsymbol{\gamma}\right)\right)\right)} \quad (4.63)$$

Ne rastet kur parashikimi nuk e kënaq modelin e saturuar të rritjes, përdoret rendi konstant i rritjes. Modeli i trendit shndërrohet në:

$$g(t) = \left(k + \mathbf{a}(t)^T \boldsymbol{\delta}\right)t + \left(m + \mathbf{a}(t)^T \boldsymbol{\gamma}\right) \quad (4.64)$$

ku:

$$\gamma_j = -s_j \delta_j.$$

#### 4.7.2 Modeli i sezonalitetit

Periudhat e shumëfishta të sezonalitetit shfaqen vazhdimisht në seritë kohore të biznesit. Modelet e sezonalitetit duhet të specifikohen si funksione periodike në lidhje me  $t$ . Seritë Fourier përdoren për të ofruar fleksibilitet për modelet periodike. Efekti sezonal tregohet në ekuacionin më poshtë:

$$s(t) = \sum_{n=1}^N \left( a_n \cos\left(\frac{2\pi n t}{P}\right) + b_n \sin\left(\frac{2\pi n t}{P}\right) \right) \quad (4.65)$$

ku  $P$  është perioda e serive kohore.  $2N$  parametra,  $\beta = [a_1, b_1, \dots, a_N, b_N]^T$  përcaktohen për sezonalitetin. Matrica e sezonalitetit ndërtohet nga çdo vlerë e  $t$  në të shkuarën dhe të ardhmen [40]. Komponentja e sezonalitetit tregohet si më poshtë:

$$S(t) = X(t) \beta$$

### 4.7.3 Ditët Festive

Ditët festive dhe eventet, zakonisht nuk janë periodike dhe shkaktojnë shumë ndryshime të parashikueshme në seritë kohore të biznesit. Duke qënë se ato ndodhin çdo vit, është e rëndësishme ti përfshijmë në parashikim [41]. Duke supozuar se efekti i tyre është i pavaruar, një listë e ditëve festive vendoset në sistem. Një funksion përcakton nëse një kohë t është pergjatë ditëve festive  $i$ , duke alokuar parametrin  $k_i$  si pikë ndryshimi në parashikim. Së fundmi krijohet një matricë për regresorët:

$$Z(t) = [\mathbf{1}(t \in D_1), \dots, \mathbf{1}(t \in D_L)] \quad (4.66)$$

Dhe  $h(t) = Z(t)k$ . Gjithashtu mund të përcaktohen parametra të tjera për ditët para dhe pas ditëve festive.

### 4.7.4 Metrikat e gabimit

Saktësia në parashikim mund të vlerësohet nëpërmjet vlerës mesatare të gabimit absolut në parashikim, MAE (Mean Average Error) ose vlerës mesatare të gabimit absolut në përqindje MAPE (Mean Absolute Percentage Error). Formulatat për secilin gabim jepen në ekuacionet 4.67 dhe 4.68:

$$MAE = \frac{1}{n} \sum_{j=1}^n |y_j - \bar{y}_j| \quad (4.67)$$

$$MAPE = \frac{1}{n} \sum_{j=1}^n \left| \frac{y_j - \bar{y}_j}{y_j} \right| \times 100\% \quad (4.68)$$

#### 4.1 Konkluzione

Në këtë kapitull u analizuan teknikat dhe algoritmat që u implementuan dhe modifikuan gjatë pjesës eksperimentale. Fillimisht u studiuuan algoritmat e klasterizimit dhe veçanërisht algoritmat K-Means, K-Medoids dhe algoritmat hierarkike. Aparati matematikor i secilit algoritëm u shpjegua dhe ilustrua me shëmbuj dhe skema. Më pas u studiuuan teknikat e klasifikimit. Në këtë pjesë u studiuuan nga ana teorike algoritmat KNN (Klasifikuesi i K-fqinjëve më të afërt), algoritmi SVM dhe rrjetat neurale.

Aparati matematikor i secilit algoritëm u shpjegua dhe analizua.

Vlen për tu theksuar se ju dha një fokus më i madh rrjetave neurale (NN) duke qënë se besojmë se janë teknikat më premtuese të klasifikimit dhe regresit. Arkitekturat kryesore të rrjetave NN u shpjeguan dhe fazat e përparme dhe të pas-shpërndarjes u shtjelluan. Gjithashtu u analizuan funksionet e ndryshme të aktivizimit dhe të gabimit që përdorën. Të gjitha këto funksione do të implementohen në modelin e krijuar në pjesën eksperimentale. Duhet të theksojmë se analiza teorike e teknikave të klasifikimit dhe e rrjetave neurale bazohet thelbësisht në librin “*Data Mining and Machine Learning Fundamental Concepts and Algorithms by Mohammed J. Zaki, Wagner Meira Jr, Cambridge University Press, 2020*” të referuar me numrin 33 duke i besuar cilesisë shumë të lartë të analizës që kryhet në të.

## KAPITULLI 5

### 5. PUNA EKSPERIMENTALE

Pjesa eksperimentale do të ndahet në 3 pjesë kryesore. Në pjesën e parë do të analizohen modelet grupuese në klastera dhe do të grupohen klientët në klasetrat kryesorë. Në pjesën e dytë do të analizojmë dhe propozojmë modelet parashikuese. Në pjesën e tretë do të analizojmë dhe propozojmë modelet e serive kohore me fokus në algoritmin Prophet. Baza e të dhënave do të ndahet në 3 pjesë kryesore të cilat do të përdoren për 3 seksionet e ndryshme të analizës. Në bazën e parë të të dhënave do të agregojmë nga baza e të dhënave transaksionale blerjet për vitin 2019 të klientëve të qendrës tregtare. Në këtë bazë të dhënash do të kemi 9931 rreshta. Kjo bazë të dhënash do të përdoret për të implementuar modelet e klasterizimit. Në bazën e dytë të të dhënave do të kemi 3797 rreshta të blerjeve të klientëve për 3-mujorin e parë të 2020 dhe kjo bazë të dhënash do të përdoret për të implementuar modelet parashikues me qëllim parashikimin e klasës së klientëve në fund të vitit. Baza e tretë e të dhënave konsiston te të gjitha shitjet ditore nga viti 2011-2019 dhe kjo bazë e dhënash do të përdoret me qëllim aplikimin e modelit Prophet të serive kohore. Qëllimi do të jete parashikimi i shitjeve për vitin 2020 për të gjithë qendrën.

Teknikat e analizuara në pjesen teorike do të përdoren për të krijuar disa modele të cilat do të përdoren për grupimin dhe klasifikimin e klientëve të qendrës tregtare. Në pjesën e parë do të aplikohen modelet e klasterizimit të të dhënave dhe do të analizohet performanca e tyre. Të gjitha algoritmat kryesore të klasterizimit do të përdoren si, K-Means, K-Medoids, klasterat hierarkike dhe dbscan [42]. Në pjesën e dytë do analizojmë dhe implementojmë modelet e klasifikimit. Do të krijojmë 3 modele parashikues. Modeli i parë do të bazohet në algoritmin KNN (K-Nearest Neighbours). Modeli do të parametrizohet dhe modifikohet për të arritur rezultatet më të mira. Modeli i dytë do të bazohet në SVM (Support Vector Machines) dhe do të modifikohet dhe parametrizohet për të arritur rezultatet maksimale. Për modelin e tretë do të ndërtojmë një rrjet neural i cili do të parametrizohet në mënyrë që të arrijmë rezultatet maksimale. Në pjesën e tretë do të analizohet dhe propozohet një model i bazuar te seritë kohore dhe në veçanti në algoritmin Prophet. Do të kryhen modifikime të modelit bazë dhe do të krahasohet performanca e modeleve.

Analiza do të kryhet duke përdorur SQL për agregimin e të dhënave dhe Python dhe Pytorch për ndërtimin e modeleve të klasterizimit apo parashikimit. Kodet e përdorura do të bashkangjiten punimit si anekse.

#### 5.1 Teknikat e Klasterizimit

##### 5.1.1 Eksplorimi i bazës së të dhënave

Baza e të dhënave që do të përdoret do të merret duke agreguar me SQL blerjet e klienteve për vitin 2019. Blerjet do të grupohen në 3 kategori kryesore si: Veshje, Aksesorë dhe Të tjera. Në grupin e veshjeve përfshihen blerjet e veshjeve të të gjitha llojeve në të gjitha dyqanet e qendrës

tregtare. Në grupin aksesorë përfshihen aksesorët e ndryshëm si bizhuteritë, kozmetika, kancelaria dhe pajisjet për shtëpi. Në grupimin e tretë, të tjera do të grupohen të gjitha shitjet e dyqaneve të tjera që vendosen në qendrën tregtare por që stoku i tyre nuk është pronë e qendrës tregtare. Këto blerje janë përfshirë në analizën tonë në mënyrë që të analizojmë korrleacionin e tyre me kategoritë e tjera nëse ka. Në atributet e bazës së të dhënave përfshihen gjithashtu, moshë, seksi dhe qyteti i lindjes së klientit. Këto tre attribute do të na ndihmojnë gjithashtu në grupimin e klientëve. Duhet të theksojmë se atributi qytet nuk do të përdoret pasi sipas analizës mbi 90% e klienteve kishin Tiranën të deklaruar si qytet. Për këtë arsye nuk e kemi marrë në konsideratë këtë atribut.

### 5.1.2 Para-proçesimi i të dhënave

Siç e kemi theksuar disa herë gjatë analizës teorike, një nga pjesët më të rëndësishme në Data Mining (DM) është para-proçesimi i të dhënave. Fillimisht ne analizojmë të dhënat tona për të kuptuar nëse kemi mungesë të dhënash për ndonjë atribut specifik. Më poshtë po paraqesim një pasqyrë përmbledhëse të të dhënave tona dhe një printim të 5 rreshtave të para të bazës së të dhënave.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 9931 entries, 0 to 9930
Data columns (total 7 columns):
#   Column      Non-Null Count  Dtype
---  ---
0   CLIENTE    9931 non-null   int64
1   SESSO      9931 non-null   object
2   CITTA      9931 non-null   object
3   AGE        7603 non-null   float64
4   Clothes    7527 non-null   float64
5   Acc        4997 non-null   float64
6   Others     2932 non-null   float64
dtypes: float64(4), int64(1), object(2)
memory usage: 543.2+ KB
```

Table 5-1 Përmbledhëse e datasetit dhe 5 rreshtat e parë

	KLIENTI	GJINIA	QYTETI	MOSHA	Clothes	Acc	Others
0	27176	F	TIRANE	40	37243	0	0
1	28569	F	ABCD	44	26033	0	0
2	19383	M	TIRANE	37	1199	0	0
3	25150	F	ABC	39	1905	0	0
4	33536	M	TIRANE	39	48345	0	0

Siç e shohim nga raporti, kemi mungesë të dhënash në atributet e moshës dhe te grupet e ndryshme të blerjeve. Në rastin e moshës do të zëvendësojmë rreshtat bosh me të mesmen e moshës së klientëve në bazën e të dhënave. Ky është një vendim strategjik i yni ,pasi në këtë mënyrë nuk do të na impaktojë analizën. Në rastin e grupeve të blerjeve, një rresht bosh nënkupton mungesë blerjesh për atë klient në atë kategori. Për këtë arsye i zëvendësojmë rreshtat bosh me 0. Pasi kryejmë ndryshimet e mësipërme shohim që nuk kemi më reshta bosh

në bazën tonë të të dhënave. Hapi final është normalizimi i të dhënave në mënyrë që të shmangim impaktin më të madh të attributeve me vlera më të mëdha në modelet tona. Të dhënat standartizohen duke përdorur funksionin `StandardScaler()` të libraries `sklearn`. Në këtë mënyrë të gjitha atributet do të trajtohen me të njëjtën peshë.

### 5.1.3 Korrelimi midis attributeve

Së pari do të analizojmë korrelimin midis attributeve që do të përdorim për klasterizim. Tabelën e korrelimit e marrim duke përdorur funksionin `sns.heatmap` të librarisë `seaborn`. Rezultatet paraqiten në figurën 1.

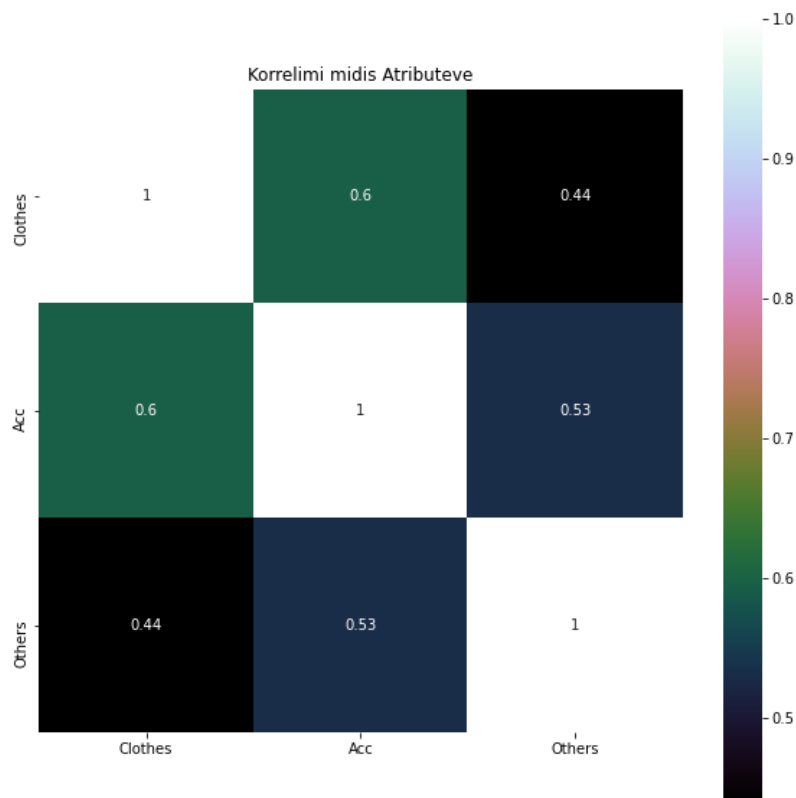


Figure 5-1 Korrelimi midis attributeve

Nga tabela e korrelacionit, rezulton se kemi një korrelacion të fortë midis blerjes së aksesorëve dhe blerjes së veshjeve me një vlerë 0.6. Kjo shpjegohet me faktin se vendodhja e aksesorëve nuk është e diferencuar nga vendodhja e veshjeve. Kjo na tregon se gjithashtu klientët që blejnë më shumë veshje, gjithashtu blejnë edhe aksesorë. Një lidhje interesante është korrelacioni më i lartë midis aksesorëve dhe të tjerave (others). Kjo shpjegohet me faktin se të gjitha dyqanet e tjera janë të vendosura në katin e parë ku kemi vetëm aksesorë dhe jo veshje. Pra klientët të cilët blejnë në dyqanet e tjera, blejnë më shumë aksesorë pasi e kanë më afër fizikisht. Kjo na tregon një lidhje të rëndësishme të ekspozimit fizik të mallit me blerjen e tij.

#### 5.1.4 Klasterizimi me K-Means

Fillimisht do të përdorim algoritmin K-Means për të klasifikuar klientët. Hapi i parë është të përcaktojmë numrin optimal të klasterave dhe për këtë qëllim do të përdorim tekniken Elbow të shpjeguar në pjesën teorike. Sipas teknikës Elbow do të vendosim në një grafik vlerën e inercisë, e cila është shuma e distancës euklidiane në katror të të dhënave nga qendrat e klasterave, përkundrejt numrit të klasterave. Më poshtë paraqesim grafikun e përfutur.

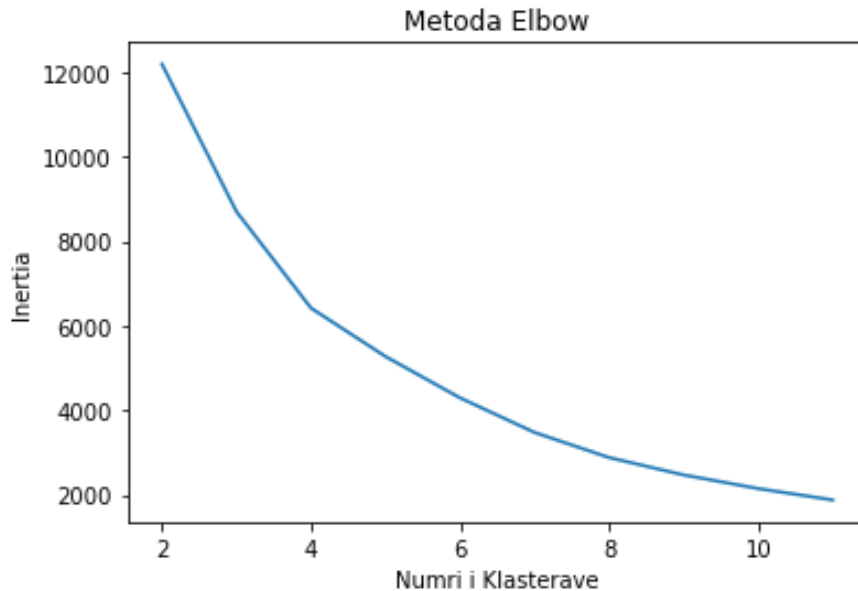


Figura 5-2 Metoda Elbow

Nga grafiku, vëme re se bërryli(elbow) vjen në numrin e klasterave të barabartë me 4. Në vazhdim të studimit tonë do të përdorim numrin 4 si numrin optimal të klasterave.

Në algoritmin K-Means, qendrat fillestare të klasterave zgjidhen në mënyre rastësore dhe më pas në mënyre iterative zgjidhet qendra e re cila bazohet në mesataren e distancës euklidiane në katror të pikave nga qendra aktuale. Ky proces vazhdon derisa sa konvergjojmë në qendrat optimale. Kemi dy faktorë të rëndësishëm në algoritmin K-means që mund të impaktojnë përzgjedhjen e Klasterave. Faktori i parë është përzgjedhja e qendrave fillestare i cili ka impakt direkt në konvergjencë dhe faktori i dytë është ndikimi i pikave të të dhënave periferike (outliers) në percaktimin e klasterave. Ne do të analizojmë vetëm impaktin e të dhënave periferike pasi ajo ka rëndësi për studimin tonë. Në vazhdim apikojmë algoritimin K-Means në bazën tonë të të dhënave dhe paraqesim rezultatet në një grafik në figurën me poshtë.

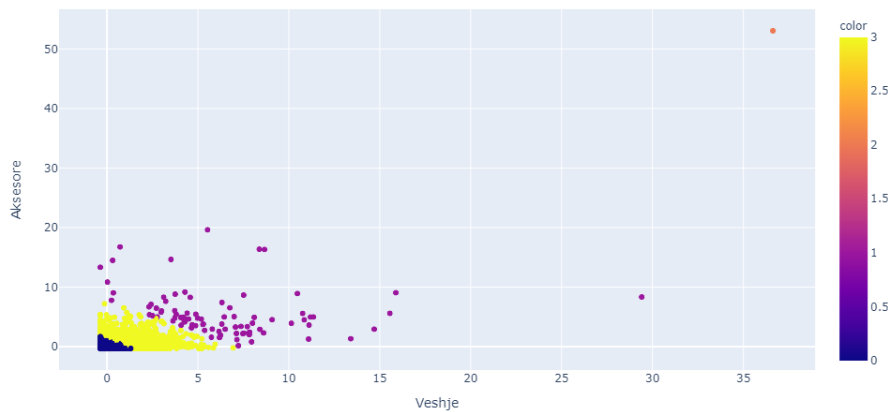


Figura 5-3 Klasterat e klienteve K-Means

Për ta bërë më të qartë vizualisht analizën kemi bërë një vizualizim në 2D të të dhënave. Çdo ngjyrë paraqet një klaster të caktuar të klientëve.

Në figurë dallojmë 4 klastera të ndryshëm por ajo që bën përshtypje më shumë është pika e të dhënave djathtas sipër. Ajo pikë është qartësisht një pikë periferike (outlier) por për algoritmin tonë është një klaster specifik. Pra kuptojmë se K-Means ndikohet nga të dhënat periferike duke alokuar qendra klasterash për to. Kjo është e kuptueshme nëse marrim në konsideratë faktin që algoritmi përdor mesataren euklidiane për të përcaktuar qendrën e klasterave dhe siç e kemi shpjeguar dhe në materialin teorik, mesatarja euklidiane është e ndikuar nga të dhënat periferike.

Një tjetër detaj që vëmë re është se grupimi është linear dhe nuk na jep shumë informacion për ndarjen sipas kategorive. Një anë positive e algoritmit është efikasiteti në kohë pasi na u deshën vetëm pak milisekonda për të aplikuar atë në bazën tonë të të dhënave.

### 5.1.5 Klasterizimi me K-Medoids

Në menyrë që të zgjidhim problematikat e sipërpërmendura do të përdorim një tjetër algoritëm, i cili nuk përdor distancën euklidiane nga qendra e të gjitha pikave por distancën në çift midis pikave. Ky proces ka një dizavantazh të dukshëm në lidhje me efikasitetin në kohë që do e analizojmë më poshtë por ka avantazhe të dukshme në lidhje me ndikimin nga të dhënat periferike. Algoritmi aplikohet në bazën tonë të të dhënave dhe rezultatet paraqiten në figurën më poshtë.



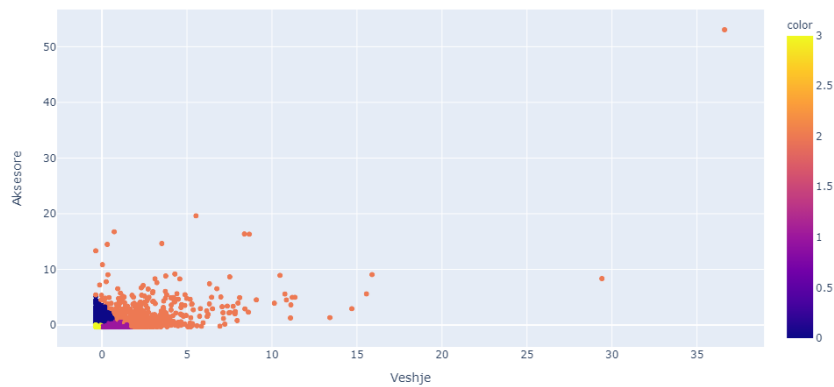


Figura 5-4 Klasterat e klienteve K-Medeoids

Siç e shohim dhe në figurë, rezultatet që marrim nga aplikimi I K-Medeoids janë më të mira se rezultatet e përfuara kur aplikojmë K-Means pasi në këtë rast ndikimi nga të dhënat periferike anulohet. Gjithashtu vëmë re se grupimi i klienteve bazohet më shumë në kategoritë e blerjeve se sa në rastin e K-Means dhe kjo është një veçori e dëshiruar nga ne. Këto avantazhe vijnë në këmbim të një kompleksiteti më të lartë në kohë.

#### 5.1.6 Klasterizimi me DBSCAN

Tani do të aplikojmë algoritmin dbscan i cili bazohet në dëndësinë e të dhenave. Algoritmi merr si parametra të tij rrezën eps dhe numrin minimal të pikave në fqinjësi min\_eps. Ne do të provojmë vlera të ndryshme të këtyre parametrave për të arritur në rezultate të kënaqshme. Më poshtë paraqesim grafikët me vlera të ndryshme të parametrave. Siç e shohim nga rezultatet e marra, në të gjitha rastet grupimet nuk janë shumë efikase.

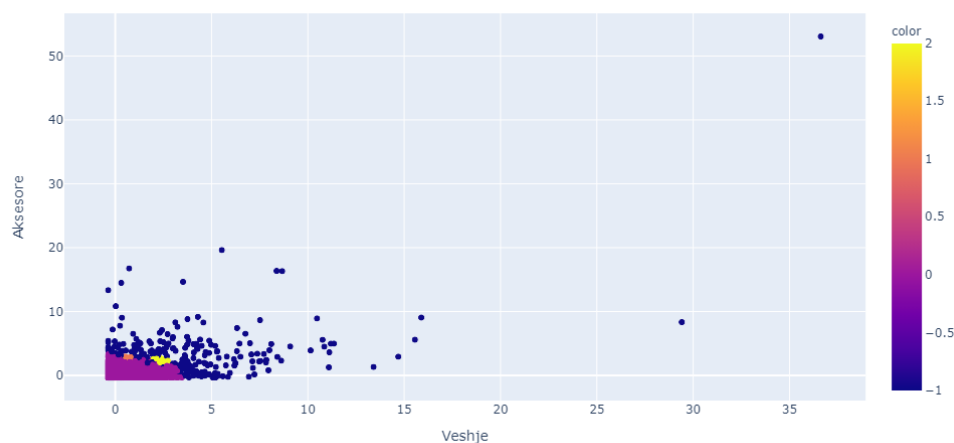


Figura 5-5: DBSCAN Min\_pts=7 eps=0.3

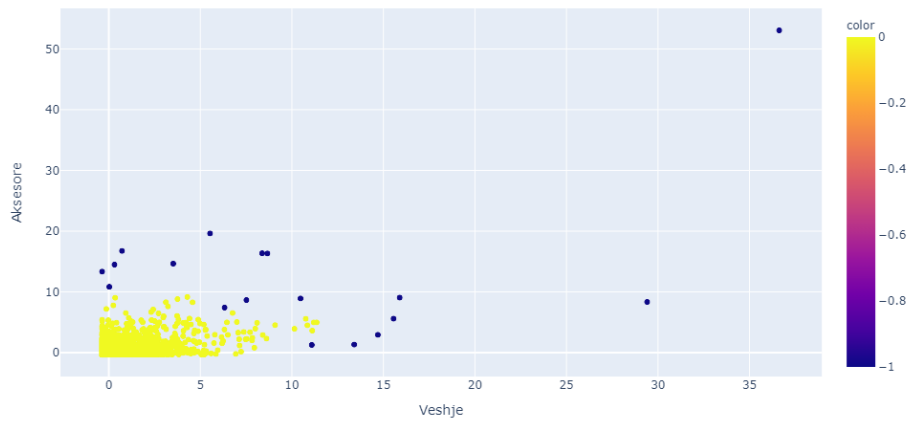


Figura 5-6 DBSCAN Min\_pts=7 ,eps=2

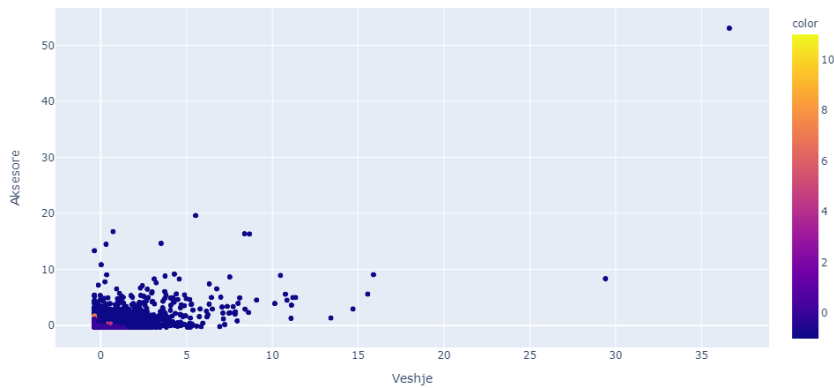


Figura 5-7 DBSCAN Min\_pts=7 ,eps=0.05

Siç e shohim, me rritjen e rrezes pikat grupohen në një klaster të vetëm, duke lënë jashtë vetëm pikat periferike (Figura 5-6). Me uljen e rrezes numri i klasterave rritet (Figura 5-7). Në figurën e parë kur kemi: Min\_pts=7 eps=0.3, kemi një numër klasterash të barabartë me 4 (marrim parasysh të dhënat periferike me label -1). Por në të gjitha rastet grupimet nuk janë efikase. Kjo shpjegohet me faktin se densiteti i të dhënave në bazën tonë është i ngjashëm, çka e bën jo efektive këtë algoritëm [44].

### 5.1.7 Algoritmi Hierarkik Aglomerativ

Së fundmi do të aplikojmë klusterizimin aglomerativ duke prodhuar një dendogram. Në klusterizimin aglomerativ dy klastera të afërt me njëri tjetrin bashkohen duke u nisur nga poshtë lart deri sa të gjitha të dhënat grupohen në një klaster. Numri i klasterave mund të dallohet në mënyrë intuitive duke ju referuar dendogramit por nuk mund të bazohemi 100% në të. Më poshtë paraqesim një dendogram të bazës sonë të të dhënave. Siç mund të vëmë re nga rezultatet është shumë e vështirë të deduktojmë rezultate nga dendogrami në rastin tonë duke qënë se numri i klientëve është i lartë.

Gjithashtu mund të themi se algoritmi hierarkik aglomerativ prodhon rezultate të ngjashme me algoritmin K-Means duke vuajtur të njëjtat probleme.

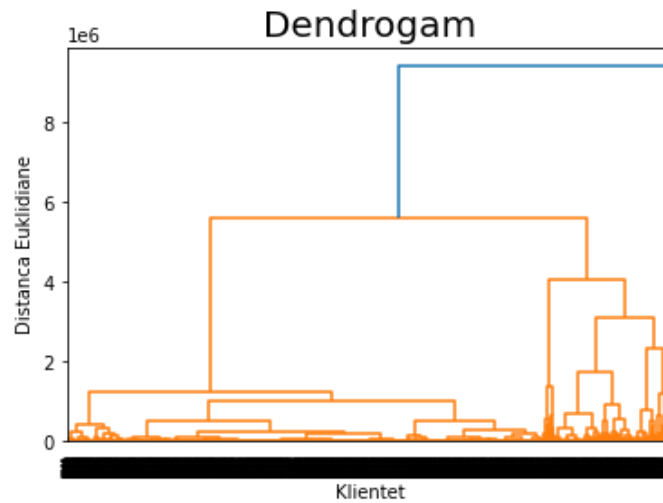


Figura 5-8 Dendogrami i të dhënave

#### 5.1.8 Rezultate të Klasterizimit

Siç pamë nga rezultatet më sipër rezultatet më të mira u morrën në rastin kur përdorëm K-medeoide. Në atë rast të dhënat tona nuk u ndikuan nga të dhënat periferike por edhe grupimi ishte sipas kategorive të blerjeve. Pamë se K-Means ndikohet nga të dhënat periferike dhe ndarja është lineare ndërmjet të gjitha kategorive të blerjeve. Algoritmi DBSCAN i cili bazohet te dendësia e të dhënave nuk është i përshtatshëm për të dhënat tona që kanë dendësi uniforme. Gjithashtu pamë se algoritmi hierarkik aglomerativ dhe në veçanti dendogrami nuk është shumë i përshtatshëm për bazën tonë të të dhënave. Në lidhje me kompleksitetin në kohë, rezultatet i kemi paraqitur në tabelën më poshtë. Siç shohim, algoritmi K-Means dhe DBSCAN janë më eficientë në kohë ndërsa K-Medeoide dhe algoritmi aglomerativ kanë kompleksitet më të lartë në kohë. Nga kjo deduktojmë se në raste kur baza e të dhënave është e madhe, K-Means është më i përshtatshëm për përdorim. Për të shmangur efektin e të dhënave periferike në këto raste, një zgjidhje do të ishte pastrimi i bazës së të dhënave nga këto pika. Në rastin e bazave të të dhënave jo sh të mëdha, mund të përdoret K-Medeoide i cili jep rezultate të kënaqshme si në rastin tonë.

Tabela 5-2 Performanca në kohë e modeleve të klasterizimit

Algoritmi	Numri i iterimeve	Koha në Sekonda
K-Means	10	3.34
K-Medeoide	10	190
Dendrogram/Agglomerative	1	1440
DBSCAN	10	7.41

Nga ana funksionale rezultatet e marra nga klasterizimi u dërguan në departamentin e Marketing. Për të ilustruar disa aspekte interesante të studimit më poshtë një grafik i cili tregon

se si janë të ndara blerjet për grupe klientësh. Mund të shohim se 31% të blerjeve të klientëve të regjistruar kryhen nga vetëm 2% e klientëve dhe 65% e tyre kryejnë vetëm 12% të blerjeve.



Figura 5-9 Rezultatet e ndarjes në klastera

## 5.2 Modelet e klasifikimit

### 5.2.1 Eksplorimi i bazës së të dhënave

Baza e të dhënave përbehet nga 3797 rekorde të cilat përshijnë blerjet për tre-mujorin e parë të vitit 2020 të klientëve të regjistruar në qendrën tregtare.



Figura 5-10 Ndarja e klientëve në klasa

.Atributet që do të përdoren për parashikim janë: seksi, mosha, vlera totale e blerjeve tremujore, kategoria e preferuar, sasia dhe blerjet e vitit të kaluar për klientet e përsëritur (kjo vlerë është 0 në rastin e klientëve të rinj). Klientët janë të grupuar në 3 klasa, klasa 0–vlerë totale blerjesh vjetore për vitin 2020 deri në 500 USD, klasa 1- vlerë totale blerjesh vjetore nga 500 deri në 1000 USD dhe klasa 2- vlerë totale blerjesh vjetore mbi 1000 USD. Së pari ne do të krijojmë një histogram të klientëve në mënyrë që të kuptojmë se si është ndarja në klasa.

Nga grafiku me figurën 5-10, kuptojmë se kemi një mosbalancë të moderuar të klasave duke qënë se klasa 1 përbën 64% të klientëve. Por gjithsesi në këtë studim nuk do të kryejmë teknika balancimi të të dhënave. Këto teknika do të kryhen në punët në vazhdim.

### 5.2.2 Para-proçesimi i të dhënave

Si në rastin e klasterizimit, para se të aplikojmë modelet e klasifikimit do të bëjmë para-proçesimin e të dhënave.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3797 entries, 0 to 3796
Data columns (total 9 columns):
#   Column      Non-Null Count  Dtype
---  -
0   CLIENTE    3797 non-null   int64
1   SESSO      3797 non-null   object
2   CITTA      3797 non-null   object
3   CAT        3797 non-null   int64
4   AGE        2878 non-null   float64
5   Val        3797 non-null   int64
6   Sasi       3797 non-null   int64
7   Prev       3002 non-null   int64
8   Label      3797 non-null   int64
dtypes: float64(1), int64(6), object(2)
memory usage: 267.1+ KB
```

Tabela 5-3: Përmbledhëse e datasetit dhe 5 rreshtat e parë

	Klienti	Gjinia	Qyteti	KAT	MOSHA	Val	Sasi	Prev	Label	Gjinia
0	232	F	TIRANE	950	40.0	162	5	5765	3	0
1	246	F	TIRANE	980	47.0	211	2	495	3	0
2	254	F	TIRANE	930	42.0	565	18	869	3	0
3	309	M	TIRANË	940	31.0	1771	19	10122	3	1
4	368	M	TIRANE	980	62.0	187	6	497	2	1

Nga përmbledhja e të dhënave shohim se kemi mungesë të dhënash për moshën dhe vitin para-ardhës në disa rreshta. Për moshën, do të përdorim të njëjtën teknike që kemi përdorur në rastin e klasterizimit duke zëvendësuar vlerat bosh me të mesmen e moshës në bazën e të dhënave. Për vlerën e vitit para-ardhës do ta zëvendësojmë vlerën bosh me 0 pasi vlera bosh nënkupton një klient të ri. Duke qënë se modelet që do të aplikojmë veprojnë mbi të dhënat numerike dhe atributi i seksit përmban shkronja, do të bëjmë enkodimin e atributit duke përdorur funksionin

`preprocessing.LabelEncoder()` te librarisë sklearn.Shkronja F(femër) do të kthehet në 0 dhe shkronja M(Mashkull) do të kthehet në numrin 1.

Baza e të dhënave do të ndahet në 80% që do të përdoret për trajnimin e modelit dhe 20% për testimin e modelit të klasifikimit.Ndarja kryhet duke përdorur funksionin `train_test_split` të librarisë sklearn.

### 5.2.3 Modeli KNN

Modeli KNN (K-Nearest Neighbour) përdor kampionimin e të dhënave për të vlerësuar densitetin duke përdorur vlerësimin e densitetit sipas fqinjëve më të afërt. Modeli merr si parametër numrin e fqinjëve.Ne do të eksperimentojmë me numra të ndryshëm fqinjësh deri sa të arrijmë në rezultatet optimale.Elementi mbi të cilin do të vleresohet performanca e modelit do të jetë saktësia (accuracy). Gjatë fazës së parë modeli do të trajnohet me 80% të të dhënave të marra nga baza e të dhënave dhe më pas do të testohet në 20% e mbetur. Saktësia gjendet duke krahasuar klasën e përcaktuar nga modeli i trajnuar me klasën reale të marrë nga 20% test i të dhënave.Më poshtë do të paraqesim saktësinë për numër të ndryshëm fqinjësh për të gjetur numrin optimal të fqinjëve.

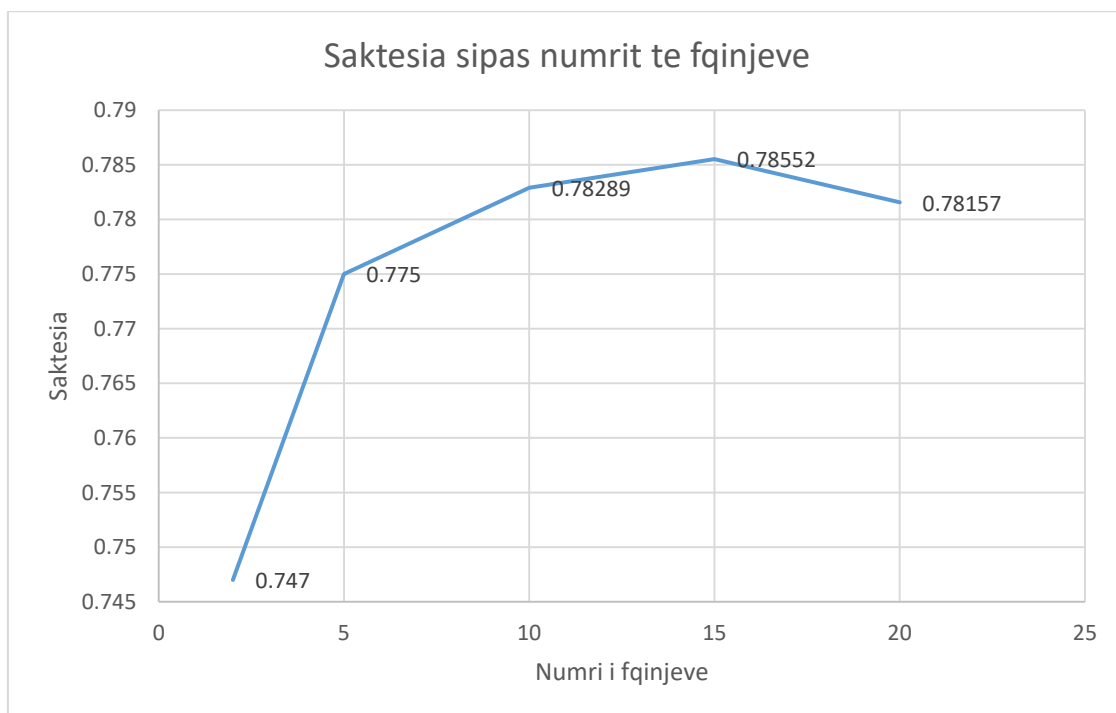


Figura 5-11: Saktësia për numer te ndryshem fqinjesh

Siç mund të dallohet nga grafiku, ne arrijmë vlerën më të lartë të saktësisë për numrin e fqinjëve të barabarte me 15. Saktësia për këtë numër fqinjësh është 0.7856 . Në mënyrë që të kuptojmë rezultatet tona janë të vlefshme duhet të kryejmë procesin e cross-validation duke përdorur funksionin `KFold`, `cross_val_score`. Proçesi i Cross Validation e ndan databazën e testimit në disa pjesë , kryhen testime të ndryshme mbi to dhe rezultatet përdoren për të kuptuar vlefshmërinë e vlerës se përftuar të saktësisë. Ne do të përdorim në rastin tonë 5 pjesë të ndryshme ndarjeje.

Tabela 5-4 Saktësia pas Cross Validation

Numri i ndarjeve të databazës së testit	Saktësia
1	0.751
2	0.777
3	0.779
4	0.771
5	0.799

Siç vëme re nga tabela, rezultatet janë të ngjashme me saktësinë tonë duke na i konfirmuar vlerën e përfituar të saktësisë. Së fundmi duhet të prodhojmë një matricë të saktësisë (confusion matrix) për të kuptuar saktësinë për klasat e ndryshme të klientëve tanë. Për të realizuar këtë përdorim funksionin *plot\_confusion\_matrix*. Më poshtë parqesim matricën dhe interpretojmë rezultatet.

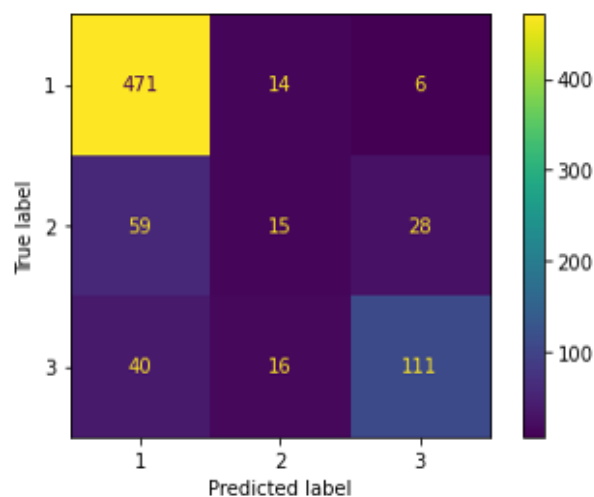


Figura 5-12 Matrica e Saktësisë

Siç vëmë re nga matrica e saktësisë dhe siç pritej kemi një saktësi të lartë për klasat 1 dhe 3 por vuajmë në saktësinë e klasës së ndërmjetme 2. Besojmë se kjo performancë e dobët në parashikimin e klasës së ndërmjetme vjen nga rangi i vogël dhe numri më i vogël i klientëve në atë klasë. Por pavarisht kësaj mund të themi se rezultat e përftuara janë të kënaqëshme duke u nisur nga baza e të dhënave e vogël që ne kemi.

#### 5.2.4 Support Vector Machines

Support Vector Machines (SVMs) është një metodë klasifikimi e bazuar në maksimumin marxhinal të diskriminuesve linear, që ka si qëllim kryesor gjetjen e planit maksimal që

maksimizon largësinë ose marzhin mes klasave. Modeli i dytë parashikues që do të krijojme është një model që bazohet në SVM.

### 5.2.5 Eksplorimi i të dhënave dhe para-procesimi

Baza e të dhënave e përdorur dhe elementet e paraprocesimit janë të njëjta me rastin e KNN.

### 5.2.6 Modeli SVM

Pas realizimit të analizave më lart, aplikohet algoritmi i SVM. Në fillim aplikohet modeli pa bërë shkallëzimin e të dhënave me qëllim theksimin e rëndësisë së standartizimit të të dhënave në aplikime SVM. 80% e të dhënave janë përdorur për datasetin e zhvillimit, ndërsa 20% e tyre janë përdorur për qëllime testimi. SVM përdoret së pari me një bërthamë lineare për të cilën regjistrohet saktësia.

Mund të vërehet se koha e ekzekutimit është jashtëzakonisht e afektuar dhe u deshën afërsisht 29 minuta për të aplikuar modelin. Në të dhënat e panormalizuara kemi një saktësi prej 76,71%. Më tej, bëhet normalizimi i të dhënave duke përdorur funksionin Standart Scaler dhe ri-aplikojmë algoritmin. Në këtë moment marrim një saktësi prej 80.79% dhe koha e ekzekutimit është pothuajse e menjëhershme.

Tabela 5-5 Saktësia në varësi të normalizimit të të dhënave

	Saktësia	Koha e ekzekutimit
<b>Të dhëna të pastandardizuara</b>	76.71%	28min 38s
<b>Të dhëna të standartizuara</b>	80.79%	250 ms

Në mënyrë që të vërtetojmë rezultatet tona, ne kryejmë cross validation në model me 10 pjesë. Rezultatet janë përshkruar në tabelën 5-6 dhe mund të shihet se ato janë mjaft në përputhje me vlerën tonë, kështu që mund ta konsiderojmë të validuar ose vërtetuar modelin.

Tabela 5-6 Cross-Validation në SVM

Pjesa	Saktësia
Pjesa 1	0.7730263
Pjesa 2	0.7796053
Pjesa 3	0.8026316
Pjesa 4	0.8026316
Pjesa 5	0.7960526
Pjesa 6	0.7960526
Pjesa 7	0.7927632
Pjesa 8	0.7788779
Pjesa 9	0.7821782
Pjesa 10	0.7887789
Mesatarja	0.7892598



Pas validimit të modelit bëjmë kontroll të matricës së saktësisë, për të kuptuar se si performon modeli për secilën klasë në veçanti.

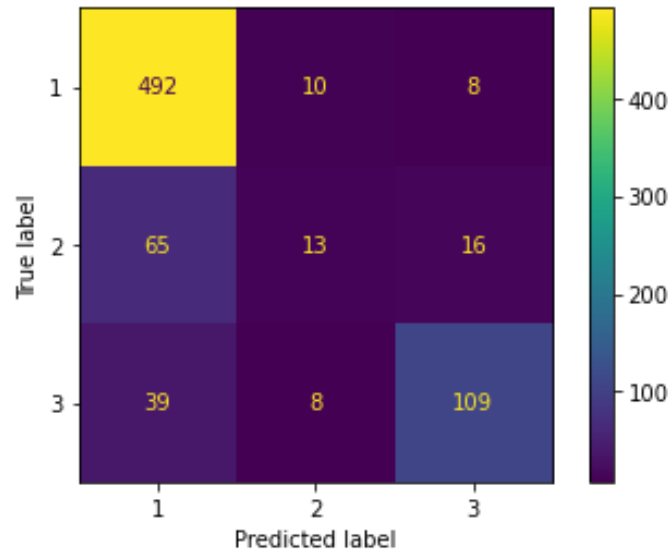


Figura 5-13 Matrica e saktësisë për SVM

Ashtu sikurse pritej, modeli ynë punon më mirë në kategorinë e parë dhe të tretë si dhe çalon më shumë në pjesën e mesme të tij. Ne mundemi të parashikojmë një klasë “2” të saktë 13 herë. Modeli na ofron një precizion të lartë prej 82% në parashikimin e klasës së parë dhe të tretë. Vlen të anaizojmë modelin tonë dhe për bërthama(kernels) të ndryshme. Në këtë mënyrë do të validojmë ose jo bërthamën lineare që përdorëm. Më poshtë është paraqitur kontrolli i bërë për saktësinë bazuar në tipe të ndryshme bërthamash (kernels).

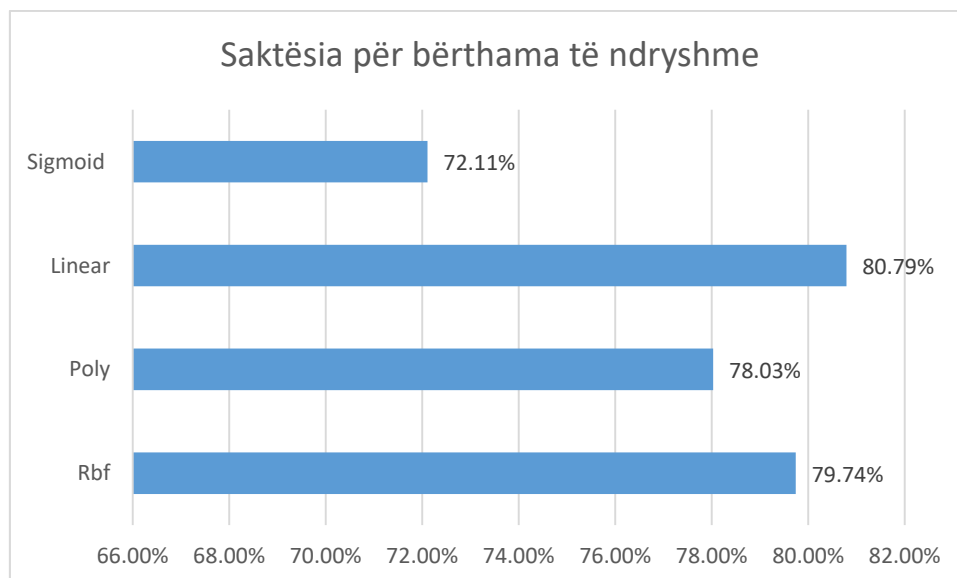


Figura 5-14 Saktësia për tipe të ndryshme bërthamash

Si përfundim, mund të themi që ne marrim rezultate më të mira nëse përdorim tipin e bërthamës lineare në datasetin tonë. Në këtë seksion, ndërtuam një model të bazuar në SVM për të

parashikuar blerjet e klientëve. Edhe pse madhësia e të dhënave ishte e kufizuar, saktësia e parashikimit rezultoi mjaft e mirë. Është vërejtur se shkallëzimi i të dhënave është mjaft i rëndësishëm para aplikimit të algoritmit SVM, duke rezultuar në një kohë ekzekutimi jashtëzakonisht të shkurtër dhe saktësi më të mirë. Gjithashtu testuam modelin tonë për lloje të ndryshme të bërthamave dhe morëm rezultatet më të mira për bërthamën lineare. U hasën disa vështirësi në parashikimin e klasave të mesme. Në të ardhmen, studimi do të kryhet për grupe më të mëdha të të dhënave dhe do të zgjerohet në dimensionet e ndryshme. Gjithashtu, teknikat si marrja e kampioneve ose SMOTE do të aplikohen për të zvogëluar efektin e çekuilibrit mes klasave [45].

### 5.2.7 Rrjetat Neurale

Modeli i tretë parashikues që do të krijojmë, është një model që bazohet në rrjetat neurale. Rrjetat neurale janë algoritmat që përdoren më së tepërmi në vitet e fundit për shkak të performancës së tyre të lartë. Rrjetat neurale mund të përshkruhen nëpërmjet një grafi të peshuar të drejtuar  $graph=(V,E)$ , ku çdo nyje  $v_i \in V$  përfaqëson një neuron, dhe çdo degë  $(v_i, v_j) \in E$  përfaqëson një lidhje sinaptike nga  $v_i$  në  $v_j$ . Peshat e çdo dege  $w_{ij}$  përfaqëson fuqinë sinaptike. Rrjetat neurale karakterizohen nga lloji i funksionit të aktivizimit që përdoret për të gjeneruar rezultatin në dalje dhe nga arkitektura e rrjetit që përcakton mënyrën se si lidhen nyjet. Ne do të krijojmë një model i cili do të ketë parametrat si në tabelën 5-7. Modeli ynë do të ketë 6 nyje në shtresën e hyrjes që përfaqësojnë 6 atributet parashikues. Gjithashtu do të kemi vetëm një shtresë të ndërmjetme, por do të variojmë numrin e nyjeve në këtë shtresë për të arritur rezultate optimale. Numri i nyjeve në shtresën e jashtme do të jetë 4, pasi kemi 3 klasa (1,2,3). Arsyeja përse e vendosim në 4 numrin e shtresave në dalje është pasi python numëron nga 0 dhe nëse duam të paraqesim klasat nga 0 deri në 3 në dalje na duhen 4 nyje. Nëse do të emërtonim klasat tona nga 0 deri në 2, do të kishim nevojë vetëm për 3 nyje në dalje. Përcaktojmë një *learning rate* të barabartë me 0.01. Kujtojmë se *learning rate* është parametri që përcakton se sa shpejt do të levizim në rastin e përhapjes së prapme kur përdorim SGD (Stochastic Gradient Descent) për optimizimin e humbjeve. Do të përdorim një numër iterimesh (epochs) nga 20000 deri në 500000. Lloji i humbjes që do të përdorim është humbja Cross Entropy. Do të përdorim disa funksione aktivizimi dhe do të analizojmë performancën e gjithësecilit. Për optimizimin e modelit do të përdorim SGD. Modeli do të ndërtohet me Pytorch. Teknikat e paraprosesimit që përdoren janë të njëjtat që përdoren në rastin e modelit në KNN.

Tabela 5-7 Parametrat e modelit NN

Numri i nyjeve në shtresën e hyrjes= <b>6</b>
Numri i nyjeve në shtresën e ndërmjetme= <b>[5:100]</b>
Numri i nyjeve në shtresën e jashtme= <b>4</b>
Learning Rate = <b>0.01</b>
num_epoch = <b>20000:50000</b>
Funksioni i Humbjes : <b>Cross Entropy Humbja</b>
Funksioni I Aktivizimit: <b>Sigmoid</b>
Model Optimizer : <b>SGD</b>

Fillimisht do të eksperimentojmë me numrin e nyjeve në shtresën e ndërmjetme. Duhet të kemi parasysh që mund të përdorim edhe më shumë se një shtresë të ndërmjetme por gjykojmë se për parashikimin tonë mjafton një shtresë dhe mund të alternojmë numrin e nyjeve në të. Rezultatet e humbjeve dhe saktësisë për numër të ndryshëm nyjesh në shtresën e ndërmjetme i paraqesim më poshtë.

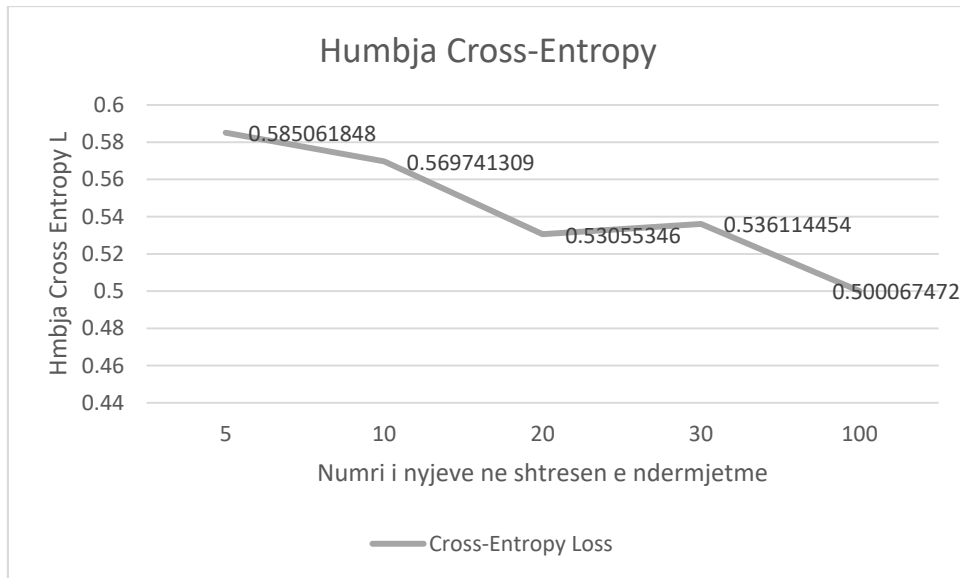


Figura 5-15 Humbja Cross Entropy për numër të ndryshëm nyjesh në shtresën e ndërmjetme

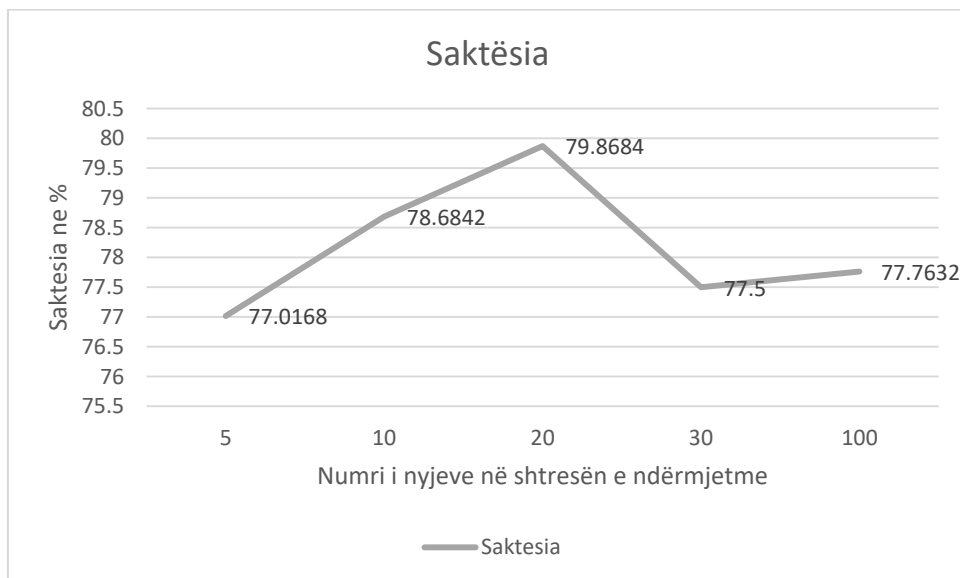


Figura 5-16 Saktësia për numër të ndryshëm nyjesh në shtresën e ndërmjetme

Siç mund të shohim, rezultatin më të mirë të saktësisë 79.864 e arrijmë për numrin 20 të nyjeve në shtresën e ndërmjetme. Gjithashtu vlera e humbjeve për këtë numër nyjesh është e ulët. Pra

do të zgjedhim numrin 20 për numrin e nyjeve oln ne modelin tonë. Gjithashtu nga eksperimentet përcaktojmë vlerën e learning rate 0.01 si vlerën optimale të lr. Modeli konvergjon pas 50000 iterimesh, pra nuk kemi nevojë ta testojmë për një num\_epochs më të madh .Pasi kryejmë procesin e cross validation si në rastin e KNN, vlerat e marra e konfirmojnë saktësinë e modelit tonë. Së fundmi duhet të përcaktojmë nëse funksioni i aktivizimit që zgjedhëm, në këtë rast Sigmoid është funksioni optimal apo jo. Për të verifikuar këtë do të ndryshojmë funksionin e aktivizimit dhe do të eksperimentojmë me 2 funksione të tjera përkatesisht ReLU dhe TanH. Analizën teorike të tyre e kemi kryer në kapitujt pararendës. Në grafikët më poshtë paqraesim grafikët përmbledhës të rezultateve.

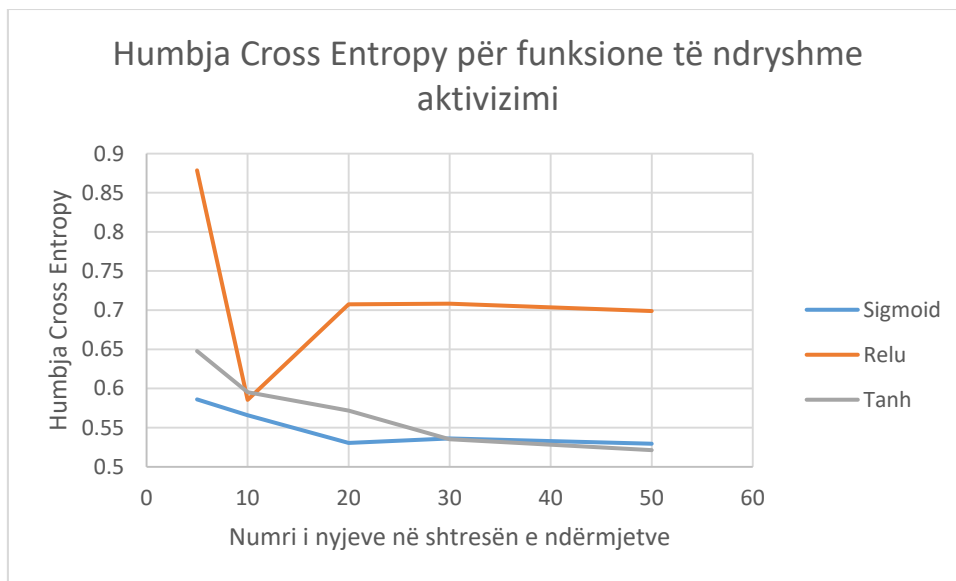


Figura 5-17 Humbja për funksione të ndryshme aktivizimi

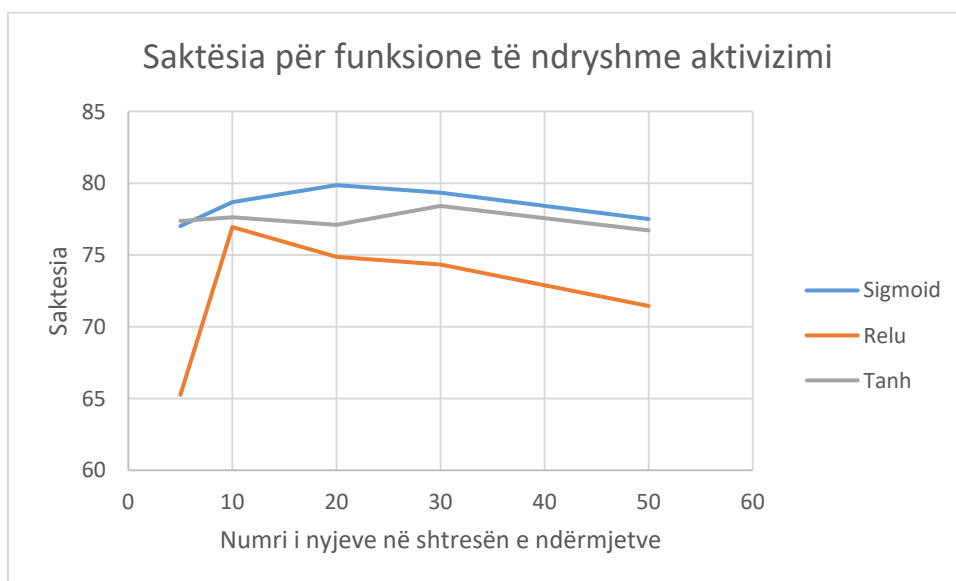


Figure 5-18 Saktësia për funksione të ndryshme aktivizimi

Sic mund te shohim nga rezultatet ,funksioni i aktivizimit Sigmoid performon më mirë se të dy funksionet e tjerë si në terma humbjeje, edhe në terma saktësie të parashikimit.

Hapi final do te jetë të prodhojmë matricën e saktësisë (confusion matrix) për modelin.

Matrica si më poshtë:

$$\begin{pmatrix} [480, 25, 5], \\ [60, 18, 16], \\ [25, 13, 118] \end{pmatrix}$$

Si edhe në rastin e KNN,kemi rezultate më të mira për klasat 1 dhe 3 dhe rezultate jo shumë të mira për klasën e mesit 2.Siç mund të dallojmë nga rezultatet,ne do të marrim rezultatet më te mira kur përzgjedhim Sigmoid si funksion aktivizimi, pra e konfirmojmë modelin tonë.Rezultatet e marra nga modeli i bazuar te rrjetat neurale jep rezultate lehtësisht më të mira se në rastin e KNN.Kjo i atribuohet edhe numrit të vogël të rekordeve në bazen tonë të të dhënave pasi kemi parasysh që fuqia reale e rrjetave neurale vjen kur kemi baze të gjërë të dhënash.Por duke u nisur nga madhësia e bazes sone te te dhenave ,rezultatet jane te kenaqshme.

## 5.3 Analiza e serive kohore

### 5.3.1 Modeli Prophet

Etapa e tretë e studimit tonë do të bazohet në propozimin e një modeli të modifikuar Prophet për parashikimin e shitjeve të qendrës tregtare. Në ndryshim nga seksioni i dytë, në të cilën kryem parashikimin për klientët duke u nisur nga shitjet 3-mujore, në këtë rast do të parashikojmë shitjet totale të qendrës tregtare duke marrë në konsideratë elementin kohë. Modeli do të bazohet në seritë kohore dhe veçanërisht në algortimin Prophet,te analizuar ne pjesen teorike.Performanca e modelit bazë dhe atij të modifikuar do të matet në bazë të gabimit absolut mesatar në parashikim MAE .Performanca e modelit të modifikuar do të matet në parashikime 6 mujore nëpërmjet gabimit absolut mesatar në përqindje MAPE. Të dyja metrikat janë shpjeguar në pjesën teorike.

### 5.3.2 Eksplorimi i te dhenave

Për këtë studim baza jonë e të dhënave do të konsistojë në 3271 rreshta që përmbajnë shitjet në qendrën tregtare nga viti 2011 në vitin 2019. Të dhënat i kemi agreguar nëpërmjet skripteve në SQL(Aneks 1) për të përftuar një dataset final i cili përmban vetëm 2 kolona: datën dhe vlerën. Printojmë një përmbledhje të bazës së të dhënave për të kuptuar atributetet.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3271 entries, 0 to 3270
Data columns (total 2 columns):
#   Column  Non-Null Count  Dtype
---  -
0   0       3271 non-null    object
1   1       3271 non-null    int64
dtypes: int64(1), object(1)
memory usage: 51.2+ KB
```

Siç mund të shohim kemi vetëm 2 kolona, përkatësisht datën dhe shitjet totale në USD për atë datë. Po printojmë dhe 5 kolonat e para të bazës së të dhënave për qartësi.

Tabela 5-8 5 rreshtat e parë të bazës së të dhënave

	ds	y
0	1/3/2011	25456
1	1/4/2011	36109
2	1/5/2011	46267
3	1/6/2011	46600
4	1/7/2011	45780

Gjithashtu kemi krijuar një bazë të dhënash me festat zyrtare të këtyre 9 vjetëve të cilën do ta importojmë më pas në model. Importimi i të dhënave të festave është i rëndësishëm pasi në ato data specifike ne do të kemi ndryshime të menjëhershme të shitjeve ditore.

Tabela 5-9 5 rreshtat e parë të databazës së festave

	holiday	ds	lower_window	upper_window
0	holiday season	12/25/2019	-1	0
1	holiday season	11/28/2019	-1	0
2	holiday season	3/7/2019	-1	0
3	holiday season	2/14/2019	-1	0
4	holiday season	12/25/2018	-1	0

Në bazën e të dhënave të festave kemi edhe dy dritare kufizuese të cilat përcaktojnë nëse duhet të ketë impakt të festave para ose/edhe pas ditës ekzakte dhe sa do të jetë ky impakt. Në rastin tonë specifikojmë impaktin vetëm në ditën para festës pasi në ditën pas, zakonisht nuk kemi rritje të fluskit të blerjeve. Nga ana tjetër shohim se baza jonë kryesore e të dhënave nuk ka mungesa, pra nuk kemi nevojë të bëjmë plotësime në të. Së fundmi paraqesim një grafik të shitjeve në datasetin tonë për të kuptuar ecurinë. Vëmë re se në pika të caktuara kemi rritje të menjëhershme të cilat mund të jenë ditë festive apo promovionale. Me anë të modelit tonë do të mundohemi të parashikojmë edhe këto rritje dhe ndryshime të menjëhershme. Një pjesë e problemit zgjidhet me futjen e të dhënave festive dhe pjesa tjetër duke përcatuar sezonalitetet dhe duke modifikuar ndikimet e tyre në parashikim.

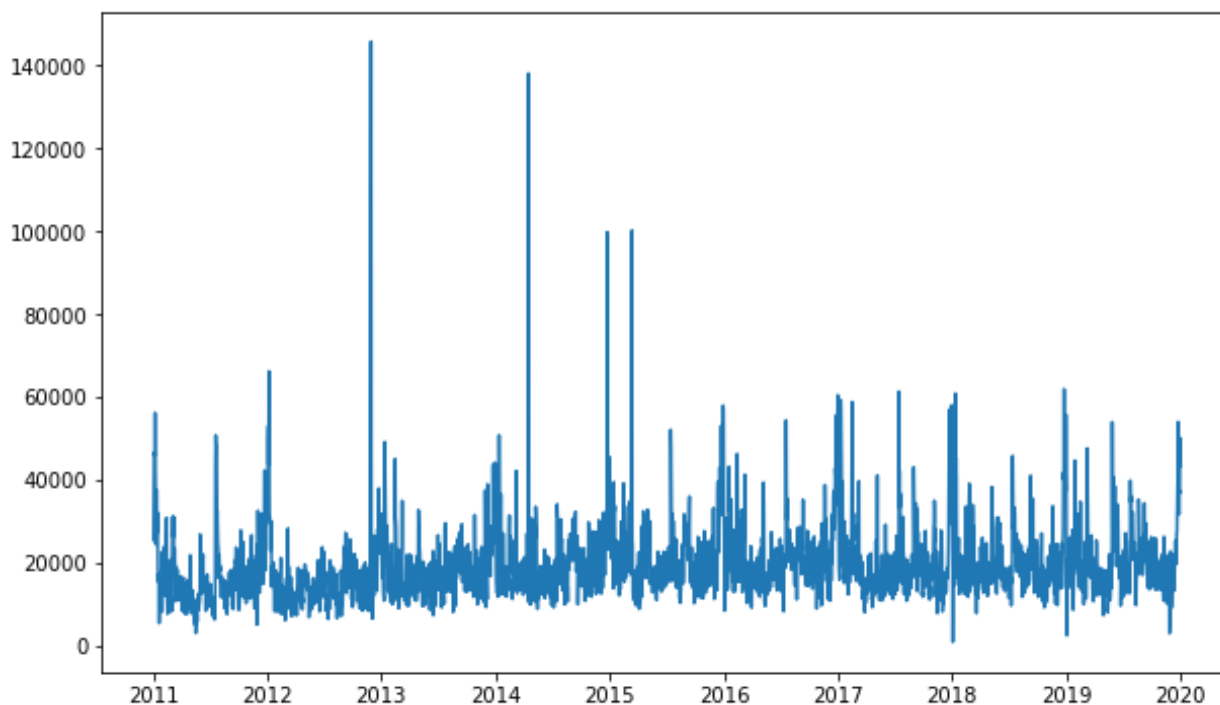


Figure 5-19 Ecuria e shitjeve në vitet 2011-2019

### 5.3.3 Aplikimi i modelit standart dhe të modifikuar

Hapi pasardhës konsiston në aplikimin e modelit në të dhënat tona. Fillimisht do të aplikojmë modelin standart dhe pastaj modelin e modifikuar. Në tabelën më poshtë do të përmbledhim ndryshimet që kemi kryer në model për të rritur saktësinë e tij. Përpos shtimit të ditëve festive si parametra të modelit do të ndryshojmë sezonalitetet standarte dhe rendin fourier të tyre. Rendi Fourier përcakton se sa shpejt ndryshon sezonaliteti. Rendi standart për sezonalitetin vjetor është 10 dhe për atë javor është 3. Ne kemi ndryshuar rendin fourier për sezonalitetin javor në 20 duke qënë se kemi ndryshime të shpejta të shitjeve në fundjavë. Nga ana tjetër kemi krijuar një sezonalitet të ri 2 mujor pasi kemi vlerësuar se ai është periodiciteti mesatar i promocioneve të qëndrës tregtare.

Tabela 5-10 Modeli i modifikuar Prophet

Modeli Baze	Modeli i modifikuar
Sezonaliteti:	Sezonaliteti:
Javor/Vjetor	1. Vjetor
Rritja : Lineare	2. Javor fourier_order=20
	3. 2mujor, fourier_order=5
	holiday_season, holiday_prior_scale=25, lower_window= -1
	Rritja : Lineare

Perpara se të bëjme krahasimin e 2 modeleve aplikojmë modelin e modifikuar në të dhënat tona dhe rezultatet i paraqesim më poshtë.

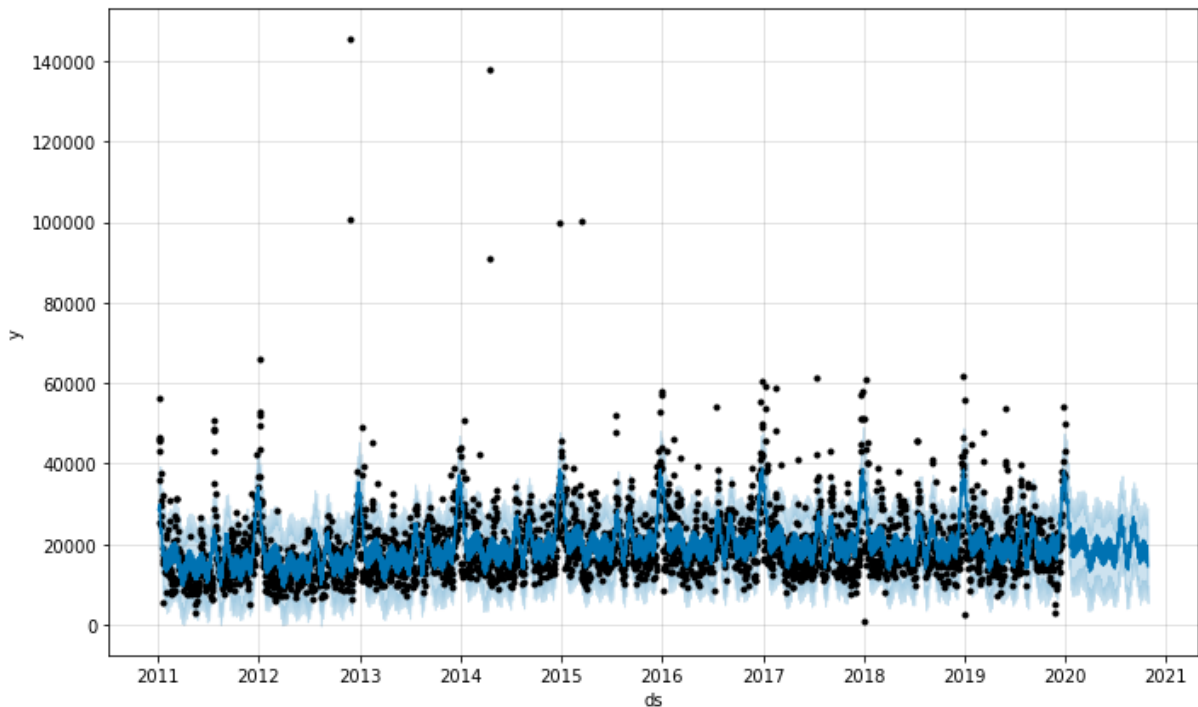


Figura 5-20 Parashikimi i shitjeve me anë të modelit të modifikuar Prophet

Pikat e zeza përfaqsojnë të dhënat reale dhe vija blu përfaqson parashikimin tonë. Siç mund të dallohet grafikisht, përputhja është shumë e mirë. Pjesa e fundit në të djathtë paraqet parashikimin e shitjeve për vitin 2020. Për të krahasuar modelin e modifikuar me modelin bazë do të përdorim vlerën e MAE (Mean Absolute Error), vlera e mesatare e gabimit absolut në të dyja rastet në parashikimin e shitjeve për 6-mujorin e dytë të 2019. Rezultatet për secilin model i paraqitim më poshtë:

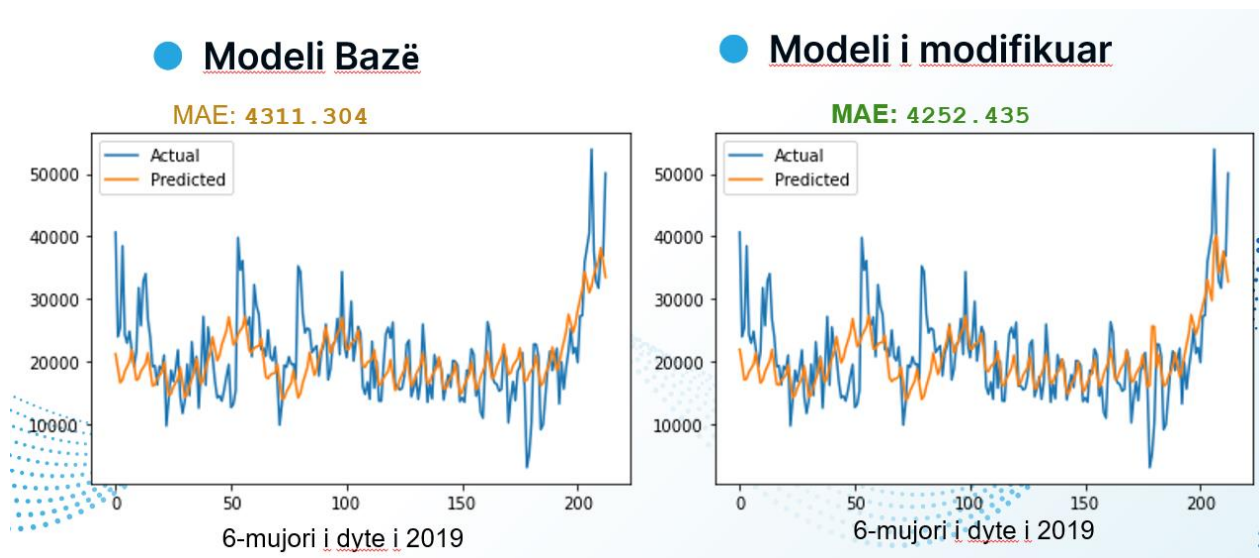


Figure 5-21 Krahasimi i modeleve për 6 mujorin e dytë të vitit 2019

Siç vëmë re, modeli bazë ka një performancë të mirë edhe pa modifikimet tona. Modifikimet tona japin një përmirësim me 1.4% të vlerës së MAE.



Hapi final është aplikimi i teknikës së cross-validation për të konfirmuar rezultatet tona. Për të kryer këtë proces do ta trajnojmë algoritmin për 2000 ditë dhe do të bëjmë një parashikim për 180 ditët në vazhdim çdo 180 dite. Në total do të bëjmë 7 parashikime dhe do të regjistrojmë vlerat e MAPE (Mean Absolute Percentage Error), e cila është MAE në përqindje. Rezultatet paraqiten më poshtë:

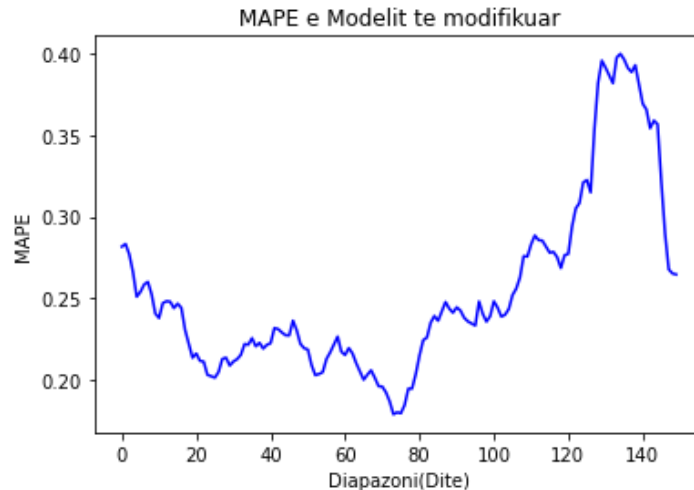


Figura 5-22 Vlerat e MAPE pas Cross Validation

Nga rezultatet mund të themi se modeli na jep parashikime shumë të mira deri në 90 ditë (MAPE<25%) dhe pastaj saktësia perkeqësohet në diapazonin 90-140 dite. Pas 140 ditësh, gabimi bie përsëri. Kjo karakteristikë mund ti atribuohet ndryshimet të sezoneve promocionale në vitet 2018/2019.

Së fundmi do të pasqyrojmë edhe trendet në shitje pasi është një informacion i vlefshëm për qëndrën tregtare. Rezultatet e përfuara nga analiza e serive kohore na dhanë rezultate shumë të mira. Parashikimet janë shumë të ngjashme me të dhënat reale dhe ky model parashikues mund të shfrytëzohet nga qendra për parashikimet e shitjeve. Parashikimet janë optimale për diapazonin 3 mujor dhe kanë një rritje të gabimit në diapazonin 90-140 ditë. Pas këtij diapazoni, gabimi bie përsëri në vlera të kënaqshme. Kjo rritje e menjehershme mund ti atribuohet ndryshimit në sezonet promocionale në vitet 2018/2019. Një zgjidhje do të ishte që ndryshimet të futeshin manualisht në algoritëm si pika ndryshimi por kjo do të analizohet dhe punohet në të ardhmen [46].

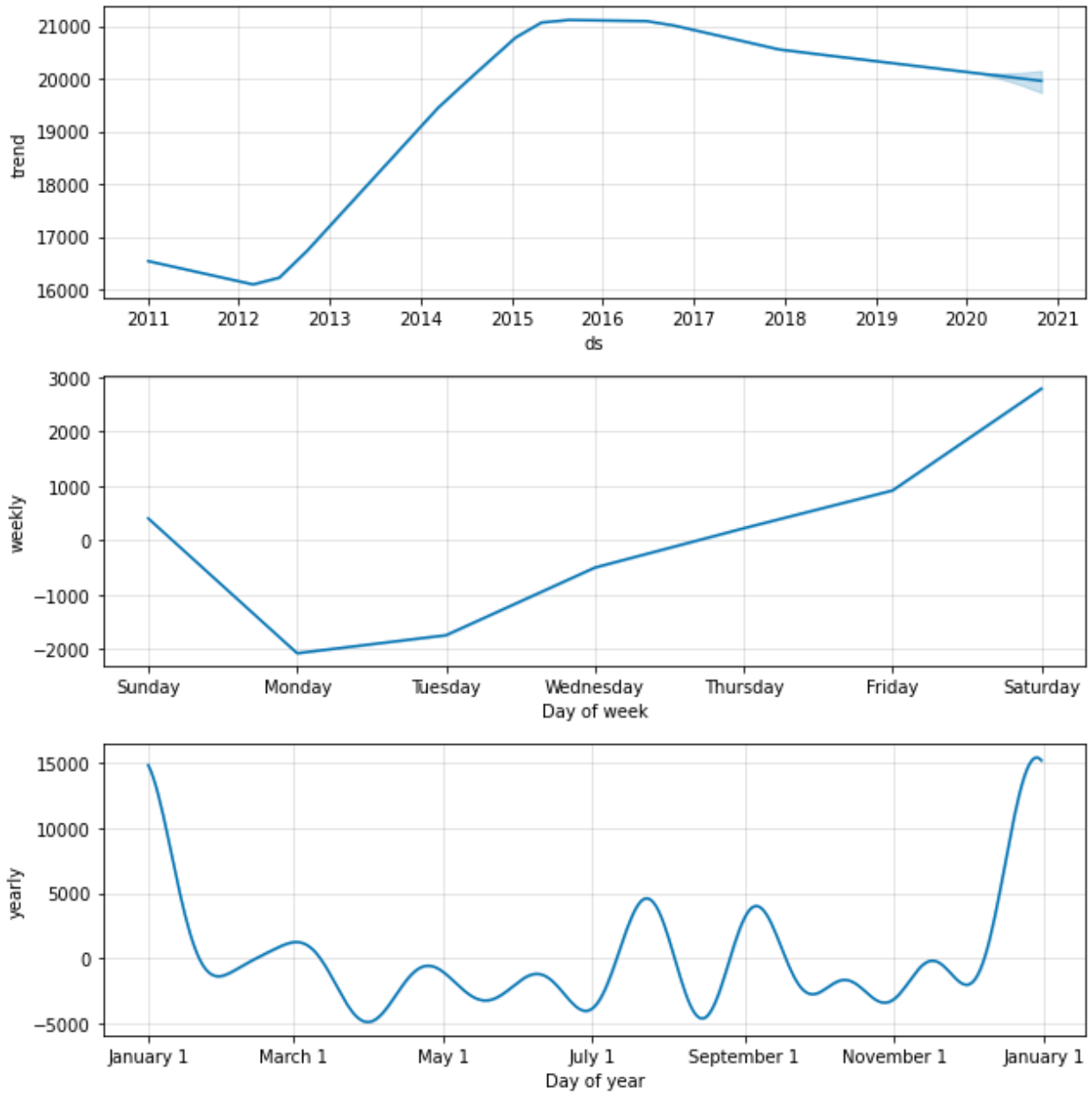


Figura 5-23 Trendet në shitje për qendrën tregtare

## 5.4 Konkluzione

Puna eksperimentale u nda në 3 seksione kryesore. Në seksionin e parë u implementuan modelet më të rëndesishme të klasterzimit për të kuptuar cili performon më mirë. Rezultoi se për bazën tonë të të dhënave, modeli K-Medoids ishte më i përshtatshëm për shkak të pranisë së të dhënave periferike (outliers) me dizavantazhin e kompleksitetit në kohë. Në pjesën e dytë të studimit u analizuan modelet e klasifikimit dhe u propozuan tre modele të ndryshme të bazuar në KNN (K-Nearest Neighbours), SVM (Support Vector Machines) dhe rrjetat Neurale NN. Vume re se rezultatet më të mira i morrëm në rastin e modelit me SVM, 80.79% saktësi në parashikim por duhet të theksojmë se rezultatet janë të ngjashme. Në pjesën e tretë të studimit u analizuan dhe propozuan modele që lidhen me seritë kohore dhe me fokus kryesor në algoritmin Prophet. U propozuan disa ndryshime nga modeli bazë të cilat na dhanë një përmirësim me 1.4 % të gabimit në parashikim dhe një MAPE < 25% për parashikime 3-mujore të shitjeve të qendrës tregtare.

## KAPITULLI 6

### 6. REZULTATET E PUNIMIT DHE PUNA NË TË ARDHMEN

#### 6.1 Rezultatet e punimit

Studimi ynë bazohet në analizën dhe aplikimin e teknikave Data Mining dhe Machine Learning në bazën e të dhënave të shitjeve në një qendër tregtare. Qëllimi ynë është të krijojmë modele klasterizimi dhe parashikuese për qendrën tregtare. Modelet e propozuara nga ne mund të përdoren në të gjitha bizneset e retail.

Pjesa teorike e studimit u nda në 4 kapituj.

Në kapitullin e parë u studiuan sistemet e inteligjencës në biznes, historiku i tyre dhe karakteristikat kryesore. Arkitekturat e ndryshme të përdorura në sistemet BI u pasqyruan dhe shpjeguan në detaje. Gjithashtu u përshkruan faktorët kryesore që duhen marrë parasysh në implementimet e sistemeve BI.

Në kapitullin e dytë, u analizuan magazinat e të dhënave DW dhe arkitekturat kryesore të tyre. U listuan avantazhet dhe disavantazhet e secilës arkitekturë. Më pas u shpjegua koncepti i kubave të të dhënave dhe arsyet e përdorimit të tyre në sistemet BI. Pamë dhe 3 teknologjitë kryesore OLAP (ROLAP, MOLAP, HOLAP), dhe karakteristikat e tyre. Kryem një studim teorik të performances së secilës teknikë dhe gjithashtu dhamë rekomandime në implementimin e sistemeve OLAP.

Në kapitullin e tretë u krye një studim i teknikave të Data Mining (DM). Fillimisht u krye një analizë e modeleve kryesore të DM, përkatesisht: modelit përshkruar dhe modelit parashikues. Algoritmat kryesore të secilit model u listuan dhe analizuan. Gjithashtu u shpjegua sesi kryhet integrimi i DM me sistemet e inteligjencës në biznes dhe statusi aktual i këtyre integritimeve. Së fundmi, u shpjeguan dhe analizuan hapat kryesore që duhet të ndërmerren gjatë implementimit të DM me fokus në paraprocesimin e të dhënave.

Në kapitullin e katërt u analizuan teknikat dhe algoritmat që u implementuan dhe modifikuan gjatë pjesës eksperimentale. Fillimisht u studiuan algoritmat e klasterizimit dhe veçanërisht algoritmat K-Means, K-Medoids dhe algoritmat hierarkike. Aparati matematikor i secilit algoritëm u shpjegua dhe ilustrua me shëmbuj dhe skema. Më pas u studiuan teknikat e klasifikimit. Në këtë pjesë u studiuan nga ana teorike algoritmat KNN (Klasifikuesi i K-fqinjëve më të afërt), algoritmi SVM dhe rrjetat neurale. Aparati matematikor i secilit algoritëm u shpjegua dhe analizua. Vlen për t'u theksuar se ju dha një fokus më i madh rrjetave neurale (NN) duke qënë se besojmë se janë teknikat më premtuese të klasifikimit dhe regresit. Arkitekturat kryesore të rrjetave NN u shpjeguan dhe fazat e përparme dhe të pas-shpërndarjes u shtjelluan. Gjithashtu u analizuan funksionet e ndryshme të aktivizimit dhe të gabimit që u përdorën. Të gjitha këto funksione u implementuan në modelin e krijuar në pjesën eksperimentale.

Së fundmi, u analizuan seritë kohore me fokus të algoritmi Prophet. U përshkrua aparati matematikor i algoritmit dhe u analizuan 3 komponentet kryesore të tij si modeli i trendit, modeli i sezonalitetit dhe ditët festive. Gjithashtu u analizuan dhe metrikat që u përdorën për të vlerësuar performancën e modeleve.

Puna eksperimentale u nda në 3 seksione kryesore. Në seksionin e parë u implementuan modelet më të rëndësishme të klasterizimit për të kuptuar cili performon më mirë. Rezultoi se për bazën tonë të të dhënave, modeli K-Medoids ishte më i përshtatshëm për shkak të pranisë së të dhënave periferike (outliers). Duke marrë në konsideratë se ky është rasti për shumicën e kompanive që operojnë në retail, ky model mund të përdoret për të pas rezultatet janë të kënaqshme. Një dizavantazh i teknikës është kompleksiteti në kohë të modelit për shkak të

përlllogaritjes së distancës për çdo çift të dhënash. Për këtë arsye, rekomandojmë përdorimin e K-Medoids vetëm në raste kur baza e të dhënave nuk ka më shumë se disa mijëra rreshta. Në baza të mëdha të dhënash, rekomandojmë përdorimin e K-Means për shkak të thjeshtësisë, efikasësisë në kohë dhe rezultateve të mira. Në rastin e përdorimit të K-Means të dhënat duhet të pastrohen nga të dhënat periferike (outliers).

Gjithashtu nuk rekomandojmë përdorimin e algoritmeve që përdorin densitetin për grupimin e klientëve, në rast se densiteti i të dhënave është pothuajse uniform.

Një tjetër rezultat interesant i seksionit të parë të studimit ishte korrelimi i zbuluar midistributeve, që i atribuohet vendodhjes fizike të stokut. Një tjetër rezultat interesant vjen nga grupimi i klientëve pas aplikimit të teknikave klasterizuese që tregojnë se një pjesë e vogël e klientëve (2%) kryejnë 1/3 e blerjeve dhe 65% e tyre kryejnë vetëm 12% të blerjeve nga klientët.

Në pjesën e dytë të studimit u analizuan modelet e klasifikimit dhe u propozuan tre modele të ndryshme të bazuar në KNN (K-Nearest Neighbours), SVM (Support Vector Machines) dhe rrjetat Neurale NN. Vëmë re se rezultatet më të mira i morrëm në rastin e modelit me SVM, 80.79% saktësi në parashikim por duhet të theksojmë se rezultatet janë të ngjashme, përkatësisht 79.864% për rrjetat neurale dhe 78.56% për KNN. Theksojmë se modeli me rrjetat neurale që propozuam është shumë premtues dhe duhet hulumtuar dhe eksperimentuar më shumë në të ardhmen me arkitektura të ndryshme të modelit. Saktësia maksimale u arrit për arkitekturën me një shtresë të ndërmjetme me 20 neurone dhe me funksion aktivizimi Sigmoid. Modeli konvergoi pas 50000 iterimesh, pra nuk kishim nevojë ta testonim për një numër epochs më të madh dhe vlera optimale e learning rate (lr) rezultoi 0.01. Ky përfundim u arrit pas shumë eksperimenteve me funksione të ndryshme aktivizimi dhe duke alternuar numrin e neuroneve në shtresën e ndërmjetme. Saktësia e arritur në parashikim mund të konsiderohet shumë e mirë nëse i referohemi numrit të të dhënave të përdorura për studimin tonë.

Në të gjitha rastet problemin me të madh në parashikim e kemi në rastin e klasës së ndërmjetme 2 (500-1000 USD). Kjo i atribuohet edhe numrit më të vogël të klientëve në këtë klasë por dhe mosbalancës në të dhëna. Të gjitha këto problematika do të adresohen në punën në vazhdim me fokus në aplikimin e teknikave SMOTE ose up sampling për reduktimin e disbalancës në të dhëna.

Në pjesën e tretë të studimit u analizuan dhe propozuan modele që lidhen me seritë kohore dhe me fokus kryesor në algoritmin Prophet. U propozuan disa ndryshime nga modeli bazë të cilat na dhanë një përmirësim me 1.4% të gabimit në parashikim (MAE). Vëmë re se ndryshimi i rendit të serive Fourier në sezonalitetet e shitjeve na dha impaktin më të madh në parashikim. Modeli i propozuar dha rezultate shumë të mira për parashikimet deri në 90 ditë (MAPE < 25%). Në diapazonin nga 90-140 ditë kishim një rritje të gabimit i cili stabilizohet përsëri pas këtij diapazoni. Ky problem, mendohet të vijë nga ndryshimi i sezoneve promovionale në vitet 2018/2019 por do të analizohet më në detaje dhe do të adresohet në studimet pasardhëse.

Si përfundim modelet e propozuara dhanë rezultate shumë të mira dhe për shkak të gjeneralitetit mund të përdoren në të gjitha bizneset retail për të rritur efikasësinë e biznesit, përmirësuar teknikat e marketing dhe shërbimin ndaj klientit.

## 6.2 Puna në të ardhmen

Sic u theksua dhe me siper fokusi i punes ne te ardhmen ,do te jete si ne pikat ne vazhdim:

- Do të aplikohen teknikat e mbikampionimit për balancimin e të dhënave dhe do të analizohet përsëri performanca e modeleve.
- Do te pasurohet dataseti me të dhëna nga vendndodhja e klienteve ,që do të merren nga sistemet e detektimit te prezencës.
- Analiza do të aplikohet në disa dimensione të të dhënave.
- Do të zgjerohet baza e të dhënave .
- Do të vazhdojë hulumtimi me rrjetat neurale : CNN (Convolutional neural network) apo RNN(Recurrent Neural Network) dhe me arkitektura të ndryshme te tyre me qëllim rritjen e saktësisë në parashikim.
- Një tjetër aspekt për të cilin do te punohet në vazhdim do të jetë krijimi i aplikacioneve për biznesin në mënyrë që të shmanget kompleksiteti i kodimit dhe këto teknika të përdoren lehtësisht pa pasur nevojën për kodim.

## REFERENCAT

- [1] Business Performance Management Software. (2009). What is Business Intelligence? <[http://www.successsoftwaresolution.com/Business\\_Performance\\_Management\\_Software.html](http://www.successsoftwaresolution.com/Business_Performance_Management_Software.html)>.
- [2] Business Objects. (2007). About Business Intelligence. as viewed by William Yeoh on 12th Nov, 2007 <[http://www.businessobjects.com/businessintelligence/default.asp?intcmp=ip\\_company2](http://www.businessobjects.com/businessintelligence/default.asp?intcmp=ip_company2)>.
- [3] David Loshin. (2008). The Incremental Business Intelligence Infrastructure: A Strategy for Evolving Analysis Services. Chicago, IL: Knowledge Integrity, Inc.
- [4] Dr. Saadia Asif. (2009). An Overview of Business Intelligence. as viewed on 29th October 2009 <<http://www.inforica.com/in/download/bipresentation.pdf>>: Inforica Inc.
- [5] ESCC, U. A. (2009). Overview: Business Intelligence Reference Guide. U.S. Army Enterprise Solutions Competency Center.
- [6] NAV, M. D. (2007). Business Intelligence: Driving better business performance for companies with changing needs.
- [7] Platon A/S. (2009). Introduction to Business Intelligence & Data Warehouse. Copenhagen, Denmark: Platon A/S.
- [8] Schiff, M. A. (2009). Business Intelligence: A Guide for Midsize Companies. SAP White Paper.
- [9] Vezzosi, P. (2009). Create a Universe: The SAP BusinessObjects Semantic Layer. <[http://www.sdn.sap.com/irj/boc/index.jsessionid=\(J2EE3417400\)ID1965100350DB00528079734869604848End?ri=library/uuid/50690e88-cec9-2b10-6b98-97c763f66b7b&overridelayout=true](http://www.sdn.sap.com/irj/boc/index.jsessionid=(J2EE3417400)ID1965100350DB00528079734869604848End?ri=library/uuid/50690e88-cec9-2b10-6b98-97c763f66b7b&overridelayout=true)>: SAP Business Objects
- [10] Romm M., Introduction to Data Warehousing, San Diego SQL User Group
- [11] Thomas Conolly and Caroly Begg., Database Systems, 3th Edition, Addison- Wesley, 2002
- [12] W.H.Inmon, Building the Data Warehouse, 3rd Edition, John Wiley, Chap.2, p. 36, 2002
- [13] S. Chaudhuri and U. Dayal. An Overview of Data Warehousing and OLAP Technology. SIGMOD Rec., 26(1):65–74,
- [14] SUGI 24: The Art of Designing HOLAP Databases 2011
- [15] K.Wu, E. J. Otoo, and A. Shoshani. A Performance Comparison of Bitmap Indexes. In CIKM '01: Proceedings of the tenth international conference on Information and knowledge management, pages 559–561, New York, NY, USA, 2001. ACM.
- [16] K. Wu, E. J. Otoo, and A. Shoshani. On the Performance of Bitmap Indices for High Cardinality Attributes. In Mario A. Nascimento, M. Tamer Ozsu, Donald Kossmann, Renée J. Miller, José A. Blakeley, and K. Bernhard Schiefer, editors,
- [17] MicroStrategy. The Case for Relational OLAP. 2010 . <https://www.microstrategy.com/Solutions/5Styles/olapanalysis.asp>.
- [18] Lico, L. (2017). Data Mining Techniques in Database Systems . European Journal of Sustainable Development Research , 2 (1) , 43-50 . Retrieved from <https://dergipark.org.tr/en/pub/ejsdr/issue/27742/294862>
- [19] Frans Coenen ,Data Mining: Past, Present and Future, The Knowledge Engineering Review, 2004, Cambridge University Press
- [20] Pradnya P. Sondwale, Overview of Predictive and Descriptive Data Mining Techniques, International Journal of Advanced Research in Computer Science and Software Engineering, April 2015
- [21] Mihika Shah, Sindhu Nair , A Survey of Data Mining Clustering Algorithms, International Journal of Computer Applications (0975 – 8887) Volume 128 – No.1, October 2015
- [22] Irina Tudor, Association Rule Mining as a Data Mining Technique, Petroleum-Gas University of Ploiești, Buletin Vol. LX No. 1/2008
- [23] Trupti A. Kumbhare et al An Overview of Association Rule Mining Algorithms, / (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 5 (1) , 2014, 927-930
- [24] N. Elavarasan, Dr. K.Mani, A Survey on Feature Extraction Techniques, International Journal of Innovative Research in Computer and Communication Engineering Vol. 3, Issue 1, January 2015
- [25] Fabricio Voznika Leonardo Viana "DATA MINING CLASSIFICATION" Springer, 2001
- [26] Han - Data Mining Concepts and Techniques 3rd Edition - 2012.pdf
- [27] Shanta Rangaswamy, Time Series Data Mining Tool, International Journal of Research in Computer and Communication Technology, Vol 2, Issue 10, October- 2013
- [28] Zanaj, Lico A multidimensional analyses in Business Intelligence systems IJCSIS May 2012
- [29] H. Zhu, "Online analytical mining of association rules," Master Thesis, Simon Fraser University, 1998, pp. 1-117.
- [30] S. Dzeroski, D. Hristovski and B. Peterlin, "Using data mining and OLAP to discover patterns in a database of patients with Y chromosome deletions," in Proc. AMIA Symp., 2000, pp. 215–219.
- [31] Han - Data Mining Concepts and Techniques 2nd Edition - 2006.pdf
- [32] Data Mining Concepts and Techniques Third Edition, Jiawei Han, University of Illinois at Urbana–Champaign, Micheline Kamber, Jian Pei, Simon Fraser University [2012]
- [33] Data Mining and Machine Learning Fundamental Concepts and Algorithms by Mohammed J. Zaki, Wagner Meira Jr, Cambridge University Press, 2020

- [34] Christopher M. Bishop. 2006. Pattern Recognition and Machine Learning (Information Science and Statistics). Springer-Verlag, Berlin, Heidelberg. 2006
- [35] Calster, T.V., Bossche, F.V., Baesens, B., & Lemahieu, W. (2020). Profit-oriented sales forecasting: a comparison of forecasting techniques from a business perspective. ArXiv, abs/2002.00949.
- [36] Sahay, A., Joseph, A. (2020). Integration of Prophet Model and Convolution Neural Network on Wikipedia Trend Data. Journal of Computational and Theoretical Nanoscience.
- [37] Benedetto, C., Satrio, A., Darmawan, W., Nadia, B. U., Hanafiah, N., (2021). Time series analysis and forecasting of coronavirus disease in Indonesia using ARIMA model and PROPHET. Procedia Computer Science, Volume 179.
- [38] Taylor SJ, Letham B. 2017. Forecasting at scale. PeerJ Preprints 5:e3190v2 <https://doi.org/10.7287/peerj.preprints.3190v2>
- [39] Žunić, E., Korjenić, K., Hodžić, K., Đonko, D. (2020). Application of Facebook's prophet algorithm for successful sales forecasting based on real-world data. International Journal of Computer Science & Information Technology.
- [40] Li, Y., Ma, Z., Pan, Z. et al. Prophet model and Gaussian process regression based user traffic prediction in wireless networks. Sci. China Inf. Sci. 63, 142301 (2020).
- [41] Toharudin, T., Pontoh, R. S., Caraka, R. E, Zahroh, S. (2021). Employing Long Short-Term Memory and Facebook Prophet Model in Air Temperature Forecasting. Communication in Statistics- Simulation and Computation. 10.1080/03610918.2020.1854302.
- [42] Indrit Enesi, Ledion Liço, Aleksander Biberaj and Desari Shahu, Analysing Clustering Algorithms Performance in CRM Systems, ICEIS 2021: 23rd International Conference on Enterprise Information Systems, April 2021
- [43] L. Lico, I, Enesi, Performance Analysis of KNN and Neural Networks for Predicting Customer Purchases in a Real Retail Department Store, 3rd International Congress on Human-Computer Interaction, Optimization and Robotic Applications, June 11-13, 2021 – IEEE Turkey Section.
- [44] Ledion Lico, Indrit Enesi, Betim Cico, Analyzing Performance of Clustering Algorithms on a Real Retail Dataset, 2021 IEEE International Conference on Information Technologies (InfoTech-2021 – 35th edition) 16-17 September 2021, (accepted for publication)
- [45] Ledion Lico; Indrit Enesi. (Volume. 6 Issue. 6, June - 2021) "Performance Evaluation of Svm in a Real Dataset to Predict Customer Purchases.", International Journal of Innovative Science and Research Technology (IJISRT), www.ijisrt.com. ISSN - 2456-2165 , PP :- 1071-1075.
- [46] Ledion Lico, Indrit Enesi, Predicting Customer Behaviour using Prophet Algorithm in a real time series dataset, European Scientific Journal, July 2021 <https://ejournal.org/index.php/esj>



## ANEKS 1

Kodet e përdorura:

### Agregimi i të dhënave

/\* Formatted on 2021/04/18 16:35 (Formatter Plus v4.8.8) \*/

#### ##Shitje Totale 2019##

```
SELECT    l.cliente, l.sesso, l.citta, round(((sysdate-l.datanascita)/365),0), round(SUM
(((d.prezzo) - (d.sconto) - (d.scontopers))),0)
  FROM liclicard l, pstesta p, psdett d, liarticoli a
 WHERE l.cliente = p.cliente
       AND p.scontrino = d.scontrino
       AND p.cassa = d.cassa
       AND p.giorno = d.giorno
       AND d.articolo=a.ARTICOLO
       AND d.giorno > '1-jan-2019'
       AND d.giorno <= '31-dec-2019'
GROUP BY l.cliente,l.sesso, l.citta,round(((sysdate-l.datanascita)/365),0);
```

#### ##Shitje Acc\_Parf 2019##

```
SELECT    l.cliente, l.sesso, l.citta, round(((sysdate-l.datanascita)/365),0), round(SUM
(((d.prezzo) - (d.sconto) - (d.scontopers))),0)
  FROM liclicard l, pstesta p, psdett d, liarticoli a
 WHERE l.cliente = p.cliente
       AND p.scontrino = d.scontrino
       AND p.cassa = d.cassa
       AND p.giorno = d.giorno
       AND d.articolo=a.ARTICOLO
       AND d.giorno > '1-jan-2019'
       AND d.giorno <= '31-dec-2019'
       AND a.classif7 in (4,5,950,8,980)
GROUP BY l.cliente,l.sesso, l.citta,round(((sysdate-l.datanascita)/365),0);
```

#### ##Shitje Others 2019##

```
SELECT    l.cliente, l.sesso, l.citta, round(((sysdate-l.datanascita)/365),0), round(SUM
(((d.prezzo) - (d.sconto) - (d.scontopers))),0)
  FROM liclicard l, pstesta p, psdett d, liarticoli a
 WHERE l.cliente = p.cliente
       AND p.scontrino = d.scontrino
       AND p.cassa = d.cassa
       AND p.giorno = d.giorno
       AND d.articolo=a.ARTICOLO
       AND d.giorno > '1-jan-2019'
```

```

AND d.giorno <= '31-dec-2019'
AND a.classif7 in (810,830)
GROUP BY l.cliente,l.sesso, l.citta,round(((sysdate-l.datanascita)/365),0);

```

### ##Blerjet 3 mujore për kliente/kategori 2020 ##

```

select * from (SELECT l.cliente, l.sesso, l.citta, round(((sysdate-l.datanascita)/365),0),SUM
(d.prezzo) as cmim1,a.classif7
FROM liclicard l, pstesta p, psdett d, liarticoli a
WHERE l.cliente = p.cliente
AND p.scontrino = d.scontrino
AND p.cassa = d.cassa
AND p.giorno = d.giorno
AND d.articolo=a.ARTICOLO
AND d.giorno > '1-jan-2020'
AND d.giorno <= '31-mar-2020'
GROUP BY l.cliente,l.sesso,a.classif7, l.citta,round(((sysdate-l.datanascita)/365),0)) kile
,(select distinct(cliente),max(cmim2) as b from (SELECT l.cliente, l.sesso, l.citta,
round(((sysdate-l.datanascita)/365),0),SUM (d.prezzo) as cmim2,a.classif7
FROM liclicard l, pstesta p, psdett d, liarticoli a
WHERE l.cliente = p.cliente
AND p.scontrino = d.scontrino
AND p.cassa = d.cassa
AND p.giorno = d.giorno
AND d.articolo=a.ARTICOLO
AND d.giorno > '1-jan-2020'
AND d.giorno <= '31-mar-2020'
GROUP BY l.cliente,l.sesso, l.citta,round(((sysdate-l.datanascita)/365),0),a.classif7) group by
cliente) maks where kile.cliente=maksi.cliente and kile.cmim1=maksi.b;

```

### ##Blerjet 3 mujore për kliente/moshe,gjini,qytet,sasi 2020 ##

```

SELECT l.cliente, l.sesso, l.citta, round(((sysdate-l.datanascita)/365),0),sum(d.quantita)
FROM liclicard l, pstesta p, psdett d
WHERE l.cliente = p.cliente
AND p.scontrino = d.scontrino
AND p.cassa = d.cassa
AND p.giorno = d.giorno
AND d.giorno > '1-jan-2020'
AND d.giorno <= '31-mar-2020'
GROUP BY l.cliente,l.sesso, l.citta,round(((sysdate-l.datanascita)/365),0);

```

**##Shitjet ditore 2011-2019 ##**

```

SELECT  round(SUM (((d.prezzo) - (d.sconto) - (d.scontopers))/120),0),p.giorno

FROM  pstesta p, psdett d, liarticoli a
WHERE  p.scontrino = d.scontrino
AND  p.cassa = d.cassa
AND  p.giorno = d.giorno
AND  d.articolo=a.ARTICOLO
AND  d.giorno > '1-jan-2010'
AND  d.giorno <= '31-dec-2019'
AND                                     a.classif7                               in
(1,2,6,10,20,21,22,31,32,33,38,39,40,47,49,51,57,56,60,62,68,820,910,920,930,940,960,990)
GROUP BY p.giorno order by p.giorno asc;

```

```

SELECT  round(SUM (((d.prezzo) - (d.sconto) - (d.scontopers))/120),0),extract(month from
p.giorno) as months,extract(year from p.giorno) as years

```

```

FROM  pstesta p, psdett d, liarticoli a
WHERE  p.scontrino = d.scontrino
AND  p.cassa = d.cassa
AND  p.giorno = d.giorno
AND  d.articolo=a.ARTICOLO
AND  d.giorno > '1-jan-2010'
AND  d.giorno <= '31-dec-2019'
AND                                     a.classif7                               in
(1,2,6,10,20,21,22,31,32,33,38,39,40,47,49,51,57,56,60,62,68,820,910,920,930,940,960,990)
GROUP BY extract(month from p.giorno),extract(year from p.giorno) order by years
asc,months asc;

```

```

SELECT  to_char(p.giorno,'MM-DD-YYYY'),round(sum(p.totalescontrino)/120,0)

```

```

FROM  pstesta p
WHERE  p.giorno > '1-jan-2011'
AND  p.giorno <= '31-dec-2019'
GROUP BY p.giorno order by p.giorno asc;

```

**Klasterizimi**

```
## Importi dhe para-proçesimi i të dhënave ##
```

```
import pandas as pd
from sklearn.preprocessing import StandardScaler
import matplotlib.pyplot as plt
from sklearn.cluster import KMeans
import plotly.express as px
import numpy as np

dstore_clients = pd.read_csv('clusters.csv', engine='python')
dstore_clients.head()
dstore_clients.info()
sex='F'
dstore_clients['SESSO']=dstore_clients['SESSO'].fillna(sex)

median= dstore_clients['AGE'].median()
dstore_clients['AGE']=dstore_clients['AGE'].fillna(median)

dstore_clients['Clothes']=dstore_clients['Clothes'].fillna(0)
dstore_clients['Acc']=dstore_clients['Acc'].fillna(0)
dstore_clients['Others']=dstore_clients['Others'].fillna(0)
```

```
##Instalimi i matësit të kohës ##
```

```
!pip install ipython-autotime
%load_ext autotime
```

```
##K-Means##
```

```
def kmeans(data, n_clusters):
    s = StandardScaler()
    data = s.fit_transform(data)
    k_m = KMeans(n_clusters=n_clusters,random_state=0)
    k_m.fit(data)
    y_pred = k_m.predict(data)
    fig = px.scatter(data, x=data[:,0], y=data[:,1],
                    color=y_pred,
                    labels={'x': 'Veshje', 'y': 'Aksesore'})
    fig.show()
    print(k_m.inertia_)
```

```
##Metoda Elbow##
```

```

inertia=[]
for i in range(2,12):
    s = StandardScaler()
    data = s.fit_transform(a_clients)

    k_m = KMeans(n_clusters=i,random_state=0)
    k_m.fit(data)
    inertia.append(k_m.inertia_)

```

```

plt.plot(range(2,12), inertia)
plt.title('Metoda Elbow')
plt.xlabel('Numri i Klasterave')
plt.ylabel('Inertia')
plt.show()

```

### ##Korrelimi##

```

import numpy as np
import matplotlib.pyplot as plt
from sklearn.metrics.pairwise import euclidean_distances
import math
from numpy import linalg as LA
from scipy import linalg
import sklearn
from sklearn import decomposition
from sklearn import datasets
from sklearn.preprocessing import StandardScaler
from sklearn.decomposition import PCA
import pandas as pd
import seaborn as sns
klient=dstore_clients.iloc[:,4:]
correlation = klient.corr()
plt.figure(figsize=(10,10))
sns.heatmap(correlation, vmax=1, square=True,annot=True,cmap='cubehelix')
plt.title('Korrelimi midis Atributeve') # we plot the correlation between features.

```

### ##Dendogram##

```

import scipy.cluster.hierarchy as sch
dendrogram = sch.dendrogram(sch.linkage(a_clients, method = 'ward'))
plt.title('Dendrogam', fontsize = 20)
plt.xlabel('Klientet')
plt.ylabel('Distanca Euklidiane')
plt.show()

```

### ##Klastering Hierarkik##

```

from sklearn.cluster import AgglomerativeClustering
def hclustering(data, n_clusters):

```

```

s = StandardScaler()
data = s.fit_transform(data)
hclustering = AgglomerativeClustering(n_clusters=n_clusters)
y_pred = hclustering.fit_predict(data)
#fig = px.scatter_3d(data, x=data[:,0], y=data[:,1], z=data[:,2], labels={'x': 'Clothes', 'y':
'Accessories_Cosmetics', 'z': 'Others'},
    #color=y_pred)
fig = px.scatter(data, x=data[:,0], y=data[:,1],
    color=y_pred,
    labels={'x': 'Clothes', 'y': 'Accessories_Cosmetics'})
fig.show()

```

### ##DBSCAN##

```

from sklearn.cluster import DBSCAN
def dbscan(data, eps, min_samples):
    s = StandardScaler()
    data = s.fit_transform(data)
    db = DBSCAN(eps=eps, min_samples=min_samples)
    db.fit(data)
    y_pred = db.fit_predict(data)
    #fig = px.scatter_3d(data, x=data[:,0], y=data[:,1], z=data[:,2], labels={'x': 'Clothes', 'y':
'Accessories_Cosmetics', 'z': 'Others'},
    #color=y_pred)
    #fig.show()
    fig = px.scatter(data, x=data[:,0], y=data[:,1], color=y_pred,
    labels={'x': 'Veshje', 'y': 'Aksesore'})
    fig.show()
    print(len(np.unique(db.labels_)))
    for i in range(-1, len(np.unique(db.labels_))):
        print(np.where(db.labels_==i))

```

```

s = StandardScaler()
data = s.fit_transform(a_clients)
for i in [0.1, 0.2]:
    for k in [2, 3, 4, 5, 6]:
        db = DBSCAN(eps=i, min_samples=k)
        db.fit(data)
        print(len(np.unique(db.labels_)))

```

### ##K-Medeoids##

```

from sklearn_extra.cluster import KMedoids

```

```

s = StandardScaler()
data = s.fit_transform(a_clients)

```

```

kmedoids = KMedoids(n_clusters=4, random_state=0).fit(data)
y_pred = kmedoids.predict(data)
plt.scatter(x=data[:,0], y=data[:,1],
            c=y_pred,cmap='Paired')
plt.title("K-Medeoids")
plt.xlabel('Veshje')
plt.ylabel('Acc')

fig = px.scatter(data, x=data[:,0], y=data[:,1],
                color=y_pred,
                labels={'x': 'Veshje', 'y': 'Aksesore'})
#fig = px.scatter_3d(data, x=data[:,0], y=data[:,1], z=data[:,2],
#                   color=y_pred,
#                   #labels={'x': 'Clothes', 'y': 'Accessories_Cosmetics', 'z': 'Others'})

fig.show()

```

## **Klasifikimi**

**##Importi dhe para-proçesimi i të dhënave##**

```

store_clients = pd.read_csv('clientp7.csv', engine='python')
store_clients['SESSO'].unique()
store_clients.info()
store_clients.head()

```

**##Histogram##**

```

import seaborn as sns
tips = sns.load_dataset("tips")
ax=sns.histplot(store_clients, x="Label", stat="count", discrete=True,shrink=0.2)
ax.set(xlabel='Customer Class', ylabel='Number of Clients')

```

```

median= store_clients['AGE'].median()
store_clients['AGE']=store_clients['AGE'].fillna(median)
sasi=0
store_clients['Sasi']=store_clients['Sasi'].fillna(sasi)

```

```

sex='F'
store_clients['SESSO']=store_clients['SESSO'].fillna(sex)

```

```

predictors=['AGE', 'SEX', 'Val', 'Sasi', 'CAT', 'Prev']
from sklearn import preprocessing
le = preprocessing.LabelEncoder()
store_clients['SEX']=le.fit_transform(store_clients['SESSO'])

```

```
##KNN##
```

```
X=store_clients[predictors].to_numpy()
y=store_clients['Label'].to_numpy()
from sklearn.model_selection import train_test_split
```

```
##Split Dataset##
```

```
X_train,X_test,y_train,y_test=train_test_split(X,y,test_size=0.2)
```

```
##Apliko Modelin##
```

```
from sklearn.neighbors import KNeighborsClassifier
knn=KNeighborsClassifier(n_neighbors=15)
```

```
knn.fit(X_train,y_train)
```

```
y_pred=knn.predict(X_test)
```

```
##Mat Saktisine##
```

```
from sklearn.metrics import accuracy_score
```

```
accuracy_score(y_test,y_pred)
```

```
##Cross-Validation##
```

```
from sklearn.model_selection import KFold,cross_val_score
kfold=KFold(n_splits=5,shuffle=True)
scores=cross_val_score(knn,X_train,y_train,cv=kfold)
scores
```

```
##Matrica e Saktise##
```

```
from sklearn.metrics import plot_confusion_matrix
plot_confusion_matrix(knn, X_test, y_test)
```

```
##SVM##
```

```
import pandas as pd
from sklearn.preprocessing import StandardScaler
import matplotlib.pyplot as plt
import plotly.express as px
store_clients = pd.read_csv('clientp7.csv', engine='python')
```



```

from sklearn.model_selection import train_test_split
#Përdorim random.seed(20) për te ruajtur
#gjenerimin random gjate eksperimenteve
import numpy as np
np.random.seed(100)
predictors=['AGE','SEX','Val','Sasi','CAT','Prev']
X=store_clients[predictors].to_numpy()
y=store_clients['Label'].to_numpy()

X_train,X_test,y_train,y_test=train_test_split(X,y,test_size=0.2)

from sklearn.preprocessing import StandardScaler
s = StandardScaler()
SX_train = s.fit_transform(X_train)
SX_test= s.transform(X_test)

##Importojme dhe aplikojme modelin me berthama te ndryshme##

from sklearn.svm import SVC

svc1 = SVC(kernel='linear')
svc1.fit(SX_train,y_train)
y_pred1=svc1.predict(SX_test)
y_pred1

svc2 = SVC(kernel='sigmoid')
svc2.fit(SX_train,y_train)
y_pred2=svc2.predict(SX_test)
y_pred2

svc3 = SVC(kernel='poly')
svc3.fit(SX_train,y_train)
y_pred3=svc3.predict(SX_test)
y_pred3

svc4 = SVC(kernel='rbf')
svc4.fit(SX_train,y_train)
y_pred4=svc4.predict(SX_test)
y_pred4

##Matim Saktesine##

from sklearn.metrics import accuracy_score
from sklearn.model_selection import KFold,cross_val_score
accuracy_score(y_test,y_pred3)

```

**##Cross Validation##**

```
kfold=KFold(n_splits=10,shuffle=True)
scores=cross_val_score(svc3,SX_train,y_train,cv=kfold)
print(scores)
np.mean(scores)
```

**##Matrica e Saktetise##**

```
from sklearn.metrics import plot_confusion_matrix
plot_confusion_matrix(svc1, SX_test, y_test)
```

**##Precisioni##**

```
from sklearn.metrics import precision_score
precision_score(y_test, y_pred3, average='micro')
```

**##Rrjetat Neurale##****##Import dhe para-proçeso te dhenat##**

```
import pandas as pd
from sklearn import preprocessing
import torch
import torch.nn as nn
import numpy as np
np.random.seed(100)

store_clients = pd.read_csv('clientp10.csv', engine='python')
median= store_clients['AGE'].median()
store_clients['AGE']=store_clients['AGE'].fillna(median)
sex='F'
```

```
store_clients['SESSO']=store_clients['SESSO'].fillna(sex)
from sklearn import preprocessing
le = preprocessing.LabelEncoder()
store_clients['SEX']=le.fit_transform(store_clients['SESSO'])
predictors=['AGE','SEX','CAT','Sasi','Val','Prev']
X=store_clients[predictors].to_numpy()
```

```
y=store_clients['Label'].to_numpy()
```

**##Ndajme dataset ne 80 % trajnim-20% testim##**

```
from sklearn.model_selection import train_test_split
X_train,X_test,y_train,y_test=train_test_split(X,y,test_size=0.2)
```

```

##Ndertojme rrjetin neural##

# input layer nodes = 4 = number of features
# hidden layer nodes = 3
# outplut layer nodes = 3 = number of categories
iln = 6
hln = 20
oln = 3
eta = 0.01
num_epoch = 60000

# fc = fully connected
class Net(nn.Module):
    def __init__(self):
        super(Net, self).__init__()
        self.fc1 = nn.Linear(iln, hln)
        self.fc2 = nn.Linear(hln, oln)
    def forward(self, x):
        #x = torch.nn.functional.relu(self.fc1(x))
        x = torch.sigmoid(self.fc1(x))
        x = self.fc2(x)
        return x

model = Net()

criterion = nn.CrossEntropyLoss()
optimizer = torch.optim.SGD(model.parameters(), lr = eta)

X = torch.Tensor(X_train).float()
y = torch.Tensor(y_train).long()

##Trajnome modelin##

for epoch in range(num_epoch):
    optimizer.zero_grad()
    out = model(X)
    loss = criterion(out, y)
    loss.backward()
    optimizer.step()
    if epoch % 10000 == 0:
        print('epoch:', epoch, 'loss:', loss.item())

##Testojme modelin##

out = model(X)
( _, predicted) = torch.max(out.data, 1)

```

```

##Percakto sakesine##

print('Accuracy is:', (100 * torch.sum(y == predicted).double() / len(y)))

##Cross-Validation##

from sklearn.model_selection import KFold,cross_val_score

kfold=KFold(n_splits=5,shuffle=True)

##Matrica e sakesise##

from sklearn.metrics import confusion_matrix
confusion_matrix(y_test,predicted )

```

## **Serite Kohore**

```

##Prophet##

##Import Algoritmin##

from fbprophet import Prophet
from fbprophet.plot import plot_plotly
import numpy as np
import pandas as pd

##Lexo te dhenat##
total=pd.read_csv('s2019.csv',header=None)
##Lexo festat##
holidays=pd.read_csv('holidays.csv',header=None)

##Dataset##

train_tot= pd.DataFrame()
train_tot['ds'] = pd.to_datetime(total[0])
train_tot['y']=total[1]
train_tot[0:5]

##Apliko dhe trajno modelin##

prophet_model = Prophet()
prophet_model.fit(train_tot)

##Grafiku i te dhenave##

```

```

fig = plt.figure(facecolor='w', figsize=(10, 6))
plt.plot(train_tot.ds, train_tot.y)

##Apliko modelin dhe shfaq te dhenat##

forecast=prophet_model.predict(future)
fig1 =prophet_model.plot(forecast)
from fbprophet.plot import add_changepoints_to_plot
fig= prophet_model.plot(forecast)
a = add_changepoints_to_plot(fig.gca(), prophet_model, forecast)

##Shfaq trendin##

fig1 = prophet_model.plot_components(forecast)

##Vendos dhe parametrizo te dhenat e festive##

holiday_season = pd.DataFrame({
    'holiday': holidays[0],
    'ds': pd.to_datetime(holidays[1]),
    'lower_window': -1,
    'upper_window': 0,
})

##Shto sezonalitetin dhe modifiko rendin Furie dhe shfaq modelin e modifikuar##

pro_holiday= Prophet(holidays=holiday_season,holidays_prior_scale =
25).add_seasonality(name='bimonthly',period=365.25/6,fourier_order=5).add_seasonality(n
ame='weekly',period=7,fourier_order=20)
pro_holiday.fit(train_tot)
future_data = pro_holiday.make_future_dataframe(periods=12, freq = 'm')

#forecast the data for future data
forecast_data = pro_holiday.predict(future_data)
pro_holiday.plot(forecast_data);

##Cross Validation##

from fbprophet.diagnostics import cross_validation
df_cv2 = cross_validation(pro_holiday, initial='2000 days', period='180 days', horizon = '180
days')
# Python
from fbprophet.diagnostics import performance_metrics
df_ph = performance_metrics(df_cv2)
df_ph[50:100]

```

```
##Mat MAPE për modelin e modifikuar##
```

```
plt.plot( df_ph['mape'][0:150], 'b') # plotting t, b separately
```

```
plt.title('MAPE e Modelit te modifikuar')
```

```
plt.xlabel('Diapazoni(Dite)')
```

```
plt.ylabel('MAPE')
```

```
##Krahaso Modelet bazuar ne MAE##
```

```
y_true = train_tot[(train_tot['ds'] >= '2019-01-01') & (train_tot['ds'] <= '2019-12-30')
][['y'].values
```

```
y_pred = forecast_data[(forecast_data['ds'] >= '2019-01-01') & (forecast_data['ds'] <=
'2019-12-30')][['yhat'].values
```

```
mae = mean_absolute_error(y_true, y_pred)
```

```
print('MAE: %.3f' % mae)
```

```
# plot expected vs actual
```

```
plt.plot(y_true, label='Actual')
```

```
plt.plot(y_pred, label='Predicted')
```

```
plt.legend()
```

```
plt.show()
```

```
y_true = train_tot[(train_tot['ds'] >= '2019-01-01') & (train_tot['ds'] <= '2019-12-30')
][['y'].values
```

```
y_pred = forecast[(forecast['ds'] >= '2019-01-01') & (forecast['ds'] <= '2019-12-
30')][['yhat'].values
```

```
mae = mean_absolute_error(y_true, y_pred)
```

```
print('MAE: %.3f' % mae)
```

```
# plot expected vs actual
```

```
plt.plot(y_true, label='Actual')
```

```
plt.plot(y_pred, label='Predicted')
```

```
plt.legend()
```

```
plt.show()
```