UNIVERSITETI POLITEKNIK – TIRANË
FAKULTETI I TEKNOLOGJISË SË INFORMACIONIT
DEPARTAMENTI I INXHINIERISË INFORMATIKE
SHESHI NËNË TEREZA, 1 – TIRANË
TEL/FAX : +355 4 2278 159

KLASIFIKIMI I EMAIL-EVE SPAM DUKE PËRDORUR MACHINE LEARNING

TEZË PËR FITIMIN E DIPLOMËS BACHELOR
NË
INXHINIERI INFORMATIKE

NGA

DORIS KOLE

UDHËHEQËS SHKENCOR

PROF. DR. ELINDA MEÇE

TIRANË, KORRIK 2024

UNIVERSITETI POLITEKNIK I TIRANËS
Fakulteti i Teknologjisë së Informacionit
Sheshi Nënë Tereza, 1 – Tiranë
Tel/Fax : +355 4 2278 159

PROJEKT – DIPLOMË
Cikli i Parë i Studimeve
Bachelor
në
Inxhinieri Informatike

TEMA: **Klasifikimi i email-eve spam duke përdorur Machine Learning**

DEKANI	***Prof.Dr. Elinda Meçe***
PËRGJEGJËSI I DEPARTAMENTIT	***Prof. Asoc. Enida Sheme***
UDHËHEQËSI	***Prof.Dr. Elinda Meçe***
DIPLOMANTI	***Doris Kole***

Tiranë 2023-2024

REPUBLIKA E SHQIPËRISË
UNIVERSITETI POLITEKNIK I TIRANËS
FAKULTETI I TEKNOLOGJISË SË INFORMACIONIT
DEPARTAMENTI I INXHINIERISË INFORMATIKE
Sheshi "Nënë Tereza", Nr. 1, Tiranë
Tel. dhe Fax : (+355) 4 2278 159

DEKANI

Prof.Dr. Elinda Meçe

FLETË – DETYRË

mbi PROJEKT – DIPLOMËN
Cikli i Parë i Studimeve
Bachelor
në
Inxhinieri Informatike

Studenti : Doris Arben Kole Nr. i Regj. PO523A100327
(emri, atësia, mbiemri)

Departamenti i Inxhinierisë Informatike

I. Tema e Projekt – Diplomës

Klasifikimi i email-eve spam duke përdorur Machine Learning

II. Afati i dorëzimit të Projekt- Diplomës: 25.06.2024

III. Të dhëna mbi Projekt- Diplomën

1. Për realizimin e kësaj projekt-diplome është përdorur programi i shkruar në Java, softueri i njohur WEKA.
2. Konceptet kryesore që bënë të mundur realizimin e kësaj projekt-diplome janë Supervised Machine Learning dhe algoritmet e lidhura me të.
3. Aplikacionet që kanë ndihmuar në zhvillimin e kësaj projekt-diplome janë Google, WEKA, Java

IV. Përmbajtja e Projekt- Diplomës

A. Relacioni

1. Hyrja dhe prezantimi me problemin
2. Rëndësia e zgjidhjes së problemit
3. Analizimi teorik mbi të cilin bazohet problemi i shqyrtuar në projekt-diplomë
4. Analizimi i para-përpunimit të të dhënave dhe trajnimi i dataset-it
5. Ekzekutimi i algoritmave në platformën WEKA
6. Përfundimet e arritura nga punimi dhe rekomandime për studime të mëtejshme

V. Kontrolloi në Departament (studenti është i detyruar që me materialet e përgatitura në atë kohë të paraqitet në Departament)

1. 18.03.2024 Kontrolloi Prof. Dr. Elinda Meçe

2. 27.05.2024 Kontrolloi Prof. Dr. Elinda Meçe

3. 25.06.2024 Kontrolloi Prof.Asoc.Enida Sheme

Udhëheqësi: Prof. Dr. Elinda Meçe
(titulli, emri, mbiemri)

Data e miratimit të temës 18.03.2024

Përgjegjësi i Departamentit

Dr. Enida Sheme

Kjo fletë-detyrë plotësohet në dy kopje, një i bashkëngjitet Projekt-Diplomës së kryer, që bashkë me të paraqitet në Komisionin e Mbrojtjes së Projekt-Diplomave dhe tjetra i jepet studentit.

Deklaratë mbi plagjaturën

Unë konfirmoj se ky material është puna ime individuale, e pa kopjuar nga puna e askujt tjetër (e publikuar ose jo), dhe nuk është dorëzuar për vlerësim më parë në Universitetin Politeknik të Tiranës apo diku tjetër. Unë konfirmoj se njoh Rregulloren e UPT dhe Kodin e Etikës si dhe masat e parashikuara në to për plagjaturën.

Universiteti Politeknik i Tiranës dhe Udhëheqësi i Diplomës kanë të drejtën e publikimit dhe përdorimit të këtij materiali.

Dt. 25.06.2024

Doris Kole

Firma

Abstrakt (Shqip)

Për shumë kohë, email-et spam kanë krijuar një sërë problemesh të ndryshme për marrësit e tyre dhe organizatat e lidhura me ta. Këto emaile të padëshiruara mund të bëhën shkak i rënies pre të mashtrimeve deri te probleme në menaxhimin e kapacitetit te memories dhe parametrave të tjerë të kompjuterit.

Përgjate viteve, shumë studiues në mbarë botën kanë arritur të zhvillojnë teknika të ndryshme për klasifikimin dhe evidentimin e këtyre email-eve spam. Disa nga teknikat më të përdorura për zgjidhjen e këtij problemi janë DT (Pemët e vendimit), teknika probabilistike, SVM (makina vektoriale mbështetëse), apo ANN (rrjeti nervor artificial). Të gjitha këto teknika, kanë si qëllim primar përdorimin e disa metodave të flirtimit që nxjerrin në pah veçori të ndryshme për të lehtësuar identifikimin e këtyre email-eve, siç mund të jetë përdorimi më i shpeshtë i disa fjalëve të caktuara në emaile spam. Në varësi të frekuencës të ndeshjes së këtyre veçorive, mund të jemi në gjendje të përcaktojmë probabilitete për çdo veçori të pranishme në email dhe më pas, të krahasojmë këto rezultate më vlera kufitare të paracaktuara. Nëse vlera kufitare kalohet, atëhere emaili klasifikohet si spam.

Në këtë punim është bërë një krahasim në lidhje me performancën e disa algoritmeve më të përdorur të Machine Learning për evidentimin e email-eve spam. Ky studim është realizuar me ndihmën e softuerit WEKA (Waikato Environment for Knowledge Analysis). Pas analizimeve dhe ekzekutimeve të kryera, shohim se algoritmi Naïve Bajes kërkon një kohë shumë më të shkurtër për zgjidhjen e problemit, ndërsa, për sa i përket treguesit të Rasteve Korrekte të Klasifikimit (Correctly Classified Instances), algoritmi i Support Vector Machine arrin rezultatin më të lartë. Nga kjo arrijmë në konkluzionin se nuk ka një algoritëm të vetëm klasifikimi të Machine Learning që mund të na japë modelin më të mirë të klasifikimit të email-eve spam. Në studimet e mëtejshme në të ardhmen, sugjerohet ndërtimi i algoritmave të reja, duke marrë në të njëjtën kohë në konsideratë një databazë më të gjerë, në mënyrë që dhe përfundimet të kenë një marzh më të ulët gabimi.

Abstract (English)

For a long time, spam emails have caused several different problems for their recipients and the organizations associated with them. These unsolicited emails can lead to vulnerabilities to scams, issues with memory capacity management, and other computer parameters.

Over the years, many researchers around the world have developed various techniques for classifying and identifying these spam emails. Some of the most commonly used techniques are Decision Trees (DT), probabilistic techniques, Support Vector Machines (SVM), and Artificial Neural Networks (ANN). All these techniques primarily aim to utilize methods that highlight different features to facilitate the identification of spam emails, such as the frequent use of certain words in spam emails. Depending on the frequency with which these features appear, we may determine probabilities for each feature present in the email and then compare these results to predefined threshold values. If the threshold value is exceeded, the email is classified as spam.

In this paper, we compare the performance of some of the most widely used Machine Learning algorithms for identifying spam emails. This study was carried out using the WEKA (Waikato Environment for Knowledge Analysis) software. After the analyses and executions performed, we find that the Naïve Bayes algorithm requires much less time to solve the problem. However, in terms of the Correctly Classified Instances indicator, the Support Vector Machine algorithm achieves the highest result. From this, we conclude that there is no single Machine Learning classification algorithm that provides the best spam email classification model. In future studies, it is suggested to develop new algorithms and consider a larger database to reduce the margin of error in the conclusions.

Falënderime

Fillimisht dëshiroj të shpreh falenderime për pedagogen time udhëheqëse, Prof. Dr. Elinda Meçe, e cila më ka ndihmuar në mënyre konsistente për realizimin e kësaj projekt-diplome. Profesionalizmi i saj me ka frymëzuar dhe shtyrë drejt këtij studimi.

Gjithashtu shpreh falenderimet e mia për stafin pedagogjik të Fakultetit të Teknologjisë së Informacionit në Universitetin Politeknik të Tiranës, të cilët kanë kontribuar në formimin tim akademik përgjatë viteve të mia të studimit.

Falenderoj personin të cilës ia dedikoj njeriun që jam sot, mamin time, e cila më ka qëndruar pranë dhe motivuar gjatë gjithë rrugëtimit tim akademik.

Përmbajtja

Lista e Tabelave

KREU III

KREU IV

Lista e Figurave

KREU II

KREU III

KREU IV

KREU V

Shkurtime

ANN	Artificial Neural Networks
IA	Inteligjenca Artificiale
CNN	Convolutional Neural Networks
RNN	Recurrent Neural Networks
NLP	Natural Language Proccessing
WEKA	Waikato Environment for Knowledge Analysis
SVM	Support Vectorial Machine
DT	Decision Tree
ML	Machine Learning
NB	Naïve Bayes
ROC	Receiver Operating Characteristic

KREU I

HYRJE

Ky kapitull trajton idenë kryesore të këtij punimi, rëndësinë e problematikës së trajtuar, motivacionin, si dhe metodologjinë e punës ku përfshihen metodat e studimit dhe lloji i këtij studimi.

1.1. Problemi

Një email spam është çdo email i padëshiruar nga ana e marrësit, dhe personi që e dërgon këtë email spam do të konsiderohet si një dërgues email-esh spam/të padëshiruar. Spam mund të dërgohet nga persona fizikë, por më shpesh, ai dërgohet nga botnet-e, të cilat janë rrjete kompjuterash (bots ose spambots) të infektuar me malware. Përveç email-it, spam-i mund të shpërndahet edhe përmes mesazheve me tekst ose rrjeteve sociale. Ky problem është shumë i përhapur ditët e sotme, për vetë faktin se dërgimi i këtyre email-eve është lehtësisht i realizueshëm nga dërguesit. Kjo pasi ata mund të marrin adresat e emailit të individëve apo kompanive të ndryshme nga faqe interneti, viruse dhe chat rooms (Awad et al., 2017).

Spam-i mund të jetë një problem modern, por ai ka një histori që shkon pas disa dekadash. Email-i i parë i padëshiruar u dërgua në vitin 1978 nga Gary Thuerk, një punonjës i Digital Equipment Corp (DEC) për të promovuar një produkt të ri. Email-i spam u dërgua në 400 nga 2600 personat që kishin llogari të postës elektronike në Rrjetin e Agjencisë së Projekteve të Avancuara të Kërkimit. Disa raporte sugjerojnë se ai gjeneroi rreth 12 milionë dollarë nga shitjet e paligjshme për DEC.
Megjithatë, termi spam për herë të parë u përdor në vitin 1993. Ky term u përdor në Usenet, një kompani lajmesh, që u bë viktimë e sulmit të parë të spam-it në shkallë të gjerë. Deri në vitin 2003, spam-i përbënte 80% deri në 85% të mesazheve të postës elektronike të dërguara në mbarë botën. Tashmë ishte bërë një problem kaq i përhapur saqë detyroi që SHBA të miratonte Aktin e Kontrollit të Sulmit të Pornografisë dhe Marketingut të Pakërkuar (CAN-SPAM) të vitit 2003. CAN-SPAM është ende rregullorja më e rëndësishme që tregtarët legjitimë të postës elektronike duhet të respektojnë për të parandaluar etiketimin si spammers.
Midis mesit të vitit 2020 dhe fillimit të vitit 2021, sasia mesatare ditore e email-eve spam ra nga 316.39 miliardë në rreth 122 miliardë. Megjithatë, 85% e të gjitha emaileve janë ende të padëshiruara, gjë që u kushton bizneseve legjitime miliarda dollarë çdo vit (Microsoft, 2023).
Më kalimin e kohës, problemi i email-eve spam po bëhet akoma e më tepër serioz dhe po kërkon më shumë vëmendje për të gjetur mënyra të ndryshme për minimizim apo zgjidhje të këtij problemi. Email-et spam mund të sjellin shumë probleme për marrësit, nga të cilat mund të përmendim vështirësi në menaxhimin e kapacitetit të ruajtjes të së dhënave, vjedhje apo humbje të informacioneve personale dhe infektimit me viruse ose malëare të ndryshme. Sa më shumë email spam të merren, aq më i madh është dëmi që mund të shkaktohet në memorien e serverave të email-it dhe fuqisë së CPU të përdorur. Sasia e email-eve të padëshiruara po rritet çdo vit dhe përbën rreth 77% të të gjithë trafikut global të emaileve (Kaspersky, 2019). Duke u bazuar në statistikat e viteve të fundit publikuar nga “The New York Times”, vitin e kaluar janë humbur rreth 2.7 milion dollarë nga kompanitë për shkak të mashtrimeve nëpërmjet email-eve spam.

1.1.1. Rëndësia e zgjidhjes së problemit

Ka shumë raste të raportuara ku individët që marrin email spam janë bërë viktima të mashtrimeve në internet, duke pësuar humbje financiare të konsiderueshme. Kjo për shkak se përmbajtja e email mund të ketë link-e të cilat mund ti drejtojnë marrësit e email-eve në faqe që mund të vjedhin informacione bankare si Numri i Verifikimit të Bankës – BVN, numri i kartës bankare, si dhe fjalëkalime të ndryshme të përdorura nga ta.
Përdoruesit e email bien shpesh pre e këtyre mashtrimeve për vetë faktin se dërguesit e email-eve spam në shumë raste mund të shkruajnë duke pretenduar se përfaqësojnë kompani të njohura serioze dhe të besueshme. Kjo situatë mund të shkaktoje stres dhe bezdi për marrësit e email-eve, duke i shtyrë ata të marrin vendime pa menduar për pasojat.
Është shumë e rëndësishme që individët të jenë të kujdesshëm kur marrin email-e të panjohura dhe të mos japin kurrë informacione personale ose financiare pa verifikuar më parë identitetin e dërguesit. Algoritmet që përdoren nga kompjuterat për dedektimin e email-eve spam duhet që gjithashtu të jenë eficentë në mënyrë që humbjet të jenë sa më minimale. Ato duhet të jenë gjithmonë në përmirësim në mënyrë që t'iu përshtaten kërkesave të përdoruesve sa më mirë.
Pra dedektimi i email-eve spam është një problem shumë i rëndësishëm që çon në nevojën e implementimit të sistemeve automatike që mund të jenë të afta të bëjnë një filtrim të email-eve që konsiderohen spam.

1.2. Objektivat e punimit

Qëllimi i këtij punimi është paraqitja e një varianti zgjidhjeje për problemin e dedektimit të email-eve spam duke përdorur teknologjinë bashkëkohore Machine Learning dhe mjete të tjera që mundësohen nga Inteligjenca Artificiale. Në këtë punim diplome paraqitet një hyrje teorike në lidhje me konceptin bashkëkohor dhe shumë të përhapur të Inteligjencës Artificiale, duke u ndalur në Machine Learning dhe teknologjitë e tjera plotësuese që do të përdoren për zgjidhjen e problemit.

Problemi specifik që do të merret në shqyrtim gjatë këtij punimi është dedektimi i email-eve spam duke përdorur një bazë të dhënash me email-e spam dhe jospam për trajnimin e makinës.
Projekti i diplomës do të zhvillohet duke përdorur WEKA, e cila është një platformë e machine learning dhe një mjet i disponueshëm për të testuar një numër të gjerë algoritmesh. Krahas trajtimit teorik, do të përfshihet nje implementim dhe ekzekutim i këtyre algoritmeve në softuerin WEKA, që ka në bazë gjuhën e programimit Java, ku do të shohim gjithashtu dhe eficencën e algoritmave të përdorur dhe do të paraqitet një krahasim me algoritme të tjera ekzistuese për zgjidhjen e problemit të dedektimit të email-eve spam. Ky implementim si dhe krahasimet e mëtejshme do të trajtohen në detaj në kapitujt vijues.

KREU II

MACHINE LEARNING

Ky kapitull ka si qëllim primar njohjen me Inteligjencen Artificiale dhe algoritmave më të rëndësishëm të Machine learning, rëndësinë që këto fusha kanë dhe si mund të na ndihmojnë në zgjidhjen e problemit të dedektimit të email-eve spam.

2.1 Inteligjenca Artificiale

Inteligjenca Artificiale është një fushë në zhvillim konstant dhe bashkëkohore, por fillimet e saj i ka që në vitin 1950. Inteligjenca Artificiale (IA) është një degë e shkencës së kompjuterave që synon të krijojë sisteme që imitojnë, përmirësojnë ose tejkalojnë inteligjencën e njeriut. IA përvec se ka ndihmuar në mënyrë të drejtpërdrejtë zhvillimin e teknologjisë, ka ndikuar në shumë fusha të jetës së përditshme, duke përfshirë shëndetësinë, arsimin, transportin dhe tregtinë.
Në fillimet e zhvillimit të inteligjencës artificiale, ajo konsistonte vetëm në parashikime statistikore. Kjo për shkak se kompjuterët dhe pajisjet e tjera të asaj kohe ishin të limituara ne burime dhe mundësi që mund të ofronin për zhvillim.

2.1.1 Metodat statistikore që qëndrojnë në thelb të Inteligjencës Artificiale

Përdorimi i teknikave statistikore është thelbësor dhe përdoret nga Inteligjenca Artificiale për të mësuar nga të dhënat dhe për të parashikuar rezultate, dhe sa më të zhvilluara këto teknika statistikore të jenë, aq më të sakta janë parashikimet përfundimtare të realizuara nga algortimet e inteligjencës artificiale.
Fillimisht, inteligjenca artificiale u zhvillua duke përdorur teknika të hershme statistikore si për shembull interpolimi statistikor apo ekstrapolimi. Kjo metodë bën që të parashikohet një vlere tjetër në bashkësine e pikave apo jashtë bashkësisë së pikave të dhëna, bazuar gjithmonë në lidhjen mes çifteve (x, y). Me kalimin e kohës dhe zhvillimin e vazhdueshëm të teknologjisë, është bërë e mundur që të zhvillohen teknika të tjera dhe më komplekse statistikore në krahasim me ato të përdorura dikur.
Nga teknikat statistikore ekzistuese të inteligjencës artificiale, si më të rëndësishme mund të përmendim:

- **Regresioni** -> E cila është një teknikë statistikore që përdoret për të parashikuar një rezultat të vazhdueshëm në bazë të një ose më shumë variablave të pavarura. Në Inteligjencën Artificiale, regresioni përdoret për të trajnuar modele që mund të parashikojnë rezultate siç janë për shembull çmimet e shtëpive ose notat e studentëve.

- **Klasifikimi** -> Një teknikë statistikore që përdoret për të ndarë të dhënat në klasa të ndryshme. Në Inteligjecën Artificiale, klasifikimi përdoret për të trajnuar modele që mund të identifikojnë kategoritë e të dhënave, siç janë spam ose jo-spam për email-et, teknikë e cila do të përdoret gjatë këtij punimi diplome për shkak se është algoritëm më eficent për zgjidhjen e problemit.

- **Rrjetet Neuronale** (Neural Networks) -> Rrjetet neuronale janë modele që imitojnë mënyrën se si truri i njeriut përpunon informacionin. Ato përdorin në vetvete teknika të tjera statistikore për të mësuar nga të dhënat.

Teknikat e lartpërmendura më së shumti përdoren në rast se baza e të dhënave e përdorur në realizimin e projektit është e limituar.

2.2 Machine Learning

Një nga arsyet kryesore pse Machine Learning është kaq e rëndësishme është aftësia e saj për të trajtuar dhe kuptuar vëllime të mëdha të të dhënave. Ditët e sotme po ndeshemi me shumë të dhëna të cilat vijne nga mediat sociale, sensorët apo burime të tjera, dhe në këtë mënyrë metodat tradicionale të analizës së të dhënave janë bërë tanimë të papërshtatshme. Algoritmet e Machine Learning mund të përpunojnë këto sasi të mëdha të dhënash, të zbulojnë modele të fshehura dhe të ofrojnë njohuri të vlefshme që mund të nxisin vendimmarrjen.
Edhe pse koncepti i machine learning filloi të marrë formë në vitet 1950, aplikimi i tij në detektimin e spam emails mori zhvillim në fund të viteve 1990 dhe fillim të viteve 2000, kur email-i u bë një mjet komunikimi masiv.
Zhvillimi i algoritmeve të machine learning për detektimin e spamit ka kaluar nëpër disa faza, duke filluar me metoda të thjeshta si filtrimi i fjalëve kyç dhe ka arritur në përdorimin e teknikave më të avancuara si rrjetet neuronale dhe Deep Learning. Këto metoda janë bërë më të sofistikuara me kalimin e kohës, duke përfshirë analizën e tekstit dhe klasifikimin e mesazheve
Në thelb, një detektor spam i email-it përdor machine learning për të identifikuar dhe flirtuar mesazhet e padëshiruara përmes trajnimit të një modeli me një sërë email-esh të cilësuar si "spam" ose "jo-spam". Modeli mëson të njohë karakteristikat që ndajnë këto dy kategori dhe përdor këtë njohuri për të filtruar mesazhet e reja. Për shembull, një model mund të mësojë se email-et që përmbajnë fjalët si "lotari" ose "ofertë e limituar" janë shpesh spam.
Përdorimi i machine learning për detektimin e email spam ka **avantazhin e adaptueshmërisë**; që do të thotë se modeli mund të përshtatet me kohën për të identifikuar lloje të reja të spamit që nuk janë hasur më parë. Kjo është shumë e rëndësishme për shkak se taktikat e individëve që dërgojnë spam emails ndryshojnë vazhdimisht për të shmangur detektimin e tyre.

2.3 Deep Learning

Deep Learning i ka rrënjët e tij në vitet 1950 me zhvillimin e perceptronit (një lloj rrjeti neuronal artificial), por u bë i njohur në dekadën e fundit falë përparimeve në harduerin e kompjuterëve dhe sasisë të madhe të të dhënave. Teknologjia e deep learning ka evoluar shpejt, duke përfshirë zhvillimin e algoritmeve të reja si Convolutional Neural Networks (CNNs) dhe Recurrent Neural Networks (RNNs). Këto rrjete janë veçanërisht të dobishme në përpunimin e tekstit.
Në fushën e detektimit të spamit, deep learning filloi të ketë vend të rëndësishëm në fillim të viteve 2010, kur u kuptua se rrjetet neuronale mund të identifikojnë modele komplekse dhe të fshehura që ishin të vështira për t'u zbuluar nga algoritmet tradicionale. Në kontekstin e email spam, deep

learning përdoret për të trajnuar modele që mund të analizojnë dhe të klasifikojnë mesazhet bazuar në përmbajtjen e tyre. Kjo nuk përfshin vetëm fjalët kyçe, por edhe stilin e shkrimit, frekuencën e fjalëve, dhe madje edhe strukturën e mesazhit. Modeli i trajnuar mund të dallojë midis email-eve legjitimë dhe atyre të padëshiruar me një saktësi të lartë.

2.4 Inteligjenca Artificiale vs Machine Learning vs Deep Learning

Pavarësisht faktit se këto koncepte ngjajnë më njëra tjetrën dhe në shumë raste mund të konfuzohen, ekzistojnë ndryshime thelbësore mës Inteligjencës Artificiale, Machine Learning dhe Deep Learning.
Inteligjenca Artificiale është një fushë e gjerë që përfshin krijimin e makinave dhe softuerit që mund të punojnë dhe të mendojnë si njerëzit, e cila mund të përfshijë të kuptuarit e gjuhës, të kuptuarit e vizionit kompjuterik etj.
Ndërkohë Machine Learning është një nëndegë e Inteligjencës Artificiale, që përqendrohet më së shumti në zhvillimin e algoritmeve që lejojnë makinat të mësojnë nga të dhënat. Në vend që të programohen specifikisht për të kryer një detyrë, makinat mund të mësojnë më shumë nga të dhënat dhe të përmirësojnë performancën e tyre me kalimin e kohës.
Nga ana tjetër, Deep Learning është një nëndegë e Machine Learning që përdor **rrjetet neuronale** me shumë shtresa (deep neural netëorks), struktura të cilat janë të frymëzuara dhe bazohen te ndërtimi i trurit të njeriut. Këto modele janë të afta për të studiuar më shumë strukturat komplekse të të dhënave dhe janë thelbësore për shumë aplikacione të Inteligjencës Artificiale, përfshirë të kuptuarit e gjuhës natyrore dhe të kuptuarit e vizionit kompjuterik gjithashtu.
Kjo lidhje mes këtyre tre fushave mund të paraqitet më qartë me anë të kësaj skeme më poshtë:

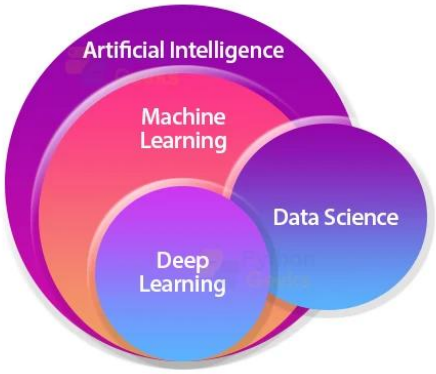


Figura 2.1 Shtresat e Inteligjencës Artificiale - ilustrim

2.5 Llojet e Machine Learning

Në bazë të metodave dhe mënyrës së funksionimit të Machine Learning, ajo ndahet kryesisht në katër lloje, të cilat janë:

- **Supervised Machine Learning**

Supervised Machine Learning, ashtu si dhe emri sugjeron, bazohet kryesisht në mbikëqyrje, pra monitorim të menyrës se si makina punon. Pra në këtë rast, makinat trajnohen duke përdorur grup të dhënash "të etiketuara" dhe bazuar në trajnimin që u bëhet këtyre pajisjeve, ato janë në gjendje të parashikojnë rezultatin përfundimtar. Në këtë rast të dhënat e etiketuara (labelled data) specifikojnë se disa nga inputet janë të lidhura në mënyrë direkte me daljen. Mund të themi se fillimisht ne trajnojmë makinën me hyrjen dhe daljen përkatëse, dhe më pas i kërkojmë makinës të parashikojë output duke përdorur grupin e të dhënave të testimit.
Shembull:
Supozojmë se kemi një grup të dhënash të imazheve të maceve dhe qenërve. Së pari, ne do ta trajnojmë makinën për të kuptuar imazhet, të tilla si forma dhe madhësia e bishtit të maces dhe qenit, forma e syve, ngjyra, gjatësia, duke marrë parasysh diferencat e maces dhe qenit. Pas përfundimit të trajnimit, ne mund të vendosim si input foton e një maceje dhe i kërkojmë makinës të identifikojë objektin dhe të parashikojë rezultatin. Duke qenë se në këtë moment makina është e trajnuar, ajo do të kontrollojë të gjitha tiparet, si lartësinë, formën, ngjyrën, sytë, veshët, bishtin, etj., dhe do të zbulojë se në këtë rast fotoja e marrë si input është një mace.
Qëllimi kryesor i supervised machine learning është të lidhë variablin hyrës (x) me variablin dalës (y).
Disa aplikacione që përdoren gjerësisht sot dhe kanë në bazë supervised machine learning janë vlerësimi i rrezikut, zbulimi i phishing, dedektimi i email-eve spam etj.
Nisur nga shpjegimi i mësipërm, shohim se rasti i marrë në studim në këtë projekt qëndron mbi bazën e konceptit të Supervised Machine Learning.

Ekzistojnë dy lloje të Supervised Machine Learning që janë:

a) Klasifikimi
Algoritmet e klasifikimit përdoren për të zgjidhur problemet e klasifikimit në të cilat variabli i daljes është kategorik, si për shembull "Po" ose “Jo”, “Mashkull” ose “Femër”, “E kuqe” ose “Blu”, “Spam” ose “Jo-spam”. Disa shembuj të algoritmeve të klasifikimit janë identifikimi spamit, filtrimi i email-eve, etj.

b) Regresioni
Algoritmet e regresionit përdoren për të zgjidhur problemet e regresionit në të cilat ekziston një marrëdhënie lineare midis ndryshoreve hyrëse dhe dalëse. Këto përdoren për të parashikuar variablat e prodhimit të vazhdueshëm, të tilla si tendencat e tregut, parashikimi i motit, etj.

Pavarësisht avantazheve të Supervised Machine learning në faktin se ajo punon me një bazë të dhënash fillestare, dhe në këtë mënyrë e ka më të thjeshtë parashikimin e rezultatit në bazë të rasteve të mëparshme ekzistojnë dhe disa disavantazhe në përdorimin e Supervised Machine Learning. Për shembull, këto algoritme nuk janë në gjendje të zgjidhin detyra komplekse dhe në disa raste mund të parashikojë në output të gabuar nëse të dhënat e testit janë të ndryshme nga të dhënat e trajnimit të makinës. Kjo metodë kërkon gjithashtu shumë kohë për të trajnuar algoritmin.

- **Unsupervised Machine Learning**

Unsupervised Machine Learning nuk ka nevojë për mbikëqyrje, pra në këtë rast makina trajnohet duke përdorur grup të dhënash të pa etiketuara dhe makina parashikon daljen pa asnjë mbikëqyrje. Në këtë rast, modelet trajnohen me të dhëna që nuk janë as të klasifikuara dhe as të etiketuara, dhe modeli vepron mbi ato të dhëna pa asnjë mbikëqyrje. Qëllimi kryesor i algoritmit është të grupojë ose kategorizojë të dhënat sipas ngjashmërive, modeleve dhe dallimeve. Makineritë udhëzohen të gjejnë modelet e fshehura nga të dhënat që jepen si input.
Për shembull, nëse supozojmë se kemi një bashkësi me imazhe perimesh dhe ne e japim këtë si input për makinën, edhe pse imazhet janë krejtësisht të panjohura për modelin në fjalë, makinë është në gjendje të gjejë modelet dhe kategoritë e objekteve. Ajo do të mundet të zbulojë modelet dhe ndryshimet e saj, të tilla si ndryshimi i ngjyrave, ndryshimi i formës dhe do të parashikojë rezultatin kur testohet me grupin e të dhënave testuese.

- **Reinforcement Learning**

Reinforcment Learning është një algoritëm më kompleks që funksionon në një proces të bazuar në feedback, në të cilin një agjent i IA (një komponent softuerik) eksploron automatikisht rrethinat e tij duke vepruar dhe gjurmuar, dhe duke mësuar nga përvojat ndërkohë që përmirëson performancën e tij.
Te reinforcement learning, nuk ka të dhëna të etiketuara si te supervised learning dhe agjentët mësojnë vetëm nga përvojat e tyre të mëparshme.
Procesi i reinforcement learning është i ngjashëm me procesin e zhvillimit të një njeriu, ku me kalimin e kohës ai zhvillohet në bazë të eksperiencave të ndryshme gjatë jetës së tij. Rezultatet që merren nga ky algoritëm janë më të sakta dhe afatgjata.
Disa shembuj të përdorimit të reinforcement learning në botën reale mund të përmendim video lojrat, robotika, text mining (që përdor NLP), etj.

2.6 Aplikimi i Machine Learning në dedektimin e email-eve spam

Për të luftuar efektivisht kërcënimin nga email apo mesazhet e padëshiruara, ofruesit e njohur të emailit si Gmail, Yahoo mail dhe Outlook kanë implementuar metoda të ndryshme të Machine Learning për të filtruar këto mesazhe.

Një nga metodat e përdorura është **Neural Networks**, të cilat siç u përmend më lart, janë sisteme kompjuterike të frymëzuara nga truri dhe struktura e sistemit nervor të njeriut. Këto metoda të Machine Leaning janë të afta të identifikojnë phishing dhe spam emails duke analizuar përmbajtjen e tyre në një varietet të gjerë kompjuterësh. Machine Learning është e aftë të funksionojë me eficencë të lartë në një varietet kushtesh, dhe për këtë arsye, filtrat e spam-it të Gmail dhe Yahoo Mail aplikojnë rregulla të paracaktuara për të kontrolluar email-et e padëshiruara. Këto rregulla janë të bazuara në analizën e të dhënave të mëparshme dhe vazhdojnë të përditësohen në mënyrë të vazhdueshme gjatë gjithë procesit të filtrimit.
Për shembull, modeli i machine learning i zhvilluar nga Google ka kapacitetin të zbulojë dhe të filtrojë mesazhet që kanë si qëllim phishing ose spam me një saktësi prej rreth 99.9%. Kjo do të

thotë se vetëm një në një mijë mesazhe të padëshiruara mund të shpëtojë nga filtri i përdorur nga Google / Gmail. Sipas të dhënave nga Google, 50-70% e të gjitha mesazheve që merr Gmail janë spam, një shifër kjo mjaft e konsiderueshme. Modelet e përdorura nga Google përfshijnë gjithashtu mjete si Google Safe Broësing për të identifikuar në këtë mënyrë faqet e internetit me URL të dëmshme, të cilat duhet të bllokohen në mënyrë që të vazhdohet navigimi i sigurt.
Aftësia e Google për të zbuluar email-et spam u rrit ndjeshëm me prezantimin e një sistemi që vonon disa mesazhe të Gmail për një periudhë të shkurtër për të kryer një kontroll më të hollësishëm paraprak, pasi këto mesazhe spam janë më të lehta për t'u identifikuar kur analizohen kolektivisht.
Synimi i analizës së këtyre mesazheve të dyshimta është që të bëhet një inspektim më i thelluar, dhe me rritjen e numrit të mesazheve që mbërrijnë në një moment të caktuar, algoritmet përditësohen në mënyrë dinamike. Është vërejtur se vetëm rreth 0.05% e mesazheve janë të prekura nga ky kontroll i detajuar shtesë.

Shumë hulumtues dhe autorë kanë sugjeruar metoda të ndryshme të klasifikimit të mesazheve të padëshiruara që i ndajnë të dhënat në kategori të veçanta. Këto metoda rrjedhin nga teoria e probabilitetit, teoria e grafeve, si pema e vendimeve (Decision Tree), sistemi imunitar artificial (ANN), makina mbështetëse vektoriale (SVM) (Bouguila dhe Amayri, 2009), dhe teknika e bazuar në raste (Fdez-Riverola et al., 2007).
Është demonstruar se për të aplikuar me sukses këto metoda të klasifikimit për të filtruar mesazhet e padëshiruara, duhet të përdoret një metodë që identifikon karakteristika të caktuara në spam emails, siç janë fjalët që përdoren më shpesh në këto mesazhe të padëshiruara. Prania e këtyre karakteristikave në një mesazh emaili përcakton probabilitetin e secilës prej tyre në email dhe më pas krahasohet me një kufi të paracaktuar nga ne. Të gjitha email-et që e tejkalojnë këtë prag konsiderohen si spam dhe është i domosdoshëm bllokimi i mbërritjes së tyre në destinacionin përkatës.

Support vector machine apo makina mbështetëse vektoriale (SVM) është një nga metodat më fitimprurëse dhe efikase për të trajtuar problemin e mesazheve të padëshiruara (Torabi et al., 2015). Këto janë kontrollues të modeleve të machine learning që analizojnë bazën e të dhënave dhe identifikojnë karakteristika specifike që do të na shërbejnë më vonë për shqyrtim, duke nxjerrë në pah lidhjen midis variablave që na interesojnë. SVM është një mjet për të gjetur një hiperplan në një hapësirë n-dimensionale që i ndan pikat në dy grupe, një në të dyja anët e planit. Hiperplani varet nga numri i veçorive. Për një hapësirë 2D, hiperplani është një vijë. Për një hapësirë 3D, një hiperplan është një plan 2D. SVM funksionon në një mënyrë që përpiqet të gjejë një hiperplan që mund të maksimizojë hapësirën midis pikave. Këto pika në hapësirë quhen vektorë mbështetës.
Teknikat e makinës vektoriale mbështetëse janë shumë efikase në zbulimin e karakteristikave të email-eve spam dhe në përcaktimin e rëndësisë së tyre për një kategori të caktuar të mesazheve. Këto kategori email-esh janë trajnuar më parë bazuar në përvojën dhe studimet e hulumtuesve, dhe rezultatet tregojnë se algoritmet e makinës mbështetëse vektoriale janë më efikase se shumë metoda të tjera të njohura për flirtimin e spam emails (Scholkopf dhe Smola, 2002).

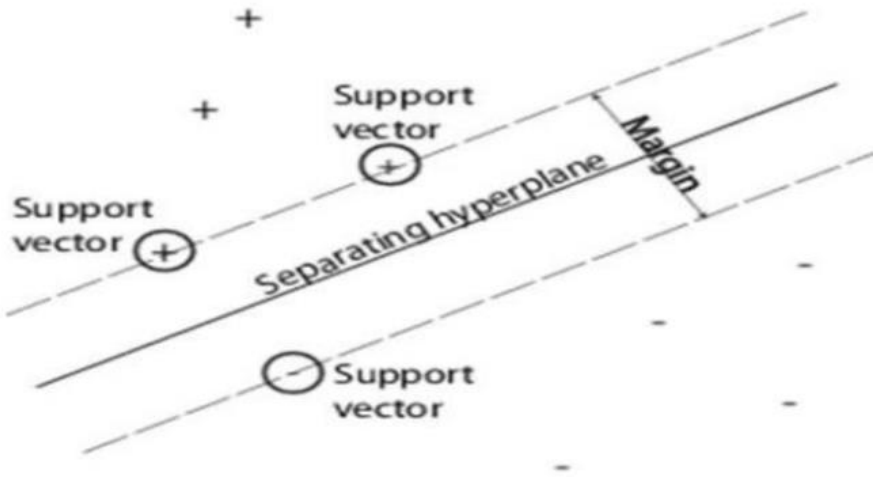


Figura 2.2 Support Vector Machine

Pema e vendimeve (DT) përfaqëson një tjetër algoritëm të Machine Learning që ka dhënë rezultate të mira në filtrimin e email-eve të padëshiruara. Ky algoritëm kërkon pak më shumë përfshirje nga përdoruesi gjatë trajnimit të të dhënave dhe kryen seleksionimin e karakteristikave të bazës të së dhënave të emailit. Një avantazh i dukshëm i këtij algoritmi është aftësia e tij për të trajtuar një gamë të gjerë të pyetjeve dhe konkluzioneve, duke ulur kështu gabimet në ekzekutimin e algoritmit.
Një tjetër përparësi e metodës së pemës së vendimeve krahasuar me metodat e tjera të machine learning është se të gjitha opsionet janë të disponueshme dhe paraqiten në një strukturë pemë, duke lejuar krahasimin direkt midis degëve dhe kulmeve të ndryshme të kësaj peme. Megjithatë, pavarësisht këtyre avantazheve të lartpërmendura, kontrolli i zgjerimit të vazhdueshëm i pemës mund të jetë i vështirë.
Algoritmi i pemës së vendimeve është një teknikë joparametrike e machine learning që është e ndikuar nga një sasi e madhe e të dhënave të trajnuara, duke e bërë atë në disa raste një klasifikues jo shumë të fortë dhe duke kufizuar saktësinë e tij në klasifikim.
Disa variante të tjera të algoritmit të pemës së vendimeve që mund të aplikohen me sukses në filtrimin e mesazheve të padëshiruara përfshijnë algoritmin C4.5 / J48 pema e vendimit (Masud dhe Rashedur, 2013) dhe Induksionin e Pemës së Modelit Logjistik (LMT) (Chakraborty dhe Mondal, 2012).

Te figura e mëposhtme paraqitet një ilustrim i përdorimit të Decision Tree ku çdo nyje korrespondon me një veçori (si për shembull fjalët specifike), dhe degët përfaqësojnë rezultatet që çojnë në vendimin përfundimtar: klasifikimin e një emaili si të padëshiruar ose jo të padëshiruar.

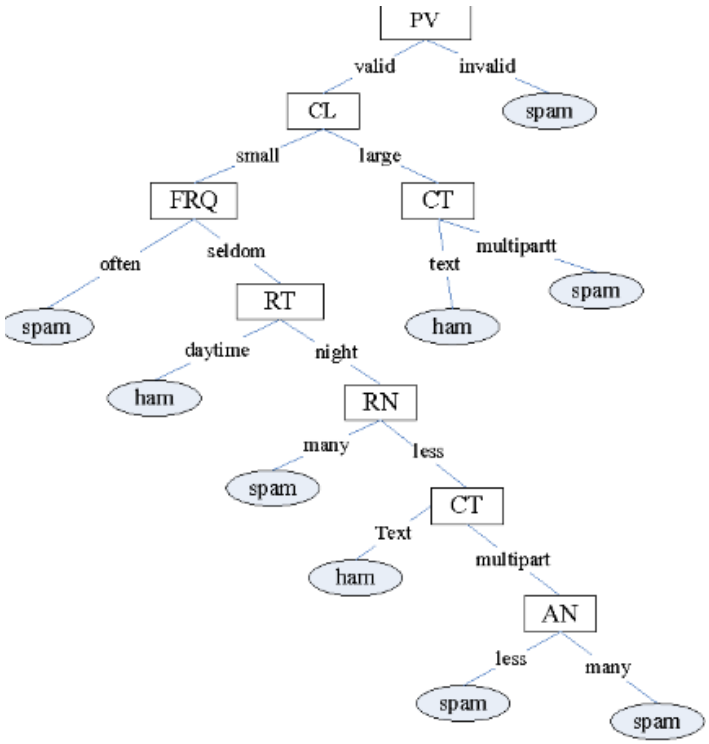


Figura 2.3 Decision Tree

Naïve Bayes (NB), sipas gjykimit nga shumica e autorëve, konsiderohet si algoritmi më i thjeshtë i machine learning që përdoret për të filtruar email-et e padëshiruara. Klasifikuesi Naïve Bayes bazohet te teorema e Bejesit, që është e njohur gjerësisht nga teoria e probabilitetit për klasifikimin e çdo email dhe **presupozon që të gjitha termat e përdshira në një mesazh email janë statistikisht të pavarur nga njëri-tjetri** (Androutsopoulos et al., 2000)**.** Naïve Bayes është një algoritëm mjaft i preferuar për shkak të thjeshtësisë së tij, lehtësisë së implementimit dhe konvergjencës së shpejtë në modele të kushtëzuara, siç janë algoritmet e tjera të machine learning (Rusland et al., 2017).

Për të marrë një informacion akoma më të thelluar që lidhet me një numër të madh algoritmesh të tjera të bazuara në Machine Learning për identifikimin e email-eve spam mund t'iu referohemi artikujve review (Dada et al. 2019, Cormack, 2006).

KREU III

PËRPUNIMI PARAPRAK

Në këtë kapitull do të flitet mbi metodat e përgatitjes paraprake të të dhënave. Kjo përfshin një transformim të nevojshëm të të dhënave të email-eve në një format që është i përshtatshëm për përpunim. Do të trajtojmë hapat kryesorë dhe sfidat që lidhen me këtë proces të përgatitjes, duke përfshirë analizën e tekstit, ndarjen në fjalë të veçanta, krijimin e sekuencave të fjalëve, identifikimin e formave bazë të fjalëve dhe shkronjave, zvogëlimin e kompleksitetit, paraqitjen e email-eve në formë vektoriale, dhe selektimin e atributeve më të rëndësishme.

3.1 Rëndësia e përpunimit paraprak të të dhënave

Në epokën digjitale të sotme, aplikohen një sërë metodash filtruese për të parandaluar mbërritjen e emaileve spam në inbox-in tonë. Një nga metodat më elementare të filtrimit është përdorimi i "listave të bardha" ose whitelist dhe "listave të zeza" ose blacklist, ku adresat e emailit lejohen ose bllokohen bazuar në kritere të caktuara.
Këto filtra janë të programuar për të bërë zgjidhjen e një problemi të qartë - nëse do të lejojnë hyrjen e një emaili apo jo. Për shembull, një shtresë filtruese mund të përfshijë një rregull të caktuar të vendosur nga përdoruesi ose një listë të adresave të emailit të përcaktuar nga përdoruesi për të rishikuar dhe vendosur nëse një email i ri duhet të bllokohet apo të pranohet. Është e rëndësishme të kuptojmë se nëse një email arrin tek një shtresë e tillë, ai ka kaluar tashmë nëpër të gjitha filtrat e mëparshëm.
Kështu, nëse një email arrin tek përdoruesi, ai ka kaluar suksesshëm nëpër të gjitha nivelet e filtrimit, njëri pas tjetrit. Fokusi ynë do të jetë në një shtresë specifike që skanon për modele të padëshiruara në të gjithë përmbajtjen e emailit, duke aplikuar fillimisht një filtrim të përgjithshëm mbi të dhënat e emailit spam dhe më pas duke identifikuar modele specifike të spamit. Filtrimi i tillë zakonisht bazohet në teknikat e machine learning, të cilat janë ndër aplikacionet më të suksesshme në fushën e teknologjisë së informacionit së fundmi. Në këtë tezë, ne do të analizojmë dhe vlerësojmë disa nga këto algoritme të machine learning, të cilat janë implementuar në platformën WEKA, për të trajtuar dhe testuar efikasitetin e tyre në identifikimin e emaileve spam duke përdorur një bazë të dhënash të mëparshme të spamit.

3.2 Hapat e përpunimit të mesazhit

Nga katër algoritmet e përdorura për filtrimin e postës së padëshiruar të cilat janë përmendur dhe më lart, tre prej tyre – algoritmi Naïve Bayes, algoritmi i pemës vendimmarrëse C4.5 dhe algoritmi i Support Vector Machine – kërkojnë përpunim paraprak si gjatë trajnimit të filtrit ashtu edhe gjatë klasifikimit të mesazheve hyrëse. Këto algoritme kanë nevojë për një paraqitje strukturore të secilit mesazh për të mundësuar trajnimin dhe klasifikimin e saktë. Për këtë qëllim, është përdorur një bashkësi fjalësh, e njohur edhe si modeli i hapësirës vektoriale, një qasje e zakonshme për këtë problem (Guzella dhe Caminhas, 2009).

Përpunimi paraprak i një mesazhi për të krijuar një paraqitje të përshtatshme që klasifikuesit të mund ta interpretojnë, përfshin katër hapa kryesorë. Siç tregohet në figurën 2.1, këto hapa përfshijnë: marrjen e mesazhit hyrës "I am coming", tokenizimin ku mesazhi ndahet në fjalë, përcaktimin e rrënjëve ku ruhen vetëm format bazë të fjalëve dhe, së fundmi, paraqitjen ku çdo fjalë vendoset në një pozicion të caktuar brenda një vektori veçorish.

Pra, katër hapat e nevojshëm për përpunimin paraprak të një mesazhi që klasifikuesit të mund ta analizojnë janë: emaili hyrës, tokenizimi, përcaktimi i rrënjëve dhe paraqitja. Në seksionet e mëtejshme, do t'i shqyrtojmë këto hapa një nga një.

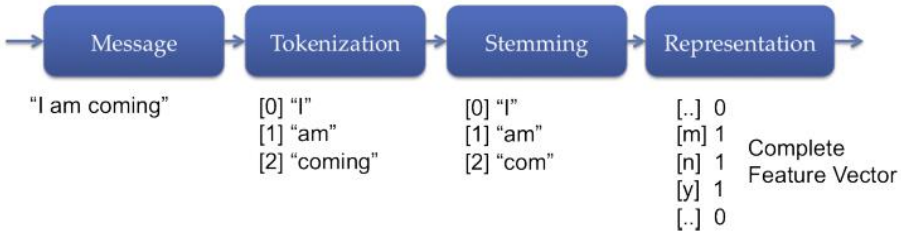


Figura 3.1 Etapat e para - përpunimit

3.3 Mesazhi

Ka disa aspekte kryesore që mund të ndryshojnë strukturalisht midis mesazheve, përveç përmbajtjes së tyre. Një ndryshim i mundshëm është kodimi i karaktereve për kompjuter. Zgjedhja e kodimit ndikon në hapësirën e karaktereve, madhësinë dhe kompleksitetin e paraqitjes së tyre.

Ka shumë mënyra të kodimeve të karaktereve. Është e rëndësishme të përdorim një kodim të përbashkët gjatë komunikimit për të siguruar që të lexojmë përmbajtjen e një mesazhi në hyrje siç duhet. Disa nga kodimet më të zakonshme janë ASCII dhe UTF-8. Për mesazhet e emaileve në hyrje, mund të përdoret kodimi UCS-2, një format i thjeshtë me gjatësi fikse që përdor sistemin me 2 bajt për të përfaqësuar çdo karakter. Ky kodim mbështetet në shumicën e gjuhëve të tjera moderne, duke përfshirë shkrimin arab, kinezisht dhe koreanisht.

Përveç kodimit të karaktereve, një aspekt tjetër kryesor është gjuha në të cilën është shkruar mesazhi. Në varësi të gjuhës së përdorur, mund të ketë pjesë të caktuara në një filtrues ku mund të dështojmë. Për shembull, nëse filtri është trajnuar për të analizuar përmbajtjen e mesazheve në suedisht, por mesazhi që vjen është i shkruar në anglisht, filtri mund të konfuzohet pa përdorur të dhëna të mëparshme të trajnimit.

3.4 Tokenizimi

Tokenizimi është hapi i parë i përpunimit ku një mesazh i plotë ndahet në pjesë më të vogla duke gjetur karaktere të vetme ose modele më të gjata që korrespondojnë me një kufizues të caktuar. Siç shihet në figurën 3.1, mesazhi i sapoardhur "I am coming" është ndarë në tokenet "I", "am", "coming" me kufizues me hapësira boshe (space).

Për tokenizimin e mesazheve, duhet të supozohet se mesazhet që do të filtrohen janë nga teksti romak. Ky supozim është i rëndësishëm pasi disa gjuhë janë shumë të vështira për t'u vecuar dhe analizuar për shkak të strukturës së tyre komplekse. Për shembull, gjuha e shkruar kineze nuk përdor domosdoshmërisht aq shumë përcaktues të dukshëm sa në alfabetin latin (p.sh., hapësira e bardhë ose një pikë pyetje). Prandaj në këto raste të vecanta është e vështirë të tokenizosh një fjali, sepse fjalia mund të jetë vetëm një fjalë e gjatë dhe sistemi mund të konfuzohet për të përcaktuar ku do të ndahet mesazhi.

Kufizuesit që përdoren janë të përshtatur nga ky supozim. Për të gjetur tokenet përkatëse nga një tëkst i marrë në shqyrtim, duhet të përdoren kufizues të zakonshëm për fjalë si hapësira dhe presje, të tilla si këto që janë paraqitur më poshtë:

\r\n\t. , ; : " " ( ) ? !.

3.5 *N* - gramët

N-gramët janë një përzierje e N elementëve nga një varg, në këtë rast një sekuence teksti të shkruar me fjalë ose karaktere. Qëllimi primar është të zgjerohet një fushë më e gjerë fjalësh për të kapur më shumë informacion duke identifikuar fjalë që zakonisht janë afër njëra-tjetrës në një lloj mesazhi të caktuar. Madhësia e N-gramëve në filtrat zakonisht është midis një (unigram), dy (bigram) ose tre (trigram). Edhe pse nuk kemi ndonjë pengesë për të pasur N-gramë më të mëdha se këto, duket se fusha e fjalëve nuk është e lehtë për t'u menaxhuar dhe kemi nevojë për më shumë të dhëna për t'u trajnuar.

Në vlerësimin e performancës së filtrave të ndryshëm në këtë punim diplome, n- gramët e fjalëve deri në madhësinë prej dy janë përdorur që të tre filtrat të përdorin vektorët e veçorive për të vepruar, për të testuar dhe për të gjetur një konfigurim të përshtatshëm për këtë problem.

3.6 Rrënjët e fjalëve dhe rastet e veçanta

Identifikimi i rrënjëve është një veprim opsional pasi një mesazh është tokenizuar, ku çdo token i fjalëve të mesazhit shndërrohet në formën e tij të rrënjës morfologjike (Cormack, 2006). Megjithatë, një sfidë me rrënjët është se ato janë të varura nga gjuha në të cilën janë formuar, e cila mund të sjellë rezultate të pasakta ose asnjë rezultat te raste te tjera, nëse do të vinte një mesazh nga një gjuhë e ndryshme. Një ilustrim tregohet më poshtë se si fjalët e ndryshme rrjedhin të gjitha nga e njëjta formë e rrënjës morfologjike, që është fjala "catch". Pra kemi Catching = = Catch, Catched = = Catch, Catcher = = Catch.

Zgjedhja, në qoftë se përfaqësimi me shkronja të mëdha dhe të vogla të së njëjtës shkronjë është i dallueshëm apo jo, është një tjetër nga hapat opsionalë në përpunimin paraprak me të njëjtin qëllim si përcaktimi i rrënjëve. Ky veprim ka për qëllim të ulë dimensionin e hapësirës së fjalëve ashtu si dhe të përmirësojë saktësinë e parashikimit për klasifikuesit duke kapërcyer problemin e mungeses së shpeshtesisë së të dhënave në rast se të dhënat e trajnuara mund të jenë shumë të vogla në krahasim me dimensionin e hapësirës së fjalëve (Cormack, 2006).

Rrënjët e fjalëve dhe përfaqësimi me shkronja të mëdha dhe të vogla kanë treguar "rezultate të pacaktuara në marrjen dhe filtrimin e informacionit" (Cormack, 2006) për filtrimin e postës elektronike të padëshiruar. Me kalimin e kohës u mor vendim për të përdorur vetëm shkronja të vogla në paraqitjen e shenjave për të zvogëluar hapësirën e fjalëve, ndërsa rrënjët e fjalëve angleze u përdorën në gjysmën e testeve për të analizuar se cili përfaqësim i tokeneve tregon performancën më të mirë.

3.7 Dobësitë

Një disavantazh i dukshëm kur ne përdorim vektorët e veçorive është mundësia që një dërgues keqdashës i emailit të shmangë përfshirjen e fjalëve në vektorin e veçorive duke shkruar fjalë të zakonshme të emailit të padëshiruar në mënyra të ndryshme. Për shembull, pët ta konkretizuar, nëse marrim në studim rastin e mesazhit të padëshiruar si "Viagra", karakteri "I" mund të përfaqësohet në të paktën 12 mënyra të ndryshme "I, i, 1, l, k,¨ı,`i, ´i, :, `I, ´I or ¨ı". E gjithë fjala mund të ndryshohet gjithashtu duke futur karaktere të huaja si "V_i_a_g_r_a", që e bën sasinë totale të kombinimeve të paktën 1,300,925,111,156,286,160,896 (Hajes, 2007).

Një metodë tjetër e përdorur për të provuar dhe ulur efektivitetin e një filtri duke marrë parasysh faktin në qoftë se ai është një filtër bajesian, do të ishte **Bajesiani puasonian**. Kjo metodë ka si qëllim futjen e një fjale specifike në mesazhin e dërguar, të cilat do të ndryshonin dukshëm saktësinë e klasifikuesit. Personi i identifikuar si një dërgues i padëshiruar, i cili përpiqet dukshëm të shkaktojë një Bajesian puasonian, në këtë rast do të përpiqet ta mbushë mesazhin me fjalë që zakonisht nuk gjenden në një mesazh të padëshiruar, në mënyrë që të konfuzojë filtrat e përdorur.

3.8 Paraqitja e mesazheve

Paraqitja vektoriale e mesazheve tregon se si duhet të formatohet një mesazh në mënyrë që klasifikuesi i marrë në shqyrtim të mund të kuptojë dhe të analizojë se cilat veçori dalluese ka çdo mesazh. Një shembull tipik i kësaj paraqitjeje është përfaqësimi me bashkësinë e fjalëve që do të përdoret në këtë punim.

Paraqitja është një vektor i veçorive N - dimensional, domethënë një vektor me çdo bosht (apo tipar) që përfaqëson një fjalë specifike ose fjali më të gjatë dhe vlerën për secilën veçori në varësi të faktit në qoftë se ka ndonjë token në mesazhin në fjalë apo jo. Ka dy metoda të zakonshme për të përfaqësuar vektorin e veçorive. E para është metoda binare, ku çdo veçori në vektorin e veçorive përfaqëson nëse një fjalë ose fjali ekziston ose jo në mesazh, duke përdorur simbolet 0

dhe 1. E dyta është metoda numerike e vektorit të veçorive, e cila tregon numrin e herëve që një fjalë ose fjali është përsëritur në mesazh, duke përdorur numra të ndryshëm në varësi të shpeshtësisë që këto fjalë gjenden në mesazh, apo në këtë rast në email-in që po merret në shqyrtim për ta përcatuar nëse është spam apo jo. Me anë të tabelës 3.1 tregohet një paraqitje e tillë për mesazhin e thjeshtë "tre plus tre është gjashtë".

"tre plus tre është gjashtë"

Binare		Numerike	
…..	0	…..	0
tre	1	tre	2
plus	1	plus	1
është	1	është	1
gjashtë	1	gjashtë	1
…..	0	…..	0

Tabela 3.1 Diferenca e vektorit të veçorive për mesazhin kur përdoret një paraqitje binare dhe një paraqitje numerike.

Siç mund të vërehet në tabelën 3.1, vektori binar i veçorive përmban ose numrin 0 ose numrin 1 për çdo veçori, ndërsa vektori numerik i veçorive regjistron numrin e përgjithshëm të rasteve për çdo veçori. Është e qartë se mesazhi ka 2 tokene që korrespondon me veçorinë "three" dhe shembulli numerik e pasqyron këtë rezultat ashtu siç pritej. Të gjitha veçoritë e tjera që nuk korrespondonin me ndonjë token në mesazh janë vendosur në vlerën e tyre të paracaktuar prej 0.

Pas procesimit paraprak të të gjitha të dhënave të trajnimit nga filtri, çdo veçori do të ketë një pozicion të përhershëm në vektorin e veçorive, kështu që çdo klasifikues që bazohet në këtë paraqitje do të ketë gjithmonë një paraqitje të njëjtë për çdo mesazh hyrës, me përjashtim të ndryshimit në numërimin e veçorive, e cila është e vetmja gjë që do të ndryshojë për çdo vektor të veçorive.

Një filtër spam mund të trajnohet në një numër të madh mesazhesh dhe dimensioni i vektorit të veçorive mund të zgjerohet me çdo mesazh për të përfaqësuar çdo token të vetëm nga çdo mesazh. Kjo mund të jetë një problem jo vetëm për ruajtjen e hapësirës dhe kujtesës, por edhe për një rënie të caktuar të performancës në shpejtësi në shumë filtra kur dimensioni i vektorit zgjerohet shumë, si dhe nevojën për më shumë të dhëna trajnimi për më shumë veçori. Për të luftuar këtë dhe për të zvogëluar dimensionin e vektorit të veçorive, përdoret një përzgjedhje e veçorive (Cormack, 2006).

3.9 Përzgjedhja e veçorive - Entropia

Seleksioni i veçorive është një teknikë që zvogëlon numrin e veçorive në përfaqësim. Kjo teknikë synon të reduktojë numrin e veçorive në model, duke ruajtur ato që janë më të rëndësishme për një klasifikim të saktë. Veçoritë që konsiderohen më të rëndësishme zakonisht janë ato që shfaqen më shpesh në një kategori mesazhesh sesa në një tjetër. Për shembull, një fjalë si “PRIZE” shfaqet më shpesh në mesazhe spam sesa në mesazhe të zakonshme. Duke njohur këtë, ajo mund të zgjidhet si një nga veçoritë që duhet të merren parasysh.

Ne kemi zgjedhur të përdorim fitimin e informacionit si algoritmin e seleksionit të veçorive në këtë studim, pasi është i përdorshëm dhe i lehtë për t’u kuptuar. Fitimi i informacionit ofron një vlerësim të aftësisë së një veçorie të caktuar në vektorin e veçorive për të klasifikuar mesazhet. Për shembull, për një grup mesazhesh spam dhe një grup mesazhesh të zakonshme, entropia e grupit origjinal krahasohet me entropinë e grupeve pas ndarjes së veçorive. Sa më të ndara të jenë grupet mes mesazheve të zakonshme dhe mesazheve spam, aq më i lartë dhe eficent do të jetë vlerësimi.

Entropia, e cila përdoret për të gjetur fitimin e informacionit për një veçori, mund të kuptohet si një masë që tregon se sa homogjen apo i përzier është një koleksion mesazhesh të klasifikuara. Sipas përkufizimit, entropia është "një masë e 'pasigurisë' ose 'rastësisë' së një fenomeni të rastësishëm" (Heylighen dhe Joslyn, 2001). Një entropi e ulët tregon që një koleksion është shumë homogjen, ndërsa një entropi e lartë tregon që është më i përzier.

Në përzgjedhjen e veçorive, synojmë të gjejmë ato veçori që ndajnë koleksionin tonë të mesazheve në grupe sa më homogjene të jetë e mundur. Për shembull, nëse kemi një grup me mesazhe të padëshiruara dhe mesazhe të ligjshme, duam të gjejmë një veçori që është e zakonshme në mesazhet e padëshiruara dhe jo në ato të ligjshme. Duke marrë si shembull fjalën "PRIZE", e cila është e zakonshme në mesazhe spam, duam t'i grupojmë të gjitha mesazhet që përmbajnë fjalën "PRIZE" në një grup dhe pjesën tjetër të mesazheve në një grup tjetër.

Tipi	PRIZE
Spam	Po
Legitimate	Jo
Legitimate	Po
Legitimate	Jo
Spam	Po
Spam	Po
Spam	Po

Spam	Po

Tabela 3.2 Koleksion i mesazheve të klasifikuara të llojit të padëshiruar (spam) ose legjitim të cilat mund të përmbajnë ose jo fjalën PRIZE

Tani marrim dy grupe të dhënash dhe, nëse supozimi ynë është i saktë, njëri grup duhet të ketë një përqindje më të lartë të mesazheve të padëshiruara krahasuar me grupin origjinal, ndërsa tjetri duhet të ketë një përqindje më të lartë të mesazheve të ligjshme sesa grupi origjinal.

Për të llogaritur entropinë, përdorim formulën e mëposhtme:

$$H(X) = -\sum_{i=1}^{n} P(x_i) \log_2 P(x_i)$$

ku n është numri total i emaileve në dalje (që mund të jenë të padëshiruar ose legjitimë), $P(x_i)$ është probabiliteti që veçoria X i takon klasës $i = 1,..,k$. Në shembullin tonë kemi $n = 8, P(x_1) = \frac{5}{8}, P(x_2) = \frac{3}{8}$. (3 -> numri i email legjitime ; 5 -> numri i email-eve spam)

Për të gjetur entropinë e koleksioneve origjinale mund të llogaritim:

$$H(Spam) = -(\frac{5}{8}\log_2 \frac{5}{8} + \frac{3}{8}\log_2 \frac{3}{8}) = 0.95$$

dhe entropia e re, në qoftë se ne heqim mesazhet që nuk përmbajnë fjalën “PRIZE” është

$$H(prise = Po) = -(\frac{5}{6}\log_2 \frac{5}{6} + \frac{1}{6}\log_2 \frac{1}{6}) = 0.65$$

ku në këtë rast të ri kemi: $n = 6, P(x_1) = \frac{5}{6}, P(x_2) = \frac{1}{6}$

Nga ana tjetër, në qoftë se llogarisim vetëm mesazhet që nuk përmbajnë fjalën “PRIZE” do të kishim llogaritjen e mëposhtme në bazë të formulës së mësipërme:

$$H(prise = Jo) = -(\frac{0}{2}\log_2 \frac{0}{2} + \frac{2}{2}\log_2 \frac{2}{2}) = 0$$

ku $n = 2, P(x_1) = 0, P(x_2) = 1$.

Kemi $\frac{0}{2}\log_2 \frac{0}{2} = 0$ sepse $\lim_{x \to 0^+} x \log_2 x = 0$

Nga rezultatet mund të shohim se dy grupet e reja kanë një entropi më të ulët se grupi origjinal, gjë që tregon se tipari PRIZE mund të konsiderohet të jetë një veçori e mirë për t'u zgjedhur për

qëllimin e klasifikimit të email-eve legjitimë. Ky është një shembull i thjeshtuar që përdor vetëm vlera binare për veçoritë, dhe për këtë arsye kërkohet vetëm një test për veçorinë.
Për një veçori numerike do të ishte e nevojshme të ndahej në çdo vlerë të disponueshme të veçorisë si $x \leq$ dhe $x >$, ku x është një vlerë që mund të ketë veçoria.

Fitimi i informacionit përdoret për të vlerësuar sa shumë një veçori specifike kontribuon në reduktimin e entropisë së përgjithshme kur klasifikojmë mesazhet.

Formula për fitimin e informacionit është:

$$IG(F, fe) = H(F) - H(F, fe).$$

Ky është reduktimi i pritshëm i entropisë së shpërndarjes objektive F kur zgjidhet veçoria fe. $H(F)$ është entropia për koleksionin fillestar të marrë në studim dhe $H(F \mid fe)$ është entropia e koleksioneve të reja që vjen pas ndarjes sipas fe. Shohim se fitimi i informacionit për një veçori A përkufizohet si ndryshimi i entropisë së sistemit përpara dhe pas ndarjes së të dhënave sipas vlerave të veçorisë A. Rrjedhimisht, dalim në rezultatin se për sa kohë që fitimi i informacionit është më i madh se zero, do të thotë se nëse do të ndodhte një ndarje mbi atë veçori të caktuar (Pra $H(F) > H(F, fe)$), koleksionet e reja do të ishin më homogjene se koleksioni origjinal.

Veçoritë e marra në konsideratë ishin të gjitha të vlerave binare ose të vazhdueshme. Për veçoritë binare, mjafton vetëm një kalim për të llogaritur fitimin e informacionit. Ndërsa për veçoritë e vazhdueshme, është e nevojshme të bëhen teste për të gjitha ndarjet e mundshme binare të veçorisë. Për shembull, nëse diapazoni i vlerave që mund të marrë një veçori është 1, 2 dhe 3, do të nevojiten dy teste për të llogaritur fitimin e informacionit për ndarjet binare: njëri test do të jetë për ndarjen {1} dhe {2,3}, dhe tjetri për ndarjen {1,2} dhe {3}.

Pasi të kemi llogaritur fitimin e informacionit për çdo veçori të disponueshme, mund të rendisim lehtësisht veçoritë nga fitimi më i lartë në më të ulët për të identifikuar ato elemente që përfaqësojnë më mirë një mesazh të padëshiruar ose të ligjshëm. Pasi të kemi bërë këtë, do të zgjedhim një numër të kufizuar veçorish që shpresojmë se përfaqësojnë më së miri secilin nga llojet e ndryshme të mesazheve.

3.10 Përfundime të procesit të para-përpunimit

Të dhënat për paraqitjen e tekstit mund të kodohen në mënyra të ndryshme, duke marrë parasysh alfabetet që dëshirojmë të na mbështesin dhe qëllimet që kemi. Në këtë punim, përdoret një kodim 2-bajtësh i quajtur UCS-2 për të gjitha testet e filtrave të ndryshëm. Ky kodim mbështet të gjithë alfabetet kryesore në botë, por supozohet se të dhënat përmbajnë kryesisht karaktere të alfabetit latin.

Para se çdo e dhënë të futet në një filtër, kryhet një përpunim paraprak, ashtu si u shpjegua më sipër me detaje, që përfshin tokenizimin dhe, në mënyrë opsionale, ndërtimin e n-gramëve dhe gjetjen e rrënjëve, për një klasifikim më të lehtë. Tokenizimi ndan një mesazh në pjesë më të vogla të quajtura tokene. Këto ndarje bëhen sipas karaktereve ose modeleve të caktuara, duke rezultuar në disa nënvargje pas përfundimit të procesit të tokenizimit.

Procesi n-gram shërben për të krijuar tokene të reja duke kaluar kështu nëpër sekuencën ekzistuese të tokeneve dhe duke kombinuar tokenin aktual me n tokene para tij në çdo hap. Ky proces krijon një hapësirë më të madhe fjalësh, që mund të japë më shumë informacion nga një mesazh. Duke qenë se n-gramët mund të jenë të çdo madhësie pasi nuk ekziston një kufizim në madhësi, një hapësirë shumë e madhe fjalësh bëhet e pamenaxhueshme për kompjuterin që bën këtë ndarje.

Gjetja e rrënjëve është e kundërta e procesit të n-gram. Kjo përpiqet të zvogëlojë hapësirën e fjalëve duke mbajtur vetëm rrënjën e një fjale. Por problemi te kjo metodë është që një hapësirë shumë e madhe fjalësh mund të jetë e vështirë për të trajnuar një klasifikues. Një hapësirë më e madhe fjalësh gjithashtu nga ana tjetër kërkon më shumë të dhëna trajnimi.

Tokenizimi për mesazhet në këtë punim përdor vetëm kufijtë më tipikë të përdorur në tekstin e shkruar, duke përfshirë hapësirën boshe, pikëpyetjen dhe pikën. Në gjysmën e testeve është bërë gjetja e rrënjëve, e përshtatur për gjuhën angleze. Kjo është përdorur për të zvogëluar hapësirën e fjalëve me koston e humbjes së informacionit, për të testuar nëse kjo përmirëson performancën për një numër të madh veçorish.

Janë kryer gjithashtu teste me n-gramë të madhësive të ndryshme për të rritur hapësirën e fjalëve dhe për të shtuar informacionin e disponueshëm në çdo mesazh, për të parë se si kjo ndikon në rezultatet e filtrit. Për përzgjedhjen e veçorive, është përdorur fitimi i informacionit për të renditur veçoritë dhe për të përdorur ato më të rëndësishmet për domenin specifik. Kjo metodë është e zakonshme në filtrimin e postës së padëshiruar kur zgjedhim veçoritë për vektorin e veçorive. Fitimi i informacionit përdoret gjithashtu në algoritmin C4.5 gjatë ndërtimit të pemës së tij të vendimit.

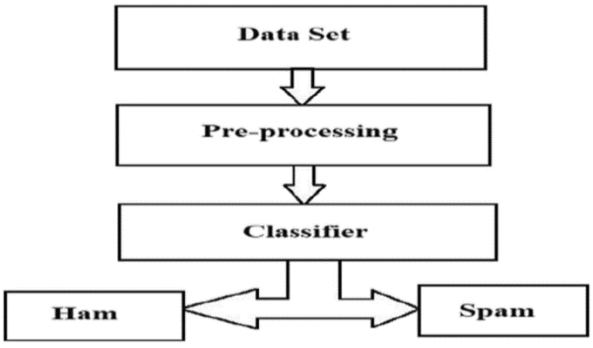


Figura 3.2 Diagramë e rrugës së klasifikimit të email-eve

KREU IV

ANALIZA E ALGORITMEVE TË KLASIFIKIMIT

Ky kapitull ka në fokus përshkrimin më në detaj të secilës prej algoritmave të Machine Learning duke u fokusuar te mënyra se si ato trajnohen me të dhëna dhe më pas si ndodh klasifikimi i email-eve.

4.1 Algoritmi i Naïve Bajes

Algoritmi Naive Bayes, i përdorur për herë të parë në filtrimin e emaileve të padëshiruar në vitin 1998, çoi në zhvillimin dhe përdorimin praktik të shumë algoritmeve të tjera të machine learning (Guzella dhe Caminhas, 2009). Implementimi i algoritmit Naïve Bayes në machine learning është shumë i thjeshtë, mund të jetë mjaft efektiv nga pikëpamja e llogaritjeve dhe të demonstrojë një saktësi të konsiderueshme të parashikimit duke u krahasuar me thjeshtësisë e tij. Megjithëse në kohët e sotme ky algoritëm është zëvendësuar me algortima të tjerë më fitimprurës, studiuesit zakonisht e përdorin atë si bazë për algoritme të tjera, dhe kjo është një nga arsyet është zgjedhur për t'u përdorur në këtë studim.

Formula e Naive Bayes mbështetet në teoremën e Bayes-it, por supozon pavarësi të kushtëzuar. Kjo do të thotë që veçoritë janë krejtësisht të pavarura nga njëra-tjetra kur llogaritet probabiliteti i kushtëzuar për to. Edhe pse ky supozim shpesh nuk është i saktë, është dëshmuar se klasifikimi duke përdorur këtë metodë shpesh funksionon mjaft mirë. Është vendosur të përdoret klasifikuesi multinomial Naive Bayes, pasi ai duket i përshtatshëm për klasifikimin e dokumenteve tekst dhe merr parasysh numërimin e fjalëve që i përshtatet një vektori të veçorive numerike të përdorura në këtë studim.

Dimë se nga teoria e probabilitetit, formula e Bajes për probabilitetin e kushtëzuar është:

$$P(K|X) = \frac{P(K)P(X|K)}{P(X)}$$

Prej nga ku në qoftë se Xështë vektori i veçorive, K është klasa e pritshme. Meqenëse emëruesi është një numër konstant, ne jemi të interesuar vetëm për vlerën numerike të numëruesit. Në këtë rast ne përdorim relacionin që vijon:

$$P(C_j|X) = P(x_1, x_2, \ldots, x_n|C_j)P(C_j),$$

ku x_n janë elementët e vektorit të veçorive, n është numri i veçorive nga 1 deri tek dimensioni i vektorit të veçorive dhe C_j është klasa e j -të (për shembull klasa e emaileve spam ose klasa e emaileve legjitimë). Nga llogaritja e probabilitetit të C_j me kusht X dhe bazuar në supozimin tonë që veçoritë e X janë të pavarura me kusht për Naïve Bajes, ne mund të shkruajmë relacionin e mësipërm në trajtën e mëposhtme:

$$P(x_1, x_2, \ldots, x_n|C_j) = \prod_{n}^{k=1} P(x_k|C_j),$$

dhe formula do të llogaritet në formën e mëposhtme, ku C_j është klasa me probabilitetin posterior më të lartë do të jetë etiketuar me X

$$P(C_j|X) = P(C_j) \prod_{k=1}^{n} P(x_k|C_j).$$

Duke e ditur këtë, ne mund të shohim se ka dy lloje parametrash që duhen gjetur gjatë trajnimit të të dhënave, probabiliteti i klasës dhe probabiliteti i kushtëzuar për secilën veçori të dhënë.

4.1.1 Trajnimi i të dhënave

Për të ndërtuar klasifikuesin, fillimisht duhet të nxjerrim çdo veçori nga të dhënat e përpunuara të trajnimit për të vlerësuar parametrat përkatës. Ky proces përcakton sa e mundshme është që një veçori e caktuar të shfaqet në një mesazh të një klase të caktuar në krahasim me veçoritë e tjera. Parametrat për veçoritë llogariten si më poshtë:

$$P(t|c) = \frac{T_{ct}+1}{\sum_{t' \in V} T_{ct'}+|V|},$$

ku T_{ct}është numri i ndeshjeve të veçorisë t në klasën C, dhe $\sum_{t \in V} T_{ct}$është numri i tokeneve që gjenden në mesazhet email të klasës C. Numrat 1 dhe$|V|$ në emërues dhe numërues, respektivisht, veprojnë si elementë rregullues për të shmangur çdo probabilitet që mund të jetë i barabartë me zero, ku $|V|$është numri total i veçorive ekzistuese (McCallum dhe Nigam, 1998).
Supozojmë se jemi duke punuar me vektorë të veçorive numerike, kështu që numri i veçorive mund të variojë nga 0 deri në n, ku n është një numër i plotë pozitiv. Le të supozojmë se kemi të dhëna të trajnimit nga tabela 4.1.

Klasa	Mesazhi
Spam	Buy tickets today
Spam	tickets tickets tickets
Spam	You won
Legitimate	Have you got the tickets
Legitimate	Where are you going

Tabela 4.1 Emaile që ndodhen në klasat e emaileve spam dhe legjitime

Me të dhënat nga tabela 4.1 që përdoren si të dhëna trajnimi, mund të llogarisim tashmë probabilitetin e klasës dhe probabilitetin e kushtëzuar të një veçorie.

Përdorim si shkurtime "*S*" për emailin e padëshiruar/spam dhe "*L*" për emailin e ligjshëm. Nga tabela shohim se kemi të bëjmë me tre mesazhe të tipit spam dhe dy mesazhe të tipit legjitim. Probabiliteti i klasës së emaileve të padëshiruar është $P(S) = \frac{3}{5}$ dhe probabiliteti i klasës së emaileve të ligjshëm është $P(L) = \frac{2}{5}$.

Si shembull, për veçorinë kemi zgjedhur tokenin “tickets”. Nga tabela 4.1 mund të shohim se fjala “tickets” gjendet katër herë në emailet më lart të shqyrtuara. Kemi $T_{ct} = 4$. Nga ana tjetër, shohim se numri total i tokeneve në emailet e padëshiruar është $\sum_{t' \in V} T_{ct'} = 8$, vlerë të cilën e gjejmë duke numëruar çdo fjalë në këto email-e spam. Për të gjetur |V| duhet të numërojmë numrin total të veçorive që po përdorim, i cili në këtë rast të marrë në studim do të ishte 11 (sepse tokenet që merren në shqyrtim janë: “buy”, “tickets”, “today”, “you”, “won”, “have”, “got”, “the”, “where”, “are”, “now”).

Si rrjedhim, me vlerat e gjetura kryejmë llogaritjet duke zëvendësuar te formula lart:

$$P(tickets|S) = \frac{4+1}{8+11} = \frac{5}{19}$$

Për $P(tickets|L)$ bëjmë llogaritje të ngjashme, prej nga marrim:

$$P(tickets|L) = \frac{1+1}{9+11} = \frac{2}{20}$$

Supozojmë se kemi emaili-n e mëposhtëm (tabela 4.2) dhe kërkojmë ta klasifikojme atë, pra të përcaktojmë në qoftë se ai është email spam apo jo.

Klasa	Mesazhi
Unknown	where are the tickets

Tabela 4.2 Shembulli i një emaili që do të klasifikohet

Probabiliteti	Vlera
$P(S) =$	$\frac{3}{5}$
$P(L) =$	$\frac{2}{5}$
$P(where\|S) =$	$\frac{1}{19}$
$P(are\|S) =$	$\frac{1}{19}$
$P(the\|S) =$	$\frac{1}{19}$
$P(tickets\|S) =$	$\frac{5}{19}$
$P(where\|L) =$	$\frac{2}{20}$

$P(are\|L) =$	$\frac{2}{20}$
$P(the\|L) =$	$\frac{2}{20}$
$P(tickets\|L) =$	$\frac{2}{20}$

Tabela 4.3 Llogaritjet e probabiliteteve me kusht për secilën veçori

Llogaritjet bëhen për të gjitha veçoritë në të gjithë bashkësinë e trajnimit. Rezultatet janë ato që përdoren më vonë në pjesën e klasifikimit për të gjetur probabilitetin që një mesazh të jetë spam apo i ligjshëm.

4.1.2 Klasifikimi

Bazohemi te të dhënat e tabelës 4.1, për të shpjeguar me anë të një shembulli se si do të bëhej klasifikimi i një mesazhi të ri, klasën e të cilit nuk e dimë. Siç u përmend më lart, marrim supozimin se duam të klasifikojmë mesazhin email të dhënë me anë të tabelës 4.2. Konsiderojmë vlerat e nevojshme të paraqitura te tabela 4.3.

Në këtë moment, i kemi të gjitha vlerat e nevojshme për të llogaritur probabilitetin që një dokument t'i përkasë klasës së emaileve spam dhe probabilitetin që ai t'i përkasë klasës së emaileve legjitimë. Zbatojmë formulën e Bajesit, prej nga do të kemi:

$$P(S|unknown) = P(S)P(where|S)P(are|S)P(the|S)P(tickets|S) =$$

$$= \frac{3}{5} \cdot \frac{1}{19} \cdot \frac{1}{19} \cdot \frac{1}{19} \cdot \frac{5}{19} = 2.3 \cdot 10^{-5}$$

$$P(L|unknown) = P(L)P(where|L)P(are|L)P(the|L)P(tickets|L) =$$

$$= \frac{2}{5} \cdot \frac{2}{20} \cdot \frac{2}{20} \cdot \frac{2}{20} \cdot \frac{2}{20} = 4 \cdot 10^{-5}$$

Nga llogaritjet shohim se$P(L|unknown)$ është më i madh se $P(S|unknown)$. Nga ky rezultat, themi se klasifikuesi ka caktuar që email i dhënë në tabelën 4.2 është një mesazh legjitim dhe jo spam.

4.2 Algoritmi i Decision Tree C4.5 Learning

Algoritmi i Decision Tree është një algoritëm që ka ardhur si një përmirësim i algoritmit të mëparshëm ID3 dhe përfshin disa përditësime, të tilla si aftësia për të trajtuar veçori me vlera të vazhdueshme (Quinlan, 1993). Ashtu si dhe me algoritmin e mëparshem NB dhe algoritmet e tjera pasues, fillimisht është e nevojshme të trajnojmë të dhënat. Bazuar në të dhënat e trajnuara, procesi i klasifikimit fillon me një nyje rrënjë dhe hap pas hapi ndahen nyjet sipas veçorisë më të rëndësishme, që zgjidhet me disa metoda opsionale përzgjedhjeje të veçorive (Cormack, 2006).

Në momentin që ndahet një nyje, krijohet një nyje vendimi, e cila përcakton se cilën nëndegë do të zgjedhim në çdo hap, për shembull për të vlerësuar nëse një email i ri është gati për t'u analizuar. Nyja e vendimit mban mend se cila veçori ndan të dhënat e trajnuara dhe vlera e veçorisë kërkohet për të gjitha degët. Ky proces kryhet në mënyrë rekursive derisa në fund të arrijmë në një nyje ku ndarja e mëtejshme ose nuk është e mundur ose nuk ul më tej entropinë e informacionit të të dhënave të trajnuara. Në këtë pikë, krijohet një nyje finale e cila etikohet sipas klasës dominante në të dhënat aktuale të trajnuara.

Pasi kemi përfunduar me krijimin e pemës së vendimeve, zakonisht kryejmë një proces të krasitjes për të zvogëluar madhësinë e pemës dhe për të përmirësuar performancën e saj në klasifikimin e të dhënave të reja. Për këtë, përdorim një test heuristik krasitjeje për reduktimin e gabimeve (Quinlan, 1993), i cili vlerëson gabimet e nyjës kundrejt degëve të saj për të vendosur nëse do të zëvendësohet apo jo me një nyje të varur.

4.2.1 Trajnimi i të dhënave

Për ndërtuar një peme vendimi me këtë algoritëm, përdorim një bashkësi të dhënash të trajnuara të përbërë nga vektorë veçorish të klasifikuar në kategori të ndryshme, të shënuar me $C = c_1, c_2, \ldots, c_m$, ku $c_1, c_2, \ldots, c_m$ përfaqësojnë klasat në të cilat do të ndahen të dhënat e trajnuara. Procesi i krijimit të pemës së vendimit fillon me një kulm rrënjë që përmban të gjitha të dhënat e trajnuara në bashkësinë T. Ndërtohet duke ndjekur disa hapa bazuar në metodën e Hunt (Quinlan, 1993), sipas rasteve të mëposhtme:

1. Nëse T përmban vetëm një lloj klase, atëherë pema përbëhet nga një kulm i vetëm dhe ai përcakton klasën e të dhënave.

2. Nëse T nuk përmban asnjë rast, atëherë pema përbëhet nga një kulm i vetëm dhe algoritmi C4.5 Learning përcakton klasën e këtij kulmi duke gjetur klasën maxhoritare të pemëve paraardhëse.

3. Nëse T përmban një përzierje klasash, atëherë përpiqemi ta ndajmë atë duke përdorur një veçori të vetme me qëllim që çdo nënbashkësi të ketë të dhëna që i përkasin një klase të vetme. Veçoria përbëhet nga një ose më shumë përfundime reciprokisht të papërshkueshme me rezultate $O_1, O_2, \ldots, O_n$ duke dhënë nënbashkësitë $T_1, T_2, \ldots, T_n$. Kulmi aktual i pemës do të bëhet një kulm vendimi bazuar në veçorinë e zgjedhur. Rezultatet do të krijojnë n degë dhe ato përpunohen hap pas hapi, ku brinja e i -të duke patur rezultatin O_i do të formojë nënpemën me të dhënat e trajnuara T_i.

4.2.2 Ndarja e të dhënave

Për të vendosur se si të ndajmë të dhënat e trajnuara duke përdorur veçorinë më të përshtatshme, ne testojmë fitimin e informacionit (IG) ose raportin e fitimit të informacionit (IGR) për secilën veçori të mundshme në vektorin e veçorive. Veçoria me raportin më të lartë të fitimit të informacionit zgjidhet për ndarjen e T, megjithëse kjo nuk garanton që çdo nënbashkësi e T të ketë një numër të mjaftueshëm rastesh. Numri minimal i rasteve në një nënbashkësi mund të ndryshojë, por vlera e paracaktuar në C4.5 është 2 (Guzella dhe Caminhas, 2009). Nëse ndonjë nga nënbashkësitë ka më pak raste se ky numër minimal, ndarja nuk mund të bëhet dhe krijohet një kulm që varet në vend të një kulmi vendimi, duke ndërprerë kështu rekursionin në atë degë. Algoritmi C4.5 përdor tre lloje të ndryshme ndarjesh që do t'i shqyrtojmë më poshtë.

1. Ndahet një veçori diskrete, e cila për çdo rezultat ka mundësi të prodhojë një degë

2. Ndahet në mënyrë të ngjashme me rastin e parë, por kur rezultate të ndryshme mund të grupohen së bashku. Në këtë rast, disa rezultate mund të ndajnë të njëjtën degë, në vend që të kenë vetem një degë për një rezultat.

3. Ndahet një veçori me vlera numerike të vazhdueshme. Kjo ndarje do të jetë binare; kështu ne kemi dy rezultate. Për të ndarë T mbi një tipar të vazhdueshëm f me një vlerë A për veçorinë e vazhdueshme kushtet duhen të jenë të tilla si $A \leq B$ ose $A > B$, ku B është një vlerë numerike që i takon një ndarjeje të mundshme.

Mënyra e ndarjes të së dhënave e përdorur në këtë studim është mënyra e tretë dhe raporti i fitimit të informacionit është përdorur si krahasues për ndarjen. Një shembull i si llogaritim raportin e fitimit të informacionit nga ndarja për një veçori, ku do të shënojmë numrin e përgjithshëm të rasteve me $|T|$, atëherë, T_j është një nënklasë e mundshme e marrë nga ndarja mbi bashkësinë T me anë të një nyje të vendimit dhe $freq(c_i, T_i)$ është frekuenca apo shpeshtësia e ndodhjes së një klase c_i në nënbashkësinë T_i. Si përfundim, me $proportion(c_i, T_i)$ do të shënojmë raportin $\frac{freq(c_i, T_i)}{|T|}$.

Raporti i asaj që quhet fitimi i informacionit, do të jetë krahasuesi i zgjedhur se për cilën veçori do të duhet të ndahen të dhënat e trajnuara. Fitimi i informacionit tregon një zhvendosje ndaj ndarjeve me shumë rezultate, ndërsa raporti i fitimit të informacionit e zgjidh këtë problem.

Për fitimim e informacionit IG përcaktojme barazimin:

$$IG(T, \alpha) = H(T) - H(T|\alpha).$$

Supozojmë se kemi dy veçori f_1 dhe f_2 që na interesojnë të rezultojnë në figurën vijuese 4.1 të dy pemëve

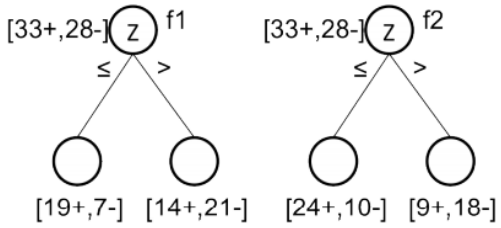


Figura 4.1 Dy ndarje të ndryshme të një nyjeje për veçoritë fe_1 dhe fe_2 kur përdorim të njëjtat të dhëna të trajnuara

Në figurën e mësipërme klasa pozitive është shënuar me c_1 dhe klasa negative me c_2. Qëllimi ynë është të gjejmë se cila nga këto ndarje krijon rezultatin më të mirë. Për të realizuar këtë, së pari duhet të llogaritim entropinë, sepse në qoftë se nuk do të kishte ndarje, që do të thotë se një kulm që varet do të krijohet atëherë në vend të saj. Ne mund të shohim nga figura 4.2 se mund të marrim që $|T| = 61$, $freq(c_1, T) = 33$ dhe $freq(c_2, T) = 28$. Tani mund të llogaritim:

$$H(T) = -proportion(c_1, T)\, log_2(\, proportion(c_1, T)) -$$

$$-proportion(c_2, T)\, log_2(\, proportion(c_2, T)) =$$

$$= -(\frac{33}{61})\, log_2(\frac{33}{61}) - (\frac{28}{61})\, log_2(\frac{28}{61}) \approx 0.995$$

Në momentin që entropia aktuale është e njohur, tani duhet të gjejmë entropinë për të zgjedhur një nga ndarjet. Në këtë shembull do zgjidhet veçoria f_1 duke përdorur ekuacionin $H(T|\alpha)$ ku α është veçoria që ne kemi zgjedhur. Kemi:

$$H(T|f_1) = \sum_{i=1}^{2} 2H(T_i) - proportion(c_1, T_i)\, log_2(\, proportion(c_1, T_i)) -$$

$$-proportion(c_2, T_i)\, log_2(\, proportion(c_2, T_i))$$

T_i janë dy bashkësi të dhënash të krijuara nga një ndarje mbi veçorinë f_1 duke dhënë kështu entropinë për bashkësinë 1

$$H(T_1) = -\frac{19}{26} log_2(\frac{19}{26}) - \frac{7}{26} log_2(\frac{7}{26}) \approx 0.84$$

dhe entropinë për bashkësinë 2

$$H(T_1) = -\frac{20}{35} log_2(\frac{20}{35}) - \frac{15}{35} log_2(\frac{15}{35}) \approx 0.98$$

Fitimi i informacionit IG jepet nga:

$$IGT(t, \alpha) = 0.24 - H(T|f_1) = 0.995 - \frac{26}{61} \cdot 0.84 - 35 \cdot 0.98 \approx 0.075$$

Shohim se nga llogaritjet, fitimi i informacionit mbi veçorinë f_1 është 0.075. Duke kryer të njëjtën metodë për veçorinë f_2, gjejmë vlerën 0.10. Për të gjetur raportin e fitimit të informacionit (IGR) tani mund të pjestojmë fitimin e informacionit me vlerën e brendshme të ndarjes së mundshme.

Duke përdorur formulën e njohur, kemi:

$$IV(T, \alpha) = -\sum_{v \in values(\alpha)} \frac{|\{x \in T | value(x, \alpha) = v|}{|T|} log_2(\frac{|\{x \in T | value(x, \alpha) = v|}{|T|})$$

Vlera e brendshme për ndarjen e parë merret ≈ 0.984 duke na dhënë një raport të fitimit të informacionit për veçorinë f_1 prej 0.076 dhe një raport të fitimit të informacionit për veçorinë f_2 rreth vlerës 0.101. Në këtë rast mund të shohim që nuk ka ndonjë diferencë reale nga fitimi i informacionit. Vemë re se si f_2 jep vlerën më të lartë të raportit të fitimit të informacionit, dhe kështu arrijmë në përfundimin se kjo është veçoria që do të zgjidhet. Më pas, këto të dhëna të trajnuara tashmë, do të ndahen në dy bashkësi dhe secila nga këto bashkësi do të vazhdojë të ndërtojë një nënpemë të re.

4.3.2 Krasitja

Kur ndërtimi i pemës së vendimit ka përfunduar, mund të jetë e dobishme të kryhet një proces krasitjeje për të shmangur mbivendosjen, duke përdorur një metodë të quajtur krasitje me reduktimin e gabimit. Mbivendosja ndodh kur pema nuk ka një normë të lartë gabimi në klasifikimin e të dhënave të testimit. Kjo mund të ndodhë kur pema është shumë komplekse.

Ndërtimi i pemës së vendimit fillon me një kulm rrënjë. C4.5 ofron dy metoda krasitjeje: zëvendësimi i nënpemës dhe ngritja e nënpemës. Zbatimi i këtyre metodave është opsional pas përfundimit të procesit të ndërtimit.

Procesi i krasitjes kryhet nga e majta në të djathtë dhe nga lart poshtë. Kulmet e vendimit afër gjetheve krahasohen dhe, në mënyrë rekursive, procesi zbret deri në rrënjë. Qëllimi është të gjejmë kulmet e vendimit ose nënpemët me një shkallë më të ulët gabimi në klasifikim. Nëse gjendet një kulm i tillë, atëherë zëvendësojmë kulmin aktual me një kulm që varet ose me një nënpemë më të përshtatshme. Kjo duhet të ulë normën mesatare të gabimit në të dhënat e trajnuara dhe të përmirësojë përgjithësimin e pemës.

Vlerësimi i gabimit bëhet duke llogaritur intervalin e sipërm të besimit $U_{cf}(E,N)$ për shpërndarjen binomiale të një kulmi që varet, për një nivel të caktuar besimi (kemi zgjedhur një interval besimi deri në 25%). Në këtë kontekst, E përfaqëson numrin e shembujve të trajnuar që janë klasifikuar gabim dhe N është numri total i shembujve të trajnuar. Vlerësimi i gabimit llogaritet $N \cdot U_{cf}(E,N)$ duke shumëzuar intervalin e sipërm të besimit me numrin e përgjithshëm të rasteve në kulme, duke marrë parasysh numrin total të shembujve dhe numrin e gabimeve (shuma e klasave në minorancë).

Zëvendësimi i nënpemës ndodh kur një kulm vendimi ka një normë gabimi në klasifikim teorikisht më të ulët se shuma e peshuar e gabimit të degëve të tij. Nëse kjo është e vërtetë, kulmi i vendimit zëvendësohet me një kulm që varet dhe krijohet një shpërndarje e klasës që jep probabilitetin e secilës klasë për kulmin që varet.

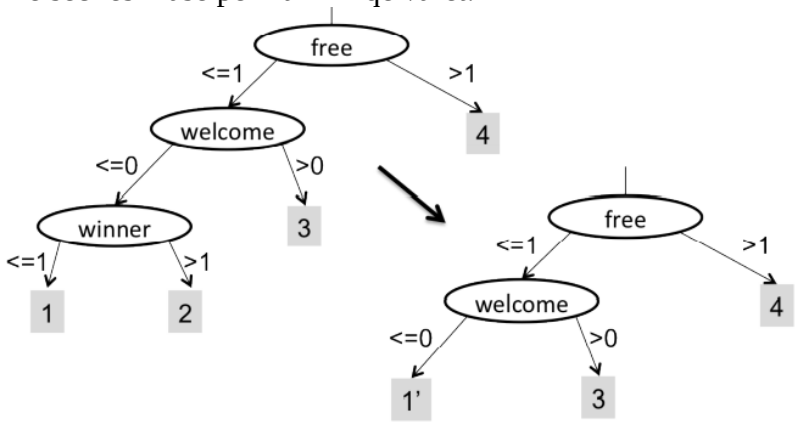


Figura 4.2 Shembull zëvendësimi i një nënpeme

Në këtë shembull, është konstatuar se kulmi i vendimit me veçorinë "winner" ka një gabim të vlerësuar më të ulët se degët e tij. Për këtë arsye, bëhet një zëvendësim i nënpemës: kulmi i vendimit shndërrohet në një kulm gjethe dhe shpërndarja probabilitare nga degët e mëparshme formon kulmin e ri. "1" në figurë tregon shpërndarjen e re probabilitare për kulmin që varet.

Nga ana tjetër, nënpema e formuar do të krahasohet me një kulm vendimi që ka vlerësimin më të madh të gabimit në degët e tij dhe me vlerësimin e gabimit të pemës duke filluar nga kulmi. Nëse pema që fillon nga dega me gabimin më të madh ka një vlerësim më të ulët të gabimit, atëherë kulmi rrënjë i asaj peme zëvendëson kulmin paraardhës, dhe të dhënat e trajnuara për degën më të vogël do të rishpërndahen tek kulmi i degës më të madhe. Efektet e nënpemëve të ngritura mund të jenë të paqarta; në disa raste, ato mund të përmirësojnë rezultatin e klasifikuesit (Witten dhe Frank, 2005).

4.3.3 Klasifikimi

Klasifikimi fillon nga nyja rrënjë dhe zbret një nga një degët në kulmin aktual të vendimit derisa të arrihet tek një kulm gjethe. Zgjedhja e degës në një kulm vendimi bazohet në veçorinë e specifikuar nga kulmi dhe pragu për atë veçori. Nëse veçoria në mesazhin që po klasifikohet ka një numër vlerash të barabartë ose më të ulët se pragu i veçorisë së zgjedhur nga kulmi i vendimit, zgjidhet dega e majtë; në të kundërt, zgjidhet dega e djathtë.

Si shembull, supozojmë se po klasifikojmë një mesazh me vektorin e veçorive të paraqitura në tabelën 4.4.

Veçoritë	Vlera
free	1
thanks	0
charge	1
welcome	0
winner	2

Tabela 4.4 Vektor i veçorive për DT

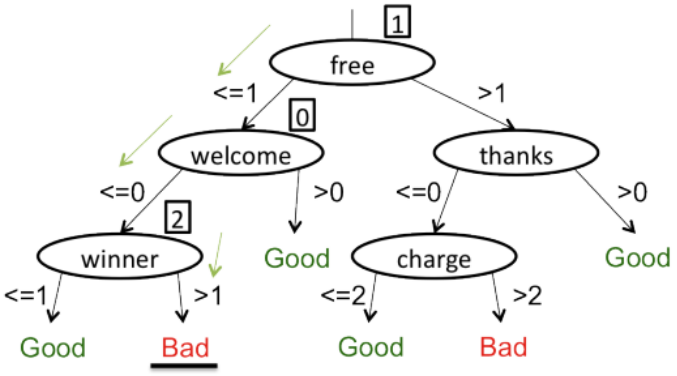


Figura 4.3 Klasifikimi për një pemë vendimi

Në këtë shembull, kemi të bëjmë vetëm me numra natyrorë, pasi në kontekstin e filtrimit të emaileve të padëshiruara, bëhet fjalë vetëm për numërimin e fjalëve. Kulmi rrënjë në këtë pemë vendimi merret me veçorinë "free" dhe pragu për të është më i vogël ose i barabartë me 1 për degën e majtë dhe më i madh se 1 për degën e djathtë. Vektori i tipareve të shembullit ka vlerën 1 për këtë veçori, që do të thotë se merret dega e majtë. Në kulmin tjetër, vektori i veçorive përsëri merr degën e majtë. Në kulmin e fundit të vendimit për këtë degë, vlera për veçorinë "winner" tejkalon pragun, që do të thotë se merret dega e djathtë. Më në fund, arrihet një kulm gjethe dhe jepet një vendim. Në shembull, një rezultat i mirë nënkupton që është një

mesazh i ligjshëm dhe një rezultat i keq nënkupton që është një mesazh i padëshiruar. Në këtë rast, mesazhi do të klasifikohet si i padëshiruar dhe do të bllokohet.

Kur në pemë arrihet një kulm gjethe, kjo tregon klasën me etiketën e klasës dhe probabilitetin e saj. Probabiliteti llogaritet si raporti $\frac{K}{N}$, ku K është numri i shembujve të trajnuar nga klasa C në këtë kulm dhe N është numri i përgjithshëm i shembujve të trajnuar që kanë arritur në këtë kulm.

4.4 Algoritmi i Support Vector Machine (SVM)

Makina Mbështetëse Vektoriale (SVM) konsiderohet sot si një nga algoritmet më efektive të machine learning. Thelbi i këtij klasifikuesi është trajtimi i çdo vektori veçorish si një pikë në një hapësirë me shumë dimensione, ku numri i dimensioneve kontrollohet nga një funksion kernel. Në klasifikimin e tekstit, shpesh përdoret një kernel linear për shkak të natyrës së të dhënave, duke krijuar një hapësirë n-dimensionale, ku n është numri i veçorive. Synimi është të gjendet një hiperplan në këtë hapësirë që mund të ndajë vektorët në dy klasa të dallueshme, si p.sh. mesazhet spam dhe ato legjitime. Hiperplani duhet të ndajë pikat në mënyrë të tillë që distanca midis tij dhe pikave më të afërta të jetë sa më e madhe që të jetë e mundur, duke siguruar kështu një ndarje të qartë të klasave.

4.4.1 Trajnimi i të dhënave për SVM

Supozojmë se kemi dy klasa, ku $y \in \{1, -1\}$ është një klasë e etiketuar, $x \in \mathbb{R}^n$ një vektor i ndërtuar nga të dhënat e trajnuara, ku n është dimensioni i vektorit të veçorive. Kemi një bashkësi të trajnuar të çifteve të renditura $(x_1, y_1),\ldots,(x_n, y_n)$, ku n është numri i rasteve që janë trajnuar.

Në këtë moment duhet të kërkojmë një hiperplan $w \cdot x - b = 0$, i cili do të ketë distancën gjeometrike më të madhe të mundshme nga pikat më të afërta dhe ku kushtet për çdo rast janë të tilla si $y_i(w \cdot x_i - b) \geq 1$, për çdo i. Këtu w është një vektor normal i planit dhe distanca gjeometrike është e barabartë me dyfishin e madhësisë së distancës me pikën më të afërt nga hiperplani.

Kërkojmë këtë hiperplan të trajtës $w \cdot x - b = 0$, i cili ndan plotësisht të dy klasat, ku hiperplani është sa më larg të jetë e mundur nga secila prej pikave më të afërta për secilën klasë. Për ta gjetur hiperplanin supozojme se kemi dy kushte $y_i(w \cdot x_i - b) = 1$ dhe $y_i(w \cdot x_i - b) = -1$, siç shihet edhe në figurën 4.4, që të dy i ndajnë të dhënat pa u mbivendosur. Distanca midis këtyre dy planeve është përcaktuar të jetë $\frac{2}{\|w\|}$.

Duhet gjetur një plan që i ndan të dy klasat në mënyrë që diferenca më e mirë midis këtyre dy planeve do të jetë sa më madhe e mundshme, pra që do të thotë se w duhet të jetë sa më e vogël. Shtohet kushti $w \cdot x - b \geq 1$ për $y_i = +1$ dhe $w \cdot x - b \leq 1$ për $y_i = -1$ shtohet për të ndaluar çdo pikë të të dhënave që të shfaqet në kufij. Kufizimi i parë do të zbatohej në klasën që është ndarë nga hiperplani$y_i(w \cdot x_i - b) = 1$ dhe kushti i dytë për hiperplanin $y_i(w \cdot x_i - b) = -1$, të cilën e shohim në figurën 4.4. Mund të shkuajmë formën: $y_i(w \cdot x_i - b) \geq 1$, për çdo $1 \leq i \leq n$.

Tani mund të formulojmë këtë problem si një problem të optimizimit të kushtëzuar me funksion të qëllimit $\underset{w,b}{Minimize}\|w\|$, kur plotësohen kushtet $y_i(w \cdot x_i - b) \geq 1$, për çdo $1 \leq i \leq n$, të përmendura më sipër (Burges, 1998).

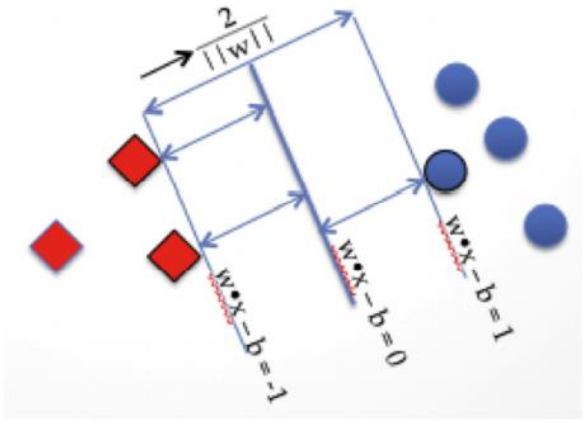


Figura 4.4. Dy klasa të ndryshme me SVM

Gjatë trajnimit të të dhënave, vetëm pikat që ndodhen pikërisht në kufirin e ndarjes kanë ndikim në hiperplan dhe këto quhen vektorë mbështetës. Nga figura 4.4 mund të vërejmë se, me zvogëlimin e *w*, rritet distanca, kështu që për të gjetur hiperplanin optimal që ndan klasat, duhet të minimizojmë *w*. Norma e *w* përfshin një rrënjë katrore, por ajo mund të hiqet për të thjeshtuar llogaritjet dhe funskioni merr trajtën $\underset{w,b}{Minimize}\frac{1}{2}w^2$.

Ndonjëherë, nuk është e mundur të gjendet një hiperplan që ndan plotësisht dy klasat. Në raste të tilla, duhet të konsiderojmë rastet ku nuk mund të krijojmë një hiperplan ndarës dhe minimizimin e distancës. Një zgjidhje për këtë është metoda e Soft Margin, e cila minimizon distancën. Duke përdorur këtë metodë, kufizimi i mëparshëm modifikohet në formën:

$$y_i(w \cdot x_i - b) \geq 1 - v_i$$

ku v_i është gabimi që tregon sa larg është pika nga marzhi. Funksioni fillestar ndryshon gjithashtu në formën:

$$\underset{w,v,b}{Minimize}(\frac{1}{2}w^2 + C\sum_{i=1}^{n} w_i)$$

duke marrë parasysh gabimin (Fletcher, 2008).

4.4.2 Klasifikimi për SVM

Klasifikimi duke përdorur algoritmin e Makinës Vektoriale Mbështetëse (SVM) është i thjeshtë për t'u zbatuar. Pasi të jetë përcaktuar hiperplani dhe mesazhi hyrës të jetë shndërruar në një vektor veçorish, llogaritet diferenca midis hiperplanit dhe vektorit të veçorive $w \cdot x - b = 0$. Vektori i veçorive përfaqësohet nga *x* , dhe rezultati përcakton etiketimin e email-it hyrës.

Nëse vlera e rezultatit do të jetë më e vogël se zero, mesazhi etiketohet si një klasë e caktuar dhe nëse është më e madhe se zero, etiketohet si klasa tjetër. Pragu zero mund të rregullohet, duke lejuar klasifikimin të bëhet më i saktë në mënyrë për të zvogëluar shkallën e rezultateve false pozitive ose për të rritur shkallën e rezultateve të sakta pozitive.

KREU V

PLATFORMA WEKA

Ky kapitull do të ketë në fokus përdorimin e platformës WEKA për trajnimin e algoritmeve të sqaruar me detaje më lart si Naïve Bajes (NB), Pemës së Vendimit (DT) dhe Makinës së Mbështetjes Vektoriale (SVM).

5.1 Prezantim mbi softuerin WEKA

Waikato Environment for Knoëledge Analysis (WEKA) është një koleksion softuerësh që shërben për të trajnuar makinat dhe për analizën e të dhënave të licencuar sipas licencës së përgjithshme publike GNU. Ai u zhvillua në Universitetin e Waikatos, Zelandën e Re dhe është softueri shoqërues i librit "Data Mining: Practical Machine Learning Tools and Techniques". WEKA ofron shumë mundësi testimi të disa prej algoritmeve të Machine Learning.

Figura 5.1 Softueri WEKA

WEKA mbështet parapërpunimin e të dhënave, grupimin, klasifikimin, regresionin, vizualizimin dhe përzgjedhjen e veçorive të ndryshme. Hyrja në WEKA pritet të formatohet sipas formatit të skedarit atribut-relacional dhe me emrin e skedarit që përfundon me .arff.
ss
WEKA siguron akses në bazat e të dhënave SQL duke përdorur lidhjen e bazës së të dhënave Java dhe mund të përpunojë rezultatin e kthyer nga një query e bazës së të dhënave. WEKA ofron akses në Deep Learning me anë të Deeplearning4j. WEKA është një program i shkruar në gjuhën e programimit Java, i cili përmban mjete për një mjedis testimi dhe është në të njëjtën kohë i pajisur me shumë algoritme të machine learning. Kjo platformë gjithashtu lejon paraqitjen e rezultateve nga vlerësimet e algoritmeve përmes grafikëve, tekstit si dhe mjeteve të tjera, si për shembull ndërtimet e Decision Tree.

Explorer është ndërfaqja kryesore për softuerin WEKA ku ndodhen një listë panelesh, dhe secila prej tyre mund të përdoret për të kryer një detyrë specifike. Në qoftë se kemi ngarkuar një grup të dhënash, një panel tjetër i kërkuar në Explorer mund të përdoret për të bërë analiza të mëtejshme.
Më poshtë paraqitet një figurë e cila ilustron këtë ndërfaqe.

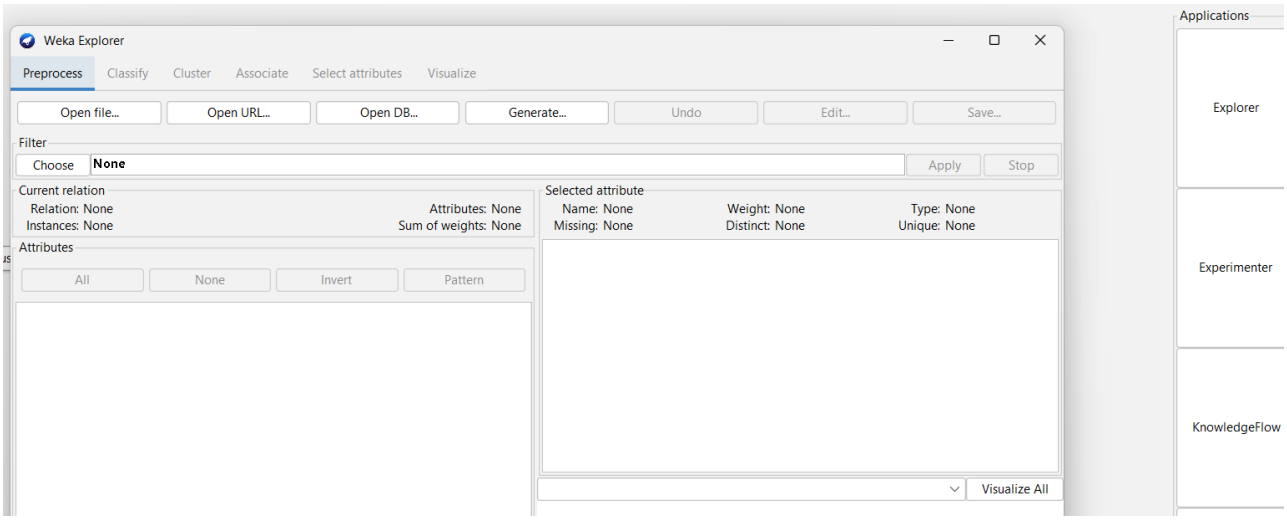


Figura 5.2 Ndërfaqja Explorer e softuerit WEKA

WEKA përdoret për të lexuar dhe interpretuar të dhënat e ruajtura të komunikimeve email nga një bazë të dhënash, dhe për të krijuar një paraqitje të strukturuar nga ky informacion, i cili është i kuptueshem për algoritmet e Machine Learning. Përdoret gjithashtu për të vlerësuar performancën e algoritmit duke analizuar kohën e nevojshme për përfundimin e procesit, si dhe saktësinë e çdo filtri. Këto rezultate mund të regjistrohen për të mbajtur një histori të procesimit dhe për të vizualizuar grafikët e quajtur si vija ROC, që paraqesin saktësinë e filtrave të përdorur. WEKA gjithashtu përdoret për të vlerësuar efikasitetin e procesit duke matur kohën e nevojshme për të përfunduar një detyrë dhe efikasitetin e çdo filtri. Këto rezultate ruhen më pas në kohëra të caktuara midis fillimit dhe përfundimit të ekzekutimit të detyrës së përcaktuar nga përdoruesi, dhe me anë të krijimit të vijave ROC paraqitet efikasiteti i filtrimit të çdo vlerësimi.

Figura 5.3 Karakteristikat e platformës WEKA

5.2 Vlerësimi i kryqëzuar i WEKA

Vlerësimi i kryqëzuar (Cross validation) është një metodë statistikore e përdorur për të vlerësuar aftësinë e modeleve të Machine Learning për të krahasuar dhe zgjedhur një model për një problem të caktuar.
Vlerësimi i kryqëzuar qëndron mbi bazën e marrjes në studim të një grupi trajnimi dhe krijimin e një klasifikuesi. Vlerësimi i kryqëzuar i *K*-fishtë i ndan të dhënat në K nënzgjedhje (apo palosje /folds) ku K është sasia e zgjedhjeve të marra nga përdoruesi. *K-1* nënzgjedhjet gjithmonë përdoren për trajnimin e të dhënave, kurse një nënzgjedhje përdoret për testim, apo vlerësimin e filtrit të përdorur. Për shembull, në një vlerësim të kryqëzuar të 10-fishtë, ndajmë grupin e të dhënave në 10 pjesë (apo palosje) mbajmë secilën pjesë me radhë dhe gjejmë mesataren e rezultateve. Pra, çdo nënzgjedhje në grupin e të dhënave përdoret një herë për testim dhe 9 herë për trajnim.
Pasi llogaritet mesatarja, mund të përftojmë vijat ROC si një vlerësim i performancës së filtrave. Pas kësaj llogaritjeje mund të përftojmë gjithashtu dhe mesataren e shpejtësisë së klasifikimit apo trajnimit.

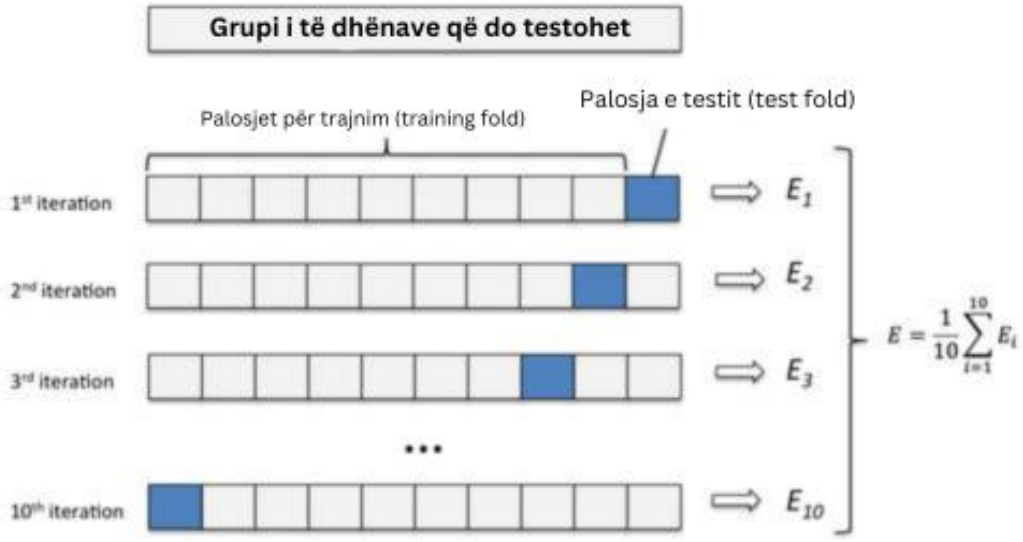


Figura 5.4 Shembull i vlerësimit të kryqëzuar të 10-fishtë

5.3 Vijat ROC (Receiver Operating Characteristic)

Një vijë ROC (Receiver Operating Characteristic) është një ndërtim grafik që përdoret zakonisht për të treguar se sa e saktë është shkalla e klasifikimit të një klasifikuesi, pra performancën e tij. Boshtet korrespondojnë respektivisht me normat e vërteta pozitive (*TPR- true positive rate*) dhe normat pozitive false (*FPR- false positive rate*). Boshti i x - eve bazohet në *FPR*, i cili mund të përfaqësojë për shembull mesazhe legjitime të klasifikuara si mesazhe të padëshiruara. Boshti i y - eve për *tpr*, i cili përfaqëson për shembull emailet e padëshiruar të klasifikuar saktësisht si emaile të padëshiruar. Grafiku është ndërtuar duke pasur rastet e klasifikuara të të dhënave të testit të shënuara dhe duke ulur në mënyrë të njëpasnjëshme një vlerë të pragut të klasifikimit për të llogaritur pikat e reja në grafik, bazuar në vlerat aktuale të *fpr* dhe *tpr*. Një rezultat i lartë tregon se ka shumë të ngjarë që të dhënat e trajnuara në këtë rast mund të jenë emaile të padëshiruara, ndërsa një rezultat i ulët na tregon se ka më shumë të

ngjarë që të dhënat e trajnuara të jenë emaile të lejuara. Vlera e pragut vendos në qoftë se një rast do të shënohet si një mesazh i padëshiruar ose si një mesazh legjitim, në varësi të faktit në qoftë se rezultati i klasifikimit është më i lartë ose i barabartë me pragun, ose në qoftë se nuk është i tillë.

Për të përdorur siç duhet vijat ROC për krahasimin e saktësisë së klasifikuesve të ndryshëm, vlera e variancës së vlerësimit duhet të merret parasysh. Në qoftë se ne përdorim vlerësimin e kryqëzuar të K - fishtë, ne marrim K rezultate të performancës së testit, një për çdo nënzgjedhje. Për shkak të këtyre disa rasteve të testimit, ne mund të nxjerrim një variacion kur do të gjenerojmë vijën e përfunduar ROC.

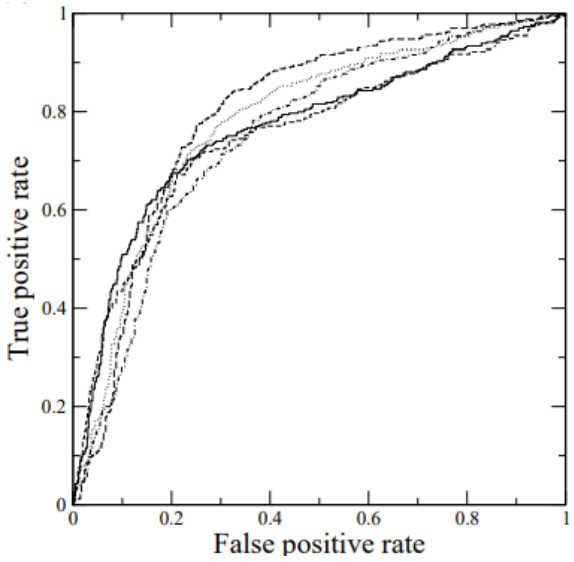


Figura 5.5 Vija të shumëfishta të vizatuara nga të dhënat nga nënzgjedhje të pavarura. Figura është marrë nga (Fawcett, 2006)

Por thjesht unifikimi i vijave rezultuese të krijuara nga rastet e provës nga një vlerësim i kryqëzuar për të ndërtuar grafikun e kurbës së përfunduar ROC, heq mundësinë për të marrë parasysh variancën, që është një nga arsyet për të pasur edhe disa nënzgjedhje testimi.

Për ta bërë këtë, ne kemi nevojë për një "metodë që kampionon vijat individuale në pika të ndryshme dhe mesatarizon zgjedhjet" (Fawcett, 2006).

Një metodë për ta bërë këtë quhet Mesatarizimi Vertikal, e cila është e përshtatshme për t'u përdorur, kur ne mund të fiksojmë vlerën e *fpr*. Në këtë rast ne mund ta kontrollojmë atë deri në një masë të caktuar. Ai ka vlera të fiksuara *fpr* dhe, siç nënkupton emri mesatarisht, *tpr* të secilës vijë nga nënzgjedhja e testit për çdo vlerë të dhënë *fpr*. Vlerat më të larta të *tpr* nxirren nga çdo nënzgjedhje për çdo vlerë të *fpr* dhe në qoftë se një vlerë korresponduese e *fpr* nuk ekziston në vijë, vlera e *tpr* do të interpolohet midis pikave ekzistuese. Vlerat e *tpr* për *fpr* e dhënë mesatarizohen dhe ruhen. Dhe kështu vlera e *fpr* rritet me një shumë të caktuar dhe e njëjta procedurë përsëritet me radhë. Pra, në thelb për çdo vlerë të dhënë të *fpr* në grafik, vlera e *tpr* jepet nga funksioni $R(frate) = mean[R_i(frate)]$, ku R_i është çdo vijë ROC e krijuar nga nënzgjedhjet e testit.

Duke patur vijën mesatare ROC dhe vijat ROC të gjeneruara nga çdo rast testimi, tani mund të gjejmë variancën për *tpr* për çdo vlerë të *fpr* të dhënë dhe rezultati mund të jetë diçka si në figurën 4.2.

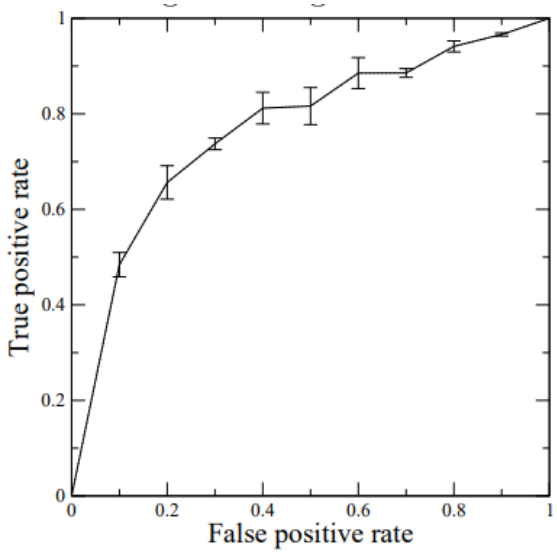


Figura 5.6 Vija ROC e vizatuar me intervale të llogaritura nga mesatarja vertikale bazuar në të dhënat nga rizgjedhjet e shumëfishta. (Fawcett, 2006)

Në një kurbë ROC, njëra nga matjet e përdorura për të krahasuar performancën e përgjithshme do të ishte ajo që bëhej duke krahasuar syprinën nën vijën e grafikut (Area Under Curve), dhe sa më e madhe të jetë zona, aq më mirë do të ishte kjo për rezultatin e përftuar nga zbatimi i algoritmit. Por shumë herë si në këtë studim mund të jetë me interes të studiohet vetëm një zonë specifike e kurbës. Në këtë studim është e rëndësishme që vlera e *fpr* të jetë në intervalin nga zero deri në gjysmë për qind që të jetë afër normës 0.1% të pranueshme për mesazhet legjitime të filtrimit. Pra, zona e interesit është në fillim të vijave ROC.

5.4 Dataset i përdorur dhe metoda e ndjekur

Grupi i të dhënave yë përdorura për këtë punim janë marrë nga SPAMBASE i cili është marrë nga UCI machine learning repository dhe është krijuar nga Mark Hopkins, Erik Reeber, George Forman dhe Jaap Suermondt. Ky dataset përmban 4601 mesazhe emaili dhe 58 atribute. Ky koleksion i grupit të të dhënave të email-eve jo-spam është marrë nga email-e personale dhe llogari email-esh biznesi. Ky grup të dhënash përfshin një përzgjedhje e mesazheve email, të përshtatshme për t'u përdorur në testimin e sistemeve të filtrimit të spamit. Çdo shembull në SPAMBASE përbëhet nga 58 atribute. Shumica e atributeve përfaqësojnë frekuencën e një fjale ose karakteri të caktuar në email që korrespondon me instancën. Për ta përdorur këtë dataset në platformën WEKA, duhet të përdorim konvertimin në formatin arff.

Pasi shkarkojmë spambase.arff, zgjedhim tab-in *Preprocess* në platformën Weka, dhe pasi klikojmë *Open file...* zgjedhim spambase.arff file që kemi shkarkuar më parë. Para se të përdorim dataset-in, duhet të fshijmë veçoritë `capital_run_length_average, capital_run_length_longest` dhe `capital_run_length_total`. Veçoritë e mbetura përfaqësojnë frekuencat relative të fjalëve dhe karaktereve të ndryshme të rëndësishme në email.

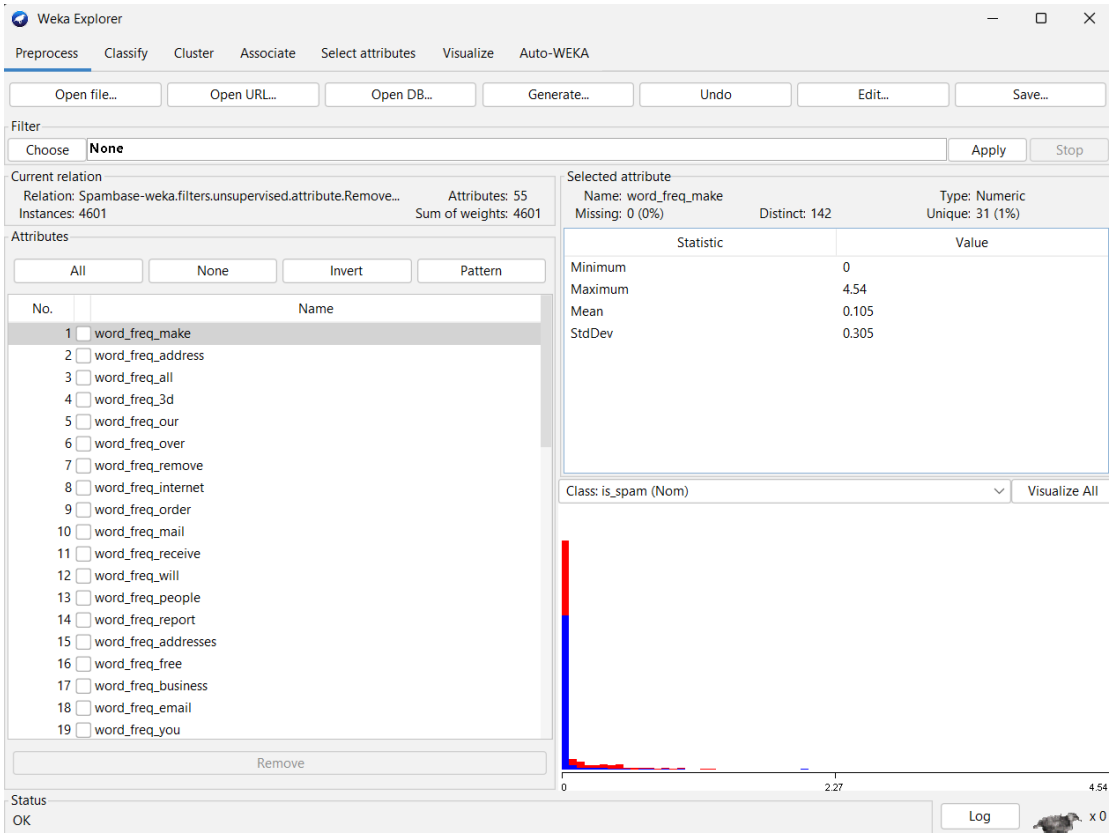


Figura 5.7 Ndërfaqja në platformën WEKA para konvertimit në vlera boolean

Qëllimi ynë është që t'i konvertojmë këto në vlera booleane: 1 në qoftë se fjala ose karakteri është i pranishëm në email dhe 0 në qoftë se fjala ose karakteri nuk është i pranishëm në email. Për të bërë këtë konvertim ne përdorim filtrin Numeric To Binary duke ndjekur rrugën *filters* > *unsupervised* > *attribute* > *NumericToBinary*. Të gjitha atributet numerike të frekuencës në këtë moment janë konvertuar në vlera boolean. Çdo e-mail përfaqësohet nga një vektor 55 dimensional që përfaqëson nëse ekziston apo jo një fjalë e veçantë në një e-mail. [www.inf.ed.ac.uk/teaching/courses/iaml/lab/lab1.html]

Në figurën 5.8 kemi paraqitur listën e veçorive të përcaktuara që në fillim të studimit, ndërsa në figurën 5.9 është shfaqur një shembull i ndeshjeve të veçorisë "word_freq_direct_binarized". Kemi zgjedhur vlerësimin e kryqëzuar të K-fishtë, ku vlera e K është vendosur të jetë 10. Pas kësaj, kemi ekzekutuar tre algoritmet e machine learning të përmendura më parë: algoritmin Naïve Bayes, algoritmin Decision Tree dhe algoritmin Support Vector Machine. Në seksionin e mëposhtëm kemi paraqitur rezultatet e vlerësimeve dhe vijat ROC për secilin nga tre algoritmet.

Ndër parametrat e vlerësuar të modeleve përmendim: rastet e klasifikuara saktë (Correctly Classified Instances), rastet e klasifikuara gabimisht (Incorrectly Classified Instances), statistikat Kappa (Kappa Statistics), gabimi mesatar absolut (Mean Absolute Error), gabimi mesatar katror (Root Mean Squared Error), gabimi absolut relativ (Relative Absolute Error), gabimi relativ në rrënjë katrore (Root Relative Squared Error), numri total i rasteve (Total Number of Instances), shkalla TP (TP Rate), shkalla FP (FP Rate), saktësia (Precision), rikujtimi (Recall), masa F (F-Measure), MCC, zona ROC (ROC Area), dhe zona PRC dhe klasa (PRC Area and Class).

Testet u kryen me një laptop me parametra 12th Gen Intel(R) Core(TM) i5-1235U 1.30 dhe RAM 8.00 GB (7.68 GB usable)

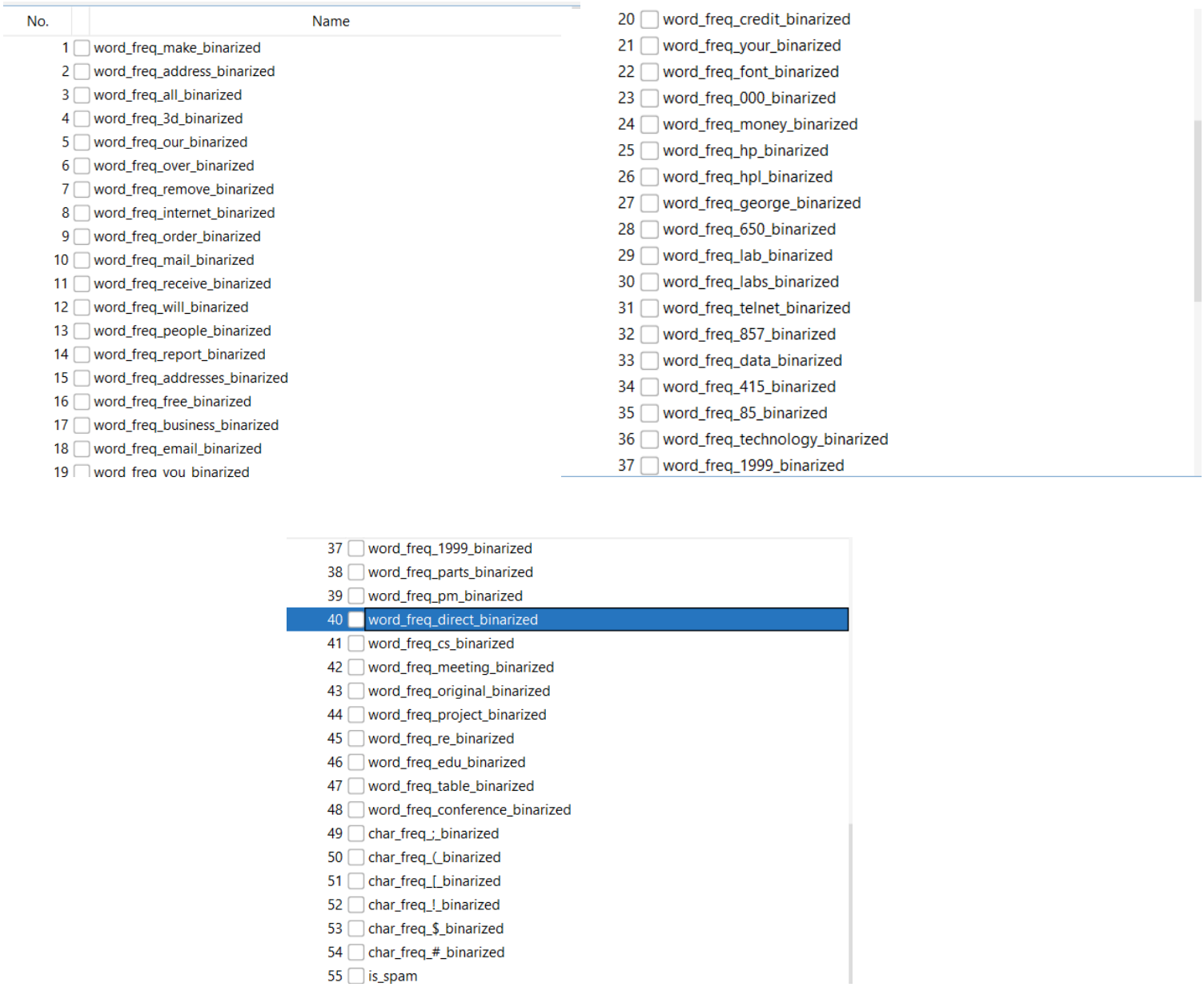
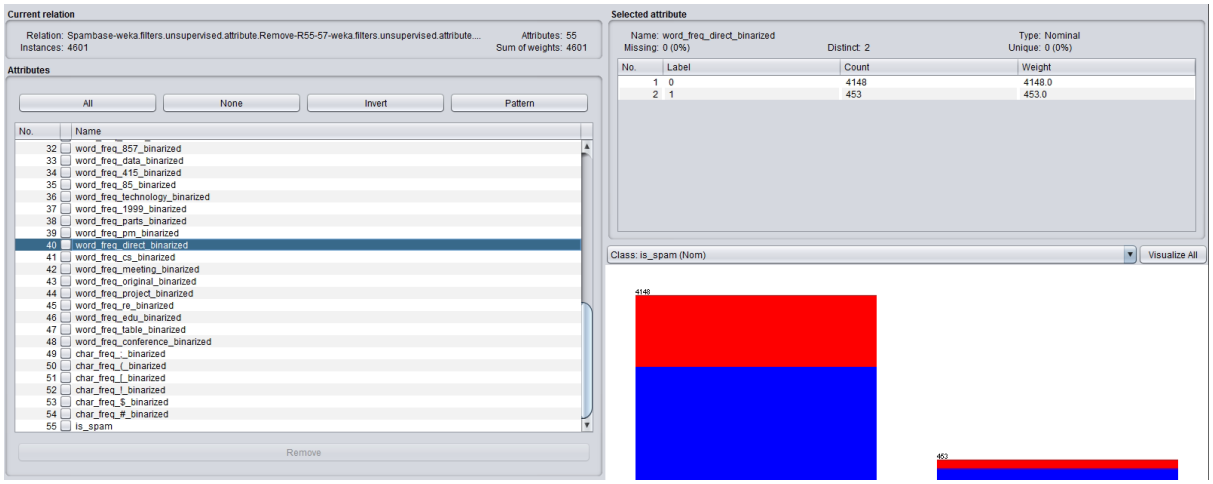


Figura 5.8 Lista e veçorive të përcaktuara për përcaktimin e emaileve spam dhe email-eve jospam



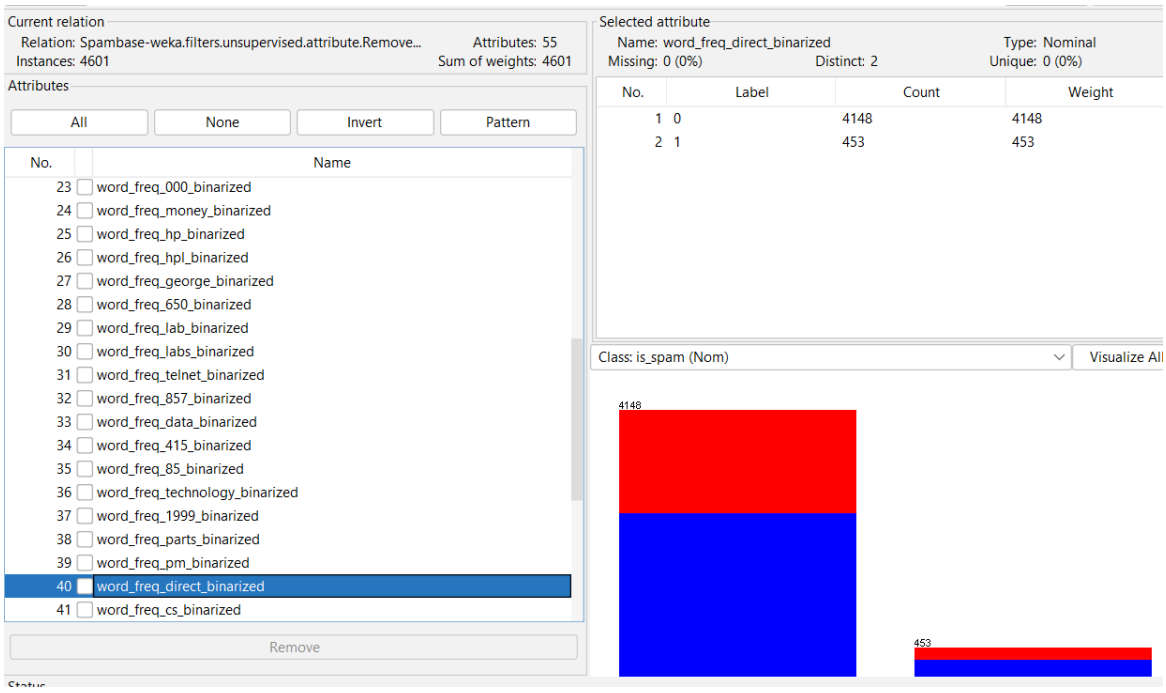


Figura 5.9 Shembull për frekuencën e ndeshjeve të veçorisë “word_freq_direct_binarized”

5.5 Vlerësimi i algoritmit të Naïve Bajes

Siç është përmendur më lart, Naïve Bayes është një nga filtrat më të thjeshtë për vlerësim. Duke pasur parasysh grupin e të dhënave, duhet të përdorim një klasifikues Naïve Bayes për të trajnuar të dhënat, në mënyrë që të dallojmë emailet e padëshiruara nga ato legjitime duke përdorur një shpërndarje të përshtatshme të frekuencës së secilës fjalë në të gjitha emailet e padëshiruara dhe ato legjitime.

Në tab-in Classify në Weka, zgjedhim algoritmin Naïve Bayes.

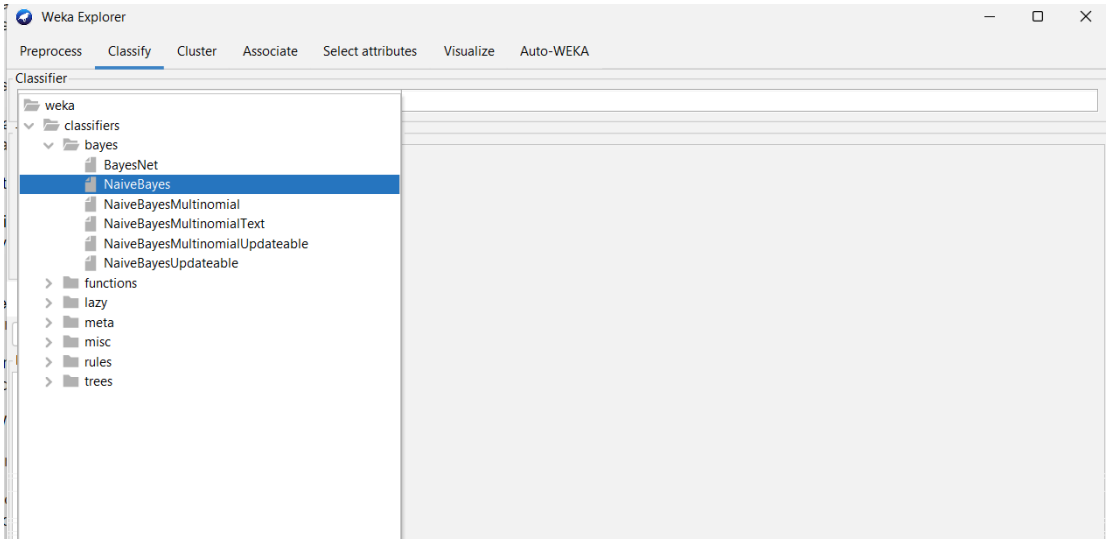


Figura 5.10 Ekzekutimi i algoritmit të Naïve Bajes në platformën WEKA

Shohim rezultatet e marra nga algoritmi:

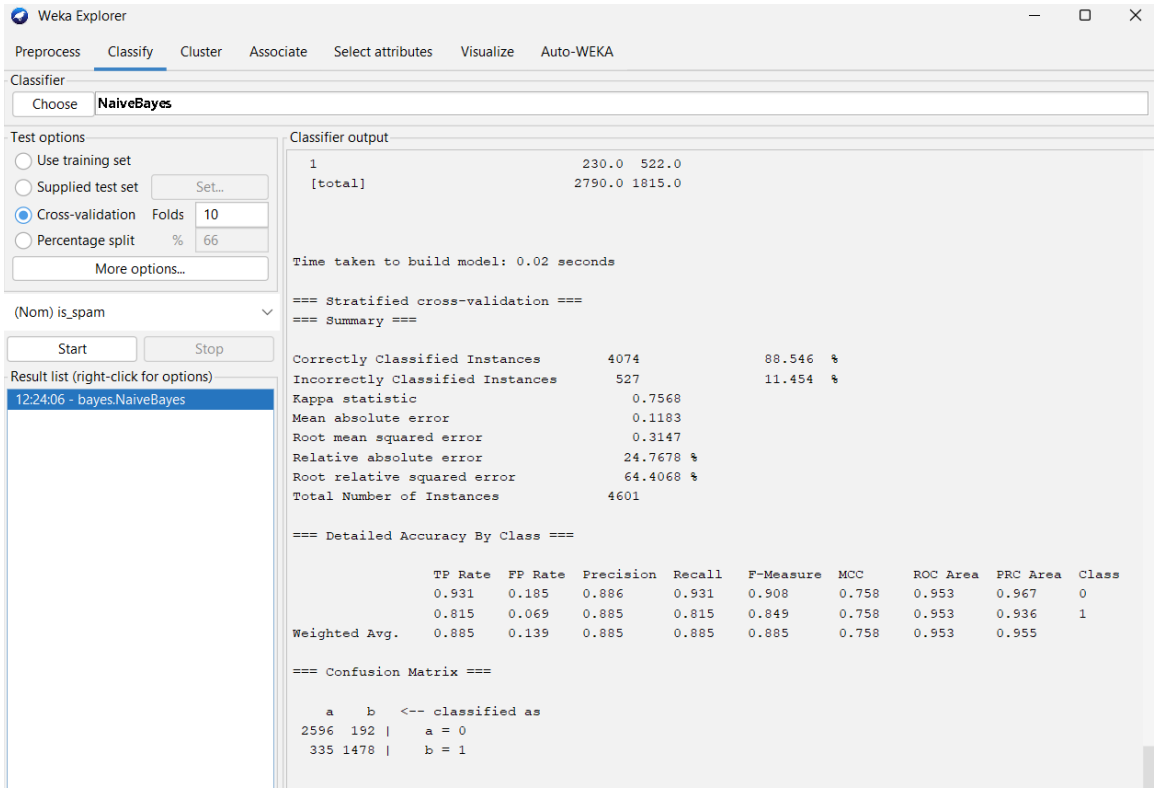


Figura 5.11 Rezultatet e vlerësimit të algoritmit Naïve Bajes në platformën WEKA

Për të paraqitur vijat ROC për këtë klasifikues, zgjedhim opsionin “Weka Classifyer Visualize: ThresholdCurve”

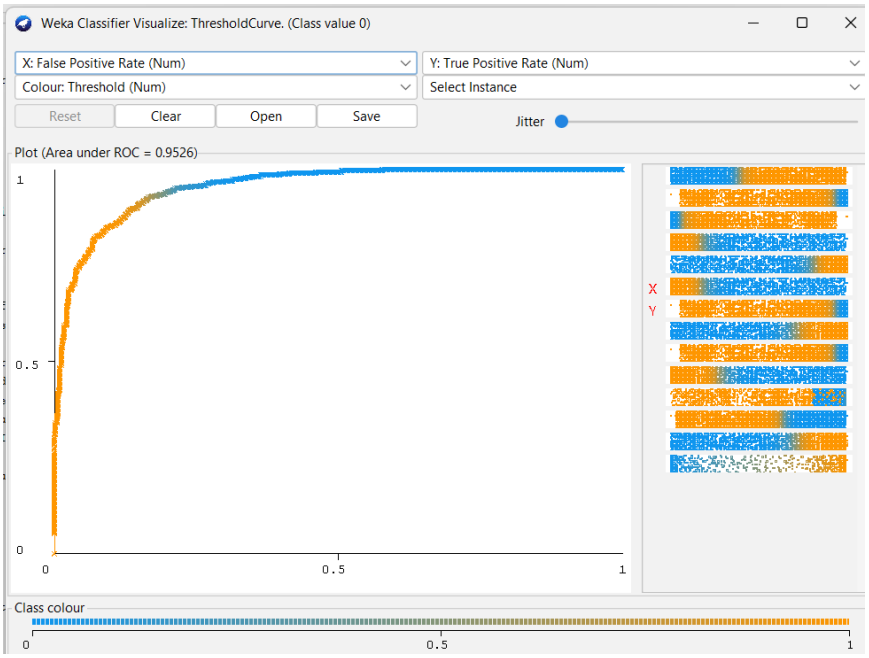


Figura 5.12 Vijat ROC për algoritmin Naïve Bajes në platformën WEKA

5.6 Vlerësimi i algoritmit të Decision Tree C4.5 Learning

Për të ekzekutuar algoritmin e Pemës së Vendimit C4.5, ndjekim një proces të ngjashëm me atë të Naïve Bayes. Në panelin Classifiers, fillimisht zgjedhim kategorinë Trees dhe pastaj zgjedhim opsionin J48, i cili është një emër alternativ për këtë algoritëm.

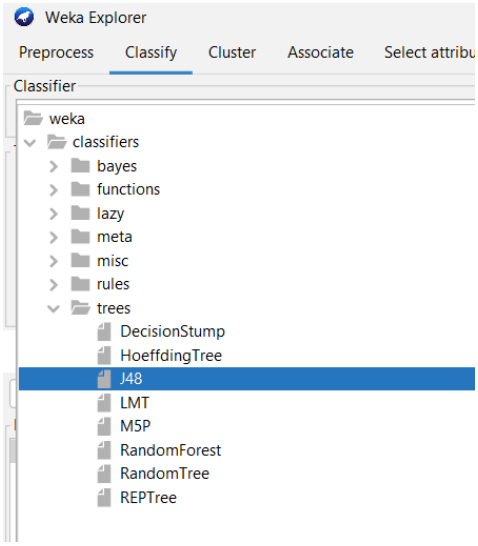


Figura 5.13 Ekzekutimi i algoritmit të pemës së vendimit C4.5 Learning në platformën WEKA

Classifier output

```
Number of Leaves  :     103

Size of the tree :      205


Time taken to build model: 0.15 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances        4249               92.3495 %
Incorrectly Classified Instances       352                7.6505 %
Kappa statistic                          0.8396
Mean absolute error                      0.1063
Root mean squared error                  0.2589
Relative absolute error                 22.2489 %
Root relative squared error             52.9779 %
Total Number of Instances             4601

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.939    0.100    0.935      0.939    0.937      0.840    0.945     0.945     0
                 0.900    0.061    0.906      0.900    0.903      0.840    0.945     0.894     1
Weighted Avg.    0.923    0.085    0.923      0.923    0.923      0.840    0.945     0.925

=== Confusion Matrix ===

    a    b   <-- classified as
 2618  170 |    a = 0
  182 1631 |    b = 1
```

Figura 5.14 Rezultatet e ekzekutimit të algoritmit të pemës së vendimit C4.5 Learning në platformën WEKA

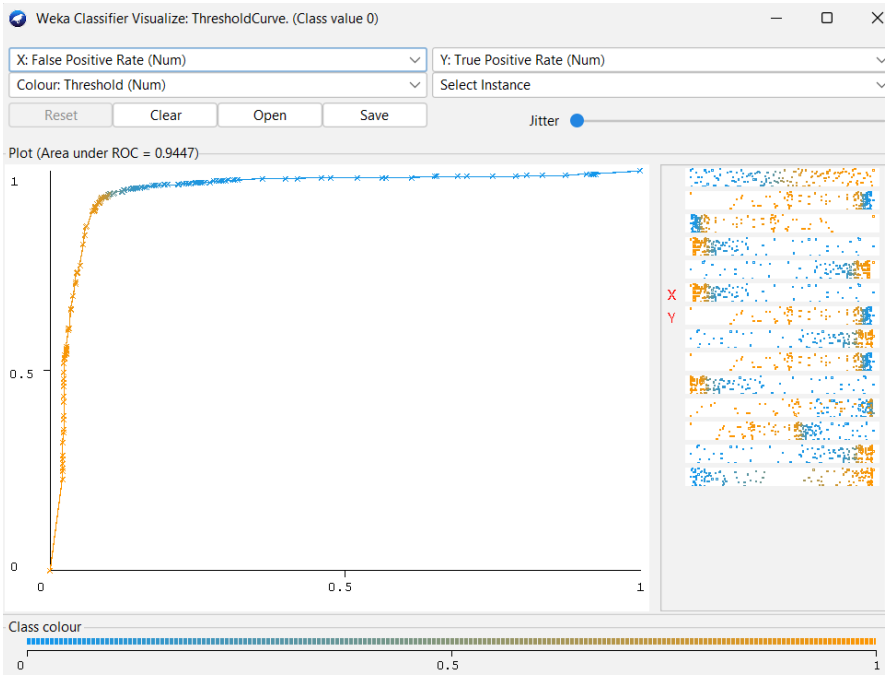


Figura 5.15 Vijat ROC për algoritmin e pemës së vendimit C4.5 Learning në platformën WEKA

5.7 Vlerësimi i algoritmit të Support Vector Machine

Për të ekzekutuar algoritmin e Makinës Vektoriale Mbështetëse, në panelin Classifiers zgjedhim panelin Functions dhe më pas zgjedhim opsionin SMO, i cili është një tjetër emër për algoritmin SVM.

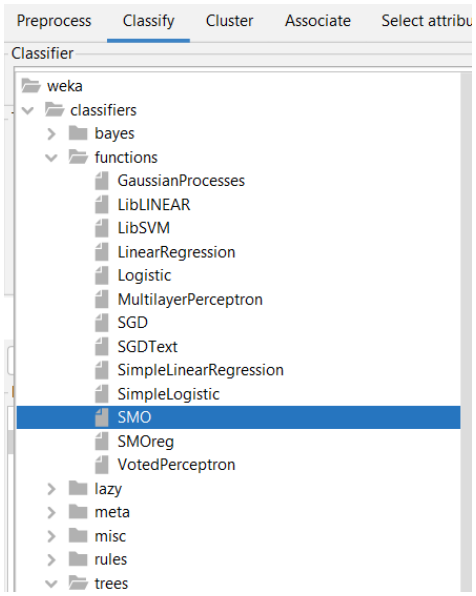


Figura 5.16 Ekzekutimi i algoritmit të Support Vector Machine në në platformën WEKA

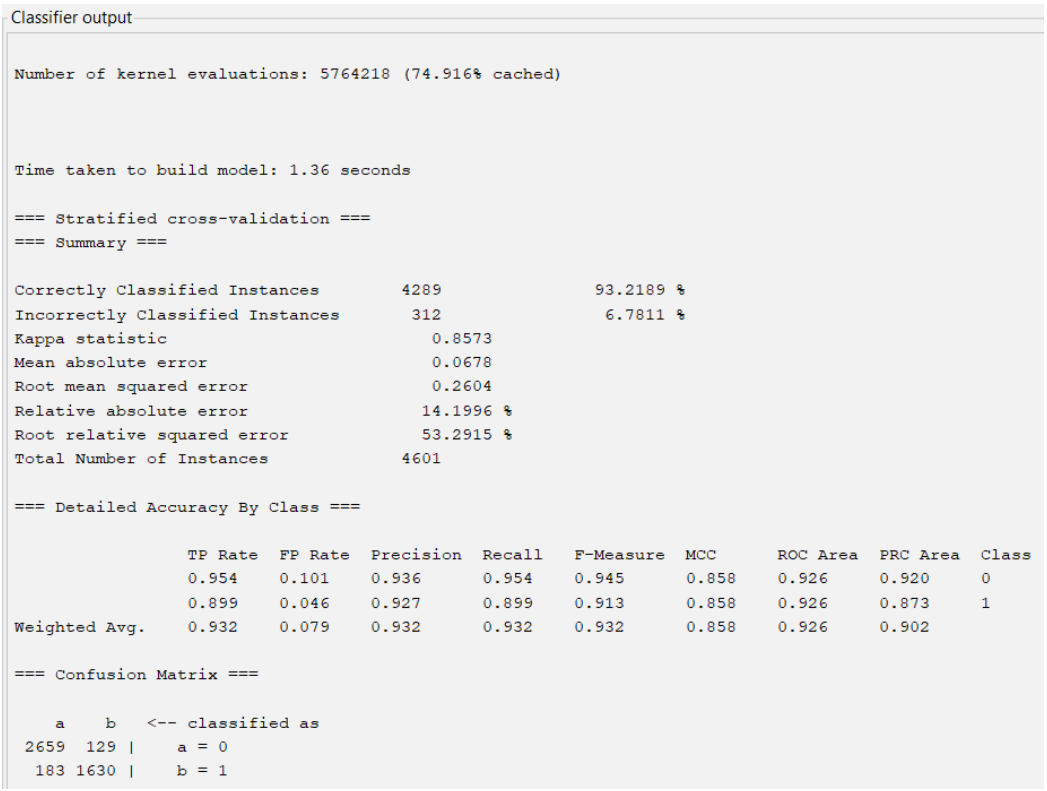


Figura 5.17 Rezultatet e vlerësimit të algoritmit të Support Vector Machine në platformën WEKA

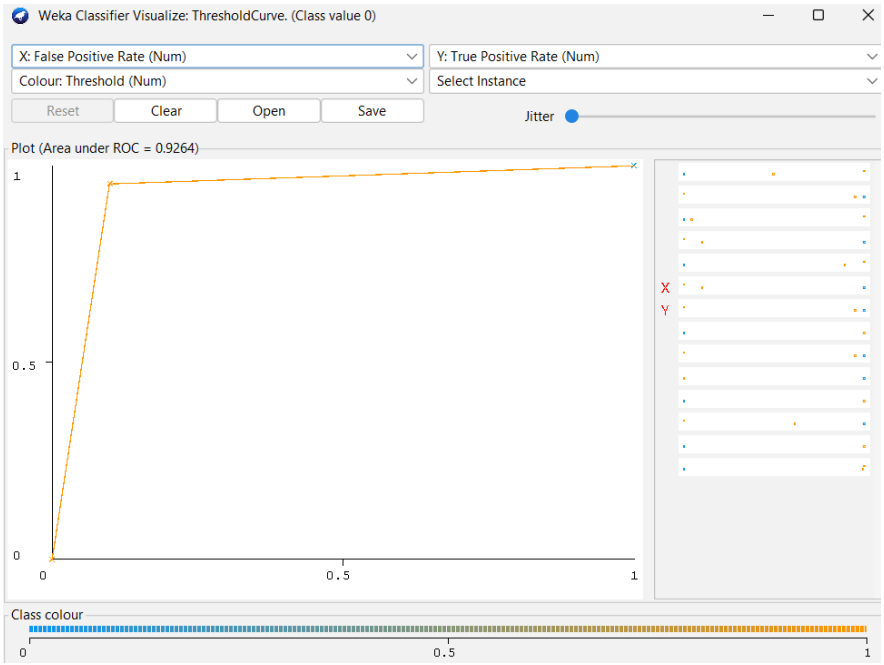


Figura 5.18 Vija ROC për algoritmin e Support Vector Machine në platformën WEKA

PËRFUNDIME

Klasifikimi i email-eve të padëshiruara (apo spam) nga ato legjitime është një çështje me rëndësi të madhe në botën e sotme. Ky problem është trajtuar nga studiues të ndryshëm, me qëllim që të zvogëlohet koha dhe përpjekjet e përdoruesve të email për të menaxhuar emailet e tyre. Në këtë punim diplome, janë përdorur disa algoritme të machine learning të zbatuar në platformën WEKA për të identifikuar emailet e padëshiruara. Qëllimi kryesor i këtij studimi është të vlerësojë tre algoritmet e machine learning si filtra të emaileve: Naïve Bayes, Decision Tree C4.5 dhe Support Vector Machine. Performanca e këtyre algoritmeve është vlerësuar duke përdorur parametra të ndryshëm, si për shembull numri i rasteve të klasifikuara saktë, numri i rasteve të klasifikuara gabim, statistikat Kappa, gabimi mesatar absolut, gabimi mesatar i rrënjës katrore dhe vijat ROC.

Algoritmet janë testuar në WEKA për të treguar rezultatet më të mira me bazën e të dhënave të përdorur. Në aspektin e kohës së nevojshme për trajnimin e modelit, Naïve Bayes rezulton të jetë shumë më i shpejtë, me një kohë ekzekutimi prej 0.02 sekondash, ndërsa Pema e Vendimit C4.5 dhe Makina Mbështetëse Vektoriale kërkojnë më shumë kohë, përkatësisht 0.15 sekonda dhe 1.36 sekonda. Për sa i përket numrit të rasteve të klasifikuara saktë, Makina Mbështetëse Vektoriale ka performancën më të lartë me 93,2189%, ndërsa Naïve Bayes dhe Pema e Vendimit C4.5 kanë performancë përkatësisht 88,546% dhe 92,3495%. Nga këto rezultate, siç pritej nga teoritë, Naïve Bayes është më i shpejtë për ndërtimin e modelit, por ka një performancë më të ulët në klasifikimin e saktë të rasteve.

Rezultatet e analizave të kryera me WEKA, evidentojnë se nuk ka një algoritëm të vetëm të machine learning që mund të ofrojë modelin optimal të klasifikimit për të gjitha tiparet e të dhënave. Performanca e modelit të parashikuar ndikohet nga përzgjedhja e veçorive të përdorura. Kjo nënkupton se algoritmet e ndryshme të klasifikimit janë zhvilluar për të qenë më eficent në lloje të caktuara të të dhënave. Ndërkaq, algoritmet e machine learning kanë potencial për të përmirësuar shpejtësinë e procesit të klasifikimit për të dhënat e reja të trajnuara. Në praktikë, një algoritëm i machine learning mund të përfitojë nga një klasifikues i trajnuar për të ndihmuar në klasifikimin pjesërisht të emaileve të reja të dërguara për trajnim. Për shkak se shumë prej këtyre emaileve kanë përmbajtje të ngjashme, është e mundur të identifikohet një përfaqësues i rëndësishëm dhe të klasifikohet me disa veprime të thjeshta, duke e reduktuar kohën e nevojshme për përdoruesit e internetit. Algoritmet e machine learning kanë një avantazh të qartë, duke u aplikuar efektivisht përmes përdorimit të të dhënave të trajnuara më herët të diversifikuara.

Sugjerime për studim të mëtejshëm

Pas përfundimit të analizave të algoritmeve të machine learning në këtë studim, është vërejtur se efikasiteti i tyre ishte i pranueshëm, megjithatë, ka nevojë për përmirësim duke përfshirë algoritme të tjera të klasifikimit për të trajtuar rastin e rritjeve të numrit të emaileve spam dhe të bazave të të dhënave të mesazheve të emailit. Për projektet e ardhshme, është e rëndësishme të eksperimentohet me më shumë baza të dhënash dhe algoritme të reja për të vlerësuar performancën e algoritmave. Vlen të konsiderohet koha e përdorur për vlerësimin e modeleve, efikasiteti i përgjithshëm i algoritmeve dhe aspektet teknike të implementimit. Përveç kësaj, ka interes për shqyrtimin e algoritmeve të machine learning të panjohura në WEKA, siç është Kodimi Dinamik i Markovit dhe të eksplorohet mundësia e tyre e aplikimit në një platformë të re me kapacitete më të përshtatshme për kërkime të thelluara.

Lista e referencave

(Microsoft, 2023) Microsoft. (2023, May 12). *Why is it called spam, anyway? A brief inbox history*. Marre nga: https://www.microsoft.com/en-us/microsoft-365-life-hacks/privacy-and-safety/what-is-email-spam

Arora R., Suman R. (2012) Comparative Analysis of Classification Algorithms on Different Datasets using WEKA, International Journal of Computer Applications, Volume 54, No.13, September 2012 p.22-25.

Awad M., Foqaha M. (2016) Email i padëshiruar classification using hybrid approach of RBF neural network and particle swarm optimization, Int. J. Netw. Secur. Appl. 8 (4).

Fawcett T. (2006) An introduction to ROC analysis. Pattern Recogn. Lett. 27.8 (June 2006), pp. 860–873.

Burges C. J. C. (1998) A Tutorial on Support Vector Machines for Pattern Recognition. Data Min. Knowl. Discov. 2.2 (June 1998), pp. 120–167.

Cormack G V. (2006) Email Spam Filtering: A Systematic Review. Foundations and Trends in Information Retrieval Vol. 1, No. 4, p. 334–455.

Desai A., Rai S., (2011) Analysis of ML Algorithm using WEKA, International Journal of Computer Applications, p.25-33.

Fdez-Riverola F., IglesiasE.L., Diaz F., Mendez J.R., Corchado J.M. (2007) Spam Hunting: an instance-based reasoning system for spam labelling and filtering, Decis. Support Syst. 43 (3) 720–734.

Torabi Z.S., Nadimi-Shahraki M.H., Nabiollahi A. (2015) Efficient support vector machines for spam detection: a survey. (IJCSIS), Int. J. Comput. Sci. Inf. Secur. 13 (1) 10–29.

Scholkopf B., Smola A.J. Learning with Kernels: Support Vector Machines, Regularization, Optimization, and beyond, MIT press, 2002.

Chakraborty S., Mondal B., (2012) Spam mail filtering technique using different DT classifiers through data mining approach - a comparative performance analysis, Int. J. Comput. Appl. 47 (16) 27–30, 0975 – 888.

Androutsopoulos I., Koutsias J., Chandrinos K.V., Paliouras G., Spyropoulos C.D. An evaluation of naive Bajesian anti-spam filtering, in: Proceedings of 11th European Conference on ML (ECML 2000), Barcelona, 2000, pp. 10–18.

Rusland N. F., Wahid N., Kasim Sh., Hafit H. (2017) Analysis of Naıve Bajes algorithm for email spam filtering across multiple datasets, IOP Conf. Ser. Mater. Sci. Eng. 226, 012091.

Dada E. G., Bassi J. S., Chiroma H., Abdulhamid Sh. M., Adetunmbi A. O., Ajibuwa O. E. (2019) ML for email spam filtering: review, approaches and open research problems 5, 2019.

Guzella T. S., Caminhas W. M. (2009) Review: A review of ML approaches to SE. In: Expert Syst. Appl. 36.7 (Sept. 2009), pp. 10209–10221.

Cormack G V. (2006) Email Spam Filtering: A Systematic Review. Foundations and Trends in Information Retrieval Vol. 1, No. 4, p. 332–457.

Hajes B. (2007) How many ways can you spell Viagra, American Scientist 95 (2007).

Heylighen F., JoslynC., Entropy and Information. PrincipaCybernetica Web, 2001.

Quinlan J. R., C4.5: programs for ML. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1993.

Fletcher T. “SVM Explained”. University College London. Dec. 2008.

Fawcett T. (2006) An introduction to ROC analysis. Pattern Recogn. Lett. 27.8 (June 2006), pp. 861–874