SVEUČILIŠTE U ZAGREBU

FAKULTET ORGANIZACIJE I INFORMATIKE

V A R A Ž D I N

**Ivica Čelig**

**Dino Kliček**

BAYES SPAM FILTER

Projektni rad

Varaždin, siječanj 2019.

Sadržaj

[Sadržaj ii](#_Toc535149976)

[1. Uvod 1](#_Toc535149977)

[2. E-mail pošta 2](#_Toc535149978)

[2.1. Bayesov klasifikator 3](#_Toc535149979)

[2.2. Bayesov klasifikator i spam mail 4](#_Toc535149980)

[2.2.1. Trening faza 5](#_Toc535149981)

[2.2.2. Faza predviđanja 5](#_Toc535149982)

[2.3. Prednosti i nedostaci Bayesovog klasifikatora 6](#_Toc535149983)

[3. Izrada Bayesovog spam filtra 8](#_Toc535149984)

[3.1. Čitanje e-mailova 9](#_Toc535149985)

[3.2. Učenje spam i ham riječi 10](#_Toc535149986)

[3.3. Bayes klasifikator 12](#_Toc535149987)

[4. Testiranje funkcionalnosti 16](#_Toc535149988)

[5. Zaključak 20](#_Toc535149989)

[Popis literature 21](#_Toc535149990)

[Popis slika 22](#_Toc535149991)

1. Uvod

E-pošta ili *e-mail* danas pruža savršen način za slanje milijuna oglasa bez troška za pošiljatelja, a tu nesretnu činjenicu danas u velikoj mjeri iskorištava jako velik broj organizacija koje masovno šalju takvu poštu milijunima primatelja. Kao rezultat toga, e-mail sandučići milijuna ljudi postaju pretrpani ovom takozvanom neželjenom masovnom e-poštom poznatijom kao *spam* ili možemo reći "bezvrijedna pošta". Budući da je za slanje nevjerojatno jeftina, neželjeni e-mailovi uzrokuju velike probleme internetskoj zajednici. Problemi nastaju ponajprije zbog velike količine neželjenog prometa između poslužitelja što uzrokuje kašnjenja u isporuci legitimnih (ne spam) poruka e-pošte. Također, oni ljudi s *dial-up* pristupom internetu moraju trošiti propusnost za preuzimanje neželjene pošte što uzrokuje pak nepotrebni financijski trošak. Međutim, poruke bez neželjene pošte su rijetke, u kontekstu one mjere u kojoj se ta dva pojma mogu međusobno zamjenjivati zavisno od osobe do osobe. Treba imati na umu da poruke koje šalju virusi ne smatraju se neželjenom poštom, iako se i one mogu slati slijepo velikom broju korisnika, dok postoje i poruke neprikladnog sadržaja koji ne bi trebao biti dostupan djeci.

Stoga će se ovaj projektni rad posvetiti razvrstavanju neželjenih poruka na automatizirani način. Običnom korisniku je potrebno određeno vrijeme i javlja se mogućnost pogreške prilikom brisanja takve vrste pošte. Cilj ovog rada je izgraditi Više anti-spam filter, softverski alati koji automatski prepoznaje dolazne neželjene poruke. Dakle, radi se o *machine* *learning* sustavu koji na temelju naučenih podataka o tome što je dobar mail, a što loš može donositi zaključke za buduće mailove. Spomenuto zaključivanje temeljiti će se na Bayesovom teoremu - popularnoj statističkoj tehnici koja se koristi za filtriranje e-pošte za otkrivanje neželjene pošte. Prvi dio rada bazirati će se na opisivanju teorije koja stoji iza svega toga – što je to spam e-mail, Bayesov teorem i kakva je njihova međusobna poveznica. U drugom dijelu rada će biti prikazan postupak izgradnje takvog spam filtera kroz objašnjenje implementacije programskog koda i vidjeti će se njegov rad na stvarnom pretincu e-pošte.

1. E-mail pošta

U današnjem računalno povezanom društvu, e-pošta je postala najbrža i najekonomičniji oblik komunikacije. Elektronska pošta ili skraćeno e-mail je brz, učinkovit i jeftin način razmjene poruka putem Interneta. Bilo da je riječ o osobnoj poruci člana obitelji, poruci šefa u cijeloj tvrtki, istraživačima diljem kontinenata koji dijele najnovija otkrića ili astronautima koji su u kontaktu sa svojom obitelji (putem e-mail adresa ili IP telefona), e-pošta je poželjna kao sredstvo za komunikaciju. Koristi se u svijetu od 2,3 milijarde korisnika, a kako navode Bhowmick i Hazarika (2016) u svome radu, predviđa se daljnje povećanje korištenja e-pošte do brojke od 4,3 milijarde korisnika. Međutim, sve veća ovisnost o e-pošti izazvala je pojavu mnogih problema uzrokovanih "nezakonitom" e-poštom, tj. *spamom*. Spam e-pošta je neželjena e-pošta koja je poslana nasumice i koja nije odobrena te je to obično masovna pošta. Tijekom nekoliko desetljeća volumen neželjene e-pošte povećao se eksponencijalno i nije samo smetnja nego i sigurnosna prijetnja budući da se nastavlja razvijati i slati s ciljem da napravi ozbiljnu štetu pojedincima i poduzećima. Tu se može raditi o uznemiravanju pojedinačnih korisnika jer dobivaju takvu e-poštu, gubitak produktivnosti rada, zlouporaba propusnosti mreže, gubitak prostora za pohranu datotečnog poslužitelja i računsku snagu, širenje virusa, crva i trojanskih konja, ali i financijske gubitke zbog krađe identiteta, uskraćivanje usluga (DOS napada) i slično.(Bhowmick & Hazarika, 2016)

Dakle, elektroničku poštu dijelimo na neželjenu (***spam***) i željenu (tzv. ***ham***) poštu. Neželjena pošta se još naziva *junk* *mail*, dok za razliku od takve vrste pošte postoje poruke koje su legitimne i originalne prirode, a koje se nazivaju ne spam ili ham pošta. S obzirom da se spam pošta pojavljuje svakodnevno u našem pretincu predložene i korištene su proteklih nekoliko godina brojne *anti-spam* tehnike za borbu protiv neželjene pošte kao što su različita dopuštenja, crne liste pošiljatelja, listanje imena domena, filtriranje na temelju ključnih riječi, heurističko filtriranje i tako dalje. Međutim, takve tehnike zahtijevaju teško održavanje i ne mogu postići vrlo visoku ukupnu točnost. Stoga postoji potreba za razvojem klasifikatora koji će omogućiti klasificiranje pošte kao spam ili ham pošte. Ovaj rad bavi se upravo takvim jednim klasifikatorom koji se naziva Naivan Bayesov klasifikator. Naivni Bayes klasifikatori su tehnika koja je popularna već niz godina i nedvojbeno su najpoznatiji klasični klasifikatori spama. Posjeduje naziv "naivan“ jer zanemaruje moguće zavisnosti ili korelacije među ulazima i reducira multivarijatni problem na skup različitih varijabli. Koristi se probabilistički pristup zaključivanju za što nije potrebna nikakva komplicirana iterativna shema procjene parametara, kao u diskriminacijskoj analizi. Također, takav klasifikator je lako konstruirati, interpretirati, a s druge strane je iznenađujuće učinkovit i lako se može primijeniti na velike skupove podataka što ga čini iznimno popularnim među korisnicima. (Jensi, 2008)

* 1. Bayesov klasifikator

Bayesovi klasifikatori ili filtri kombiniraju sve koncepte statistike dolazne poruke do ukupne ocjene pomoću Bayesovog izračuna vjerojatnosti. Konačno, odluka o filtriranju se donosi na temelju rezultata i unaprijed definiranog praga. Dakle, prema Tretyakov (2004), Bayesov klasifikator definira se kao vjerojatnosti klasifikator koji za klasifikaciju koristi Bayesov teorem i oslanja se na pretpostavku neovisnosti između značajki. Takav klasifikator zahtijeva učenje koje se provodi nad skupom kojeg opisuju značajke i označen je klasama skupa . Nakon što klasifikator nauči iz danog skupa dobiva se naučeni klasifikator koji omogućava predviđanje, odnosno klasifikaciju novih instanci. Dakle, nad danim značajkama računa se *Maximum a posteriori* ili najvjerojatnija vrijednost :

Promatrajući prethodnu formulu možemo zaključiti da se ona oslanja na Bayesov teorem koji zapisujemo:

To su temelji Bayesovog klasifikatora. U ovom radu će se raditi s **Naivnim Bayesovim klasifikatorom** koji se oslanja na neovisnost između značajki u odnosu na klasifikacijsku vrijednost. Dokaz formule Naivnog Bayes klasifikatora slijedi kroz razdvajanje dokaza na neovisne dijelove.

Prema Bayesovom teoremu vrijedi formula:

Gornju formulu možemo izraziti u slijedećem obliku:

Kako se Naivni Bayes oslanja na neovisnost između značajki u odnosu na klasifikacijsku vrijednost, nazivnik je uvijek konstanta (vjerojatnost za do je jednaka) za bilo koji dani ulaz čime ga možemo izbaciti i izraz Naivnog Bayesovog teorema zapisati u obliku:

Dakle, učenje Bayesovog klasifikatora mora uključiti računanje vjerojatnosti pojavljivanja klase u odnosu na dane značajke . Stvaranjem skupa ovih procjena dobivamo naučeni klasifikator koji se kasnije koristi za klasificiranje novih primjera na korištenjem izraza . U slijedećem pod poglavlju će biti prikazan primjer radi boljeg razumijevanja prethodno izrečenog.

* 1. Bayesov klasifikator i spam mail

Bayesovo filtriranje popularna je statistička tehnika koja se koristi za filtriranje e-pošte, odnosno za otkrivanje neželjene pošte. Razdvajanjem pretinca poruka e-pošte na dvije različite hrpe, jedna gomila sadrži spam, a druga ne sadrži neželjenu poštu, može se izračunati statističku vjerojatnost pripadnosti određene riječi spam pošti ili ne spam pošti. Kada se stvori statistička pregledna tablica, može se koristiti za kategorizaciju budućih poruka e-pošte. Iako je Bayesovo filtriranje popularizirano za upotrebu u otkrivanju neželjene pošte, ono ima i mnoge druge upotrebe izvan e-pošte. (Metsis, Androutsopoulos & Paliouras, 2006)

Zapravo, gornji način omogućava klasifikaciju koji se koristi u različitim *machine* *learning* sustavima. *Machine* *learning* ili strojno učenje je područje računalne znanosti gdje računala mogu naučiti nešto raditi, bez potrebe da se izričito programiraju za zadatak. Prvo, napravljen je algoritam za gledanje određenog skupa podataka, kako bi ga se treniralo za zadatak. Zatim dajemo podatke o algoritmu koje nikada prije nisu vidjeli i izvršavamo zadatak na tim podacima. Stoga se može smatrati da algoritam strojnog učenja ima dvije faze: "trening" i "predviđanje". Za svaku od tih faza koristimo različite matematičke metode kao što su klasifikacija ili regresija. Kod klasifikacije postoje algoritmi čiji rezultat je kategorizacija podataka. Na primjer, algoritam koji uzima medicinske informacije o pacijentu i proizvodi dijagnozu koja može biti samo jedna od mogućnosti: "nema raka", "rak pluća" ili "rak debelog crijeva". S druge strane, u regresiji izlazni tipovi su kontinuirano vrednovani. Na primjer, kod predviđanja cijena kuća, predviđena cijena ovisit će o veličini zemljišta, lokaciji i tako dalje. Za razliku od klasifikacije, ovdje ne postoji rezultat kojim bi se lijepo kategorizirali podaci. (Martins & Cormack, 2009)

U ovom projektnom radu korištena je klasifikacija s ciljem učenja, odnosno treniranja našeg filtar pomoću zbirke željenih i nepoželjnih e-poruka. Dakle, mora se dati pravi, realne e-mail poruke kako bismo trenirali filtar, a kasnije u fazi predviđanja, koristili taj trening za dobivanje izlaza za određenu poruku koja može biti ili "spam" ili "ham". Dakle, ovaj filtar je zapravo primjer nadziranog klasifikacijskog algoritma.

* + 1. Trening faza

Kod trening faze naš filtar trebamo naučiti koje to riječi se često pronalaze u spam pošti. Iskusno govoreći to su najčešće riječi poput „popust“, „poklon“, „sniženje“ i slično. Te riječi se rjeđe javljaju u ham pošti, odnosno onoj željenoj pošti gdje se zapravo najčešće javljaju riječi vezane uz naš posao, studiranje i slično. Dakle, tijekom faze treninga, našem programu dajemo naredbu da pogleda skup neželjene pošte (spam) ili željene e-pošte (ham). Za svaku različitu riječ u poruci e-pošte zabilježit će se vjerojatnost s kojom se riječ pojavljuje u spamu ili hamu. Tako na primjer imamo 10000 e-mail poruka u našem skupu podataka za trening od kojih je 6000 mailova spam, a preostalih 4000 ham mailovi. Također, znamo da se riječ *popust* pojavljuje u 2000 spam mailova, dok u ham mailovima samo 10 puta. Dakle, možemo lako izračunati i . Ovdje se može vidjeti da se ne izračunava samo spam vrijednost neke riječi, već je potrebno izračunati i ham vrijednost. Razlog tome je da ako spam i ham mailovi često sadrži riječ "automobil", imali bismo visoke vrijednosti spama i hama. Ako ne uzmemo u obzir ham e-pošte kod izračuna, onda bismo svaku poruku e-pošte koja sadrži "automobil" klasificirali kao neželjenu poštu.

* + 1. Faza predviđanja

Nakon što smo u trening fazi izračunali spam i ham vrijednost svake riječi mala, sada možemo lako izračunati da li je neki mail spam ili ham. Prema gore opisanom primjeru znamo da je vjerojatnost maila da je spam , a vjerojatnost maila da je ham . Sada, recimo da imamo novu poruku e-pošte koja sadrži proizvoljnu riječ *R* u njoj. Želimo saznati da li je poruka spam, s obzirom da sadrži riječ *R*. To je upravo suprotna situacija od pronalaženja spam vrijednosti za riječ popust, dok se ova situacija može se predstaviti kao . Koristeći se Bayesovim teoremom tu vjerojatnost možemo izračunati prema formuli:

Ako u gornju formulu uvrstimo vrijednosti dobivene iz faze testiranja i predikcije, možemo zapisati slijedeće:

Do sada smo razmatrali samo jednu riječ. Naravno, e-mail se obično sastoji od više riječi, pa Bayesov teorem upotrebljavamo toliko puta koliko je potrebno da bismo kako bismo saznali ukupnu vjerojatnost da je taj mail spam. Ovaj model pretpostavlja da se sve riječi jednako mogu pojaviti na bilo kojem mjestu u poruci. Međutim, zbog gramatičkih pravila jezika, to se nikada ne događa, tako da ovaj model nije savršen. Međutim, to je dovoljno dobro za naše potrebe, a mnogi filteri za neželjenu poštu koriste ovaj model zbog svoje jednostavnosti.

Može se dogoditi da filtar naiđe na potpuno novu riječ kada klasificira novu, n1epoznatu e-poštu. Kod takve situacije, najprije se odbacuju takve riječi jer ne postoji način na koji možemo nešto predvidjeti o njima. Nakon toga, pretpostavimo da u e-pošti ima *n* različitih riječi . Tada za svaku riječ *R* moramo saznati vjerojatnost da pripada spam mailu, Također, znamo da vrijedi . Sada lako možemo saznati da li je neki mail spam ili ne upotrebom formule Naivnog Bayesovog teorema i normalizacije dobivenih vrijednosti:

Ukoliko se prema gornjoj formuli dobije vrijednost koja je veće primjerice od 0.5 tada možemo zaključiti da se radi o spam mailu, dok u suprotnom se radi o ham mailu.

Također valja spomenuti još nekoliko stvari. U formuli za u obzir su uzete vrijednosti za i . Međutim, u stvarnim situacijama, može biti visok kao npr. 0.8, što bi dovelo do pogrešne klasifikacije mailova gdje bi se ham mailovi klasificirali kao spam mailovi. S obzirom da predviđamo nove mailove (koji nisu dio trening podatak), nemamo razloga za sumnju da je to spam, pa je korišteno koristiti i = 0.5. Drugim riječima, pretpostavljamo da je jednako vjerojatno da će poruka biti ili spam ili ham. Tako da te vrijednosti predstavljaju konstante u nazivniku Naivnog Bayesevog teorema koje se mogu izbaciti van iz formule. Nadalje, postoji još jedna optimizacija koja se može napraviti. U svakom mailu mogu se pronaći neke uobičajene riječi poput "ako", "biti", "tada" i tako dalje. Nema potrebe razmatrati te riječi jer nisu korisne za otkrivanje je li poruka neželjena.

* 1. Prednosti i nedostaci Bayesovog klasifikatora

Upotreba Bayesovog klasifikatora, odnosno filtra ima svoje dobre i loše strane. Pozitivne strane takvog načina klasificiranja su (Bhowmick & Hazarika, 2016):

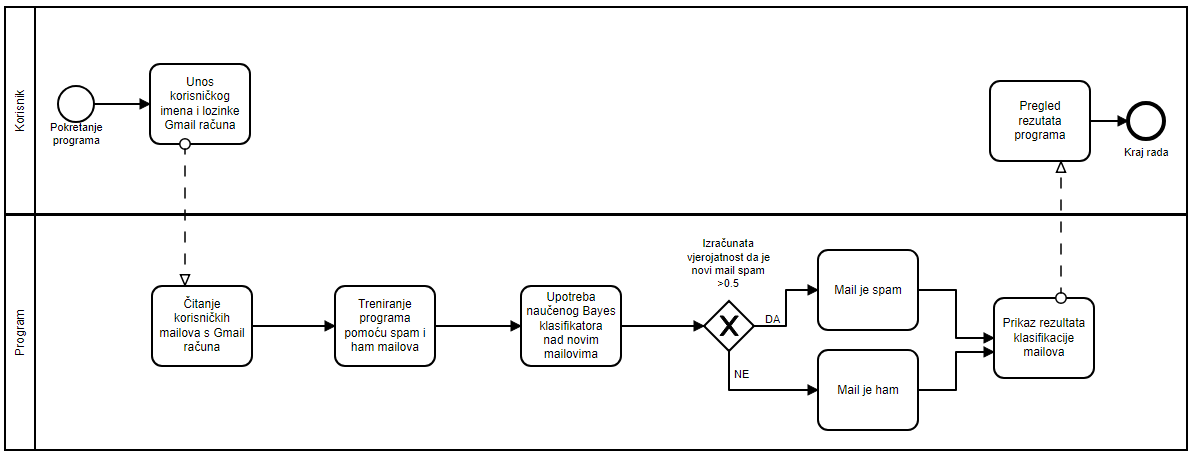
* Lako se prilagođava korisniku ili organizaciji koja ga koristi budući da se trenira s njihovim podacima. Također ne ovisi o jeziku.
* Lako i brzo može predvidjeti skup podataka. Također dobro funkcionira u predviđanju više međusobno povezanih podataka ili riječi.
* Dosta niski stupanj pogreške – npr. da kaže za neki ham mail da je spam.
* Samoprilagođavajući pošto uči konstantno kroz nove spam i ham mailove.
* Kada pretpostavka o neovisnosti vrijedi, klasifikator Naivnog Bayesa ima bolje performanse u usporedbi s drugim modelima kao što je logistička regresija, a ujedno je potrebno manje podataka za trening.
* Dobar je u slučaju kategorijskih ulaznih varijabli u usporedbi s numeričkim varijablama. Za numeričku varijablu pretpostavlja se normalna distribucija (zvonasta krivulja, što je jaka pretpostavka).

S druge strane, Bayesov klasifikator bilježi i neke negativnosti poput (Metsis, Androutsopoulos & Paliouras, 2006):

* Jedno ograničenje Bayesovih filtara je da zanemaruje korelaciju između ulaza ili događaja; tj. takvi filtri ne smatraju da se riječi "posebne" i "ponude" najvjerojatnije pojavljuju zajedno u spam e-pošti nego u ham e-pošti
* Lako trovanje riječima koji nisu karakteristične za ne spam mailove što uzrokuje krivu klasifikaciju
* Ako kategorička varijabla ima kategoriju (u skupu podataka za trening), koja nije bila uočena u skupu podataka treninga, tada će model dodijeliti 0 (nultu) vjerojatnost i neće moći napraviti predviđanje. To se često naziva "nultom frekvencijom". Da bismo to riješili, može se koristiti tehnika izglađivanja. Jedna od najjednostavnijih tehnika izglađivanja naziva se Laplaceova procjena.
* Još jedno ograničenje Naivnog Bayesa je pretpostavka nezavisnih prediktora. U stvarnom životu gotovo je nemoguće dobiti skup prediktora koji su potpuno neovisni.

1. Izrada Bayesovog spam filtra

U ovom poglavlju biti će objašnjen postupak izrade programa koji omogućuje klasifikaciju mailova korisnika sa njegovog Gmail računa. S ciljem što boljeg dočaravanja što će program raditi i koje mogućnosti će korisnik imati na raspolaganju, izrađen je dijagram procesa toka korisnika i programa sa pripadajućim aktivnostima koji je vidljiv na slijedećoj slici.



Slika 1. Tijek procesa korisničkog korištenja programa i izvođenja programa

Slika 1. prikazuje dvije uloge koje ovaj rad obuhvaća, a to je uloga korisnika koji unosom svojeg korisničkog imena i lozinke Gmail računa u program dobiva klasificirane mailove u pretincu na spam ili ham. To klasificiranje mailova obavlja program kroz nekoliko koraka. Program prvo, temeljem korisničkog imena i lozinke korisnika, skida mailove i sprema ih za kasniju klasifikaciju. Kako bi se klasifikacija uspješno izvršila program prvo uči (trenira) koje riječi češće pripadaju spam mailovima, a koje ham mailovima – ti mailovi su ugrađeni u program. Na taj način je omogućena klasifikacija novih, dosad nepoznatih mailova odnosno riječi koje taj mail sadržava. Zahvaljujući Bayesovom klasifikatoru program izračunava vjerojatnost svakog pojedinog učitanog maila korisnika da li je on spam. Ukoliko je ta vjerojatnost veća od 0.5 tada je mail koji se procesuira proglašen kao spam, dok u suprotnom ako je izračunata vjerojatnost manja od 0.5 tada je taj mail proglašen ham mailom. Nakon što program klasificira sve mailove korisnika – korisniku omogućava prikaz rezultata klasifikacije.

U nastavku ovog poglavlja slijedi opis programskog koda gore spomenutog programa. Program je izrađen u C# programskom jeziku koji omogućava objektno orijentirano programiranje sačinjeno od više zasebnih klasa koje se mogu višestruko iskoristiti.

* 1. Čitanje e-mailova

Prema Slici 1. program prvo učitava mailove korisnika iz njegovog Gmail poštanskog sandučića. S tom svrhom stvorene je klasa *MailLoader* s programskim kodom:

class MailLoader

{

public List<string> titles = new List<string>();

public List<string> summaries = new List<string>();

public List<string> senders = new List<string>();

public void readMail(string username, string password)

{

try

{

System.Net.WebClient objClient = new System.Net.WebClient();

string response;

string title;

string content;

string name;

XmlDocument doc = new XmlDocument();

objClient.Credentials = new System.Net.NetworkCredential(username,password);

response = Encoding.UTF8.GetString(

objClient.DownloadData(@"https://mail.google.com/mail/feed/atom"));

response = response.Replace(

@"<feed version=""0.3"" xmlns=""http://purl.org/atom/ns#"">", @"<feed>");

doc.LoadXml(response);

foreach (XmlNode node in doc.SelectNodes(@"/feed/entry"))

{

title = node.SelectSingleNode("title").InnerText;

titles.Add(title);

content = node.SelectSingleNode("summary").InnerText;

summaries.Add(content);

name = node.SelectSingleNode("author/name").InnerText;

senders.Add(name);

}

}

catch (Exception exe)

{

MessageBox.Show("Neispravno korisničko ime ili lozinka");

}

}

Klasa *MailLoader* omogućava dohvaćanje naslova, sažetka i pošiljatelja mail s korisničkog Gmail računa. Funkcija readMail je omoogućena zahvaljujući Gmail Atomu, sustavu koji olakšava primanje redovitih ažuriranja s web-lokacija s vijestima, blogova i/ili Gmaila na jednom mjestu. Ono što takav sustav zahtijeva je korisničko ime i lozinku Gmail računa koje korisnik na početku unosi u program. Preko Atoma dobivaju se podaci u obliku XML dokumenta koji se zatim dekomponira na naslov, sadržaj i pošiljatelja maila. Ti podaci se pohranjuju u listu za daljnje korištenje, odnosno za klasifikaciju na spam ili ham.

* 1. Učenje spam i ham riječi

Najvažnija stvar ovog programa uz klasifikaciju je njegovo učenje ili takozvano treniranje programa. Ovdje je u tu svrhu stvorene klase *SpamHamReader* i *Trainer*. Klasa *SpamHamReader* ima jednu zadaću, a to je čitanje ugrađene spam i ham mailove korisnika na kojima će se temeljiti učenje programa. U ovom slučaju tipičan spam mail bi glasio: „*Zgrabite popuste za vikend do 42%! Uz kupnju parfema iznad 300 kn dobivate poklon!*“ S druge strane, tipičan ham mail glasi: „*Poštovani, me je uputila na Vas s obzirom da tražite osobu za poziciju Direktora kontrolinga. Budući da Vaša kompanija kontinuirano ulaže u unapređenje poslovnih procesa i procesa izvještavanja, htjela bi doprinijeti tome svojim iskustvom*.“ Prethodne i preostale poruke čita klasa *SpamHamReader* preko programskog koda:

public class SpamHamReader

{

public List<string> Read(string category)

{

List<string> list = new List<string>();

List<string> resourceNames = this.GetType().Assembly.GetManifestResourceNames().ToList();

resourceNames.RemoveRange(0, 3);//prva tri resursa miče koji nisu emailovi

List<string> hamResourceNames = new List<string>();

List<string> spamResourceNames = new List<string>();

foreach(var resName in resourceNames)

{

if (resName.Contains("ham"))

hamResourceNames.Add(resName);

else

spamResourceNames.Add(resName);

}

if (category == "ham")

{

foreach (string resourceName in hamResourceNames)

{

Assembly myAssembly = Assembly.GetExecutingAssembly();

using (var streamReader = new StreamReader(myAssembly.GetManifestResourceStream(resourceName), Encoding.Default, true))

{

var text = streamReader.ReadToEnd();

list.Add(text);

}

}

}

else

{

foreach (string resourceName in spamResourceNames)

{

Assembly myAssembly = Assembly.GetExecutingAssembly();

using (var streamReader = new StreamReader(myAssembly.GetManifestResourceStream(resourceName), Encoding.Default, true))

{

var text = streamReader.ReadToEnd();

list.Add(text);

}

}

}

return list;

}

}

Ova klasa prvo pronalazi sva imena ham i spam mailova u programu kako bi ih kasnije mogla čitati, odnosno spremiti text u listu koja šalje klasi *BayesClassifier* za sprovođenje treninga ili učenja nad tim mailovima.

Trening Bayesovog klasifikatora omogućava klasa *Trainer* s porgramskim kodom:

public class Trainer

{

public Dictionary<string, int> WordCounter

{

get { return wordCounter; }

set { wordCounter = value; }

}

public void Train(string textToParse)

{

string[] textOriginal = textToParse.Split(' ');

List<string> textWithoutNumbers = new List<string>();

foreach (var word in textOriginal)

{

var removedNumberWord = Regex.Replace(word, @"\d", "");

textWithoutNumbers.Add(removedNumberWord.ToLower());

}

List<string> wordsList = new List<string>();

foreach (var word in textOriginal)

{

var removedPunctuationWord = Regex.Replace(word, @"[^\w\s]", "");

wordsList.Add(removedPunctuationWord.ToLower());

}

var commonWords = new CommonWords();

wordsList.RemoveAll(x => commonWords.commonWords.Contains(x));

wordsList.RemoveAll(x => x==String.Empty);

wordsList = wordsList.Distinct().ToList();

foreach (var str in wordsList)

{

if (wordCounter.ContainsKey(str.ToLower()))

{

wordCounter[str.ToLower()] += 1;

}

else

{

wordCounter.Add(str.ToLower(), 1);

}

}

}

}

Klasa *Trainer* omogućuje zapis pojavljivanja određene riječi iz spam ili ham mala u varijablu *WordCounter* koji je tipa *Dictionary* (tip koji omogućuje zapisivanje riječi i njoj pridruženu brojku ponavljanja). Kako bi se to napravila prvo je potrebno obraditi tekst mailova na nčind da se uklone sve brojke, alfanumerički znakovi, prazni stringovi i slično, a valja i spomenuti uklanjanje tipičnihi riječi koje se ponavljaju (npr. „a“, „oko“, „su“, „mi“, „već“ itd.) u bilo spam ili ham mailovima te temeljem kojih ne možemo odrediti pripadnost maila određenoj skupini. Te riječi spremljene su u klasi CommonWords. Nadalje, klasa *Trainer* u funkciji *Train* prebrojava broj ponavaljanja svake riječi u mailovima kako bi se kasnije mogla izračunati vjerojatnost pojavljivanja neke riječi u spam ili ham mail te provoditi daljnji izračuni.

* 1. Bayes klasifikator

Nakon što se preko klase Trainer naučilo koliko se koja riječ pojavljuje u spam, a koliko u ham mailovima, može se prijeći na upotrebu naučenog Bayesovog klasifikatora. Njegov programski kod dan je klasom *BayesClassifier*:

public class BayesClassifier

{

private List<string> spamMails;

private List<string> notSpamMails;

private int countSpamMails;

private int countNotSpamMails;

public BayesClassifier()

{

SpamHamReader reader = new SpamHamReader();

spamMails = reader.Read("spam");

notSpamMails = reader.Read("ham");

countSpamMails = spamMails.Count();

countNotSpamMails = notSpamMails.Count();

}

public Tuple<string, double> CheckEmail(string text)

{

Trainer trainer = new Trainer();

foreach (var spamMail in spamMails)

{

trainer.Train(spamMail);

}

var spamWords = trainer.WordCounter;

trainer = new Trainer();

foreach (var notSpamMail in notSpamMails)

{

trainer.Train(notSpamMail);

}

var notSpamWords = trainer.WordCounter;

return CheckIfSpam(text, countSpamMails, spamWords, countNotSpamMails, notSpamWords);

}

private Tuple<string, double> CheckIfSpam(string text,

int countSpamMails, Dictionary<string, int> spamWordList,

int countNotSpamMails, Dictionary<string, int> notSpamWordList)

{

string[] textOriginal = text.Split(' ');

List<string> textWithoutNumbers = new List<string>();

foreach (var word in textOriginal)

{

var removedNumberWord = Regex.Replace(word, @"\d", "");

textWithoutNumbers.Add(removedNumberWord.ToLower());

}

List<string> wordsList = new List<string>();

foreach(var word in textWithoutNumbers)

{

var removedPunctuationWord = Regex.Replace(word, @"[^\w\s]", "");

wordsList.Add(removedPunctuationWord.ToLower());

}

var wordsToRemove = new CommonWords();

wordsList.RemoveAll(x => wordsToRemove.commonWords.Contains(x));

wordsList.RemoveAll(x => x == String.Empty);

wordsList = wordsList.Distinct().ToList();

List<double> PvaluesSpam = new List<double>();

List<double> PvaluesHam = new List<double>();

foreach (var word in wordsList)

{

if(notSpamWordList.ContainsKey(word) && spamWordList.ContainsKey(word))

{

var p = CalculateProbability(word.ToLower(),

countSpamMails, spamWordList,

countNotSpamMails, notSpamWordList);

PvaluesSpam.Add(p);//P(S|W)

PvaluesHam.Add(1 - p);//P(H|W)

}

}

//konacni izracun

double probabilityOfSpamMail;

if (PvaluesSpam.Count == 0 || PvaluesHam.Count == 0)

probabilityOfSpamMail = 0.5;// nepoznate su sve riječi emaila

else

{

double PSpam = PvaluesSpam.Aggregate((a, x) => a \* x); ;//NB teorem

double PHam = PvaluesHam.Aggregate((a, x) => a \* x); //NB teorem

probabilityOfSpamMail = PSpam / (PSpam + PHam);

}

if (probabilityOfSpamMail > 0.5)

{

return Tuple.Create("SPAM", probabilityOfSpamMail);

}

else

{

return Tuple.Create("HAM", probabilityOfSpamMail);

}

}

private double CalculateProbability(string word,

int countSpamMails, Dictionary<string, int> spamWordList,

int countNotSpamMails, Dictionary<string, int> notSpamWordList)

{

double ProbMailIsSpam = 0.5;//P(S) //konstanta

double ProbMailIsNotSpam = 0.5;//P(H) //konstanta

//vjerojatnost da riječ pripada spam mailu

double wordCountInSpam = spamWordList[word];

double ProbWordInSpam = wordCountInSpam / countSpamMails; //P(W|S);

// vjerojatnost da riječ pripada ne spam mailu

double wordCountInNotSpam = notSpamWordList[word];

double ProbWordInNotSpam = wordCountInNotSpam / countNotSpamMails; //P(W|H)

return (double)(ProbWordInSpam \* ProbMailIsSpam) / (double)((ProbWordInSpam \* ProbMailIsSpam) + (ProbWordInNotSpam \* ProbMailIsNotSpam));//P(S|W)

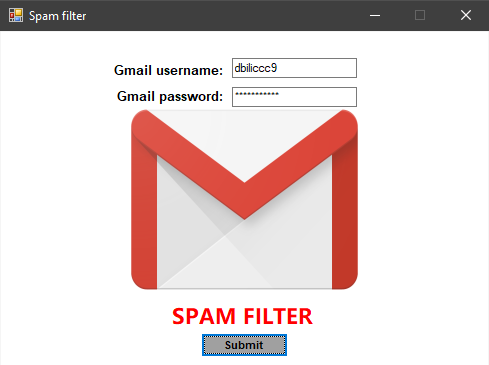
}

}

Klasa *BayesClassifier* sastoji se od nekoliko funkcija koje su međusobno povezane. No, prije njih postoji istoimeni konstruktor ove klase koji omogućava da se bilo kad kod pozivanja te klase sprovede čitanje ham i spam mailova. To je važno jer prva funkcija ove klase koja se naziva *CheckEmail* sprovodi treniranje nad prethodno pročitanim mailovima. Ova funkcija poziva nadalje funkciju *CheckIfSpam* s parametrom *text* (tekst novog maila) i preostalim parametrima dobiveni preko klase *Trainer*. Proslijeđeni parametar *text* se obrađuje kako bi se iz njega maknule sve brojke, ponavljajuće riječi i tako dalje. Nakon što se *text* pročisti dobivaju se temelji za izračun vjerojatnosti da li je mail temeljem tog teksta spam ili ham. Valja spomenuti da se vjerojatnost računa samo za one riječi koje su prošle kroz trening fazu jer inače ne postoje temelji za donošenje zaključaka o nekoj riječi. Dakle, za te određene riječi se upotrebljava funkcija *CalculateProbability* kojaomogućava izračun vjerojatnosti da se radi o spam mailu temeljem te riječi prema Bayesovom teoremu, odnosno izračunava se gdje predstavlja broj riječi teksta novog maila. Izračunate vrijednosti spremaju se u listu *PvaluesSpam* (sadrži sve vjerojatnosti da je mail spam temeljem pojedine riječi) i u listu *PvaluesHam* (sadrži sve vjerojatnosti da je mail ham temeljem pojedine riječi – vrijedi ). Nakon što se popune te dvije liste za samo jedan mail, može se krenuti u konačni izračun vjerojatnosti prema napisanoj formuli Naivnog Bayesovog klasifikatora. Dakle, vjerojatnost da je mail spam prema formuli dobiva se množenjem svih vjerojatnosti u listi *PvaluesSpam,* a rezultat je zapisan u varijabli *PSpam*. S druge strane, vjerojatnost da je mail ham prema formuli dobiva se množenjem svih vjerojatnosti u listi *PvaluesHam.* Ono što valja spomenuti da se kod tih izračunavanja izbacuju vrijednosti i jer su one konstante, tj. imaju vrijednost 0.5 pošto novi mail se ne može klasificirati u startu kao spam ili ham. Nakon izračuna *PvaluesSpam* i *PvaluesHam,* ovi brojevi mogu se pretvoriti u vjerojatnost tako da zbroj bude jednak 1 (preko formule normalizacije) kojom se dobiva vrijednost varijable *probabilityOfSpamMail.* Ukoliko je ta vrijednost veća od 0.5 tada se mail klasificira kao SPAM, dok u suprotnom kao HAM.

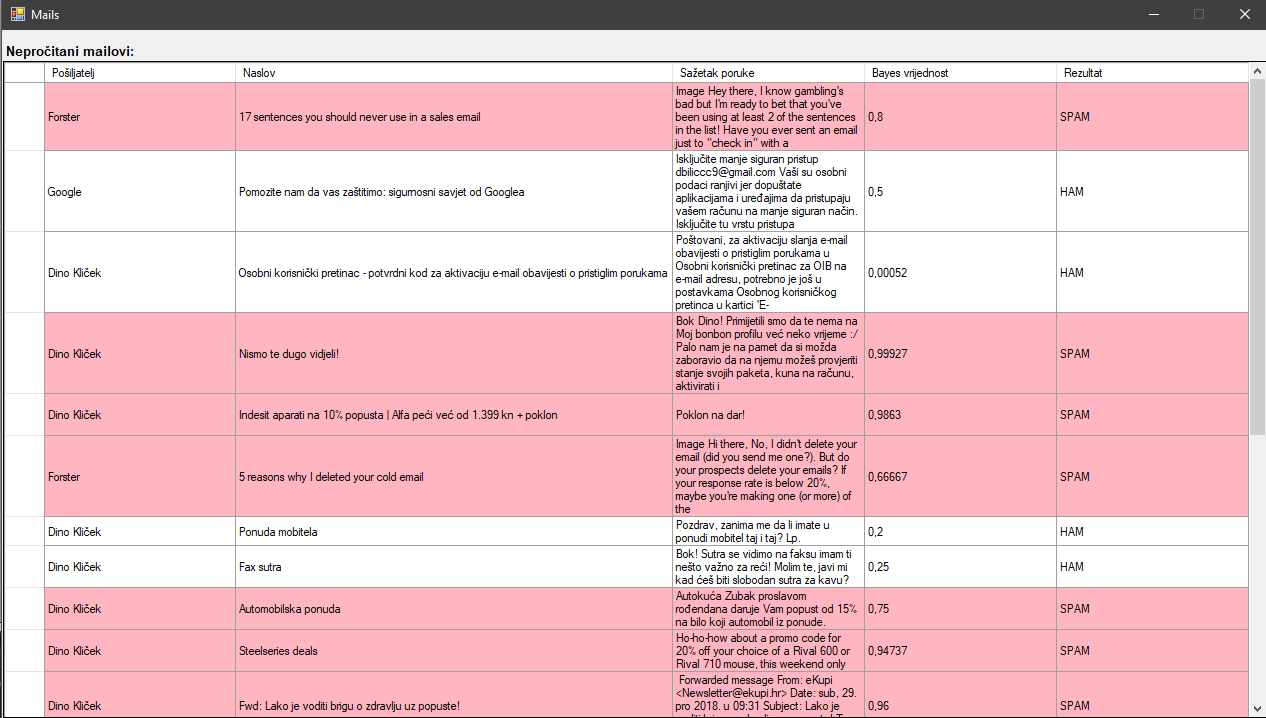
1. Testiranje funkcionalnosti

U ovom poglavlju biti će prikazan primjer korištenja programa za klasifikaciju mailova na spam ili ham. Za potrebe testiranja programa sami smo kreirali i slali mailove s ciljem promatranja ponašanja samog programa kod različitih mailova. Kako bi se program testirao prvo je potrebno unijeti podatke korisničkog Gmail računa, odnoso *username* i *password*. Sučelje za taj unos prikazano je na slijedećoj slici.



Slika 2. Početni zaslon programa

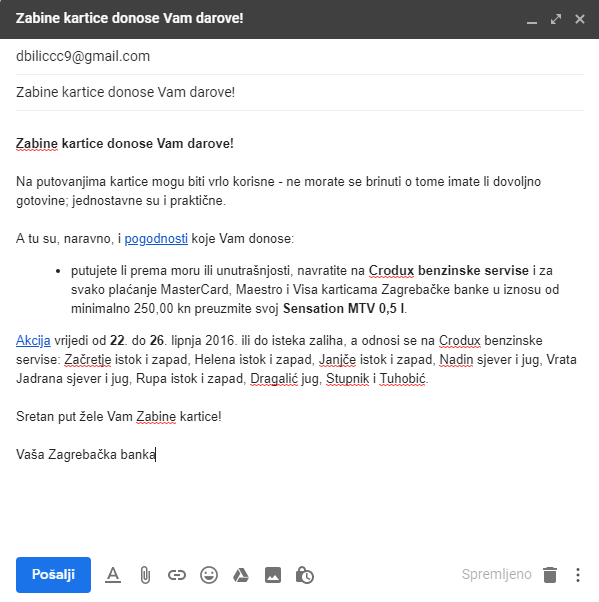
Nakon unosa podataka korisnik klikom na gumb Submit dobiva rezultate klasifikacije vlastitih, novo pristiglih mailova iz poštanskog sandučića. Dakle, pritiskom na gumb *Submit* program radi sve aktivnosti navedene na Slici 1. Prikaz rezultata klasifikacije može se vidjeti na Slici 3.



Slika 3. Prikaz rezultata klasifikacije mailova korisnika

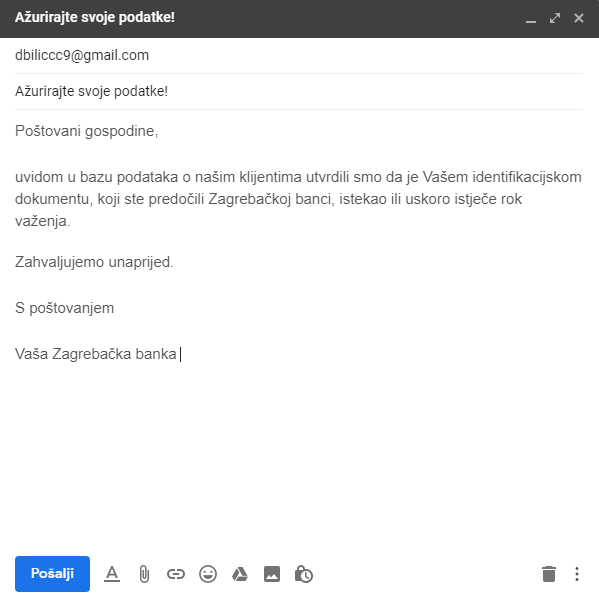
Slika 3. prikazuje rezultat klasifikacije preuzetih korisnikovih mailova s Gmail računa. Za svaki mail zna se pošiljatelj, naslov maila kao i njegov sadržaj. Nadalje, stupac Bayes vrijednost dobivena je primjenom formule Naivnog Bayesovog teorema i temeljem njega se klasificiraju mailovi na spam ili ham. Mogu se vidjeti različite izračunate vrijednosti, od onih koji su blizu 1 do onih koji su blizu 0. Ono što je bitno kod takvog programa su zapravo podaci na kojima se program trenira jer temeljem njih izračunava Bayesove vrijednosti. Dakle, što je više riječi u ham i spam mailovima to će bolje, odnosno preciznije klasificirati mail. Pošto je takav klasifikator više jezični, kod testiranja smo upotrijebili i engleski jezik, odnosno mailove.

U nastavku ćemo napraviti dva nova maila i poslati ovom korisniku da vidimo kako će klasifikator reagirati na njih. Oba mail vezana su uz banku koja svojim klijentima šalje i promotivne mailove, ali ponekad i mailove koji su ozbiljnog karaktera. Prvi mail je prikazan na slijedećoj slici.



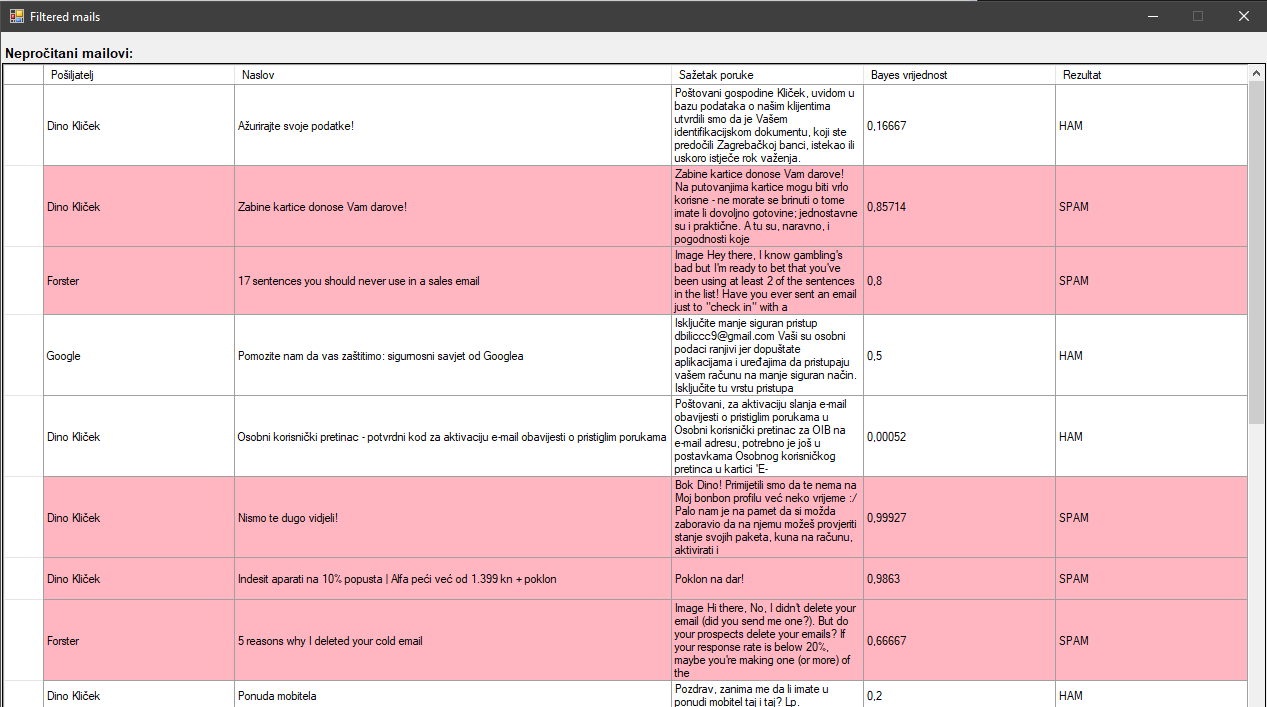
Slika 4. Spam mail banke

Na Slici 4. može se vidjeti jedan tipičan spam mail koji ima nekoliko važnih riječi sadržanih u sebi kao što su „darove“, „pogodnosti“ i tako dalje. S druge strane, napravljena je kopija maila koji banka šalje svojim klijentima kako bi ažurirali svoje podatke – radi se o ham mailu prikazanom na Slici 5.



Slika 5. Ham mail banke

Slika 5. prikazuje mail sa sadržajem ozbiljnog karaktera koji program mora klasificirati kao ham mail. Slanjem tih mailova i ponovnim pokretanjem programa dobivamo slijedeći rezultat klasifikacije.



Slika 6. Rezultat klasifikacije prethodno poslanih mailova

Prema Slici 6. može se vidjeti da je program ispravno klasificirao poslane mailove. Razlog zašto su tako klasificirani leži u nekoliko riječi koji su bile spomenute kod pisanja spam maila. Riječi poput „darove“ i „pogodnosti“ su riječi koje se spominju jako često u spam mailovima, a program je to kroz fazu treninga nad unesenim spam mailovima i naučio te sada pravilno klasificirao. S druge strane, kod ham maila postoji vjerojatnost od skoro 17% da je to spam mail. No, ponovno, naučeni Bayesov klasifikator raspoznaje da taj mail sadrži riječi karakteristične za ham mailove.

1. Zaključak

U ovom projektnom radu opisane su mailovi i njihova rastuća popularnost i upotreba. Samim time dolazi do potrebe uklanjanja takvim mailova jer nepotrebno troše resurse korisnika. Stoga se ovaj rad usmjerio na izradu jednog takvog programskog alata koji omogućava klasifikaciju mailova korisnika na spam ili ham mailove. Kod procesa klasifikacije uvelike je pomogla statistička tehnika temeljena na Bayesovom teoremu, takozvana Naivni Bayesov teorem. Njegovom implementacijom u programu omogućena je laka gore spomenuta klasifikacija koja vidi kroz poglavlje testiranja funkcionalnosti.

Zaključna misao ovog rada veže se na promatranje korisnosti kod upotrebe ovakvih vrsta programe takvih programa. Svakako, upotreba klasifikacije mailova donosi mnoge koristi. Najveća korist je suzbijanje mailova koji sadrže viruse gdje bi se oni lako klasificirali kao spam, ali ujedno preko programa se mogu promatrati bez rizika da će se računalo korisnika zaraziti. Dodavanjem funkcionalnosti spremanja mailova koji su spam, a koji ham zapravo se dobiva izuzetno kvalitetan program za klasifikaciju mailova i to ponajprije zbog stupnja točnosti same klasifikacije. No, takav način klasificiranja sadrži i neke uočene loše strane. Primjerice, ukoliko neki spam prihvatimo kao ham mail tada zapravo trujemo ovaj program. To je negativnost koju danas hakeri iskorištavaju u svrhu prevladavanja takvih spam filtera. Ukupni broj i kvaliteta literature vezane uz ovu temu zapravo pokazuju da je došlo do napretka koji se i dalje nastavljaju ostvarivati. Međutim, neki problemi u e-mail spamu filtriranje kao što je gore istaknuto i dalje ostaje. Sve dok se ne jave znatnija poboljšanja filtriranja neželjene pošte ili neki drugi načini, anti-spam istraživanja će ostati i dalje aktivno područje istraživanja.

Popis literature

Bhowmick, A., & Hazarika, S. M. (2016). Machine Learning for E-mail Spam Filtering: Review, Techniques and Trends. Dostupno na https://goo.gl/DykcgL

Jensi, R. (2008). A new approach to spam mail detection. Dostupno na https://goo.gl/TDr9dJ

Martins, J.-M., & Cormack, G. V. (2009). Using old Spam and Ham Samples to Train Email Filters. Dostupno na https://goo.gl/wrs4AD

Tretyakov, K. (2004). Machine Learning Techniques in Spam Filtering, Institute of Computer Science, University of Tartu. Dostupno na https://goo.gl/x3dP5d

Metsis, V., Androutsopoulos, I. & Paliouras G. (2006). Spam Filtering with Naive Bayes – Which Naive Bayes? CEAS 2006 Third Conference on Email and AntiSpam, Mountain View, California USA. Dostupno na <https://goo.gl/SJdt1D>

Popis slika

[Slika 1. Tijek procesa korisničkog korištenja programa i izvođenja programa 8](#_Toc535152249)

[Slika 2. Početni zaslon programa 16](#_Toc535152250)

[Slika 3. Prikaz rezultata klasifikacije mailova korisnika 17](#_Toc535152251)

[Slika 4. Spam mail banke 18](#_Toc535152252)

[Slika 5. Ham mail banke 18](#_Toc535152253)

[Slika 6. Rezultat klasifikacije prethodno poslanih mailova 19](#_Toc535152254)