

Fakultet elektrotehnike, računarstva i informacijskih tehnologija Osijek

Računarstvo usluga i analiza podataka

Klasifikacija spam SMS poruka

Projektni rad

Mateo Dubinjak

Ivan Ivković

Osijek, 2020.

SADRŽAJ

1. UVOD	1
2. OPIS PROBLEMA.....	2
2.1. Korišteni podatci	3
2.2. Korišteni postupci strojnog učenja	4
2.3. Usporedba rezultata.....	5
3. OPIS PROGRAMSKOG RJEŠENJA	9
3.1. Model strojnog učenja	9
3.2. Način korištenja API-ja	14
3.3. Klijentska aplikacija	16
4. ZAKLJUČAK	17
POVEZNICE I LITERATURA	18

1. UVOD

U ovom projektnom zadatku razrađuje se problem klasifikacije spam SMS poruka pomoću modela strojnog učenja. Za dostupni podatkovni skup, kojeg je prvo potrebno proučiti i predstaviti, treba provjeriti prikladnost nekih od metoda izdvajanja značajki, opisati ih te iskoristiti za dobivanje podatkovnog skupa na kojem se potom izgrađuje klasifikacijski model. Kao priprema podatkovnog skupa prikazuje se njegova osnovna deskriptivna statistika i zatim se on koristi za vrednovanje nekoliko različitih tipova klasifikatora, čije se performanse potom analiziraju na problemu koji se razmatra u projektnom radu. Dobivene rezultate potrebno je analizirati te odabrati model na čijem će se temelju izraditi aplikacija i u konačnici izložiti kao web usluga na platformi Azure.

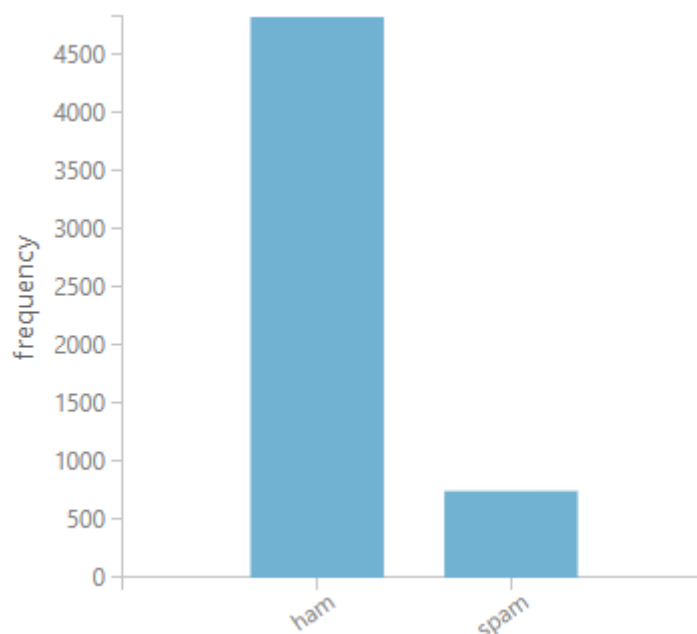
2. OPIS PROBLEMA

Intenzivan razvoj znanosti i tehnologije, u prvom redu informacijskih i komunikacijskih tehnologija, kroz koje svijet prolazi zadnjih nekoliko desetljeća, doveo je do globalizacije. Pojam je to široke uporabe za kojeg, zbog njegove velike složenosti, ne postoji jednoznačna definicija. Ipak, najčešće se, prema [1], govori o gospodarskim, društvenim, političkim i kulturnim procesima koji vode prema preobrazbi životnih uvjeta te sve većoj povezanosti i međuovisnosti pojedinih dijelova svijeta. Svaki od spomenutih segmenata globalizacije ima svoje prednosti i nedostatke. U komunikacijskom smislu, globalizacija je ljudima iz različitih dijelova svijeta omogućila povezanost te razmjenu mišljenja i stavova, a razlog tomu je širok spektar načina razmjenjivanja poruka uz jednostavnost i nisku cijenu korištenja. Te činjenice dovele su do masovnosti uporabe elektroničke komunikacije, odnosno telekomunikacije, što je, pored brojnih prednosti, neminovno donijelo i određene nedostatke. Neželjene poruke, odnosno *spam*, predstavljaju veliki problem. Radi se o porukama najčešće komercijalnog sadržaja, nerijetko vezanima za prijevare, a koje se masovno šalju slučajno odabranim primateljima [2]. Pritom nepoznati pošiljatelj nudi svoje proizvode ili usluge korisnicima, tj. primateljima, koji uopće nisu pokazali zanimanje za reklamirani proizvod ili uslugu, te šalje veliki broj poruka primateljima zatrpavajući njihove „sandučice“, što otežava razlikovanje pravih i neželjenih poruka i samim time uzrokuje ozbiljne probleme u komunikaciji. Osim neželjenih reklama, *spam* porukama smatraju se i obavijesti o temama na koje se primatelji nisu pretplatili, lažne privatne poruke koje nerijetko vode do pornografskih sadržaja i mnogi drugi načini. *Spam* se najčešće šalje putem e-pošte i uglavnom se odnosi na e-poštu, no česte su i zlouporabe drugih medija, uključujući SMS poruke. Premda je u mnogim državama kažnjivo, slanje neželjenih poruka u različitim oblicima i dalje je vrlo rašireno. S obzirom da ono radi velike probleme službama koje održavaju poslužitelje elektroničke pošte, pružatelji usluga elektroničke pošte neprestano rade na suzbijanju *spama*. Pomoću raznih algoritama za analizu sadržaja poruke omogućuje se filtriranje poruka, odnosno prepoznavanje radi li se o željenoj ili neželjenoj poruci. Takva rješenja uglavnom nisu dugog vijeka s obzirom da im se pošiljatelji neželjenih poruka uspješno prilagođavaju te ih prije ili kasnije uspiju zavarati. Iz tog razloga, kasnije se počelo raditi na tzv. SAV zaštiti (*Sender Address Verification*), koja je zasnovana na verifikaciji adrese pošiljatelja, a koja se provodi interakcijom s pošiljateljevim poslužiteljem elektroničke pošte. Pritom se određena poruka isporučuje primatelju samo ako je uspješno prošla verifikaciju [3]. Što se tiče *spam* SMS poruka, suočavanje s istima se, zbog njihove relativne malobrojnosti u odnosu na *spam* elektroničku poštu, ignoriranje od mnogih korisnika i pružatelja usluga te ograničene dostupnosti softvera za filtriranje *spam* poruka na

mobilnim uređajima, dobrim dijelom svodi tek na prijavljivanje neželjenih SMS poruka, koje na određene načine pojedini pružatelji telekomunikacijskih usluga omogućuju svojim korisnicima. U Sjedinjenim Američkim Državama potom je moguće pisati žalbe, a u Ujedinjenom Kraljevstvu i podizati tužbe protiv pošiljatelja *spam* SMS poruka, (samo) onih s područja države. U lipnju 2009. u Kini su tamošnje glavne telekomunikacijske tvrtke, radi borbe protiv *spam* SMS poruka, uvele ograničenje po kojem se sa svakog broja može poslati najviše 200 poruka u jednom satu i najviše 1.000 poruka u jednom radnom danu [4]. Najjednostavnije i najkraće rečeno, *spam* je svaka poruka koja se šalje primatelju bez njegovog dopuštenja, odnosno svaka masovno distribuirana poruka koju korisnici nisu zatražili. Nasuprot *spamu*, *ham* označava prave poruke, odnosno poruke koje nisu neželjene.

2.1. Korišteni podatci

Korišteni podatci preuzeti su iz skupa podataka [Kaggle](#), koji sadrži 5.572 SMS poruke na engleskom jeziku, pri čemu je 4.825 poruka (86,59% ukupnog broja poruka) označeno kao *ham*, a njih 747 (13,41% ukupnog broja poruka) kao *spam*. Navedeni omjer između broja *ham* poruka i broja *spam* poruka u promatranom podatkovnom skupu vizualiziran je grafičkim prikazom na slici 2.1.



Sl. 2.1. Omjer između broja *ham* poruka i broja *spam* poruka u podatkovnom skupu

Detaljnim proučavanjem samog podatkovnog skupa utvrđuju se neke značajke *spam* poruka, odnosno razlike u odnosu na *ham* poruke. Prva indikacija je duljina poruke, odnosno broj znakova koje ta poruka sadrži. U promatranom skupu prosječna duljina *spam* poruka je približno 138 znakova, dok, s druge strane, *ham* poruke u prosjeku sadrže gotovo upola manje znakova, njih približno 71. Razlog velikom broju znakova u *spam* porukama je činjenica da je naknada za slanje jedne poruke, čija je duljina ograničena na najviše 160 znakova, jednaka bez obzira na broj znakova, što pošiljatelji *spam* SMS poruka iskorištavaju nastojeći upotrijebiti što veći broj znakova, po mogućnosti što bliže predviđenoj gornjoj granici [5]. Druga značajka *spam* poruka, donekle povezana s brojem znakova u poruci, veći je broj riječi u odnosu na *ham* poruke. Tako *spam* poruke u prosjeku imaju 23 riječi, a *ham* poruke 14 riječi. Mogu biti važne i same riječi koje se pojavljuju. Već je spomenuto da su *spam* poruke pretežno komercijalnog sadržaja i nerijetko vezane za prijevare. Sukladno tomu, neke od riječi koje se u promatranom skupu podataka najčešće pojavljuju u *spam* porukama su: *call, free, mobile, claim, stop, reply, prize, get, new, send, nokia, urgent, cash, win, contact, service, please, guaranteed, customer, week, tone, phone* i druge. *Ham* poruke sadrže najčešće riječi poput nesuvislih *u, ur, n* te razumljivih *I'm, get, OK, don't, go, know, got, like, call, come, good, time, day, love, going, want, one, home, need, sorry, still, see* itd. Nesuvislost pojedinih riječi u *ham* porukama proizlazi iz činjenice da se u takvim porukama uglavnom radi o jeziku iz svakodnevne komunikacije, u kojem se ne pridaje velika pozornost gramatičkoj i pravopisnoj ispravnosti, pa pri tome nije neobična uporaba svojevrskih „kratica“ (*u* umjesto *you*, *ur* umjesto *you are*, *n* umjesto *and...*), izostanak ili nepravilna uporaba interpunkcijskih znakova te druge pogreške. Ipak, i kod *spam* poruka zabilježena je, primjerice, česta uporaba „kratica“ poput *txt, u* i *ur*, no u manjoj mjeri nego što je to slučaj kod *ham* poruka, pa se u cjelini može zaključiti da *spam* poruke imaju manje gramatičkih i pravopisnih pogrešaka.

2.2. Korišteni postupci strojnog učenja

Strojno učenje područje je računalne znanosti koje se bavi oblikovanjem algoritama koji svoju učinkovitost poboljšavaju automatski, temeljem iskustva [6]. Takvi algoritmi pritom izgrađuju model iz primjera radi provođenja predikcija ili donošenja odluka na temelju empirijskih podataka, a bez striktnog izvođenja statičkih programskih instrukcija. Tri su osnovna oblika strojnog učenja: nadzirano učenje, nenadzirano učenje i podržano učenje. Nadzirano učenje je vrsta strojnog učenja čiji je cilj odrediti nepoznatu funkcionalnu ovisnost između ulaznih veličina i izlazne veličine temeljem podatkovnih primjera. Dvije su osnovne namjene modela dobivenog nadziranim

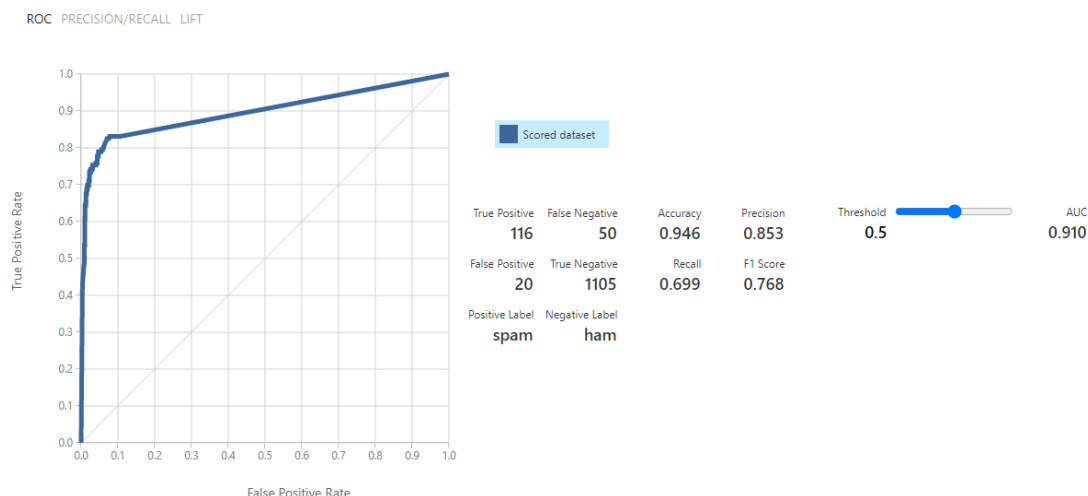
učenjem: predikcija i zaključivanje. Cilj predikcije je, temeljem modela i ulaznih veličina, procijeniti vrijednost izlazne veličine. S druge strane, zaključivanjem se, temeljem modela, pokušava saznati više o postupku generiranja podataka. Razredi problema nadziranog učenja su regresija i klasifikacija. Zadatak regresije je predvidjeti stvarnu vrijednost za svaku stavku, pa njena primjena uključuje primjerice procjenu cijena kuća i automobila, procjenu temperature itd. Klasifikacija ima za cilj dodijeliti kategoriju svakoj stavci te se, kao takva, koristi upravo za prepoznavanje *spam* poruka, ali i u medicinskoj dijagnostici, u otkrivanju tumora, te u prepoznavanju izgovorenih riječi, prepoznavanju lica, prepoznavanju brojeva pisanih rukom itd. Nenadzirano učenje je drugi oblik strojnog učenja, u čijim su problemima na raspolaganju samo podatci o ulaznim veličinama jer izlazna veličina ne postoji, a razredi problema su: grupiranje podataka (podjela stavki u homogene skupine), smanjivanje dimenzionalnosti (transformiranje inicijalnog prikaza stavki u prikaz manjih dimenzija uz očuvanje svojstava) i detekcija nepravilnosti u podacima. Treći oblik strojnog učenja je podržano učenje, koje omogućuje agentu samostalno otkrivanje optimalnog ponašanja metodom pokušaja i pogreški. Pritom agent za svaku radnju koju izvodi dobiva povratnu informaciju je li ona bila dobra ili loša. Kao što je već spomenuto, problem prepoznavanja *spam* poruka pripada nadziranom učenju, točnije klasifikaciji, koja može biti binarna ili višeklasna. Razlika je u broju kategorija koje mogu biti dodijeljene stavkama, pa tako kod binarne klasifikacije postoje samo dvije kategorije, a kod višeklasne klasifikacije više od dvije kategorije. S obzirom da je kod problema prepoznavanja *spam* poruka potrebno utvrditi tek je li neka poruka *spam* ili *ham* (nije *spam*), dovoljno je pritom koristiti binarnu klasifikaciju.

2.3. Usporedba rezultata

Binarni klasifikatori koji su razmatrani u ovom radu su: *Two-Class Neural Network*, *Two-Class Decision Forest*, *Two-Class Decision Jungle*, *Two-Class Logistic Regression* i *Two-Class Support Vector Machine*.

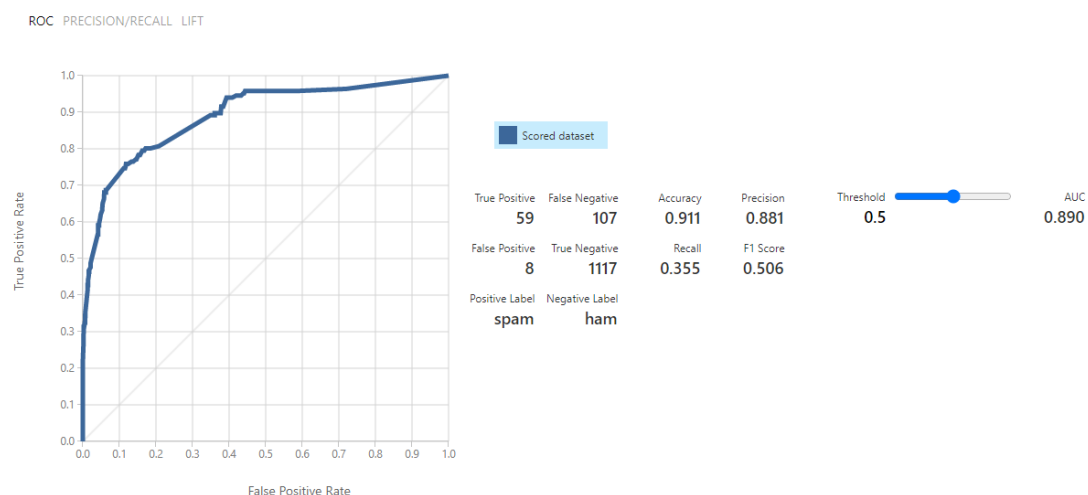
Two-Class Neural Network binarni je klasifikator koji koristi algoritam neuronskih mreža. Umjetne neuronske mreže su računalni modeli obrade informacija čiji je koncept zasnovan na promatranju ljudskog mozga te su, kao takve, jedan od najčešće korištenih modela strojnog učenja. Sastoje se od neurona te veza među slojevima neurona koje imaju pripadajuće težine. U pravilu se sastoje od triju slojeva: ulaznog, skrivenog i izlaznog sloja, pri čemu mogu sadržavati i više od

jednog skrivenog sloja. Na slici 2.2. prikazani su rezultati testiranja klasifikatora *Two-Class Neural Network*.



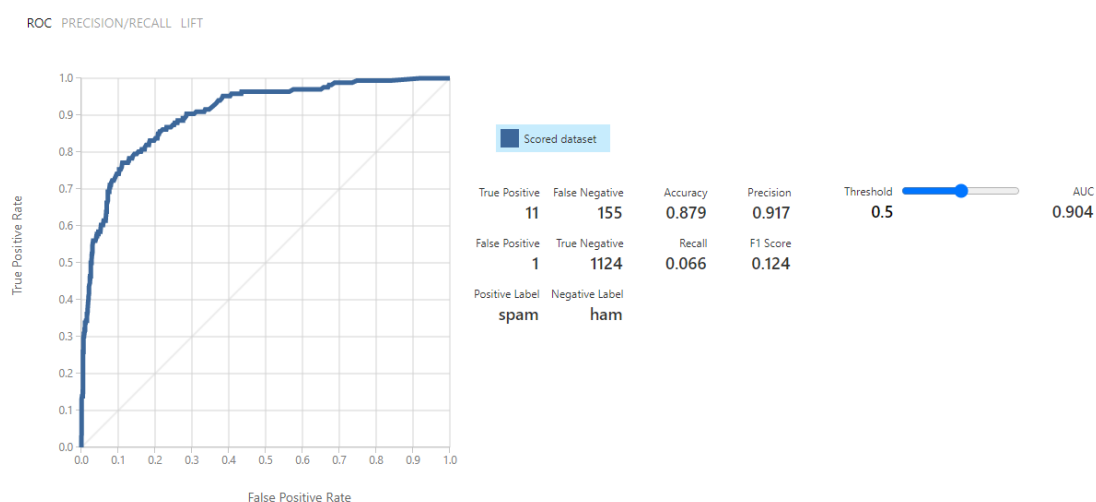
Sl. 2.2. Rezultati testiranja klasifikatora *Two-Class Neural Network*

Two-Class Decision Forest klasifikator je koji koristi algoritam nasumičnih šuma. Šume odlučivanja jedan su od algoritama strojnog učenja, zasnovane na stablima odlučivanja, koja su uglavnom nadzirano učenje. Kod nasumičnih šuma koristi se, dakle, više stabala odlučivanja, pri čemu je svako stablo klasifikator, a odluke se na kraju objedinjuju glasanjem, slaganjem ili nekim drugim načinima. Ta cjelina koju čini više stabala odlučivanja u pravilu ima bolje performanse u odnosu na pojedinačna stabla. Na slici 2.3. prikazani su rezultati testiranja klasifikatora *Two-Class Decision Forest*.



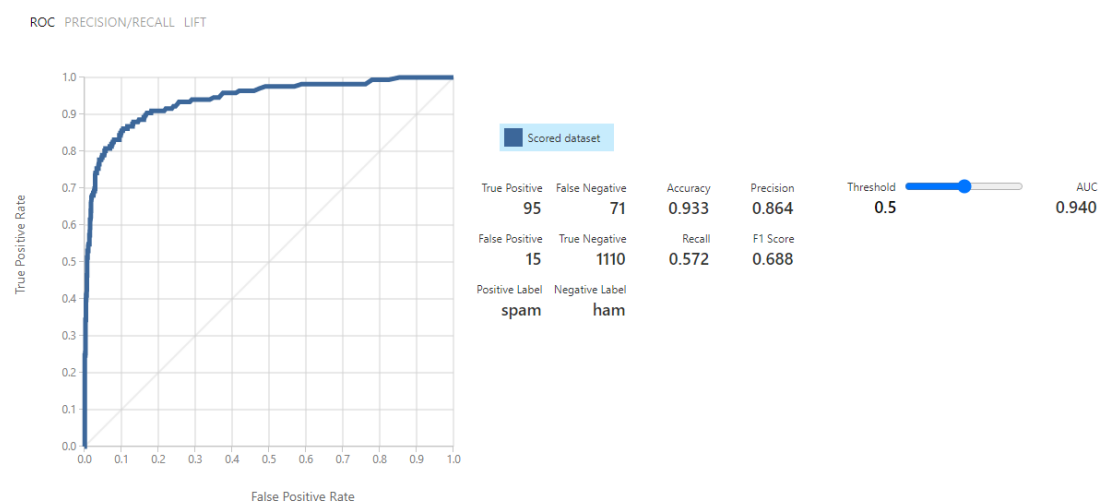
Sl. 2.3. Rezultati testiranja klasifikatora *Two-Class Decision Forest*

Džungle odlučivanja su algoritam strojnog učenja također zasnovan na stablima odlučivanja. Sukladno tomu, klasifikator *Two-Class Decision Jungle* sličan je klasifikatoru *Two-Class Decision Forest*, čiju nadogradnju zapravo predstavlja i stoga su njegove značajke bolja optimizacija koda te zauzimanje manje memorije, koja u velikim sustavima može biti ograničena. Jedini nedostatak u odnosu na *Two-Class Decision Forest* je dulje vrijeme treniranja. Na slici 2.4. prikazani su rezultati testiranja klasifikatora *Two-Class Decision Jungle*.



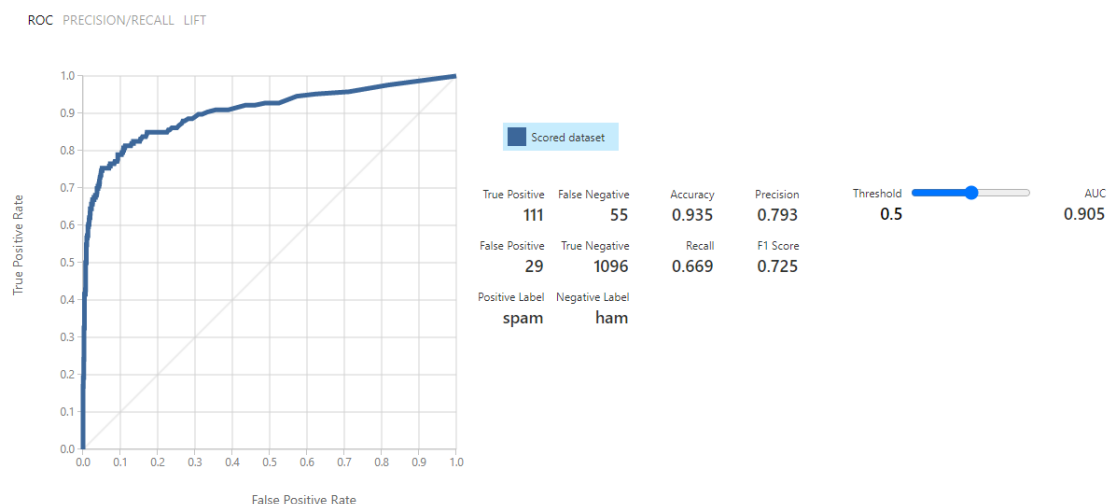
Sl. 2.4. Rezultati testiranja klasifikatora *Two-Class Decision Jungle*

Two-Class Logistic Regression četvrti je razmatrani klasifikator, vezan za algoritam logističke regresije, koja je brz i jednostavan alat za rješavanje dvorazrednih i višerazrednih problema. Koristi se za predviđanje vjerojatnosti događaja na način da se podatci prilagođavaju logističkoj krivulji, prepoznatljivoj po svom S-obliku. Na slici 2.5. prikazani su rezultati testiranja klasifikatora *Two-Class Logistic Regression*.



Sl. 2.5. Rezultati testiranja klasifikatora *Two-Class Logistic Regression*

Two-Class Support Vector Machine binarni je algoritam klasificiranja koji pronalazi hiper-ravnine s najvećim marginama razdvajanja klasa, pri čemu margine predstavljaju udaljenosti između točaka najbližih plohi razdvajanja, a te točke najbliže plohi razdvajanja nazivaju se vektorima podrške. Na slici 2.6. prikazani su rezultati testiranja klasifikatora *Two-Class Support Vector Machine*.



Sl. 2.6. Rezultati testiranja klasifikatora *Two-Class Support Vector Machine*

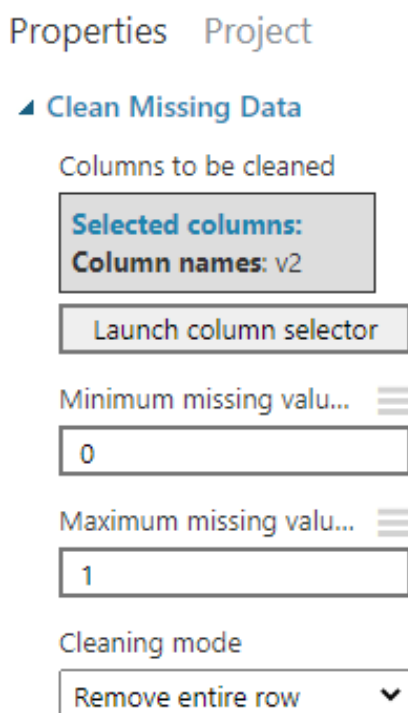
Svi navedeni klasifikatori testirani su više puta, svaki put s drugim parametrima kako bi se dobila što bolja predodžba o performansama svakoga. Budući da je promatrani podatkovni skup neuravnotežen (na jednu *spam* poruku dolazi šest *ham* poruka), treba imati na umu da se pritom ne treba slijepo držati svojstva *accuracy* jer ono, naime, može prikrivati tu „pristranost“ modela. Prema [7], u ovakvim je slučajevima poželjno dobro razmotriti svojstvo *AUC* pri donošenju konačnog suda o performansama klasifikatora. Uzevši u obzir te činjenice i dobivene rezultate, u cjelini se najboljim klasifikatorom pokazao *Two-Class Logistic Regression* koji je, stoga, uzet kao temelj za izradu sučelja za prepoznavanje poruka.

3. OPIS PROGRAMSKOG RJEŠENJA

Temeljem odabranog modela strojnog učenja napravljena je aplikacija, odnosno sučelje koje korisniku omogućuje unos neke poruke, za koju se potom dobiva povratna informacija radi li se o *spam* ili *ham* poruci. Aplikacija je izrađena u razvojnom okruženju Microsoft Visual Studio uporabom programskog okvira .NET i programskog jezika C#.

3.1. Model strojnog učenja

Postupak izrade modela strojnog učenja počinje s učitavanjem podatkovnog skupa na čijem se temelju radi model. U ovom se slučaju radi o podatkovnom skupu koji je sadržan u CSV datoteci, pa se prvi modul naziva *spam.csv*. Tijekom proučavanja promatranog podatkovnog skupa, već pri samom njegovom početku primijećen je redak u kojem je navedena samo vrsta poruke (*v1*), dok je stavka predviđena za sadržaj poruke (*v2*) ostala prazna. Iz tog razloga korišten je modul *Clean Missing Data*, kojim su u potpunosti uklonjeni svi takvi retci, a postavke tog modula prikazane su slikom 3.1.








Sl. 3.1. Postavke modula *Clean Missing Data*

Također, primjetno je višestruko ponavljanje određenih redaka. Budući da to nije od pomoći pri treniranju modela strojnog učenja, uporabom modula *Remove Duplicate Rows*, u kojem je odabrana mogućnost *Retain first duplicate row*, uklonjeni su svi ponavljajući retci osim prvih pojavljivanja. Sljedeći korišteni modul je *Preprocess Text*, koji se, kako mu ime kaže, koristi za obradu, točnije pojednostavljivanje i čišćenje teksta, što je važno radi lakšeg izdvajanja značajki iz teksta, u ovom slučaju poruka. Stoga je u postavkama modula odabran stupac *v2* kao onaj u kojem će se raditi promjene, a među kojima je za izvršavanje odabrano sljedeće: *Normalize case to lowercase*, *Remove numbers* i *Remove special characters*. Rezultat izvršavanja modula *Preprocess Text* sa spomenutim postavkama prikazan je slikom 3.2.





RUAP-spam-ham > Preprocess Text > Results dataset

rows 5164 columns 6

v1	v2	Column 2	Column 3	Column 4	Preprocessed v2
					
ham	Go until jurong point, crazy.. Available only in bugis n great world la e buffet... Cine there got amore wat...				go until jurong point crazy available only in bugis n great world la e buffet cine there got amore wat
ham	Ok lar... Joking wif u oni...				ok lar joking wif u oni
spam	Free entry in 2 a wkly comp to win FA Cup final tkts 21st May 2005. Text FA to 87121 to receive entry question(std txt rate)T&C's apply				free entry in a wkly comp to win fa cup final tkts 21st may text fa to tc receive entry question std txt rate t c 's apply 08452810075over18 c...

Sl. 3.2. Rezultat izvršavanja modula *Preprocess Text*

Na slici se vide dva stupca sličnog sadržaja: *v2* i *Preprocessed v2*. Prvi od njih (*v2*) izvorni je stupac, onakav kakav je u podatkovnom skupu, dok je *Preprocessed v2* stupac nastao izvršavanjem modula *Preprocess Text*, što je vidljivo po izostanku točki, zareza, velikih slova itd. Također, vidljivi su i stupci *Column 2*, *Column 3* i *Column 4*, koji su posljedica neke pogreške u CSV datoteci i kao takvi su uglavnom suvišni. Korištenjem modula *Select Columns in Dataset*, u čijim se postavkama odabiru stupci *v1* i *Preprocessed v2*, uklanjaju se svi nepotrebni stupci, odnosno *v2*, *Column 2*, *Column 3* i *Column 4*. Slika 3.3. prikazuje rezultat izvršavanja tog modula.

rows	columns
5164	2
	v1 Preprocessed v2
view as	 
	 
ham	go until jurong point crazy available only in bugis n great world la e buffet cine there got amore wat
ham	ok lar joking wif u oni free entry in a wkly comp to win fa cup final tkts 21st may text fa to to receive entry question std txt rate t c 's apply 08452810075over18 s
spam	u dun say so early hor. u c already then say

Sl. 3.3. *Rezultat izvršavanja modula Select Columns in Dataset*

Postojeća imena, odnosno oznake stupaca, *v1* i *Preprocessed v2*, nisu osobito opisna. Iz tog razloga dodaje se modul *Edit Metadata*, u kojem se, kako je prikazano slikom 3.4., stupcima mijenjaju imena. Tako *v1* postaje *label*, a *Preprocessed v2* postaje *text*.

Properties Project

▲ Edit Metadata

Column

Selected columns:

Column names:

v1,Preprocessed v2

Launch column selector

Data type

Unchanged ▼

Categorical

Unchanged ▼

Fields

Unchanged ▼

New column names

label, text

Sl. 3.4. *Postavke modula Edit Metadata*

Podatkovni skup sada je očišćen, te je kao takav znatno pogodniji za treniranje modela. Ipak, prije samog treniranja potrebno je obaviti još pripreme. Uporabom modula *Feature Hashing* sadržaj, odnosno tekst svake poruke podatkovnog skupa pretvara se u skup značajki, predstavljenih kao cijeli broj. Slika 3.5. prikazuje postavke ovog modula. Odabire se stupac *text*, vrijednost *Hashing bitsize*, koja označava broj bitova koji se koriste pri stvaranju hash tablice, postavlja se na 8, a vrijednost *N-grams*, koja znači slijed od N riječi koje se promatraju kao jedinstvena cjelina, postavlja se na 2, što znači da pojedina cjelina sadrži 2 riječi.

Properties Project

▲ Feature Hashing

Target column(s)

Selected columns:
Column names: text

Launch column selector

Hashing bitsize

8

N-grams

2

Sl. 3.5. Postavke modula *Feature Hashing*

U strojnom se učenju podatkovni skup obično dijeli u nekom omjeru na skup za treniranje modela i skup za procjenu učinkovitosti. Na slici 3.6. vidi se da je za podjelu promatranog podatkovnog skupa korišten modul *Split Data* u kojem je određeno da se 75% nasumičnih podataka koristi za treniranje modela, što znači da preostalih 25% ostaje za procjenu učinkovitosti.

Properties Project

▲ Split Data

Splitting mode

Split Rows

Fraction of rows in the first output dataset

0.75

☒ Randomized split

Random seed

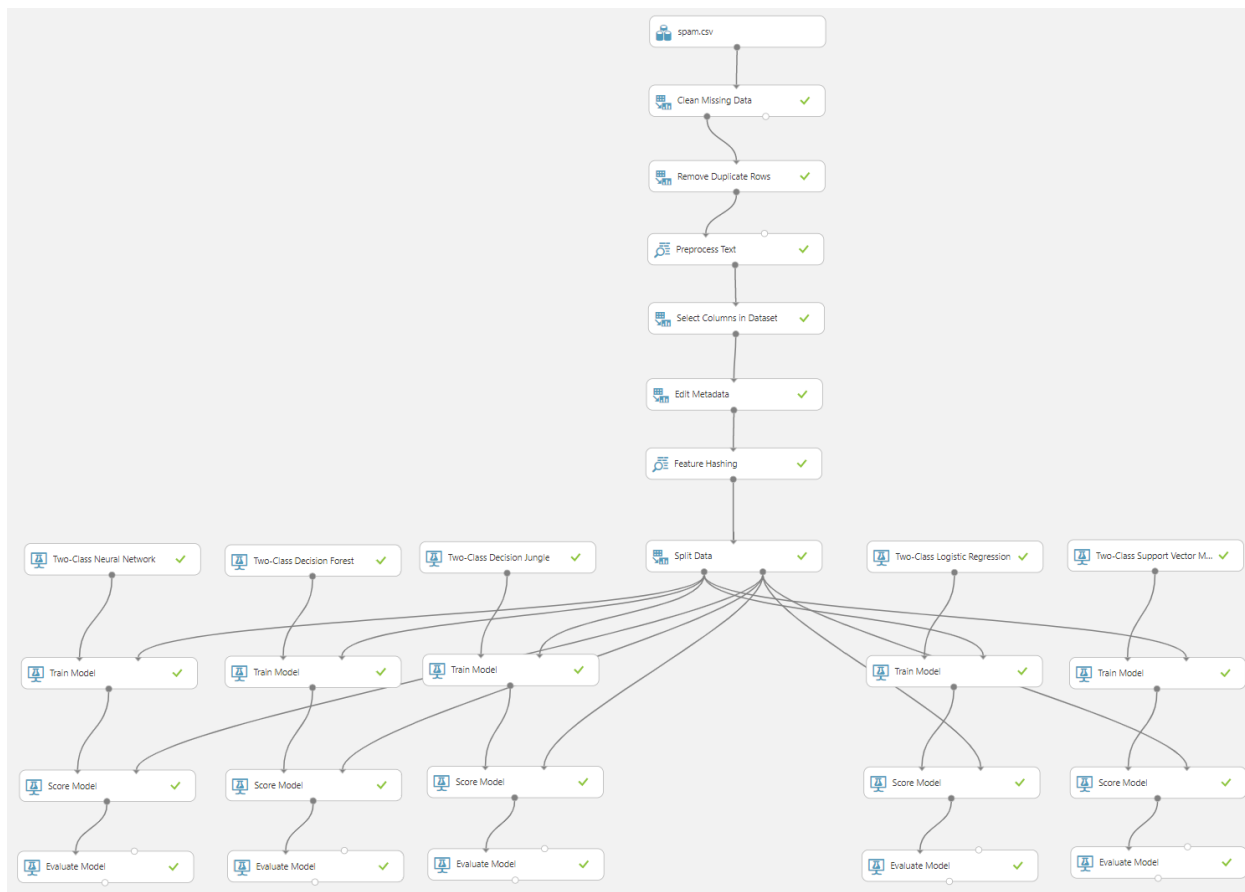
0

Stratified split

False

Sl. 3.6. Postavke modula *Split Data*

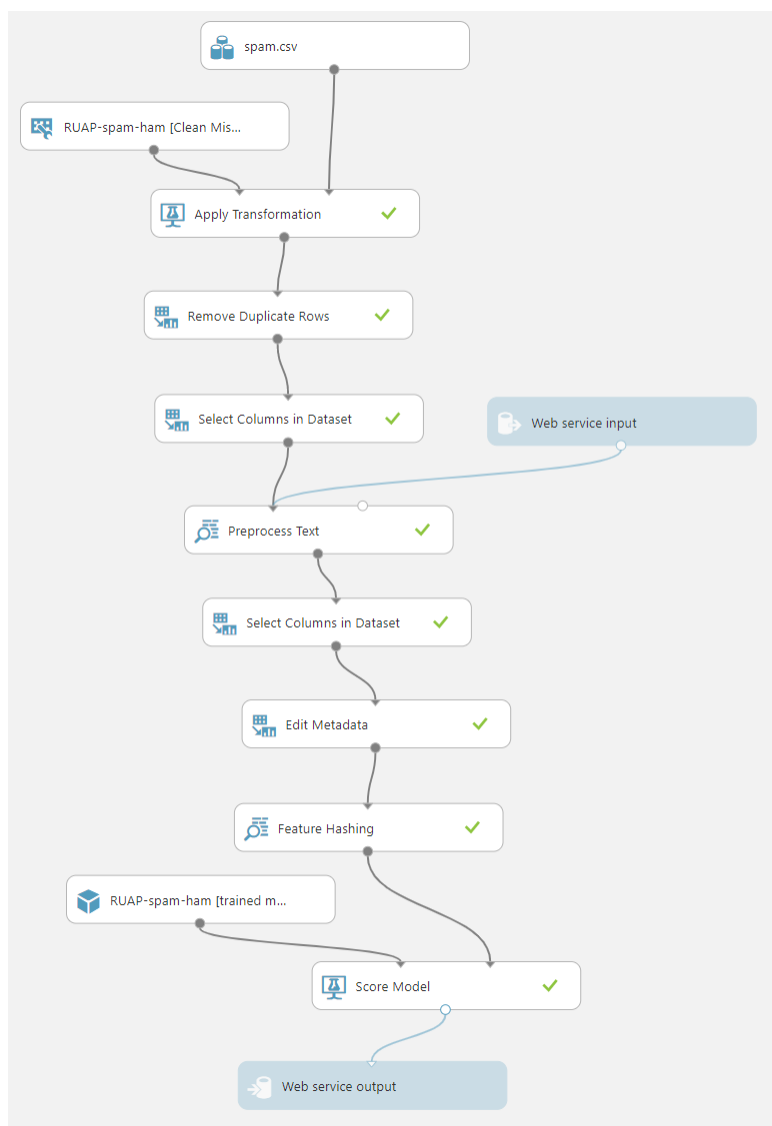
Sljedeći umetnuti moduli predstavljaju pet korištenih klasifikatora, navedenih i ukratko objašnjenih te analiziranih u potpoglavlju 2.3. Ispod svakog od njih dodan je po jedan modul za treniranje, *Train model*, u kojem je odabran stupac *label*, da bi na kraju uslijedili moduli *Score Model*, koji generira predikcije korištenjem istrenirane klasifikacije, te modul *Evaluate Model*, koji mjeri performanse klasifikatora. Cjelokupni model strojnog učenja nalazi se na slici 3.7.



Sl. 3.7. Model strojnog učenja

3.2. Način korištenja API-ja

Samo treniranje modela strojnog učenja nema osobitog smisla ako korisniku nije omogućena primjena istoga u vidu nekakvog sučelja. Azure ML omogućuje jednostavno stvaranje web servisa na temelju istreniranog modela: klikom na gumb *Set up web service*, koji se nalazi u donjoj alatnoj traci, odabire se *Predictive web service [Recommended]*, nakon čega se stvara prediktivni eksperiment, koji nakon određenih preinaka izgleda onako kako je prikazano slikom 3.8.



Sl. 3.8. Prediktivni eksperiment

Unutar prediktivnog eksperimenta potrebno je kliknuti gumb *Deploy web service*, kojim se stvara web servis. Pritom se generira API ključ, kojim se sprječava neovlašteni pristup servisu, a dobiva se i opis cjelokupnog API-ja, koji uključuje primjere koda (u C#, Pythonu i R-u) koji ga koriste.

API je važan jer omogućuje komunikaciju između web servisa i klijentske aplikacije, a komunikacija se odvija pomoću transportnog formata podataka JSON. Slikom 3.9. prikazan je dio dobivenog C# koda u kojem se, među ostalim, vidi uporaba API ključa te povezivanje web servisa i klijentske aplikacije, napravljene u C#.

```
static async Task InvokeRequestResponseService()
{
    using (var client = new HttpClient())
    {
        var scoreRequest = new
        {
            Inputs = new Dictionary<string, StringTable>() {
                {
                    "input1",
                    new StringTable()
                    {
                        ColumnNames = new string[] { "v2" },
                        Values = new string[,] { { "value" }, { "value" }, }
                    }
                },
            },
            GlobalParameters = new Dictionary<string, string>()
            {
            }
        };
        const string apiKey = "abc123"; // Replace this with the API key for the web service
        client.DefaultRequestHeaders.Authorization = new
        AuthenticationHeaderValue("Bearer", apiKey);

        client.BaseAddress = new
        Uri("https://ussouthcentral.services.azureml.net/workspaces/72be0458837440c49c931631169970b
        e/services/e195e015f24d4aefb37fae91bfa80993/execute?api-version=2.0&details=true");

        HttpResponseMessage response = await client.PostAsJsonAsync("", scoreRequest);

        if (response.IsSuccessStatusCode)
        {
            string result = await response.Content.ReadAsStringAsync();
            Console.WriteLine("Result: {0}", result);
        }
        else
        {
            Console.WriteLine(string.Format("The request failed with status code: {0}",
            response.StatusCode));

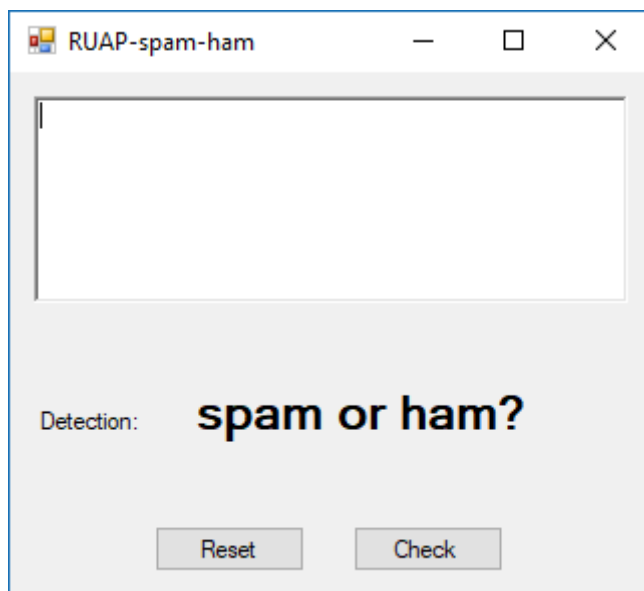
            Console.WriteLine(response.Headers.ToString());

            string responseContent = await response.Content.ReadAsStringAsync();
            Console.WriteLine(responseContent);
        }
    }
}
```

Sl. 3.9. C# kod

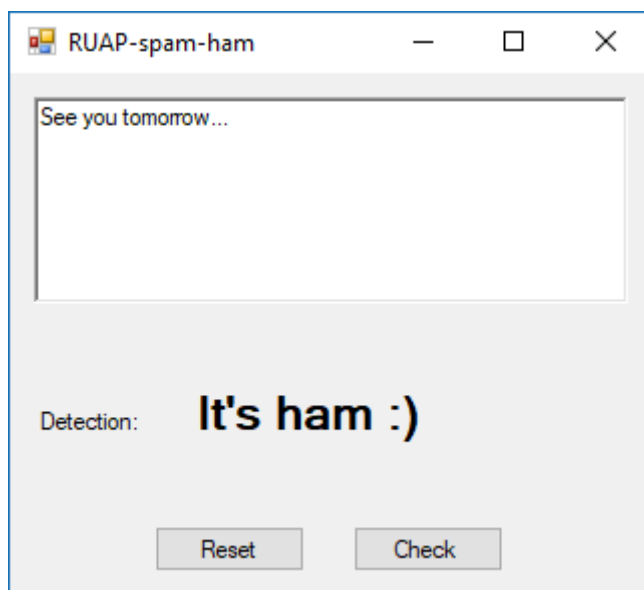
3.3. Klijentska aplikacija

Klijentska aplikacija realizirana je u obliku jednostavne Windows forme, koju redom čine: tekstualni okvir predviđen za unos poruke, prostor za ispis rezultata provjere te gumbi *Reset* i *Check*. Izgled aplikacije nakon otvaranja prikazan je slikom 3.10.



Sl. 3.10. Početni izgled klijentske aplikacije

Nakon unosa poruke potrebno je kliknuti gumb *Check* kako bi se iznad njega, a ispod tekstualnog okvira, ispisalo je li unesena poruka *ham* ili *spam*. Klikom na gumb *Reset* aplikacija se vraća u početno stanje. Slikom 3.11. prikazan je izgled aplikacije nakon provjere poruke.



Sl. 3.11. Izgled klijentske aplikacije nakon provjere poruke

4. ZAKLJUČAK

Strojno učenje korisno je u slučajevima kada iz nekih razloga nije moguće uobičajenim pristupom riješiti neke probleme, poput problema neželjenih poruka, pri čemu je potreban veliki uzorak podataka koji još k tome imaju veliki broj značajki. Rad na rješavanju ovakvih problema dodatno olakšavaju i različite usluge oblaka, poput Azure ML-a, u kojem se potrebni moduli jednostavno postavljaju na radni prostor i prilagođavaju sukladno potrebama. Također, u svakom koraku i u svakom trenutku moguće je vizualizirati promjene na podatkovnom skupu, što omogućuje ispravljanje, dopunjavanje i uređivanje te ponovno izvođenje samo onih dijelova za koje je to potrebno, bez ponovnog izvođenja cijelog modela.

POVEZNICE I LITERATURA

Programskom rješenju moguće je pristupiti preko:

[Programsko rješenje na GitHubu](#)

[ML model](#)

- [1] Hrvatska enciklopedija, *Globalizacija*,
<https://www.enciklopedija.hr/natuknica.aspx?id=22329>, pristupljeno 20. kolovoza 2020.
- [2] Hrvatska enciklopedija, *Neželjena pošta*,
<https://www.enciklopedija.hr/Natuknica.aspx?ID=68390>, pristupljeno 20. kolovoza 2020.
- [3] Nacionalni CERT, *Zaštita od neželjenih poruka verifikacijom adresa – SAV*,
<https://www.cert.hr/zastita-od-nezeljenih-poruka-verifikacijom-adresa-sav/>,
pristupljeno 20. kolovoza 2020.
- [4] China Daily, *You may get fewer spams on cell phones*,
https://www.chinadaily.com.cn/china/2009-06/13/content_8280507.htm,
pristupljeno 20. kolovoza 2020.
- [5] O. Stipetić, G. Valentić, V. Vazdar, *Prepoznavanje spam SMS poruka*,
https://web.math.pmf.unizg.hr/nastava/su/index.php/download_file/-/view/145/,
pristupljeno 26. kolovoza 2020.
- [6] T. Mitchell, *Machine Learning*, <https://www.cs.cmu.edu/~tom/mlbook.html>,
pristupljeno 27. kolovoza 2020.
- [7] Adatis, *Evaluating Models in Azure Machine Learning (Part 1: Classification)*,
<https://adatis.co.uk/evaluating-models-in-azure-machine-learning-part-1-classification/>,
pristupljeno 7. rujna 2020.