Obrada spam mail-ova metodom klasifikacije

Seminarski rad u okviru kursa
Istraživanje podataka
Matematički fakultet, Univerzitet u Beogradu

Tijana Todorov
485/2018
tijana.todorov710@gmail.com

19.08.2019

Sažetak

Cilj ovog rada je da se odredi najbolji spam filter za dobro odvajanje spam mailova od drugih. Ovo je testirano i obrađeno u programskom jeziku Python i IBM-ovom alatu SPSS Modeler pomoću nekoliko metoda koje su obrađene na predavanjima i vežbama.

Sadržaj

1 Uvod **2**

2 Upoznavanje sa podacima **2**

3 Priprema podataka za obradu **2**

4 Drveta odlučivanja **3**
4.1 SPSS Modeler . . . 3
4.1.1 C5.0 . . . 3
4.1.2 C&RT . . . 3
4.2 Python . . . 4

5 Najbliži susedi - KNN **6**
5.1 SPSS Modeler . . . 6
5.2 Python . . . 7

6 Neuronske mreže **8**
6.1 SPSS Modeler . . . 8
6.2 Python . . . 9

7 Metod potpornih vektora - SVM **10**

8 Gausova klasifikacija **11**

9 Zaključak **12**

Literatura **12**

1 Uvod

Spam poruke su zapravo neželjena pošta koja primaocu samo zatrpava sanduče. Ona može biti poruka koja sadrži nešto što primaoca ne zanima, da predstavlja reklamu, online prodavnicu i razne druge stvari. Isto tako ove poruke mogu predstavljati opasnost za primaoca ukoliko su zaražene virusom pa je zato najbolje takve poruke obrisati bez otvaranja. Zbog velikog broja neželjenih poruka koje se iz dana u dan sve više šalju iz raznih razloga kao što je jednostavniji i jeftiniji marketing vrlo često se dešava da poruke koje primalac očekuje završe greškom u Spam folderu. Iz tog razloga kako bi se što bolje napravila razlika između neželjene pošte i očekivane pošte veoma je bitno napraviti dobar spam filter koji će to razvrstavati.

2 Upoznavanje sa podacima

Podaci koji su korišćeni u ovom istraživanju se mogu pronaći na https://web.stanford.edu/~hastie/CASI_files/DATA/SPAM.html pod nazivom **SPAM.csv**. Skup sadrži podatke o spam porukama. U skupu se nalazi 4601 email poruka upućene istom korisniku sa 59 različitih atributa. Korisnik je označio 1813 email poruka od pristiglih kao spam.

U tabeli ima 57 numeričkih atributa koji predstavljaju najčešće korišćene reči u email porukama koje nisu trivijalne. Za svaku poruku predstavljena je frekvencija tih reči u njoj(procenat pojavljivanja). Osim njih postoje još 2 kategorička binarna atributa **spam** i **testid**.

- **spam** - označava da li je pošta neželjena ili ne
- **testid** - označava da li se instanca nalazi u trening ili test skupu
- 48 atributa koji predstavljaju u kom procentu se ta reč pojavljuje u email poruci, po formuli: **100*broj_pojavljivanja_reči** / **ukupan_broj_reči** (Za reč se smatra da sadrži niz alfanumeričkih karaktera)
- 6 atributa (oblika: ch; , ch(, ch[, ch! , ch$, ch#) koji predstavljaju u kom procentu se taj karakter nalazi u email poruci, po formuli: **100*broj_pojavljivanja_karaktera** / **ukupan_broj_karaktera**
- crl.ave označava prosečnu dužinu neprekidnih nizova velikih slova.
- crl.long označava dužinu najduže sekvence velikih slova.
- crl.tot označava zbir dužina neprekidnih sekvenci velikih slova tj. ukupan broj velikih slova u email poruci.

.

3 Priprema podataka za obradu

Zbog velike razlike u opsezima kod atributa, npr: atribut make je u segmentu [0 - 4.54], a atribut crl.tot u segmentu [1 - 15841] sve atribute sam normalizovala i svela na isti opseg [0 - 1]. U SPSS-u je to obrađeno pomoću čvora **_Norm** koji normalizuje izabrane atribute nad kojima sam vršila testiranje.

Ovakva normalizacija je obrađena i u Python-u sto je prikazano u Listingu 1 pored čega su i atributi **spam** i **testid** izmenjeni iz tipa Bool u String zbog modela koji zahtevaju da ciljni atribut bude tipa String.

Ciljni atribut nad kojim vršimo testiranje je atribut **spam**.

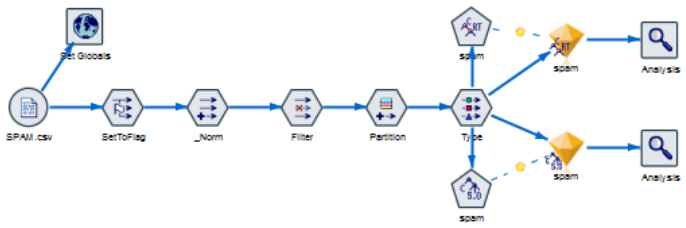
```
booleandf = df.select_dtypes(include=[bool])
booleanDictionary = {True: "tacno", False: "netacno"}

for column in booleandf:
    df[column] = df[column].map(booleanDictionary)

features1 = df.columns[0]
features5 = df.columns[4]
features9 = df.columns[8]
features10 = df.columns[9]
features12 = df.columns[11]
features16 = df.columns[15]
features17 = df.columns[16]
features19 = df.columns[18]
features26 = df.columns[25]

features = [features5, features9, features10, features12,
     features16, features17, features19, features26]
x_original = df[features]

x=pd.DataFrame(prep.MinMaxScaler().fit_transform(x_original))
```

Listing 1: Obrada podataka u Python-u

4 Drveta odlučivanja

4.1 SPSS Modeler

U nastavku će biti upoređeni rezultati primene algoritama C5.0 i C&Rt što je prikazano na slici 1. U čvoru *Partition* se vrši podela na trening i test podatke i validacioni skup uzimajući 70% podataka.



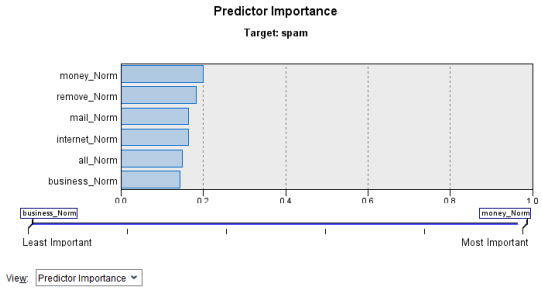
Slika 1: Primena modela C5.0 i C&Rt u SPSS-u

4.1.1 C5.0

Čvor *C5.0* se spaja sa *Type* čvorom gde se nalaze već sređeni podaci. C5.0 metod se poziva sa opcijom Group Symbolics i dobijen rezultat se tumači Analyze čvorom čiji su rezultati prikazani na slikama 2 i 3.

4.1.2 C&RT

Model se pravi pomoću čvora C&Rt. Cilj je izgraditi novi model u vidu drveta odlučivanja maksimalne dubine 3. Minimalan broj instanci u grani roditelja je 4%, a u grani deteta 2%. Kao mera nečistoće koristi se Ginijev kriterijum i minimalnom promenom u nečistoći od 0.0001%.



Slika 2: Bitnost atributa u modelu C5.0

Results for output field spam

Comparing $C-spam with spam

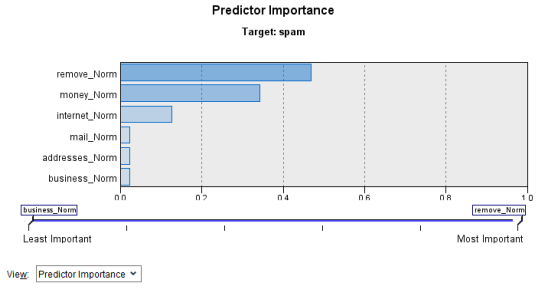
'Partition'	1_Training		2_Testing	
Correct	2,743	85.77%	1,197	85.32%
Wrong	455	14.23%	206	14.68%
Total	3,198		1,403	

Coincidence Matrix for $C-spam (rows show actuals)

'Partition' = 1_Training	FALSE	TRUE
FALSE	1,833	113
TRUE	342	910

'Partition' = 2_Testing	FALSE	TRUE
FALSE	783	59
TRUE	147	414

Slika 3: Rezultat Analyze čvora nad modelom C5.0



Slika 4: Bitnost atributa u modelu C&Rt

Analiza dobijenih rezultata može se videti na slikama 4 i 5.
Na slici 6 je prikazano drvo odlučivanja dobijeno generisanjem modela.

4.2 Python

Primena algoritma drveta odlučivanja u programskom jeziku Python prikazana je u fajlu dtree.py. Podaci se dele u trening i test skup, pri čemu je veličina test skupa 70% i prethodno je izvršena normalizacija podataka.

Drvo ima maksimalnu dubinu 12, za kriterijum podele koristi se Ginijev kriterijum i ostvareni su sledeći rezultati za trening i test podatke koji su prikazani u Listingu 2:

Results for output field spam

Comparing $R-spam with spam

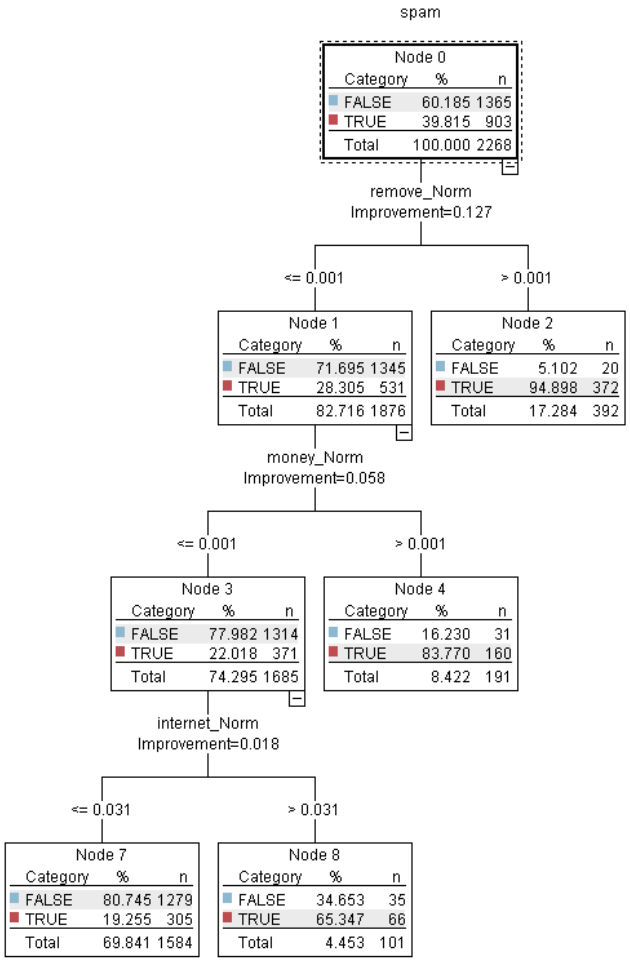
'Partition'	1_Training		2_Testing	
Correct	2,661	83.21%	1,188	84.68%
Wrong	537	16.79%	215	15.32%
Total	3,198		1,403	

Coincidence Matrix for $R-spam (rows show actuals)

'Partition' = 1_Training	FALSE	TRUE
FALSE	1,818	128
TRUE	409	843

'Partition' = 2_Testing	FALSE	TRUE
FALSE	793	49
TRUE	166	395

Slika 5: Rezultat Analyze čvora nad modelom C&Rt



Slika 6: Drvo odlučivanja - C&RT

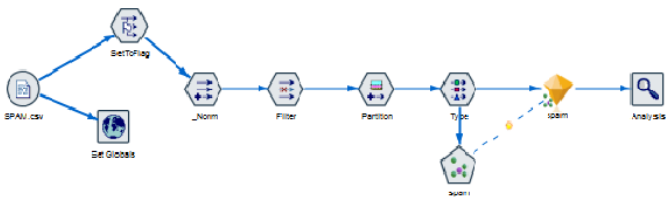
```
#Skup  Trening
Matrica konfuzije
         netacno  tacno
netacno     1882     69
tacno        374    895
Preciznost 0.8624223602484472
Preciznost po klasama [0.83421986 0.92842324]
Odziv po klasama [0.96463352 0.70527975]
Izvestaj klasifikacije
              precision    recall  f1-score   support

     netacno       0.83      0.96      0.89      1951
       tacno       0.93      0.71      0.80      1269

   micro avg       0.86      0.86      0.86      3220
   macro avg       0.88      0.83      0.85      3220
weighted avg       0.87      0.86      0.86      3220

#Skup  Test
Matrica konfuzije
         netacno  tacno
netacno      802     35
tacno        161    383
Preciznost 0.8580738595220855
Preciznost po klasama [0.83281412 0.91626794]
Odziv po klasama [0.95818399 0.70404412]
Izvestaj klasifikacije
              precision    recall  f1-score   support

     netacno       0.83      0.96      0.89       837
       tacno       0.92      0.70      0.80       544

   micro avg       0.86      0.86      0.86      1381
   macro avg       0.87      0.83      0.84      1381
weighted avg       0.87      0.86      0.85      1381
```

Listing 2: Rezultat nad trening i test podacima

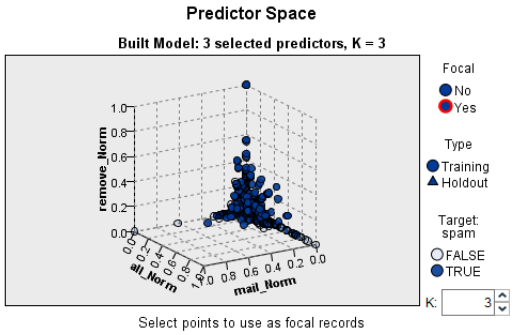
5 Najbliži susedi - KNN

5.1 SPSS Modeler

U SPSS-u učitavamo podatke, normalizujemo, filtriramo i povezujemo ih sa čvorom *Partition*, koji deli podatke na isti način kao u poglavlju 4. Model se generiše pokretanjem čvora *KNN*, koji prima particionisane podatke kao svoj ulaz. Ciljno polje je označeno kao *spam*, a ostala polja su ulazna. Minimalan broj k je postavljen na 3, maksimalan na 5, a udaljenost se računa Euklidskim rastojanjem. Rezultat ovog modela 7 je prikazan na grafiku 8 i analiziran pomoću čvora Analyze 9.



Slika 7: Analiza modela k najbližih suseda - SPSS



Slika 8: Grafikom prikazan rezultat KNN-a u SPSS-u

Results for output field spam

Comparing $KNN-spam with spam

'Partition'	1_Training		2_Testing	
Correct	1,683	52.63%	701	49.96%
Wrong	1,515	47.37%	702	50.04%
Total	3,198		1,403	

Coincidence Matrix for $KNN-spam (rows show actuals)

'Partition' = 1_Training	FALSE	TRUE
FALSE	558	1,388
TRUE	127	1,125
'Partition' = 2_Testing	**FALSE**	**TRUE**
FALSE	230	612
TRUE	90	471

Slika 9: Rezultat Analyze čvora nad modelom KNN

5.2 Python

Primena KNN algoritma je opisana u fajlu KNN.py. Veličina trening skupa je postavljena na 70%.

Najbolji rezultati dobijaju se za k = 4, Euklidsko rastojanje i kada svi susedi imaju podjednak uticaj, što se vidi na Listingu 3.

```
weights_values = ['uniform', 'distance']
#uniform
Matrica konfuzije
[[793  44]
 [209 335]]

Preciznost 0.8167994207096307
Izvestaj klasifikacije:
              precision    recall  f1-score   support

     netacno       0.79      0.95      0.86       837
       tacno       0.88      0.62      0.73       544

   micro avg       0.82      0.82      0.82      1381
   macro avg       0.84      0.78      0.79      1381
weighted avg       0.83      0.82      0.81      1381

#distance
Matrica konfuzije
[[770  67]
 [163 381]]

Preciznost 0.833454018826937
Izvestaj klasifikacije:
              precision    recall  f1-score   support
```

```
1026        netacno       0.83      0.92      0.87       837
              tacno       0.85      0.70      0.77       544
1028
          micro avg       0.83      0.83      0.83      1381
1030      macro avg       0.84      0.81      0.82      1381
   weighted avg       0.84      0.83      0.83      1381
```

Listing 3: Rezultat KNN-a

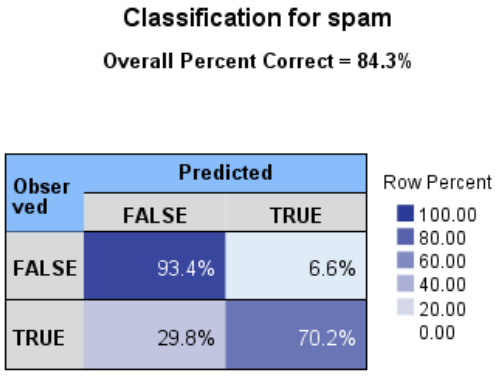
6 Neuronske mreže

6.1 SPSS Modeler

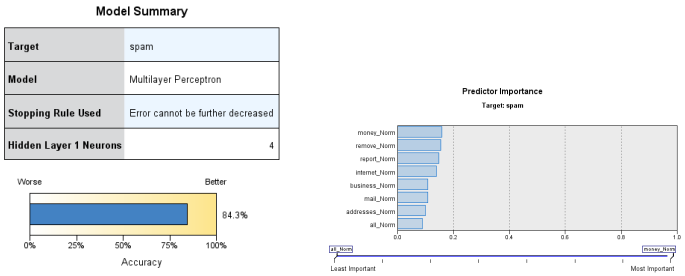
U SPSS-u učitavamo podatke, normalizujemo, filtriramo i povezujemo ih sa čvorom *Partition*, koji deli podatke na isti način kao u poglavlju 4. Model se generiše pokretanjem čvora *Neural Net*, povezanim sa čvorom *Type*. Za cilj je odabrano kreiranje novog višeslojnog modela.

Slika 10: Model Neuronske mreže u SPSS-u

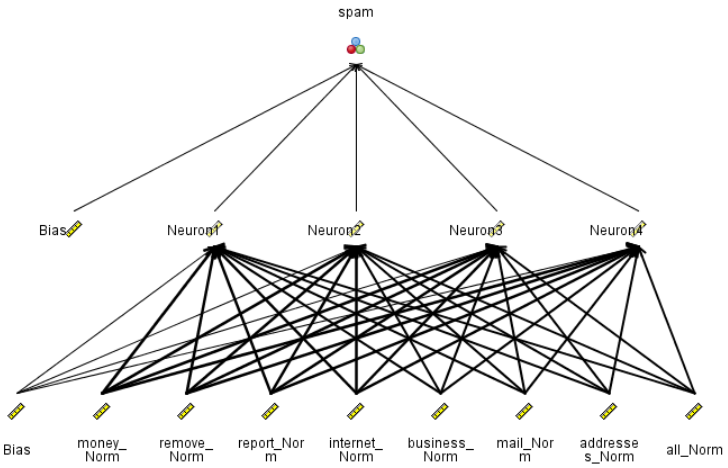
Kreirani model ima 1 skriveni sloj, kao što se vidi na slici 13 i razvijao se do trenutka kada više nije bilo moguće smanjiti grešku. Analiza modela prikazana je na slici 14.



Slika 11: Matrica konfuzije za neuronske mreže



Slika 12: Neuronska mreža



Slika 13: Neuronska mreža

Results for output field spam

Comparing $N-spam with spam

'Partition'	1_Training		2_Testing	
Correct	2,697	84.33%	1,199	85.46%
Wrong	501	15.67%	204	14.54%
Total	3,198		1,403	

Coincidence Matrix for $N-spam (rows show actuals)

'Partition' = 1_Training	FALSE	TRUE
FALSE	1,818	128
TRUE	373	879

'Partition' = 2_Testing	FALSE	TRUE
FALSE	780	62
TRUE	142	419

Slika 14: Rezultat Analyze čvora nad modelom Neuronske mreže

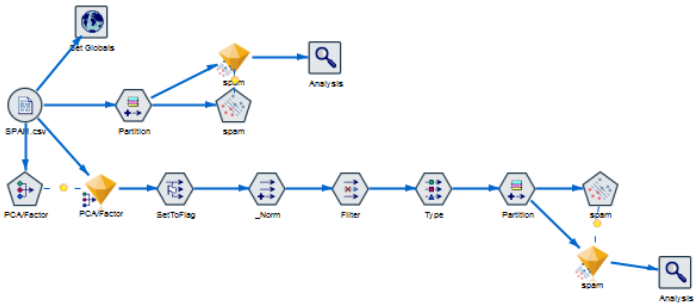
6.2 Python

Ovaj metod je primenjen na podatke koji su normalizovani kao sto je pomenuto u poglavlju 3, a opisan je u NeuronskeMLP.py fajlu, čiji je rezultat nad test podacima prikazan u Listingu 4.

```
Izvestaj za test skup:
Matrica konfuzije
[[786  51]
 [162 382]]

Preciznost 0.8457639391745112

Izvestaj klasifikacije
              precision    recall  f1-score   support

     netacno       0.83      0.94      0.88       837
       tacno       0.88      0.70      0.78       544

   micro avg       0.85      0.85      0.85      1381
   macro avg       0.86      0.82      0.83      1381
weighted avg       0.85      0.85      0.84      1381

Broj iteracija:  275
Broj slojeva:  4
```

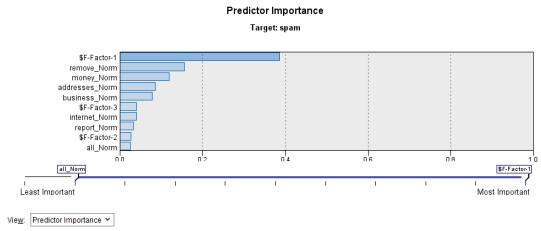
Listing 4: Rezultat nad test podacima

7 Metod potpornih vektora - SVM

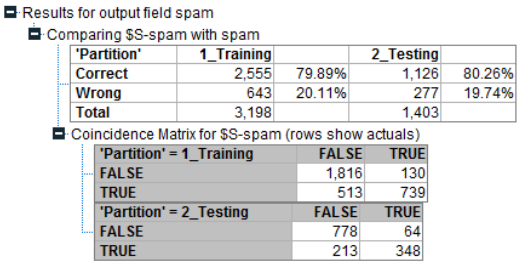


Slika 15: Model SVM u SPSS-u

Ovaj metod je primenjen u SPSS-u 15. Nad učitanim podacima prvo primenjujemo PCA model pomoću PCA čvora kako bismo smanjili skup atributa nad kojim radimo nakon čega ih delimo na trening i test podatke i pokrećemo model SVM pomoću istoimenog čvora. Rezultati dobijeni primenom ovog modela su analizirani pomoću čvora Analyze 17 i značajnost atributa su prikazani na 16.



Slika 16: Bitnost atributa u modelu SVM



Results for output field spam

Comparing $S-spam with spam

'Partition'	1_Training		2_Testing	
Correct	2,555	79.89%	1,126	80.26%
Wrong	643	20.11%	277	19.74%
Total	3,198		1,403	

Coincidence Matrix for $S-spam (rows show actuals)

'Partition' = 1_Training	FALSE	TRUE
FALSE	1,816	130
TRUE	513	739

'Partition' = 2_Testing	FALSE	TRUE
FALSE	778	64
TRUE	213	348

Slika 17: Rezultat Analyze čvora nad SVM metodom.

8 Gausova klasifikacija

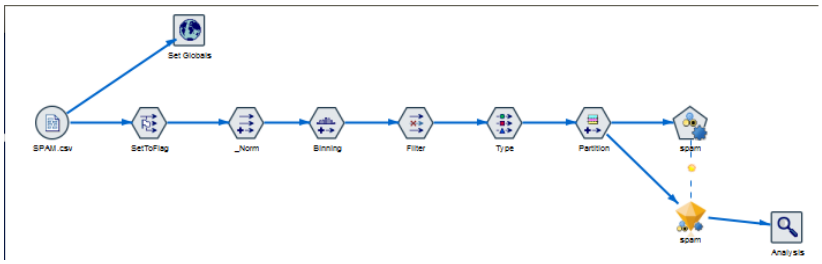
Ovaj metod je primenjen na podatke koji su normalizovani kao sto je pomenuto u poglavlju 3, a opisan je u Gaus.py fajlu, čiji je jedan deo prikazan u Listingu. Nad podacima je izvršena strafifikacija i za trening skup uzeto je 70% podataka i rezultat je prikazan u Listingu 5.

```
Matrica konfuzije
[[808  29]
 [306 238]]

Preciznost 0.7574221578566256
Izvestaj klasifikacije
              precision    recall  f1-score   support

       tacno       0.73      0.97      0.83       837
     netacno       0.89      0.44      0.59       544

   micro avg       0.76      0.76      0.76      1381
   macro avg       0.81      0.70      0.71      1381
weighted avg       0.79      0.76      0.73      1381
```

Listing 5: Rezultat nad trening podacima

9 Zaključak

Analiziranjem SPAM.csv skupa podataka metodom klasifikacije i primenom sledecih metoda: C5.0, C&Rt, KNN, Neuronske mreže, SVM i Gausa dolazimo do zaključka da **C&Rt** daje najbolje rezultate. Sve ove metode su obrađene u SPSS-u i Python-u ali je njihovo poređenje izvšeno u SPSS-u pomoću čvora **Auto Classifier** 18 koji je primenjen nad već obrađenim, normalizovanim podacima.



Slika 18: Poređenje svih metoda u SPSS-u

Pokretanjem **Auto Classifier** dobijamo za rezultat sledecu tabelu koja je prikazana na slici 19 gde su metodi sortirani od onog koji daje najbolje do onog koji daje najlošije rezultate nad ovim skupom podataka.

Use?	Graph	Model	Build Time (mins)	Overall Accuracy (%)	No. Fields Used
☑		C&R Tree 1	< 1	86.386	8
☑		Neural Net 1	< 1	85.460	8
☑		C5 1	< 1	85.317	7
☑		SVM 1	< 1	80.969	8
☑		KNN Algorithm 1	< 1	53.457	8

Slika 19: Rezultat poređenja svih metoda

Rezultat dobijen na prethodnoj slici smo analizirali čvorom Analyze koji je vratio sledeci rezultat 20.

Results for output field spam

Comparing $XS-spam with spam

'Partition'	1_Training		2_Testing	
Correct	2,719	85.02%	1,206	85.96%
Wrong	479	14.98%	197	14.04%
Total	3,198		1,403	

Coincidence Matrix for $XS-spam (rows show actuals)

'Partition' = 1_Training	FALSE	TRUE
FALSE	1,847	99
TRUE	380	872
'Partition' = 2_Testing	**FALSE**	**TRUE**
FALSE	796	46
TRUE	151	410

Slika 20: Rezultat Analyze čvora nad čvorom koji poredi sve metode