Klasifikaciija regulatornih i pomoćnih T ćelija

Seminarski rad u okviru kursa
Istraživanje podataka 2
Matematički fakultet

Stefan Stanišić

Septembar 2019

Sažetak

U ovom seminarskom radu obrađena je klasifikacija i primena velikog broja algoritama klasifikacije kako bismo pronašli onaj koji najbolje vrši klasifikaciju nad podacima. Podaci nad kojima je vršena klasifikacija su dve vrste T ćelija. Prikazaćemo rezultate primene različitih algoritama kako bismo otkrili koji najbolje klasifikuje date podatke.

Sadržaj

1 Uvod **2**

2 Pretprocesiranje **2**

3 Klasifikacija **2**
3.1 Jednostavne metode i rezultati . . . 3
3.1.1 K najbližih suseda . . . 3
3.1.2 Drvo odlučivanja . . . 4
3.1.3 Mašine sa potpornim vektorima . . . 5
3.2 Ansambl klasifikacione tehnike . . . 6
3.2.1 Nasumična šuma (eng. Random forest) . . . 6
3.2.2 Pakovanje (eng. Bagging) . . . 7
3.2.3 Pojačavanje (eng. Boosting) . . . 8
3.2.4 Glasanje (eng. Voting) . . . 9

4 Analiza dobijenih rezultata **10**
4.1 K najbližih suseda . . . 10
4.2 Drvo odlučivanja . . . 11
4.3 Mašine sa potpornim vektorima . . . 11
4.4 Nasumična šuma . . . 11
4.5 Pakovanje . . . 12
4.6 Pojačavanje . . . 12
4.7 Glasanje . . . 12
4.8 Jednostavne metode . . . 13
4.9 Ansambl metode . . . 13

5 Zaključak **14**

1 Uvod

Pre nego što počnemo da govorimo o metodama klasifikacije i njihovim primenama, neophodno je prvo da definišemo šta je klasifikacija. Klasifikacija predstavlja vid nadgledanog učenja, što znači da se podaci dele na dva skupa, trening i test skup. Na osnovu trening skupa klasifiktor vrši klasifikaciju u test skupu. Trening skup je u potpunosti poznat, to jest, za svaki podatak, osim njegovih karakteristika, je dato i kojoj klasi pripada, dok u slučaju test skupa, na osnovu trening skupa, klasifikator ima zadatak da pronade ciljnu klasu. Sada kada smo se upoznali sa pojmom klasifikacije treba i predstaviti podatke nad kojima ćemo vršiti klasifikaciju. Klasifikaciju ćemo primenjivati nad podacima koji se tiču regulatornih(supresorskih) i pomoćnih T ćelija. T ćelije su vrsta limfocita koje se razvijaju u grudnoj žlezdi i imaju jednu od ključnih uloga u imunitetu ljudskog organizma. Podaci su organizovani u 2 datoteke:

- 087_CD4+_Helper_T_Cells_csv.csv
- 088_CD4+CD25+_Regulatory_T_Cells_csv.csv

Pre nego što započnemo sa primenom algoritama klasifikacije nad datim podacima potrebno je da izvršimo pretprocesiranje. O tome će biti više reči u sledećem odeljku. Kada završimo sa pretprocesiranjem započećemo sa primenom algoritama klasifikacije. Neki algoritmi će biti primenjeni više puta sa različitim vrednostima parametara kako bismo dobili što bolje rezultate. Pretprocesiranje i algoritmi klasifikacije su implementirani u programskom jeziku Python, uz pomoć biblioteka pandas, numpy, sklearn, os i time.

2 Pretprocesiranje

Pretprocesiranje predstavlja jako bitan korak prilikom klasifikacije podataka. Ovaj proces je neophodan zarad poboljšanja efikasnosti algoritama klasifikacije. Preprocesiranje se sastoji od sledećih koraka:

- Obradivanje svake od datoteka na ulazu - neophodno je transponovati svaku datoteku, zatim ukloniti nultu kolonu koja nam nije od koristi(nastala je transponovanjem)
- Spajanje datoteka - vršimo spajanje svih datoteka u jednu glavnu datoteku. Pre samog procesa spajanja neophodno je svakoj od njih dodati oznaku klase. Za prvu datoteku 1, a za drugu datoteku 2.
- Prečišćavanje - iz glavne datoteke dobijene u prethodnom koraku, vrši se brisanje svih kolona koje imaju vrednost 0 za svaki red. Ovo radimo iz razloga što nula-redovi i nula-kolone nisu značajni za generisanje modela klasifikacije, a njihovim uklanjanjem podižemo efikasnost izvršavanja programa.

Nakon pretprocesiranja dobijena je datoteka dimenzije (22075, 16708)
.

3 Klasifikacija

Nakon što smo pripremili podatke, možemo početi sa generisanjem modela za klasifikaciju. Skup podataka delimo na dva dela, prvi predstavlja matricu sa svim podacima bez kolone 'class', a drugi vektor-kolonu

koja sadrži oznake klasa kojima pripadaju redovi matrice. Sada ćemo dobijene podatke podeliti na trening i test skup. Podatkes ćemo podeliti u odnosu 70:30, gde je 70 % trening podataka, a 30 % test podataka.

Za klasifikaciju podataka koristićemo smo sledeće algoritme:

- Jednostavne metode:
  - K najbližih suseda (KNN)
  - Drvo odlučivanja (DTC)
  - Mašine sa potpornim vektorima (SVM)
- Ansambl tehnike:
  - Nasumična šuma (eng. Random forest)
  - Pakovanje (eng. Bagging)
  - Pojačavanje (eng. Boosting)
  - Glasanje (eng. Voting)

Za svaki metod ćemo prikazivate rezultate na trening i test podacima kao i vreme izvršavanja izraženo u sekundama.

3.1 Jednostavne metode i rezultati

3.1.1 K najbližih suseda

Osnovna ideja ovog algoritma je da na osnovu k najbližih suseda datog sloga odredimo kojoj klasi on pripada. Vrši se određivanje najbližih suseda, zatim prebrojavanje koliko suseda pripada kojoj klasi. Ona klasa kojoj pripada najviše suseda je dodeljena posmatranom slogu. Optimalan odabir vrednosti za k je jako zavisno od podataka nad kojim se vrši klasifikacija [1] . Generalno, veća vrednost broja k suzbija efekte šuma, ali čini da se granice klasifikacije manje razlikuju. U slučajevima gde podaci nisu uniformno uzorkovani, bolje je koristiti drugu vrstu klasifikacije, kao što je "RadiusNeighborsClassifier". Mi ćemo algoritam primenjivati za 3, 5 i 10 suseda. Dobijeni su sledeci rezultati:

```
KNN za k = 3 i uniformnim težinama:
Rezultat trening skupa: 0.832
Rezultat test skupa: 0.674
Matrica konfuzije trening skupa:
[[6866 1058]
 [1543 5985]]
Matrica konfuzije test skupa:
[[2516  927]
 [1233 1947]]
Vreme izvrsavanja: 14914.421
```

```
KNN za k = 5 i uniformnim težinama:
Rezultat trening skupa: 0.799
Rezultat test skupa: 0.704
Matrica kofuzije trening skupa:
[[6745 1155]
 [1953 5599]]
Matrica kofuzije test skupa:
[[2728  739]
 [1222 1934]]
Vreme izvrsavanja: 15016.877
```

```
KNN za k = 10 i uniformnim težinama:
Rezultat trening skupa: 0.760
Rezultat test skupa: 0.711
Matrica kofuzije trening skupa:
[[7102  798]
 [2914 4638]]
Matrica kofuzije test skupa:
[[2981  486]
 [1426 1730]]
Vreme izvrsavanja: 15176.648
```

3.1.2 Drvo odlučivanja

Problem klasifikacije rešavamo postavljanjem pitanja o vrednostima atributa podataka iz trening skupa. Svaki put kada dobijemo odgovor postavljamo novo pitanje, dok ne dođemo do zaključka o klasi posmatranog sloga. Klasifikator 'DecisionTreeClassifier' u python3 je implementiran korišćenjem CART (Classification And Regression Trees) algoritma [2] . Mi ćemo koristiti samo rešenje klasifikacionog problema algoritma, koji predviđa vrednost kategoričke klase na osnovu neprekidnih i/ili kategoričkih atributa. Ukoliko se ne navede drugačije, kao meru nečistoće uzimamo Ginijev indeks. Funkciji šaljemo trening skup, test skup i jednu od dve mere nečistoće koje ćemo koristiti za pravljenje modela (Ginijev indeks i entropiju). U prvom testiranju nećemo ograničavati dubinu drveta. Dobijeni su sledeci rezultati:

```
Drvo odlučivanja sa Ginijevim indeksom nečistoće
i neograničenom dubinom:
Rezultat trening skupa: 1.000
Rezultat test skupa: 0.667
Matrica kofuzije trening skupa:
[[7935    0]
 [   0 7517]]
Matrica kofuzije test skupa:
[[2320 1112]
 [1096 2095]]
Vreme izvrsavanja: 91.499
```

```
Drvo odlučivanja sa ''entropy'' merom nečistoće
i neograničenom dubinom:
Rezultat trening skupa: 1.000
Rezultat test skupa: 0.673
Matrica kofuzije trening skupa:
[[7889    0]
 [   0 7563]]
Matrica kofuzije test skupa:
[[2332 1146]
 [1018 2127]]
Vreme izvrsavanja: 67.906
```

```
Drvo odlučivanja sa Ginijevim indeksom nečistoće
i ograničenom dubinom do petog nivoa.
Rezultat trening skupa: 0.733
Rezultat test skupa: 0.714
Matrica kofuzije trening skupa:
[[6356 1646]
 [2484 4966]]
Matrica kofuzije test skupa:
[[2602  763]
 [1128 2130]]
Vreme izvrsavanja: 50.617
```

```
Drvo odlučivanja sa ''entropy'' merom nečistoće
i ograničenom dubinom do petog nivoa.
Rezultat trening skupa: 0.728
Rezultat test skupa: 0.725
Matrica kofuzije trening skupa:
[[6082 1916]
 [2294 5160]]
Matrica kofuzije test skupa:
[[2550  819]
 [1004 2250]]
Vreme izvrsavanja: 49.863
```

3.1.3 Mašine sa potpornim vektorima

Mašine sa potpornim vektorima predstavljaju skup metoda sa nadgledanim učenjem koji se koriste za klasifikciju, regresiju kao i detekciju elemenata van granice. Osnovne prednosti ovog metoda su:

- Efikasna u visoko dimenzionim prostorima
- Efikasna u slučajevima gde je broj dimenzija veći od broja uzoraka
- Svestranost - moguća primena različitih jezgara za funkciju odredivanja

Mane ovog metoda su:

- Ne daje direktno ocene verovatnoća, već se one izračunavaju koristeći petostruku unakrsnu validaciju

Osnovna ideja na kojoj je baziran ovaj metod jeste pronalaženje hiperravni koja treba da razdvoji podatke tako da se svi podaci iste klase nalaze sa iste strane hiper-ravni [3] .Neke od vrsta jezgara koje ćemo koristiti su: linear, poly, rbf . Dobijeni su sledeci rezultati:

```
Mašine sa potpornim vektorima i Gausovim jezgrom:
Rezultat trening skupa: 0.864
Rezultat test skupa: 0.829
Matrica kofuzije trening skupa:
[[6995 1011]
 [1083 6363]]
Matrica kofuzije test skupa:
[[2823  538]
 [ 595 2667]]
Vreme izvrsavanja: 7521.270
```

```
Mašine sa potpornim vektorima i linearnim jezgrom:
Rezultat trening skupa: 1.000
Rezultat test skupa: 0.803
Matrica kofuzije trening skupa:
[[7966    0]
 [   0 7486]]
Matrica kofuzije test skupa:
[[2748  653]
 [ 655 2567]]
Vreme izvrsavanja: 11129.016
```

```
Mašine sa potpornim vektorima i polinomijalnim' jezgrom:
Rezultat trening skupa: 0.846
Rezultat test skupa: 0.825
Matrica kofuzije trening skupa:
[[6599 1414]
 [ 967 6472]]
Matrica kofuzije test skupa:
[[2702  652]
 [ 505 2764]]
Vreme izvrsavanja: 7754.807
```

3.2 Ansambl klasifikacione tehnike

Ansambl tehnike se nazivaju i meta klasifikacione metode, zato što ne možemo odmah napraviti model klasifikacije, već moramo konstruisati nekoliko jednostavnih modela, pomenutih u prethodnoj sekciji čije rezultate ove tehnike kombinuju u cilju smanjenja nivoa greške.

3.2.1 Nasumična šuma (eng. Random forest)

Deli skup podataka na komplementarne podskupove i za svaki od podskupova, generiše zaseban model drveta odlučivanja. Krajnji model predstavlja srednju vrednost rezultata dobijenih iz generisanih modela [4] . Pojedinačno drvo odlučivanja smo već obradili u prethodnom odeljku. Jedan

novi parametar koji trebamo poslati funkcija je broj drveta odlučivanja koje treba napraviti. Konstruisaćemo i uporediti rezultate modela za 10, 50 i 100 drveta odlučivanja. Dobijeni su sledeci rezultati:

```
RFC, 10 modela drveta odlučiivanja:
Rezultat trening skupa: 0.712
Rezultat test skupa: 0.679
Matrica kofuzije trening skupa:
[[6391 1579]
 [2865 4617]]
Matrica kofuzije test skupa:
[[2671  726]
 [1400 1826]]
Vreme izvrsavanja: 27.704
```

```
RFC, 50 modela drveta odlučiivanja:
Rezultat trening skupa: 0.729
Rezultat test skupa: 0.700
Matrica kofuzije trening skupa:
[[6513 1426]
 [2768 4745]]
Matrica kofuzije test skupa:
[[2713  715]
 [1274 1921]]
Vreme izvrsavanja: 33.489
```

```
RFC, 100 modela drveta odlučiivanja:
Rezultat trening skupa: 0.731
Rezultat test skupa: 0.719
Matrica kofuzije trening skupa:
[[6562 1394]
 [2761 4735]]
Matrica kofuzije test skupa:
[[2806  605]
 [1256 1956]]
Vreme izvrsavanja: 40.287
```

3.2.2 Pakovanje (eng. Bagging)

Ansambl tehnika koja deli ulazni skup podataka na podskupove u kojima se elementi mogu ponavljati i za svaki skup formira zaseban model. Krajnji model se formira računanjem srednje vrednosti svih prethodno formiranih parcijalnih modela [5]. Testiraćemo ovu tehniku korišćenjem 2 osnovna modela. Jedan će biti drvo odlučivanja sa ograničenom dubinom, a drugi mašina sa potpornim vektorima koja koristi linearni kernel. Pakovanje sa drvetom odlučivanja kao primarnom metodom ćemo pozivati sa 10 i 50 različitih modela, dok ćemo kod mašina sa potpornim vektorima koristiti 5 modela. Dobijeni su sledeci rezultati:

```
Bagging, drvo odlučivanja, 10 modela:
Rezultat trening skupa: 0.744
Rezultat test skupa: 0.722
Matrica kofuzije trening skupa:
[[6175 1719]
 [2233 5325]]
Matrica kofuzije test skupa:
[[2658  815]
 [1023 2127]]
Vreme izvrsavanja: 730.368
```

```
Bagging, drvo odlučivanja, 50 modela
Rezultat trening skupa: 0.764
Rezultat test skupa: 0.737
Matrica kofuzije trening skupa:
[[6328 1586]
 [2068 5470]]
Matrica kofuzije test skupa:
[[2670 783]
 [958 2212]]
Vreme izvrsavanja: 3574.389
```

```
Bagging, svm, 5 modela
Rezultat trening skupa: 0.973
Rezultat test skupa: 0.825
Matrica kofuzije trening skupa:
[[7695 219]
 [197 7341]]
Matrica kofuzije test skupa:
[[2848  605]
 [553 2617]]
Vreme izvrsavanja: 25933.935
```

3.2.3 Pojačavanje (eng. Boosting)

Na početku se formira loš klasifikator i svim slogovima se dodaju jednake težine. Kroz iteracije se vrši prepravka težina na osnovu rezultata iz prethodne iteracije. Ako je podatak tačno klasifikovan, težina sloga se smanjuje, dok ako je klasifikovan pogrešno, težina se povećava [6] . Glavna ideja iz ovog meta-klasifikatora je da na osnovu nekoliko slabih klasifikatora, napravi jedan jak. Kako tehnika pojačavanja ne može da radi sa mašinama sa potpornim vektorima, kao osnovni model ćemo koristiti samo drvo odlučivanja sa ograničenom dubinom. Algoritam ćemo pozivati sa 10, 50 i 100 različitih modela drveta odlučivanja. Dobijeni su sledeci rezultati:

```
Boosting, 10 modela:
Rezultat trening skupa: 0.846
Rezultat test skupa: 0.760
Matrica kofuzije trening skupa:
[[6756 1138]
 [1245 6313]]
Matrica kofuzije test skupa:
[[2651  822]
 [765 2385]]
Vreme izvrsavanja: 577.704
```

```
Boosting, 50 modela:
Rezultat trening skupa:0.989
Rezultat test skupa: 0.761
Matrica kofuzije trening skupa:
[[7971   68]
 [ 105 7308]]
Matrica kofuzije test skupa:
[[2598  730]
 [ 852 2443]]
Vreme izvrsavanja: 2790.917
```

```
Boosting, 100 modela:
Rezultat trening skupa: 1.000
Rezultat test skupa: 0.755
Matrica kofuzije trening skupa:
[[8039    0]
 [   0 7413]]
Matrica kofuzije test skupa:
[[2567  761]
 [ 863 2432]]
Vreme izvrsavanja: 5569.837
```

3.2.4 Glasanje (eng. Voting)

Svaki ulazni slog se klasifikuje svim prosleđenim osnovnim modelima i na osnovu dobijenih rezultata određuje se klasa svakog sloga [7]. U našem primeru koristićemo tri modela: nasumičnu šumu koju formiramo pomoću 100 drveta odlučivanja sa ograničenom dubinom, mašinu sa potpornim vektorima uz korišćenje linearnog kernela i drvo odlučivanja sa ograničenom dubinom. Glasanje u oba poziva funkcija je "hard", što znači da će rezultati tri osnovna modela biti upoređivani na svakom slogu i ona klasa koja ima više glasova, da se podsetimo postoje dve klase, biće dodeljena posmatranom slogu. Dobijeni su sledeci rezultati:

```
Voting:
Rezultat trening skupa: 0.847
Rezultat test skupa: 0.772
Matrica kofuzije trening skupa:
[[7311  728]
 [1631 5782]]
Matrica kofuzije test skupa:
[[2849  479]
 [1031 2264]]
Vreme izvrsavanja: 17620.884
```

4 Analiza dobijenih rezultata

U narednoj sekciji ćemo analizirati dobijene rezultate i razmatrati koji algoritmi su se najbolje pokazali:

4.1 K najbližih suseda

Možemo primetiti da se povećanjem broja suseda, rezultati klasifikovanja neznatno poboljšavaju i da je od ispitanih modela najbolji onaj sa 10 suseda. Model sa 3 suseda je dao najbolje rezultate na trening skupu, ali se najlošije pokazao na test podacima, dok za modela za 10 suseda važi obrnuto. Na trening podacima se najslabije pokazao od sva 3 modela, ali zato je na test podacima dao najbolje rezultate.

Broj suseda	Rezultat trening skupa	Rezultat test skupa	Vreme izvršavanja
3	0.832	0.674	14914.421
5	0.799	0.704	15016.877
10	0.760	0.711	15176.648

Tabela 1: Rezultati algoritma KNN

4.2 Drvo odlučivanja

Kao što smo naveli u odeljku konstruisanje modela nad podacima bez ograničavanja dubine dovodi do blagog preprilagođavanja podacima. Uz ograničavanje dubine (odsecanje stabla) dobili smo lošije rezultate nad trening podacima, tj nije došlo do preprilagođavanja podacima, ali smo dobili bolje rezultate na test podacima kao i kraće vreme izvršavanja.

Mere nečistoće	Dubina drveta	Rezultat trening skupa	Rezultat test skupa	Vreme izvršavanja
Gini	Neograničena	1.000	0.667	91.499
Entropija	Neograničena	1.000	0.673	67.906
Gini	5	0.733	0.714	50.617
Entropija	5	0.728	0.725	49.863

Tabela 2: Rezultati algoritma drveta odlučivanja

4.3 Mašine sa potpornim vektorima

Kod linearnog kernela došlo je do preprilagođavanja podacima, a kasnije kod trening podataka dao je najslabije rezultate i najduže mu je trebalo da se izvrši zbog tog preprilagovanja. Gausov i polinomijalan kernel su dali bolje rezultate od kojih je model sa Gausovim jezgrom najbolji.

Kernel	Rezultat trening skupa	Rezultat test skupa	Vreme izvršavanja
Gausov	0.864	0.829	7521.270
Linearan	1.000	0.803	11129.016
Polinomijalan	0.846	0.825	7754.807

Tabela 3: Rezultati algoritma mašine sa potpornim vektorima

4.4 Nasumična šuma

Uporedjivanjem rezultata nasumiče šume i jednog drveta odlučivanja vidimo da nije došlo do prepriilagodjavanja kao kod neograničene dubine jednog drveta odlučivanja. Takodje rezultati su bolji nego kod jednog drveta odlučivanja. Sva tri dobijena modela se brže izvršavaju u odnosu jedno stablo odlučivanja. Rezultati na trening skupu su približni, ali na test podacima algoritam jednog drveta odlučivanja je dao bolje rezultate nego nasumična šuma. Od tri dobijena rezultata algoritma nasumične šume najbolje se pokazao onaj sa 100 modela, ali se i najduže izvršavao.

Broj modela	Rezultat trening skupa	Rezultat test skupa	Vreme izvršavanja
10	0.712	0.679	27.704
50	0.729	0.700	33.489
100	0.731	0.719	40.287

Tabela 4: Rezultati algoritma nasumičn šume

4.5 Pakovanje

Prvu stvar koju možemo primetiti kod modela je proporcionalno vreme izvšavanja broju osnovnih modela korišćenih za generisanje. Model koji je konstruisan pomoću mašina sa potpornim vektorima je dao bolje rezultate od modela sa drvetom odlučivanja što je i očekivano jer su i kod primene tih algoritama pojedinačno na podatke dobijeni bolji rezultati.

Osnovni model	Broj modela	Rezultati trening skupa	Rezultat test skupa	Vreme izvršavanja
Drvo odlučivanja	10	0.744	0.722	730.368
Drvo odlučivanja	50	0.764	0.737	3574.389
SVC	5	0.973	0.825	25933.935

Tabela 5: Rezultati algoritma pakovanja

4.6 Pojačavanje

Kao i kod klasifikacije pakovanjem i u ovom slučaju je srazmerno broju osnovnih modela korišćenih za generisanje. Model dobijen pomoću 100 osnovnih modela je bio nepogrešiv na trening podacima, ali se zato najslabije pokazao od sva 3 modela na test podacima. Rezultati modela dobijenog sa 50 osnovnih modela su odlični na trening podacima, ali su slabiji na test podacima. Ipak ti rezultati na test podacima su najbolji od sva 3 dobijena modela.

Broj modela	Rezultat trening skupa	Rezultat test skupa	Vreme izvršavanja
10	0.846	0.760	577.704
50	0.989	0.761	2790.917
100	1.000	0.755	5569.837

Tabela 6: Rezultati algoritma pojačavanja

4.7 Glasanje

Dobijeni rezultati klasifikacije glasanjem su solidni, jedini problem je što se malo duže izvršava proces klasifikaciije.

Rezultat trening skupa	Rezultat test skupa	Vreme izvršavanja
0.847	0.772	17620.884

Tabela 7: Rezultati algoritma glasanja

4.8 Jednostavne metode

Nakon što smo završili analizu svih metoda pojedinačno, u ovom odeljku ćemo analizirati grupe metoda, da bismo viideli koje su se najbolje pokazale nad našim podacima. Za svaku metodu uzimamo parametre sa kojima se ta metoda najbolje pokazala. Kod algoritma KNN najbolje rezultati su dobiijeni za 10 komšija (tj. vrednost k = 10). Za drvo odlučivanja najbolje se pokazalo ograničavanje do petog nivoa i entropijom kao merom nečistoće, a za mašinu sa potpornim vektorima najbolje se pokazao Gausov kernel.

Metoda	Rezultat trening podataka	Rezultat test skupa	Vreme izvršavanja	Tačno klasifiovani	Pogrešno klasifikovani
KNN	0.760	0.711	15176.648	4711	1912
Drvo odlučivanja	0.728	0.725	49.863	4800	1823
SVM	0.864	0.829	7521.270	5490	1133

Tabela 8: Poredjenja najboljih rezultata jednostavnih metoda

Kod rezultata svim metodama vidimo veoma malu razliku izmedju rezultata dobijenih za trening i test skup. Mašina sa potpornim vektorima se najbolje pokazala i kod nje smo dobili vrlo dobre rezultate.

4.9 Ansambl metode

U ovom odeljku ćemo kao i kod jednostavnih metoda analizirati grupe metoda kako bismo videli koje su se najbolje pokazale. Od algoritma nasumične šume ćemo uzeti model dobijen pomoću 100 drveta odlučivanja bez ograničavanje dubine, kod pakovanja ćemo uzeti model dobijen pomoću mašine sa potpornim vektorima, od algoritma pojačavanja ćemo uzeti onaj generisan pomoću 50 osnovnih modela. I kao poslednji jeste jedinig model dobijen glasanjem.

Metoda	Rezultat trening skupa	Rezultat test skupa	Vreme izvršavanja	Tačno klasifikovani	Pogrešno klasifikovani
Nasumična šuma	0.731	0.719	40.287	4762	1861
Pakovanje	0.973	0.825	25933.935	5465	1158
Pojačavanje	0.989	0.761	2790.917	5041	1582
Glasanje	0.847	0.772	17620.884	5113	1510

Tabela 9: Poredjenja najboljih rezultata ansambl metoda

Od dobijenih rezultata najslabije se pokaza model dobijen algoritmom nasumične šume, dok je kao i kod jednostavnih metoda najbolji model

onaj koji je dobijen pomoću pakovanja i mašine sa potpornim vektorima. Takođe možemo uvideti da su dobijeni rezultati proporcionalni sa vremenom izvršavanja. Model dobijen nasumičnom šumom se najbrže generisao, ali je dao i najslabije rezultate, dok je model generisan pakovanjem najsporije generisan ali i dao najbolje rezultate. Uopšteno, rezultati dobijeni ansambl metodama su bolji od onih dobijenih jednostavnim metodama što je i očekivano.

5 Zaključak

Kroz ovaj seminarski rad, ideja je bila pokazati primenu različitih metoda klasifikacije kako bismo pronašli onaj koji će dati najbolje rezultate. Oprobane su različite metode sa različitim vrednostima parametara i svaka od njih je imala manju ili veću uspešnost. Kada pogledamo sveobuhvatne rezultate prvo što se može zaključiti jeste da je najslabiije rezultate dao model dobijen metodom najbližih suseda. Drvo odlučivanja i nasumična šuma su dali solidne rezultate za veoma kratko vreme u poredjenju sa vremenima konstruisanja drugih modela, ali isto tako su i slabijii u odnosu na rezultate drugih metoda. Najbolje rezultate od svih su dali modeli dobijeni pomoću metoda mašine sa potpornim vektorima što je donekle i očekivano jer je to najbolji linearni klasifikator. Zbog velike količine podataka nije izvršena klasifikacija algoritma pakovanja pomoću mašine sa potpornim vektorima i 20 osnovnih modela za koju verujem da bi dala najbolje rezultate od svih metoda.

Literatura

[1] K nearest neighbors. https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html

[2] Decision tree clasifier. https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html

[3] Support vector machine. https://scikit-learn.org/stable/modules/svm.html

[4] Random forest classifier. https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html

[5] Bagging classifier. https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.BaggingClassifier.html

[6] Boosting classifier. https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.AdaBoostClassifier.html

[7] Voting classifier. https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.VotingClassifier.html