

Fakultet elektrotehnike, računarstva i informacijskih tehnologija

**Kolegij:**

**Raspoznavanje uzoraka i strojno učenje**

**Naziv projekta:**

**Klasifikacija podataka putem Random forest metode**

**Bošnjak Petar**

Sadržaj

[1. Uvod 3](#_Toc493550796)

[2. Klasifikacija 4](#_Toc493550797)

[3. Random forest algoritam 7](#_Toc493550798)

[4. Predstavljanje problema 11](#_Toc493550799)

[5. Rješenje problema 12](#_Toc493550800)

[6. Zaključak 14](#_Toc493550801)

[Literatura 16](#_Toc493550802)

# Uvod

Tema projektnog zadatka jest klasifikacija podataka analize testova krvi i tumorskih markera. Testovi su rađeni na skupini pacijenata sa dijagnosticiranim tumorom dojke, placebo skupini, te skupini pacijenata sa benignim tumorom.

Cilj je ustanoviti može li se poboljšati dijagnoza tumora kombinirajući testove tumorskih markera Ca 15-3 i CEA sa rutinskim testovima krvi.

Neki od korištenih testova krvi su količina leukocita, podvrsta leukocita, eritrocita, hemoglobina, željeza, kolesterola i sl. u litri krvi.

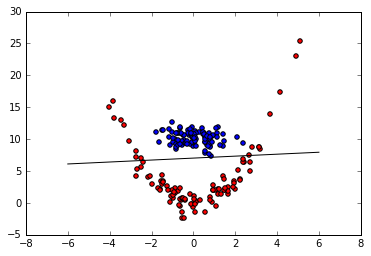
# Klasifikacija

Problem klasifikacije, u statističkom smislu, jest problem prepoznavanja kojom od zadanih i nama poznatih kategorija promatrani uzorak pripada [1]. U strojnom učenju klasifikacija se smatra primjerom nadziranog učenja, tj. tipu učenja kod kojega su poznate izlazne veličine.

Kod nadziranog učenja nama dostupni podaci sastoje se od vektora ulaznih veličina i vrijednosti izlazne veličine (x(i),y(i)) [2]. Svrha ovoga jest naučiti sustav kako na što točniji način odrediti izlaznu vrijednost budućim pristiglim podacima na temelju njihovih ulaznih veličina.

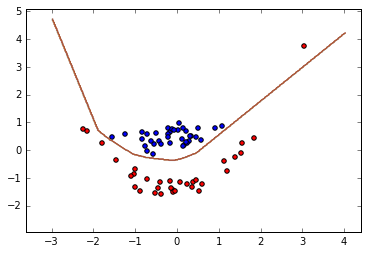
U ovome radu vektor ulaznih veličina sastoji se od numeričkih vrijednosti sveukupno 32 testa, uključujući testove tumorskih markera i rutinske testove krvi. Korištene testove možemo saznati u dokumentu [Abbreviations.docx](https://gitlab.com/PBosnjak/RUSU_projekt/blob/master/Abbreviations.docx) [3].

Jedna od jednostavnijih metoda klasifikacije jest binarna logistička regresija. Cilj je podijeliti promatrani prostor mogućih podataka linearno na dva dijela tako da svi podaci koji se nalaze iznad određene granice imaju jednu izlaznu vrijednost (npr: 1), a podaci koji se nalaze ispod granice drugu vrijednost (npr: 0).



Slika 1 Linearni model logističke regresija

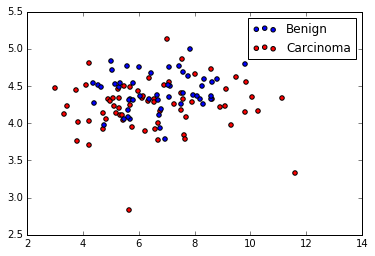
Na slici 1 možemo vidjeti primjer liearnog modela binarne logističke regresije. Plavo obojeni podaci predstavljaju jednu od dvije moguće vrijednosti izlazne varijable (ovdje 1), dok crveni drugu vrijednost (ovdje 0). Cilj nam je podijeliti promatrani prostor pravcem tako da nam se podaci plave boje nalaze sa jedne strane pravca, dok crveni sa suprotne strane. Gledajući sliku, očito je to nemoguće u potpunosti točno napraviti ovakvim pristupom. Potrebno je nelinarno transformirati ulaze, npr. dodati sve kvadratne članove. Na slici 2 možemo vidjeti rezultat.



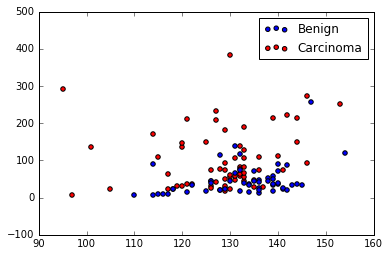
Slika 2 Nelinearna transformacija ulaza

Važno je napomenuti da nam ovakav pristup ne može garantirati 100% točne rezultate, što je vidljivo na slici 2. Problem u većini slučajeva predstavlja nemogućnost linearne podjele dostupnih podataka.

U ovom radu dostupni podaci imaju 32 različita ulaza. Eksplorativnom analizom te vizualnom interpretacijom u 2D prostoru nemoguće je značajno podijeliti podatke u dvije grupe.



Slika 3 WBC-RBC prikaz



Slika 4 HGB-FER prikaz

Na slici 3 uzeti su u obzir količina crvenih i bijelih krvnih zrnaca u litri krvi, dok je na slici 4 uzeta u obzir količna hemoglobina i količina bjelančevine feritina u litri krvi.

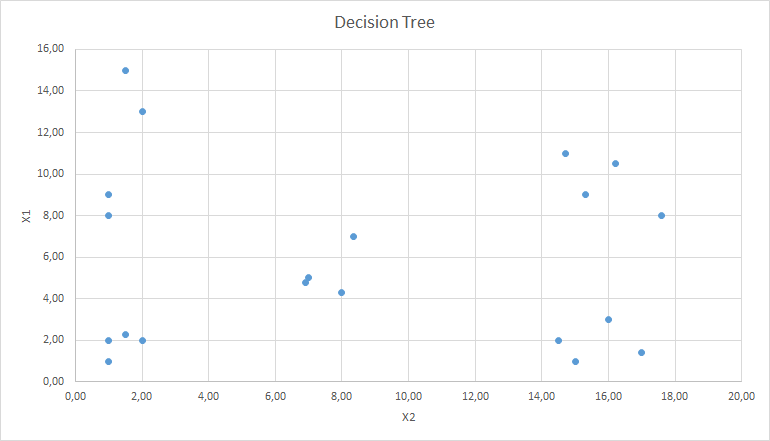
Kada smo suočeni sa ovakvim problemom potrebno je pokušati sa drugim algoritmom učenja. U ovom radu korišten je Random forest algoritam.

# Random forest algoritam

Random forest algoritam je metoda učenja za klasifikaciju, regresiju i druge zadatke koja funkcionira na način da konstruira mnoštvo stabala odlučivanja (*desicion tree*) u vrijeme treniranja te kao rezultat daje najčešće pojavljivanji rezultat (u statistici *mod*) među svim stablima odlučivanja [4]. Potrebno je prvo objasniti princip rada decision tree algoritma.

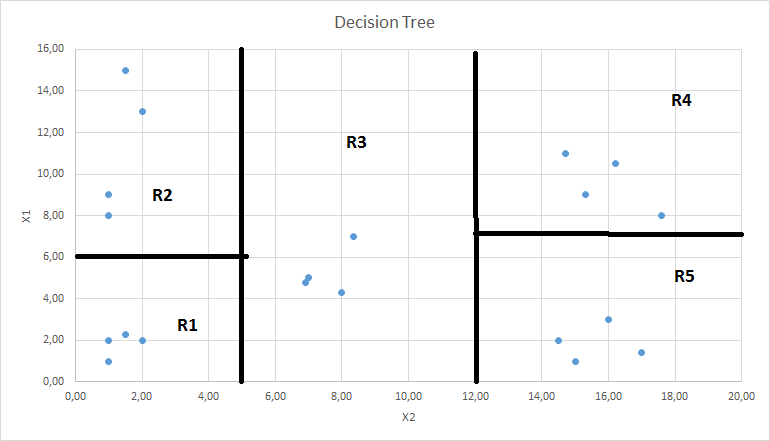
Decision tree algoritam je struktura slična dijagramu toka u kojem svaki unutarnji čvor predstavlja nekakav oblik testa na ulaznom podatku, a svaka grana ishod ispitivanja. [5]

Njegov princip rada najbolje je objasniti na primjeru. Zamislimo da imamo 20 mjerenja sa ulazima u algoritam X1 i X2, te njihov raspored u prostoru:



Slika 5 Raspored podataka u prostoru

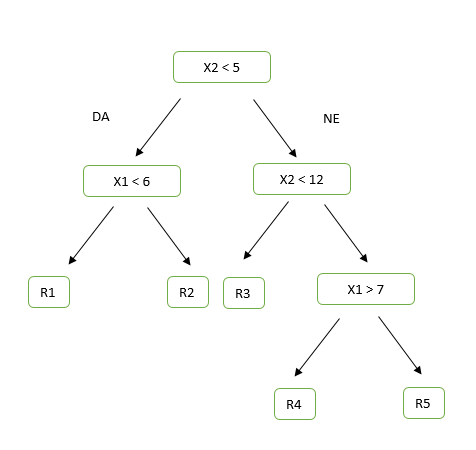
Njih možemo vizualno podijeliti na 5 skupova: R1, R2, R3, R4, R5.



Slika 6 Podjela podataka na 5 kategorija

Sada zamislimo da želimo ispitati kojem skupu pripada neki ulazni podatak koji sadrži proizvoljne koordinate (X2,X1). To ćemo učiniti na način da njegove koordinate ispitamo po pravilima određenim pri podjeli podataka na kategorije. Prvo možemo postaviti pitanje je li X2 koordinata manja od 5. Ukoliko je, ispitujemo je li njegova X1 koordinata manja od 6. Ukoliko je, podatak pripada skupu R1, ukoliko nije, podatak pripada skupu R2.

Na slici 7 možemo vidjeti ostatak dijagrama toka kojim bi mogli odrediti kojem skupu pripada ulazni podatak.



Slika 7 Decision tree dijagram toka

Kao što je već rečeno, random forest algoritam kreira mnoštvo decision tree stabala. Svako stablo prima određeni broj mjerenja te na način na koji funkcionira „glasa“ za jedan od mogućih točnih rješenja (izlaz sustava). Random forest, na principu „većina odlučuje“, uzima kao rješenje algoritma najčešće pojavljivano rješenje svih stabala.

Pretpostavimo da imamo skup podataka S, sačinjen od 3 ulaza (fa, fb i fc), 1 izlaza (C), te N mjerenja:

U kojem fa1 predstavlja 1. ulaz 1. mjerenja, faN 1. ulaz N-tog mjerenja, dok C1 predstalja izlaz sustava 1. mjerenja itd. Od toga skupa radimo M podskupova sa m nasumično odabranih mjerenja (m < M):

Od svakog od tih podskupova S1,S2...SM generiramo jedno stablo odluke. Svako stablo nastavlja sa svojim algoritmom zaključivanja te donosi odluku o rješenju, tj. izlazu podatka kojeg primamo kao ulaz (klasificira). Nakon toga rješenje koje je najpopularnije među svim stablima uzima se kao globalno rješenje algoritma.

Najveće prednosti korištenja ovog algoritma su [6]:

* Mogućnost korištenja za klasifikaciju i regresiju
* Vrlo dobro podnosi mogući nedostatak vrijednosti ulaza
* Dobro sprječava *overfitting*
* Pogodan je za korištenje na velikim višedimenzionalnim *dataset-ovima*

# Predstavljanje problema

Kao što je rečeno u uvodu, cilj ovoga rada je ustanoviti može li se poboljšati dijagnoza tumora dojke kombinirajući testove tumorskih markera cancer antigen Ca 15-3 i carcinoembryonic antigen CEA sa rutinskim testovima krvi. Razlog tome je taj što su navedeni tumorski markeri, kao naprediji oblik neinvazivne metode detekcije tumora, dosta limitirani kod same dijagnoze. Još jedna važna stvar za ispitati jest koliko je potrebno koristiti rutinskih testova krvi, te hoće li se njihovim smanjenjem doći do boljeg rezultata.

Razlog zašto je random forest algoritam odabran pri rješavanju ovog problema je taj što algoritam ima mogućnost određivanja važnosti ulaza (*feature importance*). Baš ta odlika može nam pomoći kod određivanja koji test nam je važan za točniju klasifikaciju.

Za rješavanje problema korišten je programski jezik Python, te njegov *library* scikit-learn [7].

Korišteni skup podataka nalazi se na poveznici [Input.xlsx](https://gitlab.com/PBosnjak/RUSU_projekt/blob/master/Input.xlsx) [8], a rezultati projekta na [Output.xlsx](https://gitlab.com/PBosnjak/RUSU_projekt/blob/master/Output.xlsx) [9]

Za evaluaciju rješenja korišteni su *accuracy, recall i precision*, a za metodu njihovog računanja korištena je cross-validacija. U ovom radu korištena je *10-fold* cross-validacija. Cross-validacija je način podjele skupa podataka na trening i test skup tako da prvo podijelimo cijeli skup na N istih dijelova. Uzmimo da nam je N = 10 u ovom slučaju. Pri prvoj evaluaciji modela koristili bi prvih 90% skupa za treniranje, a ostalih 10% za testiranje modela. Nakon toga bi za treniranje koristili 0-80% i 90-100% skupa, dok bi 80-90% skupa koristili za testiranje. To bismo radili dok ne bi odradili svaku moguću kombinaciju. U ovom slučaju dobili bi kao rješenje vektor od 10 vrijednosti za odabranu evaluaciju modela.

# Rješenje problema

Korišteni *library-ji* su:

**import** numpy **as** np

**import** pandas **as** pd

**import** matplotlib**.**pyplot **as** plt

**from** sklearn**.**ensemble **import** RandomForestClassifier

**from** sklearn**.**model\_selection **import** cross\_val\_score

**from** sklearn**.**preprocessing **import** LabelBinarizer

Za učitavanje podataka korišten je pandas.dataframe. Podaci su se dijelili na matricu ulaza X i vektor izlaza y.

df **=** pd**.**read\_excel**(**'Input.xlsx'**)**

column\_names **=** list**(**df**)**

column\_names**.**pop**()**

X **=** df**.**ix**[:,** 'WBC'**:**'CEA'**].**values

y **=** df**.**ix**[:,** 'Disease'**].**values

Izlaz trenutno sadrži vrijednosti „Carcinoma“ i „Benign“. Zbog potrebe cross-validacije te vrijednosti su binazirizane na sljedeći način: (Carcinoma = 1, Benign = 0). Definirane su potrebne liste za program, te broj estimatora (stabala) koje koristimo u algoritmu. Za potrebe ovog problema korištene količine estimatora su u rasponu od [10,20,30,...300].

lb **=** LabelBinarizer**()**

y **=** np**.**array**([**number**[**0**]** **for** number **in** lb**.**fit\_transform**(**y**)])**

accuracy**,** recall**,** precision**,** result **=** **[],** **[],** **[],** **[]**

i\_begin **=** 10

i\_end **=** 301

step **=** 10

best\_result **=** pd**.**DataFrame**(**columns**=[**'Data'**,** 'Best Score'**,** 'Accuracy'**,** 'Recall'**,** 'Precision'**])**

# Number of trees that are going to be used in cross validation

possible\_estimators **=** np**.**arange**(**i\_begin**,** i\_end**,** step**)**

Evaluacija modela čini se pozivom funkcije *cross\_val\_score* koja radi *fit* na podatke sa odabranim algoritom, računa vrijenost tražene evaluacije (*accuracy, recall, precision...*) te vraća njezinu vrijednost.

**for** i **in** xrange**(**i\_begin**,** i\_end**,** step**):**

clf **=** RandomForestClassifier**(**n\_estimators **=** i**)**

accuracy**.**append**(**cross\_val\_score**(**clf**,** X**,** y**,** cv **=** 10**,** scoring **=** 'accuracy'**).**mean**())**

recall**.**append**(**cross\_val\_score**(**clf**,** X**,** y**,** cv **=** 10**,** scoring **=** 'recall'**).**mean**())**

precision**.**append**(**cross\_val\_score**(**clf**,** X**,** y**,** cv **=** 10**,** scoring **=** 'precision'**).**mean**())**

**print** "1. run: " **+** str**(**i**)** **+** " estimators"

Nakon što su izračunate vrijednosti za *accuracy, recall i precision* za svaki od mogućih broja estimatora, odabiremo konačni broj estimatora ovisno o najboljem rezultatu za evaluaciju modela.

**for** i **in** range**(**0**,** len**(**accuracy**)):**

result**.**append**((**recall**[**i**]** **+** accuracy**[**i**]** **+** precision**[**i**])** **/** 3**)**

# Setting the best result for selected blood tests

best\_result**.**set\_value**(**0**,** 'Data'**,** column\_names**)**

best\_result**.**set\_value**(**0**,** 'Best Score'**,** max**(**result**))**

best\_result**.**set\_value**(**0**,** 'Accuracy'**,** accuracy**[**result**.**index**(**max**(**result**))])**

best\_result**.**set\_value**(**0**,** 'Recall'**,** recall**[**result**.**index**(**max**(**result**))])**

best\_result**.**set\_value**(**0**,** 'Precision'**,** precision**[**result**.**index**(**max**(**result**))])**

# Choosing the most efficient number of trees

num\_of\_estimators **=** possible\_estimators**[**result**.**index**(**max**(**result**))]**

**print** "Selected number of estimators: " **+** str**(**num\_of\_estimators**)**

Nakon toga, radimo *fit* na model sa odabranim brojem estimatora, kako bi mogli odrediti važnost svakog ulaza i poredati ulaze po njihovoj važnosti

clf **=** RandomForestClassifier**(**n\_estimators **=** num\_of\_estimators**)**

clf**.**fit**(**X**,** y**)**

feature\_importn **=** sorted**(**zip**(**column\_names**,** clf**.**feature\_importances\_**),** key**=lambda** tup**:** tup**[**1**],** reverse **=** **True)**

Zatim odabiremo N najvažnijih ulaza (N ∈ [20,10,5,2]) te ponavljamo prijašnji postupak za N odabranih ulaza. Prije toga potrebno je srediti ulaz u sustav na način da kreiramo novu praznu matricu dimenzija (119,1), te nadodajemo potrebne ulaze kao novi stupac u novokreiranu matricu i na kraju obrišemo prvi, prazni stupac.

**for** i **in** **[**20**,** 10**,** 5**,** 2**]:**

# Selecting first N important features

first\_N\_features **=** feature\_importn**[**0**:**i**]**

# Resetting column\_names

column\_names **=** **[]**

**for** j **in** range**(**0**,** len**(**first\_N\_features**)):**

column\_names**.**append**(**first\_N\_features**[**j**][**0**])**

# Selecting new input for cross validation

temp\_array **=** np**.**empty**((**119**,** 1**))**

**for** element **in** first\_N\_features**:**

x **=** df**.**ix**[:,** element**[**0**]].**values**[:,** np**.**newaxis**]**

temp\_array **=** np**.**hstack**((**temp\_array**,** x**))**

# Delete first column of zeroes

X **=** np**.**delete**(**temp\_array**,** 0**,** 1)

Na kraju, poziva se *pandas.to\_excel* funkcija kako bi se rezultat spremio u excel tablicu.

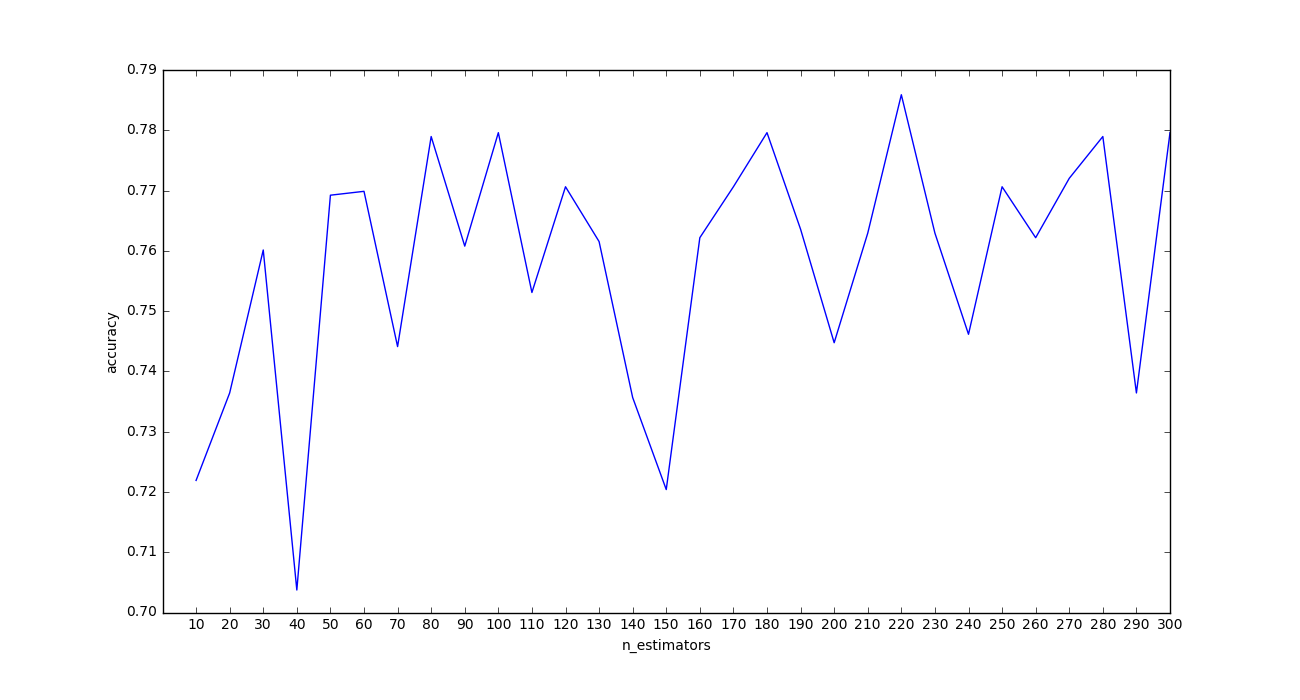
writer **=** pd**.**ExcelWriter**(**'Output.xlsx'**)**

best\_result**.**to\_excel**(**writer**,** 'Sheet1'**)**

writer**.**save**()**

# Zaključak

U folderu [graphs](https://gitlab.com/PBosnjak/RUSU_projekt/tree/master/graphs) možemo vidjeti grafički prikaz rezultata za *accuracy, recall i precision* parametre za svaki testirani broj estimatora i odabranih N značajnih ulaza [10].



Slika 8 10 najznačajnih ulaza

Na slici 8 možemo vidjeti primjer vrijednosti za parametar *accuracy* za svaki od korištenog broja estimatora za 10 najznačajnijih ulaza.

U tablici 1 možemo vidjeti rezultate projekta. Najbolji rezultati zabilježeni su kada se koristilo 10 najznačajnijih ulaza, među kojima su već poznati tumorski markeri Ca153 i CEA, količina feritina, hemoglobina, limfocita, kolesterola, eritrocita itd. No, rezultati rijetko kada premašuju 80% točnosti što ih još uvijek čini neuvjerljivima za pozdanu klasifikaciju uzoraka. Neka od mogućih rješenja jest pribavljivanje većeg skupa uzoraka i kombiniranje različitih algoritama za učenje.

Trenutno, korištenje ovakve medote detekcije tumora je moguće samo kao mjera vjerojatnosti pojave tumora, nikako kao definitivna metoda detekcije.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Data** | **Best Score** | **Accuracy** | **Recall** | **Precision** |
| [u'WBC', u'NEUT', u'LYMPH', u'MONO', u'EO', u'BASO', u'RBC', u'HGB', u'HTC', u'MCV', u'MCH', u'MCHC', u'PLT', u'MPV', u'CRP', u'Fe', u'UIBC', u'TIBC', u'SATTRF', u'TRF', u'FER', u'CHOL', u'TGC', u'HDLC', u'LDLC', u'HDLC\_CHOL', u'LDLC\_HDL', u'ApoAI', u'ApoB', u'ApoB\_ApoA', u'Ca153', u'CEA'] | 0,753659 | 0,726573 | 0,77619 | 0,758214 |
| [u'FER', u'HTC', u'RBC', u'Ca153', u'HGB', u'CEA', u'LYMPH', u'MCH', u'PLT', u'HDLC\_CHOL', u'HDLC', u'LDLC\_HDL', u'NEUT', u'MCHC', u'MPV', u'TGC', u'CRP', u'ApoAI', u'CHOL', u'WBC'] | 0,778049 | 0,742599 | 0,821429 | 0,770119 |
| [u'FER', u'HTC', u'RBC', u'Ca153', u'HGB', u'CEA', u'LYMPH', u'MCH', u'PLT', u'HDLC\_CHOL'] | 0,807487 | 0,762937 | 0,838095 | 0,821429 |
| [u'FER', u'HTC', u'RBC', u'Ca153', u'HGB'] | 0,774815 | 0,745396 | 0,804762 | 0,774286 |
| [u'FER', u'HTC'] | 0,76878 | 0,735664 | 0,807143 | 0,763532 |

Tablica 1 Rezultati mjerenja

# Literatura

[1] https://en.wikipedia.org/wiki/Statistical\_classification

[2] https://loomen.carnet.hr/mod/resource/view.php?id=204456

[3] https://gitlab.com/PBosnjak/RUSU\_projekt/blob/master/Abbreviations.docx

[4] https://en.wikipedia.org/wiki/Random\_forest

[5] https://en.wikipedia.org/wiki/Decision\_tree

[6] https://www.stat.berkeley.edu/~breiman/RandomForests/cc\_home.htm

[7] http://scikit-learn.org/stable/

[8] https://gitlab.com/PBosnjak/RUSU\_projekt/blob/master/Input.xlsx

[9] https://gitlab.com/PBosnjak/RUSU\_projekt/blob/master/Output.xlsx

[10] https://gitlab.com/PBosnjak/RUSU\_projekt/tree/master/graphs