# Prepoznavanje vrste kancera metodom klasifikacije

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

Nadežda Bogdanović
1093/2018
nadezdabogdanovic1@gmail.com

19.06.2019

**Sažetak**

Cilj ovog rada je da potvrdi da li je moguće na osnovu podataka o mutiraim genima utvrditi koji je organ oboleo. Prilikom odgovora na ovo pitanje korišćene su različite metode klasifikacije obrađene u programskom jeziku Python, ili IBM-ovom alatu SPSS Modeler.

# Sadržaj

# 1 Uvod

Kada zdrava ćelija postane kancerogena, dese se promene u njenom genetskom materijalu - on se izmeni. Postavlja se pitanje: Da li je moguće utvrditi koje tkivo je kontaminirano na osnovu podataka dobijenih sekvenciranjem i upoređivanja sa referentnim genomom? Ovo istraživanje daje odgovor na takvo pitanje.

U drugom poglavlju čitalac se može upoznati sa podacima[2], koji su dobijeni kao rezultat rada alata ABSOLUTE[1]. Ovaj alat obrađuje podatke dobijene sekvenciranjem kancerogenih tkiva, upoređuje ih sa referentnim genomom i broji duplikacije i delecije koje su se desile u kontaminiranom genomu.

Primena klasifikacije na sirove podatke nije moguća. Treće poglavlje objašnjava kako je potrebno obraditi podatke, kako bi se dobili što bolji rezultati.

Četvrto, peto, šesto sedmo i osmo poglavlje upoređuju rezultate primenom različitih algoritama klasifikacije (drveta odlučivanja, najbliži susedi, neuronske mreže, potporni vektori i Gausova klasifikacija), obrađivanih u programskom jeziku Python, ili u SPSS Modeleru. Kodovi koji su predstavljeni u ovom radu preuzeti su sajta asistenta ovog predmeta[5], Mirjane Maljković, i izmenjeni u skladu sa potrebom.

# 2 Upoznavanje sa podacima

Podaci se mogu pronaći na https://portals.broadinstitute.org/ccle/data pod nazivom **CCLE_ABSOLUTE_combined 20181227.xlsx**. U okviru fajla postoje 3 lista:

- **ABSOLUTE_combined.segtab** - sadrži podatke nad kojima se vrši istraživanje: 20 kolona i 188653 redova
- **segtab annotations** - sadrži objašnjenje naziva atribta sa prvog lista
  - Sample: naziv ćelijske linije sa koje je uzet uzorak. Sastoji se od naziva tkiva koje je sekvencirano i jedinstvenog identifikatora.
  - Chromosome: hromozom na kojem se nalazi mutirani segment.
  - Start: početna pozicija mutiranog segmenta na datom hromozomu.
  - End: krajnja pozicija mutiranog segmenta na datom hromozomu.
  - Num_Probes: broj SNP satelita (sateliti su markeri koji se koriste prilikom sekvencirnja) na datom segnemtu[3].
  - Length : Dužina mutiranog segmenta.
  - Modal_HSCN_1: broj apsolutnih duplikacija/delecija na alternativnom alelu.
  - Modal_HSCN_2: broj apsolutnih duplikacija/delecija na referentnom alelu.
  - Modal_HSCN_TOTAL: Modal_HSCN_1 + Modal_HSCN_2.
  - Subclonal_HSCN_a1: broj apsolutnih duplikacija/delecija na alternativnom alelu u subklonalnoj populaciji (populacija mutiranih kancerogenih ćelija koja je mutirala od početne kancerogene ćelije)[4].
  - Subclonal_HSCN_a1: broj apsolutnih duplikacija/delecija na referentnom alelu u subklonalnoj populaciji.
  - Cancer_cell_frac_a1: maksimalna procena za udeo ćelija koje nose mutaciju na prvom alelu.
  - Ccf_ci95_low_a1: donja granica 95% intervala poverenja za udeo ćelija koje nose mutaciju na prvom alelu.
  - Ccf_ci95_high_a1: gornja granica 95% intervala poverenja za udeo ćelija koje nose mutaciju na prvom alelu.
  - ancer_cell_frac_a2: maksimalna procena za udeo ćelija koje nose mutaciju na drugom alelu.
  - Ccf_ci95_low_a2: donja granica 95% intervala poverenja za udeo ćelija koje nose mutaciju na drugom alelu.
  - Ccf_ci95_high_a2: gornja granica 95% intervala poverenja za udeo ćelija koje nose mutaciju na drugom alelu.
  - LOH: gubitak heterozigosnosti na datom segmentu.
  - Homozygos_deletion: oznaka da li je dati segment homozigotno obrisan ili ne.
  - depMapID: jedinstveni ID uzorka, koji se koristi za povezivanje sa drugim fajlovima.
- **ABSOLUTE_combined.table**: sadrži opise podataka sa prvog lista

.

# 3 Priprema podataka za obradu

Ako se algoritam drveta odlučivanja pisanog u programskom jeziku Python primeni na neobrađene podatke, pogađanje klase odvija se sa preciznošću od 0,5%. To se dešava iz nekoliko razloga koji će u daljem tekstu biti razmotreni.

## 3.1 Problem prevelikog broja klasa

Kada pokušamo da odredimo koje su sve moguće klase, videćemo da se u koloni *Sample* nalazi 999 različitih vrednosti, samim tim i 999 različitih klasa. To je zato što se vrednosti u ovoj koloni sastoje od naziva tkiva koje je sekvencirano i jedinstvenog identifikatora. Kako nas zanima samo koji je organ oboleo, neophodno je grupisati sve podatke iz istog tkiva.

Koristeći SPSS Modeler i čvor *Split* (fajl *find_ most_ frequent.str*), možemo izdvojiti tkiva po uslovu *hassubstring*, što nam daje 24 različite klase sa brojem instanci koje im pripadaju:

- LUNG: 38,882
- SALIVARY_GLAND: 358
- FIBROBLAST: 1,164
- PLEURA: 1,529
- THYROID: 2,824
- PANCREAS: 6,943
- BONE: 3,579
- STOMACH: 8,720
- UPPER_AERODIGESTIVE: 5,970
- BREAST: 15,488
- CENTRAL_NERVOUS_SYSTEM: 12,357
- PROSTATE: 1,356
- INTESTINE: 9,369
- SOFT_TISSUE: 3,475
- ENDOMETRIUM: 5,277
- AUTNOMIC_GANGLIA: 2,115
- HAEMATOPOIETIC_AND_LYMPHOID_TISSUE: 27,115
- KIDNEY: 4,171
- BILIARY_TRACT: 1,462
- OESOPHAGUS: 7,798
- OVARY: 11,584
- URINARY_TRACT: 4,073
- LIVER: 4,718
- SKIN: 8,384

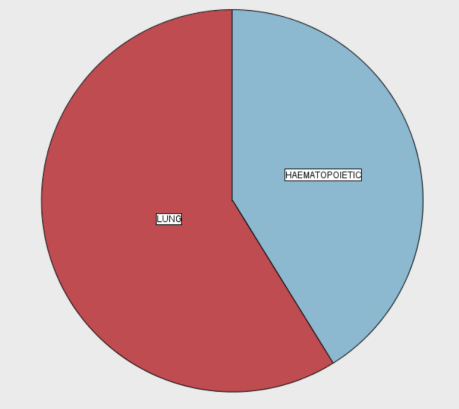
Ako algoritam primenimo sada, dobićemo preciznosti od 5%. To nas dovodi do zaključka da je nedovoljan broj instanci po klasi, što znači da ćemo najpreciznija pogađanja dobiti ako odaberemo da radimo samo s klasama sa najvećim brojem instanci: LUNG i HAEMATOPOIETIC_AND

LYMPHOID_TISSUE. Koristeći čvor *Append* (fajl *MERGE.str*), objedinićemo podatke sa oznakama ove dve klase u poseban *EXTRACTED CLASSES.xlsx* fajl. Međutim, on i dalje sadrži različite oznake za ista tkiva, tako da ćemo sve one koji sadrže podstring *LU* predstaviti kao *LUNG*, a one koji sadrže *HA*, predstaviti kao *HAEMATOPOIETIC* (fajl *create_finals.py*) i sačuvati u fajlu *final_X.xlsx* sa kojim ćemo nadalje raditi, gde umesto X može stajati implicit ili explicit, što zavisi od korišćene metode objašnjene u poglavlju 3.3.

```
i = 0
for item in y:
    if 'LU' in item:
        df["sample"][i] = 'LUNG'
    elif 'HA' in item:
        df["sample"][i] = 'HAEMATOPOIETIC'
    else:
        df = df.drop([i])
    i = i+1
    print(i)
```

Listing 1: Prečišćavanje klasa

Dobijene klase predstavljene su na slici 1:




Slika 1: Klase

## 3.2 Problem nedostajućih vrednosti

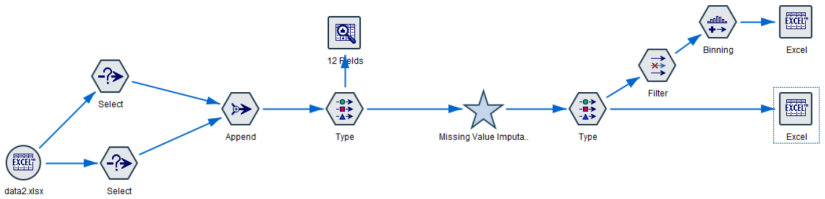
Pre nego što spojimo željene klase u odgovarajući fajl, proverićemo da li postoje nedostajuće vrednosti, ili elementi van granica. Ako pokrenemo čvor *Data Audit* i u njemu otvorimo karticu *Quality*, videćemo da su podaci 98.47% kompletni. Sa nedostajućim vrednostima se nosimo tako što generišemo super čvor u kojem nedostajuće vrednosti zamenjujemo srednjim vrednostima.

Atributi Start, End, Num_Probes, Length, Modal_HSCN_1 i Modal_HSCN_2 sadrže ekstremne vrednosti i vrednosti van granica. Sa atributima Start, End i Num_Probes ćemo se pozabaviti u odeljku 3.3, a sa Length, Modal_HSCN_1 i Modal_HSCN_2 u odeljku 3.4.

## 3.3 Problem definisanja mutiranih gena

Kao što je već rečeno, pokušavamo da dokažemo da na osnovu mutiranih (grupa) gena možemo da odredimo koje tkivo je obolelo. Iako u ovim podacima ne nalazimo eksplicitne oznake gena, ipak ih možemo razlikovati na osnovu njihove lokacije koja je sačinjena od početka i kraja mutiranog segmenta koji se nalaze na određenom hromozomu. Međutim, razlika između najmanjih i najvećih vrednosti u ovim atributima, kao i u Num_Probes je tako velika (24,9087,712 za Start, 249,133,375 za End i 72,607 za Num_Probes), da ih je teško pratiti, a pored toga, javljaju se i ekstremne vrednosti i elementi van granica. Postoje 2 načina da se ovaj problem reši i ovo istraživanje će ispratiti oba:

1. Implicitno definianje: Koristeći SPSS Modelerov čvor *Binning*, vrednostima gorenavedenih atributa su dodeljene oznake brojevnih opsega kojima pripadaju, kao što je prikazano na slici 2. Vrednosti se grupišu tako da u svakoj grupi bude jednak broj vrednosti (*TILES*) po decilima.




Slika 2: Implicitan i eksplicitan način

2. Eksplicitno definisanje gena: Podaci se u SPSS Modeleru pomoću čvora *Split* razvrstaju po hromozomima i čuvaju u posebnim fajlovima (*chromosome_pp.str*). Zatim se, kao što je prikazano na kodu, svakoj instanci dodeljuju oznake genskih grupa koje su zahvaćene mutacijom (*Genes.py*). Nakon toga se obrađeni fajlovi objedinjuju u jedan (*EXCTRACTED_CLASSES_chromosome.xlsx*) pomoću čvora *Append* u SPSS Modeleru (*merge_chromosome.str*). Konačno, na taj fajl se primeni već prikazani algoritam prečišćavanja vrednosti atributa *sample*.

```
range_index = int(df['Length'].mean())
df = df.sort_values("Start")
df = df.reset_index(drop=True)

pd.options.mode.chained_assignment = None
# g = 0 for the first file
# for the rest of them, g is appended
g = g+1
df.iloc[0, df.columns.get_loc('depMapID')] = "g" + str(g)
start_old = df["depMapID"][0]
n = len(df.index)
for i in range(1,n):
    start_new = df["depMapID"][i]
    if start_new != start_old and not (start_new < start_old
     + range_index):
        g = g + 1
        start_old = start_new
    df.iloc[i, df.columns.get_loc('depMapID')] = "g" + str(g)
```

Listing 2: Definisanje klasa

## 3.4 Problem korelisanih atributa

Pozivanjem *dataframe.corr()['atribut']* u programskom jeziku Python nad skupom podataka, izračunate su sledeće stope korelacije i prikazane u tabelama 1, 2, 3, 4, 5:

Tabela 1: Korelacija između atributa

| | **Chromosome** | **Start** | **End** | **Gene** |
|---|---|---|---|---|
| **Chromosome** | 1.0 | x | x | 0.99 |
| **Start** | x | 1.0 | 0.93 | x |
| **End** | x | 0.93 | 1.0 | x |
| **Gene** | 0.99 | x | x | 1.0 |

Tabela 2: Korelacija između atributa

| | **Sample** | **Num Probes** | **Length** | **depMapID** |
|---|---|---|---|---|
| **Sample** | 1.0 | x | x | 1.0 |
| **Num Probes** | x | 1.0 | 0.98 | x |
| **Length** | x | 0.98 | 1.0 | x |
| **depMapID** | 1.0 | x | x | 1.0 |

Tabela 3: Korelacija između atributa

| | **Modal 1** | **Modal 2** | **Modal Total** | **LOH** |
|---|---|---|---|---|
| **Modal 1** | 1.0 | x | x | -0.82 |
| **Modal 2** | x | 1.0 | 0.88 | x |
| **Modal Total** | x | 0.88 | 1.0 | x |
| **LOH** | -0.82 | x | x | 1.0 |

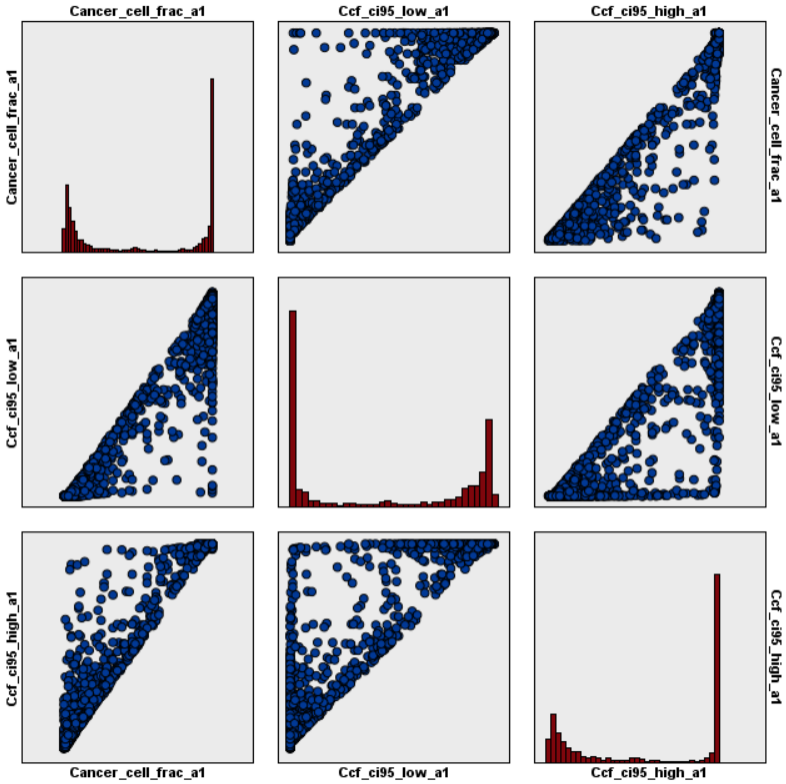
Tabela 4: Korelacija između atributa

| | **Cancer Frac a1** | **Ccf low a1** | **Ccf high a1** |
|---|---|---|---|
| **Cancer Frac a1** | 1.0 | 0.97 | 0.98 |
| **Ccf low a1** | 0.97 | 1.0 | 0.94 |
| **Ccf high a1** | 0.98 | 0.94 | 1.0 |

Tabela 5: Korelacija između atributa

| | **Cancer Frac a2** | **Ccf low a2** | **Ccf high a2** |
|---|---|---|---|
| **Cancer Frac a2** | 1.0 | 0.97 | 0.97 |
| **Ccf low a2** | 0.97 | 1.0 | 0.92 |
| **Ccf high a2** | 0.97 | 0.92 | 1.0 |

Za metodu koja koristi grupisanje postoje još i dodatne korelacije prikazane u tabelama 6 i 7:




Slika 3: Korelacija atributa Cancer_Frac_a1, Ccf_low_a1, Ccf_high_a1

Tabela 6: Korelacija između atributa

| | **Start** | **End** | **Start TILE10** | **End TILE10** |
|---|---|---|---|---|
| **Start** | 1.0 | 0.92 | 0.94 | 0.87 |
| **End** | 0.92 | 1.0 | 0.86 | 0.95 |
| **Start TILE10** | 0.94 | 0.86 | 1.0 | 0.89 |
| **End TILE10** | 0.87 | 0.95 | 0.89 | 1.0 |

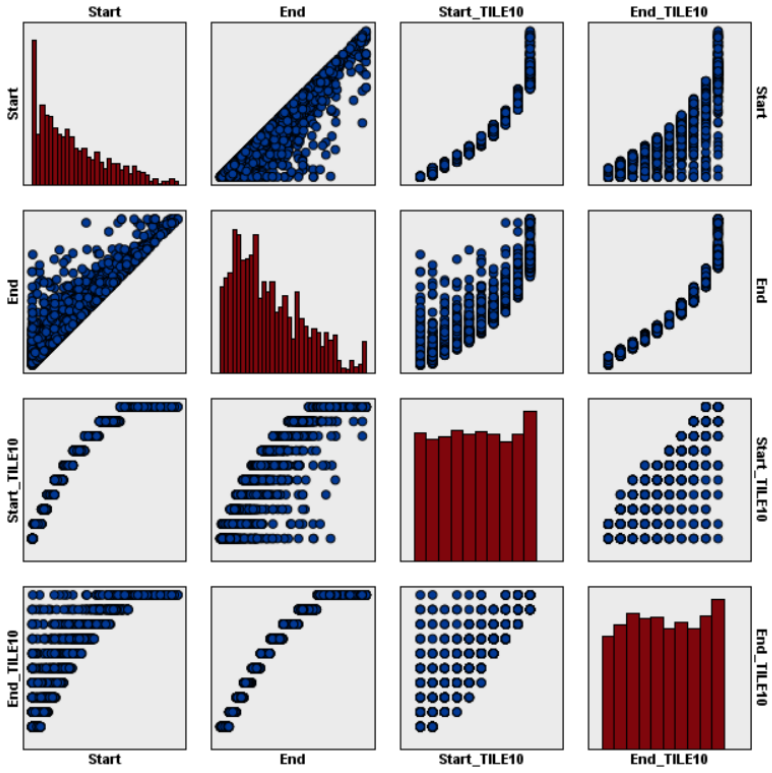
Ovo je grafički prikazano na slici 4.

Tabela 7: Korelacija između atributa

| | **Num Probes** | **Num Probes TILE10** |
|---|---|---|
| **Num Probes** | 1.0 | 0.75 |
| **Num Probes TILE10** | 0.75 | 1.0 |

Ovo je grafički prikazano na slici 5.

Na osnovu predloženog, odabrani su atributi koji će se koristiti za modeliranje: Start, Num Probes, Modal HSCn 1, Modal HSCN 2, Cancer cell frac a1, Cancer cell frac a2, Homozygos deletion, Gene; a za metodu




Slika 4: Korelacija atributa Start, End, Start_TILE10 i End_TILE10

koja koristi grupisanje Chromosome, Start TILE10, Num Probes TILE10, Modal HSCn 1, Modal HSCN 2, Cancer cell frac a1, Cancer cell frac a2, Homozygos deletion.

# 4 Drveta odlučivanja

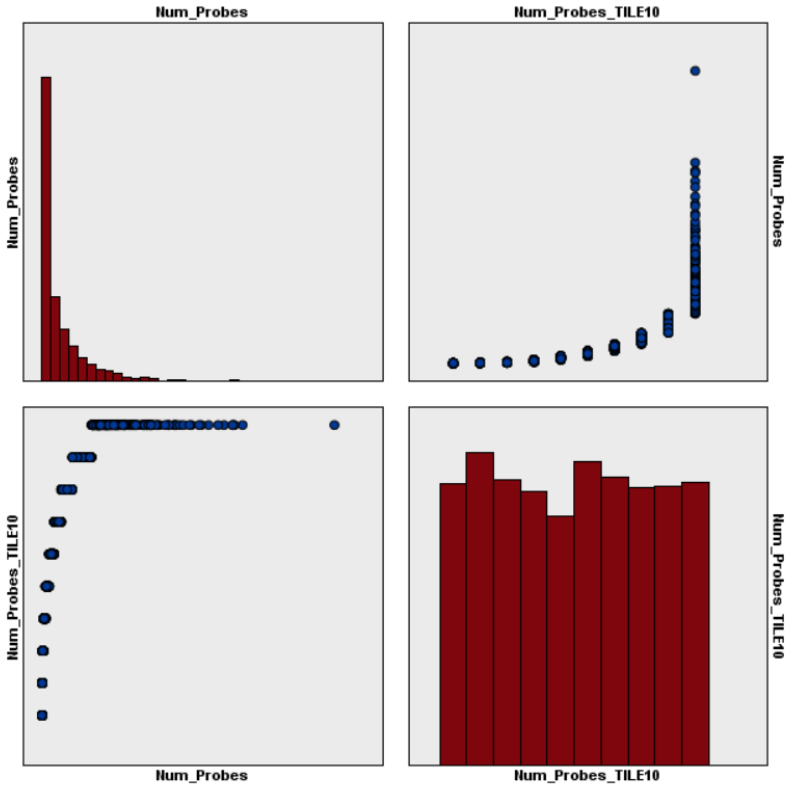
## 4.1 SPSS Modeler

U nastavku će biti upoređeni rezultati primene algoritama C5.0 (u fajlu *C50.str*) i C&R (u fajlu *CRT.str*). U čvoru *Partition* se vrši podela na traning, test i validacioni skup uzimajući 70%, 20% i 10% podataka.

### 4.1.1 C5.0

Čvor *C.50* se spaja sa *Partition* i to sa sledećeim opcijama: *Group symbolics, Use boosting i Cross-validate.* Pokretanjem dobija se model koji analiziramo pomoću čvora *Analyze.* Rezultati dobijeni implicitnim i eksplicitnim grupisanjem, mogu se uporediti na sledeći način:

- eksplicitno
  - Dubina drveta: 17
  - Standardna greška: 0.2
  - Sredina: 65.2




Slika 5: Korelacija atributa Num_Probes i Num_Probes_TILE10

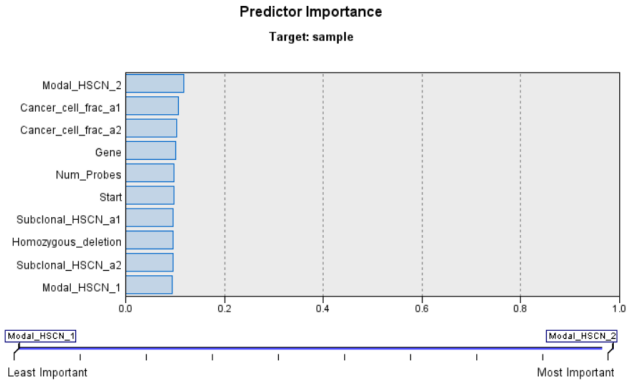
  - Najbitniji atribut za odlučivanje (slika 6): Modal HSCN 2
  - Najmanje bitan atribut za odlučivanje (slika 6): Modal HSCN1
  - preciznost je prikazana na slici 8:
- implicitno
  - Dubina drveta: 23
  - Sredina 70.5
  - Standardna greška: 0.2
  - Najbitniji atribut za odlučivanje (slika 7): Modal HSCN 2
  - Najmanje bitan atribut za odlučivanje (slika 7): Subclonal HSCN a2
  - preciznost je prikazana na slici 9:

### 4.1.2 C&RT

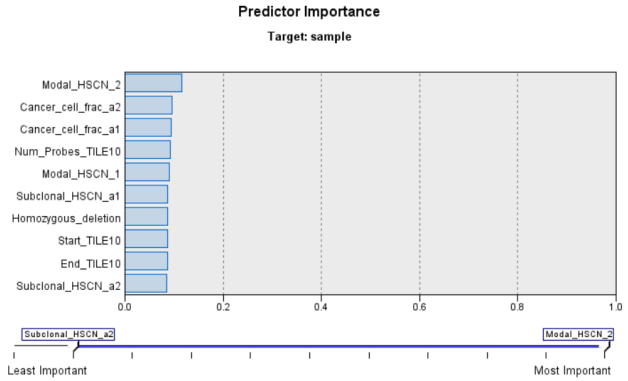
Model se pravi pomoću čvora *C&RT* nad *final_ implicit.xlsx* podacima. Cilj je izgraditi novi model u vidu drveta odlučivanja maksimalne dubine 12. Minimalan broj instanci u grani roditelja je 5%, a u grani deteta 2%. Kao mera nečistoće koristi se Ginijev kriterijum i minimalnom promenom u nečistoći od 0.0001%.

Analiza dobijenih rezultata može se videti na slici 10.

Na slici 11 je prikazano drvo odlučivanja dobijeno generisanjem modela.




Slika 6: Bitnost atributa pri eksplicitnom grupisanju - C5.0




Slika 7: Bitnost atributa pri implicitnom grupisanju - C5.0

## 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 30% i prethodno je izvršeno mešanje podataka.

```
features1 = df.columns[1:2].tolist()
features2 = df.columns[6:].tolist()
features = features1 + features2

x = df[features]
y = df["sample"]

x_train, x_test, y_train, y_test = train_test_split(
    x, y, test_size = 0.3, shuffle=True)

dt = tree.DecisionTreeClassifier(max_depth=12)
dt.fit(x_train, y_train)
y_pred = dt.predict(x_test)
```

Listing 3: Drvo odlučivanja

Drvo ima maksimalnu dubinu 12, za kriterijum podele koristi se Ginijev kriterijum i ostvareni su sledeći rezultati:

| 'Partition' | 1_Training | | 2_Testing | | 3_Validation | |
|---|---|---|---|---|---|---|
| Correct | 37,406 | 72.53% | 9,483 | 65.08% | 4,933 | 65.39% |
| Wrong | 14,166 | 27.47% | 5,088 | 34.92% | 2,611 | 34.61% |
| Total | 51,572 | | 14,571 | | 7,544 | |

Slika 8: Analiza dobijenog modela pri eksplicitnom grupisanju - C5.0

| 'Partition' | 1_Training | | 2_Testing | | 3_Validation | |
|---|---|---|---|---|---|---|
| Correct | 33,265 | 72.09% | 9,281 | 71.28% | 4,770 | 70.43% |
| Wrong | 12,879 | 27.91% | 3,739 | 28.72% | 2,003 | 29.57% |
| Total | 46,144 | | 13,020 | | 6,773 | |

Slika 9: Analiza dobijenog modela pri implicitnom grupisanju - C5.0

| 'Partition' | 1_Training | | 2_Testing | | 3_Validation | |
|---|---|---|---|---|---|---|
| Correct | 32,266 | 69.92% | 9,138 | 70.18% | 4,731 | 69.85% |
| Wrong | 13,878 | 30.08% | 3,882 | 29.82% | 2,042 | 30.15% |
| Total | 46,144 | | 13,020 | | 6,773 | |

Slika 10: Analiza drveta odlučivanja

- Eksplicitno grupisanje je predstavljeno u tabelama 8 8 i 9:

Tabela 8: Analiza drveta odlučivanja - eksplicitno grupisanja

| | **preciznost** | **f1-skor** | **tačnost** |
|---|---|---|---|
| **LUNG** | 64.00% | 66.00% | 63.45% |
| **HAEMATOPOIETIC** | 62.00% | 59.00% | 63.45% |

Tabela 9: Matrica konfuzije - eksplicitno grupisanje

| | **HAEPATOPOTETIC** | **LUNG** |
|---|---|---|
| **HAEMATOPOIETIC** | 5,591 | 4,482 |
| **LUNG** | 3,102 | 8,572% |

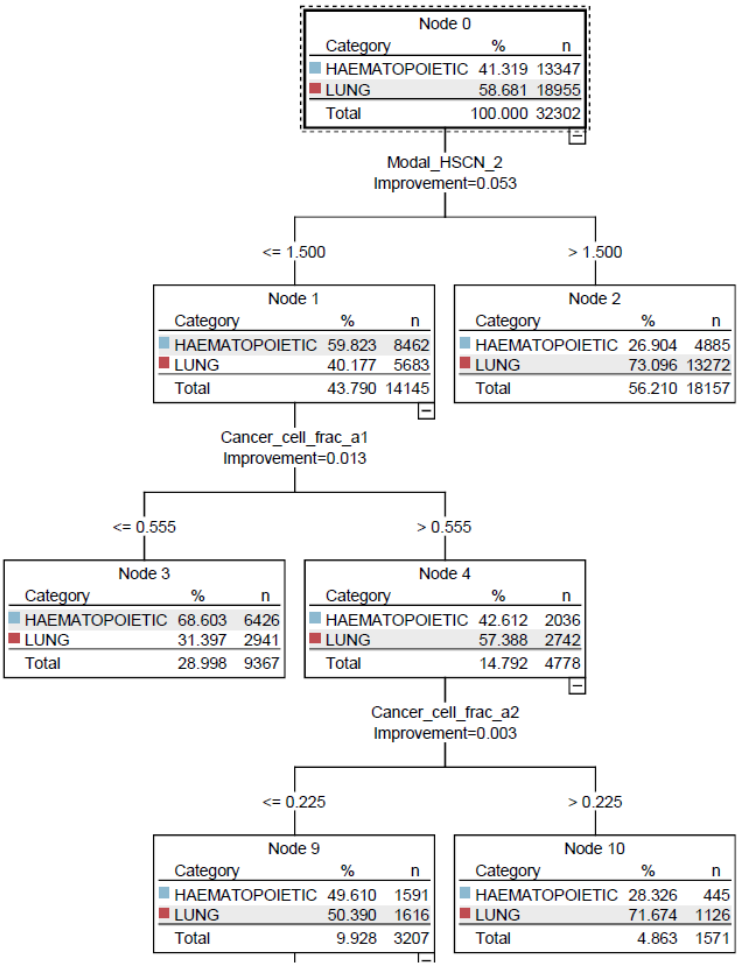
- Implicitno grupisanje je predstavljeno u tabelama 10 i 11:

Tabela 10: Analiza drveta odlučivanja - implicitno grupisanje

| | **preciznost** | **f1-skor** | **tačnost** |
|---|---|---|---|
| **LUNG** | 73.00% | 74.00% | 68.42% |
| **HAEMATOPOIETIC** | 63.00% | 61.00% | 68.42% |

# 5 Najbliži susedi

Podaci koji se nalaze u *final_ implicit.xlsx* će biti obrađeni metodom k najbližih suseda u SPSS Modeleru i programskom jeziku Python.




Slika 11: Drvo odlučivanja - C&RT

Tabela 11: Matrica konfuzije - implicitno grupisanje

| | **HAEMATOPOIETIC** | **LUNG** |
|---|---|---|
| **HAEMATOPOIETIC** | 4,838 | 3,243 |
| **LUNG** | 3,012 | 8,689 |

## 5.1 SPSS Modeler

U fajlu *KNN.str* učitavamo podatke 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. Definisani cilj je predviđanje klase, tako da analiza bude balansirana, brza i tačna. Ciljno polje je označeno kao *sample*, a ostala polja su ulazna. Minimalan broj k je postavljen na 3, maksimalan na 5,a udaljenost se računa Euklidskim rastojanjem.

Pogledom na generisani model, primećuje se da se najmanja greška

dostiže za k = 5 i to sa vrednostima prikazanim na slici 12

| 'Partition' | 1_Training | | 2_Testing | | 3_Validation | |
|---|---|---|---|---|---|---|
| Correct | 35,662 | 77.28% | 10,045 | 77.15% | 5,080 | 75% |
| Wrong | 10,482 | 22.72% | 2,975 | 22.85% | 1,693 | 25% |
| Total | 46,144 | | 13,020 | | 6,773 | |

Slika 12: Analiza modela k najbližih suseda - SPSS

## 5.2 Python

Primena KNN algoritma je opisana u fajlu *KNN.py*. Veličina trening skupa je postavljena na 95% i izvršena je stratifikacija.

```
features1 = df.columns[1:2].tolist()
features2 = df.columns[6:].tolist()
features = features1 + features2

x_original=df[features]
x=pd.DataFrame(prep.MinMaxScaler().fit_transform(x_original))

x.columns = features
y=df["sample"]

x_train, x_test, y_train, y_test = train_test_split(x, y,
     train_size=0.95, stratify=y)

k_values = range(3,10)
p_values = [1, 2]
weights_values = ['uniform', 'distance']

for k in k_values:
    for p in p_values:
        for weight in weights_values:
            clf = KNeighborsClassifier(n_neighbors=k,
                                       p=p,
                                       weights=weight)

            print("k="+ str(k))
            print("p="+str(p))
            print("weight=" + weight)

            class_info(clf, x_train, y_train, x_test, y_test)
```

Listing 4: Najbliži susedi

Najbolji rezultati dobijaju se za k = 9, Euklidsko rastojanje i kada svi susedi imaju podjednak uticaj, što se vidi u tabeli 14.

Tabela 12: Analiza modela k najbližih suseda - Python

| | **preciznost** | **f1-skor** | **tačnost** |
|---|---|---|---|
| **LUNG** | 72.00% | 74.00% | 68.51% |
| **HAEMATOPOIETIC** | 63.00% | 60.00% | 68.51% |

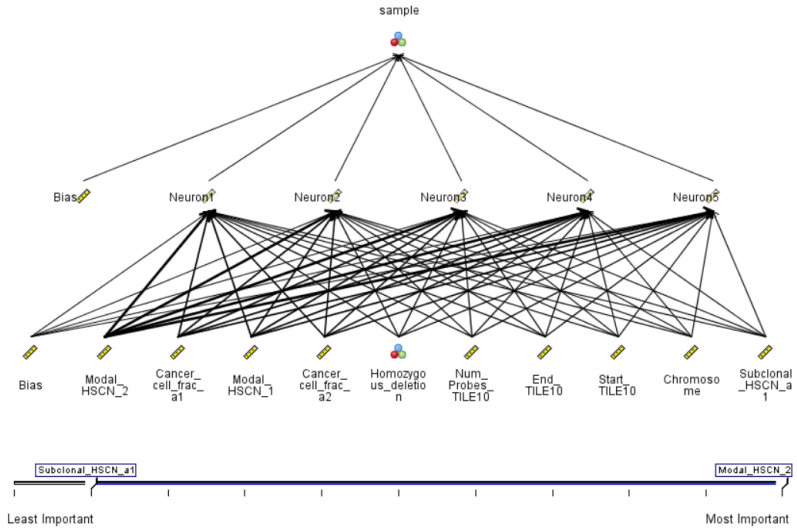
# 6 Neuronske mreže

Podaci koji se nalaze u *final_ implicit.xlsx* će biti obrađeni metodom neuronskih mreža u SPSS Modeleru i programskom jeziku Python.

## 6.1 SPSS Modeler

U fajlu *Neurone.str* učitavamo podatke 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 *Partition*. Za cilj je odabrano kreiranje novog višeslojnog modela.

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 13: Neuronska mreža

| 'Partition' | 1_Training | | 2_Testing | | 3_Validation | |
|---|---|---|---|---|---|---|
| Correct | 32,180 | 69.74% | 9,140 | 70.2% | 4,737 | 69.94% |
| Wrong | 13,964 | 30.26% | 3,880 | 29.8% | 2,036 | 30.06% |
| Total | 46,144 | | 13,020 | | 6,773 | |

Slika 14: Analiza modela neuronske mreže - SPSS

## 6.2 Python

Ovaj metod je primenjen na podatke u fajlu *final_ implicit.xlsx*, a opisan je u *MLP.py* fajlu, čiji je jedan deo prikazan u Listingu. Nad podacima je izvršena strafifikacija i za trening skup uzeto je 70% podataka.

```
1000 x=df[features]
     x.columns = features
1002 y=df["sample"]

1004 scaler = preprocessing.StandardScaler().fit(x)
     x =pd.DataFrame(scaler.transform(x))
1006 x.columns = features

1008 x_train, x_test, y_train, y_test = train_test_split(x, y,
         train_size=0.7, stratify=y)

1010 params = [{'solver':['sgd'],
```

```
            'learning_rate':['constant', 'invscaling', 'adaptive'],
            'learning_rate_init':[0.01, 0.005, 0.002, 0.001],
             'activation' : ['identity', 'logistic', 'tanh', 'relu'
    ],
             'hidden_layer_sizes' : [(10,3), (10,10),],
            'max_iter': [500]
            }]

clf = GridSearchCV(MLPClassifier(), params, cv=5)
clf.fit(x_train, y_train)
```

Listing 5: Neuronske mreže

Izvršeno je 47 iteracija i to sa 4 sloja. Analiza modela je predstavljena u tabeli

Tabela 13: Analiza MLP modela - Python

| | **preciznost** | **f1-skor** | **tačnost** |
|---|---|---|---|
| **LUNG** | 71.00% | 76.00% | 69.74% |
| **HAEMATOPOIETIC** | 67.00% | 59.00% | 69.74% |

# 7 Metod potpornih vektora

Ovaj metod je primenjen na podatke u fajlu *final_ implicit.xlsx*, a opisan je u *SVM.str* fajlu u kojem najpre učitavamo podatke, a zatim na njima primenjujemo PCA modeliranje koristeći *PCA* čvor, kako bismo odabrali 5 najznačajnijih komponenti i smanjili skup atributa sa kojima se radi. Zatim se vrši podela podataka na trening, test i validacioni skup, nakon čega se pravi model potpornih vektora pomoću čvora *SVM*. Analiza dobijenog modela predstavljena je na slici 15.

| 'Partition' | 1_Training | | 2_Testing | | 3_Validation | |
|---|---|---|---|---|---|---|
| Correct | 31,323 | 67.88% | 8,843 | 67.92% | 4,598 | 67.89% |
| Wrong | 14,821 | 32.12% | 4,177 | 32.08% | 2,175 | 32.11% |
| Total | 46,144 | | 13,020 | | 6,773 | |

Slika 15: Analiza modela potpornih vektora

# 8 Gausova klasifikacija

Ovaj metod je primenjen na podatke u fajlu *final_ implicit.xlsx*, a opisan je u *Gaussian.py* fajlu, čiji je jedan deo prikazan u Listingu. Nad podacima je izvršena strafifikacija i za trening skup uzeto je 70% podataka.

Analiza podataka prikazana je u tabeli

Tabela 14: Analiza modela dobijenog Gausovom klasifikacijom

| | **preciznost** | **f1-skor** | **tačnost** |
|---|---|---|---|
| **LUNG** | 58.00% | 60.00% | 65.74% |
| **HAEMATOPOIETIC** | 72.00% | 70.00% | 65.74% |

```
features1 = df.columns[1:2].tolist()
features2 = df.columns[6:].tolist()
features = features1 + features2

x = df[features]
y = df["sample"]

x_train, x_test, y_train, y_test = train_test_split(x, y,
    train_size=0.7, stratify=y)

clf_gnb = GaussianNB()

clf_gnb.fit(x_train, y_train)

y_pred = clf_gnb.predict(x_test)
```

Listing 6: Gausova klasifikacija

# 9 Zaključak

Kako bi algoritmi klasifikacije bili u stanju da daju što preciznije rezultate, neophodno je prethodno ih pripremiti za klasifikaciju. Utvrđeno je da se nad datim skupom podataka najbolji rezultati postižu prilikom rada sa dve klase sa najvećim brojem instanci. Dodatno, potrebno je izvršiti grupisanje nad instancama koje imaju veliku razliku između minimalne i maksimalne vrednosti.

S obzirom na to da početak i kraj segmenta baznih parova na određenom hromozomu definiše njima odgovarajuću grupu gena, očekivano je da su samo te iste informacije dovoljne da se utvrdi koje je kontaminirano tkivo u pitanju. Međutim, utvrđeno je da algoritmi klasifikacije nisu u stanju da kvalitetno razluče koje tkivo je obolelo samo na osnovu datih podataka. Razlog leži u činjenici da različite vrste kancera imaju mutirane iste gene, te se zajedno pojavljuju u organizmu. Jedan od najboljih primera su rak dojke i jajnika: pacijenti sa rakom dojke imaju mutirane gene bitne za normalno funkcionisanje jajnika i obrnuto. Međutim, njihove kancerogene ćelije poseduju različite stope duplikacija i delecija, kao i SNP satelita, što olakšava njihovo razlikovanje.

Prilikom ovog istraživanja, najbolji rezultati postignuti su primenom algoritma k najbližih suseda u SPSS Modeleru - utvrđena tačnosti je 75%. Najlošiji rezultati dobijeni su primenom Gausove klasifikacije - utvrđena tačnost je 65%.

Konačan odgovr na pitanje da li je istraživanje uspelo ne postoji, samim tim što zavisi iz kog se ugla problem i rešenje posmatraju: ako se uzme u obzir da se od tačnosti predviđanja koja iznosi 0.5% došlo do tačnosti koja iznosi , odgovor je *da*; ako se posmatra početni skup podataka, koji sadrži veliki broj klasa, odgovor je *ne*; ako se rezultat istraživanja koristi u naučne svrhe gde je potrebna preciznost po klasi veća od 90%, odgovor je *ne*; ako se rezultat istraživanja koristi radi upoznavanja sa zakonitostima koje vladaju među sekvenciranim podacima i uopšteno osobinama kancerogenih tkiva, onda je odgovor *da*.

# Literatura


[1] Absolute tool. Online at: http://software.broadinstitute.org/cancer/software/genepattern/modules/docs/ABSOLUTE.

[2] Ccle_absolute_combined_20181227. Online at: https://portals.broadinstitute.org/ccle/data.

[3] Snp genotyping. Online at: https://en.wikipedia.org/wiki/SNP_genotyping.

[4] Tumor heterogenity. Online at: https://en.wikipedia.org/wiki/Tumour_heterogeneity.

[5] M. Maljković. Materijali sa vežbi. Online at: http://www.matf.bg.ac.rs/p/mirjana/kurs/606/istrazivanje-podataka-1/.