Klasifikacija teksta

Borisav Damnjanović

Uvod

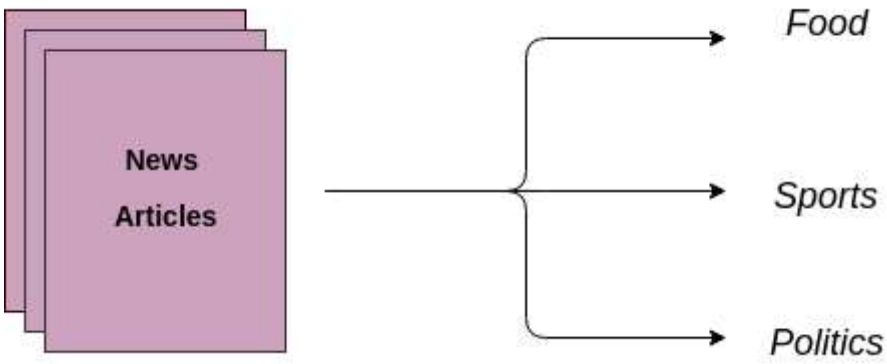
U ovom radu bavićemo se klasifikacijom teksta. Prvo ćemo pričati uopšteno o tome šta je klasifikacija teksta i zašto je ona bitna, zatim koje vrste klasifikacije teksta postoje i kako iz ugla matematike možemo posmatrati ovaj problem. Na kraju ćemo obraditi primer nad realnim podacima.

Šta je klasifikacija teksta?

Klasifikacija teksta predstavlja svrstavanje teksta u jednu ili više predefinisanih kategorija. Klasifikatori teksta mogu da se koriste za organizaciju, strukturu i klasifikaciju teksta, od dokumenata, naučnih spisa, medicinske “papirologije”, preko tekstova širom interneta. Na primer: novinski članci se mogu svrstati po rubrici, elektronska pošta se može razvrtati na spam i ham, “sentiment analysis” upitnika o nekom brendu, itd.

U klasifikaciji teksta svaka instanca (novinski članak, naučni rad itd.) se može predstaviti skupom njenih atributa. Svakoj instanci se može dodeliti oznaka klase kojoj instanca pripada (ciljna vrednost). Problem klasifikacije se sastoji u određivanju ovih vrednosti na osnovu atributa instance.

Formalnije, problem klasifikacije se može posmatrati kao aproksimacija funkcije koja svakoj instanci dodeljuje oznaku klase kojoj ta instanca pripada.



Zašto je klasifikacija teksta bitna?

Procenjuje se da je oko 80% podataka nestrukturisano, pri čemu je tekst jedan od najčešćih tipova nestrukturisanih podataka. Analiza, razumevanje, organizacija i sortiranje tekstualnih podataka je zahtevno i zbog toga mnoge kompanije ne koriste njegov pun potencijal.

Zbog toga je klasifikacija mašinskim učenjem bitna. Korišćenjem klasifikatora teksta kompanije mogu brzo i efikasno da srede različite vrste teksta. Time štede na vremenu prilikom analiziranja teksta i donose odluke na osnovu dobijenih podataka.

Vrste klasifikacije teksta

Klasifikacija teksta se može razlikovati po broju elemenata skupa ciljnih vrednosti. Ako skup ciljnih vrednosti sadrži tačno dve klase problem klasifikacije teksta je binarni. Primer binarne klasifikacije teksta je filtriranje elektronske pošte u dve klase – ”željena” i ”neželjena” pošta (eng.ham i spam).

Slično, problem klasifikacije teksta čiji skup ciljnih vrednosti sadrži više od dva elementa kažemo da je višeklasni (eng. multi-class). Primer višeklasnog problema klasifikacije teksta je problem prepoznavanja jezika na kom je tekst napisan.

U mnogo slučajeva tekst može spadati u više klasa u isto vreme. Na primer, naučni rad može u isto vreme biti svrstan u oblast pretraga informacija (eng. information retirval), mašinsko učenje (eng. machine learning) i u još neku njihovu podoblast. Ovaj tip klasifikacije teksta naziva se višeznačna (eng. multi-label) klasifikacija .

Po načinu primene, tekst klasifikacija bi se mogla podeliti na dva tipa: ručni i automatski. Ručna klasifikacija zahteva osobu koja će tumačiti tekst i kategorizovati ga prema tome. Ručna metoda pruža dobre rezultate, ali je vremenski zahtevna i spora. Automatsko klasifikovanje primenjuje mašinsko učenje, obradu prirodnih jezika (eng. Natural Language Processing - NLP) i druge tehnike veštačke inteligencije kako bi klasifikovala tekst brže, efektivnije i preciznije.

Klasifikacija mašinskim učenjem

Tekst klasifikacija mašinskim učenjem vrši klasifikaciju na osnovu prethodnih opservacija koje su već klasifikovane. Taj skup opservacija se naziva trenažni skup (eng. training data).U prvom koraku je potrebno tekst predstaviti numeričkom prezentacijom u formi vektora. Jedan od najčešćih pristupa je takozvana vreća reči(eng. bag of words), gde pomenuti vektor predstavlja učestalost svake reči u predefinisanom rečniku. Na primer, ako definišemo rečnik da sadrži sledeće reči {knjiga, je, loša, onako, dobra} i želimo da napravimo vektor od teksta “knjiga je dobra”, njega ćemo predstaviti na ovaj način (1, 1, 0, 0, 1). Onda na osnovu trenažnog skupa koji se sastoji od parova instanci (vektor za svaki primer teksta) i oznake klase gradimo naš klasifikacioni model.

Matematička postavka problema

Problem klasifikacije teksta možemo definisati na sledeći način :

PROBLEM. Neka je x instanca (tekst) i $V = \{v_1, v_2, \ldots, v_k\}$ diskretan skup klasa. Zadatak je što bolje opisati instancu x atributima $a_1, a_2, \ldots, a_n$ a zatim pronaći jednu ili više vrednosti skupa V kojoj instanca x pripada.

Formalnije, klasifikacija teksta je problem aproksimacije funkcije $f(x)$ gde je svaka instanca teksta x predstavljena kao skup atributa $a_1, a_2, \ldots, a_n$, pri čemu $f(x)$ uzima vrednosti iz diskretnog skupa ciljnih vrednosti V. Zadatak je predvideti vrednost funkcije f nove instance x.

Postoji veliki broj metoda koje se ovim problemom bave i one se dele na metode nadgledanog i na metode nenadgledanog učenja.

Neke metode nadgledanog učenja su: učenje stabla odlučivanja, metoda potpornih vektora, neuralne mreže itd. Kod metoda nadgledanog učenja trening skup nam je unapred klasifikovan.

Kada trening skup nije unapred klasifikovan, problem rešavamo nenadgledanim učenjem. Primer nenadgledanog učenja je takozvano klasterovanje – uočavanje klasa sličnih objekata kada nemamo prethodno znanje o tome koliko klasa postoji ili koje su njihove karakteristike. Pored klasterovanja najpoznatije metode nenadgledanog učenja su skriveni Markovljevi modeli, EM algoritam, algoritmi analize komponenata itd.

Primena na podacima

U ovom primeru imamo skup delova teksta iz knjiga “Zločin i kazna” i “Proces” i cilj nam je da odredimo njihovu pripadnost jednoj ili drugoj knjizi.Klasifikacija smo odradili logistickom regresijom.

Primer:

```
library(tidyverse)
library(gutenbergr)
library(tidytext)
library(stopwords)
library(glmnet)
library(rsample)
library(yardstick)
library(dplyr)
library(broom)

knjige <-gutenberg_download(c(2554,7849),meta_fields = 'title') %>%
  mutate(document =  row_number())
#izbacujemo id kolonu
knjige<-knjige[,-1]

knjige_token <- knjige %>% unnest_tokens(word, text) %>%
  group_by(word) %>% filter(n()>10) %>% ungroup()

knjige_token
```

```
## # A tibble: 264,847 × 3
##    title                document word
##    <chr>                   <int> <chr>
##  1 Crime and Punishment        1 crime
##  2 Crime and Punishment        1 and
##  3 Crime and Punishment        3 by
##  4 Crime and Punishment        7 by
##  5 Crime and Punishment       14 a
##  6 Crime and Punishment       14 few
##  7 Crime and Punishment       14 words
##  8 Crime and Punishment       14 about
##  9 Crime and Punishment       14 himself
## 10 Crime and Punishment       14 may
## # ℹ 264,837 more rows
```

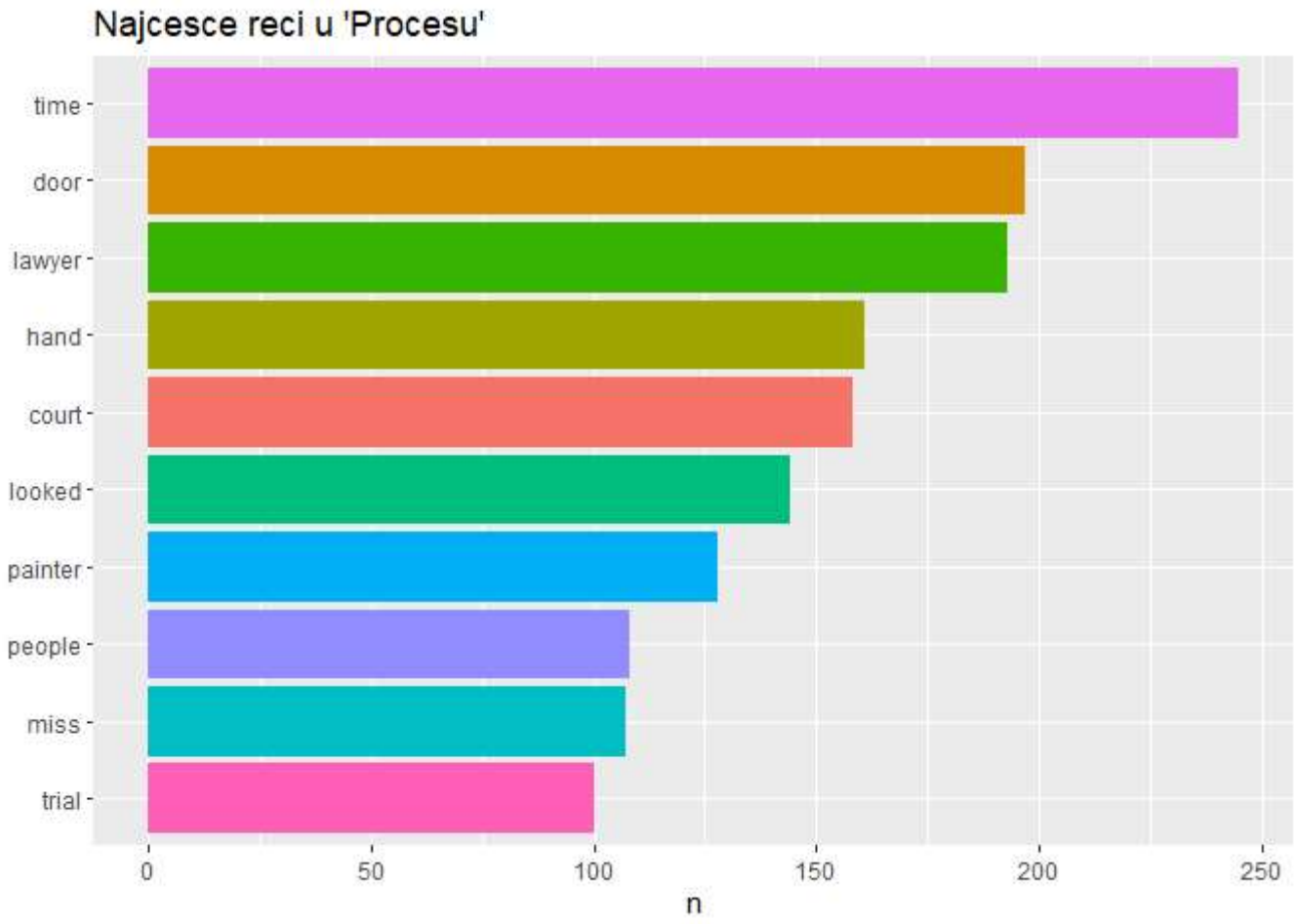
```
#sumarizovane reci u knjigama bez stopwords-a(the,a etc)
Proces<-knjige_token[knjige_token$title=='The Trial',] #uzimamo samo reči iz procesa
Proces<-group_by(Proces,word) %>% summarise(n=n()) #prebrojavamo frekvenciju reči
Proces<-anti_join(Proces,stop_words,by = 'word') #izbacujemo stopwords
Proces
```

```
## # A tibble: 1,281 × 2
##    word            n
##    <chr>       <int>
##  1 abandoned       5
##  2 abruptly        2
##  3 absolute       10
##  4 absolutely      1
##  5 abuse           2
##  6 accept         12
##  7 accompanied     5
##  8 account         5
##  9 accused        35
## 10 ach             1
## # ℹ 1,271 more rows
```

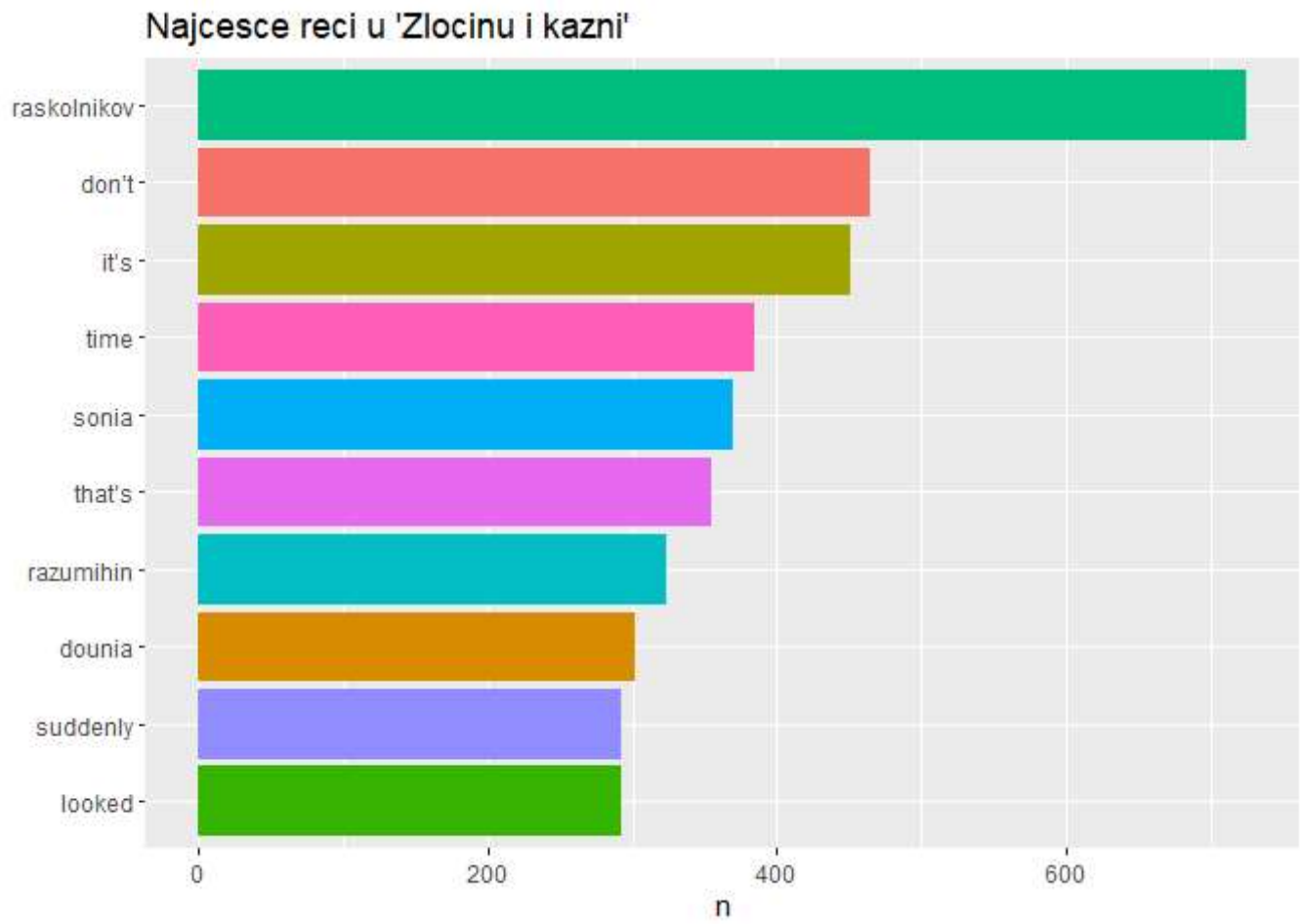
```
ZiK<-knjige_token[knjige_token$title=='Crime and Punishment',]
ZiK<-group_by(ZiK,word) %>% summarise(n=n())
ZiK<-anti_join(ZiK,stop_words,by = 'word')
ZiK
```

```
## # A tibble: 1,610 × 2
##    word            n
##    <chr>       <int>
##  1 abandoned      10
##  2 abruptly       15
##  3 absolute        2
##  4 absolutely     17
##  5 absurd         17
##  6 abuse          10
##  7 accept         22
##  8 accompanied     8
##  9 account        40
## 10 accused         6
## # ℹ 1,600 more rows
```

```
#izdvajamo 10 najcescih reci i graficki ih prikazujemo
Proces[order(Proces$n,decreasing = T),][1:10,] %>% ggplot(aes(y = fct_reorder(word, n), x = n, f
ill = word)) +
  geom_col(show.legend = FALSE)+
  labs(title = "Najcesce reci u 'Procesu'") + ylab(NULL)
```




```
ZiK[order(ZiK$n,decreasing = T),][1:10,] %>% ggplot(aes(y = fct_reorder(word, n), x = n, fill = word)) +
  geom_col(show.legend = FALSE)+
  labs(title = "Najcesce reci u 'Zlocinu i kazni'") + ylab(NULL)
```



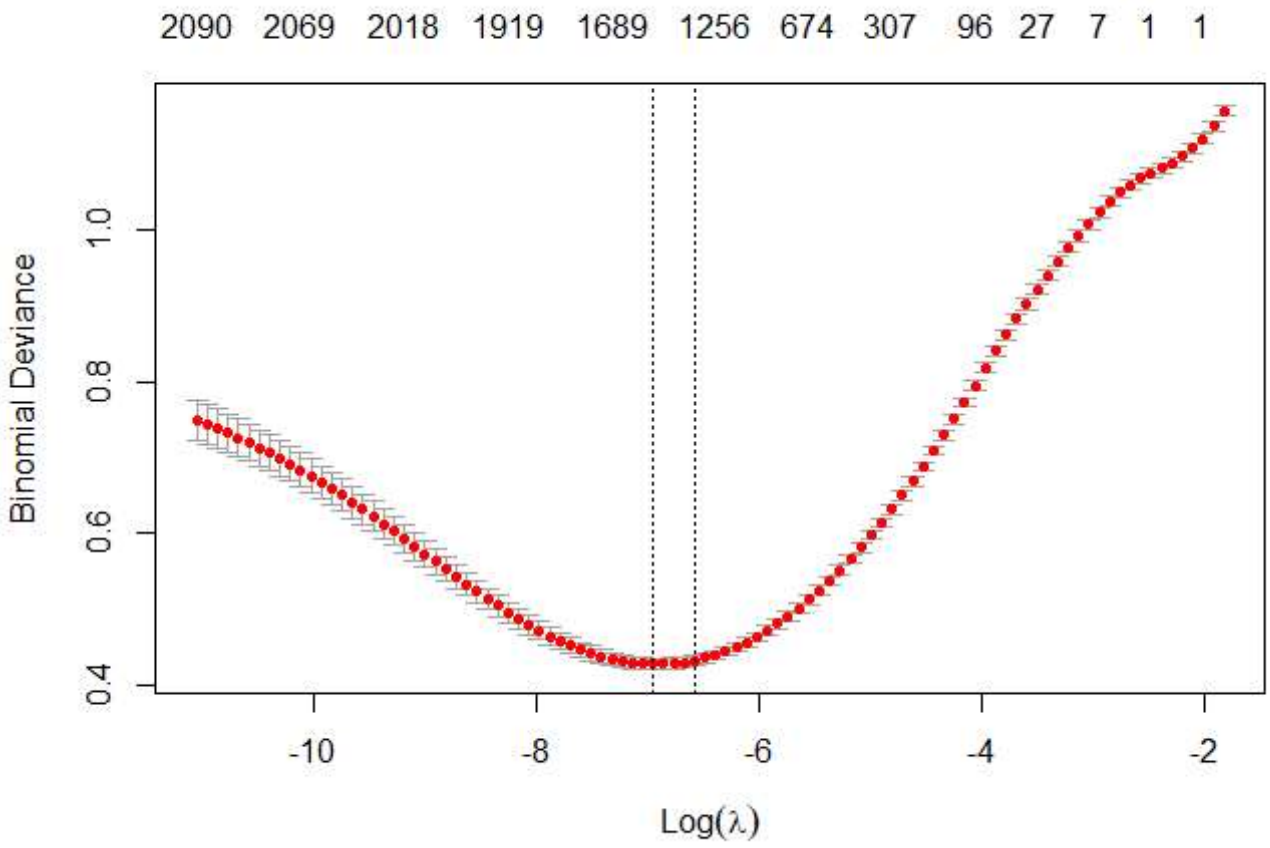

```
#Delimo bazu po dokumentima na trening i test set koristeci rsample biblioteku
knjige_1 <- knjige %>%
  select(document) %>%
  initial_split()
trening <- training(knjige_1)
valid <- testing(knjige_1)

#Pravimo sparse matricu(X promenljiva) za nas klasifikacioni model
#gde su redovi document, kolone word i elementi n
sparse_matrica<-group_by(knjige_token,document,word) %>% summarise(n=n()) #sumarizujemo reči i d
okumente
sparse_matrica<-inner_join(sparse_matrica,trening) #uzimamo samo one koji su i u trening setu
sparse_matrica<-cast_sparse(sparse_matrica,document, word, n) #pravimo sparse matricu

dim(sparse_matrica)
```

```
## [1] 18302  2128
```

```
#pravimo Y promenljivu koja ce odgovarati spars matrici.
p <- as.integer(rownames(sparse_matrica)) # int vektor koji odgovara rednom broju dokumenta iz s
parse matrice

knjige_2 <- data_frame(document = p) #novi data_frame da bismo baratali f-jama left_join i selec
t
knjige_2<-left_join(knjige_2,knjige)  #uzimamo one dokumente koji su u sparse matrici
knjige_2<- select(knjige_2,document, title)  #izbacujemo tekst kolonu, ostavljamo document i tit
le.
knjige_2
```

```
## # A tibble: 18,302 × 2
##    document title
##       <int> <chr>
##  1        1 Crime and Punishment
##  2        3 Crime and Punishment
##  3        7 Crime and Punishment
##  4       14 Crime and Punishment
##  5       19 Crime and Punishment
##  6       20 Crime and Punishment
##  7       21 Crime and Punishment
##  8       25 Crime and Punishment
##  9       29 Crime and Punishment
## 10       30 Crime and Punishment
## # ℹ 18,292 more rows
```

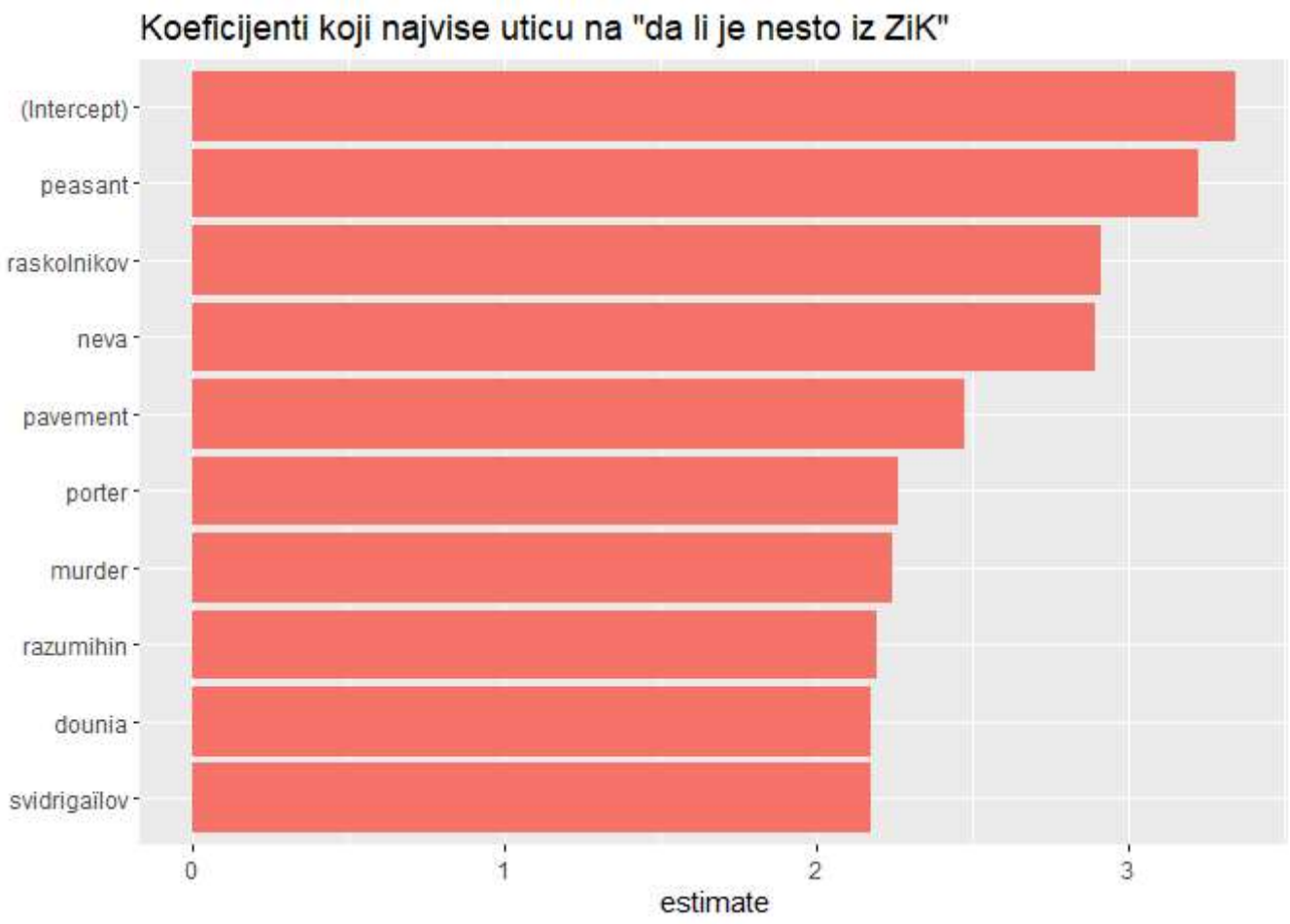
```
#Vektor True i False koji odgovara da li je observacija iz Zlocina i kazne.
da_li_je_ZiK <- knjige_2$title == "Crime and Punishment"

#Pozivamo cv.glmnet koji radi regularizaciju ali i pamti  koeficijente za svako lambda u model$g
lmnet.fit
model <- cv.glmnet(sparse_matrica, da_li_je_ZiK,
                   family = "binomial",alpha=1 )
plot(model)
```



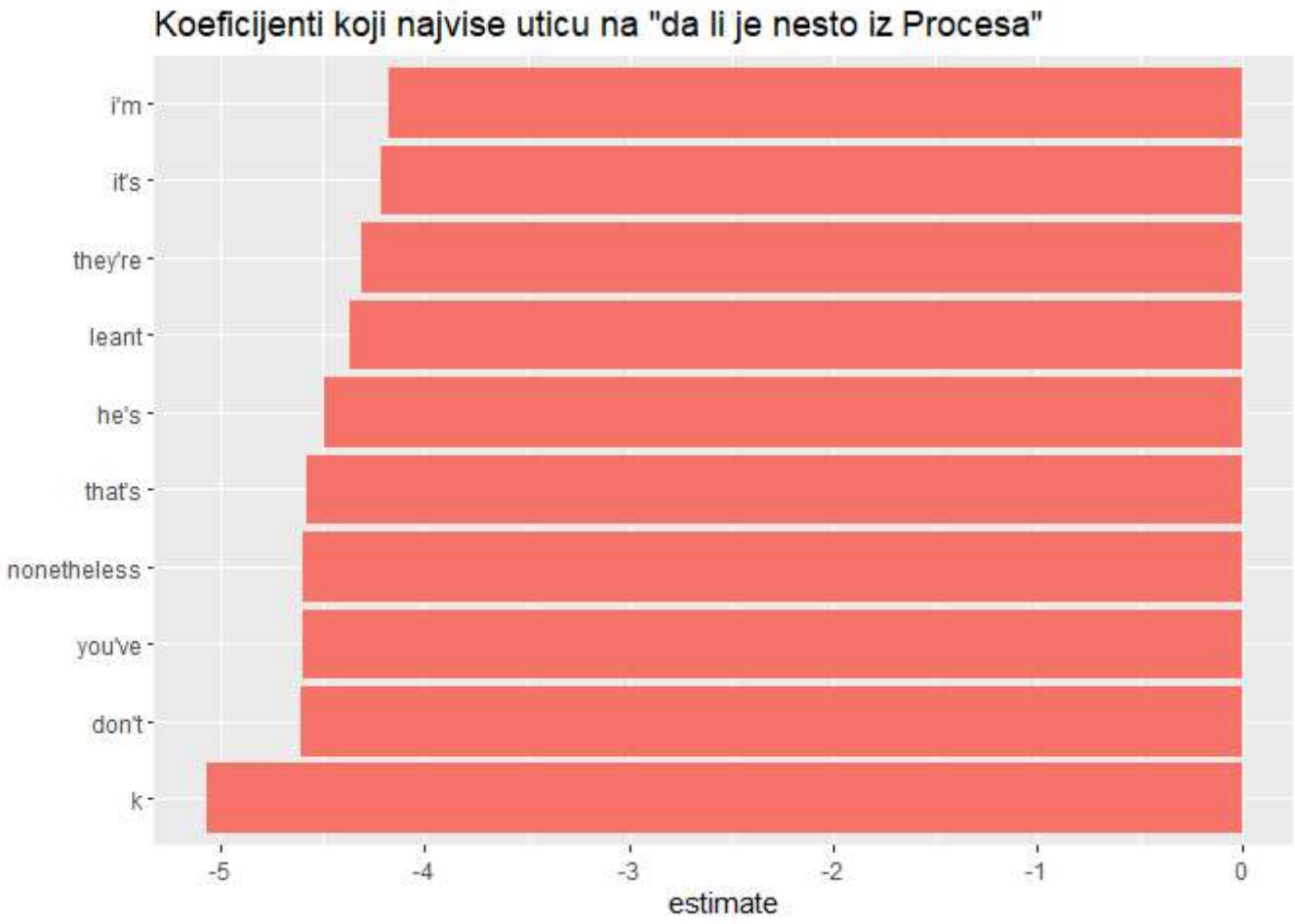

```
#uzimmo koeficijente za lambda.1se
koeficijenti <- model$glmnet.fit %>% tidy()
koeficijenti<-koeficijenti[koeficijenti$lambda== model$lambda.1se,]

koeficijenti
```

```
## # A tibble: 1,365 × 5
##    term         step estimate  lambda dev.ratio
##    <chr>       <dbl>    <dbl>   <dbl>     <dbl>
##  1 (Intercept)    52    3.35  0.00140     0.717
##  2 and            52    0.187 0.00140     0.717
##  3 crime          52    1.75  0.00140     0.717
##  4 by             52   -0.182 0.00140     0.717
##  5 about          52   -0.523 0.00140     0.717
##  6 few            52   -1.29  0.00140     0.717
##  7 help           52   -0.229 0.00140     0.717
##  8 himself        52   -0.297 0.00140     0.717
##  9 may            52    0.761 0.00140     0.717
## 10 the            52   -0.546 0.00140     0.717
## # ℹ 1,355 more rows
```

```
#najznacajniji koeficijenti za Zlocin i kaznu
koeficijenti[order(koeficijenti$estimate,decreasing = T),][1:10,] %>% #koef koji imaju najveci estimate
  ggplot(aes(fct_reorder(term, estimate), estimate, fill = estimate >0)) + geom_col( show.legend = FALSE) + coord_flip() + labs(
    x = NULL,
    title = 'Koeficijenti koji najvise uticu na "da li je nesto iz ZiK"'
  )
```




```
#najznacajniji koeficijenti za Proces
koeficijenti[order(koeficijenti$estimate,decreasing = F),][1:10,]%>% #koef koji imaju najmanji estimate
  ggplot(aes(fct_reorder(term, estimate), estimate, fill = estimate <0)) + geom_col( show.legend = FALSE) + coord_flip() + labs(
    x = NULL,
    title = 'Koeficijenti koji najvise uticu na "da li je nesto iz Procesa"'
  )
```



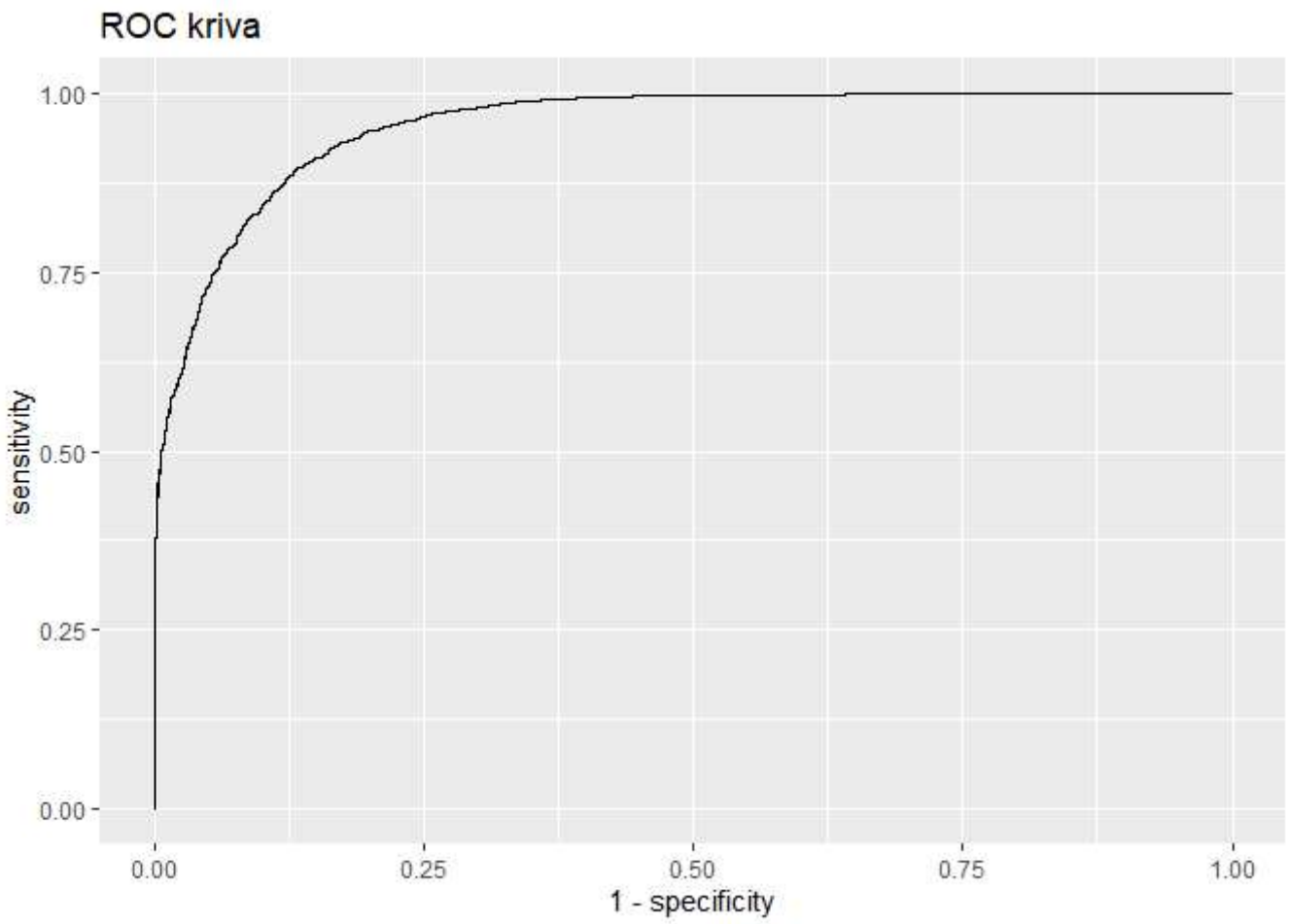

```
#izdvajamo intercept
intercept<-koeficijenti$estimate[1]

#Radimo klasifikaciju na valid skupu,
#prosledivsi odgovarajucu vrednost koeficijenata f-ji raspodele logisticke raspodele(plogis),
#dodajemo kolonu za vracenu verovatnocu.
klasifikacija<- knjige_token %>%
  inner_join(valid) %>%  #uzimamo samo redove koji su i u valid skupu
  inner_join(koeficijenti, by = c('word' = 'term')) %>%  #dodajemo izračunate koficijente
  group_by(document) %>%
  summarize(suma = sum(estimate)) #računamo zbir estimate za svaki dokument

klasifikacija$verovatnoca<-plogis(intercept + klasifikacija$suma)

klasifikacija
```

```
## # A tibble: 6,038 × 3
##    document   suma verovatnoca
##       <int>  <dbl>       <dbl>
##  1       15 -1.41        0.874
##  2       17 -3.14        0.551
##  3       18 -1.40        0.875
##  4       23  1.34        0.991
##  5       24  0.143       0.970
##  6       27 -2.02        0.790
##  7       28 -1.16        0.899
##  8       31 -2.31        0.739
##  9       34 -2.64        0.669
## 10       41 -0.245       0.957
## # ℹ 6,028 more rows
```

```
#Klasifikaciji dodajemo kolonu koja odgovara naslovu knjige.
klasifikacija <- left_join(klasifikacija, select(knjige,title, document), by ='document')
klasifikacija$title<-as.factor(klasifikacija$title)

klasifikacija
```

```
## # A tibble: 6,038 × 4
##    document   suma verovatnoca title
##       <int>  <dbl>       <dbl> <fct>
##  1       15 -1.41        0.874 Crime and Punishment
##  2       17 -3.14        0.551 Crime and Punishment
##  3       18 -1.40        0.875 Crime and Punishment
##  4       23  1.34        0.991 Crime and Punishment
##  5       24  0.143       0.970 Crime and Punishment
##  6       27 -2.02        0.790 Crime and Punishment
##  7       28 -1.16        0.899 Crime and Punishment
##  8       31 -2.31        0.739 Crime and Punishment
##  9       34 -2.64        0.669 Crime and Punishment
## 10       41 -0.245       0.957 Crime and Punishment
## # ℹ 6,028 more rows
```

```
#Crtamo ROC krivu
roc_curve(klasifikacija,title, verovatnoca) %>%
  ggplot(aes(x = 1-specificity, y = sensitivity)) +
  geom_line() + labs(
    title = 'ROC kriva'
  )
```




```
#Povrsina ispod krive
roc_auc(klasifikacija,title, verovatnoca)
```

```
## # A tibble: 1 × 3
##   .metric .estimator .estimate
##   <chr>   <chr>          <dbl>
## 1 roc_auc binary         0.956
```

```
#Matrica konfuzije, dodajemo klasifikaciji novu kolonu za prognozu
klasifikacija$prognoza<-if_else(klasifikacija$verovatnoca >0.5,'Crime and Punishment','The Tria
l')
klasifikacija$prognoza<-as.factor(klasifikacija$prognoza)
klasifikacija %>% conf_mat(title,prognoza)
```

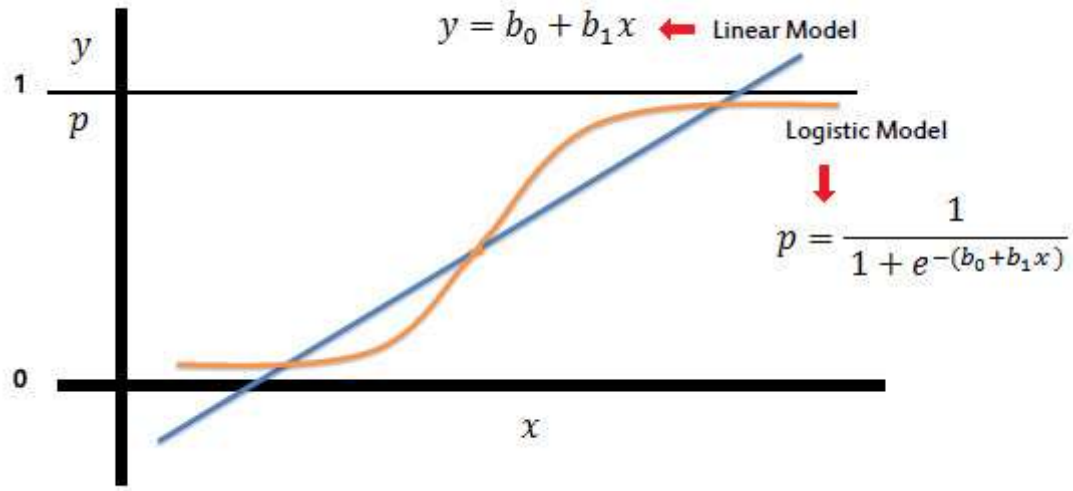
```
##                      Truth
## Prediction            Crime and Punishment The Trial
##   Crime and Punishment                 4267       403
##   The Trial                             144      1224
```

Ukratko Logistička klasifikacija

Logistička regresija je statistički model koji u osnovi koristi logističku funkciju za modeliranje binarne zavisne promenljive. Umesto traženja ocene y_i ocenićemo verovatnoću da y_i uzme svaku od mogućih vrednosti. Ne može se koristiti linearna regresija jer bismo mogli da izađemo iz intervala [0, 1]. Recimo da ciljna promenljiva uzima vrednosti 0 i 1, može se reći da je ona Bernulijeva raspodela sa parametrom koji zavisi od prediktora. Ocenu verovatnoće da ciljna promenljiva uzima 0 ili 1, ocenjujemo logističkom funkcijom, tj.

$$\bar{P}(y_i = 1) = f(x_{i1}, x_{i2}, \ldots, x_{ip}) = \frac{1}{1 + e^{-B_0 - B_1 x_{i1} \ldots\ldots - B_p x_{ip}}}$$

Ovakva ocena ostaje u intervalu [0, 1], a ocene parametra modela se računaju metodom maksimalne verodostojnosti. Za primenu maksimalne verodostojnosti potrebno je pretpostaviti da su sve Bernulijeve slučajne veličine međusobno nezavisne.



Nakon izgradnje modela, odnosno dobijenih ocena parametara, istim modelom vršimo klasifikaciju novih tačaka tako što im dodeljujemo onu kategoriju označenu jedinicom kada je ta verovatnoća veća od određenog praga. U suprotnom joj dodeljujemo kategoriju označenu nulom.

Zaključak

Spomenuli smo kako se može klasifikovati tekst korišćenjem različitih tehnika. Napravili smo model upotrebom logističke regresije koji pravi razliku između “rukopisa” Dostojevskog i Kafke. Sličan pristup je primenila Emil Hvifeldt radi predikcije autora nekih delova Federalist Papers (https://www.emilhvitfeldt.com/post/2018-01-30-predicting-authorship-in-the-federalist-papers-tidytext/).

Takodje, videli smo na koji način se može srediti tekst (stemming , stopwords, …) tako da nam posle njegovog sređivanja rezultati klasifikacije budu bolji.

Tekst klasifikacija ima pregršt korisnih svojstava i primenjena je na širokom spektru problema. Neretko funkcionise iza kulisa da obogati i unapredi korišćenje raznih aplikacija (npr. spam i ham kod elektronske pošte). U nekim drugim slučajevima je koriste marketniški stručnjaci, produkt menadzeri i trgovci kako bi automatizovali biznis i izbegli mnogo utrošenog vremena na ručnu obradu podataka.

Sve to čini klasifikaciju teksta veoma bitnom oblašću i jednom od oblasti koja će nastaviti mnogo da se razvija i u budućnosti.