Projekat iz Statističkog softvera 3

Jana Živković

2025-07-23

Učitavanje podataka

```
library(readODS)
rakete <- read_ods(
  "/home/janazivkovic/Documents/Faks/7 semestar/Statisticki softver 3/Projekat/rakete.ods",
  sheet = 1)
```

U bazi podataka rakete postoje dve kolone za ime rakete (rusko_ime i NATO_ime) i tri kolone koje predstavljaju kategoričke vrednosti.

```
library(dplyr)
rakete <- rakete %>%
  mutate(
    klasa_po_dometu = case_when(
      klasa_po_dometu == "S" ~ 1,
      klasa_po_dometu == "M" ~ 2,
      klasa_po_dometu == "L" ~ 3
    ),
    tip_goriva = case_when(
      tip_goriva == "Tecno" ~ 1,
      tip_goriva == "Cvrsto" ~ 2
    ),
    nacin_lansiranja = case_when(
      nacin_lansiranja == "Silos" ~ 1,
```

```
      nacin_lansiranja == "RM TEL" ~ 2,
      nacin_lansiranja == "Mornaricki" ~ 3,
      nacin_lansiranja == "Avion" ~ 4
    )
  )
```

Iz imena mogu izvući još neke informacije o podacima. U koloni NATO_ime stoje imena raketa po NATO notaciji. U tri reda imena ne prate kriterijum zapisivanja raketa, njih mozemo da prepravimo.

```
rakete$NATO_ime[39] <- "CSS-X-11"
rakete$NATO_ime[33] <- "CSS-18"
rakete$NATO_ime[16] <- "AS-24"
```

NATO imena raketa prate sledeće pravilo: ukoliko ime počinje sa CSS to je raketa kineske proizvodnje dok je inače raketa ruske proizvodnje. Ovo takođe možemo da dodamo kao kategoričku kolonu u bazi podataka.

```
rakete <- rakete %>%
  mutate(drzava = case_when(
    startsWith(NATO_ime, "SS") ~ "R",
    startsWith(NATO_ime, "CSS") ~ "Ch",
    TRUE ~ "Unknown"
  ))

rakete$drzava[16] <- "R"
rakete <- rakete %>%
  mutate(
    drzava = case_when(
      drzava == "R" ~ 1,
      drzava == "Ch" ~ 2
    )
  )
```

Drugo pravilo koje prati ova kolona je sledeće, ukoliko raketa u svom nazivu ima slovo N ona je

lansirana sa broda ili podmornice, inače je lansirana sa kopna, osim u slučaju rakete AS-24 koja je raketa koja se lansira iz aviona.

Rad sa nedostajućim podacima

```
## # A tibble: 2 x 13
##   rusko_ime    NATO_ime masa_kg duzina_m kalibar_m broj_podglava payload_kg
##   <chr>        <chr>      <dbl>    <dbl>     <dbl>         <dbl>      <dbl>
## 1 Dong Feng 16 CSS-11        NA       NA       1.2             3       1500
## 2 Dong Feng 26 CSS-18     20000       14       1.4            NA       1800
## # i 6 more variables: tip_goriva <dbl>, max_domet_km <dbl>,
## #   nacin_lansiranja <dbl>, broj_faza_motora <dbl>, klasa_po_dometu <dbl>,
## #   drzava <dbl>
```

Nedostajuće vrednosti se nalaze u dve vrste i odnose se na rakete Dong Feng, pa možemo da detaljnije posmatramo baš te rakete.

```
## # A tibble: 10 x 13
##    rusko_ime    NATO_ime masa_kg duzina_m kalibar_m broj_podglava payload_kg
##    <chr>        <chr>      <dbl>    <dbl>     <dbl>         <dbl>      <dbl>
##  1 Dong Feng 11 CSS-7       3800      7.5      0.8             1        800
##  2 Dong Feng 12 CSS-X-15    4010      7.3      0.92            1        480
##  3 Dong Feng 15 CSS-6       6200      9.1      1               1        500
##  4 Dong Feng 16 CSS-11        NA     NA        1.2             3       1500
##  5 Dong Feng 21 CSS-5      14700     10.7      1.4             6        600
##  6 Dong Feng 26 CSS-18     20000     14        1.4            NA       1800
##  7 Dong Feng 31 CSS-10     42000     14.5      2               3       1750
##  8 Dong Feng 5  CSS-4     183000     32.6      3.35           10       4000
##  9 Dong Feng 4  CSS-5      82000     28        2.25            1       2200
## 10 Dong Feng 41 CSS-20     80000     22        2.25           10       2500
## # i 6 more variables: tip_goriva <dbl>, max_domet_km <dbl>,
## #   nacin_lansiranja <dbl>, broj_faza_motora <dbl>, klasa_po_dometu <dbl>,
## #   drzava <dbl>
```

Rakete Dong Feng 4 i 5 su jedine koje imaju drugačiji način lansiranja i tip goriva. Takođe ostale

rakete u tabeli su numerisene od 10 pa naviše i njihova numeracija prati rastući poredak, kao i njihove mase i dužine. Zbog ovoga ću te dve vrste da zanemarim na trenutak dok pretpostavljam nedostajuće vrednosti za raketu Dong Feng 16.

```
## # A tibble: 8 x 13
##   rusko_ime    NATO_ime masa_kg duzina_m kalibar_m broj_podglava payload_kg
##   <chr>        <chr>      <dbl>    <dbl>     <dbl>         <dbl>      <dbl>
## 1 Dong Feng 11 CSS-7       3800      7.5      0.8               1        800
## 2 Dong Feng 12 CSS-X-15    4010      7.3      0.92              1        480
## 3 Dong Feng 15 CSS-6       6200      9.1      1                 1        500
## 4 Dong Feng 16 CSS-11        NA     NA        1.2               3       1500
## 5 Dong Feng 21 CSS-5      14700     10.7      1.4               6        600
## 6 Dong Feng 26 CSS-18     20000     14        1.4              NA       1800
## 7 Dong Feng 31 CSS-10     42000     14.5      2                 3       1750
## 8 Dong Feng 41 CSS-20     80000     22        2.25             10       2500
## # i 6 more variables: tip_goriva <dbl>, max_domet_km <dbl>,
## #   nacin_lansiranja <dbl>, broj_faza_motora <dbl>, klasa_po_dometu <dbl>,
## #   drzava <dbl>
```

Pored linearnosti koju možemo da uočimo, možemo i da prepostavimo prirodno da ove vrednosti rastu kako raste redni broj rakete iz određene serije. Zbog rastućeg poretka, nedostajuće vrednosti ovde ću da pretpostavim metodom interpolacije. **na.approx** funkcija zamenjuje nedostajuće vrednosti (NA) u nizu linearnom interpolacijom (tj. srednjom vrednošću susednih vrednosti).

```
library(zoo)
interpolacija <- rakete_dong_feng
posmatrane_kolone <- c("masa_kg", "duzina_m")
for (col in posmatrane_kolone) {
  interpolacija[[col]] <- na.approx(rakete_dong_feng[[col]],
                                    na.rm = FALSE)
}
print(interpolacija)
```

```
## # A tibble: 8 x 13
```

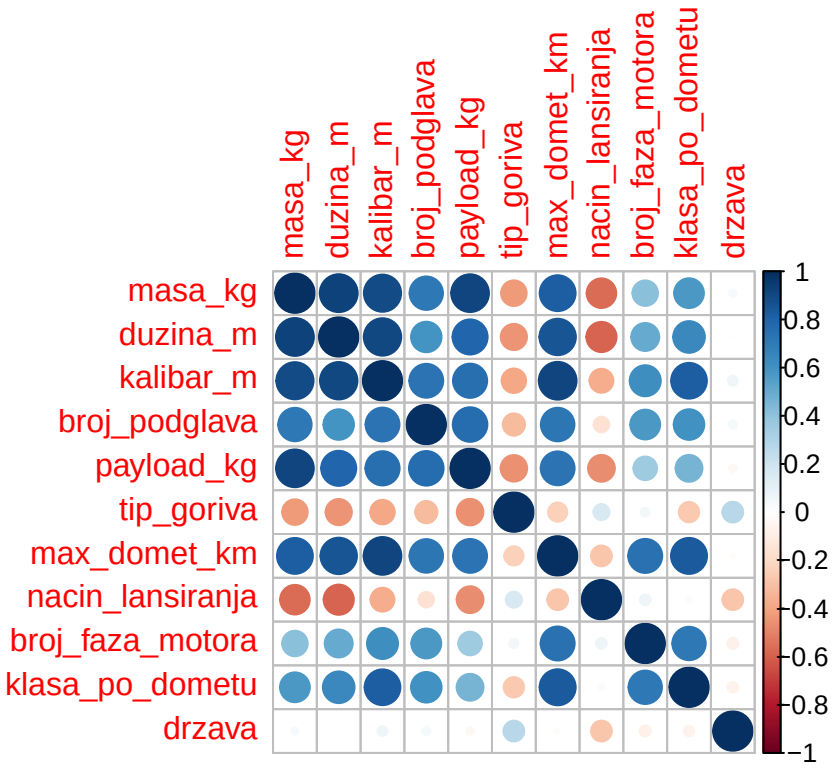
```
##   rusko_ime    NATO_ime masa_kg duzina_m kalibar_m broj_podglava payload_kg
##   <chr>        <chr>      <dbl>    <dbl>     <dbl>         <dbl>      <dbl>
## 1 Dong Feng 11 CSS-7       3800      7.5      0.8               1        800
## 2 Dong Feng 12 CSS-X-15    4010      7.3      0.92              1        480
## 3 Dong Feng 15 CSS-6       6200      9.1      1                 1        500
## 4 Dong Feng 16 CSS-11     10450      9.9      1.2               3       1500
## 5 Dong Feng 21 CSS-5      14700     10.7      1.4               6        600
## 6 Dong Feng 26 CSS-18     20000     14        1.4              NA       1800
## 7 Dong Feng 31 CSS-10     42000     14.5      2                 3       1750
## 8 Dong Feng 41 CSS-20     80000     22        2.25             10       2500
## # i 6 more variables: tip_goriva <dbl>, max_domet_km <dbl>,
## #   nacin_lansiranja <dbl>, broj_faza_motora <dbl>, klasa_po_dometu <dbl>,
## #   drzava <dbl>
```

```
rakete[31,3]<-interpolacija[4,3]
rakete[31,4]<- interpolacija[4,4]
```

Ostaje samo da popunimo i nedostajuću vrednost rakete Dong Feng 26 u koloni broj_podglava. Za ovu kolonu ne uočavamo pravilnosti koje se uočavaju za masu i dužinu.

Ideja je da ovde primenimo linearnu regresiju, ali pre toga možemo da posmatramo korelisanost.

```
library(corrplot)
corrplot(cor(rakete[-33,-c(1,2)]))
```



Zbog velike korelisanosti, koristićemo analzu glavnih komponenti.

```
library(caret)
preProc <- preProcess(rakete[-33, c(3,4,5,7,8)], method = "pca", pcaComp = 2)
PCA <- predict(preProc, rakete[-33, c(3,4,5,7,8)])
PCA$broj_podglava <- rakete$broj_podglava[-33]

model <- train(broj_podglava ~ ., data = PCA, method = "lm")

testPCA <- predict(preProc, rakete[33,c(3,4,5,7,8)])

broj_podglava_NA <- predict(model, newdata = testPCA)
broj_podglava_NA <- round(broj_podglava_NA)

rakete[33,6] <- broj_podglava_NA
print(rakete[33,])
```

```
## # A tibble: 1 x 13
```

```
##   rusko_ime    NATO_ime masa_kg duzina_m kalibar_m broj_podglava payload_kg
##   <chr>        <chr>      <dbl>    <dbl>     <dbl>         <dbl>      <dbl>
## 1 Dong Feng 26 CSS-18     20000       14       1.4             3       1800
## # i 6 more variables: tip_goriva <dbl>, max_domet_km <dbl>,
## #   nacin_lansiranja <dbl>, broj_faza_motora <dbl>, klasa_po_dometu <dbl>,
## #   drzava <dbl>
```

Napomena: Za popunjavanje nedostajućih vrednosti nisam koristila vrednosti koje kasnije pokušavam da predvidim i modeliram kao max_domet_km, nacin_lansiranja itd.

Predviđanje maksimalnog dometa

Zbog velike korelisanosti među kolonama očekujem se da linearni model ponaša malo čudno.

```
linearni_model<- lm(max_domet_km ~ . -rusko_ime - NATO_ime , data = rakete)
summary(linearni_model)
```

```
##
## Call:
## lm(formula = max_domet_km ~ . - rusko_ime - NATO_ime, data = rakete)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2695.5  -605.0  -129.6   764.3  3632.8
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)     -9.094e+03  2.693e+03  -3.377  0.00217 **
## masa_kg          1.018e-02  2.661e-02   0.383  0.70493
## duzina_m         1.961e+02  1.347e+02   1.456  0.15649
## kalibar_m        1.044e+03  1.768e+03   0.591  0.55948
## broj_podglava    9.132e+00  1.406e+02   0.065  0.94869
## payload_kg       4.738e-01  4.169e-01   1.137  0.26533
## tip_goriva       1.508e+03  7.638e+02   1.974  0.05830 .
## nacin_lansiranja 3.517e+02  6.718e+02   0.523  0.60477
```

```
## broj_faza_motora  1.063e+03  5.958e+02   1.784  0.08524 .
## klasa_po_dometu   1.772e+03  7.789e+02   2.275  0.03078 *
## drzava           -2.581e+02  6.667e+02  -0.387  0.70157
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1562 on 28 degrees of freedom
## Multiple R-squared:  0.9343, Adjusted R-squared:  0.9109
## F-statistic: 39.84 on 10 and 28 DF,  p-value: 6.581e-14
```

Vidimo da je p vrednost F statistike jako mala, ali da su pojedinačne p vrednosti posmatranih karakteristika velike, što je i očekivano zbog kolinearnosti. Sada ću posmatrati podskup ovih obeležja i probati da napravim linearni model.

```
linearni_model_2 <- lm(max_domet_km ~ duzina_m + payload_kg + broj_faza_motora
                       + klasa_po_dometu + tip_goriva, data = rakete)
summary(linearni_model_2)
```

```
##
## Call:
## lm(formula = max_domet_km ~ duzina_m + payload_kg + broj_faza_motora +
##     klasa_po_dometu + tip_goriva, data = rakete)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2816.9  -995.4    37.0   835.6  3594.7
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)      -8796.4720  1316.1641  -6.683 1.31e-07 ***
## duzina_m           239.2430    59.1571   4.044 0.000297 ***
## payload_kg           0.6900     0.1995   3.459 0.001515 **
## broj_faza_motora  1150.1937   457.2813   2.515 0.016943 *
## klasa_po_dometu   2329.3454   476.8945   4.884 2.59e-05 ***
```

```
## tip_goriva        1387.3380   612.9618   2.263 0.030317 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1484 on 33 degrees of freedom
## Multiple R-squared:  0.9301, Adjusted R-squared:  0.9195
## F-statistic: 87.79 on 5 and 33 DF,  p-value: < 2.2e-16
```

```
anova(linearni_model_2, linearni_model)
```

```
## Analysis of Variance Table
##
## Model 1: max_domet_km ~ duzina_m + payload_kg + broj_faza_motora + klasa_po_dometu +
##     tip_goriva
## Model 2: max_domet_km ~ (rusko_ime + NATO_ime + masa_kg + duzina_m + kalibar_m +
##     broj_podglava + payload_kg + tip_goriva + nacin_lansiranja +
##     broj_faza_motora + klasa_po_dometu + drzava) - rusko_ime -
##     NATO_ime
##   Res.Df      RSS Df Sum of Sq      F Pr(>F)
## 1     33 72722835
## 2     28 68290773  5   4432063 0.3634 0.8691
```

Kada uporedimo ove modele vidimo da možemo da prihvatimo hipotezu da su koeficijenti uz obeležja koja nisu navedena u drugom modelu 0. Dakle biramo drugi model. Proverimo i VIF koeficijente drugog modela

```
library(car)
vif(linearni_model_2)
```

```
##         duzina_m       payload_kg broj_faza_motora  klasa_po_dometu
##         3.782808         2.816088         2.540071         2.863192
##       tip_goriva
##         1.573794
```

Sve vrednosti su ispod 5.

Za proveravanje kvaliteta modela ću koristiti unakrsnu validaciju. Pored modela linearne regresije, napraviću i ridge i lasso model, a takođe ću da primenim analizu glavnih komponenti i napravim i taj model.

```
set.seed(126)
foldovi <- createFolds(rakete$max_domet_km, k = 13)
df<-rakete[,-c(1,2)]

cv_results_lm <- lapply(foldovi, function(fold_indeks) {
  train_data <- df[-fold_indeks, ]
  test_data <- df[fold_indeks, ]

  model <- lm(max_domet_km ~ . , data = train_data)
  predikcije <- predict(model, test_data)

  stvarne_vrednosti <- test_data$max_domet_km
  r2 <- 1 - sum((stvarne_vrednosti - predikcije)^2) / sum((stvarne_vrednosti - mean(stvarne_vr
  return(r2)
})
r2_linearni_model_2 <- mean(unlist(cv_results_lm))

library(glmnet)
cv_results_lasso <- lapply(foldovi, function(fold_indeksi) {
  train_data <- rakete[-fold_indeksi, ]
  test_data <- rakete[fold_indeksi, ]

  X_train <- as.matrix(train_data[, -c(1,2,9)])
  y_train <- train_data$max_domet_km
  X_test <-  as.matrix(test_data[, -c(1,2,9)])
  y_test <- test_data$max_domet_km

  lasso_model <- cv.glmnet(X_train, y_train, alpha = 1)
```

```
  najbolje_lambda <- lasso_model$lambda.min

  predikcije <- predict(lasso_model, newx = X_test, s = najbolje_lambda)

  r2 <- 1 - sum((y_test - predikcije)^2) / sum((y_test - mean(y_test))^2)
  return(r2)
})
r2_lasso <- mean(unlist(cv_results_lasso))

cv_results_ridge <- lapply(foldovi, function(fold_indeksi) {
  train_data <- rakete[-fold_indeksi, ]
  test_data <- rakete[fold_indeksi, ]

  X_train <- as.matrix(train_data[, -c(1,2,9)])
  y_train <- train_data$max_domet_km
  X_test <-  as.matrix(test_data[, -c(1,2,9)])
  y_test <- test_data$max_domet_km

  ridge_model <- cv.glmnet(X_train, y_train, alpha = 0)
  najbolje_lambda <- ridge_model$lambda.min

  predikcije <- predict(ridge_model, newx = X_test, s = najbolje_lambda)

  r2 <- 1 - sum((y_test - predikcije)^2) / sum((y_test - mean(y_test))^2)
  return(r2)
})
r2_ridge <- mean(unlist(cv_results_ridge))

cv_results_PCA <- lapply(foldovi, function(fold_indeksi){
  train_data <- rakete[-fold_indeksi, ]
  test_data <- rakete[fold_indeksi, ]
```

```
  preProc <- preProcess(train_data[, -c(1,2,9)], method = "pca", pcaComp = 5)
  trainPCA <- predict(preProc, train_data[, -c(1,2,9)])
  testPCA <- predict(preProc, test_data[, -c(1,2,9)])

  trainPCA$max_domet_km <- train_data$max_domet_km
  testPCA$max_domet_km <- test_data$max_domet_km

  model <- train(max_domet_km ~ ., data = trainPCA, method = "lm")
  predikcije <- predict(model, newdata = testPCA)

  r2 <- 1 - sum((test_data$max_domet_km - predikcije)^2) /
    sum((test_data$max_domet_km - mean(test_data$max_domet_km ))^2)
  return(r2)
})
r2_PCA <- mean(unlist(cv_results_PCA))

rezultati <- data.frame(
  Model = c("Linearni model 2", "Lasso", "Ridge", "PCA"),
  Mean_R2 = c(r2_linearni_model_2, r2_lasso, r2_ridge, r2_PCA)
)
print(rezultati)
```

```
##              Model   Mean_R2
## 1 Linearni model 2 0.7018022
## 2            Lasso 0.7428392
## 3            Ridge 0.8007795
## 4              PCA 0.8328309
```

Najbolji je PCA model.

```
final_preProc <- preProcess(rakete[, -c(1,2,9)], method = "pca", pcaComp = 5)
raketePCA <- predict(final_preProc, rakete[, -c(1,2,9)])
raketePCA$max_domet_km <- rakete$max_domet_km
```

```
finalni_model <- train(max_domet_km ~ ., data = raketePCA, method = "lm")
```

Klasifikacija lansera

Sada možemo da napravimo novu kolonu koja govori da li je lansiranje rakete sa kopna ili ne. Jedine rakete koje imaju fiksni lanser su one lansirane iz silosa, a to su jedan od dva kopnena načina lansiranja, pa ova kolona izdvojena na osnovu imena može biti korisna u daljem radu.

```
rakete_za_klasifikaciju <- rakete
rakete_za_klasifikaciju <- rakete %>%
  mutate(kopneno_lansiranje = case_when(
    grepl("^SS-\\d+$", NATO_ime) ~ 0,
    grepl("^SS-X-\\d+$", NATO_ime) ~ 0,
    grepl("^SS-N-\\d+$", NATO_ime) ~ 1,
    grepl("^AS-\\d+$", NATO_ime) ~ 2,
    grepl("^CSS-\\d+$", NATO_ime) ~ 0,
    grepl("^CSS-N-\\d+$", NATO_ime) ~ 1,
    grepl("^CSS-X-\\d+$", NATO_ime) ~ 1,
    TRUE ~ NA_real_
    ))
rakete_za_klasifikaciju <- rakete_za_klasifikaciju %>%
   mutate(
    kopneno_lansiranje = case_when(
      kopneno_lansiranje == 0 ~ 1,
      kopneno_lansiranje == 1 ~ 0,
      kopneno_lansiranje == 2 ~ 0
    )
  )
```

Sada kada su izvođene korisne informacije iz kolona za ime možemo da ih uklonimo iz baze.

Preostaje da napravimo još jednu kolonu koja govori o tome da li je lanser fiksan ili ne.

```
rakete_za_klasifikaciju <- rakete_za_klasifikaciju %>%
   mutate(
```

```
    fiksni_lanser = case_when(
      nacin_lansiranja == 1 ~ 1,
      nacin_lansiranja == 2 ~ 0,
      nacin_lansiranja == 3 ~ 0,
      nacin_lansiranja == 4 ~ 0
    )
  )
rakete_za_klasifikaciju <- rakete_za_klasifikaciju %>% select(-nacin_lansiranja)
rakete_za_klasifikaciju$fiksni_lanser <-
  as.factor(rakete_za_klasifikaciju$fiksni_lanser)
```

Sada treba odrediti klasifikator za ovo obeležje.

Poredиće se dva načina predviđanja unakrsnom validacijom. Prvi je **RandomForest** gde se problem nebalansiranih podataka rašava uvođenjem težina.

```
library(pROC)
library(MLmetrics)
library(randomForest)
rakete_za_klasifikaciju$fiksni_lanser <-
  as.factor(rakete_za_klasifikaciju$fiksni_lanser)

set.seed(126)
foldovi <- createFolds(rakete_za_klasifikaciju$fiksni_lanser, k = 5)
cv_random_forest <- lapply(foldovi, function(fold_indeksi) {
  train_data <- rakete_za_klasifikaciju[-fold_indeksi, ]
  test_data <- rakete_za_klasifikaciju[fold_indeksi, ]

  train_data$fiksni_lanser <- as.factor(train_data$fiksni_lanser)
  test_data$fiksni_lanser <- as.factor(test_data$fiksni_lanser)

  w1 <- sum(train_data$fiksni_lanser == 1)/length(train_data$fiksni_lanser)
  w0 <- 1
```

```
  rf_model <- randomForest(fiksni_lanser ~ ., data = train_data, mtry = 3,
                           ntree = 500, classwt = c(w0,w1))

  test_predikcije <- predict(rf_model, test_data, type = "prob")[, 2]

  auc <- roc(test_data$fiksni_lanser, test_predikcije)$auc
  binarne_predikcije <- ifelse(test_predikcije > 0.5, 1, 0)
  f1 <- F1_Score(test_data$fiksni_lanser, binarne_predikcije)
  tacnost <- mean(binarne_predikcije == test_data$fiksni_lanser)
  tabela <- matrix(c(auc, tacnost, f1))

  return(tabela)
})

vektor <- 1:15
auc <- mean(unlist(cv_random_forest)[vektor %% 3 == 1])
tacnost <- mean(unlist(cv_random_forest)[vektor %% 3 == 2])
f1 <- mean(unlist(cv_random_forest)[vektor %% 3 == 0])

rezultati <- data.frame(
  Metrika = c("Tačnost", "AUC", "F1-score"),
  Srednji_R2 = c(tacnost, auc, f1)
)
print(rezultati)
```

```
##    Metrika Srednji_R2
## 1  Tačnost  0.9500000
## 2      AUC  1.0000000
## 3 F1-score  0.9712821
```

Nebalansiranost podataka možemo rešiti i na drugi način, pomoću funkcije SMOTE iz paketa smotefamily.

SMOTE (Synthetic Minority Oversampling Technique) je metoda za rešavanje problema nebalansiranih podataka, koja radi tako što generiše sintetičke primere za manjinsku klasu na osnovu postojećih primera, koristeći interpolaciju između k najbližih suseda, čime se povećava zastupljenost manjinske klase i balansira skup podataka, što može poboljšati tačnost i generalizaciju modela.

```
library(smotefamily)
balansirani_podaci <- SMOTE(rakete_za_klasifikaciju[,-12],
                            rakete_za_klasifikaciju[,12], K=3)
balansirani_podaci <- balansirani_podaci$data
```

Na balansiranim podacima napraviću Ridge model.

```
set.seed(126)
foldovi <- createFolds(balansirani_podaci$class, k = 13)

balansirani_podaci$class <- as.numeric(balansirani_podaci$class)

library(glmnet)
cv_results_ridge <- lapply(foldovi, function(fold_indeksi) {
  train_data <- balansirani_podaci[-fold_indeksi, ]
  test_data <- balansirani_podaci[fold_indeksi, ]

  X_train <- as.matrix(train_data[, -12])
  y_train <- train_data$class
  X_test <-  as.matrix(test_data[, -12])
  y_test <- test_data$class



  ridge_model <- cv.glmnet(X_train, y_train, alpha = 0)
  najbolje_lambda <- ridge_model$lambda.min
  final_ridge_model <- glmnet(X_train, y_train, alpha = 1, lambda = najbolje_lambda)
  predikcije <- predict(final_ridge_model, X_test, type = "response")
  binarne_predikcije <- ifelse(predikcije > 0.5, 1, 0)
```

```
  tacnost <- mean(binarne_predikcije == y_test)
  f1 <- F1_Score(y_test, binarne_predikcije, positive = "1")
  roc_curve <- roc(response = y_test, predictor = as.vector(predikcije))
  auc <- auc(roc_curve)

  return(c(auc,tacnost,f1))
})

vektor <- 1:15

auc <- mean(unlist(cv_results_ridge)[vektor %% 3 == 1])
tacnost <- mean(unlist(cv_results_ridge)[vektor %% 3 == 2])
f1 <- mean(unlist(cv_results_ridge)[vektor %% 3 == 0])

rezultati <- data.frame(
  Metrika = c("Tačnost", "AUC", "F1-score"),
  Mean_R2 = c(tacnost, auc, f1)
)
print(rezultati)
```

```
##    Metrika   Mean_R2
## 1  Tačnost 0.9807692
## 2      AUC 1.0000000
## 3 F1-score 0.9743590
```

Malo boljie rezultate daje Ridge model.

```
X <- as.matrix(balansirani_podaci[, -12])
y <- balansirani_podaci$class
ridge_model <- cv.glmnet(X, y, alpha = 0)
najbolje_lambda <- ridge_model$lambda.min
finalni_ridge_model <- glmnet(X, y, alpha = 1, lambda = najbolje_lambda)
```

Klasterizacija

Sve podatke ću staviti u novu tabelu.

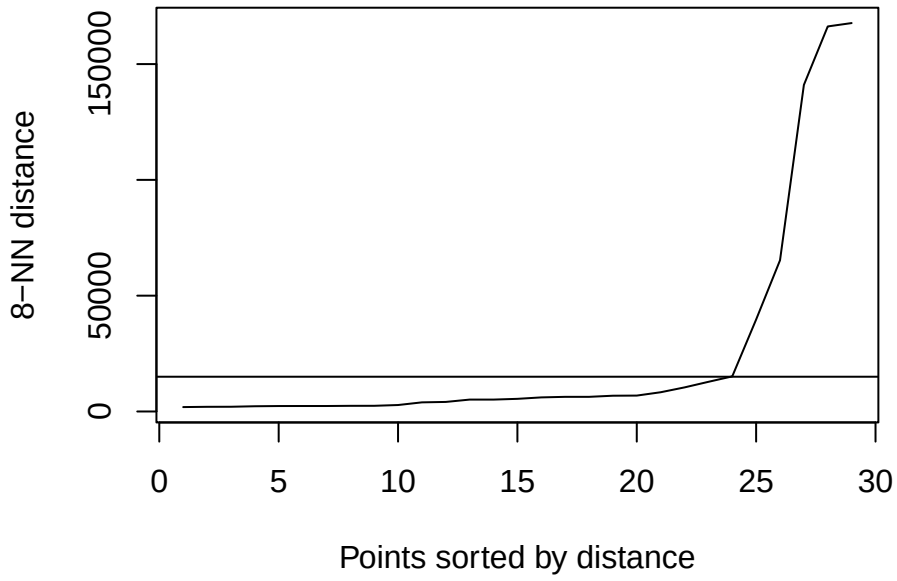
```
rakete_za_klasterizaciju <- rakete[rakete$klasa_po_dometu != 2,]
rakete_za_klasterizaciju <- rakete_za_klasterizaciju %>% select(-max_domet_km)
rakete_za_klasterizaciju <- rakete_za_klasterizaciju %>% select(-klasa_po_dometu)
rakete_za_klasterizaciju <- rakete_za_klasterizaciju %>% select(-NATO_ime)
rakete_za_klasterizaciju <- rakete_za_klasterizaciju %>% select(-rusko_ime)
```

Sada ću za ovako sređene podatke da nađem klastere.

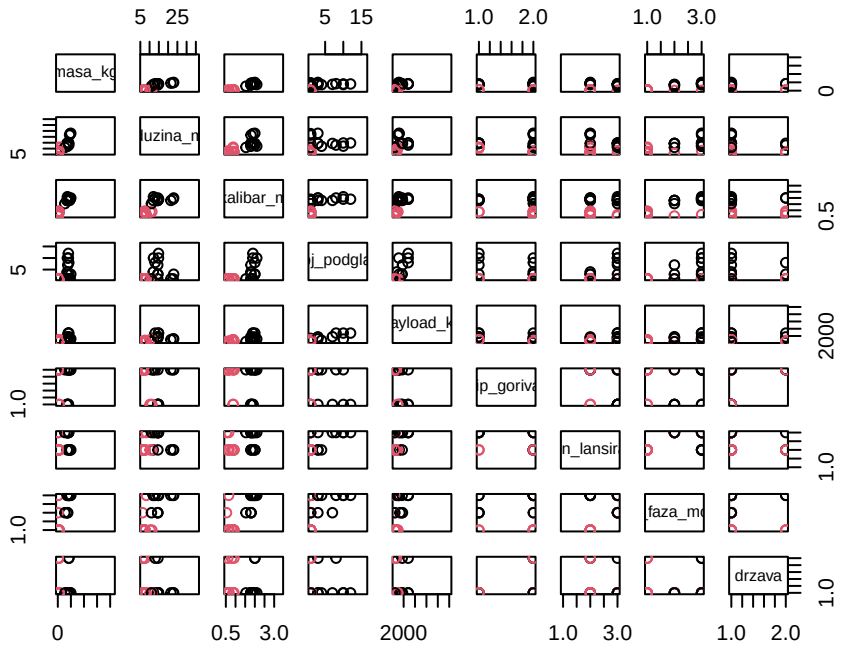
```
library(dbscan)
```

```
##
## Attaching package: 'dbscan'
## The following object is masked from 'package:stats':
##
##     as.dendrogram
```

```
kNNdistplot(rakete_za_klasterizaciju, k = 8)
abline(h=15000)
```



```
dbscan_model <- dbscan::dbscan(rakete_za_klasterizaciju, eps = 15000, minPts = 9)
plot(rakete_za_klasterizaciju, col = dbscan_model$cluster)
```




```
dbscan_model$cluster
```

```
##  [1] 0 0 1 1 1 1 1 1 1 1 1 0 2 2 2 2 2 1 2 2 2 1 2 2 1 0 1 0 2
```

Posmatrajmo sada rakete u klasterima.

```
prvi_klaster <- rakete_za_klasterizaciju[dbscan_model$cluster == 1, ]
drugi_klaster <- rakete_za_klasterizaciju[dbscan_model$cluster == 2,]
summary(prvi_klaster)
```

```
##     masa_kg         duzina_m       kalibar_m     broj_podglava
##  Min.   :26900   Min.   :11.00   Min.   :1.540   Min.   : 1.000
##  1st Qu.:36000   1st Qu.:13.00   1st Qu.:1.800   1st Qu.: 1.000
##  Median :40300   Median :14.40   Median :1.900   Median : 3.000
##  Mean   :39600   Mean   :15.58   Mean   :1.872   Mean   : 4.923
##  3rd Qu.:42000   3rd Qu.:14.80   3rd Qu.:2.000   3rd Qu.: 8.000
##  Max.   :49600   Max.   :23.00   Max.   :2.100   Max.   :12.000
##    payload_kg     tip_goriva    nacin_lansiranja broj_faza_motora
##  Min.   : 450   Min.   :1.000   Min.   :2.000    Min.   :2.000
```

```
##  1st Qu.:1150   1st Qu.:1.000   1st Qu.:2.000    1st Qu.:2.000
##  Median :1350   Median :2.000   Median :3.000    Median :3.000
##  Mean   :1596   Mean   :1.615   Mean   :2.615    Mean   :2.692
##  3rd Qu.:1800   3rd Qu.:2.000   3rd Qu.:3.000    3rd Qu.:3.000
##  Max.   :2800   Max.   :2.000   Max.   :3.000    Max.   :3.000
##      drzava
##  Min.   :1.000
##  1st Qu.:1.000
##  Median :1.000
##  Mean   :1.154
##  3rd Qu.:1.000
##  Max.   :2.000
```

```
summary(drugi_klaster)
```

```
##     masa_kg        duzina_m        kalibar_m      broj_podglava   payload_kg
##  Min.   :1570   Min.   : 6.100   Min.   :0.5340   Min.   :1     Min.   :450.0
##  1st Qu.:2005   1st Qu.: 6.400   1st Qu.:0.6500   1st Qu.:1     1st Qu.:480.0
##  Median :3800   Median : 7.300   Median :0.8000   Median :1     Median :482.0
##  Mean   :3281   Mean   : 7.677   Mean   :0.7617   Mean   :1     Mean   :618.1
##  3rd Qu.:4190   3rd Qu.: 7.875   3rd Qu.:0.8850   3rd Qu.:1     3rd Qu.:750.0
##  Max.   :5900   Max.   :11.250   Max.   :0.9200   Max.   :1     Max.   :985.0
##    tip_goriva    nacin_lansiranja broj_faza_motora     drzava
##  Min.   :1.000   Min.   :2.000    Min.   :1.000    Min.   :1.000
##  1st Qu.:2.000   1st Qu.:2.000    1st Qu.:1.000    1st Qu.:1.000
##  Median :2.000   Median :2.000    Median :1.000    Median :1.000
##  Mean   :1.818   Mean   :2.182    Mean   :1.273    Mean   :1.273
##  3rd Qu.:2.000   3rd Qu.:2.000    3rd Qu.:1.000    3rd Qu.:1.500
##  Max.   :2.000   Max.   :3.000    Max.   :3.000    Max.   :2.000
```

Rakete u prvom klasteru su značajno teže i duže, imaju veći kalibar i veći broj podglava i broj motora. Sve ukazuje na to da su ove rakete veće i snažnije, pa možemo pretpostaviti da klasteri predstavljaju domet raketa.

Ostaje nam da posmatramo tačke šuma.

```
noise_points <- rakete_za_klasterizaciju[dbscan_model$cluster == 0, ]
summary(noise_points)
```

```
##     masa_kg          duzina_m       kalibar_m    broj_podglava    payload_kg
##  Min.   : 80000   Min.   :22.00   Min.   :2.25   Min.   : 6.0   Min.   : 2500
##  1st Qu.:105600   1st Qu.:27.00   1st Qu.:2.50   1st Qu.:10.0   1st Qu.: 3355
##  Median :183000   Median :32.20   Median :3.00   Median :10.0   Median : 4000
##  Mean   :157260   Mean   :29.86   Mean   :2.83   Mean   :10.4   Mean   : 5731
##  3rd Qu.:208100   3rd Qu.:32.60   3rd Qu.:3.05   3rd Qu.:10.0   3rd Qu.: 8800
##  Max.   :209600   Max.   :35.50   Max.   :3.35   Max.   :16.0   Max.   :10000
##    tip_goriva  nacin_lansiranja broj_faza_motora     drzava
##  Min.   :1.0   Min.   :1.0      Min.   :2.0      Min.   :1.0
##  1st Qu.:1.0   1st Qu.:1.0      1st Qu.:2.0      1st Qu.:1.0
##  Median :1.0   Median :1.0      Median :2.0      Median :1.0
##  Mean   :1.2   Mean   :1.2      Mean   :2.4      Mean   :1.4
##  3rd Qu.:1.0   3rd Qu.:1.0      3rd Qu.:3.0      3rd Qu.:2.0
##  Max.   :2.0   Max.   :2.0      Max.   :3.0      Max.   :2.0
```

Od 5 tačaka šuma 4 imaju vrednost 1 za način lansiranja, tj. njihov način lansiranja Silos. Takođe imaju veliki broj podglava, duže su od ostalih i imaju veći kalibar, a i teže su značajno. Po svim osobinama one liče na podatke iz klastera 1, tj. deluju kao da su još jače i snažnije od ovih i samim tim imaju i veći domet od onih iz klastera 2.

```
rakete_za_klasterizaciju$klasterizacija <- dbscan_model$cluster
rakete_za_klasterizaciju$klasterizacija[dbscan_model$cluster==0] <- 1
print(rakete_za_klasterizaciju$klasterizacija)
```

```
##  [1] 1 1 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 1 2 2 2 1 2 2 1 1 1 1 2
```

Sažetak o političkim situacijama i razvoju oružja (raketa) u Rusiji i Kini (1988-2018)

Ključna imena i događaji:

1. Ronald Reagan i Mihail Gorbačev Potpisivanje INF sporazuma (1987): Sporazum o nuklearnim snagama srednjeg dometa između SAD-a i Sovjetskog Saveza bio je presudan događaj u kontroli naoružanja. Sporazum je stupio na snagu 1988. godine i zahtevao je eliminaciju svih kopnenih balističkih i krstarećih raketa dometa od 500 do 5.500 kilometara. Kao rezultat, uništene su hiljade sovjetskih raketa, što je označilo kraj razvoja srednjedometnih raketa u tom periodu.

2. Raspad Sovjetskog Saveza (1991) Nakon raspada, Rusija je nasledila većinu infrastrukturnih kapaciteta za razvoj raketa, ali se suočila s ekonomskim izazovima koji su ograničili vojna ulaganja. Fokus je preusmeren na interkontinentalne balističke rakete (ICBM) i krstareće rakete koje nisu bile obuhvaćene INF sporazumom.

3. Obnova sumnji na kršenje INF sporazuma (2010-te) Do 2010-ih, SAD su optužile Rusiju za razvoj raketa koje krše sporazum. Rusija je negirala optužbe, ali je tvrdila da su američki raketni sistemi u Evropi takođe u suprotnosti sa sporazumom.

4. Kolaps INF sporazuma (2018-2019) Godine 2018, SAD su najavile povlačenje iz sporazuma zbog ruskih kršenja. Povlačenje je završeno 2019. godine, što je označilo kraj restrikcija za razvoj i raspoređivanje srednjedometnih raketa.

Politički kontekst i uticaj na proizvodnju oružja:

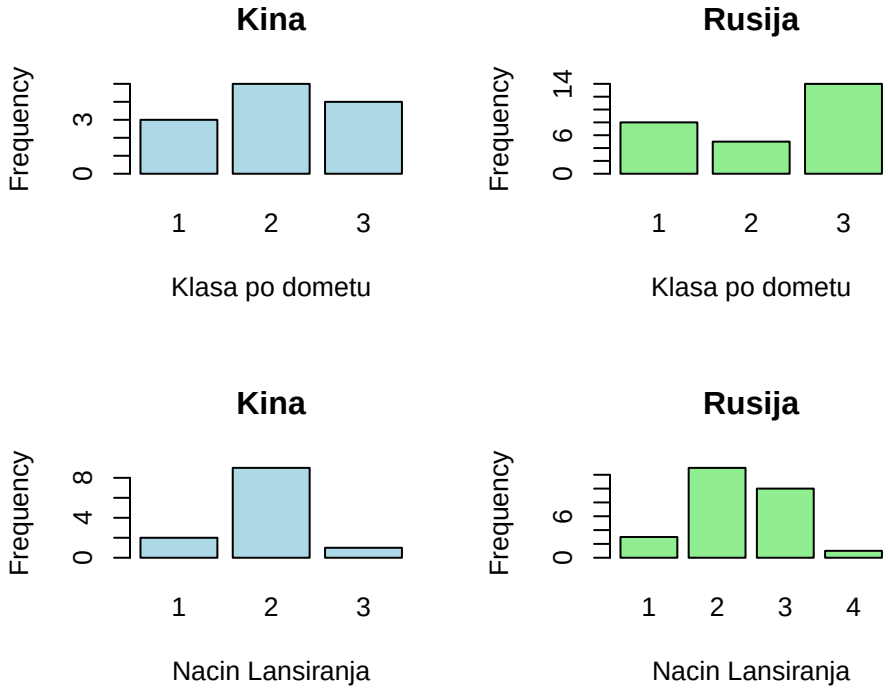
1. Rusija (1988-2018) U okviru INF sporazuma, Rusija je obustavila razvoj srednjedometnih kopnenih raketa. Ekonomske poteškoće 1990-ih usporile su modernizaciju, ali je ponovni rast finansiranja 2000-ih doveo do razvoja naprednih sistema, uključujući hipersonične projektile.

2. Kina (istorijski razvoj): Kina nije bila deo INF sporazuma i nastavila je s razvojem srednjedometnih raketa tokom ovog perioda (Dongfeng serija (DF)). Kina je razvila raznovrsne balističke rakete, poput DF-21 (srednjeg dometa) i DF-26, koje su omogućile projekciju sile u Pacifičkom regionu. Fokus na regionalnim kapacitetima omogućio je Kini stratešku prednost u Istočnoj Aziji, posebno u odnosu na Tajvan i američke baze u regionu.

Značaj političkih dogovora na vojnu industriju: INF sporazum (1988): Direktno je zaustavio razvoj kopnenih raketa srednjeg dometa u Rusiji i SAD-u, što je uticalo na globalni balans snaga. Raspad sporazuma (2018): Omogućio je ponovni razvoj srednjedometnih sistema, što je dovelo do povećane napetosti između SAD-a, Rusije i Kine.

Kroz bazu podataka možemo uočiti ovu i neke druge specifičnosti između ove dve države.

```
##
## Attaching package: 'tidyr'
## The following objects are masked from 'package:Matrix':
##
##     expand, pack, unpack
```



Zašto Rusija ima više raketa lansiranih iz vode od Kine (do 2018)?

1. Sovjetski Savez Tokom Hladnog rata bio je globalna supersila sa razvijenom mornaricom i značajnim ulaganjem u podmorničke nuklearne snage. Zašto?

2. Geopolitika i strategijska doktrina Rusija (kao deo Sovjetskog saveza) je površinski velika zemlja sa razvijenim uticajem i projekcijom snage u svetskim okeanima (Arktik, Pacifik, Crno more). Voda je prirodni prostor za pozicioniranje mornaričkih snaga, jer omogućava globalni domet i fleksibilnost, čak i u slučaju ugroženosti kopnenih baza.

3. Kina: Fokus na regionalne kapacitete Kina je tradicionalno bila kontinentalna sila s prioritetom na kopnenim snagama. Modernizacija mornarice započela je tek krajem 20. veka. Većina raketa bila je namenjena za kopnenu upotrebu ili regionalne ciljeve (DF serija), jer je njen fokus bio na sukobe u Pacifičkom regionu (npr. Tajvan, Južnokinesko more). Dakle, Kina is-

torijski ima manje iskustva s pomorskim nuklearnim sistemima jer joj je fokus na regionalnoj dominaciji, a ne na globalnoj projekciji sile.

4. Tehnološke i finansijske prednosti Rusije Rusija je nasledila sovjetsku infrastrukturu i stručnost, što je omogućilo kontinuirani razvoj mornaričkih sistema. Kina je do 2018. još uvek bila u fazi "sustizanja" Rusije i SAD-a u tehnološkom smislu, ali je nakon 2018. ubrzala razvoj mornaričkih sistema.