Uvod u R

Wednesday, May 31, 2023 12:49 PM

Numeric - svi brojevi
Integer - celobrojni samo
3L
A %/% B
A %% B

String
Substr(ime, 1, 1)
Nchar(ime)

Logical
A & B, A | B, !A

Factor
Faktor<-factor(c("A","B","A"))
Levels(faktor)

razliciti novoi

If (uslov){
Print("Tekst");
} else {
}

For (i in 1:5){
Print(i)
}

ProveriParnost <- **function**(broj){
Return broj % 2 == 0
}

Rezultat <- ProveriParnost(5)

Vracanje vise vrednosti:
Return(c(1,2,3))

Matrica:
matrica<-matrix(c(1,2,3,4,5,6), nrow = 2, ncol = 3, byrow = TRUE)
el<-matrica[1,1]
red2<-matrica[2,]
kol3<-matrica[,3]

Niz:
prviEl<-niz[1]
length(niz)
…sum, mean, min, max, sort,
unique

sekvNiz<-seq(1,10,by=2)
ponavlj<-rep(0, times = 5)

PRIPREMA - Sablon

Wednesday, May 31, 2023 1:56 PM

PRIPREMA PODATAKA:
CSV
.csv fajl se samo nalepi u project folder
data<-read.csv("naziv", stringsAsFactors = F)
ISLR
install.packages("ISLR")
library("ISLR")
?College
data<-College

str(data)
summary(data)
data$reviews

koloni reviews
tabele data

percent75<-quantile(data$reviews, 0.75)

sum(is.na(data$price))

jel ima NA

sum(is.na(data$Price==""), na.rm=T)

jel ima praznih (SKLONI NA)

data$price(data$price==="-")<-NA

zameni crtice sa NA

shapiro.test(data$price)
shapiro.test(sample(data$price.size=5000))

H0 - promenljiva ima N raspodelu
ako je p<0.005 odbacujemo H0
(*mean se koristi*)
p>0.005 prihvatamo H0
(*median se koristi*)

med<-median(data$price, na.rm=T)
data$price[is.na(data$price)]<-med

apply(podskup, MARGIN = 2, FUN =
function(x) sum(is.na(x)))

length(unique(data$current.version))
data$current.ver<-NULL

brisanje kolone
brisemo sve irelevantne kolone
one sto imaju brda razl vrednosti
i one koje su vec koriscene za
izlazne varijable

podskup$category<-as.factor(podskup
$factory)

ostale pretvaramo u faktor

Kreiranje izlazne varijable
percent75<-quantile(data$reviews, 0.75)
data$novRed<-ifelse(data$reviews > percent75, yes =
"yes", no = "no")

podskup<-subset(data, (data$price<=350))

KLASIFIKACIONA STABLA

Thursday, June 1, 2023 11:19 AM

KORACI - SREDJIVANJE PODATAKA

```
data<-read.csv("imeFajla.csv", stringsAsFactors = F)

podskup <- subset(data, (data$country = "Argentina"))

data<-podskup

 apply(data, MARGIN = 2, FUN = function(x) sum(is.na(x)))
                                    -| |-     sum(x=="-")
```

Popunjavanje vrednosti:

Numericke:
Shapiro test? -> median / mean
String:
Previse vrednosti? -> brisemo ili factor → menjamo najzastupljenijom vrednoscu

```
shapiro.test(data$price)  # p<0.005
medijana<-median(data$numerickaVarijabla, na.rm = T)
data$price[is.na(data$price)]<-medijana

length(unique(data$region)) #ima previse vrednosti
data$region<-NULL

length(---)
data$country <- as.factor(data$country)
```

KROSVALIDACIJA

```
library(caret)
library(e1071)
umFolds<-trainControl(method = "cv", number = datUZadatku)
cpGrid<-expand.grid(.cp = seq(from = 0.001,
                              to = 0.05,
                              by = 0.001))
set.seed(500)
crossvalidation <- train(x = train.data(,-rbIzlazne],
                         y=train.data$izlazna,
                         method = "rpart",
                         trControl = numFolds,
                         tuneGrid = cpGrid)
plot(crossvalidation)
cpValue <-crossvalidation$bestTune$cp
```

ispisuje optimalan CP

```
percentil30 <-quantile(data$price, 0.3)
data$price_category <- ifelse(data$price<=percentil30,
                              yes = "cheap", no = "not_cheap")
data$price<-NULL
data$price_category<-as.factor(---)
```

Podela na trening i test

```
library(caret)
set.seed(1010)
indexes<-createDataPartition(data$izlazna, p=0.8, list = FALSE)

train.data<-data[indexes,]
test.data<-data[-indexes,]
```

uzima random 80% sadrzaja

pravi drvo odlucivanja od ostalih varijabli za izlaznu

```
library(rpart)
tree1 <- rpart(izlazna ~ ., data = train.data, method ="class")

library(rpart.plot)
rpart.plot(tree1, extra = 104)
```

graficki prikaz

matrica konfuzije za proveru predikcija na test podacima

```
tree1.pred <- predict(tree1, newdata = test.data, type = "class")
tree1.cm <- table(true = test.data$izlazna, predicted = tree1.pred)

getEvalMetrics<-function(tree1.cm)
```

FUNKCIJA MATRICE KONFUZIJE

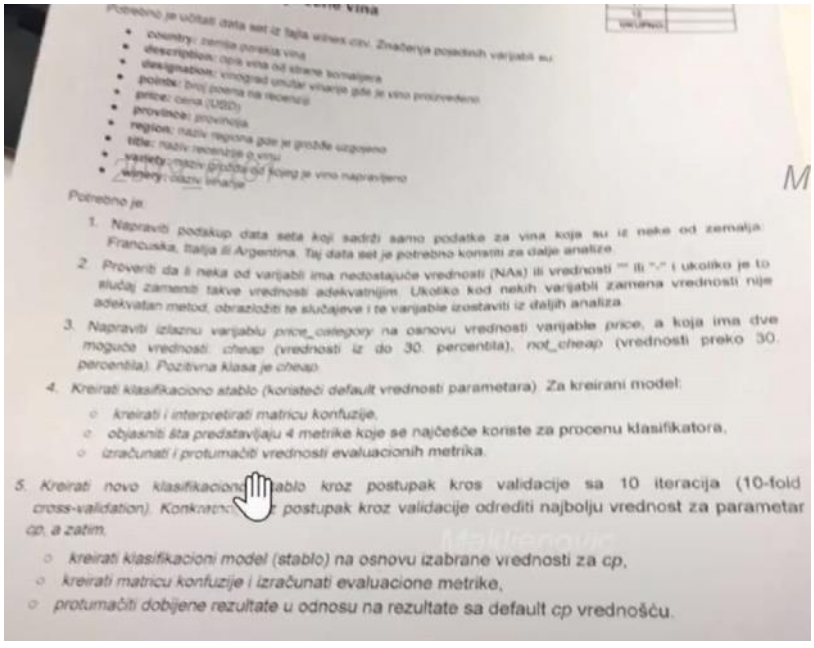
```
getEvalMetrics<-function(cm){
TP<-cm[2,2]
TN<-cm[1,1]
FP<-cm[1,2]
FN<-cm[2,1]
acc<-sum(diag(cm))/sum(cm)
prec<- TP / (TP + FP)
rec<- TP / (TP + FN)
F! <- (2*prec*rec) / (prec+rec)
c(Accuracy = acc, Precision = prec, Recall = rec,
      F1 = F1)
```

TP - true positive
FN - false negative itd
naglasice koja kategorija je positive a koja negative

vraca sve cetri ove vrednosti

```
tree2 <- rpart(price_category ~ ., data = train.data, method = "class",
               control = rpart.control(cp = cpValue)

rpart.plot(tree2, extra = 104)
tree2.pred <- predict(---)
tree2.cm <- table(---)
eval.tree2<-getEvalMetrics(cm)
```



Potrebno je:

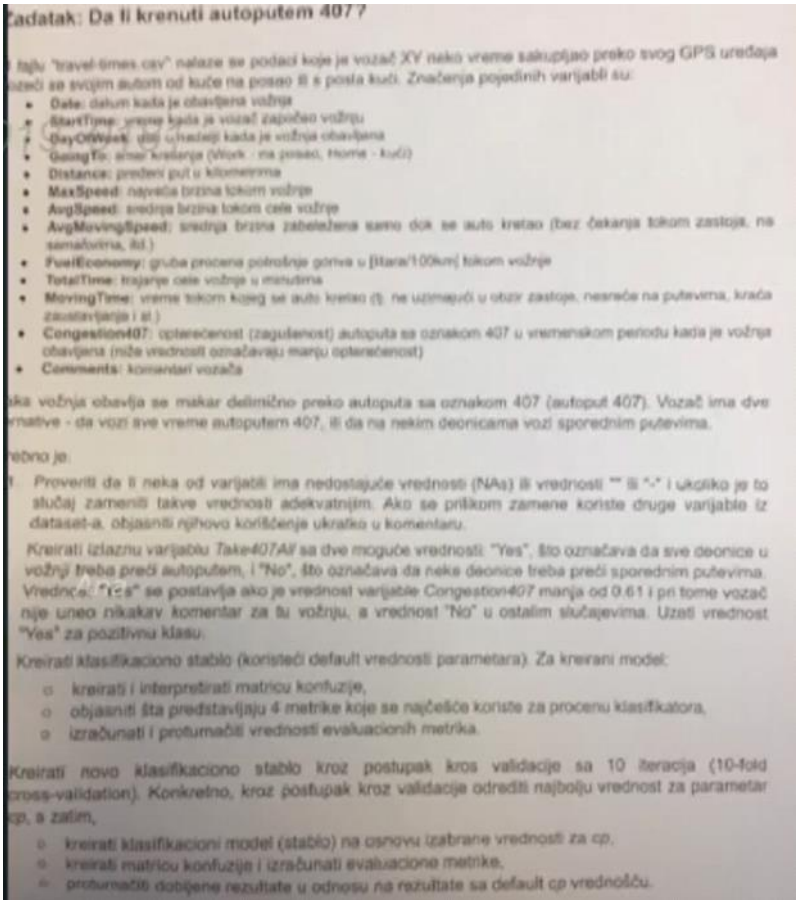
1. Napraviti podskup data seta koji sadrži samo podatke za vina koja su iz neke od zemalja: Francuska, Italija ili Argentina. Taj data set je potrebno koristiti za dalje analize.
2. Proveriti da li neka od varijabli ima nedostajuće vrednosti (NAs) ili vrednosti "" ili "-" i ukoliko je to slučaj zameniti takve vrednosti adekvatnijim. Ukoliko kod nekih varijabli zamena vrednosti nije adekvatan metod, obrazložiti te slučajeve i te varijable izostaviti iz daljih analiza.
3. Napraviti izlaznu varijablu *price_category* na osnovu vrednosti varijable *price*, a koja ima dve moguće vrednosti: *cheap* (vrednosti iz do 30. percentila), *not_cheap* (vrednosti preko 30. percentila). Pozitivna klasa je *cheap*.
4. Kreirati klasifikaciono stablo (koristeći default vrednosti parametara). Za kreirani model:
   - kreirati i interpretirati matricu konfuzije,
   - objasniti šta predstavljaju 4 metrike koje se najčešće koriste za procenu klasifikatora,
   - izračunati i protumačiti vrednosti evaluacionih metrika.
5. Kreirati novo klasifikaciono stablo kroz postupak kros validacije sa 10 iteracija (10-fold cross-validation). Konkretno, postupak kroz validacije odrediti najbolju vrednost za parametar *cp*, a zatim
   - kreirati klasifikacioni model (stablo) na osnovu izabrane vrednosti za *cp*,
   - kreirati matricu konfuzije i izračunati evaluacione metrike,
   - protumačiti dobijene rezultate u odnosu na rezultate sa default *cp* vrednošću.

KS2 - autoput

Thursday, June 1, 2023 1:22 PM

```
data<-read.csv("imeFajla.csv", stringsAsFactors = F)
str(data)
PRIPREMA PODATAKA
apply(data, MARGIN = 2, FUN = function(x) sum(is.na(x)))
---

                                                    bio je char
data$fuelEconomy <- as.Numeric(data$FuelEconomy)
shapiro.test(---)
medijana---
#popunjavanje NA polja
```



Zadatak: Da li krenuti autoputem 407?

U fajlu "travel-times.csv" nalaze se podaci koje je vozač XY neko vreme sakupljao preko svog GPS uređaja dok je išao svojim autom od kuće na posao ili s posla kući. Značenja pojedinih varijabli su:

- **Date**: datum kada je obavljena vožnja
- **StartTime**: vreme kada je vozač započeo vožnju
- **DayOfWeek**: dan u nedelji kada je vožnja obavljena
- **GoingTo**: smer kretanja (Work - na posao, Home - kući)
- **Distance**: predeni put u kilometrima
- **MaxSpeed**: najveća brzina tokom vožnje
- **AvgSpeed**: srednja brzina tokom cele vožnje
- **AvgMovingSpeed**: srednja brzina zabeležena samo dok se auto kretao (bez čekanja tokom zastoja, na semaforima, itd.)
- **FuelEconomy**: gruba procena potrošnje goriva u [litara/100km] tokom vožnje
- **TotalTime**: trajanje cele vožnje u minutima
- **MovingTime**: vreme tokom kojeg se auto kretao (tj. ne uzimajući u obzir zastoje, nesreće na putevima, kraća zaustavljanja i sl.)
- **Congestion407**: opterećenost (zagušenost) autoputa sa oznakom 407 u vremenskom periodu kada je vožnja obavljena (niže vrednosti označavaju manju opterećenost)
- **Comments**: komentari vozača

Svaka vožnja obavlja se makar delimično preko autoputa sa oznakom 407 (autoput 407). Vozač ima dve alternative - da vozi sve vreme autoputem 407, ili da na nekim deonicama vozi sporednim putevima.

Potrebno je:

1. Proveriti da li neka od varijabli ima nedostajuće vrednosti (NAs) ili vrednosti "" ili "-" i ukoliko je to slučaj zameniti takve vrednosti adekvatnijim. Ako se prilikom zamene koriste druge varijable iz dataset-a, objasniti njihovo korišćenje ukratko u komentaru.

Kreirati izlaznu varijablu *Take407All* sa dve moguće vrednosti: "Yes", što označava da sve deonice u vožnji treba preći autoputem, i "No", što označava da neke deonice treba preći sporednim putevima. Vrednost "Yes" se postavlja ako je vrednost varijable *Congestion407* manja od 0.61 i pri tome vozač nije uneo nikakav komentar za tu vožnju, a vrednost "No" u ostalim slučajevima. Uzeti vrednost "Yes" za pozitivnu klasu.

Kreirati klasifikaciono stablo (koristeći default vrednosti parametara). Za kreirani model:

- kreirati i interpretirati matricu konfuzije,
- objasniti šta predstavljaju 4 metrike koje se najčešće koriste za procenu klasifikatora,
- izračunati i protumačiti vrednosti evaluacionih metrika.

Kreirati novo klasifikaciono stablo kroz postupak kros validacije sa 10 iteracija (10-fold cross-validation). Konkretno, kroz postupak kroz validacije odrediti najbolju vrednost za parametar cp, a zatim,

- kreirati klasifikacioni model (stablo) na osnovu izabrane vrednosti za cp,
- kreirati matricu konfuzije i izračunati evaluacione metrike,
- protumačiti dobijene rezultate u odnosu na rezultate sa default cp vrednošću.

1.

```
data$take407All <- ifelse(data$Congestion < 0.61 & data&Comments == "",
                          yes = "Yes", no = "No")
#pretvaranje u faktor
#comments i congestion postaju NULL

Podela na trening i test
library(caret)
#seed
indexes<-
train.data<-
test.data<-

Kreiranje stabla
library(rpart)
tree1<-rpart(---)
library(rpart.plot)
rpart.plot(---)

Predikcije i matrica konfuzije
tree1.pred<-predict(---)
tree1.cm <- table(---)
eval.tree1<-getEvalMetrics(---)

Krosvalidacija
library(caret)
numFolds<-
cpGrid<-
#seed
crossv<-train(---)
cpValue(crossv$bestTune$cp
```

KS3 - apps

Friday, June 2, 2023 5:03 PM

perc75 <- quantile(data$reviews, 0.75)
data$highReviews<-ifelse(data$reviews >= perc75, yes = "yes", no = "no")

data$reviews<-null

dara$price<-as.numeric(data$price)
shapiro.test(---)
data$price[is.na(data$price)]<-medijana

podskup <- subset(data, data$price<=350)

length(unique(data$---))
data$...<-as.factor(data$...)…<-NULL
ili
data$

apply(data, MARGIN = 2, FUN = function(x) sum(is.na(x)))

indexes<-createDataPartition
train.data<-…
test.data<-…

KROSVALIDACIJA

numFolds<-trainContrtol(…)
cpGrid<-expand.grid(…)

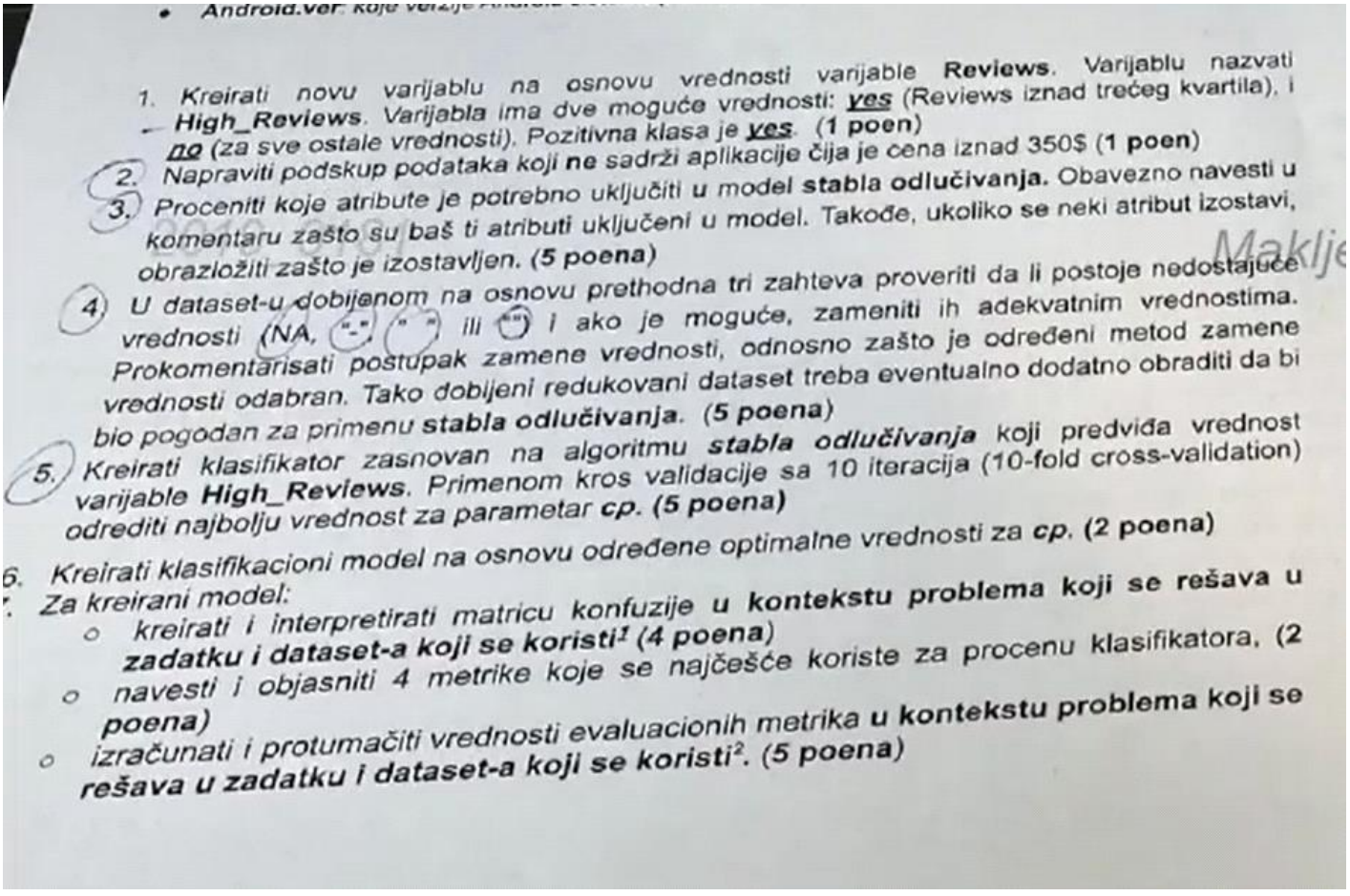
crossv<-train(…)
cpValue<-crossv$bestTune$cp

tree<-rpart(….cpValue)
rpart.plot(tree…)

tree.pred<-predict(…)

tree.cm<-table(…)

eval.tree<- getEvalMetrics(tree…)

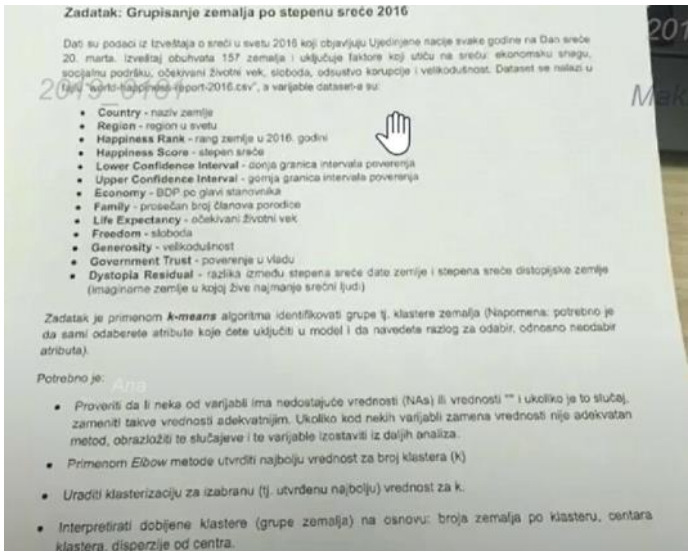


Android.ver: koje verzije

1. Kreirati novu varijablu na osnovu vrednosti varijable **Reviews**. Varijablu nazvati **High_Reviews**. Varijabla ima dve moguće vrednosti: ***yes*** (Reviews iznad trećeg kvartila), i ***no*** (za sve ostale vrednosti). Pozitivna klasa je ***yes***. **(1 poen)**
2. Napraviti podskup podataka koji **ne** sadrži aplikacije čija je cena iznad 350$ **(1 poen)**
3. Proceniti koje atribute je potrebno uključiti u model **stabla odlučivanja.** Obavezno navesti u komentaru zašto su baš ti atributi uključeni u model. Takođe, ukoliko se neki atribut izostavi, obrazložiti zašto je izostavljen. **(5 poena)**
4. U dataset-u dobijenom na osnovu prethodna tri zahteva proveriti da li postoje nedostajuće vrednosti (NA, "-", " ", ili "") i ako je moguće, zameniti ih adekvatnim vrednostima. Prokomentarisati postupak zamene vrednosti, odnosno zašto je određeni metod zamene vrednosti odabran. Tako dobijeni redukovani dataset treba eventualno dodatno obraditi da bi bio pogodan za primenu **stabla odlučivanja**. **(5 poena)**
5. Kreirati klasifikator zasnovan na algoritmu ***stabla odlučivanja*** koji predviđa vrednost varijable **High_Reviews**. Primenom kros validacije sa 10 iteracija (10-fold cross-validation) odrediti najbolju vrednost za parametar **cp. (5 poena)**
6. Kreirati klasifikacioni model na osnovu određene optimalne vrednosti za **cp. (2 poena)**
7. Za kreirani model:
   - kreirati i interpretirati matricu konfuzije **u kontekstu problema koji se rešava u zadatku i dataset-a koji se koristi[1] (4 poena)**
   - navesti i objasniti 4 metrike koje se najčešće koriste za procenu klasifikatora, **(2 poena)**
   - izračunati i protumačiti vrednosti evaluacionih metrika **u kontekstu problema koji se rešava u zadatku i dataset-a koji se koristi[2]. (5 poena)**

K-MEANS KLASTEROVANJE

Friday, June 2, 2023 5:46 PM



Zadatak: Grupisanje zemalja po stepenu sreće 2016

Dati su podaci iz izveštaja o sreći u svetu 2016 koji objavljuju Ujedinjene nacije svake godine na Dan sreće 20. marta. Izveštaj obuhvata 157 zemalja i uključuje faktore koji utiču na sreću: ekonomsku snagu, socijalnu podršku, očekivani životni vek, slobodu, odsustvo korupcije i velikodušnost. Dataset se nalazi u fajlu "world-happiness-report-2016.csv", a varijable dataseta su:

- **Country** - naziv zemlje
- **Region** - region u svetu
- **Happiness Rank** - rang zemlje u 2016. godini
- **Happiness Score** - stepen sreće
- **Lower Confidence Interval** - donja granica intervala poverenja
- **Upper Confidence Interval** - gornja granica intervala poverenja
- **Economy** - BDP po glavi stanovnika
- **Family** - prosečan broj članova porodice
- **Life Expectancy** - očekivani životni vek
- **Freedom** - sloboda
- **Generosity** - velikodušnost
- **Government Trust** - poverenje u vladu
- **Dystopia Residual** - razlika između stepena sreće date zemlje i stepena sreće distopijske zemlje (imaginarne zemlje u kojoj žive najmanje srećni ljudi)

*Zadatak je primenom **k-means** algoritma identifikovati grupe tj. klastere zemalja (Napomena: potrebno je da sami odaberete atribute koje ćete uključiti u model i da navedete razlog za odabir, odnosno neodabir atributa).*

Potrebno je:

- *Proveriti da li neka od varijabli ima nedostajuće vrednosti (NAs) ili vrednosti "" i ukoliko je to slučaj, zameniti takve vrednosti adekvatnijim. Ukoliko kod nekih varijabli zamena vrednosti nije adekvatan metod, obrazložiti te slučajeve i te varijable izostaviti iz daljih analiza.*
- *Primenom Elbow metode utvrditi najbolju vrednost za broj klastera (k)*
- *Uraditi klasterizaciju za izabranu (tj. utvrđenu najbolju) vrednost za k.*
- *Interpretirati dobijene klastere (grupe zemalja) na osnovu: broja zemalja po klasteru, centara klastera, disperzije od centra.*

trazenje patterna u datasetu
radi samo s numerickim varijablama

Utility.R fajl i excel se stavljaju u folder projekta

data<-read.csv(...)

priprema podataka - sve u numeric

Provera outlayer-a

```
apply(data[,a:B], 2, FUN = function(x) length(boxplot.stats(x)$out))
```

sad trazimo jesu li ekstremno visoke ili ekstremno niske vrednosti

```
boxplot(data$var, xlab= "imeVarijable")
```

za gornji i za donji
ako nema gornjih umesto 0.95 ide 1
ako nema donjih umesto 0.05 ide 0

```
library(DescTools)
Government.Trust_W <- Winsorize(data$Government.Trust, probs = c(0.05, 0.95))
```

ako ne ocisti sve
smanjivati/povecavati
granice dok ih
ne skloni sve

Normalizacija numerickih

```
normalize_var<-function(x){
if (sum(x, na.rm = T) == 0) x
else ((x-min(x, na.rm = T))/(max(x,na.tm = T) - min(x, na.rm = T)))
}
data.norm <- as.data.frame(apply(data[,a:b], 2, normalize_var))
```

taj deo je neophodan
samo ako nisu sve
kolone numericke

Izbacivanje visoko korelisanih varijabli

```
install.packages("corrplot")
library(corrplot)
```

obavezna normalizacija pre

```
data_cor <- cor(data.norm)
corrplot.mixed(data_cor)
```

iterativno izbacujemo one s
debelim krugovima i onda opet

```
data.norm$Happiness.rank<-NULL
```

- | | -

Kreiranje modela

Utvrdjivanje najboljeg broja klastera

Elbow metoda

```
eval.metrics <- data.frame()
for(k in 2:8){
set.seed(1010)
km <- kmeans(data.norm, centers = k, iter.max = 20, nstart = 1000)
eval.metrics <- rbind(eval.metrics, c(k, km$tot.withinss, km$betweenss/km$totss))
}
colnames(eval.metrics) <- c("clusters", "tot.withinss", "ratio")
```

kolko su medjusobno
klasteri razliciti

zbir udaljenosti elemenata centra klastera

```
source("Utility.R") ime fajla
diff(df<-apply(eval.metrics[,2:3], 2, compute.difference)
diff_df <- cbind(k = 2:8, diff_df)
diff_df  prikazuje opt broj klastera (najveca vrednost iz tabele)
sample.3k <- kmeans(x = data.norm, centers = 3, iter.max = 20, nstart = 1000)

eval.metrics prikazuje ove vrednosti
library(ggplot2)
ggplot(data = eval.metrics, mapping = aes(x = 2:8, y = tot.withinss)) + geom_line() +
geom_point()
```

ILI OVO

ovo

na osnovu grafika ili tabele se utvrdjuje optimalan broj klastera
gleda se najveca razlika izmedju withinss ili razlika u nagibu na graf

```
k = 3
sample.3k <- kmeans(x = data.norm, centers = 3, iter.max = 20, nstart =1000)
prokomentarisati velicine klastera (jesu/nisu priblizno istih velicina)
sum.stats <- summary.stats(data.norm, sample.3k$cluster, 3)
sad treba naci pattern
devijacije ako ne variraju -> medjusobno su slicne
```

```
#Prvi klaster: Zemlje u prvom klasteru imaju malo poverenja u vladu, dok su ostali rezultati
#(Economy, Family, Life.Expectancy, Freedom, Generosity, Dystopia.Residual) između rezultata
#druga dva klastera. Na osnovu toga možete zaključiti da su ove zemlje verovatno srednjeg statusa.

#Drugi klaster: Zemlje u drugom klasteru imaju nisku vrednost poverenja u vladu. Takođe, imaju niže
#vrednosti za ekonomiju (Economy), porodične odnose (Family), očekivani životni vek (Life.Expectancy)
#i BDP po glavi stanovnika (GDP per capita). Možete zaključiti da su ove zemlje verovatno siromašnije i
#nižeg socijalno-ekonomskog statusa.

#Treći klaster: Zemlje u trećem klasteru imaju najviše vrednosti za sve varijable osim za Dystopia.Residual,
#gde je vrednost niža. Možete zaključiti da su ove zemlje verovatno bogatije, višeg socio-ekonomskog
#statusa i da imaju visok standard života.
```

KM happiness

Sunday, June 4, 2023 7:26 PM

```
data<- read.csv(...)
str(data)

apply(data, MARGIN = 2, FUN = function(x) sum(is.na(x)))
takodje za razmak, prazan string, crtica

podskup <- subset(data, data$Economy.GPD.per.Capita>0)
data<-podskup
```

Brisanje autlejera

```
apply(data, 2, FUN = function(x) length(boxplot.stats(x)$out))
#ovaj ima tolko autlejera, onaj tolko
boxplot(data$Family, xlab = "Family")

library(DescTools)
FamilyW <- Winsorize(data$family, probs = c(0.05, 1) donji autlejer
data$Family <-FamilyW
TrustW<-Winsorize(... = c(0, 0.95) gornji autlejer
data$Trust<-TrustW
```

Normalizacija

```
normalize_var <- .....
data.norm <- as.data.frame(apply(data, 2, normalize_var))
```

Brisanje korelisanih varijabli

```
data_cor <- cor(data.norm)
library(corrplot)
corrplot.mixed(data_cor)
```

Elbow

```
eval.metrics <- data.frame()
for (...)
...
colnames(...)<-...

#odredjivanje optimalnog broja klastera = 3
```

Pravljenje modela

```
sample.3k <- kmeans(x = data.norm, centers = 3, iter,max = 20, nstart = 1000)
sum.stats  <- summary.stats(data.norm, samle.3k$cluster, 3)
#drzave iz prvog klastera su bogatije od ovih iz drugog klastera, ovaj onaj pattern
```

fajlu "world-happiness-report-2017.csv", a varijable dataset-a su:

2019_0181

- **Country** - naziv zemlje
- **Happiness Rank** - rang zemlje u 2017. godini
- **Happiness Score** - stepen sreće
- **Whisker high** - maksimalna vrednost stepena sreće date zemlje (ne računajući outliers)
- **Whisker low** - minimalna vrednost stepena sreće date zemlje (ne računajući outliers)
- **Economy GDP per Capita** - BDP po stanovniku
- **Family** - prosečan broj članova porodice
- **Health Life Expectancy** - očekivani životni vek
- **Freedom** - sloboda
- **Generosity** - velikdušnost
- **Trust Government Corruption** - poverenje u vladu i odsustvo korupcije
- **Dystopia Residual** - razlika između stepena sreće date zemlje i stepena sreće distopijske zemlje (imaginarne zemlje u kojoj žive najmanje srećni ljudi)

Maklj

Zadatak je primenom ***k-means*** algoritma identifikovati grupe tj. klastere instanci.

Potrebno je:

- Potrebno je odabrati atribute koji će biti uključeni u model i navesti razloge za odabir, odnosno neodabir atributa.
- Pripremiti podatke za primenu algoritma. **U analizi koristiti podskup podataka koji sadrži samo podatke za zemlje koje imaju vrednost varijable *Economy* veći od 0**.
- Primenom *Elbow* metode utvrditi najbolju vrednost za broj klastera (k).
- Uraditi klasterizaciju za izabranu (tj. utvrđenu najbolju) vrednost za k.
- Interpretirati dobijene klastere (grupe) na osnovu: broja instanci po klasteru, centara klastera, disperzije od centra.

KM zadatak vezbe

Monday, June 5, 2023 8:07 PM

```
data<-read.csv("wholesale_customers.csv", stringsAsFactors = FALSE)
str(data)
summary(data)
unique(data$Channel)
data$Channel <- as.factor(data$Channel)
unique(data$Region)
data$Region <- as.factor(data$Region)
data
apply(data, MARGIN = 2, FUN = function(x) sum(x=="-"))
data$Region <- NULL
data$Channel <- NULL
str(data)

podskup <- subset(data, data$Frozen>1000)
str(podskup)

apply(data, 2, FUN = function(x) length(boxplot.stats(x)$out))
boxplot(data$Milk, xlab = "Milk")
install.packages("DescTools")
library(DescTools)
data$MilkW <- Winsorize(data$Milk, probs = c(0, 0.935))
boxplot(data$MilkW)
data$Milk <- data$MilkW


apply(data, 2, FUN = function(x) length(boxplot.stats(x)$out))
boxplot(data$Grocery)
data$GroceryW <-  Winsorize(data$Grocery, probs = c(0,0.945))
boxplot(data$GroceryW)
data$Grocery <- data$GroceryW

apply(data, 2, FUN = function(x) length(boxplot.stats(x)$out))
boxplot(data$Detergents_Paper)
data$DPW <-  Winsorize(data$Detergents_Paper, probs = c(0,0.91))
boxplot(data$DPW)
data$Detergents_Paper <- data$DPW

str(data)
data$DPW <- NULL
data$GroceryW <- NULL
data$FrozenW <- NULL
data$MilkW <- NULL
data$FreshW <- NULL

normalize_var <-  function(x){
  if(sum(x, na.rm = T)==0) x
  else ((x-min(x, na.rm = T))/(max(x, na.rm = T)-min(x, na.rm = T)))
}
```

```
str(data.norm)

install.packages("corrplot")
library(corrplot)

data_cor <- cor(data.norm)
corrplot.mixed(data_cor)

data.norm$Grocery <- NULL

eval.metrics <- data.frame()
for (k in 2:8){
  set.seed(1010)
  km <- kmeans(data.norm, k, iter.max = 20, nstart = 1000)
  eval.metrics <- rbind(eval.metrics, c(k, km$tot.withinss, km$betweenss/km$totss))
}
eval.metrics
colnames(eval.metrics) <- c("Clusters","totWithinss", "Ratio")

sample.3k <- kmeans(data.norm, 3, 20, 1000)
sum.stats <- summary.stats(data.norm, sample.3k$cluster, 3)
sample.3k
```

KNN PREDIKCIJE

Monday, June 5, 2023 6:19 PM

ulazne numericke varijable
izlazne varijable su faktori
opisne varijable -> faktor -> numeric
- data$var<-as.numeric(as.factor(data$var))

#Priprema podataka

percentil60 <- quantila(data$Congestion407, 0,6)
data$Take407All <- ifelse (data$Congestion407 < percentil60 & data$Comments = "", yes = "YES", no = "NO")
data$Take407All <- as.factor(data$Take407All)

Provera autlejera
apply(data, 2, FUN = function(x) length(boxplot.stats(x)$out))

Standardizacija
apply(data, 2, FUN = function(x) shapiro.test(x))
za sve redove bez normalne raspodele
ako imaju normalnu raspodelu (p>=0.05) ide TRUE umesto ove dve prom
data.std <- apply(data[,1:8], 2, FUN = function(x) scale(x, center = median(x), scale = IQR(x)))

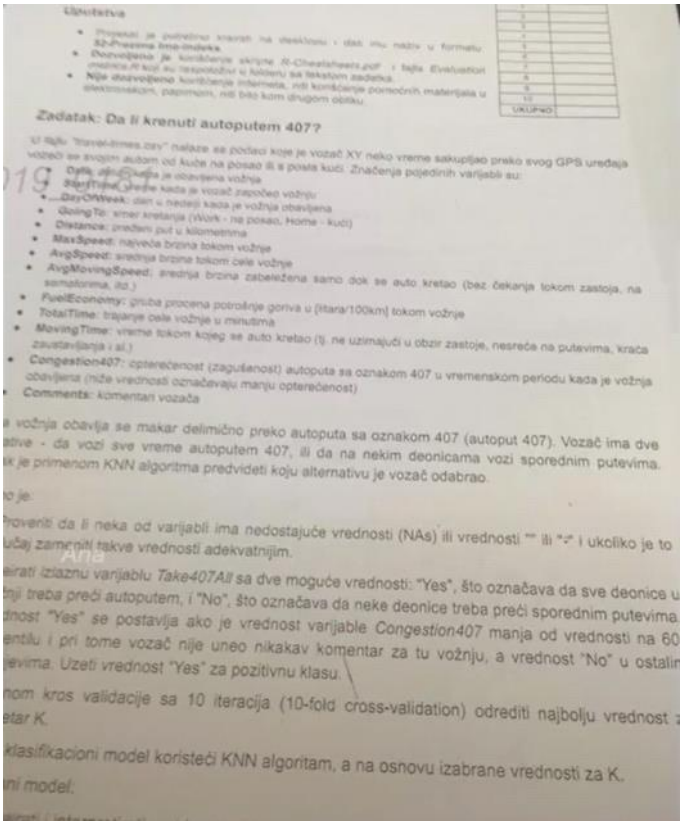
data.std <- as.data.frame(data.std)
data.std$izlazna <- as.factor(data$izlazna)

Podela na Train i Test
library(caret)
set.seed(1010)
indexes<-createDatePartition(data.std$izlazna, p= 0.8, list = F)
train.data<-data.std[indexes,]
test.data<-data.std[-indexes,]

Krosvalidacija
numFolds <- trainControl(method = "cv", number = 10)
kGrid = expand.grid(x = train.data[,-rbIzlazne], y = train.data$izlazna, method = "knn",
trControl = numFolds, tuneGrid = kGrid)
best_k <- knn.cv$bestTune$k
#plot(knn.cv)

Kreiranje modela / predikcije
library(class)
knn.pred<-knn(train.data[,-rbIzlazne], test = test.data[,rbIzlazne],cl = train.data$izlazna, k = best_k)

Kreriranje matrice konfuzije
knn.cm <- table(true = test.data$izlazna, predicted = knn.pred)
knn.eval <- getEvalMetrics(knn.cm)

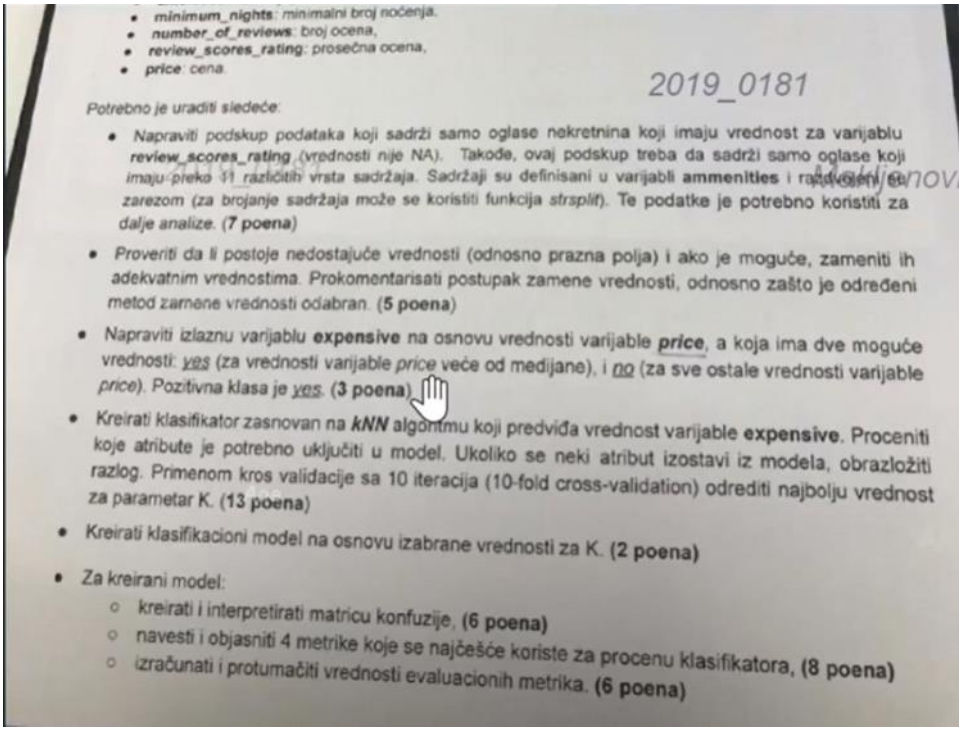


Zadatak: Da li krenuti autoputem 407?

FUNKCIJA MATRICE KONFUZIJE
getEvalMetrics<-function(cm){
TP<-cm[2,2]
TN<-cm[1,1]
FP<-cm[1,2]
FN<-cm[2,1]
acc<-sum(diag(cm))/sum(cm)
prec<- TP / (TP + FP)
rec<- TP / (TP + FN)
F! <- (2*prec*rec) / (prec+rec)
c(Accuracy = acc, Precision = prec, Recall = rec,
F1 = F1)

KNN airbnb

Monday, June 5, 2023 8:08 PM



- **minimum_nights**: minimalni broj noćenja,
- **number_of_reviews**: broj ocena,
- **review_scores_rating**: prosečna ocena,
- **price**: cena.

2019_0181

Potrebno je uraditi sledeće:

- Napraviti podskup podataka koji sadrži samo oglase nekretnina koji imaju vrednost za varijablu **review_scores_rating** (vrednosti nije NA). Takođe, ovaj podskup treba da sadrži samo oglase koji imaju preko 11 različitih vrsta sadržaja. Sadržaji su definisani u varijabli **ammenities** i razdvojeni su zarezom (za brojanje sadržaja može se koristiti funkcija *strsplit*). Te podatke je potrebno koristiti za dalje analize. (**7 poena**)
- Proveriti da li postoje nedostajuće vrednosti (odnosno prazna polja) i ako je moguće, zameniti ih adekvatnim vrednostima. Prokomentarisati postupak zamene vrednosti, odnosno zašto je određeni metod zamene vrednosti odabran. (**5 poena**)
- Napraviti izlaznu varijablu **expensive** na osnovu vrednosti varijable ***price***, a koja ima dve moguće vrednosti: *yes* (za vrednosti varijable *price* veće od medijane), i *no* (za sve ostale vrednosti varijable *price*). Pozitivna klasa je *yes*. (**3 poena**)
- Kreirati klasifikator zasnovan na ***kNN*** algoritmu koji predviđa vrednost varijable **expensive**. Proceniti koje atribute je potrebno uključiti u model. Ukoliko se neki atribut izostavi iz modela, obrazložiti razlog. Primenom kros validacije sa 10 iteracija (10-fold cross-validation) odrediti najbolju vrednost za parametar K. (**13 poena**)
- Kreirati klasifikacioni model na osnovu izabrane vrednosti za K. (**2 poena**)
- Za kreirani model:
  - kreirati i interpretirati matricu konfuzije, (**6 poena**)
  - navesti i objasniti 4 metrike koje se najčešće koriste za procenu klasifikatora, (**8 poena**)
  - izračunati i protumačiti vrednosti evaluacionih metrika. (**6 poena**)

data<- read.csv(…)

?strsplit

```
indeksi<-c()
for(i in 1:length(data$id){
if(length(unlist(strsplit(data$amenities[i], ", "))>11){
        indeksi<-append(indeksi, i)
  }
}
#ako se splituje po tacki mora "[.]"

podskup<-data[c(indeksi),]
data<-podskup[complete.cases(podskup[,5]),]

apply(… sum(is.na(x))…

NA vrednosti menjamo medijanom jer je p<0.05
shapiro.test(data$bathrooms)

Brise prvi karakter ($)
price<-substring(price, first = 2)
  I L I
price <- "$350$"
zameni sve $ slovom d
price<-gsub("\\$","d", price)
zameni sve karaktere koji nisu brojevi praznim stringom
price<-gsub("[^0-9.]","",price)

#Nebitne kolone u NULL, ostale u faktor pa num

data$expensive <- ifelse(…)
```

Autlejeri i standardizacija

```
apply(… boxplot.stats(x))
apply(… shapiro.test(x))

data.std <- (…)
#bedrooms pravi problem pri std pa necemo da ga standardizujemo
data.std<-(data[,c(1,2,3,5,6,7)…)
data.std$bedrooms<-data$bedrooms
data.std$expensive <-as.factor(data$expensive)
```

Podela na trening i test

```
indexes<-…
train.data<-data.std[indexes]
…
```

Krosvalidacija

```
library(e1071)
library(caret)
numfolds<-
kGrid =
set.seed…
knn.cv<-
best_k<-
```

Kreiranje modela

…

Matrica konfuzije

…

Evaluacija

…

LINEARNA REGRESIJA

Tuesday, June 6, 2023 11:15 AM

predvidjanje varijabli
sve varijable su numericke

data<-read.csv(…)

podskup<-subset(data, (data$Platform == "PS2" | data$Platform == "PS3" | data$Platform == "PS4"))

Priprema podataka
sum(is.na(data$User_Score))
apply(…)

data$Rating<-as.factor(data$Rating)
data[data$Rating==""]<-"T" jer se najcesce pojavljuje
levels(data$Rating) ovde vidimo da je ostao level viska pa moramo da ga obrisemo
data$Rating <- factor(data$Rating, levels=c("E", "E10+", "M", "T"))

shapiro.test(data$UserCount)
medijana<-median(data$UserCount, na.tm = T)
data$UserCount[is.na(data$UserCount))<-medijana

Matrica korelacije
data_cor <- cor(data)
library(corrplot)
corrplot(data_cor, method = "number", type = "upper", diag = F)

Podela na trening i test
library(caret)
set.seed(1010)
indexes<-createDataPartition(data$UserScore, p=0.8, list = F)
train.data<-data[indexes,]
test.data<-data[-indexes,]

Kreiranje linearnog modela
lm1<-lm(IzlaznaVarijabla ~ Ulazna1 + Ulazna2, data = train.data)
ako hocemo sve kolone za model onda
lm1<-lm(IzlaznaVarijabla ~ ., data = train.data)
summary(lm1)
reziduali predstavljaju odstupanja predvidjenih od stvarnih vrednosti
koeficijenti govore kako svaka varijabla utice na izlaznu varijablu
zadrzavamo sve varijable sa nekoliko zvezdica
F-statistic: nijedna varijabla nije bitna (p>mali broj)

- **Year_of_Release** - godina objavljivanja
- **Genre** - žanr
- **Publisher** - izdavač
- **NA_Sales** - prodaja u Severnoj Americi (u milionima jedinica)
- **EU_Sales** - prodaja u Evropskoj Uniji (u milionima jedinica)
- **JP_Sales** - prodaja u Japanu (u milionima jedinica)
- **Other_Sales** - prodaja u ostatku sveta: Afrika, Azija bez Japana, Australija, Evropa bez Evropske Unije i Južna Amerika (u milionima jedinica)
- **Global_Sales** - ukupna prodaja u svetu (u milionima jedinica)
- **Critic_Score** - agregatni broj poena od strane uredništva sajta Metacritic
- **Critic_Count** - broj ocenjivača uredništva sajta Metacritic na osnovu čijih ocena je dobijen Critic_Score
- **User_Score** - agregatni broj poena od strane posetilaca sajta Metacritic
- **User_Count** - broj ocenjivača posetilaca sajta Metacritic na osnovu čijih ocena je dobijen Critic_Score
- **Developer** - kompanija koja je napravila igricu
- **Rating** - ESRB oznaka kojoj ciljnoj grupi je namenjena igrica sa mogućim vrednostima: *Everyone 10+ (E10+), Teen (T), Everyone (E), Mature 17+ (M)*

Potrebno je učitati dataset *Video_Games_Sales_2017_reduced.csv* i uraditi sledeće:

1. Napraviti podskup data seta koji sadrži video igrice koje su pravljene za konzole (atribut *Platform*) "PS2", "PS3" ili "PS4". Taj data set je potrebno koristiti za dalje analize.
2. Proveriti da li postoje nedostajuće vrednosti i ako je moguće, zameniti ih adekvatnim vrednostima. Prokomentarisati postupak zamene vrednosti, odnosno zašto je određeni metod zamene vrednosti odabran.
3. Izvršiti predviđanje ocene posetilaca sajta (varijable *User_Score*) na osnovu ostalih atributa. Proceniti koje atribute je potrebno uključiti u model. Ukoliko se neki atribut izostavi iz modela, obrazložiti razlog.
4. Na osnovu rezultata modela:
   - protumačiti koeficijente svake varijable, odnosno interpretirati relaciju nezavisne i zavisne varijable na osnovu vrednosti koeficijenta
   - navesti koji atributi su značajni za predikciju i na osnovu čega je to zaključeno
   - objasniti šta je koeficijent determinacije (R-squared) i protumačiti njegovu vrednost
5. Napisati šta predstavlja svaki od četiri grafikona za dijagnostiku modela linearne regresije. Protumačiti svaki grafikon u kontekstu dobijenih rezultata.
6. Izvršiti predviđanja i izračunati koeficijent determinacije (R-squared) nad predikcijama, kao i standardnu devijaciju reziduala (RMSE). Protumačiti dobijene vrednosti.

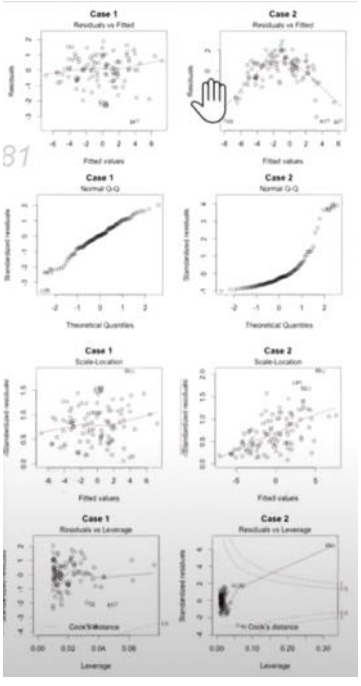
Izbacivanje korelisanih varijabli

library(car)
sort(sqrt(vif(lm1)))
trazimo varijablu s korelacijama > 2
data$NASales<-NULL
lm1<-lm(UserScore ~ ., data = train.data)

Kreiranje 4 plota
2 reda i
2 kolone
graphics.off()
par(mfrow = c(2,2))
plot(lm1)

Kreiranje predikcije
lm.pred <- predict(lm1, newdata = test.data)
ovo ispod je samo za grafik
test.data$UserScore_Pred <- lm1.pred
library(ggplot2)
ggplot(test.data) + gom_density(aes(x=UserScore, color = "actual")) +
geom_density(aes(x = UserScore_Pred, color = "predicted"))

dobri vs losi



zadovoljena pretpostavka o lineonarnosti?
linija treba da bude prava,
blizu nule a tacke oko nje

reziduali imaju normalnu raspodelu?
kruzici moraju da prate
isprekidanu liniju

reziduali imaju jednake varijanse?
linija mora biti iole horizontalna

da li postoje mnogo velike/male vrednosti?
sve vrednosti moraju biti ispod ili iznad
isprekidanih linija (Cook-ovih distanci)

Kreiranje metrike

```
RSS <- sum((lm5.pred - test.data$izlazna)^2)
TSS <- sum((mean(train.data$izlazna) - test.data$izlazna)^2)
rsquared <- 1 - RSS / TSS
RMSE <- sqrt(RSS/nrow(test.data))

RMSE/mean(test.data$izlazna)
```

RSS - pokazuje u kojoj meri uspevamo da predvidimo varijalibitet izlazne
TSS - koliki je varilijabitet izlazne
rsquared - kolko prosto varijabiliteta objasnjava nas model (sto veci to bolje)
RMSE - prosecna greska koju pravimo (ista jedinica kao izlazna varijabla)
RMSE/mean - koliko % smo pogresili u modelu (najbitnija!) 0.16 = 16%

LR kredit

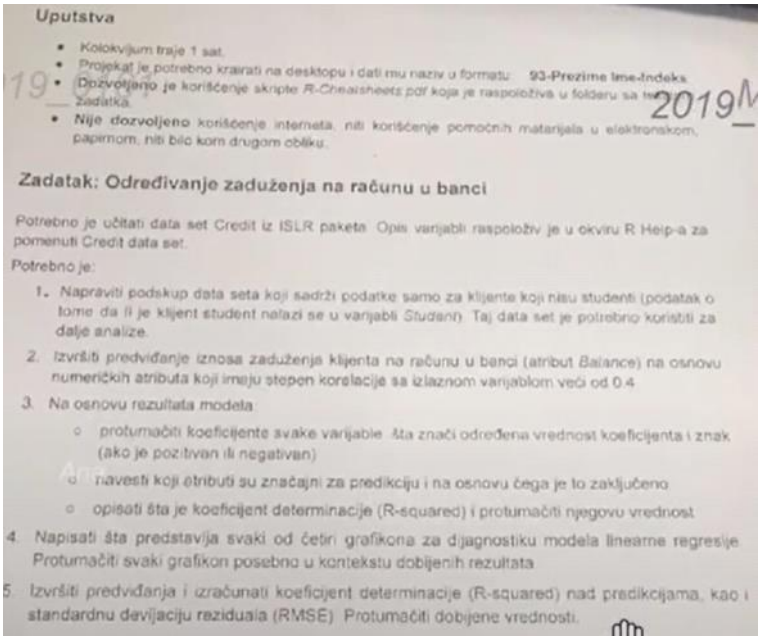
Tuesday, June 6, 2023 5:23 PM

```
data <- read.csv(…)
library("ISLR")
?Credit

Podskup
podskup <- subset(data, data$student = "No")
data<-podskup
data$Student<-NULL

Priprema podataka
data$ID<-NULL
data$MarriedM-as.numeric(data$Married)
…
Podela na train i test
library(caret)
set.seed(1010)
indexes<-…
train.data<-…
test.data<-…
Korelaciona matrica
data_cor <- cor(data)
library(corrplot)
corrplot(data_cor, method = "number", type = "upper", diag = F)
odavde vidimo da nam trebaju samo kolone Income Limit i Rating
Kreiranje modela
lm1 <-lm(Balance ~ Income+Limit+Rating, data = train.data)
summary(lm1)
reziduali sto manji, p mora biti mala vrednost, Rsquared sto veca
Izbacivanje visoko korelisanih prom
library(car)
sort(sqrt(vif(lm1)))
Provera pretpostavke linearne regresije
graphics.off()
par(mfrow = c(2,2))
plot(lm2)
Kreiranje predikcije
lm2.pred<- predict(…)

test.data$Balance_Pred <- …
ggplot(test.data) <- …
Kreiranje metrike
RSS<-…
TSS<-…
rsquared<-…
RMSE<-…
RMSE/mean(…)
```



Uputstva

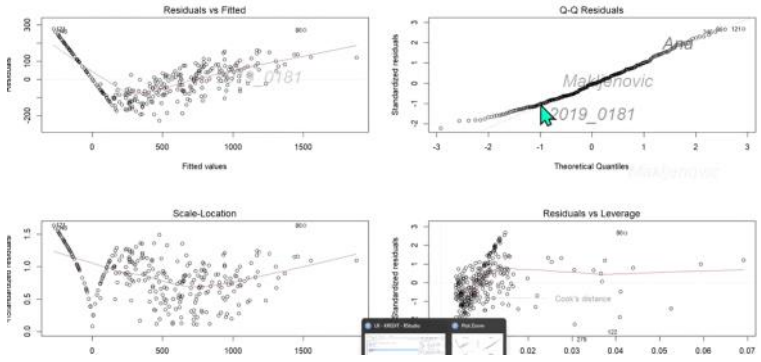
- Kolokvijum traje 1 sat.
- Projekat je potrebno kreirati na desktopu i dati mu naziv u formatu: **93-Prezime Ime-Indeks**
- **Dozvoljeno** je korišćenje skripte *R-Cheatsheets.pdf* koja je raspoloživa u folderu sa testom Zadatka.
- ***Nije dozvoljeno*** korišćenje interneta, niti korišćenje pomoćnih materijala u elektronskom, papirnom, niti bilo kom drugom obliku.

Zadatak: Određivanje zaduženja na računu u banci

Potrebno je učitati data set Credit iz ISLR paketa. Opis varijabli raspoloživ je u okviru R Help-a za pomenuti Credit data set.

Potrebno je:

1. Napraviti podskup data seta koji sadrži podatke samo za klijente koji nisu studenti (podatak o tome da li je klijent student nalazi se u varijabli *Student*). Taj data set je potrebno koristiti za dalje analize.
2. Izvršiti predviđanje iznosa zaduženja klijenta na računu u banci (atribut *Balance*) na osnovu numeričkih atributa koji imaju stepen korelacije sa izlaznom varijablom veći od 0.4
3. Na osnovu rezultata modela:
   - protumačiti koeficijente svake varijable. Šta znači određena vrednost koeficijenta i znak (ako je pozitivan ili negativan)
   - navesti koji atributi su značajni za predikciju i na osnovu čega je to zaključeno
   - opisati šta je koeficijent determinacije (R-squared) i protumačiti njegovu vrednost
4. Napisati šta predstavlja svaki od četiri grafikona za dijagnostiku modela linearne regresije. Protumačiti svaki grafikon posebno u kontekstu dobijenih rezultata
5. Izvršiti predviđanja i izračunati koeficijent determinacije (R-squared) nad predikcijama, kao i standardnu devijaciju reziduala (RMSE). Protumačiti dobijene vrednosti.



1. Linija nije prava
2. Tackice su ok
3. Linije je otprilike horizontalna, onako
4. Ima ekstremnih vrednosti

   model nije savrsen ali je prihvatljiv

LR medalje

Tuesday, June 6, 2023 6:18 PM

data<-read.csv(…)

podskup <- subset(data, (data$Team.NOC != "-"))
data<-…

izbacivanje NA, -, itd…
silver ide u numeric
popunjavanje NA vrednosti medijanom ili meanom
posto je gold izlazna varijabla za nju se ne radi popunjavanje na vrednosti

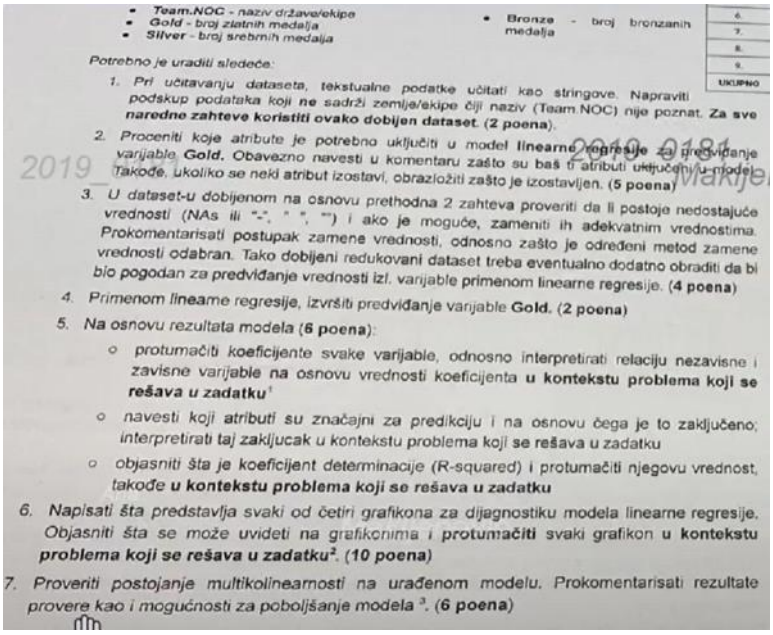
podela na test i trening

Kreiranje linearnog modela
TUMACENJE LINEARNOG MODELA
(intercept) da su silver i bronza 0, izlaz bi bio -0.097
Svaki put kad se silver poveca za 1, izlaz raste za 0.65
Svaki put kad bronza poraste za 1, izlaz raste za 0.25
Linearna kriva: y = 0.25*B + 0.65*S - 0.097
Na osnovu zvezdica vidi se da su bitne
Rsquared govori da smo objasnili 92% varijabiliteta izlazne

provera medjuzavisnosti varijabli
sort(sqrt(vif(lm1)))
izbacujemo bronzu jer ostaje veci rsquared
Crtanje modela

Kreiranje predikcije nije moralo!
lm1.pred <- predict(…)

Kreiranje metrike
greska je 125% tkd smece je

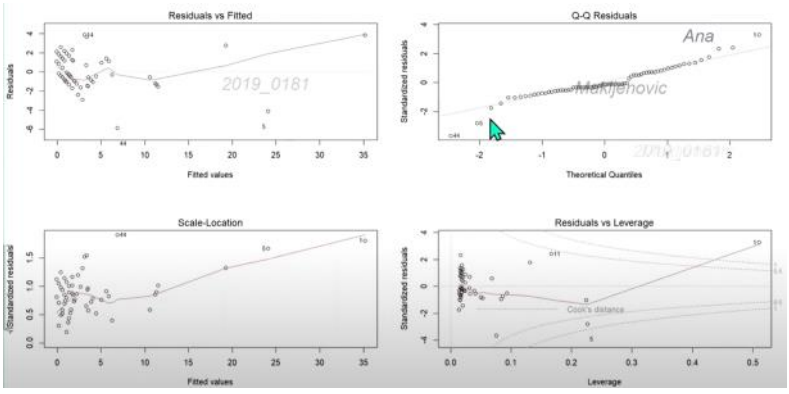


- **Team.NOC** - naziv države/ekipe
- **Gold** - broj zlatnih medalja
- **Silver** - broj srebrnih medalja
- **Bronze** - broj bronzanih medalja

Potrebno je uraditi sledeće:

1. Pri učitavanju dataseta, tekstualne podatke učitati kao stringove. Napraviti podskup podataka koji **ne** sadrži zemlje/ekipe čiji naziv (Team.NOC) nije poznat. **Za sve naredne zahteve koristiti ovako dobijen dataset. (2 poena)**.
2. Proceniti koje atribute je potrebno uključiti u model **linearne regresije** za predviđanje varijable **Gold.** Obavezno navesti u komentaru zašto su baš ti atributi uključeni u model. Takođe, ukoliko se neki atribut izostavi, obrazložiti zašto je izostavljen. **(5 poena)**
3. U dataset-u dobijenom na osnovu prethodna 2 zahteva proveriti da li postoje nedostajuće vrednosti (NAs ili "-", " ", "") i ako je moguće, zameniti ih adekvatnim vrednostima. Prokomentarisati postupak zamene vrednosti, odnosno zašto je određeni metod zamene vrednosti odabran. Tako dobijeni redukovani dataset treba eventualno dodatno obraditi da bi bio pogodan za predviđanje vrednosti izl. varijable primenom linearne regresije. **(4 poena)**
4. Primenom **lineame** regresije, izvršiti predviđanje varijable **Gold. (2 poena)**
5. Na osnovu rezultata modela **(6 poena)**:
   - protumačiti koeficijente svake varijable, odnosno interpretirati relaciju nezavisne i zavisne varijable na osnovu vrednosti koeficijenta **u kontekstu problema koji se rešava u zadatku**[1]
   - navesti koji atributi su značajni za predikciju i na osnovu čega je to zaključeno; interpretirati taj zaključak u kontekstu problema koji se rešava u zadatku
   - objasniti šta je koeficijent determinacije (R-squared) i protumačiti njegovu vrednost, takođe **u kontekstu problema koji se rešava u zadatku**
6. Napisati šta predstavlja svaki od četiri grafikona za dijagnostiku modela linearne regresije. Objasniti šta se može uvideti na grafikonima i **protumačiti** svaki grafikon **u kontekstu problema koji se rešava u zadatku**[2]. **(10 poena)**
7. Proveriti postojanje multikolinearnosti na urađenom modelu. Prokomentarisati rezultate provere kao i mogućnosti za poboljšanje modela [3]. **(6 poena)**

```
Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -0.09714    0.23876  -0.407 0.685350
Silver       0.65681    0.06966   9.428 4.36e-14 ***
Bronze       0.25263    0.07276   3.472 0.000889 ***
---
```



1. nije zadovoljena predpostavka
2. u glavnom prate liniju
3. reziduali nemaju jednake varijanse
4. postoje 2 opservacije van kukove distance
   model nije idealan ali je zadovoljavajuci