

UNIVERZITET U SARAJEVU
ELEKTROTEHNIČKI FAKULTET SARAJEVO

DOMAĆA ZADAĆA 2

MAŠINSKO UČENJE

Odsjek: Računarstvo i Informatika

Datum: 26.12.2019

Studenti:

- Mašović Haris, 1689/17993**
- Muminović Amir, 1661/17744**

Opis seta podataka

U zadaći 2 koristi se isti set podataka kao za zadaću 1, attrition_train.csv. Cilj zadaće 2 je, također, izgraditi klasifikacijski model koji će utvrditi da li će uposlenik neke kompanije napustiti tu kompaniju (Attrition=yes).

Kao podsjetnik, u setu se nalaze podaci o uposlenicima (demografski, o vrsti posla, uspješnosti na poslu, zadovoljstvu na poslu, edukaciji, itd). Dodatne informacije o kategoričkim varijablama:

- Education: 1 'Below College', 2 'College', 3 'Bachelor', 4 'Master', 5 'Doctor'
- EnvironmentSatisfaction: 1 'Low', 2 'Medium', 3 'High', 4 'Very High'
- JobInvolvement: 1 'Low', 2 'Medium', 3 'High', 4 'Very High'
- JobSatisfaction: 1 'Low', 2 'Medium', 3 'High', 4 'Very High'
- PerformanceRating: 1 'Low', 2 'Good', 3 'Excellent', 4 'Outstanding'
- RelationshipSatisfaction: 1 'Low', 2 'Medium', 3 'High', 4 'Very High'
- WorkLifeBalance: 1 'Bad', 2 'Good', 3 'Better', 4 'Best'

Zadatak 1.a

Izgradite najmanje tri predikcijska modela koji moraju biti neki od algoritama koje smo obrađivali u sklopu nastave: k-nn, Bayes (tj. LDA, QDA), logistička regresija, ili SVM, ali isključujući drvo odlučivanja.

Dokumentujte proces izgradnje modela, njihovog treniranja i testiranja. Evaluirajte vaše modele pomoću konfuzijske matrice (tačnost, specifičnost, osjetljivost, kappa statistika, itd.) i ROC krive. Na osnovu ostvarenih rezultata odaberite najbolji predikcijski model i obrazložite vaš odabir.

Uvod

Za sve modele obavljeno je isto predprocesiranje kao u prošlog zadaci zbog činjenice da su ostaverni rezultati imali dobru tačnost. Trening podaci su podijeljeni u 70% trening i 30% test odnosu.

Podsjetimo se da smo u inicijalnoj analizi skupa podataka otkrili da su sljedeće varijable povezane sa ciljanim atributom (Attrition):

- | | |
|--|---|
| <input type="checkbox"/> NumCompaniesWorked | <input type="checkbox"/> HourlyRate |
| <input type="checkbox"/> DistanceFromHome | <input type="checkbox"/> JobInvolmen |
| <input type="checkbox"/> Department | <input type="checkbox"/> EnvironmentSatisfction |
| <input type="checkbox"/> BusinessTravel | <input type="checkbox"/> EducationField |
| <input type="checkbox"/> WorkLifeBalance | <input type="checkbox"/> OverTime |
| <input type="checkbox"/> TrainingTimesLastYear | <input type="checkbox"/> PercentSalaryHike |
| <input type="checkbox"/> TotalWorkingYears | <input type="checkbox"/> StockOptionLevel |
| <input type="checkbox"/> JobSatisfaction | <input type="checkbox"/> YearsWithCurrManager |
| <input type="checkbox"/> MaritalStatus | <input type="checkbox"/> YearsInCurrentRole |
| <input type="checkbox"/> JobLevel | <input type="checkbox"/> YearsAtCompany |

Bayes

LDA

Treniranje

Treniranje inicijalnog modela obavljeno je koristeći lda funkciju:

```
ldaModel <- lda(Attrition ~ NumCompaniesWorked + DistanceFromHome +  
Department + BusinessTravel + WorkLifeBalance + TrainingTimesLastYear +  
TotalWorkingYears + JobSatisfaction + MaritalStatus + JobLevel + HourlyRate  
+ JobInvolvement + EnvironmentSatisfaction + EducationField + OverTime +  
PercentSalaryHike + StockOptionLevel + YearsWithCurrManager +  
YearsInCurrentRole + YearsAtCompany, train)
```

Testiranje

Sljedeći korak je predviđanje testnih podataka.

```
ldaPredictions <- predict(ldaModel, test)
```

Performanse modela možemo naći sa funkcijom confusionMatrix:

```
ldaResults <- confusionMatrix(ldaPredictions$class, test$Attrition)
```

Nakon izvršavanja funkcije dobijemo sljedeće rezultate:

```
Confusion Matrix and Statistics

              Reference
Prediction    No  Yes
      No    290   30
      Yes    11   22

              Accuracy : 0.8839
              95% CI : (0.8457, 0.9153)
      No Information Rate : 0.8527
      P-Value [Acc > NIR] : 0.054073
```

```
Kappa : 0.4553

McNemar's Test P-Value : 0.004937

Sensitivity : 0.9635
Specificity : 0.4231
Pos Pred Value : 0.9062
Neg Pred Value : 0.6667
Prevalence : 0.8527
Detection Rate : 0.8215
Detection Prevalence : 0.9065
Balanced Accuracy : 0.6933

'Positive' Class : No
```

Prikažimo informacije koje se traže u zadatku dva u tabeli radi bolje preglednosti:

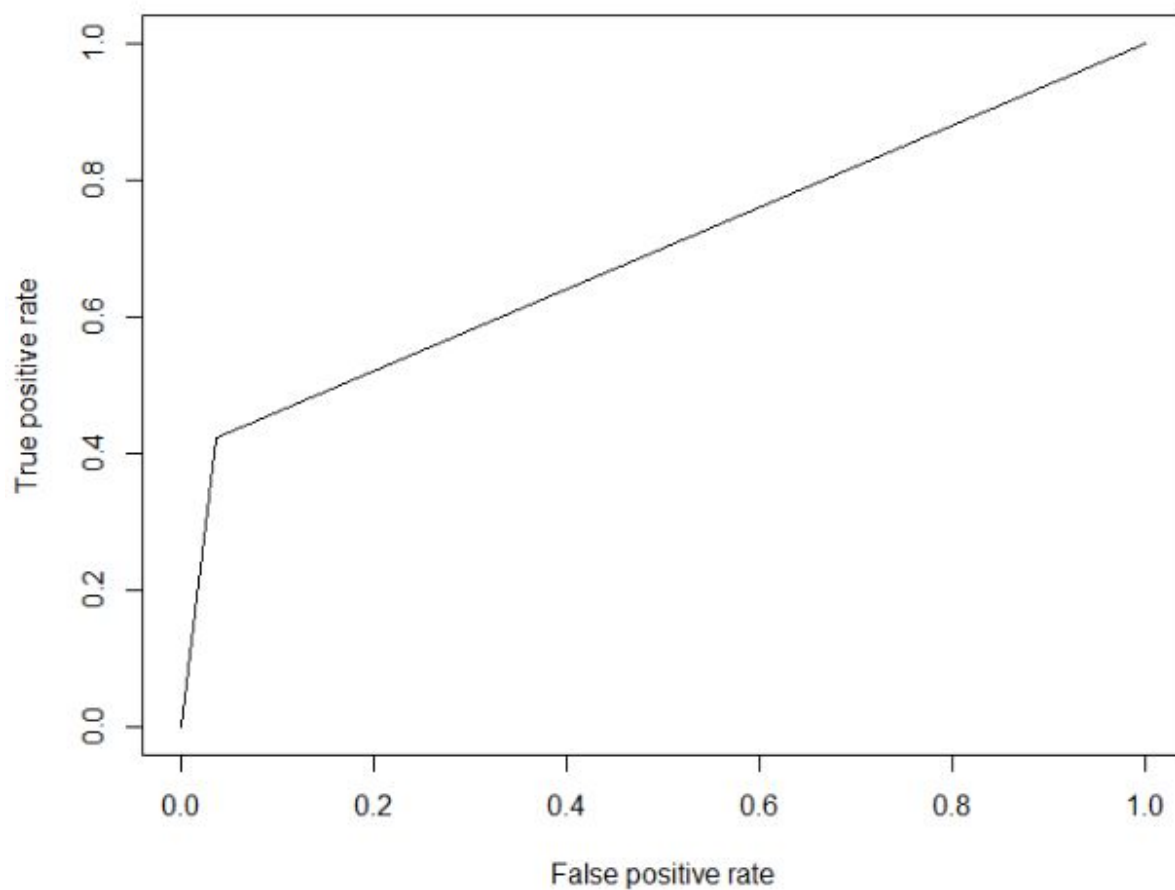
Metrika	Ostvareni rezultat
Pos. Pred. Value	0.9062
Balanced Accuracy	0.6933
Kappa statistics	0.4553
Lower bound of 95% CI	0.8675

ROC kriva

ROC krivu možemo skicirati koristeći sljedeće komande:

```
perf<-prediction(predictions=c(as.factor(ldaPredictions$class)),labels=test
$Attrition)
graph = performance(perf, measure = "tpr", x.measure = "fpr")
plot(graph)
```

Nakon izvršavanja koda, dobijen je sljedeći grafik sa ROC krivom:



QDA

Treniranje

Treniranje inicijalnog modela obavljeno je koristeći qda funkciju:

```
ldaModel <- qda(Attrition ~ NumCompaniesWorked + DistanceFromHome +  
Department + BusinessTravel + WorkLifeBalance + TrainingTimesLastYear +  
TotalWorkingYears + JobSatisfaction + MaritalStatus + JobLevel + HourlyRate  
+ JobInvolvement + EnvironmentSatisfaction + EducationField + OverTime +  
PercentSalaryHike + StockOptionLevel + YearsWithCurrManager +  
YearsInCurrentRole + YearsAtCompany, train)
```

Testiranje

Sljedeći korak je predviđanje testnih podataka.

```
qdaPredictions <- predict(qdaModel, test)
```

Performanse modela možemo naći sa funkcijom confusionMatrix:

```
qdaResults <- confusionMatrix(qdaPredictions$class, test$Attrition)
```

Nakon izvršavanja funkcije dobijemo sljedeće rezultate:

```
Confusion Matrix and Statistics

          Reference
Prediction  No  Yes
      No  273  30
      Yes   28  22

              Accuracy : 0.8357
              95% CI : (0.7928, 0.8728)
      No Information Rate : 0.8527
      P-Value [Acc > NIR] : 0.8358
```

```
Kappa : 0.3354

McNemar's Test P-Value : 0.8955

Sensitivity : 0.9070
Specificity : 0.4231
Pos Pred Value : 0.9010
Neg Pred Value : 0.4400
Prevalence : 0.8527
Detection Rate : 0.7734
Detection Prevalence : 0.8584
Balanced Accuracy : 0.6650

'Positive' Class : No
```

Prikažimo informacije koje se traže u zadatku dva u tabeli radi bolje preglednosti:

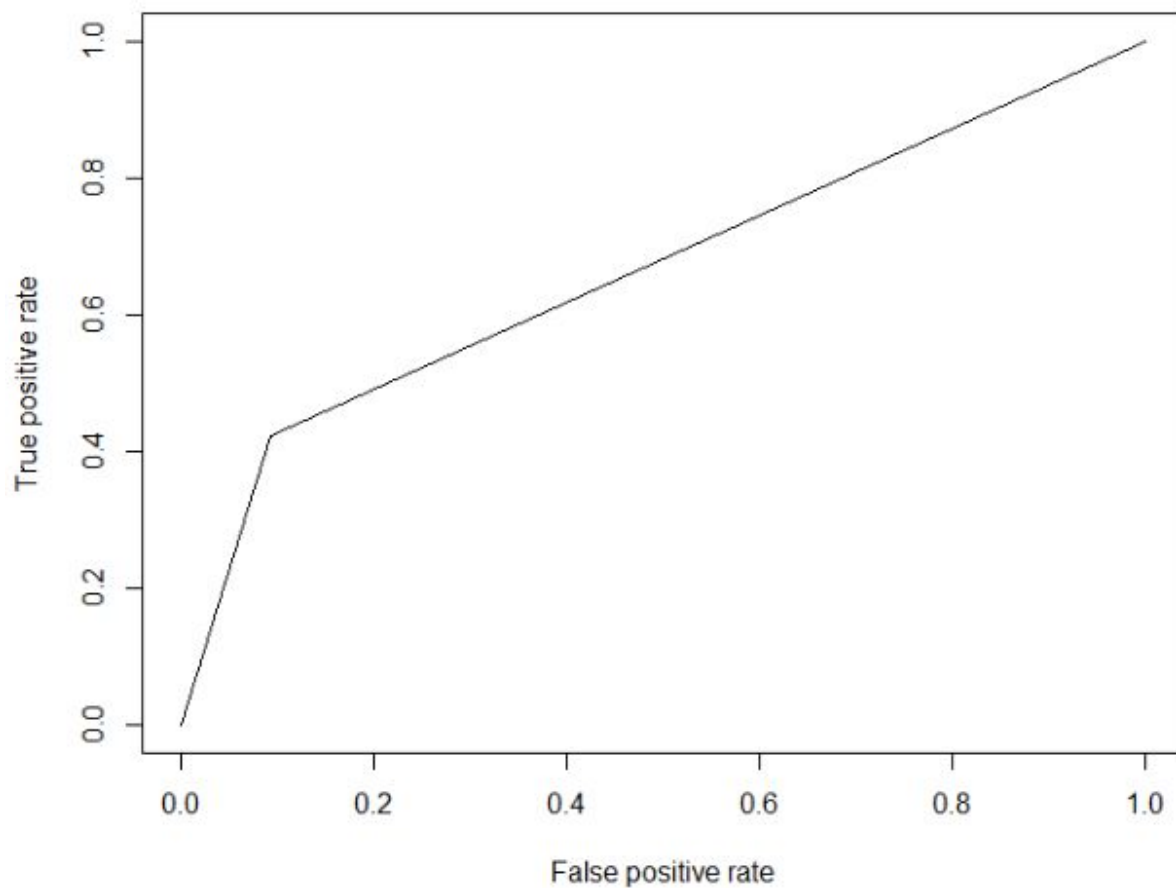
Metrika	Ostvareni rezultat
Pos. Pred. Value	0.9010
Balanced Accuracy	0.6650
Kappa statistics	0.3354
Lower bound of 95% CI	0.7928

ROC kriva

ROC krivu možemo skicirati koristeći sljedeće komande:

```
perf<-prediction(predictions=c(as.factor(qdaPredictions$class)),labels=test
$Attrition)
graph = performance(perf, measure = "tpr", x.measure = "fpr")
plot(graph)
```

Nakon izvršavanja koda, dobijen je sljedeći grafik sa ROC krivom:



Logistička regresija

Treniranje

Treniranje inicijalnog modela obavljeno je koristeći glm funkciju sa vrijednosti binominal za familz parametar:

```
logitMod <- glm(Attrition~., family = 'binomial',
```

Testiranje

Sljedeći korak je predviđanje testnih podataka.

```
ypred<-predict(logitMod,test, type='response')
```

Predviđanja će biti predstavljena kao vrijednosti od 0 do 1. Vrijednosti manje ili jednake od 0.5 će imati labelu "Ne" a vrijednosti veće od 0.5 će imati vrijednost "Da".

```
ypred=ifelse(ypred > 0.5, "Yes", "No")
```

Performanse modela možemo naći sa funkcijom confusionMatrix:

```
confusionMatrix(as.factor(ypred), test$Attrition)
```

Nakon izvršavanja funkcije dobijemo sljedeće rezultate:

Confusion Matrix and Statistics

	Reference	
Prediction	No	Yes
No	236	29
Yes	12	17

Accuracy : 0.8605

95% CI : (0.8156, 0.898)

No Information Rate : 0.8435

```
P-Value [Acc > NIR] : 0.23776

Kappa : 0.3781

McNemar's Test P-Value : 0.01246

Sensitivity : 0.9516
Specificity : 0.3696
Pos Pred Value : 0.8906
Neg Pred Value : 0.5862
Prevalence : 0.8435
Detection Rate : 0.8027
Detection Prevalence : 0.9014
Balanced Accuracy : 0.6606

'Positive' Class : No
```

Prikažimo informacije koje se traže u zadatku dva u tabeli radi bolje preglednosti:

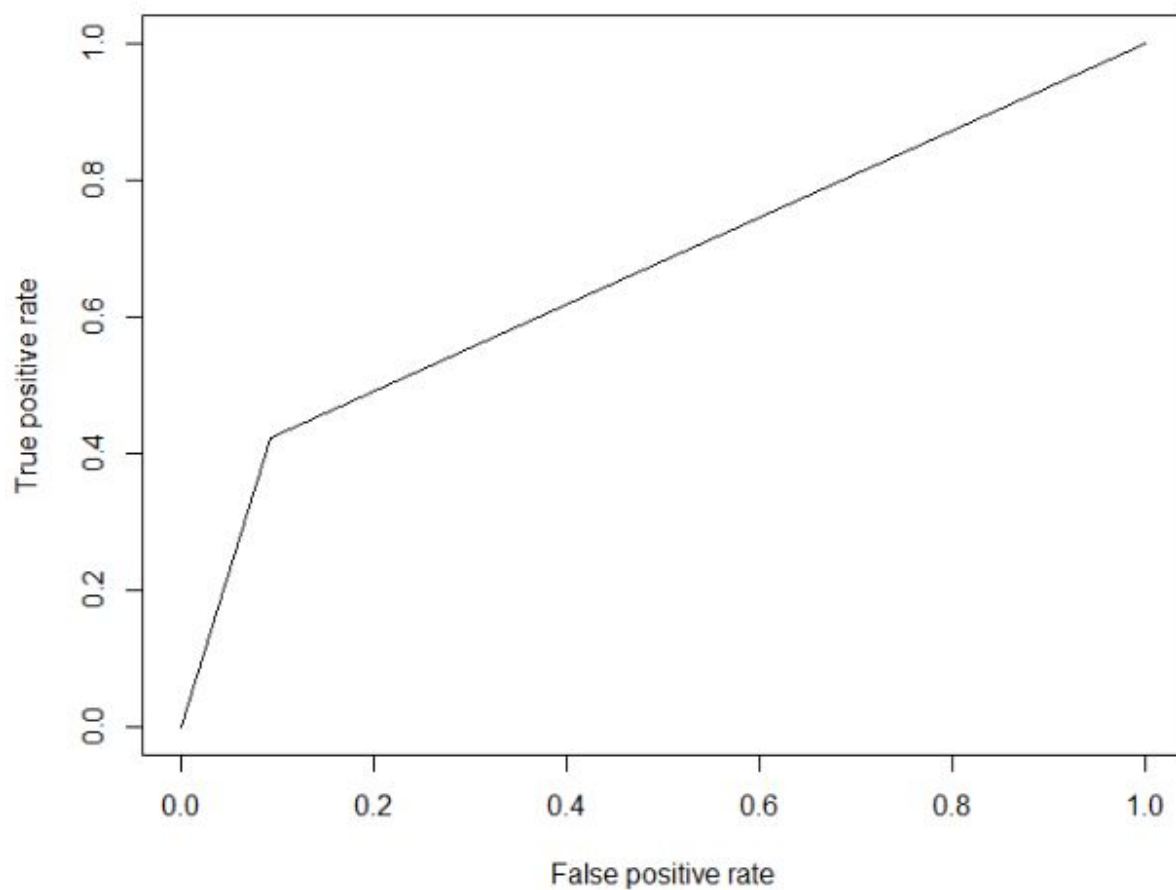
Metrika	Ostvareni rezultat
Pos. Pred. Value	0.8906
Balanced Accuracy	0.6606
Kappa statistics	0.3781
Lower bound of 95% CI	0.8156

ROC kriva

ROC krivu možemo skicirati koristeći sljedeće komande:

```
perf<-prediction(predictions=c(as.factor(ypred)),labels=test$Attrition)  
graph = performance(perf, measure = "tpr", x.measure = "fpr")  
plot(graph)
```

Nakon izvršavanja koda, dobijen je sljedeći grafik sa ROC krivom:



SVM

Treniranje

Na početku kreiran je model sa istim predprocesiranjem kao u prošloj zadaći. SVM model je kreiran koristeći svm funkciju iz paketa e1071. Početni cost je pretpostavljen na vrijednost 10. Opcija skaliranja je uključena. Kernel funkcija je postavljena na linear.

```
svmfit <- svm(Attrition~., data=train,  
kernel='linear', cost=10, scale=TRUE)
```

Nakon kreiranja modela, obavljene su predikcije koristeći testne podatke i funkciju predict:

```
ypred<-predict(svmfit,test)
```

Testiranje

Performanse modela možemo dobiti koristeći funkcije confusionMatrix, stvarnih i predviđenih rezultata.

```
confusionMatrix(ypred, test$Attrition)
```

Sa prvim modelom dobijen je sljedeći rezultat:

```
Confusion Matrix and Statistics

          Reference
Prediction No  Yes
   No    286   30
   Yes    15   22

      Accuracy : 0.8725
      95% CI : (0.8332, 0.9055)
No Information Rate : 0.8527
P-Value [Acc > NIR] : 0.16460

      Kappa : 0.4238
```

McNemar's Test P-Value : 0.03689

Sensitivity : 0.9502

Specificity : 0.4231

Pos Pred Value : 0.9051

Neg Pred Value : 0.5946

Prevalence : 0.8527

Detection Rate : 0.8102

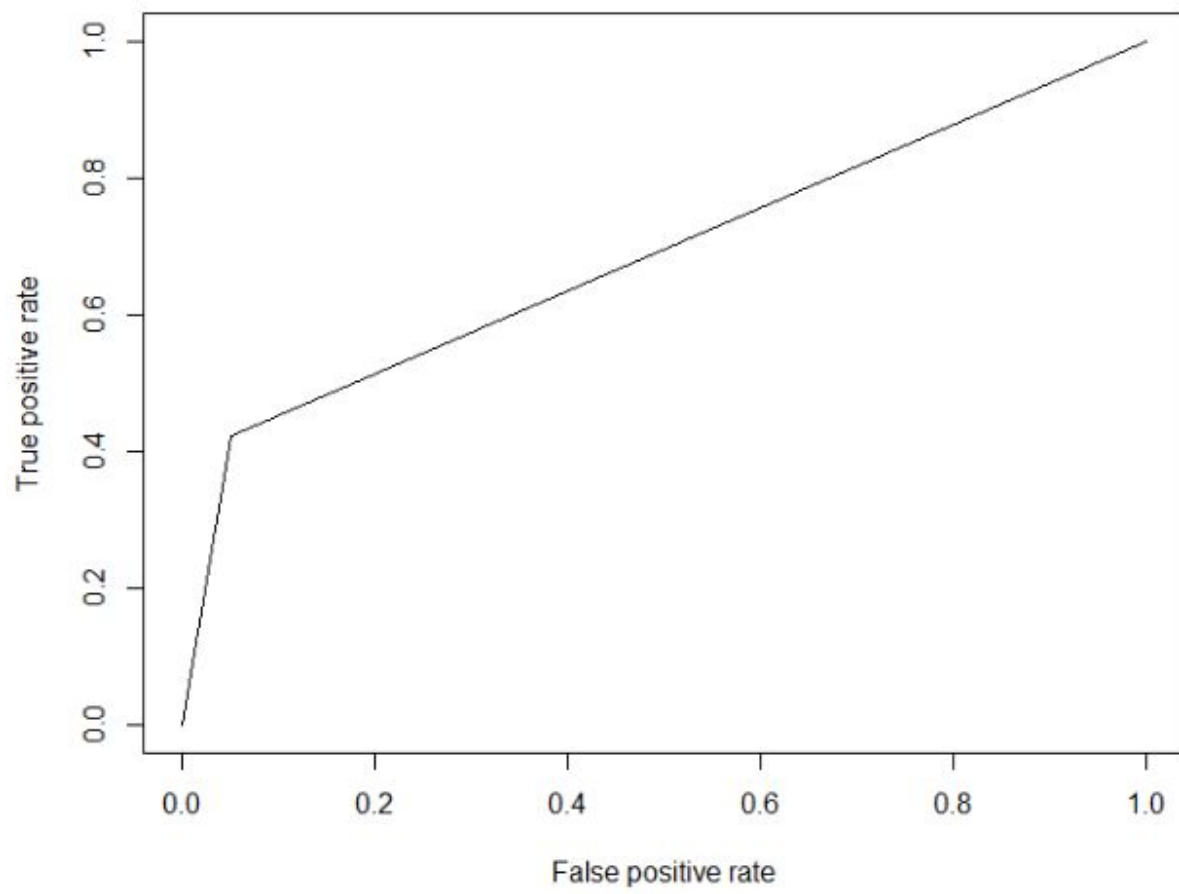
Detection Prevalence : 0.8952

Balanced Accuracy : 0.6866

'Positive' Class : No

Metrika	Ostvareni rezultat
Pos. Pred. Value	0.9051
Balanced Accuracy	0.6866
Kappa statistics	0.4238
Lower bound of 95% CI	0.8332

ROC Kriva



Poboljšanje

Tuning vrijednosti cost parametra

Jedan od načina kako bi se mogao poboljšati model je ako se uradi kros-validacija nad parametrom cost sa tune funkcijom iz paketa e1071.

```
set.seed(123)
tune.out<-tune(svm,Attrition~.,data=train_im, kernel ="linear",
               ranges=list(cost=c(0.001, 0.01, 0.1, 1, 5, 10, 100)))
```

Nakon izvršavanja tune-ing-a, dobijen je sljedeći rezultat:

Parameter tuning of 'svm':

```
- sampling method: 10-fold cross validation

- best parameters:
  cost
    1

- best performance: 0.1241634
```

Kada se postavi da bude cost = 1 u treniranju modela, dobiju se sljedeći rezultati:

```
Confusion Matrix and Statistics

      Reference
Prediction No Yes
No      288  30
Yes     13  22

      Accuracy : 0.8782
      95% CI   : (0.8394, 0.9104)
No Information Rate : 0.8527
P-Value [Acc > NIR] : 0.09868

      Kappa : 0.4393

McNemar's Test P-Value : 0.01469

      Sensitivity : 0.9568
```



```
Specificity : 0.4231
Pos Pred Value : 0.9057
Neg Pred Value : 0.6286
Prevalence : 0.8527
Detection Rate : 0.8159
Detection Prevalence : 0.9008
Balanced Accuracy : 0.6899

'Positive' Class : No
```

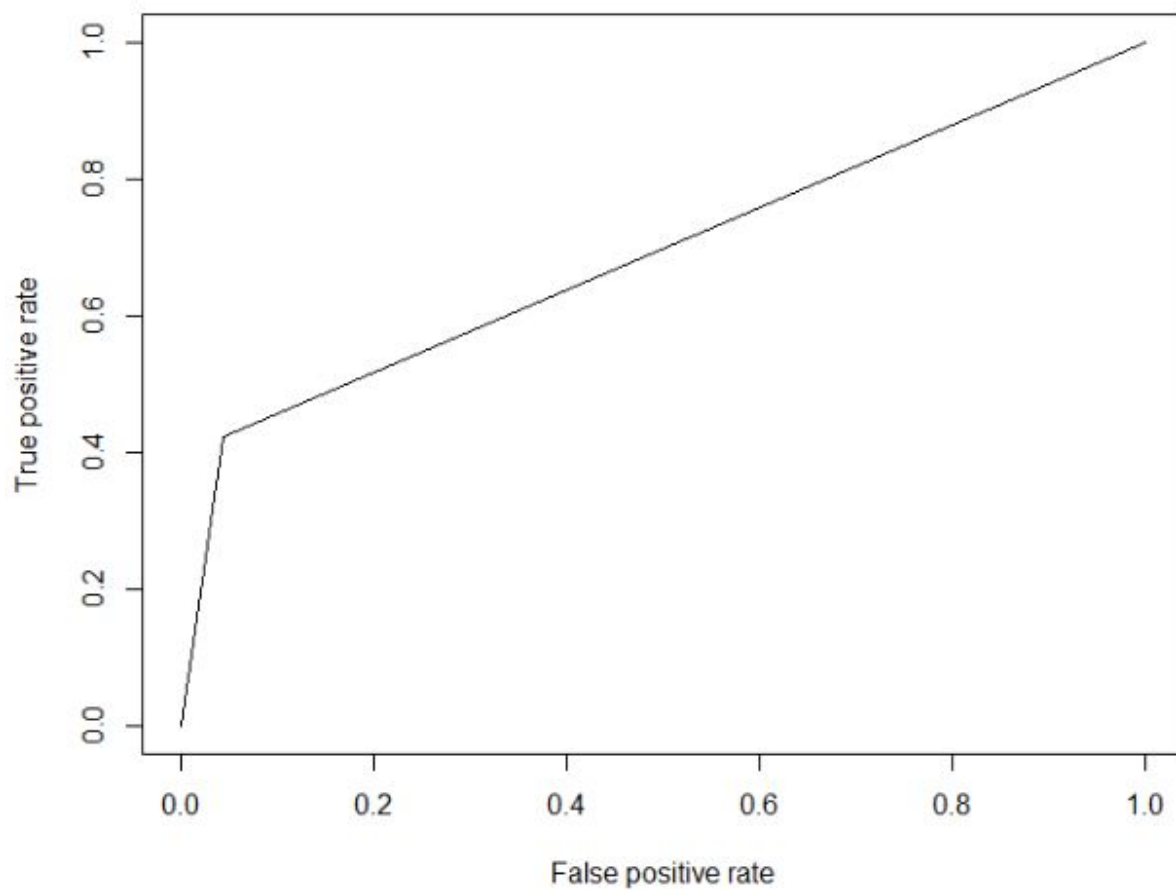
Ovim smo postigli bolje performanse modela.

Metrika	Model 1	Model 2
Pos. Pred. Value	0.9051	0.9057
Balanced Accuracy	0.6866	0.6899
Kappa statistics	0.4238	0.4393
Lower bound of 95% CI	0.8332	0.8394

Sljedeća ideja za poboljšanje modela je promjena tipa kernel funkcije. Promjene na radial i sigmoid su dovele do znatno lošijih rezultata. Rezultati polinomijalne kernel funkcije stepena 3 su slični linearnom.

ROC kriva - cost = 1

Sa rezultatima dobivenim za cost = 1, možemo skicirati sljedeću krivu.



Izbor najboljeg modela

Za izbor najboljeg modela je uzet LDA model, koji daje ujedno i najbolje performanse za sve navedene parametre koje se traže.

Zadatak 1.b

Prethodno odabrani najbolji model evaluirajte na setu podataka `attrition_test.csv`. Dokumentujte rezultate evaluacije. Grupa koja ostvari najbolji rezultat na setu "`attrition_test.csv`" će dobiti maksimalni broj bodova za ovaj zadatak, ostale grupe će biti rangirane u odnosu na najbolju grupu. Pri tome, da biste ostvarili bodove za zadatak 1-b, vaš model treba da ima minimalne performanse:

- Pos. Pred. Value $> \sim 0.7$
- Balanced Accuracy $> \sim 0.7$
- Kappa statistics $> \sim 0.3$
- Lower bound of (donja granica) 95% CI $> \sim 0.80$

Ukoliko model ne zadovoljava zahtijevane performanse, možete ponoviti proces iz zadatka 1-a. Dokumentujte promjene koje su dovele do poboljšanja performansi modela.

Bayes - LDA

Testiranje

Testiranje će se obaviti nad skupom testnih podataka. U kodu taj skup podataka se nalazi u varijabli `test_im`.

```
ldaPredictions <- predict(ldaModel,  
test_im)
```

Performanse modela možemo naći sa funkcijom `confusionMatrix`:

```
ldaResults <- confusionMatrix(ldaPredictions$class,  
test_im$Attrition)
```

Nakon izvršavanja funkcije dobijemo sljedeće rezultate:

```
Confusion Matrix and Statistics

          Reference
Prediction  No  Yes
      No  238  27
      Yes   10  19

      Accuracy : 0.8741
      95% CI : (0.8307, 0.9098)
      No Information Rate : 0.8435
      P-Value [Acc > NIR] : 0.083394

      Kappa : 0.4388

      Mcnemar's Test P-Value : 0.008529

      Sensitivity : 0.9597
      Specificity : 0.4130
      Pos Pred Value : 0.8981
      Neg Pred Value : 0.6552
      Prevalence : 0.8435
      Detection Rate : 0.8095
      Detection Prevalence : 0.9014
```

Balanced Accuracy : 0.6864

'Positive' Class : No

Prikažimo informacije koje se traže u zadatku dva u tabeli radi bolje preglednosti:

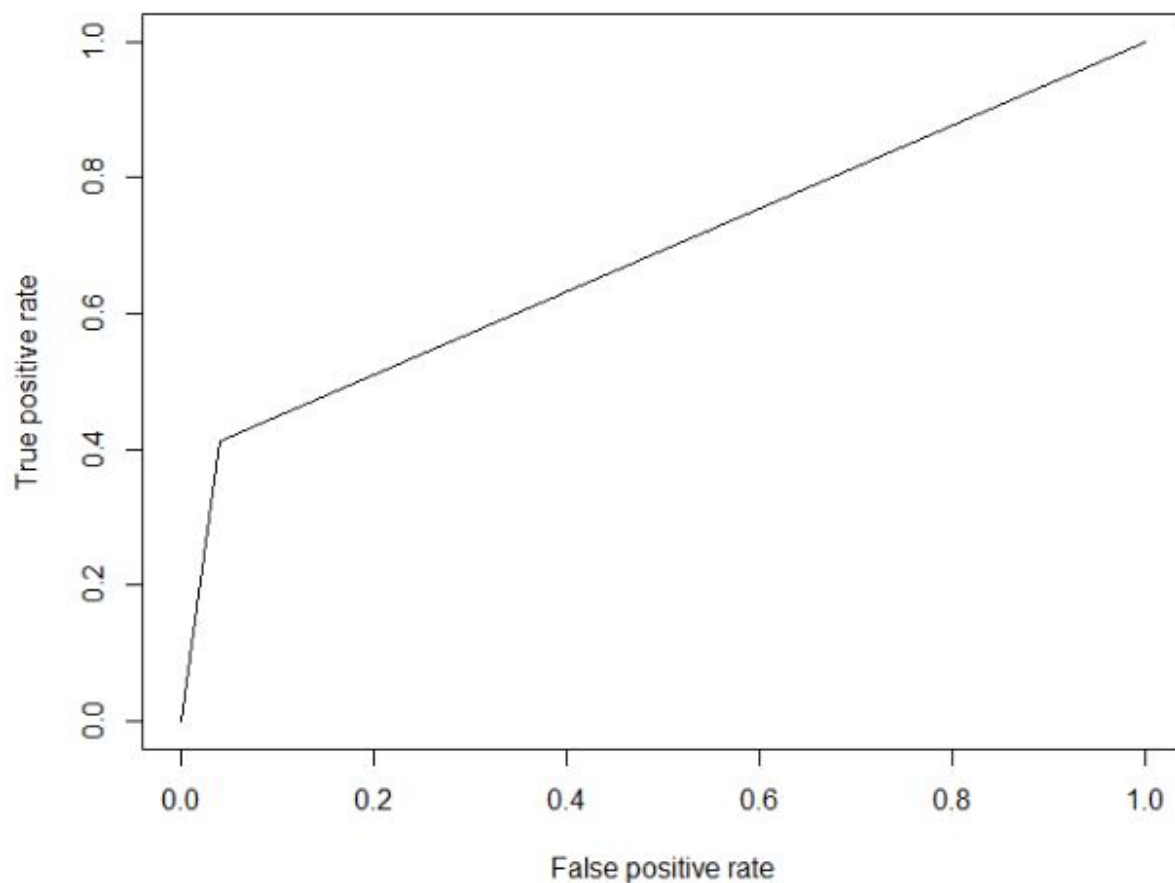
Metrika	Ostvareni rezultat
Pos. Pred. Value	0.8981
Balanced Accuracy	0.6864
Kappa statistics	0.4388
Lower bound of 95% CI	0.8307

ROC kriva

ROC krivu možemo skicirati koristeći sljedeće komande:

```
perf<-prediction(predictions=c(as.factor(ldaPredictions$class)),labels=test_im$Attrition)
graph = performance(perf, measure = "tpr", x.measure = "fpr")
plot(graph)
```

Nakon izvršavanja koda, dobijen je sljedeći grafik sa ROC krivom:



Zadatak 2 (3 boda)

Koristeći attrition set podataka (spojeni attrition_train.csv i attrition_test.csv), potrebno je postaviti hipotezu koju ćete ispitati pomoću modela višestruke linearne regresije. **Cilj je dobiti što je moguće veći R^2 , adjusted R^2 , i što je moguće manji Mean Absolute Error (MAE) za ispitanu hipotezu.** Tjj. potrebno je evaluirati podobnost modela, i izvršiti sve potrebne korake kako biste unaprijedili inicijalno kreirani model:

1) Ispitivanje pretpostavke linearnosti

Pretpostavimo da set podataka modeliramo nelinearnom regresijom, razlog ovakve pretpostavke predstavlja predznanje o ulaznim i izlaznoj varijabli tj. da je taj odnos nelinearan (zadaca 1), što ćemo dokazati kroz sljedećih niz koraka. Par karakteristika:

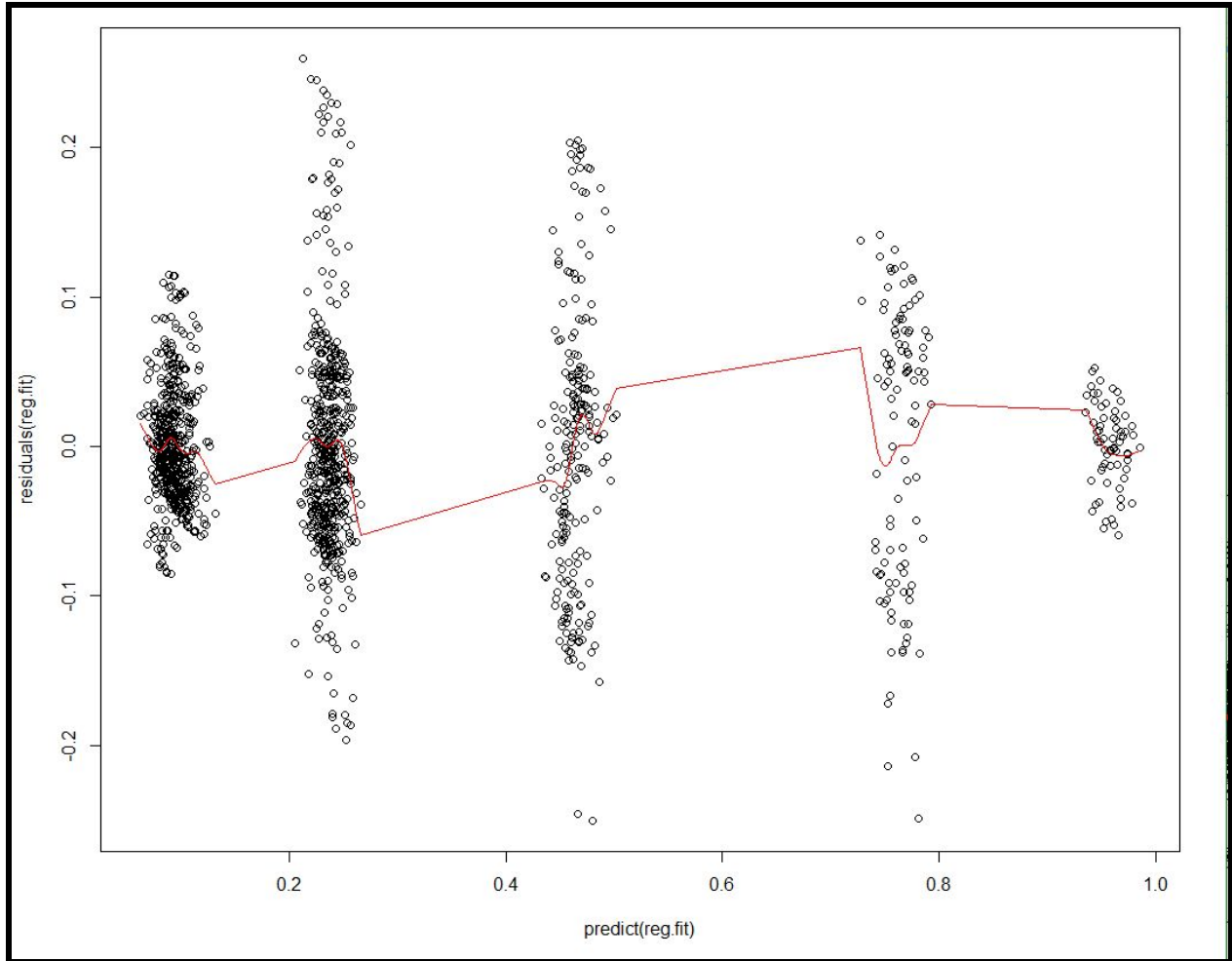
- Iz prijašnje zadace attrition.full varijabla predstavlja ukupni data set spojen zajedno i očišćen
- Iskoristit ćemo isti model koji nam je dao najbolje performanse, ali da bi prepoznavali kontinualnu varijablu koristit ćemo varijablu MonthlyIncome (random izabrana kontinualna varijabla)
- U nastavku je prikazan inicijalni kod pomoću kojeg pokazujemo pretpostavku modeliranja nelinearnom višestrukom regresijom:

```
attrition.testing = attrition.full  
# korak 1 evaluacija modela
```

```
reg.fit<-lm(MonthlyIncome~NumCompaniesWorked+DistanceFromHome+Department+BusinessTravel+WorkLifeBalance+TrainingTimesLastYear+TotalWorkingYears+JobSatisfaction+MaritalStatus+JobLevel+HourlyRate+JobInvolvement+EnvironmentSatisfaction+EducationField+OverTime+PercentSalaryHike+StockOptionLevel+YearsWithCurrManager+YearsInCurrentRole+YearsAtCompany,data=attrition.testing)
```

```
plot(predict(reg.fit),residuals(reg.fit))  
lines(smooth.spline(predict(reg.fit),residuals(reg.fit)),col="red")
```

Izvršavanjem sljedećeg koda kao rezultat prikaza ćemo dobiti sljedeće:



Iz prethodne slike možemo zaključiti da odnos predikcijskih vrijednosti u odnosu na rezidualne vrijednosti tj. da spline vrijednost odlično prati ponašanje samih prikaza tih varijabli tj. prilikom plotanja odnosa te dvije vrijednosti.

Ispitajmo sada vrijednosti R^2 , adjusted R^2 i MAE:

```
sumari = summary(reg.fit)

sumari$adj.r.squared
sumari$r.squared
MAE(predict(reg.fit), attrition.full$MonthlyIncome)
```


Kao rezultat poziva prethodnog koda imamo:

```
>
> sumari = summary(reg.fit)
>
> sumari$adj.r.squared
[1] 0.9252628
> sumari$r.squared
[1] 0.9273488
> MAE(predict(reg.fit), attrition.full$MonthlyIncome)
[1] 0.04888009
> |
```

Shodno tome možemo zaključiti da je ispitivanje nelinearnosti modela veoma pozitivno, te da smo krenuli ispitivati linearnost modela sa random varijablom ulazne predikcije dobili smo loš prethodni opisani odnos u svakom slučaju, shodno tome uticaj više varijabli je očigledan i ne moramo vršiti nikakvu transformaciju prediktora.

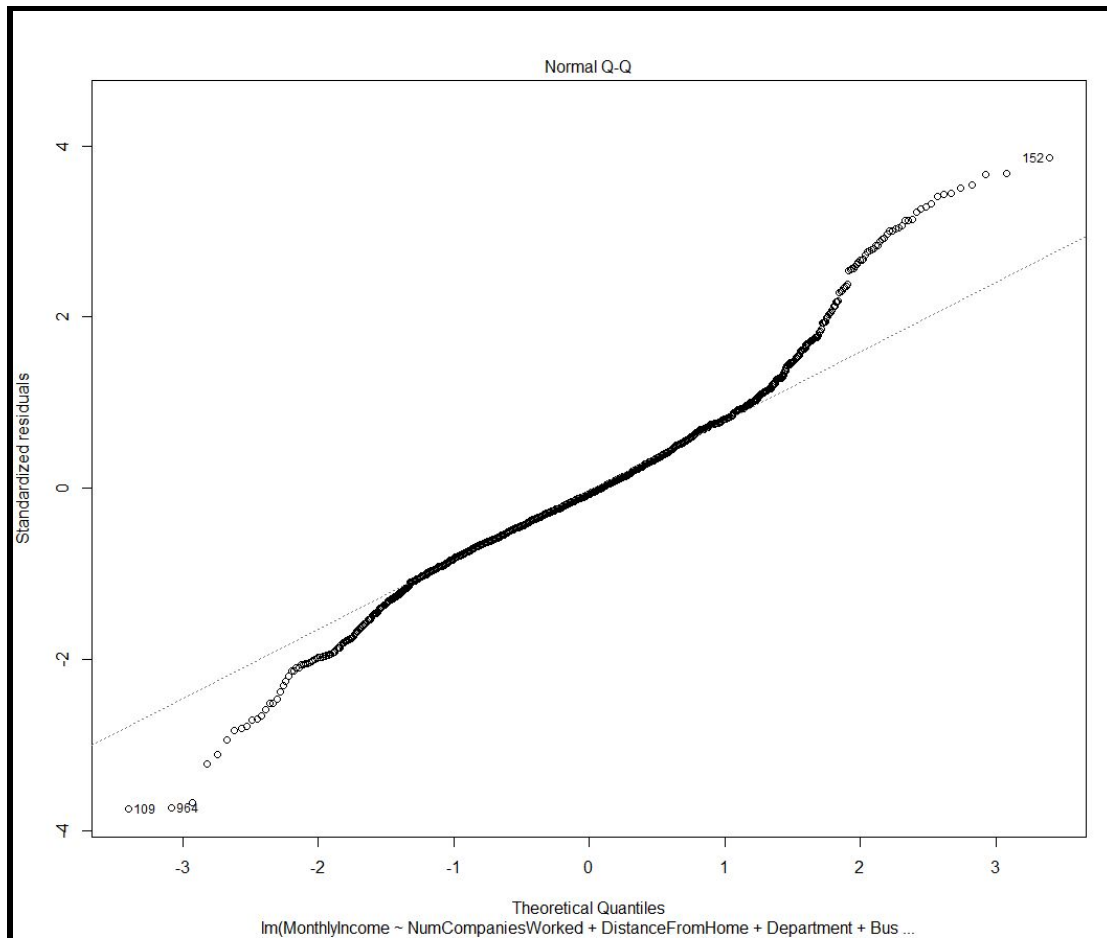
Na osnovu prethodnog grafika također možemo zaključiti da će najveći napredak za naš model biti detekcija nepodobnih izlaznih vrijednosti (eng. outliers) i otklananje istih, jer vidimo da su zastupljeni na grafiku.

2) Analiza auto-korelacije rezidualnih vrijednosti

Ukoliko je regresijski model koji smo dobili pogodan, a na osnovu prethodnih rezultata i logikom da R^2 što veći, a MAE što manje možemo i pretpostaviti, odnosno ispravan, neophodno je da **ne postoji** auto-korelacija rezidualnih vrijednosti.

Za ispitivanje toga primijenimo Durbin-Watson test, a prije toga moramo se uvjeriti da dobivene rezidualne vrijednosti prate normalnu distribuciju, to ćemo uraditi pomoću Q-Q plot:

```
# korak 2 korelacija rezidualnih vrijednosti
## Q-Q plot
plot(density(reg.fit$residuals), main="Residuals", xlab="Value")
plot(reg.fit, which=2)
```



Iz prethodnog grafika vidimo da dobivene rezidualne vrijednosti prate normalnu distribuciju u većinskoj mjeri, te da ovo ispitivanje zadovoljava normalnu distribuciju. Radi komparacije nakon 4og koraka ćemo ponoviti ovaj proces i vidjeti razliku na grafu (očekivanja da ćemo se približiti normalnoj distribuciji još bliže). Zaključujemo da rezidualne vrijednosti prate normalnu distribuciju.

Na kraju za ispitivanje da postoji auto-korelacija rezidualnih vrijednosti koristit ćemo Durbin-Watson test:

```
## durbin watston test
```

```
dwtest(MonthlyIncome~NumCompaniesWorked+DistanceFromHome+Department+BusinessTravel+WorkLifeBalance+TrainingTimesLastYear+TotalWorkingYears+JobSatisfaction+MaritalStatus+JobLevel+HourlyRate+JobInvolvement+EnvironmentSatisfaction+EducationField+OverTime+PercentSalaryHike+StockOptionLevel+YearsWithCurrManager+YearsInCurrentRole+YearsAtCompany,data=attrition.testing)
```

Kao rezultat dw testa dobijamo:

```
DW = 2.0905, p-value = 0.9589  
alternative hypothesis: true autocorrelation is greater than 0
```

Vidimo da durbin-watson test govori da ne postoji auto-korelacija rezidualnih vrijednosti.

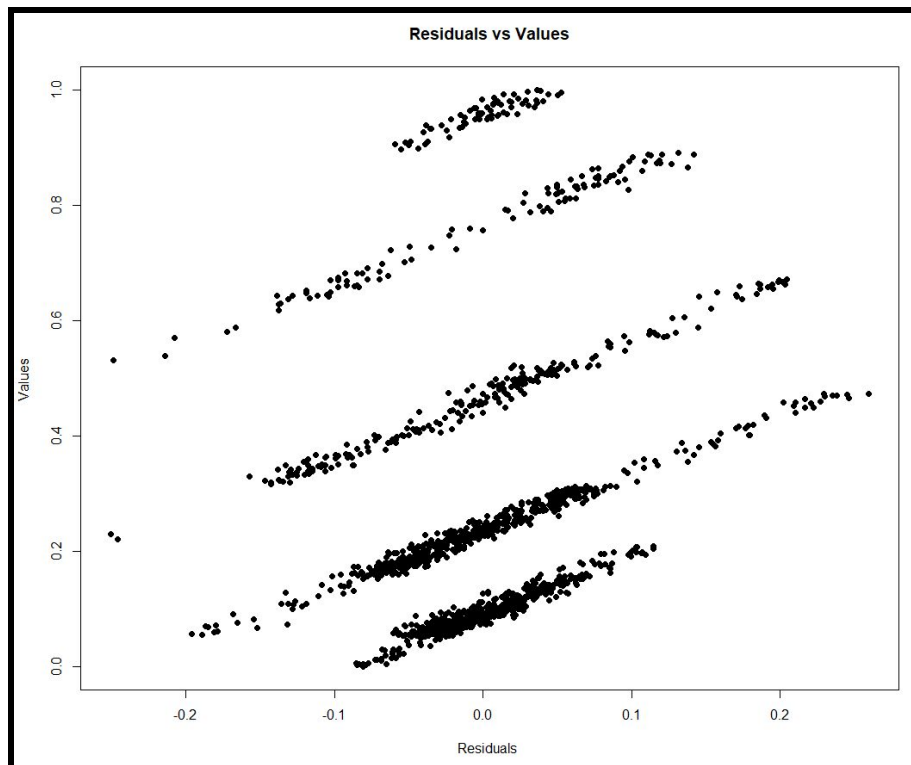
3) Analiza "konstantnosti" varianse rezidualnih vrijednosti

Da bi napravili analizu konstantnosti varianse rezidualni vrijednosti, iscratati ćemo scatter plot rezidualnih vrijednosti u odnosu na predviđene vrijednosti izlazne varijable. Ukoliko takav scatter plot ima oblik "lijevka" znači da rezidualne vrijednosti nemaju konstantnu variansu. To ćemo uraditi pomoću sljedećeg koda:

```
# korak 3 - nekonstantna variansa rezidualnih vrijednosti
```

```
plot(reg.fit$residuals, attrition.testing$MonthlyIncome, main="Residuals vs  
Values", xlab="Residuals ", ylab="Values ", pch=19)
```

Kao rezultat pokretanja ovog koda imamo:



Vidimo da naš graf nema izgled lijevka, odnosno da rezidualne vrijednosti imaju konstantnu varijansu.

4) Detekcija nepodobnih izlaznih vrijednosti (eng. outliers)

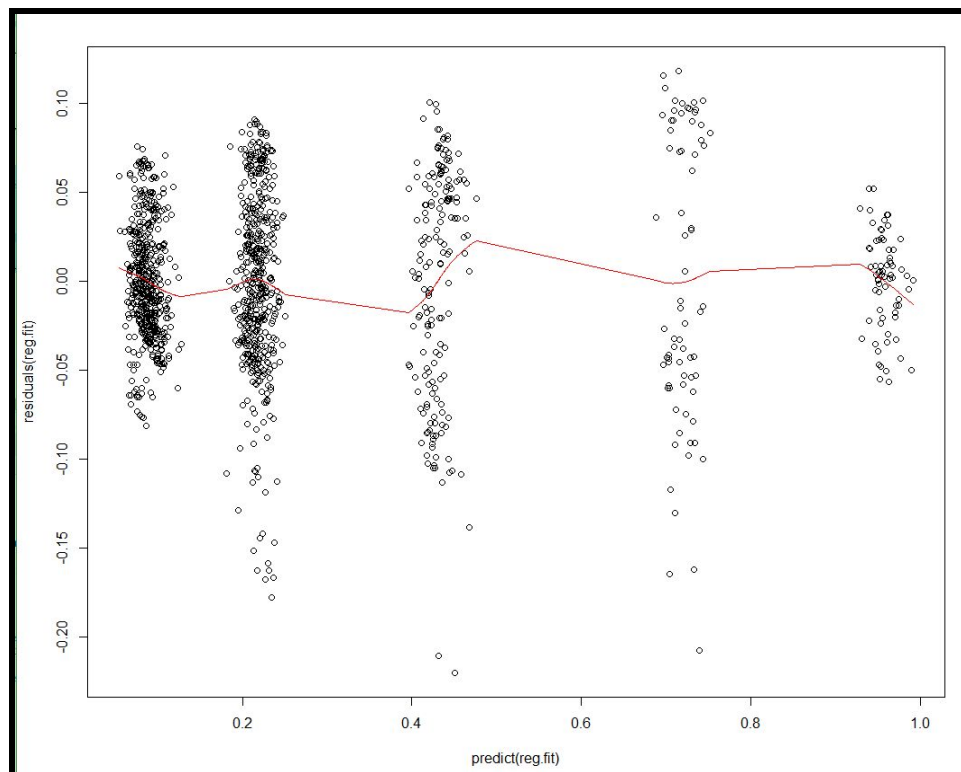
U ovom koraku potrebno je pronaći određenu granicu pomoću koje ćemo otkloniti naše outliers, odnosno određenu granicu za koju ćemo definisati šta je outlier. U nastavku je prikazan kod:

```
# korak 4 uklanjanje outlierera
vectorToClear = as.numeric(names(rstudent(reg.fit)[rstudent(reg.fit)>1]))
attrition.clear = attrition.testing[-vectorToClear,]

reg.fit<-lm(MonthlyIncome~NumCompaniesWorked+DistanceFromHome+Department+BusinessTravel+WorkLifeBalance+TrainingTimesLastYear+TotalWorkingYears+JobSatisfaction+MaritalStatus+JobLevel+HourlyRate+JobInvolvement+EnvironmentSatisfaction+EducationField+Overtime+PercentSalaryHike+StockOptionLevel+YearsWithCurrManager+YearsInCurrentRole+YearsAtCompany,data=attrition.clear)

plot(predict(reg.fit),residuals(reg.fit))
lines(smooth.spline(predict(reg.fit),residuals(reg.fit)),col="red")
```

Rezultat poziva prethodnog koda je prikazan sljedećom slikom:



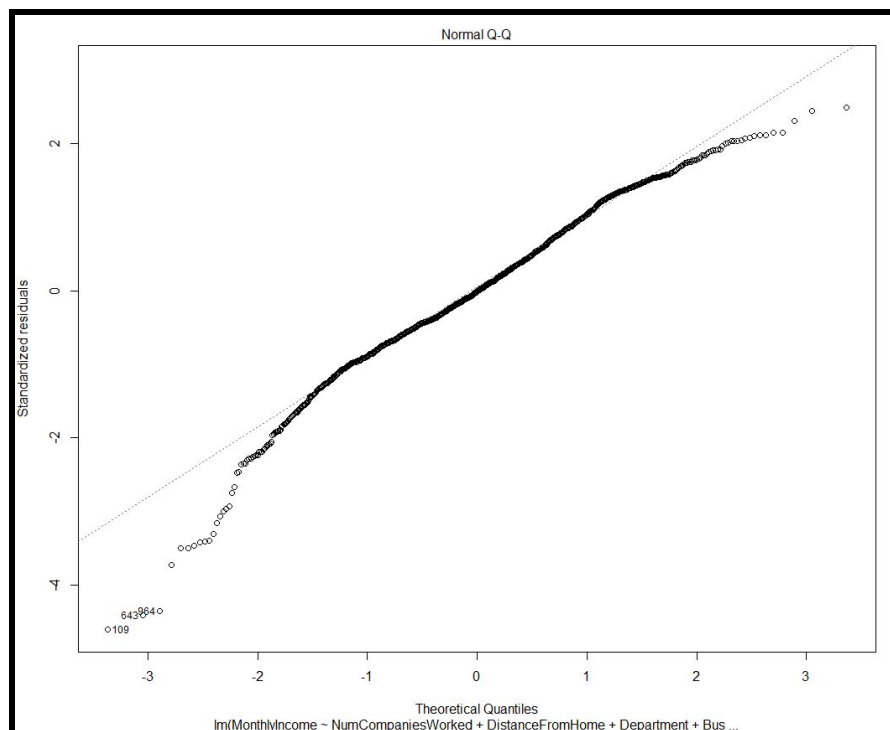
Iterativnim postupkom određena je granica 1, prikazana u sljedećem kodu, a prilikom izbora granica se gledalo koliko instanci dataset-a se ostavljaju kao outliers, u odnosu na rezultate R^2 , adj R^2 , MAE.

```
sumari = summary(reg.fit)
sumari$adj.r.squared
sumari$r.squared
MAE(predict(reg.fit), attrition.clear$MonthlyIncome)
```

Izvršavanjem prethod kod kao rezultat dobijamo sljedeće rezultate:

```
> sumari = summary(reg.fit)
>
> sumari$adj.r.squared
[1] 0.9582295
> sumari$r.squared
[1] 0.9595479
> MAE(predict(reg.fit), attrition.clear$MonthlyIncome)
[1] 0.03713419
> |
```

Vidimo povećanje u vrijednostima R^2 , adj R^2 i smanjenje MAE, što je i bilo inicijalno za cilj. Ponovimo na kraju proces iz koraka 2, i prikazimo normalnu distribuciju:



Vidimo da su rezultati za normalnu distribuciju bolji, te ujedno i tačna pretpostavka.

5) Detekcija nepodobnih ulaznih vrijednosti (eng. high leverage points)

Detekciju nepodobnih ulaznih vrijednosti možemo uraditi pomoću hat values na sljedeći način:

```
# korak 5 nepodobne vrijednosti ulaznih varijabli (eng. high leverage points)

hatvalues(reg.fit)[hatvalues(reg.fit)>10*(ncol(attrition.clear[, -10])+1)/nrow(a
ttrition.clear)]
```

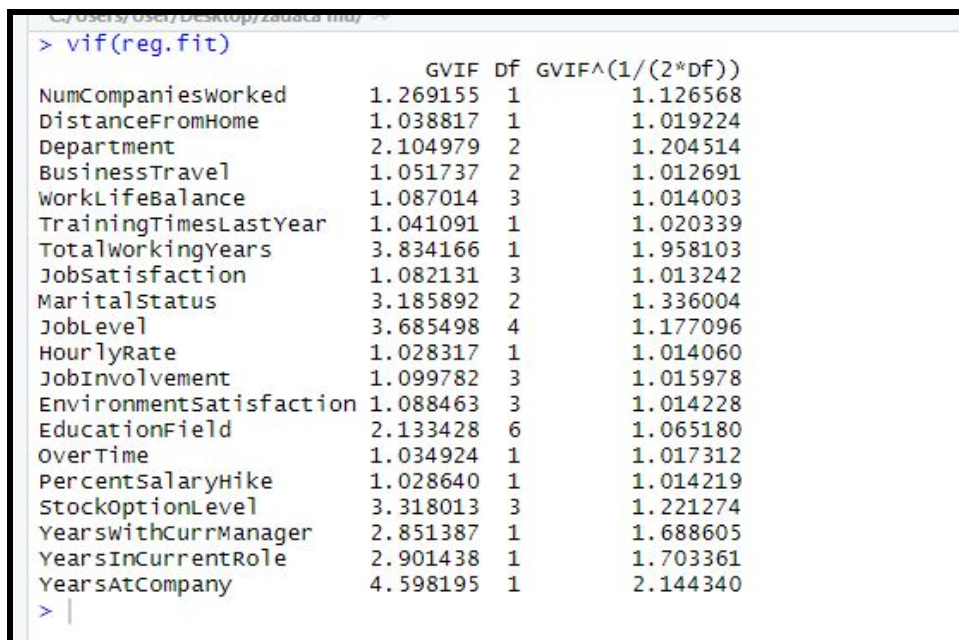
Izvršavanjem prethodnog koda dobijemo 0 ulaznih instanci koji zadovoljavaju uslove, te možemo zaključiti da nemamo nepodobnih ulaznih vrijednosti.

6) Analiza ko-linearnosti ulaznih varijabli

Analizirati ćemo ko-linearnost ulaznih varijabli pomoću VIF funkcije odnosno, kolinearnost našeg trenutnog modela pomoću sljedećeg koda:

```
# korak 6 ko-linearnost ulaznih varijabli
vif(reg.fit)
```

Na osnovu prethodnog koda dobijamo sljedeću tabelu:



	GVIF	Df	GVIF^(1/(2*Df))
NumCompaniesWorked	1.269155	1	1.126568
DistanceFromHome	1.038817	1	1.019224
Department	2.104979	2	1.204514
BusinessTravel	1.051737	2	1.012691
WorkLifeBalance	1.087014	3	1.014003
TrainingTimesLastYear	1.041091	1	1.020339
TotalWorkingYears	3.834166	1	1.958103
JobSatisfaction	1.082131	3	1.013242
MaritalStatus	3.185892	2	1.336004
JobLevel	3.685498	4	1.177096
HourlyRate	1.028317	1	1.014060
JobInvolvement	1.099782	3	1.015978
Environmentsatisfaction	1.088463	3	1.014228
EducationField	2.133428	6	1.065180
OverTime	1.034924	1	1.017312
Percentsalaryhike	1.028640	1	1.014219
StockOptionLevel	3.318013	3	1.221274
YearswithCurrManager	2.851387	1	1.688605
YearsInCurrentRole	2.901438	1	1.703361
YearsAtCompany	4.598195	1	2.144340

VIF vrijednosti veće od 5-10 predstavljaju problematičnu količinu ko-linearnosti u setu podataka. Kao što vidimo za naš model (treća kolona), takvih vrijednosti nema, što ukazuje na dobar pre-processing u zadaći 1.

Pored 6 tačaka za evaluaciju podobnosti regresijskog modela, potrebno je koristiti najmanje **dvije metode za selekciju** najznačajnijih varijabli koje trebaju biti uključene u finalni regresijski model (npr. forward selection, ili backward selection). Objasnite odabrane metode selekcije i razlike u rezultatima istih ukoliko razlike postoje.

Da bismo evaluirali podobnost regresijskog modela, koristiti ćemo metode `stepAIC` i `leapBackward` za selekciju najznačajnijih varijabli, unutar R-a.

Metoda stepAIC je najpopularnija metoda za odabir najboljih varijabli. StepAIC ne znači nužno i poboljšanje performansi modela, no metoda se koristi za pojednostavljenje modela bez utjecaja na performanse. AIC kvantificira količinu gubitka informacija zbog ovog pojednostavljenja. AIC označava Akaike informacije kriterije. Ako imamo dva modela, model sa što manjom vrijednošću AIC predstavlja ono što tražimo. AIC je veoma sličan adjusted R^2 , ali također kažnjava ukoliko se dodavaju varijable u model u većoj mjeri u odnosu na R^2

Backward selection započinje sa svim prediktorima u modelu (puni model), iterativno uklanja prediktore s najmanje doprinosa i zaustavlja se kada su svi prediktori u tom modelu statistički značajni (forward selection radi kontra tome i kreće sa 0 varijabli). Ove dvije metode ćemo prikazati na našem modelu, prvenstveno sa stepAIC metodom, te ćemo se ograničiti na prvih nekoliko varijabli.

```
# lmStepAIC
train.control <- trainControl(method = "cv", number = 10)
step.model <-
train(MonthlyIncome~NumCompaniesWorked+DistanceFromHome+Department+BusinessTravel+WorkLifeBalance+TrainingTimesLastYear+TotalWorkingYears+JobSatisfaction+MaritalStatus+JobLevel+HourlyRate+JobInvolvement+EnvironmentSatisfaction+EducationField+OverTime+PercentSalaryHike+StockOptionLevel+YearsWithCurrManager+YearsInCurrentRole+YearsAtCompany, data = attrition.clear, trControl = train.control,
method = "lmStepAIC", trace = FALSE)step.model$results
step.model$finalModel
summary(step.model$finalModel)
```


Rezultat poziva prethodnog koda je sljedeći:

```
> summary(step.model$finalModel)

Call:
lm(formula = .outcome ~ NumCompaniesworked + DistanceFromHome +
    BusinessTravelTravel_Rarely + TotalWorkingYears + JobLevel2 +
    JobLevel3 + JobLevel4 + JobLevel5 + JobInvolvement2 + JobInvolvement3 +
    JobInvolvement4 + Environmentsatisfaction4 + StockOptionLevel1 +
    StockOptionLevel3 + YearsInCurrentRole + YearsAtCompany,
    data = dat)

Residuals:
    Min       1Q   Median       3Q      Max
-0.223100 -0.028388 -0.001207  0.032823  0.114756

Coefficients:
                Estimate Std. Error t value Pr(>|t|)
(Intercept)      0.0886141  0.0068125   13.008 < 2e-16 ***
NumCompaniesworked  0.0019317  0.0006013    3.212 0.001349 **
DistanceFromHome   -0.0003727  0.0001683   -2.215 0.026927 *
BusinessTravelTravel_Rarely 0.0065161  0.0029713    2.193 0.028486 *
TotalWorkingYears  0.0012245  0.0003322    3.686 0.000237 ***
JobLevel2          0.1219873  0.0034053   35.822 < 2e-16 ***
JobLevel3          0.3271114  0.0050704   64.514 < 2e-16 ***
JobLevel4          0.6052325  0.0082668   73.213 < 2e-16 ***
JobLevel5          0.8413284  0.0085788   98.071 < 2e-16 ***
JobInvolvement2    -0.0133701  0.0061673   -2.168 0.030349 *
JobInvolvement3    -0.0175097  0.0058281   -3.004 0.002713 **
JobInvolvement4    -0.0145657  0.0070143   -2.077 0.038038 *
Environmentsatisfaction4 -0.0056193  0.0029302   -1.918 0.055367 .
StockOptionLevel1   0.0074573  0.0028046    2.659 0.007935 **
StockOptionLevel3   -0.0097206  0.0059683   -1.629 0.103625
YearsInCurrentRole  0.0015638  0.0006019    2.598 0.009477 **
YearsAtCompany     -0.0007204  0.0004150   -1.736 0.082778 .
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.04827 on 1283 degrees of freedom
Multiple R-squared:  0.9592,    Adjusted R-squared:  0.9587
F-statistic: 1884 on 16 and 1283 DF,  p-value: < 2.2e-16
```

Vidimo da najznačajnije varijable su JobLevel (JobLevel[2-5], metoda prilikom treniranja napravi ovoliko podjela za JobLevel jer predstavlja factor varijablu), zatim TotalWorkingYears, NumCompaniesWorked i StockOptionLevel1 (naravno sa slike se vide još značajnih, ali radi komparacije da backward metodom uzeli smo ovu granicu.)

U nastavku ćemo prikazati backward model i vidjeti koje on varijable nama daje kao najznačajnije (ograničiti ćemo se radi jednostavnosti sa nvmax = 7), te uporediti varijable ova dva modela.


```

train.control <- trainControl(method = "cv", number = 10)
step.model
train(MonthlyIncome~NumCompaniesWorked+DistanceFromHome+Department+BusinessTravel+WorkLifeBalance+TrainingTimesLastYear+TotalWorkingYears+JobSatisfaction+MaritalStatus+JobLevel+HourlyRate+JobInvolvement+EnvironmentSatisfaction+EducationField+OverTime+PercentSalaryHike+StockOptionLevel+YearsWithCurrManager+YearsInCurrentRole+YearsAtCompany, data = attrition.clear, method = "leapBackward",
tuneGrid = data.frame(nvmax = 7), trControl = train.control, trace = FALSE)

step.model$results
step.model$finalModel
summary(step.model$finalModel)

```

Izvršavanjem prethodnog koda dobijamo:

```

1 subsets of each size up to 7
selection Algorithm: backward
NumCompaniesWorked DistanceFromHome DepartmentResearch & Development DepartmentsSales BusinessTravelTravel_Frequently
1 (1)
2 (1)
3 (1)
4 (1)
5 (1)
6 (1)
7 (1)
BusinessTravelTravel_Rarely workLifeBalanceBest workLifeBalanceBetter workLifeBalanceGood TrainingTimesLastYear TotalworkingYears
1 (1)
2 (1)
3 (1)
4 (1)
5 (1)
6 (1)
7 (1)
JobSatisfaction2 JobSatisfaction3 JobSatisfaction4 MaritalStatusMarried MaritalStatusSingle JobLevel2 JobLevel3 JobLevel4 JobLevel5 HourlyRate
1 (1)
2 (1)
3 (1)
4 (1)
5 (1)
6 (1)
7 (1)
JobInvolvement2 JobInvolvement3 JobInvolvement4 EnvironmentSatisfaction2 EnvironmentSatisfaction3 EnvironmentSatisfaction4
1 (1)
2 (1)
3 (1)
4 (1)
5 (1)
6 (1)
7 (1)
EducationFieldHuman Resources EducationFieldLife Sciences EducationFieldMarketing EducationFieldMedical EducationFieldother
1 (1)
2 (1)
3 (1)
4 (1)
5 (1)
6 (1)
7 (1)
EducationFieldTechnical Degree OverTimeYes PercentSalaryHike StockOptionLevel1 StockOptionLevel2 StockOptionLevel3 YearsWithCurrManager
1 (1)
2 (1)
3 (1)
4 (1)
5 (1)
6 (1)
7 (1)
YearsInCurrentrole YearsAtCompany
1 (1)
2 (1)
3 (1)
4 (1)
5 (1)
6 (1)
7 (1)

```

Vidimo da smo dobili iste varijable kao u prethodnom modelu, odnosno da nema razlike između rezultata u oba ova modela, odnosno da daju iste varijable kao najznačajnije.