Univerzitet u Sarajevu

Elektrotehnički fakultet

Odsjek za računarstvo i informatiku

Mašinsko učenje

ZADAĆA 2

- izvještaj -

Studenti:
Demir Faris 1930/18444
Đokić Milica 1870/18536
Hodžić Lejla 1904/18547

Grupa 4

Sarajevo, januar 2022.

Zadatak 1. (Izgradnja modela klasifikacije)

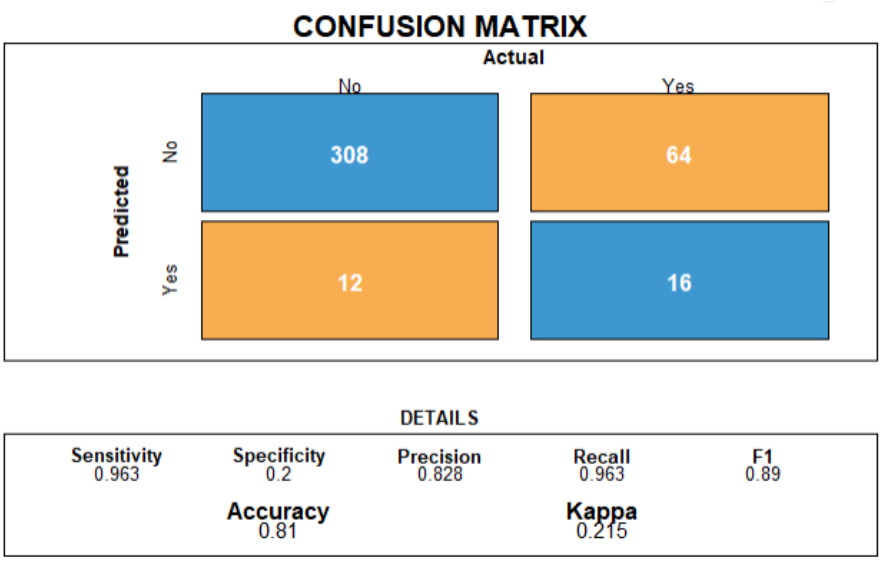
a) Izgradnja predikcijskih modela za zadani problem klasifikacije

U okviru pripreme seta podataka smo, za sve modele predikcije, izvršili zamjenu nedostajućih vrijednosti sa medianom (ukoliko je varijabla numerička) i mode vrijednosti (ukoliko je varijabla kategorička), te odbacivanje outlier-a. Pored navedenog, izbacili smo varijable MaxTemp i Cloud5pm koje su imale najveći stepen korelacije. Na kraju smo izvršili i min-max normalizaciju varijable Pressure9am.

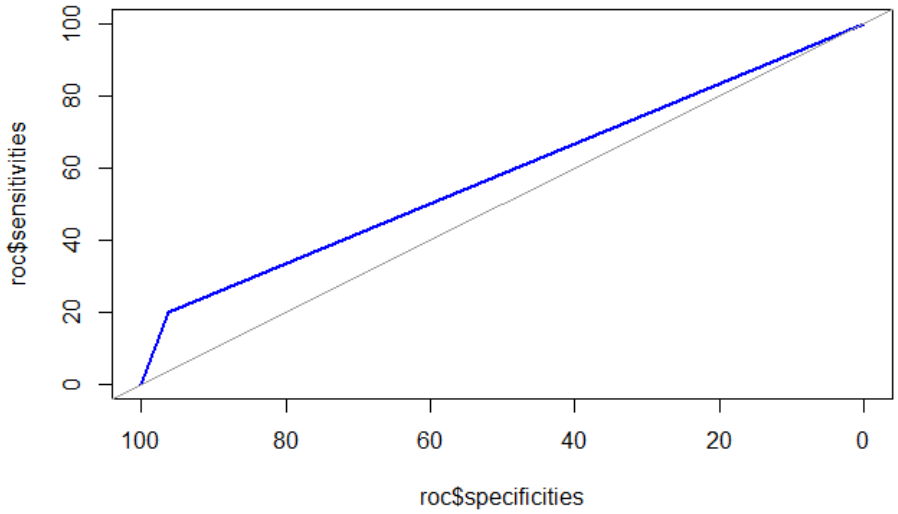
- KNN model predikcije

Kako bi mogli izgraditi KNN model predikcije, bilo je potrebno pretvoriti sve nenumeričke varijable u numeričke, obzirom da KNN radi samo sa numeričkim varijablama. Podjelu na trening i testni skup smo izvršili sa indeksom podjele 0.8.

Model je izgrađen pozivom funkcije `knn()`, pri čemu broj susjeda je postavljen na 15. Prikaz evaluacije osnovnog modela je dat ispod.



Konfuzijska matrica za osnovni model



ROC kriva za osnovni model

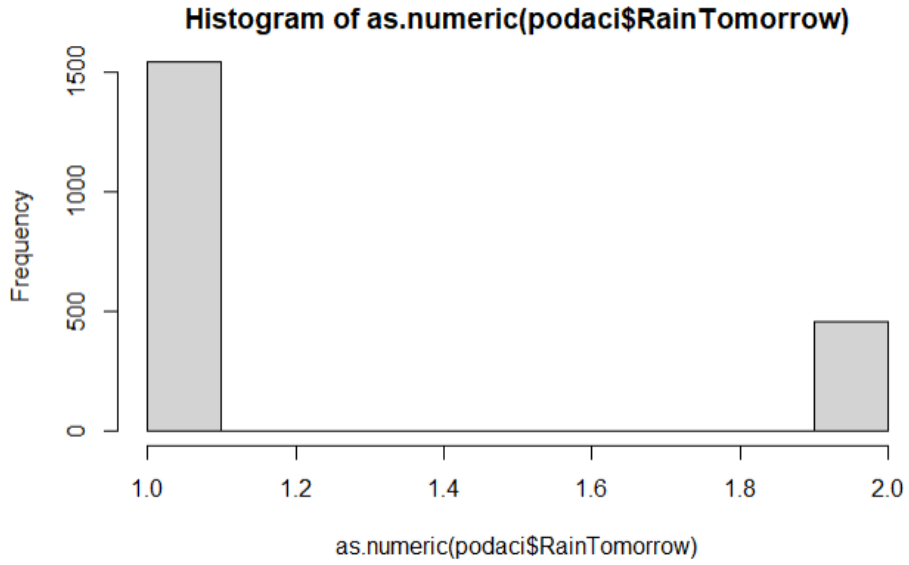
Za cross-validaciju je korištena pomoćna funkcija `kfold_knn()`, koja prima dva parametra: podaci i k (prilagođena funkcija sa zadaće 1). Prikazane su vrijednosti 10-fold i 5-fold validacije.

```
10-fold validacija
Najveca tacnost: 0.83 , fold: 2, najveca kappa: 0.384058 , fold: 5
Najmanja tacnost: 0.745 , fold: 6, najmanja kappa: 0.1126543 , fold: 8
Srednja tacnost: 0.7918116, srednja kappa: 0.2437423

5-fold validacija
Najveca tacnost: 0.8220551 , fold: 4, najveca kappa: 0.2824287 , fold: 1
Najmanja tacnost: 0.7475 , fold: 3, najmanja kappa: 0.1452991 , fold: 3
Srednja tacnost: 0.7877895, srednja kappa: 0.2231437
```

Za bootstrap je korištena pomoćna funkcija `bootstrap_knn()`, koja prima dva parametra: podaci i k (prilagođena funkcija sa zadaće 1). Prikazane su vrijednosti 10-fold i 5-fold bootstrapinga.

```
10-fold bootstrap
Najveca tacnost: 0.8643216 , fold: 2, najveca kappa: 0.4594024 , fold: 2
Najmanja tacnost: 0.7135678 , fold: 1, najmanja kappa: 0.06971213 , fold: 1
Srednja tacnost: 0.8090452, srednja kappa: 0.295527

5-fold bootstrap
Najveca tacnost: 0.8220551 , fold: 2, najveca kappa: 0.3626915 , fold: 2
Najmanja tacnost: 0.7744361 , fold: 1, najmanja kappa: 0.119594 , fold: 1
Srednja tacnost: 0.8075188, srednja kappa: 0.2606227
```

Na osnovu histograma balansiranosti podataka koji je prikazan na slici ispod, možemo zaključiti da podaci nisu balansirani.



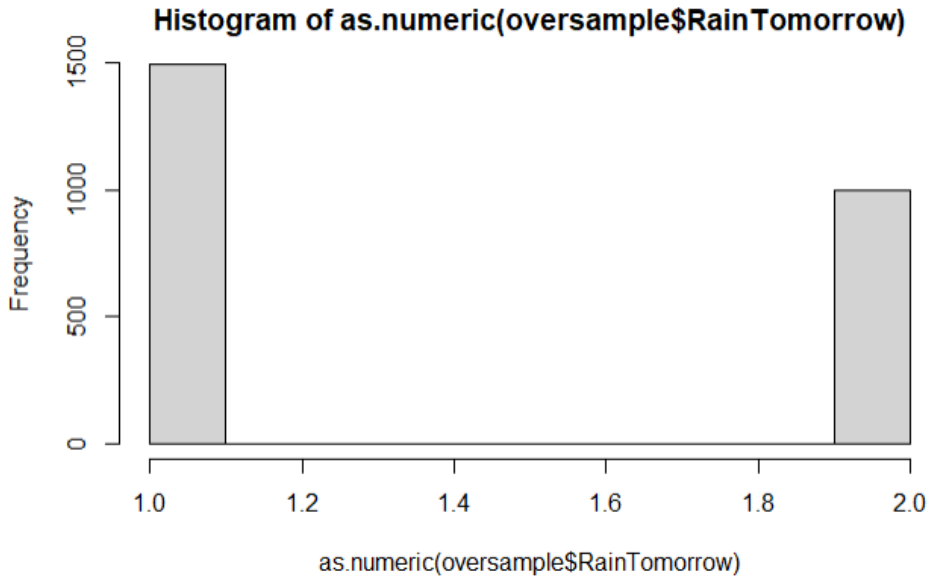
Analiziranjem ranije prikazanih rezultata za nebalansirani set podataka, možemo zaključiti da postoji značajan prostor za poboljšanje određenih metrika, kao što su kappa i specifičnost, koje imaju male vrijednosti. Također, ROC kriva nam sugestira da naš klasifikator ne posjeduje zadovoljavajuće performanse, te bi ih trebalo dodatno unaprijediti ukoliko to bude moguće.

Balansiranje podataka (oversampling)

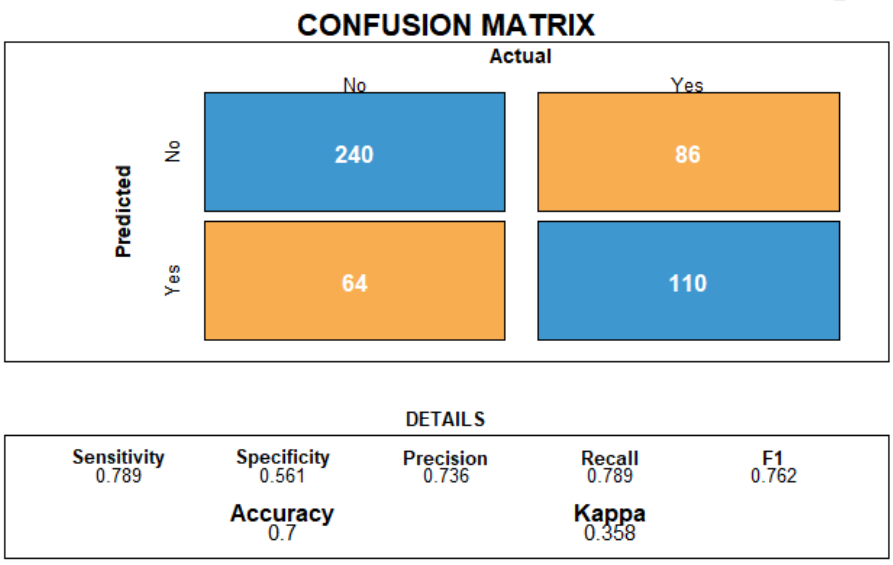
Balansiranje podataka je izvršeno korištenjem funkcije `ovun.sample()` na sljedeći način

```
oversample <- ovun.sample(RainTomorrow ~ ., data = podaci, method = "both",
N = 2500, p=0.4)$data
```

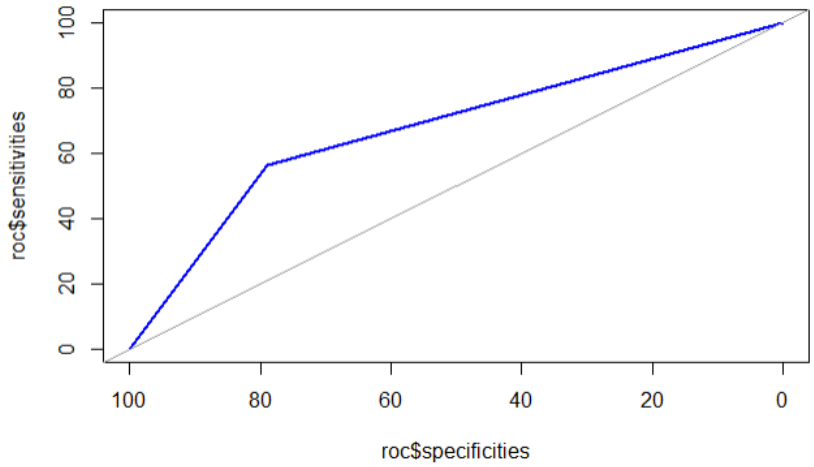
Prikaz histograma podataka nakon balansiranja je dat na sljedećoj slici:



Evaluacija modela nakon balansiranja je data ispod:



Konfuzijska matrica za model nakon balansiranja



ROC kriva nakon balansiranja

Cross-validacija i bootstrap su izvršeni na isti način kako je opisano ranije, a rezultati su prikazani na slikama ispod:

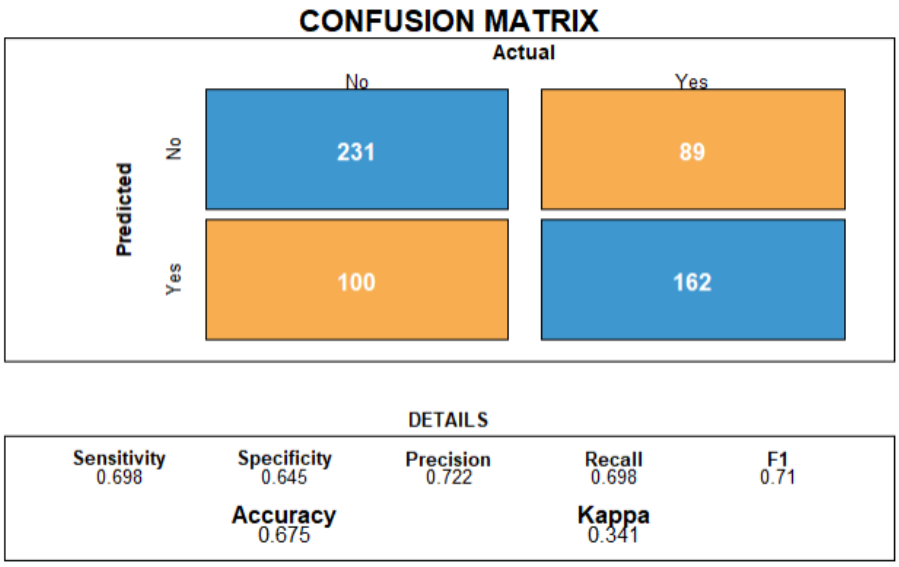
```
10-fold validacija
Najveca tacnost: 0.736 , fold: 3, najveca kappa: 0.4503298 , fold: 3
Najmanja tacnost: 0.648 , fold: 6, najmanja kappa: 0.2337699 , fold: 7
Srednja tacnost: 0.6948, srednja kappa: 0.354313

5-fold validacija
Najveca tacnost: 0.724 , fold: 5, najveca kappa: 0.4242707 , fold: 5
Najmanja tacnost: 0.674 , fold: 1, najmanja kappa: 0.3143424 , fold: 1
Srednja tacnost: 0.6956, srednja kappa: 0.3584874

10-fold bootstrap
Najveca tacnost: 0.772 , fold: 2, najveca kappa: 0.5273005 , fold: 7
Najmanja tacnost: 0.696 , fold: 10, najmanja kappa: 0.3475723 , fold: 10
Srednja tacnost: 0.7416, srednja kappa: 0.4520996

5-fold bootstrap
Najveca tacnost: 0.75 , fold: 4, najveca kappa: 0.4681756 , fold: 4
Najmanja tacnost: 0.72 , fold: 1, najmanja kappa: 0.4267933 , fold: 1
Srednja tacnost: 0.7332, srednja kappa: 0.4398655
```

Analiziranjem dobivenih rezultata nakon balansiranja podataka, možemo zaključiti da su se značajnije poboljšale vrijednosti metrika kappa i specifičnosti, a vrijednosti ostalih metrika su se neznatno smanjile. Također, ROC kriva nakon balansiranja ukazuje na najbolje performanse klasifikatora.

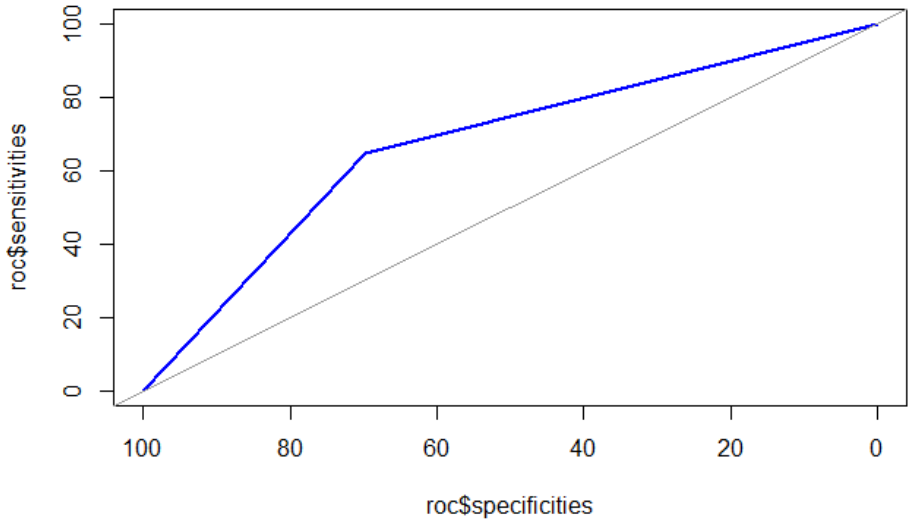
Balansiranje podataka (SMOTE)

Za primjenu SMOTE algoritma za balansiranje podataka korištena je funkcija `SMOTE()`. Korišteni su parametri `K = 7` (broj susjeda koji se koristi za kreiranje novih vještačkih instanci) i `dup_size = 0` (0 označava da će se generisanje vještačkih instanci zaustaviti nakon što se dostigne isti broj instanci kao u više zastupljenoj klasi).

Evaluacija modela nakon SMOTE balansiranja je prikazana na slikama ispod:



Prikaz konfuzijske matrice nakon SMOTE-a



Prikaz ROC krive nakon SMOTE-a

Rezultati 10-fold i 5-fold cross-validacije i 10-fold i 5-fold bootstrapinga su dati ispod:

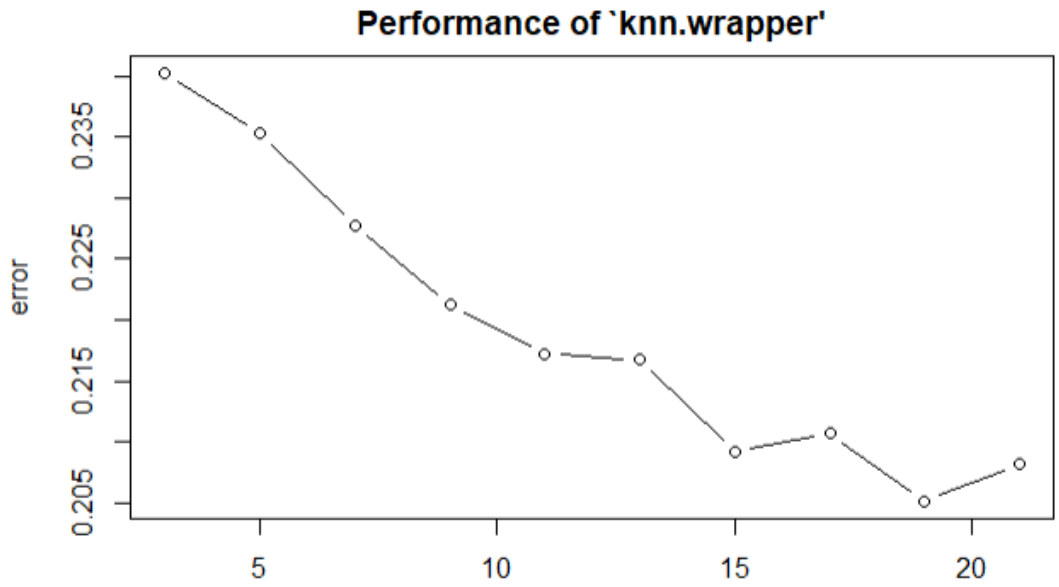
```
10-fold validacija
Najveca tacnost: 0.7457045 , fold: 5, najveca kappa: 0.4948391 , fold: 5
Najmanja tacnost: 0.6975945 , fold: 10, najmanja kappa: 0.3992681 , fold: 10
Srednja tacnost: 0.7218035, srednja kappa: 0.4466696

5-fold validacija
Najveca tacnost: 0.7353952 , fold: 3, najveca kappa: 0.4749016 , fold: 3
Najmanja tacnost: 0.6919105 , fold: 2, najmanja kappa: 0.3933902 , fold: 2
Srednja tacnost: 0.7166415, srednja kappa: 0.4369746
```

```
10-fold bootstrap
Najveca tacnost: 0.7896552 , fold: 3, najveca kappa: 0.5849561 , fold: 8
Najmanja tacnost: 0.7275862 , fold: 5, najmanja kappa: 0.458418 , fold: 5
Srednja tacnost: 0.7596552, srednja kappa: 0.5215676

5-fold bootstrap
Najveca tacnost: 0.7745267 , fold: 2, najveca kappa: 0.5514042 , fold: 2
Najmanja tacnost: 0.7435456 , fold: 5, najmanja kappa: 0.489699 , fold: 5
Srednja tacnost: 0.7531842, srednja kappa: 0.5089215
```

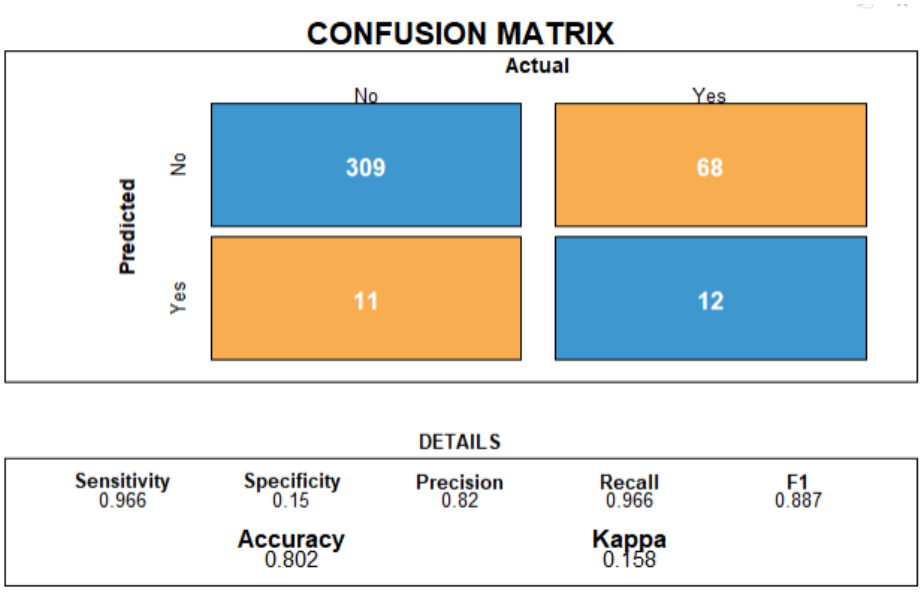
Tuning hiperparametara

Tuning hiperparametara klasifikatora KNN smo izvršili pozivom funkcije tune.knn(), pri čemu smo proslijedili listu susjeda pomoću koje će se praviti različiti modeli, te odabrati najbolji od njih. Tuning je prvo izvršen nad nebalansiranim podacima.
Vrijednost k i tačnost najboljeg modela su prikazani ispod.

```
Najbolja vrijednost k: 19
Najveca tacnost: 0.7948015
```



Vizualizacija greške klasifikacije pri tuningu parametra k

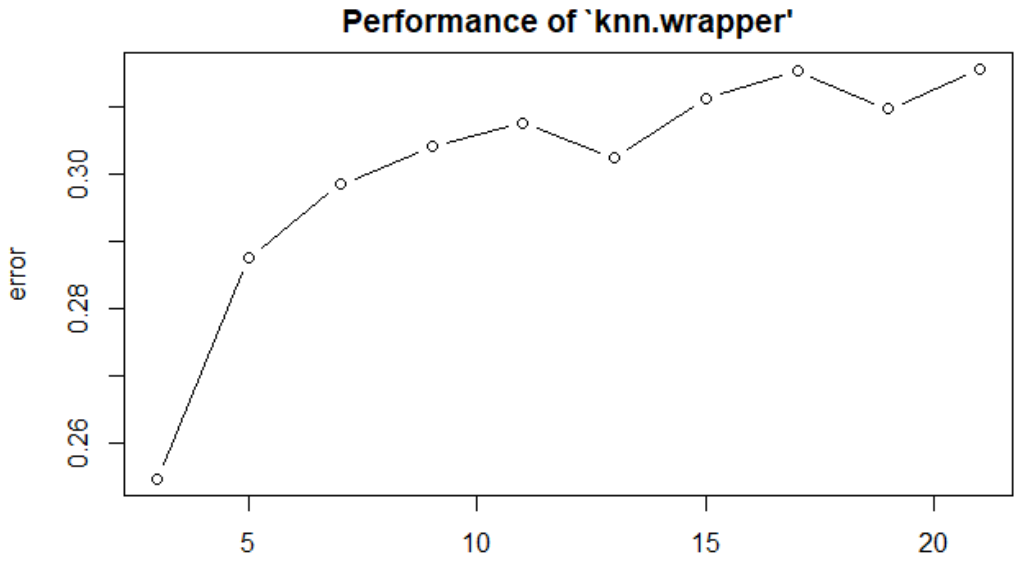


Konfuzijska matrica tuniranog modela KNN

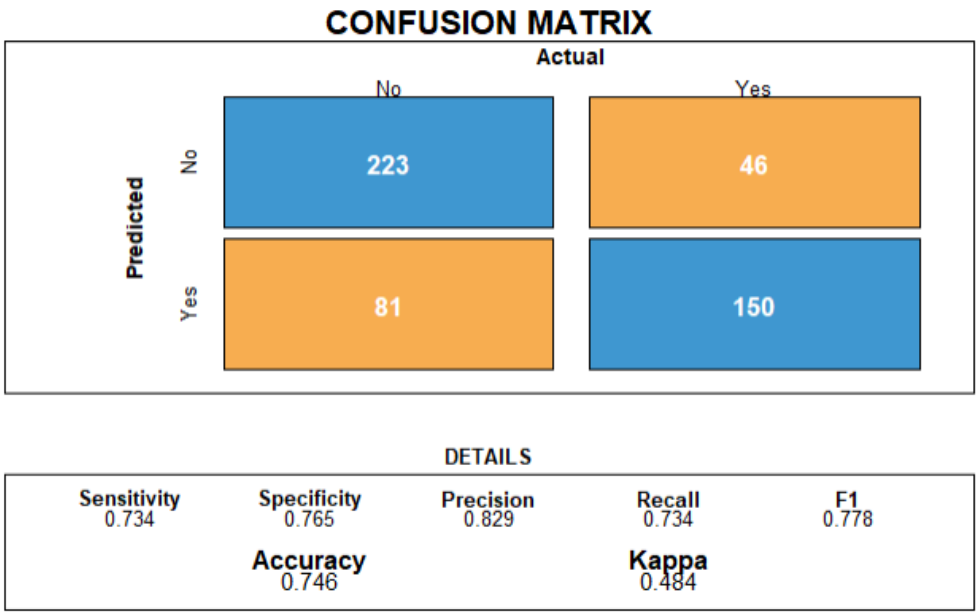
Rezultati tuninga nad oversample-anim podacima:

```
Najbolja vrijednost k: 3
Najveca tacnost: 0.7452
```

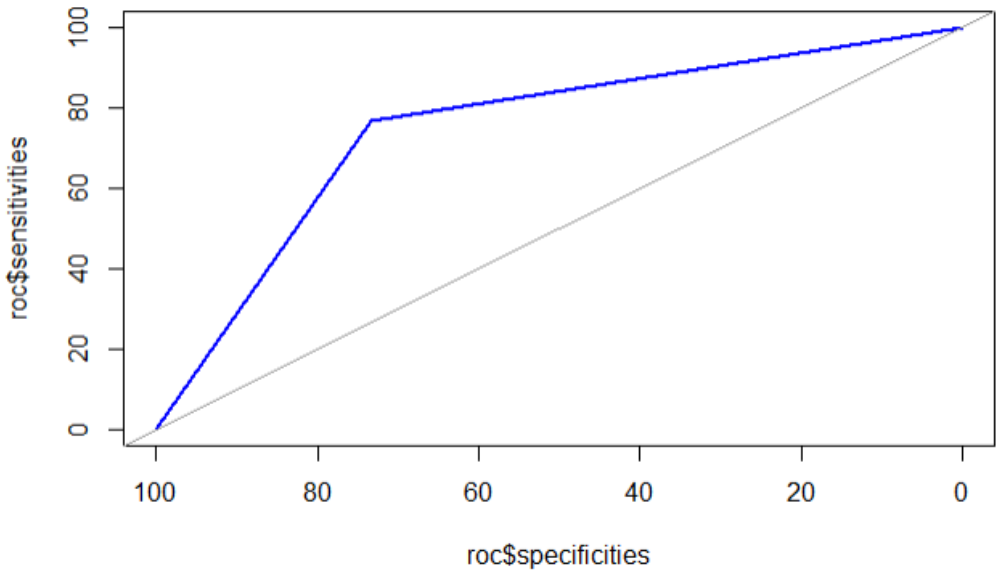
Vrijednost k i tačnost najboljeg modela



Vizualizacija greške klasifikacije pri tuningu parametra k



Konfuzijska matrica tuniranog modela KNN

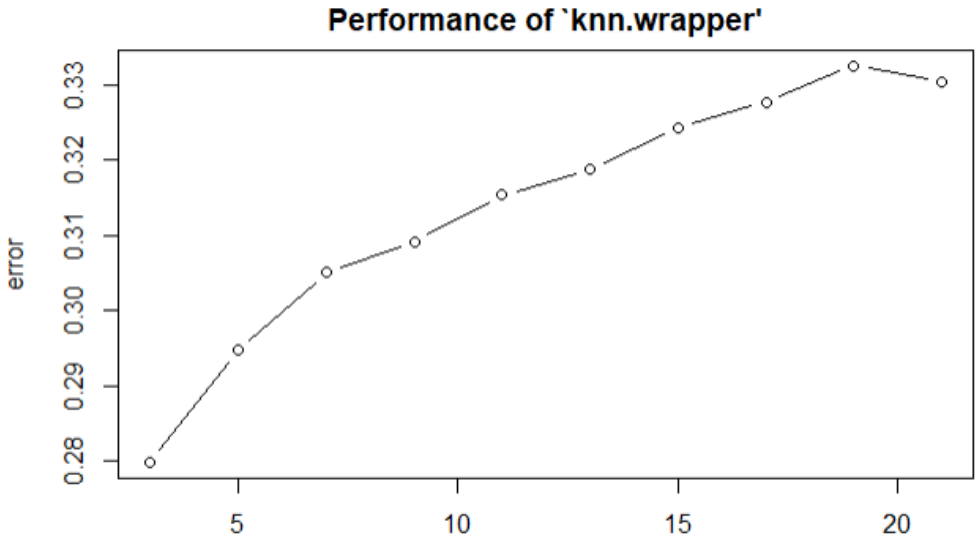


ROC kriva tuniranog modela KNN

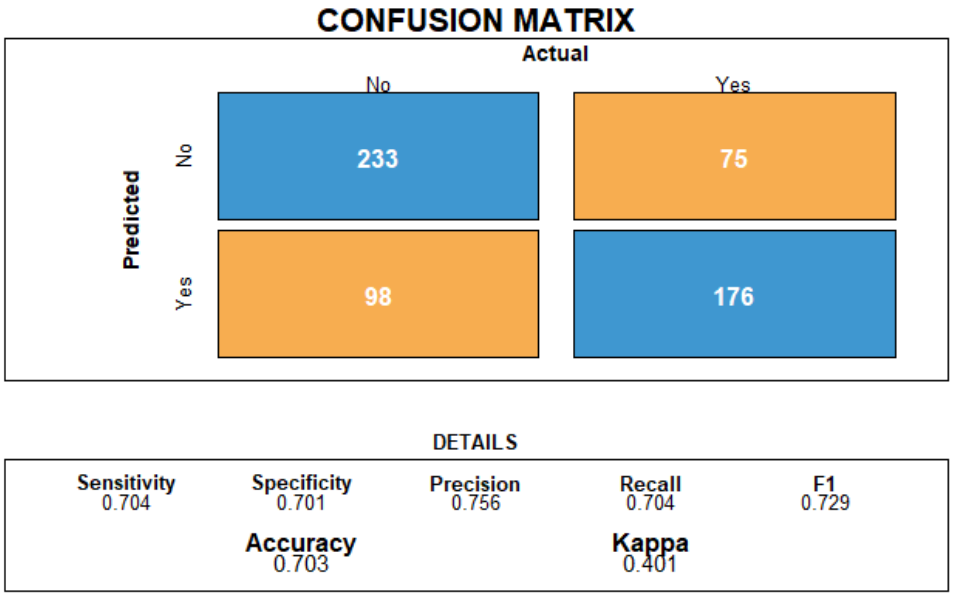
Rezultati tuninga nakon SMOTE balansiranja:

```
Najbolja vrijednost k: 3
Najveca tacnost: 0.7200984
```

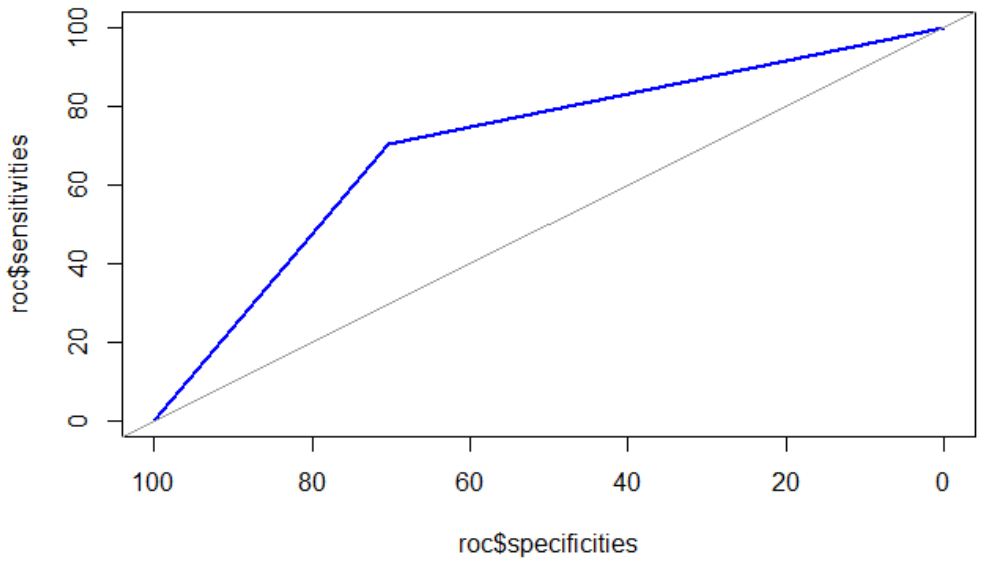
Vrijednost k i tačnost najboljeg modela



Vizualizacija greške klasifikacije pri tuningu parametra k



Konfuzijska matrica tuniranog modela KNN



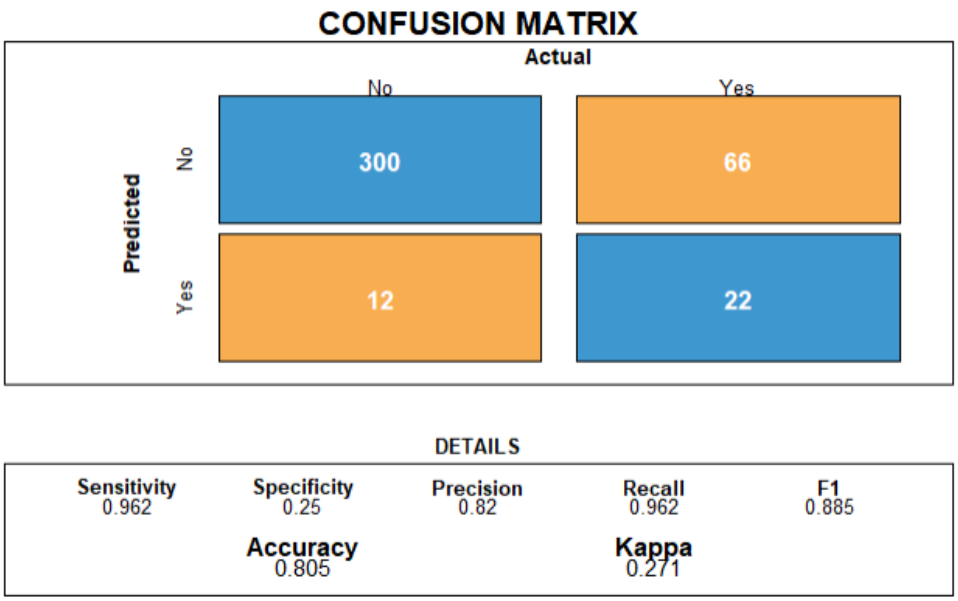
ROC kriva tuniranog modela KNN

Ensamble tehnike - *bagging*

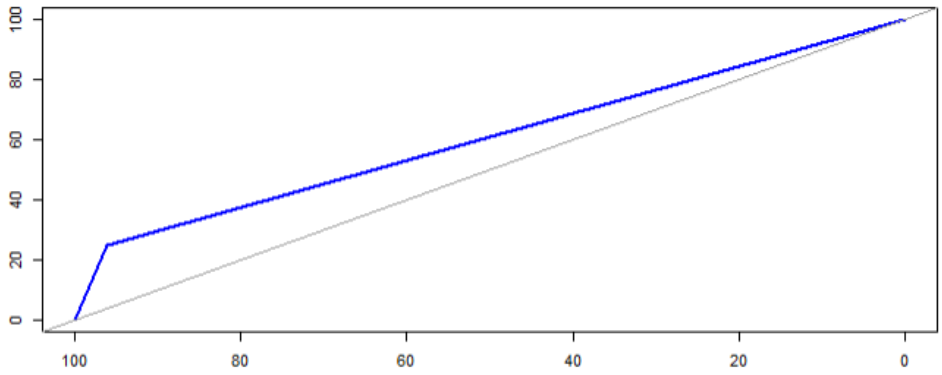
Kako bi primijenili bagging ensamble tehniku nad naše modele klasifikacije, bilo je potrebno napisati odgovarajuće pomoćne funkcije. Funkcija `predict_bagging()` prima kao prvi parametar čitav data set, zatim prima testne podatke, listu predikcija, te broj kreiranih modela. U tijelu funkcije se nalazi for petlja koja inicijalizira dva brojača “Yes” i “No” predikcija, te prolazi kroz proslijeđenu listu predikcija i vrši brojanje odgovarajućih vrijednosti. Na kraju se vrijednost predikcije sa najviše “glasova” dodaje u listu koja predstavlja povratnu vrijednost funkcije.

Druga pomoćna funkcija, koja se kreira za svaki model sa minimalnim izmjenama, je `bagging_knn()`, koja prima iste parametre kao i prethoda funkcija, osim liste predikcija, koja se zapravo kreira u tijelu funkcije, te vraća kao njen rezultat izvršavanja. Trening podaci se za svaki model kreiraju na isti način kao u pomoćnoj funkciji za bootstrapping, a testiranje se vrši nad svim testnim podacima. Unutar for petlje koja ide sve do broja modela koje je potrebno kreirati, kreiramo modele, vršimo testiranje, a rezultate testiranja spašavamo u listu predikcija.

Rezultati bagging-a osnovnog modela:

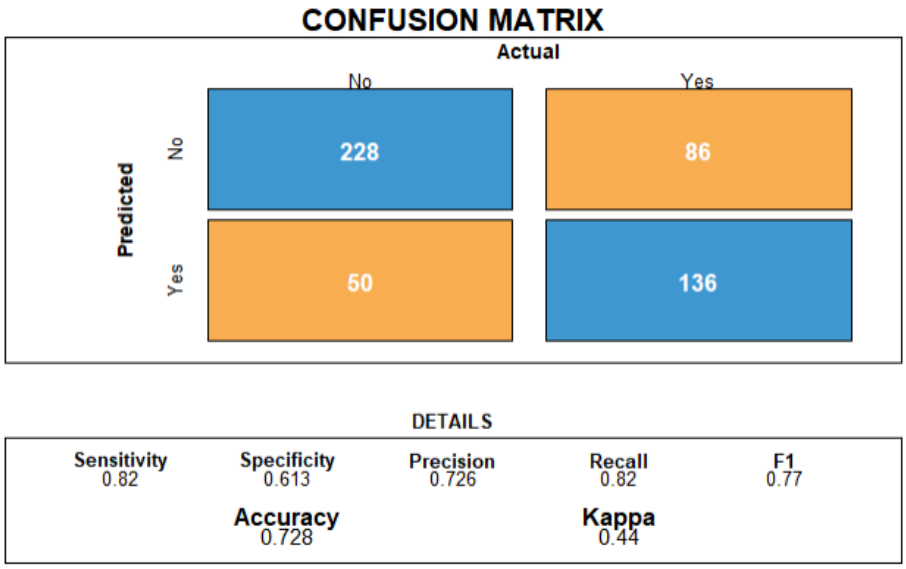


Konfuzijska matrica

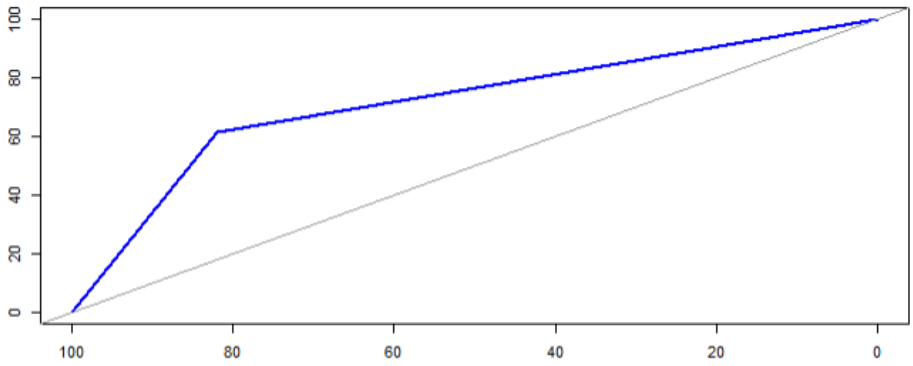


ROC kriva

Prikaz rezultata nakon oversampling-a:

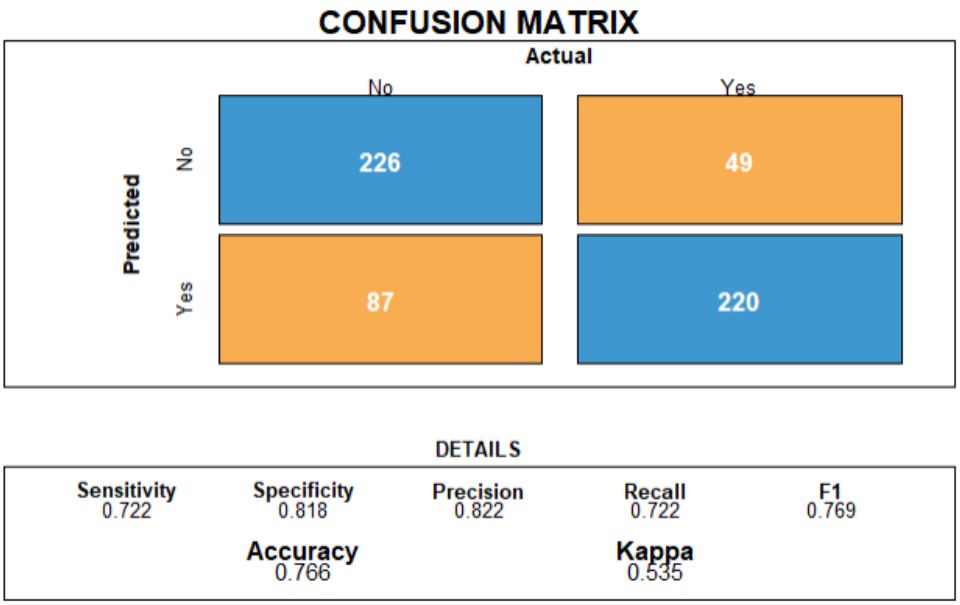


Konfuzijska matrica

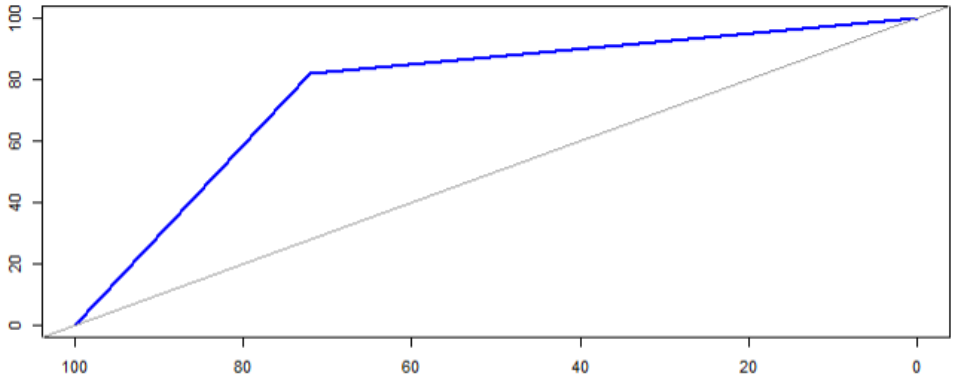


ROC kriva

Prikaz rezultata nakon SMOTE balansiranja:



Konfuzijska matrica

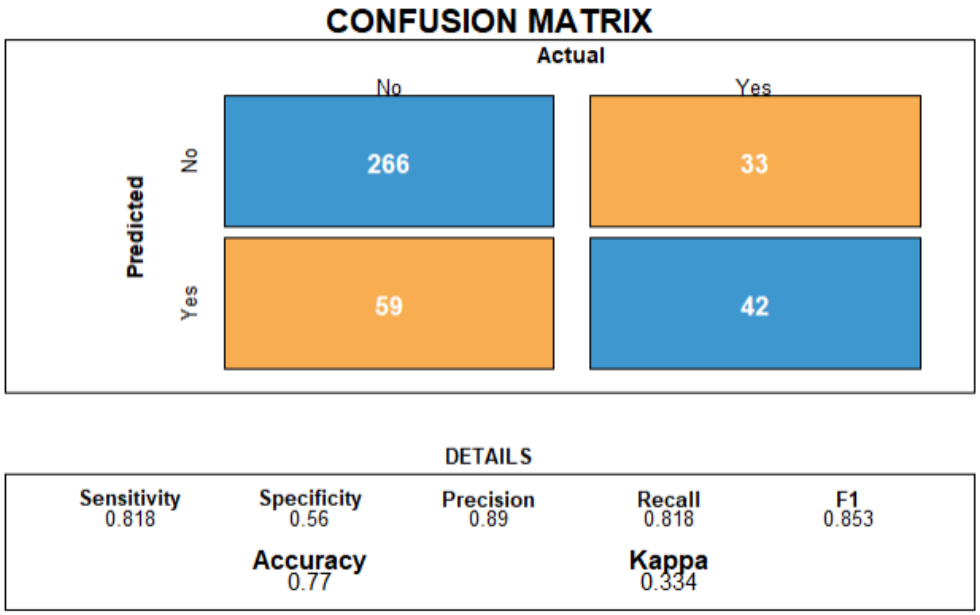


ROC kriva

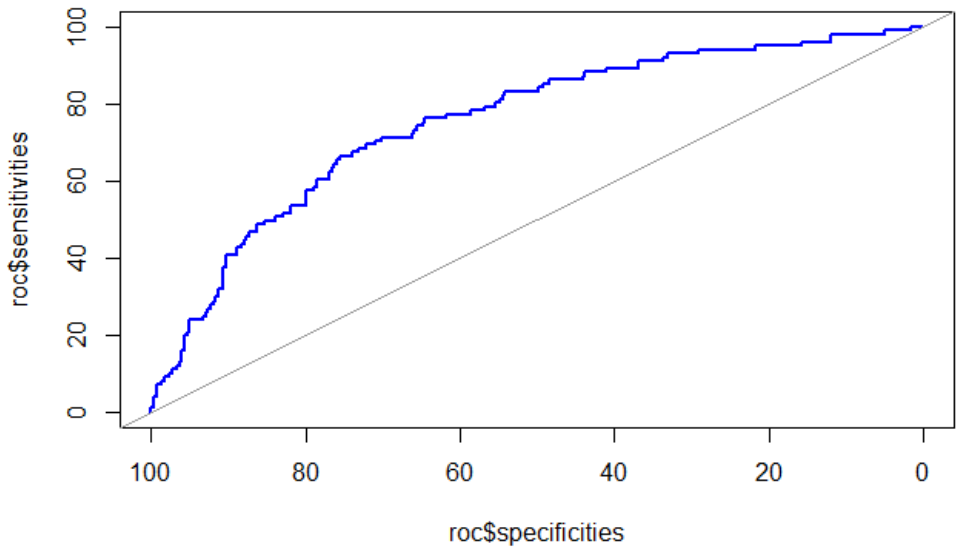
- <u>Naivni Bayesov model predikcije</u>

Pri izradi modela predikcije Naivni Bayes, nije bilo potrebno vršiti nikakvu pretvorbu ulaznih varijabli, a podjela podataka na trening i testni skup je urađena na isti način kao za model knn.

Model je izgrađen pozivom funkcije `naiveBayes()`. Prikaz evaluacije osnovnog modela je dat ispod.



Konfuzijska matrica za osnovni model



ROC kriva za osnovni model

Za cross-validaciju je korištena pomoćna funkcija `kfold_bayes()`, koja prima dva parametra: podaci i k (prilagođena funkcija sa zadaće 1). Prikazane su vrijednosti 10-fold i 5-fold validacije.

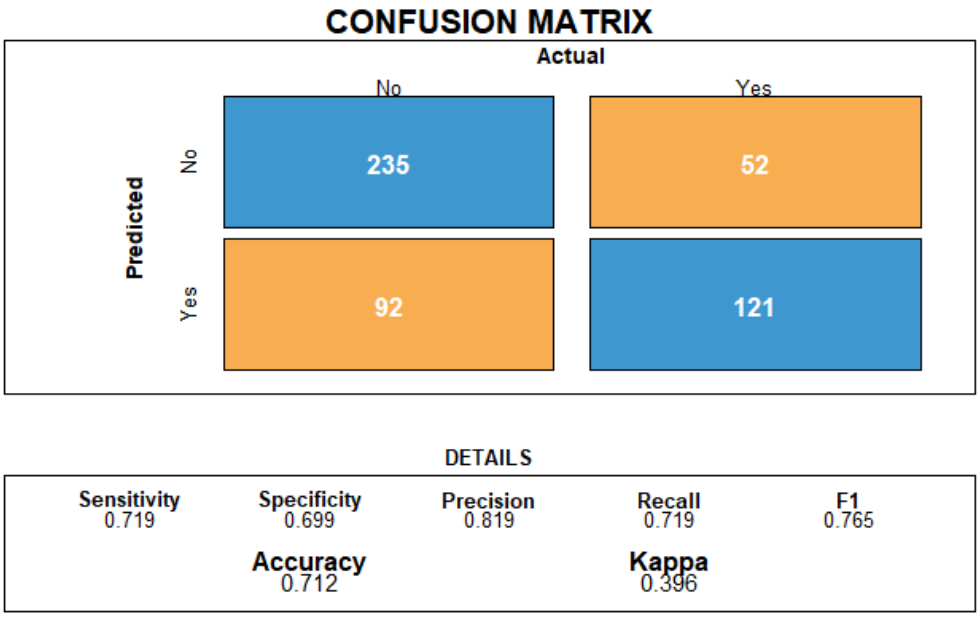
```
10-fold validacija
Najveca tacnost: 0.84 , fold: 6, najveca kappa: 0.527745 , fold: 6
Najmanja tacnost: 0.745 , fold: 9, najmanja kappa: 0.2366412 , fold: 3
Srednja tacnost: 0.7802965, srednja kappa: 0.3409524

5-fold validacija
Najveca tacnost: 0.8225 , fold: 3, najveca kappa: 0.4450957 , fold: 3
Najmanja tacnost: 0.7475 , fold: 5, najmanja kappa: 0.2831798 , fold: 1
Srednja tacnost: 0.7757794, srednja kappa: 0.3305658
```

Za bootstrap je korištena pomoćna funkcija `bootstrap_bayes()`, koja prima dva parametra: podaci i k (prilagođena funkcija sa zadaće 1). Prikazane su vrijednosti 10-fold i 5-fold bootstrapinga.

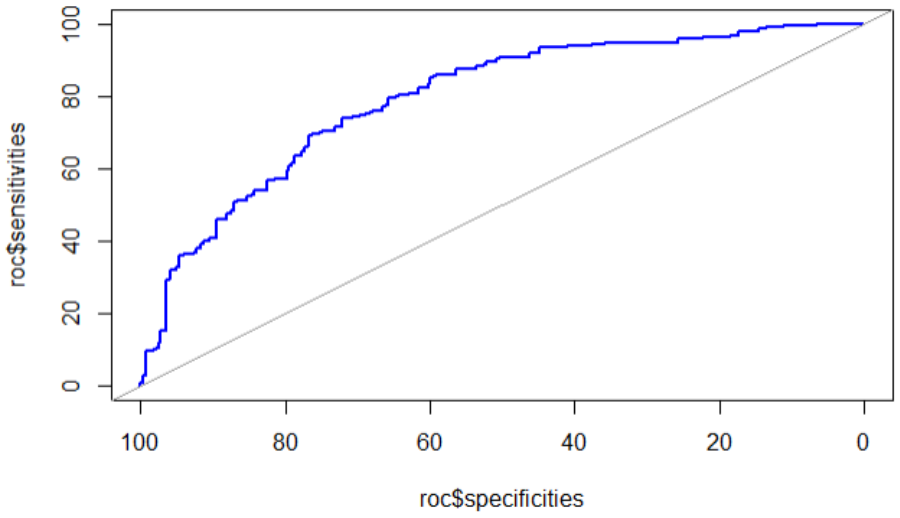
```
10-fold bootstrap
Najveca tacnost: 0.7135678 , fold: 1, najveca kappa: 0.295772 , fold: 1
Najmanja tacnost: 0.7135678 , fold: 1, najmanja kappa: 0.295772 , fold: 1
Srednja tacnost: 0.7135678, srednja kappa: 0.295772

5-fold bootstrap
Najveca tacnost: 0.764411 , fold: 1, najveca kappa: 0.3470177 , fold: 1
Najmanja tacnost: 0.764411 , fold: 1, najmanja kappa: 0.3470177 , fold: 1
Srednja tacnost: 0.764411, srednja kappa: 0.3470177
```

Balansiranje podataka (oversampling)

Balansiranje podataka je izvršeno na isti način kao i za model knn. Evaluacija modela nakon balansiranja je data ispod:



Konfuzijska matrica za model nakon balansiranja



ROC kriva nakon balansiranja

Cross-validacija i bootstrap su izvršeni na isti način kako je opisano ranije, a rezultati su prikazani na slikama ispod:

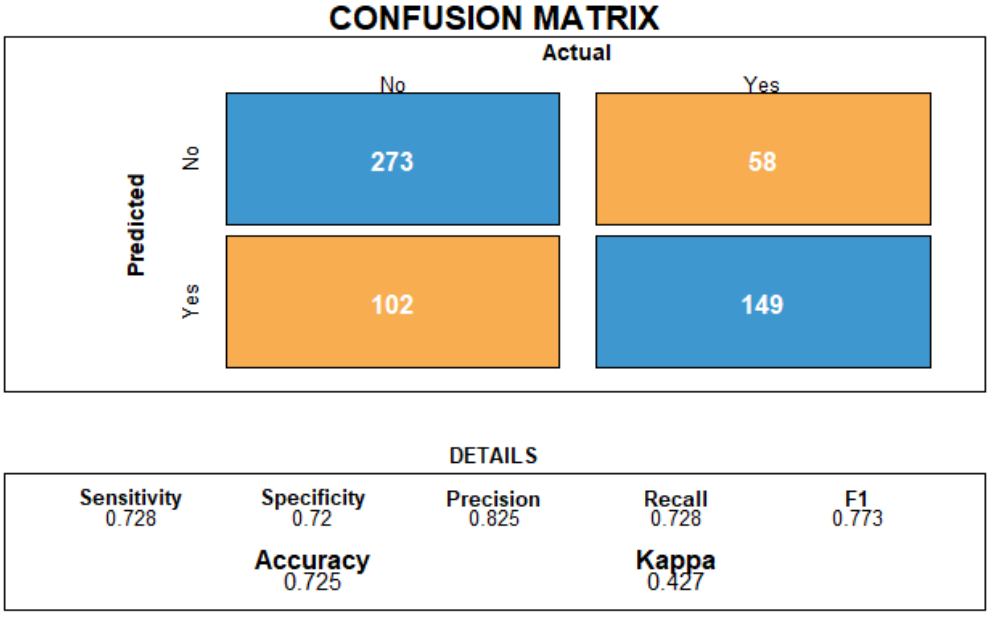
```
10-fold validacija
Najveca tacnost: 0.748 , fold: 9, najveca kappa: 0.4533907 , fold: 9
Najmanja tacnost: 0.684 , fold: 5, najmanja kappa: 0.3023667 , fold: 5
Srednja tacnost: 0.7128, srednja kappa: 0.3832191

5-fold validacija
Najveca tacnost: 0.718 , fold: 2, najveca kappa: 0.3934962 , fold: 5
Najmanja tacnost: 0.686 , fold: 3, najmanja kappa: 0.3158445 , fold: 3
Srednja tacnost: 0.706, srednja kappa: 0.3709363


10-fold bootstrap
Najveca tacnost: 0.72 , fold: 1, najveca kappa: 0.3872978 , fold: 1
Najmanja tacnost: 0.72 , fold: 1, najmanja kappa: 0.3872978 , fold: 1
Srednja tacnost: 0.72, srednja kappa: 0.3872978

5-fold bootstrap
Najveca tacnost: 0.718 , fold: 1, najveca kappa: 0.4001225 , fold: 1
Najmanja tacnost: 0.718 , fold: 1, najmanja kappa: 0.4001225 , fold: 1
Srednja tacnost: 0.718, srednja kappa: 0.4001225
```

Analiziranjem dobivenih rezultata nakon balansiranja podataka, možemo zaključiti da su se poboljšale vrijednosti metrika kappa i specifičnosti, dok su se vrijednosti ostalih metrika neznatno smanjile.

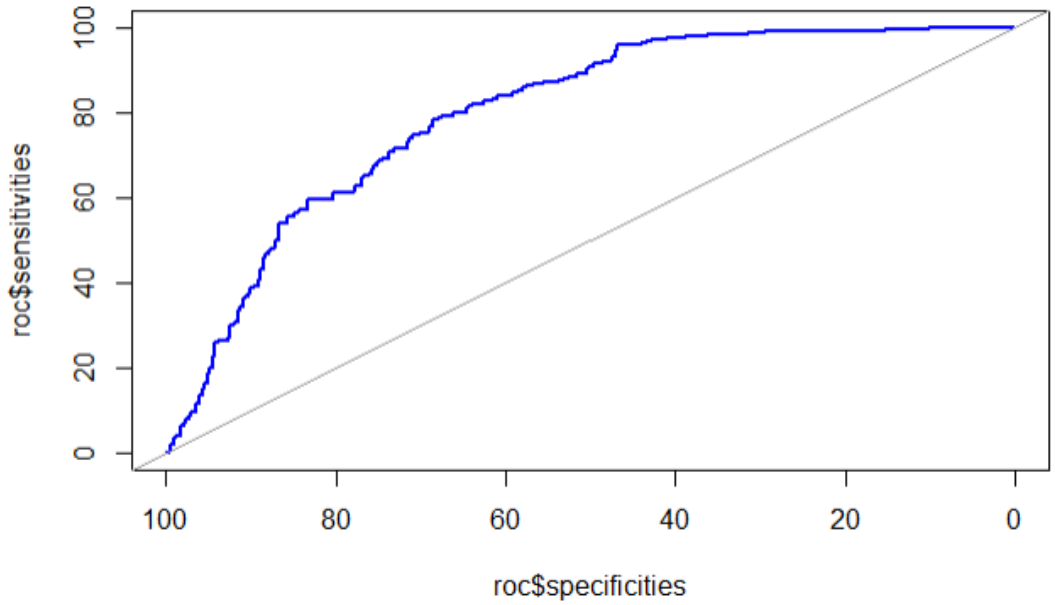
Balansiranje podataka (SMOTE)

Balansiranje podataka SMOTE algoritmom je urađeno na isti način kao i za knn model.

Evaluacija modela nakon SMOTE balansiranja je prikazana na slikama ispod:



Prikaz konfuzijske matrice nakon SMOTE-a



Prikaz ROC krive nakon SMOTE-a

Rezultati 10-fold i 5-fold cross-validacije i 10-fold i 5-fold bootstrapinga su dati ispod:

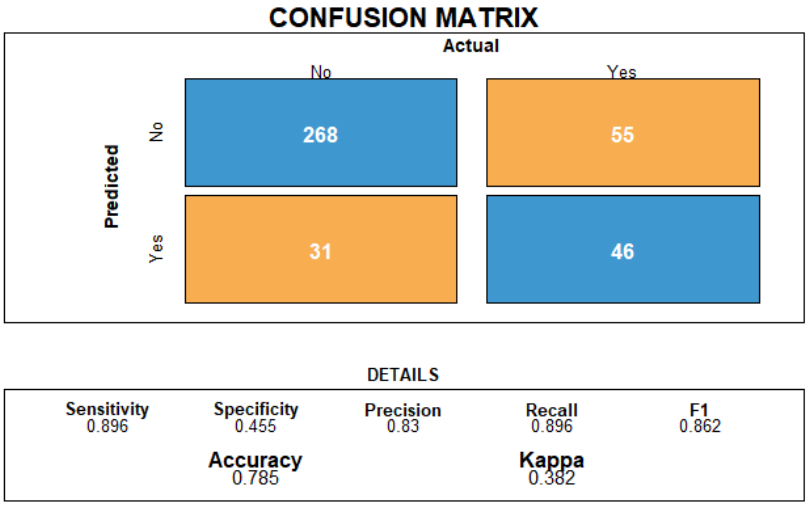
```
10-fold validacija
Najveca tacnost: 0.7388316 , fold: 8, najveca kappa: 0.4307336 , fold: 8
Najmanja tacnost: 0.6494845 , fold: 2, najmanja kappa: 0.2943945 , fold: 2
Srednja tacnost: 0.6980744, srednja kappa: 0.3733013

5-fold validacija
Najveca tacnost: 0.7319588 , fold: 1, najveca kappa: 0.4190799 , fold: 1
Najmanja tacnost: 0.6786942 , fold: 5, najmanja kappa: 0.3364671 , fold: 5
Srednja tacnost: 0.695664, srednja kappa: 0.3690007


10-fold bootstrap
Najveca tacnost: 0.7 , fold: 1, najveca kappa: 0.3495076 , fold: 1
Najmanja tacnost: 0.7 , fold: 1, najmanja kappa: 0.3495076 , fold: 1
Srednja tacnost: 0.7, srednja kappa: 0.3495076

5-fold bootstrap
Najveca tacnost: 0.6987952 , fold: 1, najveca kappa: 0.3653365 , fold: 1
Najmanja tacnost: 0.6987952 , fold: 1, najmanja kappa: 0.3653365 , fold: 1
Srednja tacnost: 0.6987952, srednja kappa: 0.3653365
```

Analiziranjem dobivenih rezultata nakon balansiranja podataka SMOTE metodom, možemo zaključiti da su se ponovo značajnije poboljšale vrijednosti metrika kappa i specifičnosti (više

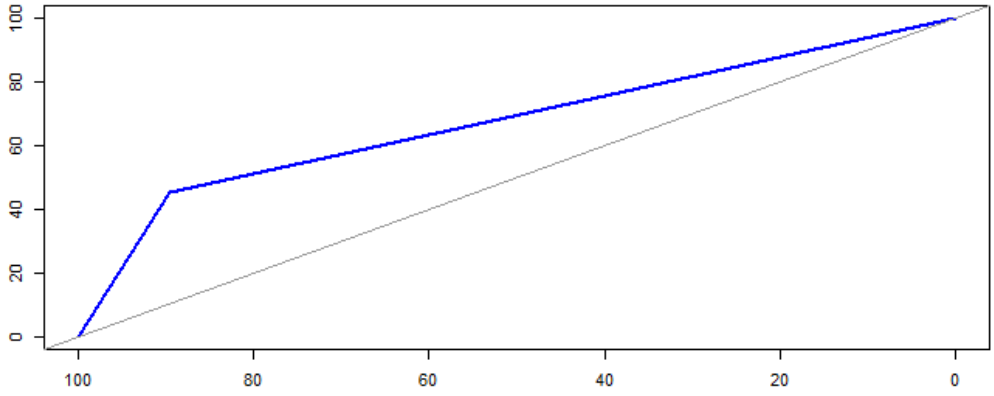
nego pri balansiranju podataka oversamplingom), dok su se vrijednosti ostalih metrika ponovo neznatno smanjile.

Ensamble tehnike - *bagging*

Kako bi primijenili bagging ensamble tehniku na model predikcije Naivni Bayes, korištene su ranije opisane funkcije `bagging_knn()` i `predict_bagging()`. Rezultati bagging-a osnovnog modela:

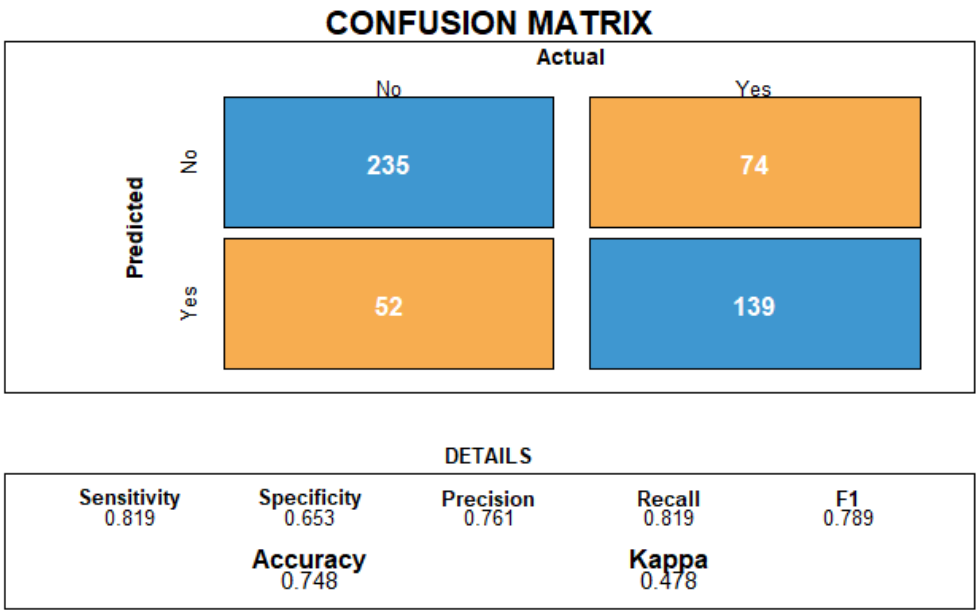


Konfuzijska matrica

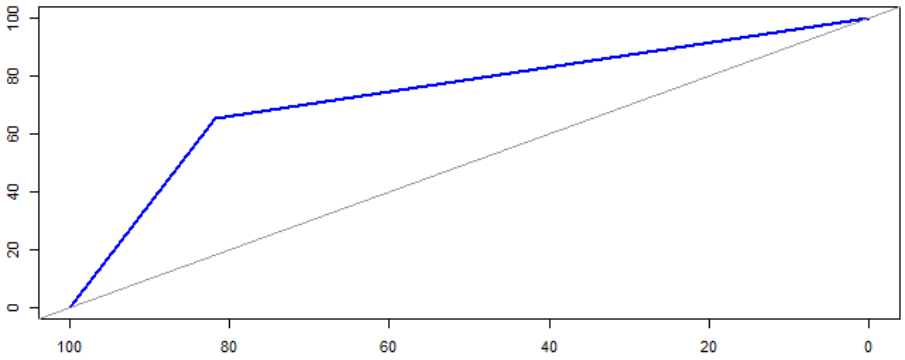


ROC kriva

Prikaz rezultata nakon oversampling-a:

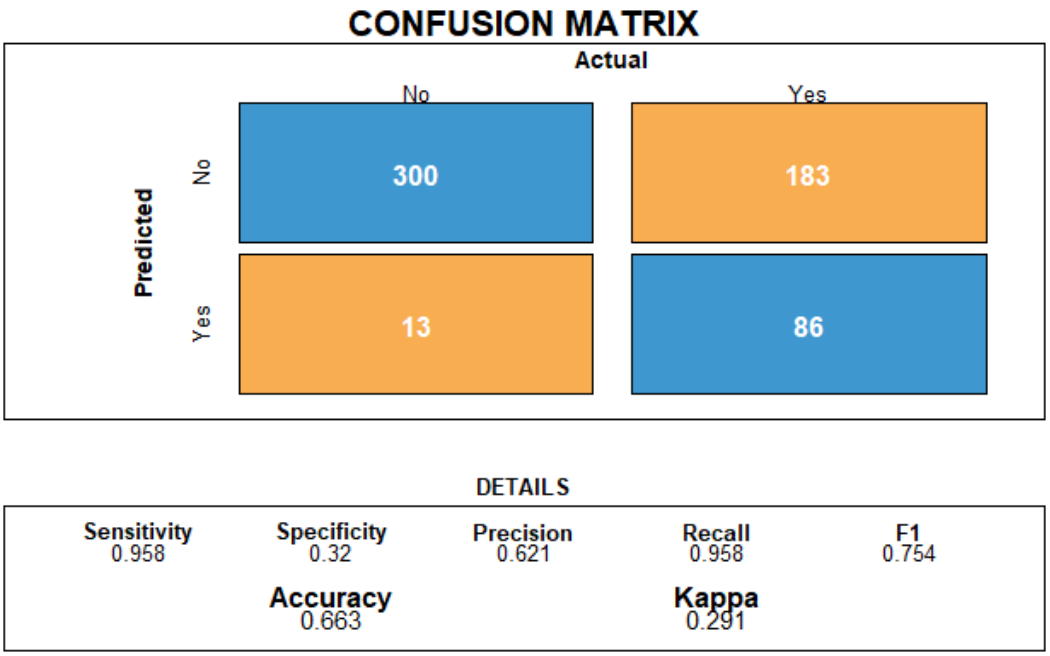


Konfuzijska matrica



ROC kriva

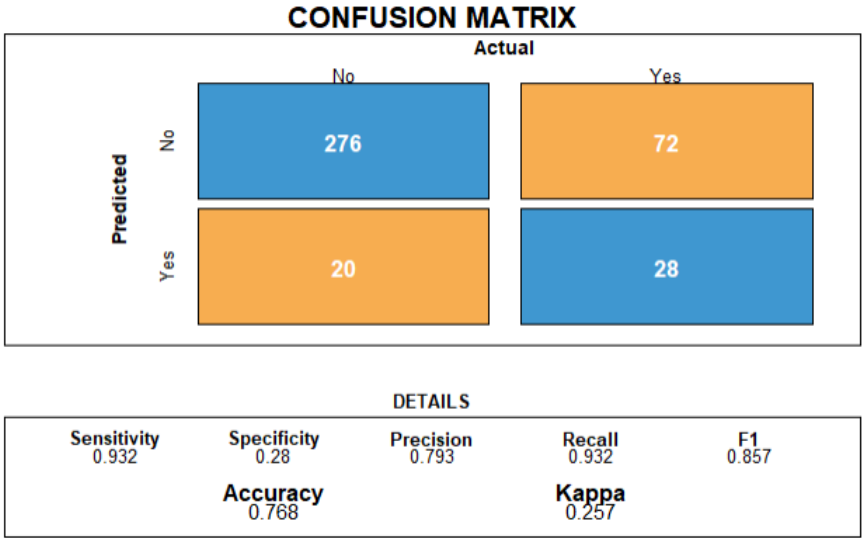
Prikaz rezultata nakon SMOTE balansiranja:



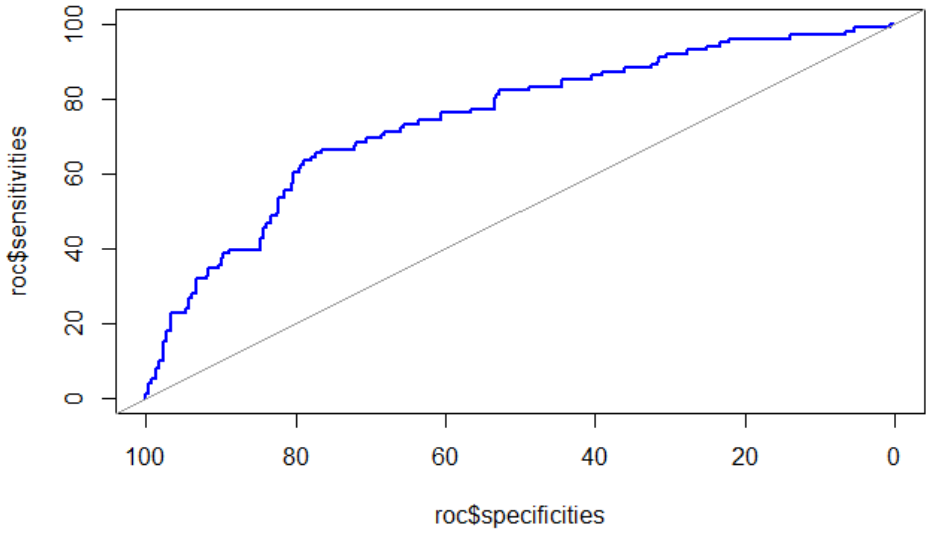
Konfuzijska matrica

- <u>model logističke regresije</u>

Model je izgrađen pozivom funkcije `glm()`. Prikaz evaluacije osnovnog modela je dat ispod.



Konfuzijska matrica za osnovni model



ROC kriva za osnovni model

Za cross-validaciju je korištena pomoćna funkcija `kfold_logistic()`, koja prima dva parametra: podaci i k (prilagođena funkcija sa zadaće 1). Prikazane su vrijednosti 10-fold i 5-fold validacije.

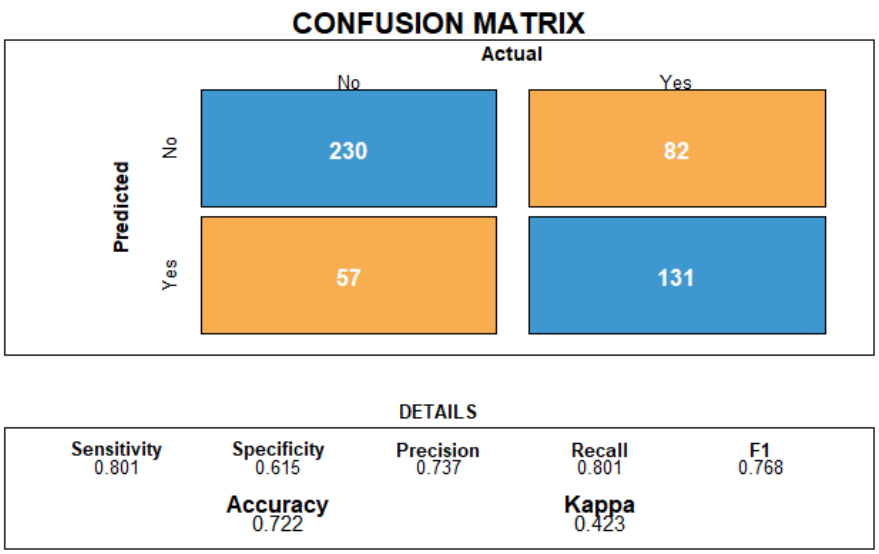
```
10-fold validacija
Najveca tacnost: 0.8291457 , fold: 4, najveca kappa: 0.4134432 , fold: 6
Najmanja tacnost: 0.705 , fold: 9, najmanja kappa: 0.1520552 , fold: 9
Srednja tacnost: 0.7903166, srednja kappa: 0.2936877

5-fold validacija
Najveca tacnost: 0.7975 , fold: 1, najveca kappa: 0.3627065 , fold: 1
Najmanja tacnost: 0.7669173 , fold: 4, najmanja kappa: 0.2118142 , fold: 4
Srednja tacnost: 0.787782, srednja kappa: 0.284977
```

Za bootstrap je korištena pomoćna funkcija `bootstrap_logistic()`, koja prima dva parametra: podaci i k (prilagođena funkcija sa zadaće 1). Prikazane su vrijednosti 10-fold i 5-fold bootstrapinga.

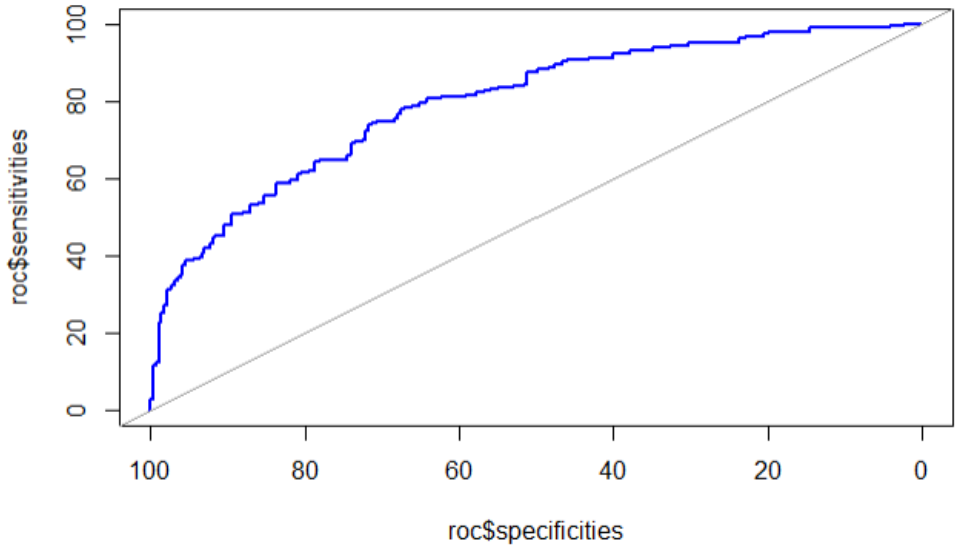
```
10-fold bootstrap
Najveca tacnost: 0.8241206 , fold: 6, najveca kappa: 0.3743174 , fold: 9
Najmanja tacnost: 0.7638191 , fold: 8, najmanja kappa: 0.1915464 , fold: 8
Srednja tacnost: 0.7914573, srednja kappa: 0.3140277

5-fold bootstrap
Najveca tacnost: 0.839599 , fold: 1, najveca kappa: 0.4143388 , fold: 1
Najmanja tacnost: 0.7794486 , fold: 2, najmanja kappa: 0.2956469 , fold: 2
Srednja tacnost: 0.8095238, srednja kappa: 0.3675381
```

Balansiranje podataka (oversampling)

Balansiranje podataka je izvršeno na isti način kao i za prethodne modele. Evaluacija modela nakon balansiranja je data ispod:



Prikaz konfuzijske matrice nakon balansiranja



Prikaz ROC krive nakon balansiranja

Cross-validacija i bootstrap su izvršeni na isti način kako je opisano ranije, a rezultati su prikazani na slikama ispod:

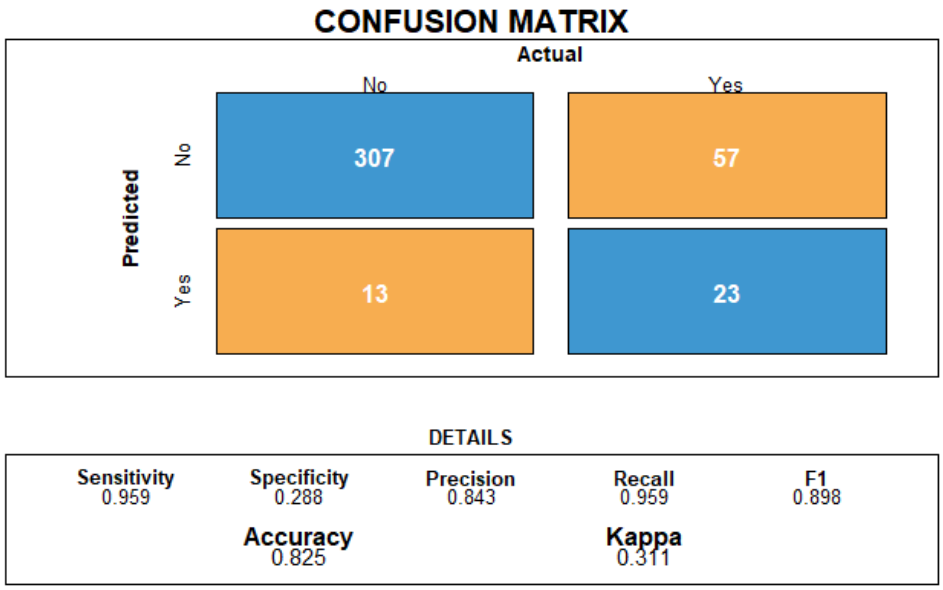
```
10-fold validacija
Najveca tacnost: 0.748 , fold: 3, najveca kappa: 0.4667787 , fold: 8
Najmanja tacnost: 0.696 , fold: 9, najmanja kappa: 0.3523316 , fold: 9
Srednja tacnost: 0.7172, srednja kappa: 0.3949665

5-fold validacija
Najveca tacnost: 0.728 , fold: 1, najveca kappa: 0.4154227 , fold: 2
Najmanja tacnost: 0.706 , fold: 4, najmanja kappa: 0.3754886 , fold: 4
Srednja tacnost: 0.7164, srednja kappa: 0.3988611
```

```
10-fold bootstrap
Najveca tacnost: 0.764 , fold: 8, najveca kappa: 0.5091514 , fold: 8
Najmanja tacnost: 0.68 , fold: 4, najmanja kappa: 0.33643 , fold: 4
Srednja tacnost: 0.7244, srednja kappa: 0.4189372

5-fold bootstrap
Najveca tacnost: 0.756 , fold: 3, najveca kappa: 0.4988086 , fold: 3
Najmanja tacnost: 0.7 , fold: 4, najmanja kappa: 0.3766104 , fold: 4
Srednja tacnost: 0.728, srednja kappa: 0.430921
```

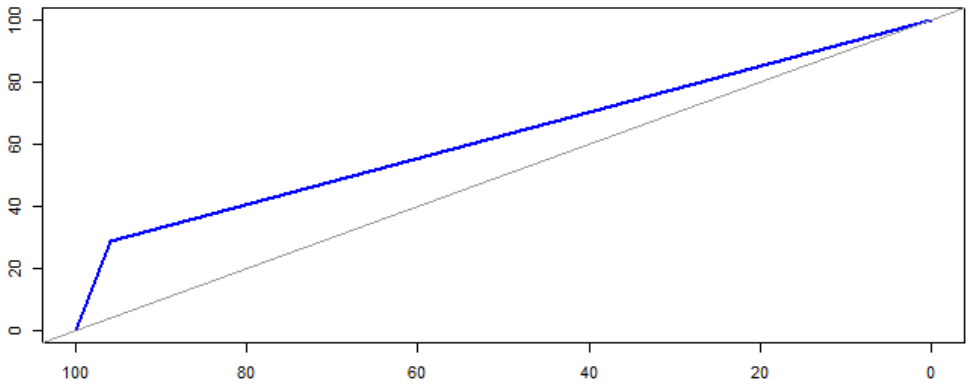
Analiziranjem dobivenih rezultata nakon balansiranja podataka, možemo zaključiti da su se poboljšale vrijednosti metrika kappa i specifičnosti, dok su se vrijednosti ostalih metrika smanjile.

Ensamble tehnike - *bagging*

Kako bi primijenili bagging ensamble tehniku na model logističke regresije, korištene su ranije opisane funkcije `bagging_logistic()` i `predict_bagging()`. Rezultati bagging-a osnovnog modela:

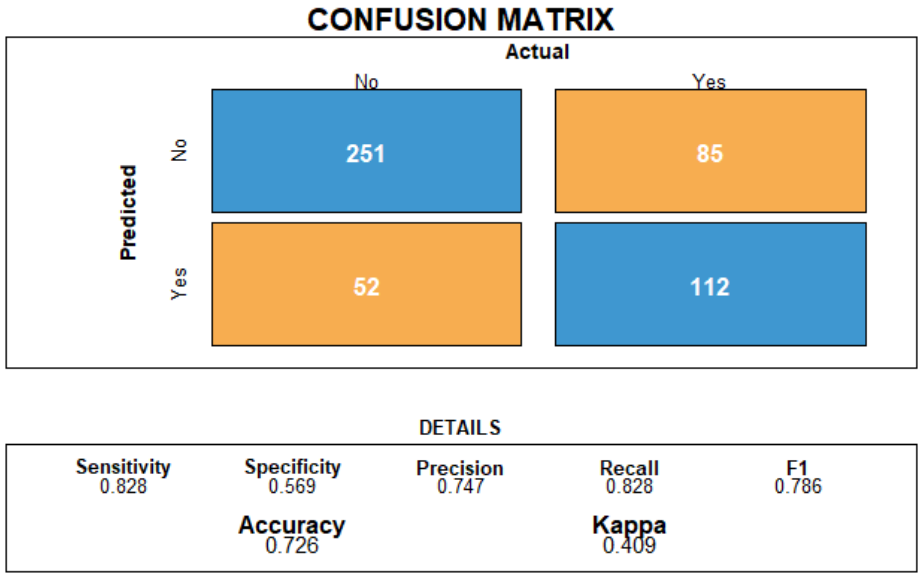


Konfuzijska matrica

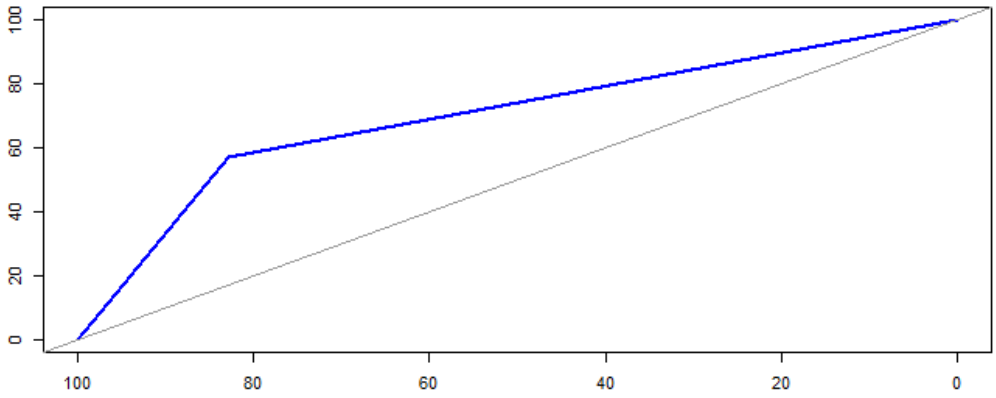


ROC kriva

Prikaz rezultata nakon oversampling-a:



Konfuzijska matrica

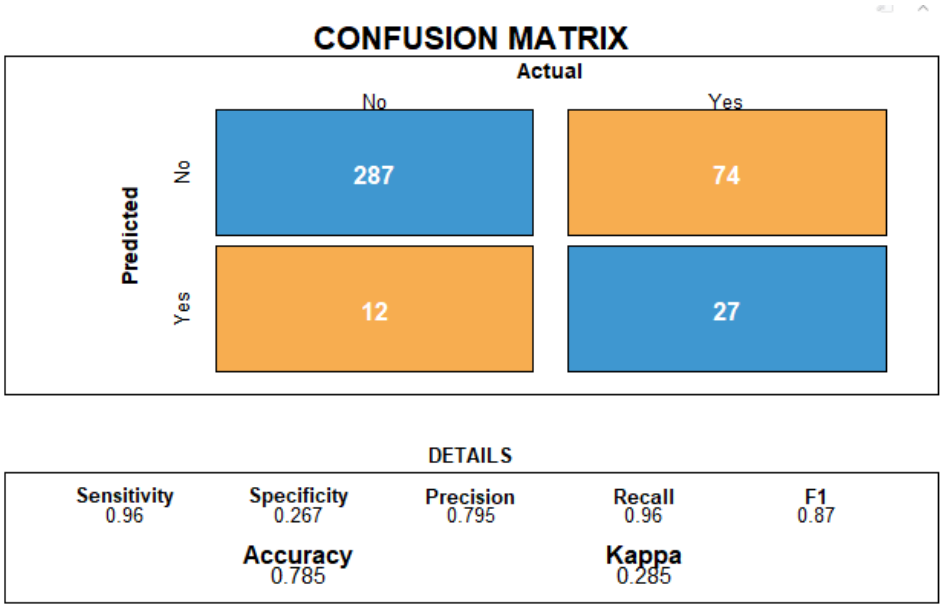


ROC kriva

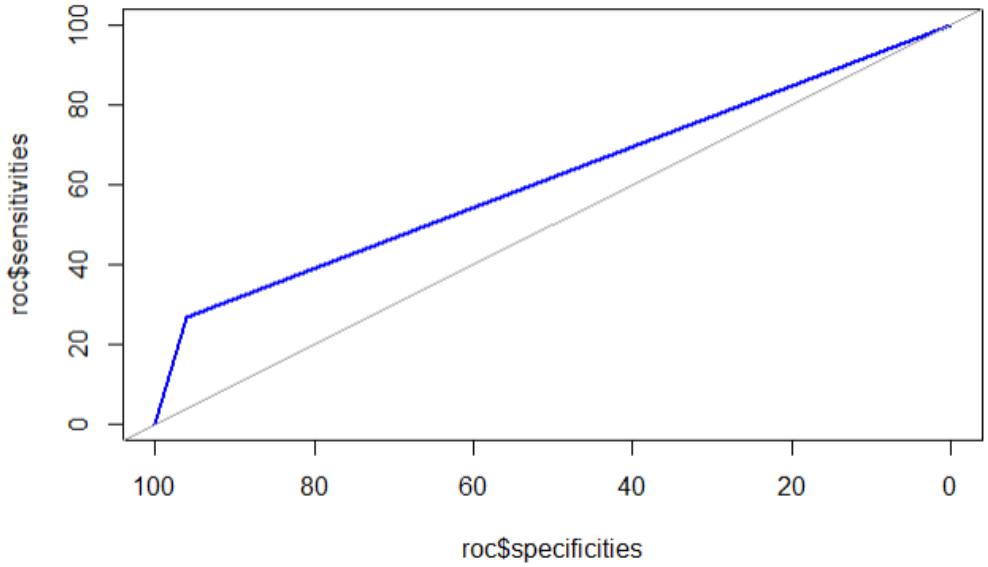
- SVM model klasifikacije

Pri izradi modela predikcije SVM, nije bilo potrebno vršiti nikakvu pretvorbu ulaznih varijabli, a podjela podataka na trening i testni skup je urađena na isti način kao za prethodne modele.

Model je izgrađen pozivom funkcije `svm()`. Prikaz evaluacije osnovnog modela je dat ispod.



Konfuzijska matrica za osnovni model



ROC kriva za osnovni model

Za cross-validaciju je korištena pomoćna funkcija `kfold_svm()`, koja prima dva parametra: podaci i k (prilagođena funkcija sa zadaće 1). Prikazane su vrijednosti 10-fold i 5-fold validacije.

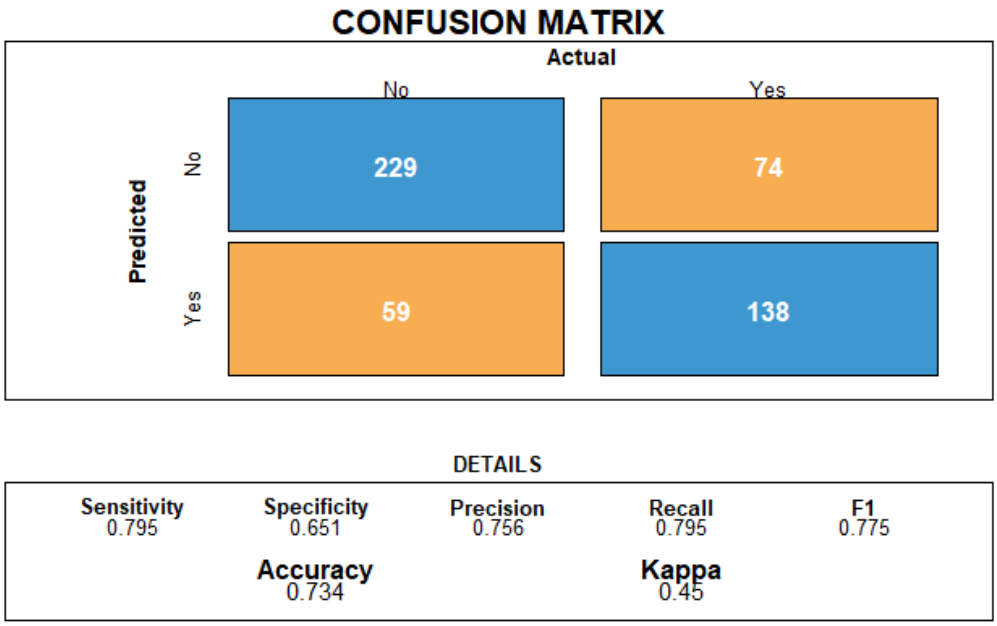
```
10-fold validacija
Najveca tacnost: 0.82 , fold: 5, najveca kappa: 0.2902208 , fold: 5
Najmanja tacnost: 0.745 , fold: 6, najmanja kappa: 0.0470852 , fold: 8
Srednja tacnost: 0.782809, srednja kappa: 0.1691844

5-fold validacija
Najveca tacnost: 0.7994987 , fold: 2, najveca kappa: 0.2206346 , fold: 3
Najmanja tacnost: 0.7725 , fold: 5, najmanja kappa: 0.1057234 , fold: 1
Srednja tacnost: 0.782287, srednja kappa: 0.1742217
```

Za bootstrap je korištena pomoćna funkcija `bootstrap_svm()`, koja prima dva parametra: podaci i k (prilagođena funkcija sa zadaće 1). Prikazane su vrijednosti 10-fold i 5-fold bootstrapinga.

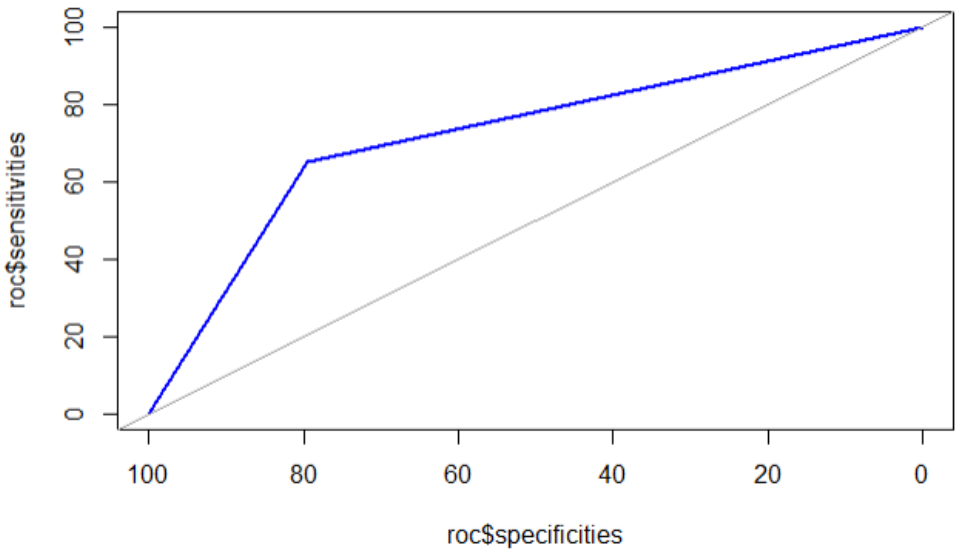
```
10-fold bootstrap
Najveca tacnost: 0.8291457 , fold: 2, najveca kappa: 0.3274354 , fold: 2
Najmanja tacnost: 0.718593 , fold: 10, najmanja kappa: 0.05855574 , fold: 1
Srednja tacnost: 0.7944724, srednja kappa: 0.2078306

5-fold bootstrap
Najveca tacnost: 0.8070175 , fold: 2, najveca kappa: 0.3019562 , fold: 2
Najmanja tacnost: 0.7944862 , fold: 3, najmanja kappa: 0.1496517 , fold: 3
Srednja tacnost: 0.802005, srednja kappa: 0.2212262
```

Balansiranje podataka (oversampling)

Balansiranje podataka je izvršeno na isti način zao i za prethodne modele. Evaluacija modela nakon balansiranja je data ispod:



Konfuzijska matrica za model nakon balansiranja



ROC kriva nakon balansiranja

Cross-validacija i bootstrap su izvršeni na isti način kako je opisano ranije, a rezultati su prikazani na slikama ispod:

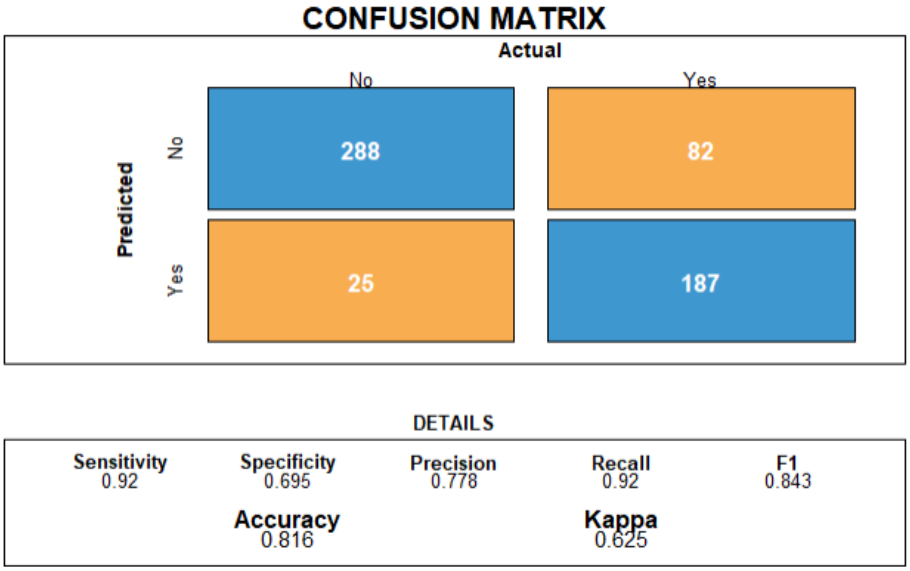
```
10-fold validacija
Najveca tacnost: 0.788 , fold: 6, najveca kappa: 0.5537218 , fold: 6
Najmanja tacnost: 0.68 , fold: 5, najmanja kappa: 0.3331555 , fold: 5
Srednja tacnost: 0.7376, srednja kappa: 0.4364569

5-fold validacija
Najveca tacnost: 0.774 , fold: 1, najveca kappa: 0.5181154 , fold: 1
Najmanja tacnost: 0.71 , fold: 4, najmanja kappa: 0.3933573 , fold: 5
Srednja tacnost: 0.738, srednja kappa: 0.4397877


10-fold bootstrap
Najveca tacnost: 0.784 , fold: 2, najveca kappa: 0.5404725 , fold: 2
Najmanja tacnost: 0.732 , fold: 6, najmanja kappa: 0.4542196 , fold: 6
Srednja tacnost: 0.7636, srednja kappa: 0.4993329

5-fold bootstrap
Najveca tacnost: 0.764 , fold: 1, najveca kappa: 0.50426 , fold: 1
Najmanja tacnost: 0.74 , fold: 3, najmanja kappa: 0.4107515 , fold: 3
Srednja tacnost: 0.7528, srednja kappa: 0.4645744
```

Analiziranjem dobivenih rezultata nakon balansiranja podataka, možemo zaključiti da su se značajnije poboljšale vrijednosti metrika kappa i specifičnosti, a vrijednosti ostalih metrika su se neznatno smanjile.

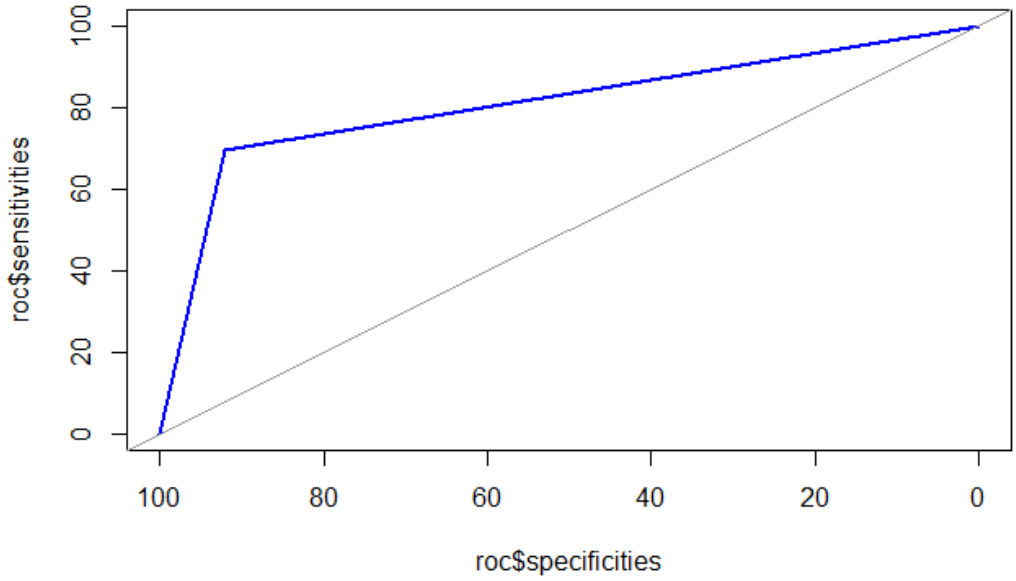
Balansiranje podataka (SMOTE)

Balansiranje podataka SMOTE algoritmom je urađeno na isti način kao i za prethodne modele.

Evaluacija modela nakon SMOTE balansiranja je prikazana na slikama ispod:



Prikaz konfuzijske matrice nakon SMOTE-a



Prikaz ROC krive nakon SMOTE-a

Rezultati 10-fold i 5-fold cross-validacije i 10-fold i 5-fold bootstrapinga su dati ispod:

```
10-fold validacija
Najveca tacnost: 0.8689655 , fold: 7, najveca kappa: 0.7316123 , fold: 7
Najmanja tacnost: 0.7972509 , fold: 6, najmanja kappa: 0.5895431 , fold: 6
Srednja tacnost: 0.8242872, srednja kappa: 0.6425381

5-fold validacija
Najveca tacnost: 0.847079 , fold: 1, najveca kappa: 0.6847468 , fold: 1
Najmanja tacnost: 0.8024055 , fold: 5, najmanja kappa: 0.6057236 , fold: 5
Srednja tacnost: 0.8277203, srednja kappa: 0.6500348
```

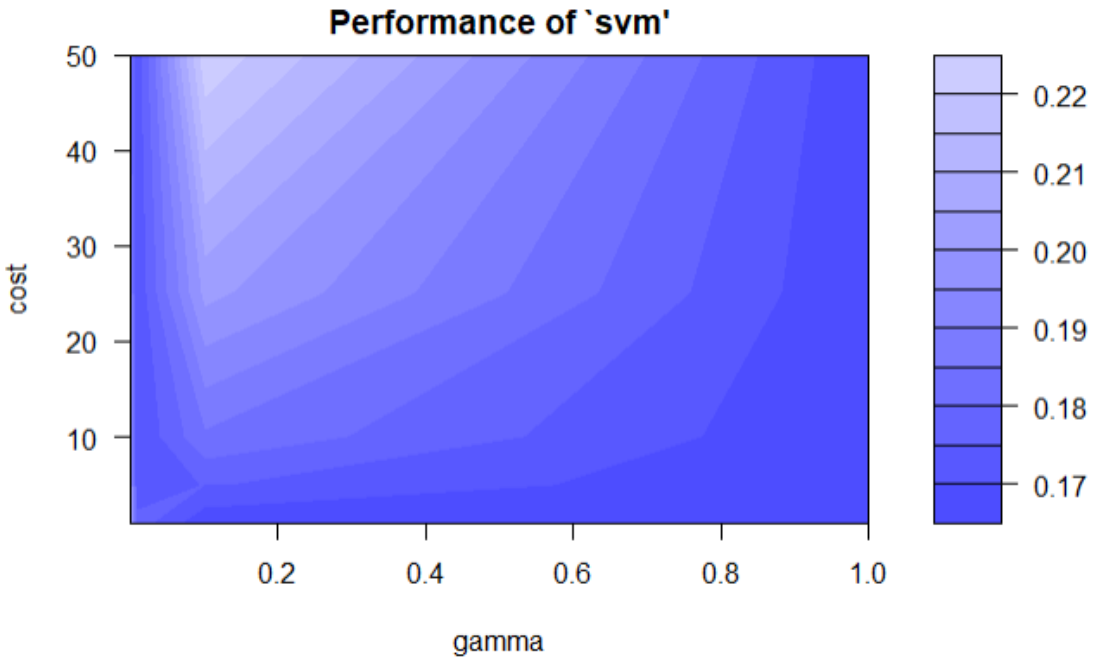
```
10-fold bootstrap
Najveca tacnost: 0.8827586 , fold: 6, najveca kappa: 0.7574058 , fold: 6
Najmanja tacnost: 0.8068966 , fold: 8, najmanja kappa: 0.6061503 , fold: 8
Srednja tacnost: 0.8586207, srednja kappa: 0.7141184

5-fold bootstrap
Najveca tacnost: 0.8846816 , fold: 3, najveca kappa: 0.7653572 , fold: 3
Najmanja tacnost: 0.8519793 , fold: 4, najmanja kappa: 0.6978387 , fold: 4
Srednja tacnost: 0.8640275, srednja kappa: 0.7235427
```

Analizom metrika možemo uočiti da je SMOTE balansiranje podataka značajno poboljšalo vrijednosti metrika kappa, specificity i accuracy.

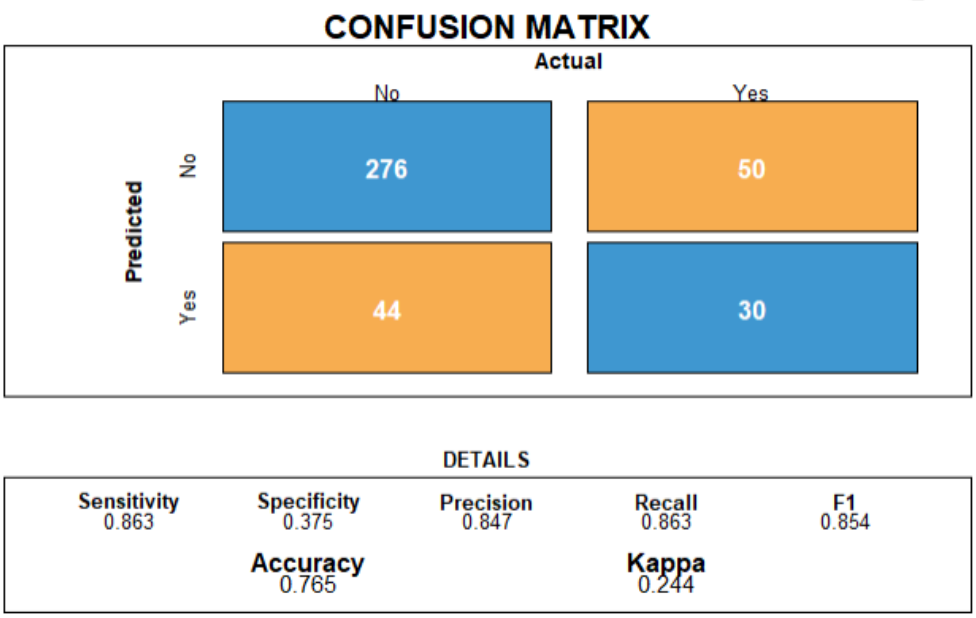
Tuning hiperparametara

Tuning hiperparametara klasifikatora KNN smo izvršili pozivom funkcije tune.svm(). Rezultati tuniranja modela prikazani su u nastavku:

```
Najbolja vrijednost cost: 5
Najbolja vrijednost gamma: 1
Najveca tacnost: 0.8347931
```



Vizualizacija rezultata tuninga hiperparametara za svm model

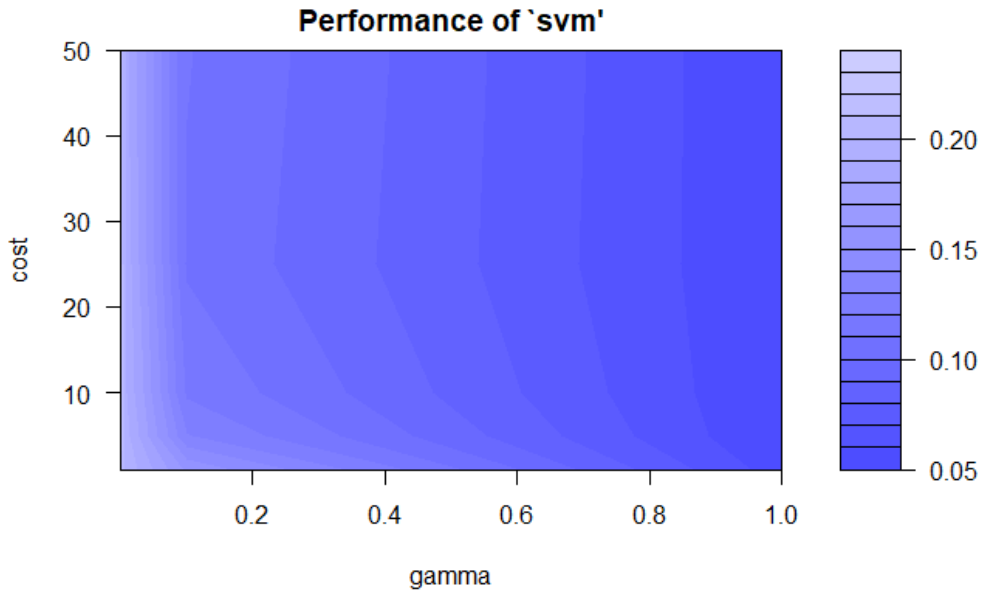


Konfuzijska matrica tuniranog modela SVM

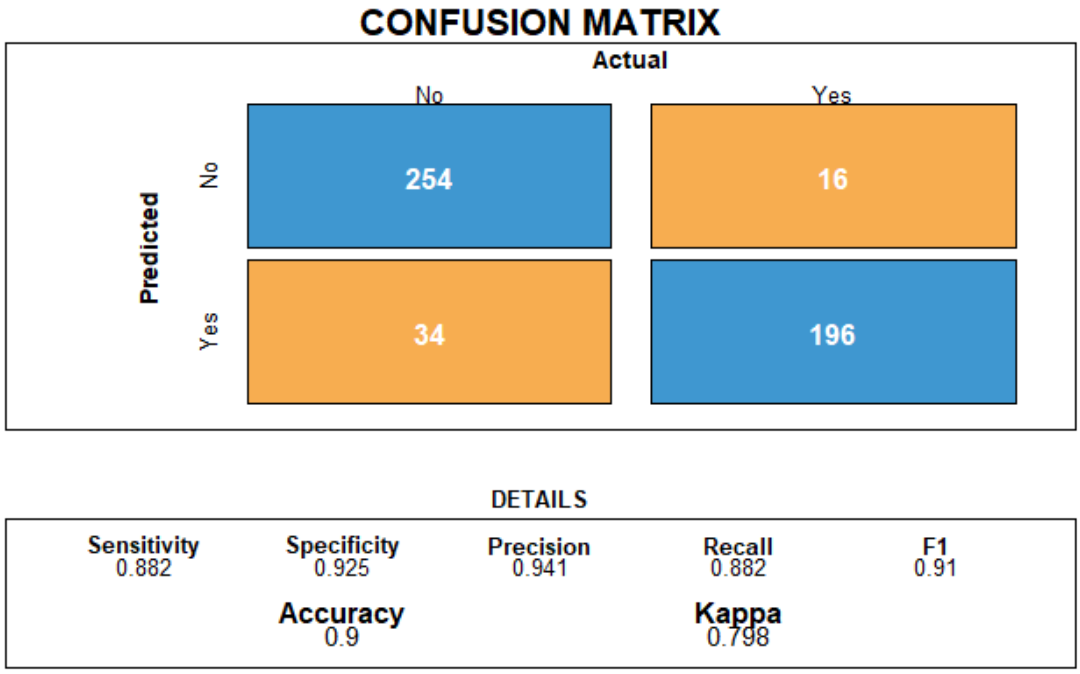
Rezultati tuninga nad oversample-anim podacima:

```
Najbolja vrijednost cost: 10
Najbolja vrijednost gamma: 1
Najveca tacnost: 0.9498837
```

Vrijednost hiperparametara cost i gamma najboljeg svm modela



Vizualizacija rezultata tuninga hiperparametara za svm model (oversamplani podaci)

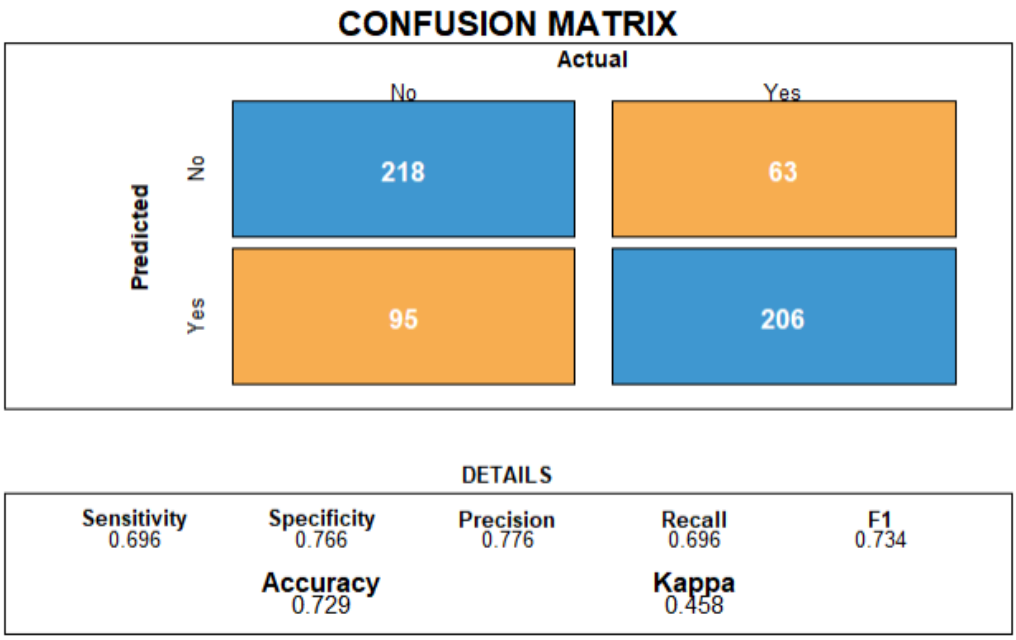


Konfuzijska matrica tuniranog modela SVM (oversamplani podaci)

Rezultati tuninga nakon SMOTE balansiranja:

```
Najbolja vrijednost cost: 5
Najbolja vrijednost gamma: 1
Najveca tacnost: 0.9105866
```

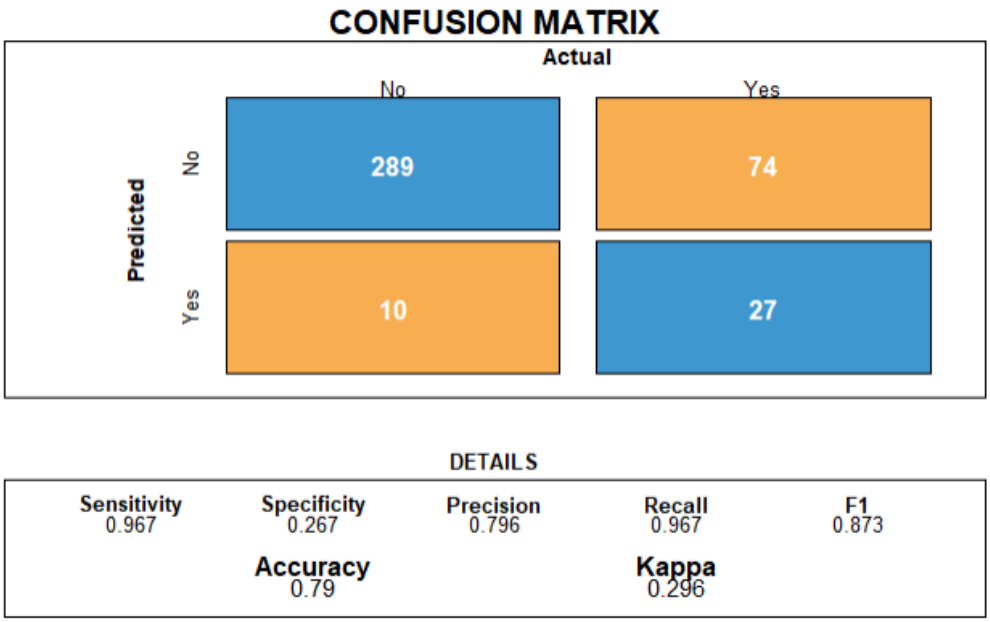
Vrijednost hiperparametara cost i gamma najboljeg svm modela



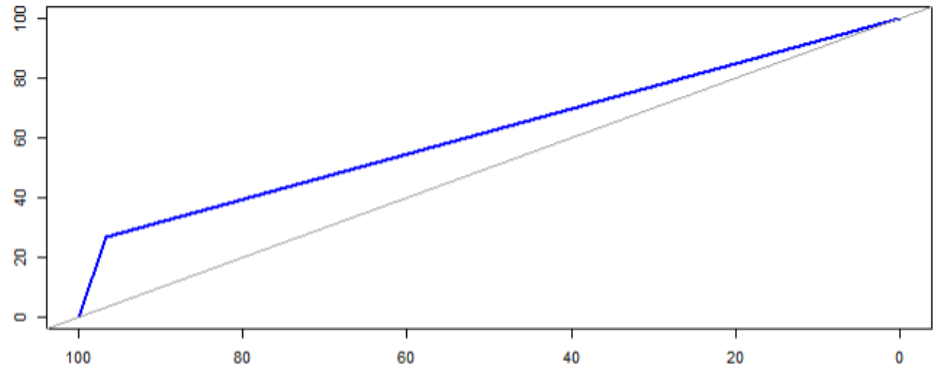
Konfuzijska matrica tuniranog modela SVM (SMOTE balansiranje)

Ensamble tehnike - *bagging*

Kako bi primijenili bagging ensamble tehniku na model predikcije Naivni Bayes, korištene su ranije opisane funkcije `bagging_svm()` i `predict_bagging()`. Rezultati bagging-a osnovnog modela:

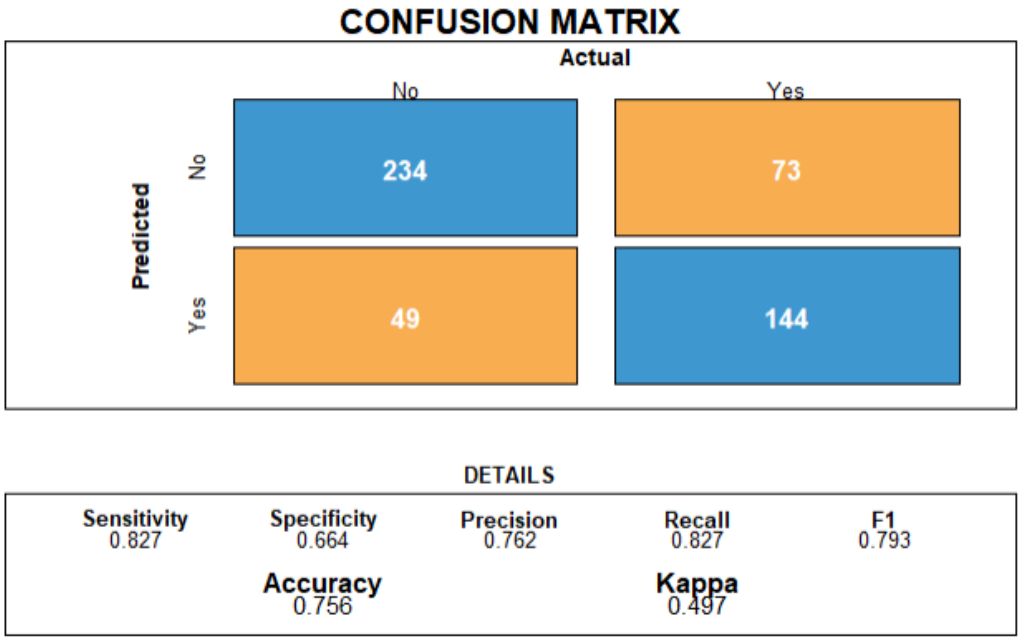


Konfuzijska matrica

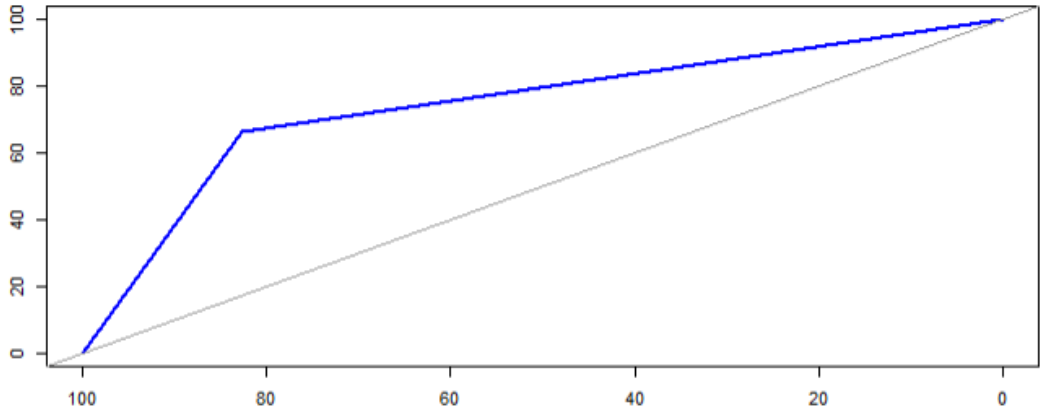


ROC kriva

Prikaz rezultata nakon oversampling-a:

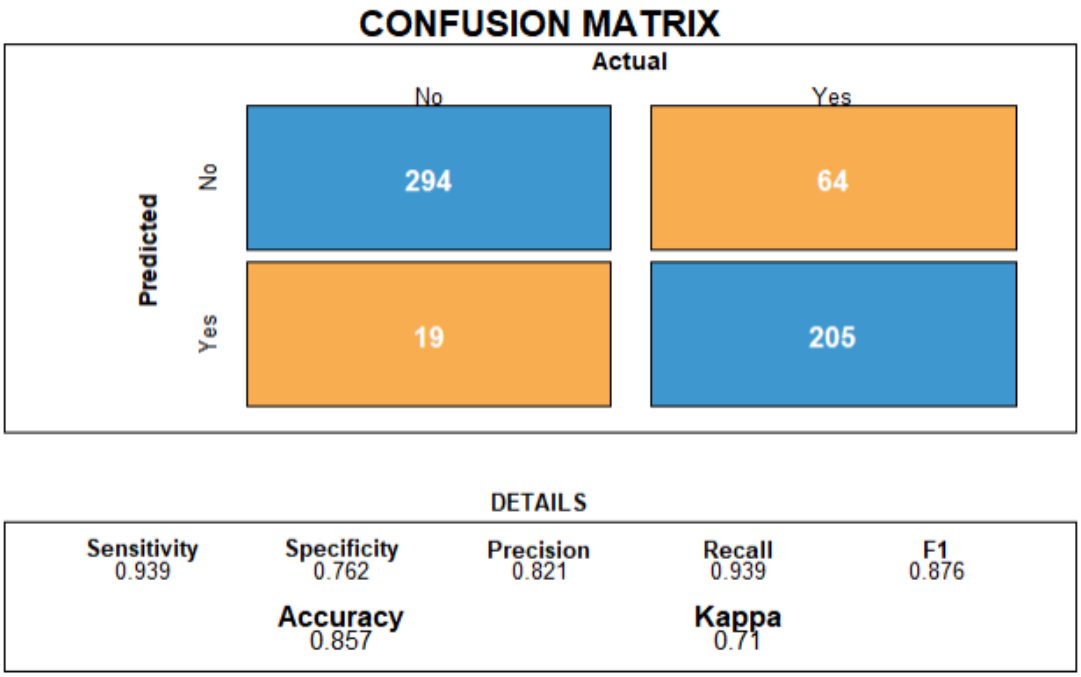


Konfuzijska matrica

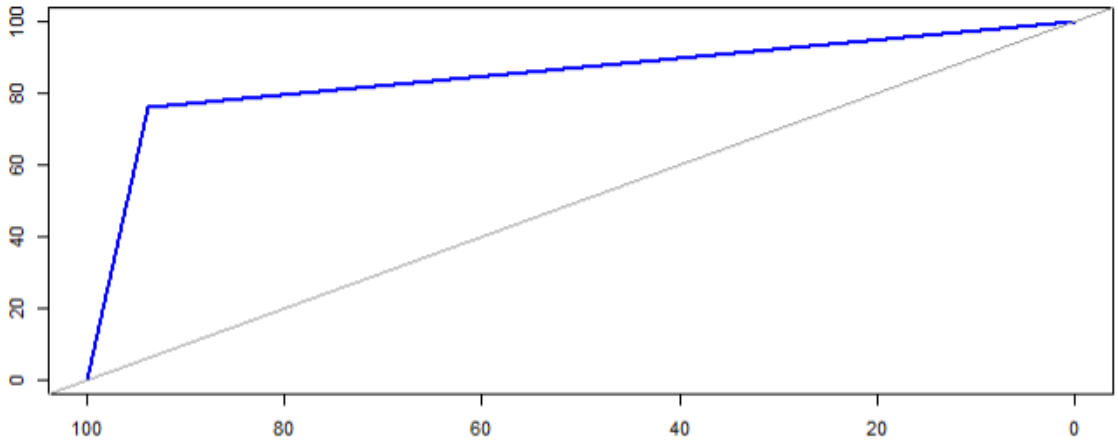


ROC kriva

Prikaz rezultata nakon SMOTE balansiranja:



Konfuzijska matrica



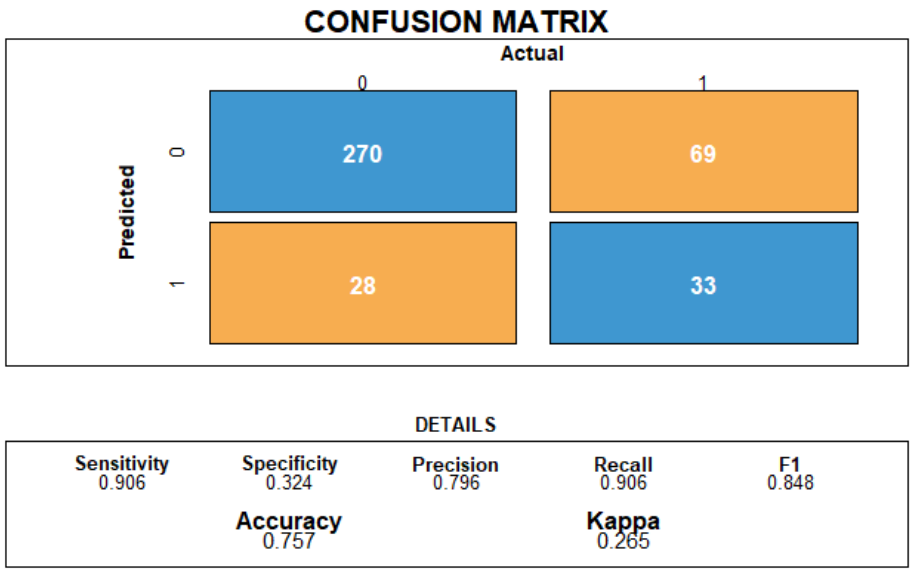
ROC kriva

- <u>model klasifikacije koji koristi neuralne mreže</u>

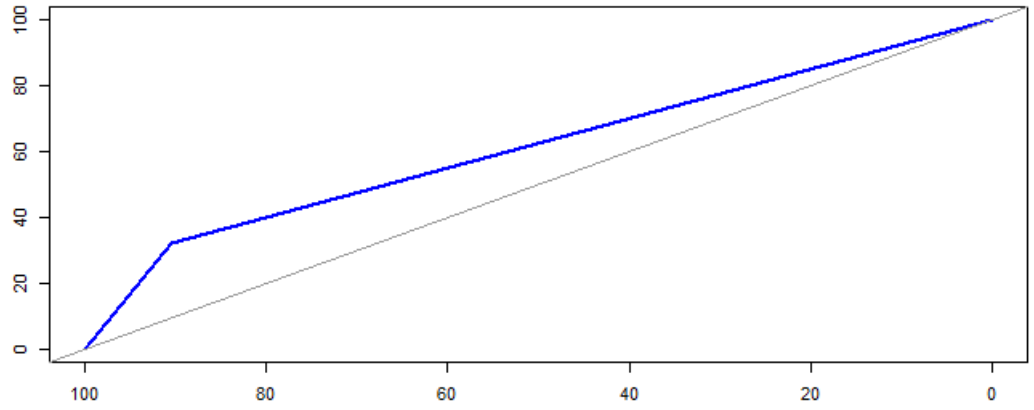
Prije kreiranja modela koje koriste neuralne mreže izvršena je priprema podataka prema uputama datim na vježbama. Podjela podatak na trening i tesni skup je urađena na isti način kao za prethodne vježbe.

- Perceptron

Model je izgrađen pozivom funkcije `neuralnet()`, pri čemu je parametar `hidden` postavljen na 1. Prikaz evaluacije osnovnog modela je dat ispod.



Konfuzijska matrica za osnovni model



ROC kriva za osnovni model

Za cross-validaciju je korištena pomoćna funkcija `kfold_model_one()`, koja prima dva parametra: podaci i k (prilagođena funkcija sa zadaće 1). Prikazane su vrijednosti 3-fold validacije:

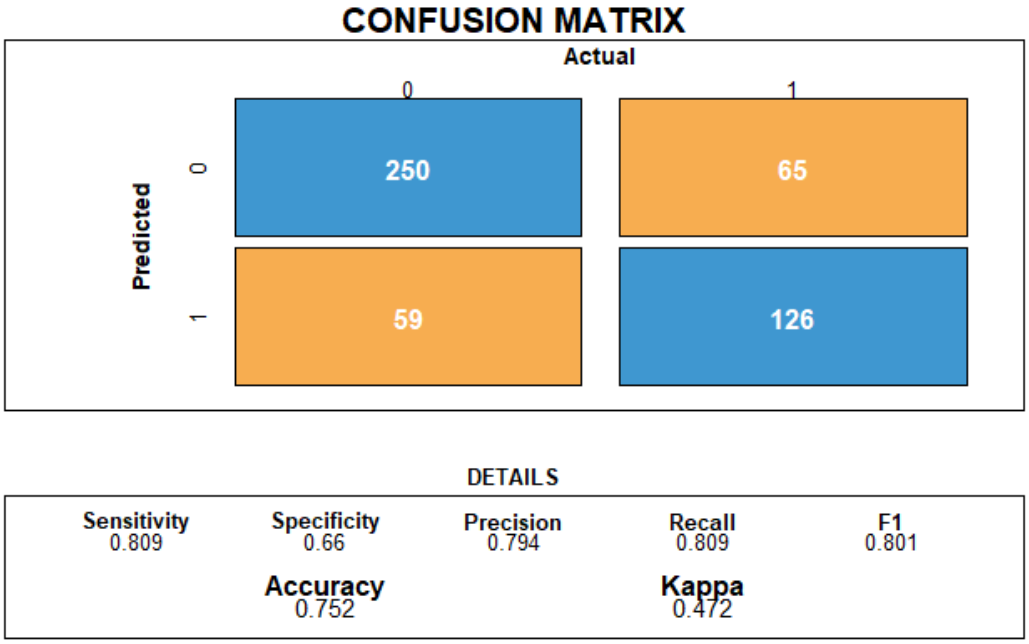
```
3-fold validacija
Najveca tacnost: 0.8078078 , fold: 2, najveca kappa: 0.3370764 , fold: 2
Najmanja tacnost: 0.7807808 , fold: 1, najmanja kappa: 0.2668959 , fold: 1
Srednja tacnost: 0.7937938, srednja kappa: 0.2975
```

Za bootstrap je korištena pomoćna funkcija `bootstrap_model_one()`, koja prima dva parametra: podaci i k (prilagođena funkcija sa zadaće 1). Prikazane su vrijednosti 3-fold bootstrapinga.

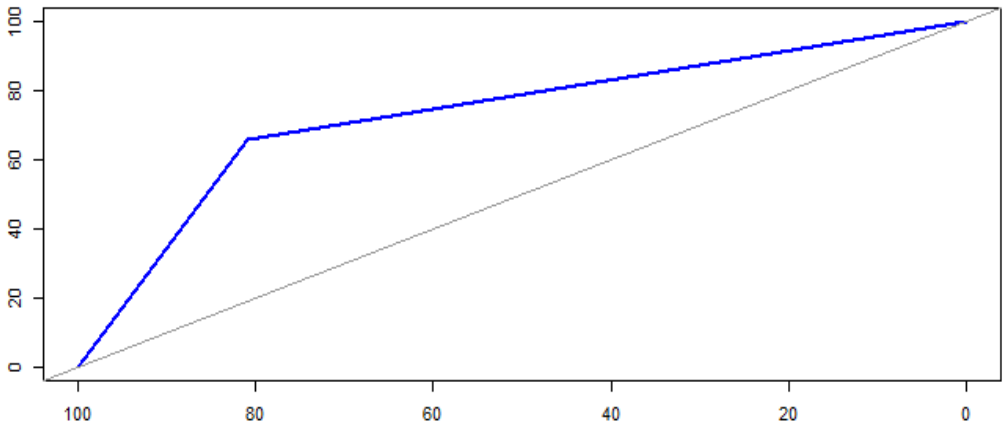
```
3-fold bootstrap
Najveca tacnost: 0.8138138 , fold: 2, najveca kappa: 0.3557587 , fold: 2
Najmanja tacnost: 0.7657658 , fold: 3, najmanja kappa: 0 , fold: 3
Srednja tacnost: 0.7852853, srednja kappa: 0.2124218
```

Balansiranje podataka (oversampling)

Balansiranje podataka je izvršeno na isti način zao i za prethodne modele. Evaluacija modela nakon balansiranja je data ispod:



Konfuzijska matrica za model nakon balansiranja



ROC kriva nakon balansiranja

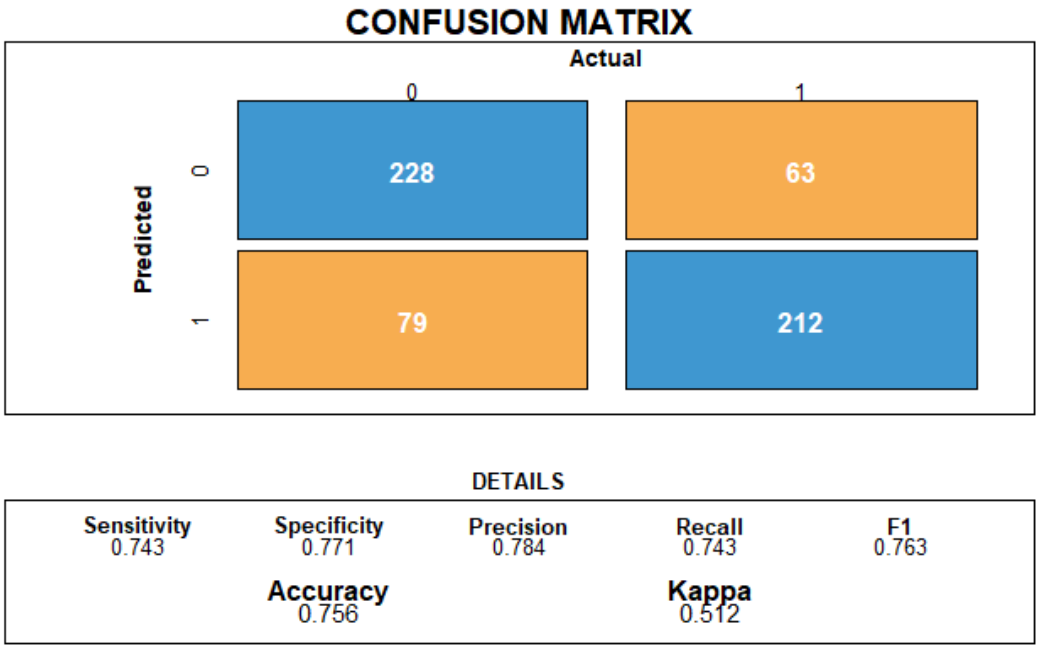
Cross-validacija i bootstrap su izvršeni na isti način kako je opisano ranije, a rezultati su prikazani na slikama ispod:

```
                                                    5721 error: 881.48777
time: 2.37 secs
3-fold validacija
Najveca tacnost: 0.7517986 , fold: 1, najveca kappa: 0.4752357 , fold: 1
Najmanja tacnost: 0.7226891 , fold: 3, najmanja kappa: 0.4225884 , fold: 3
Srednja tacnost: 0.7355935, srednja kappa: 0.4466145

0.01023406121111191
                                                    6017 error: 875.79403
time: 4.41 secs
3-fold bootstrap
Najveca tacnost: 0.7503001 , fold: 3, najveca kappa: 0.4711725 , fold: 3
Najmanja tacnost: 0.7442977 , fold: 1, najmanja kappa: 0.4563046 , fold: 1
Srednja tacnost: 0.7466987, srednja kappa: 0.4650974
```

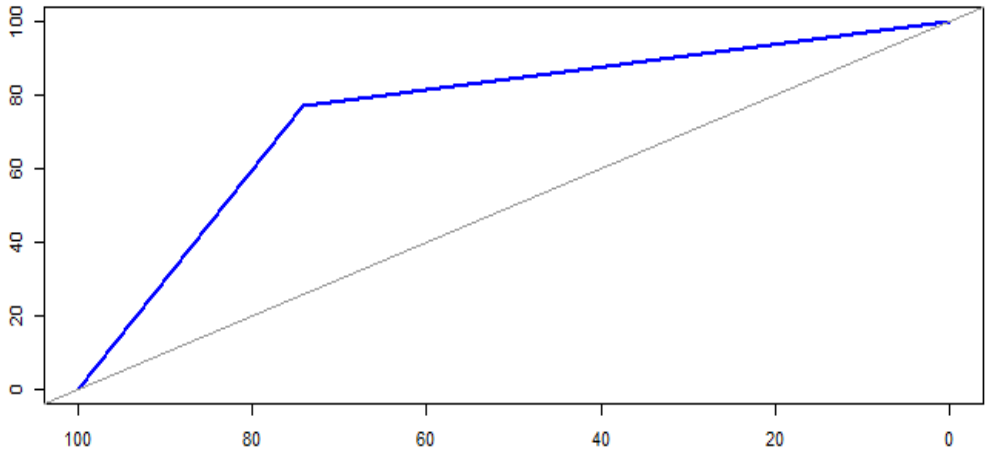
Balansiranje podataka (SMOTE)

Balansiranje podataka SMOTE algoritmom je urađeno na isti način kao i za prethodne modele.

Evaluacija modela nakon SMOTE balansiranja je prikazana na slikama ispod:



Prikaz konfuzijske matrice nakon SMOTE-a



Prikaz ROC krive nakon SMOTE-a

Rezultati 3-fold cross-validacije i 3-fold bootstrapinga su dati ispod:

```
time: 0.66 secs
3-fold validacija
Najveca tacnost: 0.7340206 , fold: 1, najveca kappa: 0.4667081 , fold: 1
Najmanja tacnost: 0.7120743 , fold: 3, najmanja kappa: 0.4233361 , fold: 3
Srednja tacnost: 0.7262697, srednja kappa: 0.4510965
```

```
match.
3-fold bootstrap
Najveca tacnost: 0.7358101 , fold: 2, najveca kappa: 0.46349 , fold: 2
Najmanja tacnost: 0.5252838 , fold: 3, najmanja kappa: 0 , fold: 3
Srednja tacnost: 0.6563467, srednja kappa: 0.2933134
```

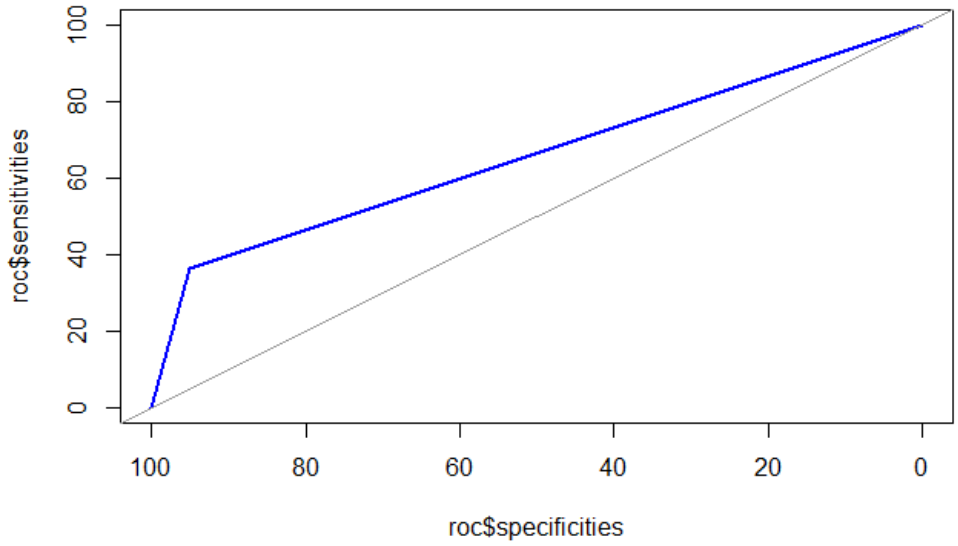
Ensamble tehnike - *bagging*

Kako bi primijenili bagging ensamble tehniku na model logističke regresije, korištene su ranije opisane funkcije `bagging_model_one()` i `predict_bagging()`. Rezultati bagging-a osnovnog modela:

```
0.0158961994535622
0.0158961994535622
0.0158961994535622
0.0155494341517644
0.0150435080832512
0.0145879208556313
0.0145879208556313
0.0132849139889402
0.0132849139889402
0.0132849139889402
0.0132849139889402
0.0132849139889402
time: 17.62 secs
Setting levels: control = 0, case = 1
Setting direction: controls < cases
```

```
28000 min thresh:
29000 min thresh:
30000 min thresh:
31000 min thresh:
32000 min thresh:
33000 min thresh:
34000 min thresh:
35000 min thresh:
36000 min thresh:
37000 min thresh:
38000 min thresh:
38530 error: 592.50729
```

Greška bagging modela perceptrona

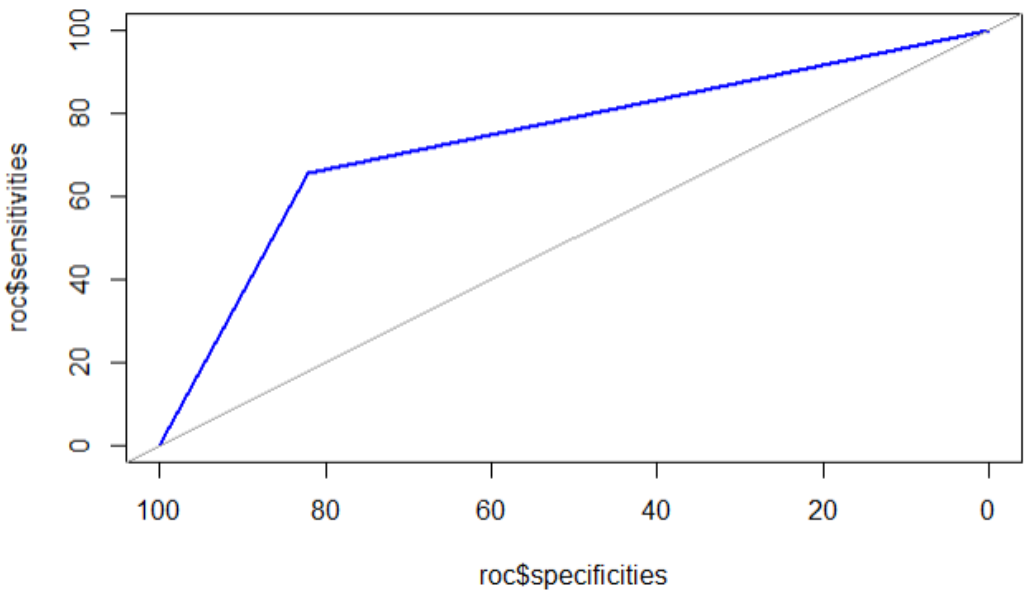


ROC kriva

Prikaz rezultata nakon oversampling-a:

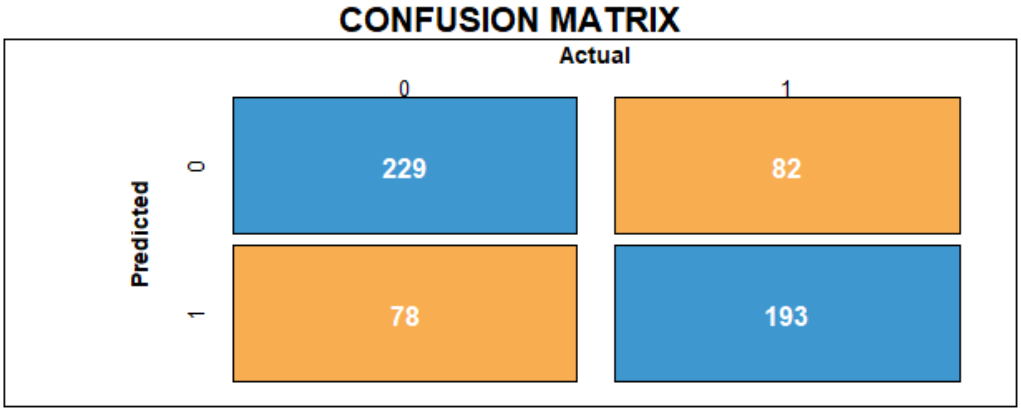
```
time: 2.5 secs
hidden: 1    thresh: 0.01    rep: 1/1    steps:    1000 min thresh:
0.350912768437524
                                                    2000 min thresh:
0.182368719525977
                                                    3000 min thresh:
0.102424255062134
                                                    4000 min thresh:
0.0706032581364212
                                                    5000 min thresh:
0.0530905660715442
                                                    6000 min thresh:
0.0415794944060699
                                                    7000 min thresh:
0.0326229369609885
                                                    8000 min thresh:
0.0235511736374914
                                                    9000 min thresh:
0.0212054995996411
                                                   10000 min thresh:
0.0153337393157684
                                                   11000 min thresh:
0.012675768891485
                                                   11980 error: 879.36924
time: 5.95 secs
Setting levels: control = 0, case = 1
Setting direction: controls < cases
```

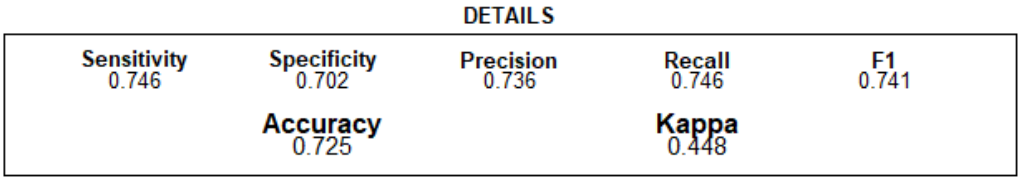
Greška bagging modela perceptrona (oversampling)



ROC kriva (oversampling)

Prikaz rezultata nakon SMOTE balansiranja:

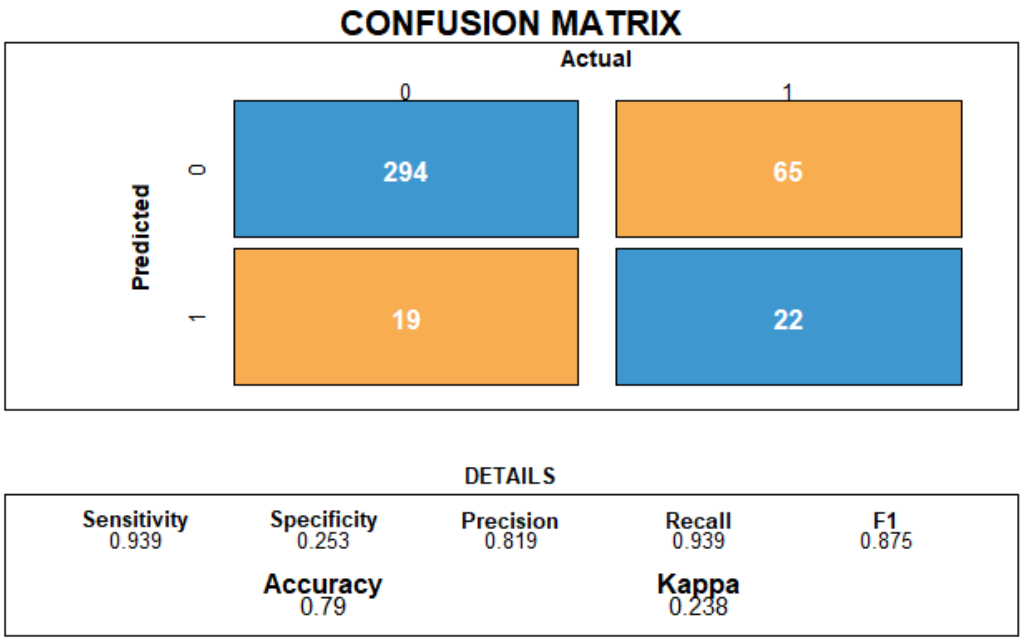




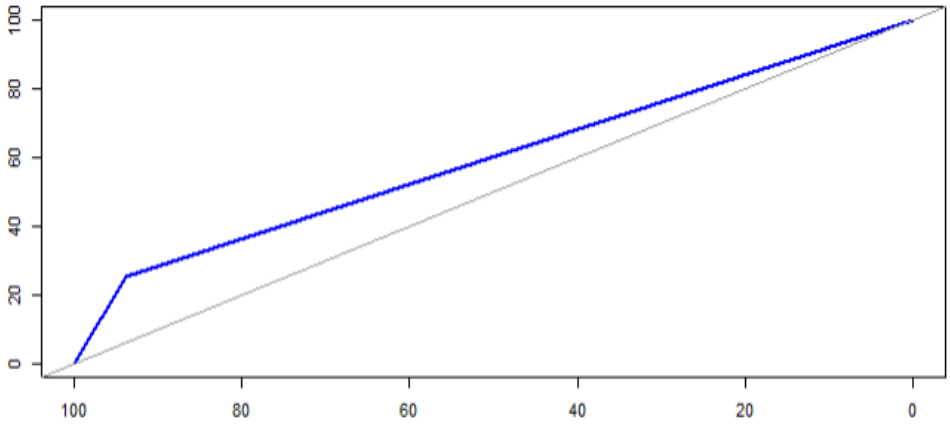
Konfuzijska matrica (SMOTE balansiranje)

- Perceptron sa više epoha

Model je izgrađen pozivom funkcije `neuralnet()`, pri čemu je parametar `hidden` postavljen na 1 a parametar `rep` na 3. Prikaz evaluacije osnovnog modela je dat ispod.



Konfuzijska matrica za osnovni model



ROC kriva za osnovni model

```
Train accuracy: 0.8022528
Test accuracy: 0.79
```

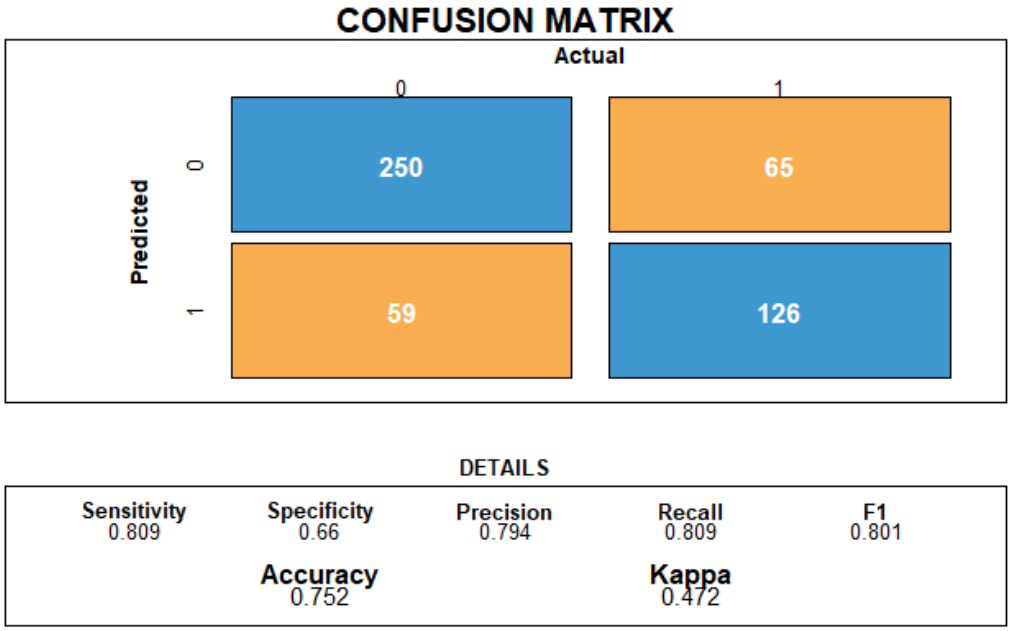
Accuracy osnovni model

Za bootstrap je korištena pomoćna funkcija `bootstrap_model_one_rep()`, koja prima dva parametra: podaci i k (prilagođena funkcija sa zadaće 1). Prikazane su vrijednosti 3-fold bootstrapinga.

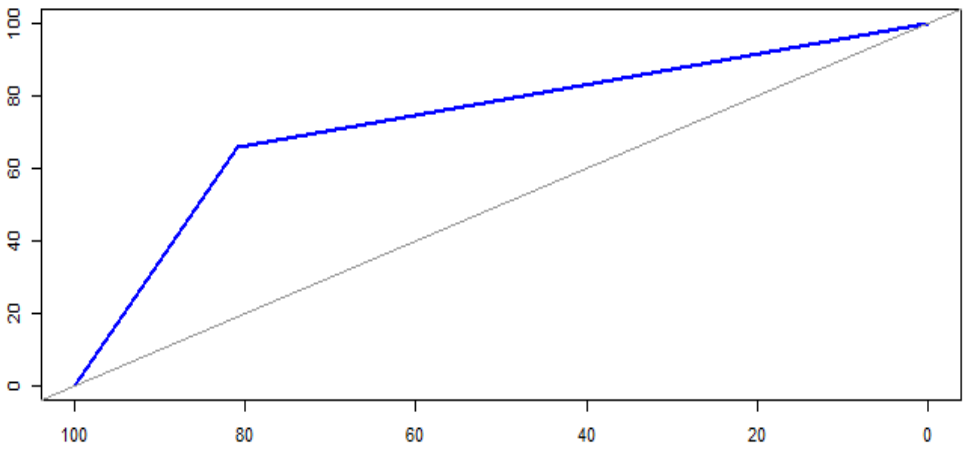
```
3-fold bootstrap
Najveca tacnost: 0.8153153 , fold: 2, najveca kappa: 0.3640598 , fold: 2
Najmanja tacnost: 0.7702703 , fold: 3, najmanja kappa: 0.2312254 , fold: 3
Srednja tacnost: 0.7942943, srednja kappa: 0.2950655
```

Balansiranje podataka (oversampling)

Balansiranje podataka je izvršeno na isti način zao i za prethodne modele. Evaluacija modela nakon balansiranja je data ispod:



Konfuzijska matrica za model nakon balansiranja



ROC kriva nakon balansiranja

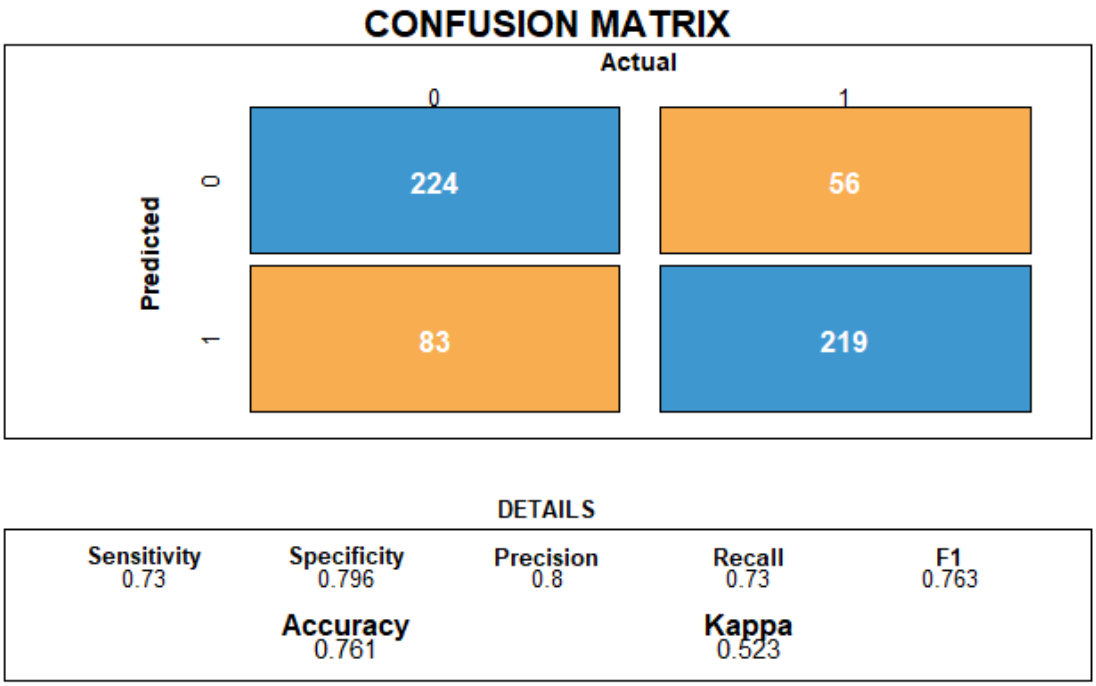
```
hidden: 1    thresh: 0.01    rep: 1/3    steps:   11941 error: 1069.80897
time: 7.39 secs
hidden: 1    thresh: 0.01    rep: 2/3    steps:   13920 error: 1069.80946
time: 13.4 secs
hidden: 1    thresh: 0.01    rep: 3/3    steps:   12503 error: 1069.80949
time: 9.89 secs
```

Greška za perceptron sa više epoha (oversampling)

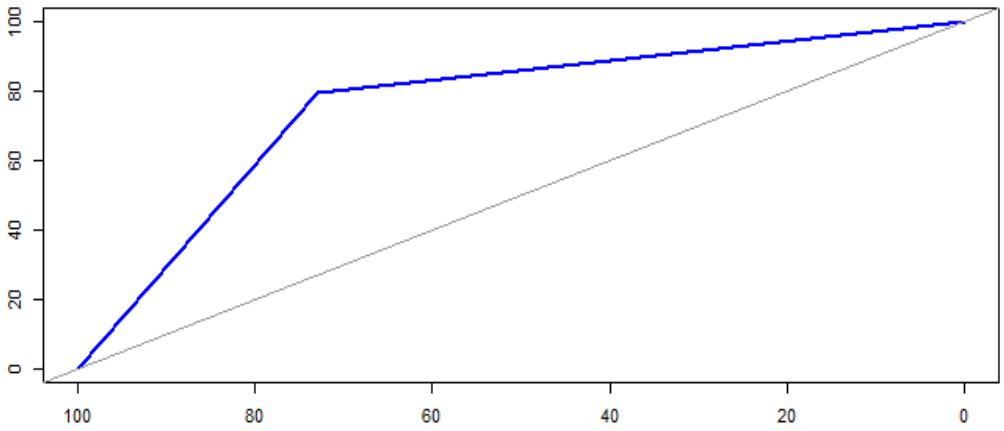
Balansiranje podataka (SMOTE)

Balansiranje podataka SMOTE algoritmom je urađeno na isti način kao i za prethodne modele.

Evaluacija modela nakon SMOTE balansiranja je prikazana na slikama ispod:



Prikaz konfuzijske matrice nakon SMOTE balansiranja



Prikaz ROC krive nakon SMOTE balansiranja

```
hidden: 1    thresh: 0.01    rep: 1/3    steps:    1329 error: 1270.38796
time: 0.67 secs
hidden: 1    thresh: 0.01    rep: 2/3    steps:    1281 error: 1270.42743
time: 0.69 secs
hidden: 1    thresh: 0.01    rep: 3/3    steps:    8415 error: 1281.96893
time: 4.29 secs
```

Greška za perceptron sa više epoha (SMOTE)

Rezultati 3-fold cross-validacije i 3-fold bootstrapinga su dati ispod:

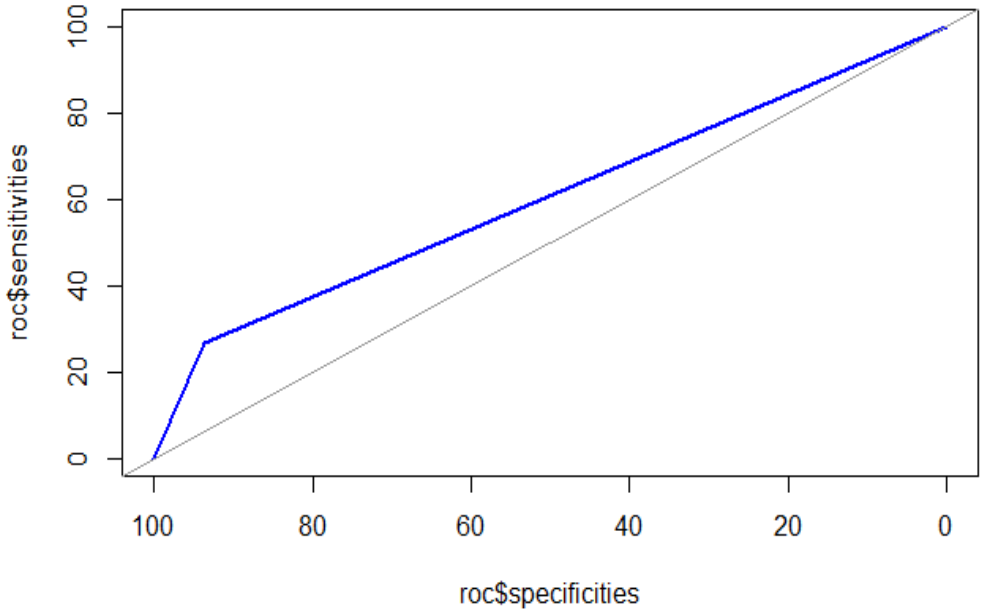
```
3-fold validacija
Najveca tacnost: 0.7172343 , fold: 2, najveca kappa: 0.4335606 , fold: 2
Najmanja tacnost: 0.7110423 , fold: 3, najmanja kappa: 0.4212433 , fold: 3
Srednja tacnost: 0.7149238, srednja kappa: 0.4292227

3-fold bootstrap
Najveca tacnost: 0.7275542 , fold: 2, najveca kappa: 0.4548337 , fold: 2
Najmanja tacnost: 0.7213622 , fold: 1, najmanja kappa: 0.4410093 , fold: 1
Srednja tacnost: 0.7241142, srednja kappa: 0.4472983
```

Ensamble tehnike - *bagging*

Kako bi primijenili bagging ensamble tehniku na model logističke regresije, korištene su ranije opisane funkcije `bagging_model_one_rep()` i `predict_bagging()`. Rezultati bagging-a osnovnog modela:

```
hidden: 1    thresh: 0.01    rep: 1/3    steps:   36006 error: 579.83054
time: 17.62 secs
hidden: 1    thresh: 0.01    rep: 2/3    steps:   40565 error: 579.83012
time: 20.28 secs
hidden: 1    thresh: 0.01    rep: 3/3    steps:      36 error: 719.08335
time: 0.02 secs
hidden: 1    thresh: 0.01    rep: 1/3    steps:   59171 error: 579.16169
time: 29.08 secs
hidden: 1    thresh: 0.01    rep: 2/3    steps:   57687 error: 579.14652
time: 28.26 secs
hidden: 1    thresh: 0.01    rep: 3/3    steps:   62085 error: 579.16193
time: 30.12 secs
hidden: 1    thresh: 0.01    rep: 1/3    steps:      35 error: 725.06396
time: 0.02 secs
hidden: 1    thresh: 0.01    rep: 2/3    steps:      34 error: 725.06418
time: 0.01 secs
hidden: 1    thresh: 0.01    rep: 3/3    steps:   19744 error: 582.42614
time: 9.3 secs
Setting levels: control = 0, case = 1
Setting direction: controls < cases
```

Greška bagging modela perceptrona sa više epoha

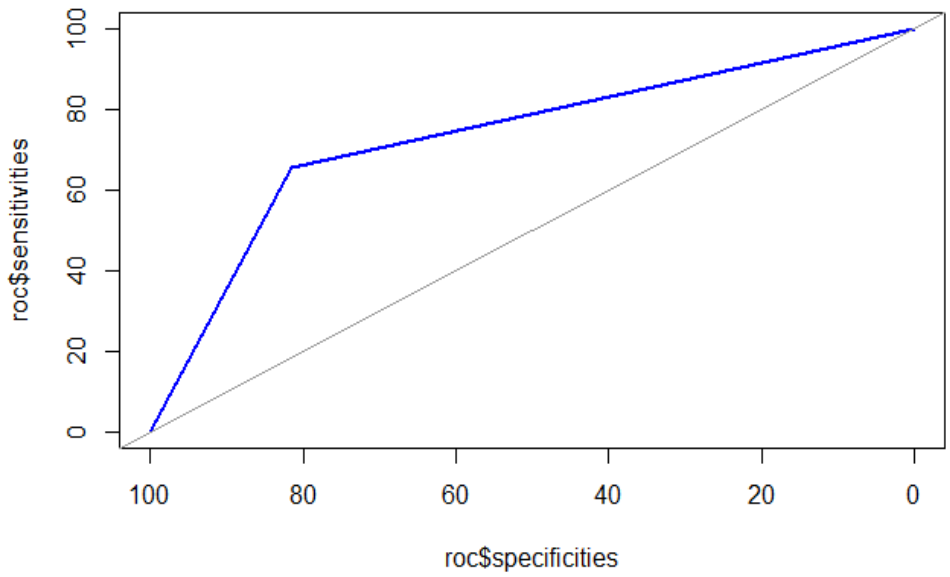


ROC kriva

Prikaz rezultata nakon oversampling-a:

```
hidden: 1    thresh: 0.01    rep: 1/3    steps:   15453 error: 885.23846
time: 7.75 secs
hidden: 1    thresh: 0.01    rep: 2/3    steps:   17136 error: 885.23692
time: 9.78 secs
hidden: 1    thresh: 0.01    rep: 3/3    steps:   25146 error: 885.23882
time: 13.41 secs
hidden: 1    thresh: 0.01    rep: 1/3    steps:   20207 error: 892.43095
time: 10.4 secs
hidden: 1    thresh: 0.01    rep: 2/3    steps:   22230 error: 892.42983
time: 11.96 secs
hidden: 1    thresh: 0.01    rep: 3/3    steps:   21310 error: 892.43031
time: 11.07 secs
hidden: 1    thresh: 0.01    rep: 1/3    steps:    1171 error: 900.49977
time: 0.51 secs
hidden: 1    thresh: 0.01    rep: 2/3    steps:    3416 error: 900.50007
time: 1.94 secs
hidden: 1    thresh: 0.01    rep: 3/3    steps:     988 error: 900.49945
time: 0.48 secs
Setting levels: control = 0, case = 1
Setting direction: controls < cases
```

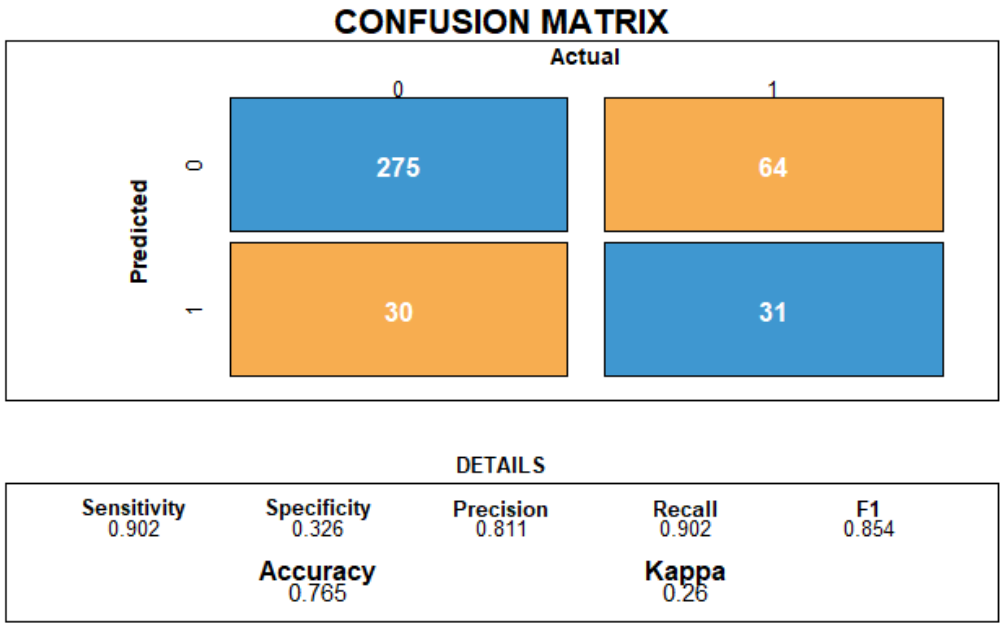
Greška bagging modela perceptrona sa više epoha (oversampling)



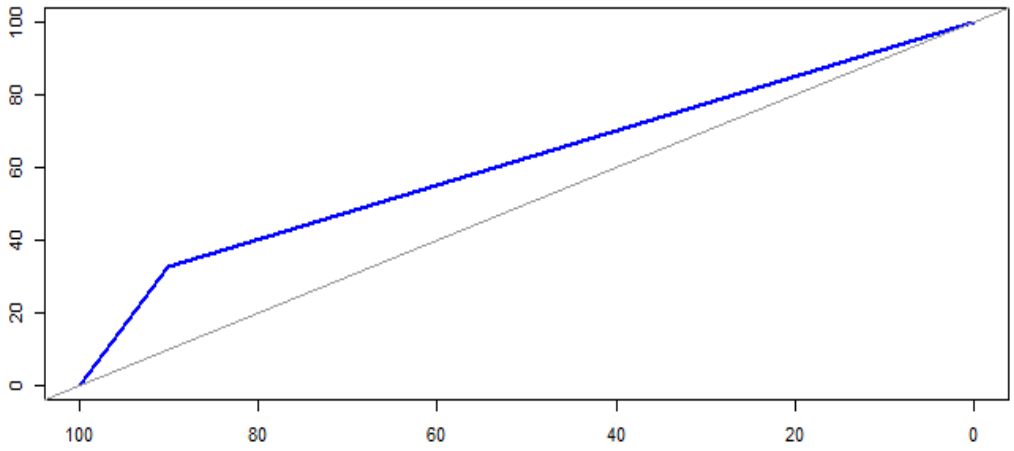
ROC kriva (oversampling)

- Perceptron sa više neurona

Model je izgrađen pozivom funkcije `neuralnet()`, pri čemu je parametar `hidden` postavljen na 4 a parametar `rep` na 5. Prikaz evaluacije osnovnog modela je dat ispod.



Konfuzijska matrica za osnovni model



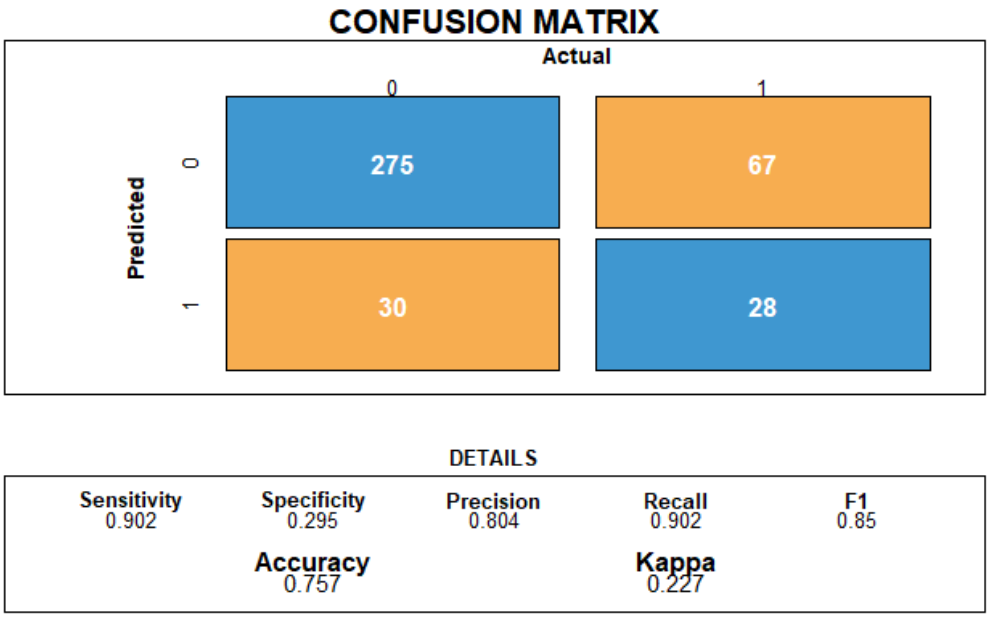
ROC kriva za osnovni model

```
Train accuracy: 0.8254068
Test accuracy: 0.765
```

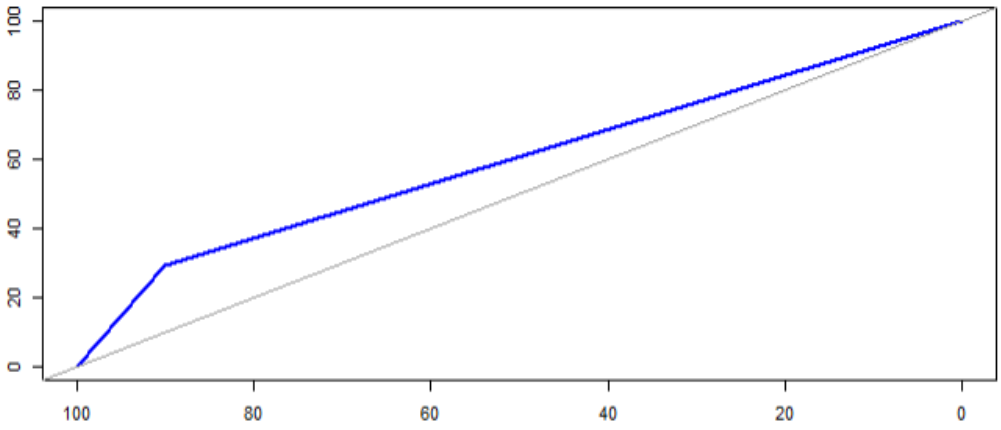
Accuracy osnovni model

- Perceptron sa više slojeva

Model je izgrađen pozivom funkcije `neuralnet()`, pri čemu je parametar `hidden` postavljen na c(2,1). Prikaz evaluacije osnovnog modela je dat ispod.



Konfuzijska matrica za osnovni model



ROC kriva za osnovni model

```
Train accuracy: 0.8091364
Test accuracy: 0.7575
```

Accuracy osnovni model

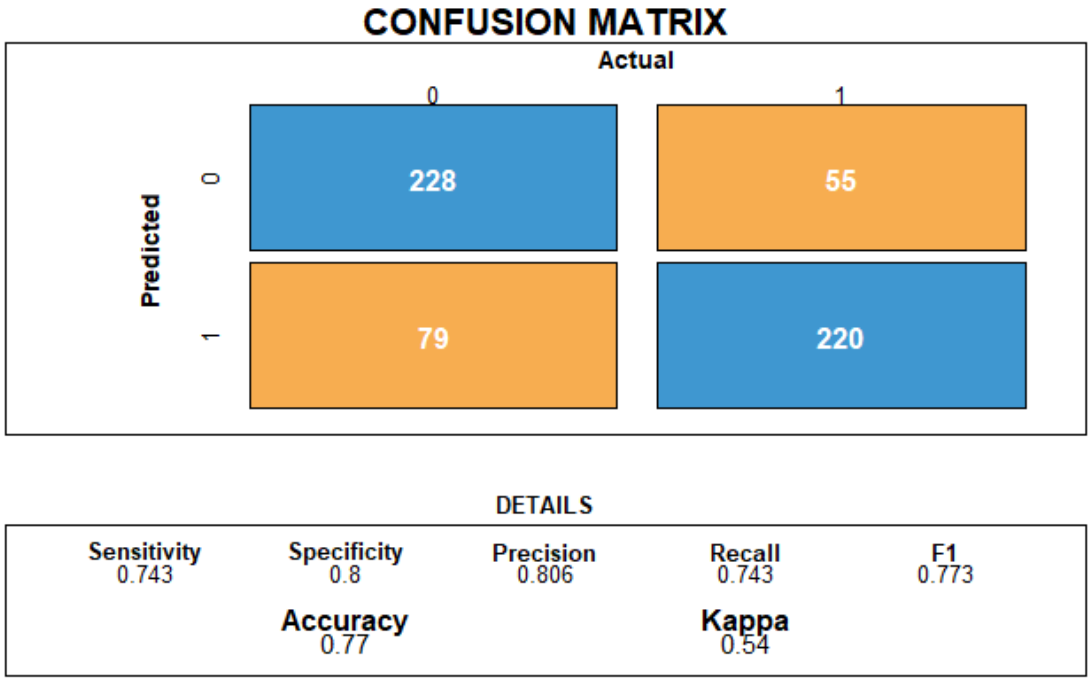
Za k-fold bootstrap su korištene pomoćne funkcije `kfold_model_multiple_layers()` i `bootstrap_model_multiple_layers()`. Prikazane su vrijednosti 3-fold-a i 3-fold bootstrapinga.

```
Time: 31.6 secs
3-fold validacija
Najveca tacnost: 0.8153153 , fold: 3, najveca kappa: 0.3551896 , fold: 3
Najmanja tacnost: 0.7507508 , fold: 2, najmanja kappa: 0.2248771 , fold: 1
Srednja tacnost: 0.7797798, srednja kappa: 0.2841426

3-fold bootstrap
Najveca tacnost: 0.7937938 , fold: 1, najveca kappa: 0.3756868 , fold: 2
Najmanja tacnost: 0.7597598 , fold: 2, najmanja kappa: 0.3293423 , fold: 1
Srednja tacnost: 0.7767768, srednja kappa: 0.3525145
```

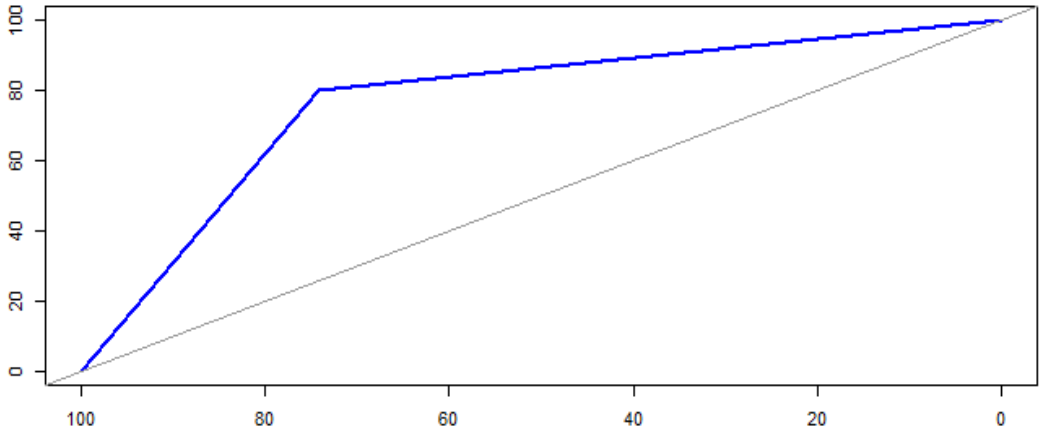
Balansiranje podataka (SMOTE)

Balansiranje podataka SMOTE algoritmom je urađeno na isti način kao i za prethodne modele.

Evaluacija modela nakon SMOTE balansiranja je prikazana na slikama ispod:



Prikaz konfuzijske matrice nakon SMOTE balansiranja



Prikaz ROC krive nakon SMOTE balansiranja

```
hidden: 2, 1    thresh: 0.01    rep: 1/1    steps:    23257      error: 1225.03985
time: 23.62 secs
```

Greška za perceptron sa više slojeva(SMOTE)

```
Train accuracy: 0.7317283
Test accuracy: 0.7697595
```

Tačnost za perceptron sa više slojeva(SMOTE)

NAPOMENA

Pri pokretanju neuralnih mreža povremeno su nam se javljale greške prikazane u nastavku:

```
Error in cbind(1, pred) %*% weights[[num_hidden_layers + 1]] :
 requires numeric/complex matrix/vector arguments
```

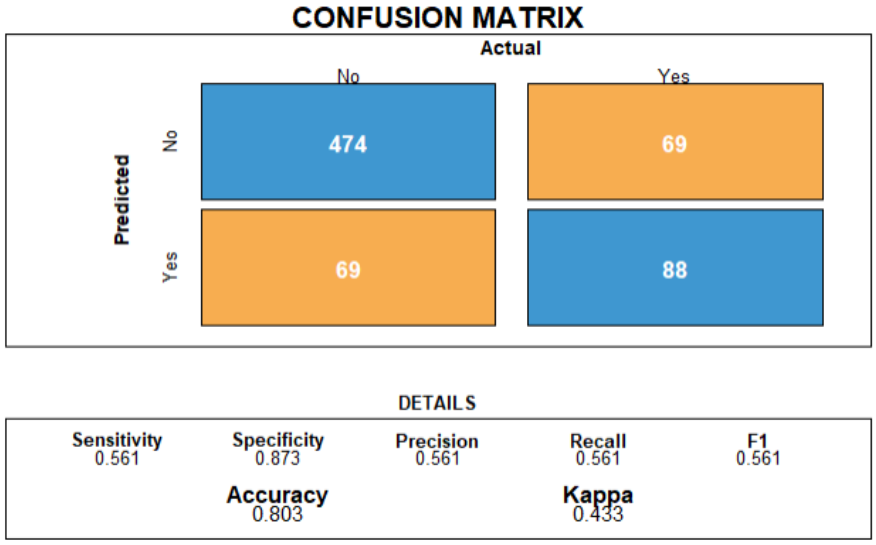
Show Traceback

```
hidden: 1    thresh: 0.01    rep: 1/3    steps: stepmax min thresh:
0.0131678163064632
hidden: 1    thresh: 0.01    rep: 2/3    steps: stepmax min thresh:
0.0116548824090853
hidden: 1    thresh: 0.01    rep: 3/3    steps: stepmax min thresh:
0.0117245244226313
warning: Algorithm did not converge in 3 of 3 repetition(s) within the stepmax.
```

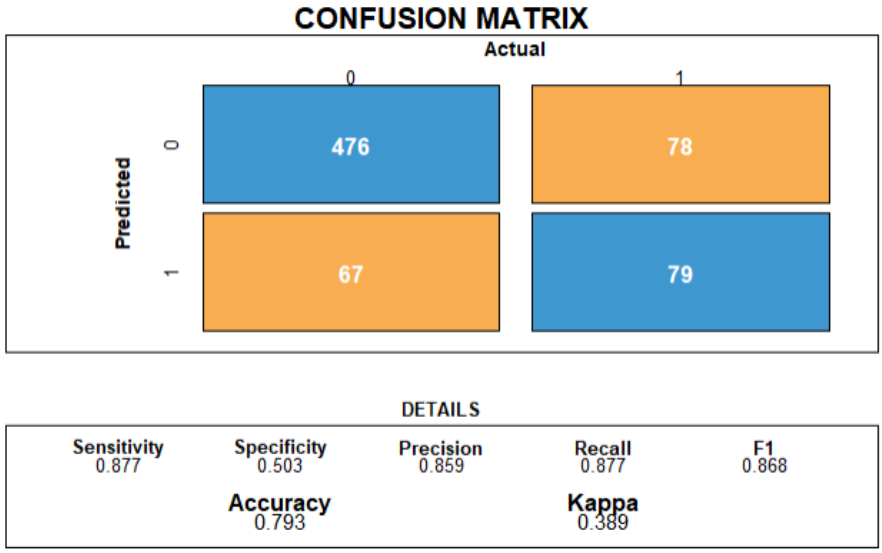
I jedna i druga greška su onemogućile dokumentovanje performansi određenih modela kod neuralnih mreža zbog nekonzistentnosti prilikom više uzastopnih pokretanja, obzirom da se znalo desiti da neuralna mreža nekada konvergira a nekada ne, bez promjene bilo kakvih ulaznih parametara ili ponovne podjele podataka na trening i testni skup.

b) Testiranje najboljeg modela

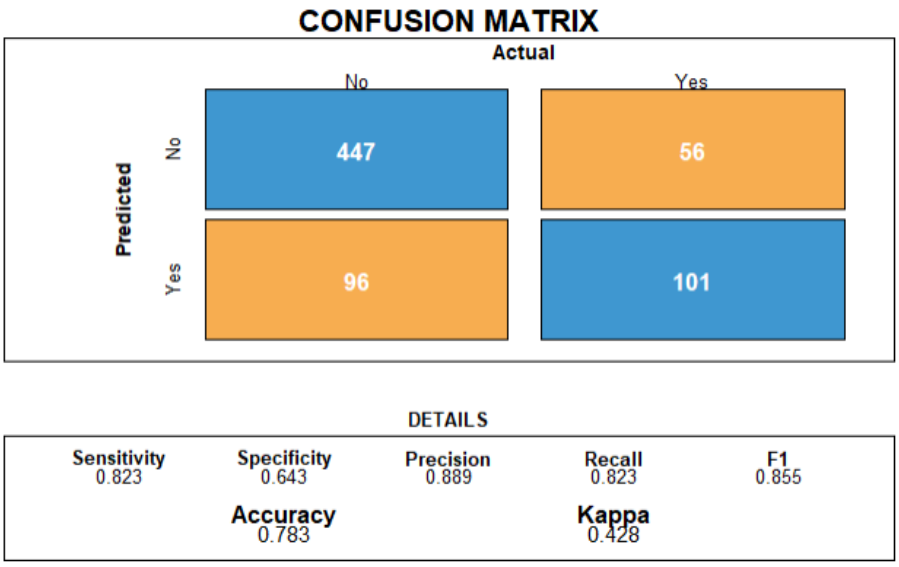
U nastavku su prikazane performanse testiranja 3 od kreiranih modela nad testnim skupom weather_data_test:



Prikaz konfuzijske matrice za model Naive Bayes



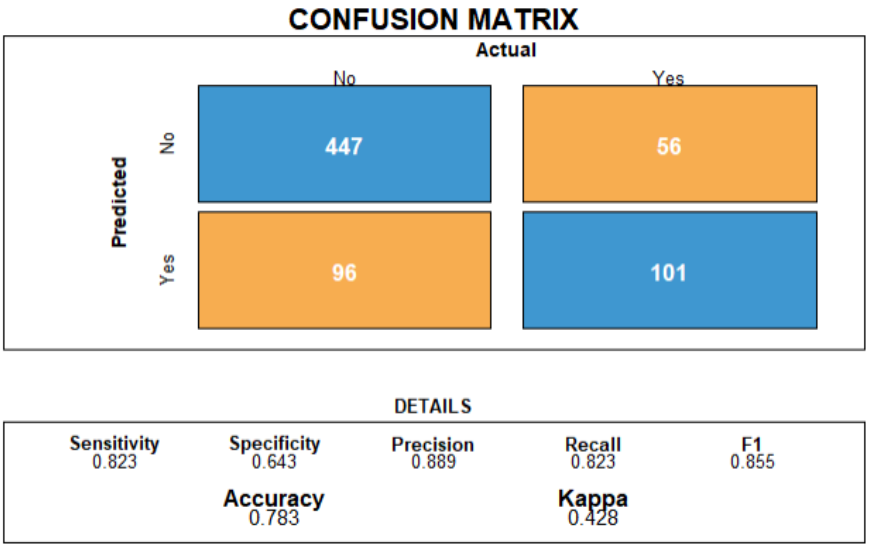
Prikaz konfuzijske matrice za perceptron sa više epoha



Prikaz konfuzijske matrice za model logističke regresije

Prikazane konfuzijske matrice odgovaraju modelima koji su dali najbolje rezultate nad testnim skupom podataka. Model logističke regresije je izgrađen pomoću balansiranih podataka metodom oversampling-a, te je dao najbolje rezultate od svih modela, kao i od modela logističke regresije izgrađene pomoću podataka prije balansiranja, što opravdava činjenica da je početni skup podataka u trening datasetu značajno nebalansiran. Perceptron sa više epoha je izgrađen nad balansiranim podacima, te ima slične performanse kao i najbolji model, sa izuzetkom metrike specifičnosti koja značajnije odstupa od najboljeg modela u negativnom smislu. Kod modela Naive Bayes koji je izgrađen nad nebalansiranim podacima, vrijednost metrike specifičnosti je značajno bolji nego kod ostala dva modela, ali vrijednosti ostalih metrika većinom poprimaju značajno manje vrijednosti u poređenju sa druga dva modela.

Na osnovu prethodne analize, možemo zaključiti da smo najbolje performanse postigli sa modelom logističke regresije kreiranim nad balansiranim podacima metodom oversampling-a. Ako uporedimo rezultate najboljeg modela zadaće 2 sa rezultatima najboljeg modela zadaće 1, možemo uočiti da je u zadaći 2 došlo do povećanja vrijednosti metrika osjetljivosti, preciznosti, recall, F1, te tačnosti. Kappa vrijednost je približno ista, a specifičnost je u zadaći dva smanjena. Kada sumiramo sve navedene promjene metrika, možemo zaključiti da je generalno došlo do poboljšanja modela predikcije vremenske prognoze u zadaći 2.



Prikaz konfuzijske matrice najboljeg modela

Zadatak 2. (Višestruka linearna regresija)

Prvo je kreiran novi atribut WeatherStability na sljedeći način:

```
podaci$WeatherStability = -1
```

Na ovaj način su sve vrijednosti u toj koloni postavljene na -1. Zatim je izračunat parametar k po uputama iz zadatka, i popunjene su vrijednosti u redovima koje ispunjavaju zadate uslove. Na kraju su po uputama iz zadatka popunjene vrijednosti 0 i 1 u odgovarajućim redovima.

Prikaz jednog dijela vrijednosti navedene kolone je dat na slici ispod:

```
 [1] 0.9904051 0.9849777 0.9817794 0.9782904 0.9797441 0.9850746 0.9831363 0.9828455 0.9800349 0.9785811 0.9819733 0.9865284 0.9897267 0.9819733
[15] 0.9867222 0.9806164 0.9809072 0.9849777 0.9846870 0.9787750 0.9770304 0.9810041 0.9978678 0.9833301 0.9866253 0.9918589 0.9962202 0.9788719
[29] 0.9762551 0.9738321 0.9818763 0.9905020 0.9808102 0.9800349 0.9851715 0.9907928 0.9891452 0.9895329 0.9865284 0.9805195 0.9849777 0.9883698
[43] 0.9829424 0.8673469 0.9838147 0.9920527 0.9893390 0.9886606 0.9961233 0.9872068 0.9896298 0.9790657 0.9937972 0.9943788 0.9787750 0.9775150
[57] 0.9795503 0.9938942 0.9893390 0.9781934 0.9807133 0.9948633 0.9881760 0.9814887 0.9849777 0.9883698 0.9739291 0.8469388 0.9934096 0.9830393
[71] 0.9935065 0.9838147 0.9791626 0.9858500 0.9748982 0.9842993 0.9821671 0.9825548 0.9834270 0.9811979 0.9772243 0.9819733 0.9724753 0.9929250
[85] 0.9808102 0.9960264 0.9847839 0.9887575 0.9992247 0.9901144 0.9924404 0.9849777 0.9849777 0.9802287 0.9935065 0.9833301 0.9774181 0.9888544
[99] 0.9825548 0.9849777 0.9798411 0.9908897 0.9805195 0.9849777 0.9778058 0.9858500 0.9847839 0.9807133 0.9867222 0.9922466 0.9813917 0.9849777
```

Podjela podataka na trening i testni skup je urađena kao što je ranije objašnjeno. Model je kreiran korištenjem funkcije `lm()`. Prikaz metrika nakon kreiranog osnovnog modela je dat na slici ispod:

```
Residual standard error: 0.05256 on 1510 degrees of freedom
Multiple R-squared:  0.1236,    Adjusted R-squared:  0.07311
F-statistic: 2.448 on 87 and 1510 DF,  p-value: 1.729e-11
```

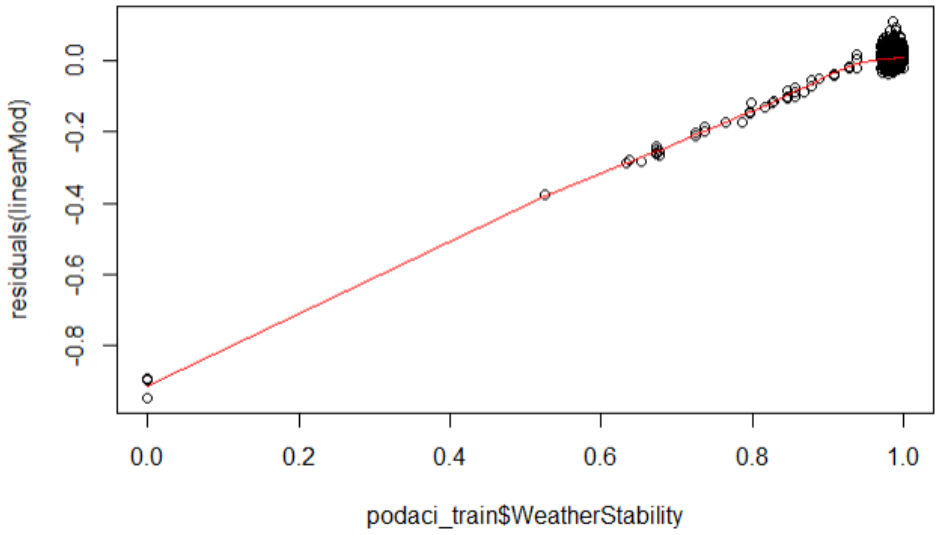
Dati prikaz se dobije pozivom funkcije `summary()`.

Rezultati RMSE i MAE metrika se dobiju pozivom funkcija `rmse()` i `mae()` respektivno, koje primaju kao parametre kolonu WeatherStability i predikcije. Dobijeni rezultati su prikazani ispod:

```
RMSE: 0.07728836
MAE:  0.02254158
```

Ispitivanje pretpostavke linearnosti rezidualnih vrijednosti

Funkcija `residuals(model)` se koristi za prikaz grafika rezidualnih vrijednosti u odnosu na sve instance iz trening skupa podataka. Dobijeni grafik je prikazan na slici ispod:



Grafik rezidualnih vrijednosti

Može se uočiti da prava rezidualnih vrijednosti ima linearan oblik, pa se zaključuje da je pretpostavka linearnosti ispunjena.

Analiza kolinearnosti ulaznih varijabli

Kako bi se odredilo da li postoji kolinearnost ulaznih varijabli, neophodno je izračunati VIF koeficijent za izvršene predikcije. Za izračunavanje tog koeficijenta korištena je funkcija `vif()`. Dobijena vrijednost je prikazana ispod:

```
VIF: 0.06884102
```

S obzirom na to da je dobijena VIF vrijednost manja od 1, zaključujemo da je kolinearnost ulaznih varijabli niska.

Analiza auto-korelacije rezidualnih vrijednosti

Kako bi se analizirala auto-korelacija rezidualnih vrijednosti, korišten je Durbin-Watson test. Rezultat navedenog testa nakon poziva funkcije `durbinWatsonTest()` je prikazan ispod:

```
Rezultat Durbin-Watsonovog testa: 2.046662
```

S obzirom na to da je dobijena vrijednost približno jednaka 2, može se zaključiti da nema auto-korelacije.

Unapređenje modela

Kako bismo unaprijedili performanse kreiranog modela, odbacili smo kolone za koje je na prethodnoj zadaći utvrđeno da imaju visok stepen korelacije, a to su kolone MaxTemp i Cloud5pm. Nakon odbacivanja tih kolona, ponovo je izvršena podjela na trening i testni skup na već opisani način. Rezultati nakon unapređenja su prikazani na slici ispod:

```
Residual standard error: 0.05487 on 1511 degrees of freedom
Multiple R-squared:  0.1645,    Adjusted R-squared:  0.117
F-statistic:  3.46 on 86 and 1511 DF,  p-value: < 2.2e-16

RMSE: 0.07082576
MAE:  0.02021303
```

Može se uočiti da su se vrijednosti sve tri metrike (adjusted R-squared, RMSE i MAE) poboljšale.

Ensamble tehnike - Bagging

Kako bi primijenili bagging ensamble tehniku na dati model, korištene su ranije opisane funkcije `bagging_lm()` i `predict_bagging()`. Prikaz metrika nakon bagging-a je dat ispod:

```
RMSE: 0.06788603
MAE:  0.01895388
```

I u ovom slučaju možemo uočiti poboljšanje u odnosu na prethodne modele.