Predikcija plaćanja kredita

February 1, 2025

Student
Jovana Đukić
Mentor
Doc. dr Milana Grbić

Sadržaj

1 Uvod

Finansijski sektor igra značajnu ulogu u savremenim ekonomskim sisitemima omogućavajući kako pojedincima tako i kompanijama mogućnost dobijanja različitih sredstava putem kredita. Potreba za ovakvim mogućnostima kontinuirano raste a raste i potreba zajmodavca da procjeni mogućnost vraćanja kredita od strane pojedinca ili kompanije. Potrebno je procijeniti kada dužnik ispunjava uslove da uspori ili skroz ne otplati dug. Veliki broj ljudi je uključen kako bi se napravio model koji je precizan, skalabilan i pouzdan.

Predviđanje kreditnog zastoja je najvažniji aspekt upravljanja rizicima u bankarstvu i finansijama. Razvojem tehnologije i sa sve većom dostupnošću podataka, moderne metode analize, uključujući mašinsko učenje i statističko modelovanje, pružaju mogućnost preciznijeg procjenjivanja vjerovatnoće neispunjavanja obaveza od strane klijenata. Ove metode omogućavaju bankama da bolje razumiju profile rizika svojih klijenata, optimizuju politike odobravanja kredita i minimizuju potencijalne gubitke.

Cilj ovog rada je da se procjeni kreditni zastoj od strane klijenata na osnovu dostupnih podataka. Za potrebe istraživanja korišćeni su podaci dostupni na linku
https://www.kaggle.com/datasets/kmldas/loan-default-prediction/data [1].

U prvom dijelu se analiziraju podaci u cjelosti, zasebni atributi i međusobni odnosi atributa. Takođe analiziraju se korelacija, elementi izvan granica i obrada podataka.

U drugom dijelu se obrađuju korišćeni modeli: Logistička regresija, Drvo odlučivanja i *RandomForest* model. Opisuju se i korišćene metrike.

U trećem dijelu se analiziraju i upoređuju rezultati.

2 Analiza i priprema podataka

Skup podataka se sastoji od 10 000 instanci (redova) i 5 kolona (atributa). Ovih 5 atributa su od velikog značaja za cilj ovog rada i oni su ključni za analizu ovog problema. Svi atributi su numerički. Na slici 2 je prikazan učitan skup podataka gdje su atributi:

1. *Index* kolona - jedinstveni identifikacioni broj.
2. *Employed* kolona - označava zaposlenu (1) odnosno nezaposlenu osobu (0).
3. *Bank Balance* kolona - označava novčano stanje osobe na računu u banci.
4. *Annual Salary* kolona - označava godišnji prihode osobe.
5. *Defaulted?* kolona - indikator koji govori da li je osoba izvršila obavezu (1) odnosno nije izvršila obavezu (0) plaćanja kredita.

Unutar podataka ne postoje nedostajuće vrijednosti niti duplirani podaci. Na slici 1 su prikazani osnovni statistički podaci za atribute. Podaci imaju ukupno 10 000 instanci. Srednja vrijednost, standardna devijacija, minimum, prvi kvartil, medijana, treći kvartil i maksimum svakog atributa je prikazan u vrstama *mean, std, min,* 25%, 50%, 75% i *max* respektivno.

	Index	Employed	Bank Balance	Annual Salary	Defaulted?
count	10000.00000	10000.000000	10000.000000	10000.000000	10000.000000
mean	5000.50000	0.705600	10024.498524	402203.782224	0.033300
std	2886.89568	0.455795	5804.579486	160039.674988	0.179428
min	1.00000	0.000000	0.000000	9263.640000	0.000000
25%	2500.75000	0.000000	5780.790000	256085.520000	0.000000
50%	5000.50000	1.000000	9883.620000	414631.740000	0.000000
75%	7500.25000	1.000000	13995.660000	525692.760000	0.000000
max	10000.00000	1.000000	31851.840000	882650.760000	1.000000

Slika 1. Karakteristike skupa podataka.

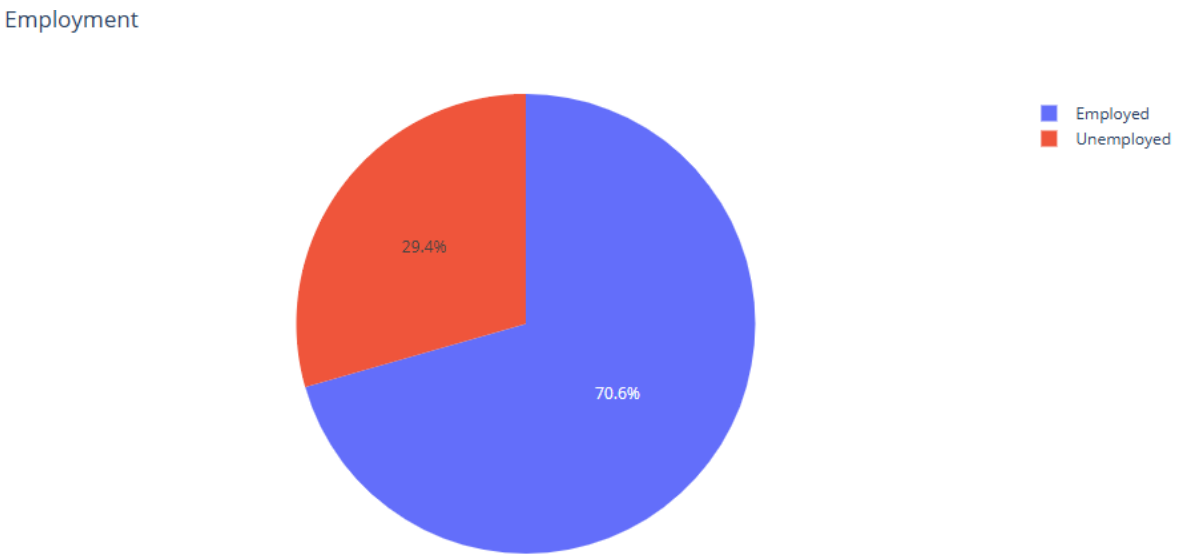
	Index	Employed	Bank Balance	Annual Salary	Defaulted?
0	1	1	8754.36	532339.56	0
1	2	0	9806.16	145273.56	0
2	3	1	12882.60	381205.68	0
3	4	1	6351.00	428453.88	0
4	5	1	9427.92	461562.00	0
...	...	...	...	...	...
9995	9996	1	8538.72	635908.56	0
9996	9997	1	9095.52	235928.64	0
9997	9998	1	10144.92	703633.92	0
9998	9999	1	18828.12	440029.32	0
9999	10000	0	2411.04	202355.40	0

10000 rows × 5 columns

Slika 2. Skup podataka za predikciju plaćanja/neplaćanja kredita.

2.1 Analiza atributa *Employed*

Odnos zaposlenih i nezaposlenih unutar atributa *Employed* je 7056:2944 u korist zaposlenih osoba. Na slici 3 je prikazan procentualni odnos ovih atributa. Ovaj grafik daje dobar uvid u balansiranost klasa. Primjetna je značajna razlika između zaposlenih i nezaposlenih osoba.



Slika 3. Odnos između zaposlenih i nezaposlenih osoba u procentima.

2.2 Analiza atributa *Bank Balance*

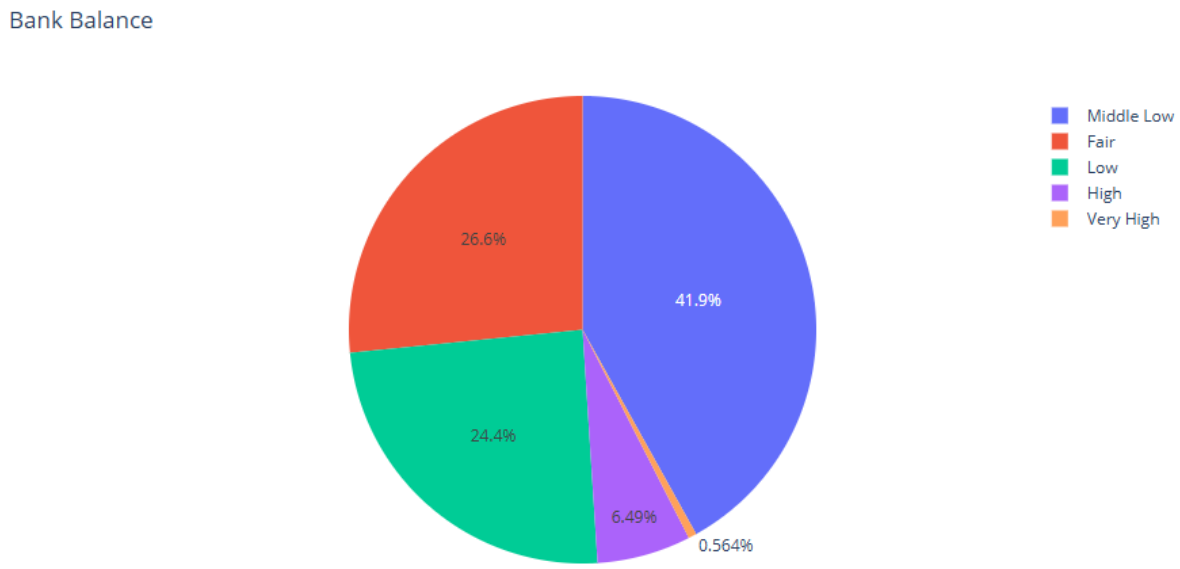
Ukupan broj osoba koje nemaju na računu novca je 499, odnosno u koloni *Bank Balance* imaju iznos 0.0 (slika broj 4).

	Bank Balance	Number
0	0.00	499
1	12382.44	3
2	6273.24	3
3	9278.04	3
4	9324.24	3
...	...	...
9222	1327.92	1
9223	5224.80	1
9224	6465.00	1
9225	12005.04	1
9226	2411.04	1

9227 rows × 2 columns

Slika 4. Brojčani prikaz atributa *Bank Balance* sumiran po istim vrijednostima.

Kako bi se stekao što bolji odnos među ovim vrijednostima uvedeno je pet kategorija koje su dobijene tako što je opseg od najniže do najviše vrijednosti ovog atributa (minimum i maksimum atributa) podjeljen u 5 jednakih kategorija. Ova podjela razmatra najmanju i najveću količinu novca na računu pojedinca a zatim se definišu kategorije. Rezultati ove podjele prikazani su na slici 5. Grafik sadrži 5 imenovanih kategorija i procentualni prikaz svake kategorije. Na grafiku se primjećuje da najveći broj osoba spada u prvu kategoriju (na grafiku obojeni plavom bojom). To su osobe koji imaju najmanju količinu novca ako je poredimo sa osobama koje imaju najveće količine novca na računu (osobe obojane narandžastom bojom).



Slika 5. Grafički prikaz podjele atributa *Bank Balance* na 5 jednakih kategorija.

2.3 Analiza atributa *Annual Salary*

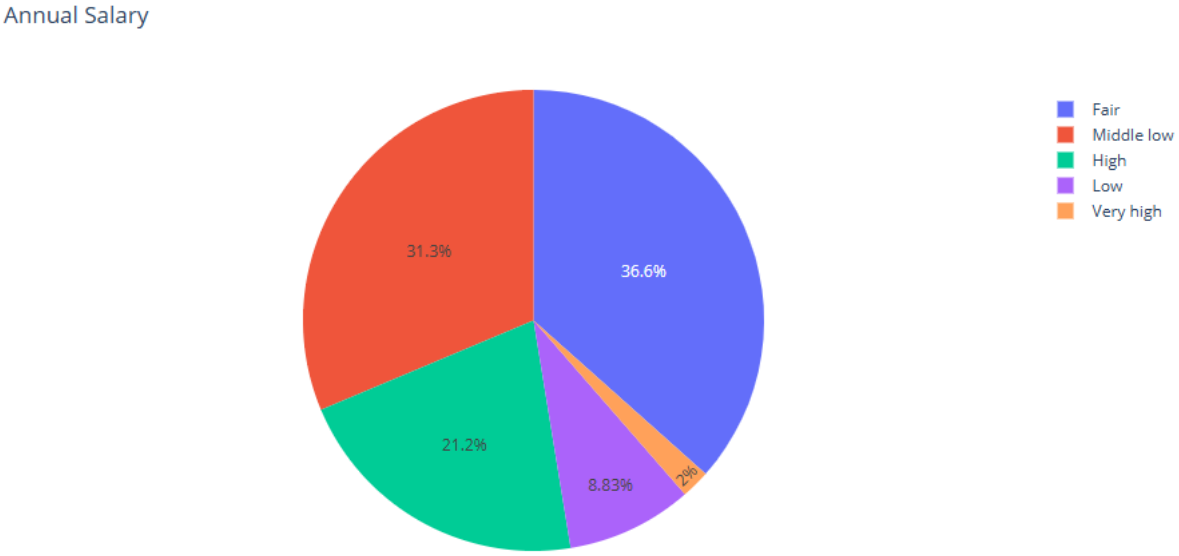
Vrijednosti atributa *Annual Salary* su godišnji prihodi pojedinaca a ove vrijednosti se međusobno razlikuju. Vrijednosti ovog atributa su realni brojevi. Samo 22 osobe imaju istu platu kao i druga osoba ako posmatramo vrijednosti atributa *Annual Salary* kao realne vrijednosti odnosno 93 osobe ako posmatramo vrijednosti ovog atributa kao cjelobrojne. Dakle najveći broj osoba koje imaju istu godišnju platu je dva. Na slici 6 je prikazan ovaj odnos ako posmatramo vrijednosti kao cjelobrojne.

	Annual Salary	Number
0	532339	2
1	186561	2
2	291496	2
3	283617	2
4	486292	2
...	...	...
9902	445722	1
9903	590653	1
9904	179401	1
9905	507982	1
9906	202355	1

9907 rows × 2 columns

Slika 6. Prikaz atributa *Annual Salary* sumirane po istim vrijednostima.

Na slici 7 je grafički prikazana podjela atributa *Annual Salary* u pet kategorija. Podjela kategorija je urađena kao i kod atributa *Bank Balance*.



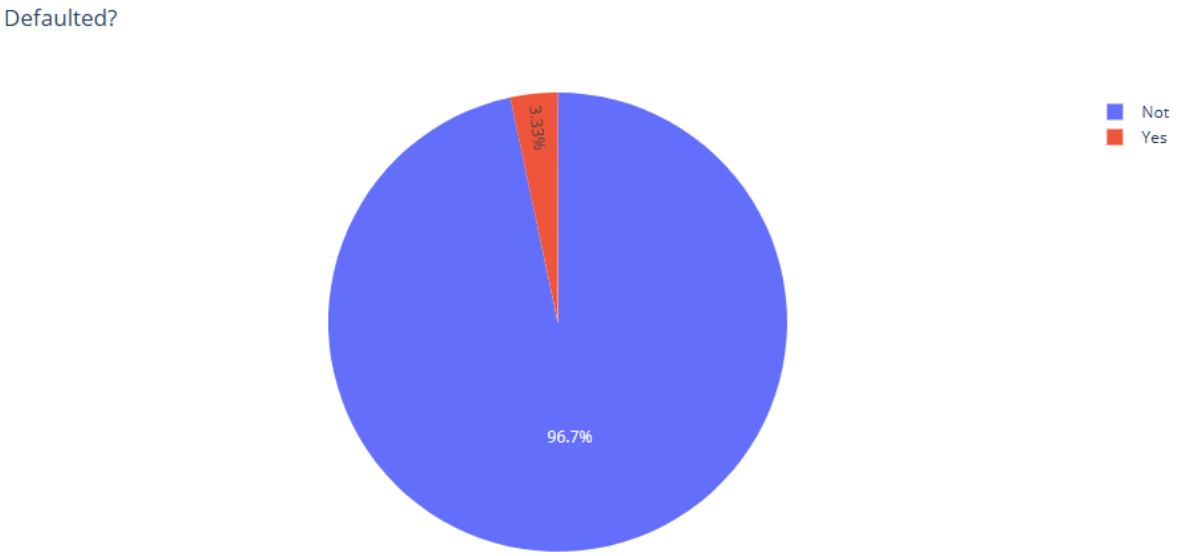
Slika 7. Grafički prikaz atributa *Annual Salary* podjeljennog u pet kategorija.

2.4 Analiza atributa *Defaulted?*

Vrijednosti atributa *Defaulted?* su 0 i 1. 0 označava da osoba nije izvršila izmirenje obaveza kredita dok 1 označava da osoba jeste izvršila izmirenje obaveza kredita. Na slici 8 je prikazana suma ovih klasa a na slici 9 je prikazan grafički odnos ove dvije klase. Primjetna je ogromna nebalansiranost među ovim klasama. Svega 333 osobe od 1000 osoba ima indikator 1 što bi moglo da dovede do preprilagođavanja modela. Ovaj problem će biti razmatran u daljem tekstu.

	Defaulted?	Number
0	Not	9667
1	Yes	333

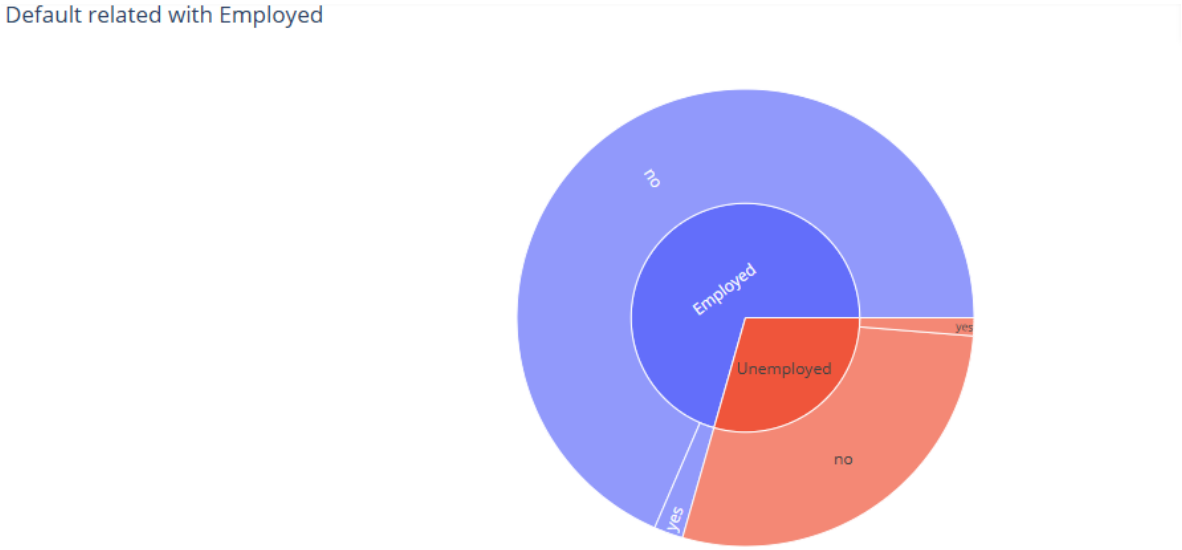
Slika 8. Suma klasa atributa *Defaulted?*.



Slika 9. Grafički prikaz atributa *Defaulted?* sumiranog po klasama.

2.5 Odnos atributa *Defaulted?* i atributa *Employed*

Na slici 10 je prikazan odnos kolone *Defaulted?* i kolone *Employed* pomoću metode *sunburst*. Obzirom na jako mali procenat klase sa oznakom 1 iz kolone *Defaulted?* većina osoba koje su zaposlene i koje nisu zaposlene upadaju u klasu *no* odnosno u klasu koja ne izmiruje obavezu plaćanja kredita. Mali procenat zaposlenih i nezaposlenih upada u klasu *yes* i taj procenat između zaposlenih i nezaposlenih je dosta uravnotežen.



Slika 10. Grafički prikaz odnosa atributa *Defaulted?* i atributa *Employed*.

2.6 Analiza korelacije

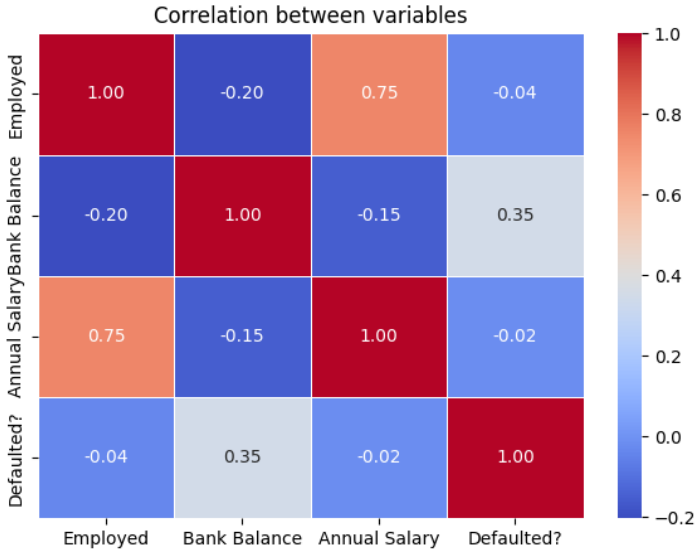
Korelecija ima važnu ulogu prilikom analize atributa. Važno je izdvojiti visoko korelisane atribute kako bi se smanjila preprilagođenost modela [4] Jedan od načina računanja korelacije je:

- Pirsonov koeficijent korelacije [4]

$$r = \frac{\sum(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum(x_i - \bar{x})^2 \sum(y_i - \bar{y})^2}}$$

x_i - uzorci promjenljive x [4],
y_i - uzorci promjenljive y [4],
$\overline{x}$ - srednja vrijednost x varijable [4],
$\overline{y}$ - srednja vrijednost y varijable [4].

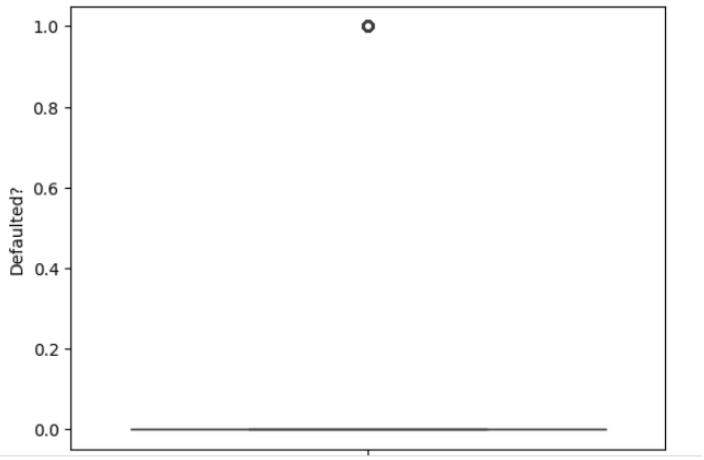
Na slici 11 su prikazane korelacije između atributa. Najveću korelaciju, osim korelacije atributa sa samom sobom, imaju atribut *Annual Salary* i atribut *Employed* i ρ iznosi 0.75. Ostali atributi pokazuju nisku korelaciju sa ostatkom atributa. Obzirom na mali broj atributa, i na same informacije koje prikazuju ovi atributi (važne informacije), ostaju nepromijenjeni.



Slika 11. Grafički prikaz korelacije između atributa.

2.7 Elementi van opsega (*outliers*)

Postoji nekoliko načina za identifikaciju elementa van granice u skupu podataka. Kod ovih podataka postoji jako velika nebalansiranost u ciljnoj klasi. Samim tim moguće tehnike za prepoznavanje elemenata van opsega mogu da lažno detektuju ovakve elemente. Na slici 12 je pomoću *Box plot*-a prikazana raspodjela klasa ciljnog atributa. Osa Y označava atribut *Defaulted?* i većine vrijednosti koncentrisane su oko nula tako da elementi sa vrijednošću jedan bi bile vrijednosti koje su izvan granica.



Slika 12. Grafički prikaz atributa *Defaulted?* pomoću *Box plot*-a.

2.8 Obrada podataka

Ulazni podaci su podjeljeni na podatke za trening i test u odnosu 0.75:0.25. Atribut *Default?* je ciljni atribut na osnovu kojeg se vrši klasifikacija. Izvršena je standardizacija podataka (osim podataka ciljnog atributa) pomoću *MinMaxScaler*-a.

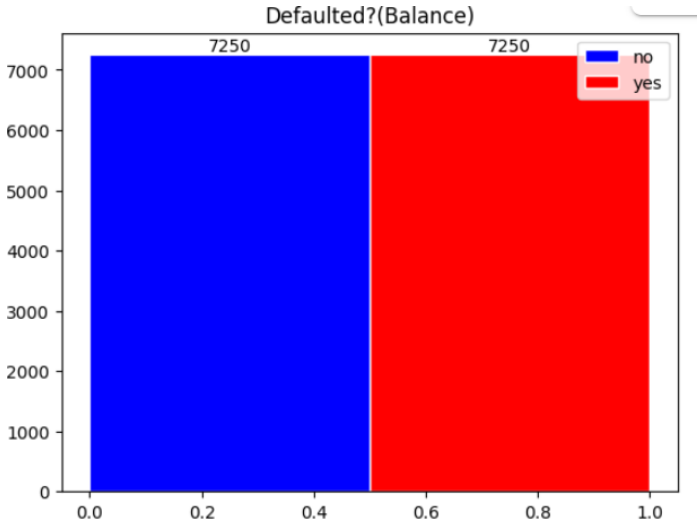
Kao što je rečeno u prethodnom tekstu postoji izražena neuravnoteženost između klasa 0 i 1 kod ciljnog atributa, na osnovu kojeg vršimo klasifikaciju. Ovo je problem dominacije većinske klase zbog čega dolazi do pristrasnosti modela. *SMOTE* (*Synthetic Minority Oversampling Technique*) je tehnika koja rješava problem neuravnoteženosti podataka.

Karakteristike rada *SMOT*-a:

- Identifikacija manjinske klase [7].
- Za svaki uzorak iz ovakve klase, metoda pronalazi nekoliko njegovih najbližih susjeda. Najčešće se koristi euklidska udaljenost [7].
- Na osnovu interpolacije između postojećeg uzorka i njegovog susjeda generišu se novi sintetički uzorci [7].
- Novi uzorci se dodaju u polazni skup podataka [7].

Ovom metodom se dobija povećan broj instanci unutar trening skupa podataka. Trening skup nakon primjene ove metode sadrže 14500 instanci.

Tehnika je primjenjena samo na trening podatke. Pomoću parametra sampling *strategy='minority'* omogućeno je kreiranje vještačkih instanci samo manjinske klase atributa *Default?*. Podrazumjevano je podešeno da se broj instaci manjinske klase dopuni do broja koji je jednak broju instanci većinske klase. Na slici 13 je prikazana balansiraost klasa atributa *Default?* nakon primjene tehnike *SMOTE* kod trening podataka.



Slika 13. Grafički prikaz balansiranosti klasa atibuta *Default?* nakon primjene tehnike *SMOT*

3 Metode i metrike

3.1 Metode

Za rješavanje problema korišćeni su modeli: Logistička regresija (*LogisticRegression*), Stablo odlučivanja (*DecisionTreeClassifier*) i *RandomForestClassifier* model.

3.1.1 Logistička regresija

Logistička regresija je metoda koja se koristi za binarnu klasifikaciju. Koristi Sigmoidnu funkciju koja je odgovorna za postavljanje izlaza na 0 ili 1. Ukoliko je vrijednost sigmoidne funkcije za ulazne parametre veća od 0.5 (prag) onda je izlaz 1 odnosno ako je manji od 0.5 onda je izlaz 0 [2].
Ključne karakteristike Logističke regresije:

1. Nezavisna opažanja: Svako opažanje je nezavisno od drugog opažanja odnosno ne postoji korelacija između bilo koje dvije ulazne promjenljive [2].

2. Binarno zavisne promjenljive: Zavisna promjenljiva mora biti binarna što znači da može da ima samo dvije vrijednosti (0 i 1). *SoftMax* funkcija se koristi za probleme višeklasne klasifikacije [2].

3. Linearni odnos između nezavisnih promjenjivih i logaritamskih vjerovatnoća (*log odds*). Odnos između nezavisnih promjenljivih i logaritamskih vjerovatnoća zavisnih promjenljivih je linearan [2].

Funkcija *LogisticRegression* dostupna je u klasi **sklearn.linear_model**. Parametri ove funkcije prikazani su u sklopu funkcije sa podrazumjevanim vrijednostima parametara:

LogisticRegression(*penalty='l2'*, *, *dual=False*, *tol*=0.0001, *C*=1.0, *fit_intercept=True*, *intercept_scaling* =1, *class_weight=None*, *random_state=None*, *solver='lbfgs'*, *max_iter*=100, *multi_class='deprecated'*, *verbose*=0, *warm_start=False*, *n_jobs=None*, *l1_ratio=None*) [9].

Najvažniji parametar na osnovu kojega će se testiranje vršiti je parametar *solver*. Postoje različite vrste solvera: *lbfgs*, *liblinear*, *newton-cg*, *newton-cholesky*, *sag* i *saga*. Najvažnije karakteristike ovih solvera su:

- *Liblinear* solver je dobar izbor za male skupove podataka, dok su solveri *sag* i *saga* dobri za veće skupove podataka, tj. brži su [9].
- *Liblinear* može obrađivati samo binarnu klasifikaciju. Za sve ostale klasifikacije koristi se, *OneVsRestClassifier* [9].
- *Newtown-cholesky* solver je dobar izbor kada je broj uzoraka *n_samples* znatno veći od *n_features* * *n_classes* (broj atributa * broj klasa). Korišćenje ovog solvera ima kvadratnu zavisnost u memoriji (*n_features * n_classes*), jer eksplicitno računa punu *Hessian* matricu[1] [9].

Za proces testiranja je važan i parametar *penalty* sa vrijednostima: *l1*, *l2*, *elasticnet*, *None*. Najvažnije karakteristike ovog parametra su:

- *None* - ne dodaje kaznene poene (penale) [9].
- *l1, l2, elasticnet* - dodaju kaznene poene [9].

[1] *Hessian* matrica je kvadratna matrica koja sadrži druge parcijalne izvode neke skalarne funkcije. Ona opisuje kako se zakrivljenost funkcije mijenja u različitim pravcima i često se koristi u optimizaciji i analizi funkcija sa više promjenljivih [8].

3.1.2 Stablo odlučivanja

Stablo odluke je struktra koja se koristi za pravljenje odluka (predikcija). Dijelovi stabla odluke su:

- Korjeni čvor (*Root node*) - obuhvata čitav skup podataka i početnu odluku koja treba da se donese.
- Unutrašnji čvorovi (*Internal noodes*) - obuhvata odluke nad atributima. Svaki od ovih čvorova ima jednu ili više grana.
- Grane (*Branches*) - obuhvataju ishod odluke i vode do narednog čvora.
- Listovi (*Leaf Nodes*) - obuhvataju konačnu odluku (predikciju). Od ovog čvora se dalje ne vrše podjele.

Koraci prilikom kreiranja stabla odluke:
1. Izbor najboljeg atributa pomoću metodologija odabira [3].
2. Podjela skupa podataka na osnovu izabranog atributa [3].
3. Ponavljanje procesa na osnovu kojih se dodaju novi unutrašnji čvorovi ili listovi, sve dok se ne zadovolji kriterijum za zaustavljanje [3].

Metodologije za odabir podjela:

- Gini nečistoća (*Gini Impurity*)

$$\text{Gini} = 1 - \sum_{i=1}^{n} (p_i)^2$$

  p_i - vjerovatnoća da instanca pripada određenoj klasi [3].

- Entropija (*Entropy*)

$$\text{Entropy} = -\sum_{i=1}^{n} p_i \log(p_i)$$

  p_i - vjerovatnoća da instanca pripada određenoj klasi [3].

- Dobitak informacija (*Information Gain*)

$$\text{Information Gain} = Entropy_{parent} - \sum_{i=1}^{n} (\frac{|D_i|}{|D|} \cdot Entropy(D_i))$$

  D_i je podskup od D nakon podjele po atrbutu [3].

Funkcija *DecisionTreeClassifier* je dostupna u klasi ***sklearn.tree*** [10]. Parametri i njihove podrazumjevane vrijednosti dati su u sklopu funkcije:

DecisionTreeClassifier(*, *criterion='gini'*, *splitter='best'*, *max_depth=None*, *min_samples_split=2*, *min_samples_leaf*=1, *min_weight_fraction_leaf*=0.0, *max_features=None*, *random_state=None*, *max_leaf_nodes=None*, *min_impurity_decrease*=0.0, *class_weight=None*, *ccp_alpha*=0.0, *monotonic_cst=None*) [10].

Najvažniji parametar pomoću kojeg se vrši testiranje je *criterion* sa vrijednostima *gini, entropy* i *log_ loss*. Osobine ovog parametra su:

- *Gini* – za Gini nečistoću (*Gini impurity*) [10].
- *Log_ loss* i *entropy* – za Šenonov informacioni dobitak (*Shannon information gain*) [10].

Ostali parametri su:

- *max_ depth* - predavlja maksimalnu dubinu stabla. Za vrijednost *None* čvorovi se proširuju sve dok svi listovi ne postanu potpuno čisti ili dok svi listovi ne sadrže manje od *min_ samples_ split* uzoraka [10].
- *min_ samples_ split* - ovaj parametar može da ima *int* ili *float* vrijednost. Ako je vrijednost *int*, tada se *min_ samples_ split* smatra minimalnim brojem uzoraka a ako je *float* tada je *min_ samples_ split* udio od ukupnog broja uzoraka, pri čemu je minimalan broj uzoraka za svaku podjelu *ceil(min_ samples_ split * n_ samples)* [10].
- *min_ samples_ leaf* - predstavlja minimalan broj uzoraka potreban da bi čvor bio list. Tačka podjele na bilo kojoj dubini biće uzeta u obzir samo ako ostavi bar *min_ samples_ leaf* uzoraka za treniranje u svakoj od lijevih i desnih grana. Ako je vrijednost *int*, tada se *min_ samples_ leaf* smatra minimalnim brojem uzoraka a ako je *float*, tada je *min_ samples_ leaf* udio od ukupnog broja uzoraka, pri čemu je minimalan broj uzoraka za svaki čvor *ceil(min_ samples_ leaf * n_ samples)* [10].

3.1.3 *RandomForest* model

RandomForest algoritam je tehnika koja se zasniva na treniranju modela pomoću više stabala odluke. Svako stablo se kreira koristeći proizvoljan podskup podataka. Ovim se uvodi promjenljivost među stablima čime se smanjuje preprilagođavanje a sami tim se poboljšavaju performanse modela [6].

Koraci prilikom izvođenja modela su:
1. Skup stabala odluke - svako stablo se zasebno specijalizuje za odgovarajući skup podataka. Stabla rade nezavisno. Dakle nema uticaja stabla na druga stabla [6].

2. Nasumičan odabir karakteristika - ovim se omogućava da stabla imaju fokus na različite aspekte podataka [6].

3. Bootstrap agregacija ili "*bagging*" - predstavlja temelj ovog algoritma. Podrazumjeva kreiranje više uzoraka korišćenjem metode *bootstrap*-a iz polaznog skupa podataka, pri čemu se uzorci uzimaju sa ponavljanjem. Ovim se uvodi varijabilnost u proces treniranja i povećava se otpornost modela [6].

4. Donošenje odluke i glasanje - u procesu predikcije, svako stablo odluke daje svoj glas. Kod klasifikacije, konačna predikcija se određuje na osnovu najčešće predikcije koju daju sva stabla [6].

Funkcija *RandomForestClassifier* je dostuplna u klasi ***sklearn.ensemble*** [11]. Parametri i njihove podrazumjevane vrijednosti su dati u sklopu funkcije:

*RandomForestClassifier(n_estimators=100, *, criterion='gini', max_depth=None, min_samples_split=2, min_samples_leaf=1, min_weight_fraction_leaf=0.0, max_features='sqrt', max_leaf_nodes=None, min_impurity_decrease=0.0, bootstrap=True, oob_score=False, n_jobs=None, random_state=None, verbose=0, warm_start=False, class_weight=None, ccp_alpha=0.0, max_samples=None, monotonic_cst=None)* [11].

Najvažniji parametar pomoću kojeg se vrši testiranje je paramatar *criterion* koji ima vrijednosti *gini, entropy* i *log_loss*. Karakteristike vrijednosti ovog parametra identično je kao i kod modela Drveta odlučivanja [11].

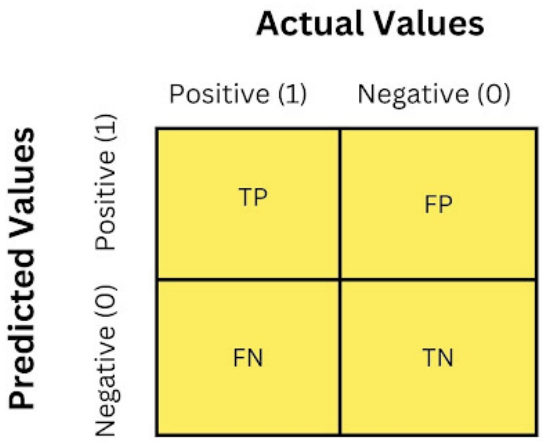
Drugi važan parametar je *n_estimators* koji označava broj drveća u čitavom skupu drveća [11].

3.2 Metrike

Za testiranje modela korišćene su metrike: *accuracy, precision, recall* i *f1 score*. Da bi se ove metrike objasnile važno je uvesti pojam matrice konfuzije.

3.2.1 Matrica konfuzije (*confussion matrix*)

Matrica konfuzije je tabela koja ocjenjuje performanse klasifikacionog algoritma. Ova matrica koristi ciljne vrijednosti za poređenje sa vrijednostima koje je predvidio algoritam. Svaki red u matrici predstavlja instance u predviđenoj klasi, dok svaka kolona predstavlja instance u stvarnoj klasi, ili obrnuto [12]. Na slici 14 je prikazana matrica konfuzije.



Slika 14. Matrica konfuzije [13][2]

3.2.2 Tačnost (*Accuracy*)

Accuracy se definiše kao odnos broja tačno predviđenih pozitivnih i negativnih primjeraka (TP + TN) prema ukupnom broju pozitivnih i negativnih posmatranja, odnosno tačnost nam govori koliko često možemo očekivati da će naš model ispravno predvidjeti ishod u odnosu na ukupan broj predikcija koje je napravio [12].

Formula za računanje tačnosti je:

$$Accuracy = \frac{TP + TN}{TP + FN + TN + FP} [12].$$

3.2.3 Preciznost (*Precision*)

Precision mjeri proporciju pozitivno predviđenih oznaka koje su zaista tačne. Preciznost je poznata i kao pozitivna prediktivna vrijednost. Ova metrika se koristi zajedno sa odzivom (*recall*) kako bi se balansirali lažno pozitivni i lažno negativni rezultati. Preciznost "kažnjava" lažno pozitivne slučajeve (kada model pogrešno predviđa uzorak kao pozitivan). Ako želimo da minimizujemo lažno pozitivne rezultate, koristimo preciznost kao metriku modela [12].

[2] TP(*true positive*), TN(*true negative*), FP(*false positive*), FN(*false negative*).

Formula za računanje preciznosti je:

$$Precision = \frac{\text{TP}}{\text{FP} + \text{TP}} [12].$$

3.2.4 Odziv (*Recall*)

Recall modela klasifikacije predstavlja sposobnost modela da tačno predvidi pozitivne slučajeve među stvarno pozitivnim. Ovo se razlikuje od preciznosti, koja mjeri koliko su predikcije modela za pozitivne klase tačne u odnosu na sve pozitivne predikcije koje je model napravio. Ako model pokušava da prepozna pozitivne slučajeve, *recall* pokazuje koliki procenat tih pozitivnih slučajeva je model ispravno predvidio kao pozitivne. *Recall* "kažnjava" lažno negativne predikcije tako što smanjuje skor kada model pogrešno klasifikuje pozitivnu klasu kao negativnu [12].

Formula za računanje odziva je:

$$Recall = \frac{\text{TP}}{\text{FN} + \text{TP}} [12].$$

3.2.5 *F1 score*

F1 score je harmonijska sredina preciznosti i odziva. Izračunava se pomoću preciznosti i odziva. Daje jednaku težinu i preciznosti i odzivu pri mjerenju performansi modela u smislu tačnosti, čime je alternativa za metriku *Accuracy*, jer ne zahtjeva poznavanje ukupnog broja posmatranja [12].

Formula za računanje *F1 score*-a je:

$$F1 \quad Score = 2 \times Precision \times Recall/(Precision + Recall) [12].$$

4 Rezultati

4.1 Logistička regresija

U ovom poglavlju testirani su različiti parametri. Testirani su solveri *lbfgs*, *liblinear*, *newton-cg*, *newton-cholesky*, *sag* i *saga*. Takođe je testiran i parametar *penalty* za *l2* i *None* vrijednosti. Najbolji parametri su *lbfgs* solver i *l2 penalty*. Testirani su pojedinačni solveri a svi rezultati za različite solvere i najbolji model (model sa najboljim parametrima) su prikazani u tabelama. U tabeli 1 su prikazani modeli koji koriste trening podatke a u tabeli 2 su prikazani modeli koji koriste test podatke.

Model	*Accuracy*	*Precision*(0 klasa-1 klasa)	*Recall*(0 klasa-1 klasa)	*F1 score*(0 klasa-1 klasa)
Najbolji	0.89	0.88 (0.9-0.88)	0.91 (0.87-0.91)	0.89 (0.89-0.89)
Lbfgs	0.88	0.88 (0.9-0.88)	0.91 (0.87-0.91)	0.89 (0.89-0.89)
Liblinear	0.89	0.87 (0.9-0.88)	0.91 (0.87-0.91)	0.89 (0.89-0.89)
Newton cg	0.89	0.88 (0.9-0.88)	0.91 (0.87-0.91)	0.89 (0.89-0.89)
Newton cholesky	0.89	0.88 (0.9-0.88)	0.91 (0.87-0.91)	0.89 (0.89-0.89)
Sag	0.89	0.88 (0.9-0.88)	0.91 (0.87-0.91)	0.89 (0.89-0.89)
Saga	0.89	0.88 (0.9-0.88)	0.91 (0.87-0.91)	0.89 (0.89-0.89)

Tabela 1. Poređenje rezultata različitih parametara za model Logističke regresije nad trening podacima.

Model	*Accuracy*	*Precision*(0 klasa-1 klasa)	*Recall*(0 klasa-1 klasa)	*F1 score*(0 klasa-1 klasa)
Najbolji	0.87	0.19 (1.0-0.19)	0.88 (0.87-0.88)	0.32 (0.93-0.32)
Lbfgs	0.87	0.19 (1.0-0.19)	0.88 (0.87-0.88)	0.31 (0.93-0.32)
Liblinear	0.87	0.19 (1.0-0.19)	0.88 (0.87-0.88)	0.31 (0.93-0.31)
Newton cg	0.87	0.19 (1.0-0.19)	0.88 (0.87-0.88)	0.31 (0.93-0.32)
Newton cholesky	0.87	0.19 (1.0-0.19)	0.91 (0.87-0.88)	0.31 (0.93-0.32)
Sag	0.87	0.19 (1.0-0.19)	0.88 (0.87-0.88)	0.31 (0.93-0.32)
Saga	0.87	0.19 (1.0-0.19)	0.88 (0.87-0.88)	0.31 (0.93-0.32)

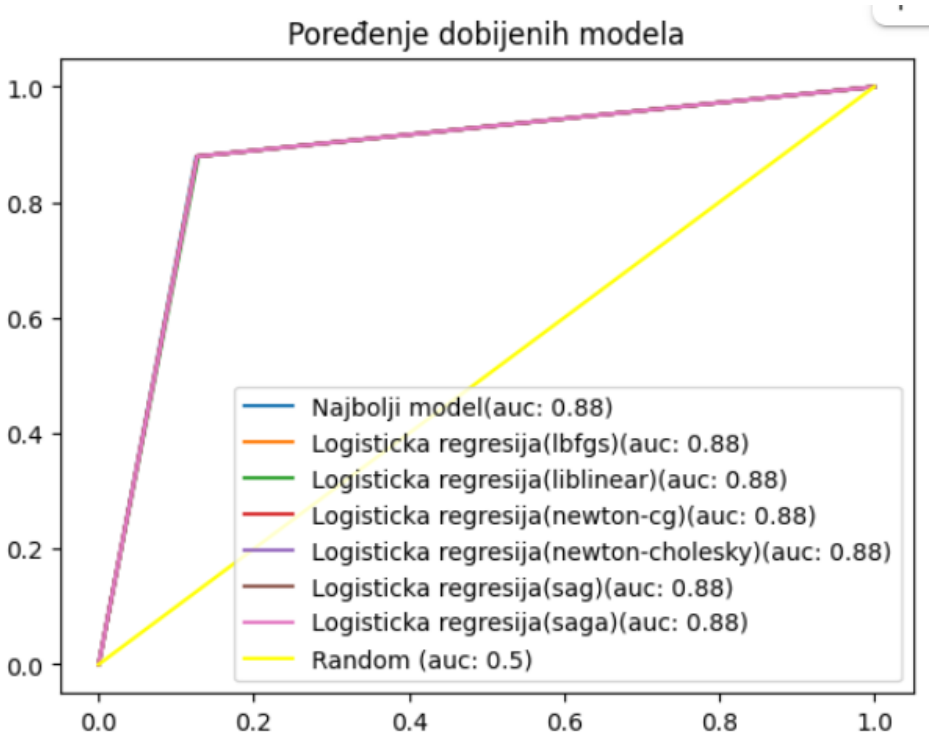
Tabela 2. Poređenje rezultata različitih parametara za model Logističke regresije nad test podacima.

Model	TP(train-test)	FP(train-test)	FN(train-test)	TN(train-test)
Najbolji	6318-2110	932-307	681-10	6569-73
Lbfgs	6318-2111	932-306	681-10	6569-73
Liblinear	6314-2106	936-311	680-10	6570-73
Newton cg	6318-2110	932-307	680-10	6570-73
Newton cholesky	6318-2110	932-307	681-10	6569-73
Sag	6318-2110	932-307	679-10	6571-73
Saga	6318-2110	932-307	681-10	6569-73

Tabela 3. Poređenje rezultata matrice kofuzije različitih parametara za model Logističke regresije nad trening i test podacima.

U narednim razmatranjima rezultata tumačiće se isključivo dobijeni rezultati nad test podacima. Svi parametri (*accuracy, precision, recall, f1 score*) kod modela nad trening podacima daju dobre rezultate.
Slijede rezultati za test skup podataka.

Na osnovu dobijenih rezultata svi modeli daju visoku tačnost što znači da je model dobro klasifikovao veliki procenat uzoraka, tj. da je većina predikcija tačna. Ukupna preciznost svih modela je ista i pokazuje nisku vrijednost. Ako analiziramo posebno preciznost po klasama vidimo da je preciznost za klasu 1 jako niska. To znači da model često klasifikuje pripadnike druge klase kao pripadnika odgovarajuće klase što dovodi do velikog broja lažno pozitivnih slučajeva(FP). Ovo je dovelo do niske vrijednosti ukupne preciznosti. *Recall* kod ovih modela je dosta dobar što znači da modeli dosta dobro klasifikuju stvarne pozitivne slučajeve (TP). Na osnovu podataka za preciznost i *recall* očekivan je i lošiji *f1 score*. To znači da modeli ne balansiraju dobro između preciznosti i *recall*-a.



Slika 15. Poređenje dobijenih modela nad test podacima pomoću *ROC* krivih.

Na slici 15 prikazano je poređenje dobijenih modela nad test podacima pomoću *ROC* krivih. Dobijeni *auc score* je isti za sve modele i iznosi 0.88. Ovaj rezultat je dosta blizu visokom rezultatu (*auc score* od 0.9 na dalje je visok) što znači da model ima približno dobru sposobnost da razlikuje pozitivne od negativnih klasa.

4.2 Drvo odlučivanja

Model Drvo odlučivanja je testirano nad različitim parametrima. Za *criterion* parametar razmatrani su: *gini*, *entropy* i *log loss*. Za *max depth* kriterijum razmatrane su vrijednosti 2, 4, 6, 8, 10, 12, 14, 16, 18 i 20. Za kriterijum *min samples split* razmatrane su vrijednosti 2, 3, 4, 5 i 6 a za kriterijum *min samples leaf* vrijednosti 2, 3, 4 5, 6 i 7. Testirani su i pojedinačni kriterijumi. Najbolji model ima parametre *criterion*: *gini*, *max depth*: 14, *min samples leaf*: 6, *min samples split*: 4.

Model	*Accuracy*	*Precision*(0 klasa-1 klasa)	*Recall*(0 klasa-1 klasa)	*F1 score*(0 klasa-1 klasa)
Najbolji	0.94	0.94 (0.95-0.94)	0.95 (0.94-0.95)	0.94 (0.94-0.94)
Entropy	1.0	1.0 (1.0-1.0)	1.0 (1.0-1.0)	1.0 (1.0-1.0)
Gini	1.0	1.0 (1.0-1.0)	1.0 (1.0-1.0)	1.0 (1.0-1.0)
Log-loss	1.0	1.0 (1.0-1.0)	1.0 (1.0-1.0)	1.0 (1.0-1.0)

Tabela 4. Poređenje rezultata različitih parametara za model Drvo odlučivanja nad trening podacima.

Model	*Accuracy*	*Precision*(0 klasa-1 klasa)	*Recall*(0 klasa-1 klasa)	*F1 score*(0 klasa-1 klasa)
Najbolji	0.9	0.2 (0.99-0.2)	0.66 (0.91-0.66)	0.3 (0.95-0.31)
Entropy	0.88	0.18 (0.99-0.17)	0.7 (0.89-0.67)	0.28 (0.94-0.28)
Gini	0.88	0.17 (0.98-0.17)	0.64 (0.9-0.6)	0.27 (0.94-0.26)
Log-loss	0.88	0.17 (0.99-0.18)	0.66 (0.89-0.69)	0.27 (0.94-0.28)

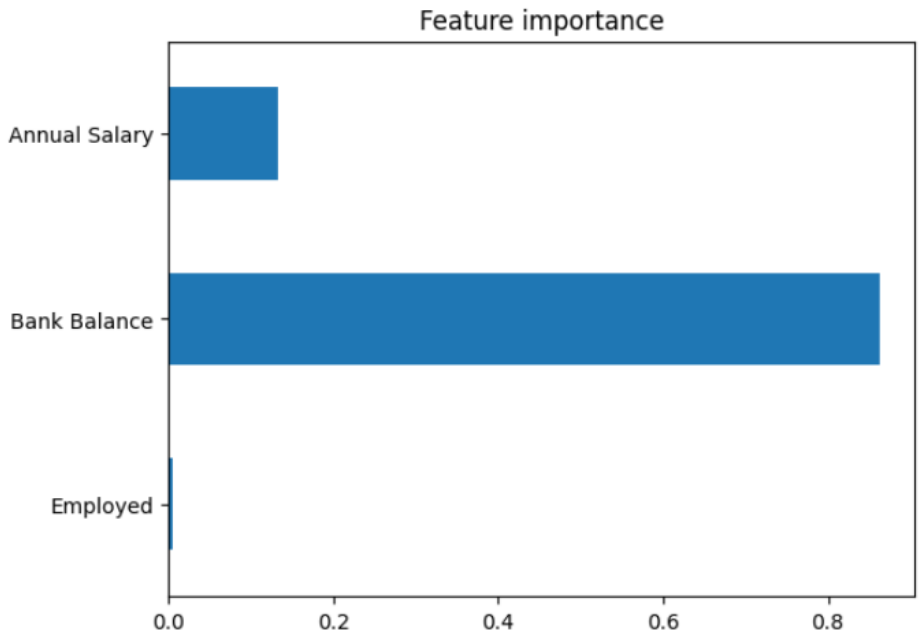
Tabela 5. Poređenje rezultata različitih parametara za model Drvo odlučivanja nad test podacima.

Model	TP(train-test)	FP(train-test)	FN(train-test)	TN(train-test)
najbolji	6803-2200	447-217	372-28	6878-55
entropy	7250-2153	0-264	0-27	7250-56
gini	7250-2167	0-250	0-33	7250-50
log-loss	7250-2149	0-268	0-26	7250-57

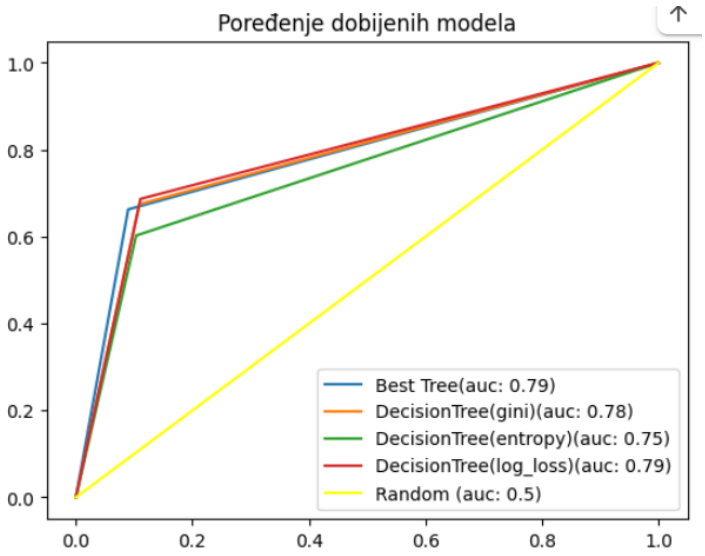
Tabela 6. Poređenje rezultata matrice kofuzije različitih parametara za model Drvo odlučivanja nad trening i test podacima.

Tačnost modela je prilično visoka. Kao i kog Logističke regresije postoji niska preciznost. *Recall* kod ovih modela ima srednju vrijednost. Ovo ukazuje na to da modeli imaju problema sa prepoznavanjem određenih klasa, posebno onih koje su slabije zastupljene u podacima a to je u ovom slučaju klasa 1. *F1 score* ima dosta nisku vrijednost što je očekivano zbog nižih vrijednosti preciznosti i *recall*-a.

Na slici 16 je grafički prikazan odnos između važnosti atributa koji su učestvovali u izgradnji modela Drveta odlučivanja za najbolje parametre. Na slici se vidi da je atribut *Bank balance* imao presudnu važnost na donošenje odluka unutar stabla.



Slika 16. Grafik važnosti atributa za drvo odlučivanja za model sa najboljim parametrima.



Slika 17. Poređenje dobijenih modela pomoću *ROC* krivih nad test podacima.

Na slici 17 je prikazano poređenje dobijenih modela pomoću *ROC* krivih nad test podacima za različite kriterijume. Dobijeni *auc score* za svaki model je različit. Svi rezultati imaju srednju vrijednost. Najvišu vrijednost ima najbolji model i model sa kriterijumom *log loss* i iznosi 0.79. To znači da modeli imaju nižu sposobnost da razlikuju klase.

4.3 *RandomForest* model

Kod modela *RandomForest* testrani su različiti parametri. Za *criterion* parametar korišćeni su *gini*, *entropy* i *log loss* a za parametar *n_ estimators* korišćene su vrijednosti 90, 100 i 120. Testirani su i različiti kriterijumi. Utvrđen je najbolji model sa parametrima *criterion*: *log loss* i *n_ estimators* za vrijednost 100.

Model	*Accuracy*	*Precision*(0 klasa-1 klasa)	*Recall*(0 klasa-1 klasa)	*F1 score*(0 klasa-1 klasa)
Najbolji	1.0	1.0 (1.0-1.0)	1.0 (1.0-1.0)	1.0 (1.0-1.0)
Gini	1.0	1.0 (1.0-1.0)	1.0 (1.0-1.0)	1.0 (1.0-1.0)
Log-loss	1.0	1.0 (1.0-1.0)	1.0 (1.0-1.0)	1.0 (1.0-1.0)
Entropy	1.0	1.0 (1.0-1.0)	1.0 (1.0-1.0)	1.0 (1.0-1.0)

Tabela 7. Poređenje rezultata različitih parametara za model *Random Forest* nad trening podacima.

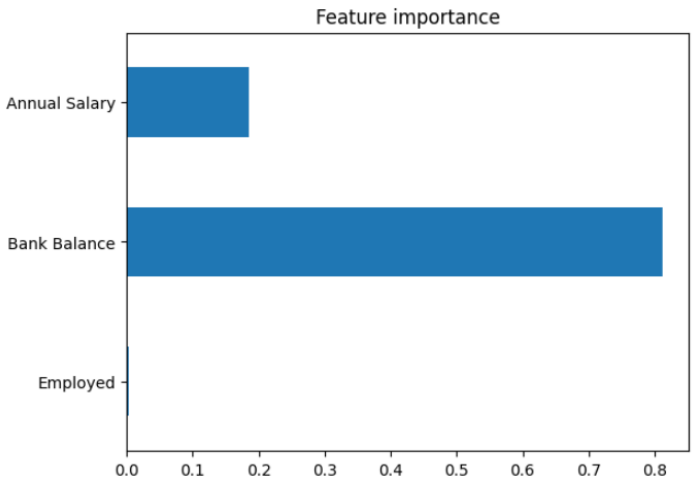
Model	*Accuracy*	*Precision*(0 klasa-1 klasa)	*Recall*(0 klasa-1 klasa)	*F1 score*(0 klasa-1 klasa)
Najbolji	0.9	0.21 (0.99-0.21)	0.71 (0.91-0.71)	0.33 (0.95-0.33)
Gini	0.9	0.22 (0.99-0.22)	0.72 (0.91-0.72)	0.34 (0.95-0.34)
Log-loss	0.9	0.22 (0.99-0.22)	0.71 (0.91-0.71)	0.33 (0.95-0.33)
Entropy	0.9	0.21 (0.99-0.21)	0.7 (0.91-0.7)	0.33 (0.95-0.33)

Tabela 8. Poređenje rezultata različitih parametara za model *Random Forest* nad test podacima.

Model	TP(train-test)	FP(train-test)	FN(train-test)	TN(train-test)
Najbolji	7250-2201	0-216	0-24	7250-59
Gini	7250-2208	0-209	0-23	7250-60
Log-loss	7250-2203	0-214	0-24	7250-59
Entropy	7250-2202	0-215	0-25	7250-58

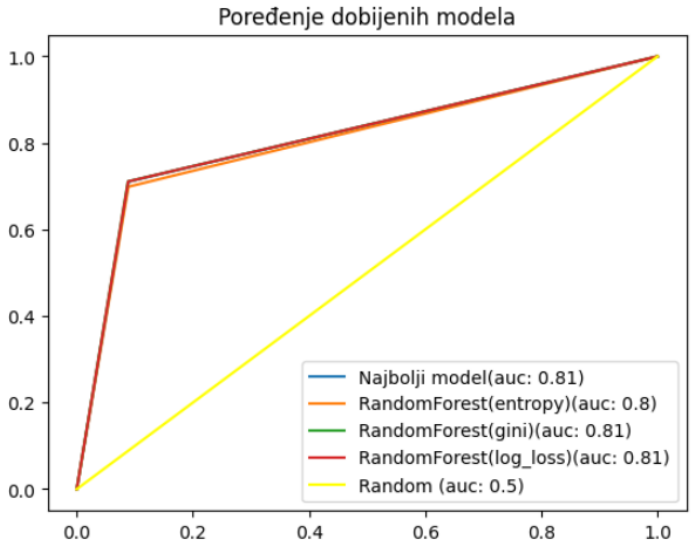
Tabela 9. Poređenje rezultata matrice kofuzije različitih parametara za model *RandomForest* nad trening i test podacima.

Na osnovu rezultata priloženih u tabelama vidimo da svi modeli daju dobru tačnost. Preciznost je loša što je rezultat velikog broja lažno pozitivnih klasifikovanja. *Recall* ima srednju vrijednost i otprilike je iste vrijednosti za sve poređene modele. *F1 score* ima nisku vrijednost što opet ukazuje na loš odnos između preciznosti i *recall* parametra.



Slika 18. Grafik važnosti atributa za *RandomForest* model sa najboljim parametrima.

Na slici 18 je prikazan odnos učestvovanja atributa u kreiranju modela gdje vidimo da opet atribut *Bank Balance* ima glavni uticaj.



Slika 19. Poređenje dobijenih modela nad test podacima.

Na slici 19 je grafički prikazan *auc score* pomoću *ROC* krivih. Svi modeli imaju jednak *auc score* i iznosi 0.8.

4.4 Poređenje sa dostupnim rezultatima

Na slikama su prikazani rezultati dobijeni u relevantnom radu. Rad je dostupan na linku: **https://www.kaggle.com/code/douglasparis/modeling-defaults-with-different-models-95-f1**.

Na slikama 20, 21 i 22 su prikazani rezultati za metrike *Accuracy, Auc* i *F1 score* respektivno. U ovom radu su rađene različite standardizacije podataka i njihov nazivi su prikazani u prvoj koloni. Takođe su korišćeni i različiti modeli za testiranje. Dakle, za ovo poređenje su od značaja prva, treća i četvrta kolona u presjeku sa drugom vrstom.

	LogReg	SVM	DecTree	RandFor	Bayes	KNN	MLP
Standard	0.972727	0.970606	0.95697	0.971515	0.968182	0.971515	0.9733
MinMax	0.970606	0.97	0.956667	0.971212	0.968182	0.971212	0.9730
MaxAbs	0.970606	0.97	0.956667	0.971818	0.968182	0.971818	0.9733
RobustAbs	0.972727	0.970606	0.95697	0.970909	0.968182	0.970606	0.9730
Yeo	0.97303	0.969394	0.955455	0.970909	0.966061	0.971515	0.9733

Slika 20. Rezultati za *Accuracy* metriku

	LogReg	SVM	DecTree	RandFor	Bayes	KNN	MLP
Standard	0.662826	0.592809	0.68913	0.675121	0.630321	0.692351	0.6803
MinMax	0.592809	0.579573	0.680358	0.679272	0.630321	0.687886	0.6629
MaxAbs	0.592809	0.579573	0.688973	0.679585	0.630321	0.696815	0.6889
RobustAbs	0.662826	0.592809	0.684823	0.679115	0.630321	0.678958	0.6715
Yeo	0.658675	0.570644	0.688346	0.674807	0.5	0.679428	0.6760

Slika 21. Rezultati za *Auc* metriku.

	LogReg	SVM	DecTree	RandFor	Bayes	KNN	MLP
Standard	0.45122	0.302158	0.387931	0.45977	0.363636	0.483516	0.4823
MinMax	0.302158	0.266667	0.375546	0.463277	0.363636	0.475138	0.4539
MaxAbs	0.302158	0.266667	0.386266	0.468571	0.363636	0.491803	0.4942
RobustAbs	0.45122	0.302158	0.382609	0.460674	0.363636	0.458101	0.4670
Yeo	0.447205	0.240602	0.379747	0.454545	0.0	0.465909	0.4761

Slika 22. Rezultati za *F1 score* metriku.

Poređenja rezultata ovog rada sa dobijenim rezultatima su sljedeća:

- Rezultati za *accuracy*, za modele Drvo odlučivanja i *Random Forest*, su vrlo slična dok je za model Logističke regresije blago niži u odnosu na rezultate u dostupnom radu.
- Rezultati za *auc* za sva tri modela su značajno viša u odnosu na rezultate u pomenutom radu.
- Rezultati za *F1 score*, za model Logističke regresije je značajno viši a za modele Drvo odlučivanja i *Random Forest* su blago niža u odnosu na rezultate u pomenutom radu.

5 Zaključak

Tabela 10 prikazuje mdele sa najboljim parametrima za svaki korišćeni model.

Model	*Accuracy*	*Precision*(0 klasa-1 klasa)	*Recall*(0 klasa-1 klasa)	*F1 score*(0 klasa-1 klasa)
Logistička regresija	0.87	0.19 (1.0-0.19)	0.88 (0.87-0.88)	0.31 (0.93|0.32)
Drvo odučivanja	0.9	0.2 (0.99-0.2)	0.66 (0.91-0.66)	0.3 (0.95-0.31)
RandomForest model	0.9	0.21 (0.99-0.21)	0.71 (0.91-0.71)	0.33 (0.95-0.33)

Tabela 10. Poređenje rezultata modela sa najboljim parametrima nad test podacima.

Na osnovu rezultata svi modeli daju podjednake rezutate.

- Blagu prednost za tačnost i preciznost imaju modeli Drvo odlučivanja i *RandomForest* dok za *recall* model Logističke regresije ima blagu prednost.
- Najbolji *auc score* ima model Logističke regresije.

Obzirom na važnosti ovakvih modela u današnjici ovi modeli pokazuju dobar materijal za dalju nadogradnju u ovoj oblasti.

References

[1] Kamal Das , *Loan Default Prediction*, 2020.

[2] *Geeksforgeeks*, *Logistic Regression*, Jun,2024.

[3] *Geeksforgeeks*, *Decision Tree*, Maj,2024.

[4] Abdallah Ashraf, *Correlation in machine learning*, Sep,2023.

[5] Abdallah Ashraf, *Z-Score: Meaning and Formula*, Apr,2024.

[6] Geeksforgeeks, *Correlation in machine learning*, Dec,2024.

[7] Cory Maklin, *Synthetic Minority Over-sampling Technique (SMOTE)*, Maj,2022

[8] *Wikipedia*, *Hessian matrix*.

[9] *scikit-learn*, *LogisticRegression*.

[10] *scikit-learn*, *DecisionTreeClassifier*.

[11] *scikit-learn*, *RandomForestClassifier*.

[12] Ajitesh Kumar, *Accuracy, Precision, Recall F1-Score – Python Examples*, Avgust,2024.

[13] Wiki, *What is a Confusion Matrix?*, Avgust,2024.