SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU

FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I INFORMACIJSKIH TEHNOLOGIJA OSIJEK

Sveučilišni studij

OTKRIVANJE PRIJEVARA S KREDITNIM KARTICAMA

Projektni zadatak

Tomislav Barbarić

Osijek, 2023.

Sadržaj

1. UVOD

Cilj projektnog zadatka je usporediti dostupne modele za klasifikaciju prijevara kreditnim karticama. Projekt je pisan jezikom Python dok je podatkovni skup preuzet sa Kaggle repozitorija. Prijevare kreditnim karticama sve su češća pojava razvojem tehnologija. Načina prevare ima mnogo, od krađe kreditnih kartica do prikupljanja podatak raznim phishing metodama. Bankama je prioritet predvidjeti takve transakcije i na vrijeme ih spriječiti. Pri tome koriste razne metode poput stvaranja profila korisnika, računanja ocjene prijevare i umjetnu inteligenciju za predviđanje prijevara. U ovome projektu usporedit će se četiri modela koja konceptualno prikazuju način detekcije prijevara.

2. OPIS PODATKOVNOG SKUPA I PREDOBRADA PODATAKA

2.1. Opis problematike

Otkrivanje prijevara sa kreditnim karticama spada u binarni klasifikacijski problem. Zadatak problema je klasificirati elemente skupa u dvije grupe s obzirom na određeno pravilo klasifikacije.

Modeli klasifikacije su izgrađeni metodama nadziranog učenja. Dostupni labelirani podatci mu se predaju i na temelju njih klasificira ih na 0 ili 1 odnosno stvarnu ili lažnu transakciju.

2.2. Opis dostupnog skupa podataka

Za treniranje modela potrebno je imati određene podatke sa što većim brojem instanci. Skup podataka korišten u ovome radu sastoji se od 2840000 bankovnih transakcija i 31 značajke. Dostupan je na Kaggle repozitoriju[1].

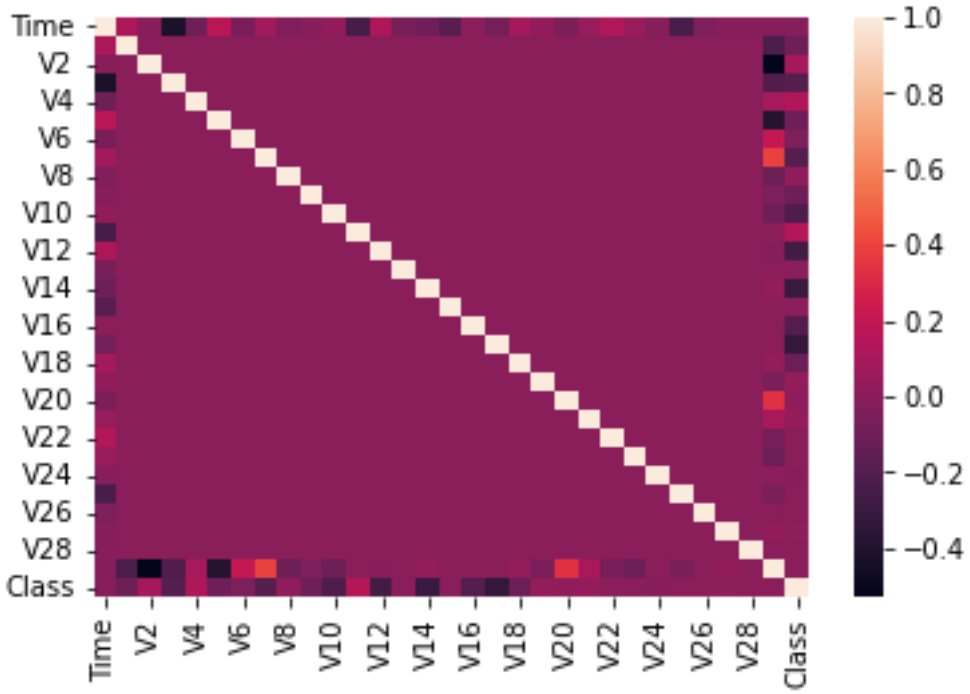
Tablica 1. Prikaz skupa podataka

Time	V1	V2	…	V28	Amount	Class
0	-1.359807	-0.072781	…	-0.021053	149.62	0
1	-0.966272	-0.185226	…	0.061458	123.50	0
2	-1.158233	0.877737	…	0.215153	69.99	0

Tablica 1. prikazuje podatke i značajke koji se nalaze u skupu podataka. Značajka Time predstavlja vrijeme u sekundama koje je proteklo između svake transakcije i prve transakcije u skupu podatka. Vrijeme nije vezano uz pojedinog korisnika. Pošto ima jako veliki raspon i ne utječe na krajnji rezultat briše se iz skupa podatka. Značajke V1 do V28 su glavne komponente koje se koriste pri analizi transakcije. Dobiveni su PCA analizom podataka. Nemaju specifično ime zbog anonimnosti te predstavljaju parametre koje banka prikuplja pri svakoj transakciji. Amount predstavlja količinu novaca koja je korištena pri transakciji. Raspon mu je između 0 i 25000 te se zbog toga mora skalirati. Class značajka predstavlja klasu, odnosno ako je vrijednost 0 transakcija je regularna, a ako je vrijednost 1 transakcija je prevara.

2.3. Analiza podataka

Prije treniranja modela potrebno je znati s kakvim podatcima se radi. U prošlom odjeljku prikazana su imena podataka. Korištenjem funkcije za provjeru tipa podatka utvrđeno je da su 30 značajki tipa float dok je jedna značajka i to klasa tipa int. Funkcijom za provjeru null vrijednosti podatak pokazano je da ne postoje izostale vrijednosti u podatkovnom skupu.



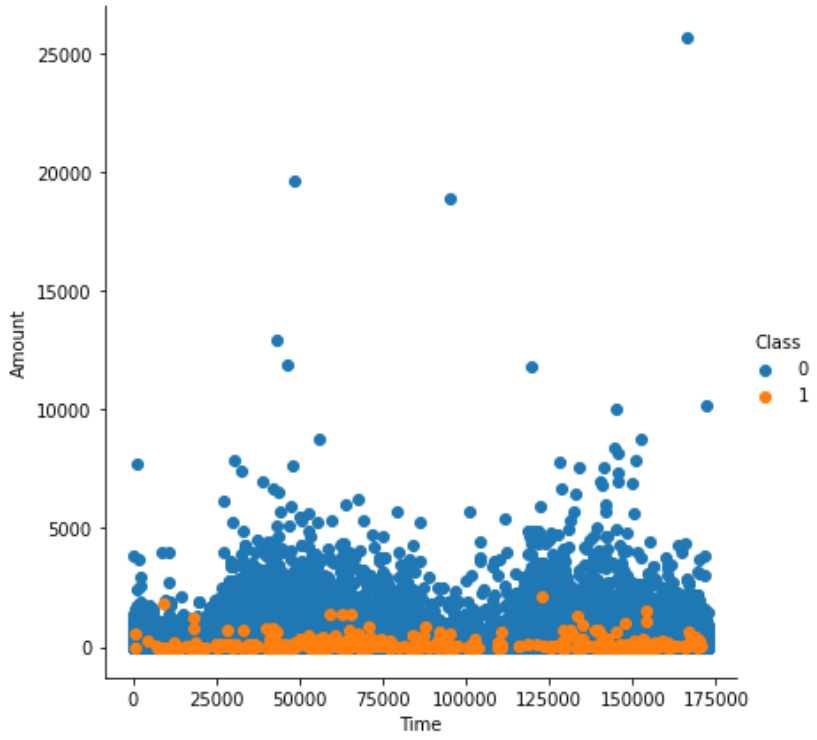
Slika 2.1 Matrica korelacije podataka

Na slici 2.1 prikazana je mapa korelacije. Njome se provjerava kakav je suodnos, odnosno međusobna povezanost između podataka. Analizom ove matrice utvrđeno je da podatci V1-V28 ne koreliraju što i treba biti slučaj jer su dobiveni PCA analizom podataka.

Tablica 2.2 Deskriptivna analiza Time i Amount značajke

	Time	Amount
Count	284807.000000	284807.000000
Mean	94813.859575	88.349619
Std	47488.145955	250.120109
Min	0	0
25%	54201.500000	5.600000
50%	84692.000000	22.000000
75%	139320.500000	77.165000
max	172792.000000	25691.160000

Pošto su značajke V1-V28 dobivene korištenjem PCA analize nema ih smisla deskriptivno analizirati. Tablica 2.2 prikazuje deskriptivnu analizu Time i Amount značajke. Kako je prije rečeno da će se Time značajka izbaciti iz skupa podataka neće se detaljno obrađivati. Analizom Amount značajke vidljivo je da je srednja vrijednost 88.35. Medijan je 22 što je znatno manje od srednje vrijednosti te ukazuje na to da postoje outlieri u skupu podataka ili su podatci pozitivno nakošeni što utječe na medijan. Minimalna transakcija je 0 dok je najveća 25691.



Slika 2.2 Dijagram raspršenja klasa

Na slici 2.2 vidljiv je dijagram raspršenja klasa. Narančastom bojom su označene klase prijevare dok su plavom bojom označene klase regularnih transakcija. Također vidljivi su outlieri u podatkovnom skupu.

2.4. Skaliranje značajke Amount

Kako bi se podatci skalirali na vrijednosti koje približno odgovaraju značajkama V1 do V28 koristi se RobustScaler koji je dostupan u Sklearn biblioteci. Na slici 2.1 vidljiv je isječak koda koji skalira podatke Amount. Podatci su skalirani na vrijednosti između -0.307413 i 358.683155. RobustScaler je korišten zato što je otporan na outliere zbog korištenja medijana i interkvartilnog raspona. Standardni scaler koristi srednju vrijednost i standardnu devijaciju kako bi skalirao podatke. Outlieri te dvije vrijednosti iskrivljuju te je konačan rezultat iskrivljen za razliku od RobustScalera.

```
from sklearn.preprocessing import RobustScaler

rob_scaler = RobustScaler()
data['scaled_amount'] = rob_scaler.fit_transform(data['Amount'].values.reshape(-1,1))
```

Slika 2.1 Skaliranje podataka

Ova metoda koristi medijan i interkvartilni raspon kako bi odredila skaliranu vrijednost. Formula kojom računa novu vrijednost podatka je:

$$X_{scale} = \frac{X_i - X_{med}}{X_{75} - X_{25}}$$

Nakon uklanjanja Time značajke i skaliranja Amount značajke podatci izgledaju kao što je predstavljeno u tablici 2.

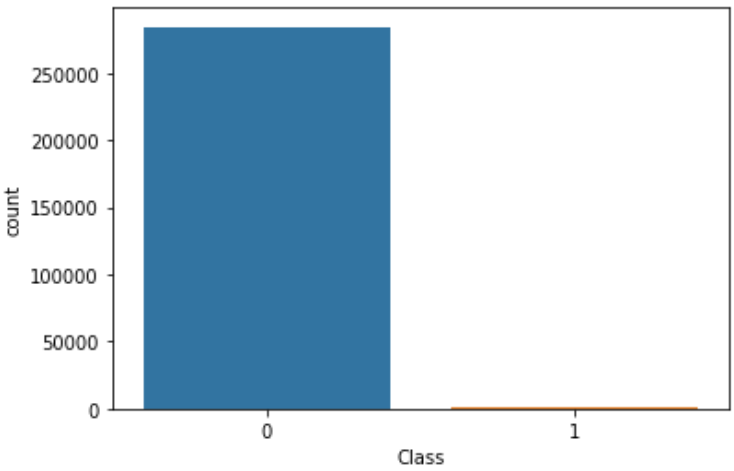
Tablica 2. Konačni podatci

V1	V2	…	V28	Class	scaled_amount
-1.359807	- 0.072781	…	-0.021053	0	1.783274
-1.358354	-1.340163	…	0.014724	0	-0.269825
-0.966272	-0.185226	…	-0.059752	0	4.983721

3. BALANSIRANJE PODATAKA

Prije treniranja modela potrebno je provjeriti balansiranost klasa. Ako se modeli istreniraju sa nebalansiranim skupom podataka model će biti pristran na klasu koja ima više instanci. Analizom podatkovnog skupa utvrđeno je da je samo 0.2% podatak klase 1 odnosno transakcija prevare. Na slici 3.1 i 3.2. su grafički vidljivi problemi nebalansiranosti.

Slika 3.1 Postotni prikaz nebalansiranosti

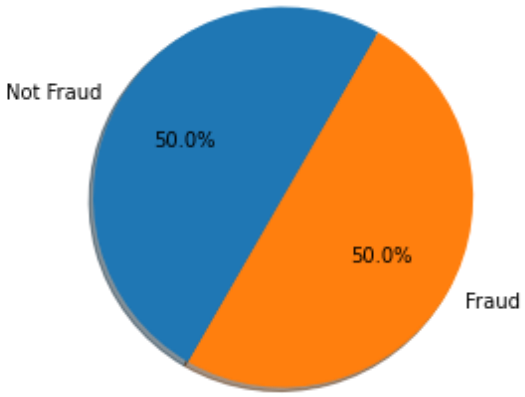


Slika 3.2 Prikaz nebalansiranosti stupčastim grafom

Postoje dva pristupa rješavanju problema balansiranosti, a to su undersampling i oversampling. Undersampling je metoda kojom se broj instanci dominante klase nastoji izjednačiti sa brojem instanci manjinske klase. Ovresampling radi suprotno tako što broj instanci manjinske klase nastoji izjednačiti sa brojem instanci dominante klase. Za ovaj podatkovni skup bolje je koristiti oversampling zbog postotka manjinske klase te se ovim načinom ne gube korisne informacije za treniranje modela. U ovome projektu korištena je SMOTE analiza za balansiranje podataka.

SMOTE analiza za balansiranje odabire jednu instancu manjinske klase nasumično. Zatim za K susjeda izračunava udaljenost i množi ju sa nasumičnim brojem između 0 i 1. Novi podatak stavlja se na dobivenu poziciju. Analiza se ponavlja dok se ne izjednači broj klasa.

Prije izvođenja analize podatci su podijeljeni na trening i test podatke. Balansiranje se provodi samo na trening podatcima kako model ne bi bio pristran. Na slici 3.3 vidljivi su krajnji rezultati SMOTE analize odnosno jednak broj instanci obje klase. Ukupan broj trening podataka klase prevare se povećao zbog analize što je vidljivi u tablici 3.1.



Slika 3.3 Balansiranost trening podataka

Tablica 3.1 Broj podatka prije i poslije SMOTE analize

	0	1
Prije SMOTE analize	227454	391
Poslije SMOTE analize	227454	227454

4. MODELI

4.1. Logistička regresija

Logistička regresija je algoritam koji služi za računanje vjerojatnosti binarnih klasa. Analiziranjem veza između varijabli određuje se vjerojatnost za klasu što predstavlja izlaz iz modela. Određeni prag se postavlja za pojedinu klasu.

Prednost logističke regresije je to što je računski brz, pogotovo za male podatkovne skupove. Također ne zahtjeva predobradu podataka poput skaliranja.

Nedostatak logističke regresije je taj da pretpostavlja da su podatci linearno ovisni što nije uvijek slučaj. Također je osjetljiv na outliere i overfitting.

4.2. Klasifikator stabla odluke

Klasifikator stabla odluke radi tako da konstruira model odluka i njihovih mogućih posljedica u obliku stabla. Svaki unutarnji čvor predstavlja test značajke dok listovi predstavljaju oznake klasa. Da bi napravio predviđanje za novu instancu, algoritam slijedi put odluke u stablu, počevši od korijena i završavajući u čvoru lista, na temelju vrijednosti značajki. Oznaka klase dosegnutog lisnog čvora je predviđanje za novu instancu.

Prednost ovog klasifikatora je brzina i mogućnost vizualizacije modela. Također ne zahtjeva predobradu podataka i može ostvariti vezu između nelinearno povezanih podataka čime je dobar izbor za kompleksne probleme.

Nedostatak klasifikatora je nestabilnost gdje male promjene u podatcima mogu uzrokovati drugačije strukture stabla i drugačiji krajnji rezultat. Također algoritam je pohlepan što znači da možda neće pronaći optimalna rascjep za svaki čvor.

4.3. Klasifikator nasumične šume

Klasifikator nasumične šume radi tako da za vrijeme treninga konstruira veliki broj stabala odluke i ispisuje klasu pojedinačnog stabla. Ključna ideja iza nasumičnih šuma je ukrasiti stabla, tj. napraviti stabla koja su što neovisnija, tako da prosjek procjene bude bolji od procjene bilo kojeg pojedinačnog stabla. Da bi se to postiglo algoritam pri svakom grananju stabla odabire nasumičan podskup značajki za grananje. Ovo čini stabla raznolikima i smanjuje overfitting.

Prednost ovog klasifikatora je to što je precizan čime ostvaruje visoke performanse na velikom broju podatkovnih skupova. Također je otporan na outliere i overfitting zato što koristi prosjek svih stabla kako bi smanjio utjecaj greške pojedinog stabla.

Nedostatak ovog algoritma je sporo predviđanje rezultata zato što mora uzeti prosjek svih stabla. Također je memorijski intenzivan, pogotovo za velike podatkovne skupove, zato što sprema više stabla u memoriju. Teško ga je interpretirati i shvatiti kako model radi predviđanje.

4.4. XGBoost klasifikator

XGBoost je algoritam za povećanje gradijenta za stabla odluke. Implementacija je frameworka za povećanje gradijenta koja je skalabilna, brza i točna. Algoritam radi stvaranjem skupa stabala odluke kako bi analizirao podatke za obuku i korištenjem postupka optimizacije spuštanja gradijenta kako bi se smanjila funkcija gubitka i poboljšala točnost predviđanja.

Prednost ovog klasifikatora je brzina treniranja, pogotovo na velikim podatkovnim skupovima. Koristi paralelnu obradu strukture i efektivne algoritme kako bi postigao optimalno vrijeme treniranja. Također je otporan na outliere i šumove u podatkovnim skupovima.

Nedostatak klasifikatora je računalna zahtjevnost. Zbog velikog broja stabla odluke koji se treniraju potrebna je velika snaga računala, pogotovo na velikim skupovima podataka. Također model je kompleksan i teško ga je interpretirati čime je teško shvatiti kako model radi predviđanje.

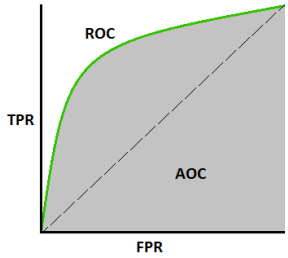
5. IZRADA I EVALUACIJA MODELA

Modeli korišteni u ovome projektu su dostupni u Sklearn biblioteci. Kako bi se olakšala izrada i evaluacija modela napisane su funkcije kojima se predaje instanca modela i trening i test podatci. Na slici 5.1 vidljiv je isječak koda za te dvije funkcije.

```
def model(classifier,x_train,y_train,x_test,y_test):

    classifier.fit(x_train,y_train)
    prediction = classifier.predict(x_test)
    cv = RepeatedStratifiedKFold(n_splits = 10,n_repeats = 3,random_state = 1)
    print("Cross Validation Score : ",'{0:.2%}'.format(cross_val_score(classifier,x_train,y_train,cv = cv,scoring = 'roc_auc').mean()))
    print("ROC_AUC Score : ",'{0:.2%}'.format(roc_auc_score(y_test,prediction)))
    plot_roc_curve(classifier, x_test,y_test)
    plt.title('ROC_AUC_Plot')
    plt.show()

def model_evaluation(classifier,x_test,y_test):


    cm = confusion_matrix(y_test,classifier.predict(x_test))
    names = ['True Neg','False Pos','False Neg','True Pos']
    counts = [value for value in cm.flatten()]
    percentages = ['{0:.2%}'.format(value) for value in cm.flatten()/np.sum(cm)]
    labels = [f'{v1}\n{v2}\n{v3}' for v1, v2, v3 in zip(names,counts,percentages)]
    labels = np.asarray(labels).reshape(2,2)
    sns.heatmap(cm,annot = labels,cmap = 'Blues',fmt ='')


    print(classification_report(y_test,classifier.predict(x_test)))
```

Slika 5.1 Funkcije za treniranje i evaluaciju modela

Pošto je korištena SMOTE analiza modeli se evaluiraju pomoću Cross Validation ocjena i ROC-AUC ocjena. Cross validation se koristi kako bi se procijenila sposobnost modela na trening podatcima, prije uporabe na test podatcima. Trening podatci se dijele na K broj foldova, u ovome slučaju 10, i svaki fold se koristi validaciju modela. Proces se ponavlja n puta kako bi se dobila što točnija evaluacija modela. Konačna evaluacija modela je prosječna vrijednost dobivena evaluacijom pojedinim foldom. ROC-AUC ocjena se koristi za evaluaciju modela binarnih klasifikatora. Evaluacija se provodi na test podatcima. Na slici 5.2 vidljiv je primjer ROC krivulje. Na osima krivulje nalaze se TPR (true positive rate) i FPR (false positive rate). Savršeni klasifikator ima vrijednost TPR 1 i FPR 0 čime krivulja prolazi kroz gornji lijevi kut. AUC je područje ispod ROC krivulje i sposobnosti modela ocjenjuje sa brojem između 0 i 1. Ako model ima AUC ocjenu 1 znači da savršeno raspoznaje klase. Cross validation i ROC-AUC ocjene se iskazuju u postotcima.



Slika 5.2 Primjer ROC-AUC krivulje

True positive rate (TPR) predstavlja udio instanci koje je model ispravno klasificirao kao pozitivne. Formula za TPR je:

$$TPR = \frac{TRUE\ POSITIVE}{TRUE\ POSITIVE + FALSE\ NEGATIVE}$$

False positive rate (FPR) udio je negativnih instanci koje je model netočno klasificirao kao pozitivne. Formula za FPR je:

$$FPR = \frac{FALSE\ POSITIVE}{FALSE\ POSITIVE + TRUE\ NEGATIVE}$$

Također je napisana funkcija koja ispisuje matricu zabune za model. Matrica zabune uspoređuje klase predviđene modelom sa stvarnim klasama test skupa podataka. Iz matrice se mogu izračunati određene vrijednosti za evaluaciju poput točnosti i F1 ocjene modela. Točnost je osjetljiva na nebalansirane podatke te zbog korištenja SMOTE analize nije dobar indikator modela. F1 ocjena je bolja vrijednost za evaluaciju modela, dok je najvjerodostojnija procjena pomoću ROC-AUC krivulje. F1 ocjena se računa pomoću formule:

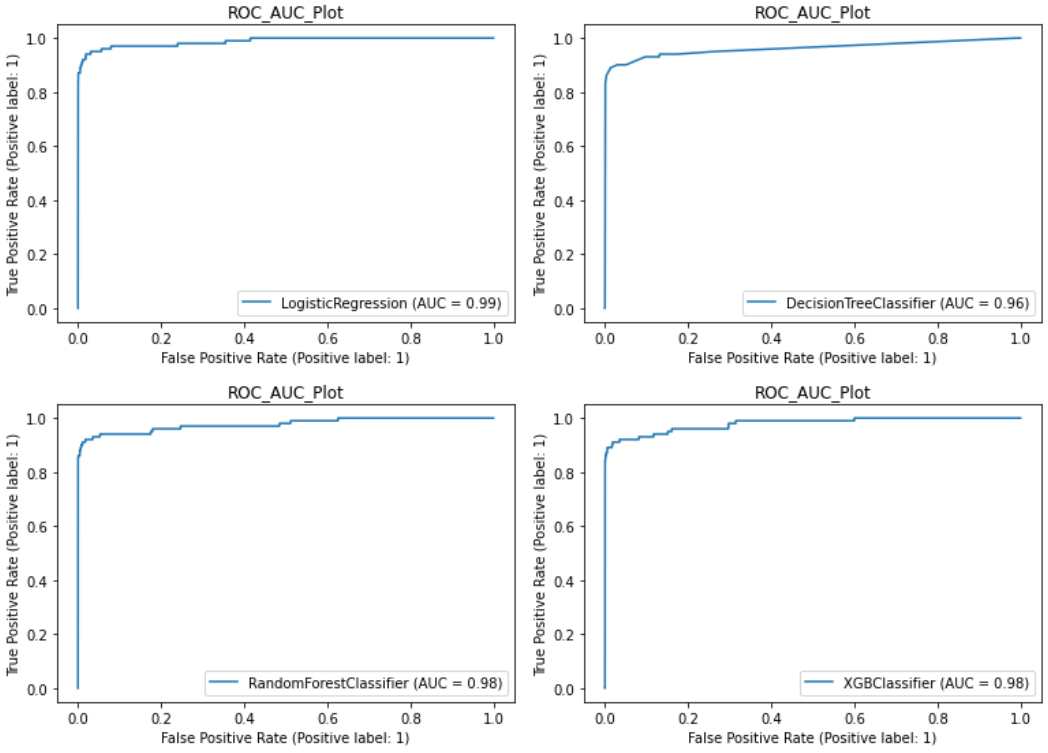
$$F1 = \frac{2 * TRUE\ POSITIVE}{2 * TRUE\ POSITIVE + FLASE\ POSITIVE + FALSE\ NEGATIVE}$$

5.1. Rezultati modela

Rezultati za svaki model prema gore navedenim metrikama se nalaze u tablici 5.1. Iz rezultata je vidljivo da je na trening podatcima, odnosno Cross validation ocjena, XGBoost klasifikator imao najbolje rezultate. Logistička regresija je imala najbolje rezultate na test skupu podataka, odnosno ROC-AUC ocjena.

Tablica 5.1 Rezultati evaluacije modela

	LOGISTIČKA REGRESIJA	KLASIFIKATOR STABLA ODLUKE	KLASIFIKATOR NASUMIČNE ŠUME	XGBOOST KLASIFIKATOR
CROSS VALIDATION OCJENA	98.84%	98%	98.71%	99.86%
ROC-AUC OCJENA	95.81%	93.16%	93.81%	93.96%
AUC OCJENA	0.99	0.96	0.98	0.98



Slika 5.3 ROC krivulja za sve modele

Na slici 5.3 vidljive su ROC krivulje na sve modele. Krivulja za logističku regresiju je najbliža gornjem lijevom kutu te je zato ROC-AUC ocjena najveća za taj model. Model klasifikatora stabla odluke ima najmanju površinu ispod grafa, odnosno najmanju AUC ocjenu.

Slika 5.4 Matrice zabune modela

Slika 5.4 pokazuje matricu zabune za sve modele. Gore lijevo je matrica za model logističke regresije, gore desno klasifikator stabla odluke, dole lijevo klasifikator nasumične šume i dole desno XGBoost klasifikator. F1 ocjena je izračunata za svaki model i nalazi se u tablici 5.2. Iako točnost nije najbolja vrijednost za evaluaciju, također je izračunata.

Tablica 5.2 Vrijednosti F1 i točnosti za modele

	LOGISTIČKA REGRESIJA	KLASIFIKATOR STABLA ODLUKE	KLASIFIKATOR NASUMIČNE ŠUME	XGBOOST KLASIFIKATOR
F1 ocjena	0.1202	0.0779	0.3771	0.2081
Točnost	0.9756	0.9622	0.9948	0.9880

6. ZAKLJUČAK

Klasifikacija se sve više primjenjuje u svakodnevnom životu kako bi se obradile velike količine podataka u malo vremena što čovjek ručno ne može napraviti. U radu su pokazani samo neki modeli kojima se postiže prepoznavanje prijevara kreditnim karticama. S obzirom na rezultate evaluacije, odnosno ROC-AUC ocjene za test podatke, model logističke regresije pokazao je da ima najveću preciznost određivanja prevare. Ostali modeli se mogu poboljšati podešavanjem parametara što je različito za svaki podatkovni skup.

Literatura

[1] Podatkovni skup prijevare kreditnim karticama: https://www.kaggle.com/datasets/mlg-ulb/creditcardfraud