**SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU**

**FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I INFORMACIJSKIH TEHNOLOGIJA OSIJEK**

**Sveučilišni studij**

**OTKRIVANJE PRIJEVARA U KARTIČNIM TRANSAKCIJAMA**

**Projektni zadatak**

**Andrej Horvat**

**Mihovil Janković**

**Osijek, 2024.**

Sadržaj

[1. UVOD 1](#_Toc157527683)

[2. PREDOBRADA PODATAKA 2](#_Toc157527684)

[2.1. Opis zadatka 2](#_Toc157527685)

[2.2. Podatkovni skup 2](#_Toc157527686)

[2.3. Analiza i predobrada podataka 2](#_Toc157527687)

[2.4. Skaliranje značajke Amount 5](#_Toc157527688)

[2.5. Balansiranje podataka 5](#_Toc157527689)

[3. MODELI 8](#_Toc157527690)

[3.1. Nadzirano učenje 8](#_Toc157527691)

[3.1.1 Logistička regresija 8](#_Toc157527692)

[3.1.2 Stablo odluke 8](#_Toc157527693)

[3.1.3. XGBoost 9](#_Toc157527694)

[3.1.4. CatBoost 9](#_Toc157527695)

[3.2. Nenadzirano učenje 9](#_Toc157527696)

[3.2.1. Autoenkoder 9](#_Toc157527697)

[4. EVALUACIJA MODELA 11](#_Toc157527698)

[5. ZAKLJUČAK 14](#_Toc157527699)

[LITERATURA 15](#_Toc157527700)

# UVOD

U ovom projektnom zadatku obrađena je tema prepoznavanja prijevara kod kartičnih transakcija. Obrađene su metode nadziranog učenja poput logističke regresije, stabla odluke, XGBoost klasifikatora i Catboost klasifikatora kao i metoda nenadziranog učenja poput autoenkodera. Kartične transakcije česte se u današnjem svijetu te banke nastoje što bolje zaštiti svoje korisnike kako bi što spriječile gubitke svojih klijenata. Banke moraju imati razvijene sustave kojima je cilj što prije prepoznati i spriječiti prijevare te za to koriste brojne metode za ocjenu prijevare, umjetnu inteligenciju koja uz pomoć brojnih parametara i istreniranih modela predviđa radi li se o prijevari ili ne, te brojne druge načine. Za podatkovni skup korišten je podatkovni skup „creditcard.csv“ sa stranice Kaggle. Jedan od glavnih modela kod svih podatkovnih skupova vezanih za kartične prijevare je neizbalansiranost skupova, odnosno omjer prijevara naspram omjera stvarnih transakcija je vrlo malen. Kako bi se podatkovni skup izbalansirao korišten je SMOTE.

# PREDOBRADA PODATAKA

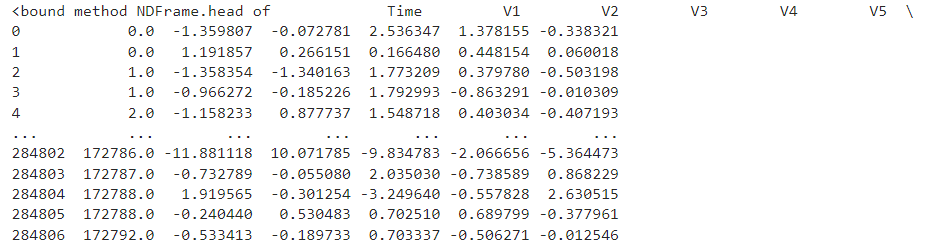
## Opis zadatka

Otkrivanja prijevare u kartičnim transakcijama kod nadziranog učenja spada u probleme binarne klasifikacije, svaka transakcije može biti prijevara ili ispravna transakcija. Kod nenadziranog učenja ne koristi se informacija o tome je li transakcija ispravna ili ne nego se nastoji naučiti autoenkoder reprezentaciju normalnih transakcija te identificirati anomalije kroz rekonstrukcijsku pogrešku kako bi se omogućilo efikasno otkrivanje uzoraka koji mogu ukazati na prijevaru.

## Podatkovni skup

Kod treniranja modela korišten je podatkovni skup „dreditcard.csv“ s Kagglea koji sadrži preko 280,000 podataka i 31 značajku[1].

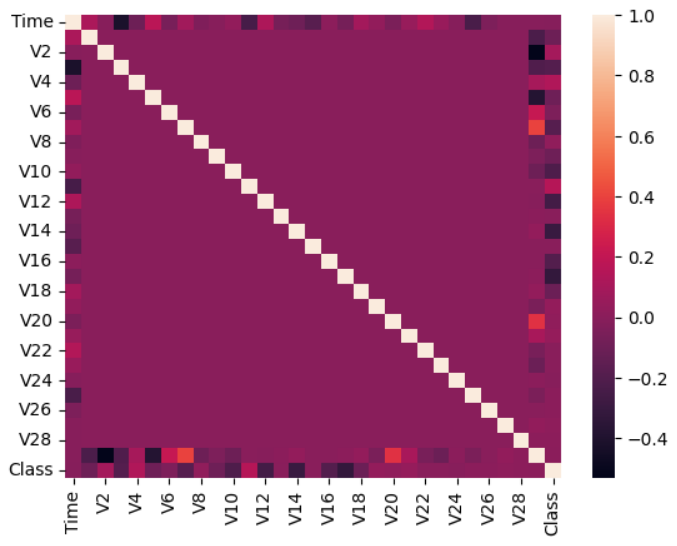
Slika 2.1. prikazuje neke od parametara skupa podataka nakon poziva funkcije head(). Značajke s oznakama V1 do V28 predstavljaju glavne komponente dobivene PCA analizom anonimnih podataka o transakcijma koje banka prikuplja. „Amount „ označava iznos novca korišten pri transakciji, raspon iznosa je od 0 do 25000 pa je taj stupac potrebno skalirati. Značajka „Time“ označava vrijeme u sekundama od prve transakcije te nije vezano za pojedinog korisnika i s obzirom da neće imati utjecaj na krajnji rezultat ta značajka se izbacuje.



Slika 2.1. Prikaz dijela podatak podatkovnog skupa nakon poziva funkcije .head()

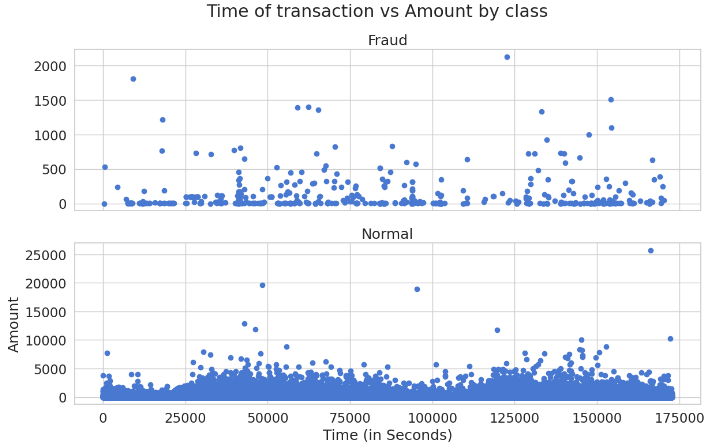
## Analiza i predobrada podataka

Već smo utvrdili da nam Time stupac nije potreban prilikom izrade modela, no podatke je potrebne dodatno predobraditi. U drugim situacijama trebalo bi provjeriti nalaze li se null vrijednosti unutar podataka no na Kaggle stranici podatkovnog skupa navedeno je da je podatkovni skup potpun. Nadalje, potrebno je vidjeti korelaciju između podataka. Slika 2.2. prikazuje korelaciju svih značajki podatkovnog skupa. Korelacija između podataka trebala bi biti mala s obzirom da su podatci dobiveni PCA analizom, što se i može vidjeti iz slike.



Slika 2.2. Prikaz korelacije značajki

Kako bi se potvrdilo da značajka Time neće biti potrebna za treniranja modela može se pogledati dijagram raspršenja značajki Time i Amount. Slika 2.3. prikazuje dijagram raspršenja Time i Amount značajki ovisno o tome je li transakcija prijevara ili stvarna transakcija.



Slika 2.3. Dijagram raspršenja ovisnosti Time i Amount značajke

Kao što je vidljivo iz dijagram Time značajke kod prijevara i stvarnih transakcija imaju sličnu te se značajka Time neće koristiti kod treniranja modela.

S obzirom da se značajka Time izbacuje, a značaje V1 do V28 dobivene su PCA analizom za deskriptivnu analizu preostaje samo značajka Amount.

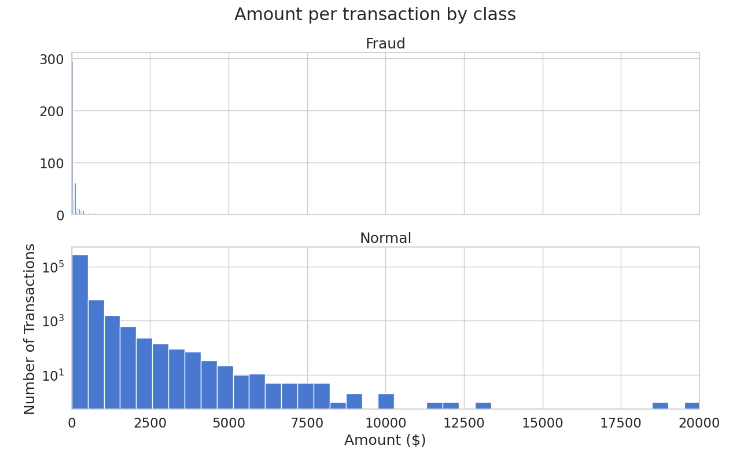
Tablica 2.1. prikazuje deskriptivnu analizu Amount značajke.

|  |  |
| --- | --- |
|  | Amount |
| Count | 284807.000000 |
| Mean | 88.349619 |
| Std | 250.120109 |
| Min | 0 |
| 25% | 5.600000 |
| 50% | 22.000000 |
| 75% | 77.165000 |
| max | 25691.160000 |

Tablica 2.1. Deskriptivna analiza značajke Amount

Iz deskriptivne analize može se uočiti da je srednja vrijednost podataka veća i od medijana i od gornjeg kvartala vrijednosti što je dobar pokazatelj da u podatkovnom skupu postoji nerazmjer u veličini transakcija što znači da je značajku Amount sigurno potrebno skalirati.

Slika 2.4. prikazuje graf učestalosti pojavljivanja određene veličine unutar značajke Amount.



Slika 2.4. Graf učestalosti pojavljivanja određene veličine unutar značajke Amount

Iz grafa je vidljivo nekoliko stvari. Naznačena je neravnopravnost količine stvarnih transakcija i prijevara. Također, vidljivo je da u slučaju kada je transakcija prijevara obično se radi o manjim novčanim iznosima te su naznačeni outlieri u stvarnim transakcijama zbog kojih je potrebno skalirati stavku Amount.

## Skaliranje značajke Amount

Nakon skaliranja značajka Amount trebala bi biti usklađena s ostalim značajkama. Kako bi se to omogućilo korišten je RobustScaler. Ovaj odabir motiviran je sposobnošću RobusrScalera da se nosi s outlierima zahvaljujući upotrebi medijana i interaktivnog raspona. Klasične metode skaliranja mogu biti osjetljive na prisutnost ekstremnih vrijednosti, što može iskriviti konačni rezultat. Korištenjem RobustScalera osigurava se kompatibilnost s ostalim značajkama u skupu podataka dok istovremeno održava robusnost u prisutnosti outliera.

RobustSclaer funkcionira tako što uzima medijan (Q2) skupa podataka, odnosno vrijednost koja dijeli skup na dva jednaka dijela i interkvartalni raspon (IQR) koji predstavlja raspon između prvog (Q1) i trećeg (Q3) kvartila. Q1 predstavlja vrijednost ispod koje leži četvrtina najmanjih vrijednosti, dok Q3 predstavlja vrijednost iznad koje leži četvrtina najvećih vrijednosti.

Formula za RobustScaler glasi: RobustScaler= , gdje X predstavlja svaku pojedinačnu vrijednost skupa podataka, Q2 je medijan skupa podataka i IQR je interaktivni raspon, odnosno Q3-Q1.

RobustScaler zatim skalira podatke pomoću razlike između svake vrijednosti i medijana, podijeljene interkvartalnim rasponom.

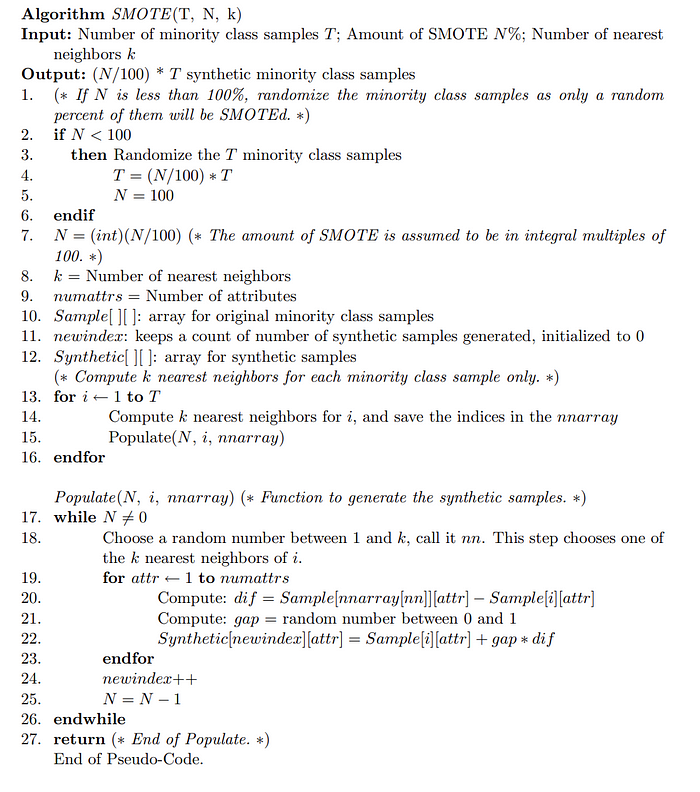
## Balansiranje podataka

Jedan od glavnih problema ovog zadatka je balansiranje podatka. Prijevare je teško uočiti jer se odnose na mal broj ukupnih transakcija pa tako nisu dovoljno prisutne ni u ovom podatkovnom skupu. Zbog toga potrebno je izbalansirati skup kako bi modeli mogli biti istrenirani i na ispravnim i na lažnim transakcijama kako bi bolje mogli napraviti razlike između njih. Postoje dva glavna pristupa balansiranju podataka, undersampling koji izjednačava broj dominantne klase s brojem manjinske klase i oversampling koji je primjereniji za korištenje u ovom primjeru, koji izjednačava broj manjinske klase s brojem većinske klase.

Kako bi postigli oversampling korišten je SMOTE (engl. Synthetic Minority Over-sampling Techniqur). SMOTE je tehnika za obradu neuravnoteženih skupova podataka, posebno kod problema s klasifikacijom gdje je jedna klasa značajno manje zastupljena od druge, u ovom slučaju omjer ispravnih i lažnih transakcija. SMOTE radi tako da generira sintetičke primjerke manjinske klase kako bi se izjednačila distribucija klasa.[2]

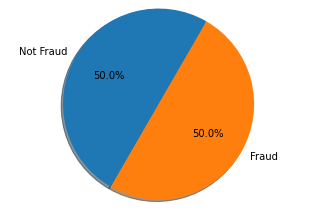
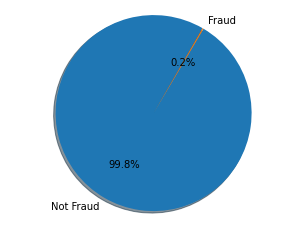
Pojednostavljeno SMOTE funkcionira tako da izračunava vektor razlike između uzoraka i njegovog najbližeg susjeda, zatim pomnoži vektor razlike s nasumičnim brojem odabranim između 0 i 1. Nakon toga dodaje taj umnoženi vektor razlike izvornom uzorku, generirajući novi sintetički primjer u prostoru značajki te ponavlja postupak za sljedećeg najbližeg susjeda, sve dok se ne generira unaprijed definiran broj sintetičkih primjera. [3]

Slika 2.5. prikazuje potpuni algoritam SMOTE.



Slika 2.5. SMOTE algoritam u cijelosti [3]

Prije korištenja SMOTE algoritma, podatkovni skup dijeli se na testne i trening podatke te se balansiranje radi samo na trening podatcima. Slika 2.6. prikazuje omjer ispravnih transakcija i prijevara prije i poslije primjene SMOTE-a, s lijeve strane nalaze se originalni podatkovni skup, a s desne modificirani.



Slika 2.6. Prikaz omjera stvarnih transakcija i prijevara prije i nakon korištenja SMOTE-a

# MODELI

U ovom projektnom zadatku korišteni su modeli nadziranog i nenadziranog učenja. Za nadzirano učenje korištena je linearna regresija, stablo odluke, XGBoost klasifikator i CatBoost klasifikator dok je za nenadzirano učenje izgrađen autoenkoder.

## Nadzirano učenje

Nadzirano učenje je vrsta strojnog učenja u kojoj se model uči na označenim podatcima. U ovom procesu, algoritam se trenira na skupu podataka koji sadrži ulazne podatke zajedno s odgovarajućim izlaznim oznakama. Cilj je naučiti mapirati predviđanja ili klasificirati nove, neviđene podatke.

U ovom poglavlju dodatno su opisane metode logističke regresija, stabla odlučivanja, XGBoost klasifikatora i CatBoost klasifikatora te su navedeni krajnji rezultati točnosti i gubitka modela prilikom treniranja.

Kod modela logističke regresije i stabla odluke za odabir hiperparametara korištena je tehnika optimizacije GridSearchCV koja je dio sklearn biblioteke. GridSearchCV pretražuje zadani prostor hiperparametara kako bi pronašao najbolje kombinacije hiperparametara, podatci se dijele na više skupova (obično 5 ili 10) i model se trenira na jednom skupu podataka i testira na drugom, to se radi više puta za svaku kombinaciju hiperparametara, a konačne performanse modela izračunavaju se kao prosjek rezultata unakrsne validacije. Nakon što se provede pretraživanje odabire se najbolja kombinacije hiperparametara. Za XGBoost i CatBoost korištene su osnovne postavke hiperparametara jer su modeli davali dovoljno dobre rezultate.

### 3.1.1 Logistička regresija

Logistička regresija tip je statističkog modela koji procjenjuje vjerojatnost da će se događaj dogoditi, odnosno u slučaju ovog zadatka radi li se o kartičnoj prijevari ili stvarnoj transakciji, na temelju danog skupa nezavisnih varijabli. Često se koristi za klasifikaciju i analizu predviđanja. Budući da je ishod vjerojatnost, zavisna varijabla je omeđena između 0 i 1. [4] Prednosti logističke regresije je pružanje jednostavnog modela koji se lako interpretira, daje dobre rezultate za linearne probleme, ima brzu konvergenciju i efikasnost, dobro se nosi s visokodimenzionalnim podacima i dobro se prilagođava probabilističkom okviru. Neki od nedostatak su potencijalni problemi s modeliranjem složenih nelinearnih odnosa u podacima, osjetljivost na ekstremne vrijednosti i outliere i ograničena sposobnost rješavanja višeklasnih problema.

### 3.1.2 Stablo odluke

Stablo odluke je neparametarksa metoda nadziranog učenja koja se koristi za klasifikaciju i regresiju. Cilj je stvoriti model koji predviđa vrijednost ciljane varijable učeći jednostavna pravila odlučivanje iz značajki podataka. Neke od prednosti stabla odluke se jednostavnost razumijevanja i interpretacije, zahtijeva malo pripreme podataka, mogu raditi s numeričkim i kategoričkim podatcima i mogu rješavati probleme s više izlaza. Nedostatci su sklonost overfittingu, predviđanja nisu kontinuirana i stvaranje pristranih stabala kod dominacije određenih klasa. [5]

Kod korištenja GridSearchCV-a za pronalazak najboljih hiperparametara, GridSearchCV dao je dobre rezultate preciznosti i odziva. Dodatnim podešavanjem parametara povećao se odziv, a smanjila preciznost, što je dobro za otkrivanje prijevara u kartičnim transakcijama uz pretpostavku da je bolje pogrešno označiti stvarne transakcije nego da se pogrešno označe prijevare.

### 3.1.3. XGBoost

XGBoost je optimizirana distribuirana knjižnica za pojačano učenje (engl. gradient boosting), osmišljena da bude iznimno efikasna, fleksibilna i prenosiva. Implementira algoritme strojnog učenja unutar okvira Gradient Boosting. [6] Prednosti XGBoosta su učinkovitost, fleksibilnost u rješavanju raznih problema, ugrađena regularizacija i mogućnost rada s raznovrsnim tipovima podataka. Neki od nedostataka su sklonost overfittingu kod malih skupova podataka, neprilagođenost radu s velikim brojem kategoričkih značajki i veća potrošnja resursa u usporedbi s jednostavnijim modelima.

### 3.1.4. CatBoost

CatBoost je visokoučinkovita open-source knjižnica za pojačano učenje na stablima odlučivanja koja se može koristiti za zadatke klasifikacije, regresija i rangiranja. CatBoost koristi kombinaciju poretka pojačavanja, slučajnih permutacija i optimizacije temeljene na gradijentu kako bi postigao visoke performanse na velikim i složenim skupovima podataka s kategoričkim značajkama. [7] Neke od prednosti Catboosta su automatsko rukovanje kategoričkim značajkama, potpora za paralelno izvođenje, optimizacija na temelju gradijenta i efikasnost na velikim skupovima podataka. Nedostatci su veća potrošnja resursa u usporedbi s jednostavnijim modelima, složenost implementacije modela i duže vrijeme treniranja u usporedbi s jednostavnijim modelima.

## Nenadzirano učenje

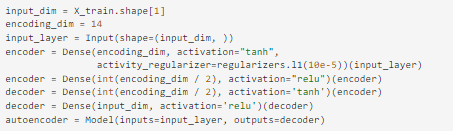
Nenadzirano učenje je grana strojnog učenja u kojoj algoritmi obrađuju skup podataka bez unaprijed označenih izlaznih oznaka. Umjesto toga, algoritam pokušava otkriti obrasce, strukturu ili skrivene informacije iz samih podatka.

U ovom projektu za svrhe nenadziranog učenja korišten je autoenkoder.

### Autoenkoder

Autoenkoderi su klasa algoritama dizajnirana za nenadzirano učenje, sposobna naučiti učinkovite reprezentacije ulaznih podataka bez potrebe za oznakama. Osnovni princip autoenkodera je komprimiranje i učinkovito predstavljanje ulaznih podataka bez specifičnih oznaka . To se postiže korištenjem dvoslojne strukture koja se sastoji od enkodera i dekodera. Enkoder transformira ulazne podatke u smanjenu dimenzijsku reprezentaciju. Iz te reprezentacije, dekoder ponovno gradi početni ulaz. Taj proces se odvija kako bi mreža stekla značajne uzorke u podacima. [8]

Autoenkoder za ovaj projektni zadatak sastoji se od 4 potpuo povezana sloja sa 14, 7, 7 i 29 neurona, Prva dva sloja koriste za enkoder dok su zadnja dva sloja dekoder. Prilikom treniranja koristi se L1 regularizacija. Za pomoć u izgradnji strukture autoenkodera korišten je članak Credit Card Fraud Detection using Autoencoder in Keras [9]. Slika 3.1. prikazuje strukturu autoenkodera.



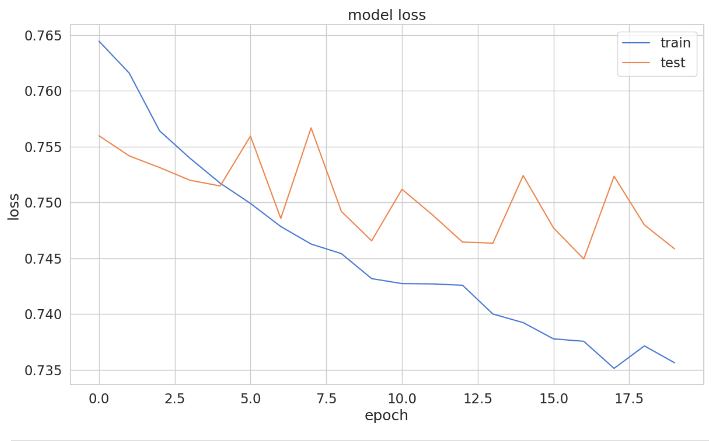
Slika 3.1. Struktura autoenkodera

Model se trenira na 20 epoha sa veličinom batcha 32 i najbolji model se sprema. Slika 3.2. prikazuje kod za treniranje i spremanje modela.



Slika 3.2. Treniranje autoenkodera

Slika 3.3. prikazuje graf funkcije gubitka prilikom treniranja.



Slika 3.3. Graf funkcije gubitka

# EVALUACIJA MODELA

Za evaluacija modela korišteni su ROC-AUC (engl Reciver Operating Characteristic- Area Under the Curve), unakrsna validacija (engl. Cross validation), AUC (engl. Area Under the Curve), ROC-AUC grafovi, matrice zabune, preciznost, odziv, i F1-ocjena. ROC-AUC je mjera koja koristi AUC za procjenu performansi binarnih klasifikacijskih modela, fokusira se na sposobnost modela da razlikuje između pozitivnih i negativnih instanci, prateći odnos osjetljivosti (engl. true positive rate) i specifičnosti (engl. false postive rate). AUC označava površinu ispod krivulje i može se primijeniti na bilo koju krivulju gdje se uspoređuje neka varijabla na osnovi različitih pragova i vrijednosti. Cross validation je tehnika koja se koristi za procjenu performansi modela i smanjenje varijabilnosti u procijeni.

Za svaki od modela računane su vrijednosti preciznosti, odziva i F1-ocjene. Preciznost je omjer istinski pozitivnih instanci prema svim pozitivnim predikcijama. Odziv je omjer istinski pozitivnih instanci koje su pravilno klasificirane prema svim stvarnim pozitivnim instancama. F1-score je harmonijska sredina preciznosti i odziva.

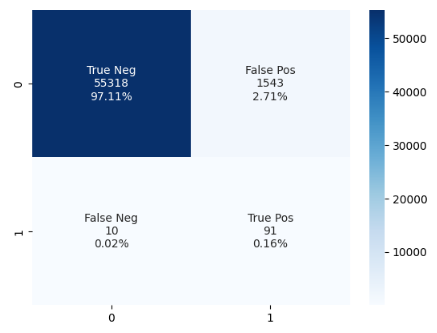
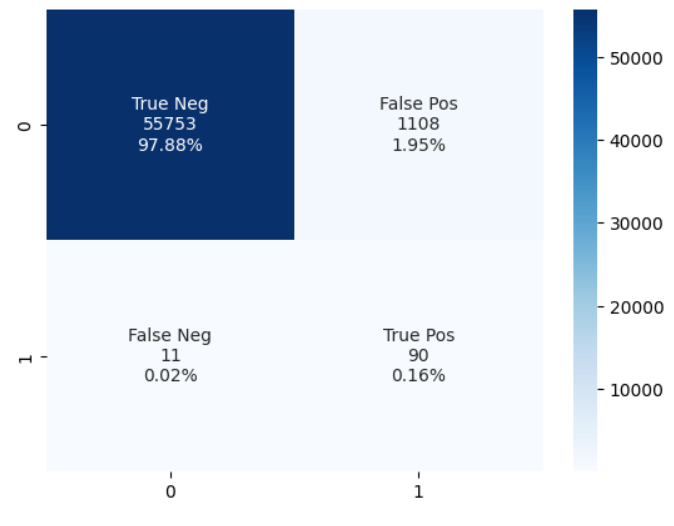
Tablica 4.1. prikazuje vrijednosti preciznosti, odziva i F1-score za svaku klasu logističke regresije, stabla odluke, XGBoosta i CatBoosta.

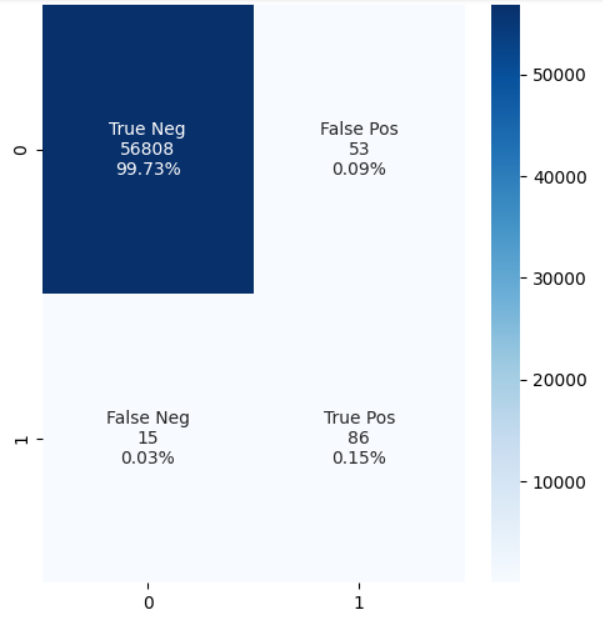
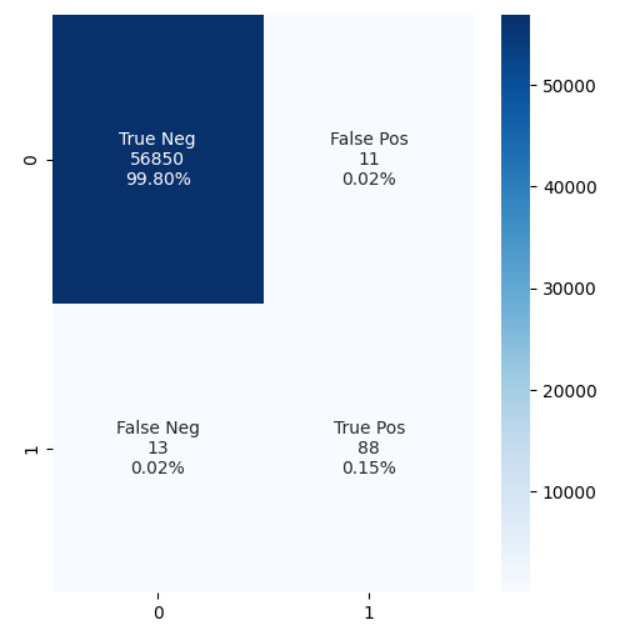
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Preciznost | | Odziv | | F1-ocjena | |
| Klasa 0 | Klasa 1 | Klasa 0 | Klasa 1 | Klasa 0 | Klasa1 |
| Logistička regresija | 1.00 | 0.08 | 0.98 | 0.89 | 0.99 | 0.14 |
| Stablo odluke | 1.00 | 0.06 | 0.97 | 0.90 | 0.99 | 0.10 |
| XGBoost | 1.00 | 0.85 | 1.00 | 0.88 | 1.00 | 0.86 |
| CatBoost | 1.00 | 0.62 | 1.00 | 0.85 | 1.00 | 0.72 |

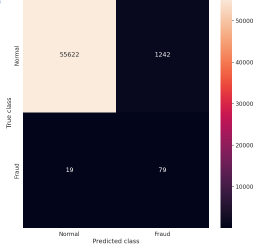
Tablica 4.1. vrijednosti preciznosti, odziva i F1-ocjena

Autoenkoderi se često koriste u nenadziranom učenju, gdje model nije izričito opremljen oznakama za svaki primjer (prijevara ili ne prijevara). Model uči reprezentirati normalne uzorke u podacima tijekom obuke. Kao rezultat toga, nema označenih instanci kako bi izračunao tradicionalne metrike klasifikacije poput točnosti ili odziva. S obzirom na to računala se vrijednost metrike MSE (engl. Mean Squared Error). MSE je mjera koja se koristi za procjenu performansi regresijskih modela. Ova metrika kvantificira prosječnu kvadratnu razliku između stvarnih i predviđenih vrijednosti te za primjer ovog autoenkodera iznosi 0.7343.

Slika 4.1. prikazuje matrice zabune modela logističke regresije, stabla odluke, XGBoosta, CatBoosta i autoenkodera redom s lijeva na desno, odozgora prema dolje, matrice su rađene na testnom skupu.



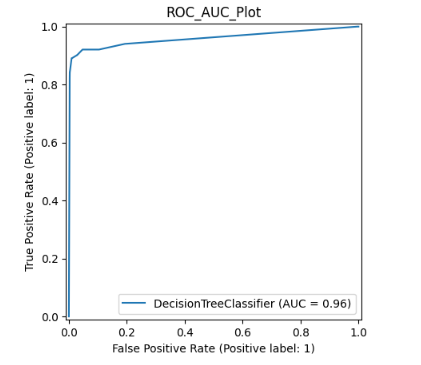
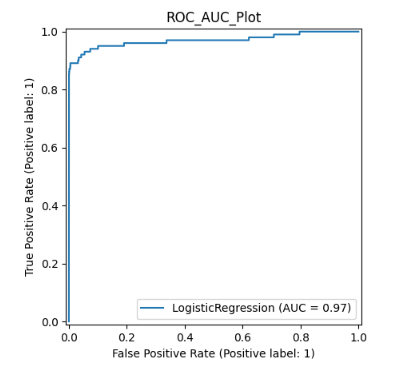


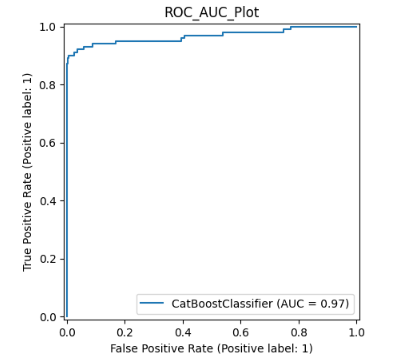
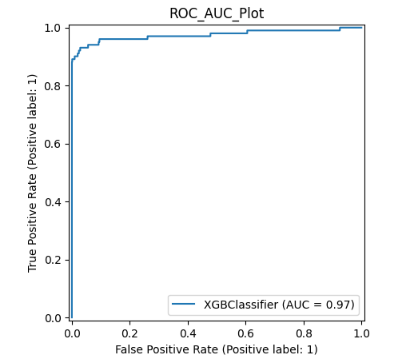


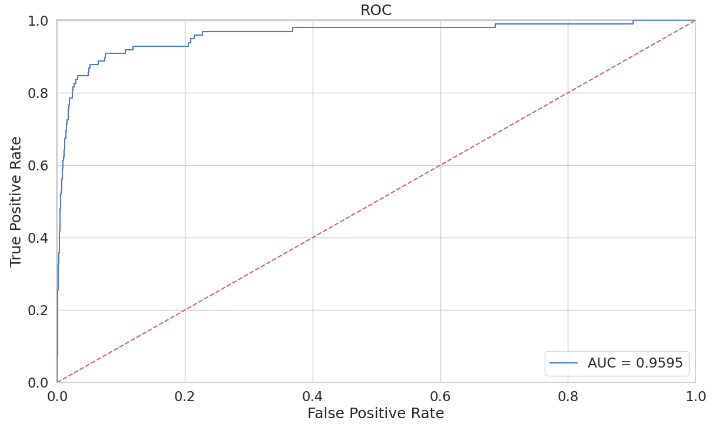
Slika 4.1. Matrice zabune modela

Svi korišteni modeli imaju slične rezultate evaluacije. XGBoost u odnosu na druge modele odvaja se sa svojom vrijednosti preciznosti, no za otkrivanje prijevara bitniji je odziv, uz pretpostavku da je bolje stvarnu transakciju označiti kao prijevaru nego ne prepoznati prijevaru i označiti ju kao stvarnu transakciju. Odziv pokazuje koliki je dio stvarno negativnih transakcija model dobro označio. S obzirom na to najbolje rezultate daje stablo odluke, iako i drugi modeli imaju slične performanse. To se dobro može vidjeti iz matrica zabune (1 označuje transakcije koje su prijevara, a 0 označuje stvarne transakcije), no i vrijednosti preciznosti i odziva. XGBoost je napravio pogrešnu procjenu na svega 26 primjera, odnosno 16 ih je označio kao lažno pozitivne, a 12 kao lažno negativne, stablo odluke označilo je 10 primjera kao lažno negativne, no 1543 kao lažno pozitivne. Kada se uzme u obzir velika razlika između lažno pozitivnih primjera i mala razlika između točno pozitivnih primjera, XGBoost pokazao se kao najbolji algoritam. Slične rezultate ima i CatBoost klasifikator, dok se logistička regresija kao i stablo odluke pokazala puno lošije u usporedbi s XGBoost i CatBoost kod označavanja lažno pozitivnih primjera jer je logistička regresija označila 1108 stvarnu transakciju kao prijevaru.

Slika 4.2. prikazuje ROC-AUC krivulje logističke regresije, stabla odluke, XGBoosta, CatBoosta i autoenkodera redom s lijeva na desno, odozgora prema dolje.







Slika 4.2. ROC-AUC krivulje modela

U tablici 4.2. prikazane su vrijednosti ROC-AUC, AUC i Cross validation ocjene za algoritme nadziranog učenja.

|  |  |  |  |
| --- | --- | --- | --- |
|  | Cross validation | ROC-AUC | AUC |
| Logistička regresija | 99.46% | 93.72% | 0.97 |
| Stablo odluke | 98.73% | 93.69% | 0.96 |
| XGBoost | 100% | 94.05% | 0.97 |
| CatBoost | 100% | 92.53% | 0.97 |

Tablica 4.2. vrijednosti ROC-AUC, AUC i Cross validation

U tablici 4.3. Prikazne su vrijednosti ROC-AUC i AUC za autoenkoder

|  |  |  |
| --- | --- | --- |
|  | ROC-AUC | AUC |
| Autoenkoder | 89.21% | 0.96 |

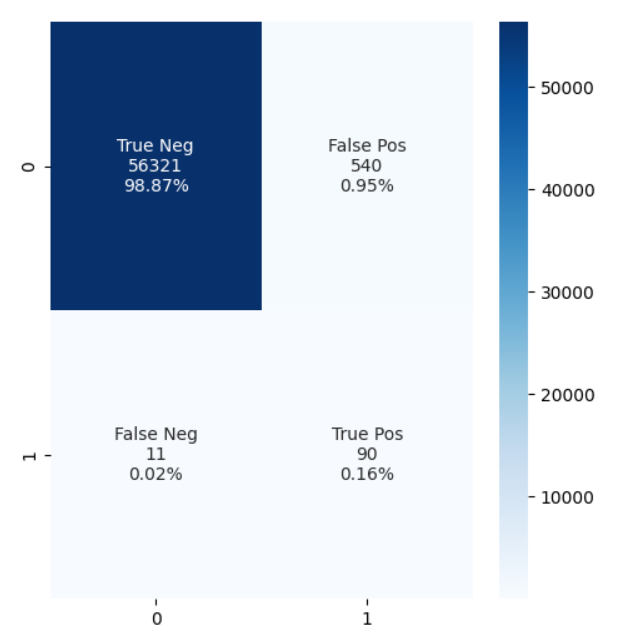
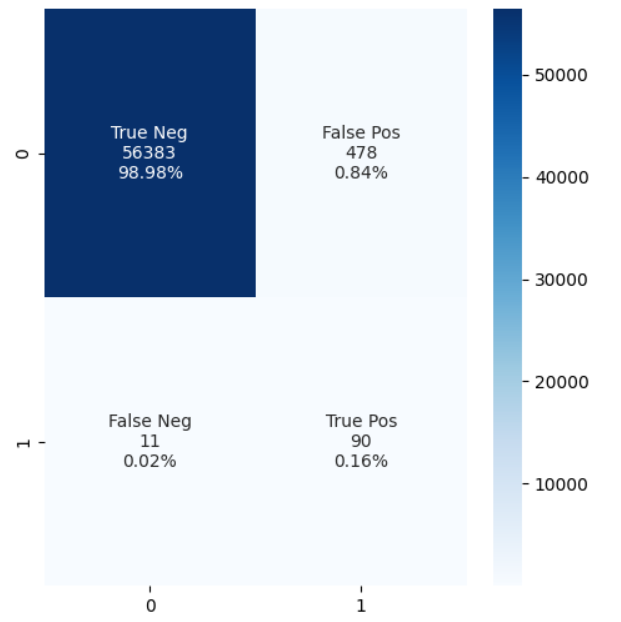
Tablica 4.3. Vrijednosti ROC-AUC i AUC za autoenkoder

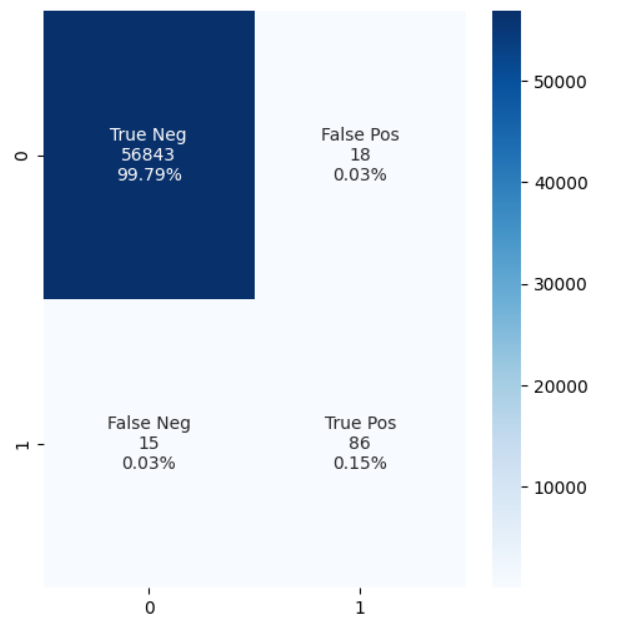
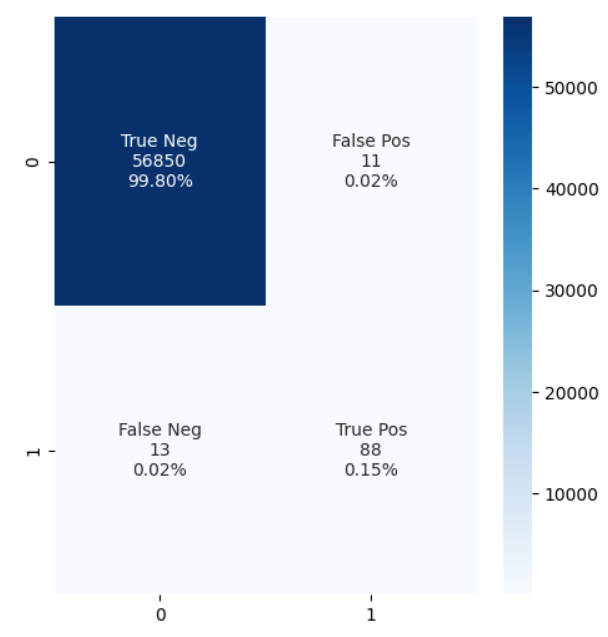
Kako bi se dobio bolji uvid u same modele postavljena je prag na svakom od modela za ROC-AUC krivulje. Za sve modele granica odluke postavljena je na vrijednost 0.8, izračunati su TPR (engl. True postive rate) i FPR (engl. False postive rate) za tu vrijednost te su dane matrice zabune. Tablica 4.3. prikazuje vrijednosti TPR i FPR za modele nadziranog učenja za prag 0.8.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Logistička regresija | Stablo odluke | XGBoost | CatBoost |
| TPR | 0.8911 | 0.8911 | 0.8713 | 0.8515 |
| FPR | 0.0084 | 0.0095 | 0.0002 | 0.0003 |

Tablica 4.3. Vrijednosti TPR i FPR za modele nadziranog učenja za prag 0.8

Slika 4.3. prikazuje matrice zabune za prag 0.8.





Slika 4.3. Matrice zabune za prag 0.8

Nakon postavljanja praga na ROC-AUC krivulje i ponovnog računanja matrica zabune vidljivo je da su točno klasificirane vrijednosti i dalje podjednake za svaki od primjera, što se može vidjeti i iz TPR-a, no vidi se velika razlika u klasificiranju lažno pozitivnih primjera kada se postavi prag na ROC-AUC krivulje kod kojih su korišteni modeli logističke regresije i stabla odluke.

# ZAKLJUČAK

U ovom seminaru o otkrivanju prijevara u kartičnim transakcijama korišteni su različiti pristupi strojnog učenja kako bi se riješio problem prepoznavanja prijevara. Upotreba logističke regresije, stabla odluke, XGBoosta i autoenkodera omogućila je dublje razumijevanje raznolikosti tehnika u rješavanju problema detekcije prijevara.

Analizom rezultata ustanovljeno je da se XGBoost klasifikator istaknuo kao najučinkovitiji model u kontekstu ovog problema. Dodatno, prikaz rada autoenkodera, tehnike nenadziranog učenja, pružio je dodatnu perspektivu o mogućnostima otkrivanja anomalija u podacima. Iako se u ovom slučaju XGBoost pokazao kao najbolji, uz optimizaciju parametra i dodatne metode moguće je i drugim modelima, kao i XGBoost dodatno povećati performanse.

# LITERATURA

[1] Credit Card Fraud Detection – podatkovni skup- <https://www.kaggle.com/datasets/mlg-ulb/creditcardfraud>

[2] SMOTE for Imbalanced Classification with Python - <https://www.analyticsvidhya.com/blog/2020/10/overcoming-class-imbalance-using-smote-techniques/>

[3] Synthetic Minority Over-sampling Techniqu (SMOTE) - <https://medium.com/@corymaklin/synthetic-minority-over-sampling-technique-smote-7d419696b88c>

[4] What is logistic regresion? - <https://www.ibm.com/topics/logistic-regression>

[5] Decision Trees - <https://scikit-learn.org/stable/modules/tree.html>

[6] XGBoost documentation - <https://xgboost.readthedocs.io/en/stable/>

[7] What is CatBoost? - <https://builtin.com/machine-learning/catboost>

[8] Autoencoders – Machine Learing - <https://www.geeksforgeeks.org/auto-encoders/>

[9] Credit Card Fraud Detecting using Autoencoders in Keras - <https://venelinvalkov.medium.com/credit-card-fraud-detection-using-autoencoders-in-keras-tensorflow-for-hackers-part-vii-20e0c85301bd>