**Fakultet organizacionih nauka**

**Jove Ilića 15**



Modelovanje finansijskih sistema

Detekcija prevara kreditnih kartica primenom metode random forest

|  |  |
| --- | --- |
|  | Student: |
|  | Vojislav Stefanović 94/17 |
|  |  |
| Beograd, Jul 2021. |  |
|  |  |

Sadržaj

[**Uvod** 3](#_Toc75369733)

[**Učitavanje i prikaz podataka** 3](#_Toc75369734)

[**Deskriptivna statistika** 4](#_Toc75369735)

[**Standardizacija podataka** 7](#_Toc75369736)

[**Z skor** 7](#_Toc75369737)

[**Min / Max normalizacija** 7](#_Toc75369738)

[**Matrica korelacije** 9](#_Toc75369739)

[**Vizuelizacija podataka** 12](#_Toc75369740)

[**Preklapanje između vremena i prevare** 12](#_Toc75369741)

[**Preklapanje između veličine transakcije i prevare** 13](#_Toc75369742)

[**Oversampling i undersampling** 14](#_Toc75369743)

[**Analiza glavnih komponenti - PCA** 15](#_Toc75369744)

[**Nestandardne opservacije - outliers** 17](#_Toc75369745)

[**Detekcija autlajera** 17](#_Toc75369746)

[**Druge resampling metode** 18](#_Toc75369747)

[**SMOTE metoda - Synthetic Minority Oversampling Technique** 19](#_Toc75369748)

[**Slučajne šume** 22](#_Toc75369749)

[**Algoritam** 22](#_Toc75369750)

[**Pretpostavke** 22](#_Toc75369751)

[**Kada koristimo slučajne šume** 23](#_Toc75369752)

[**Primene Random Forest** 23](#_Toc75369753)

[**Primena metode random forest classification** 25](#_Toc75369754)

[**Početni data set** 25](#_Toc75369755)

[**Normalizovan data set** 29](#_Toc75369756)

[**Početni balansirani data set** 31](#_Toc75369757)

[**Normalizovani balansirani data set** 33](#_Toc75369758)

[**Zaključak** 36](#_Toc75369759)

[**Pregled literature** 39](#_Toc75369760)

[**„Detecting Credit Card Fraud by ANN and Logistic Regression”** 39](#_Toc75369761)

[**„Detecting Credit Card Fraud By Decision Trees And Support Vector Machines”** 40](#_Toc75369762)

[**„Random Forest For Credit Card Fraud Detection”** 41](#_Toc75369763)

[**“Adversarial Support Vector Machine Learning”** 42](#_Toc75369764)

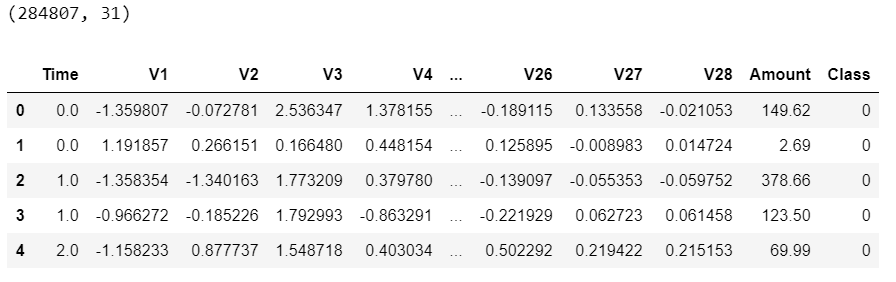
[**Literatura** 43](#_Toc75369765)

## **Uvod**

U ovom trenutku napredak interneta je ubrzao veliki broj industrija i procesa, među kojima je i finansijski sektor. Potražnja za brzim i jednostavnim pristupom novca je sve veća, međutim, sa tim dolazi i dosta rizika, jer proces verifikacije podataka mora biti brži i precizniji. Stoga jedan od problema kojim ćemo se baviti u ovom radu je upravo detekcija prevara pri plaćanju kreditnim karticama. Neophodno je što pre detektovati ove prevare da nedužna lica ne bi finansijski ispaštala, a broj transakcija je abnormalno viši nego ikada u istoriji čovečanstva. Potreba za efikasnim i efektivnim sistemom provere je ogromna.

Skup podataka (eng. dataset) nad kojim radimo je uveliko poznat u industriji i zove se: ”Credit Card Fraud Detection”. Podaci u ovom skupu su anonimizovani, imamo 30 ulaznih varijabli i jednu izlaznu. Svi podaci u skupu su numerički, i dobijeni su kroz analizu glavnih komponenti. Jedine varijable koje nisu transformisane su ulazne varijable “Time” i “Amount”, i izlazna varijabla “Class”. Zbog privatnosti podataka, nemamo pristup originalnom skupu podataka, već samo glavnim komponentama koje su dobijene kroz analizu i obeležene su sa V1,V2, …,V28. Varijabla “Time” predstavlja broj sekundi koje su prošle od prve transakcije u skupu. Varijabla “Amount” predstavlja iznos transakcije u dolarima. Varijabla “Class” predstavlja podatak da li je transakcija prevara ili ne, vrednost 0 u ukazuje da nije, dok vrednost 1 ukazuje da je došlo do prevare.

## **Učitavanje i prikaz podataka**



*Slika 1. Vizualni prikaz podataka iz zadatog Dataset-a*

Na ovoj slici je prikazan naš skup podataka. U njemu se nalaze podaci koje ćemo obrađivati u daljoj dokumentaciji. Poznate kolone su nam Time, Amount i Class, dok su kolone od V1 do V28 nama nepoznate. Ima 284.807 redova i 31 kolona ukupno.

## **Deskriptivna statistika**

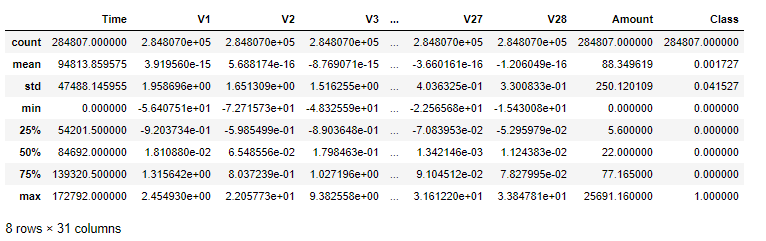
Deskriptivna statistička analiza predstavlja skup metoda kojima se vrši izračunavanje, prikazivanje i opisivanje osnovnih karakteristika statističkih serija.

Zadaci deskriptivne statistikčke analize su sledeći:

1. Grupisanje i sređivanje statističkih podataka.

2. Prikazivanje statističkih podataka.

3. Određivanje osnovnih pokazatelja statističkih serija.



*Slika 2. Vizualni prikaz deskriptivne statističke analize*

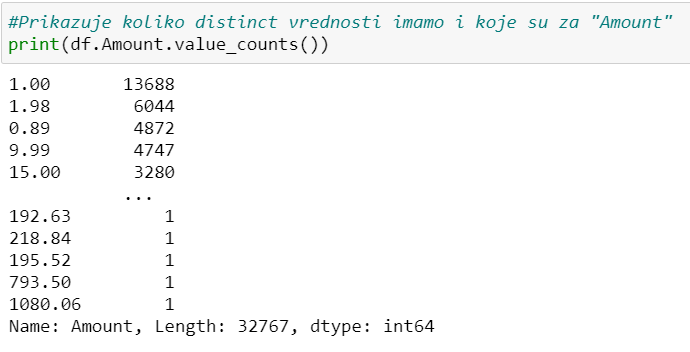
Pokretanjem sledeće linije koda možemo videti da li naš skup podataka ima neke nedostajuće vrednosti.



*Slika 3.Linija koda za prikaz nedostajućih vrednosti*

Kao što možemo videti, u našem skupu nema nikakvih nedostajućih vrednosti.

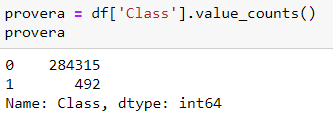
Dodatno, možemo videti, po kolonama, koliko imamo unikatnih vrednosti. Sledeća linija koda nam pokazuje unikatne vrednosti za kolonu “Amount”.



*Slika 4. Prikaz “distinct” vrednosti za naš skup podataka*

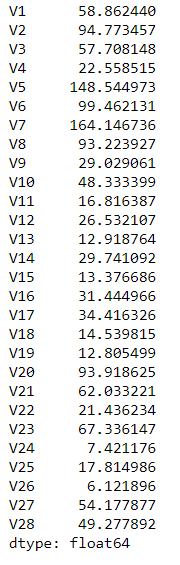
Možemo videti da se vrednost 1 pojavljuje 1.688 puta u našem skupu podataka, što je 0,64% vrednosti u našem skupu. U kasnijem delu procesa ova informacija nam može biti od koristi ako postoji korelisanost između ovih vrednosti i transakcija koje su obeležene kao “prevare”.

Takođe nakon filtriranja možemo uočiti da su sve vrednosti u koloni “Class” isključivo 0 i 1.



*Slika 5. Prikaz vrednosti u koloni “Class”*

Ovde je prikazan raspon vrednosti kolona od V1 do V28 među samim sobom, tj razlika između minimalnih i maksimalnih vrednosti za svaku kolonu:



*Slika 6. Raspon vrednosti za svaku kolonu*

Minimalna vrednost kolona od V1 do V28 se nalazi u koloni V5 i iznosi -113.743307, a maksimalna vrednost se nalazi u koloni V7 i iznosi 120.589494.

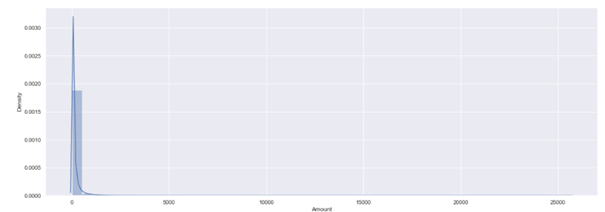
## **Standardizacija podataka**

Standardizacija je tehnika skaliranja, gde su vrednosti centrirane oko srednje vrednosti sa jedinicom standardne devijacije. To znači da sredina atributa postaje nula, a rezultujuća raspodela ima jedinicu standardne devijacije. Kako bismo dobili podatke koji su optimalni za korišćenje zbog samog algoritma, prvo ćemo odraditi standardizaciju podataka korišćenjem dve metode - Z skor i Min/Max standardizaciju.

### **Z skor**

Prvo ćemo odraditi Z skor nad kolonom “Amount”, kako bismo normalizovali podatke. Z skor ili normalizovano odstupanje je mera varijacije koja pokazuje odstupanje pojedinačne vrednosti obeležja od srednje vrednosti u broju standardnih devijacija.

Koristićemo sledeći kod: *z = stats.zscore(df['Amount'])*

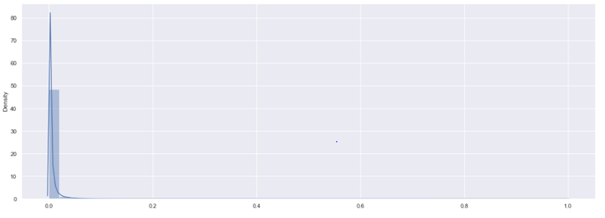


*Slika 7. Vizualizacija podataka nakon odrađenog Z skora*

Na prethodnoj slici možemo videti vizualizaciju podataka nad kolonom “Amount” nakon odrađene Z skor standardizacije. Z skor standardizacija se uglavnom koristi kada je potrebno zbog algoritma imati pretpostavku normalne raspodele.

### **Min / Max normalizacija**

Sada ćemo odraditi Min/Max normalizaciju nad kolonom “Amount”, koja nam je korisna kada svi podaci imaju pozitivne vrednosti, gde dobijamo normalizovane vrednosti u intervalu od 0 do 1. Koriščenjem funkcije MinMaxScaler(), dobijamo normalizovane podatke koje prikazujemo vizuelno radi bolje preglednosti.



*Slika 8. Vizualizacija podataka nakon odrađene Min/Max standardizacije*

Nakon vizualizacije rezultata, možemo uočiti sličnost između prikaza Z skor i Min/Max normalizacije, s tim što su podaci kod poslednje u intervalu između 0 i 1.

## 

## **Matrica korelacije**

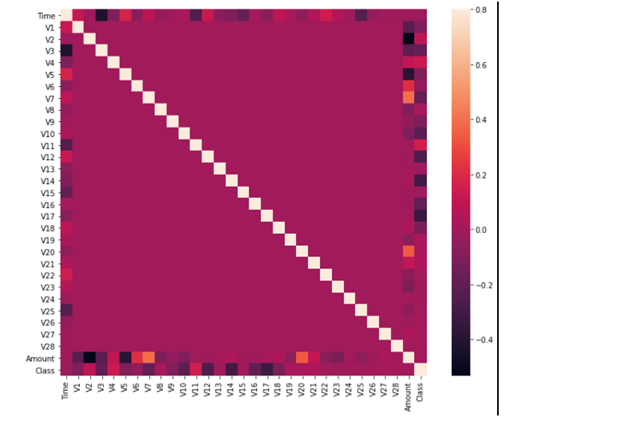
Ponekad nam u istraživanju nije dovoljna informacija o korelaciji dve posmatrane varijable, već nas zanima na koji način više varijabli međusobno utiče jedna na drugu. Nakon što se posmatranjem međusobnog odnosa svih parova dveju varijabli utvrdi njihova međusona korelacija, izrađuje se matrica korelacije.



*Slika 9 Matrica korelacije*

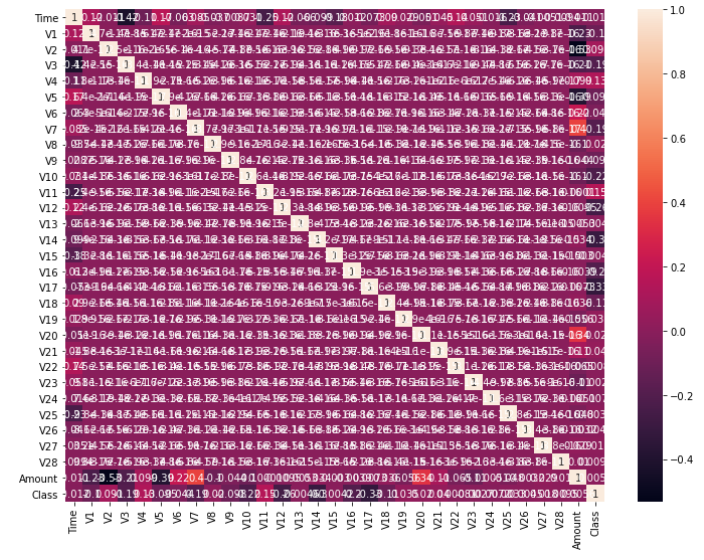
Uz pomoć funkcije df.corr(), kreirali smo korelacionu matricu sa našim podacima. Na osnovu ove tabele može se zaključiti da je kolona V9, kolona sa najvećom vrednošću korelacije.

Takođe ovu tabelu možemo grafički prikazati putem toplotne karte (heat map). Heat mapa je primer tehnike za data vizuelizaciju koja pokazuje posmatrani fenomen u dve dimenzije u boji. Intenzivnije boje nam pokazuju veću korelisanost između kolona tj. naših podataka.



*Slika 10. Heat map*

Alternativno možemo prikazati heat mapu sa više vrednosti i detalja.



*Slika 11. Detaljnija Heat map sa prikazanim decimalnim korelacijama*

Na osnovu grafika možemo zaključiti da su kolone V2 i V5 visoko negativno korelisane u odnosu na kolonu “Amount”. Takođe za kolonu V20 je izražen veliki nivo korelacije sa kolonom “Amount”.

# 

### 

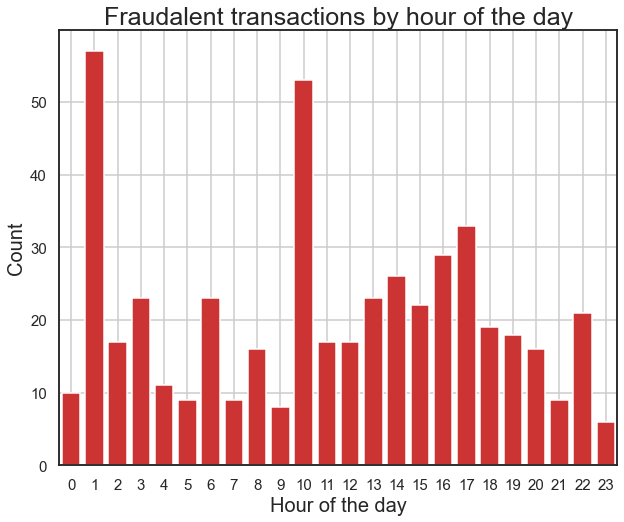
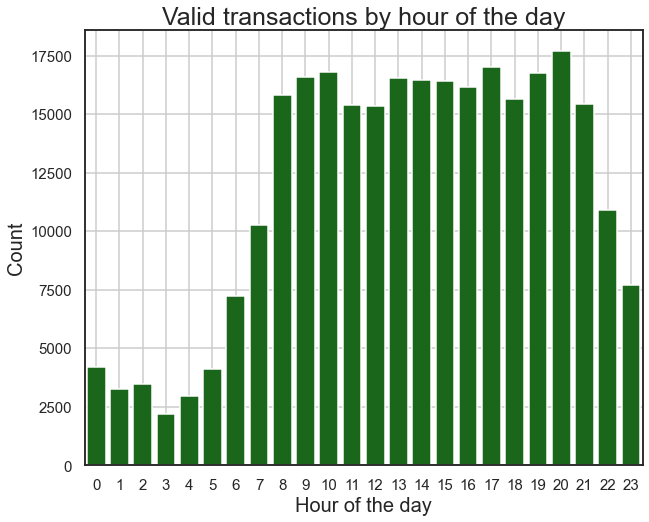
## **Vizuelizacija podataka**

Vizuelizacija podataka je odlična metoda za predstavljane podataka na intuitivan i lako prepoznatljiv način. Dosta nam pomaže u procesu upoznavanja samog skupa podataka. S obzirom na to da u našem skupu uglavnom radimo sa dosta nepoznatih parametara, vizuelizacijom možemo doći do njihovog pravog značenja za rešavanje našeg problema.

U ovom delu rada ćemo prikazati vizuelizaciju jedinih poznatih i ključnih varijabli kao što su “Time”, “Amount” i “Class”. Tačnije prikazaćemo da li postoji preklapanje između određenih suma novca kao i perioda vremena kad je izvršena transakcija, sa time da li je u pitanju prevara ili ne.

### **Preklapanje između vremena i prevare**

Pošto nam je varijabla “Time” data u sekundama, pretvorićemo je u sate radi lakseg razumevanja podataka. Nakon toga napravićemo dva grafika, na jednom ćemo prikazati validne transakcije,a drugom prevare.

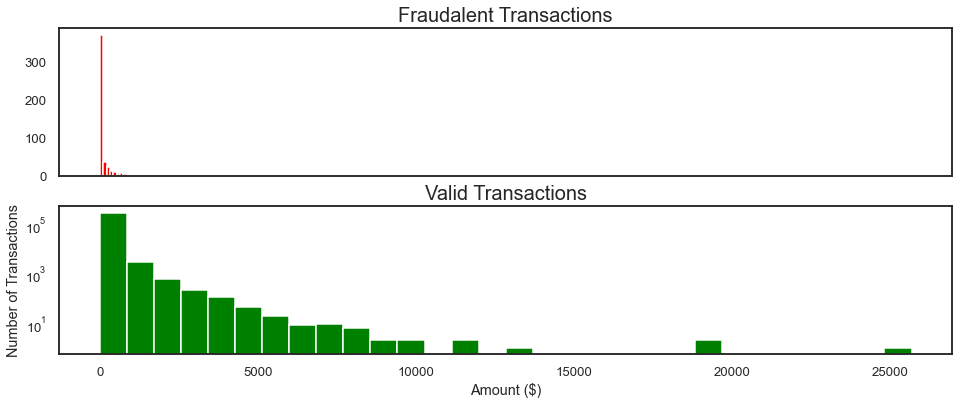


Možemo primetiti da u satima 1 i 10 imamo najviše prevara. U prvom satu količina ukupnih transakcija je dosta niska (2500-5000), dok u 10. satu su prevare “ugnježdene” unutar velike količine validnih transakcija (15000-17500).

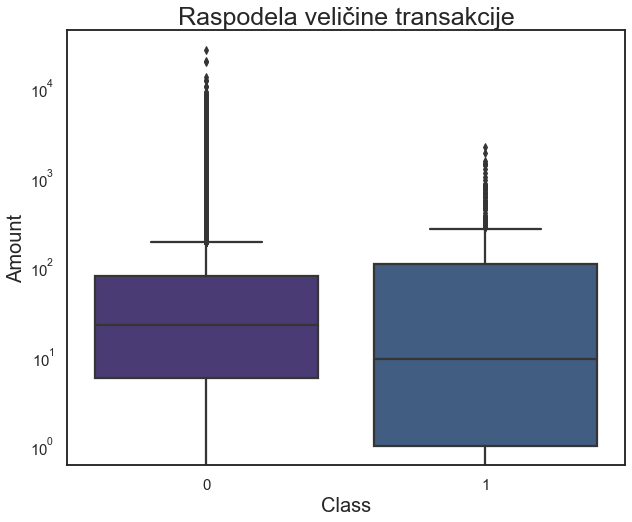
### 

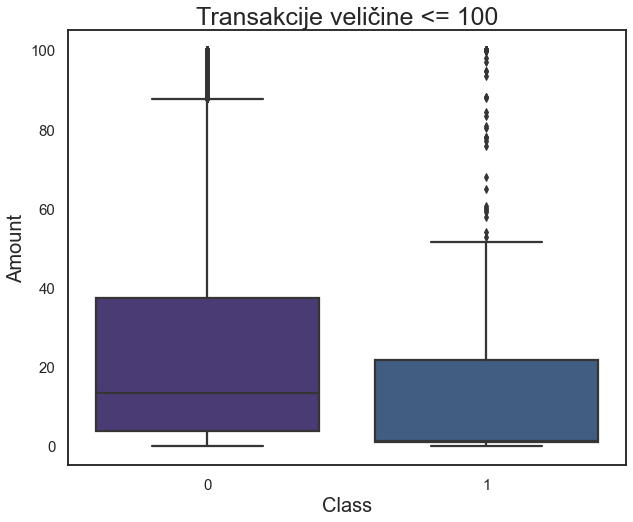
### **Preklapanje između veličine transakcije i prevare**

Za ovaj deo smo probali više vrsta vizuelizacije, jedan preko histograma i drugi preko boxplot-a. Histogram izgleda ovako:



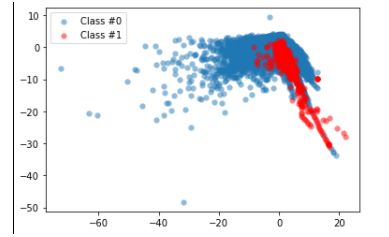
Histogram nam daje dobru sliku generalne raspodele obe vrste transakcija, i možemo videti da je velika većina prevara uglavnom u relativno niskom opsegu. Preko boxplot-a možemo videti malo precizniju sliku tačne količinske raspodele u tom nižem opsegu.

Ovde vidimo raspodelu prevara po kvatilima. Prva stvar koju možemo primetiti je, da je 50% prevara u količini manjoj od 10 dolara. Ostatak prevara se rasprostire do reda veličine stotine dolara. Imamo i autlajere koji vrše prevaru reda veličine hiljade dolara. Sad ćemo izdvojiti transakcije do 100 dolara, da bismo videli precizniju raspodelu u tom rasponu.

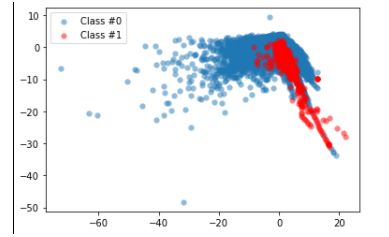
Ovde vidimo da je čak 75% prevarnih transakcija do 20,30 dolara, a ostalih 25% do 60 dolara. Sa ova dva boxplot-a možemo zaključiti da transakcije koje su klasifikovane kao prevare nemaju veliki iznos, odnosno uglavnom su manje od 100 dolara, i te koje su manje su većinom do 30 dolara.

## **Oversampling i undersampling**

Konvertovali smo data tabelu u 2 varijable x i y, x uzima vrednosti kolona od V1 do V28, a y uzima kolonu Class. Uz pomoć f-je *prep\_data* možemo kreirati x i y osu. Nakon kreiranja običnog grafika, koristimo resamplovanu verziju naših x i y promenljivih i upoređujemo te vrednosti i uočavamo over samplovane klastere podataka.



*Slika 12. Graf za x i y promenljive*



*Slika 13. Graf za x i y resamplovane promenljive*

Nakon sagledavanja oba grafa možemo uočiti razliku u intenzivnosti boja na grafiku br.2, što nam označava prisustvo outliera. Gde je intenzivnije (u ovom slučaju crvena boja) tu je došlo do over samplovanja.

## **Analiza glavnih komponenti - PCA**

Potreba za analizom glavnih komponenti u fazi pripreme podataka je ta što nam ona nudi da broj varijabli u skupu podataka značajno smanjimo na samo najbitnije, odatle naziv “Analiza glavnih komponenti”. Na ovom skupu podataka je već urađena analiza glavnih komponenti stoga nema potrebe ponovo je raditi, međutim, u ovom radu ćemo ipak opisati proces dolaženja do glavnih komponenti kao da radimo na “sirovom” skupu podataka.

Pri rađenju analize glavnih komponenti nije nam potrebna izlazna varijabla, već samo ulazne i to je, u neku ruku, prednost ove analize. Kroz nju samo gledamo skup podataka i na osnovu korelisanosti izvlačimo glavne komponente. Ovaj tip redukcije je linearan. Doduše, pretpostavka analize glavnih komponenti je to da su podaci standardizovani. Neki glavni koraci koje treba ispoštovati da bi analiza bila uspešna su sledeći:

1. Normalizacija/Standardizacija podataka
2. Izvršavanje analize (u Python jeziku)
3. Provera objašnjene varijabilnosti koju svaka komponenta nosi

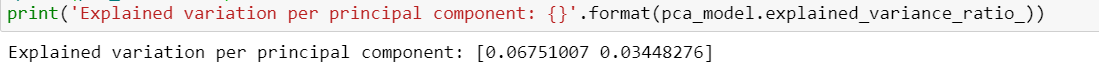
Normalizaciju i standardizaciju smo prešli u prethodnim delovima ovog rada. Što se tiče izvršavanja same analize, u Python-u ona izgleda ovako:



*Slika 14.Linija koda za analizu glavnih komponenti*

Kao što vidimo, paket “sklearn” nam dosta olakšava posao jer ima već ugrađenu analizu glavnih komponenti u sebe. Pozivom funkcije *PCA* potrebno je samo proslediti kao parametar broj komponenti koje želimo kao rezultat analize i naš model je završen. U sledećoj liniji koristimo funkciju *fit\_transform* i prosleđujemo normalizovan skup podataka. Ova funkcija ima dve bitne svrhe, prva je “fit” koja primenjuje naš model nad skupom podataka i traži glavne komponente, a “transform” transformiše naš skup podataka prema modelu.

Poslednji korak je provera objašnjene varijabilnosti komponenti, i izgleda ovako:

*Slika 15.Linija koda za proveru varijabilnosti komponenti*

Sam model ima atribut *explained\_variance\_ratio\_* koji nam upravo daje potrebne podatke u decimalnom obliku. Ispod imamo primer primene analize nad našim skupom podataka, gde se po samim brojkama može uočiti da je već rađena analiza glavnih komponenti.

Obično očekujemo da glavne komponente nose značajnu varijabilnost (reda veličine dvocifrenih procenata), dok u ovom slučaju vidimo da je to dosta niže (6% za prvu komponentu, 3% za drugu) i to nam ukazuje da sve varijable iz skupa podataka nose određenu značajnu težinu za predviđanje izlaza, odnosno da sve nose skoro sličan nivo varijabilnosti.

## **Nestandardne opservacije - outliers**

U analizi podataka mi pravimo predviđanja o nekoj određenoj populaciji. Iz tog razloga nam je bitno da nemamo ekstremne, nestandardne observacije, jer one mogu da naprave značajan uticaj na donošenje zaključaka.

Autlajer se može definisati kao opažanje koje mnogo odstupa od drugih opažanja u nekom skupu podataka. Iako su autlajeri često smatrani greškom ili šumom, oni mogu nositi važnu informaciju.

### **Detekcija autlajera**

Postoje razne metode za detekciju i uklanjanje autlajera. Najčešće su:

* Vizuelne (korišćenjem box plot, scatter plot...)
* Matematičke (Z-score, IQR...)

S’ obzirom da se u našem skupu podataka nalaze podaci koji su dosta nebalansirani i rastojanje između autlajera i običnih vrednosti je visoko, korišćenjem vizuelnih metoda nećemo dobiti preciznu sliku koji su tačno autlajeri. Te smo se odlučili da koristimo neku od matematičkih metoda, i to upravo Z-score, iz razloga što smo ga ranije u radu već primenjivali.

Na osnovu Z-score metode dolazimo do zaključka da sve kolone sadrže relativno mali procenat autlajera i on varira u rasponu od 0,41% za kolonu V13 do 1,67% za kolonu V27. Sledeći korak je da uklonimo te vrednosti.

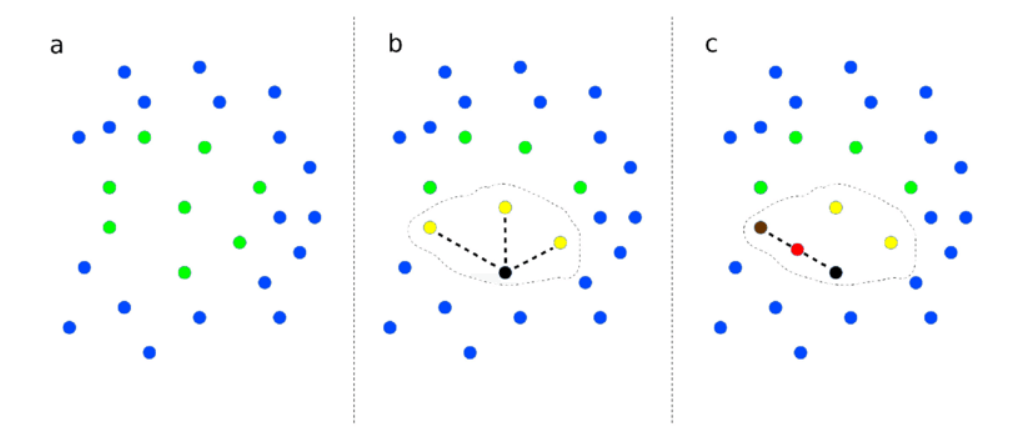
## **Druge resampling metode**

Postoje različite metode za resamplovanje podataka, a to su:

* **Random Undersampling (RUS):**
  + Ako postoji mnogo podataka i mnogo slučajeva manjina, tada bi poduzorkovanje moglo biti računski pogodnije
  + U većini slučajeva bacanje podataka nije poželjno
  + Uzima random poduzorak naše većinske klase da bi nadoknadio klasnu nebalansiranost. RUS prilagođava podatke tako što smanjuje većinsku klasu
* **Random Oversampling (ROS):**
  + Direktan je
  + Obradjuje model na brojnim duplikatima, veća verovatnoća za tačnošću konačnih ishoda
* **SMOTE:**
  + Sofisticiraniji
  + Radi se nad realnim skupom podataka
  + Obradjuje se nad sintetičkim skupom podataka, ne narušava se integritet realnog skupa
  + Dobro funkcioniše samo ako su osobine manjinskih slučajeva slične (ako se prevara širi kroz većinu podataka i nije različita)
  + Koristi najbliže susede za stvaranje više slučajeva prevara i unosi diskontinuitet u skup podataka, jer najbliži susedi možda neće biti slučajevi prevare

## **SMOTE metoda - Synthetic Minority Oversampling Technique**

SMOTE predstavlja strategiju ponovnog uzorkovanja za nebalansirane (neuravnotežene) skupove podataka. Ovo je jedan od načina na koje se može postići ravnoteža, tj balansiranost podataka. On koristi karakteristike najbližih suseda slučajeva prevara i stvara nove slučajeve sintetičkih prevara. Ovim izbegava dupliranje zapažanja. U suštini dodaje više vrednosti u manjinsku klasu.

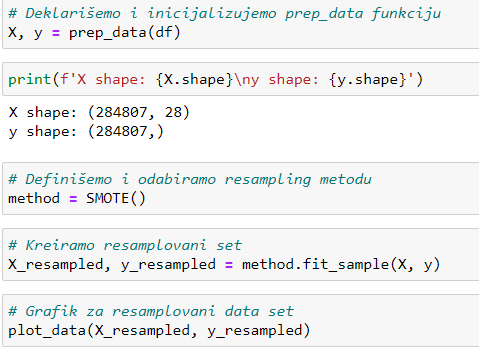


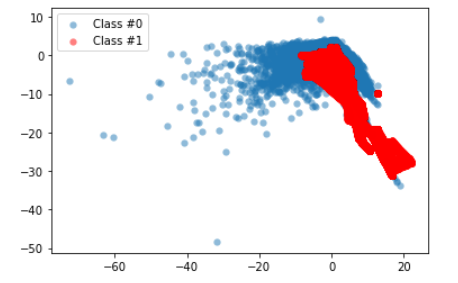
*Slika 16.Prikaz SMOTE metode*

Resampling metode se koriste kada radimo na training setu podataka, a ne na test setu. Cilj je da proizvedemo bolji model nad kojim radimo tako što pružamo balansirane podatke. Cilj nam nije da predvidimo sintetičke uzorke. Kada radimo resampling podaci o testovima ne bi trebalo da sadrže duplikate i sintetičke podatke, takođe testiranje modela se vrši samo nad stvarnim (realnim) podacima.

Koristeći tehniku SMOTE uravnotežili smo naše podatke. Za razliku od ROS-a, SMOTE ne stvara tačne kopije zapažanja već stvara nove sintetičke uzorke koji su slični postojećim zapažanjima u manjinskoj klasi. Skup podataka našeg data seta je dostupan i paketi koji su potrebni za radjenje SMOTE-a se mogu importovati u sam projekat.

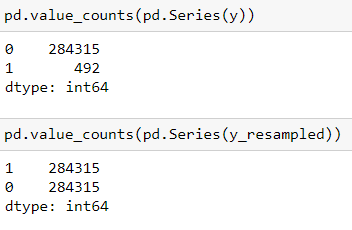
Da bismo dalje odradili i upustili se u SMOTE analizu potrebno je prvo da definišemo i odradimo prep\_data i plot\_data funkcije. Definisanjem funkcije plot\_data(X,y) dobićemo lepo iscrtani dati skup karakteristka X sa oznakama u formi scatter (raspršenog) grafa, sa druge strane prep\_data koristimo da bismo kreirali skup X i y varijable.

Način na koji je urađen SMOTE je sledeći:

*Slika 17 Segment koda za realizaciju SMOTE-a*

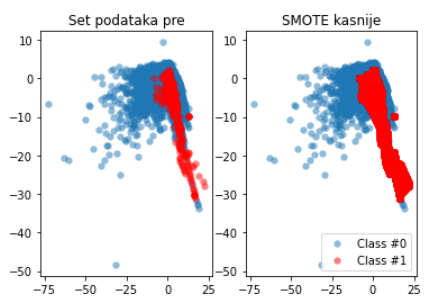
*Slika 18 Grafički prikaz za SMOTE*

Međutim ovaj grafik nam ne prikazuje jasnu sliku da li je SMOTE zaista odrađen nad našim data setom. Brzom proverom u Pythonu možemo ovo da verifikujemo, i dobijamo sledeće:



*Slika 19 Prikaz uvećanja instanci podataka u manjinskoj klasi*

Nakon izjednačavanja podataka za 0 i 1 radi lepšeg prikaza možemo uporediti naš originalni set i onaj nakon primene SMOTE-a. Međutim moramo da isprogramiramo funkciju za uporedni prikaz i definišemo njeno ponašanje.



*Slika 20 Komparativni prikaz rezultata*

## **Slučajne šume**

Slučajne šume ili nasumične šume su metoda učenja za klasifikaciju, regresiju i druge zadatke koja deluje konstruisanjem mnoštva stabala odluka u vreme treninga i iznošenjem klase koja je modifikovana klasa i/ili predviđanja srednje / prosečne vrednosti ( kod regresije) za pojedinačna stabla. Slučajne šume uglavnom nadmašuju stabla odlučivanja, ali njihova tačnost je niža od stabala podstaknutih gradijentom. Međutim, karakteristike podataka mogu uticati na njihove performanse. Što su složeniji podaci slučajne šume tj. random forest mora više da obrađuje podatke, time trošeći raspoloživu memoriju računara. Slučajna šuma je popularan algoritam. Može se koristiti i za probleme klasifikacije i regresije. Zasnovan je na konceptu ansambl učenja, što je proces kombinovanja više klasifikatora za rešavanje složenog problema i poboljšanje performansi modela. (Yiu, 2019)

### **Algoritam**

Kao što i samo ime govori, „Slučajna šuma je klasifikator koji sadrži niz stabala odluka na različitim podskupovima datog skupa podataka i uzima prosek da bi poboljšao prediktivnu tačnost tog skupa podataka.“ Umesto da se oslanja na jedno stablo odluke, slučajna šuma uzima predviđanje sa svakog drveta i na osnovu većine glasova predviđanja predviđa konačni rezultat (Pratt, 2018). Konkretno, drveće koje se trenira vrlo temeljno ima tendenciju da uči veoma nepravilne obrasce: oni prekomerno opremljuju svoje setove za obuku, tj. Imaju malu pristrasnost, ali vrlo veliku varijansu. Slučajne šume su način za uprosečavanje više stabala, obučenih na različitim delovima istog seta za obuku, sa ciljem smanjenja varijanse. To dolazi na račun malog povećanja pristrasnosti i određeni gubitak interpretabilnosti, ali generalno uveliko poboljšava performanse konačnog modela.

Timski rad mnogih stabala poboljšava performanse jednog slučajnog stabla. Iako nisu sasvim slične, šume daju efekte K-fold unakrsne validacije.

### **Pretpostavke**

Budući da slučajna šuma kombinuje više stabala za predviđanje klase skupa podataka, moguće je da neka stabla odluka mogu predvideti tačan izlaz, dok druga ne. Ali zajedno, sva stabla predviđaju tačan izlaz. Prema tome, dole su dve pretpostavke za bolji klasifikator slučajnih šuma:

* Trebalo bi da postoje neke stvarne vrednosti u promenljivoj obeležja skupa podataka tako da klasifikator može predvideti tačne rezultate, a ne pretpostavljeni rezultat.
* Predviđanja sa svakog drveta moraju imati vrlo niske korelacije.

### **Kada koristimo slučajne šume**

1. Potrebno je manje vremena za obuku u poređenju sa drugim algoritmima.
2. Predviđa izlaz sa velikom tačnošću, čak i za veliki skup podataka koji se efikasno koristi.
3. Takođe može da održi tačnost kada nedostaje veliki deo podataka.

Random Forest radi u dve faze, prvo je stvaranje slučajne šume kombinovanjem N stabla odluka, a drugo je davanje predviđanja za svako drvo stvoreno u prvoj fazi.

Proces rada se može objasniti u sledećim koracima i dijagramom:

Korak 1: Izaberite slučajne K tačke podataka iz skupa treninga.

Korak 2: Izgradite stabla odluka povezana sa odabranim tačkama podataka (podskupovi).

Korak 3: Izaberite broj N za stabla odlučivanja koja želite da napravite.

Korak 4: Ponovite korake 1 i 2.

Korak 5: Za nove tačke podataka pronađite predviđanja svakog stabla odluka i dodelite nove tačke podacima kategoriji koja osvaja većinu glasova.

### **Primene Random Forest**

Postoje uglavnom četiri sektora u kojima se uglavnom koristi slučajna šuma:

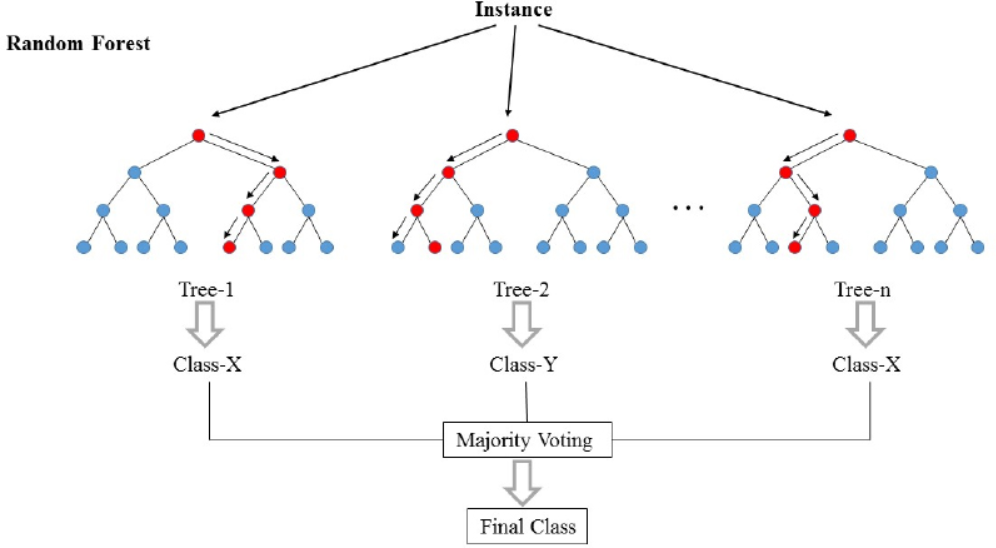
* Bankarstvo: Bankarski sektor uglavnom koristi ovaj algoritam za identifikaciju kreditnog rizika.
* Medicina: Pomoću ovog algoritma mogu se identifikovati trendovi bolesti i rizici bolesti.
* Upotreba zemljišta: Pomoću ovog algoritma možemo identifikovati područja slične upotrebe zemljišta.
* Marketing: Marketinški trendovi se mogu identifikovati pomoću ovog algoritma.

#### **Prednosti:**

* Prevazilazi problem prekomerne opreme usrednjavanjem ili kombinovanjem rezultata različitih stabala odlučivanja.
* Slučajne šume dobro funkcionišu za širok spektar podataka nego što to čini jedno stablo odlučivanja.
* Slučajna šuma ima manje varijanse od stabla pojedinačne odluke.
* Slučajne šume su vrlo fleksibilne i poseduju vrlo visoku tačnost.
* Skaliranje podataka nije potrebno u algoritmu slučajnih šuma. Održava dobru tačnost čak i nakon pružanja podataka bez skaliranja.
* Random Forest algoritmi održavaju dobru tačnost čak i kada nedostaje veliki deo podataka.

#### **Nedostaci:**

* Kompleksnost je glavni nedostatak algoritama slučajnih šuma.
* Izgradnja nasumičnih šuma je mnogo teža i dugotrajnija od stabala odlučivanja.
* Za primenu algoritma Random Forest potrebno je više računarskih resursa.
* Manje je intuitivan u slučaju kada imamo veliku kolekciju stabala odluka.
* Proces predviđanja korišćenjem slučajnih šuma veoma je dugotrajan u poređenju sa drugim algoritmima (North, 2021).

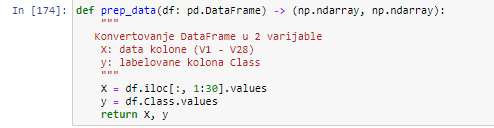
Primer: 

Slika 21 Grafički prikaz šume stabala

## **Primena metode random forest classification**

### **Početni data set**

Pre nego što primenimo ovu metodu klasifikacije na našem početnom data setu moraćemo da kreiramo prep data funkciju. Definisaćemo promenljive x i y koje predstavljaju vrednosti kolona od V1 do V28 za (x), i kolonu Class za (y) (McKinney, 2019).



Slika 22 Priprema data seta

Posle ovoga sledi podela data seta na training i test uzorak.



Slika 23 Podela seta na test i train

U ovom koraku mi odabiramo kao naš model random forest metodu.



Slika 24 Primena random forest algoritma

Potom vršimo predikciju modela i izračunavamo metrike za zadati data set.

Metrike koje se sa dosta uspeha koriste kada je potrebno otkriti jednu od klasa

su preciznost i odziv. Preciznost (eng. precision) je procenat stvarno pozitivnih

objekata među objektima koji su klasifikovani kao pozitivni i za binarnu klasifikaciju

definiše se formulom:

Preciznost = =

Veća preciznost takođe znači da postoji manji broj lažno pozitivnih objekata.

Odziv (eng. recall) predstavlja procenat pozitivnih objekata koji su ispravno klasifikovani i za binarnu klasifikaciju definiše se formulom:

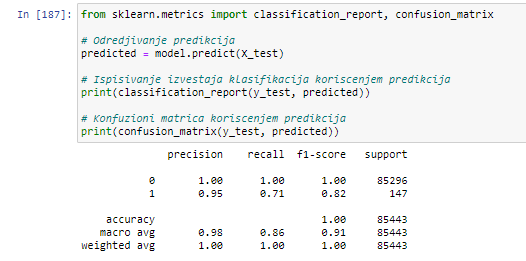
Odziv = =

Visok odziv znači da postoji mali broj lažno negativnih objekata. Preciznost

i odziv se mogu sumirati u metriku koja se zove 𝐹1 𝑚𝑒𝑟𝑎 (eng. 𝐹1 measure) i za

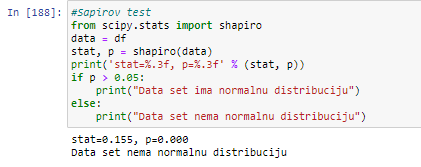
binarnu klasifikaciju definiše se formulom (Vidič, 2019).

𝐹1 𝑚𝑒𝑟𝑎 = =



Slika 25 Prikaz metrika za početni data set

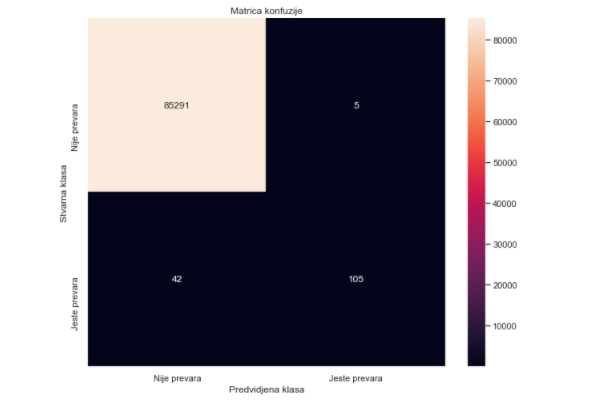
Šapiro – Wilk test predstavlja test za proveru normalne raspodele u statistici. Projektovan je tako da registruje sva odudaranja od normalnosti. Ukoliko je vrednost p>0.5 hipoteza Šapirovog testa je zadovoljena i ta kolona ima normalnu raspodelu, a ukoliko je manja onda kolona nema normalnu raspodelu.



Slika 26 Primena Šapirovog testa na početnom data setu

Matrica konfuzije je tabelarni prikaz brojeva ispravno i pogrešno klasifikovanih objekata na osnovu kojih se mogu vršiti ocene modela klasifikacije. Za problem binarne klasifikacije sa nejednako zastupljenim klasama, ređe zastupljena klasa često nosi naziv pozitivna klasa, a zastupljenija klasa nosi naziv negativna klasa. Vrednosti matrice imaju sledeće značenje: Stvarno pozitivni (TP - true positive) ili 𝑓11 je broj objekata koji pripadaju pozitivnoj klasi, a dodeljena im je pozitivna klasa. Lažno negativni (FN - false negative) ili 𝑓10 je broj objekata koji pripadaju pozitivnoj klasi a dodeljena im je negativna klasa. Lažno pozitivni (FP - false positive) ili 𝑓01 je broj objekata koji pripadaju negativnoj klasi a dodeljena im je pozitivna klasa. Stvarno negativni (TN - true negative) ili 𝑓00 je broj objekata koji pripadaju negativnoj klasi a dodeljena im je negativna klasa.

U našem radu vrednosti su drugačije dodeljene (Amin, 2020).



Slika 27 Matrica konfuzije za početni data set

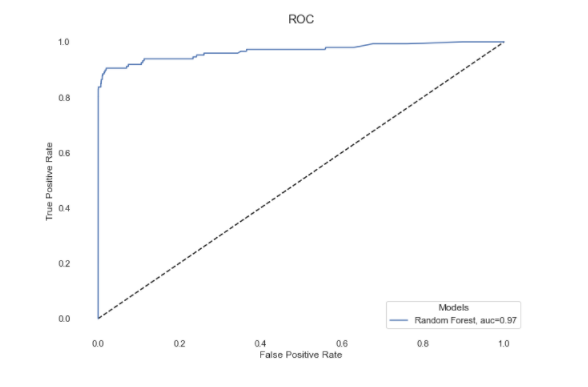
Receiver operating characteristic curve ili ROC kriva je grafik ili kriva koja ilustruje dijagnostičku sposobnost binarnog klasifikacionog siste

ma . Ona pokazuje 2 parametra, True Positive Rate I False Positive Rate. Metoda je originalno razvijena za programere koji su kontrolisali I nadgledali vojne radare 1941. godine, odatle je metoda I dobila svoj naziv.

**True Positive Rate** (**TPR**) je sinonim za recall I definiše se kao:

**False Positive Rate** (**FPR**) se definiše kao:

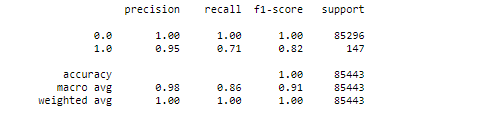
ROC kriva prikazuje odnose TPR I FPR u različitim klasifikacionim fazama. Ukoliko spuštamo rang za klasifikaciju, klasifikovaćemo više stavki kao pozitivne, time povećavajući I False Postitive I True Positive vrednosti.



Slika 28 ROC kriva za početni data set

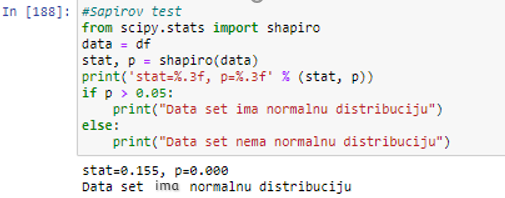
### **Normalizovan data set**

Metrike koje dobijamo na osnovu seta sa normalizovanim podacima su:



Slika 29 Prikaz metrika za normalizovani data set

Dobijeni rezultati za Šapirov test nad normalizovanim podacima su:



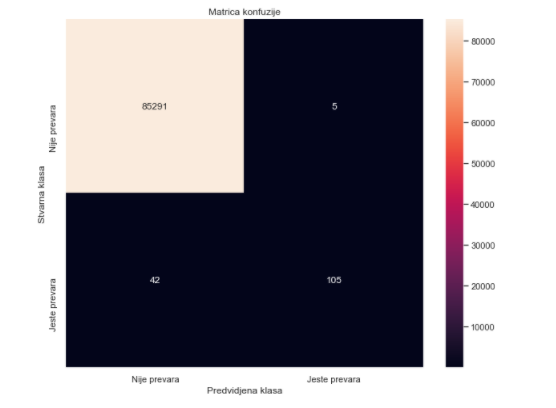
Slika 30 Primena Šapirovog testa na normalizovanom data setu

Na osnovu dole priložene matrice konfuzije zaključujemo da je metoda uspela da prepozna I predpostavi u 85291 slučajeva da nije prevara, što je bilo validno.

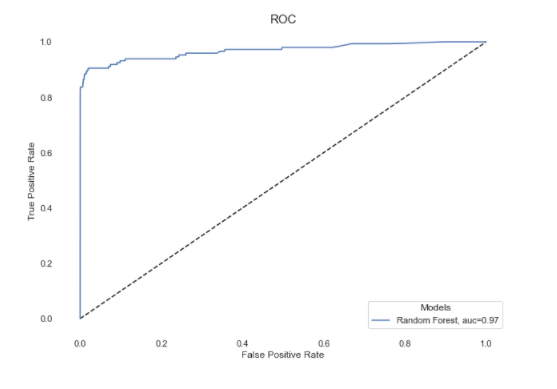
Predvidela je u 5 slučajeva prevaru, a ispostavilo se da nije.

U 42 je smatrala da nije prevara, a ispostavilo se da jeste.

I na kraju u 105 slučajeva metoda je predvidela prevaru, što se i ispostavilo kao tačno (Amin, 2020).



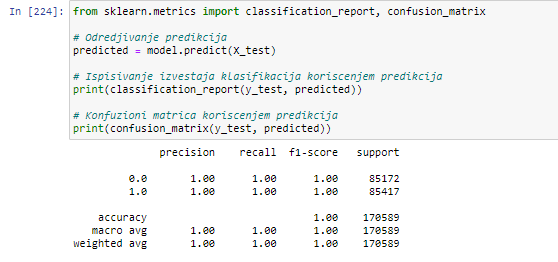
Slika 31 Matrica konfuzije za normalizovani data set

ROC kriva za normalizovane podatke: 

Slika 32 ROC kriva za normalizovani data set

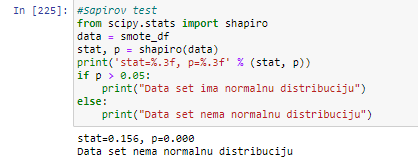
### **Početni balansirani data set**

Metrike koje dobijamo na osnovu seta sa balansiranim podacima su:



Slika 33 Prikaz metrika za početni balansirani data set

Dobijeni rezultati za Šapirov test nad balansiranim podacima su:



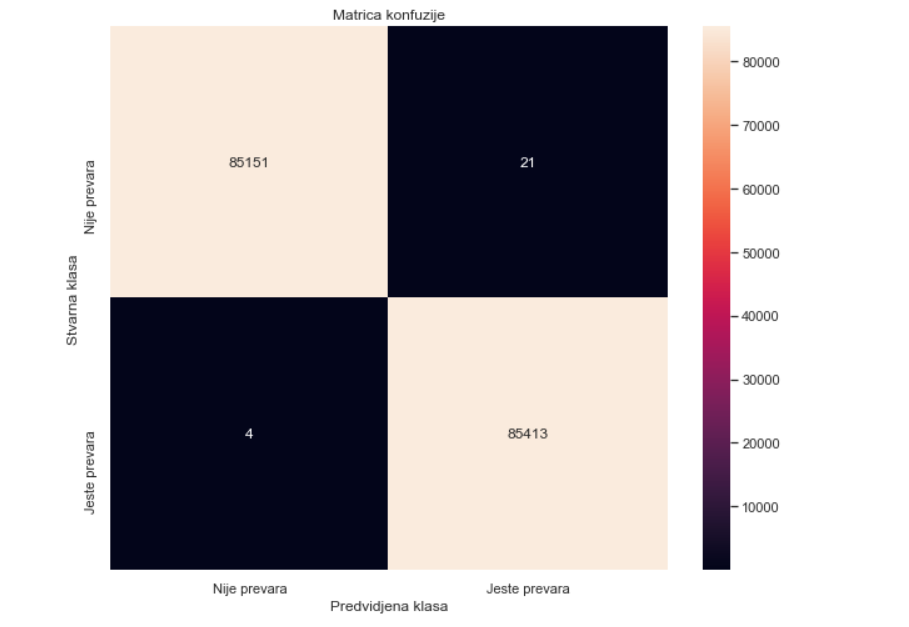
Slika 34 Primena Šapirovog testa na početnom balansiranom data setu

Na osnovu dole priložene matrice konfuzije zaključujemo da je metoda uspela da prepozna I predpostavi u 85147 slučajeva da nije prevara, što je bilo validno.

Predvidela je u 25 slučajeva prevaru, a ispostavilo se da nije.

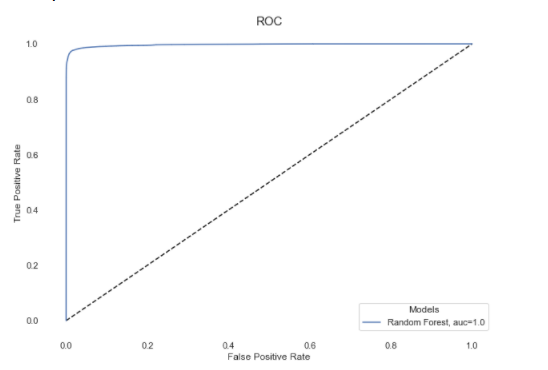
U 4 je smatrala da nije prevara, a ispostavilo se da jeste.

I na kraju u 85413 slučajeva metoda je predvidela prevaru, što se i ispostavilo kao tačno (Amin, 2020).



Slika 35 Matrica konfuzije za početni balansirani data set

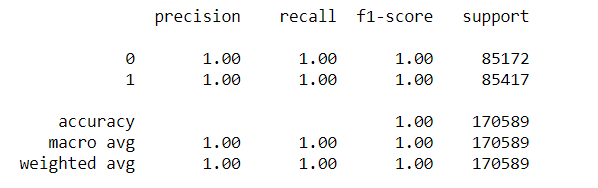
Vrednosti ROC krive su se promenile u odnosu na početni i normalizovani data set.



Slika 36 ROC kriva za početni balansirani data set

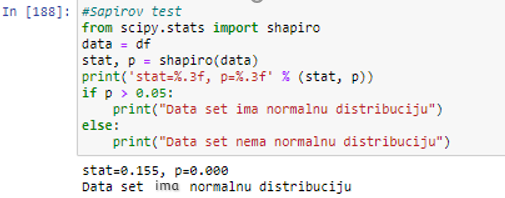
### **Normalizovani balansirani data set**

Metrike koje dobijamo na osnovu seta sa normalizovnim balansiranim podacima su:



Slika 37 Prikaz metrika za normalizovani balansirani data set

Dobijeni rezultati za Šapirov test nad balansiranim podacima su: Dobijeni rezultati za Šapirov test nad normalizovanim balansiranim podacima su:



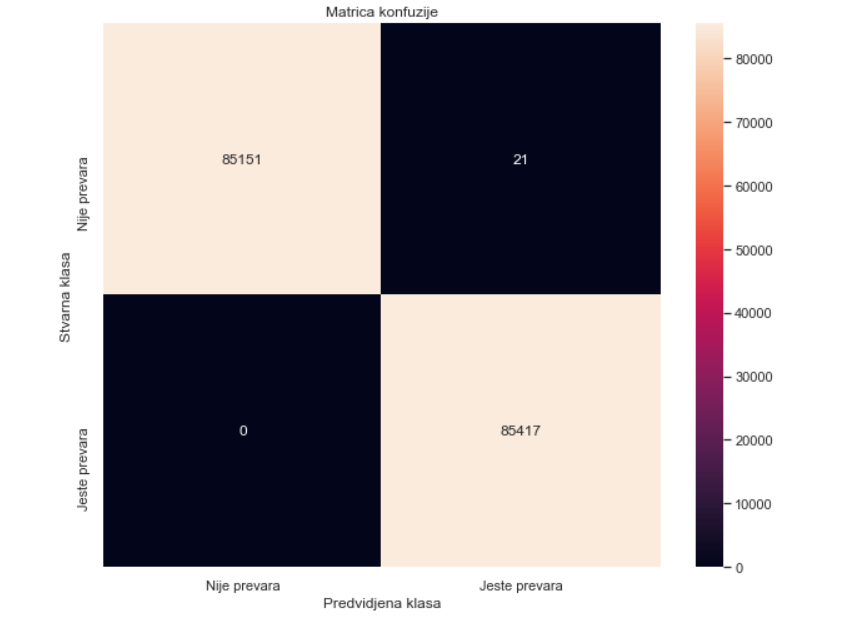
Slika 38 Primena Šapirovog testa na normalizovanom balansiranom data setu

Na osnovu dole priložene matrice konfuzije zaključujemo da je metoda uspela da prepozna I predpostavi u 85151 slučajeva da nije prevara, što je bilo validno.

Predvidela je u 21 slučajeva prevaru, a ispostavilo se da nije.

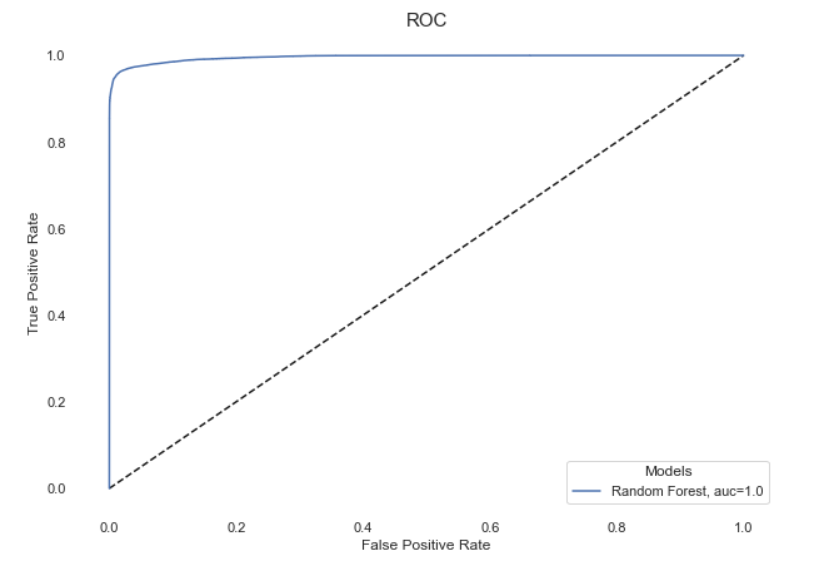
U 0 je smatrala da nije prevara, a ispostavilo se da jeste.

I na kraju u 85417 slučajeva metoda je predvidela prevaru, što se i ispostavilo kao tačno (Amin, 2020).



Slika 39 Matrica konfuzije za normalizovani balansirani data set

Vrednosti ROC krive su:



Slika 40 ROC kriva za normalizovani balansirani data set

## **Zaključak**

|  |  |  |
| --- | --- | --- |
| **Uporedni prikaz: Matrica konfuzije** | Nenormalizovan | Normalizovan |
| Nebalansiran |  |  |
| Balansiran |  |  |

Tabela 1 Matrica konfuzije - uporedni prikaz

|  |  |  |
| --- | --- | --- |
| **Uporedni prikaz: ROC kriva** | Nenormalizovan | Normalizovan |
| Nebalansiran |  |  |
| Balansiran |  |  |

Tabela 2 ROC kriva - uporedni prikaz

Upoređivanje metrika za sva 3 data seta:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Početni data set | Normalizovan | Početni Balansirani | Normalizovani Balansirani |
| Precision | 0.95 | 0.95 | 1 | 1 |
| Recall | 0.71 | 0.71 | 1 | 1 |
| F1-score | 0.82 | 0.82 | 1 | 1 |
| AUC | 0.97 | 0.97 | 1 | 1 |
| Accuracy | 0.999426 | 0.999438 | 0.974 | 0.960 |

Na osnovu dobijenih podataka našeg data seta možemo zaključiti da prilikom obrade početnog i normalizovanog data seta **precision** iznosi 0.95 dok se prilikom balansiranih data setova ta vrednost povećala na 1, čime zaključujemo da što je veća preciznost to je manji broj lažno pozitivnih vrednosti u ukupnom setu podataka.

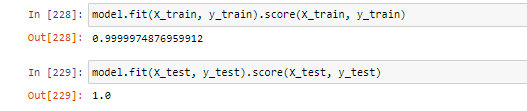
Prilikom analize **recall** metrike uvideli smo da je vrednost za početni i normalizovan 0.71, a za balansirane 1, na osnovu ovoga zaključujemo što je veća vrednost manji je broj lažno negativnih vrednosti za slučajeve prevare.

**F1-score** nam predstavlja međusoban odnos između **precision i recall** metrike, možemo primetiti da za početan data set i normalizovan iznosti 0.82, a u balansiranim iznosi 1.

**AUC** (area under curve) vrednosti smo dobili prilikom izrade ROC krive, auc predstavlja stepen ili meru separabilnosti, ona govori koliko je model sposoban da pravi razliku između klasa. Što je veća vrednost auc-a to model bolje predviđa 0 klase kao 0 i 1 klase kao 1.

**Accuracy** predstavlja tačnost našeg modela i ova tačnost u naša tri data seta iznosi 0.999426, 0.999438, 0.974 i 0.960 respektivno. Kao što vidimo preciznost se povećala nad normalizovanim data setom u odnosu na ostala dva.

Training i test error su izračunati i prikazani dole:



Slika 41 Greške treniranja i testiranja

Vrednost greške treniranja iznosi 0.999, a testiranja 1. Grešku treniranja dobijamo kada računamo grešku prilikom klasifikacije modela na istim podacima na kojima je model upravo bio treniran. Što je veća vrednost znači da je greška klasifikacije manja.

Greška testiranja nam ukazuje koliko će dobro model da se pokazuje kao efikasan na budućim podacima koje nije obrađivao.

## **Pregled literature**

U nastavku ćemo napraviti pregled najznačajnijih radova vezanih za problem detekcije prevara pri korišćenju kreditnih kartica. Ova tema je dosta obrađena i dostupan je veliki broj metoda klasifikacije za rešavanje problema. Sa fokusom na input, output, metodu klasifikacije i njenog rezultata, izdvojićemo najznačajnije radove koje se bave ovim problemom.

### **„Detecting Credit Card Fraud by ANN and Logistic Regression”**

U ovoj studiji, na problem otkrivanja prevara sa kreditnim karticama su primenjeni modeli klasifikacija zasnovani na veštačkoj neuronskoj mreži (Artificial Neuron Network – ANN) i logističkoj regresiji (LR u nastavku). Ova studija je jedna od prvih u kojoj su upoređivane performanse ANN i LR metoda u otkrivanju prevara sa kreditnim karticama sa stvarnim skupom podataka. U datom skupu podataka, svaki račun se posmatra odvojeno pomoću odgovarajućeg deksriptora, a klasifikator pokušava da predvidi da li će naredna transakcija biti normalna ili prevara – što čini izlaz modela. Svaki profil kartice sastoji se od promenljivih od kojih svaka otkriva ponašanje karakteristika upotrebe kartice. Ove promenljive mogu prikazivati potrošačke navike kupaca s obzirom na geografske lokacije, dani u mesecu, sati u danu ili kodovi kategorija koji pokazuju vrstu trgovca gde se transakcija odvija.

Period koji se koristio za izgradnju uzorka u ovoj studiji je uhvatio 978 prevara evidencije i 22 miliona normalnih sa odnosom oko 1: 22500.

Promenljive koje su korišćene kao ulaz su neke od pet glavnih informacija: statistika svih transakcija, regionalna statistika, MCC statistika, statistika dnevnog iznosa i dnevni broj transakcija.

U ovoj studiji su dobijena 13 alternativnih modela zasnovana na ANN i LR , koristeći odgovarajući set podataka (train) za svaki uzorak.

Što se tiče dobijenih rezultata, pokazalo se da su modeli dobijeni metodom ANN bolji od LR modela kada se upotrebe na test setu podataka. Na train setu podataka, njihove performanse su veoma slične (Sahin & Duman, 2011).

### **„Detecting Credit Card Fraud By Decision Trees And Support Vector Machines”**

Sa razvojem informacionih tehnologija i softvera u komunikacionim kanalima, koncept prevare se proširio po celom svetu, što je rezultiralo u ogromnim finansijskim gubicima. Iako su razvijeni mehanizmi za sprečavanje prevara kao što su CHIP i PIN, ovi mehanizmi ne sprečavaju najčešće tipove prevara kao što su prevara sa upotrebom kreditnih kartica preko virtuelnih POS terminala ili naloga za poštu. Kao rezultat, otkrivanje prevara je osnovni alat i verovatno najbolji način za zaustavljanje takvih vrsta prevara. U ovoj studiji su razvijeni i primenjeni modeli klasifikacije zasnovani na stablima odlučivanja i mašinama za vektore podrške (SVM) na problemu otkrivanja prevara sa kreditnim karticama. Ova studija je jedna od prvih koja je uporedila performanse SVM-a i metoda stabla odlučivanja u otkrivanju prevara sa kreditnim karticama sa stvarnim skupom podataka.

U ovoj studiji, sedam alternativnih modela zasnovanih na odluci algoritama stabla i SVM su izgrađeni koristeći odgovarajući set podataka o obuci za svaki uzorak. Da bismo procenili ove modele, koristili smo preostale transakcije u odgovarajućim testovima za svaki model. Lažne transakcije u skupovima testova uzoraka su bili identični. Stope tačnosti opisuju korisnost modela generalno. Tačnost je verovatno najčešće korišćena metrika za merenje performansi modela ciljanja u aplikacijama za klasifikaciju.

Međutim, tačnost pokazuje stopu tačnosti za sve zadatke bez obzira da li je u pitanju bio zadatak prevara ili random case model zadatak. Kada se pojavljivanja takvih zadataka na modelu uporede sa drugim, vidi se da se ukupan broj train data seta povećao. Kao posledicu ovo ponašanje postaje manje zapaženo i modeli performansi zasnovani na SVM-u postaju uporedivi sa stablom. Ali broj prevara koje je SVM uhvatio su i dalje daleko manje od prevara koje je uhvatilo i primetilo stabla odlučivanja, posebno C&RT model. Ipak, broj stvarnih lažnih transakcija koje su modeli dodelili kao lažne nisu u bliskoj vezi sa tačnošću zadataka koje su locirali ti isti modeli (Sahin & Duman, 2011).

### **„Random Forest For Credit Card Fraud Detection”**

Prevare sa kreditnim karticama se odvijaju često i rezultiraju ogromnim finansijskim gubicima. Kriminalci mogu da koriste neke tehnologije kao što su virus trojanac ili phishing metodu za krađu informacija i podataka sa tuđih kreditnih kartica. Prevare sa kreditnim karticama predstavljaju vrlo ozbiljan poslovni problem. Zbog toga kompanije ulažu sve više novca za razvoj novih ideja i načina koji će pomoći za otkrivanje i sprečavanje prevara. Stoga efikasno otkrivanje prevara je izuzetno važno u današnjem svetu, da bi što veći broj korisnika bio pošteđen malverzacije. Jedna od metoda je da u potpunosti iskoriste podatke o prethodnim transakcijama, uključujući i normalne transakcija radi uspostavljanja razlike između normalnog ponašanja i prevare.

Ovo se radi preko tehnika zasnovanih na mašinskom učenju. U ovom radu se za obuku (training data set) koriste dve vrste random forest-a kao karakteristični predstavnici radi uspostavljanja razlike između normalnih i nenormalnih transakcija. Pravimo poređenje dva random foresta koji predstavljaju osnovne klasifikatore i onda analiziramo njihove učinke. Podaci korišćeni u našim eksperimentima potiču iz kompanija za e-trgovinu u Kini.

Glavni cilj ovog rada je bilo upoređivanje određenih algoritama mašinskog učenja za otkrivanje transakcija koje su prevare. Random forest najbolje klasifikuje da li je nešto bila prevara ili ne. Ovo je utvrđeno korišćenjem različitih pokazatelja, kao što su opoziv, tačnost i preciznost. Za ovu vrstu problema jeste važno je imati opoziv sa velikom vrednošću. Izbor osobina i balansiranje skupa podataka pokazalo se izuzetno važnim u postizanju značajnih rezultata.

Konačni ishod skupa testova se odredio na osnovu maksimalne ponderisane šume. Ostvareni eksperimentalni rezultati su se manifestovali u povećanju efikasnosti predloženog modela čak i u neuravnoteženim slučajevima gde je bila prevara. Prevara je utvrđena sa tačnošću od 99.48% u svim slučajevima (Zhenchuan et al. 2018).

### **“Adversarial Support Vector Machine Learning”**

Kako upotreba kreditnih kartica postaje sve češća u svim poljima svakodnevnog života, prevara sa kreditnim karticama je postal mnogo razuzdanija. Da bi se poboljšala finansijska sigurnost transakcionih sistema na automatski i efikasan način, jedan od ključnih zadataka za finansijske institucije je stvaranje tačnog i efikasnog sistema za otkrivanje prevara sa kreditnim karticama.

Sporni napadi mogu dovesti do ozbiljnog lažnog predstavljanja stvarne distribucije podataka u prostoru karakteristika. Algoritmi učenja kojima nedostaje fleksibilnost rukovanja strukturnim promena u uzorcima, ne mogu da se nose sa napadima koji modifikuju podatke da bi se poboljšao skup uzorka. Ovaj rad predstavlja dva modela napada i kontradiktorni SVM model učenja nasuprot svakom modelu napada. U radu je pokazano da je kontradiktorni SVM model je mnogo otporniji na napade od standardnih SVM i jednoklasnih SVM modela.

Takođe pokazuje se da optimalne strategije učenja izvedene za suprotstavljanje preterano pesimističnim modelima napada mogu proizvesti nezadovoljavajuće rezultate kada su pravi napadi mnogo slabiji. S druge strane, modeli učenja izgrađeni na uzdržanim modelima napada izvode se doslednije kako se parametri napada razlikuju. Jedan od budućih pravaca ovog rada je dodavanje troškovno osetljivih metrika u modele učenja. Drugi pravac jeste širenje jedinstvenog modela učenja ansamblu u kome se svaka baza učenik bavi različitim skupovima napada (Zhou et al. 2012).

## **Literatura**

[1] Yiu, T. (2019). Understanding Random Forest. Preuzeto 18. juna 2021, sa <https://towardsdatascience.com/understanding-random-forest-58381e0602d2>

[2] Vidič, N. (2019). Primena mašinskog učenja u verifikaciji softvera. Master rad, Univerzitet u Beogradu, Matematički fakultet

[3] Amin, H. (2020). Credit Card Fraud Detection using Random Forest. Preuzeto 18. juna 2021, sa <https://www.kaggle.com/hassanamin/credit-card-fraud-detection-using-random-forest>

[4] McKinney, T. (2019, July). Fraud Detection in Python. Preuzeto 12. juna sa

<https://trenton3983.github.io/files/projects/2019-07-19_fraud_detection_python/2019-07-19_fraud_detection_python.html>

[5] North, P. (2021, June). Credit Card Fraud Detection with Python & Machine Learning. Preuzeto 12. juna sa

<https://data-flair.training/blogs/credit-card-fraud-detection-python-machine-learning/>

[6] Pratt, A. (2017). Python 3 programiranje i GUI. Beograd: Agencija Eho

[7] Sahin, Y., & Duman, E. (2011, June). Detecting credit card fraud by ANN and logistic regression. In 2011 International Symposium on Innovations in Intelligent Systems and Applications (pp. 315-319). IEEE

[8] Şahin, Y. G., Duman, E. (2011). Detecting credit card fraud by decision trees and support vector machines. In S. I. Ao, O. Castillo, C. Douglas, D. D. Feng, J. A. Lee (Eds.), Proceedings of the International MultiConference of Engineers and Computer Scientists 2011 (Volume 1) (pp. 442-447). Hong Kong: International Association of Engineers.

[9]Zhenchuan Li, Guanjun Liu, Shuo Wang, Shiyang Xuan, Changjun Jiang, "Random Forest for Credit Card Fraud Detection", SmartWorld Ubiquitous Intelligence & Computing Advanced & Trusted Computing Scalable Computing & Communications Cloud & Big Data Computing Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI) 2018 IEEE, pp. 1249-1254, 2018.

[10] Zhou, Y., Kantarcioglu, M., Thuraisingham, B., & Xi, B. (2012, August). Adversarial support vector machine learning. In Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining (pp. 1059-1067).