

**SVEUČILIŠTE U ZAGREBU  
FAKULTET ORGANIZACIJE I INFORMATIKE  
VARAŽDIN**

**Franco Kušek**

# **ANALIZA I DETEKCIJA SUMNJIVIH FINANCIJSKIH TRANSAKCIJA**

**PROJEKT**

## **VIŠEAGENTNI SUSTAVI**

**Varaždin, 2024.**

**SVEUČILIŠTE U ZAGREBU**  
**FAKULTET ORGANIZACIJE I INFORMATIKE**  
**V A R A Ž D I N**

**Franco Kušek**

**Matični broj: 35918/07–R**

**Studij: Informacijski i poslovni sustavi**

**ANALIZA I DETEKCIJA SUMNJIVIH FINANCIJSKIH TRANSAKCIJA**

**PROJEKT**

**Mentor:**

Markus Schatten, Marcel Maretić

**Varaždin, siječanj 2024.**

### **Izjava o izvornosti**

Izjavljujem da je ovaj projekt izvorni rezultat mojeg rada te da se u izradi istoga nisam koristio drugim izvorima osim onima koji su u njemu navedeni. Za izradu rada su korištene etički prikladne i prihvatljive metode i tehnike rada.

*Autor potvrdio prihvatanjem odredbi u sustavu FOI Radovi*

---

## Sažetak

U ovom projektu fokusiramo se na analizu i detekciju sumnjivih financijskih transakcija, koristeći podatke preuzete s Kaggle-a. Dataset: <https://www.kaggle.com/datasets/berkanoztas/synthetic-transaction-monitoring-dataset-aml> obuhvaća detaljne informacije o transakcijama koje mogu biti indicije na aktivnosti pranja novca. Podaci uključuju različite atribute transakcija, kao što su iznosi, datumi, identifikatori transakcija i druge relevantne informacije. Glavni cilj projekta je razviti metodologiju i alate za identifikaciju i analizu potencijalno sumnjivih transakcija. Ovaj projekt teži razumijevanju karakteristika koje razlikuju uobičajene transakcije od onih koje bi mogle biti povezane s pranjem novca. Za obradu i analizu ovih podataka koristit ćemo Python i njegove biblioteke poput Pandas, što nam omogućava efikasnu manipulaciju i analizu velikih skupova podataka. Za vizualizaciju podataka koristit ćemo biblioteke kao što su Matplotlib i Seaborn, koje omogućuju jasno i učinkovito prikazivanje kompleksnih podatkovnih setova.

**Ključne riječi:** riječ; riječ; ...riječ; Obuhvaća  $7 \pm 2$  ključna pojma koji su glavni predmet rasprave u radu.

# Sadržaj

<b>1. Uvod</b>	<b>1</b>
<b>2. Metode i tehnike rada</b>	<b>2</b>
<b>3. Razrada teme</b>	<b>3</b>
3.1. Opis implementacije aplikacije	3
3.2. Prikaz rada aplikacije	6
3.3. Kritički osvrt	9
<b>4. Zaključak</b>	<b>11</b>
<b>Popis literature</b>	<b>11</b>
<b>Popis slika</b>	<b>12</b>
<b>Popis isječaka koda</b>	<b>13</b>

# 1. Uvod

Financijske transakcije predstavljaju vitalni dio globalnog ekonomskog sustava. Međutim, one također mogu biti iskorištene za nezakonite aktivnosti kao što je pranje novca, što predstavlja značajnu prijetnju ekonomskoj stabilnosti i sigurnosti. Upravo zbog toga, analiza i detekcija sumnjivih financijskih transakcija postaju ključne u modernom financijskom sektoru. Cilj ovog projekta je upotrijebiti napredne tehnike obrade podataka za identifikaciju potencijalno sumnjivih aktivnosti unutar velikog volumena financijskih transakcija.

Glavni cilj ovog projekta je razviti robustnu metodologiju koja može detektirati i analizirati sumnjive financijske transakcije, s potencijalom da se primijeni u stvarnim financijskim sustavima. Kroz ovaj rad, težimo ne samo identificiranju rizičnih transakcija, već i boljem razumijevanju kako se financijski kriminal manifestira unutar transakcijskih mreža. Time će projekt doprinijeti jačanju sigurnosti i transparentnosti u financijskom sektoru.

## 2. Metode i tehnike rada

Projekt će koristiti detaljni dataset preuzet s Kaggle-a koji obuhvaća različite attribute transakcija poput iznosa, datuma, identifikatora transakcija i drugih relevantnih informacija.

Koristeći Python i njegove biblioteke kao što su Pandas, Matplotlib i Seaborn, projekt će obraditi i analizirati podatke kako bi identificirao uzorke i trendove karakteristične za sumnjive transakcije. Pandas će nam omogućiti efikasno upravljanje i analizu velikih skupova podataka, dok će Matplotlib i Seaborn biti korišteni za vizualizaciju rezultata, pružajući jasan i razumljiv uvid u naše nalaze. Te uz pomoc Flask servera ćemo omogućiti i pristup REST servisima.

## 3. Razrada teme

U ovome ću dijelu opisati implementaciju to jest opis pojedinih dijelova implementiranog sustava te prikazati kako aplikacija radi.

### 3.1. Opis implementacije aplikacije

Učitani skup podataka kroz kôd predstavlja transakcije koje su označene kao uobičajene ili sumnjive, potencijalno u kontekstu pranja novca. Kôd je dio analitičkog procesa u Pythonu, koristeći biblioteke kao što su pandas, numpy, matplotlib, seaborn, sqlite3 i Flask. Evo detaljnog opisa onoga što se događa u sklopu ovog procesa:

Učitavanje podataka: Podaci se učitavaju iz CSV i JSON datoteka u pandas DataFrame. CSV datoteka sadrži transakcije, a JSON datoteka dodatne generirane podatke.

---

```
1 # Loading the datasets
2 df_transactions = pd.read_csv('transakcije.csv')
3 df_generated = pd.read_json('franco.json')
```

---

Isječak koda 1: Učitavanje CSV i JSON datoteke

Uzorkovanje podataka: Iz podataka se uzorkuje određeni broj transakcija koje su označene kao pranje novca *laundering\_sample\_size* = 100 i koje nisu označene kao pranje novca *non\_laundering\_sample\_size* = 400.

---

```
1 # Određujemo veličinu uzorka za pranje novca i ne pranje novca
2 laundering_sample_size = 100 # Želimo 100 transakcija pranja novca
3 non_laundering_sample_size = 400 # i 400 transakcija koje nisu pranje novca

4 # Filtriranje transakcija koje su i nisu pranje novca
5 laundering_transactions = df_transactions[df_transactions['Is_laundering'] == 1]
6 non_laundering_transactions = df_transactions[df_transactions['Is_laundering'] == 0]
```

---

Isječak koda 2: Uzorkovanje podataka

Kombiniranje podataka: Kombiniraju se uzorci transakcija i generirani podaci u jedan DataFrame. DataFrame je zatim promiješan da bi se osigurala slučajnost.

---

```
1 # Spajamo uzorke u jedan DataFrame
2 df_sample = pd.concat([laundering_sample, non_laundering_sample]).sample(frac=1,
↳ random_state=42) # Promiješamo redove

3 # Sada kombiniramo uzorak s generiranim podacima
4 df_combined = pd.concat([df_sample.reset_index(drop=True), df_generated], axis=1)
```

---

Isječak koda 3: Kombiniranje podataka

Priprema podataka: Izvršava se one-hot encoding kategorijskih varijabli, odbacuju se ne-numerički stupci, rješavaju se nedostajući podaci, i sve vrijednosti se pretvaraju u numerički



format.

---

```
1 # One-hot encoding categorical variables
2 df_combined = pd.get_dummies(df_combined, columns=['Payment_type',
↳ 'Sender_bank_location', 'Receiver_bank_location'])

3 # Identifying non-numeric columns
4 non_numeric_columns = df_combined.select_dtypes(exclude=[np.number]).columns
5 print("Non-numeric columns:", non_numeric_columns)

6 # Drop non-numeric columns for correlation analysis
7 df_combined.drop(non_numeric_columns, axis=1, inplace=True)

8 # Handle missing values
9 df_combined.fillna(df_combined.mean(), inplace=True)

10 # Convert all columns to numeric
11 for col in df_combined.columns:
12     df_combined[col] = pd.to_numeric(df_combined[col], errors='coerce')
```

---

#### Isječak koda 4: Priprema podataka

Analiza korelacije: Izračunava se korelacija između svih numeričkih značajki i ciljne varijable *Is\_laundering* koja označava da li je transakcija sumnjiva. Normalizacija podataka:

---

```
1 # Correlation analysis
2 print(df_combined.corr()['Is_laundering'].sort_values())
```

---

#### Isječak koda 5: Analiza korelacije

Stupac Amount se normalizira za potrebe vizualizacije.

---

```
1 # Normalizing the 'Amount' column for visualization
2 df_combined['Amount'] = (df_combined['Amount'] - df_combined['Amount'].min()) /
↳ (df_combined['Amount'].max() - df_combined['Amount'].min())

3 grouped_data = df_combined.groupby('Is_laundering')['Amount'].sum().reset_index()
```

---

#### Isječak koda 6: Normalizacija podataka

Vizualizacija podataka: Izrađuje se parni graf (pairplot) za numeričke stupce te bar graf za usporedbu ukupnih iznosa transakcija koje su označene kao uobičajene i sumnjive.

Pairplot: Prikaže distribuciju pojedinačnih varijabli i odnose između njih. Svaki redak i stupac predstavlja jednu varijablu, a gdje se oni presijecaju, prikazan je graf odnosa između te dvije varijable. Dijagonala obično prikazuje distribuciju varijable s histogramom ili gustoćom.

---

```
1 grouped_data['Is_laundering'] = grouped_data['Is_laundering'].map({0: 'Uobičajene',  
↪ 1: 'Sumnjive'})  
2 numerical_columns = df_combined.select_dtypes(include=[np.number]).columns  
3 sns.pairplot(df_combined[numerical_columns])  
4 plt.show()
```

---

#### Isječak koda 7: Pairplot graf

Barplot: Prikazuje usporedbu ukupnih iznosa transakcija koje su uobičajene i one koje su sumnjive, omogućujući vizualnu usporedbu između dviju grupa.

---

```
1 # Vizualizacija usporedbe ukupnih iznosa za uobičajene i sumnjive transakcije  
2 sns.barplot(x='Is_laundering', y='Amount', data=grouped_data)  
3 plt.title('Usporedba ukupnih iznosa sumnjivih i uobičajenih transakcija')  
4 plt.ylabel('Ukupan iznos transakcija')  
5 plt.xlabel('Vrsta transakcije')  
6 plt.show()
```

---

#### Isječak koda 8: Barplot graf

Statistika: Izračunava se deskriptivna statistika za normalizirane iznose transakcija.

---

```
1 print(df_combined['Amount'].describe())
```

---

#### Isječak koda 9: Statistika podataka

Pohrana u bazu podataka: Podaci se pohranjuju u SQLite bazu podataka.

---

```
1 # Saving DataFrame to a SQLite database
2 conn = sqlite3.connect('transakcije.db')
3 df_combined.to_sql('transakcije', conn, if_exists='replace', index=False)
4 conn.close()
```

---

#### Isječak koda 10: Baza podataka

Flask aplikacija: Postavlja se Flask aplikacija s jednom rutom (/transakcije) koja omogućava dohvat transakcija iz baze podataka.

---

```
1 app = Flask(__name__)

2 def get_db_connection():
3     conn = sqlite3.connect('transakcije.db')
4     conn.row_factory = sqlite3.Row
5     return conn

6 @app.route('/transakcije', methods=['GET'])
7 def get_transakcije():
8     conn = sqlite3.connect('transakcije.db')
9     conn.row_factory = sqlite3.Row
10    cur = conn.cursor()
11    cur.execute("SELECT * FROM transakcije")
12    transakcije_rows = cur.fetchall()
13    conn.close()
14    return jsonify([dict(row) for row in transakcije_rows])

15 if __name__ == '__main__':
16     app.run(debug=True)
```

---

#### Isječak koda 11: Flask aplikacija

## 3.2. Prikaz rada aplikacije

Aplikaciju pokreće u komandnoj liniji komandom python (projekt\_franci.py). Aplikacije se nalazi u folderu sa dvije datoteke gdje je jedna u CSV formatu a druga u JSON formatu.

Ovo su dva izlaza iz programa koji koristite za analizu podataka o transakcijama.

Non-numeric columns:

Prvi dio ispisuje indeks ne-numeričkih stupaca koji se nalaze u vašem DataFrame-u. Ovi stupci sadrže tekstualne podatke ili kategorijske podatke koji su već pretvoreni u numerički oblik putem one-hot encodinga. Ne-numerički stupci = {'Payment\_type\_', 'Sender\_bank\_location\_', ...} One-hot encoding je metoda koja pretvara kategorijske varijable u oblik pogodan za strojno učenje, stvarajući novi stupac za svaku kategoriju i označavajući prisutnost (s vrijednošću 1) ili odsutnost (s vrijednošću 0) te kategorije. Korelacijski koeficijenti:

Drugi dio ispisuje korelacijske koeficijente između nekih od numeričkih varijabli i ciljne varijable *Is\_laundering*. Korelacijski koeficijent mjeri snagu i smjer linearne veze između dvije varijable. Vrijednosti se kreću od -1 do 1, gdje 1 označava savršenu pozitivnu linearnu ko-

```

C:\Users\Franz\Desktop\Projekt_pap>python projekt_franco.py
Non-numeric columns: Index(['Time', 'Date', 'Payment_currency', 'Received_currency',
    'Laundering_type', 'name', 'phone', 'email', 'address',
    'Payment_type_ACH', 'Payment_type_Cash Deposit',
    'Payment_type_Cash Withdrawal', 'Payment_type_Cheque',
    'Payment_type_Credit card', 'Payment_type_Cross-border',
    'Payment_type_Debit card', 'Sender_bank_location_Albania',
    'Sender_bank_location_Austria', 'Sender_bank_location_France',
    'Sender_bank_location_Germany', 'Sender_bank_location_India',
    'Sender_bank_location_Italy', 'Sender_bank_location_Japan',
    'Sender_bank_location_Mexico', 'Sender_bank_location_Morocco',
    'Sender_bank_location_Netherlands', 'Sender_bank_location_Nigeria',
    'Sender_bank_location_Pakistan', 'Sender_bank_location_Switzerland',
    'Sender_bank_location_Turkey', 'Sender_bank_location_UK',
    'Sender_bank_location_USA', 'Receiver_bank_location_Albania',
    'Receiver_bank_location_Austria', 'Receiver_bank_location_France',
    'Receiver_bank_location_Germany', 'Receiver_bank_location_India',
    'Receiver_bank_location_Italy', 'Receiver_bank_location_Japan',
    'Receiver_bank_location_Mexico', 'Receiver_bank_location_Morocco',
    'Receiver_bank_location_Netherlands', 'Receiver_bank_location_Nigeria',
    'Receiver_bank_location_Pakistan', 'Receiver_bank_location_Turkey',
    'Receiver_bank_location_UAE', 'Receiver_bank_location_UK',
    'Receiver_bank_location_USA'],
    dtype='object')
Receiver_account    0.033788
Sender_account      0.041129
Amount              0.082764
Is_laundering       1.000000
Name: Is_laundering, dtype: float64

```

Slika 1: Prvi ispis programa

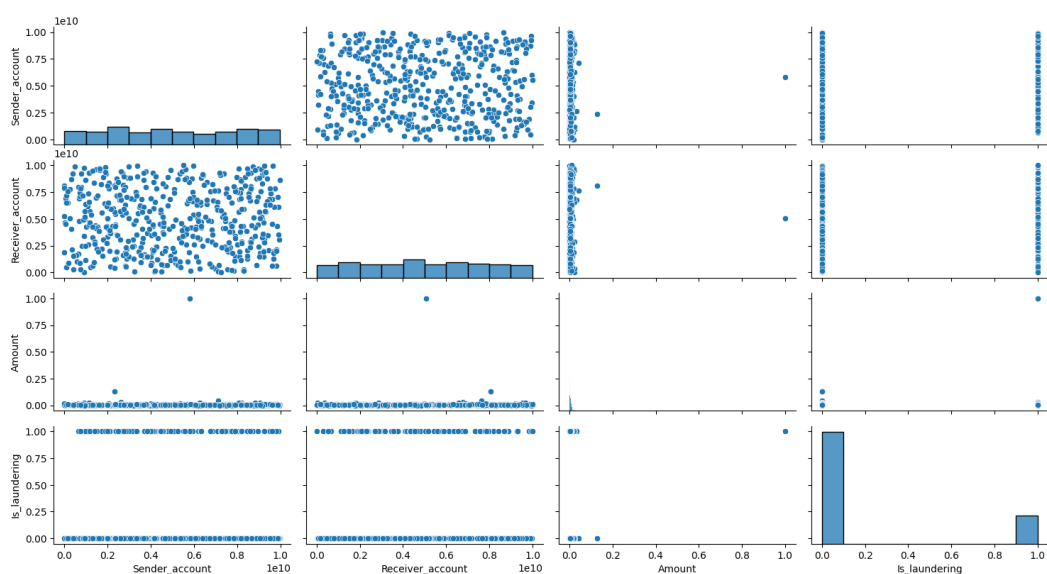
relaciju, -1 savršenu negativnu linearnu korelaciju, a 0 označava da nema linearnog odnosa. Receiver\_account : Ima vrlo slabu pozitivnu korelaciju s varijablom Is\_laundering (0.033788). Sender\_account: Također ima vrlo slabu pozitivnu korelaciju s Is\_laundering (0.041129). Amount: Ima nešto jaču, ali i dalje vrlo slabu pozitivnu korelaciju s Is\_laundering (0.082764). Is\_laundering: Ima savršenu korelaciju sa samim sobom (1.000000), što je očekivano jer je to ista varijabla. Korelacijski koeficijenti ukazuju na to da nijedna od ovih varijabli nema jaku korelaciju s varijablom Is\_laundering, što bi moglo značiti da model za detekciju pranja novca ne može se osloniti samo na te varijable i da bi bilo potrebno istražiti druge značajke ili koristiti složenije modele za predviđanje. Zatim se otvara prva slika grafova. Prvi redak i prvi stupac prikazuju varijablu Sender\_account, gdje je histogram prikazan na dijagonali. Histogram pokazuje distribuciju vrednosti početnih računa pošiljatelja. Drugi redak i drugi stupac prikazuju varijablu Receiver\_account s histogramom na dijagonali, prikazujući distribuciju vrijednosti računa primatelja. Treći redak i treći stupac prikazuju varijablu Amount (iznos), gdje je histogram također na dijagonali, prikazujući distribuciju iznosa transakcija. Četvrti redak i stupac prikazuju varijablu Is\_laundering s dvije vrijednosti (0 ili 1), prikazane kao dvije barske kolone na histogramu koji prikazuje broj transakcija koje su i nisu označene kao pranje novca. Scatterplotovi izvan dijagonale:

Scatterplot između Sender\_account i Receiver\_account pokazuje točke raspršene na grafu, bez jasne korelacije između broja računa pošiljatelja i primatelja. Scatterplot između Sender\_account i Amount prikazuje iznos transakcija početnih računa pošiljatelja. Većina točaka je koncentrirana blizu dna, što sugerira da su većina transakcija male vrijednosti. Scatterplot između Receiver\_account i Amount ima sličnu distribuciju točaka kao i scatterplot između Sender\_account i Amount. Scatterplotovi koji uključuju Is\_laundering pokazuju jasnu distinkciju

između transakcija koje su označene kao pranje novca (vrijednost 1) i onih koje nisu (vrijednost 0). Većina transakcija koje nisu pranje novca su koncentrirane na jednoj vrijednosti, dok one koje jesu, izgledaju kao odvojeni outliers. Boxplotovi na dijagonalama:

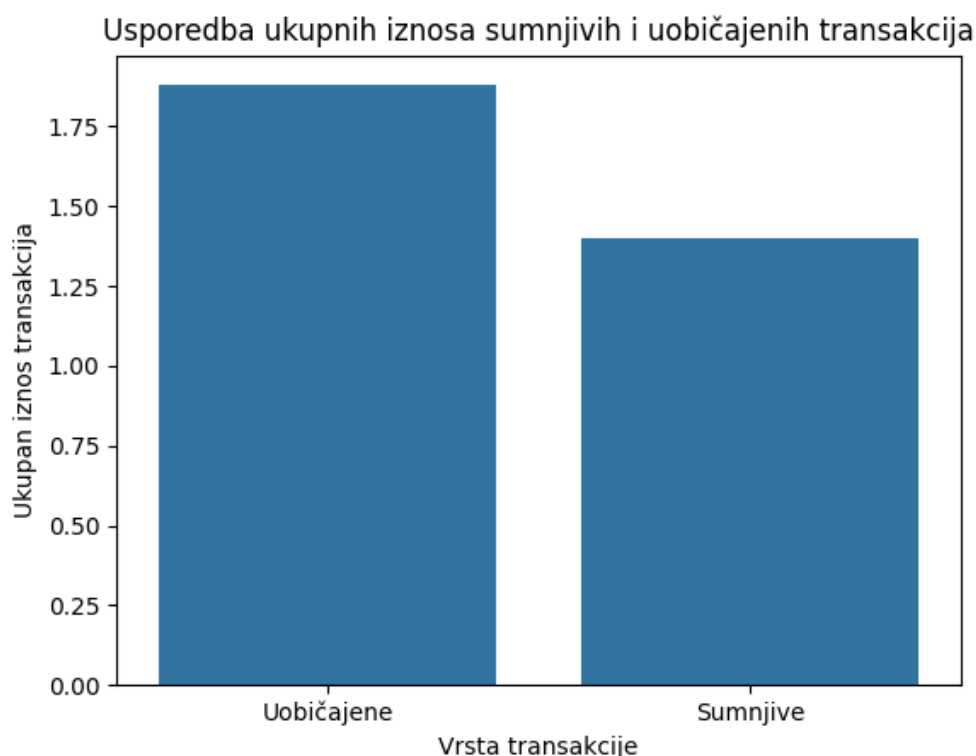
Boxplotovi su prikazani na mjestima gdje bi trebali biti scatterplotovi između istih varijabli. Oni omogućuju vizualnu ocjenu raspodjele (distribucije) transakcija po računima i iznosima, kao i identifikaciju outlier-a (vrijednosti koje odstupaju). Zaključci:

Postoji velik broj transakcija s malim iznosom, što je vidljivo iz koncentracije točaka blizu dna scatterplotova za Amount. Postoji nekoliko outlier-a kod varijable Amount, što može ukazivati na neobično velike transakcije. Postoji jasna separacija između običnih transakcija i onih koje su označene kao pranje novca u odnosu na Amount, što se vidi u scatterplotu Amount vs Is\_laundering.



Slika 2: Scatterplot

Nakon toga slijedi slika boxplota koji uspoređuje kupan iznos između sumnjivih i nesumnjivih transakcija



Slika 3: Sumnjive i nesumnjive transakcije

Zatim se otvara flask server na kojem će se također izvesti sve te operacije te se serveru pristupa na stranici "http://127.0.0.1:5000/transakcije".

```
Name: Is_laundering, dtype: float64
count    500.000000
mean      0.006562
std       0.045098
min       0.000000
25%       0.001243
50%       0.003217
75%       0.005808
max       1.000000
Name: Amount, dtype: float64
* Serving Flask app 'projekt_franco'
* Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on http://127.0.0.1:5000
Press CTRL+C to quit
* Restarting with stat
```

Slika 4: Otvaranje flask servera

### 3.3. Kritički osvrt

Praktična izvedivost:

Prikupljanje i obrada podataka:

Prikupljanje podataka iz različitih izvora (CSV i JSON datoteke) i njihovo kombiniranje

je praktično izvedivo i čest je pristup u analizi podataka. Korištenje pandas biblioteke za obradu podataka je standardna praksa i pokazuje visoku razinu praktičnosti. Uzorkovanje podataka za stvaranje balansiranog seta između transakcija koje su i nisu pranje novca je dobar pristup, no postavlja se pitanje da li ovo uzorkovanje može uzrokovati gubitak važnih informacija. Analiza podataka:

One-hot encoding i normalizacija podataka su dobre prakse u pripremi podataka za modeliranje, ali je važno provjeriti da li one-hot encoding generira previše stupaca što može dovesti do problema visoke dimenzionalnosti. Analiza korelacije je koristan alat za početno razumijevanje strukture podataka, ali slabe korelacije između značajki i ciljne varijable mogu upućivati na to da linearni modeli neće biti dovoljno dobri. Vizualizacija podataka:

Vizualizacija s seaborn bibliotekom je moćan način za dobivanje uvida u podatke, ali u ovom slučaju, pairplotovi s velikim brojem točaka mogu biti teški za interpretaciju i mogli bi koristiti dodatne metode vizualizacije za bolje razumijevanje. Primjena modela:

Projekt ne opisuje primjenu niti evaluaciju modela strojnog učenja, što je ključni korak u predviđanju pranja novca. Primjena:

Detekcija pranja novca:

Projekt ima potencijal za primjenu u financijskim institucijama, gdje bi se model strojnog učenja mogao koristiti za automatsko označavanje sumnjivih transakcija. Implementacija modela u stvarnom vremenu zahtijevala bi robustan sustav s visokom točnošću i niskom stopom lažnih pozitiva. Skalabilnost i performanse:

Učinkovitost obrade podataka i modeliranja bi se trebala testirati na velikim setovima podataka kako bi se osigurala skalabilnost. Flask aplikacija može biti korisna za prototipiranje, ali za proizvodnu upotrebu može biti potrebna robustnija arhitektura (npr., upotreba Flask-a u kombinaciji s WSGI serverom). Sigurnost i privatnost:

Postupanje s financijskim transakcijama zahtijeva visoku razinu sigurnosti i usklađenost s regulatornim standardima poput GDPR-a i PCI DSS-a. Kritično je osigurati anonimizaciju i sigurnost osobnih podataka klijenata.

## 4. Zaključak

Projekt analize transakcija za detekciju pranja novca predstavlja važan korak prema razumijevanju i identifikaciji potencijalno sumnjivih financijskih aktivnosti. Kroz korištenje modernih alata za obradu i vizualizaciju podataka, kao što su pandas, seaborn i matplotlib, projekt uspješno postavlja temelje za pripremu, istraživanje i analizu transakcijskih podataka. Praktična izvedba učitavanja, obrade i vizualizacije podataka je ispravno provedena, omogućujući temeljitu inicijalnu analizu koja može biti korisna za izradu prediktivnih modela.

Međutim, za krajnji cilj detekcije pranja novca, projekat bi se morao proširiti kako bi uključivao razvoj i evaluaciju modela strojnog učenja koji mogu efikasno razlikovati legitimne transakcije od sumnjivih. Također, treba razmotriti uvođenje naprednijih metoda analize podataka, poput algoritama za nenadgledano učenje koji mogu otkriti skrivene uzorke u podacima.

Daljnji razvoj projekta bi također trebao uključivati procjenu performansi modela na većim i raznovrsnijim skupovima podataka kako bi se osigurala skalabilnost i robustnost sistema. Također, uvođenje komponenti sigurnosti i privatnosti je neophodno za zaštitu osjetljivih podataka i usklađenost s regulatornim zahtjevima.

U kontekstu realnog svijeta, projekt bi trebao biti optimiziran za brzu obradu velikog broja transakcija u realnom vremenu, uz održavanje niske stope lažnih pozitiva i lažnih negativa, što je ključno za praktičnu primenu u financijskim institucijama.

Za konačnu implementaciju, ključno je razviti jasnu strategiju uvođenja modela u operativne tokove rada, kako bi se osiguralo da detekcija pranja novca doprinosi efikasnosti i regulatornoj usklađenosti, a istovremeno minimizira poremećaje u poslovanju. Uz kontinuirani razvoj i poboljšanje, ovaj projekt ima potencijal postati vrijedan alat u borbi protiv financijskog kriminala.



# Popis slika

1.	Prvi ispis programa . . . . .	7
2.	Scatterplot . . . . .	8
3.	Sumnjive i nesumnjive transakcije . . . . .	9
4.	Otvaranje flask servera . . . . .	9

# Popis isječaka koda

1.	Učitavanje CSV i JSON datoteke . . . . .	3
2.	Uzorkovanje podataka . . . . .	3
3.	Kombiniranje podataka . . . . .	3
4.	Priprema podataka . . . . .	4
5.	Analiza korelacije . . . . .	4
6.	Normalizacija podataka . . . . .	4
7.	Pairplot graf . . . . .	5
8.	Barplot graf . . . . .	5
9.	Statistika podataka . . . . .	5
10.	Baza podataka . . . . .	6
11.	Flask aplikacija . . . . .	6

# Bibliografija

- [1] Stuart Russell and Peter Norvig. *Artificial Intelligence: A Modern Approach*. Pearson, 3rd edition, 2009.
- [2] Gary Linscott. *Python Chess Programming*. Online: <https://python-chess.readthedocs.io/en/latest/>, Accessed: 2024-01-14.
- [3] George T. Heineman, Gary Pollice, and Stanley Selkow. *Algorithms in a Nutshell*. O'Reilly Media, 2nd edition, 2015.
- [4] Jeremy Silman. *The Complete Book of Chess Strategy: Grandmaster Techniques from A to Z*. Siles Press, 1998.