



Univerzitet u Sarajevu  
Elektrotehnički fakultet Sarajevo  
Odsjek za računarstvo i informatiku



# Vještačka inteligencija

## Credit card fraud detection

Amila Kukić 19065  
Sara Kardaš 19180

Sarajevo, 2025

## Sadržaj

Izbor teme i opis problema.....	3
Osnovni pojmovi.....	3
Postojeći dataset-ovi i izvori.....	4
Pregled stanja u oblasti.....	4
Analiza literature i aktuelnih istraživanja.....	4
Primjeri iz literature.....	5
Postignuti rezultati.....	5
Pravci za poboljšanje.....	5
Zaključak.....	6
Izbor, analiza i pretprocesiranje dataset-a.....	6
Osnovne informacije o datasetu.....	7
Pretprocesiranje podataka.....	8
Razlog primjene pretprocesiranja.....	10
Identifikovani rizici.....	10
Odabir, formiranje, treniranje i testiranje modela.....	11
Korištene tehnologije.....	12
Priprema podataka.....	12
Treniranje modela.....	12
Nebalansirani (nauravnoteženi skup podataka).....	13
Balansirani trening skup - SMOTE metoda.....	17
Balansirani trening skup - Undersampling.....	20
Cjelokupni osvrt na problem i dobijeno rješenje.....	24
Poređenje s rezultatima iz prethodne faze.....	25
Šta se moglo bolje?.....	25
Zaključak.....	26

# Izbor teme i opis problema

Finansijske prevare putem platnih kartica predstavljaju ozbiljan i sveprisutni problem u digitalnom dobu. Sa sve većim brojem online transakcija i sveprisutnim korištenjem platnih kartica, dolazi do porasta broja pokušaja neovlaštenih transakcija. Ovakve prevare mogu uzrokovati ozbiljne finansijske gubitke, kako za korisnike kartica, tako i za banke i finansijske institucije. Rano otkrivanje prevara i zaštita korisnika od neautorizovanih transakcija je od ključnog značaja kako za sigurnost korisnika, tako i za očuvanje reputacije finansijskih institucija.

## Problem koji se rješava:

Detekcija sumnjivih transakcija u realnom vremenu, koristeći binarne klasifikacione modele mašinskog učenja koji automatski klasifikuju svaku transakciju kao legitimnu (non-fraud) ili prevarantsku (fraud), testirajući modele kako na nebalansiranom tako i na balansiranom skupu podataka.

## Koristi ovog rješenja:

- Otkrivanje sumnjivih aktivnosti na vrijeme.
- Smanjenje finansijskih gubitaka i štete po korisnike i institucije.
- Automatizacija procesa detekcije, smanjenje potrebe za ručnom provjerom.
- Unapređenje sigurnosti i povjerenja korisnika u digitalne sisteme plaćanja.

## Osnovni pojmovi

- **Klasifikacija:** Proces svrstavanja podataka u unaprijed definisane kategorije.
- **Neuravnotežen (nebalansiran) skup podataka:** kada broj instanci po klasama značajno varira.
- **Precision, Recall, F1-score:** Metričke vrijednosti za procjenu performansi modela.
- **Oversampling:** Tehnika povećanja broja instanci manjinske klase kreiranjem sintetičkih primjera.
- **Undersampling:** Tehnika smanjenja broja instanci većinske klase radi postizanja balansiranog skupa.
- **SMOTE:** Popularna metoda oversamplinga koja generiše sintetičke podatke za manjinsku klasu.

## Postojeći dataset-ovi i izvori

- **Credit Card Fraud Detection** (Kaggle): skup anonimnih podataka iz evropske banke, sa 284.807 transakcija - dataset koji smo mi koristile. (<https://www.kaggle.com/datasets/mlg-ulb/creditcardfraud/data>)
- **IEEE-CIS Fraud Detection** (Kaggle): realni podaci o online transakcijama, korišteni u industrijskim rješenjima (<https://www.kaggle.com/competitions/ieee-fraud-detection/data>)
- **Credit Card Fraud Detection Dataset 2023** (Kaggle): Ažurirani skup iz 2023. sa dodatim statističkim karakteristikama transakcija- balansirani dataset. (<https://www.kaggle.com/datasets/nelgiriyeewithana/credit-card-fraud-detection-dataset-2023> ) Ovo je dataset koji nam je bio prvobitno dodijeljen. Međutim analizom istog, shvatile smo da je ovo dataset koji je već unaprijed balansiram, pri čemu naš pristup ovom projektu ne bi imao smisla. Zbog toga je odlučeno da se koristi prvi navedeni dataset.
- **Fraud Detection Credit Card** (Kaggle): Skup sličan originalnom, koristi generisane podatke na osnovu statističkog modela. (<https://www.kaggle.com/datasets/yashpaloswal/fraud-detection-credit-card>)
- **Credit Card Fraud Detection by Shayannaveed** (Kaggle): Fokusiran na balansirane verzije izvornog skupa i primjenu osnovnih klasifikatora. (<https://www.kaggle.com/datasets/shayannaveed/credit-card-fraud-detection> )
- **Credit Card Fraud by Dhanush Narayanan** (Kaggle): Skup s fokusom na jednostavnije varijante podataka za edukativne svrhe. (<https://www.kaggle.com/datasets/dhanushnarayananr/credit-card-fraud>)

## Pregled stanja u oblasti

Detekcija prevara je jedno od najaktuelnijih i najzahtjevnijih područja primjene mašinskog učenja, posebno zbog karakterističnog problema neuravnoteženih skupova podataka, gdje pozitivni slučajevi (prevare) čine izrazito mali procenat svih transakcija.

## Analiza literature i aktuelnih istraživanja

Najčešće korištene metode su:

- Random Forest
- Logistic Regression
- XGBoost
- Neural Networks (MLP, Keras)
- Support Vector Machine (SVM)

U većini radova prisutna je potreba za balansiranjem podataka, s obzirom da fraud transakcije čine ispod 0.2% svih slučajeva. Najčešće korištene metode za balansiranje:

- **SMOTE** (Synthetic Minority Oversampling Technique)
- **ADASYN** (Adaptive Synthetic Sampling)
- **Undersampling**

### Primjeri iz literature

- *"Credit Card Fraud Detection using SMOTE and Ensemble Methods"* (Elsevier, 2020, [https://ijoer.com/assets/articles\\_menuscripts/file/IJOER-AUG-2021-3.pdf](https://ijoer.com/assets/articles_menuscripts/file/IJOER-AUG-2021-3.pdf)) pokazao je da kombinacija Random Forest-a i SMOTE-a postiže F1-score iznad 0.85.
- *"Fraud Detection Handbook"* (<https://fraud-detection-handbook.github.io/fraud-detection-handbook>) - jedan od najdetaljnijih otvorenih izvora, koji kroz praktične primjere objašnjava upotrebu klasifikacionih metoda, balansiranja i anomaly detection pristupa na problemima detekcije prevara na platnim karticama

### Postignuti rezultati

Većina istraživanja ukazuje da:

- **Random Forest** i **XGBoost** uz SMOTE ili ADASYN daju najbolje rezultate za F1-score i Recall.
- **Logistic Regression** i **SVM** bilježe pad performansi bez balansiranja podataka.
- **Neuronske mreže** postižu stabilne performanse, ali zahtijevaju optimizaciju hiperparametara.

### Pravci za poboljšanje

Umjesto standardnih pristupa, u literaturi se sve više preporučuju napredne metode koje mogu dodatno poboljšati detekciju rijetkih prevarantskih transakcija:

- **Kombinacija više modela (ensemble stacking i boosting)**: Umjesto oslanjanja na jedan model, preporučuje se kombinacija više njih (npr. XGBoost, Random Forest i Logistic Regression zajedno), čime se koristi snaga svakog modela i postižu stabilnije i tačnije predikcije.
- **Primjena naprednih metoda za balansiranje podataka**: Osim klasičnog SMOTE-a, koriste se i metode kao **Borderline-SMOTE** (koji stvara nove podatke na granici između klasa) i **ADASYN** (koji kreira više podataka tamo gdje je detekcija teža), čime se poboljšava prepoznavanje sumnjivih transakcija koje se nalaze blizu legitimnih.

- **Korištenje anomaly detection tehnika:** Metode poput **Isolation Forest** i **Autoencoders** mogu detektovati neobične obrasce ponašanja koji odstupaju od uobičajenog, što je korisno za slučajeve gdje klasične metode klasifikacije nisu dovoljne.
- **Primjena modela osjetljivih na troškove grešaka (cost-sensitive learning):** U ovim pristupima se greške pri klasifikaciji prevare tretiraju ozbiljnije od grešaka kod legitimnih transakcija, čime se smanjuje vjerovatnoća da prevara ostane neotkrivena, čak i po cijenu povećanja lažnih alarma.

Ove metode obećavaju dodatno unapređenje performansi modela, posebno u detekciji rijetkih, ali važnih prevara.

## Zaključak

Na osnovu sprovedenog istraživanja i pregleda stanja u oblasti otkrivanja prevara kreditnim karticama, može se zaključiti da je riječ o izuzetno izazovnom problemu zbog velike neuravnoteženosti podataka. Tradicionalne metode klasifikacije bez prethodnog balansiranja skupa daju slabe rezultate za prevarantske transakcije. Najbolje performanse, prema dostupnoj literaturi i vlastitom opažanju, postižu ensemble metode kao što su Random Forest i XGBoost, posebno kada se koriste uz SMOTE ili ADASYN za balansiranje podataka.

Naprednije tehnike poput ensemble stacking-a, anomaly detection modela (Isolation Forest, Autoencoders) i cost-sensitive pristupa predstavljaju perspektivne smjerove za dalje unapređenje performansi. Njihova primjena može dodatno poboljšati Recall i smanjiti broj prevara koje ostaju neotkrivene.

Zaključno, detekcija prevara putem mašinskog učenja zahtijeva pažljivo odabrane algoritme, prilagođene tehnike balansiranja i pravilno definisane metrike evaluacije. Dalja istraživanja treba usmjeriti ka integraciji više komplementarnih metoda i kombinovanih modela kako bi se povećala sigurnost platnih sistema u digitalnom okruženju.

## Izbor, analiza i preprocesiranje dataset-a

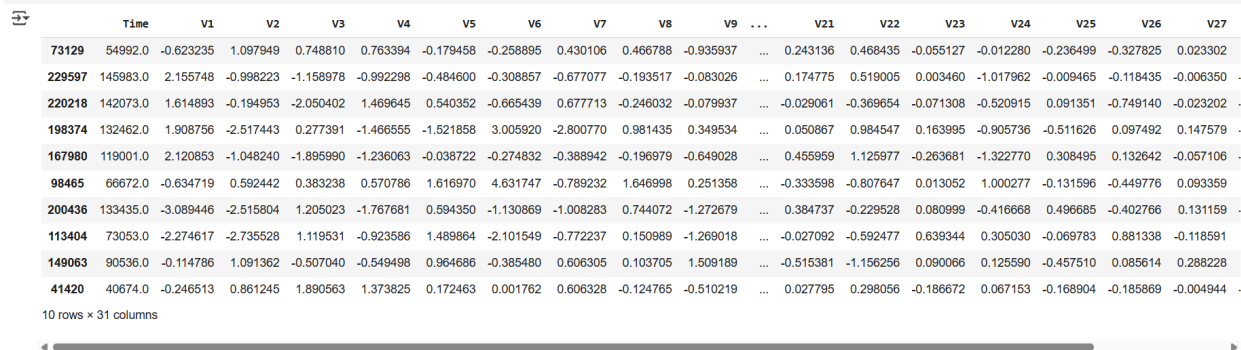
**Izabrani dataset:** Za rješavanje ovog problema korišten je "**Credit Card Fraud Detection**" dataset dostupan na Kaggle platformi:  
<https://www.kaggle.com/datasets/mlg-ulb/creditcardfraud/data>

## Osnovne informacije o datasetu

- **Izvor:** Kaggle (originalno podaci iz evropske banke)
- **Format:** CSV datoteka
- **Način preuzimanja:** Direktno preuzimanje sa Kaggle stranice nakon registracije

Učitavanje dataseta i prikaz 10 random vrijednosti

```
[ ] data = pd.read_csv("creditcard.csv")
data.sample(10, random_state=123)
```



	Time	V1	V2	V3	V4	V5	V6	V7	V8	V9	...	V21	V22	V23	V24	V25	V26	V27
73129	54992.0	-0.623235	1.097949	0.748810	0.763394	-0.179458	-0.258895	0.430106	0.466788	-0.935937	...	0.243136	0.468435	-0.055127	-0.012280	-0.236499	-0.327825	0.023302
229597	145983.0	2.155748	-0.998223	-1.158978	-0.992298	-0.484600	-0.308857	-0.677077	-0.193517	-0.083026	...	0.174775	0.519005	0.003460	-1.017962	-0.009465	-0.118435	-0.006350
220218	142073.0	1.614893	-0.194953	-2.050402	1.469645	0.540352	-0.665439	0.677713	-0.246032	-0.079937	...	-0.029061	-0.369654	-0.071308	-0.520915	0.091351	-0.749140	-0.023202
198374	132462.0	1.908756	-2.517443	0.277391	-1.466555	-1.521858	3.005920	-2.800770	0.981435	0.349534	...	0.050867	0.984547	0.163995	-0.905736	-0.511626	0.097492	0.147579
167990	119001.0	2.120853	-1.048240	-1.895990	-1.236063	-0.038722	-0.274832	-0.389942	-0.196979	-0.649028	...	0.455959	1.125977	-0.263681	-1.322770	0.308495	0.132642	-0.057106
98465	66672.0	-0.634719	0.592442	0.383238	0.570786	1.616970	4.631747	-0.789232	1.646998	0.251358	...	-0.333598	-0.807647	0.013052	1.000277	-0.131596	-0.449776	0.093359
200436	133435.0	-3.089446	-2.515804	1.205023	-1.767681	0.594350	-1.130869	-1.008283	0.744072	-1.272679	...	0.384737	-0.229528	0.080999	-0.416668	0.496685	-0.402766	0.131159
113404	73053.0	-2.274617	-2.735528	1.119531	-0.923586	1.489864	-2.101549	-0.772237	0.150989	-1.269018	...	-0.027092	-0.592477	0.639344	0.305030	-0.069783	0.881338	-0.118591
149063	90536.0	-0.114786	1.091362	-0.507040	-0.549498	0.964686	-0.385480	0.606305	0.103705	1.509189	...	-0.515381	-1.156256	0.090066	0.125590	-0.457510	0.085614	0.288228
41420	40674.0	-0.246513	0.861245	1.890563	1.373825	0.172463	0.001762	0.606328	-0.124765	-0.510219	...	0.027795	0.298056	-0.186672	0.067153	-0.168904	-0.185869	-0.004944

10 rows x 31 columns

- **Broj instanci (redova):** 284.807
- **Broj atributa (kolona):** 31 (30 atributa + ciljna varijabla Class)
- **Broj klasa:** 2 (0 - legitimna transakcija, 1 - prevara)
- **Broj instanci po klasama:**
  - Klasa 0 (legitimno): 284.315 (99.8%)
  - Klasa 1 (prevara): 492 (0.17%)
- **Veličina datoteke:** 143.84 MB

Osnovne informacije o veličini skupa (broj redova i kolona), ukupna količina memorije koju dataset zauzima i iskorištena memorija

```
[4] file_path = 'creditcard.csv'
file_size = os.path.getsize(file_path) / (1024 * 1024)
print(f"Veličina datoteke: {file_size:.2f} MB")
print(f"Broj redova i kolona u dataset-u: {data.shape}")
print(f"Iskorištena memorija: {data.memory_usage(deep=True).sum() / 1024**2:.2f} MB")
```

Veličina datoteke: 143.84 MB  
Broj redova i kolona u dataset-u: (284807, 31)  
Iskorištena memorija: 67.36 MB

Prikaz broja i postotnog udjela svake klase u ciljnoj koloni Class

```
[5] class_distribution = data['Class'].value_counts()
```

```
[6] print(class_distribution)
```

Class  
0 284315  
1 492  
Name: count, dtype: int64

- **Podjela podataka:**
  - Trening skup: 80%
  - Test skup: 20%

Priprema podataka

Skaliranje svih numeričkih kolona osim id i class

```
[11] numeric_cols = [col for col in data.columns if col not in ['id', 'class']]
    scaler = MinMaxScaler()
    data[numeric_cols] = scaler.fit_transform(data[numeric_cols])
```

Podjela podataka na trening i test skup

trening skup -> 80%

test skup -> 20%

```
[12] X = data[numeric_cols]
    y = data['class']
    X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, stratify=y, random_state=42)
```

Broj instanci u trening i test skupu

```
print(f"Broj instanci u trening skupu: {X_train.shape[0]}")
print(f"Broj instanci u test skupu: {X_test.shape[0]}")
```

```
Broj instanci u trening skupu: 226988
Broj instanci u test skupu: 56746
```

## Pretprocesiranje podataka

Zbog specifičnosti dataseta izvršene su sljedeće aktivnosti:

- **Provjera nedostajućih vrijednosti:** Nije bilo NaN vrijednosti u skupu.

Provjera da li u skupu podataka postoje nedostajuće vrijednosti, duplikati te koji su sve tipovi podataka prisutni. Ove informacije su važne za eventualno čišćenje skupa i pripremu podataka za dalju obradu.

```
print(f"Broj nedostajućih vrijednosti: \n{data.isnull().sum().sum()}")
print(f"Broj duplikata: {data.duplicated().sum()}")
print(f"Tipovi podataka: \n{data.dtypes.value_counts()}")

print(f"\nRaspodjela klasa")
class_dist = data['class'].value_counts()
print(f"Non-Fraud (0): {class_dist[0]:,} ({class_dist[0]/len(data)*100:.2f}%)")
print(f"Fraud (1): {class_dist[1]:,} ({class_dist[1]/len(data)*100:.2f}%)")
```

```
Broj nedostajućih vrijednosti:
0
Broj duplikata: 1081
Tipovi podataka:
float64    30
int64       1
Name: count, dtype: int64

Raspodjela klasa
Non-Fraud (0): 284,315 (99.83%)
Fraud (1): 492 (0.17%)
```

- **Detekcija i uklanjanje duplikata:** Uočen mali broj duplikata koji su uklonjeni.



## ▼ Čišćenje podataka

Prikaz kolona i redova koji posjeduju NaN vrijednosti

```
[9] columns_with_nan = (data.isna().sum() > 0).sum()
    print(f"Broj kolona koje sadrže NaN vrijednosti: {columns_with_nan}")
```

Broj kolona koje sadrže NaN vrijednosti: 0

Dataset je sadržavao 1081 duplikat koji su uklonjeni radi očuvanja kvalitete podataka i tačnosti modela.

```
[10] before = len(data)
     data = data.drop_duplicates()
     after = len(data)
     print(f"Obrisan je: {before - after} red sa NaN vrijednostima.")
```

Obrisan je: 1081 red sa NaN vrijednostima.

- **Skaliranje numeričkih atributa:** Pošto podaci sadrže numeričke vrijednosti različitih raspona, korišten je **MinMaxScaler** radi normalizacije.

Priprema podataka

Skaliranje svih numeričkih kolona osim id i class

```
[11] numeric_cols = [col for col in data.columns if col not in ['id', 'class']]
     scaler = MinMaxScaler()
     data[numeric_cols] = scaler.fit_transform(data[numeric_cols])
```

- **Balansiranje skupa:** Korištene metode:
  - **SMOTE:** za kreiranje dodatnih sintetičkih primjera manjinske klase.

## ▼ 2. Balansiran samo trening skup - oversampling

Korištena je SMOTE tehnika za balansiranje samo trening skupa, gdje su sintetički primjera manjinske klase, klase 1 (fraud).

Ova metoda pomaže u smanjenju problema neravnoteže u dataset-u.

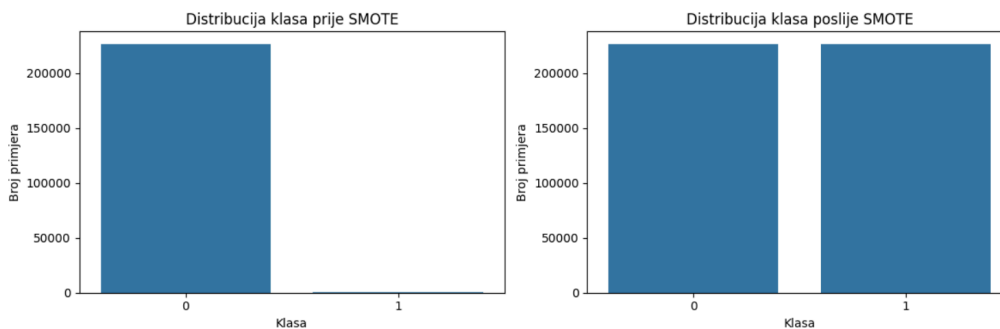
Test skup ostaje nepromijenjen kako bi se postigla realna i nepristrana evaluacija modela.

```
[ ] X_train2 = X_train
    X_test2 = X_test
    y_train2 = y_train
    y_test2 = y_test
```

Primjena SMOTE tehnike na trening skupu podataka

```
[ ] from imblearn.over_sampling import SMOTE
    sm = SMOTE(random_state=42)
    X_train2_res, y_train2_res = sm.fit_resample(X_train2, y_train2)
```

13



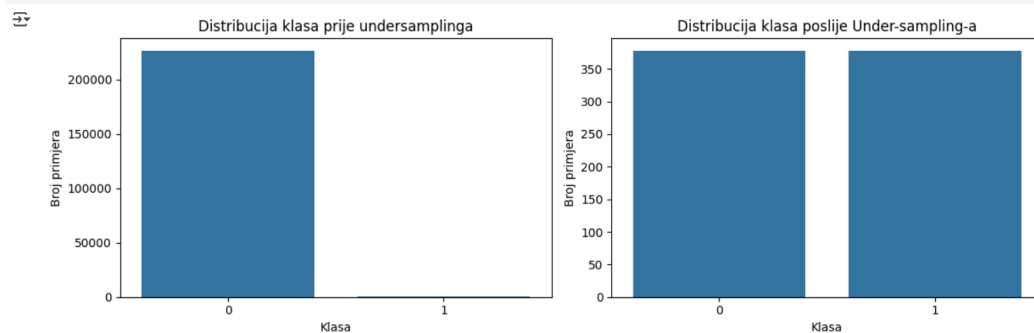
- **Undersampling:** nasumično smanjivanje broja instanci većinske klase.

### 3. Balansiran trening skup podataka - undersampling

Kod ovog pristupa prilagođen je trening skup, tako što su se nasumično uklonili primjerci većinske klase, klase 0 (non-fraud), kako bi se postigao balans između obje klase.

```
[ ] X_train3 = X_train
    X_test3 = X_test
    y_train3 = y_train
    y_test3 = y_test

[ ] from imblearn.under_sampling import RandomUnderSampler
    rus = RandomUnderSampler(random_state=42)
    X_train3_res, y_train3_res = rus.fit_resample(X_train3, y_train3)
```



## Razlog primjene pretprocesiranja

- **Skaliranje** je neophodno jer većina algoritama osjetljivo reaguje na različite raspona vrijednosti.
- **Balansiranje podataka** značajno poboljšava Recall i F1-score za rijetku klasu.
- **Detekcija i uklanjanje duplikata** spriječava negativan uticaj višestrukih identičnih zapisa na model.

Time su podaci pripremljeni za fazu treniranja i testiranja različitih klasifikacionih modela.

## Identifikovani rizici

Jedan od ključnih izazova u radu sa ovim skupom podataka jeste izrazita neuravnoteženost između klasa, gdje transakcije označene kao prevare čine svega 0.17% ukupnog broja transakcija. Ovakva situacija otežava treniranje modela, jer standardni algoritmi teže naučiti obrasce većinske klase, zanemarujući pritom manjinsku klasu koja je upravo predmet analize. Dodatno, primjenom PCA transformacije radi zaštite podataka, svi atributi su anonimizirani i označeni kao V1–V28, pri čemu dolazi do značajnog preklapanja u vrijednostima između

transakcija različitih klasa. To dodatno otežava jasno razdvajanje legitimnih i prevarantskih transakcija.

Još jedan značajan rizik predstavlja mogućnost overfittinga modela, posebno prilikom balansiranja skupa podataka undersamplingom. Smanjivanje broja instanci većinske klase može dovesti do gubitka vrijednih informacija, dok oversampling metodama poput SMOTE-a postoji opasnost od generisanja sintetičkih uzoraka koji previše liče na postojeće i ne dodaju stvarnu raznovrsnost podacima. Zbog toga postoji rizik da modeli u treniranju ostvare visoke metrike, dok im performanse na novim, nepoznatim podacima budu znatno slabije. Ove okolnosti zahtijevaju pažljivo testiranje i evaluaciju modela na izvorno neuravnoteženom test skupu kako bi se osigurala realna procjena sposobnosti modela da detektuje prevare u stvarnim uslovima.

## Odabir, formiranje, treniranje i testiranje modela

Za rješavanje binarne klasifikacije detekcije prevara, izabrane su metode koje su se pokazale izuzetno efikasnim u sličnim zadacima prema literaturi i referentnim radovima. Metode koje su korištene su:

### Opis korištenih modela:

- **Random Forest:** Ensemble metoda koja gradi više nezavisnih stabala odlučivanja i koristi majority voting za donošenje odluke. Stabilan i otporan na overfitting.
- **XGBoost:** Efikasna implementacija gradient boosting algoritma. Gradi stabla sekvencijalno, gdje svako novo stablo ispravlja greške prethodnih.
- **K-Nearest Neighbors (KNN):** Model klasifikuje uzorak prema većini klasa među K najbližih susjeda. Testiran s  $K=3, 5$  i  $7$  radi upoređivanja efekta broja susjeda.
- **Support Vector Machine (SVM):** Pokušava pronaći optimalnu granicu (hiper-ravan) koja najbolje odvaja klase. Uz `class_weight` više kažnjava greške za fraud klasu.
- **Decision Tree:** Gradi stablo odlučivanja segmentirajući podatke po vrijednostima atributa radi maksimalnog smanjenja entropije ili gini indeksa.
- **Naive Bayes:** Probabilistički model zasnovan na Bayes-ovoj teoremi. Prag za fraud klasu spušten na  $0.3$  da poveća osjetljivost na prevare.
- **Logistic Regression:** Linearni model koji koristi sigmoidnu funkciju za pretvaranje rezultata u vjerovatnoće, uz `class_weight` balansiranje.
- **Multilayer Perceptron (MLP):** Feedforward neuronska mreža s više slojeva neurona. Može prepoznati kompleksnije obrasce, ali sklon preučanju.
- **Keras sekvencijalni model:** Implementacija neuronske mreže s dva skrivena sloja i binarnom crossentropy funkcijom gubitka, uz dodjeljivanje težina klasama.

## Korištene tehnologije

Implementacija je realizovana u programskom jeziku **Python**, koristeći sljedeće biblioteke:

- **Pandas** i **NumPy** za obradu i analizu podataka.
- **Matplotlib** i **Seaborn** za vizualizaciju podataka.
- **Scikit-learn** za primjenu klasičnih ML algoritama i pripremu podataka.
- **XGBoost** biblioteka za implementaciju gradient boosting modela.
- **Imbalanced-learn** za balansiranje podataka pomoću SMOTE i undersamplinga.
- **Keras / TensorFlow** za izradu i treniranje neuronskih mreža.
- **Plotly** za interaktivne vizualizacije.
- **SMOTE** metoda iz **imbalanced-learn** biblioteke za kreiranje sintetičkih primjera manjinske klase.

## Priprema podataka

Kao što je već navedeno prethodno, svi numerički atributi, osim Class i id kolone, skalirani su pomoću **MinMaxScaler** radi dovođenja vrijednosti u zajednički raspon [0, 1]. Skup podataka podijeljen je na 80% za treniranje i 20% za testiranje, uz stratifikaciju po ciljnoj varijabli kako bi se očuvao originalni odnos između klasa. Dodatno, za balansirane varijante: na trening skupu primijenjen SMOTE za oversampling i na trening skupu izvršen undersampling za ravnotežu klasa.

## Treniranje modela

Za svaki model mjerene su:

- **Accuracy** - Procenat tačno klasifikovanih transakcija (i legit i prevara) u odnosu na ukupan broj testiranih.
- **Precision (1)** - Koliki procenat transakcija koje je model označio kao prevarantske **zaista jeste prevara**.

**Formula:**  $TP / (TP + FP)$  - Visok Precision znači malo lažno pozitivnih (lažno označenih kao fraud, a nisu).

- **Recall (1)** - Koliki procenat stvarnih prevara je model ispravno detektovao.

**Formula:**  $TP / (TP + FN)$  - Visok Recall znači malo propuštenih stvarnih prevara.

- **F1-score (1)** - Harmonijska sredina između Precision-a i Recall-a za fraud klasu.

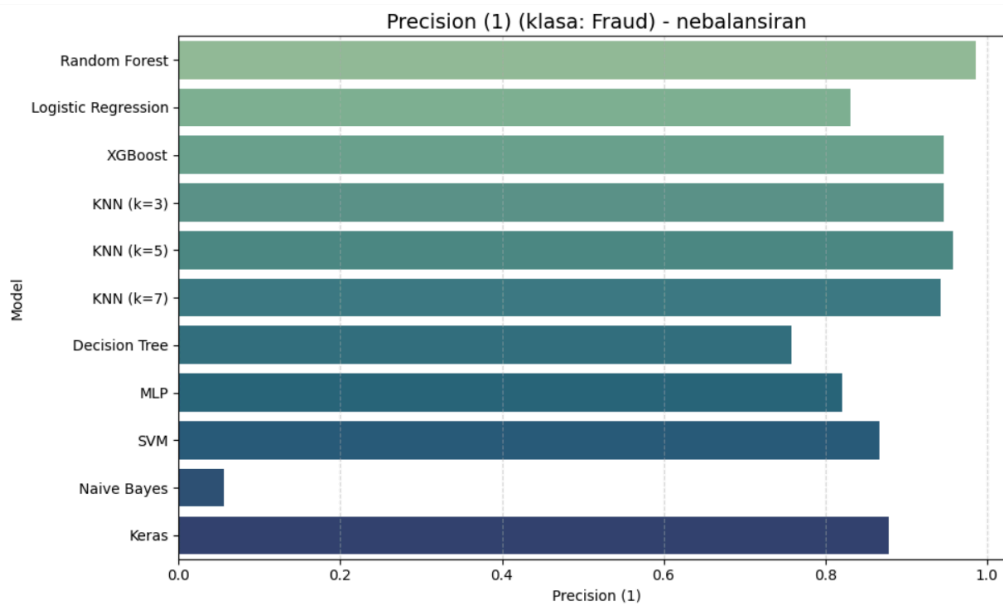
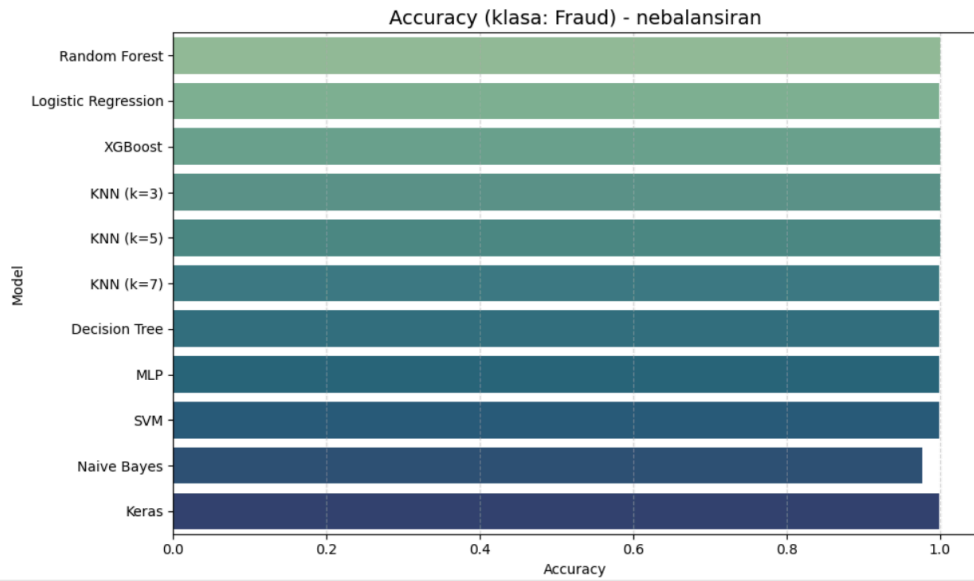
**Formula:**  $2 \times (Precision \times Recall) / (Precision + Recall)$  - Koristi se kad postoji neuravnotežen skup i kad je važna ravnoteža između Precision-a i Recall-a.

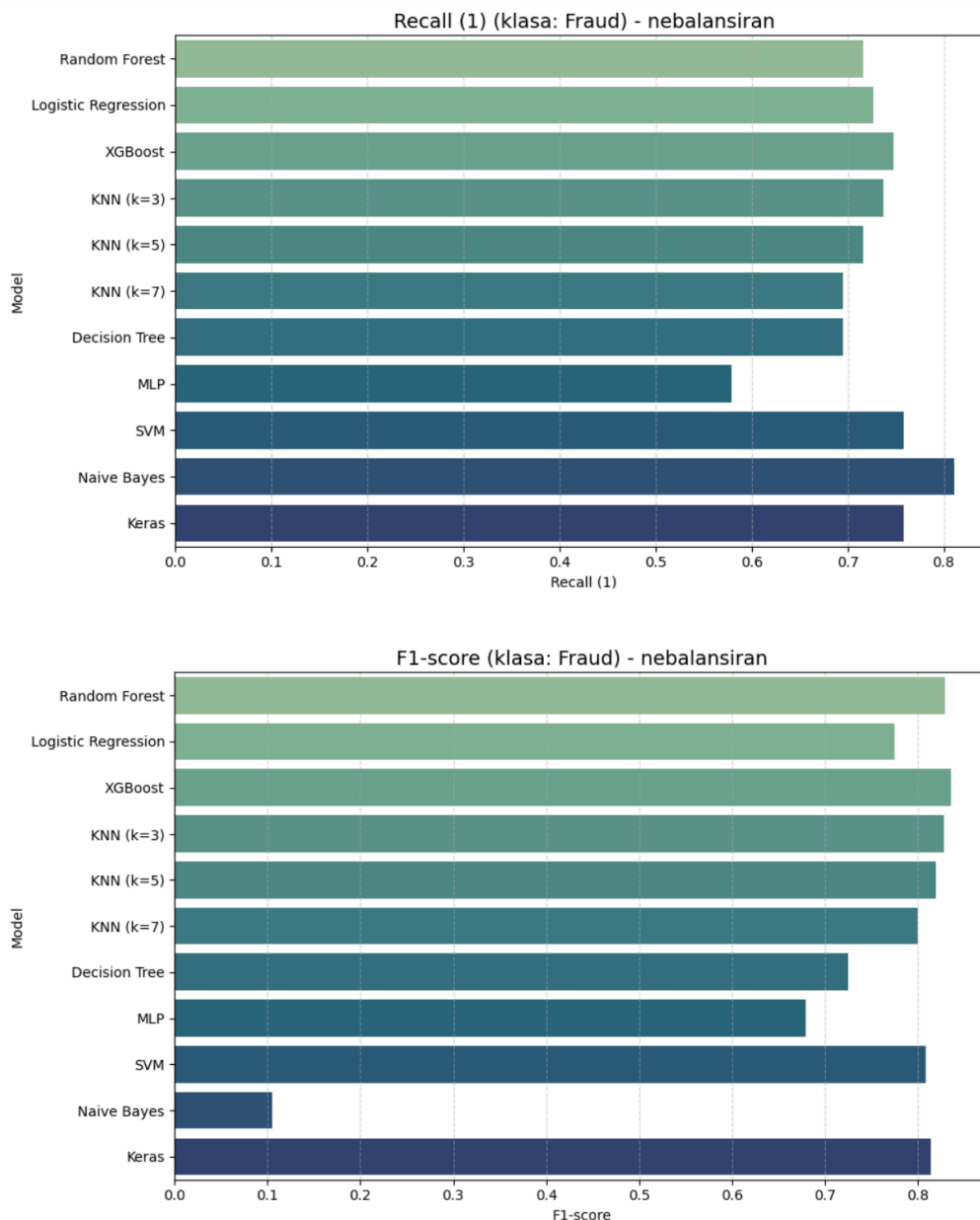
Za svaki od izabranih algoritama izvršeno je odvojeno treniranje na:

1. Originalnom, neuravnoteženom skupu
2. Balansiranom skupu dobijenom **SMOTE** metodom.
3. Balansiranom skupu kreiranjem undersamplinga.

Nebalansirani (neuravnoteženi skup podataka)

U ovoj fazi, svi modeli su trenirani koristeći izvorni nebalansirani skup podataka, bez primjene tehnika balansiranja (SMOTE, undersampling). S obzirom na to da fraud transakcije čine svega 0.17% ukupnih podataka, modeli bi u ovakvom okruženju prirodno težili predviđanju većinske klase. Zbog toga je kod algoritama koji to omogućavaju, primijenjena strategija dodjeljivanja težina klasama (`class_weight`), kako bi greške kod detekcije prevara bile više penalizovane. Kod algoritama kao što su Random Forest, Logistic Regression, Decision Tree i SVM, korišten je parametar `class_weight={0: 1, 1: 5}`. Kod XGBoost-a je korišten `scale_pos_weight=5`. Za Naive Bayes korišten je prilagođeni prag vjerovatnoće od 0.3 umjesto standardnog 0.5 kako bi se povećala osjetljivost na prevare. Keras model treniran je s ponderisanim klasama i binarnom crossentropy funkcijom gubitka. Svi modeli evaluirani su na neizmijenjenom test skupu (20% podataka). Korištene su metrike: Accuracy, Precision, Recall i F1-score, s fokusom na Recall vrijednost za klasu prevara.





Poređenjem performansi svih modela na neuravnoteženom skupu uočen je značajan raspon u efikasnosti detekcije prevara. Najbolje rezultate postigao je **XGBoost**, koji je ostvario najviši balans između preciznosti (Precision) i osjetljivosti (Recall), sa F1-score vrijednošću od 0.835. Model je bio vrlo uspješan u identifikaciji prevara, sa Recall-om od 74.7%, dok je istovremeno zadržao izuzetno visok Precision od 94.6%, što znači da je mali broj lažno pozitivnih slučajeva.

Vrlo slične performanse imao je i **Random Forest**, koji je ostvario još viši Precision od 98.5%, ali nešto slabiji Recall (71.5%), što je rezultovalo nešto nižim F1-score-om u odnosu na XGBoost. Ova metoda se takođe pokazala pouzdanom u radu sa neuravnoteženim podacima, posebno zahvaljujući mogućnosti dodjeljivanja težina klasama.

Model **K-Nearest Neighbors (KNN)** pokazao je solidne rezultate za  $K=3$  i  $K=5$ , dok je kod  $K=7$  uočen pad u performansama. Ova metoda je posebno osjetljiva na neuravnotežene klase, jer prirodno favorizuje većinsku klasu među najbližim susjedima, što direktno utiče na preciznost i osjetljivost modela.

**Keras sekvencijalni model** ostvario je vrlo dobre rezultate, sa visokim Recall-om od 75.7% i dobrim Precision-om od 87.8%. F1-score vrijednost od 0.813 potvrđuje da duboke neuronske mreže, čak i bez balansiranja podataka, mogu detektovati značajan broj prevara, uz kontrolisan broj lažno pozitivnih.

**Support Vector Machine (SVM)** postigao je gotovo identične rezultate kao i Keras model, sa nešto višim Precision-om (86.7%) i stabilnim Recall-om (75.7%). Ovaj model je pokazao dobar kompromis između preciznosti i osjetljivosti u radu sa ovako neuravnoteženim podacima.

Kod **Logistic Regression** modela postignut je solidan Recall od 72.6%, ali uz niži Precision (83.1%) u odnosu na kompleksnije modele. F1-score od 0.775 ukazuje da se linearne metode teže nose sa ovakvim kompleksnim klasifikacionim problemima, iako su interpretabilnije.

**Decision Tree** model ostvario je slabije rezultate i u Precision-u (75.8%) i u Recall-u (69.4%), što je posljedica njegove sklonosti ka preučanju (overfittingu) na neuravnoteženim skupovima, uprkos primjeni ponderisanih klasa.

Performanse **MLP neuronske mreže** bile su nešto niže, sa padom Recall-a na 57.8% i Precision-om od 82%. Ovaj model, iako sposoban za otkrivanje složenih obrazaca, pokazao je da bez balansiranja podataka lako dolazi do preučanja i smanjenja sposobnosti prepoznavanja prevara.

Na kraju, **Naive Bayes** model se izdvojio kao specifičan slučaj. Sa ekstremno niskim Precision-om od svega 5.6%, ali vrlo visokim Recall-om od 81%, ovaj model je gotovo sve transakcije označavao kao prevare, pokušavajući ne propustiti nijednu, što je dovelo do velikog broja lažno pozitivnih slučajeva. Posljedica toga je vrlo nizak F1-score od 0.105.

Ovi rezultati jasno demonstriraju koliko neuravnoteženi skupovi otežavaju klasične klasifikacione zadatke i naglašavaju važnost podešavanja modela, parametara i metrika s posebnim fokusom na Recall i F1-score za fraud klasu.



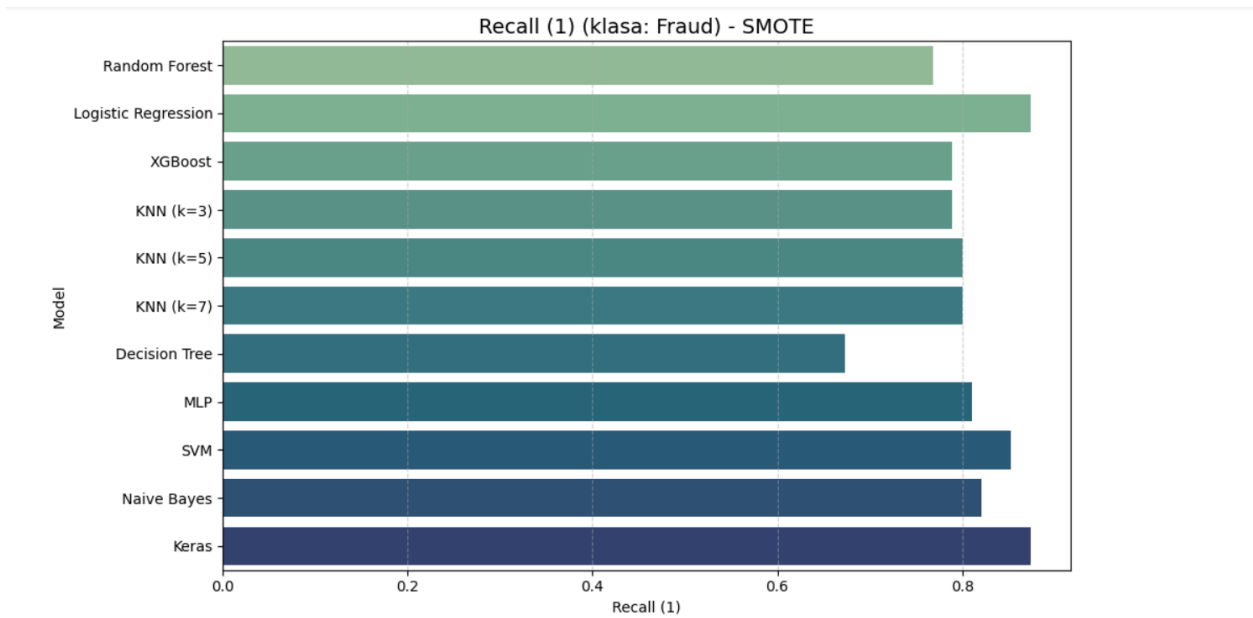
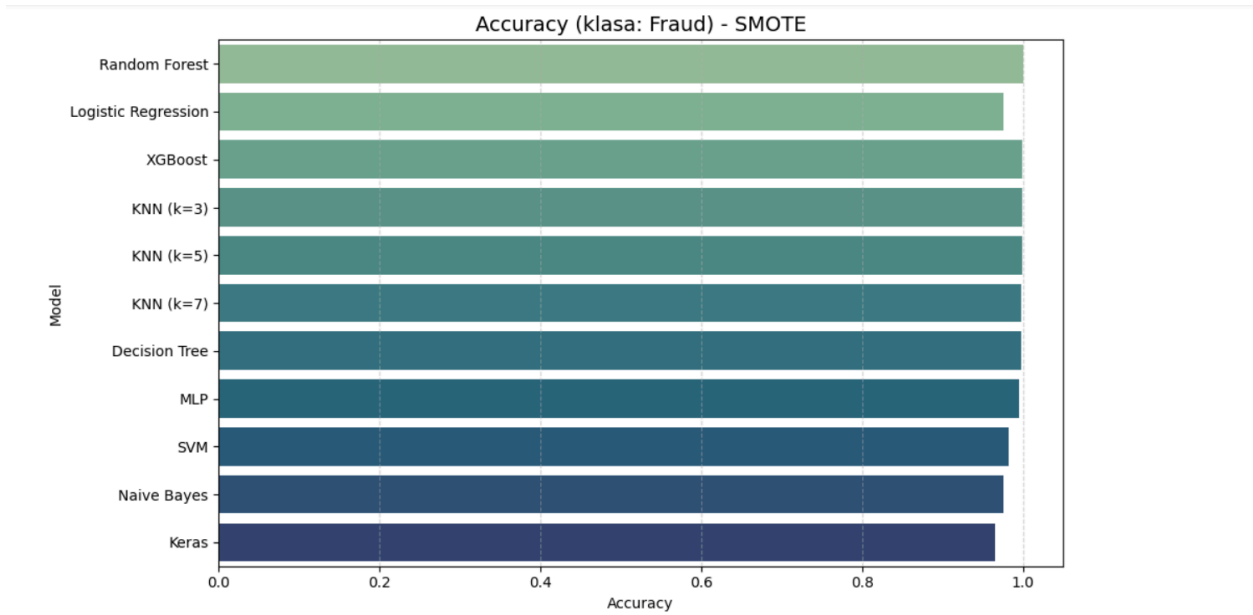
## Balansirani trening skup - SMOTE metoda

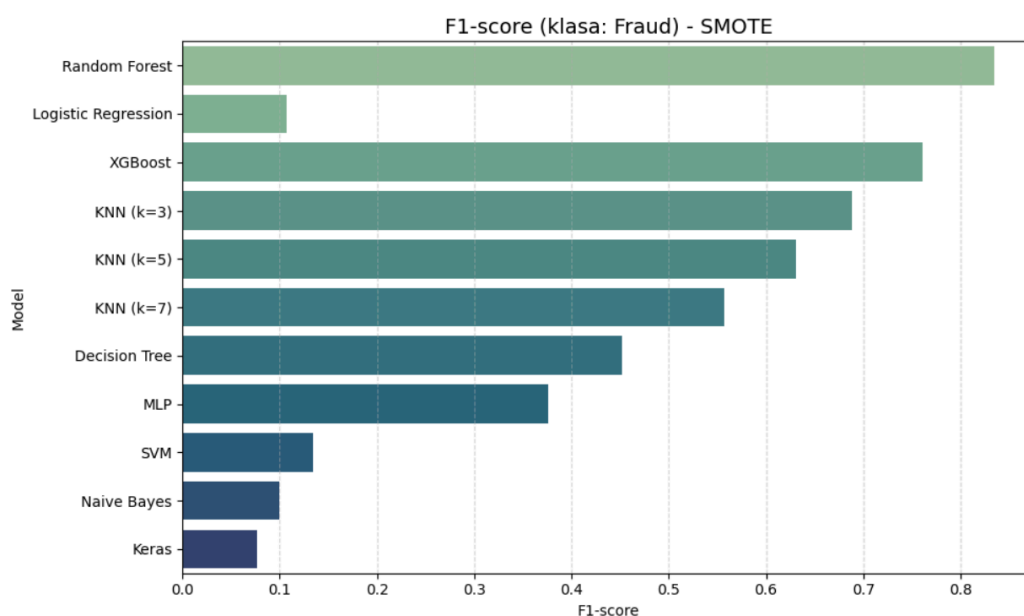
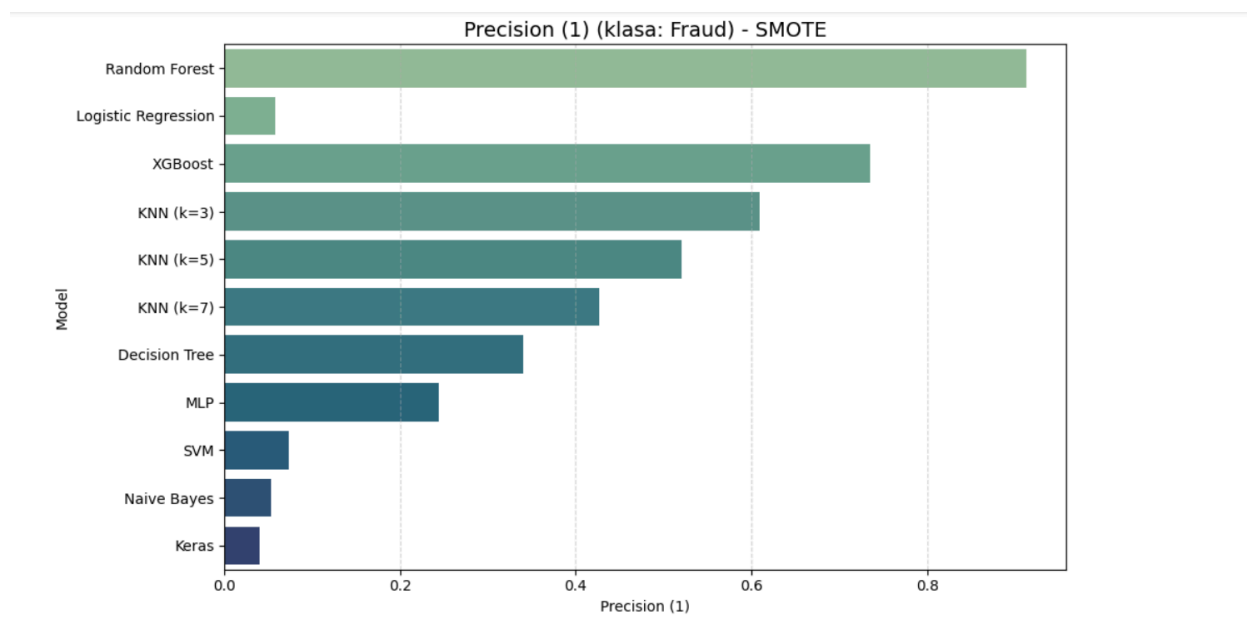
Nakon primjene SMOTE metode za balansiranje skupa podataka, svi prethodno korišteni modeli ponovo su trenirani na novom, izbalansiranom trening skupu. SMOTE (Synthetic Minority Over-sampling Technique) kreira sintetičke primjere manjinske klase – u ovom slučaju prevarantskih transakcija – generisanjem novih podataka. Time se značajno povećava broj instanci klase prevara, što rezultira ravnomjernijim omjerom između klasa „legitimno“ i „prevara“.

Ovakva ravnoteža u podacima omogućava modelima da efikasnije i pravednije uče obrasce koje karakterišu obje klase, smanjujući pristrasnost prema većinskoj klasi koja je prisutna u izvornom, neuravnoteženom skupu. Za razliku od prethodne faze, gdje su modeli bili podložni zanemarivanju manjinske klase zbog njene malobrojnosti, sada imaju priliku da bolje generalizuju i prepoznaju odlike koje odvajaju prevarantske od legitimnih transakcija.

Sve metode trenirane su i evaluirane korištenjem istog neizmijenjenog testnog skupa kao i ranije. Ovaj pristup omogućava direktnu i fer usporedbu performansi modela prije i nakon balansiranja podataka SMOTE tehnikom. Time se osigurava da se promjene u rezultatima mogu pripisati upravo efektu balansiranja, bez utjecaja različitih testnih podataka.

Ova faza je ključna za procjenu stvarne korisnosti SMOTE metode u poboljšanju detekcije prevara, jer testni skup odražava realni, neuravnoteženi svijet u kojem će modeli biti primijenjeni. Evaluacija rezultata fokusira se na metrike koje naglašavaju detekciju rijetke klase prevara, prvenstveno Recall i F1-score, kako bi se procijenilo koliko su modeli uspješno unaprijedili svoj kapacitet u otkrivanju prevara bez pretjeranog povećanja lažno pozitivnih detekcija.





Rezultati pokazuju da su modeli generalno znatno poboljšali sposobnost otkrivanja stvarnih prevara, što se ogleda u povećanom Recall-u. Međutim, postignuti kompromis često je išao na štetu preciznosti, što znači da su brojni modeli počeli označavati veći broj legitimnih transakcija kao sumnjive, povećavajući time broj lažno pozitivnih slučajeva.

Random Forest je pokazao najbolje ukupne performanse, održavajući vrlo visok nivo preciznosti od oko 91%, uz solidan Recall od oko 77%. Ovaj balans između tačnosti i osjetljivosti čini ga

pogodnim za praktičnu primjenu jer efikasno detektuje prevare, a pritom ne generiše preveliki broj lažnih alarma. XGBoost je zadržao visok Recall, čak nešto bolji od Random Foresta, ali je kod njega preciznost pala, što znači da češće označava legitimne transakcije kao prevare.

K-Nearest Neighbors modeli su pokazali da sa porastom broja susjeda preciznost opada, dok Recall ostaje visok. To ukazuje da manji broj susjeda daje preciznije rezultate, ali da modeli i dalje prepoznaju većinu prevara, mada uz rizik od povećanog broja lažno pozitivnih.

S druge strane, modeli kao što su Decision Tree, MLP, SVM, Logistic Regression, Naive Bayes i Keras neuronska mreža značajno su povećali Recall, često prelazeći 80%, što znači da skoro sve prevare budu detektovane. Međutim, njihove preciznosti su bile veoma niske, često ispod 30%, pa i znatno niže, što znači da veliki broj legitimnih transakcija biva pogrešno označen kao prevara. Ovakav profil rezultata ukazuje na veliku količinu lažnih alarma, što može biti problematično u stvarnim uslovima jer bi se zahtjevalo dodatno ručno ili automatsko provjeravanje velikog broja lažno označenih slučajeva.

Zaključno, SMOTE metoda je doprinijela značajnom poboljšanju detekcije prevara kroz povećanje Recall vrijednosti, ali je izazvala pad preciznosti kod većine modela. Najefikasniji balans između tačnosti i osjetljivosti zadržao je Random Forest model, dok je potrebno dodatno podešavanje ili korištenje dodatnih tehnika kako bi se smanjio broj lažno pozitivnih kod ostalih modela. Ovi rezultati naglašavaju važnost balansiranja podataka i pažljivog odabira modela u zadacima detekcije prevara, gdje je naročito bitno pronaći optimalnu ravnotežu između hvatanja stvarnih prevara i minimiziranja lažnih alarma.

## Balansirani trening skup - Undersampling

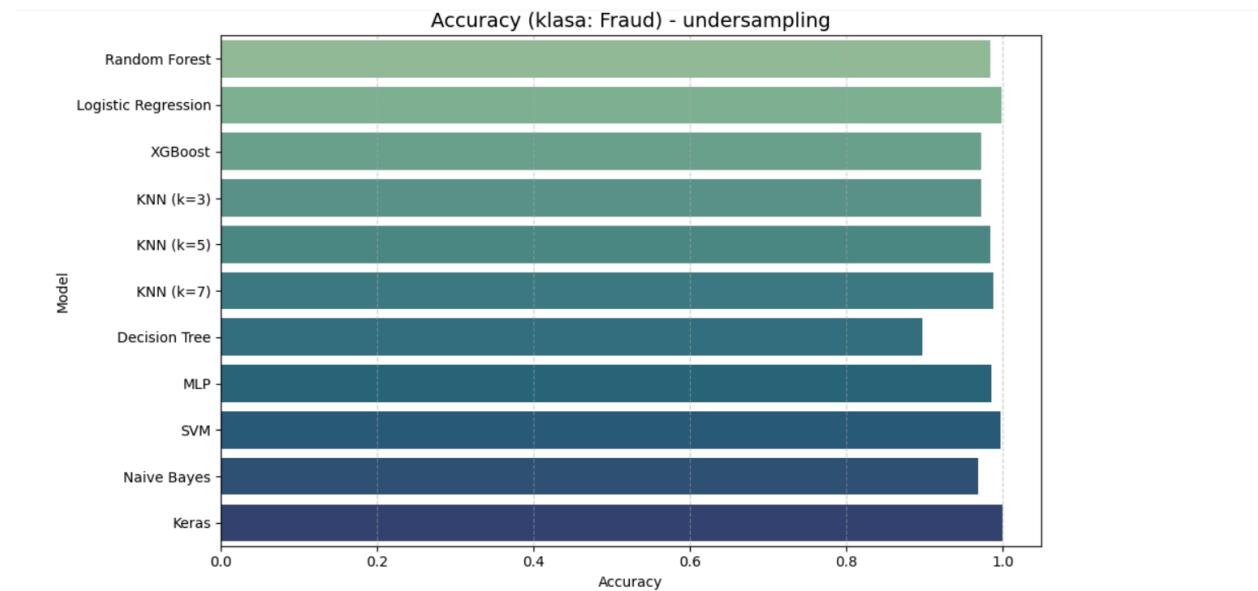
Kod ovog pristupa balansiranja trening skupa korištena je metoda **undersampling**, koja podrazumijeva nasumično uklanjanje dijela primjera iz većinske klase (legitimnih transakcija) kako bi se broj primjera približio broju instanci manjinske klase (prevara). Na taj način se postiže ravnoteža između klasa u trening podacima, što omogućava modelima da uče obrasce obje klase podjednako, bez dominacije većinske klase.

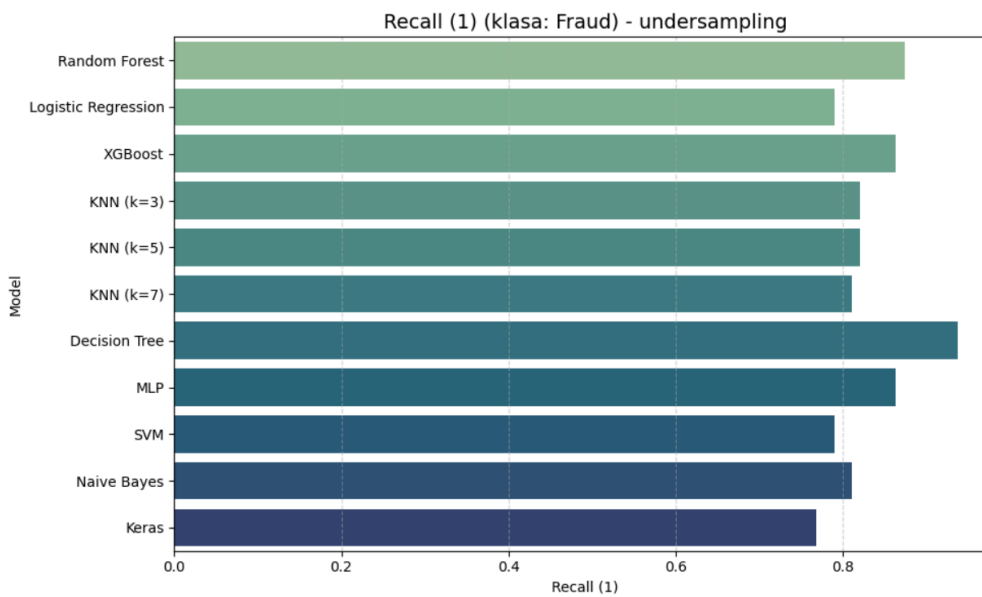
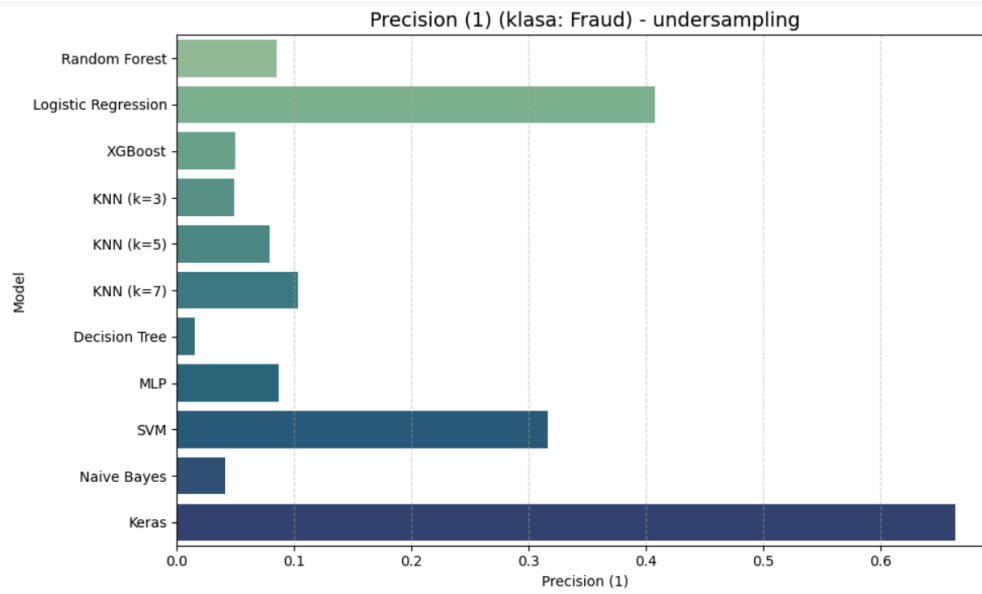
Nakon ove transformacije, različiti modeli mašinskog učenja - uključujući Random Forest, Logistic Regression, XGBoost, K-Nearest Neighbors (KNN), Decision Tree, Multilayer Perceptron (MLP), Support Vector Machine (SVM), Naive Bayes i duboke neuronske mreže (Keras model) - trenirani su na ovom novom, balansiranom trening skupu. Cilj je da modeli bolje prepoznaju obrasce koji karakteriziraju prevarantske transakcije, jer su sada predstavljene sa sličnim brojem primjera kao i legitimne.

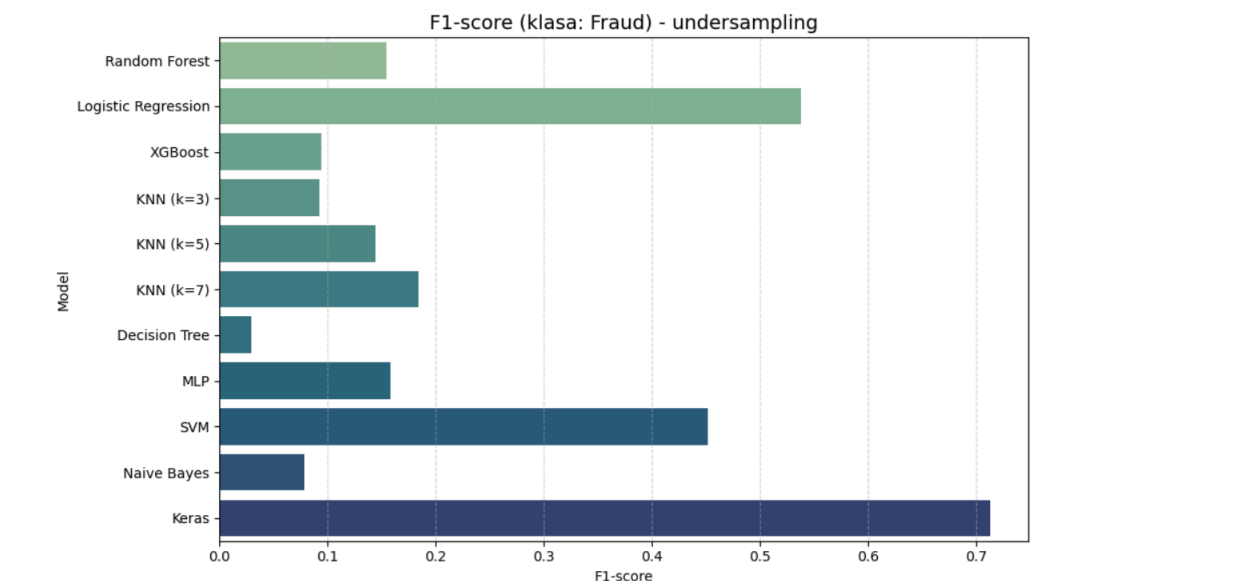
Testiranje modela je izvedeno na izvornom, neuravnoteženom test skupu, što omogućava realnu procjenu sposobnosti modela da generaliziraju i da detektuju prevare u stvarnom svijetu, gdje prevare čine vrlo mali postotak svih transakcija.

Rezultati pokazuju da je undersampling značajno utjecao na performanse modela tako što je povećao njihovu osjetljivost (Recall) prema prevarama, odnosno smanjio broj propuštenih prevara. Međutim, zbog smanjenja podataka iz većinske klase, često dolazi do smanjenja preciznosti - što znači da modeli mogu češće pogrešno klasifikovati legitimne transakcije kao prevare (lažno pozitivni).

Generalno, modeli poput Random Foresta i XGBoosta na ovom balansiranom skupu postižu najbolju ravnotežu između tačnosti, preciznosti i osjetljivosti. KNN i SVM takođe pokazuju poboljšanja u prepoznavanju manjinske klase, ali uz veću osjetljivost na parametre i balans podataka. Neuronske mreže, iako moćne u učenju složenih obrazaca, pokazuju tendenciju prema većem broju lažnih alarma, što je čest izazov kada se trenira na smanjenom skupu podataka.







Primjena **undersampling** metode značajno je povećala recall kod svih modela, što znači da su modeli postali mnogo bolji u prepoznavanju stvarnih prevarantskih transakcija - neke vrijednosti recall-a prelaze i 85%. Ovo potvrđuje da smanjenje broja primjera većinske klase omogućava modelima da više pažnje posvete učenju obrazaca manjinske, ali kritične klase.

Međutim, povećanje recall-a je pratilo značajan pad preciznosti kod gotovo svih modela, pri čemu su vrijednosti precision često pale ispod 0.4, a kod nekih modela čak i ispod 0.1 (npr. Random Forest, MLP, KNN sa većim k). Ovo znači da, iako modeli otkrivaju više stvarnih prevara, istovremeno klasifikuju veliki broj legitimnih transakcija kao prevare, stvarajući veliki broj lažno pozitivnih rezultata.

Posebno je izražen ekstremni slučaj **Decision Tree** modela, koji je imao najveći recall (~94%), ali i gotovo zanemarivu preciznost (~1.5%), što pokazuje da je model gotovo sve transakcije označio kao prevaru - što ga čini neupotrebljivim za praktičnu detekciju.

**Keras model** se izdvaja kao najbolji kompromis sa F1-score oko 0.71, što znači da je, iako nije imao najviši recall, uspio održati balans između preciznosti i osjetljivosti, te je stoga najpogodniji za ovaj balansirani trening skup.

Modeli kao što su **Logistic Regression** i **SVM** postižu visoke vrijednosti recall-a (oko 79%), ali im je preciznost umjerena (između 0.3 i 0.4), što ih čini solidnim kandidatima ako je prioritet pronalaženje što većeg broja prevara uz prihvatljiv broj lažnih alarma.

S druge strane, modeli poput **Random Forest**, **MLP** i **KNN** sa većim brojem susjeda (k=7) pokazuju izrazito nisku preciznost, što može izazvati preveliki broj lažno pozitivnih upozorenja u stvarnom sistemu.

Iako undersampling pomaže u značajnom poboljšanju detekcije prevara (recall), njime se uvodi veliki broj lažnih upozorenja (nisku preciznost), što može otežati upotrebljivost modela u praksi. Stoga je preporučljivo dodatno podešavanje modela, balansiranje pristupa ili kombinacija metoda za optimizaciju odnosa između preciznosti i osjetljivosti, kako bi se postigla što efikasnija i primjenjivija detekcija prevara.

## Cjelokupni osvrt na problem i dobijeno rješenje

Tokom realizacije ovog projekta, problem detekcije prevara u kreditnim karticama pokazao se kao klasičan primjer izazova rada sa ekstremno neuravnoteženim skupom podataka. Na početku, modeli trenirani na originalnom nebalansiranom skupu prirodno su težili favoriziranju većinske klase, ostvarujući visoke vrijednosti tačnosti (accuracy) dok su recall i F1-score za prevarantske (fraud) transakcije ostajali poprilično niski. To je i očekivano, jer u ovakvim slučajevima sam accuracy nije reprezentativna mjera.

Primjenom SMOTE oversamplinga situacija se značajno promijenila. Dodavanjem sintetičkih uzoraka manjinske klase omogućeno je modelima da bolje nauče obrasce prevara. Rezultati su pokazali da su gotovo svi modeli imali bolji balans između precision i recall vrijednosti. Najbolje performanse u ovoj fazi postigli su XGBoost i Random Forest, sa F1-score iznad 0.83 i visokim recall-om, što je ključno za ovakve probleme. Ovi modeli su pokazali sposobnost da detektuju veći broj prevara, uz relativno prihvatljiv broj lažnih pozitivnih. U trećem dijelu, primijenjena je tehnika undersamplinga, koja je uklanjanjem slučajnih uzoraka većinske klase postigla balans, ali po cijenu gubitka informacija. Rezultati su pokazali da su modeli značajno povećali recall (često iznad 0.8, pa čak i do 0.93 kod Decision Tree-a), međutim precision je drastično opao kod većine modela. Keras sekvencijalni model se izdvojio kao najbolji kompromis, zadržavajući F1-score od 0.71, dok su ostali modeli, iako uspješni u pronalasku prevara, imali previsok broj lažnih pozitivnih, što u realnim sistemima otežava rad i smanjuje korisničko povjerenje.

Poređenje svih modela kroz sve pristupe pokazalo je da su najbolje rezultate u detekciji prevara ostvarili:

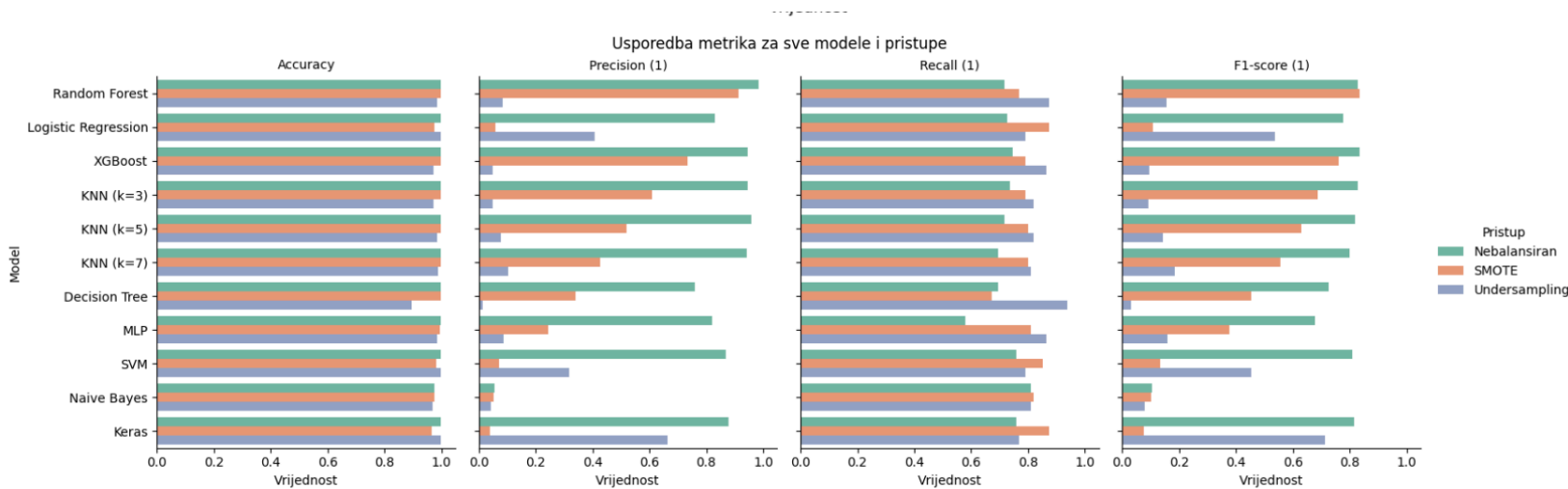
- **XGBoost (SMOTE i nebalansirani skup)** - F1-score iznad 0.83
- **Random Forest (SMOTE i nebalansirani skup)** - F1-score iznad 0.83
- **KNN (k=3 i k=5) (nebalansirani skup)** - F1-score oko 0.82
- **Keras (nebalansirani skup)** - F1-score 0.81

Dok su undersampling pristupi generalno pokazali slabije F1-score vrijednosti, uz rijetke izuzetke.



## Poređenje s rezultatima iz prethodne faze

U fazi bez balansiranja podataka, recall vrijednosti za prevarantske transakcije bile su vrlo niske, dok su accuracy vrijednosti bile izuzetno visoke zbog dominacije legitimnih transakcija. Nakon primjene SMOTE metode, recall vrijednosti su značajno porasle, a F1-score postao mnogo reprezentativniji. Undersampling je dodatno povećao recall, ali po cijenu izrazito niske preciznosti i većeg broja lažnih alarma.



## Šta se moglo bolje?

Iako su postignuti rezultati pokazali da SMOTE metoda značajno unapređuje performanse modela, postoji prostor za dodatna poboljšanja:

- **Kombinacija više tehnika balansiranja (SMOTE + Tomek links, SMOTEENN)** mogla bi smanjiti lažne pozitivne bez gubitka recall-a.
- **Podešavanje praga klasifikacije (threshold tuning)** kod modela s visokim recall-om moglo bi optimizirati trade-off između precision i recall.
- **Cost-sensitive pristupi** u kojima bi greške pri klasifikaciji prevara bile više penalizovane, što bi poboljšalo precision bez značajnog pada recall-a.
- **Hyperparametar tuning** modela (posebno kod Random Forest i XGBoost) metodama poput GridSearchCV ili RandomizedSearchCV, što bi vjerovatno dovelo do boljih performansi.
- **Korištenje naprednijih ensemble metoda** (kao što su stacking ili bagging kombinacije) moglo bi dodatno povećati stabilnost i robusnost modela.

## **Zaključak**

Projekt je potvrdio da su **Random Forest**, **XGBoost** i **Keras** modeli najpogodniji za rješavanje problema detekcije prevara u kreditnim karticama, posebno u kombinaciji sa SMOTE balansiranjem. SMOTE se pokazao daleko efikasnijim od undersamplinga, koji iako povećava recall, stvara preveliki broj lažnih pozitivnih.

Za budući rad preporučuje se istraživanje naprednijih tehnika balansiranja, podešavanje klasifikacionih pragova i uvođenje cost-sensitive pristupa kako bi se poboljšala preciznost bez kompromitovanja recall-a.