10/4/2023

Bank Fraud Detection

Project Report

By

1.Shravni Dhokare

2.Pankaj Kshirsagar

3.Nithesh N Gurjar

4.Yathish V

5.Chethan Kumar

(Group 1)

Under the Guidance of

Mr.Akash Maurya

Introduction

Bank fraud detection is a specialized field within the financial sector dedicated to the identification and prevention of fraudulent activities occurring in banks and other financial institutions. It serves as a vital safeguard for protecting financial assets, ensuring the security of customer data, and upholding the integrity and trustworthiness of the entire financial system. This field encompasses a wide range of fraudulent activities, including credit card fraud, identity theft, account takeovers, check fraud, insider fraud, phishing, and online banking fraud. To tackle these threats, bank fraud detection relies heavily on advanced data analytics, machine learning, and artificial intelligence. Banks collect extensive transaction data and customer information, employing these tools to detect unusual patterns, anomalies, and potential indicators of fraud.

Real-time monitoring is a cornerstone of bank fraud detection, enabling the prompt identification and prevention of fraudulent transactions. Automated systems continuously scrutinize activities, flagging any suspicious behavior as it occurs and triggering alerts for further investigation. Behavioral analysis is another crucial aspect, wherein changes or unusual activities in customer behavior over time are monitored to raise potential red flags. Machine learning models, both supervised and unsupervised, are used to classify transactions as legitimate or fraudulent. These models learn from historical data and are continuously updated to adapt to evolving fraud patterns.

Bank fraud detection is closely linked to regulatory compliance, with financial institutions required to adhere to strict laws and regulations governing anti-money laundering (AML), Know Your Customer (KYC) requirements, and the reporting of suspicious activities. Moreover, customer education plays a pivotal role in fraud prevention, as customers are encouraged to adopt security best practices and report any suspicious activities promptly. Robust cybersecurity measures, including encryption, firewalls, and intrusion detection systems, are essential to protect customer data and financial systems from cyberattacks and data breaches. The field of bank fraud detection is dynamic, continuously evolving to keep pace with fraudsters who adapt their tactics. As such, it necessitates continuous improvement in detection methods and the implementation of updated fraud prevention strategies. Ultimately, bank fraud detection is an essential pillar of maintaining the security and trustworthiness of the global financial system.

Data Description

There are Following features in data:

1. **Label:** This is the target variable or the label you want to predict in a supervised machine learning model. In the context of a bank fraud detection system, it could represent whether a transaction is fraudulent (1) or not (0).

2. **MSISDN:** MSISDN stands for Mobile Station International Subscriber Directory Number. It is a unique identifier for a mobile phone user.

3. **AON (Age on Network):** AON represents the duration (in days) for which a customer has been using the network or mobile services.

4. **Daily_Decr30:** This could refer to the average daily debit amount over the last 30 days.

5. **Daily_Decr90:** Similarly, this could refer to the average daily debit amount over the last 90 days.

6. **Rental30:** Rental30 may indicate the rental cost incurred by a customer for a specific service or product over the last 30 days.

7. **Last_Rech_Date_MA:** This is the date of the last recharge done by the customer using mobile services.

8. **Last_Rech_Date_DA:** Similarly, this could represent the date of the last recharge but with a different service or product.

9. **Last_Rech_Amt_MA:** This is the amount of the last recharge made by the customer.

10. **Cnt_Ma_Rech30:** This may indicate the count of recharges made by the customer in the last 30 days.

11. **Fr_Ma_Rech30:** Fr_Ma_Rech30 could represent the frequency or count of recharges made by the customer in the last 30 days.

12. **Sumamnt_Ma_Rech30:** This feature may indicate the total amount spent on recharges by the customer in the last 30 days.

13. **Medianamnt_Ma_Rech30:** This could represent the median amount spent on recharges by the customer in the last 30 days.

14. **Medianmarechprebal30:** This feature may indicate the median balance before recharging for the last 30 days.

15. **Cnt_Ma_Rech90:** Similar to Cnt_Ma_Rech30, this represents the count of recharges but for the last 90 days.

16. **Fr_Ma_Rech90:** Fr_Ma_Rech90 represents the frequency or count of recharges made by the customer in the last 90 days.

17. **Sumamnt_Ma_Rech90:** This is the total amount spent on recharges by the customer in the last 90 days.

18. **Medianamnt_Ma_Rech90:** This could represent the median amount spent on recharges by the customer in the last 90 days.

19. **Medianmarechprebal90:** Similar to Medianmarechprebal30, this represents the median balance before recharging, but for the last 90 days.

20. **Cnt_Da_Rech30:** This could indicate the count of data recharges made by the customer in the last 30 days.

21. **Fr_Da_Rech30:** Fr_Da_Rech30 represents the frequency or count of data recharges made by the customer in the last 30 days.

22. **Cnt_Da_Rech90:** Similar to Cnt_Da_Rech30, this represents the count of data recharges but for the last 90 days.

23. **Fr_Da_Rech90:** Fr_Da_Rech90 represents the frequency or count of data recharges made by the customer in the last 90 days.

24. **Cnt_Loans30:** This feature could indicate the count of loans taken by the customer in the last 30 days.

25. **Amnt_Loans30:** Amnt_Loans30 represents the total amount of loans taken by the customer in the last 30 days.

26. **Maxamnt_Loans30:** This may indicate the maximum loan amount taken by the customer in the last 30 days.

27. **Medianamnt_Loans30:** This could represent the median loan amount taken by the customer in the last 30 days.

28. **Cnt_Loans90:** Similar to Cnt_Loans30, this represents the count of loans taken but for the last 90 days.

29. **Amnt_Loans90:** Amnt_Loans90 represents the total amount of loans taken by the customer in the last 90 days.

30. **Maxamnt_Loans90:** This may indicate the maximum loan amount taken by the customer in the last 90 days.

31. **Medianamnt_Loans90:** This could represent the median loan amount taken by the customer in the last 90 days.

32. **Payback30:** Payback30 may indicate the average payback time for loans taken in the last 30 days.

33. **Payback90:** Similarly, this could represent the average payback time for loans taken in the last 90 days.

34. **PCircle:** PCircle might represent the telecom circle or region in which the customer's mobile services are provided.

35. **PDate:** PDate represents the date of the data record or transaction.

Approach

I will be using step by step processing approach for Building Project. Steps will be as follows :

1. Data Collection:
Training and Testing dataset is already Given so no need to search for more historical data.

2. Feature Selection/Engineering:
Identifing which features (variables) are most relevant for predicting fraud . Correlation analysis Techinque is used to reduce redundancy. Selected fetures should impact accuracy of model in positive way so selected features should be more corelated with output variable i.e label which helps improving accuracy hence last_rech,median_amnt_loan,Daily_dcr,last_rech_date,rech_count,cnt_loan,total_amt_loan,max_loan,payback_r are selected features.

3.Feature Extraction:
Adding similar type of features helps to cover more feature in single column. Which reduces redundancy and avoids data loss. Ratio can be also taken from some features and not defined values generated by them can be filled with 0.

3. Data Preprocessing:
This is Most important step in order to filter out data. Data preprocessing have huge impact on accuracy of our model. It contains various tasks as follows :
a).Handling Missing values: There are various approach to do this. Here we have very limited dataset hence will not be removing tuples from dataset. I'll fill null values using Mean and Mode statistical terms.

4.Data Visualization :
Visualizing data using various plots like scatter, histogram, heatmap, pairplot, etc.

5.Data Normalization:
The data values have high scale difference. Hence to bring them on single scale of 0 to 1 , I used Normalization on all features. It improves efficiency of model

6.Train_Test_split:
It is very important step. Splits data in traing and testing data.

7. Model Selection:

Choosing a machine learning algorithm suitable for classification tasks. We have selected KNNmodel Because it is Binary Classification problem and after trying Logistic classification , KNN Gives more accuracy than others.

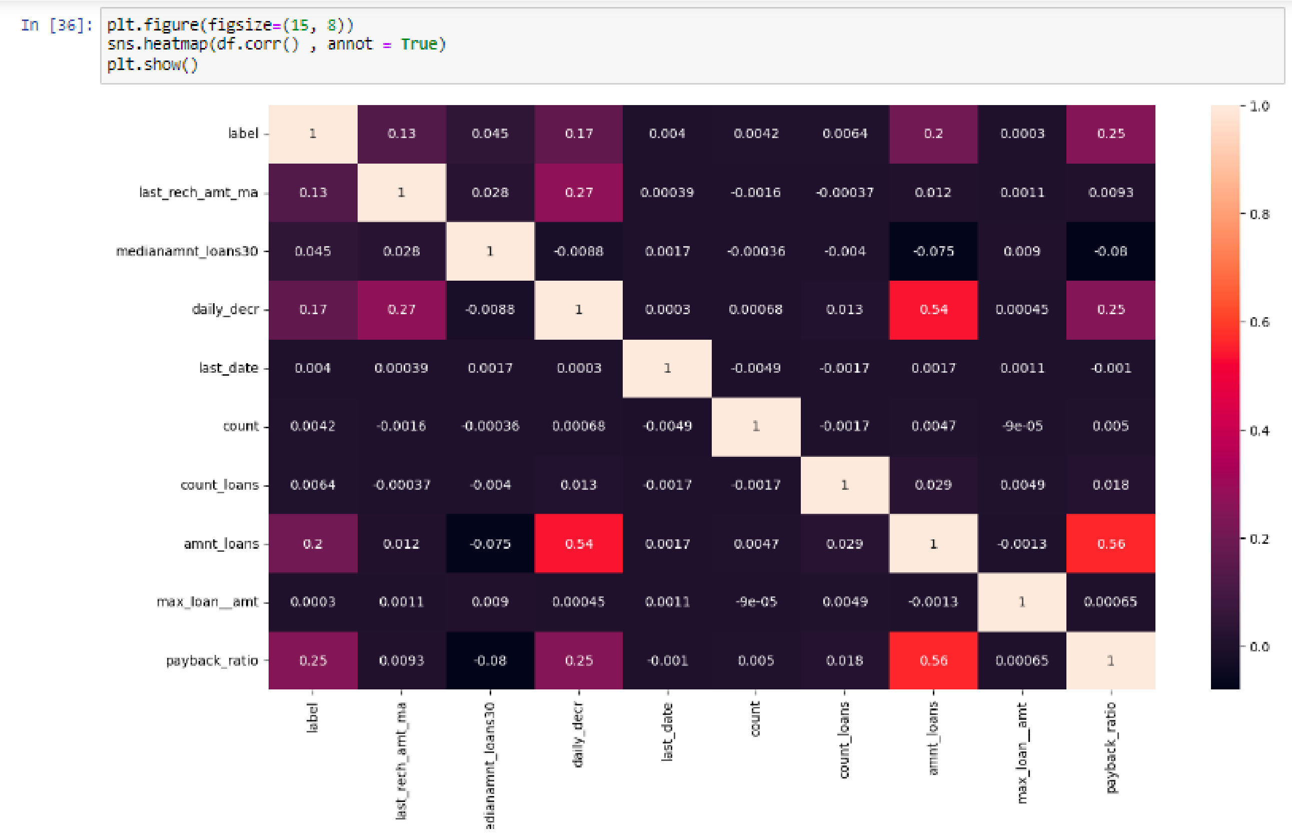
Visualization

Data visualization is the graphical representation of information and data in a pictorial or graphical format(Example: charts, graphs, and maps). Data visualization tools provide an accessible way to see and understand trends, patterns in data, and outliers.

Here are some of the Common Ways in Which the Visualization can be Done :
1.Heatmaps : Used to visualize the correlation between features. Heatmaps can highlight which features are most strongly correlated with loan approval.

2. Other Various Visualization are Histogram ,Column Chart , Plot, Boxplot, Pie Chart, Scatter Chart And etc's.

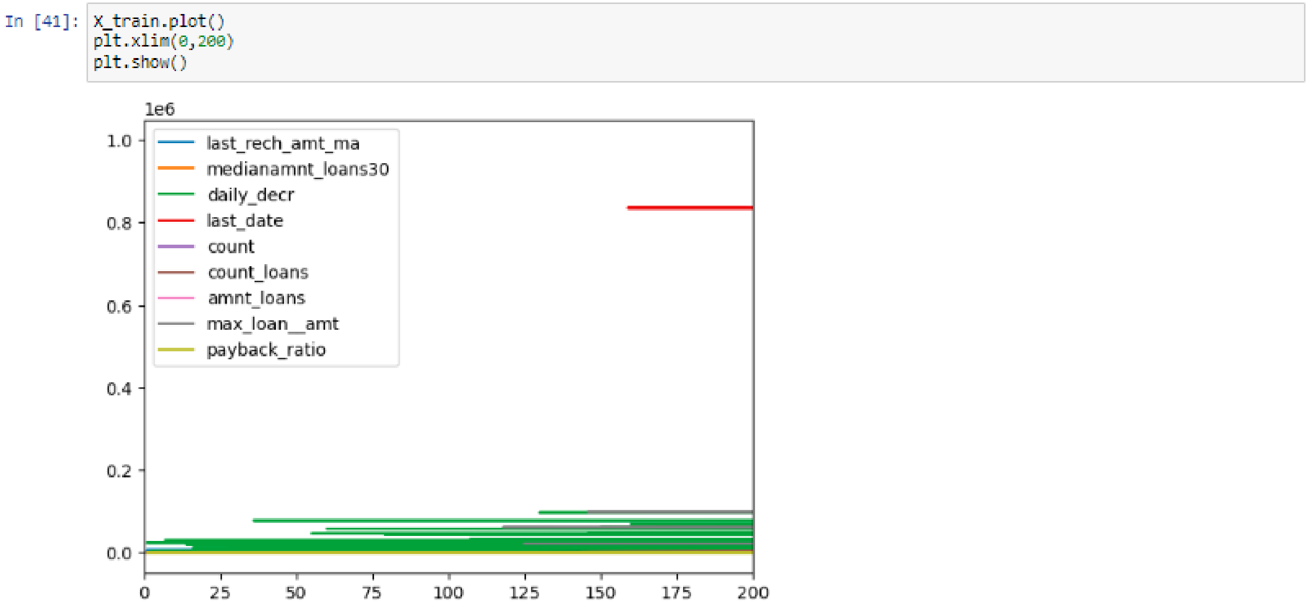
1.Heatmap :



```
In [36]: plt.figure(figsize=(15, 8))
         sns.heatmap(df.corr() , annot = True)
         plt.show()
```

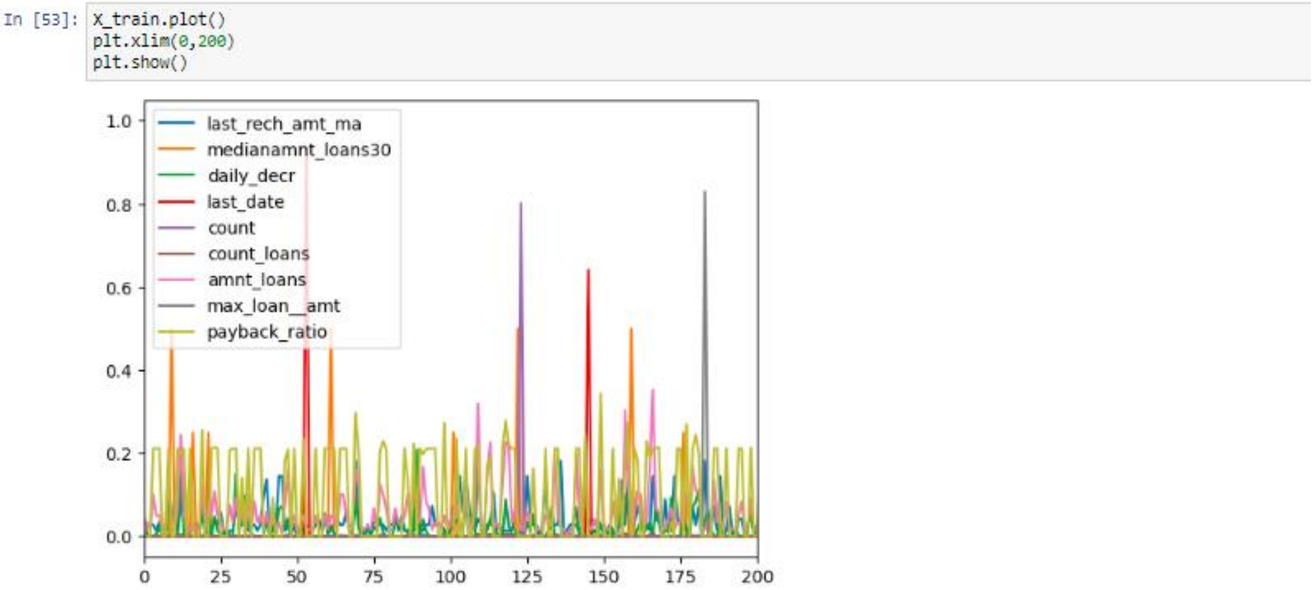
	label	last_rech_amt_ma	edianamnt_loans30	daily_decr	last_date	count	count_loans	amnt_loans	max_loan__amt	payback_ratio
label	1	0.13	0.045	0.17	0.004	0.0042	0.0064	0.2	0.0003	0.25
last_rech_amt_ma	0.13	1	0.028	0.27	0.00039	-0.0016	-0.00037	0.012	0.0011	0.0093
medianamnt_loans30	0.045	0.028	1	-0.0088	0.0017	-0.00036	-0.004	-0.075	0.009	-0.08
daily_decr	0.17	0.27	-0.0088	1	0.0003	0.00068	0.013	0.54	0.00045	0.25
last_date	0.004	0.00039	0.0017	0.0003	1	-0.0049	-0.0017	0.0017	0.0011	-0.001
count	0.0042	-0.0016	-0.00036	0.00068	-0.0049	1	-0.0017	0.0047	-9e-05	0.005
count_loans	0.0064	-0.00037	-0.004	0.013	-0.0017	-0.0017	1	0.029	0.0049	0.018
amnt_loans	0.2	0.012	-0.075	0.54	0.0017	0.0047	0.029	1	-0.0013	0.56
max_loan__amt	0.0003	0.0011	0.009	0.00045	0.0011	-9e-05	0.0049	-0.0013	1	0.00065
payback_ratio	0.25	0.0093	-0.08	0.25	-0.001	0.005	0.018	0.56	0.00065	1

2.Plot : Plot of Features Before the Normalization.



```
In [41]: X_train.plot()
         plt.xlim(0,200)
         plt.show()
```

3.Plot : After the Features Normalization.



```
In [53]: X_train.plot()
         plt.xlim(0,200)
         plt.show()
```

Algorithms

As I mentioned, I have used various classification models on this dataset and they have different accuracy and other performance measures. I have used the KNN machine learning algorithms on our dataset.

I have used GridSearchCV (Grid search Cross-validation) with Logistic regression.It istechnique for finding the best combination of hyperparameters for your logistic regression model. GridSearchCV systematically searches through a range of hyperparameters you specify and evaluates the model's performance using cross-validation.

Firstly, we have tried GridSearchCV on logistic regression and KNN algorithms. From that after comparison, KNN provided more accurate and efficient model so we used KNN algorithm.

Following is Step by Step Process :

1) Loading preprocessing dataset and split it into features (X) and the target variable (y).
2) Create a Logistic Regression model.

   knn = KNeighborsClassifier()
3) Define a grid of hyperparameters which are related to Your selected model

   k = range(1, 15, 2)
   mat = ["euclidean", "manhattan"]
   w = ["uniform"]
   search_space = dict(n_neighbors = k, metric = mat, weights = w)
4) Create a GridSearchCV object, specifying the model, parameter grid, cross-validation (cv), and scoring metric.

   cv =StratifiedKFold(n_splits=5)
   search=RandomizedSearchCV(estimator=model,
   param_distributions=search_space,n_iter=10,cv=cv,
   scoring="accuracy", n_jobs=-1)
5) Fit the GridSearchCV object to the training data to find the best hyperparameters.

   search_result = search.fit(X_train, y_train)

6) Retrieve the best hyperparameters for selected model.

   search_result.best_params_
7) Create object of KNN using best parameters:

knn_model=KNeighborsClassifier(n_neighbors=13,weights='uniform',metric=

'manhattan')

8)Fit data in model object:

knn_model.fit(X_train, y_train)

9)Predict using model:

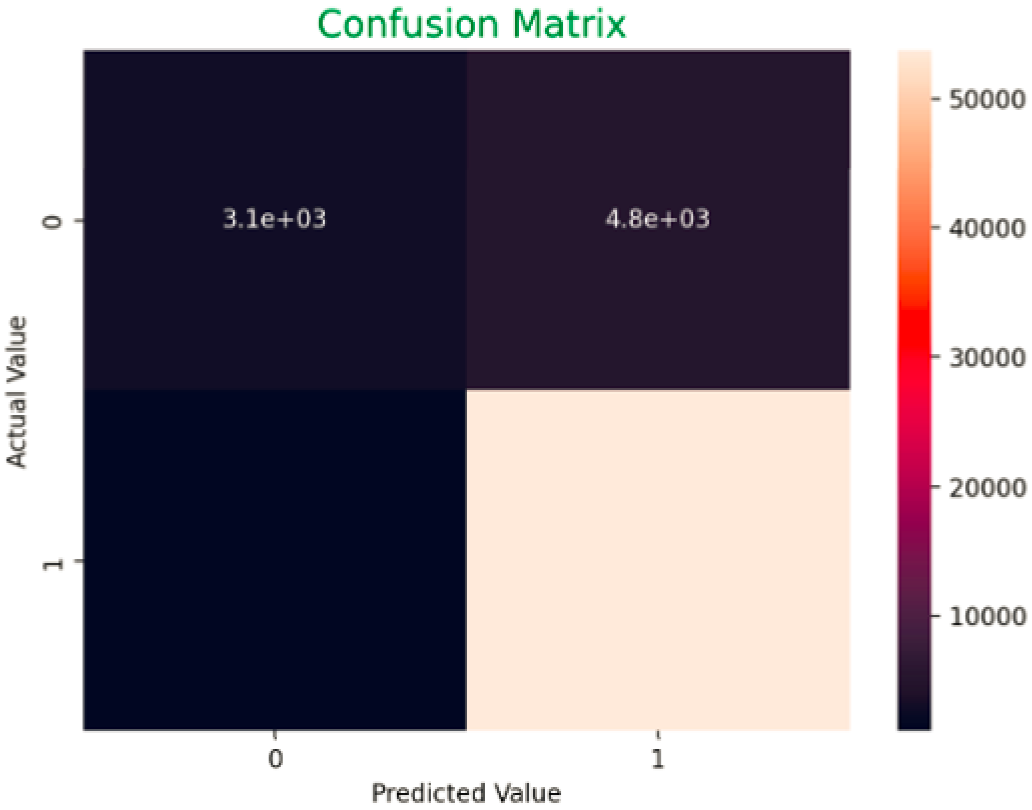
y_pred_knn = knn_model.predict(X_test)

10)Calculate Accuracy score:

accuracy_score(y_test, y_pred_knn)

Evalution

Confusion matrix For KNN Model:



accuracy_score: 0.9056314891309548
recall_score: 0.9800361320462052
precision_score: 0.9175949972662657
f1_score: 0.9477882587556363

Result and discussion

After developing and fine-tuning our Bank Fraud Detection Model using Logistic Regression, we conducted a comprehensive evaluation to assess its performance.
The Performace of our model in Terms of accuracy is nearly 90% .

The Factors Responsible for Perfromance are discussed follow:

Hyperparameter Tuning: The success of our KNN model can be attributed to the diligent hyperparameter tuning performed using GridSearchCV.

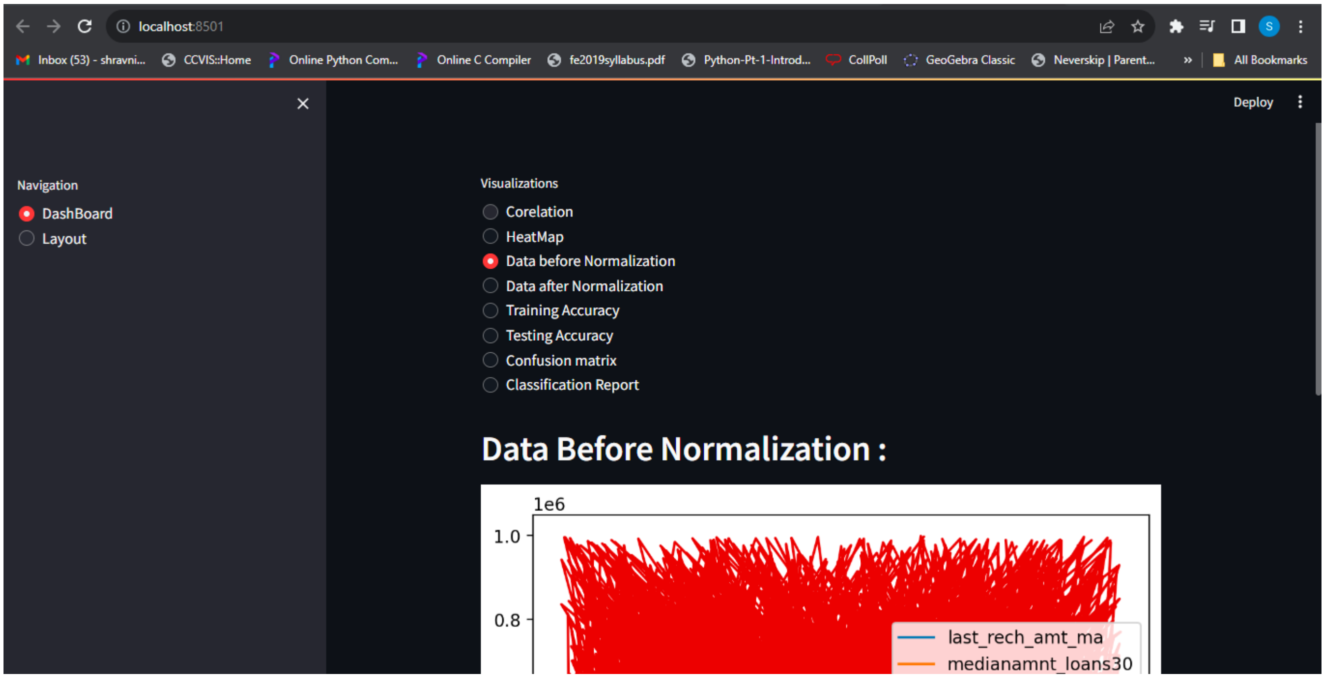
Data Quality: The model's performance is heavily dependent on the quality and representativeness of the training data.

Interpretability: KNN is considered a "black-box" model in the sense that it lacks inherent interpretability because it doesn't produce explicit rules or coefficients like linear models do.

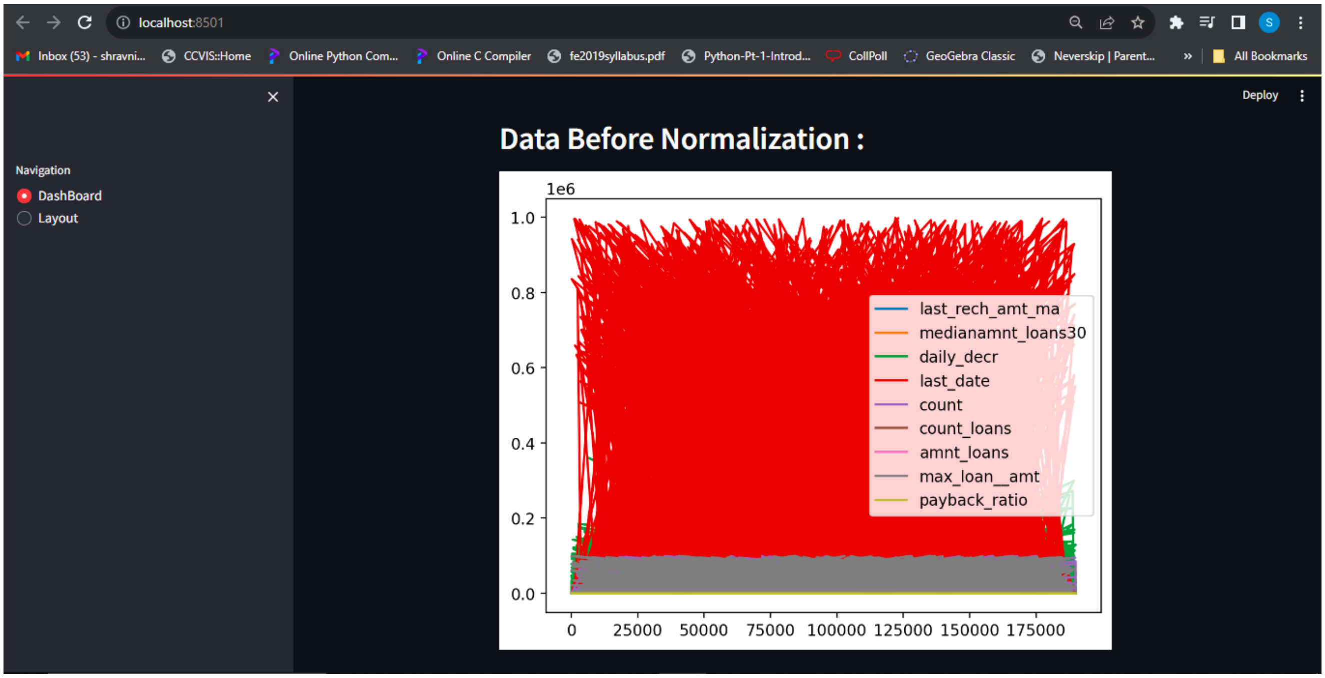
User Interface: A user-friendly interface for bank officers and applicants should be developed to facilitate interaction with the model. This interface should provide explanations for model decisions, fostering transparency and trust.

Model Deployment: To fully realize the model's potential, we recommend deploying it within the lending institution's loan approval system. This would facilitate real-time Bank fraud decision-making, allowing for faster and more consistentoutcomes.

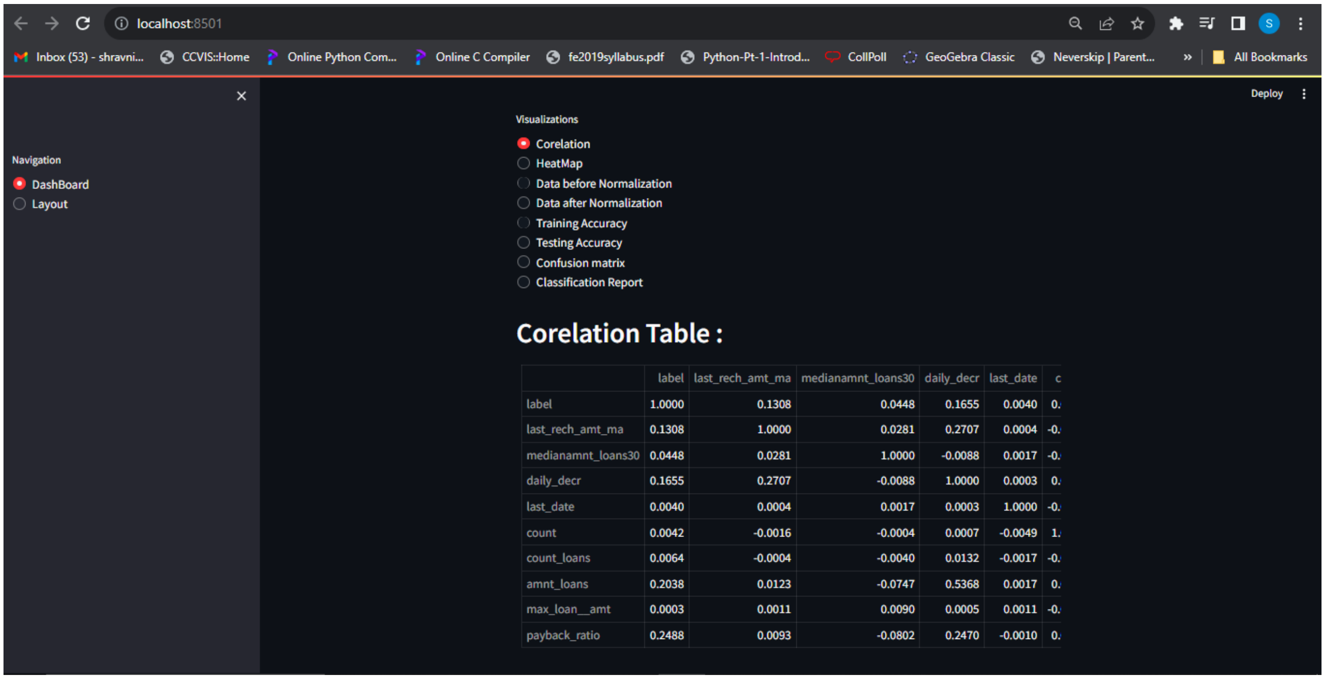
DashBoard Images on Streamlit:



Data Before Normalization :



Data Before Normalization :



Navigation

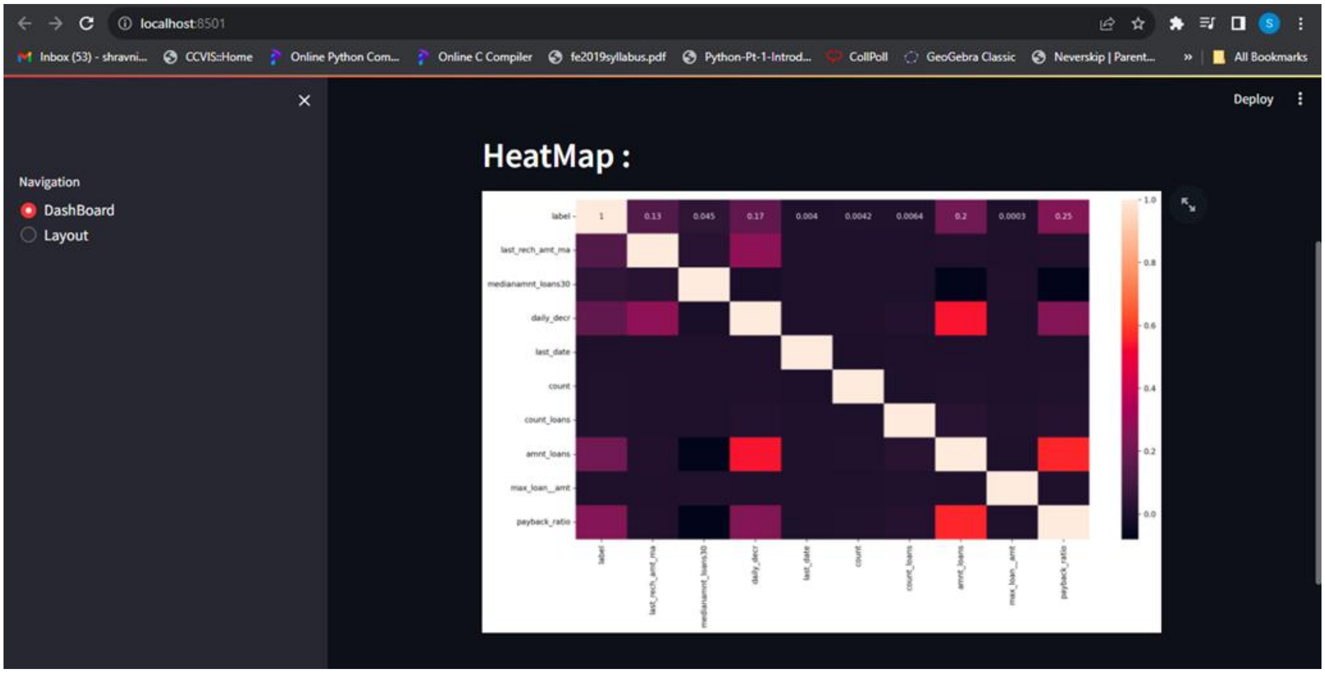
● DashBoard
○ Layout

Visualizations

● Corelation
○ HeatMap
○ Data before Normalization
○ Data after Normalization
○ Training Accuracy
○ Testing Accuracy
○ Confusion matrix
○ Classification Report

Corelation Table :

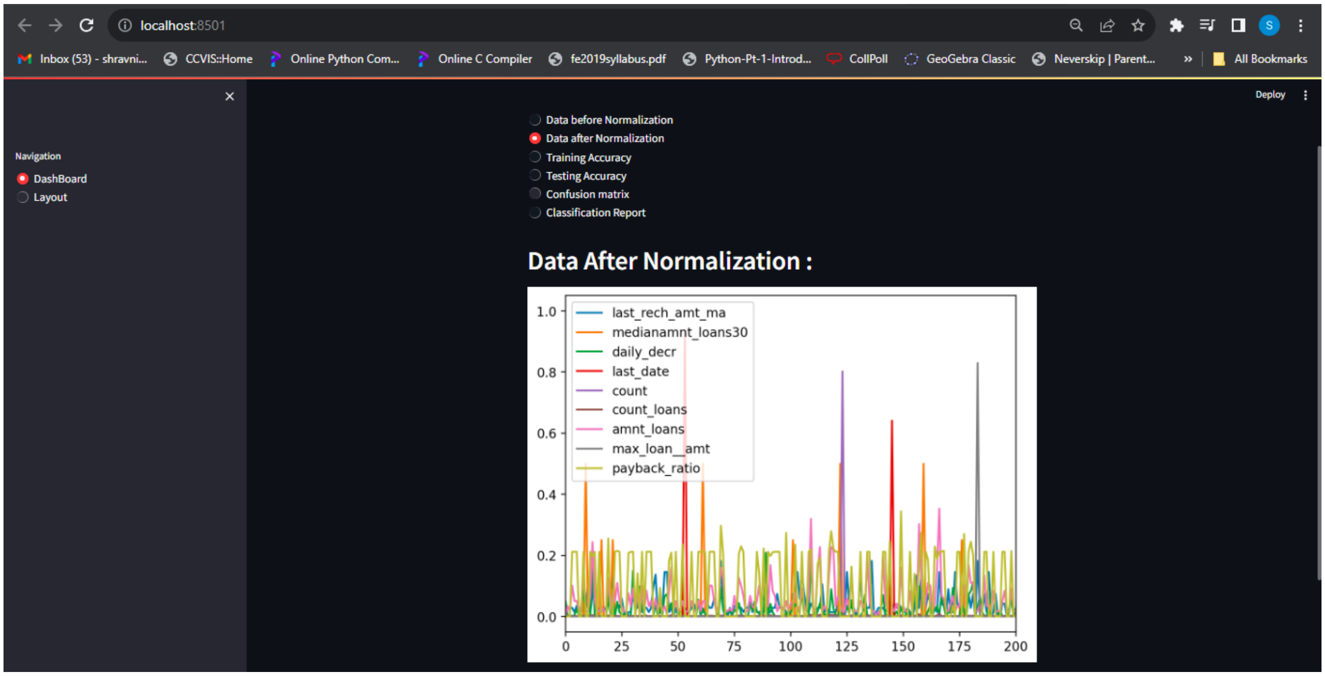
	label	last_rech_amt_ma	medianamnt_loans30	daily_decr	last_date	c
label	1.0000	0.1308	0.0448	0.1655	0.0040	0.
last_rech_amt_ma	0.1308	1.0000	0.0281	0.2707	0.0004	-0.
medianamnt_loans30	0.0448	0.0281	1.0000	-0.0088	0.0017	-0.
daily_decr	0.1655	0.2707	-0.0088	1.0000	0.0003	0.
last_date	0.0040	0.0004	0.0017	0.0003	1.0000	-0.
count	0.0042	-0.0016	-0.0004	0.0007	-0.0049	1.
count_loans	0.0064	-0.0004	-0.0040	0.0132	-0.0017	-0.
amnt_loans	0.2038	0.0123	-0.0747	0.5368	0.0017	0.
max_loan__amt	0.0003	0.0011	0.0090	0.0005	0.0011	-0.
payback_ratio	0.2488	0.0093	-0.0802	0.2470	-0.0010	0.



Navigation

● DashBoard
○ Layout

HeatMap :

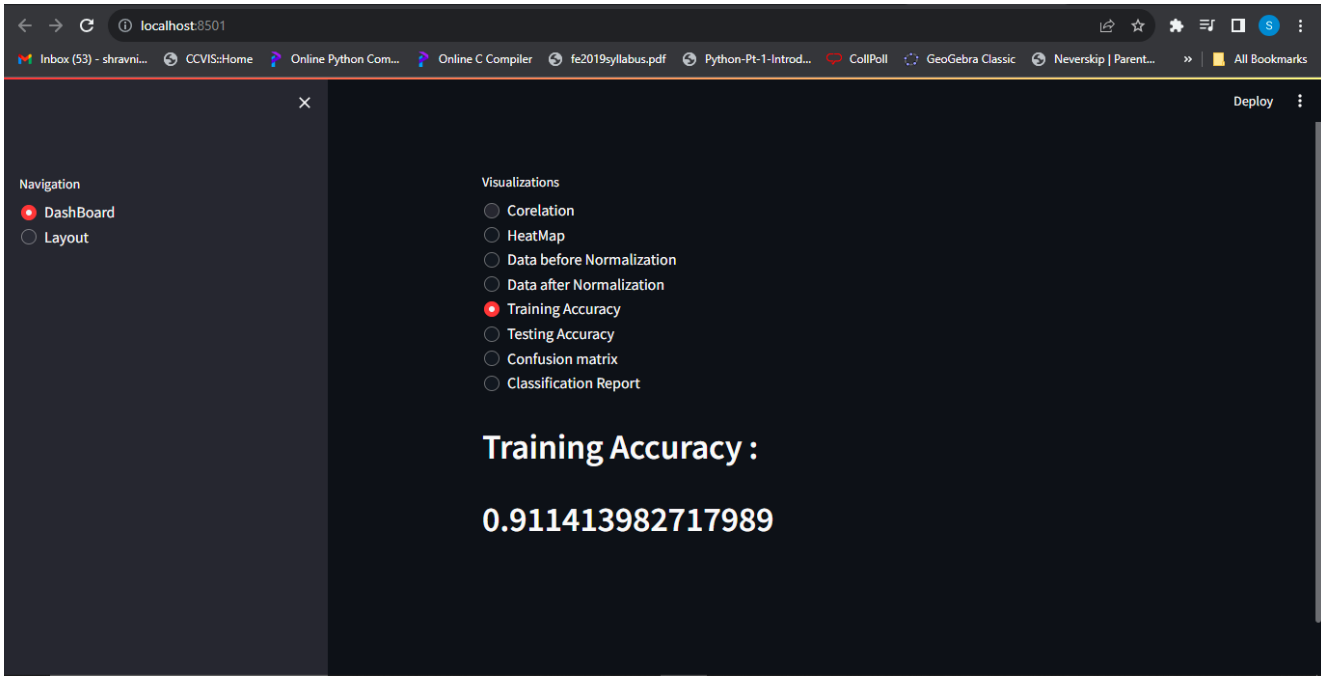


Navigation

● DashBoard
○ Layout

○ Data before Normalization
● Data after Normalization
○ Training Accuracy
○ Testing Accuracy
○ Confusion matrix
○ Classification Report

Data After Normalization :



Navigation

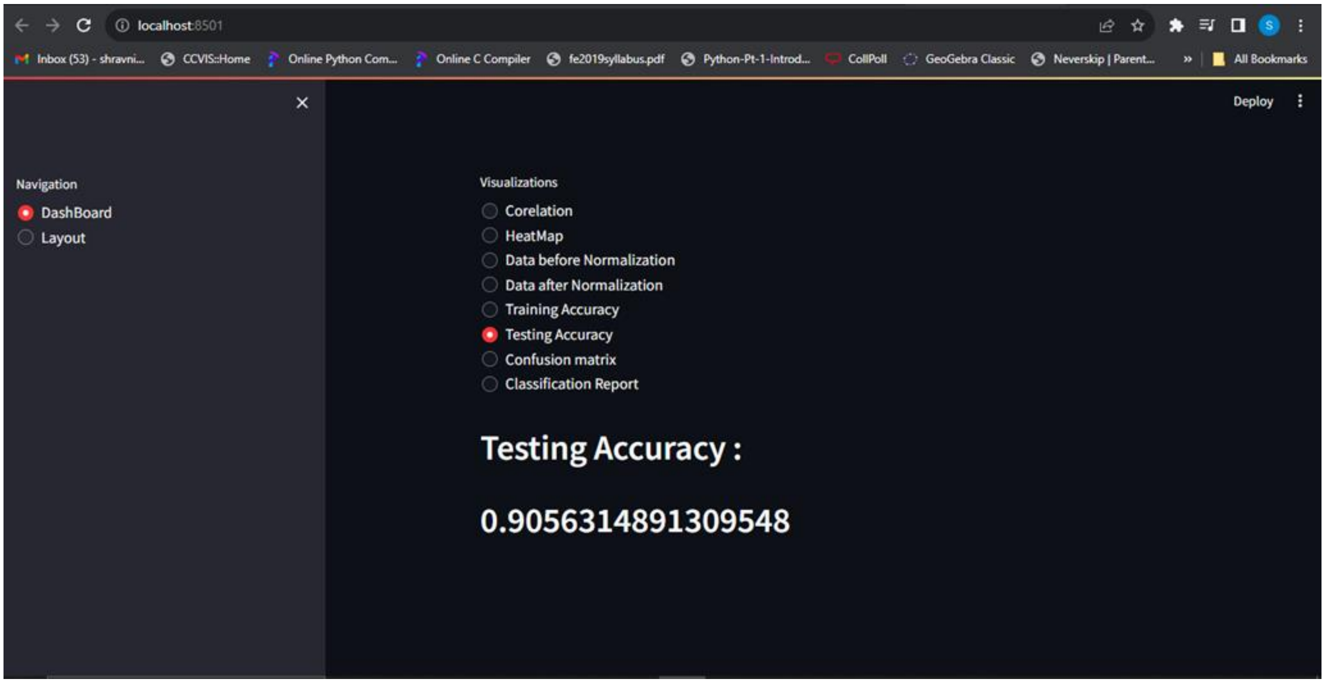
● DashBoard
○ Layout

Visualizations

○ Corelation
○ HeatMap
○ Data before Normalization
○ Data after Normalization
● Training Accuracy
○ Testing Accuracy
○ Confusion matrix
○ Classification Report

Training Accuracy :

0.911413982717989



Navigation

● DashBoard

○ Layout

Visualizations

○ Corelation

○ HeatMap

○ Data before Normalization

○ Data after Normalization

○ Training Accuracy

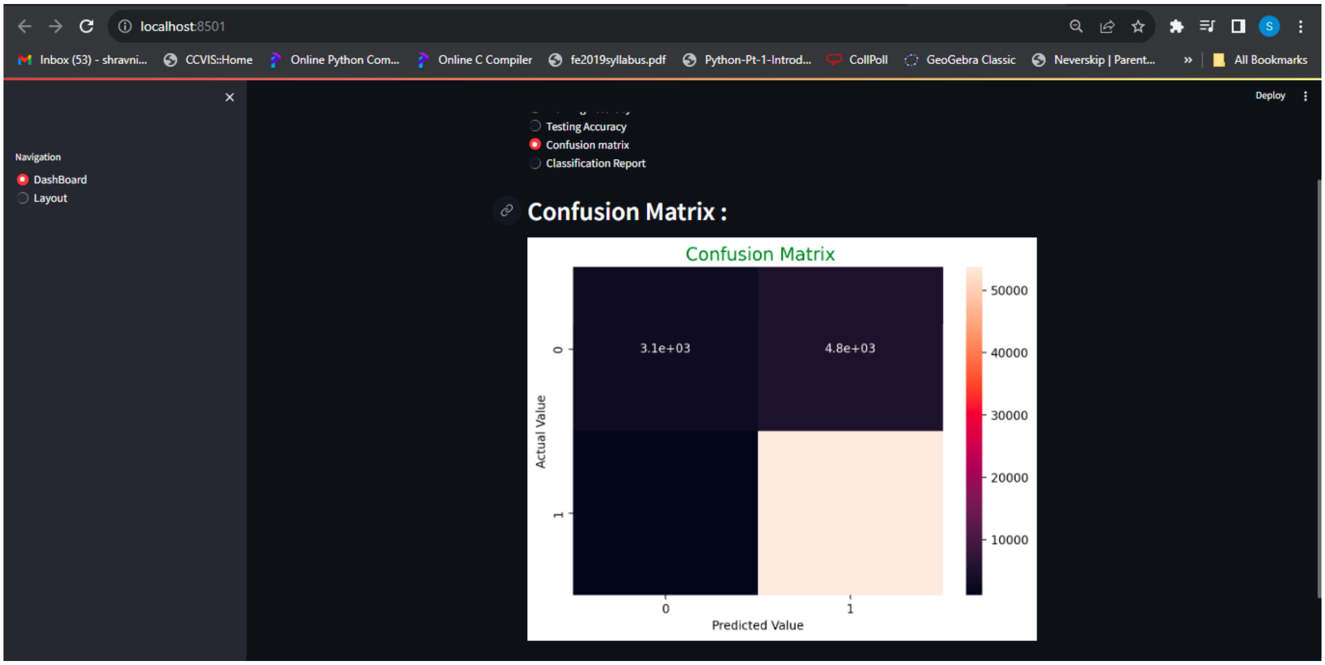
● Testing Accuracy

○ Confusion matrix

○ Classification Report

Testing Accuracy :

0.9056314891309548



Navigation

● DashBoard

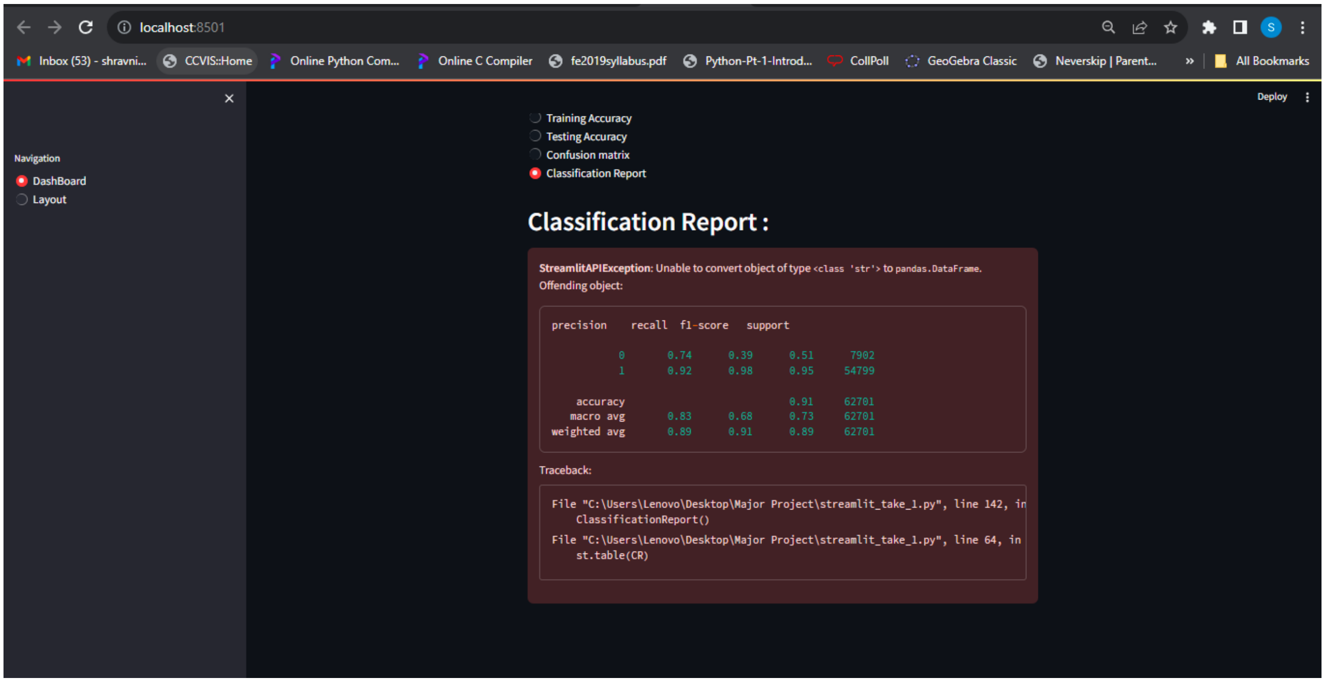
○ Layout

○ Testing Accuracy

● Confusion matrix

○ Classification Report

Confusion Matrix :



Navigation

○ DashBoard
○ Layout

○ Training Accuracy
○ Testing Accuracy
○ Confusion matrix
● Classification Report

Classification Report :

StreamlitAPIException: Unable to convert object of type `<class 'str'>` to `pandas.DataFrame`.
Offending object:

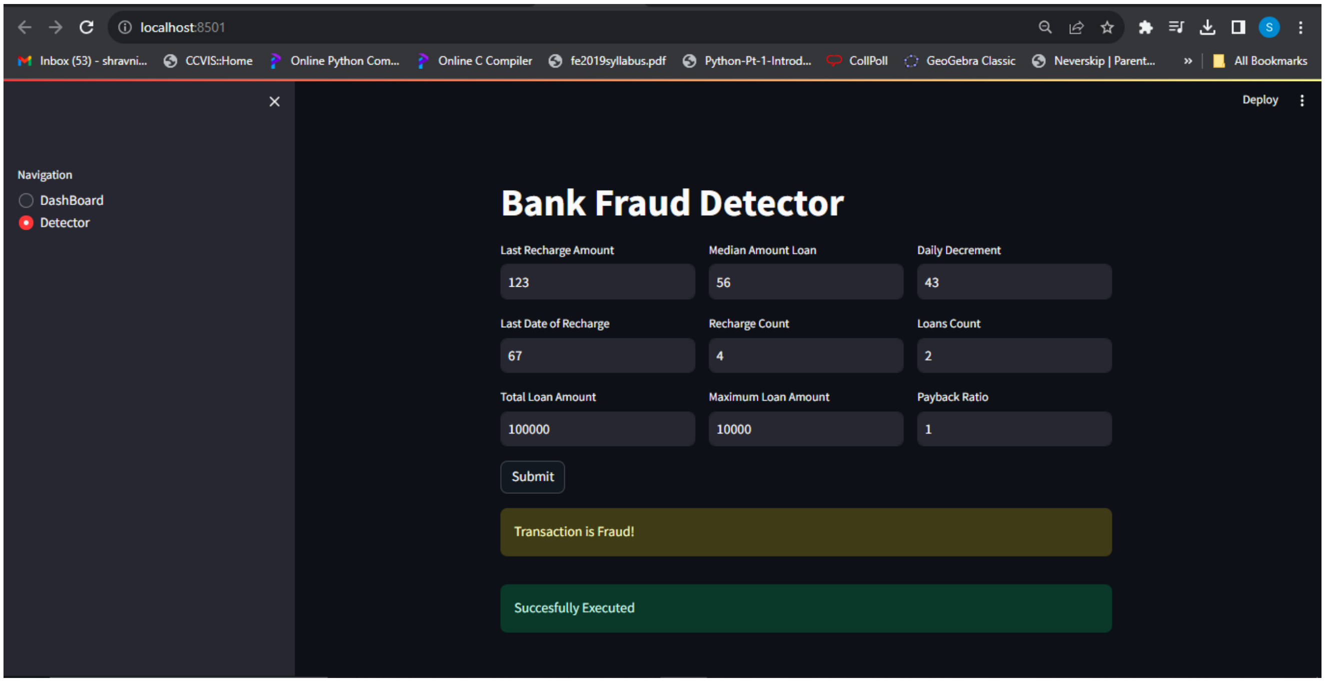
```
precision    recall  f1-score   support

           0       0.74      0.39      0.51      7902
           1       0.92      0.98      0.95     54799

    accuracy                           0.91     62701
   macro avg       0.83      0.68      0.73     62701
weighted avg       0.89      0.91      0.89     62701
```

Traceback:

```
File "C:\Users\Lenovo\Desktop\Major Project\streamlit_take_1.py", line 142, in
    ClassificationReport()
File "C:\Users\Lenovo\Desktop\Major Project\streamlit_take_1.py", line 64, in
    st.table(CR)
```



Navigation

○ DashBoard
● Detector

Bank Fraud Detector

Last Recharge Amount: 123

Median Amount Loan: 56

Daily Decrement: 43

Last Date of Recharge: 67

Recharge Count: 4

Loans Count: 2

Total Loan Amount: 100000

Maximum Loan Amount: 10000

Payback Ratio: 1

Transaction is Fraud!

Succesfully Executed

Conclusion

In conclusion, using K-Nearest Neighbors (KNN) for bank fraud detection is like having a super-smart detective that looks at all the bank transactions. This detective compares each new transaction with the ones it already knows about and decides if it's normal or suspicious. If a transaction looks suspicious because it's too different from the normal ones, the detective raises a red flag. This helps the bank catch bad guys trying to steal money or cheat the system. KNN helps keep your money safe by quickly spotting unusual activities, and it's like having a vigilant guard for your bank account.

Future Work

Future work in the field of bank fraud prediction holds promising avenues for further improvement and innovation. Here are some directions for future work:

1. Advanced Machine Learning Models: Continue exploring and developing more advanced machine learning and deep learning models. Techniques such as neural networks, gradient boosting, and ensemble methods could enhance the accuracy of fraud detection.

2. Real-Time Detection: Improve real-time fraud detection capabilities. Faster processing and analysis of transactions can help in preventing fraud as it happens, reducing potential losses.

3. Explainable AI (XAI): Develop methods to make fraud detection models more interpretable. Explainable AI techniques can help auditors and investigators understand why a particular transaction was flagged as fraudulent.

4. Big Data and Stream Processing: Embrace big data technologies and stream processing to handle the increasing volume, velocity, and variety of financial data. This will enable more efficient and effective fraud detection.

5. Behavioral Biometrics: Explore the use of behavioral biometrics, such as keystroke dynamics, mouse movement patterns, and voice recognition, to enhance user authentication and fraud detection.

6. Graph Analytics: Utilize graph analytics to uncover complex fraud networks and

connections among fraudulent actors. This can be particularly useful in detecting organized fraud rings.

7. Unsupervised Learning: Continue research in unsupervised learning techniques for anomaly detection. Clustering and outlier detection methods can identify unusual patterns without relying on labeled data.

8. Cross-Channel Fraud Detection: Develop strategies for detecting fraud that spans multiple channels, including online, mobile, and physical transactions. A holistic approach to fraud detection can be more effective.

9. Blockchain and Cryptocurrency Monitoring: As cryptocurrencies gain popularity, research and develop fraud detection methods specific to blockchain transactions and crypto-related fraud.

10. AI Ethics and Fairness: Address ethical considerations and fairness in fraud detection. Ensure that the use of AI in fraud prevention does not result in biased or unfair outcomes.

11. Collaboration and Information Sharing: Encourage collaboration and information sharing among financial institutions and regulatory bodies to collectively combat fraud. Sharing threat intelligence can enhance detection capabilities.

12. Regulatory Compliance: Stay updated with evolving regulatory requirements and compliance standards. Adapt fraud detection systems to meet the latest legal and regulatory obligations.

13. User Education: Continue educating customers and users about common fraud tactics and the importance of safeguarding personal and financial information.

The field of bank fraud prediction is dynamic, evolving in response to emerging threats and technological advancements. Future work should strive to enhance accuracy, speed, and fairness while also addressing the ethical and regulatory aspects of fraud detection.

You can also add difficulty faced

We faced following difficulties:

Imbalanced Data: Fraudulent transactions are typically rare compared to legitimate ones, leading to class imbalance. This imbalance can make it challenging to train a model that accurately detects fraud without generating too many false positives.

Feature Engineering: Selecting and engineering relevant features from the available data can be complex. Identifying the right set of features that effectively distinguish between legitimate and fraudulent transactions is crucial.

References

References which I found helpful

1. https://www.geeksforgeeks.org/machine-learning/
2. https://scikit-learn.org/stable/supervised_learning.html
3. https://streamlit.io/
4. https://seaborn.pydata.org/