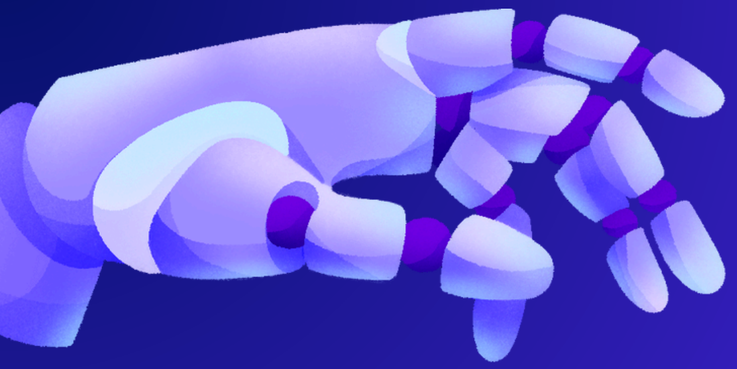
Project

Fraud Detection



The Business Idea 💡

- The goal is to predict whether a financial transaction is fraudulent or not.
- This is crucial because fraudulent transactions can lead to significant financial losses for both the company and the customers.
- By identifying these transactions early, we can prevent further harm.
- The project involves building a model to classify transactions as either fraudulent or legitimate, based on features such as transaction details and user information.

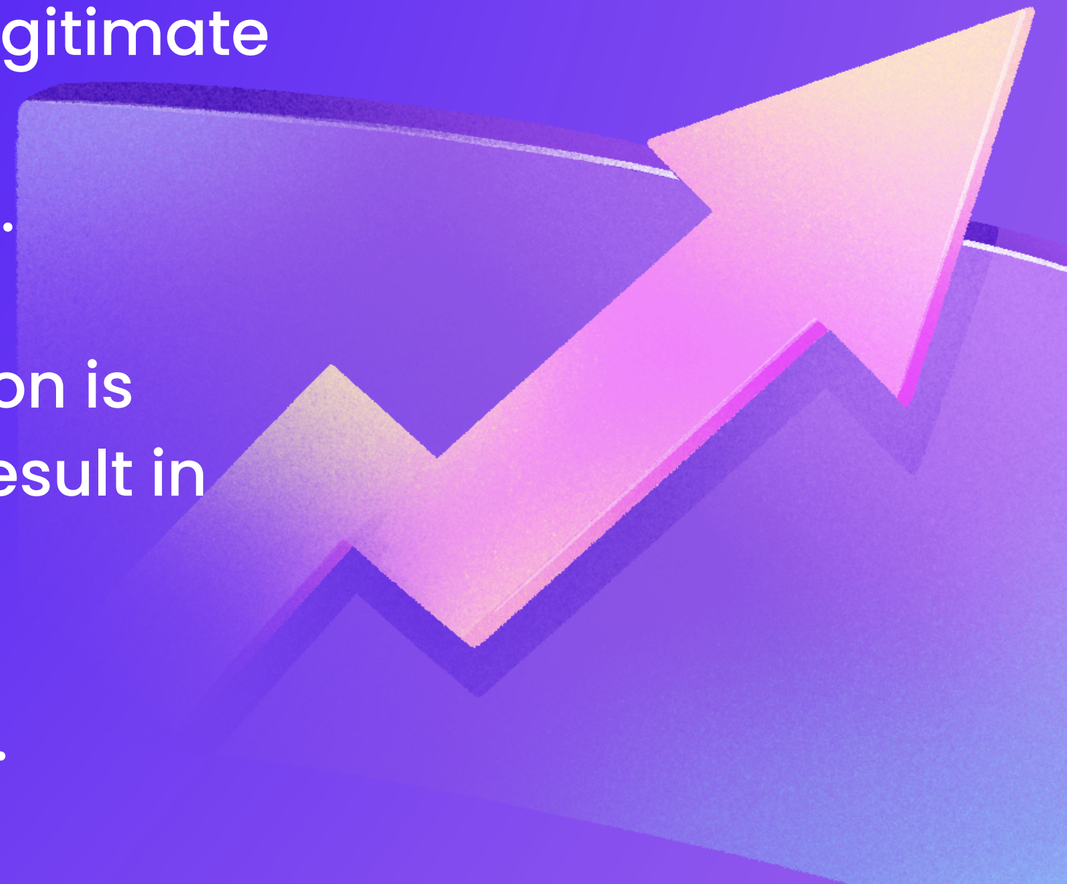
Objective

•The model's objective is to detect whether a transaction is fraudulent or not.

•This is important because if we classify a legitimate transaction as fraudulent, it may cause inconvenience or frustration to the customer.

•On the other hand, if a fraudulent transaction is incorrectly classified as legitimate, it could result in financial losses for the company.

• The goal is to minimize both types of errors. (False positive & False negative)



Data description

•The dataset consists of two main parts:

1-Transaction data: Contains information related to individual transactions such as TransactionID, TransactionAmt, TransactionDT, and others.

2-Identity data: Contains information about the user involved in the transaction, including id_01 to id_38, as well as DeviceType, DeviceInfo, etc.

Key features in the dataset include:

*TransactionID: Unique identifier for each transaction.

*TransactionAmt: Amount of the transaction.

*DeviceType, DeviceInfo: Information about the device used for the transaction.

*id_01 to id_38: These represent various identifiers associated with the user's account or device.

- The target variable is a binary classification:

isFraud: Indicates whether the transaction is fraudulent (1) or not (0).

Now Let's See The Project EDA👀

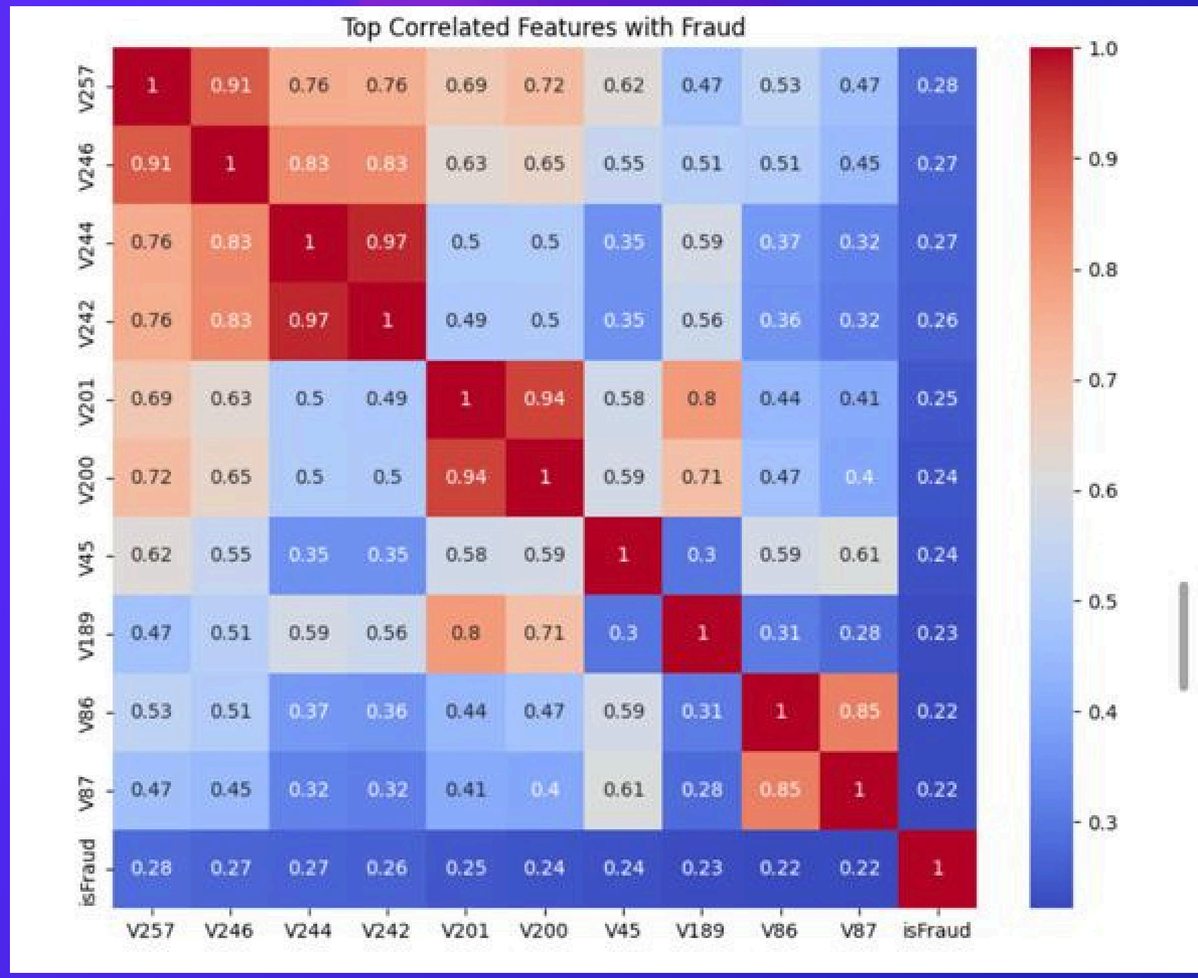
The heatmap shows the top features that are most correlated with fraudulent transactions (isFraud).

Although the correlation values are not very high (maximum is 0.28), they are still useful for feature selection in fraud detection.

Features V257, V246, V244, and V242 show the strongest positive correlation with isFraud, and may be considered important predictors.

The model may benefit from including these features during training, but other methods like feature importance from tree models or SHAP values could provide deeper insight.

No features show a strong negative correlation with isFraud in this subset

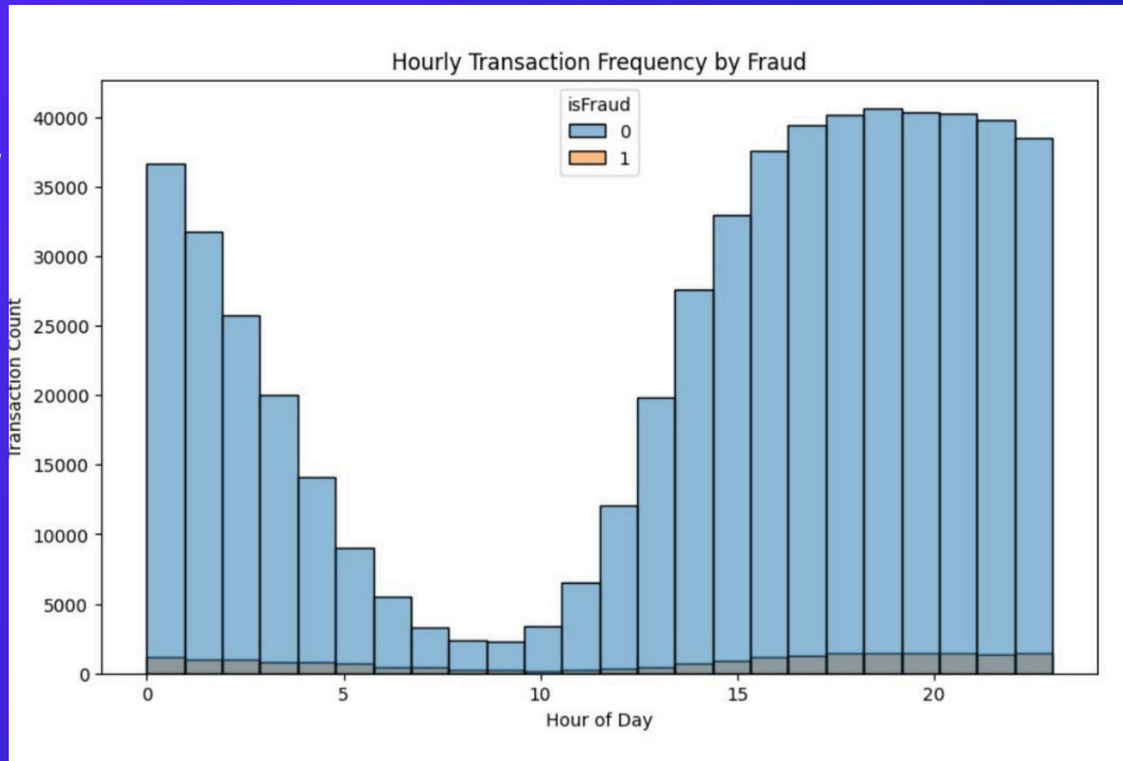


Most fraudulent transactions happen between 0–4 AM. Although the total transaction volume is lower during this time, the ratio of fraud is higher.

During working hours (12 PM to 10 PM), transaction volume is highest, but the fraud rate appears relatively lower.

This pattern may suggest fraudsters are more active late at night, possibly to avoid detection or during off-hours when monitoring may be weaker.

Including the hour of transaction as a feature in the model could help identify fraud patterns based on time of day.

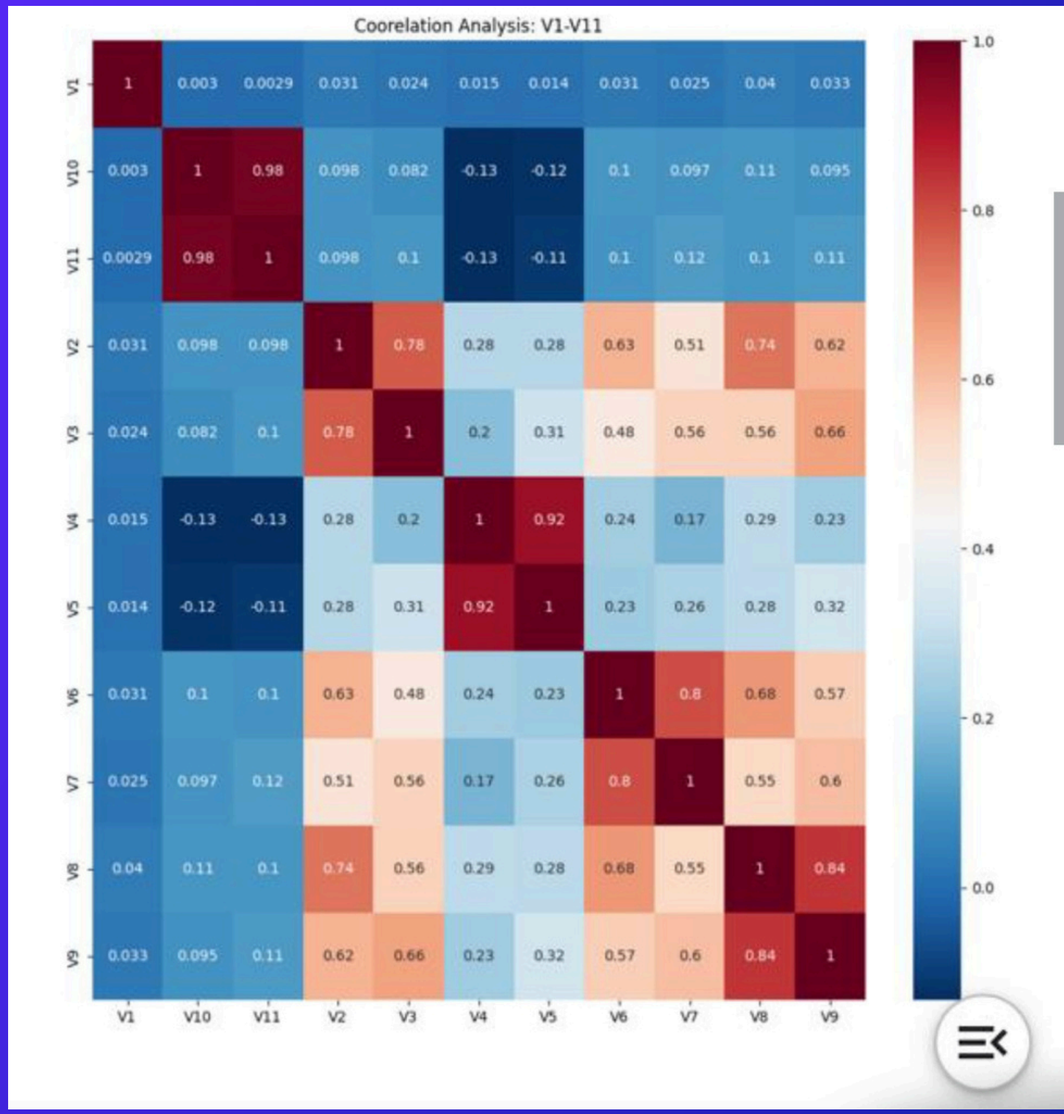


•Several variables show strong positive correlations, such as:

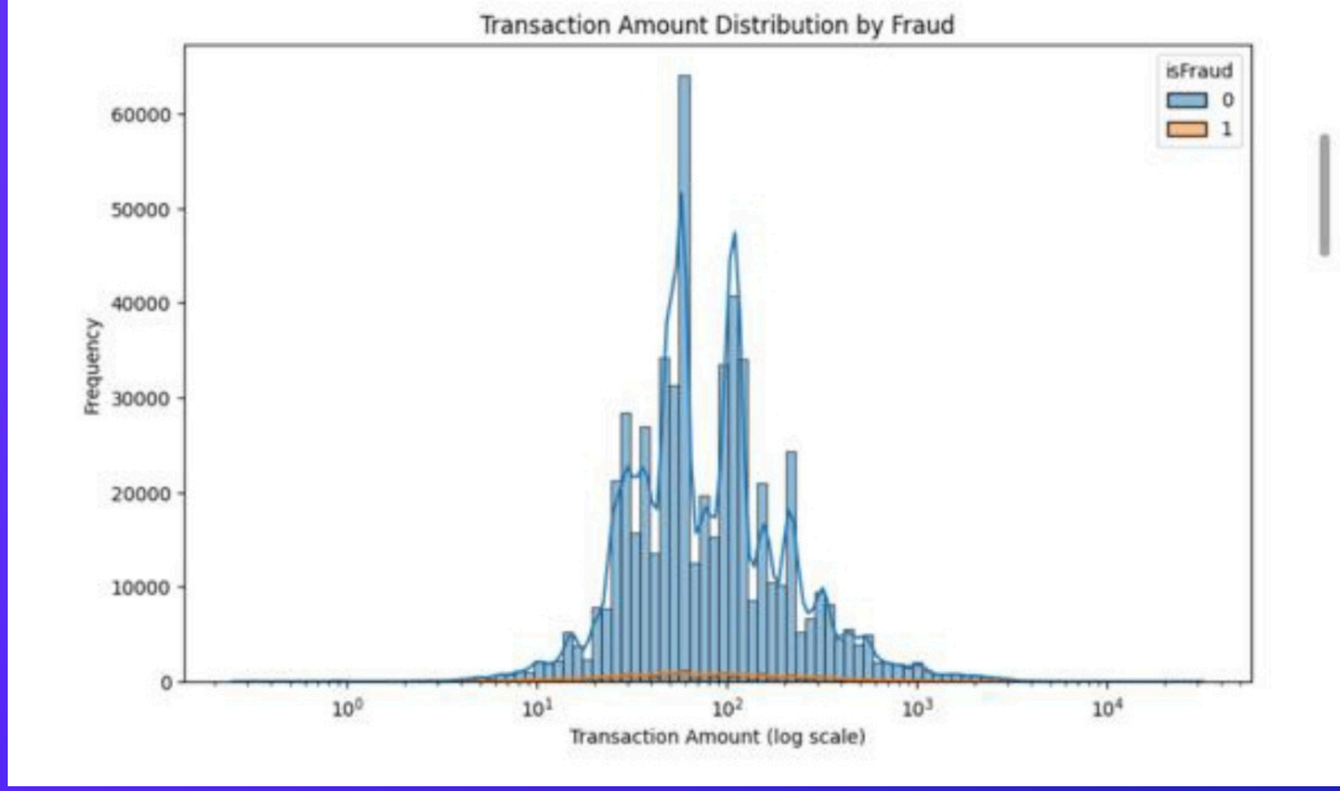
V4 and V5 (0.92)
V9 and V6 (0.84)
V2 and V3 (0.78)

•Variables like V1, V10, and V11 have very low correlation with other features.

•Some variables show slight negative correlation (e.g., V10 with V4, V5, V6), but the relationships are weak and likely not impactful.



The chart shows that fraudulent transactions (orange) mostly happen in the same range of amounts as normal transactions (blue). However, there are fewer fraudulent transactions overall.



Fraud isn’t limited to large amounts—it also occurs with small and medium transactions. So, relying only on the transaction amount to detect fraud won’t be enough. To improve fraud detection, we need to include other features like time, location, or transaction type in the analysis

The dataset exhibits a notable proportion of missing values (41.07%), primarily within card-related features (card2, card3, card4, and card5). Fortunately, essential fields such as TransactionID, isFraud, and key transaction details are complete, ensuring the dataset retains its core integrity for modeling. To mitigate potential performance issues, it is crucial to implement robust missing data handling strategies, including imputation, encoding missingness as a feature, or excluding non-informative attributes.

```
TransactionID         0
isFraud               0
TransactionDT         0
TransactionAmt        0
ProductCD             0
card1                 0
card2              8933
card3              1565
card4              1577
card5              4259
dtype: int64
% of missing data =  41.07343120011112
```

Data Issues

•The dataset presented the following issues during EDA:

•Missing values: Several identity columns (e.g., id_01 to id_38) had missing or constant values.

•Class imbalance: The number of non-fraudulent transactions significantly outnumbered the fraudulent ones, which could bias the model toward predicting the majority class.

•Feature types: Some features were categorical and required encoding (e.g., DeviceType, DeviceInfo), while others were numerical and required scaling or transformation.

Data preprocessing

To handle the data issues, the following preprocessing steps were implemented:

*Handling missing values: Missing values in identity columns were filled with a placeholder value to prevent errors during model training.

*Label encoding: Categorical features (such as DeviceType, DeviceInfo) were encoded using label encoding to convert them into numeric values.

*Feature selection: Only the relevant features (e.g., TransactionID, TransactionAmt, id_01 to id_38) were retained for training.

*Feature scaling: Some numerical features (e.g., TransactionAmt) were scaled to ensure they had similar ranges.

*Handling class imbalance: The models were trained with a balanced class weight (class_weight='balanced') to account for the class imbalance.

Modeling

•Two models were used for fraud detection:

1-Random Forest Classifier:

*A Random Forest model was trained with the features and target variable.

*The model's class weights were adjusted to handle the imbalance in the dataset.

*Hyperparameters: Default settings were used initially .

2-LightGBM:

*LightGBM (Light Gradient Boosting Machine) was also used as an alternative to Random Forest.

*It is known for its efficiency with large datasets and better handling of categorical features.

*Similar to Random Forest, class weights were set to handle the imbalance in the data.

After training both models, their output probabilities were averaged to create a final prediction. This combined approach leverages the strengths of both models.

Evaluation & Results

- Model performance was evaluated using:

1-AUC-ROC curve: To assess the model's ability to distinguish between fraudulent and non-fraudulent transactions.

```
Random Forest ROC AUC: 0.932850645439
```

```
LightGBM ROC AUC: 0.931518243943953 4
```

The AUC values for both Random Forest (0.932) and LightGBM (0.931) are significantly high, indicating that both models are performing excellently in distinguishing between fraudulent and non-fraudulent transactions.

2-F1 Score: Used to evaluate the model's performance by balancing precision and recall.

```
Random Forest F1 Score: 0.57335083688
```

```
LightGBM F1 Score: 0.344228094575799 7
```

Random Forest with an F1 score of 0.57, this model shows a moderate performance in balancing both precision and recall. It performs better in identifying fraudulent transactions compared to LightGBM.

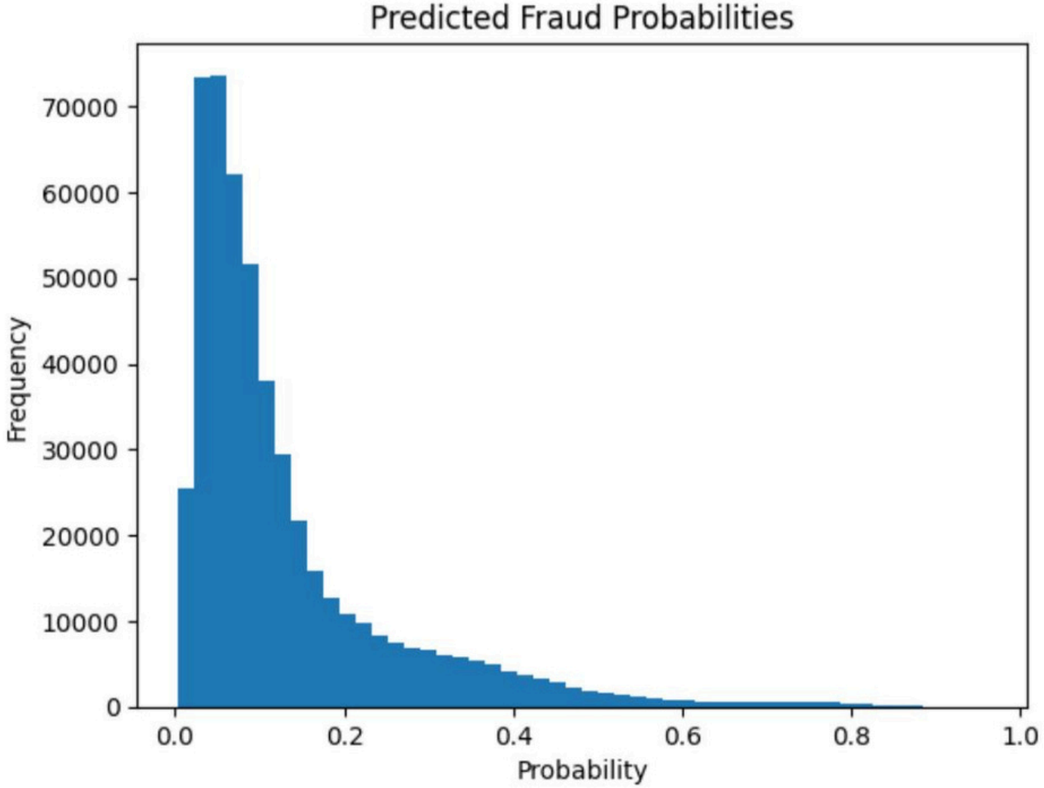
•Results

- •The predicted probabilities for fraudulence were calculated and submitted for evaluation.
- •Sample results for the first few transactions:

TransactionID	isFraud
3663549	0.054905
3663550	0.087531
3663551	0.101016

- A histogram of the predicted probabilities was plotted to understand the distribution of fraud likelihood.

The histogram shows that the majority of transactions are predicted with low probabilities of fraud (close to 0), reflecting the class imbalance in the dataset. A small number of transactions have higher predicted probabilities (closer to 1), indicating that the model is identifying potential frauds effectively.



Conclusion

- The model was able to predict the likelihood of fraudulent transactions with reasonable accuracy

- The combined approach of using both Random Forest and LightGBM improved the overall performance.

Resources

https://www.kaggle.com/competitions/ieee-fraud-detection/discussion/111308

https://www.kaggle.com/code/vigneshbalachander/fraud-detection-eda-preprossing-competation

THANK YOU!