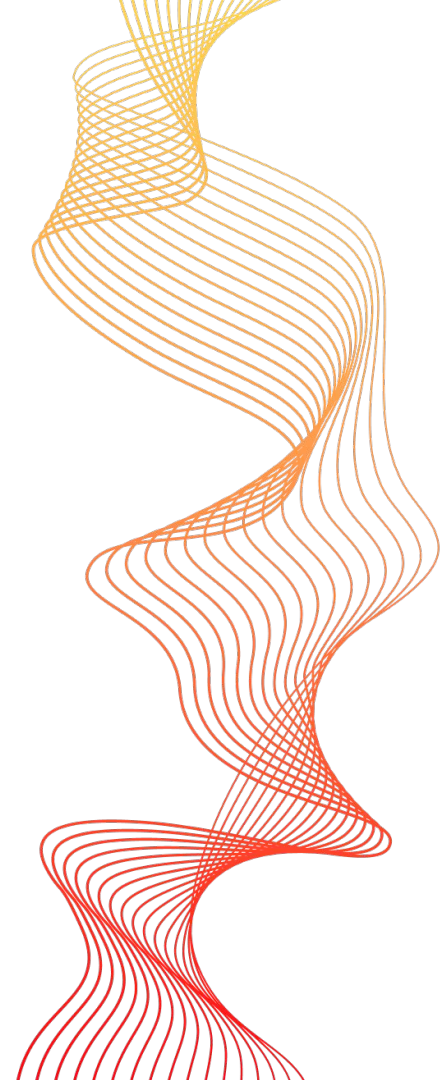




Bank Account Fraud

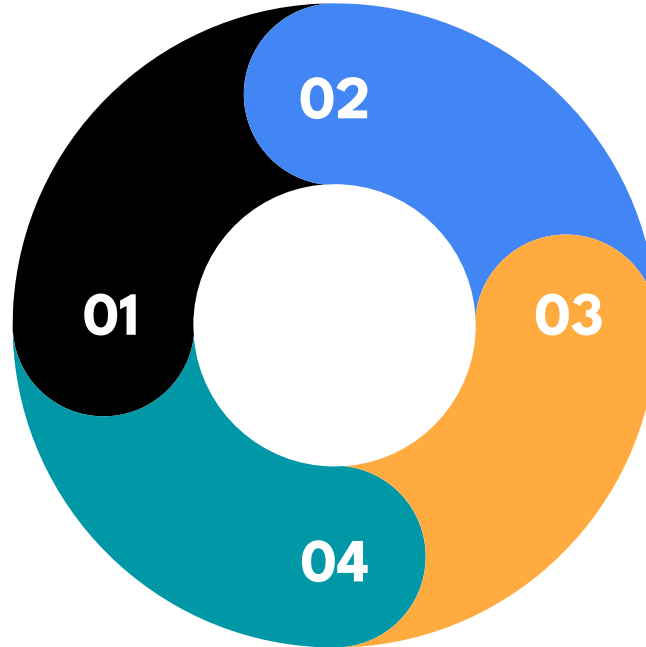
In this presentation, we will explore the problem of bank account fraud and propose a predictive modeling solution to accurately detect fraudulent applications.



Problem Definition

Large consumer banks face the challenge of detecting and preventing bank account fraud.

Fraudsters use identity theft or fictional identities to gain access to banking services.



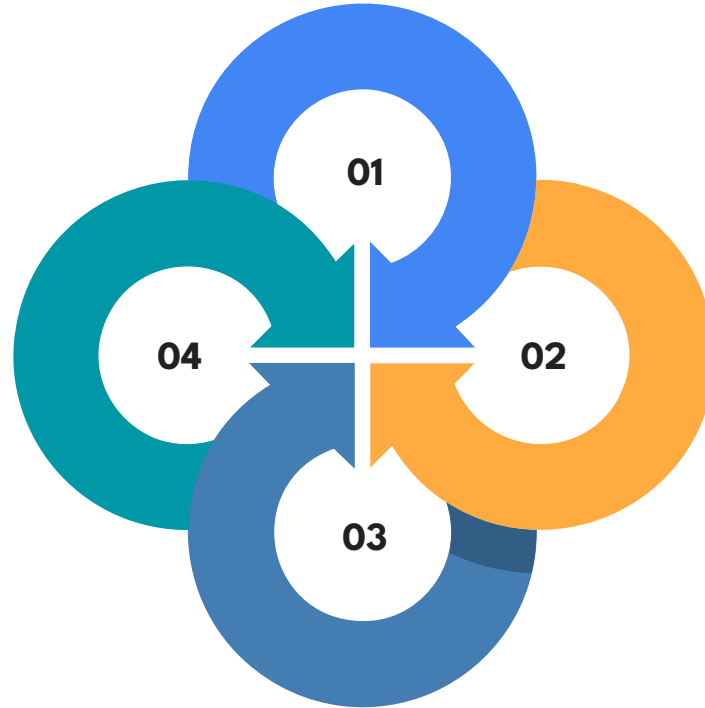
The bank sustains all costs when fraudsters successfully open fraudulent accounts.

The goal is to develop a predictive modeling solution to detect fraudulent applications.

Data Understanding

The dataset is imbalanced, with fraud cases accounting for 18.8% of the total.

Both `credit_utilization_ratio` and `transaction_amount_ratio` are highly related to income.

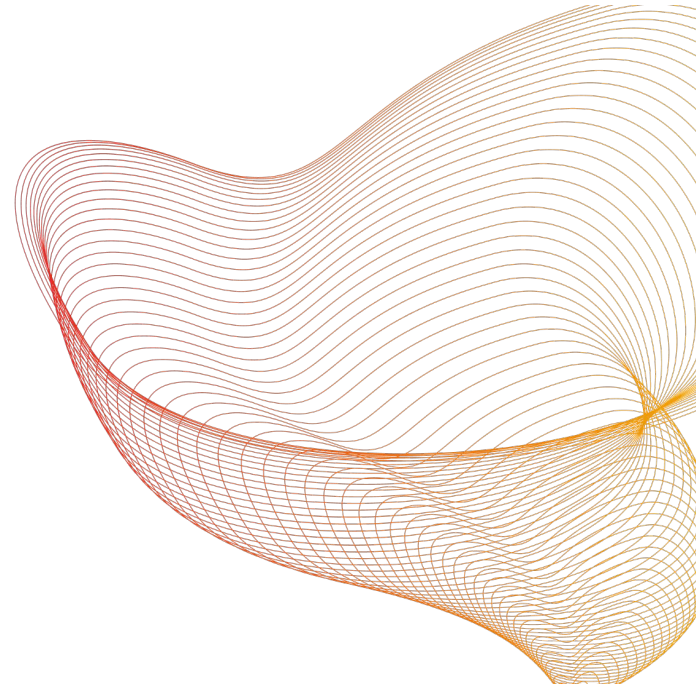


There are missing values in several columns.

Some features have negative or seemingly erroneous values (e.g., negative customer age).


Data Understanding

- 01** There are higher ratio of frauds in some months mostly in the initial ones.
- 02** Windows has more occurrences of fraud than other operating systems.
- 03** In housing_status 'BA' is the type of housing with the most amount of frauds





Data Preparation - Step 1

- Substitute "-1" for NaN.
 - Rows with 50% or more "NaN" fields will be removed.
 - Columns with 70% or more "NaN" fields will be deleted.
 - Replace all missing values with the mode for the respective column
- 

Data Preparation - Step 2

- 01** Remove all redundant and unnecessary features, such as `device_fraud_count`, `prev_address_months_count`, and `id`.
- 02** Normalized data by using a Standard Scaler, to ensure that all features contributed equally to the model.
- 03** Applied One Hot Encoding to all categorical features.
- 04** Used Repeated Stratified K Fold, to split the data into folds.

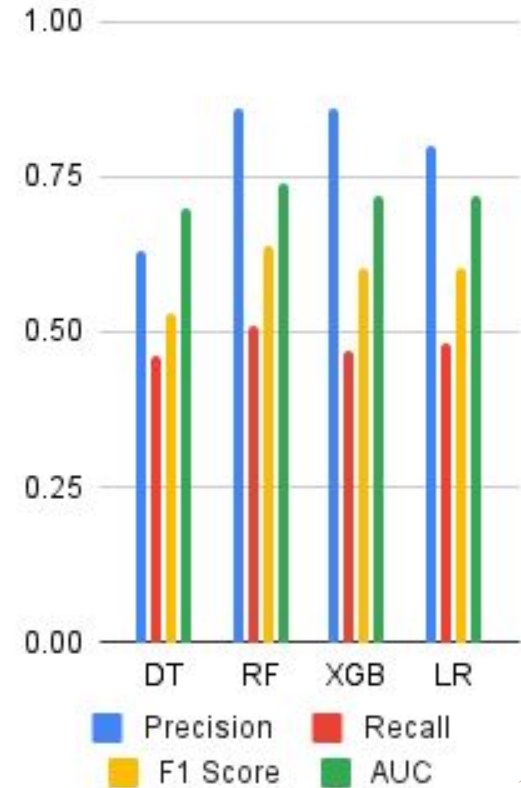


Predictive Modelling - Testing algorithms

Throughout my analysis, I tested four machine learning algorithms:

Random Forest, Logistic Regression, Decision Tree, and XGBoost.

The goal was clear: to identify the most effective algorithm that aligns itself with the data characteristics.





Predictive Modelling - Random Forest

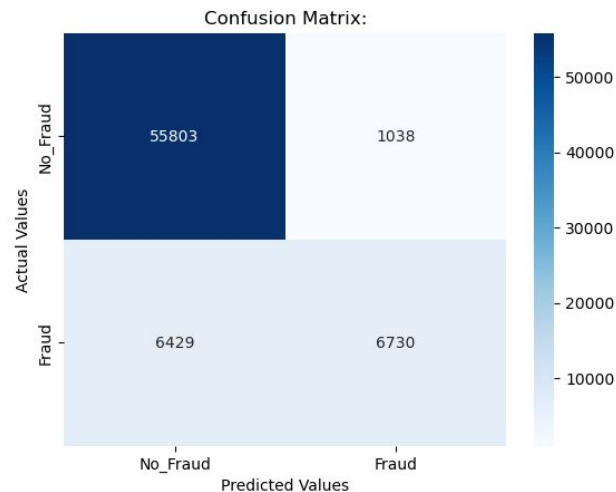
Scoring Metrics

```
{  
  "Recall": "recall",  
  "BACC": "balanced_accuracy",  
  "F1": "f1",  
  "ROC-AUC": "roc_auc"  
}
```

Hyperparameters tuning with
Grid Search CV

```
{  
  'class_weight': {0: 1, 1: 4},  
  'criterion': 'entropy',  
  'max_depth': 10,  
  'max_features': 'sqrt',  
  'min_samples_leaf': 2,  
  'min_samples_split': 5,  
  'n_estimators': 150  
}
```

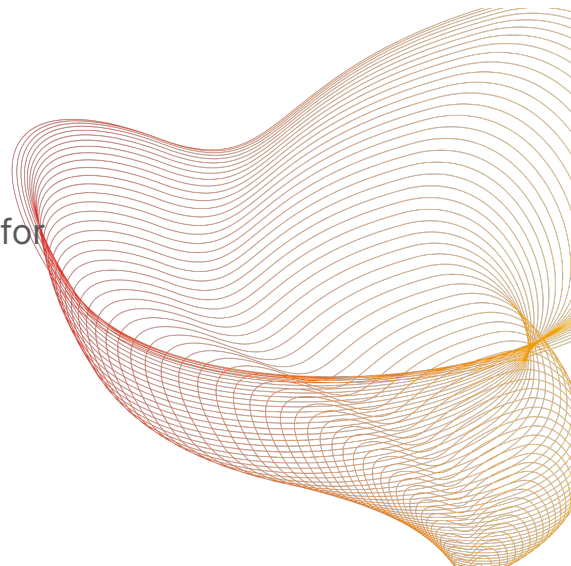
Results achieved





Conclusions

- Despite the imbalance in the dataset provided, i still achieved results that surpassed random outcomes through thorough processes of data comprehension and preparation.
- Looking ahead, the integration of additional entries on the dataset, combined with a more extensive data analysis, could pave the way for enhanced model results in the future.





Thank you. Please feel free to ask any questions.