Predictive classification model for identifying fraudulent credit card transactions

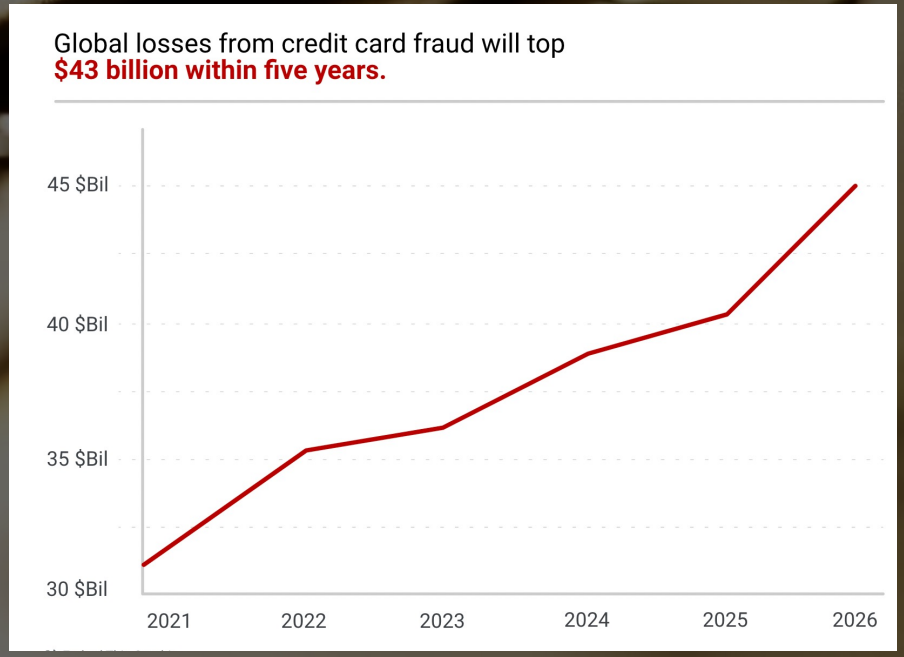
Himani Kaushik

What is credit card fraud?

- Credit card fraud is when someone who is not authorized to use your credit card or account information makes purchases you didn’t authorize.
  - Skimming
  - Manipulation of genuine cards
  - Creation of counterfeit cards
  - Stolen/lost credit cards
  - Fraudulent telemarketing

And why is it a problem?

Direct losses by merchants and banks exceeded $28.58 billion globally in 2020.



source: Nilson Report

Need

- **CreditCards** needs to predict fraudulent credit card transactions to:
  - Prevent
    - Financial loss
    - Loss of credibility and trust
  - Monitor and implement fraud prevention mechanism
  - Reduce chargebacks and fees
  - Save time on lengthy manual reviews
  - Decrease denials of legitimate transactions

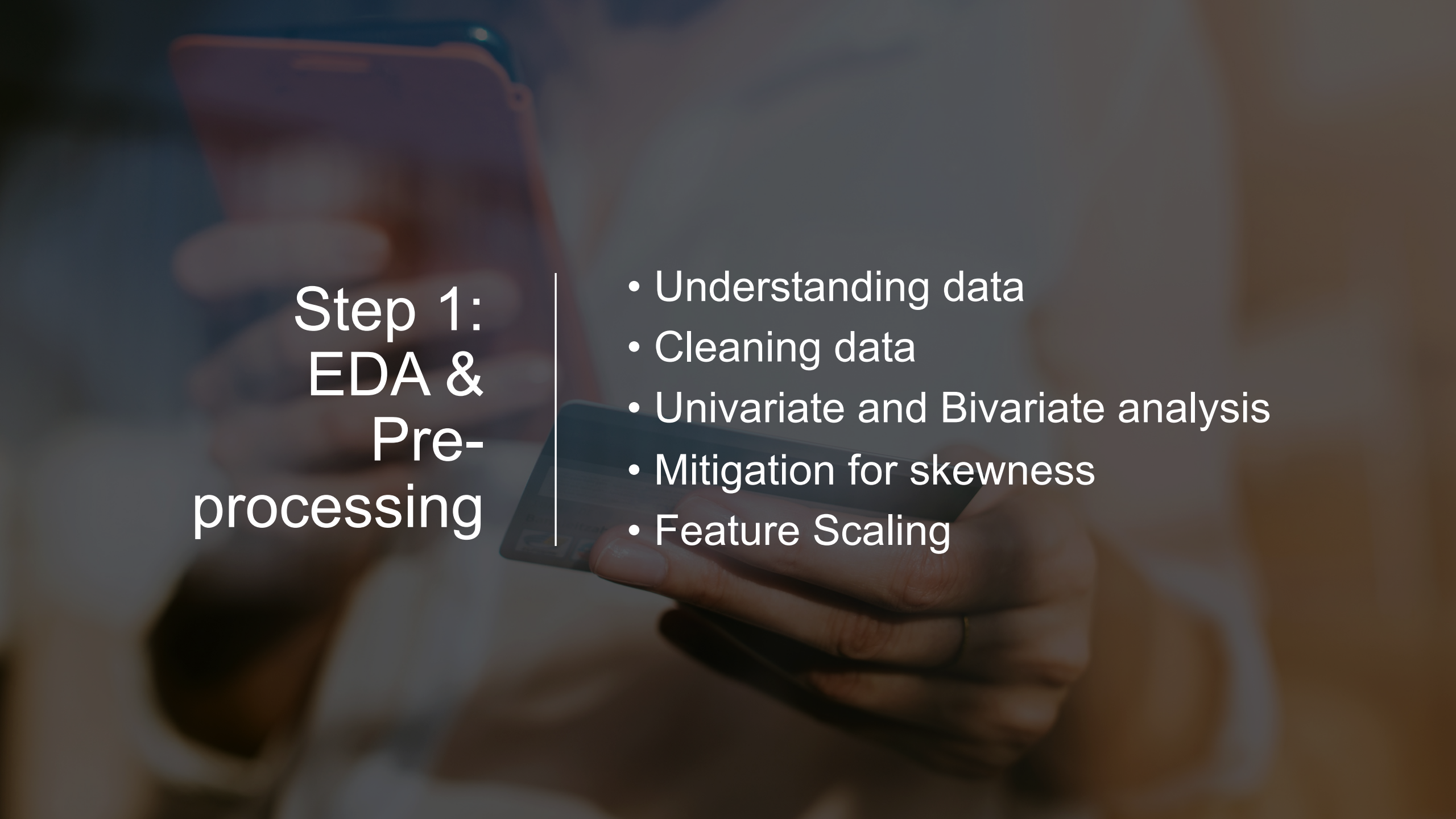
Data

- Credit card transactions by European cardholders in September 2013
- 284,807 transactions, out of which 492 are frauds
- Individual sample includes time, amount, class and other numerical input variables
- Target Variable – Class, indicates whether the transaction is fraudulent or not
  - Highly unbalanced and positive class (frauds) account for 0.172%

Tools

- Pandas and NumPy:
  - Preprocessing and EDA
- Scikit-learn:
  - Classification models
  - Cross validation
  - Hyperparameter Tuning
  - Feature Engineering
  - Class Imbalance
  - Metrics
- Matplotlib and Seaborn:
  - Visualization

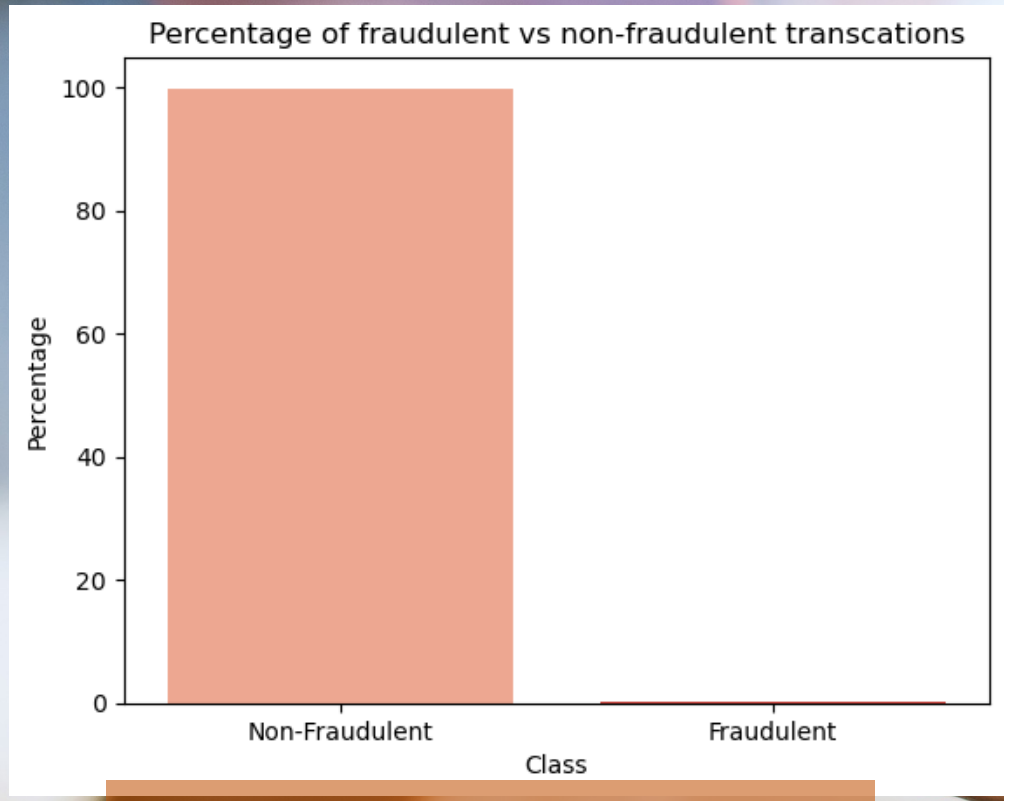
Methodology



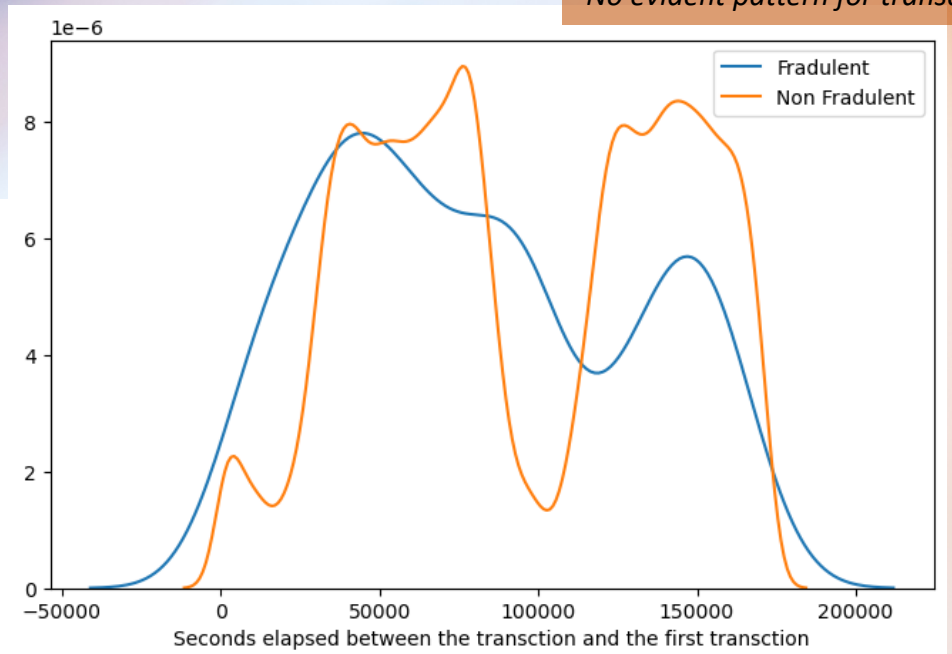
Step 1: EDA & Pre-processing

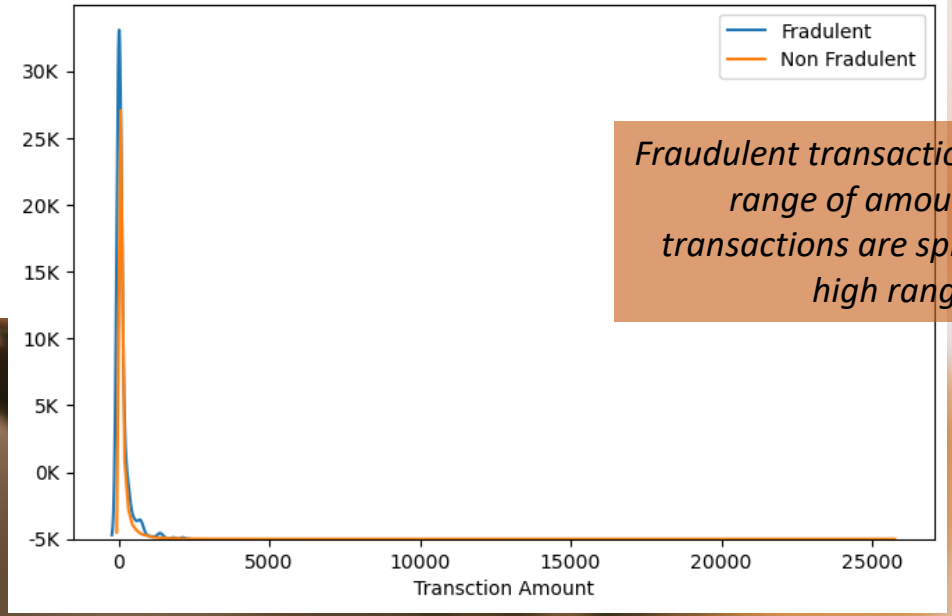
- Understanding data
- Cleaning data
- Univariate and Bivariate analysis
- Mitigation for skewness
- Feature Scaling

No evident pattern for transactions with respect to time



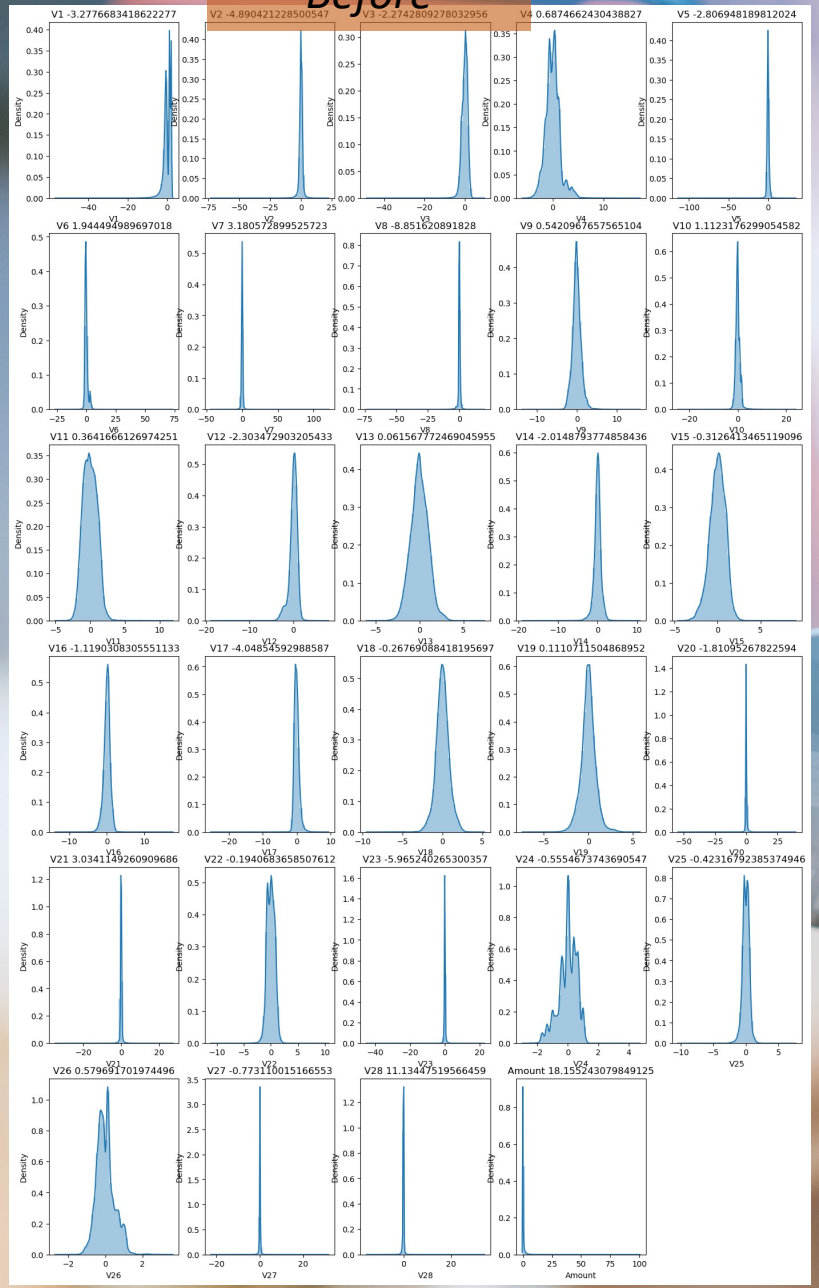
Only 0.17% frauds





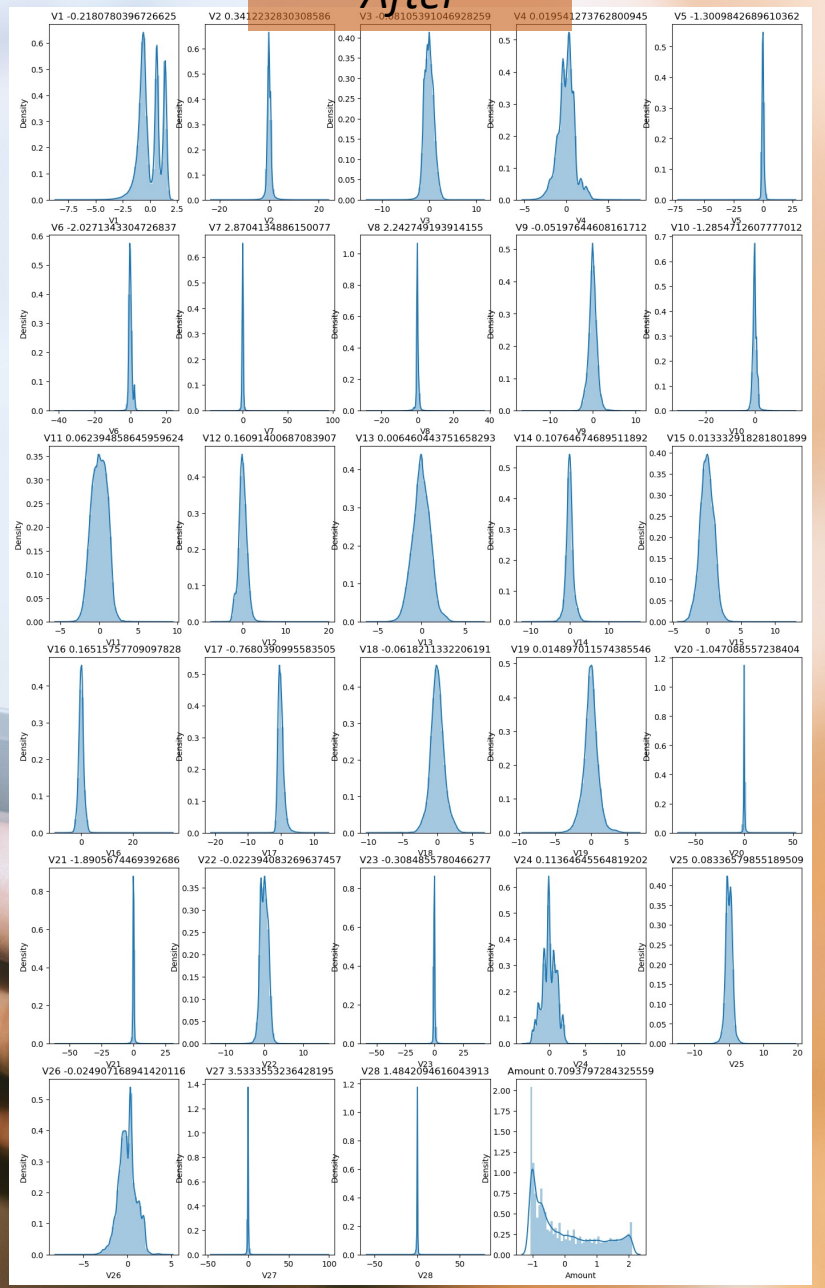
Fraudulent transactions are clustered in lower range of amount. Non-fraudulent transactions are spread throughout low to high range of amount.

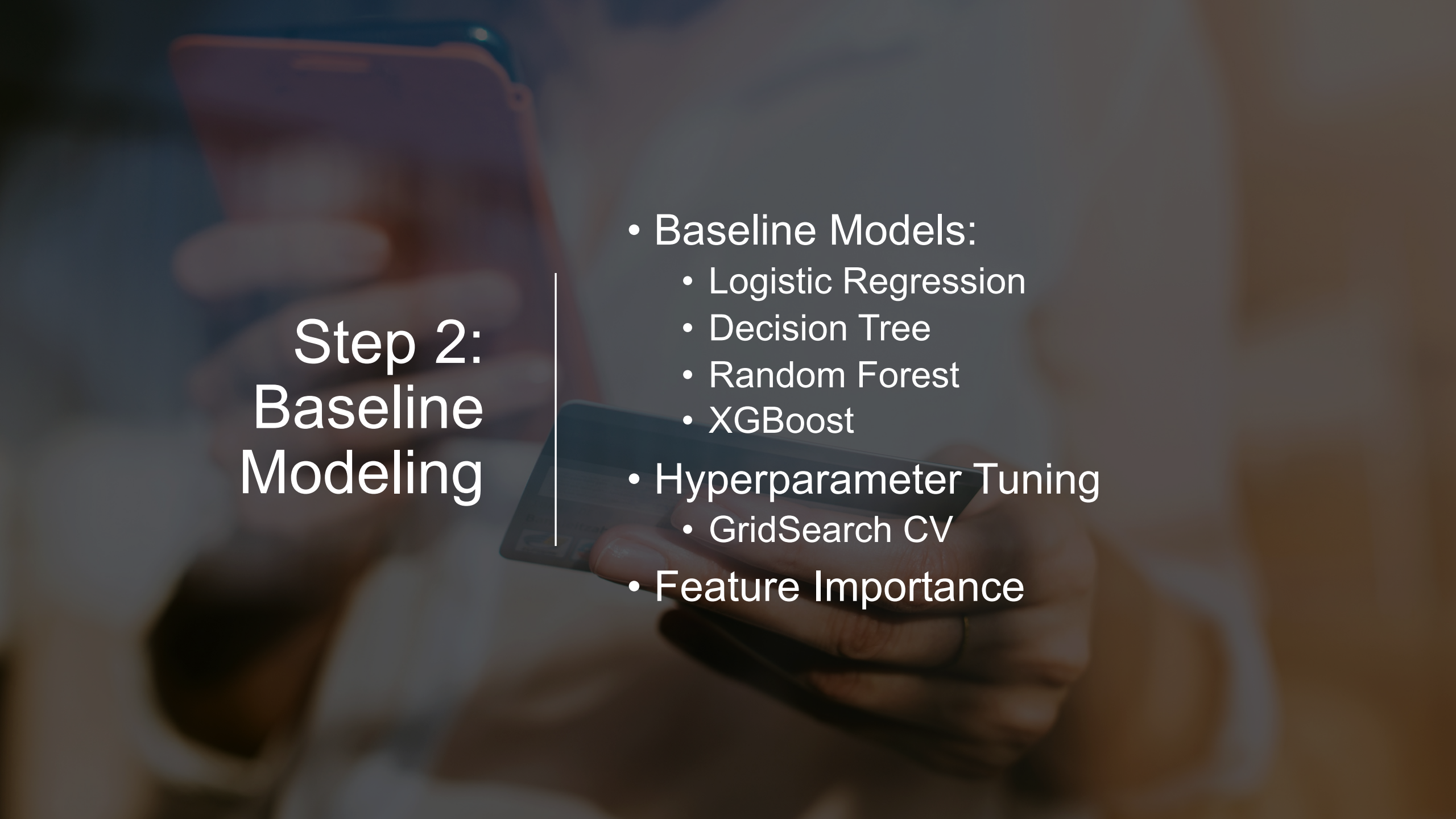
Before



Skewed variables were transformed to have normal distribution

After





Step 2: Baseline Modeling

- Baseline Models:
  - Logistic Regression
  - Decision Tree
  - Random Forest
  - XGBoost
- Hyperparameter Tuning
  - GridSearch CV
- Feature Importance

Baseline Models	Accuracy	Precision	Recall	F1-Score	ROC-AUC
Logistic Regression	0.998	0.771	0.562	0.651	0.97
Decision Tree	0.998	0.651	0.583	0.749	0.92
Random Forest	0.999	0.726	0.635	0.807	0.96
XGBoost	0.999	0.818	0.75	0.783	0.96

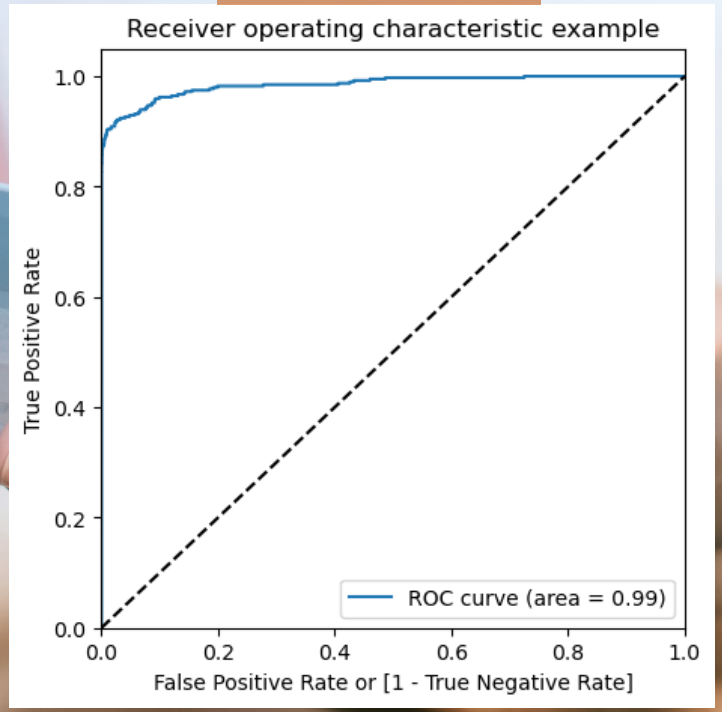
ROC-AUC is the best metric for imbalanced data

0.01 increase of score may convert into huge amount of savings for client.

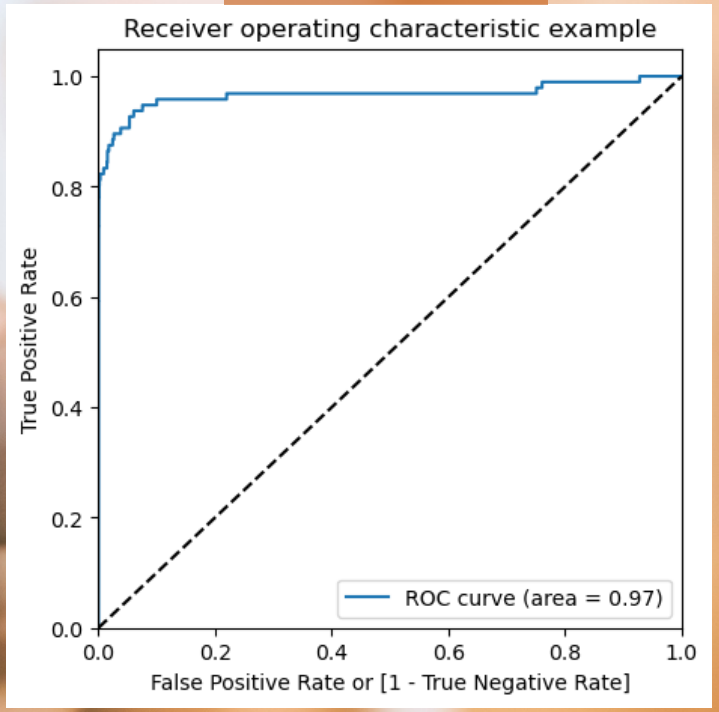
Logistic Regression is the best model on the imbalanced data

- Train set
  - Accuracy:- 0.999
  - Precision:- 0.922
  - Recall:- 0.659
  - F1-Score:- 0.769
  - ROC :- 0.99
- Test set
  - Accuracy:- 0.998
  - Precision:- 0.771
  - Recall:- 0.562
  - F1-Score:- 0.651
  - ROC :- 0.97

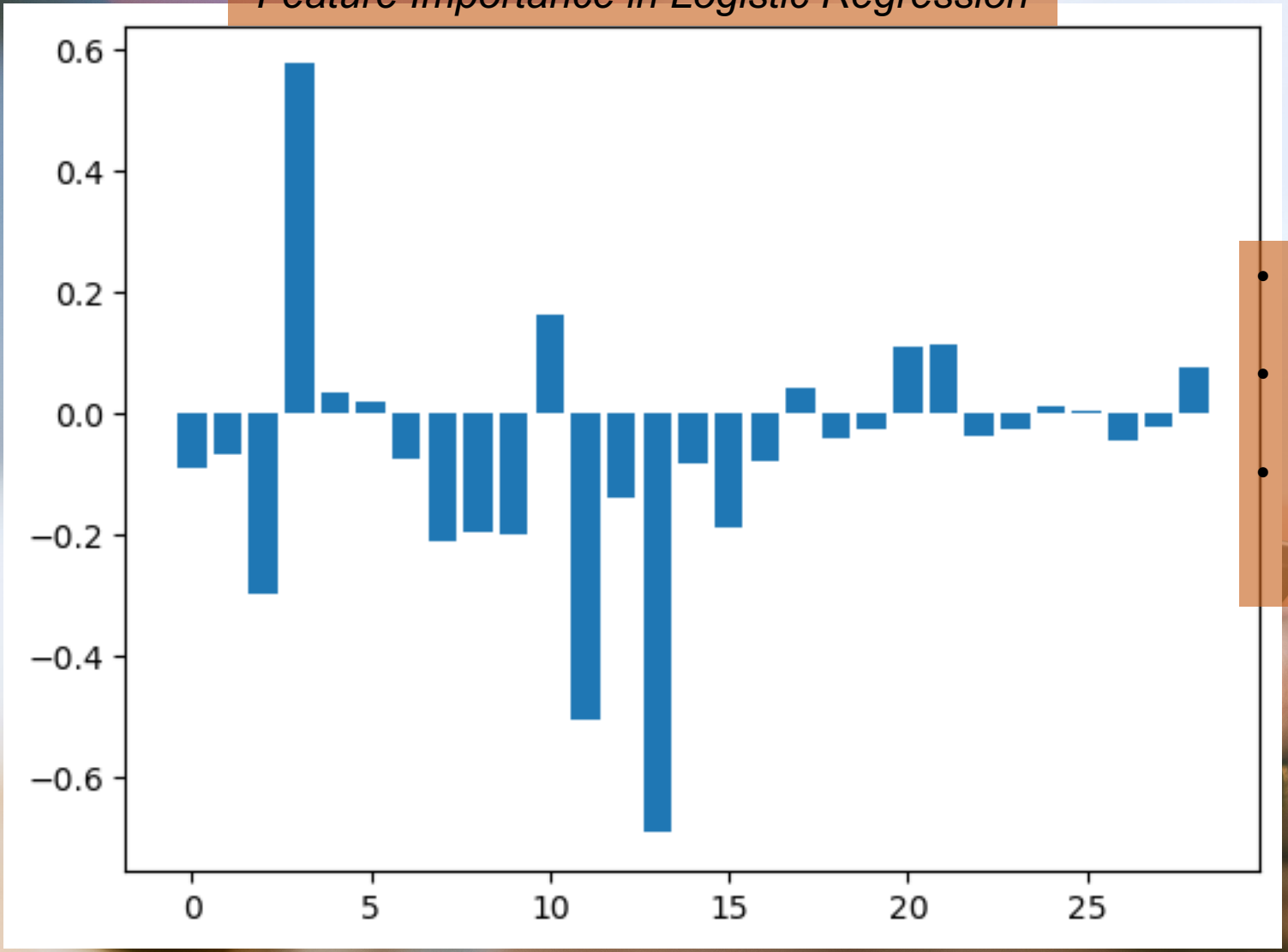
Train score



Test score



Feature Importance in Logistic Regression



- *Positive scores indicate a feature that predicts class 1*
- *Negative scores indicate a feature that predicts class 0.*
- *No clear pattern of important and unimportant features can be identified.*

Step 3: Class Balancing

- Undersampling
  - Logistic Regression
  - Decision Tree
  - Random Forest
  - XGBoost
- Oversampling
  - Logistic Regression
  - Decision Tree
  - XGBoost
- SMOTE
  - Logistic Regression
  - Decision Tree
  - XGBoost

Undersampling

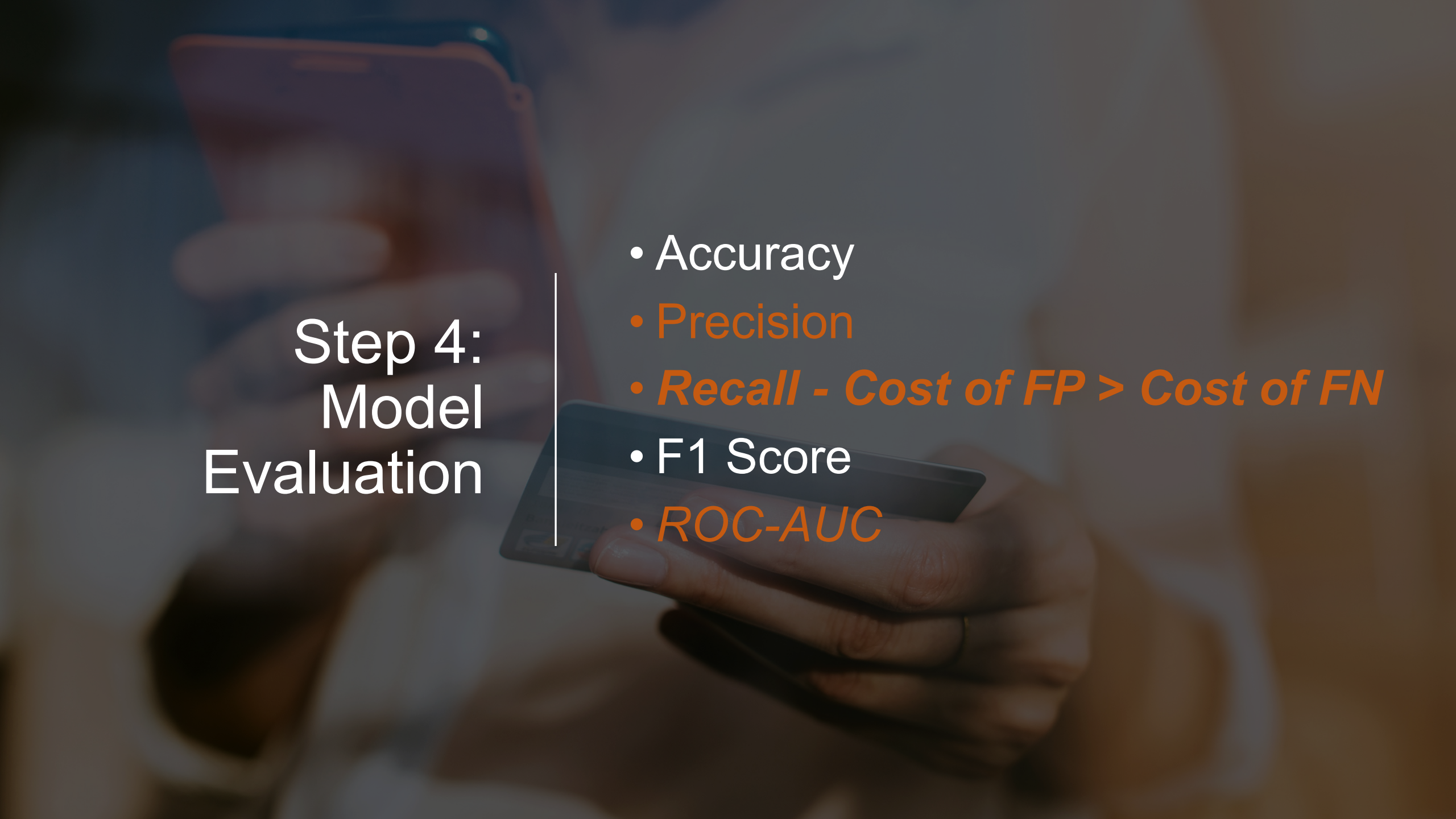
Baseline Models	Accuracy	Precision	Recall	ROC-AUC
Logistic Regression	0.978	0.062	0.875	0.96
Decision Tree	0.981	0.068	0.802	0.97
Random Forest	0.988	0.106	0.823	0.97
XGBoost	0.960	0.037	0.906	0.98

SMOTE

Baseline Models	Accuracy	Precision	Recall	ROC-AUC
Logistic Regression	0.974	0.056	0.896	0.97
Decision Tree	0.982	0.071	0.802	0.86
XGBoost	0.999	0.738	0.792	0.96

Oversampling

Baseline Models	Accuracy	Precision	Recall	ROC-AUC
Logistic Regression	0.976	0.059	0.885	0.97
Decision Tree	0.989	0.109	0.781	0.89
XGBoost	0.999	0.894	0.792	0.98



Step 4: Model Evaluation

- Accuracy
- Precision
- ***Recall - Cost of FP > Cost of FN***
- F1 Score
- *ROC-AUC*

Models	Recall	ROC-AUC
Logistic Regression	0.562	0.97
Decision Tree	0.583	0.92
Random Forest	0.635	0.96
XGBoost	0.75	0.96
Logistic Regression - Undersampling	0.875	0.96
Decision Tree - Undersampling	0.802	0.97
Random Forest - Undersampling	0.823	0.97
XGBoost - Undersampling	0.906	0.98
Logistic Regression - Oversampling	0.885	0.97
Decision Tree - Oversampling	0.781	0.89
XGBoost - Oversampling	0.792	0.98
Logistic Regression - SMOTE	0.896	0.97
Decision Tree - SMOTE	0.802	0.86
XGBoost - SMOTE	0.792	0.96

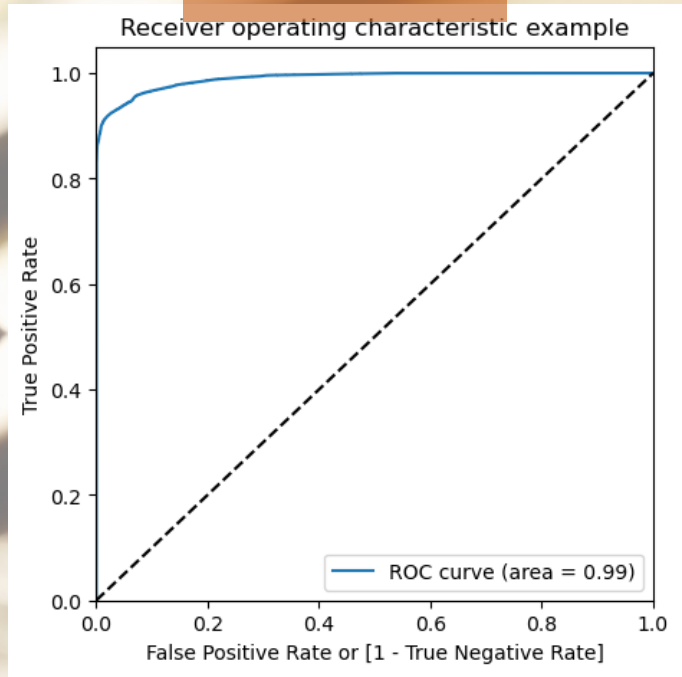

Best Model

Logistic regression model with SMOTE is the best model for its simplicity and less resource requirement.

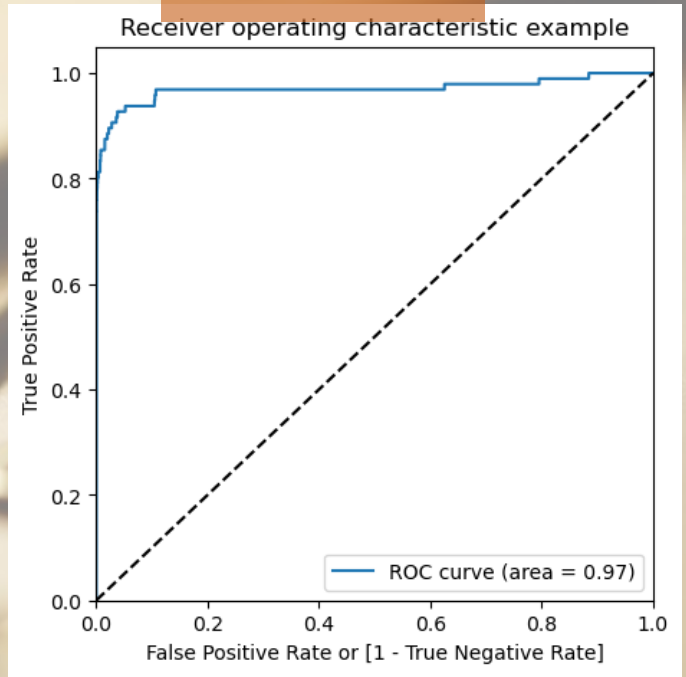
Best Model Metrics

- Train set
  - *Accuracy:- 0.948*
  - *Precision:- 0.974*
  - *Recall:- 0.922*
  - *ROC :- 0.99*
- Test set
  - *Accuracy:- 0.974*
  - *Precision:- 0.056*
  - *Recall:- 0.896*
  - *ROC :- 0.97*

Train score



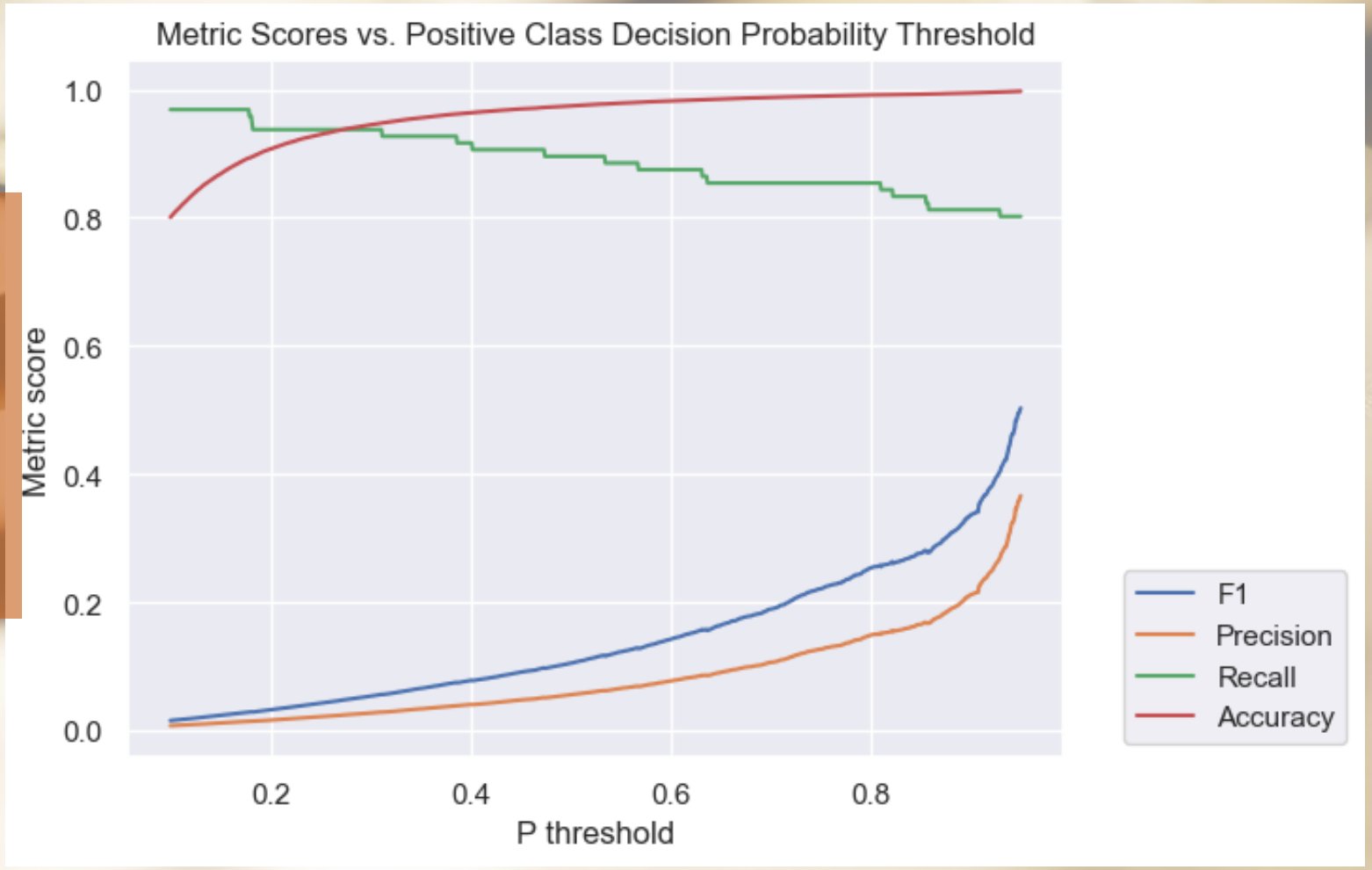
Test score



Probability threshold is 0.53, which results in highest TPR, lowest FPR and best ROC score.

Threshold Adjustment to Optimize F1

Logistic Regression Model with SMOTE has the best F1 score 0.503 at probability decision threshold >= 0.950



Business Takeaways

- Cost Benefit analysis
  - Consider required infrastructure, resources or computational power
  - Cost of deploying versus performance
  - Amount of monetary loss with little change in metrics
- Focus more on ***high recall*** for higher value transactions
- Focus more on high precision for smaller value transactions

Future Work

- Other ensembling methods
- More cross validation to reduce overfitting
- Track the best model’s performance over time

Thank you!