Credit Card Fraud Detection:

Data Storytelling

- Pallav Rajput
- IIIT-B

CONTENTS

01 Introduction

02 Problem Statement

03 Root Cause Analysis

04 Findings

05 Recommendations

06 Cost-Benefit Analysis

07 Conclusion

01 Improving Credit Card Fraud Detection with Machine Learning

Credit card fraud involves dishonest activities to obtain financial information without account holder authorization, often through skimming. Other methods include manipulating genuine cards, creating counterfeit cards, stealing or losing cards, and fraudulent telemarketing.

02 Problem Statement

The rise in fraud transactions has significantly impacted credit card companies, posing significant challenges to their business goals of retaining profitable customers and threatening financial loss, trust, and credibility. The rise in digital payment channels further exacerbates this issue.

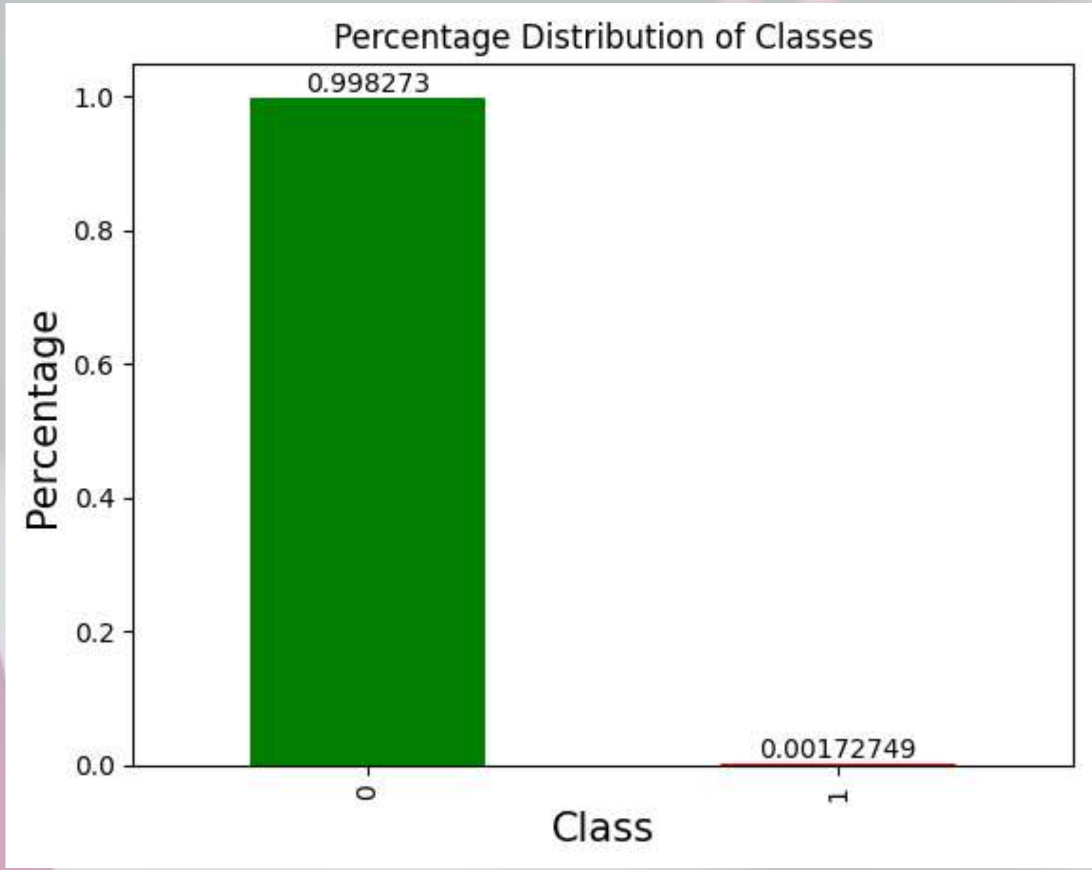
The project pipeline consists of several steps:

1. **Understanding Data**: Load and analyze the data to identify important features for the model.
2. **Exploratory Data Analysis**: Explore the data to gain insights and prepare it for model training.
3. **Data Splitting**: Split the data into training and testing sets using stratified k-fold cross-validation.
4. **Model Building & Hyperparameter Tuning**: Train models and adjust hyperparameters for optimal performance.
5. **Model Evaluation**: Evaluate models using suitable metrics, with a focus on accurately detecting fraudulent transactions.
6. **Business Impact Analysis**: Assess how the model's results impact business goals.

03

Root Cause Analysis

- **Imbalanced Dataset:** Unequal class representation.
- **Causes:**
- Real-world distributions (e.g., rare events like fraud).
- Data collection biases.
- Operational realities (e.g., more normal than malicious activity in cybersecurity).
- **Impact on Model Performance:**
- Bias toward the majority class.
- Poor generalization to minority class patterns.
- Misleading performance metrics (e.g., high accuracy despite poor minority class detection).
- **Mitigation Techniques:**
- **SMOTE** (Synthetic Minority Over-sampling Technique).
- **AdaSYn** (Adaptive Synthetic Sampling)Using appropriate metrics like precision, recall, F1-score, and AUC-ROC.

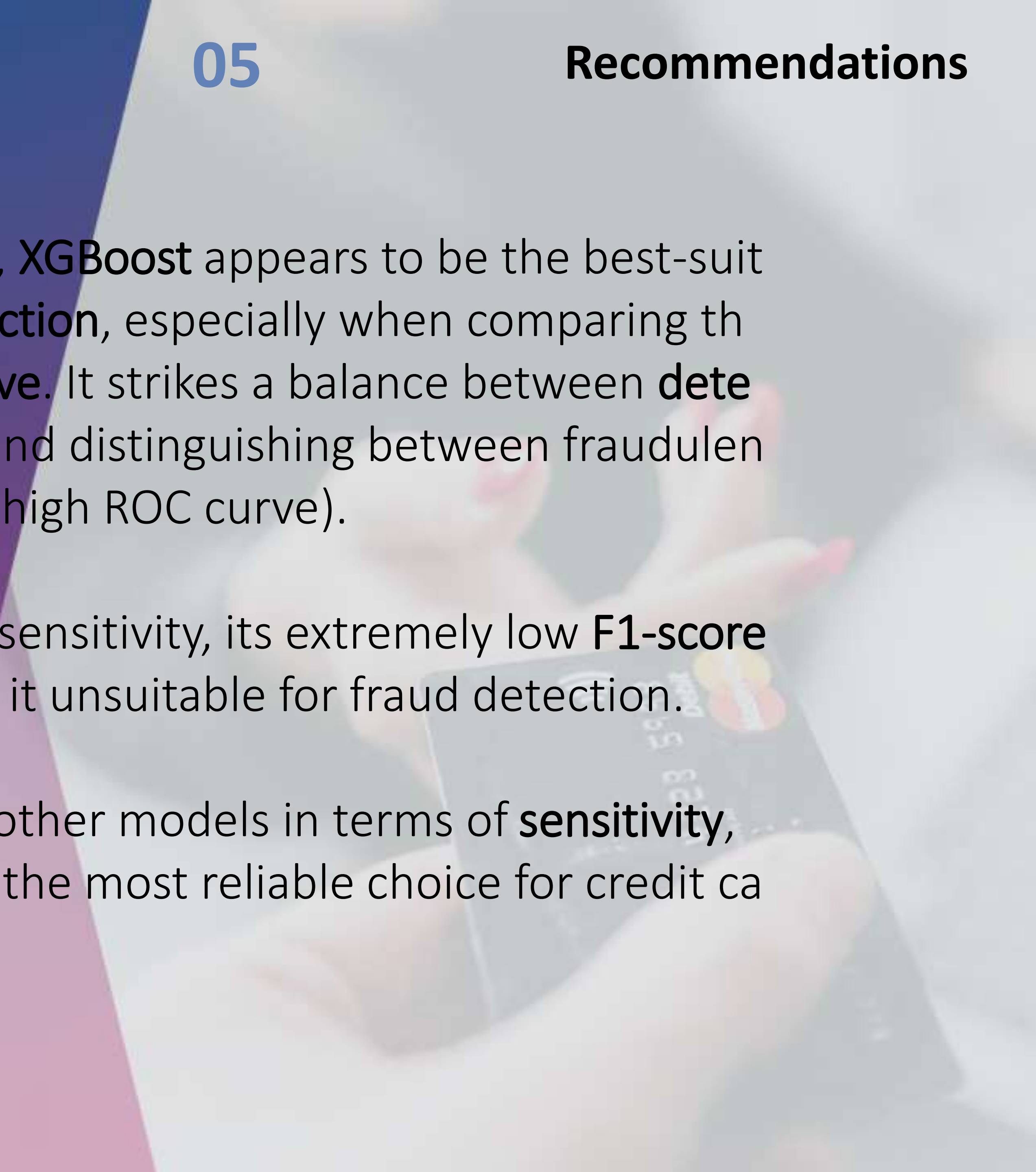


04

Findings

	Accuracy	Senstivity	Specificity	F1-Score	ROC Curve
IMBALANCED					
Logistic Regression	0.99	0.68	0.99	0.7	0.94
XGBoost	0.99	0.76	0.99	0.86	0.97
Decision Tree	0.99	0.69	0.99	0.73	0.93
Random Forest	0.94	0.88	0.94	0.05	0.95
SMOTE					
Logistic Regression	0.98	0.84	0.98	0.14	0.96
XGBoost	0.99	0.82	0.99	0.8	0.96
Decision Tree	0.98	0.81	0.99	0.16	0.88
AdaSYn					
Logistic Regression	0.94	0.89	0.95	0.058	0.96
XGBoost	0.99	0.82	0.99	0.8	0.96
Decision Tree	0.97	0.81	0.97	0.11	0.91

05 Recommendations



Based on the performance metrics, **XGBoost** appears to be the best-suited model for **credit card fraud detection**, especially when comparing the **sensitivity**, **F1-score**, and **ROC curve**. It strikes a balance between **detecting fraud cases** (high sensitivity) and distinguishing between fraudulent and non-fraudulent transactions (high ROC curve).

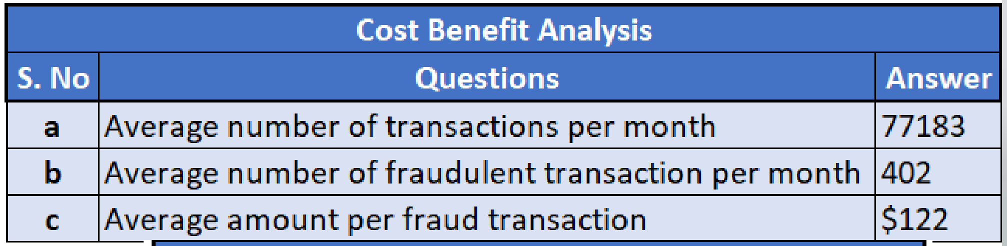
Although **Random Forest** has good sensitivity, its extremely low **F1-score** in the IMBALANCED dataset makes it unsuitable for fraud detection.

XGBoost consistently outperforms other models in terms of **sensitivity**, **F1-score**, and **ROC curve**, making it the most reliable choice for credit card fraud detection.

06

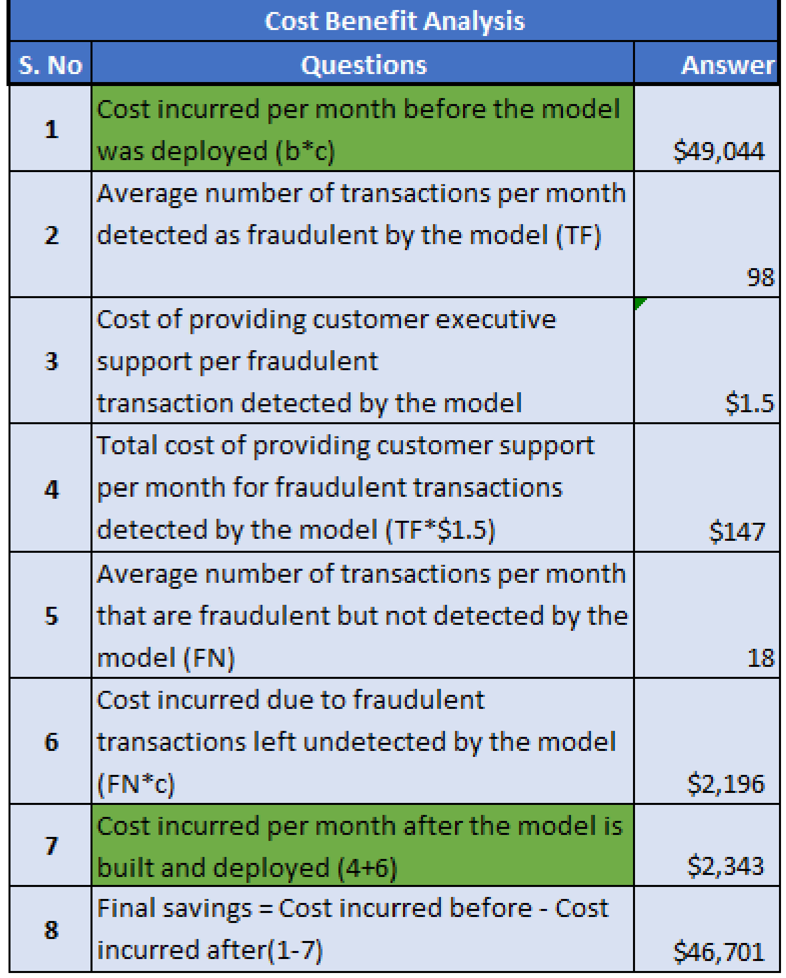
Cost-Benefit Analysis

Banks need to decide whether they need high precision or high recall for their fraud detection. For smaller banks with smaller average transaction values, high precision is necessary to label relevant transactions as fraudulent. Human verification can be added to every flagged transaction, but low precision can be burdensome. For larger banks with larger transaction values, low recall can hinder detection of non-fraudulent transactions. To save banks from high-value fraudulent transactions, a high recall is needed to detect actual fraudulent transactions. The best model should be chosen based on the profit or dollar/rupee value saved.



Cost Benefit Analysis

S. No	Questions	Answer
a	Average number of transactions per month	77183
b	Average number of fraudulent transaction per month	402
c	Average amount per fraud transaction	$122



Cost Benefit Analysis

S. No	Questions	Answer
1	Cost incurred per month before the model was deployed (b*c)	$49,044
2	Average number of transactions per month detected as fraudulent by the model (TF)	98
3	Cost of providing customer executive support per fraudulent transaction detected by the model	$1.5
4	Total cost of providing customer support per month for fraudulent transactions detected by the model (TF*$1.5)	$147
5	Average number of transactions per month that are fraudulent but not detected by the model (FN)	18
6	Cost incurred due to fraudulent transactions left undetected by the model (FN*c)	$2,196
7	Cost incurred per month after the model is built and deployed (4+6)	$2,343
8	Final savings = Cost incurred before - Cost incurred after(1-7)	$46,701

07

Conclusion

Based on the performance metrics, XGBoost appears to be the best-suited model for credit card fraud detection, especially when comparing the sensitivity, F1-score, and ROC curve.

It strikes a balance between detecting fraud cases (high sensitivity) and distinguishing between fraudulent and non-fraudulent transactions (high ROC curve).Although Random Forest has good sensitivity, its extremely low F1-score in the IMBALANCED dataset makes it unsuitable for fraud detection.

XGBoost consistently outperforms other models in terms of sensitivity, F1-score, and ROC curve, making it the most reliable choice for credit card fraud detection.

THANK YOU