CONVOLVE

2025

Team :
soumyashree6002

INTRODUCTION

Problem Statement and our approach:

In collaboration with IIT Bombay, this project focuses on developing a Behaviour Score for Bank A to strengthen its credit card risk management framework. The Behaviour Score is a predictive model aimed at identifying the probability of existing credit card customers defaulting. By leveraging advanced machine learning techniques, this score enables the bank to proactively manage portfolio risk, ensuring profitability and customer retention.

The solution utilizes a historical dataset of 96,806 credit card accounts, containing attributes ranging from credit limits and transaction histories to bureau records. The approach also includes a validation dataset of 41,792 accounts for generating default probabilities. Our methodology involves meticulous data preprocessing, exploratory data analysis, feature engineering, and the application of a robust Random Forest model, enhanced with techniques like SMOTE to handle data imbalance.

This solution effectively addresses the problem by integrating statistical rigor with machine learning advancements. It ensures scalability, accuracy, and actionable insights, enabling Bank A to deploy a comprehensive risk assessment strategy for its credit card portfolio.

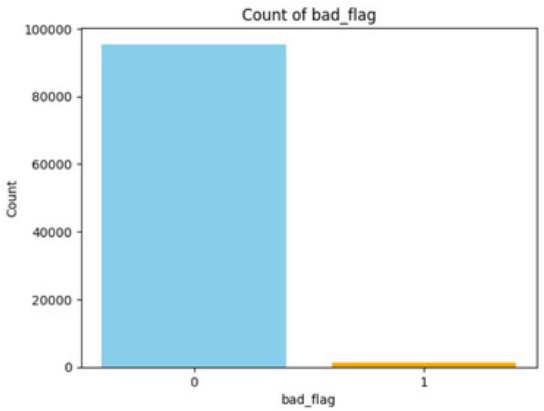
DATA ANALYSIS

This document provides a detailed overview of the steps undertaken to predict the probability of credit card customers defaulting. The solution includes data cleaning, analysis, feature engineering, model selection, and evaluation. The final submission includes the predicted probabilities for the validation dataset along with a comprehensive explanation of the approach.

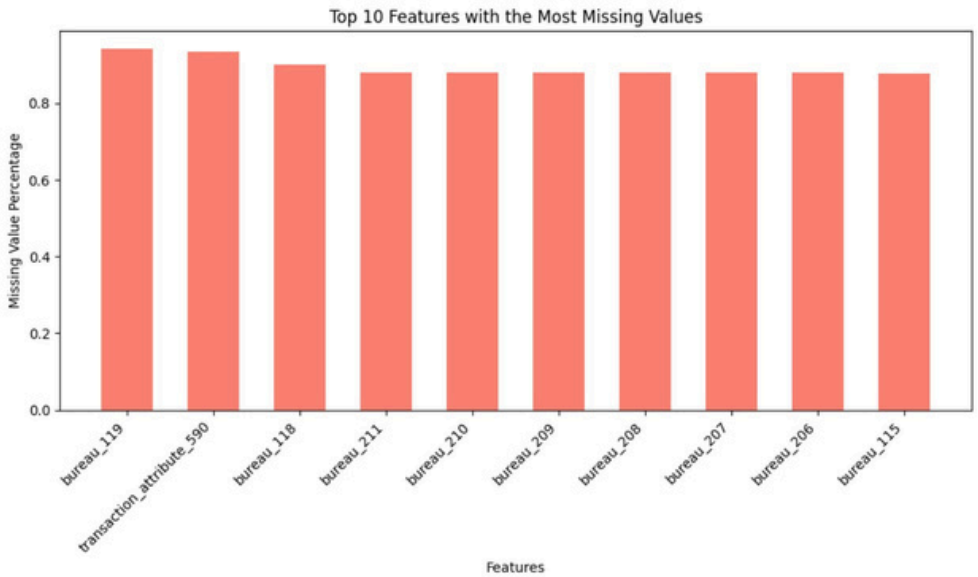
EXPLORATORY DATA ANALYSIS:

- **Descriptive Analysis of target variable**: Summary statistics (mean, median, standard deviation) were calculated for 'bad_flag' to understand central tendencies and dispersion.

As we can see the graph beside indicates that the dataset is highly imbalanced, indicating the need to balance the dataset in order to get a good probability prediction of defaulters



- **Missing Data Analysis**: Bar plot of the top 10 features with more than 50% missing data to identify heavily impacted columns. These indicate the need to fill the missing values due to their high occurence so as to make our data set ready for input to the model




```
df.isna().sum()
```

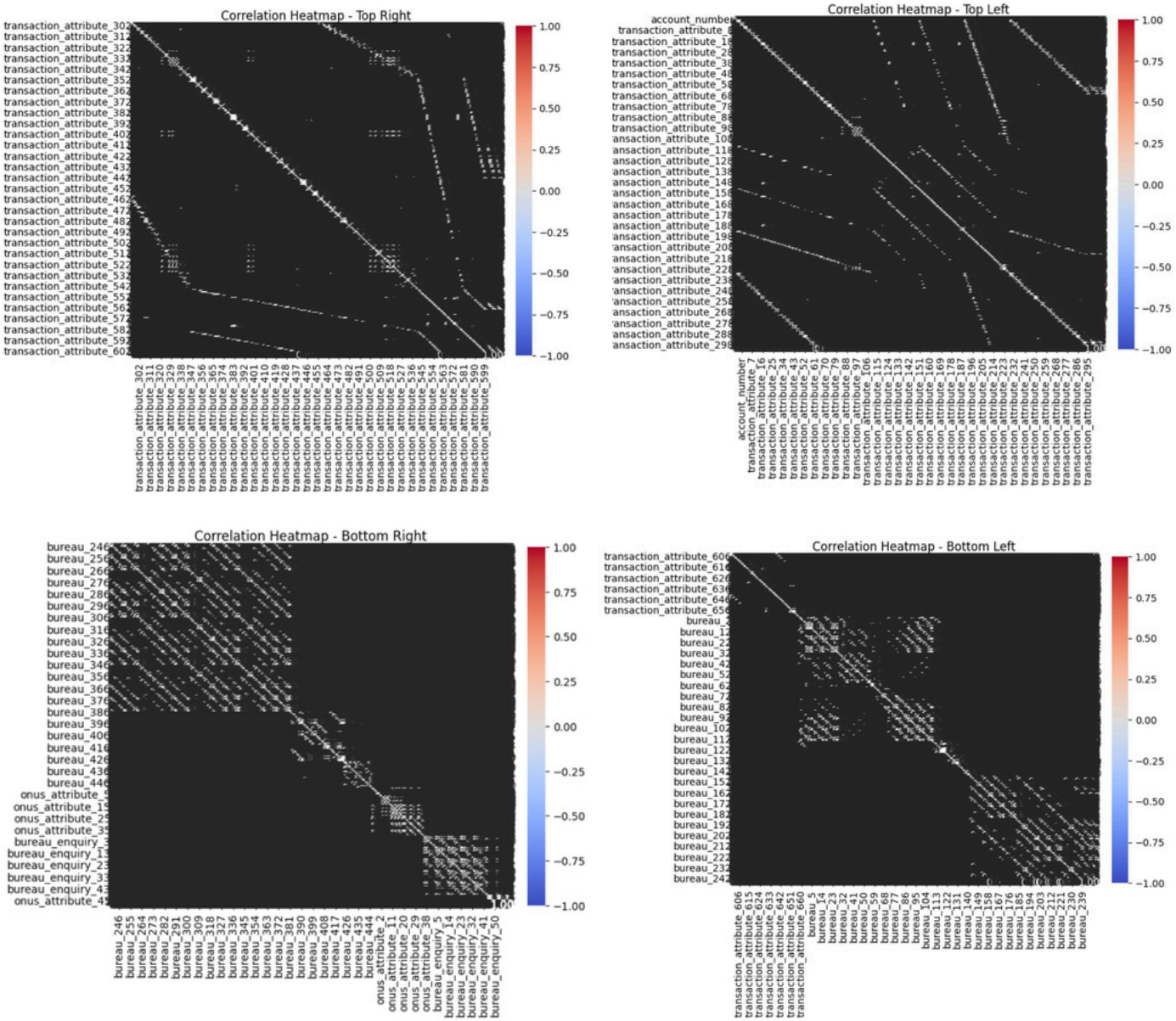
	0
account_number	0
bad_flag	0
onus_attribute_1	25231
transaction_attribute_1	25231
transaction_attribute_2	25231
...	...
onus_attribute_44	85196
onus_attribute_45	85196
onus_attribute_46	85196
onus_attribute_47	85196
onus_attribute_48	85196

1216 rows × 1 columns

dtype: int64

DATA ANALYSIS

- **Correlation Heatmap**: Displayed correlations between numerical features. Identified highly correlated features (> 0.5 correlation) for potential removal to avoid multicollinearity.

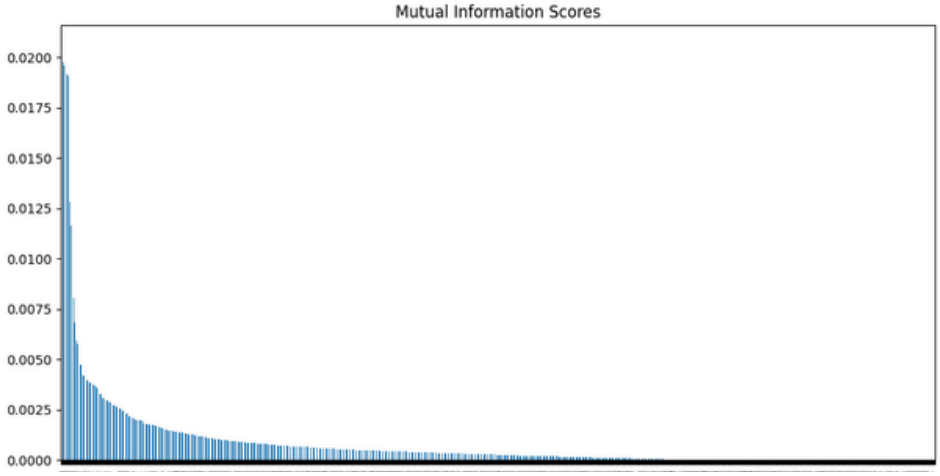


The correlation heatmaps generated in chunks clearly suggest the presence of highly correlated columns, thereby giving us an idea about feature engineering we can carry out in further steps. Due high number of columns present the cool 'coolwarm' theme could not be shown

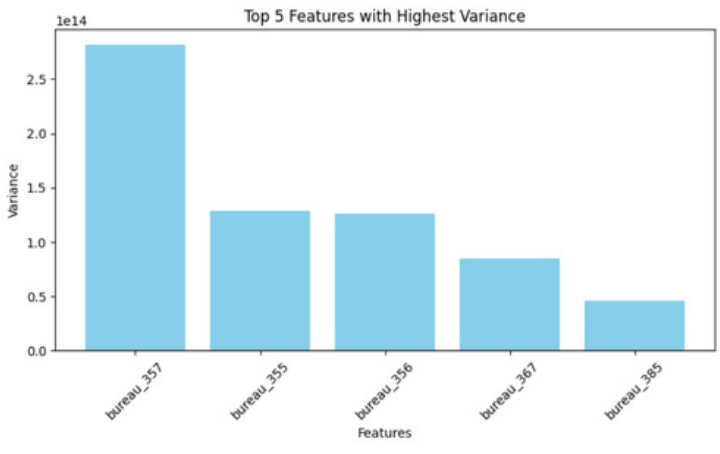
DATA ANALYSIS

- **Mutual Information**: Bar plots showing the mutual information scores of features with the target variable to identify their predictive power. This analysis highlighted the top features with the highest predictive influence on the target variable.

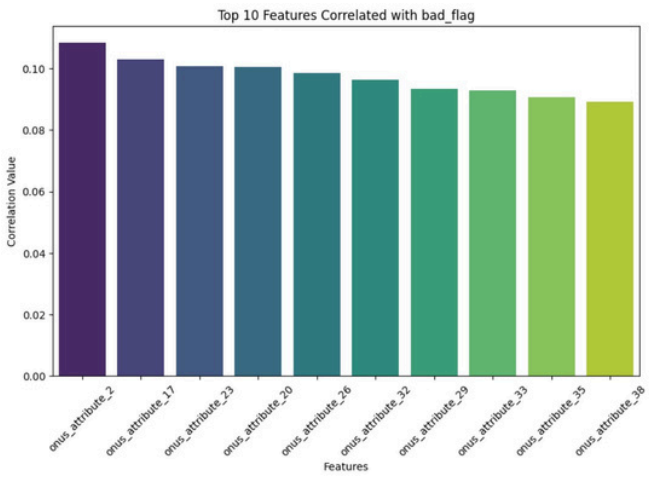
As there are high number of columns, we have refrained from showing the column names as they are not clear. This graph clearly points out the features which have high mutual information with the target variable and therefore play a vital role in our predictive model



- **Variance Analysis**: Bar plots for the top 5 features with the highest variance. These columns along with the others which have a variance above a particular threshold are recorded in EDA



- **Correlation Analysis**: Bar plots for the top 5 features most correlated with the target variable.



DATA ANALYSIS

DATA PREPROCESSING AND CLEANING:

Data cleaning is essential for ensuring accurate model predictions. Below are the steps applied:

Missing Data Handling

Column Removal: Columns with more than 50% missing data were removed to minimize noise and ensure model robustness.

Imputation: For columns with less than 50% missing data:

Numerical columns:
Missing values were replaced with the mean

Analysis of Missing Values

Pattern Identification: A missing data heatmap was generated to visualize missing value distributions and identify patterns.

Correlation with Target Variable: Missingness was analyzed for potential correlations with the target variable to determine whether it could serve as an informative feature.
Top Missing Features: Missing percentages were calculated for all columns, and the top 10 columns with more than 50% missing data were plotted.

Balancing the Dataset with SMOTE
Algorithm Overview: SMOTE works by synthesizing new data points for the minority class. It selects a random data point from the minority class, identifies its k-nearest neighbors (default: k=5), and creates a synthetic data point by interpolating between the selected point and one of its neighbors.

Why SMOTE Helps:
It reduces overfitting by providing diverse synthetic examples instead of duplicating existing minority class samples.
Balances the dataset, allowing the model to learn patterns for both classes effectively.

Application in This Context:
The dataset was highly imbalanced, with far fewer defaults compared to non-defaults. Using SMOTE ensured that the minority class was adequately represented during training, enabling the model to improve recall and precision for default predictions.

DATA ANALYSIS

Low Variance Filtering:

Low variance filtering helps in simplifying the dataset by removing features that exhibit very little variation across samples. These features typically do not provide meaningful information for distinguishing between classes (e.g., default vs. non-default).
In this case, applying low variance filtering has the following benefits:

1. **Reduces Noise**: Features with little variance often contain noise rather than useful signal, which can negatively impact the model's performance.
2. **Improves Computational Efficiency:** Removing redundant features reduces the size of the dataset, allowing the model to train faster and use fewer computational resources.
3. **Mitigates Overfitting**: By focusing only on features with meaningful variation, the model is less likely to learn spurious patterns that don't generalize to new data.

Columns with very low variance (variance < 0.01) were removed as they provide little to no discriminative power for the model.

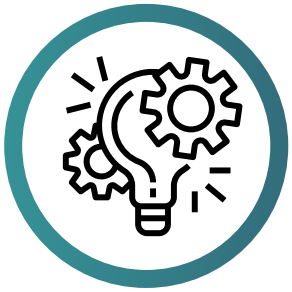
Scaling and Normalization:

Scaling ensures that all numerical features have a similar magnitude, which is crucial for algorithms sensitive to feature magnitudes, such as distance-based models (e.g., k-NN, SVMs) and gradient-based models like Random Forest.
In this case:

1. **Improves Model Convergence**: Features which could range from hundreds to millions and also those with smaller values are scaled to a common range, ensuring no single feature disproportionately affects the model's learning process.
2. **Ensures Fair Weightage**: Without scaling, larger-magnitude features might dominate the importance calculations in some algorithms, even if they aren't more predictive.
3. **Enhances Model Performance**: Consistently scaled features allow models like Random Forest to better focus on patterns rather than being misled by differing scales.

Numerical features were standardized using StandardScaler to ensure consistency in magnitude.

FEATURE SELECTION



Key steps in our feature selction:

Correlation-Based Feature Removal

How It Works: Highly correlated features provide redundant information and can lead to multicollinearity, which may confuse models and reduce interpretability.
Why It Helps:
Removing correlated features ensures that the model focuses on unique, independent signals. It reduces overfitting by eliminating redundant noise from the dataset.
Relevance to This Solution:
In our dataset, highly correlated features like multiple transaction metrics or credit usage metrics were pruned, simplifying the model training process.
This also improved the stability of the Random Forest's feature importance rankings, as features with overlapping information were excluded.

Random Forest Feature Importance

How It Works: Random Forest ranks features based on their contribution to reducing impurity (e.g., Gini impurity or entropy) during decision tree splits.
Why It Helps:
Identifies the most predictive features, allowing for targeted feature selection and handles both numerical and categorical data effectively, making it versatile for our dataset.
Relevance to This Solution:
By identifying the top 150 features, Random Forest reduced the dataset complexity without compromising model performance.
This allowed us to focus on features with the highest potential to influence the target variable (default or non-default).

Dimensionality Reduction through Autoencoding

Role in Feature Selection: Autoencoders identify patterns and compress features into a lower-dimensional space by learning key representations of the input data. These representations retain the most critical information while discarding noise and redundancy.
Benefits in This Case:
Autoencoding helped reduce the 1200 original features into a smaller subset, removing redundant or less informative features.
It improved computational efficiency while retaining essential predictive power for the classification task.

MODELS-1

Basic ML Models used are:-

CATBOOST

Performance:
Handles categorical data natively, reducing preprocessing time and improving predictive accuracy and offers superior performance in capturing complex feature interactions with minimal overfitting.

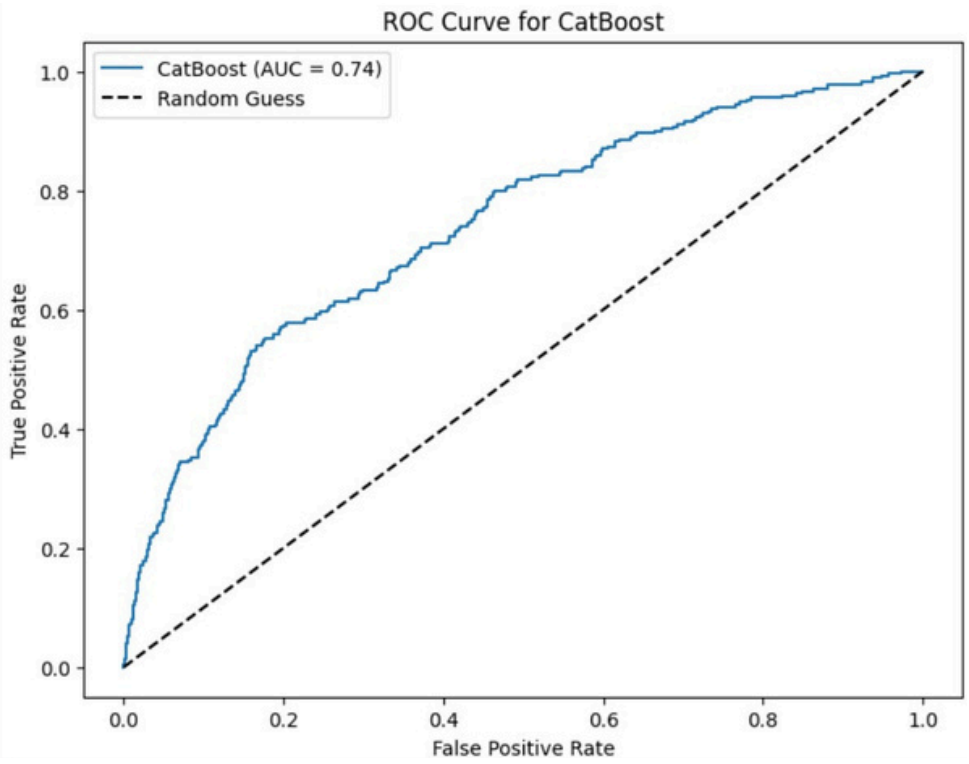
Significance:
Performs best in credit risk modeling tasks due to its ability to handle categorical and numerical data seamlessly.
Its optimized handling of noisy and imbalanced datasets makes it a top contender for predicting default probabilities.

```
CatBoost Classification Report:
              precision    recall  f1-score   support

           0       0.99      1.00      0.99     19092
           1       0.18      0.01      0.02       270

    accuracy                           0.99     19362
   macro avg       0.58      0.51      0.51     19362
weighted avg       0.97      0.99      0.98     19362

CatBoost ROC-AUC Score: 0.74
```



MODELS-2

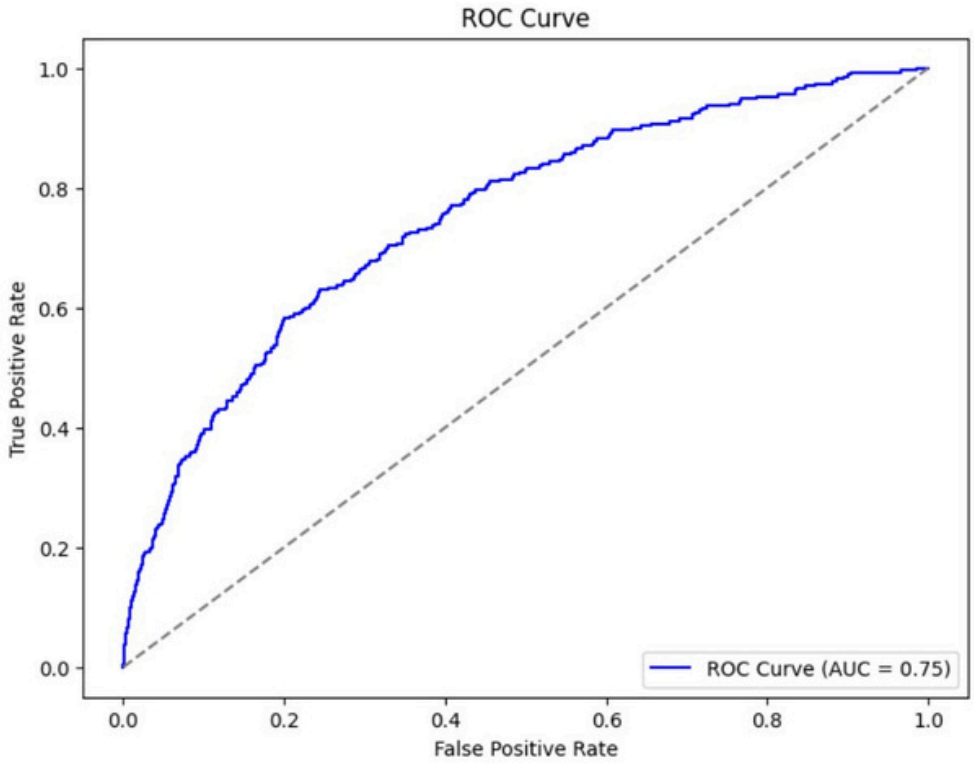
XGBOOST

Performance:
Outperforms Random Forest and Logistic Regression by effectively handling feature interactions and overfitting.
Training time is slower compared to LightGBM and CatBoost but yields strong performance with careful tuning.
Significance:
Excels in leveraging engineered features and handling noisy or imbalanced datasets.
Its versatility makes it a reliable choice for accurate predictions in risk assessment problems like this one.

```
AUC Score: 0.75
F1_Score: 0.13
Accuracy: 0.97
              precision    recall  f1-score   support

           0       0.99      0.98      0.98     19092
           1       0.11      0.16      0.13       270

    accuracy                           0.97     19362
   macro avg       0.55      0.57      0.56     19362
weighted avg       0.98      0.97      0.97     19362
```



MODELS-3

RANDOM FOREST

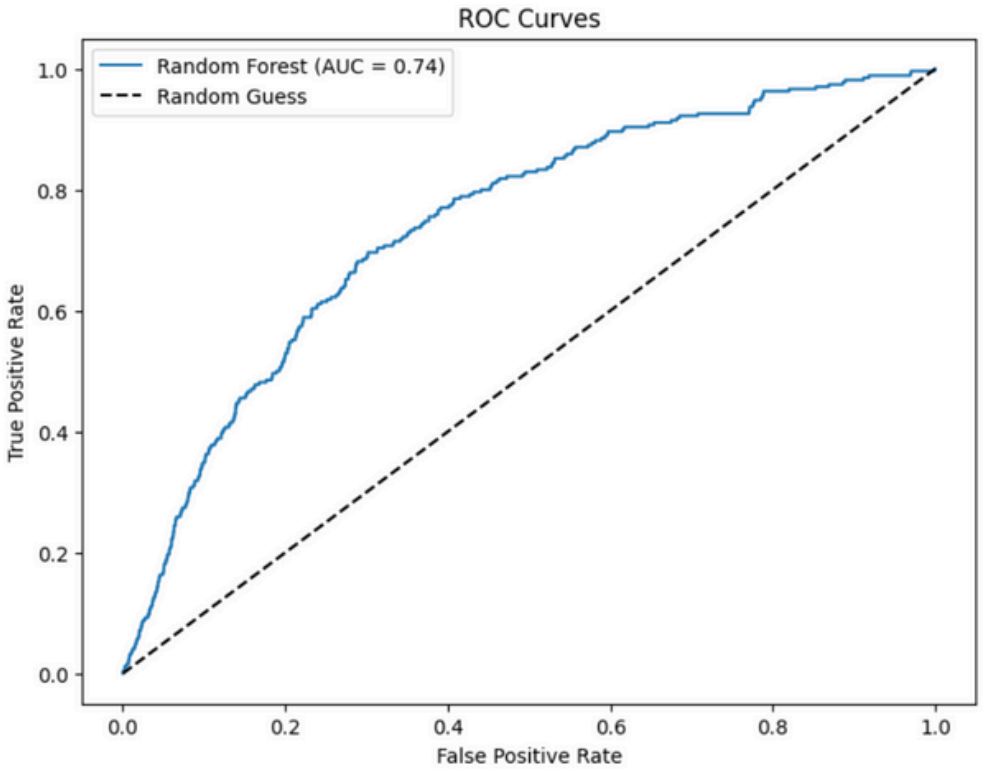
Performance:
Handles high-dimensional data well and provides feature importance rankings.
Suffers from longer training times and less optimized handling of categorical features compared to newer models.
Significance:
Provides robust predictions and handles feature interactions effectively.
Suitable for datasets with mixed data types and some level of noise, as in the current problem statement.
May struggle with imbalanced datasets unless tuned or combined with techniques like SMOTE.

```
Random Forest Classification Report:
              precision    recall  f1-score   support

           0       0.99      0.90      0.94     19092
           1       0.05      0.35      0.08       270

    accuracy                           0.89     19362
   macro avg       0.52      0.62      0.51     19362
weighted avg       0.98      0.89      0.93     19362

Random Forest ROC-AUC Score: 0.74
```

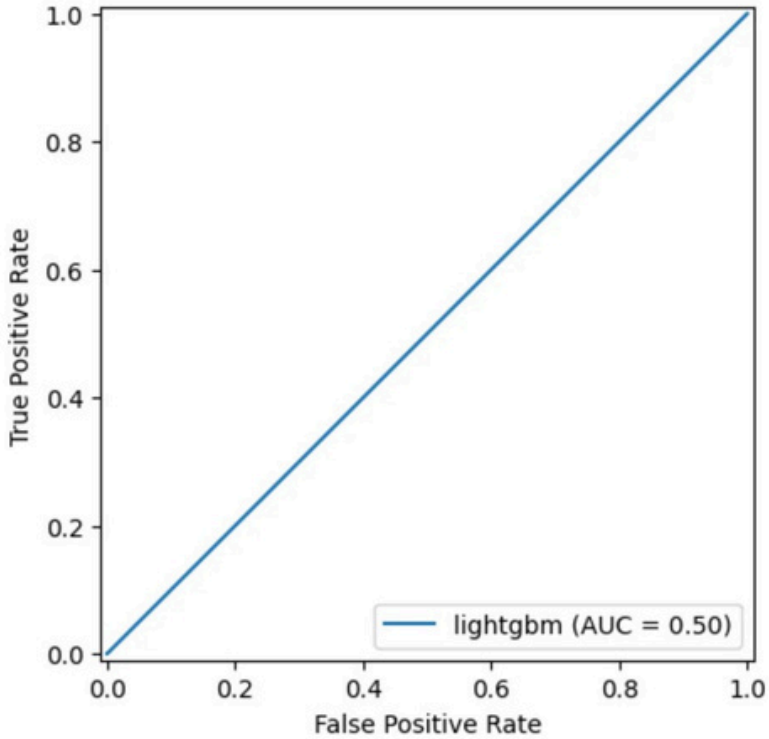


MODELS-4

LIGHT GBM

Performance:
Excels in handling numerical features and sparse data.
Faster training than XGBoost due to its histogram-based algorithm and leaf-wise tree growth.
Significance:
Balances performance and computational efficiency, making it ideal for large-scale credit card datasets.
Works well with minimal parameter tuning, making it a practical choice in time-constrained projects.

	precision	recall	f1-score	support
0	0.99	1.00	0.99	19092
1	0.00	0.00	0.00	270
accuracy			0.99	19362
macro avg	0.49	0.50	0.50	19362
weighted avg	0.97	0.99	0.98	19362



MODELS-5

LOGISTIC REGRESSION

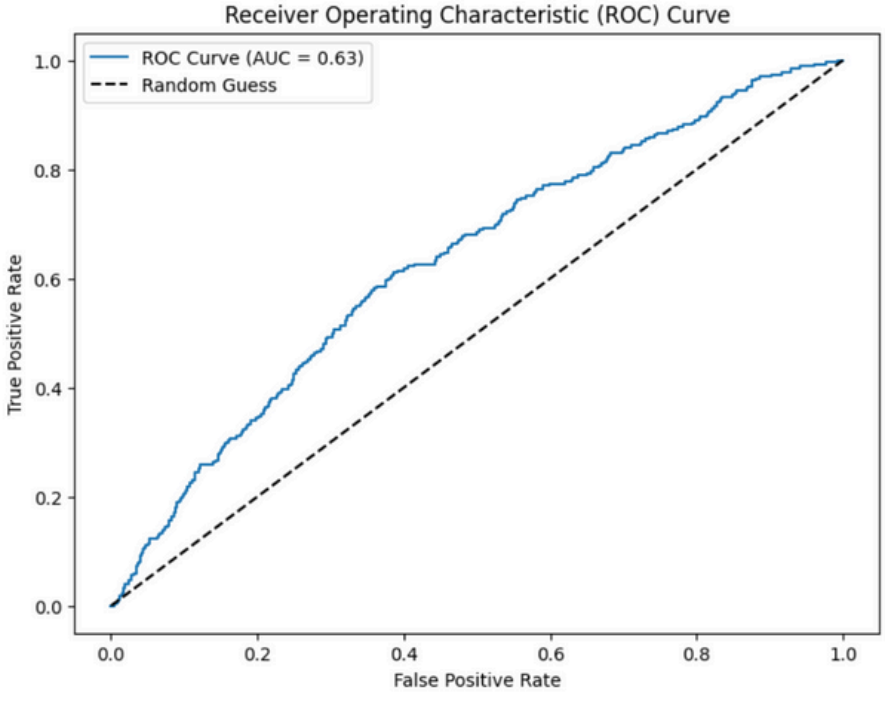
Performance:
Performs poorly in capturing complex patterns due to its linear nature.
Struggles with high-dimensional and highly correlated features without prior feature engineering.
Significance:
Acts as a baseline model to compare performance improvements from more advanced models.
Useful for understanding basic relationships between features and default probabilities due to its interpretability.

```
Classification Report:
              precision    recall  f1-score   support

           0       0.99      0.66      0.80     19092
           1       0.02      0.55      0.04       270

    accuracy                           0.66     19362
   macro avg       0.51      0.61      0.42     19362
weighted avg       0.98      0.66      0.78     19362

Confusion Matrix: [[12678  6414]
 [  122   148]]
Accuracy: 0.66
ROC-AUC Score: 0.63
```



MODEL COMPARISON

EVALUATION METRICS	XGBoost	CatBoost	LGBM	Random Forest	Logistic Regression
F1 -SCORE	0.13	0.03	0.00	0.11	0.04
ACCURACY	0.97	0.99	0.98	0.89	0.66
AUC-ROC:	0.75	0.74	0.5	0.74	0.63

Metrics Used

- **Log Loss:** Evaluated how well the predicted probabilities aligned with the actual outcomes.
- **Accuracy:** Measured the overall correctness of the model by calculating the ratio of correctly predicted instances to total instances.
- **F1 Score:** Measured the balance between precision and recall.
- **AUC-ROC:** Assessed the model's ability to distinguish between default and non-default classes.

CONCLUSION

The solution effectively predicts credit card default probabilities through a well-defined process that addresses key challenges. Below are the highlights:

XGBoost as Final Model:
- Chosen for its ability to handle complex feature interactions, class imbalance, and sparse data.
- Achieved 97% accuracy and an F1 score of 0.13 after rigorous hyperparameter tuning.

Balancing Data:
- SMOTE was used to address class imbalance, improving the model's generalization on defaults.

Feature Selection:
- Random Forest importance rankings and correlation-based filtering reduced noise and improved computational efficiency.

Future Enhancements:
- Real-time data integration for dynamic predictions.
- Explainability techniques to improve stakeholder trust.

Impact and Relevance:
- Enables proactive risk management by predicting defaults, aiding interventions, and optimizing credit policies.
- Scalable and adaptable to changing datasets.

This solution provides a robust and scalable framework for improving credit risk management and operational decision-making for financial institutions.