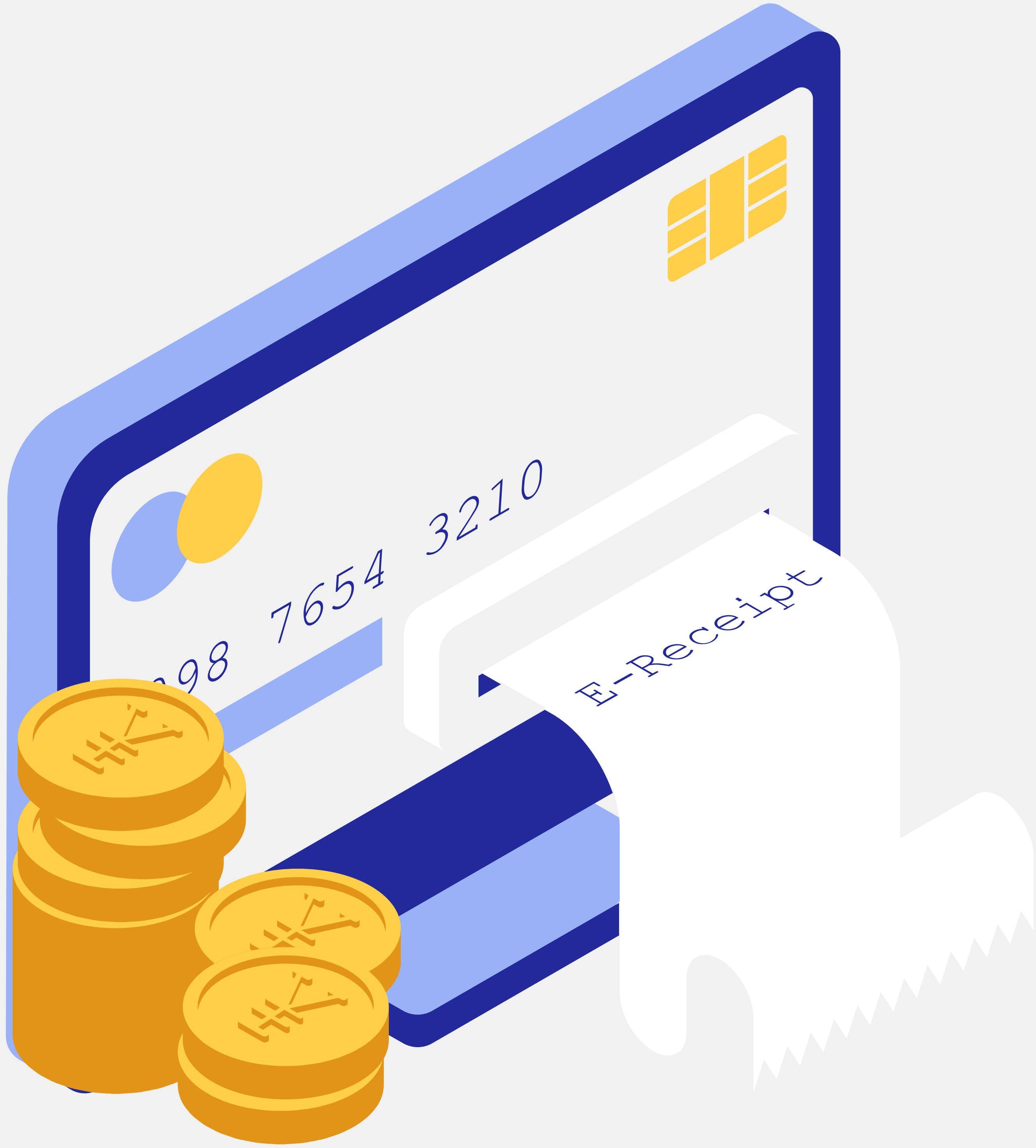


DS 861- Final Project | Spring 2024

Predicting Credit Card Approvals

Ananyaa Shahi and Vivid Liu

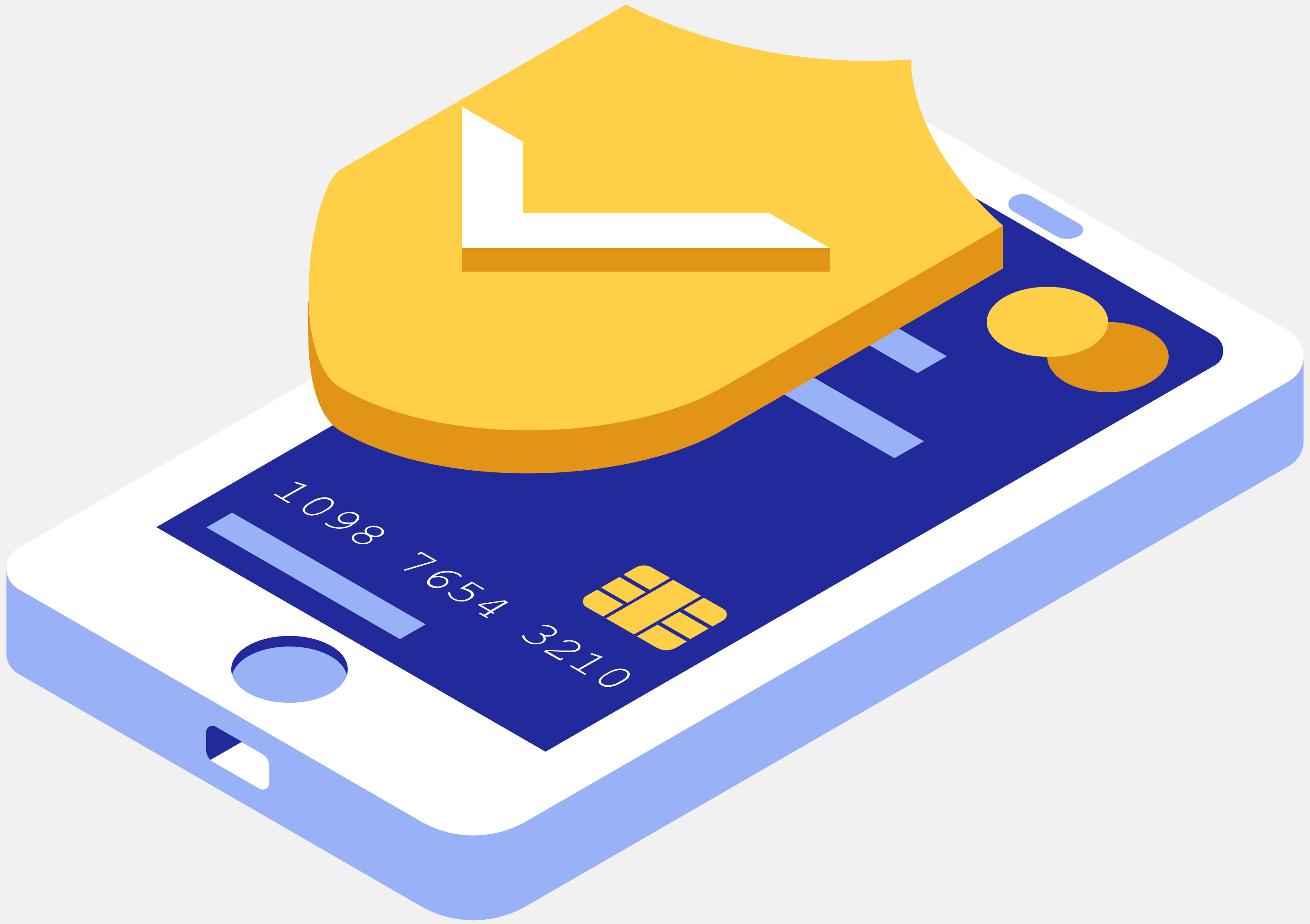
Introduction

Commercial banks receive numerous credit card applications.

- **What factors matter?** Age, income, credit score, credit history, etc.
- **Common reasons application gets denied:** high loan balances, low-income levels, or excessive inquiries on the applicant's credit report.
- Manual reviewing process can be tedious, prone to errors, and time-consuming.
- With Machine Learning, the **process can be automated**.

Objective

In this project, we will use a credit card dataset containing information such as individual’s total income, job title, family and marital status to **develop a machine learning model that predicts whether an applicant is a 'good' or 'bad' client for issuing a credit card.**



About Dataset

Contains **25128 rows** and **21 columns:**

#	Column	Non-Null Count	Dtype
---	------	--------------	-----
0	Applicant_ID	25128 non-null	int64
1	Applicant_Gender	25128 non-null	object
2	Owned_Car	25128 non-null	int64
3	Owned_Realty	25128 non-null	int64
4	Total_Children	25128 non-null	int64
5	Total_Income	25128 non-null	int64
6	Income_Type	25128 non-null	object
7	Education_Type	25128 non-null	object
8	Family_Status	25128 non-null	object
9	Housing_Type	25128 non-null	object
10	Owned_Mobile_Phone	25128 non-null	int64
11	Owned_Work_Phone	25128 non-null	int64
12	Owned_Phone	25128 non-null	int64
13	Owned_Email	25128 non-null	int64
14	Job_Title	25128 non-null	object
15	Total_Family_Members	25128 non-null	int64
16	Applicant_Age	25128 non-null	int64
17	Years_of_Working	25128 non-null	int64
18	Total_Bad_Debt	25128 non-null	int64
19	Total_Good_Debt	25128 non-null	int64
20	Status	25128 non-null	int64

Sourced from Kaggle

Why we picked it?

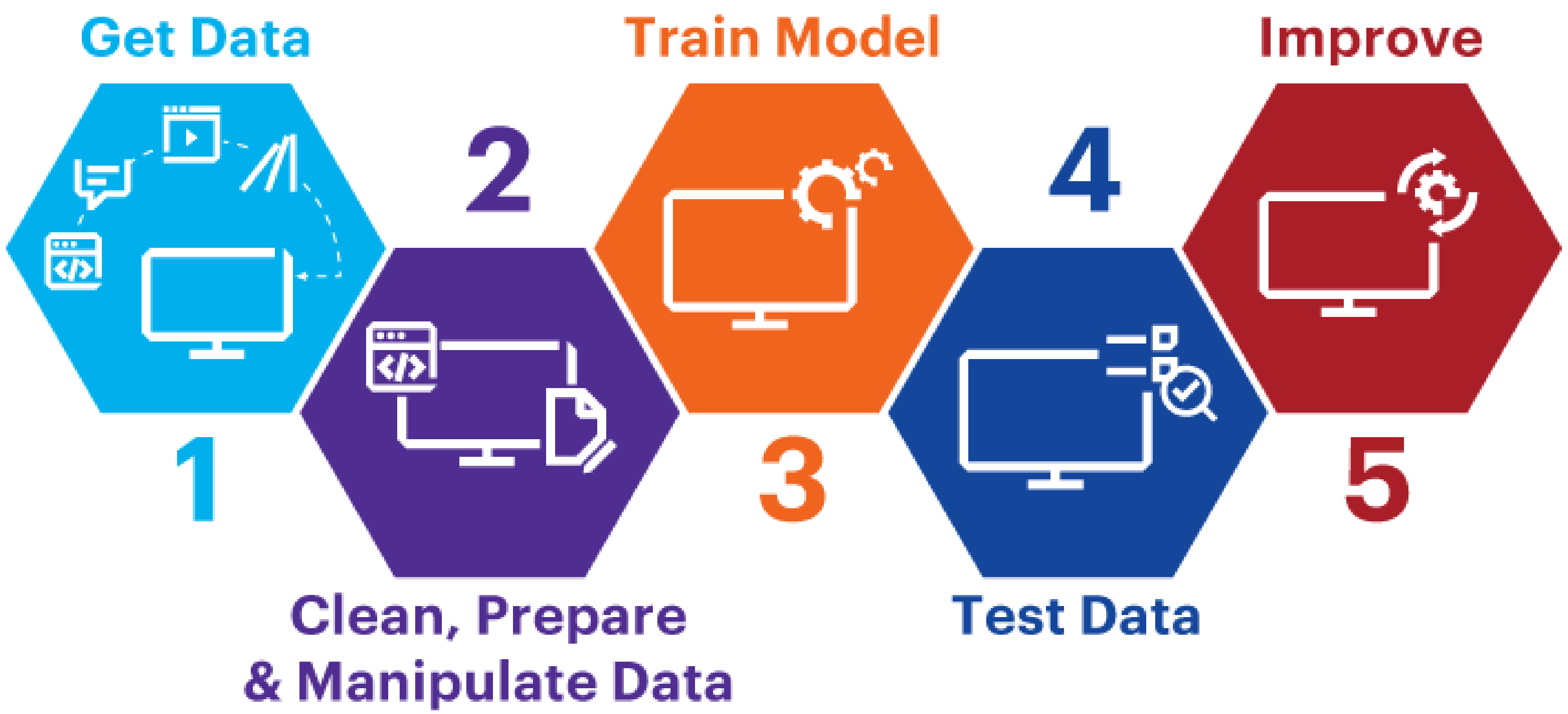
- **Importance of Credit Scoring:** Accurate credit scoring is vital for reducing default risks and improving lending decisions in the financial industry.
- **Imbalanced Data Challenges:** The dataset likely has more 'good' than 'bad' clients (or vice versa), needing techniques to avoid model bias.
- **Use of Machine Learning:** It involves using methods like correlation metrics, logistic regression, and random forest, offering practical experience with real-world data and binary classification models.

Literature Overview

A few studies have utilized this particular dataset and utilized different machine learning techniques to predict credit card approvals.

PyCaret	Light Gradient Boosting Machine
K-Means Clustering	Hyperparameters
Random Forest	AUC Curve
Naive Bayes	Confusion Matrix

Model Development Process

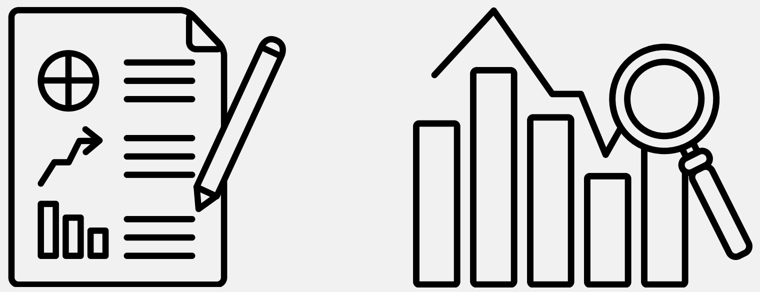


Aspects Explored

✓	Step 1	**Data Exploration** Inspected the dataset for missing values, duplicates, unique values, and the distribution of features.
✓	Step 2	**Data Preprocessing** Cleaned the dataset by handling outliers, converting categorical variables to numerical, and dropping constant or irrelevant features.
✓	Step 3	**Feature Engineering** Created new categorical features by grouping job titles into broader categories.
✓	Step 4	**Imbalanced Data Handling** Addressed class imbalance using Synthetic Minority Over-sampling Technique (SMOTE).
✓	Step 5	**Model Building and Evaluation** Built and evaluated logistic regression and random forest models to classify the target variable.

Techniques Used

Performed Exploratory Data Analysis and Data Cleaning



Identifying Outliers Using Box Plot

Utilized **box plots** to detect outliers and filtered out extreme values using **percentile-based benchmarks**.

One-Hot Encoding

Converted categorical variables into numerical using **pd.get_dummies**.

Class Imbalance Handling

Applied **SMOTE** to balance the classes in the target variable during the training phase.

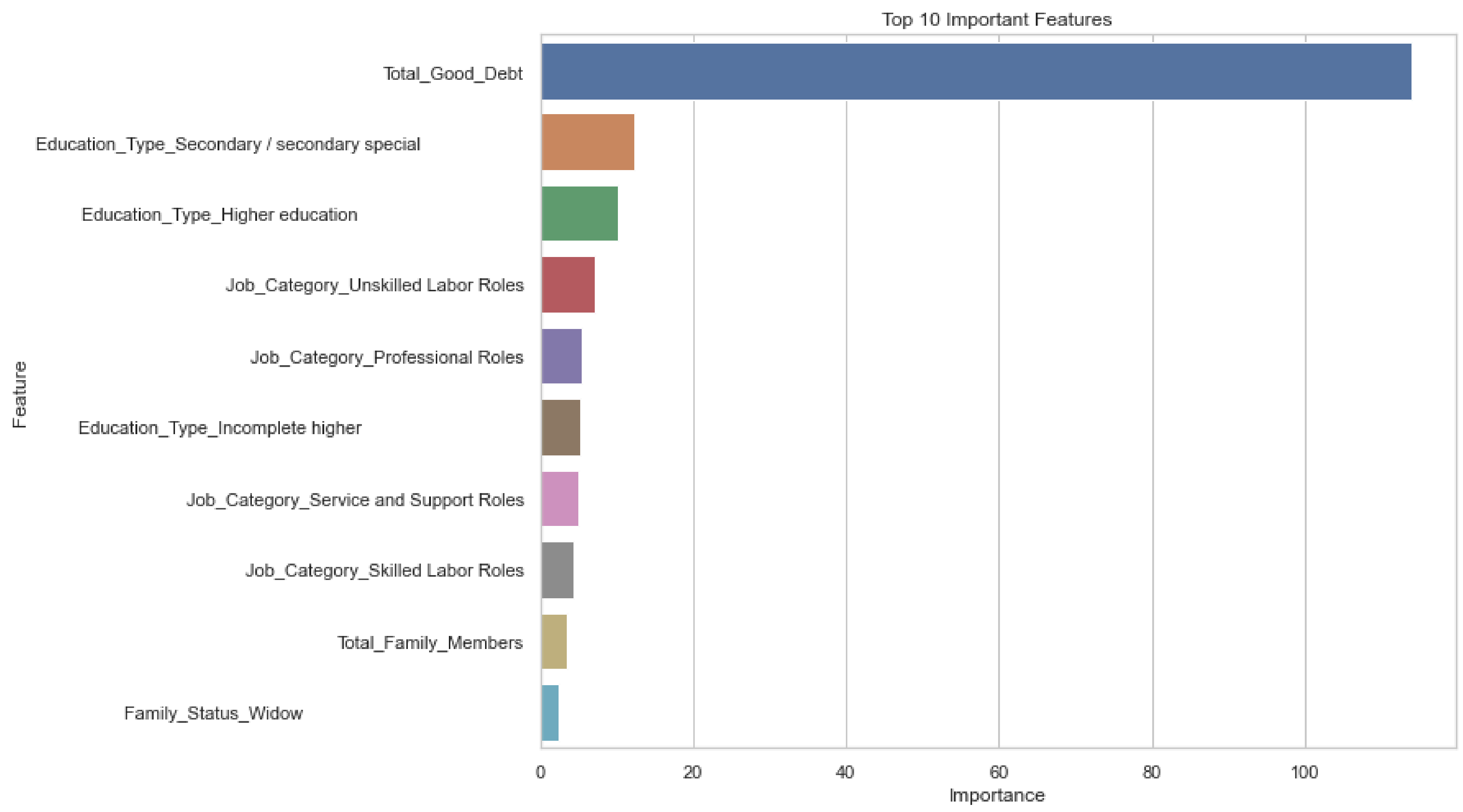
Modeling Techniques

Logistic Regression: Implemented to find the best hyperparameters.
Random Forest Classifier: Tuned to optimize model performance.

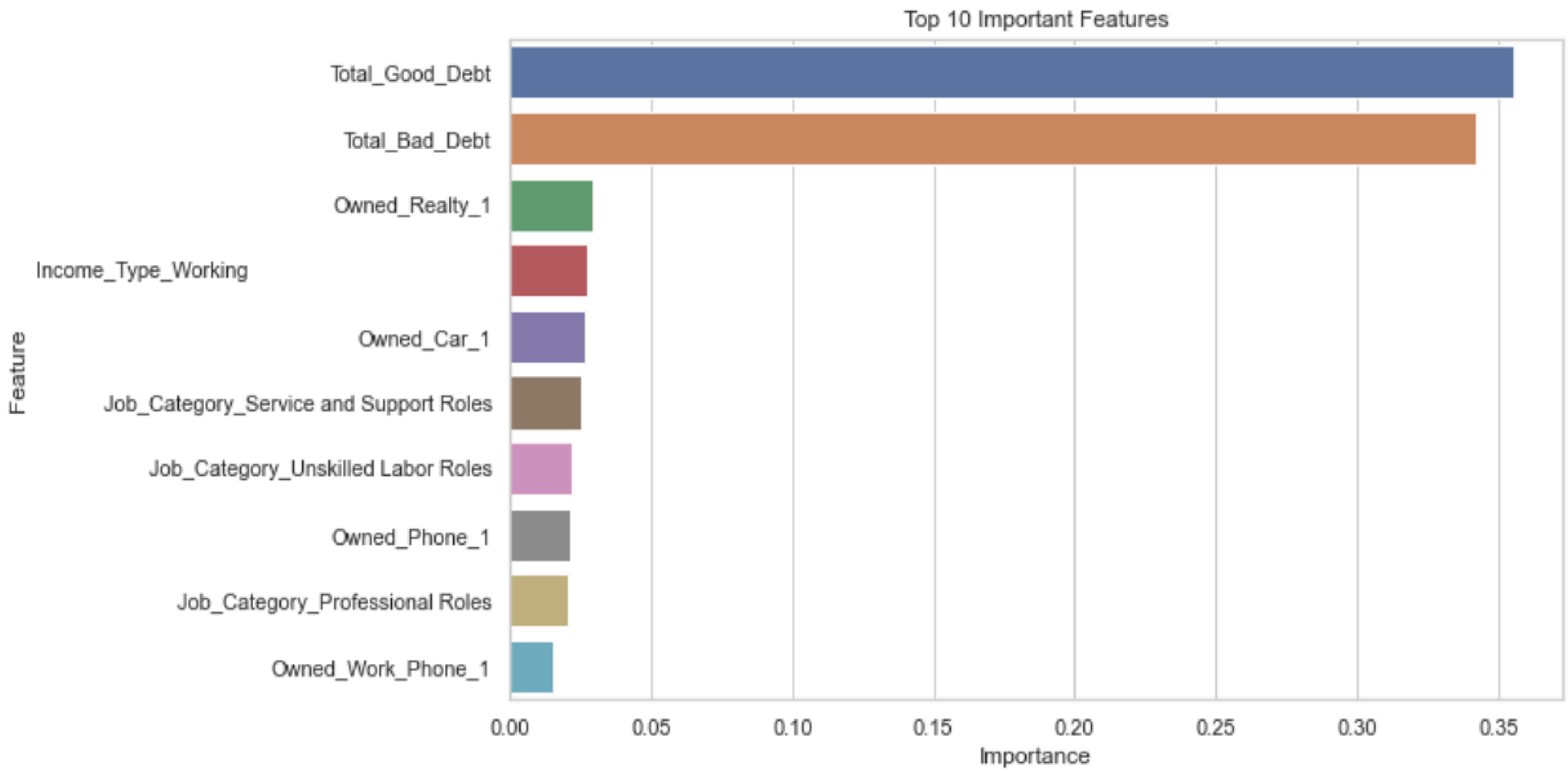
Logistic Regression

Defined the pipeline with standard scaling and logistic regression. Fitted GridSearchCV object on resampled training data.

- **Best Hyperparameters:**
  - C: 44668.35921509626
  - class_weight: 'balanced'
  - max_iter: 100
  - penalty: 'l2'
  - solver: 'liblinear'
- **F1-Score on Test Set:** 1.0
- **Precision:** 1.0
- **Recall:** 1.0
- **Accuracy:** 1.0



Top 10 Important Features



Top 10 Important Features

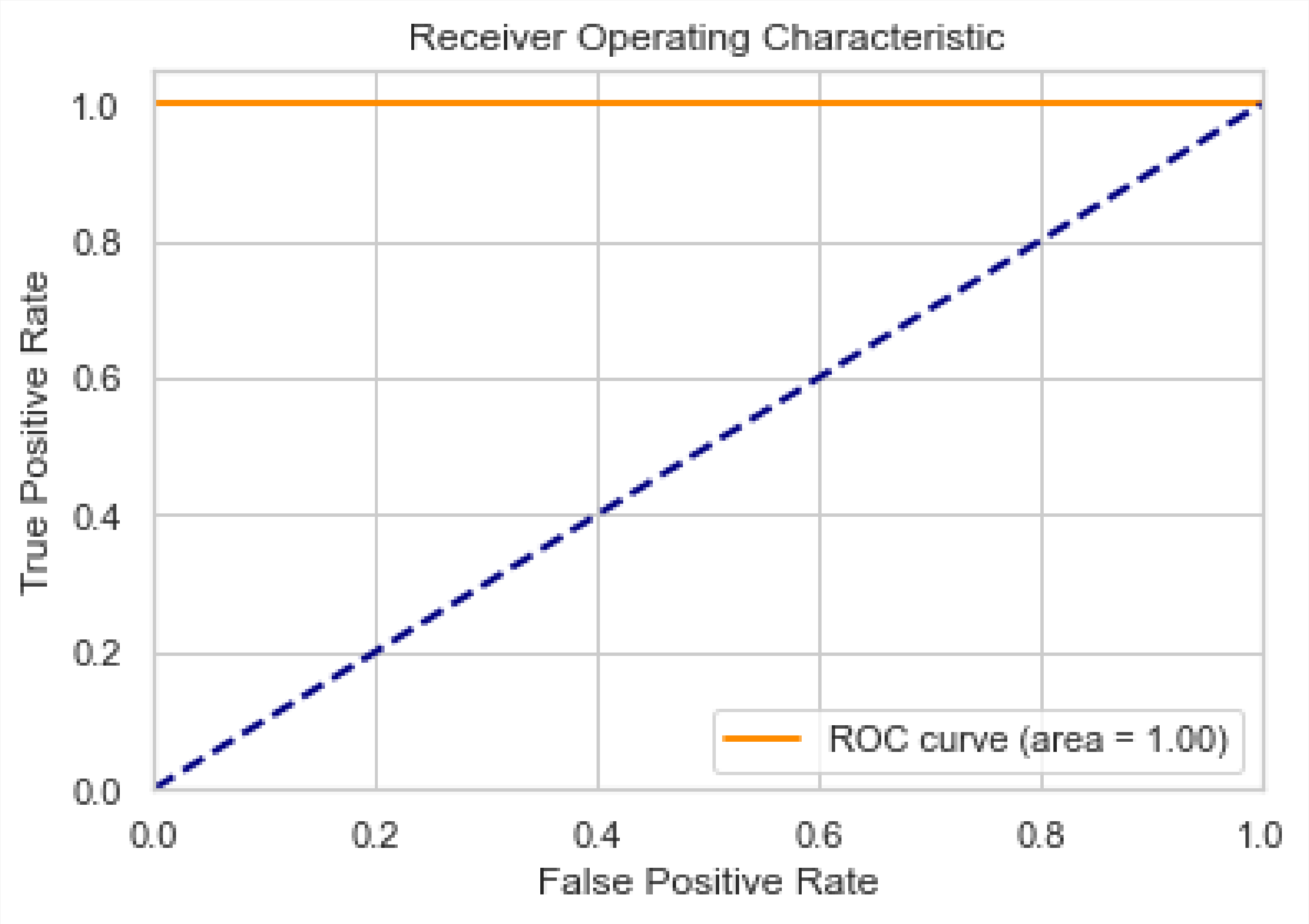
Random Forest

Utilized Random Forest Classifier to extract the best hyperparameters and GridSearchCV to find the best estimator.

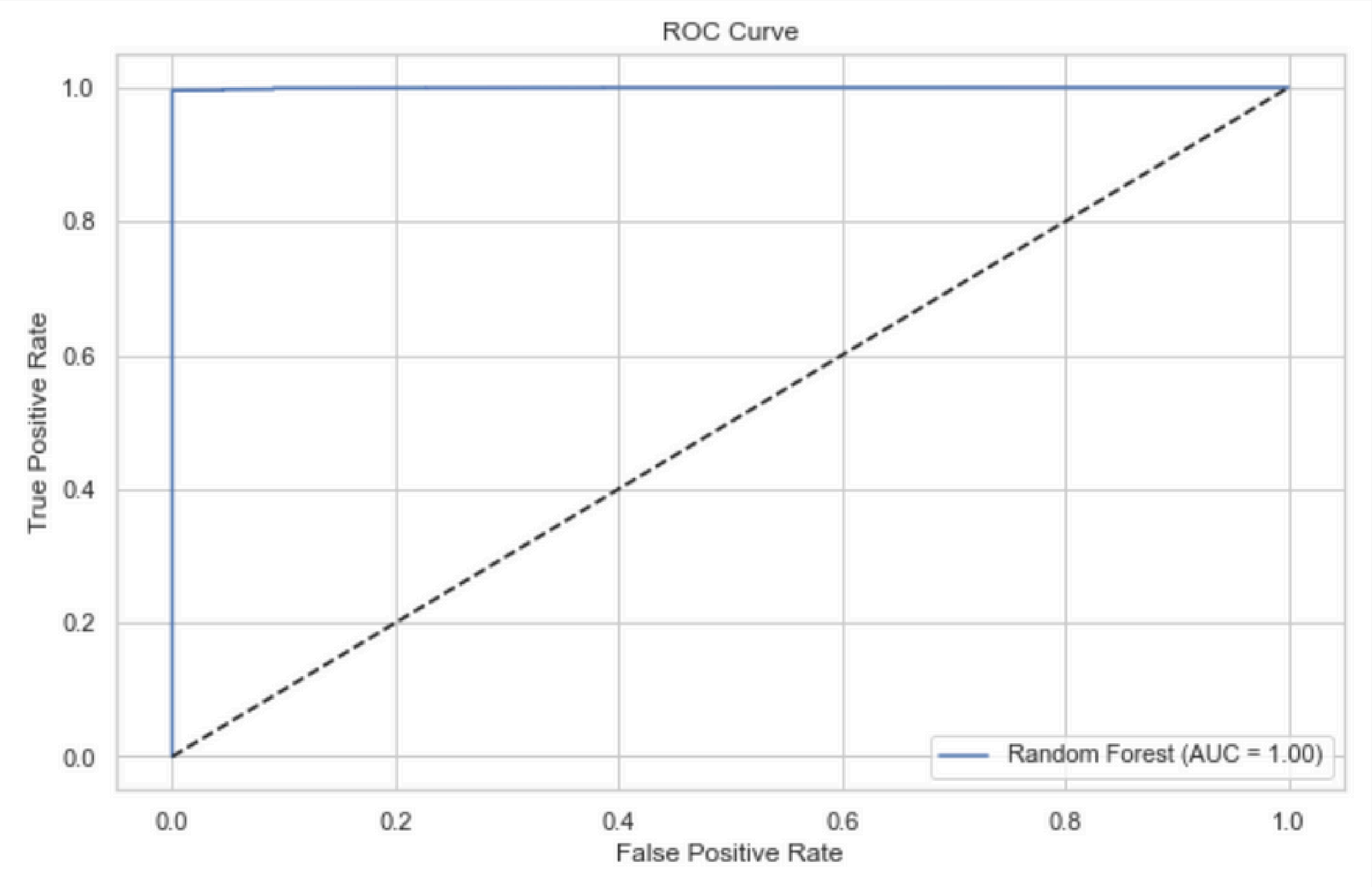
- **Best Hyperparameters:**
  - max_depth: 20
  - max_features: 'sqrt'
  - min_samples_split: 2
  - n_estimators: 200
- **F1-Score on Test Set:** 1.0
- **Precision:** 1.0
- **Recall:** 1.0
- **Accuracy:** 1.0

ROC Curves

Logistic Regression



Random Forest



- The ROC curves for both models showed excellent separation, indicating **high model performance**.
- Confusion Matrix for both models showed **high true positive rates and low false positive rates.**

Learnings

- The exploration and preprocessing stages were crucial in identifying and handling data imbalance and feature importance.
- Both models performed exceedingly well, with Logistic regression & Random Forest achieving a perfect score across all metrics on the test set.
- However, the class imbalance presented challenges in interpreting these results, suggesting that more balanced datasets could be beneficial.

What else would we have done?

- Try and test other methods for handling data imbalance.
- Try other machine learning models on our dataset.
- Conduct more in depth feature engineering.

Thank you!

Do you have any questions?