Credit Card Approval Prediction Report

1. Business Understanding

1.1 Introduction

In the financial industry, credit score cards are widely used to evaluate the risk associated with issuing credit cards to applicants. These cards use historical data to anticipate potential defaults and credit card borrowing patterns, enabling banks to make well-informed decisions regarding card approval. However, the reliability of traditional approaches, such as assessing financial metrics like debt-to-income ratios and utilization ratios, can be impacted by economic fluctuations. To enhance credit scoring, alternative methods like machine learning algorithms have been introduced. Although these methods enhance prediction accuracy, they often lack transparency, creating challenges in providing clear explanations for acceptance or rejection decisions to both customers and regulatory bodies.

1.2 Problem Statement

The existing credit scoring methods rely on historical data and traditional financial metrics like debt-to-income ratios, which limits their ability to accurately predict creditworthiness in the face of economic fluctuations. The introduction of machine learning algorithms has shown promise in enhancing credit scoring accuracy but lacks transparency in explaining acceptance or rejection decisions. As a result, there is a need to develop a credit scoring model that leverages both historical data and advanced machine learning techniques to improve prediction capabilities while providing transparent and interpretable results for customers and regulators. The objective is to create a reliable and transparent credit scoring system that effectively assesses the risk of issuing credit cards based on applicants' personal information, contributing to informed decision-making by financial institutions and ensuring fair treatment of applicants.

1.3 Project Objectives

The project objectives are:

- Develop a credit scoring model that incorporates personal and social factors and machine learning algorithms to enhance the accuracy of creditworthiness predictions.

- Improve transparency in credit scoring by utilizing interpretable machine learning techniques, allowing for clear explanations of acceptance or rejection decisions to customers and regulatory bodies.
- Mitigate the impact of economic fluctuations on credit scoring models by incorporating dynamic factors and adapting the model to changing economic conditions.
- Optimize the balance between prediction accuracy and interpretability to ensure a fair and reliable credit scoring system for both applicants and financial institutions.

2. Data Understanding

2.1 Collect Initial Data

The data for this project was sourced from Kaggle . This data consists of information of past applications and credits data.

2.2 Describe Data

There are two datasets,the Applications Dataset and the Credits Dataset. The Applications dataset has 438557 rows and 18 columns. The Credits dataset has 1048575 rows and 3 columns.

There are also social and financial factors. The dataset consists of numerical and categorical data.

The column features are described as:

- ID: Client number
- CODE_GENDER: Gender
- FLAG_OWN_CAR: Is there a car
- FLAG_OWN_REALTY: Is there a property
- CNT_CHILDREN: Number of children
- AMT_INCOME_TOTAL: Annual income
- NAME_INCOME_TYPE: Income category
- NAME_EDUCATION_TYPE: Education level
- NAME_FAMILY_STATUS: Marital status
- NAME_HOUSING_TYPE: Way of living

- DAYS_BIRTH: Birthday
- DAYS_EMPLOYED: Start date of employment
- FLAG_MOBIL: Is there a mobile phone
- FLAG_WORK_PHONE: Is there a work phone
- FLAG_PHONE: Is there a phone
- FLAG_EMAIL: Is there an email
- OCCUPATION_TYPE: Occupation
- CNT_FAM_MEMBERS: Family size

We merged the datasets on the ID column and the resulting data frame had 777715 rows and 20 columns.

3. Data Preparation

3.1 Clean Data

We will be looking at the missing values, duplicate records in the dataset as well as the outliers in the dataset.

3.1.1 Completeness

We have missing values in the `Occupation_Type` column.The `Occupation_Type` column with missing values exhibits a diverse range of income types. Since the missing values account for approximately 31 percent of the column, we cannot make assumptions about the income types associated with these occupations. Consequently, we will impute the NaN values in the column with the term **unspecified** to handle the missing data appropriately.

3.1.2 Uniformity

Labeling - We renamed the columns in the dataset to a format that is more interpretable. Eg CODE_GENDER to gender,CNT_CHILDREN to Num_Children and FLAG_MOBIL to Mobile.

Formatting - Here we check the datatype of each column.The Num_Family has a float Data type so we shall convert it to integer data type. The column age(days) is converted into age(years).

3.1.3 Validity

We will be checking for duplicated data and outliers.

Duplicates - The data frame had no duplicates.

Outliers - Although the dataset contains outliers, we will proceed with exploratory data analysis (EDA) to gain deeper insights into the data. By conducting EDA, we aim to uncover patterns, relationships, and underlying trends that can help us understand the data better.

We are not going to drop the outliers in the columns Num_Children and Num_family since it's hard to determine the number of children a family can have and the family size a client can have.Also for case of total income its dense to determine the total income of a client since the occupation differs from one client to another.

3.2 Exploratory Data Analysis

We shall be performing the following:- Univariate Analysis
- Bivariate Analysis
- Multivariate Analysis

In this section, we'll explore each column in the dataset to see the distributions of features and obtain some useful insights.

3.2.1 Univariate Analysis

Status:
We are plotting counts starting with status distribution. We found that the value "C" was the most common status, accounting for almost half of the total counts. The next most common value was "X", representing 26.5% of the total counts. The value "0" had a count of 8456, making up 23.2% of the total. The other values, "1", "5", "2", "3", and "4", each represented a small percentage of the total count.

Gender:
The majority of individuals in the dataset are female (67.05%), while males account for the remaining 32.95%.

Occupation:
The most common occupation in the dataset is "Unspecified" (30.95%), followed by "Laborers" (16.97%), "Core staff" (9.82%), "Sales staff" (9.52%), "Managers" (8.24%), and "Drivers" (5.85%).

Income Type:
The most common income type is "Working" (51.32%), followed by "Commercial associate" (23.11%), and "Pensioner" (16.76%).

Education Level:
"Secondary / secondary special" is the most common education level (66.17%), followed by "Higher education" (26.28%), and "Incomplete higher" (3.77%).

Housing Type:
Most individuals in the dataset live in a "House / apartment" (91.29%), while a smaller percentage live with their parents (4.99%), in a municipal apartment (3.17%), or in a rented apartment (1.62%).

Family Status:
The most common family status is "Married" (70.34%), followed by "Single / not married" (13.56%), and "Civil marriage" (8.26%).

Number of Children:
The majority of individuals in the dataset do not have any children (67.03%), while some have one child (19.88%) or two children (8.64%). There are a few outliers with larger numbers of children (e.g., 14, 19).

Total Income:
The distribution of total income is skewed to the right, with most individuals having a total income of less than 500,000. The distribution appears to be unimodal, with the highest peak around 100,000 total income. There are a few outliers with very high total incomes (greater than 4,000,000).

3.2.2 Bivariate Analysis

Bivariate Analysis examines the relationship between two variables in a dataset.We are evaluating the relationship between the features columns, including our target column, **"Status."**

Status and Income Type:
`Working` and `Commercial Associate` categories have a higher proportion of loan approvals.

Status and Property Ownership:
Individuals who own property have a higher proportion of loan approvals compared to those who do not own property.

Status and Gender:
Females have a higher proportion of loan approvals compared to Males.

3.2.3 Multivariate Analysis

The most correlated features are `Num_family` and `Num_children` with a value of 0.89.This indicates that the number of children is strongly related to the number of family members.

`Employment_duration` and `Age(years)` with a value of 0.61.This suggests that as the duration of employment increases, the age of the individual also tends to increase.

`Age(days)` and `Age(years)` with a value of -1.This is expected, as "Age(days)" is a more granular measure of age than "Age(years)", and the two variables measure the same underlying construct.

Overall, the heatmap provides valuable insights into the relationship between the numeric columns in the dataset. The correlated features can be used to create more accurate predictive models, while the uncorrelated features can be removed to simplify the model and reduce the risk of overfitting.

3.3 Data Preprocessing

/

3.3.1 Feature Engineering

We start with some data preprocessing steps, such as encoding the 'Status' column to a more readable format using a mapping dictionary.

It then creates a new DataFrame called 'credit_record' to view the statuses of various individuals based on the 'ID' and 'Status' columns.
Duplicate rows based on the 'ID' column are dropped from the 'credit_record' DataFrame.
Unnecessary columns such as 'ID', 'age(days)', 'Mobile', 'Num_Children', and 'Recorded_date(months)' are dropped from the 'Final_data' DataFrame.
We create a new feature called 'Income_Range' by categorizing the 'Total_Income' column into specific income ranges.
The 'Income_Range' values are mapped into a desired format for better readability.
Redundant columns are dropped from the 'model_df' DataFrame, which include 'Total_Income', 'Status', 'Work_Phone', and 'Phone'.
The 'model_df' DataFrame is sorted based on the 'CREDIT_APPROVAL_STATUS' column.

The income range is created in increments of 27000 and individuals' total income is categorized based on the ranges. The Label Encoder and StandardScaler are used to encode and standardize some of the features.

3.3.2 Feature selection

The 'X' and 'y' variables are defined for feature selection, where 'y' represents the target variable ('CREDIT_APPROVAL_STATUS') and 'X' represents the feature matrix.
This is performed using the recursive feature elimination method, which selects 11 features, including Own_Car, Own_Property, Income_Type, Income_Range, Education_Level, Family_Status, Housing_Type, Employment_Duration, Occupation, Num_Family, and age(years).
The function splits the data into training and testing sets, applies RFE, selects the features, and evaluates the accuracy of the model.
The selected features and their corresponding accuracy are printed.
A list of columns to include in the feature matrix ('X') is defined based on the selected features.
The 'X' and 'y' variables are updated to include only the selected columns.
Overall, the code demonstrates how data preprocessing, feature engineering, and feature selection are performed to prepare the data for machine learning modeling.

4. Modeling

In the modeling phase, we will address the issue of class imbalance before training our models. We will start with a decision tree as our baseline model, which provides a good understanding of feature-target relationships. Then, we will explore ensemble models such as Random Forest, AdaBoost, Gradient Boosting, and XGBoost. Random Forest combines multiple decision trees to improve performance and reduce overfitting. AdaBoost creates a strong learner by iteratively adjusting weights based on misclassified instances. Gradient Boosting builds models sequentially to minimize prediction errors. XGBoost is an optimized implementation of gradient boosting known for its scalability and performance. By considering these models, we aim to build robust and accurate predictive models for our dataset.
Here's a summary of the models and their performance:

- **Decision Tree Classifier (Baseline Model):**
  Test Accuracy: 0.786
  Recall Score (Class 0): 0.40
  Recall Score (Class 1): 0.85

- **Random Forest Classifier:**
  Test Accuracy: 0.778
  Recall Score (Class 0): 0.37

- **Recall Score (Class 1): 0.84**
  Best Parameters:n_estimators: 100
  criterion: 'entropy'
  max_depth: None
  min_samples_split: 8
  min_samples_leaf: 5
- **AdaBoostClassifier:**
  Test Accuracy: 0.571
  Recall Score (Class 0): 0.45
  Recall Score (Class 1): 0.59
  Best Parameters:learning_rate: 0.5
  n_estimators: 100
- **Gradient Boosting Classifier:**
  Test Accuracy: 0.737
  Recall Score (Class 0): 0.23
  Recall Score (Class 1): 0.82
  Best Parameters:learning_rate: 0.3
  loss: 'log_loss'
  n_estimators: 99
- **XGB Classifier:**
  Test Accuracy: 0.812
  Recall Score (Class 0): 0.25
  Recall Score (Class 1): 0.90
  Best Parameters:learning_rate: 0.5
  max_depth: 5
  n_estimators: 150

It's important to note that these results are specific to the provided dataset and the chosen parameters. The model performances may vary depending on the specific problem and dataset characteristics. The models were evaluated based on accuracy, recall scores, and cross-validation accuracy to assess their performance on unseen data.

Additionally, learning curves were plotted for each model to visualize the relationship between training set size and the model's performance. These curves provide insights into the model's ability to generalize and detect potential underfitting or overfitting.

5. Evaluation

We have evaluated several classification models for predicting credit card approval status. Here is a summary of the models, their rationale, results, and limitations:

- **Decision Tree Classifier:**
  Rationale: Chosen as the baseline model due to its interpretability and robustness to outliers.
  Results: Training Accuracy: 0.870, Testing Accuracy: 0.789, Recall Score: 0.85
  Limitations: Tendency to overfit training data.
- **Random Forest Classifier:**
  Rationale: Chosen to overcome the limitations of individual decision trees by combining multiple trees.
  Results: Training Accuracy: 0.835, Testing Accuracy: 0.776, Recall Score: 0.84
  Limitations: Requires more computational resources and training time.
- **AdaBoost:**
  Rationale: Chosen for its ability to handle complex relationships and interactions between variables.
  Results: Training Accuracy: 0.571, Testing Accuracy: 0.571, Recall Score: 0.59
  Limitations: Interpretability may be compromised due to the ensemble nature.
- **Gradient Boosting:**
  Rationale: Chosen for its ability to handle complex relationships and deliver high predictive performance.
  Results: Training Accuracy: 0.748, Testing Accuracy: 0.737, Recall Score: 0.82
  Limitations: Potential susceptibility to overfitting, higher computational complexity.
- **XGBoost (Extreme Gradient Boosting):**
  Rationale: Chosen for its outstanding performance, scalability, and advanced regularization capabilities.
  Results: Training Accuracy: 0.841, Testing Accuracy: 0.812, Recall Score: 0.90
  Limitations: Requires careful hyperparameter tuning, higher computational complexity.

Based on the information, the XGBoost model achieved the highest recall score of 0.90 and overall better performance in terms of training and testing accuracy compared to other models. However, it requires careful tuning of hyperparameters and has higher computational complexity.

To incorporate SHAP (SHapley Additive exPlanations) values for recommendation purposes, we have created a function compute_shap_values() to compute and visualize the importance of features. We have also demonstrated how to use the function to analyze a specific instance's SHAP values.

Furthermore, you have created a prediction function that takes in data and makes predictions using the final trained model. If the predicted result is "Approved" (1), it displays the SHAP values for the given data, providing insights into the reasons for approval or denial.
Lastly, we have shown an example of encoding new data using the LabelEncoder and pickling (saved) the necessary preprocessing steps and the final trained model for future use.
Overall, we have performed model evaluation, selected the best model, and incorporated SHAP values for interpretability and recommendation purposes.

6. Deployment