Credit Card Default Prediction

Satyam Sharma

Data Scientist

Intern at PW skills

codexistslonglastingnotfog@gmail.com

Abstract---The Credit Card Default Prediction System is an innovative tool designed for organizations, particularly banks, to enhance their credit card approval processes. Powered by cutting-edge machine learning algorithms, this system analyses an applicant's financial history and risk factors to accurately predict the likelihood of credit card defaults. Its seamless integration with various software technologies ensures accessibility for a wide range of organizations. The primary aim of this project is to empower financial institutions with data-driven insights, enabling them to make informed decisions while minimizing the risk of credit card defaults. By leveraging historical data, this system not only streamlines credit approval but also contributes to responsible lending practices. Financial stability and risk management are bolstered as a result. In summary, the Credit Card Default Prediction System represents a significant advancement in the financial industry, facilitating efficient credit risk assessment and promoting a more secure lending environment for both financial institutions and customers.

INTRODUCTION

In the dynamic landscape of modern finance, the issuance of credit cards is a ubiquitous element of economic growth and personal financial management. However, this convenience also presents financial institutions, especially banks, with the formidable challenge of effectively assessing and mitigating credit risk. The rise in credit card defaults underscores the critical importance of accurate risk assessment. To address this challenge, our project introduces a groundbreaking Credit Card Default Prediction System that seamlessly integrates into existing software technologies, providing a proactive solution for organizations, with a focus on banks.

This project's core mission is to empower financial institutions with a robust predictive tool, driven by advanced machine learning algorithms, to enable a more precise evaluation of an applicant's creditworthiness. By employing techniques such as Linear Regression and Decision Trees, this system is capable of analyzing historical financial data and relevant risk factors to predict the probability of a customer defaulting on their credit card payments. This, in turn, equips financial institutions with the data-driven insights required to make well-informed decisions when issuing credit cards, thus reducing potential financial risks associated with credit defaults and enhancing overall risk management.

At the heart of this project lies the strategic application of feature engineering techniques. Through careful observation and analysis of data, we identify and extract the most relevant and informative features. These features not only enhance the prediction accuracy but also provide a deeper understanding of the underlying factors influencing credit card defaults.

The evolution of this technology marks a significant milestone in the financial industry, offering a more efficient and data-driven approach to credit risk assessment. The objective is not only to streamline credit approval processes but also to promote responsible lending practices that benefit both financial institutions and their customers. Furthermore, the adaptability of this system to various software technologies ensures accessibility for organizations across diverse sectors, fostering a more secure and informed lending environment.

In the forthcoming sections, we will delve deeper into the methodology, techniques employed, features extracted, and the potential implications of the Credit Card Default Prediction System. This project is a testament to the power of technology to reshape financial practices and safeguard the financial stability of institutions while ensuring responsible lending in a data-rich era.

I. TECHNOLOGY USED

The project leveraged a range of technologies to create a robust and efficient workflow for credit card default prediction:

Python: The core programming language used for data analysis and machine learning model development.

Jupyter Notebooks (.ipynb) on Visual Studio Code (VSCode): This combination provided a collaborative and versatile environment for code development, data exploration, and documentation. Jupyter Notebooks offer an interactive space for data analysis and model development, while VSCode's integrated features enhance the coding experience.

Virtual Environments: Virtual Python environments were employed to manage dependencies and ensure a consistent working environment across different systems. This practice enhances reproducibility and version control.

Version Control (Git): Git, along with platforms like GitHub or GitLab, facilitated version control, enabling team members to work concurrently, manage code changes, and track project progress effectively.

Documentation Tools: To maintain comprehensive project documentation, Markdown and reStructuredText were used within Jupyter Notebooks. Additionally, Sphinx documentation tools were employed to generate detailed project documentation, ensuring clarity and transparency.

1.1 Library Used

The following libraries played a crucial role in the project:

NumPy: NumPy, short for Numerical Python, is fundamental for efficient numerical computations and data manipulation. It provides support for multi-dimensional arrays and mathematical functions, making it a cornerstone of data analysis.

Pandas: Pandas is a versatile library for data manipulation, offering powerful data structures for handling structured data, including DataFrames and Series. It simplifies data cleaning, transformation, and exploration.

Scikit-Learn: This library is integral for machine learning model development and evaluation, offering a wide range of algorithms and tools for predictive modeling.

Matplotlib and Seaborn: These libraries were used for data visualization, aiding in the understanding and communication of data insights.

XGBoost and RandomForest: These machine learning libraries were employed for advanced predictive modeling and improved model accuracy.

II. LITERATURE SURVEY

The foundation of the Credit Card Default Prediction System is rooted in extensive research and a comprehensive review of existing literature on credit risk assessment, predictive modeling, and data-driven solutions. This section provides an overview of the key findings from the literature survey, outlining the insights and methodologies that have informed our project.

Credit Risk Assessment: A central theme in the literature is the assessment of credit risk. Various studies have explored the significance of accurate risk assessment in financial institutions, emphasizing the need for predictive models that can reliably identify potential defaulters.

Predictive Modeling Techniques: The literature review revealed a range of predictive modeling techniques applied in credit risk assessment. Notably, techniques such as Logistic Regression, Decision Trees, Random Forest, Support Vector Machines, and Gradient Boosting have been employed to develop accurate models for predicting credit card defaults.

Feature Engineering: Feature engineering emerged as a critical aspect of credit risk prediction. Researchers have highlighted the importance of selecting and transforming relevant features, which directly impact the predictive power of the models. This includes considering factors such as an applicant's credit history, income, debt-to-income ratio, and employment status.

Data Sources: Literature has emphasized the significance of reliable and diverse data sources for model training. Historical credit card data, socio-economic indicators, and macroeconomic factors have been commonly used to build predictive models.

Evaluation Metrics: To assess the performance of predictive models, established evaluation metrics such as accuracy, precision, recall, F1-score, and ROC-AUC have been employed in the reviewed studies. These metrics help measure the effectiveness of models in terms of both identifying defaulters and minimizing false positives.

Challenges and Limitations: The literature has identified several challenges and limitations in credit risk assessment, including data quality, class imbalance, and the potential impact of economic fluctuations. Researchers have proposed various strategies to address these challenges.

Regulatory Frameworks: Compliance with regulatory requirements is a critical consideration in credit risk assessment. The literature survey revealed the importance of developing models that align with regulatory frameworks to ensure responsible lending practices.

By conducting a thorough literature survey, our project benefited from the valuable insights and methodologies outlined in prior research. This foundation allowed us to adopt the most effective techniques and best practices to create a reliable and robust Credit Card Default Prediction System.

III. DATASET AND FEATURE DESCRIPTION

The dataset used in this project comprises a comprehensive set of features that serve as inputs for the Credit Card Default Prediction System. Each feature holds valuable information about the credit card applicants and their repayment behavior. Below is an overview of the dataset's features and their descriptions:

- ID: A unique identifier for each client.
- LIMIT_BAL: The amount of credit provided in New Taiwan (NT) dollars, including individual and supplementary credit.
- SEX: Gender of the applicant.
  - 1: Male
  - 2: Female

- EDUCATION: Educational background of the applicant.
    - 1: Graduate School
    - 2: University
    - 3: High School
    - 4: Others
    - 5: Unknown
- MARRIAGE: Marital status of the applicant.
    - 1: Married
    - 2: Single
    - 3: Others
- AGE: The age of the client in years.

- PAY_0 to PAY_6: Repayment status for the respective months, from September (PAY_0) to April (PAY_6).
    - -2: No consumption
    - -1: Pay duly
    - 0: Use of revolving credit
    - 1: Payment delay one month
    - 2: Payment delay for two months
    - 3: Payment delay for three months
    - 8: Payment delay for eight months
    - 9: Payment delay for nine months or above
- BILL_AMT1 to BILL_AMT6: Bill statement amounts for the respective months, from September (BILL_AMT1) to April (BILL_AMT6).

- PAY_AMT1 to PAY_AMT6: Amount of the previous payment for the respective months, from September (PAY_AMT1) to April (PAY_AMT6).

- default.payment.next.month: The target variable indicating whether the client defaulted on the payment next month.

- 1: Defaulted (Yes)
- 0: Not Defaulted (No)

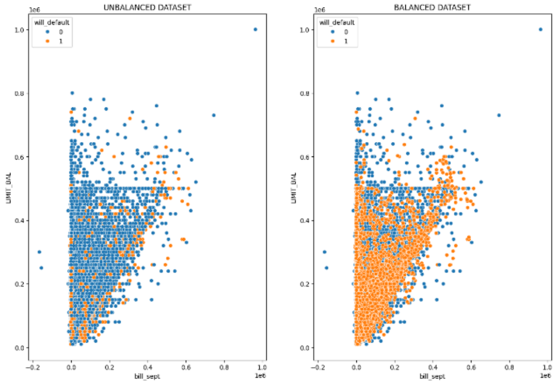
This dataset forms the basis of our Credit Card Default Prediction System. Each feature contains valuable information that will be used to train predictive models and make data-driven decisions on credit card approvals.

IV. EXPLORATORY DATA ANALYSIS

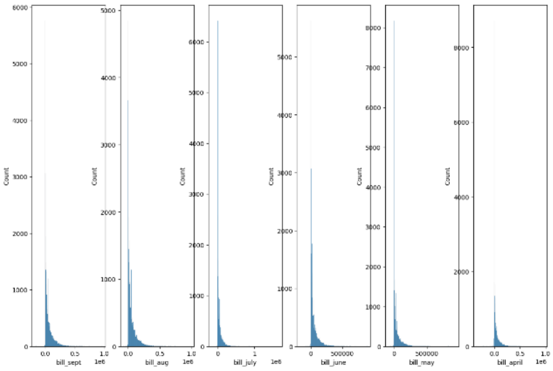
Exploratory Data Analysis (EDA) is an essential phase in understanding the credit card dataset, uncovering patterns, and gaining insights. In this section, we'll conduct a thorough EDA, including univariate and bivariate analyses, to explore and visualize the data. The goal is to unveil trends, anomalies, and relationships within the dataset, which will inform subsequent steps in our credit card default prediction process.

4.1 Univariate Analysis

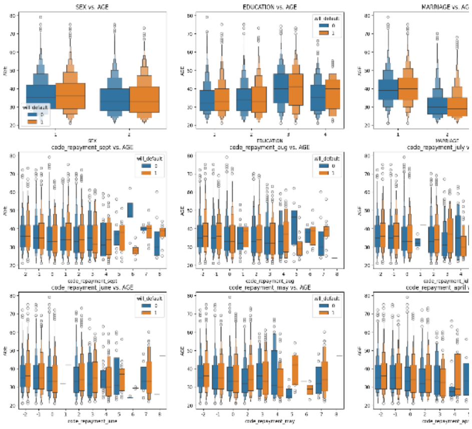
Univariate analysis focuses on understanding individual features in isolation, providing insights into their distributions, central tendencies, and variability. We employ various techniques to gain a comprehensive understanding of the dataset:



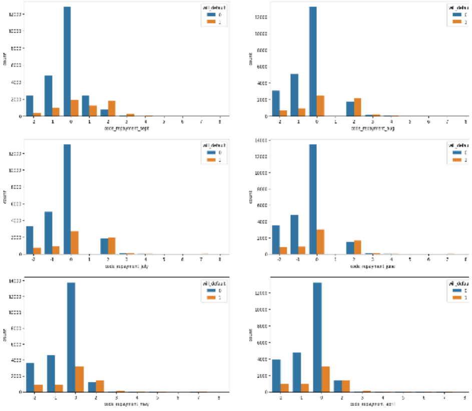
4.1.1 Histogram Plots: Histograms provide visual representations of the data's distribution, allowing us to identify patterns and outliers within numeric features.



4.1.2 Box Plots: Box plots reveal the central tendency and spread of numeric variables while highlighting potential outliers and their distribution.



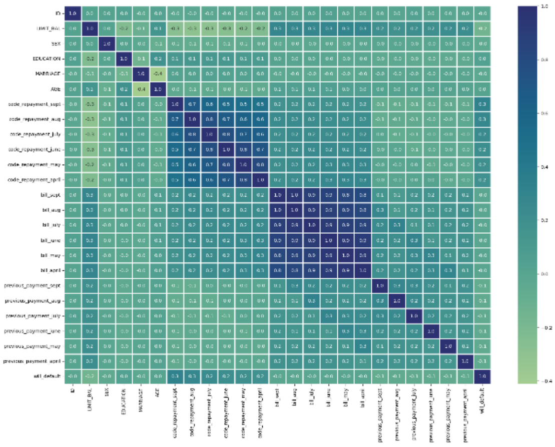
4.1.3 Count Plots: For categorical features, count plots help visualize the distribution of categories, offering insights into the frequency of different classes.



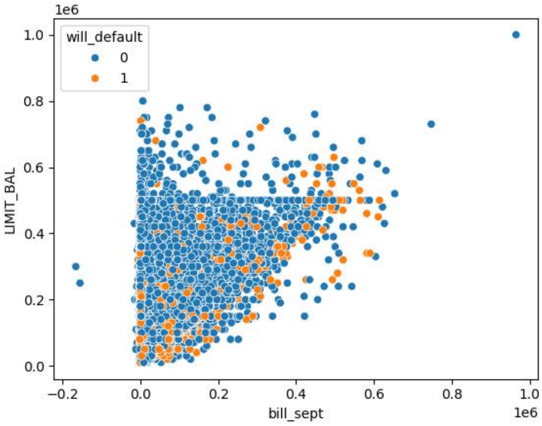
4.2 Bivariate Analysis

Bivariate analysis, on the other hand, explores relationships between pairs of features. This allows us to identify correlations and dependencies within the dataset, providing a deeper understanding of how variables interact:

4.2.1 Correlation Matrix: We compute a correlation matrix to quantify the strength and direction of linear relationships between pairs of numeric features. This matrix helps identify potential multicollinearity and guides variable selection.



4.2.2 Scatter Plots: Scatter plots are employed to visualize relationships between two numeric variables, aiding in the identification of potential trends, patterns, or outliers.



4.2.3 Categorical vs. Numeric Plots: Comparing categorical variables against numeric ones helps uncover differences in distributions and central tendencies across different categories

By conducting a comprehensive EDA with univariate and bivariate analyses, we will gain a clear understanding of the dataset's characteristics, relationships, and key factors influencing credit card default behavior. These insights will be crucial in shaping our data preprocessing and model development efforts.

V. FEATURE ENGINEERING

Feature engineering is a pivotal step in shaping the dataset to improve the predictive power of our Credit Card Default Prediction System. In this phase, we create new features by combining or transforming existing ones, enhancing the dataset's ability to capture patterns and relationships. The following feature engineering steps have been applied to four different dataframes:

1. Original Dataset (df_new):

Dues: A new feature 'Dues' was created by summing the bill amounts from April to September. This feature reflects the cumulative outstanding balance over this period, providing insights into the client's payment behavior.

Previous_payments: 'Previous_payments' was derived by summing the previous payment amounts from April to September. This feature represents the total payments made by the client during this period, offering a perspective on their financial management.

2. SMOTE Balanced Dataset (df_smote):

Similar to the original dataset, 'Dues' and 'Previous_payments' features were created by summing the respective bill amounts and previous payment amounts. In the balanced dataset obtained using the Synthetic Minority Over-sampling Technique (SMOTE), these features contribute to a more balanced and representative dataset for modelling.

3. Scaled Unbalanced Dataset (scaled_unbalanced):

In the unscaled, unbalanced dataset, feature engineering was applied in the same way as the original dataset. 'Dues' and 'Previous_payments' were created by summing the bill amounts and previous payment amounts for April to September.

4. Scaled Balanced Dataset (scaled_balanced):

The scaled, balanced dataset also underwent feature engineering, resulting in the creation of 'Dues' and 'Previous_payments' by summing the corresponding bill amounts and previous payment amounts.

Feature engineering is a crucial aspect of data preparation, providing enhanced predictive variables that can improve the performance of our credit card default prediction models. By introducing 'Dues' and 'Previous_payments,' we aim to capture a client's financial behavior more comprehensively, which can lead to more accurate and robust model predictions.

The effectiveness of these engineered features will be evaluated in subsequent sections as we assess model performance and predictive accuracy across the four different dataframes.

VI. FEATURE SCALING

Feature scaling is an essential data preprocessing step aimed at bringing the dataset's features to a common scale. In this phase, we apply various scaling techniques to address the issue of non-normally distributed features. These scaling techniques are pivotal in ensuring that our predictive models perform optimally by mitigating the impact of feature scaling differences.

The feature scaling techniques employed include:

6.1 *Log Transformation*: Applying the natural logarithm to the data to reduce the impact of outliers and bring skewed features closer to a normal distribution.

6.2 *L1 Norm (Lasso Norm):* Scaling features based on the L1 norm, which can effectively handle sparsity and outliers.

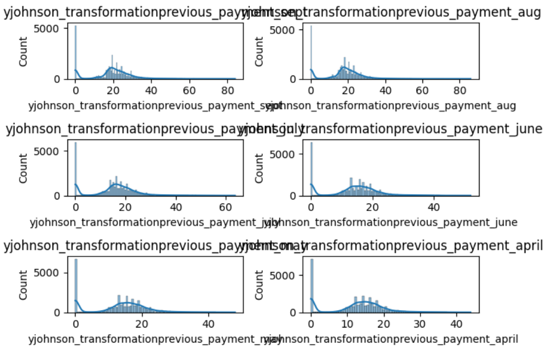
6.3 *L2 Norm (Ridge Norm):* Scaling features using the L2 norm, promoting the uniform scaling of features while being less sensitive to outliers.

6.4 *Yeo-Johnson Transformation:* A versatile power transformation method designed to address skewness and heteroscedasticity in data.

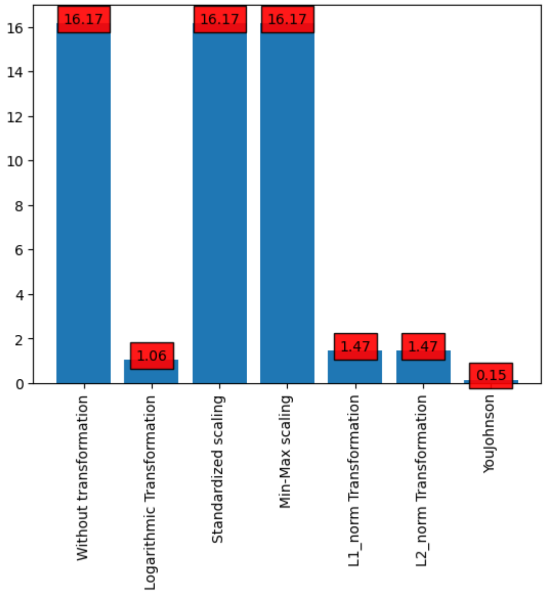
6.5 *Standardization (Z-score Scaling*): Transforming data to have a mean of 0 and a standard deviation of 1, facilitating the comparison of features with different units.

6.6 *Min-Max Scaling (Normalization):* Rescaling features to a specified range (typically between 0 and 1), ensuring that all features have a similar scale.

During the feature scaling process, an essential aspect that requires attention is the assessment of feature skewness. Skewness indicates how asymmetrically the data is distributed. In this section, we will present the skewness results for the features after applying each scaling technique. A visual representation of skewness results will be provided to offer a comprehensive view of the effectiveness of each technique in normalizing the data distribution.



In our analysis, the Yeo-Johnson transformation displayed the minimum skewness among all techniques, making it a promising choice for our predictive models.



VII. DATA ENCODING

Data encoding is a fundamental step in preparing the dataset for machine learning models, particularly when dealing with categorical features. In this section, we will explore the application of one-hot encoding to specific categorical features within the dataset, namely 'Sex,' 'Marriage,' and 'Education.'

One-hot encoding is a technique that transforms non-numerical categorical variables into a binary format, creating new binary columns for each category. This process ensures that the categorical variables do not introduce any ordinal relationships, preserving the integrity of the data while making it compatible with machine learning algorithms.

By applying one-hot encoding to these selected features, we create a representation that allows our models to effectively process and utilize the categorical information during the predictive modelling phase. This data encoding step is essential for ensuring the accurate and meaningful utilization of categorical features in our Credit Card Default Prediction System.

VIII. MODEL BUILDING APPROACH

The selection of an appropriate classification algorithm is crucial in building an effective Credit Card Default Prediction System. In this section, we present the diverse set of classification algorithms that were applied to our dataset for model training. The algorithms used in this analysis include:

Logistic Regression

Support Vector Classifier

Decision Tree

Random Forest Classifier

XGBoost Classifier

AdaBoost Classifier

These algorithms offer a wide range of approaches to solving classification problems, each with its unique strengths and characteristics. By employing multiple algorithms, we aim to explore their performance across different datasets, evaluating their ability to predict credit card defaults.

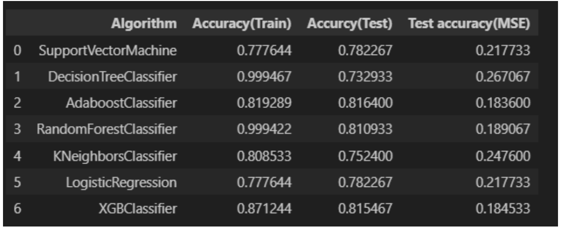
For model evaluation, we utilized key performance metrics, including:

Accuracy Score: A measure of the percentage of correctly predicted instances, providing insight into the overall model accuracy.

Mean Squared Error (MSE): An evaluation metric that quantifies the average squared difference between predicted and actual values, emphasizing prediction precision.

Through rigorous experimentation with these algorithms on the four datasets, we sought to identify the most suitable candidate for our Credit Card Default Prediction System. The dataset labelled as 'df_new' emerged as the best-performing dataset, showcasing an accuracy of 81.92% on the training set and 81.64% on the testing set.

The subsequent sections will delve into the specifics of our model training process, detailing the individual algorithm performances and key findings.



	Algorithm	Accuracy(Train)	Accurcy(Test)	Test accuracy(MSE)
0	SupportVectorMachine	0.777644	0.782267	0.217733
1	DecisionTreeClassifier	0.999467	0.732933	0.267067
2	AdaboostClassifier	0.819289	0.816400	0.183600
3	RandomForestClassifier	0.999422	0.810933	0.189067
4	KNeighborsClassifier	0.808533	0.752400	0.247600
5	LogisticRegression	0.777644	0.782267	0.217733
6	XGBClassifier	0.871244	0.815467	0.184533

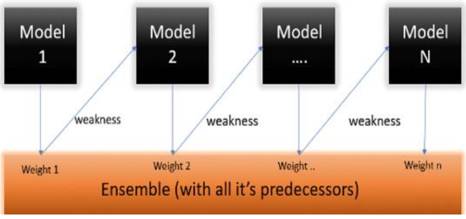
8.1 Model selection

In the process of choosing the most appropriate model for our Credit Card Default Prediction System, the evaluation of various classification algorithms led us to a significant decision. The selection criteria were primarily based on the accuracy score, which measures the percentage of correctly predicted instances. After thorough analysis, it was determined that the *AdaBoost Classifier* demonstrated the highest accuracy and emerged as the preferred choice for our predictive model.

8.2 AdaboostClassifier Working

The AdaBoost (Adaptive Boosting) Classifier is an ensemble learning technique that aims to improve the accuracy of weak learners by combining them into a strong classifier. It works by assigning more weight to misclassified instances during training, thereby focusing on the challenging cases and continuously improving the model's performance. The working of the AdaBoost Classifier can be summarized as follows:

- Initialization: Each instance in the dataset is assigned an equal weight, and a weak learner (typically a decision stump) is trained on the data.
- Weighted Learning: After the first weak learner is trained, misclassified instances are given higher weight in the dataset. This makes the algorithm concentrate more on the difficult instances, allowing the subsequent weak learners to improve on these cases.
- Iterative Process: The process is repeated iteratively, with new weak learners being trained on the updated dataset with weighted instances.
- Combining Weak Learners: The weak learners' predictions are combined to form a strong, weighted model, where each learner's contribution is weighted based on its accuracy.
- Final Model: The final model is a weighted combination of multiple weak learners, which results in a robust classifier capable of handling complex decision boundaries.



8.3 Model Optimization (Hypertuning)

Model hyperparameter tuning is a pivotal phase in our Credit Card Default Prediction System's development, focusing on optimizing the AdaBoost Classifier's performance. To fine-tune this classifier, we considered specific hyperparameters that significantly impact its predictive accuracy and behavior.

The hyperparameters under scrutiny during the tuning process include:

- n_estimators: The number of weak learners (estimators) used in the AdaBoost ensemble. We explored values of [30, 50, 70, 100] to assess the impact of the number of estimators on the model's performance.

- algorithm: The boosting algorithm employed, with choices of 'SAMME' and 'SAMME.R.' This hyperparameter affects the boosting process and its efficiency.

- learning rate: The rate at which the model adapts to errors made by previous estimators. We investigated learning rates of [0.5, 0.7, 1, 1.4] to gauge their influence on the model's performance.

Hyperparameter tuning was conducted using GridSearch, a systematic and exhaustive search technique. GridSearch methodically evaluates the model's performance across different combinations of hyperparameter values, allowing us to identify the most optimal configuration for the AdaBoost Classifier. The objective is to strike a balance between model complexity and predictive accuracy, ensuring the model is fine-tuned to deliver exceptional results in our credit card default prediction task.

8.4 Model Evaluation

The evaluation of our fine-tuned AdaBoost Classifier is a critical phase in gauging the performance and reliability of our Credit Card Default Prediction System. While accuracy serves as a fundamental performance metric, we delve deeper into the model's assessment, encompassing a comprehensive array of evaluation techniques to ensure a robust and reliable prediction system.

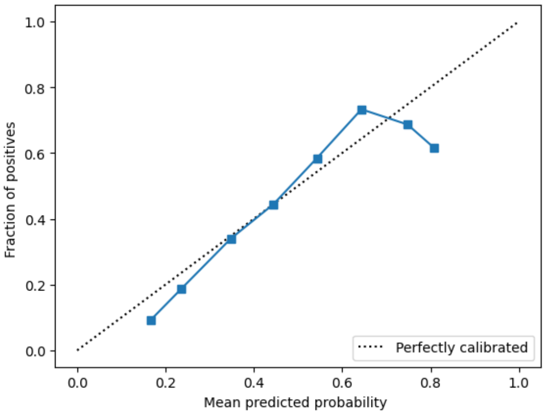
Key Evaluation Metrics and Techniques

Accuracy: A measure of the percentage of correctly predicted instances. Our model achieved an accuracy score of 81.64%, highlighting its ability to correctly predict credit card defaults.

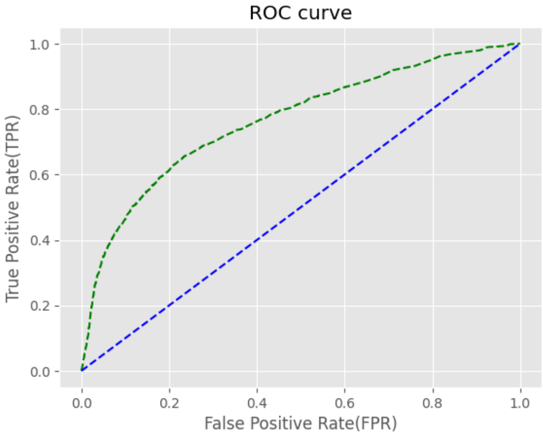
Classification Report: We conducted a detailed examination of the model's performance, generating a classification report. This report provides valuable insights into key metrics such as precision, recall, and F1-score, offering a comprehensive view of the model's performance across different classes.

```
Final accuracy of the model is 81.96%
Classification Report
              precision    recall f1-score   support

           0       0.84      0.95      0.89      5873
           1       0.67      0.34      0.45      1627

    accuracy                           0.82      7500
   macro avg       0.75      0.65      0.67      7500
weighted avg       0.80      0.82      0.80      7500

Confusion Matrix
[[5596  277]
 [1076  551]]
```

Calibration Curve: A calibration curve was plotted to assess the relationship between predicted probabilities and actual outcomes. This curve aids in understanding the model's reliability in predicting default probabilities.

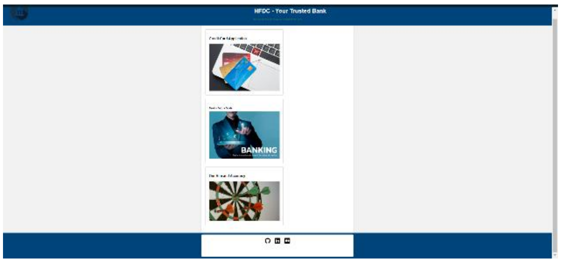


Receiver Operating Characteristic (ROC) Curve: The ROC curve is a fundamental tool for evaluating a classifier's performance. By plotting the true positive rate against the false positive rate, we gauge the model's ability to distinguish between default and non-default cases.



IX. WEB APPLICATION

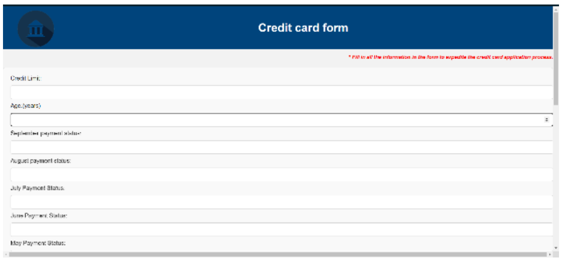
In the pursuit of making our Credit Card Default Prediction System accessible and user-friendly, we developed a web application using Flask, a popular web framework for Python. The web application serves as a user interface, allowing individuals and organizations to interact with our predictive model conveniently.



9.1 User interface architecture

- Subpages within UI

Credit Card Default Prediction: The first subpage within our UI serves as the gateway to the core functionality of our system. Here, users can input relevant data, and our predictive model processes this information to deliver real-time predictions regarding credit card issuance. By leveraging the power of data science and machine learning, we empower users to make informed decisions on extending credit, mitigating potential financial risks, and ensuring responsible lending practices.



Credit card form

* Fill in all the information in the form to expedite the credit card application process.

Credit Limit:

Age (years)

September payment status:

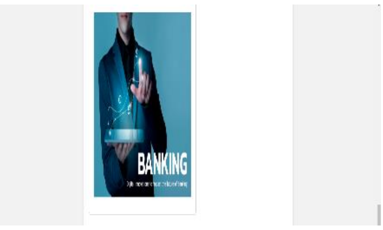
August payment status:

July Payment Status:

June Payment Status:

May Payment Status:

Future Goals: Our commitment to the future is articulated in this subpage, where we outline our vision and aspirations. We share our dedication to advancing the field of predictive modelling and data-driven decision-making, aiming to continually enhance our system's capabilities and expand its horizons to address broader financial challenges.



Aims and Accuracy: In this subpage, we delve deeper into the technical aspects of our algorithm. Here, we elaborate on the core principles and methodologies that drive our predictive model. We emphasize the accuracy and precision achieved through our data analysis, machine learning techniques, and fine-tuning, offering users an insight into the robustness of our credit card default prediction system.