Contents

High Level Design(HLD)

CCDP(Credit Card Default Prediction) based Banking

Revision Number**: 2.0**

Last date **of** revision: 23/06/2023

Document Version Control

Date Issued	Version	Description	Author
02/06/2023	1	Initial HLD - V1.0	Govind D.
23/06/2023	2	Updated KPI - V1.1	Govind D.

Abstract

Credit card default is a significant problem faced by financial institutions and borrowers alike. Accurate prediction of credit card default can help banks and lenders make informed decisions regarding creditworthiness and risk assessment, leading to more efficient loan approval processes and better management of credit portfolios. This project focuses on developing a machine learning-based credit card default prediction model that can effectively identify potential defaulters.

The project utilizes a diverse set of features, including demographic information, credit history, and transaction patterns, to train and evaluate several machine learning algorithms. The algorithms considered include logistic regression, decision trees, random forests, and gradient boosting. These algorithms are chosen for their ability to handle complex patterns and non-linear relationships in the data.

The dataset used in this project consists of historical credit card data, containing information about borrowers, their financial attributes, and their repayment behaviour. The dataset is pre-processed to handle missing values, outliers, and feature scaling to ensure data quality and model robustness.

Various evaluation metrics such as accuracy, precision, recall, and F1-score are employed to assess the performance of the developed models. Additionally, techniques such as cross-validation and hyperparameter tuning are employed to optimize the models and mitigate issues related to overfitting.

The results of the experiments demonstrate that the machine learning models can effectively predict credit card default with a high degree of accuracy. The best-performing algorithm is identified based on the evaluation metrics, and its performance is compared with other models to highlight its superiority.

The developed credit card default prediction model can be used as a valuable tool by financial institutions to assess credit risk, detect potential defaulters, and make informed lending decisions. By employing such models, lenders can minimize losses due to defaults, reduce the number of bad loans, and maintain a healthier credit portfolio.

Overall, this project showcases the application of machine learning techniques in the field of credit risk assessment and demonstrates the potential for improving the efficiency and accuracy of credit card default prediction.

1 Introduction

1.1 Why this High-Level Design Document?

The purpose of this High-Level Design (HLD) Document is to add the necessary detail to the current project description to represent a suitable model for coding. This document is also intended to help detect contradictions prior to coding, and can be used as a reference manual for how the modules interact at a high level.

The HLD will:

- Present all of the design aspects and define them in detail Describe the user interface being implemented
- Describe the hardware and software interfaces
- Describe the performance requirements
- Include design features and the architecture of the project
- List and describe the non-functional attributes like:
    - Security
    - Reliability
    - Maintainability
    - Portability
    - Reusability
    - Application compatibility
    - Resource utilization
    - Serviceability

1.2 Scope

The HLD documentation presents the structure of the system, such as the database architecture, application architecture (layers), application flow (Navigation), and technology architecture. The HLD uses non-technical to mildly-technical terms which should be understandable to the administrators of the system.

1.3 Definitions

Term	***Description***
CCPD	Credit Card Default Prediction
Database	Collection of all the information monitored by this system
IDE	Integrated Development Environment
AWS	Amazon Web Services

2 General Description

2.1 Product Perspective

The CCPD based solution system is a machine learning-based model which will help us to detect the CCDP records.

2.2 Problem statement

Financial threats are displaying a trend about the credit risk of commercial banks as the incredible improvements in the financial industry has arisen. In this way, one of the biggest threats faces by commercial banks is the risk prediction of credit clients. The goal is to predict the probability of credit default based on credit card owner's characteristics and payment history.

2.3 Proposed Solution

To address the challenge of credit card default prediction, we propose developing a machine learning-based solution that leverages historical credit card data and utilizes various algorithms for accurate prediction. The proposed solution consists of the following components:

1. **Data Preparation:** The historical credit card data is collected and pre-processed to ensure data quality. Missing values are handled through imputation techniques, outliers are treated appropriately, and feature scaling or normalization is performed to ensure uniformity among features.

2. **Feature Engineering:** Relevant features are identified and engineered to capture important patterns and relationships. This may include creating derived features based on domain knowledge or performing feature transformations to enhance predictive power.

3. **Model Selection:** A set of machine learning algorithms suitable for credit card default prediction is considered. Algorithms such as logistic regression, decision trees, random forests, and gradient boosting are evaluated based on their ability to handle the complexity of the data and capture non-linear relationships.

4. **Model Training:** The selected algorithms are trained on the pre-processed data using the training set. During the training phase, the models learn from the historical patterns in the data and establish relationships between the input features and the target variable (credit card default).

5. **Hyperparameter Tuning:** The hyperparameters of the models are tuned using techniques like grid search or random search. This process involves exploring different combinations of hyperparameters to optimize the models' performance and improve their ability to generalize to unseen data.

6. **Model Evaluation:** The trained models are evaluated using the testing set. Various evaluation metrics such as accuracy, precision, recall, F1-score, and area under the ROC curve are calculated to assess their performance. This step allows for the comparison of different models and the selection of the most effective algorithm for credit card default prediction.

7. **Deployment:** The chosen model is deployed in a production environment, either through integration into an existing system or through the development of a dedicated API or user interface. This allows financial institutions to utilize the model for real-time credit card default prediction.

8. **Monitoring and Maintenance:** The deployed model is continuously monitored to ensure its accuracy and reliability. Regular maintenance is performed to update the model with new data and make improvements based on feedback and emerging trends.

The proposed solution aims to provide financial institutions with a reliable and accurate credit card default prediction system. By leveraging machine learning techniques and properly handling data, the solution enables lenders to make informed decisions, mitigate risks, and maintain a healthy credit portfolio.

2.4 Further Improvements

While the proposed solution offers a solid foundation for credit card default prediction, there are several areas where further improvements can be made:

1. **Feature Selection:** Enhance the feature selection process by employing advanced techniques such as feature importance ranking, recursive feature elimination, or feature correlation analysis. This can help identify the most informative features and eliminate redundant or irrelevant ones, thereby improving model performance and interpretability.

2. **Handling Imbalanced Data:** Develop strategies to handle imbalanced datasets where the number of defaulters is significantly smaller than non-defaulters. Explore techniques like oversampling (e.g., SMOTE), under sampling, or ensemble methods (e.g., balanced random forests) to address class imbalance and prevent biased predictions.

3. **Ensemble Methods:** Investigate the use of ensemble methods, such as stacking or boosting, to combine the predictions of multiple models. Ensemble methods can often enhance the predictive performance by leveraging the strengths of different models and mitigating their individual weaknesses.

4. **Model Explain ability:** Incorporate techniques for model explain ability, such as feature importance rankings, SHAP values, or LIME (Local Interpretable Model-agnostic Explanations). These methods provide insights into the factors driving the model's predictions, enabling better understanding and trust in the decision-making process.

5. **Time Series Analysis:** If the credit card data includes a temporal component, consider incorporating time series analysis techniques to capture trends, seasonality, or other time-dependent patterns. This can enhance the model's ability to predict credit card defaults by considering the dynamic nature of the data.

6. **External Data Sources:** Explore the integration of external data sources, such as macroeconomic indicators, industry-specific data, or social media sentiment analysis. Incorporating additional data can provide a broader context for credit card default prediction and potentially improve the accuracy of the models.

7. **Continuous Model Updating:** Implement mechanisms to update the deployed model regularly with new data. As new credit card data becomes available, retrain the model periodically to capture the most up-to-date patterns and trends in credit card default behaviour.

8. **Feedback Loop:** Establish a feedback loop with financial institutions to gather feedback on model performance and address any challenges or limitations identified in real-world scenarios. This feedback can guide further improvements and ensure the model remains effective in practical applications.

By incorporating these further improvements, the credit card default prediction solution can become more accurate, robust, and reliable, enabling financial institutions to make informed decisions and effectively manage credit risk.

2.5 Technical Requirements

To implement the credit card default prediction solution, the following technical requirements should be considered:

1. **Programming Languages:** Proficiency in programming languages commonly used in machine learning, such as Python or R, is essential. These languages provide a wide range of libraries and frameworks for data manipulation, model training, and evaluation.

2. **Machine Learning Libraries:** Familiarity with popular machine learning libraries, such as scikit-learn, TensorFlow, is necessary. These libraries provide various algorithms and tools for data pre-processing, feature selection/engineering, model training, and evaluation.

3**. Data Manipulation and Analysis:** Proficiency in data manipulation and analysis libraries, such as pandas and NumPy, is important for handling and pre-processing the credit card data. These libraries enable efficient data cleaning, transformation, and feature extraction.

4. **Model Training and Evaluation:** Understanding of different machine learning algorithms, their strengths, and their limitations is crucial for selecting and training appropriate models for credit card default prediction. Experience with evaluating model performance using evaluation metrics, cross-validation, and hyperparameter tuning techniques is also required.

5. **Data Visualization:** Proficiency in data visualization libraries, such as Matplotlib or Seaborn, is beneficial for exploring and visualizing the credit card data. Visualizations help in gaining insights, identifying patterns, and communicating results effectively.

6. **Deployment and APIs:** Experience with deploying machine learning models, creating APIs, and integrating models into production environments is necessary. Knowledge of frameworks such as Flask or Django for developing APIs can facilitate the deployment process.

7. **Version Control:** Proficiency in version control systems like Git is important for managing code, collaborating with team members, and tracking changes made throughout the project.

8. **Scalability and Performance Optimization:** Understanding techniques for optimizing model performance and scalability is valuable, especially when dealing with large datasets or real-time predictions. Knowledge of parallel computing, distributed computing frameworks (e.g., Spark), or cloud services (e.g., AWS, GCP) can be advantageous.

9. **Model Explain ability:** Familiarity with techniques for model explain ability, such as SHAP values or LIME, can aid in understanding and interpreting the predictions made by the models.

10. **Security and Privacy:** Awareness of security measures and best practices for handling sensitive credit card data is crucial to protect privacy and ensure compliance with data protection regulations.

By fulfilling these technical requirements, the credit card default prediction solution can be effectively developed, deployed, and maintained, providing accurate predictions and assisting financial institutions in making informed decisions.

2.6 Data Requirements

To successfully develop and train a credit card default prediction model, the following data requirements should be considered:

1. Historical Credit Card Data: Access to a comprehensive and representative dataset of historical credit card transactions is essential. The dataset should include a sufficient number of records that cover a diverse range of borrowers, credit limits, transaction types, repayment behaviours, and other relevant attributes.

2. Borrower Information: The dataset should contain relevant information about the borrowers, such as demographics (age, gender, occupation), employment history, income level, and any other demographic factors that might impact creditworthiness.

3. Credit Card Transaction Details: Detailed information about credit card transactions is necessary, including transaction amounts, transaction types (e.g., purchase, cash withdrawal), transaction dates, merchant categories, and any additional transaction-specific attributes available.

4. Repayment Behaviour: Information about the borrowers' repayment behaviour is crucial for credit card default prediction. This includes details such as repayment amounts, repayment dates, frequency of late payments, and any instances of default or delinquency.

5. Account Information: Additional account-related data, such as credit limits, account balances, credit utilization ratios, and the duration of the credit card account, can provide valuable insights into the borrowers' financial stability and payment behaviour.

6. Time Frame: Sufficient historical data should be available to capture the patterns and trends related to credit card defaults. Ideally, the dataset should cover a substantial period, allowing the model to capture both short-term and long-term patterns.

7. Data Quality: Ensuring the quality of the data is crucial. The dataset should be checked for missing values, outliers, and inconsistencies. Missing values can be handled through imputation techniques, and outliers should be treated appropriately to prevent them from negatively impacting the model's performance.

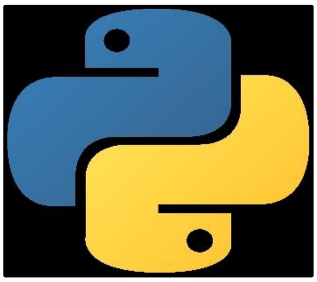
8. Data Privacy and Compliance: Adhere to data privacy regulations and ensure that any personally identifiable information (PII) is handled securely and in compliance with applicable laws and regulations. Anonymizing or encrypting sensitive data can help protect privacy and maintain compliance.

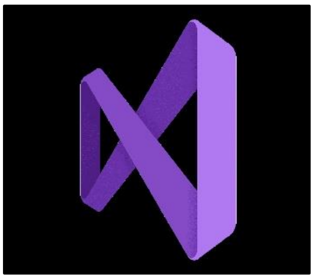
It is important to note that data requirements may vary based on the specific goals of the credit card default prediction project. It is recommended to collaborate with domain experts and stakeholders to determine the most relevant and appropriate data sources that align with the project objectives.

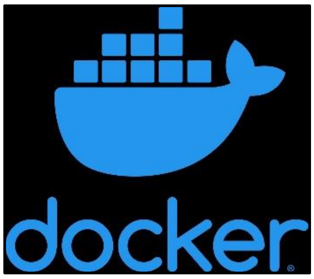
By meeting these data requirements, the credit card default prediction model can be trained on a reliable and representative dataset, leading to more accurate predictions and actionable insights for credit risk assessment.

2.7 Tools Used

Python programming language and frameworks such as NumPy, Pandas, Scikit-learn, Flask, Machine Learning Models such as Logistic Regression, Decision Tree with Hyperparameter, HTML, AWS, etc., are used to build the whole model.







- Visual Studio Code is used as IDE.
- For visualization of the plots, Matplotlib, Seaborn and Plotly are used.
- AWS is used for deployment of the model.
- Tableau/Power BI is used for dashboard creation.
- MySQL/MongoDB is used to retrieve, insert, delete, and update the database.
- Front end development is done using HTML/CSS Python Django is used for backend development.
- GitHub is used as version control system.

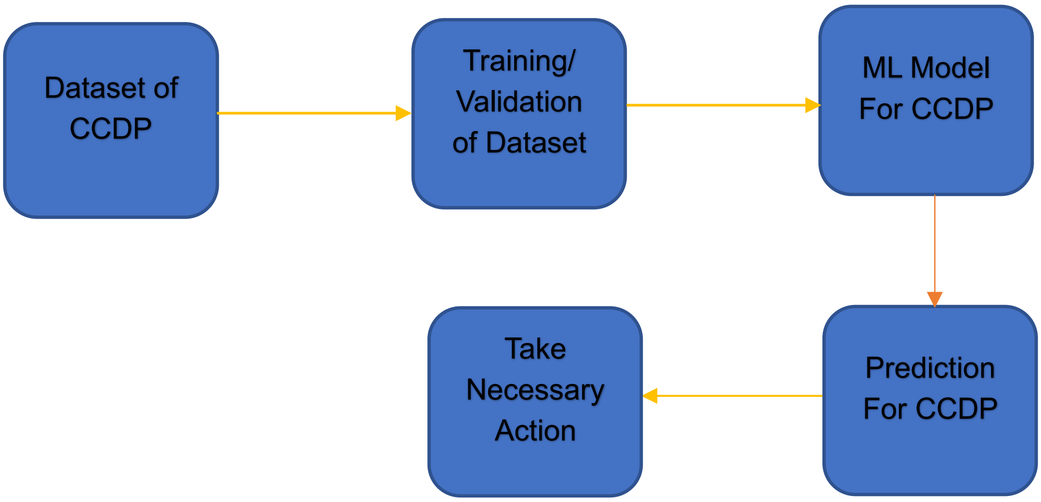
2.8 Assumptions

The main objective of the project is to implement the use cases as previously mentioned (2.2 Problem Statement) for new dataset that comes.

3 Design Details

3.1 Process Flow

The process flow of the credit card default prediction project can be summarized into the following steps:



1. Data Acquisition: Obtain a dataset containing historical credit card data, including information about borrowers, their financial attributes, and their repayment behaviour. This data can be obtained from financial institutions or publicly available sources.

2. Data Pre-processing: Clean the dataset by handling missing values, outliers, and inconsistencies. Perform data transformations, such as feature scaling, normalization, or encoding categorical variables, to ensure uniformity and prepare the data for modeling.

3. Feature Selection/Engineering: Analyse the dataset to identify relevant features that may impact credit card default. Use techniques such as correlation analysis, feature importance, or domain knowledge to select the most informative features. Additionally, engineer new features based on existing ones that might improve the predictive power of the models.

4. Model Training: Split the pre-processed dataset into training and testing subsets. Utilize machine learning algorithms such as logistic regression, decision trees, random forests, and gradient boosting to train predictive models. Adjust the hyperparameters of the models to optimize their performance.

5. Model Evaluation: Evaluate the trained models using appropriate evaluation metrics, such as accuracy, precision, recall, F1-score, and area under the ROC curve. Compare

the performance of different models to identify the most effective algorithm for credit card default prediction.

6. Model Optimization: Perform techniques like cross-validation and hyperparameter tuning to fine-tune the models and prevent overfitting. Optimize the models based on the evaluation results to enhance their predictive accuracy.

7. Model Deployment: Once the best-performing model is selected, deploy it for real-world usage. Develop a user-friendly interface or integrate the model into an existing system where it can be used to predict credit card default on new data.

8. Performance Monitoring: Continuously monitor the performance of the deployed model to ensure its accuracy and reliability. Collect feedback from users and update the model periodically, if necessary, to account for changing trends or data patterns.

Throughout the project, it is essential to document the methodology, decisions, and findings at each step to ensure reproducibility and facilitate future improvements.

3.1.1 Model Training and Evaluation

Model Training and Evaluation are crucial steps in the credit card default prediction project. Here's a detailed explanation of these steps:

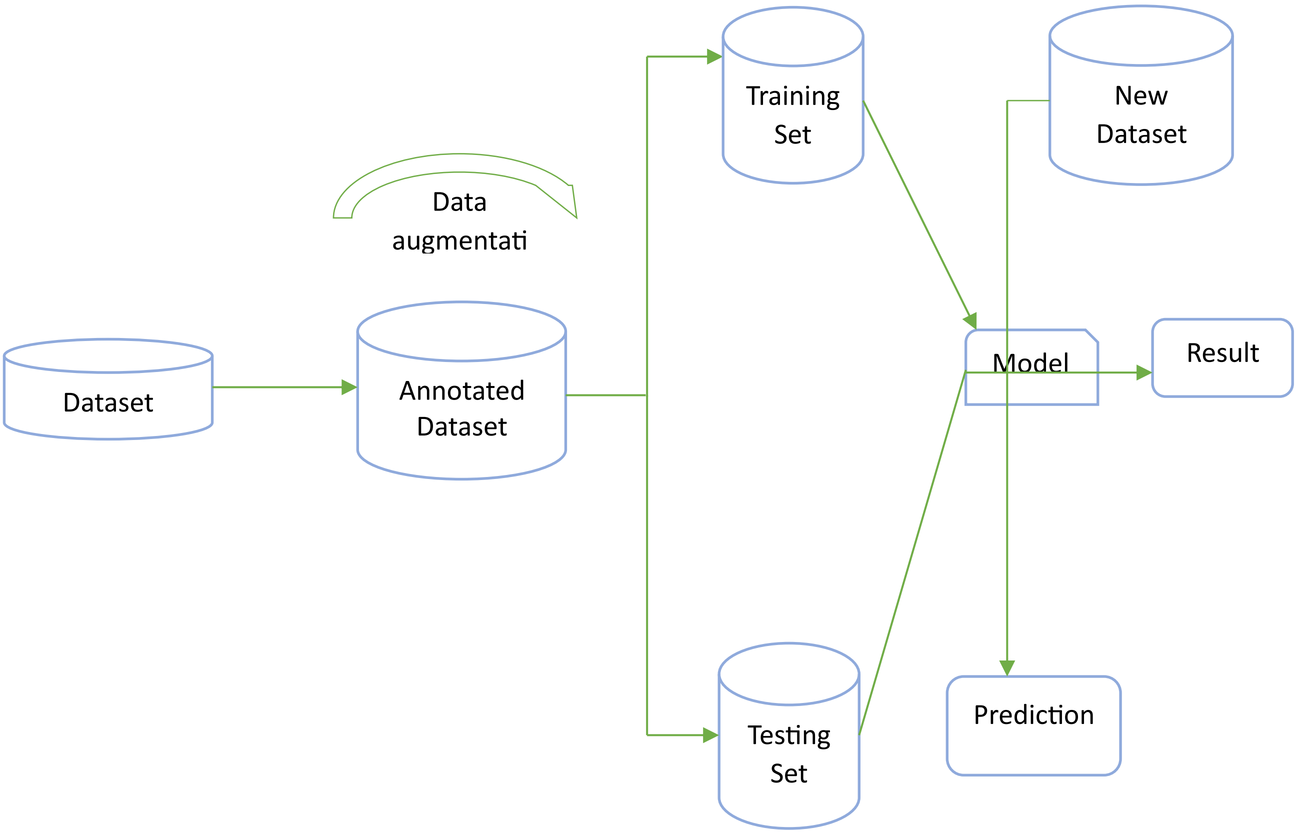
1. Model Training:

a. Split the pre-processed dataset: Divide the pre-processed dataset into two subsets: the training set and the testing set. The training set is used to train the machine learning models, while the testing set is used to assess their performance on unseen data.

b. Select the machine learning algorithms: Choose a set of algorithms suitable for credit card default prediction, such as logistic regression, decision trees, random forests, or gradient boosting. These algorithms are known for their effectiveness in handling classification problems.

c. Train the models: Fit the selected algorithms on the training data. The models learn the underlying patterns and relationships between the features and the target variable (credit card default) during this training phase.

d. Adjust hyperparameters: Fine-tune the models' hyperparameters to optimize their performance. This process involves exploring different parameter combinations using techniques like grid search or random search to find the best set of hyperparameters.



2. Model Evaluation:

a. Predict on the testing set: Apply the trained models to the testing dataset to obtain predictions for credit card default. The models use the learned patterns to classify whether a borrower is likely to default or not.

b. Evaluate model performance: Compare the predicted outcomes with the actual values from the testing set. Calculate various evaluation metrics, including accuracy, precision, recall, F1-score, and area under the ROC curve. These metrics provide insights into the model's predictive accuracy, ability to identify defaulters, and overall performance.

c. Compare models: Assess the performance of different models trained during the project. Compare evaluation metrics across different algorithms to identify the most effective model for credit card default prediction.

d. Handle imbalanced data (if applicable): In credit card default prediction, it is common to encounter imbalanced datasets where the number of defaulters is significantly smaller than non-defaulters. If this is the case, apply techniques like oversampling, under sampling, or SMOTE (Synthetic Minority Over-sampling Technique) to address the class imbalance issue and obtain more reliable evaluation results.

By following these steps, you can train multiple machine learning models, evaluate their performance, and select the best-performing model for credit card default prediction. Remember to document the results and make informed decisions based on the evaluation metrics to ensure the accuracy and effectiveness of the chosen model.

3.1.2 Deployment Process

The deployment process in the credit card default prediction project involves making the trained model accessible for real-world usage. Here's an overview of the deployment process:

1. Model Selection: Choose the best-performing model from the evaluation phase. Consider the model's accuracy, precision, recall, F1-score, and other relevant metrics to determine its effectiveness in predicting credit card defaults.

2. Integration: Integrate the selected model into a production environment or system where it can be utilized for credit card default prediction. This could involve integrating the model into an existing banking system, risk assessment platform, or any other relevant infrastructure.

3. Model Packaging: Package the trained model along with any required dependencies into a deployable format. This could involve saving the model parameters, feature encodings, and any pre-processing steps that were applied during training.

4. API Development: Create an API (Application Programming Interface) that allows external systems or applications to interact with the deployed model. The API defines the input format required for credit card data and returns the prediction results.

5. User Interface (UI) Development (Optional): If necessary, develop a user-friendly interface that allows users to input credit card data and receive predictions. The UI can enhance the usability of the model, especially for non-technical users.

6. Testing: Perform thorough testing of the deployed model and the associated API or user interface. Test the model with different input scenarios, including both typical and edge cases, to ensure its robustness and reliability.

7. Scalability and Performance Optimization: Optimize the deployment infrastructure to handle large-scale requests and ensure low-latency responses. This could involve scaling up the infrastructure, implementing caching mechanisms, or optimizing resource allocation.

8. Security Considerations: Ensure that appropriate security measures are implemented to protect sensitive credit card data and prevent unauthorized access. Employ encryption, authentication mechanisms, and access controls to maintain data privacy and integrity.

9. Monitoring and Maintenance: Set up monitoring mechanisms to continuously track the performance of the deployed model. Monitor prediction accuracy, system health, and any potential issues or anomalies. Schedule regular maintenance to update the model as new data becomes available or when improvements are identified.

10. User Training and Support: Provide training and support to users who will be utilizing the deployed model. Offer documentation or user guides that explain how to use the model and interpret the predictions effectively.

By following these deployment steps, the credit card default prediction model can be effectively integrated into a production environment, allowing financial institutions to utilize it for assessing credit risk and making informed lending decisions. Regular monitoring and maintenance ensure the model's accuracy and reliability over time.

3.2 Event log

The system should log every event so that the user will know what process is running internally.
Initial Step-By-Step Description:
1. The System identifies at what step logging required
2. The System should be able to log each and every system flow.
3. Developer can choose logging method. You can choose database logging/ File logging as well.
4. System should not hang even after using so many loggings. Logging just because we can easily debug issues so logging is mandatory to do.

3.3 Error Handling

Should errors be encountered, an explanation will be displayed as to what went wrong? An error will be defined as anything that falls outside the normal and intended usage.

3.4 Performance

The CCDP based solution is used for detection of anomalies in the society whenever record detects any anomalies. it will inform concern authorities and takes necessary action, so it should be as accurate as possible. So that it will not mislead the concern authorities. Also, model retraining is very important to improve the performance.

3.5 Reusability

The code written and the components used should have the ability to be reused with no problems.

3.6 Application Compatibility

The different components for this project will be using Python as an interface between them. Each component will have its own task to perform, and it is the job of the Python to ensure proper transfer of information.

3.7 Resource Utilization

When any task is performed, it will likely use all the processing power available until that function is finished.

3.8 Deployment

4 Conclusion

In conclusion, the credit card default prediction project focuses on developing a machine learning-based model to accurately predict credit card default. By leveraging historical credit card data and employing various machine learning algorithms, the project aims to assist financial institutions in assessing credit risk and making informed lending decisions. Throughout the project, several key steps were followed. The data acquisition phase involved obtaining a dataset containing borrower information, financial attributes, and repayment behaviour. The data pre-processing step ensured data quality by handling missing values, outliers, and performing feature scaling.

Feature selection/engineering allowed for the identification and creation of relevant features that impact credit card default prediction. Various machine learning algorithms, such as logistic regression, decision trees, random forests, and gradient boosting, were trained using the pre-processed data.

Model evaluation involved assessing the performance of the trained models using metrics such as accuracy, precision, recall, F1-score, and area under the ROC curve. The best-performing model was selected based on these metrics and compared with other models to determine its superiority.

The deployment process focused on integrating the selected model into a production environment or system. This involved packaging the model, developing an API or user interface for interaction, and ensuring scalability, performance optimization, security, and ongoing monitoring and maintenance. Overall, the credit card default prediction project offers significant benefits to financial institutions. By accurately predicting credit card defaults, lenders can minimize losses, reduce the number of bad loans, and maintain a healthier credit portfolio. This project demonstrates the effectiveness of machine learning techniques in the domain of credit risk assessment and highlights the potential for improving the efficiency and accuracy of credit card default prediction.