

**Credit Card Default Prediction**

High Level Design

Domain: Machine Learning

Creator: Arun Kumar Maurya

Date: 28/06/2024

Arun Kumar Maurya

**Document Version Control**

|  |  |  |  |
| --- | --- | --- | --- |
| Date issued | Version | Description | Author |
| June 28, 2024 | 1.1 | First Draft | Arun Kumar Maurya |
| June 29, 2024 | 1.2 | Added unit test cases | Arun Kumar Maurya |
| June 30, 2024 | 1.3 | Correcting font sizes | Arun Kumar Maurya |

Contents

[Abstract 3](#_Toc110595993)

[Introduction 3](#_Toc110595994)

[What is High-Level Design Document? 3](#_Toc110595995)

[Scope 4](#_Toc110595996)

[General Description 4](#_Toc110595997)

[Definitions 4](#_Toc110595998)

[Product Description 4](#_Toc110595999)

[Problem Statement 4](#_Toc110596000)

[Proposed solution 5](#_Toc110596001)

[Further improvements 5](#_Toc110596002)

[Data requirements 5](#_Toc110596003)

[Tools used 5](#_Toc110596004)

[Hardware Requirements 7](#_Toc110596005)

[Constraints 7](#_Toc110596006)

[Assumptions 7](#_Toc110596007)

[Design Details 7](#_Toc110596008)

[Process Flow 7](#_Toc110596009)

[Event Log 8](#_Toc110596010)

[Error Handling 9](#_Toc110596011)

[Performance 9](#_Toc110596012)

[Reusability 9](#_Toc110596013)

[Application Compatibility 9](#_Toc110596014)

[Resource Utilization 9](#_Toc110596015)

[Dashboards 10](#_Toc110596016)

[KPIs (Key Performance Indicators) 10](#_Toc110596017)

[Conclusion 10](#_Toc110596018)

# **Abstract**

As the financial industry continues to evolve, commercial banks face increasing challenges, particularly in managing credit risk. One of the most significant threats is predicting the likelihood of credit defaults among their clients. Accurate risk prediction is crucial for maintaining financial stability and minimizing losses.

The primary objective is to develop a predictive solution that can estimate the probability of a credit card owner defaulting on their payments. This solution should leverage data on the owner's characteristics and their payment history to provide reliable risk assessments.

# **Introduction**

# 

# **What is High-Level Design Document?**

### High-Level Design Document Overview

The goal 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.

#### Context

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

### Objective

The primary objective is to develop a predictive solution that can estimate the probability of a credit card owner defaulting on their payments. This solution should leverage data on the owner's characteristics and their payment history to provide reliable risk assessments.

The HLD will:

* Present all of design aspects and define them in detail
* Describe all user interfaces being implemented
* Describe the hardware and software interfaces
* Describe the performance requirements
* Include design features and architecture of the project
* List and describe the non-functional attributes such as security, reliability, maintainability, portability, reusability, application compatibility. resource utilization, serviceability

## Scope

The HLD documentation presents the structure of the system, such as 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.

# **General Description**

## Definitions

|  |  |
| --- | --- |
| Term | Description |
| CCDP | Credit Card Default Prediction |
| Database | Collection of the Information |
| Cloud | A data center full of services connected to the internet performing service |
| IDE | Integrated Development Environment |
| UI | User Interface |
| Anvil | A Python based UI builder |
| Heroku | A cloud service |

# **Product Description**

### Predictive Model for Credit Default Risk

#### Overview

The predictive model for credit default risk is designed to assess the likelihood that a credit card owner will default on their payments. The model leverages a variety of data points related to the credit card owner's characteristics and payment history to make these assessments. The ultimate goal is to enable commercial banks to better manage credit risk and make informed lending decisions.

# **Problem Statement**

To create an ML based solution for predictive analysis of a person’s annual income and also deploy it in the form of a UI.

The aim to predict the traffic volume on specific weekday of month. This is basically regression problem.

# **Proposed solution**

### Proposed Solution for Predicting Credit Default Risk

#### Objective

The primary objective is to develop a predictive model that estimates the probability of a credit card owner defaulting on their payments based on their characteristics and payment history. This model will help commercial banks manage credit risk more effectively and make informed lending decisions.

#### Solution Overview

The proposed solution involves several key components, including data collection, preprocessing, model development, and implementation. The solution will leverage machine learning techniques to build an accurate and reliable predictive model.

#### Key Components

**Data Collection and Integration**

* 1. **Data Sources**: Gather data from internal bank systems and external sources, including demographic information, employment details, financial status, and payment history.
  2. **Data Integration**: Ensure seamless integration of data from various sources into a unified dataset for analysis.

**Data Preprocessing**

* 1. **Data Cleaning**: Address missing values, remove duplicates, and correct inconsistencies.
  2. **Feature Engineering**: Create relevant features such as credit utilization ratio, payment-to-income ratio, account age, and delinquency features.
  3. **Normalization and Scaling**: Standardize numerical features to ensure uniformity and consistency.

**Model Development**

* 1. **Algorithm Selection**: Evaluate various machine learning algorithms such as logistic regression, decision trees, random forests, and gradient boosting machines to identify the best-performing model.
  2. **Training and Validation**: Split the dataset into training and validation sets, train the model, tune hyperparameters, and validate performance using metrics like accuracy, precision, recall, F1 score, and AUC-ROC.
  3. **Model Evaluation**: Conduct cross-validation to ensure robustness and reliability of the model.

**Prediction and Risk Scoring**

* 1. **Probability Estimation**: The model will output a probability score indicating the likelihood of default for each credit card owner.
  2. **Risk Categorization**: Define risk categories based on probability scores (e.g., low risk, medium risk, high risk) and set thresholds according to business requirements.

**Implementation**

* 1. **System Integration**: Integrate the predictive model into the bank’s existing credit evaluation system to enable real-time predictions.
  2. **User Interface**: Develop a user-friendly interface for bank staff to input data and view risk predictions and detailed reports.
  3. **Monitoring and Maintenance**: Regularly update the model with new data, monitor performance, and retrain as necessary to adapt to changing trends.

#### Implementation Steps

**Project Kickoff**

* 1. Define project scope, objectives, and timeline.
  2. Assemble a cross-functional team including data scientists, data engineers, and business analysts.

**Data Preparation**

* 1. Collect and integrate data from various sources.
  2. Perform data cleaning, feature engineering, and normalization.

**Model Development**

* 1. Explore different machine learning algorithms.
  2. Train and validate models, and select the best-performing one.
  3. Conduct cross-validation to ensure model robustness.

**System Integration**

* 1. Integrate the model into the bank’s credit evaluation system.
  2. Develop a user interface for easy interaction and reporting.

**Testing and Validation**

* 1. Conduct thorough testing to ensure the model and system integration work as expected.
  2. Validate the model with historical data to assess its predictive power.

**Deployment**

* 1. Deploy the model in a live environment.
  2. Train bank staff on how to use the system and interpret the predictions.

**Monitoring and Maintenance**

* 1. Monitor the model’s performance regularly.
  2. Update and retrain the model with new data to maintain accuracy.

#### Benefits

* **Improved Risk Management**: Enhanced ability to predict and manage credit risk.
* **Data-Driven Decisions**: More informed lending decisions based on data-driven insights.
* **Financial Stability**: Reduced risk of defaults, leading to improved financial health for the bank.
* **Customer Relationship Management**: Ability to offer customized credit products based on individual risk profiles.

By implementing this solution, commercial banks can significantly improve their credit risk assessment processes, leading to more secure and profitable lending practices.

### Further Improvements

The proposed predictive model can be seamlessly integrated into any website or application, allowing users to quickly obtain risk assessments by inputting the necessary data through a user-friendly interface. Further improvements can be achieved by incorporating additional data and continuously retraining the model.

#### Data Requirements

The data requirements are critical to enhance the model's accuracy and reliability. For the predictive model to perform effectively, it needs a comprehensive dataset that includes the following features:

**Customer Demographics**

* 1. Age
  2. Gender
  3. Employment status
  4. Income level
  5. Education level

**Credit History**

* 1. Credit score
  2. Number of open credit lines
  3. Length of credit history
  4. Previous defaults or delinquencies

**Financial Behavior**

* 1. Credit card balance
  2. Credit utilization ratio
  3. Payment history (on-time or late payments)
  4. Monthly spending patterns

**Account Information**

* 1. Account age
  2. Type of credit card
  3. Credit limit

**External Factors**

* 1. Economic indicators (e.g., inflation rate, unemployment rate)
  2. Industry-specific risks

#### Additional Data Sources

To further improve the model, additional data can be acquired from reliable external sources. For instance, financial data from national and local databases, such as the MN Department of Transportation for region-specific economic indicators, can be utilized. This data should include:

**Economic Indicators**

* + Unemployment rates
  + Inflation rates
  + Local economic conditions

**External Financial Data**

* + Trends in the financial markets
  + Industry-specific risk factors

**Socioeconomic Data**

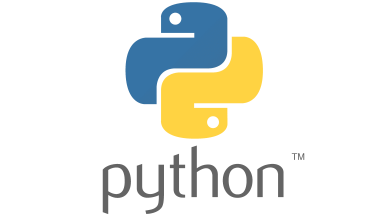
* + Demographic changes
  + Regional financial health indicators

These parameters will enrich the dataset, allowing the model to capture a more comprehensive picture of potential credit risks and improving its predictive accuracy. By continuously updating and expanding the dataset, the predictive model can remain robust and adaptive to changing conditions, ensuring reliable risk assessments for commercial banks.

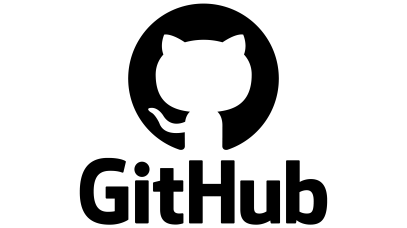
# 

# **Tools used**

Python programming language and frameworks such as NumPy, Pandas, Scikit-learn, Flask, Anvil, and a few other libraries were used to build the whole model.



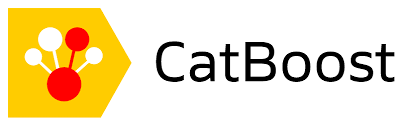












* For visualization tasks, matplotlib, seaborn and plotly were used
* For visualization tasks, matplotlib, seaborn and plotly were used
* Anvil and Flask were used for building the web application and server to run the code
* Apache Cassandra was used to storage and retrieval of data
* GitHub is used as version control system
* NumPy and Pandas were used to clean and interpret data
* Scikit-learn was used to cross validate and compare different models
* CatBoost Regressor was used to build the final model

# **Hardware Requirements**

* Windows Server, Linux, or any operating system that can run as a webserver, capable of delivering HTML5 content.
* Minimum 1.10 GHz processor or equivalent.
* Between 1-2 GB of free storage
* Minimum 512 MB of RAM
* 3 GB of hard-disk space

# **Constraints**

The front-end must be user friendly and should not need any one to have any prior knowledge in order to use it.

### Assumptions

The primary objective of this project is to implement the use case outlined in section 2.3 (Problem Statement) for new datasets received through the user interface (UI). It is assumed that all components of the project, including data processing, model training, and the UI, are designed to work seamlessly together as intended by the designer. The data used to train the predictive model is assumed to be accurate, comprehensive, and representative of real-world scenarios. Any new datasets input through the UI are assumed to be of similar quality and accuracy as the training data. The datasets must include all required features mentioned in the Data Requirements section, ensuring the model has sufficient information to make accurate predictions. The predictive model is assumed to generalize well to new, unseen data, meaning it can accurately predict credit risk for new credit card owners based on the provided features. It is assumed that users will input data correctly and in the required format through the UI, ensuring the model receives valid and useful information for making predictions. The underlying infrastructure, including data storage, processing capabilities, and the user interface, is assumed to be reliable and capable of handling the expected data volume and processing demands. By adhering to these assumptions, the project aims to provide a robust and accurate solution for predicting the probability of credit default based on credit card owners' characteristics and payment history.

# **Design Details**

## **Process Flow**

For accomplishment of the task, we will use a trained Machine Learning model. The process flow diagram is shown below:

**Data Preparation**

**Model**

**Development**

**Deployment**

**Deployment**

**Event Logging** The system must comprehensively log all events to provide users with internal process visibility. The logging process includes:

1. Determining the required logging level.
2. Logging every system flow seamlessly.
3. Allowing developers to select between database and file logging methods.
4. Ensuring system stability under high logging loads to facilitate effective debugging.

**Error Handling** Error handling should clearly communicate issues that deviate from normal usage, providing explanatory messages for each encountered error.

**Performance** The MITVP tool predicts traffic conditions accurately based on numerical traffic volume, serving governmental, non-governmental, and private agencies. Regular model retraining is crucial to continually improve accuracy and avoid misleading authorities.

**Reusability** Code and components should be designed for seamless reuse across projects without compatibility issues.

**Application Compatibility** Python serves as the interface between project components, ensuring efficient data transfer and task execution.

**Resource Utilization** Tasks are optimized to utilize available processing power efficiently without causing system slowdowns.





**Dashboards**

As and when, the system starts to capture the historic/ periodic data for a user, the dashboards will be included display charts over time with progress on various indicators or factors.



**KPIs (Key Performance Indicators)**

* Key Performance Indicators of MITVPCCFP
* Latency or the amount of time the application takes to display results for some specific input.
* The processing power our application takes to run
* The memory and RAM our application takes to run on a web server.

**Conclusion:**

In summary, this report has provided a detailed overview of the project architecture, design specifics, technologies employed, and performance considerations. The CCFP tool delivers instant traffic volume predictions, offering substantial potential benefits to government organizations, agencies, and other stakeholders.