**High-Level Design (HLD)**

**Stores Sales Prediction**

|  |  |
| --- | --- |
| Written By | Avinash Durugkar |
| Version | 1.0 |
| Date |  |

**Document Change Control Record**

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Date** | **Author** | **Comments** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Review

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Date** | **Reviewer** | **Comments** |
|  |  |  |  |

**Approval**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Version** | **Review date** | **Reviewed By** | **Approved By** | **Comments** |
|  |  |  |  |  |

**Abstract**

### This project is aimed at predicting the case of customers default payments in Taiwan. From the perspective of risk management, the result of predictive accuracy of the estimated probability of default will be more valuable than the binary result of classification - credible or not credible clients. We can use the [K-S chart](https://www.listendata.com/2019/07/KS-Statistics-Python.html) to evaluate which customers will default on their credit card payments[¶](https://www.kaggle.com/code/bansodesandeep/credit-card-default-prediction#This-project-is-aimed-at-predicting-the-case-of-customers-default-payments-in-Taiwan.-From-the-perspective-of-risk-management,-the-result-of-predictive-accuracy-of-the-estimated-probability-of-default-will-be-more-valuable-than-the-binary-result-of-classification---credible-or-not-credible-clients.-We-can-use-the-K-S-chart-to-evaluate-which-customers-will-default-on-their-credit-card-payments)

1. Introduction

**1.1 Why these High-Level Design Documents?**

What does high level document mean?

A high-level design document (HLDD) describes the architecture used in the development of a particular software product. It usually includes a diagram that depicts the envisioned structure of the software system. Since this is a high-level document, non-technical language is often used. The HLD will be:

* Present all of the design aspects and define them in detail.
* Describe the user interface being implemented.
* Describe the needed Python libraries for the coding.
* 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 and mildly-technical terms which should be understandable to the administrators of the system

**1.3 Definition**

|  |  |
| --- | --- |
| TERM | Description |
| DB | Database, the cloud platform where the data will be stored. Can be considered  cloud storage. |
| ML | Machine Learning |
| API or APIs | Application Programming Interface can be considered a website link from there we can extract information. |

**2. General Description**

**2.1 Product Perspective**

Big Mart prediction gives us the ability to predict store sale price when given in the features that helps the ML algorithm to predict the sales price of the given entities.

**2.2 Problem Statement**

The aim is to build a predictive model and find out the sales of each product at a particular store. Using this model, BigMart will try to understand the properties of products and stores which play a key role in increasing sales.

**2.3 Proposed Solution**

We will use Exploratory Data Analysis (EDA) to find the Key relations between different attributes and will use a ML algorithm to predict the future sales demand. We can tell the company what are all the challenges they may face, what are the brands or products which is sold the most & other such kind of things, this helps sales team to understand which product to sell & which product to promote & other such kind of things.

**2.4 Data Requirements**

The data was provided by the iNeuron team in their dashboard. The project aimed at predicting the case of customer defaults. For building the ML model we will use the dataset that is given. The data is consisted of 30000 rows and various information about products like

- ID: ID of each client

- LIMIT\_BAL: Amount of given credit in NT dollars (includes individual and family/supplementary credit

- SEX: Gender (1=male, 2=female)

- EDUCATION: (1=graduate school, 2=university, 3=high school, 4=others, 5=unknown, 6=unknown)

- MARRIAGE: Marital status (1=married, 2=single, 3=others)

- AGE: Age in years

- PAY\_0: Repayment status in September, 2005 (-1=pay duly, 1=payment delay for one month, 2=payment delay for two months, … 8=payment delay for eight months, 9=payment delay for nine months and above)

- PAY\_2: Repayment status in August, 2005 (scale same as above)

- PAY\_3: Repayment status in July, 2005 (scale same as above)

- PAY\_4: Repayment status in June, 2005 (scale same as above)

- PAY\_5: Repayment status in May, 2005 (scale same as above)

- PAY\_6: Repayment status in April, 2005 (scale same as above)

- BILL\_AMT1: Amount of bill statement in September, 2005 (NT dollar)

- BILL\_AMT2: Amount of bill statement in August, 2005 (NT dollar)

- BILL\_AMT3: Amount of bill statement in July, 2005 (NT dollar)

- BILL\_AMT4: Amount of bill statement in June, 2005 (NT dollar)

- BILL\_AMT5: Amount of bill statement in May, 2005 (NT dollar)

- BILL\_AMT6: Amount of bill statement in April, 2005 (NT dollar)

- PAY\_AMT1: Amount of previous payment in September, 2005 (NT dollar)

- PAY\_AMT2: Amount of previous payment in August, 2005 (NT dollar)

- PAY\_AMT3: Amount of previous payment in July, 2005 (NT dollar)

- PAY\_AMT4: Amount of previous payment in June, 2005 (NT dollar)

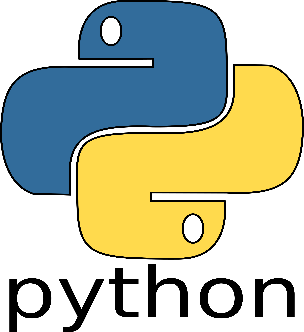
- PAY\_AMT5: Amount of previous payment in May, 2005 (NT dollar)

- PAY\_AMT6: Amount of previous payment in April, 2005 (NT dollar)

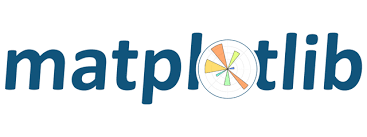
- default.payment.next.month: Default payment (1=yes, 0=no)

**2.5 Tool Used**

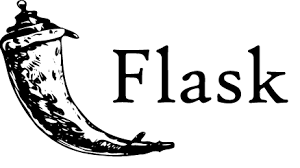
The programming language we used is python as Python is known to be the best programming language for data science, and it is commonly used by big tech companies for data science tasks. we are going to use some other python-based libraries such as NumPy and pandas for data Manipulation data cleaning and for some pre-processing tasks. To perform EDA, we will be switching between seaborn and matplotlib library. For model training we will use various regression-based Machine learning Algorithms such as random forest Classifier, XGBClassifier, AdaBoostClassifier from the very famous Sci-kit learn library. After reaching a decent/good evaluation score we will then save the model using dill Library. Now, for creating an app which we are further going to deploy we will be using Flask as our web framework supported by HTML, and for deployment we will then use Heroku cloud service to deploy our ML model.











**2.6 Constraints**

The System should be user-friendly, the user should get all proper messages while using the web app. He/she also should get a proper error message if he/she has done something wrong on the web-app page. All the errors and results should be delivered in the easiest possible way and all the buttons are going to insert on the webpage should be labelled properly, so the user did not get confused to use the system.

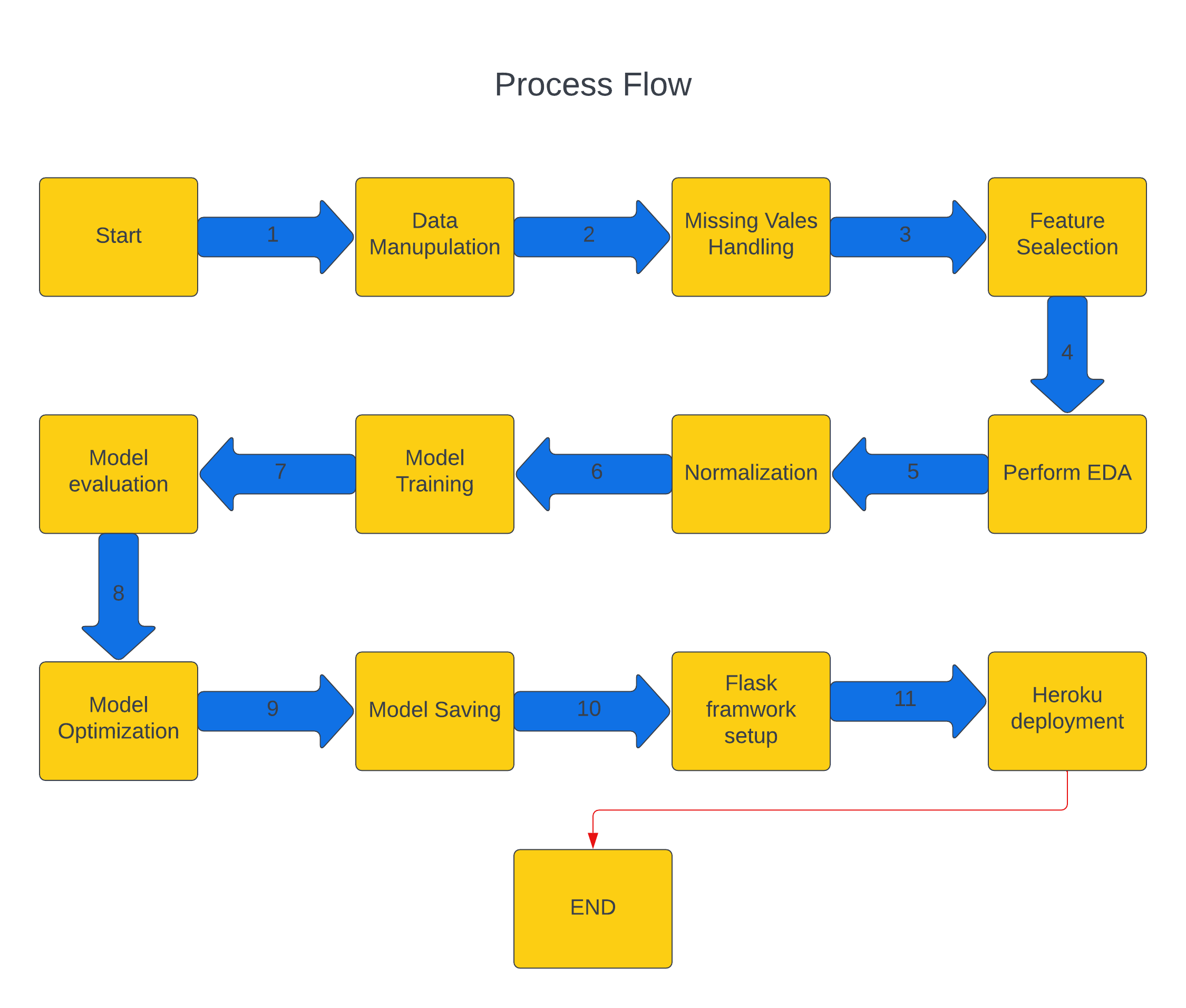
**2.7 Assumptions**

The main objective is to implement a system that will produce approximate future demand for a product in stores.

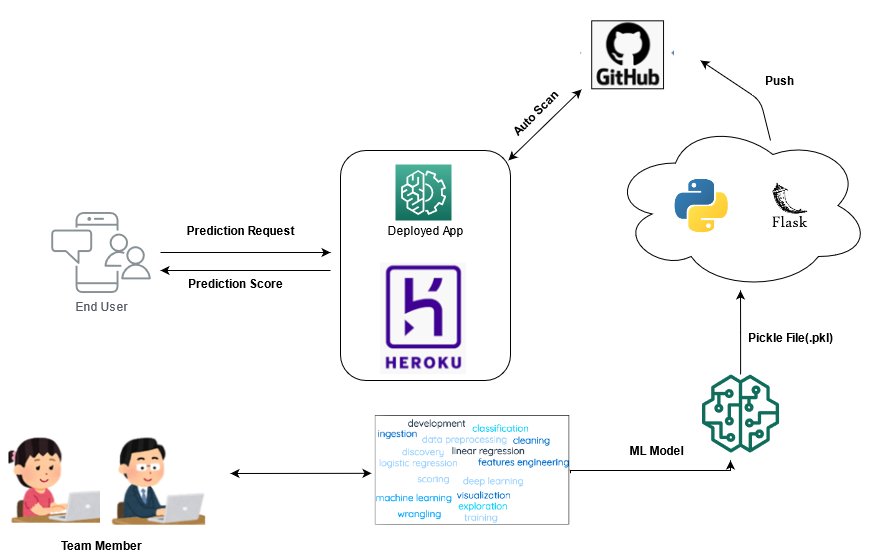
**3. Design Details**

**3.1 Process Flow**

We will be using following process flow for this project. The process flow will be based on End-To-End Machine learning work flow right from data gathering to model deployment.



**3.2 Deployment Process**



**3.3 Error Handling**

If any error occurred in the processing way, then the error message should be shown to the user in a completely non-technical way that can be understandable by any person. And Meaningful error message should be shown, so the user can spot his mistake and re-run the process with improvement. All the errors that are will occur should be handled properly. And we have to log every error for our application and have to manage the same.

**4. Performance**

The accurate prediction of credit card default is contingent upon selecting an appropriate machine learning (ML) algorithm. To achieve this, we will train multiple ML algorithms and carefully evaluate their suitability in predicting the target variable. The performance of our system will be heavily influenced by the quality and relevance of the data we provide to these algorithms. Additionally, the overall performance will depend on the selection of the final model, as well as the successful setup of the web application and deployment server. Ensuring the proper functioning of all these components is crucial for the smooth execution of our program.

**4.1 Reusability**

The code and the module are created during the time of building the project should maintain all coding guidelines and full project code is written in a Modular fashion. Our system should have the flexibility to work properly from any location. And it should handle any improper input value from the user and should give a meaningful error message so the user can correct his/her mistake and enter valid input to get the result. And the system should be reusable in every manner with different types of inputs values that are all are it has been trained.

**4.2 Application Compatibility**

The different libraries and python programming languages are used to build the system. Every library has its own functionality and it should work properly with our fluctuate system. Flask will be used for making the web APIs and HTML/CSS will be used to make the web application. All the components of the application should work properly and it should produce a result without any interpretation.

**4.3 Resource Utilization**

Our application should utilize the given resource properly and it should use a minimal amount of internet to work and call the APIs on the Web page. Our system should not use much number of computational resources hence it will make the application slow. Our application will be deployed cloud platform and it should utilize the resource given on the cloud and work properly.

**5. Deployment**

For the deployment process, we will be using Heroku cloud to Deploy our model. Heroku is a cloud platform that lets companies build, deliver, monitor and scale apps.

**6. Conclusion**

In conclusion, the credit card default prediction model is a crucial tool for evaluating the likelihood of credit card holders defaulting on their payments. By training and evaluating multiple machine learning algorithms, we can identify the most suitable approach for predicting credit card defaults. Based on the evaluation, the Random Forest Classifier algorithm achieved an impressive accuracy score of 84%. The performance of the system heavily relies on the quality and relevance of the training data. The accuracy and effectiveness of the model are further influenced by the chosen algorithm, as well as the successful deployment of the model through a web application and deployment server. By optimizing these components, the credit card default prediction model can provide valuable insights to mitigate the risk of default and facilitate informed decision-making in credit card management.

**7. Reference**

Google images reference to showcase the framework/library used.

lucid chart for drawing process flow and deployment process.