**High-Level Design (HLD)**

**Stores Sales Prediction**

|  |  |
| --- | --- |
| Written By | Suhail Fazal |
| Version | 1.0 |
| Date | 18 Feb 2022 |

**Document Change Control Record**

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

Review

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

**Approval**

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

**Abstract**

Machine Learning is a category of algorithms that allows software applications to become more accurate in predicting outcomes without being explicitly programmed. The basic premise of machine learning is to build models and employ algorithms that can receive input data and use statistical analysis to predict an output while updating outputs as new data becomes available. These models can be applied in different areas and trained to match the expectations of management so that accurate steps can be taken to achieve the organization’s target.

In this paper, the case of Big Mart has been discussed to predict the sales of different types of items and for understanding the impact of different factors on item sales.

By mining the data store from the data warehouse, more anomalies and common patterns can be discovered. Taking various aspects of a dataset collected for Big Mart, and the methodology followed for building a predictive model, results with high levels of accuracy are generated, and these observations can be employed to make decisions to improve sales.

1. Introduction

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

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

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**

The Store Sales Prediction is an ML infused Web Application that is capable of predicting future product demand by analysing past records. It predicts the sales of different stores of Big Mart.

**2.2 Problem Statement**

Nowadays, shopping malls and Big Marts keep track of individual item sales data in order to forecast future client demand and adjust inventory management. In a data warehouse, these data stores hold a significant amount of consumer information and particular item details.

**2.3 Proposed Solution**

Approach is to find relationship between features and output, aligning it with business understanding. With Correlation Analysis relationship between features can be understood, case of multicollinearity can be handled. Random Forest Regressor with Randomised Search CV is the best model with highest score and lowest error, this model will be used to predict future sales demand.

A dedicate web application has been created for the sole purpose of providing insights pertaining to store sales in future with input features getting processed at the back end to predict the output. The machine learning algorithm mentioned has been hyperparameter tuned in order to find the best parameters responsible for predicting store sales with minimal error.

**2.4 Data Requirements**

The data is required for the building of the project is already available on the dashboard. The Store Sales Prediction data is recorded many product descriptions along with past sales quantity. For building the ml model given dataset will be used. The data is consisting 8523 rows and various information about products like product id, product category, outlet id, outlet location, outlet type etc.

**2.5 Further Improvements**

Save User’s history of prediction and identify future demand.

**2.6 Tools Used**

- For Programming Python is used, which provides some of the most powerful libraries and packages in order to predict store sales.

- Jupyter notebook is used as python ide

- For Data Visualization, plots and charts Seaborn and matplotlib is used, Klib library for data cleaning.

- Flask is used as web framework for building the web application.

- Front end development is done using HTML

- Python is used for backend development

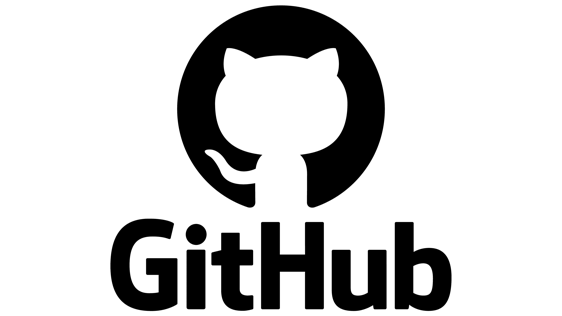
- Github is used to store the files.

- Heroku is used to deploy the model.

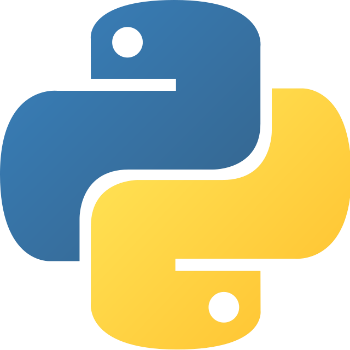
These libraries include but are not limited to Pandas, NumPy, Sklearn, Klib Data Cleaning library, Matplotlib and Seaborn for visualisation. Flask web framework along with PyCharm has been leveraged for creating a dedicated application and for all modular coding. All the files are stored on Github for easy access and efficient extraction.

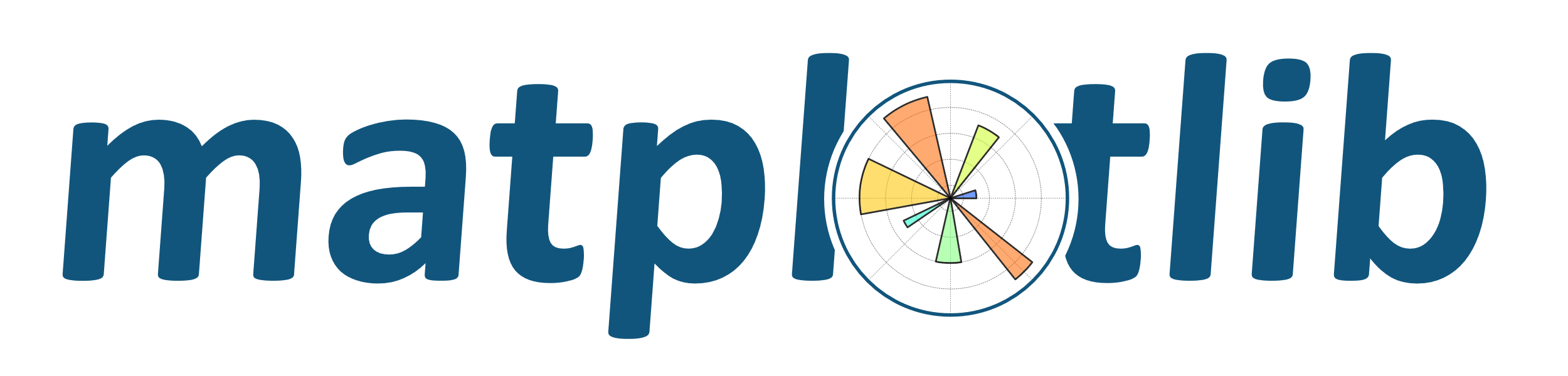




 A picture containing icon

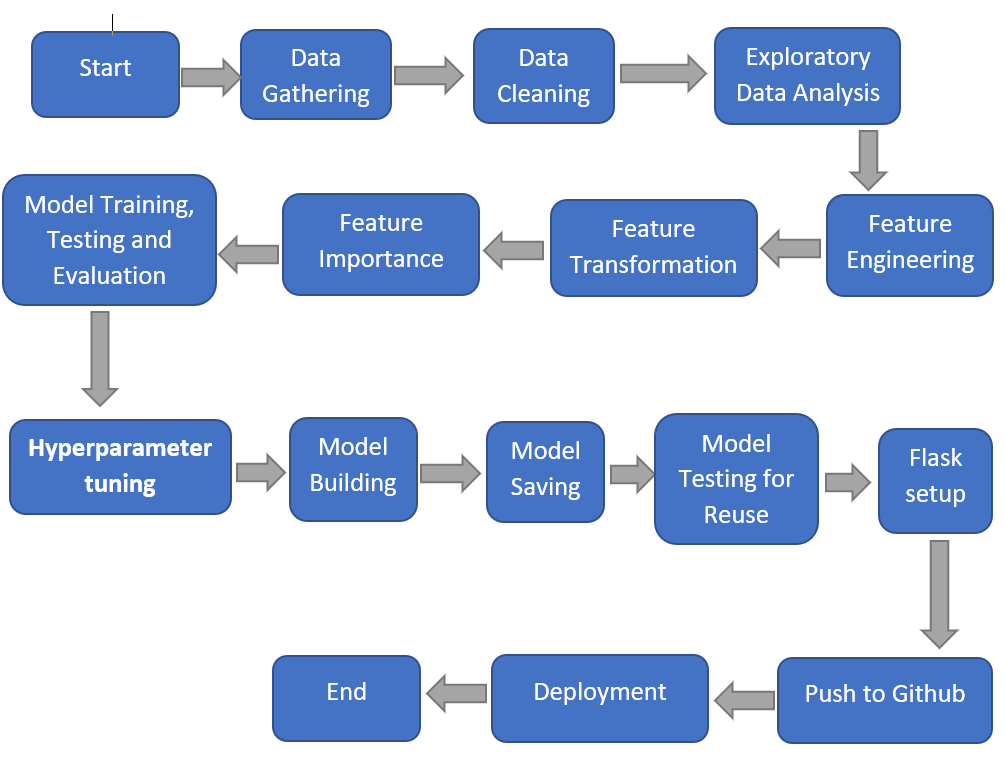
Description automatically generated



**3. Design Details**

**3.1 Process Flow**



**3.2 Deployment Process**



**3.3 Error Handling**

If any error occurs in inputting data (in web application), the resulting error message should be concise and understandable by users. This will enable the user to spot data entry errors and rerun the process with necessary amendments. And we have to log every error for our application and have to manage the same. Necessary log files have been created to keep a record of all data entered by the user each time the application is used for predicting item sales.

**4. Performance**

The Stores Sales Price Prediction is solely dependent on evidence-based machine-learning algorithms. Trained various ML algorithms to find the best performing model i.e. Random Forest Regressor in order to predict the target feature Random Forest Regressor gave 71.5% R score and RMSE of 0.5497, System performance will be based on the data fed to the algorithms. Model performance will involve the finalized model, the web application and the deployment server collectively.

**4.1 Reusability**

The code and the modules created at the time of building the project is required to maintain all coding guidelines and full project code is written in a modular fashion. Our system incorporates the flexibility to work properly from any location. And it should handle any improper input value from the user by providing a meaningful error message so the user can correct his/her mistake and enter valid input to get the desired results.

**4.2 Application Compatibility**

The different libraries in Python programming language, and HTML have been used to build the system. Flask has been used for making the web APIs and HTML has been leveraged to make the web application. All the components of the application are supposed to work properly and it is required to produce a result without any major impediments in place.

**4.3 Resource Utilization**

Our application should utilize the given resources efficiently and effectively. It should use a optimal amount of internet to work and call the APIs on the web page. Our system has been designed to use less computational in order to make the application faster. Our application will be deployed on cloud platform and it should utilize the resource given on the cloud and perform effectively.

**5. Deployment**

Heroku cloud platform is used for deployment and hosting the application, cloud platform provides flexibility to run and use the application globally.

App link-[Store Sales Prediction - INeuron](https://ineuron-storesalesprediction.herokuapp.com/)



**6. Conclusion**

* The Store Sales Prediction consists of correlation analysis and feature importance which help us understand the impact various features have on the target variable (Item Outlet Sales).
* Categorical variables and their contributions towards sales explain their importance as well.
* Frequency distribution, outliers, missing values and their effects understood during the model helped prepare the data for further modelling.
* Item Mrp can be further segmented into 4 different groups.
* Item Mrp is positively correlated with Item Outlet Sales.
* Linear Regression, Ridge and Lasso Regression, Random Forest Regressor were applied, Random Forest Regressor was hyperparameter tuned to get the best parameters and best score.
* Highest score of 71.5% was with Random Forest Regressor using Randomised Search CV.

**7. Reference**

Google image for collection the logos and images.

Sketch diagram for drawing the diagrams.