High Level Design

Thyroid Disease Detection

Revision number: 1.0

Last date of revision: 08/02/2022

Afrin R Peshimam

# Document Version Control

|  |  |  |  |
| --- | --- | --- | --- |
| **Date Issued** | **Version** | **Description** | **Author** |
| 08/02/2022 | 1 | Initial HLD-V1.0 | AFRIN R PESHIMAM |

**Contents**

[Document Version Control 2](#_Toc16093)

[Abstract 4](#_Toc16094)

[1. Introduction 5](#_Toc16095)

[1.1. Why this High-Level Document? 5](#_Toc16096)

[1.2. Scope 5](#_Toc16097)

[1.3. Definitions 5](#_Toc16098)

[2. General Description 6](#_Toc16099)

[2.1 Product Perspective 6](#_Toc16100)

[2.2 Problem Statement 6](#_Toc16101)

[2.3 Proposed Solution 6](#_Toc16102)

[2.4 Data Requirements 6](#_Toc16103)

[2.5 Tools Used 7](#_Toc16104)

[2.6 Constraints 8](#_Toc16105)

[2.7 Assumptions 8](#_Toc16106)

[3. Design Details 9](#_Toc16107)

[3.1 Process flow 9](#_Toc16108)

[3.1.1 Model Training & Evaluation 9](#_Toc16109)

[3.1.2 Deployment Process 10](#_Toc16110)

[3.2 Event log 10](#_Toc16111)

[3.3 Error Handling 10](#_Toc16112)

[4. Performance 11](#_Toc16113)

[4.1 Reusability 11](#_Toc16114)

[4.2 Application Capability 11](#_Toc16115)

[4.3 Resource Utilization 11](#_Toc16116)

[4.4 Deployment 11](#_Toc16117)

[5. Conclusion 12](#_Toc16118)

[6. References 13](#_Toc16119)

# Abstract

Build a predictive model and find out the sales of each product at a particular store

# Introduction

## Why this High-Level Document?

The purpose of this High-Level Design (HLD) Document is to add necessary details to the current project description to represent a suitable model for coding. This model 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 the design aspects and define them in detail
* Describe the user interface being implemented - Describe the hardware and software interfaces
* Describe the performance and 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

## Scope

The HLD documentation presents the structure of the system, such as the database architecture, application architecture, application flow (Navigations), and technology architecture. The HLD uses non-technical to miIdIy-technical term which should be understandable to the administrator of the system.

## Definitions

**TERM DESCRIPTION**

|  |  |
| --- | --- |
| **Database** | Collection of all the information monitored by this system |
| **IDE** | Integrated Development Environment |
| **AWS** | Heroku |

# General Description

## Product Perspective

The Big mart sales prediction is a machine learning based regression model which will help us to detect the sales of each product’s, across various stores.

## Problem Statement

The main goal of this project is to predict the sales of various products, in each store’s. It will play a decisive role in Inventory management and reducing operational cost.

## Proposed Solution

The proposed solution for this project is Machine learning algorithms can be implemented to predict the sales. Considering various features like Item Identifier Item Weight, Item Fat,Item Visibility, Item Type, Item MRP, Outlet Identifier, Outlet Establishment Year, Outlet Size, Outlet Location, Outlet Type as inputs from the web app, the implemented regression model will predict the output as Item Outlet Sales.

Here, we have used XGBOOST Regressor to predict the sales.

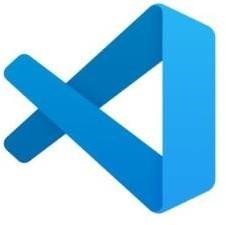
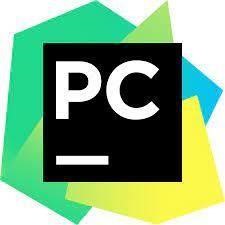
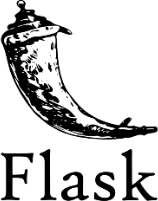
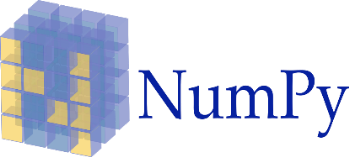
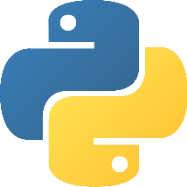
## Data Requirements

There are a total of 8,523 records in the training set and 5,681 records in the test set.

|  |  |
| --- | --- |
| **Data** | **Description** |
| Item Identifier | Unique product ID |
| Item Weight | Weight of product |
| Item Fat | Low Fat or Regular |
| Item Visibility | The % of total display area of all products |
| Item Type | The category to which the product belongs |
| Item MRP | Maximum Retail Price |
| Outlet Identifier | Unique Store ID |
| Outlet Establishment Year | Year store was created |
| Outlet Size | The size of store in term of ground area. |
| Outlet Location | Type of City store is located |
| Outlet Type | Whether outlet is grocery or supermarket |
| Item Outlet Sales | Sales of Product |

## Tools used

Python programming language and frameworks such as NumPy, Pandas, Scikit-learn, Flask, Heroku, Git.



* PyCharm is used as IDE.
* For visualization of the plots, Matplotlib, Seaborn and Plotly are used.
* Heroku is used for deployment of the model.
* MongoDB is used to retrieve, insert, delete, and update the database • Frontend development is done using HTML/CSS
* Python Flask is used for backend development.
* GitHub is used as version control system.

## Constraints

The sales prediction application must be user friendly, as automated as possible and users should not be required to know any of the workings.

## Assumptions

A machine learning-based classification model has been used for detecting the use case based on the user input. It is also assumed that all aspects of this project have the ability to work together as the designers is expecting.

# Design Details

## Process Flow

For predicting the thyroid disease, we will use classification model. Below is the process flow diagram is as shown below.

Proposed Methodology

Capture the data

from the web app

Training /

validation on

dataset

ML model for

classification

Prediction of the

output

### Model Training and Evaluation

**Data collection**

**from source**

**Insertion**

**into**

**database**

**Data pre**

**-**

**processioning /**

**Feature Eng.**

**Split the data**

**into train & test**

**Feature**

**selection**

**Train the**

**model**

**Test the**

**model**

**Improve the**

**model**

### Deployment Process

Data

ML model

training

Model

prediction

Flask

HTTP /

Rest API

WSGI

HTTP

server

Heroku

User

## Event Log

The system should log every event so that the user will know that process is running internally.

**Initial Step-By-Step Description:**

* The system identifies at what step logging required.
* The system should be able to log each and every system flow.
* Developer can choose logging method. You can choose database logging / File logging as well.
* System should not hang even after using loggings. Logging just because we can easily debug issues, so logging is mandatory to do.

## Error Handling

Should error 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

# Performance

We can observe that the r2 of the predicted output was seen at0.89 using XGBOOST CLASSIFIER and RMSE of 532.

## Reusability

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

## Application Compatibility

The different components for this project will be using 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.

## Resource Utilization

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

## Deployment



# Conclusion

In this project, about machine learning classification models are evaluated for the given dataset to predict the sales. The data was used to develop the classification models, and the predicted the output from these models by comparing with actual target to check the accuracies of these models. It has been found that XGBOOST model which is built upon decision tree is the best performing the model among them and improved the model further with hyper parameter tuning.

# References