i

Low Level Design

Store Sales Prediction

|  |  |
| --- | --- |
| Written By | Mohammad Salman |
| Document Version | 1.0 |
| Last Revised Date | 26-11-2022 |

Store Sales Prediction

LOW LEVEL DESIGN (LLD)

# **Document Control**

|  |
| --- |
| ii |

**Change Record:**

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

**Reviews:**

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

**Approval Status:**

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

FOOD RECOMMENDATION LLD

iii

# Contents

1. **Introduction ............................................................................................................................ 1** 
   1. **What is Low-Level design document? ................................................................................ 1**
   2. **Scope ............................................................................................................................... 1**
2. **Architecture ............................................................................................................................ 2**
3. **Architecture Description .......................................................................................................... 3** 
   1. **Data Description ....................................................................................................... …….. 3**
   2. **Data Gathering ................................................................................................................. 4**
   3. **Raw Data Validation .......................................................................................................... 4**
   4. **Data Transformation......................................................................................................... 4**
   5. **Data Insertion into Database............................................................................................. 4**
   6. **Data Preprocessing ........................................................................................................... 4**
   7. **Feature Engineering ........................................................................................................... 5**
   8. **Parameter Tuning .............................................................................................................. 5**
   9. **Model Building ................................................................................................................ 5**
   10. **Model Saving ................................................................................................................... 5**
   11. **Flask Setup for Data Extraction ........................................................................................ 5**
   12. **GitHub ............................................................................................................................ 5**
   13. **Deployment .................................................................................................................... 5**

# 1. Introduction

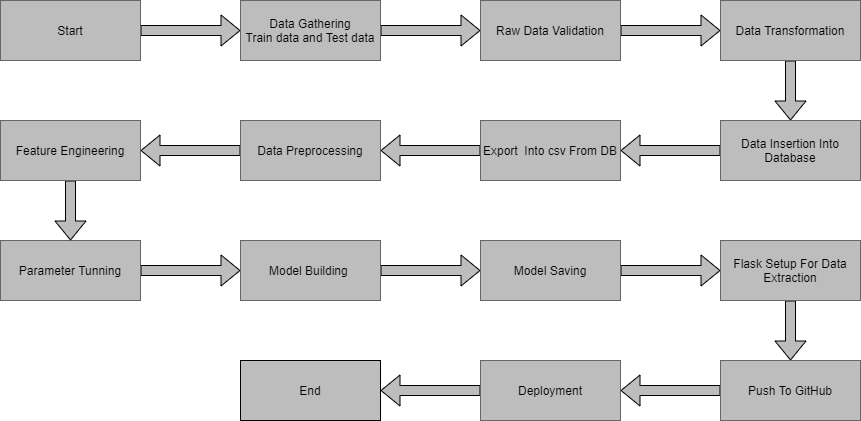
1.1.What is Low-Level design document?

The goal of LLD or a low-level design document (LLDD) is to give the internal logical design of the actual program code for Food Recommendation System. LLD describes the class diagrams with the methods and relations between classes and program specs. It describes the modules so that the programmer can directly code the program from the document.

1.2.Scope

Low-level design (LLD) is a component-level design process that follows a step-bystep [refinement](https://en.wikipedia.org/wiki/Refinement_(computing)) process. This process can be used for designing data structures, required software architecture, source code and ultimately, performance algorithms. Overall, the data organization may be defined during requirement analysis and then refined during data design work

# 2.Architecture



# 3. Architecture Description

## 3.1 Data Description

Given is the variable name, variable type, the measurement unit, and a brief description. The concrete compressive strength is the regression problem. The order of this listing corresponds to the order of numerals along the rows of the database.

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Measurement |
| Item\_Identifier | String | Unique product ID |
| Item\_Weight | Float | Weight of product |
| Item\_Fat\_Content | String | Whether the product is low fat or not |
| Item\_Visibility | Float | The % of a total display area of all products in a store allocated to the particular product |
| Item\_Type | String | The category to which the product belongs |
| Item\_MRP | Float | Maximum Retail Price (list price) of the product |
| Outlet\_Identifier | String | Unique store ID |
| Outlet\_Establishment\_Year | Integer | The year in which the store was established |
| Outlet\_Size | String | The size of the store in terms of ground area covered |
| Outlet\_Location\_Type | String | The type of city in which the store is located |
| Outlet\_Type | String | Whether the outlet is just a grocery store or some sort of supermarket |
| Item\_Outlet\_Sales | Float | Sales of the product in the particular store. This is the outcome variable to be predicted. |

## 3.2 Data Gathering

Data source: <https://www.google.com/url?q=https://archive.ics.uci.edu/ml/datasets/Bike%2BSharing%2BDataset&sa=D&source=apps-viewer-frontend&ust=1669523611018571&usg=AOvVaw2w2CNen9aLPM9FfBoI2J16&hl=en>

Train and Test data are stored in .csv format.

## 3.3 Raw Data Validation

After data is loaded, various types of validation are required before we proceed further with any operation. Validations like checking for zero standard deviation for all the columns, checking for complete missing values in any columns, etc. These are required because The attributes which contain these are of no use. It will not play role in contributing to the sales of an item from respective outlets.

Like if any attribute is having zero standard deviation, it means that’s all the values are the same, its mean is zero. This indicates that either the sale is increasing or decrease that attribute will remain the same. Similarly, if any attribute is having full missing values, then there is no use in taking that attribute into an account for operation. It’s unnecessary increasing the chances of dimensionality curse.

## 3.4 Data Transformation

Before sending the data into the database, data transformation is required so that data are converted into such form with which it can easily insert into the database.

## 3.5 Database Insertion

Both train and test data set are inserted into the database. Here MongoDB database is used to store the data set. Separate collections were created for both train and test sets.

## 3.6 Export as `CSV` from Database

From the database both the train and test data set are exported into the local system and stored into CSV files. Now this CSV file will have proceeded for further processing.

## 3.7 Data Preprocessing

* Season: Month column has a direct mapping with season (Winter: January to March, Summer: April to June, Fall: July to September and Spring: October to December). Hence we will drop season column
* Holiday and 'day': workingday = weekday and not a holiday. Since we noticed that there were two kinds of bike rental behavoirs - during working days and not a working day, we will retain only the workingday column and drop 'day' and 'holiday' column
* Workingday: After observing the bike rental trend, we propose to build 2 separate models for 1. if it is a working day, and 2. if it is a non-working day. Hence, we can separate out the data based on this column and drop the column
* Weather: Split weather column to weather\_1, weather\_2 and weather 3 (recall that we had relabelled all the weather = 4 data points to weather = 3 due to its sparseness). Drop weather\_3 since it is a function of the rest of the weather columns
* Temp: temp and atemp are highly correlated. Hence retain only the temp column
* Windspeed: Very poorly correlated with count. Hence drop this column
* Casual and registered: These are individual components of our to be predicted column (count). Hence drop these columns
* Month: Split month column to month\_1, month\_2, ..., month\_12. Drop month\_12 since it is a function of the rest of the month columns
* Date: Intuitively, there is should be no dependency on date. Hence drop this column
* Hour: Split hour column to hour\_0, hour\_1, ..., hour\_23. Drop hour\_23 since it is a function of the rest of the hour columns

## 3.8 Feature Engineering

After preprocessing it was found that some of the attributes are not important to the item sales for the particular outlet. So those attributes are removed. Even one hot encoding is also performed to convert the categorical features into numerical features.

## 3.9 Parameter Tuning

Parameters are tuned using Randomized searchCV. Four algorithms are used in this problem, Linear Regression, Gradient boost, Random Forest, and XGBoost regressor. The parameters of all these 4 algorithms are tunned and passed into the model.

## 3.10 Model Building

After doing all kinds of preprocessing operations mention above and performing scaling and hyperparameter tuning, the data set is passed into all four models, Linear Regression, Gradient boost, Random Forest, and XGBoost regressor. It was found that Gradient boost performs best with the smallest RMSE value i.e. 587.0 and the highest R2 score equals 0.55. So ‘Gradient boost’ performed well in this problem.

**3.11 Model Saving**

Model is saved using pickle library in `.pkl` format.

## 3.12 Flask Setup for Data Extraction

After saving the model, the API building process started using Flask. Web application creation was created here. Whatever the data user will enter and then that data will be extracted by the model to predict the prediction of sales, this is performed in this stage.

**3.13 GitHub**

The whole project directory will be pushed into the GitHub repository.

## 3.14 Deployment

The cloud environment was set up and the project was deployed from GitHub into the Heroku cloud platform.

App link-**https://bike-new-share.herokuapp.com/**