**Architecture Design**

**STORE SALES PREDICTION**

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

**Document Control**

Change Record:

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

**Approval Status:**

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

**Index**

|  |  |
| --- | --- |
| **Content** | **Page No** |
| Abstract | 3 |
| 1. Introduction | 3 |
| 1.1 What is Architecture Design? | 3 |
| 1.2 Scope | 3 |
| 1.3 Constraints | 4 |
| 2. Technical Specification | 5 |
| 2.1 Dataset | 5 |
| 2.2 Logging | 7 |
| 2.3 DataBase | 7 |
| 2.4 Deployment | 7 |
| 3. Technology Stack | 8 |
| 4. Proposed Solution | 8 |
| 5. Architecture | 8 |
| 5.1 Architecture Description | 9 |
| 6. Model Training/Validation Workflow | 11 |
| 7. User Input/output Workflow | 10 |

# 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.

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.

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 What is Architecture Design?**

The goal of Architecture Design (AD) or a low-level design document is to give the internal design of the actual program code for the `Store Sales Prediction Project`. AD describes the class diagrams with the methods and relation between classes and program specification. It describes the modules so that the programmer can directly code the program from the document.

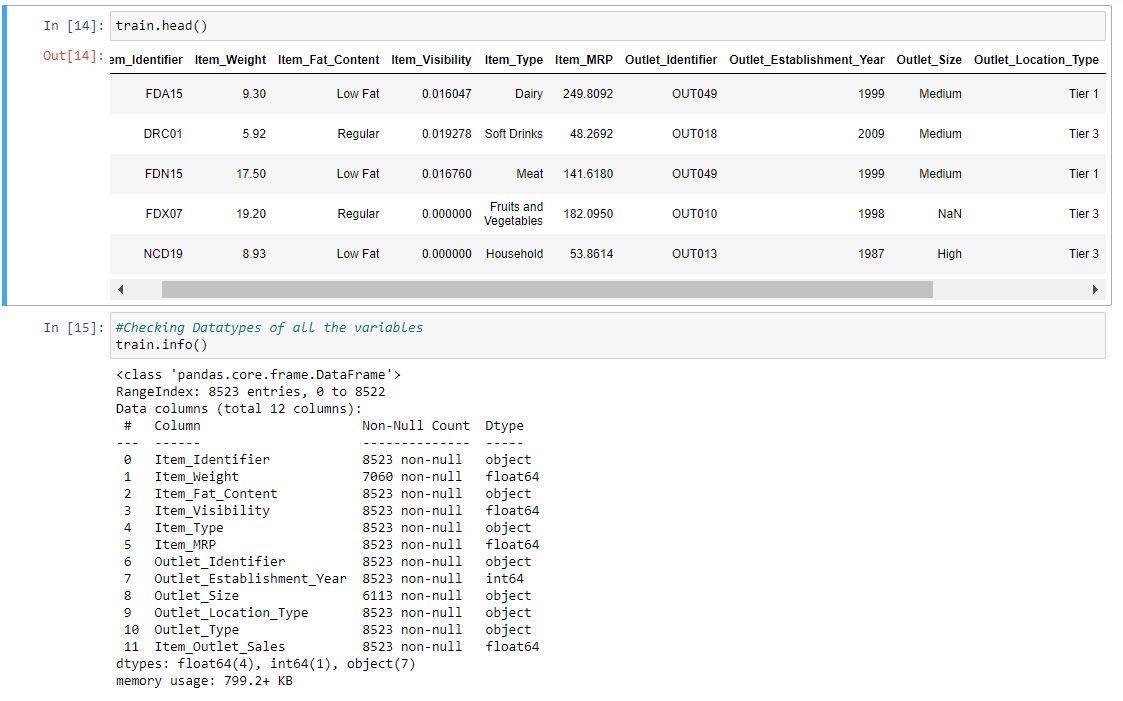
## 1.2 Scope

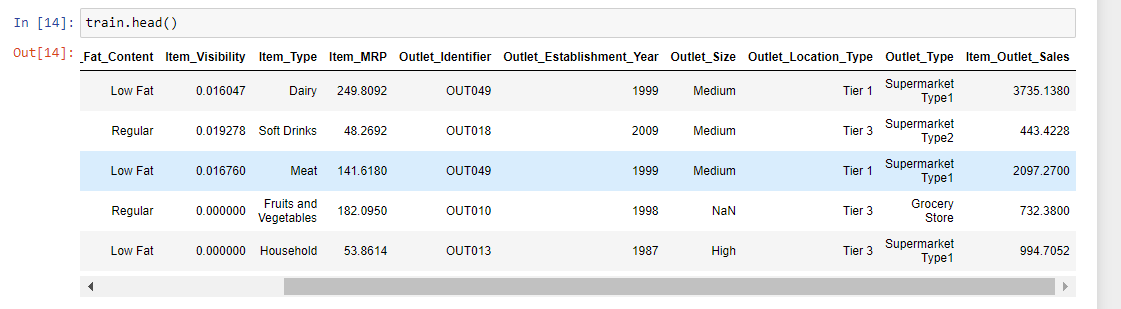
Architecture Design (AD) is a component-level design process that follows a step-by-step refinement process. This process can be used for designing data structures, required software, architecture, source code, and ultimately, performance algorithms. Overall, the data organization may be defined during requirement analysis and then refined during data design work. And the complete workflow.

# 2. Technical Specification

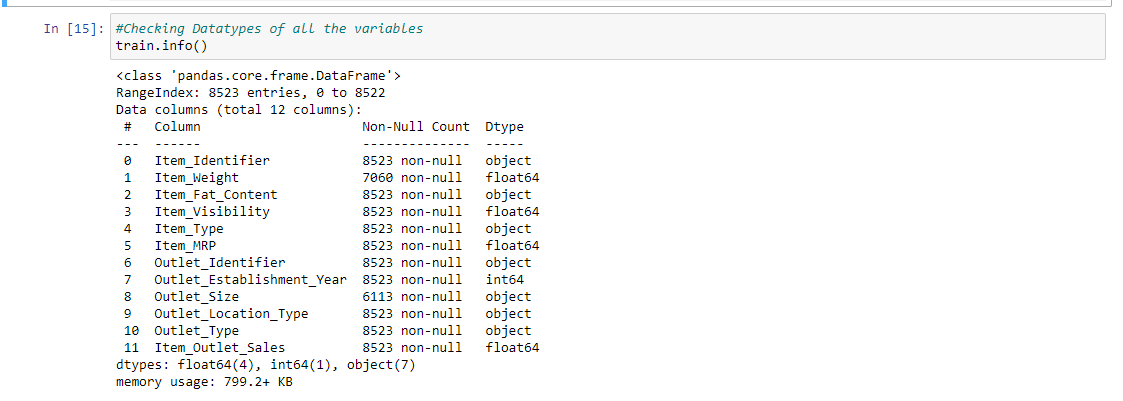
## 2.1 Dataset

Big Mart’s data scientists collected sales data of their 10 stores situated at different locations with each store having 1559 different products as per data collection. Using all the observations it is inferred what role certain properties of an item play and how they affect their sales. The dataset looks like this:





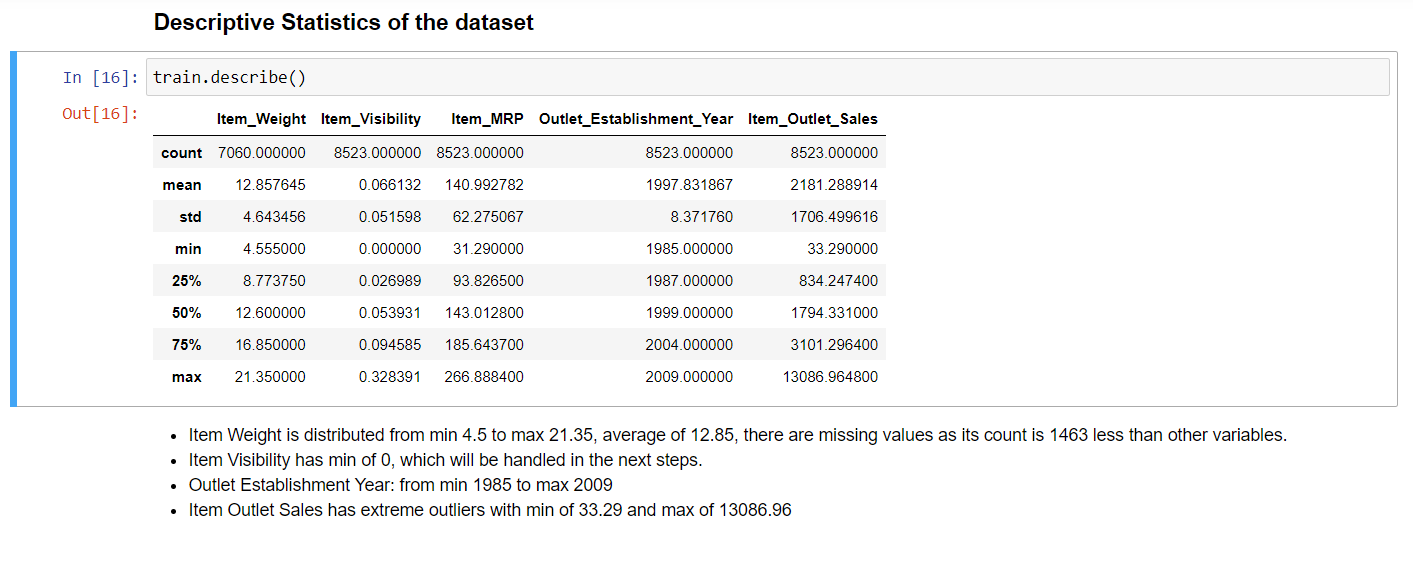
The data set has integers, float values and object datatypes as shown in the Fig.



In the raw data, there can be various types of underlying patterns which also gives an in-depth knowledge about the subject of interest and provides insights into the problem. But caution should be observed with respect to data as it may contain null values, or redundant values, or various types of ambiguity, which also demands pre-processing of data. The dataset should therefore be explored as much as possible.

Five-point summary, Quartiles, Mean, Standard Deviation, Value counts, Minimum and Maximum value, etc. are shown below for numerical attributes.

Data Pre-processing includes analysing independent variables to check for null values in each column and then replacing or filling them with supported appropriate values so that analysis and model fitting is not hindered from their way to accuracy. Shown above are some of the representations obtained by using Pandas tools which tell about variable count for numerical columns. Maximum and minimum values in numerical columns, along with their percentile values for median, play an important factor in deciding which value to be chosen at priority for further exploration tasks and analysis. Data types of different columns are used further in label processing and a one-hot encoding scheme during the model building.



## 2.2 Logging

We should be able to log every activity done by the user

* The system identifies at which step logging require.
* The system should be able to log each and every system flow.
* Developers can choose logging methods. Also, can choose database logging.
* The system should be not be hung even after using so much logging. Logging just because we can easily debug issuing so logging is mandatory to do.

## 2.3 DataBase

The system needs to store every request into the database and we need to store it in such a way that it is easy to retain and look into the records.

The system should capture every data that any user gave and the prediction that has been made by that input.

**2.4 Deployment**

Heroku is used to deploy this project.



## 3. Technology Stack

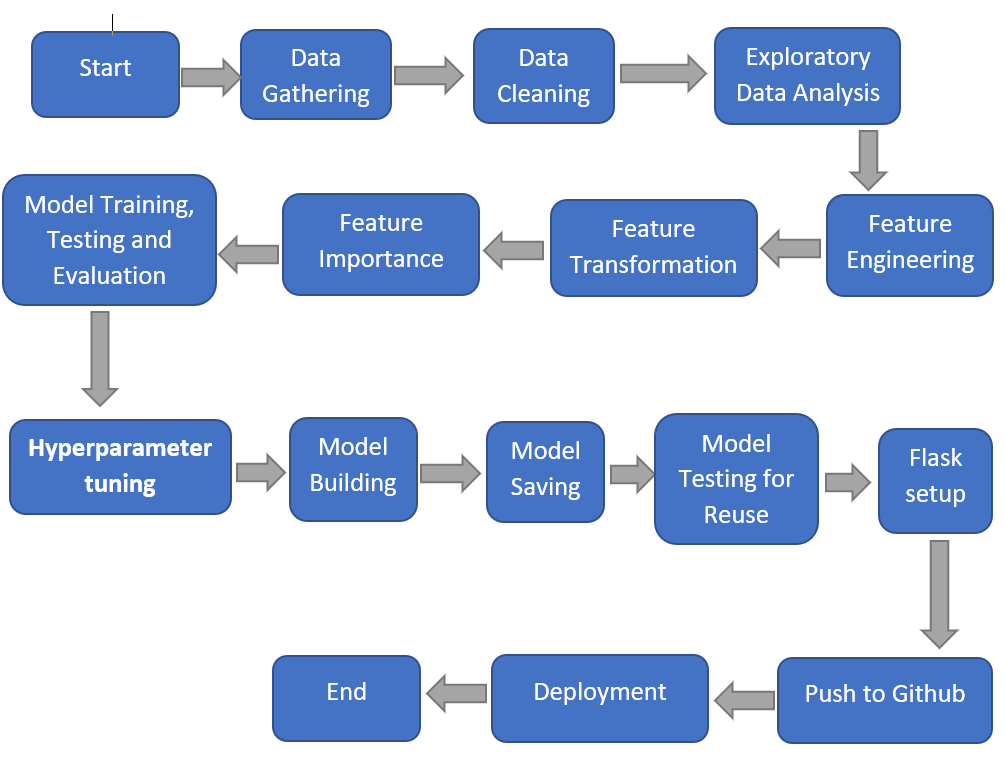
|  |  |
| --- | --- |
| **Front End** | HTML |
| **Backend** | Python/ Flask |
| **Deployment** | Heroku |

## 4. Proposed Solution

Perform Exploratory Data Analysis, find the relationship between attributes, and predict future sales using machine learning algorithms. Clients can fill the required info as inputs and get results through the web application.

Features are passed to the backend where they are pre-processed and then passed to hyperparameter tuned model to predict the output.

## 5. Architecture



### 5.1 Data Gathering

Data source: INeuron Internship Portal, Train and Test data are stored in .csv format.

### 5.2 Data Cleaning

* The Data might contain Repeated/Duplicate entries, removing them.
* Data Validity check, Item Visibility has 0 as minimum value, all items need to have some visibility.
* Missing values imputation of the products will give better output than deleting them.
* Outliers – Outliers can be valid or a data entry mistake, treating them with transformations to convert skewed data to normal distribution.

### 5.3 Exploratory Data Analysis

### Uni-variate analysis = Checking how each variable is distributed.

### Bi-variate/Multi-variate analysis = how variables are related with each other and with output variable, which is Item Outlet Sales in this case.

### Pivots – to check variables and their contribution/importance on the output variable.

### Visualization and Data Insights – Visualising data makes it much easier to interpret the data.

### 5.4 Feature Engineering

Feature Creation:

* Creating new features from “Outlet Establishment Year” as “Outlet Year” subtract it with 2013 to get the age of the outlet when data was collected.
* New Item Type attribute to be created after extracting first 2 characters from the “Item Identifier” feature. FC: Food, DR: Drinks, and NC: Non-Consumables.
* Mapping and combining: Item Fat Content
  + 'LF', 'low fat’ to “Low Fat”
  + 'reg' to “Regular”

Feature Transformation:

* Outlier handling – Log Transformation.
* Dropping “Outlet Establishment Year”, “Item Identifier” and “Outlet Identifier”.
* Label Encoding.
* One Hot Encoding.
* Feature Scaling.
  + Feature scaling is scaling down the data of all the

### 5.5 Feature Importance - using Extra Trees Regressor to plot features having more impact on the outlet sales.

### 5.6 Model Training and Testing - Trained the model on Linear Regression, Ridge and Lasso Regression, and Random Forest Regressor. Best model with highest Prediction score (R Squared) and Lowest Error (Root Mean Squared Error) is selected.

### 5.4 Model Evaluation – Predict and Evaluate the model on validation dataset.

**5.5 Hyperparameter tuning -** tuning parameters to get the best score and best parameters combinations using Randomised Search Cross Validation.

**5.6 Model Building -** Building the model with suggested parameters from Hyperparameter tuned model, testing and evaluating the model. Tuned Random Forest Regressor got highest accuracy and lowest error score. With 71.56% R Squared, and RMSE of 0.5499.

**5.7 Saving the model -** Model issaved in pickle format as pkl.

**5.8 Model Testing for Reuse -** Predicting from the saved pickle file to validate if it’s working.

### 5.9 Flask Setup - Web application was created using Flask, which takes user inputs and passes it to the model to predict sales.

**6.0 Push to GitHub -** Project Directory will be pushed to Github.

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

App link- [**https://salespredictionapp.herokuapp.com/**](https://salespredictionapp.herokuapp.com/)

**7. User Input / Output Workflow.**

