Architecture Design

# STORE SALES PREDICTION

| Written By | Raghav Kachroo |
| --- | --- |
| Document Version | 1.0 |
| Last Revised Date | 02-Jan-2022 |

**Document Control**

Change Record:

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

**Approval Status:**

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

# Index

| **Content** | **Page No** |
| --- | --- |
| Abstract | 4 |
| 1. Introduction | 4 |
| 1.1 What is Architecture Design? | 4 |
| 1.2 Scope | 4 |
| 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 | 12 |

**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, a one-stop-shopping- center, has been discussed to predict the sales of different types of items and for understanding the

effects of different factors on the items’ sales. 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.

# Introduction

## 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 `. 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.

## 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.

## Constraints

The Store Sales Prediction System should be easy to use and should give an accurate prediction regarding the sales.

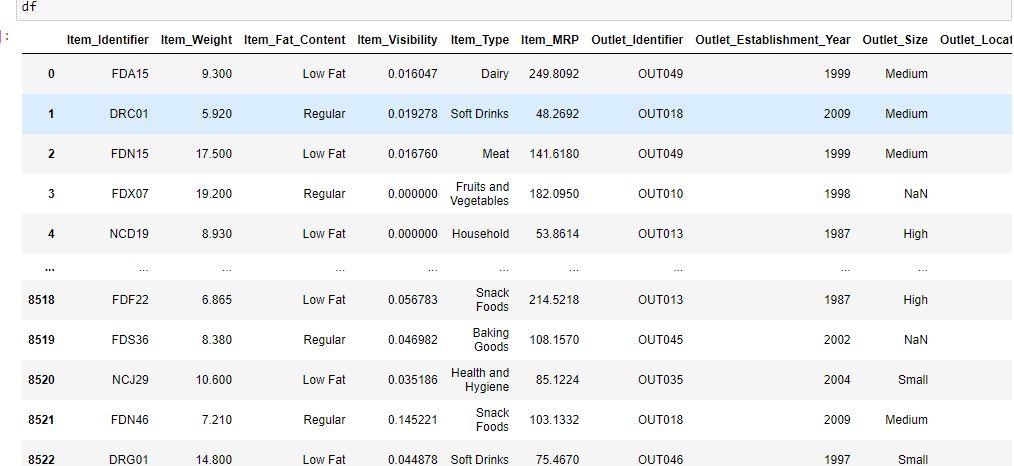


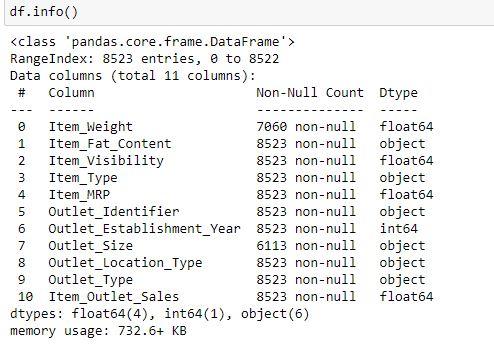
# Technical Specification

## 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 as follow:

The data set consists of various data types from integer to floating to object as shown in 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.

The sales data set consists of 8523 training data sets with 12 feature variables

In data preprocessing all the processes required before sending the data for model building are performed. Like, here the ‘Item Visibility’ attribute is having some values equal to 0, which is not appropriate because if an item is present in the market, then its visibility can be 0. So, it has been replaced with the average value of the item visibility of the respective ‘Item Identifier’ category. New attributes were added named ‘’Outlet years”, where a given establishment year is subtracted from the current year. New “Item Type” attribute was added which just takes the first two characters of the Item Identifier which indicates the types of the items. Then mapping of “Fat content” is done based on ‘Low’ and ‘Reg’.

## Logging

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

* + - The system identifies at which step logging is required.
    - 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 not be hung even after using so much logging. Logging just because we can easily debug issuing so logging is mandatory to do.

## Deployment

For the hosting of the project, we will use Heroku



## Technology Stack

| **Front End** | HTML/JavaScript |
| --- | --- |
| **Backend** | Python/ Streamlit |
| **Database** | MongoDB |
| **Deployment** | Heroku |

1. **Proposed Solution**

After data processing we apply the algorithms and build a model to check accuracy and the model with least error will be selected to be dumped in a pickle so that we can deploy the model through the Heroku console which will make the model user friendly overall.

The user can just input the values on the Front end page and the model can draw out results.

For data processing, some of the concepts of missing values, outliers and correlation is kept in mind while altering the dataset and for model training regression algorithms are used to get the best accuracy.

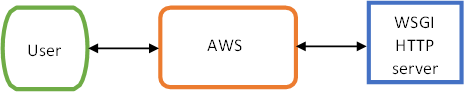
## Architecture (Training)

## 

## Architecture (Deployment)

## 





### Data Gathering

Data source: <https://www.kaggle.com/brijbhushannanda1979/bigmart-sales-data> Train and Test data are stored in .csv format.

### 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.

### 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. Here, the ‘Item Weight’ and “Outlet Type’ attributes contain the missing values. So they are filled in both the train set as well as the test set with supported appropriate data types.

### Data Preprocessing

In data preprocessing all the processes required before sending the data for model building are performed. Like, here the ‘Item Visibility’ attributes are having some values equal to 0, which is not appropriate because if an item is present in the market, then how its visibility

can be 0. So, it has been replaced with the average value of the item visibility of the

respective ‘Item Identifier’ category. New attributes were added named ‘’Outlet years”, where the given establishment year is subtracted from the current year. A new “Item Type” attribute was added which just takes the first two characters of the Item Identifier which indicates the types of the items. Then mapping of “Fat content” is done based on ‘Low’,

‘Reg’ and ‘Non-edible’.

### 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. Dummy Columns are created for attributes which cannot be ranked.

### Parameter Tuning

Parameters are tuned using Grid search CV . Algorithms are used in this problem, Linear Regression, Random Forest regressor. The parameters of all these algorithms are tuned and passed into the model.

### Model Building

After doing all kinds of preprocessing operations mentioned above and performing scaling and hyper parameter tuning, the data set is passed into models, Linear Regression, Random Forest .

### Model Saving

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

### Flask Setup for Data Extraction

After saving the model, the API building process started using Streamlit. 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.

### GitHub

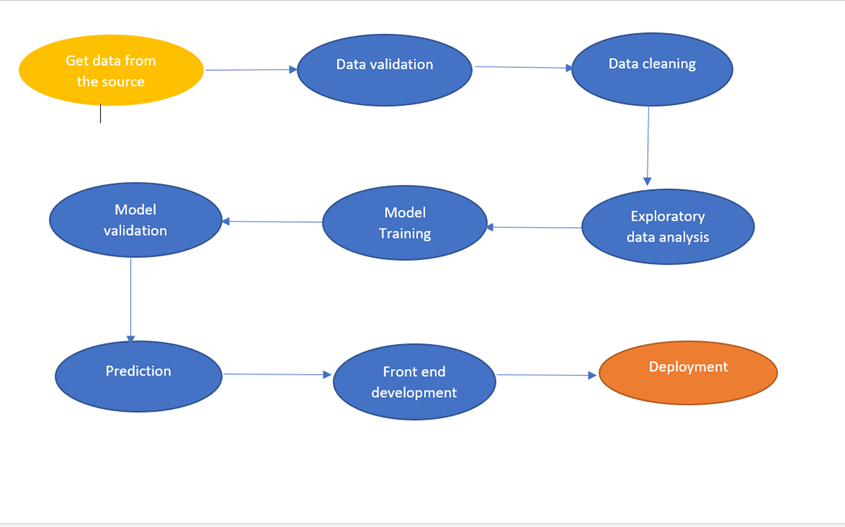
The whole project directory will be pushed into the GitHub repository.

### Deployment

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

App link- <https://store-sales-04.herokuapp.com/>

# 7. Workflow.

****