Architecture Design

Store Sales Prediction

Revision Number: 1.0

Last date of revision: 29/05/2023

Surya Prakash S

# Document Version Control

|  |  |  |  |
| --- | --- | --- | --- |
| Date Issued | Version | Description | Author |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**Contents**

[Document Version Control 2](#_Toc136279588)

[1 Introduction 4](#_Toc136279589)

[1.1 What is Architecture Design? 4](#_Toc136279590)

[1.2 Scope 4](#_Toc136279591)

[1.3 Constraints 4](#_Toc136279592)

[2 Technical specifications 4](#_Toc136279593)

[2.1 Dataset 4](#_Toc136279594)

[2.2 Logging 6](#_Toc136279595)

[2.3 Database 6](#_Toc136279596)

2.4 Deployment 7

[3 Technology stack 7](#_Toc136279597)

[4 Proposed Solution 7](#_Toc136279598)

[5 Architecture detail 8](#_Toc136279599)

[5.1. Data Gathering 8](#_Toc136279600)

[5.2. Importing CSV 8](#_Toc136279601)

[5.3. Data Preprocessing 8](#_Toc136279602)

[5.4. Feature Engineering 8](#_Toc136279603)

[5.5. Parameter Tuning 8](#_Toc136279604)

[5.6. Model Building 9](#_Toc136279605)

[5.7. Model Saving 9](#_Toc136279606)

[5.8. Django Setup for Data Extraction 9](#_Toc136279607)

[5.9. Push to Github 9](#_Toc136279608)

[5.10. Deployment 9](#_Toc136279609)

[6 User Input / Output Workflow: 9](#_Toc136279610)

**Abstract**

Machine Learning refers to a set of algorithms that enable software applications to enhance their predictive accuracy without explicit programming. The fundamental idea behind machine learning involves constructing models and employing algorithms capable of receiving input data. Through statistical analysis, these models predict an output and continuously update their predictions as new data becomes available. These versatile models can be applied to various domains and trained to align with management's expectations. By accurately predicting outcomes, organizations can make informed decisions to achieve their goals. This paper focuses on the case of Big Mart, a comprehensive shopping center, where the objective is to predict sales for different types of items and comprehend the impact of various factors on these sales. By leveraging a dataset specific to Big Mart and following a well-defined methodology for building a predictive model, highly accurate results are obtained. These observations can be utilized to enhance decision-making processes and drive sales improvement.

# 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

We only predict the expected sales of the specific product or item based on the user’s input.

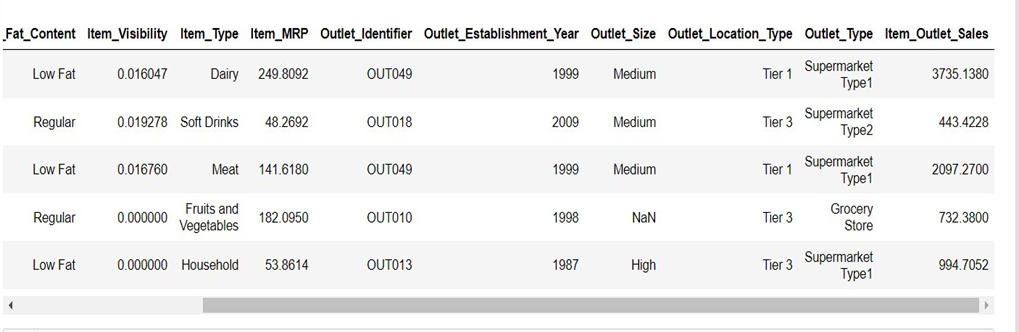
# Technical specifications

## Dataset

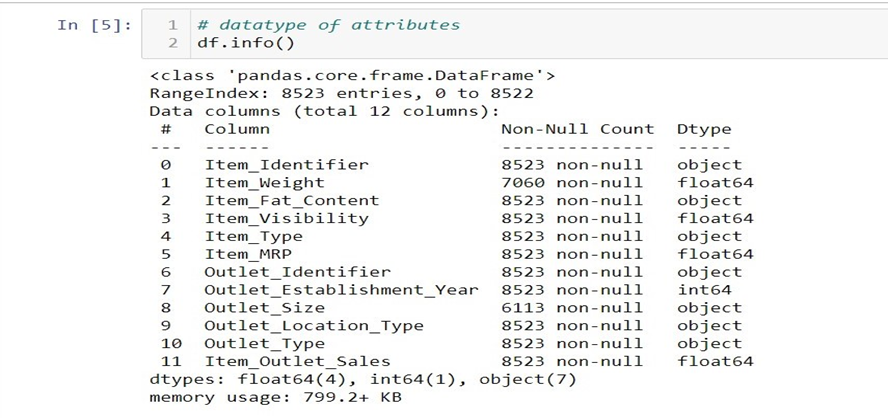
Big Mart’s data scientists collected sales data of their 10 stores situated at different locations with each store having 1559 different products as per data collection. Using all the observations it is inferred what role certain properties of an item play and how they affect their sales. The dataset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

A screenshot of a computer

Description automatically generated with medium confidence



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



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.

Various factors important by statistical means like mean, standard deviation, median, count of values and maximum value, etc. are shown below for numerical attributes.

A screenshot of a computer

Description automatically generated with medium confidence

Preprocessing of this dataset includes doing analysis on the independent variables like checking for null values in each column and then replacing or filling them with supported appropriate data types 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 and model values for categorical 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 what step logging required
* The System should be able to log each and every system flow.
* Developers can choose logging methods. You can choose database logging/ File logging as well.
* System should not be hung even after using so many loggings. Logging just because we can easily debug issues 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**

1. AWS



# Technology stack

|  |  |
| --- | --- |
| **Front End** | HTML/CSS/JS/React |
| **Backend** | Python Django |
| **Deployment** | AWS |

# Proposed Solution

We will analyze the data to explore the meaningful connections between various attributes, aiming to understand the relationship between them. To forecast future sales demand, we will employ a machine-learning algorithm. The client will provide the necessary features as input, and the web application will deliver the results. The system will receive the features, which will be checked and prepared in the backend before being passed to a machine learning model with optimized parameters. This model will then predict the final outcome.

# Architecture detail

## A picture containing diagram, plan, rectangle, screenshot Description automatically generatedData Gathering

Data source: <https://www.kaggle.com/brijbhushannanda1979/bigmart-sales-data>

Train and Test data which are stored in .csv format.

## Importing CSV

Once we gather data from the data source, we will import the csv files with the help of pandas.

## 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. Mapping of “Item\_Type” is done based on ‘Drinks’,’Food’,’Non-Consumable’. And mapping of “Fat content” is done based on ‘Low Fat’, ‘Regular’ and ‘Non-edible’.

## Feature Engineering

After preprocessing, it was discovered that certain attributes do not significantly impact the sales of the specific outlet. Therefore, those attributes were eliminated. Additionally, one-hot encoding was utilized to transform the categorical features into numerical features.

## Parameter Tuning

Parameters are tuned using RandomizedSearchCV. The algorithm used in this problem is Random Forest Regressor. The parameters of the algorithm are tunned and passed into the model.

## Model Building

After completing various preprocessing tasks mentioned earlier and conducting scaling and hyperparameter tuning, the dataset will be fed into the Random Forest regressor. It was observed that the Random Forest regressor exhibited the best performance with an RMSE value of 1120.40. Therefore, the 'Random Forest regressor' proved to be effective in solving this problem.

## Model Saving

Model is saved using pickle library in ‘rf.pkl’ format.

## Django Setup for Data Extraction

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

## Push to Github

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

## Deployment

We will be deploying the model to AWS.

## User Input / Output Workflow:

A picture containing diagram, text, plan, screenshot

Description automatically generated