Architecture

Store Sales Prediction

Written By:

Mohamed Afaque Mulla

Version: 1.0

Date: 16-02-2023

Architecture

Index

Abstract

Abstract

The goal of the project is to develop a system that can forecast future consumer demand for specific products in retail settings. by looking into a product's previous sales data. The sales of products at large shopping centers are tracked in order to predict future demand. Manufacturing and product warehouses are useful for storing a lot of things. The major objective of this study is to examine historical data, identify significant relationships among various characteristics, and develop a system that can make predictions about how much a certain product will be in demand. This technology will assist in controlling the storage capacity of warehouses.

1. Introduction

1.1 What is Architecture Design

An Architecture Design (AD), aims to provide the internal design of the actual computer code for the "Store Sales Production", In regard to the techniques and connections between classes and programme specifications, AD describes the class diagrams. In order for the programmer to create the programme directly from the document, it describes the modules.

1.2 Scope

Architecture Design (AD) is a component-level design method that incorporates a sequential process of refinement. Data structures, necessary software, architecture, source code, and finally performance algorithms can all be designed using this method. Overall, during requirement analysis, the data organisation may be created, and then refined, during data design work. and the entire process.

2. Technical Specification

2.1 Dataset

The data scientists at Big Mart gathered sales information from 10 stores spread across various locales, each of which sold 1559 distinct products. It is deduced what function specific attributes of an item play and how they impact sales using all the observations.

2.2 Logging

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

- The system determines the step at which logging is necessary.
- The system should be able to log each and every system flow.
- Developers can choose logging methods. Also, can choose database logging.
- Even after using that much logging, the system shouldn't hang. Logging is required since it makes it simple to troubleshoot issues.

2.3 Database

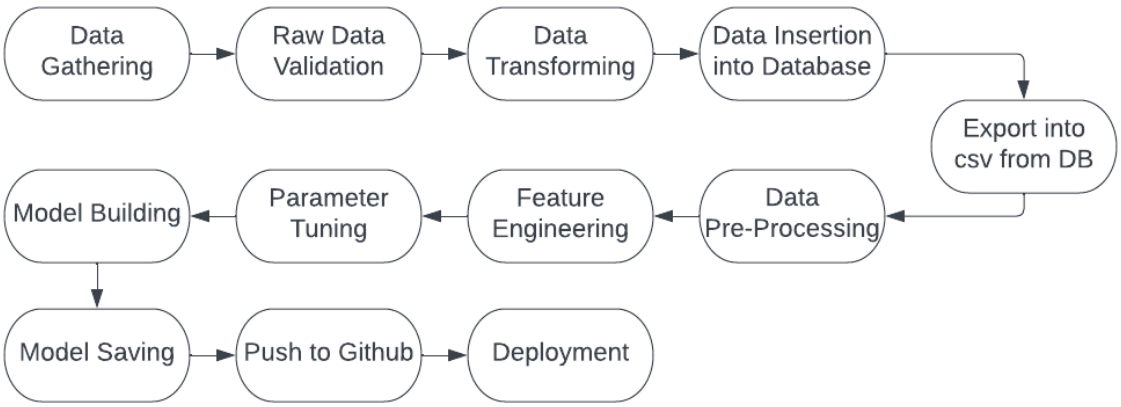
Every request must be entered into the system's database, and it must be stored in a fashion that makes it simple to keep track of and review the data.

Every piece of information a user provided, as well as any predictions made using that information, should be recorded by the system.

3. Proposed Solution

To identify the significant relationships between various parameters, we will do EDA. To forecast future sales demand, we will utilise a machine-learning system. The client will input the necessary feature and receive results via the online application. The system will collect features, which will then be sent to the backend for validation and Pre-processing before being fed to a machine learning model with hyperparameter tuning to forecast the result.

4. Architecture



2.2 Data Gathering

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

2.3 Raw Data Validation

Before moving on with any operation, multiple sorts of validation must be performed on the loaded data. Validations include ensuring that all of the columns have a standard deviation of zero and ensuring that no columns have any complete missing data. They are necessary because the qualities that include them are useless. It won't have any impact on how much a product sells at the relevant stores.

For example, if an attribute has zero standard deviation, all of its values are the same and the attribute's mean is zero. This suggests that regardless of whether sales are up or down, the attribute will remain the same. According to this, it serves no purpose to take any property into account when

operating if all of its values are missing. Increasing the likelihood of the dimensionality curse is needless.

2.4 Data Transformation

Data transformation is necessary before transferring the data to the database so that it can be transformed into a format that makes database insertion simple. The two properties in this case, "Item Weight" and "Outlet Type," have the missing data. So, they are filled out with supported relevant data types in both the train set and the test set.

2.6 Data Pre-processing

All of the steps necessary before transmitting the data for model development are completed in data pre-processing. For instance, some of the "Item Visibility" characteristics in this instance have values of 0, which is inappropriate given that if an item is available on the market, how can its visibility be 0? In its place, the average value of item visibility for the relevant "Item Identifier" category has been used. A new characteristic called "Outlet years" was added, which subtracts the current year from the supplied establishment year.

2.7 Feature Engineering

It was discovered after Pre-processing that certain of the attributes are not crucial to the item sales for the specific retailer. Hence, their qualities are dropped. To turn the categorical data into numerical features, even one hot encoding is carried out.

2.8 Parameter Tuning

Randomized search CV is used to fine-tune the parameters. In order to solve this issue, Linear Regression and Random Forest are utilised. These two algorithms' parameters are adjusted and sent to the model.

2.9 Model Building

After executing all kinds of Pre-processing operations indicated above and doing scaling and hyperparameter tweaking, the data set is passed into the models, Lasso Regression and Random Forest Regressor and Gradient Boosting Regressor. It was determined that Gradient Boosting Regressor performs best with the maximum Accuracy value 65%. As a result, the " Gradient Boosting Regressor " did well in this problem.

2.10 Model Saving

Model is saved in ".sav" format using the pickle library. So as to use the model in further applications.