**Architecture Design**

**STORES SALES PREDICTION**

|  |  |
| --- | --- |
| **Written By** | Rakesh Ranjan |
| **Document Version** | LLD - V.1.1 |
| **Last Revised Date** | 15-Apr-2023 |

**Document Version Control**

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Date** | **Author** | **Comments** |
| **LLD - V.1.0** | 10-Apr-2023 | Rakesh Ranjan | Architecture data creation |
| **LLD - V.1.1** | 15-Apr-2023 | Rakesh Ranjan | Document updated |

# **Contents**

Abstract ……………………………………………………………………………………………………………………………………. 4

1. Introduction.................................................................................................................... 5
   1. What is an Architecture design document
   2. Scope
   3. Constraints
2. Technical Specification................................................................................................... 5

2.1 Dataset

2.2 Logging

2.3 Deployment

3. Technology Stack.............................................................................................................. 7

4. Proposed Solution.............................................................................................................. 7

5. Architecture Description..................................................................................................... 7

* 1. Data Gathering
  2. Raw Data Validation
  3. Exploratory Data Analysis
  4. Feature Engineering
  5. Model Building
  6. Model Saving
  7. Flask Setup for Web Application
  8. Github
  9. Deployment

1. User Input / Output workflow..................................................................................... 8

**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 project, we will estimate the amount of insurance premium on the basis of personal health information. Taking various aspects of a dataset collected from people, and the methodology followed for building a predictive model.

1. **Introduction**

1.1. What is an Architecture Design?

The goal of Architecture Design (AD) is to give the internal design of the actual program code for the `Insurance Premium Prediction`. AD describes the class diagrams with the methods and relation between classes and program specifications. 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 to design data structures, required software, architecture, source code, and performance algorithms. Overall, the data organization may be defined during requirement analysis and then refined during data design work. And the complete workflow.

1.3 Constraints

We only predict the expected estimated cost of expenses to customers based on some personal health information.

2. **Technical Specification**

2.1 Dataset

[Big Mart Sales Data](https://www.kaggle.com/brijbhushannanda1979/bigmart-sales-data) is the biggest publicly available recipe dataset. We have the train (8523) and test (5681) datasets, and the train dataset has both input and output variable(s). We need to predict the sales for the test data set.

The variable name, variable type, measurement unit, and a brief description are given. 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 the 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. |

Pre-processing 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 the median, play an important factor in deciding which value to prioritize 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.

• The system should not be hung even after using so much logging. Logging is just because we can easily debug issuing so logging is mandatory to do.

Icon

Description automatically generated2.3 Deployment

For the hosting of the project, we use Heroku

**3**. **Technology Stack**

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

1. **Proposed Solution**

We will use performed EDA to find the important relation between different attributes and will use a machine-learning algorithm to estimate the cost of expenses. The client will fill in the required feature as input and get results through the web application. The system will get features and it will be passed into the backend where the features will be validated and pre-processed and then it will be passed to a hyperparameter-tuned machine learning model to predict the final outcome.

1. **Architecture**

5.1 Data Gathering

Data source: [**https://www.kaggle.com/brijbhushannanda1979/bigmart-sales-data**](https://www.kaggle.com/brijbhushannanda1979/bigmart-sales-data)

Train and Test data are stored in .csv format.

5.2 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 a role in contributing to the estimated cost of the premium.

5.3 Exploratory Data Analysis

Visualized the relationship between the dependent and independent features. Also checked the relationship between independent features to get more insights about the data.

5.4 Feature Engineering

After pre-processing standard scalar is performed to scale down all the numeric features. Even one hot encoding is also performed to convert the categorical features into numerical features. For this process, the pipeline is created to scale numerical features and encode the categorical features.

5.5 Model Building

After doing all kinds of pre-processing operations mentioned above and performing scaling and encoding, the data set is passed through a pipeline to all the models, Linear Regression, and Random Forest using ML. It was found that Random Forest performs best with the smallest RMSE value i.e., 4652.33, and the highest R2 score equals 0.8527 on test data So ‘Gradient boosting’ performed well in this problem.

5.6 Model Saving

The model is saved using the pickle library in ‘pickle’ format.

5.7 Flask Setup for Web Application

After saving the model, the API building process started using Flask. Web application creation was created in Flask for testing purposes. Whatever user will enter the data and then that data will be extracted by the model to estimate the premium of insurance, this is performed in this stage

5.8 GitHub

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

5.9 Deployment

The project was deployed from GitHub into the Heroku platform

1. **User Input / Output Workflow**:

Diagram

Description automatically generated