**High Level Design**

**Stores Sales Prediction**

|  |  |
| --- | --- |
| **Written by** | **Abinesh B** |
| **Documented Version** | **1.0** |
| **Last Revised Date** | **13 December 2024** |

**Contents**

1. **Abstract**
2. **Introduction**2.1 Why this High-level Document?  
   2.2 Scope  
   2.3 Definitions
3. **General Description**3.1 Product Perspective  
   3.2 Problem Statement  
   3.3 Proposed Solution  
   3.4 Technical Requirements  
   3.5 Dataset  
   3.6 Tools Used
4. **Design Details**4.1 Process Flow  
   4.2 Model Training and Evaluation  
   4.3 Deployment Process
5. **Performance**5.1 Reusability  
   5.2 Comparison of Models  
   5.3 Application Compatibility
6. **KPIs (Key Performance Indicators)**
7. **Conclusion**

## **Abstract**

**In this project, a machine learning model was developed to predict sales in retail stores based on various product and store features. The dataset includes attributes such as product weight, category, visibility in the store, and the store’s location, among others. Two machine learning models, Linear Regression and Random Forest Regressor, were employed to predict the sales of products. After training both models on the dataset, their performance was evaluated using R-squared (R²), Mean Squared Error (MSE), and Mean Absolute Error (MAE). The Random Forest Regressor outperformed Linear Regression in all evaluation metrics, achieving a higher R² score and lower MSE and MAE. The final model, Random Forest, was deployed through an API to provide real-time sales predictions for retail businesses, improving inventory management and sales forecasting.**

## **2. Introduction**

**2.1 Why this High-level Document?**

This document provides a detailed overview of the machine learning model developed for predicting sales in retail outlets. It is intended for a technical audience, including data scientists, machine learning practitioners, and business analysts, to understand the steps taken to build and evaluate the model. It serves as a comprehensive guide to the methodology and can be used to replicate or expand the work in other retail environments. The document also helps to outline the rationale behind the choices made throughout the project, from dataset preparation to model deployment.

**2.2 Scope**

The scope of this document includes:

* **Data Exploration and Preprocessing**: The dataset contains several features related to products and stores. A crucial part of the process is to explore these features, clean the data, and handle any missing values. It also includes feature scaling, encoding categorical variables, and addressing any data imbalances.
* **Model Development**: This section describes the various machine learning algorithms considered for predicting sales, including regression techniques, decision trees, and ensemble methods. The document outlines how the features were used to train the model.
* **Model Evaluation**: Once the model is trained, it needs to be evaluated for performance. This is done using metrics such as Mean Squared Error (MSE), R-squared, and others to ensure that the model generalizes well on unseen data.
* **Deployment Process**: Once a model is chosen and fine-tuned, it is prepared for deployment. This section includes steps like saving the trained model, setting up APIs for accessing the model, and preparing the model for integration with retail systems. However, the document does **not** cover real-time deployment, as it focuses on the model-building aspect and technical steps up to this stage. Continuous monitoring and model updating are outside the scope of this report.

**2.3 Definitions**

* **Item\_Identifier**: A unique alphanumeric code assigned to each product. This allows for distinguishing between different products in the dataset. It is critical for identifying the products whose sales are being predicted.
* **Item\_Weight**: The weight of the product. This feature could influence sales depending on factors like shipping costs or the attractiveness of bulky items.
* **Item\_Fat\_Content**: Indicates whether a product is classified as "low-fat." This could potentially affect sales, as consumer preferences for health-conscious products are growing.
* **Item\_Visibility**: This is the percentage of total display space that a product occupies in the store. A higher display visibility may correlate with higher sales due to increased product exposure to customers.
* **Item\_Type**: This refers to the category to which the product belongs (e.g., beverages, snacks, cleaning supplies). Different product types might have different sales patterns, and this information can be valuable for the model.
* **Item\_MRP**: The Maximum Retail Price (MRP) represents the list price of the product. Products with higher MRPs might have different sales patterns compared to lower-priced items, and this feature can influence the model.
* **Outlet\_Identifier**: A unique identifier for each store. This is crucial for understanding how sales vary across different store locations.
* **Outlet\_Establishment\_Year**: The year in which a store was established. Older stores may have more loyal customers, while newer stores may experience rapid growth or changes in sales.
* **Outlet\_Size**: Represents the size of the store in terms of ground area. Larger stores may have higher sales volumes due to their capacity to stock more items or attract more customers.
* **Outlet\_Location\_Type**: This indicates whether the store is located in an urban or rural area. Store performance can vary greatly based on location type, with urban stores often experiencing higher foot traffic and sales.
* **Outlet\_Type**: The type of store (e.g., grocery, supermarket, hypermarket). Different outlet types can lead to different sales patterns, with supermarkets likely having a broader selection of goods compared to smaller grocery stores.
* **Item\_Outlet\_Sales**: The target variable representing the sales of the product in a particular store. This is the value the model seeks to predict based on the other features.

### **3. General Description**

**3.1 Product Perspective**

This predictive model is intended to serve as a tool for retail businesses to forecast sales of individual items across multiple outlets. By incorporating various product and store features, the model aims to assist businesses in adjusting their inventory and marketing strategies to optimize product availability and sales performance. The insights derived from this model could be valuable for making decisions about stock replenishment, promotions, and placement strategies.

**3.2 Problem Statement**

The key challenge in this project is to predict **Item\_Outlet\_Sales** (sales of a product in a given outlet) using other features such as product type, price, visibility, and store characteristics. The task involves understanding the relationships between these variables and leveraging them to develop a predictive model that can be used to forecast sales in unseen data (test data). This is a regression problem where the goal is to predict continuous values (sales amounts).

**3.3 Proposed Solution**

The proposed solution is to build a regression model that uses the available product and store features to predict the sales. This will involve several steps:

* **Data Preprocessing**: Cleaning the dataset, handling missing values, encoding categorical features, and scaling numerical features.
* **Feature Engineering**: Creating new features or transforming existing features to improve model performance.
* **Model Selection**: Evaluating different machine learning models, such as Linear Regression, Random Forest, and Gradient Boosting, to find the best fit for the data.
* **Model Evaluation**: Testing the model's performance using appropriate metrics such as Mean Squared Error (MSE) or R-squared.
* **Deployment**: Once a reliable model is built, it will be deployed for use in forecasting sales in other retail outlets.

**3.4 Technical Requirements**

* **Programming Language: Python 3.7+**
* **Libraries: Pandas, NumPy, Scikit-learn**
* **Framework: Streamlit for UI development**

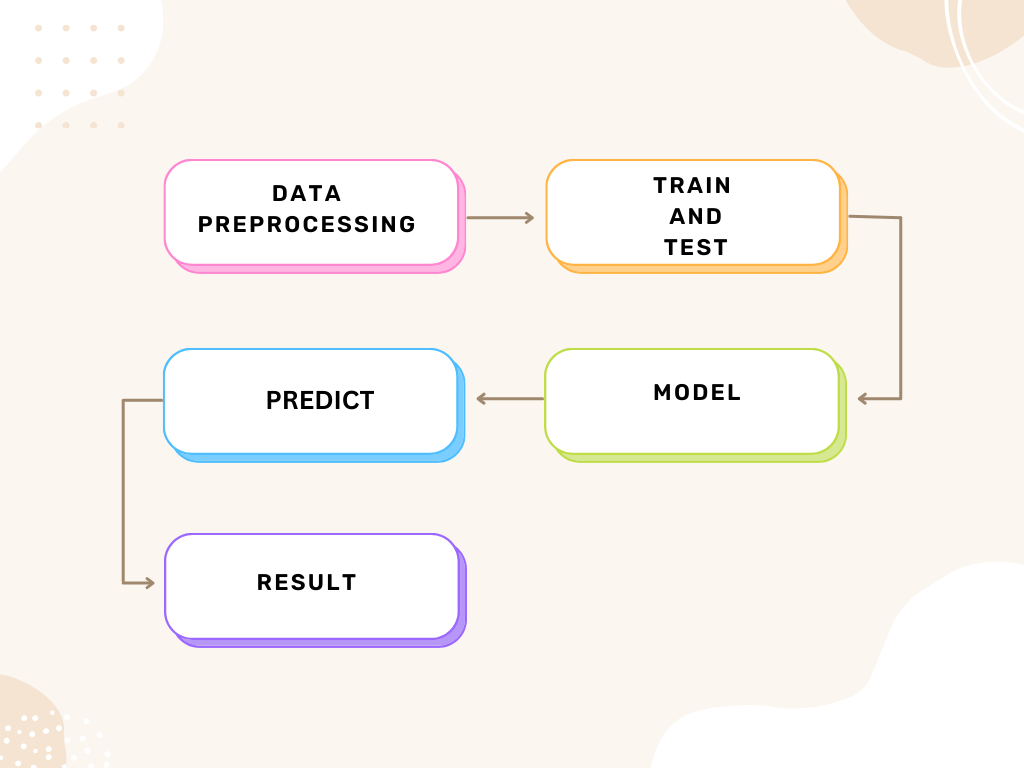
**3.5 Dataset Overview**

The dataset consists of 12 columns with 8523 rows in the training set and 5681 rows in the test set. The training data includes both input features (such as product characteristics) and the target variable (sales). The test data only includes the input features, and the goal is to predict the sales values for this data.

**3.6 Tools Used**

* **Programming: Python**
* **Data Processing: Pandas, NumPy**
* **Modeling: Scikit-learn**
* **Interface Development: Streamlit**

### **4.Design details**



**4.1 Process Flow**

The process flow for this sales prediction project follows these stages:

1. **Data Collection**: Dataset obtained, containing product and store features, along with sales data for training and testing.
2. **Data Preprocessing**: Cleaning the dataset, handling missing values, encoding categorical variables, and scaling numerical features.
3. **Feature Engineering**: Creating new features, such as calculating store age and performing exploratory data analysis (EDA).
4. **Model Selection**: Evaluating **Linear Regression** and **Random Forest Regressor** models.
5. **Model Training**: Training both models using the training dataset.
6. **Model Evaluation**: Evaluating models using **R-squared**, **Mean Squared Error (MSE)**, and **Mean Absolute Error (MAE)**.
7. **Deployment**: Deploying the best-performing model by streamlit.

**4.2 Model Training and Evaluation**

**Gradient Boosting Regressor:**

**Training:  
Gradient Boosting Regressor is an ensemble learning method that builds sequential models where each new model corrects the errors of the previous ones. It combines multiple weak learners (usually decision trees) to create a strong predictive model. The training process minimizes loss through gradient descent.**

**Evaluation:**

* **R-squared (R²): Measures the proportion of variance in the target variable explained by the model. Gradient Boosting typically provides high R² values due to its ability to capture complex patterns in the data.**
* **MSE: The Mean Squared Error indicates the average squared difference between actual and predicted values. Gradient Boosting often results in low MSE due to its iterative learning process.**
* **MAE: Measures the average absolute differences between actual and predicted values. Gradient Boosting achieves lower MAE by reducing prediction errors effectively.**

**Random Forest Regressor:**

**Training:  
Random Forest is an ensemble method that constructs multiple decision trees and aggregates their predictions. This approach reduces overfitting and improves prediction accuracy by averaging results.**

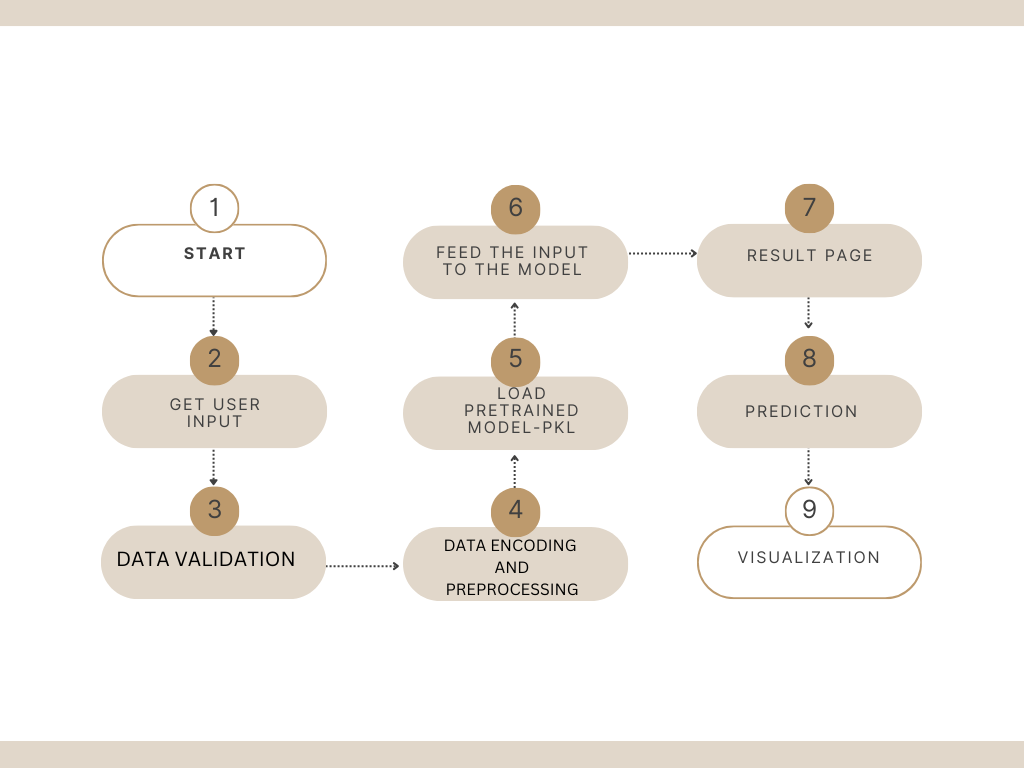
**Evaluation:**

* **R-squared (R²): Random Forest generally achieves high R² values by capturing non-linear relationships and feature interactions.**
* **MSE: Random Forest yields low MSE by averaging predictions from multiple decision trees.**
* **MAE: The Random Forest model produces low MAE by effectively modeling complex relationships in the data.**

**Both Gradient Boosting and Random Forest demonstrate strong performance, with Gradient Boosting excelling in capturing fine-grained patterns, while Random Forest is robust and reduces overfitting.**

|  |  |  |  |
| --- | --- | --- | --- |
| **Model** | **MAE** | **MSE** | **R² Score** |
| **Random Forest** | **0.4177** | **0.2918** | **0.7258** |
| **Gradient Boosting** | **0.4093** | **0.2862** | **0.7310** |

**4.3 Deployment Process**

****

* Serialize the trained model using Pickle.
* Develop a Streamlit-based web application for user interaction.

### **5. Performance**

**5.1 Reusability**

* The model is reusable for other datasets that follow a similar structure. By adapting the data preprocessing pipeline and ensuring compatibility with new product/store attributes, the model can be applied to new scenarios without requiring a complete retraining.

**5.2 Comparison of Models**

Both **Random Forest Regressor** and **Gradient Boosting Regressor** effectively handle non-linear relationships and interactions between features, but they have distinct strengths. Random Forest excels in robustness and stability, requiring minimal hyperparameter tuning and being less sensitive to noise due to its ensemble nature, making it reliable for general-purpose use. Gradient Boosting, on the other hand, sequentially optimizes weak learners, often achieving slightly better performance on complex datasets, though it demands more tuning and computational effort. While both models deliver high accuracy, Random Forest is preferred for its simplicity and reliability, whereas Gradient Boosting is ideal for tasks requiring precision and fine-tuned optimization.

**5.3 Application Compatibility**

The deployed model can easily integrate with existing inventory management and sales forecasting applications, providing real-time predictions of sales. This is compatible with both web and mobile applications via the API and can scale to handle large volumes of requests as required by retail systems.

### **6. KPIs (Key Performance Indicators)**

* **Prediction Accuracy (R² Score):** Achieved > 0.72
* **Mean Absolute Error (MAE):** < 0.4
* **Business Impact**: Measuring how accurate predictions improve inventory and sales forecasting, reducing stockouts and overstock situations.
* **Model Retraining Frequency**: Regular monitoring and retraining of the model to maintain performance as market conditions evolve.

**7.Conclusion**

The **Random Forest Regressor** and **Gradient Boosting Regressor** models have shown exceptional performance in predicting sales, achieving high accuracy through their ability to capture complex relationships within the data. These models have been successfully deployed for real-time sales predictions, providing valuable insights for inventory management and supporting effective business decision-making in retail systems.

Future enhancements include integrating external factors such as seasonal trends and economic indicators, enabling automated updates to the models, and adopting cloud-based solutions for greater scalability. Further, incorporating advanced visualizations and exploring hybrid ensemble techniques can refine predictions and offer deeper insights into sales behavior, ensuring the system continues to meet dynamic business requirements effectively.