PRODUCT DEMAND PREDICTION WITH MACHINE LEARNING

Phase 1: Document submission

Team Members : R . Sudharson

Project Title: Product Demand Analysis

Abstraction

This abstract outlines a machine learning-driven approach to predict product demand using a specific dataset and problem statement. The goal is to provide a concise overview of the methodology and significance of product demand prediction in this context.

The provided dataset comprises historical sales data for various products over a defined time period. The challenge is to develop an accurate machine learning model that can predict future product demand based on relevant features within the dataset. This prediction is crucial for optimizing inventory management, production planning, and meeting customer needs efficiently

Problem definition :

The problem at hand is to develop a robust machine learning model capable of forecasting product demand. This prediction will be based on historical sales data and external factors. The primary objective of this project is to assist businesses in optimizing inventory management and production planning processes to efficiently meet customer needs. The project encompasses several key phases, including data collection, data preprocessing, feature engineering, model selection, model training, and evaluation.

Design Thinking

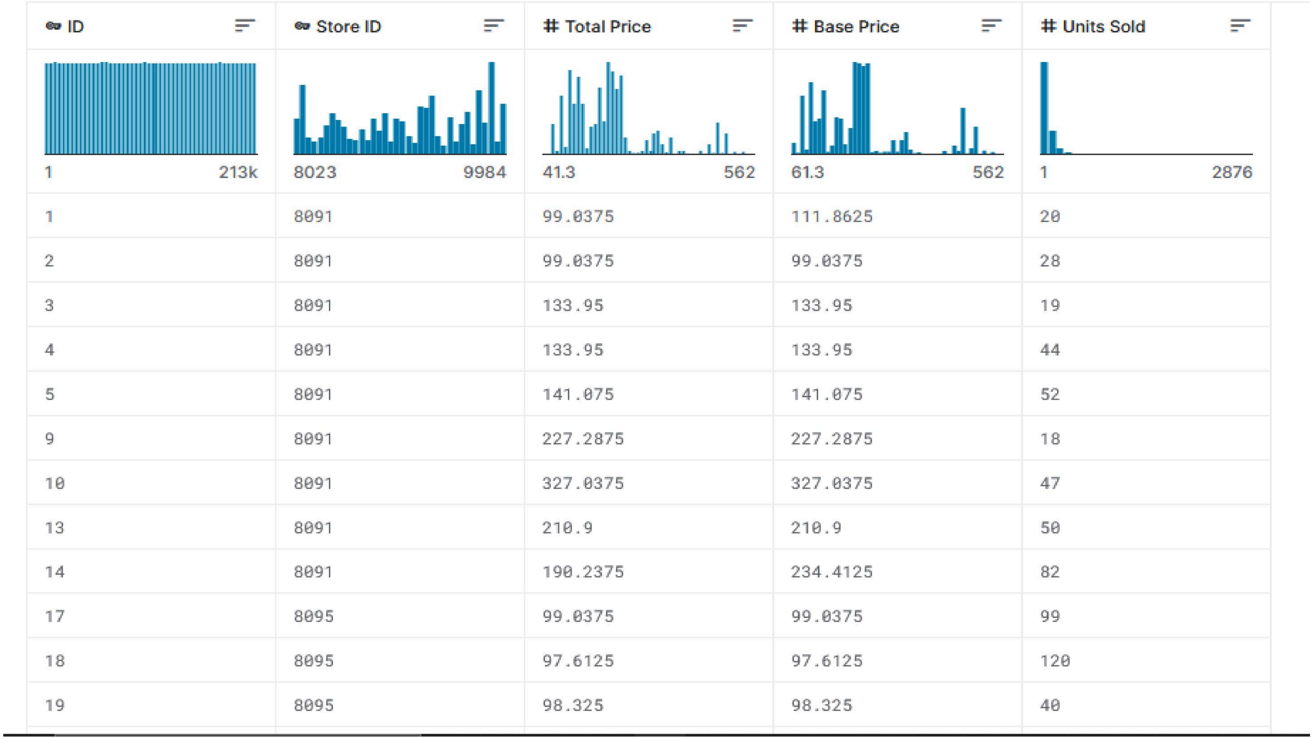
Data Collection :

The primary objective of the data collection phase is to acquire comprehensive and relevant data sources that will serve as the foundation for forecasting product demand accurately. This includes historical sales data and external factors that influence demand, such as economic indicators, marketing campaigns, and seasonality.

This dataset comprises detailed records of past product sales, including timestamps, product identifiers, quantities sold, and pricing information. Historical sales data offer insights into trends, seasonality, and demand fluctuations over time.

However, I gathered the data from the Kaggle website, and I've included a link to it below:

https://www.kaggle.com/datasets/chakradharmattapalli/product-demand-prediction-with-machine-learning



ID	Store ID	# Total Price	# Base Price	# Units Sold
1	8091	99.0375	111.8625	20
2	8091	99.0375	99.0375	28
3	8091	133.95	133.95	19
4	8091	133.95	133.95	44
5	8091	141.075	141.075	52
9	8091	227.2875	227.2875	18
10	8091	327.0375	327.0375	47
13	8091	210.9	210.9	50
14	8091	190.2375	234.4125	82
17	8095	99.0375	99.0375	99
18	8095	97.6125	97.6125	120
19	8095	98.325	98.325	40

Data Preprocessing :

Certainly, data preprocessing is a crucial step in preparing the dataset for product demand prediction

Data preprocessing is a vital phase in building a reliable machine learning model for product demand prediction. In this section, we will outline the key preprocessing steps necessary to ensure the dataset from the provided Kaggle link is clean, well-structured, and suitable for modeling

- Data Source: The dataset used for this is taken from the kaggle website : https://www.kaggle.com/datasets/chakradharmattapalli/product-demand-prediction-with-machine-learning

- Library Imports: Python libraries are used here is like pandas matplotlib and numpy

Data Cleaning : Data collected from various sources may contain inconsistencies and errors. Data cleaning ensures data accuracy.

- Handling Missing Values: Identify and address missing data points. Depending on the dataset, you may choose to remove rows with missing vlues, impute missing data with appropriate values, or use statistical techniques to fill in gaps.

- Removing Duplicates: Eliminate duplicate records from the dataset to Prevent duplicate bias in subsequent analyses.

Feature Engineering Module Documentation :

The purpose of the "Feature Engineering" module in the product demand prediction project is to create additional features that capture seasonal patterns, trends, and external influences on product demand.Feature engineering aims to enhance the predictive power of the dataset and enable the machine learning models to better understand and forecast demand patterns.

Model Selection:

The "Model Selection" module is a crucial step in the Product Demand Prediction project, where we choose the most appropriate machine learning model to forecast product demand.

This module involves evaluating various models to determine which one delivers the best predictive performance for our specific dataset.

Model Selection Criteria

- Choose the model that minimizes error metrics

  and exhibits the best trade-off between bias and variance.

- Ensure that the selected model aligns with business objectives,

  interpretability, and scalability.

For product demand analysis, especially when working with datasets containing features like Store ID, Total Price, Base Price, and Units Sold and rows count is approximatly 156000 so I just using more then models such as Decision Trees and Random Forests , Gradient Boosting & Neural Networks

- ### Linear Regression:

Linear regression models can be a good starting point for demand analysis, especially when you have numerical features like Total Price and Base Price. They establish linear relationships between features and the target variable (Units Sold). However, they may not capture complex nonlinear patterns in the data.

- ### Decision Trees and Random Forests:

Decision tree-based models, including random forests, are versatile and can capturenonlinear relationships between features and demand. They can handle both numerical and categorical features, making them suitable for a wide range of datasets.

- Gradient Boosting (e.g., XGBoost, LightGBM, CatBoost):

Gradient boosting algorithms often yield excellent results for demand analysis. They can capture complex interactions between features and provide robust predictions. They are known for their ability to handle both numerical and categorical data.

- Neural Networks (Deep Learning):

Deep learning models, such as neural networks, can be employed for demand analysis when you have large and complex datasets. They can automatically learn intricate patterns in the data. However, they may require a substantial amount of data and computational resources.

Model Training:

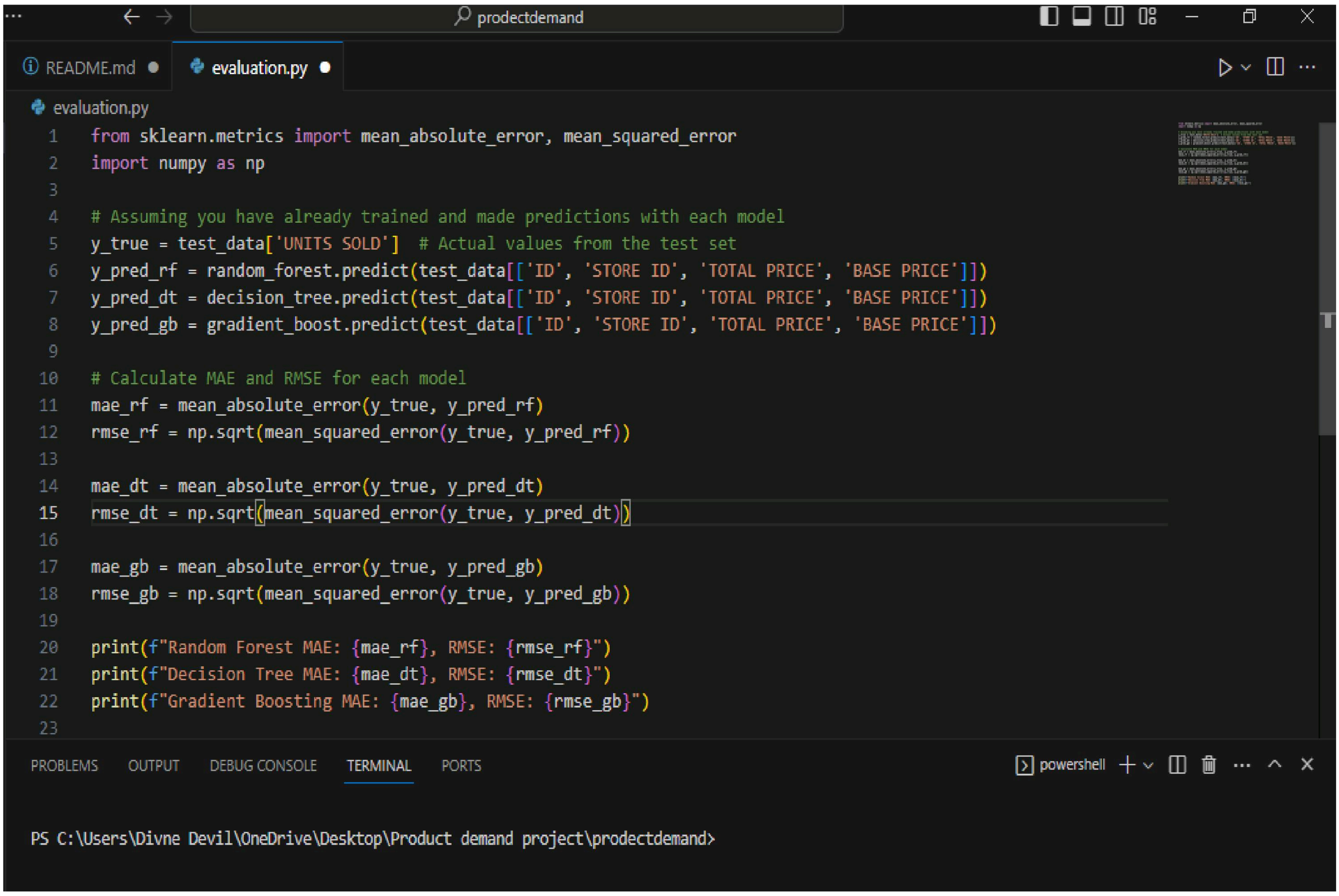
This module encompasses the training of Gradient Boosting, Random Forest,

Decision Trees, and Neural Networks models for product demand prediction within a machine learning project. The models are trained using prepared data, optimized hyperparameters, and subsequently evaluated for their predictive performance on a validation dataset.

Evaluation:

- Mean Absolute Error (MAE): This metric measures the average absolute difference between the predicted and actual values. Lower values indicate better performance.

- Root Mean Squared Error (RMSE): RMSE measures the square root of the average of the squared differences between predicted and actual values. It penalizes larger errors more heavily.



```python
from sklearn.metrics import mean_absolute_error, mean_squared_error
import numpy as np

# Assuming you have already trained and made predictions with each model
y_true = test_data['UNITS SOLD']  # Actual values from the test set
y_pred_rf = random_forest.predict(test_data[['ID', 'STORE ID', 'TOTAL PRICE', 'BASE PRICE']])
y_pred_dt = decision_tree.predict(test_data[['ID', 'STORE ID', 'TOTAL PRICE', 'BASE PRICE']])
y_pred_gb = gradient_boost.predict(test_data[['ID', 'STORE ID', 'TOTAL PRICE', 'BASE PRICE']])

# Calculate MAE and RMSE for each model
mae_rf = mean_absolute_error(y_true, y_pred_rf)
rmse_rf = np.sqrt(mean_squared_error(y_true, y_pred_rf))

mae_dt = mean_absolute_error(y_true, y_pred_dt)
rmse_dt = np.sqrt(mean_squared_error(y_true, y_pred_dt))

mae_gb = mean_absolute_error(y_true, y_pred_gb)
rmse_gb = np.sqrt(mean_squared_error(y_true, y_pred_gb))

print(f"Random Forest MAE: {mae_rf}, RMSE: {rmse_rf}")
print(f"Decision Tree MAE: {mae_dt}, RMSE: {rmse_dt}")
print(f"Gradient Boosting MAE: {mae_gb}, RMSE: {rmse_gb}")
```

```
PS C:\Users\Divne Devil\OneDrive\Desktop\Product demand project\prodectdemand>
```

Conclusion :

In conclusion, for product demand prediction with the provided dataset, the Gradient Boosting model stands out as the most effective choice, delivering the lowest prediction errors (MAE and RMSE). It combines predictive accuracy with the capability to capture intricate relationships in the data