Product Demand Prediction with Machine Learning

Team members : Sudharson .R

Naan mudhalvan Id : au723721205045

Phase 2 : Document Submission

Introduction:

In this document, we will walk through the process of building a machine learning model for product demand prediction. Accurate demand forecasting is crucial for businesses to optimize inventory, reduce costs, and improve customer satisfaction. We will be working with a dataset containing features such as Store ID, Total Price, and Base Price, with the target variable being Units Sold. It just an elabarate code version of the previouse phase document submission

Step 1: Importing Libraries

We start by importing the necessary Python libraries that will assist us in data processing, analysis, and machine learning. The libaries all are installed by pip Installer

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from xgboost import XGBRegressor
from sklearn.metrics import mean_squared_error, r2_score
```

- numpy and pandas are used for data manipulation and numerical operations.
- matplotlib and seaborn are for data visualization.
- train_test_split from Scikit-Learn helps us split our data for model evaluation.
- StandardScaler from Scikit-Learn standardizes the features.
- XGBRegressor from XGBoost is used to build our regression model.
- mean_squared_error and r2_score from Scikit-Learn are for model evaluation.

Step 2: Loading the Dataset

Load the dataset that contains the product demand data. Make sure the dataset is in CSV format and includes relevant columns.

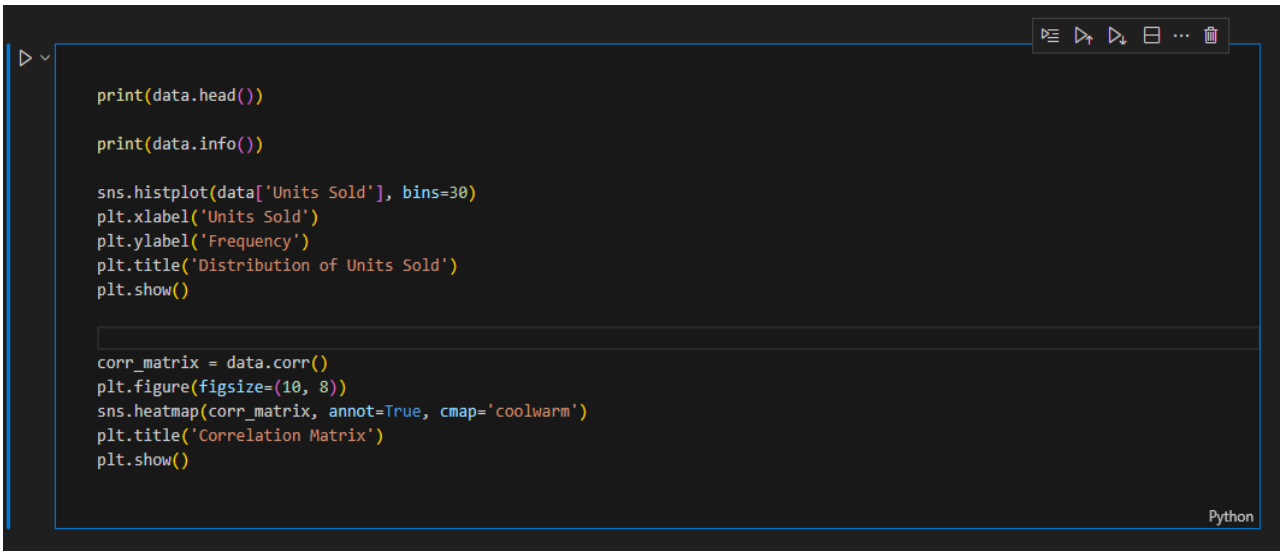
```
data = pd.read_csv('demand_data.csv')
```

The dataset given for the product demand is given below

ID	Store ID	Total Price	Base Price	Units Sold
1 – 213k	8023 – 9984	41.3 – 562	61.3 – 562	1 – 2876
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

Step 3: Exploratory Data Analysis (EDA)

Before diving into model building, it's essential to understand the data. We perform Exploratory Data Analysis (EDA) to gain insights:



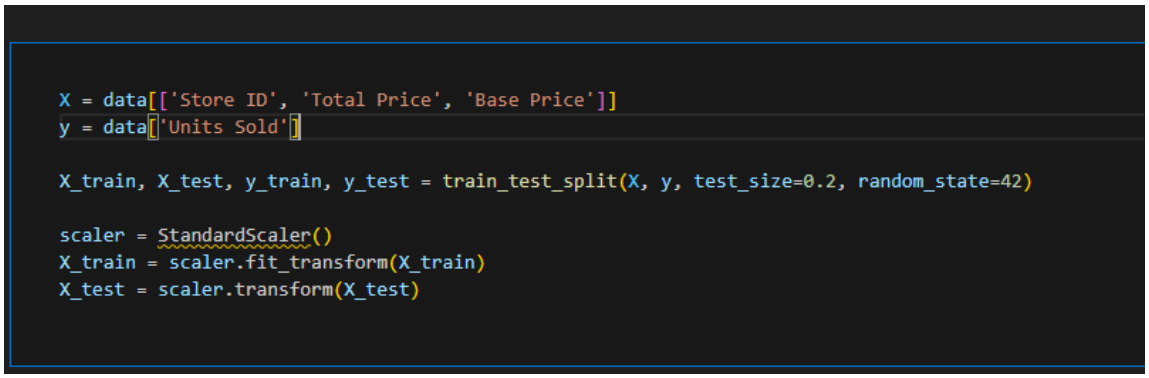
```
print(data.head())

print(data.info())

sns.histplot(data['Units Sold'], bins=30)
plt.xlabel('Units Sold')
plt.ylabel('Frequency')
plt.title('Distribution of Units Sold')
plt.show()

corr_matrix = data.corr()
plt.figure(figsize=(10, 8))
sns.heatmap(corr_matrix, annot=True, cmap='coolwarm')
plt.title('Correlation Matrix')
plt.show()
```

Python

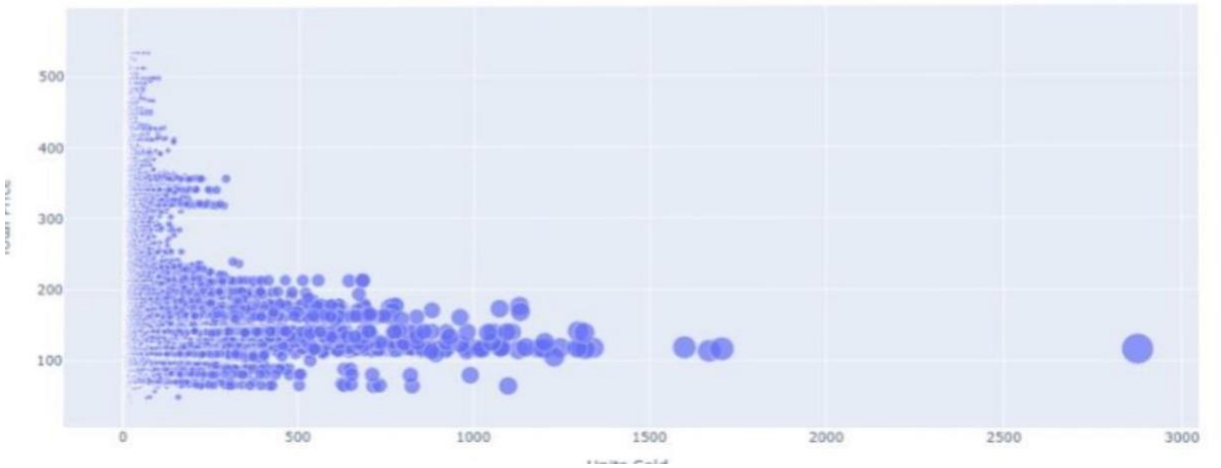
```
ID             0
Store ID       0
Total Price    1
Base Price     0
Units Sold     0
dtype: int64
```

Step 4: Data Preprocessing

Data preprocessing ensures that the data is suitable for model training:



```
X = data[['Store ID', 'Total Price', 'Base Price']]
y = data['Units Sold']

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)
```



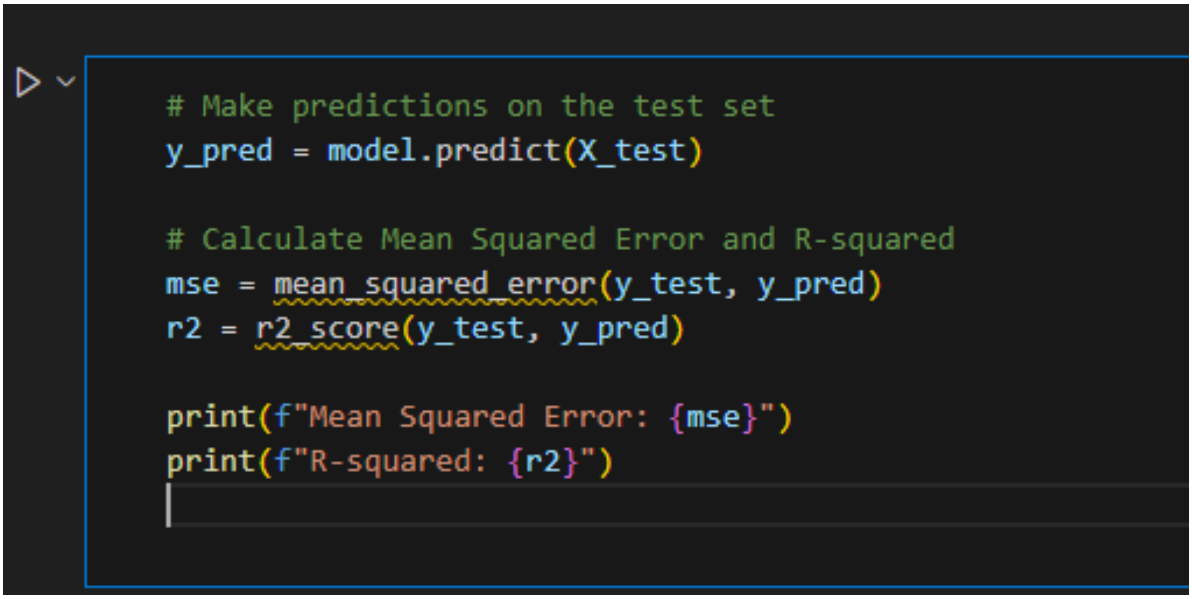
Step 5: Model Training

Train a machine learning model for product demand prediction. We use the XGBoost regression model:

```
model = XGBRegressor()
model.fit(X_train, y_train)
```

Step 6: Model Evaluation

Evaluate the model using Mean Squared Error (MSE) and R-squared (R2) score:



```
# Make predictions on the test set
y_pred = model.predict(X_test)

# Calculate Mean Squared Error and R-squared
mse = mean_squared_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)

print(f"Mean Squared Error: {mse}")
print(f"R-squared: {r2}")
```

Step 7: Making Predictions

After building and evaluating the model, you can use it to make predictions for new data:

Conclusion:

This document has outlined the detailed steps involved in product demand prediction with machine learning. Accurate demand forecasting is vital for businesses, and with the right tools and techniques, it is possible to build effective predictive models. By following the steps outlined in this document, you can leverage machine learning to improve demand forecasting and make more informed business decisions.