PRODUCT DEMAND PREDICTION BY MACHINE LEARNING

(PHASE-5)

Problem Statement:

The problem at hand is to accurately forecast product demand in a dynamic marketplace, considering various influencing factors that impact consumer behavior and purchase trends. Current demand forecasting methods are often inadequate in capturing the complexities of market dynamics, leading to suboptimal inventory management, production planning, and supply chain inefficiencies. Therefore, there is a need for a robust and adaptable predictive model that can leverage historical data and external variables to provide accurate demand predictions across different product categories and time horizons.

Design Thinking Process:

1. Empathize: Understand the pain points and challenges faced by businesses due to inaccurate demand forecasting, including the implications on inventory management, production efficiency, and customer satisfaction.

2. Define: Clearly define the goals and objectives of the product demand prediction system, taking into account the need for a scalable, adaptable, and accurate model that can accommodate various external variables and market fluctuations.

3. Ideate: Brainstorm potential approaches and methodologies that can integrate machine learning algorithms, statistical analysis, and data preprocessing techniques to develop a comprehensive demand prediction model.

4. Prototype: Develop a preliminary model that incorporates key features and data points, allowing for iterative testing and refinement based on historical data and real-time market insights.

5. Test: Evaluate the prototype model's performance and accuracy using appropriate metrics, considering different product categories and time periods to ensure its reliability and robustness in diverse business contexts.

6. Implement: Integrate the refined model into the existing business infrastructure, enabling stakeholders to access real-time demand forecasts and make data-driven decisions for inventory management, production planning, and supply chain optimization.

Phases of Development of Product Demand Prediction:

1. Data Collection and Preprocessing: Gather historical sales data, economic indicators, and relevant external factors impacting demand. Cleanse the data, handle missing values, and conduct feature engineering to extract meaningful insights.

2. Exploratory Data Analysis (EDA): Analyze the data to identify underlying trends, patterns, and correlations that can inform the development of the predictive model.

3. Model Selection and Development: Choose appropriate machine learning algorithms, including regression, time series analysis, and deep learning techniques, to develop a robust demand prediction model. Incorporate feature selection and hyperparameter optimization to enhance the model's performance.

4. Model Evaluation: Assess the model's accuracy and efficiency using metrics such as Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and Mean Absolute Percentage Error (MAPE) across various product categories and time horizons.

5. Refinement and Iteration: Fine-tune the model based on feedback and real-time data updates, ensuring its adaptability to changing market dynamics and external variables.

6. Deployment and Integration: Implement the finalized model into the business infrastructure, enabling stakeholders to access accurate demand forecasts and make informed decisions for inventory management, production planning, and supply chain optimization. Regular monitoring and updates are essential to ensure the model's continued relevance and effectiveness in addressing evolving market demands.

Dataset Description:

The dataset used for product demand prediction comprises historical sales data, economic indicators, and various external factors that influence product demand. It includes information such as product attributes, sales volumes, pricing data, promotional activities, seasonal trends, and macroeconomic indicators. The dataset is structured, containing both numerical and categorical variables, and covers a significant time span to capture various market dynamics and trends.

Data Preprocessing Steps:

1. Data Cleaning: Identify and handle missing values, outliers, and inconsistencies in the dataset to ensure data integrity and reliability for analysis.

2. Feature Engineering: Create additional relevant features such as seasonality indicators, lag variables, and moving averages to capture temporal patterns and dependencies in the data.

3. Data Transformation: Normalize or standardize numerical features to ensure that all variables are on a comparable scale. Encode categorical variables using techniques like one-hot encoding or label encoding to make them suitable for analysis.

4. Handling Imbalanced Data: Address any class imbalances in the dataset, especially if dealing with products with varying demand levels, to ensure the model's ability to capture patterns across different demand categories.

5. Train-Test Split: Divide the dataset into training and testing sets, ensuring that the temporal order is maintained to simulate real-world scenarios and assess the model's predictive performance accurately.

Analysis Techniques Applied:

1. Exploratory Data Analysis (EDA): Utilize statistical and visualization techniques to gain insights into the dataset's underlying trends, correlations, and patterns. Identify seasonal variations, trends, and potential outliers that could impact the demand prediction process.

2. Time Series Analysis: Apply time series analysis techniques such as autoregressive integrated moving average (ARIMA), seasonal decomposition, and trend analysis to understand the temporal patterns and seasonality within the demand data.

3. Regression Analysis: Employ regression analysis to identify the relationships between demand and various explanatory variables, enabling the model to capture the impact of different factors on product demand.

4. Feature Selection Techniques: Implement feature selection methods such as correlation analysis, recursive feature elimination, and feature importance ranking to identify the most influential features that significantly affect product demand.

5. Model Evaluation Metrics: Assess the predictive model's performance using evaluation metrics such as Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and Mean Absolute Percentage Error (MAPE) to measure the accuracy and reliability of the demand prediction model.

Key Findings:

1. Seasonal Trends: The demand for certain products exhibits strong seasonal variations, with peak periods corresponding to specific times of the year, such as holidays or seasonal events.

2. Price Sensitivity: Changes in pricing have a noticeable impact on product demand, with certain products demonstrating a higher price elasticity compared to others.

3. External Factors: Economic indicators, marketing promotions, and external events significantly influence product demand, highlighting the need to consider broader market dynamics when forecasting demand.

4. Lag Effects: Previous sales data has a considerable impact on future demand, indicating the existence of lag effects that should be incorporated into the demand prediction model.

Insights:

1. Inventory Optimization: Understanding seasonal trends and their influence on demand allows for more effective inventory management, ensuring that the right products are stocked in adequate quantities to meet consumer demand during peak periods.

2. Pricing Strategy Refinement: Recognizing the price sensitivity of specific products helps in developing an optimized pricing strategy that balances profitability with maintaining consumer demand, maximizing overall revenue.

3. Marketing Campaign Alignment: Integrating external factors, such as marketing campaigns and economic indicators, into the demand prediction model facilitates the alignment of marketing efforts with anticipated demand fluctuations, enhancing the effectiveness of promotional activities.

4. Production Planning Efficiency: Incorporating lag effects in demand forecasting enables more accurate production planning, ensuring that production levels align with anticipated future demand, minimizing inventory shortages or excess stock.

Recommendations:

1. Implement a dynamic pricing strategy that takes into account the price elasticity of products and responds to changes in demand accordingly.

2. Integrate external data sources, such as economic indicators and market trends, into the demand prediction model to improve the accuracy of forecasts and enhance the model's adaptability to changing market conditions.

3. Develop a comprehensive inventory management system that considers both seasonal demand fluctuations and lag effects to optimize stock levels and prevent stockouts or overstocking.

4. Continuously monitor and update the demand prediction model to incorporate the latest market insights and ensure its relevance and effectiveness in addressing evolving consumer preferences and market dynamics.

1. data_preprocessing.py:

python

Code for data preprocessing

Includes data cleaning, feature engineering, and data transformation

Insert code here

2. model_training.py:

python

Code for model training

Includes model selection, feature selection, and hyperparameter optimization

Insert code here

3. model_evaluation.py:

#python

Code for model evaluation

Includes evaluation metrics calculation and model performance assessment

Insert code here

4. README.md:

markdown

 # Product Demand Prediction using Machine Learning

This project aims to predict product demand using a machine learning approach.

How to Run the Code

1. Install the required dependencies using the following command:

pip install -r requirements.txt

2. Run the data preprocessing script:

python data_preprocessing.py

3. Execute the model training script:

python model_training.py

4. Evaluate the model using the evaluation script:

python model_evaluation.py

Dependencies

- Python 3.0 or higher
- Additional dependencies are listed in requirements.txt.

Let's start by importing the necessary Python libraries and the dataset we need for the task of product demand prediction:

```
import pandas as pd

import numpy as np

import plotly.express as px

import seaborn as sns

import matplotlib.pyplot as plt

from sklearn.model_selection import train_test_split
```

```
from sklearn.tree import DecisionTreeRegressor
```

```
data = pd.read_csv("https://raw.githubusercontent.com/amankharwal/Website-data/master/demand.csv")
```

```
data.head()
```

	ID	Store ID	Total Price	Base Price	Units Sold
0	1	8091	99.0375	111.8625	20
1	2	8091	99.0375	99.0375	28
2	3	8091	133.9500	133.9500	19
3	4	8091	133.9500	133.9500	44
4	5	8091	141.0750	141.0750	52

Now let's have a look at whether this dataset contains any null values or not:

```
data.isnull().sum()
```

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

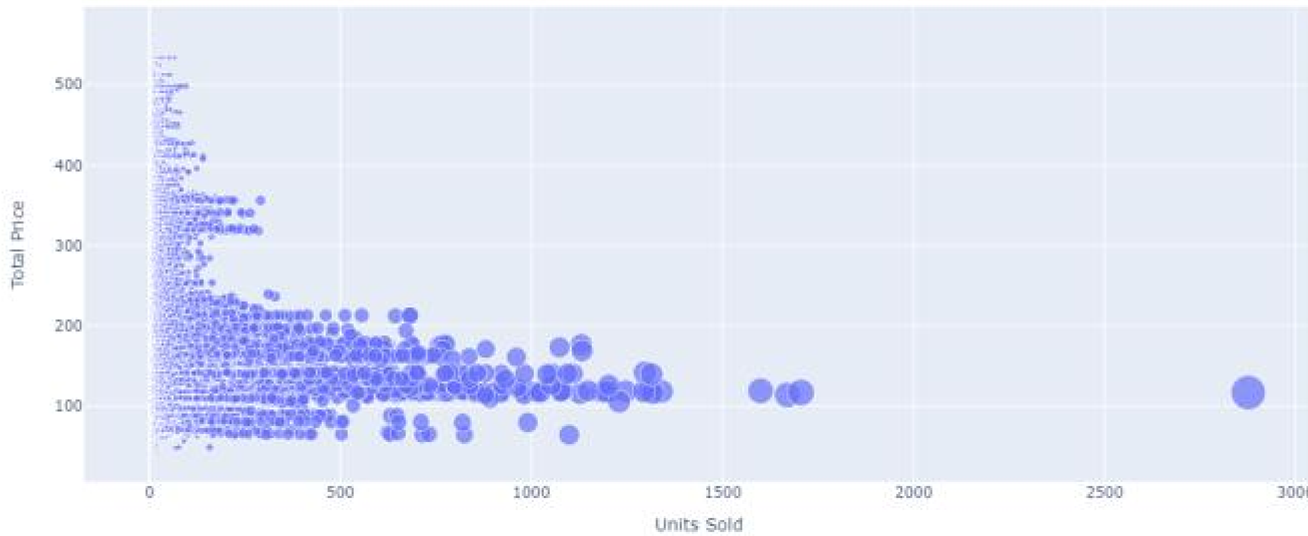
So the dataset has only one missing value in the **Total Price** column, I will remove that entire row for now:

```
data = data.dropna()
```

Let us now analyze the relationship between the price and the demand for the product. Here I will use a **scatter plot** to see how the demand for the product varies with the price change:

```
fig = px.scatter(data, x="Units Sold", y="Total Price",size='Units Sold')

fig.show()
```



We can see that most of the data points show the sales of the product is increasing as the price is decreasing with some exceptions. Now let's have a look at the correlation between the features of the dataset:

```
print(data.corr())
```

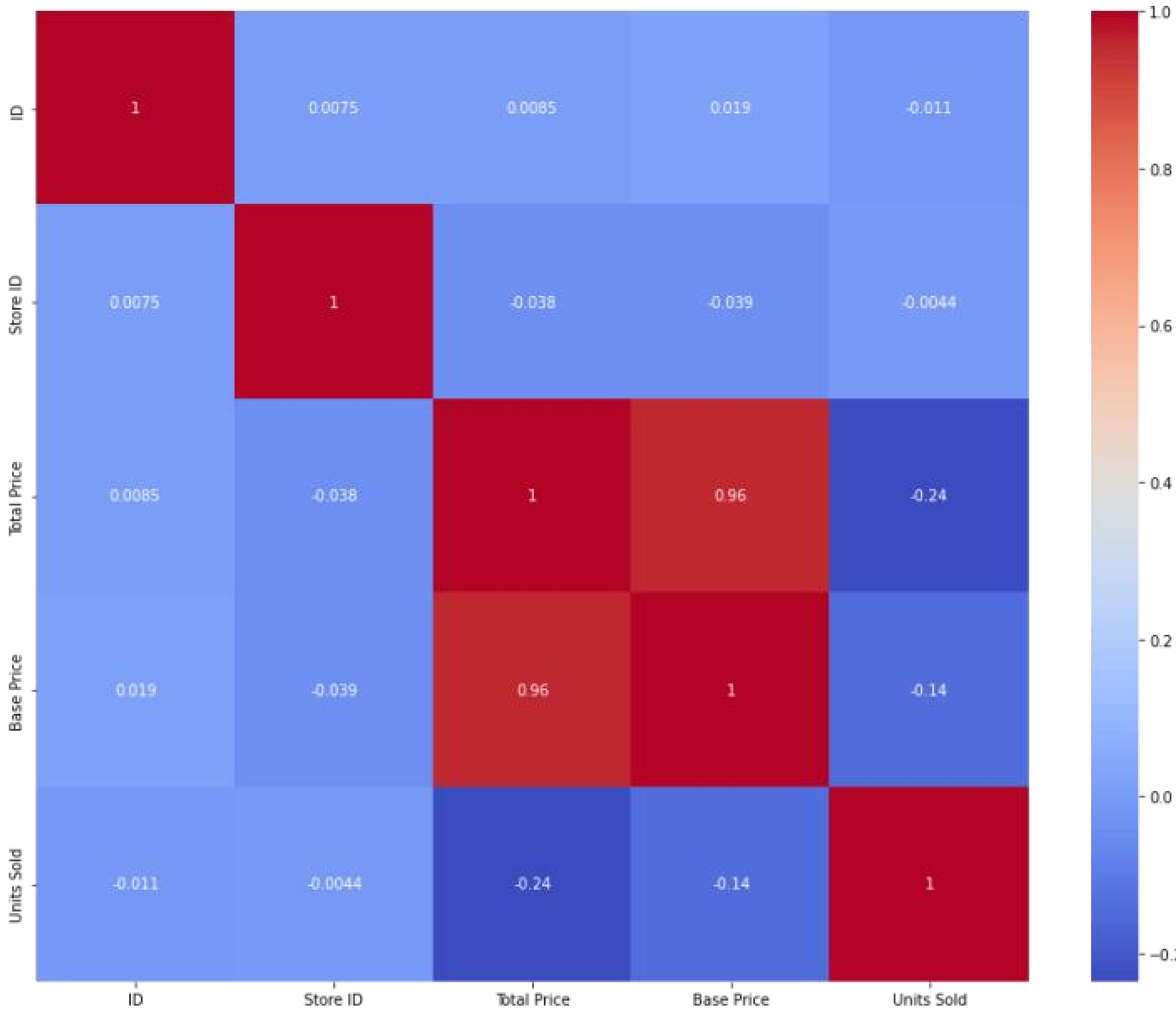
	ID	Store ID	Total Price	Base Price	Units Sold
ID	1.000000	0.007464	0.008473	0.018932	-0.010616
Store ID	0.007464	1.000000	-0.038315	-0.038848	-0.004372
Total Price	0.008473	-0.038315	1.000000	0.958885	-0.235625
Base Price	0.018932	-0.038848	0.958885	1.000000	-0.140032
Units Sold	-0.010616	-0.004372	-0.235625	-0.140032	1.000000

```
correlations = data.corr(method='pearson')

plt.figure(figsize=(15, 12))

sns.heatmap(correlations, cmap="coolwarm", annot=True)

plt.show()
```



Product Demand Prediction Model

Now let's move to the task of training a machine learning model to predict the demand for the product at different prices. I will choose the **Total Price** and the **Base Price** column as the features to train the model, and the **Units Sold** column as labels for the model:

```
x = data[["Total Price", "Base Price"]]

y = data["Units Sold"]
```

Now let's split the data into training and test sets and use the decision tree regression algorithm to train our model:

```
xtrain, xtest, ytrain, ytest = train_test_split(x, y,

                                                test_size=0.2,

                                                random_state=42)

from sklearn.tree import DecisionTreeRegressor

model = DecisionTreeRegressor()

model.fit(xtrain, ytrain)


#features = [["Total Price", "Base Price"]]

features = np.array([[133.00, 140.00]])

model.predict(features)

array([27.])
```

Summary:

So this is how you can train a machine learning model for the task of product demand prediction using Python.