*Forecasting Product Demand with Machine Learning*

Problem Statement :-

The problem at hand is to develop a machine learning model that can predict product demand based on historical sales data and external factors. The key challenges include dealing with large datasets, handling missing data, selecting appropriate features, and choosing the right machine learning algorithms for accurate predictions.

Machine Learning Model used :-

A decision tree is a powerful and interpretable machine learning algorithm used for both classification and regression tasks. It recursively splits the dataset into subsets based on the most significant attribute at each node. The goal is to create a tree-like structure where each internal node represents a decision based on a particular feature, and each leaf node represents the predicted outcome.

In the context of regression, like in your code, decision trees predict a continuous target variable. The splitting process involves selecting the feature and the split point that minimizes the variance of the target variable within the subsets. The result is a tree that can make predictions for new data by traversing the branches from the root to a leaf node.

Decision trees are advantageous for their simplicity, interpretability, and ability to handle non-linear relationships in data. However, they are prone to overfitting, especially when deep trees are constructed. Techniques like pruning or using ensemble methods (e.g., Random Forests) can help mitigate this issue and enhance the model's generalization performance.

Code :-

*import pandas as pd*

*import numpy as np*

*import seaborn as sns*

*import matplotlib.pyplot as plt*

*from sklearn.model\_selection import train\_test\_split*

*from sklearn.tree import DecisionTreeRegressor,plot\_tree*

*data = pd.read\_csv('C:\\Users\\Mugun\\Desktop\\Dataset\\PoductDemand.csv')*

*print("Description of columns : \n")*

*print(data.describe()) #decription of each column*

*print("\nNo.of Null Columns :\n",data.isnull().sum()) #count of null values in columns*

*data = data.dropna() #to remove null data*

*print("\nHeatmap to show correlation between attributes :-")*

*correlation=data.corr(method='pearson')*

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

*plt.show() #To show correlation between attributes*

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

*y=data["Units Sold"]*

*xtr,xte,ytr,yte=train\_test\_split(x,y,test\_size=0.2,random\_state=42)*

*model = DecisionTreeRegressor()*

*model.fit(xtr,ytr)*

*features = np.array([[100, 120]])*

*us=model.predict(features)*

*print("Total Price : Rs.",features[0][0])*

*print("Base Price : Rs.",features[0][1])*

*print("Predicted value for Unit Sold : ",us[0])*

Explanation :-

This code uses the Python libraries pandas, numpy, seaborn, and scikit-learn to build a decision tree regression model for predicting the number of units sold based on the total price and base price of a product. It starts by loading a dataset from a CSV file and performs data cleaning by dropping rows with null values. The code then visualizes the correlation between attributes using a heatmap. The dataset is split into training and testing sets, and a decision tree regression model is trained on the training data. Finally, the model is used to predict the number of units sold for a given set of total and base prices, and the result is printed.

Output:-

