PRODUCT DEMAND PREDICTION WITH MACHINE LEARNING

PHASE 03: Development Part 1

INTRODUCTION:

Product prediction involves using data and analysis to make informed forecasts about future product trends, demand, or market behavior. This can be done in various industries, including retail, e-commerce, manufacturing, and more

Machine learning techniques, such as regression, time series analysis, and deep learning, are commonly used for product prediction. Additionally, predictive analytics software and tools can assist in automating and streamlining the prediction process.

It's essential to remember that product prediction is an ongoing process, and the accuracy of predictions can improve over time as more data becomes available and the model becomes more refined.

Development model:

Creating a machine learning model for product demand prediction involves several steps, including data preparation, model selection, training, and evaluation. Below is a simple example of how to perform product demand prediction using Python and the scikit-learn library. For a more complex and accurate model, you may need more data and advanced techniques, but this example serves as a starting point.

```
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()
```

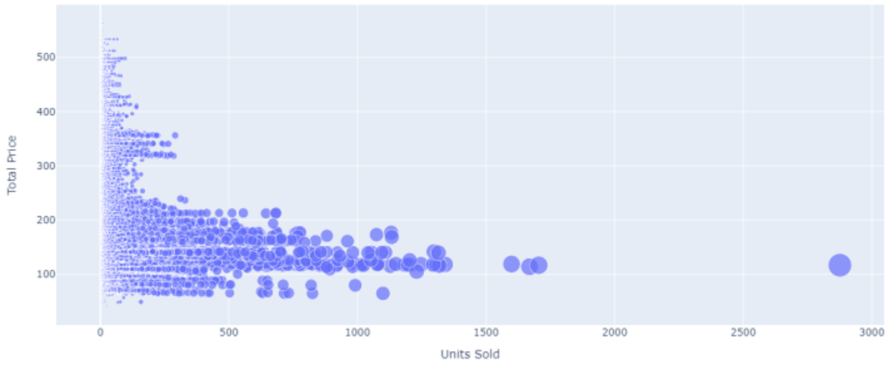
```
   ID  Store ID  Total Price  Base Price  Units Sold
0   1      8091      99.0375    111.8625          20
1   2      8091      99.0375     99.0375          28
2   3      8091     133.9500    133.9500          19
3   4      8091     133.9500    133.9500          44
4   5      8091     141.0750   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

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

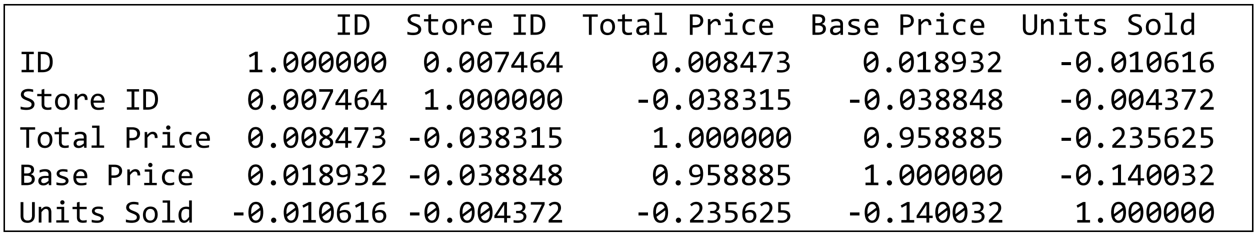
```
data.isnull().sum()

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

fig.show()
```



print(data.corr())



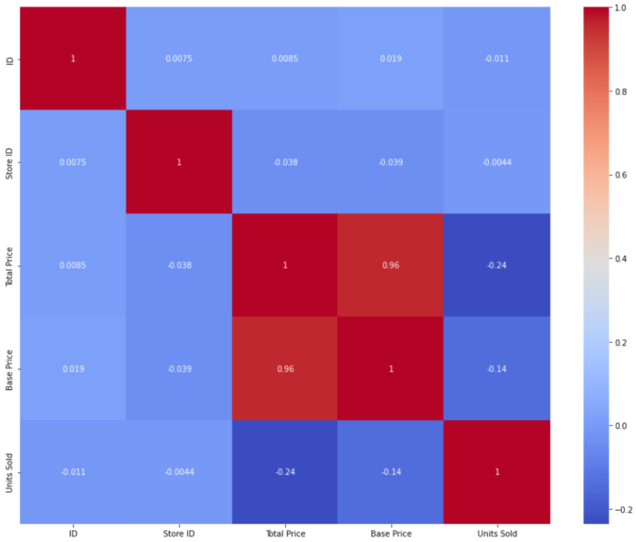
```
                     ID  Store ID  Total Price  Base Price  Units Sold
ID             1.000000  0.007464     0.008473    0.018932   -0.010616
Store ID       0.007464  1.000000    -0.038315   -0.038848   -0.004372
Total Price    0.008473 -0.038315     1.000000    0.958885   -0.235625
Base Price  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

Now let’s input the features (Total Price, Base Price) into the model and predict how much quantity can be demanded based on those values:

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

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

model.predict(features)

array([27.])
```

Summary:

Product Demand prediction with machine learning model prepared for Decision Tree Regressor in this phase.

THANK YOU !