

Product Demand Prediction (Case Study)

A product company plans to offer discounts on its product during the upcoming holiday season. The company wants to find the price at which its product can be a better deal compared to its competitors. For this task, the company provided a dataset of past changes in sales based on price changes. You need to train a model that can predict the demand for the product in the market with different price segments.

The **dataset** that we have for this task contains data about:

1. the product id;
2. store id;
3. total price at which product was sold;
4. base price at which product was sold;
5. Units sold (quantity demanded);

I hope you now understand what kind of problem statements you will get for the product demand prediction task. In the section below, I will walk you through predicting product demand with machine

Product Demand Prediction using Python

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

```
1 import pandas as pd
2 import numpy as np
3 import plotly.express
4 import seaborn as sns
5 import matplotlib.pyplot as plt
6 from sklearn.model_selection import train_test_split
7 from sklearn.tree import DecisionTreeRegressor
8
9 data = pd.read_csv("https://raw.githubusercontent.com/IBM/retail-demand-prediction/master/data/retail-demand.csv")
10 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:

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

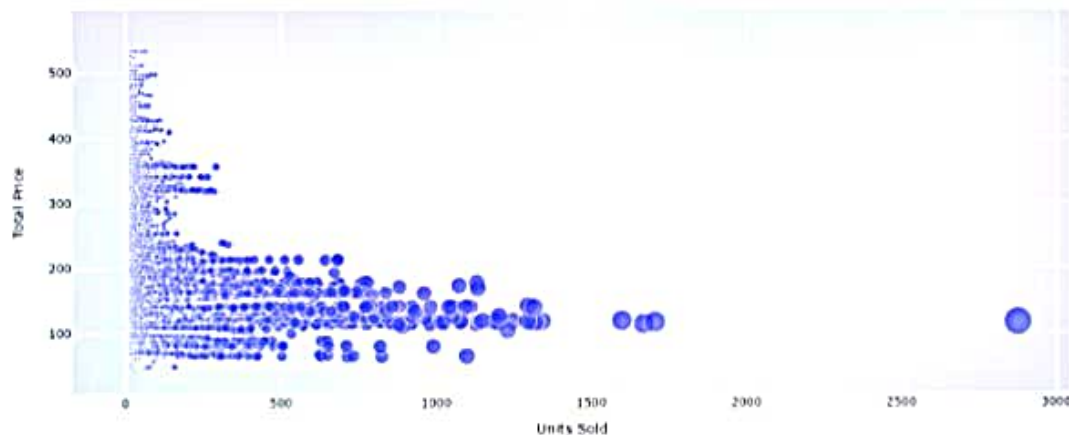
```
ID          0
Store ID     0
Total Price  1
Base Price   0
Units Sold   0
dtype: int64
```

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

```
1 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


```
1 fig = px.scatter(data,  
2                     size=  
3 fig.show())
```



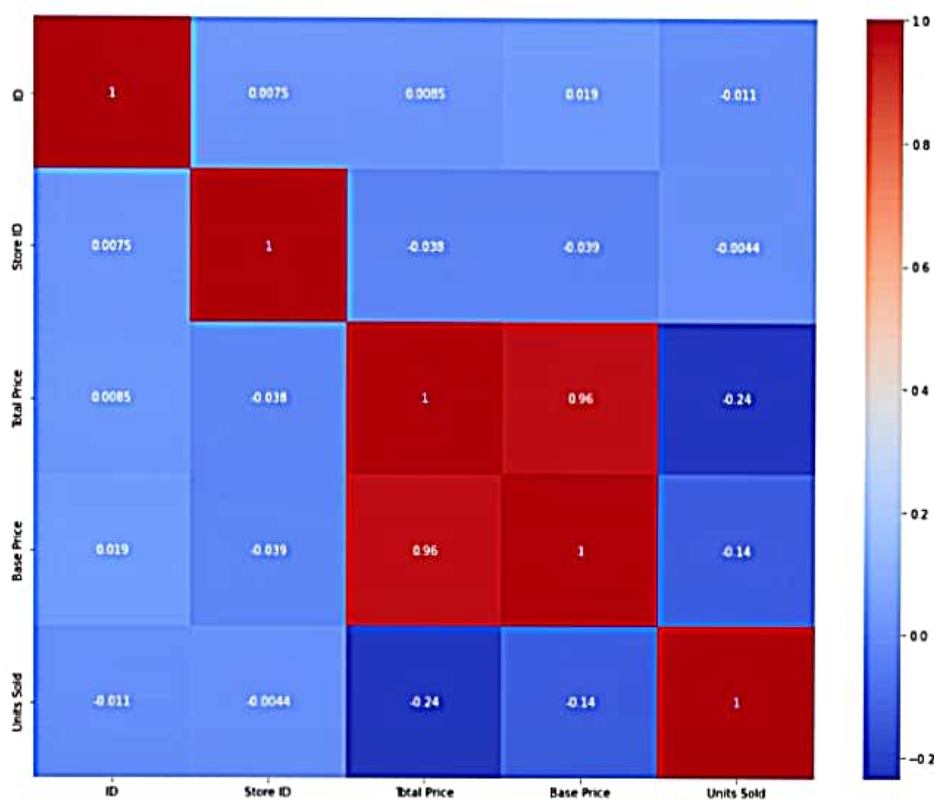
We can see that most of the data points show the sales of the product is increasing as the price is decreasing with some exceptions. Now let's have a look at the correlation between the features of the dataset:

```
1 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
```

```
1 correlations = data.co
2 plt.figure(figsize=(15
3 sns.heatmap(correlatio
4 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

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:

```
1 x = data[["Total Price  
2 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:

```
1 xtrain, xtest, ytrain,  
2  
3  
4 from sklearn.tree impo  
5 model = DecisionTreeRe  
6 model.fit(xtrain, ytra
```

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:

```
1 #features = ["Total P  
2 features = np.array([[  
3 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. Price is one of the major factors that affect the demand for the product. If a product is not a necessity, only a few people buy the product even if the price increases. I hope you liked this article on product demand prediction with machine learning using Python. Feel free to ask your valuable questions in the comments section below.