

Product Demand prediction

Using Machine Learning

Presented by:S Dhanasekar



Product Demand (Case study)

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:

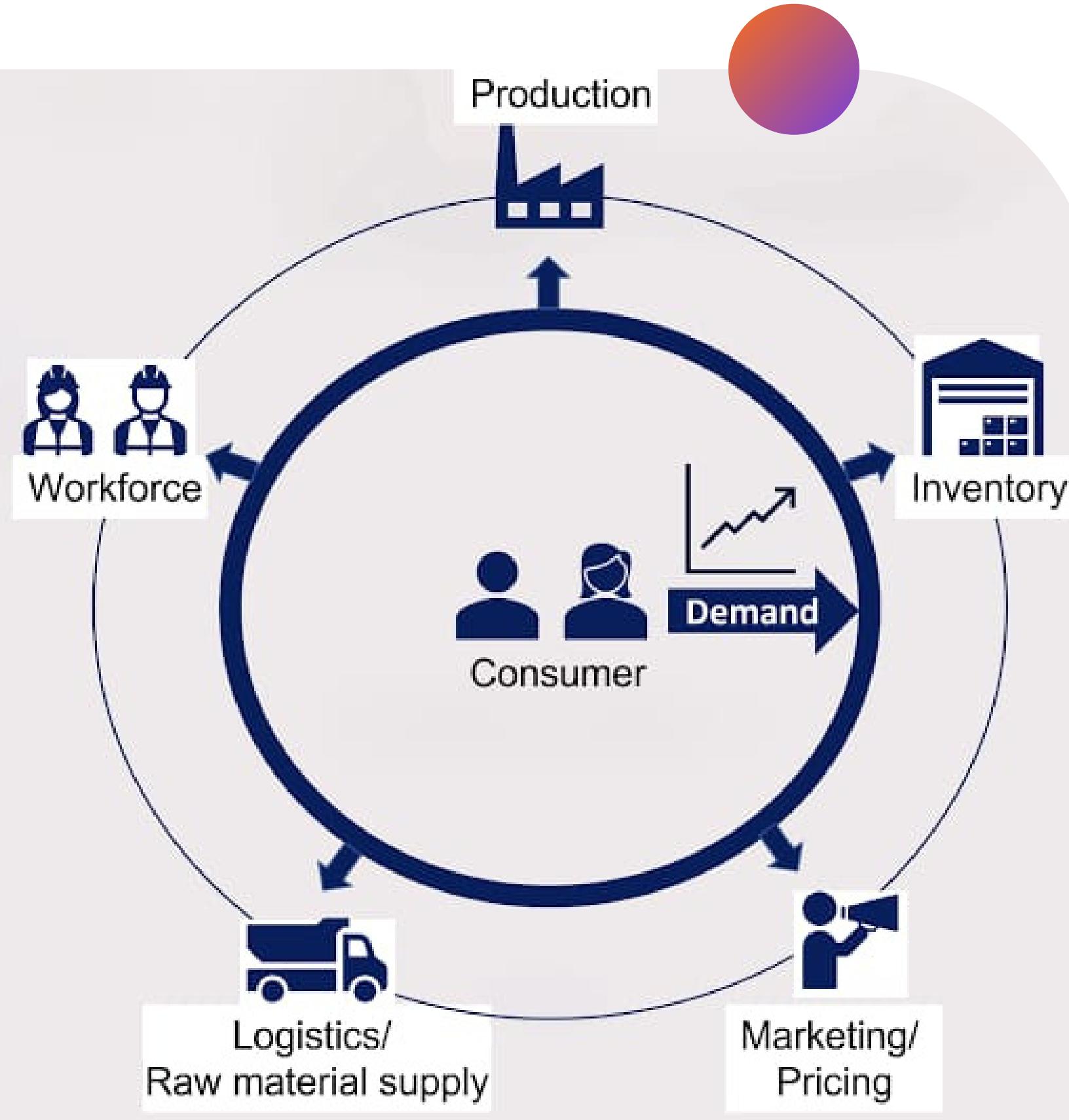
- 1.The product id;
- 2.Store id;
- 3.Total price at which product was sold;
- 4.Base price at which product was sold;
- 5.Units sold (quantity demanded);





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:



Program For Product demand prediction using Machine Learning

```
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()
```

Dataset:

ID	Store ID	Total Price	Base Price	units sold
0	1	8091	99.0375	111.8625
1	2	8091	99.0374	99.0375
2	3	8091	133.9500	133.9500

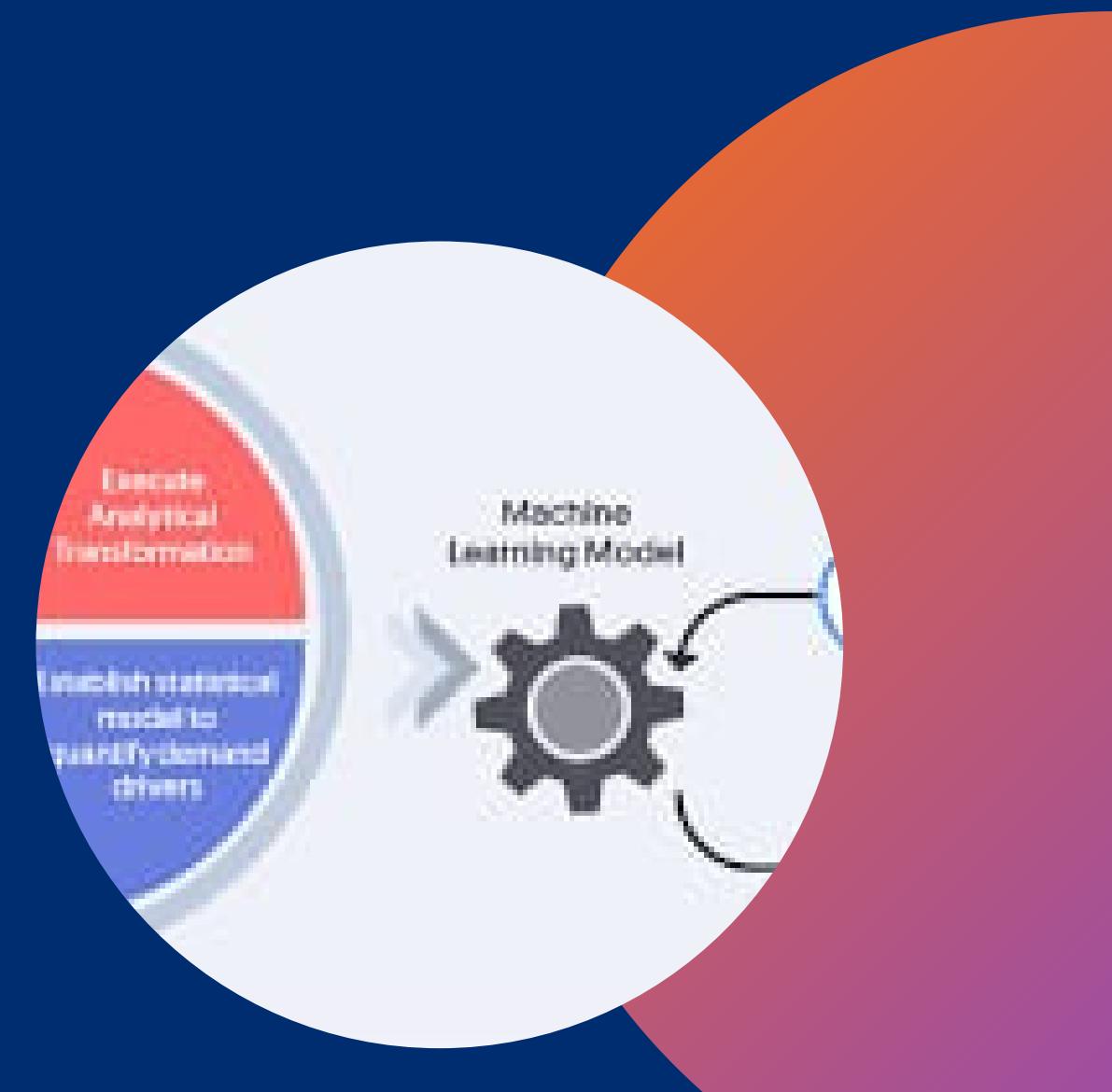


Now let's have a look at whether this dataset contains any null values or not:

`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 value in remove that entire row for now:

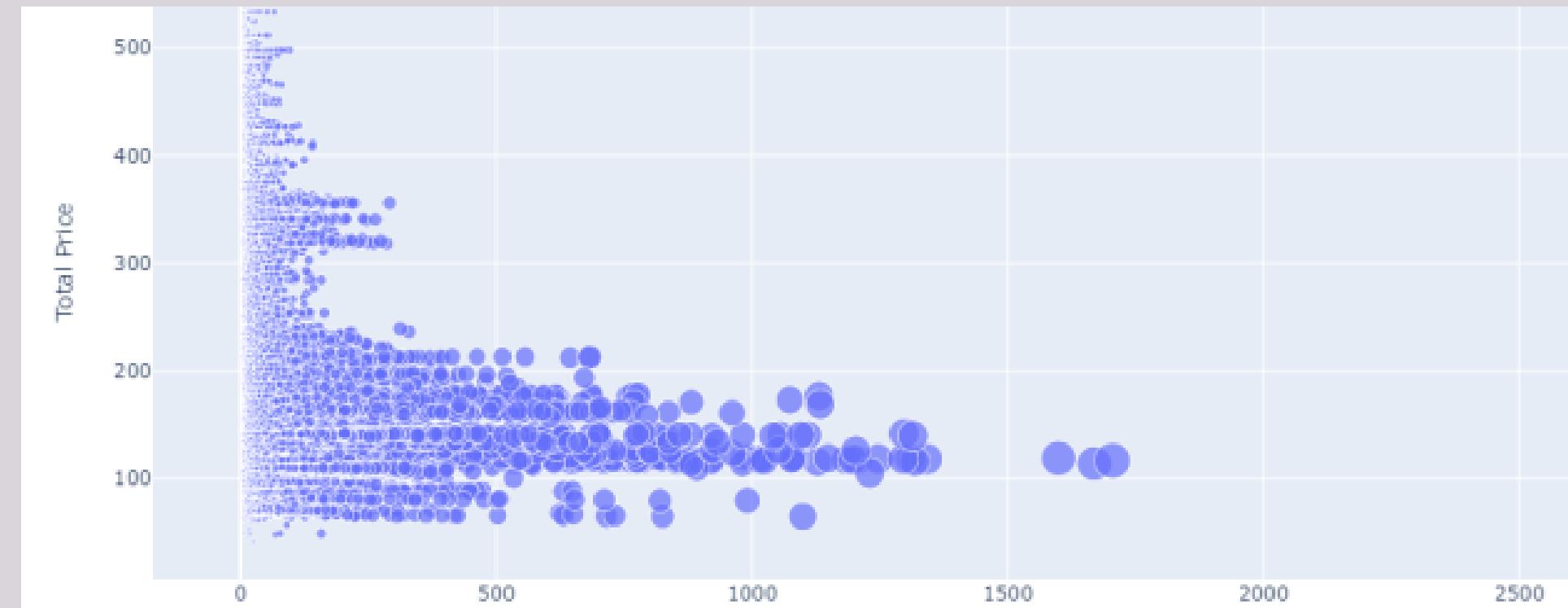


data=data.dropna()

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="Unitssold",y="Totalprice",  
'Size=units sold')
```



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
1.00	0.74	0.00	0.01	111.86 25
1.20	1.00	-0.03	-0.03	99.037 5
0.00 7	-0.83	1.000	0.95	133.95 00



```
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:

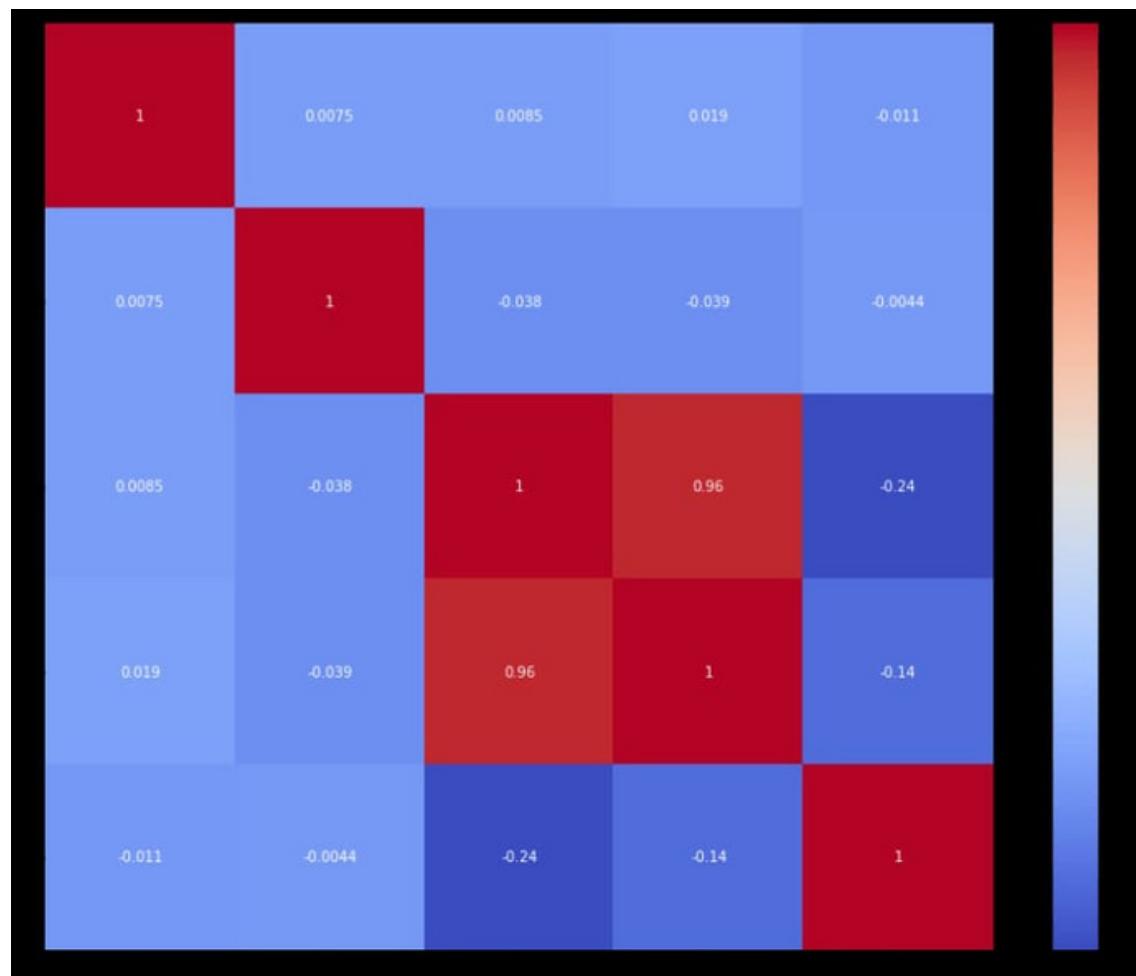
```
x=data[["Total price","Base price"]]  
y=data[["Units sold"],
```

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=0.1,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)
```





STUDIO SHODWE

Thank You

SLIDE PRESENTATIONS DESIGN