# IBM NAAN MUDHALVAN PHASE – I

**Name :** M.Thanes

**Roll No :** 2021503712

You must have studied that the demand for a product varies with the change in its price. If you take real-world examples, you will see if the product is not a necessity, then its demand decreases with the increase in its price and the demand increases with the decrease in its price. If you want to know how we can predict demand for a product with machine learning, this project is for you. In this project, let's walk through the task of product demand prediction with machine learning using Python.

## 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:

the product id; store id; total price at which product was sold; base price at which product was sold; Units sold (quantity demanded);

Hope you now understand what kind of problem statements you will get for the product demand prediction task. In the section below, let's walk through predicting product demand with machine learning using Python.

## Product Demand Prediction using Python

libraries and the dataset need for the task of product demand prediction:

```
In [1]:
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("demand.csv")
data.head()
```

Out[1]:

| ID | Store ID | Total Price | Base Price | Units Sold |
|---|---|---|---|---|
| **0** | 1 | 8091 | 99.0375 | 111.8625 |
| **1** | 2 | 8091 | 99.0375 | 99.0375 |
| **2** | 3 | 8091 | 133.9500 | 133.9500 |
| **3** | 4 | 8091 | 133.9500 | 133.9500 |
| **4** | 5 | 8091 | 141.0750 | 141.0750 |

In [2]:

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

Out[2]:

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

In [3]:

```
#So the dataset has only one missing value in the Total Price column,  remove that entire row for now:
data = data.dropna()
```

In [4]:

```
#Let us now analyze the relationship between the price and the demand for the product. We are using 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()
```

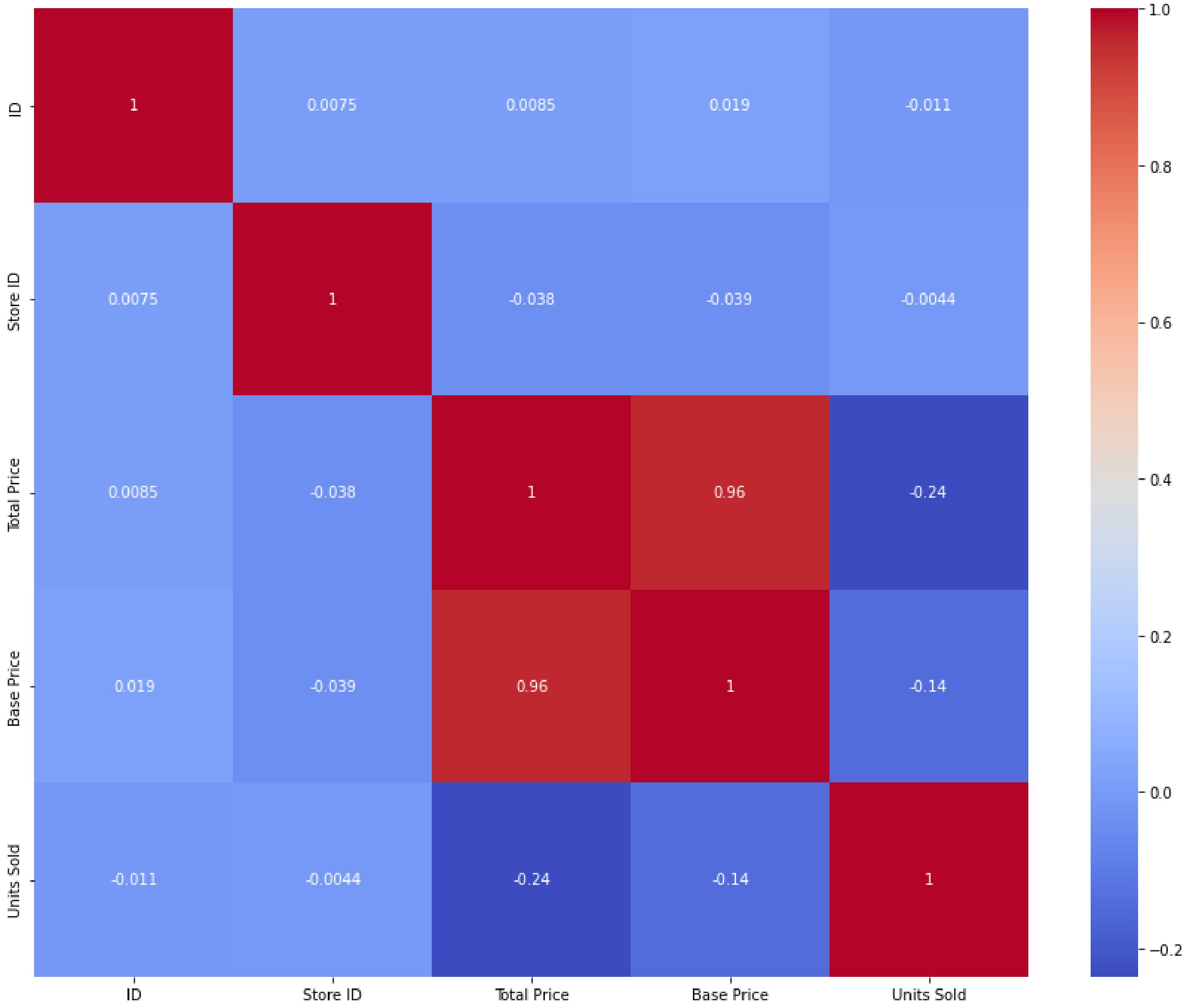
We can see that most of the data points show the sales of the product is increasing as the price is decreasing with some exceptions.

In [5]:

```
#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  0.007461     0.008473    0.018911   -0.010608
Store ID    0.007461  1.000000    -0.038315   -0.038855   -0.004369
Total Price  0.008473 -0.038315     1.000000    0.958885   -0.235625
Base Price   0.018911 -0.038855     0.958885    1.000000   -0.140022
Units Sold  -0.010608 -0.004369    -0.235625   -0.140022    1.000000
In [6]:
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. We 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:

In [7]:

```
x = data[["Total Price", "Base Price"]]
y = data["Units Sold"]
```

In [8]:

```
#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)
```

Out[8]:

```
DecisionTreeRegressor()
```

**In a Jupyter environment, please rerun this cell to show the HTML representation or trust the notebook. On GitHub, the HTML representation is unable to render, please try loading this page with nbviewer.org.**

In [12]:

```
#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, 150.00]])
model.predict(features)
```

```
C:\Python39\lib\site-packages\sklearn\base.py:450: UserWarning:

X does not have valid feature names, but DecisionTreeRegressor was fitted with feature names
```

Out[12]:

```
array([16.])
```

## 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.