**Product Demand Prediction Using Machine Learning**

A product manufacturer intends to provide sales on its merchandise in the run-up to the holidays. The business is trying to figure out how much its product can provide that is more than that of its rivals. The business made available a dataset of previous price-based changes in sales for this job. In order to forecast product demand across various price points in the market, we must train a model.

The dataset we are using for this challenge includes information regarding:

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);

Now that you know, hopefully, what sort of issue statements you will encounter in the work of predicting product demand. I'll demonstrate how to use Python and machine learning to estimate product demand in the part that follows.

To begin, let's import the dataset and required Python modules for the product demand forecast task:

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

Lets now load our data set and print the information on the data set

data = pd.read\_csv("/content/dataset.csv")

data.head()

Output for the above code:

|  | **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**](https://thecleverprogrammer.com/2020/12/20/scatter-plot-with-python/) 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()

The plot for the above code is as follows:

A screenshot of a computer

Description automatically generated

With a few exceptions, most of the data indicators indicate that product sales are rising while prices are falling. Let's now examine the relationship between the dataset's features:

print(data.corr())

A close-up of numbers

Description automatically generated

correlations = data.corr(method='pearson')

plt.figure(figsize=(15, 12))

sns.heatmap(correlations, cmap="cool", annot=True)

plt.show()

A screenshot of a computer screen

Description automatically generated

The next step is to train a machine learning model to forecast the product's demand at various pricing points. I'll use the Units Sold column as the model's labels and the Total Price and Base Price columns as the model's training features:

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

y = data["Units Sold"]

Let's now divide the data into training and test sets and train our model using the decision tree regression algorithm:

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)

Let's now enter the characteristics (Total Price, Base Price) into the model and use those values to forecast the maximum amount that may be demanded:

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

features = np.array([[120.00, 135.00]])

model.predict(features)

array([49.16666667])

Summary:

Thus, this is how to use Python to train a machine learning model for the purpose of predicting product demand. One of the main elements influencing the product's demand is price. Few individuals purchase a thing if it is not absolutely necessary, even if the cost goes up. This post on predicting product demand using machine learning and Python is hopefully interesting to you. Please feel free to post any insightful queries in the space below the comments.