**Product Demand Prediction with machine learning**

Name; suriya.s

reg no; 912721104033

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**](https://raw.githubusercontent.com/amankharwal/Website-data/master/demand.csv) 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 learning using Python.

**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:

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()

<https://i0.wp.com/thecleverprogrammer.com/wp-content/uploads/2021/11/product-demand-relationships.png?resize=768%2C348&ssl=1>

**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.corr(method='pearson')

2

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

3

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

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 model:



1

x = data[["Total Price", "Base 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, ytest = train\_test\_split(x, y,

2

test\_size=0.2,

3

random\_state=42)

4

from sklearn.tree import DecisionTreeRegressor

5

model = DecisionTreeRegressor()

6

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:



1

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

2

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

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.