**PRODUCT DEMAND PREDICTION WITH MACHINE LEARNING**

**Phase-4 submission document**

**Project Title: Product Demand Prediction With Machine Learning**

**Phase 4: Development Part 2**

**Introduction:**

* The product demand 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 article is for you.
* In this article, I will walk you through the task of product demand prediction with machine learning using Python.
* 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.

**Given data set:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID | STORE ID | TOTAL  PRICE | BASE  PRICE | UNITS  SOLD |
| 1 | 8091 | 99.0375 | 111.8625 | 20 |
| 2 | 8091 | 99.0375 | 99.0375 | 28 |
| 3 | 8091 | 133.95 | 133.95 | 19 |
| 4 | 8091 | 133.95 | 133.95 | 44 |
| 5 | 8091 | 141.075 | 141.075 | 52 |
| 9 | 8091 | 227.2875 | 227.2875 | 18 |
| 10 | 8091 | 327.0375 | 327.0375 | 47 |
| 13 | 8091 | 210.0 | 210.9 | 50 |
| 14 | 8091 | 234.4125 | 234.4125 | 82 |
| 17 | 8095 | 99.0375 | 327.00375 | 99 |
| 19 | 8095 | 97.6125 | 210.9 | 120 |
| 22 | 8095 | 98.325 | 234.4125 | 40 |
| 23 | 8095 | 133.2375 | 99.0375 | 60 |

**Overview of the process:**

* The given above dataset contain “5000”Rows and 5 columns.
* 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);

**Feature Engineering:**

* Feature engineering is the pre-processing step of machine learning, which is used to transform raw data into features that can be used for creating a predictive model using Machine learning or statistical Modelling.
* Feature engineering in machine learning aims to improve the performance of models.

**Program:**

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(“[www.Kaggle.com/datasets/chakradharmattapalli/product-demand-prediction-with-machine-learning](http://www.Kaggle.com/datasets/chakradharmattapalli/product-demand-prediction-with-machine-learning)”)

data.head()

**Output:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID | STORE ID | TOTAL  PRICE | BASE  PRICE | UNITS  SOLD |
| 1 | 8091 | 99.0375 | 111.8625 | 20 |
| 2 | 8091 | 99.0375 | 99.0375 | 28 |
| 3 | 8091 | 133.95 | 133.95 | 19 |
| 4 | 8091 | 133.95 | 133.95 | 44 |
| 5 | 8091 | 141.075 | 141.075 | 52 |

**Model Training:**

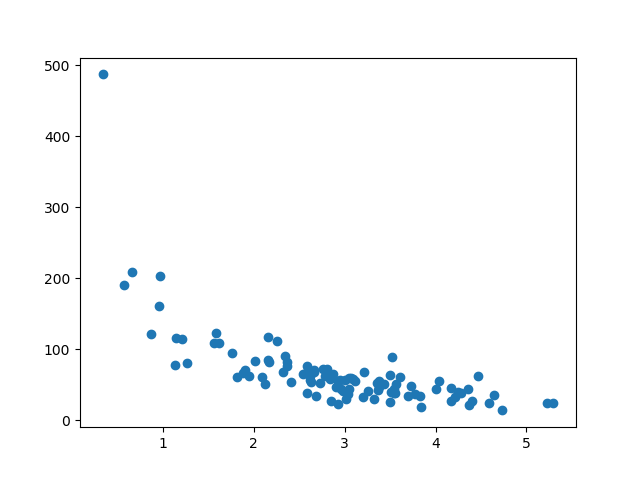
* Model training is the phase in the data science development lifecycle.
* where practitioners try to fit the best combination of weights and bias to a machine learning algorithm to minimize a loss function over the prediction range.

**Program:**

import numpy  
import matplotlib.pyplot as plt  
numpy.random.seed(2)  
  
x = numpy.random.normal(3, 1, 100)  
y = numpy.random.normal(150, 40, 100) / x  
  
plt.scatter(x, y)  
plt.show()

**Output:**

* The x axis represents the number of minutes before making a purchase.
* The y axis represents the amount of money spent on the purchase.

****

**Evaluation:**

* The profit a business can generate always depends on how accurately you understand the current demand.
* However, demand forecasting and planning might become complicated when you operate more than one selling point, there are different items in the stock, and some of them don’t have sales data at all.
* To calculate the demand for a single venue, we have to take into account multiple factors such as:

1. **Seasonality** – past changes in demand which correlate with season changes or different time periods, f.e. day of the week, or other periods of time.
2. **Geography** – the difference in demand between regions of sales.
3. **Marketing** – or the impact of your marketing campaigns on sales.
4. **Competition** – the impact of competitor actions, like price changes, ongoing marketing campaigns on your own demand.
5. **Global economy** – such as the response to overall economic events and the buyers ability, inflation, etc.
6. **Product types** – or the correlation between all the aforementioned factors with each separate product type.

If you perform demand forecasting in some manual form with the help of spreadsheets, you might already have a set of data that you use for calculations. So at this point, it already makes sense to think about automating the whole process if you have a number of preconditions:

### DEMAND DATA AVAILABILITY:

* Sales data is the most available piece of information that allows us to start working with [forecasting models](https://mobidev.biz/blog/ai-machine-learning-forecasting-algorithms-models-for-business).
* But, demand forecasting requires more than just historical sales.
* And if you already set up some processes and infrastructure to collect data on your internal and external factors, it can be used for machine learning easier than collecting it from scratch.
* **A three-month period of historical data for one item** is enough to kick start forecasting with ML. And the more data you’ll have, the better.
* Additionally, if you already have historical data for multiple years from the past, it will allow you to start developing an ML solution and already bring value.
* Since predicting inventory level implies we need your specific sales, which can’t be sourced elsewhere or bought on a dataset market.

### DATA USABILITY:

* For each product, the seasonality cycle plays a crucial role in predicting demand. Because, for instance, if you sell tents for hiking, its seasonal growth in demand appears to be during the summer period, with peaks in a certain month in your area.
* If we take the data for only 5 months for training the ML model, the prediction of the machine learning model won’t give adequate results, since we need a year’s data as a bare minimum, to calculate the seasonality.
* For products that don’t fluctuate in demand seasonally, say forks and spoons, a three-month data will be enough to start training the model and producing forecasts.
* But it always depends on the item itself and other external factors like competition, or amount of holidays for a given period.

### PREDICTION FREQUENCY:

* Different business types will require different frequency of forecasting.
* Conducting analysis on a monthly or even weekly basis can be a challenging task in terms of manual forecasting.
* In this case, it’s a no-brainer that using machine learning frees up the resources, because it suggests an automated pipeline that gathers data and provides demand forecasts for given periods.
* One thing to keep in mind is that, **the shorter the forecasting period, the less accurate it will be**, since it’s impossible to generate enough data to cover all the required time-changes.

### MULTIPLE VENUE DEMAND FORECASTING:

* For entrepreneurs that own multiple selling points that are set across different regions, forecasting demand will require a lot of human resources.
* Since seasonality, geography, and competition will all be different.
* Setting up a system that sources data scattered across multiple databases, and presents analytics through a single dashboard, reduces the cost of demand forecasting itself. In its turn, a unified analytical solution presents a 360 view on your venues, inventory and sales activities allowing more flexibility in terms of planning and inventory management.
* Based on our experience building a demand forecasting module for our client, SmartTab, we want to point out that the use of ML for demand forecasting is a large competitive advantage.
* Since your business obtains a better visibility and frees the human resources for development activities, rather than focusing on operational things.

**Program:**

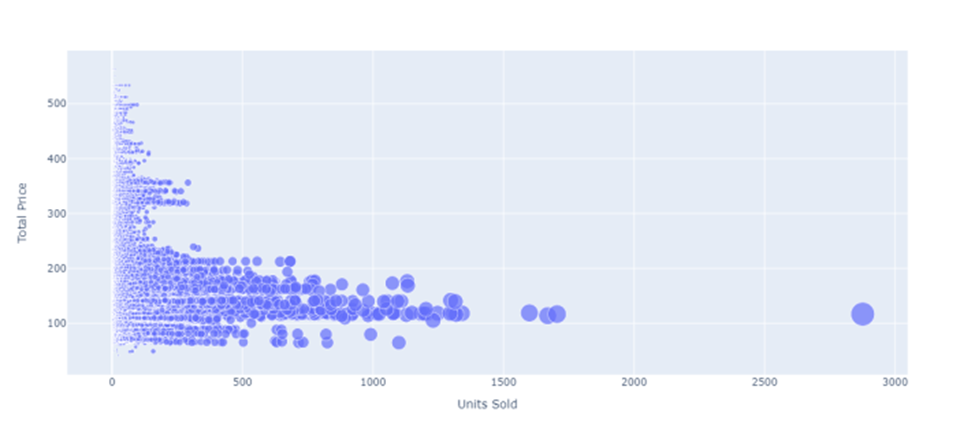
Let us now analyze the relationship between the price and the demand for the product. Here I will use a [**scatter plot**](https://thecleverprogrammer.com/2020/12/20/scatter-plot-with-python/) to see how the demand for the product varies with the price change.

fig = px.scatter(data, x="Units Sold", y="Total Price",

size='Units Sold')

fig.show()

**Output:**

****

**Data preprocessing:**

**Once the dataset is loaded into the machine learning environment, you may need to preprocess it before you can start training and evaluating your model. This may involve cleaning the data, transforming the data into a suitable format, and splitting the data into training and**

**test sets.**

**Program:**

1. #handling missing data (Replacing missing data with the mean value)
2. from sklearn.preprocessing import Imputer
3. imputer= Imputer(missing\_values ='NaN', strategy='mean', axis = 0)
4. #Fitting imputer object to the independent variables x.
5. imputerimputer= imputer.fit(x[:, 1:3])
6. #Replacing missing data with the calculated mean value
7. x[:, 1:3]= imputer.transform(x[:, 1:3])

**output:**

array([['India', 38.0, 68000.0],

['France', 43.0, 45000.0],

['Germany', 30.0, 54000.0],

['France', 48.0, 65000.0],

['Germany', 40.0, 65222.22222222222],

['India', 35.0, 58000.0],

['Germany', 41.111111111111114, 53000.0],

['France', 49.0, 79000.0],

['India', 50.0, 88000.0],

['France', 37.0, 77000.0]], dtype=object

**Import the libraries**:

**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(“E:/product\_demand.csv”)**

**data.head()**

**OUTPUT:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **STORE ID** | **TOTAL PRICE** | **BASE PRICE** | **UNITS SOLD** |
| **1** | **8091** | **99.0375** | **111.8625** | **20** |
| **2** | **8091** | **99.0375** | **99.0375** | **28** |
| **3** | **8091** | **133.95** | **133.95** | **19** |
| **4** | **8091** | **133.95** | **133.95** | **44** |

**Conclusion:**

* It helps businesses make informed decisions that affect everything from inventory planning to supply chain optimization.
* With customer expectations changing faster than ever, businesses need a method to forecast demand accurately.
* In the quest to build a product demand prediction model, we have embarked on a critical journey that begins with loading and preprocessing the dataset.
* We have traversed through essential steps, starting with importing the necessary libraries to facilitate data manipulation and analysis.