PRODUCT DEMAND PREDICTION WITH MACHINE LEARNINGS...

PHASE-5

TEAM MEMBERS:

1.A.ASHKA(821021104011)
2.S.BANUPRIYA(821021104014)
3.J.JAYAPRIYA(821021104023)
4.L.PRIYA NANDHINI(821021104036)
5.N.VARSHA(821021104053)

TABLE OF CONTENTS

PROBLEM DEFINITION:

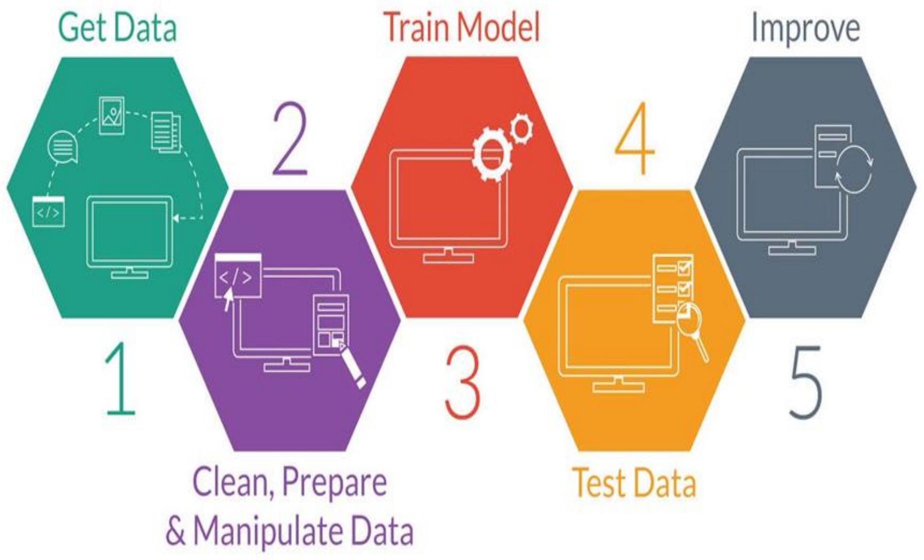
- *The problem is to create a machine learning model that forecasts product demand based on historical sales data and external factors.*
- *The goal is to help businesses optimize inventory management and production planning to efficiently meet customer needs.*
- *In real-world, if the product is not a necessity, then its demand decreases with the increase in its price and the demand increases with the decrease in its price.*
- *So, it is necessary to predict the demand of product for business.*

DESIGN THINKING:

- *Demand forecasters attempt to do just that by using predictive analysis techniques to spin data about past and present sales into predictions of what customers will want in the future.*
- *But generating predictions is only the beginning. Demand forecasting success demands that those insights inform decisions about product direction, pricing, company expansion, hiring and more*
- *And that those interpreting results don’t fall into the trap of simply trying to deliver faster horses.*

PHASES OF DEVELOPMENT:

- *This project involves 6 steps.*
- *They are,*
    - ✓ *Data Collection*
    - ✓ *Data Preprocessing*
    - ✓ *Feature Engineering*
    - ✓ *Model Selection*
    - ✓ *Model Training*
    - ✓ *Model Evaluation.*



*In **Phase1**:*

*we provided the overall **outline** and how we are going to handle the given dataset for product demand prediction.*

*In **Phase2:***

*Here we described about the **forecasting technique** that is used for the product demand prediction.*

*In **Phase3**:*

*We briefly explained about the **data collection, data preprocessing** processes.*

*In **Phase4:***

*We continued build the product demand prediction model by **Feature Engineering, Model Training and Model Evaluation.***

OUTLINE OF DATA SCIENCE PROCESS:

1.DATA COLLECTION:

• *In this part datas are collected to be first from **various sources**. Then the datas that are used for train the model.*

• *But in our case, we are already provided by datasets for product demand prediction from **KAGGLE platform.***

Dataset link: *https://www.kaggle.com/datasets/chakradharmattapalli/productdemandprediction-with-machine-learning.*

2. DATA PREPROCESSING:

In this part include major of 3 things;

1. Clean and preprocess the data:

Here, we are going to clear outliers that perform distinct significant from the other observation.

2.Handle the missing values:

- *we are working with large amount of data, therefore every is need to be consistent and every datas are need to be verified before the process*
- *If we found any entity with missing values, we have to replace a ZN () function replaces a Null value for the data field that is placed inside the brackets with a zero.*

<u>3.Convert categorical features into numerical values:</u>

• Here, we are converting the data values into numerical values. For example, most of the price values are in INR and some of them are in $, so here we have to convert them ($) into INR.

3. FEATURE ENGINEERING:

In this step we are adding additional feature that capture seasonal patterns/ trends, and external influences on product demand.

1.seasonal patterns/trends: *A certain time series with repetitive or predictable patterns of demand due to re-occurring seasonal events. These patterns can re-occur over days, weeks, months or* *quarters and can make it harder for businesses to forecast future demand trends.*

E.g., MEESHO Maha Indian sale.

2. External influences on products demand: *Such as,*

- *based on demand of the product.*
- *Competition between same product but on different manufacturers*
- *Suppliers of raw materials and goods*
- *economic conditions*
- *buyers*
- *Government*

4. MODEL SELECTION:

- *In this part we are choosing a suitable model* ***decision tree*** *algorithm for demand forecasting.*
- *Decision tree is both for classification and regression.*

5. MODEL TRAINING:

- *Decision Tree is a Supervised learning technique that can be used for both classification and Regression problems, but mostly it is preferred for solving Classification problems. It is a tree-structured classifier, where internal nodes represent the features of a dataset, branches represent the decision rules and each leaf node represents the outcome.*
- *In a Decision tree, there are two nodes, which are the Decision Node*

and Leaf Node. Decision nodes are used to make any decision and have multiple branches, whereas Leaf nodes are the output of those decisions and do not contain any further branches.

- *The decisions or the test are performed on the basis of features of the given dataset.*
- *It is a graphical representation for getting all the possible solutions to a problem/decision based on given conditions.*
- *It is called a decision tree because, similar to a tree, it starts with the root node, which expands on further branches and constructs a tree-like structure.*

- *In order to build a tree, we use the CART algorithm, which stands for Classification and Regression Tree algorithm.*
- *A decision tree simply asks a question, and based on the answer (Yes/No), it further split the tree into subtrees.*

6. MODEL EVALUATION:

• At this point, we have a trained DECISION TREE model, but we need to find out whether it is making accurate predictions.

• The simplest way to evaluate this model is using accuracy;

- *We check the predictions against the actual values in the test set and count up how many the model got right.*

These are the overall processes that are carried out there for, modelling the machine learning algorithm.

Next, we dive into all of the process deeply, and understand how the product demand prediction model has been created.

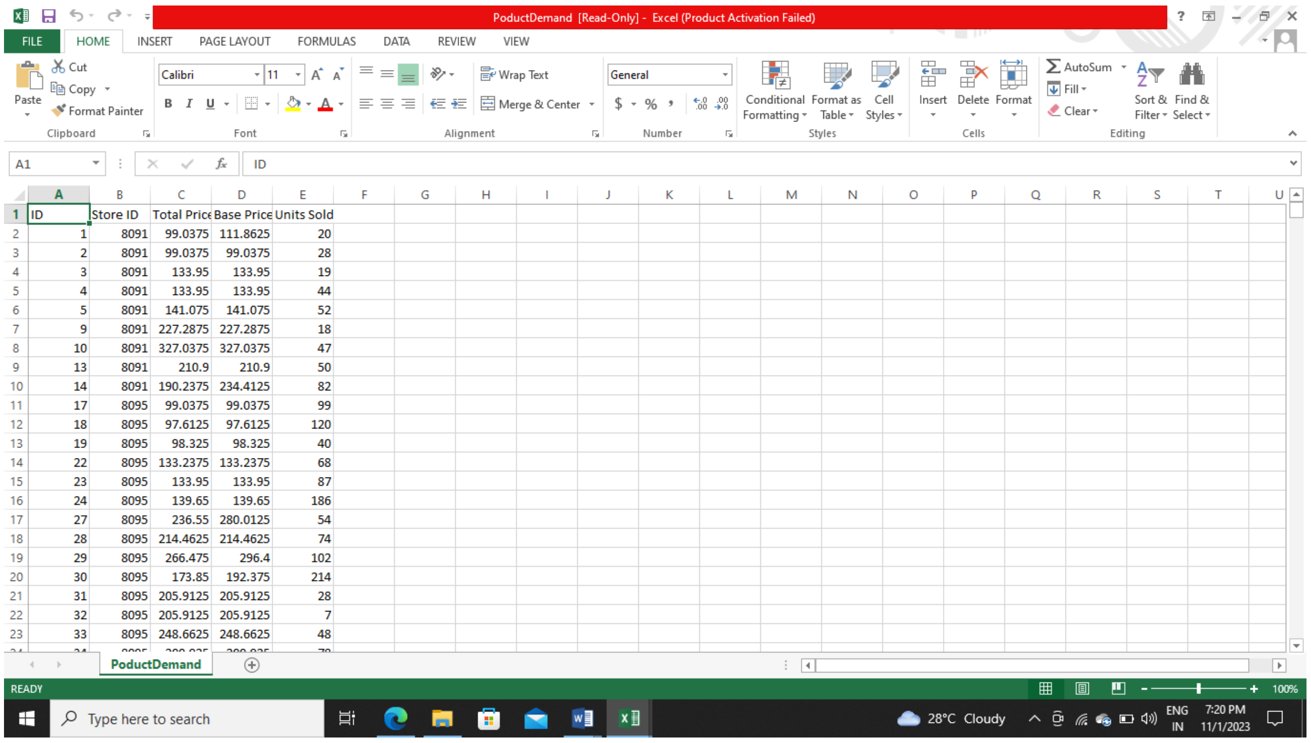
1.DATA COLLECTION:

In data collection, we don't have to collect data from various sources, because we are already provided by some datasets which is from kaggle site

LINK:

https://www.kaggle.com/datasets/chakradharmattapalli/productdemandprediction-with-machine-learning

This dataset is used for predicting the product demand forecasting.



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	210.9	210.9	50
14	8091	190.2375	234.4125	82
17	8095	99.0375	99.0375	99
18	8095	97.6125	97.6125	120
19	8095	98.325	98.325	40
22	8095	133.2375	133.2375	68
23	8095	133.95	133.95	87
24	8095	139.65	139.65	186
27	8095	236.55	280.0125	54
28	8095	214.4625	214.4625	74
29	8095	266.475	296.4	102
30	8095	173.85	192.375	214
31	8095	205.9125	205.9125	28
32	8095	205.9125	205.9125	7
33	8095	248.6625	248.6625	48

*This dataset contains **5 columns and 150150 rows.***

DATASET DESCRIPTION:

- ***The datas in the columns are based on***

1. ID of the product,

2. ID of the store,

3. Total price of the product(i.e manufacturing price)

4. Base price of the product (i.e retail price)

5. No.of.unit sold.

2. DATASET LOADING:

In dataset loading, we use the dataset that is mentioned in the data collection step.

To load the dataset, first we have to

1 .importing the required libraries :

First,we imported the numpy and pandas.

NUMPY- *It is the fundamental package for* ***scientific computing*** *with Python. Besides its obvious scientific uses, Numpy can also be used as an efficient multi-dimensional container of generic data.*

PANDAS - *Pandas is an open-source library that is made mainly for working with relational or labeled data both easily and intuitively. It provides various data structures and operations for manipulating numerical data and time series.*

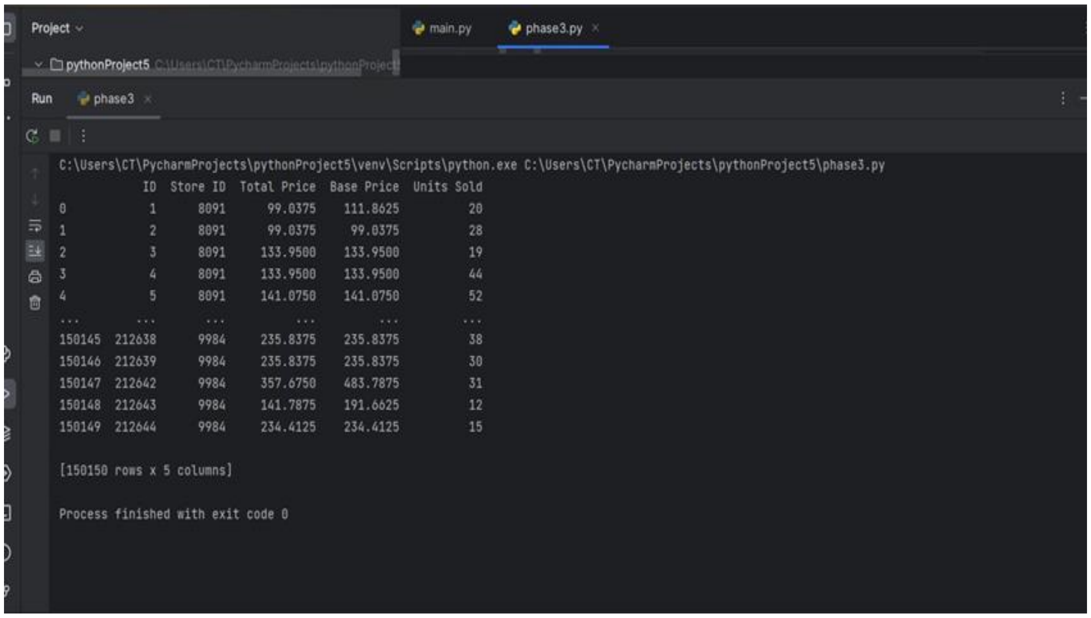
```
# importing required libraries

import pandas as pd

import numpy as np
```

2 .loading the dataset:

For loading the dataset,

```
# importing required libraries

import pandas as pd

import numpy as np


#loading the dataset

data=pd.read_csv(r"C:\Users\Administrator\PoductDemand.csv")

print(data)
```

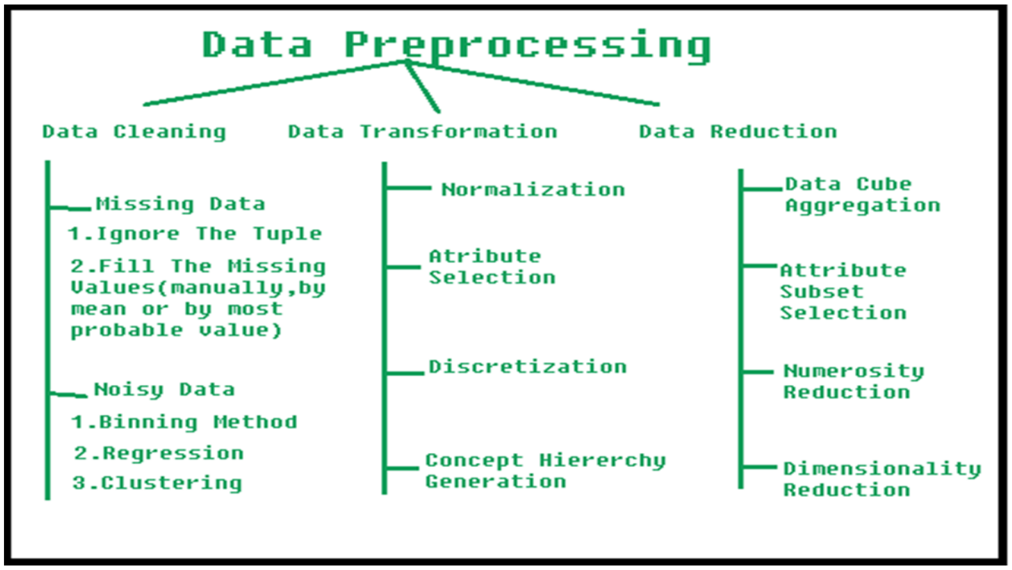
The output of the above code will be,



```
C:\Users\CT\PycharmProjects\pythonProject5\venv\Scripts\python.exe C:\Users\CT\PycharmProjects\pythonProject5\phase3.py
            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
...        ...       ...          ...         ...         ...
150145  212638      9984     235.8375    235.8375          38
150146  212639      9984     235.8375    235.8375          30
150147  212642      9984     357.6750    483.7875          31
150148  212643      9984     141.7875    191.6625          12
150149  212644      9984     234.4125    234.4125          15

[150150 rows x 5 columns]

Process finished with exit code 0
```

<u>***3. DATA PREPROCESSING:***</u>

Data *Cleaning uses methods to handle* ***incorrect, incomplete, inconsistent****, or* ***missing values.***



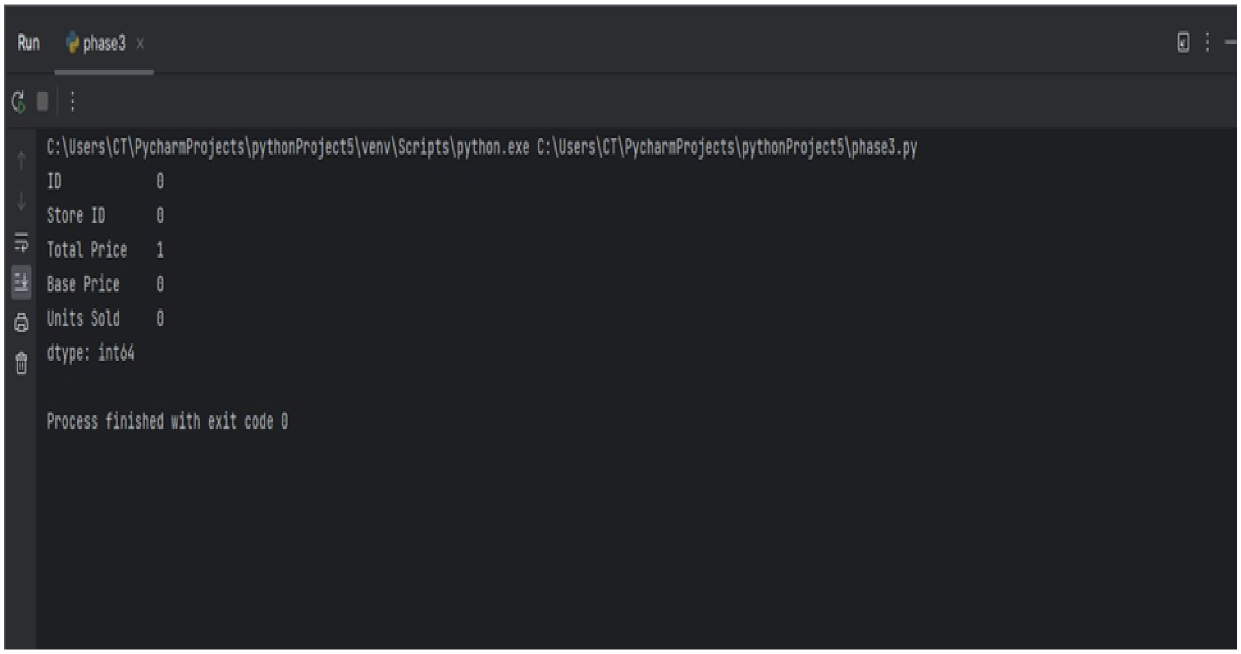
<u>**1 .handling missing values:**</u>

Count NaN values using isnull(): *It returns a Boolean same-sized object indicating if the values are NA. Missing values get mapped to True and non-missing value gets mapped to False.*

Calling the sum() method on the isnull() series returns the count of True values which actually corresponds to the number of NaN values.

```
# importing required libraries
import pandas as pd
import numpy as np

#loading the dataset
data=pd.read_csv(r"C:\Users\Administrator\PoductDemand.csv")
print(data)
#data preprocessing

#handling missing values
print(data.isnull().sum())
```

*The **output** of the above code will be,*



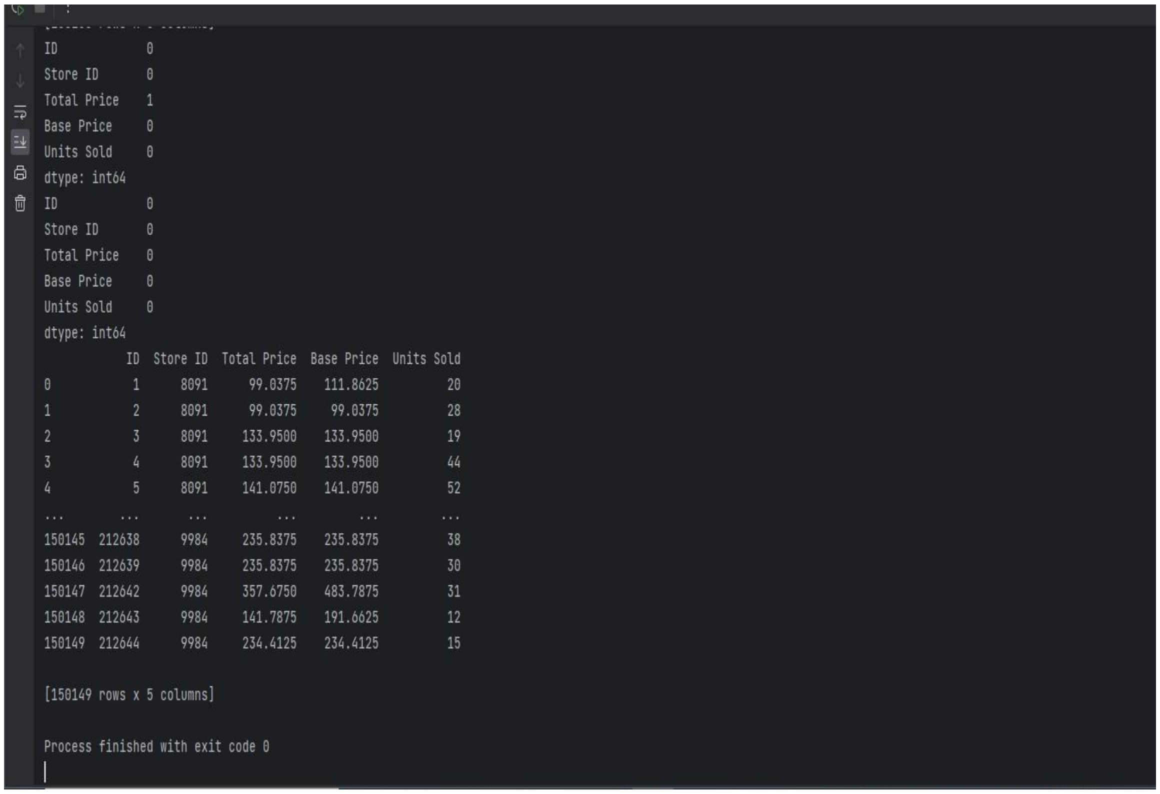
```
C:\Users\CT\PycharmProjects\pythonProject5\venv\Scripts\python.exe C:\Users\CT\PycharmProjects\pythonProject5\phase3.py
ID            0
Store ID      0
Total Price   1
Base Price    0
Units Sold    0
dtype: int64


Process finished with exit code 0
```

2 .removing missing values:

The first method is to remove all rows that contain missing values or, in extreme cases, entire columns that contain missing values. This can be performed by using df.dropna() function. axis=0 or axis=1 is used to delete rows/columns with NaN values.

```
# importing required libraries

import pandas as pd

import numpy as np



#loading the dataset

data=pd.read_csv(r"C:\Users\Administrator\PoductDemand.csv")

print(data)


#data preprocessing

#handling missing values
print(data.isnull().sum())

#removing missing values

data.dropna(inplace=True)
print(data.isnull().sum())
print(data)
```

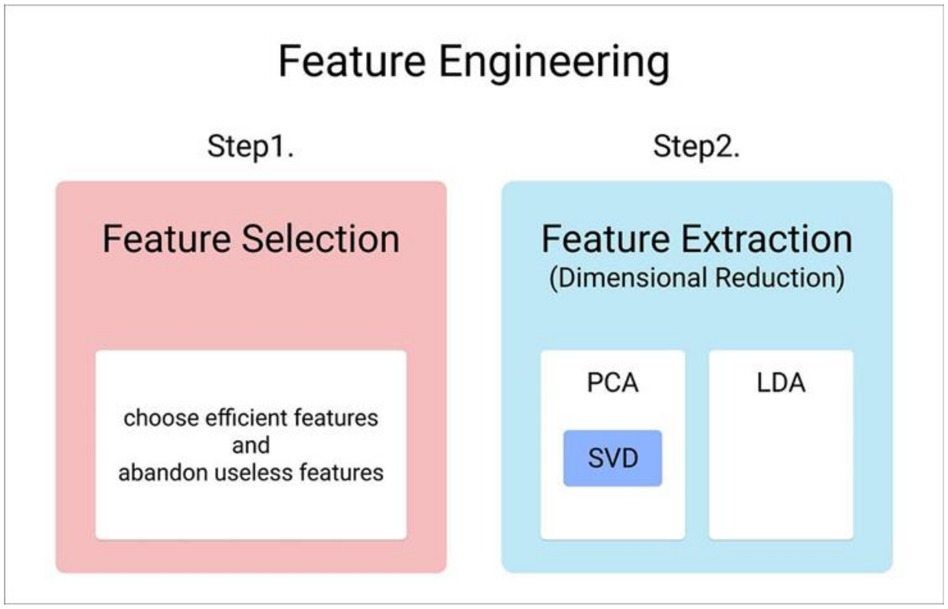
*The **output** of the above code will be,*



```
ID             0
Store ID       0
Total Price    1
Base Price     0
Units Sold     0
dtype: int64
ID             0
Store ID       0
Total Price    0
Base Price     0
Units Sold     0
dtype: int64
            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
...        ...       ...          ...         ...         ...
150145  212638      9984     235.8375    235.8375          38
150146  212639      9984     235.8375    235.8375          30
150147  212642      9984     357.6750    483.7875          31
150148  212643      9984     141.7875    191.6625          12
150149  212644      9984     234.4125    234.4125          15

[150149 rows x 5 columns]

Process finished with exit code 0
```

<u>4. FEATURE ENGINEERING:</u>

It is the process that takes raw data and transforms it into features that can be used to create a predictive model using machine learning or statistical modeling.



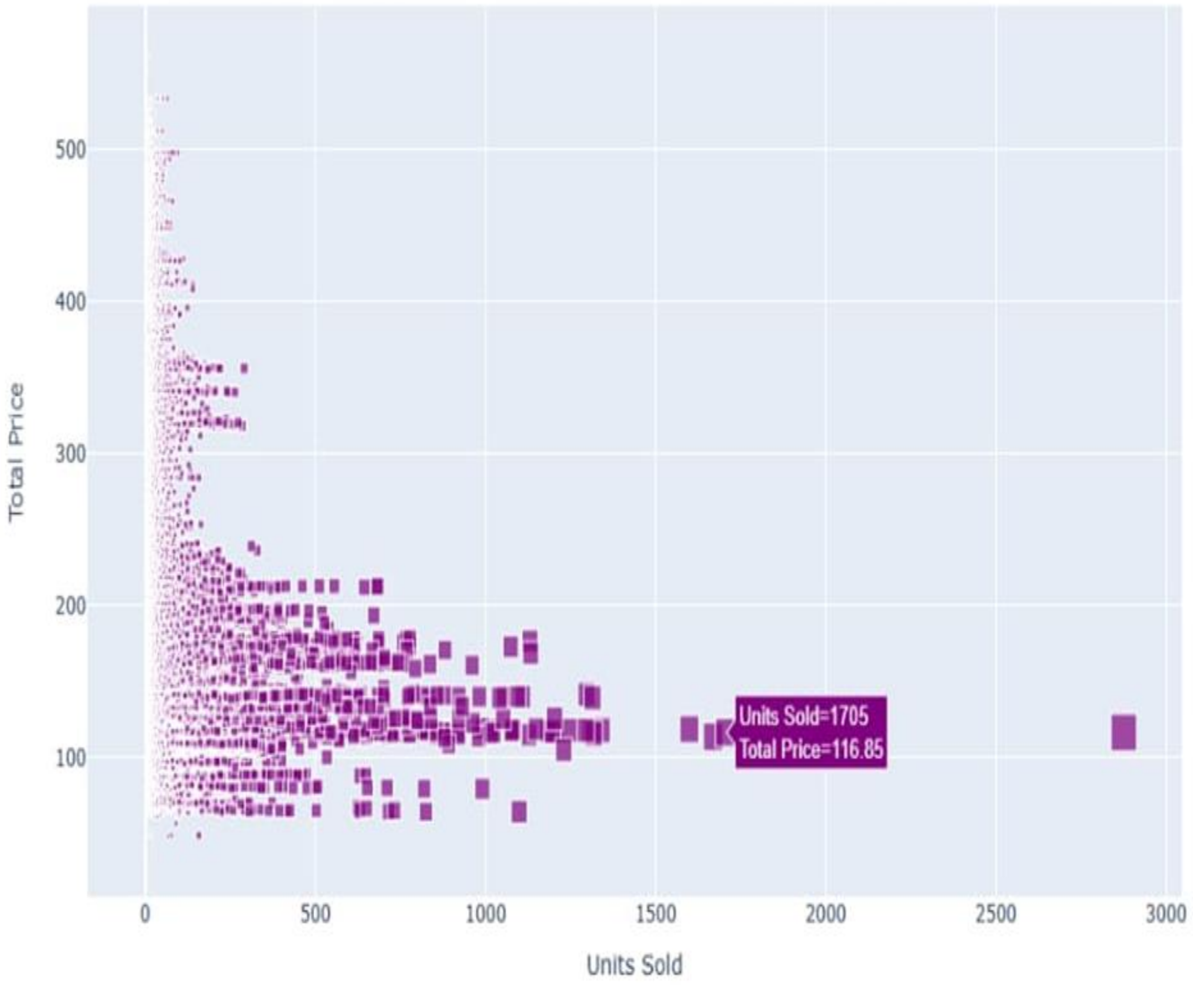
Here, we work with the efficient datas, and neglect the features that are not efficient and go ahead with datas that makes the model's output very accurate and perfect.

Our dataset, is already preprocessed which is out from error values, null values, and we cleaned the data.

So, we have to check the dataset has any outlier values.

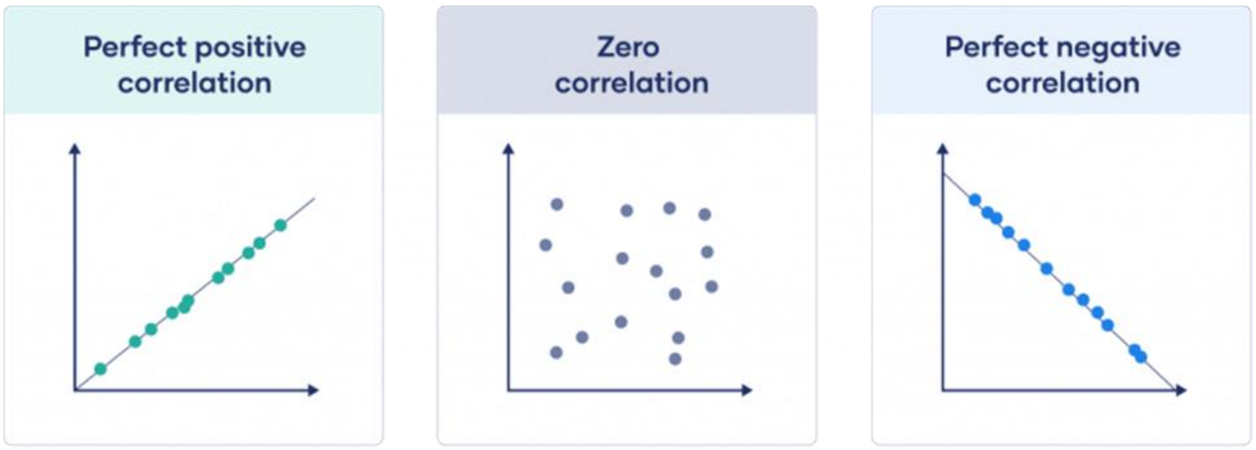
```
#FEATURE ENGINEERING

# Data visualization
symbol = ['square']
plot = px.scatter(data, x="Units Sold", y="Total Price",
                  size='Units Sold',symbol_sequence = symbol)
plot.update_traces(marker=dict(color='purple'))
plot.show()
```

The output of the above code will be,



CORRELATION BETWEEN THE DATA:

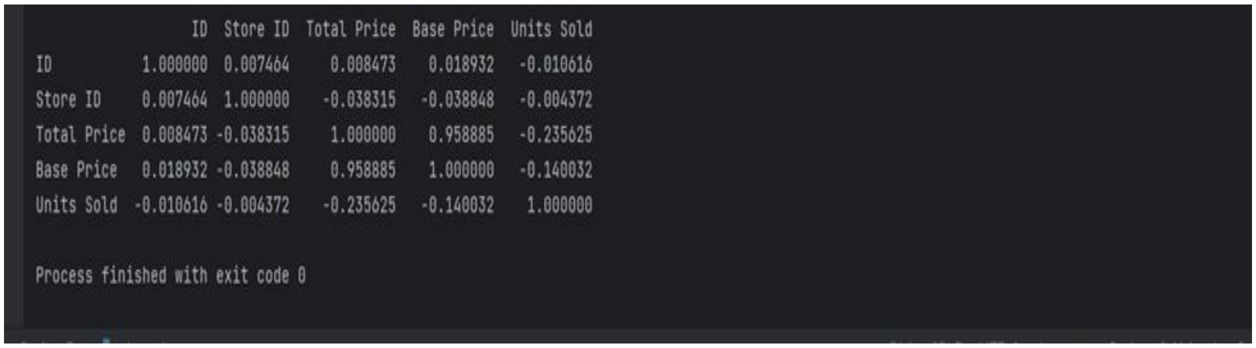
- *A correlation is a statistical measure of the relationship between two variables.*
- *The measure is best used in variables that demonstrate a linear relationship between each other.*
- *The fit of the data can be visually represented in a scatterplot.*
- *It values range from -1 to +1.*



Codes for correlation between datas are,

```
# Correlation betweeen the data
print(data.corr())
```

The output of the code is,



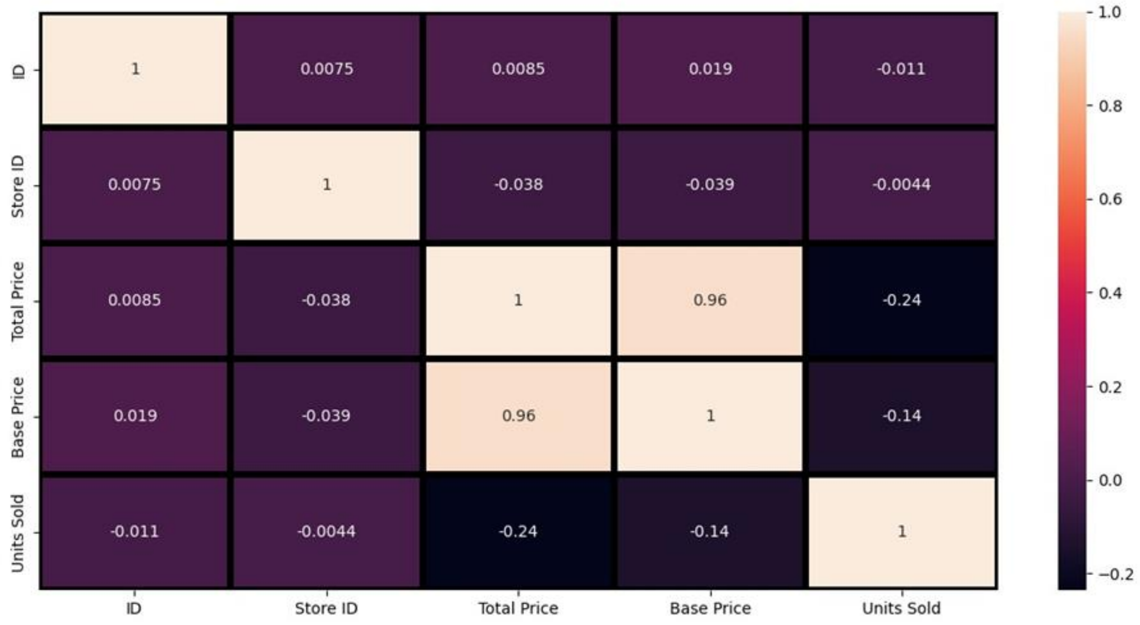
```
                ID  Store ID  Total Price  Base Price  Units Sold
ID          1.000000  0.007464     0.008473    0.018932   -0.010616
Store ID    0.007464  1.000000    -0.038315   -0.038848   -0.004372
Total Price 0.008473 -0.038315     1.000000    0.958885   -0.235625
Base Price  0.018932 -0.038848     0.958885    1.000000   -0.140032
Units Sold -0.010616 -0.004372    -0.235625   -0.140032    1.000000

Process finished with exit code 0
```

TO VISUALIZE THE CORRELATION:

Codes for visualizing the correlation between datas are,

```
#Visualizing correlation
correlations = data.corr(method='pearson')
plt.figure(figsize=(20, 20))
cmap1=cmap.Colormap(['red','green','purple'])
sns.heatmap(correlations, annot=True,linecolor='black',linewidths=3)
plt.show()
```

The output of the above code is,

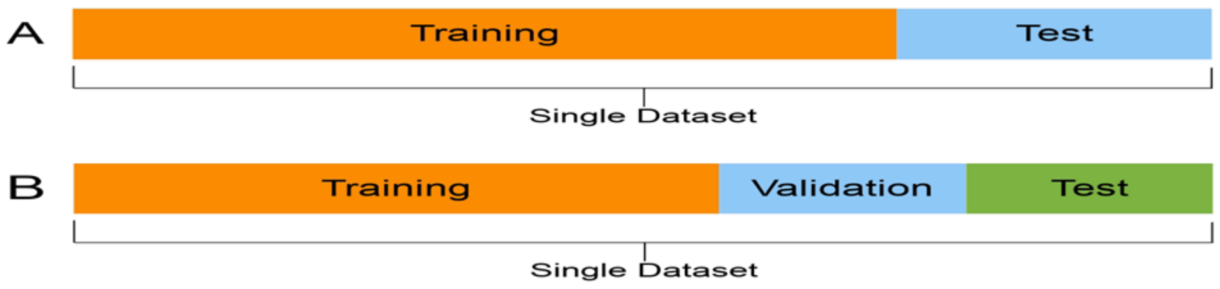


5 .MODEL SELECTION:

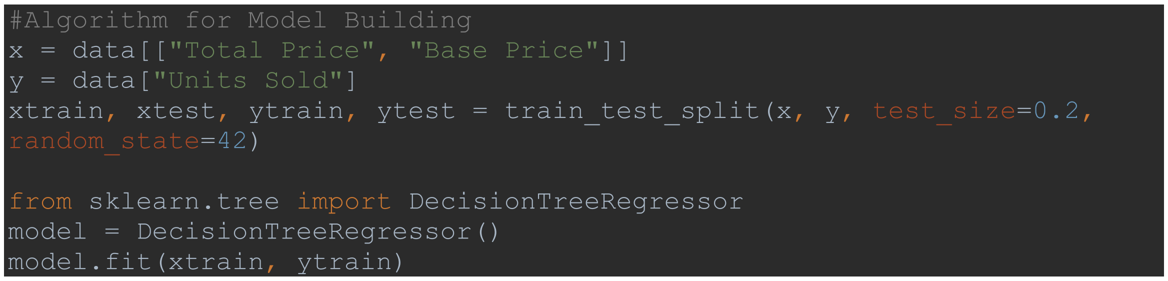
Already we know that we are using decision tree algorithm for the dataset to perform product demand prediction.

6 .MODEL TRAINING:

Model training aims to build the best mathematical representation of the relationship between data and a target (supervised) or among the data itself (unsupervised). 80% of train the dataset, 20% of testing.



Now let’s move to the task of training of 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:



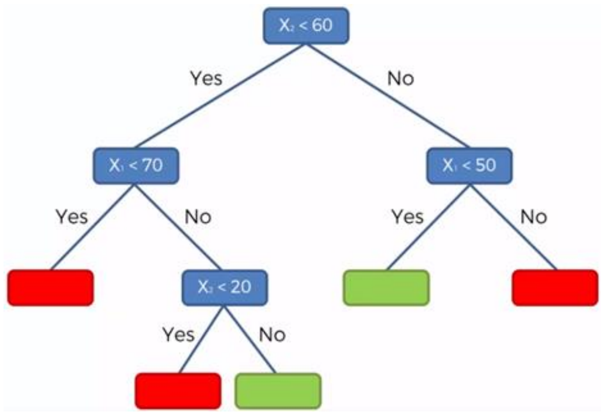
```
#Algorithm for Model Building
x = data[["Total Price", "Base Price"]]
y = data["Units Sold"]
xtrain, xtest, ytrain, ytest = train_test_split(x, y, test_size=0.2,
random_state=42)

from sklearn.tree import DecisionTreeRegressor
model = DecisionTreeRegressor()
model.fit(xtrain, ytrain)
```

And we have to choose the predictive model. Here, we choose the decision *tree regression algorithm to train our model*

DECISION TREE REGRESSOR:

- *Decision tree builds regression or classification models in the form of a tree structure.*
- *It breaks down a dataset into smaller and smaller subsets while at the same time an associated decision tree is incrementally developed.*
- *The final result is a tree with decision nodes and leaf nodes.*

- *The decision tree model can be used for both classification and regression problems,*
- *And it is easy to interpret, understand, and visualize.*

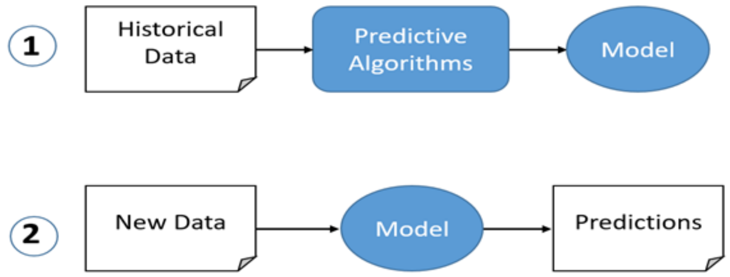


```
#Algorithm for Model Building
x = data[["Total Price", "Base Price"]]
y = data["Units Sold"]
xtrain, xtest, ytrain, ytest = train_test_split(x, y, test_size=0.2,
random_state=42)

from sklearn.tree import DecisionTreeRegressor
model = DecisionTreeRegressor()
model.fit(xtrain, ytrain)
```

Here, we trained our model successfully.

<u>*6.MODEL PREDICTION:*</u>

In model prediction, decision tree predicts the value of a target variable by learning simple decision rules inferred from the data features.

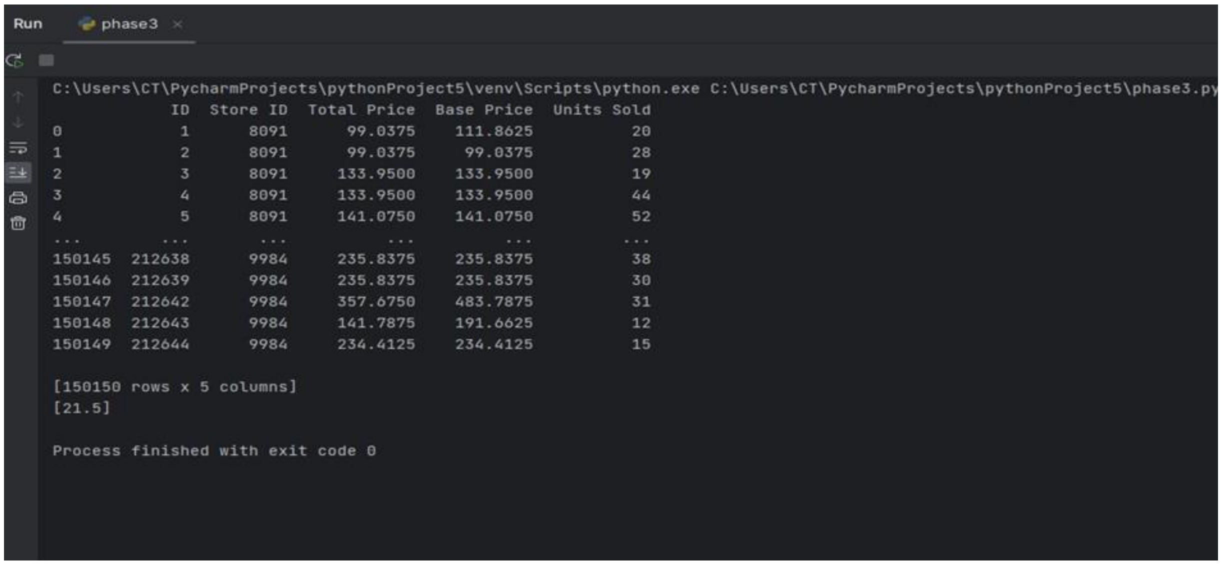


Now let's input the features ***(TOTAL PRICE, BASE PRICE)*** *into the model and predict how much quantity can be demanded based on those values.*

Codes *for model prediction:*

```
#PREDICTING THE DEMAND
features = np.array([[133.00, 140.00]])
prediction=model.predict(features)
print(prediction)
```

The output of the above code will be,



```
C:\Users\CT\PycharmProjects\pythonProject5\venv\Scripts\python.exe C:\Users\CT\PycharmProjects\pythonProject5\phase3.py
            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
...        ...       ...          ...         ...         ...
150145  212638      9984     235.8375    235.8375          38
150146  212639      9984     235.8375    235.8375          30
150147  212642      9984     357.6750    483.7875          31
150148  212643      9984     141.7875    191.6625          12
150149  212644      9984     234.4125    234.4125          15

[150150 rows x 5 columns]
[21.5]

Process finished with exit code 0
```

<u>CONCLUSION:</u>

- By using the decision tree algorithm, we train a machine learning model for 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 decreases.
- This is how, the demand of a product can vary based on the price of the product.

THANK YOU…