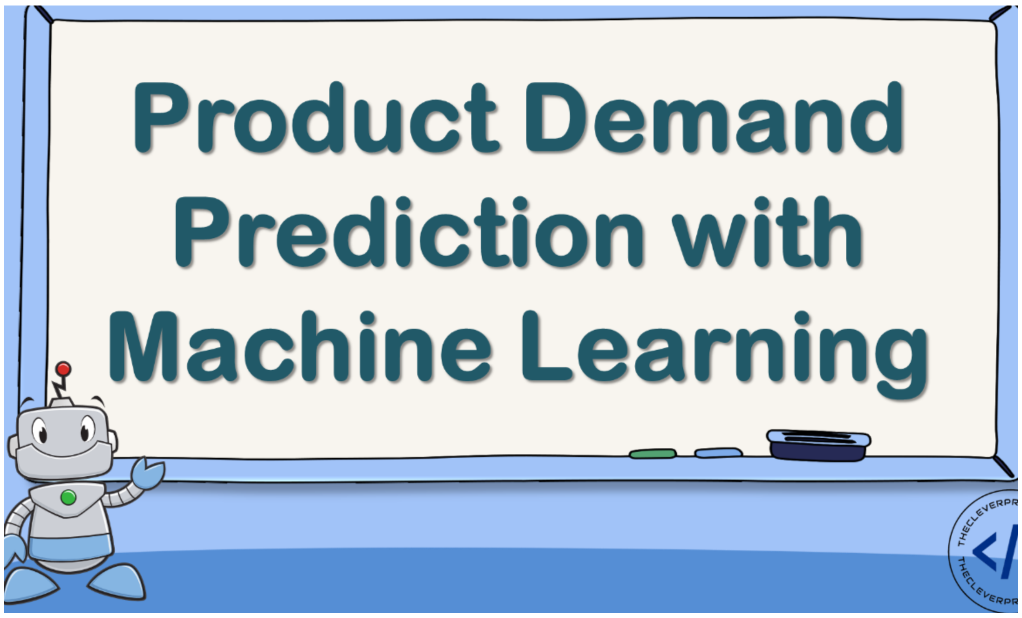
DOMAIN: APPLIED DATA SCIENCE

PROJECT: PRODUCT DEMAND PREDICTION WITH MACHINE LEARNINGS



TEAM MEMBERS:

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

PHASE 4 : DEVELOPMENT PART-2

In this phase, we have to continue developing the product demand model by feature engineering, model training, and evaluation.

It consists of 3 things. They are

1.feature engineering

2. model training

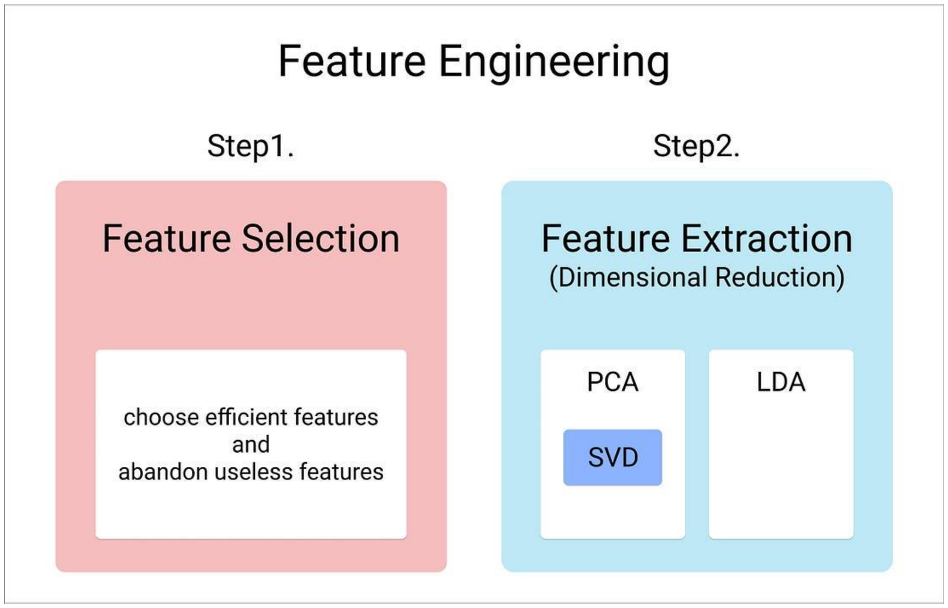
3.model evaluation.

FEATURE ENGINEERING- 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.

TRAINING- in this phase, we try to fit the best combination of weights and bias to a machine learning algorithm to minimize a loss function over the prediction range.

MODEL EVALUATION- is the process of using different evaluation metrics to understand a machine learning model's performance, as well as its strengths and weaknesses. Model evaluation is important to assess the efficacy of a model during initial research phases, and it also plays a role in model monitoring.

1. FEATURE ENGINEERING:


Feature Engineering
Step1.
Step2.
Feature Selection
choose efficient features and abandon useless features
Feature Extraction (Dimensional Reduction)
PCA
SVD
LDA

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.

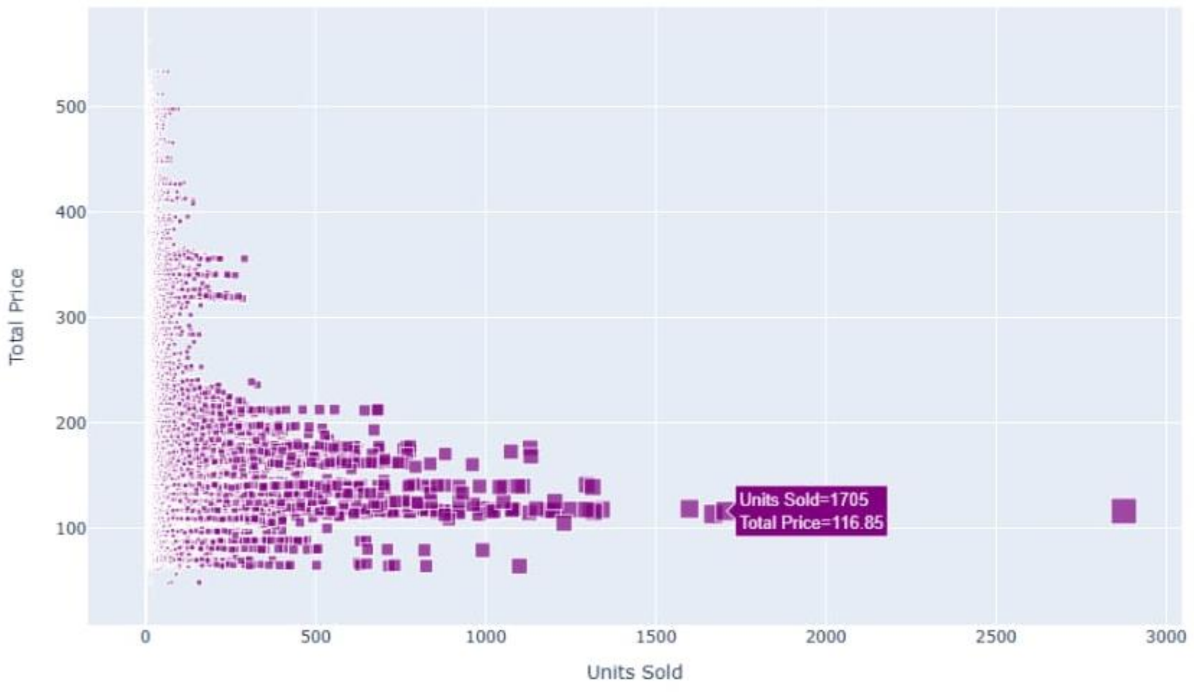
our dataset, is already preprocessed which is out from error values, null values, and we cleaned the data.

So, we have to check the dataset has any outlier values. Codes for visualizing the dataset are,

```
#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()
```

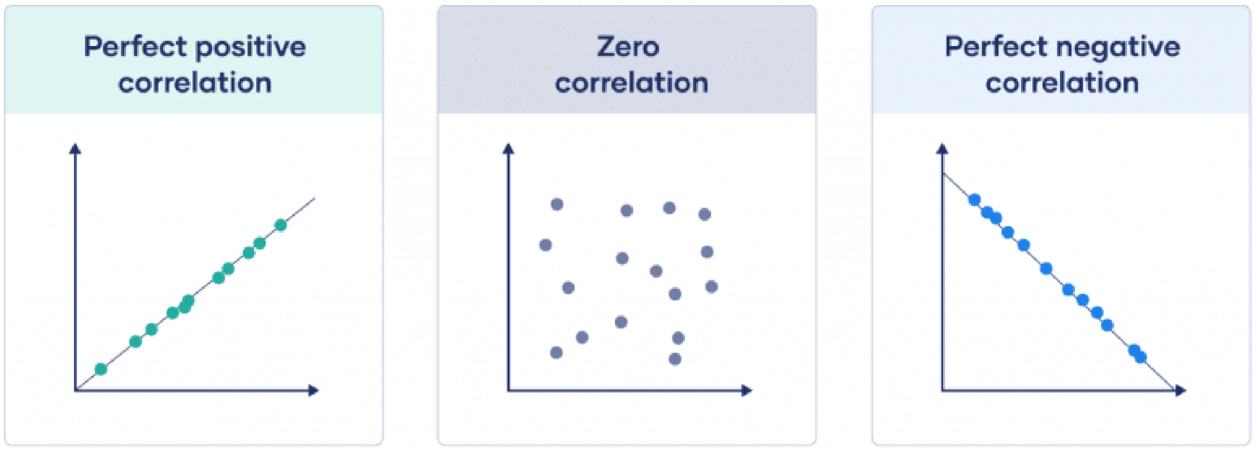
The output of the above code will be,



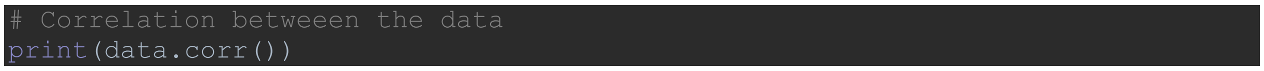
CORRELATION BETWEEN THE DATA:

- 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.

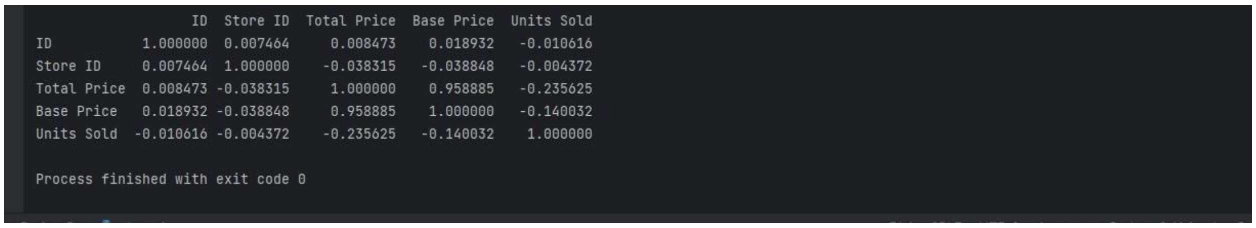


Codes for correlation between datas are,



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

The output of the above code will be,



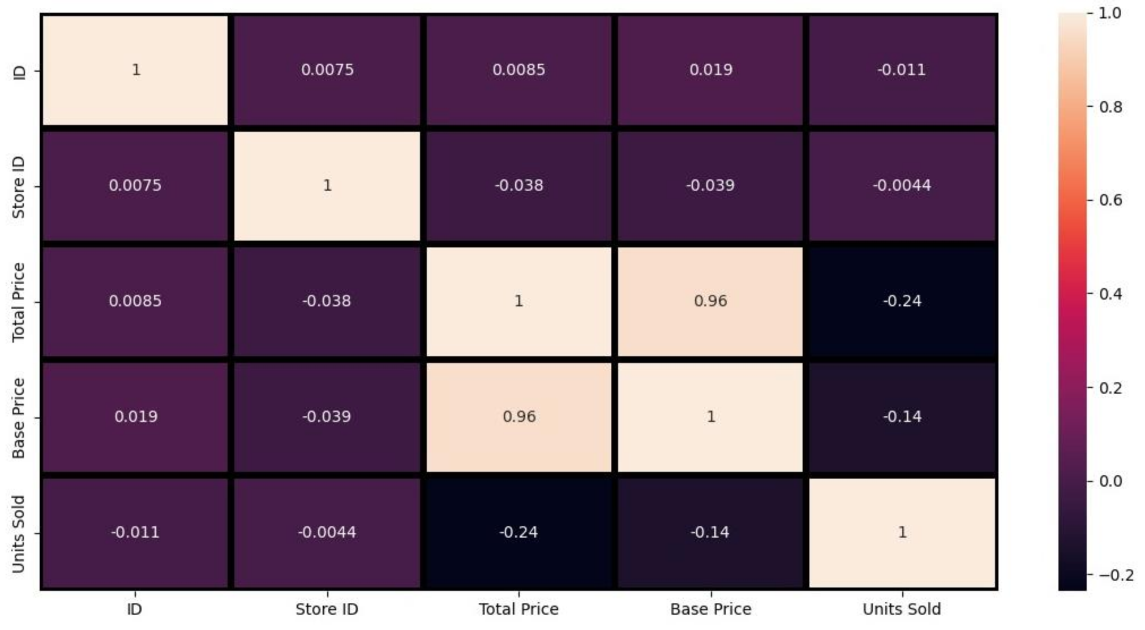
```
                  ID  Store ID  Total Price  Base Price  Units Sold
ID          1.000000  0.007464     0.008473    0.018932   -0.010616
Store ID    0.007464  1.000000    -0.038315   -0.038848   -0.004372
Total Price 0.008473 -0.038315     1.000000    0.958885   -0.235625
Base Price  0.018932 -0.038848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

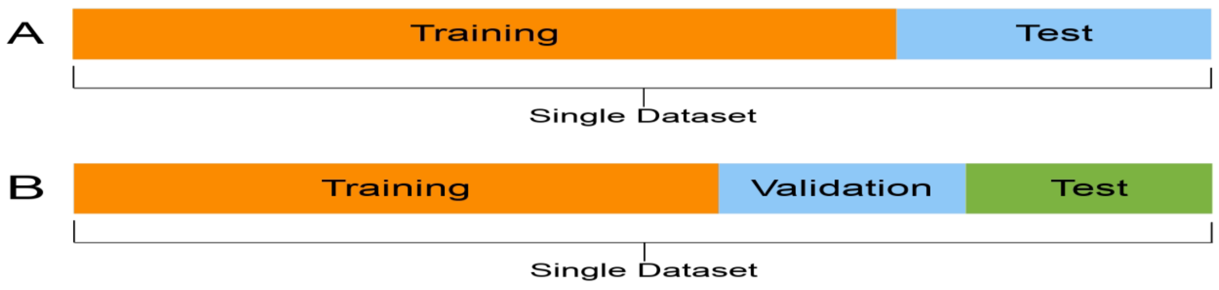
```
#Visualizing correlation
correlations = data.corr(method='pearson')
plt.figure(figsize=(20, 20))
cmap1=cmap.Colormap(['red','green','purple'])
sns.heatmap(correlations, annot=True,linecolor='black',linewidths=3)
plt.show()
```

The output of the above code will be,



2.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"]
```

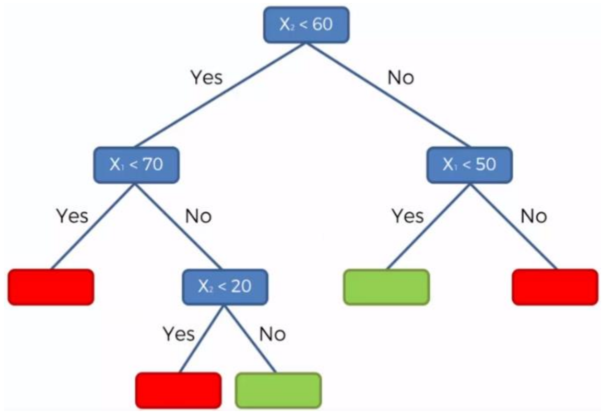
Now let's split the data into training and test sets :

```
#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)
```

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

DECISION TREE REGRESSION:

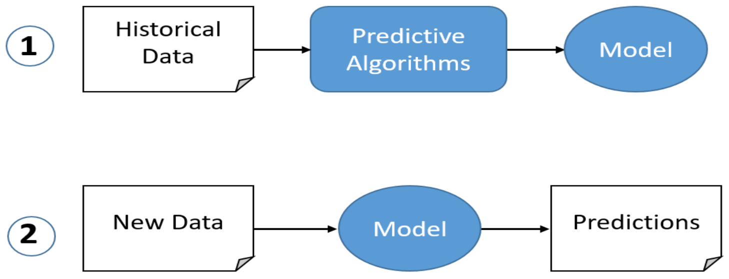
- Decision tree builds regression or classification models in the form of a tree structure.
- It breaks down a dataset into smaller and smaller subsets while at the same time an associated decision tree is incrementally developed.
- The final result is a tree with decision nodes and leaf nodes.
- The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

3. MODEL PREDICTION:

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

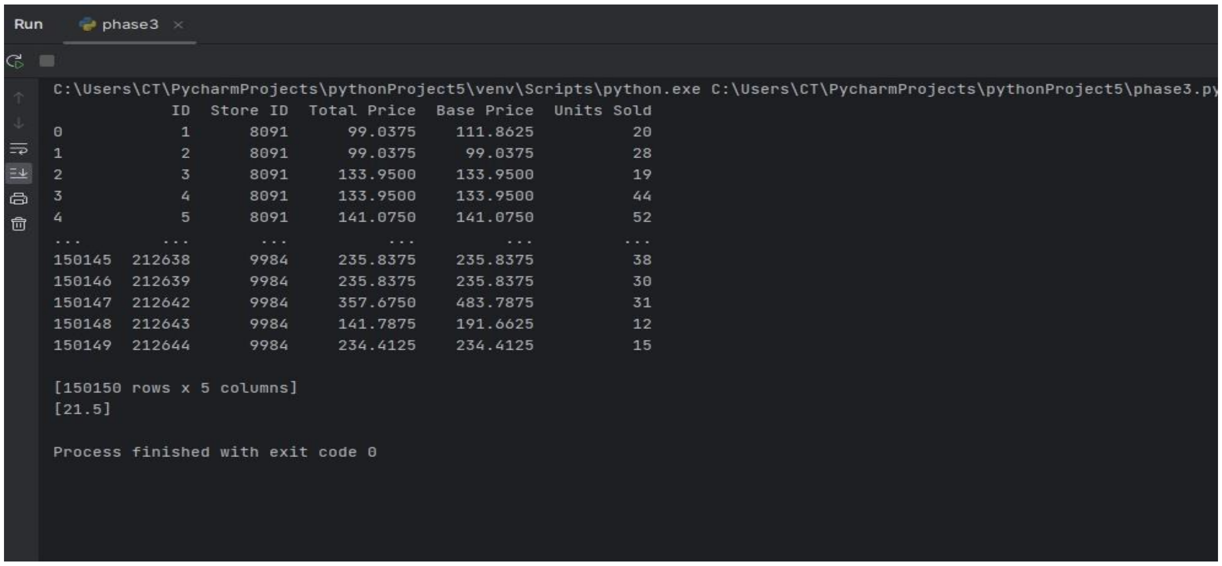


Now let's input the features (TOTAL PRICE, BASE PRICE) into the model and predict how much quantity can be demanded based on those values:

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
```

CONCLUSION:

- 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.