PHASE-4

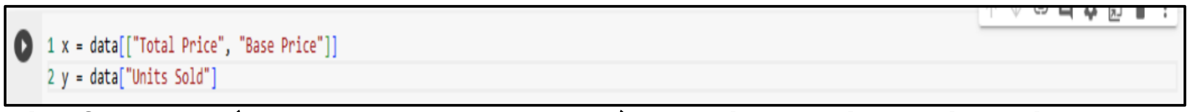
product demand prediction with machine learning

<u>product demand prediction model :</u>

Features extraction:

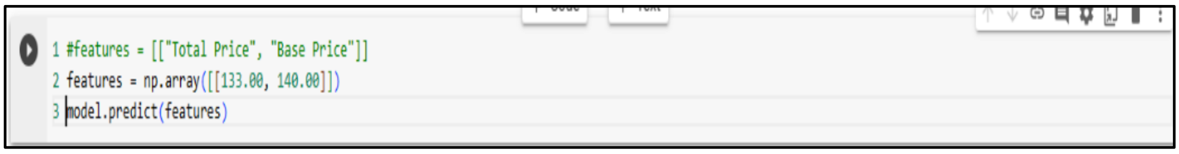
The process of transforming raw data into numerical features that can be processed while preserving the information in the original data set.

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:



```
x = data[["Total Price", "Base Price"]]
y = data["Units Sold"]
```

The features (Total Price, Base Price) into the model and predict how much quantity can be demanded based on those values:

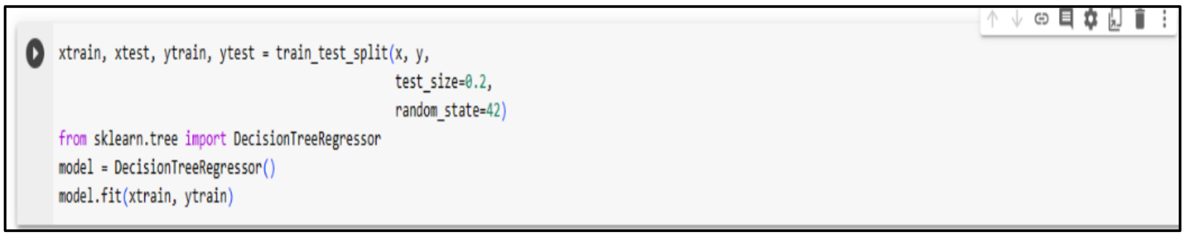


```
#features = [["Total Price", "Base Price"]]
features = np.array([[133.00, 140.00]])
model.predict(features)
```

<u>Traning and testing :</u>

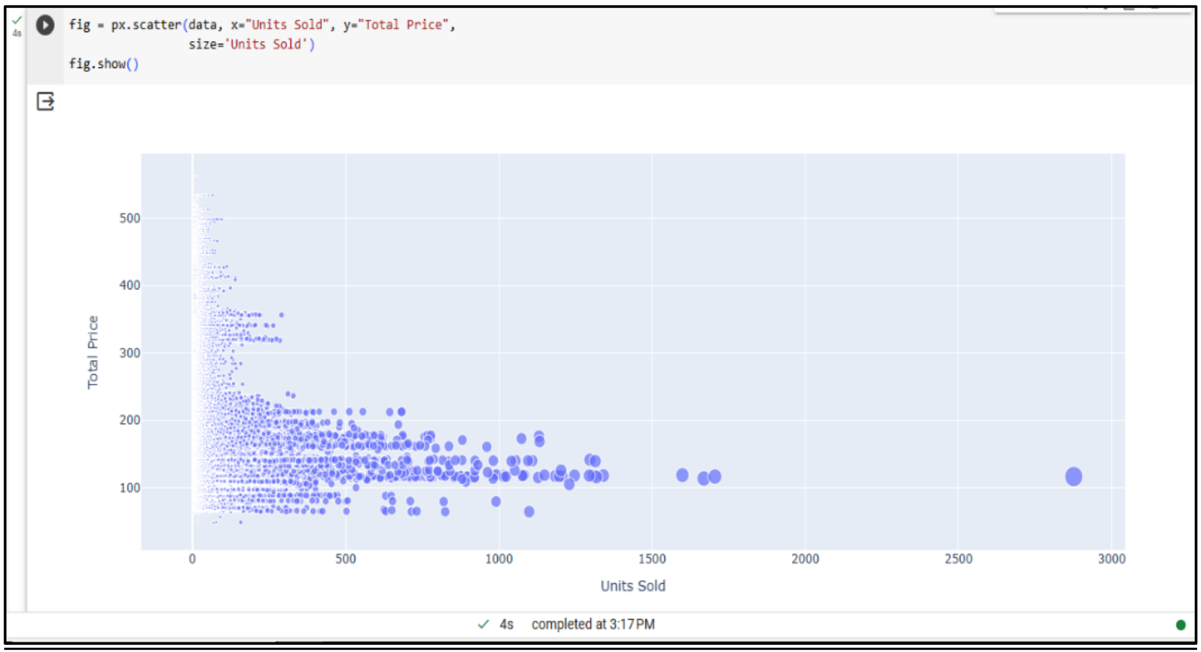
Therefore, train and test datasets are the two key concepts of machine learning, where the training dataset is used to fit the model, and the test dataset is used to evaluate the model.

Lets split the data into training and test sets and use the decision tree regression algorithm to train our model.

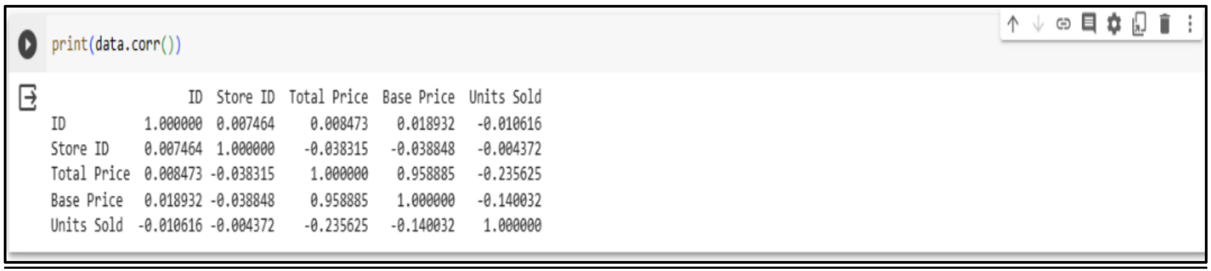


```
xtrain, xtest, ytrain, ytest = train_test_split(x, y,
                                                test_size=0.2,
                                                random_state=42)
from sklearn.tree import DecisionTreeRegressor
model = DecisionTreeRegressor()
model.fit(xtrain, ytrain)
```

<u>Evaluation :</u>

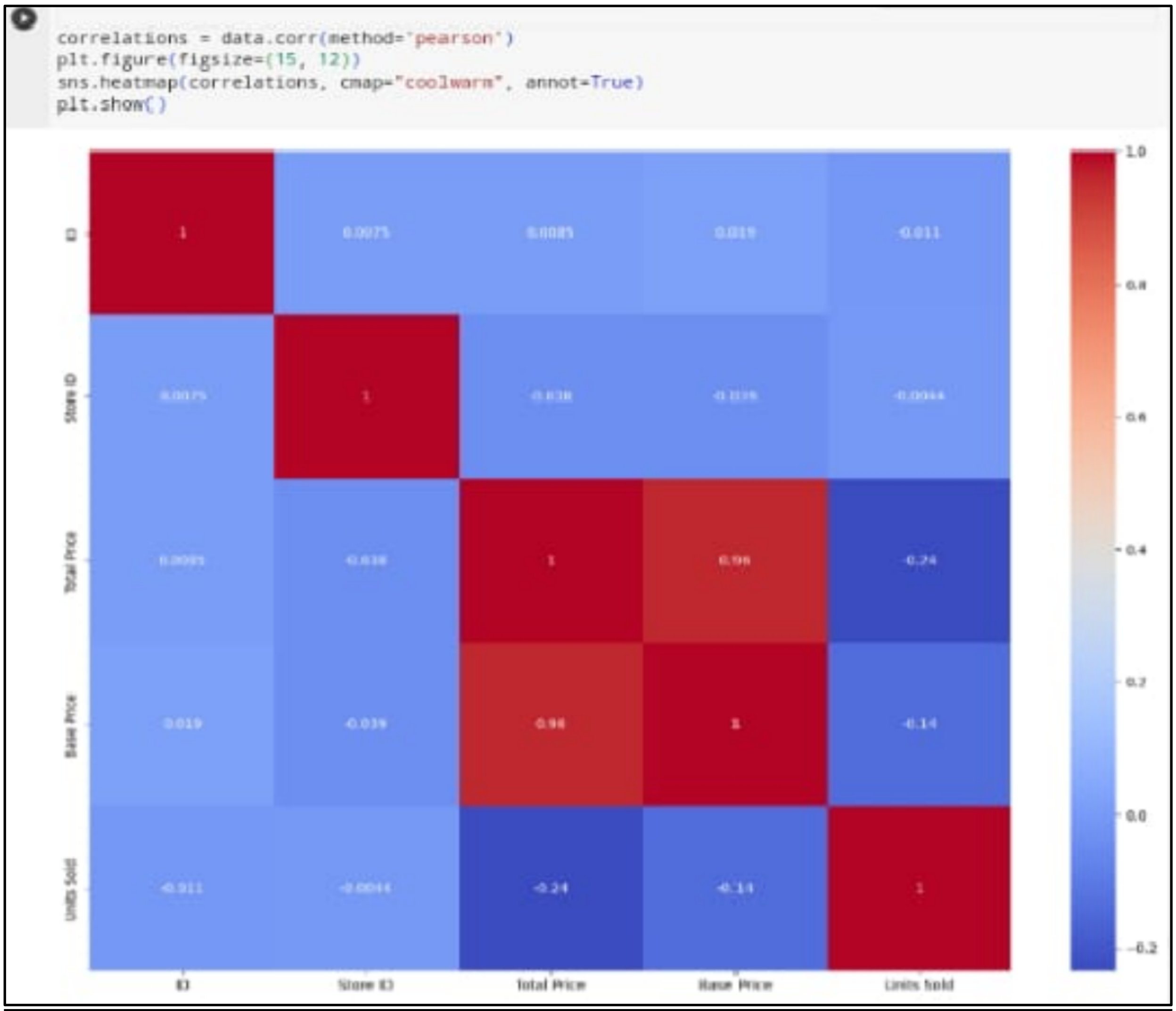


```
fig = px.scatter(data, x="Units Sold", y="Total Price",
                 size='Units Sold')
fig.show()
```



```
print(data.corr())
```

```
                  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
```



```
correlations = data.corr(method='pearson')
plt.figure(figsize=(15, 12))
sns.heatmap(correlations, cmap="coolwarm", annot=True)
plt.show()
```