**PRODUCT DEMAND PREDICTION WITH MACHINE LEARNING**

Submitted by:

R.Umashankar

au723921243051

umaraji6383@gmail.com

phase-4Developmentpart-2

## 

ProductDemandpredictionWithMachine Learning:

Predicting productdemandwith machinelearning isavaluableapplication in variousindustries. Toget started, here'sa high-level overviewofthestepsinvolved:

**Data Collection:**

Gather historical dataonproductsales,including factorsthat mayinfluencedemand,suchas pricing,promotions,seasonality, andexternal events.

Clean andpre-processthedata. Thisinvolveshandling missing values, outliers,andencoding categorical variables.

FeatureSelection/Engineering : Identifyrelevantfeaturesthat canaffectproductdemand.You may needtocreatenewfeaturesortransformexisting ones.

DataSplitting : Splitthedata intotraining,validation, andtest sets toevaluatethemodel's performance.

Model Selection: Choosetheappropriatemachine learning model. Common choicesinclude linearregression, decisiontrees, randomforests, or moreadvancedtechniques likeneural networks.Model Training: Traintheselectedmodel using thetrainingdata.

Model Evaluation :Evaluatethemodel's performanceusing thevalidation set. Common evaluationmetrics includeMean AbsoluteError(MAE), Mean Squared Error (MSE), and RootMean Squared Error (RMSE).

Hyper-parameterTuning : Fine-tunethemodel'shyper-parameters tooptimizeitsperformance.

Model Validation : Assess themodel's generalization performance onthetest set toensureit can makeaccurate predictions onunseen data.

Deployment : Oncesatisfiedwith themodel'sperformance,deploy it inareal-worldenvironment tomakepredictions on futuredemand.

## DataLoading ancPre-processing**:**

DataSources : Identify the sourcesofyour data,whether it'sstoredindatabases, spreadsheets, text files, or obtainedfromAPIs.Ensureyouhaveaccess tothe datayouneed.

DataRetrieval : Useappropriate libraries or toolstoloadyour dataintoyour analysis environment. For example,inPython, you canuselibrarieslikePandastoreaddatafromvarious fileformats or SQLdatabases.

DataInspection: Onceyou'veloadedthedata,inspectthefirst fewrows togetaninitial understanding ofitsstructureandcontents. Thisstep helpsyouverify thatthedatawas loaded correctly.

Handling Missing Values : Identify andhandlemissing data. You canchoose toremoverowswith missing values, fill themwith suitablevalues(e.g., mean,median),orusemore advanced imputationtechniques.

Dealing with Duplicates: Checkfor andremoveduplicaterecords iftheyexistinthedataset.

Data Encoding : Convert categorical variablesintonumerical formatusing techniques likeone-hot encoding or label encoding. Thisis necessary formostmachinelearning algorithms thatrequire numerical inputs.

## Visualization:

Visualization isapowerful tool for understanding andcommunicating theresults ofproduct demandprediction.Herearesomecommon typesofvisualizations you canuse:

TimeSeries Plots :Ifyourpredictioninvolvestime-dependentdata,createtimeseriesplots to visualizehistorical demandandpredicteddemandover time. You can use linecharts toshowthe actualdemandandforecasteddemandonthesamegraph.

Actual vs.Predicted Plots : Compareactual productdemandwith your model'spredictions. Scatterplots or linechartscanhelp you assess howcloselyyourmodel's predictions align withthe real data.

Residual Plots : Plot theresiduals(thedifferences betweenactual andpredictedvalues) overtime. Thiscanhelp you identify patternsortrendsinpredictionerrors.

Histograms and Density Plots : Visualizethedistributionof prediction errors toassess their normalityandidentify potential bias or skewin thepredictions.

BoxPlots : Useboxplots tovisualizethespreadanddistributionof errors, includingoutliers. This canhelp youidentify areas wherethemodel performs exceptionallywell orpoorly

## EvaluationPerformance:

Evaluation Metrics : Thesearequantitativemeasuresusedtoassesshowwell amodel or algorithmperforms on aparticulartask.Commonevaluationmetrics includeMeanAbsoluteError (MAE), Mean Squared Error(MSE), RootMeanSquared Error (RMSE), accuracy,precision, recall, F1 score, andmany others.Thechoiceof metricdepends on the problemyou aretryingtosolve.

For example,classificationtasks often usemetrics likeaccuracy, whileregressiontasks use metrics likeRMSE.

Trainingand Testing : Inmachine learning,youtypically splityour datasetintoa trainingsetanda testing set.Themodel istrainedonthetraining set andthenevaluatedon the testing settoassess howwell itgeneralizestonew,unseen data.

## LoadingandImportingthelibraries

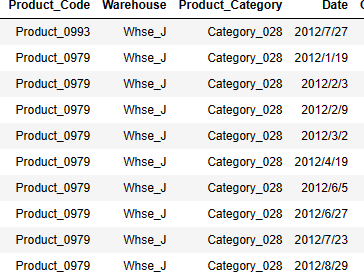
import numpyas np import pandasaspd

import matplotlib.pyplot asplt import seabornas sns

data=pd.read\_csv("Historical ProductDemand.csv") data



data.head(10)



data.tail(10)



data.info()



**<class 'pandas.core.frame.DataFrame'> RangeIndex: 1048575 entries, 0 to 1048574 Data columns (total 5 columns):**

**# Column**

**- - -**

**0**

**1**

**2**

**3**

**Product\_Code**

**Non-Null Count**

**- - -- 1048575 non-null**

**Warehouse**

**1048575 non-null**

**Product\_Category 1048575 non-null Date**

**4 Order\_Demand**

**1037336 non-null**

**1048575 non-null**

**Dtype**

**- object object object object object**

**dtypes: object(5) memory usage: 40.0+ MB**

data.isna().sum()

**Product\_Code 0**

**Warehouse** **0**

**Product\_Category 0**

**Date 11239**

**Order\_Demand** **0**

**dtype: int64**

**data.dropna(axis = 0, inplace = True) data.isna().sum()**



**Product\_Code Warehouse Product\_Category Date Order\_Demand dtype: int64**

**0**

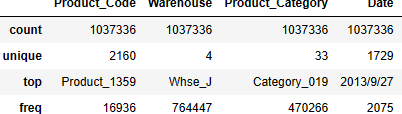
**0**

**0**

**0**

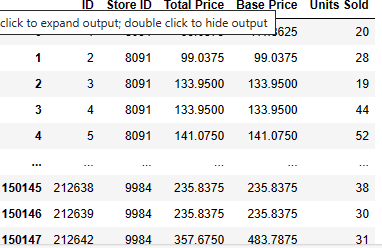
**0**

**data.describe(include= 'all')**

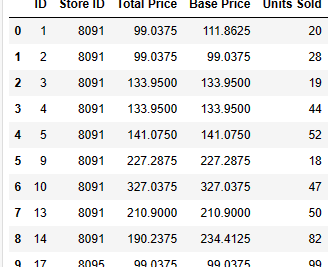


## LoadingTheData:

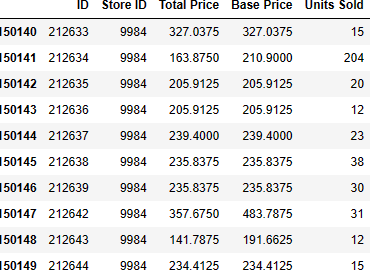
**data1 = pd.read\_csv("PoductDemand.csv") data1**



**data1.head(10)**



**data1.tail(10)**



# Future Engineering and pre-proc essing visualization:

**data1.isna().sum()**



**ID**

**Store ID Total Price Base Price Units Sold dtype: int64**

**0**

**0**

**1**

**0**

**0**

**data1.dropna(axis = 0, inplace = True) data1.isna().sum()**



**ID**

**Store ID Total Price Base Price Units Sold dtype: int64**

**0**

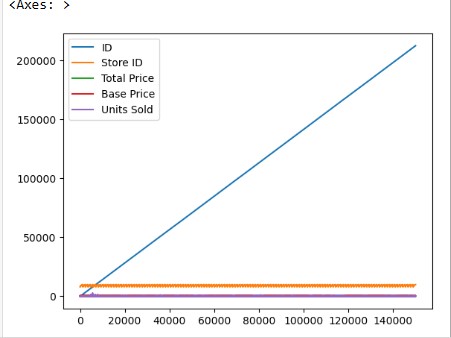
**0**

**0**

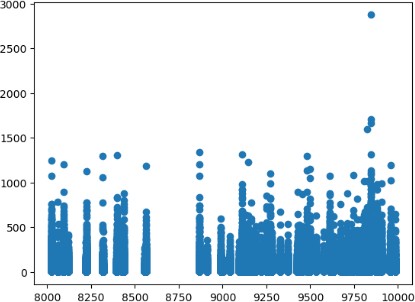
**0**

**0**

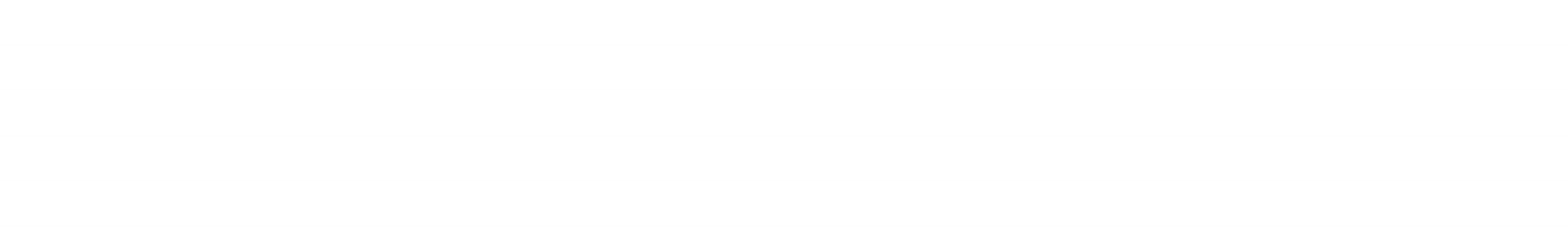
**data1.plot()**



**plt.scatter(data1["Store ID"], data1["Units Sold"]) plt.show()**



**data1.mean()**



**ID** **106270.971795**

**Store ID** **9199.420935**

**Total Price** **206.626751**

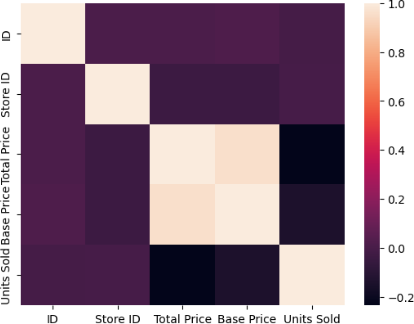
**Base Price** **219.424262**

**Units Sold** **51.674543**

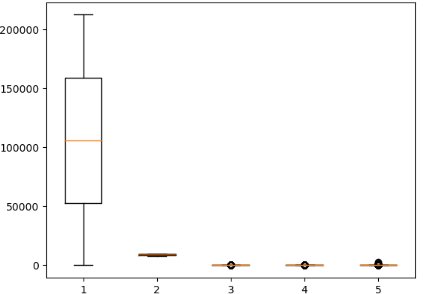
**dtype: float64**

**data1.corr()**

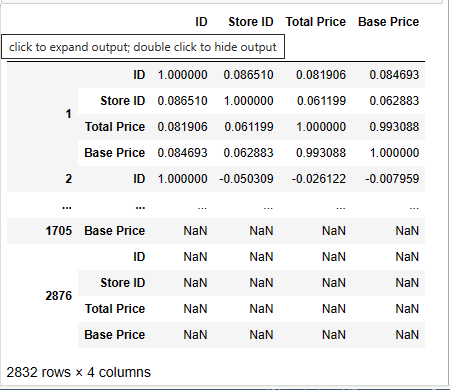


**sns.heatmap(data1.corr()) plt.show()**

**plt.boxplot(data1)**



**data1.groupby('Units Sold').corr()**



**data1.mean()**



**ID** **106270.971795**

**Store ID** **9199.420935**

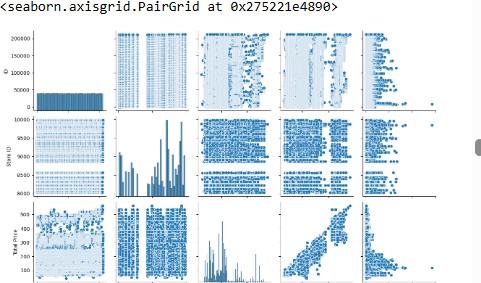
**Total Price** **206.626751**

**Base Price** **219.424262**

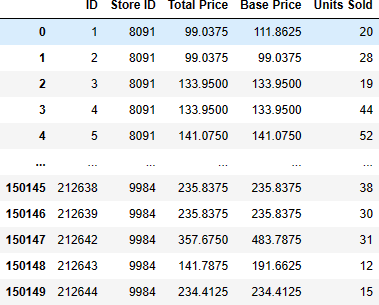
**Units Sold** **51.674543**

**dtype: float64**

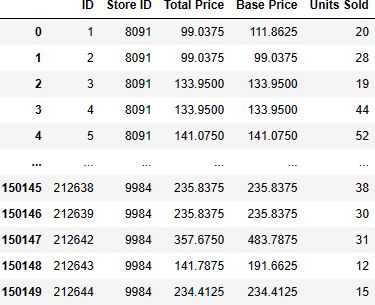
**sns.pairplot(data1)**



**data1.dropna()**



**data1.fillna(0)**



# Linear Regression:

**from sklearn.linear\_model import LinearRegression lr = LinearRegression()**

**model = lr.fit(x\_train, y\_train)**

**model.intercept\_**



**66.30203634254558**

**model.coef\_**



**array([**

**-4.39017685,**

**0.95330134, -342.45636915,**

**268.59705875])**

**y\_pred = model.predict(x\_test)**

|  |  |  |  |
| --- | --- | --- | --- |
| **from** | **sklearn.metrics** | **import** | **mean\_squared\_error** |
| **from** | **sklearn.metrics** | **import** | **mean\_absolute\_error** |
| **from** | **sklearn.metrics** | **import** | **r2\_score** |

**mean\_absolute\_error(y\_test, y\_pred)**



**33.233948162416986**

**mean\_squared\_error(y\_test, y\_pred)**



**2935.7936428617845**

**r2\_score(y\_test, y\_pred)**



**0.1607003705784842**

**y\_pred.mean()**

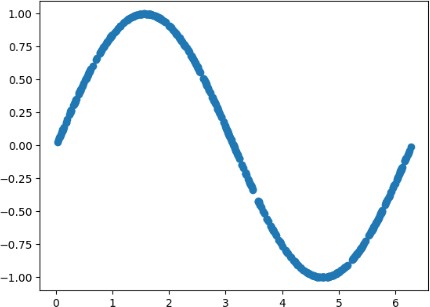
**51.573538982561075**

**def generateX(N):**

**x = np.random.random(N)\*2\*np.pi yd = np.sin(x)**

**return x, yd**

**x, y = generateX(500) plt.scatter(x, y) plt.show()**



**def plotmodel(x, y, yd): i = x.argsort() plt.figure()**

**plt.plot(x[i], y[i], "g-o")**

**plt.plot(x[i], yd[i], "r-o") plt.ylabel("F(X)")**

**plt.xlabel("X")**

**plt.legend(["estimated", "True"]) plt.title("Comparision") plt.show()**

**plotmodel(x, y\_pred, y)**

