**Topic:**

Building the product demand prediction model by feature engineering, model training and evaluation.

**Program:**

**# Data selection & loading**

import pandas

df=pandas.read\_csv("PoductDemand.csv")

print(df.tail())

print(df.head(8))

print(df.info())

print(df.describe())

**# Data cleaning & handling missing values**

print(df.drop\_duplicates())

print(df.dropna())

print(df.isna().any())

print(df.isna().sum())

df.fillna(df.mean(), inplace=True)

cf=df.copy()

**# Feature selection**

x=df[["Total Price","Base Price"]]

y=df["Units Sold"]

**# Data splitting**

from sklearn.model\_selection import train\_test\_split

x\_train,x\_test,y\_train,y\_test=train\_test\_split(x,y,test\_size=0.25)

**# Feature scaling**

from sklearn.preprocessing import StandardScaler

scaler = StandardScaler()

x\_train\_scaled = scaler.fit\_transform(x\_train)

x\_test\_scaled = scaler.transform(x\_test)

**# Model training**

print("Random Forest Algorithm")

from sklearn.ensemble import RandomForestRegressor

model = RandomForestRegressor(n\_estimators=100, random\_state=42)

model.fit(x\_train, y\_train)

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

**# Evaluation**

from sklearn.metrics import mean\_absolute\_error, mean\_squared\_error, r2\_score

import math

mae= mean\_absolute\_error(y\_test, y\_pred)

mse= mean\_squared\_error(y\_test, y\_pred)

rmse= math.sqrt(mse)

r\_squared= r2\_score(y\_test, y\_pred)

print(f'Mean Absolute Error: {mae}')

print(f'Mean Squared Error: {mse}')

print(f'Root Mean Squared Error: {rmse}')

print(f'R-squared: {r\_squared}')

**Output:**

**Data selection & loading:**

ID Store ID Total Price Base Price Units Sold

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

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

5 9 8091 227.2875 227.2875 18

6 10 8091 327.0375 327.0375 47

7 13 8091 210.9000 210.9000 50

<class 'pandas.core.frame.DataFrame'>

RangeIndex: 150150 entries, 0 to 150149

Data columns (total 5 columns):

# Column Non-Null Count Dtype

--- ------ -------------- -----

0 ID 150150 non-null int64

1 Store ID 150150 non-null int64

2 Total Price 150149 non-null float64

3 Base Price 150150 non-null float64

4 Units Sold 150150 non-null int64

dtypes: float64(2), int64(3)

memory usage: 5.7 MB

None

ID Store ID ... Base Price Units Sold

count 150150.000000 150150.000000 ... 150150.000000 150150.000000

mean 106271.555504 9199.422511 ... 219.425927 51.674206

std 61386.037861 615.591445 ... 110.961712 60.207904

min 1.000000 8023.000000 ... 61.275000 1.000000

25% 53111.250000 8562.000000 ... 133.237500 20.000000

50% 106226.500000 9371.000000 ... 205.912500 35.000000

75% 159452.750000 9731.000000 ... 234.412500 62.000000

max 212644.000000 9984.000000 ... 562.162500 2876.000000

[8 rows x 5 columns]

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]

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]

ID False

Store ID False

Total Price True

Base Price False

Units Sold False

dtype: bool

ID 0

Store ID 0

Total Price 1

Base Price 0

Units Sold 0

dtype: int64

**Random Forest Algorithm:**

Mean Absolute Error: 25.116842276084803

Mean Squared Error: 1893.5624978996275

Root Mean Squared Error: 43.51508356765074

R-squared: 0.433494702840748