Part-1

November 30, 2022

```
[1]: # importing required libraries
     import pandas as pd
     import seaborn as sns
     import matplotlib.pyplot as plt
     import numpy as np
     import sklearn
```

```
[2]: #loading dataset
     df = pd.read_excel("DS-Assignment_Part_1_data_set.xlsx")
```

```
[3]: # looking at first 5 rows of the dataset
     df.head()
```

```
[3]:    Transaction date  House Age  Distance from nearest Metro station (km)  \
     0       2012.916667       32.0                                  84.87882
     1       2012.916667       19.5                                 306.59470
     2       2013.583333       13.3                                 561.98450
     3       2013.500000       13.3                                 561.98450
     4       2012.833333        5.0                                 390.56840

        Number of convenience stores  latitude  longitude  Number of bedrooms  \
     0                            10  24.98298  121.54024                   1
     1                             9  24.98034  121.53951                   2
     2                             5  24.98746  121.54391                   3
     3                             5  24.98746  121.54391                   2
     4                             5  24.97937  121.54245                   1

        House size (sqft)  House price of unit area
     0                575                      37.9
     1               1240                      42.2
     2               1060                      47.3
     3                875                      54.8
     4                491                      43.1
```

```
[4]: # checking for null values
     df.isna().sum()
```

```
[4]: Transaction date                            0
     House Age                                   0
     Distance from nearest Metro station (km)    0
     Number of convenience stores                0
     latitude                                    0
     longitude                                   0
     Number of bedrooms                          0
     House size (sqft)                           0
     House price of unit area                    0
     dtype: int64
```

```
[5]: df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 414 entries, 0 to 413
Data columns (total 9 columns):
 #   Column                                    Non-Null Count  Dtype
---  ------                                    --------------  -----
 0   Transaction date                          414 non-null    float64
 1   House Age                                 414 non-null    float64
 2   Distance from nearest Metro station (km)  414 non-null    float64
 3   Number of convenience stores              414 non-null    int64
 4   latitude                                  414 non-null    float64
 5   longitude                                 414 non-null    float64
 6   Number of bedrooms                        414 non-null    int64
 7   House size (sqft)                         414 non-null    int64
 8   House price of unit area                  414 non-null    float64
dtypes: float64(6), int64(3)
memory usage: 29.2 KB
```

```
[6]: #finding correlation between columns of the dataset
     df.corr()
```

```
[6]:                                           Transaction date  House Age  \
     Transaction date                                  1.000000   0.017542
     House Age                                         0.017542   1.000000
     Distance from nearest Metro station (km)          0.060880   0.025622
     Number of convenience stores                      0.009544   0.049593
     latitude                                          0.035016   0.054420
     longitude                                        -0.041065  -0.048520
     Number of bedrooms                                0.061985  -0.008756
     House size (sqft)                                 0.068405  -0.060361
     House price of unit area                          0.087529  -0.210567

                                                Distance from nearest Metro station
     (km)  \
     Transaction date
     0.060880
```

```
House Age
0.025622
Distance from nearest Metro station (km)
1.000000
Number of convenience stores
-0.602519
latitude
-0.591067
longitude
-0.806317
Number of bedrooms
-0.046856
House size (sqft)
0.001795
House price of unit area
-0.673613

                                          Number of convenience stores  \
Transaction date                                              0.009544
House Age                                                     0.049593
Distance from nearest Metro station (km)                     -0.602519
Number of convenience stores                                  1.000000
latitude                                                      0.444143
longitude                                                     0.449099
Number of bedrooms                                            0.043638
House size (sqft)                                             0.033286
House price of unit area                                      0.571005

                                          latitude  longitude  \
Transaction date                          0.035016  -0.041065
House Age                                 0.054420  -0.048520
Distance from nearest Metro station (km) -0.591067  -0.806317
Number of convenience stores              0.444143   0.449099
latitude                                  1.000000   0.412924
longitude                                 0.412924   1.000000
Number of bedrooms                        0.043921   0.041680
House size (sqft)                         0.031696   0.009322
House price of unit area                  0.546307   0.523287

                                          Number of bedrooms  \
Transaction date                                    0.061985
House Age                                          -0.008756
Distance from nearest Metro station (km)           -0.046856
Number of convenience stores                        0.043638
latitude                                            0.043921
longitude                                           0.041680
Number of bedrooms                                  1.000000
```

```
House size (sqft)                                          0.752276
House price of unit area                                   0.050265

                                          House size (sqft)  \
Transaction date                                   0.068405
House Age                                         -0.060361
Distance from nearest Metro station (km)           0.001795
Number of convenience stores                       0.033286
latitude                                           0.031696
longitude                                          0.009322
Number of bedrooms                                 0.752276
House size (sqft)                                  1.000000
House price of unit area                           0.046489

                                          House price of unit area
Transaction date                                          0.087529
House Age                                                -0.210567
Distance from nearest Metro station (km)                 -0.673613
Number of convenience stores                              0.571005
latitude                                                  0.546307
longitude                                                 0.523287
Number of bedrooms                                        0.050265
House size (sqft)                                         0.046489
House price of unit area                                  1.000000
```

1. From the correlation table we see that "Number of convenience stores", "latitude" and "longitude" have a high positive relation with "House price", while "Distance from nearest Metro Station" has a high negative relation which means the higher the distance the lower would be the House Price.

2. The "Number of bedrooms" and "House size" have a very less relation with the "House price", concluding that "Distance from nearest Metro station" and "Number of convenience stores" are a greater factor determining house prices.

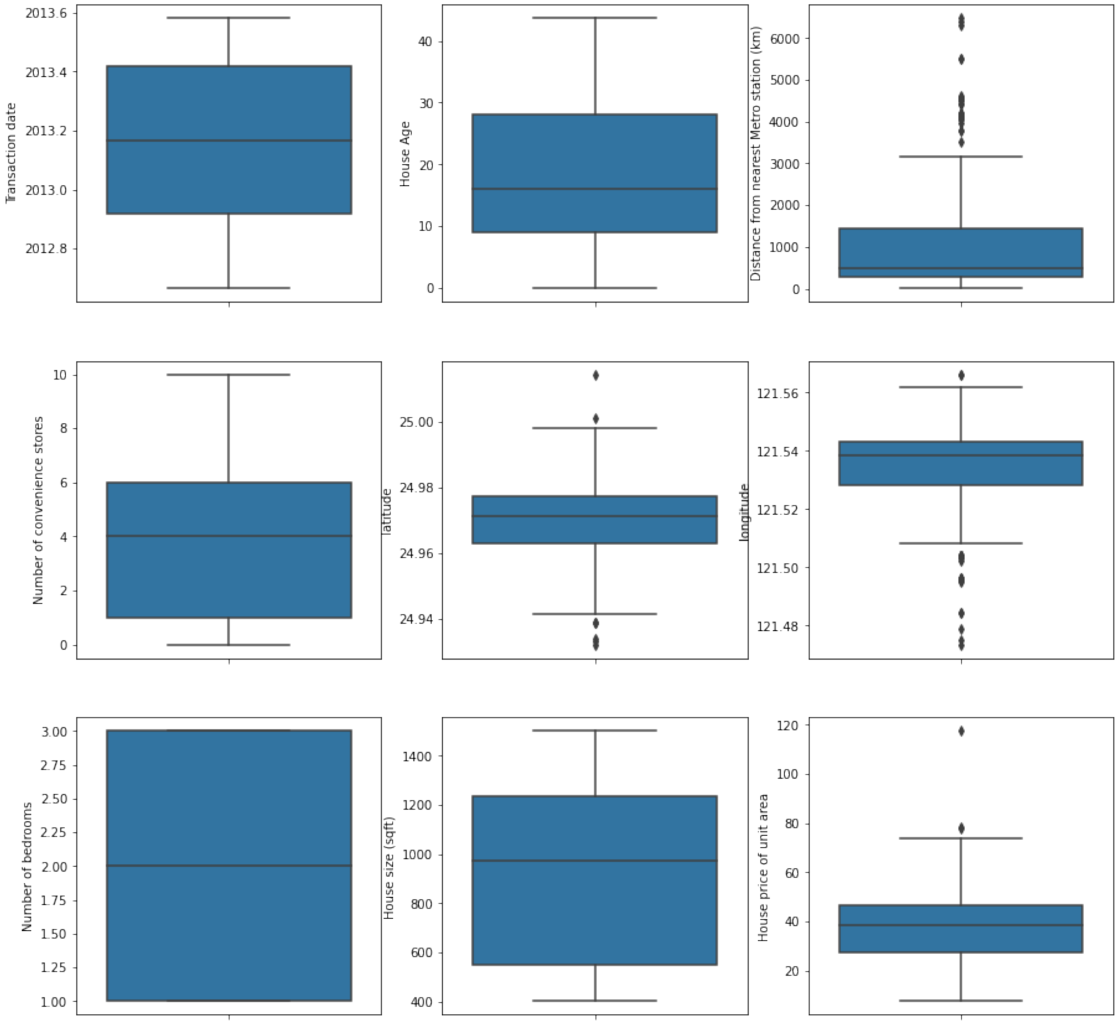
[7]:
```
# statistical analysis of each column
df.describe()
```

[7]:
```
       Transaction date   House Age  Distance from nearest Metro station (km)  \
count        414.000000  414.000000                                414.000000
mean        2013.148953   17.712560                               1083.885689
std            0.281995   11.392485                               1262.109595
min         2012.666667    0.000000                                 23.382840
25%         2012.916667    9.025000                                289.324800
50%         2013.166667   16.100000                                492.231300
75%         2013.416667   28.150000                               1454.279000
max         2013.583333   43.800000                               6488.021000

       Number of convenience stores    latitude   longitude  \
count                       414.000000  414.000000  414.000000
mean                          4.094203   24.969030  121.533361
std                           2.945562    0.012410    0.015347
min                           0.000000   24.932070  121.473530
25%                           1.000000   24.963000  121.528085
50%                           4.000000   24.971100  121.538630
75%                           6.000000   24.977455  121.543305
max                          10.000000   25.014590  121.566270

       Number of bedrooms  House size (sqft)  House price of unit area
count          414.000000         414.000000                414.000000
mean             1.987923         931.475845                 37.980193
std              0.818875         348.910269                 13.606488
min              1.000000         402.000000                  7.600000
25%              1.000000         548.000000                 27.700000
50%              2.000000         975.000000                 38.450000
75%              3.000000        1234.750000                 46.600000
max              3.000000        1500.000000                117.500000
```

1. From the above table we can see that "Distance from nearest Metro station (km)" and "House size (sqft)" have high standard deviation values which could mean presence of outliers.

[8]:

```python
j=1
plt.figure(figsize = (15,15))
for i in df.columns:
    plt.subplot(3,3,j)
    sns.boxplot(y = df[i])
    j+=1
```



From the above boxplots we can confirm the presence of outliers.

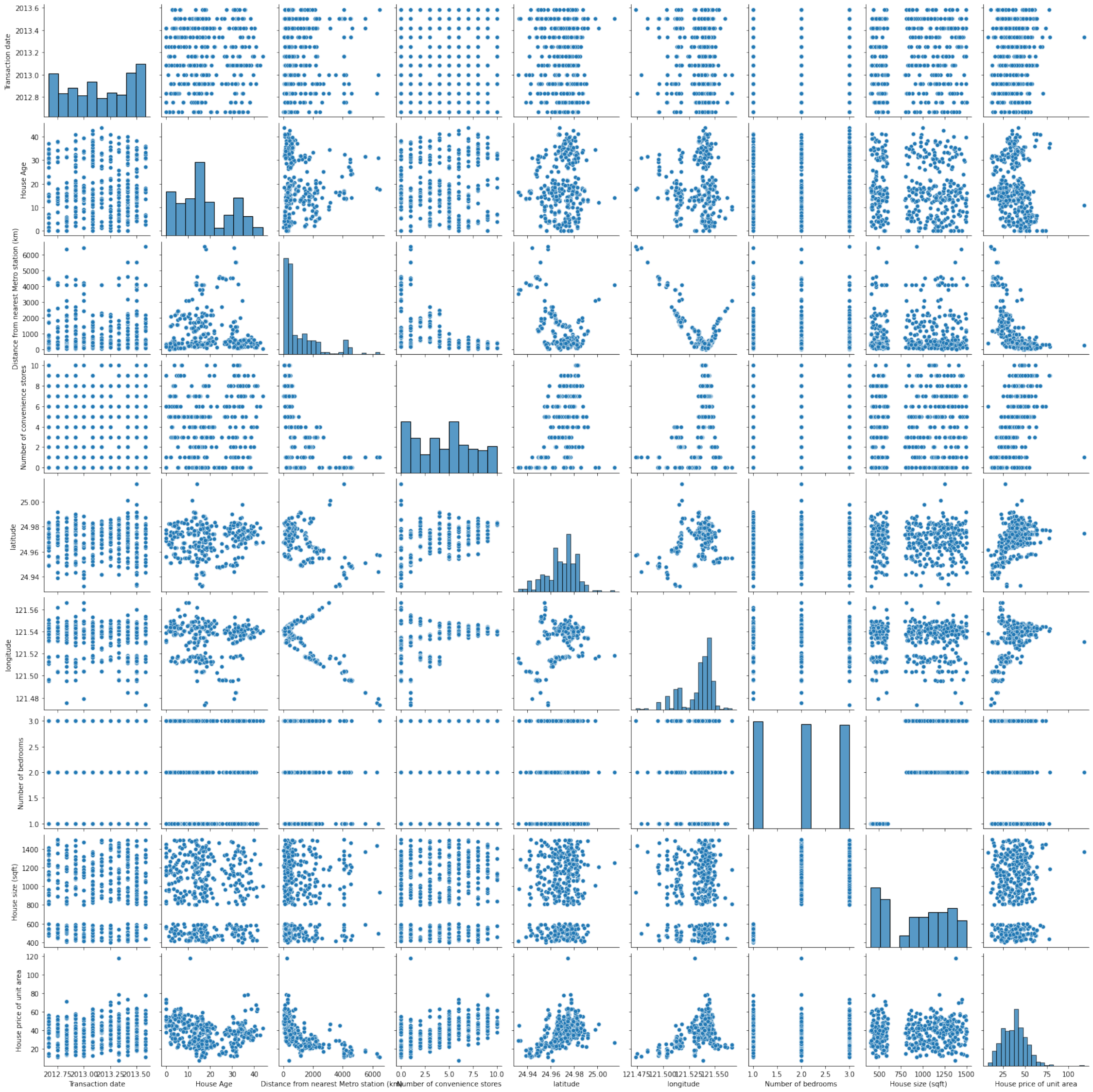
[9]:
```python
# checking number of unique elements in each column
df.nunique()
```

[9]:
```
Transaction date                            12
House Age                                  236
Distance from nearest Metro station (km)   259
Number of convenience stores                11
latitude                                   234
longitude                                  232
Number of bedrooms                           3
House size (sqft)                          328
House price of unit area                   270
dtype: int64
```

We see that Number of bedrooms have only 3 unique values

```
[10]: sns.pairplot(df)
```

```
[10]: <seaborn.axisgrid.PairGrid at 0x27690725130>
```



From the pairplot we can conclude that House Price lacks strong relation with other features

```
[11]: #handling outliers
      Q1 = df.quantile(q=.25)
      Q3 = df.quantile(q=.75)

      #calculating iqr range
      IQR = Q3 - Q1
```

```
#only keep rows in dataframe that have values within 1.5*IQR of Q1 and Q3
data_clean = df[~((df < (Q1-1.5*IQR)) | (df > (Q3+1.5*IQR))).any(axis=1)]

#find how many rows are left in the dataframe
data_clean.shape
```

[11]:
```
(371, 9)
```

1 Linear Regression

[12]:
```
from sklearn import linear_model
```

[13]:
```
# selecting feature columns
X = data_clean.iloc[:,0:-1]

#selecting target column
y = data_clean.iloc[:,-1]
```

[14]:
```
#splitting data into trainig and testing data
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.
 ↪3,random_state=1)
```

[15]:
```
#creating model object
reg = linear_model.LinearRegression()

# train the model using the training sets
reg.fit(X_train, y_train)
```

[15]:
```
LinearRegression()
```

[16]:
```
#predicting results using the model
y_pred = reg.predict(X_test)
```

[17]:
```
#importing performance metrics
from sklearn.metrics import r2_score
```

[18]:
```
print(r2_score(y_test, y_pred))
```

```
0.6474184595584653
```

2 Random Forest Regressor

[19]:
```
from sklearn.ensemble import RandomForestRegressor
```

[20]:

```
# selecting feature columns
X1 = data_clean.iloc[:,0:-1]

#selecting target column
y1 = data_clean.iloc[:,-1]


X1_train, X1_test, y1_train, y1_test = train_test_split(X1, y1, test_size=0.
 ↪3,random_state=1)
```

[21]:

```
#Randomized Search CV

#number of n_estimators in random forest
n_estimators = [int(x) for x in np.linspace(start = 100, stop = 1200, num = 12)]

# Number of trees in random forest
n_estimators = [int(x) for x in np.linspace(start = 100, stop = 1200, num = 12)]

# Number of features to consider at every split
max_features = ['auto', 'sqrt']

# Maximum number of levels in tree
max_depth = [int(x) for x in np.linspace(5, 30, num = 6)]

# Minimum number of samples required to split a node
min_samples_split = [2, 5, 10, 15, 100]

# Minimum number of samples required at each leaf node
min_samples_leaf = [1, 2, 5, 10]
```

[22]:

```
random_grid = {'n_estimators': n_estimators,
               'max_features': max_features,
               'max_depth': max_depth,
               'min_samples_split': min_samples_split,
               'min_samples_leaf': min_samples_leaf}

print(random_grid)
```

```
{'n_estimators': [100, 200, 300, 400, 500, 600, 700, 800, 900, 1000, 1100,
1200], 'max_features': ['auto', 'sqrt'], 'max_depth': [5, 10, 15, 20, 25, 30],
'min_samples_split': [2, 5, 10, 15, 100], 'min_samples_leaf': [1, 2, 5, 10]}
```

[23]:

```
#creating model object
rf = RandomForestRegressor()
```

[24]:

```
from sklearn.model_selection import RandomizedSearchCV
rf_random = RandomizedSearchCV(estimator = rf, param_distributions =␣
 ↪random_grid,scoring='r2', n_iter = 10, cv = 5, verbose=2, random_state = 0)
```

[25]:

```
rf_random.fit(X1_train,y1_train)
```

```
Fitting 5 folds for each of 10 candidates, totalling 50 fits
[CV] END max_depth=30, max_features=sqrt, min_samples_leaf=2,
min_samples_split=10, n_estimators=900; total time=   0.7s
[CV] END max_depth=30, max_features=sqrt, min_samples_leaf=2,
min_samples_split=10, n_estimators=900; total time=   0.8s
[CV] END max_depth=30, max_features=sqrt, min_samples_leaf=2,
min_samples_split=10, n_estimators=900; total time=   0.7s
[CV] END max_depth=30, max_features=sqrt, min_samples_leaf=2,
min_samples_split=10, n_estimators=900; total time=   0.7s
[CV] END max_depth=30, max_features=sqrt, min_samples_leaf=2,
min_samples_split=10, n_estimators=900; total time=   0.7s

C:\Users\LENOVO\AppData\Local\Programs\Python\Python38\lib\site-
packages\sklearn\ensemble\_forest.py:416: FutureWarning: `max_features='auto'`
has been deprecated in 1.1 and will be removed in 1.3. To keep the past
behaviour, explicitly set `max_features=1.0` or remove this parameter as it is
also the default value for RandomForestRegressors and ExtraTreesRegressors.
  warn(

[CV] END max_depth=30, max_features=auto, min_samples_leaf=10,
min_samples_split=10, n_estimators=400; total time=   0.3s

C:\Users\LENOVO\AppData\Local\Programs\Python\Python38\lib\site-
packages\sklearn\ensemble\_forest.py:416: FutureWarning: `max_features='auto'`
has been deprecated in 1.1 and will be removed in 1.3. To keep the past
behaviour, explicitly set `max_features=1.0` or remove this parameter as it is
also the default value for RandomForestRegressors and ExtraTreesRegressors.
  warn(

[CV] END max_depth=30, max_features=auto, min_samples_leaf=10,
min_samples_split=10, n_estimators=400; total time=   0.3s

C:\Users\LENOVO\AppData\Local\Programs\Python\Python38\lib\site-
packages\sklearn\ensemble\_forest.py:416: FutureWarning: `max_features='auto'`
has been deprecated in 1.1 and will be removed in 1.3. To keep the past
behaviour, explicitly set `max_features=1.0` or remove this parameter as it is
also the default value for RandomForestRegressors and ExtraTreesRegressors.
  warn(

[CV] END max_depth=30, max_features=auto, min_samples_leaf=10,
min_samples_split=10, n_estimators=400; total time=   0.3s

C:\Users\LENOVO\AppData\Local\Programs\Python\Python38\lib\site-
packages\sklearn\ensemble\_forest.py:416: FutureWarning: `max_features='auto'`
has been deprecated in 1.1 and will be removed in 1.3. To keep the past
behaviour, explicitly set `max_features=1.0` or remove this parameter as it is
also the default value for RandomForestRegressors and ExtraTreesRegressors.
  warn(
[CV] END max_depth=30, max_features=auto, min_samples_leaf=10,
min_samples_split=10, n_estimators=400; total time=   0.3s

C:\Users\LENOVO\AppData\Local\Programs\Python\Python38\lib\site-
packages\sklearn\ensemble\_forest.py:416: FutureWarning: `max_features='auto'`
has been deprecated in 1.1 and will be removed in 1.3. To keep the past
behaviour, explicitly set `max_features=1.0` or remove this parameter as it is
also the default value for RandomForestRegressors and ExtraTreesRegressors.
  warn(

[CV] END max_depth=30, max_features=auto, min_samples_leaf=10,
min_samples_split=10, n_estimators=400; total time=   0.3s

C:\Users\LENOVO\AppData\Local\Programs\Python\Python38\lib\site-
packages\sklearn\ensemble\_forest.py:416: FutureWarning: `max_features='auto'`
has been deprecated in 1.1 and will be removed in 1.3. To keep the past
behaviour, explicitly set `max_features=1.0` or remove this parameter as it is
also the default value for RandomForestRegressors and ExtraTreesRegressors.
  warn(

[CV] END max_depth=20, max_features=auto, min_samples_leaf=10,
min_samples_split=10, n_estimators=1000; total time=   0.9s

C:\Users\LENOVO\AppData\Local\Programs\Python\Python38\lib\site-
packages\sklearn\ensemble\_forest.py:416: FutureWarning: `max_features='auto'`
has been deprecated in 1.1 and will be removed in 1.3. To keep the past
behaviour, explicitly set `max_features=1.0` or remove this parameter as it is
also the default value for RandomForestRegressors and ExtraTreesRegressors.
  warn(

[CV] END max_depth=20, max_features=auto, min_samples_leaf=10,
min_samples_split=10, n_estimators=1000; total time=   0.8s

C:\Users\LENOVO\AppData\Local\Programs\Python\Python38\lib\site-
packages\sklearn\ensemble\_forest.py:416: FutureWarning: `max_features='auto'`
has been deprecated in 1.1 and will be removed in 1.3. To keep the past
behaviour, explicitly set `max_features=1.0` or remove this parameter as it is
also the default value for RandomForestRegressors and ExtraTreesRegressors.
  warn(

[CV] END max_depth=20, max_features=auto, min_samples_leaf=10,
min_samples_split=10, n_estimators=1000; total time=   0.9s

C:\Users\LENOVO\AppData\Local\Programs\Python\Python38\lib\site-
packages\sklearn\ensemble\_forest.py:416: FutureWarning: `max_features='auto'`
has been deprecated in 1.1 and will be removed in 1.3. To keep the past
behaviour, explicitly set `max_features=1.0` or remove this parameter as it is
also the default value for RandomForestRegressors and ExtraTreesRegressors.
  warn(

[CV] END max_depth=20, max_features=auto, min_samples_leaf=10,
min_samples_split=10, n_estimators=1000; total time=   0.9s
```

```
C:\Users\LENOVO\AppData\Local\Programs\Python\Python38\lib\site-
packages\sklearn\ensemble\_forest.py:416: FutureWarning: `max_features='auto'`
has been deprecated in 1.1 and will be removed in 1.3. To keep the past
behaviour, explicitly set `max_features=1.0` or remove this parameter as it is
also the default value for RandomForestRegressors and ExtraTreesRegressors.
  warn(
```

```
[CV] END max_depth=20, max_features=auto, min_samples_leaf=10,
min_samples_split=10, n_estimators=1000; total time=   0.9s
[CV] END max_depth=10, max_features=sqrt, min_samples_leaf=2,
min_samples_split=100, n_estimators=800; total time=   0.5s
[CV] END max_depth=10, max_features=sqrt, min_samples_leaf=2,
min_samples_split=100, n_estimators=800; total time=   0.5s
[CV] END max_depth=10, max_features=sqrt, min_samples_leaf=2,
min_samples_split=100, n_estimators=800; total time=   0.5s
[CV] END max_depth=10, max_features=sqrt, min_samples_leaf=2,
min_samples_split=100, n_estimators=800; total time=   0.5s
[CV] END max_depth=10, max_features=sqrt, min_samples_leaf=2,
min_samples_split=100, n_estimators=800; total time=   0.5s
[CV] END max_depth=10, max_features=sqrt, min_samples_leaf=1,
min_samples_split=15, n_estimators=800; total time=   0.6s
[CV] END max_depth=10, max_features=sqrt, min_samples_leaf=1,
min_samples_split=15, n_estimators=800; total time=   0.6s
[CV] END max_depth=10, max_features=sqrt, min_samples_leaf=1,
min_samples_split=15, n_estimators=800; total time=   0.6s
[CV] END max_depth=10, max_features=sqrt, min_samples_leaf=1,
min_samples_split=15, n_estimators=800; total time=   0.6s
[CV] END max_depth=10, max_features=sqrt, min_samples_leaf=1,
min_samples_split=15, n_estimators=800; total time=   0.6s
[CV] END max_depth=20, max_features=sqrt, min_samples_leaf=1,
min_samples_split=100, n_estimators=400; total time=   0.2s
[CV] END max_depth=20, max_features=sqrt, min_samples_leaf=1,
min_samples_split=100, n_estimators=400; total time=   0.2s
[CV] END max_depth=20, max_features=sqrt, min_samples_leaf=1,
min_samples_split=100, n_estimators=400; total time=   0.2s
[CV] END max_depth=20, max_features=sqrt, min_samples_leaf=1,
min_samples_split=100, n_estimators=400; total time=   0.2s
[CV] END max_depth=20, max_features=sqrt, min_samples_leaf=1,
min_samples_split=100, n_estimators=400; total time=   0.2s
```

```
C:\Users\LENOVO\AppData\Local\Programs\Python\Python38\lib\site-
packages\sklearn\ensemble\_forest.py:416: FutureWarning: `max_features='auto'`
has been deprecated in 1.1 and will be removed in 1.3. To keep the past
behaviour, explicitly set `max_features=1.0` or remove this parameter as it is
also the default value for RandomForestRegressors and ExtraTreesRegressors.
  warn(
```

```
[CV] END max_depth=15, max_features=auto, min_samples_leaf=2,
min_samples_split=5, n_estimators=200; total time=   0.2s
```

```
C:\Users\LENOVO\AppData\Local\Programs\Python\Python38\lib\site-
packages\sklearn\ensemble\_forest.py:416: FutureWarning: `max_features='auto'`
has been deprecated in 1.1 and will be removed in 1.3. To keep the past
behaviour, explicitly set `max_features=1.0` or remove this parameter as it is
also the default value for RandomForestRegressors and ExtraTreesRegressors.
  warn(

[CV] END max_depth=15, max_features=auto, min_samples_leaf=2,
min_samples_split=5, n_estimators=200; total time=   0.1s
[CV] END max_depth=15, max_features=auto, min_samples_leaf=2,
min_samples_split=5, n_estimators=200; total time=   0.1s

C:\Users\LENOVO\AppData\Local\Programs\Python\Python38\lib\site-
packages\sklearn\ensemble\_forest.py:416: FutureWarning: `max_features='auto'`
has been deprecated in 1.1 and will be removed in 1.3. To keep the past
behaviour, explicitly set `max_features=1.0` or remove this parameter as it is
also the default value for RandomForestRegressors and ExtraTreesRegressors.
  warn(
C:\Users\LENOVO\AppData\Local\Programs\Python\Python38\lib\site-
packages\sklearn\ensemble\_forest.py:416: FutureWarning: `max_features='auto'`
has been deprecated in 1.1 and will be removed in 1.3. To keep the past
behaviour, explicitly set `max_features=1.0` or remove this parameter as it is
also the default value for RandomForestRegressors and ExtraTreesRegressors.
  warn(

[CV] END max_depth=15, max_features=auto, min_samples_leaf=2,
min_samples_split=5, n_estimators=200; total time=   0.1s

C:\Users\LENOVO\AppData\Local\Programs\Python\Python38\lib\site-
packages\sklearn\ensemble\_forest.py:416: FutureWarning: `max_features='auto'`
has been deprecated in 1.1 and will be removed in 1.3. To keep the past
behaviour, explicitly set `max_features=1.0` or remove this parameter as it is
also the default value for RandomForestRegressors and ExtraTreesRegressors.
  warn(

[CV] END max_depth=15, max_features=auto, min_samples_leaf=2,
min_samples_split=5, n_estimators=200; total time=   0.1s
[CV] END max_depth=5, max_features=sqrt, min_samples_leaf=1,
min_samples_split=15, n_estimators=200; total time=   0.1s
[CV] END max_depth=5, max_features=sqrt, min_samples_leaf=1,
min_samples_split=15, n_estimators=200; total time=   0.1s
[CV] END max_depth=5, max_features=sqrt, min_samples_leaf=1,
min_samples_split=15, n_estimators=200; total time=   0.1s
[CV] END max_depth=5, max_features=sqrt, min_samples_leaf=1,
min_samples_split=15, n_estimators=200; total time=   0.1s
[CV] END max_depth=5, max_features=sqrt, min_samples_leaf=1,
min_samples_split=15, n_estimators=200; total time=   0.1s
[CV] END max_depth=20, max_features=sqrt, min_samples_leaf=2,
min_samples_split=15, n_estimators=300; total time=   0.2s
[CV] END max_depth=20, max_features=sqrt, min_samples_leaf=2,
min_samples_split=15, n_estimators=300; total time=   0.2s
[CV] END max_depth=20, max_features=sqrt, min_samples_leaf=2,
min_samples_split=15, n_estimators=300; total time=   0.2s
[CV] END max_depth=20, max_features=sqrt, min_samples_leaf=2,
min_samples_split=15, n_estimators=300; total time=   0.2s
[CV] END max_depth=20, max_features=sqrt, min_samples_leaf=2,
min_samples_split=15, n_estimators=300; total time=   0.2s
[CV] END max_depth=20, max_features=sqrt, min_samples_leaf=5,
min_samples_split=10, n_estimators=500; total time=   0.3s
[CV] END max_depth=20, max_features=sqrt, min_samples_leaf=5,
min_samples_split=10, n_estimators=500; total time=   0.3s
[CV] END max_depth=20, max_features=sqrt, min_samples_leaf=5,
min_samples_split=10, n_estimators=500; total time=   0.3s
[CV] END max_depth=20, max_features=sqrt, min_samples_leaf=5,
min_samples_split=10, n_estimators=500; total time=   0.3s
[CV] END max_depth=20, max_features=sqrt, min_samples_leaf=5,
min_samples_split=10, n_estimators=500; total time=   0.3s
```

```
[25]: RandomizedSearchCV(cv=5, estimator=RandomForestRegressor(),
                         param_distributions={'max_depth': [5, 10, 15, 20, 25, 30],
                                              'max_features': ['auto', 'sqrt'],
                                              'min_samples_leaf': [1, 2, 5, 10],
                                              'min_samples_split': [2, 5, 10, 15,
                                                                    100],
                                              'n_estimators': [100, 200, 300, 400,
                                                               500, 600, 700, 800,
                                                               900, 1000, 1100,
                                                               1200]},
                         random_state=0, scoring='r2', verbose=2)
```

```
[26]: #checking best parameters for the model
      rf_random.best_params_
```

```
[26]: {'n_estimators': 900,
       'min_samples_split': 10,
       'min_samples_leaf': 2,
       'max_features': 'sqrt',
       'max_depth': 30}
```

```
[27]: #creating model object with the best parameters
      mdl = RandomForestRegressor(n_estimators= 200,min_samples_split=␣
        ↪5,min_samples_leaf= 2,max_features= 'auto',
                                  max_depth= 15, random_state = 0)
```

```
[28]: mdl.fit(X1_train, y1_train)
```

```
C:\Users\LENOVO\AppData\Local\Programs\Python\Python38\lib\site-
packages\sklearn\ensemble\_forest.py:416: FutureWarning: `max_features='auto'`
```

```
has been deprecated in 1.1 and will be removed in 1.3. To keep the past
behaviour, explicitly set `max_features=1.0` or remove this parameter as it is
also the default value for RandomForestRegressors and ExtraTreesRegressors.
  warn(
```

```
[28]: RandomForestRegressor(max_depth=15, max_features='auto', min_samples_leaf=2,
                            min_samples_split=5, n_estimators=200, random_state=0)
```

```
[29]: y1_pred = mdl.predict(X1_test)
```

```
[30]: print(r2_score(y1_test, y1_pred))
```

```
0.6915293457407516
```

3 Conclusion

```
[31]: print('Accuracy of linear regression : ',r2_score(y_test, y_pred))
      print('Accuracy of random forest regressor : ',r2_score(y1_test, y1_pred))
```

```
Accuracy of linear regression :  0.6474184595584653
Accuracy of random forest regressor :  0.6915293457407516
```

3.0.1 Linear Regression

1. The main assumption is that the dependent variable is linearly dependent on independent variables, which is not the case with this data.

2. Other reason being multicollinearity, meaning the independent features also exibit relationship between themselves.

3.0.2 Random Forest Regressor

1. Random Forest is unable to discover trends based on the data. The predictions it makes are always in the range of the training set.