Mini Project

Name : Rahul Sampatrao Patil
Roll No :4176

Prediction of House Rent Data

September 23, 2024

[1]:
```
import numpy as np # linear algebra
import pandas as pd # data processing

import os
for dirname, _, filenames in os.walk('/kaggle/input'):
    for filename in filenames:
        print(os.path.join(dirname, filename))
```

[2]:
```
df=pd.read_csv('C:/Users/USER/Desktop/Mini_Project ML/archive/
↪House_Rent_Dataset.csv')
df.head()
```

[2]:

	Posted On	BHK	Rent	Size	Floor	Area Type	\
0	2022-05-18	2	10000	1100	Ground out of 2	Super Area	
1	2022-05-13	2	20000	800	1 out of 3	Super Area	
2	2022-05-16	2	17000	1000	1 out of 3	Super Area	
3	2022-07-04	2	10000	800	1 out of 2	Super Area	
4	2022-05-09	2	7500	850	1 out of 2	Carpet Area	

	Area Locality	City	Furnishing Status	Tenant Preferred	\
0	Bandel	Kolkata	Unfurnished	Bachelors/Family	
1	Phool Bagan, Kankurgachi	Kolkata	Semi-Furnished	Bachelors/Family	

```
2   Salt Lake City Sector 2  Kolkata     Semi-Furnished  Bachelors/Family
3                Dumdum Park  Kolkata       Unfurnished  Bachelors/Family
4              South Dum Dum  Kolkata       Unfurnished          Bachelors

   Bathroom Point of Contact
0         2     Contact Owner
1         1     Contact Owner
2         1     Contact Owner
3         1     Contact Owner
4         1     Contact Owner
```

[3]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4746 entries, 0 to 4745
Data columns (total 12 columns):
 #   Column              Non-Null Count  Dtype
---  ------              --------------  -----
 0   Posted On           4746 non-null   object
 1   BHK                 4746 non-null   int64
 2   Rent                4746 non-null   int64
 3   Size                4746 non-null   int64
 4   Floor               4746 non-null   object
 5   Area Type           4746 non-null   object
 6   Area Locality       4746 non-null   object
 7   City                4746 non-null   object
 8   Furnishing Status   4746 non-null   object
 9   Tenant Preferred    4746 non-null   object
 10  Bathroom            4746 non-null   int64
 11  Point of Contact    4746 non-null   object
dtypes: int64(4), object(8)
memory usage: 445.1+ KB
```

[4]:
```
df["Area Type"].value_counts()
```

[4]:
```
Area Type
Super Area     2446
Carpet Area    2298
Built Area        2
Name: count, dtype: int64
```

[5]:
```
df.City.value_counts()
```

[5]:
```
City
Mumbai       972
Chennai      891
Bangalore    886
```

```
Hyderabad    868
Delhi        605
Kolkata      524
Name: count, dtype: int64
```

[6]:
```
df['Furnishing Status'].value_counts()
```

[6]:
```
Furnishing Status
Semi-Furnished    2251
Unfurnished       1815
Furnished          680
Name: count, dtype: int64
```

[7]:
```
df['Point of Contact'].value_counts()
```

[7]:
```
Point of Contact
Contact Owner      3216
Contact Agent      1529
Contact Builder       1
Name: count, dtype: int64
```

[8]:
```
df['Tenant Preferred'].value_counts()
```

[8]:
```
Tenant Preferred
Bachelors/Family    3444
Bachelors            830
Family               472
Name: count, dtype: int64
```

[9]:
```
# remove outlier
import plotly.express as px
fig = px.box(df, x="Rent", title='Boxplot for Rent Prices')
fig.show()
```

[10]:
```
print(np.where(df['Rent']>=100000))
```

```
(array([  104,   530,   531,   533,   542,   543,   545,   547,   556,   560,   579,
         589,   591,   593,   595,   600,   601,   603,   606,   607,   615,   617,
         623,   625,   627,   631,   634,   635,   639,   640,   643,   651,   658,
         666,   673,   680,   682,   685,   687,   688,   699,   700,   701,   702,
         706,   710,   722,   726,   727,   728,   733,   735,   742,   745,   746,
         749,   754,   757,   766,   768,   770,   771,   773,   777,   778,   780,
         788,   789,   790,   792,   795,   798,   799,   815,   827,   829,   832,
         835,   839,   842,   847,   848,   850,   851,   857,   858,   859,   867,
         868,   869,   871,   874,   876,   889,   898,   902,   906,   914,   921,
         923,   927,   930,   932,   936,   951,   952,   973,   977,   984,   985,
         986,   988,   991,   992,   994,   995,  1001,  1004,  1005,  1010,  1019,
```

```
       1023, 1024, 1027, 1029, 1030, 1031, 1032, 1034, 1035, 1037, 1038,
       1042, 1045, 1052, 1055, 1057, 1064, 1065, 1071, 1084, 1086, 1087,
       1089, 1093, 1099, 1105, 1112, 1113, 1115, 1120, 1122, 1143, 1146,
       1151, 1159, 1160, 1161, 1163, 1165, 1170, 1171, 1175, 1182, 1187,
       1189, 1196, 1199, 1202, 1205, 1208, 1221, 1222, 1230, 1233, 1237,
       1238, 1239, 1242, 1244, 1247, 1251, 1255, 1260, 1261, 1273, 1275,
       1287, 1290, 1292, 1302, 1303, 1309, 1319, 1329, 1336, 1341, 1344,
       1345, 1352, 1366, 1369, 1378, 1380, 1383, 1384, 1388, 1389, 1391,
       1392, 1393, 1399, 1401, 1402, 1421, 1425, 1431, 1438, 1439, 1449,
       1451, 1459, 1460, 1471, 1476, 1482, 1484, 1485, 1489, 1495, 1553,
       1576, 1620, 1680, 1718, 1798, 1810, 1829, 1837, 1877, 1910, 1935,
       2028, 2048, 2079, 2108, 2186, 2209, 2213, 2229, 2236, 2264, 2340,
       2371, 2399, 2403, 2486, 2533, 2561, 2577, 2595, 2598, 2623, 2642,
       2656, 2718, 2750, 2755, 2794, 2823, 2845, 2846, 2848, 2849, 2857,
       2864, 2869, 2904, 2912, 2923, 2932, 2964, 2990, 3041, 3134, 3148,
       3288, 3298, 3320, 3401, 3453, 3457, 3509, 3518, 3581, 3622, 3639,
       3656, 3702, 3709, 3770, 3792, 3795, 3824, 3879, 3975, 3989, 4004,
       4021, 4041, 4097, 4161, 4185, 4241, 4425, 4457, 4543, 4669, 4716],
      dtype=int64),)
```

[11]:
```
df.drop([ 104,  530,  531,  533,  542,  543,  545,  547,  556,  560,  579,
        589,  591,  593,  595,  600,  601,  603,  606,  607,  615,  617,
        623,  625,  627,  631,  634,  635,  639,  640,  643,  651,  658,
        666,  673,  680,  682,  685,  687,  688,  699,  700,  701,  702,
        706,  710,  722,  726,  727,  728,  733,  735,  742,  745,  746,
        749,  754,  757,  766,  768,  770,  771,  773,  777,  778,  780,
        788,  789,  790,  792,  795,  798,  799,  815,  827,  829,  832,
        835,  839,  842,  847,  848,  850,  851,  857,  858,  859,  867,
        868,  869,  871,  874,  876,  889,  898,  902,  906,  914,  921,
        923,  927,  930,  932,  936,  951,  952,  973,  977,  984,  985,
        986,  988,  991,  992,  994,  995, 1001, 1004, 1005, 1010, 1019,
       1023, 1024, 1027, 1029, 1030, 1031, 1032, 1034, 1035, 1037, 1038,
       1042, 1045, 1052, 1055, 1057, 1064, 1065, 1071, 1084, 1086, 1087,
       1089, 1093, 1099, 1105, 1112, 1113, 1115, 1120, 1122, 1143, 1146,
       1151, 1159, 1160, 1161, 1163, 1165, 1170, 1171, 1175, 1182, 1187,
       1189, 1196, 1199, 1202, 1205, 1208, 1221, 1222, 1230, 1233, 1237,
       1238, 1239, 1242, 1244, 1247, 1251, 1255, 1260, 1261, 1273, 1275,
       1287, 1290, 1292, 1302, 1303, 1309, 1319, 1329, 1336, 1341, 1344,
       1345, 1352, 1366, 1369, 1378, 1380, 1383, 1384, 1388, 1389, 1391,
       1392, 1393, 1399, 1401, 1402, 1421, 1425, 1431, 1438, 1439, 1449,
       1451, 1459, 1460, 1471, 1476, 1482, 1484, 1485, 1489, 1495, 1553,
       1576, 1620, 1680, 1718, 1798, 1810, 1829, 1837, 1877, 1910, 1935,
       2028, 2048, 2079, 2108, 2186, 2209, 2213, 2229, 2236, 2264, 2340,
       2371, 2399, 2403, 2486, 2533, 2561, 2577, 2595, 2598, 2623, 2642,
       2656, 2718, 2750, 2755, 2794, 2823, 2845, 2846, 2848, 2849, 2857,
       2864, 2869, 2904, 2912, 2923, 2932, 2964, 2990, 3041, 3134, 3148,
       3288, 3298, 3320, 3401, 3453, 3457, 3509, 3518, 3581, 3622, 3639,
```

```
      3656, 3702, 3709, 3770, 3792, 3795, 3824, 3879, 3975, 3989, 4004,
      4021, 4041, 4097, 4161, 4185, 4241, 4425, 4457, 4543, 4669, 4716],
 ↪axis=0, inplace=True)

fig = px.box(df, x="Rent",title='Boxplot for Rent Prices')
fig.show()
```

data preprocessing**

[12]:
```
houserent_data = pd.get_dummies(df, columns=['Area Type', 'City', 'Furnishing
 ↪Status', 'Tenant Preferred', 'Point of Contact'])
houserent_data.head()
```

[12]:

	Posted On	BHK	Rent	Size	Floor	Area Locality	\
0	2022-05-18	2	10000	1100	Ground out of 2	Bandel	
1	2022-05-13	2	20000	800	1 out of 3	Phool Bagan, Kankurgachi	
2	2022-05-16	2	17000	1000	1 out of 3	Salt Lake City Sector 2	
3	2022-07-04	2	10000	800	1 out of 2	Dumdum Park	
4	2022-05-09	2	7500	850	1 out of 2	South Dum Dum	

	Bathroom	Area Type_Built Area	Area Type_Carpet Area	\
0	2	False	False	
1	1	False	False	
2	1	False	False	
3	1	False	False	
4	1	False	True	

	Area Type_Super Area	...	City_Mumbai	Furnishing Status_Furnished	\
0	True	...	False	False	
1	True	...	False	False	
2	True	...	False	False	
3	True	...	False	False	
4	False	...	False	False	

	Furnishing Status_Semi-Furnished	Furnishing Status_Unfurnished	\
0	False	True	
1	True	False	
2	True	False	
3	False	True	
4	False	True	

	Tenant Preferred_Bachelors	Tenant Preferred_Bachelors/Family	\
0	False	True	
1	False	True	
2	False	True	
3	False	True	
4	True	False	

```
   Tenant Preferred_Family  Point of Contact_Contact Agent  \
0                    False                           False
1                    False                           False
2                    False                           False
3                    False                           False
4                    False                           False

   Point of Contact_Contact Builder  Point of Contact_Contact Owner
0                             False                           True
1                             False                           True
2                             False                           True
3                             False                           True
4                             False                           True

[5 rows x 25 columns]
```

[13]:
```
houserent_data = houserent_data.drop(['Posted On','Area_
↪Locality','Floor'],axis=1)
houserent_data.head()
```

[13]:
```
   BHK   Rent  Size  Bathroom  Area Type_Built Area  Area Type_Carpet Area  \
0    2  10000  1100         2                 False                  False
1    2  20000   800         1                 False                  False
2    2  17000  1000         1                 False                  False
3    2  10000   800         1                 False                  False
4    2   7500   850         1                 False                   True

   Area Type_Super Area  City_Bangalore  City_Chennai  City_Delhi  ...  \
0                  True           False         False       False  ...
1                  True           False         False       False  ...
2                  True           False         False       False  ...
3                  True           False         False       False  ...
4                 False           False         False       False  ...

   City_Mumbai  Furnishing Status_Furnished  Furnishing Status_Semi-Furnished  \
0        False                        False                             False
1        False                        False                              True
2        False                        False                              True
3        False                        False                             False
4        False                        False                             False

   Furnishing Status_Unfurnished  Tenant Preferred_Bachelors  \
0                           True                       False
1                          False                       False
2                          False                       False
3                           True                       False
```

```
4                                  True                                  True

   Tenant Preferred_Bachelors/Family  Tenant Preferred_Family  \
0                                True                    False
1                                True                    False
2                                True                    False
3                                True                    False
4                               False                    False

   Point of Contact_Contact Agent  Point of Contact_Contact Builder  \
0                           False                             False
1                           False                             False
2                           False                             False
3                           False                             False
4                           False                             False

   Point of Contact_Contact Owner
0                            True
1                            True
2                            True
3                            True
4                            True

[5 rows x 22 columns]
```

[14]:
```
houserent_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 4427 entries, 0 to 4745
Data columns (total 22 columns):
 #   Column                            Non-Null Count  Dtype
---  ------                            --------------  -----
 0   BHK                               4427 non-null   int64
 1   Rent                              4427 non-null   int64
 2   Size                              4427 non-null   int64
 3   Bathroom                          4427 non-null   int64
 4   Area Type_Built Area              4427 non-null   bool
 5   Area Type_Carpet Area             4427 non-null   bool
 6   Area Type_Super Area              4427 non-null   bool
 7   City_Bangalore                    4427 non-null   bool
 8   City_Chennai                      4427 non-null   bool
 9   City_Delhi                        4427 non-null   bool
 10  City_Hyderabad                    4427 non-null   bool
 11  City_Kolkata                      4427 non-null   bool
 12  City_Mumbai                       4427 non-null   bool
 13  Furnishing Status_Furnished       4427 non-null   bool
 14  Furnishing Status_Semi-Furnished  4427 non-null   bool
```

```
 15  Furnishing Status_Unfurnished        4427 non-null   bool
 16  Tenant Preferred_Bachelors           4427 non-null   bool
 17  Tenant Preferred_Bachelors/Family    4427 non-null   bool
 18  Tenant Preferred_Family              4427 non-null   bool
 19  Point of Contact_Contact Agent       4427 non-null   bool
 20  Point of Contact_Contact Builder     4427 non-null   bool
 21  Point of Contact_Contact Owner       4427 non-null   bool
dtypes: bool(18), int64(4)
memory usage: 250.7 KB
```

[15]:
```
X=houserent_data.drop("Rent",axis=1)
y=houserent_data["Rent"]
```

[16]:
```
# split the 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.
↪25,random_state=23)
```

[17]:
```
from sklearn.preprocessing import MinMaxScaler
scaler=MinMaxScaler()
fit=scaler.fit(x_train)
X_train=fit.transform(x_train)
X_test=fit.transform(x_test)
```

[18]:
```
from sklearn.ensemble import RandomForestRegressor
rf = RandomForestRegressor(random_state=0)
rf.fit(X_train,y_train)
print(rf.score(X_train,y_train))
print(rf.score(X_test,y_test))
```

```
0.9348895419561847
0.7301176301115039
```

[19]:
```
# Hyperparameter tunning
param_grid = {  'bootstrap': [True], 'max_depth': [5, 10, None], 'max_features':
↪ ['auto', 'sqrt'], 'n_estimators': [100, 300]}
```

[20]:
```
from sklearn.model_selection import GridSearchCV
rf_Grid = GridSearchCV(estimator = rf, param_grid = param_grid,

                       cv = 5, n_jobs = 1, verbose = 0,␣
↪return_train_score=True)
```

[21]:
```
rf_Grid.fit(X_train, y_train)
```

```
c:\Users\USER\AppData\Local\Programs\Python\Python312\Lib\site-
packages\sklearn\model_selection\_validation.py:547: FitFailedWarning:
```

```
30 fits failed out of a total of 60.
The score on these train-test partitions for these parameters will be set to
nan.
If these failures are not expected, you can try to debug them by setting
error_score='raise'.

Below are more details about the failures:
--------------------------------------------------------------------------------
30 fits failed with the following error:
Traceback (most recent call last):
  File "c:\Users\USER\AppData\Local\Programs\Python\Python312\Lib\site-
packages\sklearn\model_selection\_validation.py", line 895, in _fit_and_score
    estimator.fit(X_train, y_train, **fit_params)
  File "c:\Users\USER\AppData\Local\Programs\Python\Python312\Lib\site-
packages\sklearn\base.py", line 1467, in wrapper
    estimator._validate_params()
  File "c:\Users\USER\AppData\Local\Programs\Python\Python312\Lib\site-
packages\sklearn\base.py", line 666, in _validate_params
    validate_parameter_constraints(
  File "c:\Users\USER\AppData\Local\Programs\Python\Python312\Lib\site-
packages\sklearn\utils\_param_validation.py", line 95, in
validate_parameter_constraints
    raise InvalidParameterError(
sklearn.utils._param_validation.InvalidParameterError: The 'max_features'
parameter of RandomForestRegressor must be an int in the range [1, inf), a float
in the range (0.0, 1.0], a str among {'log2', 'sqrt'} or None. Got 'auto'
instead.


c:\Users\USER\AppData\Local\Programs\Python\Python312\Lib\site-
packages\sklearn\model_selection\_search.py:1051: UserWarning:

One or more of the test scores are non-finite: [       nan       nan 0.68441712
0.6901288        nan        nan
 0.74996592 0.75117293        nan        nan 0.72678883 0.72778081]

c:\Users\USER\AppData\Local\Programs\Python\Python312\Lib\site-
packages\sklearn\model_selection\_search.py:1051: UserWarning:

One or more of the train scores are non-finite: [       nan       nan 0.7075125
0.71271664       nan       nan
 0.86998246 0.87082268        nan        nan 0.93698346 0.93800361]


[21]: GridSearchCV(cv=5, estimator=RandomForestRegressor(random_state=0), n_jobs=1,
                   param_grid={'bootstrap': [True], 'max_depth': [5, 10, None],
```

```
                                'max_features': ['auto', 'sqrt'],
                                'n_estimators': [100, 300]},
                    return_train_score=True)
```

[22]:
```
rf_Grid.best_params_
```

[22] :
```
{'bootstrap': True,
 'max_depth': 10,
 'max_features': 'sqrt',
 'n_estimators': 300}
```

[23]:
```
print (f'Train Accuracy - : {rf_Grid.score(X_train,y_train):.4f}')
print (f'Test Accuracy - : {rf_Grid.score(X_test,y_test):.4f}')
```

```
Train Accuracy - : 0.8638
Test Accuracy - : 0.7726
```