assignment-4

April 16, 2024

1 Name: Sourabh Barala

2 Course: M.Sc. Data Science

3 Year: 1st

4 Reg. No.: 23MSD7044

5 Subject: Machine Learning and its Applications

6

7

8

9

10

11

[1]:
```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
```

[44]:
```
Housing=pd.read_csv('Housing.csv')
```

[45]:
```
Housing.head(5)
```

[45]:

	price	area	bedrooms	bathrooms	stories	mainroad	guestroom	basement	\
0	13300000	7420	4	2	3	yes	no	no	
1	12250000	8960	4	4	4	yes	no	no	
2	12250000	9960	3	2	2	yes	no	yes	
3	12215000	7500	4	2	2	yes	no	yes	
4	11410000	7420	4	1	2	yes	yes	yes	

```
  hotwaterheating airconditioning  parking prefarea furnishingstatus
0              no              yes        2      yes        furnished
1              no              yes        3       no        furnished
2              no               no        2      yes   semi-furnished
3              no              yes        3      yes        furnished
4              no              yes        2       no        furnished
```

[46]:
```
Housing.shape
```

[46]:
```
(545, 13)
```

[47]:
```
Housing.describe()
```

[47]:
```
              price          area    bedrooms   bathrooms     stories  \
count  5.450000e+02    545.000000  545.000000  545.000000  545.000000
mean   4.766729e+06   5150.541284    2.965138    1.286239    1.805505
std    1.870440e+06   2170.141023    0.738064    0.502470    0.867492
min    1.750000e+06   1650.000000    1.000000    1.000000    1.000000
25%    3.430000e+06   3600.000000    2.000000    1.000000    1.000000
50%    4.340000e+06   4600.000000    3.000000    1.000000    2.000000
75%    5.740000e+06   6360.000000    3.000000    2.000000    2.000000
max    1.330000e+07  16200.000000    6.000000    4.000000    4.000000

          parking
count  545.000000
mean     0.693578
std      0.861586
min      0.000000
25%      0.000000
50%      0.000000
75%      1.000000
max      3.000000
```

[48]:
```
Housing.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 545 entries, 0 to 544
Data columns (total 13 columns):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   price             545 non-null    int64
 1   area              545 non-null    int64
 2   bedrooms          545 non-null    int64
 3   bathrooms         545 non-null    int64
 4   stories           545 non-null    int64
 5   mainroad          545 non-null    object
```

```
 6   guestroom         545 non-null    object
 7   basement          545 non-null    object
 8   hotwaterheating   545 non-null    object
 9   airconditioning   545 non-null    object
 10  parking           545 non-null    int64
 11  prefarea          545 non-null    object
 12  furnishingstatus  545 non-null    object
dtypes: int64(6), object(7)
memory usage: 55.5+ KB
```

[49]:
```
Housing.corr()
```

```
/tmp/ipykernel_12463/3434865805.py:1: FutureWarning: The default value of
numeric_only in DataFrame.corr is deprecated. In a future version, it will
default to False. Select only valid columns or specify the value of numeric_only
to silence this warning.
  Housing.corr()
```

[49]:

	price	area	bedrooms	bathrooms	stories	parking
price	1.000000	0.535997	0.366494	0.517545	0.420712	0.384394
area	0.535997	1.000000	0.151858	0.193820	0.083996	0.352980
bedrooms	0.366494	0.151858	1.000000	0.373930	0.408564	0.139270
bathrooms	0.517545	0.193820	0.373930	1.000000	0.326165	0.177496
stories	0.420712	0.083996	0.408564	0.326165	1.000000	0.045547
parking	0.384394	0.352980	0.139270	0.177496	0.045547	1.000000

[50]:
```
object_cols=Housing.columns[Housing.dtypes=='object']
```

[51]:
```
object_cols
```

[51]:
```
Index(['mainroad', 'guestroom', 'basement', 'hotwaterheating',
       'airconditioning', 'prefarea', 'furnishingstatus'],
      dtype='object')
```

[52]:
```
for i in object_cols:
    print(f'{i}: {Housing[i].unique()}')

binary_obj_cols=['mainroad', 'guestroom', 'basement',␣
 ↪'hotwaterheating','airconditioning', 'prefarea']
```

```
mainroad: ['yes' 'no']
guestroom: ['no' 'yes']
basement: ['no' 'yes']
hotwaterheating: ['no' 'yes']
airconditioning: ['yes' 'no']
prefarea: ['yes' 'no']
furnishingstatus: ['furnished' 'semi-furnished' 'unfurnished']
```

[53]:
```
for i in range(len(Housing)):
    for j in binary_obj_cols:
        if Housing[j].iloc[i]=='yes':
            Housing[j].iloc[i]=1
        else:
            Housing[j].iloc[i]=0
# Housing['mainroad']=='yes'
```

```
/tmp/ipykernel_12463/2840574408.py:4: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  Housing[j].iloc[i]=1
/tmp/ipykernel_12463/2840574408.py:6: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  Housing[j].iloc[i]=0
```

[54]:
```
Housing
```

[54]:

	price	area	bedrooms	bathrooms	stories	mainroad	guestroom	basement	\
0	13300000	7420	4	2	3	1	0	0	
1	12250000	8960	4	4	4	1	0	0	
2	12250000	9960	3	2	2	1	0	1	
3	12215000	7500	4	2	2	1	0	1	
4	11410000	7420	4	1	2	1	1	1	
..	…	…	…	…	…	…	…	…	
540	1820000	3000	2	1	1	1	0	1	
541	1767150	2400	3	1	1	0	0	0	
542	1750000	3620	2	1	1	1	0	0	
543	1750000	2910	3	1	1	0	0	0	
544	1750000	3850	3	1	2	1	0	0	

	hotwaterheating	airconditioning	parking	prefarea	furnishingstatus
0	0	1	2	1	furnished
1	0	1	3	0	furnished
2	0	0	2	1	semi-furnished
3	0	1	3	1	furnished
4	0	1	2	0	furnished
..	…	…	…	…	…
540	0	0	2	0	unfurnished
541	0	0	0	0	semi-furnished
542	0	0	0	0	unfurnished
543	0	0	0	0	furnished

```
544               0                0         0         0      unfurnished

[545 rows x 13 columns]
```

[55]:
```
Housing.isnull().sum()
```

[55]:
```
price             0
area              0
bedrooms          0
bathrooms         0
stories           0
mainroad          0
guestroom         0
basement          0
hotwaterheating   0
airconditioning   0
parking           0
prefarea          0
furnishingstatus  0
dtype: int64
```

[56]:
```
for i in binary_obj_cols:
    Housing[i]=pd.to_numeric(Housing[i])
```

[57]:
```
Housing.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 545 entries, 0 to 544
Data columns (total 13 columns):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   price             545 non-null    int64
 1   area              545 non-null    int64
 2   bedrooms          545 non-null    int64
 3   bathrooms         545 non-null    int64
 4   stories           545 non-null    int64
 5   mainroad          545 non-null    int64
 6   guestroom         545 non-null    int64
 7   basement          545 non-null    int64
 8   hotwaterheating   545 non-null    int64
 9   airconditioning   545 non-null    int64
 10  parking           545 non-null    int64
 11  prefarea          545 non-null    int64
 12  furnishingstatus  545 non-null    object
dtypes: int64(12), object(1)
memory usage: 55.5+ KB
```

[ ]:

[59]:
```
for i in range(len(Housing)):
    if Housing['furnishingstatus'].iloc[i]=='furnished':
        Housing['furnishingstatus'].iloc[i]=2

    elif Housing['furnishingstatus'].iloc[i]=='semi-furnished':
        Housing['furnishingstatus'].iloc[i]=1

    elif Housing['furnishingstatus'].iloc[i]=='unfurnished':
        Housing['furnishingstatus'].iloc[i]=0

    # '' 'semi-furnished' 'unfurnished'
```

```
/tmp/ipykernel_12463/1222928369.py:3: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  Housing['furnishingstatus'].iloc[i]=2
/tmp/ipykernel_12463/1222928369.py:6: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  Housing['furnishingstatus'].iloc[i]=1
/tmp/ipykernel_12463/1222928369.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  Housing['furnishingstatus'].iloc[i]=0
```

[60]:
```
Housing
```

[60]:

	price	area	bedrooms	bathrooms	stories	mainroad	guestroom	\
0	13300000	7420	4	2	3	1	0	
1	12250000	8960	4	4	4	1	0	
2	12250000	9960	3	2	2	1	0	
3	12215000	7500	4	2	2	1	0	
4	11410000	7420	4	1	2	1	1	
..	…	…	…	…	…	…	…	
540	1820000	3000	2	1	1	1	0	
541	1767150	2400	3	1	1	0	0	
542	1750000	3620	2	1	1	1	0	
543	1750000	2910	3	1	1	0	0	
544	1750000	3850	3	1	2	1	0	

```
     basement  hotwaterheating  airconditioning  parking  prefarea  \
0           0                0                1        2         1
1           0                0                1        3         0
2           1                0                0        2         1
3           1                0                1        3         1
4           1                0                1        2         0
..        …                …                …      …       …
540         1                0                0        2         0
541         0                0                0        0         0
542         0                0                0        0         0
543         0                0                0        0         0
544         0                0                0        0         0

    furnishingstatus
0                  2
1                  2
2                  1
3                  2
4                  2
..               …
540                0
541                1
542                0
543                2
544                0

[545 rows x 13 columns]
```

[62]:
```
Housing['furnishingstatus']=pd.to_numeric(Housing['furnishingstatus'])
```

[64]:
```
Housing.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 545 entries, 0 to 544
Data columns (total 13 columns):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   price             545 non-null    int64
 1   area              545 non-null    int64
 2   bedrooms          545 non-null    int64
 3   bathrooms         545 non-null    int64
 4   stories           545 non-null    int64
 5   mainroad          545 non-null    int64
 6   guestroom         545 non-null    int64
 7   basement          545 non-null    int64
 8   hotwaterheating   545 non-null    int64
```

```
 9   airconditioning   545 non-null    int64
 10  parking           545 non-null    int64
 11  prefarea          545 non-null    int64
 12  furnishingstatus  545 non-null    int64
dtypes: int64(13)
memory usage: 55.5 KB
```

[65]:
```
Housing.corr()
```

[65]:

	price	area	bedrooms	bathrooms	stories	mainroad
price	1.000000	0.535997	0.366494	0.517545	0.420712	0.296898
area	0.535997	1.000000	0.151858	0.193820	0.083996	0.288874
bedrooms	0.366494	0.151858	1.000000	0.373930	0.408564	-0.012033
bathrooms	0.517545	0.193820	0.373930	1.000000	0.326165	0.042398
stories	0.420712	0.083996	0.408564	0.326165	1.000000	0.121706
mainroad	0.296898	0.288874	-0.012033	0.042398	0.121706	1.000000
guestroom	0.255517	0.140297	0.080549	0.126469	0.043538	0.092337
basement	0.187057	0.047417	0.097312	0.102106	-0.172394	0.044002
hotwaterheating	0.093073	-0.009229	0.046049	0.067159	0.018847	-0.011781
airconditioning	0.452954	0.222393	0.160603	0.186915	0.293602	0.105423
parking	0.384394	0.352980	0.139270	0.177496	0.045547	0.204433
prefarea	0.329777	0.234779	0.079023	0.063472	0.044425	0.199876
furnishingstatus	0.304721	0.171445	0.123244	0.143559	0.104672	0.156726

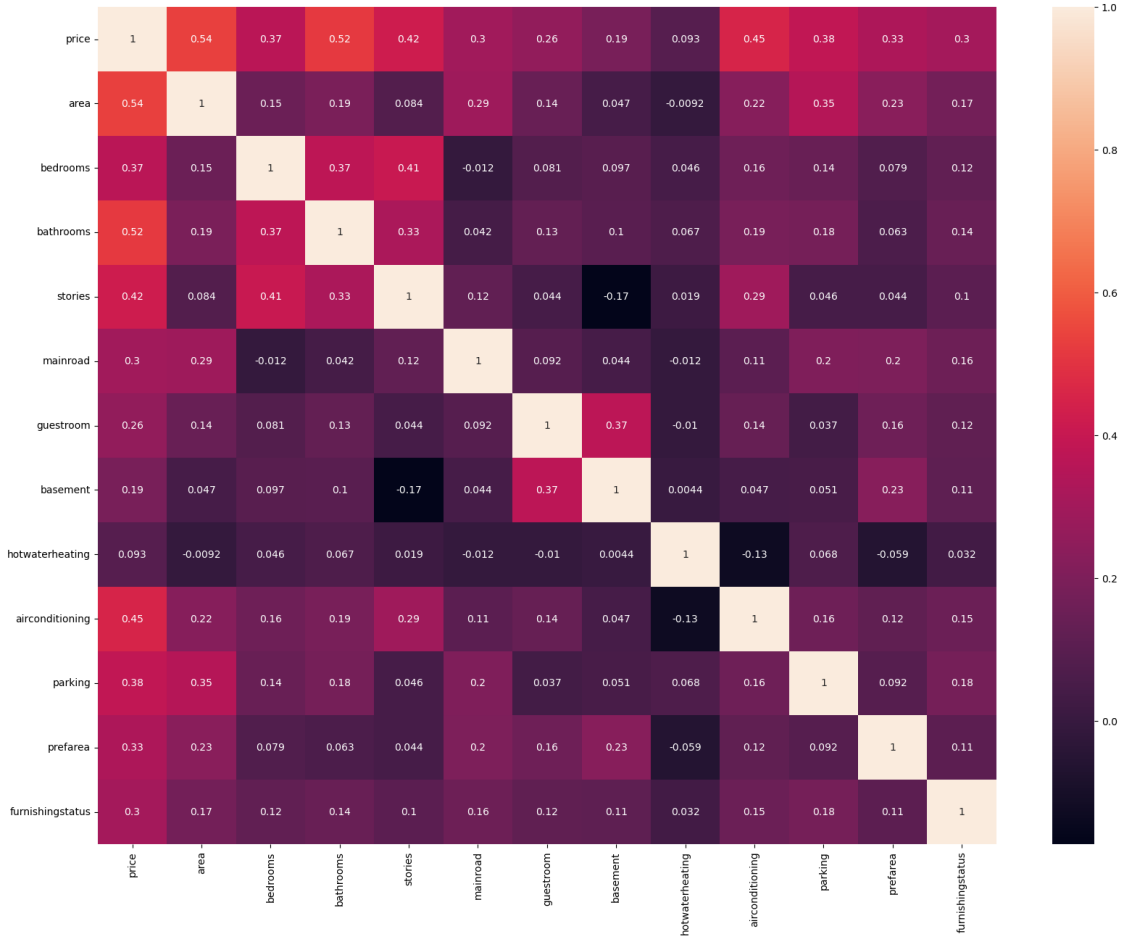
	guestroom	basement	hotwaterheating	airconditioning
price	0.255517	0.187057	0.093073	0.452954
area	0.140297	0.047417	-0.009229	0.222393
bedrooms	0.080549	0.097312	0.046049	0.160603
bathrooms	0.126469	0.102106	0.067159	0.186915
stories	0.043538	-0.172394	0.018847	0.293602
mainroad	0.092337	0.044002	-0.011781	0.105423
guestroom	1.000000	0.372066	-0.010308	0.138179
basement	0.372066	1.000000	0.004385	0.047341
hotwaterheating	-0.010308	0.004385	1.000000	-0.130023
airconditioning	0.138179	0.047341	-0.130023	1.000000
parking	0.037466	0.051497	0.067864	0.159173
prefarea	0.160897	0.228083	-0.059411	0.117382
furnishingstatus	0.118328	0.112831	0.031628	0.150477

	parking	prefarea	furnishingstatus
price	0.384394	0.329777	0.304721
area	0.352980	0.234779	0.171445
bedrooms	0.139270	0.079023	0.123244
bathrooms	0.177496	0.063472	0.143559
stories	0.045547	0.044425	0.104672
mainroad	0.204433	0.199876	0.156726
guestroom	0.037466	0.160897	0.118328

```
basement           0.051497  0.228083          0.112831
hotwaterheating    0.067864 -0.059411          0.031628
airconditioning    0.159173  0.117382          0.150477
parking            1.000000  0.091627          0.177539
prefarea           0.091627  1.000000          0.107686
furnishingstatus   0.177539  0.107686          1.000000
```

[67]:
```
plt.figure(figsize=(20,15))
sns.heatmap(Housing.corr(),annot=True)
```

[67]: <Axes: >



11.0.1 Which features seem to influence house prices the most?

Answer: Area seem to influence house price the most

[72]:
```
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
```

[70]:
```
X=Housing.drop('price',axis=1)
```

[71]:
```
y=Housing['price']
```

[78]:
```
X_train,X_test,y_train,y_test=train_test_split(X,y,random_state=100,test_size=0.
↪3)
```

[79]:
```
lr_model=LinearRegression()
```

[80]:
```
lr_model.fit(X_train,y_train)
```

[80]:
```
LinearRegression()
```

[81]:
```
pred=lr_model.predict(X_test)
```

[98]:
```
from sklearn.metrics import mean_absolute_error
```

[99]:
```
mean_absolute_error(y_true=y_test,y_pred=pred)
```

[99]:
```
832364.898688936
```

11.0.2 How accurate are the model's predictions?

Answer: Models predictions seem to be off by large number.

11.0.3 Can you identify any limitations in the model?

Answer: Model might be overfitting.

11.0.4 What could be done to improve the model's accuracy or address its limitations?

Answer: Ensemble techniques can be used to improve predictions.

[ ]:
```

```