U18ISI6204 – Machine Learning Techniques

LAB EXPERIMENT- 3

Name: Aaron Mathew

Roll no: 20BIS001

Multiple Linear Regression:

import pandas as pd

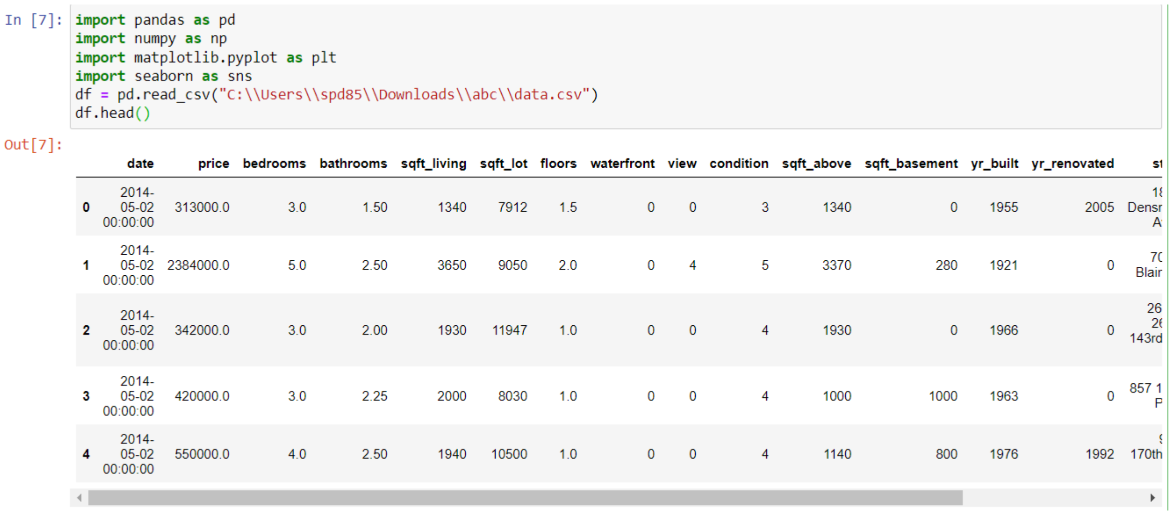
import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns

df = pd.read_csv("C:\\Users\\spd85\\Downloads\\abc\\data.csv")

df.head()



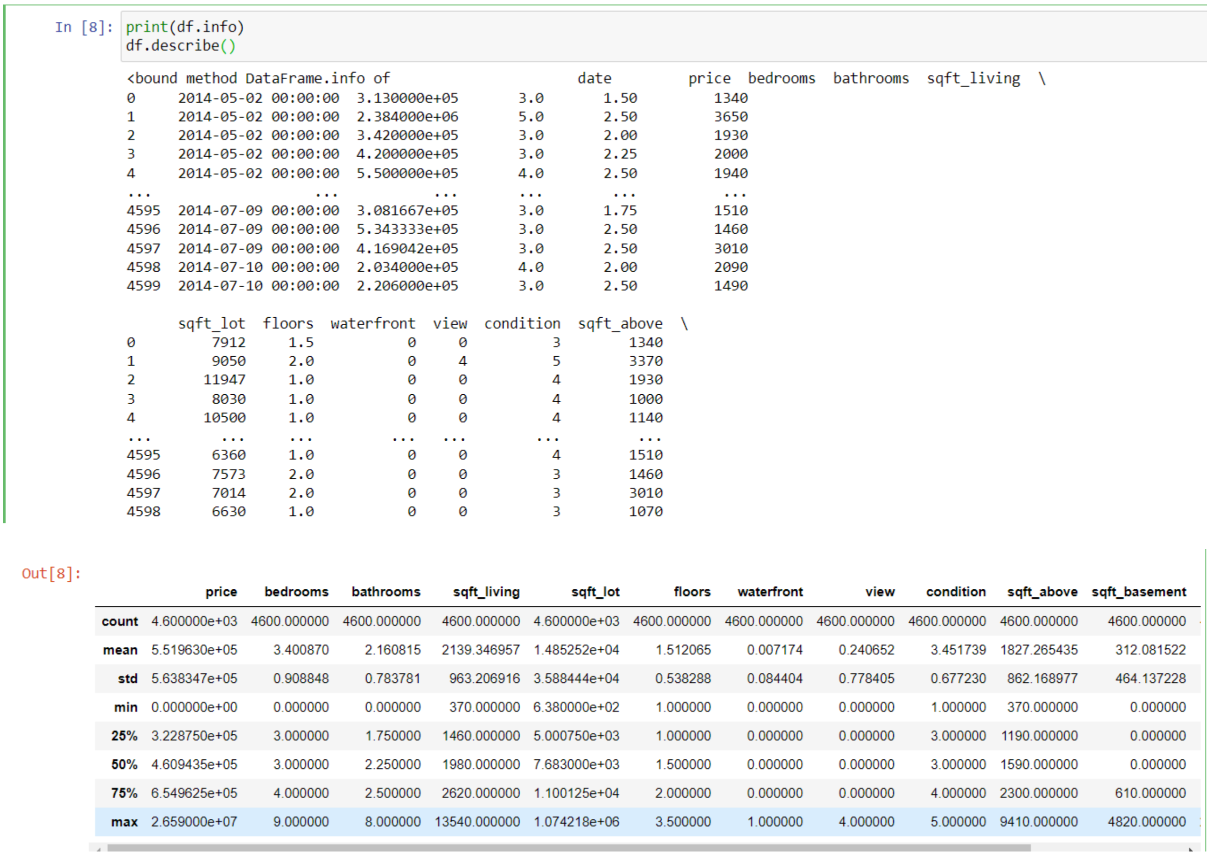
```
In [7]: import pandas as pd
        import numpy as np
        import matplotlib.pyplot as plt
        import seaborn as sns
        df = pd.read_csv("C:\\Users\\spd85\\Downloads\\abc\\data.csv")
        df.head()
```

Out[7]:

	date	price	bedrooms	bathrooms	sqft_living	sqft_lot	floors	waterfront	view	condition	sqft_above	sqft_basement	yr_built	yr_renovated	st
0	2014-05-02 00:00:00	313000.0	3.0	1.50	1340	7912	1.5	0	0	3	1340	0	1955	2005	18 Densr A
1	2014-05-02 00:00:00	2384000.0	5.0	2.50	3650	9050	2.0	0	4	5	3370	280	1921	0	70 Blair
2	2014-05-02 00:00:00	342000.0	3.0	2.00	1930	11947	1.0	0	0	4	1930	0	1966	0	26 26 143rd
3	2014-05-02 00:00:00	420000.0	3.0	2.25	2000	8030	1.0	0	0	4	1000	1000	1963	0	857 1 P
4	2014-05-02 00:00:00	550000.0	4.0	2.50	1940	10500	1.0	0	0	4	1140	800	1976	1992	9 170th

print(df.info)

df.describe()



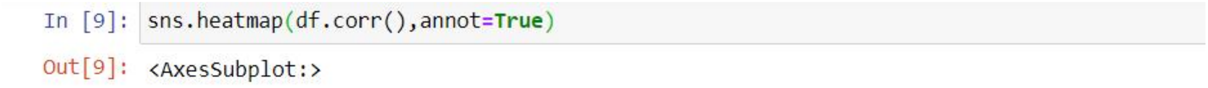
```
In [8]: print(df.info)
        df.describe()
```

```
<bound method DataFrame.info of                date         price  bedrooms  bathrooms  sqft_living  \
0    2014-05-02 00:00:00  3.130000e+05       3.0       1.50         1340
1    2014-05-02 00:00:00  2.384000e+06       5.0       2.50         3650
2    2014-05-02 00:00:00  3.420000e+05       3.0       2.00         1930
3    2014-05-02 00:00:00  4.200000e+05       3.0       2.25         2000
4    2014-05-02 00:00:00  5.500000e+05       4.0       2.50         1940
...                  ...           ...       ...        ...          ...
4595 2014-07-09 00:00:00  3.081667e+05       3.0       1.75         1510
4596 2014-07-09 00:00:00  5.343333e+05       3.0       2.50         1460
4597 2014-07-09 00:00:00  4.169042e+05       3.0       2.50         3010
4598 2014-07-10 00:00:00  2.034000e+05       4.0       2.00         2090
4599 2014-07-10 00:00:00  2.206000e+05       3.0       2.50         1490

      sqft_lot  floors  waterfront  view  condition  sqft_above  \
0         7912     1.5           0     0          3        1340
1         9050     2.0           0     4          5        3370
2        11947     1.0           0     0          4        1930
3         8030     1.0           0     0          4        1000
4        10500     1.0           0     0          4        1140
...        ...     ...         ...   ...        ...         ...
4595      6360     1.0           0     0          4        1510
4596      7573     2.0           0     0          3        1460
4597      7014     2.0           0     0          3        3010
4598      6630     1.0           0     0          3        1070
```

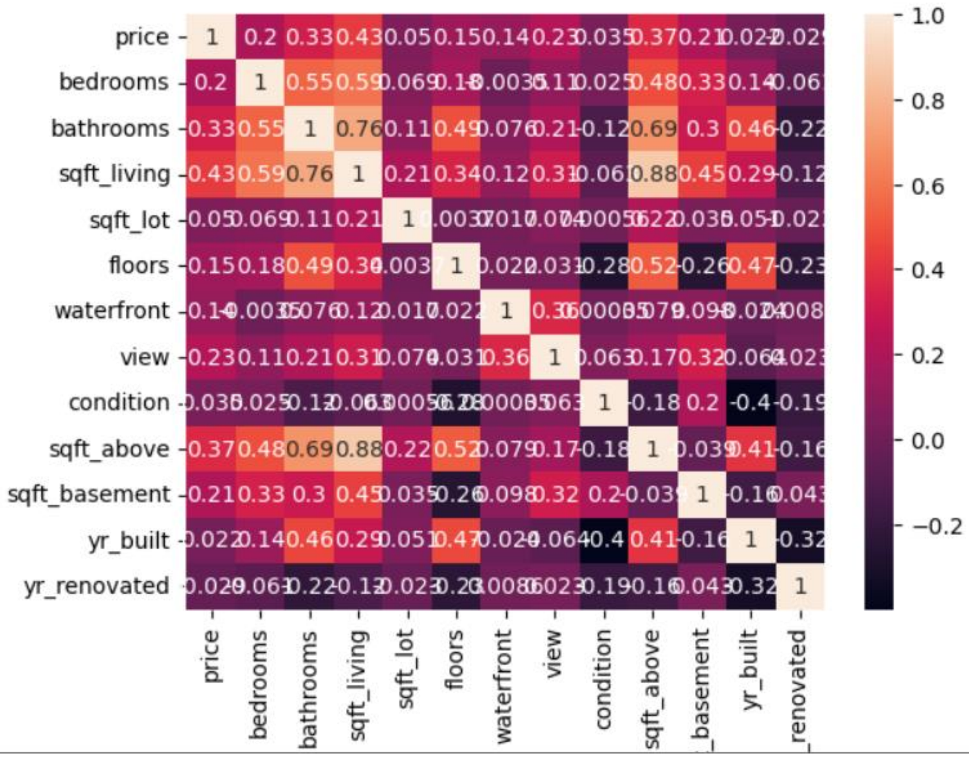
Out[8]:

	price	bedrooms	bathrooms	sqft_living	sqft_lot	floors	waterfront	view	condition	sqft_above	sqft_basement
count	4.600000e+03	4600.000000	4600.000000	4600.000000	4.600000e+03	4600.000000	4600.000000	4600.000000	4600.000000	4600.000000	4600.000000
mean	5.519630e+05	3.400870	2.160815	2139.346957	1.485252e+04	1.512065	0.007174	0.240652	3.451739	1827.265435	312.081522
std	5.638347e+05	0.908848	0.783781	963.206916	3.588444e+04	0.538288	0.084404	0.778405	0.677230	862.168977	464.137228
min	0.000000e+00	0.000000	0.000000	370.000000	6.380000e+02	1.000000	0.000000	0.000000	1.000000	370.000000	0.000000
25%	3.228750e+05	3.000000	1.750000	1460.000000	5.000750e+03	1.000000	0.000000	0.000000	3.000000	1190.000000	0.000000
50%	4.609435e+05	3.000000	2.250000	1980.000000	7.683000e+03	1.500000	0.000000	0.000000	3.000000	1590.000000	0.000000
75%	6.549625e+05	4.000000	2.500000	2620.000000	1.100125e+04	2.000000	0.000000	0.000000	4.000000	2300.000000	610.000000
max	2.659000e+07	9.000000	8.000000	13540.000000	1.074218e+06	3.500000	1.000000	4.000000	5.000000	9410.000000	4820.000000

sns.heatmap(df.corr(),annot=True)

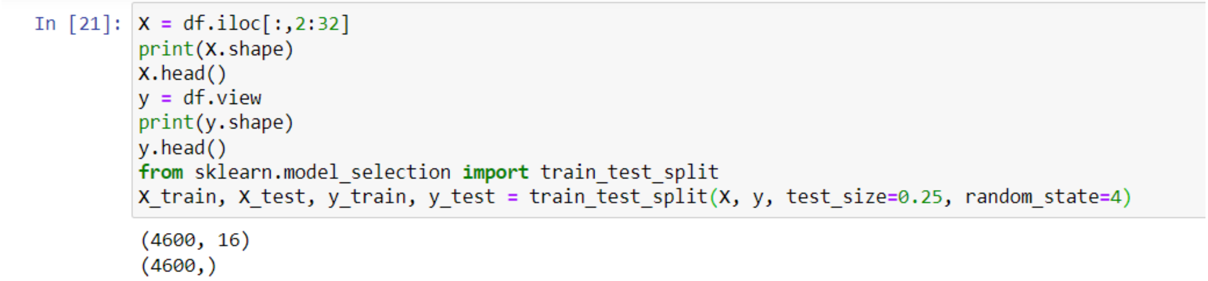


```
In [9]: sns.heatmap(df.corr(),annot=True)
Out[9]: <AxesSubplot:>
```



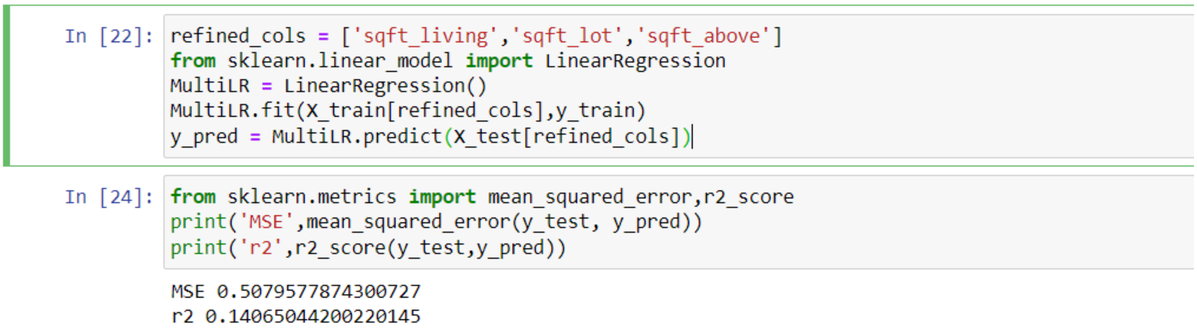
X = df.iloc[:,2:32]

```
print(X.shape)

X.head()

y = df.view

print(y.shape)

y.head()

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=4)
```



```
In [21]: X = df.iloc[:,2:32]
         print(X.shape)
         X.head()
         y = df.view
         print(y.shape)
         y.head()
         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=4)

         (4600, 16)
         (4600,)
```

```
refined_cols = ['sqft_living','sqft_lot','sqft_above']

from sklearn.linear_model import LinearRegression

MultiLR = LinearRegression()

MultiLR.fit(X_train[refined_cols],y_train)

y_pred = MultiLR.predict(X_test[refined_cols])

from sklearn.metrics import mean_squared_error,r2_score

print('MSE',mean_squared_error(y_test, y_pred))

print('r2',r2_score(y_test,y_pred))
```



```
In [22]: refined_cols = ['sqft_living','sqft_lot','sqft_above']
         from sklearn.linear_model import LinearRegression
         MultiLR = LinearRegression()
         MultiLR.fit(X_train[refined_cols],y_train)
         y_pred = MultiLR.predict(X_test[refined_cols])

In [24]: from sklearn.metrics import mean_squared_error,r2_score
         print('MSE',mean_squared_error(y_test, y_pred))
         print('r2',r2_score(y_test,y_pred))

         MSE 0.5079577874300727
         r2 0.14065044200220145
```