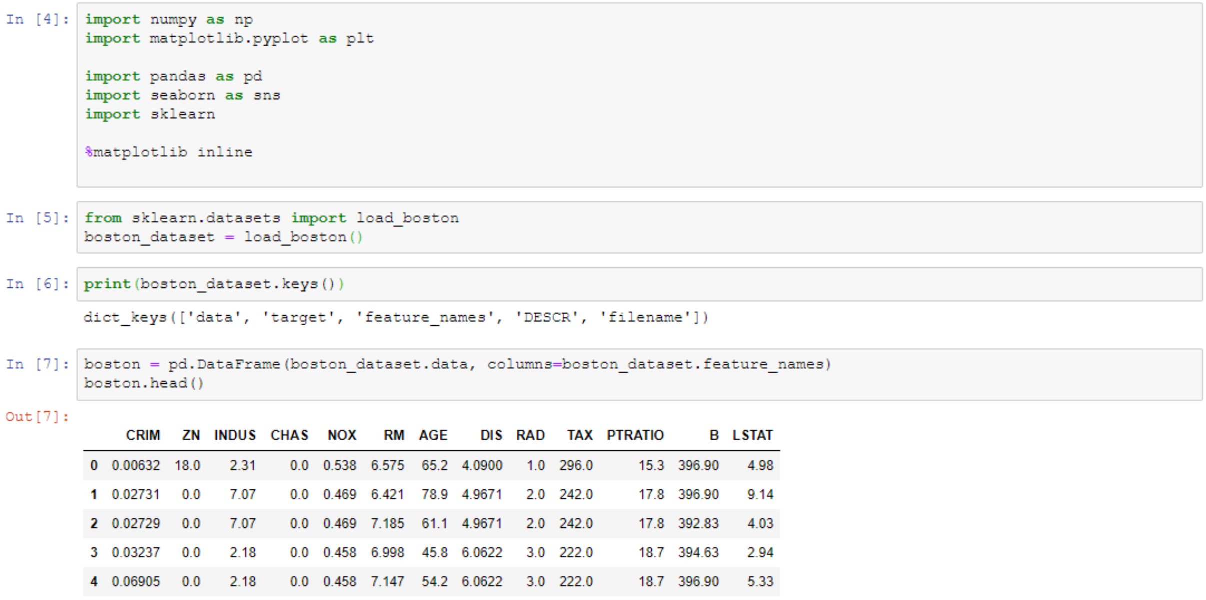
گزارش تمرین دوم داده کاوی

۱–۵:



```
In [4]: import numpy as np
        import matplotlib.pyplot as plt

        import pandas as pd
        import seaborn as sns
        import sklearn

        %matplotlib inline
```

```
In [5]: from sklearn.datasets import load_boston
        boston_dataset = load_boston()
```

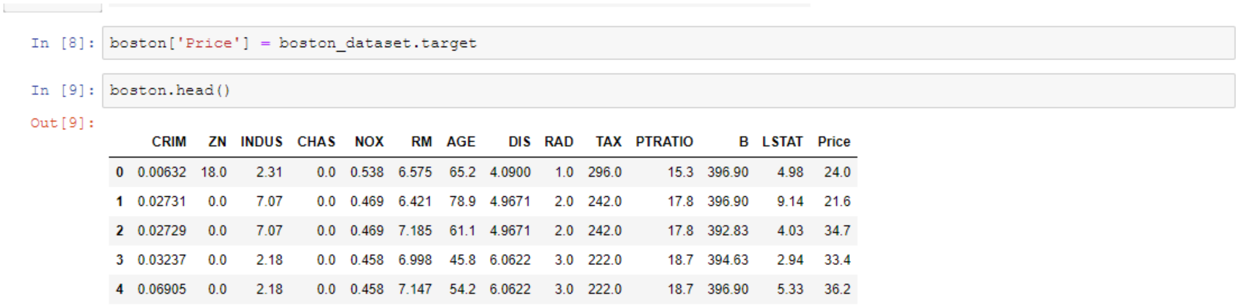
```
In [6]: print(boston_dataset.keys())

        dict_keys(['data', 'target', 'feature_names', 'DESCR', 'filename'])
```

```
In [7]: boston = pd.DataFrame(boston_dataset.data, columns=boston_dataset.feature_names)
        boston.head()
```

Out[7]:

	CRIM	ZN	INDUS	CHAS	NOX	RM	AGE	DIS	RAD	TAX	PTRATIO	B	LSTAT
0	0.00632	18.0	2.31	0.0	0.538	6.575	65.2	4.0900	1.0	296.0	15.3	396.90	4.98
1	0.02731	0.0	7.07	0.0	0.469	6.421	78.9	4.9671	2.0	242.0	17.8	396.90	9.14
2	0.02729	0.0	7.07	0.0	0.469	7.185	61.1	4.9671	2.0	242.0	17.8	392.83	4.03
3	0.03237	0.0	2.18	0.0	0.458	6.998	45.8	6.0622	3.0	222.0	18.7	394.63	2.94
4	0.06905	0.0	2.18	0.0	0.458	7.147	54.2	6.0622	3.0	222.0	18.7	396.90	5.33

۲–۵:



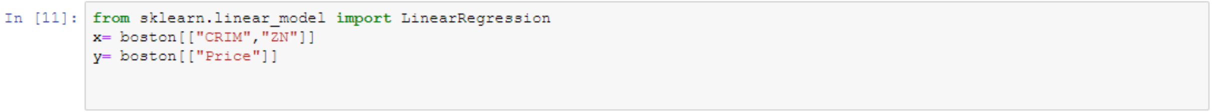
```
In [8]: boston['Price'] = boston_dataset.target
```

```
In [9]: boston.head()
```

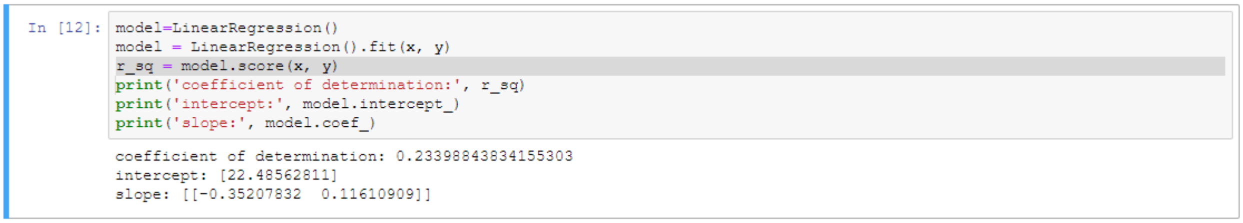
Out[9]:

	CRIM	ZN	INDUS	CHAS	NOX	RM	AGE	DIS	RAD	TAX	PTRATIO	B	LSTAT	Price
0	0.00632	18.0	2.31	0.0	0.538	6.575	65.2	4.0900	1.0	296.0	15.3	396.90	4.98	24.0
1	0.02731	0.0	7.07	0.0	0.469	6.421	78.9	4.9671	2.0	242.0	17.8	396.90	9.14	21.6
2	0.02729	0.0	7.07	0.0	0.469	7.185	61.1	4.9671	2.0	242.0	17.8	392.83	4.03	34.7
3	0.03237	0.0	2.18	0.0	0.458	6.998	45.8	6.0622	3.0	222.0	18.7	394.63	2.94	33.4
4	0.06905	0.0	2.18	0.0	0.458	7.147	54.2	6.0622	3.0	222.0	18.7	396.90	5.33	36.2

۳–۵:

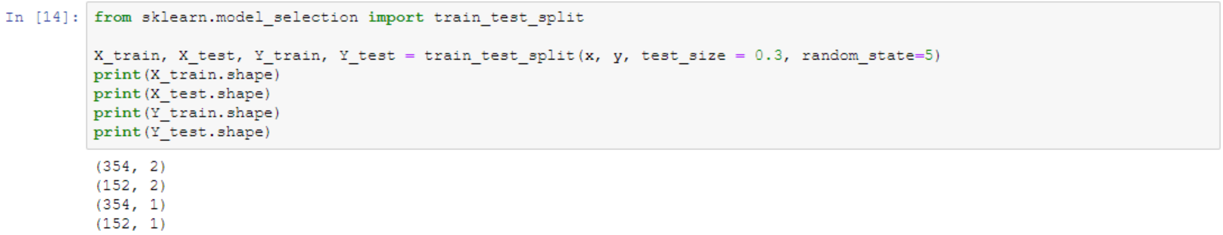


```
In [11]: from sklearn.linear_model import LinearRegression
         x= boston[["CRIM","ZN"]]
         y= boston[["Price"]]
```



```
In [12]: model=LinearRegression()
         model = LinearRegression().fit(x, y)
         r_sq = model.score(x, y)
         print('coefficient of determination:', r_sq)
         print('intercept:', model.intercept_)
         print('slope:', model.coef_)

         coefficient of determination: 0.23398843834155303
         intercept: [22.48562811]
         slope: [[-0.35207832  0.11610909]]
```

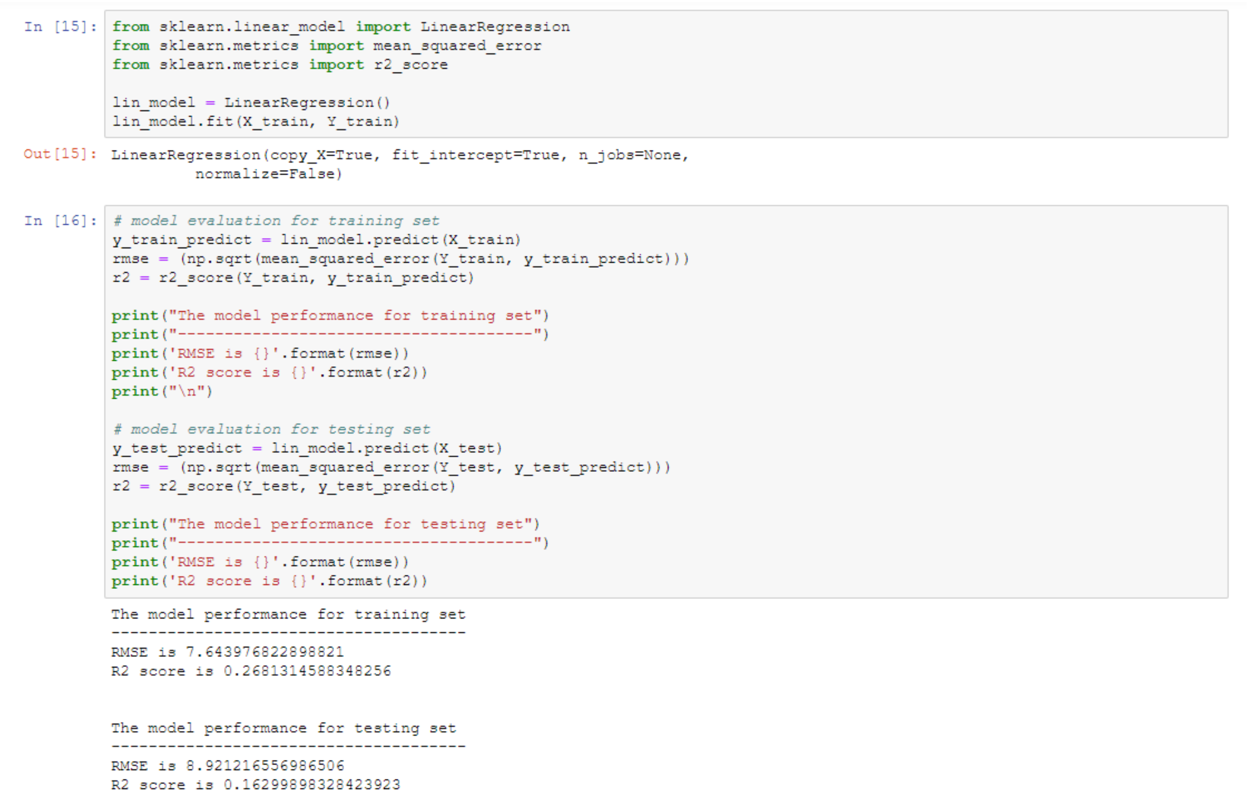
۴–۵:



```
In [14]: 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=5)
         print(X_train.shape)
         print(X_test.shape)
         print(Y_train.shape)
         print(Y_test.shape)

         (354, 2)
         (152, 2)
         (354, 1)
         (152, 1)
```

۵–۵: و

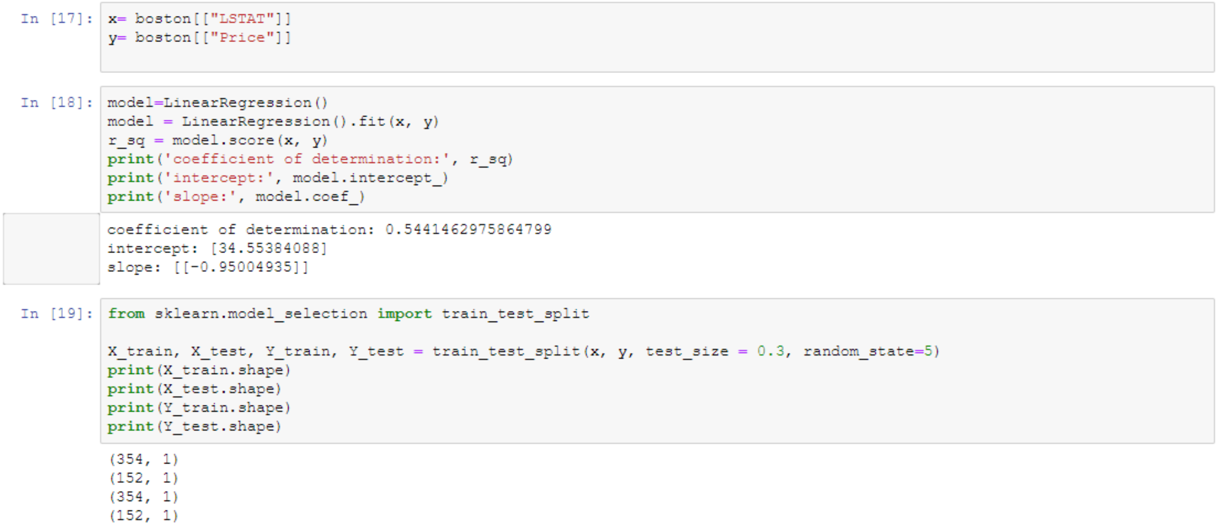
۵–۶:



```
In [15]: from sklearn.linear_model import LinearRegression
         from sklearn.metrics import mean_squared_error
         from sklearn.metrics import r2_score

         lin_model = LinearRegression()
         lin_model.fit(X_train, Y_train)

Out[15]: LinearRegression(copy_X=True, fit_intercept=True, n_jobs=None,
                  normalize=False)
```

```
In [16]: # model evaluation for training set
         y_train_predict = lin_model.predict(X_train)
         rmse = (np.sqrt(mean_squared_error(Y_train, y_train_predict)))
         r2 = r2_score(Y_train, y_train_predict)

         print("The model performance for training set")
         print("--------------------------------------")
         print('RMSE is {}'.format(rmse))
         print('R2 score is {}'.format(r2))
         print("\n")

         # model evaluation for testing set
         y_test_predict = lin_model.predict(X_test)
         rmse = (np.sqrt(mean_squared_error(Y_test, y_test_predict)))
         r2 = r2_score(Y_test, y_test_predict)

         print("The model performance for testing set")
         print("--------------------------------------")
         print('RMSE is {}'.format(rmse))
         print('R2 score is {}'.format(r2))

         The model performance for training set
         --------------------------------------
         RMSE is 7.643976822898821
         R2 score is 0.2681314588348256


         The model performance for testing set
         --------------------------------------
         RMSE is 8.921216556986506
         R2 score is 0.16299898328423923
```

۵–۷:

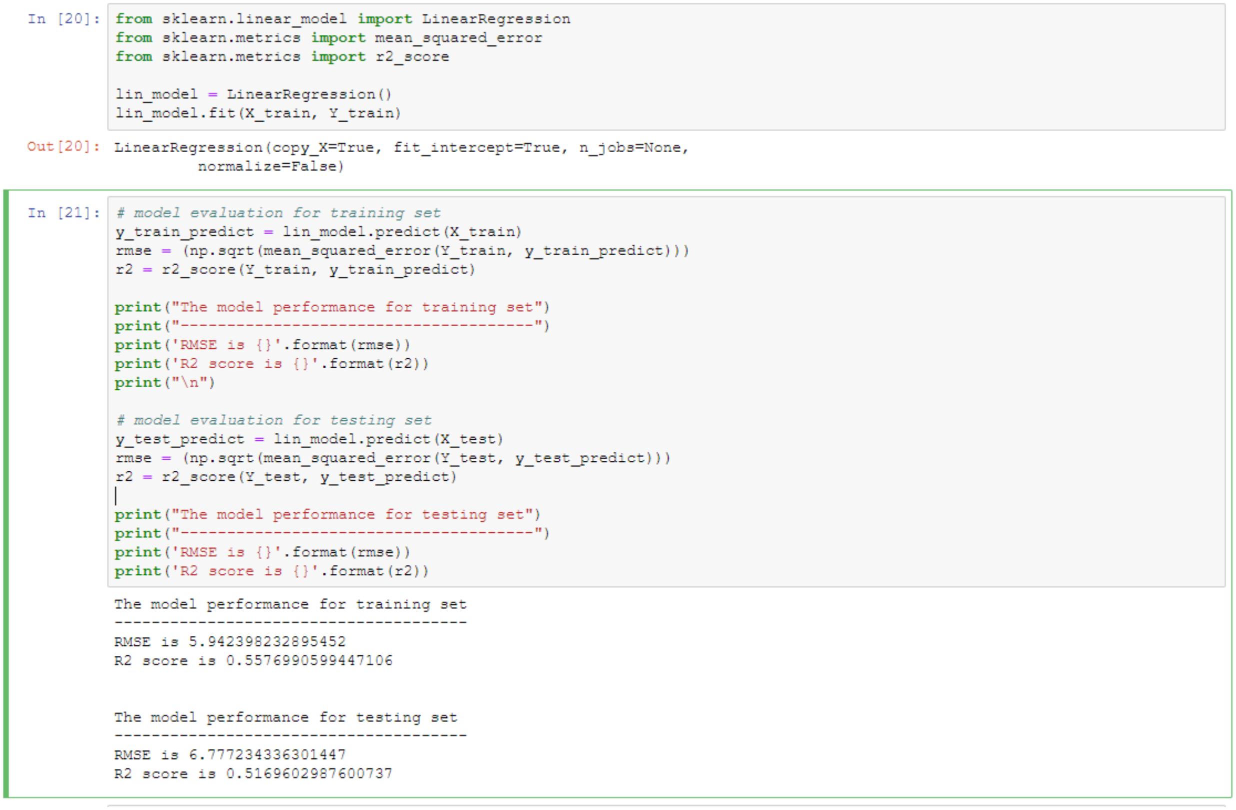


```
In [17]: x= boston[["LSTAT"]]
         y= boston[["Price"]]
```

```
In [18]: model=LinearRegression()
         model = LinearRegression().fit(x, y)
         r_sq = model.score(x, y)
         print('coefficient of determination:', r_sq)
         print('intercept:', model.intercept_)
         print('slope:', model.coef_)

         coefficient of determination: 0.5441462975864799
         intercept: [34.55384088]
         slope: [[-0.95004935]]
```

```
In [19]: 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=5)
         print(X_train.shape)
         print(X_test.shape)
         print(Y_train.shape)
         print(Y_test.shape)

         (354, 1)
         (152, 1)
         (354, 1)
         (152, 1)
```

۵–۸:



```
In [20]: from sklearn.linear_model import LinearRegression
         from sklearn.metrics import mean_squared_error
         from sklearn.metrics import r2_score

         lin_model = LinearRegression()
         lin_model.fit(X_train, Y_train)

Out[20]: LinearRegression(copy_X=True, fit_intercept=True, n_jobs=None,
                  normalize=False)
```

```
In [21]: # model evaluation for training set
         y_train_predict = lin_model.predict(X_train)
         rmse = (np.sqrt(mean_squared_error(Y_train, y_train_predict)))
         r2 = r2_score(Y_train, y_train_predict)

         print("The model performance for training set")
         print("--------------------------------------")
         print('RMSE is {}'.format(rmse))
         print('R2 score is {}'.format(r2))
         print("\n")

         # model evaluation for testing set
         y_test_predict = lin_model.predict(X_test)
         rmse = (np.sqrt(mean_squared_error(Y_test, y_test_predict)))
         r2 = r2_score(Y_test, y_test_predict)

         print("The model performance for testing set")
         print("--------------------------------------")
         print('RMSE is {}'.format(rmse))
         print('R2 score is {}'.format(r2))
```

```
The model performance for training set
--------------------------------------
RMSE is 5.942398232895452
R2 score is 0.5576990599447106


The model performance for testing set
--------------------------------------
RMSE is 6.777234336301447
R2 score is 0.5169602987600737
```

۹–۵: