multiple-linear-regression

November 20, 2024

[1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import statsmodels.formula.api as smf
from sklearn.preprocessing import LabelEncoder
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
import warnings
warnings.filterwarnings('ignore')
df=pd.read_csv('/content/ToyotaCorolla - MLR.csv')
df
```

[1]:

	Price	Age_08_04	KM	Fuel_Type	HP	Automatic	cc	Doors	\
0	13500	23	46986	Diesel	90	0	2000	3	
1	13750	23	72937	Diesel	90	0	2000	3	
2	13950	24	41711	Diesel	90	0	2000	3	
3	14950	26	48000	Diesel	90	0	2000	3	
4	13750	30	38500	Diesel	90	0	2000	3	
…	…	…	…	…	…	…	…	…	
1431	7500	69	20544	Petrol	86	0	1300	3	
1432	10845	72	19000	Petrol	86	0	1300	3	
1433	8500	71	17016	Petrol	86	0	1300	3	
1434	7250	70	16916	Petrol	86	0	1300	3	
1435	6950	76	1	Petrol	110	0	1600	5	

	Cylinders	Gears	Weight
0	4	5	1165
1	4	5	1165
2	4	5	1165
3	4	5	1165
4	4	5	1170
…	…	…	…
1431	4	5	1025
1432	4	5	1015
1433	4	5	1015
1434	4	5	1015

```
1435          4      5    1114

[1436 rows x 11 columns]
```

[2]:
```
df.head()
```

[2]:
```
    Price  Age_08_04     KM Fuel_Type  HP  Automatic    cc  Doors  Cylinders  \
0  13500         23  46986    Diesel  90          0  2000      3          4
1  13750         23  72937    Diesel  90          0  2000      3          4
2  13950         24  41711    Diesel  90          0  2000      3          4
3  14950         26  48000    Diesel  90          0  2000      3          4
4  13750         30  38500    Diesel  90          0  2000      3          4

   Gears  Weight
0      5    1165
1      5    1165
2      5    1165
3      5    1165
4      5    1170
```

[3]:
```
df.shape
```

[3]:
```
(1436, 11)
```

[4]:
```
df.size
```

[4]:
```
15796
```

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

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1436 entries, 0 to 1435
Data columns (total 11 columns):
 #   Column      Non-Null Count  Dtype
---  ------      --------------  -----
 0   Price       1436 non-null   int64
 1   Age_08_04   1436 non-null   int64
 2   KM          1436 non-null   int64
 3   Fuel_Type   1436 non-null   object
 4   HP          1436 non-null   int64
 5   Automatic   1436 non-null   int64
 6   cc          1436 non-null   int64
 7   Doors       1436 non-null   int64
 8   Cylinders   1436 non-null   int64
 9   Gears       1436 non-null   int64
 10  Weight      1436 non-null   int64
dtypes: int64(10), object(1)
```

```
memory usage: 123.5+ KB
```

[6]:
```
df.describe()
```

[6]:
```
              Price    Age_08_04             KM           HP    Automatic  \
count   1436.000000  1436.000000    1436.000000  1436.000000  1436.000000
mean   10730.824513    55.947075   68533.259749   101.502089     0.055710
std     3626.964585    18.599988   37506.448872    14.981080     0.229441
min     4350.000000     1.000000       1.000000    69.000000     0.000000
25%     8450.000000    44.000000   43000.000000    90.000000     0.000000
50%     9900.000000    61.000000   63389.500000   110.000000     0.000000
75%    11950.000000    70.000000   87020.750000   110.000000     0.000000
max    32500.000000    80.000000  243000.000000   192.000000     1.000000

                cc        Doors  Cylinders        Gears      Weight
count   1436.00000  1436.000000     1436.0  1436.000000  1436.00000
mean    1576.85585     4.033426        4.0     5.026462  1072.45961
std      424.38677     0.952677        0.0     0.188510    52.64112
min     1300.00000     2.000000        4.0     3.000000  1000.00000
25%     1400.00000     3.000000        4.0     5.000000  1040.00000
50%     1600.00000     4.000000        4.0     5.000000  1070.00000
75%     1600.00000     5.000000        4.0     5.000000  1085.00000
max    16000.00000     5.000000        4.0     6.000000  1615.00000
```

[7]:
```
df.isnull().sum()
```

[7]:
```
Price        0
Age_08_04    0
KM           0
Fuel_Type    0
HP           0
Automatic    0
cc           0
Doors        0
Cylinders    0
Gears        0
Weight       0
dtype: int64
```

[8]:
```
df.duplicated().sum()
```

[8]:
```
1
```

[9]:
```
df1=df.drop_duplicates(ignore_index=True)
df1
```

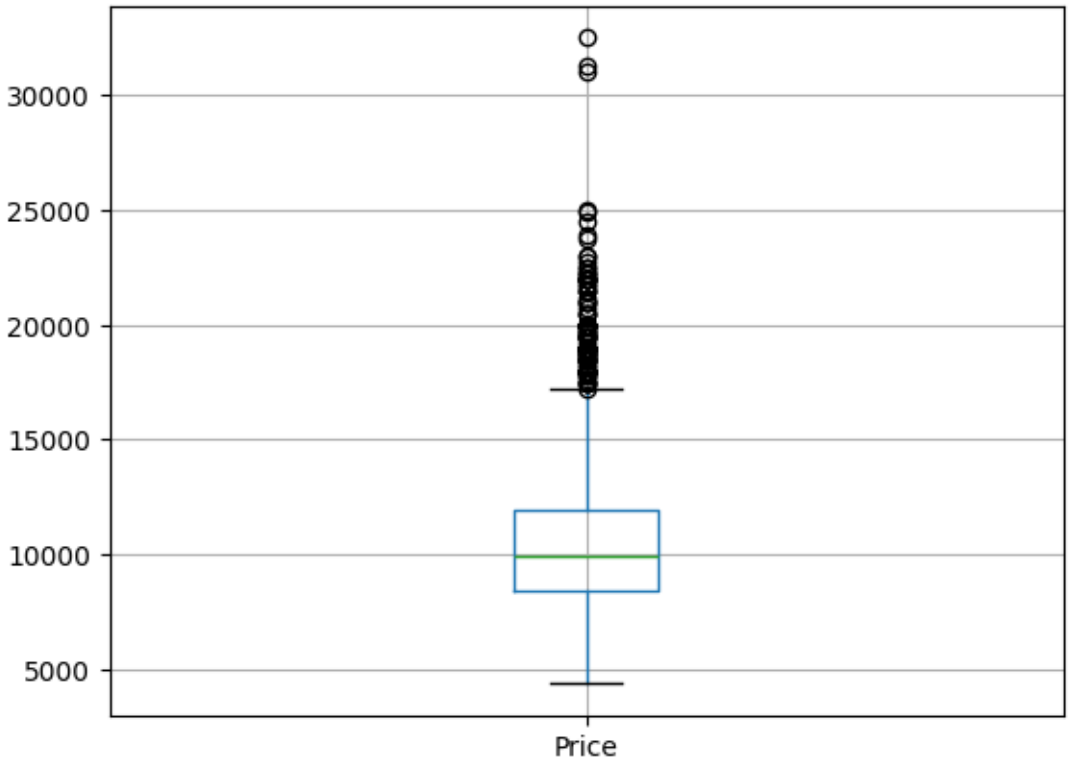
[9]:
```
      Price  Age_08_04     KM Fuel_Type   HP  Automatic    cc  Doors  \
0     13500         23  46986    Diesel   90          0  2000      3
1     13750         23  72937    Diesel   90          0  2000      3
2     13950         24  41711    Diesel   90          0  2000      3
3     14950         26  48000    Diesel   90          0  2000      3
4     13750         30  38500    Diesel   90          0  2000      3
…       …          …      …         …  …         …   …     …
1430   7500         69  20544    Petrol   86          0  1300      3
1431  10845         72  19000    Petrol   86          0  1300      3
1432   8500         71  17016    Petrol   86          0  1300      3
1433   7250         70  16916    Petrol   86          0  1300      3
1434   6950         76      1    Petrol  110          0  1600      5

      Cylinders  Gears  Weight
0             4      5    1165
1             4      5    1165
2             4      5    1165
3             4      5    1165
4             4      5    1170
…           …    …      …
1430          4      5    1025
1431          4      5    1015
1432          4      5    1015
1433          4      5    1015
1434          4      5    1114

[1435 rows x 11 columns]
```

[12]:
```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
<ipython-input-12-49b3fcfeb4d1> in <cell line: 1>()
----> 1 df1.corr()

/usr/local/lib/python3.10/dist-packages/pandas/core/frame.py in corr(self,␣
 ↪method, min_periods, numeric_only)
  11047         cols = data.columns
  11048         idx = cols.copy()
> 11049         mat = data.to_numpy(dtype=float, na_value=np.nan, copy=False)
  11050
  11051         if method == "pearson":

/usr/local/lib/python3.10/dist-packages/pandas/core/frame.py in to_numpy(self,␣
 ↪dtype, copy, na_value)
   1991         if dtype is not None:
   1992             dtype = np.dtype(dtype)
```

```
-> 1993         result = self._mgr.as_array(dtype=dtype, copy=copy,␣
 ↪na_value=na_value)
   1994         if result.dtype is not dtype:
   1995             result = np.asarray(result, dtype=dtype)

/usr/local/lib/python3.10/dist-packages/pandas/core/internals/managers.py in␣
 ↪as_array(self, dtype, copy, na_value)
   1692                 arr.flags.writeable = False
   1693         else:
-> 1694             arr = self._interleave(dtype=dtype, na_value=na_value)
   1695             # The underlying data was copied within _interleave, so no␣
 ↪need
   1696             # to further copy if copy=True or setting na_value

/usr/local/lib/python3.10/dist-packages/pandas/core/internals/managers.py in␣
 ↪_interleave(self, dtype, na_value)
   1751             else:
   1752                 arr = blk.get_values(dtype)
-> 1753             result[rl.indexer] = arr
   1754             itemmask[rl.indexer] = 1
   1755

ValueError: could not convert string to float: 'Diesel'
```

[11]:
```
df1.boxplot('Price')
```

[11]: <Axes: >

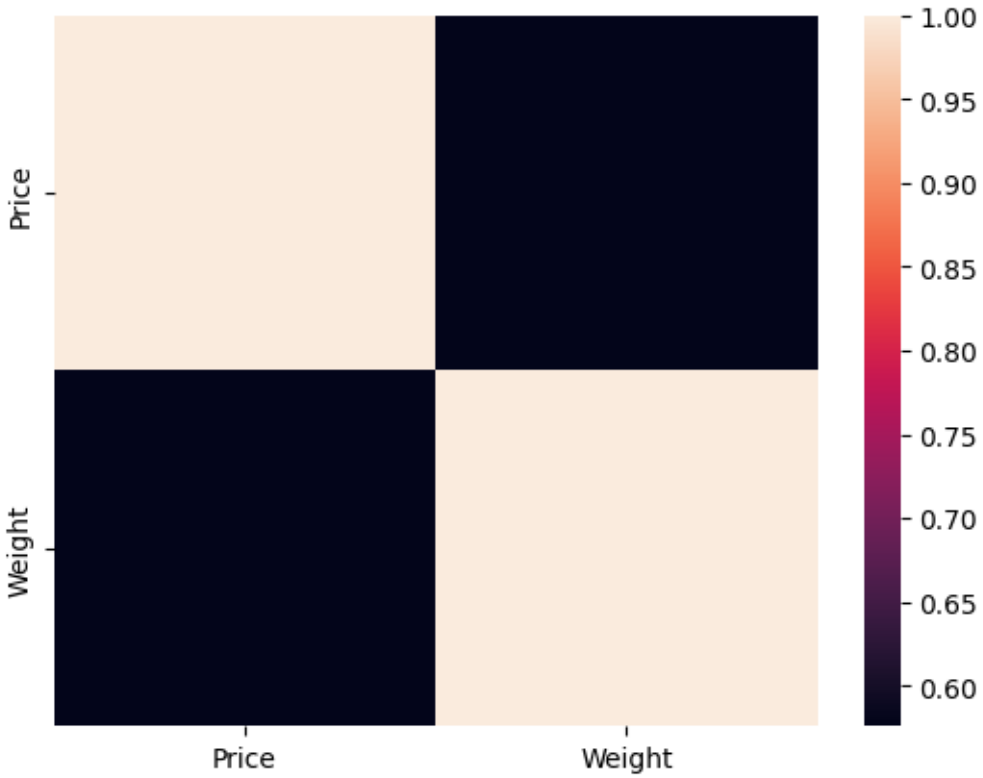


[13]:
```
df1[['Price','Weight']].corr()
```

[13]:
```
            Price    Weight
Price    1.000000  0.575869
Weight   0.575869  1.000000
```

[14]:
```
sns.heatmap(df1[['Price','Weight']].corr())
```

[14]:
```
<Axes: >
```



[15]:
```
lab_encoder=LabelEncoder()
```

[16]:
```
df1['Fuel_Type']=lab_encoder.fit_transform(df1['Fuel_Type'])
```

[17]:
```
df1
```

[17]:

	Price	Age_08_04	KM	Fuel_Type	HP	Automatic	cc	Doors	\
0	13500	23	46986	1	90	0	2000	3	
1	13750	23	72937	1	90	0	2000	3	
2	13950	24	41711	1	90	0	2000	3	
3	14950	26	48000	1	90	0	2000	3	
4	13750	30	38500	1	90	0	2000	3	
…	…	…	…	…	…	…	…	…	
1430	7500	69	20544	2	86	0	1300	3	
1431	10845	72	19000	2	86	0	1300	3	
1432	8500	71	17016	2	86	0	1300	3	
1433	7250	70	16916	2	86	0	1300	3	
1434	6950	76	1	2	110	0	1600	5	

	Cylinders	Gears	Weight
0	4	5	1165
1	4	5	1165

```
2              4      5    1165
3              4      5    1165
4              4      5    1170
…            …      …      …
1430           4      5    1025
1431           4      5    1015
1432           4      5    1015
1433           4      5    1015
1434           4      5    1114

[1435 rows x 11 columns]
```

[19]:
```
target=df1[['Price']]
features=df1.drop('Price',axis=1)
x_train,x_test,y_train,y_test=train_test_split(features,target,train_size=0.
 ↪8,random_state=100)
print(x_train.shape)
print(y_train.shape)
print(x_test.shape)
print(y_test.shape)
```

```
(1148, 10)
(1148, 1)
(287, 10)
(287, 1)
```

[20]:
```
linear_model=LinearRegression()
linear_model.fit(x_train,y_train)
```

[20]:
```
LinearRegression()
```

[21]:
```
y_pred=linear_model.predict(x_test)
y_pred
```

[21]:
```
array([[15542.62555409],
       [10795.00465001],
       [13121.14886739],
       [10471.29701081],
       [ 8059.24993951],
       [ 9484.92082893],
       [13895.11035986],
       [ 6392.05054479],
       [13278.78914353],
       [ 8487.15330411],
       [13250.37560992],
       [ 7743.19431421],
       [11526.23520905],
```

```
[ 5001.18363973],
[18082.98097044],
[ 8297.25467271],
[13567.20064033],
[15555.1406489 ],
[16633.93885677],
[ 9369.18652309],
[12436.20906731],
[10838.29480074],
[10870.59653511],
[11879.29740799],
[ 8315.44486122],
[ 8290.58158432],
[ 9812.63806864],
[ 8449.38280687],
[ 9450.70618018],
[ 8070.2821265 ],
[ 9088.08901167],
[12020.70509149],
[ 6557.66900243],
[19087.60517921],
[ 9480.02273032],
[ 9438.4901857 ],
[ 7186.42954035],
[ 8087.33601898],
[14178.41596104],
[10673.41201977],
[ 7670.48529843],
[ 8000.31948339],
[ 9794.86276846],
[13881.21684855],
[ 8502.81737528],
[ 9231.54490419],
[12562.57336001],
[ 6908.56259324],
[10998.84248341],
[12805.62081138],
[10040.08994053],
[12899.35937552],
[12707.24799107],
[14527.71216811],
[19496.62798265],
[11049.66765095],
[20498.88334658],
[ 6614.12055927],
[14268.18873955],
[ 9912.19981627],
```

```
[19109.00520418],
[ 6569.38973246],
[13160.77846335],
[11722.20208598],
[11312.17451325],
[17927.78838396],
[10117.63355645],
[11885.26348995],
[12093.26769379],
[ 8511.63123049],
[10880.45875496],
[ 6135.03578677],
[10036.49818128],
[18897.59854111],
[ 9314.19897839],
[ 8576.69177179],
[ 9873.73335372],
[13359.69004884],
[10276.12652011],
[11784.07110213],
[ 8467.48795121],
[10972.88227241],
[12941.68206085],
[ 9866.10118673],
[16649.6700893 ],
[12968.75944146],
[ 9066.4629487 ],
[ 8797.72265605],
[11736.75970386],
[ 7862.62876124],
[ 6954.56976839],
[ 8528.35313457],
[ 7069.37487216],
[11220.32741742],
[10291.50365879],
[ 9545.15297581],
[11185.84521338],
[13432.07106437],
[10598.59778996],
[10833.47003453],
[11975.65727557],
[ 8931.6941927 ],
[ 8386.40351779],
[ 8696.68001957],
[ 8105.42160139],
[ 8149.07728054],
[ 8474.02797664],
```

```
[ 8399.04474864],
[ 7300.08449028],
[ 8199.02397717],
[16124.8261578 ],
[ 8293.56748731],
[ 8495.70949317],
[10987.52085253],
[ 8416.76539611],
[15199.2693106 ],
[ 7972.33009628],
[12979.68372702],
[ 6509.28846651],
[ 8458.76257734],
[ 7634.37523556],
[18234.58203075],
[18115.25281904],
[16681.9623425 ],
[ 7120.47657759],
[10602.8075349 ],
[19492.73498628],
[ 8436.85101267],
[ 9398.10911287],
[16453.51937117],
[12758.39332378],
[ 9314.57262147],
[16514.98715293],
[14118.39365049],
[ 9830.8457914 ],
[ 9615.84614333],
[ 7490.90848997],
[10544.2756757 ],
[11196.636541  ],
[ 8946.65861546],
[ 9726.65693061],
[ 7618.34087033],
[10315.62447878],
[16407.86157254],
[10887.29651929],
[11416.91393994],
[ 6437.46742098],
[10590.31245082],
[11462.81571542],
[13276.50443233],
[ 9768.14923102],
[ 7884.26145166],
[14081.13376439],
[ 8046.17151275],
```

```
[11609.14232819],
[ 6501.16598695],
[ 6040.87103431],
[ 9255.65739445],
[19066.24040964],
[ 7637.54249985],
[15978.29933659],
[ 7609.23072848],
[ 9372.85583284],
[ 9607.93591697],
[13316.97402354],
[12374.11480159],
[11067.55026113],
[ 7971.93230922],
[11627.23989629],
[ 8596.45136754],
[10471.16346446],
[11245.37372826],
[13589.91865235],
[17045.18966544],
[ 8997.11630002],
[ 8567.93914466],
[10979.32226525],
[ 7818.1070394 ],
[ 8854.62953914],
[17130.36164771],
[ 9598.53582638],
[18031.06568674],
[10458.33253838],
[ 9208.93830277],
[11347.18383474],
[10695.08410395],
[10161.30583273],
[10824.86152919],
[ 5429.29100841],
[12689.00271931],
[10033.52937273],
[10485.33041173],
[12335.7791817 ],
[15289.84774742],
[10762.36143488],
[13003.34394039],
[14000.76506786],
[ 9910.18743249],
[12372.70711738],
[ 7006.97202158],
[10068.6455025 ],
```

```
[10058.40923472],
[ 9616.12555789],
[13449.20285254],
[ 8341.04977618],
[ 8125.67689062],
[11505.44455513],
[ 7997.59909666],
[11371.27654038],
[13584.39031097],
[13079.57506093],
[11559.20091986],
[ 9377.73825005],
[14360.02841828],
[ 8691.62652089],
[ 9638.42628436],
[10870.75488975],
[14354.36851372],
[13061.06542679],
[12238.65883743],
[ 7382.40935632],
[ 7440.40030917],
[ 9592.48406358],
[12926.1583226 ],
[ 7369.04893556],
[11260.56521996],
[ 9394.57436322],
[15597.73142591],
[ 8457.32996299],
[10211.49588286],
[10787.87808683],
[ 8709.3190788 ],
[ 6022.47616893],
[10687.72246929],
[14898.74128797],
[11030.67247991],
[ 8511.72274044],
[ 9646.67192106],
[ 6948.68478194],
[ 9234.03916501],
[ 8716.43826656],
[ 5860.66815422],
[ 6266.81507289],
[ 9304.03589563],
[ 6281.46294168],
[ 9557.24800381],
[11161.55475723],
[10979.22411776],
```

```
       [ 9170.92887028],
       [ 7209.66877866],
       [ 8273.56579879],
       [10613.64066946],
       [11772.15026334],
       [ 6900.44604329],
       [10126.03119616],
       [ 8535.70904471],
       [10052.22979065],
       [11375.4997035 ],
       [ 8910.644635  ],
       [12237.51345486],
       [ 9844.34202688],
       [ 9274.24036563],
       [ 8861.45529273],
       [16536.64129187],
       [ 7674.88539265],
       [11503.84698914],
       [ 8062.5127525 ],
       [ 6886.8303812 ],
       [ 9258.32123447],
       [11027.30632312],
       [10993.93143234],
       [ 9273.54644331],
       [ 7362.83512412],
       [13059.38457248],
       [ 9854.70621881],
       [ 9671.92963234],
       [ 7468.77406276],
       [ 9175.1231995 ],
       [17703.73569406],
       [11931.9458539 ],
       [ 6271.003884  ],
       [ 8025.28303298],
       [26136.19057591],
       [12656.08053759],
       [16262.88652499],
       [11705.43298529],
       [11427.04450319]])
```

[22]:
```
linear_model.score(x_train,y_train)
```

[22]:
```
0.8626130385692874
```

[23]:
```
linear_model.score(x_test,y_test)
```

[23]:
```
0.8671580177926699
```

[24]:
```
linear_model.intercept_
```

[24]:
```
array([-8689.50713129])
```

[25]:
```
linear_model.coef_
```

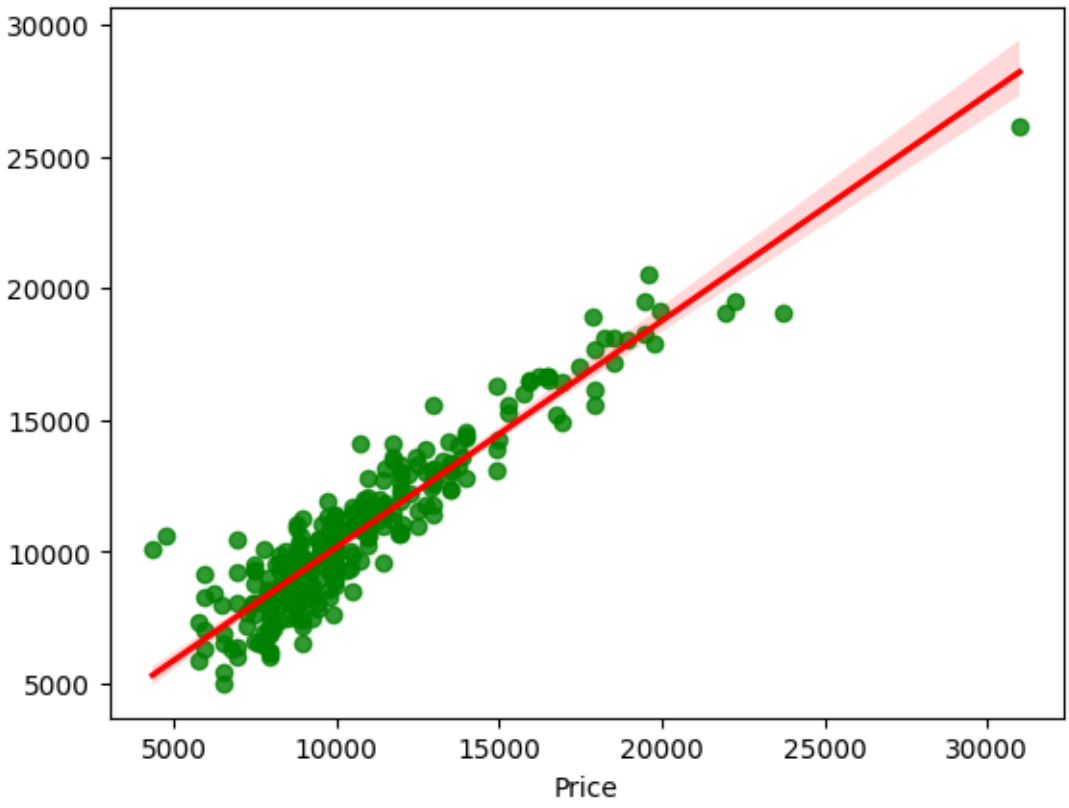
[25]:
```
array([[-1.24537995e+02, -1.76276977e-02,  3.95421517e+02,
         2.62373819e+01,  3.94335343e+02, -3.00403510e-02,
         4.30425267e+00,  3.06954462e-12,  7.33844905e+02,
         1.91383362e+01]])
```

[26]:
```
from sklearn.metrics import r2_score
r2_score(y_test,y_pred)
```

[26]:
```
0.8671580177926699
```

[27]:
```
sns.regplot(data=df1,x=y_test,y=y_pred,scatter_kws={'color':
 ↪'green'},line_kws={'color':'red'})
```

[27]:
```
<Axes: xlabel='Price'>
```



[28]:
```
import statsmodels.formula.api as smf
model1=smf.
  ↪ols('Price~Age_08_04+KM+Fuel_Type+HP+cc+Doors+Gears+Weight',data=df1).fit()
```

[29]:
```
model1.summary()
```

[29]:

Dep. Variable:	Price	**R-squared:**	0.863
Model:	OLS	**Adj. R-squared:**	0.863
Method:	Least Squares	**F-statistic:**	1127.
Date:	Wed, 20 Nov 2024	**Prob (F-statistic):**	0.00
Time:	11:23:44	**Log-Likelihood:**	-12362.
No. Observations:	1435	**AIC:**	2.474e+04
Df Residuals:	1426	**BIC:**	2.479e+04
Df Model:	8		
Covariance Type:	nonrobust		

	coef	**std err**	**t**	**P> |t|**	**[0.025**	**0.975]**
Intercept	-9852.8426	1478.988	-6.662	0.000	-1.28e+04	-6951.617
Age_08_04	-123.2082	2.608	-47.237	0.000	-128.325	-118.092
KM	-0.0177	0.001	-13.494	0.000	-0.020	-0.015
Fuel_Type	617.0386	145.592	4.238	0.000	331.441	902.637
HP	23.2994	2.872	8.114	0.000	17.666	28.932
cc	-0.0418	0.090	-0.465	0.642	-0.218	0.134
Doors	-35.7894	40.222	-0.890	0.374	-114.690	43.112
Gears	582.2973	196.381	2.965	0.003	197.070	967.525
Weight	20.9344	1.014	20.653	0.000	18.946	22.923

Omnibus:	270.225	**Durbin-Watson:**	1.608
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	2870.669
Skew:	-0.545	**Prob(JB):**	0.00
Kurtosis:	9.843	**Cond. No.**	3.29e+06

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 3.29e+06. This might indicate that there are strong multicollinearity or other numerical problems.

[30]:
```
model2=smf.ols('Price~cc',data=df1).fit()
```

[31]:
```
model2.summary()
```

[31]:

Dep. Variable:	Price	**R-squared:**	0.015
Model:	OLS	**Adj. R-squared:**	0.015
Method:	Least Squares	**F-statistic:**	22.52
Date:	Wed, 20 Nov 2024	**Prob (F-statistic):**	2.29e-06
Time:	11:23:58	**Log-Likelihood:**	-13779.
No. Observations:	1435	**AIC:**	2.756e+04
Df Residuals:	1433	**BIC:**	2.757e+04
Df Model:	1		
Covariance Type:	nonrobust		

	coef	**std err**	**t**	**P> |t|**	**[0.025**	**0.975]**
Intercept	9053.5368	363.894	24.880	0.000	8339.715	9767.359
cc	1.0576	0.223	4.745	0.000	0.620	1.495

Omnibus:	463.846	**Durbin-Watson:**	0.269
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	1386.822
Skew:	1.645	**Prob(JB):**	7.17e-302
Kurtosis:	6.518	**Cond. No.**	6.28e+03

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 6.28e+03. This might indicate that there are strong multicollinearity or other numerical problems.

[32]:
```
model3=smf.ols('Price~Doors',data=df1).fit()
model3.summary()
```

[32]:

Dep. Variable:	Price	**R-squared:**	0.034
Model:	OLS	**Adj. R-squared:**	0.033
Method:	Least Squares	**F-statistic:**	49.99
Date:	Wed, 20 Nov 2024	**Prob (F-statistic):**	2.40e-12
Time:	11:24:10	**Log-Likelihood:**	-13765.
No. Observations:	1435	**AIC:**	2.753e+04
Df Residuals:	1433	**BIC:**	2.755e+04
Df Model:	1		
Covariance Type:	nonrobust		

	coef	**std err**	**t**	**P> |t|**	**[0.025**	**0.975]**
Intercept	7916.1452	407.596	19.422	0.000	7116.596	8715.694
Doors	695.4978	98.366	7.071	0.000	502.541	888.454

Omnibus:	465.543	**Durbin-Watson:**	0.289
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	1403.980
Skew:	1.647	**Prob(JB):**	1.35e-305
Kurtosis:	6.554	**Cond. No.**	19.0

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

[33]:
```
model4=smf.ols('Price~Doors+cc',data=df1).fit()
model4.summary()
```

[33]:

Dep. Variable:	Price	**R-squared:**	0.046
Model:	OLS	**Adj. R-squared:**	0.045
Method:	Least Squares	**F-statistic:**	34.40
Date:	Wed, 20 Nov 2024	**Prob (F-statistic):**	2.55e-15
Time:	11:24:21	**Log-Likelihood:**	-13756.
No. Observations:	1435	**AIC:**	2.752e+04
Df Residuals:	1432	**BIC:**	2.753e+04
Df Model:	2		
Covariance Type:	nonrobust		

	coef	**std err**	**t**	**P> |t|**	**[0.025**	**0.975]**
Intercept	6568.3395	513.700	12.786	0.000	5560.655	7576.024
Doors	662.3187	98.089	6.752	0.000	469.906	854.732
cc	0.9398	0.220	4.268	0.000	0.508	1.372

Omnibus:	448.494	**Durbin-Watson:**	0.291
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	1297.612
Skew:	1.602	**Prob(JB):**	1.69e-282
Kurtosis:	6.382	**Cond. No.**	9.09e+03

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 9.09e+03. This might indicate that there are strong multicollinearity or other numerical problems.

[34]:
```
model4.resid.mean()
```

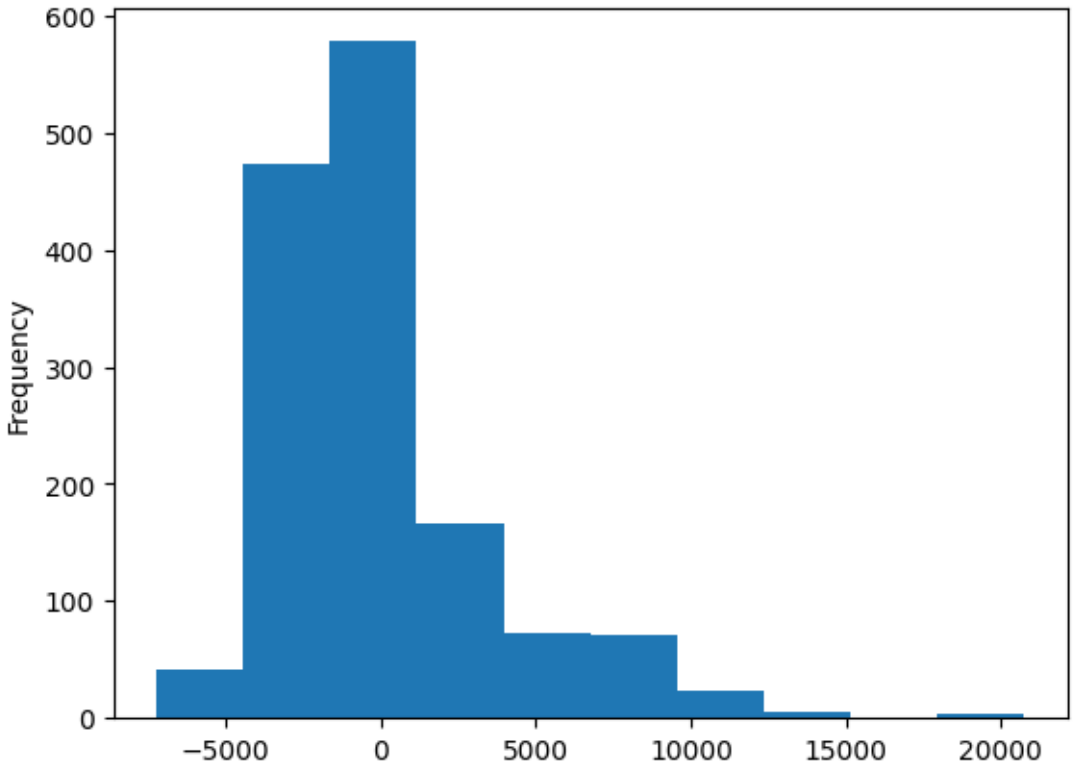
[34]:
```
-3.019497058117431e-10
```

[35]:
```
model4.resid_pearson
```

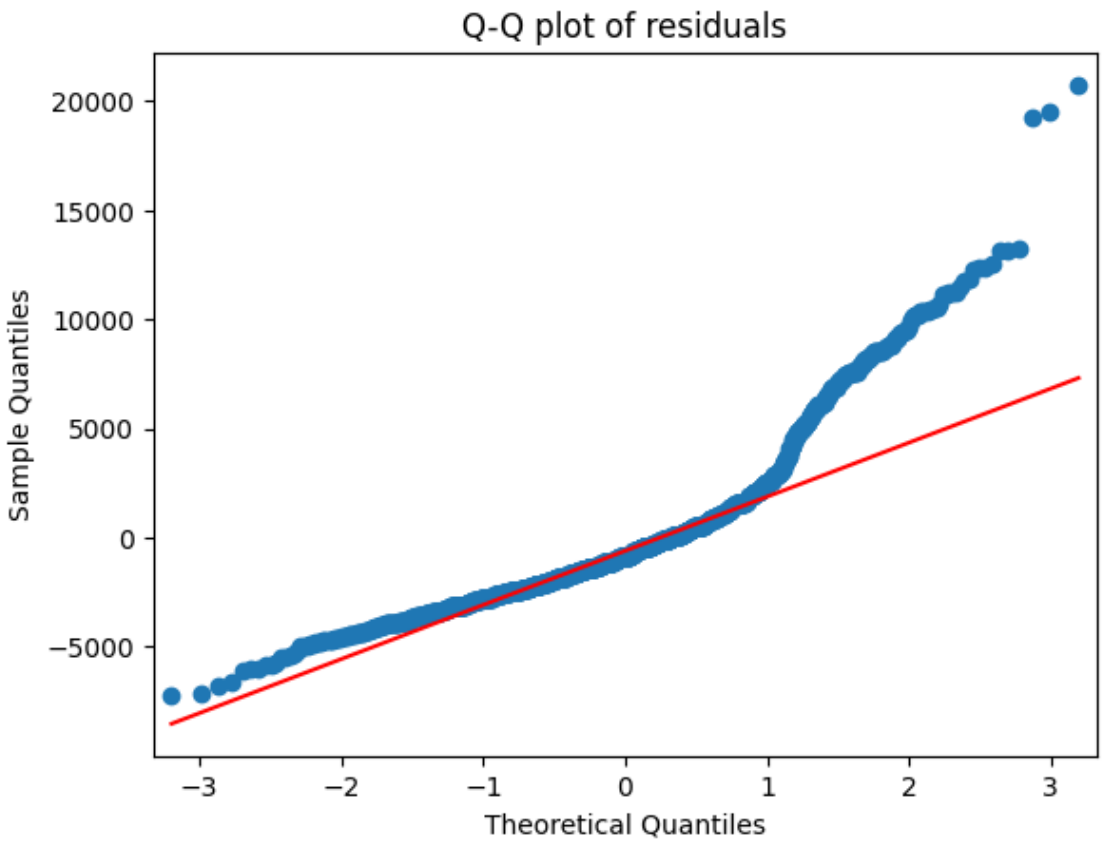
[35]:
```
array([ 0.86893322,  0.93980501,  0.99650243, …, -0.36201304,
       -0.71637196, -1.25685978])
```

[36]:
```
model4.resid.plot(kind='hist')
```

[36]:
```
<Axes: ylabel='Frequency'>
```



[37]:
```
import statsmodels.api as sm
qqplot=sm.qqplot(model4.resid,line='q')
plt.title("Q-Q plot of residuals")
plt.show()
```



[38]:
```
from sklearn.model_selection import train_test_split,GridSearchCV
from sklearn.linear_model import Lasso,Ridge,ElasticNet
from sklearn.metrics import r2_score
lasso=Lasso(alpha=2)
```

[39]:
```
lasso.fit(x_train,y_train)
y_pred=lasso.predict(x_test)
y_pred
```

[39]:
```
array([15535.33357727, 10798.60278209, 13128.65182423, 10472.98259672,
        8061.21665822,  9451.62149365, 13895.88023501,  6388.12445434,
       13277.77884749,  8485.9943513 , 13260.72638017,  7699.81763217,
       11528.34608186,  4984.8824338 , 18084.63657131,  8308.82969744,
       13565.8663738 , 15560.49497551, 16637.24873992,  9382.75511014,
       12445.05192929, 10780.37675906, 10879.21582621, 11884.32081269,
        8321.49461064,  8283.93939524,  9824.64107373,  8456.4439936 ,
        9460.65953027,  8081.38716205,  9085.4409467 , 12026.70510086,
        6561.78365607, 19086.03847221,  9490.95748096,  9448.37435544,
        7192.85479828,  8083.06723788, 14177.52871051, 10680.01531916,
        7682.66711741,  8011.8436919 ,  9803.82916223, 13882.9614978 ,
```

```
 8496.56171973,  9230.52462183, 12564.84546836,  6898.98616673,
11006.06842547, 12815.66095063, 10054.19640552, 12895.05843796,
12703.81638076, 14530.43377176, 19509.83638842, 11058.15551644,
20469.04921123,  6613.98819916, 14279.41121449,  9913.40339175,
19107.55968757,  6572.75538359, 13162.36137399, 11716.20642121,
11279.10454975, 17926.99192625, 10121.47256165, 11890.65977804,
12096.06423286,  8512.94742761, 10877.2817775 ,  6126.78645058,
10044.38684522, 18906.47778314,  9324.19465321,  8564.41080782,
 9874.56037242, 13359.05854363, 10278.33759305, 11795.90258672,
 8473.01418464, 10915.7264125 , 12942.42100131,  9877.59158047,
16643.75514318, 12980.61818731,  9036.08813068,  8798.08441101,
11734.28850085,  7864.72961904,  6963.87921441,  8533.02804597,
 7071.8757258 , 11221.1427306 , 10302.14163506,  9548.93212227,
11193.80365363, 13432.66483793, 10606.02503211, 10836.88920602,
11939.51265795,  8925.42117335,  8389.566278  ,  8702.30818227,
 8056.24623983,  8152.4196945 ,  8477.52824505,  8365.70565543,
 7303.07649649,  8200.07213468, 16134.54856246,  8306.01346864,
 8497.86050736, 10985.68619737,  8426.57108717, 15200.97754743,
 7971.27908647, 12973.39153959,  6508.2368678 ,  8452.55057639,
 7647.08857916, 18233.65414239, 18122.96307466, 16675.41489239,
 7072.75800713, 10601.73435566, 19455.66360318,  8442.20374617,
 9393.21918963, 16455.80752147, 12769.21946363,  9317.0920238 ,
16517.62340117, 14117.16648786,  9842.13653046,  9620.62117504,
 7491.20803401, 10481.43106302, 11195.17074993,  8941.57886035,
 9662.443577  ,  7617.86229495, 10318.95396932, 16409.71279849,
10888.1557966 , 11417.51097904,  6441.71673965, 10603.45798808,
11465.81964122, 13286.76509635,  9776.71813782,  7896.07226151,
14082.14172982,  8009.07732399, 11606.41185889,  6490.33588876,
 6038.79749898,  9265.32154266, 19064.5527119 ,  7639.61879951,
15980.3424046 ,  7610.33131774,  9373.34158068,  9603.28446129,
13331.71784876, 12382.27950156, 11079.96570318,  7980.35981034,
11624.70398339,  8605.72324001, 10482.00349334, 11256.60065229,
13587.32944133, 17049.19774515,  9001.86875281,  8586.06782125,
10990.67327056,  7831.04549571,  8812.70981758, 17134.45069076,
 9614.42644841, 18029.22334205, 10424.50407645,  9205.18514518,
11349.83211993, 10694.85486391, 10125.79528006, 10836.88920602,
 5418.35829305, 12701.64395161, 10038.22365028, 10492.84917152,
12341.93272964, 15282.75502187, 10707.27488776, 13011.24229222,
14003.02801264,  9911.22089413, 12374.7193731 ,  6998.08732074,
10112.7218778 , 10066.09641866,  9617.12447517, 13452.41913818,
 8329.88048334,  8136.35961266, 11511.13602651,  7998.45291889,
11373.21258253, 13585.03130491, 13080.61875016, 11565.52333491,
 9375.99447715, 14361.0643934 ,  8730.13329557,  9640.36687091,
10873.07188772, 14312.51695312, 13060.53225545, 12248.01370974,
 7373.88561326,  7435.97809123,  9602.4249373 , 12927.38665251,
 7374.55206401, 11262.08468491,  9359.78929162, 15591.56689706,
 8449.42402787, 10153.28973125, 10784.17681616,  8715.0188177 ,
```

```
 6031.89729129, 10689.36053259, 14903.72831446, 11028.01966097,
 8507.14827307,  9643.63251107,  6953.7636685 ,  9231.21022992,
 8722.9911962 ,  5867.54218602,  6270.15303582,  9309.27591576,
 6281.48792078,  9553.75724248, 11158.01238094, 10917.07747996,
 9173.4834866 ,  7218.47468359,  8275.93083066, 10616.13236116,
11782.94014344,  6909.00307545, 10122.25798677,  8538.05883772,
10052.04010179, 11386.6481653 ,  8959.27251259, 12240.31149339,
 9852.77324832,  9285.64050723,  8863.93970301, 16530.08625283,
 7674.5960593 , 11501.67482595,  8065.31332696,  6889.62444954,
 9221.9628314 , 11032.57288867, 10937.71015396,  9273.93686461,
 7366.614738  , 13071.09987653,  9856.97688251,  9672.86144088,
 7477.61364294,  9187.59278733, 17708.84868889, 11932.27741441,
 6271.42962048,  8037.89246158, 26147.71166947, 12697.73890157,
16256.41194267, 11701.70913111, 11383.18244336])
```

[40]:
```
r2_score(y_test,y_pred)
```

[40]:
```
0.8672128573315887
```

[41]:
```
lasso.intercept_
```

[41]:
```
array([-8286.44378335])
```

[42]:
```
lasso.coef_
```

[42]:
```
array([-1.24427888e+02, -1.77275250e-02,  3.71285103e+02,  2.66341196e+01,
        3.51346244e+02, -2.99874423e-02,  0.00000000e+00,  0.00000000e+00,
        6.66925493e+02,  1.90995146e+01])
```

[43]:
```
params={'alpha':[1,2,3,4,5,6]}
```

[44]:
```
grid_search=GridSearchCV(lasso,params)
grid_search.fit(x_train,y_train)
```

[44]:
```
GridSearchCV(estimator=Lasso(alpha=2), param_grid={'alpha': [1, 2, 3, 4, 5, 6]})
```

[45]:
```
grid_search.best_params_
```

[45]:
```
{'alpha': 6}
```

[46]:
```
lasso1=Lasso(alpha=6)
```

[47]:
```
lasso1.fit(x_train,y_train)
```

[47]:
```
Lasso(alpha=6)
```

[48]:
```
y_pred1=lasso1.predict(x_test)
```

[49]:
```
y_pred1
```

[49]:
```
array([15526.24858617, 10813.39447402, 13137.41411112, 10483.8982614 ,
        8072.33117303,  9386.19857596, 13904.73978644,  6386.68226839,
       13282.48758289,  8489.83956192, 13272.73795677,  7617.33258868,
       11539.69551447,  4960.90233284, 18095.64508333,  8323.4773102 ,
       13572.2550817 , 15560.67421343, 16633.07425123,  9401.76798123,
       12454.30559568, 10663.65363334, 10890.26160787, 11903.35620476,
        8323.94032091,  8281.00193877,  9840.29837353,  8477.91752653,
        9471.87052615,  8095.00710298,  9086.10459869, 12039.90687935,
        6564.47605595, 19087.15305755,  9504.2892958 ,  9459.43527881,
        7206.42989411,  8081.52487611, 14182.82597822, 10694.17169176,
        7698.31756192,  8026.36481495,  9823.03863613, 13893.10895149,
        8494.47089524,  9234.71128535, 12576.60155855,  6889.63472406,
       11011.7495    , 12827.79283771, 10074.40860065, 12895.00343755,
       12703.26444694, 14543.5434501 , 19535.65455432, 11066.02025183,
       20399.03282969,  6617.62330903, 14293.3625378 ,  9922.43051067,
       19108.93717048,  6573.83110418, 13172.66302782, 11712.15344914,
       11204.73218883, 17929.7042639 , 10136.74614217, 11894.80331832,
       12109.1983059 ,  8523.76086306, 10877.17656368,  6120.2569664 ,
       10061.28278803, 18912.91005149,  9335.4893367 ,  8549.04148931,
        9883.31632368, 13364.86183496, 10290.3769391 , 11811.02874147,
        8484.72030406, 10800.65668295, 12949.52759929,  9892.14500283,
       16637.70609938, 12996.41419189,  8967.68076286,  8806.04702553,
       11740.02326441,  7869.00509233,  6973.27670594,  8537.04146299,
        7076.97014444, 11215.3569176 , 10314.87842224,  9564.31619018,
       11203.4402372 , 13441.11799168, 10630.15201   , 10852.65691638,
       11859.06445247,  8923.31706326,  8402.73779076,  8708.38948144,
        7968.6500502 ,  8153.53440592,  8491.97760691,  8290.85977663,
        7309.24449659,  8209.48312203, 16141.82881254,  8322.27061022,
        8508.85350014, 10988.75759511,  8437.40487693, 15193.26359976,
        7976.14449035, 12969.04131479,  6499.77091162,  8445.48300413,
        7664.15677685, 18236.91616468, 18126.82695928, 16668.00188878,
        6980.86225919, 10609.17408455, 19369.01356214,  8453.53349329,
        9388.78738099, 16449.41659401, 12782.22974778,  9327.34878196,
       16511.98760043, 14121.72638635,  9856.25684314,  9623.29511989,
        7498.96255006, 10354.0532661 , 11199.05044944,  8936.98371635,
        9532.06407013,  7610.69628506, 10333.13851725, 16413.85620037,
       10897.0386166 , 11425.85743606,  6444.69314036, 10621.62628457,
       11464.77344483, 13299.399748  ,  9784.68553093,  7912.01171692,
       14091.52589617,  7930.06512364, 11605.95165182,  6477.43819473,
        6028.09840764,  9275.89704753, 19065.40483278,  7637.97068263,
       15973.39544008,  7606.57472553,  9382.41041796,  9603.90383885,
       13355.14026022, 12392.42854059, 11112.06252328,  7998.30274148,
       11628.38844095,  8615.14242747, 10495.18671915, 11270.39095602,
       13588.91840233, 17046.55410428,  9019.32357216,  8615.68820426,
       11004.71648866,  7848.61015621,  8729.18120709, 17140.54981448,
```

```
 9639.00025811, 18029.68309797, 10348.44567364,  9204.0134756 ,
11362.60395081, 10701.37636997, 10046.08792825, 10852.65691638,
 5407.08290593, 12719.38586494, 10040.7467972 , 10513.96725923,
12355.48383276, 15274.08601093, 10596.66159685, 13018.76052184,
14014.84549624,  9920.4246808 , 12385.90363941,  6985.7106815 ,
10197.20072397, 10072.4483247 ,  9626.23707057, 13466.26406225,
 8316.89832456,  8148.7987114 , 11515.89552451,  7994.18539806,
11382.3257689 , 13593.59502075, 13090.02352163, 11569.28502984,
 9378.35508418, 14370.25272808,  8790.79211732,  9651.51409645,
10885.37352503, 14238.19589507, 13065.98691281, 12258.3618226 ,
 7366.83400464,  7430.39663818,  9613.61699547, 12936.36529471,
 7376.03607699, 11273.51900219,  9291.69719712, 15584.91954464,
 8442.9509746 , 10035.90910169, 10782.93425633,  8721.25538669,
 6042.57154054, 10703.46711444, 14901.53029841, 11031.41800624,
 8508.68818531,  9648.55117691,  6964.41808923,  9229.91363599,
 8745.12400707,  5872.7073019 ,  6266.8995327 ,  9319.31285059,
 6275.6058439 ,  9553.44388918, 11159.49726935, 10795.7558028 ,
 9185.93021436,  7227.88749019,  8287.91753091, 10628.79670615,
11796.36057386,  6917.05501249, 10123.84055697,  8536.77992409,
10059.97100659, 11400.27638856,  9054.00387443, 12253.4569464 ,
 9870.82990338,  9299.96563743,  8863.24191547, 16522.64865941,
 7681.08938166, 11504.04657837,  8078.22859267,  6903.26532109,
 9141.98783513, 11044.1348489 , 10824.65967916,  9267.12368283,
 7382.68264906, 13086.04714627,  9869.12111685,  9674.71468351,
 7496.41798586,  9204.22160621, 17720.10846176, 11939.80763183,
 6271.99811879,  8055.56435568, 26158.52628167, 12793.82621597,
16249.13262916, 11702.8328494 , 11302.46865693])
```

[50]:
```
r2_score(y_test,y_pred)
ridge=Ridge()
params={'alpha':[1,2,3,4,5]}
```

[51]:
```
grid_search=GridSearchCV(ridge,params)
```

[52]:
```
grid_search.fit(x_train,y_train)
grid_search.best_params_
```

[52]:
```
{'alpha': 5}
```

[53]:
```
ridge=Ridge(alpha=5)
ridge.fit(x_train,y_train)
y_pred=ridge.predict(x_test)
y_pred
```

[53]:
```
array([[15537.36894727],
       [10799.65439913],
       [13126.77157123],
```

```
[10474.81670742],
[ 8063.09380891],
[ 9457.74254083],
[13897.74650441],
[ 6388.89177058],
[13280.31834694],
[ 8488.92477436],
[13259.95072944],
[ 7705.73368854],
[11529.85539183],
[ 4989.68165119],
[18085.84219858],
[ 8308.02664543],
[13566.24127883],
[15558.60694186],
[16636.04094327],
[ 9381.12179657],
[12444.98727408],
[10758.06665715],
[10877.09474767],
[11880.83135846],
[ 8322.71634433],
[ 8284.86559612],
[ 9823.55874852],
[ 8456.43120959],
[ 9460.37264508],
[ 8080.77782376],
[ 9088.88136331],
[12027.49743552],
[ 6559.71343184],
[19087.91591327],
[ 9490.30104716],
[ 9448.1143181 ],
[ 7193.82665065],
[ 8083.91225755],
[14179.99281306],
[10680.64644313],
[ 7681.64490451],
[ 8011.08299699],
[ 9803.64506756],
[13884.33999972],
[ 8497.32659476],
[ 9233.34463874],
[12566.2129789 ],
[ 6901.11900896],
[11006.72181676],
[12815.16059571],
```

```
[10052.30907747],
[12896.59598107],
[12707.30687361],
[14531.51996883],
[19509.24001311],
[11058.24996545],
[20449.82099421],
[ 6613.2633082 ],
[14278.22936009],
[ 9915.33772706],
[19109.39009647],
[ 6570.96451782],
[13163.93976146],
[11718.42930706],
[11284.71547068],
[17928.67230769],
[10122.48728974],
[11889.66499953],
[12097.31375549],
[ 8511.71053762],
[10880.82058586],
[ 6128.48349554],
[10044.587047  ],
[18905.39521715],
[ 9323.90266049],
[ 8567.38320578],
[ 9876.72137069],
[13361.4913036 ],
[10279.9904063 ],
[11794.68810982],
[ 8474.22175428],
[10893.1205194 ],
[12944.12560334],
[ 9876.6968739 ],
[16645.18820015],
[12979.38311331],
[ 9040.90841297],
[ 8800.54546093],
[11735.52229474],
[ 7867.26652501],
[ 6960.50076393],
[ 8530.59325954],
[ 7074.32530776],
[11221.91444093],
[10301.53246658],
[ 9550.05561632],
[11191.90508713],
```

```
[13434.63340384],
[10602.33731187],
[10838.21159242],
[11946.90756907],
[ 8926.15883286],
[ 8390.88741808],
[ 8699.50345347],
[ 8063.65058223],
[ 8150.51334817],
[ 8478.79893031],
[ 8371.68764602],
[ 7305.32415153],
[ 8202.32339243],
[16133.24434837],
[ 8304.84372233],
[ 8499.55165463],
[10988.75565234],
[ 8426.42073257],
[15200.48186983],
[ 7974.33368117],
[12975.66790243],
[ 6508.10367945],
[ 8457.3741025 ],
[ 7645.91167116],
[18235.48368445],
[18122.37944677],
[16677.08209651],
[ 7080.34776649],
[10600.87708098],
[19462.11120577],
[ 8443.4786485 ],
[ 9397.43278368],
[16454.99624384],
[12768.30460137],
[ 9318.34103227],
[16516.67703201],
[14119.76250514],
[ 9841.31930684],
[ 9620.03726747],
[ 7493.79508949],
[10460.9876164 ],
[11198.08573629],
[ 8945.94034292],
[ 9642.57659017],
[ 7617.42749859],
[10320.14379622],
[16410.03124219],
```

```
[10890.22504944],
[11419.63673035],
[ 6439.60557201],
[10601.88699126],
[11465.75330223],
[13286.14520496],
[ 9776.88299838],
[ 7891.85016949],
[14083.90428194],
[ 8012.85810836],
[11609.51947649],
[ 6492.84578953],
[ 6039.08354484],
[ 9265.15821039],
[19066.47710763],
[ 7638.22644904],
[15979.66019384],
[ 7609.30630763],
[ 9372.34774911],
[ 9603.2380719 ],
[13325.49466447],
[12380.21034439],
[11077.57863599],
[ 7980.52066834],
[11625.68104576],
[ 8605.71715271],
[10481.30459326],
[11255.65475033],
[13590.47668503],
[17047.6963529 ],
[ 9002.67111169],
[ 8579.42781511],
[10989.70185064],
[ 7829.76990683],
[ 8822.18892462],
[17135.06007659],
[ 9608.5015533 ],
[18031.28698027],
[10430.46594937],
[ 9209.11232236],
[11351.19576789],
[10697.34655383],
[10132.40994688],
[10835.71807973],
[ 5421.28040684],
[12700.17942413],
[10037.63712802],
```

```
[10492.13536908],
[12342.64284221],
[15284.73570158],
[10683.91119795],
[13011.52786032],
[14004.28589892],
[ 9913.30177503],
[12376.19904656],
[ 7004.10783448],
[10107.92806261],
[10066.60977184],
[ 9619.24139831],
[13453.36363758],
[ 8332.45541273],
[ 8135.86258629],
[11510.060646  ],
[ 7997.48972117],
[11374.51821347],
[13586.97022726],
[13082.44666478],
[11566.47078108],
[ 9379.03237017],
[14362.75317579],
[ 8756.95656963],
[ 9642.12966453],
[10874.63832087],
[14321.09192644],
[13062.86896265],
[12247.77298719],
[ 7375.58728891],
[ 7436.79353047],
[ 9602.13157595],
[12929.03343845],
[ 7376.0941048 ],
[11259.95505655],
[ 9366.55613946],
[15593.17603965],
[ 8450.6859248 ],
[10131.13671264],
[10787.91909476],
[ 8712.18631126],
[ 6028.71578124],
[10687.6038397 ],
[14904.41657345],
[11029.087365  ],
[ 8507.28334057],
[ 9643.10393777],
```

```
[ 6955.25802718],
[ 9234.46693873],
[ 8719.77859999],
[ 5865.32652709],
[ 6272.59193477],
[ 9306.87200011],
[ 6281.01114655],
[ 9557.60021504],
[11159.40267258],
[10892.18440401],
[ 9175.05325679],
[ 7215.43006764],
[ 8277.642084  ],
[10617.65363443],
[11782.19910638],
[ 6909.43916888],
[10122.44855285],
[ 8536.45217683],
[10054.77435021],
[11385.72139528],
[ 8952.94099177],
[12241.54911605],
[ 9852.78552787],
[ 9284.82609835],
[ 8862.29842659],
[16531.76772015],
[ 7677.38901307],
[11504.82999047],
[ 8066.87818892],
[ 6887.92089131],
[ 9229.81680635],
[11033.71777006],
[10914.752882  ],
[ 9275.02060369],
[ 7364.54633188],
[13069.82862508],
[ 9858.64052364],
[ 9675.69358444],
[ 7477.65985151],
[ 9186.38597869],
[17707.8446993 ],
[11934.42065824],
[ 6274.71847809],
[ 8033.36043773],
[26146.65180758],
[12692.62981921],
[16258.08482206],
```

```
        [11703.12457937],
        [11391.84744236]])
```

[54]:
```
r2_score(y_test,y_pred)
ridge.coef_
```

[54]:
```
array([[-1.24459301e+02, -1.76887835e-02,  3.75990013e+02,
         2.66707636e+01,  3.58197632e+02, -3.00276827e-02,
         1.24675635e+00,  0.00000000e+00,  6.44721526e+02,
         1.91042643e+01]])
```

[55]:
```
ridge.intercept_
```

[55]:
```
array([-8198.6517118])
```

[ ]:
```

```