sirilasso-ridgemodel

August 21, 2025

[1]:

```
import pandas as pd
import numpy as np

#Import graphical plotting libraries
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline

#Import Linear Regression Machine Learning Libraries
from sklearn import preprocessing
from sklearn.preprocessing import PolynomialFeatures
from sklearn.model_selection import train_test_split

from sklearn.linear_model import LinearRegression, Ridge, Lasso
from sklearn.metrics import r2_score
```

[2]:

```
data = pd.read_csv(r'C:\Users\ttwrd\Downloads\car-mpg.csv')
data.head()
```

[2]:

```
    mpg  cyl   disp   hp    wt   acc  yr  origin  car_type  \
0  18.0    8  307.0  130  3504  12.0  70       1         0
1  15.0    8  350.0  165  3693  11.5  70       1         0
2  18.0    8  318.0  150  3436  11.0  70       1         0
3  16.0    8  304.0  150  3433  12.0  70       1         0
4  17.0    8  302.0  140  3449  10.5  70       1         0

                    car_name
0  chevrolet chevelle malibu
1          buick skylark 320
2         plymouth satellite
3              amc rebel sst
4                ford torino
```

[3]:

```
data = data.drop(['car_name'], axis = 1)
```

[4]:

```
data
```

```
[4]:      mpg  cyl   disp   hp    wt   acc  yr  origin  car_type
     0    18.0    8  307.0  130  3504  12.0  70       1         0
     1    15.0    8  350.0  165  3693  11.5  70       1         0
     2    18.0    8  318.0  150  3436  11.0  70       1         0
     3    16.0    8  304.0  150  3433  12.0  70       1         0
     4    17.0    8  302.0  140  3449  10.5  70       1         0
     ..    …    …     …    …     …     …   ..       …         …
     393  27.0    4  140.0   86  2790  15.6  82       1         1
     394  44.0    4   97.0   52  2130  24.6  82       2         1
     395  32.0    4  135.0   84  2295  11.6  82       1         1
     396  28.0    4  120.0   79  2625  18.6  82       1         1
     397  31.0    4  119.0   82  2720  19.4  82       1         1

     [398 rows x 9 columns]
```

```
[5]: data['origin'] = data['origin'].replace({1: 'america', 2: 'europe', 3: 'asia'})
```

```
[6]: data
```

```
[6]:      mpg  cyl   disp   hp    wt   acc  yr   origin  car_type
     0    18.0    8  307.0  130  3504  12.0  70  america         0
     1    15.0    8  350.0  165  3693  11.5  70  america         0
     2    18.0    8  318.0  150  3436  11.0  70  america         0
     3    16.0    8  304.0  150  3433  12.0  70  america         0
     4    17.0    8  302.0  140  3449  10.5  70  america         0
     ..    …    …     …    …     …     …   ..       …          …
     393  27.0    4  140.0   86  2790  15.6  82  america         1
     394  44.0    4   97.0   52  2130  24.6  82   europe         1
     395  32.0    4  135.0   84  2295  11.6  82  america         1
     396  28.0    4  120.0   79  2625  18.6  82  america         1
     397  31.0    4  119.0   82  2720  19.4  82  america         1

     [398 rows x 9 columns]
```

```
[7]: data = pd.get_dummies(data,columns = ['origin'])
```

```
[8]: data.head()
```

```
[8]:    mpg  cyl   disp   hp    wt   acc  yr  car_type  origin_america  \
     0  18.0    8  307.0  130  3504  12.0  70         0            True
     1  15.0    8  350.0  165  3693  11.5  70         0            True
     2  18.0    8  318.0  150  3436  11.0  70         0            True
     3  16.0    8  304.0  150  3433  12.0  70         0            True
     4  17.0    8  302.0  140  3449  10.5  70         0            True

        origin_asia  origin_europe
     0        False          False
```

```
1        False          False
2        False          False
3        False          False
4        False          False
```

[9]:
```
data = data.replace('?', np.nan)
```

[10]:
```
data
```

[10]:
```
      mpg  cyl   disp   hp    wt   acc  yr car_type  origin_america  \
0    18.0    8  307.0  130  3504  12.0  70         0            True
1    15.0    8  350.0  165  3693  11.5  70         0            True
2    18.0    8  318.0  150  3436  11.0  70         0            True
3    16.0    8  304.0  150  3433  12.0  70         0            True
4    17.0    8  302.0  140  3449  10.5  70         0            True
..    …  …    …  …   …   …   ..      …             …
393  27.0    4  140.0   86  2790  15.6  82         1            True
394  44.0    4   97.0   52  2130  24.6  82         1           False
395  32.0    4  135.0   84  2295  11.6  82         1            True
396  28.0    4  120.0   79  2625  18.6  82         1            True
397  31.0    4  119.0   82  2720  19.4  82         1            True

     origin_asia  origin_europe
0          False          False
1          False          False
2          False          False
3          False          False
4          False          False
..           …            …
393        False          False
394        False           True
395        False          False
396        False          False
397        False          False

[398 rows x 11 columns]
```

[26]:
```
data=data.apply(pd.to_numeric,errors='ignore')
```

```
C:\Users\ttwrd\AppData\Local\Temp\ipykernel_67368\3768586041.py:1:
FutureWarning: errors='ignore' is deprecated and will raise in a future version.
Use to_numeric without passing `errors` and catch exceptions explicitly instead
  data=data.apply(pd.to_numeric,errors='ignore')
```

[13]:
```
data
```

[13]:
```
      mpg  cyl   disp     hp    wt   acc  yr  car_type  origin_america  \
0    18.0    8  307.0  130.0  3504  12.0  70         0            True   
1    15.0    8  350.0  165.0  3693  11.5  70         0            True   
2    18.0    8  318.0  150.0  3436  11.0  70         0            True   
3    16.0    8  304.0  150.0  3433  12.0  70         0            True   
4    17.0    8  302.0  140.0  3449  10.5  70         0            True   
..     …    …      …      …     …     …  ..        …              …
393  27.0    4  140.0   86.0  2790  15.6  82         1            True   
394  44.0    4   97.0   52.0  2130  24.6  82         1           False   
395  32.0    4  135.0   84.0  2295  11.6  82         1            True   
396  28.0    4  120.0   79.0  2625  18.6  82         1            True   
397  31.0    4  119.0   82.0  2720  19.4  82         1            True   

     origin_asia  origin_europe  
0          False          False  
1          False          False  
2          False          False  
3          False          False  
4          False          False  
..           …              …
393        False          False  
394        False           True  
395        False          False  
396        False          False  
397        False          False  

[398 rows x 11 columns]
```

[27]:
```
numeric_cols=data.select_dtypes(include=[np.number]).columns
data[numeric_cols] = data[numeric_cols].apply(lambda x: x.fillna(x.median()))
```

[28]:
```
data.head()
```

[28]:
```
    mpg  cyl   disp     hp    wt   acc  yr  car_type  origin_america  \
0  18.0    8  307.0  130.0  3504  12.0  70         0            True   
1  15.0    8  350.0  165.0  3693  11.5  70         0            True   
2  18.0    8  318.0  150.0  3436  11.0  70         0            True   
3  16.0    8  304.0  150.0  3433  12.0  70         0            True   
4  17.0    8  302.0  140.0  3449  10.5  70         0            True   

   origin_asia  origin_europe  
0        False          False  
1        False          False  
2        False          False  
3        False          False  
4        False          False  
```

[29]:
```
X = data.drop(['mpg'], axis = 1)
y = data[['mpg']]
```

[30]:
```
X_s = preprocessing.scale(X)
X_s = pd.DataFrame(X_s, columns = X.columns)

y_s = preprocessing.scale(y)
y_s = pd.DataFrame(y_s, columns = y.columns) #
```

[31]:
```
X_s
```

[31]:
```
          cyl      disp        hp        wt       acc        yr  car_type  \
0    1.498191  1.090604  0.673118  0.630870 -1.295498 -1.627426 -1.062235
1    1.498191  1.503514  1.589958  0.854333 -1.477038 -1.627426 -1.062235
2    1.498191  1.196232  1.197027  0.550470 -1.658577 -1.627426 -1.062235
3    1.498191  1.061796  1.197027  0.546923 -1.295498 -1.627426 -1.062235
4    1.498191  1.042591  0.935072  0.565841 -1.840117 -1.627426 -1.062235
..        …         …         …         …         …         …         …
393 -0.856321 -0.513026 -0.479482 -0.213324  0.011586  1.621983  0.941412
394 -0.856321 -0.925936 -1.370127 -0.993671  3.279296  1.621983  0.941412
395 -0.856321 -0.561039 -0.531873 -0.798585 -1.440730  1.621983  0.941412
396 -0.856321 -0.705077 -0.662850 -0.408411  1.100822  1.621983  0.941412
397 -0.856321 -0.714680 -0.584264 -0.296088  1.391285  1.621983  0.941412

     origin_america  origin_asia  origin_europe
0          0.773559    -0.497643      -0.461968
1          0.773559    -0.497643      -0.461968
2          0.773559    -0.497643      -0.461968
3          0.773559    -0.497643      -0.461968
4          0.773559    -0.497643      -0.461968
..              …            …              …
393        0.773559    -0.497643      -0.461968
394       -1.292726    -0.497643       2.164651
395        0.773559    -0.497643      -0.461968
396        0.773559    -0.497643      -0.461968
397        0.773559    -0.497643      -0.461968

[398 rows x 10 columns]
```

[32]:
```
y_s
```

[32]:
```
          mpg
0   -0.706439
1   -1.090751
2   -0.706439
3   -0.962647
4   -0.834543
```

```
..         …
393  0.446497
394  2.624265
395  1.087017
396  0.574601
397  0.958913

[398 rows x 1 columns]
```

[34]:

```
X_train, X_test, y_train,y_test = train_test_split(X_s, y_s, test_size = 0.30,␣
 ↪random_state = 1)
X_train.shape
```

[34]:

```
(278, 10)
```

[35]:

```
regression_model = LinearRegression()
regression_model.fit(X_train, y_train)

for idx, col_name in enumerate(X_train.columns):
    print('The coefficient for {} is {}'.format(col_name, regression_model.
 ↪coef_[0][idx]))

intercept = regression_model.intercept_[0]
print('The intercept is {}'.format(intercept))
```

```
The coefficient for cyl is 0.3210223856916108
The coefficient for disp is 0.3248343091848394
The coefficient for hp is -0.2291695005943759
The coefficient for wt is -0.7112101905072299
The coefficient for acc is 0.014713682764191435
The coefficient for yr is 0.3755811949510741
The coefficient for car_type is 0.38147694842331
The coefficient for origin_america is -0.0747224754758417
The coefficient for origin_asia is 0.04451525203567813
The coefficient for origin_europe is 0.04834854953945371
The intercept is 0.019284116103639715
```

[37]:

```
ridge_model = Ridge(alpha = 0.4)
ridge_model.fit(X_train, y_train)

print('Ridge model coef: {}'.format(ridge_model.coef_))
```

```
Ridge model coef: [ 0.31495967  0.30948411 -0.22861679 -0.69782283  0.01239531
0.37411266
  0.37586629 -0.07408168  0.04437854  0.0476772 ]
```

[40]:

```
lasso_model = Lasso(alpha = 0.1)
lasso_model.fit(X_train, y_train)

print('Lasso model coef: {}'.format(lasso_model.coef_))
```

```
Lasso model coef: [-0.         -0.         -0.01690287 -0.51890013  0.
0.28138241
  0.1278489  -0.01642647  0.          0.        ]
```

[39]:

```
print(regression_model.score(X_train, y_train))
print(regression_model.score(X_test, y_test))

print('*************************')
#Ridge
print(ridge_model.score(X_train, y_train))
print(ridge_model.score(X_test, y_test))

print('*************************')
#Lasso
print(lasso_model.score(X_train, y_train))
print(lasso_model.score(X_test, y_test))
```

```
0.8343770256960538
0.8513421387780067
*************************
0.8343502868181134
0.8520594956782537
*************************
0.7938010766228453
0.8375229615977084
```

[ ]: