Q03

July 1, 2020

[1]:
```
import pandas as pd
import numpy as np
from sklearn.preprocessing import StandardScaler
from sklearn.linear_model import RidgeCV
from sklearn.linear_model import LassoCV
from sklearn.metrics import mean_squared_error
from sklearn.linear_model import ElasticNetCV
import warnings
warnings.filterwarnings('ignore')
```

[2]:
```
train = pd.read_csv("Shiptrain-1.csv", header=None)
test = pd.read_csv("Shiptest-1.csv", header=None)

x_train = train.values[:,:-1]
y_train = train.values[:,-1]

x_test = test.values[:,:-1]
y_test = test.values[:,-1]

scaler = StandardScaler()
scaler.fit(x_train)
x_train_scaled = scaler.transform(x_train)
x_test_scaled = scaler.transform(x_test)
```

0.0.1 Ridge Regression

[3]:
```
rdg = RidgeCV(cv=5).fit(x_train_scaled, y_train)
rdg.alpha_
```

[3]:
```
0.1
```

Ridge Coefficients

[4]:
```
best_rdg_coef = rdg.coef_
best_rdg_coef
```

[4]:
```
array([ 0.17428573, -0.11884258, -0.03370589, -0.00967058,  0.04598974,
       -0.12243568,  0.03190865, -0.27621958,  0.19242037,  0.0951851 ,
       -0.02704903, -0.01458064,  0.06480845])
```

[5]:
```
y_hat = rdg.predict(x_test_scaled)
mean_squared_error(y_test, y_hat)
```

[5]:
```
3.0306735052865293e-05
```

0.0.2 Lasso Regression

[6]:
```
lasso = LassoCV(fit_intercept=False, cv=5, random_state=0, max_iter=70000).
↪fit(x_train_scaled, y_train)
lasso.alpha_
```

[6]:
```
0.0006622625066455708
```

Lasso Coefficients

[13]:
```
lasso.coef_
```

[13]:
```
array([ 0.00000000e+00,  0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
       -0.00000000e+00,  0.00000000e+00, -0.00000000e+00, -8.46681712e-18,
        0.00000000e+00,  0.00000000e+00,  0.00000000e+00, -0.00000000e+00,
       -0.00000000e+00])
```

[7]:
```
y_hat = lasso.predict(x_test_scaled)
mean_squared_error(y_test, y_hat)
```

[7]:
```
0.9503206870548805
```

0.0.3 Elastic Net

[8]:
```
en = ElasticNetCV(cv=5, random_state=0, max_iter=55000).fit(x_train_scaled,␣
↪y_train)
en.alpha_
```

[8]:
```
1.3245250132910862e-06
```

EN Coefficients

[14]:
```
en.coef_
```

[14]:
```
array([ 0.18664293, -0.13026094, -0.05028474, -0.01226128,  0.0458922 ,
       -0.11768202,  0.03617479, -0.27911135,  0.21607935,  0.08647057,
       -0.02706473, -0.01467713,  0.06215156])
```

[9]:
```
en.l1_ratio_
```

[9]:
```
0.5
```

[10]:
```
y_hat = en.predict(x_test_scaled)
mean_squared_error(y_test, y_hat)
```

[10]:
```
3.0259507402732806e-05
```

[11]:
```
# To use in R notebook
pd.DataFrame(x_train_scaled).to_csv("x_train_scaled.csv")
pd.DataFrame(x_test_scaled).to_csv("x_test_scaled.csv")
```

0.0.4 Discussion

The best alpha for ridge, lasso, elastic net and adaptive lasso (seen in R notebook) was: .1, .0007, 1.3e-06, and .012 respectively. The elastic net l1/l2 ratio was .5. THe most interesting part to me, however, are the coefficients. The lasso model was able to eliminate all features, but one and still have a 0.95 mse. The other models performed better using mse as a metric, 3.03e-5, 3.02e-5, and 6.6e-5 for ridge, elastic net and adaptive lasso respectively.

As to which model I would use I think it depends. If I am using the model to explain how a system works, then Lasso, because it offers the simplest model. If I want to use the model in production and have the greatest predictive power, then adaptive lasso.