ML6

May 23, 2022

[2]:
```
# Apply Linear, Ridge, Lasso Regression technique of machine learning
# to analyze and build the model of the Diabetesdataset.
# Display and compare the accuracy(Cross-Validation, R2 Score)
# of all the models
```

[2]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.linear_model import LinearRegression, Ridge, RidgeCV, Lasso,␣
 ↪LassoCV
from sklearn.datasets import load_diabetes
from sklearn.model_selection import cross_val_score
from sklearn.metrics import r2_score
```

[3]:
```
diabetes = load_diabetes()
X = diabetes.data
y = diabetes.target
```

[9]:
```
X.shape
```

[9]:
```
(442, 10)
```

[4]:
```
print(X.shape)
print(y.shape)
```

```
(442, 10)
(442,)
```

[5]:
```
lr = LinearRegression()
lr.fit(X,y)
y_predict = lr.predict(X)

print(lr.score(X,y))
lr_scores = cross_val_score(lr,X,y,cv=5)
print(lr_scores.mean())
print(lr.coef_)
print(lr.intercept_)
print(r2_score(y,y_predict))
```

```
0.5177494254132934
0.4823181221114939
[ -10.01219782 -239.81908937  519.83978679  324.39042769 -792.18416163
  476.74583782  101.04457032  177.06417623  751.27932109   67.62538639]
152.1334841628965
0.5177494254132934
```

[30]:

```python
# Lasso
lr = Lasso(alpha=0.05)
lr.fit(X,y)
y_predict = lr.predict(X)

print(lr.score(X,y))
lr_scores = cross_val_score(lr,X,y,cv=5)
print(lr_scores.mean())
# print(lr.coef_)
# print(lr.intercept_)
```

```
0.5131486010621044
0.48204217044282477
[  -0.         -194.04749923  521.81917521  295.22384739  -99.44970769
   -0.         -222.72076612    0.          512.04940782   52.92594445]
152.13348416289645
```

[31]:

```python
# LassoCV
lr = LassoCV(alphas=[0.01,0.02,0.04,0.05,0.1,0.2,0.5])
lr.fit(X,y)
y_predict = lr.predict(X)

print(lr.alpha_)
print(lr.score(X,y))
lr_scores = cross_val_score(lr,X,y,cv=5)
print(lr_scores.mean())
# print(lr.coef_)
# print(lr.intercept_)
```

```
0.04
0.5138646937185254
0.47995535949978924
[  -0.         -202.27296915  522.96840122  299.88446204 -118.13019877
   -0.         -212.94729492   16.5039356   517.11486339   56.3117281 ]
152.13348416289645
```

[32]:

```python
# Ridge
lr = Ridge(alpha=0.05)
lr.fit(X,y)
y_predict = lr.predict(X)
```

```
print(lr.score(X,y))
lr_scores = cross_val_score(lr,X,y,cv=5)
print(lr_scores.mean())
# print(lr.coef_)
# print(lr.intercept_)
```

```
0.5144431310030684
0.48081634735342826
[  -2.69045402 -221.28733251  507.59872561  311.48379569 -137.96353066
  -36.98110839 -174.62635298  113.63280903  483.13024134   79.20916664]
152.13348416289642
```

[33]:

```
# LassoCV
lr = RidgeCV(alphas=[0.01,0.02,0.04,0.05,0.1,0.2,0.5])
lr.fit(X,y)
y_predict = lr.predict(X)

print(lr.alpha_)
print(lr.score(X,y))
lr_scores = cross_val_score(lr,X,y,cv=5)
print(lr_scores.mean())
# print(lr.coef_)
# print(lr.intercept_)
```

```
0.01
0.5166287840315842
0.4802027531860105
[  -7.19945679 -234.55293001  520.58313622  320.52335582 -380.60706569
  150.48375154  -78.59123221  130.31305868  592.34958662   71.1337681 ]
152.13348416289645
```

[ ]:

```

```