ML-Assignment-1 - Navya Renduchintala - 2306AML112

1. Implement Linear Regression, Ridge Regression and Lasso regression on teams dataset
2. Use cross validation score and RMSE, R2 score.
3. compare the results of various regression techniques
4. Finally write your analysis

Linear Regression

In [2]:
```
import numpy as np
import pandas as pd
```

In [3]:
```
data = pd.read_csv('teams.csv')
```

In [4]:
```
X = data.iloc[:,1:8]
Y = data['medals']
```

In [5]:
```
data.isnull().sum()
```

Out[5]:
```
team           0
year           0
athletes       0
events         0
age            0
height         0
weight         0
prev_medals    0
medals         0
dtype: int64
```

In [6]:
```
X
```

Out[6]:

	year	athletes	events	age	height	weight	prev_medals
0	1964	8	8	22.0	161.0	64.2	0.0
1	1968	5	5	23.2	170.2	70.0	0.0
2	1972	8	8	29.0	168.3	63.8	0.0
3	1980	11	11	23.6	168.4	63.2	0.0
4	2004	5	5	18.6	170.8	64.8	0.0
...	...	...	...	...	...	...	...
2009	2000	26	19	25.0	179.0	71.1	0.0
2010	2004	14	11	25.1	177.8	70.5	0.0
2011	2008	16	15	26.1	171.9	63.7	3.0
2012	2012	9	8	27.3	174.4	65.2	4.0
2013	2016	31	13	27.5	167.8	62.2	0.0

2014 rows × 7 columns

In [7]:
```
Y
```

Out[7]:
```
0       0
1       0
2       0
3       0
4       0
       ..
2009    0
2010    3
2011    4
2012    0
2013    0
Name: medals, Length: 2014, dtype: int64
```

In [8]:
```
import pandas as pd
one_hot_encoded = pd.get_dummies(data, columns = ['team'])
```

In [9]:
```
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)

print(X_train.shape)
print(X_test.shape)
print(Y_train.shape)
print(Y_test.shape)
```

```
(1409, 7)
(605, 7)
(1409,)
(605,)
```

In [10]:
```
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error

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

Out[10]:
```
▾ LinearRegression
LinearRegression()
```

In [11]:
```
from sklearn.metrics import r2_score
```

In [12]:
```
y_test_predict = lin_model.predict(X_test)
```

In [13]:
```
rmse = (np.sqrt(mean_squared_error(Y_test, y_test_predict)))
r2 = r2_score(Y_test, y_test_predict)

print('RMSE is {}'.format(rmse))
print('R2 score is {}'.format(r2))
```

```
RMSE is 10.270951521656123
R2 score is 0.8543923284524518
```

In [14]:
```
from sklearn.linear_model import Lasso
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import cross_val_score
from sklearn.model_selection import RepeatedKFold
from sklearn.linear_model import Ridge
import pandas as pd
import numpy as np

data = pd.read_csv('teams.csv')
X = data.iloc[:,1:8]
Y = data['medals']

lr_model = LinearRegression()

cv = RepeatedKFold(n_splits=10, n_repeats=3, random_state=1)

scores = cross_val_score(lr_model, X, Y, scoring='neg_mean_absolute_error', cv=cv, n_jobs=-1)

scores = np.absolute(scores)
print('Mean MAE: %.3f (%.3f)' % (np.mean(scores), np.std(scores)))
```

```
Mean MAE: 4.730 (0.597)
```

Lasso Regression

In [15]:
```
import pandas as pd
import numpy as np
import warnings
warnings.filterwarnings('ignore')

from sklearn.model_selection import cross_val_score
from sklearn.model_selection import RepeatedKFold
from sklearn.linear_model import Lasso

data = pd.read_csv('teams.csv')
X = data.iloc[:,1:8]
Y = data['medals']
```

In [16]:
```
lasso_model = Lasso(alpha=1.0)

cv = RepeatedKFold(n_splits=10, n_repeats=3, random_state=1)

scores = cross_val_score(lasso_model, X, Y, scoring='neg_mean_absolute_error', cv=cv, n_jobs=-1)

scores = np.absolute(scores)
print('Mean MAE: %.3f (%.3f)' % (np.mean(scores), np.std(scores)))
```

```
Mean MAE: 4.657 (0.604)
```

Ridge Regression

In [17]:
```
from sklearn.model_selection import cross_val_score
from sklearn.model_selection import RepeatedKFold
from sklearn.linear_model import Ridge
import pandas as pd
import numpy as np
```

```
data = pd.read_csv('teams.csv')
X = data.iloc[:,1:8]
Y = data['medals']

ridge_model = Ridge(alpha=1.0)

cv = RepeatedKFold(n_splits=10, n_repeats=3, random_state=1)

scores = cross_val_score(ridge_model, X, Y, scoring='neg_mean_absolute_error', cv=cv, n_jobs=-1)

scores = np.absolute(scores)
print('Mean MAE: %.3f (%.3f)' % (np.mean(scores), np.std(scores)))
```

```
Mean MAE: 4.730 (0.597)
```

In [18]:

```
from sklearn.model_selection import KFold
from sklearn.model_selection import cross_val_score
from sklearn.linear_model import LinearRegression
from sklearn.linear_model import Lasso
from sklearn.linear_model import Ridge
from sklearn.metrics import mean_squared_error
from sklearn.model_selection import train_test_split

data = pd.read_csv('teams.csv')
X = data.iloc[:,1:8]
Y = data['medals']

X_train, X_test, Y_train, Y_test =  train_test_split(X, Y, test_size = 0.3)

folds   = 10
metric  = "neg_mean_squared_error"


models = {}
models["Linear"]        = LinearRegression()
models["Lasso"]         = Lasso()
models['Ridge']         = Ridge()

model_results = []
model_names   = []
for model_name in models:
        model   = models[model_name]
        k_fold  = KFold(n_splits=folds)
        results = cross_val_score(model, X_train, Y_train, cv=k_fold, scoring=metric)

        model_results.append(results)
        model_names.append(model_name)
        print("{}: {}, {}".format(model_name, round(results.mean(), 2), round(results.std(), 2)))
```

```
Linear: -145.04, 76.22
Lasso: -144.98, 76.71
Ridge: -145.04, 76.22
```

Analysis:

Linear and Ridge regression are best suited for teams dataset based on mean values.