In [58]:
```
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[:,2:4]
Y = data['medals']
X
```

Out[58]:

	athletes	events
0	8	8
1	5	5
2	8	8
3	11	11
4	5	5
...	...	...
2009	26	19
2010	14	11
2011	16	15
2012	9	8
2013	31	13

2014 rows × 2 columns

In [59]:
```
Y
```

Out[59]:
```
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 [60]:
```
from sklearn.linear_model import LinearRegression
```

In [61]:
```
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: 7.007 (0.799)
```

In [63]:
```
lr_model = LinearRegression()

lr_model.fit(X, Y)

row = [8,8]

yhat = lr_model.predict([row])

print('Predicted: %.3f' % yhat)
```

```
Predicted: -1.648
```

In [66]:
```
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: 7.000 (0.802)
```

In [67]:
```
lasso_model1 = Lasso(alpha=0.01)

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

scores = cross_val_score(lasso_model1, 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: 7.007 (0.799)
```

In [68]:
```
lasso_model1 = Lasso(alpha=0.01)
lasso_model.fit(X, Y)

row = [8,8]

yhat = lasso_model.predict([row])

print('Predicted: %.3f' % yhat)
```

```
Predicted: -1.681
```

In [44]:
```
from sklearn.linear_model import Ridge
```

In [69]:
```
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: 7.007 (0.799)
```

In [70]:
```
ridge_model = Ridge(alpha=0.5)

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: 7.007 (0.799)
```

In [71]:
```
ridge_model = Ridge(alpha=1.0)

ridge_model.fit(X, Y)

row = [8,8]

yhat = ridge_model.predict([row])

print('Predicted: %.3f' % yhat)
```

```
Predicted: -1.648
```

In [72]:
```
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


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: -293.97, 168.12
Lasso: -294.01, 169.13
Ridge: -293.97, 168.12
```