Introduction to Cross Validation

In this lecture series we will do a much deeper dive into various methods of cross-validation. As well as a discussion on the general philosphy behind cross validation. A nice official documentation guide can be found here: https://scikit-learn.org/stable/modules/cross_validation.html

Imports

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

Data Example

```
df= pd.read_csv("D:\\Study\\Programming\\python\Python course from udemy\\Udemy - 2022 Python for Machine Learning & Data Science Masterclass\\01 - Introduction to Course\\1UNZIP-FOR-NOTEBOOKS-FINAL\\08-Linear-Regression-Models\\Advertising.csv")
df.head()
```

```
      TV  radio  newspaper  sales
0  230.1   37.8       69.2   22.1
1   44.5   39.3       45.1   10.4
2   17.2   45.9       69.3    9.3
3  151.5   41.3       58.5   18.5
4  180.8   10.8       58.4   12.9
```

Train | Test Split Procedure

1. Clean and adjust data as necessary for X and y
2. Split Data in Train/Test for both X and y
3. Fit/Train Scaler on Training X Data
4. Scale X Test Data
5. Create Model
6. Fit/Train Model on X Train Data
7. Evaluate Model on X Test Data (by creating predictions and comparing to Y_test)
8. Adjust Parameters as Necessary and repeat steps 5 and 6

CREATE X and y

```
X=df.drop('sales',axis=1)

Y=df['sales']
```

TRAIN TEST SPLIT

```
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,rando
m_state=101)
```

SCALE DATA

```
from sklearn.preprocessing import StandardScaler

scaler = StandardScaler()

scaler.fit(X_train)

StandardScaler()

scaler_x_train =  scaler.transform(X_train)
scaler_x_test = scaler.transform(X_test)
```

Create Model

```
from sklearn.linear_model import Ridge

# Poor Alpha Choice on purpose!
model = Ridge(alpha=100)

model.fit(scaler_x_train,Y_train)

Ridge(alpha=100)

r1predict = model.predict(scaler_x_test)
```

Evaluation

```
from sklearn.metrics import mean_absolute_error,mean_squared_error

MAE = mean_absolute_error(Y_test,r1predict)
MAE

2.1631741364394363

RMSE = np.sqrt(mean_squared_error(Y_test,r1predict))
RMSE

2.709571144855608

mean_squared_error(Y_test,r1predict)

7.341775789034129
```

Adjust Parameters and Re-evaluate

```
model2= Ridge(alpha=1)

model2.fit(scaler_x_train,Y_train)

Ridge(alpha=1)
```

```
r2predict = model2.predict(scaler_x_test)
```

Another Evaluation

```
MAE2 = mean_absolute_error(Y_test,r2predict)
MAE2
```

```
1.2168768443580582
```

```
MSE = mean_squared_error(Y_test,r2predict)
MSE
```

```
2.3190215794287514
```

```
RMSE = np.sqrt(MSE)
RMSE
```

```
1.5228334050147283
```

Much better! We could repeat this until satisfied with performance metrics. (We previously showed RidgeCV can do this for us, but the purpose of this lecture is to generalize the CV process for any model).

Train | Validation | Test Split Procedure

This is often also called a "hold-out" set, since you should not adjust parameters based on the final test set, but instead use it *only* for reporting final expected performance.

1. Clean and adjust data as necessary for X and y
2. Split Data in Train/Validation/Test for both X and y
3. Fit/Train Scaler on Training X Data
4. Scale X Eval Data
5. Create Model
6. Fit/Train Model on X Train Data
7. Evaluate Model on X Evaluation Data (by creating predictions and comparing to Y_eval)
8. Adjust Parameters as Necessary and repeat steps 5 and 6
9. Get final metrics on Test set (not allowed to go back and adjust after this!)

```
#######################################################################
#### SPLIT TWICE! Here we create TRAIN | VALIDATION | TEST  #########
#####################################################################

# 70% of data is training data, set aside other 30%
X_train,X_other,Y_train,Y_other=train_test_split(X,Y,test_size=0.3,ran
dom_state=101)

# Remaining 30% is split into evaluation and test sets
```

```python
# Each is 15% of the original data size
X_eval,X_test,Y_eval,Y_test =
train_test_split(X_other,Y_other,test_size=0.50,random_state=101)
```

```python
# Scale Data
scaler= StandardScaler()
scaler.fit(X_train)
scaler_x_train = scaler.transform(X_train)
scaler_x_eval = scaler.transform(X_eval)
scaler_x_test = scaler.transform(X_test)
```

Create Model

```python
# Poor Alpha Choice on purpose!
model = Ridge(alpha=100)

model.fit(scaler_x_train,Y_train)

Ridge(alpha=100)
```

Evaluation

```python
y_eval_pred = model.predict(scaler_x_eval)

MAE = mean_absolute_error(Y_eval,y_eval_pred)
MSE = mean_squared_error(Y_eval,y_eval_pred)
RMSE = np.sqrt(mean_squared_error(Y_eval,y_eval_pred))
MAE, MSE,RMSE

(2.1754243744399884, 7.320101458823872, 2.7055686017589484)
```

Adjust Parameters and Re-evaluate

```python
model = Ridge(alpha=1)

model.fit(scaler_x_train,Y_train)

Ridge(alpha=1)

y_eval2_pred = model.predict(scaler_x_eval)

MAE = mean_absolute_error(Y_eval,y_eval2_pred)
MSE = mean_squared_error(Y_eval,y_eval2_pred)
RMSE = np.sqrt(mean_squared_error(Y_eval,y_eval2_pred))
MAE, MSE,RMSE

(1.195143424023704, 2.3837830750569866, 1.5439504768796786)
```

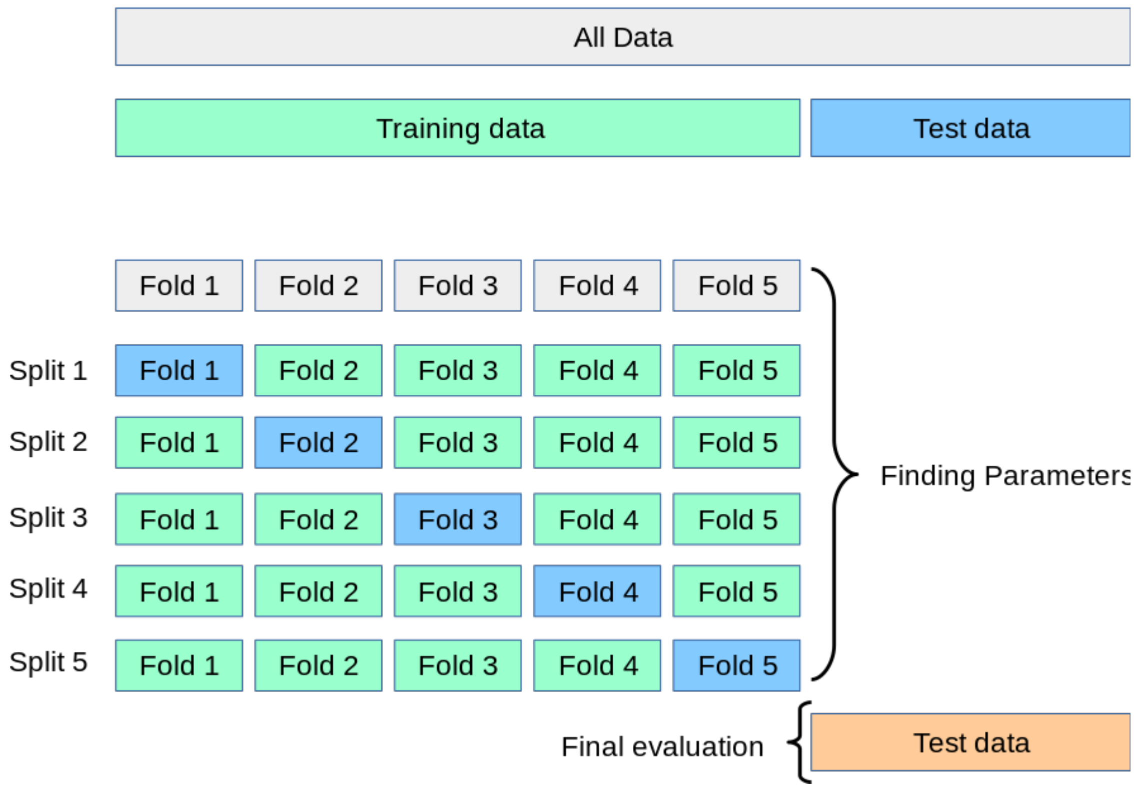
Final Evaluation (Can no longer edit parameters after this!)

```python
y_final_test_pred = model.predict(scaler_x_test)

MAE = mean_absolute_error(Y_test,y_final_test_pred)
MSE = mean_squared_error(Y_test,y_final_test_pred)
RMSE = np.sqrt(mean_squared_error(Y_test,y_final_test_pred))
MAE, MSE,RMSE
```

```
(1.2386102646924124, 2.254260083800517, 1.5014193564093001)
```

Cross Validation with cross_val_score




```
## CREATE X and y
X = df.drop('sales',axis=1)
Y = df['sales']

# TRAIN TEST SPLIT
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, random_state=101)

# SCALE DATA
from sklearn.preprocessing import StandardScaler
scaler = StandardScaler()
scaler.fit(X_train)
scaler_x_train = scaler.transform(X_train)
scaler_x_test = scaler.transform(X_test)

model = Ridge(alpha=100)

from sklearn.model_selection import cross_val_score
```

```python
# SCORING OPTIONS:
# https://scikit-learn.org/stable/modules/model_evaluation.html
scores = 
cross_val_score(model,scaler_x_train,Y_train,scoring='neg_mean_squared
_error',cv=5)
```

```python
scores
```

```
array([ -9.32552967,  -4.9449624 , -11.39665242,  -7.0242106 ,
        -8.38562723])
```

```python
# Average of the MSE scores (we set back to positive)
abs(scores.mean())
```

```
8.215396464543607
```

Adjust model based on metric

```python
model = Ridge(alpha=1)
```

```python
# SCORING OPTIONS:
# https://scikit-learn.org/stable/modules/model_evaluation.html
scores = cross_val_score(model,scaler_x_train,Y_train,
                         scoring='neg_mean_squared_error',cv=5)
```

```python
# Average of the MSE scores (we set back to positive)
abs(scores.mean())
```

```
3.344839296530695
```

Final Evaluation (Can no longer edit parameters after this!)

```python
# Need to fit the model first!
model.fit(scaler_x_train,Y_train)
```

```
Ridge(alpha=1)
```

```python
y_final_test_pred = model.predict(scaler_x_test)
```

```python
mean_squared_error(Y_test,y_final_test_pred)
```

```
2.3190215794287514
```

Cross Validation with cross_validate

The cross_validate function differs from cross_val_score in two ways:

It allows specifying multiple metrics for evaluation.

It returns a dict containing fit-times, score-times (and optionally training scores as well as fitted estimators) in addition to the test score.

For single metric evaluation, where the scoring parameter is a string, callable or None, the keys will be:

```
- ['test_score', 'fit_time', 'score_time']
```

And for multiple metric evaluation, the return value is a dict with the following keys:

```
['test_<scorer1_name>', 'test_<scorer2_name>', 'test_<scorer...>', 'fit_time', 'score_time']
```

return_train_score is set to False by default to save computation time. To evaluate the scores on the training set as well you need to be set to True.

```
## CREATE X and y
X = df.drop('sales',axis=1)
Y = df['sales']

# TRAIN TEST SPLIT
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, random_state=101)

# SCALE DATA
from sklearn.preprocessing import StandardScaler
scaler = StandardScaler()
scaler.fit(X_train)
scaler_x_train = scaler.transform(X_train)
scaler_x_test = scaler.transform(X_test)

model = Ridge(alpha=100)

from sklearn.model_selection import cross_validate

# SCORING OPTIONS:
# https://scikit-learn.org/stable/modules/model_evaluation.html
scores = cross_validate(model,scaler_x_train,Y_train,

scoring=['neg_mean_absolute_error','neg_mean_squared_error','max_error'],cv=5)

scores

{'fit_time': array([0.00199461, 0.00199389, 0.00099802, 0.00099492, 0.00199723]),
 'score_time': array([0.00199461, 0.0009985 , 0.00199485, 0.0019989 , 0.00199604]),
 'test_neg_mean_absolute_error': array([-2.31243044, -1.74653361, -2.56211701, -2.01873159, -2.27951906]),
 'test_neg_mean_squared_error': array([ -9.32552967,  -4.9449624 , -11.39665242,  -7.0242106 ,
         -8.38562723]),
 'test_max_error': array([ -6.44988486,  -5.58926073, -10.33914027,  -
```

```
6.61950405,
         -7.75578515])}
```

```
pd.DataFrame(scores)
```

```
   fit_time  score_time  test_neg_mean_absolute_error  \
0  0.001995    0.001995                     -2.312430
1  0.001994    0.000998                     -1.746534
2  0.000998    0.001995                     -2.562117
3  0.000995    0.001999                     -2.018732
4  0.001997    0.001996                     -2.279519

   test_neg_mean_squared_error  test_max_error
0                    -9.325530       -6.449885
1                    -4.944962       -5.589261
2                   -11.396652      -10.339140
3                    -7.024211       -6.619504
4                    -8.385627       -7.755785
```

```
pd.DataFrame(scores).mean()
```

```
fit_time                        0.001596
score_time                      0.001797
test_neg_mean_absolute_error   -2.183866
test_neg_mean_squared_error    -8.215396
test_max_error                 -7.350715
dtype: float64
```

Adjust model based on metrics

```
model = Ridge(alpha=1)
```

```
# SCORING OPTIONS:
# https://scikit-learn.org/stable/modules/model_evaluation.html
scores = cross_validate(model,scaler_x_train,Y_train,

scoring=['neg_mean_absolute_error','neg_mean_squared_error','max_error
'],cv=5)
```

```
pd.DataFrame(scores).mean()
```

```
fit_time                        0.010571
score_time                      0.002195
test_neg_mean_absolute_error   -1.319685
test_neg_mean_squared_error    -3.344839
test_max_error                 -5.161145
dtype: float64
```

Final Evaluation (Can no longer edit parameters after this!)

```
# Need to fit the model first!
model.fit(scaler_x_train,Y_train)
```

```
Ridge(alpha=1)
```

```
y_final_test_pred = model.predict(scaler_x_test)
```

```
mean_squared_error(Y_test,y_final_test_pred)
```

```
2.3190215794287514
```

Grid Search

We can search through a variety of combinations of hyperparameters with a grid search. While many linear models are quite simple and even come with their own specialized versions that do a search for you, this method of a grid search will can be applied to *any* model from sklearn, and we will need to use it later on for more complex models, such as Support Vector Machines.

Formatting Data

```
## CREATE X and y
X = df.drop('sales',axis=1)
Y = df['sales']

# TRAIN TEST SPLIT
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, random_state=101)

# SCALE DATA
from sklearn.preprocessing import StandardScaler
scaler = StandardScaler()
scaler.fit(X_train)
scaler_x_train = scaler.transform(X_train)
scaler_x_test = scaler.transform(X_test)
```

Model

```
from sklearn.linear_model import ElasticNet
```

```
#help(ElasticNet)
```

```
base_elastic_model = ElasticNet()
```

Grid Search

A search consists of:

- an estimator (regressor or classifier such as sklearn.svm.SVC());
- a parameter space;
- a method for searching or sampling candidates;

- a cross-validation scheme
- a score function.

```
param_grid = {'alpha':[0.1,1,5,10,50,100],
              'l1_ratio':[.1, .5, .7, .9, .95, .99, 1]}

from sklearn.model_selection import GridSearchCV

# verbose number a personal preference
grid_model = GridSearchCV(estimator=base_elastic_model,
                          param_grid=param_grid,
                          scoring='neg_mean_squared_error',
                          cv=5,
                          verbose=2)

grid_model.fit(scaler_x_train,Y_train)
```

```
Fitting 5 folds for each of 42 candidates, totalling 210 fits
[CV] END ............................alpha=0.1, l1_ratio=0.1; total
time=   0.0s
[CV] END ............................alpha=0.1, l1_ratio=0.1; total
time=   0.0s
[CV] END ............................alpha=0.1, l1_ratio=0.1; total
time=   0.0s
[CV] END ............................alpha=0.1, l1_ratio=0.1; total
time=   0.0s
[CV] END ............................alpha=0.1, l1_ratio=0.1; total
time=   0.0s
[CV] END ............................alpha=0.1, l1_ratio=0.5; total
time=   0.0s
[CV] END ............................alpha=0.1, l1_ratio=0.5; total
time=   0.0s
[CV] END ............................alpha=0.1, l1_ratio=0.5; total
time=   0.0s
[CV] END ............................alpha=0.1, l1_ratio=0.5; total
time=   0.0s
[CV] END ............................alpha=0.1, l1_ratio=0.5; total
time=   0.0s
[CV] END ............................alpha=0.1, l1_ratio=0.7; total
time=   0.0s
[CV] END ............................alpha=0.1, l1_ratio=0.7; total
time=   0.0s
[CV] END ............................alpha=0.1, l1_ratio=0.7; total
time=   0.0s
[CV] END ............................alpha=0.1, l1_ratio=0.7; total
time=   0.0s
[CV] END ............................alpha=0.1, l1_ratio=0.7; total
time=   0.0s
[CV] END ............................alpha=0.1, l1_ratio=0.9; total
time=   0.0s
[CV] END ............................alpha=0.1, l1_ratio=0.9; total
time=   0.0s
```

```
[CV] END ............................alpha=0.1, l1_ratio=0.9; total
time=   0.0s
[CV] END ............................alpha=0.1, l1_ratio=0.9; total
time=   0.0s
[CV] END ............................alpha=0.1, l1_ratio=0.9; total
time=   0.0s
[CV] END ...........................alpha=0.1, l1_ratio=0.95; total
time=   0.0s
[CV] END ...........................alpha=0.1, l1_ratio=0.95; total
time=   0.0s
[CV] END ...........................alpha=0.1, l1_ratio=0.95; total
time=   0.0s
[CV] END ...........................alpha=0.1, l1_ratio=0.95; total
time=   0.0s
[CV] END ...........................alpha=0.1, l1_ratio=0.95; total
time=   0.0s
[CV] END ...........................alpha=0.1, l1_ratio=0.99; total
time=   0.0s
[CV] END ...........................alpha=0.1, l1_ratio=0.99; total
time=   0.0s
[CV] END ...........................alpha=0.1, l1_ratio=0.99; total
time=   0.0s
[CV] END ...........................alpha=0.1, l1_ratio=0.99; total
time=   0.0s
[CV] END ...........................alpha=0.1, l1_ratio=0.99; total
time=   0.0s
[CV] END ..............................alpha=0.1, l1_ratio=1; total
time=   0.0s
[CV] END ..............................alpha=0.1, l1_ratio=1; total
time=   0.0s
[CV] END ..............................alpha=0.1, l1_ratio=1; total
time=   0.0s
[CV] END ..............................alpha=0.1, l1_ratio=1; total
time=   0.0s
[CV] END ..............................alpha=0.1, l1_ratio=1; total
time=   0.0s
[CV] END ..............................alpha=1, l1_ratio=0.1; total
time=   0.0s
[CV] END ..............................alpha=1, l1_ratio=0.1; total
time=   0.0s
[CV] END ..............................alpha=1, l1_ratio=0.1; total
time=   0.0s
[CV] END ..............................alpha=1, l1_ratio=0.1; total
time=   0.0s
[CV] END ..............................alpha=1, l1_ratio=0.1; total
time=   0.0s
[CV] END ..............................alpha=1, l1_ratio=0.5; total
time=   0.0s
[CV] END ..............................alpha=1, l1_ratio=0.5; total
time=   0.0s
```

```
[CV] END ..............................alpha=1, l1_ratio=0.5; total
time=   0.0s
[CV] END ..............................alpha=1, l1_ratio=0.5; total
time=   0.0s
[CV] END ..............................alpha=1, l1_ratio=0.5; total
time=   0.0s
[CV] END ..............................alpha=1, l1_ratio=0.7; total
time=   0.0s
[CV] END ..............................alpha=1, l1_ratio=0.7; total
time=   0.0s
[CV] END ..............................alpha=1, l1_ratio=0.7; total
time=   0.0s
[CV] END ..............................alpha=1, l1_ratio=0.7; total
time=   0.0s
[CV] END ..............................alpha=1, l1_ratio=0.7; total
time=   0.0s
[CV] END ..............................alpha=1, l1_ratio=0.9; total
time=   0.0s
[CV] END ..............................alpha=1, l1_ratio=0.9; total
time=   0.0s
[CV] END ..............................alpha=1, l1_ratio=0.9; total
time=   0.0s
[CV] END ..............................alpha=1, l1_ratio=0.9; total
time=   0.0s
[CV] END ..............................alpha=1, l1_ratio=0.9; total
time=   0.0s
[CV] END .............................alpha=1, l1_ratio=0.95; total
time=   0.0s
[CV] END .............................alpha=1, l1_ratio=0.95; total
time=   0.0s
[CV] END .............................alpha=1, l1_ratio=0.95; total
time=   0.0s
[CV] END .............................alpha=1, l1_ratio=0.95; total
time=   0.0s
[CV] END .............................alpha=1, l1_ratio=0.95; total
time=   0.0s
[CV] END .............................alpha=1, l1_ratio=0.99; total
time=   0.0s
[CV] END .............................alpha=1, l1_ratio=0.99; total
time=   0.0s
[CV] END .............................alpha=1, l1_ratio=0.99; total
time=   0.0s
[CV] END .............................alpha=1, l1_ratio=0.99; total
time=   0.0s
[CV] END .............................alpha=1, l1_ratio=0.99; total
time=   0.0s
[CV] END ................................alpha=1, l1_ratio=1; total
time=   0.0s
[CV] END ................................alpha=1, l1_ratio=1; total
time=   0.0s
```

```
[CV] END ................................alpha=1, l1_ratio=1; total
time=   0.0s
[CV] END ................................alpha=1, l1_ratio=1; total
time=   0.0s
[CV] END ................................alpha=1, l1_ratio=1; total
time=   0.0s
[CV] END ..............................alpha=5, l1_ratio=0.1; total
time=   0.0s
[CV] END ..............................alpha=5, l1_ratio=0.1; total
time=   0.0s
[CV] END ..............................alpha=5, l1_ratio=0.1; total
time=   0.0s
[CV] END ..............................alpha=5, l1_ratio=0.1; total
time=   0.0s
[CV] END ..............................alpha=5, l1_ratio=0.1; total
time=   0.0s
[CV] END ..............................alpha=5, l1_ratio=0.5; total
time=   0.0s
[CV] END ..............................alpha=5, l1_ratio=0.5; total
time=   0.0s
[CV] END ..............................alpha=5, l1_ratio=0.5; total
time=   0.0s
[CV] END ..............................alpha=5, l1_ratio=0.5; total
time=   0.0s
[CV] END ..............................alpha=5, l1_ratio=0.5; total
time=   0.0s
[CV] END ..............................alpha=5, l1_ratio=0.7; total
time=   0.0s
[CV] END ..............................alpha=5, l1_ratio=0.7; total
time=   0.0s
[CV] END ..............................alpha=5, l1_ratio=0.7; total
time=   0.0s
[CV] END ..............................alpha=5, l1_ratio=0.7; total
time=   0.0s
[CV] END ..............................alpha=5, l1_ratio=0.7; total
time=   0.0s
[CV] END ..............................alpha=5, l1_ratio=0.9; total
time=   0.0s
[CV] END ..............................alpha=5, l1_ratio=0.9; total
time=   0.0s
[CV] END ..............................alpha=5, l1_ratio=0.9; total
time=   0.0s
[CV] END ..............................alpha=5, l1_ratio=0.9; total
time=   0.0s
[CV] END ..............................alpha=5, l1_ratio=0.9; total
time=   0.0s
[CV] END .............................alpha=5, l1_ratio=0.95; total
time=   0.0s
[CV] END .............................alpha=5, l1_ratio=0.95; total
time=   0.0s
```

```
[CV] END .............................alpha=5, l1_ratio=0.95; total
time=   0.0s
[CV] END .............................alpha=5, l1_ratio=0.95; total
time=   0.0s
[CV] END .............................alpha=5, l1_ratio=0.95; total
time=   0.0s
[CV] END .............................alpha=5, l1_ratio=0.99; total
time=   0.0s
[CV] END .............................alpha=5, l1_ratio=0.99; total
time=   0.0s
[CV] END .............................alpha=5, l1_ratio=0.99; total
time=   0.0s
[CV] END .............................alpha=5, l1_ratio=0.99; total
time=   0.0s
[CV] END .............................alpha=5, l1_ratio=0.99; total
time=   0.0s
[CV] END ................................alpha=5, l1_ratio=1; total
time=   0.0s
[CV] END ................................alpha=5, l1_ratio=1; total
time=   0.0s
[CV] END ................................alpha=5, l1_ratio=1; total
time=   0.0s
[CV] END ................................alpha=5, l1_ratio=1; total
time=   0.0s
[CV] END ................................alpha=5, l1_ratio=1; total
time=   0.0s
[CV] END .............................alpha=10, l1_ratio=0.1; total
time=   0.0s
[CV] END .............................alpha=10, l1_ratio=0.1; total
time=   0.0s
[CV] END .............................alpha=10, l1_ratio=0.1; total
time=   0.0s
[CV] END .............................alpha=10, l1_ratio=0.1; total
time=   0.0s
[CV] END .............................alpha=10, l1_ratio=0.1; total
time=   0.0s
[CV] END .............................alpha=10, l1_ratio=0.5; total
time=   0.0s
[CV] END .............................alpha=10, l1_ratio=0.5; total
time=   0.0s
[CV] END .............................alpha=10, l1_ratio=0.5; total
time=   0.0s
[CV] END .............................alpha=10, l1_ratio=0.5; total
time=   0.0s
[CV] END .............................alpha=10, l1_ratio=0.5; total
time=   0.0s
[CV] END .............................alpha=10, l1_ratio=0.7; total
time=   0.0s
[CV] END .............................alpha=10, l1_ratio=0.7; total
time=   0.0s
```

```
[CV] END .............................alpha=10, l1_ratio=0.7; total
time=   0.0s
[CV] END .............................alpha=10, l1_ratio=0.7; total
time=   0.0s
[CV] END .............................alpha=10, l1_ratio=0.7; total
time=   0.0s
[CV] END .............................alpha=10, l1_ratio=0.9; total
time=   0.0s
[CV] END .............................alpha=10, l1_ratio=0.9; total
time=   0.0s
[CV] END .............................alpha=10, l1_ratio=0.9; total
time=   0.0s
[CV] END .............................alpha=10, l1_ratio=0.9; total
time=   0.0s
[CV] END .............................alpha=10, l1_ratio=0.9; total
time=   0.0s
[CV] END ............................alpha=10, l1_ratio=0.95; total
time=   0.0s
[CV] END ............................alpha=10, l1_ratio=0.95; total
time=   0.0s
[CV] END ............................alpha=10, l1_ratio=0.95; total
time=   0.0s
[CV] END ............................alpha=10, l1_ratio=0.95; total
time=   0.0s
[CV] END ............................alpha=10, l1_ratio=0.95; total
time=   0.0s
[CV] END ............................alpha=10, l1_ratio=0.99; total
time=   0.0s
[CV] END ............................alpha=10, l1_ratio=0.99; total
time=   0.0s
[CV] END ............................alpha=10, l1_ratio=0.99; total
time=   0.0s
[CV] END ............................alpha=10, l1_ratio=0.99; total
time=   0.0s
[CV] END ............................alpha=10, l1_ratio=0.99; total
time=   0.0s
[CV] END ...............................alpha=10, l1_ratio=1; total
time=   0.0s
[CV] END ...............................alpha=10, l1_ratio=1; total
time=   0.0s
[CV] END ...............................alpha=10, l1_ratio=1; total
time=   0.0s
[CV] END ...............................alpha=10, l1_ratio=1; total
time=   0.0s
[CV] END ...............................alpha=10, l1_ratio=1; total
time=   0.0s
[CV] END .............................alpha=50, l1_ratio=0.1; total
time=   0.0s
[CV] END .............................alpha=50, l1_ratio=0.1; total
time=   0.0s
```

```
[CV] END .............................alpha=50, l1_ratio=0.1; total
time=   0.0s
[CV] END .............................alpha=50, l1_ratio=0.1; total
time=   0.0s
[CV] END .............................alpha=50, l1_ratio=0.1; total
time=   0.0s
[CV] END .............................alpha=50, l1_ratio=0.5; total
time=   0.0s
[CV] END .............................alpha=50, l1_ratio=0.5; total
time=   0.0s
[CV] END .............................alpha=50, l1_ratio=0.5; total
time=   0.0s
[CV] END .............................alpha=50, l1_ratio=0.5; total
time=   0.0s
[CV] END .............................alpha=50, l1_ratio=0.5; total
time=   0.0s
[CV] END .............................alpha=50, l1_ratio=0.7; total
time=   0.0s
[CV] END .............................alpha=50, l1_ratio=0.7; total
time=   0.0s
[CV] END .............................alpha=50, l1_ratio=0.7; total
time=   0.0s
[CV] END .............................alpha=50, l1_ratio=0.7; total
time=   0.0s
[CV] END .............................alpha=50, l1_ratio=0.7; total
time=   0.0s
[CV] END .............................alpha=50, l1_ratio=0.9; total
time=   0.0s
[CV] END .............................alpha=50, l1_ratio=0.9; total
time=   0.0s
[CV] END .............................alpha=50, l1_ratio=0.9; total
time=   0.0s
[CV] END .............................alpha=50, l1_ratio=0.9; total
time=   0.0s
[CV] END .............................alpha=50, l1_ratio=0.9; total
time=   0.0s
[CV] END ............................alpha=50, l1_ratio=0.95; total
time=   0.0s
[CV] END ............................alpha=50, l1_ratio=0.95; total
time=   0.0s
[CV] END ............................alpha=50, l1_ratio=0.95; total
time=   0.0s
[CV] END ............................alpha=50, l1_ratio=0.95; total
time=   0.0s
[CV] END ............................alpha=50, l1_ratio=0.95; total
time=   0.0s
[CV] END ............................alpha=50, l1_ratio=0.99; total
time=   0.0s
[CV] END ............................alpha=50, l1_ratio=0.99; total
time=   0.0s
```

```
[CV] END ............................alpha=50, l1_ratio=0.99; total
time=   0.0s
[CV] END ............................alpha=50, l1_ratio=0.99; total
time=   0.0s
[CV] END ............................alpha=50, l1_ratio=0.99; total
time=   0.0s
[CV] END ...............................alpha=50, l1_ratio=1; total
time=   0.0s
[CV] END ...............................alpha=50, l1_ratio=1; total
time=   0.0s
[CV] END ...............................alpha=50, l1_ratio=1; total
time=   0.0s
[CV] END ...............................alpha=50, l1_ratio=1; total
time=   0.0s
[CV] END ...............................alpha=50, l1_ratio=1; total
time=   0.0s
[CV] END ............................alpha=100, l1_ratio=0.1; total
time=   0.0s
[CV] END ............................alpha=100, l1_ratio=0.1; total
time=   0.0s
[CV] END ............................alpha=100, l1_ratio=0.1; total
time=   0.0s
[CV] END ............................alpha=100, l1_ratio=0.1; total
time=   0.0s
[CV] END ............................alpha=100, l1_ratio=0.1; total
time=   0.0s
[CV] END ............................alpha=100, l1_ratio=0.5; total
time=   0.0s
[CV] END ............................alpha=100, l1_ratio=0.5; total
time=   0.0s
[CV] END ............................alpha=100, l1_ratio=0.5; total
time=   0.0s
[CV] END ............................alpha=100, l1_ratio=0.5; total
time=   0.0s
[CV] END ............................alpha=100, l1_ratio=0.5; total
time=   0.0s
[CV] END ............................alpha=100, l1_ratio=0.7; total
time=   0.0s
[CV] END ............................alpha=100, l1_ratio=0.7; total
time=   0.0s
[CV] END ............................alpha=100, l1_ratio=0.7; total
time=   0.0s
[CV] END ............................alpha=100, l1_ratio=0.7; total
time=   0.0s
[CV] END ............................alpha=100, l1_ratio=0.7; total
time=   0.0s
[CV] END ............................alpha=100, l1_ratio=0.9; total
time=   0.0s
[CV] END ............................alpha=100, l1_ratio=0.9; total
time=   0.0s
```

```
[CV] END ............................alpha=100, l1_ratio=0.9; total
time=   0.0s
[CV] END ............................alpha=100, l1_ratio=0.9; total
time=   0.0s
[CV] END ............................alpha=100, l1_ratio=0.9; total
time=   0.0s
[CV] END ...........................alpha=100, l1_ratio=0.95; total
time=   0.0s
[CV] END ...........................alpha=100, l1_ratio=0.95; total
time=   0.0s
[CV] END ...........................alpha=100, l1_ratio=0.95; total
time=   0.0s
[CV] END ...........................alpha=100, l1_ratio=0.95; total
time=   0.0s
[CV] END ...........................alpha=100, l1_ratio=0.95; total
time=   0.0s
[CV] END ...........................alpha=100, l1_ratio=0.99; total
time=   0.0s
[CV] END ...........................alpha=100, l1_ratio=0.99; total
time=   0.0s
[CV] END ...........................alpha=100, l1_ratio=0.99; total
time=   0.0s
[CV] END ...........................alpha=100, l1_ratio=0.99; total
time=   0.0s
[CV] END ...........................alpha=100, l1_ratio=0.99; total
time=   0.0s
[CV] END ..............................alpha=100, l1_ratio=1; total
time=   0.0s
[CV] END ..............................alpha=100, l1_ratio=1; total
time=   0.0s
[CV] END ..............................alpha=100, l1_ratio=1; total
time=   0.0s
[CV] END ..............................alpha=100, l1_ratio=1; total
time=   0.0s
[CV] END ..............................alpha=100, l1_ratio=1; total
time=   0.0s
```

```
GridSearchCV(cv=5, estimator=ElasticNet(),
             param_grid={'alpha': [0.1, 1, 5, 10, 50, 100],
                         'l1_ratio': [0.1, 0.5, 0.7, 0.9, 0.95, 0.99,
1]},
             scoring='neg_mean_squared_error', verbose=2)
```

```
grid_model.best_estimator_
```

```
ElasticNet(alpha=0.1, l1_ratio=1)
```

```
grid_model.best_params_
```

```
{'alpha': 0.1, 'l1_ratio': 1}
```

```python
# pd.DataFrame(grid_model.cv_results_)
```

Using Best Model From Grid Search

```python
y_pred = grid_model.predict(scaler_x_test)
```

```python
from sklearn.metrics import mean_squared_error
```

```python
mean_squared_error(Y_test,y_pred)
```

```
2.3873426420874737
```