

Untitled

October 9, 2020

```
In [48]: import numpy as np

In [31]: from sklearn.datasets import load_boston

In [32]: data = load_boston()

In [6]: data.keys()

Out[6]: dict_keys(['data', 'target', 'feature_names', 'DESCR'])

In [63]: x = data['data']
         y = data['target']

In [64]: x.shape

Out[64]: (506, 13)

In [34]: from sklearn.model_selection import train_test_split
         x_train, x_test, y_train, y_test = train_test_split(x, y, test_size = 0.2)

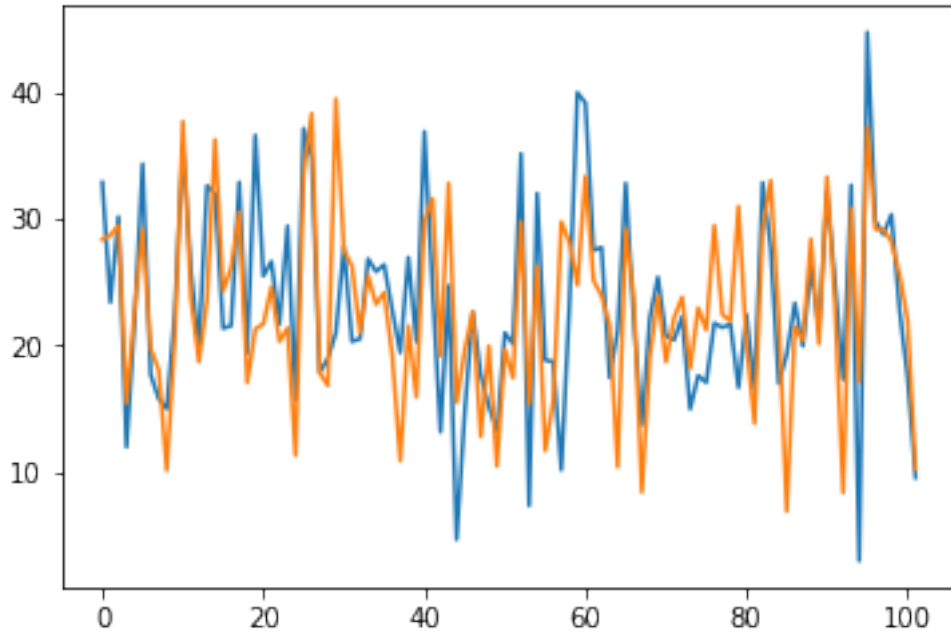
In [35]: from sklearn.linear_model import LinearRegression

In [36]: model = LinearRegression().fit(x_train,y_train)
         y_pred = model.predict(x_test)

In [26]: from sklearn.neighbors import KNeighborsRegressor
         model1 = KNeighborsRegressor().fit(x_train,y_train)
         y_pred_kn = model1.predict(x_test)

In [27]: import matplotlib.pyplot as plt

In [30]: plt.plot(range(len(y_pred)),y_pred)
         plt.plot(range(len(y_pred_kn)),y_pred_kn)
         plt.show()
```



```
In [37]: from sklearn.metrics import mean_absolute_error, r2_score
```

```
In [38]: mean_absolute_error(y_true=y_test,y_pred=y_pred)
```

```
Out[38]: 2.757729306076394
```

```
In [39]: r2_score(y_test,y_pred)
```

```
Out[39]: 0.8022469379448663
```

```
In [40]: from sklearn.model_selection import KFold
```

```
In [42]: kf = KFold(5)
         kf.split(x)
```

```
Out[42]: <generator object _BaseKFold.split at 0x1a13818af0>
```

```
In [61]: i = 0
         r2value = np.zeros((5,1))
         for train_index, test_index in kf.split(x):
             x_train = x[train_index]
             x_test = x[test_index]
             y_train = y[train_index]
             y_test = y[test_index]
             model = LinearRegression().fit(x_train,y_train)
             y_pred = model.predict(x_test)
             r2value[i] = r2_score(y_test,y_pred)
             i = i + 1
```

```
In [62]: r2value

Out[62]: array([[ 0.63861069],
                 [ 0.71334432],
                 [ 0.58645134],
                 [ 0.07842495],
                 [-0.26312455]])

In [65]: from sklearn.datasets import load_iris

In [66]: data = load_iris()

In [67]: data.keys()

Out[67]: dict_keys(['data', 'target', 'target_names', 'DESCR', 'feature_names'])

In [69]: x = data['data']
         y = data['target']

In [73]: x_train, x_test, y_train, y_test = train_test_split(x,y)

In [76]: from sklearn.linear_model import LogisticRegression
         model = LinearRegression().fit(x_train,y_train)
         y_pred_lr = model.predict(x_test)
```