

```
1 from sklearn.datasets import load_boston
2 from sklearn.linear_model import ElasticNet,ElasticNetCV
3 from sklearn.metrics import mean_squared_error
4 from sklearn.model_selection import train_test_split
5 import matplotlib.pyplot as plt
6 import numpy as np

1 alphas = [0.0001, 0.001, 0.01, 0.1, 0.3, 0.5, 0.7, 1]

1 boston = load_boston()
2 x, y = boston.data, boston.target
3 xtrain, xtest, ytrain, ytest = train_test_split(x, y, test_size=0.15)

1 for a in alphas:
2     model = ElasticNet(alpha=a).fit(x,y)
3     score = model.score(x, y)
4     pred_y = model.predict(x)
5     mse = mean_squared_error(y, pred_y)
6     print("Alpha:{0:.4f}, R2:{1:.2f}, MSE:{2:.2f}, RMSE:{3:.2f}"
7         .format(a, score, mse, np.sqrt(mse)))

↳ Alpha:0.0001, R2:0.74, MSE:21.90, RMSE:4.68
Alpha:0.0010, R2:0.74, MSE:21.92, RMSE:4.68
Alpha:0.0100, R2:0.74, MSE:22.32, RMSE:4.72
Alpha:0.1000, R2:0.73, MSE:23.16, RMSE:4.81
Alpha:0.3000, R2:0.71, MSE:24.08, RMSE:4.91
Alpha:0.5000, R2:0.71, MSE:24.85, RMSE:4.99
Alpha:0.7000, R2:0.70, MSE:25.55, RMSE:5.05
Alpha:1.0000, R2:0.69, MSE:26.50, RMSE:5.15

1 elastic=ElasticNet(alpha=0.01).fit(xtrain, ytrain)
2 ypred = elastic.predict(xtest)
3 score = elastic.score(xtest, ytest)
4 mse = mean_squared_error(ytest, ypred)
5 print("R2:{0:.3f}, MSE:{1:.2f}, RMSE:{2:.2f}"
6     .format(score, mse, np.sqrt(mse)))

↳ R2:0.714, MSE:22.49, RMSE:4.74
```

```
1 x_ax = range(len(xtest))
2 plt.scatter(x_ax, ytest, s=5, color="blue", label="original")
3 plt.plot(x_ax, ypred, lw=0.8, color="red", label="predicted")
4 plt.legend()
5 plt.title('ElasticNet Model')
6 plt.show()
```

