RIDGE

# example code

|  |
| --- |
| # Import list  import numpy as np  import pandas as pd  from matplotlib.pylab import rcParams  rcParams['figure.figsize'] = 8, 5  from sklearn.linear\_model import Ridge  import matplotlib.pyplot as plt  # Make X,y data set  x = np.array([i\*np.pi/180 for i in range(60,300,4)])  np.random.seed(10)  y = np.sin(x) + np.random.normal(0,0.15,len(x))  data = pd.DataFrame(np.column\_stack([x,y]), columns=['x','y'])  # Draw plot.  plt.plot(data['x'],data['y'], '.')  for i in range(2,16):      colname = 'x\_%d'%i      data[colname] = data['x']\*\*i  predictors=['x']  predictors.extend(['x\_%d'%i for i in range(2,16)])  # Split alpha value (parameters of Ridge)  alpha\_ridge = [1e-15,1e-10,1e-8,1e-4,1e-3,1e-2,1,5,10,20]  models\_to\_plot = {1e-15:231,1e-10:232,1e-4:233,1e-3:234,1e-2:235,5:236}  # Make table to save result.  col = ['rss','intercept'] + ['coef\_x\_%d'%i for i in range(1,16)]  ind = ['alpha\_%.2g'%alpha\_ridge[i] for i in range(0,10)]  coef\_matrix\_ridge = pd.DataFrame(index = ind, columns = col)  # def of drawing ridge\_regression graph.  def ridge\_regression(data,predictors,alpha,models\_to\_plot):      # Model train and predict      ridgereg = Ridge(alpha=alpha,normalize=True)      ridgereg.fit(data[predictors],data['y'])      y\_pred = ridgereg.predict(data[predictors])      # Draw plot of RIDGE regression      if alpha in models\_to\_plot:          plt.subplot(models\_to\_plot[alpha])          plt.tight\_layout()          plt.plot(data['x'],y\_pred)          plt.plot(data['x'],data['y'],'.')          plt.title("Plot for alpha : %.3g"%alpha)        rss = sum((y\_pred-data['y'])\*\*2)      ret = [rss]      ret.extend([ridgereg.intercept\_])      ret.extend(ridgereg.coef\_)      return ret  # Make Ridge\_Regression.  for i in range(10):      coef\_matrix\_ridge.iloc[i,] = ridge\_regression(data, predictors,alpha\_ridge[i],models\_to\_plot)  # print result table.  coef\_matrix\_ridge |

# testing result

|  |
| --- |
|  |