

## Writing my own class

#def class I will made as a Custom class.

class Meta\_Ridge\_TwoD:

def \_\_init\_\_(self, alpha):

self.alpha = alpha

self.m = None

self.b = None

$$m = \frac{\sum_{i=1}^n (Y_i - \bar{Y})(X_i - \bar{X})}{\sum_{i=1}^n (X_i - \bar{X})^2 + \lambda}$$

num  
deno

def fit(self, X\_train, Y\_train):

# Initially setting the values of num and deno = 0.

num = 0

deno = 0

for i in range(X\_train.shape[0]):

num = (num + (Y\_train[i] - Y\_train.mean()) \*  
(X\_train[i] - X\_train.mean()))

deno = (deno + (X\_train[i] - X\_train.mean()) \*  
(X\_train[i] - X\_train.mean()))

self.m = (num / (deno + self.alpha))

self.b = (Y\_train.mean() - (self.m \* X\_train.mean()))

def predict(self, X\_test):

return (self.m \* X\_test + self.b)

$$b = \bar{Y} - m\bar{X}$$