# Ridge Regression

**RidgeRegression(method = 'defaultDense', ridgeParameters = 1,interceptFlag = True, dtype = float64)**

Constructor to set Ridge Regression training parameters

*parameters:*

**method**: default: ' defaultDense '

Used to decide the calculation method. ' defaultDense ' is normal equation

**ridgeParameters:** int/list/numpy array, default: 1

All dependent variables will have the same lamba value if one value is given. A list/numpy array can also be given with different values for all dependent variables.

**interceptFlag:** True/False, default: 'True'

Decides whether or not intercept component to be evaluated

**dtype:** intc/float32, float64, default: float64

## Methods

1. **training(trainData, trainDependentVariables)**

*parameters:*

train data feature values(type nT), train data target values(type nT)

*returns*:

training results object

1. **predict(trainingResult, testData)**

*parameters:*

training result object, test data feature values(type nT)

*returns*:

predicted values of type nT

1. **compress(arrayData)**

*parameters:*

serialized numpy array

*returns*:

Compressed numpy array

1. **decompress(arrayData)**

*parameters:*

deserialized numpy array

*returns*:

decompressed numpy array

1. **serialize(data, fileName=None, useCompression=False)**

*parameters:*

Method 1: data(type nT/model)

-Returns serialized numpy array

Method 2: data(type nT/model), fileName(.npy file to save serialized array to disk)

- Saves serialized numpy array as "fileName" argument

Method 3: data(type nT/model), useCompression = True

-Returns compressed numpy array

Method 4: data(type nT/model), fileName(.npy file to save serialized array to disk), useCompression = True

-Saves compressed numpy array as "fileName" argument

1. **deserialize(serialObjectDict=None, fileName=None, useCompression=False)**

*parameters:*

serialized/ compressed numpy array or serialized/ compressed .npy file

*returns*:

deserialized/ decompressed numeric table/model