# Class Linear Regression:

## Methods:

1. **def \_\_init\_\_(self, method = 'normEq', interceptFlag = True, dtype = float64):**

Constructor to set Linear Regression training parameters

*parameters:*

**method**: 'normeq'/'qr', default: 'normeq'

Used to decide the calculation method. 'normeq' is normal equation, 'qr' is QR decomposition

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

Decides whether or not intercept component to be evaluated

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

1. **def training(self, trainData, trainDependentVariables):**

*parameters:*

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

*returns*:

training results object

1. **def predict(self, trainingResult, testData):**

*parameters:*

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

*returns*:

predicted values of type nT

1. **def compress(self, arrayData):**

*parameters:*

serialized numpy array

*returns*:

Compressed numpy array

1. **def decompress(self, arrayData):**

*parameters:*

deserialized numpy array

*returns*:

decompressed numpy array

1. **def serialize(self, 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. **def deserialize(self, serialObjectDict=None, fileName=None, useCompression=False):**

*parameters:*

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

*returns*:

deserialized/ decompressed numeric table/model

1. **def predictWithQualityMetrics(self, trainingResult, testData,testGroundTruth):**

*parameters*:

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

*returns*:

predicted values(type nT), reduced model predicted values (type nT), single beta metrics result, group beta metrics result

1. **def printAllQualityMetrics(self, resultSingleBeta, resultGroupBeta):**

*parameters*:

single beta metrics result, group beta metrics result

Prints RMSE, variance, z-score statistic, confidenceIntervals, inverseOfXtX matrix, variance-covariance matrix, expectedMean, expectedVariance, SSR, SST, R-square, f-statistic