# Linear Regression

**LinearRegression(method = ‘defaultDense’, interceptFlag = True, dtype = float64)**

Constructor to set Linear Regression training parameters

*parameters:*

**method**: ‘defaultDense’/’qrDense’, default: ' defaultDense '

Used to decide the calculation method. ' defaultDense ' is normal equation, ' qrDense ' is QR decomposition

**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

1. **predictReducedModelResults(trainingResult,trainData, reducedBeta,nDependentVariables=1 )**

*parameters:*

training result object, test data feature values of type nT, number of dependent variables, insignificant beta indexes (type list)

*returns* :

reduced model predicted values (type nT)

1. **qualityMetrics(trainingResult, predictResults, testGroundTruth,**

**predictedReducedModelResults=None,noReducedBetas = 1)**

*parameters:*

training result object, prediction values(type nT), test data actual target values(type nT), predictedReducedModelResults(type nT) ,insignificant beta indexes (type list)

*returns* :

qualityMetrics object with singleBeta and groupBeta attributes

1. **predictWithQualityMetrics(trainingResult, testData,testGroundTruth,** **reducedBetaIndex = None)**

*parameters*:

training result object, test data feature values of type nT, test data actual target values(type nT), insignificant beta indexes (type list)

*returns*:

predicted values(type nT), reduced model predicted values (type nT), qualityMetrics object with singleBeta and groupBeta attributes

1. **printAllQualityMetrics(qualityMet)**

*parameters*:

qualityMetrics object with singleBeta and groupBeta attributes

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