KMeansModel

NAME

KMeansModel - A data structure used in modeling the output of the frovedis server side kmeans clustering algorithm at client spark side.

SYNOPSIS

import com.nec.frovedis.mllib.clustering.KMeansModel

Public Member Functions

Int predict (Vector data)
`RDD[Int]` predict (`RDD[Vector]` data)
Int getK()
Unit save(String path)
Unit save(SparkContext sc, String path)
KMeansModel KMeansModel.load(String path)
KMeansModel KMeansModel.load(SparkContext sc, String path)
Unit debug_print()
Unit release()

DESCRIPTION

KMeansModel models the output of the frovedis kmeans clustering algorithm.

Note that the actual model with centroid information is created at frovedis server side only. Spark side KMeansModel contains a unique ID associated with the frovedis server side model, along with some generic information like k value etc. It simply works like a pointer to the in-memory model at frovedis server.

Any operations, like prediction etc. on a KMeansModel makes a request to the frovedis server along with the unique model ID and the actual job is served by the frovedis server. For functions which returns some output, the result is sent back from frovedis server to the spark client.

Pubic Member Function Documentation

Int predict (Vector data)

This function can be used when prediction is to be made on the trained model for a single sample. It returns with the predicted value from the frovedis server.

RDD[Int] predict (RDD[Vector] data)

This function can be used when prediction is to be made on the trained model for more than one samples distributed among spark workers.

It is performed by all the worker nodes in parallel and on success the function returns a `RDD[Int]` object containing the distributed predicted values at worker nodes.

Int getK()

It returns the number of clusters in the target model.

KMeansModel KMeansModel.load(String path)

This static function is used to load the target model with data in given filename stored at frovedis server side at specified location (filename with relative/absolute path) as little-endian binary data. On success, it returns the loaded model.

KMeansModel KMeansModel.load(SparkContext sc, String path)

This is Spark like static API provided for compatibility with spark code. But the "sc" parameter is simply ignored in this case and internally it calls the above load() method as "KMeansModel.load(path)".

Unit save(String path)

This function is used to save the target model with given filename. Note that the target model is saved at frovedis server side at specified location (filename with relative/absolute path) as little-endian binary data.

Unit save(SparkContext sc, String path)

This is Spark like API provided for compatibility with spark code. But the "sc" parameter is simply ignored in this case and internally it calls the above save() method as "save(path)".

Unit debug_print()

It prints the contents of the server side model on the server side user terminal. It is mainly useful for debugging purpose.

Unit release()

This function can be used to release the existing in-memory model at frovedis server side.