/\*\*

\* Train a K-means model on the given set of points; `data` should be cached for high

\* performance, because this is an iterative algorithm.

\*/

@Since("0.8.0")

def run(data: RDD[Vector]): KMeansModel = {

if (data.getStorageLevel == StorageLevel.NONE) {

logWarning("The input data is not directly cached, which may hurt performance if its"

+ " parent RDDs are also uncached.")

}

// Compute squared norms and cache them.

val norms = data.map(Vectors.norm(\_, 2.0))

norms.persist()

val zippedData = data.zip(norms).map { case (v, norm) =>

new VectorWithNorm(v, norm)

}

val model = runAlgorithm(zippedData)

norms.unpersist()

// Warn at the end of the run as well, for increased visibility.

if (data.getStorageLevel == StorageLevel.NONE) {

logWarning("The input data was not directly cached, which may hurt performance if its"

+ " parent RDDs are also uncached.")

}

model

}