package com.twitter.ann.scalding.offline

import com.twitter.ann.common.EmbeddingType.EmbeddingVector

import com.twitter.ann.common.{Distance, NeighborWithDistance, Queryable, RuntimeParams}

import com.twitter.util.Future

private[offline] case class ParameterlessQueryable[T, P <: RuntimeParams, D <: Distance[D]](

queryable: Queryable[T, P, D],

runtimeParamsForAllQueries: P) {

/\*\*

\* ANN query for ids with distance.

\*

\* @param embedding : Embedding/Vector to be queried with.

\* @param numOfNeighbors : Number of neighbours to be queried for.

\*

\* @return List of approximate nearest neighbour ids with distance from the query embedding.

\*/

def queryWithDistance(

embedding: EmbeddingVector,

numOfNeighbors: Int

): Future[List[NeighborWithDistance[T, D]]] =

queryable.queryWithDistance(embedding, numOfNeighbors, runtimeParamsForAllQueries)

}