package com.twitter.simclusters\_v2.hdfs\_sources.injections

import com.twitter.hermit.candidate.thriftscala.Candidates

import com.twitter.scalding\_internal.multiformat.format.keyval.KeyValInjection

import com.twitter.scalding\_internal.multiformat.format.keyval.KeyValInjection.{

Long2BigEndian,

ScalaBinaryThrift,

ScalaCompactThrift

}

import com.twitter.simclusters\_v2.thriftscala.{

PersistedFullClusterId,

SimClustersEmbedding,

SimClustersEmbeddingId,

TopProducersWithScore,

TopSimClustersWithScore

}

object ProducerEmbeddingsInjections {

final val ProducerTopKSimClusterEmbeddingsInjection: KeyValInjection[

Long,

TopSimClustersWithScore

] =

KeyValInjection(

keyCodec = Long2BigEndian,

valueCodec = ScalaCompactThrift(TopSimClustersWithScore))

final val SimClusterEmbeddingTopKProducersInjection: KeyValInjection[

PersistedFullClusterId,

TopProducersWithScore

] =

KeyValInjection(

keyCodec = ScalaCompactThrift(PersistedFullClusterId),

valueCodec = ScalaCompactThrift(TopProducersWithScore))

final val SimilarUsersInjection: KeyValInjection[Long, Candidates] =

KeyValInjection(keyCodec = Long2BigEndian, valueCodec = ScalaCompactThrift(Candidates))

final val ProducerSimClustersEmbeddingInjection: KeyValInjection[

SimClustersEmbeddingId,

SimClustersEmbedding

] =

KeyValInjection(

keyCodec = ScalaBinaryThrift(SimClustersEmbeddingId),

valueCodec = ScalaBinaryThrift(SimClustersEmbedding))

}