package com.twitter.follow\_recommendations.common.feature\_hydration.sources

import com.twitter.finagle.stats.StatsReceiver

import com.twitter.ml.api.DataRecord

import com.twitter.ml.api.IRecordOneToOneAdapter

import scala.util.Random

/\*\*

\* Helper functions for FeatureStoreSource operations in FRS are available here.

\*/

object Utils {

private val EarlyExpiration = 0.2

private[common] def adaptAdditionalFeaturesToDataRecord(

record: DataRecord,

adapterStats: StatsReceiver,

featureAdapters: Seq[IRecordOneToOneAdapter[DataRecord]]

): DataRecord = {

featureAdapters.foldRight(record) { (adapter, record) =>

adapterStats.counter(adapter.getClass.getSimpleName).incr()

adapter.adaptToDataRecord(record)

}

}

// To avoid a cache stampede. See https://en.wikipedia.org/wiki/Cache\_stampede

private[common] def randomizedTTL(ttl: Long): Long = {

(ttl - ttl \* EarlyExpiration \* Random.nextDouble()).toLong

}

}