package com.twitter.timelines.prediction.common.aggregates.real\_time

import com.twitter.ml.api.DataRecord

import com.twitter.ml.featurestore.lib.UserId

import com.twitter.ml.featurestore.lib.data.PredictionRecord

import com.twitter.ml.featurestore.lib.entity.Entity

import com.twitter.ml.featurestore.lib.online.{FeatureStoreClient, FeatureStoreRequest}

import com.twitter.storehaus.ReadableStore

import com.twitter.timelines.prediction.common.adapters.TimelinesAdapterBase

import com.twitter.util.Future

import scala.collection.JavaConverters.\_

class UserFeaturesReadableStore(

featureStoreClient: FeatureStoreClient,

userEntity: Entity[UserId],

userFeaturesAdapter: TimelinesAdapterBase[PredictionRecord])

extends ReadableStore[Set[Long], DataRecord] {

override def multiGet[K <: Set[Long]](keys: Set[K]): Map[K, Future[Option[DataRecord]]] = {

val orderedKeys = keys.toSeq

val featureStoreRequests: Seq[FeatureStoreRequest] = orderedKeys.map { key: Set[Long] =>

FeatureStoreRequest(

entityIds = key.map(userId => userEntity.withId(UserId(userId))).toSeq

)

}

val predictionRecordsFut: Future[Seq[PredictionRecord]] = featureStoreClient(

featureStoreRequests)

orderedKeys.zipWithIndex.map {

case (userId, index) =>

val dataRecordFutOpt = predictionRecordsFut.map { predictionRecords =>

userFeaturesAdapter.adaptToDataRecords(predictionRecords(index)).asScala.headOption

}

(userId, dataRecordFutOpt)

}.toMap

}

}