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

import com.twitter.ml.api.DataRecord

import com.twitter.ml.api.Feature

import com.twitter.ml.api.FeatureContext

import com.twitter.ml.featurestore.catalog.entities.core.Tweet

import com.twitter.ml.featurestore.catalog.features.trends.TweetTrendsScores

import com.twitter.ml.featurestore.lib.TweetId

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

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

import com.twitter.ml.featurestore.lib.feature.BoundFeature

import com.twitter.ml.featurestore.lib.feature.BoundFeatureSet

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

import java.util

import scala.collection.JavaConverters.\_

object TweetFeaturesAdapter extends TimelinesAdapterBase[PredictionRecord] {

private val ContinuousFeatureMap: Map[BoundFeature[TweetId, Double], Feature.Continuous] = Map()

val TweetFeaturesSet: BoundFeatureSet = new BoundFeatureSet(ContinuousFeatureMap.keys.toSet)

val AllFeatures: Seq[Feature[\_]] =

ContinuousFeatureMap.values.toSeq

private val adapter = PredictionRecordAdapter.oneToOne(TweetFeaturesSet)

override def getFeatureContext: FeatureContext = new FeatureContext(AllFeatures: \_\*)

override def commonFeatures: Set[Feature[\_]] = Set.empty

override def adaptToDataRecords(record: PredictionRecord): util.List[DataRecord] = {

List(adapter.adaptToDataRecord(record)).asJava

}

}