package com.twitter.home\_mixer.product.scored\_tweets.feature\_hydrator.adapters.non\_ml\_features

import com.twitter.ml.api.constant.SharedFeatures

import com.twitter.ml.api.Feature

import com.twitter.ml.api.FeatureContext

import com.twitter.ml.api.RichDataRecord

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

import com.twitter.timelines.prediction.features.common.TimelinesSharedFeatures

import java.lang.{Long => JLong}

case class NonMLCommonFeatures(

userId: Long,

predictionRequestId: Option[Long],

servedTimestamp: Long,

)

/\*\*

\* define non ml features adapter to create a data record which includes many non ml features

\* e.g. predictionRequestId, userId, tweetId to be used as joined key in batch pipeline.

\*/

object NonMLCommonFeaturesAdapter extends TimelinesMutatingAdapterBase[NonMLCommonFeatures] {

private val featureContext = new FeatureContext(

SharedFeatures.USER\_ID,

TimelinesSharedFeatures.PREDICTION\_REQUEST\_ID,

TimelinesSharedFeatures.SERVED\_TIMESTAMP,

)

override def getFeatureContext: FeatureContext = featureContext

override val commonFeatures: Set[Feature[\_]] = Set(

SharedFeatures.USER\_ID,

TimelinesSharedFeatures.PREDICTION\_REQUEST\_ID,

TimelinesSharedFeatures.SERVED\_TIMESTAMP,

)

override def setFeatures(

nonMLCommonFeatures: NonMLCommonFeatures,

richDataRecord: RichDataRecord

): Unit = {

richDataRecord.setFeatureValue[JLong](SharedFeatures.USER\_ID, nonMLCommonFeatures.userId)

nonMLCommonFeatures.predictionRequestId.foreach(

richDataRecord.setFeatureValue[JLong](TimelinesSharedFeatures.PREDICTION\_REQUEST\_ID, \_))

richDataRecord.setFeatureValue[JLong](

TimelinesSharedFeatures.SERVED\_TIMESTAMP,

nonMLCommonFeatures.servedTimestamp)

}

}