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 NonMLCandidateFeatures(

tweetId: Long,

sourceTweetId: Option[Long],

originalAuthorId: Option[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 NonMLCandidateFeaturesAdapter extends TimelinesMutatingAdapterBase[NonMLCandidateFeatures] {

private val featureContext = new FeatureContext(

SharedFeatures.TWEET\_ID,

// For Secondary Engagement data generation

TimelinesSharedFeatures.SOURCE\_TWEET\_ID,

TimelinesSharedFeatures.ORIGINAL\_AUTHOR\_ID,

)

override def getFeatureContext: FeatureContext = featureContext

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

override def setFeatures(

nonMLCandidateFeatures: NonMLCandidateFeatures,

richDataRecord: RichDataRecord

): Unit = {

richDataRecord.setFeatureValue[JLong](SharedFeatures.TWEET\_ID, nonMLCandidateFeatures.tweetId)

nonMLCandidateFeatures.sourceTweetId.foreach(

richDataRecord.setFeatureValue[JLong](TimelinesSharedFeatures.SOURCE\_TWEET\_ID, \_))

nonMLCandidateFeatures.originalAuthorId.foreach(

richDataRecord.setFeatureValue[JLong](TimelinesSharedFeatures.ORIGINAL\_AUTHOR\_ID, \_))

}

}