package com.twitter.home\_mixer.product.scored\_tweets.feature\_hydrator.adapters.content

import com.twitter.home\_mixer.model.ContentFeatures

import com.twitter.ml.api.Feature

import com.twitter.ml.api.FeatureContext

import com.twitter.ml.api.RichDataRecord

import com.twitter.ml.api.util.DataRecordConverters.\_

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

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

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

import com.twitter.timelines.prediction.features.conversation\_features.ConversationFeatures

import scala.collection.JavaConverters.\_

object ContentFeatureAdapter extends TimelinesMutatingAdapterBase[Option[ContentFeatures]] {

override val getFeatureContext: FeatureContext = new FeatureContext(

ConversationFeatures.IS\_SELF\_THREAD\_TWEET,

ConversationFeatures.IS\_LEAF\_IN\_SELF\_THREAD,

TimelinesSharedFeatures.ASPECT\_RATIO\_DEN,

TimelinesSharedFeatures.ASPECT\_RATIO\_NUM,

TimelinesSharedFeatures.BIT\_RATE,

TimelinesSharedFeatures.CLASSIFICATION\_LABELS,

TimelinesSharedFeatures.COLOR\_1\_BLUE,

TimelinesSharedFeatures.COLOR\_1\_GREEN,

TimelinesSharedFeatures.COLOR\_1\_PERCENTAGE,

TimelinesSharedFeatures.COLOR\_1\_RED,

TimelinesSharedFeatures.FACE\_AREAS,

TimelinesSharedFeatures.HAS\_APP\_INSTALL\_CALL\_TO\_ACTION,

TimelinesSharedFeatures.HAS\_DESCRIPTION,

TimelinesSharedFeatures.HAS\_QUESTION,

TimelinesSharedFeatures.HAS\_SELECTED\_PREVIEW\_IMAGE,

TimelinesSharedFeatures.HAS\_TITLE,

TimelinesSharedFeatures.HAS\_VISIT\_SITE\_CALL\_TO\_ACTION,

TimelinesSharedFeatures.HAS\_WATCH\_NOW\_CALL\_TO\_ACTION,

TimelinesSharedFeatures.HEIGHT\_1,

TimelinesSharedFeatures.HEIGHT\_2,

TimelinesSharedFeatures.HEIGHT\_3,

TimelinesSharedFeatures.HEIGHT\_4,

TimelinesSharedFeatures.IS\_360,

TimelinesSharedFeatures.IS\_EMBEDDABLE,

TimelinesSharedFeatures.IS\_MANAGED,

TimelinesSharedFeatures.IS\_MONETIZABLE,

TimelinesSharedFeatures.MEDIA\_PROVIDERS,

TimelinesSharedFeatures.NUM\_CAPS,

TimelinesSharedFeatures.NUM\_COLOR\_PALLETTE\_ITEMS,

TimelinesSharedFeatures.NUM\_FACES,

TimelinesSharedFeatures.NUM\_MEDIA\_TAGS,

TimelinesSharedFeatures.NUM\_NEWLINES,

TimelinesSharedFeatures.NUM\_STICKERS,

TimelinesSharedFeatures.NUM\_WHITESPACES,

TimelinesSharedFeatures.RESIZE\_METHOD\_1,

TimelinesSharedFeatures.RESIZE\_METHOD\_2,

TimelinesSharedFeatures.RESIZE\_METHOD\_3,

TimelinesSharedFeatures.RESIZE\_METHOD\_4,

TimelinesSharedFeatures.TWEET\_LENGTH,

TimelinesSharedFeatures.TWEET\_LENGTH\_TYPE,

TimelinesSharedFeatures.VIDEO\_DURATION,

TimelinesSharedFeatures.VIEW\_COUNT,

TimelinesSharedFeatures.WIDTH\_1,

TimelinesSharedFeatures.WIDTH\_2,

TimelinesSharedFeatures.WIDTH\_3,

TimelinesSharedFeatures.WIDTH\_4,

)

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

private def getTweetLengthType(tweetLength: Int): Long = {

tweetLength match {

case x if 0 > x || 280 < x => TweetLengthType.INVALID

case x if 0 <= x && x <= 30 => TweetLengthType.VERY\_SHORT

case x if 30 < x && x <= 60 => TweetLengthType.SHORT

case x if 60 < x && x <= 90 => TweetLengthType.MEDIUM

case x if 90 < x && x <= 140 => TweetLengthType.LENGTHY

case x if 140 < x && x <= 210 => TweetLengthType.VERY\_LENGTHY

case x if x > 210 => TweetLengthType.MAXIMUM\_LENGTH

}

}

override def setFeatures(

contentFeatures: Option[ContentFeatures],

richDataRecord: RichDataRecord

): Unit = {

if (contentFeatures.nonEmpty) {

val features = contentFeatures.get

// Conversation Features

richDataRecord.setFeatureValueFromOption(

ConversationFeatures.IS\_SELF\_THREAD\_TWEET,

Some(features.selfThreadMetadata.nonEmpty)

)

richDataRecord.setFeatureValueFromOption(

ConversationFeatures.IS\_LEAF\_IN\_SELF\_THREAD,

features.selfThreadMetadata.map(\_.isLeaf)

)

// Media Features

richDataRecord.setFeatureValueFromOption(

TimelinesSharedFeatures.ASPECT\_RATIO\_DEN,

features.aspectRatioDen.map(\_.toDouble)

)

richDataRecord.setFeatureValueFromOption(

TimelinesSharedFeatures.ASPECT\_RATIO\_NUM,

features.aspectRatioNum.map(\_.toDouble)

)

richDataRecord.setFeatureValueFromOption(

TimelinesSharedFeatures.BIT\_RATE,

features.bitRate.map(\_.toDouble)

)

richDataRecord.setFeatureValueFromOption(

TimelinesSharedFeatures.HEIGHT\_1,

features.heights.flatMap(\_.lift(0)).map(\_.toDouble)

)

richDataRecord.setFeatureValueFromOption(

TimelinesSharedFeatures.HEIGHT\_2,

features.heights.flatMap(\_.lift(1)).map(\_.toDouble)

)

richDataRecord.setFeatureValueFromOption(

TimelinesSharedFeatures.HEIGHT\_3,

features.heights.flatMap(\_.lift(2)).map(\_.toDouble)

)

richDataRecord.setFeatureValueFromOption(

TimelinesSharedFeatures.HEIGHT\_4,

features.heights.flatMap(\_.lift(3)).map(\_.toDouble)

)

richDataRecord.setFeatureValueFromOption(

TimelinesSharedFeatures.NUM\_MEDIA\_TAGS,

features.numMediaTags.map(\_.toDouble)

)

richDataRecord.setFeatureValueFromOption(

TimelinesSharedFeatures.RESIZE\_METHOD\_1,

features.resizeMethods.flatMap(\_.lift(0)).map(\_.toLong)

)

richDataRecord.setFeatureValueFromOption(

TimelinesSharedFeatures.RESIZE\_METHOD\_2,

features.resizeMethods.flatMap(\_.lift(1)).map(\_.toLong)

)

richDataRecord.setFeatureValueFromOption(

TimelinesSharedFeatures.RESIZE\_METHOD\_3,

features.resizeMethods.flatMap(\_.lift(2)).map(\_.toLong)

)

richDataRecord.setFeatureValueFromOption(

TimelinesSharedFeatures.RESIZE\_METHOD\_4,

features.resizeMethods.flatMap(\_.lift(3)).map(\_.toLong)

)

richDataRecord.setFeatureValueFromOption(

TimelinesSharedFeatures.VIDEO\_DURATION,

features.videoDurationMs.map(\_.toDouble)

)

richDataRecord.setFeatureValueFromOption(

TimelinesSharedFeatures.WIDTH\_1,

features.widths.flatMap(\_.lift(0)).map(\_.toDouble)

)

richDataRecord.setFeatureValueFromOption(

TimelinesSharedFeatures.WIDTH\_2,

features.widths.flatMap(\_.lift(1)).map(\_.toDouble)

)

richDataRecord.setFeatureValueFromOption(

TimelinesSharedFeatures.WIDTH\_3,

features.widths.flatMap(\_.lift(2)).map(\_.toDouble)

)

richDataRecord.setFeatureValueFromOption(

TimelinesSharedFeatures.WIDTH\_4,

features.widths.flatMap(\_.lift(3)).map(\_.toDouble)

)

richDataRecord.setFeatureValueFromOption(

TimelinesSharedFeatures.NUM\_COLOR\_PALLETTE\_ITEMS,

features.numColors.map(\_.toDouble)

)

richDataRecord.setFeatureValueFromOption(

TimelinesSharedFeatures.COLOR\_1\_RED,

features.dominantColorRed.map(\_.toDouble)

)

richDataRecord.setFeatureValueFromOption(

TimelinesSharedFeatures.COLOR\_1\_BLUE,

features.dominantColorBlue.map(\_.toDouble)

)

richDataRecord.setFeatureValueFromOption(

TimelinesSharedFeatures.COLOR\_1\_GREEN,

features.dominantColorGreen.map(\_.toDouble)

)

richDataRecord.setFeatureValueFromOption(

TimelinesSharedFeatures.COLOR\_1\_PERCENTAGE,

features.dominantColorPercentage

)

richDataRecord.setFeatureValueFromOption(

TimelinesSharedFeatures.MEDIA\_PROVIDERS,

features.mediaOriginProviders.map(\_.toSet.asJava)

)

richDataRecord.setFeatureValueFromOption(

TimelinesSharedFeatures.IS\_360,

features.is360

)

richDataRecord.setFeatureValueFromOption(

TimelinesSharedFeatures.VIEW\_COUNT,

features.viewCount.map(\_.toDouble)

)

richDataRecord.setFeatureValueFromOption(

TimelinesSharedFeatures.IS\_MANAGED,

features.isManaged

)

richDataRecord.setFeatureValueFromOption(

TimelinesSharedFeatures.IS\_MONETIZABLE,

features.isMonetizable

)

richDataRecord.setFeatureValueFromOption(

TimelinesSharedFeatures.IS\_EMBEDDABLE,

features.isEmbeddable

)

richDataRecord.setFeatureValueFromOption(

TimelinesSharedFeatures.NUM\_STICKERS,

features.stickerIds.map(\_.length.toDouble)

)

richDataRecord.setFeatureValueFromOption(

TimelinesSharedFeatures.NUM\_FACES,

features.faceAreas.map(\_.length.toDouble)

)

richDataRecord.setFeatureValueFromOption(

TimelinesSharedFeatures.FACE\_AREAS,

// guard for exception from max on empty seq

features.faceAreas.map(faceAreas =>

faceAreas.map(\_.toDouble).reduceOption(\_ max \_).getOrElse(0.0))

)

richDataRecord.setFeatureValueFromOption(

TimelinesSharedFeatures.HAS\_SELECTED\_PREVIEW\_IMAGE,

features.hasSelectedPreviewImage

)

richDataRecord.setFeatureValueFromOption(

TimelinesSharedFeatures.HAS\_TITLE,

features.hasTitle

)

richDataRecord.setFeatureValueFromOption(

TimelinesSharedFeatures.HAS\_DESCRIPTION,

features.hasDescription

)

richDataRecord.setFeatureValueFromOption(

TimelinesSharedFeatures.HAS\_VISIT\_SITE\_CALL\_TO\_ACTION,

features.hasVisitSiteCallToAction

)

richDataRecord.setFeatureValueFromOption(

TimelinesSharedFeatures.HAS\_APP\_INSTALL\_CALL\_TO\_ACTION,

features.hasAppInstallCallToAction

)

richDataRecord.setFeatureValueFromOption(

TimelinesSharedFeatures.HAS\_WATCH\_NOW\_CALL\_TO\_ACTION,

features.hasWatchNowCallToAction

)

// text features

richDataRecord.setFeatureValueFromOption(

TimelinesSharedFeatures.NUM\_CAPS,

Some(features.numCaps.toDouble)

)

richDataRecord.setFeatureValueFromOption(

TimelinesSharedFeatures.TWEET\_LENGTH,

Some(features.length.toDouble)

)

richDataRecord.setFeatureValueFromOption(

TimelinesSharedFeatures.TWEET\_LENGTH\_TYPE,

Some(getTweetLengthType(features.length.toInt))

)

richDataRecord.setFeatureValueFromOption(

TimelinesSharedFeatures.NUM\_WHITESPACES,

Some(features.numWhiteSpaces.toDouble)

)

richDataRecord.setFeatureValueFromOption(

TimelinesSharedFeatures.HAS\_QUESTION,

Some(features.hasQuestion)

)

richDataRecord.setFeatureValueFromOption(

TimelinesSharedFeatures.NUM\_NEWLINES,

features.numNewlines.map(\_.toDouble)

)

}

}

}