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.RichDataRecordWrapper

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

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

object InReplyToContentFeatureAdapter

extends TimelinesMutatingAdapterBase[Option[ContentFeatures]] {

override val getFeatureContext: FeatureContext = new FeatureContext(

// Media Features

InReplyToTweetTimelinesSharedFeatures.ASPECT\_RATIO\_DEN,

InReplyToTweetTimelinesSharedFeatures.ASPECT\_RATIO\_NUM,

InReplyToTweetTimelinesSharedFeatures.HEIGHT\_1,

InReplyToTweetTimelinesSharedFeatures.HEIGHT\_2,

InReplyToTweetTimelinesSharedFeatures.VIDEO\_DURATION,

// TextFeatures

InReplyToTweetTimelinesSharedFeatures.NUM\_CAPS,

InReplyToTweetTimelinesSharedFeatures.TWEET\_LENGTH,

InReplyToTweetTimelinesSharedFeatures.HAS\_QUESTION,

)

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

override def setFeatures(

contentFeatures: Option[ContentFeatures],

richDataRecord: RichDataRecord

): Unit = {

if (contentFeatures.nonEmpty) {

val features = contentFeatures.get

richDataRecord.setFeatureValueFromOption(

InReplyToTweetTimelinesSharedFeatures.ASPECT\_RATIO\_DEN,

features.aspectRatioNum.map(\_.toDouble)

)

richDataRecord.setFeatureValueFromOption(

InReplyToTweetTimelinesSharedFeatures.ASPECT\_RATIO\_NUM,

features.aspectRatioNum.map(\_.toDouble)

)

richDataRecord.setFeatureValueFromOption(

InReplyToTweetTimelinesSharedFeatures.HEIGHT\_1,

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

)

richDataRecord.setFeatureValueFromOption(

InReplyToTweetTimelinesSharedFeatures.HEIGHT\_2,

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

)

richDataRecord.setFeatureValueFromOption(

InReplyToTweetTimelinesSharedFeatures.VIDEO\_DURATION,

features.videoDurationMs.map(\_.toDouble)

)

richDataRecord.setFeatureValueFromOption(

InReplyToTweetTimelinesSharedFeatures.NUM\_CAPS,

Some(features.numCaps.toDouble)

)

richDataRecord.setFeatureValueFromOption(

InReplyToTweetTimelinesSharedFeatures.TWEET\_LENGTH,

Some(features.length.toDouble)

)

richDataRecord.setFeatureValueFromOption(

InReplyToTweetTimelinesSharedFeatures.HAS\_QUESTION,

Some(features.hasQuestion)

)

}

}

}