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

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.search.common.features.{thriftscala => sc}

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

import com.twitter.timelines.prediction.features.recap.RecapFeatures

import com.twitter.timelines.util.UrlExtractorUtil

import java.lang.{Boolean => JBoolean}

import java.lang.{Double => JDouble}

import java.util.{Map => JMap}

import scala.collection.JavaConverters.\_

object EarlybirdAdapter extends TimelinesMutatingAdapterBase[Option[sc.ThriftTweetFeatures]] {

override val getFeatureContext: FeatureContext = new FeatureContext(

RecapFeatures.BIDIRECTIONAL\_FAV\_COUNT,

RecapFeatures.BIDIRECTIONAL\_REPLY\_COUNT,

RecapFeatures.BIDIRECTIONAL\_RETWEET\_COUNT,

RecapFeatures.BLENDER\_SCORE,

RecapFeatures.CONTAINS\_MEDIA,

RecapFeatures.CONVERSATIONAL\_COUNT,

RecapFeatures.EMBEDS\_IMPRESSION\_COUNT,

RecapFeatures.EMBEDS\_URL\_COUNT,

RecapFeatures.FAV\_COUNT,

RecapFeatures.FAV\_COUNT\_V2,

RecapFeatures.FROM\_INACTIVE\_USER,

RecapFeatures.FROM\_MUTUAL\_FOLLOW,

RecapFeatures.FROM\_VERIFIED\_ACCOUNT,

RecapFeatures.HAS\_CARD,

RecapFeatures.HAS\_CONSUMER\_VIDEO,

RecapFeatures.HAS\_HASHTAG,

RecapFeatures.HAS\_IMAGE,

RecapFeatures.HAS\_LINK,

RecapFeatures.HAS\_MENTION,

RecapFeatures.HAS\_MULTIPLE\_HASHTAGS\_OR\_TRENDS,

RecapFeatures.HAS\_MULTIPLE\_MEDIA,

RecapFeatures.HAS\_NATIVE\_IMAGE,

RecapFeatures.HAS\_NATIVE\_VIDEO,

RecapFeatures.HAS\_NEWS,

RecapFeatures.HAS\_PERISCOPE,

RecapFeatures.HAS\_PRO\_VIDEO,

RecapFeatures.HAS\_TREND,

RecapFeatures.HAS\_VIDEO,

RecapFeatures.HAS\_VINE,

RecapFeatures.HAS\_VISIBLE\_LINK,

RecapFeatures.IS\_AUTHOR\_BOT,

RecapFeatures.IS\_AUTHOR\_NEW,

RecapFeatures.IS\_AUTHOR\_NSFW,

RecapFeatures.IS\_AUTHOR\_PROFILE\_EGG,

RecapFeatures.IS\_AUTHOR\_SPAM,

RecapFeatures.IS\_BUSINESS\_SCORE,

RecapFeatures.IS\_OFFENSIVE,

RecapFeatures.IS\_REPLY,

RecapFeatures.IS\_RETWEET,

RecapFeatures.IS\_RETWEETER\_BOT,

RecapFeatures.IS\_RETWEETER\_NEW,

RecapFeatures.IS\_RETWEETER\_NSFW,

RecapFeatures.IS\_RETWEETER\_PROFILE\_EGG,

RecapFeatures.IS\_RETWEETER\_SPAM,

RecapFeatures.IS\_RETWEET\_OF\_REPLY,

RecapFeatures.IS\_SENSITIVE,

RecapFeatures.LANGUAGE,

RecapFeatures.LINK\_COUNT,

RecapFeatures.LINK\_LANGUAGE,

RecapFeatures.MATCH\_SEARCHER\_LANGS,

RecapFeatures.MATCH\_SEARCHER\_MAIN\_LANG,

RecapFeatures.MATCH\_UI\_LANG,

RecapFeatures.MENTIONED\_SCREEN\_NAMES,

RecapFeatures.MENTION\_SEARCHER,

RecapFeatures.NUM\_HASHTAGS,

RecapFeatures.NUM\_MENTIONS,

RecapFeatures.PREV\_USER\_TWEET\_ENGAGEMENT,

RecapFeatures.PROBABLY\_FROM\_FOLLOWED\_AUTHOR,

RecapFeatures.REPLY\_COUNT,

RecapFeatures.REPLY\_COUNT\_V2,

RecapFeatures.REPLY\_OTHER,

RecapFeatures.REPLY\_SEARCHER,

RecapFeatures.RETWEET\_COUNT,

RecapFeatures.RETWEET\_COUNT\_V2,

RecapFeatures.RETWEET\_DIRECTED\_AT\_USER\_IN\_FIRST\_DEGREE,

RecapFeatures.RETWEET\_OF\_MUTUAL\_FOLLOW,

RecapFeatures.RETWEET\_OTHER,

RecapFeatures.RETWEET\_SEARCHER,

RecapFeatures.SIGNATURE,

RecapFeatures.SOURCE\_AUTHOR\_REP,

RecapFeatures.TEXT\_SCORE,

RecapFeatures.TWEET\_COUNT\_FROM\_USER\_IN\_SNAPSHOT,

RecapFeatures.UNIDIRECTIONAL\_FAV\_COUNT,

RecapFeatures.UNIDIRECTIONAL\_REPLY\_COUNT,

RecapFeatures.UNIDIRECTIONAL\_RETWEET\_COUNT,

RecapFeatures.URL\_DOMAINS,

RecapFeatures.USER\_REP,

RecapFeatures.VIDEO\_VIEW\_COUNT,

// shared features

TimelinesSharedFeatures.WEIGHTED\_FAV\_COUNT,

TimelinesSharedFeatures.WEIGHTED\_RETWEET\_COUNT,

TimelinesSharedFeatures.WEIGHTED\_REPLY\_COUNT,

TimelinesSharedFeatures.WEIGHTED\_QUOTE\_COUNT,

TimelinesSharedFeatures.EMBEDS\_IMPRESSION\_COUNT\_V2,

TimelinesSharedFeatures.EMBEDS\_URL\_COUNT\_V2,

TimelinesSharedFeatures.DECAYED\_FAVORITE\_COUNT,

TimelinesSharedFeatures.DECAYED\_RETWEET\_COUNT,

TimelinesSharedFeatures.DECAYED\_REPLY\_COUNT,

TimelinesSharedFeatures.DECAYED\_QUOTE\_COUNT,

TimelinesSharedFeatures.FAKE\_FAVORITE\_COUNT,

TimelinesSharedFeatures.FAKE\_RETWEET\_COUNT,

TimelinesSharedFeatures.FAKE\_REPLY\_COUNT,

TimelinesSharedFeatures.FAKE\_QUOTE\_COUNT,

TimelinesSharedFeatures.QUOTE\_COUNT,

TimelinesSharedFeatures.EARLYBIRD\_SCORE,

// Safety features

TimelinesSharedFeatures.LABEL\_ABUSIVE\_FLAG,

TimelinesSharedFeatures.LABEL\_ABUSIVE\_HI\_RCL\_FLAG,

TimelinesSharedFeatures.LABEL\_DUP\_CONTENT\_FLAG,

TimelinesSharedFeatures.LABEL\_NSFW\_HI\_PRC\_FLAG,

TimelinesSharedFeatures.LABEL\_NSFW\_HI\_RCL\_FLAG,

TimelinesSharedFeatures.LABEL\_SPAM\_FLAG,

TimelinesSharedFeatures.LABEL\_SPAM\_HI\_RCL\_FLAG,

// periscope features

TimelinesSharedFeatures.PERISCOPE\_EXISTS,

TimelinesSharedFeatures.PERISCOPE\_IS\_LIVE,

TimelinesSharedFeatures.PERISCOPE\_HAS\_BEEN\_FEATURED,

TimelinesSharedFeatures.PERISCOPE\_IS\_CURRENTLY\_FEATURED,

TimelinesSharedFeatures.PERISCOPE\_IS\_FROM\_QUALITY\_SOURCE,

// VISIBLE\_TOKEN\_RATIO

TimelinesSharedFeatures.VISIBLE\_TOKEN\_RATIO,

TimelinesSharedFeatures.HAS\_QUOTE,

TimelinesSharedFeatures.IS\_COMPOSER\_SOURCE\_CAMERA,

// health features

TimelinesSharedFeatures.PREPORTED\_TWEET\_SCORE,

// media

TimelinesSharedFeatures.CLASSIFICATION\_LABELS

)

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

override def setFeatures(

ebFeatures: Option[sc.ThriftTweetFeatures],

richDataRecord: RichDataRecord

): Unit = {

if (ebFeatures.nonEmpty) {

val features = ebFeatures.get

richDataRecord.setFeatureValue[JDouble](

RecapFeatures.PREV\_USER\_TWEET\_ENGAGEMENT,

features.prevUserTweetEngagement.toDouble

)

richDataRecord

.setFeatureValue[JBoolean](RecapFeatures.IS\_SENSITIVE, features.isSensitiveContent)

richDataRecord

.setFeatureValue[JBoolean](RecapFeatures.HAS\_MULTIPLE\_MEDIA, features.hasMultipleMedia)

richDataRecord

.setFeatureValue[JBoolean](RecapFeatures.IS\_AUTHOR\_PROFILE\_EGG, features.isAuthorProfileEgg)

richDataRecord.setFeatureValue[JBoolean](RecapFeatures.IS\_AUTHOR\_NEW, features.isAuthorNew)

richDataRecord

.setFeatureValue[JDouble](RecapFeatures.NUM\_MENTIONS, features.numMentions.toDouble)

richDataRecord.setFeatureValue[JBoolean](RecapFeatures.HAS\_MENTION, features.numMentions > 0)

richDataRecord

.setFeatureValue[JDouble](RecapFeatures.NUM\_HASHTAGS, features.numHashtags.toDouble)

richDataRecord.setFeatureValue[JBoolean](RecapFeatures.HAS\_HASHTAG, features.numHashtags > 0)

richDataRecord

.setFeatureValue[JDouble](RecapFeatures.LINK\_LANGUAGE, features.linkLanguage.toDouble)

richDataRecord.setFeatureValue[JBoolean](RecapFeatures.IS\_AUTHOR\_NSFW, features.isAuthorNSFW)

richDataRecord.setFeatureValue[JBoolean](RecapFeatures.IS\_AUTHOR\_SPAM, features.isAuthorSpam)

richDataRecord.setFeatureValue[JBoolean](RecapFeatures.IS\_AUTHOR\_BOT, features.isAuthorBot)

richDataRecord.setFeatureValueFromOption(

RecapFeatures.LANGUAGE,

features.language.map(\_.getValue.toLong))

richDataRecord.setFeatureValueFromOption(

RecapFeatures.SIGNATURE,

features.signature.map(\_.toLong))

richDataRecord

.setFeatureValue[JBoolean](RecapFeatures.FROM\_INACTIVE\_USER, features.fromInActiveUser)

richDataRecord

.setFeatureValue[JBoolean](

RecapFeatures.PROBABLY\_FROM\_FOLLOWED\_AUTHOR,

features.probablyFromFollowedAuthor)

richDataRecord

.setFeatureValue[JBoolean](RecapFeatures.FROM\_MUTUAL\_FOLLOW, features.fromMutualFollow)

richDataRecord.setFeatureValue[JBoolean](

RecapFeatures.FROM\_VERIFIED\_ACCOUNT,

features.fromVerifiedAccount)

richDataRecord.setFeatureValue[JDouble](RecapFeatures.USER\_REP, features.userRep)

richDataRecord

.setFeatureValue[JDouble](RecapFeatures.IS\_BUSINESS\_SCORE, features.isBusinessScore)

richDataRecord

.setFeatureValue[JBoolean](RecapFeatures.HAS\_CONSUMER\_VIDEO, features.hasConsumerVideo)

richDataRecord.setFeatureValue[JBoolean](RecapFeatures.HAS\_PRO\_VIDEO, features.hasProVideo)

richDataRecord.setFeatureValue[JBoolean](RecapFeatures.HAS\_VINE, features.hasVine)

richDataRecord.setFeatureValue[JBoolean](RecapFeatures.HAS\_PERISCOPE, features.hasPeriscope)

richDataRecord

.setFeatureValue[JBoolean](RecapFeatures.HAS\_NATIVE\_VIDEO, features.hasNativeVideo)

richDataRecord

.setFeatureValue[JBoolean](RecapFeatures.HAS\_NATIVE\_IMAGE, features.hasNativeImage)

richDataRecord.setFeatureValue[JBoolean](RecapFeatures.HAS\_CARD, features.hasCard)

richDataRecord.setFeatureValue[JBoolean](RecapFeatures.HAS\_IMAGE, features.hasImage)

richDataRecord.setFeatureValue[JBoolean](RecapFeatures.HAS\_NEWS, features.hasNews)

richDataRecord.setFeatureValue[JBoolean](RecapFeatures.HAS\_VIDEO, features.hasVideo)

richDataRecord.setFeatureValue[JBoolean](RecapFeatures.CONTAINS\_MEDIA, features.containsMedia)

richDataRecord

.setFeatureValue[JBoolean](RecapFeatures.RETWEET\_SEARCHER, features.retweetSearcher)

richDataRecord.setFeatureValue[JBoolean](RecapFeatures.REPLY\_SEARCHER, features.replySearcher)

richDataRecord

.setFeatureValue[JBoolean](RecapFeatures.MENTION\_SEARCHER, features.mentionSearcher)

richDataRecord.setFeatureValue[JBoolean](RecapFeatures.REPLY\_OTHER, features.replyOther)

richDataRecord.setFeatureValue[JBoolean](RecapFeatures.RETWEET\_OTHER, features.retweetOther)

richDataRecord.setFeatureValue[JBoolean](RecapFeatures.IS\_REPLY, features.isReply)

richDataRecord.setFeatureValue[JBoolean](RecapFeatures.IS\_RETWEET, features.isRetweet)

richDataRecord.setFeatureValue[JBoolean](RecapFeatures.IS\_OFFENSIVE, features.isOffensive)

richDataRecord.setFeatureValue[JBoolean](RecapFeatures.MATCH\_UI\_LANG, features.matchesUILang)

richDataRecord

.setFeatureValue[JBoolean](

RecapFeatures.MATCH\_SEARCHER\_MAIN\_LANG,

features.matchesSearcherMainLang)

richDataRecord.setFeatureValue[JBoolean](

RecapFeatures.MATCH\_SEARCHER\_LANGS,

features.matchesSearcherLangs)

richDataRecord

.setFeatureValue[JDouble](

RecapFeatures.BIDIRECTIONAL\_FAV\_COUNT,

features.bidirectionalFavCount)

richDataRecord

.setFeatureValue[JDouble](

RecapFeatures.UNIDIRECTIONAL\_FAV\_COUNT,

features.unidirectionalFavCount)

richDataRecord

.setFeatureValue[JDouble](

RecapFeatures.BIDIRECTIONAL\_REPLY\_COUNT,

features.bidirectionalReplyCount)

richDataRecord

.setFeatureValue[JDouble](

RecapFeatures.UNIDIRECTIONAL\_REPLY\_COUNT,

features.unidirectionalReplyCount)

richDataRecord

.setFeatureValue[JDouble](

RecapFeatures.BIDIRECTIONAL\_RETWEET\_COUNT,

features.bidirectionalRetweetCount)

richDataRecord

.setFeatureValue[JDouble](

RecapFeatures.UNIDIRECTIONAL\_RETWEET\_COUNT,

features.unidirectionalRetweetCount)

richDataRecord

.setFeatureValue[JDouble](RecapFeatures.CONVERSATIONAL\_COUNT, features.conversationCount)

richDataRecord.setFeatureValue[JDouble](

RecapFeatures.TWEET\_COUNT\_FROM\_USER\_IN\_SNAPSHOT,

features.tweetCountFromUserInSnapshot

)

richDataRecord

.setFeatureValue[JBoolean](

RecapFeatures.IS\_RETWEETER\_PROFILE\_EGG,

features.isRetweeterProfileEgg)

richDataRecord

.setFeatureValue[JBoolean](RecapFeatures.IS\_RETWEETER\_NEW, features.isRetweeterNew)

richDataRecord

.setFeatureValue[JBoolean](RecapFeatures.IS\_RETWEETER\_BOT, features.isRetweeterBot)

richDataRecord

.setFeatureValue[JBoolean](RecapFeatures.IS\_RETWEETER\_NSFW, features.isRetweeterNSFW)

richDataRecord

.setFeatureValue[JBoolean](RecapFeatures.IS\_RETWEETER\_SPAM, features.isRetweeterSpam)

richDataRecord

.setFeatureValue[JBoolean](

RecapFeatures.RETWEET\_OF\_MUTUAL\_FOLLOW,

features.retweetOfMutualFollow)

richDataRecord

.setFeatureValue[JDouble](RecapFeatures.SOURCE\_AUTHOR\_REP, features.sourceAuthorRep)

richDataRecord

.setFeatureValue[JBoolean](RecapFeatures.IS\_RETWEET\_OF\_REPLY, features.isRetweetOfReply)

richDataRecord.setFeatureValueFromOption(

RecapFeatures.RETWEET\_DIRECTED\_AT\_USER\_IN\_FIRST\_DEGREE,

features.retweetDirectedAtUserInFirstDegree

)

richDataRecord

.setFeatureValue[JDouble](

RecapFeatures.EMBEDS\_IMPRESSION\_COUNT,

features.embedsImpressionCount.toDouble)

richDataRecord

.setFeatureValue[JDouble](RecapFeatures.EMBEDS\_URL\_COUNT, features.embedsUrlCount.toDouble)

richDataRecord

.setFeatureValue[JDouble](RecapFeatures.VIDEO\_VIEW\_COUNT, features.videoViewCount.toDouble)

richDataRecord

.setFeatureValue[JDouble](RecapFeatures.REPLY\_COUNT, features.replyCount.toDouble)

richDataRecord

.setFeatureValue[JDouble](RecapFeatures.RETWEET\_COUNT, features.retweetCount.toDouble)

richDataRecord.setFeatureValue[JDouble](RecapFeatures.FAV\_COUNT, features.favCount.toDouble)

richDataRecord.setFeatureValue[JDouble](RecapFeatures.BLENDER\_SCORE, features.blenderScore)

richDataRecord.setFeatureValue[JDouble](RecapFeatures.TEXT\_SCORE, features.textScore)

richDataRecord

.setFeatureValue[JBoolean](RecapFeatures.HAS\_VISIBLE\_LINK, features.hasVisibleLink)

richDataRecord.setFeatureValue[JBoolean](RecapFeatures.HAS\_LINK, features.hasLink)

richDataRecord.setFeatureValue[JBoolean](RecapFeatures.HAS\_TREND, features.hasTrend)

richDataRecord.setFeatureValue[JBoolean](

RecapFeatures.HAS\_MULTIPLE\_HASHTAGS\_OR\_TRENDS,

features.hasMultipleHashtagsOrTrends

)

richDataRecord.setFeatureValueFromOption(

RecapFeatures.FAV\_COUNT\_V2,

features.favCountV2.map(\_.toDouble))

richDataRecord.setFeatureValueFromOption(

RecapFeatures.RETWEET\_COUNT\_V2,

features.retweetCountV2.map(\_.toDouble)

)

richDataRecord.setFeatureValueFromOption(

RecapFeatures.REPLY\_COUNT\_V2,

features.replyCountV2.map(\_.toDouble))

richDataRecord.setFeatureValueFromOption(

RecapFeatures.MENTIONED\_SCREEN\_NAMES,

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

)

val urls = features.urlsList.getOrElse(Seq.empty)

richDataRecord.setFeatureValue(

RecapFeatures.URL\_DOMAINS,

urls.toSet.flatMap(UrlExtractorUtil.extractDomain).asJava)

richDataRecord.setFeatureValue[JDouble](RecapFeatures.LINK\_COUNT, urls.size.toDouble)

// shared features

richDataRecord.setFeatureValueFromOption(

TimelinesSharedFeatures.WEIGHTED\_FAV\_COUNT,

features.weightedFavoriteCount.map(\_.toDouble)

)

richDataRecord.setFeatureValueFromOption(

TimelinesSharedFeatures.WEIGHTED\_RETWEET\_COUNT,

features.weightedRetweetCount.map(\_.toDouble)

)

richDataRecord.setFeatureValueFromOption(

TimelinesSharedFeatures.WEIGHTED\_REPLY\_COUNT,

features.weightedReplyCount.map(\_.toDouble)

)

richDataRecord.setFeatureValueFromOption(

TimelinesSharedFeatures.WEIGHTED\_QUOTE\_COUNT,

features.weightedQuoteCount.map(\_.toDouble)

)

richDataRecord.setFeatureValueFromOption(

TimelinesSharedFeatures.EMBEDS\_IMPRESSION\_COUNT\_V2,

features.embedsImpressionCountV2.map(\_.toDouble)

)

richDataRecord.setFeatureValueFromOption(

TimelinesSharedFeatures.EMBEDS\_URL\_COUNT\_V2,

features.embedsUrlCountV2.map(\_.toDouble)

)

richDataRecord.setFeatureValueFromOption(

TimelinesSharedFeatures.DECAYED\_FAVORITE\_COUNT,

features.decayedFavoriteCount.map(\_.toDouble)

)

richDataRecord.setFeatureValueFromOption(

TimelinesSharedFeatures.DECAYED\_RETWEET\_COUNT,

features.decayedRetweetCount.map(\_.toDouble)

)

richDataRecord.setFeatureValueFromOption(

TimelinesSharedFeatures.DECAYED\_REPLY\_COUNT,

features.decayedReplyCount.map(\_.toDouble)

)

richDataRecord.setFeatureValueFromOption(

TimelinesSharedFeatures.DECAYED\_QUOTE\_COUNT,

features.decayedQuoteCount.map(\_.toDouble)

)

richDataRecord.setFeatureValueFromOption(

TimelinesSharedFeatures.FAKE\_FAVORITE\_COUNT,

features.fakeFavoriteCount.map(\_.toDouble)

)

richDataRecord.setFeatureValueFromOption(

TimelinesSharedFeatures.FAKE\_RETWEET\_COUNT,

features.fakeRetweetCount.map(\_.toDouble)

)

richDataRecord.setFeatureValueFromOption(

TimelinesSharedFeatures.FAKE\_REPLY\_COUNT,

features.fakeReplyCount.map(\_.toDouble)

)

richDataRecord.setFeatureValueFromOption(

TimelinesSharedFeatures.FAKE\_QUOTE\_COUNT,

features.fakeQuoteCount.map(\_.toDouble)

)

richDataRecord.setFeatureValueFromOption(

TimelinesSharedFeatures.QUOTE\_COUNT,

features.quoteCount.map(\_.toDouble)

)

richDataRecord.setFeatureValue[JDouble](

TimelinesSharedFeatures.EARLYBIRD\_SCORE,

features.earlybirdScore

)

// safety features

richDataRecord.setFeatureValueFromOption(

TimelinesSharedFeatures.LABEL\_ABUSIVE\_FLAG,

features.labelAbusiveFlag

)

richDataRecord.setFeatureValueFromOption(

TimelinesSharedFeatures.LABEL\_ABUSIVE\_HI\_RCL\_FLAG,

features.labelAbusiveHiRclFlag

)

richDataRecord.setFeatureValueFromOption(

TimelinesSharedFeatures.LABEL\_DUP\_CONTENT\_FLAG,

features.labelDupContentFlag

)

richDataRecord.setFeatureValueFromOption(

TimelinesSharedFeatures.LABEL\_NSFW\_HI\_PRC\_FLAG,

features.labelNsfwHiPrcFlag

)

richDataRecord.setFeatureValueFromOption(

TimelinesSharedFeatures.LABEL\_NSFW\_HI\_RCL\_FLAG,

features.labelNsfwHiRclFlag

)

richDataRecord.setFeatureValueFromOption(

TimelinesSharedFeatures.LABEL\_SPAM\_FLAG,

features.labelSpamFlag

)

richDataRecord.setFeatureValueFromOption(

TimelinesSharedFeatures.LABEL\_SPAM\_HI\_RCL\_FLAG,

features.labelSpamHiRclFlag

)

// periscope features

richDataRecord.setFeatureValueFromOption(

TimelinesSharedFeatures.PERISCOPE\_EXISTS,

features.periscopeExists

)

richDataRecord.setFeatureValueFromOption(

TimelinesSharedFeatures.PERISCOPE\_IS\_LIVE,

features.periscopeIsLive

)

richDataRecord.setFeatureValueFromOption(

TimelinesSharedFeatures.PERISCOPE\_HAS\_BEEN\_FEATURED,

features.periscopeHasBeenFeatured

)

richDataRecord.setFeatureValueFromOption(

TimelinesSharedFeatures.PERISCOPE\_IS\_CURRENTLY\_FEATURED,

features.periscopeIsCurrentlyFeatured

)

richDataRecord.setFeatureValueFromOption(

TimelinesSharedFeatures.PERISCOPE\_IS\_FROM\_QUALITY\_SOURCE,

features.periscopeIsFromQualitySource

)

// misc features

richDataRecord.setFeatureValueFromOption(

TimelinesSharedFeatures.VISIBLE\_TOKEN\_RATIO,

features.visibleTokenRatio.map(\_.toDouble)

)

richDataRecord.setFeatureValueFromOption(

TimelinesSharedFeatures.HAS\_QUOTE,

features.hasQuote

)

richDataRecord.setFeatureValueFromOption(

TimelinesSharedFeatures.IS\_COMPOSER\_SOURCE\_CAMERA,

features.isComposerSourceCamera

)

// health scores

richDataRecord.setFeatureValueFromOption(

TimelinesSharedFeatures.PREPORTED\_TWEET\_SCORE,

features.pReportedTweetScore

)

// media

richDataRecord.setFeatureValueFromOption(

TimelinesSharedFeatures.CLASSIFICATION\_LABELS,

features.mediaClassificationInfo.map(\_.toMap.asJava.asInstanceOf[JMap[String, JDouble]])

)

}

}

}