package com.twitter.home\_mixer.product.scored\_tweets.response\_transformer

import com.twitter.tweet\_mixer.{thriftscala => tmt}

import com.twitter.home\_mixer.model.HomeFeatures.\_

import com.twitter.product\_mixer.core.feature.Feature

import com.twitter.product\_mixer.core.feature.featuremap.FeatureMap

import com.twitter.product\_mixer.core.feature.featuremap.FeatureMapBuilder

import com.twitter.product\_mixer.core.functional\_component.transformer.CandidateFeatureTransformer

import com.twitter.product\_mixer.core.model.common.identifier.TransformerIdentifier

import com.twitter.timelineservice.suggests.logging.candidate\_tweet\_source\_id.{thriftscala => cts}

import com.twitter.timelineservice.suggests.{thriftscala => st}

import com.twitter.tsp.{thriftscala => tsp}

object ScoredTweetsTweetMixerResponseFeatureTransformer

extends CandidateFeatureTransformer[tmt.TweetResult] {

override val identifier: TransformerIdentifier =

TransformerIdentifier("ScoredTweetsTweetMixerResponse")

override val features: Set[Feature[\_, \_]] = Set(

CandidateSourceIdFeature,

FromInNetworkSourceFeature,

IsRandomTweetFeature,

StreamToKafkaFeature,

SuggestTypeFeature,

TSPMetricTagFeature

)

override def transform(candidate: tmt.TweetResult): FeatureMap = {

val tweetMixerMetricTags = candidate.metricTags.getOrElse(Seq.empty)

val tspMetricTag = tweetMixerMetricTags

.map(TweetMixerMetricTagToTspMetricTag)

.filter(\_.nonEmpty).map(\_.get).toSet

FeatureMapBuilder()

.add(CandidateSourceIdFeature, Some(cts.CandidateTweetSourceId.Simcluster))

.add(FromInNetworkSourceFeature, false)

.add(IsRandomTweetFeature, false)

.add(StreamToKafkaFeature, true)

.add(SuggestTypeFeature, Some(st.SuggestType.ScTweet))

.add(TSPMetricTagFeature, tspMetricTag)

.build()

}

private def TweetMixerMetricTagToTspMetricTag(

tweetMixerMetricTag: tmt.MetricTag

): Option[tsp.MetricTag] = tweetMixerMetricTag match {

case tmt.MetricTag.TweetFavorite => Some(tsp.MetricTag.TweetFavorite)

case tmt.MetricTag.Retweet => Some(tsp.MetricTag.Retweet)

case \_ => None

}

}