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

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

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

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.timelineranker.{thriftscala => tlr}

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

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

object ScoredTweetsUtegResponseFeatureTransformer

extends CandidateFeatureTransformer[tlr.CandidateTweet] {

override val identifier: TransformerIdentifier = TransformerIdentifier("ScoredTweetsUtegResponse")

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

override def transform(candidate: tlr.CandidateTweet): FeatureMap = {

val baseFeatures = TimelineRankerResponseTransformer.transform(candidate)

val features = FeatureMapBuilder()

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

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

.build()

baseFeatures ++ features

}

}