package com.twitter.home\_mixer.candidate\_pipeline

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

import com.twitter.product\_mixer.component\_library.candidate\_source.tweetconvosvc.TweetWithConversationMetadata

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.thriftscala.SuggestType

object ConversationServiceResponseFeatureTransformer

extends CandidateFeatureTransformer[TweetWithConversationMetadata] {

override val identifier: TransformerIdentifier =

TransformerIdentifier("ConversationServiceResponse")

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

AuthorIdFeature,

InReplyToTweetIdFeature,

IsRetweetFeature,

SourceTweetIdFeature,

SourceUserIdFeature,

ConversationModuleFocalTweetIdFeature,

AncestorsFeature,

SuggestTypeFeature

)

override def transform(candidate: TweetWithConversationMetadata): FeatureMap = FeatureMapBuilder()

.add(AuthorIdFeature, candidate.userId)

.add(InReplyToTweetIdFeature, candidate.inReplyToTweetId)

.add(IsRetweetFeature, candidate.sourceTweetId.isDefined)

.add(SourceTweetIdFeature, candidate.sourceTweetId)

.add(SourceUserIdFeature, candidate.sourceUserId)

.add(ConversationModuleFocalTweetIdFeature, candidate.conversationId)

.add(AncestorsFeature, candidate.ancestors)

.add(SuggestTypeFeature, Some(SuggestType.RankedOrganicTweet))

.build()

}