package com.twitter.product\_mixer.component\_library.pipeline.candidate.who\_to\_follow\_module

import com.twitter.adserver.{thriftscala => ad}

import com.twitter.hermit.{thriftscala => h}

import com.twitter.peoplediscovery.api.{thriftscala => t}

import com.twitter.product\_mixer.component\_library.model.candidate.UserCandidate

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

object AdImpressionFeature extends Feature[UserCandidate, Option[ad.AdImpression]]

object HermitContextTypeFeature extends Feature[UserCandidate, Option[h.ContextType]]

object SocialTextFeature extends Feature[UserCandidate, Option[String]]

object TrackingTokenFeature extends Feature[UserCandidate, Option[String]]

object ScoreFeature extends Feature[UserCandidate, Option[Double]]

object WhoToFollowResponseFeatureTransformer

extends CandidateFeatureTransformer[t.RecommendedUser] {

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

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

Set(

AdImpressionFeature,

HermitContextTypeFeature,

SocialTextFeature,

TrackingTokenFeature,

ScoreFeature)

override def transform(input: t.RecommendedUser): FeatureMap = FeatureMapBuilder()

.add(AdImpressionFeature, input.adImpression)

.add(HermitContextTypeFeature, input.reason.flatMap(\_.contextType))

.add(SocialTextFeature, input.socialText)

.add(TrackingTokenFeature, input.trackingToken)

.add(ScoreFeature, input.mlPredictionScore)

.build()

}