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

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

import com.twitter.account\_recommendations\_mixer.{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 ContextTypeFeature extends Feature[UserCandidate, Option[t.ContextType]]

object WhoToFollowArmResponseFeatureTransformer

extends CandidateFeatureTransformer[t.RecommendedUser] {

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

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

Set(

AdImpressionFeature,

ContextTypeFeature,

HermitContextTypeFeature,

SocialTextFeature,

TrackingTokenFeature,

ScoreFeature)

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

.add(AdImpressionFeature, input.adImpression)

.add(ContextTypeFeature, input.contextType)

.add(

HermitContextTypeFeature,

input.contextType.map(contextType => h.ContextType(contextType.value)))

.add(SocialTextFeature, input.socialText)

.add(TrackingTokenFeature, input.trackingToken)

.add(ScoreFeature, input.mlPredictionScore)

.build()

}