package com.twitter.home\_mixer.product.scored\_tweets.feature\_hydrator

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

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

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

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

import com.twitter.product\_mixer.component\_library.feature\_hydrator.query.social\_graph.SGSFollowedUsersFeature

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

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.feature\_hydrator.BulkCandidateFeatureHydrator

import com.twitter.product\_mixer.core.model.common.CandidateWithFeatures

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

import com.twitter.product\_mixer.core.pipeline.PipelineQuery

import com.twitter.product\_mixer.core.util.OffloadFuturePools

import com.twitter.stitch.Stitch

object IsExtendedReplyFeatureHydrator

extends BulkCandidateFeatureHydrator[PipelineQuery, TweetCandidate] {

override val identifier: FeatureHydratorIdentifier = FeatureHydratorIdentifier("IsExtendedReply")

override def features: Set[Feature[\_, \_]] = Set(IsExtendedReplyFeature)

private val TrueFeatureMap = FeatureMapBuilder().add(IsExtendedReplyFeature, true).build()

private val FalseFeatureMap = FeatureMapBuilder().add(IsExtendedReplyFeature, false).build()

override def apply(

query: PipelineQuery,

candidates: Seq[CandidateWithFeatures[TweetCandidate]]

): Stitch[Seq[FeatureMap]] = OffloadFuturePools.offload {

val followedUsers =

query.features.map(\_.get(SGSFollowedUsersFeature)).getOrElse(Seq.empty).toSet

candidates.map { candidate =>

val features = candidate.features

val isExtendedReply = features.getOrElse(InReplyToTweetIdFeature, None).nonEmpty &&

!features.getOrElse(IsRetweetFeature, false) &&

features.getOrElse(InReplyToUserIdFeature, None).exists(!followedUsers.contains(\_))

if (isExtendedReply) TrueFeatureMap else FalseFeatureMap

}

}

}