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

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

import com.twitter.ml.api.DataRecord

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.FeatureWithDefaultOnFailure

import com.twitter.product\_mixer.core.feature.datarecord.DataRecordInAFeature

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

import com.twitter.timelines.prediction.adapters.realtime\_interaction\_graph.RealTimeInteractionGraphFeaturesAdapter

import com.twitter.timelines.prediction.features.realtime\_interaction\_graph.RealTimeInteractionGraphEdgeFeatures

import com.twitter.util.Time

import javax.inject.Inject

import javax.inject.Singleton

import scala.collection.JavaConverters.\_

object RealTimeInteractionGraphEdgeFeature

extends DataRecordInAFeature[TweetCandidate]

with FeatureWithDefaultOnFailure[TweetCandidate, DataRecord] {

override def defaultValue: DataRecord = new DataRecord()

}

@Singleton

class RealTimeInteractionGraphEdgeFeatureHydrator @Inject() ()

extends BulkCandidateFeatureHydrator[PipelineQuery, TweetCandidate] {

override val identifier: FeatureHydratorIdentifier =

FeatureHydratorIdentifier("RealTimeInteractionGraphEdge")

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

private val realTimeInteractionGraphFeaturesAdapter = new RealTimeInteractionGraphFeaturesAdapter

override def apply(

query: PipelineQuery,

candidates: Seq[CandidateWithFeatures[TweetCandidate]]

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

val userVertex =

query.features.flatMap(\_.getOrElse(RealTimeInteractionGraphUserVertexQueryFeature, None))

val realTimeInteractionGraphFeaturesMap =

userVertex.map(RealTimeInteractionGraphEdgeFeatures(\_, Time.now))

candidates.map { candidate =>

val feature = candidate.features.getOrElse(AuthorIdFeature, None).flatMap { authorId =>

realTimeInteractionGraphFeaturesMap.flatMap(\_.get(authorId))

}

val dataRecordFeature =

realTimeInteractionGraphFeaturesAdapter.adaptToDataRecords(feature).asScala.head

FeatureMapBuilder().add(RealTimeInteractionGraphEdgeFeature, dataRecordFeature).build()

}

}

}