package com.twitter.home\_mixer.product.scored\_tweets.candidate\_pipeline

import com.twitter.explore\_ranker.{thriftscala => ert}

import com.twitter.home\_mixer.product.scored\_tweets.feature\_hydrator.TweetypieStaticEntitiesFeatureHydrator

import com.twitter.home\_mixer.product.scored\_tweets.gate.MinCachedTweetsGate

import com.twitter.home\_mixer.product.scored\_tweets.model.ScoredTweetsQuery

import com.twitter.home\_mixer.product.scored\_tweets.param.ScoredTweetsParam.CachedScoredTweets

import com.twitter.home\_mixer.product.scored\_tweets.param.ScoredTweetsParam.CandidatePipeline

import com.twitter.home\_mixer.product.scored\_tweets.response\_transformer.ScoredTweetsPopularVideosResponseFeatureTransformer

import com.twitter.home\_mixer.util.CachedScoredTweetsHelper

import com.twitter.product\_mixer.component\_library.candidate\_source.explore\_ranker.ExploreRankerImmersiveRecsCandidateSource

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

import com.twitter.product\_mixer.core.functional\_component.candidate\_source.BaseCandidateSource

import com.twitter.product\_mixer.core.functional\_component.feature\_hydrator.BaseCandidateFeatureHydrator

import com.twitter.product\_mixer.core.functional\_component.gate.Gate

import com.twitter.product\_mixer.core.functional\_component.marshaller.request.ClientContextMarshaller

import com.twitter.product\_mixer.core.functional\_component.transformer.CandidateFeatureTransformer

import com.twitter.product\_mixer.core.functional\_component.transformer.CandidatePipelineQueryTransformer

import com.twitter.product\_mixer.core.functional\_component.transformer.CandidatePipelineResultsTransformer

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

import com.twitter.product\_mixer.core.pipeline.candidate.CandidatePipelineConfig

import com.twitter.timelines.configapi.decider.DeciderParam

import javax.inject.Inject

import javax.inject.Singleton

@Singleton

class ScoredTweetsPopularVideosCandidatePipelineConfig @Inject() (

exploreRankerCandidateSource: ExploreRankerImmersiveRecsCandidateSource,

tweetypieStaticEntitiesFeatureHydrator: TweetypieStaticEntitiesFeatureHydrator)

extends CandidatePipelineConfig[

ScoredTweetsQuery,

ert.ExploreRankerRequest,

ert.ExploreTweetRecommendation,

TweetCandidate

] {

override val identifier: CandidatePipelineIdentifier =

CandidatePipelineIdentifier("ScoredTweetsPopularVideos")

private val MaxTweetsToFetch = 40

override val enabledDeciderParam: Option[DeciderParam[Boolean]] =

Some(CandidatePipeline.EnablePopularVideosParam)

override val gates: Seq[Gate[ScoredTweetsQuery]] = Seq(

MinCachedTweetsGate(identifier, CachedScoredTweets.MinCachedTweetsParam)

)

override val queryTransformer: CandidatePipelineQueryTransformer[

ScoredTweetsQuery,

ert.ExploreRankerRequest

] = { query =>

val excludedTweetIds = query.features.map(

CachedScoredTweetsHelper.tweetImpressionsAndCachedScoredTweets(\_, identifier))

ert.ExploreRankerRequest(

clientContext = ClientContextMarshaller(query.clientContext),

product = ert.Product.HomeTimelineVideoInline,

productContext = Some(

ert.ProductContext.HomeTimelineVideoInline(ert.HomeTimelineVideoInline(excludedTweetIds))),

maxResults = Some(MaxTweetsToFetch)

)

}

override def candidateSource: BaseCandidateSource[

ert.ExploreRankerRequest,

ert.ExploreTweetRecommendation

] = exploreRankerCandidateSource

override val featuresFromCandidateSourceTransformers: Seq[

CandidateFeatureTransformer[ert.ExploreTweetRecommendation]

] = Seq(ScoredTweetsPopularVideosResponseFeatureTransformer)

override val resultTransformer: CandidatePipelineResultsTransformer[

ert.ExploreTweetRecommendation,

TweetCandidate

] = { sourceResult => TweetCandidate(id = sourceResult.tweetId) }

override val preFilterFeatureHydrationPhase1: Seq[

BaseCandidateFeatureHydrator[ScoredTweetsQuery, TweetCandidate, \_]

] = Seq(tweetypieStaticEntitiesFeatureHydrator)

}