package com.twitter.home\_mixer.product.list\_recommended\_users

import com.twitter.hermit.candidate.{thriftscala => t}

import com.twitter.home\_mixer.product.list\_recommended\_users.candidate\_source.SimilarityBasedUsersCandidateSource

import com.twitter.home\_mixer.product.list\_recommended\_users.feature\_hydrator.IsGizmoduckValidUserFeatureHydrator

import com.twitter.home\_mixer.product.list\_recommended\_users.feature\_hydrator.IsListMemberFeatureHydrator

import com.twitter.home\_mixer.product.list\_recommended\_users.feature\_hydrator.IsSGSValidUserFeatureHydrator

import com.twitter.home\_mixer.product.list\_recommended\_users.feature\_hydrator.RecentListMembersFeature

import com.twitter.home\_mixer.product.list\_recommended\_users.filter.DropMaxCandidatesByAggregatedScoreFilter

import com.twitter.home\_mixer.product.list\_recommended\_users.filter.PreviouslyServedUsersFilter

import com.twitter.home\_mixer.product.list\_recommended\_users.model.ListRecommendedUsersFeatures.IsListMemberFeature

import com.twitter.home\_mixer.product.list\_recommended\_users.model.ListRecommendedUsersQuery

import com.twitter.product\_mixer.component\_library.decorator.urt.UrtItemCandidateDecorator

import com.twitter.product\_mixer.component\_library.decorator.urt.builder.item.user.UserCandidateUrtItemBuilder

import com.twitter.product\_mixer.component\_library.decorator.urt.builder.metadata.ClientEventInfoBuilder

import com.twitter.product\_mixer.component\_library.filter.PredicateFeatureFilter

import com.twitter.product\_mixer.component\_library.gate.NonEmptySeqFeatureGate

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

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

import com.twitter.product\_mixer.core.functional\_component.decorator.CandidateDecorator

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

import com.twitter.product\_mixer.core.functional\_component.filter.Filter

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

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.model.common.identifier.FilterIdentifier

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

import javax.inject.Inject

import javax.inject.Singleton

@Singleton

class ListMemberBasedUsersCandidatePipelineConfig @Inject() (

similarityBasedUsersCandidateSource: SimilarityBasedUsersCandidateSource,

isGizmoduckValidUserFeatureHydrator: IsGizmoduckValidUserFeatureHydrator,

isListMemberFeatureHydrator: IsListMemberFeatureHydrator,

isSGSValidUserFeatureHydrator: IsSGSValidUserFeatureHydrator)

extends CandidatePipelineConfig[

ListRecommendedUsersQuery,

Seq[Long],

t.Candidate,

UserCandidate

] {

override val identifier: CandidatePipelineIdentifier =

CandidatePipelineIdentifier("ListMemberBasedUsers")

override val gates: Seq[Gate[ListRecommendedUsersQuery]] =

Seq(NonEmptySeqFeatureGate(RecentListMembersFeature))

override val queryTransformer: CandidatePipelineQueryTransformer[ListRecommendedUsersQuery, Seq[

Long

]] = { query =>

query.features.map(\_.getOrElse(RecentListMembersFeature, Seq.empty)).getOrElse(Seq.empty)

}

override val candidateSource: BaseCandidateSource[Seq[Long], t.Candidate] =

similarityBasedUsersCandidateSource

override val featuresFromCandidateSourceTransformers: Seq[

CandidateFeatureTransformer[t.Candidate]

] = Seq(ListMemberBasedUsersResponseFeatureTransfromer)

override val resultTransformer: CandidatePipelineResultsTransformer[

t.Candidate,

UserCandidate

] = { candidate =>

UserCandidate(id = candidate.userId)

}

override val preFilterFeatureHydrationPhase1: Seq[

BaseCandidateFeatureHydrator[ListRecommendedUsersQuery, UserCandidate, \_]

] = Seq(isListMemberFeatureHydrator)

override val filters: Seq[Filter[ListRecommendedUsersQuery, UserCandidate]] =

Seq(

PreviouslyServedUsersFilter,

PredicateFeatureFilter.fromPredicate(

FilterIdentifier("IsListMember"),

shouldKeepCandidate = { features => !features.getOrElse(IsListMemberFeature, false) }

),

DropMaxCandidatesByAggregatedScoreFilter

)

override val postFilterFeatureHydration: Seq[

BaseCandidateFeatureHydrator[ListRecommendedUsersQuery, UserCandidate, \_]

] = Seq(

isGizmoduckValidUserFeatureHydrator,

isSGSValidUserFeatureHydrator

)

override val decorator: Option[CandidateDecorator[ListRecommendedUsersQuery, UserCandidate]] = {

val clientEventInfoBuilder = ClientEventInfoBuilder("user")

val userItemBuilder = UserCandidateUrtItemBuilder(clientEventInfoBuilder)

Some(UrtItemCandidateDecorator(userItemBuilder))

}

}