package com.twitter.product\_mixer.component\_library.scorer.cr\_ml\_ranker

import com.twitter.product\_mixer.component\_library.feature\_hydrator.query.cr\_ml\_ranker.CrMlRankerCommonFeatures

import com.twitter.product\_mixer.component\_library.feature\_hydrator.query.cr\_ml\_ranker.CrMlRankerRankingConfig

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.scorer.Scorer

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

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

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

import com.twitter.stitch.Stitch

import javax.inject.Inject

import javax.inject.Singleton

object CrMlRankerScore extends Feature[TweetCandidate, Double]

/\*\*

\* Scorer that scores tweets using the Content Recommender ML Light Ranker: http://go/cr-ml-ranker

\*/

@Singleton

class CrMlRankerScorer @Inject() (crMlRanker: CrMlRankerScoreStitchClient)

extends Scorer[PipelineQuery, TweetCandidate] {

override val identifier: ScorerIdentifier = ScorerIdentifier("CrMlRanker")

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

override def apply(

query: PipelineQuery,

candidates: Seq[CandidateWithFeatures[TweetCandidate]]

): Stitch[Seq[FeatureMap]] = {

val queryFeatureMap = query.features.getOrElse(FeatureMap.empty)

val rankingConfig = queryFeatureMap.get(CrMlRankerRankingConfig)

val commonFeatures = queryFeatureMap.get(CrMlRankerCommonFeatures)

val userId = query.getRequiredUserId

val scoresStitch = Stitch.collect(candidates.map { candidateWithFeatures =>

crMlRanker

.getScore(userId, candidateWithFeatures.candidate, rankingConfig, commonFeatures).map(

\_.score)

})

scoresStitch.map { scores =>

scores.map { score =>

FeatureMapBuilder()

.add(CrMlRankerScore, score)

.build()

}

}

}

}