package com.twitter.follow\_recommendations.common.rankers.utils

import com.twitter.follow\_recommendations.common.models.CandidateUser

import com.twitter.follow\_recommendations.common.models.Score

import com.twitter.follow\_recommendations.common.rankers.common.RankerId.RankerId

object Utils {

/\*\*

\* Add the ranking and scoring info for a list of candidates on a given ranking stage.

\* @param candidates A list of CandidateUser

\* @param rankingStage Should use `Ranker.name` as the ranking stage.

\* @return The list of CandidateUser with ranking/scoring info added.

\*/

def addRankingInfo(candidates: Seq[CandidateUser], rankingStage: String): Seq[CandidateUser] = {

candidates.zipWithIndex.map {

case (candidate, rank) =>

// 1-based ranking for better readability

candidate.addInfoPerRankingStage(rankingStage, candidate.scores, rank + 1)

}

}

def getCandidateScoreByRankerId(candidate: CandidateUser, rankerId: RankerId): Option[Score] =

candidate.scores.flatMap { ss => ss.scores.find(\_.rankerId.contains(rankerId)) }

def getAllRankerIds(candidates: Seq[CandidateUser]): Seq[RankerId] =

candidates.flatMap(\_.scores.map(\_.scores.flatMap(\_.rankerId))).flatten.distinct

}