package com.twitter.follow\_recommendations.common.rankers.ml\_ranker.scoring

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

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

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

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

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

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

import com.twitter.ml.api.DataRecord

import com.twitter.product\_mixer.core.model.marshalling.request.HasClientContext

import com.twitter.stitch.Stitch

import com.twitter.timelines.configapi.HasParams

trait Scorer {

// unique id of the scorer

def id: RankerId.Value

// type of the output scores

def scoreType: Option[ScoreType] = None

// Scoring when an ML model is used.

def score(records: Seq[DataRecord]): Stitch[Seq[Score]]

/\*\*

\* Scoring when a non-ML method is applied. E.g: Boosting, randomized reordering, etc.

\* This method assumes that candidates' scores are already retrieved from heavy-ranker models and

\* are available for use.

\*/

def score(

target: HasClientContext with HasParams with HasDisplayLocation with HasDebugOptions,

candidates: Seq[CandidateUser]

): Seq[Option[Score]]

}