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

import com.twitter.cortex.deepbird.thriftjava.DeepbirdPredictionService

import com.twitter.finagle.stats.StatsReceiver

import com.twitter.follow\_recommendations.common.constants.GuiceNamedConstants

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

import com.twitter.ml.api.DataRecord

import com.twitter.ml.api.Feature

import com.twitter.util.Future

import javax.inject.Inject

import javax.inject.Named

import javax.inject.Singleton

/\*\*

\* This scorer assigns random values between 0 and 1 to each candidate as scores.

\*/

@Singleton

class RandomScorer @Inject() (

@Named(GuiceNamedConstants.WTF\_PROD\_DEEPBIRDV2\_CLIENT)

override val deepbirdClient: DeepbirdPredictionService.ServiceToClient,

override val baseStats: StatsReceiver)

extends DeepbirdScorer {

override val id = RankerId.RandomRanker

private val rnd = new scala.util.Random(System.currentTimeMillis())

override def predict(dataRecords: Seq[DataRecord]): Future[Seq[Option[Double]]] = {

if (dataRecords.isEmpty) {

Future.Nil

} else {

// All candidates are assigned a random value between 0 and 1 as score.

Future.value(dataRecords.map(\_ => Option(rnd.nextDouble())))

}

}

override val modelName = "PostNuxRandomRanker"

// This is not needed since we are overriding the `predict` function, but we have to override

// `predictionFeature` anyway.

override val predictionFeature: Feature.Continuous =

new Feature.Continuous("prediction.pfollow\_pengagement")

}