package com.twitter.follow\_recommendations.common.candidate\_sources.stp

import com.twitter.cortex.deepbird.runtime.prediction\_engine.TensorflowPredictionEngine

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

import com.twitter.ml.api.Feature.Continuous

import com.twitter.ml.api.util.SRichDataRecord

import com.twitter.ml.prediction\_service.PredictionRequest

import com.twitter.stitch.Stitch

import com.twitter.wtf.scalding.jobs.strong\_tie\_prediction.STPRecord

import com.twitter.wtf.scalding.jobs.strong\_tie\_prediction.STPRecordAdapter

import javax.inject.Inject

import javax.inject.Named

import javax.inject.Singleton

/\*\*

\* STP ML ranker trained using DeepBirdV2

\*/

@Singleton

class Dbv2StpScorer @Inject() (

@Named(GuiceNamedConstants.STP\_DBV2\_SCORER) tfPredictionEngine: TensorflowPredictionEngine) {

def getScoredResponse(record: STPRecord): Stitch[Option[Double]] = {

val request: PredictionRequest = new PredictionRequest(

STPRecordAdapter.adaptToDataRecord(record))

val responseStitch = Stitch.callFuture(tfPredictionEngine.getPrediction(request))

responseStitch.map { response =>

val richDr = SRichDataRecord(response.getPrediction)

richDr.getFeatureValueOpt(new Continuous("output"))

}

}

}