package com.twitter.product\_mixer.component\_library.scorer.tensorbuilder

import com.twitter.product\_mixer.component\_library.scorer.common.ModelSelector

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

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

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

import inference.GrpcService.InferParameter

import inference.GrpcService.ModelInferRequest

import scala.collection.JavaConverters.\_

class ModelInferRequestBuilder[-Query <: PipelineQuery, -Candidate <: UniversalNoun[Any]](

queryInferInputTensorBuilders: Seq[QueryInferInputTensorBuilder[Query, Any]],

candidateInferInputTensorBuilders: Seq[

CandidateInferInputTensorBuilder[Candidate, Any]

],

modelSignatureName: String,

modelSelector: ModelSelector[Query]) {

private val modelSignature: InferParameter =

InferParameter.newBuilder().setStringParam(modelSignatureName).build()

def apply(

query: Query,

candidates: Seq[CandidateWithFeatures[Candidate]],

): ModelInferRequest = {

val inferRequest = ModelInferRequest

.newBuilder()

.putParameters("signature\_name", modelSignature)

modelSelector.apply(query).foreach { modelName =>

inferRequest.setModelName(modelName)

}

queryInferInputTensorBuilders.foreach { builder =>

inferRequest.addAllInputs(builder(query).asJava)

}

candidateInferInputTensorBuilders.foreach { builder =>

inferRequest.addAllInputs(builder(candidates).asJava)

}

inferRequest.build()

}

}