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

import com.twitter.ml.api.thriftscala.FloatTensor

import inference.GrpcService.ModelInferRequest.InferInputTensor

case object FloatTensorInferInputTensorBuilder extends InferInputTensorBuilder[FloatTensor] {

private[tensorbuilder] def extractTensorShape(featureValues: Seq[FloatTensor]): Seq[Int] = {

val headFloatTensor = featureValues.head

if (headFloatTensor.shape.isEmpty) {

Seq(

featureValues.size,

featureValues.head.floats.size

)

} else {

Seq(featureValues.size) ++ headFloatTensor.shape.get.map(\_.toInt)

}

}

def apply(

featureName: String,

featureValues: Seq[FloatTensor]

): Seq[InferInputTensor] = {

if (featureValues.isEmpty) throw new EmptyFloatTensorException(featureName)

val tensorShape = extractTensorShape(featureValues)

val floatValues = featureValues.flatMap { featureValue =>

featureValue.floats.map(\_.toFloat)

}

InferInputTensorBuilder.buildFloat32InferInputTensor(featureName, floatValues, tensorShape)

}

}

class EmptyFloatTensorException(featureName: String)

extends RuntimeException(s"FloatTensor in feature $featureName is empty!")