package com.twitter.simclusters\_v2.scalding.evaluation

import com.twitter.algebird.AveragedValue

import com.twitter.scalding.Execution

import com.twitter.scalding.typed.TypedPipe

import com.twitter.simclusters\_v2.scalding.common.Util

/\*\*

\* Utility object for correlation measures between the algorithm scores and the user engagements,

\* such as the number of Likes.

\*/

object LabelCorrelationsHelper {

private def toDouble(bool: Boolean): Double = {

if (bool) 1.0 else 0.0

}

/\*\*

\* Given a pipe of labeled tweets, calculate the cosine similarity between the algorithm scores

\* and users' favorite engagements.

\*/

def cosineSimilarityForLike(labeledTweets: TypedPipe[LabeledTweet]): Execution[Double] = {

labeledTweets

.map { tweet => (toDouble(tweet.labels.isLiked), tweet.algorithmScore.getOrElse(0.0)) }

.toIterableExecution.map { iter => Util.cosineSimilarity(iter.iterator) }

}

/\*\*

\* Given a pipe of labeled tweets, calculate cosine similarity between algorithm score and users'

\* favorites engagements, on a per user basis, and return the average of all cosine

\* similarities across all users.

\*/

def cosineSimilarityForLikePerUser(labeledTweets: TypedPipe[LabeledTweet]): Execution[Double] = {

val avg = AveragedValue.aggregator.composePrepare[(Unit, Double)](\_.\_2)

labeledTweets

.map { tweet =>

(

tweet.targetUserId,

Seq((toDouble(tweet.labels.isLiked), tweet.algorithmScore.getOrElse(0.0)))

)

}

.sumByKey

.map {

case (userId, seq) =>

((), Util.cosineSimilarity(seq.iterator))

}

.aggregate(avg)

.getOrElseExecution(0.0)

}

/\*\*

\* Calculates the Pearson correlation coefficient for the algorithm scores and user's favorite

\* engagement. Note this function call triggers a writeToDisk execution.

\*/

def pearsonCoefficientForLike(labeledTweets: TypedPipe[LabeledTweet]): Execution[Double] = {

labeledTweets

.map { tweet => (toDouble(tweet.labels.isLiked), tweet.algorithmScore.getOrElse(0.0)) }

.toIterableExecution.map { iter => Util.computeCorrelation(iter.iterator) }

}

}