package com.twitter.graph.batch.job.tweepcred

import com.twitter.twadoop.user.gen.CombinedUser

import com.twitter.util.Time

import com.twitter.wtf.scalding.jobs.common.DateUtil

case class UserMassInfo(userId: Long, mass: Double)

/\*\*

\* helper class to calculate user mass, borrowed from repo reputations

\*/

object UserMass {

private val currentTimestamp = Time.now.inMilliseconds

private val constantDivisionFactorGt\_threshFriendsToFollowersRatioUMass = 5.0

private val threshAbsNumFriendsUMass = 500

private val threshFriendsToFollowersRatioUMass = 0.6

private val deviceWeightAdditive = 0.5

private val ageWeightAdditive = 0.2

private val restrictedWeightMultiplicative = 0.1

def getUserMass(combinedUser: CombinedUser): Option[UserMassInfo] = {

val user = Option(combinedUser.user)

val userId = user.map(\_.id).getOrElse(0L)

val userExtended = Option(combinedUser.user\_extended)

val age = user.map(\_.created\_at\_msec).map(DateUtil.diffDays(\_, currentTimestamp)).getOrElse(0)

val isRestricted = user.map(\_.safety).exists(\_.restricted)

val isSuspended = user.map(\_.safety).exists(\_.suspended)

val isVerified = user.map(\_.safety).exists(\_.verified)

val hasValidDevice = user.flatMap(u => Option(u.devices)).exists(\_.isSetMessaging\_devices)

val numFollowers = userExtended.flatMap(u => Option(u.followers)).map(\_.toInt).getOrElse(0)

val numFollowings = userExtended.flatMap(u => Option(u.followings)).map(\_.toInt).getOrElse(0)

if (userId == 0L || user.map(\_.safety).exists(\_.deactivated)) {

None

} else {

val mass =

if (isSuspended)

0

else if (isVerified)

100

else {

var score = deviceWeightAdditive \* 0.1 +

(if (hasValidDevice) deviceWeightAdditive else 0)

val normalizedAge = if (age > 30) 1.0 else (1.0 min scala.math.log(1.0 + age / 15.0))

score \*= normalizedAge

if (score < 0.01) score = 0.01

if (isRestricted) score \*= restrictedWeightMultiplicative

score = (score min 1.0) max 0

score \*= 100

score

}

val friendsToFollowersRatio = (1.0 + numFollowings) / (1.0 + numFollowers)

val adjustedMass =

if (numFollowings > threshAbsNumFriendsUMass &&

friendsToFollowersRatio > threshFriendsToFollowersRatioUMass) {

mass / scala.math.exp(

constantDivisionFactorGt\_threshFriendsToFollowersRatioUMass \*

(friendsToFollowersRatio - threshFriendsToFollowersRatioUMass)

)

} else {

mass

}

Some(UserMassInfo(userId, adjustedMass))

}

}

}