package com.twitter.visibility.builder.tweets

import com.twitter.finagle.stats.StatsReceiver

import com.twitter.stitch.Stitch

import com.twitter.tweetypie.thriftscala.Tweet

import com.twitter.visibility.builder.FeatureMapBuilder

import com.twitter.visibility.builder.users.ViewerVerbsAuthor

import com.twitter.visibility.common.UserRelationshipSource

import com.twitter.visibility.features.TweetIsExclusiveTweet

import com.twitter.visibility.features.ViewerIsExclusiveTweetRootAuthor

import com.twitter.visibility.features.ViewerSuperFollowsExclusiveTweetRootAuthor

import com.twitter.visibility.models.ViewerContext

class ExclusiveTweetFeatures(

userRelationshipSource: UserRelationshipSource,

statsReceiver: StatsReceiver) {

private[this] val scopedStatsReceiver = statsReceiver.scope("exclusive\_tweet\_features")

private[this] val viewerSuperFollowsAuthor =

scopedStatsReceiver.scope(ViewerSuperFollowsExclusiveTweetRootAuthor.name).counter("requests")

def rootAuthorId(tweet: Tweet): Option[Long] =

tweet.exclusiveTweetControl.map(\_.conversationAuthorId)

def viewerIsRootAuthor(

tweet: Tweet,

viewerIdOpt: Option[Long]

): Boolean =

(rootAuthorId(tweet), viewerIdOpt) match {

case (Some(rootAuthorId), Some(viewerId)) if rootAuthorId == viewerId => true

case \_ => false

}

def viewerSuperFollowsRootAuthor(

tweet: Tweet,

viewerId: Option[Long]

): Stitch[Boolean] =

rootAuthorId(tweet) match {

case Some(authorId) =>

ViewerVerbsAuthor(

authorId,

viewerId,

userRelationshipSource.superFollows,

viewerSuperFollowsAuthor)

case None =>

Stitch.False

}

def forTweet(

tweet: Tweet,

viewerContext: ViewerContext

): FeatureMapBuilder => FeatureMapBuilder = {

val viewerId = viewerContext.userId

\_.withConstantFeature(TweetIsExclusiveTweet, tweet.exclusiveTweetControl.isDefined)

.withConstantFeature(ViewerIsExclusiveTweetRootAuthor, viewerIsRootAuthor(tweet, viewerId))

.withFeature(

ViewerSuperFollowsExclusiveTweetRootAuthor,

viewerSuperFollowsRootAuthor(tweet, viewerId))

}

def forTweetOnly(tweet: Tweet): FeatureMapBuilder => FeatureMapBuilder = {

\_.withConstantFeature(TweetIsExclusiveTweet, tweet.exclusiveTweetControl.isDefined)

}

}