package com.twitter.frigate.pushservice.model

import com.twitter.events.recos.thriftscala.TrendsContext

import com.twitter.finagle.stats.StatsReceiver

import com.twitter.frigate.common.base.TrendTweetCandidate

import com.twitter.frigate.common.base.TweetAuthorDetails

import com.twitter.frigate.pushservice.model.PushTypes.PushCandidate

import com.twitter.frigate.pushservice.model.PushTypes.RawCandidate

import com.twitter.frigate.pushservice.config.Config

import com.twitter.frigate.pushservice.ml.PushMLModelScorer

import com.twitter.frigate.pushservice.model.candidate.CopyIds

import com.twitter.frigate.pushservice.model.ibis.TrendTweetIbis2Hydrator

import com.twitter.frigate.pushservice.model.ntab.TrendTweetNtabHydrator

import com.twitter.frigate.pushservice.take.predicates.BasicTweetPredicatesForRFPH

import com.twitter.gizmoduck.thriftscala.User

import com.twitter.stitch.tweetypie.TweetyPie

import com.twitter.util.Future

class TrendTweetPushCandidate(

candidate: RawCandidate with TrendTweetCandidate,

author: Option[User],

copyIds: CopyIds

)(

implicit stats: StatsReceiver,

pushModelScorer: PushMLModelScorer)

extends PushCandidate

with TrendTweetCandidate

with TweetAuthorDetails

with TrendTweetIbis2Hydrator

with TrendTweetNtabHydrator {

override val statsReceiver: StatsReceiver = stats

override val weightedOpenOrNtabClickModelScorer: PushMLModelScorer = pushModelScorer

override val tweetId: Long = candidate.tweetId

override lazy val tweetyPieResult: Option[TweetyPie.TweetyPieResult] = candidate.tweetyPieResult

override lazy val tweetAuthor: Future[Option[User]] = Future.value(author)

override val target: PushTypes.Target = candidate.target

override val landingUrl: String = candidate.landingUrl

override val timeBoundedLandingUrl: Option[String] = candidate.timeBoundedLandingUrl

override val pushCopyId: Option[Int] = copyIds.pushCopyId

override val ntabCopyId: Option[Int] = copyIds.ntabCopyId

override val trendId: String = candidate.trendId

override val trendName: String = candidate.trendName

override val copyAggregationId: Option[String] = copyIds.aggregationId

override val context: TrendsContext = candidate.context

}

case class TrendTweetPredicates(override val config: Config)

extends BasicTweetPredicatesForRFPH[TrendTweetPushCandidate] {

implicit val statsReceiver: StatsReceiver = config.statsReceiver.scope(getClass.getSimpleName)

}