package com.twitter.frigate.pushservice.model

import com.twitter.finagle.stats.StatsReceiver

import com.twitter.frigate.common.base.TopTweetImpressionsCandidate

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.TopTweetImpressionsCandidateIbis2Hydrator

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

import com.twitter.frigate.pushservice.predicate.TopTweetImpressionsPredicates

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

import com.twitter.frigate.thriftscala.CommonRecommendationType

import com.twitter.hermit.predicate.NamedPredicate

import com.twitter.notificationservice.thriftscala.StoryContext

import com.twitter.notificationservice.thriftscala.StoryContextValue

import com.twitter.stitch.tweetypie.TweetyPie

/\*\*

\* This class defines a hydrated [[TopTweetImpressionsCandidate]]

\*

\* @param candidate: [[TopTweetImpressionsCandidate]] for the candidate representing the user's Tweet with the most impressions

\* @param copyIds: push and ntab notification copy

\* @param stats: finagle scoped states receiver

\* @param pushModelScorer: ML model score object for fetching prediction scores

\*/

class TopTweetImpressionsPushCandidate(

candidate: RawCandidate with TopTweetImpressionsCandidate,

copyIds: CopyIds

)(

implicit stats: StatsReceiver,

pushModelScorer: PushMLModelScorer)

extends PushCandidate

with TopTweetImpressionsCandidate

with TopTweetImpressionsNTabRequestHydrator

with TopTweetImpressionsCandidateIbis2Hydrator {

override val target: PushTypes.Target = candidate.target

override val commonRecType: CommonRecommendationType = candidate.commonRecType

override val tweetId: Long = candidate.tweetId

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

candidate.tweetyPieResult

override val impressionsCount: Long = candidate.impressionsCount

override val statsReceiver: StatsReceiver = stats.scope(getClass.getSimpleName)

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

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

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

override val weightedOpenOrNtabClickModelScorer: PushMLModelScorer = pushModelScorer

override val storyContext: Option[StoryContext] =

Some(StoryContext(altText = "", value = Some(StoryContextValue.Tweets(Seq(tweetId)))))

}

case class TopTweetImpressionsPushCandidatePredicates(config: Config)

extends BasicTweetPredicatesForRFPH[TopTweetImpressionsPushCandidate] {

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

override val preCandidateSpecificPredicates: List[

NamedPredicate[TopTweetImpressionsPushCandidate]

] = List(

TopTweetImpressionsPredicates.topTweetImpressionsFatiguePredicate

)

override val postCandidateSpecificPredicates: List[

NamedPredicate[TopTweetImpressionsPushCandidate]

] = List(

TopTweetImpressionsPredicates.topTweetImpressionsThreshold()

)

}