package com.twitter.cr\_mixer.filter

import com.twitter.cr\_mixer.model.CandidateGeneratorQuery

import com.twitter.cr\_mixer.model.InitialCandidate

import com.twitter.cr\_mixer.param.UtegTweetGlobalParams

import com.twitter.util.Future

import javax.inject.Inject

import javax.inject.Singleton

/\*\*

\* Remove unhealthy candidates

\* Currently Timeline Ranker applies a check on the following three scores:

\* - toxicityScore

\* - pBlockScore

\* - pReportedTweetScore

\*

\* Where isPassTweetHealthFilterStrict checks two additions scores with the same threshold:

\* - pSpammyTweetScore

\* - spammyTweetContentScore

\*

\* We've verified that both filters behave very similarly.

\*/

@Singleton

case class UtegHealthFilter @Inject() () extends FilterBase {

override def name: String = this.getClass.getCanonicalName

override type ConfigType = Boolean

override def filter(

candidates: Seq[Seq[InitialCandidate]],

config: ConfigType

): Future[Seq[Seq[InitialCandidate]]] = {

if (config) {

Future.value(

candidates.map { candidateSeq =>

candidateSeq.filter { candidate =>

candidate.tweetInfo.isPassTweetHealthFilterStrict.getOrElse(false)

}

}

)

} else {

Future.value(candidates)

}

}

override def requestToConfig[CGQueryType <: CandidateGeneratorQuery](

query: CGQueryType

): ConfigType = {

query.params(UtegTweetGlobalParams.EnableTLRHealthFilterParam)

}

}