package com.twitter.frigate.pushservice.predicate.quality\_model\_predicate

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

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

import com.twitter.frigate.pushservice.params.PushFeatureSwitchParams

import com.twitter.util.Future

object ExplicitOONCFilterPredicate extends QualityPredicateBase {

override lazy val name = "open\_or\_ntab\_click\_explicit\_threshold"

override lazy val thresholdExtractor = (t: Target) =>

Future.value(t.params(PushFeatureSwitchParams.QualityPredicateExplicitThresholdParam))

override def scoreExtractor = (candidate: PushCandidate) =>

candidate.mrWeightedOpenOrNtabClickRankingProbability

}

object WeightedOpenOrNtabClickQualityPredicate extends QualityPredicateBase {

override lazy val name = "weighted\_open\_or\_ntab\_click\_model"

override lazy val thresholdExtractor = (t: Target) => {

Future.value(0.0)

}

override def scoreExtractor =

(candidate: PushCandidate) => candidate.mrWeightedOpenOrNtabClickFilteringProbability

}