package com.twitter.usersignalservice.signals

import com.twitter.twistly.thriftscala.EngagementMetadata.FavoriteMetadata

import com.twitter.twistly.thriftscala.RecentEngagedTweet

import com.twitter.usersignalservice.thriftscala.SignalType

import com.twitter.util.Time

// Shared Logic for filtering signal across different signal types

object SignalFilter {

final val LookBackWindow90DayFilterEnabledSignalTypes: Set[SignalType] = Set(

SignalType.TweetFavorite90dV2,

SignalType.Retweet90dV2,

SignalType.OriginalTweet90dV2,

SignalType.Reply90dV2)

/\* Raw Signal Filter for TweetFavorite, Retweet, Original Tweet and Reply

\* Filter out all raw signal if the most recent {Tweet Favorite + Retweet + Original Tweet + Reply}

\* is older than 90 days.

\* The filter is shared across 4 signal types as they are stored in the same physical store

\* thus sharing the same TTL

\* \*/

def lookBackWindow90DayFilter(

signals: Seq[RecentEngagedTweet],

querySignalType: SignalType

): Seq[RecentEngagedTweet] = {

if (LookBackWindow90DayFilterEnabledSignalTypes.contains(

querySignalType) && !isMostRecentSignalWithin90Days(signals.head)) {

Seq.empty

} else signals

}

private def isMostRecentSignalWithin90Days(

signal: RecentEngagedTweet

): Boolean = {

val diff = Time.now - Time.fromMilliseconds(signal.engagedAt)

diff.inDays <= 90

}

def isPromotedTweet(signal: RecentEngagedTweet): Boolean = {

signal match {

case RecentEngagedTweet(\_, \_, metadata: FavoriteMetadata, \_) =>

metadata.favoriteMetadata.isAd.getOrElse(false)

case \_ => false

}

}

}