package com.twitter.timelines.prediction.features.time\_features

import com.twitter.dal.personal\_data.thriftjava.PersonalDataType.\_

import com.twitter.ml.api.Feature.\_

import scala.collection.JavaConverters.\_

import com.twitter.util.Duration

import com.twitter.conversions.DurationOps.\_

object TimeDataRecordFeatures {

val TIME\_BETWEEN\_NON\_POLLING\_REQUESTS\_AVG = new Continuous(

"time\_features.time\_between\_non\_polling\_requests\_avg",

Set(PrivateTimestamp).asJava

)

val TIME\_SINCE\_TWEET\_CREATION = new Continuous("time\_features.time\_since\_tweet\_creation")

val TIME\_SINCE\_SOURCE\_TWEET\_CREATION = new Continuous(

"time\_features.time\_since\_source\_tweet\_creation"

)

val TIME\_SINCE\_LAST\_NON\_POLLING\_REQUEST = new Continuous(

"time\_features.time\_since\_last\_non\_polling\_request",

Set(PrivateTimestamp).asJava

)

val NON\_POLLING\_REQUESTS\_SINCE\_TWEET\_CREATION = new Continuous(

"time\_features.non\_polling\_requests\_since\_tweet\_creation",

Set(PrivateTimestamp).asJava

)

val TWEET\_AGE\_RATIO = new Continuous("time\_features.tweet\_age\_ratio")

val IS\_TWEET\_RECYCLED = new Binary("time\_features.is\_tweet\_recycled")

// Last Engagement features

val LAST\_FAVORITE\_SINCE\_CREATION\_HRS = new Continuous(

"time\_features.earlybird.last\_favorite\_since\_creation\_hrs",

Set(CountOfPrivateLikes, CountOfPublicLikes).asJava

)

val LAST\_RETWEET\_SINCE\_CREATION\_HRS = new Continuous(

"time\_features.earlybird.last\_retweet\_since\_creation\_hrs",

Set(CountOfPrivateRetweets, CountOfPublicRetweets).asJava

)

val LAST\_REPLY\_SINCE\_CREATION\_HRS = new Continuous(

"time\_features.earlybird.last\_reply\_since\_creation\_hrs",

Set(CountOfPrivateReplies, CountOfPublicReplies).asJava

)

val LAST\_QUOTE\_SINCE\_CREATION\_HRS = new Continuous(

"time\_features.earlybird.last\_quote\_since\_creation\_hrs",

Set(CountOfPrivateRetweets, CountOfPublicRetweets).asJava

)

val TIME\_SINCE\_LAST\_FAVORITE\_HRS = new Continuous(

"time\_features.earlybird.time\_since\_last\_favorite",

Set(CountOfPrivateLikes, CountOfPublicLikes).asJava

)

val TIME\_SINCE\_LAST\_RETWEET\_HRS = new Continuous(

"time\_features.earlybird.time\_since\_last\_retweet",

Set(CountOfPrivateRetweets, CountOfPublicRetweets).asJava

)

val TIME\_SINCE\_LAST\_REPLY\_HRS = new Continuous(

"time\_features.earlybird.time\_since\_last\_reply",

Set(CountOfPrivateReplies, CountOfPublicReplies).asJava

)

val TIME\_SINCE\_LAST\_QUOTE\_HRS = new Continuous(

"time\_features.earlybird.time\_since\_last\_quote",

Set(CountOfPrivateRetweets, CountOfPublicRetweets).asJava

)

val TIME\_SINCE\_VIEWER\_ACCOUNT\_CREATION\_SECS =

new Continuous(

"time\_features.time\_since\_viewer\_account\_creation\_secs",

Set(AccountCreationTime, AgeOfAccount).asJava)

val USER\_ID\_IS\_SNOWFLAKE\_ID =

new Binary("time\_features.time\_user\_id\_is\_snowflake\_id", Set(UserType).asJava)

val IS\_30\_DAY\_NEW\_USER =

new Binary("time\_features.is\_day\_30\_new\_user", Set(AccountCreationTime, AgeOfAccount).asJava)

val IS\_12\_MONTH\_NEW\_USER =

new Binary("time\_features.is\_month\_12\_new\_user", Set(AccountCreationTime, AgeOfAccount).asJava)

val ACCOUNT\_AGE\_INTERVAL =

new Discrete("time\_features.account\_age\_interval", Set(AgeOfAccount).asJava)

}

object AccountAgeInterval extends Enumeration {

val LTE\_1\_DAY, GT\_1\_DAY\_LTE\_5\_DAY, GT\_5\_DAY\_LTE\_14\_DAY, GT\_14\_DAY\_LTE\_30\_DAY = Value

def fromDuration(accountAge: Duration): Option[AccountAgeInterval.Value] = {

accountAge match {

case a if (a <= 1.day) => Some(LTE\_1\_DAY)

case a if (1.day < a && a <= 5.days) => Some(GT\_1\_DAY\_LTE\_5\_DAY)

case a if (5.days < a && a <= 14.days) => Some(GT\_5\_DAY\_LTE\_14\_DAY)

case a if (14.days < a && a <= 30.days) => Some(GT\_14\_DAY\_LTE\_30\_DAY)

case \_ => None

}

}

}

case class TimeFeatures(

isTweetRecycled: Boolean,

timeSinceTweetCreation: Double,

isDay30NewUser: Boolean,

isMonth12NewUser: Boolean,

timeSinceSourceTweetCreation: Double, // same as timeSinceTweetCreation for non-retweets

timeSinceViewerAccountCreationSecs: Option[Double],

timeBetweenNonPollingRequestsAvg: Option[Double] = None,

timeSinceLastNonPollingRequest: Option[Double] = None,

nonPollingRequestsSinceTweetCreation: Option[Double] = None,

tweetAgeRatio: Option[Double] = None,

lastFavSinceCreationHrs: Option[Double] = None,

lastRetweetSinceCreationHrs: Option[Double] = None,

lastReplySinceCreationHrs: Option[Double] = None,

lastQuoteSinceCreationHrs: Option[Double] = None,

timeSinceLastFavoriteHrs: Option[Double] = None,

timeSinceLastRetweetHrs: Option[Double] = None,

timeSinceLastReplyHrs: Option[Double] = None,

timeSinceLastQuoteHrs: Option[Double] = None,

accountAgeInterval: Option[AccountAgeInterval.Value] = None)