package com.twitter.timelines.prediction.features.user\_health

import com.twitter.ml.api.Feature

import com.twitter.timelines.author\_features.user\_health.thriftscala.UserState

import com.twitter.dal.personal\_data.thriftjava.PersonalDataType.{UserState => UserStatePDT}

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

import scala.collection.JavaConverters.\_

object UserHealthFeatures {

val UserState = new Feature.Discrete("user\_health.user\_state", Set(UserStatePDT, UserType).asJava)

val IsLightMinusUser =

new Feature.Binary("user\_health.is\_light\_minus\_user", Set(UserStatePDT, UserType).asJava)

val AuthorState =

new Feature.Discrete("user\_health.author\_state", Set(UserStatePDT, UserType).asJava)

val NumAuthorFollowers =

new Feature.Continuous("author\_health.num\_followers", Set(CountOfFollowersAndFollowees).asJava)

val NumAuthorConnectDays = new Feature.Continuous("author\_health.num\_connect\_days")

val NumAuthorConnect = new Feature.Continuous("author\_health.num\_connect")

val IsUserVerifiedUnion = new Feature.Binary("user\_account.is\_user\_verified\_union")

}

case class UserHealthFeatures(id: Long, userStateOpt: Option[UserState])