package com.twitter.timelines.prediction.features.real\_graph

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

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

import com.twitter.timelines.real\_graph.v1.thriftscala.RealGraphEdgeFeature

import scala.collection.JavaConverters.\_

object RealGraphDataRecordFeatures {

// the source user id

val SRC\_ID = new Discrete("realgraph.src\_id", Set(UserId).asJava)

// the destination user id

val DST\_ID = new Discrete("realgraph.dst\_id", Set(UserId).asJava)

// real graph weight

val WEIGHT = new Continuous("realgraph.weight", Set(UsersRealGraphScore).asJava)

// the number of retweets that the source user sent to the destination user

val NUM\_RETWEETS\_MEAN =

new Continuous("realgraph.num\_retweets.mean", Set(PrivateRetweets, PublicRetweets).asJava)

val NUM\_RETWEETS\_EWMA =

new Continuous("realgraph.num\_retweets.ewma", Set(PrivateRetweets, PublicRetweets).asJava)

val NUM\_RETWEETS\_VARIANCE =

new Continuous("realgraph.num\_retweets.variance", Set(PrivateRetweets, PublicRetweets).asJava)

val NUM\_RETWEETS\_NON\_ZERO\_DAYS = new Continuous(

"realgraph.num\_retweets.non\_zero\_days",

Set(PrivateRetweets, PublicRetweets).asJava)

val NUM\_RETWEETS\_ELAPSED\_DAYS = new Continuous(

"realgraph.num\_retweets.elapsed\_days",

Set(PrivateRetweets, PublicRetweets).asJava)

val NUM\_RETWEETS\_DAYS\_SINCE\_LAST = new Continuous(

"realgraph.num\_retweets.days\_since\_last",

Set(PrivateRetweets, PublicRetweets).asJava)

val NUM\_RETWEETS\_IS\_MISSING =

new Binary("realgraph.num\_retweets.is\_missing", Set(PrivateRetweets, PublicRetweets).asJava)

// the number of favories that the source user sent to the destination user

val NUM\_FAVORITES\_MEAN =

new Continuous("realgraph.num\_favorites.mean", Set(PublicLikes, PrivateLikes).asJava)

val NUM\_FAVORITES\_EWMA =

new Continuous("realgraph.num\_favorites.ewma", Set(PublicLikes, PrivateLikes).asJava)

val NUM\_FAVORITES\_VARIANCE =

new Continuous("realgraph.num\_favorites.variance", Set(PublicLikes, PrivateLikes).asJava)

val NUM\_FAVORITES\_NON\_ZERO\_DAYS =

new Continuous("realgraph.num\_favorites.non\_zero\_days", Set(PublicLikes, PrivateLikes).asJava)

val NUM\_FAVORITES\_ELAPSED\_DAYS =

new Continuous("realgraph.num\_favorites.elapsed\_days", Set(PublicLikes, PrivateLikes).asJava)

val NUM\_FAVORITES\_DAYS\_SINCE\_LAST =

new Continuous("realgraph.num\_favorites.days\_since\_last", Set(PublicLikes, PrivateLikes).asJava)

val NUM\_FAVORITES\_IS\_MISSING =

new Binary("realgraph.num\_favorites.is\_missing", Set(PublicLikes, PrivateLikes).asJava)

// the number of mentions that the source user sent to the destination user

val NUM\_MENTIONS\_MEAN =

new Continuous("realgraph.num\_mentions.mean", Set(EngagementsPrivate, EngagementsPublic).asJava)

val NUM\_MENTIONS\_EWMA =

new Continuous("realgraph.num\_mentions.ewma", Set(EngagementsPrivate, EngagementsPublic).asJava)

val NUM\_MENTIONS\_VARIANCE = new Continuous(

"realgraph.num\_mentions.variance",

Set(EngagementsPrivate, EngagementsPublic).asJava)

val NUM\_MENTIONS\_NON\_ZERO\_DAYS = new Continuous(

"realgraph.num\_mentions.non\_zero\_days",

Set(EngagementsPrivate, EngagementsPublic).asJava)

val NUM\_MENTIONS\_ELAPSED\_DAYS = new Continuous(

"realgraph.num\_mentions.elapsed\_days",

Set(EngagementsPrivate, EngagementsPublic).asJava)

val NUM\_MENTIONS\_DAYS\_SINCE\_LAST = new Continuous(

"realgraph.num\_mentions.days\_since\_last",

Set(EngagementsPrivate, EngagementsPublic).asJava)

val NUM\_MENTIONS\_IS\_MISSING = new Binary(

"realgraph.num\_mentions.is\_missing",

Set(EngagementsPrivate, EngagementsPublic).asJava)

// the number of direct messages that the source user sent to the destination user

val NUM\_DIRECT\_MESSAGES\_MEAN = new Continuous(

"realgraph.num\_direct\_messages.mean",

Set(DmEntitiesAndMetadata, CountOfDms).asJava)

val NUM\_DIRECT\_MESSAGES\_EWMA = new Continuous(

"realgraph.num\_direct\_messages.ewma",

Set(DmEntitiesAndMetadata, CountOfDms).asJava)

val NUM\_DIRECT\_MESSAGES\_VARIANCE = new Continuous(

"realgraph.num\_direct\_messages.variance",

Set(DmEntitiesAndMetadata, CountOfDms).asJava)

val NUM\_DIRECT\_MESSAGES\_NON\_ZERO\_DAYS = new Continuous(

"realgraph.num\_direct\_messages.non\_zero\_days",

Set(DmEntitiesAndMetadata, CountOfDms).asJava

)

val NUM\_DIRECT\_MESSAGES\_ELAPSED\_DAYS = new Continuous(

"realgraph.num\_direct\_messages.elapsed\_days",

Set(DmEntitiesAndMetadata, CountOfDms).asJava

)

val NUM\_DIRECT\_MESSAGES\_DAYS\_SINCE\_LAST = new Continuous(

"realgraph.num\_direct\_messages.days\_since\_last",

Set(DmEntitiesAndMetadata, CountOfDms).asJava

)

val NUM\_DIRECT\_MESSAGES\_IS\_MISSING = new Binary(

"realgraph.num\_direct\_messages.is\_missing",

Set(DmEntitiesAndMetadata, CountOfDms).asJava)

// the number of tweet clicks that the source user sent to the destination user

val NUM\_TWEET\_CLICKS\_MEAN =

new Continuous("realgraph.num\_tweet\_clicks.mean", Set(TweetsClicked).asJava)

val NUM\_TWEET\_CLICKS\_EWMA =

new Continuous("realgraph.num\_tweet\_clicks.ewma", Set(TweetsClicked).asJava)

val NUM\_TWEET\_CLICKS\_VARIANCE =

new Continuous("realgraph.num\_tweet\_clicks.variance", Set(TweetsClicked).asJava)

val NUM\_TWEET\_CLICKS\_NON\_ZERO\_DAYS =

new Continuous("realgraph.num\_tweet\_clicks.non\_zero\_days", Set(TweetsClicked).asJava)

val NUM\_TWEET\_CLICKS\_ELAPSED\_DAYS =

new Continuous("realgraph.num\_tweet\_clicks.elapsed\_days", Set(TweetsClicked).asJava)

val NUM\_TWEET\_CLICKS\_DAYS\_SINCE\_LAST = new Continuous(

"realgraph.num\_tweet\_clicks.days\_since\_last",

Set(TweetsClicked).asJava

)

val NUM\_TWEET\_CLICKS\_IS\_MISSING =

new Binary("realgraph.num\_tweet\_clicks.is\_missing", Set(TweetsClicked).asJava)

// the number of link clicks that the source user sent to the destination user

val NUM\_LINK\_CLICKS\_MEAN =

new Continuous("realgraph.num\_link\_clicks.mean", Set(CountOfTweetEntitiesClicked).asJava)

val NUM\_LINK\_CLICKS\_EWMA =

new Continuous("realgraph.num\_link\_clicks.ewma", Set(CountOfTweetEntitiesClicked).asJava)

val NUM\_LINK\_CLICKS\_VARIANCE =

new Continuous("realgraph.num\_link\_clicks.variance", Set(CountOfTweetEntitiesClicked).asJava)

val NUM\_LINK\_CLICKS\_NON\_ZERO\_DAYS = new Continuous(

"realgraph.num\_link\_clicks.non\_zero\_days",

Set(CountOfTweetEntitiesClicked).asJava)

val NUM\_LINK\_CLICKS\_ELAPSED\_DAYS = new Continuous(

"realgraph.num\_link\_clicks.elapsed\_days",

Set(CountOfTweetEntitiesClicked).asJava)

val NUM\_LINK\_CLICKS\_DAYS\_SINCE\_LAST = new Continuous(

"realgraph.num\_link\_clicks.days\_since\_last",

Set(CountOfTweetEntitiesClicked).asJava)

val NUM\_LINK\_CLICKS\_IS\_MISSING =

new Binary("realgraph.num\_link\_clicks.is\_missing", Set(CountOfTweetEntitiesClicked).asJava)

// the number of profile views that the source user sent to the destination user

val NUM\_PROFILE\_VIEWS\_MEAN =

new Continuous("realgraph.num\_profile\_views.mean", Set(ProfilesViewed).asJava)

val NUM\_PROFILE\_VIEWS\_EWMA =

new Continuous("realgraph.num\_profile\_views.ewma", Set(ProfilesViewed).asJava)

val NUM\_PROFILE\_VIEWS\_VARIANCE =

new Continuous("realgraph.num\_profile\_views.variance", Set(ProfilesViewed).asJava)

val NUM\_PROFILE\_VIEWS\_NON\_ZERO\_DAYS =

new Continuous("realgraph.num\_profile\_views.non\_zero\_days", Set(ProfilesViewed).asJava)

val NUM\_PROFILE\_VIEWS\_ELAPSED\_DAYS =

new Continuous("realgraph.num\_profile\_views.elapsed\_days", Set(ProfilesViewed).asJava)

val NUM\_PROFILE\_VIEWS\_DAYS\_SINCE\_LAST = new Continuous(

"realgraph.num\_profile\_views.days\_since\_last",

Set(ProfilesViewed).asJava

)

val NUM\_PROFILE\_VIEWS\_IS\_MISSING =

new Binary("realgraph.num\_profile\_views.is\_missing", Set(ProfilesViewed).asJava)

// the total dwell time the source user spends on the target user's tweets

val TOTAL\_DWELL\_TIME\_MEAN =

new Continuous("realgraph.total\_dwell\_time.mean", Set(CountOfImpression).asJava)

val TOTAL\_DWELL\_TIME\_EWMA =

new Continuous("realgraph.total\_dwell\_time.ewma", Set(CountOfImpression).asJava)

val TOTAL\_DWELL\_TIME\_VARIANCE =

new Continuous("realgraph.total\_dwell\_time.variance", Set(CountOfImpression).asJava)

val TOTAL\_DWELL\_TIME\_NON\_ZERO\_DAYS =

new Continuous("realgraph.total\_dwell\_time.non\_zero\_days", Set(CountOfImpression).asJava)

val TOTAL\_DWELL\_TIME\_ELAPSED\_DAYS =

new Continuous("realgraph.total\_dwell\_time.elapsed\_days", Set(CountOfImpression).asJava)

val TOTAL\_DWELL\_TIME\_DAYS\_SINCE\_LAST = new Continuous(

"realgraph.total\_dwell\_time.days\_since\_last",

Set(CountOfImpression).asJava

)

val TOTAL\_DWELL\_TIME\_IS\_MISSING =

new Binary("realgraph.total\_dwell\_time.is\_missing", Set(CountOfImpression).asJava)

// the number of the target user's tweets that the source user has inspected

val NUM\_INSPECTED\_TWEETS\_MEAN =

new Continuous("realgraph.num\_inspected\_tweets.mean", Set(CountOfImpression).asJava)

val NUM\_INSPECTED\_TWEETS\_EWMA =

new Continuous("realgraph.num\_inspected\_tweets.ewma", Set(CountOfImpression).asJava)

val NUM\_INSPECTED\_TWEETS\_VARIANCE =

new Continuous("realgraph.num\_inspected\_tweets.variance", Set(CountOfImpression).asJava)

val NUM\_INSPECTED\_TWEETS\_NON\_ZERO\_DAYS = new Continuous(

"realgraph.num\_inspected\_tweets.non\_zero\_days",

Set(CountOfImpression).asJava

)

val NUM\_INSPECTED\_TWEETS\_ELAPSED\_DAYS = new Continuous(

"realgraph.num\_inspected\_tweets.elapsed\_days",

Set(CountOfImpression).asJava

)

val NUM\_INSPECTED\_TWEETS\_DAYS\_SINCE\_LAST = new Continuous(

"realgraph.num\_inspected\_tweets.days\_since\_last",

Set(CountOfImpression).asJava

)

val NUM\_INSPECTED\_TWEETS\_IS\_MISSING =

new Binary("realgraph.num\_inspected\_tweets.is\_missing", Set(CountOfImpression).asJava)

// the number of photos in which the source user has tagged the target user

val NUM\_PHOTO\_TAGS\_MEAN = new Continuous(

"realgraph.num\_photo\_tags.mean",

Set(EngagementsPrivate, EngagementsPublic).asJava)

val NUM\_PHOTO\_TAGS\_EWMA = new Continuous(

"realgraph.num\_photo\_tags.ewma",

Set(EngagementsPrivate, EngagementsPublic).asJava)

val NUM\_PHOTO\_TAGS\_VARIANCE = new Continuous(

"realgraph.num\_photo\_tags.variance",

Set(EngagementsPrivate, EngagementsPublic).asJava)

val NUM\_PHOTO\_TAGS\_NON\_ZERO\_DAYS = new Continuous(

"realgraph.num\_photo\_tags.non\_zero\_days",

Set(EngagementsPrivate, EngagementsPublic).asJava)

val NUM\_PHOTO\_TAGS\_ELAPSED\_DAYS = new Continuous(

"realgraph.num\_photo\_tags.elapsed\_days",

Set(EngagementsPrivate, EngagementsPublic).asJava)

val NUM\_PHOTO\_TAGS\_DAYS\_SINCE\_LAST = new Continuous(

"realgraph.num\_photo\_tags.days\_since\_last",

Set(EngagementsPrivate, EngagementsPublic).asJava)

val NUM\_PHOTO\_TAGS\_IS\_MISSING = new Binary(

"realgraph.num\_photo\_tags.is\_missing",

Set(EngagementsPrivate, EngagementsPublic).asJava)

val NUM\_FOLLOW\_MEAN = new Continuous(

"realgraph.num\_follow.mean",

Set(Follow, PrivateAccountsFollowedBy, PublicAccountsFollowedBy).asJava)

val NUM\_FOLLOW\_EWMA = new Continuous(

"realgraph.num\_follow.ewma",

Set(Follow, PrivateAccountsFollowedBy, PublicAccountsFollowedBy).asJava)

val NUM\_FOLLOW\_VARIANCE = new Continuous(

"realgraph.num\_follow.variance",

Set(Follow, PrivateAccountsFollowedBy, PublicAccountsFollowedBy).asJava)

val NUM\_FOLLOW\_NON\_ZERO\_DAYS = new Continuous(

"realgraph.num\_follow.non\_zero\_days",

Set(Follow, PrivateAccountsFollowedBy, PublicAccountsFollowedBy).asJava)

val NUM\_FOLLOW\_ELAPSED\_DAYS = new Continuous(

"realgraph.num\_follow.elapsed\_days",

Set(Follow, PrivateAccountsFollowedBy, PublicAccountsFollowedBy).asJava)

val NUM\_FOLLOW\_DAYS\_SINCE\_LAST = new Continuous(

"realgraph.num\_follow.days\_since\_last",

Set(Follow, PrivateAccountsFollowedBy, PublicAccountsFollowedBy).asJava)

val NUM\_FOLLOW\_IS\_MISSING = new Binary(

"realgraph.num\_follow.is\_missing",

Set(Follow, PrivateAccountsFollowedBy, PublicAccountsFollowedBy).asJava)

// the number of blocks that the source user sent to the destination user

val NUM\_BLOCKS\_MEAN =

new Continuous("realgraph.num\_blocks.mean", Set(CountOfBlocks).asJava)

val NUM\_BLOCKS\_EWMA =

new Continuous("realgraph.num\_blocks.ewma", Set(CountOfBlocks).asJava)

val NUM\_BLOCKS\_VARIANCE =

new Continuous("realgraph.num\_blocks.variance", Set(CountOfBlocks).asJava)

val NUM\_BLOCKS\_NON\_ZERO\_DAYS =

new Continuous("realgraph.num\_blocks.non\_zero\_days", Set(CountOfBlocks).asJava)

val NUM\_BLOCKS\_ELAPSED\_DAYS =

new Continuous("realgraph.num\_blocks.elapsed\_days", Set(CountOfBlocks).asJava)

val NUM\_BLOCKS\_DAYS\_SINCE\_LAST =

new Continuous("realgraph.num\_blocks.days\_since\_last", Set(CountOfBlocks).asJava)

val NUM\_BLOCKS\_IS\_MISSING =

new Binary("realgraph.num\_blocks.is\_missing", Set(CountOfBlocks).asJava)

// the number of mutes that the source user sent to the destination user

val NUM\_MUTES\_MEAN =

new Continuous("realgraph.num\_mutes.mean", Set(CountOfMutes).asJava)

val NUM\_MUTES\_EWMA =

new Continuous("realgraph.num\_mutes.ewma", Set(CountOfMutes).asJava)

val NUM\_MUTES\_VARIANCE =

new Continuous("realgraph.num\_mutes.variance", Set(CountOfMutes).asJava)

val NUM\_MUTES\_NON\_ZERO\_DAYS =

new Continuous("realgraph.num\_mutes.non\_zero\_days", Set(CountOfMutes).asJava)

val NUM\_MUTES\_ELAPSED\_DAYS =

new Continuous("realgraph.num\_mutes.elapsed\_days", Set(CountOfMutes).asJava)

val NUM\_MUTES\_DAYS\_SINCE\_LAST =

new Continuous("realgraph.num\_mutes.days\_since\_last", Set(CountOfMutes).asJava)

val NUM\_MUTES\_IS\_MISSING =

new Binary("realgraph.num\_mutes.is\_missing", Set(CountOfMutes).asJava)

// the number of report as abuses that the source user sent to the destination user

val NUM\_REPORTS\_AS\_ABUSES\_MEAN =

new Continuous("realgraph.num\_report\_as\_abuses.mean", Set(CountOfAbuseReports).asJava)

val NUM\_REPORTS\_AS\_ABUSES\_EWMA =

new Continuous("realgraph.num\_report\_as\_abuses.ewma", Set(CountOfAbuseReports).asJava)

val NUM\_REPORTS\_AS\_ABUSES\_VARIANCE =

new Continuous("realgraph.num\_report\_as\_abuses.variance", Set(CountOfAbuseReports).asJava)

val NUM\_REPORTS\_AS\_ABUSES\_NON\_ZERO\_DAYS =

new Continuous("realgraph.num\_report\_as\_abuses.non\_zero\_days", Set(CountOfAbuseReports).asJava)

val NUM\_REPORTS\_AS\_ABUSES\_ELAPSED\_DAYS =

new Continuous("realgraph.num\_report\_as\_abuses.elapsed\_days", Set(CountOfAbuseReports).asJava)

val NUM\_REPORTS\_AS\_ABUSES\_DAYS\_SINCE\_LAST =

new Continuous(

"realgraph.num\_report\_as\_abuses.days\_since\_last",

Set(CountOfAbuseReports).asJava)

val NUM\_REPORTS\_AS\_ABUSES\_IS\_MISSING =

new Binary("realgraph.num\_report\_as\_abuses.is\_missing", Set(CountOfAbuseReports).asJava)

// the number of report as spams that the source user sent to the destination user

val NUM\_REPORTS\_AS\_SPAMS\_MEAN =

new Continuous(

"realgraph.num\_report\_as\_spams.mean",

Set(CountOfAbuseReports, SafetyRelationships).asJava)

val NUM\_REPORTS\_AS\_SPAMS\_EWMA =

new Continuous(

"realgraph.num\_report\_as\_spams.ewma",

Set(CountOfAbuseReports, SafetyRelationships).asJava)

val NUM\_REPORTS\_AS\_SPAMS\_VARIANCE =

new Continuous(

"realgraph.num\_report\_as\_spams.variance",

Set(CountOfAbuseReports, SafetyRelationships).asJava)

val NUM\_REPORTS\_AS\_SPAMS\_NON\_ZERO\_DAYS =

new Continuous(

"realgraph.num\_report\_as\_spams.non\_zero\_days",

Set(CountOfAbuseReports, SafetyRelationships).asJava)

val NUM\_REPORTS\_AS\_SPAMS\_ELAPSED\_DAYS =

new Continuous(

"realgraph.num\_report\_as\_spams.elapsed\_days",

Set(CountOfAbuseReports, SafetyRelationships).asJava)

val NUM\_REPORTS\_AS\_SPAMS\_DAYS\_SINCE\_LAST =

new Continuous(

"realgraph.num\_report\_as\_spams.days\_since\_last",

Set(CountOfAbuseReports, SafetyRelationships).asJava)

val NUM\_REPORTS\_AS\_SPAMS\_IS\_MISSING =

new Binary(

"realgraph.num\_report\_as\_spams.is\_missing",

Set(CountOfAbuseReports, SafetyRelationships).asJava)

val NUM\_MUTUAL\_FOLLOW\_MEAN = new Continuous(

"realgraph.num\_mutual\_follow.mean",

Set(

Follow,

PrivateAccountsFollowedBy,

PublicAccountsFollowedBy,

PrivateAccountsFollowing,

PublicAccountsFollowing).asJava

)

val NUM\_MUTUAL\_FOLLOW\_EWMA = new Continuous(

"realgraph.num\_mutual\_follow.ewma",

Set(

Follow,

PrivateAccountsFollowedBy,

PublicAccountsFollowedBy,

PrivateAccountsFollowing,

PublicAccountsFollowing).asJava

)

val NUM\_MUTUAL\_FOLLOW\_VARIANCE = new Continuous(

"realgraph.num\_mutual\_follow.variance",

Set(

Follow,

PrivateAccountsFollowedBy,

PublicAccountsFollowedBy,

PrivateAccountsFollowing,

PublicAccountsFollowing).asJava

)

val NUM\_MUTUAL\_FOLLOW\_NON\_ZERO\_DAYS = new Continuous(

"realgraph.num\_mutual\_follow.non\_zero\_days",

Set(

Follow,

PrivateAccountsFollowedBy,

PublicAccountsFollowedBy,

PrivateAccountsFollowing,

PublicAccountsFollowing).asJava

)

val NUM\_MUTUAL\_FOLLOW\_ELAPSED\_DAYS = new Continuous(

"realgraph.num\_mutual\_follow.elapsed\_days",

Set(

Follow,

PrivateAccountsFollowedBy,

PublicAccountsFollowedBy,

PrivateAccountsFollowing,

PublicAccountsFollowing).asJava

)

val NUM\_MUTUAL\_FOLLOW\_DAYS\_SINCE\_LAST = new Continuous(

"realgraph.num\_mutual\_follow.days\_since\_last",

Set(

Follow,

PrivateAccountsFollowedBy,

PublicAccountsFollowedBy,

PrivateAccountsFollowing,

PublicAccountsFollowing).asJava

)

val NUM\_MUTUAL\_FOLLOW\_IS\_MISSING = new Binary(

"realgraph.num\_mutual\_follow.is\_missing",

Set(

Follow,

PrivateAccountsFollowedBy,

PublicAccountsFollowedBy,

PrivateAccountsFollowing,

PublicAccountsFollowing).asJava

)

val NUM\_SMS\_FOLLOW\_MEAN = new Continuous(

"realgraph.num\_sms\_follow.mean",

Set(Follow, PrivateAccountsFollowedBy, PublicAccountsFollowedBy).asJava)

val NUM\_SMS\_FOLLOW\_EWMA = new Continuous(

"realgraph.num\_sms\_follow.ewma",

Set(Follow, PrivateAccountsFollowedBy, PublicAccountsFollowedBy).asJava)

val NUM\_SMS\_FOLLOW\_VARIANCE = new Continuous(

"realgraph.num\_sms\_follow.variance",

Set(Follow, PrivateAccountsFollowedBy, PublicAccountsFollowedBy).asJava)

val NUM\_SMS\_FOLLOW\_NON\_ZERO\_DAYS = new Continuous(

"realgraph.num\_sms\_follow.non\_zero\_days",

Set(Follow, PrivateAccountsFollowedBy, PublicAccountsFollowedBy).asJava)

val NUM\_SMS\_FOLLOW\_ELAPSED\_DAYS = new Continuous(

"realgraph.num\_sms\_follow.elapsed\_days",

Set(Follow, PrivateAccountsFollowedBy, PublicAccountsFollowedBy).asJava)

val NUM\_SMS\_FOLLOW\_DAYS\_SINCE\_LAST = new Continuous(

"realgraph.num\_sms\_follow.days\_since\_last",

Set(Follow, PrivateAccountsFollowedBy, PublicAccountsFollowedBy).asJava)

val NUM\_SMS\_FOLLOW\_IS\_MISSING = new Binary(

"realgraph.num\_sms\_follow.is\_missing",

Set(Follow, PrivateAccountsFollowedBy, PublicAccountsFollowedBy).asJava)

val NUM\_ADDRESS\_BOOK\_EMAIL\_MEAN =

new Continuous("realgraph.num\_address\_book\_email.mean", Set(AddressBook).asJava)

val NUM\_ADDRESS\_BOOK\_EMAIL\_EWMA =

new Continuous("realgraph.num\_address\_book\_email.ewma", Set(AddressBook).asJava)

val NUM\_ADDRESS\_BOOK\_EMAIL\_VARIANCE =

new Continuous("realgraph.num\_address\_book\_email.variance", Set(AddressBook).asJava)

val NUM\_ADDRESS\_BOOK\_EMAIL\_NON\_ZERO\_DAYS = new Continuous(

"realgraph.num\_address\_book\_email.non\_zero\_days",

Set(AddressBook).asJava

)

val NUM\_ADDRESS\_BOOK\_EMAIL\_ELAPSED\_DAYS = new Continuous(

"realgraph.num\_address\_book\_email.elapsed\_days",

Set(AddressBook).asJava

)

val NUM\_ADDRESS\_BOOK\_EMAIL\_DAYS\_SINCE\_LAST = new Continuous(

"realgraph.num\_address\_book\_email.days\_since\_last",

Set(AddressBook).asJava

)

val NUM\_ADDRESS\_BOOK\_EMAIL\_IS\_MISSING =

new Binary("realgraph.num\_address\_book\_email.is\_missing", Set(AddressBook).asJava)

val NUM\_ADDRESS\_BOOK\_IN\_BOTH\_MEAN =

new Continuous("realgraph.num\_address\_book\_in\_both.mean", Set(AddressBook).asJava)

val NUM\_ADDRESS\_BOOK\_IN\_BOTH\_EWMA =

new Continuous("realgraph.num\_address\_book\_in\_both.ewma", Set(AddressBook).asJava)

val NUM\_ADDRESS\_BOOK\_IN\_BOTH\_VARIANCE = new Continuous(

"realgraph.num\_address\_book\_in\_both.variance",

Set(AddressBook).asJava

)

val NUM\_ADDRESS\_BOOK\_IN\_BOTH\_NON\_ZERO\_DAYS = new Continuous(

"realgraph.num\_address\_book\_in\_both.non\_zero\_days",

Set(AddressBook).asJava

)

val NUM\_ADDRESS\_BOOK\_IN\_BOTH\_ELAPSED\_DAYS = new Continuous(

"realgraph.num\_address\_book\_in\_both.elapsed\_days",

Set(AddressBook).asJava

)

val NUM\_ADDRESS\_BOOK\_IN\_BOTH\_DAYS\_SINCE\_LAST = new Continuous(

"realgraph.num\_address\_book\_in\_both.days\_since\_last",

Set(AddressBook).asJava

)

val NUM\_ADDRESS\_BOOK\_IN\_BOTH\_IS\_MISSING = new Binary(

"realgraph.num\_address\_book\_in\_both.is\_missing",

Set(AddressBook).asJava

)

val NUM\_ADDRESS\_BOOK\_PHONE\_MEAN =

new Continuous("realgraph.num\_address\_book\_phone.mean", Set(AddressBook).asJava)

val NUM\_ADDRESS\_BOOK\_PHONE\_EWMA =

new Continuous("realgraph.num\_address\_book\_phone.ewma", Set(AddressBook).asJava)

val NUM\_ADDRESS\_BOOK\_PHONE\_VARIANCE =

new Continuous("realgraph.num\_address\_book\_phone.variance", Set(AddressBook).asJava)

val NUM\_ADDRESS\_BOOK\_PHONE\_NON\_ZERO\_DAYS = new Continuous(

"realgraph.num\_address\_book\_phone.non\_zero\_days",

Set(AddressBook).asJava

)

val NUM\_ADDRESS\_BOOK\_PHONE\_ELAPSED\_DAYS = new Continuous(

"realgraph.num\_address\_book\_phone.elapsed\_days",

Set(AddressBook).asJava

)

val NUM\_ADDRESS\_BOOK\_PHONE\_DAYS\_SINCE\_LAST = new Continuous(

"realgraph.num\_address\_book\_phone.days\_since\_last",

Set(AddressBook).asJava

)

val NUM\_ADDRESS\_BOOK\_PHONE\_IS\_MISSING =

new Binary("realgraph.num\_address\_book\_phone.is\_missing", Set(AddressBook).asJava)

val NUM\_ADDRESS\_BOOK\_MUTUAL\_EDGE\_EMAIL\_MEAN =

new Continuous("realgraph.num\_address\_book\_mutual\_edge\_email.mean", Set(AddressBook).asJava)

val NUM\_ADDRESS\_BOOK\_MUTUAL\_EDGE\_EMAIL\_EWMA =

new Continuous("realgraph.num\_address\_book\_mutual\_edge\_email.ewma", Set(AddressBook).asJava)

val NUM\_ADDRESS\_BOOK\_MUTUAL\_EDGE\_EMAIL\_VARIANCE =

new Continuous("realgraph.num\_address\_book\_mutual\_edge\_email.variance", Set(AddressBook).asJava)

val NUM\_ADDRESS\_BOOK\_MUTUAL\_EDGE\_EMAIL\_NON\_ZERO\_DAYS = new Continuous(

"realgraph.num\_address\_book\_mutual\_edge\_email.non\_zero\_days",

Set(AddressBook).asJava

)

val NUM\_ADDRESS\_BOOK\_MUTUAL\_EDGE\_EMAIL\_ELAPSED\_DAYS = new Continuous(

"realgraph.num\_address\_book\_mutual\_edge\_email.elapsed\_days",

Set(AddressBook).asJava

)

val NUM\_ADDRESS\_BOOK\_MUTUAL\_EDGE\_EMAIL\_DAYS\_SINCE\_LAST = new Continuous(

"realgraph.num\_address\_book\_mutual\_edge\_email.days\_since\_last",

Set(AddressBook).asJava

)

val NUM\_ADDRESS\_BOOK\_MUTUAL\_EDGE\_EMAIL\_IS\_MISSING =

new Binary("realgraph.num\_address\_book\_mutual\_edge\_email.is\_missing", Set(AddressBook).asJava)

val NUM\_ADDRESS\_BOOK\_MUTUAL\_EDGE\_IN\_BOTH\_MEAN =

new Continuous("realgraph.num\_address\_book\_mutual\_edge\_in\_both.mean", Set(AddressBook).asJava)

val NUM\_ADDRESS\_BOOK\_MUTUAL\_EDGE\_IN\_BOTH\_EWMA =

new Continuous("realgraph.num\_address\_book\_mutual\_edge\_in\_both.ewma", Set(AddressBook).asJava)

val NUM\_ADDRESS\_BOOK\_MUTUAL\_EDGE\_IN\_BOTH\_VARIANCE = new Continuous(

"realgraph.num\_address\_book\_mutual\_edge\_in\_both.variance",

Set(AddressBook).asJava

)

val NUM\_ADDRESS\_BOOK\_MUTUAL\_EDGE\_IN\_BOTH\_NON\_ZERO\_DAYS = new Continuous(

"realgraph.num\_address\_book\_mutual\_edge\_in\_both.non\_zero\_days",

Set(AddressBook).asJava

)

val NUM\_ADDRESS\_BOOK\_MUTUAL\_EDGE\_IN\_BOTH\_ELAPSED\_DAYS = new Continuous(

"realgraph.num\_address\_book\_mutual\_edge\_in\_both.elapsed\_days",

Set(AddressBook).asJava

)

val NUM\_ADDRESS\_BOOK\_MUTUAL\_EDGE\_IN\_BOTH\_DAYS\_SINCE\_LAST = new Continuous(

"realgraph.num\_address\_book\_mutual\_edge\_in\_both.days\_since\_last",

Set(AddressBook).asJava

)

val NUM\_ADDRESS\_BOOK\_MUTUAL\_EDGE\_IN\_BOTH\_IS\_MISSING = new Binary(

"realgraph.num\_address\_book\_mutual\_edge\_in\_both.is\_missing",

Set(AddressBook).asJava

)

val NUM\_ADDRESS\_BOOK\_MUTUAL\_EDGE\_PHONE\_MEAN =

new Continuous("realgraph.num\_address\_book\_mutual\_edge\_phone.mean", Set(AddressBook).asJava)

val NUM\_ADDRESS\_BOOK\_MUTUAL\_EDGE\_PHONE\_EWMA =

new Continuous("realgraph.num\_address\_book\_mutual\_edge\_phone.ewma", Set(AddressBook).asJava)

val NUM\_ADDRESS\_BOOK\_MUTUAL\_EDGE\_PHONE\_VARIANCE =

new Continuous("realgraph.num\_address\_book\_mutual\_edge\_phone.variance", Set(AddressBook).asJava)

val NUM\_ADDRESS\_BOOK\_MUTUAL\_EDGE\_PHONE\_NON\_ZERO\_DAYS = new Continuous(

"realgraph.num\_address\_book\_mutual\_edge\_phone.non\_zero\_days",

Set(AddressBook).asJava

)

val NUM\_ADDRESS\_BOOK\_MUTUAL\_EDGE\_PHONE\_ELAPSED\_DAYS = new Continuous(

"realgraph.num\_address\_book\_mutual\_edge\_phone.elapsed\_days",

Set(AddressBook).asJava

)

val NUM\_ADDRESS\_BOOK\_MUTUAL\_EDGE\_PHONE\_DAYS\_SINCE\_LAST = new Continuous(

"realgraph.num\_address\_book\_mutual\_edge\_phone.days\_since\_last",

Set(AddressBook).asJava

)

val NUM\_ADDRESS\_BOOK\_MUTUAL\_EDGE\_PHONE\_IS\_MISSING =

new Binary("realgraph.num\_address\_book\_mutual\_edge\_phone.is\_missing", Set(AddressBook).asJava)

}

case class RealGraphEdgeDataRecordFeatures(

edgeFeatureOpt: Option[RealGraphEdgeFeature],

meanFeature: Continuous,

ewmaFeature: Continuous,

varianceFeature: Continuous,

nonZeroDaysFeature: Continuous,

elapsedDaysFeature: Continuous,

daysSinceLastFeature: Continuous,

isMissingFeature: Binary)