package com.twitter.timelines.prediction.features.recap

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

import com.twitter.ml.api.Feature.Binary

import com.twitter.ml.api.Feature.Continuous

import com.twitter.ml.api.Feature.Discrete

import com.twitter.ml.api.Feature.SparseBinary

import com.twitter.ml.api.Feature.Text

import scala.collection.JavaConverters.\_

object RecapFeatures extends RecapFeatures("")

object InReplyToRecapFeatures extends RecapFeatures("in\_reply\_to\_tweet")

class RecapFeatures(prefix: String) {

private def name(featureName: String): String = {

if (prefix.nonEmpty) {

s"$prefix.$featureName"

} else {

featureName

}

}

val IS\_IPAD\_CLIENT = new Binary(name("recap.client.is\_ipad"), Set(ClientType).asJava)

val IS\_WEB\_CLIENT = new Binary(name("recap.client.is\_web"), Set(ClientType).asJava)

val IS\_IPHONE\_CLIENT = new Binary(name("recap.client.is\_phone"), Set(ClientType).asJava)

val IS\_ANDROID\_CLIENT = new Binary(name("recap.client.is\_android"), Set(ClientType).asJava)

val IS\_ANDROID\_TABLET\_CLIENT =

new Binary(name("recap.client.is\_android\_tablet"), Set(ClientType).asJava)

// features from userAgent

val CLIENT\_NAME = new Text(name("recap.user\_agent.client\_name"), Set(ClientType).asJava)

val CLIENT\_SOURCE = new Discrete(name("recap.user\_agent.client\_source"), Set(ClientType).asJava)

val CLIENT\_VERSION = new Text(name("recap.user\_agent.client\_version"), Set(ClientVersion).asJava)

val CLIENT\_VERSION\_CODE =

new Text(name("recap.user\_agent.client\_version\_code"), Set(ClientVersion).asJava)

val DEVICE = new Text(name("recap.user\_agent.device"), Set(DeviceType).asJava)

val FROM\_DOG\_FOOD = new Binary(name("recap.meta.from\_dog\_food"), Set(UserAgent).asJava)

val FROM\_TWITTER\_CLIENT =

new Binary(name("recap.user\_agent.from\_twitter\_client"), Set(UserAgent).asJava)

val MANUFACTURER = new Text(name("recap.user\_agent.manufacturer"), Set(UserAgent).asJava)

val MODEL = new Text(name("recap.user\_agent.model"), Set(UserAgent).asJava)

val NETWORK\_CONNECTION =

new Discrete(name("recap.user\_agent.network\_connection"), Set(UserAgent).asJava)

val SDK\_VERSION = new Text(name("recap.user\_agent.sdk\_version"), Set(AppId, UserAgent).asJava)

// engagement

val IS\_RETWEETED = new Binary(

name("recap.engagement.is\_retweeted"),

Set(PublicRetweets, PrivateRetweets, EngagementsPrivate, EngagementsPublic).asJava)

val IS\_FAVORITED = new Binary(

name("recap.engagement.is\_favorited"),

Set(PublicLikes, PrivateLikes, EngagementsPrivate, EngagementsPublic).asJava)

val IS\_REPLIED = new Binary(

name("recap.engagement.is\_replied"),

Set(PublicReplies, PrivateReplies, EngagementsPrivate, EngagementsPublic).asJava)

// v1: post click engagements: fav, reply

val IS\_GOOD\_CLICKED\_CONVO\_DESC\_V1 = new Binary(

name("recap.engagement.is\_good\_clicked\_convo\_desc\_favorited\_or\_replied"),

Set(

PublicLikes,

PrivateLikes,

PublicReplies,

PrivateReplies,

EngagementsPrivate,

EngagementsPublic).asJava)

// v2: post click engagements: click

val IS\_GOOD\_CLICKED\_CONVO\_DESC\_V2 = new Binary(

name("recap.engagement.is\_good\_clicked\_convo\_desc\_v2"),

Set(TweetsClicked, EngagementsPrivate).asJava)

val IS\_GOOD\_CLICKED\_CONVO\_DESC\_FAVORITED = new Binary(

name("recap.engagement.is\_good\_clicked\_convo\_desc\_favorited"),

Set(PublicLikes, PrivateLikes, EngagementsPrivate, EngagementsPublic).asJava)

val IS\_GOOD\_CLICKED\_CONVO\_DESC\_REPLIED = new Binary(

name("recap.engagement.is\_good\_clicked\_convo\_desc\_replied"),

Set(PublicReplies, PrivateReplies, EngagementsPrivate, EngagementsPublic).asJava)

val IS\_GOOD\_CLICKED\_CONVO\_DESC\_RETWEETED = new Binary(

name("recap.engagement.is\_good\_clicked\_convo\_desc\_retweeted"),

Set(PublicRetweets, PrivateRetweets, EngagementsPrivate, EngagementsPublic).asJava)

val IS\_GOOD\_CLICKED\_CONVO\_DESC\_CLICKED = new Binary(

name("recap.engagement.is\_good\_clicked\_convo\_desc\_clicked"),

Set(TweetsClicked, EngagementsPrivate).asJava)

val IS\_GOOD\_CLICKED\_CONVO\_DESC\_FOLLOWED = new Binary(

name("recap.engagement.is\_good\_clicked\_convo\_desc\_followed"),

Set(EngagementsPrivate).asJava)

val IS\_GOOD\_CLICKED\_CONVO\_DESC\_SHARE\_DM\_CLICKED = new Binary(

name("recap.engagement.is\_good\_clicked\_convo\_desc\_share\_dm\_clicked"),

Set(EngagementsPrivate).asJava)

val IS\_GOOD\_CLICKED\_CONVO\_DESC\_PROFILE\_CLICKED = new Binary(

name("recap.engagement.is\_good\_clicked\_convo\_desc\_profile\_clicked"),

Set(EngagementsPrivate).asJava)

val IS\_GOOD\_CLICKED\_CONVO\_DESC\_UAM\_GT\_0 = new Binary(

name("recap.engagement.is\_good\_clicked\_convo\_desc\_uam\_gt\_0"),

Set(EngagementsPrivate, EngagementsPublic).asJava)

val IS\_GOOD\_CLICKED\_CONVO\_DESC\_UAM\_GT\_1 = new Binary(

name("recap.engagement.is\_good\_clicked\_convo\_desc\_uam\_gt\_1"),

Set(EngagementsPrivate, EngagementsPublic).asJava)

val IS\_GOOD\_CLICKED\_CONVO\_DESC\_UAM\_GT\_2 = new Binary(

name("recap.engagement.is\_good\_clicked\_convo\_desc\_uam\_gt\_2"),

Set(EngagementsPrivate, EngagementsPublic).asJava)

val IS\_GOOD\_CLICKED\_CONVO\_DESC\_UAM\_GT\_3 = new Binary(

name("recap.engagement.is\_good\_clicked\_convo\_desc\_uam\_gt\_3"),

Set(EngagementsPrivate, EngagementsPublic).asJava)

val IS\_TWEET\_DETAIL\_DWELLED = new Binary(

name("recap.engagement.is\_tweet\_detail\_dwelled"),

Set(TweetsClicked, EngagementsPrivate).asJava)

val IS\_TWEET\_DETAIL\_DWELLED\_8\_SEC = new Binary(

name("recap.engagement.is\_tweet\_detail\_dwelled\_8\_sec"),

Set(TweetsClicked, EngagementsPrivate).asJava)

val IS\_TWEET\_DETAIL\_DWELLED\_15\_SEC = new Binary(

name("recap.engagement.is\_tweet\_detail\_dwelled\_15\_sec"),

Set(TweetsClicked, EngagementsPrivate).asJava)

val IS\_TWEET\_DETAIL\_DWELLED\_25\_SEC = new Binary(

name("recap.engagement.is\_tweet\_detail\_dwelled\_25\_sec"),

Set(TweetsClicked, EngagementsPrivate).asJava)

val IS\_TWEET\_DETAIL\_DWELLED\_30\_SEC = new Binary(

name("recap.engagement.is\_tweet\_detail\_dwelled\_30\_sec"),

Set(TweetsClicked, EngagementsPrivate).asJava)

val IS\_PROFILE\_DWELLED = new Binary(

"recap.engagement.is\_profile\_dwelled",

Set(ProfilesViewed, ProfilesClicked, EngagementsPrivate).asJava)

val IS\_PROFILE\_DWELLED\_10\_SEC = new Binary(

"recap.engagement.is\_profile\_dwelled\_10\_sec",

Set(ProfilesViewed, ProfilesClicked, EngagementsPrivate).asJava)

val IS\_PROFILE\_DWELLED\_20\_SEC = new Binary(

"recap.engagement.is\_profile\_dwelled\_20\_sec",

Set(ProfilesViewed, ProfilesClicked, EngagementsPrivate).asJava)

val IS\_PROFILE\_DWELLED\_30\_SEC = new Binary(

"recap.engagement.is\_profile\_dwelled\_30\_sec",

Set(ProfilesViewed, ProfilesClicked, EngagementsPrivate).asJava)

val IS\_FULLSCREEN\_VIDEO\_DWELLED = new Binary(

"recap.engagement.is\_fullscreen\_video\_dwelled",

Set(MediaEngagementActivities, EngagementTypePrivate, EngagementsPrivate).asJava)

val IS\_FULLSCREEN\_VIDEO\_DWELLED\_5\_SEC = new Binary(

"recap.engagement.is\_fullscreen\_video\_dwelled\_5\_sec",

Set(MediaEngagementActivities, EngagementTypePrivate, EngagementsPrivate).asJava)

val IS\_FULLSCREEN\_VIDEO\_DWELLED\_10\_SEC = new Binary(

"recap.engagement.is\_fullscreen\_video\_dwelled\_10\_sec",

Set(MediaEngagementActivities, EngagementTypePrivate, EngagementsPrivate).asJava)

val IS\_FULLSCREEN\_VIDEO\_DWELLED\_20\_SEC = new Binary(

"recap.engagement.is\_fullscreen\_video\_dwelled\_20\_sec",

Set(MediaEngagementActivities, EngagementTypePrivate, EngagementsPrivate).asJava)

val IS\_FULLSCREEN\_VIDEO\_DWELLED\_30\_SEC = new Binary(

"recap.engagement.is\_fullscreen\_video\_dwelled\_30\_sec",

Set(MediaEngagementActivities, EngagementTypePrivate, EngagementsPrivate).asJava)

val IS\_LINK\_DWELLED\_15\_SEC = new Binary(

"recap.engagement.is\_link\_dwelled\_15\_sec",

Set(MediaEngagementActivities, EngagementTypePrivate, EngagementsPrivate).asJava)

val IS\_LINK\_DWELLED\_30\_SEC = new Binary(

"recap.engagement.is\_link\_dwelled\_30\_sec",

Set(MediaEngagementActivities, EngagementTypePrivate, EngagementsPrivate).asJava)

val IS\_LINK\_DWELLED\_60\_SEC = new Binary(

"recap.engagement.is\_link\_dwelled\_60\_sec",

Set(MediaEngagementActivities, EngagementTypePrivate, EngagementsPrivate).asJava)

val IS\_QUOTED = new Binary(

name("recap.engagement.is\_quoted"),

Set(PublicRetweets, PrivateRetweets, EngagementsPrivate, EngagementsPublic).asJava)

val IS\_RETWEETED\_WITHOUT\_QUOTE = new Binary(

name("recap.engagement.is\_retweeted\_without\_quote"),

Set(PublicRetweets, PrivateRetweets, EngagementsPrivate, EngagementsPublic).asJava)

val IS\_CLICKED =

new Binary(name("recap.engagement.is\_clicked"), Set(TweetsClicked, EngagementsPrivate).asJava)

val IS\_DWELLED = new Binary(name("recap.engagement.is\_dwelled"), Set(EngagementsPrivate).asJava)

val IS\_DWELLED\_IN\_BOUNDS\_V1 =

new Binary(name("recap.engagement.is\_dwelled\_in\_bounds\_v1"), Set(EngagementsPrivate).asJava)

val DWELL\_NORMALIZED\_OVERALL = new Continuous(

name("recap.engagement.dwell\_normalized\_overall"),

Set(EngagementsPrivate).asJava)

val DWELL\_CDF\_OVERALL =

new Continuous(name("recap.engagement.dwell\_cdf\_overall"), Set(EngagementsPrivate).asJava)

val DWELL\_CDF = new Continuous(name("recap.engagement.dwell\_cdf"), Set(EngagementsPrivate).asJava)

val IS\_DWELLED\_1S =

new Binary(name("recap.engagement.is\_dwelled\_1s"), Set(EngagementsPrivate).asJava)

val IS\_DWELLED\_2S =

new Binary(name("recap.engagement.is\_dwelled\_2s"), Set(EngagementsPrivate).asJava)

val IS\_DWELLED\_3S =

new Binary(name("recap.engagement.is\_dwelled\_3s"), Set(EngagementsPrivate).asJava)

val IS\_DWELLED\_4S =

new Binary(name("recap.engagement.is\_dwelled\_4s"), Set(EngagementsPrivate).asJava)

val IS\_DWELLED\_5S =

new Binary(name("recap.engagement.is\_dwelled\_5s"), Set(EngagementsPrivate).asJava)

val IS\_DWELLED\_6S =

new Binary(name("recap.engagement.is\_dwelled\_6s"), Set(EngagementsPrivate).asJava)

val IS\_DWELLED\_7S =

new Binary(name("recap.engagement.is\_dwelled\_7s"), Set(EngagementsPrivate).asJava)

val IS\_DWELLED\_8S =

new Binary(name("recap.engagement.is\_dwelled\_8s"), Set(EngagementsPrivate).asJava)

val IS\_DWELLED\_9S =

new Binary(name("recap.engagement.is\_dwelled\_9s"), Set(EngagementsPrivate).asJava)

val IS\_DWELLED\_10S =

new Binary(name("recap.engagement.is\_dwelled\_10s"), Set(EngagementsPrivate).asJava)

val IS\_SKIPPED\_1S =

new Binary(name("recap.engagement.is\_skipped\_1s"), Set(EngagementsPrivate).asJava)

val IS\_SKIPPED\_2S =

new Binary(name("recap.engagement.is\_skipped\_2s"), Set(EngagementsPrivate).asJava)

val IS\_SKIPPED\_3S =

new Binary(name("recap.engagement.is\_skipped\_3s"), Set(EngagementsPrivate).asJava)

val IS\_SKIPPED\_4S =

new Binary(name("recap.engagement.is\_skipped\_4s"), Set(EngagementsPrivate).asJava)

val IS\_SKIPPED\_5S =

new Binary(name("recap.engagement.is\_skipped\_5s"), Set(EngagementsPrivate).asJava)

val IS\_SKIPPED\_6S =

new Binary(name("recap.engagement.is\_skipped\_6s"), Set(EngagementsPrivate).asJava)

val IS\_SKIPPED\_7S =

new Binary(name("recap.engagement.is\_skipped\_7s"), Set(EngagementsPrivate).asJava)

val IS\_SKIPPED\_8S =

new Binary(name("recap.engagement.is\_skipped\_8s"), Set(EngagementsPrivate).asJava)

val IS\_SKIPPED\_9S =

new Binary(name("recap.engagement.is\_skipped\_9s"), Set(EngagementsPrivate).asJava)

val IS\_SKIPPED\_10S =

new Binary(name("recap.engagement.is\_skipped\_10s"), Set(EngagementsPrivate).asJava)

val IS\_IMPRESSED =

new Binary(name("recap.engagement.is\_impressed"), Set(EngagementsPrivate).asJava)

val IS\_FOLLOWED =

new Binary("recap.engagement.is\_followed", Set(EngagementsPrivate, EngagementsPublic).asJava)

val IS\_PROFILE\_CLICKED = new Binary(

name("recap.engagement.is\_profile\_clicked"),

Set(ProfilesViewed, ProfilesClicked, EngagementsPrivate).asJava)

val IS\_OPEN\_LINKED = new Binary(

name("recap.engagement.is\_open\_linked"),

Set(EngagementsPrivate, LinksClickedOn).asJava)

val IS\_PHOTO\_EXPANDED =

new Binary(name("recap.engagement.is\_photo\_expanded"), Set(EngagementsPrivate).asJava)

val IS\_VIDEO\_VIEWED =

new Binary(name("recap.engagement.is\_video\_viewed"), Set(EngagementsPrivate).asJava)

val IS\_VIDEO\_PLAYBACK\_START =

new Binary(name("recap.engagement.is\_video\_playback\_start"), Set(EngagementsPrivate).asJava)

val IS\_VIDEO\_PLAYBACK\_25 =

new Binary(name("recap.engagement.is\_video\_playback\_25"), Set(EngagementsPrivate).asJava)

val IS\_VIDEO\_PLAYBACK\_50 =

new Binary(name("recap.engagement.is\_video\_playback\_50"), Set(EngagementsPrivate).asJava)

val IS\_VIDEO\_PLAYBACK\_75 =

new Binary(name("recap.engagement.is\_video\_playback\_75"), Set(EngagementsPrivate).asJava)

val IS\_VIDEO\_PLAYBACK\_95 =

new Binary(name("recap.engagement.is\_video\_playback\_95"), Set(EngagementsPrivate).asJava)

val IS\_VIDEO\_PLAYBACK\_COMPLETE =

new Binary(name("recap.engagement.is\_video\_playback\_complete"), Set(EngagementsPrivate).asJava)

val IS\_VIDEO\_VIEWED\_AND\_PLAYBACK\_50 = new Binary(

name("recap.engagement.is\_video\_viewed\_and\_playback\_50"),

Set(EngagementsPrivate).asJava)

val IS\_VIDEO\_QUALITY\_VIEWED = new Binary(

name("recap.engagement.is\_video\_quality\_viewed"),

Set(EngagementsPrivate).asJava

)

val IS\_TWEET\_SHARE\_DM\_CLICKED =

new Binary(name("recap.engagement.is\_tweet\_share\_dm\_clicked"), Set(EngagementsPrivate).asJava)

val IS\_TWEET\_SHARE\_DM\_SENT =

new Binary(name("recap.engagement.is\_tweet\_share\_dm\_sent"), Set(EngagementsPrivate).asJava)

val IS\_BOOKMARKED =

new Binary(name("recap.engagement.is\_bookmarked"), Set(EngagementsPrivate).asJava)

val IS\_SHARED =

new Binary(name("recap.engagement.is\_shared"), Set(EngagementsPrivate).asJava)

val IS\_SHARE\_MENU\_CLICKED =

new Binary(name("recap.engagement.is\_share\_menu\_clicked"), Set(EngagementsPrivate).asJava)

// Negative engagements

val IS\_DONT\_LIKE =

new Binary(name("recap.engagement.is\_dont\_like"), Set(EngagementsPrivate).asJava)

val IS\_BLOCK\_CLICKED = new Binary(

name("recap.engagement.is\_block\_clicked"),

Set(TweetsClicked, EngagementsPrivate, EngagementsPublic).asJava)

val IS\_BLOCK\_DIALOG\_BLOCKED = new Binary(

name("recap.engagement.is\_block\_dialog\_blocked"),

Set(EngagementsPrivate, EngagementsPublic).asJava)

val IS\_MUTE\_CLICKED = new Binary(

name("recap.engagement.is\_mute\_clicked"),

Set(TweetsClicked, EngagementsPrivate).asJava)

val IS\_MUTE\_DIALOG\_MUTED =

new Binary(name("recap.engagement.is\_mute\_dialog\_muted"), Set(EngagementsPrivate).asJava)

val IS\_REPORT\_TWEET\_CLICKED = new Binary(

name("recap.engagement.is\_report\_tweet\_clicked"),

Set(TweetsClicked, EngagementsPrivate).asJava)

val IS\_NEGATIVE\_FEEDBACK =

new Binary("recap.engagement.is\_negative\_feedback", Set(EngagementsPrivate).asJava)

val IS\_NOT\_ABOUT\_TOPIC =

new Binary(name("recap.engagement.is\_not\_about\_topic"), Set(EngagementsPrivate).asJava)

val IS\_NOT\_RECENT =

new Binary(name("recap.engagement.is\_not\_recent"), Set(EngagementsPrivate).asJava)

val IS\_NOT\_RELEVANT =

new Binary(name("recap.engagement.is\_not\_relevant"), Set(EngagementsPrivate).asJava)

val IS\_SEE\_FEWER =

new Binary(name("recap.engagement.is\_see\_fewer"), Set(EngagementsPrivate).asJava)

val IS\_TOPIC\_SPEC\_NEG\_ENGAGEMENT =

new Binary("recap.engagement.is\_topic\_spec\_neg\_engagement", Set(EngagementsPrivate).asJava)

val IS\_UNFOLLOW\_TOPIC =

new Binary("recap.engagement.is\_unfollow\_topic", Set(EngagementsPrivate).asJava)

val IS\_UNFOLLOW\_TOPIC\_EXPLICIT\_POSITIVE\_LABEL =

new Binary(

"recap.engagement.is\_unfollow\_topic\_explicit\_positive\_label",

Set(EngagementsPrivate).asJava)

val IS\_UNFOLLOW\_TOPIC\_IMPLICIT\_POSITIVE\_LABEL =

new Binary(

"recap.engagement.is\_unfollow\_topic\_implicit\_positive\_label",

Set(EngagementsPrivate).asJava)

val IS\_UNFOLLOW\_TOPIC\_STRONG\_EXPLICIT\_NEGATIVE\_LABEL =

new Binary(

"recap.engagement.is\_unfollow\_topic\_strong\_explicit\_negative\_label",

Set(EngagementsPrivate).asJava)

val IS\_UNFOLLOW\_TOPIC\_EXPLICIT\_NEGATIVE\_LABEL =

new Binary(

"recap.engagement.is\_unfollow\_topic\_explicit\_negative\_label",

Set(EngagementsPrivate).asJava)

val IS\_NOT\_INTERESTED\_IN =

new Binary("recap.engagement.is\_not\_interested\_in", Set(EngagementsPrivate).asJava)

val IS\_NOT\_INTERESTED\_IN\_EXPLICIT\_POSITIVE\_LABEL =

new Binary(

"recap.engagement.is\_not\_interested\_in\_explicit\_positive\_label",

Set(EngagementsPrivate).asJava)

val IS\_NOT\_INTERESTED\_IN\_EXPLICIT\_NEGATIVE\_LABEL =

new Binary(

"recap.engagement.is\_not\_interested\_in\_explicit\_negative\_label",

Set(EngagementsPrivate).asJava)

val IS\_CARET\_CLICKED =

new Binary(name("recap.engagement.is\_caret\_clicked"), Set(EngagementsPrivate).asJava)

val IS\_FOLLOW\_TOPIC =

new Binary("recap.engagement.is\_follow\_topic", Set(EngagementsPrivate).asJava)

val IS\_NOT\_INTERESTED\_IN\_TOPIC =

new Binary("recap.engagement.is\_not\_interested\_in\_topic", Set(EngagementsPrivate).asJava)

val IS\_HOME\_LATEST\_VISITED =

new Binary(name("recap.engagement.is\_home\_latest\_visited"), Set(EngagementsPrivate).asJava)

// Relevance prompt tweet engagements

val IS\_RELEVANCE\_PROMPT\_YES\_CLICKED = new Binary(

name("recap.engagement.is\_relevance\_prompt\_yes\_clicked"),

Set(EngagementsPrivate).asJava)

val IS\_RELEVANCE\_PROMPT\_NO\_CLICKED = new Binary(

name("recap.engagement.is\_relevance\_prompt\_no\_clicked"),

Set(EngagementsPrivate).asJava)

val IS\_RELEVANCE\_PROMPT\_IMPRESSED = new Binary(

name("recap.engagement.is\_relevance\_prompt\_impressed"),

Set(EngagementsPrivate).asJava)

// Reciprocal engagements for reply forward engagement

val IS\_REPLIED\_REPLY\_IMPRESSED\_BY\_AUTHOR = new Binary(

name("recap.engagement.is\_replied\_reply\_impressed\_by\_author"),

Set(EngagementsPrivate).asJava)

val IS\_REPLIED\_REPLY\_FAVORITED\_BY\_AUTHOR = new Binary(

name("recap.engagement.is\_replied\_reply\_favorited\_by\_author"),

Set(EngagementsPrivate, EngagementsPublic, PrivateLikes, PublicLikes).asJava)

val IS\_REPLIED\_REPLY\_QUOTED\_BY\_AUTHOR = new Binary(

name("recap.engagement.is\_replied\_reply\_quoted\_by\_author"),

Set(EngagementsPrivate, EngagementsPublic, PrivateRetweets, PublicRetweets).asJava)

val IS\_REPLIED\_REPLY\_REPLIED\_BY\_AUTHOR = new Binary(

name("recap.engagement.is\_replied\_reply\_replied\_by\_author"),

Set(EngagementsPrivate, EngagementsPublic, PrivateReplies, PublicReplies).asJava)

val IS\_REPLIED\_REPLY\_RETWEETED\_BY\_AUTHOR = new Binary(

name("recap.engagement.is\_replied\_reply\_retweeted\_by\_author"),

Set(EngagementsPrivate, EngagementsPublic, PrivateRetweets, PublicRetweets).asJava)

val IS\_REPLIED\_REPLY\_BLOCKED\_BY\_AUTHOR = new Binary(

name("recap.engagement.is\_replied\_reply\_blocked\_by\_author"),

Set(EngagementsPrivate, EngagementsPublic).asJava)

val IS\_REPLIED\_REPLY\_FOLLOWED\_BY\_AUTHOR = new Binary(

name("recap.engagement.is\_replied\_reply\_followed\_by\_author"),

Set(EngagementsPrivate, EngagementsPublic, Follow).asJava)

val IS\_REPLIED\_REPLY\_UNFOLLOWED\_BY\_AUTHOR = new Binary(

name("recap.engagement.is\_replied\_reply\_unfollowed\_by\_author"),

Set(EngagementsPrivate, EngagementsPublic).asJava)

val IS\_REPLIED\_REPLY\_MUTED\_BY\_AUTHOR = new Binary(

name("recap.engagement.is\_replied\_reply\_muted\_by\_author"),

Set(EngagementsPrivate).asJava)

val IS\_REPLIED\_REPLY\_REPORTED\_BY\_AUTHOR = new Binary(

name("recap.engagement.is\_replied\_reply\_reported\_by\_author"),

Set(EngagementsPrivate).asJava)

// This derived label is the logical OR of REPLY\_REPLIED, REPLY\_FAVORITED, REPLY\_RETWEETED

val IS\_REPLIED\_REPLY\_ENGAGED\_BY\_AUTHOR = new Binary(

name("recap.engagement.is\_replied\_reply\_engaged\_by\_author"),

Set(EngagementsPrivate, EngagementsPublic).asJava)

// Reciprocal engagements for fav forward engagement

val IS\_FAVORITED\_FAV\_FAVORITED\_BY\_AUTHOR = new Binary(

name("recap.engagement.is\_favorited\_fav\_favorited\_by\_author"),

Set(EngagementsPrivate, EngagementsPublic, PrivateLikes, PublicLikes).asJava

)

val IS\_FAVORITED\_FAV\_REPLIED\_BY\_AUTHOR = new Binary(

name("recap.engagement.is\_favorited\_fav\_replied\_by\_author"),

Set(EngagementsPrivate, EngagementsPublic, PrivateReplies, PublicReplies).asJava

)

val IS\_FAVORITED\_FAV\_RETWEETED\_BY\_AUTHOR = new Binary(

name("recap.engagement.is\_favorited\_fav\_retweeted\_by\_author"),

Set(EngagementsPrivate, EngagementsPublic, PrivateRetweets, PublicRetweets).asJava

)

val IS\_FAVORITED\_FAV\_FOLLOWED\_BY\_AUTHOR = new Binary(

name("recap.engagement.is\_favorited\_fav\_followed\_by\_author"),

Set(EngagementsPrivate, EngagementsPublic, PrivateRetweets, PublicRetweets).asJava

)

// This derived label is the logical OR of FAV\_REPLIED, FAV\_FAVORITED, FAV\_RETWEETED, FAV\_FOLLOWED

val IS\_FAVORITED\_FAV\_ENGAGED\_BY\_AUTHOR = new Binary(

name("recap.engagement.is\_favorited\_fav\_engaged\_by\_author"),

Set(EngagementsPrivate, EngagementsPublic).asJava)

// define good profile click by considering following engagements (follow, fav, reply, retweet, etc.) at profile page

val IS\_PROFILE\_CLICKED\_AND\_PROFILE\_FOLLOW = new Binary(

name("recap.engagement.is\_profile\_clicked\_and\_profile\_follow"),

Set(ProfilesViewed, ProfilesClicked, EngagementsPrivate, Follow).asJava)

val IS\_PROFILE\_CLICKED\_AND\_PROFILE\_FAV = new Binary(

name("recap.engagement.is\_profile\_clicked\_and\_profile\_fav"),

Set(ProfilesViewed, ProfilesClicked, EngagementsPrivate, PrivateLikes, PublicLikes).asJava)

val IS\_PROFILE\_CLICKED\_AND\_PROFILE\_REPLY = new Binary(

name("recap.engagement.is\_profile\_clicked\_and\_profile\_reply"),

Set(ProfilesViewed, ProfilesClicked, EngagementsPrivate, PrivateReplies, PublicReplies).asJava)

val IS\_PROFILE\_CLICKED\_AND\_PROFILE\_RETWEET = new Binary(

name("recap.engagement.is\_profile\_clicked\_and\_profile\_retweet"),

Set(

ProfilesViewed,

ProfilesClicked,

EngagementsPrivate,

PrivateRetweets,

PublicRetweets).asJava)

val IS\_PROFILE\_CLICKED\_AND\_PROFILE\_TWEET\_CLICK = new Binary(

name("recap.engagement.is\_profile\_clicked\_and\_profile\_tweet\_click"),

Set(ProfilesViewed, ProfilesClicked, EngagementsPrivate, TweetsClicked).asJava)

val IS\_PROFILE\_CLICKED\_AND\_PROFILE\_SHARE\_DM\_CLICK = new Binary(

name("recap.engagement.is\_profile\_clicked\_and\_profile\_share\_dm\_click"),

Set(ProfilesViewed, ProfilesClicked, EngagementsPrivate).asJava)

// This derived label is the union of all binary features above

val IS\_PROFILE\_CLICKED\_AND\_PROFILE\_ENGAGED = new Binary(

name("recap.engagement.is\_profile\_clicked\_and\_profile\_engaged"),

Set(ProfilesViewed, ProfilesClicked, EngagementsPrivate, EngagementsPublic).asJava)

// define bad profile click by considering following engagements (user report, tweet report, mute, block, etc) at profile page

val IS\_PROFILE\_CLICKED\_AND\_PROFILE\_USER\_REPORT\_CLICK = new Binary(

name("recap.engagement.is\_profile\_clicked\_and\_profile\_user\_report\_click"),

Set(ProfilesViewed, ProfilesClicked, EngagementsPrivate).asJava)

val IS\_PROFILE\_CLICKED\_AND\_PROFILE\_TWEET\_REPORT\_CLICK = new Binary(

name("recap.engagement.is\_profile\_clicked\_and\_profile\_tweet\_report\_click"),

Set(ProfilesViewed, ProfilesClicked, EngagementsPrivate).asJava)

val IS\_PROFILE\_CLICKED\_AND\_PROFILE\_MUTE = new Binary(

name("recap.engagement.is\_profile\_clicked\_and\_profile\_mute"),

Set(ProfilesViewed, ProfilesClicked, EngagementsPrivate).asJava)

val IS\_PROFILE\_CLICKED\_AND\_PROFILE\_BLOCK = new Binary(

name("recap.engagement.is\_profile\_clicked\_and\_profile\_block"),

Set(ProfilesViewed, ProfilesClicked, EngagementsPrivate).asJava)

// This derived label is the union of bad profile click engagements and existing negative feedback

val IS\_NEGATIVE\_FEEDBACK\_V2 = new Binary(

name("recap.engagement.is\_negative\_feedback\_v2"),

Set(ProfilesViewed, ProfilesClicked, EngagementsPrivate).asJava)

val IS\_STRONG\_NEGATIVE\_FEEDBACK = new Binary(

name("recap.engagement.is\_strong\_negative\_feedback"),

Set(ProfilesViewed, ProfilesClicked, EngagementsPrivate).asJava)

val IS\_WEAK\_NEGATIVE\_FEEDBACK = new Binary(

name("recap.engagement.is\_weak\_negative\_feedback"),

Set(ProfilesViewed, ProfilesClicked, EngagementsPrivate).asJava)

// engagement for following user from any surface area

val IS\_FOLLOWED\_FROM\_ANY\_SURFACE\_AREA = new Binary(

"recap.engagement.is\_followed\_from\_any\_surface\_area",

Set(EngagementsPublic, EngagementsPrivate).asJava)

// Reply downvote engagements

val IS\_REPLY\_DOWNVOTED =

new Binary(name("recap.engagement.is\_reply\_downvoted"), Set(EngagementsPrivate).asJava)

val IS\_REPLY\_DOWNVOTE\_REMOVED =

new Binary(name("recap.engagement.is\_reply\_downvote\_removed"), Set(EngagementsPrivate).asJava)

// Other engagements

val IS\_GOOD\_OPEN\_LINK = new Binary(

name("recap.engagement.is\_good\_open\_link"),

Set(EngagementsPrivate, LinksClickedOn).asJava)

val IS\_ENGAGED = new Binary(

name("recap.engagement.any"),

Set(EngagementsPrivate, EngagementsPublic).asJava

) // Deprecated - to be removed shortly

val IS\_EARLYBIRD\_UNIFIED\_ENGAGEMENT = new Binary(

name("recap.engagement.is\_unified\_engagement"),

Set(EngagementsPrivate, EngagementsPublic).asJava

) // A subset of IS\_ENGAGED specifically intended for use in earlybird models

// features from ThriftTweetFeatures

val PREV\_USER\_TWEET\_ENGAGEMENT = new Continuous(

name("recap.tweetfeature.prev\_user\_tweet\_enagagement"),

Set(EngagementScore, EngagementsPrivate, EngagementsPublic).asJava)

val IS\_SENSITIVE = new Binary(name("recap.tweetfeature.is\_sensitive"))

val HAS\_MULTIPLE\_MEDIA = new Binary(

name("recap.tweetfeature.has\_multiple\_media"),

Set(PublicTweetEntitiesAndMetadata, PrivateTweetEntitiesAndMetadata).asJava)

val IS\_AUTHOR\_PROFILE\_EGG = new Binary(name("recap.tweetfeature.is\_author\_profile\_egg"))

val IS\_AUTHOR\_NEW =

new Binary(name("recap.tweetfeature.is\_author\_new"), Set(UserState, UserType).asJava)

val NUM\_MENTIONS = new Continuous(

name("recap.tweetfeature.num\_mentions"),

Set(CountOfPrivateTweetEntitiesAndMetadata, CountOfPublicTweetEntitiesAndMetadata).asJava)

val HAS\_MENTION = new Binary(name("recap.tweetfeature.has\_mention"), Set(UserVisibleFlag).asJava)

val NUM\_HASHTAGS = new Continuous(

name("recap.tweetfeature.num\_hashtags"),

Set(CountOfPrivateTweetEntitiesAndMetadata, CountOfPublicTweetEntitiesAndMetadata).asJava)

val HAS\_HASHTAG = new Binary(

name("recap.tweetfeature.has\_hashtag"),

Set(PublicTweetEntitiesAndMetadata, PrivateTweetEntitiesAndMetadata).asJava)

val LINK\_LANGUAGE = new Continuous(

name("recap.tweetfeature.link\_language"),

Set(ProvidedLanguage, InferredLanguage).asJava)

val IS\_AUTHOR\_NSFW =

new Binary(name("recap.tweetfeature.is\_author\_nsfw"), Set(UserSafetyLabels, UserType).asJava)

val IS\_AUTHOR\_SPAM =

new Binary(name("recap.tweetfeature.is\_author\_spam"), Set(UserSafetyLabels, UserType).asJava)

val IS\_AUTHOR\_BOT =

new Binary(name("recap.tweetfeature.is\_author\_bot"), Set(UserSafetyLabels, UserType).asJava)

val SIGNATURE =

new Discrete(name("recap.tweetfeature.signature"), Set(DigitalSignatureNonrepudiation).asJava)

val LANGUAGE = new Discrete(

name("recap.tweetfeature.language"),

Set(ProvidedLanguage, InferredLanguage).asJava)

val FROM\_INACTIVE\_USER =

new Binary(name("recap.tweetfeature.from\_inactive\_user"), Set(UserActiveFlag).asJava)

val PROBABLY\_FROM\_FOLLOWED\_AUTHOR = new Binary(name("recap.v3.tweetfeature.probably\_from\_follow"))

val FROM\_MUTUAL\_FOLLOW = new Binary(name("recap.tweetfeature.from\_mutual\_follow"))

val USER\_REP = new Continuous(name("recap.tweetfeature.user\_rep"))

val FROM\_VERIFIED\_ACCOUNT =

new Binary(name("recap.tweetfeature.from\_verified\_account"), Set(UserVerifiedFlag).asJava)

val IS\_BUSINESS\_SCORE = new Continuous(name("recap.tweetfeature.is\_business\_score"))

val HAS\_CONSUMER\_VIDEO = new Binary(

name("recap.tweetfeature.has\_consumer\_video"),

Set(PublicTweetEntitiesAndMetadata, PrivateTweetEntitiesAndMetadata).asJava)

val HAS\_PRO\_VIDEO = new Binary(

name("recap.tweetfeature.has\_pro\_video"),

Set(PublicTweetEntitiesAndMetadata, PrivateTweetEntitiesAndMetadata).asJava)

val HAS\_VINE = new Binary(

name("recap.tweetfeature.has\_vine"),

Set(PublicTweetEntitiesAndMetadata, PrivateTweetEntitiesAndMetadata).asJava)

val HAS\_PERISCOPE = new Binary(

name("recap.tweetfeature.has\_periscope"),

Set(PublicTweetEntitiesAndMetadata, PrivateTweetEntitiesAndMetadata).asJava)

val HAS\_NATIVE\_VIDEO = new Binary(

name("recap.tweetfeature.has\_native\_video"),

Set(PublicTweetEntitiesAndMetadata, PrivateTweetEntitiesAndMetadata).asJava)

val HAS\_NATIVE\_IMAGE = new Binary(

name("recap.tweetfeature.has\_native\_image"),

Set(PublicTweetEntitiesAndMetadata, PrivateTweetEntitiesAndMetadata).asJava)

val HAS\_CARD = new Binary(

name("recap.tweetfeature.has\_card"),

Set(PublicTweetEntitiesAndMetadata, PrivateTweetEntitiesAndMetadata).asJava)

val HAS\_IMAGE = new Binary(

name("recap.tweetfeature.has\_image"),

Set(PublicTweetEntitiesAndMetadata, PrivateTweetEntitiesAndMetadata).asJava)

val HAS\_NEWS = new Binary(

name("recap.tweetfeature.has\_news"),

Set(PublicTweetEntitiesAndMetadata, PrivateTweetEntitiesAndMetadata).asJava)

val HAS\_VIDEO = new Binary(

name("recap.tweetfeature.has\_video"),

Set(PublicTweetEntitiesAndMetadata, PrivateTweetEntitiesAndMetadata).asJava)

val HAS\_VISIBLE\_LINK = new Binary(

name("recap.tweetfeature.has\_visible\_link"),

Set(UrlFoundFlag, PublicTweetEntitiesAndMetadata, PrivateTweetEntitiesAndMetadata).asJava)

val LINK\_COUNT = new Continuous(

name("recap.tweetfeature.link\_count"),

Set(CountOfPrivateTweetEntitiesAndMetadata, CountOfPublicTweetEntitiesAndMetadata).asJava)

val HAS\_LINK = new Binary(

name("recap.tweetfeature.has\_link"),

Set(UrlFoundFlag, PublicTweetEntitiesAndMetadata, PrivateTweetEntitiesAndMetadata).asJava)

val IS\_OFFENSIVE = new Binary(name("recap.tweetfeature.is\_offensive"))

val HAS\_TREND = new Binary(

name("recap.tweetfeature.has\_trend"),

Set(PublicTweetEntitiesAndMetadata, PrivateTweetEntitiesAndMetadata).asJava)

val HAS\_MULTIPLE\_HASHTAGS\_OR\_TRENDS = new Binary(

name("recap.tweetfeature.has\_multiple\_hashtag\_or\_trend"),

Set(PublicTweetEntitiesAndMetadata, PrivateTweetEntitiesAndMetadata).asJava)

val URL\_DOMAINS = new SparseBinary(

name("recap.tweetfeature.url\_domains"),

Set(PublicTweetEntitiesAndMetadata, PrivateTweetEntitiesAndMetadata).asJava)

val CONTAINS\_MEDIA = new Binary(

name("recap.tweetfeature.contains\_media"),

Set(PublicTweetEntitiesAndMetadata, PrivateTweetEntitiesAndMetadata).asJava)

val RETWEET\_SEARCHER = new Binary(name("recap.tweetfeature.retweet\_searcher"))

val REPLY\_SEARCHER = new Binary(name("recap.tweetfeature.reply\_searcher"))

val MENTION\_SEARCHER =

new Binary(name("recap.tweetfeature.mention\_searcher"), Set(UserVisibleFlag).asJava)

val REPLY\_OTHER =

new Binary(name("recap.tweetfeature.reply\_other"), Set(PublicReplies, PrivateReplies).asJava)

val RETWEET\_OTHER = new Binary(

name("recap.tweetfeature.retweet\_other"),

Set(PublicRetweets, PrivateRetweets).asJava)

val IS\_REPLY =

new Binary(name("recap.tweetfeature.is\_reply"), Set(PublicReplies, PrivateReplies).asJava)

val IS\_RETWEET =

new Binary(name("recap.tweetfeature.is\_retweet"), Set(PublicRetweets, PrivateRetweets).asJava)

val IS\_EXTENDED\_REPLY = new Binary(

name("recap.tweetfeature.is\_extended\_reply"),

Set(PublicReplies, PrivateReplies).asJava)

val MATCH\_UI\_LANG = new Binary(

name("recap.tweetfeature.match\_ui\_lang"),

Set(ProvidedLanguage, InferredLanguage).asJava)

val MATCH\_SEARCHER\_MAIN\_LANG = new Binary(

name("recap.tweetfeature.match\_searcher\_main\_lang"),

Set(ProvidedLanguage, InferredLanguage).asJava)

val MATCH\_SEARCHER\_LANGS = new Binary(

name("recap.tweetfeature.match\_searcher\_langs"),

Set(ProvidedLanguage, InferredLanguage).asJava)

val BIDIRECTIONAL\_REPLY\_COUNT = new Continuous(

name("recap.tweetfeature.bidirectional\_reply\_count"),

Set(CountOfPrivateReplies, CountOfPublicReplies).asJava)

val UNIDIRECTIONAL\_REPLY\_COUNT = new Continuous(

name("recap.tweetfeature.unidirectional\_reply\_count"),

Set(CountOfPrivateReplies, CountOfPublicReplies).asJava)

val BIDIRECTIONAL\_RETWEET\_COUNT = new Continuous(

name("recap.tweetfeature.bidirectional\_retweet\_count"),

Set(CountOfPrivateRetweets, CountOfPublicRetweets).asJava)

val UNIDIRECTIONAL\_RETWEET\_COUNT = new Continuous(

name("recap.tweetfeature.unidirectional\_retweet\_count"),

Set(CountOfPrivateRetweets, CountOfPublicRetweets).asJava)

val BIDIRECTIONAL\_FAV\_COUNT = new Continuous(

name("recap.tweetfeature.bidirectional\_fav\_count"),

Set(CountOfPrivateLikes, CountOfPublicLikes).asJava)

val UNIDIRECTIONAL\_FAV\_COUNT = new Continuous(

name("recap.tweetfeature.unidirectiona\_fav\_count"),

Set(CountOfPrivateLikes, CountOfPublicLikes).asJava)

val CONVERSATIONAL\_COUNT = new Continuous(

name("recap.tweetfeature.conversational\_count"),

Set(CountOfPrivateTweets, CountOfPublicTweets).asJava)

// tweet impressions on an embedded tweet

val EMBEDS\_IMPRESSION\_COUNT = new Continuous(

name("recap.tweetfeature.embeds\_impression\_count"),

Set(CountOfImpression).asJava)

// number of URLs that embed the tweet

val EMBEDS\_URL\_COUNT = new Continuous(

name("recap.tweetfeature.embeds\_url\_count"),

Set(CountOfPrivateTweetEntitiesAndMetadata, CountOfPublicTweetEntitiesAndMetadata).asJava)

// currently only counts views on Snappy and Amplify pro videos. Counts for other videos forthcoming

val VIDEO\_VIEW\_COUNT = new Continuous(

name("recap.tweetfeature.video\_view\_count"),

Set(

CountOfTweetEntitiesClicked,

CountOfPrivateTweetEntitiesAndMetadata,

CountOfPublicTweetEntitiesAndMetadata,

EngagementsPrivate,

EngagementsPublic).asJava

)

val TWEET\_COUNT\_FROM\_USER\_IN\_SNAPSHOT = new Continuous(

name("recap.tweetfeature.tweet\_count\_from\_user\_in\_snapshot"),

Set(CountOfPrivateTweets, CountOfPublicTweets).asJava)

val NORMALIZED\_PARUS\_SCORE =

new Continuous("recap.tweetfeature.normalized\_parus\_score", Set(EngagementScore).asJava)

val PARUS\_SCORE = new Continuous("recap.tweetfeature.parus\_score", Set(EngagementScore).asJava)

val REAL\_GRAPH\_WEIGHT =

new Continuous("recap.tweetfeature.real\_graph\_weight", Set(UsersRealGraphScore).asJava)

val SARUS\_GRAPH\_WEIGHT = new Continuous("recap.tweetfeature.sarus\_graph\_weight")

val TOPIC\_SIM\_SEARCHER\_INTERSTED\_IN\_AUTHOR\_KNOWN\_FOR = new Continuous(

"recap.tweetfeature.topic\_sim\_searcher\_interested\_in\_author\_known\_for")

val TOPIC\_SIM\_SEARCHER\_AUTHOR\_BOTH\_INTERESTED\_IN = new Continuous(

"recap.tweetfeature.topic\_sim\_searcher\_author\_both\_interested\_in")

val TOPIC\_SIM\_SEARCHER\_AUTHOR\_BOTH\_KNOWN\_FOR = new Continuous(

"recap.tweetfeature.topic\_sim\_searcher\_author\_both\_known\_for")

val TOPIC\_SIM\_SEARCHER\_INTERESTED\_IN\_TWEET = new Continuous(

"recap.tweetfeature.topic\_sim\_searcher\_interested\_in\_tweet")

val IS\_RETWEETER\_PROFILE\_EGG =

new Binary(name("recap.v2.tweetfeature.is\_retweeter\_profile\_egg"), Set(UserType).asJava)

val IS\_RETWEETER\_NEW =

new Binary(name("recap.v2.tweetfeature.is\_retweeter\_new"), Set(UserType, UserState).asJava)

val IS\_RETWEETER\_BOT =

new Binary(

name("recap.v2.tweetfeature.is\_retweeter\_bot"),

Set(UserType, UserSafetyLabels).asJava)

val IS\_RETWEETER\_NSFW =

new Binary(

name("recap.v2.tweetfeature.is\_retweeter\_nsfw"),

Set(UserType, UserSafetyLabels).asJava)

val IS\_RETWEETER\_SPAM =

new Binary(

name("recap.v2.tweetfeature.is\_retweeter\_spam"),

Set(UserType, UserSafetyLabels).asJava)

val RETWEET\_OF\_MUTUAL\_FOLLOW = new Binary(

name("recap.v2.tweetfeature.retweet\_of\_mutual\_follow"),

Set(PublicRetweets, PrivateRetweets).asJava)

val SOURCE\_AUTHOR\_REP = new Continuous(name("recap.v2.tweetfeature.source\_author\_rep"))

val IS\_RETWEET\_OF\_REPLY = new Binary(

name("recap.v2.tweetfeature.is\_retweet\_of\_reply"),

Set(PublicRetweets, PrivateRetweets).asJava)

val RETWEET\_DIRECTED\_AT\_USER\_IN\_FIRST\_DEGREE = new Binary(

name("recap.v2.tweetfeature.is\_retweet\_directed\_at\_user\_in\_first\_degree"),

Set(PublicRetweets, PrivateRetweets, Follow).asJava)

val MENTIONED\_SCREEN\_NAMES = new SparseBinary(

"entities.users.mentioned\_screen\_names",

Set(DisplayName, UserVisibleFlag).asJava)

val MENTIONED\_SCREEN\_NAME = new Text(

"entities.users.mentioned\_screen\_names.member",

Set(DisplayName, UserVisibleFlag).asJava)

val HASHTAGS = new SparseBinary(

"entities.hashtags",

Set(PublicTweetEntitiesAndMetadata, PrivateTweetEntitiesAndMetadata).asJava)

val URL\_SLUGS = new SparseBinary(name("recap.linkfeature.url\_slugs"), Set(UrlFoundFlag).asJava)

// features from ThriftSearchResultMetadata

val REPLY\_COUNT = new Continuous(

name("recap.searchfeature.reply\_count"),

Set(CountOfPrivateReplies, CountOfPublicReplies).asJava)

val RETWEET\_COUNT = new Continuous(

name("recap.searchfeature.retweet\_count"),

Set(CountOfPrivateRetweets, CountOfPublicRetweets).asJava)

val FAV\_COUNT = new Continuous(

name("recap.searchfeature.fav\_count"),

Set(CountOfPrivateLikes, CountOfPublicLikes).asJava)

val BLENDER\_SCORE = new Continuous(name("recap.searchfeature.blender\_score"))

val TEXT\_SCORE = new Continuous(name("recap.searchfeature.text\_score"))

// features related to content source

val SOURCE\_TYPE = new Discrete(name("recap.source.type"))

// features from addressbook

// the author is in the user's email addressbook

val USER\_TO\_AUTHOR\_EMAIL\_REACHABLE =

new Binary(name("recap.addressbook.user\_to\_author\_email\_reachable"), Set(AddressBook).asJava)

// the author is in the user's phone addressbook

val USER\_TO\_AUTHOR\_PHONE\_REACHABLE =

new Binary(name("recap.addressbook.user\_to\_author\_phone\_reachable"), Set(AddressBook).asJava)

// the user is in the author's email addressbook

val AUTHOR\_TO\_USER\_EMAIL\_REACHABLE =

new Binary(name("recap.addressbook.author\_to\_user\_email\_reachable"), Set(AddressBook).asJava)

// the user is in the user's phone addressbook

val AUTHOR\_TO\_USER\_PHONE\_REACHABLE =

new Binary(name("recap.addressbook.author\_to\_user\_phone\_reachable"), Set(AddressBook).asJava)

// predicted engagement (these features are used by prediction service to return the predicted engagement probability)

// these should match the names in engagement\_to\_score\_feature\_mapping

val PREDICTED\_IS\_FAVORITED =

new Continuous(name("recap.engagement\_predicted.is\_favorited"), Set(EngagementScore).asJava)

val PREDICTED\_IS\_RETWEETED =

new Continuous(name("recap.engagement\_predicted.is\_retweeted"), Set(EngagementScore).asJava)

val PREDICTED\_IS\_QUOTED =

new Continuous(name("recap.engagement\_predicted.is\_quoted"), Set(EngagementScore).asJava)

val PREDICTED\_IS\_REPLIED =

new Continuous(name("recap.engagement\_predicted.is\_replied"), Set(EngagementScore).asJava)

val PREDICTED\_IS\_GOOD\_OPEN\_LINK = new Continuous(

name("recap.engagement\_predicted.is\_good\_open\_link"),

Set(EngagementScore).asJava)

val PREDICTED\_IS\_PROFILE\_CLICKED = new Continuous(

name("recap.engagement\_predicted.is\_profile\_clicked"),

Set(EngagementScore).asJava)

val PREDICTED\_IS\_PROFILE\_CLICKED\_AND\_PROFILE\_ENGAGED = new Continuous(

name("recap.engagement\_predicted.is\_profile\_clicked\_and\_profile\_engaged"),

Set(EngagementScore).asJava)

val PREDICTED\_IS\_CLICKED =

new Continuous(name("recap.engagement\_predicted.is\_clicked"), Set(EngagementScore).asJava)

val PREDICTED\_IS\_PHOTO\_EXPANDED = new Continuous(

name("recap.engagement\_predicted.is\_photo\_expanded"),

Set(EngagementScore).asJava)

val PREDICTED\_IS\_DONT\_LIKE =

new Continuous(name("recap.engagement\_predicted.is\_dont\_like"), Set(EngagementScore).asJava)

val PREDICTED\_IS\_VIDEO\_PLAYBACK\_50 = new Continuous(

name("recap.engagement\_predicted.is\_video\_playback\_50"),

Set(EngagementScore).asJava)

val PREDICTED\_IS\_VIDEO\_QUALITY\_VIEWED = new Continuous(

name("recap.engagement\_predicted.is\_video\_quality\_viewed"),

Set(EngagementScore).asJava)

val PREDICTED\_IS\_BOOKMARKED =

new Continuous(name("recap.engagement\_predicted.is\_bookmarked"), Set(EngagementScore).asJava)

val PREDICTED\_IS\_SHARED =

new Continuous(name("recap.engagement\_predicted.is\_shared"), Set(EngagementScore).asJava)

val PREDICTED\_IS\_SHARE\_MENU\_CLICKED =

new Continuous(

name("recap.engagement\_predicted.is\_share\_menu\_clicked"),

Set(EngagementScore).asJava)

val PREDICTED\_IS\_PROFILE\_DWELLED\_20\_SEC = new Continuous(

name("recap.engagement\_predicted.is\_profile\_dwelled\_20\_sec"),

Set(EngagementScore).asJava)

val PREDICTED\_IS\_FULLSCREEN\_VIDEO\_DWELLED\_5\_SEC = new Continuous(

name("recap.engagement\_predicted.is\_fullscreen\_video\_dwelled\_5\_sec"),

Set(EngagementScore).asJava)

val PREDICTED\_IS\_FULLSCREEN\_VIDEO\_DWELLED\_10\_SEC = new Continuous(

name("recap.engagement\_predicted.is\_fullscreen\_video\_dwelled\_10\_sec"),

Set(EngagementScore).asJava)

val PREDICTED\_IS\_FULLSCREEN\_VIDEO\_DWELLED\_20\_SEC = new Continuous(

name("recap.engagement\_predicted.is\_fullscreen\_video\_dwelled\_20\_sec"),

Set(EngagementScore).asJava)

val PREDICTED\_IS\_FULLSCREEN\_VIDEO\_DWELLED\_30\_SEC = new Continuous(

name("recap.engagement\_predicted.is\_fullscreen\_video\_dwelled\_30\_sec"),

Set(EngagementScore).asJava)

val PREDICTED\_IS\_UNIFIED\_ENGAGEMENT = new Continuous(

name("recap.engagement\_predicted.is\_unified\_engagement"),

Set(EngagementScore).asJava)

val PREDICTED\_IS\_COMPOSE\_TRIGGERED = new Continuous(

name("recap.engagement\_predicted.is\_compose\_triggered"),

Set(EngagementScore).asJava)

val PREDICTED\_IS\_REPLIED\_REPLY\_IMPRESSED\_BY\_AUTHOR = new Continuous(

name("recap.engagement\_predicted.is\_replied\_reply\_impressed\_by\_author"),

Set(EngagementScore).asJava)

val PREDICTED\_IS\_REPLIED\_REPLY\_ENGAGED\_BY\_AUTHOR = new Continuous(

name("recap.engagement\_predicted.is\_replied\_reply\_engaged\_by\_author"),

Set(EngagementScore).asJava)

val PREDICTED\_IS\_GOOD\_CLICKED\_V1 = new Continuous(

name("recap.engagement\_predicted.is\_good\_clicked\_convo\_desc\_favorited\_or\_replied"),

Set(EngagementScore).asJava)

val PREDICTED\_IS\_GOOD\_CLICKED\_V2 = new Continuous(

name("recap.engagement\_predicted.is\_good\_clicked\_convo\_desc\_v2"),

Set(EngagementScore).asJava)

val PREDICTED\_IS\_TWEET\_DETAIL\_DWELLED\_8\_SEC = new Continuous(

name("recap.engagement\_predicted.is\_tweet\_detail\_dwelled\_8\_sec"),

Set(EngagementScore).asJava)

val PREDICTED\_IS\_TWEET\_DETAIL\_DWELLED\_15\_SEC = new Continuous(

name("recap.engagement\_predicted.is\_tweet\_detail\_dwelled\_15\_sec"),

Set(EngagementScore).asJava)

val PREDICTED\_IS\_TWEET\_DETAIL\_DWELLED\_25\_SEC = new Continuous(

name("recap.engagement\_predicted.is\_tweet\_detail\_dwelled\_25\_sec"),

Set(EngagementScore).asJava)

val PREDICTED\_IS\_TWEET\_DETAIL\_DWELLED\_30\_SEC = new Continuous(

name("recap.engagement\_predicted.is\_tweet\_detail\_dwelled\_30\_sec"),

Set(EngagementScore).asJava)

val PREDICTED\_IS\_FAVORITED\_FAV\_ENGAGED\_BY\_AUTHOR = new Continuous(

name("recap.engagement\_predicted.is\_favorited\_fav\_engaged\_by\_author"),

Set(EngagementScore).asJava)

val PREDICTED\_IS\_GOOD\_CLICKED\_WITH\_DWELL\_SUM\_GTE\_60S = new Continuous(

name(

"recap.engagement\_predicted.is\_good\_clicked\_convo\_desc\_favorited\_or\_replied\_or\_dwell\_sum\_gte\_60\_secs"),

Set(EngagementScore).asJava)

val PREDICTED\_IS\_DWELLED\_IN\_BOUNDS\_V1 = new Continuous(

name("recap.engagement\_predicted.is\_dwelled\_in\_bounds\_v1"),

Set(EngagementScore).asJava)

val PREDICTED\_DWELL\_NORMALIZED\_OVERALL = new Continuous(

name("recap.engagement\_predicted.dwell\_normalized\_overall"),

Set(EngagementScore).asJava)

val PREDICTED\_DWELL\_CDF =

new Continuous(name("recap.engagement\_predicted.dwell\_cdf"), Set(EngagementScore).asJava)

val PREDICTED\_DWELL\_CDF\_OVERALL = new Continuous(

name("recap.engagement\_predicted.dwell\_cdf\_overall"),

Set(EngagementScore).asJava)

val PREDICTED\_IS\_DWELLED =

new Continuous(name("recap.engagement\_predicted.is\_dwelled"), Set(EngagementScore).asJava)

val PREDICTED\_IS\_DWELLED\_1S =

new Continuous(name("recap.engagement\_predicted.is\_dwelled\_1s"), Set(EngagementScore).asJava)

val PREDICTED\_IS\_DWELLED\_2S =

new Continuous(name("recap.engagement\_predicted.is\_dwelled\_2s"), Set(EngagementScore).asJava)

val PREDICTED\_IS\_DWELLED\_3S =

new Continuous(name("recap.engagement\_predicted.is\_dwelled\_3s"), Set(EngagementScore).asJava)

val PREDICTED\_IS\_DWELLED\_4S =

new Continuous(name("recap.engagement\_predicted.is\_dwelled\_4s"), Set(EngagementScore).asJava)

val PREDICTED\_IS\_DWELLED\_5S =

new Continuous(name("recap.engagement\_predicted.is\_dwelled\_5s"), Set(EngagementScore).asJava)

val PREDICTED\_IS\_DWELLED\_6S =

new Continuous(name("recap.engagement\_predicted.is\_dwelled\_6s"), Set(EngagementScore).asJava)

val PREDICTED\_IS\_DWELLED\_7S =

new Continuous(name("recap.engagement\_predicted.is\_dwelled\_7s"), Set(EngagementScore).asJava)

val PREDICTED\_IS\_DWELLED\_8S =

new Continuous(name("recap.engagement\_predicted.is\_dwelled\_8s"), Set(EngagementScore).asJava)

val PREDICTED\_IS\_DWELLED\_9S =

new Continuous(name("recap.engagement\_predicted.is\_dwelled\_9s"), Set(EngagementScore).asJava)

val PREDICTED\_IS\_DWELLED\_10S =

new Continuous(name("recap.engagement\_predicted.is\_dwelled\_10s"), Set(EngagementScore).asJava)

val PREDICTED\_IS\_SKIPPED\_1S =

new Continuous(name("recap.engagement\_predicted.is\_skipped\_1s"), Set(EngagementScore).asJava)

val PREDICTED\_IS\_SKIPPED\_2S =

new Continuous(name("recap.engagement\_predicted.is\_skipped\_2s"), Set(EngagementScore).asJava)

val PREDICTED\_IS\_SKIPPED\_3S =

new Continuous(name("recap.engagement\_predicted.is\_skipped\_3s"), Set(EngagementScore).asJava)

val PREDICTED\_IS\_SKIPPED\_4S =

new Continuous(name("recap.engagement\_predicted.is\_skipped\_4s"), Set(EngagementScore).asJava)

val PREDICTED\_IS\_SKIPPED\_5S =

new Continuous(name("recap.engagement\_predicted.is\_skipped\_5s"), Set(EngagementScore).asJava)

val PREDICTED\_IS\_SKIPPED\_6S =

new Continuous(name("recap.engagement\_predicted.is\_skipped\_6s"), Set(EngagementScore).asJava)

val PREDICTED\_IS\_SKIPPED\_7S =

new Continuous(name("recap.engagement\_predicted.is\_skipped\_7s"), Set(EngagementScore).asJava)

val PREDICTED\_IS\_SKIPPED\_8S =

new Continuous(name("recap.engagement\_predicted.is\_skipped\_8s"), Set(EngagementScore).asJava)

val PREDICTED\_IS\_SKIPPED\_9S =

new Continuous(name("recap.engagement\_predicted.is\_skipped\_9s"), Set(EngagementScore).asJava)

val PREDICTED\_IS\_SKIPPED\_10S =

new Continuous(name("recap.engagement\_predicted.is\_skipped\_10s"), Set(EngagementScore).asJava)

val PREDICTED\_IS\_HOME\_LATEST\_VISITED = new Continuous(

name("recap.engagement\_predicted.is\_home\_latest\_visited"),

Set(EngagementScore).asJava)

val PREDICTED\_IS\_NEGATIVE\_FEEDBACK =

new Continuous(

name("recap.engagement\_predicted.is\_negative\_feedback"),

Set(EngagementScore).asJava)

val PREDICTED\_IS\_NEGATIVE\_FEEDBACK\_V2 =

new Continuous(

name("recap.engagement\_predicted.is\_negative\_feedback\_v2"),

Set(EngagementScore).asJava)

val PREDICTED\_IS\_WEAK\_NEGATIVE\_FEEDBACK =

new Continuous(

name("recap.engagement\_predicted.is\_weak\_negative\_feedback"),

Set(EngagementScore).asJava)

val PREDICTED\_IS\_STRONG\_NEGATIVE\_FEEDBACK =

new Continuous(

name("recap.engagement\_predicted.is\_strong\_negative\_feedback"),

Set(EngagementScore).asJava)

val PREDICTED\_IS\_REPORT\_TWEET\_CLICKED =

new Continuous(

name("recap.engagement\_predicted.is\_report\_tweet\_clicked"),

Set(EngagementScore).asJava)

val PREDICTED\_IS\_UNFOLLOW\_TOPIC =

new Continuous(

name("recap.engagement\_predicted.is\_unfollow\_topic"),

Set(EngagementScore).asJava)

val PREDICTED\_IS\_RELEVANCE\_PROMPT\_YES\_CLICKED = new Continuous(

name("recap.engagement\_predicted.is\_relevance\_prompt\_yes\_clicked"),

Set(EngagementScore).asJava)

// engagement for following user from any surface area

val PREDICTED\_IS\_FOLLOWED\_FROM\_ANY\_SURFACE\_AREA = new Continuous(

"recap.engagement\_predicted.is\_followed\_from\_any\_surface\_area",

Set(EngagementScore).asJava)

// These are global engagement counts for the Tweets.

val FAV\_COUNT\_V2 = new Continuous(

name("recap.earlybird.fav\_count\_v2"),

Set(CountOfPrivateLikes, CountOfPublicLikes).asJava)

val RETWEET\_COUNT\_V2 = new Continuous(

name("recap.earlybird.retweet\_count\_v2"),

Set(CountOfPrivateRetweets, CountOfPublicRetweets).asJava)

val REPLY\_COUNT\_V2 = new Continuous(

name("recap.earlybird.reply\_count\_v2"),

Set(CountOfPrivateReplies, CountOfPublicReplies).asJava)

val HAS\_US\_POLITICAL\_ANNOTATION = new Binary(

name("recap.has\_us\_political\_annotation"),

Set(SemanticcoreClassification).asJava

)

val HAS\_US\_POLITICAL\_ALL\_GROUPS\_ANNOTATION = new Binary(

name("recap.has\_us\_political\_all\_groups\_annotation"),

Set(SemanticcoreClassification).asJava

)

val HAS\_US\_POLITICAL\_ANNOTATION\_HIGH\_RECALL = new Binary(

name("recap.has\_us\_political\_annotation\_high\_recall"),

Set(SemanticcoreClassification).asJava

)

val HAS\_US\_POLITICAL\_ANNOTATION\_HIGH\_RECALL\_V2 = new Binary(

name("recap.has\_us\_political\_annotation\_high\_recall\_v2"),

Set(SemanticcoreClassification).asJava

)

val HAS\_US\_POLITICAL\_ANNOTATION\_HIGH\_PRECISION\_V0 = new Binary(

name("recap.has\_us\_political\_annotation\_high\_precision\_v0"),

Set(SemanticcoreClassification).asJava

)

val HAS\_US\_POLITICAL\_ANNOTATION\_BALANCED\_PRECISION\_RECALL\_V0 = new Binary(

name("recap.has\_us\_political\_annotation\_balanced\_precision\_recall\_v0"),

Set(SemanticcoreClassification).asJava

)

val HAS\_US\_POLITICAL\_ANNOTATION\_HIGH\_RECALL\_V3 = new Binary(

name("recap.has\_us\_political\_annotation\_high\_recall\_v3"),

Set(SemanticcoreClassification).asJava

)

val HAS\_US\_POLITICAL\_ANNOTATION\_HIGH\_PRECISION\_V3 = new Binary(

name("recap.has\_us\_political\_annotation\_high\_precision\_v3"),

Set(SemanticcoreClassification).asJava

)

val HAS\_US\_POLITICAL\_ANNOTATION\_BALANCED\_V3 = new Binary(

name("recap.has\_us\_political\_annotation\_balanced\_v3"),

Set(SemanticcoreClassification).asJava

)

}