package com.twitter.timelines.prediction.features.itl

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 scala.collection.JavaConverters.\_

object ITLFeatures {

// engagement

val IS\_RETWEETED =

new Binary("itl.engagement.is\_retweeted", Set(PublicRetweets, PrivateRetweets).asJava)

val IS\_FAVORITED =

new Binary("itl.engagement.is\_favorited", Set(PublicLikes, PrivateLikes).asJava)

val IS\_REPLIED =

new Binary("itl.engagement.is\_replied", Set(PublicReplies, PrivateReplies).asJava)

// v1: post click engagements: fav, reply

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

"itl.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(

"itl.engagement.is\_good\_clicked\_convo\_desc\_v2",

Set(TweetsClicked, EngagementsPrivate).asJava)

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

"itl.engagement.is\_good\_clicked\_convo\_desc\_favorited",

Set(PublicLikes, PrivateLikes).asJava)

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

"itl.engagement.is\_good\_clicked\_convo\_desc\_replied",

Set(PublicReplies, PrivateReplies).asJava)

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

"itl.engagement.is\_good\_clicked\_convo\_desc\_retweeted",

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

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

"itl.engagement.is\_good\_clicked\_convo\_desc\_clicked",

Set(TweetsClicked, EngagementsPrivate).asJava)

val IS\_GOOD\_CLICKED\_CONVO\_DESC\_FOLLOWED =

new Binary("itl.engagement.is\_good\_clicked\_convo\_desc\_followed", Set(EngagementsPrivate).asJava)

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

"itl.engagement.is\_good\_clicked\_convo\_desc\_share\_dm\_clicked",

Set(EngagementsPrivate).asJava)

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

"itl.engagement.is\_good\_clicked\_convo\_desc\_profile\_clicked",

Set(EngagementsPrivate).asJava)

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

"itl.engagement.is\_good\_clicked\_convo\_desc\_uam\_gt\_0",

Set(EngagementsPrivate, EngagementsPublic).asJava)

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

"itl.engagement.is\_good\_clicked\_convo\_desc\_uam\_gt\_1",

Set(EngagementsPrivate, EngagementsPublic).asJava)

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

"itl.engagement.is\_good\_clicked\_convo\_desc\_uam\_gt\_2",

Set(EngagementsPrivate, EngagementsPublic).asJava)

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

"itl.engagement.is\_good\_clicked\_convo\_desc\_uam\_gt\_3",

Set(EngagementsPrivate, EngagementsPublic).asJava)

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

"itl.engagement.is\_tweet\_detail\_dwelled",

Set(TweetsClicked, EngagementsPrivate).asJava)

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

"itl.engagement.is\_tweet\_detail\_dwelled\_8\_sec",

Set(TweetsClicked, EngagementsPrivate).asJava)

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

"itl.engagement.is\_tweet\_detail\_dwelled\_15\_sec",

Set(TweetsClicked, EngagementsPrivate).asJava)

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

"itl.engagement.is\_tweet\_detail\_dwelled\_25\_sec",

Set(TweetsClicked, EngagementsPrivate).asJava)

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

"itl.engagement.is\_tweet\_detail\_dwelled\_30\_sec",

Set(TweetsClicked, EngagementsPrivate).asJava)

val IS\_PROFILE\_DWELLED = new Binary(

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

Set(ProfilesViewed, ProfilesClicked, EngagementsPrivate).asJava)

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

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

Set(ProfilesViewed, ProfilesClicked, EngagementsPrivate).asJava)

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

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

Set(ProfilesViewed, ProfilesClicked, EngagementsPrivate).asJava)

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

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

Set(ProfilesViewed, ProfilesClicked, EngagementsPrivate).asJava)

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

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

Set(MediaEngagementActivities, EngagementTypePrivate, EngagementsPrivate).asJava)

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

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

Set(MediaEngagementActivities, EngagementTypePrivate, EngagementsPrivate).asJava)

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

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

Set(MediaEngagementActivities, EngagementTypePrivate, EngagementsPrivate).asJava)

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

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

Set(MediaEngagementActivities, EngagementTypePrivate, EngagementsPrivate).asJava)

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

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

Set(MediaEngagementActivities, EngagementTypePrivate, EngagementsPrivate).asJava)

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

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

Set(MediaEngagementActivities, EngagementTypePrivate, EngagementsPrivate).asJava)

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

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

Set(MediaEngagementActivities, EngagementTypePrivate, EngagementsPrivate).asJava)

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

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

Set(MediaEngagementActivities, EngagementTypePrivate, EngagementsPrivate).asJava)

val IS\_QUOTED =

new Binary("itl.engagement.is\_quoted", Set(PublicRetweets, PrivateRetweets).asJava)

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

"itl.engagement.is\_retweeted\_without\_quote",

Set(PublicRetweets, PrivateRetweets).asJava)

val IS\_CLICKED = new Binary(

"itl.engagement.is\_clicked",

Set(EngagementsPrivate, TweetsClicked, LinksClickedOn).asJava)

val IS\_PROFILE\_CLICKED = new Binary(

"itl.engagement.is\_profile\_clicked",

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

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

val IS\_DWELLED\_IN\_BOUNDS\_V1 =

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

val DWELL\_NORMALIZED\_OVERALL =

new Continuous("itl.engagement.dwell\_normalized\_overall", Set(EngagementsPrivate).asJava)

val DWELL\_CDF\_OVERALL =

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

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

val IS\_DWELLED\_1S = new Binary("itl.engagement.is\_dwelled\_1s", Set(EngagementsPrivate).asJava)

val IS\_DWELLED\_2S = new Binary("itl.engagement.is\_dwelled\_2s", Set(EngagementsPrivate).asJava)

val IS\_DWELLED\_3S = new Binary("itl.engagement.is\_dwelled\_3s", Set(EngagementsPrivate).asJava)

val IS\_DWELLED\_4S = new Binary("itl.engagement.is\_dwelled\_4s", Set(EngagementsPrivate).asJava)

val IS\_DWELLED\_5S = new Binary("itl.engagement.is\_dwelled\_5s", Set(EngagementsPrivate).asJava)

val IS\_DWELLED\_6S = new Binary("itl.engagement.is\_dwelled\_6s", Set(EngagementsPrivate).asJava)

val IS\_DWELLED\_7S = new Binary("itl.engagement.is\_dwelled\_7s", Set(EngagementsPrivate).asJava)

val IS\_DWELLED\_8S = new Binary("itl.engagement.is\_dwelled\_8s", Set(EngagementsPrivate).asJava)

val IS\_DWELLED\_9S = new Binary("itl.engagement.is\_dwelled\_9s", Set(EngagementsPrivate).asJava)

val IS\_DWELLED\_10S = new Binary("itl.engagement.is\_dwelled\_10s", Set(EngagementsPrivate).asJava)

val IS\_SKIPPED\_1S = new Binary("itl.engagement.is\_skipped\_1s", Set(EngagementsPrivate).asJava)

val IS\_SKIPPED\_2S = new Binary("itl.engagement.is\_skipped\_2s", Set(EngagementsPrivate).asJava)

val IS\_SKIPPED\_3S = new Binary("itl.engagement.is\_skipped\_3s", Set(EngagementsPrivate).asJava)

val IS\_SKIPPED\_4S = new Binary("itl.engagement.is\_skipped\_4s", Set(EngagementsPrivate).asJava)

val IS\_SKIPPED\_5S = new Binary("itl.engagement.is\_skipped\_5s", Set(EngagementsPrivate).asJava)

val IS\_SKIPPED\_6S = new Binary("itl.engagement.is\_skipped\_6s", Set(EngagementsPrivate).asJava)

val IS\_SKIPPED\_7S = new Binary("itl.engagement.is\_skipped\_7s", Set(EngagementsPrivate).asJava)

val IS\_SKIPPED\_8S = new Binary("itl.engagement.is\_skipped\_8s", Set(EngagementsPrivate).asJava)

val IS\_SKIPPED\_9S = new Binary("itl.engagement.is\_skipped\_9s", Set(EngagementsPrivate).asJava)

val IS\_SKIPPED\_10S = new Binary("itl.engagement.is\_skipped\_10s", Set(EngagementsPrivate).asJava)

val IS\_FOLLOWED =

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

val IS\_IMPRESSED = new Binary("itl.engagement.is\_impressed", Set(EngagementsPrivate).asJava)

val IS\_OPEN\_LINKED =

new Binary("itl.engagement.is\_open\_linked", Set(EngagementsPrivate, LinksClickedOn).asJava)

val IS\_PHOTO\_EXPANDED = new Binary(

"itl.engagement.is\_photo\_expanded",

Set(EngagementsPrivate, EngagementsPublic).asJava)

val IS\_VIDEO\_VIEWED =

new Binary("itl.engagement.is\_video\_viewed", Set(EngagementsPrivate, EngagementsPublic).asJava)

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

"itl.engagement.is\_video\_playback\_50",

Set(EngagementsPrivate, EngagementsPublic).asJava)

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

"itl.engagement.is\_video\_quality\_viewed",

Set(EngagementsPrivate, EngagementsPublic).asJava

)

val IS\_BOOKMARKED =

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

val IS\_SHARED =

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

val IS\_SHARE\_MENU\_CLICKED =

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

// Negative engagements

val IS\_DONT\_LIKE =

new Binary("itl.engagement.is\_dont\_like", Set(EngagementsPrivate, EngagementsPublic).asJava)

val IS\_BLOCK\_CLICKED = new Binary(

"itl.engagement.is\_block\_clicked",

Set(TweetsClicked, EngagementsPrivate, EngagementsPublic).asJava)

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

"itl.engagement.is\_block\_dialog\_blocked",

Set(EngagementsPrivate, EngagementsPublic).asJava)

val IS\_MUTE\_CLICKED =

new Binary("itl.engagement.is\_mute\_clicked", Set(TweetsClicked, EngagementsPrivate).asJava)

val IS\_MUTE\_DIALOG\_MUTED =

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

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

"itl.engagement.is\_report\_tweet\_clicked",

Set(TweetsClicked, EngagementsPrivate).asJava)

val IS\_CARET\_CLICKED =

new Binary("itl.engagement.is\_caret\_clicked", Set(TweetsClicked, EngagementsPrivate).asJava)

val IS\_NOT\_ABOUT\_TOPIC =

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

val IS\_NOT\_RECENT =

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

val IS\_NOT\_RELEVANT =

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

val IS\_SEE\_FEWER =

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

val IS\_UNFOLLOW\_TOPIC =

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

val IS\_FOLLOW\_TOPIC =

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

val IS\_NOT\_INTERESTED\_IN\_TOPIC =

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

val IS\_HOME\_LATEST\_VISITED =

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

// This derived label is the logical OR of IS\_DONT\_LIKE, IS\_BLOCK\_CLICKED, IS\_MUTE\_CLICKED and IS\_REPORT\_TWEET\_CLICKED

val IS\_NEGATIVE\_FEEDBACK =

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

// Reciprocal engagements for reply forward engagement

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

"itl.engagement.is\_replied\_reply\_impressed\_by\_author",

Set(EngagementsPrivate, EngagementsPublic).asJava)

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

"itl.engagement.is\_replied\_reply\_favorited\_by\_author",

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

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

"itl.engagement.is\_replied\_reply\_quoted\_by\_author",

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

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

"itl.engagement.is\_replied\_reply\_replied\_by\_author",

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

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

"itl.engagement.is\_replied\_reply\_retweeted\_by\_author",

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

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

"itl.engagement.is\_replied\_reply\_blocked\_by\_author",

Set(EngagementsPrivate, EngagementsPublic).asJava)

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

"itl.engagement.is\_replied\_reply\_followed\_by\_author",

Set(EngagementsPrivate, EngagementsPublic).asJava)

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

"itl.engagement.is\_replied\_reply\_unfollowed\_by\_author",

Set(EngagementsPrivate, EngagementsPublic).asJava)

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

"itl.engagement.is\_replied\_reply\_muted\_by\_author",

Set(EngagementsPrivate, EngagementsPublic).asJava)

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

"itl.engagement.is\_replied\_reply\_reported\_by\_author",

Set(EngagementsPrivate, EngagementsPublic).asJava)

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

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

"itl.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(

"itl.engagement.is\_favorited\_fav\_favorited\_by\_author",

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

)

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

"itl.engagement.is\_favorited\_fav\_replied\_by\_author",

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

)

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

"itl.engagement.is\_favorited\_fav\_retweeted\_by\_author",

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

)

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

"itl.engagement.is\_favorited\_fav\_followed\_by\_author",

Set(EngagementsPrivate, EngagementsPublic).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(

"itl.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(

"itl.engagement.is\_profile\_clicked\_and\_profile\_follow",

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

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

"itl.engagement.is\_profile\_clicked\_and\_profile\_fav",

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

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

"itl.engagement.is\_profile\_clicked\_and\_profile\_reply",

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

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

"itl.engagement.is\_profile\_clicked\_and\_profile\_retweet",

Set(

ProfilesViewed,

ProfilesClicked,

EngagementsPrivate,

PrivateRetweets,

PublicRetweets).asJava)

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

"itl.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(

"itl.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(

"itl.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(

"itl.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(

"itl.engagement.is\_profile\_clicked\_and\_profile\_tweet\_report\_click",

Set(ProfilesViewed, ProfilesClicked, EngagementsPrivate).asJava)

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

"itl.engagement.is\_profile\_clicked\_and\_profile\_mute",

Set(ProfilesViewed, ProfilesClicked, EngagementsPrivate).asJava)

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

"itl.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(

"itl.engagement.is\_negative\_feedback\_v2",

Set(ProfilesViewed, ProfilesClicked, EngagementsPrivate).asJava)

// engagement for following user from any surface area

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

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

Set(EngagementsPublic, EngagementsPrivate).asJava)

// Relevance prompt tweet engagements

val IS\_RELEVANCE\_PROMPT\_YES\_CLICKED =

new Binary("itl.engagement.is\_relevance\_prompt\_yes\_clicked", Set(EngagementsPrivate).asJava)

// Reply downvote engagements

val IS\_REPLY\_DOWNVOTED =

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

val IS\_REPLY\_DOWNVOTE\_REMOVED =

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

// features from RecommendedTweet

val RECTWEET\_SCORE = new Continuous("itl.recommended\_tweet\_features.rectweet\_score")

val NUM\_FAVORITING\_USERS = new Continuous("itl.recommended\_tweet\_features.num\_favoriting\_users")

val NUM\_FOLLOWING\_USERS = new Continuous("itl.recommended\_tweet\_features.num\_following\_users")

val CONTENT\_SOURCE\_TYPE = new Discrete("itl.recommended\_tweet\_features.content\_source\_type")

val RECOS\_SCORE = new Continuous(

"itl.recommended\_tweet\_features.recos\_score",

Set(EngagementScore, UsersRealGraphScore, UsersSalsaScore).asJava)

val AUTHOR\_REALGRAPH\_SCORE = new Continuous(

"itl.recommended\_tweet\_features.realgraph\_score",

Set(UsersRealGraphScore).asJava)

val AUTHOR\_SARUS\_SCORE = new Continuous(

"itl.recommended\_tweet\_features.sarus\_score",

Set(EngagementScore, UsersSalsaScore).asJava)

val NUM\_INTERACTING\_USERS = new Continuous(

"itl.recommended\_tweet\_features.num\_interacting\_users",

Set(EngagementScore).asJava

)

val MAX\_REALGRAPH\_SCORE\_OF\_INTERACTING\_USERS = new Continuous(

"itl.recommended\_tweet\_features.max\_realgraph\_score\_of\_interacting\_users",

Set(UsersRealGraphScore, EngagementScore).asJava

)

val SUM\_REALGRAPH\_SCORE\_OF\_INTERACTING\_USERS = new Continuous(

"itl.recommended\_tweet\_features.sum\_realgraph\_score\_of\_interacting\_users",

Set(UsersRealGraphScore, EngagementScore).asJava

)

val AVG\_REALGRAPH\_SCORE\_OF\_INTERACTING\_USERS = new Continuous(

"itl.recommended\_tweet\_features.avg\_realgraph\_score\_of\_interacting\_users",

Set(UsersRealGraphScore, EngagementScore).asJava

)

val MAX\_SARUS\_SCORE\_OF\_INTERACTING\_USERS = new Continuous(

"itl.recommended\_tweet\_features.max\_sarus\_score\_of\_interacting\_users",

Set(EngagementScore, UsersSalsaScore).asJava

)

val SUM\_SARUS\_SCORE\_OF\_INTERACTING\_USERS = new Continuous(

"itl.recommended\_tweet\_features.sum\_sarus\_score\_of\_interacting\_users",

Set(EngagementScore, UsersSalsaScore).asJava

)

val AVG\_SARUS\_SCORE\_OF\_INTERACTING\_USERS = new Continuous(

"itl.recommended\_tweet\_features.avg\_sarus\_score\_of\_interacting\_users",

Set(EngagementScore, UsersSalsaScore).asJava

)

val NUM\_INTERACTING\_FOLLOWINGS = new Continuous(

"itl.recommended\_tweet\_features.num\_interacting\_followings",

Set(EngagementScore).asJava

)

// features from HydratedTweetFeatures

val REAL\_GRAPH\_WEIGHT =

new Continuous("itl.hydrated\_tweet\_features.real\_graph\_weight", Set(UsersRealGraphScore).asJava)

val SARUS\_GRAPH\_WEIGHT = new Continuous("itl.hydrated\_tweet\_features.sarus\_graph\_weight")

val FROM\_TOP\_ENGAGED\_USER = new Binary("itl.hydrated\_tweet\_features.from\_top\_engaged\_user")

val FROM\_TOP\_INFLUENCER = new Binary("itl.hydrated\_tweet\_features.from\_top\_influencer")

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

"itl.hydrated\_tweet\_features.topic\_sim\_searcher\_interested\_in\_author\_known\_for"

)

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

"itl.hydrated\_tweet\_features.topic\_sim\_searcher\_author\_both\_interested\_in"

)

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

"itl.hydrated\_tweet\_features.topic\_sim\_searcher\_author\_both\_known\_for"

)

val USER\_REP = new Continuous("itl.hydrated\_tweet\_features.user\_rep")

val NORMALIZED\_PARUS\_SCORE = new Continuous("itl.hydrated\_tweet\_features.normalized\_parus\_score")

val CONTAINS\_MEDIA = new Binary("itl.hydrated\_tweet\_features.contains\_media")

val FROM\_NEARBY = new Binary("itl.hydrated\_tweet\_features.from\_nearby")

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

"itl.hydrated\_tweet\_features.topic\_sim\_searcher\_interested\_in\_tweet"

)

val MATCHES\_UI\_LANG = new Binary(

"itl.hydrated\_tweet\_features.matches\_ui\_lang",

Set(ProvidedLanguage, InferredLanguage).asJava)

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

"itl.hydrated\_tweet\_features.matches\_searcher\_main\_lang",

Set(ProvidedLanguage, InferredLanguage).asJava

)

val MATCHES\_SEARCHER\_LANGS = new Binary(

"itl.hydrated\_tweet\_features.matches\_searcher\_langs",

Set(ProvidedLanguage, InferredLanguage).asJava)

val HAS\_CARD = new Binary(

"itl.hydrated\_tweet\_features.has\_card",

Set(PublicTweetEntitiesAndMetadata, PrivateTweetEntitiesAndMetadata).asJava)

val HAS\_IMAGE = new Binary(

"itl.hydrated\_tweet\_features.has\_image",

Set(PublicTweetEntitiesAndMetadata, PrivateTweetEntitiesAndMetadata).asJava)

val HAS\_NATIVE\_IMAGE = new Binary(

"itl.hydrated\_tweet\_features.has\_native\_image",

Set(PublicTweetEntitiesAndMetadata, PrivateTweetEntitiesAndMetadata).asJava)

val HAS\_VIDEO = new Binary("itl.hydrated\_tweet\_features.has\_video")

val HAS\_CONSUMER\_VIDEO = new Binary(

"itl.hydrated\_tweet\_features.has\_consumer\_video",

Set(PublicTweetEntitiesAndMetadata, PrivateTweetEntitiesAndMetadata).asJava)

val HAS\_PRO\_VIDEO = new Binary(

"itl.hydrated\_tweet\_features.has\_pro\_video",

Set(PublicTweetEntitiesAndMetadata, PrivateTweetEntitiesAndMetadata).asJava)

val HAS\_PERISCOPE = new Binary(

"itl.hydrated\_tweet\_features.has\_periscope",

Set(PublicTweetEntitiesAndMetadata, PrivateTweetEntitiesAndMetadata).asJava)

val HAS\_VINE = new Binary(

"itl.hydrated\_tweet\_features.has\_vine",

Set(PublicTweetEntitiesAndMetadata, PrivateTweetEntitiesAndMetadata).asJava)

val HAS\_NATIVE\_VIDEO = new Binary(

"itl.hydrated\_tweet\_features.has\_native\_video",

Set(PublicTweetEntitiesAndMetadata, PrivateTweetEntitiesAndMetadata).asJava)

val HAS\_LINK = new Binary(

"itl.hydrated\_tweet\_features.has\_link",

Set(UrlFoundFlag, PublicTweetEntitiesAndMetadata, PrivateTweetEntitiesAndMetadata).asJava)

val LINK\_COUNT = new Continuous(

"itl.hydrated\_tweet\_features.link\_count",

Set(CountOfPrivateTweetEntitiesAndMetadata, CountOfPublicTweetEntitiesAndMetadata).asJava)

val URL\_DOMAINS = new SparseBinary(

"itl.hydrated\_tweet\_features.url\_domains",

Set(PublicTweetEntitiesAndMetadata, PrivateTweetEntitiesAndMetadata).asJava)

val HAS\_VISIBLE\_LINK = new Binary(

"itl.hydrated\_tweet\_features.has\_visible\_link",

Set(UrlFoundFlag, PublicTweetEntitiesAndMetadata, PrivateTweetEntitiesAndMetadata).asJava)

val HAS\_NEWS = new Binary(

"itl.hydrated\_tweet\_features.has\_news",

Set(PublicTweetEntitiesAndMetadata, PrivateTweetEntitiesAndMetadata).asJava)

val HAS\_TREND = new Binary(

"itl.hydrated\_tweet\_features.has\_trend",

Set(PublicTweetEntitiesAndMetadata, PrivateTweetEntitiesAndMetadata).asJava)

val BLENDER\_SCORE =

new Continuous("itl.hydrated\_tweet\_features.blender\_score", Set(EngagementScore).asJava)

val PARUS\_SCORE =

new Continuous("itl.hydrated\_tweet\_features.parus\_score", Set(EngagementScore).asJava)

val TEXT\_SCORE =

new Continuous("itl.hydrated\_tweet\_features.text\_score", Set(EngagementScore).asJava)

val BIDIRECTIONAL\_REPLY\_COUNT = new Continuous(

"itl.hydrated\_tweet\_features.bidirectional\_reply\_count",

Set(CountOfPrivateReplies, CountOfPublicReplies).asJava

)

val UNIDIRECTIONAL\_REPLY\_COUNT = new Continuous(

"itl.hydrated\_tweet\_features.unidirectional\_reply\_count",

Set(CountOfPrivateReplies, CountOfPublicReplies).asJava

)

val BIDIRECTIONAL\_RETWEET\_COUNT = new Continuous(

"itl.hydrated\_tweet\_features.bidirectional\_retweet\_count",

Set(CountOfPrivateRetweets, CountOfPublicRetweets).asJava

)

val UNIDIRECTIONAL\_RETWEET\_COUNT = new Continuous(

"itl.hydrated\_tweet\_features.unidirectional\_retweet\_count",

Set(CountOfPrivateRetweets, CountOfPublicRetweets).asJava

)

val BIDIRECTIONAL\_FAV\_COUNT = new Continuous(

"itl.hydrated\_tweet\_features.bidirectional\_fav\_count",

Set(CountOfPrivateLikes, CountOfPublicLikes).asJava

)

val UNIDIRECTIONAL\_FAV\_COUNT = new Continuous(

"itl.hydrated\_tweet\_features.unidirectional\_fav\_count",

Set(CountOfPrivateLikes, CountOfPublicLikes).asJava

)

val CONVERSATION\_COUNT = new Continuous("itl.hydrated\_tweet\_features.conversation\_count")

val FAV\_COUNT = new Continuous(

"itl.hydrated\_tweet\_features.fav\_count",

Set(CountOfPrivateLikes, CountOfPublicLikes).asJava)

val REPLY\_COUNT = new Continuous(

"itl.hydrated\_tweet\_features.reply\_count",

Set(CountOfPrivateReplies, CountOfPublicReplies).asJava)

val RETWEET\_COUNT = new Continuous(

"itl.hydrated\_tweet\_features.retweet\_count",

Set(CountOfPrivateRetweets, CountOfPublicRetweets).asJava)

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

"itl.hydrated\_tweet\_features.prev\_user\_tweet\_enagagement",

Set(EngagementScore, EngagementsPrivate, EngagementsPublic).asJava

)

val IS\_SENSITIVE = new Binary("itl.hydrated\_tweet\_features.is\_sensitive")

val HAS\_MULTIPLE\_MEDIA = new Binary(

"itl.hydrated\_tweet\_features.has\_multiple\_media",

Set(PublicTweetEntitiesAndMetadata, PrivateTweetEntitiesAndMetadata).asJava)

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

"itl.hydrated\_tweet\_features.has\_multiple\_hashtag\_or\_trend",

Set(

UserVisibleFlag,

CountOfPrivateTweetEntitiesAndMetadata,

CountOfPublicTweetEntitiesAndMetadata).asJava)

val IS\_AUTHOR\_PROFILE\_EGG =

new Binary("itl.hydrated\_tweet\_features.is\_author\_profile\_egg", Set(ProfileImage).asJava)

val IS\_AUTHOR\_NEW =

new Binary("itl.hydrated\_tweet\_features.is\_author\_new", Set(UserType, UserState).asJava)

val NUM\_MENTIONS = new Continuous(

"itl.hydrated\_tweet\_features.num\_mentions",

Set(

UserVisibleFlag,

CountOfPrivateTweetEntitiesAndMetadata,

CountOfPublicTweetEntitiesAndMetadata).asJava)

val NUM\_HASHTAGS = new Continuous(

"itl.hydrated\_tweet\_features.num\_hashtags",

Set(CountOfPrivateTweetEntitiesAndMetadata, CountOfPublicTweetEntitiesAndMetadata).asJava)

val LANGUAGE = new Discrete(

"itl.hydrated\_tweet\_features.language",

Set(ProvidedLanguage, InferredLanguage).asJava)

val LINK\_LANGUAGE = new Continuous(

"itl.hydrated\_tweet\_features.link\_language",

Set(ProvidedLanguage, InferredLanguage).asJava)

val IS\_AUTHOR\_NSFW =

new Binary("itl.hydrated\_tweet\_features.is\_author\_nsfw", Set(UserType).asJava)

val IS\_AUTHOR\_SPAM =

new Binary("itl.hydrated\_tweet\_features.is\_author\_spam", Set(UserType).asJava)

val IS\_AUTHOR\_BOT = new Binary("itl.hydrated\_tweet\_features.is\_author\_bot", Set(UserType).asJava)

val IS\_OFFENSIVE = new Binary("itl.hydrated\_tweet\_features.is\_offensive")

val FROM\_VERIFIED\_ACCOUNT =

new Binary("itl.hydrated\_tweet\_features.from\_verified\_account", Set(UserVerifiedFlag).asJava)

val EMBEDS\_IMPRESSION\_COUNT = new Continuous(

"itl.hydrated\_tweet\_features.embeds\_impression\_count",

Set(CountOfImpression).asJava)

val EMBEDS\_URL\_COUNT =

new Continuous("itl.hydrated\_tweet\_features.embeds\_url\_count", Set(UrlFoundFlag).asJava)

val FAV\_COUNT\_V2 = new Continuous(

"recap.earlybird.fav\_count\_v2",

Set(CountOfPrivateLikes, CountOfPublicLikes).asJava)

val RETWEET\_COUNT\_V2 = new Continuous(

"recap.earlybird.retweet\_count\_v2",

Set(CountOfPrivateRetweets, CountOfPublicRetweets).asJava)

val REPLY\_COUNT\_V2 = new Continuous(

"recap.earlybird.reply\_count\_v2",

Set(CountOfPrivateReplies, CountOfPublicReplies).asJava)

}