Automating Data Exploration with R

Pipeline Check

Let’s put most of our generic pipeline functions together:

```
Binarize_Features <- function(data_set, features_to_ignore=c(), leave_out_one_leve
l=FALSE, max_level_count=20) {
     require(dplyr)
     text_features <- c(names(data_set[sapply(data_set, is.character)]), names(dat
a_set[sapply(data_set, is.factor)]))
     for (feature_name in setdiff(text_features, features_to_ignore)) {
          feature_vector <- as.character(data_set[,feature_name])

          # check that data has more than one level
          if (length(unique(feature_vector)) == 1)
               next

          # We set any non-data to text
          feature_vector[is.na(feature_vector)] <- 'NA'
          feature_vector[is.infinite(feature_vector)] <- 'INF'
          feature_vector[is.nan(feature_vector)] <- 'NAN'

          # only give us the top x most popular categories
          temp_vect <- data.frame(table(feature_vector)) %>% arrange(desc(Freq)) %
>% head(max_level_count)
          feature_vector <- ifelse(feature_vector %in% temp_vect$feature_vector, f
eature_vector, 'Other')

          # loop through each level of a feature and create a new column
          first_level=TRUE
          for (newcol in unique(feature_vector)) {
               if (leave_out_one_level & first_level) {
                    # avoid dummy trap and skip first level
                    first_level=FALSE
                    next
               }

               data_set[,paste0(feature_name,"_",newcol)] <- ifelse(feature_vector
==newcol,1,0)
          }
          # remove original feature
          data_set <- data_set[,setdiff(names(data_set),feature_name)]
     }
     return (data_set)
}
```

```
Get_Free_Text_Measures <- function(data_set, minimum_unique_threshold=0.9, feature
s_to_ignore=c()) {

     # look for text entries that are mostly unique
     text_features <- c(names(data_set[sapply(data_set, is.character)]), names(dat
a_set[sapply(data_set, is.factor)]))
     for (f_name in setdiff(text_features, features_to_ignore)) {
          f_vector <- as.character(data_set[,f_name])

          # treat as raw text if data over minimum_precent_unique unique
          if (length(unique(as.character(f_vector))) > (nrow(data_set) * minimum_u
nique_threshold)) {
               data_set[,paste0(f_name, '_word_count')] <- sapply(strsplit(f_vecto
r, " "), length)
               data_set[,paste0(f_name, '_character_count')]  <-  nchar(as.charact
er(f_vector))
               data_set[,paste0(f_name, '_first_word')] <- sapply(strsplit(as.char
acter(f_vector), " "), `[`, 1)
               # remove orginal field
               data_set[,f_name] <- NULL
          }
     }
     return(data_set)
}
```

```
Impute_Features <- function(data_set, features_to_ignore=c(),
                            use_mean_instead_of_0=TRUE,
                            mark_NAs=FALSE,
                            remove_zero_variance=FALSE) {

     for (feature_name in setdiff(names(data_set), features_to_ignore)) {
          print(feature_name)
          # remove any fields with zero variance
          if (remove_zero_variance) {
               if (length(unique(data_set[, feature_name]))==1) {
                    data_set[, feature_name] <- NULL
                    next
               }
          }

          if (mark_NAs) {
               # note each field that contains missing or bad data
               if (any(is.na(data_set[,feature_name]))) {
                    # create binary column before imputing
                    newName <- paste0(feature_name, '_NA')
                    data_set[,newName] <- as.integer(ifelse(is.na(data_set[,featur
e_name]),1,0)) }

               if (any(is.infinite(data_set[,feature_name]))) {
                    newName <- paste0(feature_name, '_inf')
                    data_set[,newName] <- as.integer(ifelse(is.infinite(data_set[,
feature_name]),1,0)) }
          }

          if (use_mean_instead_of_0) {
               data_set[is.infinite(data_set[,feature_name]),feature_name] <- NA
               data_set[is.na(data_set[,feature_name]),feature_name] <- mean(data_
set[,feature_name], na.rm=TRUE)

          } else {
               data_set[is.na(data_set[,feature_name]),feature_name] <- 0
               data_set[is.infinite(data_set[,feature_name]),feature_name] <- 0
          }
     }
     return(data_set)
}
```

```
Feature_Engineer_Dates <- function(data_set, remove_original_date=TRUE) {
     require(lubridate)
     data_set <- data.frame(data_set)
     date_features <- names(data_set[sapply(data_set, is.Date)])
     for (feature_name in date_features) {
          data_set[,paste0(feature_name,'_DateInt')] <- as.numeric(data_set[,featu
re_name])
          data_set[,paste0(feature_name,'_Month')] <- as.integer(format(data_set[,
feature_name], "%m"))
          data_set[,paste0(feature_name,'_ShortYear')] <- as.integer(format(data_s
et[,feature_name], "%y"))
          data_set[,paste0(feature_name,'_LongYear')] <- as.integer(format(data_se
t[,feature_name], "%Y"))
          data_set[,paste0(feature_name,'_Day')] <- as.integer(format(data_set[,fe
ature_name], "%d"))

          # week day number requires first pulling the weekday label, creating the
7 week day levels, and casting to integer
          data_set[,paste0(feature_name,'_WeekDayNumber')] <- as.factor(weekdays(d
ata_set[,feature_name]))
          levels(data_set[,paste0(feature_name,'_WeekDayNumber')]) <- list(Monday=
1, Tuesday=2, Wednesday=3, Thursday=4, Friday=5, Saturday=6, Sunday=7)
          data_set[,paste0(feature_name,'_WeekDayNumber')] <- as.integer(data_set
[,paste0(feature_name,'_WeekDayNumber')])

          data_set[,paste0(feature_name,'_IsWeekend')] <- as.numeric(grepl("Saturd
ay|Sunday", weekdays(data_set[,feature_name])))
          data_set[,paste0(feature_name,'_YearDayCount')] <- yday(data_set[,featur
e_name])
          data_set[,paste0(feature_name,'_Quarter')] <- lubridate::quarter(data_se
t[,feature_name], with_year = FALSE)
          data_set[,paste0(feature_name,'_Quarter')] <- lubridate::quarter(data_se
t[,feature_name], with_year = TRUE)
          if (remove_original_date)
               data_set[, feature_name] <- NULL
     }
     return(data_set)
}
```

```
Feature_Engineer_Integers <- function(data_set, features_to_ignore=c()) {
     require(infotheo)
     data_set <- data.frame(data_set)

     for (feature_name in setdiff(names(data_set), features_to_ignore)) {
          if (class(data_set[,feature_name])=='numeric' | class(data_set[,feature_
name])=='integer') {
               feature_vector <- data_set[,feature_name]

               if (all((feature_vector - round(feature_vector)) == 0)) {
                    # make sure we have more than 2 values excluding NAs
                    if (length(unique(data_set[,feature_name][!is.na(data_set[,fea
ture_name])])) > 2) {
                         print(feature_name)
                         data_set[,paste0(feature_name,'_IsZero')] <- ifelse(data_
set[,feature_name]==0,1,0)
                         data_set[,paste0(feature_name,'_IsPositive')] <- ifelse(d
ata_set[,feature_name]>=0,1,0)
                         # separate data into two bins
                         data_discretized <- discretize(data_set[,feature_name], d
isc='equalfreq', nbins=2)
                         data_set[,paste0(feature_name,'_2Bins')] <- data_discreti
zed$X

                         if (length(unique(data_set[,feature_name][!is.na(data_set
[,feature_name])])) > 4) {
                              # try 4 bins
                              data_discretized <- discretize(data_set[,feature_nam
e], disc='equalfreq', nbins=4)
                              data_set[,paste0(feature_name,'_4Bins')] <- data_dis
cretized$X
                         }
                    }
               }
          }
     }
     return (data_set)
}
```

```
Feature_Engineer_Numbers <- function(data_set, features_to_ignore=c()) {
     require(infotheo)
     data_set <- data.frame(data_set)
     date_features <- setdiff(names(data_set[sapply(data_set, is.numeric)]), featu
res_to_ignore)
     for (feature_name in date_features) {
          feature_vector <- data_set[,feature_name]
          if (is.integer(feature_vector) | is.numeric(feature_vector)) {
               if (any((feature_vector - round(feature_vector)) != 0)) {
                    # make sure we have more than 2 values excluding NAs
                    if (length(unique(data_set[,feature_name][!is.na(data_set[,fea
ture_name])])) > 2) {
                         print(feature_name)
                         # polynomial transformation
                         poly_vector <- poly(x=feature_vector, degree = 2)
                         data_set[,paste0(feature_name, "_poly1")] <- poly_vector
[,1]
                         data_set[,paste0(feature_name, "_poly2")] <- poly_vector
[,2]
                         # log transform
                         data_set[,paste0(feature_name, "_log")] <- log(x = featur
e_vector)
                         # exponential transform
                         data_set[,paste0(feature_name, "_exp")] <- exp(x = featur
e_vector)
                         # rounding
                         data_set[,paste0(feature_name, "_rnd")] <- round(x = feat
ure_vector, digits = 0)
                         # binning into 2 bins
                         data_discretized <- discretize(data_set[,feature_name], d
isc='equalfreq', nbins=2)
                         data_set[,paste0(feature_name,'_2Bins')] <- data_discreti
zed$X

                    }
               }
          }
     }
     return(data_set)
}
```

```
mix_dataset <- data.frame(
               id=c(1,2,3,4,5),
               gender=c('male','female','female','male','female'),
               some_date=c('2012-01-01','2013-01-01','2014-01-01','2015-01-01','20
16-01-01'),
               mood=c(0,20,20,NA,50),
               value=c(12.34, 32.2, 24.3, 83.1, 8.32),
               outcome=c(1,1,0,0,0))

library(readr)
write_csv(mix_dataset, 'mix_dataset.csv')
mix_dataset <- as.data.frame(read_csv('mix_dataset.csv'))

# automated pipeline
mix_dataset <- Get_Free_Text_Measures(data_set = mix_dataset)
mix_dataset <- Binarize_Features(data_set = mix_dataset, leave_out_one_level = FAL
SE)
```

```
## Loading required package: dplyr
##
## Attaching package: 'dplyr'
##
## The following objects are masked from 'package:stats':
##
##     filter, lag
##
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
mix_dataset <- Impute_Features(data_set = mix_dataset)
```

```
## [1] "id"
## [1] "some_date"
## [1] "mood"
## [1] "value"
## [1] "outcome"
## [1] "gender_male"
## [1] "gender_female"
```

```
mix_dataset <- Feature_Engineer_Dates(data_set = mix_dataset)
```

```
## Loading required package: lubridate
```

```
mix_dataset <- Feature_Engineer_Integers(data_set = mix_dataset)
```

```
## Loading required package: infotheo
```

```
## [1] "id"
## [1] "some_date_DateInt"
## [1] "some_date_ShortYear"
## [1] "some_date_LongYear"
## [1] "some_date_WeekDayNumber"
```

```
mix_dataset <- Feature_Engineer_Numbers(data_set = mix_dataset)
```

```
## [1] "mood"
## [1] "value"
## [1] "some_date_Quarter"
```

```
head(mix_dataset,2)
```

```
##   id mood value outcome gender_male gender_female some_date_DateInt
## 1  1    0 12.34       1           1             0             15340
## 2  2   20 32.20       1           0             1             15706
##   some_date_Month some_date_ShortYear some_date_LongYear some_date_Day
## 1               1                  12               2012             1
## 2               1                  13               2013             1
##   some_date_WeekDayNumber some_date_IsWeekend some_date_YearDayCount
## 1                       7                   1                      1
## 2                       2                   0                      1
##   some_date_Quarter id_IsZero id_IsPositive id_2Bins id_4Bins
## 1            2012.1         0             1        1        1
## 2            2013.1         0             1        1        1
##   some_date_DateInt_IsZero some_date_DateInt_IsPositive
## 1                        0                            1
## 2                        0                            1
##   some_date_DateInt_2Bins some_date_DateInt_4Bins
## 1                       1                      1
## 2                       1                      1
##   some_date_ShortYear_IsZero some_date_ShortYear_IsPositive
## 1                          0                              1
## 2                          0                              1
##   some_date_ShortYear_2Bins some_date_ShortYear_4Bins
## 1                         1                        1
## 2                         1                        1
##   some_date_LongYear_IsZero some_date_LongYear_IsPositive
## 1                         0                             1
## 2                         0                             1
##   some_date_LongYear_2Bins some_date_LongYear_4Bins
## 1                        1                       1
## 2                        1                       1
##   some_date_WeekDayNumber_IsZero some_date_WeekDayNumber_IsPositive
## 1                              0                                   1
## 2                              0                                   1
##   some_date_WeekDayNumber_2Bins some_date_WeekDayNumber_4Bins mood_poly1
## 1                             2                             4  -0.630126
## 2                             1                             1  -0.070014
##   mood_poly2 mood_log  mood_exp mood_rnd mood_2Bins  value_poly1
## 1  0.6323510     -Inf         1        0          1 -0.327724828
## 2 -0.3495951 2.995732 485165195       20          1  0.002460596
##   value_poly2 value_log    value_exp value_rnd value_2Bins
## 1   0.2483090  2.512846 2.286620e+05        12           1
## 2  -0.6629308  3.471966 9.644558e+13        32           2
##   some_date_Quarter_poly1 some_date_Quarter_poly2 some_date_Quarter_log
## 1              -0.6324555               0.5345225              7.606934
## 2              -0.3162278              -0.2672612              7.607431
##   some_date_Quarter_exp some_date_Quarter_rnd some_date_Quarter_2Bins
## 1                   Inf                  2012                      1
## 2                   Inf                  2013                      1
```

```
summary(mix_dataset)
```

```
##        id         mood           value          outcome     gender_male
##  Min.   :1   Min.   : 0.0   Min.   : 8.32   Min.   :0.0   Min.   :0.0
##  1st Qu.:2   1st Qu.:20.0   1st Qu.:12.34   1st Qu.:0.0   1st Qu.:0.0
##  Median :3   Median :20.0   Median :24.30   Median :0.0   Median :0.0
##  Mean   :3   Mean   :22.5   Mean   :32.05   Mean   :0.4   Mean   :0.4
##  3rd Qu.:4   3rd Qu.:22.5   3rd Qu.:32.20   3rd Qu.:1.0   3rd Qu.:1.0
##  Max.   :5   Max.   :50.0   Max.   :83.10   Max.   :1.0   Max.   :1.0
##  gender_female some_date_DateInt some_date_Month some_date_ShortYear
##  Min.   :0.0   Min.   :15340     Min.   :1       Min.   :12
##  1st Qu.:0.0   1st Qu.:15706     1st Qu.:1       1st Qu.:13
##  Median :1.0   Median :16071     Median :1       Median :14
##  Mean   :0.6   Mean   :16071     Mean   :1       Mean   :14
##  3rd Qu.:1.0   3rd Qu.:16436     3rd Qu.:1       3rd Qu.:15
##  Max.   :1.0   Max.   :16801     Max.   :1       Max.   :16
##  some_date_LongYear some_date_Day some_date_WeekDayNumber
##  Min.   :2012       Min.   :1     Min.   :2.0
##  1st Qu.:2013       1st Qu.:1     1st Qu.:3.0
##  Median :2014       Median :1     Median :4.0
##  Mean   :2014       Mean   :1     Mean   :4.2
##  3rd Qu.:2015       3rd Qu.:1     3rd Qu.:5.0
##  Max.   :2016       Max.   :1     Max.   :7.0
##  some_date_IsWeekend some_date_YearDayCount some_date_Quarter   id_IsZero
##  Min.   :0.0         Min.   :1              Min.   :2012      Min.   :0
##  1st Qu.:0.0         1st Qu.:1              1st Qu.:2013      1st Qu.:0
##  Median :0.0         Median :1              Median :2014      Median :0
##  Mean   :0.2         Mean   :1              Mean   :2014      Mean   :0
##  3rd Qu.:0.0         3rd Qu.:1              3rd Qu.:2015      3rd Qu.:0
##  Max.   :1.0         Max.   :1              Max.   :2016      Max.   :0
##  id_IsPositive    id_2Bins      id_4Bins   some_date_DateInt_IsZero
##  Min.   :1     Min.   :1.0   Min.   :1.0   Min.   :0
##  1st Qu.:1     1st Qu.:1.0   1st Qu.:1.0   1st Qu.:0
##  Median :1     Median :1.0   Median :3.0   Median :0
##  Mean   :1     Mean   :1.4   Mean   :2.6   Mean   :0
##  3rd Qu.:1     3rd Qu.:2.0   3rd Qu.:4.0   3rd Qu.:0
##  Max.   :1     Max.   :2.0   Max.   :4.0   Max.   :0
##  some_date_DateInt_IsPositive some_date_DateInt_2Bins
##  Min.   :1                    Min.   :1.0
##  1st Qu.:1                    1st Qu.:1.0
##  Median :1                    Median :1.0
##  Mean   :1                    Mean   :1.4
##  3rd Qu.:1                    3rd Qu.:2.0
##  Max.   :1                    Max.   :2.0
##  some_date_DateInt_4Bins some_date_ShortYear_IsZero
##  Min.   :1.0             Min.   :0
##  1st Qu.:1.0             1st Qu.:0
##  Median :3.0             Median :0
##  Mean   :2.6             Mean   :0
##  3rd Qu.:4.0             3rd Qu.:0
##  Max.   :4.0             Max.   :0
##  some_date_ShortYear_IsPositive some_date_ShortYear_2Bins
```

```
##  Min.   :1                     Min.   :1.0
##  1st Qu.:1                     1st Qu.:1.0
##  Median :1                     Median :1.0
##  Mean   :1                     Mean   :1.4
##  3rd Qu.:1                     3rd Qu.:2.0
##  Max.   :1                     Max.   :2.0
##  some_date_ShortYear_4Bins some_date_LongYear_IsZero
##  Min.   :1.0               Min.   :0
##  1st Qu.:1.0               1st Qu.:0
##  Median :3.0               Median :0
##  Mean   :2.6               Mean   :0
##  3rd Qu.:4.0               3rd Qu.:0
##  Max.   :4.0               Max.   :0
##  some_date_LongYear_IsPositive some_date_LongYear_2Bins
##  Min.   :1                     Min.   :1.0
##  1st Qu.:1                     1st Qu.:1.0
##  Median :1                     Median :1.0
##  Mean   :1                     Mean   :1.4
##  3rd Qu.:1                     3rd Qu.:2.0
##  Max.   :1                     Max.   :2.0
##  some_date_LongYear_4Bins some_date_WeekDayNumber_IsZero
##  Min.   :1.0              Min.   :0
##  1st Qu.:1.0              1st Qu.:0
##  Median :3.0              Median :0
##  Mean   :2.6              Mean   :0
##  3rd Qu.:4.0              3rd Qu.:0
##  Max.   :4.0              Max.   :0
##  some_date_WeekDayNumber_IsPositive some_date_WeekDayNumber_2Bins
##  Min.   :1                          Min.   :1.0
##  1st Qu.:1                          1st Qu.:1.0
##  Median :1                          Median :1.0
##  Mean   :1                          Mean   :1.4
##  3rd Qu.:1                          3rd Qu.:2.0
##  Max.   :1                          Max.   :2.0
##  some_date_WeekDayNumber_4Bins   mood_poly1         mood_poly2
##  Min.   :1.0                   Min.   :-0.63013   Min.   :-0.3870
##  1st Qu.:1.0                   1st Qu.:-0.07001   1st Qu.:-0.3496
##  Median :3.0                   Median :-0.07001   Median :-0.3496
##  Mean   :2.6                   Mean   : 0.00000   Mean   : 0.0000
##  3rd Qu.:4.0                   3rd Qu.: 0.00000   3rd Qu.: 0.4538
##  Max.   :4.0                   Max.   : 0.77015   Max.   : 0.6324
##     mood_log       mood_exp            mood_rnd      mood_2Bins
##  Min.   :-Inf   Min.   :1.000e+00   Min.   : 0.0   Min.   :1.0
##  1st Qu.:   3   1st Qu.:4.852e+08   1st Qu.:20.0   1st Qu.:1.0
##  Median :   3   Median :4.852e+08   Median :20.0   Median :1.0
##  Mean   :-Inf   Mean   :1.037e+21   Mean   :22.4   Mean   :1.4
##  3rd Qu.:   3   3rd Qu.:5.911e+09   3rd Qu.:22.0   3rd Qu.:2.0
##  Max.   :   4   Max.   :5.185e+21   Max.   :50.0   Max.   :2.0
##   value_poly1         value_poly2        value_log       value_exp
##  Min.   :-0.394560   Min.   :-0.6629   Min.   :2.119   Min.   :4.105e+03
##  1st Qu.:-0.327725   1st Qu.:-0.3865   1st Qu.:2.513   1st Qu.:2.287e+05
```

```
##  Median :-0.128882   Median : 0.2483   Median :3.190   Median :3.576e+10
##  Mean   : 0.000000   Mean   : 0.0000   Mean   :3.143   Mean   :2.460e+35
##  3rd Qu.: 0.002461   3rd Qu.: 0.2809   3rd Qu.:3.472   3rd Qu.:9.645e+13
##  Max.   : 0.848706   Max.   : 0.5202   Max.   :4.420   Max.   :1.230e+36
##    value_rnd     value_2Bins  some_date_Quarter_poly1
##  Min.   : 8.0   Min.   :1.0   Min.   :-0.6325
##  1st Qu.:12.0   1st Qu.:1.0   1st Qu.:-0.3162
##  Median :24.0   Median :1.0   Median : 0.0000
##  Mean   :31.8   Mean   :1.4   Mean   : 0.0000
##  3rd Qu.:32.0   3rd Qu.:2.0   3rd Qu.: 0.3162
##  Max.   :83.0   Max.   :2.0   Max.   : 0.6325
##  some_date_Quarter_poly2 some_date_Quarter_log some_date_Quarter_exp
##  Min.   :-0.5345         Min.   :7.607         Min.   :Inf
##  1st Qu.:-0.2673         1st Qu.:7.607         1st Qu.:Inf
##  Median :-0.2673         Median :7.608         Median :Inf
##  Mean   : 0.0000         Mean   :7.608         Mean   :Inf
##  3rd Qu.: 0.5345         3rd Qu.:7.608         3rd Qu.:Inf
##  Max.   : 0.5345         Max.   :7.609         Max.   :Inf
##  some_date_Quarter_rnd some_date_Quarter_2Bins
##  Min.   :2012          Min.   :1.0
##  1st Qu.:2013          1st Qu.:1.0
##  Median :2014          Median :1.0
##  Mean   :2014          Mean   :1.4
##  3rd Qu.:2015          3rd Qu.:2.0
##  Max.   :2016          Max.   :2.0
```