Understanding proposal win rates

Martijn Schroder

1/05/2019

Contents

Executive summary

In management consulting competitiveness is high. Our firm competes actively with the 'Big 4' consultancies. Better understanding of the reasons why we win and loose proposal will give us an advantage.

Our business makes decisions on our proposal management practice with the view to increase win rates. Until recently those decisions were made based on experience and perception of what works and what doesn't.

In this assignment my aim is to try machine learning approaches to gain insights into what the data tells us about the relevant features that are good predictors of win and lose rates.

Given the low number of transactions and limited cleanliness of the data, the analysis of features that should underpin decisions around proposals is relatively ambiguous. This is the main challenge to work with.

For obvious reasons the data is de identified.

Cleaning data

The data set is a raw export from the system we use to manage opportunities. A csv export was obtained with the following structure:

```r
names(proposals) # obtain the column names
```

```
##  [1] "Opportunity Name"
##  [2] "Account Name"
##  [3] "Stage"
##  [4] "Amount Currency"
##  [5] "Amount"
##  [6] "Created Date"
##  [7] "Close Date"
##  [8] "Primary Practice"
##  [9] "Business Offer"
## [10] "Sector"
## [11] "Segment"
## [12] "Proposal director"
## [13] "Proposal manager"
## [14] "Source"
## [15] "Competitive or sole sourced (compulsory)"
```

```r
str(proposals) # show structure of data
```

```
## Classes 'spec_tbl_df', 'tbl_df', 'tbl' and 'data.frame': 3555 obs. of  15 variables:
##  $ Opportunity Name                        : chr  "USY 1803 Sydney Operating Model Support" "UOW 180
##  $ Account Name                            : chr  "University of Sydney" "University of Wollongong"
##  $ Stage                                   : chr  "Client not pursuing" "Opp successful" "Opp success
##  $ Amount Currency                         : chr  "AUD" "AUD" "GBP" "AUD" ...
##  $ Amount                                  : num  300000 81735 1 99134 0 ...
##  $ Created Date                            : chr  "10/3/18" "10/3/18" "6/3/18" "4/8/16" ...
##  $ Close Date                              : chr  "1/5/18" "9/6/18" "6/3/18" "19/8/16" ...
##  $ Primary Practice                        : chr  "Org Performance and Leadership" "Org Performance
##  $ Business Offer                          : chr  "OP-Operating model design" "OP-Culture change" "S
##  $ Sector                                  : chr  NA "Education" NA "Health and Ageing" ...
##  $ Segment                                 : chr  "Institution" "Institution" "Internal" "Government
##  $ Proposal director                       : chr  "Peter Wiseman" "Megan Huisman" "Laura Gordon" "Sa
##  $ Proposal manager                        : chr  "Iris Rattley" "Kate Breheny" "Laura Gordon" "Anne
##  $ Source                                  : chr  NA NA NA "Approached by client" ...
##  $ Competitive or sole sourced (compulsory): chr  NA NA NA NA ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   `Opportunity Name` = col_character(),
##   ..   `Account Name` = col_character(),
##   ..   Stage = col_character(),
##   ..   `Amount Currency` = col_character(),
##   ..   Amount = col_double(),
##   ..   `Created Date` = col_character(),
##   ..   `Close Date` = col_character(),
##   ..   `Primary Practice` = col_character(),
##   ..   `Business Offer` = col_character(),
##   ..   Sector = col_character(),
##   ..   Segment = col_character(),
##   ..   `Proposal director` = col_character(),
##   ..   `Proposal manager` = col_character(),
##   ..   Source = col_character(),
##   ..   `Competitive or sole sourced (compulsory)` = col_character()
##   .. )
```

The data needs cleaning up. The following changes are made:

- Rename the columns with names more suitable for analysis
- Convert “amount” column from chr to double
- Convert “creationDate” and “closeDate” to Date format

Exploration

Exploration of columns

Some basic exploration of the columns to better understand what information is useful, given business rules. There columns are name, account, stage, currency, amount, creationDate, closeDate, practice, offer, sector, segment, director, manager, source, competitiveness.

```
names(proposals)
```

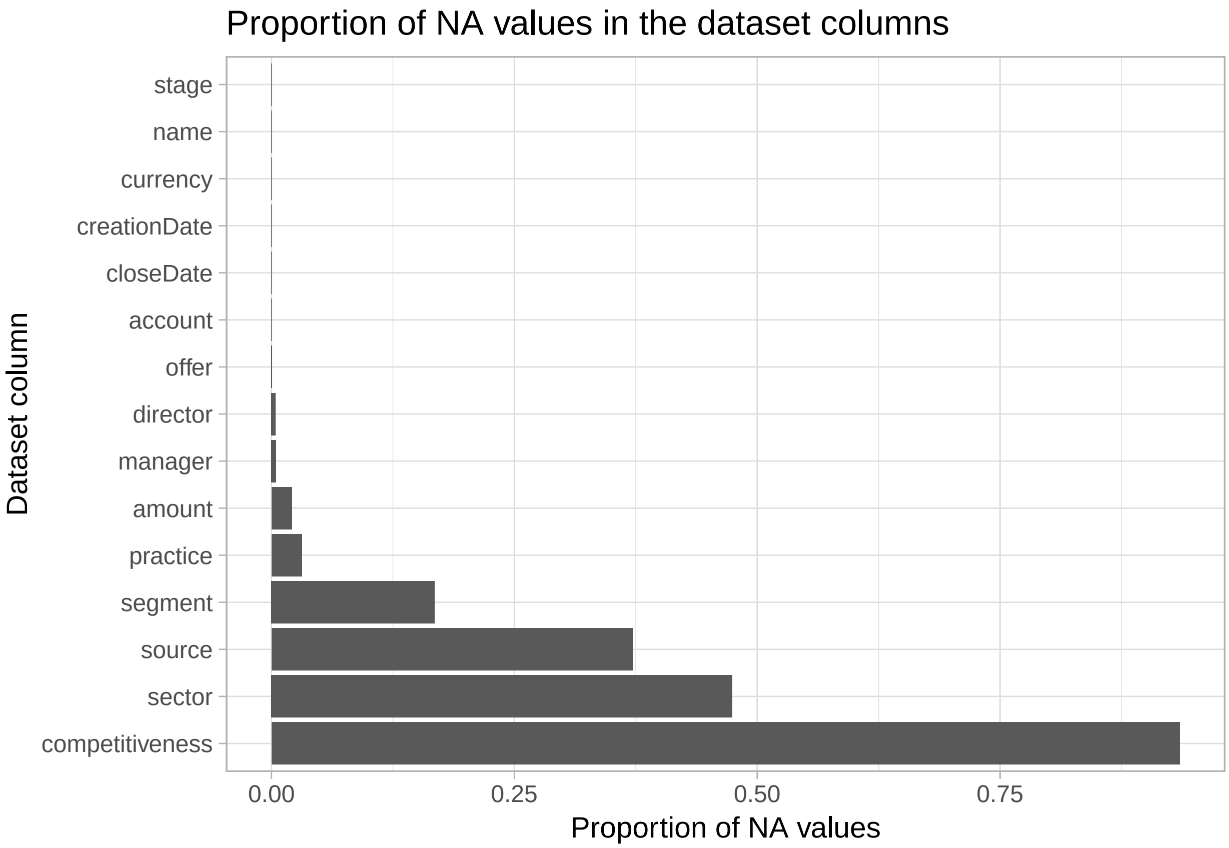
```
##  [1] "name"            "account"         "stage"
##  [4] "currency"        "amount"          "creationDate"
##  [7] "closeDate"       "practice"        "offer"
## [10] "sector"          "segment"         "director"
## [13] "manager"         "source"          "competitiveness"
```

Data integrity

The figure on the left hand side shows the proportion of NA values in the columns of the dataset. The following transformations have been applied to improve the dataset, resulting in the figure on the right hand side.

```
# Step 1: create a tibble with column names and proportions of NA values per column
n <- nrow(proposals) # number of rows in dataset
sumNAbefore <- tibble(colSums(is.na(proposals)/n)) # proportions of NA values per column
sumNAbefore <- cbind(colnames(proposals), sumNAbefore) # concatenate the vectors
names(sumNAbefore) <- c("feature", "proportion") # rename the columns to human readible

# plot NA proportions
sumNAbefore %>% ggplot(aes(reorder(feature, -proportion), proportion)) +
  geom_bar(stat = "Identity") +
  coord_flip() +
  xlab("Dataset column") +
  ylab("Proportion of NA values") +
  ggtitle("Proportion of NA values in the dataset columns") +
  theme_light()
```



The following approaches to addressing the NA values are proposed.

- Drop the column "Competitive or sole sourced (compulsory)". Although it would be interesting to see the impact of competitiveness on proposals, too many data points are missing to make it useful
- Amount: investigate if missing amounts can be replaced with amount group means for "account", "primary practice" and "business offer"
- Proposal director / manager: create "unknown" category for relevant observations. Doing this will retain observations for analysis and will not interfere with PCA
- Outline other wrangling to be conducted

```
# Step 2: transform the NA values
# 2a. drop the competitiveness column. It's too broken to be useful
proposals <- proposals %>% select(-competitiveness)

# 2b:  drop offer NA values as they only represent 2 observations
proposals <- proposals %>% filter(!is.na(offer)) # remove the offending observations

# 2c: replace amount values with average amounts for
proposals[is.na(proposals$amount),]
```

```
## # A tibble: 75 x 14
##    name  account stage currency amount creationDate closeDate  practice
##    <chr> <chr>   <chr> <chr>     <dbl> <date>       <date>     <chr>
##  1 ARU ~ Rugby ~ Nous~ AUD          NA 2017-08-23   2018-01-25 Busines~
##  2 DOF ~ Dept o~ Opp ~ AUD          NA 2017-12-04   2018-04-24 Busines~
##  3 DOH ~ NSW Mi~ Clie~ AUD          NA 2017-07-18   2017-09-07 Public ~
##  4 ANA ~ Austra~ Opp ~ AUD          NA 2016-09-09   2016-09-24 Busines~
##  5 PMC ~ Dept o~ Opp ~ AUD          NA 2017-04-21   2017-05-30 Public ~
##  6 DPC ~ Dept o~ Opp ~ AUD          NA 2016-12-10   2017-01-18 <NA>
##  7 DHV ~ Dept o~ Opp ~ AUD          NA 2018-04-05   2018-07-18 Public ~
##  8 NUK ~ Nous UK Opp ~ AUD          NA 2017-08-10   2017-08-10 Org Per~
##  9 ITA ~ Transp~ Nous~ AUD          NA 2016-08-08   2016-09-23 Public ~
## 10 ITA ~ Transp~ Nous~ AUD          NA 2016-08-08   2016-09-23 Public ~
## # ... with 65 more rows, and 6 more variables: offer <chr>, sector <chr>,
## #   segment <chr>, director <chr>, manager <chr>, source <chr>
```

```
nrow(proposals[is.na(proposals$amount),])
```

```
## [1] 75
```

```
# 75 observations are returned. If other columsn don't feature too many NA values, we can substitute NA
# with group means for account/stage/practice/offer/sector if possible
# TODO: fix amounts with group means through some method
# the following is a tempory fix - set the amount to the overall mean for amount

amounts <- as.double(proposals$amount[!is.na(proposals$amount)])
avg_amount <- mean(amounts[amounts > 999]) # calculate overall avg with values greater than 999
proposals$amount[is.na(proposals$amount)] <- avg_amount # replace NAs with avg amount
proposals$amount[proposals$amount <= 999] <- avg_amount # replace amounts < 999 with avg_amount
rm(amounts, avg_amount)

# Step 3. split name column into name and description. We don't need to reference number
proposals$account <-proposals$name %>% str_extract("^[A-Z]{3}") # 3 letter identifyer of opportunity
proposals$description <-sub("^[A-Z]{3}\\s\\d{4}\\s*[-]*\\s*", "", proposals$name) # name of opportunity
proposals <- proposals %>% select(-name) # name column now redundant. Can be dropped
```

Data exploration

First some basic exploration of the data to get an understanding of what it tells us.

Method

Analysis

Conclusions