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 and various columns have been recoded. This part of the data wrangling has not been shown.

Cleaning data

The data set is a raw export from the system we use to manage opportunities. A csv export was obtained, which after de identification has the following structure:

```r
str(proposals, give.attr = FALSE) # show structure of data
```

```
## Classes 'spec_tbl_df', 'tbl_df', 'tbl' and 'data.frame': 2821 obs. of  14 variables:
##  $ account        : chr  "account755" "account253" "account541" "account743" ...
##  $ stage          : num  1 1 1 1 0 0 0 1 1 1 ...
##  $ amount         : num  81735 99134 0 220000 150000 ...
##  $ creationDate   : chr  "10/3/18" "4/8/16" "13/3/18" "9/3/18" ...
##  $ closeDate      : chr  "9/6/18" "19/8/16" "17/7/18" "12/4/18" ...
##  $ practice       : chr  "practice3" "practice4" "practice4" "practice1" ...
##  $ offer          : chr  "offer37" "offer44" "offer44" "offer54" ...
##  $ sector         : chr  "sector2" "sector4" NA NA ...
##  $ segment        : chr  "segment25" "segment19" "segment27" "segment25" ...
##  $ director       : chr  "director92" "director131" "director124" "director143" ...
##  $ manager        : chr  "manager148" "manager17" "manager245" "manager313" ...
##  $ source         : chr  NA "source2" NA "source5" ...
##  $ competitiveness: chr  NA NA NA NA ...
##  $ code           : chr  "UOW" "DHV" "NUK" "UNS" ...
```

Description of dataset

The meaning of the columns is as follows:

Column	Explanation
account	Name of the account, i.e. the client
stage	Proposal win / loss (1 = win, 0 = loss)
amount	Estimated value of the opportunity
creationDate	Creation date of the record
closeDate	Close date of the record
practice	High level practice group for the opportunity
offer	Business offer the opportunity falls under (e.g. analytics, consulting, project management)
sector	Sector of business the client falls under (e.g. education, health, finance)
segment	Segment of the sector the opportunity falls under (e.g. insurance or loans for sector finance
director	Proposal director; in our practice we assign a proposal director and manager to each proposal
manager	Proposal manager
source	Source of the opportunity, i.e. how did we know of it (e.g. approached by client, tender
competitiveness	Level of competitiveness of the opportunity (i.e. are we only party bidding or are there more?)
code	Three letter code significing the account

Fixing data formats

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

- Convert "amount" column from num to double
- Convert "creationDate" and "closeDate" to Date format

```
# Change data types for amount, creationDate and closeDate
proposals$amount <- as.double(proposals$amount)
proposals$creationDate <- as.Date(proposals$creationDate, "%d/%m/%y")
proposals$closeDate <- as.Date(proposals$closeDate, "%d/%m/%y")
```

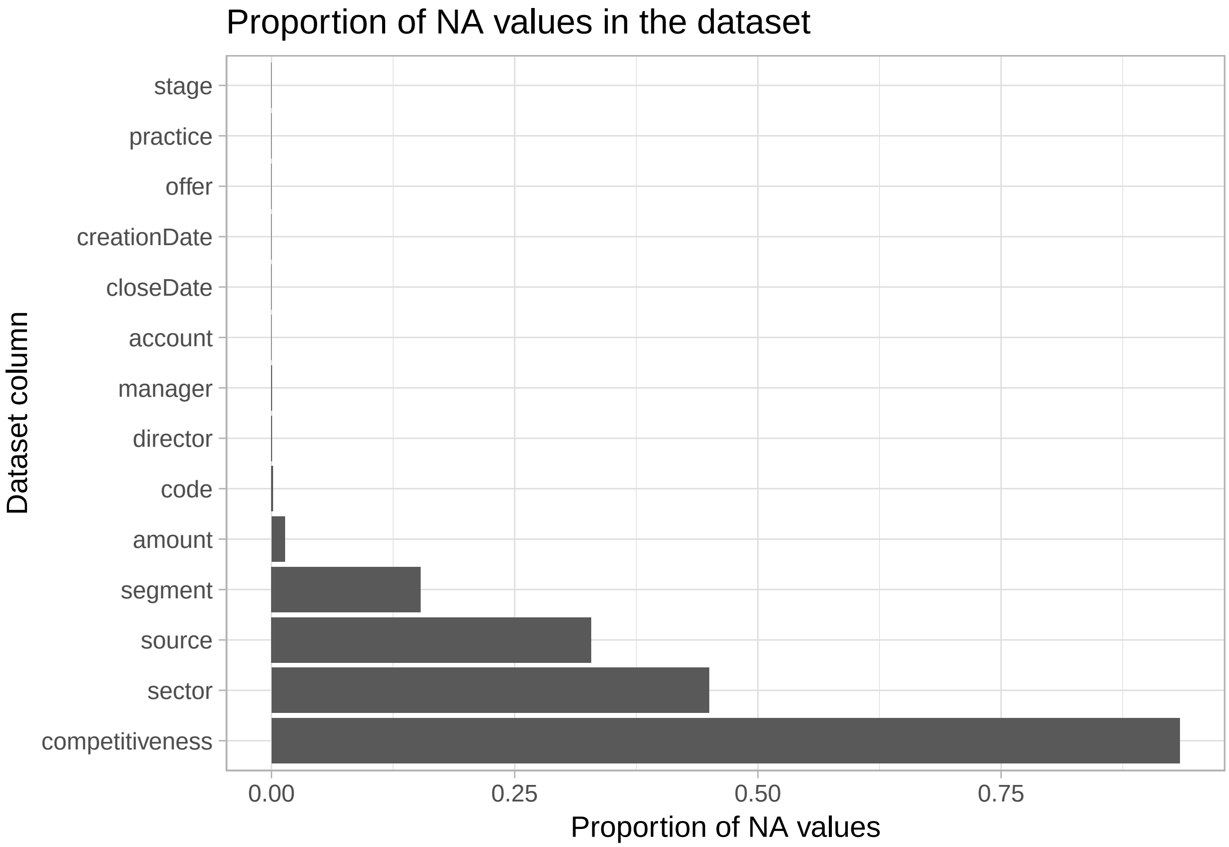
Exploration

Exploration of columns

Some basic exploration of the columns to better understand what information is useful, given business rules.

Data integrity

The figure below shows the proportion of NA values in the columns of the dataset.



The following approaches to addressing the NA values are implemented:

- *competitiveness*: drop the column. Although it would be interesting to see the impact of competitiveness on proposals, too many data points are missing to make it useful in the overall analysis. Business practice changes will need to ensure this data is consistently gathered going forward

```
proposals <- proposals %>% select(-competitiveness)
```

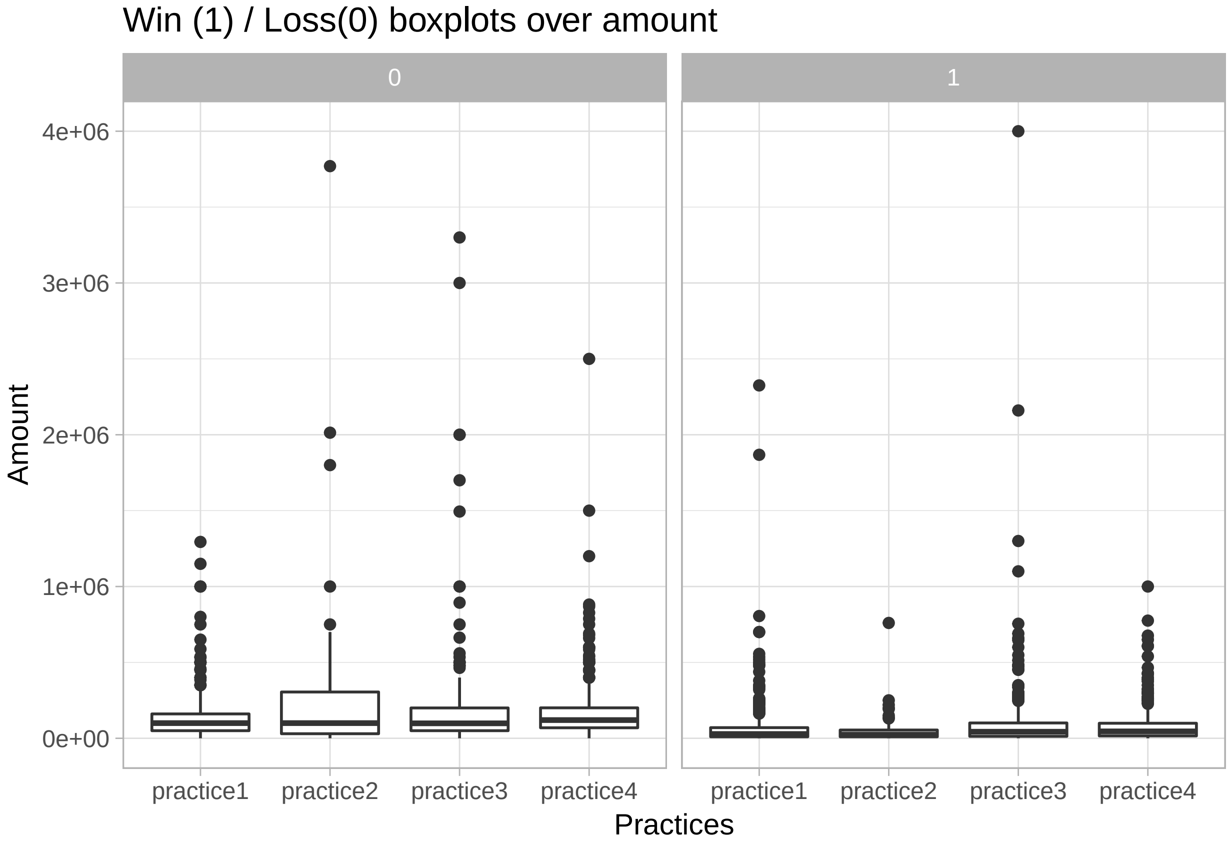
- *Amount*: Replace missing amounts with overall mean for amount, or if group means can be used (e.g. group means for *offer*). The distribution of amount and boxplots for win and loss data show significant outliers for amount. Potentially predictive analysis will need to take into account potential differences in win rates given amount of proposals

```
summary(proposals$amount)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##       0   15750   50000  107032  120000 4000000      39
```

```
proposals %>%
  ggplot(aes(practice, amount)) +
  geom_boxplot() +
  theme_light() +
  facet_wrap(~stage, ncol=2) +
  ggtitle("Win (1) / Loss(0) boxplots over amount") +
  xlab("Practices") +
  ylab("Amount")
```

```
## Warning: Removed 39 rows containing non-finite values (stat_boxplot).
```



```r
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)
```

- *Proposal director / manager*: retain the NA values as they represent uncertainty in the dataset. Replacing them with for example "Unknown", would reduce uncertainty artificially
- *Sector*: there are `sum(is.na(proposals$source))` NA values. Substitute NA values with "unknown"

```r
proposals$sector[is.na(proposals$sector)] <- "unknown"
```

- *Segment*: there are 433 NA values. We can potentially match offer data for observations with segment data and substitute missing values. Alternatively we leave NA values
- *Source*: there are 0 NA values. A quick analysis shows that there is significant predictive value in source for win/loss rates. Although many values are missing, we can try and see if a relation exists between source and other features:

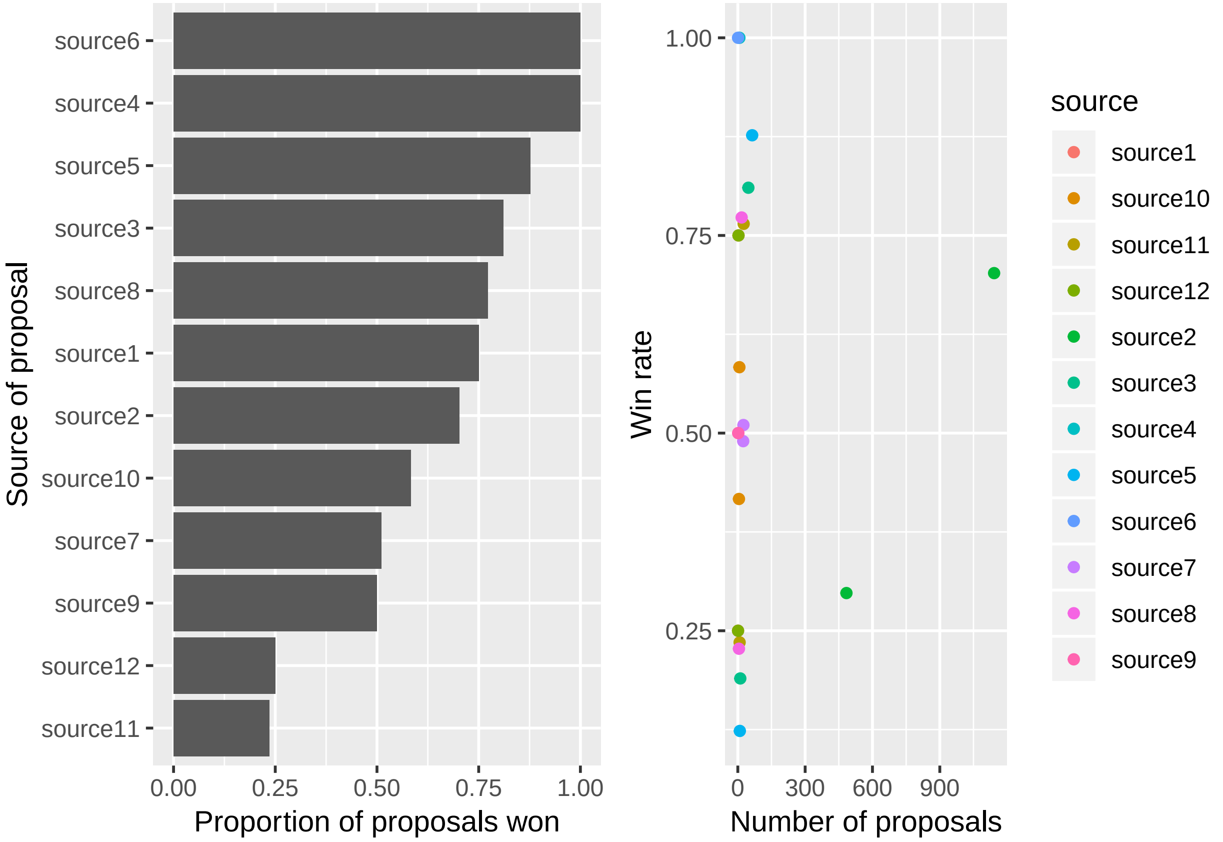
    –

```r
# proportion of winning proposals given source
sum_tab <- proposals %>%
  filter(!is.na(source)) %>%
  group_by(source, stage) %>%
  summarise(n = n(), value = sum(amount)) %>%
  ungroup() %>%
  group_by(source) %>%
  mutate(p = n / sum(n))

p1 <- sum_tab %>%
  filter(stage == 1) %>%
  ggplot(aes(reorder(source, p), p)) +
  coord_flip() +
  geom_bar(stat = "Identity") +
  xlab("Source of proposal") +
  ylab("Proportion of proposals won")

p2 <- sum_tab %>%
  ggplot(aes(n, p, colour=source)) +
  geom_point() +
  xlab("Number of proposals") +
  ylab("Win rate")

grid.arrange(p1, p2, ncol = 2)
```



```
rm(p1, p2)
```

As can be seen, the variability of win rates given sources is based on small numbers of observations. Win rates ordered by number of observations they're based on shows the following the following:

```
# table of win rates given number of observations
sum_tab %>%
  filter(stage == 1, n > 9) %>%
  arrange(desc(n)) %>%
  select(n, p, source, value)
```

```
## # A tibble: 5 x 4
## # Groups:   source [5]
##       n     p source      value
##   <int> <dbl> <chr>       <dbl>
## 1  1142 0.702 source2 75530503.
## 2    64 0.877 source5  8946054.
## 3    47 0.810 source3  3094252.
## 4    25 0.510 source7  4924297.
## 5    17 0.773 source8  1449393.
```

Data exploration

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

Method

Analysis

Conclusions