Machine-Learning-Model-For-Optimising-Banking-Campaign-Strategy

2024-02-04

This machine learning project seeks to create a valuable model capable of predicting whether a client will subscribe to a business product offered by a Portuguese bank following a marketing campaign.

Exploratory Data Analysis

Loading the libraries

```r
library(tidyverse)
```

```
## ── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──
## ✔ dplyr     1.1.4     ✔ readr     2.1.5
## ✔ forcats   1.0.0     ✔ stringr   1.5.1
## ✔ ggplot2   3.4.4     ✔ tibble    3.2.1
## ✔ lubridate 1.9.3     ✔ tidyr     1.3.1
## ✔ purrr     1.0.2     
## ── Conflicts ────────────────────────────────────────── tidyverse_conflicts() ──
## ✖ dplyr::filter() masks stats::filter()
## ✖ dplyr::lag()    masks stats::lag()
## ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to be
come errors
```

```r
library(tidymodels)
```

```
## ── Attaching packages ────────────────────────────────────── tidymodels 1.1.1 ──
## ✔ broom        1.0.5     ✔ rsample      1.2.0
## ✔ dials        1.2.0     ✔ tune         1.1.2
## ✔ infer        1.0.5     ✔ workflows    1.1.3
## ✔ modeldata    1.3.0     ✔ workflowsets 1.0.1
## ✔ parsnip      1.1.1     ✔ yardstick    1.3.0
## ✔ recipes      1.0.9     
## ── Conflicts ───────────────────────────────────────── tidymodels_conflicts() ──
## ✖ scales::discard() masks purrr::discard()
## ✖ dplyr::filter()   masks stats::filter()
## ✖ recipes::fixed()  masks stringr::fixed()
## ✖ dplyr::lag()      masks stats::lag()
## ✖ yardstick::spec() masks readr::spec()
## ✖ recipes::step()   masks stats::step()
## • Learn how to get started at https://www.tidymodels.org/start/
```

```r
library(gtsummary)
```

```
##
## Attaching package: 'gtsummary'
##
## The following objects are masked from 'package:recipes':
##
##     all_double, all_factor, all_integer, all_logical, all_numeric
```

The dataset has already been separated into training and test datasets. The dataset was published Prakhar Rathi by on Kaggle.

```
train <- read_csv2("train.csv")
```

```
## ℹ Using "','" as decimal and "'.'" as grouping mark. Use `read_delim()` for more control.
```

```
## Rows: 45211 Columns: 17
## ── Column specification ────────────────────────────────────────────────────────
## Delimiter: ";"
## chr (10): job, marital, education, default, housing, loan, contact, month, p...
## dbl  (7): age, balance, day, duration, campaign, pdays, previous
##
## ℹ Use `spec()` to retrieve the full column specification for this data.
## ℹ Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
str (train)
```

```
## spc_tbl_ [45,211 × 17] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ age      : num [1:45211] 58 44 33 47 33 35 28 42 58 43 ...
##  $ job      : chr [1:45211] "management" "technician" "entrepreneur" "blue-collar" ...
##  $ marital  : chr [1:45211] "married" "single" "married" "married" ...
##  $ education: chr [1:45211] "tertiary" "secondary" "secondary" "unknown" ...
##  $ default  : chr [1:45211] "no" "no" "no" "no" ...
##  $ balance  : num [1:45211] 2143 29 2 1506 1 ...
##  $ housing  : chr [1:45211] "yes" "yes" "yes" "yes" ...
##  $ loan     : chr [1:45211] "no" "no" "yes" "no" ...
##  $ contact  : chr [1:45211] "unknown" "unknown" "unknown" "unknown" ...
##  $ day      : num [1:45211] 5 5 5 5 5 5 5 5 5 5 ...
##  $ month    : chr [1:45211] "may" "may" "may" "may" ...
##  $ duration : num [1:45211] 261 151 76 92 198 139 217 380 50 55 ...
##  $ campaign : num [1:45211] 1 1 1 1 1 1 1 1 1 1 ...
##  $ pdays    : num [1:45211] -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 ...
##  $ previous : num [1:45211] 0 0 0 0 0 0 0 0 0 0 ...
##  $ poutcome : chr [1:45211] "unknown" "unknown" "unknown" "unknown" ...
##  $ y        : chr [1:45211] "no" "no" "no" "no" ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   age = col_double(),
##   ..   job = col_character(),
##   ..   marital = col_character(),
##   ..   education = col_character(),
##   ..   default = col_character(),
##   ..   balance = col_double(),
##   ..   housing = col_character(),
##   ..   loan = col_character(),
##   ..   contact = col_character(),
##   ..   day = col_double(),
##   ..   month = col_character(),
##   ..   duration = col_double(),
##   ..   campaign = col_double(),
##   ..   pdays = col_double(),
##   ..   previous = col_double(),
##   ..   poutcome = col_character(),
##   ..   y = col_character()
##   .. )
##  - attr(*, "problems")=<externalptr>
```

```
head (train)
```

```
## # A tibble: 6 × 17
##     age job        marital education default balance housing loan  contact   day
##   <dbl> <chr>      <chr>   <chr>     <chr>     <dbl> <chr>   <chr> <chr>   <dbl>
## 1    58 management married tertiary  no         2143 yes     no    unknown     5
## 2    44 technician single  secondary no           29 yes     no    unknown     5
## 3    33 entrepren… married secondary no            2 yes     yes   unknown     5
## 4    47 blue-coll… married unknown   no         1506 yes     no    unknown     5
## 5    33 unknown    single  unknown   no            1 no      no    unknown     5
## 6    35 management married tertiary  no          231 yes     no    unknown     5
## # ℹ 7 more variables: month <chr>, duration <dbl>, campaign <dbl>, pdays <dbl>,
## #   previous <dbl>, poutcome <chr>, y <chr>
```

1. Understanding the demographics of the clients

```
# Defining the demographic function f
# has two input variables, the tibble train data and x ( categorical variable in the train tibble)

f <- function (train, x) {
 dem <- train %>% group_by_at (vars({{x}})) %>%
  summarise ( total_clients = n(),
              percentage = round ( n()* 100 / nrow(train), 2)) %>%
  arrange (desc(total_clients))
 return(dem)
}
```

```
# Education background
f (train, education)
```

```
## # A tibble: 4 × 3
##   education total_clients percentage
##   <chr>             <int>      <dbl>
## 1 secondary         23202      51.3
## 2 tertiary          13301      29.4
## 3 primary            6851      15.2
## 4 unknown            1857       4.11
```

```
# Marital status proportion
f (train, marital)
```

```
## # A tibble: 3 × 3
##   marital  total_clients percentage
##   <chr>            <int>      <dbl>
## 1 married          27214       60.2
## 2 single           12790       28.3
## 3 divorced          5207       11.5
```

```
# Job description proportion
f (train, job)
```

```
## # A tibble: 12 × 3
##    job           total_clients percentage
##    <chr>                 <int>      <dbl>
##  1 blue-collar            9732      21.5 
##  2 management             9458      20.9 
##  3 technician             7597      16.8 
##  4 admin.                 5171      11.4 
##  5 services               4154       9.19
##  6 retired                2264       5.01
##  7 self-employed          1579       3.49
##  8 entrepreneur           1487       3.29
##  9 unemployed             1303       2.88
## 10 housemaid              1240       2.74
## 11 student                 938       2.07
## 12 unknown                 288       0.64
```

How many clients have personal loans?

```
f (train, loan)
```

```
## # A tibble: 2 × 3
##   loan  total_clients percentage
##   <chr>         <int>      <dbl>
## 1 no            37967       84.0
## 2 yes            7244       16.0
```

How many clients have credit in default?

```
f(train, default)
```

```
## # A tibble: 2 × 3
##   default total_clients percentage
##   <chr>           <int>      <dbl>
## 1 no              44396       98.2
## 2 yes               815        1.8
```

How many clients have a housing loan

```
f(train, housing)
```

```
## # A tibble: 2 × 3
##   housing total_clients percentage
##   <chr>           <int>      <dbl>
## 1 yes             25130       55.6
## 2 no              20081       44.4
```

Campaign contact summary

```
# Total contacts performed during the campaign period

sum (train$campaign)
```

```
## [1] 124956
```

```
# Types of communication methods summary
f(train, contact)
```

```
## # A tibble: 3 × 3
##   contact   total_clients percentage
##   <chr>             <int>      <dbl>
## 1 cellular          29285      64.8
## 2 unknown           13020      28.8
## 3 telephone          2906       6.43
```

```
# Average contact (last contact) duration in minutes per contact method

train %>% group_by (contact) %>%
  summarise ( total_contacts = sum(campaign),
              ave_contact_duration = round ( mean(duration/60),2)) %>%
  arrange (desc(ave_contact_duration))
```

```
## # A tibble: 3 × 3
##   contact   total_contacts ave_contact_duration
##   <chr>              <dbl>                <dbl>
## 1 cellular           78780                 4.38
## 2 unknown            36293                 4.21
## 3 telephone           9883                 3.92
```

```
# last contact day of the month
f(train, day)
```

```
## # A tibble: 31 × 3
##      day total_clients percentage
##    <dbl>         <int>      <dbl>
##  1    20          2752       6.09
##  2    18          2308       5.1
##  3    21          2026       4.48
##  4    17          1939       4.29
##  5     6          1932       4.27
##  6     5          1910       4.22
##  7    14          1848       4.09
##  8     8          1842       4.07
##  9    28          1830       4.05
## 10     7          1817       4.02
## # ℹ 21 more rows
```

```
# last contact month of the year
f(train, month)
```

```
## # A tibble: 12 × 3
##    month total_clients percentage
##    <chr>         <int>      <dbl>
##  1 may           13766      30.4 
##  2 jul            6895      15.2 
##  3 aug            6247      13.8 
##  4 jun            5341      11.8 
##  5 nov            3970       8.78
##  6 apr            2932       6.49
##  7 feb            2649       5.86
##  8 jan            1403       3.1 
##  9 oct             738       1.63
## 10 sep             579       1.28
## 11 mar             477       1.06
## 12 dec             214       0.47
```

```
# Days elapsed since the client's last contact from the last campaign ( -1 indicating that th
e client was not contacted previously)
f(train, pdays)
```

```
## # A tibble: 559 × 3
##    pdays total_clients percentage
##    <dbl>         <int>      <dbl>
##  1    -1         36954      81.7 
##  2   182           167       0.37
##  3    92           147       0.33
##  4    91           126       0.28
##  5   183           126       0.28
##  6   181           117       0.26
##  7   370            99       0.22
##  8   184            85       0.19
##  9   364            77       0.17
## 10    95            74       0.16
## # ℹ 549 more rows
```

```
# number of contacts performed before this campaign
f(train, previous)
```

```
## # A tibble: 41 × 3
##    previous total_clients percentage
##       <dbl>         <int>      <dbl>
##  1        0         36954      81.7 
##  2        1          2772       6.13
##  3        2          2106       4.66
##  4        3          1142       2.53
##  5        4           714       1.58
##  6        5           459       1.02
##  7        6           277       0.61
##  8        7           205       0.45
##  9        8           129       0.29
## 10        9            92       0.2 
## # ℹ 31 more rows
```

```
# outcome of the previous marketing campaign
f(train, poutcome)
```

```
## # A tibble: 4 × 3
##   poutcome total_clients percentage
##   <chr>            <int>      <dbl>
## 1 unknown          36959      81.8
## 2 failure           4901      10.8
## 3 other             1840       4.07
## 4 success           1511       3.34
```

Understanding the demographic effect on the campaign outcome:

```
# First we get a summary of the campaign outcome
train <- train %>% rename( outcome = y)
f(train, outcome)
```

```
## # A tibble: 2 × 3
##   outcome total_clients percentage
##   <chr>           <int>      <dbl>
## 1 no              39922       88.3
## 2 yes              5289       11.7
```

```
# Understanding the education level of clients that subscribed to the the financial product

train_yes <- train %>% filter (outcome == "yes") ## filtering clients that subscribed

# we also want add the total amount of contacts made and average duration to understand the level of resources used

f2 <- function (train_yes, x1, x2,x3) {
 dem <- train_yes %>% group_by_at (vars({{x1}})) %>%
  summarise ( total_clients = n(),
              percentage = round ( n()* 100 / nrow(train_yes), 2),
              total_contacts = sum({{x2}}),
              contacts_to_clients_ratio = round (total_contacts / total_clients,1),
              ave_duration_min = round (mean(({{x3}})/60),2)) %>%
  arrange (desc(total_clients))
 return(dem)
}

f2(train_yes, education, campaign, duration)
```

```
## # A tibble: 4 × 6
##   education total_clients percentage total_contacts contacts_to_clients_ratio
##   <chr>             <int>      <dbl>          <dbl>                     <dbl>
## 1 secondary          2450      46.3            5106                       2.1
## 2 tertiary           1996      37.7            4347                       2.2
## 3 primary             591      11.2            1348                       2.3
## 4 unknown             252       4.76            523                       2.1
## # ℹ 1 more variable: ave_duration_min <dbl>
```

```
# Understanding the marital status of clients that subscribed to the financial product

f2 (train_yes, marital, campaign, duration )
```

```
## # A tibble: 3 × 6
##   marital  total_clients percentage total_contacts contacts_to_clients_ratio
##   <chr>            <int>      <dbl>          <dbl>                     <dbl>
## 1 married           2755       52.1           6053                       2.2
## 2 single            1912       36.2           3955                       2.1
## 3 divorced           622       11.8           1316                       2.1
## # ℹ 1 more variable: ave_duration_min <dbl>
```

```
# Understanding the job description of clients that subscribed to the financial product

f2 (train_yes, job, campaign, duration )
```

```
## # A tibble: 12 × 6
##    job           total_clients percentage total_contacts contacts_to_clients_r…¹
##    <chr>                 <int>      <dbl>          <dbl>                   <dbl>
##  1 management             1301      24.6            2897                     2.2
##  2 technician              840      15.9            1812                     2.2
##  3 blue-collar             708      13.4            1548                     2.2
##  4 admin.                  631      11.9            1296                     2.1
##  5 retired                 516       9.76            966                     1.9
##  6 services                369       6.98            784                     2.1
##  7 student                 269       5.09            538                     2  
##  8 unemployed              202       3.82            394                     2  
##  9 self-employed           187       3.54            394                     2.1
## 10 entrepreneur            123       2.33            353                     2.9
## 11 housemaid               109       2.06            276                     2.5
## 12 unknown                  34       0.64             66                     1.9
## # ℹ abbreviated name: ¹contacts_to_clients_ratio
## # ℹ 1 more variable: ave_duration_min <dbl>
```

```
# Understanding the loan status of clients that subscribed to the financial product

f2 (train_yes, loan, campaign, duration )
```

```
## # A tibble: 2 × 6
##   loan  total_clients percentage total_contacts contacts_to_clients_ratio
##   <chr>         <int>      <dbl>          <dbl>                     <dbl>
## 1 no             4805      90.8           10222                       2.1
## 2 yes             484       9.15           1102                       2.3
## # ℹ 1 more variable: ave_duration_min <dbl>
```

```
# Understanding the default status of clients that subscribed to the financial product

f2 (train_yes, default, campaign, duration )
```

```
## # A tibble: 2 × 6
##   default total_clients percentage total_contacts contacts_to_clients_ratio
##   <chr>           <int>      <dbl>          <dbl>                     <dbl>
## 1 no               5237      99.0           11212                       2.1
## 2 yes                52       0.98            112                       2.2
## # ℹ 1 more variable: ave_duration_min <dbl>
```

```
# Understanding the effect of communication method on the number of clients that subscribed t
o the financial product

f2 (train_yes, contact, campaign, duration )
```

```
## # A tibble: 3 × 6
##   contact   total_clients percentage total_contacts contacts_to_clients_ratio
##   <chr>             <int>      <dbl>          <dbl>                     <dbl>
## 1 cellular           4369      82.6            9077                       2.1
## 2 unknown             530      10.0            1312                       2.5
## 3 telephone           390       7.37            935                       2.4
## # ℹ 1 more variable: ave_duration_min <dbl>
```

```
# Understanding the housing status of clients that subscribed to the financial product

f2 (train_yes, housing, campaign, duration )
```

```
## # A tibble: 2 × 6
##   housing total_clients percentage total_contacts contacts_to_clients_ratio
##   <chr>           <int>      <dbl>          <dbl>                     <dbl>
## 1 no               3354       63.4           7036                       2.1
## 2 yes              1935       36.6           4288                       2.2
## # ℹ 1 more variable: ave_duration_min <dbl>
```

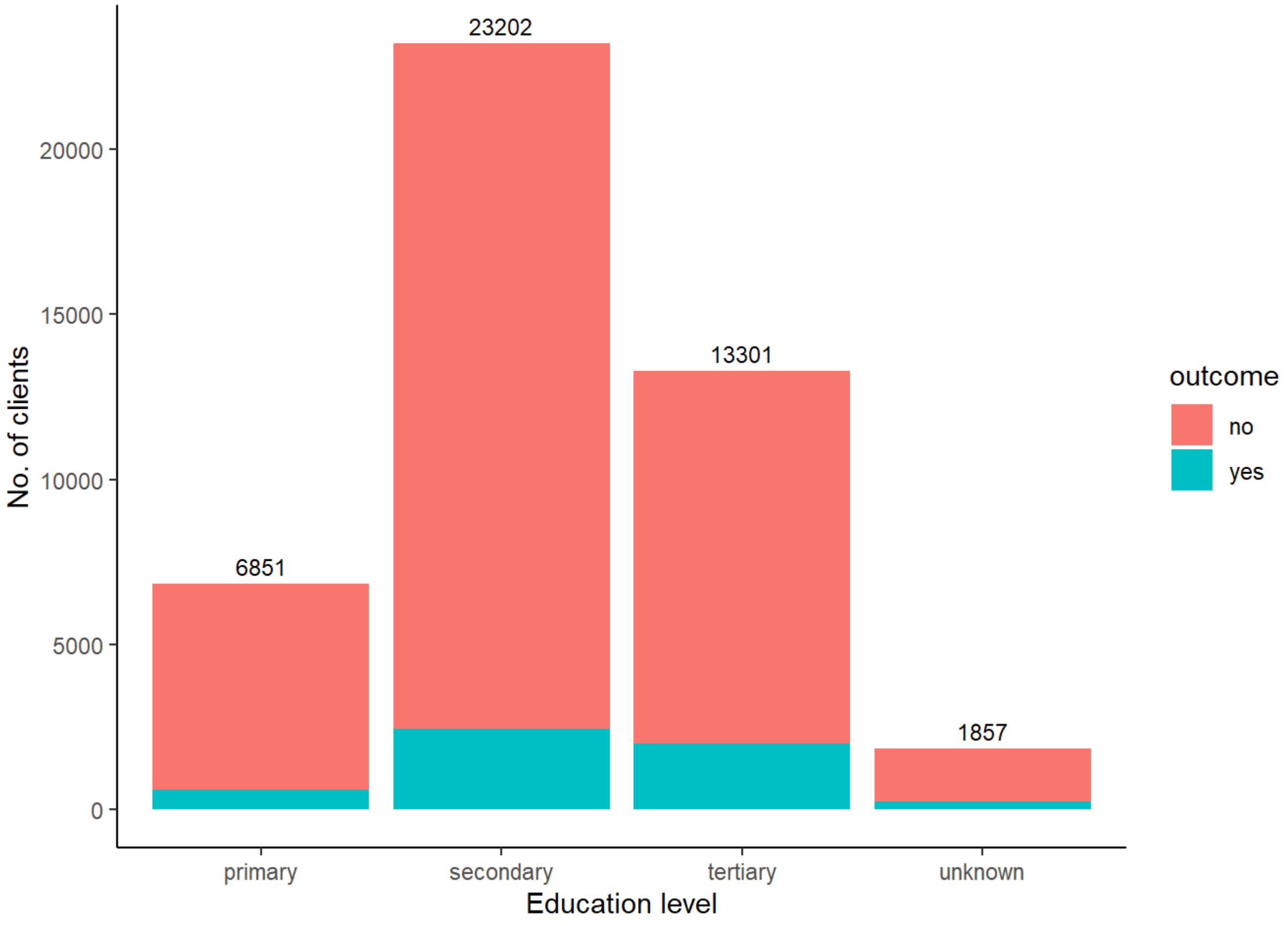
```
# Understanding the effect of outcome of previous campaign on clients that subscribed to the
financial product

f2 (train_yes, poutcome, campaign, duration )
```

```
## # A tibble: 4 × 6
##   poutcome total_clients percentage total_contacts contacts_to_clients_ratio
##   <chr>            <int>      <dbl>          <dbl>                     <dbl>
## 1 unknown           3386       64.0           7923                       2.3
## 2 success            978       18.5           1678                       1.7
## 3 failure            618       11.7           1084                       1.8
## 4 other              307        5.8            639                       2.1
## # ℹ 1 more variable: ave_duration_min <dbl>
```

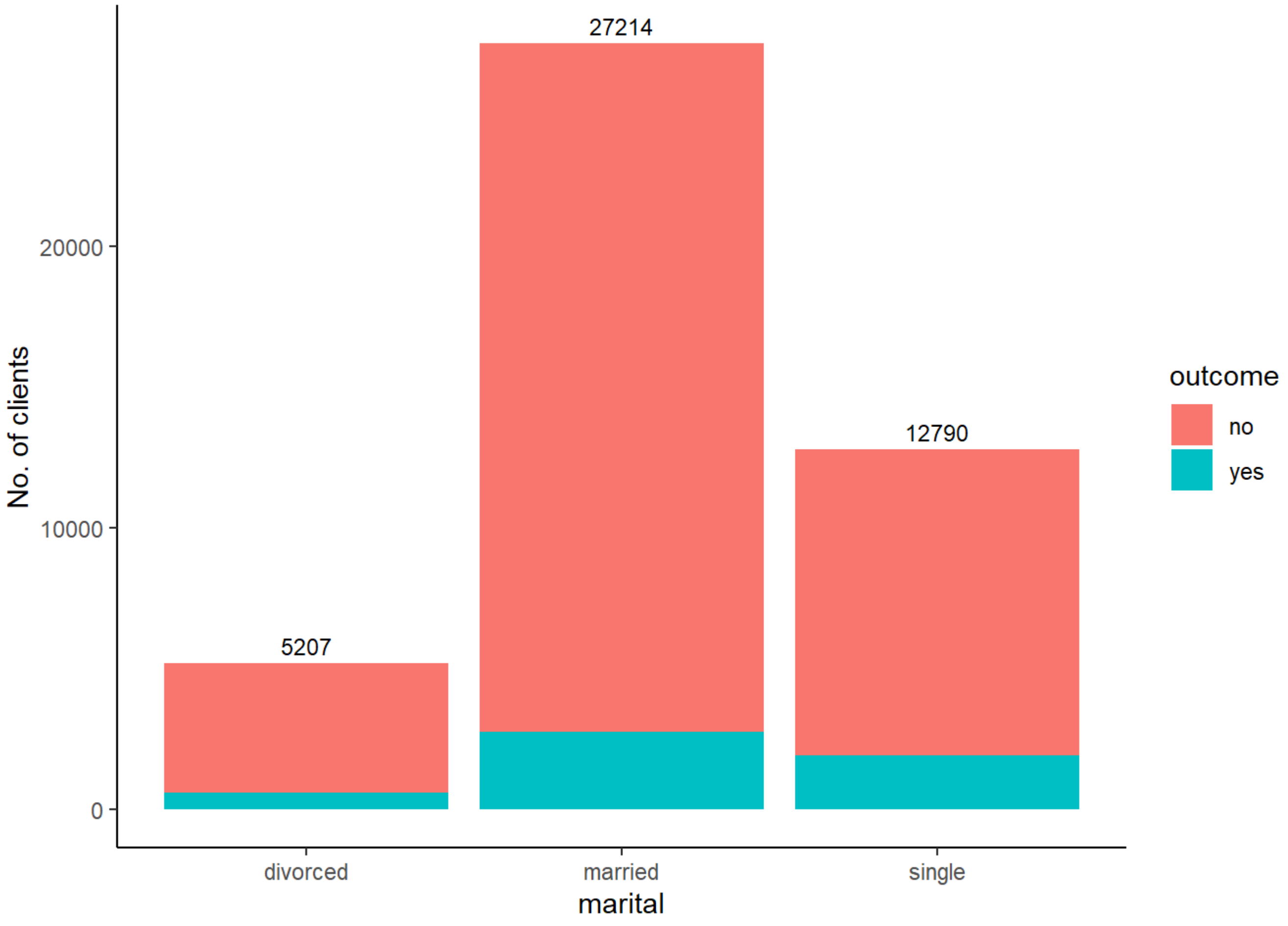
```
# Evaluating if the yearly average balance affect the campaign outcome

f3 <- function (train, x1,x2) {
 dem <- train %>% group_by_at (vars({{x1}})) %>%
  summarise ( total_clients = n(),
              percentage = round ( n()* 100 / nrow(train), 2),
              ave_balance = round (mean({{x2}}),1)) %>%
  arrange (desc(total_clients))
 return(dem)
}
f3(train, outcome, balance)
```

```
## # A tibble: 2 × 4
##   outcome total_clients percentage ave_balance
##   <chr>           <int>      <dbl>       <dbl>
## 1 no              39922       88.3       1304.
## 2 yes              5289       11.7       1804.
```

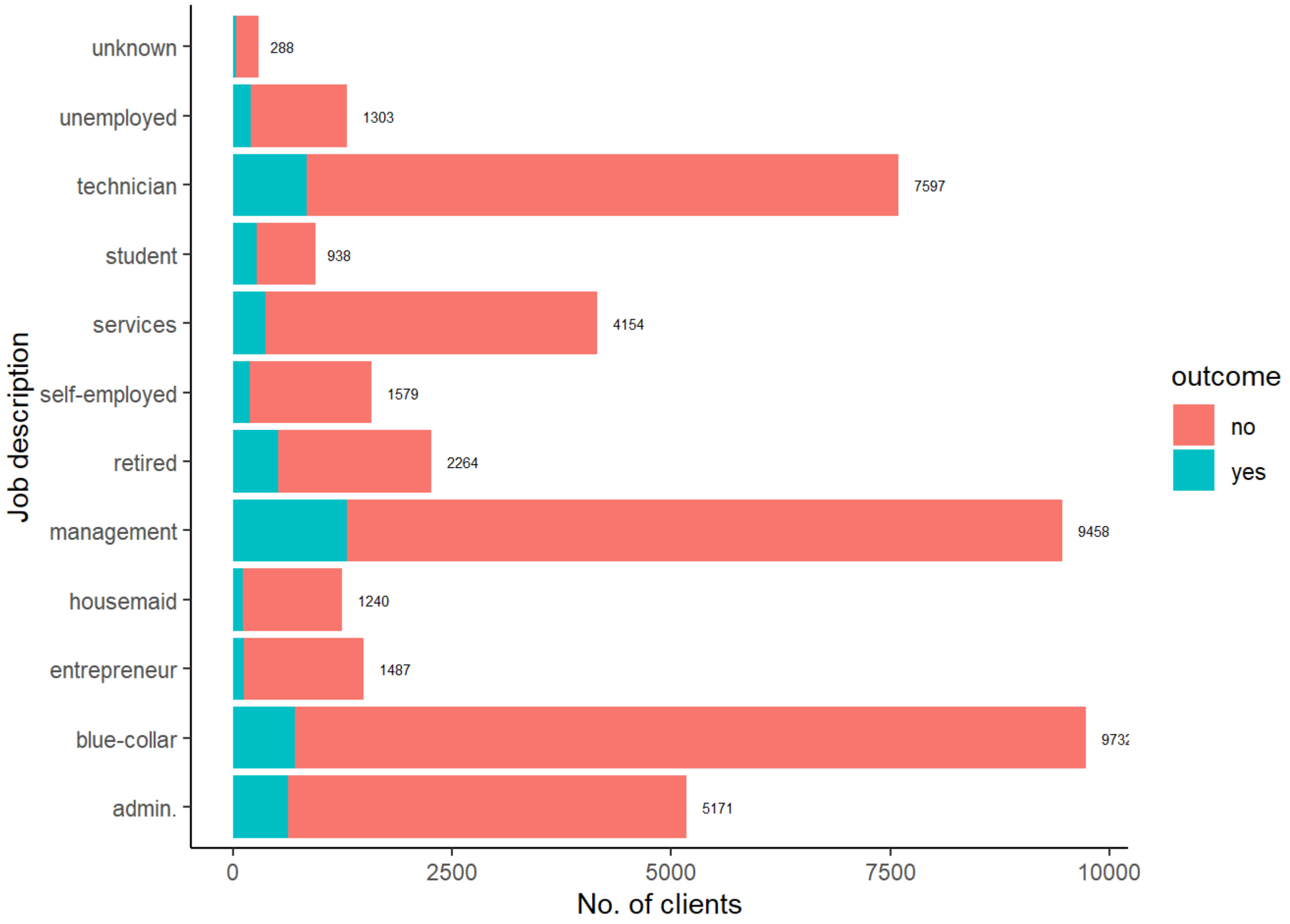
```
# Education bargraph

train %>% ggplot ( aes(education)) + geom_bar(aes(fill = outcome)) +
  geom_text(aes(label = after_stat(count)), stat = "count", position = "stack",vjust= -0.5, s
ize = 3) +
  theme_classic() + labs (x = "Education level", y = "No. of clients")
```



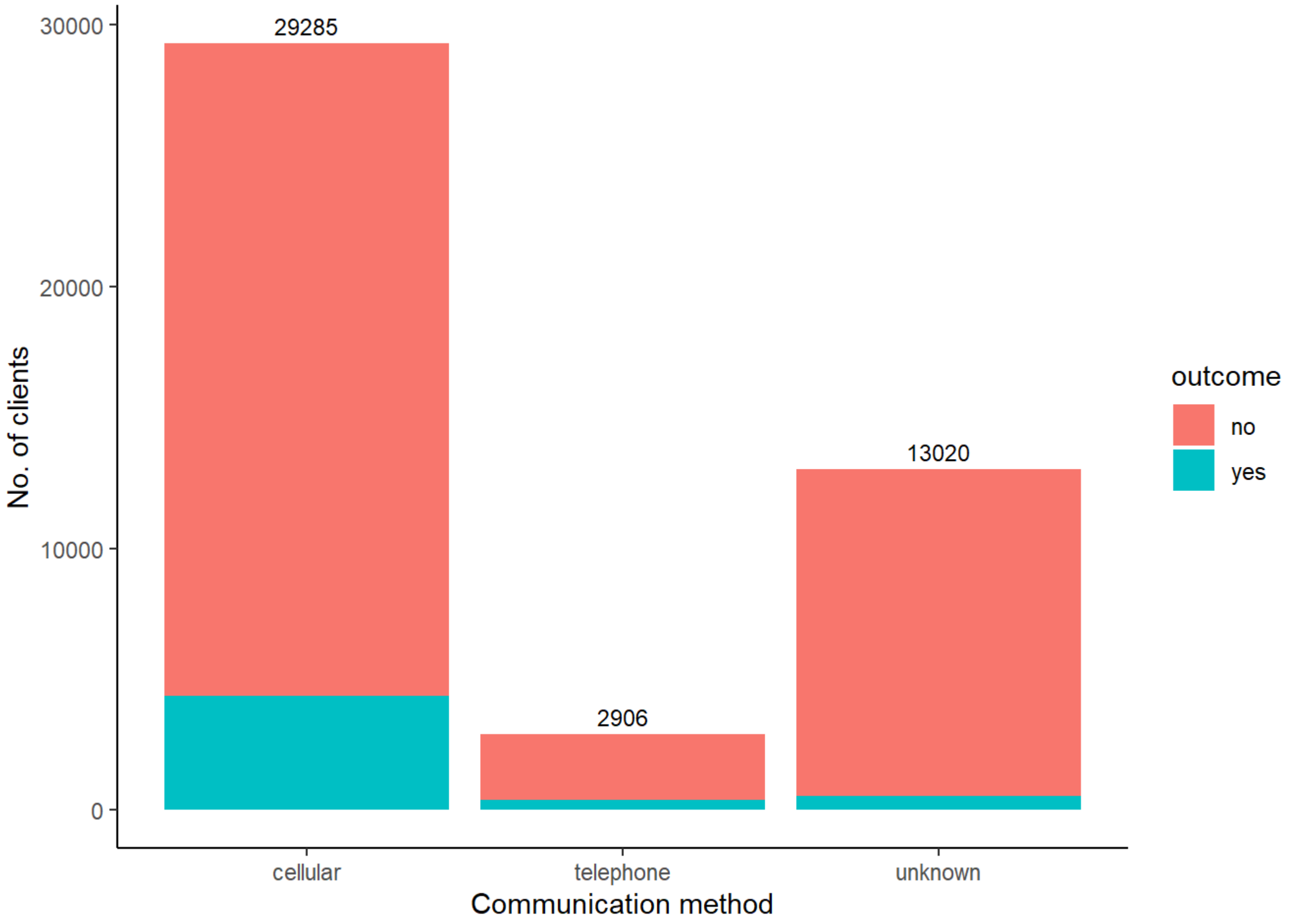
```
# Marital status bargraph
# Education bargraph

train %>% ggplot ( aes(marital)) + geom_bar(aes(fill = outcome)) +
  geom_text(aes(label = after_stat(count)), stat = "count", position = "stack",vjust= -0.5, s
ize = 3) +
  theme_classic() + labs (y = "No. of clients", X = "Marital status")
```



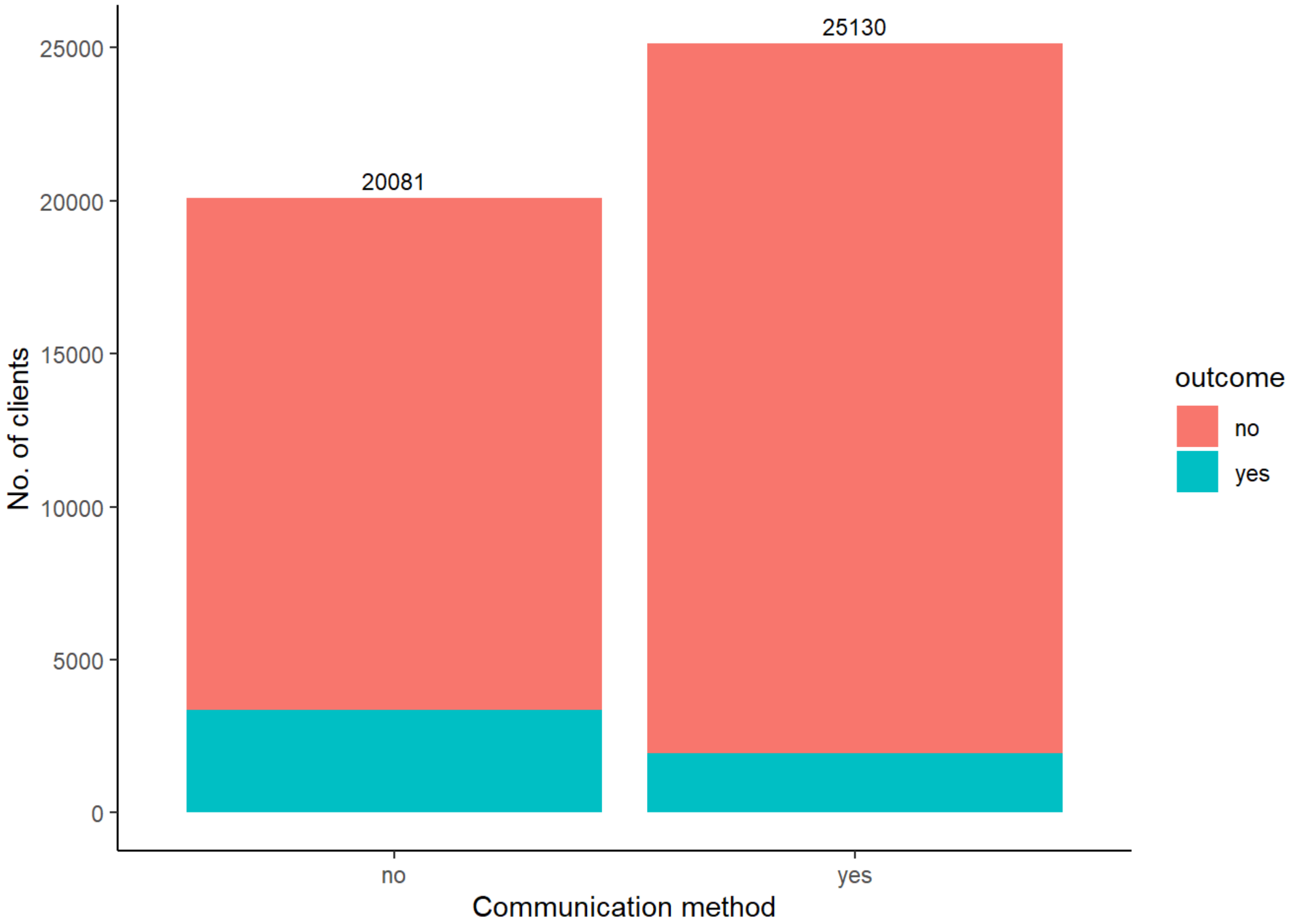
```
# Job description bargraph
train %>% ggplot ( aes(y = job)) + geom_bar(aes(fill = outcome)) +
  geom_text(aes(label = after_stat(count)), stat = "count", position = "stack",hjust= -0.5, s
ize = 2) +
  theme_classic() + labs (y = "Job description", x = "No. of clients")
```



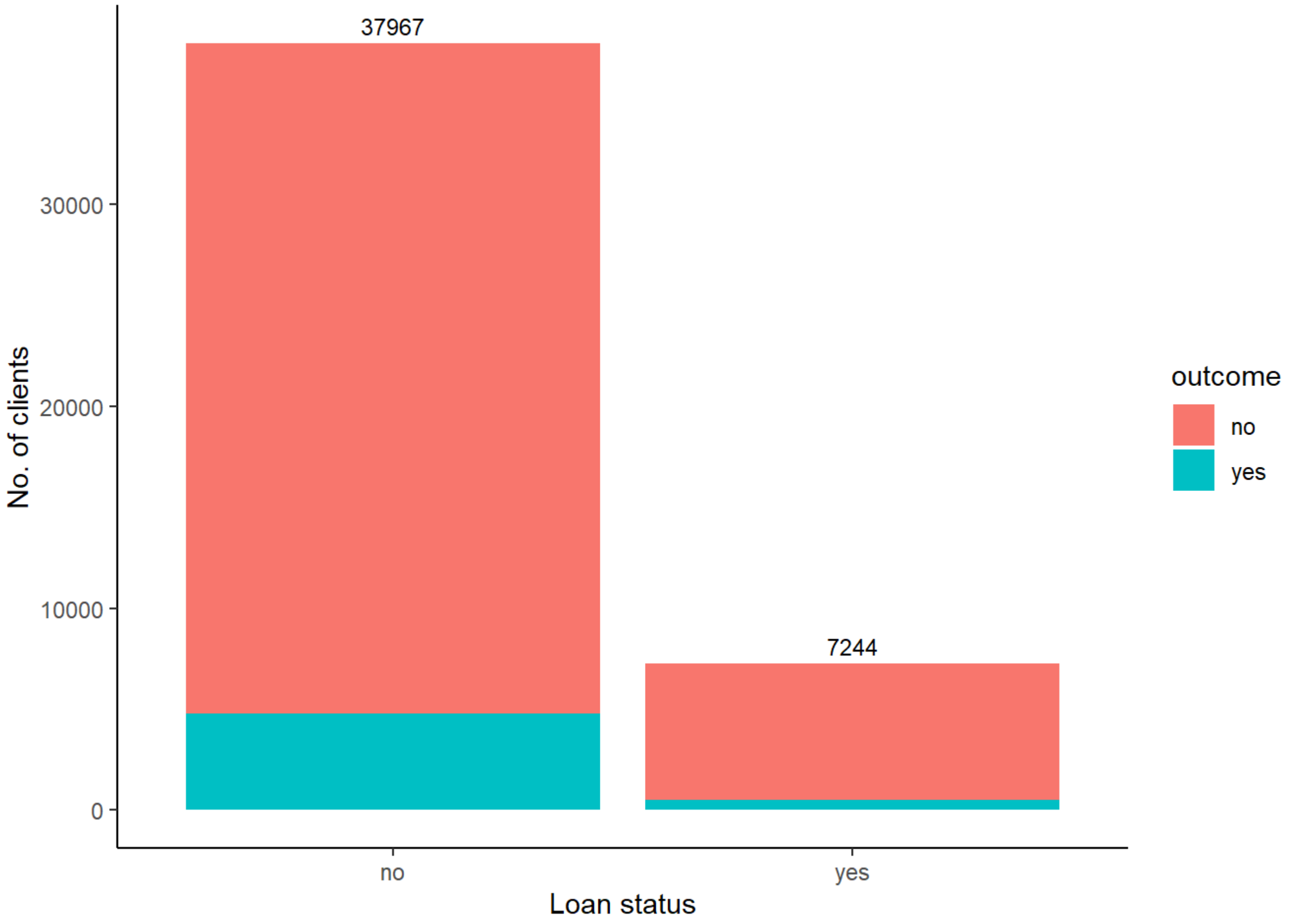
```
# Job description bargraph
train %>% ggplot ( aes(contact)) + geom_bar(aes(fill = outcome)) +
  geom_text(aes(label = after_stat(count)), stat = "count", position = "stack",vjust= - 0.5,
size = 3) +
  theme_classic() + labs (y = "No. of clients", x = "Communication method")
```



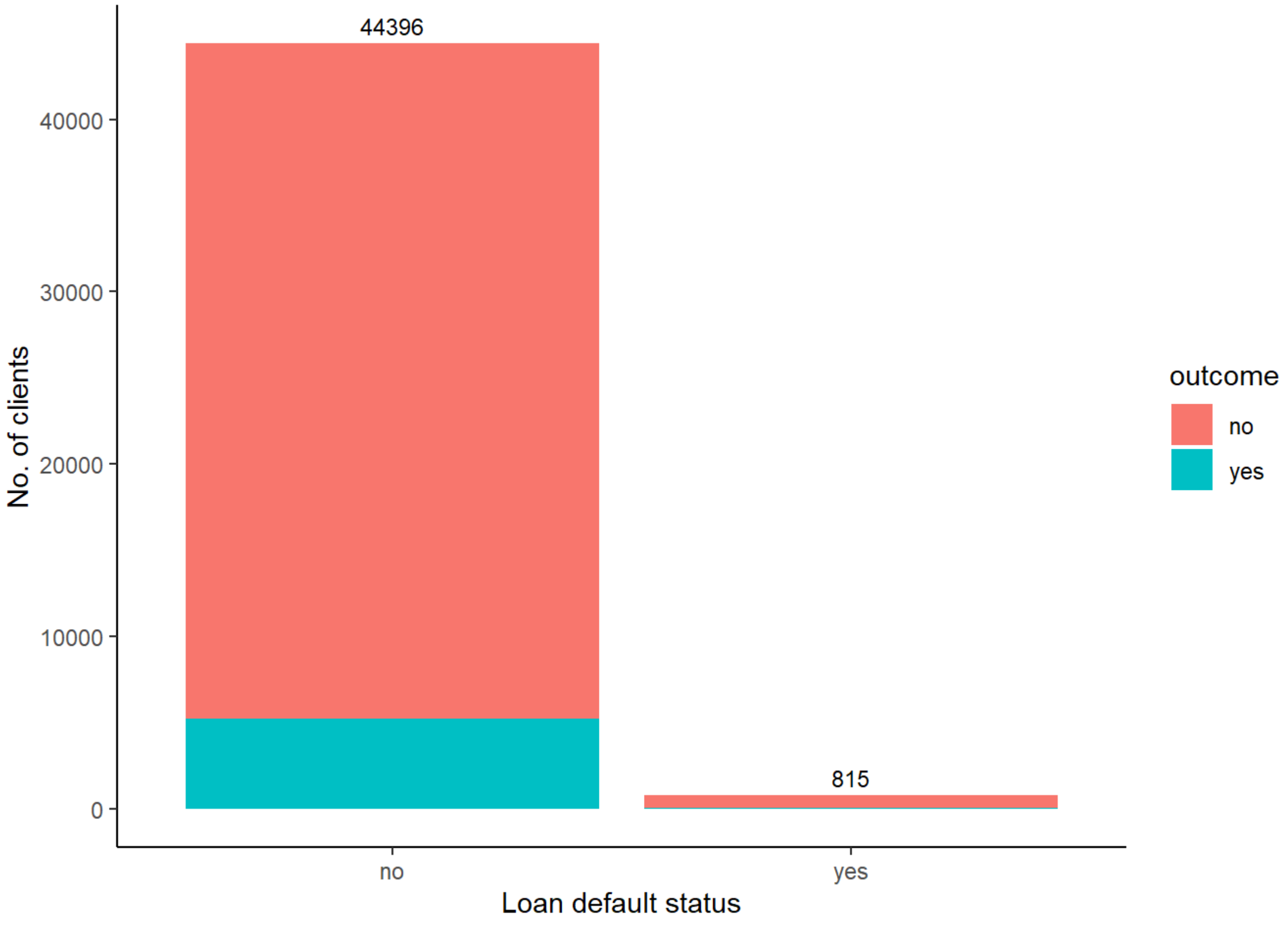
```
# Housing loan status bargraph
train %>% ggplot ( aes(housing)) + geom_bar(aes(fill = outcome)) +
  geom_text(aes(label = after_stat(count)), stat = "count", position = "stack",vjust= - 0.5,
size = 3) +
  theme_classic() + labs (y = "No. of clients", x = "Communication method")
```



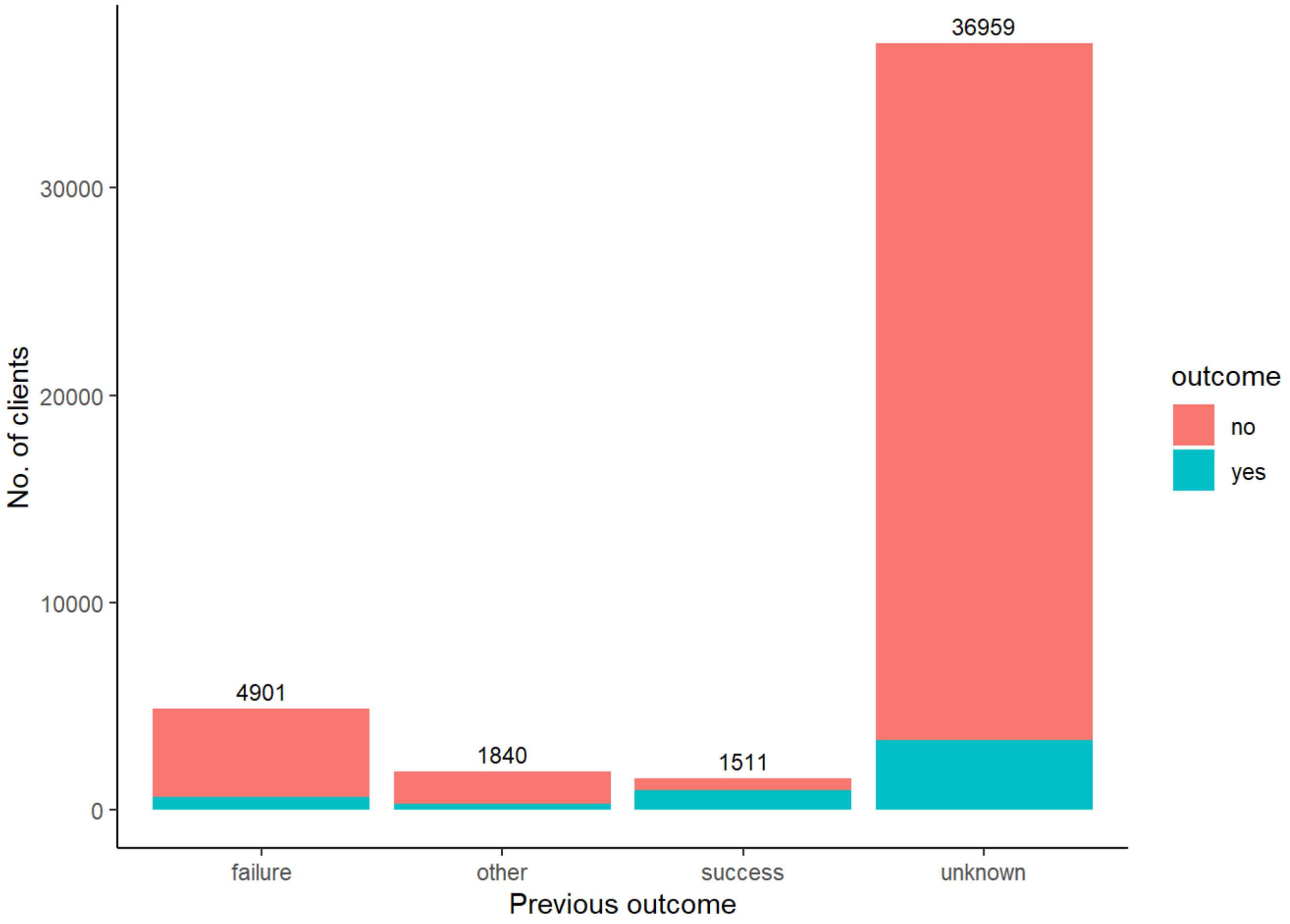
```
# Loan status bargraph
train %>% ggplot ( aes(loan)) + geom_bar(aes(fill = outcome)) +
  geom_text(aes(label = after_stat(count)), stat = "count", position = "stack",vjust= - 0.5,
size = 3) +
  theme_classic() + labs (y = "No. of clients", x = "Loan status")
```



```
# Loan default status bargraph
train %>% ggplot ( aes(default)) + geom_bar(aes(fill = outcome)) +
  geom_text(aes(label = after_stat(count)), stat = "count", position = "stack",vjust= - 0.5,
size = 3) +
  theme_classic() + labs (y = "No. of clients", x = "Loan default status")
```



```
# Previous campaign outcome  bargraph
train %>% ggplot ( aes(poutcome)) + geom_bar(aes(fill = outcome)) +
  geom_text(aes(label = after_stat(count)), stat = "count", position = "stack",vjust= - 0.5,
size = 3) +
  theme_classic() + labs (y = "No. of clients", x = "Previous outcome")
```


36959
30000
20000
10000
0
No. of clients
4901
1840
1511
failure
other
success
unknown
Previous outcome
outcome
no
yes