ML algorithm for Harvardx Capstone Course, Choose Your Own project

Rubén Vázquez del Valle

4/12/2021

Contents

1.Introduction

According to Wikipedia:

1. Marketing is currently defined by the American Marketing Association (AMA) as " the performance of business activities that direct the flow of goods, and services from producers to consumers"
2. Direct marketing is a form of communicating an offer, where organizations communicate directly to a pre-selected customer and supply a method for a direct response.

Hence we can understand that direct marketing campaigns are several processes producers undertake to engage directly its target consumers, build strong relationships to create value in order to capture value in return and get a fast and direct response.

As time goes by and technology advances those processes evolved, and keep on evolving, from different analog channels such as reply cards, reply forms to be sent in an envelope, to new and more sophisticated digital ones such as websites, text messages sent to cellular phone and/or email addresses.

One not so long old-fashioned direct marketing technique was phone calls.

The purpose of this project is analyzing if such technique applied in this case by a Portuguese banking institution not so much time ago, had positive effect or not in its clients, or in other words building a classification system to predict if the client would subscribe a term deposit.

The origin dataset has been downloaded from The UCI Machine Learning Repository

Once dataset was created, I have based my work on (Irizarry, 2021) *"HarvardX - PH125.8x course: Data Science - Machine Learning"* specifically on machine learning techniques applied to supervised learning.

2 Methodology

On this section, following tasks such as analyzing the data provided, cleaning, wrangling and preparing it in case of actions were needed to decide which kind of algorithms will be worthy in terms of classification problems, will be adressed.

Thus next steps will be:

A preliminary analysis where basically observing and sumarizing source files and datasets.

A cross validation to split source dataset into training, testing and validation datasets.

A bivariate and multivariate analysis on training dataset to decide which features include or discard from candidate models.

A modeling phase where some classifier models will be trained to predict dependent variable.

A decision phase where based on results obtained on test set with the previous models deciding which one to choose or ensemble.

2.1 Preliminary Analysis

After downloading source data it is observable that there are three files compressed: bank-additional-full.csv , bank-additional-names.txt, bank-additional.csv

bank-additional-names.txt is a text document with some basic information about the remaining ones. Here it is stated that bank-additional.csv is a random sample subset of bank-additional-full.csv.

Applying basic relational algebra, a simple way to confirm that a set B is a subset of another one A, is getting the cardinality of B/A, or equivalently getting the size of the anti_join function. While anti_join() return all rows from B without a match in A, having and antijoin by all columns with size 0 means that all rows in B are present in A, which indeed confirms that B is a subset of A.

Hence applying anti_join to bank-additional.csv on bank-additional-full.csv by all columns, the number of rows obtained is: 0 which confirms that data file to be used is bank-additional-full.csv

Besides, bank-additional-names.txt also provides us with a description of the fields in bank-additional-full.csv as follows:

```
1.  age: Age of the client.
2.  job: Type of job of the client.
3.  marital: Marital status, note "divorced" means divorced or widowed.
4.  education: Educational degree reached by the client.
5.  default: Binary variable meaning if the client has credit/credits in default.
6.  housing: Binary variable meaning if the client has a mortgage.
7.  loan: Binary variable meaning if the client has a personal loan.
8.  contact: Contact communication type with the client.
9.  month: Last contact month of year.
10. day_of_week: Last contact day of the week.
11. duration: Last contact duration, in seconds.
12. campaign: For each  client, number of contacts performed during this campaign.
13. pdays: Days since last contact from a previous campaign (999 means not previously contacted).
14. previous: For each  client, number of contacts performed before this campaign.
15. poutcome: Outcome of the previous marketing campaign if existing.
16. emp.var.rate: Employment variation rate - quarterly indicator.
17. cons.price.idx: Consumer price index - monthly indicator.
18. cons.conf.idx: Consumer confidence index - monthly indicator.
19. euribor3m: Euribor 3 month rate - daily indicator.
```

20. nr.employed: Number of employees - quarterly indicator.
21. y: Output binary variable if the client has subscribed a term deposit or not.

Now that source data has been properly addressed, let's continue by exploring and summarizing the dataset bank-additional-full.csv:

age	job	marital	education	default	housing	loan	contact
56	housemaid	married	basic.4y	no	no	no	telephone
57	services	married	high.school	unknown	no	no	telephone
37	services	married	high.school	no	yes	no	telephone
40	admin.	married	basic.6y	no	no	no	telephone
56	services	married	high.school	no	no	yes	telephone
45	services	married	basic.9y	unknown	no	no	telephone

month	day_of_week	duration	campaign	pdays	previous	poutcome
may	mon	261	1	999	0	nonexistent
may	mon	149	1	999	0	nonexistent
may	mon	226	1	999	0	nonexistent
may	mon	151	1	999	0	nonexistent
may	mon	307	1	999	0	nonexistent
may	mon	198	1	999	0	nonexistent

emp.var.rate	cons.price.idx	cons.conf.idx	euribor3m	nr.employed	y
1.1	93.994	-36.4	4.857	5191	no
1.1	93.994	-36.4	4.857	5191	no
1.1	93.994	-36.4	4.857	5191	no
1.1	93.994	-36.4	4.857	5191	no
1.1	93.994	-36.4	4.857	5191	no
1.1	93.994	-36.4	4.857	5191	no

```
##       age            job              marital           education
##  Min.   :17.00   Length:41188       Length:41188       Length:41188
##  1st Qu.:32.00   Class :character   Class :character   Class :character
##  Median :38.00   Mode  :character   Mode  :character   Mode  :character
##  Mean   :40.02
##  3rd Qu.:47.00
##  Max.   :98.00
##    default            housing              loan             contact
##  Length:41188       Length:41188       Length:41188       Length:41188
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##     month           day_of_week           duration         campaign
##  Length:41188       Length:41188       Min.   :   0.0   Min.   : 1.000
##  Class :character   Class :character   1st Qu.: 102.0   1st Qu.: 1.000
##  Mode  :character   Mode  :character   Median : 180.0   Median : 2.000
##                                        Mean   : 258.3   Mean   : 2.568
```

```
##                                      3rd Qu.: 319.0   3rd Qu.: 3.000
##                                      Max.   :4918.0   Max.   :56.000
##      pdays          previous       poutcome          emp.var.rate
##  Min.   :  0.0   Min.   :0.000   Length:41188       Min.   :-3.40000
##  1st Qu.:999.0   1st Qu.:0.000   Class :character   1st Qu.:-1.80000
##  Median :999.0   Median :0.000   Mode  :character   Median : 1.10000
##  Mean   :962.5   Mean   :0.173                      Mean   : 0.08189
##  3rd Qu.:999.0   3rd Qu.:0.000                      3rd Qu.: 1.40000
##  Max.   :999.0   Max.   :7.000                      Max.   : 1.40000
##  cons.price.idx  cons.conf.idx     euribor3m      nr.employed
##  Min.   :92.20   Min.   :-50.8   Min.   :0.634   Min.   :4964
##  1st Qu.:93.08   1st Qu.:-42.7   1st Qu.:1.344   1st Qu.:5099
##  Median :93.75   Median :-41.8   Median :4.857   Median :5191
##  Mean   :93.58   Mean   :-40.5   Mean   :3.621   Mean   :5167
##  3rd Qu.:93.99   3rd Qu.:-36.4   3rd Qu.:4.961   3rd Qu.:5228
##  Max.   :94.77   Max.   :-26.9   Max.   :5.045   Max.   :5228
##       y
##  Length:41188
##  Class :character
##  Mode  :character
##
##
##
```

At simple glance, it can be observed that the dataset is a data.frame containing 41,188 rows and 21 columns, the last one with the prediction values. It is also evident the split between numerical data and categorical data. In order to be able to address this first issue it is necessary to transform characters into factors for columns: job , marital , education , default , housing , loan , contact , month , day_of_week, poutcome , y

Let's explore possible values for the categorical columns:

```
## job possible values are :
##  [1] "admin."        "blue-collar"   "entrepreneur"  "housemaid"
##  [5] "management"    "retired"       "self-employed" "services"
##  [9] "student"       "technician"    "unemployed"    "unknown"
## ------------------------------------------------------------------------------
## marital possible values are :
## [1] "divorced" "married"  "single"   "unknown"
## ------------------------------------------------------------------------------
## education possible values are :
## [1] "basic.4y"            "basic.6y"            "basic.9y"
## [4] "high.school"         "illiterate"          "professional.course"
## [7] "university.degree"   "unknown"
## ------------------------------------------------------------------------------
## default possible values are :
## [1] "no"      "unknown" "yes"
## ------------------------------------------------------------------------------
## housing possible values are :
## [1] "no"      "unknown" "yes"
## ------------------------------------------------------------------------------
## loan possible values are :
## [1] "no"      "unknown" "yes"
## ------------------------------------------------------------------------------
## contact possible values are :
```

```
## [1] "cellular"  "telephone"
## ------------------------------------------------------------------------------
## month possible values are :
##  [1] "apr" "aug" "dec" "jul" "jun" "mar" "may" "nov" "oct" "sep"
## ------------------------------------------------------------------------------
## day_of_week possible values are :
## [1] "fri" "mon" "thu" "tue" "wed"
## ------------------------------------------------------------------------------
## poutcome possible values are :
## [1] "failure"     "nonexistent" "success"
## ------------------------------------------------------------------------------
## y possible values are :
## [1] "no"  "yes"
## ------------------------------------------------------------------------------
```

There is also a second fact observable at simple glance. There seems not to be any missing values within our data, since whenever it occurred the field was populated with "unknown".

At this point, it is fair to assume then that description of columns of our dataset provided in file bank-additional-names.txt is correct. It is also important to recall the prevalence effect on binary variable to be predicted, around 88.73458% of the answers are no, which a priori matches what I would expect in terms of this marketing campaign success ratio.

2.2.Partitioning, preprocessing and transforming original dataset

As soon as original dataset inspection is concluded next step consists on partitioning, preprocessing and transforming the source dataset.

Let's start by partitioning it. In order to avoid as much as possible over-training I will apply K-fold cross validation. As I do not have an independent dataset where validate the results of my models, I will remove randomly a part of the original dataset splitting it in two sets: the training dataset that I will refer as bankraw and the validation dataset, that I will call validation.

To do this split I applied Pareto principle, choosing 80% of original dataset as training dataset and remaining 20% as validation one. Validation dataset will only and exclusively be used for evaluation purposes at the end of the project.

Once categorical values have been transformed from character to factor, and partitions between bankraw and validation have been created, let's look again how the values of those columns are distributed within bankraw dataset:

```
##       age                job          marital                    education
##  Min.   :17.00   admin.     :8325   divorced: 3709   university.degree  :9746
##  1st Qu.:32.00   blue-collar:7410   married :19917   high.school        :7575
##  Median :38.00   technician :5417   single  : 9265   basic.9y           :4810
##  Mean   :40.03   services   :3142   unknown :   59   professional.course:4211
##  3rd Qu.:47.00   management :2308                    basic.4y           :3355
##  Max.   :98.00   retired    :1368                    basic.6y           :1844
##                  (Other)    :4980                    (Other)            :1409
##     default         housing           loan            contact
##  no     :26097   no     :14908   no     :27186   cellular :20862
##  unknown: 6852   unknown:  781   unknown:  781   telephone:12088
##  yes    :    1   yes    :17261   yes    : 4983
##
##
```

```
##
##
##      month       day_of_week    duration         campaign          pdays
##  may    :11080   fri:6238    Min.   :   0.0   Min.   : 1.000   Min.   :  0.0
##  jul    : 5696   mon:6802    1st Qu.: 103.0   1st Qu.: 1.000   1st Qu.:999.0
##  aug    : 4953   thu:6943    Median : 179.0   Median : 2.000   Median :999.0
##  jun    : 4264   tue:6488    Mean   : 257.7   Mean   : 2.564   Mean   :962.6
##  nov    : 3234   wed:6479    3rd Qu.: 318.0   3rd Qu.: 3.000   3rd Qu.:999.0
##  apr    : 2103               Max.   :4918.0   Max.   :56.000   Max.   :999.0
##  (Other): 1620
##     previous             poutcome      emp.var.rate      cons.price.idx
##  Min.   :0.0000   failure    : 3389   Min.   :-3.40000   Min.   :92.20
##  1st Qu.:0.0000   nonexistent:28466   1st Qu.:-1.80000   1st Qu.:93.08
##  Median :0.0000   success    : 1095   Median : 1.10000   Median :93.75
##  Mean   :0.1715                       Mean   : 0.08197   Mean   :93.58
##  3rd Qu.:0.0000                       3rd Qu.: 1.40000   3rd Qu.:93.99
##  Max.   :7.0000                       Max.   : 1.40000   Max.   :94.77
##
##  cons.conf.idx     euribor3m      nr.employed     y
##  Min.   :-50.8   Min.   :0.634   Min.   :4964   no :29238
##  1st Qu.:-42.7   1st Qu.:1.344   1st Qu.:5099   yes: 3712
##  Median :-41.8   Median :4.857   Median :5191
##  Mean   :-40.5   Mean   :3.621   Mean   :5167
##  3rd Qu.:-36.4   3rd Qu.:4.961   3rd Qu.:5228
##  Max.   :-26.9   Max.   :5.045   Max.   :5228
##
```

It is remarkable that at simple visual inspection this split of original dataset does not differ much from the original one.

Finally, now that bankraw and validation datasets were created, I split bankraw in train_set and test_set applying once again Pareto's law, and I will use theses train_set and test_set to train different models and test their performance over different metrics prior to decide which candidate model will be the final one. Once final model is chosen, and only at that point, validation dataset will be used.

2.3.Exploratory Analysis

At this point it is important to visualize the data we have, so I will plot first of all numeric variables and second of all factor variables within the training set.

It is important to note that a very simple and common form of statistical analysis is the bivariate analysis. In consists on comparing two variables, one against the other. In our case, it is a matter of comparing each independent variable against the dependent variable "Y" we want to predict.

Let's start with the histograms for numeric variables splitting by dependent variable. This will allow us to get a first impression on those numeric variables distributions and their influence on the final result. Figures 1 to 10 will show this analysis:

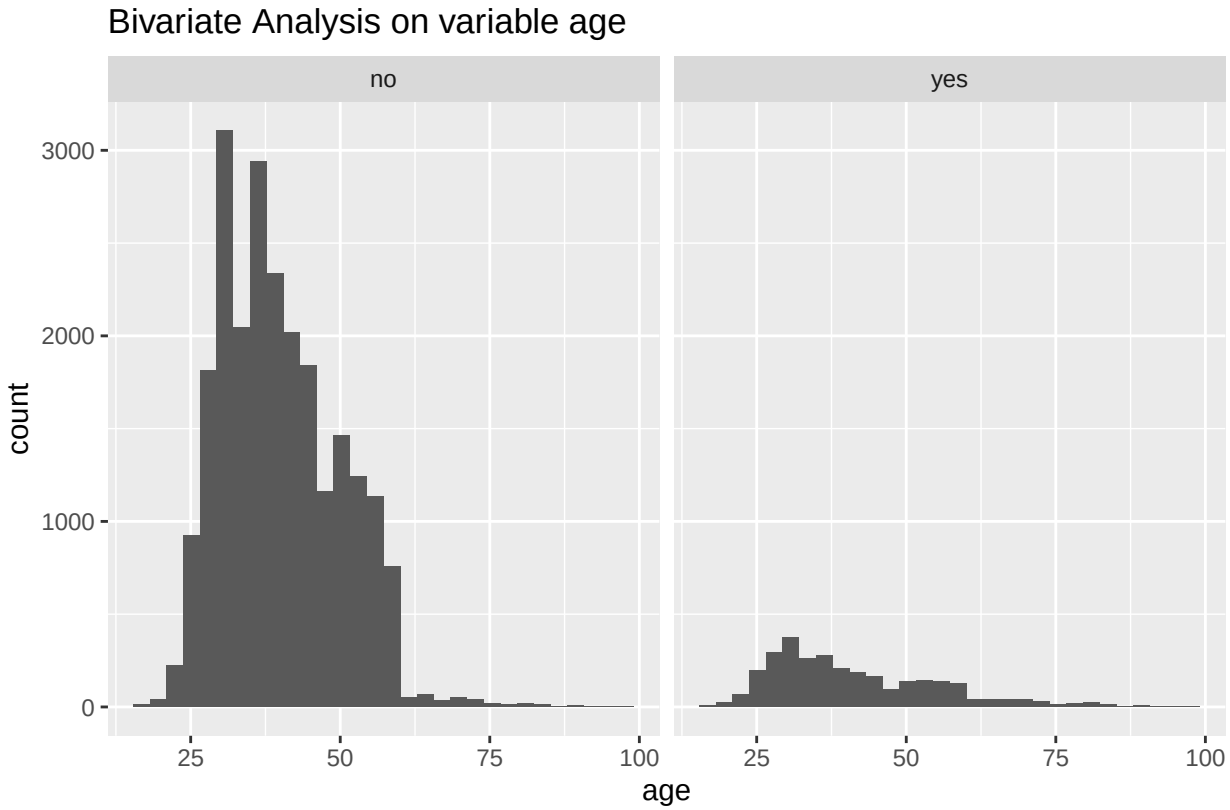
Bivariate Analysis on variable age
no
yes
3000
2000
1000
0
count
25
50
75
100
age

Figure 1

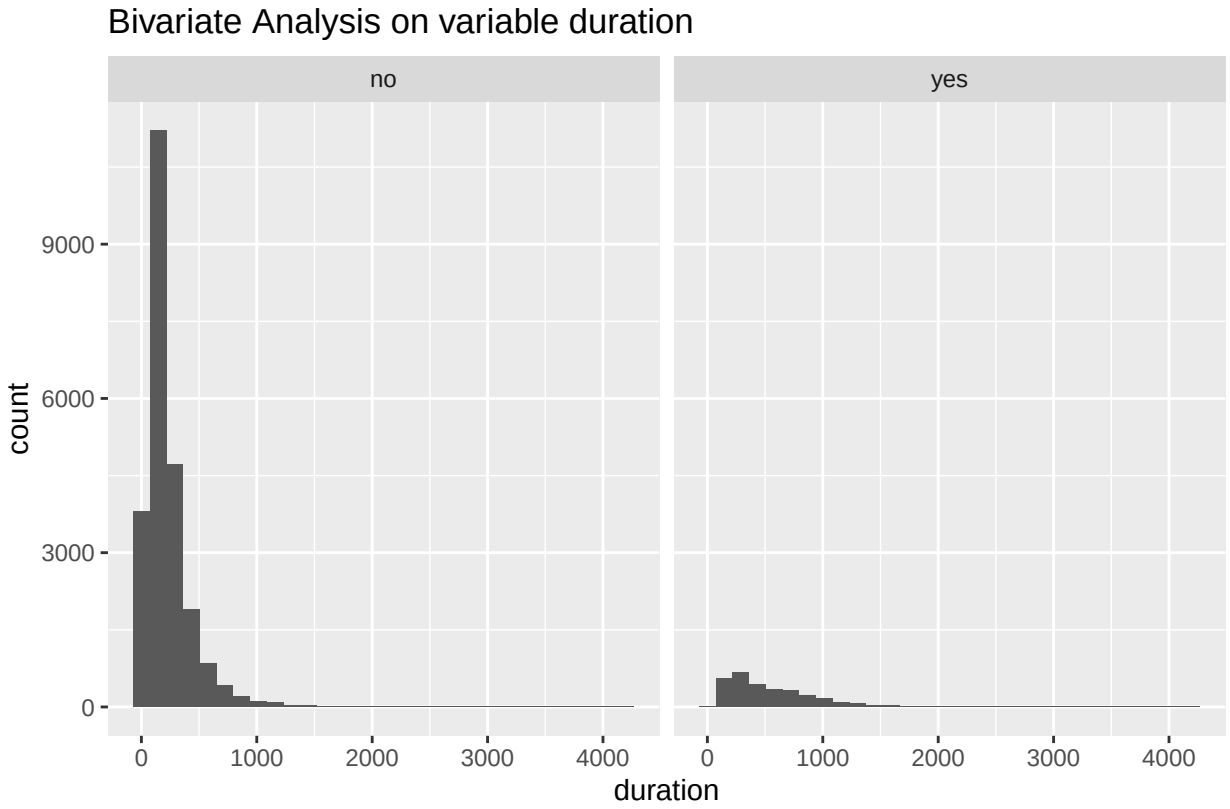
Bivariate Analysis on variable duration
no
yes
count
9000
6000
3000
0
0
1000
2000
3000
4000
duration

Figure 2

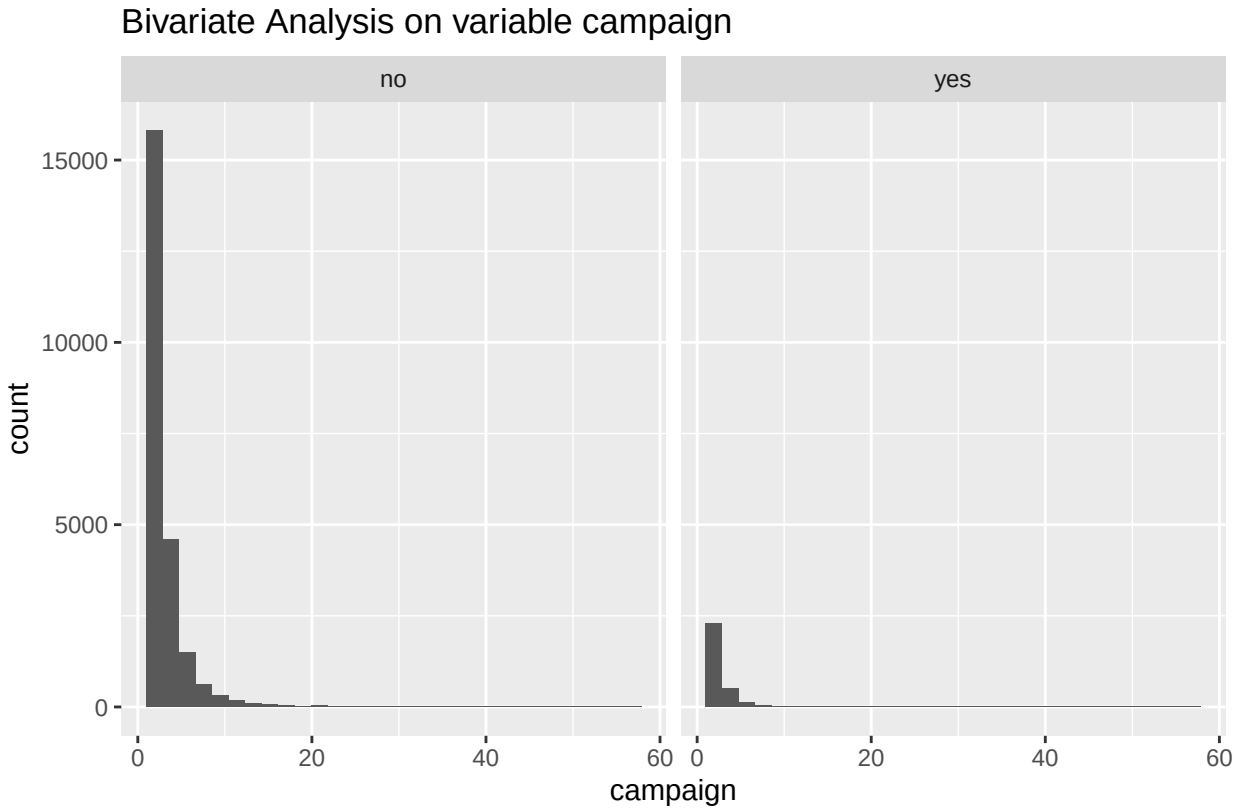
Bivariate Analysis on variable campaign
no
yes
15000
10000
5000
0
count
0
20
40
60
campaign

Figure 3

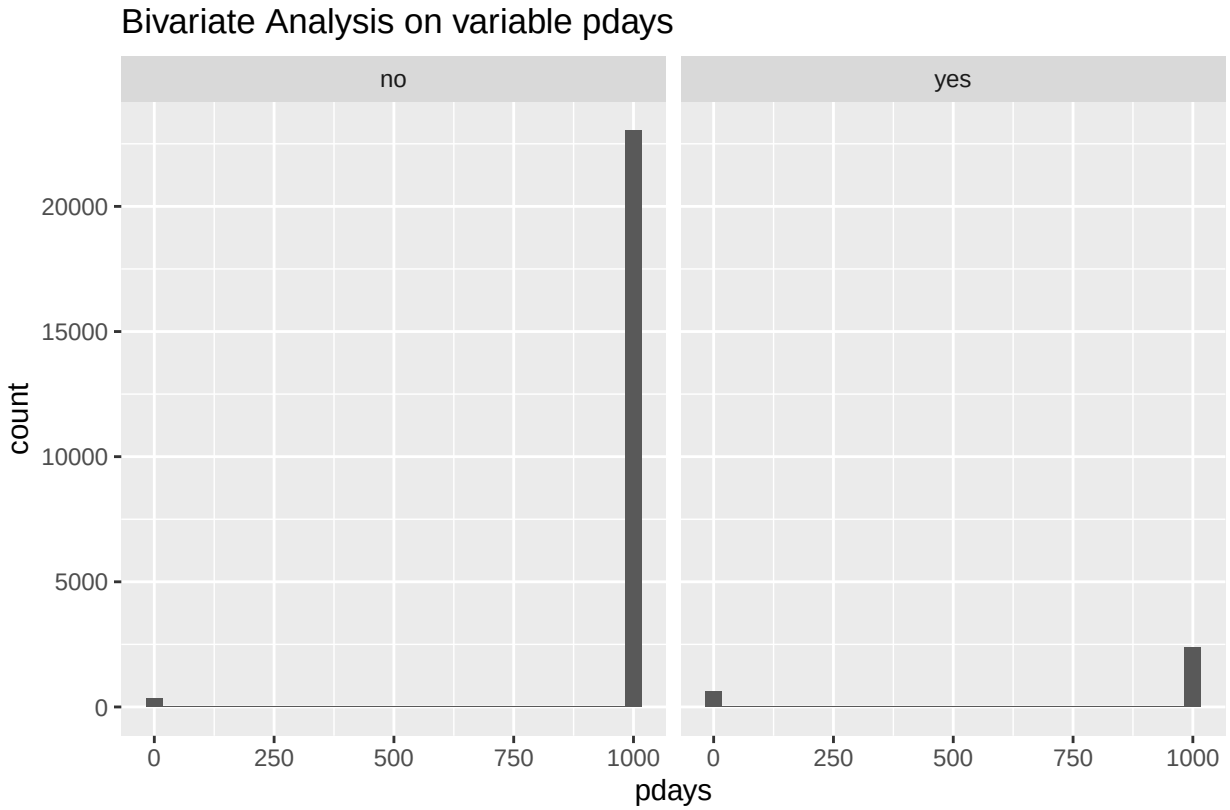
Bivariate Analysis on variable pdays
no
yes
count
20000
15000
10000
5000
0
0
250
500
750
1000
pdays

Figure 4

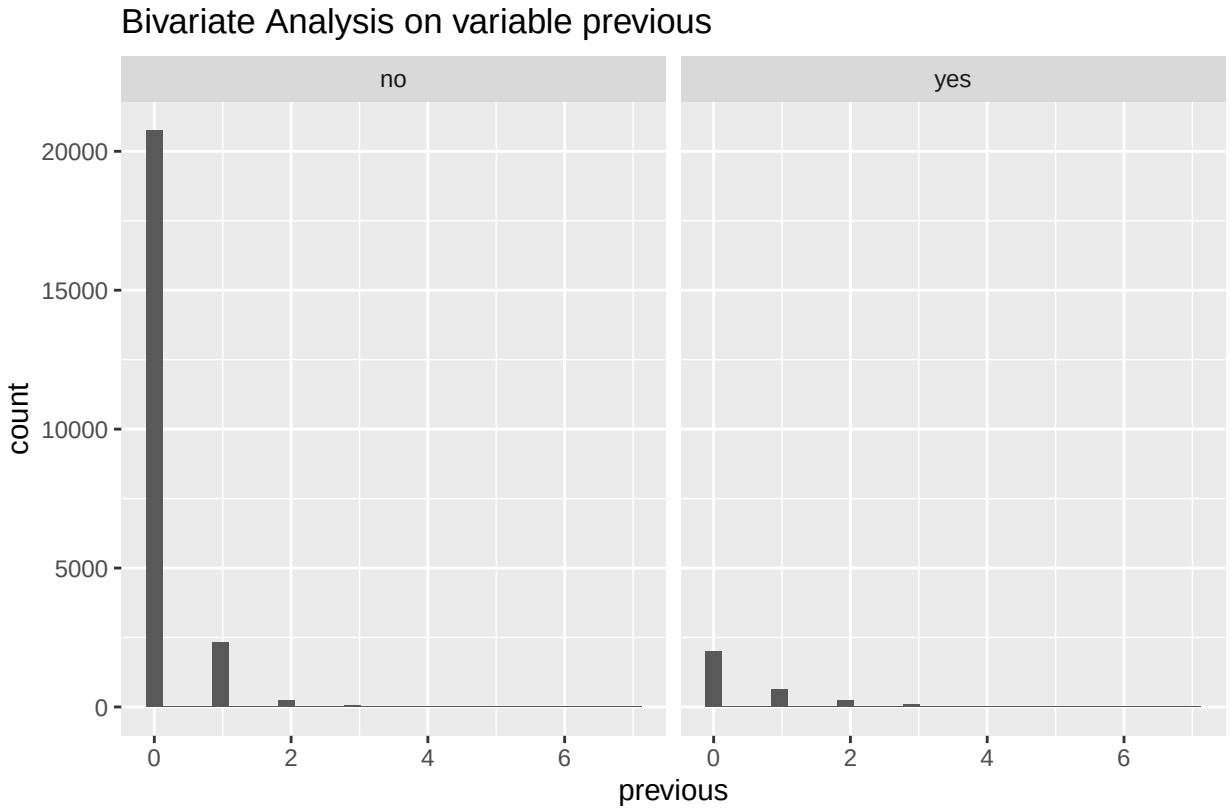
Bivariate Analysis on variable previous
no
yes
20000
15000
10000
5000
0
count
0
2
4
6
previous

Figure 5

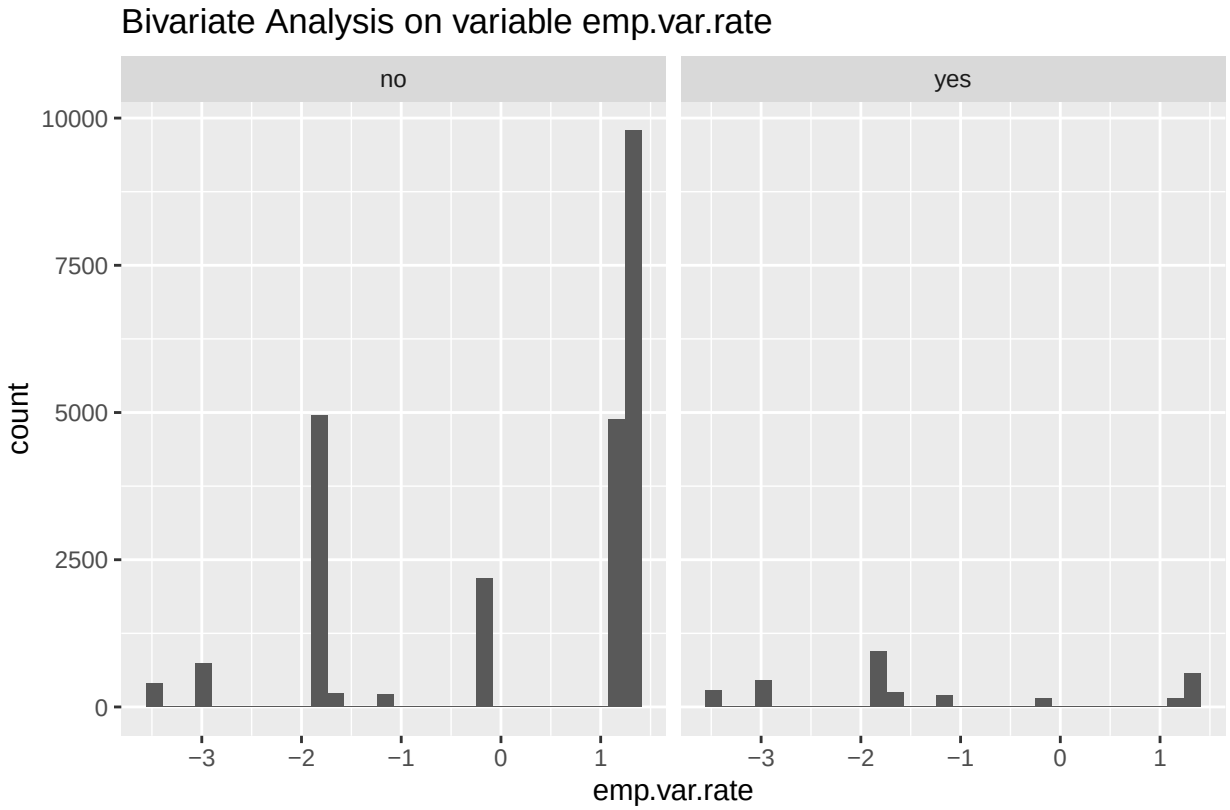
Bivariate Analysis on variable emp.var.rate
no
yes
10000
7500
5000
2500
0
count
−3
−2
−1
0
1
emp.var.rate

Figure 6

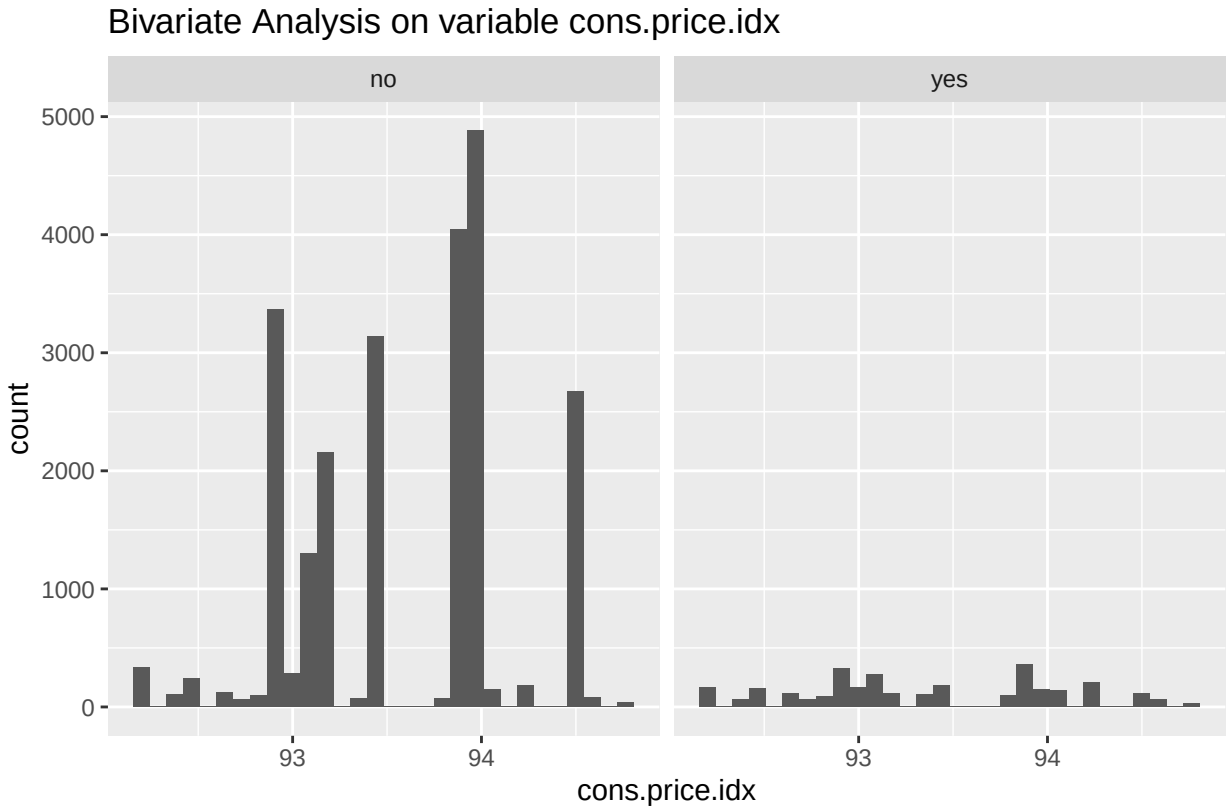
Bivariate Analysis on variable cons.price.idx
no
yes
5000
4000
3000
2000
1000
0
count
93
94
93
94
cons.price.idx

Figure 7

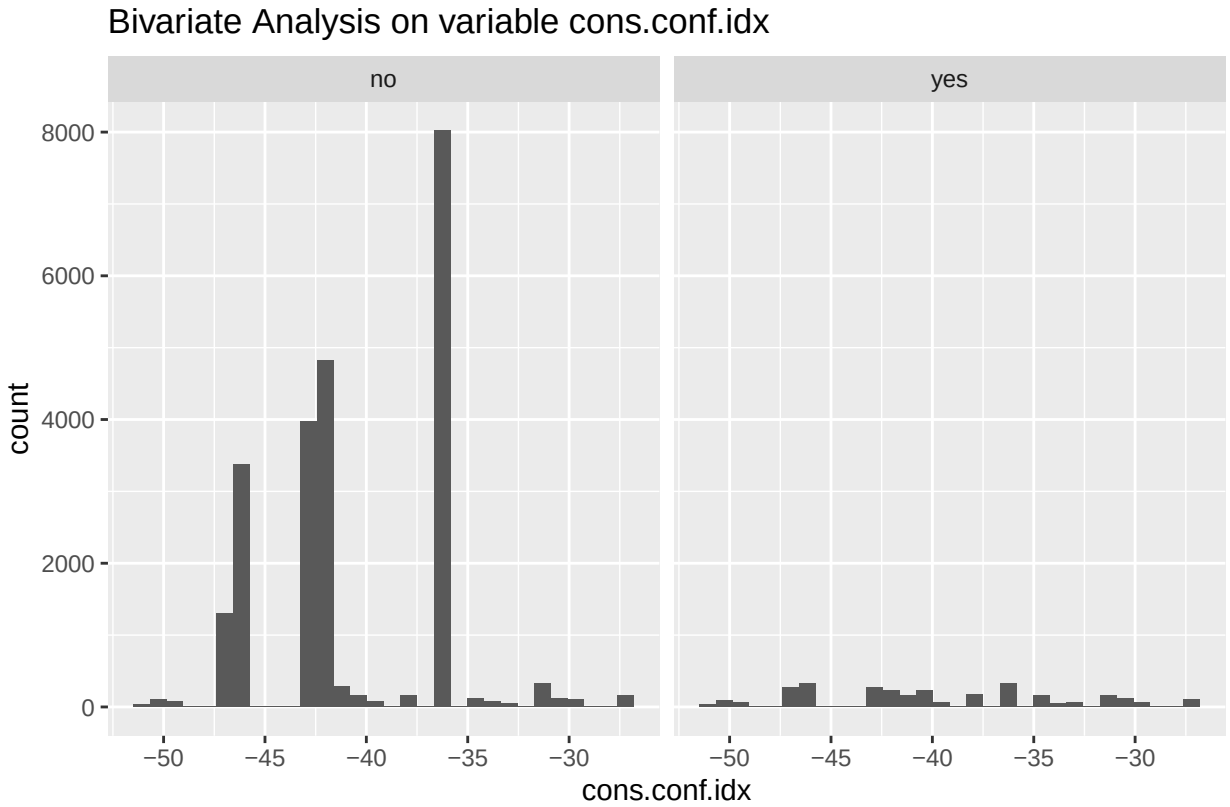
Bivariate Analysis on variable cons.conf.idx
no
yes
8000
6000
4000
2000
0
count
−50
−45
−40
−35
−30
cons.conf.idx

Figure 8

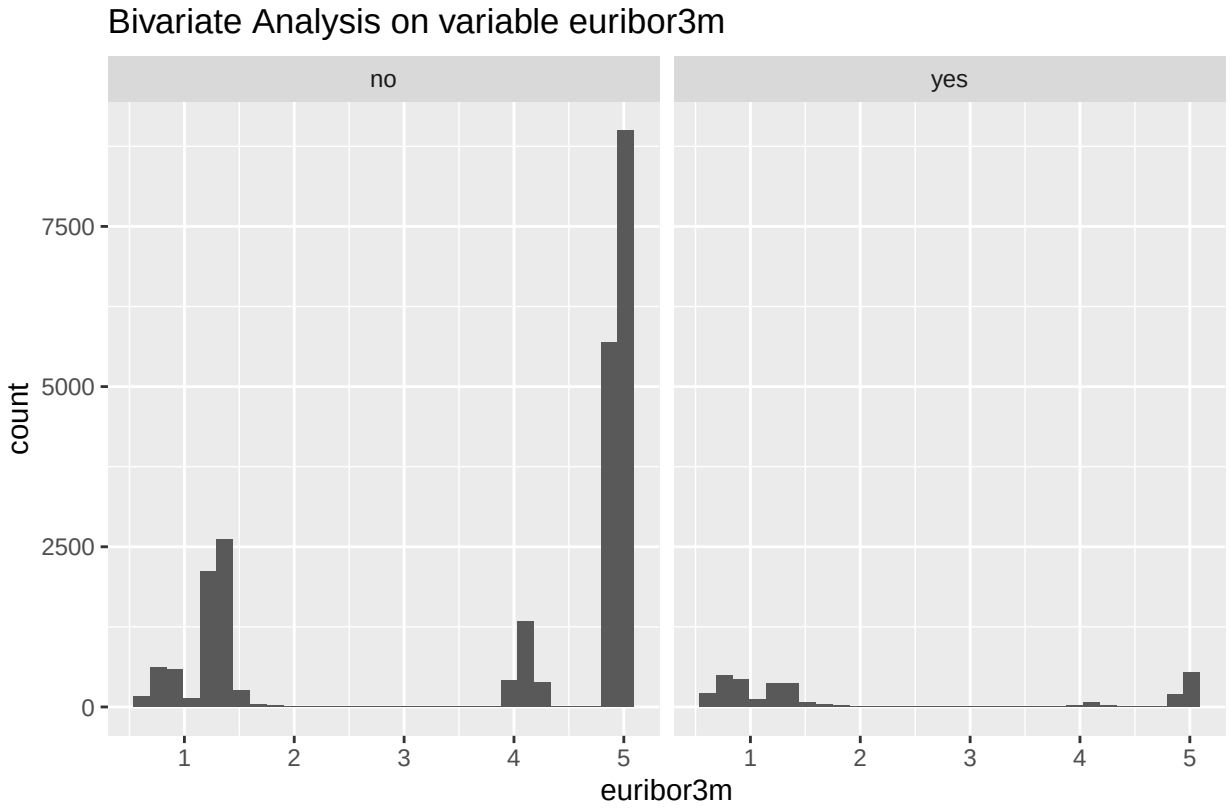
Bivariate Analysis on variable euribor3m
no
yes
count
7500
5000
2500
0
1
2
3
4
5
euribor3m

Figure 9

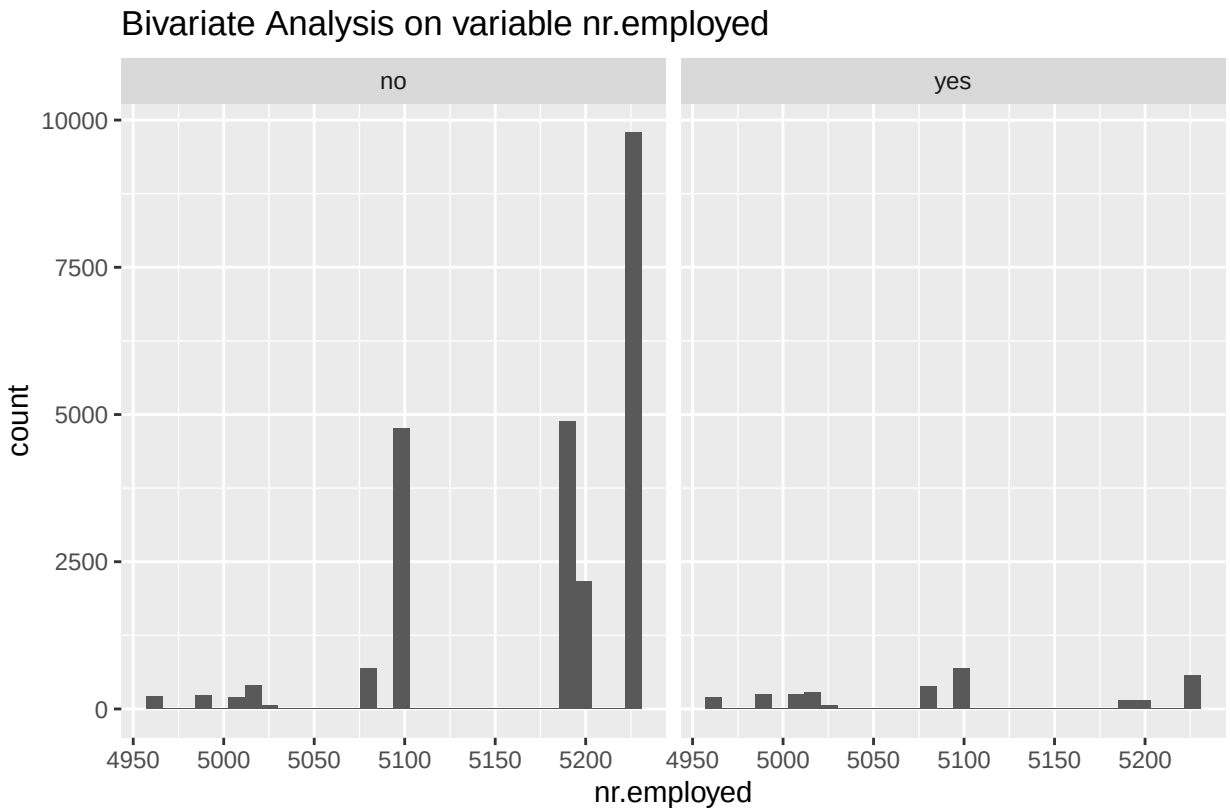

Figure 10

Now bivariate analysis has been performed on numeric independent variables, let's also apply it to discrete independent variables.

Figures 11 to 20 will show the bivariate analysis on categorical variables.

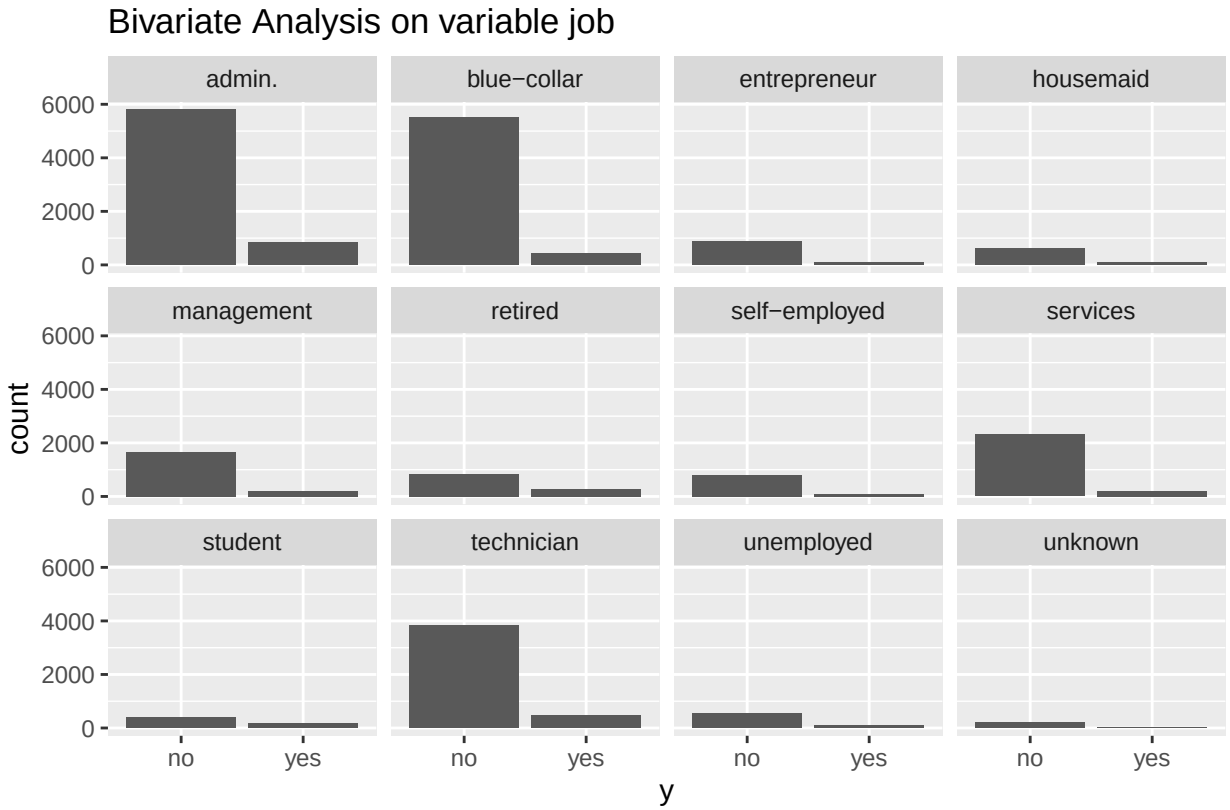
Bivariate Analysis on variable job
admin.
blue−collar
entrepreneur
housemaid
management
retired
self−employed
services
student
technician
unemployed
unknown
count
6000
4000
2000
0
no
yes
y

Figure 11

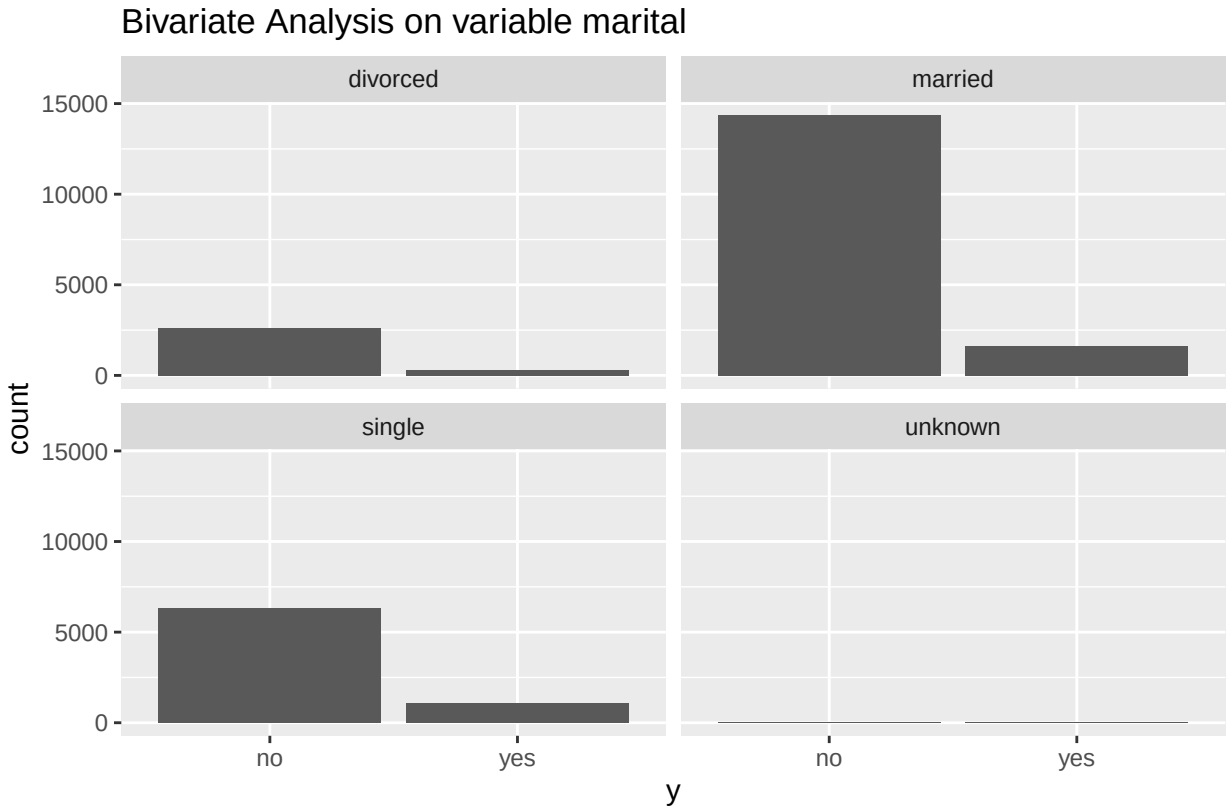
Bivariate Analysis on variable marital
divorced
married
single
unknown
count
15000
10000
5000
0
no
yes
y

Figure 12

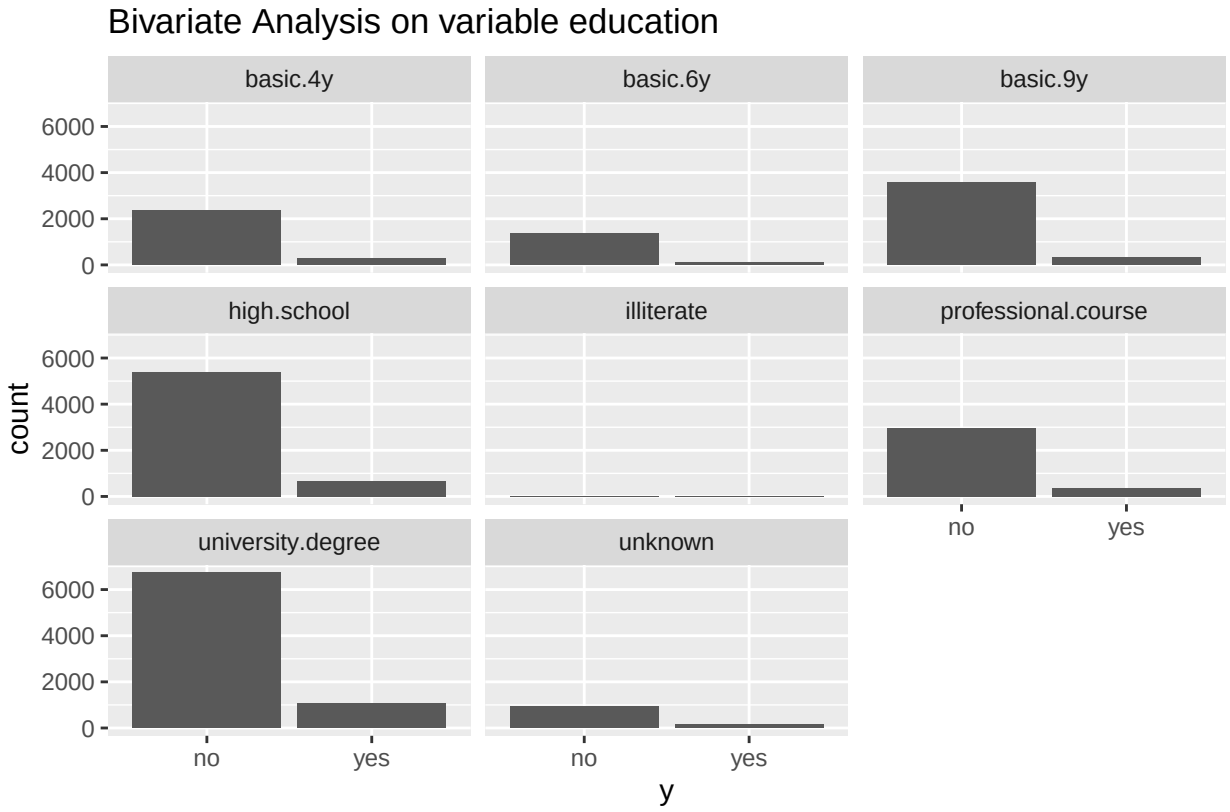
Bivariate Analysis on variable education
basic.4y
basic.6y
basic.9y
high.school
illiterate
professional.course
university.degree
unknown
count
6000
4000
2000
0
no
yes
y

Figure 13

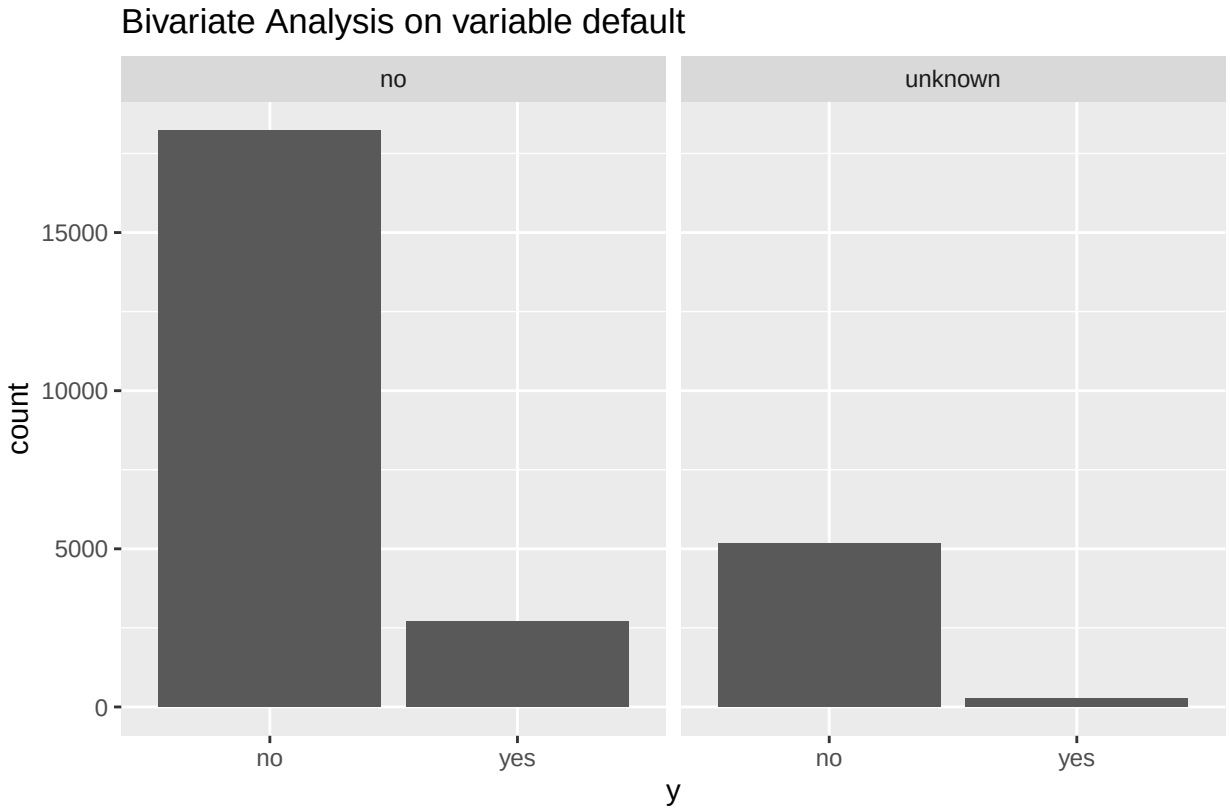
Bivariate Analysis on variable default
no
unknown
15000
10000
5000
0
count
no
yes
no
yes
y

Figure 14

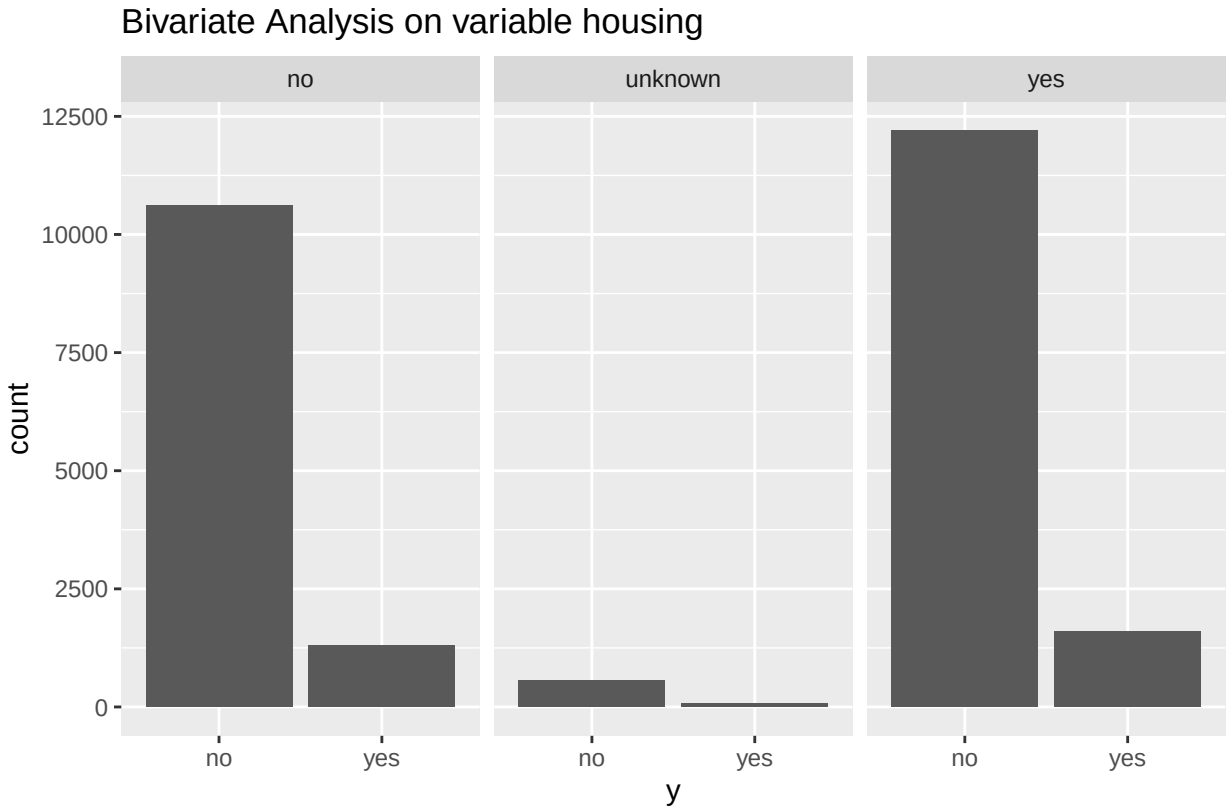
Bivariate Analysis on variable housing
no
unknown
yes
12500
10000
7500
5000
2500
0
count
no
yes
no
yes
no
yes
y

Figure 15

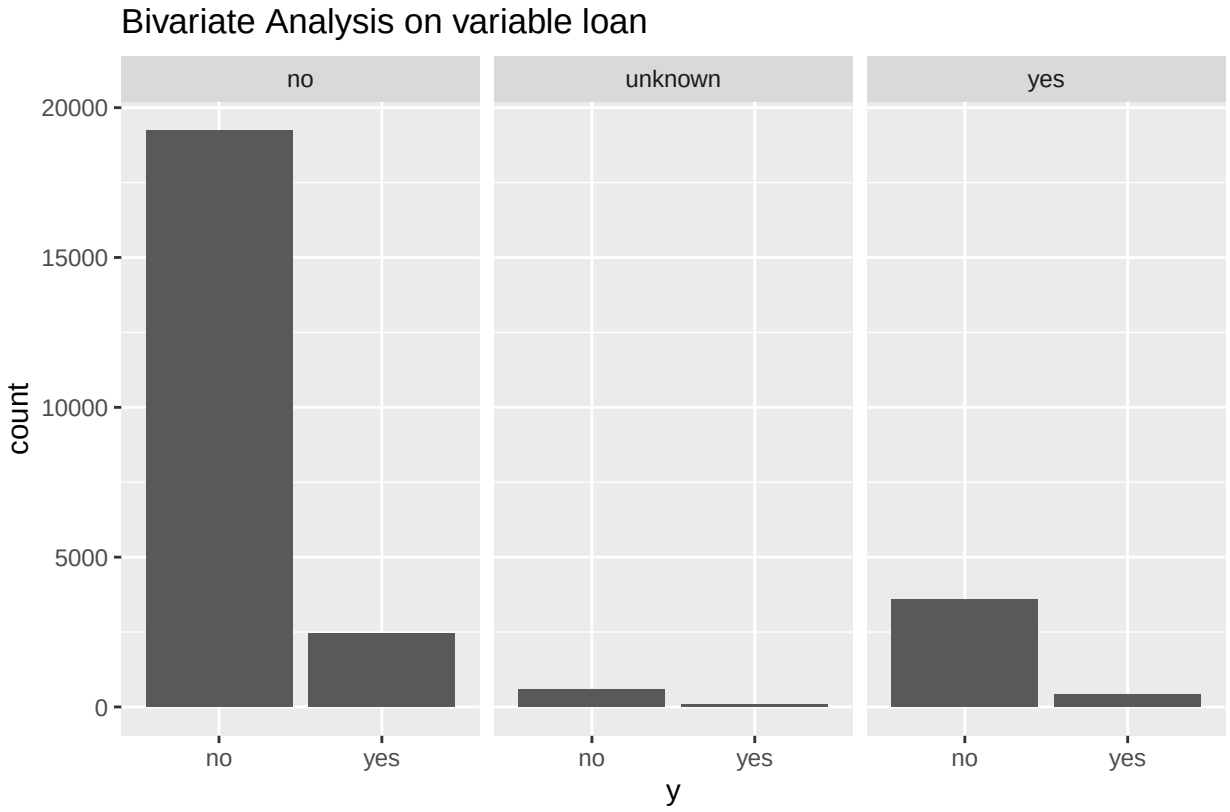
Bivariate Analysis on variable loan
no
unknown
yes
20000
15000
10000
5000
0
count
no
yes
no
yes
no
yes
y

Figure 16

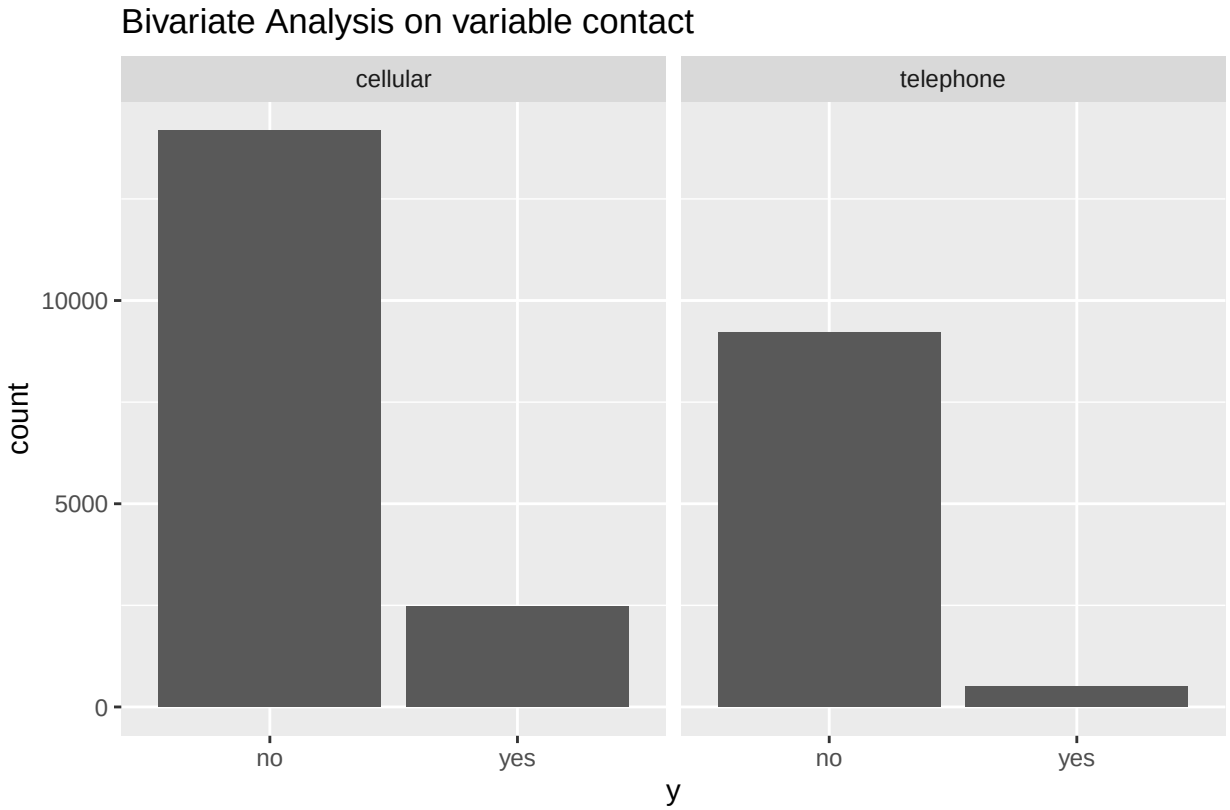
Bivariate Analysis on variable contact
cellular
telephone
10000
5000
0
count
no
yes
no
yes
y

Figure 17

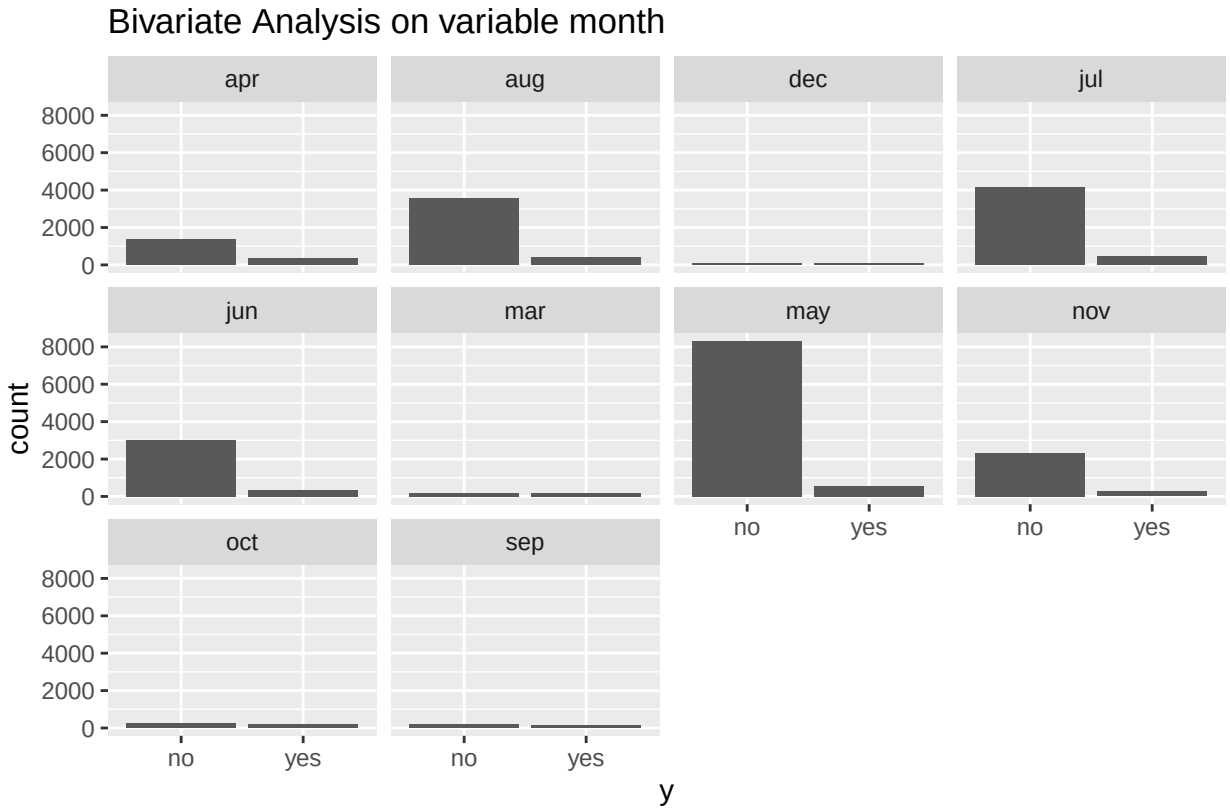
Bivariate Analysis on variable month
apr
aug
dec
jul
jun
mar
may
nov
oct
sep
count
8000
6000
4000
2000
0
no
yes
y

Figure 18

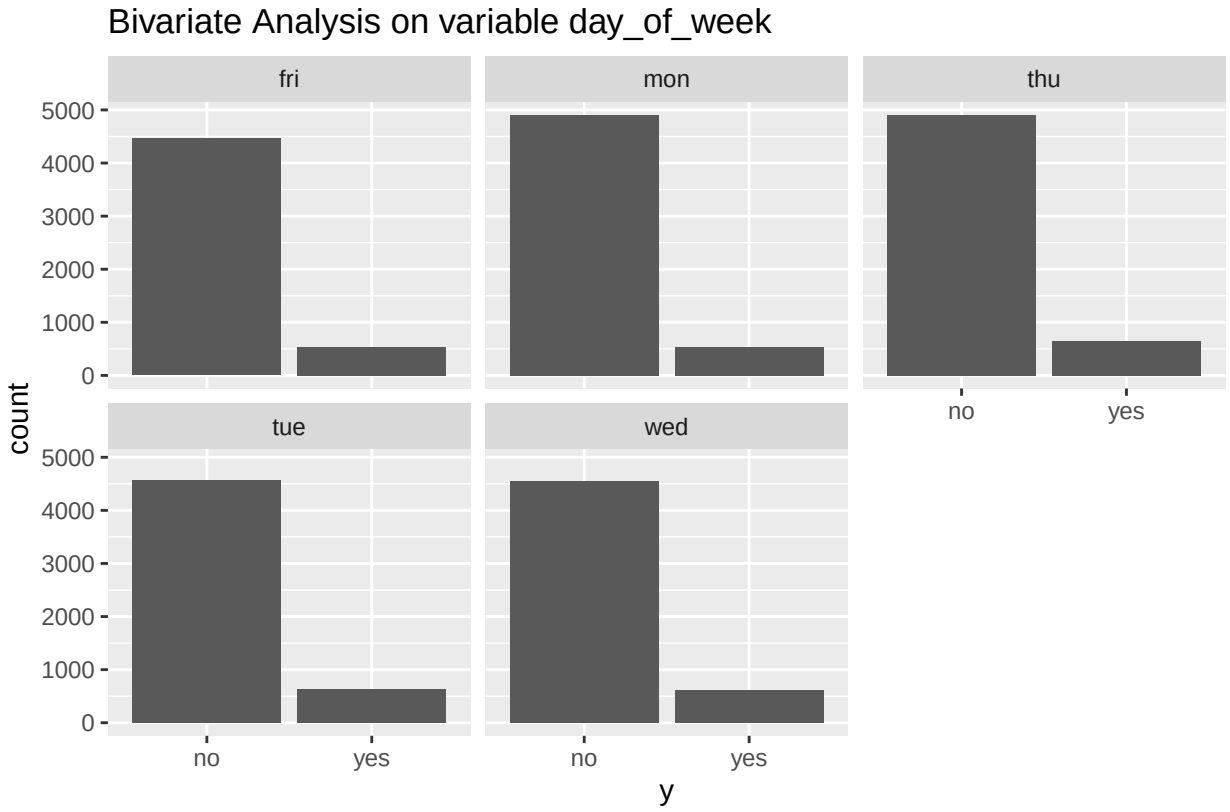
Bivariate Analysis on variable day_of_week
fri
mon
thu
tue
wed
count
5000
4000
3000
2000
1000
0
no
yes
y

Figure 19

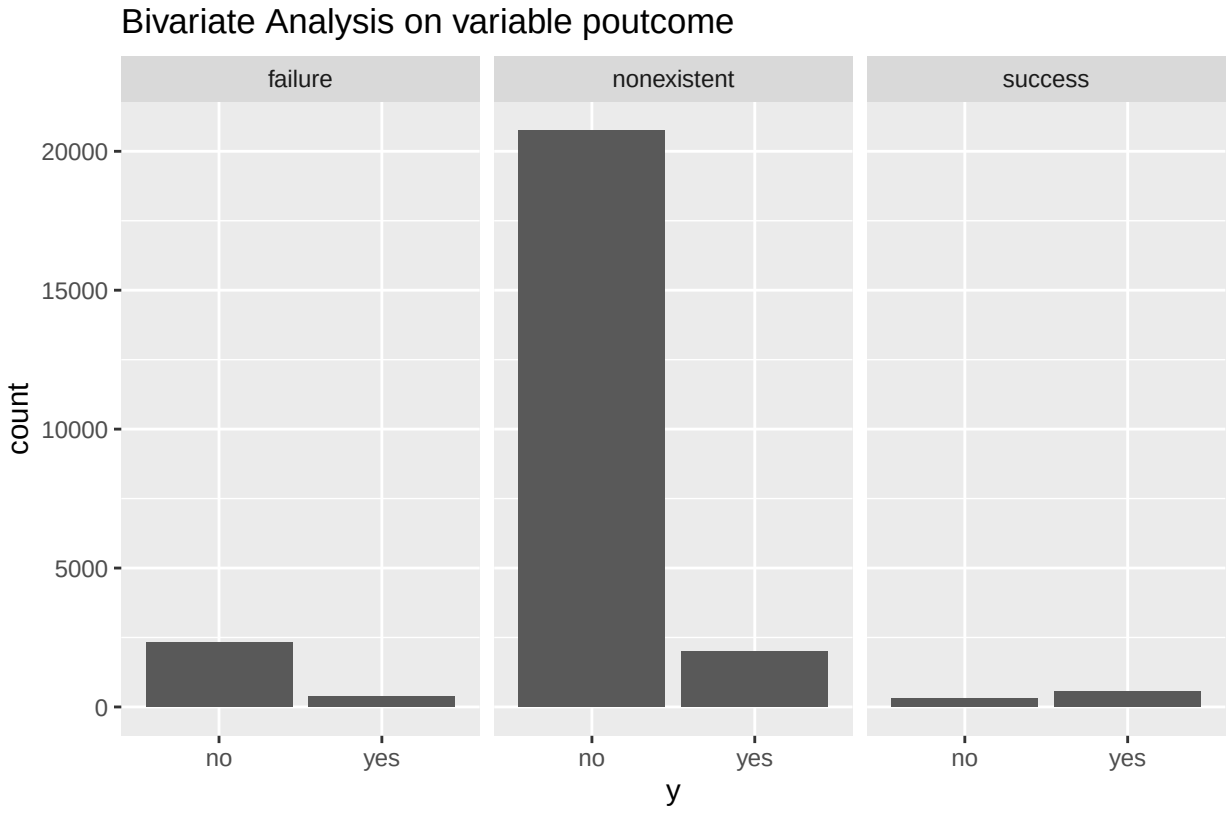

Figure 20

Bivariate analysis is a simple case of multivariate analysis, instead of comparing variables in pairs, multiple variables are compared against each other at the same time. This may allow us to identify prediction power, which variables are providing same kind of information and finally allow us to choose if all variables make sense within the model or on the other hand some of them can de disregarded.

In order to perform this multivariate analysis, let's analyze Pearson correlation between numeric variables and dependent variable "Y"

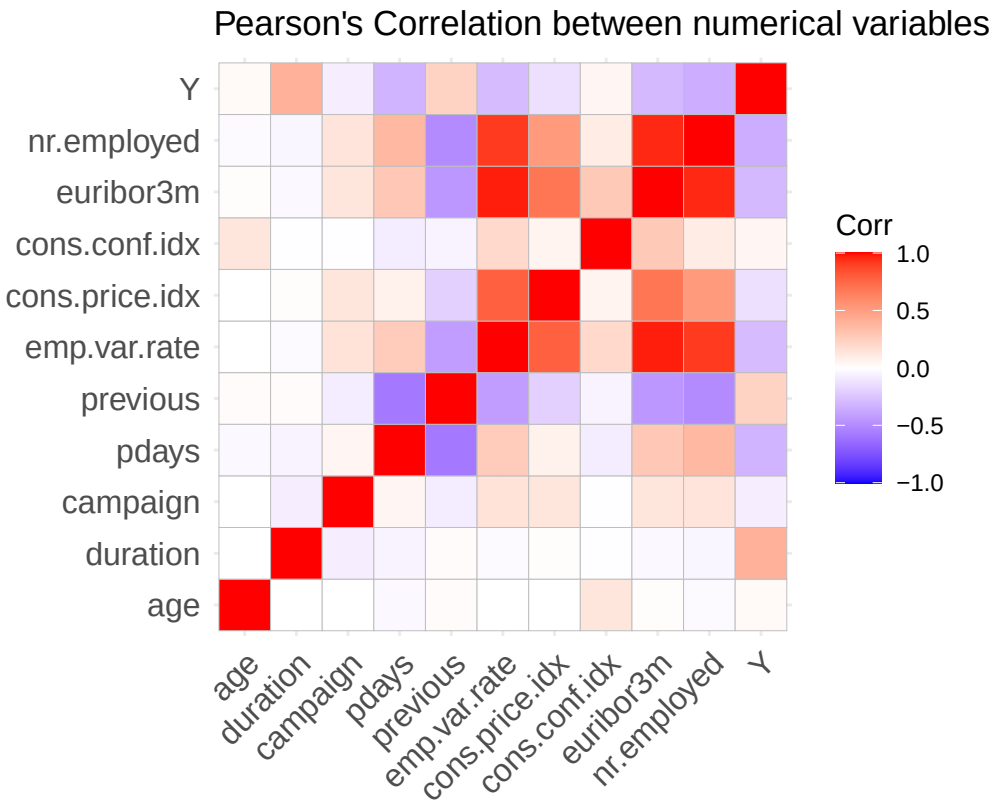

Figure 21

Now, let's do the same for categorical variables, although here some pieces of extra work will be needed. Where a categorical variable has more than two categories, it can be represented by a set of binary variables, with one variable for each category. In R, there are several ways to perform this task, being one of them the package fastDummies. Function dummy_cols on this package will allow to "binarize" discrete variables.

Once discrete variables have been "binarized", let's perform also Pearson's correlation. For simplicity's sake, due to high number of dummy variables created, instead of plotting all together, I have plotted them in two halves.

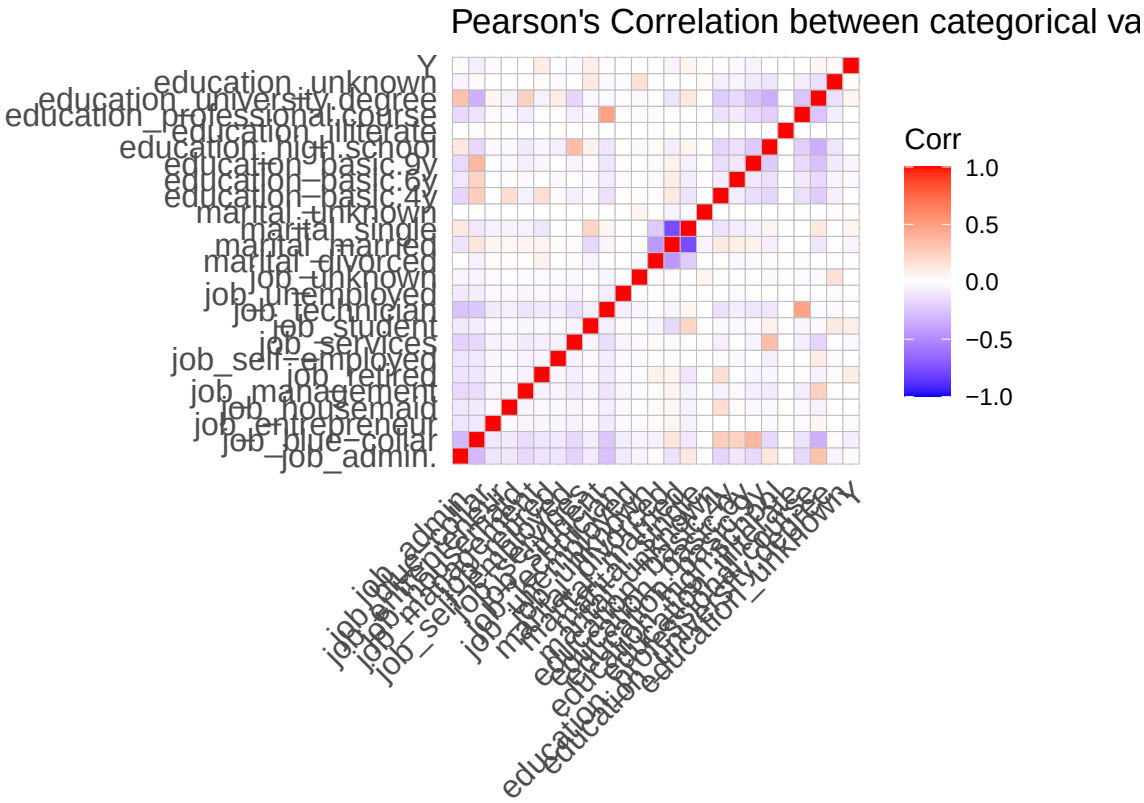
Pearson's Correlation between categorical va
Y
education_unknown
education_university.degree
education_professional.course
education_illiterate
education_high.school
education_basic.9y
education_basic.6y
education_basic.4y
marital_unknown
marital_single
marital_married
marital_divorced
job_unknown
job_unemployed
job_technician
job_student
job_services
job_self-employed
job_retired
job_management
job_housemaid
job_entrepreneur
job_blue-collar
job_admin.
Corr
1.0
0.5
0.0
−0.5
−1.0

Figure 23

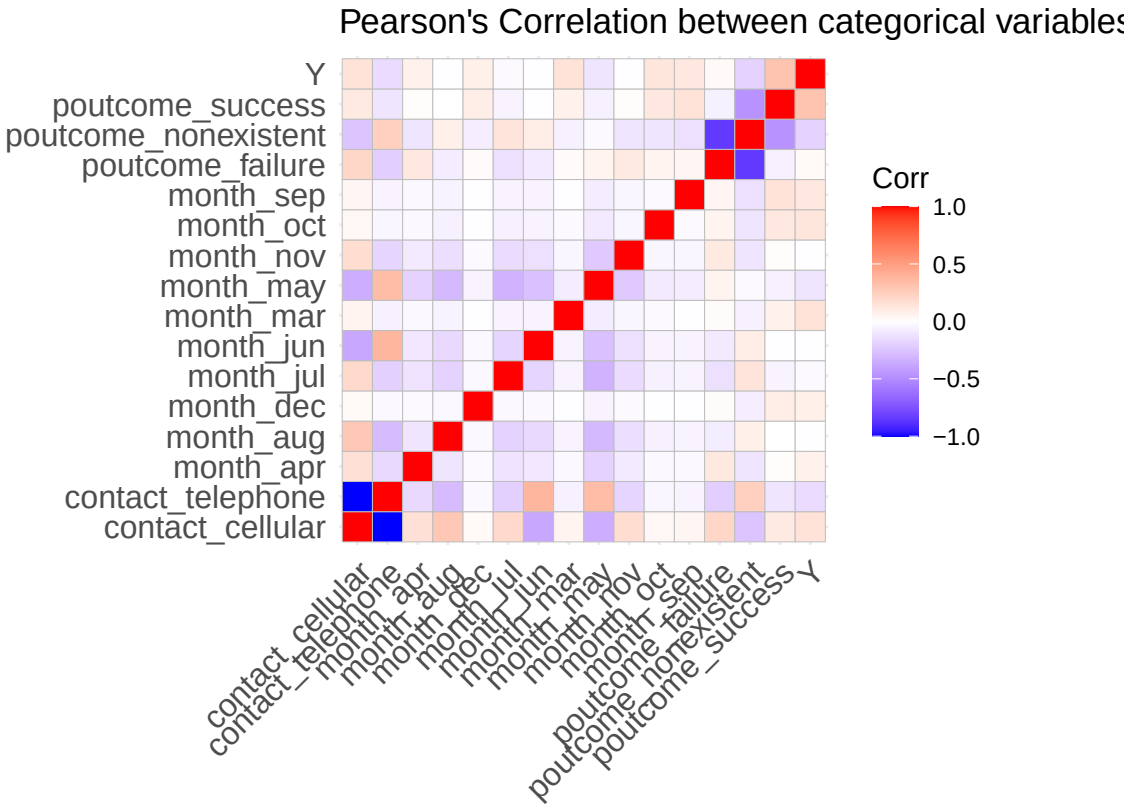

Figure 24

At this point some relevant information has been unveiled.Some variables seem to be more relevant regarding "Y" than others. Hence following variables will be the candidate ones to take part in my models:

```
## [1] "duration"    "pdays"       "euribor3m"   "nr.employed" "poutcome"
```

2.4 Modelling

As previously stated the final objective is building a classification system to predict if the client would subscribe a term deposit or not. Supervised learning classification algorithms learn from labeled data, in our case the bankraw dataset we have already described, with the "y" label being a binary variable, so we are facing the most simple classification problem.

Nowadays there are several well-known supervised learning classification algorithms such as logistic regression, k nearest neighbors, decision trees, random forest, support vector machines or neuronal networks, having all of them different pros and cons.

On their paper (S. Moro and Rita, 2010), they compared logistic regression, decision trees, support vector machines and neuronal networks finding that neuronal networks performed better than others. In this project, due to computational resources needed for running support vector machines and neuronal networks, I have discarded them focusing only on logistic regression, k nearest neighbors, decision trees and random forest.

Besides, following approach taken in my previous project ML_RVV_Capstone_Project I have also included a baseline predictor to set as reference.

Finally, some decisions regarding primary metrics used to choose candidates models will also be necessary and discussed later.

2.4.1 BASELINE PREDICTORS

Baselines for predictive models or baseline predictors are performance evaluations for given problems in statistics. Baselines are commonly used as the first approach and the limit that should at least be reached.

In doing so, the first baseline stage will be the naive one, just only assuming results will always be no.

$$\hat{Y} = 0$$

being $\hat{Y}$ the predicted result, with 0 as no suscription and 1 as suscription.

As already mentioned earlier, 88.7363% of answers are no in train set which explains why naive model has 0.8872705 accuracy. Hence prevalency is an issue within our original dataset and accuracy is not the best metric to consider. In order to mitigate prevalence as much as possible I will also explore balance accuracy, sensitivity and specificity for all candidate models.

In doing so following results are obtained:

method	Accuracy	Balanced_Accuracy	Sensitivity	Specificity
Naive	0.8872705	0.5	1	0

As observed in table above accuracy and sensitivity are reasonably good but balanced accuracy and specificity are not good at all as it can be expected due to non uniform distribution in our datasets.

2.4.2 LOGISTIC REGRESSION

Logistic regression is a specific case of a set of generalized linear models (glm) and it extends the regression approach to categorical data. It measures the relationship between the categorical dependent variable and one or more independent variables by estimating probabilities using a logistic function. The logistic regression can be understood simply as finding the β parameters that best fit:

$$\hat{Y} = \begin{cases} 1 & \beta_0 + \beta_1 x + \varepsilon > 0 \\ 0 & \text{else} \end{cases}$$

Applied to our data using train function in caret package with glm method, following results are obtained:

method	Accuracy	Balanced_Accuracy	Sensitivity	Specificity
Naive	0.8872705	0.5000000	1.0000000	0.0000000
Logistic Regression	0.9094219	0.6716814	0.9786252	0.3647376

Despite not improving to much the accuracy and worsening the sensitivity, as expected, specificity, without being good enough yet, has improved dramatically and balanced accuracy also improved 0.3433627 %

At this point, as expected, it is obvious that logistic regression is a better choice than naive one.

2.4.3 K NEAREST NEIGHBOURS

Main idea behind k nearest neighbors, as in all machine learning algorithms, will be using the five features already selected to estimate the conditional probability function:

$$p(x_1, x_2, x_3, x_4, x_5) = Pr(Y = 1 \mid X_1 = x_1, X_2 = x_2, X_3 = x_3, X_4 = x_4, X_5 = x_5)$$

First it is defined the distance between all observations based on the features. Then, for any point $(x_1, x_2, x_3, x_4, x_5)$ for which we want an estimate of $p(x_1, x_2, x_3, x_4, x_5)$, we look for the k nearest points to it and then take an average of the 0s and 1s associated with these points. The set of points used to compute this average is called the neighborhood. Hence we are able to obtain $\widehat{p}(x_1, x_2, x_3, x_4, x_5)$. In order to control the flexibility of the estimation, the only hyperparameter to fine tune in this algorithm is k

Previous steps can be done automatically in R using train function in caret package with knn method. Hyperparameter k is controlled with tuneGrid function and through trainControl function, cross validation can be controlled also.

Hence, using 10-fold cross validation, fitting 10 versions of kNN to 9 bootstrapped samples, following results are obtained:

method	Accuracy	Balanced_Accuracy	Sensitivity	Specificity
Naive	0.8872705	0.5000000	1.0000000	0.0000000
Logistic Regression	0.9094219	0.6716814	0.9786252	0.3647376
K Nearest Neighbors	0.9115460	0.7192868	0.9675103	0.4710633

Results trend observed in logistic regression model keeps also for K nearest neighbor model, thus worsening sensitivity, but improving accuracy, balanced accuracy and specificity.

2.4.4 DECISION TREES

A decision tree is a flowchart-like structure in which each internal node represents a "test" on an attribute, each branch represents the outcome of the test, and each leaf node represents a class label (decision taken after computing all attributes). The paths from root to leaf represent classification rules.

Hence decision trees clasifiers can be described as the combination of mathematical and computational techniques to categorize a given set of data.

In our case data appears as follows:

$$(x_1, x_2, x_3, x_4, x_5, Y)$$

where Y is the dependent variable and features $(x_1, x_2, x_3, x_4, x_5)$ already mentioned in previous models the ones to be partitioned to classify the dependent variable.

Due to their simplicity, and the fact of showing powerful insights via visual inspection, decision trees are among the most popular machine learning algorithms.

In R, decision trees can be managed via rpart function, included in rpart package, being complexity parameter (cp), minsplit and minbucket the hyperparameters to fine tune.

In this case I have tested 25 values ranging from 0 to 0,05 for complexity parameter value and left default minsplit value (at least 20) and default minbucket value obtaining following results:

method	Accuracy	Balanced_Accuracy	Sensitivity	Specificity
Naive	0.8872705	0.5000000	1.0000000	0.0000000
Logistic Regression	0.9094219	0.6716814	0.9786252	0.3647376
K Nearest Neighbors	0.9115460	0.7192868	0.9675103	0.4710633
Decision Trees	0.9173115	0.7407466	0.9687073	0.5127860

As observed once more time balanced accuracy and specificity improves, but this time neither sensitivity nor accuracy worsen, so this is our first model overperforming remaining ones.

2.4.5 RANDOM FOREST

Finally, let's try random forest. Random forest are ensemble of trees. The idea is selecting many predictors, by choosing different decision trees randomly picked over different features each time, and afterwards ensembling those predictors to build the final one. In our case for every observation in the test set, form a prediction $\hat{Y_j}$ using a different tree T_j with j ϵ (1,B) each one of the random trees and finally taking most frequent class among $\hat{T_1}$, ..., $\hat{T_B}$

Random forest in R can be handled in several ways, one of them, still on caret package with rf method. To control the number of variables randomly sampled as candidates at each split we can use mtry, to control the minimum size of terminal nodes we can use nodesize.

I have chosen mtry value as 2 and looked for the best nodesize among 11,21 and 31 obtaining following results:

method	Accuracy	Balanced_Accuracy	Sensitivity	Specificity
Naive	0.8872705	0.5000000	1.0000000	0.0000000
Logistic Regression	0.9094219	0.6716814	0.9786252	0.3647376
K Nearest Neighbors	0.9115460	0.7192868	0.9675103	0.4710633
Decision Trees	0.9173115	0.7407466	0.9687073	0.5127860
Random Forest	0.9159460	0.7147169	0.9745212	0.4549125

Computational time took longer than decision trees and hyperparameters are not perfectly fine tuned as it can be observed from results above since all metrics but sensitivity have worsen.

3.Results

After comparing results obtained by baseline model, logistic regression, k nearest neighbors, decision trees and random forest, decision trees are the most promising ones. Intuitively, I could expect better results from random forest but computational time taken did not help while trying to optimize hyperparameters, so this could explain the poorer performance.

For this reason and due to visual interpretation is easily achievable by human eye as it can be appreciated below in Figure 24, I have chosen decision trees as my final model.

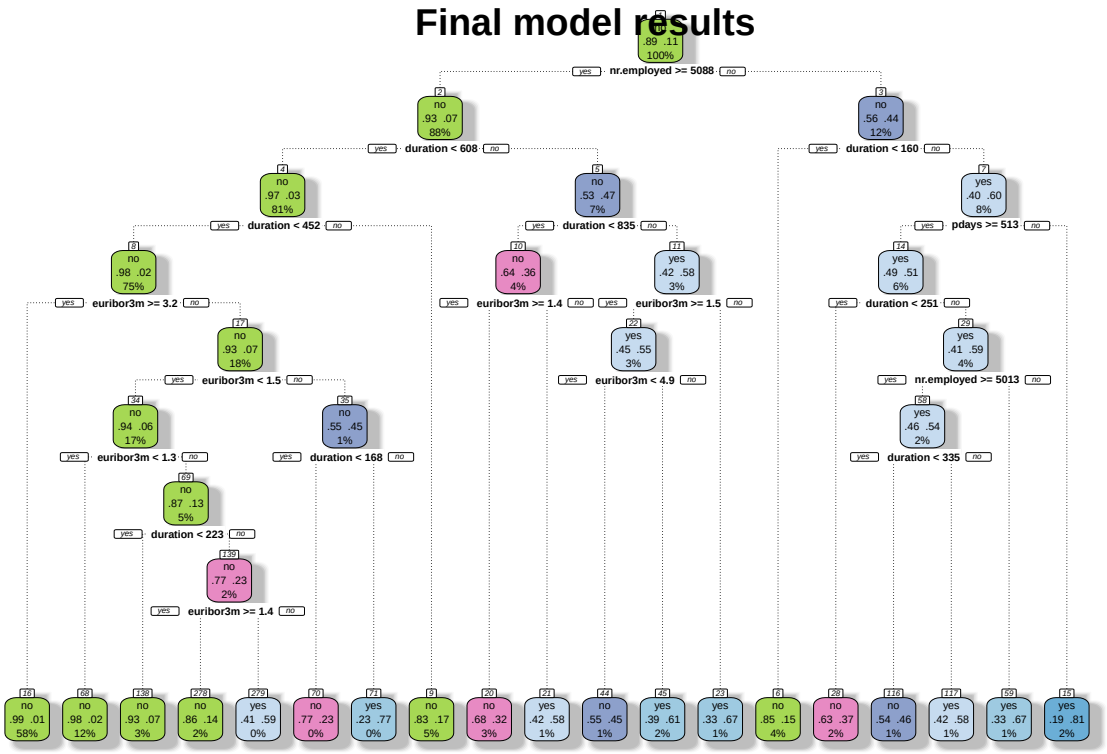

Figure 24

Applying this model to validation dataset following results are obtained:

method	Accuracy	Balanced_Accuracy	Sensitivity	Specificity
Decision Trees	0.9123574	0.7441128	0.9612859	0.5269397

4.Conclusions

Binary classification is a frequent problem faced in many disciplines among science and real life. Supervised learning whenever enough historical data is collected offers a wide range of different algorithms to help us dealing with these problems and take better decisions.

I have tested logistic regression, k nearest neighbors, decision trees and random forest for a binary classification within a non uniformly distributed dataset where prevalence is indeed a challenging issue.

Despite the fact of prevalence, decision tree built allowed to obtain good accuracy and sensitivity and reasonable balanced accuracy and almost acceptable specificity. Apart from being significantly comprehensible, as shown in previous section, being able to classify in advance the results of your campaign allows the stakeholder to discriminate which sector of population focus on to get more revenues.

Nevertheless, some more optimization could have been addressed specially on random forest because results on random forest have been far from expected ones a priori. Some extra analysis could also have been performed, specially due to imbalance nature of source dataset. Hence, building several different new samples from original dataset with more percentage of cases of non dominant segment of population to see if specificity could improve more is a possible future work. Understand the behavior of ROC curves or Gini index could be another interesting idea to progress with.

Finnaly, testing support vector machines and neuronal networks and comparing them with these results to confirm (S. Moro and Rita, 2010) analysis are also another future possible ideas for future work.

5.Bibliography

Irizarry, R. (2021). Introduction to data science: Data analysis and prediction algorithms with R. https://rafalab.github.io/dsbook/.

S. Moro, P. C. and Rita, P. (2010). A data-driven approach to predict the success of bank telemarketing. http://dx.doi.org/10.1016/j.dss.2014.03.001.