Assignment 3

Mathieu NORDIN

April 2022

ST404

Contents

1 Executive Summary

Our main findings from the EDA section are that the proportion of people who subscribe is low (11%), most customers never have been contacted before and people who were contacted and/or who subscribed before have more chances to subscribe at the end of the call. Also, the number of called performed by employees is not constant in time. Some continuous variables (such as `previous`) were transformed into ordinal variables for statistical and explanatory reasons.

The final model was obtained using a method called backward stepwise regression. Our model is robust and relevant (this was check thoroughly using statistical tools such as residual plots, anova and cross-validation).

The model needs 10 different information to work and it misses 6% of the people who want to subscribe. Furthermore, when it predicts that people will subscribe, they turn out not subscribing half the times.

Finally, separation problems were encountered when producing too complex models and the method called lasso regression did not produce any results.

2 Statistical Methodology

2.1 Introduction

This report aims to present and derive a logistic regression model to help predict if customers are likely to subscribe to a bank term deposit. To this end, we are provided with a data set which consists of 41188 observations and 19 predictor variables. There are 9 continuous variables and 10 categorical variables and they fall into one of the following categories: personal information about the customer's demographic group and its financial background, marketing information about the previous and current contact or information about the economy at the time of the call. The response variable `subscribed` is a factor that takes values ”yes” or ”no”.

2.2 Explanatory Data Analysis (EDA)

Type	**Var.**	**Min**	**1st Qu.**	**Median**	**Mean**	**3rd Qu.**	**Max**
Personnal	age	17.0	32.0	38.0	40.0	47.0	98.0
Marketing detail	compaign	1.0	1.0	2.0	2.6	3.0	56.0
	pdays	0.0	999.0	999.0	962.5	999.0	999.0
Economic environment	ConConIdx	-50.8	-42.7	-41.8	-40.5	-36.4	-26.9
	ConPrIdx	92.2	93.1	93.8	93.6	94.0	94.8
	EmpVarRt	-3.4	-1.8	1.1	0.08	1.4	1.4
	Euribor3m	0.63	1.34	4.90	3.62	4.96	5.05
	nr_employed	4964	5099	5191	5167	5228	5228

Table 1: Summary statistics for continuous variables of the BankMarket Data set.

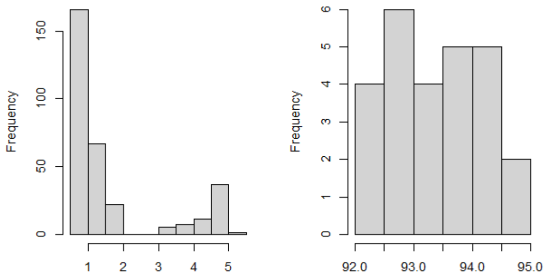

Figure 1: distribution of unique values of `euribor3m` (left) and `ConPrIdx` (right)

From the summary statistics, we notice that most people were not previously contacted (96%) as the first quantile and the maximum value for `pdays` is 999. This is an issue since we want to differentiate between people who were not contacted in a long time and people who never have been contacted. As such, we create a variable `prevcontact` which takes value 1 if the customer has been contacted before and 0 otherwise. We can also see some odd values in the table for predictors about the economic environment, for example, the mean for `EmpVarRt`

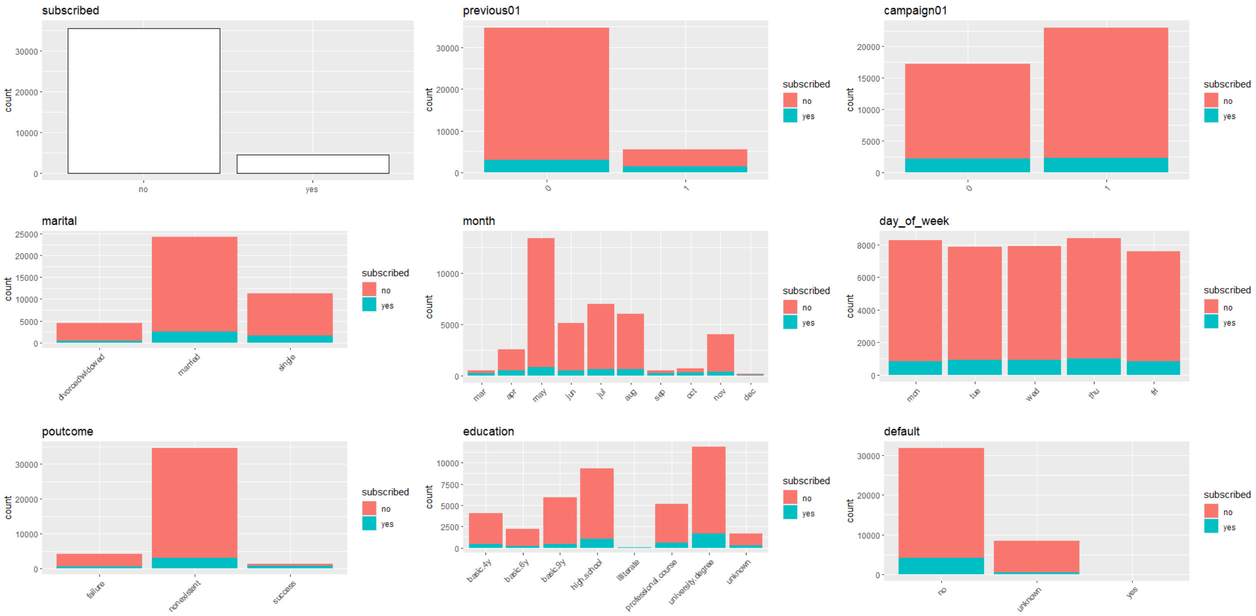

Figure 2: bar plots of some categorical variables with count of yes in green and count of no in red

is 0.08 while the median is 1.1. This simply is the consequence of many repeated values for these predictors. Therefore, to get an idea of the distribution of these variables, we plot the distribution of unique values, assuming that the precision of the observations is high enough so that the economic variables never take the same values at different times. These variables all have more or less even distribution except for `euribor3m` (see figure 1) which is bimodal.

There are no missing values in this data set. However, using the `summary` function, we observe that the variables `marital`, `job`, `default`, `education`, `housing`, `loan` have respectively 80, 330, 8597, 1731, 990, 990 unknown values. We first remove observations with "unknown" for `marital` since there are only 80 unknown values. Also, the observations having unknown values are the same for `housing` and `loan`. On the one hand, they represent only 0.02% of the whole data set, hence we could remove those to get fewer levels for our factors. On the other hand, they bring information that might be relevant to decide if customers are likely to subscribe to the bank term deposit or not (10.8% of observations with "unknown" loan or housing subscribe against 11.3% over the whole data set). Therefore, we decide to create another data set `bankmarket_unkn` without these observations, to use selection methods on both data sets and then compare models. It is clear that we should consider `previous` to be treated as an ordinal variable because it can only take integer values and because a very high number of observations take value 0. Although the variable could possibly take values greater than 7 (its maximum value in this data set) this is unlikely to happen and of little interest for prediction since it never occurs out of more than 40 000 observations. We have the same interrogation for `campaign` and therefore we decide to create three factor variables: `previousfactor`

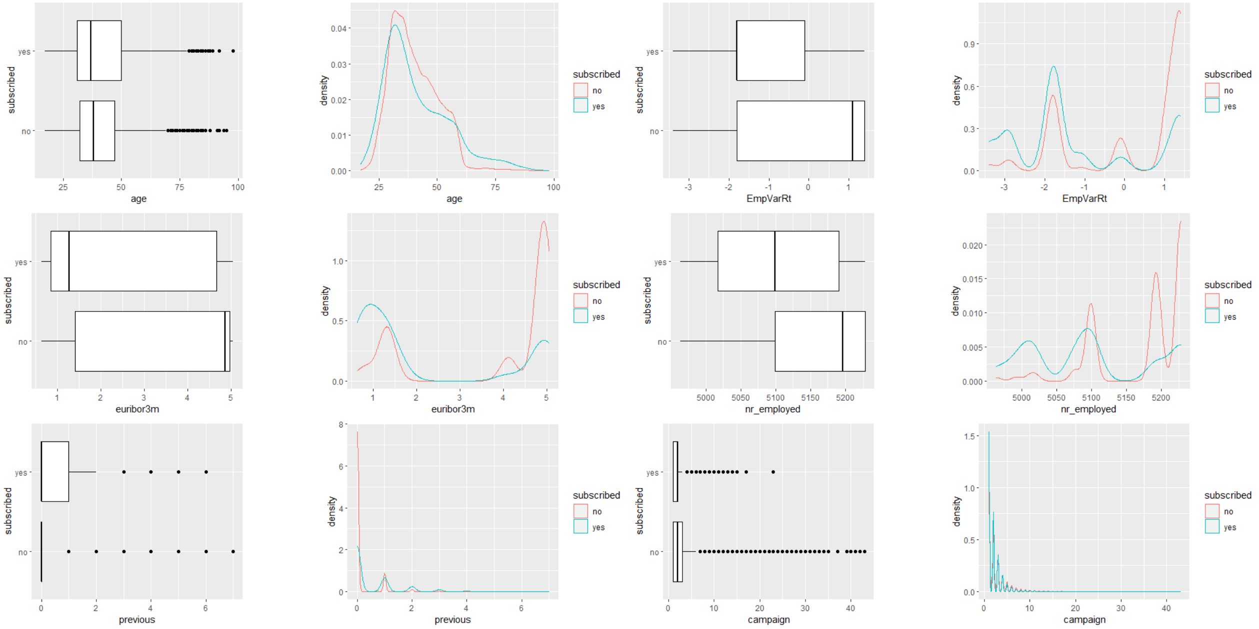

Figure 3: boxplots and density plots of some continuous variables with respect to `subscribed`

with levels from 0 to 7, `previous01` which takes value 0 if there was a contact before the current marketing campaign and 1 otherwise and `campaign01` which takes value 0 if there was only one contact during the current campaign and 1 otherwise.

From figure 2, we first observe that the outcome of interest is relatively rare (11%). The number of calls were made evenly during the week, with approximately the same number of yes each day. On the other hand, the distribution for `month` is uneven with May having far more calls than September or December. This is strange and we should ask employees about this if we can. The proportion of yes also differs with the month. We observe different proportions of yes for different levels in `marital` `previous01`, `education`, `poutcome` and `default`. These variables are likely to be relevant for our logistic model.

We observe that the economic variables have lower values when the customer subscribed to the product. We also note they are highly correlated which could lead to potential issues for interpretation and for using stepwise regression. Furthermore, it is harder to see the impact of `campaign` and `previous` when we treat them as continuous on `subscribed` than it is when we treat these variables as a factor. We did not notice any concerning outliers so far after transforming the skewed variables.

Finally, when fitting a logistic regression model, we assume that $x^T\beta = \log(\mu/(1-\mu))$.[2] We check this by plotting empirical logit plot. Only `age` (see figure 4) looks problematic and we might include a quadratic term when fitting the model. Other variables look to have a correct relationship.

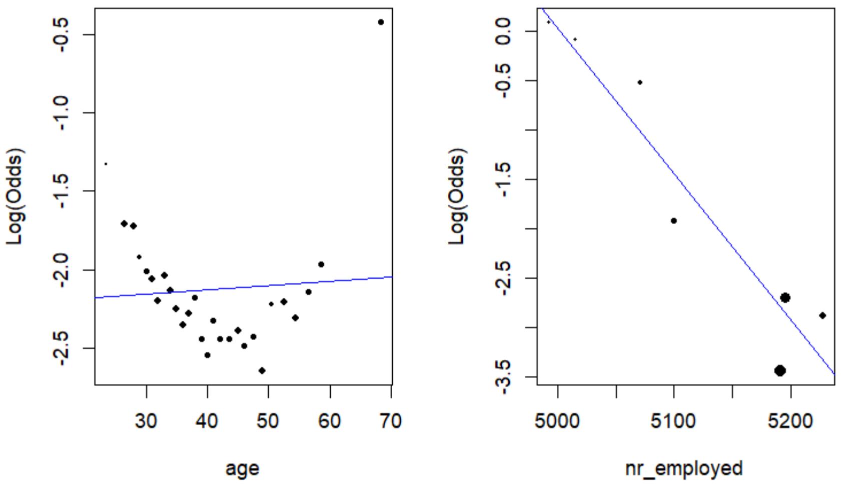

Figure 4: Empirical logit plot for age and nr_employed, the blue line is the regression line

2.3 Model choice

To choose our model, we will use ridge regression and backward stepwise regression. For both selection method we decide to include an interaction term between `month` and `day_of_week` and for ridge regression between all economic variables and `month` because we guess that the impact of economic environment variables will be different in time.

2.3.1 Backward stepwise regression

Firstly, stepwise regression has many inconveniences, mainly that "it provides regression coefficients that are biased high in absolute value" and so coefficients usually need shrinkage, "it yields P-values that are too small" and "the method yields standard errors of regression coefficient estimates that are biased low".[3] Furthermore, our data set has multicollinearity between the economic variables, which can make the variable selection procedure for these variables arbitrary. We use backward selection method because it usually performs better when there is multicollinearity and it allows us to start from a full model that does not have the drawbacks mentioned above.[3] However, stepwise regression can be useful because we lack of expertise in the banking area and it can help us get to a reduced and parsimonious model but it also allows us to avoid overfitting. We perform backward selection starting from the full model with the interactions added and using AIC as selection criteria. We do not use BIC since we do not want to penalize heavily complex models as our main goal is prediction. The variables that are selected are: `default`, `contact`, `month`, `day_of_week`, `pdays`, `poutcome`, `ConPrIdx`, `ConConIdx`, `euribor3m`, `nr_employed`, and `month:day_of_week` . To prevent arbitrary variable selection and make sure the algorithm selects the right variables, we perform backward stepwise regression on 10 folds. We obtain for every fold the same set of predictor variables, checking robustness of our stepwise model. Finally, we perform the same procedure without considering any interactions and we compare the results in 2.4

2.3.2 Ridge Regression

We also perform ridge regression using the same set of variables as in 2.3.1 with in addition the predictor variable `previouss` that was making singularities appearing when using stepwise regression. We can do this because of ridge regression's robustness. We group the factor variables and interaction terms together and perform cross validation with 10 folds to decide for the penalization parameter lambda. We do not take the lambda that minimizes the OLS (ordinary least square) but the one at one standard deviation from the one that minimize the OLS to decrease overfitting at the price of a small bias in our predictions. Because of the many levels in the factor variables and the interaction terms, it is impossible with the available space to print the whole output or to see something from the trace plot. Therefore I just enumerate the variables that have coefficients higher than 0.5 in absolute value : `intercept`, `education`, `default`, `poutcome`, `prevcontact`, `poly(age,2)`, `month:day_of_week`. In particular, we notice that these variables were selected in the stepwise model except for `age`.

2.4 Model Evaluation

data set	model	number of var./coeff.	AIC	AUC
data set without ”unknown” in loan and housing	full model	22/109	21944	0.803
	stepwise model	10/72	22162	0.797
	ridge model	23/175	NA	0.797
	full model (no interactions)	22/51	22262	0.793
data set with ”unknown” in loan and housing	full model	22/111	22800	0.804
	stepwise model	11/79	22659	0.797
	ridge model	23/177	NA	0.796
	full model (no interactions)	22/53	22765	0.794

Table 2: Summary statistics for our models

From table 2, we observe that AIC is greatly reduced for all models that are fitted on the data set where the observation with unknown values in `loan` and `housing` are deleted while the AUC is increased by only 0.001. Therefore, we decide to choose a model from the data set where the level ”unknown” has been removed from `loan` and `housing`. The full model and stepwise model have slightly higher value of AUC than the ridge model and are much less complex. Because we aim to explain the model to our client and to present him the most efficient and robust model (end of 2.3.1 section), we choose the stepwise model.

From the residuals vs leverage plot, we see that all observations have a cook distance that is less than 0.5. Furthermore, there are no sign of linearity violation from the residual vs fitted plot and marginal residual plots. Moreover, we use the `CVbinary` function from the DAAG package to check prediction against new data. We obtain a cross validation estimate accuracy of 0.9.

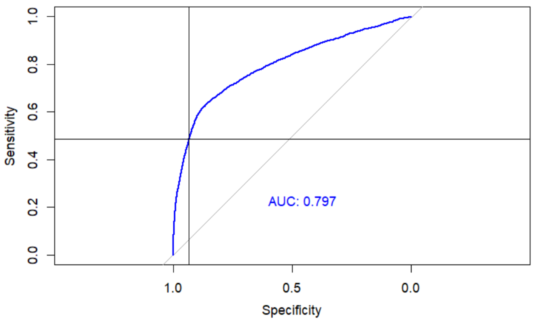

Figure 5: roc curve with lines indicating specificity and sensitivity for our chosen cut off probability

Finally, we need to choose a suitable cutoff point for our predictions. Although we were not given any instruction from the client about which misclassification he wants to avoid more, we guess that the bank prefers to call people that will not subscribe to the product than to miss potential new subscribers. So we will foster low probabilities cut-off point as long as number of false positives stay reasonable (the bank cannot afford to call too many people). Looking at the roc curve and after performing some trials, we have chosen 0.3 as the cut off point probability. Therefore, if the probability of an individual to subscribe is greater or equal to 0.3, then we classify the individual as ”subscribe”, and else as ”not subscribe”. We get a false positive rate of 0.514 (sensitivity of 0.486) and a false negative rate of 0.0676 (specificity of 0.9324).

2.5 Limitations of our model and criticisms

Originally, we tried to use more interactions for the backward selection process but we were having separation errors. We tried to handle this issue by applying the method developed by Firth[1] and implemented in the `logistf` package. However, we were unsuccessful to solve this problem. We were fitted the model better (AUC of 0.804) but it was impossible to produce log odds ratio for all levels of the categorical variables and produce effect plots. Furthermore, confidence intervals were very large. Therefore, solving this issue and check what happens would be of interest. We tried to perform lasso regression as well but we obtained terrible results. Making lasso regression works could be interesting to get shrinkage of coefficients and variable selection at the same time. Elastic net could be another option as it combine the advantages of both lasso and ridge regression.

3 Model Summary

Our model to predict if the customer will subscribe to the term deposit needs 10 different information that can be either an information specific to the person (such as if he defaulted or not) or specific to the time or economic environment (month or interest rate). The information required are : did he default, was he contacted by telephone or cellular and when (month and day of the week), the outcome of the previous marketing campaign, the number of days that passed before the previous contact, and finally the values of the consumer confidence and price index, the euribor 3 rate and the number of person employed.

3.1 Model performances, Interpretation and Predictions

Employees of the bank making the call will miss on average only 6% of the customers by not calling those that were predicted as ”no” by the model. By calling all the persons predicted as ”yes”, they will only get a negative answer from the customer half the times.

We interpret the impact of each information we have on the customer and the global environment in terms of the probability that he will subscribe. In figure 6, we can see the predicted probability of subscription on the left for every value our variables can take. Purple whiskers for categorical variable and blue font tells us about how confident we are in our prediction. For example, we can see that that people who were called by cellular are more likely to subscribe than people called by telephone.

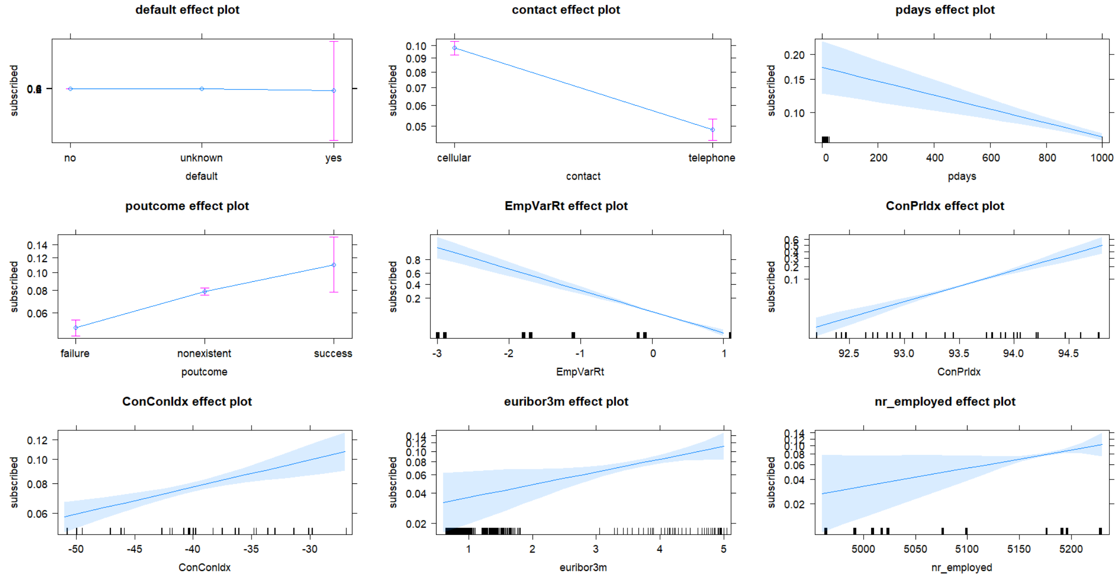

Figure 6: effect plots of all variables except dates and job (for space economy)

We will illustrate how we can use our model to predict if this customer of interest will subscribe. To compute the predicted probability of subscribing for this customer we first sum the product of the coefficients of the continuous information and the value of that information. We add to this the coefficients for the categories. We obtain:

$x = 999(-9.634e-4) + (-1.5)(-1.531) + 94(2.052) + 5(0.3014) + (-40)(2.795e-2) + 5100(5.346e-3) - 0.03148 + 0 - 0.7602 + 0.5044 - 2.085 + 0.1222 + 0.6953 = 220.32$

Finally:[2] $P(subscribed|information) = (1 + exp(-(intercept + x)))^{-1} = 0.61$ where the intercept is -219.9. Since $0.61 > 0.3$, we predict that this customer will subscribe.

3.2 Success, Limitations and Improvements

3.2.1 Success and Limitations

Our model has high prediction accuracy and is quite robust against new data. It is easy to visualise from figure 6 which factors increase the chances for the customer to subscribe. That can help them improve their marketing campaign and get a better understanding of the whole process.

Some limitations are inherent to the method we used to derive our model (see section 2). Experts' judgments are also key to have an idea what information is relevant to make predictions. This can help us reduce both the drawbacks from the method we used and the chances of getting a wrong model.

3.2.2 Improvements

The data collection could be improved in many ways by getting more information about the customer (is he already a customer in the bank? for how long?), about the employee who made the call (how experienced is he ?) about the reasons why many calls were made in some months compared to others and by getting more data. The method to select our model could be improve by using more advanced technique. The model could be implemented by writing a program that would directly give probabilities of the customer to subscribe so that employees of the bank can call first people with the highest probability to subscribe. The bank could also try to discriminate their product in terms of the customer to increase their profits like some companies give reduced price for students for example.

4 Bibliography

References

[1] David Firfth. “Bias reduction of maximum likelihood estimates”. In: *Biometrika* 80.1 (Mar. 1993), pp. 27–38. ISSN: 0006-3444. DOI: 10.1093/biomet/80.1.27. eprint: https://academic.oup.com/biomet/article-pdf/80/1/27/616333/80-1-27.pdf. URL: https://doi.org/10.1093/biomet/80.1.27.

[2] Jr. Frank E. Harrell. *Regression Modeling Strategies*. Springer International, 2015. Chap. binary logistic regression, pp. 219–221. ISBN: 978-3-319-19424-0.

[3] Jr. Frank E. Harrell. *Regression Modeling Strategies*. Springer International, 2015. Chap. variable selection, pp. 67–71. ISBN: 978-3-319-19424-0.

5 Source code

```
# note : #... means the code is repeated for other variables/models or data sets
# ---------- import required libraries
library(ggplot2)
library(tidyverse)
library(reshape2)
library(VIM)
library(corrplot)
library(cowplot)
library(car)
library(MASS)
library(Stat2Data)
library(plyr)
library(arm)
library(DAAG)
library(effects)
library(glmnet)
library(pROC)
# import data set
load("bankmarket.Rdata")
attach(bankmarket)

#  ---------- EDA ----------

# remove duration
bankmarket = subset.data.frame(bankmarket,select = -c(duration))

# relevel month and day_of_week
bankmarket$month = factor(month, levels =
    c("mar","apr","may","jun","jul","aug","sep","oct","nov","dec"))
bankmarket$day_of_week = factor(day_of_week,
        levels = c("mon","tue","wed", "thu", "fri"))

# create vectors containing variable names of numeric and factor variables
numeric_var = colnames(bankmarket)[sapply(bankmarket, is.numeric)]
factor_var = colnames(bankmarket)[sapply(bankmarket, is.factor)]
response = "subscribed"
```

```r
summary(bankmarket)

# create vectors for aggregate and individual predictor variables
numeric_var_aggregate = c("EmpVarRt","ConPrIdx","ConConIdx","euribor3m","nr_employed")
numeric_var_ind = c("age", "campaign", "pdays","previous")

# check for missing values
any(is.na.data.frame(bankmarket))

# compute the percentage of customers that had not been contacted before
sum(pdays == 999)/41188
# add a variable prevcontact that takes values 1 if the customer
# was previously contacted and 0 otherwise
bankmarket$prevcontact = factor(ifelse(bankmarket$pdays == 999, 0, 1))

# distribution of economic variable (assuming these variables do not take
#the exact same value at different times) so for example these histograms
#make sense iff euribor3m does not take the same values on 2 different days
#(since this economic indicator is updated daily)
hist(unique(bankmarket$EmpVarRt),breaks = 5)
# ...
# find the number of unknown in each predictor
count((marital == "unknown"))
# ...
# check the observations with unknown values are the same for housing and loan
a = which(housing == "unknown")
b = which(loan == "unknown")
all(a==b)
bankmarket[a,]

# compute if the proportion of yes in the observation having unknow loan and housing
#is significantly different from the proportion of yes in the whole data set
count(bankmarket[a,"subscribed"] == "yes")
# remove 80 observations that have unknown in marital
bankmarket = subset.data.frame(bankmarket, marital != "unknown")
# add factor 0 1 for previous and pdays
bankmarket$previouss = ifelse(bankmarket$previous == 0, 0, 1)
```

```r
bankmarket$campaignn = ifelse(bankmarket$campaign == 1, 0, 1)
bankmarket$previousfactor = factor(bankmarket$previous)
bankmarket$previouss = factor(bankmarket$previouss)
bankmarket$campaignn = factor(bankmarket$campaignn)

factor_var = c(factor_var, "previouss", "campaignn","previousfactor")

# create another data set without unknown values for housing and loan
bankmarket_unkn = subset.data.frame(bankmarket, housing != "unknown")
bankmarket_unkn$loan = factor(bankmarket_unkn$loan)
bankmarket_unkn$housing = factor(bankmarket_unkn$housing)

# plot of each continuous predictors with respect to subscribed
plot_bivariate <- function(var, title=var, dataframe = bankmarket_unkn) {
    p1 <- dataframe %>%
        ggplot(aes_string(x =  var, y="subscribed")) +
       geom_boxplot(fill = "white", color = "black")
    p2<-dataframe %>%
      ggplot(aes_string(x=var, group="subscribed", col = "subscribed")) +
      geom_density()
    p <- plot_grid(p1, p2, ncol = 2, align = 'v', axis = "lr")
}
for(i in numeric_var_ind){
  plot_bivariate(i)
}

for (i in numeric_var_aggregate){
  plot_bivariate(i)
}

# plot of each factor variables with respect to subscribed
plot_factor2 <- function(var, title=var, dataframe = bankmarket_unkn) {
    p1 <- dataframe %>%
        ggplot(aes_string(x = var, fill="subscribed")) +
        geom_bar() +
        xlab("") +
        theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
        ggtitle(title)
        print(p1)
}

factor_var2 = factor_var[! factor_var %in% c("subscribed")]
plots = c()
for(var in factor_var2){
    plots = c(plots,plot_factor2(var))
}
# show proportion of yes and no for each level of the factor variables
options(digits = 3)
prop.table(table(marital,subscribed),1)
# ...
# produce empirical logit plots (function given in Pima practical)
myemplogit <- function(yvar=y,xvar=x,maxbins=10,sc=1,line=TRUE,...){...}
myemplogit(subscribed,age,30,sc=75,xlab="age")
# ...


#  ---------- modelling part ----------

# transform outcome no/yes into 0/1
bankmarket_factor = subset.data.frame(bankmarket_unkn,
        select = -c(campaign,previous,previousfactor))
attach(bankmarket_factor)
bankmarket_factor$subscribed =
        factor(ifelse(bankmarket_factor$subscribed == "yes", 1,0))
attach(bankmarket_factor)

# full model no interaction
full_model_ni = glm(subscribed ~., family = "binomial", data = bankmarket_factor)

# full model (interaction)
full_model = glm(subscribed ~ marital +age + age**2 + job + education + default +
housing + loan + contact + month + day_of_week + pdays + poutcome + EmpVarRt
+ ConPrIdx + ConConIdx + euribor3m + nr_employed + prevcontact + day_of_week:month,
family = "binomial", data = bankmarket_factor)

# backward stepwise regression
```

```r
glm.step = step(full_model, direction = "backward")

# check robustness of the stepwise model by repeating the procedure for ten folds
library(caret)
# creating folds
fold <- createFolds(bankmarket_factor$subscribed, k=10)
for (folds in fold){
  bankmarket_dataset = bankmarket_factor[folds,]
  glm.step_fold = step(full_model, direction = "backward",trace = 0)
  print("-------------------------------------------------------------")
  print(names(glm.step_fold$coefficients))

# ridge regression with interactions
form <- subscribed ~ . - age + poly(age,2) +  month:day_of_week +
month:day_of_week:euribor3m + month:euribor3m + day_of_week:euribor3m
+ month:ConPrIdx + month:ConConIdx + month:nr_employed + month:EmpVarRt +
prevcontact:pdays
x<- model.matrix( form,data=bankmarket_factor[,c(1:17,19:21)])[,-1]
group <- c(rep(1,each=3),c(rep(2,each=11)),c(rep(3,each=7)),c(rep(4,each=2)),c(5:7),
           c(rep(8,each=9)), c(rep(9,each=4)),c(10),c(rep(11,each=2)),c(12:19),
           c(rep(20,each=2)),  c(rep(21,each=36)), c(rep(21,each=9)),
           c(rep(22,each=4)), c(rep(23,each=9)),c(rep(24,each=9)),c(rep(25,each=9)),
           c(rep(26,each=9)),c(27), c(rep(28,each=36)))
# create values of lambda
grid =10^seq(-5,-1,length=80)
# performing the ridge regression
cvridge1_factor <- cv.glmnet(x=x,y=subscribed,family="binomial", groups=group,
lambda=grid, alpha = 0, nfolds = 10)

# create predicted values for all models
predicted_step = predict(glm.step, type = "response")
# ...
coeff_ridge = predict(cvridge1_factor,
        s = cvridge1_factor$lambda.1se,type="coefficients")
predicted_ridge = predict(cvridge1_factor,
        s = cvridge1_factor$lambda.1se,newx = x ,type="response")

# print coefficients of the ridge regression for lambda at 1 std
```

```r
predict(cvridge1_factor, s = cvridge1_factor$lambda.1se,type="coefficients")

# residual plots, multicollinearity, anova
marginalModelPlots(glm.step)
vif(glm.step)
anova(glm.step, test = "Chisq")
# ...

# prediction and model performances
boxplot(predicted_step~bankmarket_factor$subscribed)
CVbinary(glm.step)
# ...
# ROC plots with AUC
plot.roc(subscribed,predicted_step,col="blue",
ci=TRUE,of="thresholds",ci.type="shape",print.auc=TRUE,
print.auc.x= 0.6, print.auc.y=0.25,print.auc.col="blue")
#...
# cut off probabilities for stepwise model
table_pred = addmargins(table(bankmarket_factor$subscribed, predicted_step >0.30))
false_negative = table_pred[2,1]/table_pred[3,1]
false_positive = table_pred[1,2]/table_pred[3,2]
plot.roc(subscribed,predicted_step,col="blue",ci=FALSE,of="thresholds",
ci.type="shape",print.auc=TRUE,print.auc.x= 0.6, print.auc.y=0.25,print.auc.col="blue")
abline(v = 1 -  false_negative)
abline(h = 1 - false_positive)

# log odd ratio and effect plots
library(sjPlot)
plot_model(glm.step,type="std",axis.lim = c(0.4,8),auto.label = FALSE )
plot(allEffects(glm.step)[2:10])

# prediction for the new customer
newcustomer= data.frame(marital="married", age=45, job="technician",
education="university.degree", default="no",housing="yes",loan="no",
contact ="telephone", month="nov", day_of_week="thu", pdays=999,
poutcome="nonexistent",EmpVarRt=-1.5, ConPrIdx=94, ConConIdx=-40, euribor3m=5,
nr_employed=5100, prevcontact=0, previouss=0, campaignn=0)
predict.glm(glm.step, newdata = newcustomer, type = "response")
```