

**6040 Data Mining Final Project – Banking Deposit Subscription**

Songyuan Luo

Northeastern University

Contents

[Abstract 2](#_Toc522368541)

[1. EDA 2](#_Toc522368542)

[2. Decision tree model 5](#_Toc522368543)

[3. Factor analysis 5](#_Toc522368544)

[4. SVM 8](#_Toc522368545)

[Conclusion and further improvement 9](#_Toc522368546)

# 

# **Abstract:**

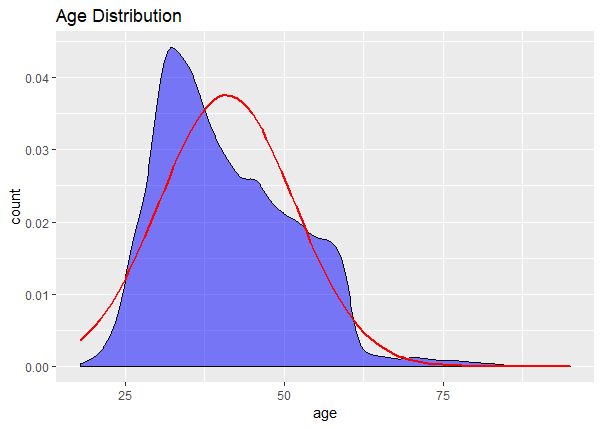
The data set is from Portuguese Bank that initiated the telemarketing campaign from 2008 to 2013. The data set is a good way to consider changes of economic condition as well as using data analysis techniques to provide business insights. There are two data set in general, and this report only consider the sample data set. The main idea of the report is to find out a data analysis solution to help bank employees issue subscription more efficiently. The report will be consisted of EDA (exploratory data analysis), decision tree model, factor analysis and SVM (supporting vector machine).

1. **EDA**

EDA, exploratory data analysis, is a common method for data scientists to get a bird view of the whole data set. For the data set (I will use ‘banking data set’ for the rest of report), there are 17 variables and 45211 observations. The variables are: age, job, marital, education, default, balance, housing, loan, contact, day, month, duration, campaign, pdays, previous, poutcome and y (has the client subscribed a term deposit?). In the data set, housing means whether client has housing loan; default means whether has credit in default; pdays means number of days that passed by after last contact, poutcome means outcome of the precious marketing campaign. All the detailed descriptions can be found in appendix provided by UCI.

Next, Let’s take a closer look at the data set. I had several assumptions during EDA part ---- whether age, education level, balance and housing loan be main decisive factor in the outcome. The following analysis will help to understand the questions.

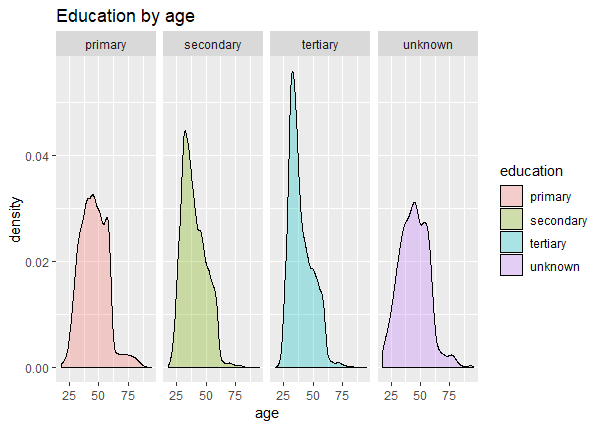
We can notice that the age distribution is quite right skewed comparing to normal distribution line(red line). The minimum age is 18, and the maximum is 95; the average age is 41.



*Fig.1 Age Distribution with normal distribution line*

*Sourced by UCI machine learning repository*

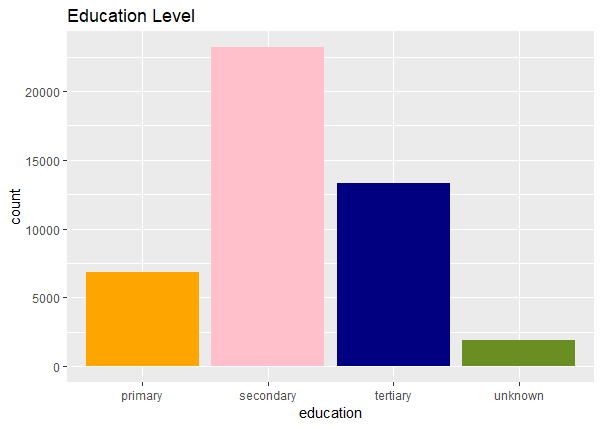
If we divide education by age, we can notice that people with more advanced education level tends to be younger than those in the opposite. However, the common trend is still right skewed as shown below. The average age of people with primary education level is 45.87, while those with secondary education level is 40, and those with tertiary education level is 39.



*Fig.2 Education by age (density plot)*

*Sourced by UCI machine learning repository*

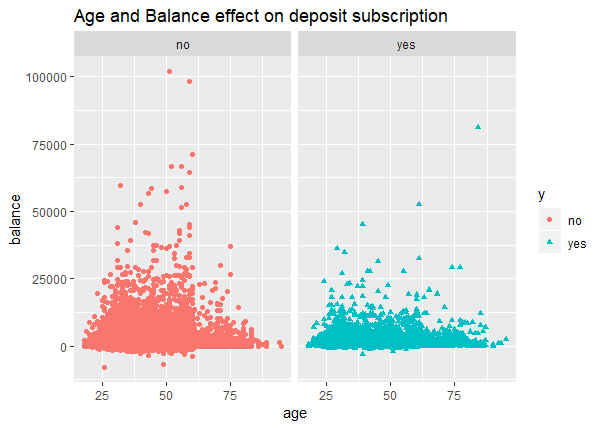
Among all the people who come for deposit subscription, secondary education level accounts for the largest proportion, with 55% in total. People with tertiary education level accounts for approximately 28%. Until now, we can not say there is a correlation.



*Fig.3 Education level by amount (Histogram)*

*Sourced by UCI machine learning repository*

In addition to education factor, I found age and balance effect interesting. The assumption is: the older the client and less balance he/she had, the less likely he/she will get new deposit subscription. However, there is no correlation among the factors.



*Fig.4 Age and Balance effect on subscription*

*Sourced by UCI machine learning repository*

Last but not least, I generated the bar graph to find out relationship between housing loan and outcome. The answer is there is no correlation between them.



*Fig.5 Relation between housing loan and subscription success*

*Sourced by UCI machine learning repository*

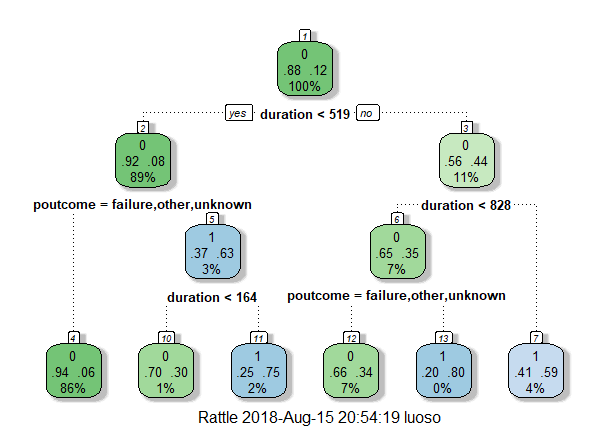
We can learn that several solvency indicators cannot be used as deposit term subscription evaluation. Next part will use decision tree model and find out a potential pattern for employees’ guidance.

1. **Decision tree model**

Decision tree model is a common classification tool that can used both in exploratory analysis and prediction. As banking data set contains 10 categorical data type, it is a perfect approach to find out pattern.

The decision tree model shows that 86% of failed deposit subscription comes from those who spent less than 519 seconds or 8.7 mins with agents, and didn’t have marketing campaign before. The 8 mins talk may differentiate potential customer as well as agents’ understanding of customer condition. In addition, people who didn’t have campaign may be considered as low solvency ability.

I pruned tree model afterwards but get same model as well; the mean of square error of the model is 0.42, which is too high to be considered as a good model. I believe the reason is because some hidden factors are not revealed.

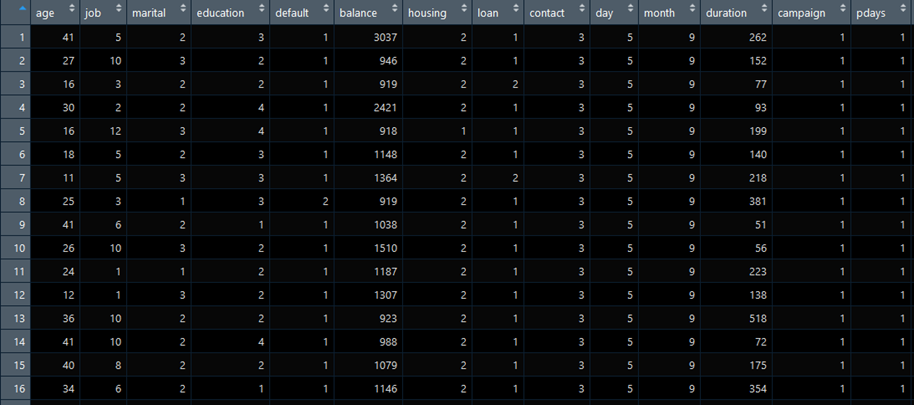


*Fig.6 Decision tree model*

1. **Factor analysis**

Factor analysis is a statistical method used to describe variability among observed, correlated variables in terms of a potentially lower number of unobserved variables called factors. In brief, we use the method to delete less useful variables without losing too much information.

How can we use the method? Firstly, we need to split raw data set into training and testing data set, in this case, we are using 70% of banking data set as training and 30% as testing. Next, we need to convert variables into as numeric factor for further analysis. Because the data set is consisting of several categorical variables, the R console automatically transformed different types of elements into sequence. For example, for “loan” variable, value ‘yes’ will be 1 and ‘no’ will be 0. After that, we will scale and center the data so as to improve analysis interpretation.

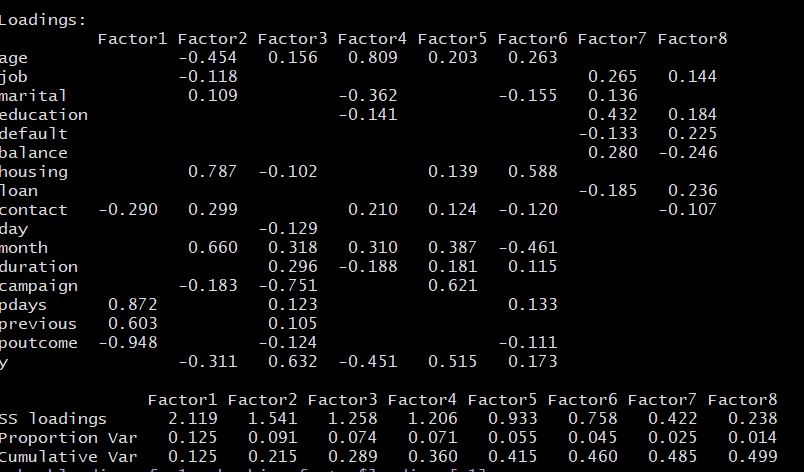


*Fig.7 Scaled and centered data after preprocess*

After preprocessing, the data set will be looks like above.

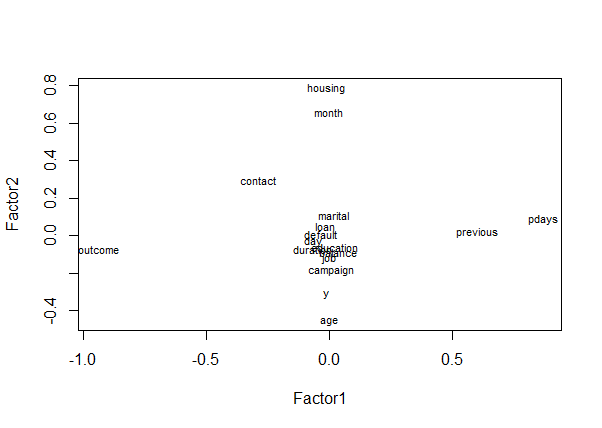
Next, we will do factor analysis with no rotation. When we are using factor analysis, it is important to determine how many factors to form. The formula: (p−k)2>p+k(p−k)2>p+k. will help us. P means data points or variables, k means factors you want to form. I will use 8 in this case.

The outcome is shown below:



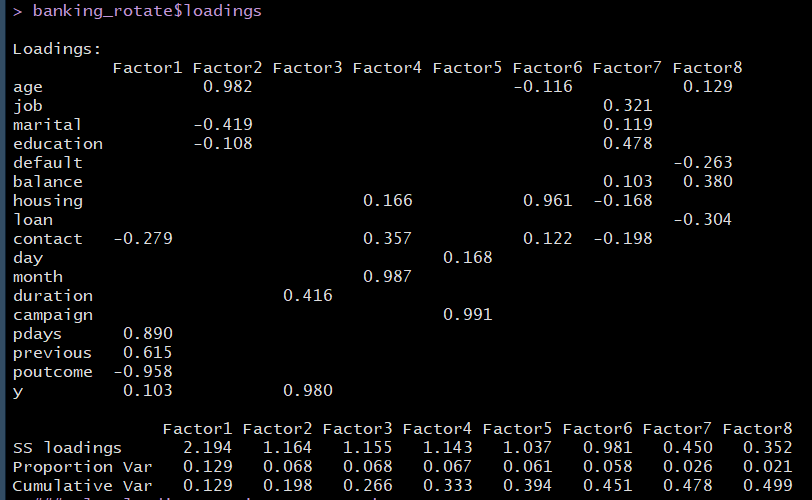
*Fig.8 Factor analysis outcome table before rotation*

As you can see, the table is about how well the variables can be explained by factors. For example, Factor1 can explain poutcome and pdays really well. However, if we plot factor 1 against to factor 2, as shown below:



*Fig.9 Factor one and Factor two with variables*

This plot means for factor 1, poutcome and pdays should be composite while housing and age should be composite for factor 2. It is because variables with most variance tend to be more informative. However, we need to rotate the factor because for factor 3 there is no significant variables. After rotation, we can see the difference from the outcome table.



*Fig.9 Factor analysis outcome table after rotation*

To be clear, we will use SS loadings to evaluate factor. The factors with SS loadings larger than 1 are relatively important. Before rotation, there are 4 important factors and after rotation, there are 5. There are other differences as well, the amount of variables being explained by multi factors decreased, and almost every important factor can explain one or more variables.

For conveience, I listed them below:

For factor 1, previous campaign outcome and days after previous campaign are more important. Factor 2: age. Factor 3: y (the outcome of deposit subscription). Factor 4: month of last contact. Factor 5: number of campaigns performed for this client. I would like to take factor 6 into account as well, as the housing loan takes 96% of the whole factor.

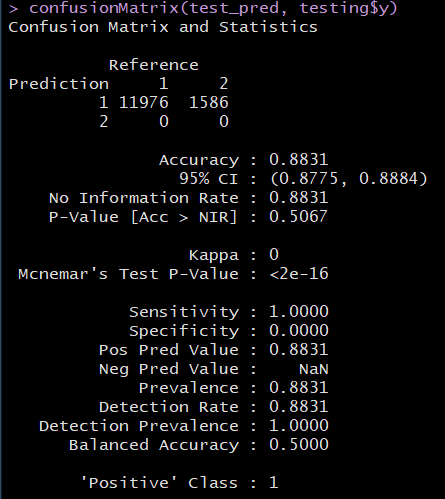
After reducing 17 variables into 6 factors, we can train another classification model to help us know the pattern.

1. **SVM**

For this part, I didn’t combine variables into factors as adding them and mean is not appropriate. I subset important factors into new data set called banking\_subset.

Supporting vector machine is another useful tool data analysts usually use when considering classification problems. The advantage of SVM is with small data set, we can form robust model for prediction.

The outcome is shown below:



*Fig.10 Supporting Vector Machine outcome table*

We can notice that the accuracy of SVM model is quite satisfying, reached to 88%. I think this model can be used for a guidance, or to use as an initial classifier that will save time and efforts.

**Conclusion and further improvement**

As I mentioned before, the dataset is collected from 2008 to 2013, after financial crisis, certain solvency indicators used commonly before are not effective nowadays. In fact, we can know from factor analysis that only housing loan still have effect on the subscription, other factors like education, balance or job are no longer useful. This report is not to provide reason for this phenomenon, I can only draw the conclusion that with supporting vector machine model with 88% accuracy, this can really help employees and the bank to work more effective. Factors that need to be taken into account are the outcome of last campaign, duration after last campaign, housing loan, age as well as month since last campaign. Most of the factors are about last marketing campaign.

For further improvement, I will try multi gradients to optimize the supporting vector machine model, or even use artificial neural network on the banking data set.

Appendix:

1. R code:

#The final project is about a banking campign to determine whether a customer will subsribe a term deposit.

#The whole project will be consist of four parts: EDA, decision tree model, factor analysis and SVM

##############################################

#############################################

library(ggplot2)

library(dplyr)

library(tidyr)

library(rpart) #classification and regression trees

library(partykit) #treeplots

library(randomForest) #random forests

library(gbm) #gradient boosting

library(caret) #tune hyper-parameter

library(rattle)

library(rpart.plot)

library(RColorBrewer)

#Loading data set

setwd("C:\\Users\\luoso\\Desktop")

banking <- read.csv('banking.csv')

dim(banking)

summary(banking)

banking$y = as.factor(banking$y)

age\_primary <- banking %>% filter(education == 'primary') %>% select(age)

summary(age\_primary)

age\_secondary <- banking %>% filter(education == 'secondary') %>% select(age)

summary(age\_secondary)

age\_tertiary <- banking %>% filter(education == 'tertiary') %>% select(age)

summary(age\_tertiary)

#EDA part

#1.plot age distribution with normal distribution curve.

ggplot(data = banking, aes(x=age)) +

geom\_density(fill = "blue",alpha = 0.5) +

labs(title="Age Distribution",x="age",y="count") +

stat\_function(fun = dnorm,args = list(mean = mean(banking$age, na.rm = TRUE),

sd = sd(banking$age, na.rm = TRUE)), color ='red',size = 1)

ggplot(banking,aes(job))+

geom\_bar() +

labs(title="Job bar plot",x="Job type",y="amount")

#2.density plot on education by age

ggplot(data = banking, aes(x=age,fill = education)) +

geom\_density(alpha = 0.3) +

facet\_grid(.~ education)+

labs(title='Education by age',x = 'age')

#3.Plot histogram on education level

ggplot(data = banking, aes(x=education)) +

scale\_color\_manual(values=c("orange","pink","navy","olivedrab")) +

geom\_histogram(fill = c("orange","pink","navy","olivedrab"),alpha = 1,stat = "count")+

labs(title = 'Education Level')

#4.age and balance with yes or no classification

ggplot(data = banking, aes(x=age,y=balance,color=y , shape = y)) +

geom\_point() + facet\_grid(.~ y) +

labs(title = 'Age and Balance effect on deposit subscription')

#5.plot bar graph on relation between housing and success

ggplot(data = banking,aes(x=housing, fill = housing)) +

geom\_bar() + facet\_grid(.~y) +

labs(title = 'Relation between housing loan and subscription success')

#As we can see from the summary, most of variables are binary or categorical data inculding target varible.

#Next, I will form a decision tree model for prediction.

banking$y = ifelse(banking$y == 'yes',1,0)

head(banking$y)

#split into 70% training and 30% testing

setseed(12345)

intrain <- createDataPartition(y = banking$y,p=0.7,list = FALSE)

training <- banking[intrain,]

testing <- banking[-intrain,]

training$y = factor(training$y)

treemodel <- rpart(y~., data = training)

plotcp(treemodel)

print(treemodel$cptable)

fancyRpartPlot(treemodel)

printcp(treemodel)

cp\_treemodel = min(treemodel$cptable[4,])

prune\_treemodel <- prune(treemodel,cp = cp\_treemodel)

fancyRpartPlot(prune\_treemodel)

fancyRpartPlot(treemodel)

predict\_treemodel <- predict(treemodel,newdata = testing)

rpart.resid = predict\_treemodel - testing$y #calculate residuals

mean(rpart.resid^2) #caluclate MSE 0.4

#MSE is really quite large. I will try to use factor analysis and then supporting vector machine to get a higher accuracy

######################################

#Factor analysis

banking$age = as.numeric(as.factor(banking$age))

banking$job = as.numeric(as.factor(banking$job))

banking$marital = as.numeric(as.factor(banking$marital))

banking$education = as.numeric(as.factor(banking$education))

banking$default = as.numeric(as.factor(banking$default))

banking$balance = as.numeric(as.factor(banking$balance))

banking$housing = as.numeric(as.factor(banking$housing))

banking$loan = as.numeric(as.factor(banking$loan))

banking$contact = as.numeric(as.factor(banking$contact))

banking$day = as.numeric(as.factor(banking$day))

banking$month = as.numeric(as.factor(banking$month))

banking$duration = as.numeric(as.factor(banking$duration))

banking$campaign = as.numeric(as.factor(banking$campaign))

banking$pdays = as.numeric(as.factor(banking$pdays))

banking$previous = as.numeric(as.factor(banking$previous))

banking$poutcome = as.numeric(as.factor(banking$poutcome))

banking$y = as.numeric(as.factor((banking$y)))

banking\_stand = as.data.frame(scale(banking))

### Factor analysis with no rotation

banking\_factor = factanal(banking\_stand, factors = 8, rotation = "none", na.action = na.omit)

banking\_factor$loadings

#factors from 1 - 4 are rather important

bankloadings\_fac1 = banking\_factor$loadings[,1]

eigenv\_fact1 = sum(bankloadings\_fac1^2); eigenv\_fact1

# Compute proportion variance

eigenv\_fact1/17

banking\_factor$uniquenesses

bankloadings\_distant = banking\_factor$loadings[1,]

bankcommunality\_distant = sum(bankloadings\_distant^2); bankcommunality\_distant

unique\_distant = 1-bankcommunality\_distant; unique\_distant

### Plot loadings against one another

load = banking\_factor$loadings[,1:2]

plot(load, type="n") # set up plot

text(load,labels=names(banking),cex=.7) # add variable names

banking\_rotate = factanal(banking\_stand, factors = 8, rotation = "varimax", na.action = na.omit)

banking\_rotate$loadings

### Plot loadings against one another

load = banking\_rotate$loadings[,1:2]

plot(load, type="n") # set up plot

text(load,labels=names(banking),cex=0.7)

###Make a subset

need\_data <- c('age','housing','month','campaign','pdays','poutcome','y')

bank\_subset <- banking[need\_data]

bank\_subset$y = as.factor(bank\_subset$y)

######Supporting vector machine

set.seed(12345)

intrain <- createDataPartition(y = bank\_subset$y, p= 0.7, list = FALSE)

training <- bank\_subset[intrain,]

testing <- bank\_subset[-intrain,]

dim(training); dim(testing);

training$y

training$y = as.factor(training$y)

trctrl <- trainControl(method = "repeatedcv", number = 10, repeats = 3)

set.seed(3233)

svm\_Linear <- train(y ~., data = training, method = "svmLinear",

trControl=trctrl,

tuneLength = 2)

svm\_Linear

test\_pred <- predict(svm\_Linear, newdata = testing)

test\_pred

testing$y = factor(testing$y)

confusionMatrix(test\_pred, testing$y)