Initial analysis

The data has 19 columns and 4000 rows

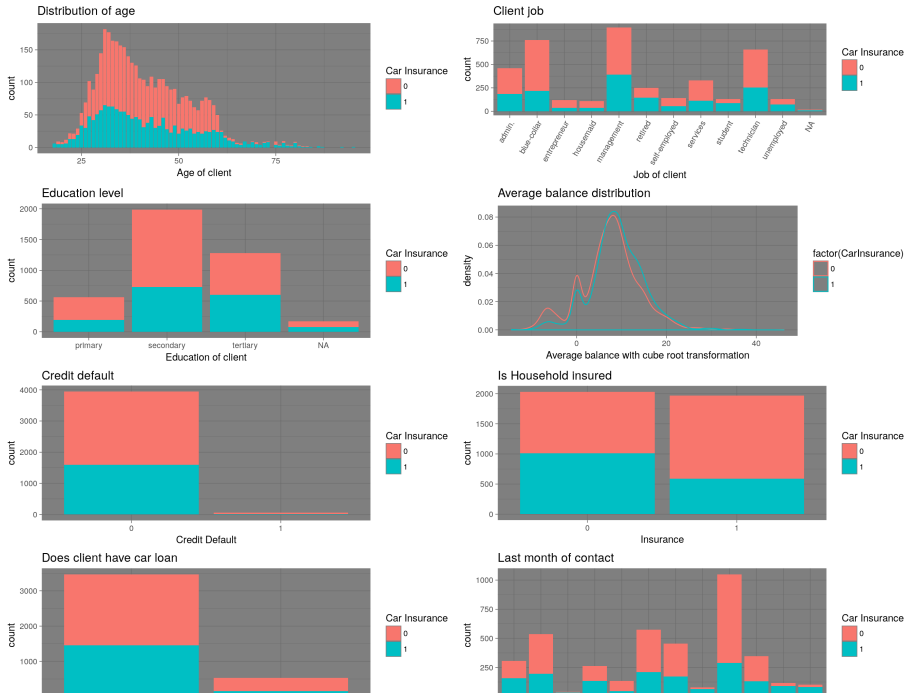
First we look at which columns are categorical, unique level in each column and the number of missing values in each column

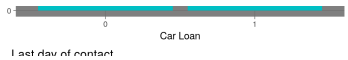
	Levels	Categorical	Missing
Id	4000	FALSE	0
Age	70	FALSE	0
Job	12	TRUE	19
Marital	3	TRUE	0
Education	4	TRUE	169
Default	2	FALSE	0
Balance	2178	FALSE	0
HHInsurance	2	FALSE	0
CarLoan	2	FALSE	0
Communication	3	TRUE	902
LastContactDay	31	FALSE	0
LastContactMonth	12	TRUE	0
NoOfContacts	35	FALSE	0
DaysPassed	330	FALSE	0
PrevAttempts	20	FALSE	0
Outcome	4	TRUE	3042
CallStart	3777	TRUE	0
CallEnd	3764	TRUE	0
CarInsurance	2	FALSE	0

Target variable distribution

```
   0    1
2396 1604
```

Univariate analysis





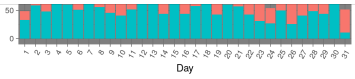
Last day of contact

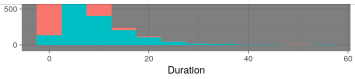
Call Duration

Data Exploration

Rmarkdown script using data from Car Insurance Cold Calls · 721 views · 2y ago

7 Copy and Edit 5 ...





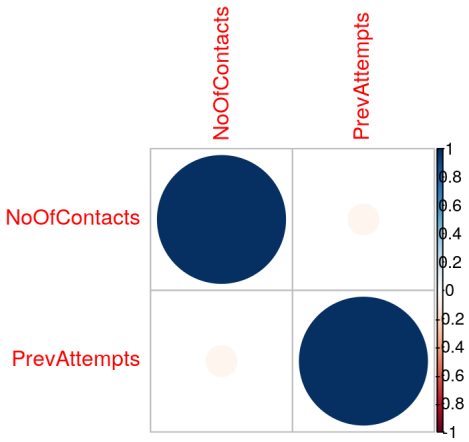
Version 4

4 commits

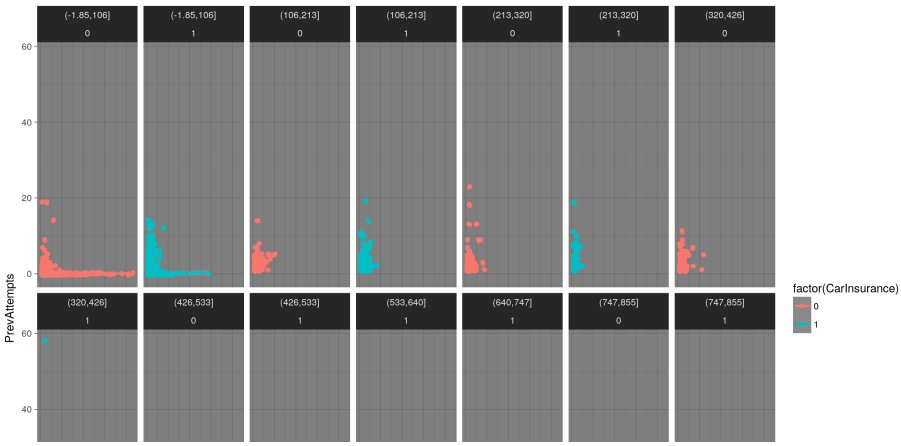
1. **Age** : Most population is between the age of 25 to 60. The distribution of whether car insurance was brought or not seems fairly even
2. **Job** : Students, unemployed and retired individuals tend to buy car insurance more often than not
3. **Education**
4. **Average Balance** : With a cube root transformation to normalize the variable. the class distribution looks pretty similar
5. **Credit Default** : The distribution looks pretty similar for both classes
6. **Insurance** : Uninsured households are more likely to buy car insurance
7. **Car Loan**
8. **Month of Contact** : Individuals are more likely to buy car insurance during months
9. **Day of Contact** : Individuals are more likely to buy insurance at the start of the month
10. **Call Duration** Longer calls lead to more purchases

Multivariate Analysis

1. **Number of Contacts made in previous campaign and Current campaign**



There is no correlation between the number of contacts made in previous campaigns and current campaigns. Let's look at how does the number of days passed and the number of contacts made in each campaign are related



NoOfContacts

It can be inferred from the above plots, higher previous contacts and lesser duration between campaigns lead to higher purchases and calls after longer durations lead to higher purchases, though we cannot substantially say this due to non-availibility of considerable number of datapoints.

More to come ! :D

Code

Download Code

````
```{r setup, include=FALSE}
knitr::opts_chunk$set(echo = TRUE, message = F, echo = F, warning = F,tidy = T,comment = NA)
```

<h3><center> Initial analysis </center></h3>

```{r shape}
df=read.csv("../input/carInsurance_train.csv", header = T, sep= ",")
```

The data has `r ncol(df)` columns and `r nrow(df)` rows

First we look at which columns are categorical, unique level in each column and the number of missing values in eac

```{r cat}
library(knitr)
library(ggplot2)
library(dplyr)
library(corrplot)
library(gridExtra)
df.init=data.frame(Levels = sapply(df,function(x)(length(unique(x)))),
                   Categorical = sapply(df,is.factor),
                   Missing = sapply(df,function(x)(sum(is.na(x)))))

kable(df.init, output = FALSE)

df$CarInsurance=factor(df$CarInsurance)

```

Target variable distribution
```{r tar}
table(df$CarInsurance)
```
````

```
<h3><center> Univariate analysis </center></h3>

```{r uni1,fig.align="center",fig.width=14,fig.height=14}
g1 = ggplot(df,aes(x=Age))+
geom_bar(stat ="count",aes(fill=factor(CarInsurance)))+
xlab("Age of client")+
guides(fill=guide_legend(title="Car Insurance"))+
theme_dark()+ggtitle("Distribution of age")

g2 = ggplot(df,aes(x=Job))+
geom_bar(stat ="count",aes(fill=factor(CarInsurance)))+
xlab("Job of client")+
guides(fill=guide_legend(title="Car Insurance"))+
theme_dark()+ggtitle("Client job")+theme(axis.text.x=element_text(angle=60, hjust=1))

g3 = ggplot(df,aes(x=Education))+
geom_bar(stat ="count",aes(fill=factor(CarInsurance)))+
xlab("Education of client")+
guides(fill=guide_legend(title="Car Insurance"))+
theme_dark()+ggtitle("Education level")

g4 = ggplot(df,aes(x=sign(Balance)*abs(Balance)^(1/3),color=factor(CarInsurance)))+
geom_density()+
xlab("Average balance with cube root transformation")+
guides(fill=guide_legend(title="Car Insurance"))+
theme_dark()+ggtitle("Average balance distribution")

g5 = ggplot(df,aes(x=factor(Default)))+
geom_bar(stat ="count",aes(fill=factor(CarInsurance)))+
xlab("Credit Default")+
guides(fill=guide_legend(title="Car Insurance"))+
theme_dark()+ggtitle("Credit default")

g6 = ggplot(df,aes(x=factor(HHInsurance)))+
geom_bar(stat ="count",aes(fill=factor(CarInsurance)))+
xlab("Insurance")+
guides(fill=guide_legend(title="Car Insurance"))+
theme_dark()+ggtitle("Is Household insured")

g7 = ggplot(df,aes(x=factor(CarLoan)))+
geom_bar(stat ="count",aes(fill=factor(CarInsurance)))+
xlab("Car Loan")+
guides(fill=guide_legend(title="Car Insurance"))+
theme_dark() + ggtitle("Does client have car loan")

g8 = ggplot(df,aes(x=LastContactMonth))+
geom_bar(stat ="count",aes(fill=factor(CarInsurance)))+
xlab("Month")+
guides(fill=guide_legend(title="Car Insurance"))+
theme_dark() +ggtitle("Last month of contact")

g9 = ggplot(df,aes(x=factor(LastContactDay)))+
geom_bar(stat ="count",aes(fill=factor(CarInsurance)))+
xlab("Day")+
guides(fill=guide_legend(title="Car Insurance"))+
theme_dark() +ggtitle("Last day of contact")+theme(axis.text.x=element_text(angle=60, hjust=1))

df$duration = difftime(strptime(df$CallEnd,format = "%H:%M:%S"),
                       strptime(df$CallStart,format = "%H:%M:%S"),units="mins")
g10 = ggplot(df,aes(x=duration))+
geom_bar(stat = "identity",binwidth = 5,aes(fill=factor(CarInsurance)))+
xlab("Duration")+
guides(fill=guide_legend(title="Car Insurance"))+
theme_dark() +ggtitle("Call Duration")

grid.arrange(g1,g2,g3,g4,g5,g6,g7,g8,g9,g10,ncol=2)

```

1. **Age** : Most population is between the age of 25 to 60. The distribution of whether car insurance was brought

2. **Job** : Students, unemployed and retired individuals tend to buy car insurance more often than not

3. **Education**

4. **Average Balance** : With a cube root transformation to normalize the variable. the class distribution looks pr
```

```
5. **Credit Default** : The distribution looks pretty similar for both classes

6. **Insurance** : Uninsured households are more likely to buy car insurance

7. **Car Loan**

8. **Month of Contact** : Individuals are more likely to buy car insurance during months

9. **Day of Contact** : Individuals are more likely to buy insurance at the start of the month

10. **Call Duration** Longer calls lead to more purchases

<h3><center> Multivariate Analysis </h3></center>

1. **Number of Contacts made in previous campaign and Current campaign**

```{r uni10,fig.align="center",fig.width=4,fig.height=4}
corrplot(corr = cor(df[,c("NoOfContacts","PrevAttempts")]))
```

There is no correlation between the number of contacts made in previous campaigns and current campaigns. Let's look

```{r uni11,fig.align="center",fig.width=12,fig.height=8}
ggplot(df,aes(x=NoOfContacts,y=PrevAttempts,color=factor(CarInsurance)))+
geom_point()+geom_jitter()+geom_smooth()+
facet_wrap(cut(DaysPassed,breaks = 8)~CarInsurance,ncol = 7)+
guides(fill=guide_legend(title="Car Insurance"))+
theme_dark()
```

It can be inferred from the above plots, higher previous contacts and lesser duration between campaigns lead to hig

More to come ! :D
```

Data

Data Sources

- Car Insurance Cold Calls
  - carInsurance_test.csv — 19 columns
  - carInsurance_train.csv — 19 columns
  - DSS_DMC_Description.pdf

Car Insurance Cold Calls

We help the guys and girls at the front to get out of Cold Call Hell

Last Updated: 2 years ago (Version 1)

About this Dataset

Introduction

Here you find a very simple, beginner-friendly data set. No sparse matrices, no fancy tools needed to understand what's going on. Just a couple of rows and columns. Super simple stuff. As explained below, this data set is used for a competition. As it turns out, this competition tends to reveal a common truth in data science: KISS - Keep It Simple Stupid

What is so special about this data set is, given it's simplicity, it pays off to use "simple" classifiers as well. This year's competition was won by a C5.0 . Can you do better?

Description

We are looking at cold call results. Turns out, same salespeople called existing insurance customers up and tried to sell car insurance. What you have are details about the called customers.

Their age, job, marital status, whether the have home insurance, a car loan, etc. As I said, super simple.

What I would love to see is some of you applying some crazy XGBoost classifiers, which we can square off against some logistic regressions. It would be curious to see what comes out on top. Thank you for your time, I hope you enjoy using the data set.

Acknowledgements

Thanks goes to the Decision Science and Systems Chair of

Run Info

Succeeded	True	Run Time	33.5 seconds
Exit Code	0	Queue Time	0 seconds
Docker Image Name	kaggle/rstats (Dockerfile)	Output Size	0
Timeout Exceeded	False	Used All Space	False
Failure Message			

Log

Download Log

```
Time  Line #  Log Message
17.8s      1
             processing file: script.Rmd
17.8s      2   |                                                                      |
            |   0%  |                                                                |.....
            |   7%
17.8s      3  label: setup (with options)
17.9s      4  List of 1
17.9s      5   $ include: logi FALSE

17.9s      6   |                                                                      |.........
            |  14%
17.9s      7    ordinary text without R code

                |                                                                    |.............
            |  21%
18.1s      8  label: shape
18.5s      9   |                                                                      |.................
            |  29%
               inline R code fragments

18.5s     10   |
            |.....................                                                 |  36%
18.5s     11  label: cat
19.7s     12
            Attaching package: 'dplyr'

19.8s     13  The following objects are masked from 'package:stats':

                filter, lag

19.8s     14  The following objects are masked from 'package:base':

                intersect, setdiff, setequal, union

19.9s     15
            Attaching package: 'gridExtra'

19.9s     16  The following object is masked from 'package:dplyr':

                combine

20.0s     17   |
            |.........................                                             |  43%
20.0s     18    ordinary text without R code

                |
            |.............................                                         |  50%
20.0s     19  label: tar
20.0s     20   |
```

```
                |.....................................                 |  57%
20.0s      21   ordinary text without R code

               |
                |.........................................             |  64%
               label: uni1 (with options)
20.0s      22  List of 3
                $ fig.align : chr "center"
                $ fig.width :
20.0s      23   num 14
                $ fig.height: num 14

24.2s      24  Don't know how to automatically pick scale for object of type difftime. Defaulting to continuous.
27.1s      25    |
                |.............................................         |  71%
27.1s      26   ordinary text without R code

               |
                |...................................................   |  79%
               label: uni10 (with options)
               List of 3
                $ fig.align : chr "center"
                $ fig.width : num 4
                $ fig.height:
27.1s      27   num 4

27.5s      28    |
                |.....................................................|  86%
27.5s      29   ordinary text without R code

               |
                |.....................................................|  93%
               label: uni11 (with options)
               List of 3
                $ fig.align : chr "center"
                $ fig.width : num 12
                $ fig.height: num 8

27.6s      30  `geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'
32.7s      31    |
                |.....................................................| 100%
32.7s      32   ordinary text without R code

32.7s      33
32.7s      34  output file: /kaggle/working/script.knit.md

32.9s      35  /usr/local/bin/pandoc +RTS -K512m -RTS /kaggle/working/script.utf8.md --to html --from
               markdown+autolink_bare_uris+ascii_identifiers+tex_math_single_backslash --output
               /kaggle/working/__results__.html --smart --email-obfuscation none --standalone --section-divs --
               template /usr/local/lib/R/site-library/rmarkdown/rmd/h/default.html --no-highlight --variable
               highlightjs=1 --variable 'theme:bootstrap' --include-in-header /tmp/RtmphZoqIe/rmarkdown-
               str14b6bc324.html --mathjax --variable 'mathjax-url:https://mathjax.rstudio.com/latest/MathJax.js?
               config=TeX-AMS-MML_HTMLorMML'
32.9s      36
               Output created: __results__.html
32.9s      37  There were 11 warnings (use warnings() to see them)
32.9s      38
32.9s      40  Complete. Exited with code 0.
```

Comments (0)

Please sign in to leave a comment.