Predictive analysis of client subscribed for term deposit

Fox325

6/11/2021

## Basic data exploration.

Data summary

|  |  |
| --- | --- |
| Name | df |
| Number of rows | 4521 |
| Number of columns | 17 |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |  |
| Column type frequency: |  |
| character | 10 |
| numeric | 7 |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |  |
| Group variables | None |

**Variable type: character**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| skim\_variable | n\_missing | complete\_rate | min | max | empty | n\_unique | whitespace |
| job | 0 | 1 | 6 | 13 | 0 | 12 | 0 |
| marital | 0 | 1 | 6 | 8 | 0 | 3 | 0 |
| education | 0 | 1 | 7 | 9 | 0 | 4 | 0 |
| default | 0 | 1 | 2 | 3 | 0 | 2 | 0 |
| housing | 0 | 1 | 2 | 3 | 0 | 2 | 0 |
| loan | 0 | 1 | 2 | 3 | 0 | 2 | 0 |
| contact | 0 | 1 | 7 | 9 | 0 | 3 | 0 |
| month | 0 | 1 | 3 | 3 | 0 | 12 | 0 |
| poutcome | 0 | 1 | 5 | 7 | 0 | 4 | 0 |
| y | 0 | 1 | 2 | 3 | 0 | 2 | 0 |

**Variable type: numeric**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| skim\_variable | n\_missing | complete\_rate | mean | sd | p0 | p25 | p50 | p75 | p100 | hist |
| age | 0 | 1 | 41.17 | 10.58 | 19 | 33 | 39 | 49 | 87 | ▅▇▅▁▁ |
| balance | 0 | 1 | 1422.66 | 3009.64 | -3313 | 69 | 444 | 1480 | 71188 | ▇▁▁▁▁ |
| day | 0 | 1 | 15.92 | 8.25 | 1 | 9 | 16 | 21 | 31 | ▆▆▇▅▆ |
| duration | 0 | 1 | 263.96 | 259.86 | 4 | 104 | 185 | 329 | 3025 | ▇▁▁▁▁ |
| campaign | 0 | 1 | 2.79 | 3.11 | 1 | 1 | 2 | 3 | 50 | ▇▁▁▁▁ |
| pdays | 0 | 1 | 39.77 | 100.12 | -1 | -1 | -1 | -1 | 871 | ▇▁▁▁▁ |
| previous | 0 | 1 | 0.54 | 1.69 | 0 | 0 | 0 | 0 | 25 | ▇▁▁▁▁ |

This dataset consist of 17 variables with 4521 observations. Among this 17 variables; 7 are numeric and 10 are character variables.

### Short discription about the variables

* age (numeric)
* job : type of job (categorical)
* marital : marital status (categorical)
* education (categorical)
* default: has credit in default? (binary)
* balance: average yearly balance, in euros (numeric)
* housing: has housing loan? (binary)
* loan: has personal loan? (binary)
* contact: contact communication type (categorical)
* day: last contact day of the month (numeric)
* month: last contact month of year (categorical)
* duration: last contact duration, in seconds (numeric)
* campaign: number of contacts performed during this campaign and for this client (numeric)
* pdays: number of days that passed by after the client was last contacted from a previous campaign (numeric, -1 means client was not previously contacted)
* previous: number of contacts performed before this campaign and for this client (numeric)
* poutcome: outcome of the previous marketing campaign (categorical)

Output variable (desired target): - y - has the client subscribed a term deposit? (binary: “yes”,“no”)

Table continues below

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| age | job | marital | education | default | balance | housing | loan |
| 30 | unemployed | married | primary | no | 1787 | no | no |
| 33 | services | married | secondary | no | 4789 | yes | yes |
| 35 | management | single | tertiary | no | 1350 | yes | no |
| 30 | management | married | tertiary | no | 1476 | yes | yes |
| 59 | blue-collar | married | secondary | no | 0 | yes | no |
| 35 | management | single | tertiary | no | 747 | no | no |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| contact | day | month | duration | campaign | pdays | previous | poutcome | y |
| cellular | 19 | oct | 79 | 1 | -1 | 0 | unknown | no |
| cellular | 11 | may | 220 | 1 | 339 | 4 | failure | no |
| cellular | 16 | apr | 185 | 1 | 330 | 1 | failure | no |
| unknown | 3 | jun | 199 | 4 | -1 | 0 | unknown | no |
| unknown | 5 | may | 226 | 1 | -1 | 0 | unknown | no |
| cellular | 23 | feb | 141 | 2 | 176 | 3 | failure | no |

### Data cleaning and feature engineering.

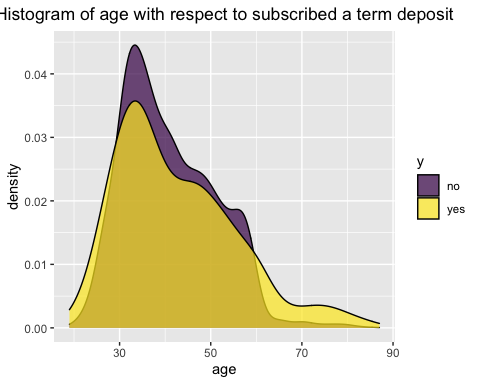
In this tase we unite the day and month variable together and then find the distance between days from decamber 31 and replace the value of -1 for variable pdays by 900. After the cleaning stuffs the data set become. Then change all the character variable type to the factor type.

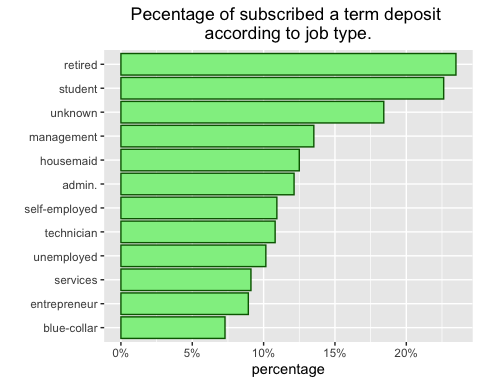
Table continues below

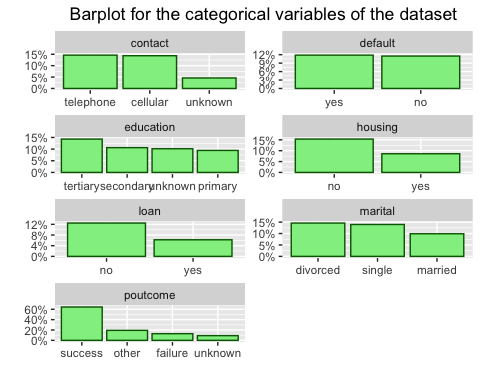
|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| age | job | marital | education | default | balance | housing | loan |
| 30 | unemployed | married | primary | no | 1787 | no | no |
| 33 | services | married | secondary | no | 4789 | yes | yes |
| 35 | management | single | tertiary | no | 1350 | yes | no |
| 30 | management | married | tertiary | no | 1476 | yes | yes |
| 59 | blue-collar | married | secondary | no | 0 | yes | no |
| 35 | management | single | tertiary | no | 747 | no | no |

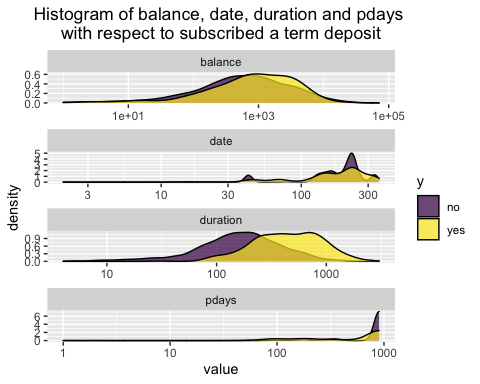
|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| contact | date | duration | campaign | pdays | previous | poutcome | y |
| cellular | 73 | 79 | 1 | 900 | 0 | unknown | no |
| cellular | 234 | 220 | 1 | 339 | 4 | failure | no |
| cellular | 259 | 185 | 1 | 330 | 1 | failure | no |
| unknown | 211 | 199 | 4 | 900 | 0 | unknown | no |
| unknown | 240 | 226 | 1 | 900 | 0 | unknown | no |
| cellular | 312 | 141 | 2 | 176 | 3 | failure | no |

### Explonatory data analysis

 So for the higher age group the tendency of subscribed a term deposit is higher.

 So student and retired people use to subscribed a term deposit more in percentage than other people.

 So from this graph we can see that the percentage of subscribed of term deposit increase for the contact variable telephone and cellular group; education variable tertiary group; housing variable no group; loan variable no group; marital variable divorced and single group; poutcome variable success group.

 From this plot we can see that lower values for balance, date and duration variables has less odds to subscribed a term deposit. On the other hand lower value of variable pdays has higer oddas to subscribe a term deposit.

### Models

We will fit some models like - Logistic Regression - Elastic net - Decision Tree - Random Forest - Xgboost

### Class imbalance

|  |  |
| --- | --- |
| y | n |
| no | 4000 |
| yes | 521 |

This data set has a class imbalance problem. We will down sample the no class to reduce the effect.

|  |  |
| --- | --- |
| y | n |
| no | 625 |
| yes | 521 |

After this down sample our sample size become 1146 in which no group has 625 observations and yes group has 521 observations.

### Spliting data into train and test set

We have 1146 observations in our data set we will split the data in 80% train and 20% test set we get 916 observation in the train set and 230 observations in the test set.

### Preprocessing of the data

Before fitting those data to the model we need to pre-processing. For logistic regression, elasticnet regression we need to scale the numeric data and make the categorical variable to dummy variable. For the tree based model we don’t need any type of pre-processing.

### Hyper-parameters

It is necessary to find out the optimal value of the hyper-parameter to ensure the maximum performance by the model. So for this task we use 10 fold cross validation and we randomly choose some hyper-parameter values in the parameter space and then find out that which set is performing the best. Then we select the values of the hyper-parameter set and fit our model on the full training dataset on this settings.

While we are using different types of models all models expect logistic regression has hyper-parameters. We will use 10 fold cross validation to tune those hyper-parameters.

From this plot we can see that all model perform equally. no we will check for the numerical evidence.

From this we can say that the area under curve for the random forest model is maximum. hence our selected model is random forest.

### Performance metric for random forest

So from this table we can see that the accuracy of the random forest model is .8 that is it can identify 80% of the cases correctly. The precision is .87 that is 87% of the cases classified as “yes” by the model is correct. The recall is .797 that is about 80% of all true “yes” class can be classified correctly by the model.

### Appendix

` ##Basic data exploration. skimr::skim(df)

##Data cleaning df <-df %>% unite(“date”,day:month,sep = “-”) %>% mutate(date = paste0(date,“-2020”), date = dmy(date), date = dmy(“31-12-2020”)-date, date = as.numeric(date), pdays = ifelse(pdays == -1,900,pdays), y = factor(y,c(“no”,“yes”))) %>% mutate\_if(is.character,as.factor)

##Explonatory data analysis df %>% ggplot(aes(age,fill = y)) + geom\_density()+ labs(title = “Histogram of age with respect to subscribed a term deposit”) + scale\_fill\_viridis\_d(alpha = .7)

df %>% group\_by(job) %>% count(y) %>% mutate(prop= n/sum(n)) %>% filter(y == “yes”) %>% ungroup() %>% mutate(job = reorder(job,prop)) %>% ggplot(aes(y=job,prop)) + geom\_col(col=“darkgreen”,fill=“lightgreen”) + scale\_x\_continuous(labels = scales::percent\_format(accuracy = 1)) + labs(title = “Pecentage of subscribed a term deposit to job type.”,x=“percentage”,y="")

df %>% select(marital:contact,poutcome,y,-balance) %>% pivot\_longer(-y) %>% group\_by(name,value) %>% count(y) %>% mutate(prop = n/sum(n)) %>% filter(y == “yes”) %>% ungroup() %>% mutate(value = tidytext::reorder\_within(value,by = -prop,within = name)) %>% ggplot(aes(value,prop)) + geom\_col(col = “darkgreen”,fill = “lightgreen”)+ tidytext::scale\_x\_reordered()+ facet\_wrap(~name,scales = “free”,ncol = 2) + scale\_y\_continuous + labs(title = “Barplot for the categorical variables of the dataset”)

df %>% select(balance,date, duration, pdays,y) %>% pivot\_longer(-y) %>% ggplot(aes(value,fill = y)) + geom\_density()+ facet\_wrap(~name,ncol = 1,scales = “free”)+ scale\_x\_log10()+ scale\_fill\_viridis\_d(alpha = .7) + labs(title = “Histogram of balance, date, duration and pdays respect to subscribed a term deposit”)

##Class imbalance df %>% count(y) %>% pander()

##Model specifications lr\_specs <- logistic\_reg(mode = “classification”) %>% set\_engine(“glm”)

enet\_specs <- logistic\_reg(mode = “classification”,penalty = tune(),mixture = tune()) %>% set\_engine(“glmnet”)

tree\_specs <- decision\_tree(mode = “classification”,cost\_complexity = tune(),tree\_depth = tune(),min\_n = tune()) %>% set\_engine(“rpart”)

rf\_specs <- rand\_forest(mode = “classification”,trees = 1000,min\_n = tune()) %>% set\_engine(“ranger”)

xgb\_specs <- boost\_tree(mode = “classification”,trees = 1000,min\_n = tune(),learn\_rate = tune(),tree\_depth = tune()) %>% set\_engine(“xgboost”)

##Cross validation set.seed(10) df\_cv <- vfold\_cv(df\_train)

##Function for model fitting model\_fitting <- function(name = “lr”,model = enet\_specs, recipe = lr\_rec) {

if (name != "logistic Regression") {  
 set.seed(1234)  
 parms\_grid <-  
 model %>%  
 parameters() %>%  
 grid\_latin\_hypercube(size = 10 \* nrow(.))  
   
 wf <-  
 workflow() %>%  
 add\_model(model) %>%  
 add\_recipe(recipe)  
   
 resample\_results <-  
 tune\_grid(  
 object = wf,  
 grid = parms\_grid,  
 resamples = df\_cv,  
 metrics = metric\_set(accuracy)  
 )  
   
 fit <-  
 wf %>%  
 finalize\_workflow(select\_best(resample\_results)) %>%  
 fit(df\_train)  
 }  
   
 else  
 fit <-  
 workflow() %>%  
 add\_model(model) %>%  
 add\_formula(y ~ .) %>%   
 fit(df\_train)  
   
 print(name)  
 return(fit)  
}

##Fitting the model fit <- tibble(id = 1:5) %>% mutate( name = c( “logistic Regression”, “Elastic net”, “Decision tree”, “Randomfprest”, “Xg boost” ), model = list(lr\_specs, enet\_specs, tree\_specs, rf\_specs, xgb\_specs), recipe = list(lr\_rec, lr\_rec, tree\_rec, tree\_rec, lr\_rec), fit = pmap( list(name, model, recipe), ~ model\_fitting( name = ..1, model = ..2, recipe = ..3 ) ) )

fit <- fit %>% transmute( name, fit, split = list(df\_split), last\_fit = map2( fit, split, ~ last\_fit(.x, .y)) )

##Plotting roc-auc fit %>% select(-c(fit,split)) %>% unnest(last\_fit) %>% unnest(.predictions) %>% select(name,.pred\_no,y) %>% group\_by(name) %>% roc\_curve(truth = y, .pred\_no) %>% autoplot()

##Roc-auc table fit %>% select(-c(fit, split)) %>% unnest(last\_fit) %>% unnest(.predictions) %>% select(name, .pred\_no, y) %>% group\_by(name) %>% roc\_auc(truth = y, .pred\_no) %>% arrange(-.estimate) %>% pander()

##Random forest performance fitfit[[4]] %>% augment(testing(df\_split)) %>% precision(truth = y,.pred\_class)) %>% bind\_rows(fit$fit[[4]] %>% augment(testing(df\_split)) %>% recall(truth = y,.pred\_class)) %>% pander() `