Train and evaluate models with tidymodels

2022-11-19

Contents

Introduction

The goal of this analysis is to **predict with precision the probability of customer churn** and consequently *help in customer retention and mitigation of the risks related to customer churn.*

Business Value

Below are some the business values that this analysis will create:

1. Reduce churn rate by detecting early clients who are about to churn.
2. Increase customer retention.
3. Understand customer base better based on their activity.

Data-set

The data set used in this project was sourced from client transaction data recorded since the inception of the banking institution.Based in this data set,informative feature sets were created which were subsequently used in the predictive model.

Inclusion Criteria

The clients records included in this study were selected from those who were determined to have done at least one transaction within the banking platform.

Exclusion Criteria

Clients who on boarded the digital platform of the bank and did not make any transaction.

Project setup and data loading

```
library(tidymodels);library(feather);library(tidyverse);library(janitor);library(flextable);library(Sma
dataset <- readxl::read_excel("E:/New folder/denis/old/data/rawdata/base_.xlsx") %>%
  mutate_at(c("Min_LoanTaken","Max_LoanTaken"),as.numeric)
```

Exploratory Data Analysis

Data Overview

```
ExpData(dataset) %>%
  flextable() %>%
  autofit()
```

```
## Warning: Warning: fonts used in ‘flextable‘ are ignored because the ‘pdflatex‘
## engine is used and not ‘xelatex‘ or ‘lualatex‘. You can avoid this warning
## by using the ‘set_flextable_defaults(fonts_ignore=TRUE)‘ command or use a
## compatible engine by defining ‘latex_engine: xelatex‘ in the YAML header of the
## R Markdown document.
```

Descriptions	Value
Sample size (nrow)	32506
No. of variables (ncol)	39
No. of numeric/interger variables	36
No. of factor variables	0
No. of text variables	1
No. of logical variables	0
No. of identifier variables	1
No. of date variables	2
No. of zero variance variables (uniform)	1
%. of variables having complete cases	66.67% (26)
%. of variables having >0% and <50% missing cases	33.33% (13)
%. of variables having >=50% and <90% missing cases	0% (0)
%. of variables having >=90% missing cases	0% (0)

Observations and inferences.

- The data set contains 32,506 records of customer transactions with 39 features.
- Only 66.67% of the cases have complete records meaning the remaining 33.33% have missing values.This is normal with most real life data sets that contain missing values.

Uni-variate Data Analysis.

Target Variable

```
dataset %>%
  group_by(Churn) %>%
```

```
  tally(name = "Count") %>%
  mutate(Percent = paste(round(Count/sum(Count),2)*100,"%")) %>%
  adorn_totals('row') %>%
  flextable() %>%
  autofit()
```

```
## Warning: Warning: fonts used in ‘flextable‘ are ignored because the ‘pdflatex‘
## engine is used and not ‘xelatex‘ or ‘lualatex‘. You can avoid this warning
## by using the ‘set_flextable_defaults(fonts_ignore=TRUE)‘ command or use a
## compatible engine by defining ‘latex_engine: xelatex‘ in the YAML header of the
## R Markdown document.
```

Churn	Count	Percent
Churn	4,315	13 %
Not Churn	28,191	87 %
Total	32,506	-

Observations and Inferences

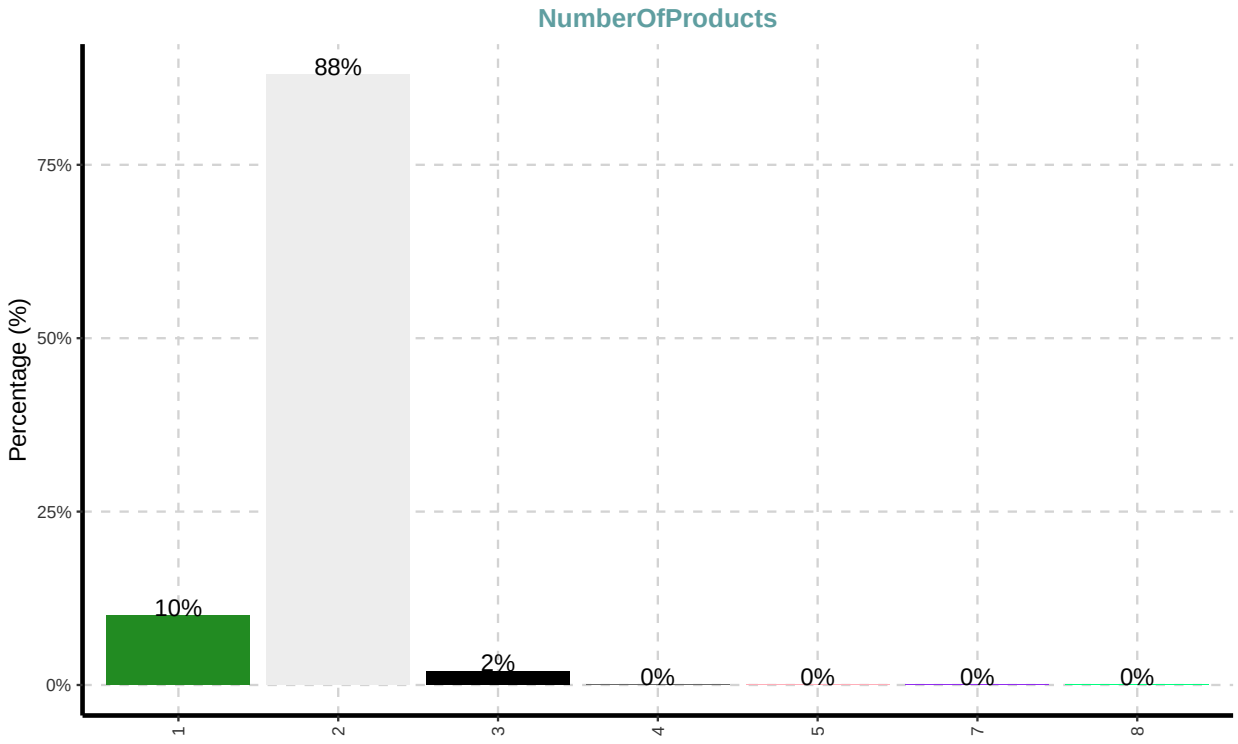
- The institution has suffered a 13% churn rate overtime.

Explanatory Variables

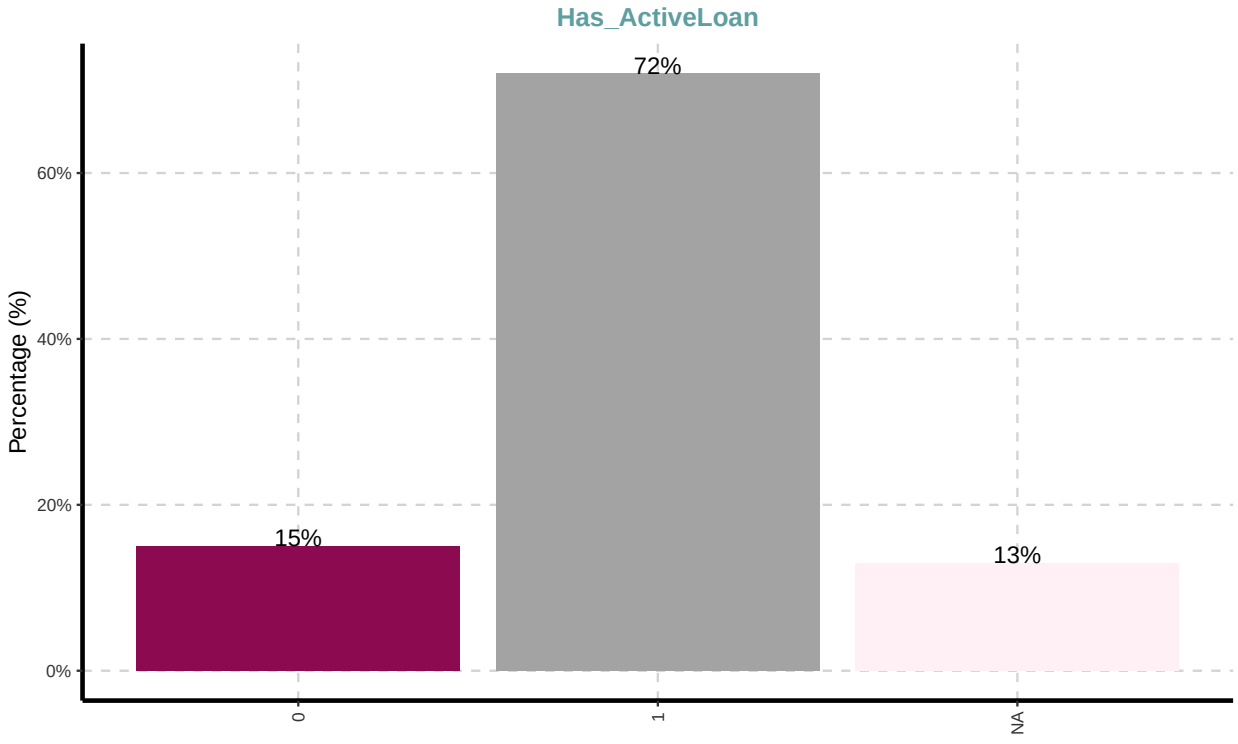
```
dataset %>% select(-Churn) %>% ExpCatViz()
```

Categorical Variables

```
## [[1]]
```


NumberOfProducts
Percentage (%)
75%
50%
25%
0%
88%
10%
2%
0%
0%
0%
0%
1
2
3
4
5
7
8

```
##
## [[2]]
```


Has_ActiveLoan
Percentage (%)
60%
40%
20%
0%
72%
15%
13%
0
1
NA

```
ExpCTable(dataset %>% select(NumberOfProducts,Has_ActiveLoan)) %>%
  flextable() %>% autofit()
```

```
## Warning: Warning: fonts used in ‘flextable‘ are ignored because the ‘pdflatex‘
## engine is used and not ‘xelatex‘ or ‘lualatex‘. You can avoid this warning
## by using the ‘set_flextable_defaults(fonts_ignore=TRUE)‘ command or use a
## compatible engine by defining ‘latex_engine: xelatex‘ in the YAML header of the
## R Markdown document.
```

Variable	Valid	Frequency	Percent	CumPercent
NumberOfProducts	1	3,128	9.62	9.62
NumberOfProducts	2	28,540	87.80	97.42
NumberOfProducts	3	789	2.43	99.85
NumberOfProducts	4	44	0.14	99.99
NumberOfProducts	5	3	0.01	100.00
NumberOfProducts	7	1	0.00	100.00
NumberOfProducts	8	1	0.00	100.00
NumberOfProducts	TOTAL	32,506		
Has_ActiveLoan	0	4,788	14.73	14.73
Has_ActiveLoan	1	23,403	72.00	86.73
Has_ActiveLoan	NA	4,315	13.27	100.00
Has_ActiveLoan	TOTAL	32,506		

Observation & inferences

- *Number of products* : Majority of clients (88%) subscribed to at least two of the financial institution's digital products while a very small percentage subscribed to more than 2 products.
- *Has_Active_Loan* : 72% of the clients were still servicing loans with the institution.

```
num_vars <- dataset %>% select(-c(Churn,ClientID,NumberOfProducts,Has_ActiveLoan))
num_vars %>%
  # select(Tenure,No_of_transactions) %>%
  ExpNumStat() %>% select(-c(Group,NegInf,PosInf,Per_of_Missing,CV,nNeg,nZero,NA_Value,nPos)) %>%
  flextable() %>% autofit()
```

Numeric Variables

```
## Warning: Warning: fonts used in ‘flextable‘ are ignored because the ‘pdflatex‘
## engine is used and not ‘xelatex‘ or ‘lualatex‘. You can avoid this warning
## by using the ‘set_flextable_defaults(fonts_ignore=TRUE)‘ command or use a
## compatible engine by defining ‘latex_engine: xelatex‘ in the YAML header of the
## R Markdown document.
```

Vname	TN	sum	min	max
ave_LoanTaken	32,506	219,381,812.83	992.250	1,200,000.000

Vname	TN	sum	min	max
Average_DepositsMade	32,506	48,990,493.92	0.000	275,108.800
Days_Since_LastTransaction	32,506	13,512,406.00	2.000	751.000
DaysSince_Last_Loan	32,506	14,090,399.00	1.000	758.000
deposits_12	32,506	19,756,971.75	0.000	4,000,000.000
deposits_3	32,506	17,515,829.06	0.000	4,000,000.000
deposits_6	32,506	17,712,001.06	0.000	4,000,000.000
First_Loan_Last_Loan_diff	32,506	-11,016,050.64	-600,000.000	1,272,500.000
First_Loan_Last_Loan_Ratio	32,506	27,700.62	0.032	159.439
FirstLoanTaken	32,506	218,767,007.16	590.000	1,300,000.000
last_deposit_amount	32,506	171,281,930.14	0.000	1,434,950.000
Last_Loan_First_Loan_Ratio	32,506	32,030.50	0.006	31.000
last_withdrawal_amount	32,506	406,870,976.06	0.000	2,480,237.000
LastLoanTaken	32,506	229,783,057.80	1,100.000	1,200,000.000
Max_ArrearsDays	32,506	10,990,129.00	-28.000	615.000
Max_DepositsMade	32,506	255,172,590.62	0.000	4,000,000.000
Max_LoanTaken	32,506	221,291,543.12	590.000	1,200,000.000
Max_WithdrawalMade	32,506	251,339,561.87	0.000	4,000,000.000
Min_LoanTaken	32,506	217,099,535.84	1,100.000	1,250,000.000
no_of_LoanTaken	32,506	78,511.00	1.000	40.000
No_of_transactions	32,506	720,251.00	1.000	3,114.000
No_of_transactions_Deposits	32,506	202,110.00	0.000	939.000
No_of_transactions_Withdrawals	32,506	506,555.00	0.000	2,161.000
Tenure	32,506	636,175.00	0.000	65.000
TotalDepositsMade	32,506	1,325,371,994.74	0.000	15,869,967.980
transactions_made_12	32,506	278,590.00	0.000	665.000
transactions_made_3	32,506	64,502.00	0.000	408.000
transactions_made_6	32,506	128,083.00	0.000	579.000
WithdrawalMade	32,506	1,483,086,187.43	0.000	16,171,410.720
withdrawals_12	32,506	92,470,697.40	0.000	700,000.000
withdrawals_3	32,506	46,664,953.64	0.000	220,000.000
withdrawals_6	32,506	65,115,775.04	0.000	220,000.000

Build models

Let's create a **model specification** for each model we want to try:

To set up your modeling code, consider using the parsnip addin or the usemodels package.

Now let's build a **model workflow** combining each model specification with a data preprocessor:

If your feature engineering needs are more complex than provided by a formula like `sex ~ .`, use a recipe. Read more about feature engineering with recipes to learn how they work.

Evaluate models

These models have no tuning parameters so we can evaluate them as they are. Learn about tuning hyperparameters here.

How did these two models compare?

We can visualize these results using an ROC curve (or a confusion matrix via `conf_mat()`):

These models perform very similarly, so perhaps we would choose the simpler, linear model. The function `last_fit()` *fits* one final time on the training data and *evaluates* on the testing data. This is the first time we have used the testing data.

This object contains a fitted workflow that we can use for prediction.

You can save this fitted `final_wf` object to use later with new data, for example with `readr::write_rds()`.