Data Preparation and Analysis

Akshay Singh, Ravi Teja, Raj Shah

2022-11-21

```
library(plyr)
library(corrplot)
```

```
## corrplot 0.92 loaded
```

```
library(ggplot2)
library(gridExtra)
library(ggthemes)
library(caret)
```

```
## Loading required package: lattice
```

```
library(lattice)
library(MASS)
library(randomForest)
```

```
## randomForest 4.7-1.1

## Type rfNews() to see new features/changes/bug fixes.

##
## Attaching package: 'randomForest'

## The following object is masked from 'package:gridExtra':
##
##     combine

## The following object is masked from 'package:ggplot2':
##
##     margin
```

```
library(party)
```

```
## Loading required package: grid

## Loading required package: mvtnorm

## Loading required package: modeltools

## Loading required package: stats4

##
## Attaching package: 'modeltools'

## The following object is masked from 'package:plyr':
##
##     empty

## Loading required package: strucchange
```

```
## Loading required package: zoo

##
## Attaching package: 'zoo'

## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric

## Loading required package: sandwich
```

```
library(sandwich)
library(rpart)
library(rattle)
```

```
## Loading required package: tibble

## Loading required package: bitops

## Rattle: A free graphical interface for data science with R.
## Version 5.5.1 Copyright (c) 2006-2021 Togaware Pty Ltd.
## Type 'rattle()' to shake, rattle, and roll your data.

##
## Attaching package: 'rattle'

## The following object is masked from 'package:randomForest':
##
##     importance
```

```
library(GoodmanKruskal)
library(e1071)
library(rpart.plot)
library(caTools)
library(class)
```

```
churn <- read.csv('BankChurners.csv')
str(churn)
```

```
## 'data.frame':    10127 obs. of  23 variables:
##  $ CLIENTNUM
##  $ Attrition_Flag
##  $ Customer_Age
##  $ Gender
##  $ Dependent_count
##  $ Education_Level
##  $ Marital_Status
##  $ Income_Category
##  $ Card_Category
##  $ Months_on_book
##  $ Total_Relationship_Count
##  $ Months_Inactive_12_mon
##  $ Contacts_Count_12_mon
##  $ Credit_Limit
##  $ Total_Revolving_Bal
##  $ Avg_Open_To_Buy
##  $ Total_Amt_Chng_Q4_Q1
##  $ Total_Trans_Amt
##  $ Total_Trans_Ct
```

```
##  $ Total_Ct_Chng_Q4_Q1
##  $ Avg_Utilization_Ratio
##  $ Naive_Bayes_Classifier_Attrition_Flag_Card_Category_Contacts_Count_12_mon_Dependent_count_Educati
##  $ Naive_Bayes_Classifier_Attrition_Flag_Card_Category_Contacts_Count_12_mon_Dependent_count_Educati
```

```r
sapply(churn, function(x) sum(is.na(x)))
```

```
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
## Naive_Bayes_Classifier_Attrition_Flag_Card_Category_Contacts_Count_12_mon_Dependent_count_Education_
##
## Naive_Bayes_Classifier_Attrition_Flag_Card_Category_Contacts_Count_12_mon_Dependent_count_Education_
##
```

```r
churn$Naive_Bayes_Classifier_Attrition_Flag_Card_Category_Contacts_Count_12_mon_Dependent_count_Educatio
churn$Naive_Bayes_Classifier_Attrition_Flag_Card_Category_Contacts_Count_12_mon_Dependent_count_Educatio
churn$CLIENTNUM <- NULL
```

```r
str(churn)
```

```
## 'data.frame':    10127 obs. of  20 variables:
##  $ Attrition_Flag          : chr  "Existing Customer" "Existing Customer" "Existing Customer" "Existi
##  $ Customer_Age            : int  45 49 51 40 40 44 51 32 37 48 ...
##  $ Gender                  : chr  "M" "F" "M" "F" ...
##  $ Dependent_count         : int  3 5 3 4 3 2 4 0 3 2 ...
##  $ Education_Level         : chr  "High School" "Graduate" "Graduate" "High School" ...
##  $ Marital_Status          : chr  "Married" "Single" "Married" "Unknown" ...
##  $ Income_Category         : chr  "$60K - $80K" "Less than $40K" "$80K - $120K" "Less than $40K" ...
##  $ Card_Category           : chr  "Blue" "Blue" "Blue" "Blue" ...
##  $ Months_on_book          : int  39 44 36 34 21 36 46 27 36 36 ...
##  $ Total_Relationship_Count: int  5 6 4 3 5 3 6 2 5 6 ...
##  $ Months_Inactive_12_mon  : int  1 1 1 4 1 1 1 2 2 3 ...
##  $ Contacts_Count_12_mon   : int  3 2 0 1 0 2 3 2 0 3 ...
##  $ Credit_Limit            : num  12691 8256 3418 3313 4716 ...
##  $ Total_Revolving_Bal     : int  777 864 0 2517 0 1247 2264 1396 2517 1677 ...
##  $ Avg_Open_To_Buy         : num  11914 7392 3418 796 4716 ...
##  $ Total_Amt_Chng_Q4_Q1    : num  1.33 1.54 2.59 1.41 2.17 ...
##  $ Total_Trans_Amt         : int  1144 1291 1887 1171 816 1088 1330 1538 1350 1441 ...
##  $ Total_Trans_Ct          : int  42 33 20 20 28 24 31 36 24 32 ...
##  $ Total_Ct_Chng_Q4_Q1     : num  1.62 3.71 2.33 2.33 2.5 ...
##  $ Avg_Utilization_Ratio   : num  0.061 0.105 0 0.76 0 0.311 0.066 0.048 0.113 0.144 ...
```

```r
summary(churn)
```

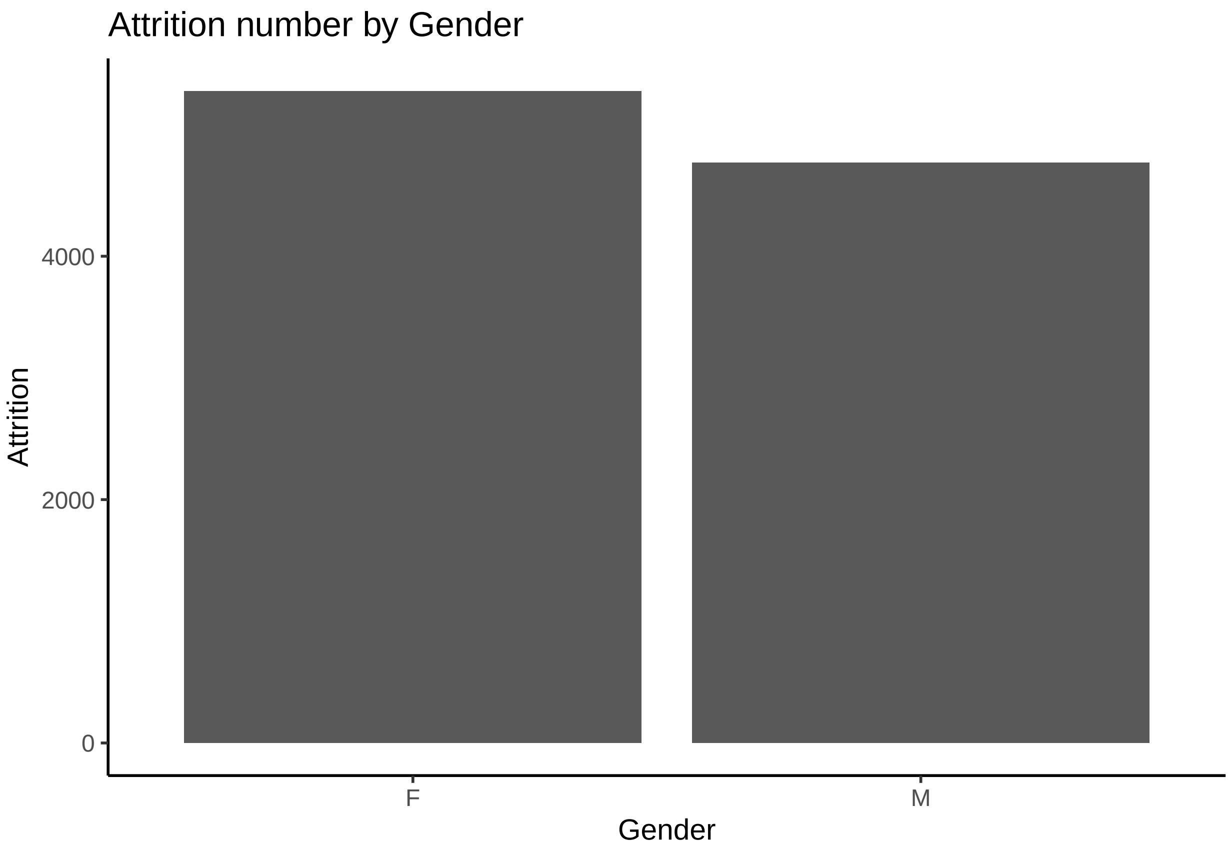
```
##  Attrition_Flag      Customer_Age      Gender          Dependent_count
##  Length:10127       Min.   :26.00   Length:10127       Min.   :0.000
##  Class :character   1st Qu.:41.00   Class :character   1st Qu.:1.000
##  Mode  :character   Median :46.00   Mode  :character   Median :2.000
##                     Mean   :46.33                      Mean   :2.346
##                     3rd Qu.:52.00                      3rd Qu.:3.000
##                     Max.   :73.00                      Max.   :5.000
##  Education_Level    Marital_Status     Income_Category    Card_Category
##  Length:10127       Length:10127       Length:10127       Length:10127
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##  Months_on_book  Total_Relationship_Count Months_Inactive_12_mon
##  Min.   :13.00   Min.   :1.000            Min.   :0.000
##  1st Qu.:31.00   1st Qu.:3.000            1st Qu.:2.000
##  Median :36.00   Median :4.000            Median :2.000
##  Mean   :35.93   Mean   :3.813            Mean   :2.341
##  3rd Qu.:40.00   3rd Qu.:5.000            3rd Qu.:3.000
##  Max.   :56.00   Max.   :6.000            Max.   :6.000
##  Contacts_Count_12_mon  Credit_Limit   Total_Revolving_Bal Avg_Open_To_Buy
##  Min.   :0.000         Min.   : 1438   Min.   :   0        Min.   :    3
##  1st Qu.:2.000         1st Qu.: 2555   1st Qu.: 359        1st Qu.: 1324
##  Median :2.000         Median : 4549   Median :1276        Median : 3474
```

```
##  Mean   :2.455         Mean   : 8632   Mean   :1163        Mean   : 7469
##  3rd Qu.:3.000         3rd Qu.:11068   3rd Qu.:1784        3rd Qu.: 9859
##  Max.   :6.000         Max.   :34516   Max.   :2517        Max.   :34516
##  Total_Amt_Chng_Q4_Q1 Total_Trans_Amt Total_Trans_Ct   Total_Ct_Chng_Q4_Q1
##  Min.   :0.0000       Min.   :  510   Min.   : 10.00   Min.   :0.0000
##  1st Qu.:0.6310       1st Qu.: 2156   1st Qu.: 45.00   1st Qu.:0.5820
##  Median :0.7360       Median : 3899   Median : 67.00   Median :0.7020
##  Mean   :0.7599       Mean   : 4404   Mean   : 64.86   Mean   :0.7122
##  3rd Qu.:0.8590       3rd Qu.: 4741   3rd Qu.: 81.00   3rd Qu.:0.8180
##  Max.   :3.3970       Max.   :18484   Max.   :139.00   Max.   :3.7140
##  Avg_Utilization_Ratio
##  Min.   :0.0000
##  1st Qu.:0.0230
##  Median :0.1760
##  Mean   :0.2749
##  3rd Qu.:0.5030
##  Max.   :0.9990
```

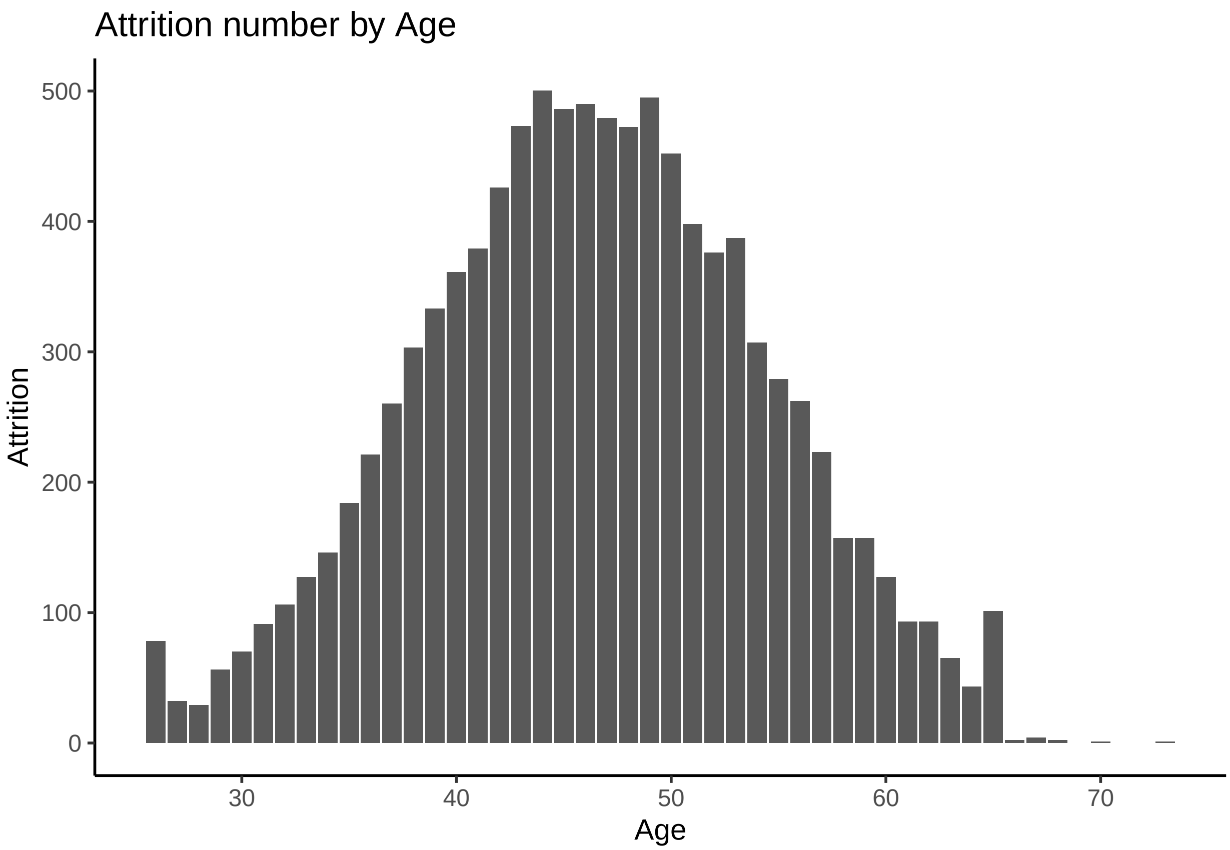
```
print(head(churn[1:3,]))
```

```
##      Attrition_Flag Customer_Age Gender Dependent_count Education_Level
## 1 Existing Customer           45      M               3     High School
## 2 Existing Customer           49      F               5        Graduate
## 3 Existing Customer           51      M               3        Graduate
##   Marital_Status Income_Category Card_Category Months_on_book
## 1        Married     $60K - $80K          Blue             39
## 2         Single  Less than $40K          Blue             44
## 3        Married    $80K - $120K          Blue             36
##   Total_Relationship_Count Months_Inactive_12_mon Contacts_Count_12_mon
## 1                        5                      1                     3
## 2                        6                      1                     2
## 3                        4                      1                     0
##   Credit_Limit Total_Revolving_Bal Avg_Open_To_Buy Total_Amt_Chng_Q4_Q1
## 1        12691                 777           11914                1.335
## 2         8256                 864            7392                1.541
## 3         3418                   0            3418                2.594
##   Total_Trans_Amt Total_Trans_Ct Total_Ct_Chng_Q4_Q1 Avg_Utilization_Ratio
## 1            1144             42               1.625                 0.061
## 2            1291             33               3.714                 0.105
## 3            1887             20               2.333                 0.000
```

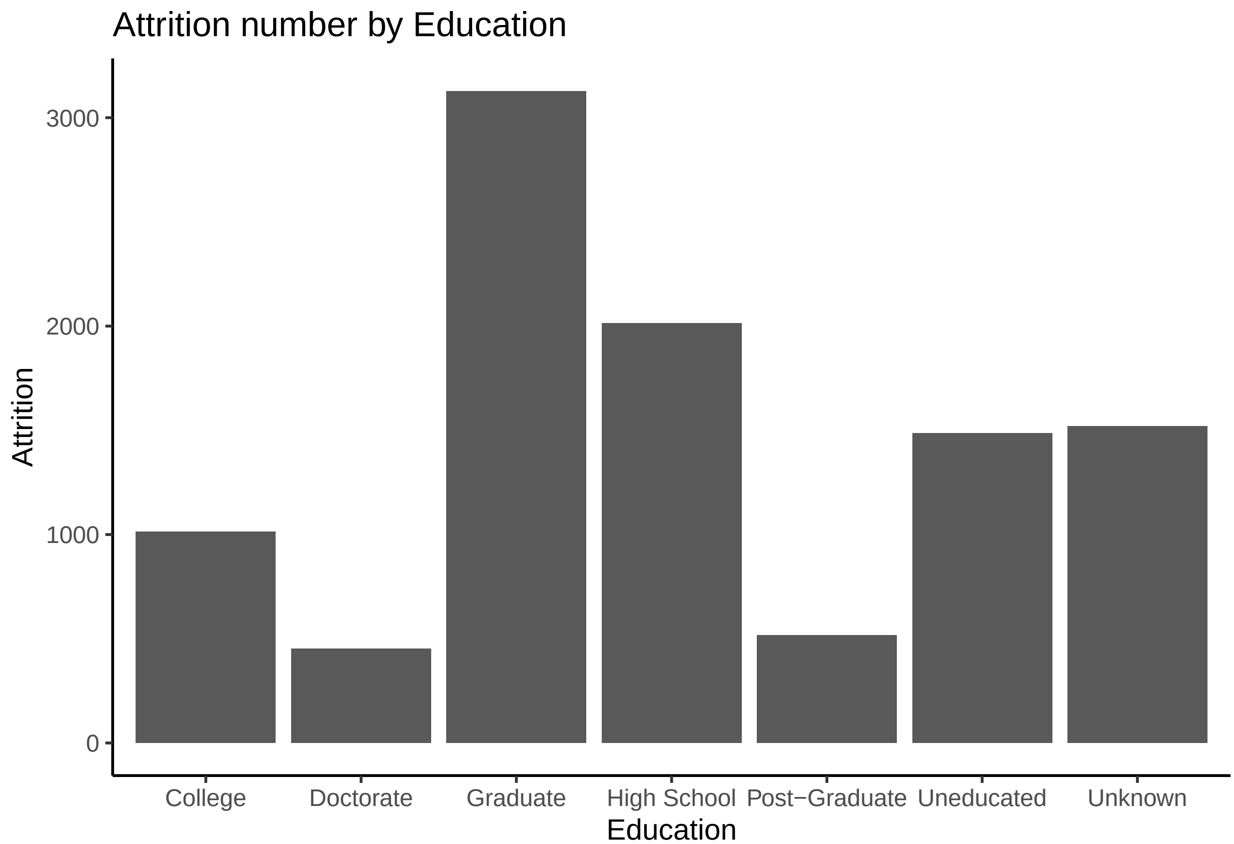
```
#Data Exploration
ggplot(churn, aes(x=Gender)) +
  geom_bar(stat="count") +
  labs(title= "Attrition number by Gender", x= "Gender", y="Attrition") +
  theme_classic() + scale_color_brewer(palette="Set2")
```



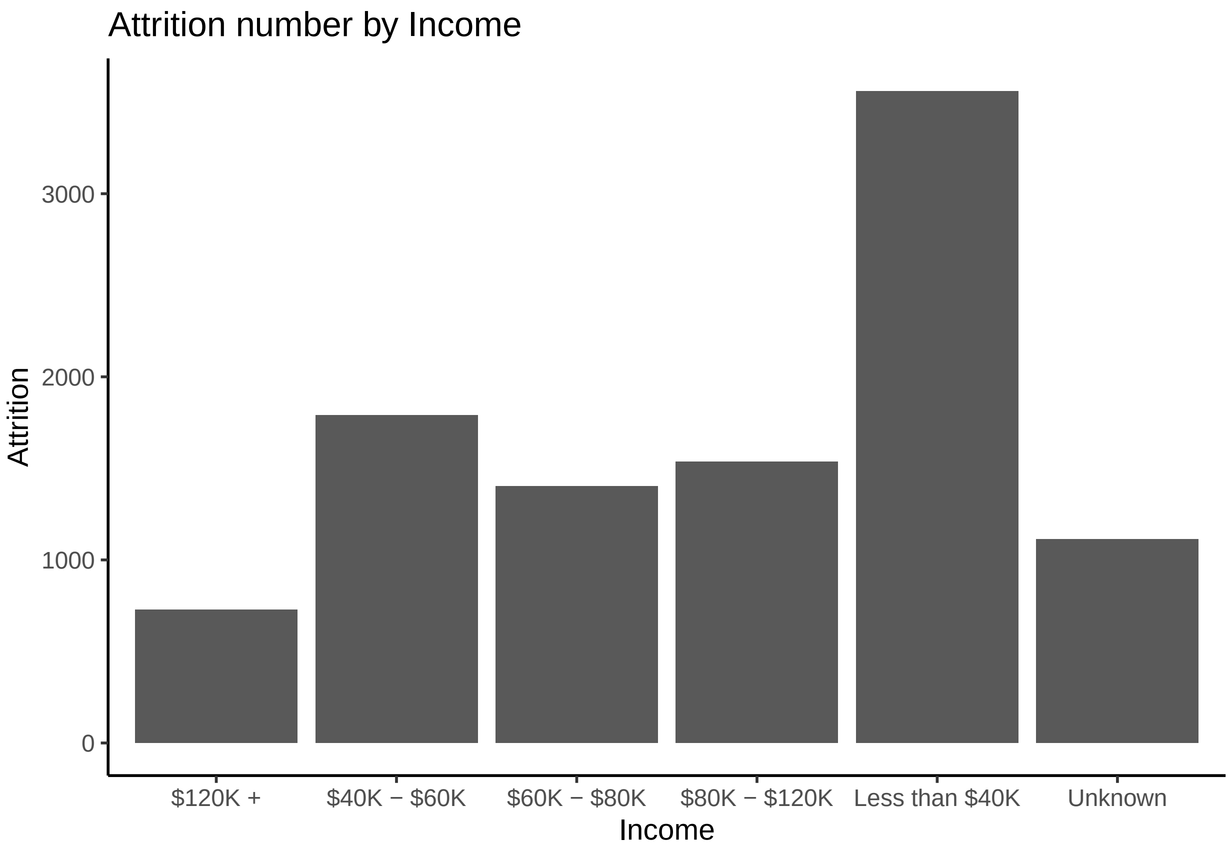

```
ggplot(churn, aes(x=Customer_Age)) +
  geom_bar(stat="count") +
  labs(title= "Attrition number by Age", x= "Age", y="Attrition") +
  theme_classic() + scale_color_brewer(palette="Set2")
```



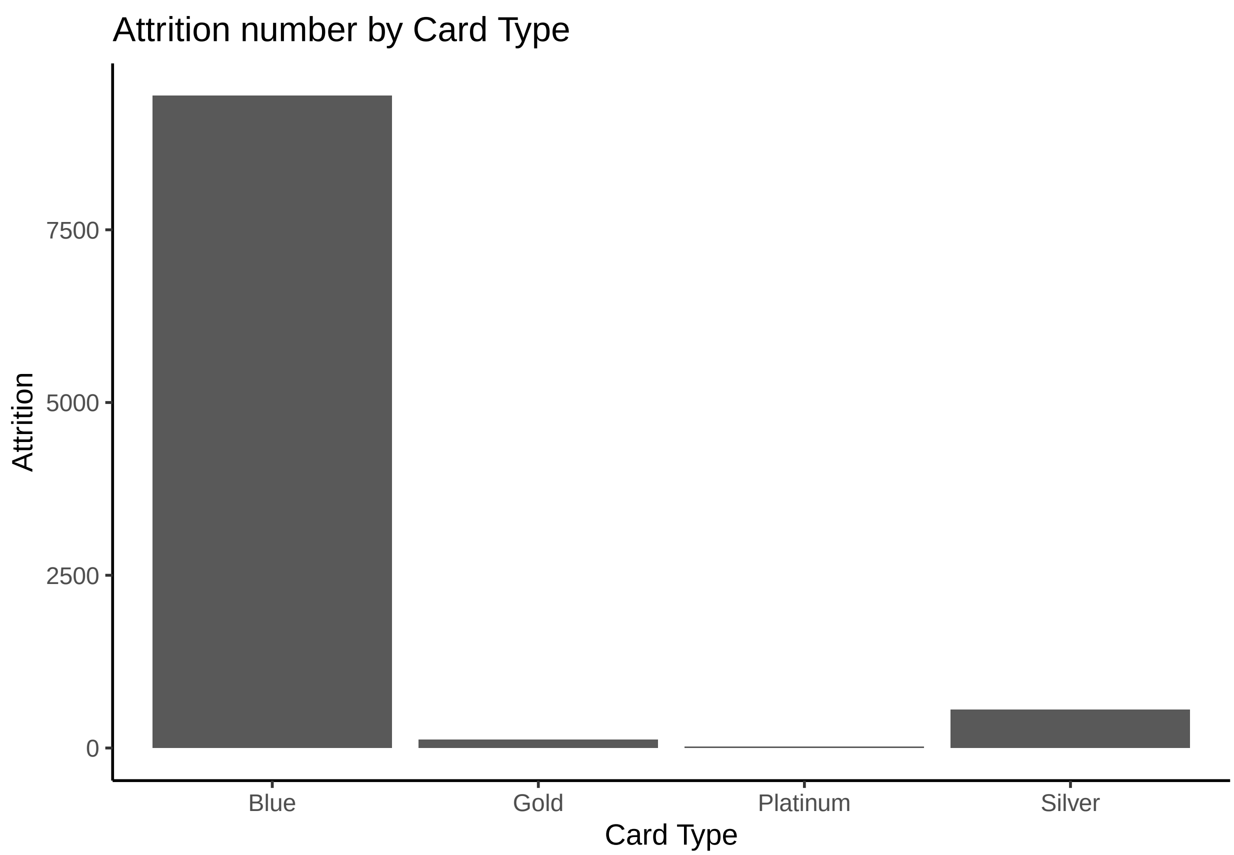

```r
ggplot(churn, aes(x=Education_Level)) +
  geom_bar(stat="count") +
  labs(title= "Attrition number by Education", x= "Education", y="Attrition") +
  theme_classic() + scale_color_brewer(palette="Set2")
```




```
ggplot(churn, aes(x=Income_Category)) +
  geom_bar(stat="count") +
  labs(title= "Attrition number by Income", x= "Income", y="Attrition") +
  theme_classic() + scale_color_brewer(palette="Set2")
```




```
ggplot(churn, aes(x=Card_Category)) +
  geom_bar(stat="count") +
  labs(title= "Attrition number by Card Type", x= "Card Type", y="Attrition") +
  theme_classic() + scale_color_brewer(palette="Set2")
```




```
table(churn$Attrition_Flag, churn$Customer_Age)
```

```
##
##                      26  27  28  29  30  31  32  33  34  35  36  37  38  39  40
##   Attrited Customer   6   3   1   7  15  13  17  20  19  21  24  37  47  48  64
##   Existing Customer  72  29  28  49  55  78  89 107 127 163 197 223 256 285 297
##
##                      41  42  43  44  45  46  47  48  49  50  51  52  53  54  55
##   Attrited Customer  76  62  85  84  79  82  76  85  79  71  58  58  59  69  51
##   Existing Customer 303 364 388 416 407 408 403 387 416 381 340 318 328 238 228
##
##                      56  57  58  59  60  61  62  63  64  65  66  67  68  70  73
##   Attrited Customer  43  33  24  40  13  17  17   8   5   9   1   0   1   0   0
##   Existing Customer 219 190 133 117 114  76  76  57  38  92   1   4   1   1   1
```

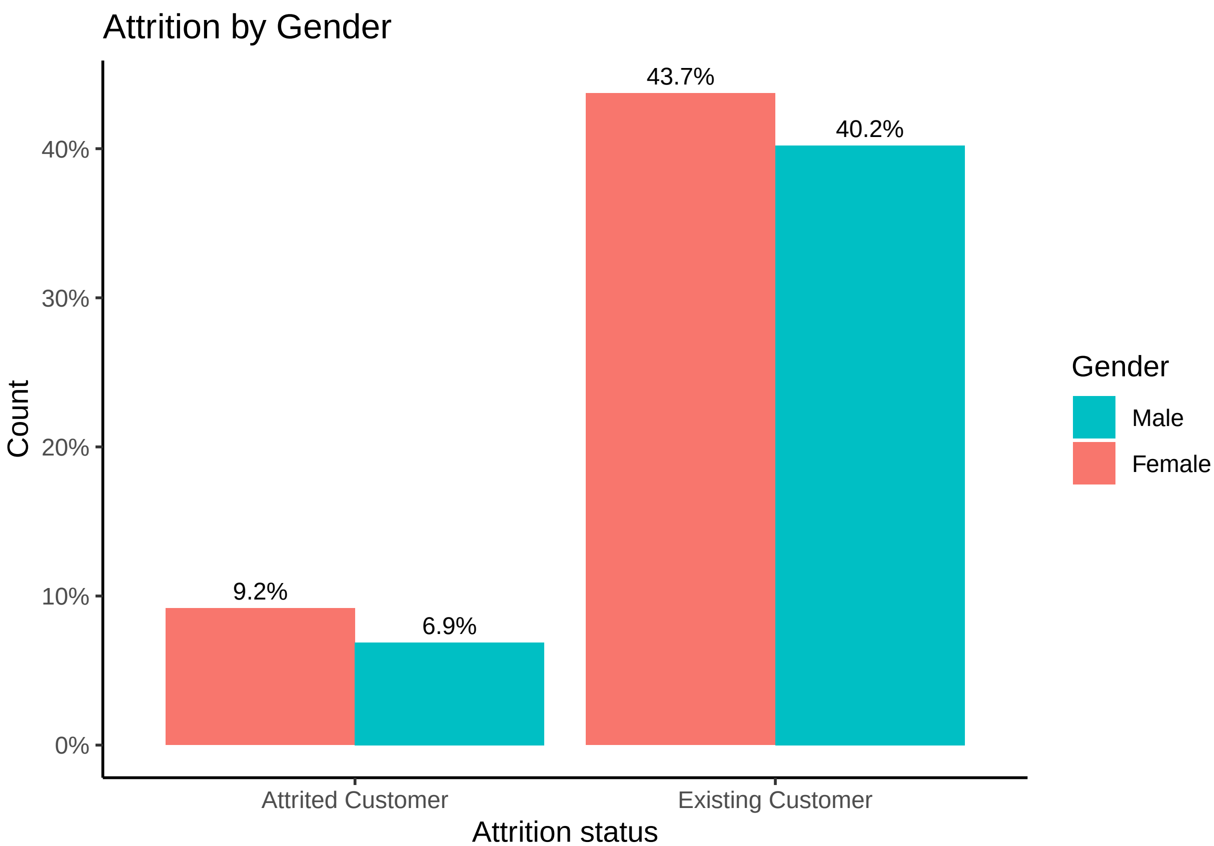
```
table(churn$Attrition_Flag, churn$Gender)
```

```
##
##                        F    M
##   Attrited Customer  930  697
##   Existing Customer 4428 4072
```

```
ggplot(churn, aes(x=Attrition_Flag,
                  y= prop.table(stat(count)),
                  fill= factor(Gender),
                  label= scales::percent(prop.table(stat(count))))) +
  geom_bar(position = position_dodge())+
  geom_text(stat="count",
```

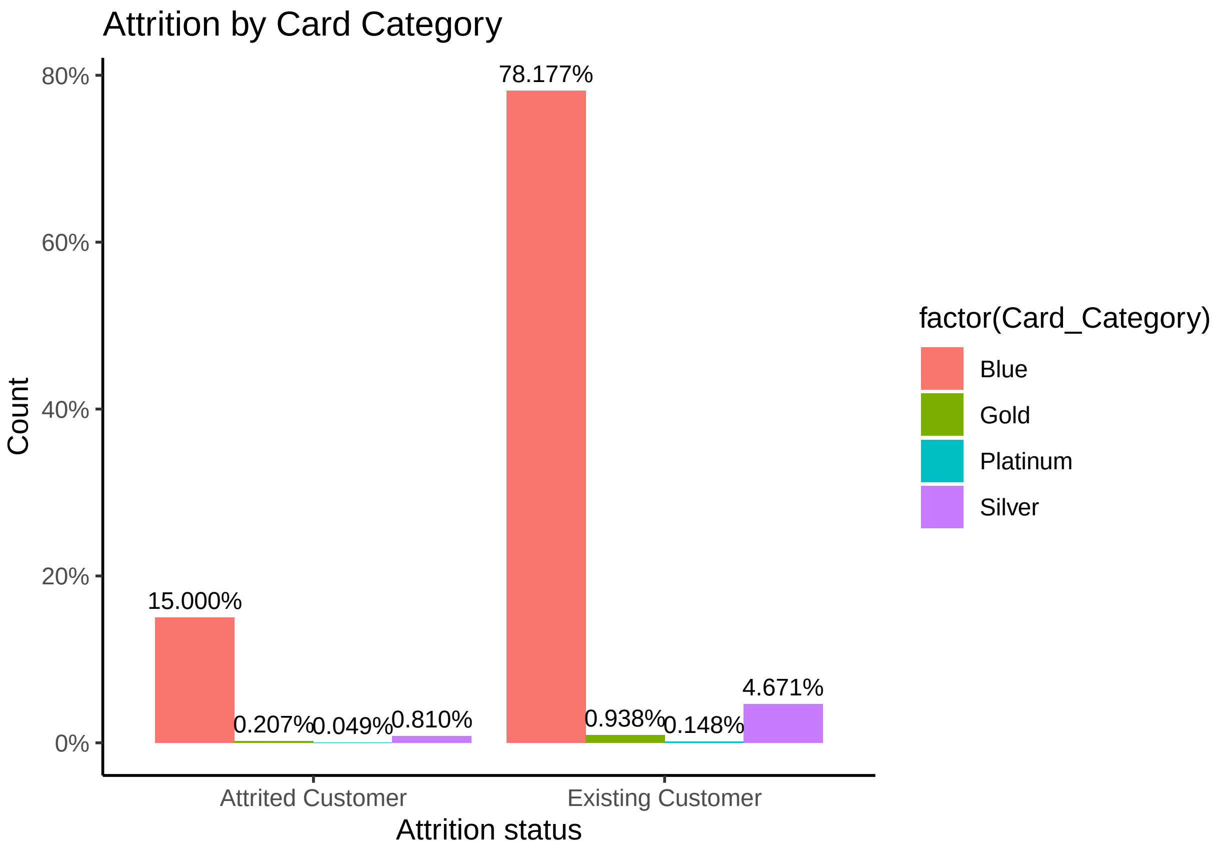
```
            position = position_dodge(.9),
            vjust= -0.5, size=3)+
  scale_y_continuous(labels = scales::percent)+
  labs(title = "Attrition by Gender",
       x= "Attrition status",
       y="Count")+
  theme_classic()+
  scale_fill_discrete(
    name="Gender",
    breaks=c("M", "F"),
    labels=c("Male", "Female" )
  )
```

```
## Warning: `stat(count)` was deprecated in ggplot2 3.4.0.
## i Please use `after_stat(count)` instead.
```

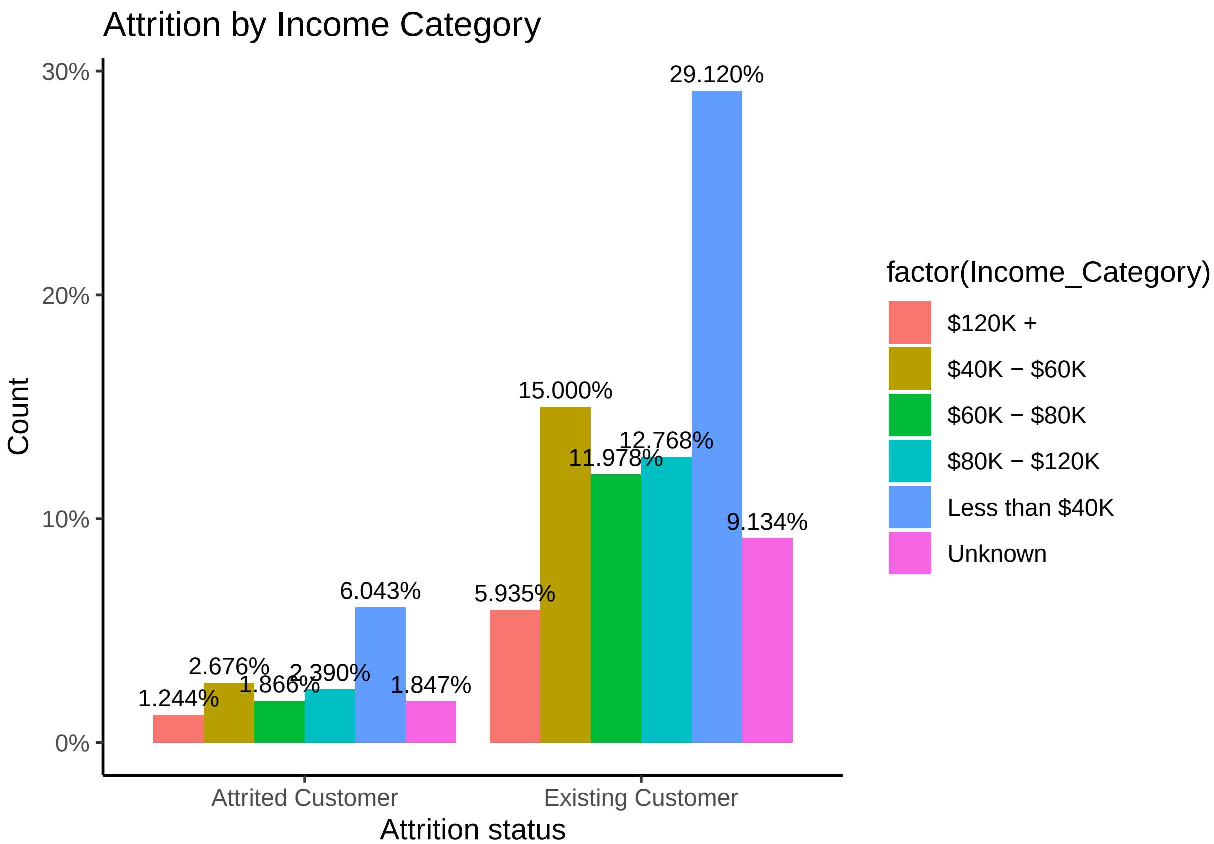



```
ggplot(churn, aes(x=Attrition_Flag,
                  y= prop.table(stat(count)),
                  fill= factor(Card_Category),
                  label= scales::percent(prop.table(stat(count))))) +
  geom_bar(position = position_dodge())+
  geom_text(stat="count",
            position = position_dodge(.9),
            vjust= -0.5, size=3)+
  scale_y_continuous(labels = scales::percent)+
  labs(title = "Attrition by Card Category",
       x= "Attrition status",
```

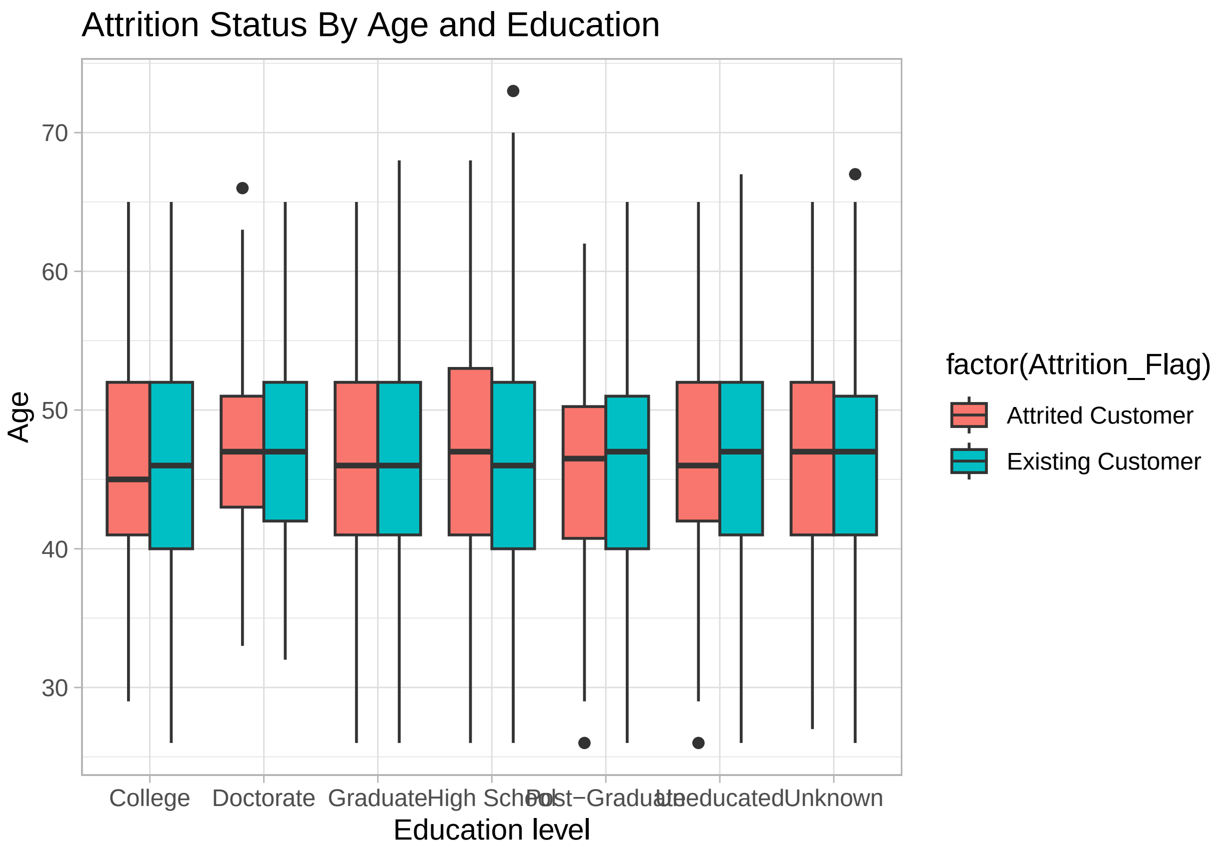
```
      y="Count")+
  theme_classic()
```



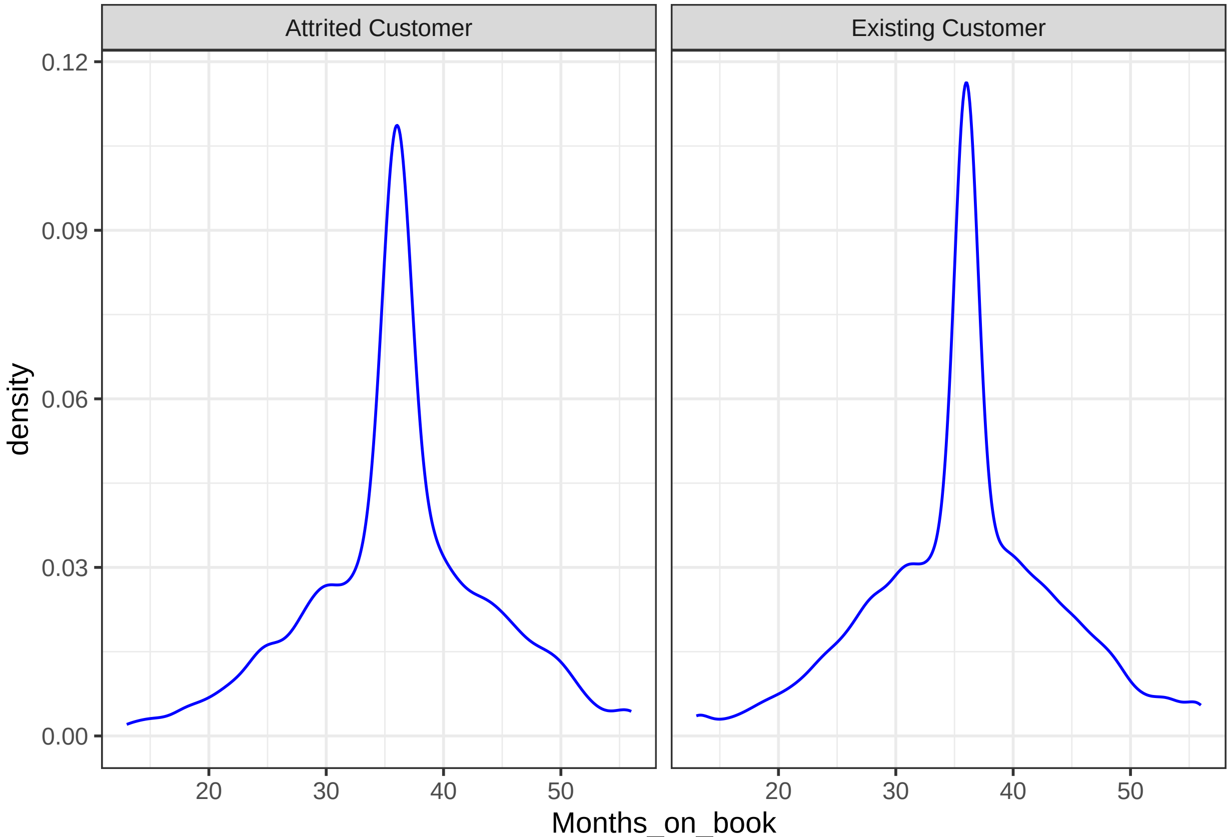

```
ggplot(churn, aes(x=Attrition_Flag,
                  y= prop.table(stat(count)),
                  fill= factor(Income_Category),
                  label= scales::percent(prop.table(stat(count))))) +
  geom_bar(position = position_dodge())+
  geom_text(stat="count",
            position = position_dodge(.9),
            vjust= -0.5, size=3)+
  scale_y_continuous(labels = scales::percent)+
  labs(title = "Attrition by Income Category",
      x= "Attrition status",
      y="Count")+
  theme_classic()
```



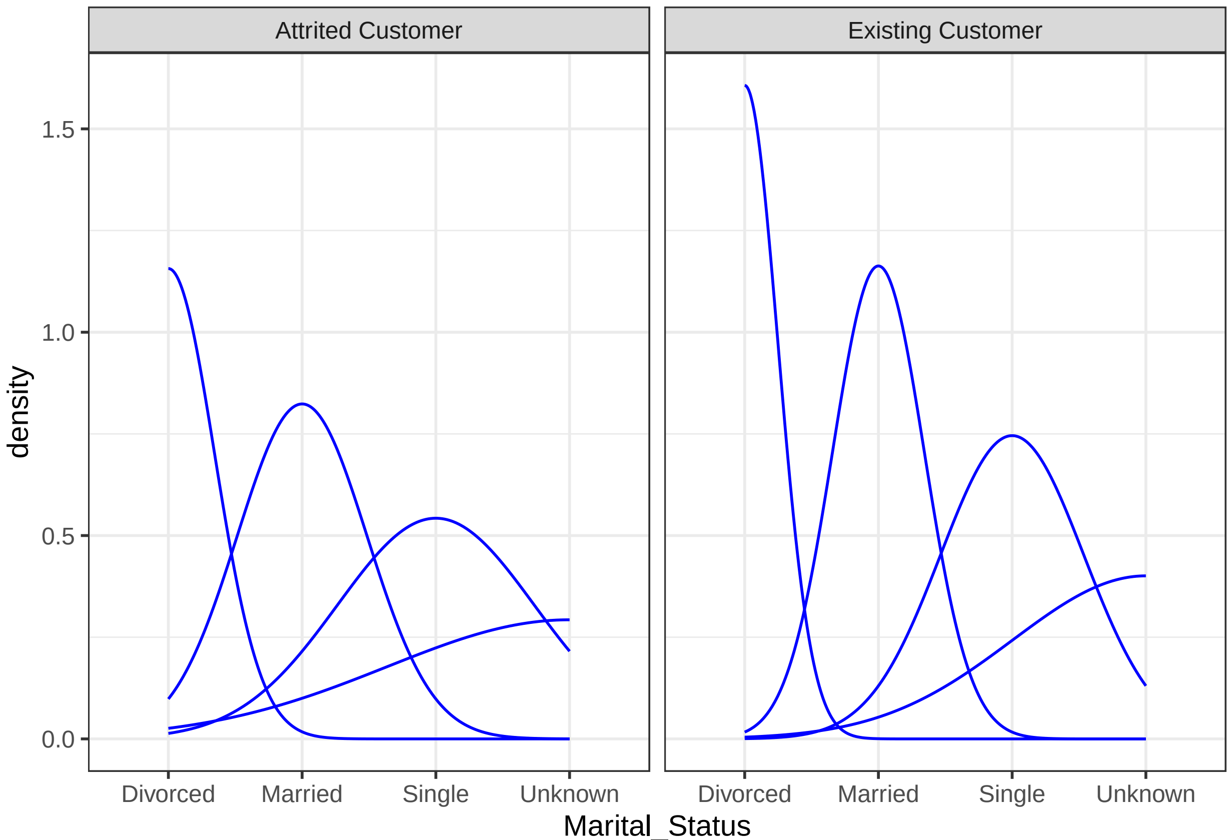

```
ggplot(churn, aes(y=Customer_Age,
                  x= Education_Level,
                  fill= factor(Attrition_Flag))) +
  geom_boxplot(position = position_dodge())+
  labs(title = "Attrition Status By Age and Education",
       x= "Education level",
       y="Age")+ theme_light()
```



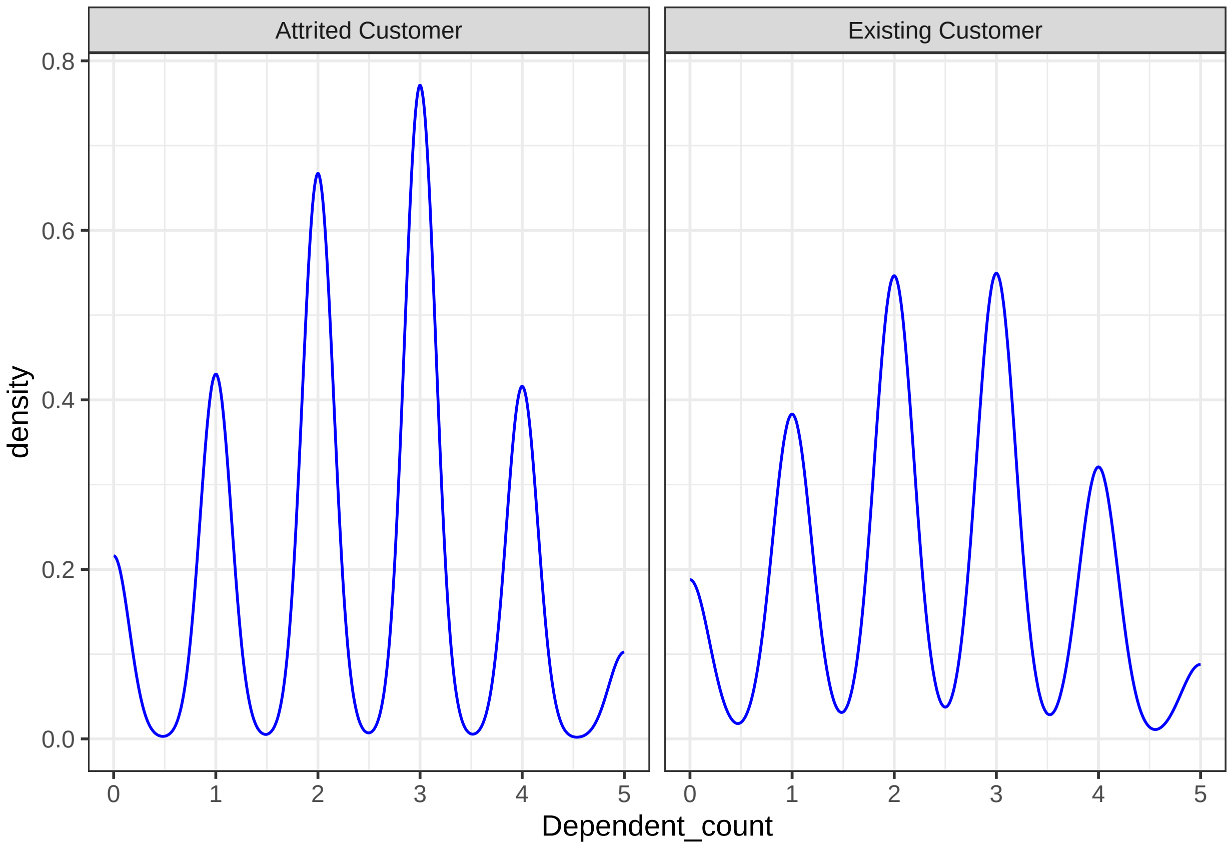

```
ggplot(churn, aes(Months_on_book))+
  geom_density(col="blue")+ facet_wrap(~Attrition_Flag)+theme_bw()
```




```
ggplot(churn, aes(Marital_Status))+
  geom_density(col="blue")+ facet_wrap(~Attrition_Flag)+theme_bw()
```




```
ggplot(churn, aes(Dependent_count))+
  geom_density(col="blue")+ facet_wrap(~Attrition_Flag)+theme_bw()
```




```r
#PCA starts here
#PCA
churn.pca <- prcomp(scale(churn[,c(2,4,9:20)]), center = TRUE)
summary(churn.pca)
```

```
## Importance of components:
##                           PC1    PC2    PC3    PC4     PC5    PC6     PC7
## Standard deviation     1.6025 1.4301 1.3408 1.2024 1.11491 1.0019 0.99250
## Proportion of Variance 0.1834 0.1461 0.1284 0.1033 0.08879 0.0717 0.07036
## Cumulative Proportion  0.1834 0.3295 0.4579 0.5612 0.64998 0.7217 0.79203
##                            PC8     PC9    PC10    PC11    PC12    PC13
## Standard deviation     0.95112 0.89829 0.77448 0.47086 0.45909 0.40948
## Proportion of Variance 0.06462 0.05764 0.04284 0.01584 0.01505 0.01198
## Cumulative Proportion  0.85665 0.91429 0.95713 0.97297 0.98802 1.00000
##                             PC14
## Standard deviation     1.067e-15
## Proportion of Variance 0.000e+00
## Cumulative Proportion  1.000e+00
```

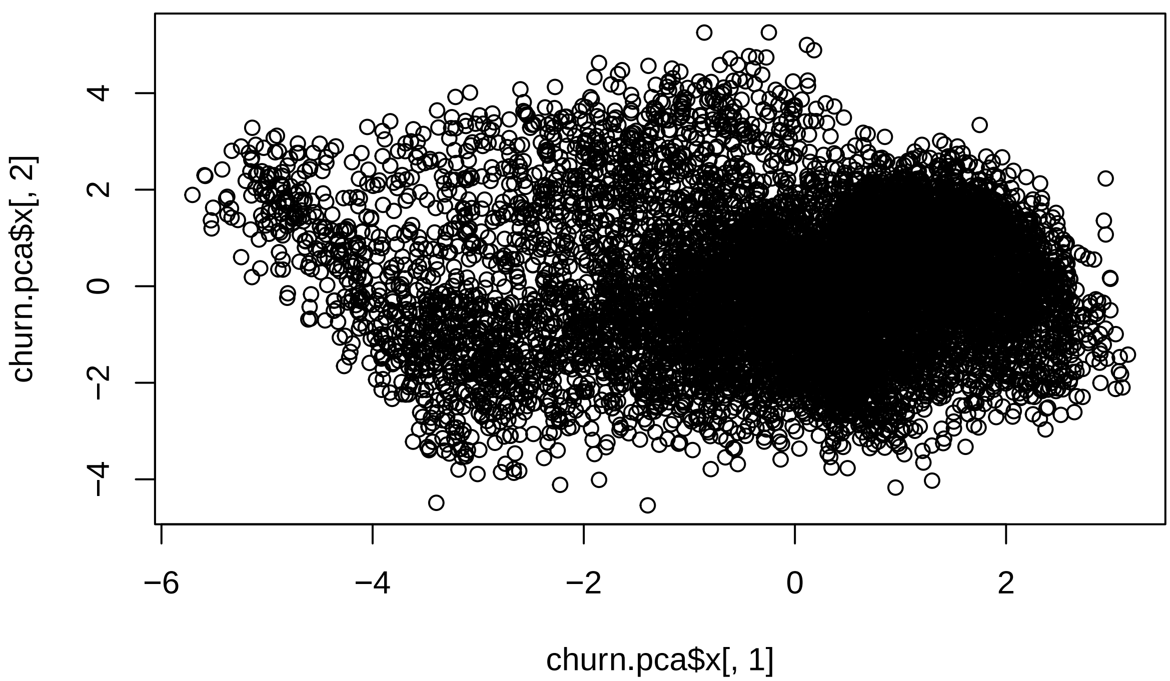
```r
pc_data <- churn.pca$x[,1:10]
cat_data <- churn[,c(1,3,5:8)]
churn_pca <-data.frame(cat_data, pc_data)
```

```r
churn_pca[sapply(churn_pca, is.character)]<- lapply(churn_pca[sapply(churn_pca, is.character)], as.facto
summary(churn_pca)
```

```
##            Attrition_Flag Gender       Education_Level  Marital_Status
##  Attrited Customer:1627   F:5358   College      :1013   Divorced: 748
```

```
##  Existing Customer:8500   M:4769   Doctorate     : 451   Married :4687
##                                    Graduate      :3128   Single  :3943
##                                    High School   :2013   Unknown : 749
##                                    Post-Graduate: 516
##                                    Uneducated    :1487
##                                    Unknown       :1519
##        Income_Category  Card_Category       PC1                PC2
##  $120K +        : 727   Blue     :9436   Min.   :-5.7066   Min.   :-4.53940
##  $40K - $60K    :1790   Gold     : 116   1st Qu.:-0.7787   1st Qu.:-1.00929
##  $60K - $80K    :1402   Platinum:  20    Median : 0.2715   Median :-0.03217
##  $80K - $120K   :1535   Silver   : 555   Mean   : 0.0000   Mean   : 0.00000
##  Less than $40K:3561                     3rd Qu.: 1.1743   3rd Qu.: 0.94732
##  Unknown        :1112                    Max.   : 3.1542   Max.   : 5.25687
##
##       PC3                PC4                 PC5                PC6
##  Min.   :-4.35199   Min.   :-12.52897   Min.   :-4.41611   Min.   :-3.56203
##  1st Qu.:-0.91203   1st Qu.: -0.61187   1st Qu.:-0.77707   1st Qu.:-0.68159
##  Median :-0.01552   Median :  0.08801   Median :-0.04466   Median :-0.01311
##  Mean   : 0.00000   Mean   :  0.00000   Mean   : 0.00000   Mean   : 0.00000
##  3rd Qu.: 0.87063   3rd Qu.:  0.78146   3rd Qu.: 0.73675   3rd Qu.: 0.68615
##  Max.   : 4.57837   Max.   :  4.26339   Max.   :10.45639   Max.   : 3.61410
##
##       PC7                 PC8                 PC9                PC10
##  Min.   :-4.838357   Min.   :-3.57531   Min.   :-3.75304   Min.   :-6.661278
##  1st Qu.:-0.642200   1st Qu.:-0.64302   1st Qu.:-0.64028   1st Qu.:-0.443576
##  Median :-0.002061   Median :-0.01053   Median :-0.01348   Median :-0.008394
##  Mean   : 0.000000   Mean   : 0.00000   Mean   : 0.00000   Mean   : 0.000000
##  3rd Qu.: 0.713608   3rd Qu.: 0.63601   3rd Qu.: 0.67782   3rd Qu.: 0.434027
##  Max.   : 3.058806   Max.   : 3.75028   Max.   : 3.01369   Max.   : 7.829975
##
```

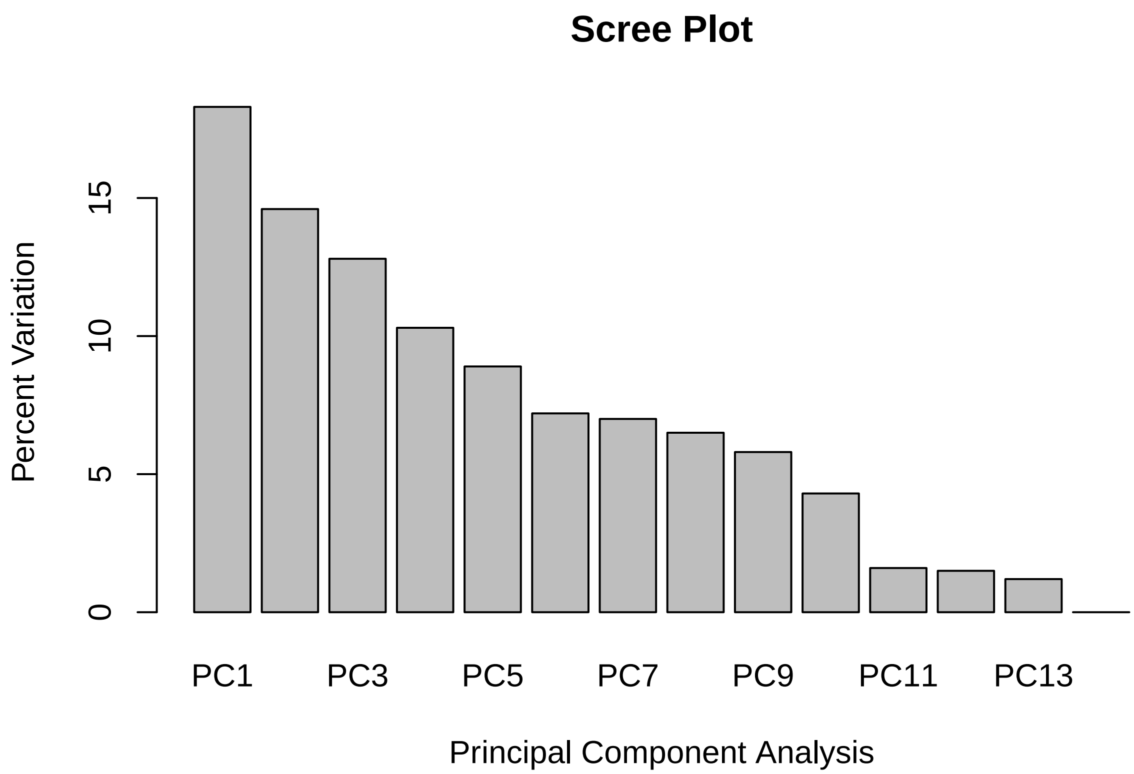
```r
#Plotting PCA
plot(churn.pca$x[,1],churn.pca$x[,2])
```



```r
#How much variation in the original data does PCA account for
churn.pca.var <- churn.pca$sdev^2
churn.pca.var.per <- round(churn.pca.var/sum(churn.pca.var)*100,1)
churn.pca.var.per
```

```
##  [1] 18.3 14.6 12.8 10.3  8.9  7.2  7.0  6.5  5.8  4.3  1.6  1.5  1.2  0.0
```

```r
#Plotting PCA percentages
barplot(churn.pca.var.per, main="Scree Plot", xlab="Principal Component Analysis", names = c("PC1", "PC2
```




```r
#PCA ends here
```

```r
#Converting all features to categorical data
churn[sapply(churn, is.character)]<- lapply(churn[sapply(churn, is.character)], as.factor)
```

```r
str(churn)
```

```
## 'data.frame':    10127 obs. of  20 variables:
##  $ Attrition_Flag          : Factor w/ 2 levels "Attrited Customer",..: 2 2 2 2 2 2 2 2 2 2 ...
##  $ Customer_Age            : int  45 49 51 40 40 44 51 32 37 48 ...
##  $ Gender                  : Factor w/ 2 levels "F","M": 2 1 2 1 2 2 2 2 2 2 ...
##  $ Dependent_count         : int  3 5 3 4 3 2 4 0 3 2 ...
##  $ Education_Level         : Factor w/ 7 levels "College","Doctorate",..: 4 3 3 4 6 3 7 4 6 3 ...
##  $ Marital_Status          : Factor w/ 4 levels "Divorced","Married",..: 2 3 2 4 2 2 2 4 3 3 ...
##  $ Income_Category         : Factor w/ 6 levels "$120K +","$40K - $60K",..: 3 5 4 5 3 2 1 3 3 4 ...
##  $ Card_Category           : Factor w/ 4 levels "Blue","Gold",..: 1 1 1 1 1 1 2 4 1 1 ...
##  $ Months_on_book          : int  39 44 36 34 21 36 46 27 36 36 ...
##  $ Total_Relationship_Count: int  5 6 4 3 5 3 6 2 5 6 ...
##  $ Months_Inactive_12_mon  : int  1 1 1 4 1 1 1 2 2 3 ...
##  $ Contacts_Count_12_mon   : int  3 2 0 1 0 2 3 2 0 3 ...
##  $ Credit_Limit            : num  12691 8256 3418 3313 4716 ...
```

```
## $ Total_Revolving_Bal     : int  777 864 0 2517 0 1247 2264 1396 2517 1677 ...
## $ Avg_Open_To_Buy         : num  11914 7392 3418 796 4716 ...
## $ Total_Amt_Chng_Q4_Q1    : num  1.33 1.54 2.59 1.41 2.17 ...
## $ Total_Trans_Amt         : int  1144 1291 1887 1171 816 1088 1330 1538 1350 1441 ...
## $ Total_Trans_Ct          : int  42 33 20 20 28 24 31 36 24 32 ...
## $ Total_Ct_Chng_Q4_Q1     : num  1.62 3.71 2.33 2.33 2.5 ...
## $ Avg_Utilization_Ratio   : num  0.061 0.105 0 0.76 0 0.311 0.066 0.048 0.113 0.144 ...
```

```
summary(churn)
```

```
##            Attrition_Flag  Customer_Age   Gender   Dependent_count
##  Attrited Customer:1627   Min.   :26.00   F:5358   Min.   :0.000  
##  Existing Customer:8500   1st Qu.:41.00   M:4769   1st Qu.:1.000  
##                           Median :46.00            Median :2.000  
##                           Mean   :46.33            Mean   :2.346  
##                           3rd Qu.:52.00            3rd Qu.:3.000  
##                           Max.   :73.00            Max.   :5.000  
##                                                                   
##       Education_Level  Marital_Status       Income_Category  Card_Category 
##  College      :1013   Divorced: 748   $120K +       : 727   Blue    :9436  
##  Doctorate    : 451   Married :4687   $40K - $60K   :1790   Gold    : 116  
##  Graduate     :3128   Single  :3943   $60K - $80K   :1402   Platinum:  20  
##  High School  :2013   Unknown : 749   $80K - $120K  :1535   Silver  : 555  
##  Post-Graduate: 516                   Less than $40K:3561                  
##  Uneducated   :1487                   Unknown       :1112                  
##  Unknown      :1519                                                        
##  Months_on_book  Total_Relationship_Count Months_Inactive_12_mon
##  Min.   :13.00   Min.   :1.000            Min.   :0.000         
##  1st Qu.:31.00   1st Qu.:3.000            1st Qu.:2.000         
##  Median :36.00   Median :4.000            Median :2.000         
##  Mean   :35.93   Mean   :3.813            Mean   :2.341         
##  3rd Qu.:40.00   3rd Qu.:5.000            3rd Qu.:3.000         
##  Max.   :56.00   Max.   :6.000            Max.   :6.000         
##                                                                 
##  Contacts_Count_12_mon  Credit_Limit   Total_Revolving_Bal Avg_Open_To_Buy
##  Min.   :0.000         Min.   : 1438   Min.   :   0        Min.   :    3  
##  1st Qu.:2.000         1st Qu.: 2555   1st Qu.: 359        1st Qu.: 1324  
##  Median :2.000         Median : 4549   Median :1276        Median : 3474  
##  Mean   :2.455         Mean   : 8632   Mean   :1163        Mean   : 7469  
##  3rd Qu.:3.000         3rd Qu.:11068   3rd Qu.:1784        3rd Qu.: 9859  
##  Max.   :6.000         Max.   :34516   Max.   :2517        Max.   :34516  
##                                                                           
##  Total_Amt_Chng_Q4_Q1 Total_Trans_Amt Total_Trans_Ct   Total_Ct_Chng_Q4_Q1
##  Min.   :0.0000       Min.   :  510   Min.   : 10.00   Min.   :0.0000     
##  1st Qu.:0.6310       1st Qu.: 2156   1st Qu.: 45.00   1st Qu.:0.5820     
##  Median :0.7360       Median : 3899   Median : 67.00   Median :0.7020     
##  Mean   :0.7599       Mean   : 4404   Mean   : 64.86   Mean   :0.7122     
##  3rd Qu.:0.8590       3rd Qu.: 4741   3rd Qu.: 81.00   3rd Qu.:0.8180     
##  Max.   :3.3970       Max.   :18484   Max.   :139.00   Max.   :3.7140     
##                                                                           
##  Avg_Utilization_Ratio
##  Min.   :0.0000       
##  1st Qu.:0.0230       
##  Median :0.1760       
##  Mean   :0.2749       
```

```
##   3rd Qu.:0.5030
##   Max.   :0.9990
##
```

```r
#Splitting the pca dataset
intrain_pca<- createDataPartition(churn_pca$Attrition_Flag, p=0.80, list = FALSE)
training_pca<- churn_pca[intrain_pca,]
testing_pca<- churn_pca[-intrain_pca,]
dim(training_pca); dim(testing_pca)
```

```
## [1] 8102   16
```

```
## [1] 2025   16
```

```r
#summary(training_pca)
#summary(testing_pca)
```

```r
#Splitting the regular dataset
intrain_reg<- createDataPartition(churn$Attrition_Flag, p=0.80, list = FALSE)
training_reg<- churn[intrain_reg,]
testing_reg<- churn[-intrain_reg,]
dim(training_reg); dim(testing_reg)
```

```
## [1] 8102   20
```

```
## [1] 2025   20
```

```r
#summary(training_reg)
#summary(testing_reg)
```

```r
#Randomforest for PCA data
random_forest <- randomForest(Attrition_Flag ~ ., ntree= 500, family="binomial", data=training_pca)
print(summary(random_forest))
```

```
##                   Length Class  Mode
## call                  5  -none- call
## type                  1  -none- character
## predicted          8102  factor numeric
## err.rate           1500  -none- numeric
## confusion             6  -none- numeric
## votes             16204  matrix numeric
## oob.times          8102  -none- numeric
## classes               2  -none- character
## importance           15  -none- numeric
## importanceSD          0  -none- NULL
## localImportance       0  -none- NULL
## proximity             0  -none- NULL
## ntree                 1  -none- numeric
## mtry                  1  -none- numeric
## forest               14  -none- list
## y                  8102  factor numeric
## test                  0  -none- NULL
## inbag                 0  -none- NULL
## terms                 3  terms  call
```

```r
random_forest
```

```
##
## Call:
```

```
##  randomForest(formula = Attrition_Flag ~ ., data = training_pca,      ntree = 500, family = "binomial
##                Type of random forest: classification
##                      Number of trees: 500
## No. of variables tried at each split: 3
##
##         OOB estimate of  error rate: 9.06%
## Confusion matrix:
##                    Attrited Customer Existing Customer class.error
## Attrited Customer               676               626  0.48079877
## Existing Customer               108              6692  0.01588235
```

```
rf_pred <- predict(random_forest, testing_pca)
caret::confusionMatrix(rf_pred, testing_pca$Attrition_Flag)
```

```
## Confusion Matrix and Statistics
##
##                      Reference
## Prediction           Attrited Customer Existing Customer
##   Attrited Customer               163                21
##   Existing Customer               162              1679
##
##                Accuracy : 0.9096
##                  95% CI : (0.8963, 0.9218)
##     No Information Rate : 0.8395
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.5933
##
##  Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.50154
##             Specificity : 0.98765
##          Pos Pred Value : 0.88587
##          Neg Pred Value : 0.91200
##              Prevalence : 0.16049
##          Detection Rate : 0.08049
##    Detection Prevalence : 0.09086
##       Balanced Accuracy : 0.74459
##
##        'Positive' Class : Attrited Customer
##
```

```
#Randomforest for Regular data
random_forest <- randomForest(Attrition_Flag ~ ., ntree= 500, family="binomial", data=training_reg)
print(summary(random_forest))
```

```
##                 Length Class  Mode
## call                5  -none- call
## type                1  -none- character
## predicted        8102  factor numeric
## err.rate         1500  -none- numeric
## confusion           6  -none- numeric
## votes           16204  matrix numeric
## oob.times        8102  -none- numeric
## classes             2  -none- character
```

```
## importance         19  -none- numeric
## importanceSD        0  -none- NULL
## localImportance     0  -none- NULL
## proximity           0  -none- NULL
## ntree               1  -none- numeric
## mtry                1  -none- numeric
## forest             14  -none- list
## y                8102  factor numeric
## test                0  -none- NULL
## inbag               0  -none- NULL
## terms               3  terms  call
```

```
random_forest
```

```
##
## Call:
##  randomForest(formula = Attrition_Flag ~ ., data = training_reg,      ntree = 500, family = "binomia
##                Type of random forest: classification
##                      Number of trees: 500
## No. of variables tried at each split: 4
##
##         OOB estimate of  error rate: 3.81%
## Confusion matrix:
##                    Attrited Customer Existing Customer class.error
## Attrited Customer               1074               228  0.17511521
## Existing Customer                 81              6719  0.01191176
```

```
rf_pred <- predict(random_forest, testing_reg)
caret::confusionMatrix(rf_pred, testing_reg$Attrition_Flag)
```

```
## Confusion Matrix and Statistics
##
##                    Reference
## Prediction          Attrited Customer Existing Customer
##   Attrited Customer               269                15
##   Existing Customer                56              1685
##
##                Accuracy : 0.9649
##                  95% CI : (0.956, 0.9725)
##     No Information Rate : 0.8395
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.8629
##
##  Mcnemar's Test P-Value : 2.063e-06
##
##             Sensitivity : 0.8277
##             Specificity : 0.9912
##          Pos Pred Value : 0.9472
##          Neg Pred Value : 0.9678
##              Prevalence : 0.1605
##          Detection Rate : 0.1328
##    Detection Prevalence : 0.1402
##       Balanced Accuracy : 0.9094
##
```

```
##        'Positive' Class : Attrited Customer
##
```

```r
#Logistic Regression for PCA Data
LogModel <- glm(Attrition_Flag ~ ., family= "binomial", data = training_pca)
print(summary(LogModel))
```

```
##
## Call:
## glm(formula = Attrition_Flag ~ ., family = "binomial", data = training_pca)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.7808   0.1023   0.2388   0.4453   2.2681
##
## Coefficients:
##                                Estimate Std. Error z value Pr(>|z|)
## (Intercept)                     1.94431    0.28187   6.898 5.28e-12 ***
## GenderM                         0.70591    0.15365   4.594 4.34e-06 ***
## Education_LevelDoctorate       -0.40085    0.20973  -1.911 0.055966 .
## Education_LevelGraduate        -0.12670    0.14349  -0.883 0.377233
## Education_LevelHigh School     -0.19665    0.15301  -1.285 0.198720
## Education_LevelPost-Graduate   -0.38488    0.20870  -1.844 0.065162 .
## Education_LevelUneducated      -0.18136    0.16307  -1.112 0.266072
## Education_LevelUnknown         -0.27858    0.15903  -1.752 0.079820 .
## Marital_StatusMarried           0.31336    0.15657   2.001 0.045355 *
## Marital_StatusSingle           -0.07485    0.15699  -0.477 0.633507
## Marital_StatusUnknown          -0.07636    0.20249  -0.377 0.706103
## Income_Category$40K - $60K      0.72740    0.20972   3.468 0.000524 ***
## Income_Category$60K - $80K      0.56011    0.18621   3.008 0.002630 **
## Income_Category$80K - $120K     0.12503    0.16965   0.737 0.461147
## Income_CategoryLess than $40K   0.62619    0.22872   2.738 0.006186 **
## Income_CategoryUnknown          0.72566    0.23928   3.033 0.002424 **
## Card_CategoryGold              -1.42836    0.37390  -3.820 0.000133 ***
## Card_CategoryPlatinum          -1.51158    0.74364  -2.033 0.042085 *
## Card_CategorySilver            -0.60229    0.20557  -2.930 0.003392 **
## PC1                            -0.10942    0.03804  -2.877 0.004019 **
## PC2                             0.95479    0.03657  26.110  < 2e-16 ***
## PC3                             0.37151    0.03080  12.062  < 2e-16 ***
## PC4                            -0.74523    0.04133 -18.030  < 2e-16 ***
## PC5                            -0.04226    0.03608  -1.171 0.241494
## PC6                            -0.25903    0.04071  -6.362 1.99e-10 ***
## PC7                             0.42054    0.03924  10.718  < 2e-16 ***
## PC8                             0.39266    0.04245   9.250  < 2e-16 ***
## PC9                             1.01930    0.04857  20.984  < 2e-16 ***
## PC10                            0.35612    0.05429   6.559 5.41e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 7143.2  on 8101  degrees of freedom
## Residual deviance: 4415.0  on 8073  degrees of freedom
## AIC: 4473
##
```

```
## Number of Fisher Scoring iterations: 6
```

```
anova(LogModel, test="Chisq")
```

```
## Analysis of Deviance Table
##
## Model: binomial, link: logit
##
## Response: Attrition_Flag
##
## Terms added sequentially (first to last)
##
##
##                 Df Deviance Resid. Df Resid. Dev  Pr(>Chi)
## NULL                             8101     7143.2
## Gender           1    10.68      8100     7132.5  0.001082 **
## Education_Level  6    11.69      8094     7120.8  0.069196 .
## Marital_Status   3     5.66      8091     7115.2  0.129333
## Income_Category  5    12.70      8086     7102.5  0.026388 *
## Card_Category    3     4.09      8083     7098.4  0.251843
## PC1              1     0.00      8082     7098.4  0.956728
## PC2              1  1169.90      8081     5928.5 < 2.2e-16 ***
## PC3              1   223.66      8080     5704.8 < 2.2e-16 ***
## PC4              1   488.29      8079     5216.5 < 2.2e-16 ***
## PC5              1     0.28      8078     5216.3  0.594594
## PC6              1    46.10      8077     5170.2 1.122e-11 ***
## PC7              1    95.97      8076     5074.2 < 2.2e-16 ***
## PC8              1    83.74      8075     4990.4 < 2.2e-16 ***
## PC9              1   530.90      8074     4459.5 < 2.2e-16 ***
## PC10             1    44.51      8073     4415.0 2.534e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
log_reg <- predict(LogModel, testing_pca[-1],  type = "response")
y_pred <- ifelse(log_reg > 0.5, 2, 1)
y_pred <- as.numeric(y_pred)
target <- as.numeric(testing_pca$Attrition_Flag)
#prop.table(table(training_pca$Attrition_Flag))
caret::confusionMatrix(table(y_pred, target))
```

```
## Confusion Matrix and Statistics
##
##       target
## y_pred    1    2
##      1  155   44
##      2  170 1656
##
##                Accuracy : 0.8943
##                  95% CI : (0.8801, 0.9074)
##     No Information Rate : 0.8395
##     P-Value [Acc > NIR] : 9.099e-13
##
##                   Kappa : 0.5349
##
##  Mcnemar's Test P-Value : < 2.2e-16
```

```
##
##             Sensitivity : 0.47692
##             Specificity : 0.97412
##          Pos Pred Value : 0.77889
##          Neg Pred Value : 0.90690
##              Prevalence : 0.16049
##          Detection Rate : 0.07654
##    Detection Prevalence : 0.09827
##       Balanced Accuracy : 0.72552
##
##        'Positive' Class : 1
##
```

```
#Logistic Regression for Regular Data
LogModel <- glm(Attrition_Flag ~ ., family= "binomial", data = training_reg)
print(summary(LogModel))
```

```
##
## Call:
## glm(formula = Attrition_Flag ~ ., family = "binomial", data = training_reg)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.5234   0.0701   0.1773   0.3678   3.0838
##
## Coefficients: (1 not defined because of singularities)
##                                  Estimate Std. Error z value Pr(>|z|)
## (Intercept)                    -6.613e+00  5.254e-01 -12.587  < 2e-16 ***
## Customer_Age                    6.915e-03  8.514e-03   0.812 0.416675
## GenderM                         8.467e-01  1.598e-01   5.300 1.16e-07 ***
## Dependent_count                -1.263e-01  3.293e-02  -3.837 0.000125 ***
## Education_LevelDoctorate       -4.427e-01  2.276e-01  -1.945 0.051738 .
## Education_LevelGraduate         3.427e-03  1.548e-01   0.022 0.982336
## Education_LevelHigh School     -4.249e-02  1.648e-01  -0.258 0.796564
## Education_LevelPost-Graduate   -3.456e-01  2.257e-01  -1.531 0.125809
## Education_LevelUneducated      -8.841e-02  1.741e-01  -0.508 0.611496
## Education_LevelUnknown         -9.255e-02  1.733e-01  -0.534 0.593390
## Marital_StatusMarried           5.555e-01  1.698e-01   3.271 0.001070 **
## Marital_StatusSingle           -2.268e-02  1.703e-01  -0.133 0.894035
## Marital_StatusUnknown          -2.897e-02  2.165e-01  -0.134 0.893578
## Income_Category$40K - $60K      9.564e-01  2.247e-01   4.256 2.08e-05 ***
## Income_Category$60K - $80K      6.553e-01  1.997e-01   3.281 0.001033 **
## Income_Category$80K - $120K     3.237e-01  1.837e-01   1.762 0.078079 .
## Income_CategoryLess than $40K   7.665e-01  2.422e-01   3.165 0.001553 **
## Income_CategoryUnknown          8.277e-01  2.564e-01   3.228 0.001248 **
## Card_CategoryGold              -1.088e+00  3.999e-01  -2.722 0.006492 **
## Card_CategoryPlatinum          -8.160e-01  8.479e-01  -0.962 0.335854
## Card_CategorySilver            -5.815e-01  2.103e-01  -2.766 0.005683 **
## Months_on_book                  1.864e-03  8.530e-03   0.219 0.826981
## Total_Relationship_Count        4.382e-01  3.043e-02  14.403  < 2e-16 ***
## Months_Inactive_12_mon         -4.946e-01  4.212e-02 -11.744  < 2e-16 ***
## Contacts_Count_12_mon          -4.936e-01  4.068e-02 -12.133  < 2e-16 ***
## Credit_Limit                    2.210e-05  7.568e-06   2.921 0.003490 **
## Total_Revolving_Bal             9.005e-04  7.939e-05  11.342  < 2e-16 ***
## Avg_Open_To_Buy                        NA         NA      NA       NA
```

```
## Total_Amt_Chng_Q4_Q1           4.397e-01  2.086e-01   2.108 0.034989 *
## Total_Trans_Amt               -4.684e-04  2.497e-05 -18.757  < 2e-16 ***
## Total_Trans_Ct                 1.160e-01  4.052e-03  28.629  < 2e-16 ***
## Total_Ct_Chng_Q4_Q1            2.681e+00  2.099e-01  12.772  < 2e-16 ***
## Avg_Utilization_Ratio          2.760e-01  2.750e-01   1.004 0.315554
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 7143.2  on 8101  degrees of freedom
## Residual deviance: 3837.8  on 8070  degrees of freedom
## AIC: 3901.8
##
## Number of Fisher Scoring iterations: 6
```

```r
anova(LogModel, test="Chisq")
```

```
## Analysis of Deviance Table
##
## Model: binomial, link: logit
##
## Response: Attrition_Flag
##
## Terms added sequentially (first to last)
##
##
##                          Df Deviance Resid. Df Resid. Dev  Pr(>Chi)
## NULL                                      8101     7143.2
## Customer_Age              1     1.13      8100     7142.1 0.2867426
## Gender                    1     7.60      8099     7134.5 0.0058283 **
## Dependent_count           1     5.41      8098     7129.1 0.0200048 *
## Education_Level           6     9.21      8092     7119.9 0.1622482
## Marital_Status            3     6.46      8089     7113.4 0.0912673 .
## Income_Category           5    11.08      8084     7102.3 0.0497376 *
## Card_Category             3     0.35      8081     7102.0 0.9501039
## Months_on_book            1     0.15      8080     7101.8 0.6971173
## Total_Relationship_Count  1   171.49      8079     6930.3 < 2.2e-16 ***
## Months_Inactive_12_mon    1   191.36      8078     6739.0 < 2.2e-16 ***
## Contacts_Count_12_mon     1   381.14      8077     6357.8 < 2.2e-16 ***
## Credit_Limit              1    13.50      8076     6344.3 0.0002384 ***
## Total_Revolving_Bal       1   502.80      8075     5841.5 < 2.2e-16 ***
## Avg_Open_To_Buy           0     0.00      8075     5841.5
## Total_Amt_Chng_Q4_Q1      1    97.73      8074     5743.8 < 2.2e-16 ***
## Total_Trans_Amt           1   329.28      8073     5414.5 < 2.2e-16 ***
## Total_Trans_Ct            1  1372.67      8072     4041.8 < 2.2e-16 ***
## Total_Ct_Chng_Q4_Q1       1   203.02      8071     3838.8 < 2.2e-16 ***
## Avg_Utilization_Ratio     1     1.01      8070     3837.8 0.3148561
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
log_reg <- predict(LogModel, testing_reg[-1],  type = "response")
```

```
## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type == :
## prediction from a rank-deficient fit may be misleading
```

```r
y_pred <- ifelse(log_reg > 0.5, 2, 1)
y_pred <- as.numeric(y_pred)
target <- as.numeric(testing_pca$Attrition_Flag)
#prop.table(table(training_pca$Attrition_Flag))
caret::confusionMatrix(table(y_pred, target))
```

```
## Confusion Matrix and Statistics
##
##       target
## y_pred    1    2
##      1   50  186
##      2  275 1514
##
##                Accuracy : 0.7723
##                  95% CI : (0.7534, 0.7905)
##     No Information Rate : 0.8395
##     P-Value [Acc > NIR] : 1
##
##                   Kappa : 0.05
##
##  Mcnemar's Test P-Value : 4.157e-05
##
##             Sensitivity : 0.15385
##             Specificity : 0.89059
##          Pos Pred Value : 0.21186
##          Neg Pred Value : 0.84628
##              Prevalence : 0.16049
##          Detection Rate : 0.02469
##    Detection Prevalence : 0.11654
##       Balanced Accuracy : 0.52222
##
##        'Positive' Class : 1
##
```

```r
#SVM for PCA Data
svmfit = svm(Attrition_Flag ~ ., data = training_pca, cross = 10, gamma = 0.5, cost = 1)
svm_pred <- predict(svmfit, testing_pca)
summary(svmfit)
```

```
##
## Call:
## svm(formula = Attrition_Flag ~ ., data = training_pca, cross = 10,
##     gamma = 0.5, cost = 1)
##
##
## Parameters:
##    SVM-Type:  C-classification
##  SVM-Kernel:  radial
##        cost:  1
##
## Number of Support Vectors:  6558
##
##  ( 5298 1260 )
##
```

```
##
## Number of Classes:  2
##
## Levels:
##  Attrited Customer Existing Customer
##
## 10-fold cross-validation on training data:
##
## Total Accuracy: 87.27475
## Single Accuracies:
##  87.28395 86.17284 86.41975 88.39506 87.42293 88.51852 87.40741 86.17284 86.91358 88.03946
```

```
caret::confusionMatrix(svm_pred, testing_pca$Attrition_Flag)
```

```
## Confusion Matrix and Statistics
##
##                    Reference
## Prediction          Attrited Customer Existing Customer
##   Attrited Customer                76                 8
##   Existing Customer               249              1692
##
##                Accuracy : 0.8731
##                  95% CI : (0.8578, 0.8873)
##     No Information Rate : 0.8395
##     P-Value [Acc > NIR] : 1.3e-05
##
##                   Kappa : 0.3273
##
##  Mcnemar's Test P-Value : < 2e-16
##
##             Sensitivity : 0.23385
##             Specificity : 0.99529
##          Pos Pred Value : 0.90476
##          Neg Pred Value : 0.87172
##              Prevalence : 0.16049
##          Detection Rate : 0.03753
##    Detection Prevalence : 0.04148
##       Balanced Accuracy : 0.61457
##
##        'Positive' Class : Attrited Customer
##
```

```
#SVM for Regular Data
svmfit = svm(Attrition_Flag ~ ., data = training_reg, cross = 10, gamma = 0.5, cost = 1)
svm_pred <- predict(svmfit, testing_reg)
summary(svmfit)
```

```
##
## Call:
## svm(formula = Attrition_Flag ~ ., data = training_reg, cross = 10,
##     gamma = 0.5, cost = 1)
##
##
## Parameters:
##    SVM-Type:  C-classification
```

```
##  SVM-Kernel:  radial
##        cost:  1
##
## Number of Support Vectors:  7325
##
##  ( 6058 1267 )
##
##
## Number of Classes:  2
##
## Levels:
##  Attrited Customer Existing Customer
##
## 10-fold cross-validation on training data:
##
## Total Accuracy: 86.49716
## Single Accuracies:
##  86.41975 86.66667 86.17284 87.28395 87.42293 86.2963 86.54321 85.06173 86.91358 86.18989
```

```
caret::confusionMatrix(svm_pred, testing_reg$Attrition_Flag)
```

```
## Confusion Matrix and Statistics
##
##                    Reference
## Prediction          Attrited Customer Existing Customer
##   Attrited Customer                60                 1
##   Existing Customer               265              1699
##
##                Accuracy : 0.8686
##                  95% CI : (0.8531, 0.8831)
##     No Information Rate : 0.8395
##     P-Value [Acc > NIR] : 0.0001427
##
##                   Kappa : 0.2741
##
##  Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.18462
##             Specificity : 0.99941
##          Pos Pred Value : 0.98361
##          Neg Pred Value : 0.86507
##              Prevalence : 0.16049
##          Detection Rate : 0.02963
##    Detection Prevalence : 0.03012
##       Balanced Accuracy : 0.59201
##
##        'Positive' Class : Attrited Customer
##
```

```
#Naive Bayes for PCA Data
naive_bayes<- naiveBayes(Attrition_Flag ~ ., data= training_pca)
naive_bayes
```

```
##
## Naive Bayes Classifier for Discrete Predictors
```

```
##
## Call:
## naiveBayes.default(x = X, y = Y, laplace = laplace)
##
## A-priori probabilities:
## Y
## Attrited Customer Existing Customer
##         0.1607011         0.8392989
##
## Conditional probabilities:
##                     Gender
## Y                            F         M
##   Attrited Customer 0.5691244 0.4308756
##   Existing Customer 0.5198529 0.4801471
##
##                     Education_Level
## Y                        College  Doctorate   Graduate High School Post-Graduate
##   Attrited Customer 0.09677419 0.06144393 0.29416283  0.18894009    0.05913978
##   Existing Customer 0.10176471 0.04338235 0.30897059  0.20044118    0.05132353
##                     Education_Level
## Y                     Uneducated    Unknown
##   Attrited Customer 0.13748080 0.16205837
##   Existing Customer 0.14588235 0.14823529
##
##                     Marital_Status
## Y                       Divorced    Married     Single    Unknown
##   Attrited Customer 0.07526882 0.43394777 0.41321045 0.07757296
##   Existing Customer 0.07264706 0.47029412 0.38279412 0.07426471
##
##                     Income_Category
## Y                        $120K + $40K - $60K $60K - $80K $80K - $120K
##   Attrited Customer 0.07910906  0.16820276  0.10983103   0.15668203
##   Existing Customer 0.07132353  0.18014706  0.14264706   0.15485294
##                     Income_Category
## Y                     Less than $40K    Unknown
##   Attrited Customer     0.36251920 0.12365591
##   Existing Customer     0.34382353 0.10720588
##
##                     Card_Category
## Y                            Blue        Gold    Platinum      Silver
##   Attrited Customer 0.936251920 0.013056836 0.003840246 0.046850998
##   Existing Customer 0.931029412 0.010735294 0.001617647 0.056617647
##
##                     PC1
## Y                           [,1]      [,2]
##   Attrited Customer 0.007850414 1.483513
##   Existing Customer 0.003357659 1.623428
##
##                     PC2
## Y                          [,1]      [,2]
##   Attrited Customer -1.1402731 1.174601
##   Existing Customer  0.1925985 1.372454
##
##                     PC3
```

```
## Y                          [,1]     [,2]
##   Attrited Customer -0.32309087 1.260835
##   Existing Customer  0.06536612 1.345442
##
##                    PC4
## Y                         [,1]     [,2]
##   Attrited Customer  0.7621174 1.100411
##   Existing Customer -0.1539898 1.171572
##
##                    PC5
## Y                          [,1]     [,2]
##   Attrited Customer  0.02871188 1.248458
##   Existing Customer -0.00502053 1.092006
##
##                    PC6
## Y                          [,1]      [,2]
##   Attrited Customer  0.18111475 0.9732688
##   Existing Customer -0.03158667 1.0030992
##
##                    PC7
## Y                         [,1]      [,2]
##   Attrited Customer -0.2420038 0.8943989
##   Existing Customer  0.0436209 1.0125524
##
##                    PC8
## Y                          [,1]      [,2]
##   Attrited Customer -0.22558638 0.9336708
##   Existing Customer  0.04795456 0.9522845
##
##                    PC9
## Y                         [,1]      [,2]
##   Attrited Customer -0.4947750 0.9496933
##   Existing Customer  0.1009862 0.8558014
##
##                    PC10
## Y                          [,1]     [,2]
##   Attrited Customer -0.11816953 0.774952
##   Existing Customer  0.02369101 0.771173
```

```
nb_pred<- predict(naive_bayes, testing_pca)
caret::confusionMatrix(nb_pred, testing_pca$Attrition_Flag)
```

```
## Confusion Matrix and Statistics
##
##                    Reference
## Prediction           Attrited Customer Existing Customer
##   Attrited Customer                135                26
##   Existing Customer                190              1674
##
##                Accuracy : 0.8933
##                  95% CI : (0.8791, 0.9064)
##     No Information Rate : 0.8395
##     P-Value [Acc > NIR] : 2.386e-12
##
##                   Kappa : 0.5027
```

```
##
##  Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.41538
##             Specificity : 0.98471
##          Pos Pred Value : 0.83851
##          Neg Pred Value : 0.89807
##              Prevalence : 0.16049
##          Detection Rate : 0.06667
##    Detection Prevalence : 0.07951
##       Balanced Accuracy : 0.70005
##
##        'Positive' Class : Attrited Customer
##
```

```r
#Naive Bayes for Regular Data
naive_bayes<- naiveBayes(Attrition_Flag ~ ., data= training_reg)
naive_bayes
```

```
##
## Naive Bayes Classifier for Discrete Predictors
##
## Call:
## naiveBayes.default(x = X, y = Y, laplace = laplace)
##
## A-priori probabilities:
## Y
## Attrited Customer Existing Customer
##         0.1607011         0.8392989
##
## Conditional probabilities:
##                    Customer_Age
## Y                        [,1]     [,2]
##   Attrited Customer 46.52688 7.628306
##   Existing Customer 46.26882 8.080203
##
##                    Gender
## Y                           F         M
##   Attrited Customer 0.5622120 0.4377880
##   Existing Customer 0.5204412 0.4795588
##
##                    Dependent_count
## Y                        [,1]     [,2]
##   Attrited Customer 2.423195 1.279478
##   Existing Customer 2.337647 1.303867
##
##                    Education_Level
## Y                       College  Doctorate   Graduate High School Post-Graduate
##   Attrited Customer 0.09600614 0.06144393 0.30414747  0.19124424    0.05529954
##   Existing Customer 0.10220588 0.04308824 0.30955882  0.19750000    0.04823529
##                    Education_Level
## Y                    Uneducated    Unknown
##   Attrited Customer 0.14208909 0.14976959
##   Existing Customer 0.15029412 0.14911765
##
```

```
##                     Marital_Status
## Y                       Divorced    Married     Single    Unknown
##   Attrited Customer 0.07910906 0.43087558 0.41090630 0.07910906
##   Existing Customer 0.07132353 0.46779412 0.38764706 0.07323529
##
##                     Income_Category
## Y                       $120K + $40K - $60K $60K - $80K $80K - $120K
##   Attrited Customer 0.07834101  0.15668203  0.11751152   0.15207373
##   Existing Customer 0.06941176  0.18235294  0.14147059   0.15367647
##                     Income_Category
## Y                     Less than $40K    Unknown
##   Attrited Customer     0.38479263 0.11059908
##   Existing Customer     0.34676471 0.10632353
##
##                     Card_Category
## Y                           Blue        Gold    Platinum      Silver
##   Attrited Customer 0.929339478 0.011520737 0.002304147 0.056835637
##   Existing Customer 0.931029412 0.011029412 0.001470588 0.056470588
##
##                     Months_on_book
## Y                         [,1]     [,2]
##   Attrited Customer 36.14209 7.806889
##   Existing Customer 35.89221 8.001434
##
##                     Total_Relationship_Count
## Y                         [,1]     [,2]
##   Attrited Customer 3.291859 1.577877
##   Existing Customer 3.911618 1.530966
##
##                     Months_Inactive_12_mon
## Y                         [,1]      [,2]
##   Attrited Customer 2.701997 0.8925237
##   Existing Customer 2.271765 1.0121906
##
##                     Contacts_Count_12_mon
## Y                         [,1]     [,2]
##   Attrited Customer 2.958525 1.087897
##   Existing Customer 2.361029 1.078113
##
##                     Credit_Limit
## Y                         [,1]     [,2]
##   Attrited Customer 8228.720 9148.195
##   Existing Customer 8719.856 9103.459
##
##                     Total_Revolving_Bal
## Y                          [,1]     [,2]
##   Attrited Customer  668.6398 928.8547
##   Existing Customer 1256.2413 758.4825
##
##                     Avg_Open_To_Buy
## Y                         [,1]     [,2]
##   Attrited Customer 7560.080 9159.224
##   Existing Customer 7463.614 9105.594
##
```

```
##                    Total_Amt_Chng_Q4_Q1
## Y                         [,1]      [,2]
##   Attrited Customer 0.6961160 0.2094967
##   Existing Customer 0.7713631 0.2175390
##
##                    Total_Trans_Amt
## Y                        [,1]     [,2]
##   Attrited Customer 3123.099 2319.419
##   Existing Customer 4663.088 3516.056
##
##                    Total_Trans_Ct
## Y                        [,1]     [,2]
##   Attrited Customer 45.08525 14.62268
##   Existing Customer 68.68721 22.87095
##
##                    Total_Ct_Chng_Q4_Q1
## Y                         [,1]      [,2]
##   Attrited Customer 0.5584032 0.2281168
##   Existing Customer 0.7413909 0.2263850
##
##                    Avg_Utilization_Ratio
## Y                         [,1]      [,2]
##   Attrited Customer 0.1579439 0.2604395
##   Existing Customer 0.2967232 0.2724107
```

```
nb_pred<- predict(naive_bayes, testing_reg)
caret::confusionMatrix(nb_pred, testing_reg$Attrition_Flag)
```

```
## Confusion Matrix and Statistics
##
##                    Reference
## Prediction          Attrited Customer Existing Customer
##   Attrited Customer               209               111
##   Existing Customer               116              1589
##
##                Accuracy : 0.8879
##                  95% CI : (0.8733, 0.9013)
##     No Information Rate : 0.8395
##     P-Value [Acc > NIR] : 3.345e-10
##
##                   Kappa : 0.5814
##
##  Mcnemar's Test P-Value : 0.7906
##
##             Sensitivity : 0.6431
##             Specificity : 0.9347
##          Pos Pred Value : 0.6531
##          Neg Pred Value : 0.9320
##              Prevalence : 0.1605
##          Detection Rate : 0.1032
##    Detection Prevalence : 0.1580
##       Balanced Accuracy : 0.7889
##
##        'Positive' Class : Attrited Customer
##
```

```r
#Decision tree for PCA data
decision_tree <- ctree(Attrition_Flag ~ ., data= training_pca)
decision_tree
```

```
##
##   Conditional inference tree with 61 terminal nodes
##
## Response:  Attrition_Flag
## Inputs:  Gender, Education_Level, Marital_Status, Income_Category, Card_Category, PC1, PC2, PC3, PC4
## Number of observations:  8102
##
## 1) PC2 <= -0.9635435; criterion = 1, statistic = 950.66
##   2) PC4 <= 0.6817504; criterion = 1, statistic = 573.357
##     3) PC9 <= -1.497358; criterion = 1, statistic = 66.133
##       4) PC4 <= -1.025709; criterion = 1, statistic = 19.675
##         5)*  weights = 19
##       4) PC4 > -1.025709
##         6)*  weights = 37
##     3) PC9 > -1.497358
##       7) PC2 <= -2.534985; criterion = 1, statistic = 48.592
##         8) PC5 <= 0.2753095; criterion = 0.974, statistic = 13.39
##           9) Marital_Status == {Married}; criterion = 0.999, statistic = 22.334
##             10)*  weights = 48
##           9) Marital_Status == {Divorced, Single, Unknown}
##             11) PC7 <= -0.6197529; criterion = 0.989, statistic = 11.351
##               12)*  weights = 10
##             11) PC7 > -0.6197529
##               13)*  weights = 34
##         8) PC5 > 0.2753095
##           14) PC9 <= 0.2845838; criterion = 0.969, statistic = 18.744
##             15)*  weights = 17
##           14) PC9 > 0.2845838
##             16)*  weights = 22
##       7) PC2 > -2.534985
##         17) PC4 <= -0.04732331; criterion = 1, statistic = 36.126
##           18) PC7 <= -0.8190895; criterion = 1, statistic = 17.913
##             19)*  weights = 130
##           18) PC7 > -0.8190895
##             20) PC10 <= -1.555832; criterion = 0.995, statistic = 12.907
##               21)*  weights = 27
##             20) PC10 > -1.555832
##               22)*  weights = 624
##         17) PC4 > -0.04732331
##           23) PC5 <= 0.206825; criterion = 0.998, statistic = 14.792
##             24) Education_Level == {College, Doctorate, High School, Uneducated, Unknown}; criterion
##               25)*  weights = 131
##             24) Education_Level == {Graduate, Post-Graduate}
##               26)*  weights = 95
##           23) PC5 > 0.206825
##             27)*  weights = 159
##   2) PC4 > 0.6817504
##     28) PC9 <= -0.5817971; criterion = 1, statistic = 106.027
##       29) PC7 <= 1.052803; criterion = 1, statistic = 30.593
##         30) PC4 <= 1.276292; criterion = 1, statistic = 17.327
```

```
##           31) PC5 <= 0.1337475; criterion = 0.961, statistic = 9.039
##             32)*  weights = 38
##           31) PC5 > 0.1337475
##             33)*  weights = 61
##         30) PC4 > 1.276292
##           34)*  weights = 198
##       29) PC7 > 1.052803
##         35)*  weights = 28
##     28) PC9 > -0.5817971
##       36) PC3 <= 0.339457; criterion = 1, statistic = 57.5
##         37) Gender == {M}; criterion = 1, statistic = 20.459
##           38) PC2 <= -1.156543; criterion = 0.999, statistic = 17.144
##             39)*  weights = 105
##           38) PC2 > -1.156543
##             40)*  weights = 16
##         37) Gender == {F}
##           41) PC9 <= 1.060199; criterion = 0.976, statistic = 9.963
##             42)*  weights = 176
##           41) PC9 > 1.060199
##             43) PC2 <= -1.529595; criterion = 0.974, statistic = 9.759
##               44)*  weights = 13
##             43) PC2 > -1.529595
##               45)*  weights = 9
##       36) PC3 > 0.339457
##         46) PC2 <= -2.337871; criterion = 1, statistic = 24.519
##           47)*  weights = 54
##         46) PC2 > -2.337871
##           48)*  weights = 89
## 1) PC2 > -0.9635435
##   49) PC9 <= -1.566984; criterion = 1, statistic = 239.533
##     50) PC4 <= -0.2457022; criterion = 1, statistic = 60.74
##       51) PC4 <= -1.139457; criterion = 0.962, statistic = 10.011
##         52)*  weights = 67
##       51) PC4 > -1.139457
##         53)*  weights = 45
##     50) PC4 > -0.2457022
##       54) PC2 <= 0.974554; criterion = 1, statistic = 33.613
##         55) PC3 <= 0.9303351; criterion = 0.994, statistic = 12.439
##           56)*  weights = 64
##         55) PC3 > 0.9303351
##           57)*  weights = 13
##       54) PC2 > 0.974554
##         58)*  weights = 21
##   49) PC9 > -1.566984
##     59) PC2 <= 0.1846674; criterion = 1, statistic = 136.267
##       60) PC9 <= -0.3929634; criterion = 1, statistic = 70.517
##         61) PC4 <= 0.2558358; criterion = 1, statistic = 64.815
##           62) PC4 <= -0.7520039; criterion = 1, statistic = 17.404
##             63)*  weights = 194
##           62) PC4 > -0.7520039
##             64) PC6 <= 0.6311245; criterion = 0.995, statistic = 12.77
##               65)*  weights = 126
##             64) PC6 > 0.6311245
##               66)*  weights = 42
```

```
##         61) PC4 > 0.2558358
##           67) PC8 <= 1.171834; criterion = 0.999, statistic = 16.693
##             68) PC7 <= 0.4260052; criterion = 1, statistic = 17.446
##               69) PC8 <= -0.8717067; criterion = 0.988, statistic = 12.163
##                 70)*  weights = 26
##               69) PC8 > -0.8717067
##                 71) PC1 <= -0.03114795; criterion = 0.988, statistic = 11.223
##                   72)*  weights = 59
##                 71) PC1 > -0.03114795
##                   73)*  weights = 69
##             68) PC7 > 0.4260052
##               74) PC3 <= -2.664937; criterion = 0.995, statistic = 12.992
##                 75)*  weights = 8
##               74) PC3 > -2.664937
##                 76)*  weights = 70
##           67) PC8 > 1.171834
##             77)*  weights = 28
##       60) PC9 > -0.3929634
##         78) Gender == {M}; criterion = 0.999, statistic = 16.443
##           79) PC7 <= -0.2020747; criterion = 0.997, statistic = 13.838
##             80) PC1 <= -4.043037; criterion = 0.989, statistic = 11.32
##               81)*  weights = 9
##             80) PC1 > -4.043037
##               82) PC6 <= 1.392423; criterion = 0.99, statistic = 11.615
##                 83)*  weights = 360
##               82) PC6 > 1.392423
##                 84)*  weights = 22
##           79) PC7 > -0.2020747
##             85)*  weights = 422
##         78) Gender == {F}
##           86) PC3 <= -1.639108; criterion = 1, statistic = 40.501
##             87) PC2 <= -0.4460463; criterion = 0.999, statistic = 16.245
##               88) PC4 <= 0.528874; criterion = 0.991, statistic = 11.763
##                 89)*  weights = 17
##               88) PC4 > 0.528874
##                 90)*  weights = 19
##             87) PC2 > -0.4460463
##               91)*  weights = 49
##           86) PC3 > -1.639108
##             92) PC5 <= -1.043276; criterion = 1, statistic = 54.455
##               93)*  weights = 89
##             92) PC5 > -1.043276
##               94) Card_Category == {Blue, Gold, Platinum}; criterion = 0.995, statistic = 18.532
##                 95) PC3 <= 0.0336585; criterion = 0.966, statistic = 16.542
##                   96)*  weights = 309
##                 95) PC3 > 0.0336585
##                   97)*  weights = 390
##               94) Card_Category == {Silver}
##                 98)*  weights = 26
##     59) PC2 > 0.1846674
##       99) PC9 <= -1.067416; criterion = 1, statistic = 49.511
##         100) PC2 <= 0.8259201; criterion = 0.99, statistic = 17.475
##           101) PC5 <= -0.8546402; criterion = 0.961, statistic = 17.884
##             102)*  weights = 40
```

```
##           101) PC5 > -0.8546402
##             103) PC1 <= -1.398596; criterion = 0.975, statistic = 15.327
##               104)*  weights = 7
##             103) PC1 > -1.398596
##               105)*  weights = 71
##         100) PC2 > 0.8259201
##           106)*  weights = 215
##       99) PC9 > -1.067416
##         107) PC3 <= -1.834285; criterion = 1, statistic = 30.95
##           108) PC1 <= -0.4544154; criterion = 0.999, statistic = 16.499
##             109)*  weights = 39
##           108) PC1 > -0.4544154
##             110)*  weights = 215
##         107) PC3 > -1.834285
##           111) PC2 <= 0.8187929; criterion = 1, statistic = 21.777
##             112) Card_Category == {Gold, Silver}; criterion = 1, statistic = 32.891
##               113)*  weights = 48
##             112) Card_Category == {Blue, Platinum}
##               114) PC9 <= -0.5227914; criterion = 0.997, statistic = 14.137
##                 115)*  weights = 152
##               114) PC9 > -0.5227914
##                 116) PC8 <= -1.989243; criterion = 0.985, statistic = 10.841
##                   117)*  weights = 12
##                 116) PC8 > -1.989243
##                   118)*  weights = 831
##           111) PC2 > 0.8187929
##             119) Gender == {F}; criterion = 0.995, statistic = 13.03
##               120)*  weights = 1106
##             119) Gender == {M}
##               121)*  weights = 682
```

```
dt_pred<- predict(decision_tree, testing_pca)
caret::confusionMatrix(dt_pred, testing_pca$Attrition_Flag)
```

```
## Confusion Matrix and Statistics
##
##                    Reference
## Prediction          Attrited Customer Existing Customer
##   Attrited Customer               161                50
##   Existing Customer               164              1650
##
##                Accuracy : 0.8943
##                  95% CI : (0.8801, 0.9074)
##     No Information Rate : 0.8395
##     P-Value [Acc > NIR] : 9.099e-13
##
##                   Kappa : 0.543
##
##  Mcnemar's Test P-Value : 1.123e-14
##
##             Sensitivity : 0.49538
##             Specificity : 0.97059
##          Pos Pred Value : 0.76303
##          Neg Pred Value : 0.90959
##              Prevalence : 0.16049
```

```
##          Detection Rate : 0.07951
##    Detection Prevalence : 0.10420
##       Balanced Accuracy : 0.73299
##
##        'Positive' Class : Attrited Customer
##
```

```r
#Decision tree for Regular data
decision_tree <- ctree(Attrition_Flag ~ ., data= training_reg)
decision_tree
```

```
##
##   Conditional inference tree with 53 terminal nodes
##
## Response:  Attrition_Flag
## Inputs:  Customer_Age, Gender, Dependent_count, Education_Level, Marital_Status, Income_Category, Ca
## Number of observations:  8102
##
## 1) Total_Trans_Ct <= 54; criterion = 1, statistic = 1109.821
##   2) Total_Revolving_Bal <= 613; criterion = 1, statistic = 466.695
##     3) Total_Ct_Chng_Q4_Q1 <= 0.645; criterion = 1, statistic = 116.853
##       4) Total_Relationship_Count <= 2; criterion = 1, statistic = 38.367
##         5)*  weights = 167
##       4) Total_Relationship_Count > 2
##         6) Total_Trans_Amt <= 2069; criterion = 1, statistic = 27.702
##           7) Total_Ct_Chng_Q4_Q1 <= 0.5; criterion = 0.999, statistic = 15.703
##             8) Months_Inactive_12_mon <= 1; criterion = 0.996, statistic = 13.886
##               9)*  weights = 28
##             8) Months_Inactive_12_mon > 1
##               10)*  weights = 132
##           7) Total_Ct_Chng_Q4_Q1 > 0.5
##             11)*  weights = 75
##         6) Total_Trans_Amt > 2069
##           12) Customer_Age <= 31; criterion = 1, statistic = 22.84
##             13)*  weights = 14
##           12) Customer_Age > 31
##             14) Total_Trans_Ct <= 51; criterion = 1, statistic = 19.79
##               15)*  weights = 243
##             14) Total_Trans_Ct > 51
##               16)*  weights = 11
##     3) Total_Ct_Chng_Q4_Q1 > 0.645
##       17) Total_Relationship_Count <= 2; criterion = 1, statistic = 34.255
##         18)*  weights = 51
##       17) Total_Relationship_Count > 2
##         19) Total_Trans_Amt <= 1970; criterion = 0.993, statistic = 15.276
##           20)*  weights = 107
##         19) Total_Trans_Amt > 1970
##           21) Total_Amt_Chng_Q4_Q1 <= 1.047; criterion = 0.999, statistic = 16.456
##             22)*  weights = 77
##           21) Total_Amt_Chng_Q4_Q1 > 1.047
##             23)*  weights = 11
##   2) Total_Revolving_Bal > 613
##     24) Total_Relationship_Count <= 2; criterion = 1, statistic = 207.413
##       25) Total_Ct_Chng_Q4_Q1 <= 0.8; criterion = 1, statistic = 49.912
##         26) Total_Amt_Chng_Q4_Q1 <= 0.861; criterion = 0.998, statistic = 14.929
```

```
##          27)*  weights = 112
##        26) Total_Amt_Chng_Q4_Q1 > 0.861
##          28)*  weights = 25
##      25) Total_Ct_Chng_Q4_Q1 > 0.8
##        29)*  weights = 30
##    24) Total_Relationship_Count > 2
##      30) Total_Trans_Amt <= 2100; criterion = 1, statistic = 108.739
##        31) Total_Ct_Chng_Q4_Q1 <= 0.4; criterion = 1, statistic = 40.69
##          32) Total_Trans_Ct <= 24; criterion = 0.987, statistic = 11.521
##            33)*  weights = 20
##          32) Total_Trans_Ct > 24
##            34) Total_Amt_Chng_Q4_Q1 <= 0.408; criterion = 0.957, statistic = 9.577
##              35) Customer_Age <= 51; criterion = 0.974, statistic = 10.23
##                36)*  weights = 17
##              35) Customer_Age > 51
##                37)*  weights = 15
##            34) Total_Amt_Chng_Q4_Q1 > 0.408
##              38)*  weights = 136
##        31) Total_Ct_Chng_Q4_Q1 > 0.4
##          39) Total_Amt_Chng_Q4_Q1 <= 0.411; criterion = 0.997, statistic = 14.311
##            40)*  weights = 43
##          39) Total_Amt_Chng_Q4_Q1 > 0.411
##            41) Marital_Status == {Divorced, Married, Unknown}; criterion = 0.955, statistic = 14.37
##              42)*  weights = 793
##            41) Marital_Status == {Single}
##              43)*  weights = 234
##      30) Total_Trans_Amt > 2100
##        44) Total_Ct_Chng_Q4_Q1 <= 0.793; criterion = 1, statistic = 86.618
##          45) Total_Amt_Chng_Q4_Q1 <= 0.889; criterion = 1, statistic = 42.539
##            46) Customer_Age <= 34; criterion = 1, statistic = 34.322
##              47)*  weights = 29
##            46) Customer_Age > 34
##              48) Total_Trans_Ct <= 45; criterion = 1, statistic = 26.935
##                49) Income_Category == {$120K +, $40K - $60K, $80K - $120K, Less than $40K, Unknown}
##                  50)*  weights = 89
##                49) Income_Category == {$60K - $80K}
##                  51)*  weights = 14
##              48) Total_Trans_Ct > 45
##                52) Avg_Utilization_Ratio <= 0.275; criterion = 0.999, statistic = 15.674
##                  53)*  weights = 31
##                52) Avg_Utilization_Ratio > 0.275
##                  54)*  weights = 46
##          45) Total_Amt_Chng_Q4_Q1 > 0.889
##            55) Total_Trans_Amt <= 2730; criterion = 0.998, statistic = 14.735
##              56) Total_Ct_Chng_Q4_Q1 <= 0.577; criterion = 0.995, statistic = 14.456
##                57)*  weights = 16
##              56) Total_Ct_Chng_Q4_Q1 > 0.577
##                58)*  weights = 39
##            55) Total_Trans_Amt > 2730
##              59)*  weights = 18
##        44) Total_Ct_Chng_Q4_Q1 > 0.793
##          60)*  weights = 133
## 1) Total_Trans_Ct > 54
##   61) Total_Trans_Ct <= 64; criterion = 1, statistic = 152.335
```

```
##      62) Total_Trans_Amt <= 5342; criterion = 1, statistic = 190.363
##        63) Total_Relationship_Count <= 2; criterion = 1, statistic = 31.997
##          64) Total_Trans_Ct <= 57; criterion = 1, statistic = 25.307
##            65)*  weights = 15
##          64) Total_Trans_Ct > 57
##            66)*  weights = 42
##        63) Total_Relationship_Count > 2
##          67) Total_Revolving_Bal <= 304; criterion = 1, statistic = 17.817
##            68)*  weights = 174
##          67) Total_Revolving_Bal > 304
##            69) Total_Trans_Ct <= 59; criterion = 0.966, statistic = 9.75
##              70)*  weights = 262
##            69) Total_Trans_Ct > 59
##              71)*  weights = 361
##      62) Total_Trans_Amt > 5342
##        72)*  weights = 55
##    61) Total_Trans_Ct > 64
##      73) Total_Amt_Chng_Q4_Q1 <= 0.891; criterion = 1, statistic = 109.308
##        74) Avg_Utilization_Ratio <= 0.027; criterion = 1, statistic = 44.738
##          75) Total_Ct_Chng_Q4_Q1 <= 0.978; criterion = 1, statistic = 26.652
##            76) Contacts_Count_12_mon <= 2; criterion = 0.998, statistic = 15.063
##              77)*  weights = 452
##            76) Contacts_Count_12_mon > 2
##              78) Total_Trans_Amt <= 5472; criterion = 1, statistic = 18.005
##                79)*  weights = 284
##              78) Total_Trans_Amt > 5472
##                80) Total_Trans_Ct <= 78; criterion = 1, statistic = 33.188
##                  81)*  weights = 17
##                80) Total_Trans_Ct > 78
##                  82)*  weights = 41
##          75) Total_Ct_Chng_Q4_Q1 > 0.978
##            83) Total_Trans_Amt <= 4919; criterion = 1, statistic = 35.491
##              84)*  weights = 29
##            83) Total_Trans_Amt > 4919
##              85)*  weights = 12
##        74) Avg_Utilization_Ratio > 0.027
##          86) Card_Category == {Blue, Platinum}; criterion = 1, statistic = 25.658
##            87)*  weights = 2672
##          86) Card_Category == {Gold, Silver}
##            88) Total_Trans_Ct <= 71; criterion = 0.995, statistic = 13.257
##              89) Total_Trans_Amt <= 4826; criterion = 1, statistic = 22.51
##                90)*  weights = 28
##              89) Total_Trans_Amt > 4826
##                91)*  weights = 9
##            88) Total_Trans_Ct > 71
##              92)*  weights = 191
##      73) Total_Amt_Chng_Q4_Q1 > 0.891
##        93) Avg_Utilization_Ratio <= 0.027; criterion = 1, statistic = 38.35
##          94) Total_Trans_Amt <= 5416; criterion = 1, statistic = 50.919
##            95) Card_Category == {Gold, Platinum, Silver}; criterion = 1, statistic = 30.638
##              96)*  weights = 8
##            95) Card_Category == {Blue}
##              97)*  weights = 118
##          94) Total_Trans_Amt > 5416
```

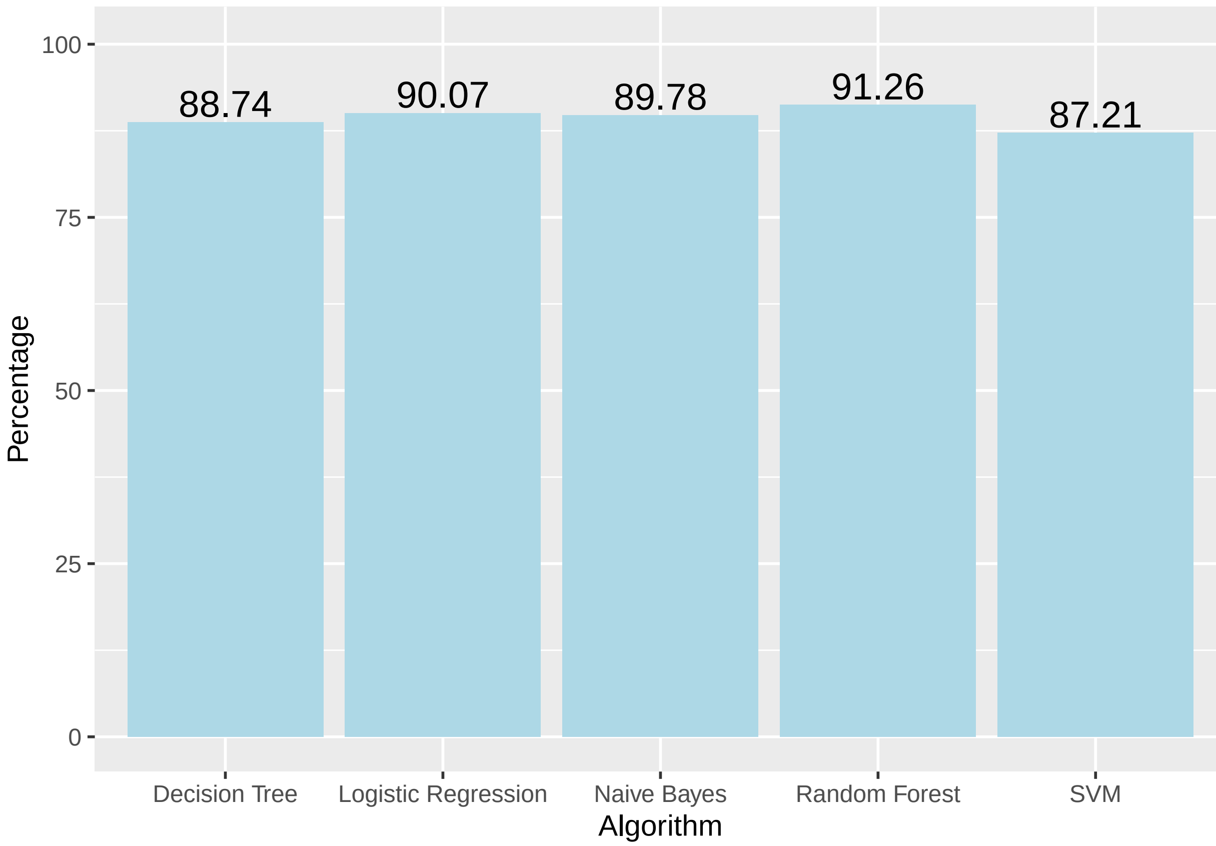
```
##           98) Total_Trans_Ct <= 89; criterion = 1, statistic = 39.1
##             99)*  weights = 44
##           98) Total_Trans_Ct > 89
##             100)*  weights = 11
##       93) Avg_Utilization_Ratio > 0.027
##         101) Total_Revolving_Bal <= 2473; criterion = 0.999, statistic = 16.199
##           102) Avg_Utilization_Ratio <= 0.182; criterion = 0.983, statistic = 10.991
##             103)*  weights = 172
##           102) Avg_Utilization_Ratio > 0.182
##             104)*  weights = 320
##         101) Total_Revolving_Bal > 2473
##           105)*  weights = 29
```

```
dt_pred<- predict(decision_tree, testing_reg)
caret::confusionMatrix(dt_pred, testing_reg$Attrition_Flag)
```

```
## Confusion Matrix and Statistics
##
##                    Reference
## Prediction          Attrited Customer Existing Customer
##   Attrited Customer               244                43
##   Existing Customer                81              1657
##
##                Accuracy : 0.9388
##                  95% CI : (0.9274, 0.9488)
##     No Information Rate : 0.8395
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.7615
##
##  Mcnemar's Test P-Value : 0.0008915
##
##             Sensitivity : 0.7508
##             Specificity : 0.9747
##          Pos Pred Value : 0.8502
##          Neg Pred Value : 0.9534
##              Prevalence : 0.1605
##          Detection Rate : 0.1205
##    Detection Prevalence : 0.1417
##       Balanced Accuracy : 0.8627
##
##        'Positive' Class : Attrited Customer
##
```

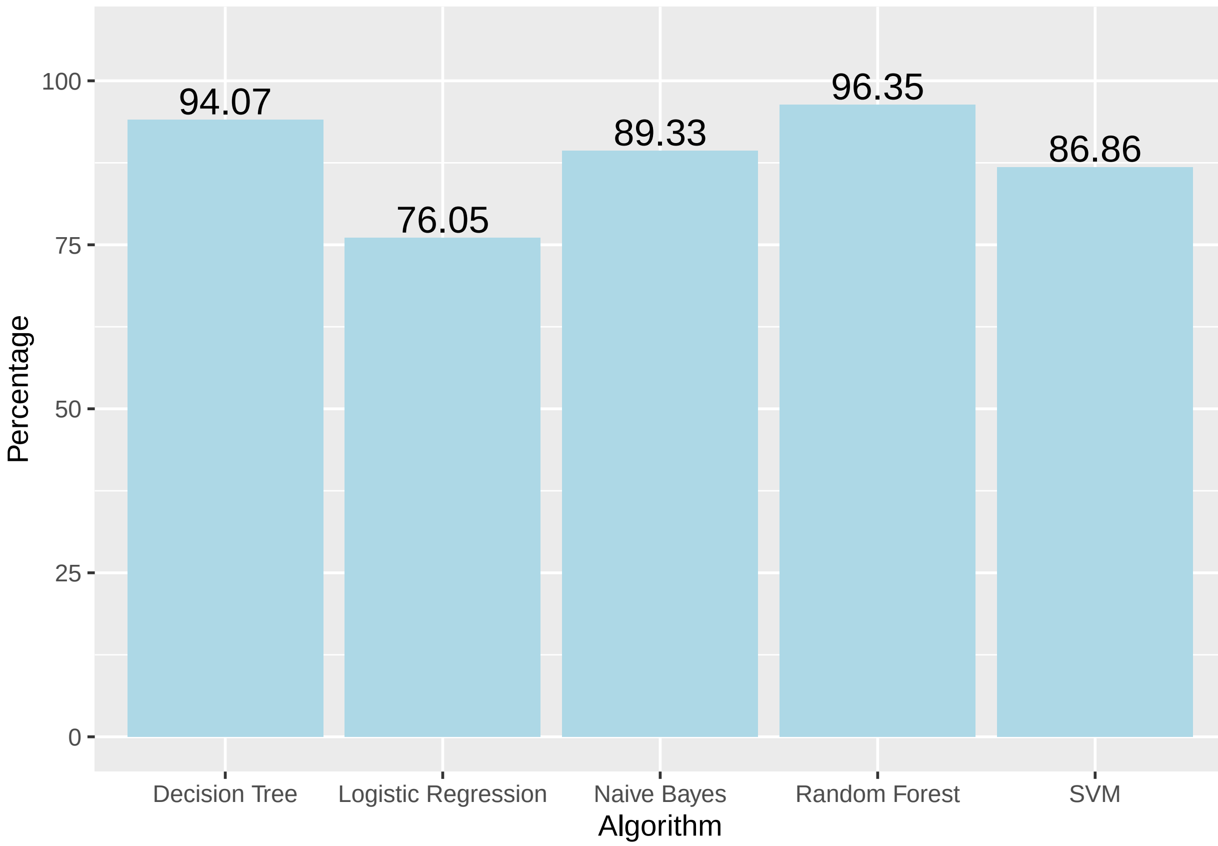
```
# Comparision of different models on PCA Data

H = c(91.26,87.21,89.78,88.74,90.07)
names1 = c("Random Forest","SVM" , "Naive Bayes","Decision Tree","Logistic Regression")
experiment <- data.frame(Algorithm = names1,
                         Percentage = H)
ggplot(data = experiment, mapping = aes(x=Algorithm, y=Percentage)) +
  geom_bar(stat="identity", position = "dodge",fill="lightblue") + scale_fill_brewer(palette = "Pastel2
  geom_text(aes(label = Percentage), vjust = -0.2, size = 5,
            position = position_dodge(0.9)) +
  ylim(0, max(experiment$Percentage)*1.1)
```




```
# Comparision of different models on Regular Data

H = c(96.35,86.86,89.33,94.07, 76.05)
names1 = c("Random Forest","SVM" , "Naive Bayes","Decision Tree","Logistic Regression")
experiment <- data.frame(Algorithm = names1,
                         Percentage = H)
ggplot(data = experiment, mapping = aes(x=Algorithm, y=Percentage)) +
  geom_bar(stat="identity", position = "dodge",fill="lightblue") + scale_fill_brewer(palette = "Pastel2
  geom_text(aes(label = Percentage), vjust = -0.2, size = 5,
            position = position_dodge(0.9)) +
  ylim(0, max(experiment$Percentage)*1.1)
```


94.07
76.05
89.33
96.35
86.86
100
75
50
25
0
Percentage
Decision Tree
Logistic Regression
Naive Bayes
Random Forest
SVM
Algorithm