India voice call quality customer experience machine learning analysis

```
# Packages
library(RColorBrewer)
library(ggridges)
library(cowplot)
```

```
##
## ************************************************************
## Note: As of version 1.0.0, cowplot does not change the
##   default ggplot2 theme anymore. To recover the previous
##   behavior, execute:
##   theme_set(theme_cowplot())
## ************************************************************
```

```
library(ggplot2)
library(mapview)
library(sp)
library(e1071)
library(caret)
```

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

```
library(randomForest)
```

```
## randomForest 4.6-14
## Type rfNews() to see new features/changes/bug fixes.
##
## Attaching package: 'randomForest'
## The following object is masked from 'package:ggplot2':
##
##     margin
```

```
library(naivebayes)
```

```
## naivebayes 0.9.7 loaded
```

```
# Loading dataset

calls <- read.csv("MyCall_Data_December_2020.csv", header = TRUE, sep = ",")

str(calls)
```

```
## 'data.frame':    20197 obs. of  8 variables:
##  $ Operator          : Factor w/ 6 levels "Airtel","BSNL",..: 3 3 3 3 3 3 3 3 3 3 ...
##  $ In.Out.Travelling : Factor w/ 3 levels "Indoor","Outdoor",..: 1 1 3 3 3 1 1 1 1 1 ...
```

```
##  $ Network.Type      : Factor w/ 4 levels "2G","3G","4G",..: 3 3 3 2 3 3 3 3 3 3 ...
##  $ Rating            : int  5 1 2 3 2 2 1 1 1 1 ...
##  $ Call.Drop.Category: Factor w/ 3 levels "Call Dropped",..: 3 2 2 2 2 2 2 2 2 1 ...
##  $ Latitude          : num  -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 ...
##  $ Longitude         : num  -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 ...
##  $ State.Name        : Factor w/ 27 levels "Andaman and Nicobar",..: NA NA NA NA NA NA NA NA NA NA .
```

```
#### Descirptive analysis ####

# Unique data Rating

c(unique(calls["Rating"]))
```

```
## $Rating
## [1] 5 1 2 3 4
```

```
# Unique data Call drop category

c(unique(calls["Call.Drop.Category"]))
```

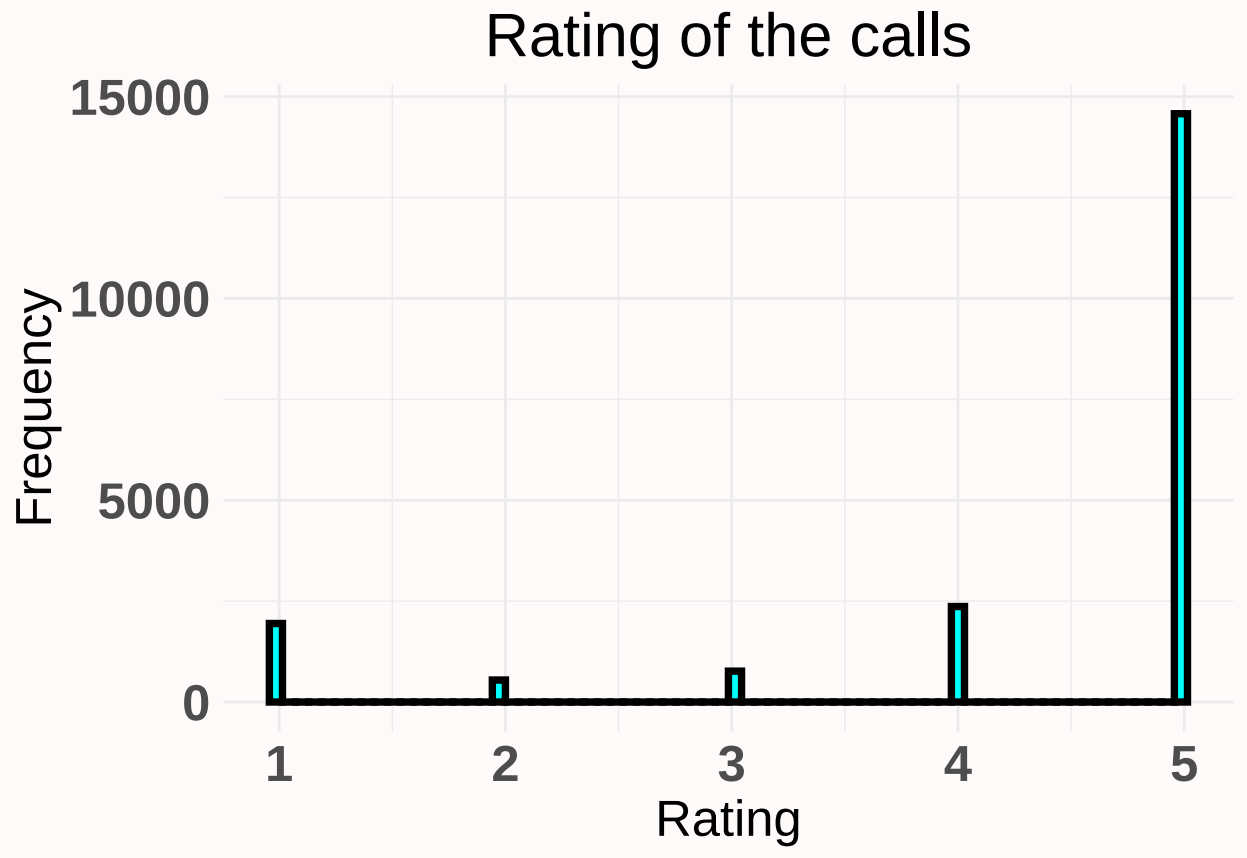
```
## $Call.Drop.Category
## [1] Satisfactory       Poor Voice Quality Call Dropped
## Levels: Call Dropped Poor Voice Quality Satisfactory
```

```
# Unique data State Name

c(unique(calls["State.Name"]))
```

```
## $State.Name
##  [1] <NA>
##  [2] Telangana
##  [3] Karnataka
##  [4] Maharashtra
##  [5] West Bengal
##  [6] Chhattisgarh
##  [7] Himachal Pradesh
##  [8] Madhya Pradesh
##  [9] Uttarakhand
## [10] Andhra Pradesh
## [11] Jharkhand
## [12] Uttar Pradesh
## [13] Gujarat
## [14] Kerala
## [15] Haryana
## [16] Tamil Nadu
## [17] Meghalaya
## [18] Rajasthan
## [19] Punjab
## [20] Assam
## [21] Delhi
## [22] Nagaland
## [23] Bihar
## [24] Dadra and Nagar Haveli and Daman and Diu
## [25] Jammu and Kashmir
## [26] Odisha
## [27] Andaman and Nicobar
```

```
## [28] Goa
## 27 Levels: Andaman and Nicobar Andhra Pradesh Assam Bihar ... West Bengal
```

```
# Histogram of ratings

tema <- theme(plot.background = element_rect(fill = "#FFFAFA", color = "#FFFAFA"),
              plot.title = element_text(size = 23, hjust = .5),
              axis.text.x = element_text(size = 19, face = "bold"),
              axis.text.y = element_text(size = 19, face = "bold"),
              axis.title.x = element_text(size = 19),
              axis.title.y = element_text(size = 19),
              legend.position = "none")

options(repr.plot.width=14, repr.plot.height=6)
a <- ggplot(data = calls, mapping = aes(x = Rating)) +
  geom_histogram(fill = "cyan", bins = 70, size = 1.3, color = "black") +
  theme_minimal() +
  ylab("Frequency") +
  xlab("Rating") +
  ggtitle("Rating of the calls") +
  tema

a
```




```
# Frequency of the states

freq_states <- data.frame(cbind(Frequency = table(calls$State.Name), Percent = prop.table(table(calls$S
```

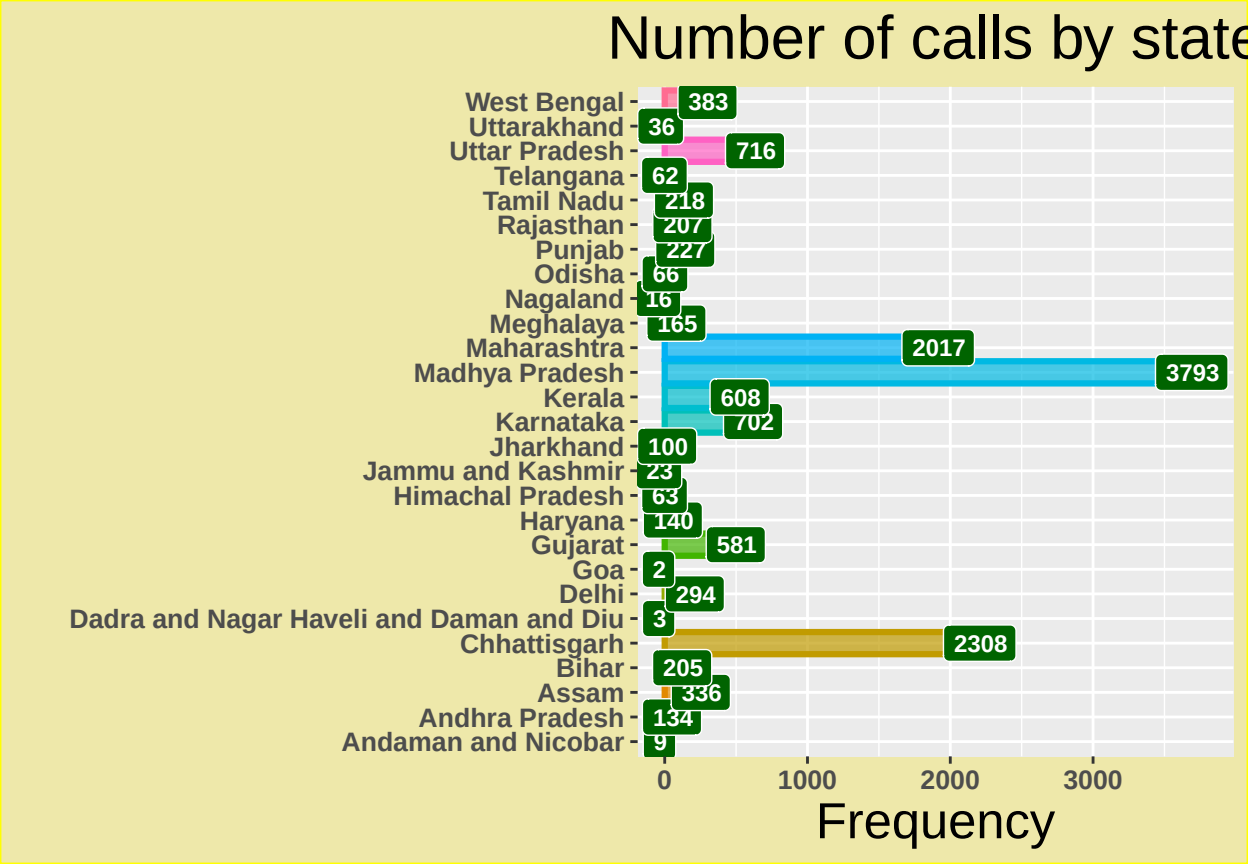
```
freq_states
```

```
##                                          Frequency     Percent
## Andaman and Nicobar                              9  0.06709408
## Andhra Pradesh                                 134  0.99895631
## Assam                                          336  2.50484568
## Bihar                                          205  1.52825406
## Chhattisgarh                                  2308 17.20590428
## Dadra and Nagar Haveli and Daman and Diu         3  0.02236469
## Delhi                                          294  2.19173997
## Goa                                              2  0.01490980
## Gujarat                                        581  4.33129566
## Haryana                                        140  1.04368570
## Himachal Pradesh                                63  0.46965857
## Jammu and Kashmir                               23  0.17146265
## Jharkhand                                      100  0.74548979
## Karnataka                                      702  5.23333830
## Kerala                                         608  4.53257790
## Madhya Pradesh                                3793 28.27642761
## Maharashtra                                   2017 15.03652900
## Meghalaya                                      165  1.23005815
## Nagaland                                        16  0.11927837
## Odisha                                          66  0.49202326
## Punjab                                         227  1.69226182
## Rajasthan                                      207  1.54316386
## Tamil Nadu                                     218  1.62516774
## Telangana                                       62  0.46220367
## Uttar Pradesh                                  716  5.33770687
## Uttarakhand                                     36  0.26837632
## West Bengal                                    383  2.85522588
```

```
str(freq_states)
```

```
## 'data.frame':    27 obs. of  2 variables:
##  $ Frequency: num  9 134 336 205 2308 ...
##  $ Percent  : num  0.0671 0.999 2.5048 1.5283 17.2059 ...
```

```
# Histogram of number of calls by state

tema <- theme(plot.background = element_rect(fill = "#EEE8AA", color = "yellow"),
              plot.title = element_text(size = 23, hjust = .5),
              axis.text.x = element_text(size = 10, face = "bold"),
              axis.text.y = element_text(size = 10, face = "bold"),
              axis.title.x = element_text(size = 19),
              axis.title.y = element_text(size = 19),
              legend.position = "none")

options(repr.plot.width=15, repr.plot.height=6)
a <- ggplot(data = freq_states, mapping = aes(x = Frequency, y = row.names(freq_states))) +
  geom_bar(stat = "identity", mapping = aes(fill = row.names(freq_states), color = row.names(freq_states
  geom_label(mapping = aes(label=Frequency), fill = "#006400", size = 3, color = "white", fontface = "bo
  ylab("") +
  ggtitle("Number of calls by state") +
  tema
a
```




```
# Frequency of network

freq_network <- data.frame(cbind(Frequency = table(calls$Network.Type), Percent = prop.table(table(calls
freq_network
```

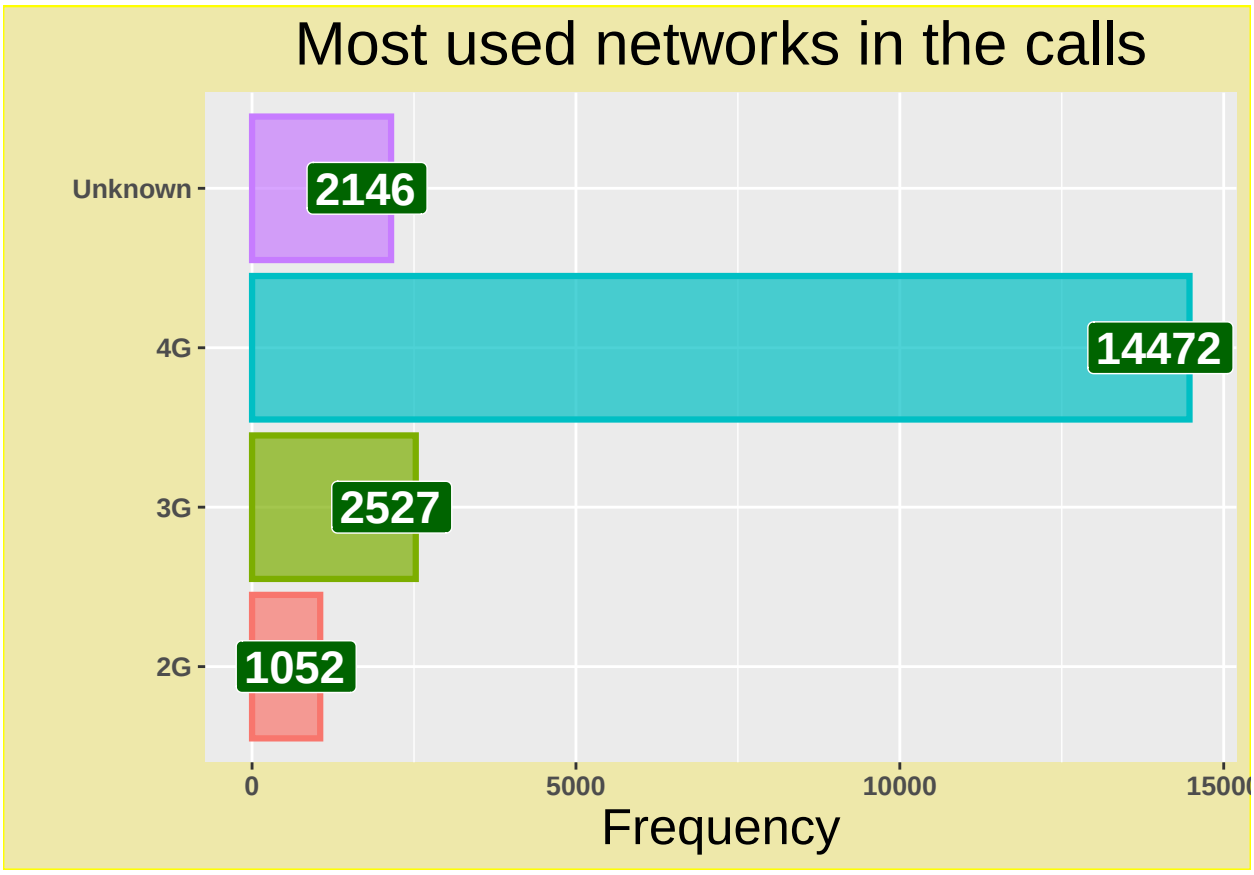
```
##         Frequency   Percent
## 2G           1052  5.208694
## 3G           2527 12.511759
## 4G          14472 71.654206
## Unknown      2146 10.625340
```

```
str(freq_network)
```

```
## 'data.frame':    4 obs. of  2 variables:
##  $ Frequency: num  1052 2527 14472 2146
##  $ Percent  : num  5.21 12.51 71.65 10.63
```

```
# Histogram of most used networks in the calls

tema <- theme(plot.background = element_rect(fill = "#EEE8AA", color = "yellow"),
              plot.title = element_text(size = 23, hjust = .5),
              axis.text.x = element_text(size = 10, face = "bold"),
              axis.text.y = element_text(size = 10, face = "bold"),
              axis.title.x = element_text(size = 19),
              axis.title.y = element_text(size = 19),
              legend.position = "none")

options(repr.plot.width=15, repr.plot.height=6)
```

```
b <- ggplot(data = freq_network, mapping = aes(x = Frequency, y = row.names(freq_network))) +
  geom_bar(stat = "identity", mapping = aes(fill = row.names(freq_network), color = row.names(freq_netw
  geom_label(mapping = aes(label=Frequency), fill = "#006400", size = 6, color = "white", fontface = "bo
  ylab("") +
  ggtitle("Most used networks in the calls") +
  tema
b
```

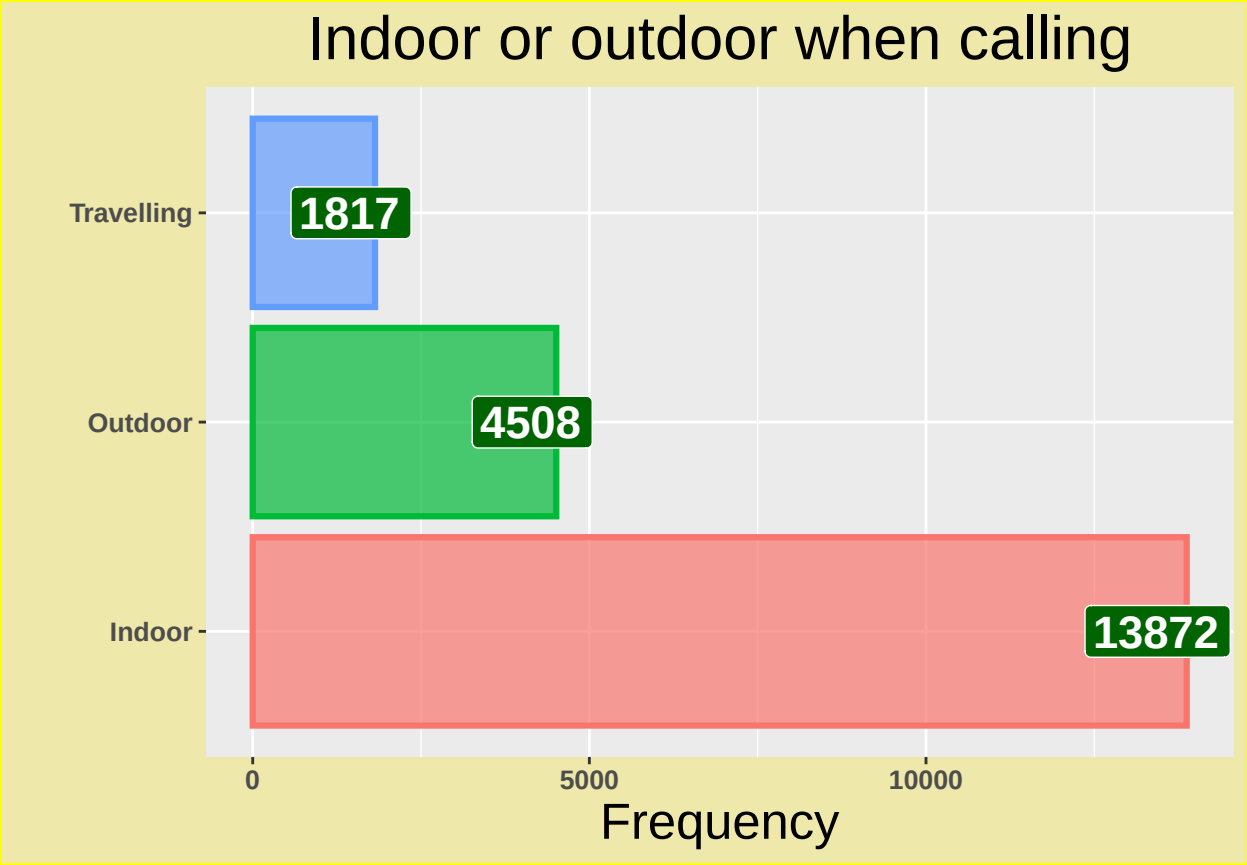



```
# Frequency of traveling in the calls

freq_travelling <- data.frame(cbind(Frequency = table(calls$In.Out.Travelling),
                                    Percent = prop.table(table(calls$In.Out.Travelling)) * 100))
freq_travelling
```

```
##            Frequency   Percent
## Indoor         13872 68.683468
## Outdoor         4508 22.320147
## Travelling      1817  8.996386
```

```
# Histogram of travelling

tema <- theme(plot.background = element_rect(fill = "#EEE8AA", color = "yellow"),
              plot.title = element_text(size = 23, hjust = .5),
              axis.text.x = element_text(size = 10, face = "bold"),
              axis.text.y = element_text(size = 10, face = "bold"),
              axis.title.x = element_text(size = 19),
              axis.title.y = element_text(size = 19),
```

```
              legend.position = "none")

options(repr.plot.width=15, repr.plot.height=6)
c <- ggplot(data = freq_travelling, mapping = aes(x = Frequency, y = row.names(freq_travelling))) +
  geom_bar(stat = "identity", mapping = aes(fill = row.names(freq_travelling), color = row.names(freq_t
  geom_label(mapping = aes(label=Frequency), fill = "#006400", size = 6, color = "white", fontface = "b
  ylab("") +
  ggtitle("Indoor or outdoor when calling") +
  tema
c
```

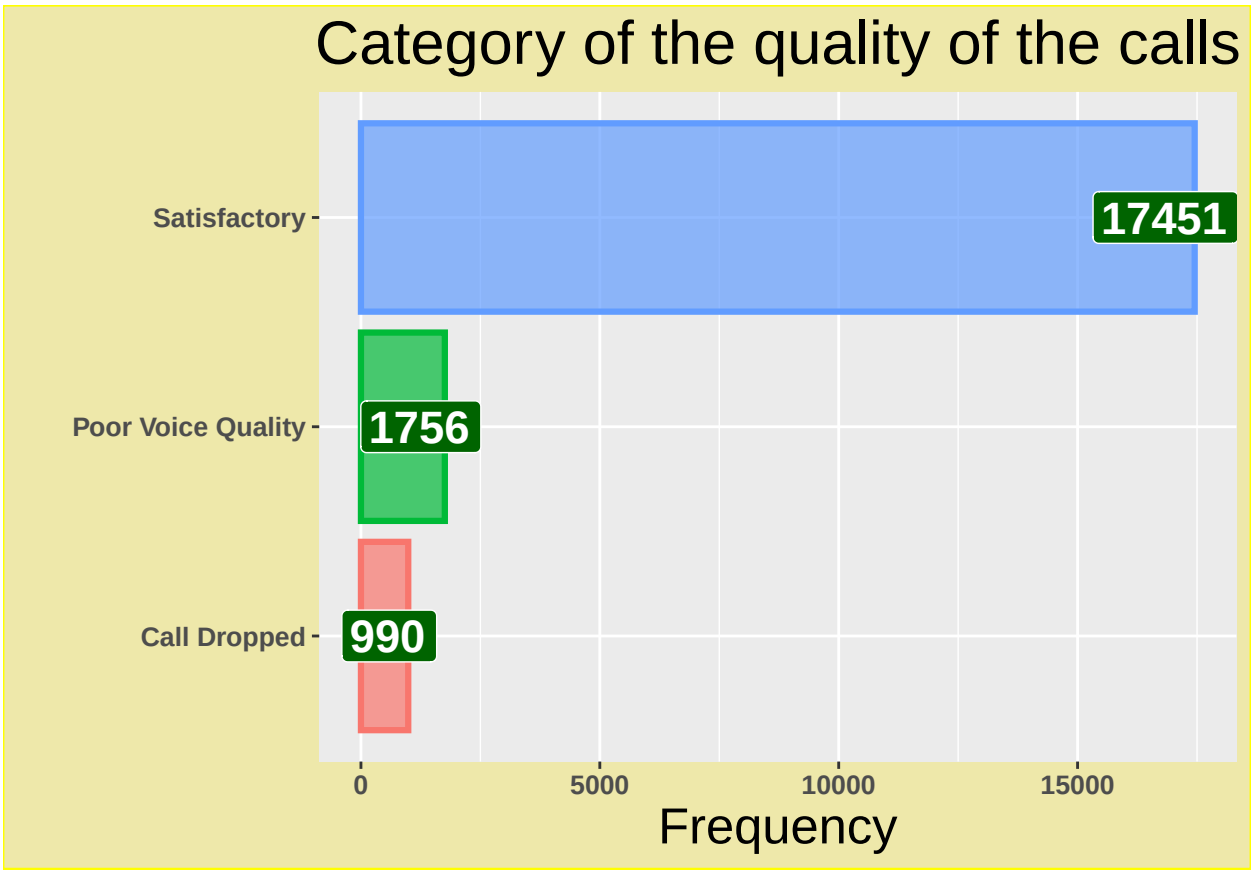



```
# Frequency of call drop category

freq_category <- data.frame(cbind(Frequency = table(calls$Call.Drop.Category),
                                  Percent = prop.table(table(calls$Call.Drop.Category)) * 100))
freq_category
```

```
##                     Frequency   Percent
## Call Dropped              990  4.901718
## Poor Voice Quality       1756  8.694361
## Satisfactory            17451 86.403921
```

```
# Histogram of calls category

tema <- theme(plot.background = element_rect(fill = "#EEE8AA", color = "yellow"),
              plot.title = element_text(size = 23, hjust = .5),
              axis.text.x = element_text(size = 10, face = "bold"),
```

```
              axis.text.y = element_text(size = 10, face = "bold"),
              axis.title.x = element_text(size = 19),
              axis.title.y = element_text(size = 19),
              legend.position = "none")

options(repr.plot.width=15, repr.plot.height=6)
d <- ggplot(data = freq_category, mapping = aes(x = Frequency, y = row.names(freq_category))) +
  geom_bar(stat = "identity", mapping = aes(fill = row.names(freq_category), color = row.names(freq_cat
  geom_label(mapping = aes(label=Frequency), fill = "#006400", size = 6, color = "white", fontface = "b
  ylab("") +
  ggtitle("Category of the quality of the calls") +
  tema
d
```




```
#### Pre processing ####

# Removing NA values

calls <- na.omit(calls)

# Transforming Rating variable into factor

calls$Rating <- factor(calls$Rating)

# Dividing the data into training and test
```

```
indexes <- sample(1:nrow(calls), size = 0.7 * nrow(calls))
train.data.calls <- calls[indexes,]
test.data.calls <- calls[-indexes,]
class(train.data.calls)
```

```
## [1] "data.frame"
```

```
class(test.data.calls)
```

```
## [1] "data.frame"
```

```
str(train.data.calls)
```

```
## 'data.frame':    9389 obs. of  8 variables:
##  $ Operator          : Factor w/ 6 levels "Airtel","BSNL",..: 3 5 6 1 2 3 3 5 6 6 ...
##  $ In.Out.Travelling : Factor w/ 3 levels "Indoor","Outdoor",..: 2 1 1 1 1 1 2 1 1 1 ...
##  $ Network.Type      : Factor w/ 4 levels "2G","3G","4G",..: 2 3 3 3 4 2 3 3 4 3 ...
##  $ Rating            : Factor w/ 5 levels "1","2","3","4",..: 5 4 5 5 5 5 5 5 1 5 ...
##  $ Call.Drop.Category: Factor w/ 3 levels "Call Dropped",..: 3 3 3 3 3 3 3 3 1 3 ...
##  $ Latitude          : num  21.89 20.72 23.16 26.72 8.51 ...
##  $ Longitude         : num  83.4 77 77.4 88.4 77 ...
##  $ State.Name        : Factor w/ 27 levels "Andaman and Nicobar",..: 5 17 16 27 15 10 16 5 17 16 ...
##  - attr(*, "na.action")= 'omit' Named int  1 2 3 4 5 6 7 8 9 10 ...
##   ..- attr(*, "names")= chr  "1" "2" "3" "4" ...
```

```
prop.table(table(train.data.calls$Rating)) * 100
```

```
##
##         1         2         3         4         5
##  9.596336  2.236660  2.822452 12.567899 72.776654
```

```
#### Machine learning ####

# Machine learning svm model 89% accuracy
ma_model_calls <- svm(Rating ~. ,data = train.data.calls)
summary(ma_model_calls)
```

```
##
## Call:
## svm(formula = Rating ~ ., data = train.data.calls)
##
##
## Parameters:
##    SVM-Type:  C-classification
##  SVM-Kernel:  radial
##        cost:  1
##
## Number of Support Vectors:  2778
##
##  ( 1028 889 386 210 265 )
##
##
## Number of Classes:  5
##
## Levels:
##  1 2 3 4 5
```

```
print(ma_model_calls)
```

```
##
## Call:
## svm(formula = Rating ~ ., data = train.data.calls)
##
##
## Parameters:
##    SVM-Type:  C-classification
##  SVM-Kernel:  radial
##        cost:  1
##
## Number of Support Vectors:  2778
```

```
# Predictions of model

pred_model_calls <- predict(ma_model_calls, test.data.calls)



table(pred_model_calls, test.data.calls$Rating)
```

```
##
## pred_model_calls    1    2    3    4    5
##                1  359   89   59    0    0
##                2    0    0    0    0    0
##                3    0    0    0    0    0
##                4    0    0   21  305   41
##                5    0    0   28  182 2941
```

```
confusionMatrix(pred_model_calls, test.data.calls$Rating)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    1    2    3    4    5
##          1  359   89   59    0    0
##          2    0    0    0    0    0
##          3    0    0    0    0    0
##          4    0    0   21  305   41
##          5    0    0   28  182 2941
##
## Overall Statistics
##
##                Accuracy : 0.8957
##                  95% CI : (0.8858, 0.9049)
##     No Information Rate : 0.7409
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.7376
##
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
```

```
##                      Class: 1 Class: 2 Class: 3 Class: 4 Class: 5
## Sensitivity           1.00000  0.00000  0.00000  0.62628   0.9863
## Specificity           0.95963  1.00000  1.00000  0.98248   0.7987
## Pos Pred Value        0.70809      NaN      NaN  0.83106   0.9334
## Neg Pred Value        1.00000  0.97789  0.97317  0.95025   0.9531
## Prevalence            0.08919  0.02211  0.02683  0.12099   0.7409
## Detection Rate        0.08919  0.00000  0.00000  0.07578   0.7307
## Detection Prevalence  0.12596  0.00000  0.00000  0.09118   0.7829
## Balanced Accuracy     0.97981  0.50000  0.50000  0.80438   0.8925
```

```
# Naive bayes model 84% accuracy

ma_model_calls <- naiveBayes(Rating~., train.data.calls)


# Predictions of model

pred_model_calls2 <- predict(ma_model_calls, test.data.calls)



table(pred_model_calls2, test.data.calls$Rating)
```

```
##
## pred_model_calls2    1    2    3    4    5
##                 1  299   57   39    0    0
##                 2   31   24    8    0    0
##                 3   27    8   12    0    0
##                 4    0    0   29  408  338
##                 5    2    0   20   79 2644
```

```
confusionMatrix(pred_model_calls2, test.data.calls$Rating)
```

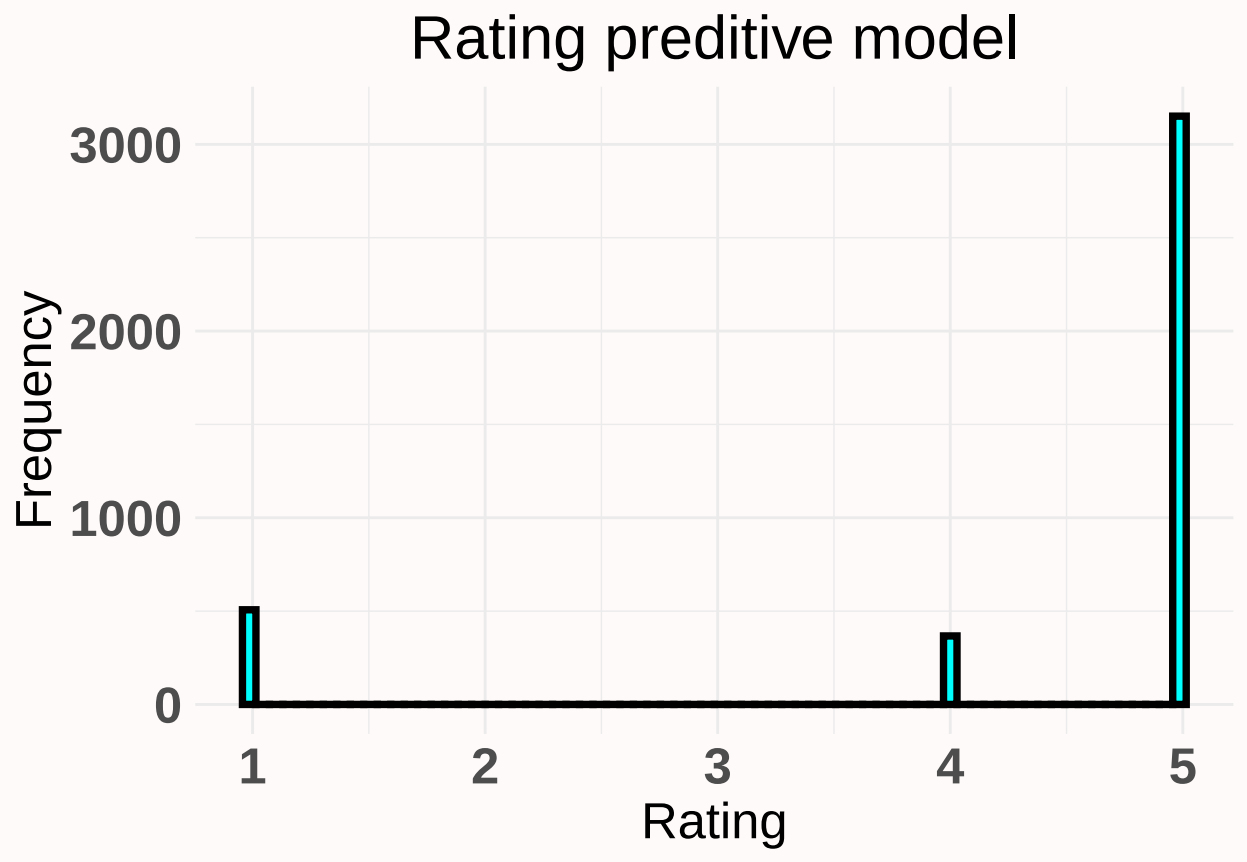
```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    1    2    3    4    5
##          1  299   57   39    0    0
##          2   31   24    8    0    0
##          3   27    8   12    0    0
##          4    0    0   29  408  338
##          5    2    0   20   79 2644
##
## Overall Statistics
##
##                Accuracy : 0.8415
##                  95% CI : (0.8298, 0.8526)
##     No Information Rate : 0.7409
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.6569
##
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
```

```
##
##                      Class: 1 Class: 2 Class: 3 Class: 4 Class: 5
## Sensitivity           0.83287 0.269663 0.111111   0.8378   0.8867
## Specificity           0.97381 0.990091 0.991065   0.8963   0.9032
## Pos Pred Value        0.75696 0.380952 0.255319   0.5265   0.9632
## Neg Pred Value        0.98347 0.983594 0.975867   0.9757   0.7359
## Prevalence            0.08919 0.022112 0.026832   0.1210   0.7409
## Detection Rate        0.07429 0.005963 0.002981   0.1014   0.6569
## Detection Prevalence  0.09814 0.015652 0.011677   0.1925   0.6820
## Balanced Accuracy     0.90334 0.629877 0.551088   0.8670   0.8949
```

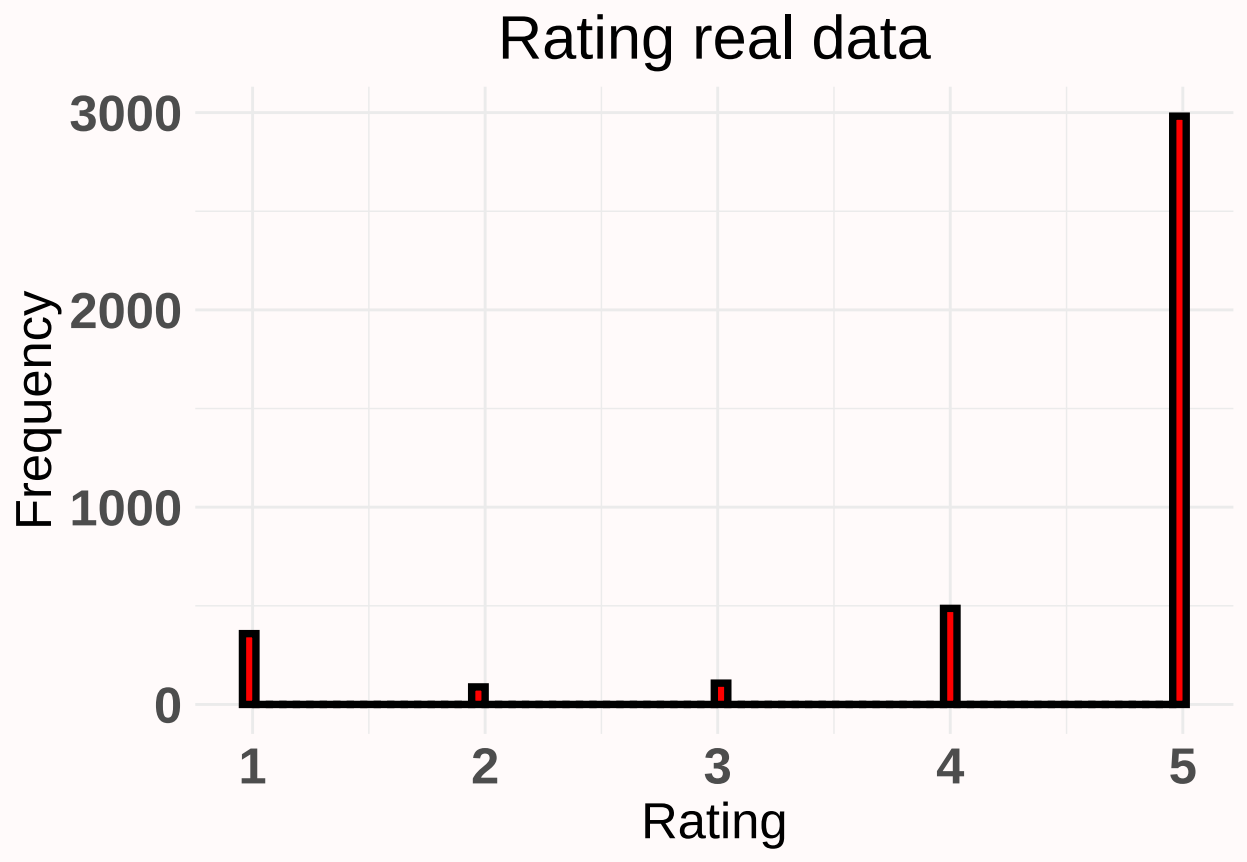
```
# predictive rating model plot

pred_model_calls_plot <- as.data.frame(pred_model_calls)
names(pred_model_calls_plot) <- c("Rating")

names(pred_model_calls_plot)
```

```
## [1] "Rating"
```

```
tema <- theme(plot.background = element_rect(fill = "#FFFAFA", color = "#FFFAFA"),
              plot.title = element_text(size = 23, hjust = .5),
              axis.text.x = element_text(size = 19, face = "bold"),
              axis.text.y = element_text(size = 19, face = "bold"),
              axis.title.x = element_text(size = 19),
              axis.title.y = element_text(size = 19),
              legend.position = "none")

options(repr.plot.width=14, repr.plot.height=6)
e <- ggplot(data = pred_model_calls_plot , mapping = aes(x = as.numeric(Rating))) +
  geom_histogram(fill = "cyan", bins = 70, size = 1.3, color = "black") +
  theme_minimal() +
  ylab("Frequency") +
  xlab("Rating") +
  ggtitle("Rating preditive model") +
  tema

e
```




```
# Real rating data plot

options(repr.plot.width=14, repr.plot.height=6)
f <- ggplot(data = test.data.calls , mapping = aes(x = as.numeric(Rating))) +
  geom_histogram(fill = "red", bins = 70, size = 1.3, color = "black") +
  theme_minimal() +
  ylab("Frequency") +
  xlab("Rating") +
  ggtitle("Rating real data") +
  tema

f
```


Rating real data
3000
2000
1000
0
Frequency
1
2
3
4
5
Rating