XYZ Health Services - Text Classification

Deeksha Aggarwal

January 20, 2018

CONTENTS

1. Problem Statement:

XYZ Health Services is a top ranked Health care provider in USA with stellar credentials and provides high quality-care with focus on end-to-end Health care services. The Heath Care Services range from basic medical diagnostics to critical emergency services. The provider follows a ticketing system for all the telephonic calls received across all the departments. Calls to the provider can be for New Appointment, Cancellation, Lab Queries, Medical Refills, Insurance Related, General Doctor Advise etc. The Tickets have the details of Summary of the call and description of the calls written by various staff members with no standard text guidelines.

The challenge is, based on the Text in the Summary and Description of the call, the ticket is to be classified to Appropriate Category (out of 5 Categories) and Subcategories (Out of 20 Sub Categories).

Problem Category : Text Classification

2. Data used:

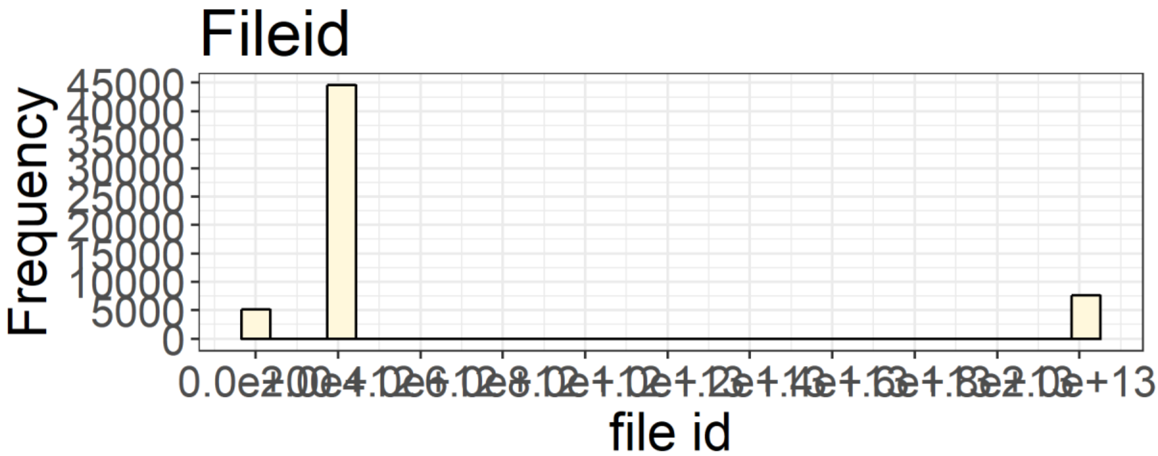
The Given dataset contains **57280 Observations and 7 Variables**
- fileid, summary, data, previous appointment, categories, sub categories and ID. SUMMARY and DATA are two very important variables which are unstructured form. And in our target variables - categories and subcategories, and previous appoinment variable there is noice which is supposed to be removed.

Variables, Data type and Data point

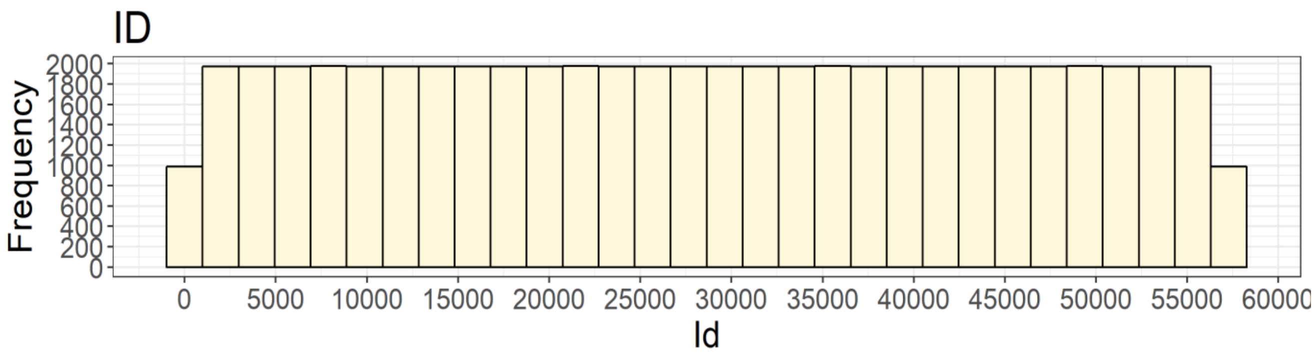
```
## [1] 57280          7
## Classes 'data.table' and 'data.frame':     57280 obs. of     7 variables:
  ##  $ fileid                 :integer64 2015561331001 2015561341001 2015561351001 2015561361001 20155613
  ##  $ SUMMARY                : chr     "Pt aware that he needs ROV for refill" "Mom wants to know if the Foca
  ##  $ DATA                   : chr     "{\\rtf1\\ansi\\ftnbj{\\fonttbl{\\f0 \\fswiss Arial;}}{\\colortbl ;\\r
  ##  $ categories             : chr     "PRESCRIPTION" "ASK_A_DOCTOR" "ASK_A_DOCTOR" "MISCELLANEOUS"
  ##  $ sub_categories         : chr     "REFILL" "MEDICATION RELATED" "MEDICATION RELATED" "OTHERS" ...
3.   $ previous_appointment: chr  "No" "No" "No" "No" ...
##   $ ID                     : chr     "2015_5_6133_1001" "2015_5_6134_1001" "2015_5_6135_1001"
"2015_5_6136_
4.   - attr(*, ".internal.selfref")=<externalptr>
```

3. **Exploration of Numerical Variable**: Out of 7 variables, 2 were numerical variable and 5 were categorical or character variable(Summary, Data, Categories, Sub-categories, Previous Appointments), out of which two are dependent variables(Categories and Sub_categories).

   1. Checking the distribution of the numerical variables:
      Histogram was plotted to check the distribution of the variables.

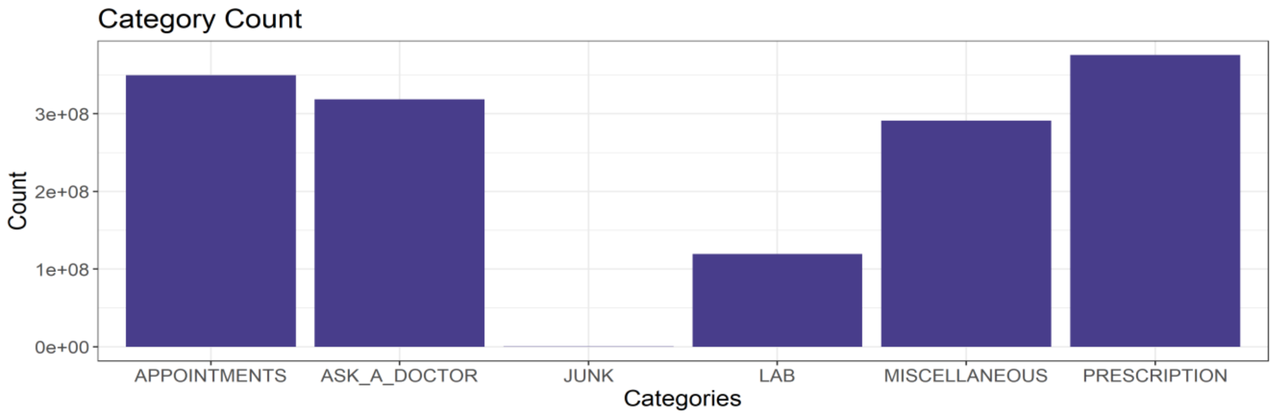


FileId Distribution

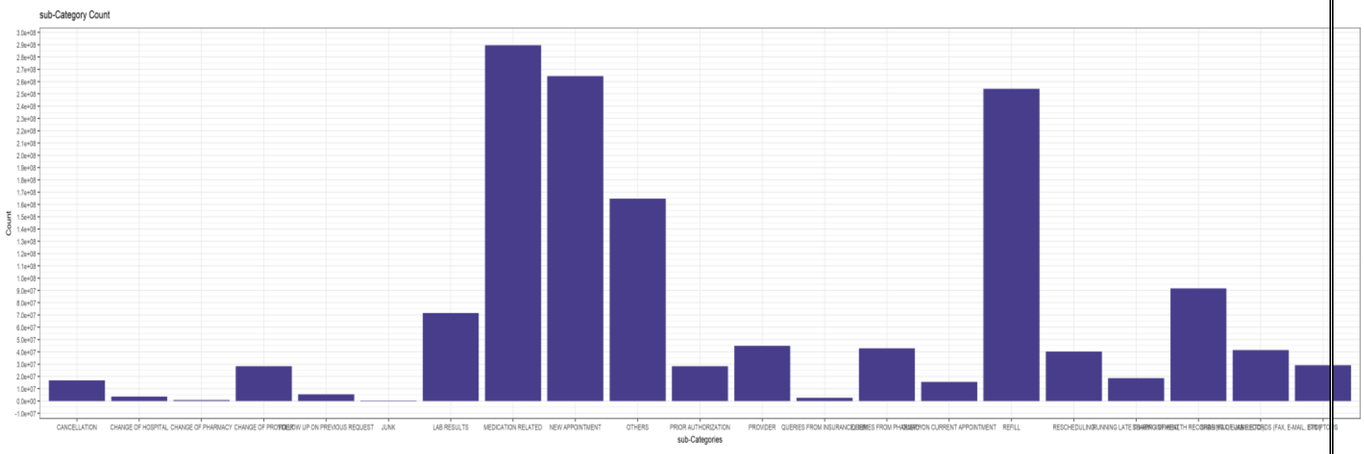


Id Distribution

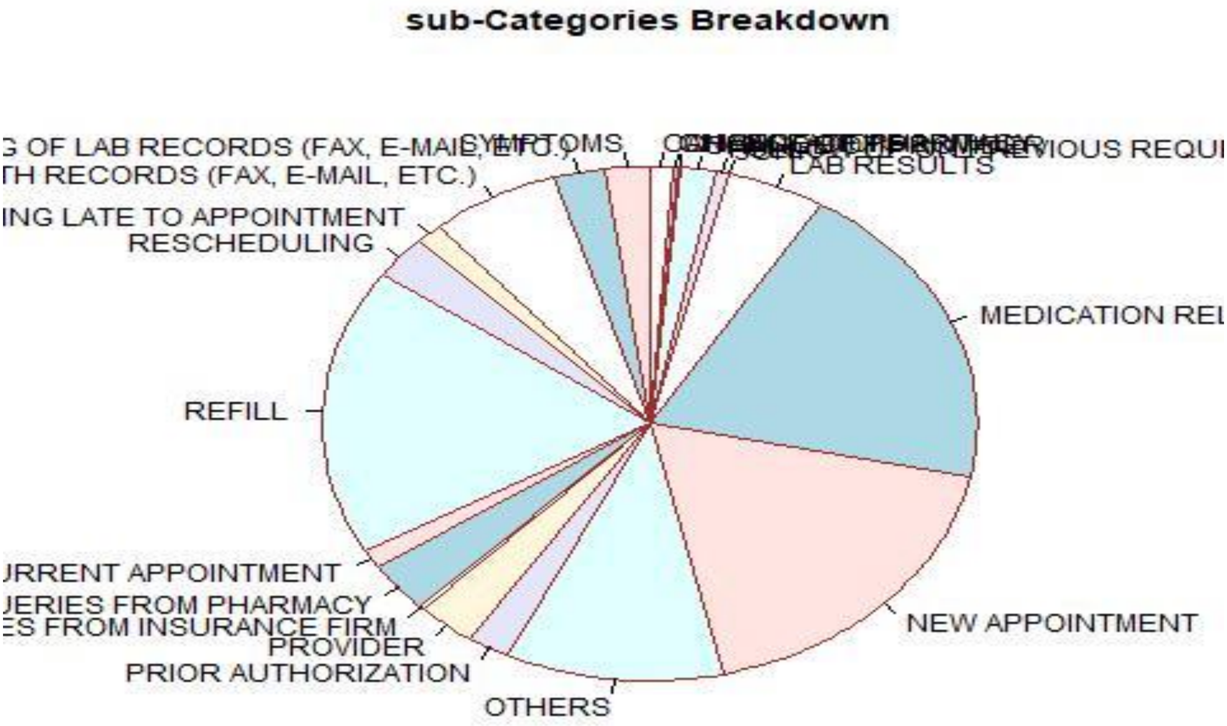
2. Checking the distribution of the categorical variables:
   Bar plot and pie plot was plotted to check the distribution of the categorical variables.



Category Distribution



Sub- Category Distribution



Sub Category Distribution

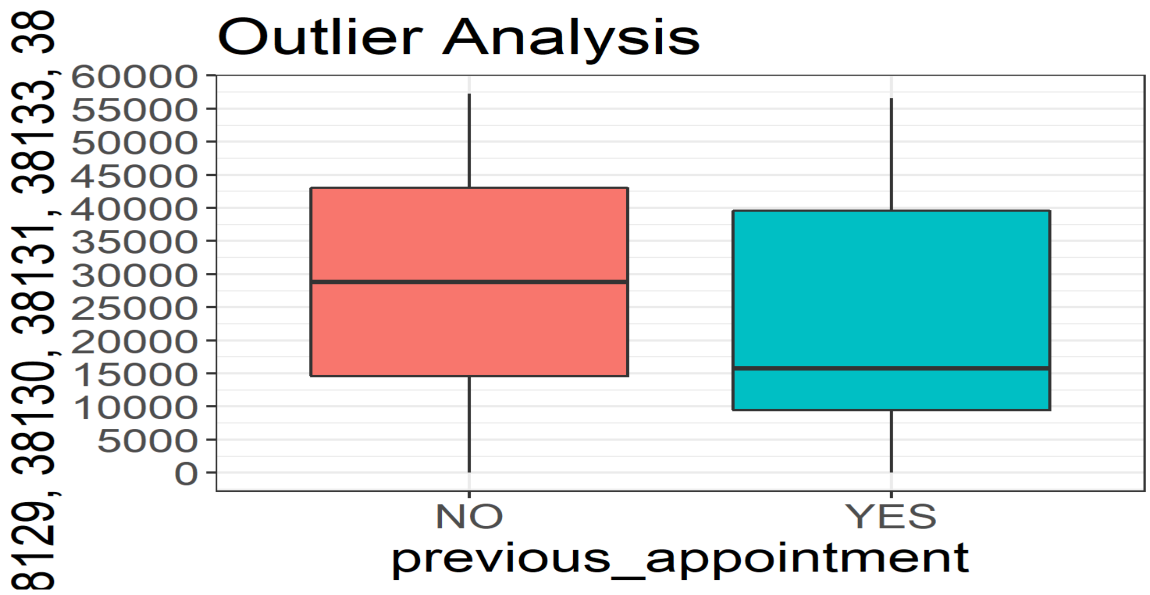
1. **Checking for missing value**: There was 3347 missing values in Summary variable and 2 in previous appointment variable.

2**. Missing Value Imputaion**:

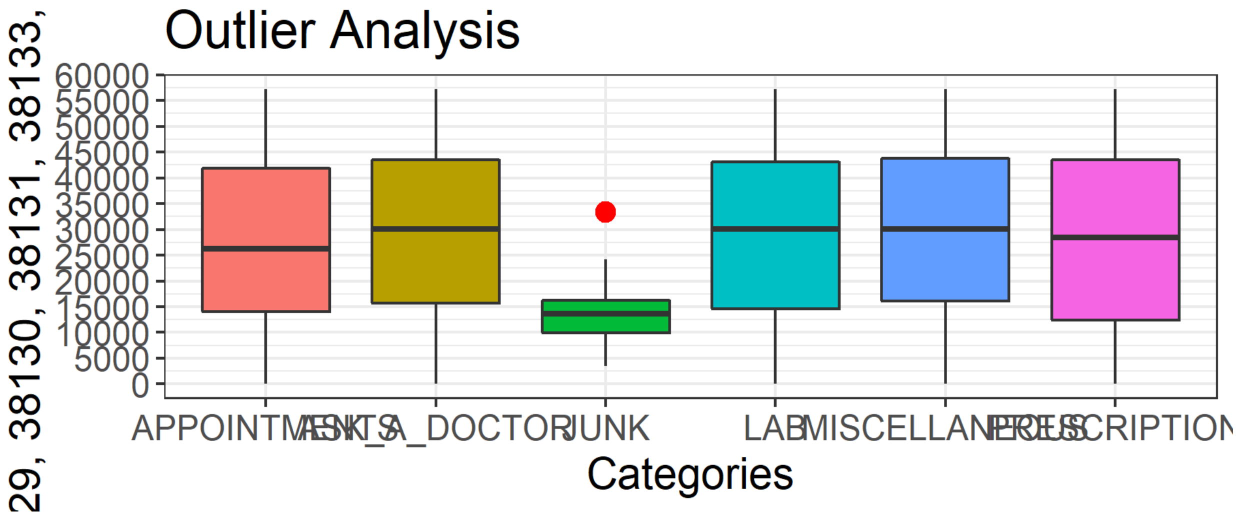
<u>For Summary variable</u>- In the later steps after data cleaning, summary and data variables which are both textual data are combined together. As there are no missing values present in "data" variable hence all 3347 m.v are removed.

<u>For Previous appointment</u>- Dropping 2 missing values will not create a problem with the dataset of 58 thousands observations.

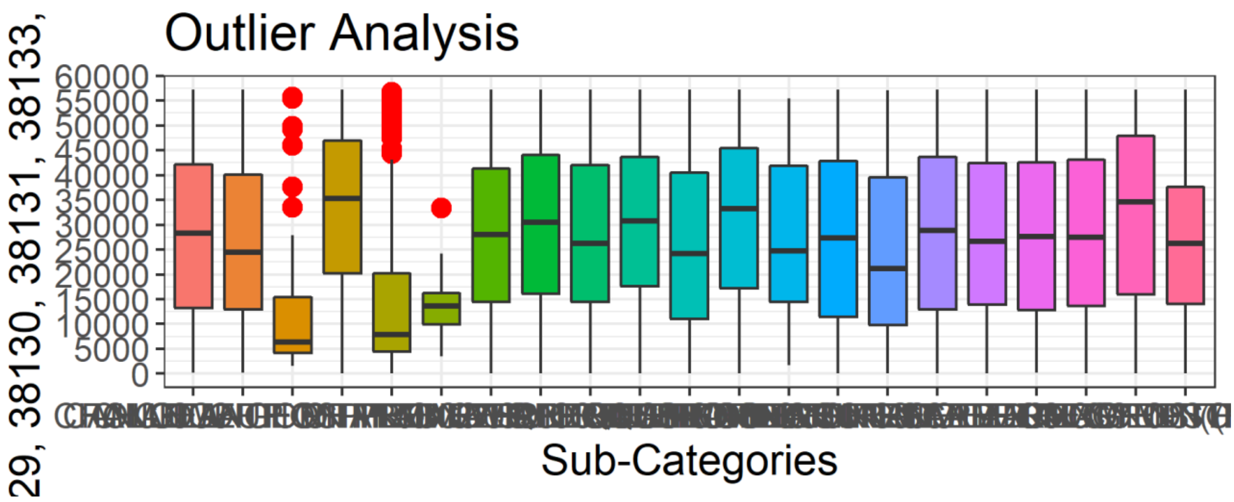
3. Checking for outliers: Boxplot was used to check the presence of outliers.



As we can see there are no outliers in previous appointment variable.



As we can see 1 outlier is present which can be dropped.



All the outliers are dropped from the data.

4. Exploration of Categorical Variable.

1 Table of Categories

```
APPOINTMENTS  ASK_A_DOCTOR          JUNK           LAB MISCELLANEOUS
PRESCRIPTION
        12960          11744          20           4246          10462
14500
```

2. Table of sub-Categories

```
CANCELLATION                                        CHANGE OF HOSPITAL
                                                    613
142
                                    CHANGE OF PHARMACY
CHANGE OF PROVIDER
                                                    54
928
                     FOLLOW UP ON PREVIOUS REQUEST
JUNK
                                                    350
20
                                    LAB RESULTS
MEDICATION RELATED
                                                    2628
10547
                                    NEW APPOINTMENT
OTHERS
                                                    9713
5796
                                    PRIOR AUTHORIZATION
PROVIDER
                                                    1155
1608
                       QUERIES FROM INSURANCE FIRM
QUERIES FROM PHARMACY
```

107
1722
QUERY ON CURRENT APPOINTMENT
REFILL
652
9665
RESCHEDULING RUNNING
LATE TO APPOINTMENT
1520
694
SHARING OF HEALTH RECORDS (FAX, E-MAIL, ETC.) SHARING OF LAB RECORDS
(FAX, E-MAIL, ETC.)
3435
1386
SYMPTOMS
1197

3. Table of Previous Appointments

NO	YES
53739	193

7. **Cleaning and Pre-processing the data**

7.1 String Manipulation for "Summary" and "Data" variables

Data and summary variables contain textual data which is very dirty.

```
> data1$SUMMARY[1:10]
 [1] "pt aware that he needs rov for refill"
 [2] "mom wants to know if the focalin needs some dosage adjusting"
 [3] "pt called to discuss nortryptiline. she says she has a weird tas"
 [4] "fyi nortryptline medication."
 [5] "letter of patient establishment request"
 [6] "appt question"
 [7] "dizzy & double vision past 45 mins after ct"
 [8] " please refax neurocog order to new fac; wake med maybe"
 [9] "pt wants to reschedule epidural from 04/30, please call"
[10] "phone note"
```

Insights:-

- Words are written in short forms which need to be converted to their original form by using the gsub function. Below function is used to achieve the cleaned data.

```
regular_summary=function(r){
 r=gsub("patient","pt ",r,perl=T)
 r=gsub("sch[\\s,ed,d]","schedule ",r,perl=T)
 r=gsub("cl[l,ld,ed,d]+\\s","called ",r,perl=T)
 r=gsub("cal[l,ld,ed,d]+\\s","called ",r,perl=T)
 r=gsub("ap[p,t,pt]\\s","appointment ",r,perl=T)
 #r=gsub("\\scal"," called ",r,perl=T)
 #r=gsub("cal\\w+","called ",r,perl=T)
 r=gsub("med[s,\\s]+\\s","medicine ",r,perl=T)
 r=gsub("m[d,s]+\\s"," medicine ",r,perl=T)
 #r=gsub("\\ssz"," seizure ",r,perl=T)
 r=gsub("sz\\s","seizure ",r,perl=T)
 r=gsub("sz","seizure ",r,perl=T)
 #r=gsub("\\sre"," regarding ",r,perl=T)
 #r=gsub("re\\s","regarding ",r,perl=T)
 #r=gsub("\\schk"," check ",r,perl=T)
 r=gsub("chk\\s","check ",r,perl=T)
 r=gsub("spk\\s","speak ",r,perl=T)
 r=gsub("abt\\s","about ",r,perl=T)
 r=gsub("dr[.,\\s]+\\s","doctor ",r,perl=T)
 #r=gsub("wt\\s","weight ",r,perl=T)
 r=gsub("req\\s","request ",r,perl=T)
 r=gsub("pl[s,z]+\\s","please ",r,perl=T)
 #r=gsub("pl\\s","please ",r,perl=T)
 r=gsub("s/w\\s","speak with ",r)
 r=gsub("confirm[//s,//w+]","confirm",r)
 r="discussion "
 r=gsub("discuss//w+//s","discuss",r,perl=T)
 r
}
```

```
> data1$DATA[1]
[1] "{\\rtf1\\ansi\\ftnbj{\\fonttbl{\\f0 \\fswiss arial;}}{\\colortbl
;\\red255\\green255\\blue255 ;\\red0\\green0\\blue255 ;\\red0\\green0\\blue0
;\\red0\\green0\\blue255 ;\\red0\\green128\\blue0
;}{\\stylesheet{\\f0\\fs20\\cf3\\cb1 normal;}{\\cs1\\additive\\cf3\\cb1
```

```
default paragraph
font;}}\\margl1440\\margr1440\\margt540\\margb1440\\headery540\\footery720\\fo
rmshade\\sectd\\marglsxn1440\\margrsxn1440\\margtsxn540\\margbsxn1440\\headery
540\\footery720\\sbkpage\\pgncont\\plain\\plain\\fs20\\pard\\plain\\fs20\\cf0\
\fs24\\sscharaux1\\b phone note \\fs20\\b0\\par\\b\\par call patient back
at:\\par\\b0 cell phonexxxx-xxxx\\par\\fs24\\b\\par call from
patient\\par\\fs20 caller name: \\b0xxxx-xxxx caller: \\b0 patient\\par\\b
call for: \\b0 nurse\\par\\par\\fs24\\b other \\fs20\\b0\\par patient is
returning nurse call. he is unable to make appt without talking to fin service
dept. however he needs medication and worried that he will have issue without
medication. please call patient to discuss. \\par\\b call taken by: \\b0 xxxx-
xxxx  may 26, 2015 5:09 pm\\par\\par\\fs24\\b call back\\fs20\\b0\\par\\b
follow-up details: \\b0 pt returned phone call. please call back to advise
@xxxx-xxxx  may 27, 2015 8:46 am\\par\\b additional follow-up details: \\b0
what is the problem?  is he without insurance?  he has been non-compliant with
instructions to come in for a follow-up appt. and cannot have refills without
one.\\par\\b additional follow-up by: \\b0 david xxxx-xxxx  may 27, 2015 8:54
am\\par\\b additional follow-up details: \\b0 rn spoke with pt and relayed the
above to him.  he requested to speak with financial services.  rn transferred
him to the business office.  rn requested business office to call once matter
has been completed.\\par\\b follow-up by: \\b0 hollie saltis rn ,  may 27,
2015 11:54 am\\par\\b additional follow-up details: \\b0 ok.\\par\\b
additional follow-up by: \\b0 david xxxx-xxxx  may 27, 2015 5:03
pm\\par\\plain\\fs20\\par\\cf3\\par\\plain\\fs20\\par}"
```

Insights:-

- Text is very messy which need to be cleaned by using the gsub function. Below function is used to achieve the cleaned data.

```
regular1=function(sr){
 sr=gsub("\\\\[^\\s]+\\s"," ",sr,perl=T)
 sr=gsub("x"," ",sr,perl=T)
 sr=gsub("}"," ",sr,perl=T)
 sr=gsub("{"," ",sr,perl=T)
 sr=gsub("-"," ",sr,perl=T)
 sr=gsub(";"," ",sr,perl=T)
 sr=gsub("ap[p,t,pt]+\\s","appointment ",sr,perl=T)
 sr=gsub("patient","pt ",sr,perl=T)
 sr=gsub("sch[\\s,ed,d]+\\s","schedule ",sr,perl=T)
 sr=gsub("cl[l,ld,ed,d]+\\s","called ",sr,perl=T)
 sr=gsub("cal[l,ld,ed,d]+\\s","called ",sr,perl=T)

 #r=gsub("\\scal"," called ",r,perl=T)
 #r=gsub("cal\\w+","called ",r,perl=T)
 sr=gsub("med[s,\\s,s,]+\\s","medicine ",sr,perl=T)
 sr=gsub("m[d,s]+\\s"," medicine ",sr,perl=T)
 #r=gsub("\\ssz"," seizure ",r,perl=T)
 sr=gsub("sz\\s","seizure ",sr,perl=T)
 sr=gsub("sz","seizure ",sr,perl=T)
 #r=gsub("\\sre"," regarding ",r,perl=T)
 #r=gsub("re\\s","regarding ",r,perl=T)
 #r=gsub("\\schk"," check ",r,perl=T)
 sr=gsub("chk\\s","check ",sr,perl=T)
 sr=gsub("spk\\s","speak ",sr,perl=T)
 sr=gsub("abt\\s","about ",sr,perl=T)
 sr=gsub("dr[.,\\s]+\\s","doctor ",sr,perl=T)
 #r=gsub("wt\\s","weight ",r,perl=T)
 sr=gsub("req\\s","request ",sr,perl=T)
 sr=gsub("pl[s,z]+\\s","please ",sr,perl=T)
 #r=gsub("pl\\s","please ",r,perl=T)
 sr=gsub("s/w\\s","speak with ",sr)
```

```
 sr=gsub("paragraph"," ",sr,perl=T)
 sr=gsub("/"," ",sr,perl=T)
 sr=gsub("\\\\"," ",sr,perl=T)
 sr=gsub("sscharaux"," ",sr,perl=T)
 sr=gsub("par"," ",sr,perl=T)
 sr=gsub("protect"," ",sr,perl=T)
 sr=gsub("wrote"," ",sr,perl=T)
 sr=gsub(":"," ",sr,perl=T)
 sr=gsub("[0-9]"," ",sr,perl=T)
 sr=gsub("pm"," ",sr,perl=T)
 sr=gsub("am"," ",sr,perl=T)
 sr=gsub("arial"," ",sr,perl=T)
 sr=gsub("font"," ",sr,perl=T)
 sr=gsub("normal"," ",sr,perl=T)
 sr=gsub("default"," ",sr,perl=T)
 sr=gsub("plain"," ",sr,perl=T)
 sr=gsub("home"," ",sr,perl=T)
 sr=gsub("phone"," ",sr,perl=T)
 sr=gsub("cell"," ",sr,perl=T)
 sr=gsub("called"," ",sr,perl=T)
 sr=gsub("note"," ",sr,perl=T)
 sr=gsub("fs"," ",sr,perl=T)
 sr=gsub("cf"," ",sr,perl=T)
 sr=gsub("\\s+", " ",sr,perl = T)
 sr=gsub("converted from flag"," ",sr,perl=T)
 sr=gsub("call[\\w+,\\s] taken by"," ",sr,perl=T)
 sr=gsub("phone note called pt back"," ",sr,perl=T)
 sr=gsub("phone note called"," ",sr,perl=T)
 sr=gsub("follow up details"," ",sr,perl=T)
 sr=gsub("\\s+", " ",sr,perl = T)
}
```

After applying above functions the data and summary variables are clubbed together to form a single variable called “text”. Below is the snap of cleaned text variable.

pt aware that he needs rov for refill. pt back at from pt caller n e caller pt for nurse other pt is returning nurse call. he is unable to make appointment without talking to fin service dept. however he needs medication and worried that he will have issue without medication. please pt to discuss. taken by may , back pt returned call. please back to advise @ may , additional what is the problem? is he without insurance? he has been non compliant with instructions to come in for a follow up appt. and cannot have refills without one. additional follow up by david may , additional rn spoke with pt and relayed the above to him. he requested to speak with financial services. rn transferred him to the business office. rn requested business office to once matter has been completed. follow up by hollie saltis rn , may , additional ok. additional follow up by david may ,

7.2 Term document matrix formation

Below code and functions are used to do preprocessing and cleaning of the corpus.It takes in a vector with all the text in it and convert it to a corpus and cleans it. Sparse terms are removed by using the function removeSparseTerms(tdm, 0.99). and finally the term document matrix is formed which is converted in to data frame and all other variables are pasted with this data frame.

```
library(stringr)
library(tm)
data1$text=as.character(data1$text)
data_corpus = VCorpus(VectorSource(data1$text)) #forming a corpus
data_corpus=tm_map(data_corpus, tolower)
data_corpus = tm_map(data_corpus,content_transformer(tolower))
data_corpus = tm_map(data_corpus,removePunctuation)  #removing punchuations
data_corpus = tm_map(data_corpus,removeNumbers) #removing numbers
#data_corpus = tm_map(data_corpus,tolower) #converting to lowercase
data_corpus =
tm_map(data_corpus,removeWords,c('pt','able','its','it','us','use','used','using','will','yes','say','can','take','one',
                                 'adderall','adol','ago','answere','action','address','afternoon','anda','anything',
                                 'ada','alt', 'arround','asked', 'area', 'april', 'assoc', 'atp', 'call','caller','cma',
                                 'comments', 'cook', 'copy', 'count', 'cover', 'daughter', 'dhe', 'harry', 'mom',
                                 'denise', 'done', 'holly', 'hollie', 'duke', 'first', 'iii', 'ily', 'ing', 'inj',
                                 'keep', 'let', 'like', 'linda', 'jones', 'mary', 'mother', 'pdf', 'wife', 'way', 'monday',
                                 'monday', 'tuesday', 'wednesday', 'thursday', 'friday','saturday','sunday',
                                 'yesterday', 'today', 'tommorow','january', 'february',
'may','march','june','july','august',
                                 'september','october','november','december','wendy', stopwords('english')))
#removing english stopwords
data_corpus = tm_map(data_corpus,stripWhitespace) #removing the whitespaces
#data_corpus = tm_map(data_corpus,removeWords,stopwords("SMART"))
corpus_copy=data_corpus
data_corpus=tm_map(data_corpus, gsub, pattern = "appointments", replacement = "appointment")
data_corpus=tm_map(data_corpus, gsub, pattern = "appt", replacement = "appointment")
  #data_corpus=tm_map(data_corpus, gsub, pattern = "", replacement = "appointment")

data_corpus = tm_map(data_corpus, PlainTextDocument)
tdm = t(TermDocumentMatrix(data_corpus))
#tdm2 <- removeSparseTerms(tdm, 0.98)
td3=removeSparseTerms(tdm, 0.99)


tdm_data_565=data.frame(as.matrix(td3))
final_data2=as.data.frame(cbind(data1$sub_categories,data1$categories,data1$previous_appointment,tdm_data
_565))
#names(final_data)
colnames(final_data)[1]="sub_categories"
colnames(final_data)[2]="categories"
colnames(final_data)[3]="previous_appointment"
```

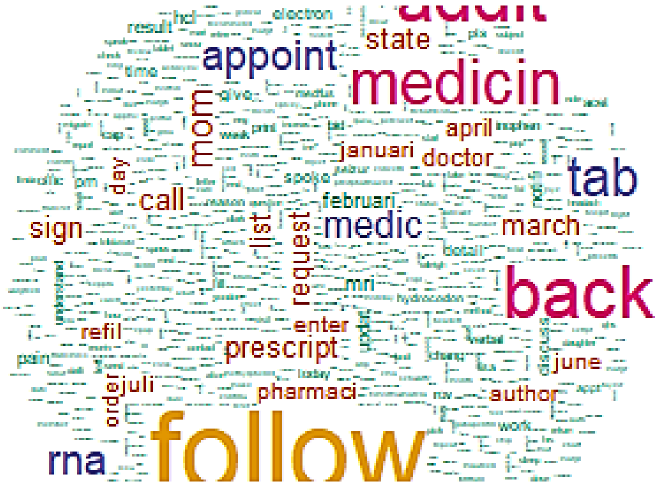
```
<<Loading required package: NLP
```

```
> tdm = t(TermDocumentMatrix(data_corpus))
> tdm
<<DocumentTermMatrix (documents: 53932, terms: 39690)>>
Non-/sparse entries: 1861411/2138699669
Sparsity           : 100%
Maximal term length: 54
Weighting          : term frequency (tf)
```

After Applying function removeSparseTerms(tdm, 0.99):-

```
> td3=removeSparseTerms(tdm, 0.99)
> td3
<<DocumentTermMatrix (documents: 53932, terms: 600)>>
Non-/sparse entries: 1354976/29116604
Sparsity           : 96%
Maximal term length: 15
Weighting          : term frequency (tf)
```

7.4 Exploratory Data Analysis through Visualizations



	Terms	frequency
## 1:	Follow	11399
## 2:	Medicin	4499
## 3:	back	4041
## 4:	tab	3967
## 5:	appoint	3605

We can clearly see ‘follow’, ‘medicin’, ‘back’, ‘tab’ and appoint’ are top 5 most frequent words in the Corpus for whole data set.

Insights of data set:

- Words like ‘‘follow’, ‘medicin’, ‘back’, ‘tab’ and appoint’ are most common frequent words in all kinds.
- There is a certain amount of noise the both the categories and subcategories, It might be due the improper data entry operations. This noice is removed using **plyr** package.
- There are many common terms between categories which may not be useful to the model. We have to figure out whether to remove most common terms or all common terms.
- Logically, if we build model for categories, it should classify sub-categories as weel. But these is an ideal condition. We have to see how this works practically.

7.5 Feature Engineering

Removing Corelated Terms

Checking correlation between the predictors is a must in Analysis.I have used tm's findAssocs and pearson's correlation matrix to detect correlation and association.

A correlation matrix is built representing positive and negative correlations between terms. I have taken 85% as correlation limit and filtered out highly correlated terms from our structured data. Words like "marcia

5. richardson", "brown , lori", "linda , clark", "tisha , walker", "cook , denni" are highly correlated pairs. A term is taken from each correlation pair and made a vector called corr.terms.

- [1] "Number of correlated terms which are to be filtered are:"
- [1] 35

We have to remove these corelated terms from the variable of our combined_data set.Thus forming a data set without highly (>85%) corelated terms.

- [1] "Some of common terms between the categories are:"
-

```
         Terms frequency
## 1:       call     11399
## 2:    patient      4499
## 3: paragraph      4319
## 4:       back      4041
## 5:       note      3967
## 6:   fontphon      3605
```

corr.terms are combined with common_unique words and together removed from the data.

- [1] "Dimensions of data after removing corelated terms and top most common terms are:"
- [1] 53280 565

Now that we have removed the most common terms and correlated terms (%85), we have our final features to predict the target variables. Our final no of variable are 858. We are going to form two master data sets each for predicting categories and sub-categories.

8. Sampling .

We are using stratified sampling to divide the data into training and test data in the ratio 90:10. This preserve the ratio of class in each variables throughout sampling and splitting. caTools package is used to implement stratified sampling**.**

```
#stratified sampling
set.seed(123)
library(caTools)
smpl = sample.split(data[,ncol(data)], 0.9)
train = subset(data, smpl == T )
test = subset(data, smpl == F )
```

- [1] "Dimensions of train and test sets are
- [1] 48539 565
- [1] 5393 565

9. Building Predictive Models

9.1 GBM Model

Among all the other models tested, GBM model gives the highest accuracy of 79.89% to predict the categories and 71.26% to predict the sub-categories with ntree=1000. The greatest drawback is that it took 7 hours to be trained on the dataset of 48539 observations and 561 variables. I have used h2o package to build the model as it is really quick in training a model using h2o.

Below is the performance of the model with MSE value of 0.168712 and RMSE value of 0.4107456.

```
> h2o.performance(gbm.model)
H2OMultinomialMetrics: gbm
** Reported on training data. **

Training Set Metrics:
=====================

Extract training frame with `h2o.getFrame("train")`
MSE: (Extract with `h2o.mse`) 0.168712
RMSE: (Extract with `h2o.rmse`) 0.4107456
Logloss: (Extract with `h2o.logloss`) 0.5111259
Mean Per-Class Error: 0.2202054
```

Below is the confusion matrix of the model with accuracy of 79.89% to predict categories.

```
> h2o.confusionMatrix(predict.gbm[,1], test.h2o[,1])
Confusion Matrix and Statistics

               Reference
```

```
Prediction       APPOINTMENTS ASK_A_DOCTOR JUNK  LAB MISCELLANEOUS
PRESCRIPTION
  APPOINTMENTS           1107           62    0   23           100
31
  ASK_A_DOCTOR             67          871    0   14           101
100
  JUNK                      0            0    0    0             1
0
  LAB                      17           19    0  324            23
12
  MISCELLANEOUS            87          105    1   51           727
88
  PRESCRIPTION             19          109    0   19            64
1251

Overall Statistics

               Accuracy : 0.7936
                 95% CI : (0.7826, 0.8044)
    No Information Rate : 0.2748
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.7345
 Mcnemar's Test P-Value : NA

Statistics by Class:

                     Class: APPOINTMENTS Class: ASK_A_DOCTOR Class: JUNK
Class: LAB Class: MISCELLANEOUS
Sensitivity                       0.8535              0.7470   0.0000000
0.75174              0.7156
Specificity                       0.9473              0.9333   0.9998145
0.98569              0.9241
Pos Pred Value                    0.8367              0.7554   0.0000000
0.82025              0.6865
Neg Pred Value                    0.9533              0.9304   0.9998145
0.97859              0.9333
Prevalence                        0.2405              0.2162   0.0001854
0.07992              0.1884
Detection Rate                    0.2053              0.1615   0.0000000
0.06008              0.1348
Detection Prevalence              0.2453              0.2138   0.0001854
0.07324              0.1964
Balanced Accuracy                 0.9004              0.8401   0.4999073
0.86872              0.8199
                     Class: PRESCRIPTION
Sensitivity                       0.8441
Specificity                       0.9460
Pos Pred Value                    0.8557
Neg Pred Value                    0.9412
Prevalence                        0.2748
Detection Rate                    0.2320
Detection Prevalence              0.2711
Balanced Accuracy                 0.8951
```

Below is the confusion matrix of the model with accuracy of 71.65% to predict sub-categories.

```
>confusionMatrix(as.matrix(predict.gbm$predict),as.matrix(test.h2o$sub_cat
egories))
```

```
Confusion Matrix and Statistics

Overall Statistics

               Accuracy : 0.7165
                 95% CI : (0.7042, 0.7285)
    No Information Rate : 0.1943
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.6722
 Mcnemar's Test P-Value : NA
```

9.2 Random Forest Model

Random Forest model gives the accuracy of 60.89% to predict the categories and 48.58% to predict the sub- categories with ntree=1000. It took 30 minutes to be trained on the dataset of 48539 observations and 561 variables. I have used h2o package to build the model as it is really quick in training a model using h2o.

Below is the performance of the model with MSE value of 0.5342424and RMSE value of 0.7309189.

```
> h2o.performance(rforest.model)
H2OMultinomialMetrics: drf
** Reported on training data. **
** Metrics reported on Out-Of-Bag training samples **

Training Set Metrics:
=====================

Extract training frame with `h2o.getFrame("train")`
MSE: (Extract with `h2o.mse`) 0.5342424
RMSE: (Extract with `h2o.rmse`) 0.7309189
Logloss: (Extract with `h2o.logloss`) 1.352635
Mean Per-Class Error: 0.6041799
```

Below is the confusion matrix of the model with accuracy of 60.86% to predict categories.

```
> h2o.confusionMatrix(predict.rf[,1], test.h2o[,1])
Confusion Matrix and Statistics

                APPOINTMENTS ASK_A_DOCTOR MISCELLANEOUS PRESCRIPTION
  APPOINTMENTS          1193           65             4          100
  ASK_A_DOCTOR           121          600            23          410
  JUNK                     0            0             0            1
  LAB                     86           70            52          201
  MISCELLANEOUS          233           74           189          524
  PRESCRIPTION            74           57            15         1301

Overall Statistics

               Accuracy : 0.6086
                 95% CI : (0.6331, 0.6588)
```

```
    No Information Rate : 0.2598
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.5502
 Mcnemar's Test P-Value : NA
```

Below is the confusion matrix of the model with accuracy of 48.58% to predict sub-categories.

```
>
confusionMatrix(as.matrix(predict.rforest$predict),as.matrix(test.h2o$sub_
categories))
Confusion Matrix and Statistics
Overall Statistics

               Accuracy : 0.4858
                 95% CI : (0.4724, 0.4993)
    No Information Rate : 0.1943
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.3689
 Mcnemar's Test P-Value : NA
```

10 Error Metrics and Recommendations

Error Metrics

As there might be situations when a patient calling for emergency situations might be misclassified as the one with least priority value and a situation where the patient is calling for general advices is put on the top of the priority order for immediate attention from the Doctor. So, we consider both False Positive and False Negative and take them as whole as Misclassification error and minimize it.

We are aiming to freeze the model which is giving least misclassification error with the minimal computatinal power and time. So, the model and seed with highest accuracy is considered to be our model of Deployment.

So, **GBM** is freezed as our deployment model with an accuracy of 79.86% for classifying categories.

Recommendations

If we analyze the Document Term Matrices generated out of the SUMMARY and DATA column along with sub categories, there are many common terms between those 20 sub categories. This makes the problem more complex because the common terms misguide the models.

If we try to remove the common terms between the sub categories, we are left with very few terms as we have only 57280 rows, which won't contribute much to the classification. So, I would suggest designing a model separately for classifying sub-categories with much more data. More data always yields the better results though it is complicated to clean it.

THANK YOU