XYZ Health Services - Text Classification

Manish Reddy Jannepally

November 24, 2017

INTRODUCTION TO THE XYZ HEALTH SERVICES - TEXT CLASSIFICATION PROJECT

Problem Statement

To build a classification model based on the Text in the Summary and Description of the call to classify the ticket to appropriate Category (out of 5 Categories) and Subcategories (Out of 20 Sub Categories).

Case Study Explanation and Domain Knowledge

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).

Pain and Gain Analysis

The Pain and Gain analysis of this project needs very subtlety in Perception and Understanding of the communication usually varies from Tone, Body language, Vocabulary and Absurdity levels while communicating.In real world scenarios, a person needs to understand the subtlety of this usage, requires second-order interpretation of the speaker's or writer's intentions; different parts of the brain must work together to understand purpose.

Using Analytics to identify the purpose of the call will be beneficial. This approach will reduce Time Consumption of text classification. Applications are such as analyzing healthcare calls helps reduce the time to classify and escalate to concerned team.

Examples: Optum Labs, an US research collaborative, has collected EHRs of over 30 million patients to create a database for predictive analytics tools that will improve the delivery of care.

Cleaning and Processing the data

The Given dataset contains **57280 Observations and 7 Variables** - fileid, summary, data, previous appointment, categories, sub categories and ID. The variables fileid and ID will be unique for every ticket, So they are of no use to our model. 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.

Given data for classification

let's look at our given data dimensions and structure.

```
## [1] 57280     7 #Dimensions of given data
```

##Structure of given data

```
## Classes 'data.table' and 'data.frame':   57280 obs. of  7 variables:
##  $ fileid              :integer64 2015561331001 2015561341001 201556135100
1 2015561361001 2015561371001 2015561401001 2015561411001 2015561421001 ...
##  $ SUMMARY             : chr  "Pt aware that he needs ROV for refill" "Mom
wants to know if the Focalin needs some dosage adjusting" "pt called to discu
ss nortryptiline. she says she has a weird tas" "FYI Nortryptline medication.
" ...
##  $ DATA                : chr  "{\\rtf1\\ansi\\ftnbj{\\fonttbl{\\f0 \\fswis
s Arial;}}{\\colortbl ;\\red255\\green255\\blue255 ;\\red0\\green0\\"| __trun
cated__ "{\\rtf1\\ansi\\ftnbj{\\fonttbl{\\f0 \\fswiss Arial;}}{\\colortbl ;\\
red255\\green255\\blue255 ;\\red0\\green0\\"| __truncated__ "xxxx-xxxx\\f0 \\
fswiss Arial;}}{\\colortbl ;\\red255\\green255\\blue255 ;\\red0\\green0\\blue
255 ;\\red0\\green"| __truncated__ "xxxx-xxxx\\f0 \\fswiss Arial;}}{\\colortb
l ;\\red255\\green255\\blue255 ;\\red0\\green0\\blue255 ;\\red0\\green"| __tr
uncated__ ...
##  $ categories          : chr  "PRESCRIPTION" "ASK_A_DOCTOR" "ASK_A_DOCTOR"
"MISCELLANEOUS" ...
##  $ sub_categories      : chr  "REFILL" "MEDICATION RELATED" "MEDICATION RE
LATED" "OTHERS" ...
##  $ previous_appointment: chr  "No" "No" "No" "No" ...
##  $ ID                  : chr  "2015_5_6133_1001" "2015_5_6134_1001" "2015_
5_6135_1001" "2015_5_6136_1001" ...
##  - attr(*, ".internal.selfref")=<externalptr>
```

From the above output, we can see that the DATA column is in RTF format which needs to be converted to text format. I used **qdap** library to perform this operation.

From the structure of the data, the columns categories, sub-categories, previous appointment are characters. I converted them to factor variables. Then, removed the noice

in those 3 columns using **plyr** package. SUMMARY and DATA colums are processed and converted to DTMs(Document Term Matrices) and further reduces the sparsity using removeSparseTerms function. Let's process the given data.

Given data –> Data required –> Processed data

I converted the RTF format to text format using **qdap** package. Let's see how the DATA column looks now.

```
## head(given_data[,3],n = 2)

## [1] "ArialNormalDefault Paragraph FontPhone Note Call patient back atCell
PhoneFROM PATIENTCaller Name Caller PatientCall For NurseOther Patient is ret
urning nurse call He is unable to make appt without talking to fin service de
pt However he needs medication and worried that he will have issue without me
dication Please call patient to discuss Call Taken by 26 2015 5PMCall backFol
lowDetails Pt returned phone call Please call back to advise May 27 2015 8AMA
dditional FollowDetails What is the problem Is he without insurance He has be
en nonwith instructions to come in for a followappt and cannot have refills w
ithout oneAdditional Followby David 27 2015 8AMAdditional FollowDetails RN sp
oke 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 offi
ce to call once matter has been completedFollowby Hollie Saltis RN May 27 201
5 11AMAdditional FollowDetails OkAdditional Followby David 27 2015 5PM"
## [2] "ArialNormalDefault Paragraph FontPhone Note Call patient back atCell
PhoneFROM PATIENTCaller Name Caller PatientOther Ptschool teacher is reportin
g pt is not able to sit still mom wants to know if the Focalin needs some dos
age adjusting andis there something pt could take that the school staff could
administer Please call back to discuss Call Taken by May 12 2015 1PMFollowDet
ails Mom sts patient is having alot of issues with meds and effectiveness LOV
1 3rescheduled to 4and that was a No Show Mom apologized for the no show She
sts they disconnected the home number and only use the cell Appt scheduled fo
r 5at 10 Advd to be here at 945Action Taken Phone Call Completed Appt Schedul
edFollowby Marcia Richardson LPN May 12 2015 4PM"
```

As I said, fileid and ID vairiables are not required, let's remove them and also remove noice in target variables. I have created a new data set data_req with the noise free target variables and without fileid and ID variables.

```
## [1] 57280     5 #Dimensions of data_req

##    SUMMARY              DATA                    categories
##  Length:57280       Length:57280       APPOINTMENTS :13872
##  Class :character   Class :character   ASK_A_DOCTOR :11800
##  Mode  :character   Mode  :character   MISCELLANEOUS:12191
##                                        LAB          : 4321
##                                        PRESCRIPTION :15096
                                          sub_categories
##  MEDICATION RELATED                           :10599
##  NEW APPOINTMENT                              :10478
```

```
##  REFILL                                          : 9819
##  OTHERS                                          : 7377
##  SHARING OF HEALTH RECORDS (FAX, E-MAIL, ETC.): 3550
##  LAB RESULTS                                     : 2650
##  (Other)                                         :12807


##  previous_appointment
##  0:57085
##  1:  195
##
```

Cleaning the text

Now, I form 2 corpuses for SUMMARY and DATA columns and clean the text. Below are the steps I performed:

Case Folding

The first preprocessing step is Case folding. Here, we are converting all the letters in the Corpus to lowercase using R's base function tolower.

Remove Numbers

In this step, we are freeing corpus from numbers. Here, we use tm's removeNumbers function

Removing Stop Words

This step is about eliminating words that doesn't make any meaning. Stopwords of English would be enough, but since the dataset contains several short words in the form of short forms which are of no meaning to use. I used stopwords("en") and stopwords("SMART")

removing Punchuation

We have use tm's removePunctuation function to remove all punctuation marks such as comma, full stop, parenthesis, various brackets etc.., from the corpus

Stemming

For grammatical reasons, document contains different inflectional forms like tense forms and derivational forms, we are performing stemming to reduce all those words to their root word. We are using tm's stemDocument function to do this. Stemming greatly help in reducing total number of terms and increase weighting

Stripping White Spaces

The above performed preprocessing steps left our corpus with many leading and trailing whitespaces within documents. We are cleaning all of them in one go using tm's stripWhitespace function. With this step our basic preprocessing is completed.

A user defined function **clean_corpus** is created to do preprocessing and cleaning of the corpus. This function takes in a vector with all the text in it and convert it to a corpus and cleans it

```
#function for cleaning the corpus
clean_corpus <- function(data){
        data_corpus = Corpus(VectorSource(data)) #forming a corpus
        data_corpus = tm_map(data_corpus,removePunctuation) #removing punchua
tions
        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,stopwords("English")) #r
emoving english stopwords
        data_corpus = tm_map(data_corpus,removeWords,stopwords("SMART"))
        data_corpus = tm_map(data_corpus,stemDocument) #performing stemming
        data_corpus = tm_map(data_corpus,stripWhitespace) #removing the white
spaces

}
```

After cleaning the DATA and SUMMARY columns using clean_corpus, form a Document Term Matrix for each corpus.

```
## <<DocumentTermMatrix (documents: 57280, terms: 9750)>>
## Non-/sparse entries: 160261/558319739
## Sparsity           : 100%
## Maximal term length: 36
## Weighting          : term frequency (tf)

## <<DocumentTermMatrix (documents: 57280, terms: 73656)>>
## Non-/sparse entries: 2077704/4216937976
## Sparsity           : 100%
## Maximal term length: 75
## Weighting          : term frequency (tf)
```

A Document Term Matrix (DTM) is created from the corpus. Term Frequency is considered as weighting to create Document term matrix to keep DTM simple. DATA DTM has 57280 documents and 73656 terms with 100% sparsity and SUMMARY DTM has 57280 documents and 9750 terms with 100% sparsity.

```
## Loading required package: NLP

## <<DocumentTermMatrix (documents: 57280, terms: 403)>>
## Non-/sparse entries: 124114/22959726
## Sparsity           : 99%
```

```
## Maximal term length: 12
## Weighting          : term frequency (tf)

## <<DocumentTermMatrix (documents: 57280, terms: 510)>>
## Non-/sparse entries: 1505072/27707728
## Sparsity           : 95%
## Maximal term length: 36
## Weighting          : term frequency (tf)
```

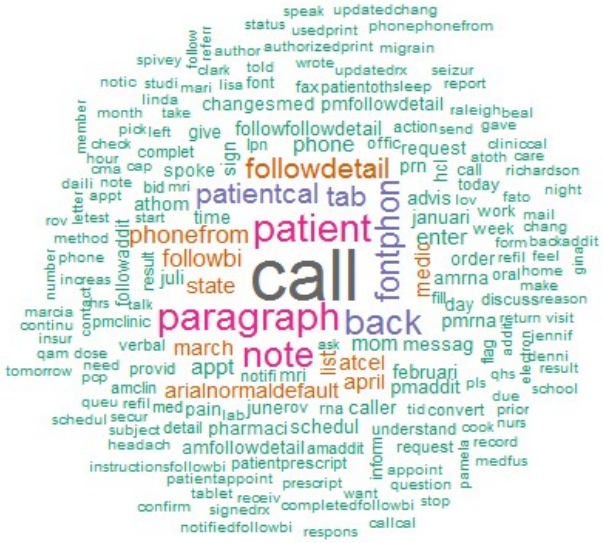
Sparsity is reduced and we made 73656+9750 terms to more relevant 510+403 terms. We try to get a balance between number of terms and vector size which R can allocate while processing. This Document term matrices is then converted to Data frame for Feature engineering.

```
## [1] 57280   403      #dimensions of data set formed by SUMMARY DTM

## [1] 57280   510      #dimensions of data set formed by DATA DTM

## [1] 57280   913      #dimension of data set by combining SUMMARY and DATA
```

Combined Data frame has 57280 observations and 913 features (excluding target class).

Exploratory Data Analysis

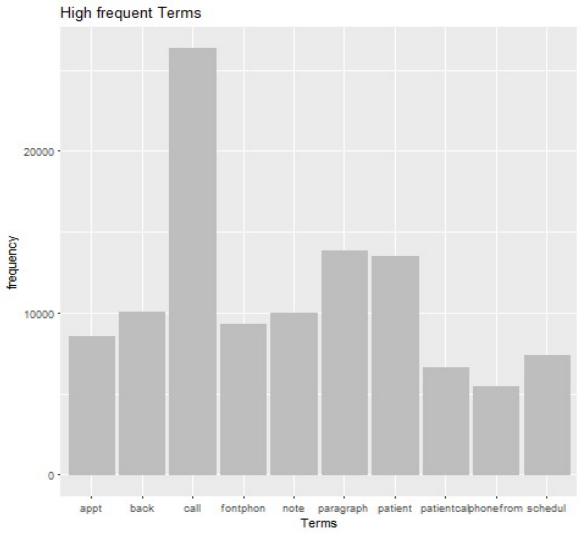
Let's see the whole data wordcloud.



Below are the most frequent terms in the corpus

```
## [1] "Most frequent terms are:"
```

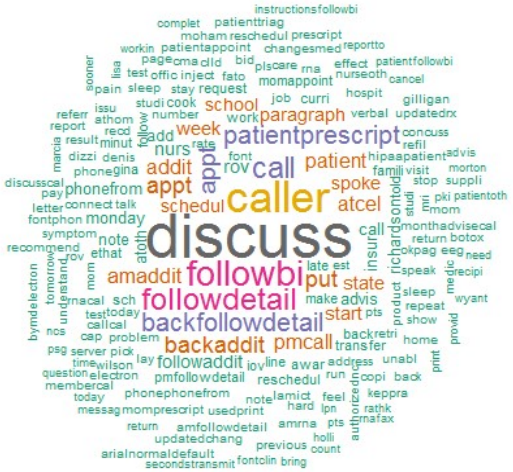
```
##         Terms frequency
## 1:       call    134005
## 2:    patient     58324
## 3: paragraph     57237
## 4:       note     50724
## 5:       back     50197
## 6:  fontphon     43730
```



We can clearly see ‘call’, ‘patient’, ‘paragraph’, ‘note’ and ‘back’ are top 5 most frequent words in the Corpus for whole data set. We further ensure this hypothesis from a bar plot.

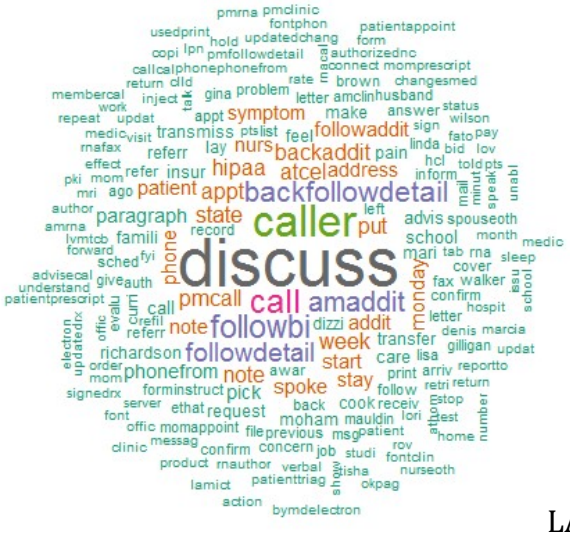
Let’s explore category wise worclouds to understand data more clearly.

APPOIMENTS CLOUD

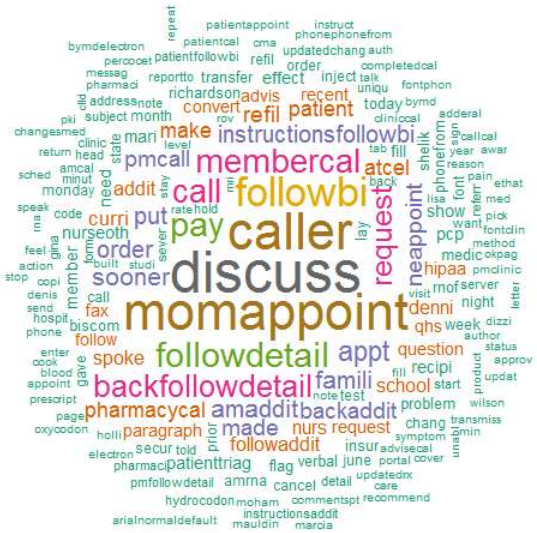


ASK A DOCTOR CLOUD

MISCELLANEOUS CLOUD

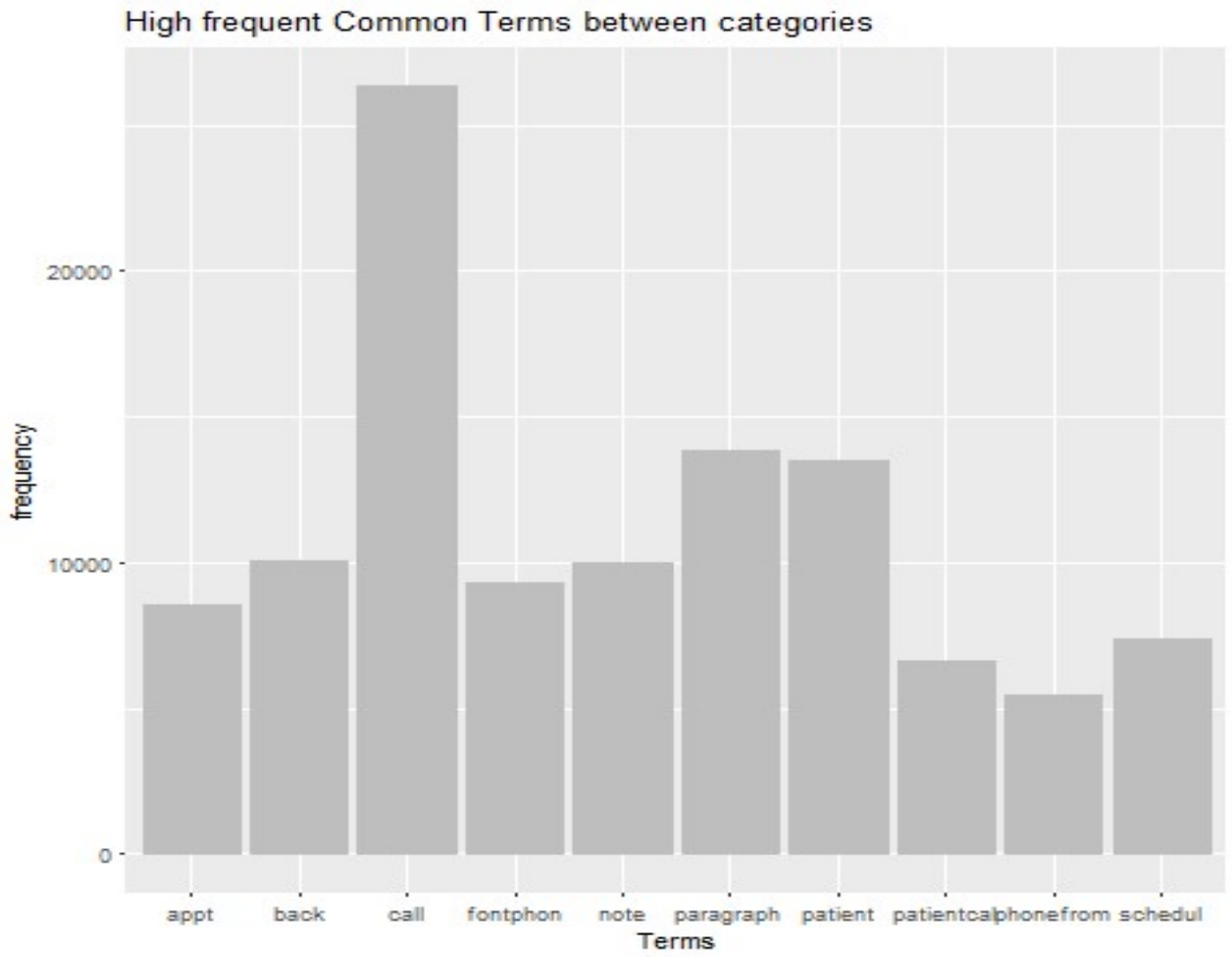


PRISCRIPTION CLOUD



LAB CLOUD

I have explored the individual datasets of each categories to have a better understanding of the generated terms. And also extracted the most common terms between the categories which I used in feature engineering.



Insights of data set:

- Words like 'call', 'patient', 'note', 'paragraph', 'back' 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.

Feature Engineering

Removing Corelated Terms

Checking correlation between the predictors is a must in Analysis. We 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. We have taken 85% as correlation limit and filtered out highly correlated terms from our structured data. Words like "marcia , 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] 40
```

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.When I designed a model with the dataset without corelated words, The accuracy is improved by 2-3% only. Let's try to remove the common words between the categories.

When we visualized the wordclouds of each category, there are many common words between the categories. These common words may effect the model accuracy. Let's see how these common words impact our model.

```
## [1] "Top 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] 57280   858
```

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.

Each master datasets is having dimensions as 57280 X 860, including the previous appointment variable and a target variable (categories or sub categories).

Sampling the master dataset and Forming train & test sets from the sample

The function sampling(), from the master data set samples it and returns train and test sets required from the sample. We are using **stratified sampling** to preserve this ratio throughout sampling and splitting. **caTools** package is used to implement stratified sampling.

```
library(caTools)
categories = data_req$categories
sub_categories = data_req$sub_categories
previous_appointment = data_req$previous_appointment
master_data_cat= as.data.frame(cbind(df,previous_appointment,sub_categories,c
ategories))
master_data_sub = as.data.frame(cbind(df,previous_appointment,sub_categories)
)

sampling <- function(master_data, set_seed = 123, samp.ratio= 0.075, train.ra
tio= 0.75){
        set.seed(seed = set_seed)
        samp_split = sample.split(master_data[,ncol(master_data)], samp.ratio
)
        sample = subset(master_data, samp_split == T)

        # training and testing
        smpl = sample.split(sample[,ncol(sample)], train.ratio)
        x_train = subset(sample, smpl == T )
        x_test  = subset(sample, smpl == F )
        y_train = x_train[,ncol(x_train)]
        y_test = x_test[,ncol(x_test)]
        x_train[,ncol(x_train)] = NULL
        x_test[,ncol(x_test)] = NULL
        train_test = list(x_train,x_test,y_train,y_test)
        return(train_test)
}

## [1] "Dimensions of each train and test sets for classifying categories exc
luding target variable are:"

## [1] 3222  860

## [1] 1073  860

## [1] "Dimensions of each train and test sets for classifying categories exc
luding target variable are:"

## [1] 3223  860

## [1] 1072  860
```

We can take different samples with different seeds to train the model.We are taking 10 samples of master data set and train our models using set.seed function. We are considering SVM as our Base model which is very significant in Text classification.

Random forest model is our ensemble model in this analysis.

All the models are trained with 4295 X 860 sample and split to 3223 X 860 Training and 1072 X 860 Testing sets (including target variable).

Model building and tuning

We have modelled SVM, Random Forest, Naive Bayes and Logistic regression models as an experiment. Principal component analysis is done, but they proved to be futile while modelling and hard to do PCA on huge data with R.

After Evaluating all these with **Naïve Bayes** model, **Random Forest** model and **SVM** model with 10 samples of data, we are more inclined to choose **SVM** as our model to freeze for our analysis though the accuracy Random Forest is more than SVM because Random Forest's system time is ~3.7 times the system time taken by SVM model. And a high computational power required for Random Forest classifier.

SVM Model

We are training with SVM as it uses a subset of training points in the decision function (called support vectors), so it is also memory efficient and also works really well with clear margin of separation.

SVM trained on 10 samples drawn with different seed gave a highest accuracy of **92%** for classifying categories and **48%** for classifying sub categories.

Note: If we use categories as an independent variable in classifying sub categories, then the SVM is classifying the sub categories with 65% accuracy.

```
## Summary of Predicted Classes - Categories

##  APPOINTMENTS  ASK_A_DOCTOR MISCELLANEOUS           LAB  PRESCRIPTION
##           273           224           264            41           271

## Confusion Matrix and Statistics
##
##                Reference
## Prediction      APPOINTMENTS ASK_A_DOCTOR MISCELLANEOUS LAB PRESCRIPTION
##   APPOINTMENTS           254            4             2   8            5
##   ASK_A_DOCTOR             1          215             2   1            5
##   MISCELLANEOUS            2            1           220  28           13
##   LAB                      0            0             1  40            0
##   PRESCRIPTION             3            1             3   4          260
##
## Overall Statistics
##
##                Accuracy : 0.9217
##                  95% CI : (0.904, 0.9371)
##     No Information Rate : 0.2637
##     P-Value [Acc > NIR] : < 2.2e-16
```

```
##
##                   Kappa : 0.8988
##  Mcnemar's Test P-Value : 3.046e-07
##
## Statistics by Class:
##
##                      Class: APPOINTMENTS Class: ASK_A_DOCTOR
## Sensitivity                       0.9769              0.9729
## Specificity                       0.9766              0.9894
## Pos Pred Value                    0.9304              0.9598
## Neg Pred Value                    0.9925              0.9929
## Prevalence                        0.2423              0.2060
## Detection Rate                    0.2367              0.2004
## Detection Prevalence              0.2544              0.2088
## Balanced Accuracy                 0.9768              0.9811
##                      Class: MISCELLANEOUS Class: LAB Class: PRESCRIPTION
## Sensitivity                        0.9649    0.49383              0.9187
## Specificity                        0.9479    0.99899              0.9861
## Pos Pred Value                     0.8333    0.97561              0.9594
## Neg Pred Value                     0.9901    0.96027              0.9713
## Prevalence                         0.2125    0.07549              0.2637
## Detection Rate                     0.2050    0.03728              0.2423
## Detection Prevalence               0.2460    0.03821              0.2526
## Balanced Accuracy                  0.9564    0.74641              0.9524
```

Summary of Predicted classes – **Sub Categories**

```
##                                 CANCELLATION
##                                            0
##                           CHANGE OF HOSPITAL
##                                            0
##                           CHANGE OF PHARMACY
##                                            0
##                           CHANGE OF PROVIDER
##                                            0
##                  FOLLOW UP ON PREVIOUS REQUEST
##                                            0
##                                       OTHERS
##                                           38
##                                  LAB RESULTS
##                                           34
##                           MEDICATION RELATED
##                                          297
##                              NEW APPOINTMENT
##                                          515
##                           PRIOR AUTHORIZATION
##                                            0
##                                     PROVIDER
##                                            1
##                    QUERIES FROM INSURANCE FIRM
##                                            0
```

```
##                         QUERIES FROM PHARMACY
##                                             0
##                   QUERY ON CURRENT APPOINTMENT
##                                             0
##                                      REFILL
##                                         169
##                                   RESCHEDULING
##                                             0
##                    RUNNING LATE TO APPOINTMENT
##                                             0
## SHARING OF HEALTH RECORDS (FAX, E-MAIL, ETC.)
##                                            18
##    SHARING OF LAB RECORDS (FAX, E-MAIL, ETC.)
##                                             0
##                                       SYMPTOMS
##                                             0

## Accuracy of Sub categories predicted

##       Accuracy          Kappa  AccuracyLower  AccuracyUpper   AccuracyNull
##   4.822761e-01   3.736446e-01   4.519813e-01   5.126685e-01   1.856343e-01

##
## AccuracyPValue  McnemarPValue
##  2.095013e-107            NaN
```

Random Forest

We have used h2o package to build random forest models as it is really quick in training a model using h2o.Random forest is flexible and can enhance the accuracy/performance of th e weak algorithm to a better extent, at the expense of heavier computational resources req uired.

Using h2o, when trained with random forests, we got a highest accuracy of **99%** for classify ing categories and **65%** accuracy for classifying sub categories. But with the same sample s ize used for SVM, the system time for random forest is ~3.7 times the system time taken by SVM.

Note: If we use categories as an independent variable in classifying sub categories, then the Random Forest is classifying the sub categories with 82% accuracy.

```
  |    Summary of Predicted classes - Categories

##           predict     APPOINTMENTS        ASK_A_DOCTOR
##  APPOINTMENTS :262   Min.   :0.0006085   Min.   :0.01447
##  ASK_A_DOCTOR :220   1st Qu.:0.0367041   1st Qu.:0.02785
##  LAB          : 78   Median :0.0808579   Median :0.06407
##  MISCELLANEOUS:229   Mean   :0.2457618   Mean   :0.19906
##  PRESCRIPTION :284   3rd Qu.:0.3217073   3rd Qu.:0.19885
##                      Max.   :0.9149333   Max.   :0.89532
##       LAB           MISCELLANEOUS       PRESCRIPTION
```

```
##  Min.   :0.001685   Min.   :0.001477   Min.   :0.01993
##  1st Qu.:0.006452   1st Qu.:0.045142   1st Qu.:0.03414
##  Median :0.011605   Median :0.088056   Median :0.06422
##  Mean   :0.074394   Mean   :0.216299   Mean   :0.26449
##  3rd Qu.:0.033485   3rd Qu.:0.203807   3rd Qu.:0.57594
##  Max.   :0.876724   Max.   :0.927609   Max.   :0.97419

## Confusion Matrix and Statistics
##
##                Reference
## Prediction      APPOINTMENTS ASK_A_DOCTOR MISCELLANEOUS LAB PRESCRIPTION
##   APPOINTMENTS           260            0             0   2            0
##   ASK_A_DOCTOR             0          220             0   0            0
##   MISCELLANEOUS            0            0           228   1            0
##   LAB                      0            0             0  78            0
##   PRESCRIPTION             0            1             0   0          283
##
## Overall Statistics
##
##                Accuracy : 0.9963
##                  95% CI : (0.9905, 0.999)
##     No Information Rate : 0.2637
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.9952
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: APPOINTMENTS Class: ASK_A_DOCTOR
## Sensitivity                       1.0000              0.9955
## Specificity                       0.9975              1.0000
## Pos Pred Value                    0.9924              1.0000
## Neg Pred Value                    1.0000              0.9988
## Prevalence                        0.2423              0.2060
## Detection Rate                    0.2423              0.2050
## Detection Prevalence              0.2442              0.2050
## Balanced Accuracy                 0.9988              0.9977
##                      Class: MISCELLANEOUS Class: LAB Class: PRESCRIPTION
## Sensitivity                        1.0000    0.96296              1.0000
## Specificity                        0.9988    1.00000              0.9987
## Pos Pred Value                     0.9956    1.00000              0.9965
## Neg Pred Value                     1.0000    0.99698              1.0000
## Prevalence                         0.2125    0.07549              0.2637
## Detection Rate                     0.2125    0.07269              0.2637
## Detection Prevalence               0.2134    0.07269              0.2647
## Balanced Accuracy                  0.9994    0.98148              0.9994
```

Summary of predicted classes – Sub Categories

```
##                                              predict
##  MEDICATION RELATED                              :247
##  NEW APPOINTMENT                                 :243
##  REFILL                                          :212
##  OTHERS                                          :139
##  SHARING OF HEALTH RECORDS (FAX, E-MAIL, ETC.): 73
##  LAB RESULTS                                     : 57
##  (Other)                                         :101
```

```
## Accuracy of sub categories predicted
```

```
##       Accuracy          Kappa  AccuracyLower  AccuracyUpper   AccuracyNull
##   6.595149e-01   6.021786e-01   6.302735e-01   6.878775e-01   1.856343e-01
```

```
## AccuracyPValue  McnemarPValue
##  2.800265e-253            NaN
```

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 min imal computational power and time. So, the model and seed with highest accuracy is our model of Deployment.

So, **SVM** with seed **555** has freeze as our deployment model with an accuracy of 92% for cl assifying 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.