

## **PROJECT REPORT**

### **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 Health 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 problem 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).**

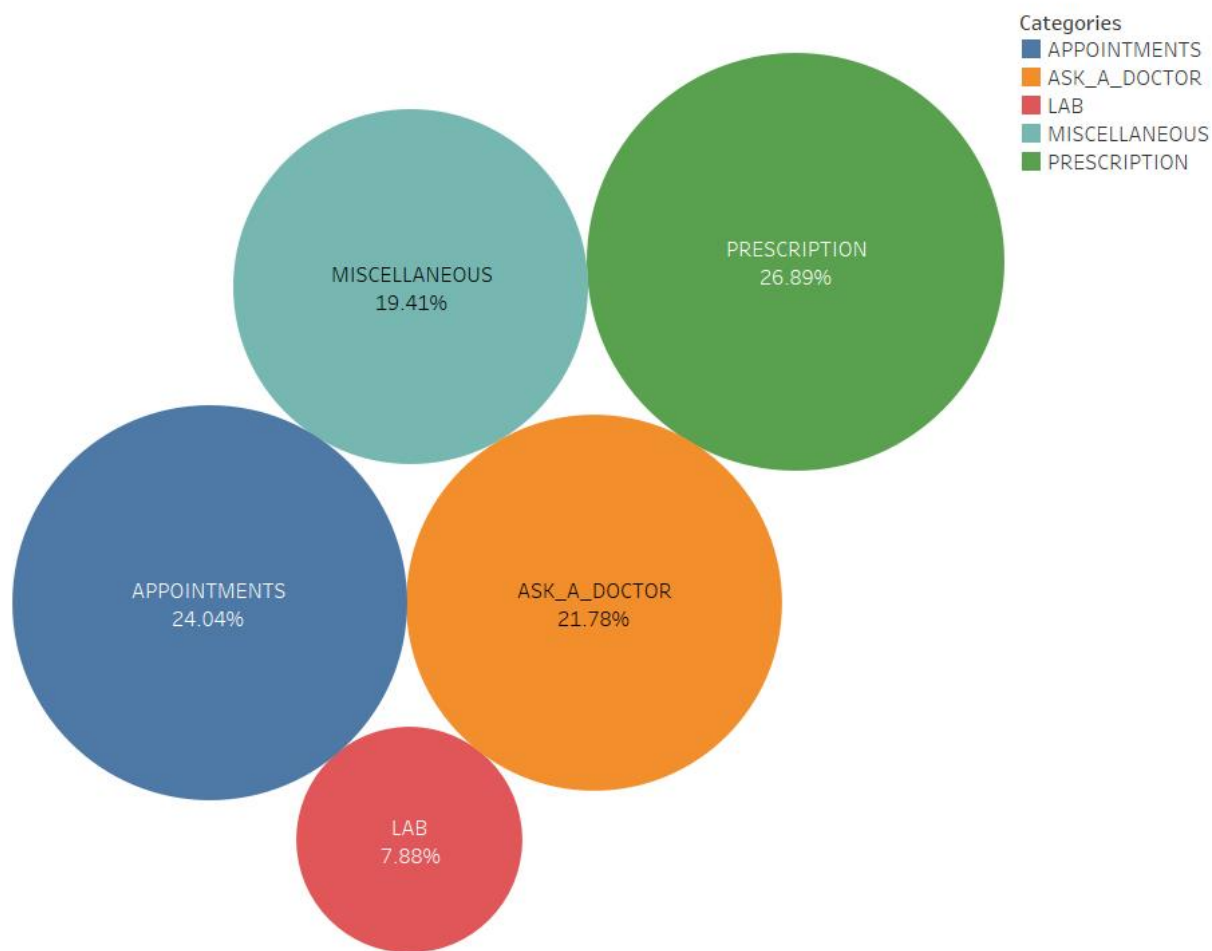
### **HYPOTHESEIS GENERATION**

Generally phone calls are done by persons who are having ailments so the phonecall data contains appointment booking ,medication related , and followup of the visit of the data.one can find a relation between runtime in overall experience of the person starting from booking appointment to cure of his ailment. This wholecycle involves diagnosis of ailment, lab reports if necessary, medication related, followup once dosage of medicine taken by the patient and results.this end to end cycle we can analyse from the patient id summary and description of call and according we can classify each and every patient experience and there by identifying pain points of the customers and providing better solutions for the patients there by enhanced services of the company in a holistic manner making it more competitive enough than before.For that to happen we need better model the text data of the information gathered from the description and summary of the phone data in better manner leading to best in classifying in to best accuracy of classifying into respective categories and subcategories.

### **Exploratory Data Analysis**

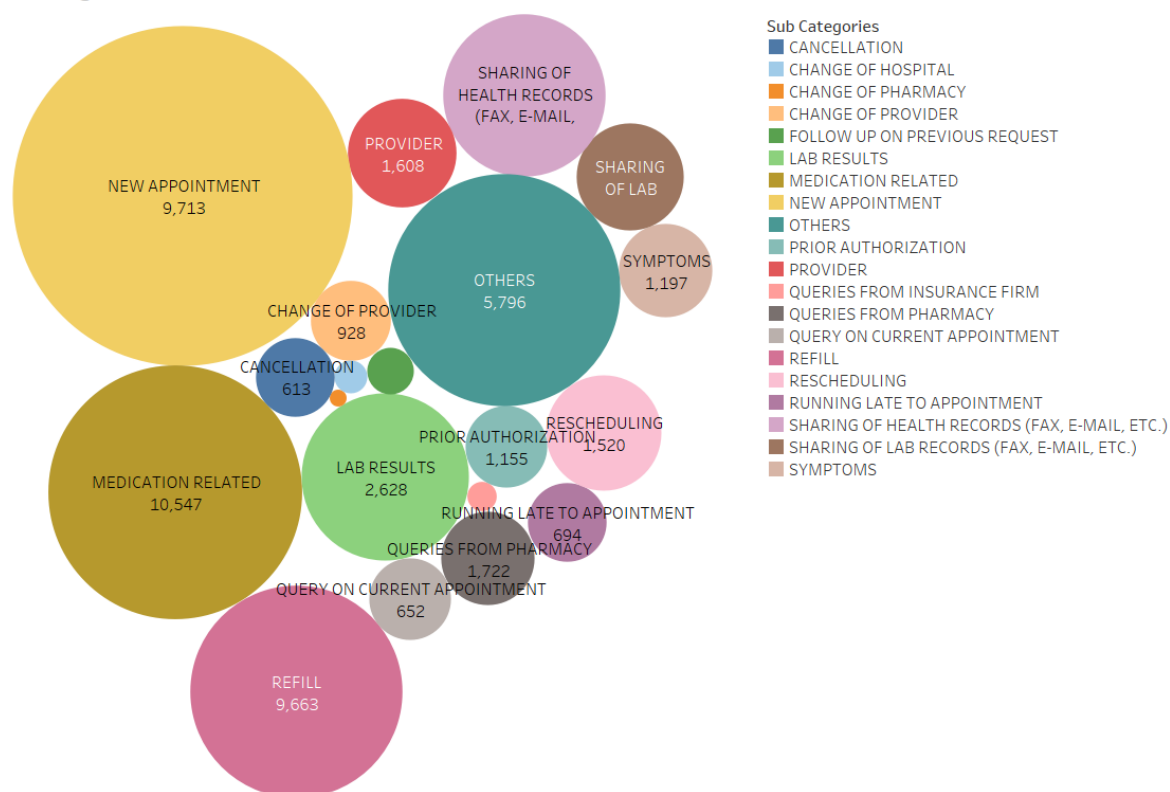
All are categorical variables mostly. since there is missing data of variable named "DATA" which accounts for 5% of data, so excluding that overall that data around 57,910 observations

Categories



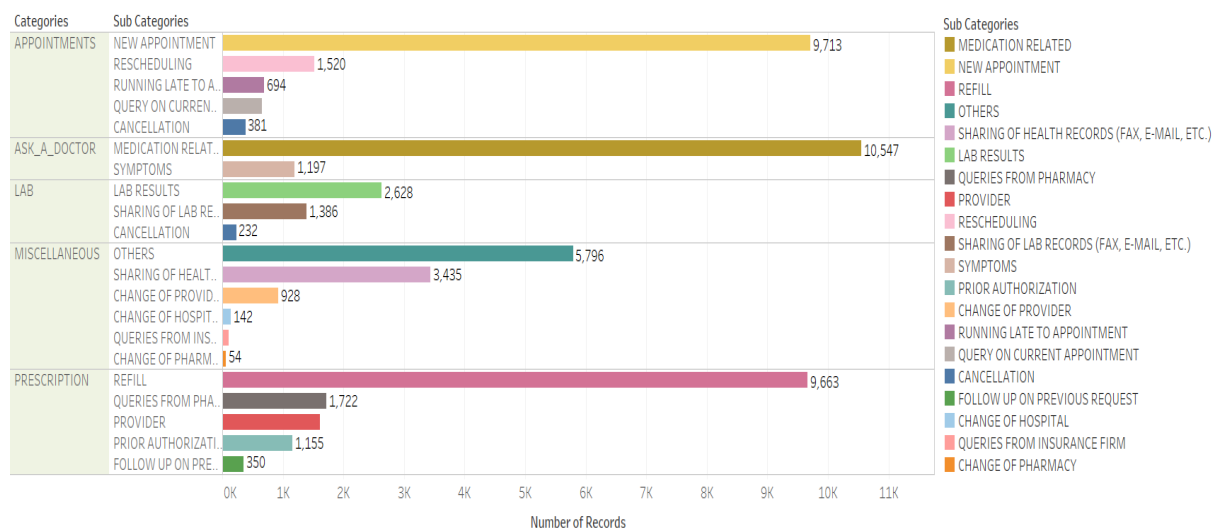
Categories and % of Total Number of Records. Color shows details about Categories. Size shows % of Total Number of Records. The marks are labeled by Categories and % of Total Number of Records. Details are shown for Categories. Percents are based on each cell of each pane of the table.

## SubCategories



Sub Categories and sum of Number of Records. Color shows details about Sub Categories. Size shows sum of FileId. The marks are labeled by Sub Categories and sum of Number of Records.

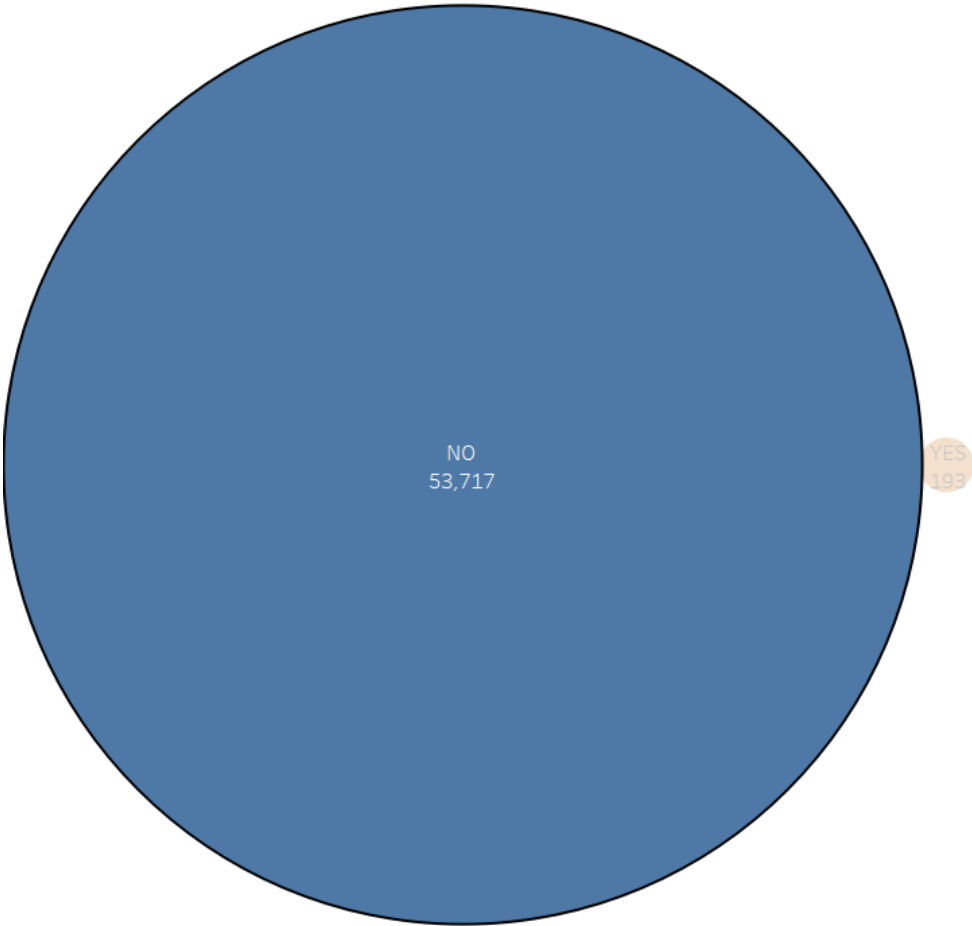
## Categories & SubCategories correlation



Sum of Number of Records for each Sub Categories broken down by Categories. Color shows details about Sub Categories. The marks are labeled by sum of Number of Records.

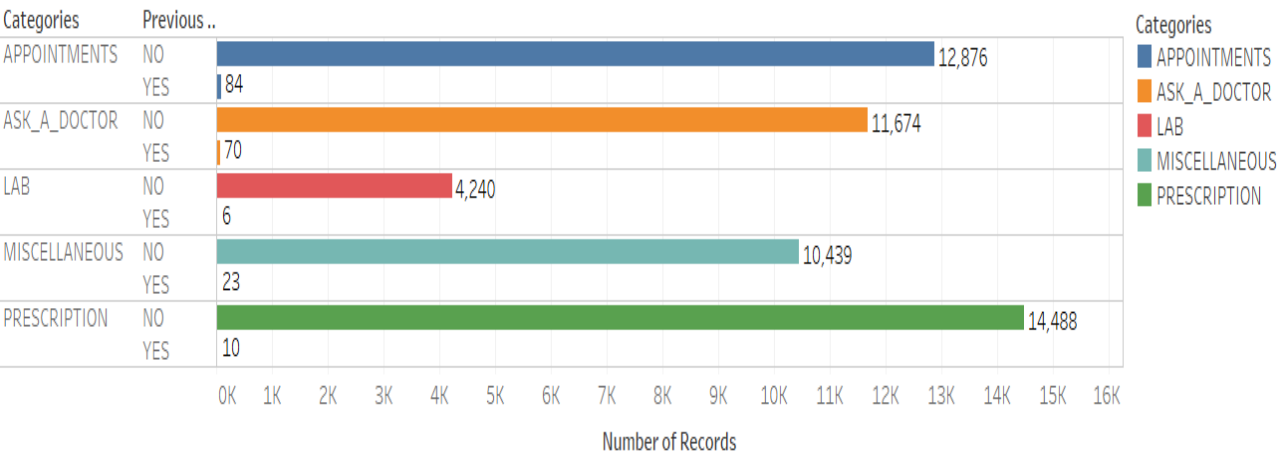
# Previous Appointments

Previous Appointment  
■ NO  
■ YES



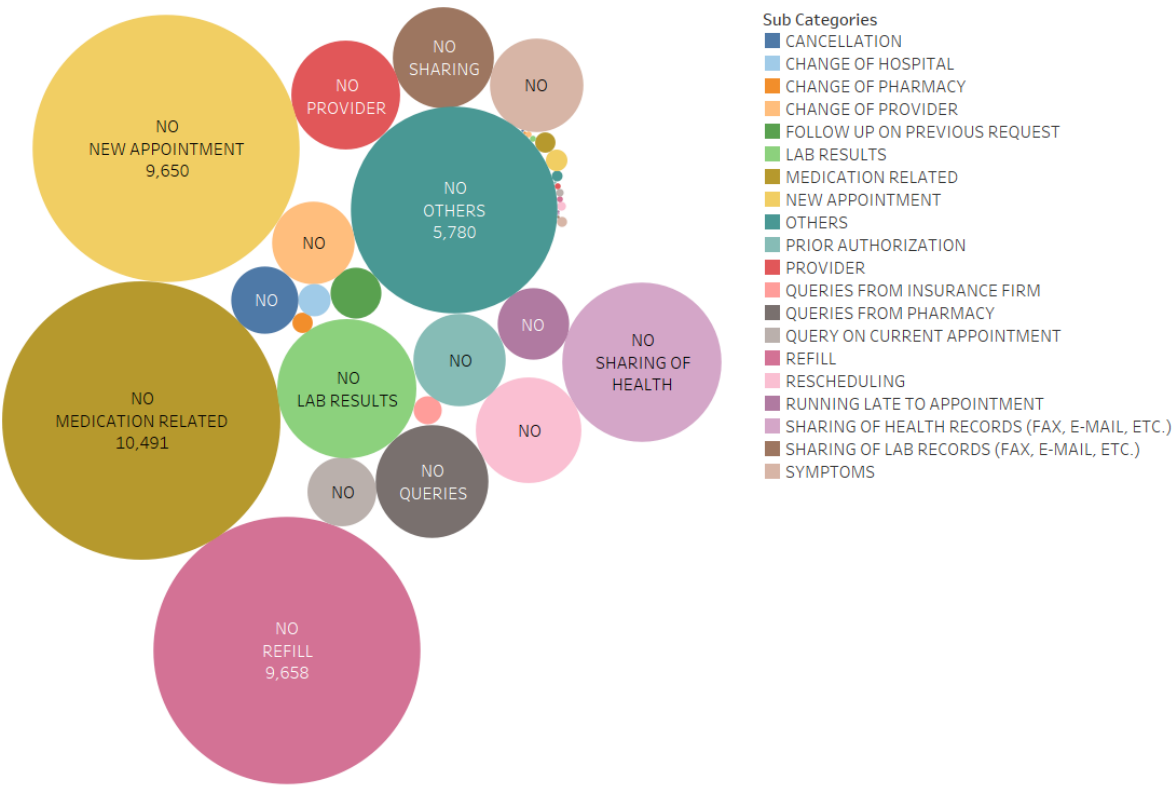
Previous Appointment and sum of Number of Records. Color shows details about Previous Appointment. Size shows sum of Number of Records. The marks are labeled by Previous Appointment and sum of Number of Records.

# Categories Vs Previous Appointments



Sum of Number of Records for each Previous Appointment broken down by Categories. Color shows details about Categories. The marks are labeled by sum of Number of Records.

## SubCategories Vs PreviousAppointment



### WORDCLOUD OF TEXTUAL DATA



carparfsb place use sign hcl mriadv clinic action convert per  
left mari close contact pmparfsbpar  
say rna action thank method  
week hcl mriadv clinic  
ask refillparb advise patientpar phone come increa  
bxxxxxxx changesparfspar now still see  
tablets fill medic end med s  
usedtab form send work just  
flag enter authori test  
check updatedparb bxxxxxxx lpn  
signedpar prescript follow note  
wrotepar

---

med advise parb notifi offic bid  
sinc par talk xxx bytab still  
pick start parfsb parb due test  
prescript signedparb print pmparb  
question provid dose note take  
left check author reason verbal now  
hcl clinic send just per increa ask receiv use  
method queu fsbpar chang end  
followup return  
bxxxxxxxxb done authorizedxxxxxxxx  
prn

---



## Findings & Recommendations

### Findings & Recommendations

In Categories section prescription tops of all followed by appointments and ask a doctor categories, which indicates better performance of the company which are major revenue generators. so better investing more in qualitative customer support further boost up the company revenue a lot

In sub category section, customers tops at new medication related, new appointment and refill these are the top sub categories which forms major revenue generation of the company but follow upon requests which is one of the least sub category need to focus on that area which improves better care for patients and improving recurring revenues for the company Also keep tracking of health records, lab records of each and every patients helps in developing better predictive measures in future for those already diagnosed patients.

Need to decrease the appointment cancellation requests finding reasons for that and according decrease the cancellations of appointments tends to zero will improve the revenue share of the company.

The previous appointment category "NO" is highest indicates either patients are feeling their experience on a whole is not upto the mark churning out to other hospitals are high, it is complex and need to be monitored in better way

As a whole new appointments, medication and doctor suggestions tops of all which is keeping the company at top position but if lab records, medication reports are taken care well and data related to that of each and every patient record when kept maintained and analysed properly will give more opportunities, open new doors and strategic advantage to this healthcare company against other since it helps in predictive measures for the patients and new patients can be treated well providing a better holistic experience which is utmost need of the hour in present health care market,

Libraries used

Matrix'

Tidyr

Tm

Dplyr

Tibble

MLr

RandomforestSRC

Rferns

MLMetrics

## Feature Engineering

In this multilabel classification we are doing one hot encoding to Category section and Sub Category variables into binary coding into 1's and 0's using sparse model matrix and then bringing back to data frame structure. And for text mining since we need to model two types of textual variables we are combining both these textual vectors and then applying tf-idf technique to bring best variables out of them.Labelled the target variables from v1 to v25 and these are binary ,logical in nature. Encoded as TRUE & FALSE taking "0 "as FALSE and "1" as TRUE.

[v1] "categoriesAPPOINTMENTS"  
[v2] "categoriesASK\_A\_DOCTOR"  
[v3] "categoriesLAB"  
[v4] "categoriesMISCELLANEOUS"  
[v5] "categoriesPRESCRIPTION"  
[v6] "sub\_categoriesCANCELLATION"  
[v7] "sub\_categoriesCHANGE OF HOSPITAL"  
[v8] "sub\_categoriesCHANGE OF PHARMACY"  
[v9] "sub\_categoriesCHANGE OF PROVIDER"  
[v10] "sub\_categoriesFOLLOW UP ON PREVIOUS REQUEST"  
[v11] "sub\_categoriesLAB RESULTS"  
[v12] "sub\_categoriesMEDICATION RELATED"  
[v13] "sub\_categoriesNEW APPOINTMENT"  
[v14] "sub\_categoriesOTHERS"  
[v15] "sub\_categoriesPRIOR AUTHORIZATION"  
[v16] "sub\_categoriesPROVIDER"  
[v17] "sub\_categoriesQUERIES FROM INSURANCE FIRM"  
[v18] "sub\_categoriesQUERIES FROM PHARMACY"  
[v19] "sub\_categoriesQUERY ON CURRENT APPOINTMENT"  
[v20] "sub\_categoriesREFILL"  
[v21] "sub\_categoriesRESCHEDULING"  
[v22] "sub\_categoriesRUNNING LATE TO APPOINTMENT"  
[v23] "sub\_categoriesSHARING OF HEALTH RECORDS (FAX, E-MAIL, ETC.)"  
[v24] "sub\_categoriesSHARING OF LAB RECORDS (FAX, E-MAIL, ETC.)"  
[v25] "sub\_categoriesSYMPTOMS"

## Model Building

Since in this problem we need to predict both category section and sub category section at a time, it is multilabel classification problem in text mining Here we need to predict 25 target variables at a time applying MLR library and implement two machine learning techniques for better accuracy and distinction.

They are 1) RandomForestSurvivalRegressionClassification known as RFSRC

2) Random Ferns known as Rferns

We got only these two machine learning algorithms to predict for multilabel classification.

Model evaluation

Both the models are giving high accuracies above 90% .

## RESULTS

Results include comparison of evaluation metrics of the two models .since we had twenty five target variables which are binary logical in nature. Here we are using AUC and Accuracy metrics for the valuation of the model of each and every variable since every variable comes under binary classification. Below are the computed values of both AUC and Accuracy metrics for each and every variable with their respective accuracies and AUC's are compared for each and every target variable and compared both model efficiencies

```
> Accuracy(y_pred=pred_rfsrc_N$data$response.v1,y_true=test_data$v1)
```

```
[1] 0.9269152
```

```
> #Accuracy of rfsrccmodel
```

```
> Accuracy(y_pred=pred_rfsrc_N$data$response.v2,y_true=test_data$v2)
```

```
[1] 0.8791504
```

```
> #Accuracy of rfsrccmodel
```

```
> Accuracy(y_pred=pred_rfsrc_N$data$response.v3,y_true=test_data$v3)
```

```
[1] 0.9633649
```

```
> #Accuracy of rfsrccmodel
```

```
> Accuracy(y_pred=pred_rfsrc_N$data$response.v4,y_true=test_data$v4)
```

```
[1] 0.8642181
```

```
> #Accuracy of rfsrccmodel
```

```
> Accuracy(y_pred=pred_rfsrc_N$data$response.v5,y_true=test_data$v5)
```

```
[1] 0.9027082
```

```
> #Accuracy of rfsrccmodel
```

```
> Accuracy(y_pred=pred_rfsrc_N$data$response.v6,y_true=test_data$v6)
[1] 0.9893341
> #Accuracy of rfsrcmodel
> Accuracy(y_pred=pred_rfsrc_N$data$response.v7,y_true=test_data$v7)
[1] 0.9967538
> #Accuracy of rfsrcmodel
> Accuracy(y_pred=pred_rfsrc_N$data$response.v9,y_true=test_data$v9)
[1] 0.9819143
> #Accuracy of rfsrcmodel
> Accuracy(y_pred=pred_rfsrc_N$data$response.v10,y_true=test_data$v10)
[1] 0.9936004
> #Accuracy of rfsrcmodel
> Accuracy(y_pred=pred_rfsrc_N$data$response.v11,y_true=test_data$v11)
[1] 0.9784827
> #Accuracy of rfsrcmodel
> Accuracy(y_pred=pred_rfsrc_N$data$response.v12,y_true=test_data$v12)
[1] 0.8737711
> #Accuracy of rfsrcmodel
> Accuracy(y_pred=pred_rfsrc_N$data$response.v13,y_true=test_data$v13)
[1] 0.9194027
> #Accuracy of rfsrcmodel
> Accuracy(y_pred=pred_rfsrc_N$data$response.v14,y_true=test_data$v14)
[1] 0.9182897
> #Accuracy of rfsrcmodel
> Accuracy(y_pred=pred_rfsrc_N$data$response.v15,y_true=test_data$v15)
[1] 0.9824708
> #Accuracy of rfsrcmodel
> Accuracy(y_pred=pred_rfsrc_N$data$response.v16,y_true=test_data$v16)
[1] 0.9728251
```

```
> #Accuracy of rfsrccmodel
> Accuracy(y_pred=pred_rfsrc_N$data$response.v17,y_true=test_data$v17)
[1] 0.9986088
> #Accuracy of rfsrccmodel
> Accuracy(y_pred=pred_rfsrc_N$data$response.v18,y_true=test_data$v18)
[1] 0.9841402
> #Accuracy of rfsrccmodel
> Accuracy(y_pred=pred_rfsrc_N$data$response.v19,y_true=test_data$v19)
[1] 0.9883139
> #Accuracy of rfsrccmodel
> Accuracy(y_pred=pred_rfsrc_N$data$response.v20,y_true=test_data$v20)
[1] 0.9513077
> #Accuracy of rfsrccmodel
> Accuracy(y_pred=pred_rfsrc_N$data$response.v21,y_true=test_data$v21)
[1] 0.9741235
> #Accuracy of rfsrccmodel
> Accuracy(y_pred=pred_rfsrc_N$data$response.v22,y_true=test_data$v22)
[1] 0.9952699
> #Accuracy of rfsrccmodel
> Accuracy(y_pred=pred_rfsrc_N$data$response.v23,y_true=test_data$v23)
[1] 0.9437952
> #Accuracy of rfsrccmodel
> Accuracy(y_pred=pred_rfsrc_N$data$response.v24,y_true=test_data$v24)
[1] 0.976164
> #Accuracy of rfsrccmodel
> Accuracy(y_pred=pred_rfsrc_N$data$response.v25,y_true=test_data$v25)
[1] 0.975422
```

>Now observing AUC of each and every target variable in randomforestSRC Model

> AUC(y\_pred=pred\_rfsrc\_N\$data\$response.v1,y\_true=test\_data\$v1)

[1] 0.884844

> #Accuracy of rfsrccmodel

> AUC(y\_pred=pred\_rfsrc\_N\$data\$response.v2,y\_true=test\_data\$v2)

[1] 0.7710847

> #Accuracy of rfsrccmodel

> AUC(y\_pred=pred\_rfsrc\_N\$data\$response.v3,y\_true=test\_data\$v3)

[1] 0.8049867

> #Accuracy of rfsrccmodel

> AUC(y\_pred=pred\_rfsrc\_N\$data\$response.v4,y\_true=test\_data\$v4)

[1] 0.7062303

> #Accuracy of rfsrccmodel

> AUC(y\_pred=pred\_rfsrc\_N\$data\$response.v5,y\_true=test\_data\$v5)

[1] 0.8477085

> #Accuracy of rfsrccmodel

> AUC(y\_pred=pred\_rfsrc\_N\$data\$response.v6,y\_true=test\_data\$v6)

[1] 0.5128373

> #Accuracy of rfsrccmodel

> AUC(y\_pred=pred\_rfsrc\_N\$data\$response.v7,y\_true=test\_data\$v7)

[1] 0.5

> #Accuracy of rfsrccmodel

> AUC(y\_pred=pred\_rfsrc\_N\$data\$response.v8,y\_true=test\_data\$v8)

[1] 0.5

> #Accuracy of rfsrccmodel

> AUC(y\_pred=pred\_rfsrc\_N\$data\$response.v9,y\_true=test\_data\$v9)

[1] 0.5

> #Accuracy of rfsrccmodel

> AUC(y\_pred=pred\_rfsrc\_N\$data\$response.v10,y\_true=test\_data\$v10)

```

[1] 0.5071997
> #Accuracy of rfsrccmodel
> AUC(y_pred=pred_rfsrc_N$data$response.v11,y_true=test_data$v11)
[1] 0.8407655
> #Accuracy of rfsrccmodel
> AUC(y_pred=pred_rfsrc_N$data$response.v12,y_true=test_data$v12)
[1] 0.7298236
> #Accuracy of rfsrccmodel
> AUC(y_pred=pred_rfsrc_N$data$response.v13,y_true=test_data$v13)
[1] 0.8171566
> #Accuracy of rfsrccmodel
> AUC(y_pred=pred_rfsrc_N$data$response.v14,y_true=test_data$v14)
[1] 0.6429286
> #Accuracy of rfsrccmodel
> AUC(y_pred=pred_rfsrc_N$data$response.v15,y_true=test_data$v15)
[1] 0.614958
> #Accuracy of rfsrccmodel
> AUC(y_pred=pred_rfsrc_N$data$response.v16,y_true=test_data$v16)
[1] 0.5715041
> #Accuracy of rfsrccmodel
> AUC(y_pred=pred_rfsrc_N$data$response.v17,y_true=test_data$v17)
[1] 0.5
> #Accuracy of rfsrccmodel
> AUC(y_pred=pred_rfsrc_N$data$response.v118y_true=test_data$v18)
Error: unexpected '=' in "AUC(y_pred=pred_rfsrc_N$data$response.v118y_true="
> #Accuracy of rfsrccmodel
> AUC(y_pred=pred_rfsrc_N$data$response.v18,y_true=test_data$v18)
[1] 0.8035025
> #Accuracy of rfsrccmodel

```

```

> AUC(y_pred=pred_rfsrc_N$data$response.v19,y_true=test_data$v19)
[1] 0.5
> #Accuracy of rfsrcmodel
> AUC(y_pred=pred_rfsrc_N$data$response.v20,y_true=test_data$v20)
[1] 0.8852304
> #Accuracy of rfsrcmodel
> AUC(y_pred=pred_rfsrc_N$data$response.v21,y_true=test_data$v21)
[1] 0.5258427
> #Accuracy of rfsrcmodel
> AUC(y_pred=pred_rfsrc_N$data$response.v22,y_true=test_data$v22)
[1] 0.8111111
> #Accuracy of rfsrcmodel
> AUC(y_pred=pred_rfsrc_N$data$response.v23,y_true=test_data$v23)
[1] 0.616871
> #Accuracy of rfsrcmodel
> AUC(y_pred=pred_rfsrc_N$data$response.v24,y_true=test_data$v24)
[1] 0.5725639
> #Accuracy of rfsrcmodel
> AUC(y_pred=pred_rfsrc_N$data$response.v25,y_true=test_data$v25)
[1] 0.5

```

#### Pros of rfsrc

- Robust model for the best accuracy performs well on almost any data
- Prediction is very good
- Accuracy results in more than 90%
- Cons of RFSRC
- Run time is very high more than 15 mins
- Model occupies huge space around 1.3GB when compared with rferns model



### Results of rferns model

> #Accuracy of rferns model

> Accuracy(y\_pred=pred\_rferns\$data\$response.v1,y\_true=test\_data\$v1)

[1] 0.9269152

> #Accuracy of rferns model

> Accuracy(y\_pred=pred\_rferns\$data\$response.v2,y\_true=test\_data\$v2)

[1] 0.8787794

> #Accuracy of rferns model

> Accuracy(y\_pred=pred\_rferns\$data\$response.v3,y\_true=test\_data\$v3)

[1] 0.9633649

> #Accuracy of rferns model

> Accuracy(y\_pred=pred\_rferns\$data\$response.v4,y\_true=test\_data\$v4)

[1] 0.8642181

> #Accuracy of rferns model

> Accuracy(y\_pred=pred\_rferns\$data\$response.v5,y\_true=test\_data\$v5)

[1] 0.9027082

> #Accuracy of rferns model

> Accuracy(y\_pred=pred\_rferns\$data\$response.v7,y\_true=test\_data\$v7)

[1] 0.9967538

> #Accuracy of rferns model

> Accuracy(y\_pred=pred\_rferns\$data\$response.v8,y\_true=test\_data\$v8)

[1] 0.998516

> #Accuracy of rferns model

> Accuracy(y\_pred=pred\_rferns\$data\$response.v9,y\_true=test\_data\$v9)

[1] 0.9819143

> #Accuracy of rferns model

> Accuracy(y\_pred=pred\_rferns\$data\$response.v10,y\_true=test\_data\$v10)

[1] 0.9936004

> #Accuracy of rferns model

> Accuracy(y\_pred=pred\_rferns\$data\$response.v11,y\_true=test\_data\$v11)

[1] 0.9784827

> #Accuracy of rferns model

> Accuracy(y\_pred=pred\_rferns\$data\$response.v12,y\_true=test\_data\$v12)

[1] 0.8740493

> #Accuracy of rferns model

> Accuracy(y\_pred=pred\_rferns\$data\$response.v13,y\_true=test\_data\$v13)

[1] 0.9194027

> #Accuracy of rferns model

> Accuracy(y\_pred=pred\_rferns\$data\$response.v14,y\_true=test\_data\$v14)

[1] 0.9182897

> #Accuracy of rferns model

> Accuracy(y\_pred=pred\_rferns\$data\$response.v15,y\_true=test\_data\$v15)

[1] 0.9824708

> #Accuracy of rferns model

> Accuracy(y\_pred=pred\_rferns\$data\$response.v16,y\_true=test\_data\$v16)

[1] 0.9728251

> #Accuracy of rferns model

> Accuracy(y\_pred=pred\_rferns\$data\$response.v17,y\_true=test\_data\$v17)

[1] 0.9986088

> #Accuracy of rferns model

> Accuracy(y\_pred=pred\_rferns\$data\$response.v18,y\_true=test\_data\$v18)

[1] 0.9841402

> #Accuracy of rferns model

> Accuracy(y\_pred=pred\_rferns\$data\$response.v19,y\_true=test\_data\$v19)

[1] 0.9883139

> #Accuracy of rferns model

> Accuracy(y\_pred=pred\_rferns\$data\$response.v20,y\_true=test\_data\$v20)

[1] 0.9513077

> #Accuracy of rferns model

> Accuracy(y\_pred=pred\_rferns\$data\$response.v21,y\_true=test\_data\$v21)

[1] 0.9741235

> #Accuracy of rferns model

> Accuracy(y\_pred=pred\_rferns\$data\$response.v22,y\_true=test\_data\$v22)

[1] 0.9952699

> #Accuracy of rferns model

> Accuracy(y\_pred=pred\_rferns\$data\$response.v23,y\_true=test\_data\$v23)

[1] 0.9437952

> #Accuracy of rferns model

> Accuracy(y\_pred=pred\_rferns\$data\$response.v24,y\_true=test\_data\$v24)

[1] 0.976164

> #Accuracy of rferns model

> Accuracy(y\_pred=pred\_rferns\$data\$response.v25,y\_true=test\_data\$v25)

[1] 0.975422

>Now evaluating with AUC metric

> #Accuracy of rferns model

> AUC(y\_pred=pred\_rferns\$data\$response.v1,y\_true=test\_data\$v1)

[1] 0.884844

> #Accuracy of rferns model

> AUC(y\_pred=pred\_rferns\$data\$response.v2,y\_true=test\_data\$v2)

[1] 0.7703884

> #Accuracy of rferns model

> AUC(y\_pred=pred\_rferns\$data\$response.v3,y\_true=test\_data\$v3)

[1] 0.8049867

> #Accuracy of rferns model

```
> AUC(y_pred=pred_rferns$data$response.v4,y_true=test_data$v4)
[1] 0.7062303

> #Accuracy of rferns model

> AUC(y_pred=pred_rferns$data$response.v5,y_true=test_data$v5)
[1] 0.8477085

> #Accuracy of rferns model

> AUC(y_pred=pred_rferns$data$response.v6,y_true=test_data$v6)
[1] 0.5128373

> #Accuracy of rferns model

> AUC(y_pred=pred_rferns$data$response.v7,y_true=test_data$v7)
[1] 0.5

> #Accuracy of rferns model

> AUC(y_pred=pred_rferns$data$response.v8,y_true=test_data$v8)
[1] 0.5

> #Accuracy of rferns model

> AUC(y_pred=pred_rferns$data$response.v9,y_true=test_data$v9)
[1] 0.5

> #Accuracy of rferns model

> AUC(y_pred=pred_rferns$data$response.v10,y_true=test_data$v10)
[1] 0.5071997

> #Accuracy of rferns model

> AUC(y_pred=pred_rferns$data$response.v11,y_true=test_data$v11)
[1] 0.8407655

> #Accuracy of rferns model

> AUC(y_pred=pred_rferns$data$response.v12,y_true=test_data$v12)
[1] 0.7303595

> #Accuracy of rferns model

> AUC(y_pred=pred_rferns$data$response.v13,y_true=test_data$v13)
[1] 0.8171566
```

```
> #Accuracy of rferns model
> AUC(y_pred=pred_rferns$data$response.v14,y_true=test_data$v14)
[1] 0.6429286
> #Accuracy of rferns model
> AUC(y_pred=pred_rferns$data$response.v15,y_true=test_data$v15)
[1] 0.614958
> #Accuracy of rferns model
> AUC(y_pred=pred_rferns$data$response.v16,y_true=test_data$v16)
[1] 0.5715041
> #Accuracy of rferns model
> AUC(y_pred=pred_rferns$data$response.v17,y_true=test_data$v17)
[1] 0.5
> #Accuracy of rferns model
> AUC(y_pred=pred_rferns$data$response.v18,y_true=test_data$v18)
[1] 0.8035025
> #Accuracy of rferns model
> AUC(y_pred=pred_rferns$data$response.v19,y_true=test_data$v19)
[1] 0.5
> #Accuracy of rferns model
> AUC(y_pred=pred_rferns$data$response.v20,y_true=test_data$v20)
[1] 0.8852304
> #Accuracy of rferns model
> AUC(y_pred=pred_rferns$data$response.v21,y_true=test_data$v21)
[1] 0.5258427
> #Accuracy of rferns model
> AUC(y_pred=pred_rferns$data$response.v22,y_true=test_data$v22)
[1] 0.8111111
> #Accuracy of rferns model
> AUC(y_pred=pred_rferns$data$response.v23,y_true=test_data$v23)
```

[1] 0.616871

> #Accuracy of rferns model

> AUC(y\_pred=pred\_rferns\$data\$response.v24,y\_true=test\_data\$v24)

[1] 0.5725639

> #Accuracy of rferns model

> AUC(y\_pred=pred\_rferns\$data\$response.v25,y\_true=test\_data\$v25)

[1] 0.5

We can see distinct variations when comparative analysis of evaluation metrics of both the models

- pros of RFERNS
- Least run time just about a minute implying it is computed very fast when compared with rfsrc model
- Accuracy is above 90%.
- Model occupies very less space when compared with rfsrc model which occupies around 1.3GB of the data.
- Cons
- Less accuracy when compared with RFSRC model.

The results contain prediction of test results containing prediction of target variables.

#### Conclusion:

On a whole both the models are efficient enough to classify better both the categories and subcategories sections and can be implemented for the production system .

Github llink of the project

<https://github.com/GRMK11/Edwisor-Project>