OSDS-1907 PROJECT

By KRUTI SHAH

United Health Group Baker's Call Prediction Challenge

Logistic Regression using R

Kruti Shah – OSDS -1907

1. Data Source /Lineage and Description

In order to find a real business problem, I reached out to the provider domain Data Analytics team of the provider organization. I was fortunate to get their help in getting a data set which had a real business problem defined. Mike Baker had given a challenge to the Data science teams to pick up the challenge to deliver a business actionable solution to improve our provider experience by predicting call volume. The formal name of the challenge was "**The Baker Innovation Challenge: Call Prediction"**

Special thanks to Cathy Olson for providing the documentation and data set for this challenge. This challenge was played in April 2018. The data in datasets are from 2017.

The documents are located at

This PPT explains the challenge objectives and the data.

https://hubconnect.uhg.com/docs/DOC-183262

The data and dictionary are located here:

https://github.optum.com/rkinsell/Baker_Challenge_Dataset

This contains 3 data sets

❖ Claim-call

The "claim-call" data set is a claim level dataset that contains UHCInsight providers' claims submitted in 2017. It includes key claim & call (for claims that have had a call) attributes. The data is de-identified.

- Provider Profile

  The “provider profile” data set contains information about the providers that serviced the claims. Two features (out of 16) that contain identification numbers are de-identified thus keeping the identity secure.

- Procedure Details:

  The procedure file contains the procedure details of which claims are present in the Claim file.

  There are 16 features in this data set. Out of 16 features, 2 features which are unique identifiers are de-identified.

  The Data Dictionary for the 3 files is provided in the Appendix.

2. Context / background

During the Claim Adjudication process, it has been observed that contacts from provider's office call the customer service to gain information about their claim(s).

As per 2017 data, United Healthcare Group is getting around 36M calls in a year. Out of that 7.6 M calls gets transferred into different departments. These 7.6 M calls in turn cost $35M. These large volumes of calls involve 6000+ advocates and 100+ applications.

The large volumes of calls indicate that either provider doesn't understand our claim adjudication process or have concerns for the same. We need to make our processes better so that the providers don't have to make the call. It can also be that provider may need some training. Provider calling the customer service also impacts our provider NPS score. The lesser the calls from the provider, the happier the providers are and higher will be the NPS score.

Based on the claim and provider data, analysis can be performed to understand which attribute /feature has impact on the generation of the call and whether a call can be predicted.

3. Business problem and question

Provider Business received around 36 M calls in 2017.Out of the 36 M calls received, 7.6 M calls were transferred to other departments further. The projected costs of this 7.6 M transfer calls is around $35M +. There are around 6000 + call advocates handling these transferred calls and they use around 100 + Operations.

Question: Can we predict a call from provider analyzing the claims call data?

Please note that the scope of this study

Using that prediction can we

- ❖ Reach out to providers before they call
- ❖ Identify education for our providers
- ❖ Design improvements to our technology and tools.

This in turn will reduce the volume of calls thus improving the provider experience.

4.Value proposition

From the data provided, 7.6M calls costs $36M, hence the cost of 1 M calls is $4.7M. That means 1 call calls costs us $4.7 . Based on how many calls we reduce, we save dollars accordingly. Also, once we understand the cause of the problem, we can identify training opportunities for provider, design improvements for technology and tools. This in turn will reduce the number of calls and improve the provider experience. Improved provider experience in turn results into increased Net Promoter Score i.e. NPS.

5.Hypothesis

Call Claim features like denial code, claim turnaround time, claim services, and claim amount will influence the number of calls for a claim.

Ho: Combination of features of the claim is NOT influencing number of calls from provider.

H1: Combination of features of the claim is influencing number of calls from provider.

6. Exploratory data analysis (EDA)

6.1 Data quality and cleanup

In order to keep the project simple, only Claims-Call data set is considered in this learning path. The project in next learning path can include the Provider and Procedure code data sets.

Claims Call dataset contains incomplete, redundant, and noisy information as expected. There were some features that could not be treated directly since they had a high percentage of missing values. So here are steps will follow to clean up the data.

- ➢ Row selection
- ➢ Column Reduction
- ➢ Data transformation

<u>Row Selection :</u> The claim-call data set is checked for its uniqueness. All the rows are unique, hence no row is discarded at this point.

<u>Column Selection (Dimension Reduction :</u>

Claim-call dataset data profile Snap Shot

The dataset examined has the following dimensions:	
Feature	Result
Number of observations	1048575
Number of variables	28

Summary table				
Variable	**Variable class**	**# unique values**	**Missing observations**	**Any problems?**
claim_id	character	1048575	0.00%	×
final_sts_cd	character	2	0.00%	
icd_ver_cd	numeric	3	0.00%	×
diag_cd	character	11179	0.00%	×
billed_amt	numeric	10165	0.00%	×
liable_chrg_amt	numeric	12721	0.00%	×
allw_amt	numeric	39401	0.00%	×
covered_amt	numeric	37845	0.00%	×
paid_amt	numeric	37860	0.00%	×
dup_charge_amt	numeric	2543	0.00%	×
fiduciary_typ_cd	numeric	6	0.00%	×
denial_ind	numeric	2	0.00%	
first_srvc_dt	character	365	0.00%	
receive_dt	character	679	0.00%	×
adjd_dt	character	761	0.00%	×
paid_dt	character	616	0.00%	×
Tin	character	114	0.00%	×
service_npi	character	10380	0.00%	×
bill_npi	character	1055	0.00%	×
claim_location_cd	numeric	2	0.00%	
claim_place_of_srvc_cd	numeric	23	0.00%	×
claim_place_of_srvc_desc	character	23	0.00%	×
form_typ_cd	numeric	2	0.00%	
num_calls	numeric	12	98.18%	×
appeal_call_count	numeric	6	98.18%	×
claim_call_count	numeric	13	98.18%	×
total_call_time	numeric	6198	98.18%	×
first_call_dt	character	5746	98.18%	×

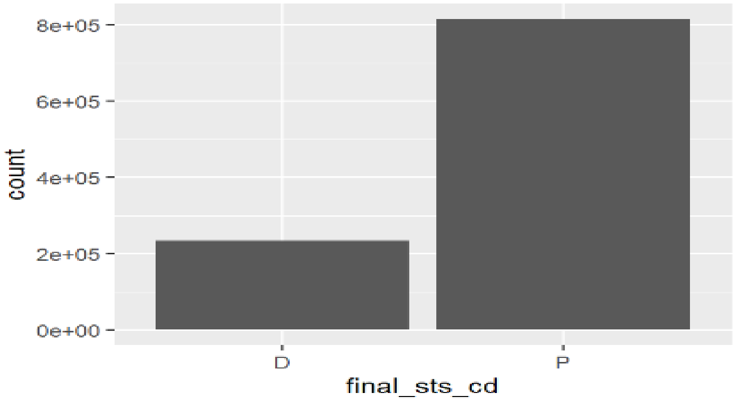
We have 28 features and each feature is analyzed further to understand its usage and decide if they are required or not for the model.

6.2 Data distribution

1. **claim_id** – Unique . This variable is a constant value for a record and doesn't impact other features or prediction – REJECTED

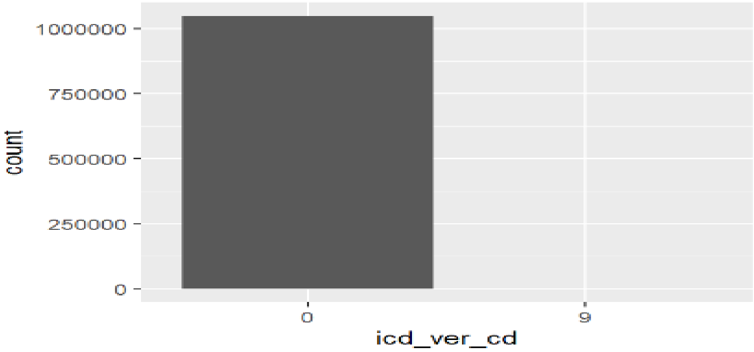
2. **final_sts_cd** - Binary variable with two values 'p' for 'paid' and 'd' for 'denied'. This is an important feature but this has collinearity with variable # 12 denial_cd , hence this variable is REJECTED and the other is kept.

final_sts_cd	
Feature	Result
Variable type	character
Number of missing obs.	0 (0 %)
Number of unique values	2
Mode	"P"



3. **icd_ver_cd** – This variable is a factor variable and has most of the codes as '0' which represents ICD10 codes. There are 35 blank values and one value as '9', which stands for 'ICD09' codes. Given the sample size of 1 M + rows, these 36 records (blank + 9) are insignificant and can be deleted.

Feature	Result
Variable type	numeric
Number of missing obs.	35 (0 %)
Number of unique values	2
Mode	"0"
Reference category	0



Now, icd_ver_cd will have constant value of '0's . Hence this variable can be REJECTED

4. **diag_cd** – The diag_cd has high cardinality as it has @ 11K unique values.

Feature	Result
Variable type	character
Number of missing obs.	35 (0 %)
Number of unique values	11178

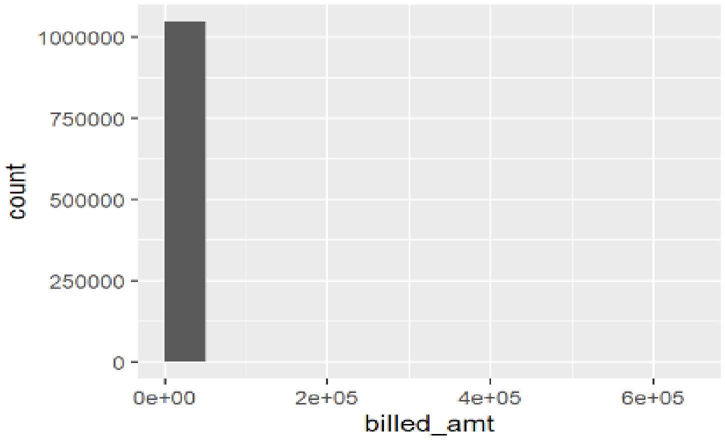
Mode	“I10”

Including high cardinality in linear technique, it may not be able to cope with huge dimensions; it leads to a model with thousands or even millions of features, thereby losing the often required comprehensibility aspect.

This variable will require to undergo data transformation by grouping it at high level code groups in section 7.3. Once the cardinality is reduced, it can be SELECTED

5. **billed_amt –** The billed_amt is a numeric field with high cardinality as it has @ 10 K unique values.

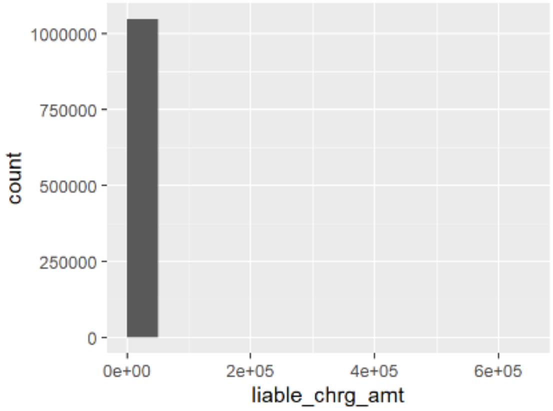
Feature	**Result**
Variable type	numeric
Number of missing obs.	0 (0 %)
Number of unique values	10165
Median	202.8
1st and 3rd quartiles	66; 360
Min. and max.	0; 627503.95



It is an important variable and can't be dropped.
This variable will have to undergo data transformation by making groups and then categorizing the amounts in certain range. This will be handled in section 7.3. Once the cardinality is reduced, it can be SELECTED

6. **liable_chrg_amt –** The liable_chrg_amt is a numeric field with high cardinality as it has @ 12 K unique values.

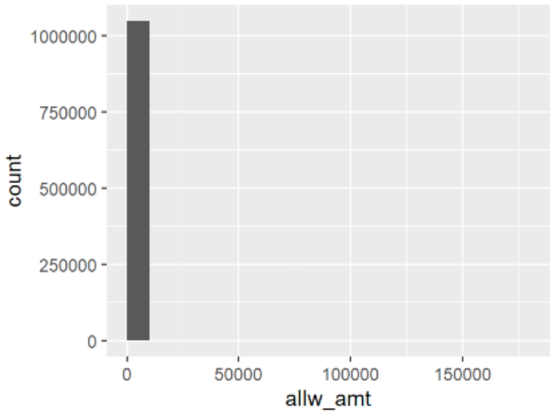
Feature	Result
Variable type	numeric
Number of missing obs.	0 (0 %)
Number of unique values	12721
Median	189.6
1st and 3rd quartiles	49.2; 360
Min. and max.	0; 627503.95



This variable has high collinearity with billed_amt. Hence this variable can be REJECTED.

7. **allw_amt- –** The allw_amt is a numeric field with high cardinality as it has @ 39 K unique values.

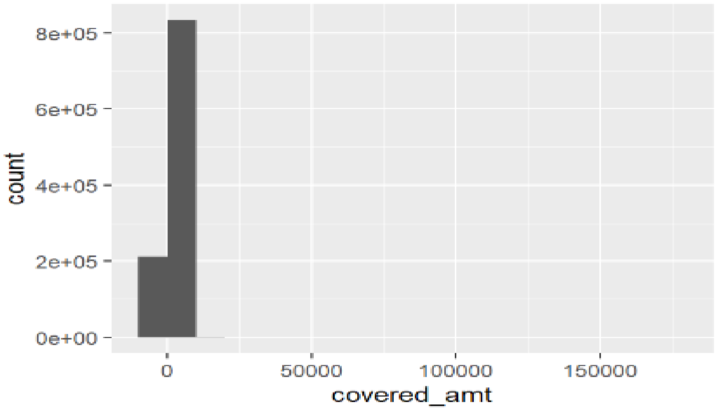
Feature	**Result**
Variable type	numeric
Number of missing obs.	0 (0 %)
Number of unique values	39401
Median	84.02
1st and 3rd quartiles	6.59; 125.22
Min. and max.	0; 172499.93



This variable has high collinearity with billed_amt. Hence this variable can be REJECTED.

8. **covered_amt –** The covered_amt is a numeric field with high cardinality as it has @ 37 K unique values

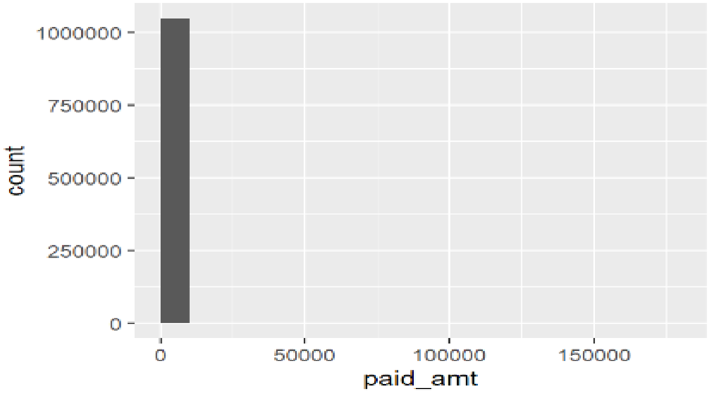
Feature	**Result**
Variable type	numeric
Number of missing obs.	0 (0 %)
Number of unique values	37845
Median	86.82
1st and 3rd quartiles	10.25; 143.1
Min. and max.	-3.24; 175976.98



This variable has high collinearity with billed_amt. Hence this variable can be REJECTED.

9. paid_amt – The paid_amt is a numeric field with high cardinality as it has @ 37 K unique values.

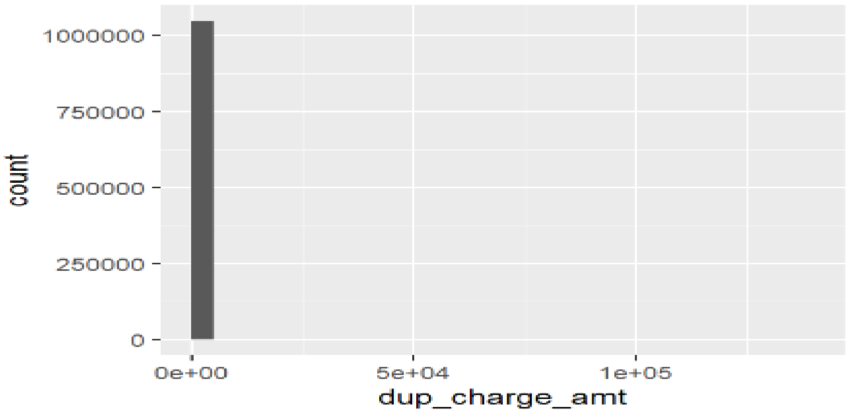
Feature	**Result**
Variable type	numeric
Number of missing obs.	0 (0 %)
Number of unique values	37860
Median	73.46
1st and 3rd quartiles	3.53; 119.86
Min. and max.	0; 170375.93



Even though, this variable has high collinearity with billed_amt, this variable has important information. The variable needs to be transformed to reduce it's cardinality and this variable can then be SELECTED.

10. **dup_charge_amt-** The dup_charge_amt is a numeric field with high cardinality as it has @ 2.5 K unique values

Feature	Result
Variable type	numeric
Number of missing obs.	0 (0 %)
Number of unique values	2543
Median	0
1st and 3rd quartiles	0; 0
Min. and max.	0; 138207.32

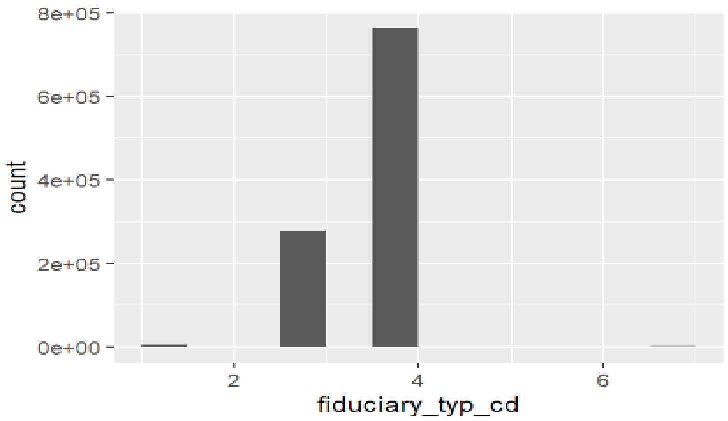


Dup_charge_amt can be one of the reasons that providers call our office. Hence this variable can't be dropped. This variable has to be treated to reduce cardinality and then it can be SELECTED

11. **fiduciary_typ_cd –** The fiduciary_typ_cd is numeric ordinal variable with 6 unique values.

Feature	Result
Variable type	numeric
Number of missing obs.	0 (0 %)
Number of unique values	6
Median	4
1st and 3rd quartiles	3; 4

Min. and max.	1; 7

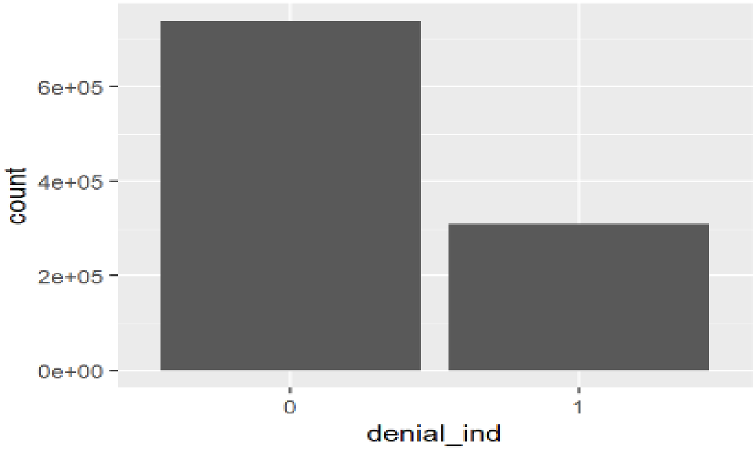


This variable has 6 unique values with value '3' and '4' having maximum occurrence.

This variable is important can be SELECTED.

12. **denial_ind –** this binary variable is indicator if the claim is denied any time in its lifecycle. It is an important feature.

Feature	Result
Variable type	numeric
Number of missing obs.	0 (0 %)
Number of unique values	2
Mode	"0"
Reference category	0



This variable can be SELECTED and final_status_cd can be ignored as both of them have collinearity.

13.first_srvc_dt
14. receive_dt
15. adjd_dt
16.paid_dt

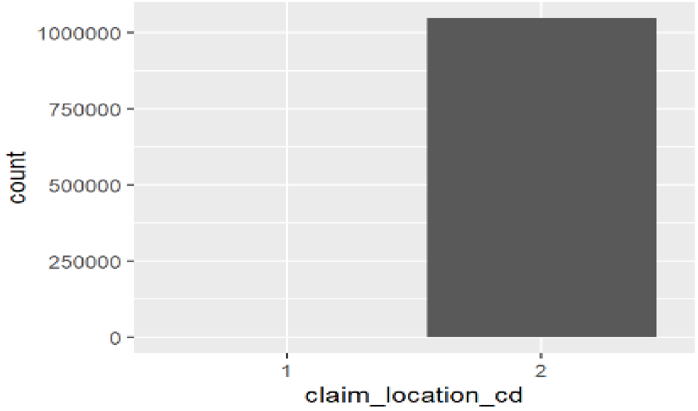
All the above dates are important but they can't be individually used. There needs to be claim turnaround time calculated from the above dates. This will take place in section 7.3 Data transformation. Individually we can consider them REJECTED.

17.tin
18.service_npi
19.bill_npi

the above variables are unique identification variables which ties the claim to the provider data set. Since at this point, we are not considering provider data set, these fields can be REJECTED.

20. **claim_location –** Claim location code is a nominal variable. Here we have 2 values of the claim location. '1' stands for facility claim and '2' stands for non-facility claims.

Feature	Result
Variable type	numeric
Number of missing obs.	0 (0 %)
Number of unique values	2
Mode	"2"
Reference category	1

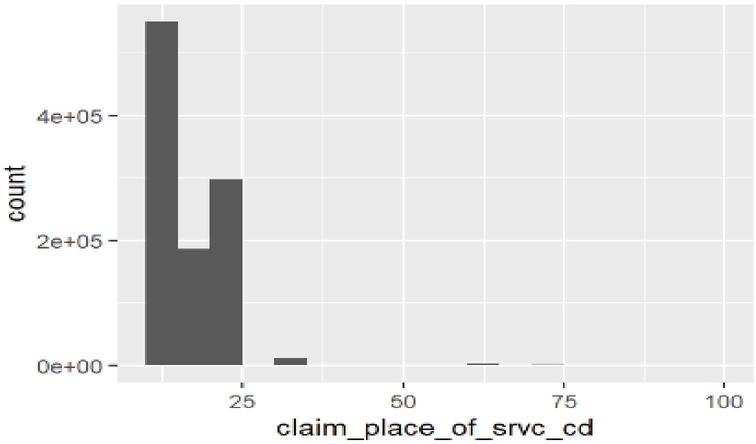


As we can see from the graph above most of the claims are from non-facility . Facility claims are handful and the data can be dropped.

Once the facility claims rows are dropped, this variable will be considered having unique values and hence be REJECTED

21. **Claim_place_of_srvc_cd –**This is a nominal variable with 23 unique values. This variable is important feature as it informs where the claim was serviced.

Feature	Result
Variable type	numeric
Number of missing obs.	0 (0 %)
Number of unique values	23
Median	11
1st and 3rd quartiles	11; 21
Min. and max.	11; 99



This variable can be SELECTED.

22. **claim_place_of_srvc_desc –** This variable is character variable which provides the description of the above claim_place_of_srvc_cd. Since we have selected claim_place_of_serv_cd, and this variable is collinear to the above variable, claim_place_of_srvc_desc can be REJECTED.

23. **form_typ_cd** has only one value, As the value remains constant, this variable can be REJECTED.

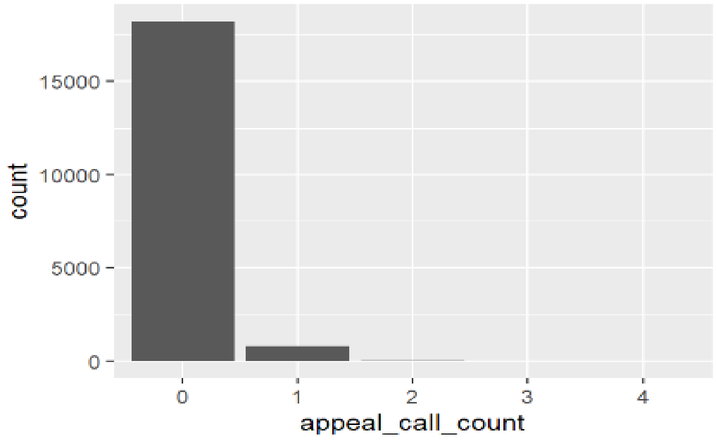
24. **Num_calls -**This numeric variable is our dependent variable. It has 98% data missing. When inquired with the source of data , it was suggested to consider blank data as '0'.

Feature	**Result**
Variable type	numeric
Number of missing obs.	1029499 (98.18 %)
Number of unique values	11
Median	1
1st and 3rd quartiles	1; 1
Min. and max.	1; 11

Even though there are 11 unique values, most of the records are with values '1' , only handful of outliers exists. This variable can be transformed into binary variable with '0' and '1' values. Data Transformation will take place in section 7.3 and then the variable can be SELECTED

25. **appeal_call_count –** This numeric variable has important information of how many calls were received due to litigation. As seen from the table below 98% data is missing. As decided in the above variable, those blank values can be considered '0' . This variable can also be transformed into binary variable.

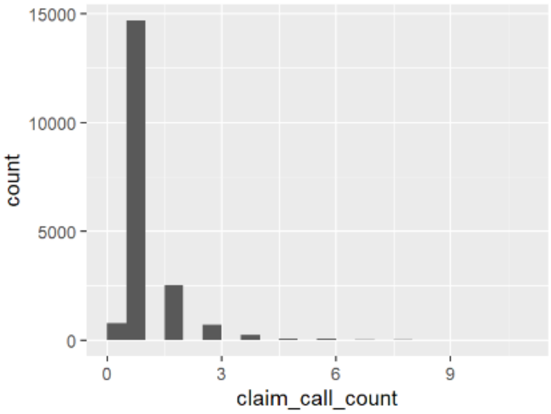
Feature	**Result**
Variable type	numeric
Number of missing obs.	1029499 (98.18 %)
Number of unique values	5
Mode	"0"
Reference category	0



After the transformation into binary variable, this variable can be SELECTED for alternate model. Currently , it is being REJECTED for the primary model due to its high collinearity with num_calls.

26. **claim_call_count** – This numeric variable has important information as to how many calls were claim related. Like the appeal_call_count, this variable also has 98% data missing.

Feature	Result
Variable type	numeric
Number of missing obs.	1029499 (98.18 %)
Number of unique values	12
Median	1
1st and 3rd quartiles	1; 1
Min. and max.	0; 11



By following the method in the above two variables, the blank values can be considered '0' . This variable can be transformed into binary variable. The variable can then be SELECTED as dependent variable for alternate model. Currently , it is being REJECTED for the primary model due to its high collinearity with num_calls.

27. **total_call_time –** This numeric variable indicates the length of the call. Since , length of call is out of scope of this study, this variable can be REJECTED.

28. **first_call_dt -** This data is not complete, there are numerous rows where the date is missing and only time is available. Hence this data will not be useful in our study and can be REJECTED.

7. Setup for analysis

7.1 Defining the variables and outcome:

Dependent Features – num_calls (dependent) variable is converted to binary hence decided to choose logistic regression model for prediction. Plan is to run logistic regression model with all qualified columns after feature reduction, apply backward elimination based on the P- value to fine tune the model.

Dependent Variable
num_calls

Independent Features - After data cleanup, analysis, Feature engineering left out with below features for further analysis.

Independent Variable
diag_cd
billed_amt
paid_amt
dup_charge_amt
fiduciary_typ_cd
denial_ind
Claim_place_of_srvc_cd

One more independent feature will be added called **call_turnaround_time** calculated from receive_dt and adj_dt .

7.2 Defining the baseline / control / denominator

Claims_Call Dataset is taking as a source for analysis. After performing cleanup and dropping rows as per the explanation in icd_ver_cd and claim_location_cd it was ended up with 1048022 records .This is our baseline for research

- **Events** : num_calls =1 , which is now a binary value, Call made by provider
- **Non-Events:** num_calls = 0 . No call made by provider.

- **Not-Events:** Calls which are not part of the denominator claims.

7.3 Data transformation / normalization as needed

Dependent variable: Dependent variable num_calls is converted to binary variable. The 98% missing value is filled with '0' and all the remaining rows having values >= 1 is transformed into '1'

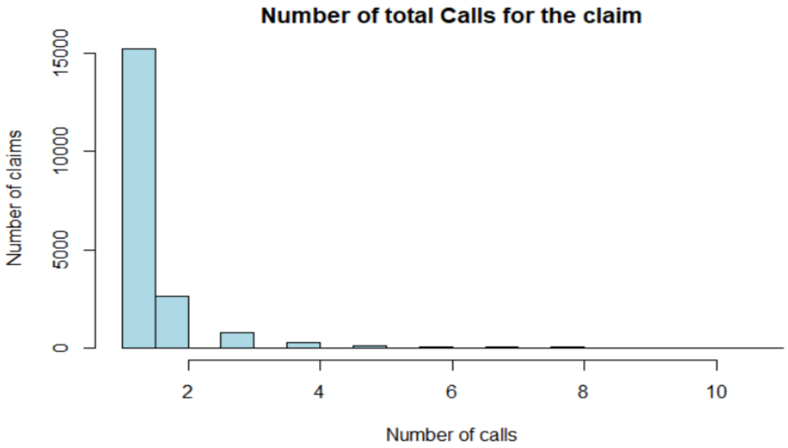


Figure 1 BEFORE

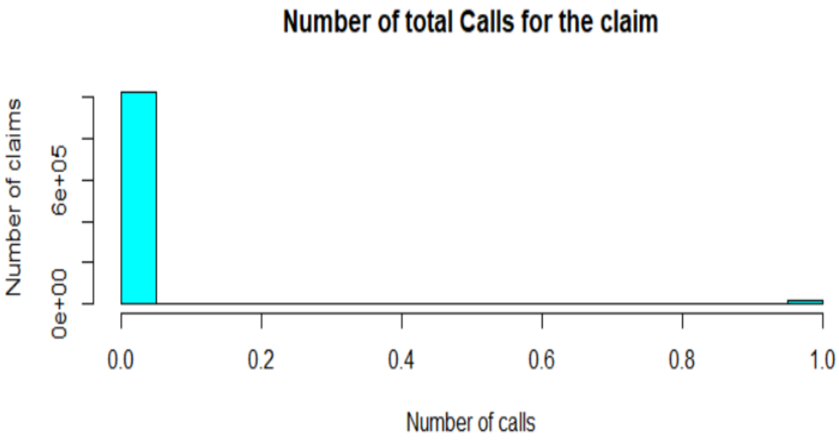
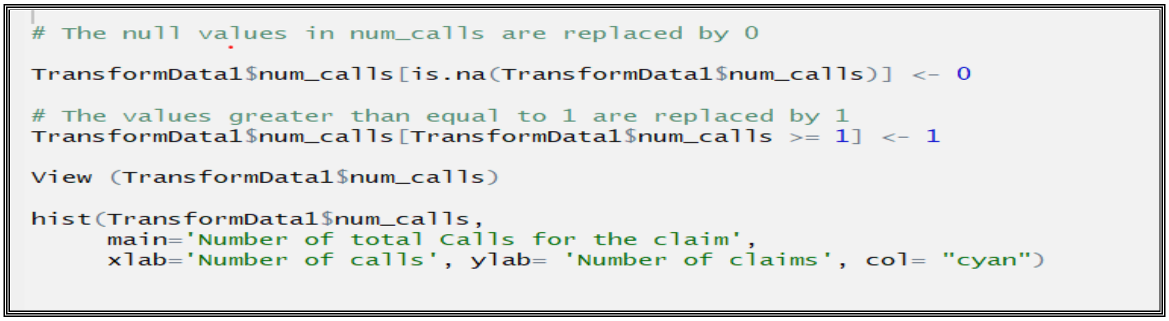


Figure 2 AFTER

Code Snippet

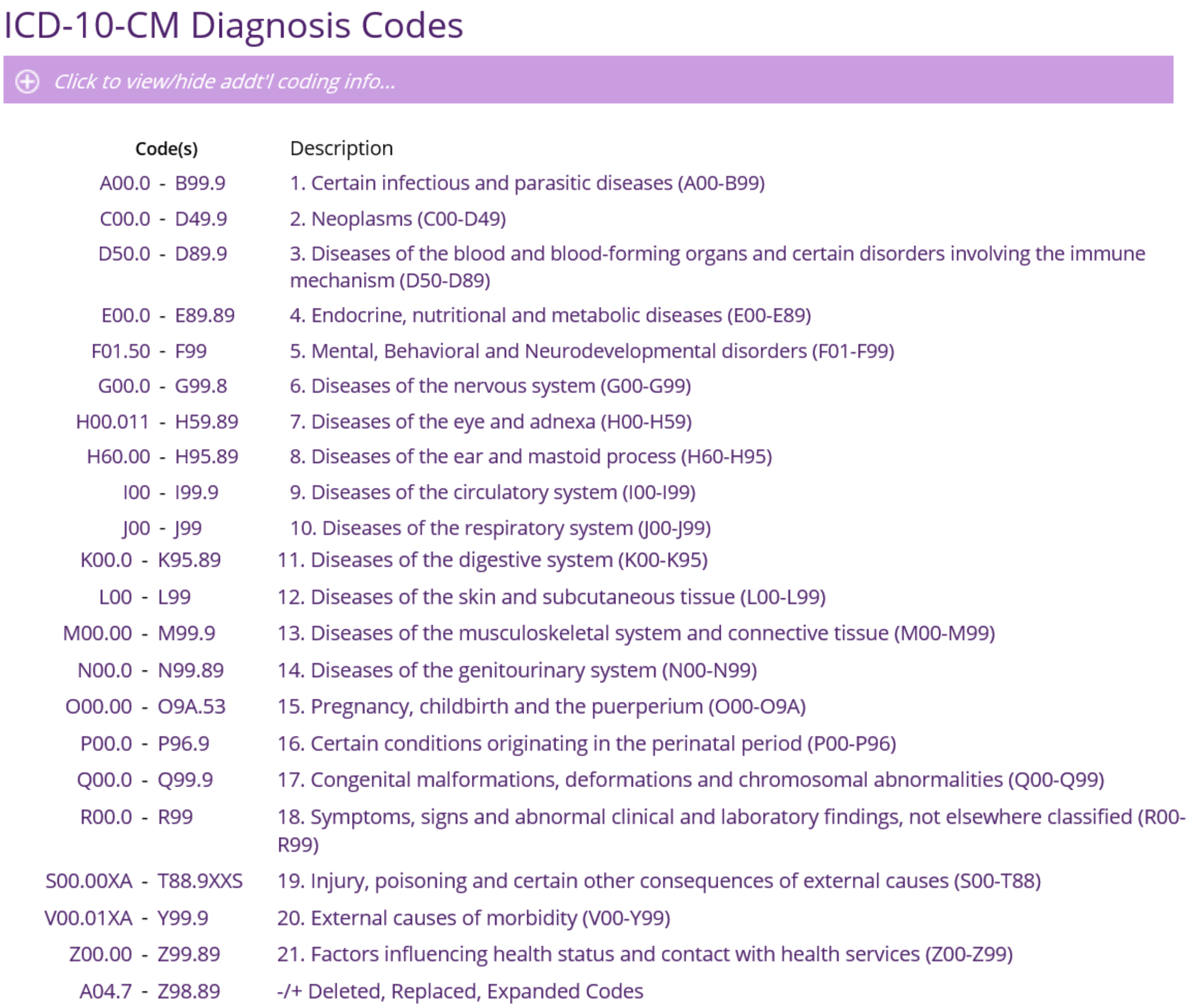


```
# The null values in num_calls are replaced by 0

TransformData1$num_calls[is.na(TransformData1$num_calls)] <- 0

# The values greater than equal to 1 are replaced by 1
TransformData1$num_calls[TransformData1$num_calls >= 1] <- 1

View (TransformData1$num_calls)

hist(TransformData1$num_calls,
     main='Number of total Calls for the claim',
     xlab='Number of calls', ylab= 'Number of claims', col= "cyan")
```

Independent Variables

*High degree categorical variable grouped next level for research and model fitments.

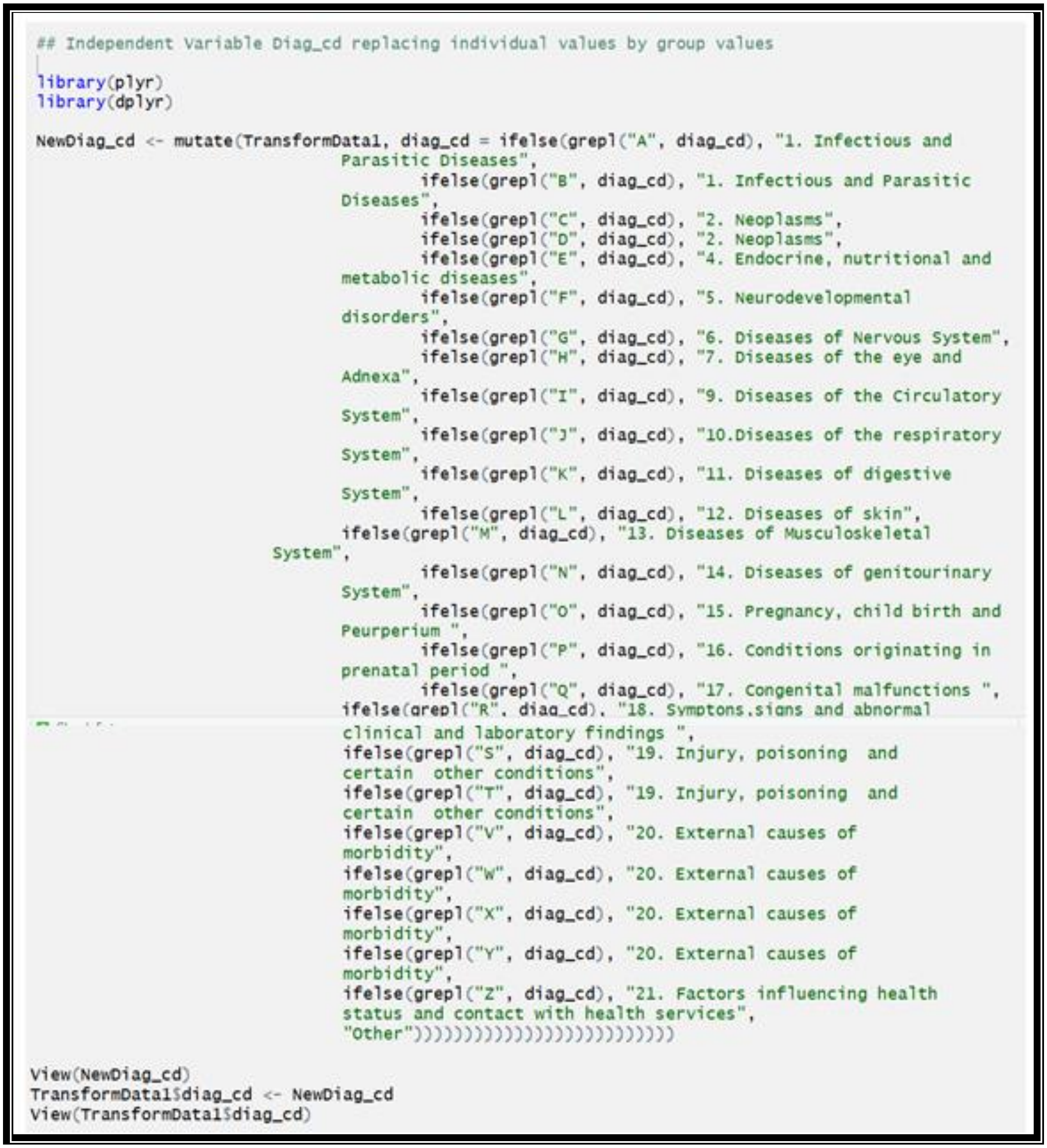
Diag_Code
The Codes are grouped as per the below mapped values.



ICD-10-CM Diagnosis Codes

Code(s)	Description
A00.0 - B99.9	1. Certain infectious and parasitic diseases (A00-B99)
C00.0 - D49.9	2. Neoplasms (C00-D49)
D50.0 - D89.9	3. Diseases of the blood and blood-forming organs and certain disorders involving the immune mechanism (D50-D89)
E00.0 - E89.89	4. Endocrine, nutritional and metabolic diseases (E00-E89)
F01.50 - F99	5. Mental, Behavioral and Neurodevelopmental disorders (F01-F99)
G00.0 - G99.8	6. Diseases of the nervous system (G00-G99)
H00.011 - H59.89	7. Diseases of the eye and adnexa (H00-H59)
H60.00 - H95.89	8. Diseases of the ear and mastoid process (H60-H95)
I00 - I99.9	9. Diseases of the circulatory system (I00-I99)
J00 - J99	10. Diseases of the respiratory system (J00-J99)
K00.0 - K95.89	11. Diseases of the digestive system (K00-K95)
L00 - L99	12. Diseases of the skin and subcutaneous tissue (L00-L99)
M00.00 - M99.9	13. Diseases of the musculoskeletal system and connective tissue (M00-M99)
N00.0 - N99.89	14. Diseases of the genitourinary system (N00-N99)
O00.00 - O9A.53	15. Pregnancy, childbirth and the puerperium (O00-O9A)
P00.0 - P96.9	16. Certain conditions originating in the perinatal period (P00-P96)
Q00.0 - Q99.9	17. Congenital malformations, deformations and chromosomal abnormalities (Q00-Q99)
R00.0 - R99	18. Symptoms, signs and abnormal clinical and laboratory findings, not elsewhere classified (R00-R99)
S00.00XA - T88.9XXS	19. Injury, poisoning and certain other consequences of external causes (S00-T88)
V00.01XA - Y99.9	20. External causes of morbidity (V00-Y99)
Z00.00 - Z99.89	21. Factors influencing health status and contact with health services (Z00-Z99)
A04.7 - Z98.89	-/+ Deleted, Replaced, Expanded Codes

11000 codes converted into 21 group categories.

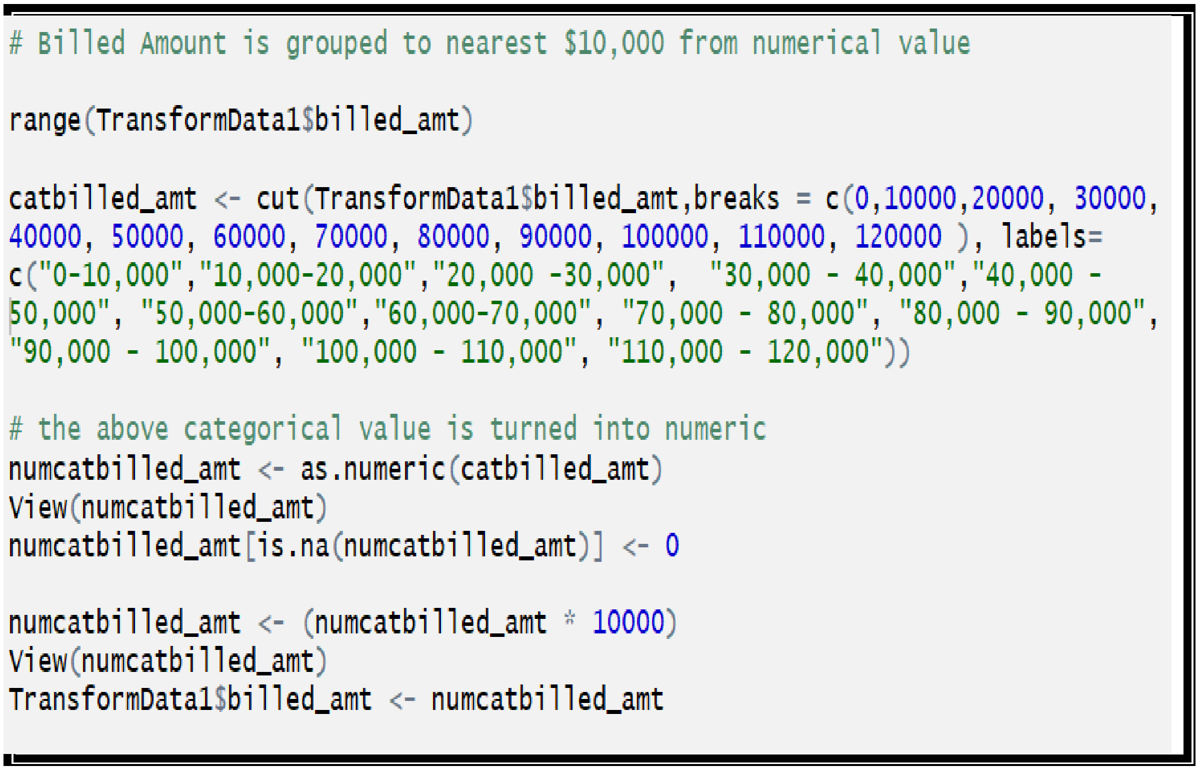


```
## Independent Variable Diag_cd replacing individual values by group values

library(plyr)
library(dplyr)

NewDiag_cd <- mutate(TransformData1, diag_cd = ifelse(grepl("A", diag_cd), "1. Infectious and
                               Parasitic Diseases",
                                       ifelse(grepl("B", diag_cd), "1. Infectious and Parasitic
                               Diseases",
                                       ifelse(grepl("C", diag_cd), "2. Neoplasms",
                                       ifelse(grepl("D", diag_cd), "2. Neoplasms",
                                       ifelse(grepl("E", diag_cd), "4. Endocrine, nutritional and
                               metabolic diseases",
                                       ifelse(grepl("F", diag_cd), "5. Neurodevelopmental
                               disorders",
                                       ifelse(grepl("G", diag_cd), "6. Diseases of Nervous System",
                                       ifelse(grepl("H", diag_cd), "7. Diseases of the eye and
                               Adnexa",
                                       ifelse(grepl("I", diag_cd), "9. Diseases of the Circulatory
                               System",
                                       ifelse(grepl("J", diag_cd), "10.Diseases of the respiratory
                               System",
                                       ifelse(grepl("K", diag_cd), "11. Diseases of digestive
                               System",
                                       ifelse(grepl("L", diag_cd), "12. Diseases of skin",
                               ifelse(grepl("M", diag_cd), "13. Diseases of Musculoskeletal
                        System",
                                       ifelse(grepl("N", diag_cd), "14. Diseases of genitourinary
                               System",
                                       ifelse(grepl("O", diag_cd), "15. Pregnancy, child birth and
                               Peurperium ",
                                       ifelse(grepl("P", diag_cd), "16. Conditions originating in
                               prenatal period ",
                                       ifelse(grepl("Q", diag_cd), "17. Congenital malfunctions ",
                               ifelse(grepl("R", diag_cd), "18. Symptons,signs and abnormal
                               clinical and laboratory findings ",
                               ifelse(grepl("S", diag_cd), "19. Injury, poisoning  and
                               certain  other conditions",
                               ifelse(grepl("T", diag_cd), "19. Injury, poisoning  and
                               certain  other conditions",
                               ifelse(grepl("V", diag_cd), "20. External causes of
                               morbidity",
                               ifelse(grepl("W", diag_cd), "20. External causes of
                               morbidity",
                               ifelse(grepl("X", diag_cd), "20. External causes of
                               morbidity",
                               ifelse(grepl("Y", diag_cd), "20. External causes of
                               morbidity",
                               ifelse(grepl("Z", diag_cd), "21. Factors influencing health
                               status and contact with health services",
                               "Other"))))))))))))))))))))))))))


View(NewDiag_cd)
TransformData1$diag_cd <- NewDiag_cd
View(TransformData1$diag_cd)
```

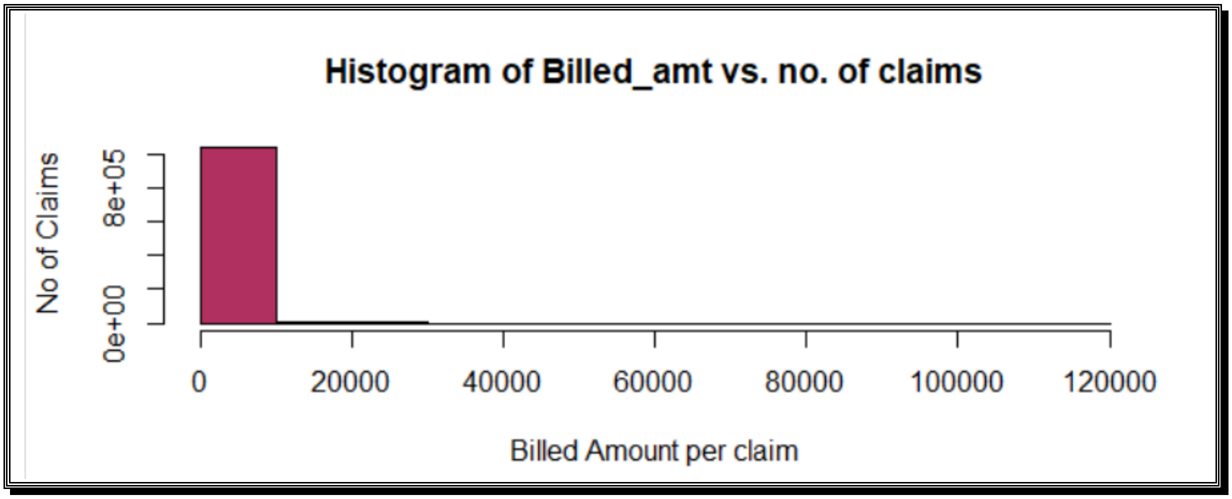
Billed_amt
Paid_amt

Dup_charge_amt

All the above features are categorized into groups to reduce the cardinality.

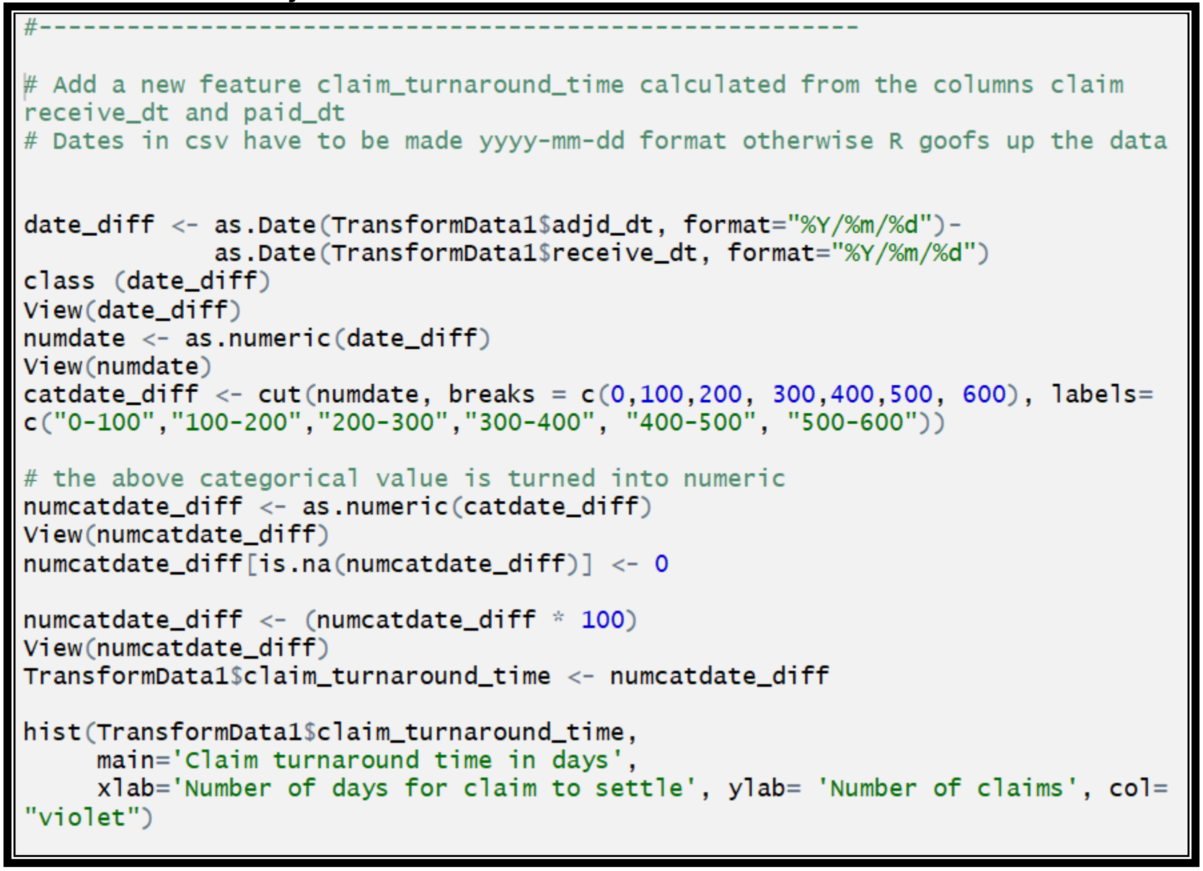


```
# Billed Amount is grouped to nearest $10,000 from numerical value

range(TransformData1$billed_amt)

catbilled_amt <- cut(TransformData1$billed_amt,breaks = c(0,10000,20000, 30000,
40000, 50000, 60000, 70000, 80000, 90000, 100000, 110000, 120000 ), labels=
c("0-10,000","10,000-20,000","20,000 -30,000",  "30,000 - 40,000","40,000 -
50,000", "50,000-60,000","60,000-70,000", "70,000 - 80,000", "80,000 - 90,000",
"90,000 - 100,000", "100,000 - 110,000", "110,000 - 120,000"))

# the above categorical value is turned into numeric
numcatbilled_amt <- as.numeric(catbilled_amt)
View(numcatbilled_amt)
numcatbilled_amt[is.na(numcatbilled_amt)] <- 0

numcatbilled_amt <- (numcatbilled_amt * 10000)
View(numcatbilled_amt)
TransformData1$billed_amt <- numcatbilled_amt
```

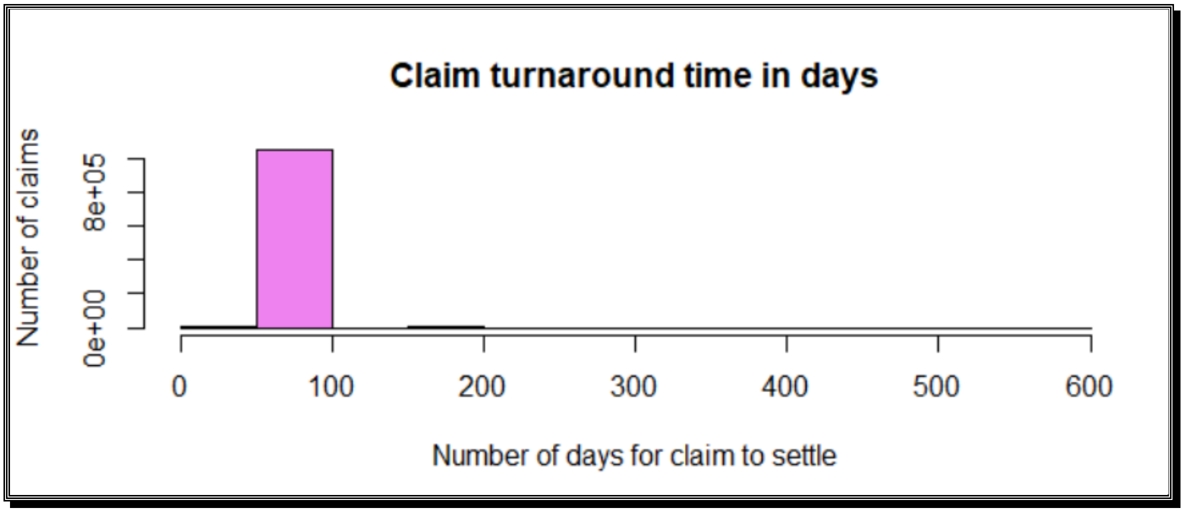


Claim_turn around_time :

Add a new feature claim_turnaround_time which will be numeric - number of days derived from calculating the days between claim receive_dt and adj_dt.



```
#-------------------------------------------------------

# Add a new feature claim_turnaround_time calculated from the columns claim
receive_dt and paid_dt
# Dates in csv have to be made yyyy-mm-dd format otherwise R goofs up the data


date_diff <- as.Date(TransformData1$adjd_dt, format="%Y/%m/%d")-
             as.Date(TransformData1$receive_dt, format="%Y/%m/%d")
class (date_diff)
View(date_diff)
numdate <- as.numeric(date_diff)
View(numdate)
catdate_diff <- cut(numdate, breaks = c(0,100,200, 300,400,500, 600), labels=
c("0-100","100-200","200-300","300-400", "400-500", "500-600"))

# the above categorical value is turned into numeric
numcatdate_diff <- as.numeric(catdate_diff)
View(numcatdate_diff)
numcatdate_diff[is.na(numcatdate_diff)] <- 0

numcatdate_diff <- (numcatdate_diff * 100)
View(numcatdate_diff)
TransformData1$claim_turnaround_time <- numcatdate_diff

hist(TransformData1$claim_turnaround_time,
     main='Claim turnaround time in days',
     xlab='Number of days for claim to settle', ylab= 'Number of claims', col=
"violet")
```

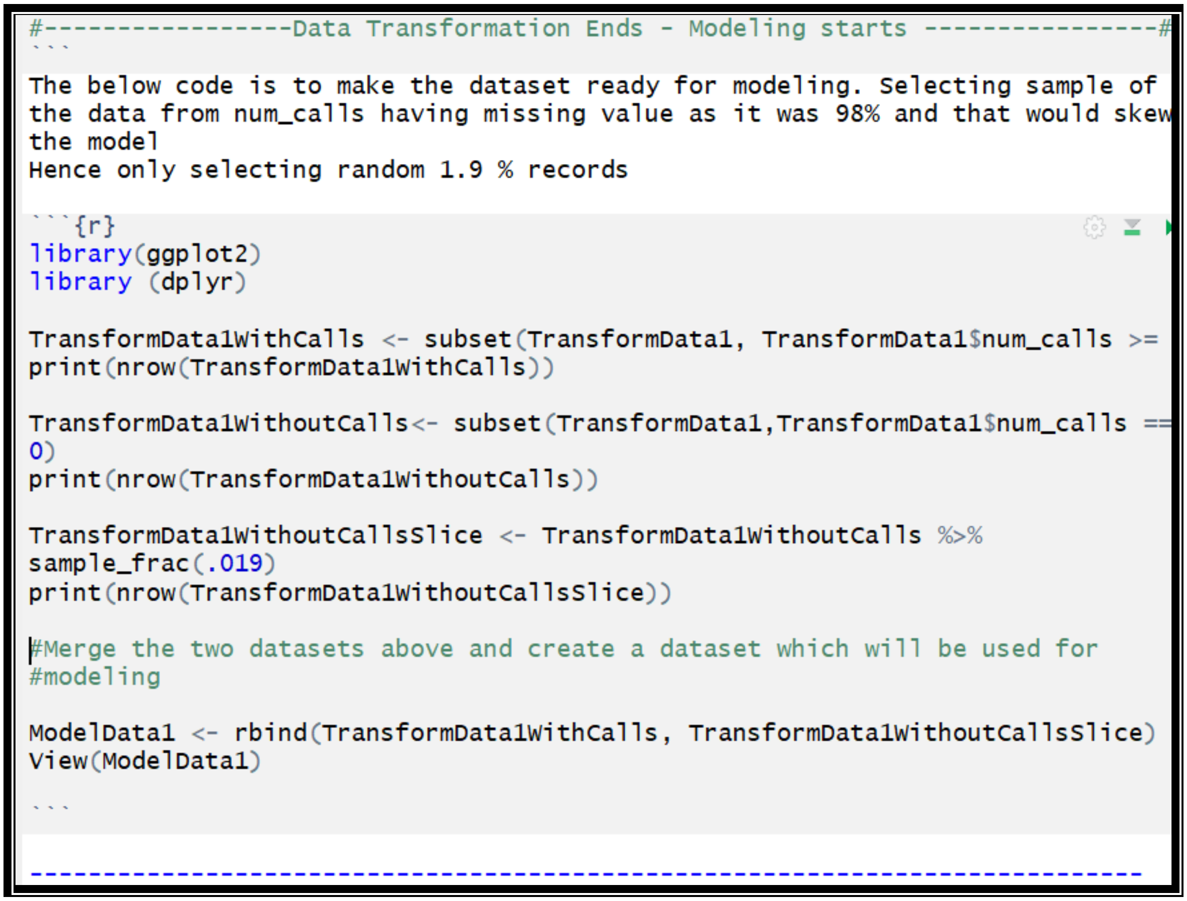


This completes the data transformation. After performing data transformation categorical variable high degree were reduced.

Features	Unique Occurrence
num_calls	2
diag_cd	21
billed_amt	12
paid_amt	7
dup_charge_amt	12
fiduciary_typ_cd	6
denial_ind	2
Claim_place_of_srvc_cd	23
claim_turnaround_time	7

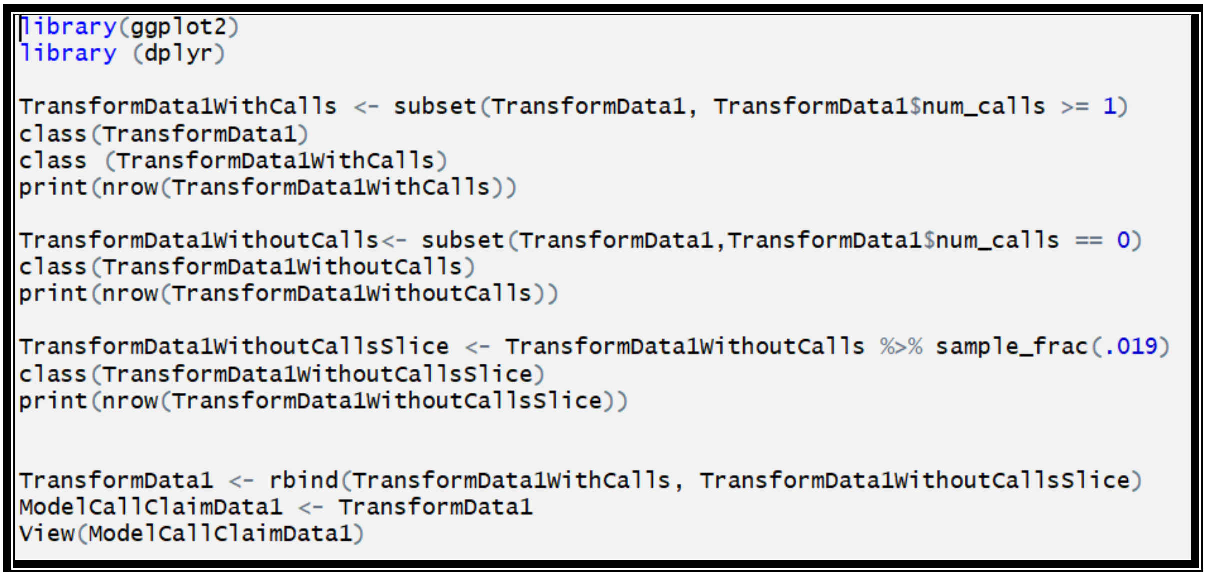
7.5 Rational for method

Logistic Regression supervised model selected to predict the num_calls as dependent variable classified as binary. Like all regression analyses, the logistic regression is a predictive analysis. Logistic Regression is used to describe data and to explain the relationship between one dependent binary variable and one or more nominal, ordinal, interval or ratio-level independent variables.



```
#-----------------Data Transformation Ends - Modeling starts ----------------#
```
~~~

The below code is to make the dataset ready for modeling. Selecting sample of
the data from num_calls having missing value as it was 98% and that would skew
the model
Hence only selecting random 1.9 % records

```{r}
library(ggplot2)
library (dplyr)

TransformData1WithCalls <- subset(TransformData1, TransformData1$num_calls >=
print(nrow(TransformData1WithCalls))

TransformData1WithoutCalls<- subset(TransformData1,TransformData1$num_calls ==
0)
print(nrow(TransformData1WithoutCalls))

TransformData1WithoutCallsSlice <- TransformData1WithoutCalls %>%
sample_frac(.019)
print(nrow(TransformData1WithoutCallsSlice))

#Merge the two datasets above and create a dataset which will be used for
#modeling

ModelData1 <- rbind(TransformData1WithCalls, TransformData1WithoutCallsSlice)
View(ModelData1)

~~~

---------------------------------------------------------------------
```

7.4 Sample Data for Model

(Training data and Test Data)

Under sampling is the technique used to handle imbalanced classification i.e. to adjust the class distribution of a data set (i.e. the ratio between the claims with calls and no calls represented).
The 2% data is randomly selected from 98% data of the claims with no calls. Hence there is balance in the data between claims with calls and claims with no calls.
```



```
library(ggplot2)
library (dplyr)

TransformData1WithCalls <- subset(TransformData1, TransformData1$num_calls >= 1)
class(TransformData1)
class (TransformData1WithCalls)
print(nrow(TransformData1WithCalls))

TransformData1WithoutCalls<- subset(TransformData1,TransformData1$num_calls == 0)
class(TransformData1WithoutCalls)
print(nrow(TransformData1WithoutCalls))

TransformData1WithoutCallsSlice <- TransformData1WithoutCalls %>% sample_frac(.019)
class(TransformData1WithoutCallsSlice)
print(nrow(TransformData1WithoutCallsSlice))


TransformData1 <- rbind(TransformData1WithCalls, TransformData1WithoutCallsSlice)
ModelCallClaimData1 <- TransformData1
View(ModelCallClaimData1)
```

K-Split fold Cross Validation will be used for splitting the dataset into train and test. This approach involves randomly dividing the set of observations into k groups, or folds, of approximately equal size. The first fold is treated as a validation set, and the method is fit on the remaining k − 1 folds. This approach involves randomly dividing the set of observations into k groups, or folds, of approximately equal size. The first fold is treated as a validation set, and the method is fit on the remaining k − 1 folds.

7.5 Rational for Model

Logistic Regression supervised model selected to predict the call from a provider as dependent variable classified as binary. Like all regression analyses, the logistic regression is a predictive analysis. Logistic Regression is used to describe data and to explain the relationship between one dependent binary variable and one or more nominal, ordinal, interval or ratio-level independent variables.

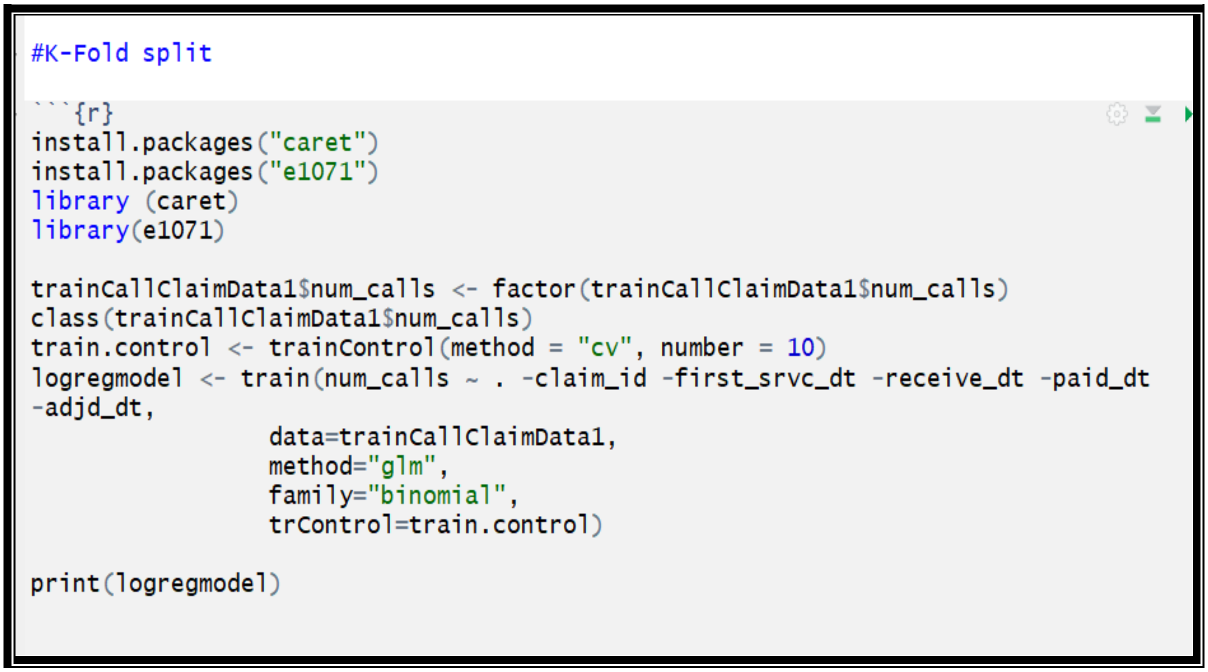
8. Model – Creation and Validation

8.1 Approach

K-fold split cross validation technique will be used .
A value of k=10 is very common in the field of applied machine learning, and is recommended. There is a bias-variance trade-off associated with the choice of k in k-fold cross-validation. Typically, given these considerations, one performs k-fold cross-validation using k = 5 or k = 10, as these values have been shown empirically to yield test error rate estimates that suffer neither from excessively high bias nor from very high variance.

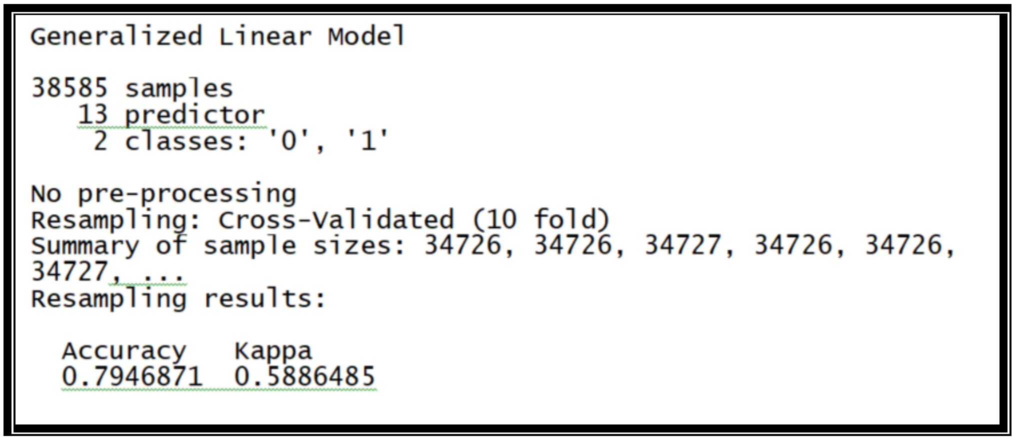
8.2 Modeling

K-fold split cross validation will be used to create the splits. Logical regression function is used to train and test the model with the logical regression.



~~~
#K-Fold split

```{r}
install.packages("caret")
install.packages("e1071")
library (caret)
library(e1071)

trainCallClaimData1$num_calls <- factor(trainCallClaimData1$num_calls)
class(trainCallClaimData1$num_calls)
train.control <- trainControl(method = "cv", number = 10)
logregmodel <- train(num_calls ~ . -claim_id -first_srvc_dt -receive_dt -paid_dt
-adjd_dt,
                 data=trainCallClaimData1,
                 method="glm",
                 family="binomial",
                 trControl=train.control)

print(logregmodel)
~~~

9. Summary of analysis result and explanation



```
Generalized Linear Model

38585 samples
   13 predictor
    2 classes: '0', '1'

No pre-processing
Resampling: Cross-Validated (10 fold)
Summary of sample sizes: 34726, 34726, 34727, 34726, 34726,
34727, ...
Resampling results:

  Accuracy   Kappa
  0.7946871  0.5886485
```

Accuracy is the percentage of correctly classifies instances out of all instances. Where else **Kappa** or Cohen's Kappa is like classification accuracy, except that it is normalized at the baseline of random chance on the dataset.

For this project the call_claim dataset is used. We have done under sampling, It has a class break down of 25% to 75% for negative and positive outcomes.

You can see that the accuracy of the model is approximately 79% which is 4 percentage points above the baseline accuracy of 75% which is not very impressive. The Kappa the other hand shows approximately 58% which is more interesting. It can be understood better with confusion matrix.

Confusion Matrix

	Predicted NO	Predicted YES
Actual :NO	0 (FN)	0 (FP)
Actual : YES	93 (TN)	4758 (TP)

Observation:

TN (True negative) I.e. we predicted no call from provider and it didn't get a call from provider. Prediction is very accurate (Specificity ~1)

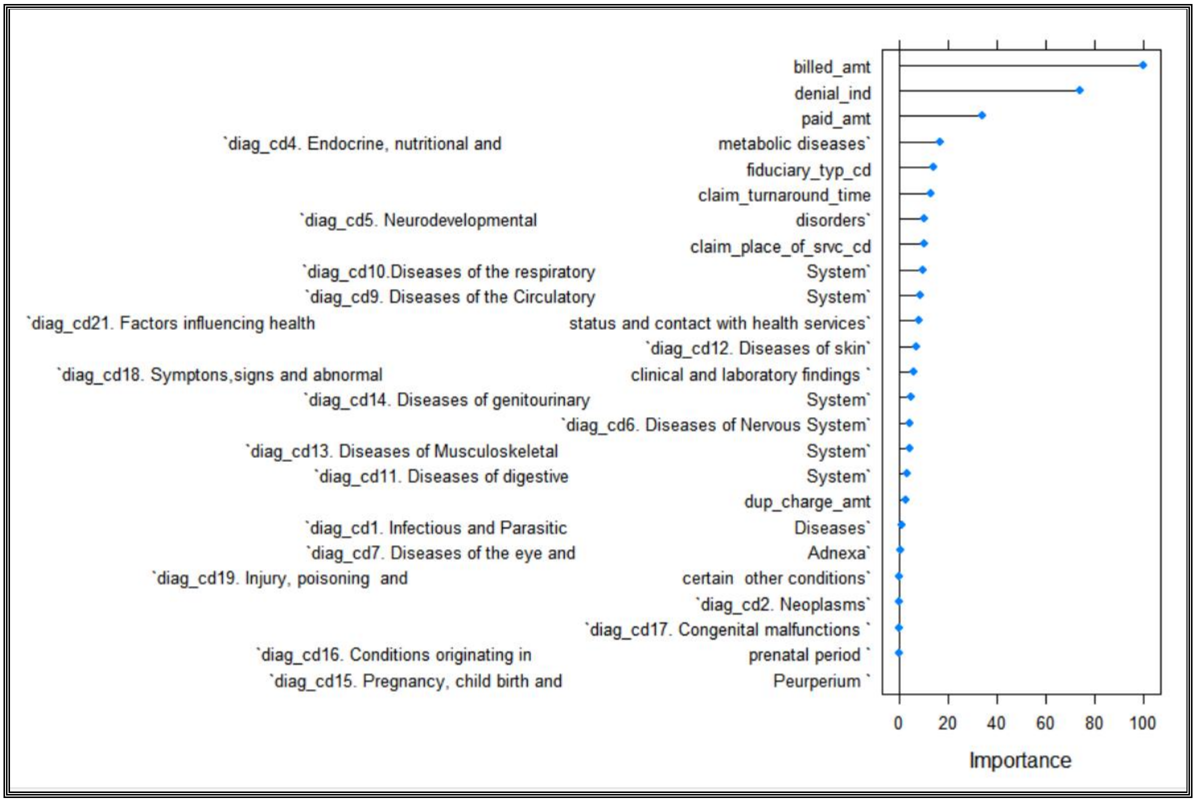
TP (True Positive) i.e. we predicted the call will come and it actually came. Prediction is 48% which is good.

Both FP (False Positive) and FN (False Negative) is zero and that is very ideal situation.

i.e. we predict provider will call and they didn't call,
we predict provider will not call and the call.

Conclusion: This is might be because of class imbalance.

IMPORTANCE CHART



The above importance chart shows that the below 4 features are the most influential features for the prediction of the call from providers.

1. Billed_amt
2. Denial_ind
3. Paid_amt
4. Diag_cd - 4 Endocrine, nutritional and metabolic diseases

10. Conclusion

The independent features billed_amt ,denial_ind, paid_amt and diag_cd has the highest impact on the call prediction. The Hypothesis H1: Combination of features of the claim is influencing number of calls from provider can be accepted.

11. Ethical Considerations

Per the information supplied with the original dataset, the original Baker challenge research is based on a preexisting HIPAA compliant dataset. The original dataset does not contain any personal identifiers and therefore is not considered human subject research.

12. Opportunity for improvement and further investigation

Performance Improvement Plan

1. Perform a model with oversampling of data and check the performance.
2. Remove the outlier values from the Model data.
3. Can do model comparison with decision tree model, SVM etc.

References

https://www.cms.gov/Medicare/Coding/ICD10/Downloads/2017-ICD-10-CM-Guidelines.pdf

https://www.findacode.com/code-set.php?set=ICD10CM
https://stackoverflow.com

https://www.analyticsvidhya.com/blog/2016/03/practical-guide-deal-imbalanced-classification-problems/#targetText=Therefore%2C%20an%20imbalanced%20classification%20problem,data%20set%20with%20100%2C000%20observations

https://beckernick.github.io/oversampling-modeling/

https://towardsdatascience.com/train-validation-and-test-sets-72cb40cba9e7

https://www.analyticsvidhya.com/blog/2016/12/practical-guide-to-implement-machine-learning-with-caret-package-in-r-with-practice-problem/#targetText=Predictions%20using%20Caret,prob%E2%80%9D%20or%20%E2%80%9Craw%E2%80%9D.