**RankMiner: Predicting Phone Agent Attrition**

RankMiner

**Final Project**

**Statistical Data Mining, ISM 6137.902S16**

**April, 2016**

*Authored by* **Vishal Punjabi**

**Deepen Panchal**

**Mohammed Ihtesham**

**Paridhi Varma**

Table of Contents

[1 INTRODUCTION 1](#_Toc449737186)

[1.1 MOTIVATION 1](#_Toc449737187)

[1.2 PROBLEM STATEMENT 1](#_Toc449737188)

[1.3 SCOPE 1](#_Toc449737189)

[2 DATA EXPLORTION 2](#_Toc449737190)

[2.1 GENERAL ASSESSMENT OF DATA QUALITY 3](#_Toc449737191)

[2.2 IDENTIFICATION OF DATA QUALITY ISSUES 4](#_Toc449737192)

[2.3 GENERAL STRATEGY FOR HANDLING DATA QUALITY ISSUES 5](#_Toc449737193)

[2.4 ADVANCED DATA EXPLORATION: VISUALIZATION OF RELATIONSHIPS 6](#_Toc449737194)

[3 DATA PREPARATION AND MATCHING 12](#_Toc449737195)

[3.1 AGENT DATA AND CALL DATA 12](#_Toc449737196)

[3.2 FEATURE DATA 17](#_Toc449737197)

[4 STATISTICAL ANALYSIS AND MODELING 25](#_Toc449737198)

[4.1 AGENT AND CALL DATA 25](#_Toc449737199)

[4.2 FEATURE DATA 36](#_Toc449737200)

[5 CONCLUSION 42](#_Toc449737201)

[5.1 MANAGARIAL IMPLECATIONS OF RESULTS 42](#_Toc449737202)

[6 FUTURE WORK 43](#_Toc449737203)

[7 SOURCE CODE 44](#_Toc449737204)

[8 ACKNOWLEDGEMENT 45](#_Toc449737205)

Confidentiality Agreement

This report is classified as internal and confidential under the following agreement. Circulation of this report is restricted only to internal stakeholders within the University of South Florida and RankMiner Co.

# INTRODUCTION

## MOTIVATION

The company RankMiner is a Predictive Analytics Company focused on voice based insights. Their focus area is mainly telephony based audio and the problem statement revolves around the Agent Attrition rate for a debt collection company.

## PROBLEM STATEMENT

The problem focuses on the agent attrition based on the audio features of the calls made by each agent throughout the period of July – December. The prediction of agent attrition on the basis of the call features is the challenging aspect of the problem. The data provided to us is Agent details such as the commission earned per month, number

The target of the problem statement is the Phone Agents and to reduce their attrition rate, report performance of the data model, identify strength and weaknesses and to suggest further research to understand problem statement.

## SCOPE

Three data sets are used in this project. Data for **298** agents with **43** attributes are provided for building this model. Information of all the calls made by each of the agents are provided in a separate data set. There are **170429** call details in the data set. Call data set have ten attributes which gives more information of the call such as duration, start time, end time, call outcome etc. In addition, a third set is provided which contains further information of each call made and captures the agents behavior on that call. A total of **62245** rows of information is provided with **178** attributes is available to analyze the characteristic of each call made by an agent.

# DATA EXPLORTION

There are 3 datasets available from the RankMiner project – Agent Data, Call Data and Feature Dataset. This is the initial structure of the dataset that was identified.

1. **Call Data**

|  |  |
| --- | --- |
| **Account** | **Unique Customer Account Id** |
| Audio File Name | Name Of The File Name |
| Skill Name | The Skill The Agent Is Currently Working In |
| Call Start Time | Call Start Time |
| Call End Time | Call End Time |
| Agent Id | Unique Agent Id |
| Call Direction | Inbound Or Outbound Call (All Calls are Outbound in your DS) |
| Call Duration (HMS) | Duration Of Call |
| Filesize(kb) | Size Of The Call Audio File |
| Rec Status | Exit Status For The Call |

1. **Agent Data**

|  |  |
| --- | --- |
| **Agent\_id** | **Unique ID Of The Agent** |
| Payroll\_id\_src1,payroll\_id\_src2 | Payroll Info Sourced From Different Systems Of The Enterprise |
| Hire\_date\_src1,hire\_date\_src2 | Hire Date Info Sourced From Different Systems Of The Enterprise |
| Term\_date\_src1,term\_date\_src2 | Term Date Info Sourced From Different Systems Of The Enterprise |
| Group\_src1 | Group The Agent Has Worked In Primarily |
| Work\_shift\_src1,work\_shift\_src2 | Work Shift Info Sourced From Different Systems Of The Enterprise |
| Term\_code | Term Type + Term Reason |
| Term\_type | I/V |
| Term\_reason | Reason For Termination |
| Jul\_group | Group The Agent Worked In For The Month |
| Jul\_hrs\_worked | Hours Worked In The Month Of July |
| Jul\_hourly\_rate | Hourly Rate Of Agent That Month |
| Jul\_revenue\_generated | Revenue Of Each Agent Month Wise |
| Jul\_commission | Commission Of Each Agent Month Wise |

1. **Feature Data**

|  |  |
| --- | --- |
| **Audio File Name** | **Name Of The Audio File** |
| Target\_value | The Value If 1, The Agent Is Likely To Stay And If 0, More Likely To Leave |
| Feature\_value 1-176 | Features Recorded As The Numeric Values For Recording Emotional Status Of The Call Of The Account |

## GENERAL ASSESSMENT OF DATA QUALITY

The data in the three files was of mediocre quality. While there weren’t any grave quality issues as such, there were some small to mediocre issues which needed to be addressed. Some of which are as below:

1. Missing/blank values
2. Outliers
3. Shifted Data

## IDENTIFICATION OF DATA QUALITY ISSUES

1. **Agent Data set:**
2. **Empty values in the data set:** The data for columns payroll\_id, hire\_date, term\_date, work\_shift are pulled from two different sources. There is a possible discrepancy between the two sources. For a single row, source values contain value while source 2 is empty.
3. Data present in the dataset are in mixed case (Upper and Lower). Though the text is same for most of data but due to mixed case they may cause issues during statistical analysis.
4. Dec\_group columns has an outlier “102474” and is removed from the dataset.
5. There are significant number of Work groups fields which have value “#N/A” for agents and months.
6. A single work group has different code and description. For example, "INB DSH” and "DSH" refer to the same DISH group. To identify groups which mean the same the values for these groups are updated to a more meaningful and uniform codes.
7. The data type of revenue generated for some months are factors and not numeric.
8. **Feature Data set:**
9. Audio file name has an outlier with value “62243” and is removed from data set.
10. Target value of some audio files are blank.
11. **Call Data set:**
12. A lot of fields in the data set are blanks. All these blanks are replaced by NA.
13. Text fields are in mixed case. Though the text is same for most of data but due to mixed case they are counted as distinct by the software. Thus, to reduce the count, all the text data is changed to upper case.
14. Columns in the data set are dot separated and thus pose a risk of being interpreted as a function when running R code.
15. The Skill name column has a number of text description which refer to the same skill group. For example – “AT and T\_HCI”, “AT and T B\_HCI” and “TMobile\_HCI” , Tmobile Tiertary A\_HCI refer the same group. To identify groups which refer the same entity are updated with a more meaningful and uniform codes.

## GENERAL STRATEGY FOR HANDLING DATA QUALITY ISSUES

The first step performed to handle data quality was a complete analysis of the current state of all three data sets. Information with errors, inconsistencies, duplicates or missing fields are updated with values that do not change the context of the information.

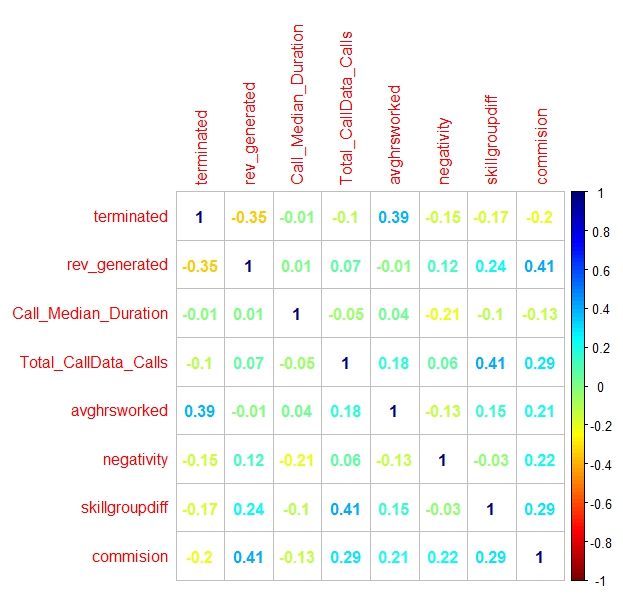
The assumptions made to tackle data issue for each data set are as follows:

1. The columns for which data is coming from two different data sources, source 1 is assumed to be authentic and wherever there is discrepancy, the values are overwritten with values from source 1.
2. Text data present in all data sets such as skill names, skill group etc are in mixed case. Thus to reduce inconsistencies all textual data are converted to upper case.
3. The work groups having values as “#N/A” are replaced with “OTHER” work group to handle missing values and not lost agents from analysis.
4. Groups and sub groups under whom the agent is working has inconsistent values such as "INB DSH”,"DSH" – which mean DISH,"PTM","VZ" – stands for VERIZON,"INB SPR","SPR" – is same as SPRINT. Inconsistencies like this are handled by replacing it with a single value for same sub groups but different codes.
5. Dots have been removed from column names with underscore so that it does not give errors while running R code.
6. Skill name in call data set are inconsistent and similarly as for agent data set these values can be grouped together and assigned a more meaningful name.

## ADVANCED DATA EXPLORATION: VISUALIZATION OF RELATIONSHIPS

### VISUALIZATION 1

Below is a correlation matrix, this matrix shows the relationship between every predictor identified for this project. The matrix presents a numerical relationship between them thus it can be easily identified which predictors have a direct and which one have indirect relationship among them.



**R Code:**

#correlation matrix

cor**(**agent\_trimmed2**[**,c**(**"terminated", "rev\_generated","Call\_Median\_Duration","Total\_CallData\_Calls","avghrsworked","negativity","skillgroupdiff","commision"**)])**;

##visualizing the correlation matrix

library**(**corrplot**)**;

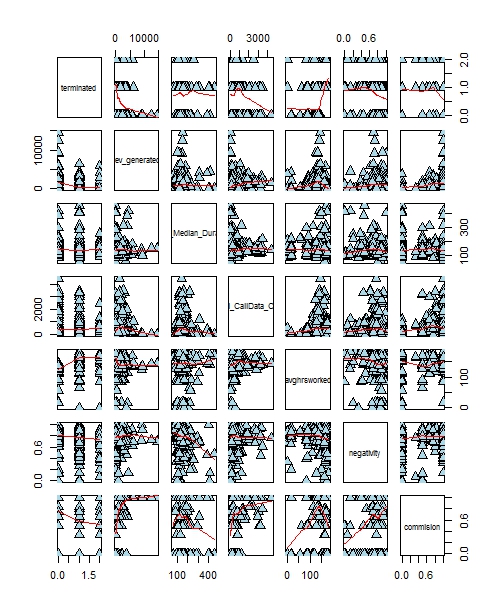
cor.agent\_trimmed2 **<-** cor**(**agent\_trimmed2**[**,c**(**"terminated", "rev\_generated","Call\_Median\_Duration","Total\_CallData\_Calls","avghrsworked","negativity","skillgroupdiff","commision"**)])**;

col1 **<-**colorRampPalette**(**c**(**"#7F0000","red","#FF7F00","yellow","#7FFF7F", "cyan", "#007FFF", "blue","#00007F"**))**

corrplot**(**cor.agent\_trimmed2, method **=** "number", col **=** col1**(**100**))**

### Visualization 2:

Another way to visualize correlation matrix is by running a scatterplot matrix.



**R Code:**

library**(**YaleToolkit**)**;

pairs**(**agent\_trimmed2**[**,c**(**"terminated", "rev\_generated","Call\_Median\_Duration","Total\_CallData\_Calls","avghrsworked","negativity"**)]**, upper.pars**=**list**(**scatter**=**"stats"**)**,panel **=** panel.smooth, cex **=** 1.5, pch **=** 24, bg **=** "light blue"**)**;

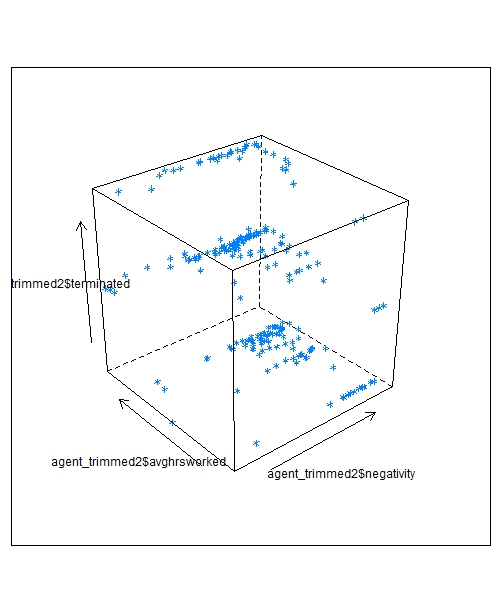
### 

**R Code:**

splom**(**agent\_trimmed2**[**,c**(**"terminated", "rev\_generated","Call\_Median\_Duration","Total\_CallData\_Calls","avghrsworked","commision"**)]**, main**=**"Agent Data"**)**

### Visualization 3:

The below plot shows that agents who work more average number of hours are more probable to left voluntarily. An important insight also from the below plot shows that agents who are working long hours are having a negative outcome from calls.



**R Code:**

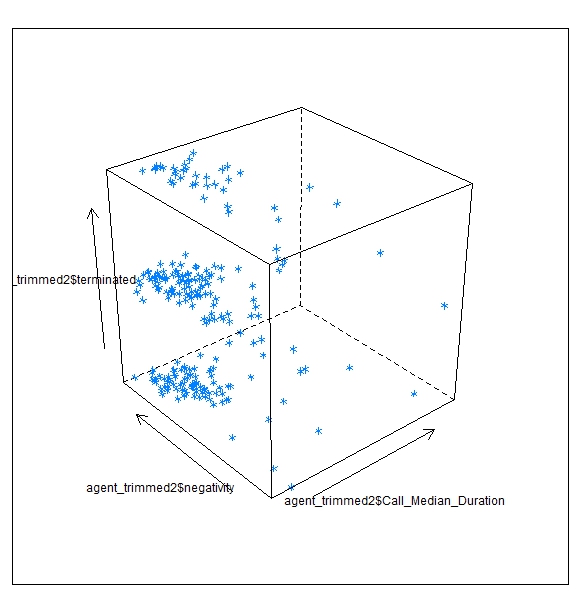
library**(**lattice**)**

cloud**(**agent\_trimmed2**$**terminated**~**agent\_trimmed2**$**negativity**+**agent\_trimmed2**$**avghrsworked**)**

cor**(**agent\_trimmed2**[**,c**(**"terminated", "negativity","avghrsworked"**)])**;

### Visualization 4:

The plot depicts that agents who exhibited negative emotions on calls were either working or left voluntarily.



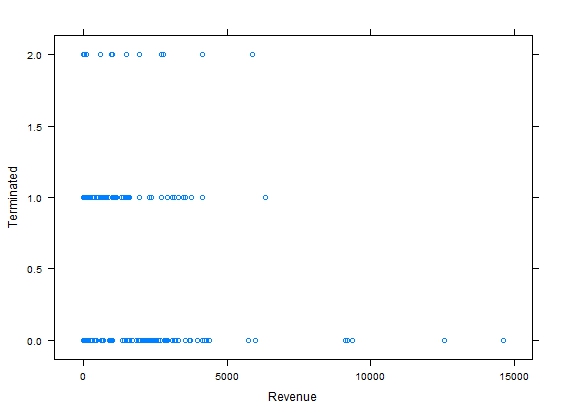
**R Code:**

cloud**(**agent\_trimmed2**$**terminated**~**agent\_trimmed2**$**Call\_Median\_Duration**+**agent\_trimmed2**$**negativity**)**

### Visualization 5:

This plot establishes relationship between agents who were terminated whether voluntarily or involuntarily and the revenue generated by them. From the graph, it can be confirmed that

agents who left the company voluntarily were a group who were generating lesser revenue as compared to the agents who did not leave the company.



**R Code:**

xyplot**(**agent\_trimmed2**$**terminated**~**agent\_trimmed2**$**rev\_generated, xlab **=** "Revenue", ylab **=** "Terminated"**)**;

# DATA PREPARATION AND MATCHING

## AGENT DATA AND CALL DATA

1. Payroll\_ID is dropped from the dataset, as the agent\_id identifies the agents uniquely.
2. Src1 and src2 columns are same, and are included to establish the veracity for data. The columns have been merged for the easy handling of data. If there is a contradiction, we picked src1 data.
   1. The shifts provided is considered irrelevant for this statistical analysis.
   2. Also, the number of hours worked is provided for each of the agents. Hence, the column is termed irrelevant for the final dataset.

agent**$**payroll\_id\_src1**[**is.na**(**agent**$**payroll\_id\_src1**)]** **<-** agent**$**payroll\_id\_src2**[**is.na**(**agent**$**payroll\_id\_src1**)]**

agent**$**payroll\_id\_src2**[**is.na**(**agent**$**payroll\_id\_src2**)]** **<-** agent**$**payroll\_id\_src1**[**is.na**(**agent**$**payroll\_id\_src2**)]**

agent**$**hire\_date\_src1**[**is.na**(**agent**$**hire\_date\_src1**)]** **<-** agent**$**hire\_date\_src2**[**is.na**(**agent**$**hire\_date\_src1**)]**

agent**$**hire\_date\_src2**[**is.na**(**agent**$**hire\_date\_src2**)]** **<-** agent**$**hire\_date\_src1**[**is.na**(**agent**$**hire\_date\_src2**)]**

agent**$**term\_date\_src1**[**is.na**(**agent**$**term\_date\_src2**)]** **<-** agent**$**term\_date\_src1**[**is.na**(**agent**$**term\_date\_src2**)]**

agent**$**term\_date\_src1**[**is.na**(**agent**$**term\_date\_src1**)]** **<-** agent**$**term\_date\_src2**[**is.na**(**agent**$**term\_date\_src1**)]**

agent**$**work\_shift\_src1**[**is.na**(**agent**$**work\_shift\_src1**)]** **<-** agent**$**work\_shift\_src2**[**is.na**(**agent**$**work\_shift\_src1**)]**

agent**$**work\_shift\_src2**[**is.na**(**agent**$**work\_shift\_src2**)]** **<-** agent**$**work\_shift\_src1**[**is.na**(**agent**$**work\_shift\_src2**)]**

# Check if src1 and src2 are different

agent**$**payroll\_id\_src1**[**agent**$**payroll\_id\_src1**!=**agent**$**payroll\_id\_src2**]**

agent**$**payroll\_id\_src2**[**agent**$**payroll\_id\_src1**!=**agent**$**payroll\_id\_src2**]**

agent**$**hire\_date\_src1**[**agent**$**hire\_date\_src1**!=**agent**$**hire\_date\_src2 **]**

agent**$**hire\_date\_src2**[**agent**$**hire\_date\_src1**!=**agent**$**hire\_date\_src2**]**

agent**$**term\_date\_src1**[**agent**$**term\_date\_src1**!=**agent**$**term\_date\_src2 **&** **!**is.na**(**agent**$**term\_date\_src1**)** **&** **!**is.na**(**agent**$**term\_date\_src2**)]**

agent**$**term\_date\_src2**[**agent**$**term\_date\_src1**!=**agent**$**term\_date\_src2 **&** **!**is.na**(**agent**$**term\_date\_src1**)** **&** **!**is.na**(**agent**$**term\_date\_src2**)]**

agent**$**work\_shift\_src1**[**agent**$**work\_shift\_src1**!=**agent**$**work\_shift\_src2 **&** **!**is.na**(**agent**$**work\_shift\_src1**)** **&** **!**is.na**(**agent**$**work\_shift\_src2**)]**

agent**$**work\_shift\_src2**[**agent**$**work\_shift\_src1**!=**agent**$**work\_shift\_src2 **&** **!**is.na**(**agent**$**work\_shift\_src1**)** **&** **!**is.na**(**agent**$**work\_shift\_src2**)]**

#Delete irrelevant columns from src2

agent**$**payroll\_id\_src2 **=** **NULL**

agent**$**hire\_date\_src2 **=** **NULL**

agent**$**term\_date\_src2 **=** **NULL**

agent**$**work\_shift\_src2 **=** **NULL**

1. A predictor for the model is calculated based on the data available for each agent under termination date or termination type. If both values are blank, then that means the agent is working and is not terminated and vice-versa. The agents who are terminated are assigned 1 and those who are working are assigned 0.

agent**$**terminated **<-** 0

agent**$**terminated**[!**is.na**(**agent**$**term\_date**)** **|** **!**is.na**(**agent**$**term\_code**)** **|** **!**is.na**(**agent**$**term\_type**)** **|** **!**is.na**(**agent**$**term\_reason**)]** **<-** 1

1. The columns Term\_Type and Term\_Reason is also termed irrelevant for the dataset as the same information is provided to the model from the column Term\_code.
2. Hours\_worked, hourly\_rate and group for each agent every month is discarded and taken as a single value, from the mean of all the data points.
3. Revenue Generated, commission of every month is also discarded and calculated as a single dataset to generate two columns –average hours worked and negativity.
   1. Productivity - (Average hours worked/month)/total # of months

We have considered only those months where data was available.

#productivity

agent**[**,"Dec\_hrs\_worked"**]** **<-** as.numeric**(**agent**[**,"Dec\_hrs\_worked"**])**

agent**$**avghrsworked **<-** **NA**

agent**$**avghrsworked **<-** rowSums**(**agent**[**,c**(**"Jul\_hrs\_worked","Aug\_hrs\_worked","Sep\_hrs\_worked","Oct\_hrs\_worked","Nov\_hrs\_worked","Dec\_hrs\_worked"**)]**,na.rm **=** **TRUE)**

agent**$**avghrsworked **<-** agent**$**avghrsworked**/**agent**$**nom

#rev\_generated

agent**$**Oct\_revenue\_generated**[**which**(**agent**$**Oct\_revenue\_generated**==**"#N/A"**)]=NA**

agent**$**Sep\_revenue\_generated**[**which**(**agent**$**Sep\_revenue\_generated**==**"#N/A"**)]=NA**

agent**[**,"Oct\_revenue\_generated"**]** **<-** as.numeric**(**agent**[**,"Oct\_revenue\_generated"**])**

agent**[**,"Sep\_revenue\_generated"**]** **<-** as.numeric**(**agent**[**,"Sep\_revenue\_generated"**])**

agent**$**rev\_generated **<-** **NA**

agent**$**rev\_generated **<-** rowSums**(**agent**[**,c**(**"Jul\_revenue\_generated","Aug\_revenue\_generated","Sep\_revenue\_generated","Oct\_revenue\_generated","Nov\_revenue\_generated","Dec\_revenue\_generated"**)]**,na.rm **=** **TRUE)-**rowSums**(**agent**[**,c**(**"Jul\_commission","Aug\_commission","Sep\_commission","Oct\_commission","Nov\_commission","Dec\_commission"**)]**,na.rm **=** **TRUE)**

agent**$**rev\_generated **<-** round**(**agent**$**rev\_generated**/**agent**$**nom,3**)**

#Calculation of commission for each month of an agent -

agent**$**commision **<-** 0

agent**$**commision **<-** ifelse**(!**is.na**(**agent**$**Jul\_commission**)**,agent**$**commision**+**1,agent**$**commision**)**

agent**$**commision **<-** ifelse**(!**is.na**(**agent**$**Aug\_commission**)**,agent**$**commision**+**1,agent**$**commision**)**

agent**$**commision **<-** ifelse**(!**is.na**(**agent**$**Sep\_commission**)**,agent**$**commision**+**1,agent**$**commision**)**

agent**$**commision **<-** ifelse**(!**is.na**(**agent**$**Oct\_commission**)**,agent**$**commision**+**1,agent**$**commision**)**

agent**$**commision **<-** ifelse**(!**is.na**(**agent**$**Nov\_commission**)**,agent**$**commision**+**1,agent**$**commision**)**

agent**$**commision **<-** ifelse**(!**is.na**(**agent**$**Dec\_commission**)**,agent**$**commision**+**1,agent**$**commision**)**

##0428

agent**$**commision **<-** agent**$**commision**/**agent**$**nom

The calls are segregated between positive, negative and neutral.

1. The column Call direction is omitted as all the calls are outbound.

####Removing Call Direction from the DATASET CALL

call**$**call\_direction **=** **NULL**

1. No individual has two or more agent\_ids, as in case an agent is re-hired; they will receive the same agent\_id.
2. Call start time and end time have been removed from the columns as the Call Duration of the agent data calls is more important and pertinent to the conversion rate of calls.
3. Records which have term\_code, term\_type and term\_reason as blanks and have no term\_dates indicates that these are those agents which never left the company.
4. Feature\_values 1-24 are measurements and statistical calculations on the ***speech of the call*** (total speaking time, etc.)

feature\_values 25-34, 45-54, 65-74, 85-94, 105-114, 125-134, and 145-154 are statistical calculations on speech that has been identified as ***expressing a negative emotion*** (e.g. anger).

feature\_values 35-44, 55-64, 75-84, 95-104, 115-124, 135-144, and 155-164 are statistical calculations on speech that has been identified ***as expressing a positive emotion*** (e.g. joy).

feature\_values 165-176 measure ***proportions of negative/positive emotions***, emotional/unemotional speech, and transitions from one to the other (positive to negative or unemotional to emotional for example).

1. The unit of Call Data is HH:MM:SS
2. Account Number in Data is the Customer Account Number.
3. The agents are assigned to a work group for a month. Since the groups are inconsistent and they refer to the same entity, thus they are made uniform by creating a new categories of groups. Also, these groups are then clubbed together into a functional group as a parent group.

A ***custom function*** “myreplace” is written to achieve this grouping of work groups.

##-------------Data Grouping--------------

myreplace **<-** **function(**v,old,new**){**

v**=** as.character**(**v**)**

**if(**length**(**old**)==**length**(**new**)){**

**for(**i **in** 1**:**length**(**old**))**

**{**v**[**which**(**v**==**old**[**i**])]=** new**[**i**]**

**}**

**}**

**else{**

print**(**"Check Old and New Values - they are unequal"**)**

**}**

v

**}**

#Creating Vectors for new and old values to categorize the monthly group data

old\_values **=** c**(**"INB DSH" ,"DSH","PTM","VZ","INB SPR","SPR","G1A","G3A","DTV","COM","INB SPAN","INB A","INB B"**)**

new\_values **=** c**(**"DISH","DISH","TMOBILE","VERIZON","SPRINT","SPRINT","CREDIT CARD","TELECOM","DIRECTV","COMMERCIAL","INBOUND","INBOUND","INBOUND"**)**

agent**[**,c**(**"Jul\_group","Aug\_group","Sep\_group","Oct\_group","Nov\_group","Dec\_group"**)]** **=**

sapply**(**agent**[**,c**(**"Jul\_group","Aug\_group","Sep\_group","Oct\_group","Nov\_group","Dec\_group"**)]**,myreplace,old**=**old\_values,new**=**new\_values**)**

#Introducing Functional Groupings

old\_functional\_group **=** c**(**"TMOBILE","SPRINT","VERIZON","TELECOM","ATT","DISH","DIRECTV","LEGAL","COMMERCIAL","AUTO","CREDIT CARD","INBOUND"**)**

new\_functional\_group **=** c**(**"CELL CARRIER","CELL CARRIER","CELL CARRIER","CELL CARRIER","CELL CARRIER","TV PROVIDER","TV PROVIDER","LEGAL","COMMERCIAL","AUTO","CREDIT CARD","INBOUND"**)**

agent**[**c**(**"JulFunctionalGroup","AugFunctionalGroup","SepFunctionalGroup","OctFunctionalGroup","NovFunctionalGroup","DecFunctionalGroup"**)]** **<-** **NA**

agent**[**,c**(**"JulFunctionalGroup","AugFunctionalGroup","SepFunctionalGroup","OctFunctionalGroup","NovFunctionalGroup","DecFunctionalGroup"**)]** **=**

sapply**(**agent**[**,c**(**"Jul\_group","Aug\_group","Sep\_group","Oct\_group","Nov\_group","Dec\_group"**)]**,myreplace,old**=**old\_functional\_group,new**=**new\_functional\_group**)**

1. A list of groups for which an agent worked is created to measure the number of group changes an agent had during the last six months.

#Group Changes

agent**$**grpchange **=** **NA**

**for(**i **in** 1**:**nrow**(**agent**))** **{**

**if(NA** %in% agent**$**usergroup**[[**i**]]){**

agent**$**grpchange**[**i**]=**length**(**unique**(**agent**$**usergroup**[[**i**]]))-**1

**}**

**else** **{** agent**$**grpchange**[**i**]=**length**(**unique**(**agent**$**usergroup**[[**i**]]))}**

**}**

#Creating List of User Groups

**for(**i **in** 1**:**nrow**(**agent**)){**

agent**$**usergroup**[**i**]=**list**(**c**(**as.character**(**agent**$**Jul\_group**[**i**])**,as.character**(**agent**$**Aug\_group**[**i**])**,as.character**(**agent**$**Sep\_group**[**i**])**,as.character**(**agent**$**Oct\_group**[**i**])**,as.character**(**agent**$**Nov\_group**[**i**])**,as.character**(**agent**$**Dec\_group**[**i**])))**

**}**

1. The group name missing for each of the agent is calculated from the mode of the skill name of the agents.

#Calculating Mode of agent skills

getmode **<-** **function(**v**){**

v **=** v**[!**is.na**(**v**)]**

temp **<-** table**(**as.vector**(**v**))**

names**(**temp**)[**temp **==** max**(**temp**)]**

**}**

agent**$**groupmode**[**log\_agent**]** **=** tapply**(**call**$**skillgroup,call**$**agent\_id,getmode**)**

agent**$**groupmode **=** as.character**(**agent**$**groupmode**)**

####JUST CHECKING IF "new group" was a character

agent**$**new\_group **=** as.character**(**agent**$**new\_group**)**

##ALL NULL VALUES IN NEW GROUP REPLACED BY THE MODE

agent**$**new\_group**[**is.na**(**agent**$**new\_group**)]=**agent**$**groupmode

1. Calculated Median Call Duration and Total Number of calls for each agent for easy data handling and manipulation

#median call duration an agent talks

library**(**data.table**)**

cz **<-** tapply**(**call**$**CALL.DURATION.HMS.,call**$**agent\_id, median**)**

cz **=** data.frame**(**names**(**cz**)**,cz**)**

names**(**cz**)** **=** c**(**"agent\_id","Call\_Median\_Duration"**)**

agent **=** **(**merge**(**x **=** agent,y **=** cz, by **=** "agent\_id", all.x **=** **TRUE))**

#no of calls an agent made during the last six months

cy **<-** data.table**(**call**)**

cy **<-** tapply**(**call**$**account, call**$**agent\_id, length**)**

cy **=** data.frame**(**names**(**cy**)**,cy**)**

names**(**cy**)** **=** c**(**"agent\_id","Total\_CallData\_Calls"**)**

agent **=** **(**merge**(**x **=** agent,y **=** cy, by **=** "agent\_id", all.x **=** **TRUE))**

1. Rec\_Status – The exit status of a call is taken from each of the call in call data

#CALL DATA : REC-STATUS

new\_statuses **=** c**(**0,0,1,0,0,1,1,1,0,1,1,1**)**

call**$**negoutcome **<-** 0

#Run the below command wisely, it takes time

call**$**negoutcome **<-** as.numeric**(**sapply**(**call**$**call\_end\_status,myreplace,old**=**unique**(**call**$**call\_end\_status**)**,new**=**new\_statuses**))**

The exit status of a call signifies whether the agent was able to close the call successfully or not. An important predictor called “Negativity” is calculated for each agent. This parameter tells the number of calls the agents closed the call in a non-satisfactory manner.

#Calculating Number of Negative calls per agent.

agent**$**negativecalls **=** 0

#Agents Not in CALL DATASET - logical

log\_agent **=** agent**$**agent\_id %in% call**$**agent\_id

agent**$**negativecalls**[**log\_agent**]** **=** tapply**(**call**$**negoutcome, call**$**agent\_id, sum**)**

#Calculating Ratio of Negative calls/total calls

agent**$**negativity **=** 0

agent**$**negativity **=** round**(**agent**$**negativecalls **/** agent**$**Total\_CallData\_Calls,2**)**

## FEATURE DATA

### Categorizing feature data columns based on four categories

1) feature\_values 1-24 are measurements and statistical calculations on the speech of the call (total speaking time, etc.).

 2) feature\_values 25-34, 45-54, 65-74, 85-94, 105-114, 125-134, and 145-154 are statistical calculations on speech that has been identified as expressing a negative emotion (e.g. anger).

3) feature\_values 35-44, 55-64, 75-84, 95-104, 115-124, 135-144, and 155-164 are statistical calculations on speech that has been identified as expressing a positive emotion (e.g. joy).

4) feature\_values 165-176 measure proportions of negative/positive emotions, emotional/unemotional speech, and transitions from one to the other (positive to negative or unemotional to emotional for example).

feature\_new **<-** feature

feature\_new**$**CallSpeech **<-** rowMeans**(**feature\_new**[**,3**:**26**])**

feature\_new\_CallSpeech **<-** feature\_new**[**,3**:**26**]**

NE1 **<-** rowSums**(**feature\_new**[**,27**:**36**])** #25-34

NE2 **<-** rowSums**(**feature\_new**[**,47**:**56**])**

NE3 **<-** rowSums**(**feature\_new**[**,67**:**76**])**

NE4 **<-** rowSums**(**feature\_new**[**,87**:**96**])**

NE5 **<-** rowSums**(**feature\_new**[**,107**:**116**])**

NE6 **<-** rowSums**(**feature\_new**[**,127**:**136**])**

NE7 **<-** rowSums**(**feature\_new**[**,147**:**156**])**

NETotal **<-** NE1 **+** NE2 **+** NE3 **+** NE4 **+** NE5 **+** NE6 **+** NE7

feature\_new**$**NegativeEmotions **<-** NETotal**/(**70**)**

PE1 **<-** rowSums**(**feature\_new**[**,37**:**46**])**

PE2 **<-** rowSums**(**feature\_new**[**,57**:**66**])**

PE3 **<-** rowSums**(**feature\_new**[**,77**:**86**])**

PE4 **<-** rowSums**(**feature\_new**[**,97**:**106**])**

PE5 **<-** rowSums**(**feature\_new**[**,117**:**126**])**

PE6 **<-** rowSums**(**feature\_new**[**,137**:**146**])**

PE7 **<-** rowSums**(**feature\_new**[**,157**:**166**])**

PETotal **<-** PE1 **+** PE2 **+** PE3 **+** PE4 **+** PE5 **+** PE6 **+** PE7

feature\_new**$**PositiveEmotions **<-** PETotal**/(**70**)**

feature\_new**$**PositiveNegativeProportion **<-** rowMeans**(**feature\_new**[**, 167**:**178**])**

### Synchronizing Target Values for each agent in Feature Data.

call**$**new\_file\_audio **<-** substr**(**call**$**audio\_file\_name,1,34**)**

feature\_new**$**new\_file\_audio **<-** **(**substr**(**feature\_new**$**AUDIO.FILE.NAME, 1,34**))**

tmp1 **=** call**[**which**(**call**$**new\_file\_audio %in% feature\_new**$**new\_file\_audio**)**,c**(**"new\_file\_audio","agent\_id"**)]**

names**(**call**)**

tmp2 **=** feature\_new**[**which**(**feature\_new**$**new\_file\_audio %in% call**$**new\_file\_audio**)**,**]**

##Merging agent id to feature

feature\_new **=** merge**(**x **=** tmp1,y **=** tmp2, by.x **=** "new\_file\_audio", by.y **=** "new\_file\_audio"**)**

# Code to find the target values assigned to each agent

tmp3 **=** feature\_new**[**which**(**feature\_new**$**new\_file\_audio %in% call**$**new\_file\_audio**)**,c**(**"new\_file\_audio","target\_value"**)]**

##creating new frame to store File Name | Agent ID | Target Values

target\_agent **=** merge**(**x **=** tmp1,y **=** tmp3, by.x **=** "new\_file\_audio", by.y **=** "new\_file\_audio"**)**

# Creating a dataframe having columns "agent\_id" and "target\_value"

tmp4 **=** data.frame**(**tapply**(**target\_agent**$**target\_value,target\_agent**$**agent\_id,unique**))**

tmp4 **<-** cbind**(**rownames**(**tmp4**)**, tmp4**)**

rownames**(**tmp4**)** **<-** **NULL**

colnames**(**tmp4**)** **<-** c**(**"agent\_id","target\_value"**)**

names**(**agent**)**

# Adding a new column "newtarget\_valuee" in agent dataset for further computations

# Null and NA values in target\_value columns have been replaced by termiated values for respectve agents

agent **=** merge**(**x **=** agent, y **=** tmp4 , by **=** "agent\_id", all.x **=** **TRUE)**

agent**$**newtarget\_value**[**agent**$**target\_value **==** "NA"**]** **<-** agent**$**terminated**[**agent**$**target\_value **==** "NA"**]**

agent**$**newtarget\_value**[**agent**$**target\_value **==** "NULL"**]** **<-** agent**$**terminated**[**agent**$**target\_value **==** "NULL"**]**

as.character**(**agent**$**target\_value**)**

agent**$**newtarget\_value**[**agent**$**target\_value **==** "c(0, NA)"**]** **<-** 0

agent**$**newtarget\_value**[**agent**$**target\_value **==** "c(1, NA)"**]** **<-** 1

agent**$**newtarget\_value**[**agent**$**target\_value **==** "0"**]** **<-** 0

agent**$**newtarget\_value**[**agent**$**target\_value **==** "1"**]** **<-** 1

target\_agent **=** merge**(**x **=** target\_agent, y **=** agent**[**c**(**"agent\_id","newtarget\_value"**)]**, by **=** "agent\_id", all.x **=** **TRUE)**

feature\_new **=** cbind**(**feature\_new,target\_agent**[**"newtarget\_value"**])**

length**(**unique**(**feature\_new**$**agent\_id**))**

### Aggregate Agents and Target values across feature and call data

feature\_agent\_target **<-** aggregate**(**newtarget\_value **~** agent\_id, data **=** feature\_new\_1, FUN **=** mode**)**

feature\_agent\_target**$**newtarget\_value **<-** **NA**

names**(**feature\_agent\_target**)[**names**(**feature\_agent\_target**)** **==** "newtarget\_value"**]** **<-** "target"

feature\_agent\_target **<-** merge**(**feature\_agent\_target,agent**[**,c**(**"agent\_id","terminated","term\_type"**)]**,by **=** "agent\_id", all.y **=** **TRUE)**

feature\_agent\_target**$**target**[**feature\_agent\_target**$**terminated **==** 1 **&** feature\_agent\_target**$**term\_type **==** 0**]** **<-** 0

feature\_agent\_target**$**target**[**feature\_agent\_target**$**terminated **==** 1 **&** feature\_agent\_target**$**term\_type **==** 1**]** **<-** 1

feature\_agent\_target**$**target**[**feature\_agent\_target**$**terminated **==** 0**]** **<-** 0

feature\_agent\_target**$**term\_type **<-** **NULL**

feature\_agent\_target**$**newtarget\_value **<-** **NULL**

table**(**feature\_agent\_target**$**target**)**

**Assumption:** The target variable is defined as follows

1 – Voluntary termination

0 – Others (Not terminated or Involuntary termination)

This creates an almost equal distribution of the target variable in the dataset of 116 agents with the following distribution

1. - 47
2. -69

### Scaling and Aggregating Feature Data by Agent ID

Functions

#functions

mode **<-** **function(**x**)** **{**

ux **<-** unique**(**x**)**

ux**[**which.max**(**tabulate**(**match**(**x, ux**)))]**

**}**

#agg\_feature\_mean <- function(v,w){

# tapply(v,w,mean)

#}

scale\_01 **<-** **function(**x**){(**x**-**min**(**x**))/(**max**(**x**)-**min**(**x**))}**

#remove unwanted var

feature\_new**$**new\_file\_audio **<-** **NULL**

feature\_new**$**AUDIO.FILE.NAME **<-** **NULL**

feature\_new**$**target\_value **<-** **NULL**

feature\_new\_1 **<-** feature\_new**[**,c**(**1,181,178,179,180,182,2**:**177**)]**

feature\_agg **<-** feature\_new\_1

1. **Usual Scaling of all features using in build Scale function and aggregating by agent id using mean and mode**

#scale function

scaled.feature\_agg **<-** scale**(**feature\_agg**[**,7**:**182**]**,center **=** **TRUE**, scale **=** **TRUE)**

scaled.feature\_agg**<-**cbind**(**feature\_agg**[**,1**:**6**]**,scaled.feature\_agg**)**

scaled.feature\_agg\_mean **=** aggregate**(**. **~** agent\_id, data **=** scaled.feature\_agg, FUN **=** mean**)**

scaled.feature\_agg\_mode **=** aggregate**(**. **~** agent\_id, data **=** scaled.feature\_agg, FUN **=** mode**)**

1. **Custom Scaling function to scale between 0 and 1 of all features and aggregating by agent id using mean and mode**

#scale\_01 function

scaled\_01.feature\_agg **<-**scale\_01**(**feature\_agg**[**,7**:**182**])**

scaled\_01.feature\_agg**<-**cbind**(**feature\_agg**[**,1**:**6**]**,scaled\_01.feature\_agg**)**

scaled\_01.feature\_agg\_mean **<-** aggregate**(**. **~** agent\_id, data **=** scaled\_01.feature\_agg, FUN **=** mean**)**

scaled\_01.feature\_agg\_mode **<-** aggregate**(**. **~** agent\_id, data **=** scaled\_01.feature\_agg, FUN **=** mode**)**

1. **0 to 1 Scaling of categories in feature data and aggregating by agent id using mean and mode**

#feature callspeech

feature\_new\_callspeech **<-** feature\_new\_1**[**,c**(**1,7**:**30**)]**

scaled\_01.feature\_callspeech\_agg **<-** scale\_01**(**feature\_new\_callspeech**[-**1**])**

scaled\_01.feature\_callspeech\_agg **<-** cbind**(**feature\_new\_callspeech**[**,"agent\_id"**]**,scaled\_01.feature\_callspeech\_agg**)**

names**(**scaled\_01.feature\_callspeech\_agg**)[**names**(**scaled\_01.feature\_callspeech\_agg**)==**"feature\_new\_callspeech[, \"agent\_id\"]"**]** **<-** "agent\_id"

scaled\_01.feature\_callspeech\_agg\_mean **<-** aggregate**(**. **~** agent\_id, data **=** scaled\_01.feature\_callspeech\_agg, FUN **=** mean**)**

scaled\_01.feature\_callspeech\_agg\_mode **<-** aggregate**(**. **~** agent\_id, data **=** scaled\_01.feature\_callspeech\_agg, FUN **=** mode**)**

#feature negetiveemotions

feature\_new\_negativeemotions **<-** feature\_new\_1**[**,c**(**1,31**:**40,51**:**60,71**:**80,91**:**100,111**:**120,131**:**140,151**:**160**)]**

scaled\_01.feature\_negativeemotions\_agg **<-** scale\_01**(**feature\_new\_negativeemotions**[-**1**])**

scaled\_01.feature\_negativeemotions\_agg **<-** cbind**(**feature\_new\_negativeemotions**[**,"agent\_id"**]**,scaled\_01.feature\_negativeemotions\_agg**)**

names**(**scaled\_01.feature\_negativeemotions\_agg**)**

names**(**scaled\_01.feature\_negativeemotions\_agg**)[**names**(**scaled\_01.feature\_negativeemotions\_agg**)==**"feature\_new\_negativeemotions[, \"agent\_id\"]"**]** **<-** "agent\_id"

scaled\_01.feature\_negetiveemotions\_agg\_mean **<-** aggregate**(**. **~** agent\_id, data **=** scaled\_01.feature\_negativeemotions\_agg, FUN **=** mean**)**

scaled\_01.feature\_negetiveemotions\_agg\_mode **<-** aggregate**(**. **~** agent\_id, data **=** scaled\_01.feature\_negativeemotions\_agg, FUN **=** mode**)**

#feature positiveemotions

feature\_new\_positiveemotions **<-** feature\_new\_1**[**,c**(**1,41**:**50,61**:**70,81**:**90,101**:**110,121**:**130,141**:**150,161**:**170**)]**

scaled\_01.feature\_positiveemotions\_agg **<-** scale\_01**(**feature\_new\_positiveemotions**[-**1**])**

scaled\_01.feature\_positiveemotions\_agg **<-** cbind**(**feature\_new\_positiveemotions**[**,"agent\_id"**]**,scaled\_01.feature\_positiveemotions\_agg**)**

names**(**scaled\_01.feature\_positiveemotions\_agg**)**

names**(**scaled\_01.feature\_positiveemotions\_agg**)[**names**(**scaled\_01.feature\_positiveemotions\_agg**)==**"feature\_new\_positiveemotions[, \"agent\_id\"]"**]** **<-** "agent\_id"

scaled\_01.feature\_positiveemotions\_agg\_mean **<-** aggregate**(**. **~** agent\_id, data **=** scaled\_01.feature\_positiveemotions\_agg, FUN **=** mean**)**

scaled\_01.feature\_positiveemotions\_agg\_mode **<-** aggregate**(**. **~** agent\_id, data **=** scaled\_01.feature\_positiveemotions\_agg, FUN **=** mode**)**

#feature positivenegetiveproportions

feature\_new\_positivenegativeproportion **<-** feature\_new\_1**[**,c**(**1,171**:**182**)]**

scaled\_01.feature\_positivenegativeproportion\_agg **<-** scale\_01**(**feature\_new\_positivenegativeproportion**[-**1**])**

scaled\_01.feature\_positivenegativeproportion\_agg **<-** cbind**(**feature\_new\_positivenegativeproportion**[**,"agent\_id"**]**,scaled\_01.feature\_positivenegativeproportion\_agg**)**

names**(**scaled\_01.feature\_positivenegativeproportion\_agg**)**

names**(**scaled\_01.feature\_positivenegativeproportion\_agg**)[**names**(**scaled\_01.feature\_positivenegativeproportion\_agg**)==**"feature\_new\_positivenegativeproportion[, \"agent\_id\"]"**]** **<-** "agent\_id"

scaled\_01.feature\_positivenegativeproportion\_agg\_mean **<-** aggregate**(**. **~** agent\_id, data **=** scaled\_01.feature\_positivenegativeproportion\_agg, FUN **=** mean**)**

scaled\_01.feature\_positivenegativeproportion\_agg\_mode **<-** aggregate**(**. **~** agent\_id, data **=** scaled\_01.feature\_positivenegativeproportion\_agg, FUN **=** mode**)**

### Principle Component Analysis on the above scaled and aggregated data

1. **Usual Scaling of all features using in build Scale function and aggregating by agent id using mean and mode**

#pca scaled data

pca\_scaled\_feature\_mean **<-** prcomp**(**scaled.feature\_agg\_mean**[**,7**:**182**])**

summary**(**pca\_scaled\_feature\_mean**)**

pca\_scaled\_feature\_mean\_ds **<-** pca\_scaled\_feature\_mean**$**x**[**,1**:**23**]** #99%

99% of the variance is explained by the first 23 principle components

pca\_scaled\_feature\_mode **<-** prcomp**(**scaled.feature\_agg\_mode**[**,7**:**182**])**

summary**(**pca\_scaled\_feature\_mode**)**

pca\_scaled\_feature\_mode\_ds **<-** as.data.frame**(**pca\_scaled\_feature\_mode**$**x**[**,1**:**21**])** #99%

99% of the variance is explained by the first 21 principle components

1. **Custom Scaling function to scale between 0 and 1 of all features and aggregating by agent id using mean and mode**

#pca scaled\_01 data

pca\_scaled\_01\_feature\_mean **<-** prcomp**(**scaled\_01.feature\_agg\_mean**[**,7**:**182**])**

summary**(**pca\_scaled\_01\_feature\_mean**)**

pca\_scaled\_01\_feature\_mean\_ds **<-** pca\_scaled\_01\_feature\_mean**$**x**[**,1**:**4**]** #98.5%

pca\_scaled\_01\_feature\_mean\_ds **<-** cbind**(**as.character**(**scaled\_01.feature\_agg\_mean**$**agent\_id**)**,pca\_scaled\_01\_feature\_mean\_ds**)**

names**(**pca\_scaled\_01\_feature\_mean\_ds**)[**names**(**pca\_scaled\_01\_feature\_mean\_ds**)==**"V1"**]** **<-** "agent\_id"

98.5% of the variance is explained by the first 4 principle components

pca\_scaled\_01\_feature\_mode **<-** prcomp**(**scaled\_01.feature\_agg\_mode**[**,7**:**182**])**

summary**(**pca\_scaled\_01\_feature\_mode**)**

pca\_scaled\_01\_feature\_mode\_ds **<-** as.data.frame**(**pca\_scaled\_01\_feature\_mode**$**x**[**,1**:**6**])** #99.2%

pca\_scaled\_01\_feature\_mode\_ds **<-** cbind**(**scaled\_01.feature\_agg\_mode**[**,"agent\_id"**]**,pca\_scaled\_01\_feature\_mode\_ds**)**

names**(**pca\_scaled\_01\_feature\_mode\_ds**)[**names**(**pca\_scaled\_01\_feature\_mode\_ds**)==**"scaled\_01.feature\_agg\_mode[, \"agent\_id\"]"**]** **<-** "agent\_id"

99.2 % of the variance is explained by the first 6 principle components

1. **0 to 1 Scaling of categories in feature data and aggregating by agent id using mean and mode**

#feature callspeech

pca\_scaled\_01\_feature\_callspeech\_mean **<-** prcomp**(**scaled\_01.feature\_callspeech\_agg\_mean**[-**1**])**

summary**(**pca\_scaled\_01\_feature\_callspeech\_mean**)**

pca\_scaled\_01\_feature\_callspeech\_mean\_ds **<-** as.data.frame**(**pca\_scaled\_01\_feature\_callspeech\_mean**$**x**[**,1**:**6**])** #80.75..99.5%

99.5 % of the variance is explained by the first 6 principle components

#feature negetiveemotions

pca\_scaled\_01\_feature\_negetiveemotions\_mean **<-** prcomp**(**scaled\_01.feature\_negetiveemotions\_agg\_mean**[-**1**])**

summary**(**pca\_scaled\_01\_feature\_negetiveemotions\_mean**)**

pca\_scaled\_01\_feature\_negetiveemotions\_mean\_ds **<-** as.data.frame**(**pca\_scaled\_01\_feature\_negetiveemotions\_mean**$**x**[**,1**:**4**])** #92.8..99.3%

99.3 % of the variance is explained by the first 4 principle components

#feature positiveemotions

pca\_scaled\_01\_feature\_positiveemotions\_mean **<-** prcomp**(**scaled\_01.feature\_positiveemotions\_agg\_mean**[-**1**])**

summary**(**pca\_scaled\_01\_feature\_positiveemotions\_mean**)**

pca\_scaled\_01\_feature\_positiveemotions\_mean\_ds **<-** as.data.frame**(**pca\_scaled\_01\_feature\_positiveemotions\_mean**$**x**[**,1**:**8**])** #85..99.2%

99.2 % of the variance is explained by the first 8 principle components

#feature positivenegetiveproportions

pca\_scaled\_01\_feature\_positivenegativeproportion\_mean **<-** prcomp**(**scaled\_01.feature\_positivenegativeproportion\_agg\_mean**[-**1**])**

summary**(**pca\_scaled\_01\_feature\_positivenegativeproportion\_mean**)**

pca\_scaled\_01\_feature\_positivenegativeproportion\_mean\_ds **<-** as.data.frame**(**pca\_scaled\_01\_feature\_positivenegativeproportion\_mean**$**x**[**,1**:**3**])** #94.5..99.2%

99.2 % of the variance is explained by the first 3 principle components

### Dataset Creation

1. **Mean Aggregation with 0-1 Scaling**

ds\_feature\_scaled01\_mean **<-** merge**(**feature\_agent\_target,pca\_scaled\_01\_feature\_mean\_ds,by.x **=** "agent\_id",by.y **=** "V1", all.y **=** **TRUE)**

str**(**ds\_feature\_scaled01\_mean**)**

ds\_feature\_scaled01\_mean**$**PC1 **<-** as.numeric**(**as.character**(**ds\_feature\_scaled01\_mean**$**PC1**))**

ds\_feature\_scaled01\_mean**$**PC2 **<-** as.numeric**(**as.character**(**ds\_feature\_scaled01\_mean**$**PC2**))**

ds\_feature\_scaled01\_mean**$**PC3 **<-** as.numeric**(**as.character**(**ds\_feature\_scaled01\_mean**$**PC3**))**

ds\_feature\_scaled01\_mean**$**PC4 **<-** as.numeric**(**as.character**(**ds\_feature\_scaled01\_mean**$**PC4**))**

1. **Mode Aggregation with 0-1 Scaling**

ds\_feature\_scaled01\_mode **<-** merge**(**feature\_agent\_target,pca\_scaled\_01\_feature\_mode\_ds,by.x **=** "agent\_id",by.y **=** "agent\_id", all.y **=** **TRUE)**

str**(**ds\_feature\_scaled01\_mode**)**

1. **Mean Aggregation of Feature categories with 0-1 Scaling**

ds\_feature\_scaled01\_emotions\_mean **<-** cbind**(**pca\_scaled\_01\_feature\_callspeech\_mean\_ds**$**PC1,pca\_scaled\_01\_feature\_negetiveemotions\_mean\_ds**$**PC1,pca\_scaled\_01\_feature\_positiveemotions\_mean\_ds**$**PC1,pca\_scaled\_01\_feature\_positivenegativeproportion\_mean\_ds**$**PC1**)**

ds\_feature\_scaled01\_emotions\_mean **<-** cbind**(**as.data.frame**(**scaled\_01.feature\_positivenegativeproportion\_agg\_mean**$**agent\_id**)**,ds\_feature\_scaled01\_emotions\_mean**)**

names**(**ds\_feature\_scaled01\_emotions\_mean**)**

names**(**ds\_feature\_scaled01\_emotions\_mean**)[**names**(**ds\_feature\_scaled01\_emotions\_mean**)==**"scaled\_01.feature\_positivenegativeproportion\_agg\_mean$agent\_id"**]** **<-** "agent\_id"

ds\_feature\_scaled01\_emotions\_mean **<-** merge**(**feature\_agent\_target,ds\_feature\_scaled01\_emotions\_mean,by.x **=** "agent\_id",by.y **=** "agent\_id", all.y **=** **TRUE)**

# STATISTICAL ANALYSIS AND MODELING

## AGENT AND CALL DATA

We decided to first build a few trees without using the Rank Miner feature vectors. The interpretation of doing so was to analyze if such data could be combined with the RankMiner data in order to build better models. The statistical language that we have used to build our models is ‘R’. So our modelling process is:

1. **Build models on the master data statistics of the agent as obtained from the agent dataset and their call dataset**
2. **Build models on to evaluate the usefulness of the Feature Vectors that are generated by Rank Miner**
3. **Integrate the above two best datasets obtained and try to find a statistically significant model**

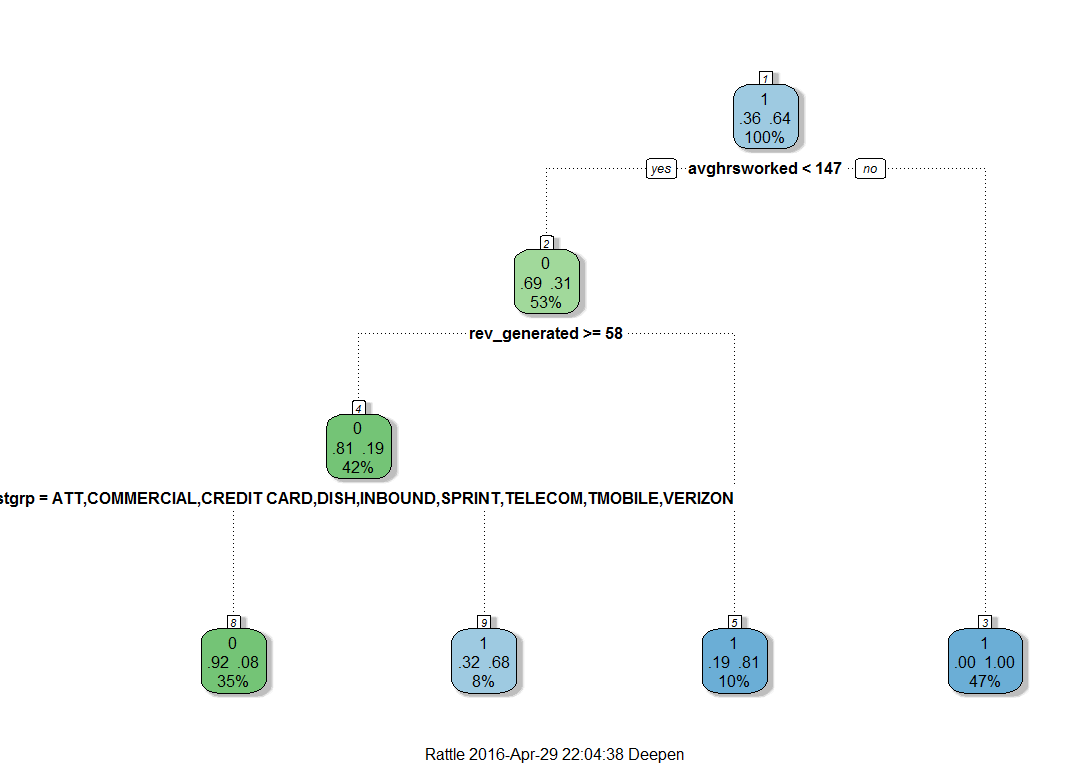
### MODEL 1: Decision trees

Code:

tree **=** rpart**(**terminated**~**.,data**=**agent\_trimmed,control **=** rpart.control**(**cp**=**0.02**)**,method**=**"class"**)**

library**(**rpart.plot**)**

fancyRpartPlot**(**tree**)**

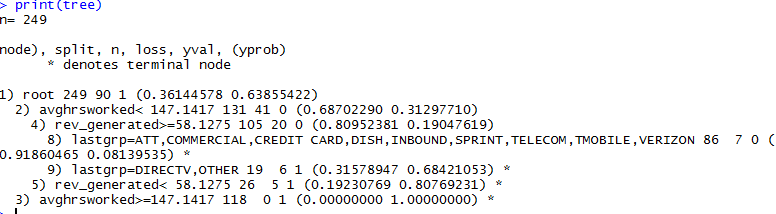


##Recall - Confusion Matrix

confusion.advisory\_tree **<-** table**(**agent\_trimmed**[**,"terminated"**]**,predict**(**tree,agent\_trimmed,type **=** "response"**))**

confusion.advisory\_tree ##Accuracy == 92.7%

#Printing the Tree:



The above snapshot clearly shows the number of agents in each of the splits and the accuracies of the splits. We can derive our predictions based on the above representation of the tree.

> confusion.advisory\_tree

0 1

0 79 11

1 7 152

##Accuracy == 92.7%

**Analysis and Interpretation (Similar for all tree):**

**The tree has given us a very good accuracy on the Recall Data.**

It has output major splits as visible in the tree above.

Significant Predictors: 1. Average Hours Worked

2. Revenue Generated

3. The last group the agent worked for

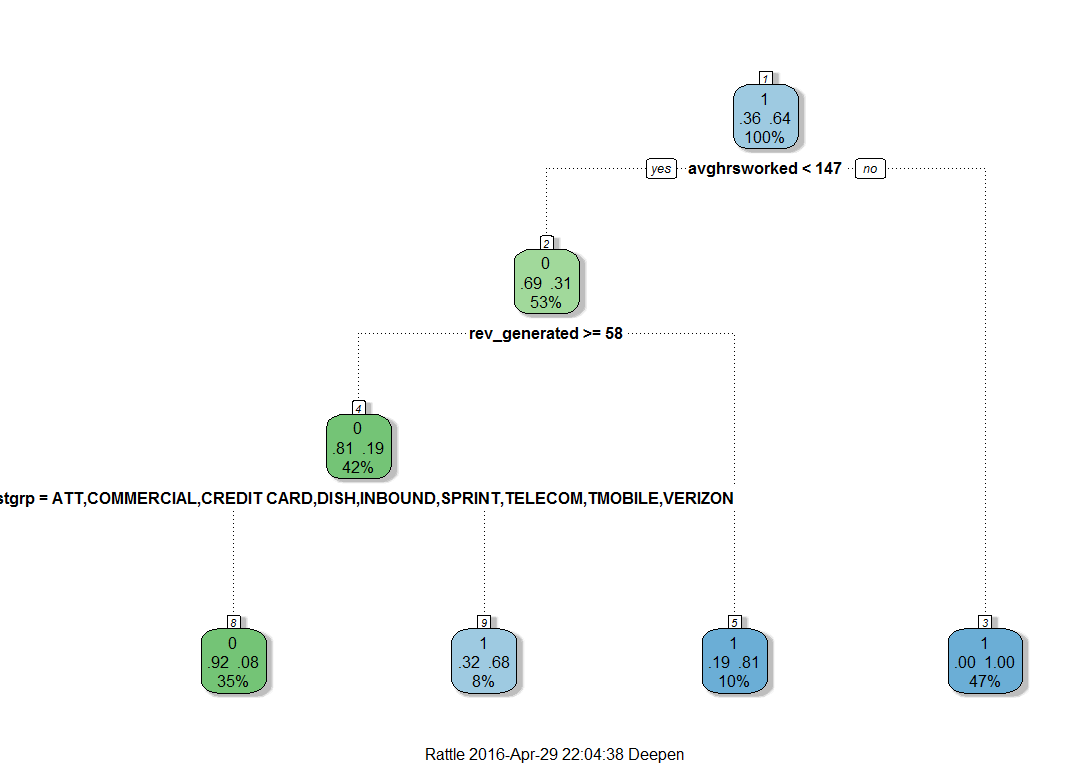
These results were also very upfront in the visualizations at the beginning of this report.

**We now try changing the complexity parameter (cp) to reduce pruning:**

####Changing complexity parameter to 0.01 -- reducing pruning

tree2 **=** rpart**(**terminated**~**.,data**=**agent\_trimmed,control **=** rpart.control**(**cp**=**0.01**)**,method**=**"class"**)**

rpart.plot**(**tree2,extra**=**101,type**=**4**)** ##Does not change , i.e. best splits obtained



No significant difference is obtained by doing so, thus we would try making some different tweaks to our data

Here, we have separated the “Voluntary” and “Involuntary” people leaving.

The output of our decision tree will be classified into:

**Voluntary = 1, Involuntary = 2 and Did Not Leave = 0.**

########TREE with 2 ---------------------------------

agent\_trimmed2 **=** agent**[**,c**(**"terminated","term\_type","lastgrp","rev\_generated","commision","Call\_Median\_Duration","Total\_CallData\_Calls", "negativity","skillgroupdiff","avghrsworked"**)]**

##Deleting 49 rows as done for agent\_trimmed

agent\_trimmed2**=** agent\_trimmed2**[!**is.na**(**agent\_trimmed2**$**lastgrp**)**,**]**

##Deleting rows that have their term type unknown - 16 agents

agent\_trimmed2 **=** agent\_trimmed2**[!(**is.na**(**agent\_trimmed2**$**term\_type**)** **&** agent\_trimmed2**$**terminated**==**1**)**,**]**

###Changing the value of INVOLUNTARILY left agents to 2

agent\_trimmed2**$**terminated**[**which**(**agent\_trimmed2**$**term\_type**==**0 **&** agent\_trimmed2**$**terminated**==**1**)]=**2

###Changing the value of INVOLUNTARILY left agents to 2

agent\_trimmed2**$**terminated**[**which**(**agent\_trimmed2**$**term\_type**==**0 **&** agent\_trimmed2**$**terminated**==**1**)]=**2

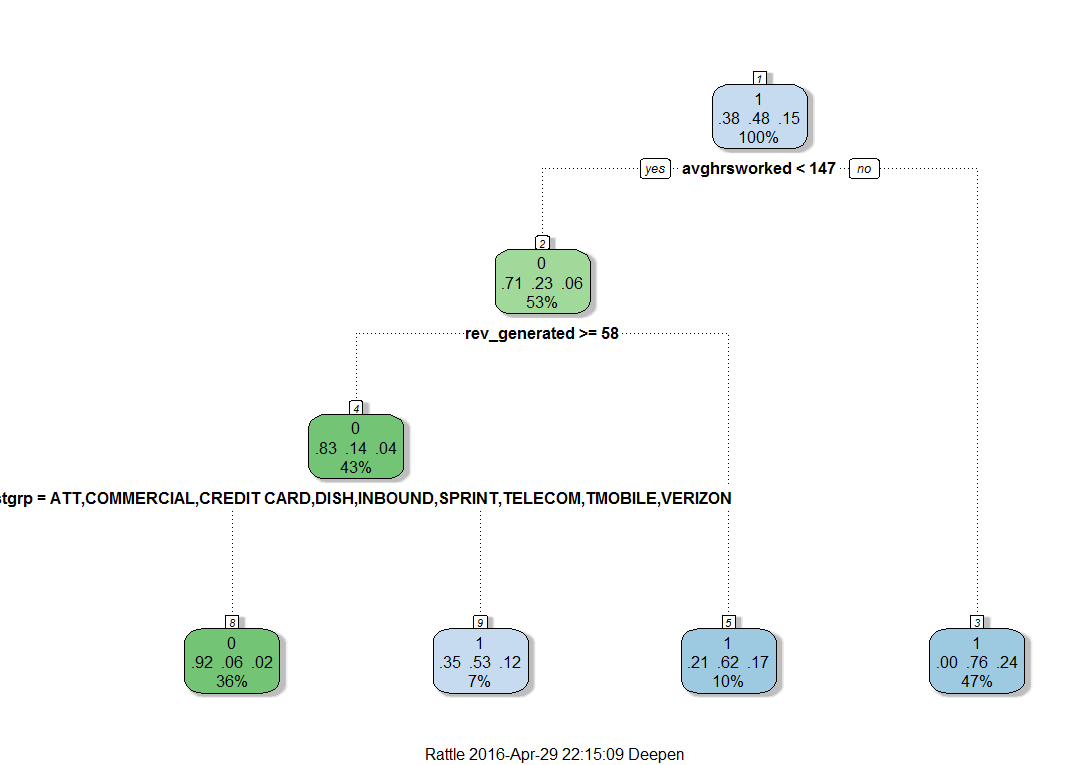
###Excluding Term Type

agent\_trimmed2**$**term\_type **=** **NULL**

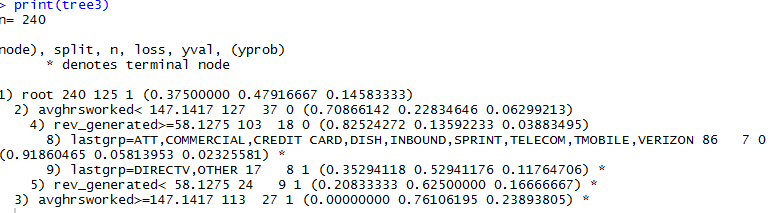
#########TREE----------------------------

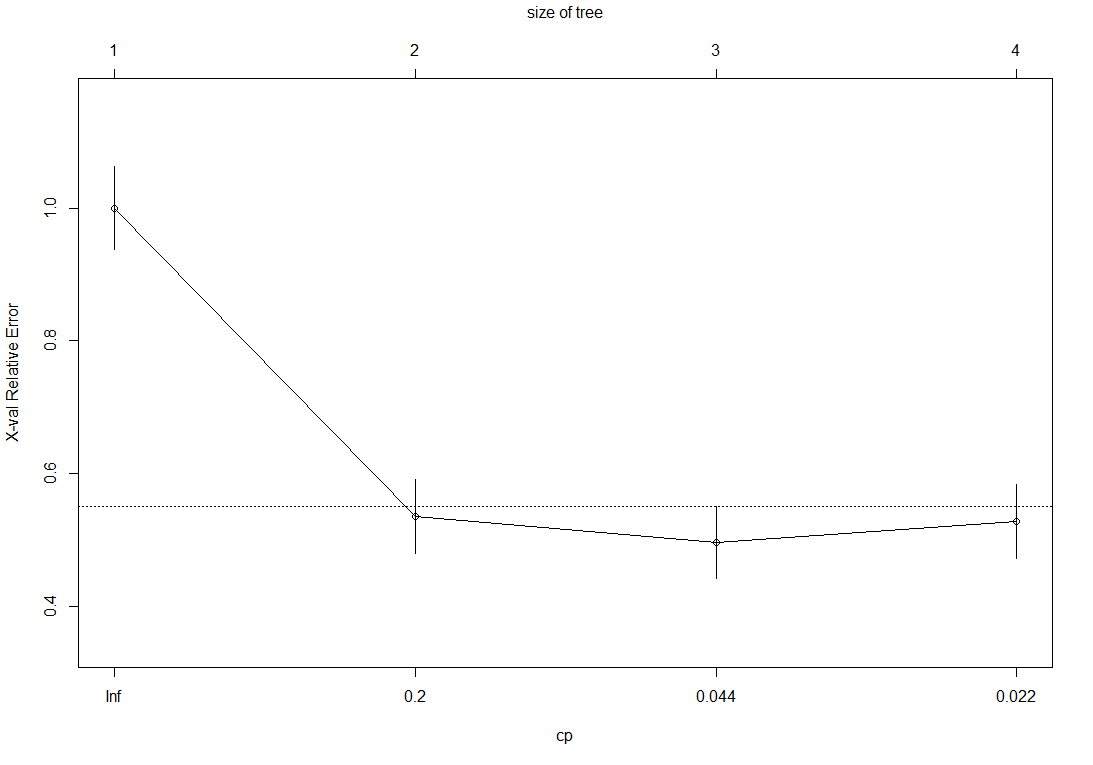
tree3 **=** rpart**(**terminated**~**.,data**=**agent\_trimmed2,control **=** rpart.control**(**cp**=**0.02**)**,method**=**"class"**)**

fancyRpartPlot**(**tree3**)**

**Plot:** 

Analysis of the tree:

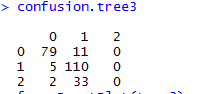


Plotting the Complexity Parameter to identify if pruning is required and until what number of split 

##Recall - Confusion Matrix

confusion.tree3 **<-** table**(**agent\_trimmed2**[**,"terminated"**]**,predict**(**tree3,agent\_trimmed2,type **=** "response"**))**

confusion.tree3 ##Accuracy == 92.7%

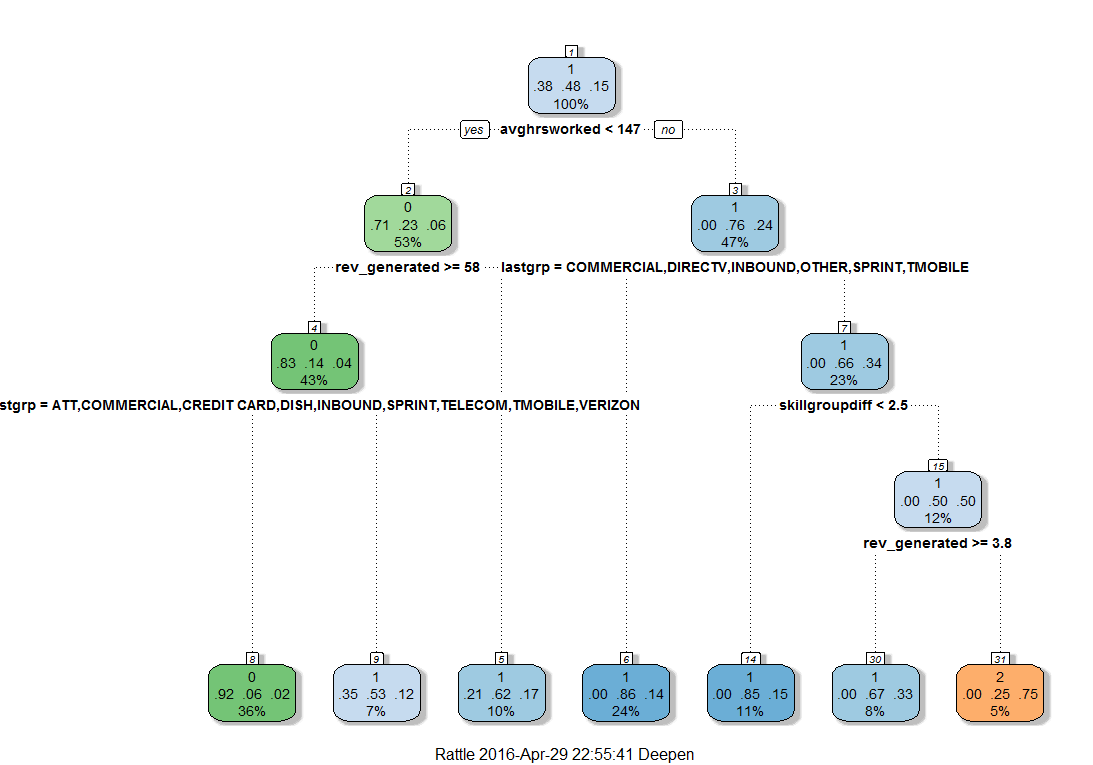


##Accuracy == 92.7%

Again changing the complexity parameter to see any possible extensions to the tree:

tree4 **=** rpart**(**terminated**~**.,data**=**agent\_trimmed2,control **=** rpart.control**(**cp**=**0.01**)**,method**=**"class"**)**

rpart.plot**(**tree4,extra**=**101,type**=**4**)** ##Does not change , i.e. best splits obtained



It does appear that the right split gives different results

confusion.tree4 **<-** table**(**agent\_trimmed2**[**,"terminated"**]**,predict**(**tree4,agent\_trimmed2,type **=** "response"**))**

confusion.tree4 ##Accuracy == 81.25%

Difference in the models:

The above 4 models have been interpreted by differing the Dependent Variable from ( 0 and 1 ) to ( 0 and [1,2] ). Thus, it would not be ideal to compare 1 & 2 with 3 & 4.

Thus, based on business decisions and implications, we can choose the model that suits best.

By now, we are confident that our important predictors are Average Worked hours, the Revenue generated and the Last group in which the agent worked before he left.

We will now run a regression model to identify the importance of the variables to support our analysis.

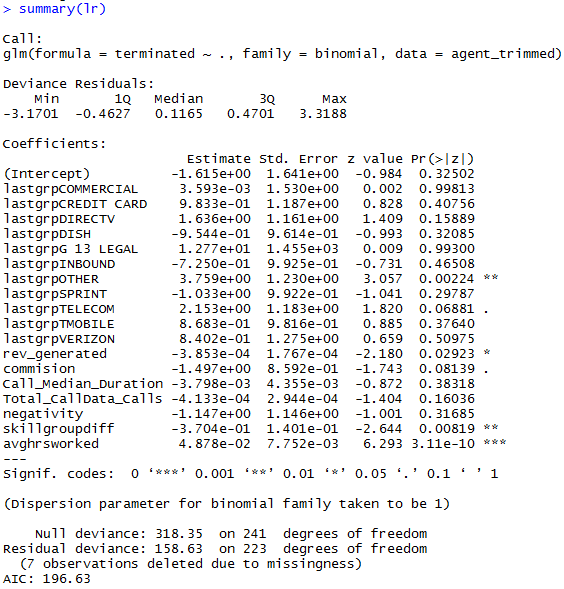
### MODEL 2: Logistic Binomial Regression

**MODEL 2.1**

We use the same dataset as we used for MODEL 1 viz. agent\_trimmed that has the Dependent variables as 0 and 1.

lr **=** glm**(**terminated **~** .,data**=**agent\_trimmed,family**=**binomial**)**

summary**(**lr**)**



Look at the AIC value above, we will use this to compare any modelling changes.

**MODEL 2.2**

**###It seems that the Lastgroup with "OTHER" is significant, so we create a dummy for last group = other or not.**

agent\_trimmed3 **=** agent\_trimmed

agent\_trimmed3**$**lastother **<-** 0

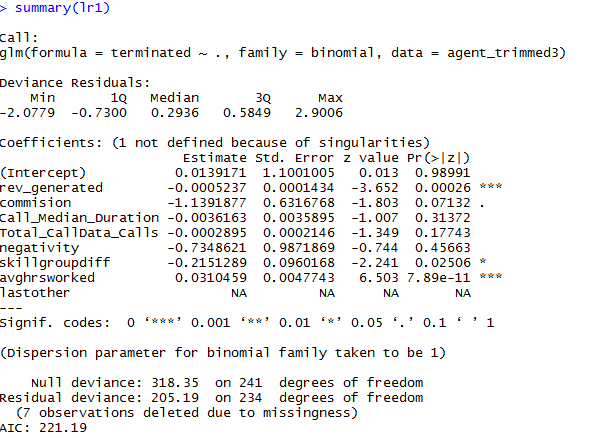
agent\_trimmed3**$**lastgrp**[**which**(**agent\_trimmed3**$**lastgrp**==**"OTHER"**)]** **=** 1

agent\_trimmed3**$**lastgrp **<-** **NULL**

**Running a new Model with the above changes:**

lr1 **=** glm**(**terminated **~** .,data**=**agent\_trimmed3,family**=**binomial**)**

summary**(**lr1**)**



We obtained a higher AIC value compared to the previous one. It seems that the model was overfitted.

Now lets try to analyze our Null Hypothesis that adding the dependent variables is a good predictor of the model i.e. the reduction in the Null Deviation to the Residual Deviance. We have used the chi-square characteristic to determine our p-value.

#Null Deviance ; P- Value =

1**-**pchisq**(**318.35,df**=**241**)**

##0.0006

#Statistical reducation in Null Deviance by adding the independent variables

1**-**pchisq**(**205.19,df**=**234**)**

##0.91

##

1**-**pchisq**(**318.35**-**205.19,df**=**241**-**234**)**

#0

**#This gives the p value of the model. it is approximating 0 which means that the null hypothesis can be rejected.**

**#Null Hypothesis being that there is no significant difference between the predictors for the ones who left and ones who did not leave.**

#AIC = 221.19

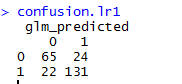
glm\_predicted **=** predict**(**lr1,agent\_trimmed3,type **=** "response"**)**

glm\_predicted**[**glm\_predicted**>=**0.5**]=**1

glm\_predicted**[**glm\_predicted**<**0.5**]=**0

##Accuracy on Recall

confusion.lr1 **=** table**(**agent\_trimmed3**[**,"terminated"**]**,glm\_predicted**)**



**(**65**+**131**)/**249

#Accuracy = 78.7%

### MODEL ANALYSIS AND RESULTS

Since Model 2.1 has a better AIC then Model 2.2, we used it for calculating the odds of the dependency on the dependent variable.

## FEATURE DATA

### Model Selection and Development:

Three models were run on three different feature dataset

1. **Top 4 PCA components (98.5% var) aggregated by mean for 117 agents**
2. **Top 6 PCA components (99.2% var) aggregated by mode for 117 agents**
3. **Top 1 PCA component from each of the 4 categories in feature dataset i.e. 4 PCAs (each > 85% var)**

**Detailed:**

1. **Top 4 PCA components (98.5% var) aggregated by mean for 117 agents**

#scaled01\_mean

net\_feature\_scaled01\_mean.sqrt **<-** neuralnet**(**ds\_feature\_scaled01\_mean**$**target**~**ds\_feature\_scaled01\_mean**$**PC1**+**ds\_feature\_scaled01\_mean**$**PC2**+**ds\_feature\_scaled01\_mean**$**PC3**+**ds\_feature\_scaled01\_mean**$**PC4,ds\_feature\_scaled01\_mean, hidden **=** 4**)**

plot**(**net\_feature\_scaled01\_mean.sqrt**)**

ds\_feature\_scaled01\_mean\_recall **<-** ds\_feature\_scaled01\_mean**[**,4**:**7**]**

net\_feature\_scaled01\_mean.results **<-** compute**(**net\_feature\_scaled01\_mean.sqrt, ds\_feature\_scaled01\_mean\_recall**)** #Run them through the neural network

ls**(**net\_feature\_scaled01\_mean.results**)**

ds\_feature\_scaled01\_mean\_predicted **<-** ds\_feature\_scaled01\_mean

ds\_feature\_scaled01\_mean\_predicted**$**predicted **<-** **NA**

ds\_feature\_scaled01\_mean\_predicted**$**predicted **<-** net\_feature\_scaled01\_mean.results**$**net.result

summary**(**ds\_feature\_scaled01\_mean\_predicted**$**predicted**)**

ds\_feature\_scaled01\_mean\_predicted**$**predicted**[**net\_feature\_scaled01\_mean.results**$**net.result **>=** 0.5**]** **<-** 1

ds\_feature\_scaled01\_mean\_predicted**$**predicted**[**net\_feature\_scaled01\_mean.results**$**net.result **<** 0.5**]** **<-** 0

length**(**ds\_feature\_scaled01\_mean\_predicted**$**agent\_id**[**ds\_feature\_scaled01\_mean\_predicted**$**target **==** ds\_feature\_scaled01\_mean\_predicted**$**predicted **&** ds\_feature\_scaled01\_mean\_predicted**$**predicted **==** 1**])**

#53/116\*100 = 45.68

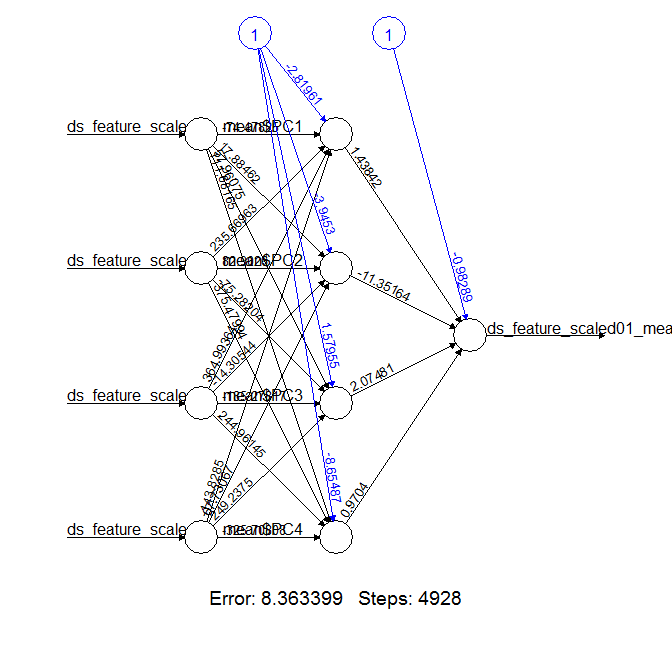
length**(**ds\_feature\_scaled01\_mean\_predicted**$**agent\_id**[**ds\_feature\_scaled01\_mean\_predicted**$**target **==** 1**])**

#69/116\*100 = 59.48

#confusion matrix

table**(**ds\_feature\_scaled01\_mean\_predicted**$**target,ds\_feature\_scaled01\_mean\_predicted**$**predicted**)**

net\_feature\_scaled01\_mean.acc **=** **(**32**+**57**)/**116



**Model Validation:**

**Net Recall Accuracy -** 76.72%

**Confusion Matrix –**

0 1

0 32 15

1 12 57

1. **Top 6 PCA components (99.2% var) aggregated by mode for 117 agents**

#scaled01\_mode

net\_feature\_scaled01\_mode.sqrt **<-** neuralnet**(**target**~**PC1**+**PC2**+**PC3**+**PC4,ds\_feature\_scaled01\_mode, hidden **=** 4**)**

plot**(**net\_feature\_scaled01\_mean.sqrt**)**

ds\_feature\_scaled01\_mode\_recall **<-** ds\_feature\_scaled01\_mode**[**,4**:**7**]**

net\_feature\_scaled01\_mode.results **<-** compute**(**net\_feature\_scaled01\_mode.sqrt, ds\_feature\_scaled01\_mode\_recall**)** #Run them through the neural network

ds\_feature\_scaled01\_mode\_predicted **<-** ds\_feature\_scaled01\_mode

ds\_feature\_scaled01\_mode\_predicted**$**predicted **<-** **NA**

ds\_feature\_scaled01\_mode\_predicted**$**predicted **<-** net\_feature\_scaled01\_mode.results**$**net.result

summary**(**ds\_feature\_scaled01\_mode\_predicted**$**predicted**)**

ds\_feature\_scaled01\_mode\_predicted**$**predicted**[**net\_feature\_scaled01\_mode.results**$**net.result **>=** 0.5**]** **<-** 1

ds\_feature\_scaled01\_mode\_predicted**$**predicted**[**net\_feature\_scaled01\_mode.results**$**net.result **<** 0.5**]** **<-** 0

length**(**ds\_feature\_scaled01\_mode\_predicted**$**agent\_id**[**ds\_feature\_scaled01\_mode\_predicted**$**target **==** ds\_feature\_scaled01\_mode\_predicted**$**predicted **&** ds\_feature\_scaled01\_mode\_predicted**$**predicted **==** 1**])**

66**/**116**\***100

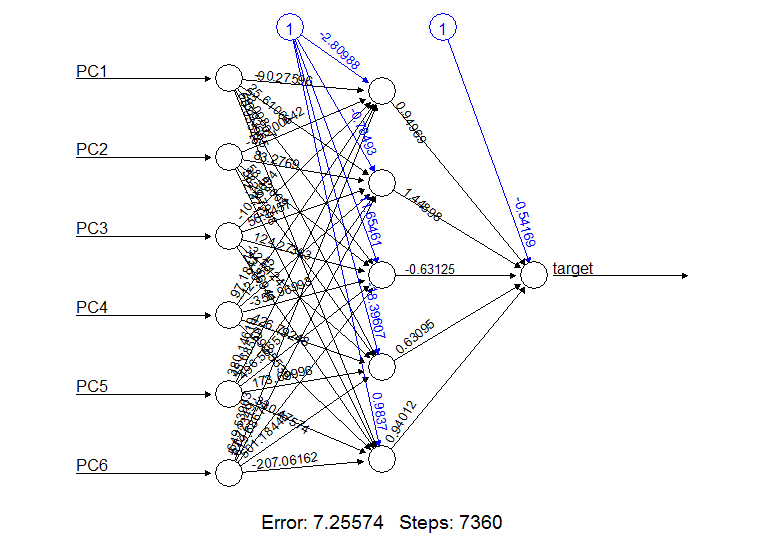
length**(**ds\_feature\_scaled01\_mode\_predicted**$**agent\_id**[**ds\_feature\_scaled01\_mode\_predicted**$**target **==** 1**])**

#69/116\*100 = 59.48

#confusion matrix

table**(**ds\_feature\_scaled01\_mode\_predicted**$**target,ds\_feature\_scaled01\_mode\_predicted**$**predicted**)**

net\_feature\_scaled01\_mode.acc **<-** **(**26**+**55**)/**116



**Model Validation:**

**Net Recall Accuracy –** 81.9 %

**Confusion Matrix –**

0 1

0 38 9

1 12 57

1. **Top PCA components from each of the 4 categories in feature dataset (each > 85% var)**

#scaled01\_feature\_emotions

net\_feature\_scaled01\_emotions.sqrt **<-** neuralnet**(**target**~**ds\_feature\_scaled01\_emotions\_mean**$**`1`**+**ds\_feature\_scaled01\_emotions\_mean**$**`2`**+**ds\_feature\_scaled01\_emotions\_mean**$**`3`**+**ds\_feature\_scaled01\_emotions\_mean**$**`4`,ds\_feature\_scaled01\_emotions\_mean, hidden **=** 4**)**

plot**(**net\_feature\_scaled01\_emotions.sqrt**)**

ds\_feature\_scaled01\_emotions\_mean\_recall **<-** ds\_feature\_scaled01\_emotions\_mean**[**,4**:**7**]**

net\_feature\_scaled01\_emotions.results **<-** compute**(**net\_feature\_scaled01\_emotions.sqrt, ds\_feature\_scaled01\_emotions\_mean\_recall**)** #Run them through the neural network

ds\_feature\_scaled01\_emotions\_mean\_predicted **<-** ds\_feature\_scaled01\_emotions\_mean

ds\_feature\_scaled01\_emotions\_mean\_predicted**$**predicted **<-** **NA**

ds\_feature\_scaled01\_emotions\_mean\_predicted**$**predicted **<-** net\_feature\_scaled01\_emotions.results**$**net.result

summary**(**ds\_feature\_scaled01\_emotions\_mean\_predicted**$**predicted**)**

ds\_feature\_scaled01\_emotions\_mean\_predicted**$**predicted**[**net\_feature\_scaled01\_emotions.results**$**net.result **>=** 0.5**]** **<-** 1

ds\_feature\_scaled01\_emotions\_mean\_predicted**$**predicted**[**net\_feature\_scaled01\_emotions.results**$**net.result **<** 0.5**]** **<-** 0

length**(**ds\_feature\_scaled01\_emotions\_mean\_predicted**$**agent\_id**[**ds\_feature\_scaled01\_emotions\_mean\_predicted**$**target **==** ds\_feature\_scaled01\_emotions\_mean\_predicted**$**predicted **&** ds\_feature\_scaled01\_emotions\_mean\_predicted**$**predicted **==** 1**])**

66**/**116**\***100

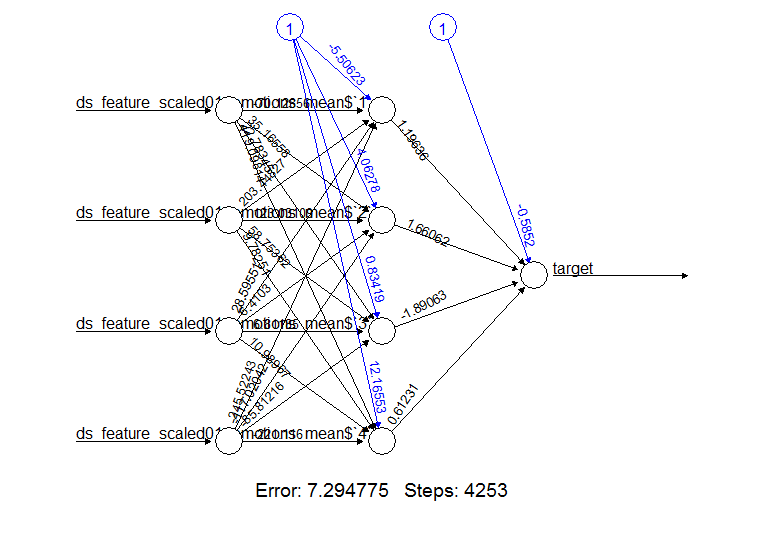
length**(**ds\_feature\_scaled01\_emotions\_mean\_predicted**$**agent\_id**[**ds\_feature\_scaled01\_emotions\_mean\_predicted**$**target **==** 1**])**

#69/116\*100 = 59.48

#confusion matrix

table**(**ds\_feature\_scaled01\_emotions\_mean\_predicted**$**target,ds\_feature\_scaled01\_emotions\_mean\_predicted**$**predicted**)**

net\_feature\_scaled01\_emotions.acc **=** **(**37**+**60**)/**116



**Model Validation:**

**Net Recall Accuracy –** 83.62 %

**Confusion Matrix –**

0 1

0 37 10

1 9 60

### Model Analysis and Results

The baseline accuracy for predicting a voluntary termination i.e. a 1 for 116 agents is 69/116 i.e. **59.48%.**

So any model with accuracy more than this is an acceptable model.

Summary of model accuracies-

|  |  |  |
| --- | --- | --- |
| Model 1 | Model 2 | Model 3 |
| 76.72% | 81.90% | 83.62% |

Hence a neural network model on Top PCA components extracted from each category i.e. Call Speech, Negative Emotions, Positive Emotions, Positive Negative Proportions is the best model with least FP and FN so far and to predict recall with 83.62 % accuracy.

### Modeling Extensions

1. Aggregating PCA components of each category based on 95% or mode instead of mean and remodeling.
2. Building models on the in-build scaled data as well
3. Using more combinations of PCA components and parameters for modeling.
4. Changing Neural Network parameters like Threshold, Hidden nodes, epochs etc.

# CONCLUSION

## MANAGARIAL IMPLECATIONS OF RESULTS

It seems that agents who are working for greater number of hours are leaving the company ~ 76% Voluntarily and 24% Involuntarily.

As a manager, this could probably imply different meanings to me for both the different types of attritions:

1. Agents who leave voluntarily are probably over burdened with their work. They are unhappy about the extra hours that they are putting. Some questions I would ask myself:

* Are these agents being supported by additional incentives for the higher number of hours they are putting in?
* Are they happy with their work?
* Are they actually productive enough for the amount of hours they put?

I would closely monitor these agents.

1. Agents who have left Involuntarily or fired might be the cause of some unproductive hours being added to the timesheet over time. This costs the company additional resources and thus management decision might have influenced their termination.
2. Agents who have generated less revenue for the company has ~60% employees who have left voluntarily. For the same reasons discussed above, their productivity is being reduced as they might not be enjoying their work or planning to switch their company.
3. A large proportion of agents who have generated higher revenue are staying back with the company.

* I might have to check if the company is focusing or paying attention to agents who have only generated revenue? What about those who are also putting in more hours and might be trying hard, but not being successful.
* Is my business model too biased on incentives to agents who generate sufficient revenue?

1. It also seems that a certain “Groups” have higher attrition rates. I might monitor these groups closely and try to find a detail analysis on them. It is highly likely that the agents are facing a huge amount of problems with their customers in these groups that are forcing them to move away from work.

Using Call Audio Feature Data, analysis on the audio categories i.e. the call speech quality, positive emotions, negative emotions , positive negative emotion proportions affect agent attrition rate and can be predicted with more than 75% accuracy.

# FUTURE WORK

1) We have only used Principle component analysis for dimension reduction, we can try using Lasso as well for variable selection to run better models.

2) Multivariate Analysis and more Logistic regressions on feature data can be considered on feature data subsets

3) Bayesian Probabilistic models could be applied to the feature, agent and call data together to understand how call audio data is influenced by the agent behavior and get probability estimations on the termination.

# SOURCE CODE



# ACKNOWLEDGEMENT

It is my proud privilege to epitomize my deepest sense of gratitude and indebtedness to the Professor [**Daniel Zantedeschi**](https://usflearn.instructure.com/courses/1107921/users/3910344) **,** an Eminent Scholar for the Information Systems Decision Sciences Department of the University of South Florida and **Eric** from RankMiner for his valuable guidance, keen and sustained interest, intuitive ideas and persistent endeavor. His inspiring assistance, laconic reciprocation and affectionate care enabled me to complete my work smoothly and successfully.