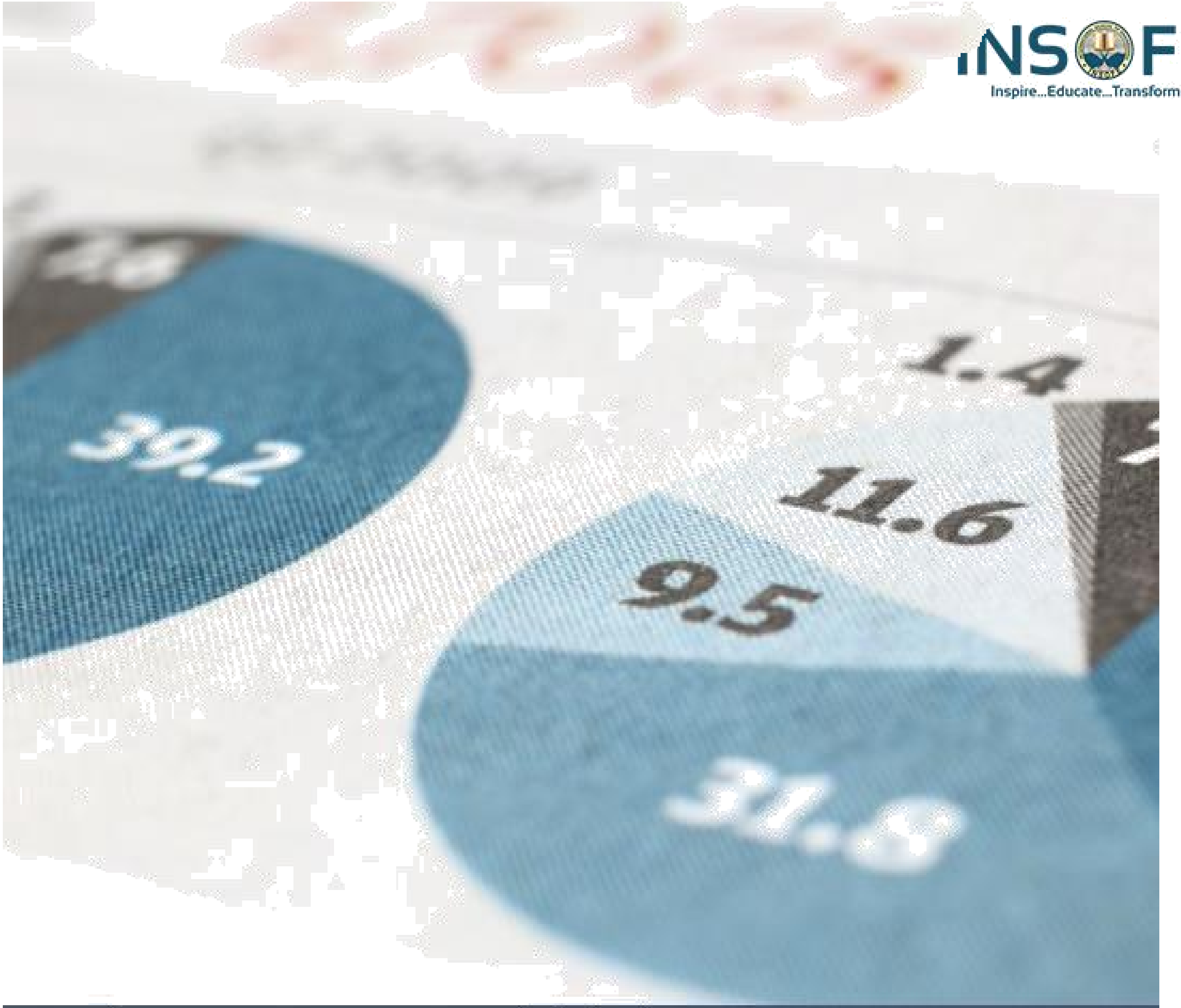


Agency Performance Analysis

CSE 9099 CPEE Project

Pratik Goyal 25/02/2017 Batch 21

CONTENTS

ABSTRACT

The Azure Insurance Company operates through various agencies to sell their insurance product. Each agency's performance is measured in terms of growth rate. Through this project we determine the important factors leads to change in growth rate of the agency

INTRODUCTION

The Azure Insurance Group, consists of 10 property and casualty insurance, life insurance and insurance brokerage companies. The property and casualty companies in the group operate in a 17-state region. Mutual group is a major regional property and casualty insurer, represented by more than 4,000 independent agents who live and work in local communities through a six-state region. Define the metrics to analyse agent performance based on several attributes like demography, products sold, new business, etc. Azure is interested in improving their existing knowledge used for agent segmentation in a supervised predictive framework

STATE - OF - ART

Domain Detail:

Insurance works by pooling risk, large group of people who want to insure against a particular loss pay their premiums into what we will call the insurance bucket, or pool. Because the number of insured individuals is so large, insurance companies can use statistical analysis to project what their actual losses will be within the given class. They know that not all insured individuals will suffer losses at the same time or at all. This allows the insurance companies to operate profitably and at the same time pay for claims that may arise. For instance, most people have auto insurance but only a few actually get into an accident. You pay for the probability of the loss and for the protection that you will be paid for losses in the event they occur.

How Insurance Agency Works:

Insurance companies create value by pooling and redistributing various types of risk. It does this by collecting liabilities (i.e. premiums) from everyone that it insures and then paying them out to the few that actually need them. The insurance company can then effectively redistribute those liabilities to entities faced with some sort of event-driven crisis, where they will ostensibly need more cash than they currently have on hand. As not everyone within the pool will actually suffer an event requiring the total use of all of their premiums, this pooling and redistribution function lowers the total cost of risk management for everyone in the pool.

Insurance companies theoretically make money in two ways:

- By charging enough premiums to cover the expected payouts that they will have to cover over the life of the policy
- By earning investment returns ("the float") using the collected premiums

Duties of Insurance Agents:

Insurance agents work for insurance providers and try to sell new policies or renewals to customers

1. Educate Your Clients.
2. Be a Consultant.
3. Be Thorough.
4. Seek New Clients

Current Industries Practice's:

All insurance companies make use of diff forms of KPI'S to check and monitor their financial flow and performance over the year. (Key Performance Indicator)

KPLs used by insurance brokers , here is a list of the four most important ones, they includes:

1) Loss ratio KPI
2) cost per claim KPI
3) revenue per policyholder KPI and
4) expense ratio KPI

Proposed Solution:

The typical agency owner doesn’t take the time to collect and analyze the important factors that will make a difference in the success and growth of their business.

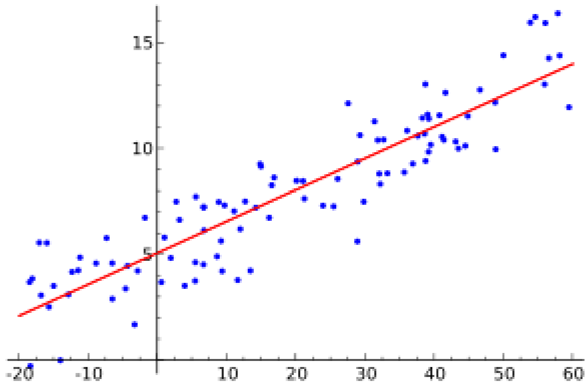
We need to find the important factors which are influencing the growth rate from previous data.

We are building the model which will give the important factors by analyzing the past data.

METHODOLOGIES USED IN TEXT CLASSIFICATION

In statistical modeling, regression analysis is a statistical process for estimating the relationships among variables. It includes many techniques for modeling and analyzing several variables, when the focus is on the relationship between a dependent variable and one or more independent variables (or 'predictors'). More specifically, regression analysis helps one understand how the typical value of the dependent variable (or 'criterion variable') changes when any one of the independent variables is varied, while the other independent variables are held fixed. Most commonly, regression analysis estimates the conditional expectation of the dependent variable given the independent variables – that is, the average value of the dependent variable when the independent variables are fixed. Less commonly, the focus is on a quantile, or other location parameter of the conditional distribution of the dependent variable given the independent variables. In all cases, the estimation target is a function of the independent variables called the regression function. In regression analysis, it is also of interest to characterize the variation of the dependent variable around the regression function which can be described by a probability distribution. A related but distinct approach is necessary condition analysis (NCA), which estimates the maximum (rather than average) value of the dependent variable for a given value of the independent variable (ceiling line rather than central line) in order to identify what value of the independent variable is necessary but not sufficient for a given value of the dependent variable

Regression analysis is widely used for prediction and forecasting, where its use has substantial overlap with the field of machine learning. Regression analysis is also used to understand which among the independent variables are related to the dependent variable, and to explore the forms of these relationships. In restricted circumstances, regression analysis can be used to infer causal relationships between the independent and dependent variables. However this can lead to illusions or false relationships, so caution is advisable;[2] for example, correlation does not imply causation.



Random Forest :- Random forests or random decision forests[1][2] are an ensemble learning method for classification, regression and other tasks, that operate by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees. Random decision forests correct for decision trees' habit of overfitting to their training set.

The first algorithm for random decision forests was created by Tin Kam Ho using the random subspace method, which, in Ho's formulation, is a way to implement the "stochastic discrimination" approach to classification proposed by Eugene Kleinberg.

An extension of the algorithm was developed by Leo Breiman[7] and Adele Cutler, and "Random Forests" is their trademark.[9] The extension combines Breiman's "bagging" idea and random selection of features, introduced first by Ho and later independently by Amit and Geman in order to construct a collection of decision trees with controlled variance.

SVD:-

Singular value decomposition (SVD) is a way to decompose a matrix into some successive approximation. This decomposition can reveal internal structure of the matrix. The method is very useful for text mining. Usually co-occurrence matrix (terms-by-documents matrix) defined over a large corpus of text documents contains a lot of noise. Singular value decomposition allows approximation of the co-occurrence matrix and thereby can reveal internal (latent) structure of text corpus. It decreases information noise, removes the unnecessary (random) links between terms and increases the value of important information. In this paper we apply singular value decomposition to improve text classification. We build co-occurrence matrix and then approximate it by SVD. Obtained matrix is very useful for creating new feature space.

DATASET

Our dataset consist of 2, 31,328 records and 49 Variables.

- AGENCY_ID
- PRIMARY_AGENCY_ID – (contains missing values) master agency if part of group
- PROD_ABBR – 33 products of which 14 are CL and 19 are PL
- PROD_LINE – commercial lines (CL) or personal lines (PL)
- STATE_ABBR
- STAT_PROFILE_DATE_YEAR – data starts in mid-2005 and continues into 2015
- RETENTION_POLY_QTY – current number of policies that are still active from previous year
- POLY_INFORCE_QTY – number of policies active for that year
- PREV_POLY_INFORCE_QTY – (contains missing values) number of policies active in the previous year
- NB_WRTN_PREM_AMT – new business in written premium
- WRTN_PREM_AMT – total written premium
- PREV_WRTN_PREM_AMT – (contains missing values) written premium during the same period in the previous year
- PRD_ERND_PREM_AMT – amount of premium taken in
- PRD_INCRD_LOSSES_AMT – losses
- MONTHS – number of months included in the data for that year; the original data was monthly and some months were missing so the aggregate doesn't make up the entire year if the months value is less than 12
- RETENTION_RATIO – (computed & contains missing values) computed for each row in the data as RETENTION_POLY_QTY / PREV_POLY_INFORCE_QTY; therefore it's a granular measure of the retention for that agency writing that particular product in that particular state from the previous year
- LOSS_RATIO – (computed & contains missing values) computed for each row in the data as PRD_INCRD_LOSSES_AMT / WRTN_PREM_AMT; currently I'm only computing results where the WRTN_PREM_AMT is greater than 0; however there are many cases where there are losses but no premium maybe from a previous claim that's still being paid, so I included codes to indicate whether there are positive or negative losses on zero premiums; they are located at the end of this document
- LOSS_RATIO_3YR – (computed & contains missing values) computed by agency by line of business by year for the three year period ending in that year, if there is data for three years, otherwise the two years or one year of data available; to make this more tangible, the first complete year of data we have for an agency will have the loss ratio for that year; the second complete year of data will have the mean loss ratio for that year and the previous year; the third complete year of data will have the mean loss ratio for those three years; then the fourth and greater will have the mean of the three year period ending in that year; note that the mean loss ratios are computed independently for PL and CL
- GROWTH_RATE_3YR – (computed & contains missing values) computed by agency by line of business by year for the three year period ending in that year; measures the average

growth in written premium for that agency in that line of business; only computes results for agencies that have data for the entire range of years; since the measure is over three years of growth, there needs to be a base year to be used as a standard so four years of data are needed; in order to include as many results as possible, the PREV_WRTN_PREM_AMT column is used if it exists in the first year of data available, so that it can be used as a base year, otherwise the WRTN_PREM_AMT is the only column used

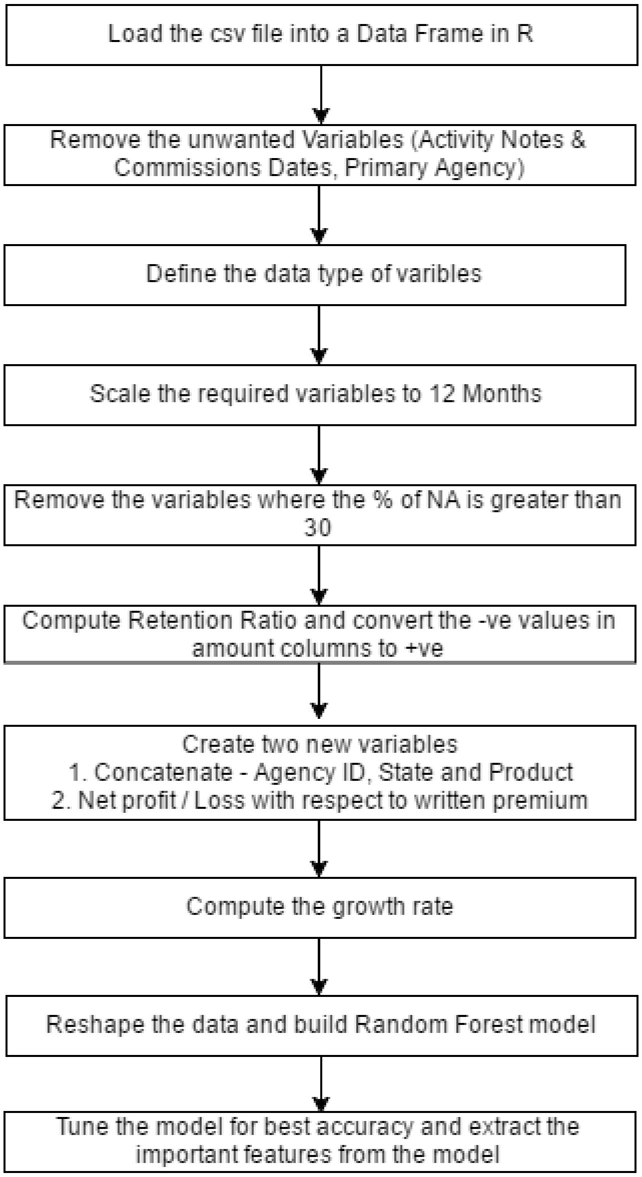
- AGENCY_APPOINTMENT_YEAR – (contains missing values) year the agency started doing business with Azure
- ACTIVE_PRODUCERS – (contains missing values) number of active producers in the agency
- MAX_AGE – (contains missing values & results may not be accurate) maximum age producer at that agency
- MIN_AGE – (contains missing values) minimum age producer at that agency
- VENDOR_IND – indicator column to specify whether the agency subscribes to a vendor
- VENDOR – (contains missing values) the vendor that the agency subscribes to
- PL_START_YEAR – (contains missing values) year the agency started using the PL vendor
- PL_END_YEAR – (contains missing values) year the agency stopped using the PL vendor
- COMMISIONS_START_YEAR – (contains missing values) year the agency started using the COMMISIONS vendor
- COMMISIONS_END_YEAR – (contains missing values) year the agency stopped using the COMMISIONS vendor
- CL_START_YEAR – (contains missing values) year the agency started using the CL vendor
- CL_END_YEAR – (contains missing values) year the agency stopped using the CL vendor
- ACTIVITY_NOTES_START_YEAR – (contains missing values) year the agency started using the ACTIVITY NOTES vendor
- ACTIVITY_NOTES_END_YEAR – (contains missing values) year the agency stopped using the ACTIVITY NOTES vendor
- CL_BOUND_CT_MDS – (contains missing values) number of bound policies quoted through a MDS (probably a data recording error, should be DSM) in the current year to date, that is the first six months of 2015, in commercial lines
- CL_QUO_CT_MDS – (contains missing values) number of quoted policies through a MDS (probably a data recording error, should be DSM) in the current year to date, that is the first six months of 2015, in commercial lines
- CL_BOUND_CT_SBZ – (contains missing values) number of bound policies quoted through a SBZ in the current year to date, that is the first six months of 2015, in commercial lines
- CL_QUO_CT_SBZ – (contains missing values) number of quoted policies through a SBZ in the current year to date, that is the first six months of 2015, in commercial lines
- CL_BOUND_CT_eQT – (contains missing values) number of bound policies quoted through an eQT in the current year to date, that is the first six months of 2015, in commercial lines
- CL_QUO_CT_eQT – (contains missing values) number of quoted policies though an eQT in the current year to date, that is the first six months of 2015, in commercial lines

- PL_BOUND_CT_ELINKS – (contains missing values) number of bound policies quoted through ELINKS since September 2013 in personal lines
- PL_QUO_CT_ELINKS – (contains missing values) number of quoted policies though ELINKS since September 2013 in personal lines
- PL_BOUND_CT_PLRANK – (contains missing values) number of bound policies quoted through PLRANK since September 2013 in personal lines
- PL_QUO_CT_PLRANK – (contains missing values) number of quoted policies though PLRANK since September 2013 in personal lines
- PL_BOUND_CT_eQTte – (contains missing values) number of bound policies quoted through eQTte since September 2013 in personal lines
- PL_QUO_CT_eQTte – (contains missing values) number of quoted policies though eQTte since September 2013 in personal lines
- PL_BOUND_CT_APPLIED – (contains missing values) number of bound policies quoted through APPLIED since September 2013 in personal lines

- PL_QUO_CT_APPLIED – (contains missing values) number of quoted policies though APPLIED since September 2013 in personal lines
- PL_BOUND_CT_TRANSACTNOW – (contains missing values) number of bound policies quoted through TRANSACTNOW since September 2013 in personal lines
- PL_QUO_CT_TRANSACTNOW – (contains missing values) number of quoted policies though TRANSACTNOW since September 2013 in personal lines

PREPROCESSING:-

1. Load the dataset and replace 99999 values with NA.
2. Removing Unwanted variables such as Activity notes, commission date etc.
3. Define the data types of variable (Converting to categorical and numeric data).
4. Scale the required variables to 12 months.
5. Removing the variables where NA value is greater than 30%.
6. Computing the retention ratio and converting the negative records in amount columns to positive
7. Creating two new variable's : - Concatenation(Agency ID, State and Product) and Net_Loss_profit
8. Computing the Growth Rate and Reshaping the data
9. Build the model
10. Tune the model for best accuracy and extract the important variable


Load the csv file into a Data Frame in R
Remove the unwanted Variables (Activity Notes & Commissions Dates, Primary Agency)
Define the data type of varibles
Scale the required variables to 12 Months
Remove the variables where the % of NA is greater than 30
Compute Retention Ratio and convert the -ve values in amount columns to +ve
Create two new variables
1. Concatenate - Agency ID, State and Product
2. Net profit / Loss with respect to written premium
Compute the growth rate
Reshape the data and build Random Forest model
Tune the model for best accuracy and extract the important features from the model

MODEL BUILDING

As my first attempt, I couldn't think of any algorithm better than linear regression. Since I have a multi-class categorical variable, I expected linear regression to do work. But, to my surprise, the linear regression model went in vain.

As Random Forest often perform well on imbalanced datasets, we went ahead to choose another model and it was a **Random forest**. A Quick test on Random forest has given me an accuracy of around 59% on training data.

Now I have plotted the graph of the entire variable to check whether the variables are dependent or independent and the results is all the variable is independent so I just took 2011 data for prediction

On 2011 data I have built the random forest model and it gave the accuracy of 89% which is better than the previous model

But my aim is to extract the important variables and predict the growth rate

By using random forest we can get the important variable as well as Predict

By using R square to get the accuracy in regression

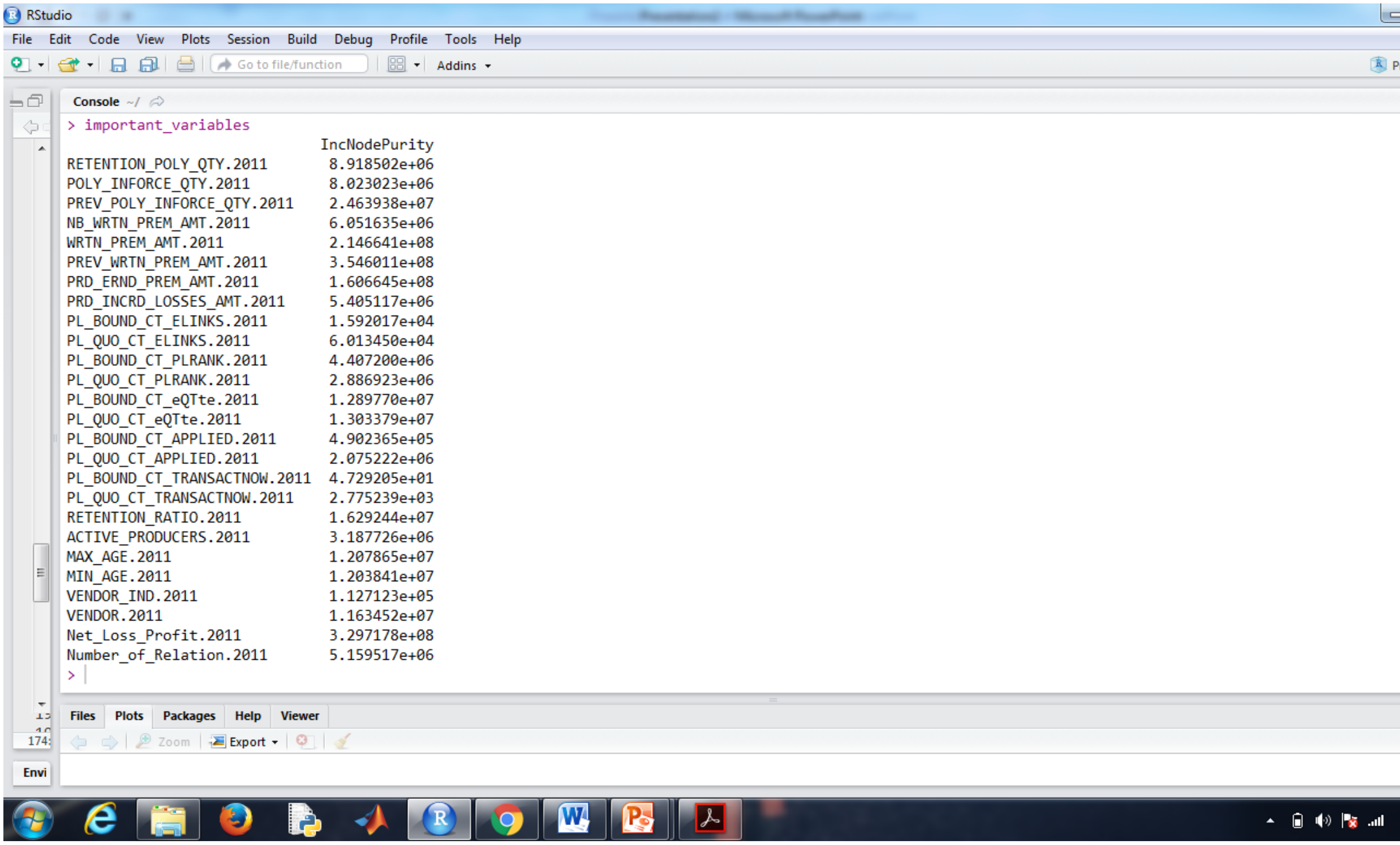
Formula:

$$R^2 = \frac{SSR}{SST} = \frac{\sum(\hat{y}_i - \bar{y})^2}{\sum(y_i - \bar{y})^2}$$

Results

Outputs:-

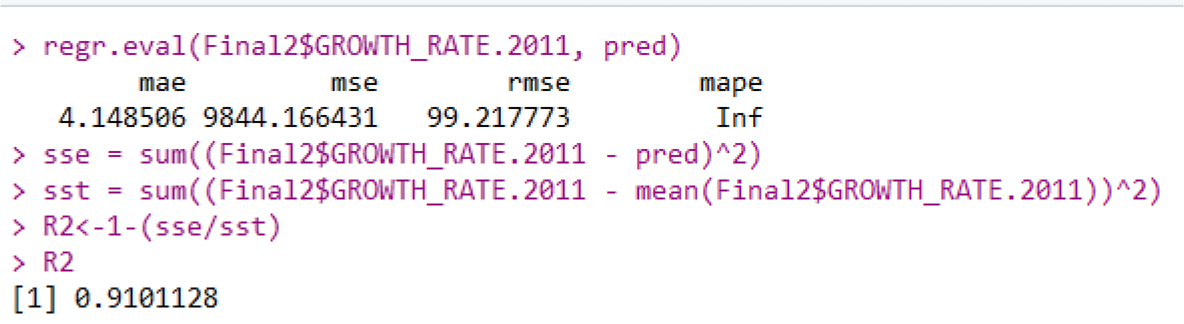
Important variables for growth rate



RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
Go to file/function
Addins
Console ~/
Files Plots Packages Help Viewer
Zoom
Export
Envi

```
> important_variables
                              IncNodePurity
RETENTION_POLY_QTY.2011        8.918502e+06
POLY_INFORCE_QTY.2011          8.023023e+06
PREV_POLY_INFORCE_QTY.2011     2.463938e+07
NB_WRTN_PREM_AMT.2011          6.051635e+06
WRTN_PREM_AMT.2011             2.146641e+08
PREV_WRTN_PREM_AMT.2011        3.546011e+08
PRD_ERND_PREM_AMT.2011         1.606645e+08
PRD_INCRD_LOSSES_AMT.2011      5.405117e+06
PL_BOUND_CT_ELINKS.2011        1.592017e+04
PL_QUO_CT_ELINKS.2011          6.013450e+04
PL_BOUND_CT_PLRANK.2011        4.407200e+06
PL_QUO_CT_PLRANK.2011          2.886923e+06
PL_BOUND_CT_eQTte.2011         1.289770e+07
PL_QUO_CT_eQTte.2011           1.303379e+07
PL_BOUND_CT_APPLIED.2011       4.902365e+05
PL_QUO_CT_APPLIED.2011         2.075222e+06
PL_BOUND_CT_TRANSACTNOW.2011  4.729205e+01
PL_QUO_CT_TRANSACTNOW.2011     2.775239e+03
RETENTION_RATIO.2011           1.629244e+07
ACTIVE_PRODUCERS.2011          3.187726e+06
MAX_AGE.2011                   1.207865e+07
MIN_AGE.2011                   1.203841e+07
VENDOR_IND.2011                1.127123e+05
VENDOR.2011                    1.163452e+07
Net_Loss_Profit.2011           3.297178e+08
Number_of_Relation.2011        5.159517e+06
>
```

Accuracy:-

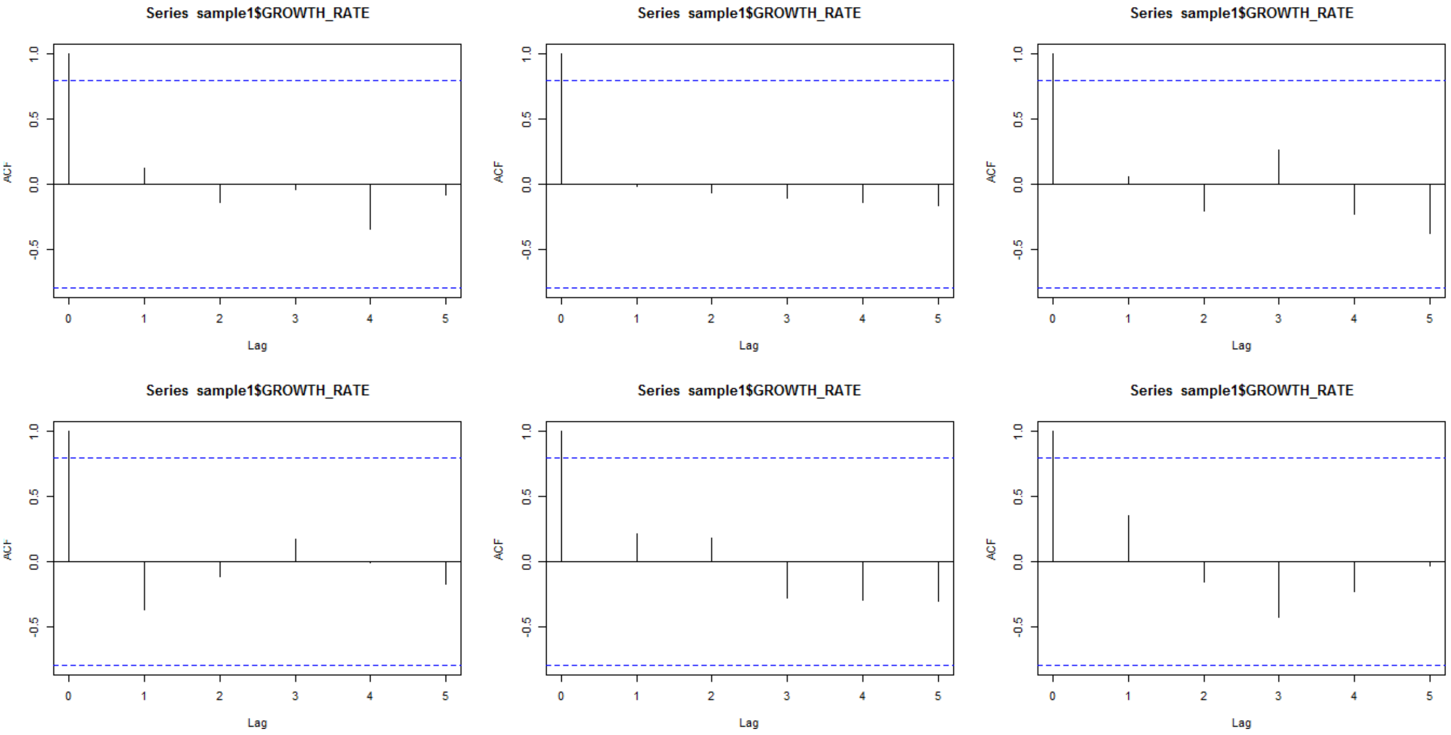


```
> regr.eval(Final2$GROWTH_RATE.2011, pred)
        mae         mse        rmse        mape
   4.148506 9844.166431   99.217773         Inf
> sse = sum((Final2$GROWTH_RATE.2011 - pred)^2)
> sst = sum((Final2$GROWTH_RATE.2011 - mean(Final2$GROWTH_RATE.2011))^2)
> R2<-1-(sse/sst)
> R2
[1] 0.9101128
```

Analysis

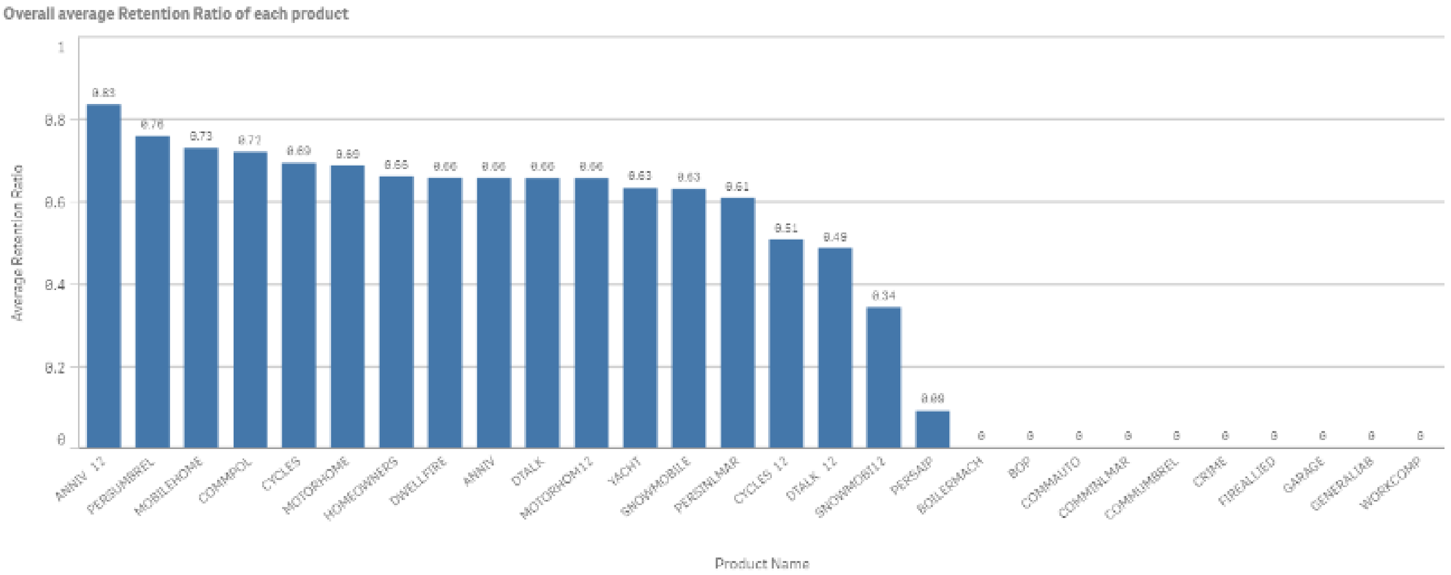
Variable Dependency:-

This graphs weather the Variable is dependent or independent
As we can see in ACF first line is default and no variable is dependent



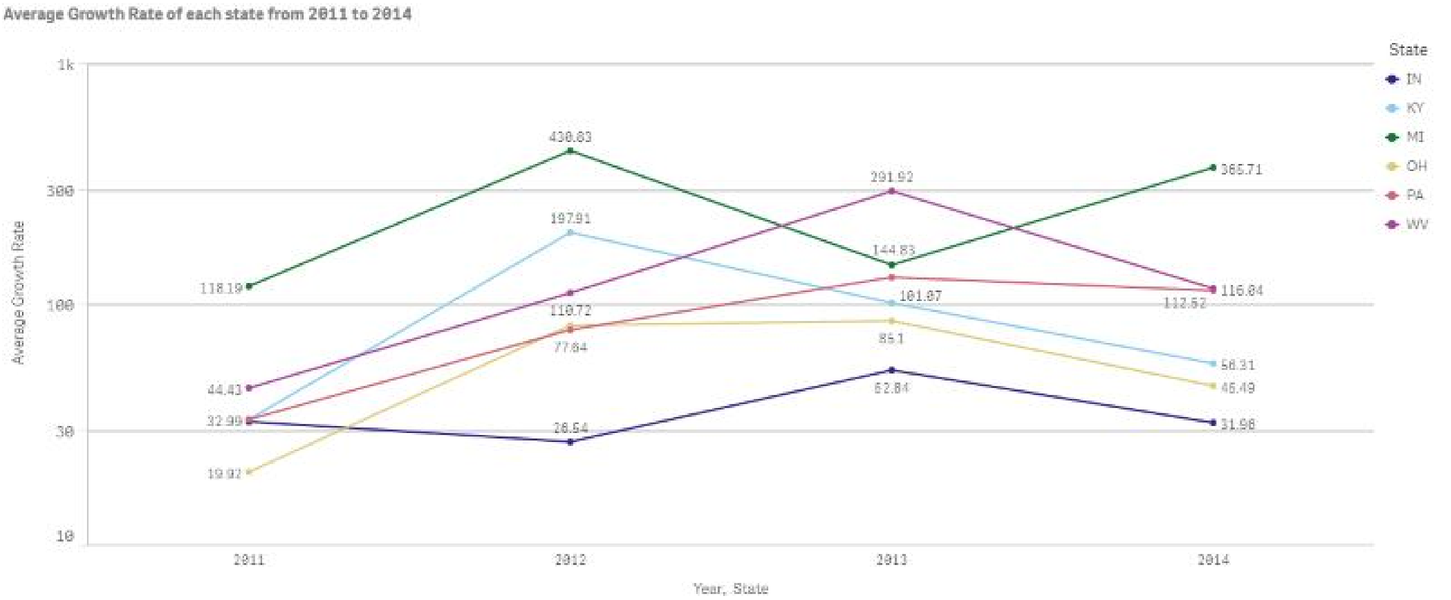
Retention Ratio:-

This graph shows the Overall average Retention Ratio of each product



Growth Rate:-

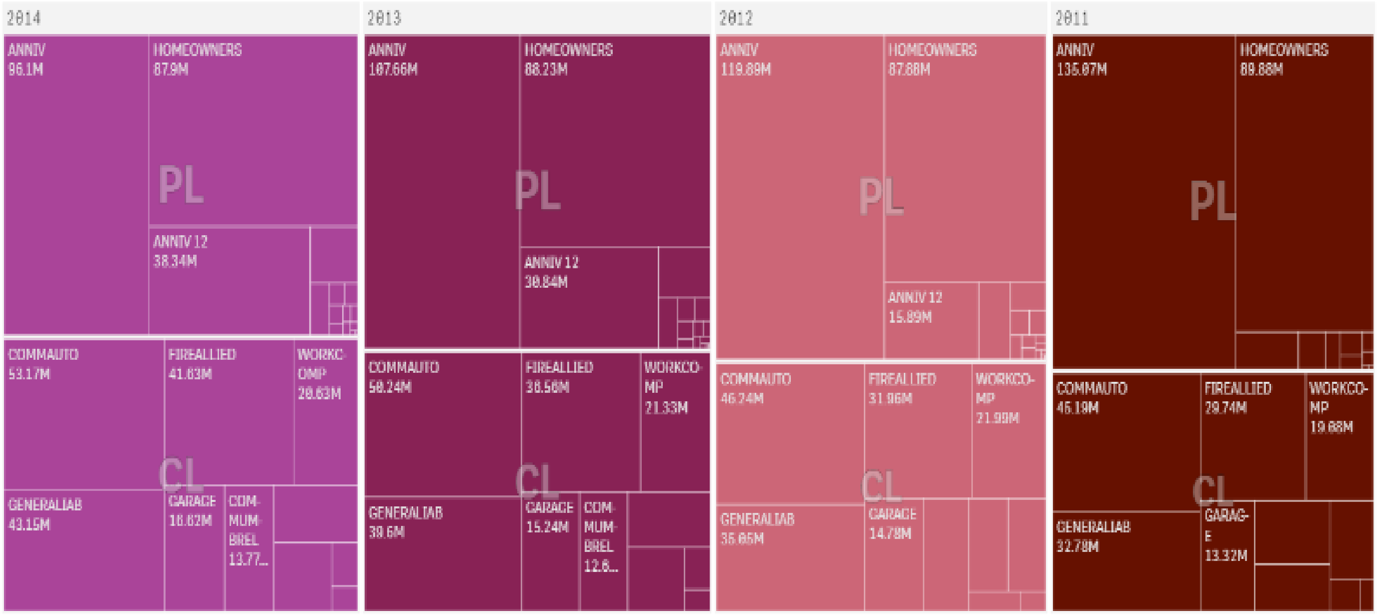
This graphs shows the growth rate of each state from 2011 to 2014



Product Line and total written premium:-

This graph displays total contribution of each product towards the total written premium

Contribution of each product towards total written premium *



* The data set contains negative or zero values that cannot be shown in this chart.

APPENDIX

The copy of R code is attached below

```
rm(list=ls())
setwd("C:/Users/pratik/Desktop/final project/AgencyPerformance/AgentPerformance")
data<-read.csv("agency_final.csv",header = TRUE, sep = ",",na.strings =99999)##Loading the data
summary(data)
str(data)
data1<-data[data$STAT_PROFILE_DATE_YEAR!=2015,]#Removing 2015 year beacasue Unsufficent
data
data1<-data1[,-c(2,28,29,32,33)]##Removing a variables that are not required
#converting data type to num and categorical
x=setdiff(names(data1),c("AGENCY_ID","PROD_ABBR","PROD_LINE","STATE_ABBR","VENDOR_IND",
"VENDOR"))
y<-setdiff(names(data1),x)
num<-data1[which(names(data1)%in% x)]
cat<-data1[which(names(data1)%in% y)]
cat<- data.frame(apply(cat,2,as.factor))
num<- data.frame(apply(num,2,as.numeric))
data2<-cbind(num,cat)
sum(is.na(data2))## checking na values

#Storing the data which has to scale for 12 months
y1<-subset(data2,data2$MONTHS!=12)
y2<-subset(data2,data2$MONTHS==12)
x<-
setdiff(names(y1),c("RETENTION_POLY_QTY","POLY_INFORCE_QTY","PREV_POLY_INFORCE_QTY","
NB_WRTN_PREM_AMT","WRTN_PREM_AMT","PREV_WRTN_PREM_AMT","PRD_ERND_PREM_AM
T","PRD_INCRD_LOSSES_AMT","MONTHS","CL_BOUND_CT_MDS","CL_QUO_CT_MDS","CL_BOUND
_CT_SBZ","CL_QUO_CT_SBZ","CL_BOUND_CT_eQT","CL_QUO_CT_eQT","PL_BOUND_CT_ELINKS","
PL_QUO_CT_ELINKS","PL_BOUND_CT_PLRANK","PL_QUO_CT_PLRANK","PL_BOUND_CT_eQTte","P
L_QUO_CT_eQTte","PL_BOUND_CT_APPLIED","PL_QUO_CT_APPLIED","PL_BOUND_CT_TRANSACT
NOW","PL_QUO_CT_TRANSACTNOW"))
y<-setdiff(names(y1),x)
x<-y1[which(names(y1)%in% x)]
y<-y1[which(names(y1)%in% y)]

##Scaling the Data to 12 months
for(i in 1:nrow(y)){

  y[i,]=(y[i,]*12)/y$MONTHS[i]
}
scale_varibles<-cbind(y,x)
scaling_variable<-rbind(scale_varibles,y2)
unique(scale_varibles$PL_QUO_CT_TRANSACTNOW)
write.csv(scaling_variable,"scale_varibles(NA).csv")
#storing all the NA variables with the count of NA values
Na_variable<-data.frame(sapply(scaling_variable,function(x)sum(is.na(x))))
Na_variable$Percentage =
(Na_variable$sapply.scaling_variable..function.x..sum.is.na.x.../nrow(scaling_variable))*100
names=data.frame(names(scaling_variable))
names2=data.frame(Na_variable$sapply.scaling_variable..function.x..sum.is.na.x...,Na_variable$Per
centage)
names1=cbind(names,names2)
names(names1)[1:3] <- c("Variable Name", "Na count", "Percentage Na")
# DROPPING THE VARIABLE WHICH HAS MORE THAN 30% NA valuse
newdata<-subset(names1$`Variable Name`, names1$`Percentage Na`>30)
scale_variblesal<-subset(scaling_variable, select = -
c(CL_BOUND_CT_MDS,CL_QUO_CT_MDS,CL_BOUND_CT_SBZ,CL_QUO_CT_SBZ,CL_BOUND_CT_eQ
T,CL_QUO_CT_eQT,PL_START_YEAR,PL_END_YEAR,CL_START_YEAR,CL_END_YEAR))

#REMOVING NA VALUES AND IMPUTING SOME VALUES
library(DMwR)
```

```
scale_varibles2<-centralImputation(scale_variblesal)
sum(is.na(scale_varibles2))
write.csv(scale_varibles,"scale_varibles2.csv")

#### Computation of retention ratio
for (i in 1:nrow(scale_varibles2)){
  if(scale_varibles2$RETENTION_POLY_QTY[i] > scale_varibles2$PREV_POLY_INFORCE_QTY[i]){
    scale_varibles2$RETENTION_POLY_QTY[i] = scale_varibles2$PREV_POLY_INFORCE_QTY[i]
  }
}
write.csv(scale_varibles2,"scale_variblesret.csv")
for (i in 1:nrow(scale_varibles2)){
  if(scale_varibles2$RETENTION_POLY_QTY[i] == 0){
    scale_varibles2$RETENTION_RATIO[i] = 0
  }
  else{
    scale_varibles2$RETENTION_RATIO[i]
=scale_varibles2$RETENTION_POLY_QTY[i]/scale_varibles2$PREV_POLY_INFORCE_QTY[i]
  }
}
write.csv(scale_varibles2,"scale_variblesret1.csv")
save.image()

scale_varibles2$LOSS_RATIO_3YR<-NULL
scale_varibles2$LOSS_RATIO<-NULL
#COMPUTING NB_WRTN_PREM_AMT,PRD_INCRD_LOSS,WRTN_PREM_AMT
for(i in 1:nrow(scale_varibles2))
{
  {
    if(scale_varibles2$NB_WRTN_PREM_AMT[i] < 0)
    {
      scale_varibles2$NB_WRTN_PREM_AMT[i]= -1 * scale_varibles2$NB_WRTN_PREM_AMT[i]}
  }
  {
    if(scale_varibles2$WRTN_PREM_AMT[i] < 0)
    {
      scale_varibles2$WRTN_PREM_AMT[i] = -1 * scale_varibles2$WRTN_PREM_AMT[i]}
  }
  {
    if(scale_varibles2$PREV_WRTN_PREM_AMT[i] < 0)
    {
      scale_varibles2$PREV_WRTN_PREM_AMT[i] = -1 * scale_varibles2$PREV_WRTN_PREM_AMT[i]}
  }
  {
    if(scale_varibles2$PRD_ERND_PREM_AMT[i] < 0)
    {
      scale_varibles2$PRD_ERND_PREM_AMT[i] = -1 * scale_varibles2$PRD_ERND_PREM_AMT[i]}
  }

  {
    if(scale_varibles2$PRD_INCRD_LOSSES_AMT[i] < 0)
    {
      scale_varibles2$PRD_INCRD_LOSSES_AMT[i] = -1 *
scale_varibles2$PRD_INCRD_LOSSES_AMT[i]}
  }
}
#Creating a New vaiable of Concatenation in which we have AGENCY_ID,STATE_ABBR,PROD_ABBR
and creating net_profit_loss variable
scale_varibles2$conc<-
paste(scale_varibles2$AGENCY_ID,scale_varibles2$STATE_ABBR,scale_varibles2$PROD_ABBR)
scale_varibles2$Net_Loss_Profit<-scale_varibles2$WRTN_PREM_AMT-
scale_varibles2$PREV_WRTN_PREM_AMT
write.csv(scale_varibles2,"scale_varibles2.csv")
scale_varibles3<-scale_varibles2[,c(1:19,21:34,20)]
scale_varibles3<-scale_varibles3[order(scale_varibles3[,32],scale_varibles3[,32],decreasing = F),]
library(plyr)
```

```
scale_varibles3 = rename(scale_varibles3,c("GROWTH_RATE_3YR"="GROWTH_RATE"))
# DIVIDING INTO TRAIN AND TEST
Train<-scale_varibles3[which(scale_varibles3$STAT_PROFILE_DATE_YEAR<=2011),]
Test<-scale_varibles3[which(scale_varibles3$STAT_PROFILE_DATE_YEAR>2011),]
save.image()

#COMPUTING GROWTH RATE
for (i in 1:nrow(Train)){
 if(Train$PREV_WRTN_PREM_AMT[i] !=0){
   Train$GROWTH_RATE[i]=((Train$Net_Loss_Profit[i])/Train$PREV_WRTN_PREM_AMT[i])*100
 }
 else{
   Train$GROWTH_RATE[i] = NA
 }
}
sum(is.na(Train$GROWTH_RATE))
Train$GROWTH_RATE<-round(Train$GROWTH_RATE,digits = 2)
freq<-data.frame(table(Train$conc))
library(dplyr)
colnames(freq)<-c("conc","Frequency")
Train1<-left_join(Train,freq,by="conc")
Train2<-Train1[Train1$Frequency==7,]
Train3<-na.omit(Train2)
sum(is.na(Train3))
Train4<-Train3[-c(1),]
save.image()
write.csv(Train4,"Final.csv")
names(Train4)
#to check the dependency of variables
i=seq(from = 1, to = nrow(Train4),by = 6)
j= head(i,6)
par(mfrow=c(2,3))
sample1<-data.frame()
for (k in j){
   sample1<-Train4[k:(k+5),]
   acf(sample1$GROWTH_RATE,lag.max = 30)
}
Train5<-Train4
Train5$Number_of_Relation<-Train4$STAT_PROFILE_DATE_YEAR-
Train4$AGENCY_APPOINTMENT_YEAR
Train5$AGENCY_APPOINTMENT_YEAR<-NULL
Final_data<-Train5
Final_data$AGENCY_ID<-NULL
# all variables being independent, we took only previous year data
Final_data<-Final_data[which(Final_data$STAT_PROFILE_DATE_YEAR==2011),]
Final_data$Frequency<-NULL
Final_data$MONTHS<-NULL
Final_data$PROD_ABBR<-NULL
Final_data$PROD_LINE<-NULL
Final_data$STATE_ABBR<-NULL
sum(is.infinite(Final_data$GROWTH_RATE))
library(plyr)
Final1<-reshape(Final_data,idvar="conc",timevar = "STAT_PROFILE_DATE_YEAR",direction = "wide")
sum(is.na(Final1))
Final2<-na.omit(Final1)
sum(is.na(Final2))
save.image()
Final2$conc<-as.factor(Final2$conc)
Final2$conc<-NULL
names(Final2)
str(Final2)
#Building the model
library(randomForest)
model<-randomForest(Final2$GROWTH_RATE.2011~.,data = Final2,keep.forest =
TRUE,ntree=400,mtry=20)
important_variables<-data.frame(randomForest::importance(model))
```

```
important_variables
pred = predict(model, Final2)
library(DMwR)
regr.eval(Final2$GROWTH_RATE.2011, pred)
sse = sum((Final2$GROWTH_RATE.2011 - pred)^2)
sst = sum((Final2$GROWTH_RATE.2011 - mean(Final2$GROWTH_RATE.2011))^2)
R2<-1-(sse/sst)
R2
save.image()
#R-Square is 88.2%
library(h2o)
localh2o <- h2o.init(ip='localhost', port = 54321, max_mem_size = '1g',nthreads = 1)
model.hex <- as.h2o(localh2o, object = Final2, key = "model.hex")
#To extract features using autoencoder method
model = h2o.deeplearning(x = setdiff(colnames(model.hex), "GROWTH_RATE.2011"),
                         y = "GROWTH_RATE.2011",
                         data = model.hex,
                         hidden = c(50,50,50),
                         activation = "RectifierWithDropout",
                         input_dropout_ratio = 0.1,
                         epochs = 100,seed=123,
                         classification = F)

names(model.hex)
features <- as.data.frame.H2OParsedData(h2o.deepfeatures(model.hex [,-20], model = model))
Final_data1<-data.frame(Final2,features)
View(Final_data1)
require(randomForest)
rf_DL <- randomForest(GROWTH_RATE.2011 ~ ., data=Final_data1, keep.forest=TRUE, ntree=30)
print (rf_DL)
# importance of attributes
round(importance(rf_DL), 2)
importanceValues = data.frame(attribute=rownames(round(importance(rf_DL),
2)),MeanDecreaseGini = round(importance(rf_DL), 2))
importanceValues
importanceValues = importanceValues[order(-importanceValues$IncNodePurity),]
importanceValues


# Top 20 Important attributes
Top30ImpAttrs = as.character(importanceValues$attribute[1:30])
Top30ImpAttrs
#Final_Data2<-Top30ImpAttrs[which(names(Top30ImpAttrs)%in% Final_data)]

#Final_Data2<-subset(Final_data,select =
c("Net_Loss_Profit.2011","PREV_WRTN_PREM_AMT.2011","WRTN_PREM_AMT.2011","DF.C1","DF
.C2","DF.C6","DF.C4","PRD_ERND_PREM_AMT.2011","DF.C3","Number_of_Relation.2011","PREV_P
OLY_INFORCE_QTY.2011","DF.C5"
,"MIN_AGE.2011","POLY_INFORCE_QTY.2011","ACTIVE_PRODUCERS.2011","MAX_AGE.2011","VEN
DOR.2011","NB_WRTN_PREM_AMT.2011","PL_QUO_CT_eQTte.2011","PRD_INCRD_LOSSES_AMT.
2011","GROWTH_RATE.2011"))
Final_Data2 = subset(Final_data1,select = c(Top30ImpAttrs,"GROWTH_RATE.2011"))
rf_DL1 <- randomForest(GROWTH_RATE.2011 ~ ., data=Final_Data2, keep.forest=TRUE, ntree=400)
print(rf_DL1)
pred = predict(rf_DL1, Final_Data2)
library(DMwR)
regr.eval(Final_Data2$GROWTH_RATE.2011, pred)
sse = sum((Final_Data2$GROWTH_RATE.2011 - pred)^2)
sst = sum((Final_Data2$GROWTH_RATE.2011 - mean(Final_Data2$GROWTH_RATE.2011))^2)
R21<-1-sse/sst
R21
#R2 value is 90.4%
save.image()
```