Capstone Project - Insurance Claims

Name : Numer P

Table of Contents

1. Introduction

The main objective of the report is to explore the Insurance claims status accepted or rejected ("Insurance Claims data.xlsx") in R and generate insights about the data set. This exploration report will consist of the following,

- Importing dataset in R
- Understanding the structure of Dataset
- Graphical exploration
- Descriptive Statistics
- Fraud Predictions using Logistics Regression, CART, Random Forest.

a) Defining Problem Statement

India is a huge market for the Insurance providing companies and insurers, where the industry is bleeding in looking more fraudsters and claim without any proven evidence is increasing day to day of the insurance sector. Insurance frauds leads to INR 40,000 crores loss which makes the industry down to the contribution of country GDP and economy rates.

The datasets provides the period of Insurance Claims by the insurer policy year of a company from FY 1998-1999 to FY 2012-2013. The data mainly deals with basic insurance claims and benefits covered up to the policy coverages. Insurance claimers are made to be fraudsters on the vehicle damages and claiming the ineligible accident amount.

b) Need of the Study

The study provides the advanced predicting of the insurance claim fraudster in the vehicle insurance claims and provides the understandable concept in various variables and to provide the higher prediction in fraudster claims. The model will provide the insights to develop the claiming structure of internal policies in insurances and reduce the ineligible claims in the initial stage of the claiming disbursement. The cluster will prove the insights in middle position and existing the plans by recurring the insurance policies from the following financial years.

c) Understanding business/ Social opportunity

The insurance claims will help the customers and society to provide the better on the accident loss and the damage on the warranty vehicle parts which helps the insurer by lower the on time funds for the accident claims and the eligible insurance policy takers will have benefits in the recurring the policy funds fully for the next policy year.

2. Exploratory Data Analysis

The insurance dataset is provided with the time period from FY 1998-1999 to FY 2012-2013 and covers the policies within the time period.

- Dataset is provided with 75200 insurance observation with 32 variables.
- The target variable is provided with “accepted” and “rejected” claim status and shows of 95% claims were accepted and 5% claims were rejected which shows the data is highly imbalance.
- The variables are classified by their characteristics and nature. The policy includes the benefits offered for the insurers is Endorsement, Statutory Cover, Discount, Anti-theft, and Total Loss.
- Basic Details of the vehicles insured by the Vehicle CC, Vehicle Colour, Vehicle RTO Location. Vehicle details has covered purpose of the vehicle, vehicle driven by, driver experience, driver qualification and road type, permit code and Zone of the vehicle.
- Policy details covers the policy year, claim year, claim amount, net premium, incurred values with disbursement date, claim intimation date, accident date.

```
Classes ‘tbl_df’, ‘tbl’ and 'data.frame': 75200 obs. of  32 variables:
$ Uniquekey                       : num  20745 20752 20757 20759 20760 ...
$ Txt_Policy_Year                 : Date, format: "2011-10-04" "2011-10-04" "2011-10-04" ...
$ Boo_Endorsement                 : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
$ Txt_Location_RTA                : chr  "GOA" "AKOLA" "NAGPUR" "MUMBAI WEST" ...
$ Txt_Policy_Code                 : chr  "2" "2" "2" "2" ...
$ Txt_Class_Code                  : chr  "11" "11" "14" "14" ...
$ Txt_Zone_Code                   : chr  "36" "36" "36" "35" ...
$ Num_Vehicle_Age                 : num  7 7 2 3 4 2 4 2 4 1 ...
$ Txt_CC_PCC_GVW_Code             : chr  "50" "50" "47" "47" ...
$ Txt_Colour_Vehicle              : chr  "OTHER COLOR" "OTHER COLOR" "OTHER COLOR" "OTHER COLOR" ...
$ Num_IDV                         : num  104000 88000 29850 30000 143140 ...
$ Txt_Permit_Code                 : chr  "1" "1" "1" "1" ...
$ Txt_Nature_Goods_Code           : chr  "2" "2" "2" "2" ...
$ Txt_Road_Type_Code              : chr  "3" "3" "3" "3" ...
$ Txt_Vehicle_Driven_By_Code      : chr  "1" "1" "1" "1" ...
$ Txt_Driver_Exp_Code             : chr  "6" "6" "1" "1" ...
$ Txt_Claims_History_Code         : chr  "4" "2" "5" "6" ...
$ Txt_Driver_Qualification_Code   : chr  "2" "1" "2" "2" ...
$ Txt_Incurred_Claims_Code        : chr  "2" "5" "2" "8" ...
$ Boo_TPPD_Statutory_Cover_only   : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
$ Txt_Claim_Year                  : Date, format: "2012-10-04" "2012-10-04" "2012-10-04" ...
$ Date_Accident_Loss              : Date, format: "2011-10-14" "2011-10-08" "2011-10-12" ...
$ Txt_Place_Accident              : chr  "GOA" "AKOLA" "NAGPUR" "MUMBAI WEST" ...
$ Date_Claim_Intimation           : Date, format: "2011-10-15" "2011-10-14" "2011-10-14" ...
$ Txt_TAC_NOL_Code                : chr  "59" "59" "59" "59" ...
$ Date_Disbursement               : Date, format: NA NA "2011-12-13" ...
$ Boo_OD_Total_Loss               : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
$ DRV_CLAIM_AMT                   : num  0 0 1876 5026 0 ...
$ DRV_CLAIM_STATUS                : Factor w/ 2 levels "0","1": 2 2 1 1 2 2 1 2 1 2 ...
$ Boo_AntiTheft                   : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
$ Boo_NCB                         : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
$ Num_Net_OD_Premium              : num  5088 4712 330 405 2832 ...
```

The datasets are classified by its characteristics and observed with character variables for codes, locations and vehicle details. The numeric variables are claim amount, net premium and Incurred values. Date variables are observed with Disbursement date, accident date, claim intimation date, policy year, claim year. Factor with 2 level variables are observed from Claim Status, Endorsement, Anti-theft, Discount and Statutory Cover.

Uniquekey	Txt_Policy_Year	Boo_Endorsement	Txt_Location_RTA
Min. : 1	Length:75200	Length:75200	Length:75200
1st Qu.:36599	Class :character	Class :character	Class :character
Median :58130	Mode :character	Mode :character	Mode :character
Mean :56217			
3rd Qu.:80838			
Max. :99991			

Txt_Class_Code	Txt_Zone_Code	Num_Vehicle_Age	Txt_CC_PCC_GVW_Code
Length:75200	Length:75200	Length:75200	Length:75200
Class :character	Class :character	Class :character	Class :character
Mode :character	Mode :character	Mode :character	Mode :character

Num_IDV	Txt_Permit_Code	Txt_Nature_Goods_Code	Txt_Road_Type_Code
Min. :1.000e+00	Length:75200	Length:75200	Length:75200
1st Qu.:1.100e+05	Class :character	Class :character	Class :character
Median :2.950e+05	Mode :character	Mode :character	Mode :character
Mean :1.334e+08			
3rd Qu.:4.850e+05			
Max. :1.000e+13			

Txt_Vehicle_Driven_By_Code	Txt_Driver_Exp_Code	Txt_Claims_History_Code	Txt_Driver_Qualification_Code
Length:75200	Length:75200	Length:75200	Length:75200
Class :character	Class :character	Class :character	Class :character
Mode :character	Mode :character	Mode :character	Mode :character

Txt_Incurred_Claims_Code	Boo_TPPD_Statutory_Cover_only	Txt_Claim_Year	Txt_Policy_Code
Length:75200	Length:75200	Length:75200	Length:75200
Class :character	Class :character	Class :character	Class :character
Mode :character	Mode :character	Mode :character	Mode :character

Date_Accident_Loss	Txt_Place_Accident	Date_Claim_Intimation	Txt_TAC_NOL_Code
Min. :1999-04-05 00:00:00	Length:75200	Min. :1999-05-17 00:00:00	Length:75200
1st Qu.:2011-01-26 00:00:00	Class :character	1st Qu.:2011-02-03 00:00:00	Class :character
Median :2011-04-06 00:00:00	Mode :character	Median :2011-04-11 00:00:00	Mode :character
Mean :2011-03-28 17:48:06		Mean :2011-04-10 04:18:15	
3rd Qu.:2011-08-27 00:00:00		3rd Qu.:2011-09-07 00:00:00	
Max. :2013-12-02 00:00:00		Max. :2013-12-07 00:00:00	

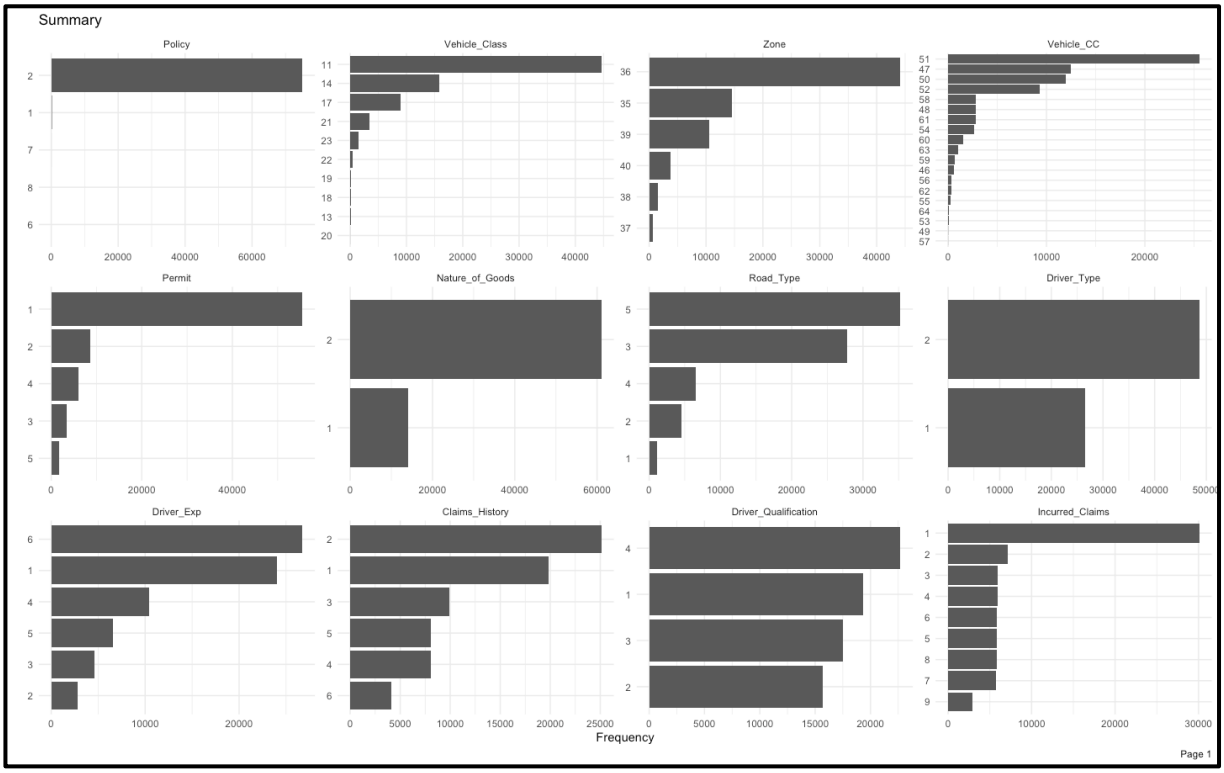
Date_Disbursement	Boo_OD_Total_Loss	DRV_CLAIM_AMT	DRV_CLAIM_STATUS
Min. :1999-06-26 00:00:00	Length:75200	Min. : 0	Length:75200
1st Qu.:2011-04-15 00:00:00	Class :character	1st Qu.: 5082	Class :character
Median :2011-05-25 00:00:00	Mode :character	Median : 13000	Mode :character
Mean :2011-06-16 08:09:35		Mean : 38986	
3rd Qu.:2011-10-15 00:00:00		3rd Qu.: 30000	
Max. :2014-01-06 00:00:00		Max. :8216000	
NA's :3735			

Boo_AntiTheft	Boo_NCB	Num_Net_OD_Premium	Txt_Colour_Vehicle
Length:75200	Length:75200	Min. : 13	Length:75200
Class :character	Class :character	1st Qu.: 1850	Class :character
Mode :character	Mode :character	Median : 4786	Mode :character
		Mean : 5501	
		3rd Qu.: 7244	
		Max. :96795	

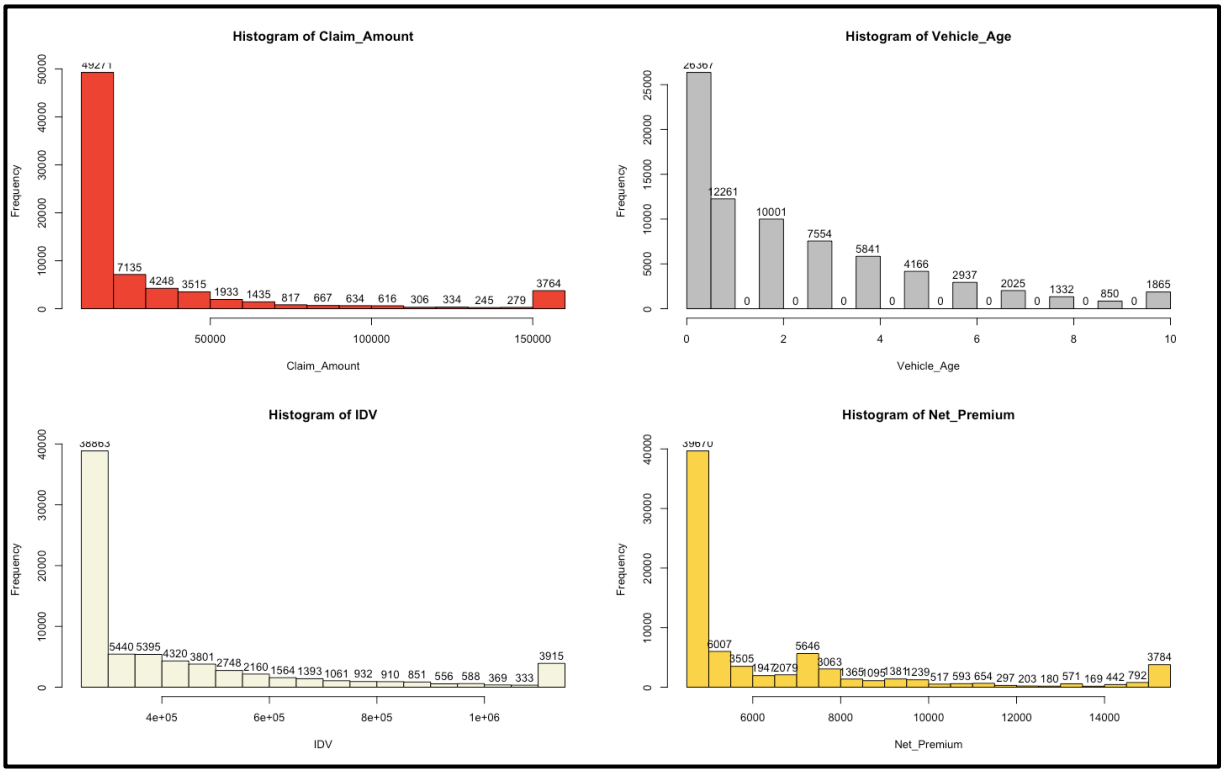
The summary shows variables with values and the provided the missing values found in Disbursement Dates and outliers are measured in Claim Amount, Net Premium and IDV.

2.1 Univariate Analysis

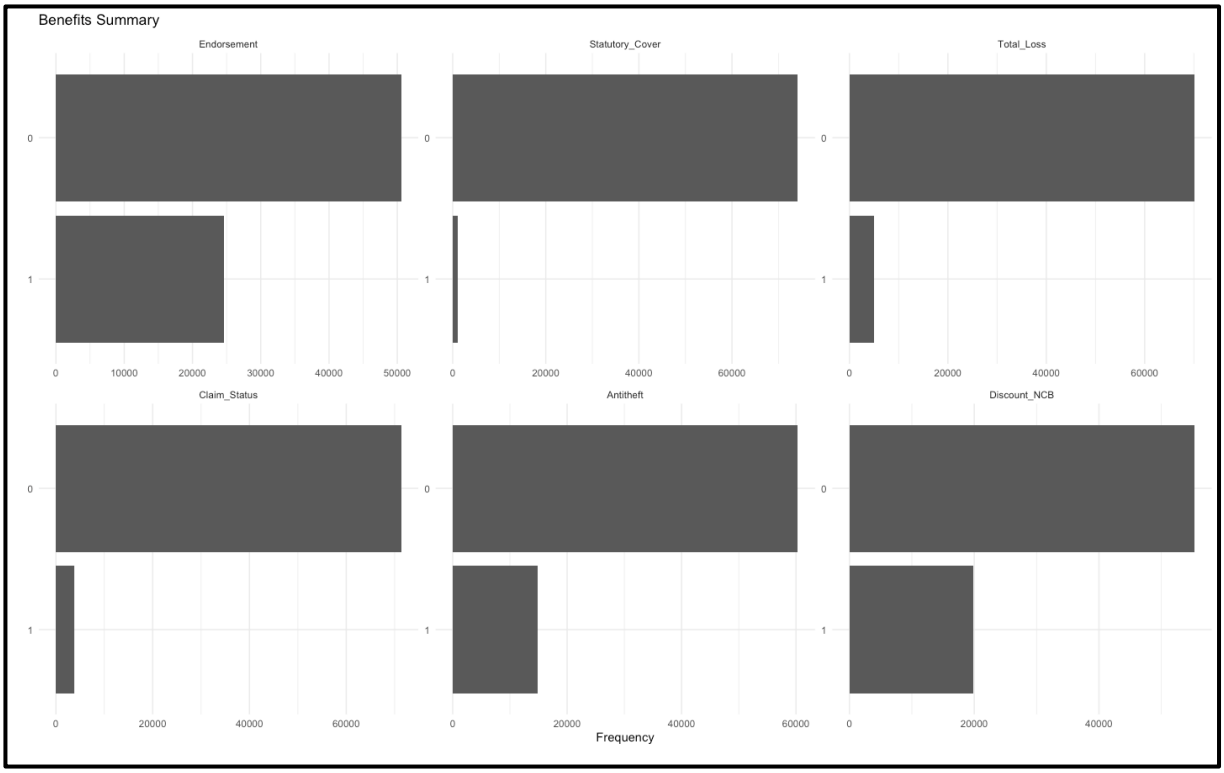
Univariate analysis is the analysis of data of one variable at time and it involves whether the datasets are descriptive or inferential statistics.



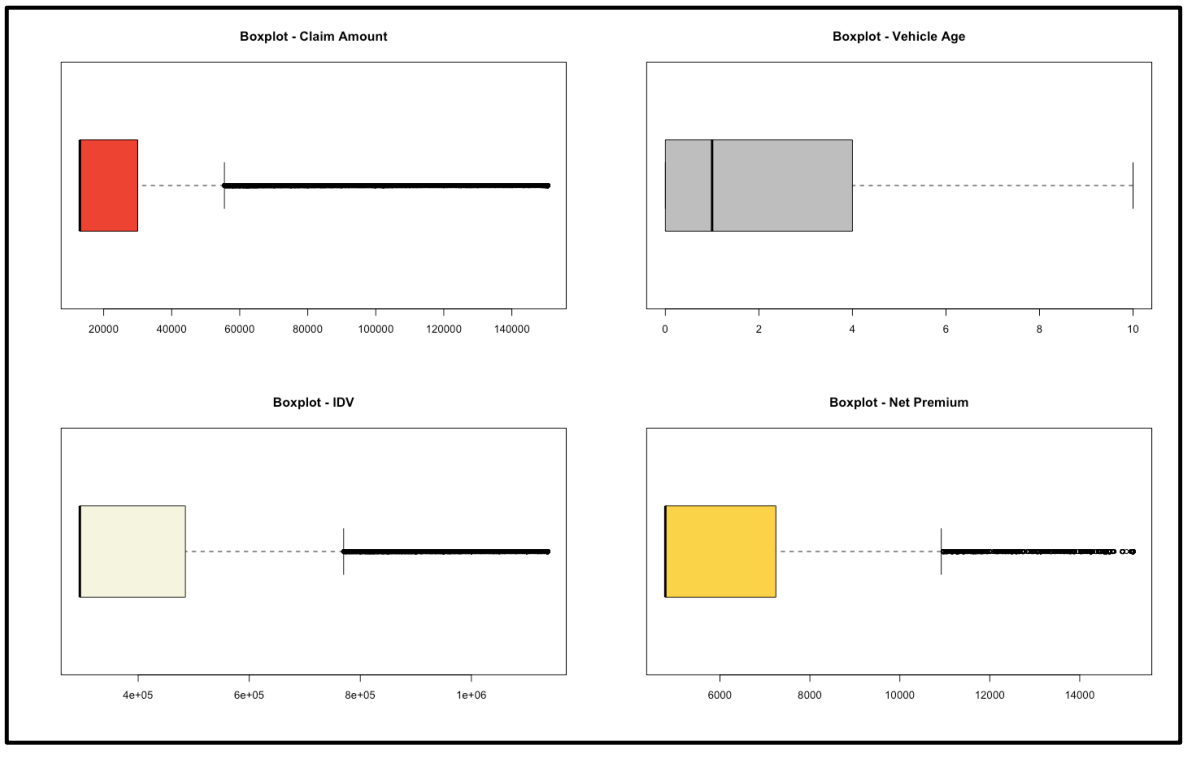
- Local Permit Code vehicles are claiming more and the road type with classified as others is claiming more which includes that insurance claims in Zone B 36 includes section 2,3,4 is claiming higher.
- The package policy insurers are measured on claiming the insurance and the most contributor in claiming the variables for the liability claims in the insurances.
- The road type is measuring for the values with others as stated that the insurance claimed vehicles are most usable in the parked status and the damages are recurred in the dent of the variables.
- The drivers other than owners are measured for the highest claiming insurance in the company. The driver experience with more than 15 years in driving experience are claiming more in the insurances.
- Driver Qualification with 10^{th} grades are claiming more and claims history shows that one time claimers are higher when compared zero time claimers.
- Vehicle with 1000CC and 1500CC which is mostly private cars are claiming more insurance when compared with others. This leads the private vehicle users are claiming more in general.



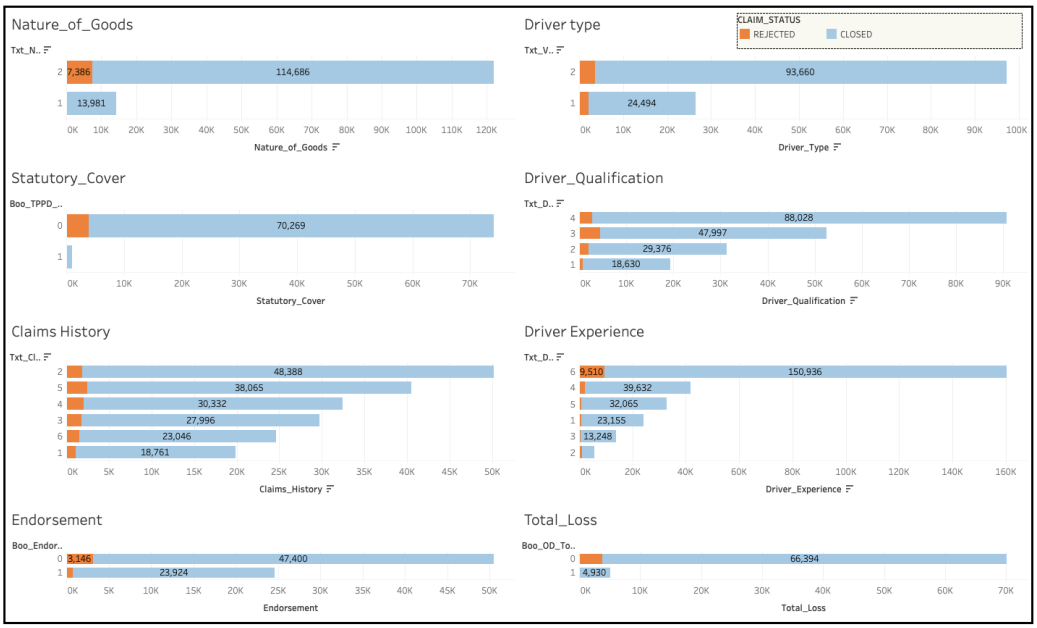
- Vehicle age is visualized with 0 vehicle age is observed with 26367 variables shows that the initial year of the insurance policy is taken and it shows insurers are claiming the most insurance with initial years and the at maximum 10 years of age is measured with 1865 in the insurance claims.
- Total Incurred Values are measured for the maximum at initial stage of the insurance claimed in the basic details for the variables and the measured values are maximum at starting of the total values with 38863 Million of the total insurance claimed amount.
- The maximum claimed amount is majorly observed with initial values for the 49 Crores in the claimed amount from the insured policies and the values are measured for the lower values in the with 2.7 crores for the total claimed items.
- Net Premium values the highest range of 396 crores and the minimum net premium generated to the insurance claims is measured of 1.8 crores.
- The categorical values are measured for the “0” and “1” levels with accepted status are higher in endorsement, targeted variables, total loss, antitheft and discount variables for the maximum accepted claims. The rejected variables are measured for the higher values in endorsement and discount in each variables.
- The additional benefits are covered with acceptance rate or included in the package is observed more and without benefits packages the policies are very less and more contribution towards insurance with covered benefits.



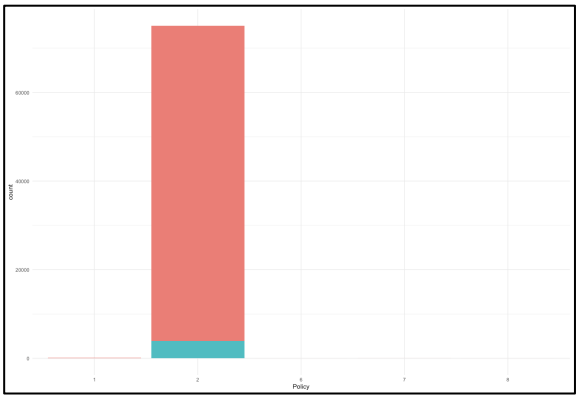
The claim amount with accepted status is measured for the higher outlier and the maximum vehicle age is converted for 10 years. The values with Incurred total values for the maximum values in accepted and rejected values are measured for the same as the rejected and accepted status. Net Premium, IDV values with 25% to 75% maximum is marked with same as accepted status and rejected status in the variables. The claim amount is mostly marked with accepted status and the measured values for the maximum outliers.

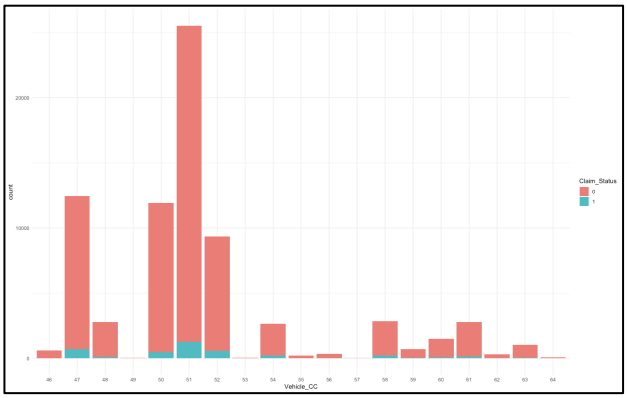


2.2 Bi-variate Analysis



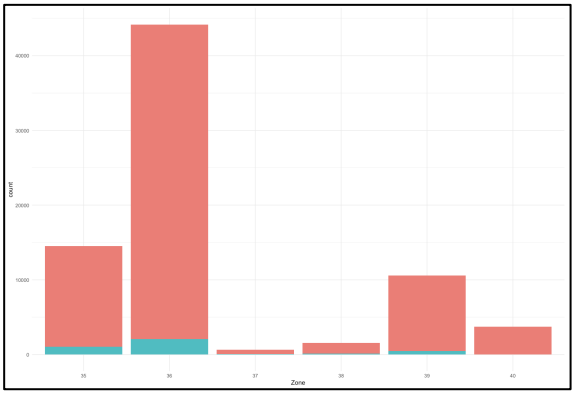
- Hazardous material carrying vehicles with no statutory covered vehicle were accepted all claims and with no total loss in damages is accepted.
- Rejected claims were observed in all fields and highest rejected claims is with driver experience more than 15 years and having claims with more than 4 times.
- Claims are more recorded for the 1-time claimers and most accepted claims were seems for the same. The rejected claims are observed higher with 4 claims then one-time claimers which is good at looking for company profit.
- Insurance claims with no endorsement benefit is observed with rejected claims is showing more frauds have been already identified.

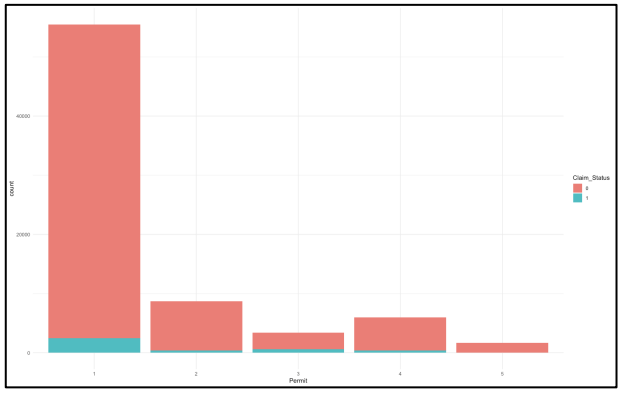




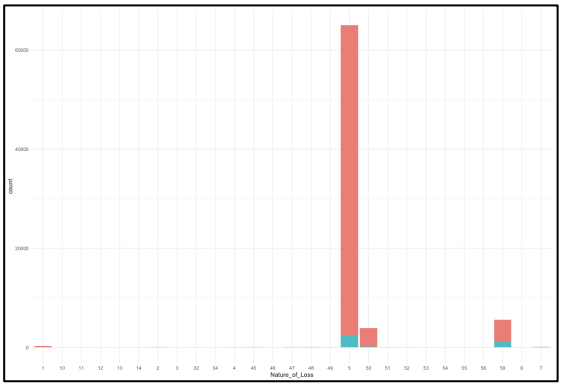
- The Policy covers with package policy is measured as highest claimer in insurance policies and the rejected status shows the very lower claimers in the package policy and the policy with liability is charged and no claims were approved to release for the liability claims.

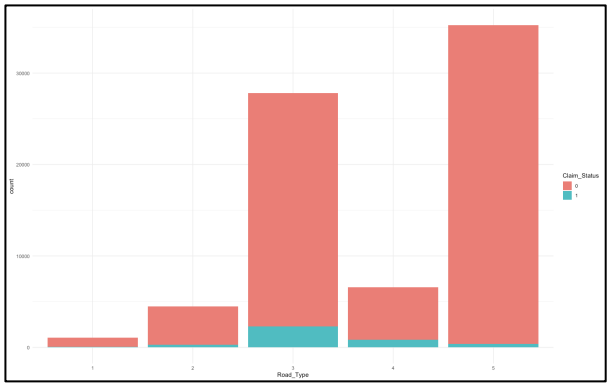
- The vehicle cc with 1000 CC to 1500 CC is measured for the private cars and the taxi for the maximum claims and the rejected claims are measured for the same. Next to 1000CC vehicles, the maximum claim is approached by two wheelers and the maximum is accepted. The vehicles like GVW vehicles are measured for the highest approved claims and the values are showing less rejected insurances for the heavy vehicles.



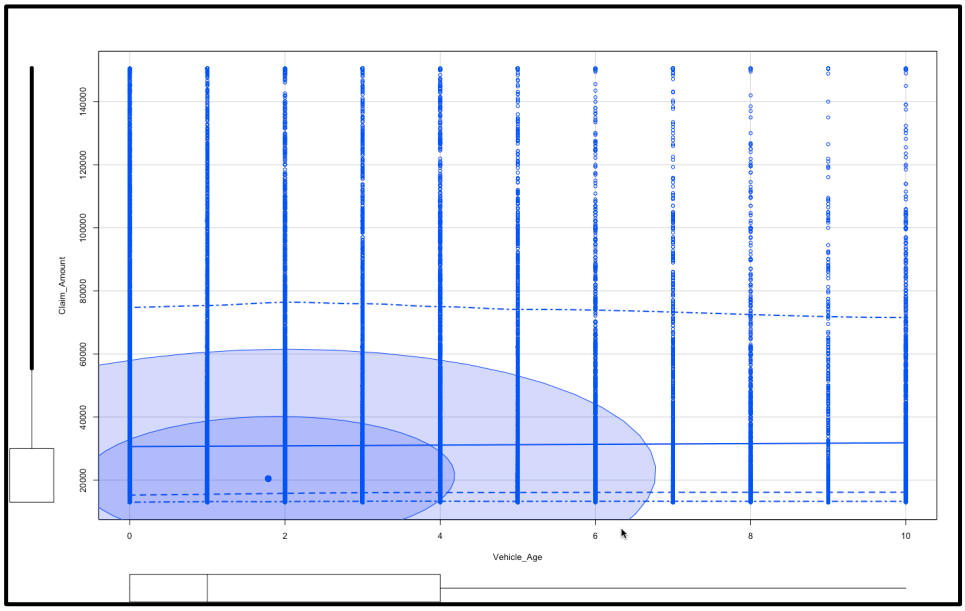


- The Zone B covered up maximum insurance claims and the Zone A positioned the second claiming status in the company and the maximum claims approved with no rejection is provided of the zone others than other zonal parts of the insurance claimed company. The maximum rejected claim status is observed with zone B and accepted claim status are measured for the other zones.
- The Local permitted vehicles are measuring with more claims other than other permitted vehicles. The highly accepted claims for the permitted vehicles is registered for the hill region and the values with claim status accepted values are marked with state vehicles and very less claims are insured. The Local vehicles status should have accompany the most increased liability values to the insurance companies.

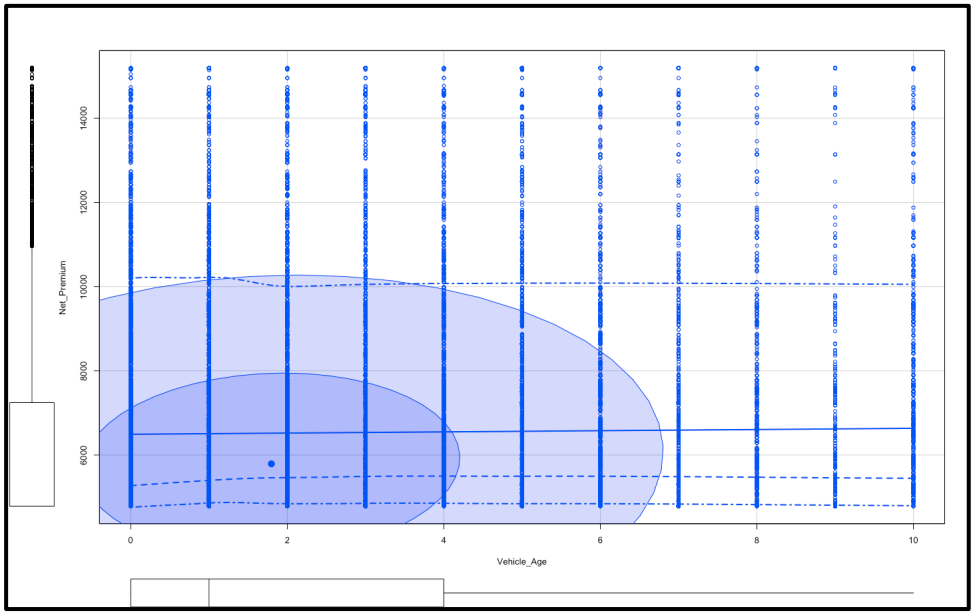




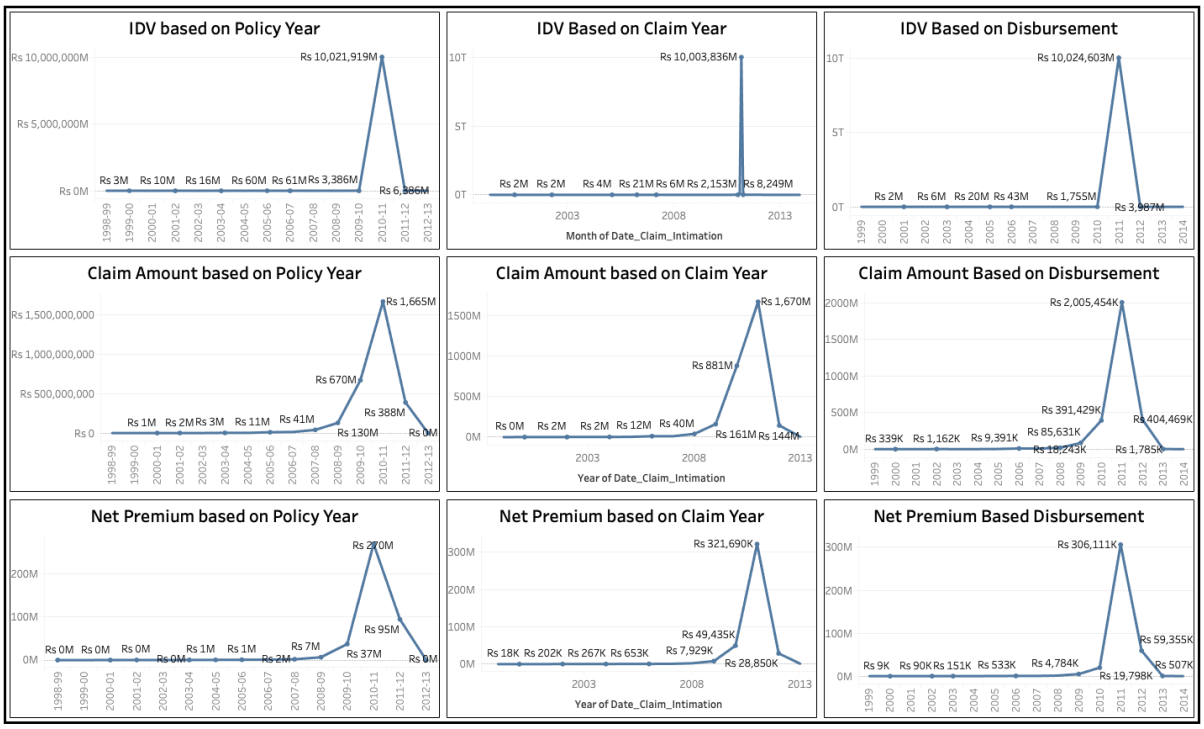
The nature of loss is measured for the accidental External are marked with higher accepted status and the values are in village roads and the categorizations shows the variables are marked with increased to decreased in rejected status. Next to nature of loss, others as reason behind highest claims in accepted as well as rejected claims of the insurance.



- The scatter plot shows the variables vehicle age with claim amount and the values are measured for the scattered towards strong value plotted in the lower ages of the vehicle and the lowest vehicle is measured for the highest age with less scattered in the values. The measured scatter shows that the marked values are controlled for making the consistency through-out all the claims increased for the variable. The eclipse is showing the maximum around 6 years of age and the values with claim amount is 600 crores to the variables.



- The scatterplot shows the values are less deviated for the age between 8 to 10 years of the vehicle age and the clutter shows the decreasing values for the values increased in the initial stage and decreasing in the net premium with increased trend in the decreased clutters in the datasets.



Time Series Plots are understand for the any trends and seasonality occurs in the variables of claim amount, net premium and Incurred Total Value of the insurance.

- The Incurred total value, net premium, claim amount is diversified by the claims and the values are mostly in measured in the year 2011 where the yearly trend is not measured and the standard normalisation of the claims in the claim amount. The amount with highest internal claim is measured for about Rs.10,029 Million is formulated in the policy year 2011-2012. The highest trend is notified in the net premium and the claim amount with the year 2011-2012.
- The month in which the claim amount is increased in March 2011 where the claim amount is increased for Rs.10,003 Millions for the Claim Year and the highest premium is measured same as the claim year 2011. The claim year is measured increased claim amount of 2009 to 2012 as a trend in the claim intimation date of the variables.
- The claim amount is increased in the year 2009 and gradually decreased from the year 2011 where the claims is noted that some observations are matched with the increased from starting of 1999 in the claim intimation.
- The net premium and the claim amount are measured for the increased trends in the year from 2008 to 2012, the lowest claim amount by the claim intimating date is observed for the 90s years.
- Disbursement dates are observed for the values are measured for the Insured Total Value and Net Premium and Claim Amount is associated for the disbursement date.

Bi-Variate Analysis – Cross Validation

	Claim_Status	
Antitheft	0	1
0	56911	3405
1	14412	471

	Claim_Status	
Statutory_Cover	0	1
0	70268	3871
1	1055	5

	Claim_Status	
Discount_NCB	0	1
0	52563	2769
1	18760	1107

	Claim_Status	
Total_Loss	0	1
0	66393	3752
1	4930	124

	Claim_Status	
Endorsement	0	1
0	47400	3146
1	23923	730

Total rejected claims in the values are measured for the 471 rejected claimers in the antitheft, 5 rejected status in the statutory claim, 730 rejected status in the endorsement claims, 124 rejected status in the total loss, where the accepted endorsement values are 47400, 66393 for the total loss, 52563 in the discount of the claims, 70268 claims accepted with statutory cover and 56911 accepted status with antitheft of the insured policy measures.

	Zone					
Claim_Status	35	36	37	38	39	40
0	13420	42058	623	1447	10102	3673
1	1075	2095	50	137	475	44

The claim status shows the higher rejected status in Zone B where the higher accepted claim status for the variables with Zone B and the values are measured for the less rejected claims in No zone.

	Policy				
Claim_Status	1	2	6	7	8
0	136	71177	1	8	1
1	6	3870	0	0	0

The policy shows the values are measured for the rejected and accepted status for the policy code with increased claims from the values with highest accepted for the package policy and no rejected status with liability with fire and theft. The values are measures for the increased in package policy and no claims were claimed for the Theft Only, Fire and Theft Only, Fire Only in the policy.

	Vehicle_Age										
Claim_Status	0	1	2	3	4	5	6	7	8	9	10
0	25271	11414	9513	7097	5514	3945	2779	1919	1263	820	1788
1	1096	847	488	457	327	221	158	106	69	30	77

The vehicle age with the accepted status are measured for the initial times of the vehicle age and the claim policy are measured for the various policy and the measured for the values in all vehicle ages and the values are increased for the accepted status and the rejected status is decreased when compared to the accepted claim status

	Vehicle_Class									
Claim_Status	11	13	14	17	18	19	20	21	22	23
0	42335	145	15006	8473	148	172	14	3214	387	1429
1	2287	0	844	461	15	9	0	146	27	87

The vehicle class with the values are measured for the no rejected claim status for the private car with less than 750 CC and the highest claim is measured for the private cars clearly indicates that the insurance policies are measured for the private usage person in the claims.

	Claims_History					
Claim_Status	1	2	3	4	5	6
0	18760	24194	9332	7583	7613	3841
1	1107	916	584	517	491	261

- The claims history is measured for the claim status and the accepted status is measured for claim in highest accepted status as 1 year claims and the values are measured for the increased values in accepted values.
- The cross tables are measured for the variables and the various levels are measured for the increased values and the making the data structures and values are measured for the increased trends in the values and the variables constructed for the model in the values.

Nature_Goods	CLOSED	REJECTED
1	Rs 982,589K	Rs 1K
2	Rs 1,949,164K	Rs 3K

Statutory_Cover	CLOSED	REJECTED
0	Rs 2,907,672K	Rs 3K
1	Rs 24,081K	Rs 0K

Total_Loss	CLOSED	REJECTED
0	Rs 2,253,219K	Rs 3K
1	Rs 678,534K	Rs 0K

Endorsement	CLOSED	REJECTED
0	Rs 2,208,029K	Rs 3K
1	Rs 723,724K	Rs 0K

AntiTheft	CLOSED	REJECTED
0	Rs 2,491,200K	Rs 3K
1	Rs 440,553K	Rs 0K

					CLAIM_STATUS	
Vehicle_Driven_By	Nature_Goods	Endorsement	Statutory_Cover	Total_Loss	CLOSED	REJECTED
1	1	0	0	0	Rs 12,209K	
				1	Rs 177K	
			1	0	Rs 41K	
				1	Rs 17K	
		1	0	0	Rs 13,730K	Rs 0K
	2	0	0	0	Rs 371,905K	Rs 1K
				1	Rs 182,430K	Rs 0K
			1	0	Rs 309K	
				1	Rs 71K	
		1	0	0	Rs 134,331K	Rs 0K
				1	Rs 13,999K	Rs 0K
			1	0	Rs 38K	
				1	Rs 46K	
2	1	0	0	0	Rs 519,120K	Rs 1K
				1	Rs 232,243K	Rs 0K
			1	0	Rs 6,808K	
				1	Rs 10,715K	
		1	0	0	Rs 135,159K	Rs 0K
				1	Rs 49,314K	Rs 0K
			1	0	Rs 2,180K	Rs 0K
				1	Rs 877K	Rs 0K
	2	0	0	0	Rs 729,962K	Rs 1K
				1	Rs 139,095K	Rs 0K
			1	0	Rs 2,428K	
				1	Rs 500K	
		1	0	0	Rs 324,947K	Rs 0K
				1	Rs 49,052K	Rs 0K
			1	0	Rs 52K	Rs 0K

The cross tables provides the rejected and accepted status in the each variables by constitute the values for the Antitheft, Discount NCB, Statutory Cover, Endorsement and Total Loss in the datasets.

3 Data Cleaning and Pre-Processing

a) Duplicate Value Treatment and Removing Redundant Variable

	Policy_Year	Endorsement	Location_RTA	Policy	Vehicle_Class	Zone	Vehicle_Age	Vehicle_CC	Vehicle_Colour	IDV
	<date>	<fct>	<chr>	<chr>	<chr>	<chr>	<dbl>	<chr>	<chr>	<dbl>
1	2011-11-07	1	IMPHAL WEST	2	11	36	0	51	OTHER COLOR	461366
2	2011-11-07	1	IMPHAL WEST	2	11	36	0	51	OTHER COLOR	461366

… with 23 more variables: Permit <chr>, Nature_of_Goods <chr>, Road_Type <chr>, Driver_Type <chr>,
Driver_Exp <chr>, Claims_History <chr>, Driver_Qualification <chr>, Incurred_Claims <chr>,
Statutory_Cover <fct>, Claim_Year <date>, Accident_Date <date>, Accident_Place <chr>,
Claim_Intimation_Date <date>, Nature_of_Loss <chr>, Disbursement_Date <date>, Total_Loss <fct>,
Claim_Amount <dbl>, Claim_Status <fct>, Antitheft <fct>, Discount_NCB <fct>, Net_Premium <dbl>,

2 Duplicate values are identified after removing Unique ID and duplicate values are removed and total observations of 75199 with 31 variables.

b) Missing Value Treatment

Missing values are observed in the Disbursement Date with 3735 values and it is treated by imputing values median values of Disbursement Date. Since, median values give the best missing value treatment in treating dates.

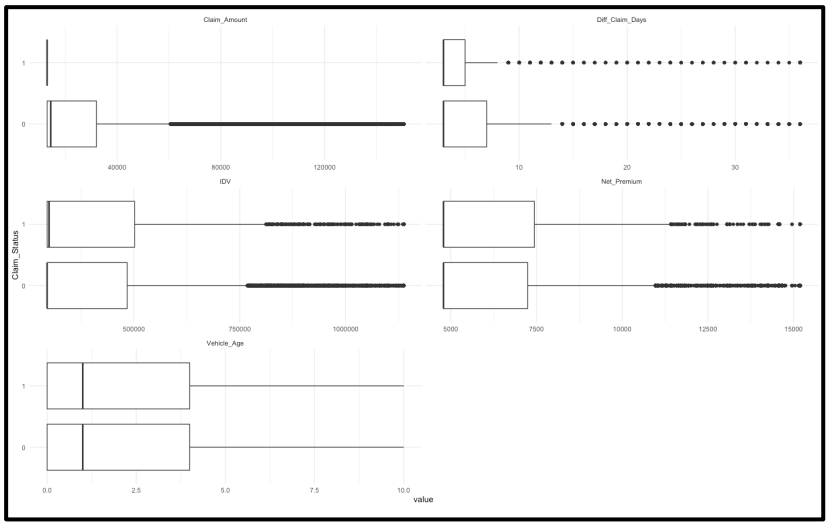
Policy_Year	Endorsement	Location_RTA	Policy
0	0	0	0
Vehicle_Class	Zone	Vehicle_Age	Vehicle_CC
0	0	0	0
Vehicle_Colour	IDV	Permit	Nature_of_Goods
0	0	0	0
Road_Type	Driver_Type	Driver_Exp	Claims_History
0	0	0	0
Driver_Qualification	Incurred_Claims	Statutory_Cover	Claim_Year
0	0	0	0
Accident_Date	Accident_Place	Claim_Intimation_Date	Nature_of_Loss
0	0	0	0
Disbursement_Date	Total_Loss	Claim_Amount	Claim_Status
3735	0	0	0
Antitheft	Discount_NCB	Net_Premium	
0	0	0	

After treatment, missing values are null in the dataset and used for further analysis of the dataset and validation of the datasets.

```
> summary(claim$Disbursement_Date)
Min.         1st Qu.      Median       Mean         3rd Qu.      Max.
1999-06-26   "2011-04-18" "2011-05-25" "2011-06-15" "2011-10-14" "2014-01-06"
```

c) Outlier Treatment

Outliers are identified in Vehicle Age, IDV, Claim Amount, Net Premium.



The numeric variables are measured for the outlier in each variables and the maximum variables are treated for the 95% quantile and minimum variables are measured for the outlier treatment with 5% for the numeric datasets.

Vehicle_Age	IDV	Claim_Amount	Net_Premium
Min. : 0.000	Min. :1.000e+00	Min. : 0	Min. : 13
1st Qu.: 0.000	1st Qu.:1.100e+05	1st Qu.: 5082	1st Qu.: 1850
Median : 1.000	Median :2.950e+05	Median : 13000	Median : 4786
Mean : 2.279	Mean :1.334e+08	Mean : 38986	Mean : 5501
3rd Qu.: 4.000	3rd Qu.:4.850e+05	3rd Qu.: 30000	3rd Qu.: 7244
Max. :29.000	Max. :1.000e+13	Max. :8216000	Max. :96795

- The maximum variables measured for the age variables is 10 and the variables are associate for the range values and the inter quantile ranges with 75% as maximum and 25% as minimum variables in the datasets.
- The outliers for vehicle age is measured for the range, quantile for the 95% and 5% is treated for the Insured total value, claim amount and Net Premium. The squish function is measured for the each variables in the maximum and minimum quantile ranges in the summary.

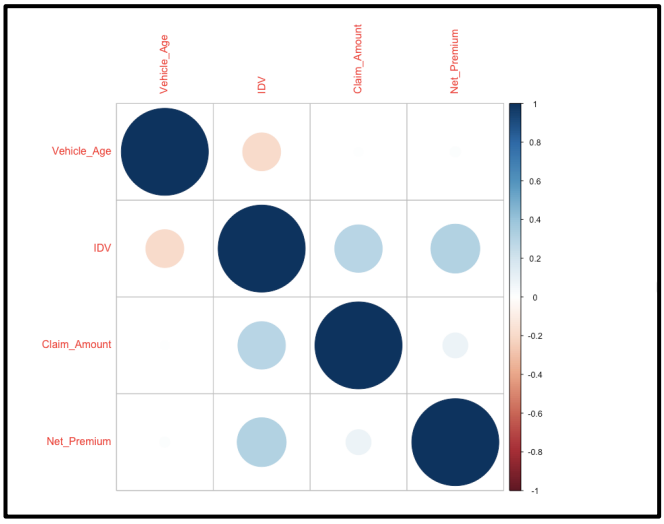
Vehicle_Age	IDV	Claim_Amount	Net_Premium
Min. : 0.000	Min. : 294980	Min. : 13000	Min. : 4786
1st Qu.: 0.000	1st Qu.: 294980	1st Qu.: 13000	1st Qu.: 4786
Median : 1.000	Median : 294980	Median : 13000	Median : 4786
Mean : 2.232	Mean : 436718	Mean : 30891	Mean : 6525
3rd Qu.: 4.000	3rd Qu.: 485000	3rd Qu.: 30000	3rd Qu.: 7244
Max. :10.000	Max. :1137192	Max. :150509	Max. :15186

e) Correlation Check and Variance Inflation Factor Rate

Correlation values are checked for the numeric variables and find the relationship between those variables. The values with -1 to +1 is measured for the correlation values with negative relationship with -1 and positive relationship with +1 values.

	Vehicle_Age	IDV	Claim_Amount	Net_Premium
Vehicle_Age	1	-0.1902678	0.00890752	0.01265259
IDV	-0.19026783	1	0.29839765	0.31654788
Claim_Amount	0.00890752	0.2983976	1	0.08275469
Net_Premium	0.01265259	0.3165479	0.08275469	1

- The correlation values are measured for -1 to +1 values. where the values with more than 0.75 are removed for the multicollinearity and the values are measured for the negative correlation for the vehicle age and IDV and the highest correlation with IDV and Net Premium for the best correlated positive values.
- Negative Values are removed from the various analysis in the models and the values are measured for the higher values are taken in correlation analysis since the variables are taken in to account for the best model approach in less than 0.75 correlated values.



The variables are not correlated as much expected and the numeric values may help in less factor for the increased values in the variables.

	Variables	VIF
1	Vehicle_Age	1.050962
2	IDV	1.286552
3	Claim_Amount	1.087727
4	Net_Premium	1.146196

The Variance Inflation Factor explains the maximum VIF factor for the analysed in the values and the VIF factor less for the numeric values are measured for the higher inflation rate is removed from the multicollinearity. Since the variables are with less VIF all numeric variables are taken for the further analysis.

4 Model Building

Dataset is split into ratio of 80:20 and the validation is used to predict the model build with the training dataset.

0	1
57058	3101

0	1
0.94845327	0.05154673

- The acceptance rate of the claim status is measured with 57058 observations and 3101 observations as rejected status.
- Proportionality of the train dataset is measured with 95% of claim acceptance and 5% of the training dataset.

0	1
14265	775

0	1
0.94847074	0.05152926

- Test validation identifies that 14265 observations in the acceptance claim rate and 775 observations as rejected claims in the period.
- Proportion of the validation dataset is predicted with 94.8% as the acceptance rate and 5% as rejected claims in the dataset.

a) Logistics Regression

- Logistics regression is used for the binary classification of **accepted claims or rejected claims** and to predict the relationship among the policies and benefits covered in the insurance policy.
- Model has been checked for the **insignificance** existed among the relationship of fields and checked for **variance inflation factor** rates.
- Model developed for the claim status to build all variables and to perform the binary classification variables in the training dataset. The model is well performed with the logit function and the highly factorized variables are removed from the model building since the Location and Vehicle will makes us to understand the basic univariate analysis of the acceptance and rejected claim status.

The **first LR model** is built with all variables except more than 50 levels (Location RTA, Accident Place and Vehicle Colour) and measured the important variables. AIC values is measured for 14090 at higher rates.

	Estimate	Std. Error	z value	Pr(>|z|)	
(Intercept)	4.18E+03	4.63E+03	0.902	0.367276	
Policy_Year	1.48E-04	1.40E-04	1.058	0.290216	
Endorsement1	-2.07E+00	9.31E-02	-22.183	< 2e-16	***
Policy2	-9.30E-01	6.50E-01	-1.431	0.152555	
Policy6	-2.76E+01	3.56E+05	0	0.999938	
Policy7	-2.18E+01	8.69E+04	0	0.9998	
Policy8	-2.54E+01	3.56E+05	0	0.999943	
Vehicle_Class13	-2.06E+01	3.21E+04	-0.001	0.999488	
Vehicle_Class14	5.01E-01	6.68E-01	0.751	0.45287	
Vehicle_Class17	1.81E+00	2.98E-01	6.075	1.24E-09	***
Vehicle_Class18	2.39E+00	5.36E-01	4.453	8.46E-06	***
Vehicle_Class19	1.44E+00	5.31E-01	2.721	0.006518	**
Vehicle_Class20	-1.87E+01	6.81E+04	0	0.99978	
Vehicle_Class21	3.23E-01	2.63E-01	1.226	0.220138	
Vehicle_Class22	1.13E+00	4.28E-01	2.645	0.008168	**
Vehicle_Class23	1.94E+00	3.20E-01	6.054	1.42E-09	***
Zone36	-2.06E-01	5.27E-02	-3.909	9.26E-05	***
Zone37	-4.60E-02	2.89E-01	-0.159	0.873344	
Zone38	1.77E-02	2.53E-01	0.07	0.944221	
Zone39	-3.83E-01	2.22E-01	-1.724	0.08475	.
Zone40	-1.22E+00	2.81E-01	-4.342	1.41E-05	***
Vehicle_Age	3.06E-02	1.02E-02	3.015	0.002574	**
Vehicle_CC47	-2.31E-01	2.68E-01	-0.862	0.388693	
Vehicle_CC48	-4.75E-01	2.90E-01	-1.639	0.101176	
Vehicle_CC49	5.31E-01	1.24E+00	0.427	0.669116	
Vehicle_CC50	6.12E-01	6.15E-01	0.996	0.319492	
Vehicle_CC51	3.84E-01	6.14E-01	0.626	0.53137	
Vehicle_CC52	8.22E-01	6.12E-01	1.344	0.178892	
Vehicle_CC53	-2.00E+01	4.14E+04	0	0.999614	
Vehicle_CC54	1.10E+00	5.78E-01	1.906	0.056685	.
Vehicle_CC55	1.19E+00	7.12E-01	1.669	0.095195	.
Vehicle_CC56	1.03E+00	6.60E-01	1.564	0.117732	
Vehicle_CC57	-2.05E+01	7.17E+04	0	0.999771	
Vehicle_CC58	-3.66E-01	5.97E-01	-0.613	0.539756	
Vehicle_CC59	-7.82E-01	6.46E-01	-1.211	0.225907	
Vehicle_CC60	-2.50E-01	6.23E-01	-0.401	0.688406	
Vehicle_CC61	5.66E-02	6.13E-01	0.092	0.926431	
Vehicle_CC62	-2.54E-01	7.43E-01	-0.341	0.733056	
Vehicle_CC63	-6.27E-01	6.47E-01	-0.97	0.331904	
Vehicle_CC64	NA	NA	NA	NA	
IDV	2.62E-07	1.47E-07	1.775	0.075875	.
Permit2	-6.55E-01	2.56E-01	-2.559	0.0105	*
Permit3	2.81E+00	2.17E-01	12.955	< 2e-16	***
Permit4	2.96E+00	1.63E-01	18.183	< 2e-16	***
Permit5	2.84E+00	3.02E-01	9.398	< 2e-16	***
Nature_of_Goods2	5.58E-01	1.35E-01	4.146	3.38E-05	***
Road_Type2	-1.95E-02	3.79E-01	-0.051	0.959013	
Road_Type3	3.09E+00	3.57E-01	8.672	< 2e-16	***
Road_Type4	3.71E+00	4.12E-01	8.988	< 2e-16	***
Road_Type5	7.82E-01	3.43E-01	2.28	0.022608	*
Driver_Type2	-4.45E-01	6.46E-02	-6.886	5.72E-12	***
Driver_Exp2	4.81E-01	1.98E-01	2.434	0.014939	*
Driver_Exp3	3.59E-01	1.13E-01	3.189	0.001426	**
Driver_Exp4	4.60E-02	8.23E-02	0.559	0.576012	
Driver_Exp5	-3.25E-01	1.11E-01	-2.921	0.003486	**
Driver_Exp6	7.60E-01	6.34E-02	11.996	< 2e-16	***
Claims_History2	6.29E-01	8.10E-02	7.77	7.87E-15	***
Claims_History3	6.49E-01	9.21E-02	7.048	1.82E-12	***
Claims_History4	6.49E-01	9.45E-02	6.867	6.55E-12	***
Claims_History5	5.88E-01	9.53E-02	6.177	6.55E-10	***
Claims_History6	6.70E-01	1.12E-01	5.974	2.32E-09	***
Driver_Qualification2	3.34E-01	7.07E-02	4.719	2.37E-06	***
Driver_Qualification3	4.64E-01	7.04E-02	6.595	4.26E-11	***
Driver_Qualification4	3.70E-01	7.84E-02	4.716	2.40E-06	***
Incurred_Claims2	-3.83E-02	8.73E-02	-0.439	0.660699	
Incurred_Claims3	7.93E-02	9.29E-02	0.853	0.393804	
Incurred_Claims4	8.71E-02	9.27E-02	0.94	0.347449	
Incurred_Claims5	-2.34E-02	9.49E-02	-0.247	0.805277	
Incurred_Claims6	-3.74E-02	9.59E-02	-0.39	0.69656	
Incurred_Claims7	-1.94E-02	9.59E-02	-0.202	0.839599	
Incurred_Claims8	1.48E-01	9.30E-02	1.589	0.112056	
Incurred_Claims9	1.47E-01	1.18E-01	1.247	0.21228	
Statutory_Cover1	-1.44E+00	5.41E-01	-2.66	0.007814	**
Claim_Year	4.41E-04	1.36E-04	3.24	0.001193	**
Accident_Date	-9.84E-05	3.24E-04	-0.304	0.761398	
Claim_Intimation_Date	-1.13E-03	3.38E-04	-3.337	0.000848	***
Nature_of_Loss10	4.56E+03	3.88E+07	0	0.999906	
Nature_of_Loss11	5.46E+02	1.27E+05	0.004	0.996557	
Nature_of_Loss12	2.79E+04	2.17E+05	0.129	0.897752	
Nature_of_Loss14	2.84E+04	2.18E+05	0.13	0.896258	
Nature_of_Loss2	4.17E-01	8.48E-01	0.491	0.623113	
Nature_of_Loss3	-2.07E+01	3.56E+05	0	0.999954	
Nature_of_Loss32	1.60E+00	9.92E-01	1.61	0.107359	
Nature_of_Loss34	2.22E+03	3.56E+05	0.006	0.995024	
Nature_of_Loss4	1.26E+00	1.24E+00	1.018	0.308598	
Nature_of_Loss45	1.14E-01	6.64E-01	0.172	0.863737	
Nature_of_Loss47	8.23E-01	7.16E-01	1.151	0.249871	
Nature_of_Loss48	1.11E+00	6.65E-01	1.662	0.096546	.
Nature_of_Loss49	1.62E+00	8.77E-01	1.848	0.064563	.
Nature_of_Loss5	-2.90E-01	3.57E-01	-0.81	0.41772	
Nature_of_Loss50	7.76E-01	3.78E-01	2.056	0.039753	*
Nature_of_Loss51	6.32E-01	1.05E+00	0.602	0.547402	
Nature_of_Loss52	2.29E-01	1.41E+00	0.162	0.871413	
Nature_of_Loss53	-2.38E+01	1.08E+05	0	0.999825	
Nature_of_Loss54	-2.25E+01	2.49E+05	0	0.999928	
Nature_of_Loss55	-1.66E+01	5.58E+04	0	0.999762	
Nature_of_Loss56	-2.31E+01	1.14E+05	0	0.999838	
Nature_of_Loss59	5.24E-01	3.70E-01	1.417	0.156401	
Nature_of_Loss6	6.18E+02	3.56E+05	0.002	0.998616	
Nature_of_Loss7	-2.12E+01	3.21E+04	-0.001	0.999472	
Disbursement_Date	1.40E-04	1.60E-04	0.874	0.3819	
Total_Loss1	7.70E-01	1.40E-01	5.513	3.53E-08	***
Claim_Amount	-3.21E-01	3.57E-01	-0.901	0.367492	
Antitheft1	-3.95E-02	1.69E-01	-0.234	0.814786	
Discount_NCB1	NA	NA	NA	NA	
Net_Premium	2.46E-05	9.98E-06	2.462	0.013805	*

The Significant variables are identified in Endorsement, Vehicle Class, Vehicle Age, Permit, Nature of Goods, Driver Experience, Claims History, Driver Qualification, Road Type, Statutory cover, Total Loss and Net Premium with AIC value of 14090 which is highly predicted for the test datasets.

2nd LR Model – The model is again build with significant variables of previous model and predicted the higher significant variables which creates more relationship with the claim status variables. The summary of model is defined by,

	Estimate	Std. Error	z value	Pr(>|z|)	
(Intercept)	-2.45E+00	1.40E+00	-1.752	0.079751	.
Endorsement1	-1.68E+00	8.22E-02	-20.48	< 2e-16	***
Vehicle_Class13	-9.60E+00	1.37E+02	-0.07	0.944263	
Vehicle_Class14	3.32E-01	5.70E-02	5.825	5.70E-09	***
Vehicle_Class17	4.05E-01	1.83E-01	2.21	0.027106	*
Vehicle_Class18	9.45E-01	3.76E-01	2.515	0.011896	*
Vehicle_Class19	8.95E-01	4.28E-01	2.089	0.036737	*
Vehicle_Class20	-1.18E+01	3.96E+02	-0.03	0.976178	
Vehicle_Class21	5.39E-02	1.60E-01	0.337	0.736024	
Vehicle_Class22	3.30E-01	2.59E-01	1.271	0.203663	
Vehicle_Class23	5.13E-01	2.22E-01	2.309	0.020923	*
Zone36	-2.55E-01	4.72E-02	-5.399	6.68E-08	***
Zone37	4.57E-01	2.30E-01	1.99	0.046571	*
Zone38	3.76E-01	1.98E-01	1.904	0.056929	.
Zone39	-2.72E-01	1.78E-01	-1.53	0.126034	
Zone40	-9.12E-01	2.22E-01	-4.107	4.01E-05	***
Vehicle_Age	3.18E-02	8.25E-03	3.853	0.000117	***
Permit2	-1.23E+00	2.33E-01	-5.276	1.32E-07	***
Permit3	2.30E+00	1.84E-01	12.489	< 2e-16	***
Permit4	2.64E+00	1.40E-01	18.795	< 2e-16	***
Permit5	2.69E+00	2.72E-01	9.861	< 2e-16	***
Nature_of_Goods2	6.35E-01	1.09E-01	5.826	5.66E-09	***
Road_Type2	5.19E-01	3.36E-01	1.548	0.121724	
Road_Type3	3.16E+00	3.22E-01	9.818	< 2e-16	***
Road_Type4	4.35E+00	3.74E-01	11.627	< 2e-16	***
Road_Type5	8.93E-01	3.10E-01	2.882	0.003949	**
Driver_Type2	-6.28E-01	5.43E-02	-11.558	< 2e-16	***
Driver_Exp2	3.04E-01	1.72E-01	1.768	0.077057	.
Driver_Exp3	8.25E-02	1.03E-01	0.804	0.421144	
Driver_Exp4	-8.79E-03	7.24E-02	-0.121	0.903464	
Driver_Exp5	-4.10E-01	1.03E-01	-4.004	6.24E-05	***
Driver_Exp6	7.24E-01	5.62E-02	12.895	< 2e-16	***
Claims_History2	5.98E-01	6.90E-02	8.666	< 2e-16	***
Claims_History3	6.45E-01	7.84E-02	8.231	< 2e-16	***
Claims_History4	6.41E-01	8.06E-02	7.961	1.71E-15	***
Claims_History5	5.75E-01	8.08E-02	7.117	1.10E-12	***
Claims_History6	6.09E-01	9.65E-02	6.312	2.75E-10	***
Driver_Qualification2	3.58E-01	6.39E-02	5.598	2.17E-08	***
Driver_Qualification3	4.50E-01	6.33E-02	7.108	1.17E-12	***
Driver_Qualification4	3.69E-01	7.03E-02	5.248	1.54E-07	***
Statutory_Cover1	-1.03E+00	5.14E-01	-2.007	0.044724	*
Claim_Year	1.27E-03	1.06E-04	12.022	< 2e-16	***
Claim_Intimation_Date	-1.57E-03	1.32E-04	-11.875	< 2e-16	***
Total_Loss1	-1.38E+00	1.13E-01	-12.2	< 2e-16	***
Net_Premium	3.08E-05	7.45E-06	4.138	3.50E-05	***

The significant variables is predicted with some levels are insignificant, hence the model is unfit in predicting the binary operations of the variables.

Min	1Q	Median	3Q	Max
-2.5566	-0.3332	-0.179	-0.076	4.242

The minimum residuals predicted for the variables is observed with -2.5 and the maximum values predicted for the 4.2 as the residuals in the developed and the model is predicted with Variance Inflation Factor to get the higher significance values provided in the datasets.

	GVIF	Df	GVIF^(1/(2*Df))
Endorsement	2.498499	1	1.580664
Vehicle_Class	22.629927	9	1.189214
Zone	16.242116	5	1.321491
Vehicle_Age	1.126621	1	1.061424
Permit	764.03159	4	2.29292
Nature_of_Goods	1.597515	1	1.263928
Road_Type	370.169808	4	2.094354
Driver_Type	1.910244	1	1.382116
Driver_Exp	12.757951	5	1.289965
Claims_History	2.341209	5	1.08879
Driver_Qualification	1.769958	3	1.099834
Statutory_Cover	1.016426	1	1.008179
Claim_Year	3.140246	1	1.772074
Claim_Intimation_Date	3.360403	1	1.83314
Total_Loss	1.21175	1	1.100795
Net_Premium	1.188068	1	1.089985

Permit and Road Type is removed from the further analysis of the Logistics Regression Model, since this variables are contributing higher variance. VIF values with 1 to 5 is taken for the best variance factor with the claim status variable.

3rd LR Model – The model is again built up with the variables have higher variance inflation factor is removed and the variance with less variance is removed from the further model developed.

The model summarizes with the variables are measured for the errors and the variables are measured for the significance level in the model.

Min	1Q	Median	3Q	Max
-2.6765	-0.3308	-0.227	-0.153	4.094

The variables with minimum value Is predicted for the -2.67 and the higher error is measured for the 4.09 in the significant variables.

The variables are predicted for the significant variables for Endorsement, Claims History, Driver Qualification, Claim Year, Claim Intimation Date, Net Premium, Nature of Goods, Driver Experience to make the higher significant variables to predict the developed model further in the analysis.

	Estimate	Std. Error	z value	Pr(>|z|)	
(Intercept)	9.02E-01	1.22E+00	0.739	0.459605	
Endorsement1	-1.44E+00	5.84E-02	-24.686	< 2e-16	***
Vehicle_Class13	-1.08E+01	8.31E+01	-0.13	0.896329	
Vehicle_Class14	1.42E-01	5.46E-02	2.596	0.009422	**
Vehicle_Class17	3.71E-01	1.67E-01	2.218	0.026551	*
Vehicle_Class18	1.03E+00	3.60E-01	2.858	0.004261	**
Vehicle_Class19	4.54E-01	4.06E-01	1.118	0.263676	
Vehicle_Class20	-1.15E+01	2.46E+02	-0.047	0.962889	
Vehicle_Class21	-4.74E-01	1.55E-01	-3.067	0.002165	**
Vehicle_Class22	2.03E-01	2.49E-01	0.814	0.415791	
Vehicle_Class23	1.74E-01	2.13E-01	0.819	0.41301	
Zone36	-3.29E-01	4.59E-02	-7.174	7.30E-13	***
Zone37	5.81E-01	2.26E-01	2.571	0.010151	*
Zone38	5.16E-01	1.89E-01	2.728	0.006378	**
Zone39	-2.26E-01	1.67E-01	-1.353	0.176196	
Zone40	-1.58E+00	2.17E-01	-7.267	3.67E-13	***
Vehicle_Age	-3.49E-04	7.99E-03	-0.044	0.965113	
Nature_of_Goods2	1.67E+00	1.03E-01	16.276	< 2e-16	***
Driver_Type2	-3.63E-01	4.62E-02	-7.862	3.79E-15	***
Driver_Exp2	2.36E+00	8.37E-02	28.149	< 2e-16	***
Driver_Exp3	1.22E-01	9.94E-02	1.232	0.21804	
Driver_Exp4	2.87E-01	6.85E-02	4.195	2.73E-05	***
Driver_Exp5	-2.51E-01	1.00E-01	-2.507	0.012185	*
Driver_Exp6	8.04E-01	5.45E-02	14.761	< 2e-16	***
Claims_History2	7.98E-02	6.03E-02	1.322	0.186036	
Claims_History3	2.32E-01	6.95E-02	3.335	0.000853	***
Claims_History4	2.63E-01	7.23E-02	3.638	0.000275	***
Claims_History5	2.29E-01	7.27E-02	3.157	0.001593	**
Claims_History6	2.37E-01	8.90E-02	2.657	0.007872	**
Driver_Qualification2	3.88E-01	6.22E-02	6.239	4.41E-10	***
Driver_Qualification3	7.13E-01	5.90E-02	12.094	< 2e-16	***
Driver_Qualification4	8.76E-02	6.98E-02	1.255	0.209416	
Statutory_Cover1	-1.46E+00	5.08E-01	-2.87	0.004101	**
Claim_Year	1.73E-03	1.09E-04	15.896	< 2e-16	***
Claim_Intimation_Date	-2.12E-03	1.22E-04	-17.424	< 2e-16	***
Total_Loss1	-1.01E+00	1.12E-01	-9.004	< 2e-16	***
Net_Premium	2.93E-05	7.14E-06	4.103	4.08E-05	***

The model provides the significant variables after removing the less significant variables and predicted the significant variables.

4th LR Model – The model is built with the significant variables and the predicted significance variables are measured for the various model development in the prediction of the training variables with the values in the model. The binary model is predicted for the significant variables in the dataset.

	Estimate	Std. Error	z value	Pr(>|z|)	
(Intercept)	9.06E-01	1.22E+00	0.746	0.455867	
Endorsement1	-1.44E+00	5.83E-02	-24.748	< 2e-16	***
Vehicle_Class13	-1.08E+01	8.31E+01	-0.13	0.896332	
Vehicle_Class14	1.42E-01	5.42E-02	2.618	0.008857	**
Vehicle_Class17	3.71E-01	1.67E-01	2.218	0.026566	*
Vehicle_Class18	1.03E+00	3.60E-01	2.858	0.004261	**
Vehicle_Class19	4.54E-01	4.06E-01	1.119	0.263311	
Vehicle_Class20	-1.15E+01	2.47E+02	-0.047	0.962897	
Vehicle_Class21	-4.74E-01	1.55E-01	-3.066	0.002168	**
Vehicle_Class22	2.03E-01	2.49E-01	0.816	0.41468	
Vehicle_Class23	1.74E-01	2.13E-01	0.82	0.412291	
Zone36	-3.29E-01	4.59E-02	-7.182	6.89E-13	***
Zone37	5.80E-01	2.26E-01	2.571	0.010153	*
Zone38	5.16E-01	1.89E-01	2.727	0.006385	**
Zone39	-2.26E-01	1.67E-01	-1.353	0.175934	
Zone40	-1.58E+00	2.17E-01	-7.269	3.62E-13	***
Nature_of_Goods2	1.68E+00	1.03E-01	16.281	< 2e-16	***
Driver_Type2	-3.63E-01	4.62E-02	-7.867	3.62E-15	***
Driver_Exp2	2.36E+00	8.36E-02	28.17	< 2e-16	***
Driver_Exp3	1.22E-01	9.94E-02	1.231	0.218269	
Driver_Exp4	2.88E-01	6.85E-02	4.2	2.67E-05	***
Driver_Exp5	-2.51E-01	1.00E-01	-2.507	0.012193	*
Driver_Exp6	8.04E-01	5.44E-02	14.78	< 2e-16	***
Claims_History2	7.99E-02	6.02E-02	1.328	0.184218	
Claims_History3	2.32E-01	6.94E-02	3.343	0.00083	***
Claims_History4	2.63E-01	7.21E-02	3.648	0.000264	***
Claims_History5	2.30E-01	7.25E-02	3.167	0.001539	**
Claims_History6	2.37E-01	8.89E-02	2.663	0.007743	**
Driver_Qualification2	3.88E-01	6.22E-02	6.239	4.40E-10	***
Driver_Qualification3	7.13E-01	5.90E-02	12.095	< 2e-16	***
Driver_Qualification4	8.76E-02	6.98E-02	1.255	0.209587	
Statutory_Cover1	-1.46E+00	5.08E-01	-2.87	0.004101	**
Claim_Year	1.73E-03	1.08E-04	16.019	< 2e-16	***
Claim_Intimation_Date	-2.12E-03	1.21E-04	-17.44	< 2e-16	***
Total_Loss1	-1.01E+00	1.12E-01	-9.009	< 2e-16	***
Net_Premium	2.93E-05	7.13E-06	4.108	4.00E-05	***

The model developed and the significant variables are measured for the statutory cover, claim intimation date, claim year, total loss and net premium for the dataset and the variables are predicted for the various model in the levels of the variables, hence the values are predicted for the increased errors in the variables for the predicted values.

Min	1Q	Median	3Q	Max
-2.6761	-0.3308	-0.227	-0.153	4.094

The residuals are calculated for the quartile 25% and the minimum values are predicted for the -2.6 and the maximum values predicted for the 75% quartile to 4.094 the variables are measured for the increased predicted variables for the further development.

	GVIF	Df	GVIF^(1/(2*Df))
Endorsement	1.411806	1	1.188195
Vehicle_Class	13.685542	9	1.156447
Zone	12.955801	5	1.291952
Nature_of_Goods	1.469709	1	1.212316
Driver_Type	1.437869	1	1.199112
Driver_Exp	2.300857	5	1.086898
Claims_History	1.708474	5	1.05502
Driver_Qualification	1.547671	3	1.075507
Statutory_Cover	1.012864	1	1.006411
Claim_Year	3.13911	1	1.771753
Claim_Intimation_Date	3.249416	1	1.802614
Total_Loss	1.187876	1	1.089897
Net_Premium	1.163542	1	1.078676

The variance inflation rate for the variables above 5 is removed from the further analysis and the values are predicted for the correctly significant to the variables as multicollinearity check of the variables.

5th LR Model – The model developed for the variables with significant variables from the previous model and the variables are predicted for the binary classification with higher significant models developed with claim status of the datasets.

Min	1Q	Median	3Q	Max
-1.2753	-0.3291	-0.246	-0.174	3.951

The model developed and predicted with the higher significant variables and the minimum error to the model is predicted for the -1.2 and the maximum variables is predicted with 3.9 for the variables.

	Estimate	Std. Error	z value	Pr(>|z|)	
(Intercept)	-1.12E+01	1.17E+00	-9.604	< 2e-16	***
Endorsement1	-1.51E+00	5.55E-02	-27.21	< 2e-16	***
Driver_Type2	-3.52E-01	4.37E-02	-8.045	8.62E-16	***
Driver_Exp2	2.40E+00	8.15E-02	29.467	< 2e-16	***
Driver_Exp3	8.04E-02	9.85E-02	0.816	0.414655	
Driver_Exp4	2.05E-01	6.80E-02	3.009	0.00262	**
Driver_Exp5	-3.18E-01	9.91E-02	-3.21	0.001329	**
Driver_Exp6	8.38E-01	5.41E-02	15.487	< 2e-16	***
Nature_of_Goods2	1.49E+00	1.01E-01	14.781	< 2e-16	***
Claims_History2	2.15E-01	5.82E-02	3.702	0.000214	***
Claims_History3	4.16E-01	6.77E-02	6.148	7.87E-10	***
Claims_History4	4.65E-01	7.08E-02	6.565	5.21E-11	***
Claims_History5	4.24E-01	7.13E-02	5.943	2.79E-09	***
Claims_History6	4.32E-01	8.76E-02	4.935	8.00E-07	***
Driver_Qualification2	3.75E-01	6.21E-02	6.034	1.60E-09	***
Driver_Qualification3	7.61E-01	5.86E-02	12.988	< 2e-16	***
Driver_Qualification4	-1.18E-02	7.02E-02	-0.168	0.866203	
Statutory_Cover1	-1.45E+00	5.05E-01	-2.874	0.004053	**
Claim_Year	4.22E-04	7.75E-05	5.439	5.35E-08	***
Total_Loss1	-9.04E-01	1.08E-01	-8.342	< 2e-16	***
Net_Premium	2.66E-05	6.64E-06	4.006	6.19E-05	***

The significant variables are measured for the increased significant variables but some level is predicted with less significance hence the variables are measured for the values and the values to increase the level of the higher significance of the measured variables.

	VIF	Df	GVIF^(1/(2*Df))
Endorsement	1.306396	1	1.142977
Driver_Type	1.307629	1	1.143516
Driver_Exp	2.128184	5	1.078452
Nature_of_Goods	1.440122	1	1.200051
Claims_History	1.610613	5	1.048816
Driver_Qualification	1.58235	3	1.079486
Statutory_Cover	1.008441	1	1.004212
Claim_Year	1.230244	1	1.109164
Total_Loss	1.128452	1	1.062286
Net_Premium	1.037548	1	1.018601

The variance inflation rate variables predicted for the measure of values around 1 to 5 and the variance factor is developed with the values for the degrees of freedom in the values and the variables with less inflation rate is developed and the model is well developed for the predicting features of the datasets.

(Intercept) 0.007351712	Endorsement1 0.220820231	Driver_Type2 0.673668239	Driver_Exp2 11.0606734	Driver_Exp3 1.097598108
Driver_Exp4 1.26062265	Driver_Exp5 0.727654414	Driver_Exp6 2.465849927	Nature_of_Goods2 4.864843887	Claims_History2 1.228417072
Claims_History3 1.492327547	Claims_History4 1.574959852	Claims_History5 1.511550281	Claims_History6 1.53262893	Driver_Qualification2 1.436229927
Driver_Qualification3 2.112828174	Driver_Qualification4 0.978520361	Statutory_Cover1 0.234134711	Total_Loss1 0.340785566	Net_Premium 1.000027079

The exponential coefficient of the variables are predicted for the increased significant variables and the values are measured with higher exponential factor of 4.4 and the less exponential factors is observed with the values net premium for 1. The values are measured for predicting features of the variables.

McFadden

0.1220207

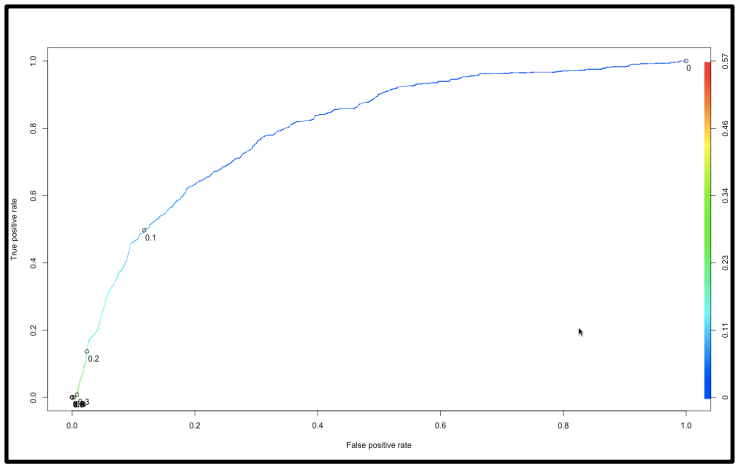
The McFadden values proves how model is well developed and the how values are predicted for the variables. McFadden with good fit is measured for the values above 0.5 and the less fitted model is measured of the 0.1 value. Since the model is measured and McFadden score for the fitted logistics model is 0.12 and it improves that model is Average Fitted for the variables. The Log Likelihood of the model is developed and measured for -10724.48 with the degrees of freedom is 20.

Logistics Regression Model Validation

The LR Model is predicted with both training dataset (Please refer Annexure I) and validation dataset. Even though, the model is built with validation developed with test (validation) dataset. The LR model is built with ROC curve, AUC values, KS Values and Gini values to check the validation performs well in the model.

The predicted model is developed with the deciles and rank the top 10 deciles for the model performed in the test dataset.

- The model is built again and AUC curve values are mentioned with the increased ROC plots for the claim status of the datasets in the variables for the increased motions of the variables in the functionality of the various models and the AUC is predicted for **0.7959825** which is measured for slight difference in the plots of the variables.
- The test variables are predicted for the models and the models are interpreted for the further development of the factors in the variables. The model prediction developed with test variables are measured for the final prediction of the models.



- The cut off of the performance predicted model is 0.1 as the precision values to the values for 0.1 and the recall is same as the variables are performing in the stage of the variables to the increased trend of the predicting values.

	FALSE	TRUE
0	12554	388
1	1711	387

- The logistics regression model developed for the positive and negative values in the datasets and predicted for the regression models developed for the 385 observations and rejected claims in the values with the Recall values is 0.4 and the precision values is taken as 0.18 for the model developed.

Accuracy	0.8628
95% CI	(0.8572, 0.8682)
No Information Rate	0.9485
P-Value [Acc > NIR]	1
Kappa	0.2128
Mcnemar's Test P-Value	<2e-16
Sensitivity	0.49677
Specificity	0.88265
Pos Pred Value	0.18698
Neg Pred Value	0.96996
Prevalence	0.05153
Detection Rate	0.0256
Detection Prevalence	0.1369
Balanced Accuracy	0.68971
'Positive' Class	1

The accuracy of the model is predicted for the validation datasets and the predicted score for the 86% and the sensitivity as well as specificity is vice versa maximized by the values in the model for the prediction of the variables. The variables are measured for the precisions and the recall is measured for the positive class of "1" in the values.

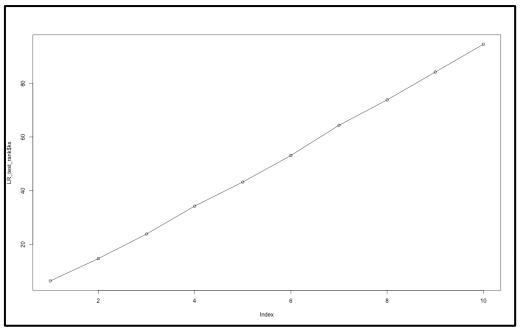
The balanced accuracy of the variance is measured for the various linked variables and the confusion matrix for the developed increased values for the test predicted values for the models developed in the logistics regressions.

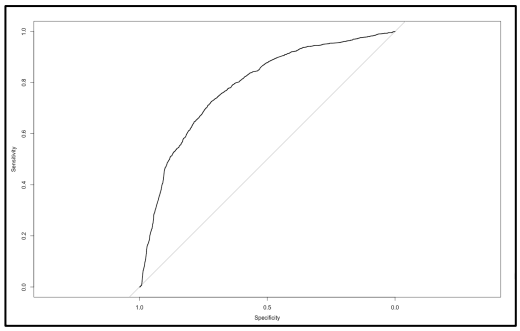
Model Performance Measure – LR Model

- The ranking variables are measured for the values for the increased and decreased deciles of the functions developed in the ranking order. The ranking order is measured for the ranking rates and the cumulative response for the acceptance claims and non-cumulative response for the rejected claims in the variables. The respondents variables are measured for the variable creation in the scaled percentage of the values.

	deciles	cnt	cnt_resp	cnt_non_resp	rrate	cum_resp	cum_non_resp	cum_rel_resp	cum_rel_non_resp	ks
1	10	1505	303	1202	20.13%	303	1202	2.12%	8.40%	6.31
2	9	1508	159	1349	10.54%	462	2551	3.24%	17.90%	14.64
3	8	1502	99	1403	6.59%	561	3954	3.93%	27.70%	23.79
4	7	1641	77	1564	4.69%	638	5518	4.47%	38.70%	34.21
5	6	1364	41	1323	3.01%	679	6841	4.76%	48.00%	43.2
6	5	1504	45	1459	2.99%	724	8300	5.08%	58.20%	53.1
7	4	1647	22	1625	1.34%	746	9925	5.23%	69.60%	64.35
8	3	1361	4	1357	0.29%	750	11282	5.26%	79.10%	73.83
9	2	1504	12	1492	0.80%	762	12774	5.34%	89.60%	84.21
10	1	1504	13	1491	0.86%	775	14265	5.43%	100.00%	94.57

- The ranking ordered tables is defined for the KS values with 94.57% for the measured developed rank tables for the r rate values and the variables are measured for the non-cumulative respondents for the deciles in descending orders. The values are bottom to top ordered deciles for the measured values. The accurate values for the cumulative respondents in the top deciles is measured for 775 observations and the model is well developed with the value.





KS values are measured for the variables to develop accurate variables for the deciles with values for the measured variables to the increased predicted values.

The ROC curves is measured for the values and the specificity and the sensitivity i.e. precision and recall values are measured for the ROC curves in the AUC values.

AUC

0.7981876

The AUC values are measured for the values in the y predicted values and 79% of the area under curve to the values predicted in the model.

KS

0.4614

The KS values are predicted for the measured values with 46% in the values and the measurements are increased from the training model and the values are increased for the various models and the variables are measured for the increased prediction in the values.

GINI

0.50548823

The GINI impurity is the measured values for the increased values and the decreased with GINI impurity of the values, 50% of the datasets is measured for the values in the dataset. The variables are measured for the training model is predicted for the 50% impurity in the model.

b) CART – Classification and Regression Trees

The model created for the CART is measured for the train dataset and the method is predicted for the class responded values for the model developed. The model is developed for the increased trends in the variables and measured the values for the decreased trees to the various values for the claim status for the training datasets.

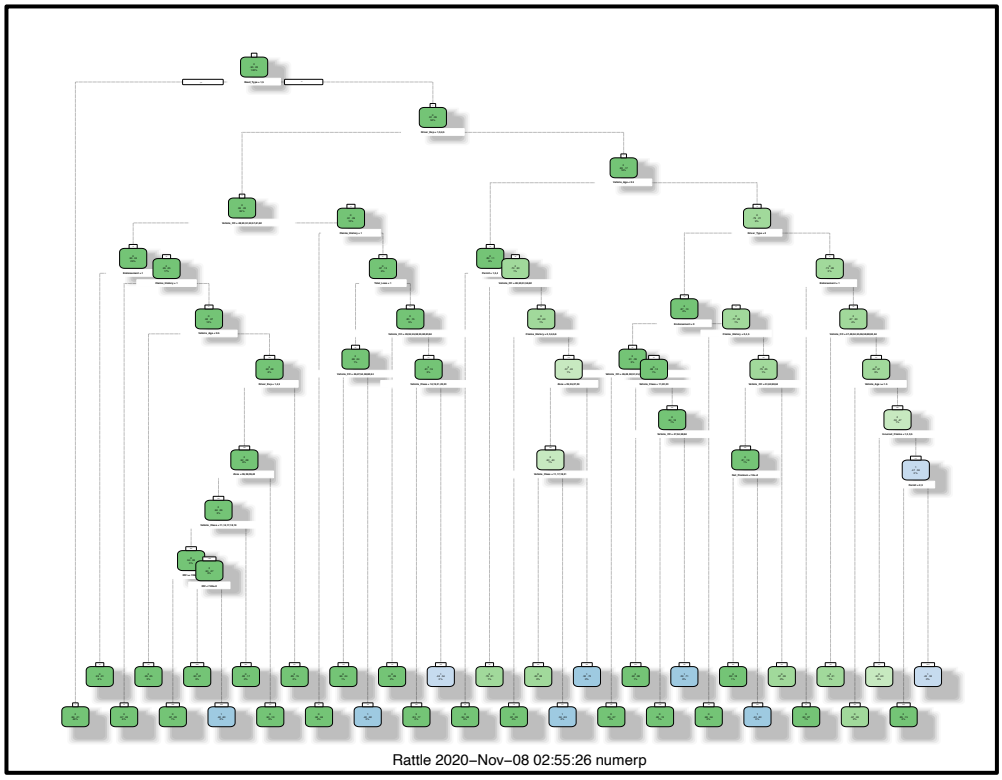
- The tree control is developed for the variables and the measured values to predicting the method class function values and the train control is the parameters developed for the minimum split of the variables is measured for 250 and the maximum split is measured for 5 splits in each nodes of the variables, the cross validation of the training datasets is measured for 5.

```
Variables actually used in tree construction:
 [1] Claims_History  Driver_Exp     Driver_Type     Endorsement     IDV          Incurred_Claims
 [7] Net_Premium     Permit          Road_Type       Total_Loss      Vehicle_Age  Vehicle_CC
[13] Vehicle_Class   Zone
```

- The tree used to developed for the important variables for the tree construction is measured for the 16 variables and the higher level factor variables are removed from the various models, the model are controlled in the training model and CART is model for the nodes developed.
- The cost proximity (CP) values are measured for the variables and the variables are increased for the error rates and relative error in the variables for the number of splits.

	CP	nsplit	rel error	xerror	xstd
1	0.00075245	0	1	1	0.017489
2	0.00064495	14	0.98517	1.0023	0.017507
3	0.00032248	19	0.98194	1.0064	0.017542
4	0.00024186	20	0.98162	1.0064	0.017542
5	0.00010749	24	0.98065	1.01	0.017571
6	0.00004031	27	0.98033	1.0087	0.01756
7	0	35	0.98001	1.0103	0.017574

- The table shows the variables are measured for the increased error and the standard values for the cost proximity values for the relative errors in the measured values.
- The cut off points measured for the values are increased for the variables with 0.00064495 to the relative error of the variables with the n split of 14.
- The tree is built with the important variables are measured for the importance of the tree construction of the variables developed in the values with 14 variables and the values are measured for the pruned trees classification of binary values.



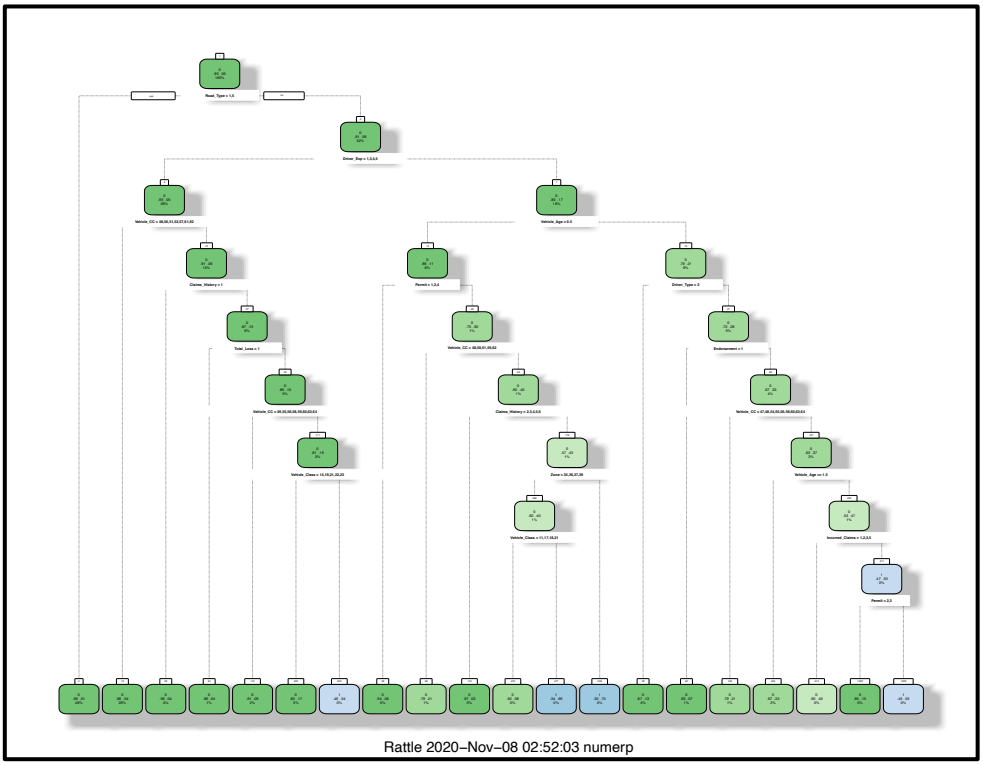
- The recall values are measured for the increased for the tree classified and the importance variables are measured for the values pruned for the variables in the differentiation of the tree developed with branches in the nodes developed for the training model.
- The pruned tree is developed with the most important variables and the significance variables are developed with the CP value of 0.00064495.

	CP	nsplit	rel error	xerror	xstd
1	0.00075245	0	1	1	0.017489
2	0.00064495	14	0.98517	1.0023	0.017507
3	0.00064495	19	0.98194	1.0064	0.017542

- The pruned tree is identified and the model is further developed with tree which identifies the various plots discussed as the important variables in developed in the model.

```
Variables actually used in tree construction:
 [1] Claims_History  Driver_Exp     Driver_Type     Endorsement     Incurred_Claims Permit
 [7] Road_Type       Total_Loss      Vehicle_Age     Vehicle_CC      Vehicle_Class   Zone
```

- Driver Experience with more than 15 years were accepted more claims in the developed with significant variables, claim year with certain years, the claims were accepted and shows the significant variables to predict the importance of the features with Vehicle CC, Vehicle Class, Permit and Zone is predicted as the importance in accepted claims.



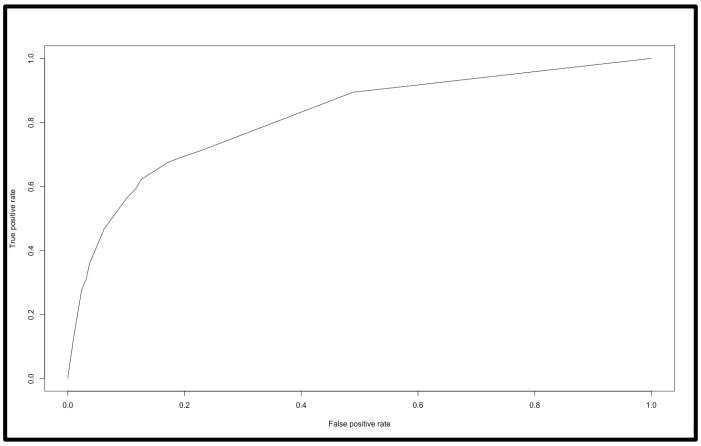
CART Model Validation

CART model is predicted with the validation dataset and the training dataset as well (please refer Annexure I). The dataset is imbalance in predicting the accuracy of fraud claims, CART is used to predict the higher balanced accuracy.

	deciles	cnt	cnt_resp	cnt_non_resp	rrate	cum_resp	cum_non_resp	cum_rel_resp	cum_rel_non_resp	rrate_perc	cum_rel_resp_perc	cum_rel_non_resp_perc	cum_cnt	cum_resp_rate
1.00	10	1868	436	1432	0.2334	436	1432	0.5626	0.1004	23.30%	56%	10.00%	1868	0.2334
2.00	9	1337	96	1241	0.0718	532	2673	0.6865	0.1874	7.20%	69%	18.70%	3205	0.166
3.00	8	4457	161	4296	0.0361	693	6969	0.8942	0.4885	3.60%	89%	48.80%	7662	0.0904
4.00	5	7378	82	7296	0.0111	775	14265	1	1	1.10%	100%	100.00%	15040	0.0515

The deciles are measured with the ranking orders and developed with vales for the cumulative respondents rate to the values and the least variables are measured for the increased values and the relative respondents for the values are measured for the 1 in the created pruned model and the values are increased for the values in the increased trends.

- The overall respondent rate is measured for the 5% in the CART model and the built validation is predicting the values of the measured values.
- The KS CART model is measured with values for the 50% of the variables are predicting with the values are increased in the values with decreased values for the validation model for the values in the test validation model.
- The y values is measured with AUC is predicted with 81% of the CART model is developed with the highest prediction of the variables in the datasets and the values are measured for the CART developed variables.



- The GINI impurity values are measured for the 58% and the impurity values are measured for the developed validation in the values.

	0	1
0	14209	733
1	56	42

- The test model is developed with the CART validations is measured for the 42 rejected claims and the values with 733 observations is developed for the highest frauds in the values with accuracy for the values.

Accuracy	: 0.9475
95% CI	: (0.9439, 0.951)
No Information Rate	: 0.9485
P-Value [Acc > NIR]	: 0.705
Kappa	: 0.0856
Mcnemar's Test P-Value	: <2e-16
Sensitivity	: 0.99607
Specificity	: 0.05419
Pos Pred Value	: 0.95094
Neg Pred Value	: 0.42857
Precision	: 0.95094
Recall	: 0.99607
F1	: 0.97299
Prevalence	: 0.94847
Detection Rate	: 0.94475
Detection Prevalence	: 0.99348
Balanced Accuracy	: 0.52513
'Positive' Class	: 0

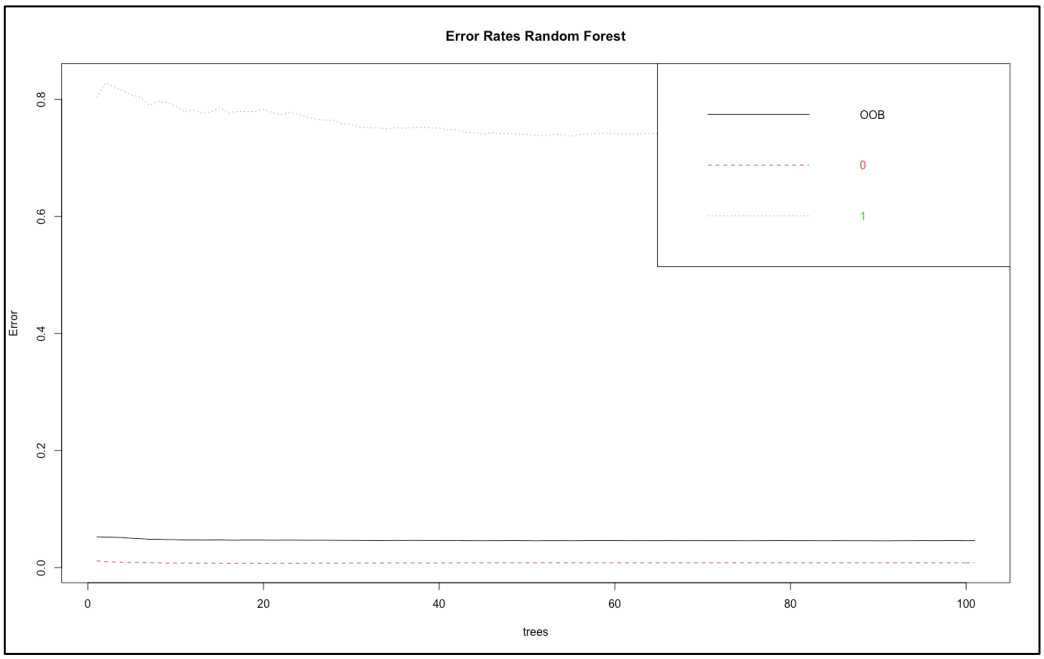
- The P-Value shows the value is nearer to 0.7 with correlated as the highest accuracy and the model developed is highly predicted with values for the 94% in the accuracy of developed variables in the CART model for the increased trend towards the values and the prediction of the values are measured for the sensitivity and specificity of the variables are predicted for the 99% and 5% respectively.

c) Random Forest

Random Forest model is measured for the variables with character variables less than 50 levels and the Random Forest model is predicted with general values except dates variables for the importance values in the node size computed as 100 with basic imputation.

Confusion matrix:			
	0	1	class.error
0	56599	459	0.008044446
1	2299	802	0.74137375

The confusion matrix developed with the 822 is predicted for the rejected status and 31 variables is false statement in the variables of the class error to the values with 0.73 for the variables in the values.



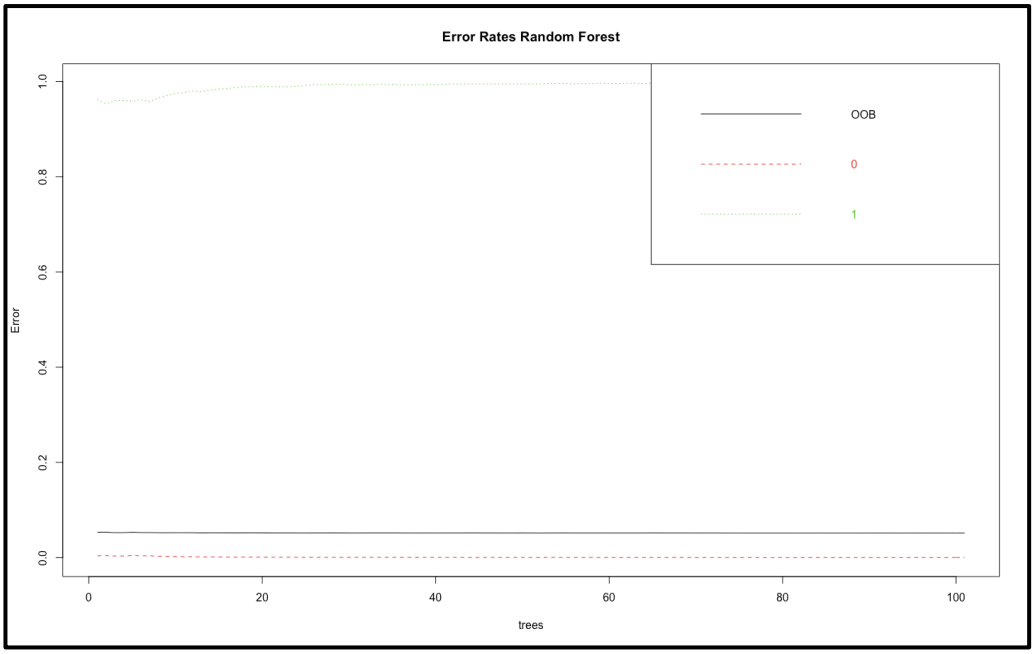
The OOB shows error rate is classified with the values in the predicting nature with increased to decreased values of the binary classification rate. The plot explains that rejected status is constant from the trees is developed in the predicting feature of the values in the variables and the values are measured for the increased size of the OOB in the initial trees.

- The error rates are measured for the increased variables and the values are measured for the decreasing variables and the maximum variables and trees are predicted for the variables and the error rates are measured for the values for the random forest value developed with majority of the OOB values are increased for the predicted values in the variables.

The Random Forest is again built without claim amount and the previous taken variables in the model. Since, the claim amount showing higher decrease in Gini but not helping in further developing the model. The claim amount shows insignificance in initial understanding of data, hence it is removed from the model.

Confusion matrix:			
	0	1	class.error
0	57046	12	0.000210312
1	3091	10	0.996775234

The model shows 11 observations are predicted correctly and the 22 observations were predicted wrongly in the fraud claims. Hence model is predicted with class error of 99% which is allows to predict the model.



The model is further tuned into find the exact mtry and node size to improvise the model further.

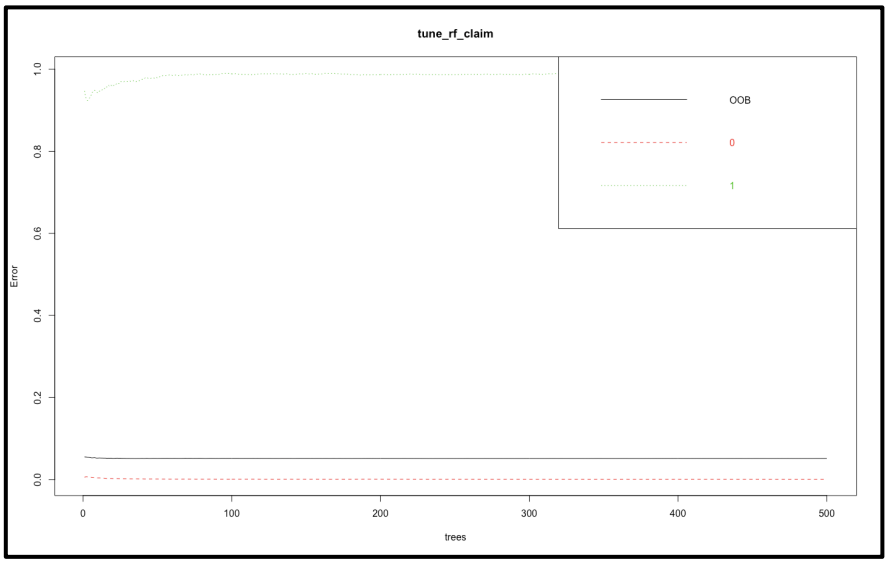
Tuned Random Forest

The importance of the variables are measured for the increased values and all variables are measured for the increased developed in the basic parameters of the controlled variables in the values with increased for the values in the random forest model is built with values and the controlled values are measured from the tuned Random Forest model.

- The increased values are measured for the increased prediction of the values and the tuning parameters are considered for the values and the prediction of the values are increased for the decreased accuracy and mean decreased GINI of the tables in the predicted values.

	0	1	MeanDecreaseAccuracy	MeanDecreaseGini
Road_Type	10.47	51.12	18.33	183.47
Driver_Exp	4.18	57.12	24.44	161.7
Vehicle_Age	0.25	40.47	22.51	132.99
IDV	24.29	-13.38	22.66	97.88
Endorsement	16.32	11.05	24.14	88.51
Vehicle_CC	23.1	-4.98	24.25	84.52
Net_Premium	19.52	-3.32	21.77	84.04
Driver_Type	0.55	52.86	8.93	75.7
Incurred_Claims	8.31	10.26	12.91	65.95
Permit	10.99	21.2	13.96	64.51
Claims_History	10.59	7	12.69	55.8
Zone	20.38	1.51	22.11	49.18
Vehicle_Class	17.13	2.63	19.09	47.7
Antitheft	9.66	2.09	11.4	42.37
Driver_Qualification	2.54	7.21	4.13	33.9
Total_Loss	6	17.69	8.75	33.68
Discount_NCB	1.22	13.76	7.34	25.49
Nature_of_Goods	1.92	8.37	4.21	10.06
Policy	13	8.34	14.09	4.24
Statutory_Cover	2.43	-2.83	1.07	0.68

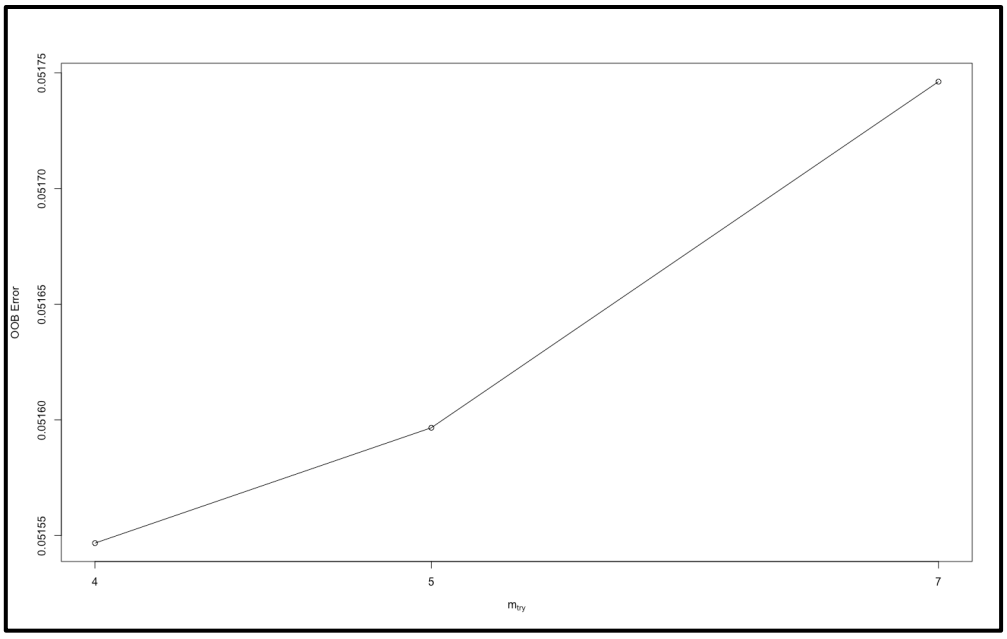
The importance of variables are predicted with the tuned random forest and the variables are predicted with higher decrease in Gini is Road Type, Driver Experience, Vehicle Age, Driver Type, Permit are higher importance and first preference will get more on the attention. Claims History, Discount, Driver Qualification, Total Loss, Nature of Goods and Policy are covered with the secondary importance variables where this variables are contribute in developing the trees but not at the significant level of variables.



The tuned random forest model is predicted with the error rate with increased to the various features predictions. The model is developed with mtry at the highest value is about 7 and the plot is defined with decreased in Gini Impurity of the dataset.

```
mtry = 5  OOB error = 5.17%
Searching left ...
mtry = 4                OOB error = 5.17%
0.0009643202 1
Searching right ...
mtry = 7                OOB error = 5.15%
0.003535841 1
```

The plot shows the values are same in the trees of 4 and 5 and the value is decreased with error rate of 5.15% for the tree 7.



The dimensionality reduction of date variables and claim amount shows the tree developed in the random forest tuned model is predicted with the highest mtry of 7. The data is predicted with the train and test validation for the best fit in predicting the claim status.

Random Forest Model Validation

Random Forest model is further validated with the training (Please refer Annexure I) and validation dataset.

The model is developed with prediction scores and predicted values. The tuned random forest model is further predicted with scores of lift, Gini, KS Value and AUC plots.

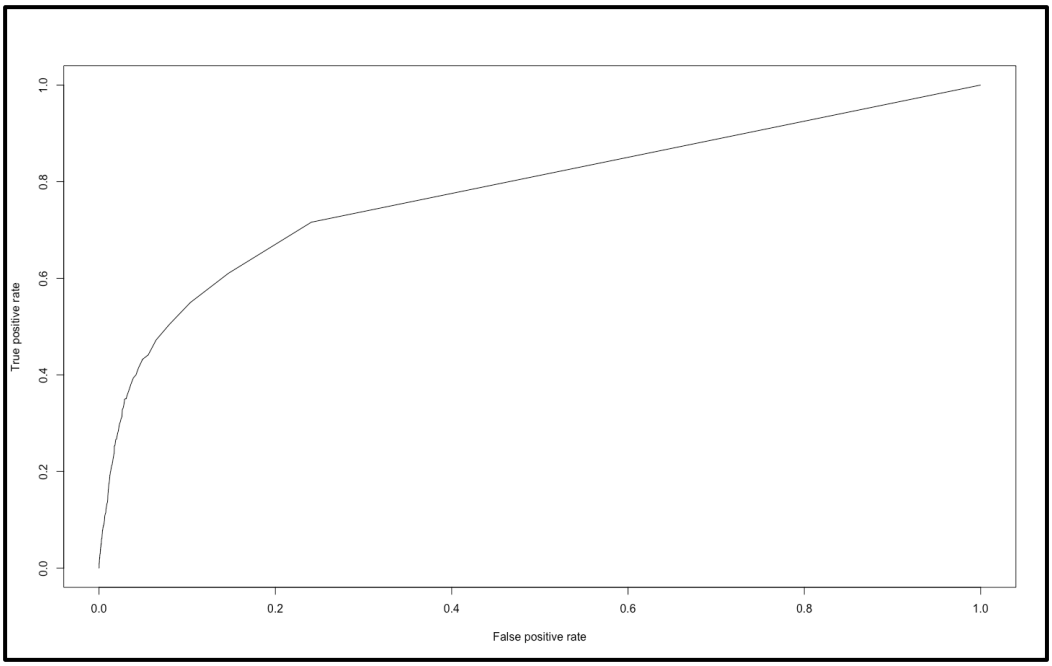
- The overall response is predicted for the 5% in the values in the variables are contacted for the values are measures with predicting values increased with the prediction of the tuned random forest.

```
rank_rf1$lift
[1] 4.96 2.70 1.00
```

- The lift of the rank is developed with 4.96 for the first rank predicted and achieved the 1 as highest rank variable is measured with the various values for the increased values.

	deciles	cnt	cnt_resp	cnt_non_resp	rrate	cum_resp	cum_non_resp	cum_rel_resp	cum_rel_non_resp	rrate_perc	cum_rel_resp_perc	cum_rel_non_resp_perc	cum_cnt	cum_resp_rate
1:	10	1530	391	1139	0.2556	391	1139	0.5045	0.0798	25.60%	50%	8%	1530	0.2556
2:	9	2461	164	2297	0.0666	555	3436	0.7161	0.2409	6.70%	72%	24%	3991	0.1391
3:	8	11049	220	10829	0.0199	775	14265	1	1	2.00%	100%	100%	15040	0.0515

- The plots shows the AUC values is 0.7793415 below the area and the plot defines in the predicting the most important significant variables.



- KS value is predicted with 70% for the increased stability in the performance of the variables.
- GINI impurity is measured for the 94% of the values is measured for the increased values and the values are predicted for the increased values in the impurity is decreased with the values for the increased values.

	0	1
0	14262	772
1	3	3

- The accuracy of the model is built with the variables 94% of the accuracy in the model prediction of the train datasets in the values are gained with the p-value is less variable for the increased for the values in the values for the developed variables

Accuracy	: 0.9485
95% CI	: (0.9448, 0.952)
No Information Rate	: 0.9485
P-Value [Acc > NIR]	: 0.5096
Kappa	: 0.0069
Mcnemar's Test P-Value	: <2e-16
Sensitivity	: 0.999790
Specificity	: 0.003871
Pos Pred Value	: 0.948650
Neg Pred Value	: 0.500000
Precision	: 0.948650
Recall	: 0.999790
F1	: 0.973549
Prevalence	: 0.948471
Detection Rate	: 0.948271
Detection Prevalence	: 0.999601
Balanced Accuracy	: 0.501830
'Positive' Class	: 0

5. Model Validation

Ensemble Methods – XG Boost

The Extreme Gradient Boosting is developed on the train and test dataset with split of 80% and the featured train data is selected with numeric variables and the labelled train is taken as factor variables which is binary classification and featured test variables is selected with numeric variables. The variables with features and the variables developed with various models are developed for the matrix formats for the model build of extreme gradient boosting.

- The Extreme Gradient Boosting is developed for the initial binary logistics model developed for the feature of the train dataset and the tree is developed stopping at 10 and the measured values of the minimum weight is measured for 3 and the eta is controlled for 0.001 as the measurement of the variables in the featured variables.

```
# of features: 4
niter: 11
best_iteration : 1
best_ntreelimit : 1
best_score : 0.051547
nfeatures : 4
evaluation_log:
   iter train_error
      1    0.051547
      2    0.051547
---
     10    0.051547
     11    0.051547
```

- The model is built with the featured train variables and the values are increased for the iteration error rate of 0.5% of the train error rate in the values and the values are measured for the increased values trends into the values for the increased values. Errors are measured for the basic build model and the values are predicted in the tree limit of 1.
- Since the prediction is measured with values of 775 as fraud claims and the model is further developed with the tuned extreme gradient boosting.

[,1] [,2] [,3] [,4] [,5] [,6] [,7]

[1,] 0 0 0 0 0 9 20

- The tuned extreme gradient boosting shows the values is predicted for the points at 6 and 7 and it shows the higher prediction in 7 as 20 predicted gradient in boosting the datasets.

		Reference	
Prediction		0	1
	0	14245	755
	1	20	20

- The prediction and reference predicted values are measured for the validation dataset and the methods is highly predicted for the values in the increased and false positive rate and 20 as the true positive rate. The variables are measured for the increased in the values for the 755 observations.

Accuracy	: 0.9485
95% CI	: (0.9448, 0.952)
No Information Rate	: 0.9485
P-Value [Acc > NIR]	: 0.5096
Kappa	: 0.0442
Mcnemar's Test P-Value	: <2e-16
Sensitivity	: 0.02581
Specificity	: 0.99860
Pos Pred Value	: 0.50000
Neg Pred Value	: 0.94967
Prevalence	: 0.05153
Detection Rate	: 0.00133
Detection Prevalence	: 0.00266
Balanced Accuracy	: 0.51220
'Positive' Class	: 1

The predicted accuracy of the claim status in the variables are predicted for 94% and the recall values for the predicted accuracy is predicted for the variables in 2% and the specificity of 99% in the values are predicted for the variables are measured for increased variables.

SMOTE

SMOTE is developed with the train datasets and test datasets with the values in the prediction control over 400 and prediction control under 100 is taken to predict the variables for the featured variables with different levels.

- The unbalanced data is expected to go down for the balanced data in the SMOTE variables and the variables are measured for the increased values in each vectors.
- The binary classification is achieved with the values are controlled for the claim status rejected in the values are increased for the values in each predicting values of the increased vectors in claiming the balanced data.

0	1
12404	15505

- The balanced data proven that 15505 is predicted as rejected in claiming the insurance and the 12404 is decreased as acceptance rate of the variables in the values.
- The matrix model is developed for the prediction of the SMOTE analysis, hence the factor or binary variables are changed into numeric variables and the characteristics variables increased for the variables prediction of the SMOTE.
- The SMOTE model is developed with the xgboost developed for the various values and the values are increased for the logistics developed for the increased tree and stopping the values for the values are predicted in the values are measuring the rounds the trees for the gradient prediction in the values for the prediction in test smote dataset.

```
niter: 50
best_iteration : 50
best_ntreelimit : 50
best_score : 0.07245
nfeatures : 9
evaluation_log:
   iter train_error
      1    0.134867
      2    0.118815
---
     49    0.072987
     50    0.072450
```

- The Iteration error rate was measured for the increased prediction of the dataset is provided the variables to pointing the increased error rate to the values are prediction of the variables in the considered values of the prediction nature of the variables in the SMOTE.

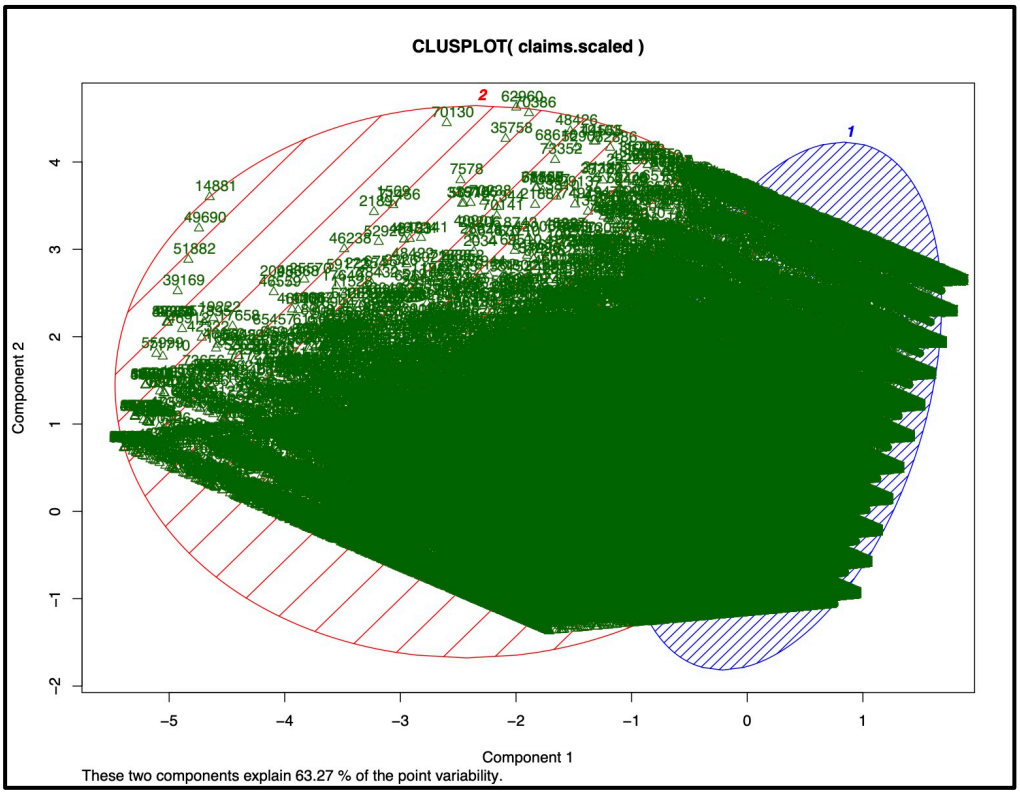
	Reference	
Prediction	0	1
0	12776	336
1	1489	439

- The predicted values for the SMOTE is discussed with the variables are predicted for the 420 as true to rejected and the false positive rate is increased with 355 and the valuation of the dataset is predicted with accuracy in the SMOTE predictions.
- The accuracy of the model is predicted for the 87% and the sensitivity of the variables are measured for the SMOTE variables for the sensitivity and specificity of the variables are predicted in the among us nature of the variables and the positive class as 1 in the prediction of the variables to increase the model accuracy.

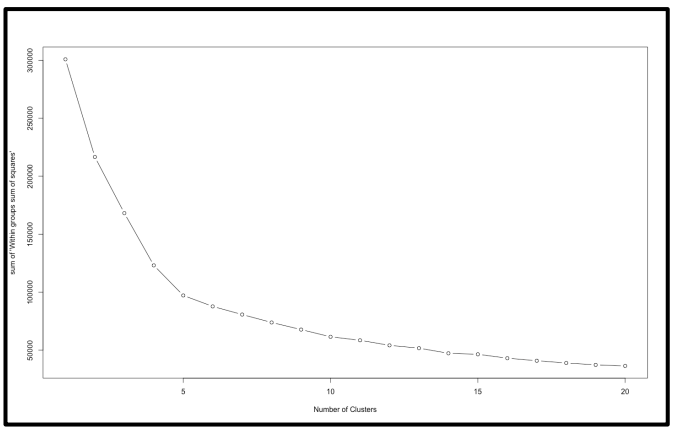
Accuracy	: 0.8787
95% CI	: (0.8733, 0.8838)
No Information Rate	: 0.9485
P-Value [Acc > NIR]	: 1
Kappa	: 0.2713
Mcnemar's Test P-Value	: <2e-16
Sensitivity	: 0.56645
Specificity	: 0.89562
Pos Pred Value	: 0.22770
Neg Pred Value	: 0.97437
Prevalence	: 0.05153
Detection Rate	: 0.02919
Detection Prevalence	: 0.12819
Balanced Accuracy	: 0.73104
'Positive' Class	: 1

K-Means Clustering

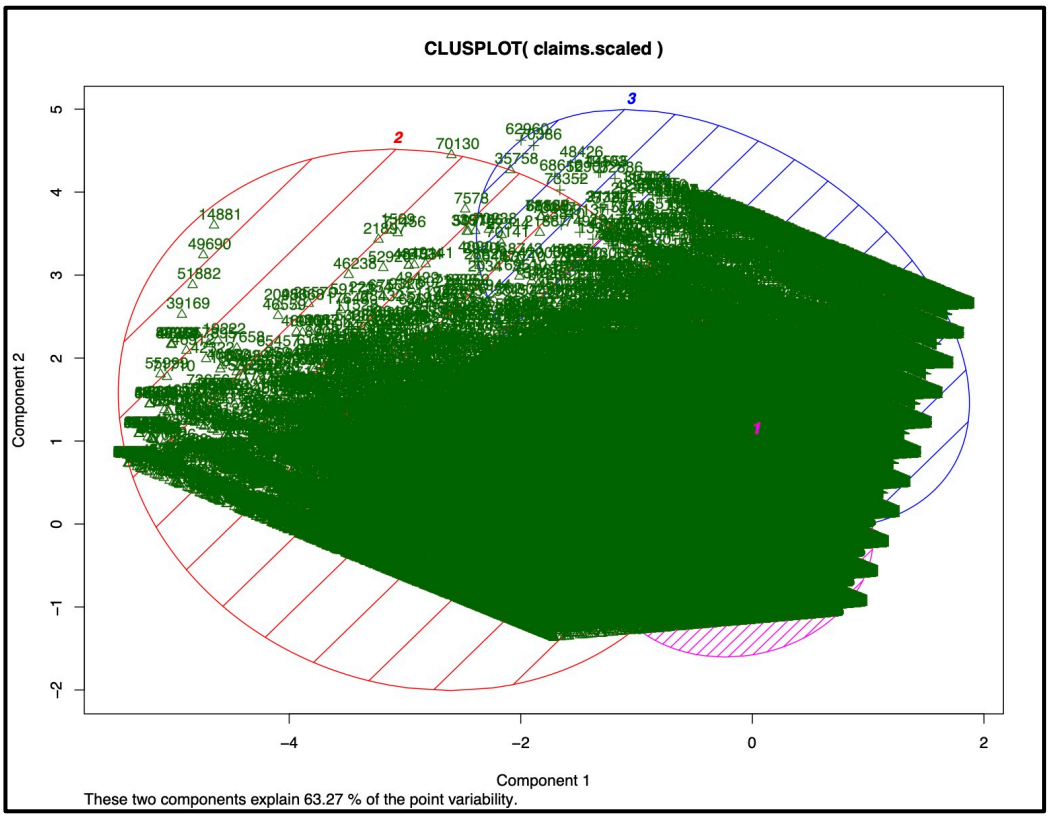
- K-Means Clustering is used to predict the nature of claims to grouped and how the clusters makes differences in predicting the Fraud claims were accepted, rejected and whether accepted or rejected.
- The cluster is initially taken for the 2 centres to define how model is predicting the nature of the variables and the k-means structure of the variables are measured for the variables for the increased in the variable values and increased with nstart of 5.



- The WSS plot is carried out for the center and the K values is prediction of the variables with the clusters and the elbow plot should carried to exact point out in the variables and the variables are prediction for the increased and developed prediction of the variables.



The variables are measured for the 3 clusters and the variables is increased for the group of clusters and the plot defines the majority of green cluster is measuring the range of the variables and the variables are increased for the numeric predictions and the variables with the decreased group of clusters are defined as the variables for the group 1 is behind the clusters and the group participates predicted as medial orders of claiming and not claiming the predictions of the variables.



The group 2 shows the major grouping of values in the Vehicle age, group 3 shows the major grouping is done with increased values of 9481.474. This implies and indicates the policies and the values are measured for the increased prediction of the values in the variables and the groups are managed with 3 clusters are predicted for the better cluster.

	Group.1	Vehicle_Age	IDV	Claim_Amount	Net_Premium
1	1	1.446391	758296.5	65161.54	9481.474
2	2	0.973054	357686.8	20037.22	5467.46
3	3	6.083299	326667.5	25157.5	6328.097

Model Comparison Measures

The Model comparison is measured for the logistics regression, cart model, random forest, extreme gradient boosting and smote model is predicted for the confusion matrix model development. The Table Model defines the Sensitivity, Specificity, Precision, Recall and F1, AUC, KS, Gini is taken for the accuracy of the variables in the various confusion matrix build.

	Sensitivity	Specificity	Precision	Recall	F1	KS	AUC	GINI
LOGISTICS REGRESSION	0.49935	0.88006	0.18446	0.49935	0.26940	0.46140	0.79819	0.50549
CART	0.99607	0.05419	0.95094	0.99607	0.97299	0.50382	0.81507	0.58521
RANDOM FOREST	0.99979	0.00387	0.94865	0.99979	0.97355	0.48259	0.78558	0.93903

The Model Performance Measures are observed for the KS Value, AUC and Gini. Since, the dataset is highly imbalance and provides the overfitting model in predicting the variables in each final developed model.

- On performing the Model Validation table, it shows that CART and Random Forest proves the highest accuracy with predicting fraud claims that were accepted. AUC and Gini values are making the significance in providing the best model.
- Since Gini 93% of RF model is without impurity, made the best performance model.
- AUC with 81% shows that CART model is able to fit around the values and to predict the highest model accuracy with 95%. The CART model is well exhibited in KS and AUC values but it lesser with the Gini Impurity of 58%.
- Random Forest model fitted the AUC values with the 78% which good to predict the most fitted values and 49% of the values are exhibits the highest relations with the variables. The Gini Impurity shows that 94% of the dataset is cleaned in the dataset and the model is well performed in the validation dataset. Random Forest is predicted for the best model out of performed model.

	Sensitivity	Specificity	Precision	Recall	F1
EXTREME GRADIENT BOOSTING	0.02581	0.99860	0.50000	0.02581	0.04908
SMOTE	0.56000	0.89548	0.22545	0.56000	0.32148

- Imbalanced dataset is defined using ensemble methods of Extreme Gradient Boosting and SMOTE analysis. Where both the models are scaled and balanced the dataset to provide the best accuracy of the balanced dataset. Since the Analysis is taken over for the Balanced Dataset, Sensitivity, Specificity, Precision, Recall and F1 Accuracy is taken.
- The SMOTE analysis performs that recall value of 56% is best to the accuracy of model and balanced SMOTE provides the best model.

6. Final Interpretation and Recommendations

Insights from EDA

- The Insurance claims associated for the highest values for the accepted claims and the lowest values for the rejected claims in the various variables. The variables are coordinates with the values with all benefits and the numeric variables covered up.
- The claims are measured for the accepted or rejected status in overall numeric variables and the numeric variables are less in identifying the values of the rejected status. Since the values are mentioned in the codes almost all variables are carried by the character variables in the most of the basic variables in code of policy, statutory, claim, discount and pre-defined vehicle details.
- Claim status will provide the exact variables in association of the variables associating in the each variables for the controlled values and the values are increased in accepted status which implies the good to customer and bad in company profiting from the claiming properties repeatedly from the continuous claims in the values.
- The variables are associated with variation in higher correlation are observed for the IDV and Net Premium shows the values are measured for the positive impacts in the values and the measured values for the increased impact with negative values is rejected for the analysis is with Age of vehicle and IDV values.
- The Exploratory analysis clearly shows that more of one even variables are taken into the account of the policy code is package policy, drivers with private car is increased values, Zone B have higher accepted status, Year 2011 has been the highest claim in the insurance policy, single time claimers are highest in the vehicle range of 1000CC to 1500CC with accident nature of loss as the increment of the accepted status in the datasets.
- The claim status and the other monetary benefits developed from the various policies are measured with most the accepted claimers in the companies.
- The claims are based on the values with higher correlation in numeric variables in the values and prompt in increased values for the insured person in single value of each variables, hence the company can focus on the values and increased the values for the general claim policy and to increase the package policy as diversified customer to increase the insurer benefits in the variables.

Insights and Recommendations from the Model

- Random Forest Model is best in predicting the dataset with less impurity and performs well in KS Value, AUC Values and Gini. The Gini with 93% is predicted for the less impurity data in the model is predicted. Random Forest Model is predicted for the tuned parameters and prediction is made for the importance of top 10 variables.
- The highest Mean Decrease in Gini is predicted with Road Type, Driver Experience, Vehicle Age, Vehicle CC, Driver Type, Zone, Permit and Claims History, Nature of Goods, Antitheft is predicted for the highest importance variables. The variables are taken in to account for the best accuracy in providing the fitted model.
- The important variables are measured along with Exploratory Data Analysis to show that Road Type of Urban Roads, Driver Experience with more than 15 years, Vehicle Age more than 9 years, Vehicle with CC of 1000 CC and 1500 CC, Vehicles driven by other than driver, which have the permit of Zone places Zone B and claims history with one time claimers are measured with the vehicle purpose of carrying other than hazardous materials are claimed more and have to recommended in releasing the claim or not.
- The claims frauds and the variables are predicted with increasing the less leakage of frauds and filter the fraud claims to measure the highest claims with frauds in the model to lower the risk of claim frauds in the insurance companies.
- Interpretation of the models are discussed with variables to predict the variations for claim amount is removed from further model as the variables is measured for higher significance but the claim amount with Rs.0 as rejected and Rs.2M is accepted where the variable is contributes less in model developing. The dates variables, claim amount variable is removed from the model and makes the model as best fitted performance.
- The insurance company have to research and suggest on claiming the claim amount is predicted for claim status with accepted or rejected status. The claim amount is closely look out to provide the further more improvement of model.
- Insurance variables with less importance of Net Premium, IDV, Endorsement, Policy, Statutory Cover, Total Loss and Driver Qualification are less contributors and will improves in EDA part alone.
- Importance variables mentioned are recommended to carried out in regular research of claim is to release or not and less importance variables to improvise further in Data.

7. Annexure I

Data Cleaning and Pre-Processing

The variables are classified by the characteristics on the date, factor, character and numeric variables from the 31 variables.

```
> names(num_var)
[1] "Vehicle_Age"  "IDV"          "Claim_Amount" "Net_Premium"
```

The numeric variables are observed for the age, IDV, claim amount and net premium.

```
> names(cat_var)
[1] "Location_RTA"          "Policy"           "Vehicle_Class"    "Zone"
[5] "Vehicle_CC"            "Vehicle_Colour"   "Permit"           "Nature_of_Goods"
[9] "Road_Type"             "Driver_Type"      "Driver_Exp"       "Claims_History"
[13] "Driver_Qualification" "Incurred_Claims"  "Accident_Place"   "Nature_of_Loss"
```

The category variables are measured for the location, vehicle class, vehicle colour are observed for the variables and the measured variables are character variables for the numeric values in the datasets.

```
> names(factor_var)
[1] "Endorsement"          "Statutory_Cover"    "Total_Loss"          "Claim_Status"       "Antitheft"
[6] "Discount_NCB"
```

The factor variables are observed with “0” and “1” level with accepted and rejected. The accepted status are measured with higher rate in each variables and the covered benefits are measured for the each variables.

```
> names(date_var)
[1] "Policy_Year"          "Claim_Year"         "Accident_Date"       "Claim_Intimation_Date"
[5] "Disbursement_Date"
```

The date variables are measured for total periods in the datasets.

Variable Transformation

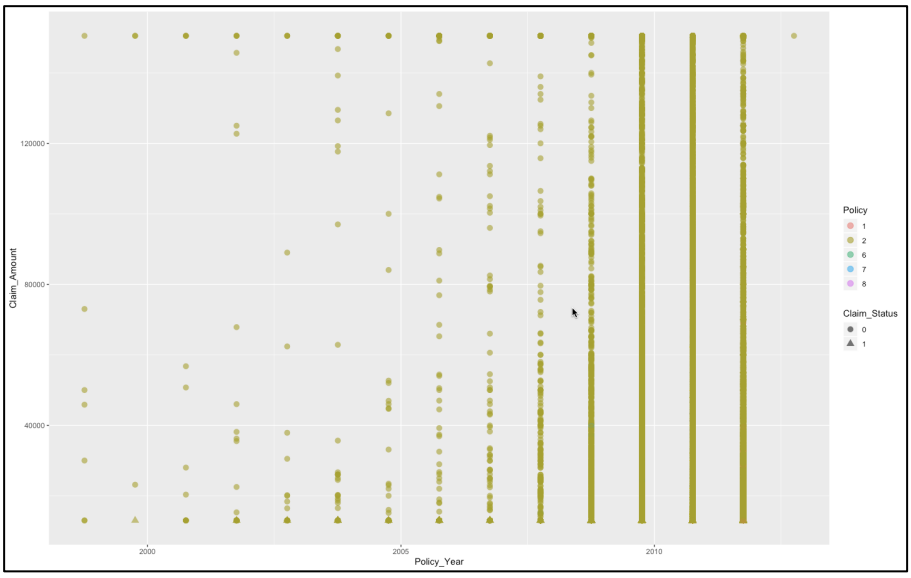
```
> summary(claim$DRV_CLAIM_STATUS)
CLOSED          REJECTED
         71324              3876
```

The claim status is observed for the target variables with accepted claims and rejected claims from the datasets. Which shows that 0.5% of the datasets is rejected claims in the datasets. The factors are converted for the variables with 0 and 1 as conversion in the datasets. The variables are transformed with the factor variables in target variable.

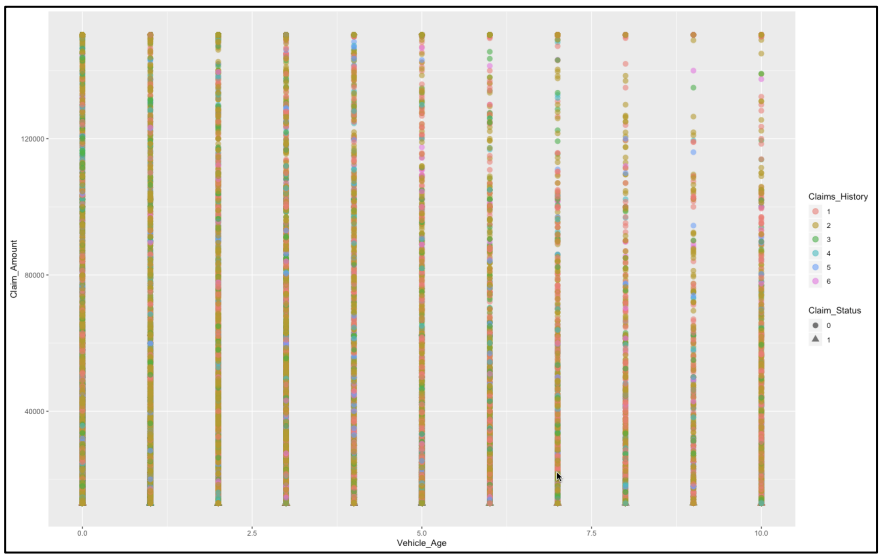
Date variables are transformed from the Financial year to single year and the values of the dates are measured for the present date with same financial year, since the dates are constructed only for the year values.

Variable transformation is necessary in each variables for the factor variables for the factored level and the variations are taken in further analysis of the values and the measured values are taken datasets.

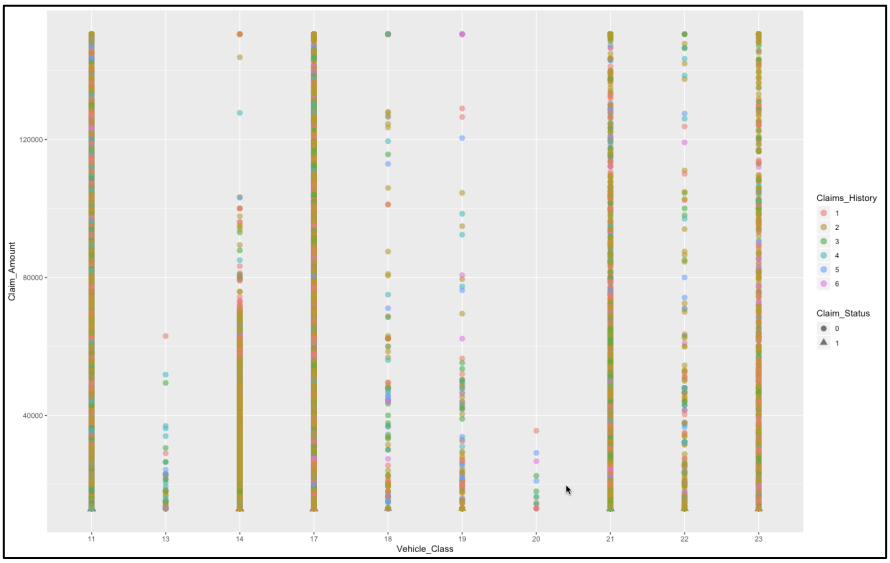
Bi-Variate Analysis



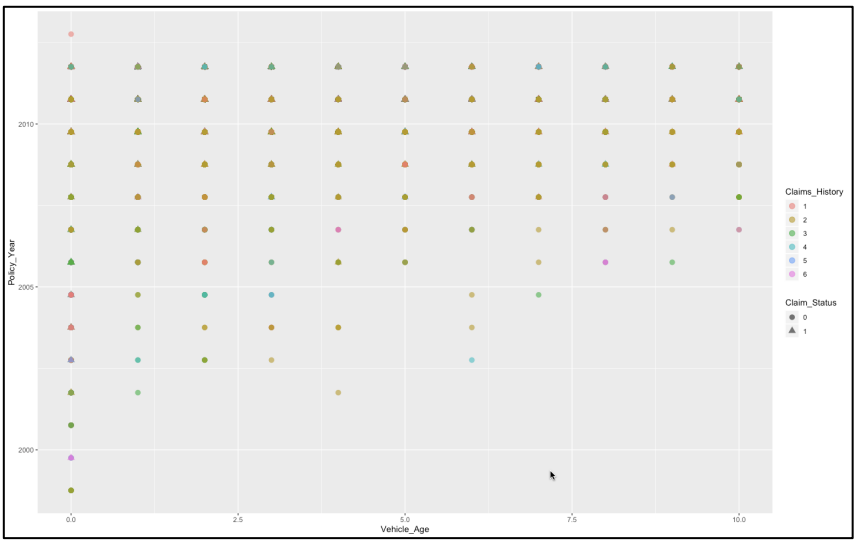
The policy year is compared value for the claim amount measured with accepted and rejected status for the values in the policy code. The more policy is measured for the values are controlled at the increased year at 2011 and the insured values are making the best lower claim towards year from 1998 to 2004 of the variables. The values which are making the increased policy claim are measured with package policies.



The measured values are increased for the claiming data in having the less claims history from the claim age of the vehicle, the claim status rejected and accepted are observed with maximum accepted status and increased the values for the various increased factors in the one-time claims of the insurer policy. The policies with less claim history are measured for the increased claim history in the values.



The vehicle class with most variables are measured for the 1000cc to 1500cc vehicles and the claim status is measured for the accepted and rejected status in the value. The vehicle with two wheelers are measured for less claims towards the policies and the values are measured for the increased accepted status in the heavy vehicles in the variables for the increased values in the claim status of the insurance company.



The vehicle age with the claimed status is marked with the accepted status for the maximum age in the vehicle age and the variables are measured for the initial age of the insurance claiming data to provide the highest claims of the basic variables to the scattered points in the basic value.

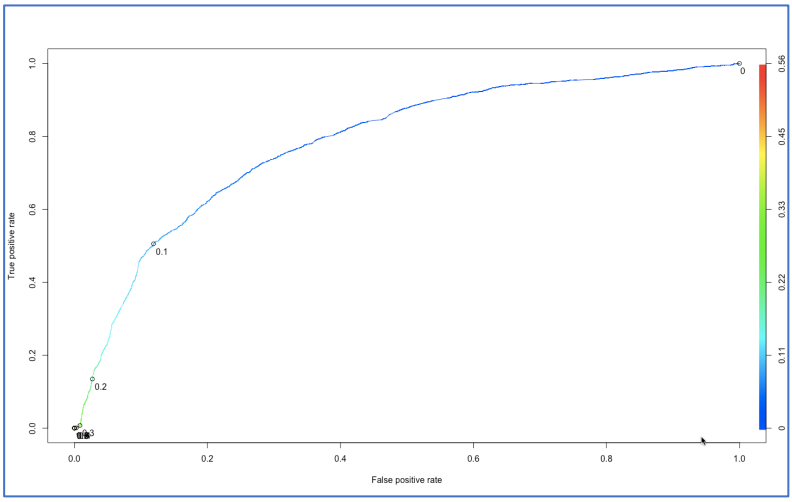
Model Building and Validation (Using Train Dataset)

Logistics Regression Model – Train

The model is predicted with the train dataset to achieve the higher inflation rate of the variables and the response is selected as the model is mainly to predict the rejected claims of the dataset in the variables.

The prediction is calculated with the roc curve and the variables are increased with prediction of the AUC as the higher predicted and how the values are distinguished in the step of the below **0.7838533** as the values covered under the ROC Prediction

Performance of the prediction is measured with the true positive rate and false positive rate for the better understanding of the values measured under area curve.



The curve shows the False Positive Rate is increased with increased True Positive rate of the variables and the area plotted towards to the values are measured for the increased trends in the variables for the higher prediction of the significant values in the various measured values. The cut off measured for the variables is 0.1 which predicted the values with precision of 0.1 and the recall value of the variables are measured about 0.5

	FALSE	TRUE
0	50274	6784
1	1534	1567

The table is measure for the recall of 5% and the precision of the training dataset in predicting the rejected claims is 1.8% which is measured for the better accuracy of the variables.

The predicted values and the actual claim status of the variables and values are measured for the confusion matrix developed variables.

```
Accuracy : 0.8617
95% CI                  : (0.8589, 0.8645)
No Information Rate     : 0.9485
P-Value [Acc > NIR]     : 1

Kappa                   : 0.2146

Mcnemar's Test P-Value : <2e-16

Sensitivity             : 0.50532
Specificity             : 0.88110
Pos Pred Value          : 0.18764
Neg Pred Value          : 0.97039
Prevalence              : 0.05155
Detection Rate          : 0.02605
Detection Prevalence    : 0.13882
Balanced Accuracy       : 0.69321

'Positive' Class        : 1
```

The accuracy of the train predicted variables are measured for 86% in the datasets and the values are measured for the increased variables for the sensitivity of the values and increase the specificity of the variables associated to the variables for the increased with the positive class "1" for the detection rate.

Deciles Creation

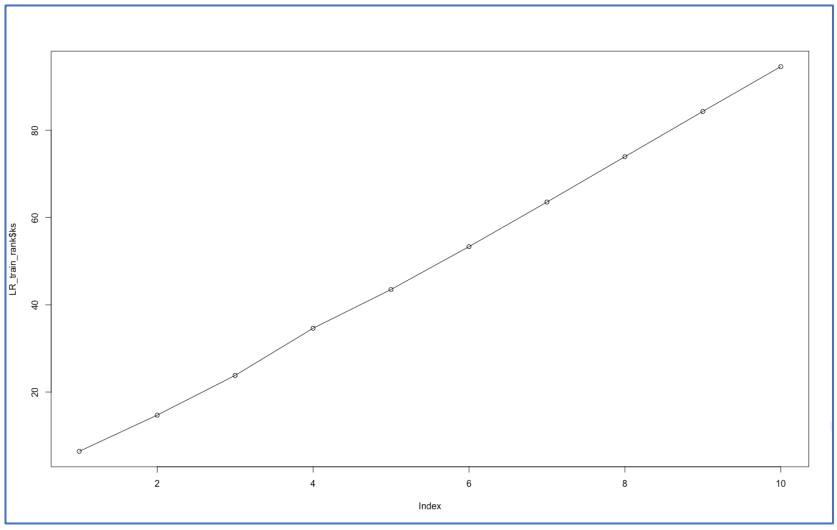
The deciles function is created for the variables for the decreasing order and the ranking orders are measured for the variables using deciles as a function of the variables for the descending order for rank orders.

The train model predicted is implied for the decile creation of the response in the variables for the model building in the rank orders and the predicted values is measured for the Response variables of the increased prediction with accuracy of 88%

Rank orders are created for the train model and the models are created for the measured values in the train prediction.

	deciles	cnt	cnt_resp	cnt_non_resp	rrate	cum_resp	cum_non_resp	cum_rel_resp	cum_rel_non_resp	ks
1:00	10	6022	1182	4840	19.63%	1182	4840	2.07%	8.50%	6.41
2:00	9	6011	636	5375	10.58%	1818	10215	3.19%	17.90%	14.71
3:00	8	6015	414	5601	6.88%	2232	15816	3.91%	27.70%	23.81
4:00	7	6683	257	6426	3.85%	2489	22242	4.36%	39.00%	34.62
5:00	6	5465	199	5266	3.64%	2688	27508	4.71%	48.20%	43.5
6:00	5	5899	146	5753	2.47%	2834	33261	4.97%	58.30%	53.32
7:00	4	6016	97	5919	1.61%	2931	39180	5.14%	68.70%	63.53
8:00	3	6016	42	5974	0.70%	2973	45154	5.21%	79.10%	73.93
9:00	2	6060	66	5994	1.09%	3039	51148	5.33%	89.60%	84.31
10:00	1	5972	62	5910	1.04%	3101	57058	5.43%	100.00%	94.57

The rank orders and the deciles are measured for the increased rate with cumulative response and non-cumulative response of the variables in the measured values with KS values and the values are increased for the r_rates and the variables are measured for the increased KS values is 94.57.



The plot increasing for the values and variables are measured for the increased trends and the KS values is plotted for the increased functionality of the datasets.

CART Model – Train

The train prediction of the deciles ranking is measured for all variables and the train model for the score developed in the predicted values and the prediction of the variables are controlled for the values is developed with probability values and the prediction values for the class variables.

The rank orders are developed for the claims in CART model developed in the training model for the cumulative response and non-cumulative response for the rank order developed for the decreasing deciles, the rank of the variables are measured for the increased trends in the deciles.

	deciles	cnt	cnt_resp	cnt_non_resp	rrate	cum_resp	cum_non_resp	cum_rel_resp	cum_rel_non_resp	rrate_perc	cum_rel_resp_perc	cum_rel_non_resp_perc	cum_cnt	cum_resp_rate
1:00	10	17722	3081	14641	0.1739	3081	14641	0.9936	0.2566	17.39%	99.36%	26%	17722	0.1739
2:00	8	9346	19	9327	0.002	3100	23968	0.9997	0.4201	0.20%	99.97%	42%	27068	0.1145
3:00	6	33091	1	33090	0	3101	57058	1	1	0.00%	100.00%	100%	60159	0.0515

The ranking predicted values for the pruned tree is measured for the values and the training model is measured for the cumulative respondents and the non-cumulative respondents is

measured for the deciles in 10, 8, 6 with the highest increased values for the count of the respondents of the rejected status in the claims and the values developed with the cumulative count in the less error rate of 0.0515 for the deciles 6.

	0	1
0	57018	97
1	40	3004

The cart model developed with accepted and rejected status for the claimed status in the variables and 3004 is predicted correctly and 97 is predicted wrong for the caret developed training model.

```
                                          Accuracy : 0.9977
                                95% CI : (0.9973, 0.9981)
                           No Information Rate : 0.9485
                           P-Value [Acc > NIR] : < 2.2e-16

                                           Kappa : 0.9765

                         Mcnemar's Test P-Value : 1.715e-06

                                    Sensitivity : 0.9993
                                    Specificity : 0.9687
                                 Pos Pred Value : 0.9983
                                 Neg Pred Value : 0.9869
                                      Precision : 0.9983
                                         Recall : 0.9993
                                             F1 : 0.9988
                                     Prevalence : 0.9485
                                 Detection Rate : 0.9478
                           Detection Prevalence : 0.9494
                              Balanced Accuracy : 0.9840

                               'Positive' Class : 0
```

The variables are predicted with confusion matrix is predicted for 99% of the dataset is predicted with variables and the values are measured for the increased values and sensitivity and specificity of the model is developed with the values are measured with 99% and 96% for the values with Recall of the 99% in the values of the predicted scores in the matrix values.

RF Model – Train

The predicted values are predicted for the train model developed with the values which are predicting the values and the values are tuned random forest model developed with trained data.

	deciles	cnt	cnt_resp	cnt_non_resp	rrate	cum_resp	cum_non_resp	cum_rel_resp	cum_rel_non_resp	rrate_perc	cum_rel_resp_perc	cum_rel_non_resp_perc	cum_cnt	cum_resp_rate	lift
1	10	6491	2354	4137	0.3627	2354	4137	0.7591	0.0725	36.30%	76%	7%	6491	0.3627	7.04
2	9	7723	362	7361	0.0469	2716	11498	0.8758	0.2015	4.70%	88%	20%	14214	0.1911	3.71
3	8	45945	385	45560	0.0084	3101	57058	1	1	0.80%	100%	100%	60159	0.0515	1

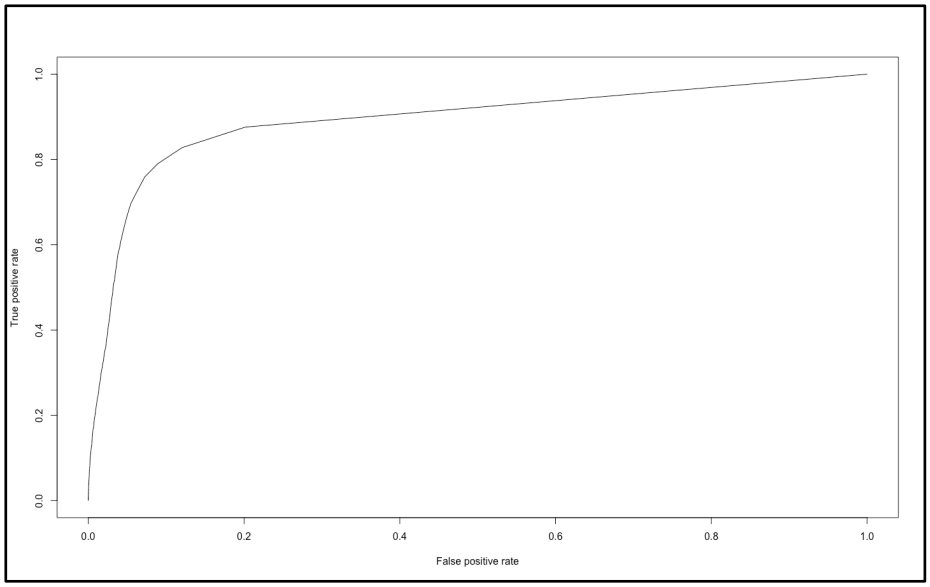
The ranking values are developed with the cumulative response and the non-cumulative response with rejected status is predicted for the cumulative respondents in the lift values with the tuned random forest is built with 100% non-cumulative response to the values to the values are increased with the values are measured for the deciles with 3.

0.05154673

The overall response is predicted for the 5% in the values in the variables are contacted for the values are measures with predicting values increased with the prediction of the tuned random forest.

7.04 3.71 1.00

The lift of the rank is developed with 7.04 for the 1st lift, 3.71 for the 2nd lift, 1st lift of value is 1 is measured with the various values for the increased values.



The false positive rate and true positive rate is measured for the values are increased for the more area in the values and the increased values are measured for the more area in the plots of the random forest predicted with tuning parameters

KS value is predicted with 70% for the increased stability in the performance of the variables.

The AUC value is measured with 89% of the values are measured for the increased values and the continued values in predicting the variables values for the increased values.

GINI impurity is measured for the 94% of the values is measured for the increased values and the values are predicted for the increased values in the impurity is decreased with the values for the increased values.

	0	1
0	57055	3069
1	3	32

Accuracy	: 0.9489
95% CI	: (0.9471, 0.9507)
No Information Rate	: 0.9485
P-Value [Acc > NIR]	: 0.3003
Kappa	: 0.0193
Mcnemar's Test P-Value	: <2e-16
Sensitivity	: 0.99995
Specificity	: 0.01032
Pos Pred Value	: 0.94896
Neg Pred Value	: 0.91429
Precision	: 0.94896
Recall	: 0.99995
F1	: 0.97378
Prevalence	: 0.94845
Detection Rate	: 0.94840
Detection Prevalence	: 0.99942
Balanced Accuracy	: 0.50513
'Positive' Class	: 0

The confusion matrix developed with values in 32 of the observations is considered for the values and the 3069 is predicted false of the train dataset in the train prediction of the random forest variables.

8. Annexure II

```
#=========================//////PROJECT NOTES
1//////=========================================
#================================Set Working
Directory=========================================
setwd("/Users/numerp/Documents/PGP-BABI/PGP-BABI Capstone Project 2020/Insurance
Claims")
getwd()
#================================Libraries
Loaded================================================
library(readxl)
library(readr)
library(dplyr)
library(psych)
library(ggplot2)
library(car)
library(DataExplorer)
library(ggcorrplot)
library(tidyverse)
library(janitor)
library(lubridate)
library(usdm)
library(raster)
library(lattice)
library(scales)
library(GGally)
library(vcd)
library(flexclust)
library(fpc)
library(factoextra)
library(rms)
library(pscl)
library(caret)
library(rpart)
library(rattle)
library(ROCR)
library(data.table)
library(ineq)
library(randomForest)
library(pROC)
library(ROSE)
library(xgboost)
library(DMwR)
library(NbClust)
library(cluster)
library(factoextra)
```

```
#================================Importing
Dataset=============================================
claim=read_excel("Insurance Claims Data.xlsx",sheet = 1)
dim(claim) #75200 Obs 32 Var
str(claim)
summary(claim)
#================================Variable and Values
Conversion==================================
claim$DRV_CLAIM_STATUS=as.factor(claim$DRV_CLAIM_STATUS)
summary(claim$DRV_CLAIM_STATUS)
claim$DRV_CLAIM_STATUS=factor(claim$DRV_CLAIM_STATUS,levels =
c("CLOSED","REJECTED"),
                      labels = c("0","1"))
summary(claim$DRV_CLAIM_STATUS)
prop.table(table(claim$DRV_CLAIM_STATUS))
#3876/(71234+3876) *****0.051*****
class(claim$DRV_CLAIM_STATUS)
claim$Boo_Endorsement=as.factor(claim$Boo_Endorsement)
summary(claim$Boo_Endorsement)
class(claim$Boo_Endorsement)
claim$Boo_TPPD_Statutory_Cover_only=as.factor(claim$Boo_TPPD_Statutory_Cover_only)
summary(claim$Boo_TPPD_Statutory_Cover_only)
class(claim$Boo_TPPD_Statutory_Cover_only)
claim$Boo_OD_Total_Loss=as.factor(claim$Boo_OD_Total_Loss)
summary(claim$Boo_OD_Total_Loss)
class(claim$Boo_OD_Total_Loss)
claim$Boo_AntiTheft=as.factor(claim$Boo_AntiTheft)
summary(claim$Boo_AntiTheft)
class(claim$Boo_AntiTheft)
claim$Boo_NCB=as.factor(claim$Boo_NCB)
summary(claim$Boo_NCB)
class(claim$Boo_NCB)
claim$Date_Accident_Loss=as_date(claim$Date_Accident_Loss)
summary(claim$Date_Accident_Loss)
class(claim$Date_Accident_Loss)
claim$Date_Claim_Intimation=as_date(claim$Date_Claim_Intimation)
summary(claim$Date_Claim_Intimation)
class(claim$Date_Claim_Intimation)
claim$Date_Disbursement=as_date(claim$Date_Disbursement)
summary(claim$Date_Disbursement)
class(claim$Date_Disbursement)
claim$Txt_Policy_Year
claim$Txt_Policy_Year=factor(claim$Txt_Policy_Year,
                      levels = c("1998-99","1999-00","2000-01","2001-02","2002-03",
                             "2003-04","2004-05","2005-06","2006-07","2007-08",
                             "2008-09","2009-10","2010-11","2011-12","2012-13"),
                      labels = c("1998","1999","2000","2001","2002","2003","2004",
```

```
                        "2005","2006","2007","2008","2009","2010","2011",
                        "2012"))
summary(claim$Txt_Policy_Year)
claim$Txt_Policy_Year=as.character(claim$Txt_Policy_Year)
claim$Txt_Policy_Year=as_date(strptime(claim$Txt_Policy_Year,format ='%Y'))
summary(claim$Txt_Policy_Year)
class(claim$Txt_Policy_Year)
claim$Txt_Claim_Year
claim$Txt_Claim_Year=factor(claim$Txt_Claim_Year,
                levels = c("1999-00","2000-01","2001-02","2002-03",
                        "2003-04","2004-05","2005-06","2006-07","2007-08",
                        "2008-09","2009-10","2010-11","2011-12","2012-13"),
                labels = c("1999","2000","2001","2002","2003","2004",
                        "2005","2006","2007","2008","2009","2010","2011",
                        "2012"))
claim$Txt_Claim_Year=as.character(claim$Txt_Claim_Year)
claim$Txt_Claim_Year=as_date(strptime(claim$Txt_Claim_Year,format ='%Y'))
summary(claim$Txt_Claim_Year)
class(claim$Txt_Claim_Year)
claim$Num_Vehicle_Age=as.numeric(claim$Num_Vehicle_Age)
str(claim)
#================================Colnames
Edit=========================================================
names(claim)
names(claim)[2]="Policy_Year"
names(claim)[3]="Endorsement"
names(claim)[4]="Location_RTA"
names(claim)[5]="Policy"
names(claim)[6]="Vehicle_Class"
names(claim)[7]="Zone"
names(claim)[8]="Vehicle_Age"
names(claim)[9]="Vehicle_CC"
names(claim)[10]="Vehicle_Colour"
names(claim)[11]="IDV"
names(claim)[12]="Permit"
names(claim)[13]="Nature_of_Goods"
names(claim)[14]="Road_Type"
names(claim)[15]="Driver_Type"
names(claim)[16]="Driver_Exp"
names(claim)[17]="Claims_History"
names(claim)[18]="Driver_Qualification"
names(claim)[19]="Incurred_Claims"
names(claim)[20]="Statutory_Cover"
names(claim)[21]="Claim_Year"
names(claim)[22]="Accident_Date"
names(claim)[23]="Accident_Place"
names(claim)[24]="Claim_Intimation_Date"
```

```
names(claim)[25]="Nature_of_Loss"
names(claim)[26]="Disbursement_Date"
names(claim)[27]="Total_Loss"
names(claim)[28]="Claim_Amount"
names(claim)[29]="Claim_Status"
names(claim)[30]="Antitheft"
names(claim)[31]="Discount_NCB"
names(claim)[32]="Net_Premium"
colnames(claim)
#================================Duplicate Value
Treatment=========================================
claim=subset(claim,select = -c(1))
get_dupes(claim)
claim=unique(claim)
#get_dupes(claim)
#================================Variable
Classification=====================================================
names(claim)
num_var=subset(claim,select = c(7,10,27,31))
names(num_var)
head(num_var,4)
cat_var=subset(claim,select = -c(1,2,7,10,19,20,21,23,25,26,27:31))
names(cat_var)
head(cat_var,4)
factor_var=subset(claim,select = -c(1,3:18,20:25,27,31))
names(factor_var)
head(factor_var,4)
date_var=subset(claim,select = c(1,20,21,23,25))
names(date_var)
head(date_var,4)
#================================Missing Value
Treatment=========================================
summary(claim)
colSums(is.na(claim))
class(claim)
claim=as.data.frame(claim)
claim$Disbursement_Date[which(is.na(claim$Disbursement_Date))] =
median(claim$Disbursement_Date,
                                          na.rm = TRUE)
any(is.na(claim))
summary(claim$Disbursement_Date)
#================================Outlier
Treatment=================================================
summary(claim[,c(7,10,27,31)])
boxplot(claim[c(7,10,27,31)],plot = FALSE)$out
IQRage = IQR(claim$Vehicle_Age)
LLage = quantile(claim$Vehicle_Age,0.25) - 1.5*IQRage
```

```
ULage = quantile(claim$Vehicle_Age,0.75) + 1.5*IQRage
ageOut = subset(claim, Vehicle_Age >= LLage & Vehicle_Age <= ULage)
dim(ageOut)
max(ageOut$Vehicle_Age)
summary(ageOut$Vehicle_Age)
claim$Vehicle_Age[claim$Vehicle_Age > 10] = 10
summary(claim$Vehicle_Age)
claim$IDV=squish(claim$IDV,round(quantile(claim$IDV,c(0.5,0.95))))
summary(claim$IDV)
claim$Claim_Amount=squish(claim$Claim_Amount,round(quantile(claim$Claim_Amount,c(0
.5,0.95))))
summary(claim$Claim_Amount)
claim$Net_Premium=squish(claim$Net_Premium,round(quantile(claim$Net_Premium,c(0.5,
0.95))))
summary(claim$Net_Premium)
summary(claim[,c(7,10,27,31)])
#================================Univariate
Analysis=============================================
attach(claim)
boxplot(num_var)
plot_bar(data = factor_var,title = "Benefits Summary",ggtheme = theme_minimal())
plot_bar(data = cat_var,title = "Summary",ncol = 4,ggtheme = theme_minimal())
plot_bar(data = claim[,c(2,14,15,12,16,17,19,20,26,31)],nrow = 3,ncol = 3,ggtheme =
theme_minimal())
#par(mfrow=c(2,2))
hist(Claim_Amount,label=TRUE,col = "red")
hist(Vehicle_Age,label=TRUE,col = "grey")
hist(IDV,label=TRUE,col = "beige")
hist(Net_Premium,label=TRUE, col = "gold")
boxplot(Claim_Amount,label =TRUE, horizontal =TRUE, col = "red",main = "Boxplot - Claim
Amount")
boxplot(Vehicle_Age,label =TRUE, horizontal =TRUE, col = "grey",main = "Boxplot - Vehicle
Age")
boxplot(IDV,label =TRUE, horizontal =TRUE, col = "beige",main = "Boxplot - IDV")
boxplot(Net_Premium,label =TRUE, horizontal =TRUE, col = "gold",main = "Boxplot - Net
Premium")
#================================Bivariate
Analysis=============================================
plot_boxplot(data = claim,by="Claim_Status",ncol = 2,nrow = 3,ggtheme = theme_minimal())
#Policy Coverages
ggplot(data = claim,aes(x=Policy,fill=Claim_Status))+geom_bar(position = "stack")+
  theme_minimal()
ggplot(data = claim,aes(x=Vehicle_CC,fill=Claim_Status))+geom_bar(position = "stack")+
  theme_minimal()
ggplot(data = claim,aes(x=Zone,fill=Claim_Status))+geom_bar(position = "stack")+
  theme_minimal()
ggplot(data = claim,aes(x=Permit,fill=Claim_Status))+geom_bar(position = "stack")+
```

```
 theme_minimal()
ggplot(data = claim,aes(x=Nature_of_Goods,fill=Claim_Status))+geom_bar(position =
"stack")+
 theme_minimal()
ggplot(data = claim,aes(x=Nature_of_Loss,fill=Claim_Status))+geom_bar(position = "stack")+
 theme_minimal()
ggplot(data = claim,aes(x=Road_Type,fill=Claim_Status))+geom_bar(position = "stack")+
 theme_minimal()
ggplot(data = claim,aes(x=Driver_Type,fill=Claim_Status))+geom_bar(position = "stack")+
 theme_minimal()
ggplot(data = claim,aes(x=Driver_Exp,fill=Claim_Status))+geom_bar(position = "stack")+
 theme_minimal()
ggplot(data = claim,aes(x=Driver_Qualification,fill=Claim_Status))+geom_bar(position =
"stack")+
 theme_minimal()
ggplot(data = claim,aes(x=Claims_History,fill=Claim_Status))+geom_bar(position = "stack")+
 theme_minimal()
ggplot(data = claim,aes(x=Incurred_Claims,fill=Claim_Status))+geom_bar(position = "stack")+
 theme_minimal()
ggplot(data =
claim,aes(x=Diff_Claim_Days,fill=Claim_Status))+geom_dotplot()+theme_minimal()
#Additional Benefits
ggplot(data = claim,aes(x=Statutory_Cover,fill=Claim_Status))+geom_bar(position =
"stack")+
 theme_minimal()
ggplot(data = claim,aes(x=Endorsement,fill=Claim_Status))+geom_bar(position = "stack")+
 theme_minimal()
ggplot(data = claim,aes(x=Antitheft,fill=Claim_Status))+geom_bar(position = "stack")+
 theme_minimal()
ggplot(data = claim,aes(x=Discount_NCB,fill=Claim_Status))+geom_bar(position = "stack")+
 theme_minimal()
ggplot(data = claim,aes(x=Total_Loss,fill=Claim_Status))+geom_bar(position = "stack")+
 theme_minimal()
scatterplot(Vehicle_Age,Claim_Amount,data=claim,ellipse = TRUE)
scatterplot(Vehicle_Age,Net_Premium,data=claim,ellipse = TRUE)
ggplot(data = claim,aes(x=Policy_Year,y=IDV,color=Policy,shape=Claim_Status))+
 geom_point(size=3,alpha=0.6)
ggplot(data = claim,aes(x=Vehicle_Age,y=IDV,color=Claims_History,shape=Claim_Status))+
 geom_point(size=3,alpha=0.6)
ggplot(data = claim,aes(x=Vehicle_Class,y=IDV,color=Claims_History,shape=Claim_Status))+
 geom_point(size=3,alpha=0.6)
ggplot(data =
claim,aes(x=Vehicle_Age,y=Policy_Year,color=Claims_History,shape=Claim_Status))+
 geom_point(size=3,alpha=0.6)
#================================Cross
Tables=====================================================
table(Claim_Status,Zone)
```

```
table(Claim_Status,Policy)
table(Claim_Status,Vehicle_Age)
table(Claim_Status,Vehicle_Class)
table(Claim_Status,Claims_History)
table(Claim_Status,Driver_Type)
table(Claim_Status,Road_Type)
table(Antitheft,Claim_Status)
table(Discount_NCB,Claim_Status)
table(Endorsement,Claim_Status)
table(Statutory_Cover,Claim_Status)
table(Total_Loss,Claim_Status)
#================================Correlation================================
====================
correlation=cor(claim[,c(7,10,27,31)])
correlation
corrplot::corrplot(correlation)
#================================Variance Inflation
Factor=====================================
num_data=unlist(lapply(claim,is.numeric))
usdm::vif(subset(claim,select=(num_data)))
#=========================///////PROJECT NOTES
2/////==============================================
#================================Splitting
Data=================================================
library(caTools)
set.seed(123)
my_split=sample.split(claim$Claim_Status,SplitRatio = 0.8)
my_train=subset(claim,my_split==TRUE)
my_test=subset(claim,my_split==FALSE)
table(my_train$Claim_Status)
prop.table(table(my_train$Claim_Status))
table(my_test$Claim_Status)
prop.table(table(my_test$Claim_Status))
attach(claim)
#================================Model
Building============================================
#================================Logistics
Regression==========================================
model1=glm(Claim_Status~. -Location_RTA -Vehicle_Colour -Accident_Place,
        data = my_train,family = binomial(link = logit))
summary(model1)
#Removing Insignificant Variables
model2=glm(Claim_Status~Endorsement+Vehicle_Class+Zone+Vehicle_Age+Permit+Nature_
of_Goods+
          Road_Type+Driver_Type+Driver_Exp+Claims_History+Driver_Qualification+
          Statutory_Cover+Claim_Year+Claim_Intimation_Date+
          Total_Loss+Net_Premium,data = my_train,family = binomial(link = logit))
```

```
summary(model2)
car::vif(model2)
#Removing higher Variance Inflation Rate Variables
model3=glm(Claim_Status~Endorsement+Vehicle_Class+Zone+Vehicle_Age+Nature_of_Goo
ds+
          Driver_Type+Driver_Exp+Claims_History+Driver_Qualification+
          Statutory_Cover+Claim_Year+Claim_Intimation_Date+Total_Loss+
          Net_Premium,data = my_train,family = binomial(link = logit))
summary(model3)
#Removing Insignificant Variables
model4=glm(Claim_Status~Endorsement+Vehicle_Class+Zone+Nature_of_Goods+Driver_Ty
pe+
          Driver_Exp+Claims_History+Driver_Qualification+Statutory_Cover+Claim_Year+
          Claim_Intimation_Date+Total_Loss+Net_Premium,
        data = my_train,family = binomial(link = logit))
summary(model4)
car::vif(model4)
#Removing Higher Variance Inflation Rate Variables
model5=glm(Claim_Status~Endorsement+Driver_Type+Driver_Exp+Nature_of_Goods+Claim
s_History+
          Driver_Qualification+Statutory_Cover+Total_Loss+
          Net_Premium,data = my_train,family = binomial(link = logit))
summary(model5)
car::vif(model5)
exp(coef(model5))
exp(coef(model5))/(1+exp(coef(model5)))
#McFadden 0to0.10 - Bad,0.10to0.15 - Average,0.15to0.3 - Moderate,0.3to0.5 - Good,>0.5
Excellent
pscl::pR2(model5)["McFadden"]
logLik(model5)
#================================Prediction on Train - LR
Model=======================================
LR_pred=predict(model5,newdata = my_train,type = "response")
roc_pred=prediction(LR_pred,my_train$Claim_Status)
as.numeric(performance(roc_pred,"auc")@y.values)
perf=performance(roc_pred,"tpr","fpr")
plot(perf)
plot(perf,colorize=TRUE,print.cutoffs.at=seq(0,1,.1),text.adj=c(-.2,1.7))
LR_table=table(my_train$Claim_Status,LR_pred>0.1)
LR_table
sum(diag(LR_table))/sum(LR_table)
1540/(1540+1561) #Recall 0.49
1540/(1540+6926) #Precision 0.18
pred=ifelse(model5$fitted.values>0.1,1,0)
actual=my_train$Claim_Status
cm_log=caret::confusionMatrix(as.factor(pred),actual,positive="1")
cm_log
```

```
#================================Top 10 Deciles
Ranking=============================================
decile <- function(x){
 deciles <- vector(length=10)
 for (i in seq(0.1,1,.1)){
  deciles[i*10] <- quantile(x, i, na.rm=T)
 }
 return (
  ifelse(x<deciles[1], 1,
       ifelse(x<deciles[2], 2,
            ifelse(x<deciles[3], 3,
                 ifelse(x<deciles[4], 4,
                      ifelse(x<deciles[5], 5,
                           ifelse(x<deciles[6], 6,
                                ifelse(x<deciles[7], 7,
                                     ifelse(x<deciles[8], 8,
                                          ifelse(x<deciles[9], 9, 10
                                          ))))))))))
}
#================================Ranking - Train LR
Model=======================================
LR_train=my_train
LR_train$pred=predict(model5,LR_train,type = "response")
LR_train$deciles=decile(LR_train$pred)
m=data.table::data.table(LR_train)
rank_lr = m[, list(cnt=length(Claim_Status),
                   cnt_resp=sum(Claim_Status==1),
                   cnt_non_resp=sum(Claim_Status==0)
), by=deciles][order(-deciles)]
rank_lr$rrate=round(rank_lr$cnt_resp/rank_lr$cnt,4)
rank_lr$cum_resp=cumsum(rank_lr$cnt_resp)
rank_lr$cum_non_resp=cumsum(rank_lr$cnt_non_resp)
rank_lr$cum_rel_resp=round(rank_lr$cum_resp/sum(rank_lr$cnt_non_resp),4)
rank_lr$cum_rel_non_resp=round(rank_lr$cum_non_resp/sum(rank_lr$cnt_non_resp),4)
rank_lr$ks=abs(rank_lr$cum_rel_resp - rank_lr$cum_rel_non_resp)*100
rank_lr$rrate=scales::percent(rank_lr$rrate)
rank_lr$cum_rel_resp=scales::percent(rank_lr$cum_rel_resp)
rank_lr$cum_rel_non_resp=scales::percent(rank_lr$cum_rel_non_resp)
LR_train_rank=rank_lr
print(LR_train_rank)
plot(LR_train_rank$ks)
lines(LR_train_rank$ks)
#================================Prediction on Test - LR
Model=======================================
LR_pred1=predict.glm(model5,newdata = my_test,type = "response")
LR_pred1
roc_pred1=prediction(LR_pred1,my_test$Claim_Status)
```

```
as.numeric(performance(roc_pred1,"auc")@y.values)
perf1=performance(roc_pred1,"tpr","fpr")
plot(perf1)
plot(perf1,colorize=TRUE,print.cutoffs.at=seq(0,1,.1),text.adj=c(-.1,1.7))
LR_table1=table(my_test$Claim_Status,LR_pred1>0.1)
LR_table1
sum(diag(LR_table1))/sum(LR_table1)
387/(387+388) #Recall 0.49
387/(387+1711) #Precision 0.18
pred1=ifelse(LR_pred1>0.1,1,0)
actual1=my_test$Claim_Status
cm_log1=caret::confusionMatrix(as.factor(pred1),actual1,positive="1")
cm_log1
#================================Ranking - Test LR
Model==========================================
LR_test=my_test
LR_test$pred=predict(model5,LR_test,type = "response")
LR_test$deciles=decile(LR_test$pred)
n=data.table::data.table(LR_test)
rank_lr1 = n[, list(cnt=length(Claim_Status),
              cnt_resp=sum(Claim_Status==1),
              cnt_non_resp=sum(Claim_Status==0)
), by=deciles][order(-deciles)]
rank_lr1$rrate=round(rank_lr1$cnt_resp/rank_lr1$cnt,4)
rank_lr1$cum_resp=cumsum(rank_lr1$cnt_resp)
rank_lr1$cum_non_resp=cumsum(rank_lr1$cnt_non_resp)
rank_lr1$cum_rel_resp=round(rank_lr1$cum_resp/sum(rank_lr1$cnt_non_resp),4)
rank_lr1$cum_rel_non_resp=round(rank_lr1$cum_non_resp/sum(rank_lr1$cnt_non_resp),4
)
rank_lr1$ks=abs(rank_lr1$cum_rel_resp - rank_lr1$cum_rel_non_resp)*100
rank_lr1$rrate=scales::percent(rank_lr1$rrate)
rank_lr1$cum_rel_resp=scales::percent(rank_lr1$cum_rel_resp)
rank_lr1$cum_rel_non_resp=scales::percent(rank_lr1$cum_rel_non_resp)
LR_test_rank=rank_lr1
print(LR_test_rank)
plot(LR_test_rank$ks)
lines(LR_test_rank$ks)
#================================KS, ROC, AUC, GINI - LR
Model=======================================
roc_curve=roc(my_train$Claim_Status,model5$fitted.values)
plot.roc(roc_curve)
AUC=as.numeric(performance(roc_pred1,"auc")@y.values)
AUC
KS=max(attr(perf1, 'y.values')[[1]]-attr(perf1, 'x.values')[[1]])
KS
GINI=ineq(LR_pred1,type = "Gini")
GINI
```

```
#================================CART=================================
==========================
modelcart=rpart(formula = Claim_Status ~ .,data = my_train,method = "class")
printcp(modelcart)
modelcart
names(my_train)
tree_control=rpart.control(minsplit=99, minbucket = 4, cp = 0, xval = 4)
#Removing Dates from CART Model
tree=rpart(formula = Claim_Status ~ .,data = my_train[,-c(3,9,22,24,1,20,21,23,25)], method
= "class",
                control = tree_control)
printcp(tree)
tree
rpart::plotcp(tree)
#Removing Claim Amount, since it is significant only
tree_control1=rpart.control(minsplit = 250,minbucket = 5,cp = 0,xval = 5)
tree1=rpart(formula = Claim_Status ~ .,data = my_train[,-c(3,9,22,24,1,20,21,23,25,27)],
         method = "class",control = tree_control1)
printcp(tree1)
tree1
rpart::plotcp(tree1)
rattle::fancyRpartPlot(tree1)
tree1$cptable[which.min(tree1$cptable[,"xerror"]),"CP"]
ptree=prune(tree1,cp=0.00064495,"CP")
printcp(ptree)
plotcp(ptree)
fancyRpartPlot(ptree)
ptree
#================================Prediction on Train - CART
Model=====================================
train_cart=my_train
train_cart$predict=predict(ptree,train_cart,type="class")
train_cart$score=predict(ptree,train_cart,type="prob")
train_cart$deciles=decile(train_cart$score[,2])
#================================Ranking - Train CART
Model=============================================
table_train_cart = data.table(train_cart)
rank = table_train_cart[, list(
 cnt = length(as.integer(as.character(Claim_Status))),
 cnt_resp = sum(as.integer(as.character(Claim_Status))),
 cnt_non_resp = sum(as.integer(as.character(Claim_Status)) == 0)) ,
 by=deciles][order(-deciles)]
rank$rrate = round(rank$cnt_resp / rank$cnt,4);
rank$cum_resp = cumsum(rank$cnt_resp)
rank$cum_non_resp = cumsum(rank$cnt_non_resp)
rank$cum_rel_resp = round(rank$cum_resp  / sum(rank$cnt_resp),4);
rank$cum_rel_non_resp = round(rank$cum_non_resp  / sum(rank$cnt_non_resp),4);
```

```
rank$rrate_perc = percent(rank$rrate)
rank$cum_rel_resp_perc = percent(rank$cum_rel_resp)
rank$cum_rel_non_resp_perc = percent(rank$cum_rel_non_resp)
rank$cum_cnt = cumsum(rank$cnt)
rank$cum_resp_rate = round(rank$cum_resp  / rank$cum_cnt,4)
overall_resp_rate = sum(as.integer(as.character(train_cart$Claim_Status)))/nrow(train_cart)
rank
overall_resp_rate
#================================Lift - Train CART
Model=============================================
rank$lift=round(rank$cum_resp_rate/overall_resp_rate,2)
rank$lift
#================================KS, AUC, GINI - Train CART
Model=================================
pred_train_cart = prediction(train_cart$score[,2], train_cart$Claim_Status)
perf_train_cart = performance(pred_train_cart, "tpr", "fpr")
plot(perf_train_cart)
KS_train_cart = max(attr(perf_train_cart, 'y.values')[[1]]-attr(perf_train_cart, 'x.values')[[1]])
KS_train_cart
auc_train_cart = performance(pred_train_cart,"auc");
performance(pred_train_cart,"auc")
gini_train_cart = ineq(train_cart$score[,2], type="Gini")
gini_train_cart
cm_train_cart=confusionMatrix((table(train_cart$predict,train_cart$Claim_Status)),
                      mode = "everything")
cm_train_cart
#================================Prediction on Test - CART
Model=================================
test_cart=my_test
test_cart$predict=predict(ptree,test_cart,type="class")
test_cart$score=predict(ptree,test_cart,type="prob")
test_cart$deciles=decile(test_cart$score[,2])
#================================Ranking - Test CART
Model=============================================
table_test_cart = data.table(test_cart)
rank1 = table_test_cart[, list(
 cnt = length(as.integer(as.character(Claim_Status))),
 cnt_resp = sum(as.integer(as.character(Claim_Status))),
 cnt_non_resp = sum(as.integer(as.character(Claim_Status)) == 0)) ,
 by=deciles][order(-deciles)]
rank1$rrate = round(rank1$cnt_resp / rank1$cnt,4);
rank1$cum_resp = cumsum(rank1$cnt_resp)
rank1$cum_non_resp = cumsum(rank1$cnt_non_resp)
rank1$cum_rel_resp = round(rank1$cum_resp  / sum(rank1$cnt_resp),4);
rank1$cum_rel_non_resp = round(rank1$cum_non_resp  / sum(rank1$cnt_non_resp),4);
rank1$rrate_perc = percent(rank1$rrate)
rank1$cum_rel_resp_perc = percent(rank1$cum_rel_resp)
```

```
rank1$cum_rel_non_resp_perc = percent(rank1$cum_rel_non_resp)
rank1$cum_cnt = cumsum(rank1$cnt)
rank1$cum_resp_rate = round(rank1$cum_resp  / rank1$cum_cnt,4)
overall_resp_rate1 = sum(as.integer(as.character(test_cart$Claim_Status)))/nrow(test_cart)
rank1
overall_resp_rate1
#================================Lift - Test CART
Model==============================================
rank1$lift=round(rank1$cum_resp_rate/overall_resp_rate1,2)
rank1$lift
plot(rank1$lift)
lines(rank1$lift)
#================================KS, AUC, GINI - Test CART
Model====================================
pred_test_cart = prediction(test_cart$score[,2], test_cart$Claim_Status)
perf_test_cart = performance(pred_test_cart, "tpr", "fpr")
plot(perf_test_cart)
KS_test_cart = max(attr(perf_test_cart, 'y.values')[[1]]-attr(perf_test_cart, 'x.values')[[1]])
KS_test_cart
auc_test_cart <- performance(pred_test_cart,"auc");
performance(pred_test_cart,"auc")
gini_test_cart = ineq(test_cart$score[,2], type="Gini")
gini_test_cart
cm_test_cart=confusionMatrix((table(test_cart$predict,test_cart$Claim_Status)),
                      mode = "everything")
cm_test_cart
#================================Random
Forest=============================================================
set.seed(123)
rftrain=subset(my_train,select=-c(3,9,22,24,1,20,21,23,25))
rftest=subset(my_test,select=-c(3,9,22,24,1,20,21,23,25))
#Removing Higher Level Variables and Date Variables
rf_claim=randomForest::randomForest(as.factor(Claim_Status) ~ .,
                           data = rftrain,
                           ntree = 101,mtry = 5, nodesize = 100,importance = TRUE)
print(rf_claim)
plot(rf_claim, main="")
legend("topright", c("OOB", "0", "1"), text.col=1:6, lty=1:3, col=1:3)
title(main="Error Rates Random Forest")
rf_claim$err.rate
importance(rf_claim)
#Removing Claim Amount
names(rftrain)
rf_claim1=randomForest::randomForest(as.factor(Claim_Status) ~ .,
                           data = rftrain[,-18],
                           ntree = 101,mtry = 5, nodesize = 100,importance = TRUE)
print(rf_claim1)
```

```
plot(rf_claim1, main="")
legend("topright", c("OOB", "0", "1"), text.col=1:6, lty=1:3, col=1:3)
title(main="Error Rates Random Forest")
rf_claim1$err.rate
important_var=round(importance(rf_claim1),2)
important_var[order(important_var[,4],decreasing = TRUE),]
#tune RF Model
set.seed(1234)
tune_rf_claim=tuneRF(x = rftrain[,-c(18,19)],
                     y=rftrain$Claim_Status,
                     mtryStart = 5,
                     ntreeTry = 61,
                     stepFactor = 1.2,
                     improve = 1.1,
                     trace = TRUE,
                     plot = TRUE,
                     doBest = TRUE,
                     nodesize = 100,
                     importance = TRUE
                     )
tune_rf_claim
importance(tune_rf_claim)
important_var=round(importance(tune_rf_claim),2)
important_var
important_var[order(important_var[,4],decreasing = TRUE),]
plot(tune_rf_claim)
legend("topright", c("OOB", "0", "1"), text.col=1:6, lty=1:3, col=1:3)
#================================Prediction on Train - RF
Model=====================================
train_rf=rftrain
train_rf$predict=predict(tune_rf_claim,train_rf,type="class")
train_rf$score=predict(tune_rf_claim,train_rf,type="prob")
train_rf$deciles=decile(train_rf$score[,2])
#================================Ranking - Train RF
Model=============================================
table_train_rf = data.table(train_rf)
rank_rf = table_train_rf[, list(
 cnt = length(as.integer(as.character(Claim_Status))),
 cnt_resp = sum(as.integer(as.character(Claim_Status))),
 cnt_non_resp = sum(as.integer(as.character(Claim_Status)) == 0)) ,
 by=deciles][order(-deciles)]
rank_rf$rrate = round(rank_rf$cnt_resp / rank_rf$cnt,4);
rank_rf$cum_resp = cumsum(rank_rf$cnt_resp)
rank_rf$cum_non_resp = cumsum(rank_rf$cnt_non_resp)
rank_rf$cum_rel_resp = round(rank_rf$cum_resp  / sum(rank_rf$cnt_resp),4);
rank_rf$cum_rel_non_resp = round(rank_rf$cum_non_resp  /
sum(rank_rf$cnt_non_resp),4);
```

```
rank_rf$rrate_perc = percent(rank_rf$rrate)
rank_rf$cum_rel_resp_perc = percent(rank_rf$cum_rel_resp)
rank_rf$cum_rel_non_resp_perc = percent(rank_rf$cum_rel_non_resp)
rank_rf$cum_cnt = cumsum(rank_rf$cnt)
rank_rf$cum_resp_rate = round(rank_rf$cum_resp  / rank_rf$cum_cnt,4)
overall_resp_rate_rf = sum(as.integer(as.character(train_rf$Claim_Status)))/nrow(train_rf)
rank_rf
overall_resp_rate_rf
#================================Lift - Train RF
Model==============================================
rank_rf$lift=round(rank_rf$cum_resp_rate/overall_resp_rate_rf,2)
rank_rf$lift
#================================KS, AUC, GINI - Train RF
Model======================================
pred_train_rf = prediction(train_rf$score[,2], train_rf$Claim_Status)
perf_train_rf = performance(pred_train_rf, "tpr", "fpr")
plot(perf_train_rf)
KS_train_rf = max(attr(perf_train_rf, 'y.values')[[1]]-attr(perf_train_rf, 'x.values')[[1]])
KS_train_rf
auc_train_rf = performance(pred_train_rf,"auc");
performance(pred_train_rf,"auc")
gini_train_rf = ineq(train_rf$score[,2], type="Gini")
gini_train_rf
cm_train_rf=confusionMatrix((table(train_rf$predict,train_rf$Claim_Status)),
                        mode = "everything")
cm_train_rf
#================================Prediction on Test - RF
Model=====================================
test_rf=rftest
test_rf$predict=predict(tune_rf_claim,test_rf,type="class")
test_rf$score=predict(tune_rf_claim,test_rf,type="prob")
test_rf$deciles=decile(test_rf$score[,2])
#================================Ranking - Test RF
Model============================================
table_test_rf = data.table(test_rf)
rank_rf1 = table_test_rf[, list(
 cnt = length(as.integer(as.character(Claim_Status))),
 cnt_resp = sum(as.integer(as.character(Claim_Status))),
 cnt_non_resp = sum(as.integer(as.character(Claim_Status)) == 0)) ,
 by=deciles][order(-deciles)]
rank_rf1$rrate = round(rank_rf1$cnt_resp / rank_rf1$cnt,4);
rank_rf1$cum_resp = cumsum(rank_rf1$cnt_resp)
rank_rf1$cum_non_resp = cumsum(rank_rf1$cnt_non_resp)
rank_rf1$cum_rel_resp = round(rank_rf1$cum_resp  / sum(rank_rf1$cnt_resp),4);
rank_rf1$cum_rel_non_resp = round(rank_rf1$cum_non_resp  /
sum(rank_rf1$cnt_non_resp),4);
rank_rf1$rrate_perc = percent(rank_rf1$rrate)
```

```
rank_rf1$cum_rel_resp_perc = percent(rank_rf1$cum_rel_resp)
rank_rf1$cum_rel_non_resp_perc = percent(rank_rf1$cum_rel_non_resp)
rank_rf1$cum_cnt = cumsum(rank_rf1$cnt)
rank_rf1$cum_resp_rate = round(rank_rf1$cum_resp  / rank_rf1$cum_cnt,4)
overall_resp_rate_rf1 = sum(as.integer(as.character(test_rf$Claim_Status)))/nrow(test_rf)
rank_rf1
overall_resp_rate_rf1
#================================Lift - Test RF
Model==============================================
rank_rf1$lift=round(rank_rf1$cum_resp_rate/overall_resp_rate_rf1,2)
rank_rf1$lift
#================================KS, AUC, GINI - Test RF
Model====================================
pred_test_rf = prediction(test_rf$score[,2], test_rf$Claim_Status)
perf_test_rf = performance(pred_test_rf, "tpr", "fpr")
plot(perf_test_rf)
KS_test_rf = max(attr(perf_test_rf, 'y.values')[[1]]-attr(perf_test_rf, 'x.values')[[1]])
KS_test_rf
auc_test_rf <- performance(pred_test_rf,"auc");
performance(pred_test_rf,"auc")
gini_test_rf = ineq(test_rf$score[,2], type="Gini")
gini_test_rf
cm_test_rf=confusionMatrix((table(test_rf$predict,test_rf$Claim_Status)),
                      mode = "everything")
cm_test_rf
#================================Extreme Gradient
Boosting=========================================
xgbtrain=my_train
xgbtest=my_test
xgbftrain=as.matrix(xgbtrain[,c(7,10,27,31)])
xgbltrain=as.matrix(xgbtrain[,28])
xgbftest=as.matrix(xgbtest[,c(7,10,27,31)])
xgbfit=xgboost::xgboost(
 data = xgbftrain,
 label = xgbltrain,
 eta = 0.001,
 max_depth = 3,
 min_child_weight = 3,
 nrounds = 100,
 nfold = 5,
 objective = "binary:logistic",
 verbose = 0,
 early_stopping_rounds = 10
)
xgbfit
xgbtest$pred.class.xgb=predict(xgbfit,xgbftest)
table.xgb=table(xgbtest$Claim_Status,xgbtest$pred.class.xgb>0.5)
```

```
table.xgb
#Extreme Gradient Boosting Tuning
t.xgb=vector()
l=c(0.001, 0.01, 0.1, 0.3, 0.5, 0.7, 1)
m=c(1,3,5,7,9,15)
n=c(2, 50, 100,1000,10000)
for (i in l) {
 xgbfit=xgboost::xgboost(
  data = xgbftrain,
  label = xgbltrain,
  eta = i,
  max_depth = 5,
  nrounds = 10,
  nfold = 5,
  objective = "binary:logistic",
  verbose = 0,
  early_stopping_rounds = 10
 )
 xgbtest$pred.class.xgb=predict(xgbfit,xgbftest)
 t.xgb=cbind(t.xgb,sum(xgbtest$Claim_Status==1 & xgbtest$pred.class.xgb>=0.5))
}
t.xgb
#Best Fit
xgbtest$pred.class.xgb1=predict(xgbfit,xgbftest)
sum(xgbtest$Claim_Status==1 & xgbtest$pred.class.xgb1>=0.5)
table.xgb1=table(xgbtest$Claim_Status,xgbtest$pred.class.xgb1>=0.5)
table.xgb1
xgbtest$pred.class.xgb1=ifelse(xgbtest$pred.class.xgb1<0.5,0,1)
cm_xgb=caret::confusionMatrix(data = factor(xgbtest$pred.class.xgb1),
                      reference = factor(xgbtest$Claim_Status),
                      positive = "1")
cm_xgb
#================================SMOTE=================================
=========================
smote_train=my_train[,c(2,7,10,19,26,27:31)]
smote_test=my_test[,c(2,7,10,19,26,27:31)]
Balanced.data=SMOTE(Claim_Status~.,data = smote_train,perc.over = 400,k=5,perc.under =
100)
table(Balanced.data$Claim_Status)
Balanced.data[,c(1,4,5,8,9)]=as.numeric(unlist(Balanced.data[,c(1,4,5,8,9)]))
smote_test[,c(1,4,5,8,9)]=as.numeric(unlist(smote_test[,c(1,4,5,8,9)]))
sftrain=as.matrix(Balanced.data[,-c(7)])
sltrain=as.matrix(Balanced.data$Claim_Status)
smote.xgb=xgboost::xgboost(
 data = sftrain,
 label = sltrain,
 eta = 0.7,
```

```
 max_depth = 5,
 nrounds = 50,
 nfold = 5,
 objective = "binary:logistic",
 verbose = 0,
 early_stopping_rounds = 10
)
smote.xgb
sftest=as.matrix(smote_test[,-c(7)])
smote_test$pred.class.smote=predict(smote.xgb,sftest)
smote_test$pred.class.smote=ifelse(smote_test$pred.class.smote<0.5,0,1)
cm_smote=caret::confusionMatrix(data = factor(smote_test$pred.class.smote),
                        reference = factor(smote_test$Claim_Status),
                        positive = "1")
cm_smote
table.smote=table(smote_test$Claim_Status,smote_test$pred.class.smote>=0.5)
table.smote
sum(smote_test$Claim_Status==1 & smote_test$pred.class.smote >= 0.5)
#================================K-Means
Clustering==============================================
num_var1=subset(claim,select=c(7,10,27,31))
num_var1=sapply(num_var1,as.numeric)
head(num_var1)
claims_scaled = scale(num_var1)
head(claims_scaled)
set.seed(123)
K_cluster = kmeans(x=claims_scaled, centers = 2, nstart = 5)
print(K_cluster)
cluster::clusplot(claims_scaled, K_cluster$cluster,
       color=TRUE, shade=TRUE, labels=2, lines=1)
WSS=rep(0,5)
#Elbow Method
for(k in 1:20){
 set.seed(123)
 clust=kmeans(x=claims_scaled, centers=k, nstart=5)
 WSS[k]=clust$tot.withinss
}
plot(c(1:20), WSS, type="b", xlab="Number of Clusters",
    ylab="sum of 'Within groups sum of squares'")
K_means = kmeans(x=claims_scaled, centers = 3, nstart = 5)
print(K_means)
cluster::clusplot(claims_scaled,K_means$cluster,
              color=TRUE,shade=TRUE,labels=2,lines=1)
claims_scaled$clusters=K_means$cluster
print(claims_scaled$clusters)
head(claims_scaled)
claims_profile=aggregate(claim[,c(7,10,27,31)],list(claims_scaled$clusters),FUN="mean")
```

```
claims_profile
#================================Confusion Matrix Cross
Validation================================
modelcomparison=c("cm_log1","cm_test_cart","cm_test_rf","cm_xgb","cm_smote")
modelcomparison
table_model=data.frame(Sensitivity = NA,
                       Specificity = NA,
                       Precision = NA,
                       Recall = NA,
                       F1 = NA)
for (i in seq_along(modelcomparison)) {
 model=get(modelcomparison[i])
 a=data.frame(Sensitivity = model$byClass["Sensitivity"],
              Specificity = model$byClass["Specificity"],
              Precision = model$byClass["Precision"],
              Recall = model$byClass["Recall"],
              F1 = model$byClass["F1"])
 rownames(a)=NULL
 table_model=rbind(table_model,a)
}
table_model=table_model[-1,]
row.names(table_model)=c("LOGISTICS REGRESSION","CART","RANDOM FOREST",
                         "EXTREME GRADIENT BOOSTING","SMOTE")
table_model
#============================================================================
==========================
```