Capstone Project

Insurane Premium Default Propensity Prediction

July 2020

Author : Snehal Gawand

Table of Contents

1. Introduction

Human beings are continuously confronted to different types of dangers which threaten their lives in many aspects. Illness, earthquake, car accident, burglary, death and many others are simple examples of inevitable dangers to which human life is exposed. The occurrence of these adverse events is random in nature and frequently it is impossible to predict even the probability of their occurrence.
Insurance is way to provide guarantee to the compensation for losses due to these eventualities. The purpose of insurance is to indemnify policy holders against the occurrence of adverse events.There is a tremendous variety of events that are covered by insurance.
In this paper, we aim to model the performance of an insurance company in order to find a better insight into the stochastic nature of the problem.

2. Project Objective

Forecasting insurance claims is not a new area of expertise, actuarial sciences exist as long as the insurance business exists. All insurance companies have their internal models to forecast claims and determine insurance premiums.
The forecasting of premium payments is an important factor as premium paid by customers is the major revenue source for their successful operation. If the claims can be forecasted accurately, premiums can be adjusted accordingly, thus creating the opportunity to be one step ahead of the competitors.
Default in premium payments result in significant revenue losses and hence insurance companies are interested to know more about the customers upfront which would default premium payments.

We now present our model which simulates the performance of an insurance company. The objective is to predict the probability that a customer will default the premium payment, so that the insurance agent can proactively reach out to the policy holder to follow up for the payment of premium.

To produce accurate forecasts and insights through the methods we use, we have constructed a framework which can be divided into below subtasks:

1. Data Report
2. Initial Exploratory Data Analysis
3. Data Pre-processing
4. Exploratory Data Analysis
5. Classification
6. Forecasting
7. Model evaluation

This research work focuses on the first four subtasks.

3. Data Report

3.1 Assumptions

The dataset presented in this research is a complete dataset containing below listed multiple variables

Variable names	Description in details	Comments/Assumptions
id	Unique customer ID	Annual insurance details for each customer has been asssigned unique identification customer ID
perc_premium_paid_by_cash_credit	What % of the premium was paid by cash payments?	
age_in_days	age of the customer in days	Age of customer in days is calculated on the day on which extract is generated
Income	Income of the customer	Income is assumed as annual income of customers
Marital Status	Married/Unmarried	Married (1), unmarried (0)
Veh_owned	Number of vehicles owned (1-3)	
Count_3-6_months_late	# of times premium was paid 3-6 months late	
Count_6-12_months_late	# of times premium was paid 6-12 months late	
Count_more_than_12_months_late	# of times premium was paid more than 12 months late	
Risk_score	Risk score of customer	It is one of the primary determinant in how much monthly insurance premium the consumer will be assessed. It is assumed that 0 is lowest risk score and 100 been highest.
No_of_dep	Number of dependents in the family on the customer (1-4)	
Accomodation	Owned / Rented	Owned (1), Rented (0)
no_of_premiums_paid	# of premiums paid till date	
sourcing_channel	channel through which customer was sourced	
residence_area_type	Residence type of the customer	
premium renewal	Y variable - 0 indicates that customer has not renewed the premium and 1 indicates that customer has renewed the premium	

3.2 Environmental Setup and Data Import

The research work would be performed using R programming. The dataset is imported in R studio, and packages and libraries for data analysis are invoked.

Please refer Appendix A – Section 1 for Source Code.

3.3 Variable Identification – Inferences

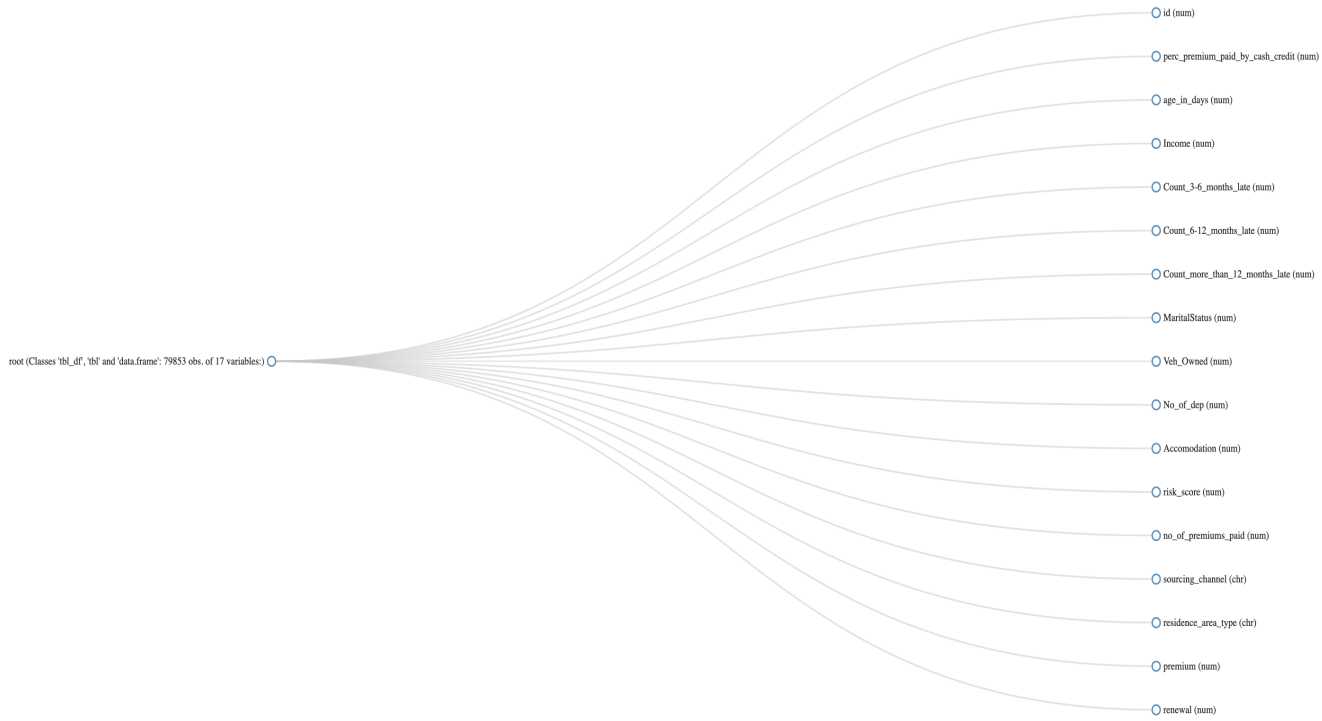


Figure 1

- Figure 1 illustrates the structure of the dataset is a data frame with 17 variables and 79853 observations.
- 2 of the variables are characters while the remaining 15 are numeric, offering hints to the type of analyses that can be performed on these variables.
- Upon closer examination, it is revealed that percentage values of premium paid are in decimals, age is in number of days and data is for both Rural and Urban population of customers.
- Furthermore, the variables for late premium payment (Count_3-6_months_late, Count_6-12_months_late, Count_more_than_12_months_late) indicate the data is collected for more than a year to perform the analysis.

Variable	Type	Values			
		Min	Max	Mean	
id	Integer	1	79853	39927	
perc_premium_paid_by_cash_credit	Integer	0	1	0.3143	
age_in_days	Integer	7640	37602	18847	
Income	Integer	24030	90262600	208847	
Marital Status	Categorical				2 levels
Veh_owned	Integer	1	3	1.998	
Count_3-6_months_late	Integer	0	13	0.2484	
Count_6-12_months_late	Integer	0	17	0.07809	
Count_more_than_12_months_late	Integer	0	11	0.05994	
Risk_score	Integer	91.90	99.89	99.07	
No_of_dep	Integer	1	4	2.503	
Accomodation	Categorical				2 levels
no_of_premiums_paid	Integer	2	60	10.86	
sourcing_channel	Categorical				5 levels
residence_area_type	Categorical				2 levels
premium renewal	categorical				2 levels

Figure 2

Figure 2 illustrates, the summary statistics of the data used in this research.

- The min, max and mean values of the variable 'Income' clearly indicates the existence of outliers in Income
- Data for customers with premium paid in cash, Income and Premium Amount is skewed towards right

Inferences:

Dataset provided has the following characteristics:

- No. of records: 79853
- 1 customer identification column, which do not hold any statistical significance
- 11 continuous numeric variables
- 4 ordinal variables with 2 levels each
- 1 ordinal variable with 5 levels
- 74855 out of 79853 have renewed their insurance, indicating 93.74% customers have renewed their premium

Please refer Appendix A – Section 1 for Source Code.

3.4 Variable modifications

We have modified the variable name 'Marital Status' to 'Marital_Status' to align it as per naming standards (adding underscore instead of space).

However, the percentage premium paid in cash has decimal values. Hence, we have converted the values to percentage format and for the ease of modelling, removed the percentage symbol.

Please refer Appendix A – Section 1 for Source Code.

4. Initial Exploratory Data Analysis

4.1 Uni-variate analysis

We will start with univariate analysis to understand the overall profiles of customers and their insurance premium payment related information.

The distribution of a single categorical variable is typically plotted with bar chart. Hence, Figure 3 illustrates the bar chart for Marital status, Accommodation, Sourcing channel, Residence area type and premium renewal.

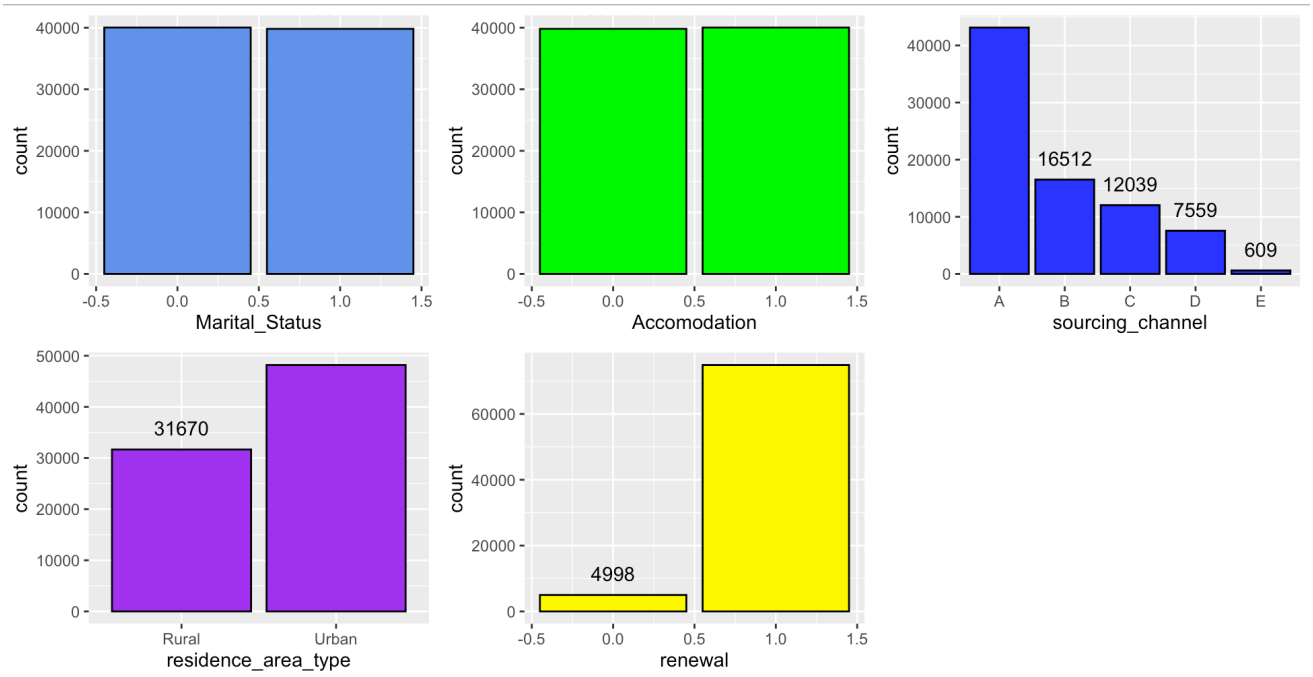


Figure 3

The distribution of numeric or continuous variables can be done via both box plots and histograms. Hence, Figure 4,5,6,7 illustrates the histograms, bar charts and box plots for the numeric variables in the dataset provided for research.

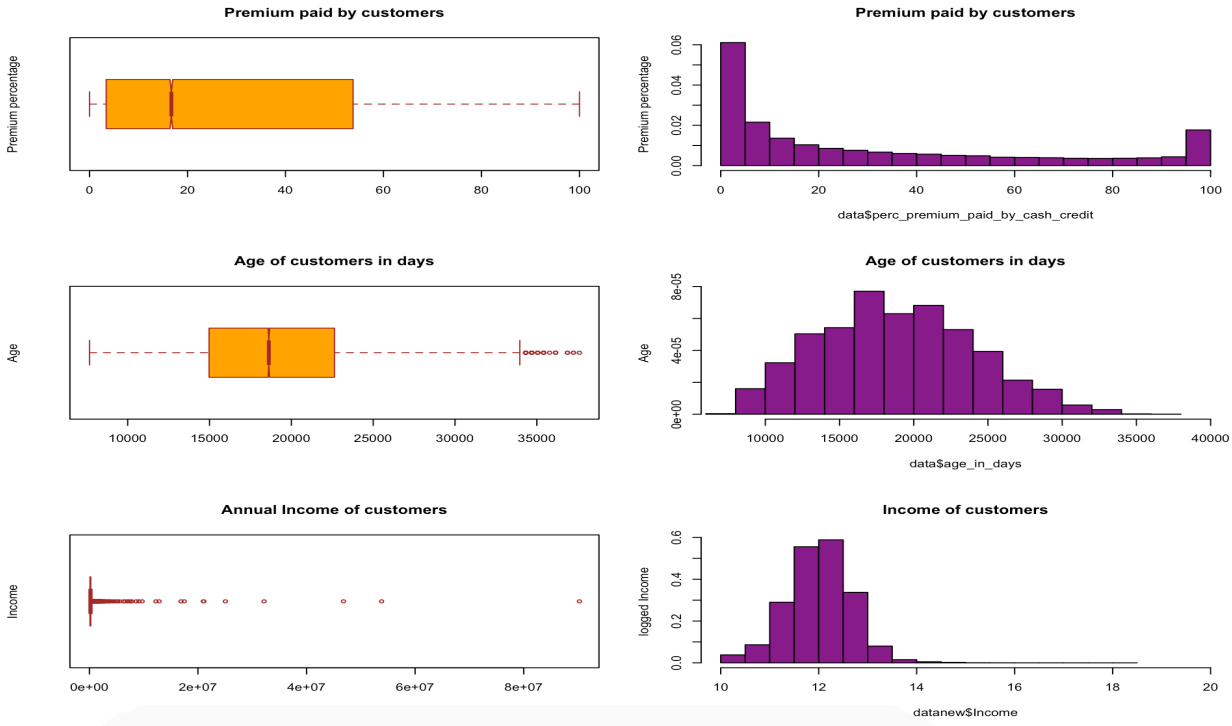


Figure 4

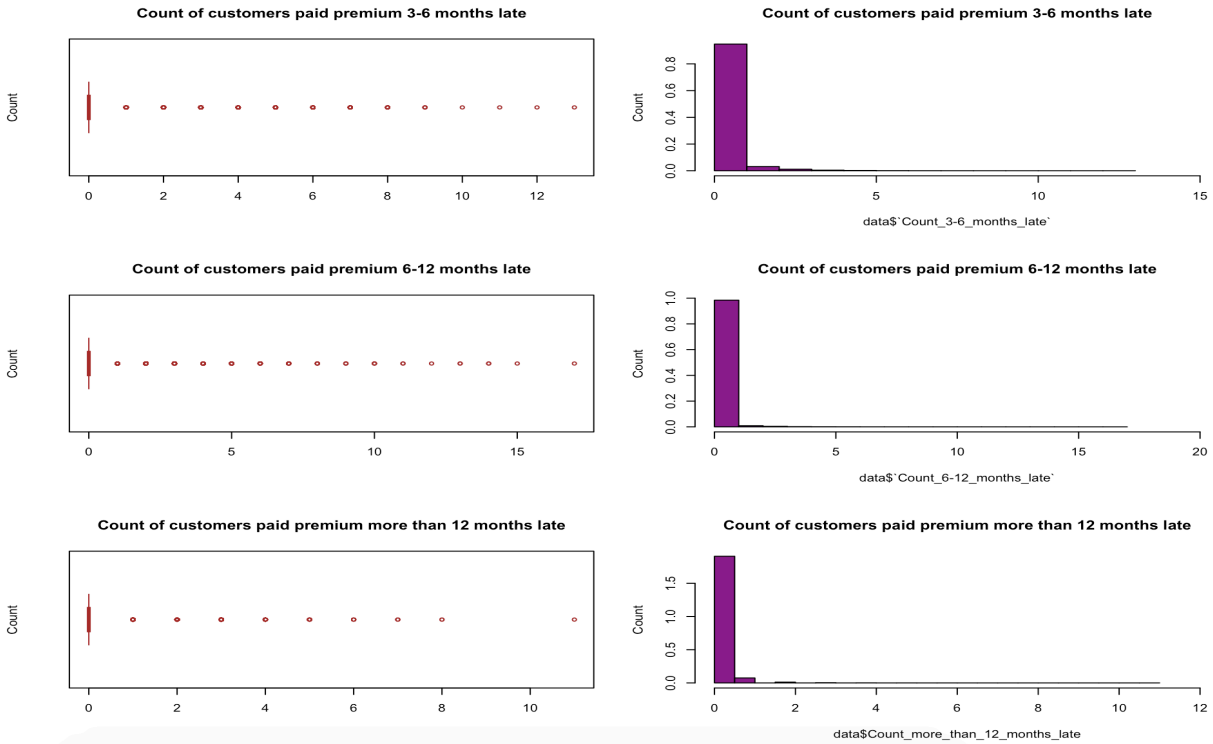


Figure 5

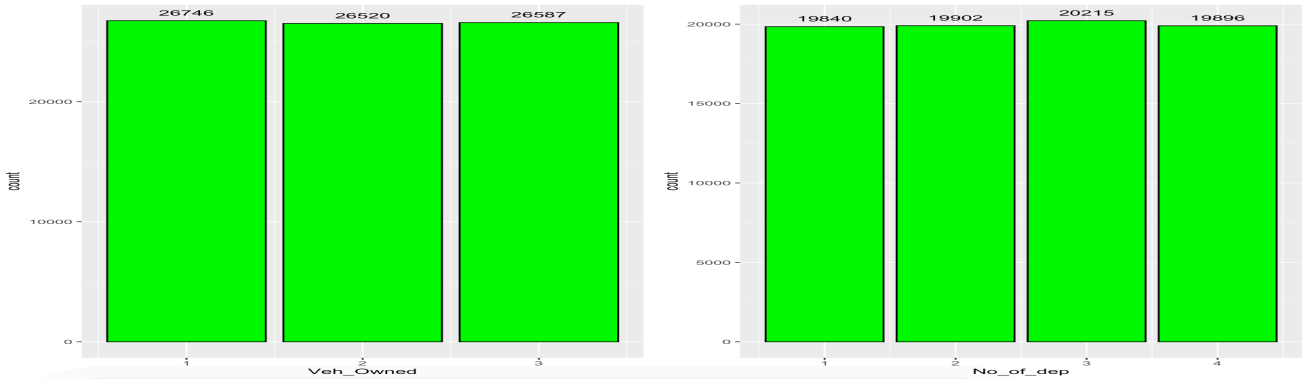


Figure 6

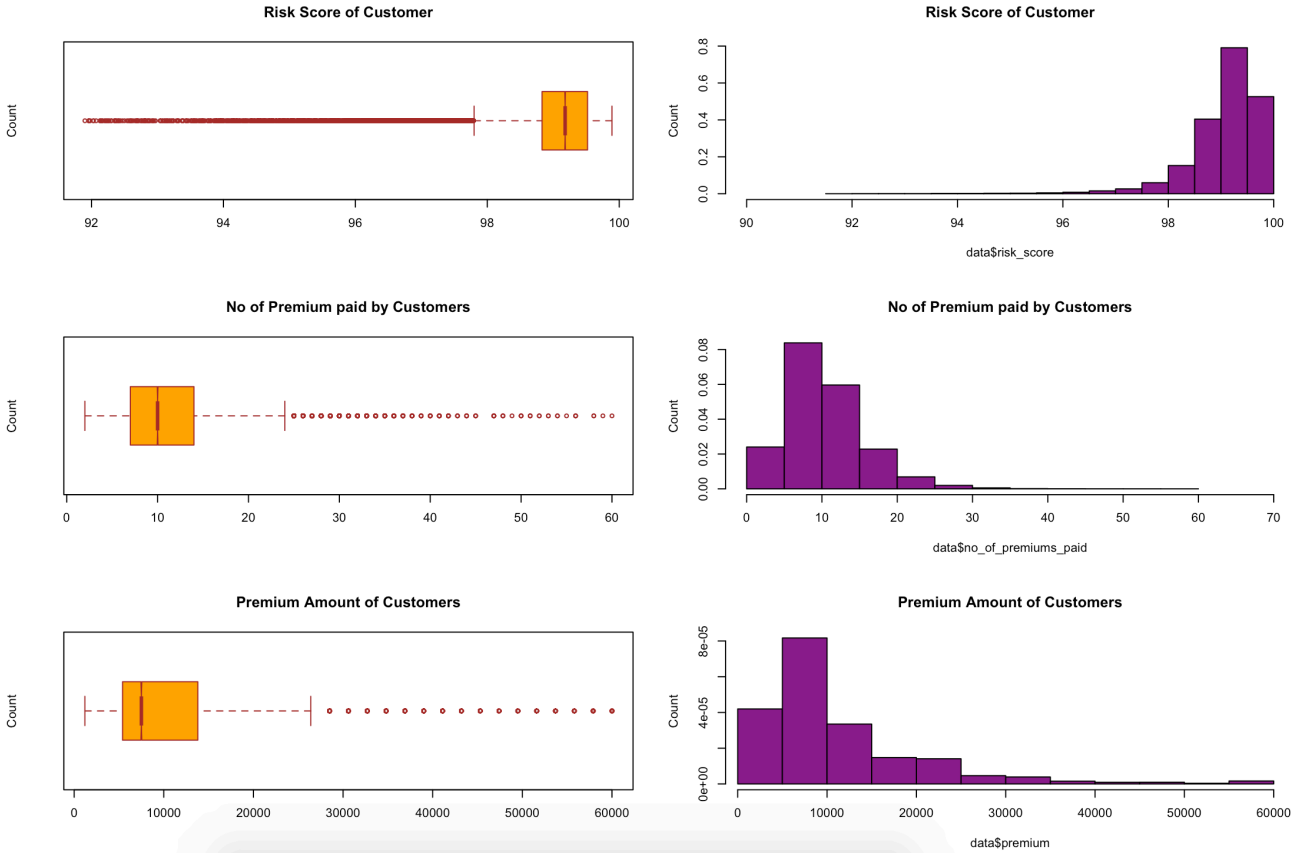


Figure 7

Please refer Appendix A- Section 2 for Source Code.

Inferences:

- Amongst the total customers having insurance policies 40032 customers are unmarried whereas 39821 customers are married. This indicates, the data is

fairly distributed amongst both types of customers, and Marital status is not an impacting factor for delay in premium payments.

- 39823 customers stay in rented accommodation whereas 40030 customers own their accommodation. This again indicates fair distribution of data, which can't be used to identify the customers who would probably delay their premium payments.
- The bar plot of Area of Residence of customer indicates around 48183 customers stay in Urban Areas, whereas 31670 in rural areas. This indicates, data is more biased towards Urban population.
- Premium Renewal Frequency plot shows, majority of the customers have renewed their premium, indicating the fact that there would be a smaller number of defaulters among the population data shared for case study.
- The sourcing channel bar plot indicates majority of the customers have their sourcing via channel A, and very few customers have sourcing channel as E.
- Majority of customers pay less than 20% of premium amount by cash
- Data is normally distributed for Age and income of customers
- Majority of customers delaying payment of premium either by 3,6,12 or more months haven't repeated the delay in payment more than 5 times
- Data for number of vehicles owned, and number of dependents is fairly distributed. This indicates, these cannot be the deciding factors for predicting defaulters
- Risk score data of customer is left skewed, with majority of customers having risk score above 99. This indicates, the risk score of customers is good.
- The number of premiums paid by customer, and the premium amount data is positively skewed.

Conclusion:

Univariate analysis depicts the facts that the data provided by insurance company is fairly distributed. However, Income, risk score, number of premiums paid, and premium amount data have outliers. Furthermore, analysis can help us to predict whether these variables with outliers can be deciding factors for payment defaulters.

4.2 Bi-variate analysis

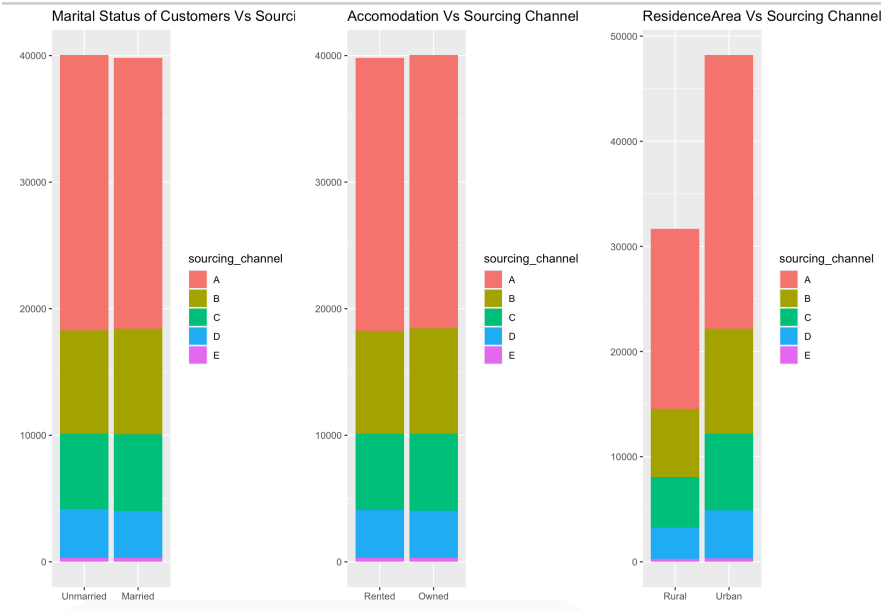


Figure 8

Figure 8 illustrates, sourcing channel preferred by most customers is A irrespective of their marital status, accommodation type or area of residence

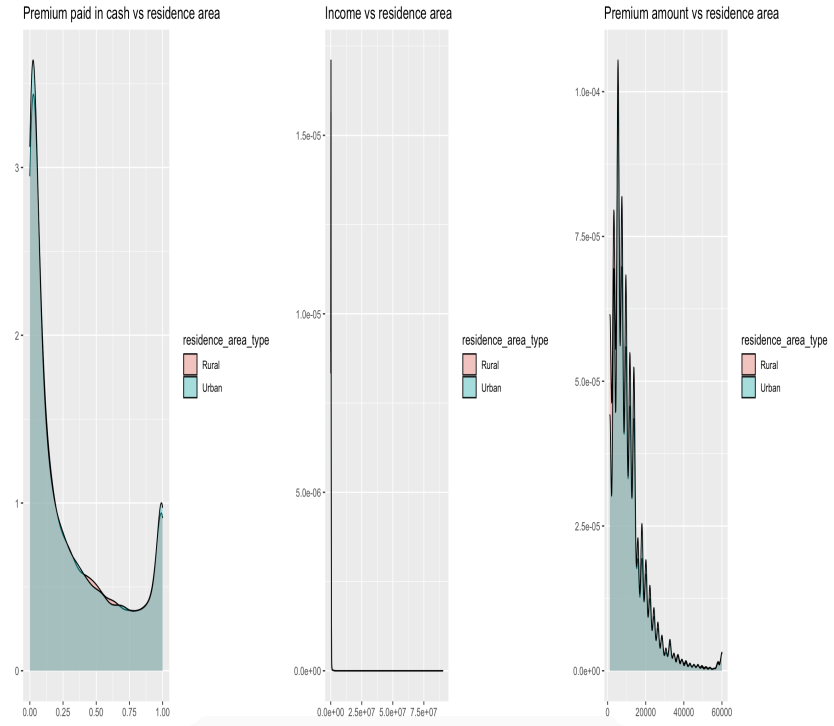


Figure 9

Figure 9 illustrates, for both rural and urban areas, the customers have similar data for premium paid in cash, Income and Premium amount.

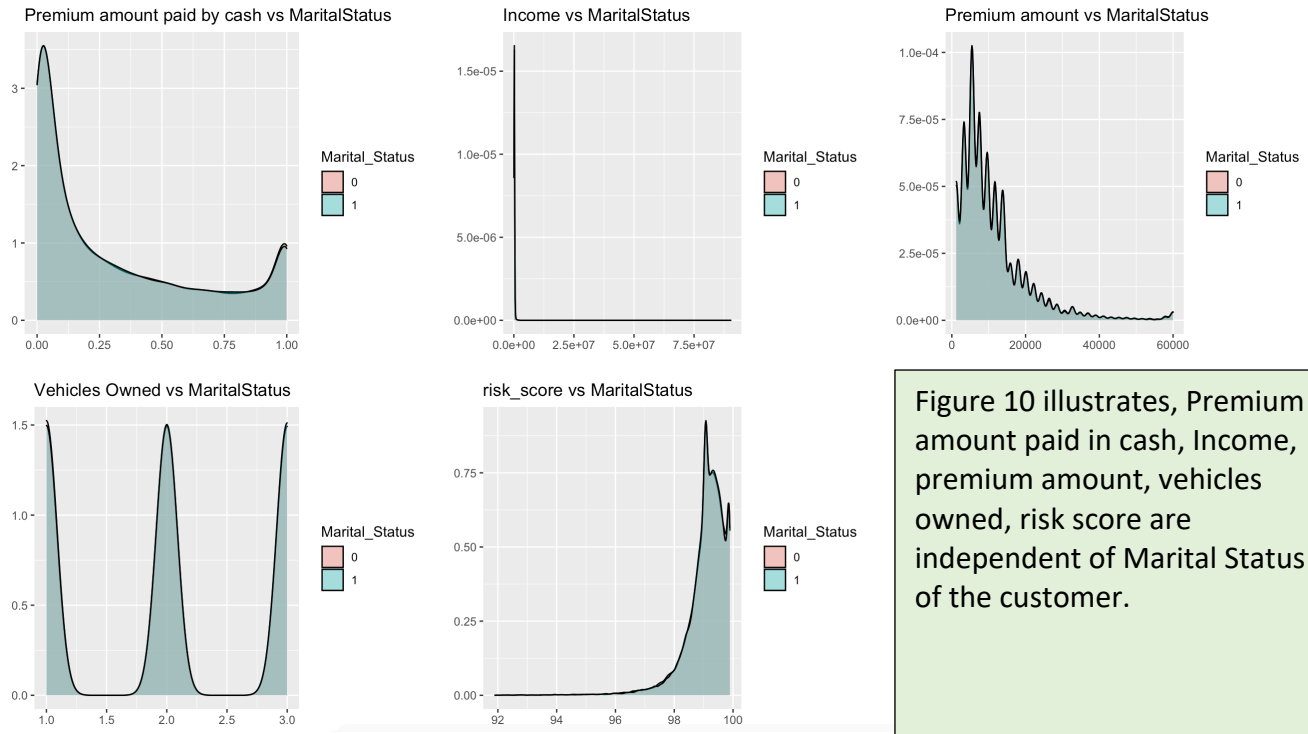


Figure 10

Figure 10 illustrates, Premium amount paid in cash, Income, premium amount, vehicles owned, risk score are independent of Marital Status of the customer.

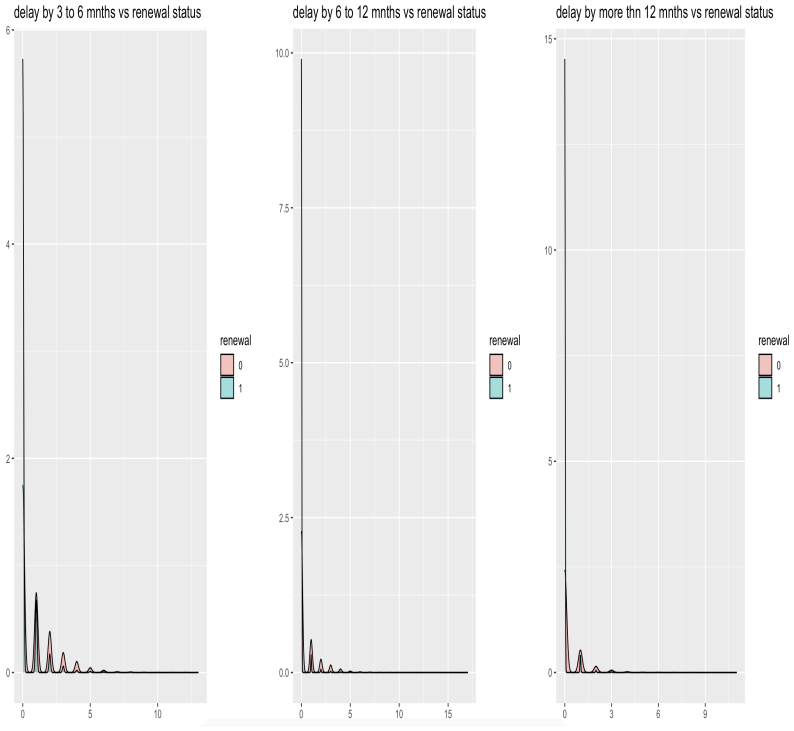


Figure 11

Figure 11 illustrates, delay in premium payment frequencies are independent of renewal status of the customer.

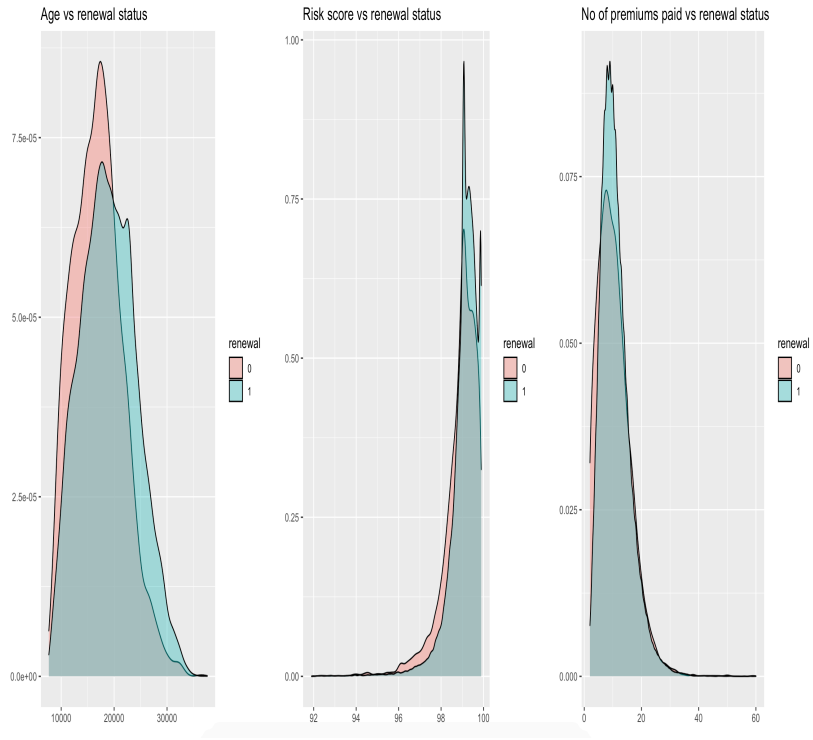


Figure 12

Figure 12 illustrates, older customers prefer to renew their insurance as compared to younger customers. Risk Score of customers who haven't renewed their insurance is moderately lower as compared to those of customers who have renewed their insurance. Also, customers who have renewed their premiums have paid a greater number of premiums

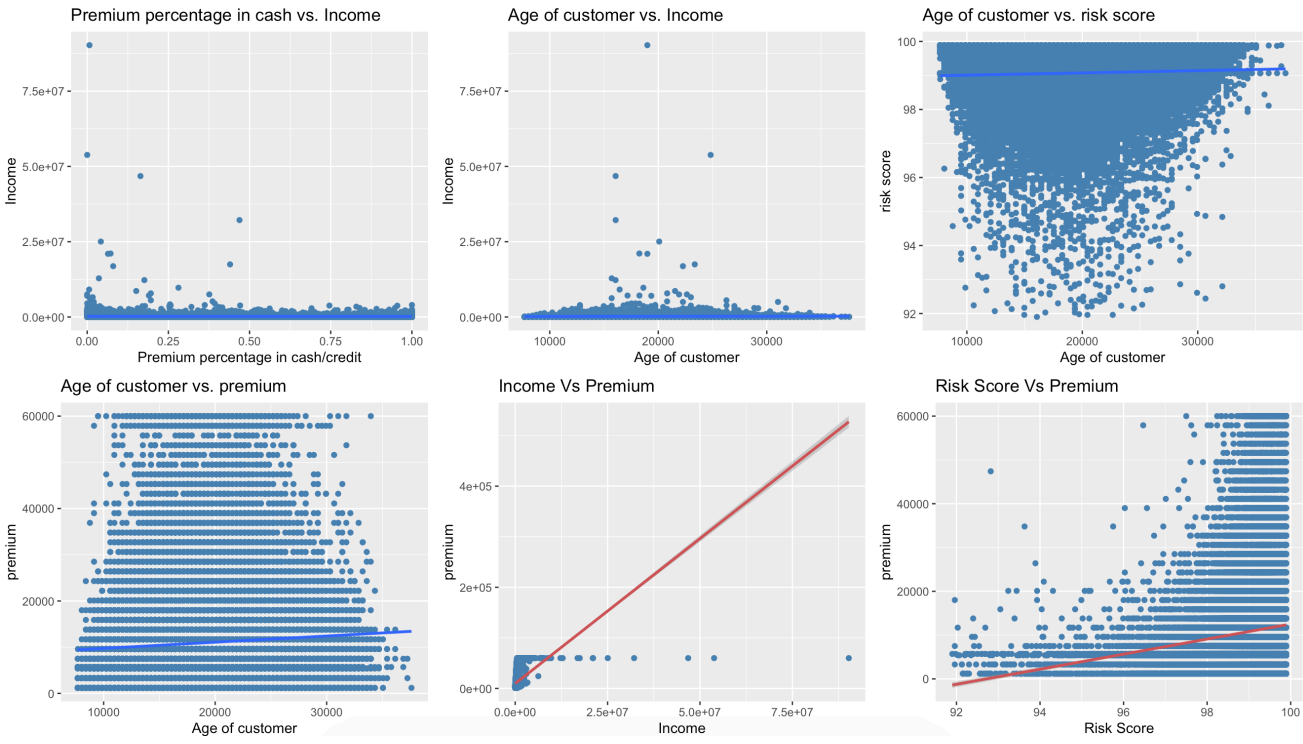


Figure 13

Inferences:

- Residence Area, Marital status can't be considered as depending factor for determining the potential customers who might delay in their premium payment.
- Most preferred sourcing channel amongst all the customer is Channel A, whereas the least preferred is Channel E.
- Older customers prefer to renew their insurance as compared to younger customers
- Customers who have renewed their premium have higher risk score and have paid moderately higher number of premiums
- Customers with higher risk score, pay higher premium amounts.
- As the income of customers increases, premium amount also rises. Indicating, a risk that customers with higher income can be potential threats for delay in premium payments

Conclusion:

Bivariate analysis depicts the facts that older customers tend to renew their insurance as compared to younger customers. Whereas, the income to age is normally distributed. This indicates, the insurance company can focus more on younger customers to check for potential delay in interest payment. Also, Income of customer and premium amount paid is directly proportional. Hence, Income can also be a significant factor to decide the potential customers which might delay their interest payment.

Please refer Appendix A – Section 2 for Source Code.

5. Data Pre-processing

In any Data Science and Machine Learning process, data pre-processing is an important step in which data gets transformed, or encoded, to bring it to such a state that now the ML algorithms can easily parse it. In other words, the features of the data can be easily interpreted by the algorithm post data pre-processing.

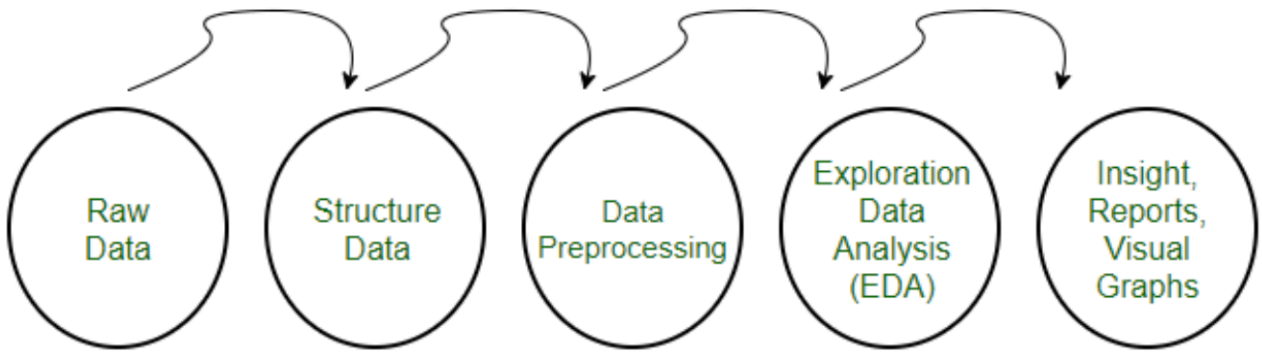


Figure 14

5.1 Variable Treatment

In this section, we will focus on modifying the variables from the dataset provided for research in order to derive better customer insights. We will also focus on feature aggregation (grouping of data). This results in reduction of memory consumption and processing time. Aggregation provides us with a high-level view of the data as the behavior of groups or aggregates is more stable than individual data objects.

- premium paid by cash credit

The observations in this variable has decimal values. For ease of use and to generate finer graphs and model, we have converted the decimals values into percentage values

- Age_in_years

The dataset currently has variable 'age in days' which depicts age of customers in number of days. Assuming the data for the insurance company was extracted as of current date, we have introduced a new variable to convert the in age into years

- Agegroup

This variable is introduced new in the dataset, and customers as per age group into different bins ranging from 20 until 110. Binning the Age might help us in deriving finer models for analysis.

Similar to Age, we have created few more feature aggregations

- Incomegroup
- perc_ premium_cash_group
- risk_score_group
- premium_amt_group

to bin the data of customers for better visibility and insights.

Please refer Appendix A – Section 3 for Source Code.

5.2 Missing value treatment

Missing values in data is a common phenomenon in real world problems. Missing values might have happened during data collection, or maybe due to some data validation rule. But regardless missing values must be taken into consideration. Simple and sometimes effective strategies are to either eliminate rows with missing values or estimate missing values and filling them with mean, median or mode value of the respective feature.

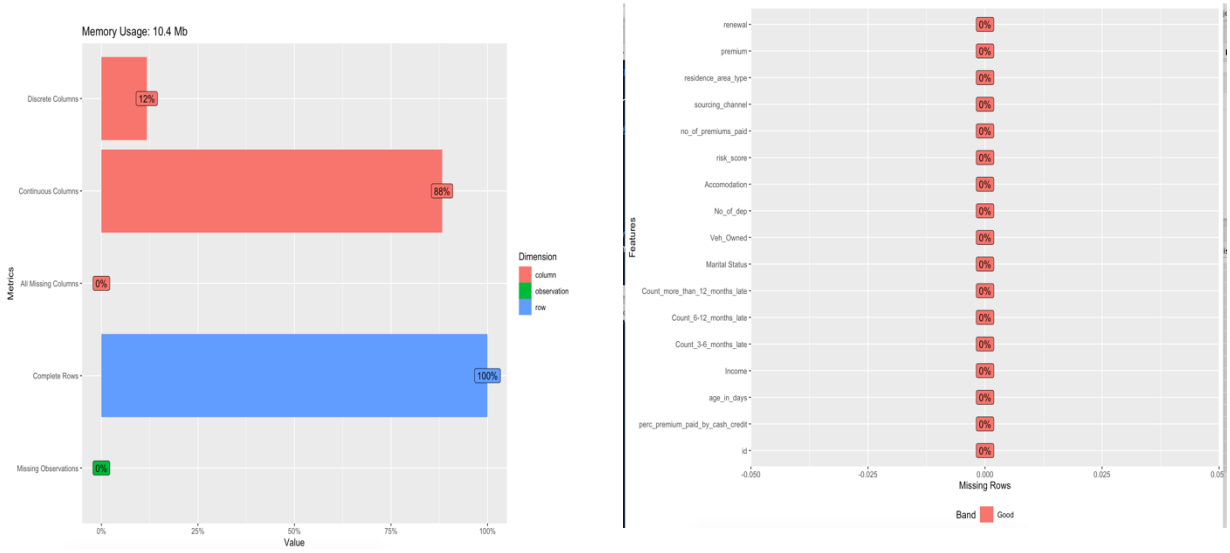


Figure 15

The above two plots indicate that all rows in the dataset are complete and that there are no missing columns and observations in the dataset. Therefore, the dataset provided for research is ready for analysis.

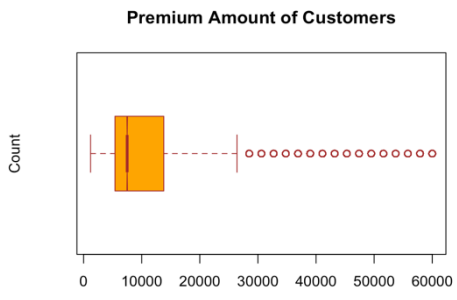
Please refer Appendix A – Section 3 for Source Code.

5.3 Outlier treatment

Outliers are values that are notably different from other data points, and they can cause problems in statistical procedures. They can lead either to miss significant findings or distort real results and violate the assumptions. Outliers increase the variability in data, which decreases statistical power. Consequently, excluding outliers can cause results to become statistically significant.

In statistical terms, one definition of outlier is any data point more than 1.5 interquartile ranges (IQRs) below the first quartile or above the third quartile. So, we would prefer to cap the outliers which are beyond the quartiles.

The summary of data generated in Section 3.3 and the histograms as well as box plots generated in Section 4.1 indicates, the variable ‘premium’ is skewed towards right.



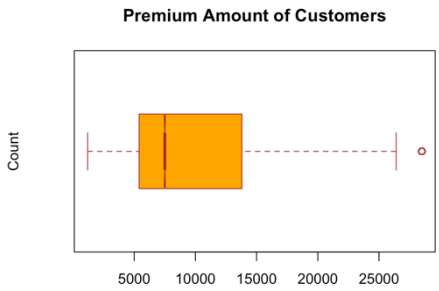


Figure 16

Figure 16 depicts the box plot of prem ıt data from research dataset, before and after outlier treatment. Here, we tried to cap the values beyond the IQR range.

Section 4.1 also shows there are large number of outliers in 'Risk Score'. The data is skewed towards left, however there are large number of records towards the right too. Hence, capping the values beyond the IQR range might result in changing the predictions of the model. Hence, we will not replace the values beyond IQR.

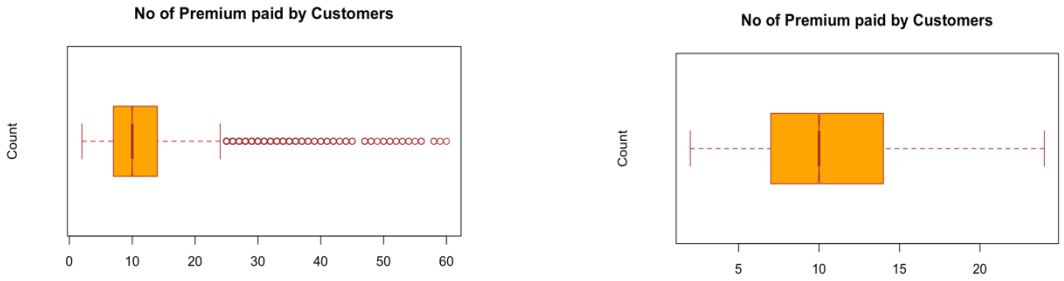


Figure 17

Figure 17 shows the box plot for 'No of premium paid' by customers, pre and post outlier treatment. Here, we have capped the values beyond IQR range.

Please refer Appendix A – Section 3 for Source Code.

6. Exploratory Data Analysis

Exploratory Data Analysis is an attitude, flexibility and graphical representation of data.
It is the practice of describing the data by means of statistical and visualization techniques to bring important aspects of that data into focus for further analysis. This is a significant step to take before diving into machine learning or statistical modeling, to make sure the data are really what they are claimed to be and that there are no obvious problems.

6.1 Insightful visualizations

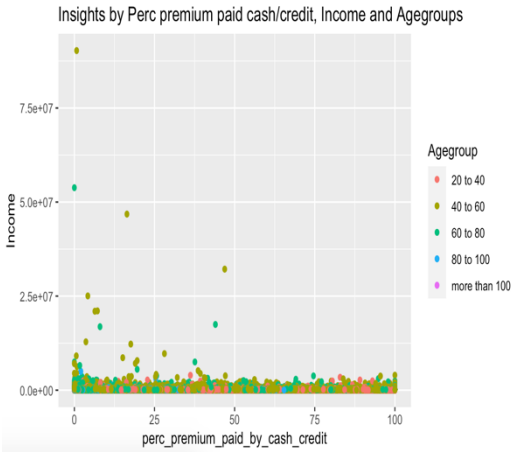
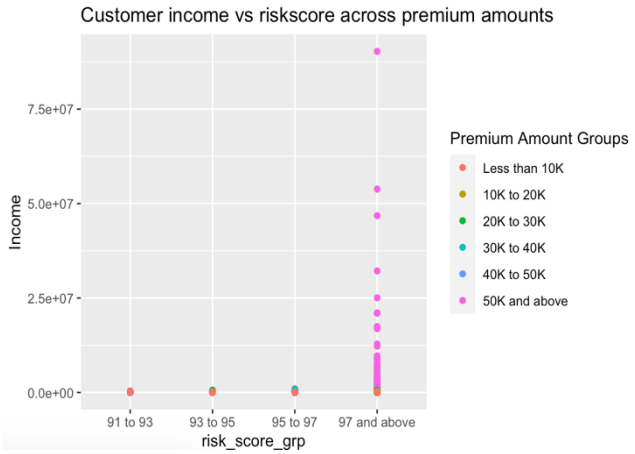


Figure 18

Figure 18 illustrates relation between percentage premium paid in cash/credit, and customer income across Age groups.
The graph shows, customers with age more than 100 years prefer paying premium via cash credit the most, as we can see the data is linear across all the percentage values.

Figure 19 helps us to know, the income of customers in rural areas is lower as compared to urban areas. But the income vs premium amount graph is the same. For customers with average income the premium amount is high, as compared to customers with comparatively lower or higher income

Figure 19



Customers with lower risk score have lower income and considerably lower premium amounts. Customers with risk score more than 97, have highest income and pay highest premium amounts.

Figure 20

Mid-aged customers have highest income as compared to younger or older customers. Also, they pay more number of premiums as compared to the other age groups.

Figure 21

Please refer Appendix A – Section 3 for Source Code.

6. 2 Relationship among variables

Building high performing machine learning algorithms depends on identifying the relationships between the variables. This helps in feature engineering as well as deciding on the machine learning algorithm.

Correlation is one of the most common statistics. Using one single value, it describes the "degree of relationship" between two variables. Correlation ranges from -1 to +1. Negative values of correlation indicate that as one variable increases the other variable decreases. Positive values of correlation indicate that as one variable increase the other variable increases as well.

6.1.1 Relationship between numeric variables

Many machine learning algorithms require that continuous variables should not be correlated with each other, a phenomenon called 'multicollinearity.' Establishing relationships between the numerical variables is a common step to detect and treat multicollinearity.

The correlogram is an important technique which can be used to identify multi-collinearity in the data.

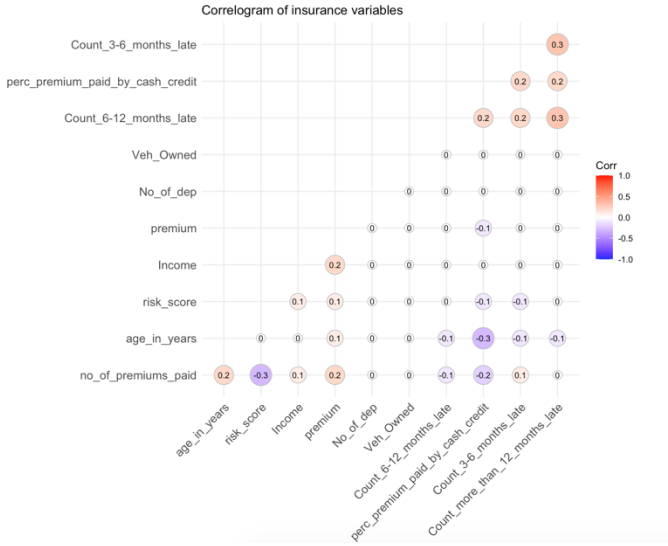


Figure 22

The matrix shows the presence of slight positive correlation between the variables

1. Income and premium
2. No of premiums paid and premium
3. No of premiums paid and age of customer

And negative correlation between:

1. No of premiums paid and risk score
2. Age of customer and perc premium paid in cash/credit

The most commonly used type of correlation is Pearson correlation, named after Karl Pearson. Pearson's r measures the linear relationships between two variables.

The Pearson correlation has two assumptions:

1. The two variables are normally distributed. We can test this assumption using
   a. A statistical test (Shapiro-Wilk)
   b. A histogram
   c. A QQ plot

2. The relationship between the two variables is linear. If this relationship is found to be curved, etc. we need to use another correlation test. We can test this assumption by examining the scatterplot between the two variables.

To calculate Pearson correlation, we can use the **cor() function**. The default method for cor() is the Pearson correlation.

- Null Hypothesis H_0: There is no correlation between the two variables: $\rho = 0$
- Alternate Hypothesis H_a: There is a nonzero correlation between the two variables: $\rho \neq$

We will verify the outcome of correlation matrix using Pearson correlation. Below are the interpretations

```
> cor(data$Income,data$Veh_Owned)
[1] 0.002601339
> cor.test(data$Income,data$Veh_Owned)

        Pearson's product-moment correlation

data:  data$Income and data$Veh_Owned
t = 0.73509, df = 79851, p-value = 0.4623
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 -0.004334653  0.009537080
sample estimates:
        cor
0.002601339
```

The p-value 0.46 is greater than 0.05. So, we fail to reject the null hypothesis that the relationship between income of customer and vehicles owned is significant.
i.e. Income and no of vehicles owned by customer are not related variables

Figure 23

```
> cor(data$Income,data$premium)
[1] 0.3028252
> cor.test(data$Income,data$premium)

        Pearson's product-moment correlation

data:  data$Income and data$premium
t = 89.788, df = 79851, p-value < 2.2e-16
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 0.2965120 0.3091118
sample estimates:
      cor
```

The p-value is less than 0.05. So, we reject the null hypothesis that the relationship between income of customer and premium amount is not significant.
i.e. As income of customer increases so does the premium amount paid.

Figure 24

Please refer Appendix A- Section 3 for Source Code.

6.1.2 Relationship between categorical variables

In the previous section, we have created plots and verified the relationships between numerical variables. It is equally important is to understand and estimate the relationship between categorical variables.
The Pearson's chi-square test of independence is used to determine whether there is an association between two or more categorical variables.
As part of this, we would first create a two-way table between the variables under study, and then run the chisq.test function.

Null Hypothesis H0: The two variables Marital_status and renewal are independent of each other.
Alternate Hypothesis H1: The two variables are related to each other.

```
> chisq.test(mar_renew,correct=FALSE)$expected

          0       1
  0 2505.603 37526.4
  1 2492.397 37328.6
> chisq.test(mar_renew,correct=FALSE)

        Pearson's Chi-squared test

data:  mar_renew
X-squared = 3.6513, df = 1, p-value = 0.05602
```

The chi-square test of correlation between Marital status and status of insurance renewal has p value approximately equal to 0.05. So we fail to reject the null hypothesis.
i.e. Marital status of customer and renewal status are not related to each other.

Figure 25

```
> chisq.test(acc_renew,correct=FALSE)$expected

          0       1
  0 2492.522 37330.48
  1 2505.478 37524.52
> chisq.test(acc_renew,correct=FALSE)

        Pearson's Chi-squared test

data:  acc_renew
X-squared = 1.3336, df = 1, p-value = 0.2482
```

The chi-square test of correlation between customer Accommodation and renewal of insurance has p value less than 0.05. So we reject the null hypothesis.
i.e. Accommodation of customer and renewal status are related to each other.

Please refer Appendix A – Section 3 for Source Code.

Conclusion:

Exploratory Data Analysis depicts
-as income of customer increases so does the premium amount
- as age of customer increases so does the number of premiums paid and the premium amount
- Customers in Owned accommodation renew their premiums more often as compared to customers in Rented accommodation
- Mid-aged customers have highest income as compared to younger or older customers. Also, they pay a greater number of premiums as compared to the other age groups
- Customers with risk score more than 97, have highest income and pay highest premium amounts.

7. Analytical Approach

We explored the data and completed the required data cleaning and pre-processing. We removed the missing values, capped the outliers, and checked for multi-colinearity.

Recommendation
We can create the regression models namely Logistic Regresion, Decision Tree model, Bagging and Boosting. Based on the validation scores of confusion matrix we can determine accuracy of model.

Further validations to check over-fitting of the model should be done before deploying the model into production. Data transformations like log transformations can also be done on the skewed variables (eg: Income of customer) and the impact to the model can be validated. Techniques of over-sampling & under sampling can be tried for the imbalanced data problem.

8. Model Building and Interpretation

8.1 Modelling Process – Validation and Interpretation

The goal of a model is to provide a simple low-dimensional summary of a dataset. In the context of this case study, we're going to use models to partition data into patterns and residuals. Strong patterns will hide subtler trends, so we'll use models to help peel back layers of structure as we explore a dataset.
There are two parts to a model

- §. First, we will define a family of models in order to express a precise, but generic pattern to capture
- §. We will generate a fitted model by finding the model from family that is closest to our data

The goal of a model is not to uncover truth, but to discover a simple approximation that is still useful. Let us begin modelling process with below mentioned steps.

8.1.1 Check class bias

Ideally, the proportion of events and non-events in the Y variable should approximately be the same.
Clearly, there is a class bias, a condition observed when the proportion of events is much greater than proportion of non-events. So, we must sample the observations in approximately equal proportions to get better models.

Please refer Appendix A – Section 4 for Source Code.

8.1.2 Create Training and Test Samples

We split the data into two chunks: training and testing set. We will build our model on the training set and evaluate its performance on the test set. This is called *the holdout-validation method* for evaluating model performance.

One way to address the problem of class bias is to draw the 0's and 1's for the training Data (development sample) in equal proportions. In doing so, we will put rest of the input Data not included for training into test Data (validation sample). As a result, the size of development sample will be smaller that validation.

Please refer Appendix A – Section 4 for Source Code.

8.1.3 Identify important variables

Selecting the most important predictor variables that explains the major part of variance of the response variable can be key to identify and build high performing models.

There are various algorithms like random forest, relative importance method, MARS method, Boruta method, etc. to identify the important variables.
Random forest can be very effective to find a set of predictors that best explains the variance in the response variable.

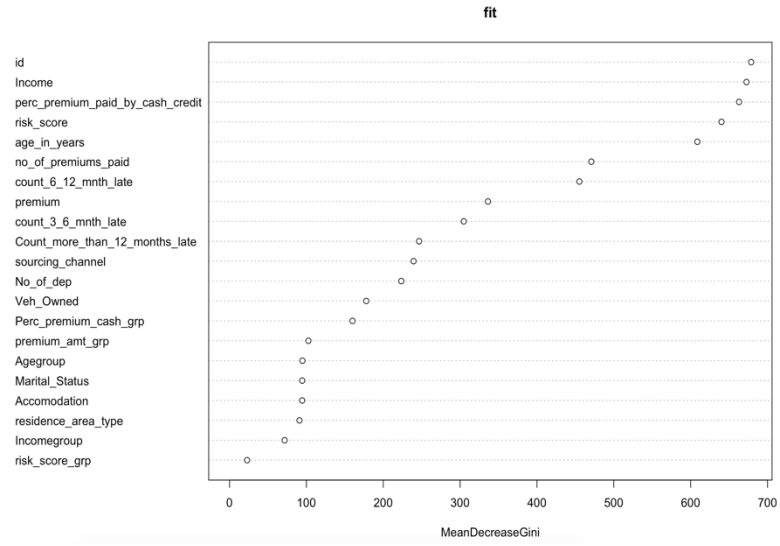


Figure 26

Figure 26 illustrates important variables identified via random forest, on basis of Gini based importance.
The top 5 important variables for predicting renewal are perc_premium_paid_by_cash_credit, risk_score, age_in_years, no_of_premiums_paid.

Please refer Appendix A – Section 4 for Source Code.

8.1.4 Model1 – Using Logistic Regression

We are provided with case study sample of 79853 customers. We need to predict the probability whether a customer will default (y) an insurance premium payment or not. As you can see, we've a categorical outcome variable, we'll use logistic regression.

Logistic regression is a method for fitting a regression curve, y = f(x), when y is a categorical variable. The typical use of this model is predicting y given a set of predictors x. The predictors can be continuous, categorical or a mix of both.
Logistic regression does not return directly the class of observations. It allows us to estimate the probability (p) of class membership. The probability will range between 0 and 1.

Important Points

- GLM does not assume a linear relationship between dependent and independent variables. However, it assumes a linear relationship between link function and independent variables in logit model.
- The dependent variable need not to be normally distributed.
- It does not use OLS (Ordinary Least Square) for parameter estimation. Instead, it uses maximum likelihood estimation (MLE).
- Errors need to be independent but not normally distributed.

We have built model on train dataset and have predicted the model on both train and test dataset.

Variables	Estimate	Std Error	z Value	Pr(> |z|)
perc_premium_paid_by_cash_credit	-1,81E+01	2,59E+00	-6.994	2.67e-12
Income	3,24E-04	2,29E-04	1.414	0.1572
count_3_6_mnth_late	-4,30E+02	1,89E+01	-22.724	< 2e-16
count_6_12_mnth_late	-6,72E+02	2,96E+01	-22.707	< 2e-16
Count_more_than_12_months_late	-6,15E+02	3,89E+01	-15.825	< 2e-16
Marital_Status	2,59E+01	3,84E+01	0.674	0.5002
Veh_Owned	1,34E+01	2,36E+01	0.567	0.5705
No_of_dep	-3,05E+01	1,72E+01	-1.771	0.0766
Accomodation	-9,21E+00	3,84E+01	-0.240	-0.240
risk_score	2,02E+02	3,67E+01	5.496	3.88e-08
no_of_premiums_paid	-2,66E+01	4,43E+00	-5.997	2.01e-09
sourcing_channelB	-3,06E+01	5,08E+01	-0.603	0.5467
sourcing_channelC	-3,66E+01	5,58E+01	-0.656	0.5117
sourcing_channelD	-1,64E+02	6,40E+01	-2.565	0.0103
sourcing_channelE	-1,94E+02	1,95E+02	-0.994	0.3201
residence_area_type	-3,73E+00	3,93E+01	-0.095	0.9245
premium	-1,32E-03	7,90E-03	0.167	0.8672
age_in_years	1,50E+01	3,81E+00	3.948	7.87e-05

Figure 27

Figure 27 shows the coefficients, their standard errors, the z-statistics and the associated p-values for the model created on the training dataset.
The logistic coefficients give the change in the log odds of the outcome for a one unit increase in the predictor variable (renewal).

For every one-unit change in perc_premium paid by cash_credit, the logs odd of renewal (versus non-renewal) decreases by **-1.81E+01**

For every one-unit change in Income, the logs odd of renewal (versus non-renewal) increases by **324E-04**

To evaluate the performance of logistic regression model, we must consider few metrics.

1. **Variable Inflation Factor for all predictors**

	GVIF	Df	GVIF^(1/(2*Df))
perc_premium_paid_by_cash_credit	23.776.151	1	4.876.079
Income	3.808.782	1	1.951.610
count_3_6_mnth_late	1.141.240	1	1.068.289
count_6_12_mnth_late	1.137.345	1	1.066.464
Count_more_than_12_months_late	1.121.219	1	1.058.876
Marital_Status	1.001.342	1	1.000.671
Veh_Owned	1.001.132	1	1.000.566
No_of_dep	1.001.275	1	1.000.637
Accomodation	1.001.244	1	1.000.622
risk_score	2.576.972	1	1.605.295
no_of_premiums_paid	1.615.725	1	1.271.112
sourcing_channel	1.146.102	4	1.017.192
residence_area_type	1.006.246	1	1.003.118
premium	13.498.103	1	3.673.977
age_in_years	6.577.312	1	2.564.627
Agegroup	7.168.456	4	1.279.170
Incomegroup	3.839.227	4	1.183.125
Perc_premium_cash_grp	24.369.022	3	1.702.706
risk_score_grp	2.187.550	3	1.139.357
premium_amt_grp	19.320.063	5	1.344.624

Figure 28

Figure 28 illustrates the multicollinearity in model based on the VIF values. As seen in the figure, all X variables have VIF below 1 except perc_premium_paid_by_cash_credit, income, premium and age of the customer. Indicating the important variables for deciding the customers who will default the renewal of insurance premiums.

2. **Misclassification error:**

Misclassification error is the percentage mismatch of predicted vs actuals, irrespective of 1's or 0's. The lower the misclassification error, the better is your model.
Misclassification error --> 0.0599

3. **ROC Curve:**

Receiver Operating Characteristic (ROC) summarizes the model's performance by evaluating the trade offs between true positive rate (sensitivity) and false positive rate(1- specificity). For plotting ROC, it is advisable to assume p > 0.5 since we are more concerned about success rate. ROC summarizes the predictive power for all possible values of p > 0.5. The area under curve (AUC), referred to as index of accuracy(A) or concordance index, is a perfect performance metric for ROC curve. Higher the area under curve, better the prediction power of the model. Below is a sample ROC curve. The ROC of a perfect predictive model has TP equals 1 and FP equals 0. This curve will touch the top left corner of the graph.

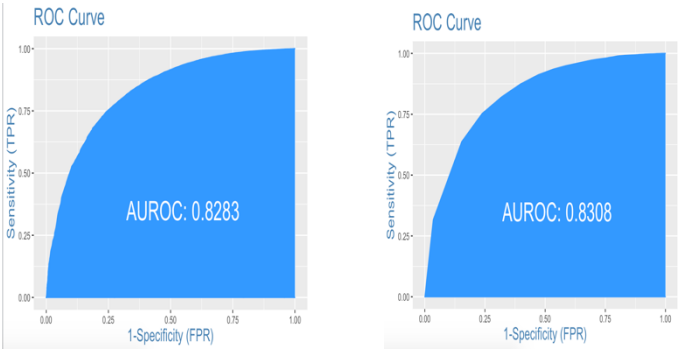


Figure 29 – ROC for train dataset (left) and test dataset (right)

The train and test ROC curve in Figure 29 illustrate a good model as

- the curve is rising steeply indicating TPR increases faster than FPR
- Area under the ROC curve is greater, indicating better predictive ability of model

Sensitivity (or True Positive Rate) is the percentage of 1's (actuals) correctly predicted by the model, while, specificity is the percentage of 0's (actuals) correctly predicted. Specificity can also be calculated as 1 − False Positive Rate.

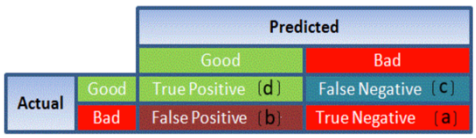
	THRESHOLD	SPECIFICITY	SENSITIVITY
TEST DATASET	2.787.837	0.7590051	0.7467079
TRAIN DATASET	0.932513	0.7373333	0.780781

Figure 30 shows the sensitivity and specificity values calculated for both test and training dataset. So, a truth detection rate of approx. 74% on test data is good

Figure 30

5. **Confusion Matrix**

A confusion matrix is a table that is often used to describe the performance of a classification model (or "classifier") on a set of test data for which the true values are known. The confusion matrix itself is relatively simple to understand, but the related terminology can be confusing.



	0	1
0	251	186
1	1249	22271

Figure 31

The above figure displays confusion matrix created for the model on testing dataset. It illustrates

- Out of total predictions, prediction for 251 customers will renew their premium is true
- We predicted 22271 customers will not renew premium and its true
- We predicted 1249 customers will renew their premium and its false
- We predicted 186 customers will not renew their premium and its false
- Accuracy of model is (251+22271)/(251+186+1249+22271) = 94.01%

Conclusion:

Logistic regression built for the model depicts

- *perc_premium_paid_by_cash_credit, income, premium and age of the customer are the important variables for predicting the insurance renewal status of the customers provided in the case study*
- *The misclassification ratio of the model which indicates the mismatch in predicted vs actuals is very low, indicating the model is good fit. However, we can try other models and compare this value*
- *Accuracy of model is 94.01%*
- *The model performance on basis of ROC is also good as the ROC value is more indicating better predictive ability of the model*

Please refer Appendix A – Section 5 for Source Code.

8.1.5 Model2 – Using Decision Tree

A Decision Tree is a supervised learning predictive model that uses a set of binary rules to calculate a target value.It is used for either classification (categorical target variable) or regression (continuous target variable). Hence, it is also known as CART (Classification & Regression Trees).

Decision trees have three main parts:
Root Node: The node that performs the first split.
Terminal Nodes/Leaves: Nodes that predict the outcome.
Branches: arrows connecting nodes, showing the flow from question to answer.

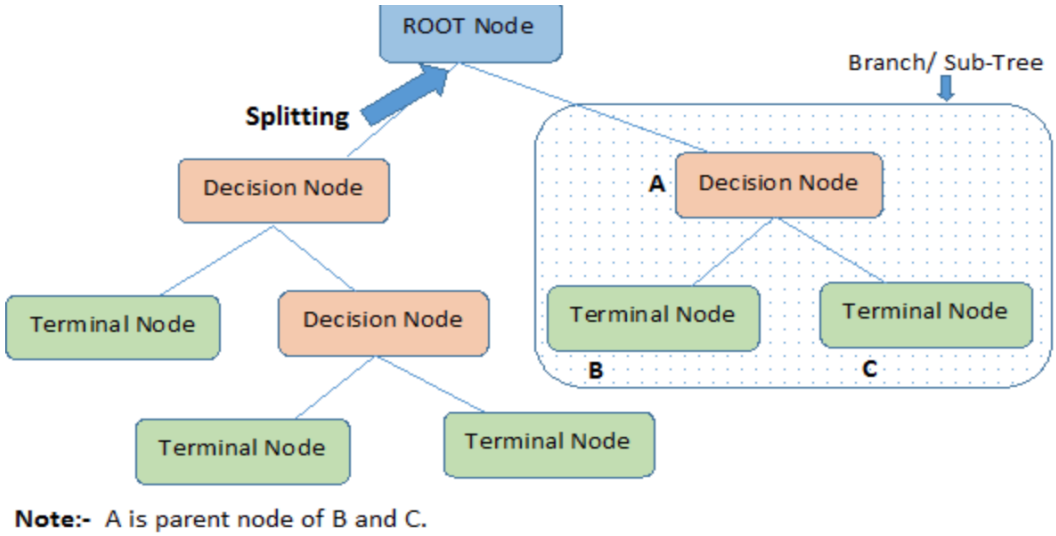


Figure 32

The algorithm of the decision tree models works by repeatedly partitioning the data into multiple sub-spaces so that the outcomes in each final sub-space is as homogeneous as possible. This approach is technically called recursive partitioning.
The produced result consists of a set of rules used for predicting the outcome variable, which can be either:

- a continuous variable, for regression trees
- a categorical variable, for classification trees

The decision rules generated by the CART (Classification & Regression Trees) predictive model are generally visualized as a binary tree. CART algorithms works by repeatedly finding the best predictor variable to split the data into two subsets. The subsets partition the target outcome better than before the split. Pruning is a technique associated with classification and regression trees.

We have built the CART model on the training dataset which consists of 55896 observations. And the below figure illustrates over-plotting. Overfitting happens when a model memorizes its training data so well that it is learning noise on top of the signal.

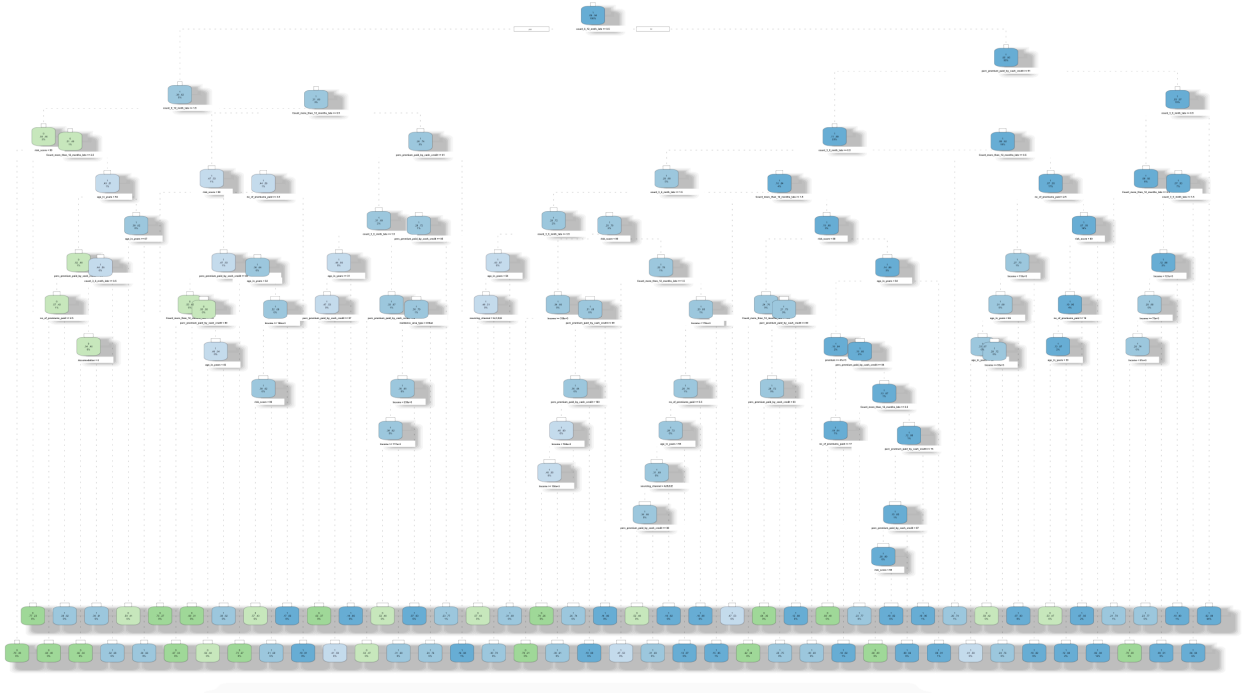


Figure 33

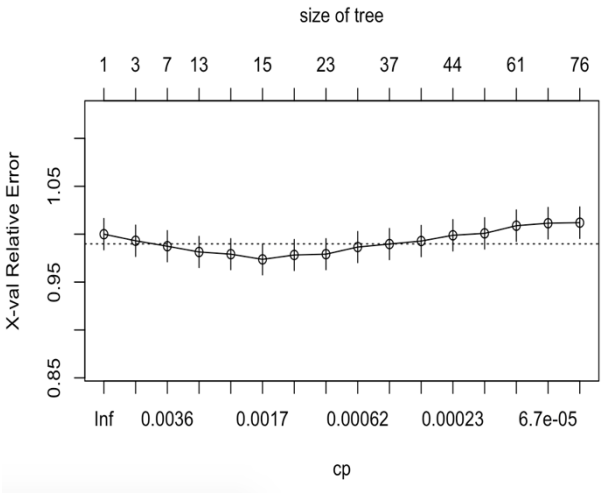


Figure 34

Figure 34 shows the graph of the complexity parameter of the CART model plotted in Figure 33.
This function provides the optimal pruning based on the cp value (here its 0.017)

Hence, we will prune the tree for cp=0.017 to avoid overfitting of the data

Pruning

As the name implies, pruning involves cutting back the tree. After a tree has been built (and in the absence of early stopping discussed below) it may be overfitted. The CART algorithm will repeatedly partition data into smaller and smaller subsets until those final subsets are homogeneous in terms of the outcome variable. In practice this often means that the final subsets (known as the leaves of the tree) each consist of only one or a few data points. The tree has learned the data exactly, but a new data point that differs very slightly might not be predicted well.

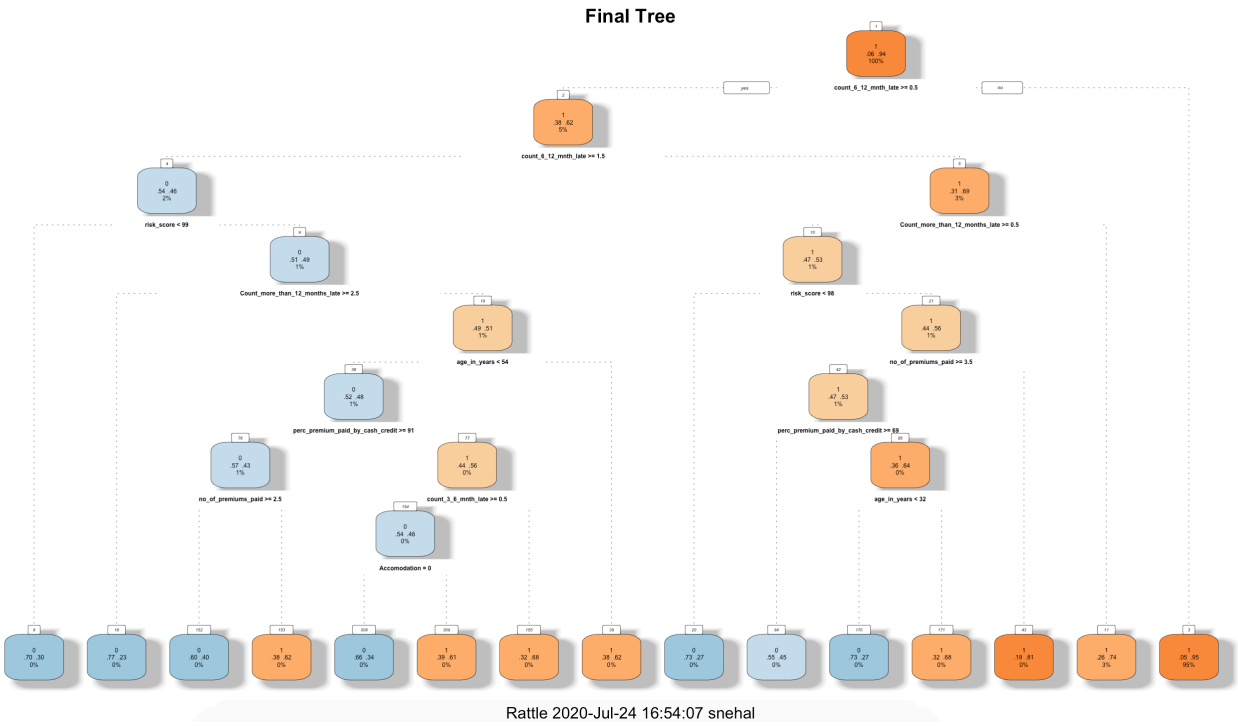
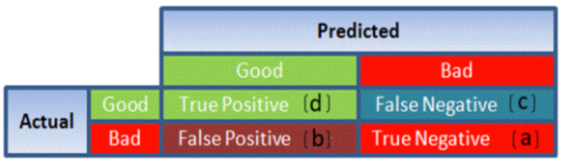


Figure 35

To evaluate the performance of CART model, we must consider few metrics.

1. **Confusion matrix**



	0	1
0	499	2999
1	298	52100

Figure 36

The above figure displays confusion matrix created for the model on training dataset. It illustrates

- Out of total predictions, prediction for 499 customers will renew their premium is true
- We predicted 52100 customers will not renew premium and its true
- We predicted 298 customers will renew their premium and its false
- We predicted 2999 customers will not renew their premium and its false

The figure 37 shows performance plot of the training dataset

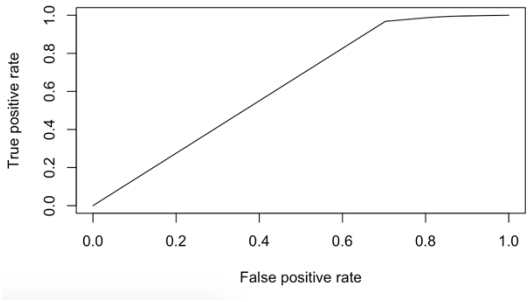


Figure 37

2. **Kolmogorov-Smirnov Test**

The test compares a known hypothetical probability distribution (e.g. the normal distribution) to the distribution generated by data — the empirical distribution function.

The KS value for training dataset is 0.2648119, indicating model needs to be improved

3. **AUC and GINI**

Gini coefficient applies to binary classification and requires a classifier that can in some way rank examples according to the likelihood of being in a positive class.

The auc value for training dataset is 63.4%, indicating model needs to be improved
Gini coefficient is 1.6%, indicating model is not fit for use

4. **Classification error rate**

The classification error rate is 1 – accuracy i.e 0.59%

Conclusion:

CART decision tree model built for the data depicts

- *Accuracy of model is 94.10%*
- *The model performance on basis of ROC, AUC, KS and gini coefficient values is not good. Hence, we can improve the model by Bagging the model*

Please refer Appendix A – Section 6 for Source Code.

8.3 Ensemble Modelling

Ensemble methods are advanced techniques often used to solve complex machine learning problems. In simple terms, an ensemble method is a process where different and independent models (also referred to as the "weak learners") are combined to produce an outcome. The hypothesis is that combining multiple models can produce better results by decreasing generalization error.

Three of the most popular methods for ensemble modeling are bagging, boosting, and stacking. The goal of ensemble modeling is to improve performance over a baseline model by combining multiple models. So, we will set the baseline performance measure by starting with one algorithm. In our case, we will build a logistic regression algorithm.

Bagging

Bagging, or bootstrap aggregation, is an ensemble method that involves training the same algorithm many times by using different subsets sampled from the training data. The final output prediction is then averaged across the predictions of all the sub-models. The two most popular bagging ensemble techniques are Bagged Decision Trees and Random Forest.

Bagged Decision Tree

We have performed the bagging algorithm on the training dataset and predict the bagged model on the testing dataset.

	0	1
0	1452	48
1	0	22457

The confusion matrix illustrates the bagged decision tree model is 99.79% accurate (higher than the logistic regression model)

Boosting

In boosting, multiple models are trained sequentially and each model learns from the errors of its predecessors. In this guide, we will implement a gradient boosting algorithm.

Stochastic Gradient Boosting

The boosting algorithm focuses on classification problems and aims to convert a set of weak classifiers into a strong one.
We follow the same steps as above, with exception that while training the algorithm, we set method="gbm" to specify that a gradient boosting model is to be built. The accuracy of the model is 92.08 percent, which is lower than the baseline accuracy.

8.4 Model Comparison and Interpretation from the best model

	Model Comparison	
	Miscallsification	ROC
Logistic Regression	0.059	0.823
Decision Tree	0.059	0.634

	Accuracy
Logistic Regression	94.01%
Decision Tree	94.10%
Bagged Decision Tree	99.79%
Stochastic Gradient Boosting	92.08%

Figure 38

The above figure 38 shows the model comparison details. The misclassification error rate for both the logistic regression model and the decision tree are almost the same. However, there is significant difference between the ROC value indicating logistic regression model is better than the decision tree model.

However, post performing ensemble modelling techniques on the training dataset and testing dataset, we can clearly see the Bagging of decision tree has provided the highest value of accuracy. Indicating, the best model is created via the bagged decision tree followed by the logistic regression model.

In general, LR and CART have identified very similar sets of significant markers. While LR model has focused more on the relative statistical significance of the assessed markers, CART results emphasize the absolute effects; results come as observed risk groups. Thus, the complementary use of both techniques seems to be a promising approach to perform the difficult task of analyzing and interpreting the results of insurance default studies.

Considering the Decision Tree model performs nearly as well as the Logistic Regression, it presents a useful alternative for insurance default propensity prediction.

Decision Trees are advantageous for predictive modeling due to:
- *Implicit variable screening and selection – the top nodes of the tree are the most important variables in the dataset!*
- *Less data prep – data does not need to be normalized, and decision trees are less sensitive to missing data and outliers*
- *Decision trees do not require assumptions of linearity*
- *Decision tree output is graphical and easy to explain – decision based on cut points*

Please refer Appendix A – Section 7 for Source Code.

8.5 Business Insights

Post performing LR, CART and ensemble modelling techniques on the insurance data provided for analysis, we conclude our case study with the below listed observations

- *perc_premium_paid_by_cash_credit, income, premium and age of the customer are the important variables for predicting the insurance renewal status of the customers*
- *Delay in premium payment frequencies is independent of status of renewal of insurance*
- *Older customers prefer to renew their insurance as compared to younger customers*

Hence, business should focus on the younger customers with lower income belonging to both Urban as well as rural areas. And the insurance agents should be suggested to proactively reach out to these customers for following up for their insurance premium payments.

9. Appendix A

9.1 Section 1

```
##install packages and invoke libraries

install.packages("ggplot2")
install.packages("formattable")

library(readxl)
library(ggplot2)
library(formattable)
library(DataExplorer)
library(dplyr)
library(gridExtra)
library(scales)
library(data.table)
library(car)
library(ggcorrplot)

#setup working directory
setwd('~/Documents/Capstone Project - Insurance Premium Default Propensity Prediction')
getwd()

#read input premium dataset
```

```
data <- read_excel("premium.xlsx")

plot_str(data)
summary(data)

#convert column names for ease of use
data = data %>% rename(Marital_Status = `Marital Status`)
```

9.2 Section 2

```
###-----------------------------------------------------------------------------##
#                          Univariate Analysis                                  #
##-----------------------------------------------------------------------------##

##Univariate analysis on categorical data


p1 = ggplot(data=data, aes(x=Marital_Status)) +
  geom_bar(fill = "cornflowerblue",
           color="black")  +
  geom_text(stat='count', aes(label=..count..), vjust=-1)

p2 = ggplot(data=data, aes(x=Accomodation)) +
  geom_bar(fill = "green",
           color="black")  +
  geom_text(stat='count', aes(label=..count..), vjust=-1)

p3 = ggplot(data=data, aes(x=sourcing_channel)) +
  geom_bar(fill = "blue",
           color="black")  +
  geom_text(stat='count', aes(label=..count..), vjust=-1)

p4 = ggplot(data=data, aes(x=residence_area_type)) +
  geom_bar(fill = "purple",
           color="black")  +
  geom_text(stat='count', aes(label=..count..), vjust=-1)

p5 = ggplot(data=data, aes(x=renewal)) +
  geom_bar(fill = "yellow",
           color="black")  +
  geom_text(stat='count', aes(label=..count..), vjust=-1)

grid.arrange(p1,p2,p3,p4,p5, nrow=4,ncol=3)

plot_histogram(data)

##Univariate analysis on numeric/continuous data

par(mfrow = c(3,2))
```

```
boxplot(data$perc_premium_paid_by_cash_credit,
        main = "Premium paid by customers",
        ylab = "Premium percentage",
        col = "orange",
        border = "brown",
        horizontal = TRUE,
        notch = TRUE)

hist(data$perc_premium_paid_by_cash_credit,
        main = "Premium paid by customers",
        ylab = "Premium percentage",
        xlim=c(0,100),
        col = "darkmagenta",
        freq = FALSE)

boxplot(data$age_in_days,
        main = "Age of customers in days",
        ylab = "Age",
        col = "orange",
        border = "brown",
        horizontal = TRUE,
        notch = TRUE)

hist(data$age_in_days,
     main = "Age of customers in days",
     ylab = "Age",
     xlim=c(7000,40000),
     col = "darkmagenta",
     freq = FALSE)

boxplot(data$"Income",
        main = "Annual Income of customers",
        ylab = "Income",
        col = "orange",
        border = "brown",
        horizontal = TRUE,
        notch = TRUE)

logincome=log(data$Income)
datanew <- data
datanew$Income <- logincome


hist(datanew$Income,
     main = "Income of customers",
     ylab = "logged Income",
     xlim=c(10,20),
     col = "darkmagenta",
     freq = FALSE)
```

```
par(mfrow = c(3,2))
boxplot(data$`Count_3-6_months_late`,
        main = "Count of customers paid premium 3-6 months late",
        ylab = "Count",
        col = "orange",
        border = "brown",
        horizontal = TRUE,
        notch = TRUE)

hist(data$`Count_3-6_months_late`,
     main = "Count of customers paid premium 3-6 months late",
     ylab = "Count",
     xlim=c(0,15),
     col = "darkmagenta",
     freq = FALSE)

boxplot(data$`Count_6-12_months_late`,
        main = "Count of customers paid premium 6-12 months late",
        ylab = "Count",
        col = "orange",
        border = "brown",
        horizontal = TRUE,
        notch = TRUE)

hist(data$`Count_6-12_months_late`,
     main = "Count of customers paid premium 6-12 months late",
     ylab = "Count",
     xlim=c(0,20),
     col = "darkmagenta",
     freq = FALSE)

boxplot(data$`Count_more_than_12_months_late`,
        main = "Count of customers paid premium more than 12 months late",
        ylab = "Count",
        col = "orange",
        border = "brown",
        horizontal = TRUE,
        notch = TRUE)

hist(data$`Count_more_than_12_months_late`,
     main = "Count of customers paid premium more than 12 months late",
     ylab = "Count",
     xlim=c(0,12),
     col = "darkmagenta",
     freq = FALSE)

p1 = ggplot(data=data, aes(x=Veh_Owned)) +
  geom_bar(fill = "green",
           color="black")  +
```

```
  geom_text(stat='count', aes(label=..count..), vjust=-1)

p2 = ggplot(data=data, aes(x=No_of_dep)) +
  geom_bar(fill = "green",
           color="black")  +
  geom_text(stat='count', aes(label=..count..), vjust=-1)

par(mfrow = c(3,2))
boxplot(data$`risk_score`,
        main = "Risk Score of Customer",
        ylab = "Count",
        col = "orange",
        border = "brown",
        horizontal = TRUE,
        notch = TRUE)

hist(data$`risk_score`,
     main = "Risk Score of Customer",
     ylab = "Count",
     xlim=c(90.0,100.0),
     col = "darkmagenta",
     freq = FALSE)

boxplot(data$`no_of_premiums_paid`,
        main = "No of Premium paid by Customers",
        ylab = "Count",
        col = "orange",
        border = "brown",
        horizontal = TRUE,
        notch = TRUE)

hist(data$`no_of_premiums_paid`,
     main = "No of Premium paid by Customers",
     ylab = "Count",
     xlim=c(0,70),
     col = "darkmagenta",
     freq = FALSE)

boxplot(data$premium,
        main = "Premium Amount of Customers",
        ylab = "Count",
        col = "orange",
        border = "brown",
        horizontal = TRUE,
        notch = TRUE)

hist(data$premium,
     main = "Premium Amount of Customers",
     ylab = "Count",
     xlim=c(1000,60000),
```

```
    col = "darkmagenta",
    freq = FALSE)


##-----------------------------------------------------------------------------##
#                        Biivariate Analysis                                  #
##-----------------------------------------------------------------------------##

##categorical vs categorical

p1 = ggplot(data,
       aes(x = factor(Marital_Status,labels=c("Unmarried","Married")),
           fill = sourcing_channel)) +
  geom_bar(position = "stack") +
  labs(title = "Marital Status of Customers Vs Sourcing Channel",
       x = "",
       y = "")

p2 = ggplot(data,
       aes(x = factor(Accomodation,labels=c("Rented","Owned")),
           fill = sourcing_channel)) +
  geom_bar(position = "stack") +
  labs(title = "Accomodation Vs Sourcing Channel",
       x = "",
       y = "")

p3 = ggplot(data,
       aes(x = factor(residence_area_type,labels=c("Rural","Urban")),
           fill = sourcing_channel)) +
  geom_bar(position = "stack") +
  labs(title = "ResidenceArea Vs Sourcing Channel",
       x = "",
       y = "")

grid.arrange(p1,p2,p3, nrow=1,ncol=3)

##categorical vs quantitative
p1 = ggplot(data,
       aes(x = perc_premium_paid_by_cash_credit,
           fill = residence_area_type)) +
  geom_density(alpha = 0.4) +
  labs(title = "Premium paid in cash vs residence area",
       x = "",
       y = "")

p2 = ggplot(data,
       aes(x = Income,
           fill = residence_area_type)) +
  geom_density(alpha = 0.4) +
  labs(title = "Income vs residence area",x = "",
```

```
      y = "")

p3 = ggplot(data,
      aes(x = premium,
          fill = residence_area_type)) +
  geom_density(alpha = 0.4) +
  labs(title = "Premium amount vs residence area",x = "",
      y = "")
grid.arrange(p1,p2,p3, nrow=1,ncol=3)

data$Marital_Status <- as.factor(data$Marital_Status)

##
p1 = ggplot(data,
           aes(x = perc_premium_paid_by_cash_credit,
               fill = Marital_Status)) +
  geom_density(alpha = 0.4) +
  labs(title = "Premium amount paid by cash vs MaritalStatus",x = "",
      y = "")

p2 = ggplot(data,
           aes(x = Income,
               fill = Marital_Status)) +
  geom_density(alpha = 0.4) +
  labs(title = "Income vs MaritalStatus",x = "",
      y = "")

p3 = ggplot(data,
      aes(x = premium,
          fill = Marital_Status)) +
  geom_density(alpha = 0.4) +
  labs(title = "Premium amount vs MaritalStatus",x = "",
      y = "")

p4 =ggplot(data,
           aes(x = Veh_Owned,
               fill = Marital_Status)) +
  geom_density(alpha = 0.4) +
  labs(title = "Vehicles Owned vs MaritalStatus",x = "",
      y = "")

p5=ggplot(data,
          aes(x = risk_score,
              fill = Marital_Status)) +
  geom_density(alpha = 0.4) +
  labs(title = "risk_score vs MaritalStatus",x = "",
      y = "")

grid.arrange(p1,p2,p3,p4,p5, nrow=2,ncol=3)
```

```
data$Accomodation <- as.factor(data$Accomodation)
data$renewal <- as.factor(data$renewal)

##
p1 = ggplot(data,
         aes(x =`Count_3-6_months_late` ,
             fill = renewal)) +
  geom_density(alpha = 0.4) +
  labs(title = "delay by 3 to 6 mnths vs renewal status",x = "",
       y = "")
p2=ggplot(data,
       aes(x =`Count_6-12_months_late` ,
           fill = renewal)) +
  geom_density(alpha = 0.4) +
  labs(title = "delay by 6 to 12 mnths vs renewal status",x = "",
       y = "")
p3=ggplot(data,
          aes(x =`Count_more_than_12_months_late` ,
              fill = renewal)) +
  geom_density(alpha = 0.4) +
  labs(title = "delay by more thn 12 mnths vs renewal status",x = "",
       y = "")
grid.arrange(p1,p2,p3, nrow=1,ncol=3)

##
p1 = ggplot(data,
       aes(x =age_in_days ,
           fill = renewal)) +
  geom_density(alpha = 0.4) +
  labs(title = "Age vs renewal status",x = "",
       y = "")

p2 = ggplot(data,
       aes(x =risk_score ,
           fill = renewal)) +
  geom_density(alpha = 0.4) +
  labs(title = "Risk score vs renewal status",x = "",
       y = "")

p3 = ggplot(data,
       aes(x =no_of_premiums_paid ,
           fill = renewal)) +
  geom_density(alpha = 0.4) +
  labs(title = "No of premiums paid vs renewal status",x = "",
       y = "")
grid.arrange(p1,p2,p3, nrow=1,ncol=3)

##quantitative vs quantitative
p1 = ggplot(data,
       aes(x = perc_premium_paid_by_cash_credit,
```

```
          y = Income)) +
geom_point(color= "steelblue") +
  geom_smooth(method = NULL ,color = "indianred3") +
  labs(x = "Premium percentage in cash/credit",
       y = "Income",
       title = "Premium percentage in cash vs. Income")

p2 = ggplot(data,
       aes(x = age_in_days,
           y = Income)) +
  geom_point(color= "steelblue") +
  geom_smooth(method = "lm") +
  labs(x = "Age of customer",
       y = "Income",
       title = "Age of customer vs. Income")

p3 = ggplot(data,
       aes(x = age_in_days,
           y = risk_score)) +
  geom_point(color= "steelblue") +
  geom_smooth(method = "lm") +
  labs(x = "Age of customer",
       y = "risk score",
       title = "Age of customer vs. risk score")

p4 = ggplot(data,
       aes(x = age_in_days,
           y = premium)) +
  geom_point(color= "steelblue") +
  geom_smooth(method = "lm") +
  labs(x = "Age of customer",
       y = "premium",
       title = "Age of customer vs. premium")

p5 = ggplot(data,
       aes(x = Income,
           y = premium)) +
  geom_point(color= "steelblue") +
  geom_smooth(method = NULL ,color = "indianred3", formula = y ~ x) +
  labs(x = "Income",
       y = "premium",
       title = "Income Vs Premium")

p6 = ggplot(data,
       aes(x = risk_score,
           y = premium)) +
  geom_point(color= "steelblue") +
  geom_smooth(method = NULL ,color = "indianred3", formula = y ~ x) +
  labs(x = "Risk Score",
       y = "premium",
```

```
      title = "Risk Score Vs Premium")

grid.arrange(p1,p2,p3,p4,p5,p6, nrow=2,ncol=3)
```

9.3 Section 3

```
##-------------------------------------------------------------------------##
#                    missing value treatement                             #
##-------------------------------------------------------------------------##

plot_intro(data)
plot_missing(data)

##-------------------------------------------------------------------------##
#                    Variable Treatment                                   #
##-------------------------------------------------------------------------##

#change decimals to percentage
data$perc_premium_paid_by_cash_credit <-
formattable::percent(data$perc_premium_paid_by_cash_credit)

#remove percentage symbol
data$perc_premium_paid_by_cash_credit <-  as.numeric(gsub("[\\%,]", "",
                                          data$perc_premium_paid_by_cash_credit))



#calculate age of customer in years using current date

data$age_in_years <- round((data$age_in_days/365),digits = 2)

#removing age in days variable
data <- data[-c(3)]

#creating age groups for further analysis
data$Agegroup <- cut(data$age_in_years, breaks = c(20,40,60,80,100,110), labels = c( "20 to 40","40
to 60","60 to 80","80 to 100", "more than 100"))

#creating income groups for further analysis
data$Incomegroup <- cut(data$Income, breaks =
c(20000,50000,100000,1000000,10000000,100000000),
                    labels = c("Less than 50thousand","50thousand to 0.1million","0.1million to
1million","1million to 10million","10million & above"))

#creating percentage premium in cash group for further analysis
data$Perc_premium_cash_grp <- cut(data$perc_premium_paid_by_cash_credit, breaks = c(-
1,25,50,75,100),
```

```
                          labels = c("Less than 25%","25% to 50%","50% to 75%","75% and above"))

#creating risk score group for further analysis
data$risk_score_grp <- cut(data$risk_score, breaks = c(91,93,95,97,100),
                           labels = c("91 to 93","93 to 95","95 to 97","97 and above"))

#creating premium group for further analysis
data$premium_amt_grp <- cut(data$premium, breaks =
c(1000,10000,20000,30000,40000,50000,60000),
                    labels = c("Less than 10K","10K to 20K","20K to 30K","30K to 40K","40K to
50K","50K and above"))



##-----------------------------------------------------------------------------##
#                      Outlier Treatment                                      #
##-----------------------------------------------------------------------------##

###premium amount
data <- read_excel("premium.xlsx")
capOutlier <- function(x){
   qnt <- quantile(x, probs=c(.25, .75), na.rm = T)
    caps <- quantile(x, probs=c(.05, .95), na.rm = T)
    H <- 1.5 * IQR(x, na.rm = T)
    x[x < (qnt[1] - H)] <- caps[1]
    x[x > (qnt[2] + H)] <- caps[2]
    return(x)
  }

data$no_of_premiums_paid=capOutlier(data$no_of_premiums_paid)
data$premium = capOutlier(data$premium)

boxplot(data$premium,
                main = "Premium Amount of Customers",
                 ylab = "Count",
                 col = "orange",
                 border = "brown",
                 horizontal = TRUE,
                 notch = TRUE)

boxplot(data$`no_of_premiums_paid`,
       main = "No of Premium paid by Customers",
       ylab = "Count",
       col = "orange",
       border = "brown",
       horizontal = TRUE,
       notch = TRUE)


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

```
#            Exploratory Data Analysis                                  #
##-------------------------------------------------------------------------##

ggplot(data, aes(x = perc_premium_paid_by_cash_credit,
                 y = Income,
                 color=Agegroup)) +
  geom_point() +
  labs(title = "Insights by Perc premium paid cash/credit, Income and Agegroups")


ggplot(data, aes(x = Income,
                 y = premium,
                 color=residence_area_type)) +
  geom_point(size=3,alpha=.6) +
  geom_smooth(se=FALSE,
              method = "lm",
              formula = y~poly(x,2),
              size = 1.5) +
  labs(title = "Customer income vs premium across residence area types",color="rsidence_area_type")

ggplot(data = data, mapping = aes(x = risk_score_grp, y = Income)) +
  geom_point(alpha = 2, aes(color = premium_amt_grp)) +
  labs(title = "Customer income vs riskscore across premium amounts",color="Premium Amount
Groups")
ggplot(data = data, mapping = aes(x = age_in_years, y = no_of_premiums_paid)) +
  geom_point(alpha = 2, aes(color = Incomegroup)) +
  labs(title = "Customer age vs premiums paid across Incomegroup",color="Income Groups")

###correlations

cordata <- data[,c(2,3,4,5,6,8,9,11,12,15,17)]
corr <- round(cor(cordata),1)
corr

ggcorrplot(corr,hc.order = TRUE, type="lower",
           lab = TRUE, lab_size = 3, method="circle",
           colors = c("blue", "white", "red"),
           outline.color = "gray", show.legend = TRUE, show.diag = FALSE,
           title="Correlogram of insurance variables")

##veryfying our outcome via test
cor(data$Income,data$Veh_Owned)
##0.002 shows positive but weak relationship
cor.test(data$Income,data$Veh_Owned)
##p-value 0.46 is greater than 0.05, we fail to reject the null hypothesis that the
##relationship between the vehicles owned by customer and their income is not significant
cor(data$Income,data$premium)
cor.test(data$Income,data$premium)

#######################################
```

```
##correlation between catgorical
#######################################
mar_renew <- table(data$Marital_Status,data$renewal)
mar_renew
chisq.test(mar_renew,correct=FALSE)$expected

chisq.test(mar_renew,correct=FALSE)
##output p is approx equal to 0.05, we accept the null hypothesis that marital status
##of customer is associated with renewal status
acc_renew <- table(data$Accomodation,data$renewal)
acc_renew
chisq.test(acc_renew,correct=FALSE)$expected
chisq.test(acc_renew,correct=FALSE)
```

9.4 Section 4

```
library(scales)
library(data.table)
library(car)
library(ggcorrplot)
library(randomForest)
library(caret)
library(varImp)
library(Boruta)
library(pROC)
library(plotROC)
library(InformationValue)
library(rpart)
library(rpart.plot)
library(rattle)
library(RColorBrewer)
library(ROSE)
library(caret)
library(ipred)
library(ROSE)
library(ineq)
library(ROCR)
library(data.table)
library(scales)

#############################################################################
##                      MODELLING                          ##
#############################################################################

table(data$renewal)

#Paritioning the data into training and test dataset

inp1 <- data[which(data$renewal == 1),] ##all 1's
inp0 <- data[which(data$renewal == 0),]  ##all 0's
```

```
set.seed(1000)
##train data
inp1_train <- sample(1:nrow(inp1),0.7*nrow(inp1))
inp0_train <- sample(1:nrow(inp0),0.7*nrow(inp0))

train1 <- inp1[inp1_train,]
train0 <- inp0[inp0_train,]

data_train <- rbind(train1,train0)

##test data
test1 <- inp1[-inp1_train,]
test0 <- inp0[-inp0_train,]
data_test <- rbind(test1,test0)

#########################
## Important variables  #
#########################

##random forest
fit = randomForest(renewal ~ ., data = data_train)
varImp(fit)
varImpPlot(fit,type = 2)
```

9.5 Section 5

```
#########################
##  logistic regression ##
#########################
#model on train dataset
model_glm <- glm(renewal ~ ., data = data_train, family = binomial(link="logit"))
summary(model_glm)

confusionMatrix(model_glm)

#predicted model on train data
predictedTrg <- predict(model_glm, data_train)

#predict model on test data
predicted <- predict(model_glm,data_test, type = "response" )

### Model Diagnostics
summary(model_glm)

#variable inflation factor
vif(model_glm)

#AUROC for training dataset
```

```
roctrain <- roc(data_train$renewal, predictedTrg)
auc(roctrain)   ##Area under the curve: 0.8283

coords(roctrain,x="b")
#  threshold specificity sensitivity
#  2.787837   0.7590051   0.7467079

plotROC(data_train$renewal,predictedTrg)

##AUROC for test dataset
roctest <- roc(data_test$renewal, predicted)
auc(roctest)
#Area under the curve: 0.8364

coords(roctest,x="b")
#  threshold specificity sensitivity
#  0.932513   0.7373333    0.780781

plotROC(data_test$renewal, predicted)

##misclassification error
optcutoff <- optimalCutoff(data_test$renewal,predicted)[1]
misClassError(data_test$renewal,predicted, threshold = optcutoff)

confusionMatrix(data_test$renewal,predicted, threshold = optcutoff)
```

9.6 Section 6

```
########################
####    CART    #######
########################

r.ctrl <- rpart.control(minsplit = 100,
                        minbucket = 10,
                        cp = 0,
                        xval = 10
)

cart.train <- data_train
names(cart.train)

##exclude id
m1 <- rpart(formula = renewal~.,
            data = cart.train[,-c(1)],
            method = "class",
            control = r.ctrl
)

fancyRpartPlot(m1)

printcp(m1)
```

```
plotcp(m1)

ptree<- prune(m1, cp= 0.0017 ,"CP")
printcp(ptree)

##plotting final CART model
fancyRpartPlot(ptree,
               uniform = TRUE,
               main = "Final Tree",
               palettes = c("Blues", "Oranges")
)


##performance measures on training dataset
cart.train$predict.class = predict(ptree, cart.train, type = "class")
cart.train$predict.score = predict(ptree, cart.train, type = "prob")

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
                                                          ))))))))))
}
#class(cart.train$predict.score)
## deciling
cart.train$deciles <- decile(cart.train$predict.score[,2])

##Ranking

tmp_DT = data.table(cart.train)


pred <- prediction(cart.train$predict.score[,2], cart.train$renewal)
perf <- performance(pred, "tpr", "fpr")

KS <- max(attr(perf, 'y.values')[[1]]-attr(perf, 'x.values')[[1]])
```

```
auc <- performance(pred,"auc");
auc <- as.numeric(auc@y.values)

gini = ineq(cart.train$predict.score[,2], type="Gini")
with(cart.train, table(renewal, predict.class))

plot(perf)
KS

auc

gini

##KS = 26.4% and auc = 63.4% which indicates that the model is not good
#gini coefficient 1.6% also indicates model is not fit for use
##confusion matrix
##accuracy = (499 + 52100)/(499+2999+298+52100) =   94.10%
##classification error rate = 1- accuracy = 0.59 %
```

9.7 Section 7

```
########################################################## ####
##bagging cart model ####
########################################################## ####

set.seed(1000)

ctrl <- trainControl(method = "cv",  number = 10)


bagged_m1 <- bagging(
  formula = renewal ~ .,
  method = "treebag",
  trControl = ctrl,
  importance = TRUE,
  data    = data_test
)

##predictions on test data
predictTest <- predict(bagged_m1,newdata = data_test)

##confusion matrix
table(data_test$renewal,predictTest)

##accuracy
```

```
(1452+22457)/(1452+22457+48+0)
##99.79% accurate now

###Stochastic Gradient Boosting ####

set.seed(1000)

control2 <- trainControl(sampling="rose",method="repeatedcv", number=5, repeats=5)

gbm_model <- train(renewal ~., data=data_test, method="gbm", metric="Accuracy",
trControl=control2)

predictTest = predict(gbm_model, newdata = data_test)

table(data_test$renewal, predictTest)

#Accuracy -- 92.08%
(576+21485)/(972+924+576+21485)
```