

Insurance PREMUIM Default

Deborah Osilade



Table of Contents

[Introduction 2](#_Toc52154962)

[Data report 2](#_Toc52154963)

[Renaming Variables 4](#_Toc52154964)

[Data pre-processing 4](#_Toc52154965)

[Variable Transformation 4](#_Toc52154966)

[Addition of new variables 4](#_Toc52154967)

[Removal of unwanted variables 4](#_Toc52154968)

[Exploratory Dataset Analysis 5](#_Toc52154969)

[Univariate 5](#_Toc52154970)

[Figure 1: Histogram of variables 6](#_Toc52154971)

[Figure 2: Boxplot of variables 7](#_Toc52154972)

[Figure 3: plot of factor variables 8](#_Toc52154973)

[Figure 4: plot of Customers 8](#_Toc52154974)

[Correlation 9](#_Toc52154975)

[Figure 5: Correlation plot 9](#_Toc52154976)

[Figure 6: Variable relationships 10](#_Toc52154977)

[Figure 7: Plot of variable relationship 10](#_Toc52154978)

[Binning Variable 11](#_Toc52154979)

[Customer Behaviour 11](#_Toc52154980)

[Outlier Treatment 12](#_Toc52154981)

[Figure 8 13](#_Toc52154982)

[Figure 9 14](#_Toc52154983)

[Analytical Approach 14](#_Toc52154984)

[Improving the Dataset 14](#_Toc52154985)

[Split dataset 15](#_Toc52154986)

[Building the Models 15](#_Toc52154987)

[Model 1: Decision Trees 15](#_Toc52154988)

[Figure 10: 15](#_Toc52154989)

[Variable importance for Cart model 16](#_Toc52154990)

[Model Performance 16](#_Toc52154991)

[Random Forest 18](#_Toc52154992)

[Figure 11: 19](#_Toc52154993)

[Variable Importance 19](#_Toc52154994)

[Figure 12: 20](#_Toc52154995)

[Model Performance 20](#_Toc52154996)

[ROC/ AUC 21](#_Toc52154997)

[Logistic Regression 22](#_Toc52154998)

[Variable Importance 23](#_Toc52154999)

[ROC/AUC 23](#_Toc52155000)

[Model Performance 25](#_Toc52155001)

[Extreme Gradient Boosting 26](#_Toc52155002)

[Variable Importance 26](#_Toc52155003)

[Model performance 27](#_Toc52155004)

[Model Comparison 27](#_Toc52155005)

[Business Insights 28](#_Toc52155006)

[Recommendations 29](#_Toc52155007)

[Appendix 29](#_Toc52155008)

# Introduction

The insurance company is challenged with major situations that rule the industry, mostly the revenue is generated by charging premiums in exchange for insurance coverage, then reinvesting those premiums into other money generating assets like bonds, treasury bills etc. to market effectively and to minimise administrative costs.

However, customer retention is far more profit priority because keeping an existing customer is much cheaper than going out and finding a new one. Mckinsey report uncovered in a statistics research that satisfied customers are 80% more likely to renew their current policies than unsatisfied. Insurance companies build their business models on assumptions and risk diversification. On the other hand, based on the dataset, this company is most likely facing a challenge of customer retention (customer satisfaction) and late premium payments. A Default in premium payments results in significant revenue losses for insurance companies which would cause a major setback for the company to generate revenue.

Problem statement

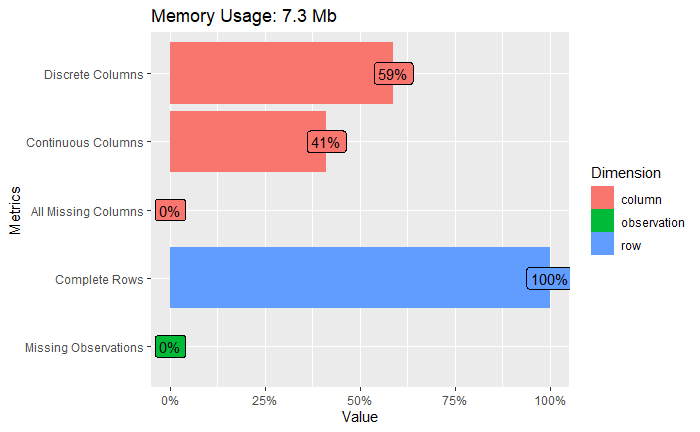
The objective of this project is to use predictive modelling to know the type of customers who are likely to default the premium payments, the Insurance premium dataset, R programming language would be used in exploring the data set.

**Design a strategy to predict the probability that a customer will default the premium payment, so that the insurance agents can proactively reach out to the policy holder to follow up for the payment of premium.**

# Data report

Data was collected and sourced from the company’s database; this dataset represents records of customers premium payments over a period of time. However, premium seems to the paid annually although the question arises from the amount of times premium was paid and the late payment frequency as it is normal for insurance companies to intercede for the insured, in areas of premium financing or premium payment split. It could be that some customers may have had their premium paid monthly and also quarterly.

The dataset shows no missing values. The dataset contains the following information of 79,854 policy holders, 17 columns.



The dataset is made up of Customer demographics includes – ID, Age in days, Income, Martial status, Vehicle owned, No of dependants, Accommodation, Residential area type. Customer Behaviour includes – Sourcing channel, Risk score, Default, Count\_3-6\_months\_late, Count\_6-12\_months\_late, Count\_more\_than\_12\_months\_late. Per premium paid by cash, no of premiums paid, premium.

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **Variable Name** | **Variable Definition** | **Expected Data Type** |
| 1 | ID | Unique customer ID | Numeric |
| 2 | Per premium paid by cash | What % of the premium was paid by cash payments? | Numeric |
| 3 | Age in days | age of the customer in days | Numeric |
| 4 | Income | Income of the customer | Numeric |
| 5 | Marital Status | Married (1), unmarried (0) | Factor |
| 6 | Veh owned | Number of vehicles owned (1-3) | Factor |
| 7 | Count\_3-6\_months\_late | Number of times premium was paid 3-6 months late | Factor |
| 8 | Count\_6-12\_months\_late | Number of times premium was paid 6-12 months late | Factor |
| 9 | Count\_more\_than\_12\_months\_late | Number of times premium was paid more than 12 months late | Factor |
| 10 | Risk score | Risk score of customers (similar to credit score) | Numeric |
| 11 | NO of Dep | Number of dependents in the family of the customer (1-4) | Factor |
| 12 | Accommodation | Owned (1), Rented (0) | Factor |
| 13 | No of premiums paid | Number of premiums paid till date | Numeric |
| 14 | Sourcing channel | Channel through which customer was sourced | Factor |
| 15 | Residence area type | Residence type of the customer | Factor |
| 16 | premium | Total premium amount paid till now | Numeric |
| 17 | Default | Y variable - 0 indicates that customer has defaulted the premium and 1 indicates that customer has not defaulted the premium | Factor |

## Renaming Variables

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **Variable Name** | **Variable changed** | **Expected Data Type** |
| 2 | Per premium paid by cash credit | Premium % by cash | Numeric |
| 3 | Age in days | Age of customer | Numeric |
| 7 | Count 3-6 months late | late in 3-6 months | Factor |
| 8 | Count 6-12 months late | late in 6-12 months | Factor |
| 9 | Count more than 12 months late | late more than 12 months | Factor |
| 10 | premium | Total premium | Numeric |

# Data pre-processing

## Variable Transformation

* **Age of customer** - Age of customer in days into Age in years: This Variable was changed because it is better to see a customer’s age in years than days. It is more realistic to reference.

## Addition of new variables

* **Cash payments**: The variable Cash Payment was gotten from the percentage of premium paid by cash, this is because it is more efficient to work with real cash payments other than payment in percentages, and it gives a convincing insight into the data of how much of cash at hand payments have been made by customers – customers who have pay more than half of premium in cash could defaulters because they only pay in cash, it is more likely to have customers that pay only cash.

## Removal of unwanted variables

* **Per premium paid by cash credit / Premium % by cash:** According to the information value, % premium is too good to be true, and it’s more likely to be dropped. Although it could contribute to making a CART model decision tree when deciding the default on how to identify a defaulter.
* **ID**: This refers to customers ID. How the customers have been identified by the insurance company over time. However, this holds no predictive power for deciding whether the customers would default or not.

Using Information Value to tell the predictive power of all variables in relation to default variable.

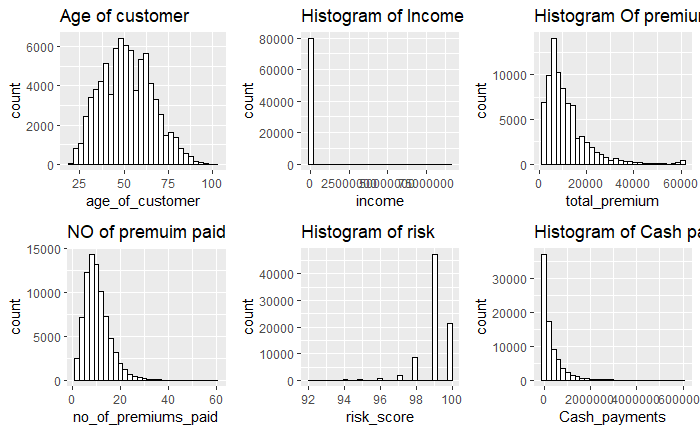
|  |  |
| --- | --- |
| **Information Value** | **Predictive Power** |
| < 0.02 | Useless for prediction |
| 0.002 to 0.1 | Weak predictor |
| 0.1 to 0.3 | Medium predictor |
| 0.3 to 0.5 | Strong predictor |
| >0.5 | Suspicious or too good to be true |

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **Variable** | **IV** | **Predictive power** |
| 1 | accommodation | 0.00028466771 | Useless for prediction |
| 2 | Age of customer | 0.19285426327 | Medium predictor |
| 3 | Cash payments | 0.52819409237 | Strong Predictor |
| 4 | ID | 0.00291218231 | Useless for prediction |
| 5 | Income | 0.08744103765 | Weak Predictor |
| 6 | Late in 3 6 months | 0.60496093530 | Suspicious |
| 7 | Late in 6 12 months | 0.71404548815 | Suspicious |
| 8 | Late more than 12 months | 0.46660020237 | Strong Predictor |
| 9 | Marital status | 0.00077955144 | Useless for prediction |
| 10 | No of dep | 0.00067100632 | Useless for prediction |
| 11 | No of premiums paid | 0.10960823056 | Weak Predictor |
| 12 | Premium percent by cash | 0.89464670214 | Suspicious |
| 13 | Residence area type | 0.00004730955 | Useless for prediction |
| 14 | Risk score | 0.07095078116 | Weak predictor |
| 15 | Sourcing channel | 0.02867375243 | Weak Predictor |
| 16 | Total premium | 0.03932802535 | Weak Predictor |
| 17 | Vehicle owned | 0.00006977014 | Useless for predictor |

## Exploratory Dataset Analysis

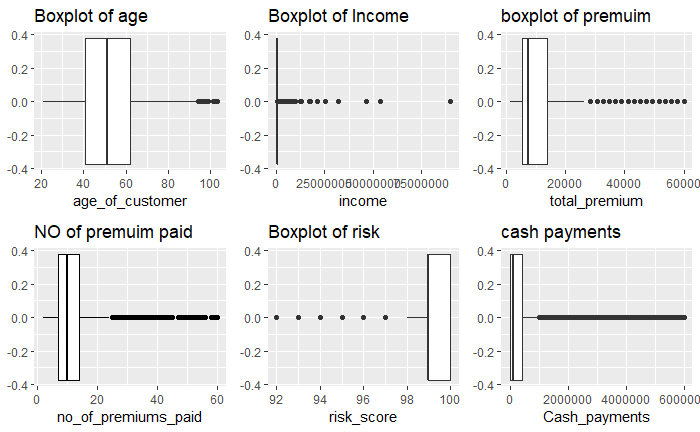
## Univariate

### Figure 1: Histogram of variables



* The histogram of premium to skewed to the right, showing the presence of outliers, doesn’t show a normal distribution.
* Income variable is affected by extreme values of 90 million, however it is impossible to see the original values. On the other hand, binning the values and treating outliers would solve this.
* The age of customers seems to be normally distributed within the data.
* Most customers have paid premium between 10 to 15 times, showing that it is more likely that older customers do not stay, depending on their insurance policy however.

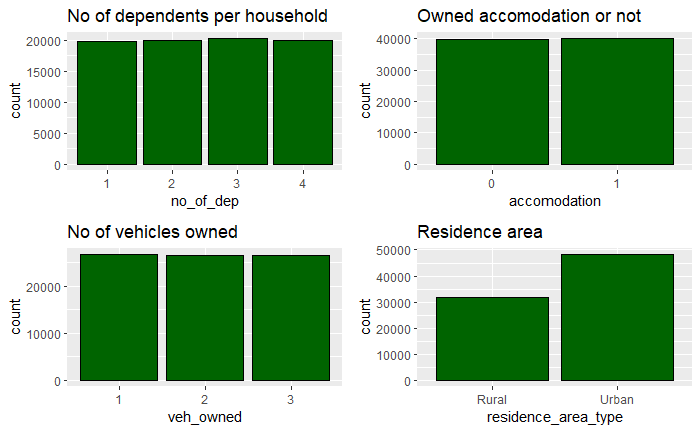
### Figure 2: Boxplot of variables



From figure 2,

* All boxplots show outliers especially income, the extreme value is 90 million.
* Risk score shows outliers, more than 50% of the risk score is at 99% showing that it is unlikely for customers who default to affect the company’s revenue.
* Risk score is also questionable because most people who default also have a perfect or an almost perfect risk score which is not acceptable.
* Customers are paying more cash payments, up to 60 million of premium has been paid in cash. This raises the question of if there are alternatives to payment made available for the customers it could be that customers have no access to easy payments.

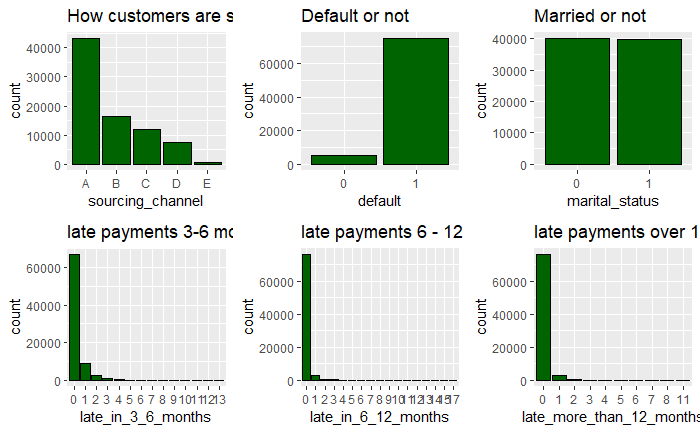
### Figure 3: plot of factor variables



**From figure 3,**

* Most customers live in the Urban residential area and have maximum of 3 dependants.

### Figure 4: plot of Customers



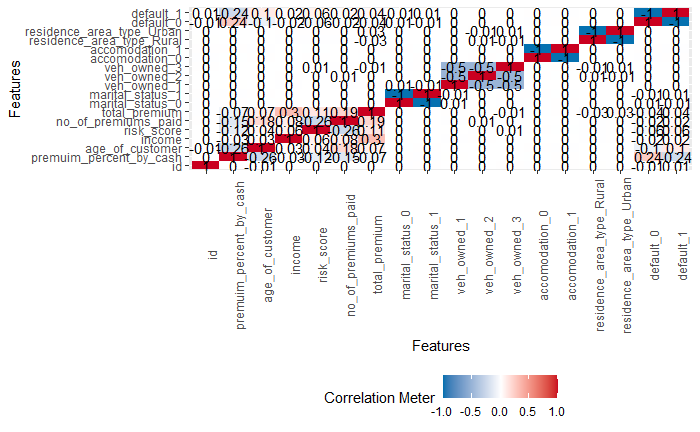
**From figure 4,**

* Customers are more likely to be 3-6 months late or stop paying premium.
* There are more customers who do not default payments and they represent 94% of the dataset.
* Customers are mostly sourced via the A sourcing channel and these customers show a high-risk score.

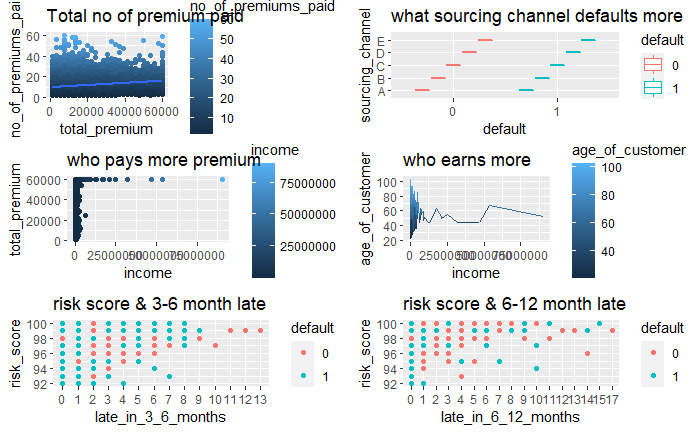
## Correlation

Before threating outliers. There is no strong correlation between the variables in figure 5. However, total premium, income and no of premium show a positive correlation.

## Figure 5: Correlation plot



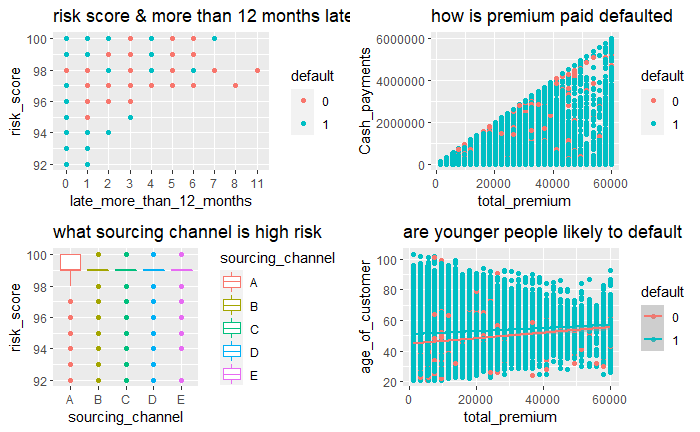
### Figure 6: Variable relationships



**From figure 6,**

* Costumers from sourcing channel D pays the highest amount of premium
* Over 75% of premium is above 20 million causing the presence of outliers in the data.
* Costumers in sourcing channel E has the highest default and non – default rate.
* The higher the age of customer the higher the income earned.

### Figure 7: Plot of variable relationship



**From figure.7,**

* People with risk score between 95 – 98 are most likely to default while customers with a risk score of 99 is most likely to pay premium between 3-6 months.
* Customers with risk score of 100 and 99, have the highest late payments between 6 -12 months. Late payments over a year are considered high risk, they seem more likely to not stay.
* Customers who pay late premium between 6-12 months and have a risk score of 97 are mostly defaulters. And this questions how the risk score have been recorded, there is a doubt that it has been recorded correctly.
* People with higher amount of premium to be paid default the most.
* Customers from sourcing channel A have the higher risk score than other customers.
* Younger people are more likely to default payment.

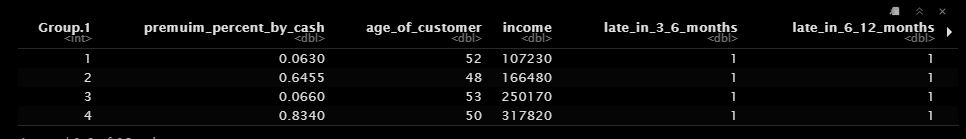
# Binning Variable

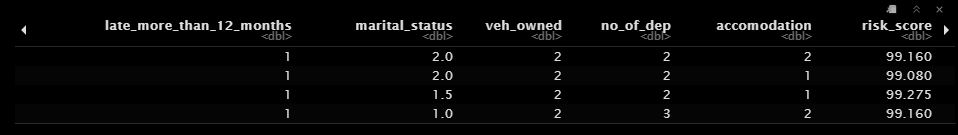
Binning the “Late” categorical variables makes understanding the level of lateness more efficient, variables Late in 3 6 months, Late in 6 12 months, Late more than 12 months.

|  |  |
| --- | --- |
| **Binning name** | **What it represents** |
| Never late | 0 |
| Unpunctual | 1- 3 |
| Delays | 4 – 7 |
| Long delayed | 8 – 11 (depending on how the entries are dispersed) |
| Behind time | Over 12 (depending on how the entries are dispersed) |

# Customer Behaviour

The clustering algorithm was used to analyse and to understand the behaviour of the customers in relation to default. Using Clara algorithms “medoids” because mean is very sensitive to outliers. The optimum number of clusters recommended is 2.







Cluster 4 = Customers highest premium percentage paid by cash, highest income earners, high no of premium paid, highest total premium paid, highest cash payment paid.

Cluster 3 = Group of older customers, Highest risk score

Cluster 2 = customers here have lowest percentage of cash payments, lowest risk score, low premium payments. Cash payments greater than 100,000, younger aged customers

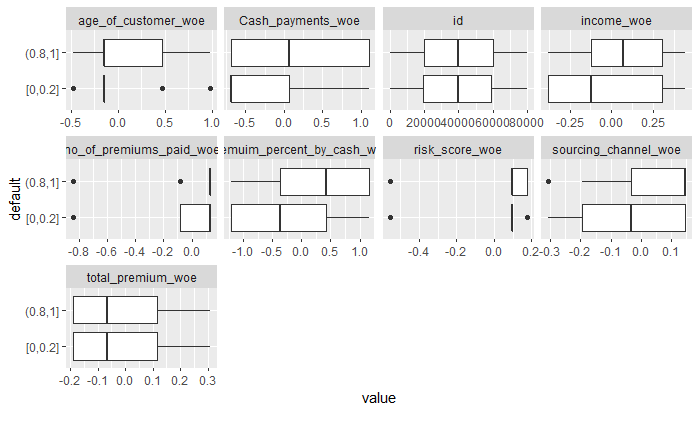
Cluster 1 = Lowest cash payments, total premium, no of premium paid and lowest income earners.

Choosing cluster 2 as the best choice to decide if a customer would default or not – as customers in the group are mainly younger customers and low-income earner, although they have not paid highest cash payments, they have the lowest risk core. The insurance agent can start a follow up with this customer and offer them benefits and policy that could make them comfortable. It could be they because they are younger, they feel that paying for insurance could be a waste of money.

# Outlier Treatment

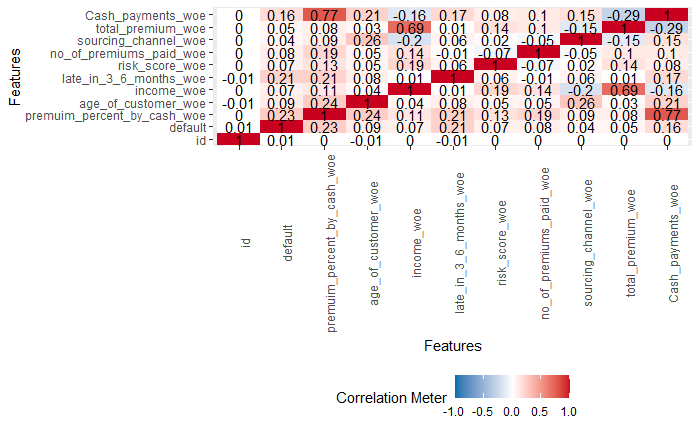
Using weight of evidence which helps to transform the continuous independent variable into a set of bins based on the similarity of the default distribution. However, using the package woe also takes care of removing unwanted variables, i.e. variables that have a useless or suspicious predictive power.

### Figure 8



After treating outliers using the weight of evidence, variables now show a strong correlation amongst each other in figure 9.

### Figure 9



* There is a very strong correlation between Income and Total premium.
* There is also correlation between sourcing channel and age of customer, it also indicates that customers are not sourced based on the earning power.

# Analytical Approach

At this phase, we are faced with the classification problem. The insurance dataset is quite imbalanced, and due to smote up to 18% of artificial values have been generated to help improve and the models. On the other hand, the insurance problem “**what customers would default**” can be address using a Random forest algorithm, Decision trees, Logistic regression and XGboost algorithms considering it is a binary classification problem and later, would be tested on the validation set to see how well the model has performed and gives the best decision.

# Improving the Dataset

As a result of imbalance in the classification problem **- Yes (4998), No (74855),** we use smote to improve the data by oversampling up to 18% in order to retain the relevant information in the real dataset and to optimise the sensitivity to get the best out of the model.

**oversampling with smote at 18%**

|  |  |  |
| --- | --- | --- |
|  | **YES** | **NO** |
| Insur smote | 14994 | 19992 |

**After using Smote and adding it to the dataset**

|  |  |  |
| --- | --- | --- |
|  | **YES** | **NO** |
| Insurance Smote | 19992 | 94847 |

# Split dataset

To solve the binary problem, the default variable has been revelled to represent 0 as Yes for customers who default and (1) as No for non-default customers. After splitting the improved smote data into test and train,

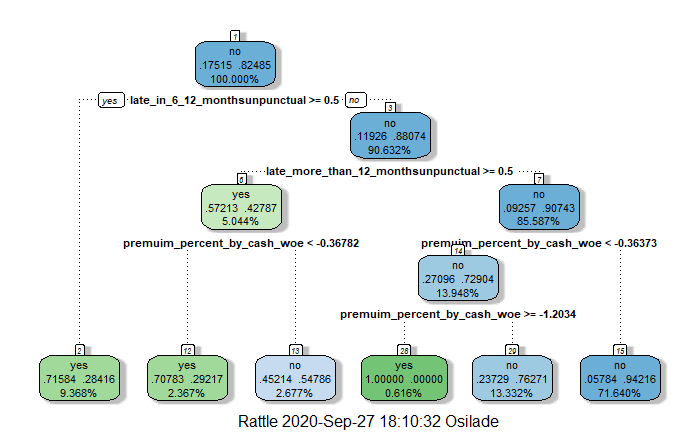
|  |  |  |
| --- | --- | --- |
|  | **YES** | **NO** |
| Train dataset | 14080 | 66307 |
| Test dataset | 5912 | 28540 |

# Building the Models

# Model 1: Decision Trees

Building a decision tree to know the customers that are likely to default the premium payment, so that the insurance agent reaches out to the policy holder to follow up for on payment of premium. Here, the CART model has been designed to help with the decision making, using the decision rules below.

### Figure 10:



The decision rules states

* Customers who defaults represents only a 6% of the dataset
* Has a customer been late in 6 – 12 months? If yes then yes, the customer is a defaulter.
* If No – has the customer has been late for more than 12 months - if yes then the customer has a 57% chance of being a defaulter.
* Has the customer paid more than 75% premium with cash? If it is true, the customer stands a 70% chance of defaulting the premium.
* If the customer has not been late then the customer has a 42% chance of not defaulting premium. However, has the customer made premium cash payments of 4% and above, if No the customer cannot only be labelled a defaulter
* Has the customer paid cash premium more than or 4 times, if true, than our customer has a 100% chance of being labelled and followed up by the insurance agent. This rule could be as a because of new customers, it is likely that a customer is on retention if premium has been paid up to 4 times and above.

Strategy that can be implemented to reduce default rate

• Offer a customer loyalty leader board program to customers who have up to 4 times premium

• A customer who has been in late more than 12 months and falls under the group of unpunctual (meaning the customer has been late for up to 3 times) should have special follow up service or sent a survey to understand how the insurance company can assist.

## Variable importance for Cart model

|  |  |
| --- | --- |
|  | Overall |
| premuim\_percent\_by\_cash\_woe | 7015.31345 |
| late\_more\_than\_12\_months | 6377.40371 |
| late\_in\_6\_12\_months | 5035.64109 |
| Late in 3 6 months woe | 4648.56883 |
| cash\_payments\_woe | 3359.89938 |

From the table above, the most important variables from the data shows the important variables used in the model as Premium percent by cash woe, late more than 12 months, Late in 6 12 months, Late in 3 6 months woe and cash payments. Values below 3000 are not considered relevant to predict the type of customers who are likely to default.

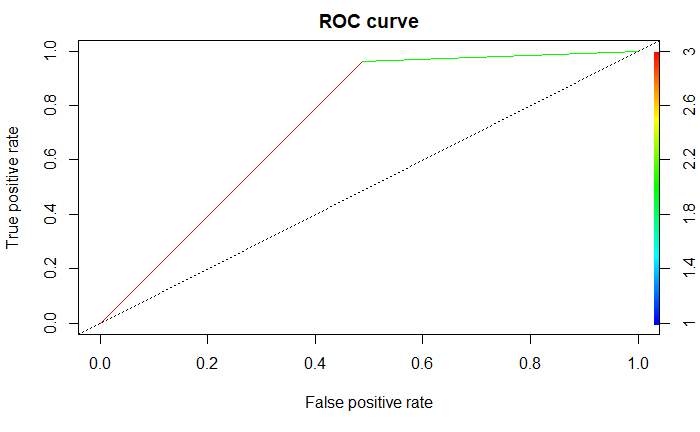
## Model Performance

The confusion Matrix for the cart model, predicted on the test data.

|  |  |  |
| --- | --- | --- |
| **Prediction** | **Reference** | |
|  | **YES** | **NO** |
| **YES** | **3322 (TP)** | **1049 (FP)** |
| **NO** | **2590 (FN)** | **27491 (TN)** |

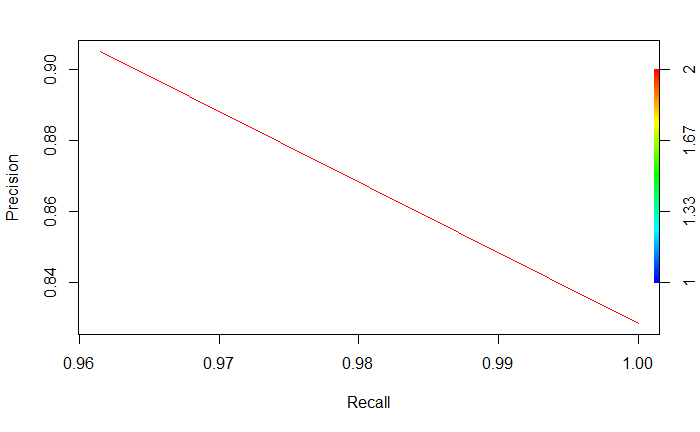
Classification error gives a clear picture of the % of the observations that have been predicted wrongly and correctly by the model. The classifier made 34,452 predictions for customers who are likely to default

* 5192 was predicted as yes and are default customers and 28540 was predicted correctly as 1 No
* TN says we predicted No and they are not defaulters, TP says we predicted Yes and they are defaulters.
* FP says We predicted yes and they are not defaulters, FN says we predicted no but they are defaulters.



AUC = 73%

AUC (Area Under Curve) is quite high which also suggests that the model is good.



Accuracy – 89%

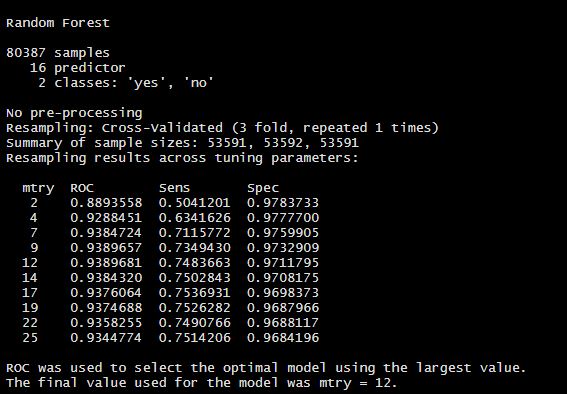
Sensitivity – 56%

Specificity – 96%

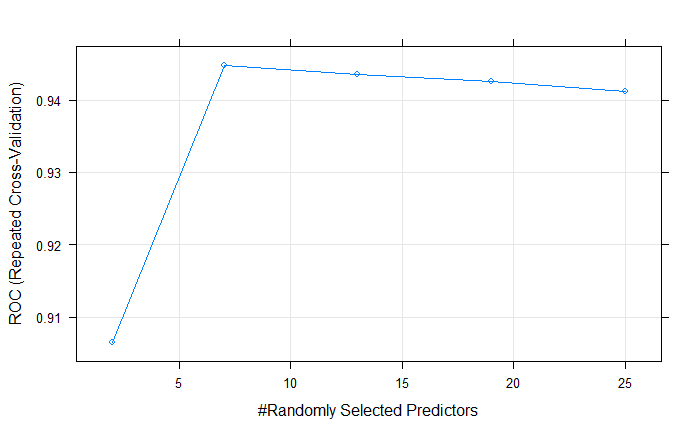
* The above numbers are calculated on the validation sample that was not used for training the model.
* Sensitivity is the percentage of yes’s correctly predicted by the model, while specificity is the percentage of no’s correctly predicted.
* So true detection rate on test data is bad.

# Random Forest

The random forest was used because random forest is historically robust, it is believed to accommodate robust data and give a better result than most models. However, because a decision tree is always faced with the problem of overfitting



### Figure 11:

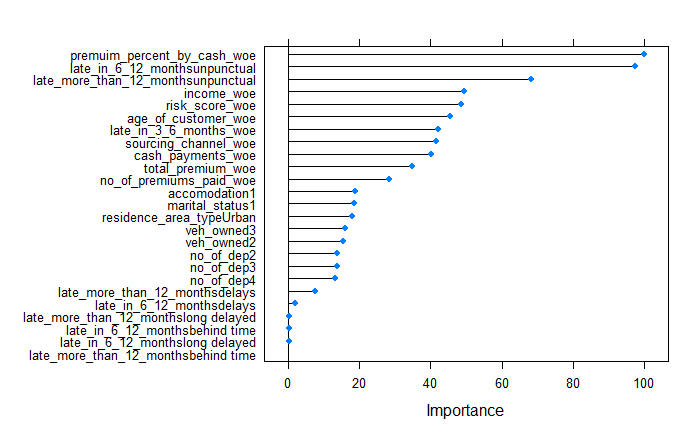


From figure 11, the No. of variables tried at each split is 5 and the model shows an Out Of Bag error rate 7.78% .

## Variable Importance

only 20 most important variables shown (out of 25)

### Figure 12:



From figure 12 above, the important variable used in the random forest is premium percent by cash with mean decrease gini of 90. Variables that a gini of 50 above are considered the variable that have the most predictive power to determine the customers that are likely to default.

## Model Performance

The confusion Matrix for the random forest model, predicted on the test data.

|  |  |  |
| --- | --- | --- |
| **Prediction** | **Reference** | |
|  | **YES** | **NO** |
| **YES** | **4463 (TP)** | **513 (FP)** |
| **NO** | **1449 (FN)** | **28027 (TN)** |

Classification error gives a clear picture of the % of the observations that have been predicted wrongly and correctly by the model. The classifier made 34,452 predictions for customers who are likely to default

* 5192 was predicted as yes and are default customers and 28540 was predicted correctly as 1 No
* TN says we predicted No and they are not defaulters, TP says we predicted Yes and they are defaulters.
* FP says We predicted yes and they are not defaulters, FN says we predicted no but they are defaulters.

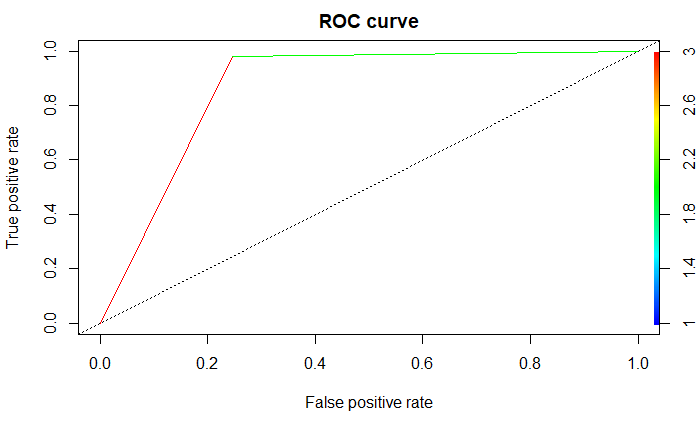
Accuracy – 94%

Sensitivity – 75%

Specificity – 98%

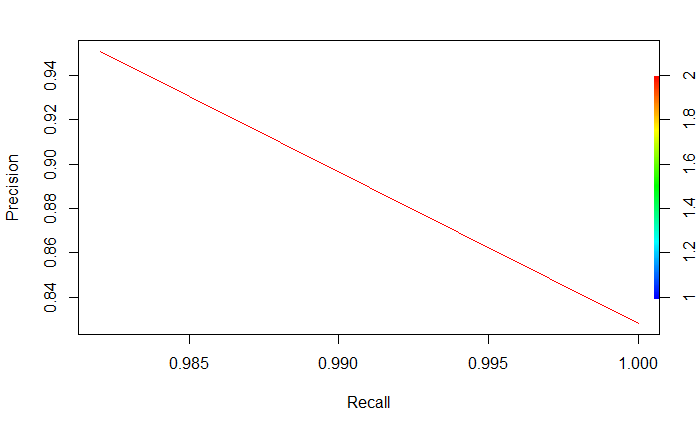
* The above numbers are calculated on the validation sample that was not used for training the model.
* Sensitivity is the percentage of yes’s correctly predicted by the model, while specificity is the percentage of no’s correctly predicted.
* So true detection rate on test data is bad.

## ROC/ AUC



AUC = 0.8684653

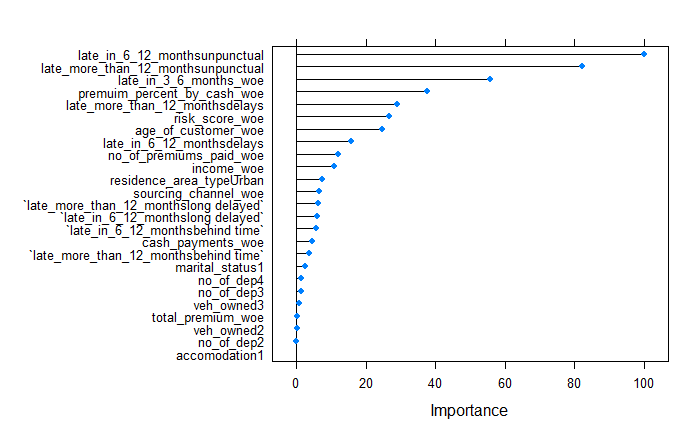
AUC (Area Under Curve) is quite high which also suggests that the model is good.



# Logistic Regression

* Most unimportant variables have p Value greater than significance level of 0.5 in determining default.
* Every 1-unit change in late payments made between 6-12 months will increase the log odds of a customer being a defaulter by 1.49

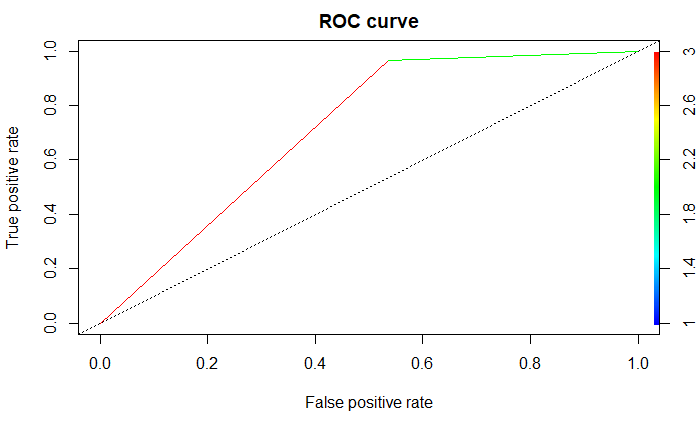
## Variable Importance



* The variables late\_in\_6\_12\_monthsunpunctual, late\_in\_3\_6\_months\_woe, late\_more\_than\_12\_monthsunpunctual, and premium percent by cash woe are considered high in importance.
* The variable total premium\_woe charity is considered least important

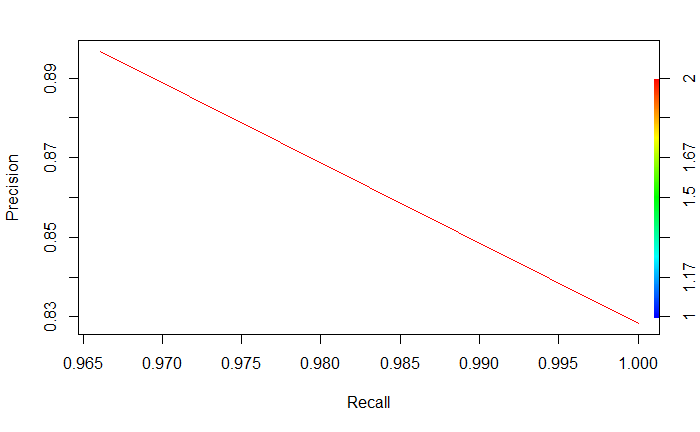
## ROC/AUC

ROC shows the percentage of true positives accurately predicted by a given logit model as the prediction probability cut off is lowered from 1 to 0.



AUC = 0.7146888

AUC (Area Under Curve) is quite high which also suggests that the model is good.



This shows trade off between the true positive rate and the positive predictive value at different probability thresholds. This shows how good the model is at predicting the positives that 5192 customers would default premium.

## Model Performance

The confusion Matrix for the logistic model, predicted on the test data.

|  |  |  |
| --- | --- | --- |
| **Prediction** | **Reference** | |
|  | **YES** | **NO** |
| **YES** | **2739 (TP)** | **968 (FP)** |
| **NO** | **3173 (FN)** | **27572 (TN)** |

* Classification error gives a clear picture of the % of the observations that have been predicted wrongly and correctly by the model. The classifier made 34,452 predictions for customers who are likely to default
* 5192 was predicted as yes and are default customers and 28540 was predicted correctly as 1 No
* TN says we predicted No and they are not defaulters, TP says we predicted Yes and they are defaulters.
* FP says We predicted yes and they are not defaulters, FN says we predicted no but they are defaulters.

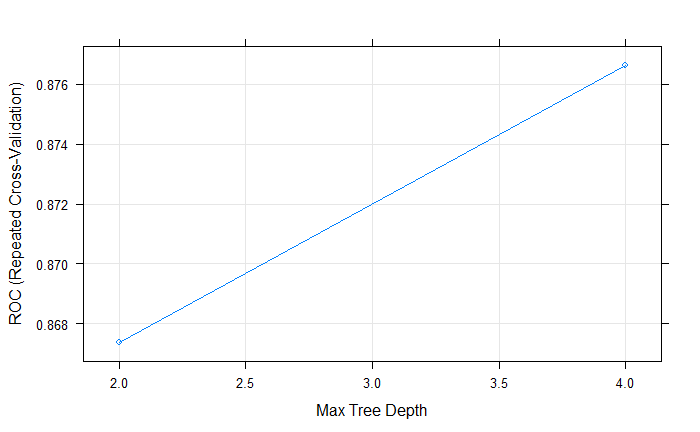
Accuracy – 87%

Sensitivity – 46%

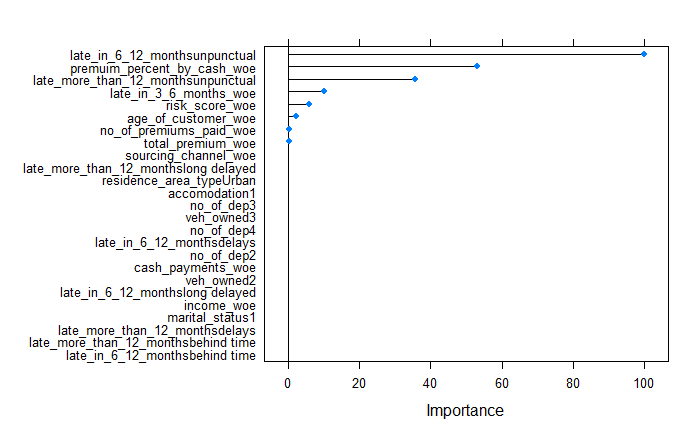
Specificity – 96%

* The above numbers are calculated on the validation sample that was not used for training the model.
* Sensitivity is the percentage of yes’s correctly predicted by the model, while specificity is the percentage of no’s correctly predicted.
* So true detection rate on test data is bad.

# Extreme Gradient Boosting



## Variable Importance



* The variables late in 6 months (unpunctual) and premium percent by cash (from the Woe data) are considered high in importance.
* The variable No of dep, income etc. is considered least important - meaning the least variables not dependent whether customers would default premium or not.

## Model performance

The confusion Matrix for the logistic model, predicted on the test data.

|  |  |  |
| --- | --- | --- |
| **Prediction** | **Reference** | |
|  | **YES** | **NO** |
| **YES** | **4051** | **579** |
| **NO** | **1861** | **27961** |

Classification error gives a clear picture of the % of the observations that have been predicted wrongly and correctly by the model. The classifier made 34,452 predictions for customers who are likely to default

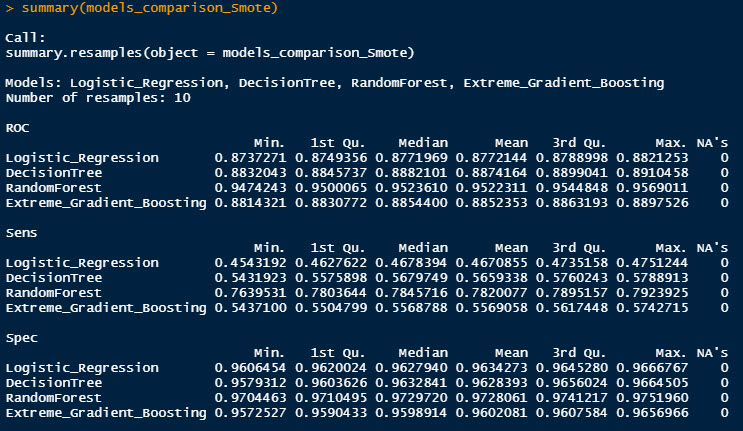
* 5192 was predicted as yes and are default customers and 28540 was predicted correctly as 1 No
* TN says we predicted No and they are not defaulters, TP says we predicted Yes and they are defaulters.
* FP says We predicted yes and they are not defaulters, FN says we predicted no but they are defaulters.

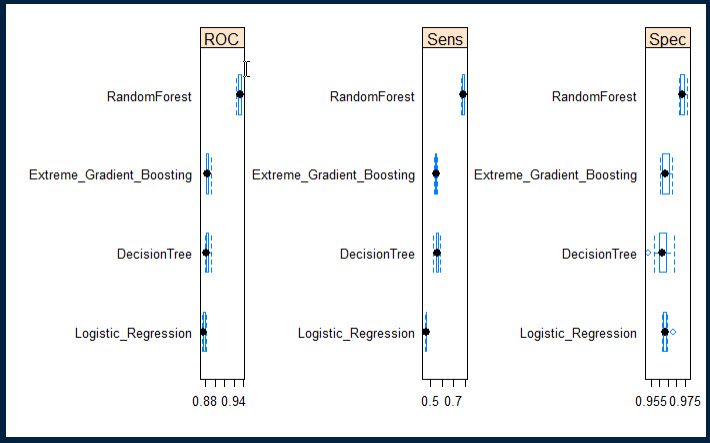
Accuracy – 92%

Sensitivity – 68%

Specificity – 97%

# Model Comparison





The random forest seems to be the most performed algorithm, although 18% of synthetic data has been generated to improve the model performance, it is likely that we are sure that a customer who might default would not default than a customer who would not default and ends up defaulting. We predicted 5192 saying yes to customers that would default premium, holding characteristics like premium percent by cash, late in 6 – 12 months and unpunctual, late in more than 12 months and unpunctual, income, risk score, age of customer, late in 3-6 months – with the help of the important variables used in building the models.

# Business Insights

* Based on the models we see that premium percent by cash, late in 6 – 12 months and unpunctual, late in more than 12 months and unpunctual, income, risk score, age of customer, late in 3-6 months, has been late has been most likely used as a factor for prediction. This explains younger customers and older retired customers are more likely to default because they are not high-income earners.
* Customers who have been late for more than 3 times in 6-12 months still show a high-risk score on the data, also late customers that have a habit of always deposit late payments have no reduced risk score
* Customers pay too much premium in cash, up to 60 million of cash payments have been made so far – this arises the question of alternative mode of payment.

# Recommendations

* Insurance agent should proactively pay attention to customer behaviour. A customer that pays most of their premium in cash only is likely to withdraw from payments because of the stress of going to deposit payment with the insurance agent. It is a recommendation that the insurance company should offer other means of payment to these customers, send reminders for payment on different intervals to ensure that all loopholes between the communication of premium payments has been closed effectively with the customer.
* The value of an insurance company is on retaining customers and a customer who has paid premium twice is considered as testing the waters – it is until a customer pays at least up to 4 times and above before they are considered “real customers”. it is recommended that these types of customers are offered good customer support and follow up, also discounts on premium and premium plans that can entice a customer to feel relax and safe with the company.
* Within these factors, we are more likely to question the risk score as it could be that there was Human error within how the risk scores have been recorded, it is unlikely that customers with good risk scores still fall under the record of those that default premium. We are recommending automation to the insurance company as this would help them keep proper track of customers behaviour with premium payment. A customer who has defaulted premium for over a year, should have a bad risk score or transferred to a follow up list as a blacklisted customer.
* younger customers default because they are not high income earners – the insurance company should look at alternatives or create a policy that can assist the younger customers or customers that fall under cluster 1&2 so they could also take part in the insurance policy.

# Appendix

---

title: "insurance premium"

author: "Deborah"

date: "26/09/2020"

output: pdf\_document

---

```{r setup, include=FALSE}

knitr::opts\_chunk$set(echo = TRUE)

```

#libraries

```{r}

library(dplyr)

library(DataExplorer)

library(janitor)

library(MASS)

library(summarytools)

```

#changing the class to a dataframe

```{r}

colnames(Insurance\_Premium)

class(Insurance\_Premium)

Insurance\_Premium <- as.data.frame(Insurance\_Premium) #making the dataset a dataframe

```

#renaming variables

```{r}

new\_vars <- c("id","Premuim\_%\_by\_cash","Age\_of\_customer", "income", "late in 3-6 months","late in 6-12 months","late more than 12 months","marital\_status", "veh\_owned" , "no\_of\_dep", "accomodation", "risk\_score",

"no\_of\_premiums\_paid","sourcing\_channel" ,"residence\_area\_type" ,"total\_premium", "default")

colnames(Insurance\_Premium) <- new\_vars

colnames(Insurance\_Premium)

Insurance\_Premium <- clean\_names(Insurance\_Premium)

```

#Changing the structure of variables

```{r}

Insurance\_Premium$marital\_status <- as.factor(Insurance\_Premium$marital\_status)

Insurance\_Premium$veh\_owned <- as.factor(Insurance\_Premium$veh\_owned)

Insurance\_Premium$default <- as.factor(Insurance\_Premium$default)

Insurance\_Premium$no\_of\_dep <- as.factor(Insurance\_Premium$no\_of\_dep)

Insurance\_Premium$accomodation <- as.factor(Insurance\_Premium$accomodation)

Insurance\_Premium$residence\_area\_type <- as.factor(Insurance\_Premium$residence\_area\_type)

Insurance\_Premium$sourcing\_channel<- as.factor(Insurance\_Premium$sourcing\_channel)

Insurance\_Premium$late\_in\_3\_6\_months <- as.factor(Insurance\_Premium$late\_in\_3\_6\_months) # better off as factor variables

Insurance\_Premium$late\_in\_6\_12\_months <- as.factor(Insurance\_Premium$late\_in\_6\_12\_months)

Insurance\_Premium$late\_more\_than\_12\_months<- as.factor(Insurance\_Premium$late\_more\_than\_12\_months)

```

#Checking for missing value

```{r}

plot\_intro(Insurance\_Premium) # 59% discrete columns ,41% continuous, all rows are complete, no missing values

View(dfSummary(Insurance\_Premium))

```

#Attaching the variables

```{r}

str(Insurance\_Premium)

attach(Insurance\_Premium)

```

#Checking for missing value

```{r}

plot\_intro(Insurance\_Premium) # 59% discrete columns ,41% continuous, all rows are complete, no missing values

View(dfSummary(Insurance\_Premium))

```

#finding the total no of default

```{r}

sum(default=='1') #74855

sum(default=='1') / 79853 \* 100 # 93% of people are non defaulters

sum(default=='0') #4998

sum(default=='0') /79853 \* 100 # 6% of customers are defaulters

```

#transforming age of customer

```{r}

Insurance\_Premium$age\_of\_customer <- Insurance\_Premium$age\_of\_customer/365 #divide by 365 to get the age in years

head(Insurance\_Premium,10)

Insurance\_Premium$age\_of\_customer <- round(Insurance\_Premium$age\_of\_customer) #change the decimals to whole numbers

```

#Creating a new variable (Cash\_payments)

```{r}

cash\_payments <- premuim\_percent\_by\_cash \* 100 \* total\_premium

View(cash\_payments)

summary(cash\_payments)

#adding cash payments to the dataset

Insurance\_Premium <- cbind(Insurance\_Premium,cash\_payments) # making cash payments a new variable

View(Insurance\_Premium)

```

#breaking scientific number

```{r}

summary(Insurance\_Premium)

options(scipen = 999)

```

#Univarte analysis

```{r}

library(ggplot2)

library(ggpubr)

```

#histogram

```{r}

a1 <- ggplot(Insurance\_Premium,aes(age\_of\_customer)) +geom\_histogram( fill="white",colour="black")+ ggtitle("Age of customer")

i1 <-ggplot(Insurance\_Premium,aes(income)) + geom\_histogram(fill= "white", colour= "black") + ggtitle("Histogram of Income")

tp1 <- ggplot(Insurance\_Premium,aes(total\_premium)) + geom\_histogram(fill= "white", colour= "black") + ggtitle("Histogram Of premium")

np1 <- ggplot(Insurance\_Premium,aes(no\_of\_premiums\_paid)) + geom\_histogram(fill= "white", colour= "black") + ggtitle("NO of premuim paid ")

r1 <- ggplot(Insurance\_Premium,aes(risk\_score)) + geom\_histogram(fill= "white", colour= "black") + ggtitle("Histogram of risk")

c1 <- ggplot(Insurance\_Premium,aes(Cash\_payments)) + geom\_histogram(fill= "white", colour= "black") + ggtitle("Histogram of Cash payment")

```

```{r}

figure\_A <- ggarrange(a1,i1,tp1,np1,r1,c1,ncol=3, nrow= 2)

figure\_A

ggsave(figure\_A, file ="plot.png")

```

#boxplots

```{r}

a2 <- ggplot(Insurance\_Premium,aes(age\_of\_customer )) + geom\_boxplot() + ggtitle("Boxplot of age")

i2 <- ggplot(Insurance\_Premium,aes(income)) + geom\_boxplot() + ggtitle("Boxplot of Income")

tp2 <- ggplot(Insurance\_Premium,aes(total\_premium )) + geom\_boxplot() + ggtitle("boxplot of premuim")

p2 <-ggplot(Insurance\_Premium,aes(no\_of\_premiums\_paid)) + geom\_boxplot(fill= "white", colour= "black") + ggtitle("NO of premuim paid ")

r2 <- ggplot(Insurance\_Premium,aes(risk\_score)) + geom\_boxplot() + ggtitle("Boxplot of risk")

c2 <- ggplot(Insurance\_Premium,aes(Cash\_payments)) + geom\_boxplot() + ggtitle("cash payments")

```

```{r}

figure\_B <- ggarrange(a2,i2,tp2,p2,r2,c2,ncol=3, nrow= 2)

figure\_B

ggsave(figure\_B, file ="plot.png")

```

#Factor variables

```{r}

sc <- ggplot(Insurance\_Premium,aes(sourcing\_channel)) +geom\_bar(fill="dark green",colour ="black") + ggtitle("How customers are sourced")

df <- ggplot(Insurance\_Premium,aes(default)) +geom\_bar(fill="dark green",colour ="black") + ggtitle("Default or not")

ms <- ggplot(Insurance\_Premium,aes(marital\_status)) +geom\_bar(fill="dark green",colour ="black") + ggtitle("Married or not ")

lt <- ggplot(Insurance\_Premium,aes(late\_in\_3\_6\_months)) +geom\_bar(fill="dark green",colour ="black") + ggtitle("late payments 3-6 months")

lp <- ggplot(Insurance\_Premium,aes(late\_in\_6\_12\_months)) +geom\_bar(fill="dark green",colour ="black") + ggtitle("late payments 6 - 12 months")

lg <- ggplot(Insurance\_Premium,aes(late\_more\_than\_12\_months)) +geom\_bar(fill="dark green",colour ="black") + ggtitle("late payments over 12 months")

nd <- ggplot(Insurance\_Premium,aes(no\_of\_dep)) +geom\_bar(fill="dark green",colour ="black") + ggtitle("No of dependents per household")

acc <- ggplot(Insurance\_Premium,aes(accomodation)) +geom\_bar(fill="dark green",colour ="black") + ggtitle("Owned accomodation or not")

veh <- ggplot(Insurance\_Premium,aes(veh\_owned)) +geom\_bar(fill="dark green",colour ="black") + ggtitle("No of vehicles owned")

rt <- ggplot(Insurance\_Premium,aes(residence\_area\_type)) +geom\_bar(fill="dark green",colour ="black") + ggtitle("Residence area")

```

```{r}

figure\_c <- ggarrange(sc,df,ms,lt,lp,lg,ncol=3, nrow= 2)

figure\_c

ggsave(figure\_c, file ="plot.png")

```

```{r}

figure\_d <- ggarrange(nd,acc,veh,rt,ncol=2, nrow= 2)

figure\_d

ggsave(figure\_d, file ="plot.png")

```

#correlation

```{r}

plot\_correlation(Insurance\_Premium, maxcat = 3)

```

#Bivirate analysis

```{r}

ps <-ggplot(Insurance\_Premium,aes(x=total\_premium,y=sourcing\_channel,

colour= sourcing\_channel))+geom\_boxplot()+ ggtitle("Total premium paid per sourcing channel")

npp <- ggplot(Insurance\_Premium,aes(x= total\_premium, y = no\_of\_premiums\_paid,colour = no\_of\_premiums\_paid))+geom\_point()+ ggtitle("Total no of premium paid")

npa <- npp+geom\_point() + stat\_smooth(method = lm)

dfs <-ggplot(Insurance\_Premium, aes(x= default , y= sourcing\_channel, colour= default)) +geom\_boxplot()+ ggtitle("what sourcing channel defaults more")

it <- ggplot(Insurance\_Premium, aes(x= income, y= total\_premium, colour= income)) + geom\_point()+ ggtitle("who pays more premium")

air <- ggplot(Insurance\_Premium, aes(x= income, y= age\_of\_customer,colour= age\_of\_customer)) + geom\_line() + ggtitle("who earns more")

```

```{r}

lrs <- ggplot(Insurance\_Premium,aes(x=late\_in\_3\_6\_months, y= risk\_score,colour = default)) + geom\_point() + ggtitle("risk score & 3-6 month late")

rsl <-ggplot(Insurance\_Premium,aes(x= late\_in\_6\_12\_months, y= risk\_score,colour= default))+ geom\_point() +

ggtitle("risk score & 6-12 month late")

rs12 <- ggplot(Insurance\_Premium,aes(x=late\_more\_than\_12\_months,y=risk\_score,colour=default))+geom\_point() + ggtitle("risk score & more than 12 months late")

cpp <-ggplot(Insurance\_Premium,aes(x=total\_premium,y=Cash\_payments ,colour=default))+geom\_point() + ggtitle("how is premium paid defaulted")

rsc <- ggplot(Insurance\_Premium,aes(x= sourcing\_channel, y = risk\_score, colour = sourcing\_channel)) +geom\_boxplot() + ggtitle("what sourcing channel is high risk ")

prs<-ggplot(Insurance\_Premium,aes(x=total\_premium,y= age\_of\_customer,colour= default))+geom\_point()+ ggtitle("are younger people likely to default")

agd <- prs+geom\_point() + stat\_smooth(method = lm)

```

```{r}

figure\_e <- ggarrange(npa,dfs,it,air,lrs,rsl,ncol=2, nrow= 3)

figure\_e

ggsave(figure\_d, file ="plot.png")

```

```{r}

figure\_f <- ggarrange(rs12,cpp,rsc,agd,ncol=2, nrow= 2)

figure\_f

ggsave(figure\_d, file ="plot.png")

```

#Binning Variables

```{r}

str(Insurance\_Premium)

levels(Insurance\_Premium$late\_in\_3\_6\_months)

levels(Insurance\_Premium$late\_in\_3\_6\_months)[1] <- "never late"

levels(Insurance\_Premium$late\_in\_3\_6\_months)[2:4] <- "unpunctual"

levels(Insurance\_Premium$late\_in\_3\_6\_months)[3:5] <- "delays"

levels(Insurance\_Premium$late\_in\_3\_6\_months)[4:6] <- "long delayed"

levels(Insurance\_Premium$late\_in\_3\_6\_months)[5:8] <- "behind time"

```

```{r}

levels(Insurance\_Premium$late\_in\_6\_12\_months)

levels(Insurance\_Premium$late\_in\_6\_12\_months)[1] <- "never late"

levels(Insurance\_Premium$late\_in\_6\_12\_months)[2:6] <- "unpunctual"

levels(Insurance\_Premium$late\_in\_6\_12\_months)[3:5] <- "delays"

levels(Insurance\_Premium$late\_in\_6\_12\_months)[4:7] <- "long delayed"

levels(Insurance\_Premium$late\_in\_6\_12\_months)[5:8] <- "behind time"

```

```{r}

levels(Insurance\_Premium$late\_more\_than\_12\_months)

levels(Insurance\_Premium$late\_more\_than\_12\_months)[1] <- "never late"

levels(Insurance\_Premium$late\_more\_than\_12\_months)[2:3] <- "unpunctual"

levels(Insurance\_Premium$late\_more\_than\_12\_months)[3:4] <- "delays"

levels(Insurance\_Premium$late\_more\_than\_12\_months)[4:5] <- "long delayed"

levels(Insurance\_Premium$late\_more\_than\_12\_months)[5:7] <- "behind time"

```

#dropping variables

```{r}

Insurance\_Premium <- Insurance\_Premium[-c(1)]

str(Insurance\_Premium)

```

```{r}

levels(Insurance\_Premium$residence\_area\_type) <- c("0", "1")

levels(Insurance\_Premium$sourcing\_channel) <- c("1", "2","3","4","5")

levels(Insurance\_Premium$late\_more\_than\_12\_months) <- c("1", "2","3","4","5")

levels(Insurance\_Premium$late\_in\_6\_12\_months) <- c("1", "2","3","4","5")

levels(Insurance\_Premium$late\_in\_3\_6\_months) <- c("1", "2","3","4","5")

```

#drawing a sample to do a cluster

```{r}

library(caTools)

set.seed(129)

index <- sample.split(Insurance\_Premium, SplitRatio = 0.80)

insur.clust = subset(Insurance\_Premium, index == F)

dim

str(insur.clust)

insur.clust$residence\_area\_type <- as.numeric(insur.clust$residence\_area\_type)

insur.clust$sourcing\_channel<- as.numeric(insur.clust$sourcing\_channel)

insur.clust$marital\_status <- as.numeric(insur.clust$marital\_status)

insur.clust$veh\_owned <- as.numeric(insur.clust$veh\_owned)

insur.clust$default <- as.numeric(insur.clust$default)

insur.clust$no\_of\_dep <- as.numeric(insur.clust$no\_of\_dep)

insur.clust$accomodation <- as.numeric(insur.clust$accomodation)

insur.clust$late\_in\_3\_6\_months <- as.numeric(insur.clust$late\_in\_3\_6\_months) # better off as factor variables

insur.clust$late\_in\_6\_12\_months <- as.numeric(insur.clust$late\_in\_6\_12\_months)

insur.clust$late\_more\_than\_12\_months<- as.numeric(insur.clust$late\_more\_than\_12\_months)

levels(insur.clust$late\_in\_3\_6\_months)

```

#building the cluster

```{r}

library(cluster)

library(factoextra)

fviz\_nbclust(insur.clust,clara,method = "silhouette") #selects 2 as the optimum and right number of clusters

nc<- clara(insur.clust,4,metric="euclidean",stand = FALSE, samples = 5,pamLike = FALSE)

nc

fviz\_cluster(nc)

```

```{r}

### Adding our clusters to the dataset so as to create a customer profile

insur.clust$clusters= nc$cluster

print(insur.clust)

```

```{r}

## customer profile

customersprofiles = aggregate(insur.clust,list(insur.clust$clusters),FUN = "median")

print(customersprofiles)

```

#Changing the structure of variables

```{r}

str(Insurance\_Premium)

```

# Ensure that the target variable is a factor & Rename the levels & Relevel

```{r}

levels(Insurance\_Premium$residence\_area\_type) <- c("Rural", "Urban")

levels(Insurance\_Premium$sourcing\_channel) <- c("A", "B","C","D","E")

levels(Insurance\_Premium$late\_more\_than\_12\_months) <- c("never late","unpunctual","delays" ,"long delayed","behind time")

levels(Insurance\_Premium$late\_in\_6\_12\_months) <- c("never late" , "unpunctual","delays" ,"long delayed","behind time")

levels(Insurance\_Premium$late\_in\_3\_6\_months) <- c("never late" , "unpunctual","delays" ,"long delayed","behind time")

```

#checking important variable

```{r}

library(Information)

```

```{r}

str(Insurance\_Premium)

Insurance\_Premium$default <- as.numeric(Insurance\_Premium$default)

```

```{r}

Insurance\_Premium$default <- recode(Insurance\_Premium$default,"2"= 1L ,"1" =0L)

InformationValue <- create\_infotables(data=Insurance\_Premium, y="default", bins= 20, parallel=TRUE)

```

```{r}

IV\_Value = data.frame(InformationValue$Summary)

InformationValue

```

#using the woe to treat outliers

```{r}

library(scorecard)

data\_filter = var\_filter(Insurance\_Premium, y="default")

```

```{r}

bins1 = woebin(data\_filter , y="default")

```

#replacing dataset

```{r}

Insurance\_Premium1 <- woebin\_ply(Insurance\_Premium,bins1)

View(Insurance\_Premium1)

library(DataExplorer)

```

#plotting the new data

```{r}

boxplot(Insurance\_Premium1)

plot\_boxplot(Insurance\_Premium1, by = "default")

```

#correlation

```{r}

plot\_correlation(Insurance\_Premium1, maxcat = 1)

```

# Ensure that the target variable is a factor & Rename the levels & Relevel

```{r}

str(Insurance\_Premium1)

Insurance\_Premium1$default <- as.factor(Insurance\_Premium1$default)

levels(Insurance\_Premium1$default)

levels(Insurance\_Premium1$default) <- c("yes", "no")

Insurance\_Premium1$default<- relevel(Insurance\_Premium1$default, ref = "yes") # Reference class : 0

```

#split dataset

```{r}

library(caTools)

library(xgboost)

library(DMwR)

library(caret)

library(gbm)

```

#using smote to improve the imbalance

```{r}

table(Insurance\_Premium1$default)

insur\_smote <- SMOTE(default ~ ., Insurance\_Premium1)

table(insur\_smote$default)

insurance\_smote<- rbind(Insurance\_Premium1,insur\_smote)

table(insurance\_smote$default)

```

#divide data into train and test

```{r}

# Divide data in "70:30"

set.seed(100)

Part\_premium\_data <- sample(1:nrow(insurance\_smote), 0.7\*nrow(insurance\_smote))

# Training set

train\_premium\_data <- insurance\_smote[Part\_premium\_data,]

# Test set

test\_premium\_data <- insurance\_smote[-Part\_premium\_data,]

dim(train\_premium\_data) #80387 17

dim(test\_premium\_data) #34452 17

table(train\_premium\_data$default) # 0 - 14080 , 1 - 66307 on the train set

table(test\_premium\_data$default) # 0 - 5912 , 1 - 28540 on the test set

```

# Setting up the general parameters for training multiple models

```{r}

# Define the training control

fitControl <- trainControl(

method = 'repeatedcv', # k-fold cross validation

number = 3, # number of folds or k

repeats = 1, # repeated k-fold cross-validation

allowParallel = TRUE,

classProbs = TRUE,

summaryFunction=twoClassSummary# should class probabilities be returned

)

```

#Setting the control parameters

```{r}

set.seed(149)

library(rpart)

insurance\_ctrl\_parameter = rpart.control(minsplit=100, minbucket = 20, cp = 0.0021 , xval = 20,trControl = fitControl)

```

#model 1 Building the CART model

```{r}

insurance\_Cart\_Model2<- rpart(formula = default~., data = train\_premium\_data,

method = "class",control = insurance\_ctrl\_parameter)

```

```{r}

insurance\_Cart\_Model<- train(default ~ ., data = train\_premium\_data,

method = "rpart",

minbucket = 301,

cp = 0.05,

tuneLength = 5,

trControl = fitControl)

```

#plotting cart model

```{r}

library(rattle)

fancyRpartPlot(insurance\_Cart\_Model)

plot(insurance\_Cart\_Model)

fancyRpartPlot(insurance\_Cart\_Model$finalModel,digits = 5 )

varImp(insurance\_Cart\_Model)

```

==================== CHECK\_PERFORMANCE : USING\_ROC\_&\_PR\_CURVES ==================

```{r}

cart\_predictions\_prob <- predict(insurance\_Cart\_Model, newdata = test\_premium\_data, type = "raw")

# Creating the prediction object using ROCR library

library(ROCR)

cart\_pred\_obj = prediction(as.numeric(cart\_predictions\_prob) , as.numeric (test\_premium\_data$default))

# Plotting the ROC curve

roc\_LR = performance(cart\_pred\_obj, "tpr", "fpr")

plot(roc\_LR, colorize=TRUE,main="ROC curve") +abline(a=0,b=1,lty=3)

plot(roc\_LR, main="ROC curve") +abline(a=0,b=1,lty=3)

# Plotting the PR curve

precision\_recall\_LR<- performance(cart\_pred\_obj, "ppv", "tpr")

plot(precision\_recall\_LR, xlab = "Recall", ylab = "Precision",colorize=TRUE)+abline(a = 0, b = 1, lty = 2)

# Computing the area under the curve

auc = performance(cart\_pred\_obj,"auc");

auc = as.numeric(auc@y.values)

auc

```

```{r}

library(caret)

cart\_predictions\_test <- predict(insurance\_Cart\_Model2, newdata = test\_premium\_data, type = "class")

confusionMatrix(as.factor(cart\_predictions\_test),as.factor(test\_premium\_data$default))

```

```{r}

library(caret)

cart\_predictions\_train <- predict(insurance\_Cart\_Model, newdata = train\_premium\_data, type = "class")

confusionMatrix(as.factor(cart\_predictions\_train),as.factor(train\_premium\_data$default))

```

#Randomforests model

```{r}

set.seed(100)

library(randomForest)

# Model\_5 : Random Forest

insurance\_rndForest<- train(default ~ ., data = train\_premium\_data,

method = "rf",

ntree = 101,

maxdepth = 5,

tuneLength = 5,

trControl = fitControl)

##Print the model to see the OOB and error rate

print(insurance\_rndForest)

plot(varImp(insurance\_rndForest,top = 20))

plot(insurance\_rndForest)

```

#checking model performance using ROC Curve

```{r}

rf\_predictions\_prob <- predict(insurance\_rndForest, newdata = test\_premium\_data, type = "raw")

# Creating the prediction object using ROCR library

library(ROCR)

rf\_pred\_obj = prediction(as.numeric(rf\_predictions\_prob) , as.numeric(test\_premium\_data$default))

# Plotting the ROC curve

roc\_rf = performance(rf\_pred\_obj, "tpr", "fpr")

plot(roc\_rf, colorize=TRUE, main="ROC curve") +abline(a=0,b=1,lty=3)

# Plotting the PR curve

precision\_recall\_rf<- performance(rf\_pred\_obj, "ppv", "tpr")

plot(precision\_recall\_rf, xlab = "Recall", ylab = "Precision",colorize=TRUE) + abline(a = 0, b = 1, lty = 2)

# Computing the area under the curve

auc = performance(rf\_pred\_obj,"auc");

auc = as.numeric(auc@y.values)

auc

```

```{r}

library(caret)

rd\_predictions\_test <- predict(insurance\_rndForest, newdata = test\_premium\_data, type = "raw")

confusionMatrix(as.factor(rd\_predictions\_test),as.factor(test\_premium\_data$default))

```

```{r}

library(caret)

rd\_predictions\_train <- predict(insurance\_rndForest, newdata = train\_premium\_data, type = "class")

confusionMatrix(as.factor(rd\_predictions\_train),as.factor(train\_premium\_data$default))

```

```{r}

lr\_model <- train(default~., data = train\_premium\_data,

method = "glm",

family = "binomial",trControl = fitControl)

summary(lr\_model)

plot(varImp(lr\_model, main = lr\_model))

```

==================== CHECK\_PERFORMANCE : USING\_ROC\_&\_PR\_CURVES ==================

```{r}

lr\_predictions\_prob <- predict(lr\_model, newdata = test\_premium\_data, type = "raw")

# Creating the prediction object using ROCR library

library(ROCR)

lr\_pred\_obj = prediction(as.numeric(lr\_predictions\_prob) , as.numeric (test\_premium\_data$default))

# Plotting the ROC curve

roc\_LR = performance(lr\_pred\_obj, "tpr", "fpr")

plot(roc\_LR, colorize=TRUE,main="ROC curve") +abline(a=0,b=1,lty=3)

plot(roc\_LR, main="ROC curve") +abline(a=0,b=1,lty=3)

# Plotting the PR curve

precision\_recall\_LR<- performance(lr\_pred\_obj, "ppv", "tpr")

plot(precision\_recall\_LR, xlab = "Recall", ylab = "Precision",colorize=TRUE)+abline(a = 0, b = 1, lty = 2)

# Computing the area under the curve

auc = performance(lr\_pred\_obj,"auc");

auc = as.numeric(auc@y.values)

auc

```

============================= CHECK\_PERFORMANCE : USING\_CONFUSION\_MATRIX =================

```{r}

library(caret)

lr\_predictions\_test <- predict(lr\_model, newdata = test\_premium\_data, type = "raw")

confusionMatrix(as.factor(lr\_predictions\_test),as.factor(test\_premium\_data$default))

```

```{r}

library(caret)

lr\_predictions\_train <- predict(lr\_model, newdata = train\_premium\_data, type = "raw")

confusionMatrix(as.factor(lr\_predictions\_train),as.factor(train\_premium\_data$default))

```

```{r}

table(as.factor(train\_premium\_data$default),as.factor(lr\_predictions\_train))

```

# Gradient Boosting Machines

```{r}

gbm\_model <- train(default ~ ., data = train\_premium\_data,

method = "gbm",

verbose = FALSE,

trControl = fitControl)

```

```{r}

library(caret)

gbm\_predictions\_test <- predict( gbm\_model, newdata = test\_premium\_data, type = "raw")

confusionMatrix(as.factor(gbm\_predictions\_test),as.factor(test\_premium\_data$default))

```

```{r}

library(caret)

gbm\_predictions\_train <- predict( gbm\_model, newdata = train\_premium\_data, type = "raw")

confusionMatrix(as.factor(gbm\_predictions\_train),as.factor(train\_premium\_data$default))

```

# Xtreme Gradient boosting Machines

```{r}

set.seed(123)

fitControl <- trainControl(

method = 'repeatedcv', # k-fold cross validation

number = 3, # number of folds or k

repeats = 1, # repeated k-fold cross-validation

allowParallel = TRUE,

classProbs = TRUE,

summaryFunction=twoClassSummary)

xgb.grid <- expand.grid(nrounds = 1000,

eta = c(0.4),

max\_depth = c(5),

gamma = 0, #default=0

colsample\_bytree = 1, #default=1

min\_child\_weight = 1, #default=1

subsample = 1 #default=1

)

xgb\_model <-train(default~.,

data=train\_premium\_data,

method="xgbTree",

trControl=fitControl,

tuneGrid=xgb.grid,

metric="ROC",

verbose=FALSE,

nthread = 2

)

```

```{r}

plot(xgb\_model)

plot(varImp(xgb\_model))

```

```{r}

library(caret)

gb\_predictions\_test <- predict( xgb\_model, newdata = test\_premium\_data, type = "raw")

confusionMatrix(as.factor(gb\_predictions\_test),as.factor(test\_premium\_data$default))

```

```{r}

library(caret)

gb\_predictions\_train <- predict( xgb\_model, newdata = train\_premium\_data, type = "raw")

confusionMatrix(as.factor(gb\_predictions\_train),as.factor(train\_premium\_data$default))

```

#model comparison

```{r}

# Compare model performances using resample()

models\_to\_compare <- resamples(list(logistic\_Regression= lr\_model, DecisionTree= insurance\_Cart\_Model, Extreme\_Gradient\_Boosting =xgb\_model,randomForest = insurance\_rndForest ))

# Summary of the models performances

summary(models\_to\_compare)

```

# Draw box plots to compare models

```{r}

scales <- list(x=list(relation="free"), y=list(relation="free"))

bwplot(models\_to\_compare, scales=scales)

```