VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

Fall Semester 2021-2022

CSE3505 – Foundations of Data Analytics

Data Analysis on medical insurance cost using R

Faculty- Dr.S.Brindha

Batch Members

1. Kalyani G -19MIA1064
2. Gaurav Trivedi -19MIA1077
3. Diya Harish - 19MIA1107

Table of Contents

Abstract

The innumerous risks associated with any kind of transactions or payments must be eliminated or at least reduced. This is where the role of insurance comes in. Medical field is an area of highly crucial transactions and insurance inthis field is inevitable for the safety and security of one's close kin. This project aims to build a machine learning model that predicts the minimum cost that a person should pay for their medical insurance based on a number of factors including their medical histories. Insurance is a policy that eliminates or decreases loss costs occurred by various risks. Various factors influence the costof insurance. These considerations contribute to the insurance policy formulation. Machine learning (ML) for the insurance industry sector can make the wording of insurance policies more efficient. This study demonstrates how different models of regression can forecast insurance costs. And we will compare the results of models, for example, Random Forest Regressor and various other regression algorithms. The costs regarding the insurances associated with medical field have been of utmost concern as it is quite high in spite of it being inevitable. This project aims to determine the factors that affect the medical insurance costs using various ML algorithms and hence help to takedecisions on cutting costs.

Introduction

People's health is the number one investment each of us must give priority to. All of people's lives, they concentrate on being in good health, be it in the form of staying fit by exercising or doing regular check-ups. The healthcare sector produces a very large amount of data related to patients, diseases, and diagnosis,but since it has not been analyzed properly, it does not provide the significance which it holds along with the patient healthcare cost.
A health insurance policy offers coverage for any future medical expenses of the customer. This is an agreement between the insurance company and the customer where the former agrees to guarantee payment/compensation˜ for medical costs if the latter is injured/ill in the future, leading to hospitalisation.

In the insurance sector, ML can help enhance the efficiency of policy wording. In healthcare, ML algorithms are particularly good at predicting high-cost, high-need patient expenditures. ML can be categorized into three different types, as shown in Figure. These types are supervised machine learning (i.e., a task- driven approach) used for classification/regression and all data labeled; unsupervised machine learning (i.e., a data-driven approach) used for clusteringand all data unlabeled; and reinforcement learning (i.e., learning from mistakes)used for decision making.

Literature Review

In the book "*Hardship financing of healthcare among rural poor in Orissa, India"* by *Erika Binnendijk1, Ruth Koren2 and David M Dror1,3,* it is said thatthe analysis of research shows that most of the people under the survey use their current income to pay for healthcare expenses followed by burning up their savings. This unexpected ranges disturbs the family planning and thus result into miss conception of costly healthcare. The study also shows that that hardship financing occurs not only in cases of expensive hospitalizations (40%)but also in many cases of expenditures for outpatient (23%) and maternity care (25%). Taking into account that the frequency of outpatient utilization is much higher, many more people actually face hardship financing due to outpatient care than due to inpatient care.

In another book called "*Cost of illness: Evidence from a study in five resource-poor locations in India"* by *David M. Dror, Olga van Putten- Rademaker & Ruth Koren,* it was found that The ratio ranged from 0.38 to 1.2, signifying thatfor half the population in each location, the cost of one illness episode ranged from 38 to 120 per cent of monthly income per person. As in previous observations, the differences across locations were considerable also in respectof this measure of financial exposure. This difference in the ratios originated from the combined effect of different levels of costs of healthcare services and different income levels. For instance, the higher ratio in location I compared tolocation II was mainly due to the different cost of healthcare, whereas income levels were quite similar. On the other hand, the higher ratio in location V compared to IV was associated mainly with lower income in V.

In the system proposed by Ranjodh Singh and others in 2019, this system takespictures of the damaged car as inputs and produces relevant details, such as costs of repair to decide on the amount of insurance claim and locations of damage. Thus the predicted car insurance claim was not taken into account inthe present analysis but was focused on calculating repair costs.

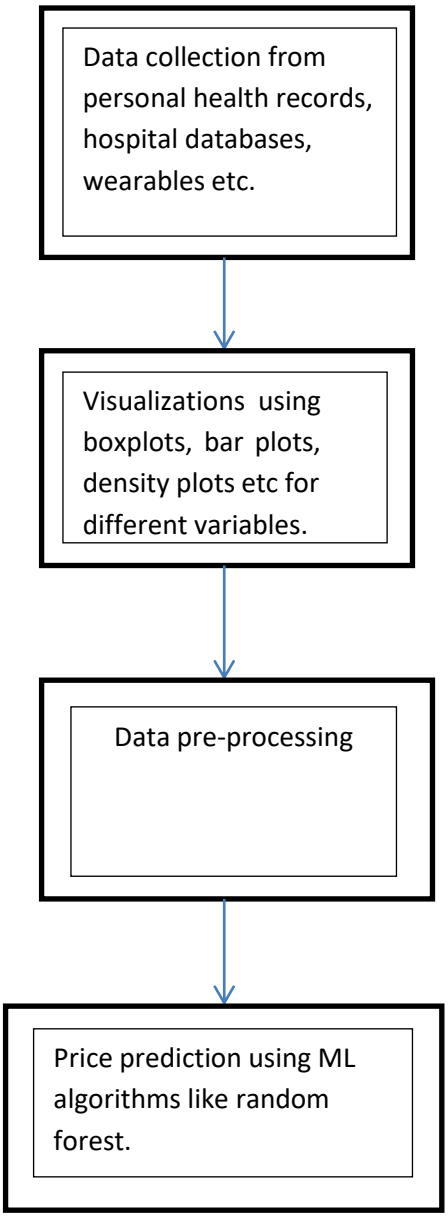
"Health Insurance Coverage and Its Impact on Medical Cost: Observations from the Floating Population in China" by *Yinjung Zao et.al*: In this paper, they report empirical observations made in a survey recently conducted on insurance coverage and medical cost. A great discrepancy of insurance coverage exists between the floating population and the general population. Demographic and personal characteristics are found as associated with insurance coverage. The findings may have important implications and can assist the development of intervention programs to further increase coverage and effect. The analysis of medical cost leads to two main observations. The first is that insurance coverage is not associated with gross and OOP medical costs. The second is the distinct associations with medical cost for the floating population.

Problem Description

Many people are unwilling or unable to pay for their medical insurance due tovarious reasons like lack of knowledge, poverty or other personal matters. To reduce this and make more people take up insurance for their own safety and security, different methods are tested to find the least cost possible for each customer based on their personal health reports and past data.

In this project we use the medical insurance premium dataset along with features from the insurance dataset to perform the data analysis using R language. The project aims to determine the factors that affect the medical insurance costs using various ML algorithms and hence help to make decisions on cutting down high insurance costs

Workflow


Data collection from personal health records, hospital databases, wearables etc.
Visualizations using boxplots, bar plots, density plots etc for different variables.
Data pre-processing
Price prediction using ML algorithms like random forest.

Dataset

The dataset used in this project is from Kaggle and it contains almost 987 records with 11 columns including age, presence of diabetes, blood pressure, chronic diseases, transplants, height, weight, known allergies, history of cancer in family, number of major surgeries and premium price.

Age	Diabetes	BloodPres	AnyTransp	AnyChroni	Height	Weight	KnownAlle	HistoryOf	NumberOf	PremiumPrice
45	0	0	0	0	155	57	0	0	0	25000
60	1	0	0	0	180	73	0	0	0	29000
36	1	1	0	0	158	59	0	0	1	23000
52	1	1	0	1	183	93	0	0	2	28000
38	0	0	0	1	166	88	0	0	1	23000
30	0	0	0	0	160	69	1	0	1	23000
33	0	0	0	0	150	54	0	0	0	21000
23	0	0	0	0	181	79	1	0	0	15000
48	1	0	0	0	169	74	1	0	0	23000
38	0	0	0	0	182	93	0	0	0	23000
60	0	1	0	0	175	74	0	0	2	28000
66	1	0	0	0	186	67	0	0	0	25000
24	0	0	0	0	178	57	1	0	1	15000
46	0	1	0	0	184	97	0	0	0	35000
18	0	0	1	0	150	76	0	0	1	15000
38	0	0	0	0	160	68	1	0	1	23000
42	0	0	0	1	149	67	0	0	0	30000
38	1	0	0	0	154	82	0	0	0	23000
57	1	0	0	0	156	61	0	0	0	25000
21	0	1	0	0	186	97	0	0	0	15000
49	1	0	0	0	160	97	0	0	2	28000
20	1	0	0	0	181	81	0	0	0	15000
35	0	0	0	0	163	92	0	0	1	32000
35	0	1	0	0	175	83	0	0	1	23000
53	0	1	0	0	151	97	0	1	1	35000
31	0	0	0	0	172	57	0	0	0	21000
22	0	0	1	0	151	97	0	0	0	15000
60	0	1	0	0	151	88	0	0	2	28000
30	0	0	0	1	162	73	1	0	0	23000

age	sex	bmi	children	smoker	region	expenses
19	female	27.9	0	yes	southwest	16884.92
18	male	33.8	1	no	southeast	1725.55
28	male	33	3	no	southeast	4449.46
33	male	22.7	0	no	northwest	21984.47
32	male	28.9	0	no	northwest	3866.86
31	female	25.7	0	no	southeast	3756.62
46	female	33.4	1	no	southeast	8240.59
37	female	27.7	3	no	northwest	7281.51
37	male	29.8	2	no	northeast	6406.41
60	female	25.8	0	no	northwest	28923.14
25	male	26.2	0	no	northeast	2721.32

Methodology

Reading data and performing exploratory data analysis. This involves visualizations.

In [30]:

```
data <- read.csv("Medicalpremium.csv")
head(data)
```

A data.frame: 6 × 11

	Age	Diabetes	BloodPressureProblems	AnyTransplants	AnyChronicDiseases	Height	Weight	KnownAllergies	HistoryOfC
	<int>	<int>	<int>	<int>	<int>	<int>	<int>	<int>	
1	45	0	0	0	0	155	57	0	
2	60	1	0	0	0	180	73	0	
3	36	1	1	0	0	158	59	0	
4	52	1	1	0	1	183	93	0	
5	38	0	0	0	1	166	88	0	
6	30	0	0	0	0	160	69	1	

In [31]:

```
str(data)
```

```
'data.frame': 986 obs. of  11 variables:
 $ Age                   : int  45 60 36 52 38 30 33 23 48 38 ...
 $ Diabetes              : int  0 1 1 1 0 0 0 0 1 0 ...
 $ BloodPressureProblems : int  0 0 1 1 0 0 0 0 0 0 ...
 $ AnyTransplants        : int  0 0 0 0 0 0 0 0 0 0 ...
 $ AnyChronicDiseases    : int  0 0 0 1 1 0 0 0 0 0 ...
 $ Height                : int  155 180 158 183 166 160 150 181 169 182 ...
 $ Weight                : int  57 73 59 93 88 69 54 79 74 93 ...
 $ KnownAllergies        : int  0 0 0 0 0 1 0 1 1 0 ...
 $ HistoryOfCancerInFamily: int  0 0 0 0 0 0 0 0 0 0 ...
 $ NumberOfMajorSurgeries : int  0 0 1 2 1 1 0 0 0 0 ...
 $ PremiumPrice          : int  25000 29000 23000 28000 23000 23000 21000 15000 23000 23
000 ...
```

In [32]:

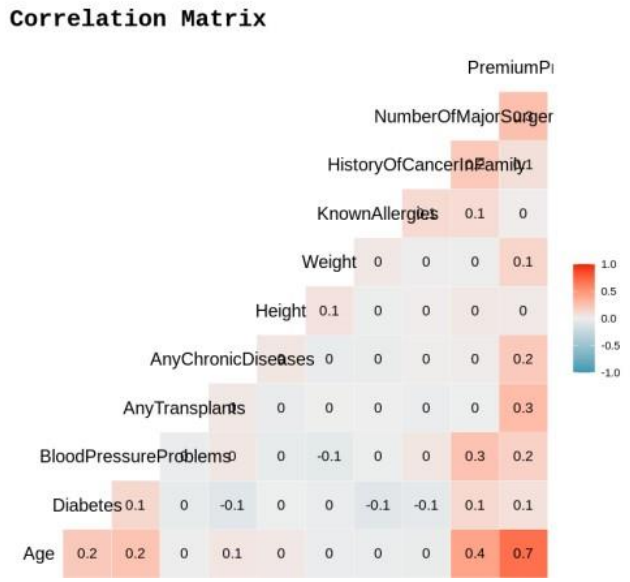
```
dim(data)
```

986 · 11

Plotting Correlattion Matrix

In [33]:

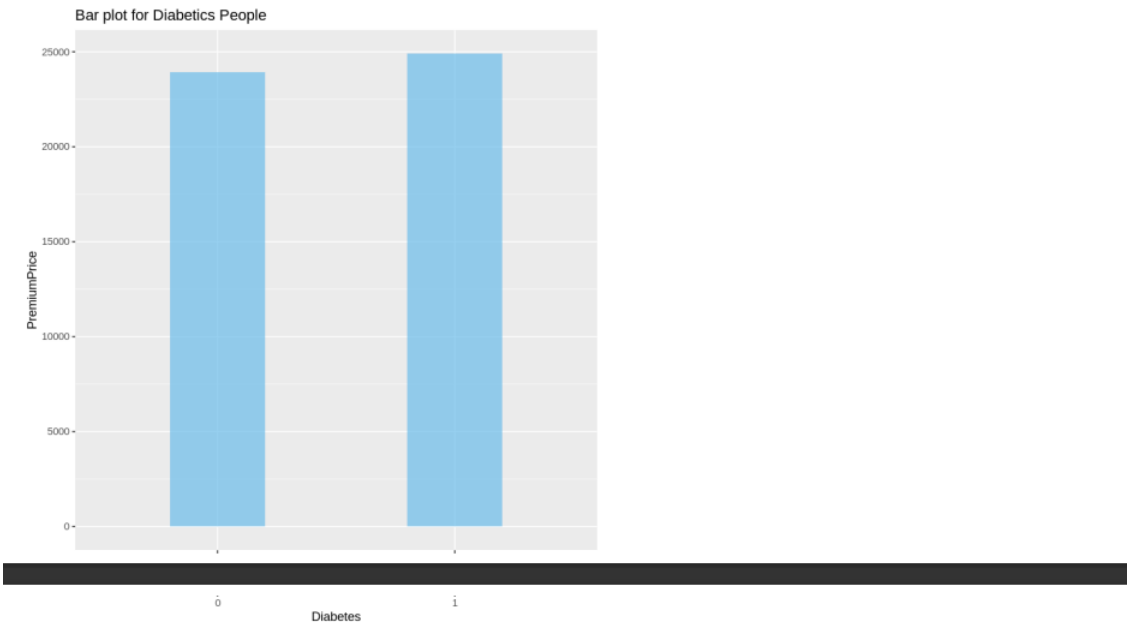
```
ggcorr(data, label = T, color = "black", size = 5)+
  labs(title = "Correlation Matrix")+
  theme(plot.title = element_text(family = "Roboto Condensed", size = 19, face = "bold",
vjust = 1),
        plot.subtitle = element_text(family = "Roboto Condensed", size = 16,vjust = 0))
```



Plot to compare the mean premium for Diabetic paitents and others

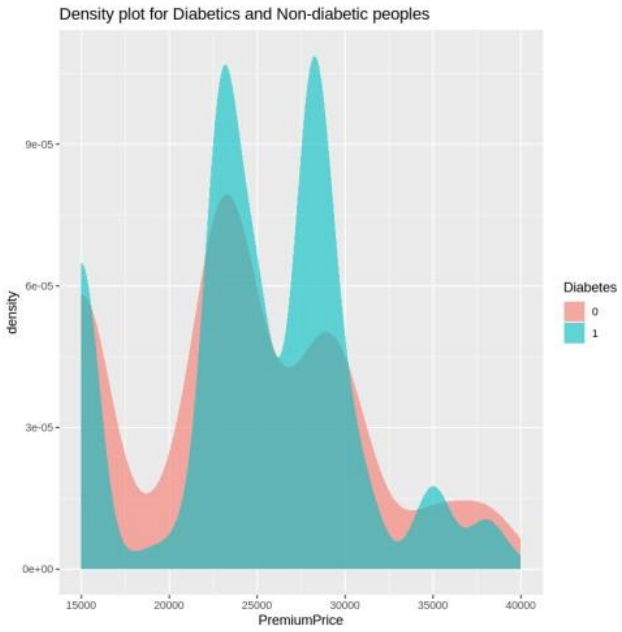
In [37]:

```
data %>%
  select(Diabetes,PremiumPrice) %>%
  group_by(Diabetes) %>%
  summarise( PremiumPrice = mean(PremiumPrice)) %>%
  ggplot(.,aes(Diabetes,PremiumPrice))+
  geom_bar(stat = "identity",width = 0.4, fill = "#56B4E9", alpha = 0.6)+
  labs(title = "Bar plot for Diabetics People")
```



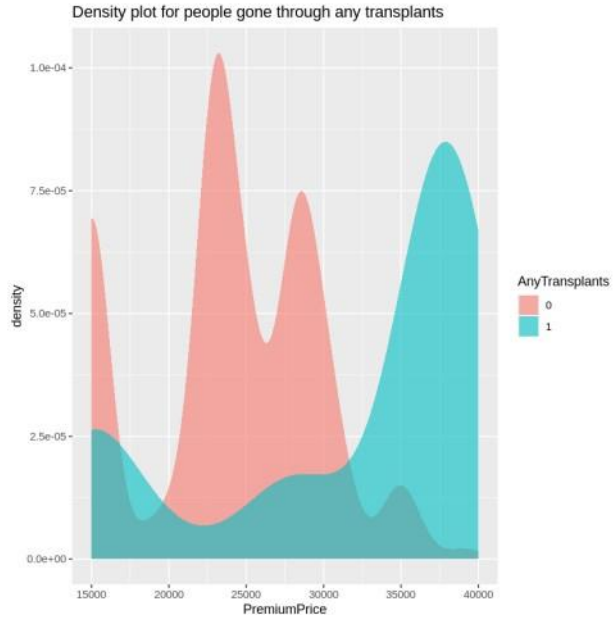
In [38]:

```
ggplot(data, aes(PremiumPrice))+
  geom_density(aes(fill = Diabetes), color = NA, alpha = 0.6)+
  labs(title = "Density plot for Diabetics and Non-diabetic peoples")
```



In [42]:

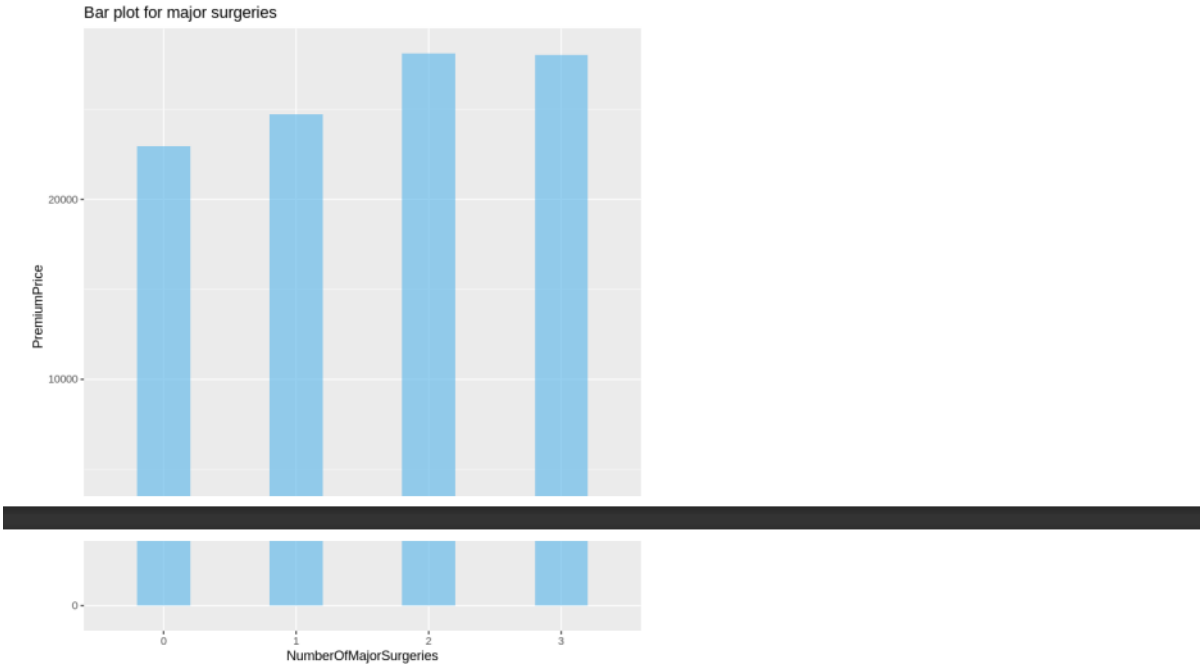
```
ggplot(data, aes(PremiumPrice))+
  geom_density(aes(fill = AnyTransplants), color = NA, alpha = 0.6)+
  labs(title = "Density plot for people gone through any transplants")
```



Plot to compare the mean premium for Number of surgeries

In [49]:

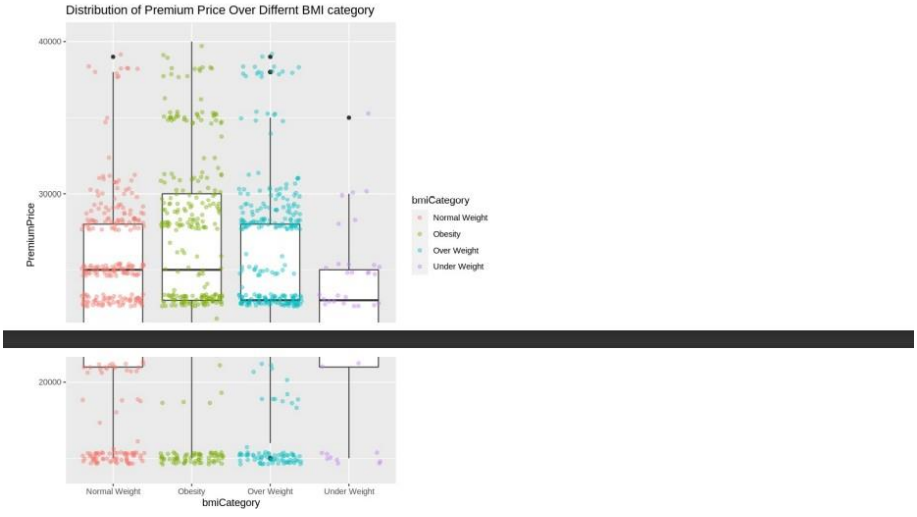
```
data %>%
  select(NumberOfMajorSurgeries,PremiumPrice) %>%
  group_by(NumberOfMajorSurgeries) %>%
  summarise( PremiumPrice = mean(PremiumPrice)) %>%
  ggplot(.,aes(NumberOfMajorSurgeries,PremiumPrice))+
  geom_bar(stat = "identity",width = 0.4, fill = "#56B4E9", alpha = 0.6)+
  labs(title = "Bar plot for major surgeries")
```



Plot to visulaize distibution of premium price over different BMI category

In [51]:

```
data %>%
  mutate(bmiCategory = str_to_title(bmiCategory)) %>%
  ggplot(aes(bmiCategory, PremiumPrice))+
  geom_boxplot()+
  geom_jitter(aes(color = bmiCategory),alpha = 0.4)+
  labs(title = "Distribution of Premium Price Over Differnt BMI category")
```



Data preprocessing is done to convert the data into a form suitable for prediction.

Converting categorical values to numeric values

In [34]:

```
data$Diabetes <- as.factor(data$Diabetes)
data$BloodPressureProblems <- as.factor(data$BloodPressureProblems)
data$AnyTransplants <- as.factor(data$AnyTransplants)
data$AnyChronicDiseases <- as.factor(data$AnyChronicDiseases)
data$KnownAllergies <- as.factor(data$KnownAllergies)
data$HistoryOfCancerInFamily <- as.factor(data$HistoryOfCancerInFamily)
data$NumberOfMajorSurgeries <- as.factor(data$NumberOfMajorSurgeries)
str(data)
```

```
'data.frame': 986 obs. of  11 variables:
 $ Age                     : int  45 60 36 52 38 30 33 23 48 38 ...
 $ Diabetes                : Factor w/ 2 levels "0","1": 1 2 2 2 1 1 1 1 2 1 ...
 $ BloodPressureProblems   : Factor w/ 2 levels "0","1": 1 1 2 2 1 1 1 1 1 1 ...
 $ AnyTransplants          : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
 $ AnyChronicDiseases      : Factor w/ 2 levels "0","1": 1 1 1 2 2 1 1 1 1 1 ...
 $ Height                  : int  155 180 158 183 166 160 150 181 169 182 ...
 $ Weight                  : int  57 73 59 93 88 69 54 79 74 93 ...
 $ KnownAllergies          : Factor w/ 2 levels "0","1": 1 1 1 1 1 2 1 2 2 1 ...
 $ HistoryOfCancerInFamily: Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
 $ NumberOfMajorSurgeries : Factor w/ 4 levels "0","1","2","3": 1 1 2 3 2 2 1 1 1 1 ...
 $ PremiumPrice            : int  25000 29000 23000 28000 23000 23000 21000 15000 23000 23
000 ...
```

Adding BMI column to the data

In [35]:

```
data$bmi <- 10000*(data$Weight/(data$Height)^2)
```

Labelling BMI Values

In [36]:

```
data <- data %>%
  mutate( bmiCategory = case_when(
    bmi<18.49999 ~ "under weight",
    bmi>18.5 & bmi<24.99999 ~ "normal weight",
    bmi>25 & bmi<29.99999 ~ "over weight",
    bmi>30 ~ "obesity"
  ))
```

In [52]:

```
data$PremiumPrice <- as.factor(data$PremiumPrice)
summary(data)
```

```
      Age        Diabetes BloodPressureProblems AnyTransplants
 Min.   :18.00   0:572    0:524                 0:931
 1st Qu.:30.00   1:414    1:462                 1: 55
 Median :42.00
 Mean   :41.75
 3rd Qu.:53.00
 Max.   :66.00

 AnyChronicDiseases     Height          Weight       KnownAllergies
 0:808              Min.   :145.0   Min.   : 51.00   0:774
 1:178              1st Qu.:161.0   1st Qu.: 67.00   1:212
                    Median :168.0   Median : 75.00
                    Mean   :168.2   Mean   : 76.95
                    3rd Qu.:176.0   3rd Qu.: 87.00
                    Max.   :188.0   Max.   :132.00

 HistoryOfCancerInFamily NumberOfMajorSurgeries  PremiumPrice      bmi
 0:870                   0:479                  23000  :249   Min.   :15.16
 1:116                   1:372                  15000  :202   1st Qu.:23.39
                         2:119                  28000  :132   Median :27.16
                         3: 16                  25000  :103   Mean   :27.46
                                                29000  : 72   3rd Qu.:30.76
                                                30000  : 47   Max.   :50.00
                                                (Other):181

 bmiCategory
 Length:986
 Class :character
 Mode  :character
```

Machine algorithm like random forest is used to predict the premium insurance price and the accuracy is found.
Random Forest Regressor Random Forest is an ensemble technique capable of performing both regression and classification tasks. The basic idea behind this is to combine multiple decision trees in determining the final output rather than relying on individual decision trees. Random Forest has multiple decision trees as base learning models.Random Forest Regression is a supervised learning algorithm that uses ensemble learning method for regression. Ensemble learning method is a technique that combines predictions from multiple machine learning algorithms to make a more accurate prediction than a single model.

Linear regression algorithm shows a linear relationship between a dependent (y) and one or more independent (y) variables, hence called as linear regression. Since linear regression shows the linear relationship, which means it finds how the value of the dependent variable is changing according to the value of the independent variable. The linear regression model provides a sloped straight line representing the relationship between the variables.

Splitting the train and test data

```
[ ]  sample <- sample.split(data$PremiumPrice, SplitRatio = 0.75)
     train <- subset(data, sample == TRUE)
     test <- subset(data, sample == FALSE)

     dim(train)
     dim(test)
```

```
743 · 13
243 · 13
```

Building a random forest model using train data

```
[ ]  rf <- randomForest(PremiumPrice ~ .,data=train)
     print(rf)
```

```
Call:
 randomForest(formula = PremiumPrice ~ ., data = train)
               Type of random forest: classification
                     Number of trees: 500
No. of variables tried at each split: 3

        OOB estimate of  error rate: 10.5%
Confusion matrix:
      15000 16000 17000 18000 19000 20000 21000 22000 23000 24000 25000 26000
15000   152     0     0     0     0     0     0     0     0     0     0     0
16000     2     0     0     0     0     0     0     0     0     0     0     0
17000     0     0     0     0     0     0     0     0     0     0     0     0
18000     1     0     0     0     0     0     0     0     0     0     0     0
19000     0     0     0     0    11     0     0     0     0     0     0     0
20000     0     0     0     0     0     0     0     0     0     0     0     0
21000     1     0     0     0     1     0     9     0     8     0     0     0
22000     0     0     0     0     0     0     0     0     0     0     0     0
23000     0     0     0     0     0     0     0     0   187     0     0     0
24000     2     0     0     0     0     0     0     0     0     0     0     0
25000     0     0     0     0     1     0     0     0     2     0    70     0
26000     2     0     0     0     0     0     0     0     2     0     0     0
27000     0     0     0     0     0     0     0     0     0     0     0     0
28000     0     0     0     0     0     0     0     0     0     0     0     0
```

	27000	28000	29000	30000	31000	32000	34000	35000	36000	38000	39000	40000
29000	0	0	0	0	0	0	0	0	0	0	0	0
30000	0	0	0	0	1	0	0	0	3	0	2	0
31000	0	0	0	0	0	0	0	1	1	0	0	0
32000	1	0	0	0	0	0	0	0	2	0	0	0
34000	0	0	0	0	0	0	0	0	2	0	0	0
35000	0	0	0	0	0	0	0	0	4	0	2	0
36000	1	0	0	0	0	0	0	0	0	0	0	0
38000	0	0	0	0	0	0	1	0	0	0	0	0
39000	4	0	0	0	0	0	0	0	0	0	0	0
40000	0	0	0	0	0	0	0	0	0	0	0	0
15000	0	0	0	0	0	0	0	0	0	0	0	0
16000	0	0	0	0	0	0	0	0	0	0	0	0
17000	0	0	0	0	0	0	0	0	0	1	0	0
18000	0	0	0	1	0	0	0	0	0	0	0	0
19000	0	0	0	0	0	0	0	0	0	0	0	0
20000	0	0	1	0	0	0	0	0	0	0	0	0
21000	0	0	0	0	0	0	0	0	0	1	0	0
22000	0	0	0	0	1	0	0	0	0	0	0	0
23000	0	0	0	0	0	0	0	0	0	0	0	0
24000	0	0	0	0	0	0	0	1	0	0	0	0
25000	0	0	0	1	3	0	0	0	0	0	0	0
26000	0	0	1	0	0	0	0	0	0	0	0	0
27000	0	0	0	0	0	0	0	0	0	1	0	0
28000	0	99	0	0	0	0	0	0	0	0	0	0
29000	0	0	54	0	0	0	0	0	0	0	0	0
30000	0	1	0	27	0	0	0	1	0	0	0	0
31000	0	0	1	1	18	0	0	1	0	0	0	0

[ ]

	32000	34000	35000	36000	38000	39000	40000
32000	0	0	0	0	0	0	0
34000	0	0	0	0	0	0	0
35000	0	0	3	4	2	0	0
36000	0	0	0	0	0	0	0
38000	1	0	0	0	1	0	0
39000	0	0	0	0	0	0	0
40000	0	1	0	0	0	0	0

	class.error
15000	0.00000000
16000	1.00000000
17000	1.00000000
18000	1.00000000
19000	0.00000000
20000	1.00000000
21000	0.55000000
22000	1.00000000
23000	0.00000000
24000	1.00000000
25000	0.09090909
26000	1.00000000
27000	1.00000000
28000	0.00000000
29000	0.00000000
30000	0.22857143
31000	0.21739130
32000	1.00000000
34000	1.00000000
35000	0.51612903
36000	1.00000000
38000	0.11538462
39000	1.00000000
40000	1.00000000

```
[ ]  p1 <- predict(rf, test)
     confusionMatrix(p1, test$PremiumPrice)
```

```
Confusion Matrix and Statistics

          Reference
Prediction 15000 16000 17000 18000 19000 20000 21000 22000 23000 24000 25000
     15000    50     0     0     0     0     0     0     0     0     0     0
     16000     0     0     0     0     0     0     0     0     0     0     0
     17000     0     0     0     0     0     0     0     0     0     0     0
     18000     0     0     0     0     0     0     0     0     0     0     0
     19000     0     0     0     0     4     0     0     0     0     0     0
     20000     0     0     0     0     0     0     0     0     0     0     0
     21000     0     0     0     0     0     0     4     0     0     0     0
     22000     0     0     0     0     0     0     0     0     0     0     0
     23000     0     0     0     0     0     0     2     0    62     0     3
     24000     0     0     0     0     0     0     0     0     0     0     0
     25000     0     1     0     0     0     0     0     0     0     0    20
     26000     0     0     0     0     0     0     0     0     0     0     0
     27000     0     0     0     0     0     0     0     0     0     0     0
     28000     0     0     0     0     0     0     0     0     0     1     0
     29000     0     0     0     0     0     0     0     0     0     0     0
     30000     0     0     0     0     0     0     0     0     0     0     1
     31000     0     0     0     0     0     0     0     0     0     0     2
     32000     0     0     0     0     0     0     0     0     0     0     0
     34000     0     0     0     0     0     0     0     0     0     0     0
     35000     0     0     0     0     0     0     0     0     0     0     0
     36000     0     0     0     0     0     0     0     0     0     0     0
     38000     0     0     0     0     0     0     0     0     0     0     0
     39000     0     0     0     0     0     0     0     0     0     0     0
     40000     0     0     0     0     0     0     0     0     0     0     0
```

```
          Reference
Prediction 26000 27000 28000 29000 30000 31000 32000 34000 35000 36000 38000
     15000     1     0     0     0     0     0     0     0     0     0     0
     16000     0     0     0     0     0     0     0     0     0     0     0
     17000     0     0     0     0     0     0     0     0     0     0     0
     18000     0     0     0     0     0     0     0     0     0     0     0
     19000     0     0     0     0     0     0     0     0     0     0     1
     20000     0     0     0     0     0     0     0     0     0     0     0
     21000     0     0     0     0     0     0     0     0     0     0     0
     22000     0     0     0     0     0     0     0     0     0     0     0
     23000     1     0     0     0     0     2     1     0     0     0     0
     24000     0     0     0     0     0     0     0     0     0     0     0
     25000     0     0     0     0     1     1     0     0     0     0     0
     26000     0     0     0     0     0     0     0     0     0     0     0
     27000     0     0     0     0     0     0     0     0     0     0     0
     28000     0     0    33     0     0     0     0     0     0     0     0
     29000     0     0     0    18     0     0     0     0     1     0     0
     30000     0     0     0     0     9     0     0     0     0     0     0
     31000     0     0     0     0     0     5     0     0     0     0     0
     32000     0     0     0     0     0     0     0     0     0     0     0
     34000     0     0     0     0     0     0     0     0     0     0     0
     35000     0     0     0     0     2     0     0     0     9     0     0
     36000     0     0     0     0     0     0     0     0     0     0     0
     38000     0     0     0     0     0     0     0     0     0     0     7
     39000     0     0     0     0     0     0     0     0     0     0     0
     40000     0     0     0     0     0     0     0     0     0     0     0
```

```
          Reference
Prediction 39000 40000
     15000     0     0
     16000     0     0
     17000     0     0
     18000     0     0
     19000     0     0
     20000     0     0
     21000     0     0
     22000     0     0
     23000     1     0
     24000     0     0
     25000     0     0
     26000     0     0
     27000     0     0
     28000     0     0
     29000     0     0
     30000     0     0
     31000     0     0
     32000     0     0
     34000     0     0
     35000     0     0
     36000     0     0
     38000     0     0
     39000     0     0
     40000     0     0
```

```
Overall Statistics

               Accuracy : 0.9095
                 95% CI : (0.8661, 0.9424)
    No Information Rate : 0.2551
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.8922

 Mcnemar's Test P-Value : NA

Statistics by Class:

                     Class: 15000 Class: 16000 Class: 17000 Class: 18000
Sensitivity                1.0000     0.000000           NA           NA
Specificity                0.9948     1.000000            1            1
Pos Pred Value             0.9804          NaN           NA           NA
Neg Pred Value             1.0000     0.995885           NA           NA
Prevalence                 0.2058     0.004115            0            0
Detection Rate             0.2058     0.000000            0            0
Detection Prevalence       0.2099     0.000000            0            0
Balanced Accuracy          0.9974     0.500000           NA           NA
                     Class: 19000 Class: 20000 Class: 21000 Class: 22000
Sensitivity               1.00000           NA      0.66667           NA
Specificity               0.99582            1      1.00000            1
Pos Pred Value            0.80000           NA      1.00000           NA
Neg Pred Value            1.00000           NA      0.99163           NA
Prevalence                0.01646            0      0.02469            0
Detection Rate            0.01646            0      0.01646            0
Detection Prevalence      0.02058            0      0.01646            0
Balanced Accuracy         0.99791           NA      0.83333           NA
                     Class: 23000 Class: 24000 Class: 25000 Class: 26000
```

```
Sensitivity                1.0000     0.000000      0.76923      0.00000
Specificity                0.9448     1.000000      0.98618      1.00000
Pos Pred Value             0.8611          NaN      0.86957          NaN
Neg Pred Value             1.0000     0.995885      0.97273      0.99177
Prevalence                 0.2551     0.004115      0.10700      0.00823
Detection Rate             0.2551     0.000000      0.08230      0.00000
Detection Prevalence       0.2963     0.000000      0.09465      0.00000
Balanced Accuracy          0.9724     0.500000      0.87770      0.50000
                     Class: 27000 Class: 28000 Class: 29000 Class: 30000
Sensitivity                    NA       1.0000      1.00000      0.75000
Specificity                     1       0.9952      0.99556      0.99567
Pos Pred Value                 NA       0.9706      0.94737      0.90000
Neg Pred Value                 NA       1.0000      1.00000      0.98712
Prevalence                      0       0.1358      0.07407      0.04938
Detection Rate                  0       0.1358      0.07407      0.03704
Detection Prevalence            0       0.1399      0.07819      0.04115
Balanced Accuracy              NA       0.9976      0.99778      0.87284
                     Class: 31000 Class: 32000 Class: 34000 Class: 35000
Sensitivity               0.62500     0.000000          NA      0.90000
Specificity               0.99149     1.000000           1      0.99142
Pos Pred Value            0.71429          NaN          NA      0.81818
Neg Pred Value            0.98729     0.995885          NA      0.99569
Prevalence                0.03292     0.004115           0      0.04115
Detection Rate            0.02058     0.000000           0      0.03704
Detection Prevalence      0.02881     0.000000           0      0.04527
Balanced Accuracy         0.80824     0.500000          NA      0.94571
                     Class: 36000 Class: 38000 Class: 39000 Class: 40000
Sensitivity                    NA      0.87500     0.000000          NA
Specificity                     1      1.00000     1.000000           1
Pos Pred Value                 NA      1.00000          NaN          NA
Neg Pred Value                 NA      0.99576     0.995885          NA
Prevalence                      0      0.03292     0.004115           0
Detection Rate                  0      0.02881     0.000000           0
Detection Prevalence            0      0.02881     0.000000           0
Balanced Accuracy              NA      0.93750     0.500000          NA
```

```
pred <- predict(rf, newdata = test[-11])
cm <- table(pred,obs = test[,11])
```

```
sum <- 0
for (i in 1:24){
  for(j in 1:24){
    if(i!=j){
      sum <-sum+cm[i,j]
    }
  }
}
sum
print(paste("The Accuracy of Random Forest Model is",(243-sum)/2.43))
```

```
22
[1] "The Accuracy of Random Forest Model is 90.9465020576132"
```

```
[ ] install.packages("caret")
```

```
Installing package into ‘/usr/local/lib/R/site-library’
(as ‘lib’ is unspecified)
```

```
[ ] library(caret)
```

```
[ ] # log10 transform of response variable
    df$logCharges<- log10(df$charges)

    # Split the data into training and test sets
    set.seed(122)                    # Set the seed to make the partition reproducible
    training.samples <- df$logCharges %>% createDataPartition(p = 0.8, list = FALSE)
    train  <- df[training.samples, ]
    test <- df[-training.samples, ]
```

```
# Train the model on the training dataset
formula <- as.formula("logCharges ~ smoker + bmi + age + children + sex + region")

model <- lm(formula, data = train)

summary(model)
```

```
Call:
lm(formula = formula, data = train)

Residuals:
     Min       1Q   Median       3Q      Max
-0.40616 -0.09020 -0.02317  0.03310  0.93634

Coefficients:
                  Estimate Std. Error t value Pr(>|t|)
(Intercept)      3.0309810  0.0342281  88.553  < 2e-16 ***
smokeryes        0.6760177  0.0144519  46.777  < 2e-16 ***
bmi              0.0058027  0.0009931   5.843 6.81e-09 ***
age              0.0153611  0.0004142  37.088  < 2e-16 ***
children1        0.0538514  0.0146930   3.665 0.000260 ***
children2        0.1287088  0.0161331   7.978 3.84e-15 ***
children3        0.1086630  0.0189419   5.737 1.26e-08 ***
children4        0.2109956  0.0411738   5.125 3.54e-07 ***
children5        0.1836043  0.0552913   3.321 0.000929 ***
sexmale         -0.0304862  0.0115908  -2.630 0.008657 **
regionnorthwest -0.0305322  0.0164325  -1.858 0.063441 .
regionsoutheast -0.0598820  0.0168307  -3.558 0.000390 ***
regionsouthwest -0.0562472  0.0165515  -3.398 0.000703 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

```
Residual standard error: 0.1884 on 1059 degrees of freedom
Multiple R-squared:  0.7789,    Adjusted R-squared:  0.7764
F-statistic: 310.9 on 12 and 1059 DF,  p-value: < 2.2e-16
```

A significant regression equation was found (F(12,1057) = 303.9, p < 0.001), with an adjusted R-squared of 0.7764. In other words, the model explains 77.6% of total variance in the sample. Null hypothesis is rejected

```
[ ] # Make predictions on the training dataset
    predictions <- model %>% predict(train)
    # Model performance
    # (a) Calculating the residuals
    residuals <- train$logCharges - predictions
    # (b) Calculating Root Mean Squared Error
    rmse <- sqrt(mean(residuals^2))

    rmse %>%
        round(digits=3)
```

0.187

+ Code + Text

```
[ ] # Make predictions on the testing dataset
    predictions <- model %>% predict(test)
    # Model performance
    # (a) Calculating the residuals
    residuals <- test$logCharges - predictions
    # (b) Calculating Root Mean Squared Error
    rmse <- sqrt(mean(residuals^2))

    rmse %>%
        round(digits=3)
```

0.208

Since the response variable had been transformed, RMSE values have lost their units and are not easily interpretable. To interpret RMSE in a meaningful way, some backtransformations need to be performed.

```
[ ] # Calculating RMSE for training data with backtransformed data

    predictions <- model %>% predict(train)
    # Model performance
    # (a) Calculating the residuals
    residuals <- 10^train$logCharges - 10^predictions # backtransform measured and predicted values
    # (b) Calculating Root Mean Squared Error
    rmse <- sqrt(mean(residuals^2))

    round(rmse)
```

8334

```
[ ] # Calculating RMSE for testing data with backtransformed data

    predictions <- model %>% predict(test)
    # Model performance
    # (a) Calculating the residuals
    residuals <- 10^test$logCharges - 10^predictions # backtransform measured and predicted values
    # (b) Calculating Root Mean Squared Error
    rmse <- sqrt(mean(residuals^2))

    round(rmse)
```

9000

To measure robustness of the model, an absolute measure of fit - RMSE was calculated, RMSE (test set) = 0.208, RMSE (training set) = 0.187. This is an indicator that the model is not overfit. After backtransforming the residuals, the RMSE for the test set was $9000, meaning the model's predictions are usually off by this amount.

Different tests were done on the dataset to understand the impact of each feature on the premium price, specifically the smokers, region and number of children columns. Tests like Wilcoxon rank test and Kruskal-Wallis tests were done.

1. Smokers

```
[ ]  df %>%
       group_by(region) %>%
         summarise(
           count = n(),
           min = min(charges),
           median = median(charges),
           max = max(charges),
           IQR = IQR(charges)
         ) %>%
       arrange(desc(median)) # sort by median in descending order
```

A tibble: 4 × 6

region	count	min	median	max	IQR
<fct>	<int>	<dbl>	<dbl>	<dbl>	<dbl>
northeast	324	1694.80	10057.65	58571.07	11493.04
southeast	364	1121.87	9294.13	63770.43	15085.40
northwest	325	1621.34	8965.80	60021.40	9992.00
southwest	325	1241.57	8798.59	52590.83	8711.45

```
[ ]  wilcox.test(df$charges ~ df$smoker)
```

```
        Wilcoxon rank sum test with continuity correction

data:  df$charges by df$smoker
W = 7403, p-value < 2.2e-16
alternative hypothesis: true location shift is not equal to 0
```

H0: There is no difference in the distribution scores. HA: There is a difference in the distribution scores. The test indicated that there is a significant difference between the groups, W = 7403, p < 0.001. The null hypothesis is rejected.

2. Region

```
[ ]  df %>%
       group_by(region) %>%
         summarise(
           count = n(),
           min = min(charges),
           median = median(charges),
           max = max(charges),
           IQR = IQR(charges)
         ) %>%
       arrange(desc(median)) # sort by median in descending order
```

A tibble: 4 × 6

region	count	min	median	max	IQR
<fct>	<int>	<dbl>	<dbl>	<dbl>	<dbl>
northeast	324	1694.80	10057.65	58571.07	11493.04
southeast	364	1121.87	9294.13	63770.43	15085.40
northwest	325	1621.34	8965.80	60021.40	9992.00
southwest	325	1241.57	8798.59	52590.83	8711.45

```
[ ]  kruskal.test(charges ~ region, data = df)
```

```
        Kruskal-Wallis rank sum test

data:  charges by region
Kruskal-Wallis chi-squared = 4.7342, df = 3, p-value = 0.1923
```

H0: There is no difference between the medians. HA: There is a difference between the medians. The test showed that the difference between the median medical charges in different regions is not significant, H(3) = 4.73, p = 0.19. A significant level of 0.19 indicates a 19% risk of concluding that a difference exists when there is no actual difference. The null hypothesis is accepted.

3. Children

```
[ ]  df %>%
       group_by(children) %>%
         summarise(
           count = n(),
           min = min(charges),
           median = median(charges),
           max = max(charges),
           IQR = IQR(charges)
         ) %>%
       arrange(desc(median)) # sort by median in descending order
```

A tibble: 6 × 6

children	count	min	median	max	IQR
<fct>	<int>	<dbl>	<dbl>	<dbl>	<dbl>
4	25	4504.66	11033.660	40182.25	9616.16
3	157	3443.06	10600.550	60021.40	12547.41
0	574	1121.87	9856.950	63770.43	11705.70
2	240	2304.00	9264.980	49577.66	14094.34
5	18	4687.80	8589.565	19023.26	4144.97
1	324	1711.03	8483.870	58571.07	10840.40

```
[ ]  kruskal.test(charges ~ children, data = df)
```

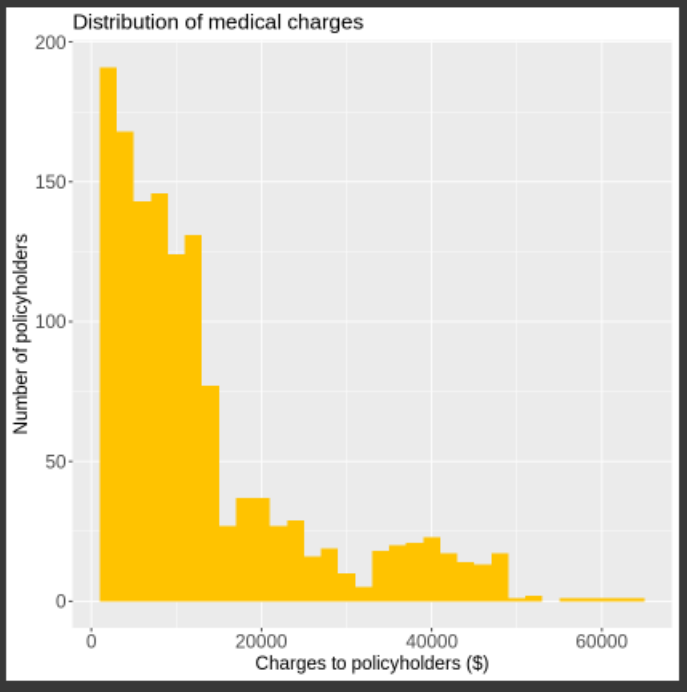
```
        Kruskal-Wallis rank sum test

data:  charges by children
Kruskal-Wallis chi-squared = 29.487, df = 5, p-value = 1.86e-05
```

A Kruskal-Wallis test (assumptions met) also showed that the number of dependents covered by the insurance policy significantly affects medical costs billed on that policy by the insurance company, $H(5) = 29.49$, $p < 0.001$.

The data is transformed into normal distribution by using logarithmic functions.

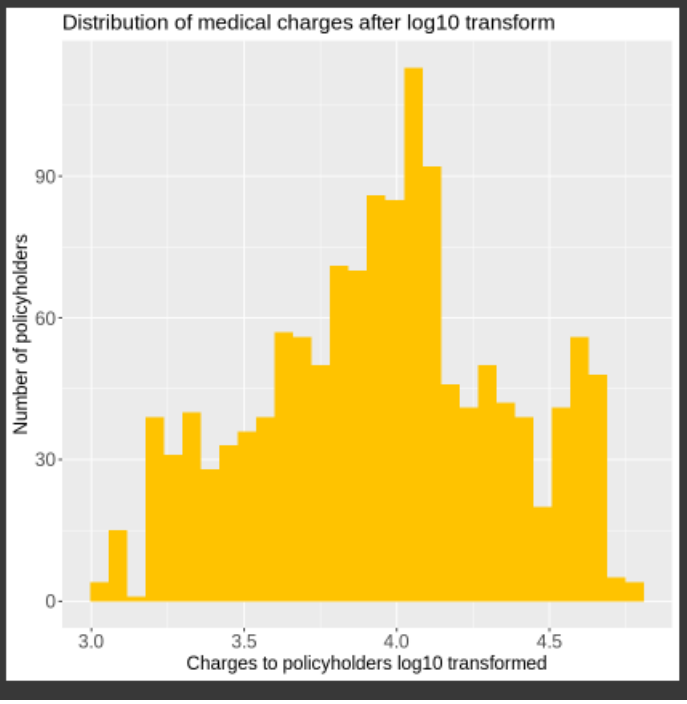
```
charges_hist <- df %>%
    ggplot(
        aes(x=charges)
        ) +
    geom_histogram(
        binwidth = 2000,
        show.legend = FALSE,
        fill = "#FFC300"
        )+
    labs(
        x = "Charges to policyholders ($)",
        y = "Number of policyholders",
        title = "Distribution of medical charges"
        )+
    # resize text
    theme(
        plot.title = element_text(size=16),
        axis.text = element_text(size=14),
        axis.title = element_text(size=14)
    )

charges_hist_log10 <- df %>%
    ggplot(
        aes(x=log10(charges))
        ) +
    geom_histogram(
        show.legend = FALSE,
        fill = "#FFC300"
        )+
    labs(
        x = "Charges to policyholders log10 transformed",
        y = "Number of policyholders",
        title = "Distribution of medical charges after log10 transform"
        )+
    # resize text
    theme(
        plot.title = element_text(size=16),
        axis.text = element_text(size=14),
        axis.title = element_text(size=14)
    )
```

```
[ ]  charges_hist
```




```
charges_hist_log10
```

```
`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



Results

Random forest regressor gave an accuracy of 90%. Smoking having the strongest effect on medical expenses is quite expected. Increases in the BMI score lead to rather small expense increases; however, it is worth pointing out that normal BMI scores are not indicative of ill health. Only people in the underweight (BMI < 18.5), overweight (BMI 25.0 to 29.9), and obese (BMI ≥ 30) ranges would be expected to have poorer health outcomes.

Same should be said of the effect of aging - 22-year-olds would be expected to enjoy the same level of health as 18-year-olds despite being 4 years older. However, middle aged and elderly people will most likely see a rapid decline in health year by year.

Medical expenses increasing with increased number of dependents is to be expected. However, having three dependents covered by insurance seems to be cheaper than having two dependents, and five dependents sees a lesser increase in charges than four. This may be explained by the uneven number of observations in each group. For example, no dependents group has 574 observations when five dependents group only has 18.

It is also interesting to note that even though the median difference of medical charges between men and women is only $43, the relationship between sex and medical charges was significant in the multiple linear regression models.

Lastly, whether the model is robust can only be determined knowing what the acceptable cost of error is. Being able to explain 77.9% of the total variance with an RMSE of $9000 may well be enough if the company can deal with the potential mis-predictions.

Conclusion

Machine learning (ML) is one aspect of computational intelligence that can solve different problems in a wide range of applications and systems when it comes to leveraging historical data. Predicting medical insurance costs is still a problem in the healthcare industry that needs to be investigated and improved. In this paper, by using a set of ML algorithms, a computational intelligence approach is applied to predict healthcare insurance costs. The medical insurance dataset was obtained from the KAGGLE repository and was utilised for training and testing the random forest regression. The regression of this dataset followed the steps of pre-processing, feature engineering, data splitting, regression, and evaluation. The resultant outcome revealed that random forest regressor achieved a high accuracy of 90%.

Future works

In future work, we will use nature-inspired and meta-heuristic algorithms to modify the parameters of machine learning and deep learning approaches on multiple medical health-related datasets. Also, real time data can be obtained via smart watches and other wearable to monitor the personal health of a person and provide discounts or offers to those who maintain good health consistently.

References

https://www.researchgate.net/publication/348559741_Predict_Health_Insurance_Cost_by_using_Machine_Learning_and_DNN_Regression_Models

www.kaggle.com

https://www.hindawi.com/journals/mpe/2021/1162553/

https://www.researchgate.net/publication/293137685_Quality_and_cost_of_healthcare_An_Indian_prespective_an_assessment_of_direct_cost_of_quality_across_hospitals_in_India

https://bmchealthservres.biomedcentral.com/articles/10.1186/1472-6963-12-23