A PROJECT ON

THE PREDICTION OF LIVER DISEASES

Submitted by

SHAIKH NAWAJ YUSUF

Under the Guidance of

Prof. B.T.THORVE

For the partial fulfilment of

M.Sc. Statistics Course ST-46

Academic year 2018-19

Department of Statistics

Ahmednagar Jilha Maratha Vidya Prasarak Samaj’s

New Arts Commerce and Science College,

Ahmednagar

CERTIFICATE

This is to certify that ***SHAIKH NAWAJ YUSUF*** of class M.Sc. has completed assigned project of the ***"Prediction of Liver Disease".***

As laid down by the Savitribai Phule Pune University for the academic year 2018-2019.

Project Guide **Head of Department**

External Examiner

INDEX

ACKNOWLEDGEMENT

I avail this opportunity to express our deep sense of gratitude and whole hearted thanks to our guide Prof. B.K. Thorve for giving this valuable guidance, inspiration and affectionate encouragement to embark to this project.

I thankful to our respected Principal Prof. Dr. B.H. Zaware, Prof. Dr. A.K. Pandharkar, Former Head of Department Prof. J.K. Kshirsagar and Head of Department Prof. M.S. Kasture for their support and co-operation towards successful completion of this project.

I extremely thankful to Dr. A.A. Kulkarni, Prof. C.G. Shelke, Prof. M.D. Rohakale, Prof. R.J. Wagh, Prof. S. Agarkar for their support and guidance of this project.

I also acknowledgement our overwhelming gratitude and immense respect to all the Non-teaching staff of department of statistics.

Last but not the least I would like to thanks all our friends who helped us directly or indirectly in our endeavour and infused their help for the success of our project.

MOTIVATION

I’m studying Statistics from the past five years. When I’m say learning, I mean that I’m discovering new dimensions to each and every part of our day to day life. As statistics have innumerable application in various fields such as Medical field, Agriculture, Actuarial Science, Software Industries, Sports, etc. So there is always an inner desire to use statistical tools practically. In the second year of M.Sc. Statistics, I have been given an opportunity to apply our statistics knowledge in one of these fields to do a project as per our academic requirement as a part of our syllabus course as project ST-46.

Health is one of the important part of human life. Doing project which is related to human life is good and important that’s why I’m doing project on prediction of liver disease which is very beneficial to our community.

As I’m also a part of young generation I always curious to learn new concepts and use my theoretical knowledge for our community and as I know the health variable is so important for humans and also all organisms. So I decided to do the project “prediction of liver disease”. Because this is very relevant topic to our health.

ABSTRACT

We know that liver plays an important role in human body because it is a huge solid organ in the human body. It is also a gland which secrets bile. The liver threader vital role in many physical functions from protein manufacturer and blood clotting to fat, sugar and iron metabolism .

Nowadays liver disease becomes a one of the major disease in India. And we want predict this by using some data mining tools to alert human beings as early as possible using their blood report and past liver disease patients reports. In that I used only data mining techniques Because data mining tools give better accuracy in comparison of other techniques and it is easy to apply and easy to handle.

Aim of this project is to build a best model to predict liver disease and use that model for human being to avoid risk of health, money and to give better health .

Keywords: R-Studio, Data Mining, Support Vector Machine (SVM), K-Nearest neighbour (KNN), Naïve Bayes classifier (NBC), Decision tree, Random forest.

INTRODUCTION

The liver is one of the important and largest organs of the body that is situated in the upper right portion of the stomach and under the diaphragm. The weight of liver is about 1.36 kg and reddish brown in colour. The liver performs more than 500 functions, some well-known functions are the production of bile, production of important proteins for blood clotting, purification of blood, helping in fat digestion, decomposing red blood cells and detoxifying harmful chemicals.

The liver doing mostly work in the function of body from protein production and blood clotting to cholesterol, glucose (sugar), and iron metabolism. When liver is infected with a virus, injured by chemicals, or under attack from own immune system, the basic danger is the same – that liver will become so damaged that it can no longer work to keep a person alive. Liver disease caused by hepatotropic viruses imposes a substantial burden on health care resources. Persistent infections from hepatitis B virus (HBV), hepatitis C virus, and hepatitis delta virus result in chronic liver disease.

The accurate diagnosis of patients and providing proper treatment is very important in medical science. Wrong medication may lead to wastage of money and time for the patients, sometimes this may lead to the irreparable loss (death). One of the fatal diseases that have affected one in five persons of India is liver disease. It is expected that India may become the “world capital” for liver disease by 2025. Medical practitioners often fail to detect the liver disease at the earlier stage because the symptoms of the disease are vague at the initial stage.

INTRODUCTION OF DATA

I took a data of liver disease from ‘Kaggle’. The liver disease data is secondary data. The data contains 583 observations collected from north east of Andhra Pradesh, India. In that there are 416 individuals are liver patients and remaining 167 individuals are not liver patients. This data contains 441 male patient/not patient records and 142 female patient/not patient records. Any observation(individual) whose age exceeded 89 is listed as being of age “90”.

In this data there are 11 variables

1) age (Age of an individual)

2) gender (Gender of an individual)

3) tot_bilirubin (Total Bilirubin)

4) direct_bilirubin (Direct Bilirubin)

5) total_proteins (Total proteins)

6) albumin (Albumin)

7) ag_ratio (Albumin and Globulin ratio)

8) sgpt (Alanine Aminotransferase)

9) sgot (Aspartate Aminotransferase)

10) alkphos (Alkaline Phosphatase)

11) Is_patient (It is a binary variable, 1 = Individual is suffering from liver disease Or 0 = individual is not suffering from liver disease)

Link of the dataset :

https://www.kaggle.com/jeevannagaraj/indian-liver-patient-dataset.

EXPLANATION OF VARIABLES :

1) Bilirubin is yellowish fluid formed in the liver by breakdown of haemoglobin and excreted in bile.

The NORMAL range of **total bilirubin** (direct and indirect) is between **01.to 1.2 mg/dl.**(some lab use the high range as up to 1.9 mg/dl values).

The NORMAL range of **direct bilirubin** is between **0 to 0.4 mg/dl.**

(milligrams/decilitre).

2) Proteins are nothing but amino acids that are essential for our body to function properly.

The NORMAL range of **total proteins** is between **150 to 600 mg/l** (15 to 60 mg/dl).

3) Albumin is a protein made by our liver and it is soluble in water, moderately soluble in concentrated salt solutions.

The NORMAL range of **albumin** is between **35 to 55 gm/l.**

4) Albumin and Globulin ratio (ag_ratio) is the amount of albumin divided by amount of globulin. It use to checks whether the individual has liver or kidney disease.

The NORMAL range of **albumin** is between **35 to 55 gm/l.**

The NORMAL range of **globulin** is between **23 to 35 gm/l.**

So, the NORMAL range of ag ratio is

Where, **Globulins** is also a protein that have higher molecular weights than albumins and it is not solvable in pure water.

5) ALANINE AMINOTRANSFERASE (sgpt) is a transaminase enzyme. It is released into blood when the liver damaged.

The NORMAL range of **sgpt** is between **7 to 56 units/l.**

6) ASPARTATE AMINOTRANSFERASE (sgot) is a pyridoxal phosphate-dependent transaminase enzyme.

The NORMAL range of **sgot** is **5 to 40 units/l.**

7) ALKALINE PHOSPHATASE (alkphos) is a homodimeric protein enzyme.

The NORMAL range of **alkphos** is **0.73 to 2.45 mukat/l** (microkatal per liter).

Note : - There may have small variations in those ranges according to age.

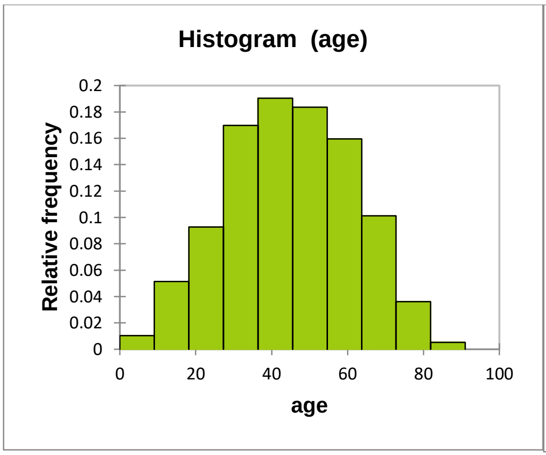
SUMMARY STATISTICS

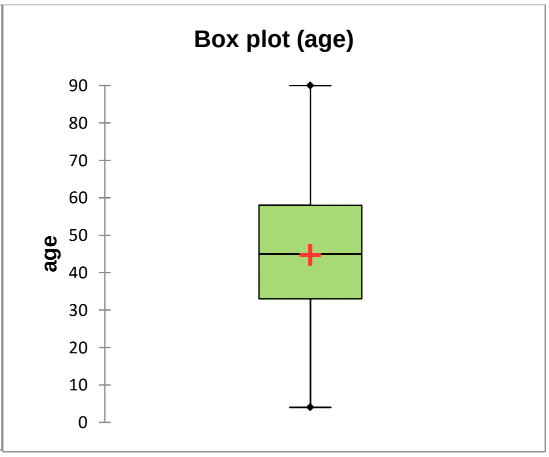
Variable	Observations	Obs. With missing data	Obs. Without missing data	Minimum	Maximum	Mean	Std. deviation
Age	583	0	583	4.000	90.000	44.746	16.190
tot_bilirubin	583	0	583	0.400	75.000	3.299	6.210
direct_bilirubin	583	0	583	0.100	19.700	1.486	2.808
tot_proteins	583	0	583	63.000	2110.000	290.576	242.938
Albumin	583	0	583	10.000	2000.000	80.714	182.620
ag_ratio	583	0	583	10.000	4929.000	109.911	288.919
Sgpt	583	0	583	2.700	9.600	6.483	1.085
Sgot	583	0	583	0.900	5.500	3.142	0.796
Alkphos	583	0	583	0.300	2.800	0.947	0.319

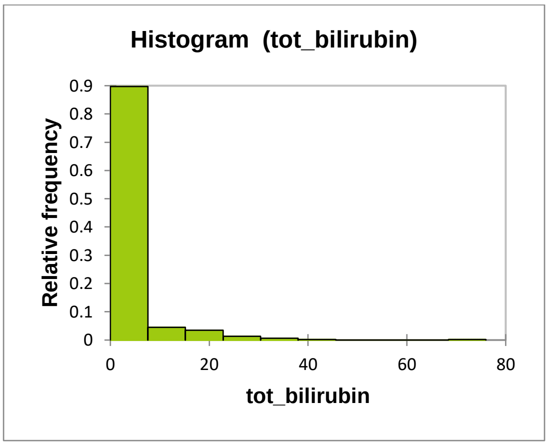
Conclusion:-

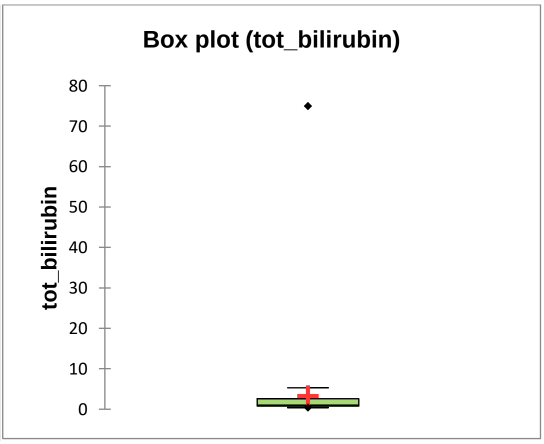
From the above table, we see that in the liver dataset minimum patient’s age is 4 and maximum patients age is 90 means almost all ages of patient’s are included in this dataset. There is high variation in the Ranges of all other variables.

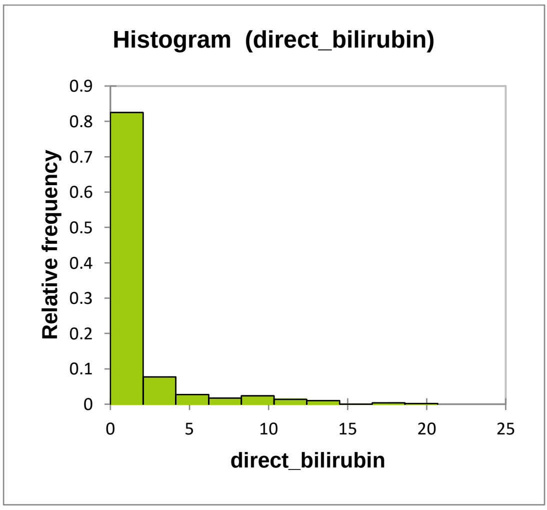
GRAPHICAL REPRESENTATION

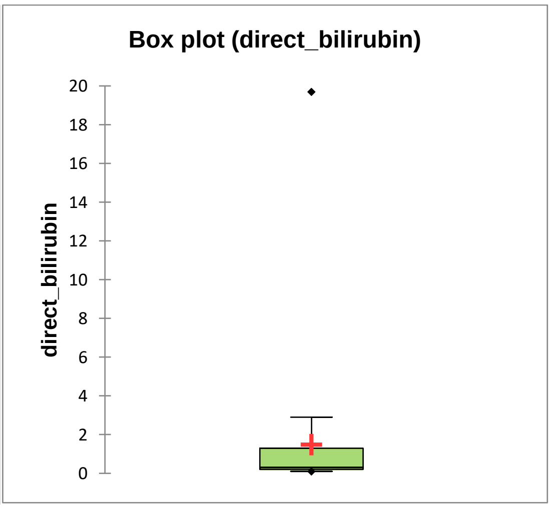


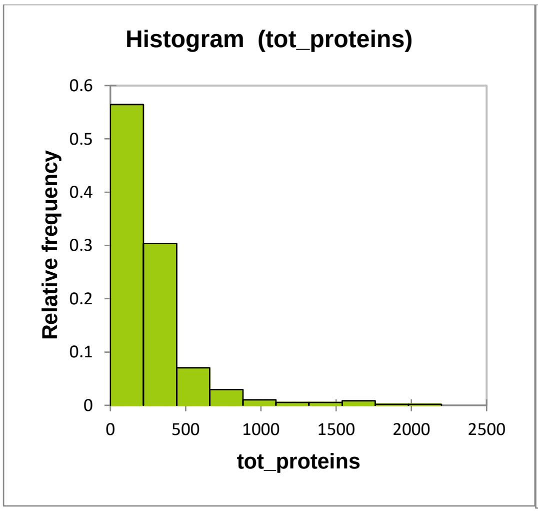


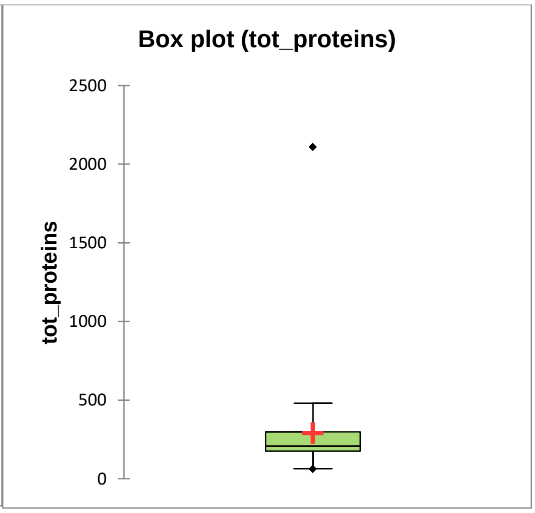


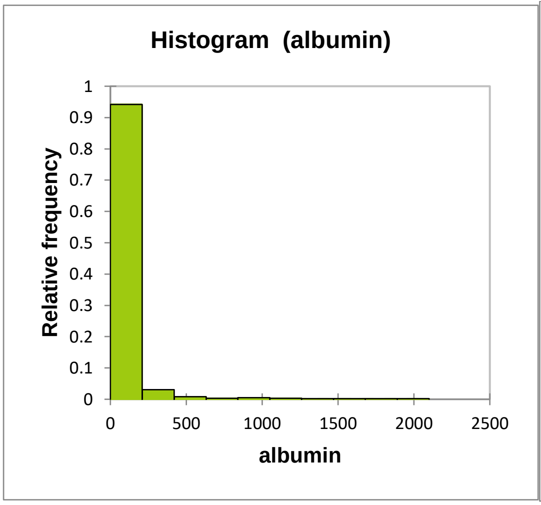


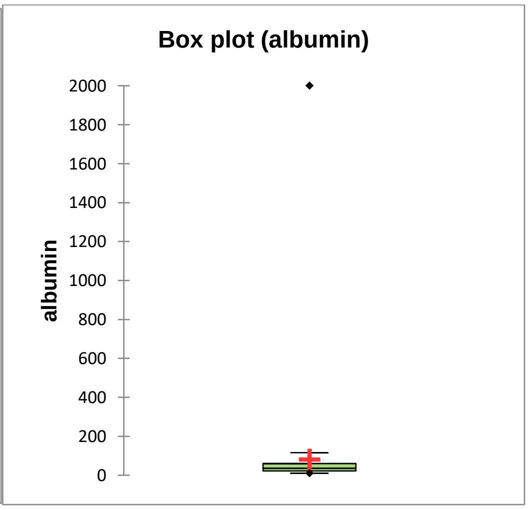


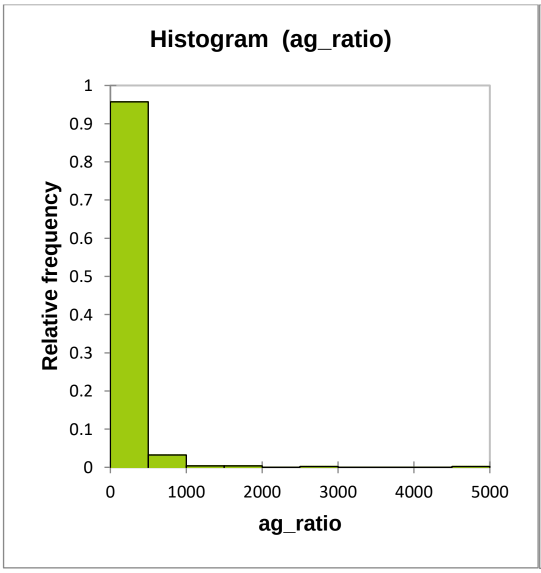


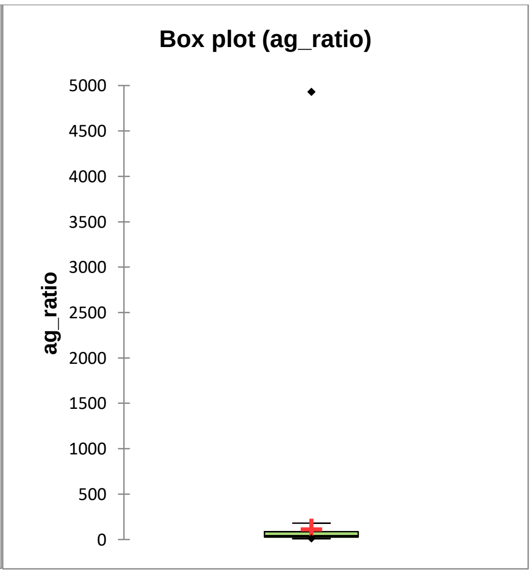

Histogram (tot_proteins)
Relative frequency
0.6
0.5
0.4
0.3
0.2
0.1
0
0
500
1000
1500
2000
2500
tot_proteins

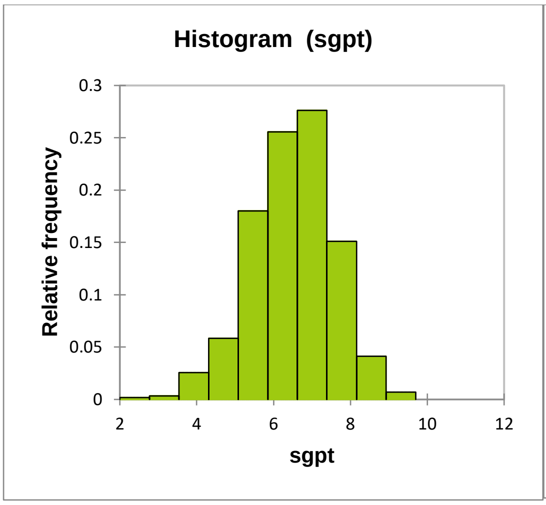

Box plot (tot_proteins)
tot_proteins
2500
2000
1500
1000
500
0

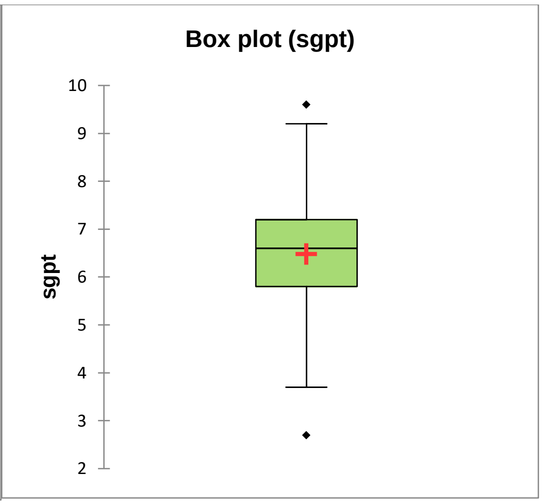

Histogram (albumin)
Relative frequency
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0
500
1000
1500
2000
2500
albumin

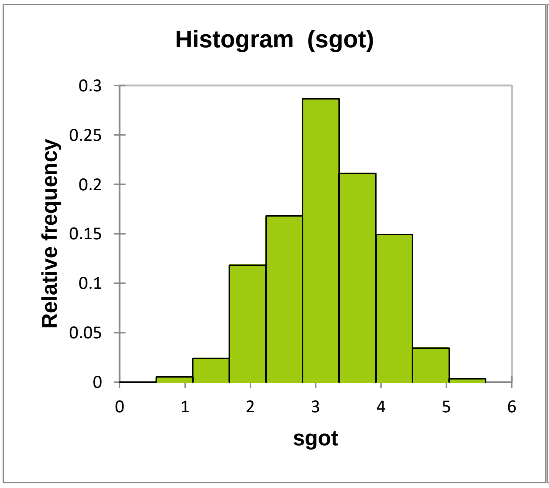

Box plot (albumin)
albumin
2000
1800
1600
1400
1200
1000
800
600
400
200
0

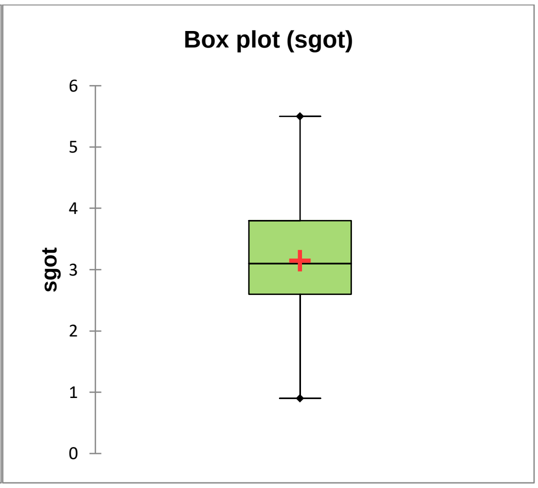

Histogram (ag_ratio)
Relative frequency
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0
1000
2000
3000
4000
5000
ag_ratio

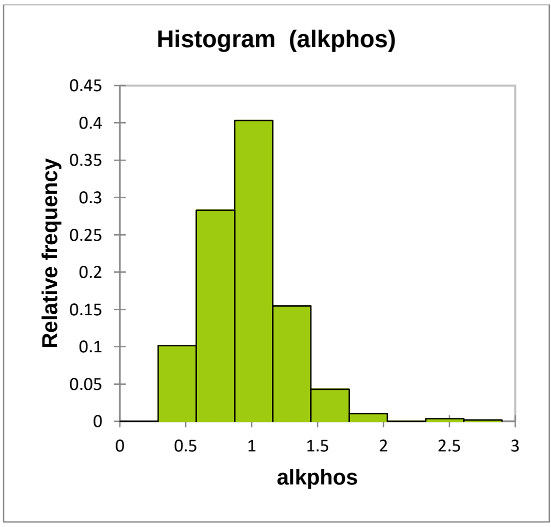

Box plot (ag_ratio)
ag_ratio
5000
4500
4000
3500
3000
2500
2000
1500
1000
500
0

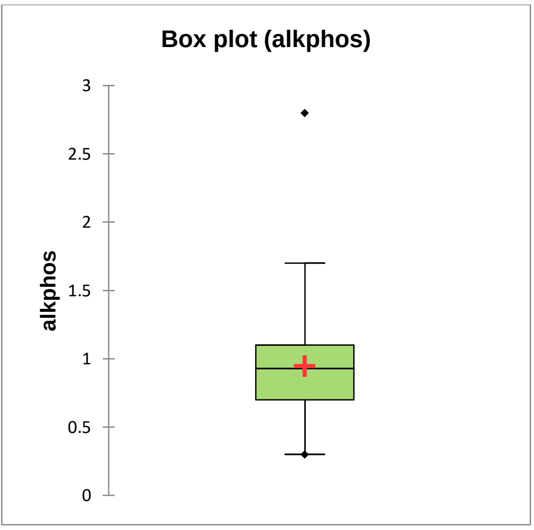










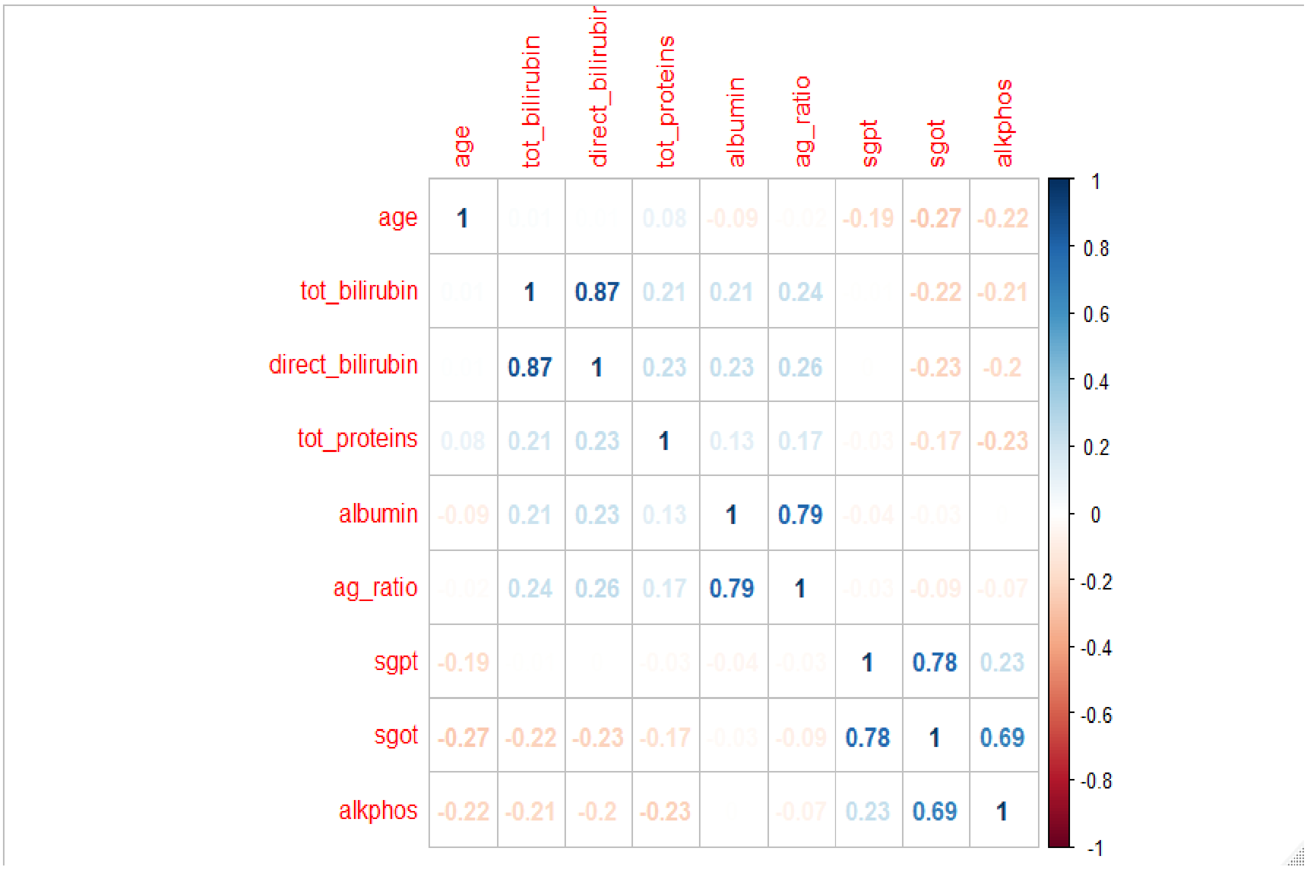


Conclusion : -

In the above histogram of age, sgpt, sgot, alkaline variables are symmetric and histogram of others are positively skewed and there are outliers present in the box plots except the box plot of age and sgot.

CORRELATION MATRIX

```
> corrplot(cor(liver_1.df[, c(1,3:10)]), method = “number”)
```



	age	tot_bilirubin	direct_bilirubir	tot_proteins	albumin	ag_ratio	sgpt	sgot	alkphos
age	1	0.01	0.01	0.08	-0.09	-0.02	-0.19	-0.27	-0.22
tot_bilirubin	0.01	1	0.87	0.21	0.21	0.24	-0.01	-0.22	-0.21
direct_bilirubin	0.01	0.87	1	0.23	0.23	0.26	0	-0.23	-0.2
tot_proteins	0.08	0.21	0.23	1	0.13	0.17	-0.03	-0.17	-0.23
albumin	-0.09	0.21	0.23	0.13	1	0.79	-0.04	-0.03	0
ag_ratio	-0.02	0.24	0.26	0.17	0.79	1	-0.03	-0.09	-0.07
sgpt	-0.19	-0.01	0	-0.03	-0.04	-0.03	1	0.78	0.23
sgot	-0.27	-0.22	-0.23	-0.17	-0.03	-0.09	0.78	1	0.69
alkphos	-0.22	-0.21	-0.2	-0.23	0	-0.07	0.23	0.69	1

Conclusion : -

all variables are dependent to each others some are positively correlated and some are negatively correlated.

METHODOLOGY

Some Important Definitions :

AIC (Akaike's information criteria) : it is used to the choose the best predicted model out of all predicted models. Means the AIC of which model is small that model considered to be best in all other models.

Kappa : it is a statistic which measures inter-rater agreement for qualitative data.

Kappa = (Accuracy – Random Accuracy)/(1 – Random Accuracy)

Where, Random Accuracy = {(TN+FP)*(TN+FN)+(FN+TP)*(FP+TP)}/(TP+FP+FN+TN)

Confusion Matrix

Data class	Reference	
Prediction	True	False
True	True positive (Correct)	False positive (Incorrect)
False	False Negative (Incorrect)	True negative (Correct)

1) **Accuracy** : It is the probability of correctly classified observations. or The accuracy of a classifier is the number of correct predictions from all predictions made.
Accuracy = (TP+TN)/(TP+FP+FN+TN)
2) **Sensitivity or Recall** : Accuracy of the data that classified in Positive class.
Sensitivity = TP/(TP+FN)
3) **Specificity** : The accuracy of the data that classified in negative class.
Specificity = TN/(FP+TN)
4) **Precision** : It is the number of positive predictions divided by the total number of positive class values.
Precision = TP/(TP+FP)
5) **False positive rate** : Percentage of miss classified (Error) in Negative Class.
False positive rate = FP/(FP+TN)
6) **True negative rate** : Percentage of miss classified (Error) in positive class.
True negative rate = FN/(FN+TP)

Regression V/s Classification Problem

Variables can be characterized as either *quantitative* or *qualitative* (also known as *categorical*). Quantitative variables take on numerical values.
Examples include age, height and temperature. On the other hand qualitative variables take on values in one of *K* different *classes*, or categories. Examples of qualitative variables include gender (male or female), the brand of product purchased (brand A, B, or C) and a patient is suffering from a liver disease (Yes or No). when response is quantitative then is a *regression* problems, while those involving a qualitative response are often referred to as *classification* problems.

So in this project I’m focusing on classification methods. Here are some classification techniques as follows.

Libraries and Pre-processing of data

```
> library(ROCR)
> library(lattice)
> library(ggplot2)
> library(caret)
> library(caTools)
> library(e1071)
> library(gplots)
> library(corrplot)
> library(readr)
> library(pROC)
```

I insert the data in R-studio.

```
> liver.df <- read.csv(“C:/Users/Shaikh Nawaj/Desktop/l.csv”)
```

Replacing 2’s with 0’s.

```
> liver.df$is_patient <- ifelse(liver.df$is_patient == 2,0,1)
```

There are four missing values in the alkphos, therefore I’m putting there median values of alkphos variable.

```
> alkphos_median <- median(liver.df$alkphos,na.rm = T)
> liver.df$alkphos[is.na(liver.df$alkphos)] <- alkphos_median
> sum(is.na(liver.df))
```

`[1] 0`   # this 0 value represents there is no missing values in the dataset.

Here I’m creating factor levels from is_patient variable.

```
> liver.df$is_patient <- factor(liver.df$is_patient , levels =  c(0,
+ 1))
> liver_1.df<- liver.df[sample(nrow(liver.df)),]
```

Here I’m spleeting the data into train and test data.

```
> train_1.df <- liver_1.df[1:as.integer(0.70*nrow(liver.df)),]
> test_1.df <- liver_1.df[-c(1:as.integer(0.70*nrow(liver.df))),]
```

Here I’m equalizing number of is_patient in the training data.

```
> train_1.df$is_patient <- factor(train_1.df$is_patient)
> train_1.df <- upSample(x = train_1.df, train_1.df$is_patient)
> prop.table(table(train_1.df$is_patient))
```

0	1
0.5	0.5

Here I’m Creating a dummy variable for Gender

```
> train_1.df$isFemale <- ifelse(train_1.df$gender == ‘Female’,1,0)
> train_1.df$isFemale <- factor(train_1.df$isFemale)
> test_1.df$isFemale <- ifelse(test_1.df$gender == ‘Female’,1,0)
> test_1.df$isFemale <- factor(test_1.df$isFemale)
```

1) BINARY LOGISTIC REGRESSION

When in the data there are quantitative regressors and qualitative response in that case we are using the logistic regression to build a model of prediction . In the liver dataset the response variable is qualitative specially binary(yes or no) so that’s why here I used binary logistic regression method for building a predictive model. Here I used two models, all variables to be considered as in a model 1 and model to consist only significant variables (from analysis of model 1).

Formula for model 1 : -

```
> mod_f1 <- is_patient ~ age + tot_bilirubin + direct_bilirubin + to
+ t_proteins + albumin + ag_ratio+ sgpt + sgot + alkphos + isFemale
```

Here I’m finding the model of binary logistic regression.

```
> model1 <- glm(mod_f1 , data = train_1.df , family = binomial(link
+ = “logit”))
> model1
```

Coefficients:

(Intercept)	-4.236
Age	0.013008
tot_bilirubin	-0.45327
direct_bilirubin	1.404182
tot_proteins	0.002625
Albumin	0.012213
ag_ratio	0.00215
Sgpt	0.819724
Sgot	-1.60498
Alkphos	2.0758
isFemale1	-0.04441

Degrees of Freedom: 563 Total (i.e. Null); 553 Residual
Null Deviance: 781.9
Residual Deviance: 615.3 AIC: 637.3

Predictive model 1 :

is_patient = - 4.236 + 0.013008*age - 0.45327*tot_bilirubin + 1.404182*direct_bilirubin + 0.002625*tot_proteins + 0.012213*albumin + 0.00215*ag_ratio + 0.819724*sgpt - 1.60498*sgot + 2.0758*alkphos - 0.04441*isFemale.

```
> pred_1 <- predict(model1, test_1.df, type = “response”)
> pred_logreg_1 <- ifelse(pred_1 >= 0.5,1,0)
> pred_logreg_1 <- factor(pred_logreg_1)
```

Here I’m evaluating the accuracy of Logistic regression model

```
> confusionMatrix(pred_logreg_1, test_1.df$is_patient)
```

Confusion Matrix and Statistics

	Reference	
Prediction	0	1
0	35	59
1	6	75

Conclusion : -

In the above confusion matrix, out of 175 observations there 35 out of 41 observations and 75 out of 134 observations are correctly classified.

Accuracy	0.6286
95% CI	(0.5524, 0.7003)
No Information Rate	0.7657
P-Value [Acc > NIR]	1
Kappa	0.2854
Mcnemar’s Test P-Value	1.12E-10
Sensitivity	0.8537
Specificity	0.5597
Pos Pred Value	0.3723
Neg Pred Value	0.9259
Prevalence	0.2343
Detection Rate	0.2
Detection Prevalence	0.5371
Balanced Accuracy	0.7067
‘Positive’ Class	0

Conclusion : -

From the above table,

i) We see that the accuracy of this model is 62% and it may varies between (55.24%, 70.03%).

ii) Kappa value = 0.2854 means this model fair(means we may use this model for future prediction).

iii) Here Sensitivity is 0.8537 that indicates proportion of non liver patients that were correctly classified to the total no. of non liver patients.

iv) Here specificity 0.5597 that indicates proportion of liver patients that were correctly classified to the total no. of liver patients.

```
> summary(model1)
```

Deviance Residuals:

Min	1Q	Median	3Q	Max
-3.142	-0.9143	-0.2847	1.0793	1.9557

Coefficients:

	Estimate	Std. Error	z value	Pr(>|z|)
(Intercept)	-4.236	1.257843	-3.368	0.000758
Age	0.013008	0.006072	2.142	0.032179
tot_bilirubin	-0.45327	0.397606	-1.14	0.254291
direct_bilirubin	1.404182	0.768055	1.828	0.067515
tot_proteins	0.002625	0.000949	2.766	0.00568
Albumin	0.012213	0.004495	2.717	0.00658
ag_ratio	0.002154	0.002759	0.781	0.435058
Sgpt	0.819724	0.347792	2.357	0.018426
Sgot	-1.60498	0.677498	-2.369	0.017837
Alkphos	2.0758	1.037039	2.002	0.055321
isFemale1	-0.04441	0.220304	-0.202	0.840238

Predictive model 1 :

is_patient = - 4.236 + 0.013008*age - 0.45327*tot_bilirubin + 1.404182*direct_bilirubin + 0.002625*tot_proteins + 0.012213*albumin +

0.00215*ag_ratio + 0.819724*sgpt - 1.60498*sgot + 2.0758*alkphos - 0.04441*isFemale.

Null deviance: 781.87 on 563 degrees of freedom
Residual deviance: 615.34 on 553 degrees of freedom
AIC: 637.34
Number of Fisher Scoring iterations: 7

Conclusion : -
We know that p-value of those variables < 0.05 are significant. Therefore age, tot_proteins, albumin, sgpt, sgot are significant.

Formula for model 2 :-
(It is a model made by only significant variables)

```
> mod_f2 <- is_patient ~ age + tot_proteins + albumin + sgpt + sgot
> model2 <- glm(mod_f2 , data = train_1.df , family = binomial(link
= “log+ it”))
> model2
```

Coefficients:

(Intercept)	Age	tot_proteins	Albumin	Sgpt	Sgot
-2.2767	0.014412	0.002984	0.018029	0.324158	-0.61547

Degrees of Freedom: 563 Total (i.e. Null); 558 Residual
Null Deviance: 781.9
Residual Deviance: 655.8 AIC: 667.8

Predictive Model 2 :
is_patient = - 2.2767 + 0.014412*age + 0.002984*tot_proteins + 0.018029*albumin + 0.324158*sgpt - 0.61547*sgot.

```
> pred_2 <- predict(model2, test_1.df, type = “response”)
> pred_logreg_2 <- ifelse(pred_2 >= 0.5,1,0)
> pred_logreg_2 <- factor(pred_logreg_2)
> confusionMatrix(pred_logreg_2, test_1.df$is_patient)
```

Confusion Matrix and Statistics

	Reference	
Prediction	0	1
0	33	48
1	8	86

Conclusion : -
In the above confusion matrix, out of 175 observations there 33 out of 41 observations and 86 out of 134 observations are correctly classified.

Accuracy	0.68
95% CI	(0.6054, 0.7484)
No Information Rate	0.7657
P-Value [Acc > NIR]	0.9963
Kappa	0.3337
Mcnemar’s Test P-Value	1.87E-07
Sensitivity	0.8049
Specificity	0.6418
Pos Pred Value	0.4074
Neg Pred Value	0.9149
Prevalence	0.2343
Detection Rate	0.1886
Detection Prevalence	0.4629
Balanced Accuracy	0.7233
‘Positive’ Class	0

Conclusion : -

From the above table,

i) We see that the accuracy of this model is 68% and it may varies between (0.6054, 0.7484).

ii) Kappa value = 0.3337 means this model fair(means we may use this model f or future prediction).

iii) Here Sensitivity is 0.8049 that indicates proportion of non liver patients that were correctly classified to the total no. of non liver patients.

iv) Here specificity 0.6418 that indicates proportion of liver patients that were c orrectly classified to the total no. of liver patients.

```
> summary(model2)
```

Deviance Residuals:

Min	1Q	Median	3Q	Max
-2.8733	-0.9808	-0.3201	1.0673	1.7915

Coefficients:

	Estimate	Std. Error	z value	Pr(>|z|)
(Intercept)	-2.27667	0.740268	-3.075	0.002102
Age	0.014412	0.005887	2.448	0.014352
tot_proteins	0.002984	0.000901	3.31	0.000933
albumin	0.018029	0.003631	4.966	6.84E-07
Sgpt	0.324158	0.157143	2.063	0.03913
Sgot	-0.61547	0.224271	-2.744	0.006064

Predictive Model 2 :

is_patient = - 2.2767 + 0.014412*age + 0.002984*tot_proteins + 0.018029*albumin + 0.324158*sgpt - 0.61547*sgot.

```
    Null deviance: 781.87  on 563  degrees of freedom
Residual deviance: 655.84  on 558  degrees of freedom
AIC: 667.84
Number of Fisher Scoring iterations: 7
```

2) Naïve Bayes classifier

Naïve Bayes classifier is based on Bayes theorem in the probability. In this method we used prior and posterior probability of regressors given each response (yes or no) to analyse the data. Here also I used two models. First one consist of all variables and second one consist only significant variables.

Formula for model 1:

```
> NB1 <- naiveBayes(is_patient ~ ., data = train_1.df)
> pred_nb1 <- predict(NB1, test_1.df,type = “raw”)
> pred_nb1.df <- data.frame(pred_nb1)
> pred_class <- ifelse(pred_nb1.df$X0 > pred_nb1.df$X1 ,0,1)
> test_1.df <- cbind(test_1.df, pred_nb1 = pred_class)
> pred_class <- factor(pred_class)
> confusionMatrix(pred_class, test_1.df$is_patient)
```

Confusion Matrix and Statistics

	Reference	
Prediction	0	1
0	40	80
1	1	54

Conclusion : -

In the above confusion matrix, out of 175 observations there 40 out of 41 observations and 54 out of 134 observations are correctly classified.

Accuracy	0.5371
95% CI	(0.4603, 0.6127)
No Information Rate	0.7657
P-Value [Acc > NIR]	1
Kappa	0.2269
Mcnemar’s Test P-Value	<2e-16
Sensitivity	0.9756
Specificity	0.403
Pos Pred Value	0.3333
Neg Pred Value	0.9818
Prevalence	0.2343
Detection Rate	0.2286
Detection Prevalence	0.6857
Balanced Accuracy	0.6893
‘Positive’ Class	0

Conclusion : -

From the above table,

i) We see that the accuracy of this model is 53.71% and it may varies between (0.4603, 0.6127)

ii) Kappa value = 0.2269 means this model fair(means we may use this model for future prediction).
iii) Here Sensitivity is 0.9756 that indicates proportion of non liver patients that were correctly classified to the total no. of non liver patients.
iv) Here specificity 0.403 that indicates proportion of liver patients that were correctly classified to the total no. of liver patients.

Formula for model 2:

```
> NB2 <- naiveBayes(is_patient ~ age + isFemale + tot_bilirubin + to
+ t_proteins + albumin + ag_ratio + sgot,data = train_1.df)
> pred_nb2 <- predict(NB2, test_1.df,type = “raw”)
> pred_nb2.df <- data.frame(pred_nb2)
> pred_class <- ifelse(pred_nb2.df$X0 >= pred_nb2.df$X1 , 0,1)
> test_1.df <- cbind(test_1.df, pred_nb2 = pred_class)
> pred_class <- factor(pred_class)
> confusionMatrix(pred_class, test_1.df$is_patient)
```

Confusion Matrix and Statistics

	Reference	
Prediction	0	1
0	40	78
1	1	56

Conclusion : -

In the above confusion matrix, out of 175 observations there 40 out of 41 observations and 56 out of 134 observations are correctly classified.

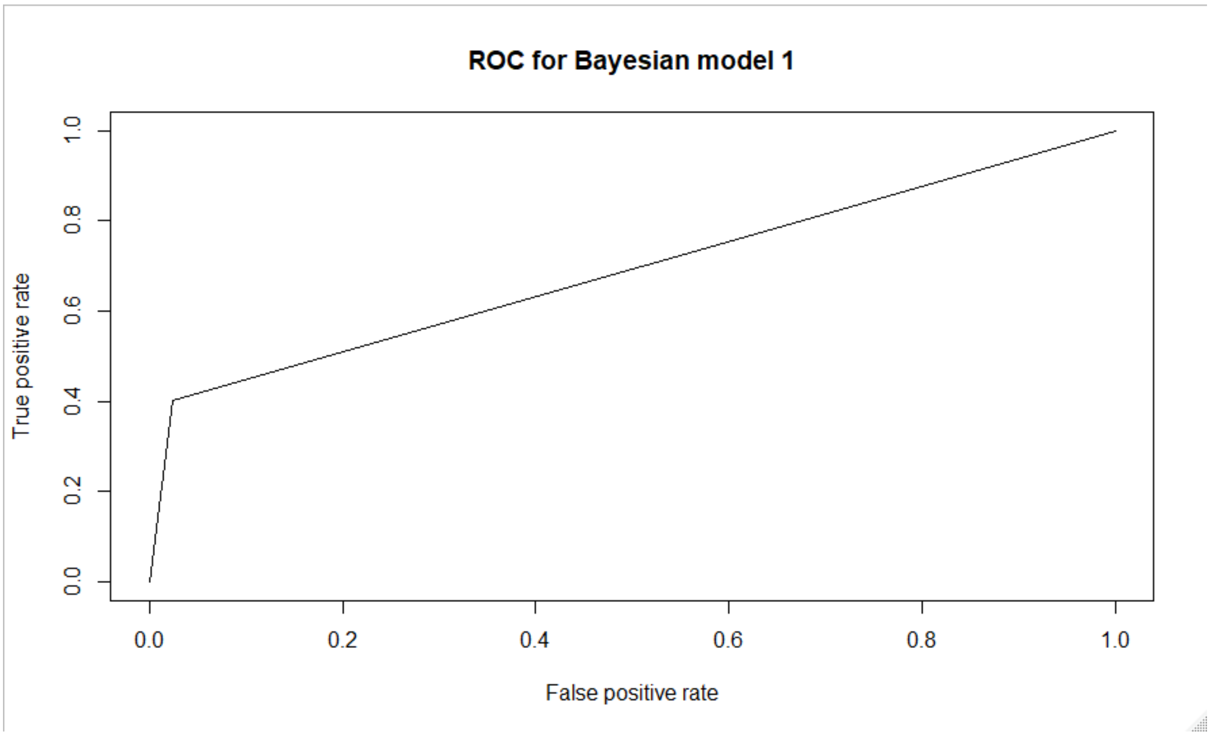
Accuracy	0.5486
95% CI	(0.4712, 0.6240)
No Information Rate	0.7657
P-Value [Acc > NIR]	1
Kappa	0.2284
Mcnemar’s Test P-Value	<2e-16
Sensitivity	0.9756
Specificity	0.4179
Pos Pred Value	0.3333
Neg Pred Value	0.9820
Prevalence	0.2343
Detection Rate	0.2246
Detection Prevalence	0.6867
Balanced Accuracy	0.6934
‘Positive’ Class	0

Conclusion : -

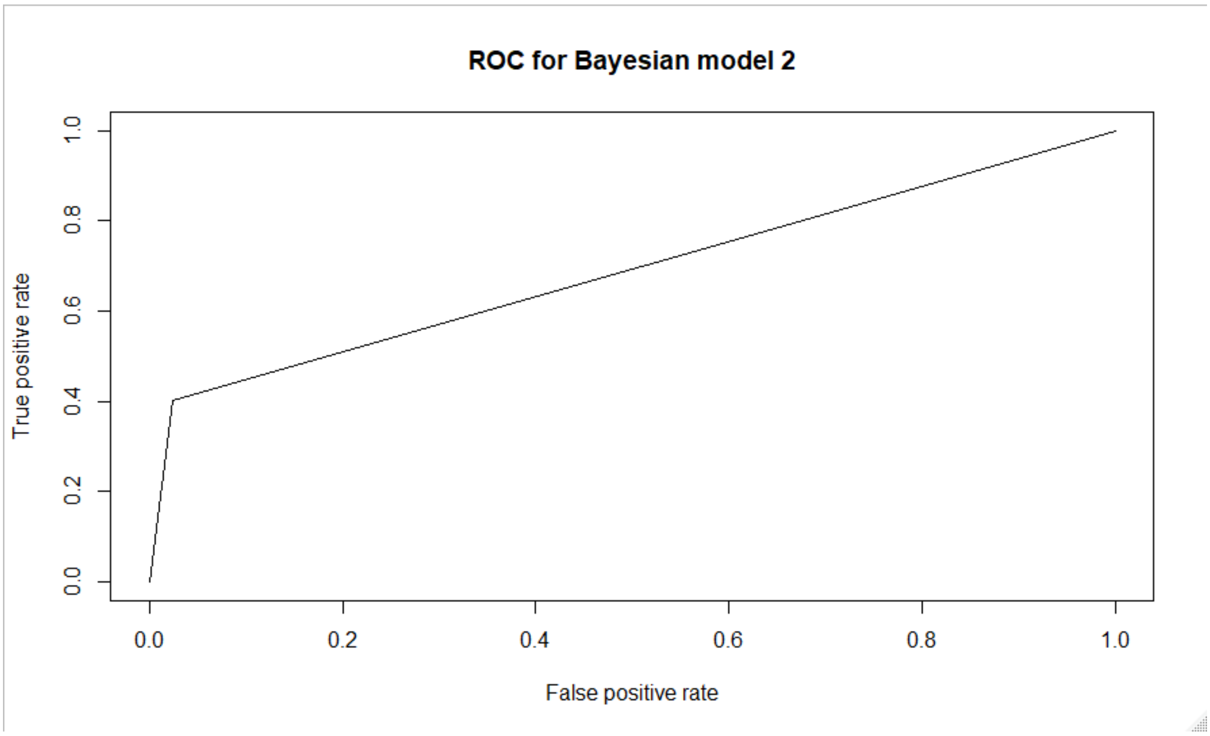
From the above table,
i) We see that the accuracy of this model is 54.86% and it may varies between

(0.4712, 0.6240).
ii) Kappa value = 0.2284 means this model fair(means we may use this model for future prediction).
iii) Here Sensitivity is 0.9756 that indicates proportion of non liver patients that were correctly classified to the total no. of non liver patients.
iv) Here specificity 0.4179 that indicates proportion of liver patients that were correctly classified to the total no. of liver patients.

```
> par(mfrow = c(1,1))
> ROC_pred1 <- prediction(test_1.df$pred_nb1, test_1.df$is_patient)
> ROC_pred1 <- performance(ROC_pred1, ‘tpr’,’fpr’)
> plot(ROC_pred1, colorize = F, text.adj = c(-0.2,1.7), main = “ROC
+ for Bayesian model 1”)
```



```
> ROC_pred2 <- prediction(test_1.df$pred_nb2, test_1.df$is_patient)
> ROC_pred2 <- performance(ROC_pred2, ‘tpr’,’fpr’)
> plot(ROC_pred2, colorize = F, text.adj = c(-0.2,1.7), main = “ROC
+ for Bayesian model 2”)
```



```
> print(paste("AUC ROC of Bayesian model 1  = ",auc(test_1.df$is_pat
+ ient, test_1.df$pred_nb1)))
[1] "AUC of model 1  =  0.689297415362213"
> print(paste("AUC ROC of Bayesian model 2  = ",auc(test_1.df$is_pat
ient, test_1.df$pred_nb2)))
[1] "AUC of model 2  =  0.689345641642546"
```

Conclusion : -

From the above plot, we see that curve of the graph slightly straight to the top left corner indicating that true positive rate is not too high and false positive rate is not too low. And AUC of both curve are nearly same.

3) Random Forest

Random forests are use for classification, regression and other works that operate by building a multitude of decision trees at training period and outputting the class. Random forests average multiple deep decision trees, which are trained on various parts of the same training set, with the aim of minimizing the variance.

```
> model_rf = train(is_patient~., data = training_data,method =’rf’,
+ trControl=trctrl, preProcess=c(“center”,”s+ cale”))
> rf_predictions = predict(model_rf, newdata = testing_data)
> rf_result = confusionMatrix(rf_predictions, testing_data[,11])
> accuracy[1:11, count] = as.data.frame(rf_result$byClass)
> accuracy[12, count] = as.data.frame(rf_result$overall[‘Accuracy’])
> names(accuracy)[count] = “Random Forest”
> count = count + 1
> print(accuracy)
```

Sensitivity	0.897436
Specificity	0.31579
Pos Pred Value	0.729167
Neg Pred Value	0.6
Precision	0.729167
Recall	0.897436
F1	0.804598
Prevalence	0.672414
Detection Rate	0.603448
Detection Prevalence	0.827586
Balanced Accuracy	0.606613
Accuracy	0.706897

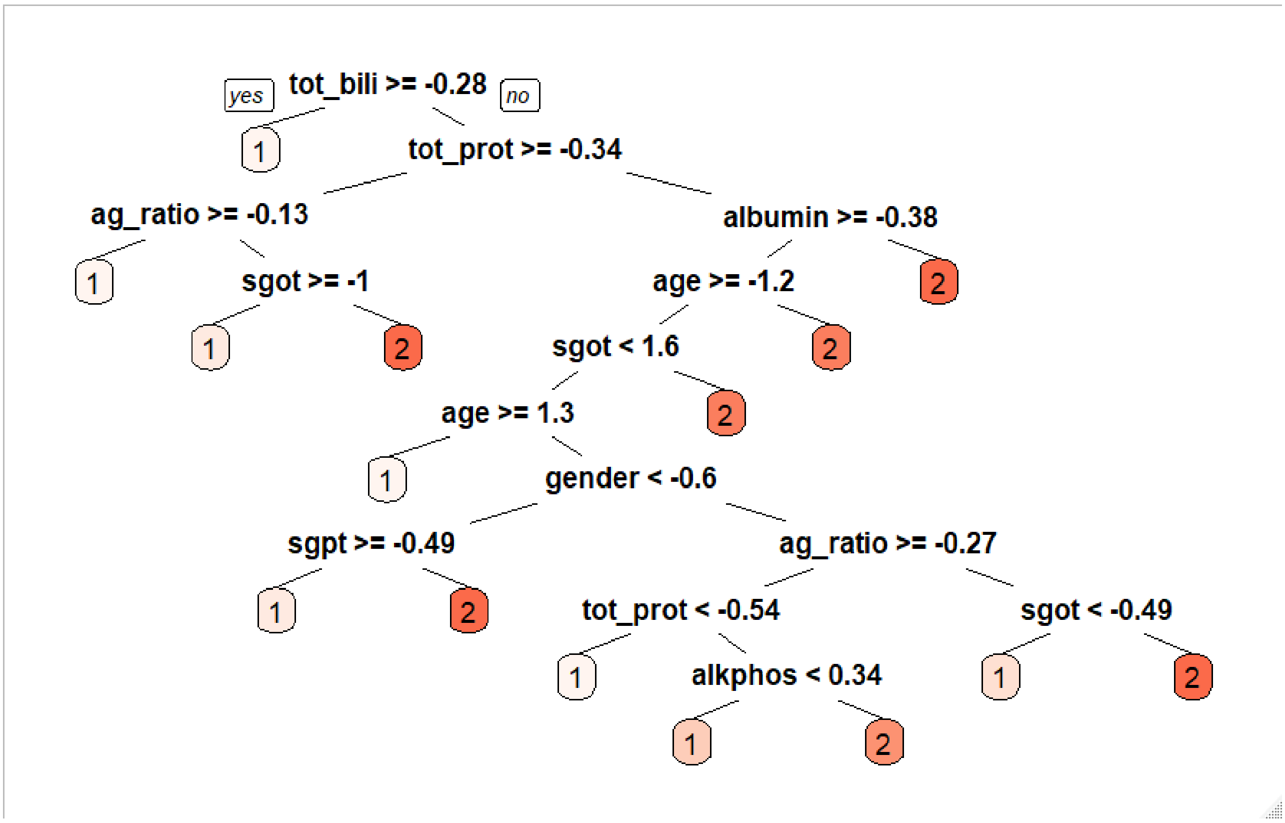
Conclusion : -

From the above table,
i) We see that the accuracy of this model is 70.69%.
ii) Precision is 0.7291 that indicates proportion of non liver patients that is correctly classified to the total no. of non liver predicted samples.
iii) Here Sensitivity(Recall) is 0.8537 that indicates proportion of non liver patients that were correctly classified to the total no. of non liver patients.
iv) Here specificity 0.3158 that indicates proportion of liver patients that were correctly classified to the total no. of liver patients.

4) Decision Tree

Decision tree is a structure consist of roots branches, leaf, nodes etc. decision tree structure is like a tree structure. Visualisation of decision tree is best for huge dataset. The structure of the Decision tree is looks like a tree structure. Here I prepare is J48 (one of the type of decision tree). J48 is advance version of C4.5. The technique of this algorithm is to use divide-and-conquer method. It uses pruning method to construct tree.

```
> model_dt <- train(
+   is_patient ~., data = training_data, method = “rpart”,
+   parms = list(split = “gini”),
+   trControl=trctrl, preProcess = c(“center”, “scale”),
+   tuneLength = 10
+ )
> prp(model_dt$finalModel, box.palette = “Reds”, tweak = 1.2)
```



```
> dt_predictions = predict(model_dt, newdata = testing_data)
> dt_result = confusionMatrix(dt_predictions, testing_data[, 11])
> accuracy[1:11, count] = as.data.frame(dt_result$byClass)
> accuracy[12, count] = as.data.frame(dt_result$overall[‘Accuracy’])
> names(accuracy)[count] = “Decision Trees”
> count = count + 1
> print(accuracy)
```

Sensitivity	1
Specificity	0
Pos Pred Value	0.672414

Neg Pred Value	NaN
Precision	0.672414
Recall	1
F1	0.804124
Prevalence	0.672414
Detection Rate	0.672414
Detection Prevalence	1
Balanced Accuracy	0.5
Accuracy	0.672414

Conclusion : -

From the above table,

i) Precision is 0.6724 that indicates proportion of non liver patients that is correctly classified to the total no. of non liver predicted samples.

ii) Here Sensitivity(Recall) is 1 that indicates proportion of non liver patients that were correctly classified to the total no. of non liver patients.

iii) Here specificity 0 that indicates proportion of liver patients that were correctly classified to the total no. of liver patients.

5) K – NEAREST NEIBHOUR

K- nearest neighbour classifier predict the class label of an unknown instance by obtaining the K- nearest neighbour’s class. The new instance will be the labelled with the class of the highest frequency form the K most similar instances. The algorithm is work as follows:

```
> library(class)
```

Normalizing the numeric data

```
> normalize <- function(x) {
+   return ((x - min(x)) /(max(x) - min(x)))}
> liver_1.df[c(1,3:10)]  <- sapply(liver_1.df[c(1,3:10)], normalize)
```

Creating a dummy variable for Gender

```
> liver_1.df$isFemale <- ifelse(liver_1.df$gender == ‘Female’,1,0)
```

Splitting into train and test

```
> train_3.df <- liver_1.df[1:as.integer(0.70*nrow(liver.df)),]
> test_3.df <- liver_1.df[-c(1:as.integer(0.70*nrow(liver.df))),]
```

For k=1

```
> pred_knn_1 <- knn(train = train_3.df[,-c(2,11)],
+                  test = test_3.df[,-c(2,11)],
+                  cl = train_3.df$is_patient ,
+                  k = 1)
> confusionMatrix(pred_knn_1, test_3.df$is_patient)
```

Confusion Matrix and Statistics

	Reference	
Prediction	0	1
0	24	36
1	25	90

Conclusion : -

In the above confusion matrix, out of 175 observations there 24 out of 49 observations and 90 out of 126 observations are correctly classified.

Accuracy	0.6514
95% CI	(0.5759, 0.7218)
No Information Rate	0.72
P-Value [Acc > NIR]	0.9807
Kappa	0.191
Mcnemar’s Test P-Value	0.2004
Sensitivity	0.4898
Specificity	0.7143
Pos Pred Value	0.4
Neg Pred Value	0.7826
Prevalence	0.28

Detection Rate	0.1371
Detection Prevalence	0.3429
Balanced Accuracy	0.602
‘Positive’ Class	0

Conclusion : -

From the above table,

i) We see that the accuracy of this model is 65.14% and it may varies between (57.59%, 72.18%).

ii) Kappa value = 0.191 means this model fair (means we may use this model for future prediction).

iii) Here Sensitivity is 0.4898 that indicates proportion of non liver patients that were correctly classified to the total no. of non liver patients.

iv) Here specificity 0.7143 that indicates proportion of liver patients that were correctly classified to the total no. of liver patients.

For k=3

```
> pred_knn_3 <- knn(train = train_3.df[,-c(2,11)],
+                  test = test_3.df[,-c(2,11)],
+                  cl = train_3.df$is_patient ,
+                  k = 3)
> #Evaluating the accuracy of KNN model
> confusionMatrix(pred_knn_3, test_3.df$is_patient)
```

Confusion Matrix and Statistics

	Reference	
Prediction	0	1
0	20	31
1	29	95

Conclusion : -

In the above confusion matrix, out of 175 observations there 20 out of 49 observations and 95 out of 126 observations are correctly classified.

Accuracy	0.6575
95% CI	(0.5817, 0.7271)
No Information Rate	0.72
P-Value [Acc > NIR]	0.9717
Kappa	0.1601
Mcnemar’s Test P-Value	0.8973
Sensitivity	0.4082
Specificity	0.7540
Pos Pred Value	0.3922
Neg Pred Value	0.7661
Prevalence	0.28

Detection Rate	0.1143
Detection Prevalence	0.2914
Balanced Accuracy	0.5811
‘Positive’ Class	0

Conclusion : -

From the above table,

i) We see that the accuracy of this model is 65.75% and it may varies between (58.17%, 72.71%).

ii) Kappa value = 0.1601 means this model fair (means we may use this model for future prediction).

iii) Here Sensitivity is 0.4082 that indicates proportion of non liver patients that were correctly classified to the total no. of non liver patients.

iv) Here specificity 0.7540 that indicates proportion of liver patients that were correctly classified to the total no. of liver patients.

For k=5

```
> pred_knn_5 <- knn(train = train_3.df[,-c(2,11)],
+                   test = test_3.df[,-c(2,11)],
+                   cl = train_3.df$is_patient ,
+                   k = 5)
> confusionMatrix(pred_knn_5, test_3.df$is_patient)
```

Confusion Matrix and Statistics

	Reference	
Prediction	0	1
0	14	27
1	35	99

Conclusion : -

In the above confusion matrix, out of 175 observations there 14 out of 49 observations and 99 out of 126 observations are correctly classified.

Accuracy	0.6457
95% CI	(0.57, 0.7164)
No Information Rate	0.72
P-Value [Acc > NIR]	0.9871
Kappa	0.0752
Mcnemar’s Test P-Value	0.3740
Sensitivity	0.2857
Specificity	0.7857
Pos Pred Value	0.3415
Neg Pred Value	0.7388
Prevalence	0.28
Detection Rate	0.08

Detection Prevalence	0.2343
Balanced Accuracy	0.5357
‘Positive’ Class	0

Conclusion : -

From the above table,

i) We see that the accuracy of this model is 64.57% and it may varies between (57%, 71.64%).

ii) Kappa value = 0.0752 means this model fair (means we may use this model for future prediction).

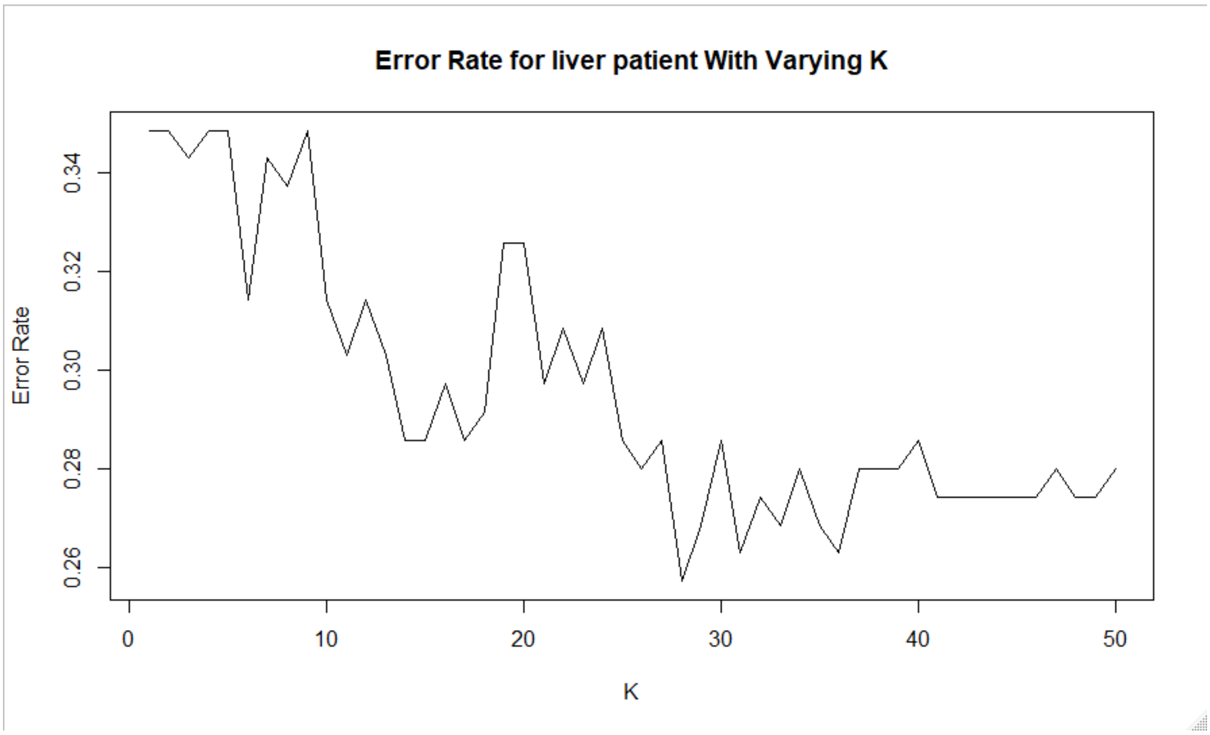
iii) Here Sensitivity is 0.2857 that indicates proportion of non liver patients that were correctly classified to the total no. of non liver patients.

iv) Here specificity 0.7857 that indicates proportion of liver patients that were correctly classified to the total no. of liver patients.

```
> liver_acc <- numeric()
> for(i in 1:50){
+   #Apply knn with k = i
+   predict <- knn(train_3.df[,-c(2,11)], test_3.df[,-c(2,11)], trai
n_3.df$is_patient, k=i)
+   liver_acc <- c(liver_acc, mean(predict == test_3.df$is_patient))
+ }
```

Plot k= 1 through 30

```
> plot(1-liver_acc,type=”l”,ylab=”Error Rate”, xlab=”K”,main=”Error
+ Rate for liver patient With Varying K”)
```



Conclusion : -

In the above plot we see that, at k=28 the error rate is minimum.

For k=28

```
> pred_knn_28 <- knn(train = train_3.df[,-c(2,11)],
+                    test = test_3.df[,-c(2,11)],
+                    cl = train_3.df$is_patient ,
+                    k = 28)
> #Evaluating the accuracy of KNN model
> confusionMatrix(pred_knn_28, test_3.df$is_patient)
```

Confusion Matrix and Statistics

	Reference	
Prediction	0	1
0	6	3
1	43	123

Conclusion : -

In the above confusion matrix, out of 175 observations there 6 out of 49 observations and 123 out of 126 observations are correctly classified.

Accuracy	0.7371
95% CI	(0.6654, 0.8007)
No Information Rate	0.72
P-Value [Acc > NIR]	0.3407
Kappa	0.1314
Mcnemar’s Test P-Value	8.192E-9
Sensitivity	0.12245
Specificity	0.98619
Pos Pred Value	0.66667
Neg Pred Value	0.74096
Prevalence	0.28
Detection Rate	0.03429
Detection Prevalence	0.05143
Balanced Accuracy	0.54932
‘Positive’ Class	0

Conclusion : -

From the above table,

i) We see that the accuracy of this model is 73.71% and it may varies between (66.54%, 80.07%).

ii) Kappa value = 0.1314 means this model fair (means we may use this model for future prediction).

iii) Here Sensitivity is 0.12245 that indicates proportion of non liver patients that were correctly classified to the total no. of non liver patients.

iv) Here specificity 0.98619 that indicates proportion of liver patients that were correctly classified to the total no. of liver patients.

6) SUPPORT VECTOR MACHINE

Support Vector Machine (SVM) is one of the most popular methods of supervised machine learning algorithm that can be used for classification and regression problems. Originally the SVM was developed for classification of linear data in two class, later it was improved that can classify the multi-classes and nonlinear data. It is based on the idea of decision hyper-planes that define the decision boundaries. Decision hyperplane separates the set of the object having a different class. In this algorithm, if we have the N-dimensional dataset (where N is the no. of the feature in a dataset) then we plot each training data points in N dimensioned space. Then we perform classification by dividing the training data points into K (where K is the number of the classes in the dataset) separate regions by hyper-planes of N different dimensions. Later to find the class of the data points, the data points are plotted in the same N-dimensional space, the points are classified into a particular class depending on the region in which the point fall. The SVM algorithm works as follows

```
> liversvm <- data.frame(liver.df)
> liversvm$is_patient <- factor(liversvm$is_patient)
> liversvm <- upSample(x = liversvm, liversvm$is_patient)
> liversvm <- liversvm[sample(nrow(liversvm)),]
> liversvm <- subset(liversvm[c(1:11)])
> intermediate2 <- createDataPartition(y = liversvm$is_patient, p=0.
+ 7, list=FALSE)
> training <- liversvm[intermediate2,]
> testing <- liversvm[-intermediate2,]
> training[[“is_patient”]] = factor(training[[“is_patient”]])
> trctrl <- trainControl(method = “repeatedcv”, number = 10, repeats
+ = 3)
> svmModel <- train(is_patient ~., data = training, method = “svmRad
+ ial”, trControl=trctrl, preProcess = c(“center”, “scale”), tuneLen
+ gth = 20)
> svmModel
```

```
Support Vector Machines with Radial Basis Function Kernel
584 samples
 10 predictor
  2 classes: ‘0’, ‘1’
Pre-processing: centered (10), scaled (10)
Resampling: Cross-Validated (10 fold, repeated 3 times)
Summary of sample sizes: 526, 526, 526, 526, 524, 526, ...
Resampling results across tuning parameters:
```

C	Accuracy	Kappa
0.25	0.717568	0.435059
0.5	0.704341	0.408648
1	0.707195	0.41432

2	0.705394	0.410774
4	0.705413	0.41084
8	0.708335	0.416676
16	0.710624	0.42116
32	0.726687	0.453275
64	0.753457	0.506837
128	0.754061	0.508016
256	0.754569	0.509019
512	0.760747	0.521364
1024	0.774511	0.548902
2048	0.776139	0.552195
4096	0.791454	0.582858
8192	0.793168	0.586302
16384	0.788696	0.577332
32768	0.789252	0.578461
65536	0.789252	0.578461
131072	0.789252	0.578461

Tuning parameter ‘sigma’ was held constant at a value of 0.1623896
The final values used for the model were sigma = 0.1623896 and C = 8192.

```
> test_pred <- predict(svmModel,newdata = testing)
> confusionMatrix(test_pred, testing$is_patient )
```

Confusion Matrix and Statistics

	Reference	
Prediction	0	1
0	112	20
1	12	104

Conclusion : -
In the above confusion matrix, out of 175 observations there 112 out of 124 observations and 104 out of 124 observations are correctly classified.

Accuracy	0.871
95% CI	(0.8228, 0.91)
No Information Rate	0.5
P-Value [Acc > NIR]	<2e-16
Kappa	0.7419
Mcnemar’s Test P-Value	0.2159
Sensitivity	0.9032
Specificity	0.8387

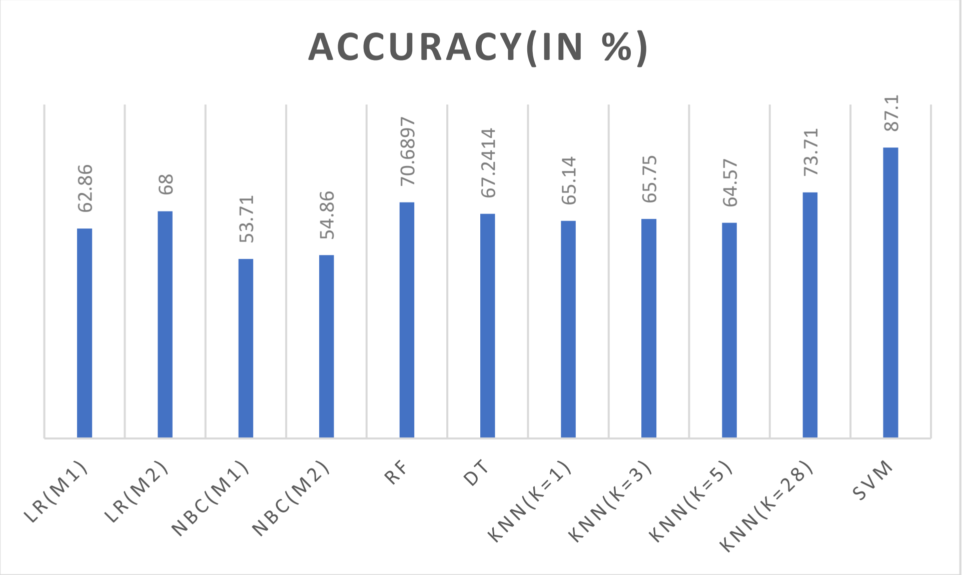
Pos Pred Value	0.8485
Neg Pred Value	0.8966
Prevalence	0.5
Detection Rate	0.4516
Detection Prevalence	0.5323
Balanced Accuracy	0.871
‘Positive’ Class	0

Conclusion : -

From the above table,
i) We see that the accuracy of this model is 87.1% and it may varies between (82.28%, 91%).
ii) Kappa value = 0.7419 means this model fair(means we may use this model for future prediction).
iii) Here Sensitivity is 0.9032 that indicates proportion of non liver patients that were correctly classified to the total no. of non liver patients.
iv) Here specificity 0.8387 that indicates proportion of liver patients that were correctly classified to the total no. of liver patients.

Comparative study of different predictive methods

	Methods	accuracy(in %)
Logistic Regression	LR(M1)	62.86
	LR(M2)	68
Naïve Bayes Classifier	NBC(M1)	53.71
	NBC(M2)	54.86
Random Forest	RF	70.6897
Decision Tree	DT	67.2414
K-Nearest Neighbour	KNN(K=1)	65.14
	KNN(K=3)	65.75
	KNN(K=5)	64.57
	KNN(K=28)	73.71
Support Vector Machine	SVM	87.1



Final Conclusion : -

In this project I used 6 classification methods, and all methods give good predictive model. Out of them support vector machine give us a best prediction model for a liver disease with accuracy 87.1%. and then second high accuracy of a method is K-Nearest Neighbours which gives 73.71% accuracy. And kappa value of all the methods are greater than 0, indicating that all the methods are good in prediction. the highest kappa value is 0.7419 of a method Support Vector Machine.

Form all the classification methods the Support Vector Machine (SVM) is best for predicting the liver disease.

REFERENCE

[1] “ Prediction and Analysis of Liver Disorder Diseases by using Data Mining Technique: Survey” from International Journal of Pure and Applied Mathematics. Volume 118 No. 9 2018, 765-770.

[2] “LIVER DISEASE PREDICTION BY USING DIFFERENT DECISION TREE TECHNIQUES” from International Journal of Data Mining & Knowledge Management Process (IJDKP) Vol.8, No.2, March 2018.

[3] “Early Detection of the Liver Disorder from Imbalance Liver Function Test Datasets” from International Journal of Innovative Technology and Exploring Engineering (IJITEE) ISSN: 2278-3075, Volume-8 Issue-4, February 2019.

[4] “An introduction to data cleaning with R” from Publisher Statistics Netherlands Henri Faasdreef 312, 2492 JP The Hague.

[5] “Liver Disease Prediction using SVM and Naïve Bayes Algorithms” from International Journal of Science, Engineering and Technology Research (IJSETR) Volume 4, Issue 4, April 2015.

[6] “Performance Analysis of Liver Disease Prediction Using Machine Learning Algorithms” from International Research Journal of Engineering and Technology (IRJET), Volume: 05 Issue: 01 | Jan-2018.

[7] “An Introduction to Statistical Learning with Applications in R” of springer. authors are Gareth James • Daniela Witten • Trevor Hastie Robert Tibshirani.

STATISTICAL SOFTWARE USE IN THS PROJECT : -
1) R-Studio
2) Excel stat
3) Ms-Excel.