

# Liver Disease Patient Classification with Indian Liver Patient Records

Eunice Ngai

2023-03-19

## Introduction

*describes the dataset and variables, and summarizes the goal of the project and key steps that were performed.* The Indian Liver Patient Records contains the records of 583 patients in India. 416 of these patients have liver disease and 167 of them are without liver disease. In this project, machine learning algorithms will be used to train models to identify liver disease patients using the same set of test results.

Data Set Variables: Age: Age of the patient, ranging from 4 to over 90 Gender: Gender of the patient Total\_Bilirubin: Total Bilirubin (direct and indirect) level in blood Direct\_Bilirubin: Direct Bilirubin level in blood Alkaline\_Phosphotase: Alamine\_Aminotransferase: Aspartate\_Aminotransferase: Total\_Protiens: Albumin: Albumin\_and\_Globulin Ratio: Dataset: label for patient with liver disease, or no disease

## Analysis

### Exploratory Analysis

By examining the summary of the data set, there are 583 records in total with 4 NA values in Albumin\_and\_Globulin\_Ratio.

```
wd <- getwd()
file <- "indian_liver_patient.csv"
patients <- read.csv(paste(wd, file, sep = "/"))
```

```
# Summary of data set
summary(patients)
```

```
##      Age      Gender      Total_Bilirubin  Direct_Bilirubin
##  Min.   : 4.00   Length:583   Min.      : 0.400   Min.      : 0.100
##  1st Qu.:33.00   Class :character  1st Qu.: 0.800   1st Qu.: 0.200
##  Median :45.00   Mode  :character  Median : 1.000   Median : 0.300
##  Mean   :44.75                      Mean   : 3.299   Mean   : 1.486
##  3rd Qu.:58.00                      3rd Qu.: 2.600   3rd Qu.: 1.300
##  Max.   :90.00                      Max.     :75.000   Max.     :19.700
##
##  Alkaline_Phosphotase Alamine_Aminotransferase Aspartate_Aminotransferase
##  Min.      : 63.0      Min.      : 10.00      Min.      : 10.0
##  1st Qu.: 175.5      1st Qu.: 23.00      1st Qu.: 25.0
```

```

## Median : 208.0      Median : 35.00      Median : 42.0
## Mean   : 290.6      Mean    : 80.71      Mean    : 109.9
## 3rd Qu.: 298.0      3rd Qu.: 60.50      3rd Qu.: 87.0
## Max.   :2110.0      Max.    :2000.00     Max.    :4929.0
##
## Total_Protiens      Albumin      Albumin_and_Globulin_Ratio      Dataset
## Min.   :2.700      Min.    :0.900      Min.    :0.3000      Min.    :1.000
## 1st Qu.:5.800      1st Qu.:2.600      1st Qu.:0.7000      1st Qu.:1.000
## Median :6.600      Median :3.100      Median :0.9300      Median :1.000
## Mean   :6.483      Mean    :3.142      Mean    :0.9471      Mean    :1.286
## 3rd Qu.:7.200      3rd Qu.:3.800      3rd Qu.:1.1000      3rd Qu.:2.000
## Max.   :9.600      Max.    :5.500      Max.    :2.8000      Max.    :2.000
##
##                                     NA's      :4

```

## Exploring Outliers

There are some extreme outliers in Alkaline\_Phosphotase, Alamine\_Aminotransferase, and Aspartate\_Aminotransferase.

Below shows the patients with highest levels of Alkaline\_Phosphotase, Alamine\_Aminotransferase, and Aspartate\_Aminotransferase respectively.

```

##      Age Gender Total_Bilirubin Direct_Bilirubin Alkaline_Phosphotase
## 234  33   Male           2.0           1.4           2110
## 118  32   Male          12.7           6.2           194
## 136  66   Male          11.3           5.6          1110
##
##      Alamine_Aminotransferase Aspartate_Aminotransferase Total_Protiens Albumin
## 234                        48                        89           6.2      3.0
## 118                      2000                      2946           5.7      3.3
## 136                      1250                      4929           7.0      2.4
##
##      Albumin_and_Globulin_Ratio Dataset
## 234                        0.9          1
## 118                        1.3          1
## 136                        0.5          1

```

Below shows the top 10 patients with highest levels of Alkaline\_Phosphotase.

```

##      Age Gender Total_Bilirubin Direct_Bilirubin Alkaline_Phosphotase
## 1    33   Male           2.0           1.4           2110
## 2    58 Female           1.7           0.8           1896
## 3    73   Male           1.9           0.7           1750
## 4    48   Male           0.7           0.1           1630
## 5    68 Female           0.6           0.1           1620
## 6    50   Male           7.3           3.6           1580
## 7    45 Female          23.3          12.8           1550
## 8     7 Female          27.2          11.8           1420
## 9    55 Female           8.2           3.9           1350
## 10   55 Female          10.9           5.1           1350
##
##      Alamine_Aminotransferase Aspartate_Aminotransferase Total_Protiens Albumin
## 1                        48                        89           6.2      3.0
## 2                        61                        83           8.0      3.9
## 3                       102                       141           5.5      2.0
## 4                        74                       149           5.3      2.0

```

## 5	95	127	4.6	2.1
## 6	88	64	5.6	2.3
## 7	425	511	7.7	3.5
## 8	790	1050	6.1	2.0
## 9	52	65	6.7	2.9
## 10	48	57	6.4	2.3
##	Albumin_and_Globulin_Ratio Dataset			
## 1	0.90	1		
## 2	0.95	1		
## 3	0.50	1		
## 4	0.60	1		
## 5	0.80	1		
## 6	0.60	2		
## 7	0.80	1		
## 8	0.40	1		
## 9	0.70	1		
## 10	0.50	1		

Below shows the top 10 patients with highest levels of Alamine\_Aminotransferase.

##	Age	Gender	Total_Bilirubin	Direct_Bilirubin	Alkaline_Phosphotase	
## 1	32	Male	12.7	6.2	194	
## 2	34	Male	6.2	3.0	240	
## 3	40	Male	1.1	0.3	230	
## 4	32	Male	15.9	7.0	280	
## 5	32	Male	18.0	8.2	298	
## 6	66	Male	11.3	5.6	1110	
## 7	40	Male	3.9	1.7	350	
## 8	34	Male	4.1	2.0	289	
## 9	34	Male	4.1	2.0	289	
## 10	7	Female	27.2	11.8	1420	
##	Alamine_Aminotransferase Aspartate_Aminotransferase Total_Protiens Albumin					
## 1			2000	2946	5.7	3.3
## 2			1680	850	7.2	4.0
## 3			1630	960	4.9	2.8
## 4			1350	1600	5.6	2.8
## 5			1250	1050	5.4	2.6
## 6			1250	4929	7.0	2.4
## 7			950	1500	6.7	3.8
## 8			875	731	5.0	2.7
## 9			875	731	5.0	2.7
## 10			790	1050	6.1	2.0
##	Albumin_and_Globulin_Ratio Dataset					
## 1			1.3	1		
## 2			1.2	1		
## 3			1.3	1		
## 4			1.0	1		
## 5			0.9	1		
## 6			0.5	1		
## 7			1.3	1		
## 8			1.1	1		
## 9			1.1	1		
## 10			0.4	1		

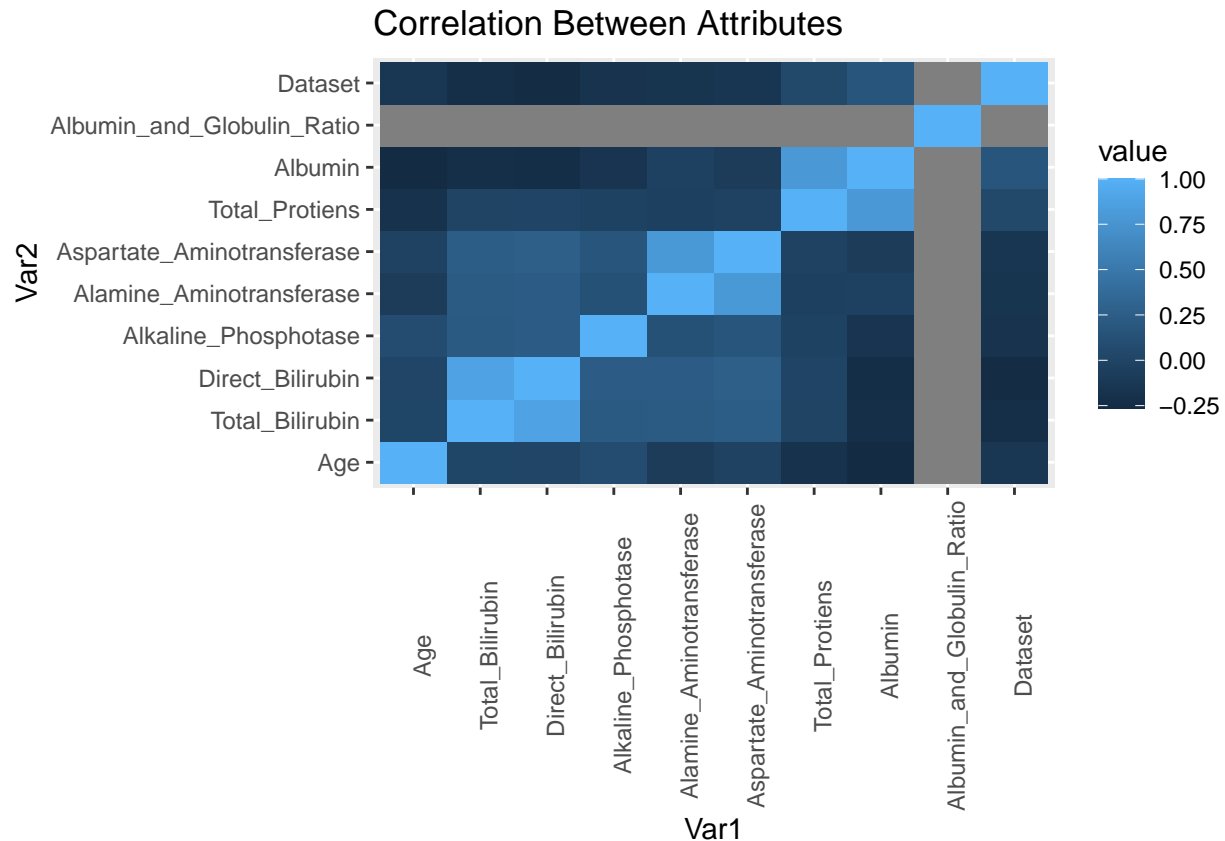
Below shows the top 10 patients with highest levels of Aspartate\_Aminotransferase.

##	Age	Gender	Total_Bilirubin	Direct_Bilirubin	Alkaline_Phosphotase	
## 1	66	Male	11.3	5.6	1110	
## 2	32	Male	12.7	6.2	194	
## 3	32	Male	15.9	7.0	280	
## 4	40	Male	3.9	1.7	350	
## 5	32	Male	18.0	8.2	298	
## 6	7	Female	27.2	11.8	1420	
## 7	40	Male	1.1	0.3	230	
## 8	39	Male	6.6	3.0	215	
## 9	34	Male	6.2	3.0	240	
## 10	60	Male	5.7	2.8	214	
##			Alamine_Aminotransferase	Aspartate_Aminotransferase	Total_Protiens	Albumin
## 1			1250	4929	7.0	2.4
## 2			2000	2946	5.7	3.3
## 3			1350	1600	5.6	2.8
## 4			950	1500	6.7	3.8
## 5			1250	1050	5.4	2.6
## 6			790	1050	6.1	2.0
## 7			1630	960	4.9	2.8
## 8			190	950	4.0	1.7
## 9			1680	850	7.2	4.0
## 10			412	850	7.3	3.2
##			Albumin_and_Globulin_Ratio	Dataset		
## 1			0.50	1		
## 2			1.30	1		
## 3			1.00	1		
## 4			1.30	1		
## 5			0.90	1		
## 6			0.40	1		
## 7			1.30	1		
## 8			0.70	1		
## 9			1.20	1		
## 10			0.78	1		

Patients with extreme values of Alkaline\_Phosphotase, Alamine\_Aminotransferase, and Aspartate\_Aminotransferase are all liver disease patients. Also the table listing top 10 patients with highest levels of Alkaline\_Phosphotase, Alamine\_Aminotransferase, and Aspartate\_Aminotransferase shows that there is a gradual change in the levels, hence the extreme outlier was not caused by an occasional typo or other random error. It is decided to keep these observations to train our model.

### Correlation Between Attributes

Some attributes are highly correlated with other attributes.



## Converting Gender and Dataset Columns into Factors

The columns **Gender**, **Dataset** can be converted into factors, the entries in these two columns are catagorical:  
 - **Gender**: 1, 2 and - **Dataset**: 1, 2.

Before converting these columns into factors, data set description was checked to identify which category (1 or 2) belongs to the patients with liver disease. According to the description, the data set contains 416 liver patient records and 167 non liver patient records. The data set has 416 records labelled as 1 and 167 records labelled as 2.

```
# 416 records numbered as 1 and 167 records numbered as 2.
summary(as.factor(patients$Dataset))
```

```
##    1    2
## 416 167
```

Therefore, the label 2 was replaced by 0 to indicate these patients had no liver disease. The final data set has 416 records numbered as 1 and 167 records numbered as 0 in the **Dataset** column.

```
# replace 2 in Dataset column by 0.
patients$Dataset[patients$Dataset == 2] <- 0

# Show final Dataset column
summary(as.factor(patients$Dataset))
```

```
##    0    1
## 167 416
```

The Gender column originally contained Male and Female entries.

```
# 142 Female patients and 441 Male patients.
summary(as.factor(patients$Gender))
```

```
## Female    Male
##    142    441
```

These entries will also be converted into 1 and 0, with 1 representing Male patients and 0 representing Female patients.

```
# replace Male with 1 and Female with 0
patients$Gender[patients$Gender == "Female"] <- 0
patients$Gender[patients$Gender == "Male"] <- 1

# converting into numeric variables
patients$Gender <- as.numeric(patients$Gender)

# Show final Gender column
summary(as.factor(patients$Gender))
```

```
##    0    1
## 142 441
```

## Handling NA Values

There are only 4 NA values in Albumin\_and\_Globulin\_Ratio column.

```
summary(is.na(patients))
```

```
##      Age      Gender      Total_Bilirubin Direct_Bilirubin
## Mode :logical   Mode :logical   Mode :logical   Mode :logical
## FALSE:583      FALSE:583      FALSE:583      FALSE:583
##
## Alkaline_Phosphotase Alamine_Aminotransferase Aspartate_Aminotransferase
## Mode :logical      Mode :logical      Mode :logical
## FALSE:583          FALSE:583          FALSE:583
##
## Total_Protiens      Albumin      Albumin_and_Globulin_Ratio Dataset
## Mode :logical      Mode :logical   Mode :logical      Mode :logical
## FALSE:583          FALSE:583      FALSE:579          FALSE:583
##
##                                     TRUE :4
```

NA values in the data set comprised of less than 1% of all the data. Also, as medical data varies among individuals with different health conditions, age and gender, therefore the rows with NA values were dropped instead of replacing by mean or other estimated statistics from the data set.

```
sum(is.na(patients))/nrow(patients)
```

```
## [1] 0.006861063
```

```
patients <- patients %>%  
  drop_na()  
summary(is.na(patients))
```

```
##      Age      Gender      Total_Bilirubin Direct_Bilirubin  
## Mode :logical   Mode :logical   Mode :logical   Mode :logical  
## FALSE:579      FALSE:579      FALSE:579      FALSE:579  
## Alkaline_Phosphotase Alamine_Aminotransferase Aspartate_Aminotransferase  
## Mode :logical      Mode :logical      Mode :logical  
## FALSE:579          FALSE:579          FALSE:579  
## Total_Protiens      Albumin      Albumin_and_Globulin_Ratio Dataset  
## Mode :logical      Mode :logical   Mode :logical      Mode :logical  
## FALSE:579          FALSE:579      FALSE:579          FALSE:579
```

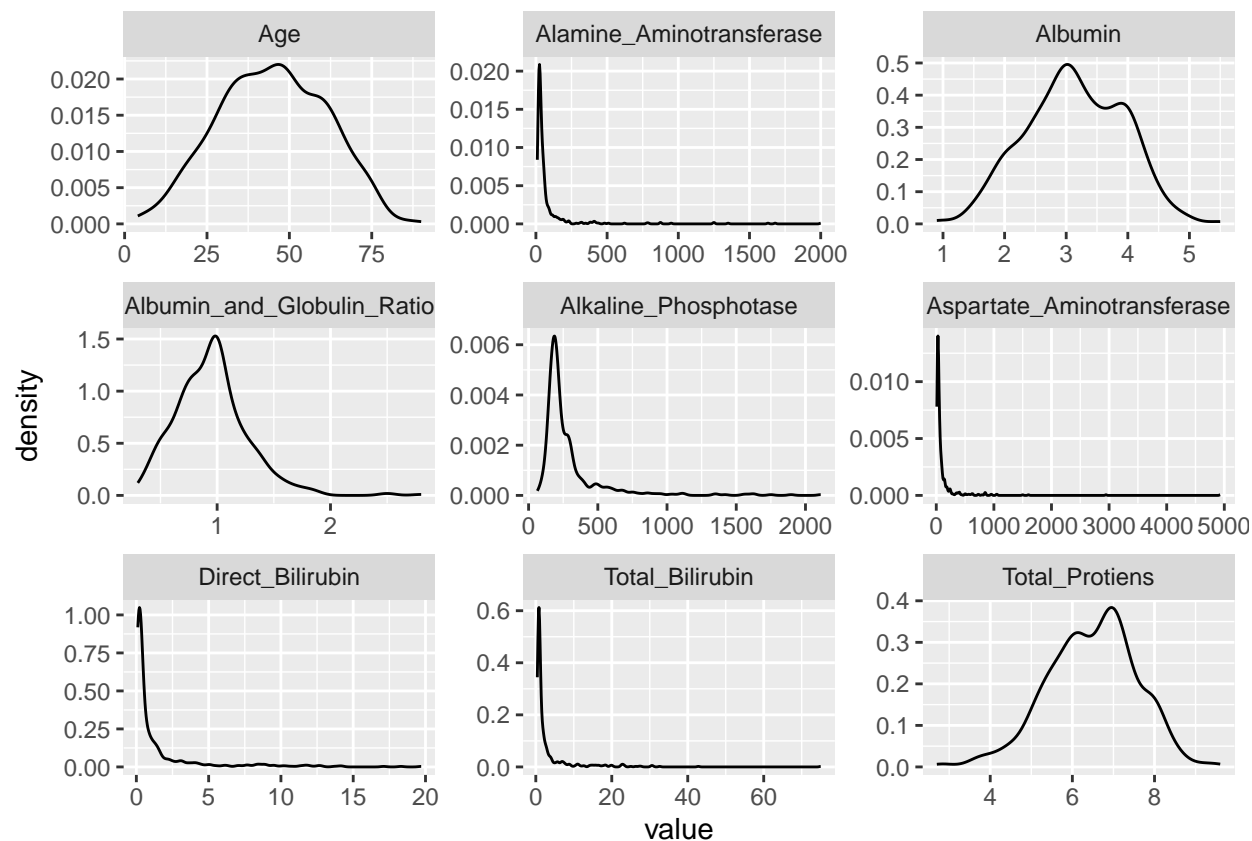
## Scaling the Variables

The means and standard deviations of each attribute. They all have different mean and standard deviation, the entries need re-scaling.

```
Attribute_SD <- sapply(patients[!names(patients) %in% c("Gender", "Dataset")], sd)  
Attribute_Mean <- sapply(patients[!names(patients) %in% c("Gender", "Dataset")], mean)  
rbind(Attribute_Mean, Attribute_SD)
```

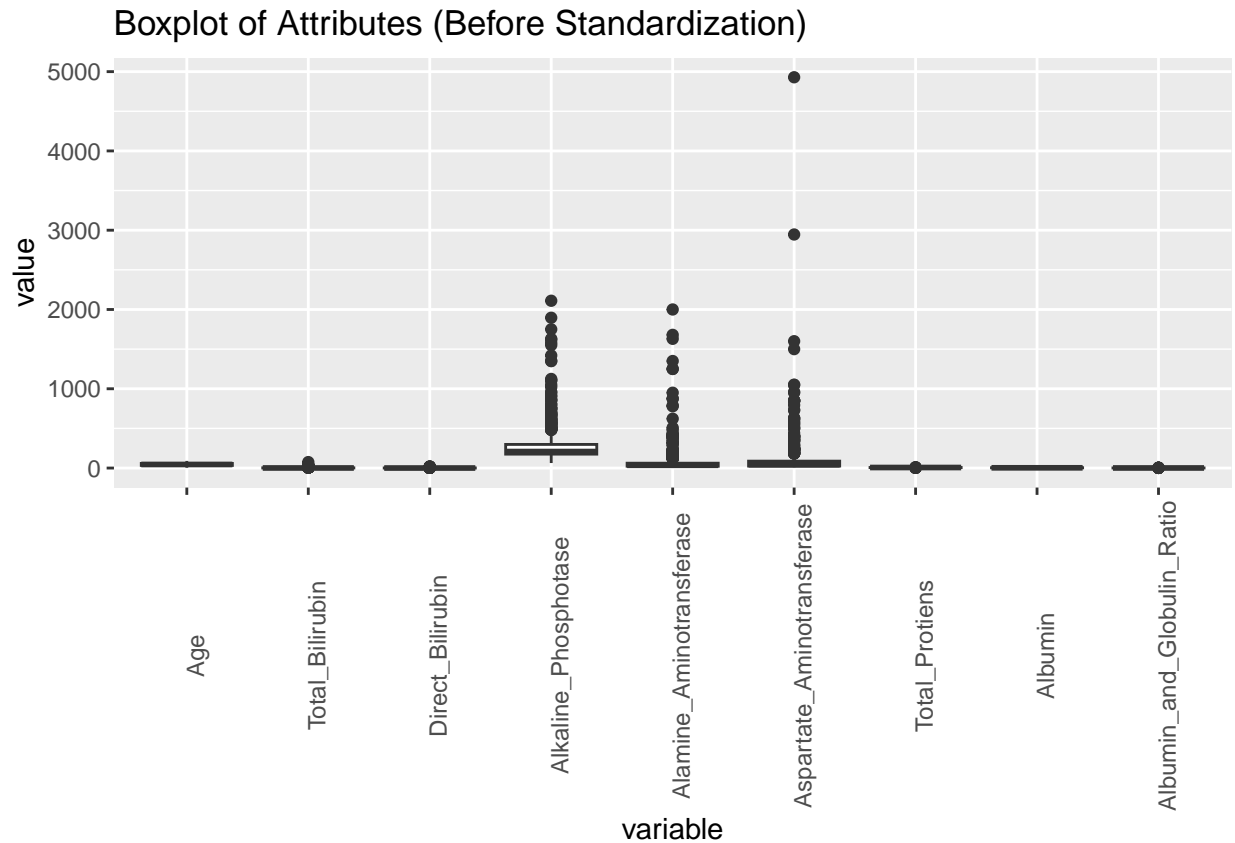
```
##      Age Total_Bilirubin Direct_Bilirubin Alkaline_Phosphotase  
## Attribute_Mean 44.78238      3.315371      1.494128      291.3661  
## Attribute_SD  16.22179      6.227716      2.816499      243.5619  
##      Alamine_Aminotransferase Aspartate_Aminotransferase  
## Attribute_Mean      81.12608      110.4145  
## Attribute_SD      183.18284      289.8500  
##      Total_Protiens      Albumin Albumin_and_Globulin_Ratio  
## Attribute_Mean      6.481693 3.1385147      0.9470639  
## Attribute_SD      1.084641 0.7944347      0.3195921
```

To choose the method of re-scaling, explore the distribution of each attribute. Some attributes such as Age, Albumin, Albumin\_and\_Globulin\_Ratio and Total\_Protiens have approximately normal distribution. However, the other features all have extreme outliers.



To look at the presence of outliers in the features, the boxplot of the data was studied. As there are outliers in our data, standardization or z-score normalization is used.





To standardize the features, each data is subtracted by the mean of that column and then divided by the standard deviation of that column.

$$X_{std} = \frac{X - \bar{X}}{\sigma_X}$$

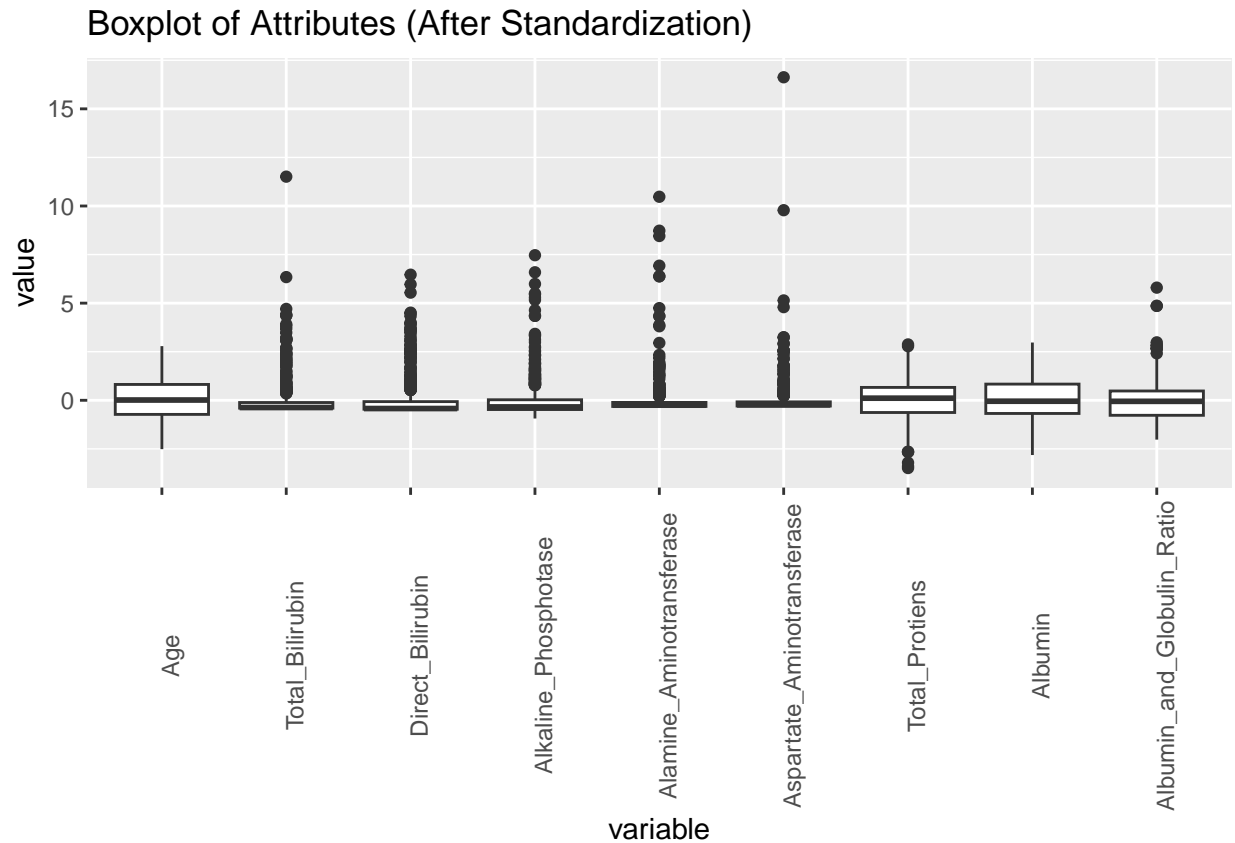
```
# Create function for standardization
standardize = function(x){
  z <- (x - mean(x)) / sd(x)
  return( z)
}

# Standardize features except gender and dataset, which are factors.
patients_std <- patients

patients_std[!names(patients_std) %in% c("Gender", "Dataset")] <- apply(patients_std[!names(patients_std) %in% c("Gender", "Dataset")], MARGIN=2, FUN=standardize)

patients_std <- as.data.frame(patients_std) # Convert back into dataframe
```

The boxplot of the data set after standardization. The attributes all have mean equal zero.



## Model Training

### Creating Train Set and Test Set

The standardized data set is split into testing set, which contains 10% of all data, and the training set will have 90% of the data.

```
set.seed(53, sample.kind = "Rounding")

test_index <- createDataPartition(patients_std$Dataset, times = 1, p = 0.1, list = FALSE)

test_att <- patients_std[test_index,] %>% select(-"Dataset") # Attributes for test set
test_dis <- patients_std[test_index,] %>% select("Dataset") # Disease indicator for test set

train_att <- patients_std[-test_index,] %>% select(-"Dataset") # Attributes for train set
train_dis <- patients_std[-test_index,] %>% select("Dataset") # Disease indicator for train set

# Convert prediction result as factor
test_dis <- as.factor(test_dis$Dataset)
train_dis <- as.factor(train_dis$Dataset)
```

Confirm the proportion of patients in test and train sets are similar.

```
tibble(
  "Patients Proportion in Train Set" = mean(as.numeric(train_dis) == 2),
  "Patients Proportion in Test Set" = mean(as.numeric(test_dis) == 2),
)

## # A tibble: 1 x 2
##   'Patients Proportion in Train Set' 'Patients Proportion in Test Set'
##                               <dbl>                               <dbl>
## 1                               0.710                               0.759
```

## Logistic Regression Model (Full Set of Attributes)

Using logistic regression model to predict the liver disease in patients, a confusion matrix is computed to illustrate the performance of the model.

```
set.seed(63, sample.kind = "Rounding") # if using R 3.6 or later

ctrl <- trainControl(method = "cv", number = 5) # train control features

train_glm <- train(train_att, train_dis, method = "glm", trControl = ctrl)
glm_preds_f <- predict(train_glm, test_att)

cfm_glm <- confusionMatrix(data = glm_preds_f, reference = test_dis) # confusion matrix comparing pred
cfm_glm
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  0  1
##           0  5  4
##           1  9 40
##
##           Accuracy : 0.7759
##           95% CI : (0.6473, 0.8749)
##           No Information Rate : 0.7586
##           P-Value [Acc > NIR] : 0.4495
##
##           Kappa : 0.3031
##
## Mcnemar's Test P-Value : 0.2673
##
##           Sensitivity : 0.35714
##           Specificity : 0.90909
##           Pos Pred Value : 0.55556
##           Neg Pred Value : 0.81633
##           Prevalence : 0.24138
##           Detection Rate : 0.08621
##           Detection Prevalence : 0.15517
##           Balanced Accuracy : 0.63312
##
##           'Positive' Class : 0
##
```

A summary table with accuracies obtained from different models is created for ease of comparison.

```
if(!exists("result_summary")){
  result_summary <- tibble("Model" = "Logistic Regression (Full attributes)",
                           "Accuracy" = mean(glm_preds_f == test_dis))
}else{
  result_summary <- rbind(result_summary,
                          c("Logistic Regression (Full attributes)",
                            mean(glm_preds_f == test_dis)))
}

result_summary
```

```
## # A tibble: 1 x 2
##   Model                      Accuracy
##   <chr>                      <dbl>
## 1 Logistic Regression (Full attributes) 0.776
```

The coefficients of the logistic regression model and their significance were examined. From the coefficients of the logistic regression model, features such as gender and Total\_Bilirubin are less significant, they have a p-value of over 0.8.

```
summary(train_glm)
```

```
##
## Call:
## NULL
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -3.1410  -1.0845   0.4037   0.9155   1.4905
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)    1.90266    0.34512   5.513 3.53e-08 ***
## Age            0.29625    0.10687   2.772 0.00557 **
## Gender         0.03435    0.24580   0.140 0.88886
## Total_Bilirubin 0.12370    0.70845   0.175 0.86139
## Direct_Bilirubin 1.30043    0.80858   1.608 0.10777
## Alkaline_Phosphotase 0.28143    0.20181   1.395 0.16315
## Alamine_Aminotransferase 2.27040    0.96782   2.346 0.01898 *
## Aspartate_Aminotransferase 0.48647    0.91842   0.530 0.59634
## Total_Protiens  0.97779    0.44044   2.220 0.02642 *
## Albumin        -1.36625    0.63303  -2.158 0.03091 *
## Albumin_and_Globulin_Ratio 0.63066    0.39140   1.611 0.10712
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##      Null deviance: 627.28  on 520  degrees of freedom
## Residual deviance: 518.47  on 510  degrees of freedom
## AIC: 540.47
```

```
##  
## Number of Fisher Scoring iterations: 7
```

### Logistic Regression Model (Reduced Set of Attributes)

Another logistic regression model is trained with gender and Total\_Bilirubin removed. The accuracy remain unchanged. the features.

```
set.seed(63, sample.kind = "Rounding") # if using R 3.6 or later  
  
# Remove gender and total_bilirubin columns in train set  
train_att_reduced <- train_att %>%  
  select(-c("Gender", "Total_Bilirubin"))  
  
# Remove gener and total_bilirubin columns in test set  
test_att_reduced <- test_att %>%  
  select(-c("Gender", "Total_Bilirubin"))  
  
ctrl <- trainControl(method = "cv", number = 5)  
  
train_glm_1 <- train(train_att_reduced, train_dis, method = "glm", trControl = ctrl)  
glm_preds_r <- predict(train_glm_1, test_att)  
  
cfm_glm <- confusionMatrix(data = glm_preds_r, reference = test_dis)  
cfm_glm
```

```
## Confusion Matrix and Statistics  
##  
##           Reference  
## Prediction  0  1  
##           0  5  4  
##           1  9 40  
##  
##           Accuracy : 0.7759  
##           95% CI : (0.6473, 0.8749)  
##    No Information Rate : 0.7586  
##    P-Value [Acc > NIR] : 0.4495  
##  
##           Kappa : 0.3031  
##  
##    McNemar's Test P-Value : 0.2673  
##  
##           Sensitivity : 0.35714  
##           Specificity : 0.90909  
##           Pos Pred Value : 0.55556  
##           Neg Pred Value : 0.81633  
##           Prevalence : 0.24138  
##           Detection Rate : 0.08621  
##    Detection Prevalence : 0.15517  
##           Balanced Accuracy : 0.63312  
##  
##           'Positive' Class : 0
```

```
##
```

```
if(!exists("result_summary")){
  result_summary <- tibble("Model" = "Logistic Regression (Reduced Attributes)",
                           "Accuracy" = mean(glm_preds_r == test_dis))
}else{
  result_summary <- rbind(result_summary,
                          c("Logistic Regression (Reduced Attributes)",
                            mean(glm_preds_r == test_dis)))
}

result_summary
```

```
## # A tibble: 2 x 2
##   Model                      Accuracy
##   <chr>                     <chr>
## 1 Logistic Regression (Full attributes) 0.775862068965517
## 2 Logistic Regression (Reduced Attributes) 0.775862068965517
```

The significance of the coefficients improved. The reduced training and testing sets with reduced attributes will be used for subsequent model training.

```
summary(train_glm_1)
```

```
##
## Call:
## NULL
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -3.1478  -1.0802   0.3979   0.9194   1.4832
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)      1.9347     0.2616   7.395 1.41e-13 ***
## Age              0.2966     0.1068   2.776  0.00550 **
## Direct_Bilirubin  1.4212     0.5362   2.651  0.00803 **
## Alkaline_Phosphotase 0.2827     0.2022   1.398  0.16220
## Alamine_Aminotransferase 2.2783     0.9675   2.355  0.01854 *
## Aspartate_Aminotransferase 0.4907     0.9193   0.534  0.59347
## Total_Protiens    0.9777     0.4408   2.218  0.02656 *
## Albumin          -1.3685     0.6329  -2.162  0.03059 *
## Albumin_and_Globulin_Ratio 0.6335     0.3904   1.623  0.10466
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##      Null deviance: 627.28  on 520  degrees of freedom
## Residual deviance: 518.54  on 512  degrees of freedom
## AIC: 536.54
##
## Number of Fisher Scoring iterations: 7
```

## K-Nearest Neighbors Model

The K-nearest neighbor algorithm was used to train the model. The accuracy obtained was higher than that obtained by the logistic regression model. The reduced set of attribute was used in training the model.

```
set.seed(63, sample.kind = "Rounding")

ctrl <- trainControl(method = "cv", number = 5)

train_knn <- train(train_att_reduced, train_dis, method = "knn",
                  trControl = ctrl)

knn_preds <- predict(train_knn, test_att)

train_knn$bestTune
```

```
##    k
## 1 5
```

The accuracy obtained was higher than the logistic regression model.

```
cfm_knn <- confusionMatrix(data = knn_preds, reference = test_dis)
cfm_knn
```

```
## Confusion Matrix and Statistics
##
##              Reference
## Prediction  0  1
##           0  4  7
##           1 10 37
##
##              Accuracy : 0.7069
##              95% CI : (0.5727, 0.8191)
##      No Information Rate : 0.7586
##      P-Value [Acc > NIR] : 0.8580
##
##              Kappa : 0.1366
##
##  Mcnemar's Test P-Value : 0.6276
##
##              Sensitivity : 0.28571
##              Specificity : 0.84091
##              Pos Pred Value : 0.36364
##              Neg Pred Value : 0.78723
##              Prevalence : 0.24138
##              Detection Rate : 0.06897
##      Detection Prevalence : 0.18966
##              Balanced Accuracy : 0.56331
##
##              'Positive' Class : 0
##
```

```

if(!exists("result_summary")){
  result_summary <- tibble("Model" = "KNN (Reduced Attributes)",
                           "Accuracy" = mean(knn_preds == test_dis))
}else{
  result_summary <- rbind(result_summary,
                          c("KNN (Reduced Attributes)",
                            mean(knn_preds == test_dis)))
}

result_summary

```

```

## # A tibble: 3 x 2
##   Model                      Accuracy
##   <chr>                     <chr>
## 1 Logistic Regression (Full attributes) 0.775862068965517
## 2 Logistic Regression (Reduced Attributes) 0.775862068965517
## 3 KNN (Reduced Attributes) 0.706896551724138

```

## Random Forest Model

The Random Forest was used to train the model. The accuracy obtained was higher than that obtained by the logistic regression model but lower than the KNN model. The reduced set of attribute was used in training the model.

```

set.seed(63, sample.kind = "Rounding")

train_rf <- train(train_att_reduced, train_dis, method = "rf", ntree = 10)

rf_preds <- predict(train_rf, test_att)

cfm_rf <- confusionMatrix(data = rf_preds, reference = test_dis)

cfm_rf

```

```

## Confusion Matrix and Statistics
##
##           Reference
## Prediction  0   1
##           0 10   4
##           1   4 40
##
##           Accuracy : 0.8621
##           95% CI : (0.7462, 0.9385)
##           No Information Rate : 0.7586
##           P-Value [Acc > NIR] : 0.03995
##
##           Kappa : 0.6234
##
## Mcnemar's Test P-Value : 1.00000
##
##           Sensitivity : 0.7143
##           Specificity : 0.9091

```



```
##          Pos Pred Value : 0.7143
##          Neg Pred Value : 0.9091
##          Prevalence : 0.2414
##          Detection Rate : 0.1724
##          Detection Prevalence : 0.2414
##          Balanced Accuracy : 0.8117
##
##          'Positive' Class : 0
##
```

```
if(!exists("result_summary")){
  result_summary <- tibble("Model" = "Random Forest (Reduced Attributes)",
                           "Accuracy" = mean(rf_preds == test_dis))
}else{
  result_summary <- rbind(result_summary,
                          c("Random Forest (Reduced Attributes)",
                            mean(rf_preds == test_dis)))
}

result_summary
```

```
## # A tibble: 4 x 2
##   Model                                     Accuracy
##   <chr>                                     <chr>
## 1 Logistic Regression (Full attributes) 0.775862068965517
## 2 Logistic Regression (Reduced Attributes) 0.775862068965517
## 3 KNN (Reduced Attributes) 0.706896551724138
## 4 Random Forest (Reduced Attributes) 0.862068965517241
```

## K-means Model

K-means was used to train the model. The accuracy obtained was the lowest among all other models.

```
#predict function taking in k_means object
predict_kmeans <- function(x, k) {
  centers <- k$centers      # extract cluster centers
  # calculate distance to cluster centers
  distances <- sapply(1:nrow(x), function(i){
    apply(centers, 1, function(y) dist(rbind(x[i,], y)))
  })
  max.col(-t(distances)) # select cluster with min distance to center
}

#k_means model building
set.seed(63, sample.kind = "Rounding")
k <- kmeans(train_att_reduced, centers = 3, nstart = 25)
kmeans_preds <- ifelse(predict_kmeans(test_att_reduced, k) == 1, "1", "0")
cfm_kmeans <- confusionMatrix(data = as.factor(kmeans_preds), reference = test_dis)

cfm_kmeans
```

```
## Confusion Matrix and Statistics
```

```
##
##           Reference
## Prediction 0  1
##           0  8 21
##           1  6 23
##
##           Accuracy : 0.5345
##           95% CI : (0.3987, 0.6666)
##           No Information Rate : 0.7586
##           P-Value [Acc > NIR] : 0.999944
##
##           Kappa : 0.069
##
## Mcnemar's Test P-Value : 0.007054
##
##           Sensitivity : 0.5714
##           Specificity : 0.5227
##           Pos Pred Value : 0.2759
##           Neg Pred Value : 0.7931
##           Prevalence : 0.2414
##           Detection Rate : 0.1379
##           Detection Prevalence : 0.5000
##           Balanced Accuracy : 0.5471
##
##           'Positive' Class : 0
##
```

The result summary table becomes

```
## # A tibble: 5 x 2
##   Model                      Accuracy
##   <chr>                     <chr>
## 1 Logistic Regression (Full attributes) 0.775862068965517
## 2 Logistic Regression (Reduced Attributes) 0.775862068965517
## 3 KNN (Reduced Attributes) 0.706896551724138
## 4 Random Forest (Reduced Attributes) 0.862068965517241
## 5 K-Means (Reduced Attributes) 0.53448275862069
```

## Building an Ensemble

An ensemble with all the models trained to explore if a model with higher accuracy could be created.

## Variable Importance in Difference Models

The variable importance of different trained models was examined and found that different models have different variable importance.

Variable importance of logistic regression (reduced attributes):

```
## glm variable importance
##
##           Overall
```

```
## Age                                100.00
## Direct_Bilirubin                  94.41
## Alamine_Aminotransferase          81.21
## Total_Protiens                    75.11
## Albumin                           72.64
## Albumin_and_Globulin_Ratio        48.56
## Alkaline_Phosphotase              38.53
## Aspartate_Aminotransferase        0.00
```

Variable importance of K-nearest neighbours (reduced Attributes):

```
## ROC curve variable importance
##
##                                     Importance
## Aspartate_Aminotransferase        100.00
## Alamine_Aminotransferase          99.52
## Direct_Bilirubin                  98.53
## Alkaline_Phosphotase              84.43
## Albumin_and_Globulin_Ratio        55.85
## Albumin                           48.86
## Age                               32.69
## Total_Protiens                    0.00
```

Variable importance of Random Forest (reduced Attributes):

```
## rf variable importance
##
##                                     Overall
## Alkaline_Phosphotase              100.00
## Aspartate_Aminotransferase        80.28
## Age                               78.71
## Direct_Bilirubin                  68.36
## Alamine_Aminotransferase          59.46
## Total_Protiens                    11.62
## Albumin                           4.69
## Albumin_and_Globulin_Ratio        0.00
```

## Accuracy of the Ensemble

The ensemble model takes the prediction of each models, and return the prediction result as 1 (liver disease patient) , when three or more models predicted the result as 1 (liver disease patinet). The threshold of “three or more models” was found by tuning, it was found that this threshold results in the highest accuracy.

```
en_pred <- ifelse(
  as.numeric(kmeans_preds == 1)+
  as.numeric(glm_preds_r == 1)+
  as.numeric(knn_preds == 1)+
  as.numeric(rf_preds == 1)
  > 2,
  1,0
)
```

```
cfm_en <- confusionMatrix(data = as.factor(en_pred), reference = test_dis)

cfm_en
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction 0  1
##           0  8 11
##           1  6 33
##
##           Accuracy : 0.7069
##           95% CI : (0.5727, 0.8191)
##           No Information Rate : 0.7586
##           P-Value [Acc > NIR] : 0.858
##
##           Kappa : 0.2865
##
## Mcnemar's Test P-Value : 0.332
##
##           Sensitivity : 0.5714
##           Specificity : 0.7500
##           Pos Pred Value : 0.4211
##           Neg Pred Value : 0.8462
##           Prevalence : 0.2414
##           Detection Rate : 0.1379
##           Detection Prevalence : 0.3276
##           Balanced Accuracy : 0.6607
##
##           'Positive' Class : 0
##
```

The summary table of results becomes:

```
## # A tibble: 6 x 2
##   Model                      Accuracy
##   <chr>                     <chr>
## 1 Logistic Regression (Full attributes) 0.775862068965517
## 2 Logistic Regression (Reduced Attributes) 0.775862068965517
## 3 KNN (Reduced Attributes) 0.706896551724138
## 4 Random Forest (Reduced Attributes) 0.862068965517241
## 5 K-Means (Reduced Attributes) 0.53448275862069
## 6 Ensemble 0.706896551724138
```

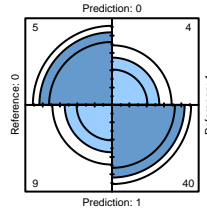
## Results

It was found that the highest accuracy obtained in the ensemble was still lower than that obtained by Random Forest model. As the same accuracy could be obtained by the Random Forest model alone, it is decided that Random Forest model will be used as the final model, as it can achieve higher level of accuracy.

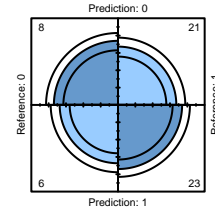
## Confusion Matrix of Different Models

The confusion matrix of different models are visualized below for comparison.

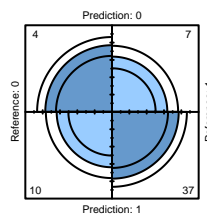
### Logistic Regression Model



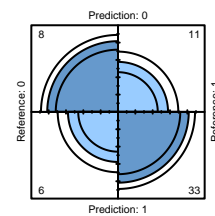
### K-Means Model



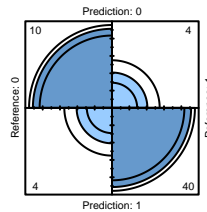
### KNN Model



### Ensemble



### Random Forest Model



The final model, random forest model obtained an accuracy of 0.8621, sensitivity of 0.7143 and specificity of 0.9091.

Given that the prevalence of liver disease patients in the test set was high, at around 0.7586, the quality of of the model cannot be judged by the accuracy achieved finally by the model (0.8621) alone. The Cohen's Kappa score of 0.6234, which indicates that there is a substantial agreement between the actual prevalence and the predicted outcome.

## Conclusion

In this study, machine learning algorithms “Logistic Regression”, “K-Nearest Neighbors”, “Random Forest” and “K-Means” were used to predict the liver disease patients. Other algorithms such as XGBoost and Vanilla Neural Networks could be adopted to see if higher accuracy could be achieved. Also, as there are a lot of outliers in the data set, having a larger data set could eliminate the effect of outliers and also improve the model accuracy as there are more training data.

## Reference

Data set source: <https://www.kaggle.com/datasets/uciml/indian-liver-patient-records>

Data set description: [https://archive.ics.uci.edu/ml/datasets/ILPD+\(Indian+Liver+Patient+Dataset\)](https://archive.ics.uci.edu/ml/datasets/ILPD+(Indian+Liver+Patient+Dataset))

Statistic Knowledge: <https://www.statology.org/>