# Predicting Liver Disease: Capstone Submission

Viswanathan Rajendran

1/10/2020

## 1. Executive Summary

### 1.1 Introduction

This paper describes an attempt to build an algorithm to predict the likely incidence of Liver Disease in patients. Our exercise examines data from liver patients concentrating on relationships between a key list of liver enzymes, proteins, age and gender using them to try and predict the likeliness of liver disease.

Models which use existing data to estimate risk of serious diseases can add substantial value to already over burdened healthcare systems. Even if our model can only achieve a limited predictive ability, it can still serve as an aid to determin when to look deeper. Patient datasets can be used to evaluate prediction algorithms in an effort to reduce burden on doctors.

### 1.2 Introducing the Dataset

This data set contains 416 liver patient records and 167 non liver patient records collected from North East of Andhra Pradesh, India. The "Dataset" column is a class label used to divide groups into liver patient (liver disease) or not (no disease). This data set contains 441 male patient records and 142 female patient records.

Any patient whose age exceeded 89 is listed as being of age "90".

Columns: -> Age of the patient -> Gender of the patient -> Total Bilirubin -> Direct Bilirubin -> Alkaline Phosphotase -> Alamine Aminotransferase -> Aspartate Aminotransferase -> Total Protiens -> Albumin -> Albumin and Globulin Ratio -> Dataset: field used to split the data into two sets (patient with liver disease, or no disease)

### 1.3 Acknowledgment

This dataset was downloaded from the UCI ML Repository:

Lichman, M. (2013). UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science.

### 1.4 Objectives

The objective of this exercise is to train a machine learning model in R that can use these patient records to help "predict" to determine which patients have liver disease and which ones do not.

As an India based student, the motivation for selecting this data set arises from an interest in exploring data sets that originate from India.

### 1.5 Outcomes

This exercise adopts multiple different methods (Logistic Regression, LDA, QDA, KNN, Classification Tree, and Random Forest) to determine which patients have liver disease and which dont. Amongst all these methods, we find that the Classification Tree offers the greatest level of accuracy at 0.763, and the QDA offers the lowest level of accuracy at 0.458.

# 2. Methods and Analysis

### 2.1 Data download and preparation

As a first step, we include all requisite libraries, and access the downloaded data in the file "indian_liver_patient.csv" from the local folder.

```
if(!require(ggplot2)) install.packages("ggplot2", repos = "http://cran.us.r-project.org")
if(!require(lubridate)) install.packages("lubridate", repos = "http://cran.us.r-project.org")
if(!require(tidyr)) install.packages("tidyr", repos = "http://cran.us.r-project.org")
if(!require(stringr)) install.packages("stringr", repos = "http://cran.us.r-project.org")
if(!require(tidyverse)) install.packages("tidyverse", repos = "http://cran.us.r-project.org")
if(!require(caret)) install.packages("caret", repos = "http://cran.us.r-project.org")
if(!require(data.table)) install.packages("data.table", repos = "http://cran.us.r-project.org")
if(!require(dplyr)) install.packages("tidyverse", repos = "http://cran.us.r-project.org")
if(!require(randomForest)) install.packages("randomForest", repos = "http://cran.us.r-project.org")
if(!require(rpart)) install.packages("rpart", repos = "http://cran.us.r-project.org")

# Read the data file from local folder
sourcedata <- read.csv("./indian_liver_patient.csv")
```

### 2.2 Data cleaning

The data in the file has already been made available in a cleaned and ready to use format, and hence does not need any further need for cleaning. However, to facilitate the eventual predictive modeling, it will help us to convert the numeric Dataset field into a factor.

```
# Convert to factor
sourcedata$Dataset<-factor(sourcedata$Dataset, levels = c(1,2), labels=c("No", "Yes"))
```

### 2.3 Data exploration, visualization and key insights gained

The following sets of analyses provide a good overview into the structure of the overall dataset.

```
# High level exploratory analysis
head(sourcedata)
```

```
##   Age Gender Total_Bilirubin Direct_Bilirubin Alkaline_Phosphotase
## 1  65 Female             0.7              0.1                  187
## 2  62   Male            10.9              5.5                  699
## 3  62   Male             7.3              4.1                  490
## 4  58   Male             1.0              0.4                  182
## 5  72   Male             3.9              2.0                  195
## 6  46   Male             1.8              0.7                  208
##   Alamine_Aminotransferase Aspartate_Aminotransferase Total_Protiens Albumin
## 1                       16                         18            6.8     3.3
## 2                       64                        100            7.5     3.2
## 3                       60                         68            7.0     3.3
## 4                       14                         20            6.8     3.4
## 5                       27                         59            7.3     2.4
## 6                       19                         14            7.6     4.4
```

```
##   Albumin_and_Globulin_Ratio Dataset
## 1                       0.90      No
## 2                       0.74      No
## 3                       0.89      No
## 4                       1.00      No
## 5                       0.40      No
## 6                       1.30      No
```

```
table(sourcedata$Dataset)
```

```
##
##  No Yes
## 416 167
```

```
str(sourcedata)
```

```
## 'data.frame':    583 obs. of  11 variables:
##  $ Age                       : int  65 62 62 58 72 46 26 29 17 55 ...
##  $ Gender                    : Factor w/ 2 levels "Female","Male": 1 2 2 2 2 2 1 1 2 2 ...
##  $ Total_Bilirubin           : num  0.7 10.9 7.3 1 3.9 1.8 0.9 0.9 0.9 0.7 ...
##  $ Direct_Bilirubin          : num  0.1 5.5 4.1 0.4 2 0.7 0.2 0.3 0.3 0.2 ...
##  $ Alkaline_Phosphotase      : int  187 699 490 182 195 208 154 202 202 290 ...
##  $ Alamine_Aminotransferase  : int  16 64 60 14 27 19 16 14 22 53 ...
##  $ Aspartate_Aminotransferase: int  18 100 68 20 59 14 12 11 19 58 ...
##  $ Total_Protiens            : num  6.8 7.5 7 6.8 7.3 7.6 7 6.7 7.4 6.8 ...
##  $ Albumin                   : num  3.3 3.2 3.3 3.4 2.4 4.4 3.5 3.6 4.1 3.4 ...
##  $ Albumin_and_Globulin_Ratio: num  0.9 0.74 0.89 1 0.4 1.3 1 1.1 1.2 1 ...
##  $ Dataset                   : Factor w/ 2 levels "No","Yes": 1 1 1 1 1 1 1 1 2 1 ...
```

```
summary(sourcedata)
```

```
##       Age           Gender    Total_Bilirubin  Direct_Bilirubin
##  Min.   : 4.00   Female:142   Min.   : 0.400   Min.   : 0.100
##  1st Qu.:33.00   Male  :441   1st Qu.: 0.800   1st Qu.: 0.200
##  Median :45.00                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             No :416
##  1st Qu.:5.800   1st Qu.:2.600   1st Qu.:0.7000             Yes:167
##  Median :6.600   Median :3.100   Median :0.9300
##  Mean   :6.483   Mean   :3.142   Mean   :0.9471
```

```
##  3rd Qu.:7.200   3rd Qu.:3.800   3rd Qu.:1.1000
##  Max.   :9.600   Max.   :5.500   Max.   :2.8000
##                                  NA's   :4
```

### 2.4 Modeling approach adopted

Our overall objective for this exercise is to attempt multiple different supervised machine learning methods (Logistic Regression, LDA, QDA, KNN, Classification Tree, and Random Forest), and then identify the most accurate method that can determine which patients have liver disease and which dont.

#### 2.4.1 Preparing the training and test data sets

The first step in the modeling approach is to define the training and test data sets. The code below splits 90% of the total data into the training set, and the remaining 10% into the test set.

```
# Create test and train datasets
set.seed(1, sample.kind="Rounding")
samplesize <- floor(0.9 * nrow(sourcedata))
index <- sample(seq_len(nrow(sourcedata)), size = samplesize)
train_set <- sourcedata[index, ]
test_set = sourcedata[-index, ]
```

#### 2.4.2 Building the first model: Logistic Regression Model

The first model we develop in this exercise is the Logistic Regression model - based on the caret package.

```
# Logistic Regression Model
set.seed(1, sample.kind="Rounding")
log_model <- train(Dataset ~ Total_Bilirubin + Alkaline_Phosphotase + Alamine_Aminotransferase + Total_
log_preds <- predict(log_model, test_set)
log_accuracy <- mean(log_preds == test_set$Dataset)
summary(log_model)
```

```
##
## Call:
## NULL
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.7438  -0.9062  -0.3862   1.0366   3.0682
##
## Coefficients:
##                              Estimate Std. Error z value Pr(>|z|)
## (Intercept)                 2.6519104  0.8587950   3.088 0.002015 **
## Total_Bilirubin            -0.2664271  0.0920632  -2.894 0.003804 **
## Alkaline_Phosphotase       -0.0009519  0.0007466  -1.275 0.202347
## Alamine_Aminotransferase   -0.0144746  0.0040446  -3.579 0.000345 ***
## Total_Protiens             -0.6083659  0.1927357  -3.156 0.001597 **
## Albumin                     0.8233482  0.2709110   3.039 0.002372 **
## Age                        -0.0196299  0.0067228  -2.920 0.003502 **
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 632.91  on 523  degrees of freedom
## Residual deviance: 515.79  on 517  degrees of freedom
## AIC: 529.79
##
## Number of Fisher Scoring iterations: 7
```

```
model_results <- tibble(method = "Logistic Regression", Accuracy = log_accuracy)
```

We publish the results of the modeling below:

```
# Outcomes of Linear Determinant Analysis approach (LDA)
summary(log_model)
```

```
##
## Call:
## NULL
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.7438  -0.9062  -0.3862   1.0366   3.0682
##
## Coefficients:
##                              Estimate Std. Error z value Pr(>|z|)
## (Intercept)                 2.6519104  0.8587950   3.088 0.002015 **
## Total_Bilirubin            -0.2664271  0.0920632  -2.894 0.003804 **
## Alkaline_Phosphotase       -0.0009519  0.0007466  -1.275 0.202347
## Alamine_Aminotransferase   -0.0144746  0.0040446  -3.579 0.000345 ***
## Total_Protiens             -0.6083659  0.1927357  -3.156 0.001597 **
## Albumin                     0.8233482  0.2709110   3.039 0.002372 **
## Age                        -0.0196299  0.0067228  -2.920 0.003502 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 632.91  on 523  degrees of freedom
## Residual deviance: 515.79  on 517  degrees of freedom
## AIC: 529.79
##
## Number of Fisher Scoring iterations: 7
```

```
model_results
```

```
## # A tibble: 1 x 2
##   method              Accuracy
##   <chr>                  <dbl>
## 1 Logistic Regression    0.712
```

### 2.4.3 Linear Determinant Analysis

The third model in our exercise is based on the Linear Determinant Analysis approach.

```
# LDA Model
set.seed(1, sample.kind="Rounding")
lda_model <- train(Dataset ~ Total_Bilirubin + Alkaline_Phosphotase + Alamine_Aminotransferase + Total_
lda_preds <- predict(lda_model, test_set)
lda_accuracy <- mean(lda_preds == test_set$Dataset)
model_results<-add_row(model_results,method = "Linear Discriminant Analysis", Accuracy = lda_accuracy)
```

We publish the results of the modeling below:

```
# Outcomes of Linear Determinant Analysis (LDA)
summary(lda_model)
```

```
##             Length Class      Mode
## prior        2     -none-     numeric
## counts       2     -none-     numeric
## means       12     -none-     numeric
## scaling      6     -none-     numeric
## lev          2     -none-     character
## svd          1     -none-     numeric
## N            1     -none-     numeric
## call         3     -none-     call
## xNames       6     -none-     character
## problemType  1     -none-     character
## tuneValue    1     data.frame list
## obsLevels    2     -none-     character
## param        0     -none-     list
```

```
model_results
```

```
## # A tibble: 2 x 2
##   method                         Accuracy
##   <chr>                             <dbl>
## 1 Logistic Regression               0.712
## 2 Linear Discriminant Analysis      0.729
```

### 2.4.4 Quadratic Determinant Analysis

The next model in our exercise is based on the Quadratic Determinant Analysis approach.

```
# QDA Model
set.seed(1, sample.kind="Rounding")
qda_model <- train(Dataset ~ Total_Bilirubin + Alkaline_Phosphotase + Alamine_Aminotransferase + Total_
qda_preds <- predict(qda_model, test_set)
qda_accuracy <- mean(qda_preds == test_set$Dataset)
model_results<-add_row(model_results,method = "Quadratic Discriminant Analysis", Accuracy = qda_accuracy
```

We publish the results of the modeling below:

```r
# Outcomes of Quadratic Determinant Analysis (QDA)
summary(qda_model)
```

```
##             Length Class      Mode
## prior        2     -none-     numeric
## counts       2     -none-     numeric
## means       12     -none-     numeric
## scaling     72     -none-     numeric
## ldet         2     -none-     numeric
## lev          2     -none-     character
## N            1     -none-     numeric
## call         3     -none-     call
## xNames       6     -none-     character
## problemType  1     -none-     character
## tuneValue    1     data.frame list
## obsLevels    2     -none-     character
## param        0     -none-     list
```

```r
model_results
```

```
## # A tibble: 3 x 2
##   method                            Accuracy
##   <chr>                                <dbl>
## 1 Logistic Regression                  0.712
## 2 Linear Discriminant Analysis         0.729
## 3 Quadratic Discriminant Analysis      0.458
```

### 2.4.5 k-Nearest Neighbors Approach

The next model in our exercise is based on the KNN approach.

```r
# KNN Model
set.seed(1, sample.kind="Rounding")
knn_model <- train(Dataset ~ Total_Bilirubin + Alkaline_Phosphotase + Alamine_Aminotransferase + Total_
                   tuneGrid = data.frame(k = seq(1, 50, 2)))
knn_preds <- predict(knn_model, test_set)
knn_accuracy <- mean(knn_preds == test_set$Dataset)
model_results<-add_row(model_results,method = "KNN", Accuracy = knn_accuracy)
```

We publish the results of the modeling below:

```r
# Outcomes of kNN
summary(knn_accuracy)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.7119  0.7119  0.7119  0.7119  0.7119  0.7119
```

```r
model_results
```

```
## # A tibble: 4 x 2
##   method                             Accuracy
##   <chr>                                 <dbl>
## 1 Logistic Regression                   0.712
## 2 Linear Discriminant Analysis          0.729
## 3 Quadratic Discriminant Analysis       0.458
## 4 KNN                                   0.712
```

### 2.4.6 Classification Trees

The next model in our exercise is based on the Classification Tree approach.

```
# Classification Tree Model
set.seed(1, sample.kind="Rounding")
tree_model <- train(Dataset ~ Total_Bilirubin + Alkaline_Phosphotase + Alamine_Aminotransferase + Total_
                    tuneGrid = data.frame(cp = seq(0, 0.05, 0.002)))
tree_preds <- predict(tree_model, test_set)
tree_accuracy <- mean(tree_preds == test_set$Dataset)
model_results<-add_row(model_results,method = "Classification Tree", Accuracy = tree_accuracy)
```

We publish the results of the modeling below:

```
# Outcomes of Classification Tree approach
summary(tree_accuracy)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.7627  0.7627  0.7627  0.7627  0.7627  0.7627
```

```
model_results
```

```
## # A tibble: 5 x 2
##   method                             Accuracy
##   <chr>                                 <dbl>
## 1 Logistic Regression                   0.712
## 2 Linear Discriminant Analysis          0.729
## 3 Quadratic Discriminant Analysis       0.458
## 4 KNN                                   0.712
## 5 Classification Tree                   0.763
```

### 2.4.7 Random Forest

The final model in our exercise is based on the Random Forest approach.

```
# Random Forest Model
set.seed(1, sample.kind="Rounding")
forest_model <- train(Dataset ~ Total_Bilirubin + Alkaline_Phosphotase + Alamine_Aminotransferase + Tota
                      tuneGrid = data.frame(mtry = seq(1:7)))
forest_preds <- predict(forest_model, test_set)
forest_accuracy <- mean(forest_preds == test_set$Dataset)
model_results<-add_row(model_results,method = "Random Forest", Accuracy = forest_accuracy)
```

We publish the results of the modeling below:

```
# Outcomes of Random Forest approach
summary(forest_accuracy)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.7288  0.7288  0.7288  0.7288  0.7288  0.7288
```

```
model_results
```

```
## # A tibble: 6 x 2
##   method                            Accuracy
##   <chr>                                <dbl>
## 1 Logistic Regression                  0.712
## 2 Linear Discriminant Analysis         0.729
## 3 Quadratic Discriminant Analysis      0.458
## 4 KNN                                  0.712
## 5 Classification Tree                  0.763
## 6 Random Forest                        0.729
```

# 3. Results

The overall results from the exercise are summarized below:

```
model_results
```

```
## # A tibble: 6 x 2
##   method                            Accuracy
##   <chr>                                <dbl>
## 1 Logistic Regression                  0.712
## 2 Linear Discriminant Analysis         0.729
## 3 Quadratic Discriminant Analysis      0.458
## 4 KNN                                  0.712
## 5 Classification Tree                  0.763
## 6 Random Forest                        0.729
```

As can be seen from the results, the Classification Tree offers the greatest level of accuracy at 0.763, and the QDA offers the lowest level of accuracy at 0.458.

# 4. Conclusions

### 4.1 Report Summary

The objective of this exercise is to train a machine learning model in R that can use patient records to help "predict" which patients have liver disease and which ones do not. This exercise adopted multiple different methods (Logistic Regression, LDA, QDA, KNN, Classification Tree, and Random Forest) to determine the same. Amongst all these methods, we find that the Classification Tree offers the greatest level of accuracy at 0.763, and the QDA offers the lowest level of accuracy at 0.458.

### 4.2 Limitations

This exercise is based on a limited dataset from a narrow geography: 416 liver patient records and 167 non liver patient records collected from North East of Andhra Pradesh, India. While this sample size is adequate for the academic purpose of testing various algorithms with limited computing power, in a real world scenario - developing a robust disease prediction tool will require significantly larger amounts of patient data.

### 4.3 Future Work

A logical next step from this paper should be an attempt to replicate the analysis on a much larger dataset, and then attempt options for further optimization to drive overall accuracy.

As discussed in the beginning, aggregate analysis from large collections of patient data records can add substantial value to overburdened healthcare systems. With this overall direction, efforts such as this can start delivering substantial cost and quality of life benefits to healthcare systems and patients alike.