HarvardX: PH125.9x Data Science: Capstone Project

Choose Your Own: Predicting Indian Liver Disease

Wilfredo A. de Vera

June 21, 2020

1. Executive Summary

This is the second part of the HarvardX PH125.9X Capstone Project - Choose Your Own. While the first capstone project was to build a recommender system using the movielens dataset, this second capstone will focus on the Indian Liver Patient Records. The dataset for this project is included in Kaggle's curated list of datasets and is already cleaned and ready for machine learning analysis. As well, it is available for download from https://www.kaggle.com/uciml/indian-liver-patient-records.

The objective of this study is to predict liver disease based of the following 10 independent variables: a.) Age of the patient; b.) Gender of the patient; c.) Total Bilirubin; d.) Direct Bilirubin; e.) Alkaline Phosphotase; f.) Alamine Aminotransferase; g.) Aspartate Aminotransferase; h.) Total Protiens; i.) Albumin; and j.) Albumin and Globulin Ratio.

The original dataset contains a total of 583 observations - 416 of which have no liver disease while the remaining 167 have liver disease. And of the 167 liver disease cases, 50 were females and 117 were males.

```
## disease
##   N   Y
## 416 167
```

```
##         disease
##            N   Y
##   Female  92  50
##   Male   324 117
```

After exploratory data analysis, wrangling and cleaning, the liver dataset shrunk to 579 observations due to 4 missing (NA) values detected in the Albumin_and_Globulin_Ratio variable. The missing values from the four observations in this variable were predicted in the main R code using the mice package that employs multivariate imputation by chained equations (MICE) algorithm; but then the decision has been made to simply remove them since they only comprise 0.69% of the original dataset. The resulting clean liver dataset, which comprises 579 observations, was then split 70%-30% corresponding to train_set and test_set, with 404 and 175 observations, respectively.

The train_set dataset was fit and trained on the twelve (12) algorithms that were built, which include: logistic regression, neural network/deep learning, random forests, gradient boosting machine (GBM), support vector machine (SVM), and naive bayes from the h2o library. In addition to these h2o models, classification tree, C5.0 classification tree, evolutionary classification tree, logistic model-based recursive partitioning, and principal components analysis were likewise developed to determine and select the best algorithm.

Subsequently, the performance of the 12 algorithms were assessed and reported using the test_set dataset in terms of accuracy, sensitivity, specificity, and precision. Since the presence of liver disease is being predicted

based of the 10 independent variables, it is necessary to particularly measure specificity, or the true negative rate, which is the proportion of True Negative/(True Negative + False Positive) found in the second column of the confusion matrix. Hence, in this context, the true negatives (TN) pertain to the presence of liver disease, while true positives (TP) pertain to its absence.

Finally, the naive bayes model was determined to be the best algorithm that yielded the highest specificity of 0.88 and raw prediction accuracy of 0.59.

2. Exploratory Data Analysis (EDA) and Wrangling

Initial analysis on the original dataset was conducted in terms of generating summary statistics and visualization. The dataset was then wrangled and cleaned after detecting the presence of NA values. Then further data analysis was performed on the clean dataset in terms of checking correlation, principal components, variable importance, multi-collinearity (variance inflation factor), normality, linearity, and outliers (chi-squared test). Observations and insights gained are noted in each step during the analysis.

2.1 Generate summary statistics

```
##                              n    mean      sd    max  min  range nunique
## Age                        583  44.746  16.190   90.0  4.0   86.0      72
## Total_Bilirubin            583   3.299   6.210   75.0  0.4   74.6     113
## Direct_Bilirubin           583   1.486   2.808   19.7  0.1   19.6      80
## Alkaline_Phosphotase       583 290.576 242.938 2110.0 63.0 2047.0     263
## Alamine_Aminotransferase   583  80.714 182.620 2000.0 10.0 1990.0     152
## Aspartate_Aminotransferase 583 109.911 288.919 4929.0 10.0 4919.0     177
## Total_Protiens             583   6.483   1.085    9.6  2.7    6.9      58
## Albumin                    583   3.142   0.796    5.5  0.9    4.6      40
## Albumin_and_Globulin_Ratio 579   0.947   0.320    2.8  0.3    2.5      70
## Dataset                    583   1.286   0.452    2.0  1.0    1.0       2
##                            nzeros   iqr lowerbound upperbound noutlier kurtosis
## Age                             0  25.0      -4.50      95.50        0   -0.574
## Total_Bilirubin                 0   1.8      -1.90       5.30       84   36.699
## Direct_Bilirubin                0   1.1      -1.45       2.95       81   11.196
## Alkaline_Phosphotase            0 123.0      -9.00     482.50       66   17.520
## Alamine_Aminotransferase        0  37.2     -32.88     116.38       73   49.954
## Aspartate_Aminotransferase      0  62.0     -68.00     180.00       66  149.095
## Total_Protiens                  0   1.4       3.70       9.30        8    0.210
## Albumin                         0   1.2       0.80       5.60        0   -0.404
## Albumin_and_Globulin_Ratio      0   0.4       0.10       1.70       10    3.222
## Dataset                         0   1.0      -0.50       3.50        0   -1.114
##                            skewness  mode miss miss%     1%     5%   25%    50%
## Age                         -0.0292  60.0    0 0.000  9.640  18.00  33.0  45.00
## Total_Bilirubin              4.8823   0.8    0 0.000  0.582   0.60   0.8   1.00
## Direct_Bilirubin             3.1959   0.2    0 0.000  0.100   0.10   0.2   0.30
## Alkaline_Phosphotase         3.7458 198.0    0 0.000 97.820 137.00 175.5 208.00
## Alamine_Aminotransferase     6.5155  25.0    0 0.000 11.820  15.00  23.0  35.00
## Aspartate_Aminotransferase  10.4920  23.0    0 0.000 12.000  15.10  25.0  42.00
## Total_Protiens              -0.2842   7.0    0 0.000  3.682   4.61   5.8   6.60
## Albumin                     -0.0435   3.0    0 0.000  1.482   1.80   2.6   3.10
## Albumin_and_Globulin_Ratio   0.9872   1.0    4 0.686  0.386   0.50   0.7   0.93
## Dataset                      0.9423   1.0    0 0.000  1.000   1.00   1.0   1.00
##                               75%    95%     99%
## Age                          58.0  72.00   75.00
```

```
## Total_Bilirubin              2.6  16.35   28.20
## Direct_Bilirubin             1.3   8.40   12.96
## Alkaline_Phosphotase       298.0 698.10 1555.40
## Alamine_Aminotransferase    60.5 232.00 1004.00
## Aspartate_Aminotransferase  87.0 400.90  976.20
## Total_Protiens               7.2   8.10    8.62
## Albumin                      3.8   4.39    4.90
## Albumin_and_Globulin_Ratio   1.1   1.50    1.81
## Dataset                      2.0   2.00    2.00
```

```
##          n miss miss% unique     top5levels:count
## Gender 583    0     0      2 Male:441, Female:142
```

According to the Kaggle documentation, the records were collected from North East of Andhra Pradesh, India, and that the "Dataset" column is a class label used to divide groups into liver patient (liver disease) or not (no disease). This is the dependent variable but was encoded as integer class.

Moreover, the above summary statistics reveal 4 missing values in Albumin_and_Globulin_Ratio variable which comprise only 0.686% of the dataset. The 4 specific observations with truncated variables are presented briefly below:

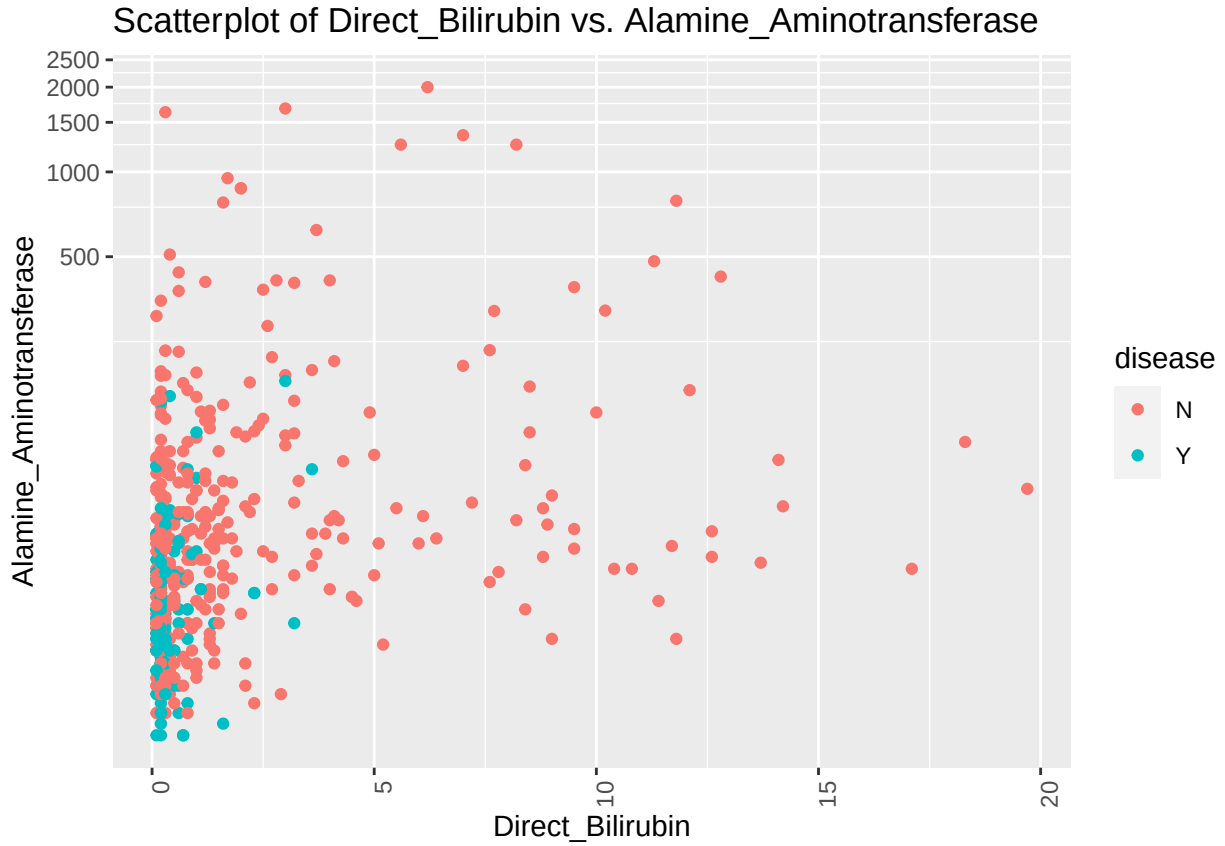
```
##                            miss miss%
## Albumin_and_Globulin_Ratio    4 0.686
```

```
##   Age Gender Albumin_and_Globulin_Ratio disease
## 1  45      1                        NA       N
## 2  51      2                        NA       N
## 3  35      1                        NA       Y
## 4  27      2                        NA       Y
```

2.2 Visualization

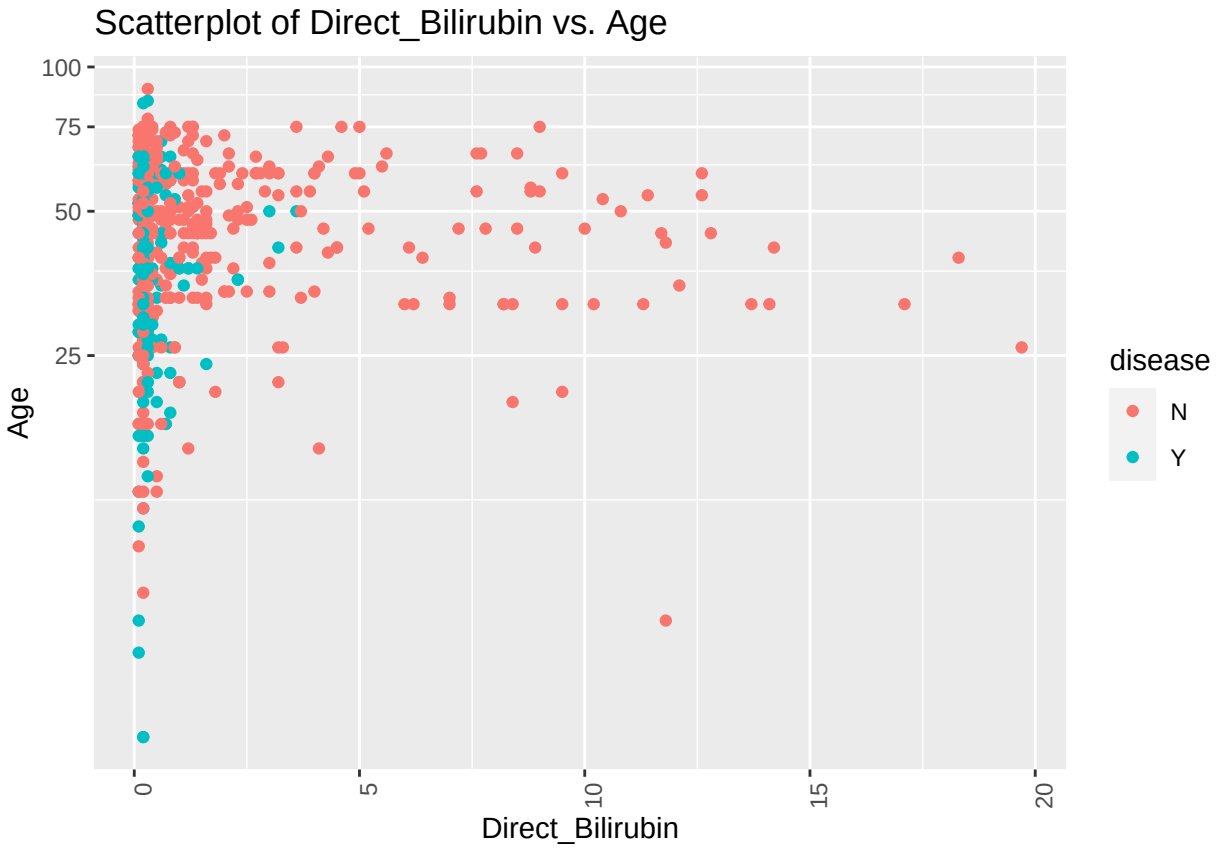
Since there are 10 independent variables to visualize, we may be looking at 45 possible combinations of variable-pairs to plot. However in this section, we will only present 6 scatterplots of the variables Direct_Bilirubin, Alamine_Aminotransferase, Age, and Alkaline_Phosphotase, which were deemed to be important later on as indicated in Section 2.6 - Variable importance.

2.2.1 Plot of Direct_Bilirubin vs. Alamine_Aminotransferase



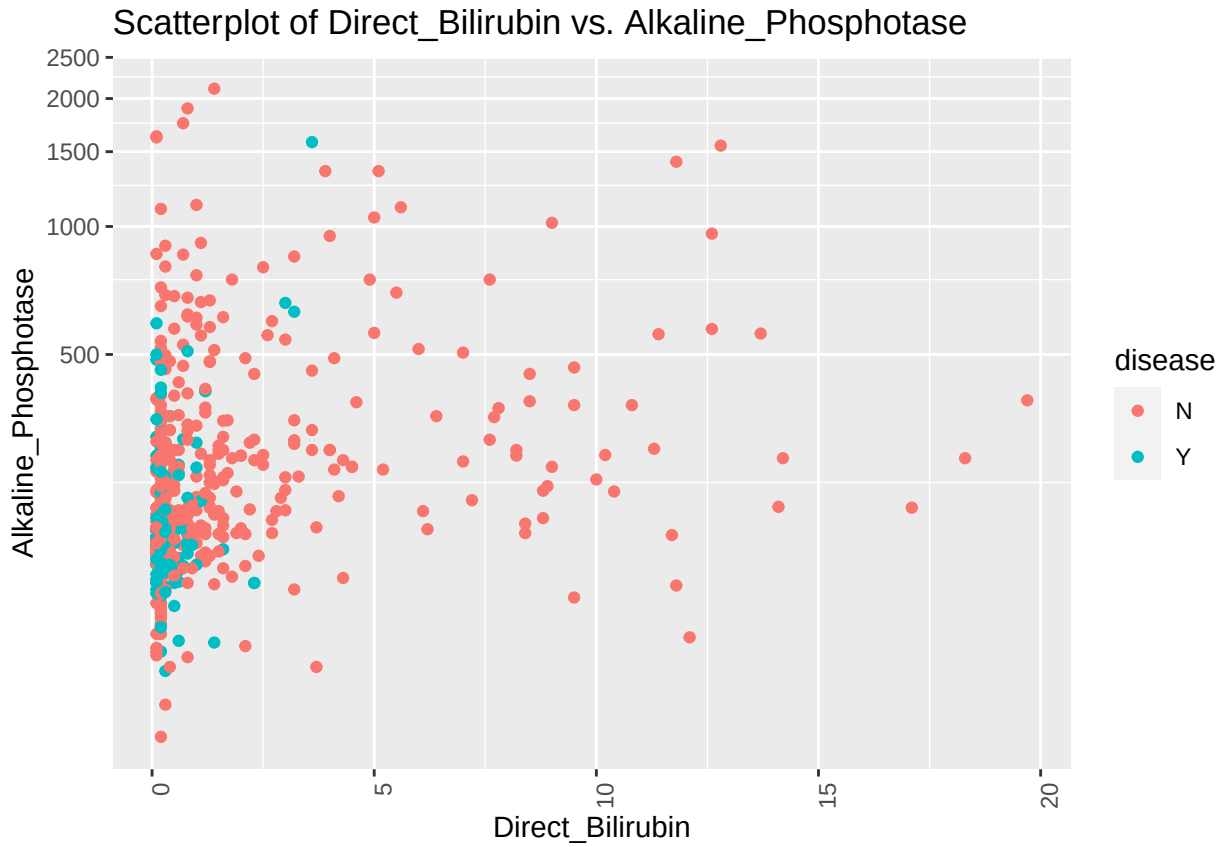
It may be observed that there is no liver disease when Direct_Bilirubin is above 3 and Alamine_Aminotransferase is above 250.

2.2.2 Plot of Direct_Bilirubin vs. Age



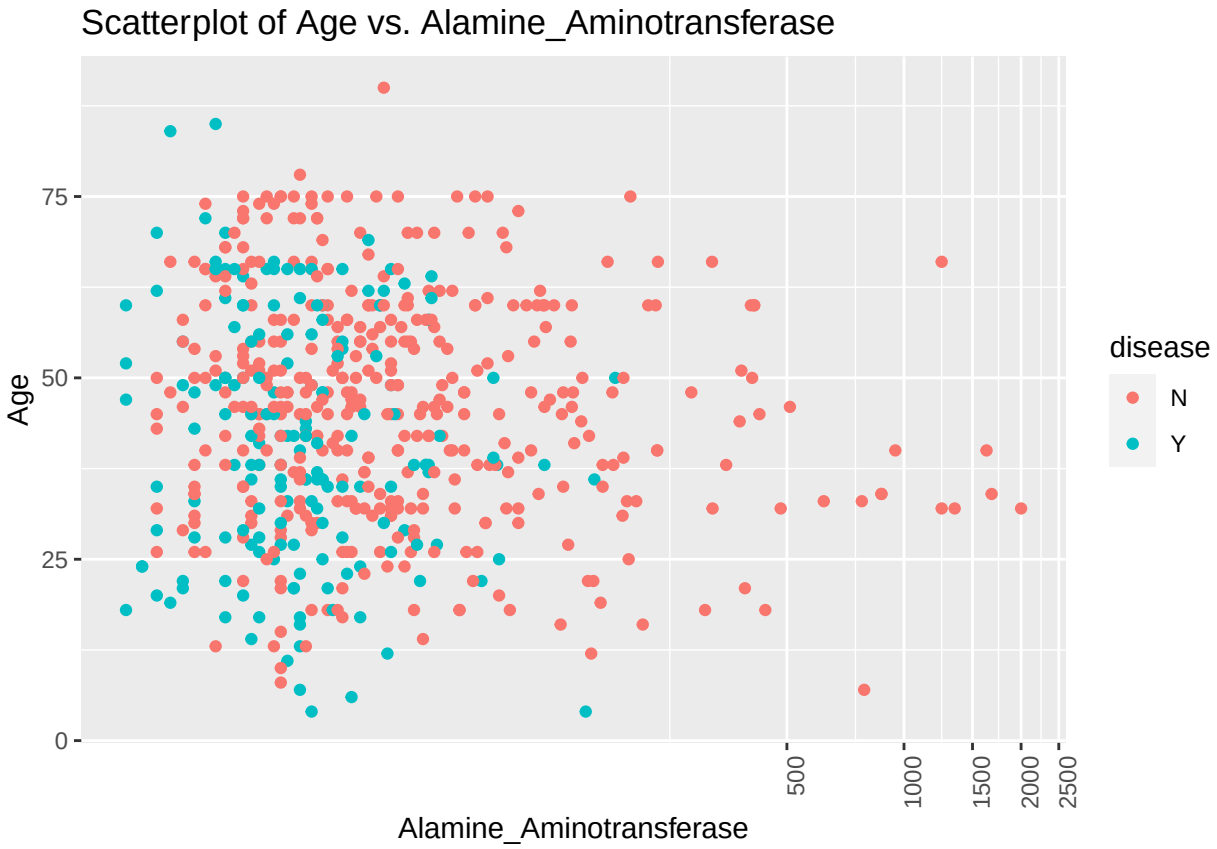
It may be observed that liver disease may occur regardless of Age. But then it still shows that there is no liver disease when Direct_Bilirubin is above 3.

2.2.3 Plot of Direct_Bilirubin vs. Alkaline_Phosphotase



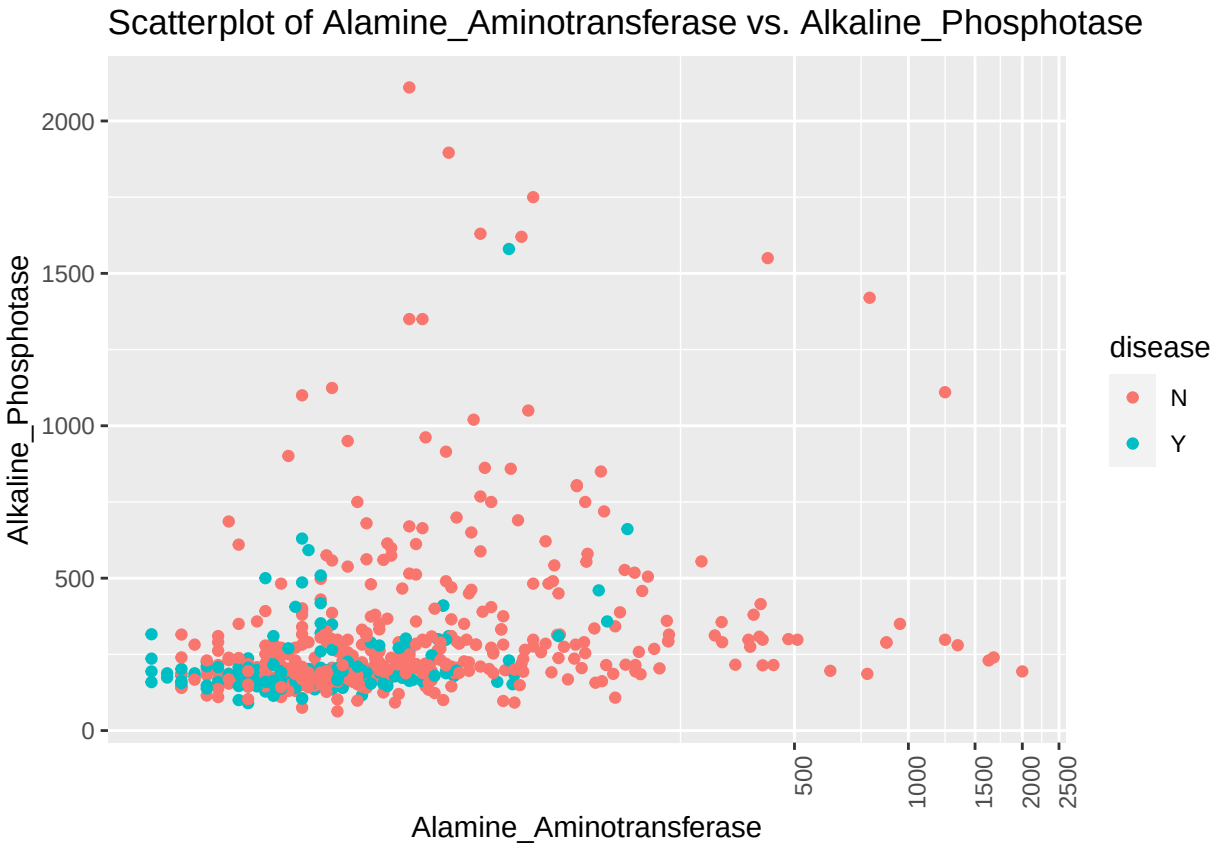
It may be observed that liver disease may be rare when Alkaline_Phosphotase is above 700 and that it is absent when Direct_Bilirubin is above 3.

2.2.4 Plot of Alamine_Aminotransferase vs. Age



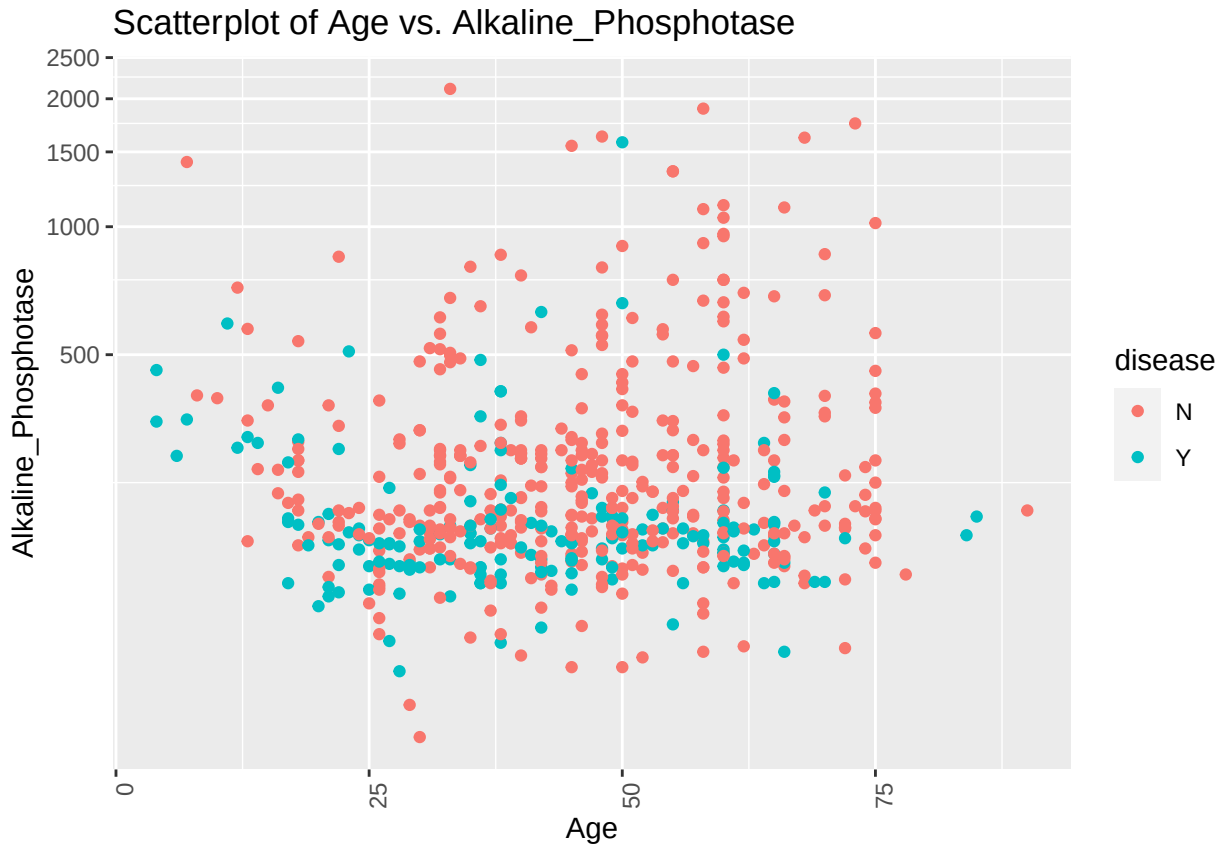
It appears that liver disease occurs regardless of Age but is absent when Alamine_Aminotransferase is above 250.

2.2.5 Plot of Alamine_Aminotransferase vs. Alkaline_Phosphotase



It may be observed that liver disease is rare when Alkaline_Phosphotase is above 700 and absent when Alamine_Aminotransferase is above 250.

2.2.6 Plot of Age vs. Alkaline_Phosphotase



It is evident that liver disease may be present regardless of Age but is rare when Alkaline_Phosphotase is above 700.

2.3 Data cleaning and wrangling

In this section, the original liver dataset will be cleaned of missing values (NAs). As previously noted in Section 2.1, there were only 4 observations whose Albumin_and_Globulin_Ratio is NA, and that the proportion with respect to the dependent variable are the same - i.e. 2 records for the presence and 2 records for the absence of liver disease as shown in Section 2.1. Since the amount of missing data is small at 0.686%, we could just easily remove them. The resulting clean liver dataset would yield 579 observations with 11 variables.

```
## 'data.frame':    579 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                   : int  1 1 1 1 1 1 1 1 2 1 ...
```

After cleaning the dataset of NAs, the dependent variable "Dataset" will be renamed as integer variable "y". As well, from the same "Dataset" variable, we will introduce a factor variable "disease" with values "N" and Y", which corresponds to no-liver and with-liver disease, respectively. The rationale for this is that, while most binary classification algorithms require them to be factor, there are certain algorithms like principal components analysis (PCA) in Section 3.11 that require them to be numeric. Hence, there is no harm in maintaining both class types and that it could be easily referenced during the modeling process. Hence, from the cleaned and wrangled liver dataset, the distribution of liver disease by gender will be:

```
## [1] "liver_clean dataset: 579 obs  12 vars"
```

```
##
##            N   Y
##   Female  91  49
##   Male   323 116
```

From here on, we will check for correlation, principal components, variable importance, multi-collinearity, normality, linearity, and outliers using the cleaned liver dataset.

2.4 Correlation

```
##                               Age Total_Bilirubin Direct_Bilirubin
## Age                       1.00000         0.01100         6.78e-03
## Total_Bilirubin           0.01100         1.00000         8.74e-01
## Direct_Bilirubin          0.00678         0.87448         1.00e+00
## Alkaline_Phosphotase      0.07888         0.20574         2.34e-01
## Alamine_Aminotransferase -0.08780         0.21338         2.33e-01
## Aspartate_Aminotransferase -0.02050       0.23732         2.57e-01
## Total_Protiens            -0.18625       -0.00791         3.27e-05
## Albumin                   -0.26421       -0.22209        -2.28e-01
## Albumin_and_Globulin_Ratio -0.21641      -0.20627        -2.00e-01
##                           Alkaline_Phosphotase Alamine_Aminotransferase
## Age                                     0.0789                 -0.08780
## Total_Bilirubin                         0.2057                  0.21338
## Direct_Bilirubin                        0.2340                  0.23318
## Alkaline_Phosphotase                    1.0000                  0.12478
## Alamine_Aminotransferase                0.1248                  1.00000
## Aspartate_Aminotransferase              0.1666                  0.79186
## Total_Protiens                         -0.0271                 -0.04243
## Albumin                                -0.1634                 -0.02866
## Albumin_and_Globulin_Ratio             -0.2342                 -0.00237
##                           Aspartate_Aminotransferase Total_Protiens Albumin
## Age                                          -0.0205      -1.86e-01 -0.2642
## Total_Bilirubin                               0.2373      -7.91e-03 -0.2221
## Direct_Bilirubin                              0.2570       3.27e-05 -0.2284
## Alkaline_Phosphotase                          0.1666      -2.71e-02 -0.1634
## Alamine_Aminotransferase                      0.7919      -4.24e-02 -0.0287
## Aspartate_Aminotransferase                    1.0000      -2.58e-02 -0.0849
## Total_Protiens                               -0.0258       1.00e+00  0.7831
## Albumin                                      -0.0849       7.83e-01  1.0000
## Albumin_and_Globulin_Ratio                   -0.0700       2.35e-01  0.6896
```

```
##                             Albumin_and_Globulin_Ratio
## Age                                            -0.21641
## Total_Bilirubin                                -0.20627
## Direct_Bilirubin                               -0.20012
## Alkaline_Phosphotase                           -0.23417
## Alamine_Aminotransferase                       -0.00237
## Aspartate_Aminotransferase                     -0.07004
## Total_Protiens                                  0.23489
## Albumin                                         0.68963
## Albumin_and_Globulin_Ratio                      1.00000
```

The following points could be observed from the correlation table:

1.) Direct_Bilirubin is correlated with Total_Bilirubin at 0.87448
2.) Alamine_Aminotransferase is correlated with Aspartate_Aminotransferase at 0.7919
3.) Albumin is correlated with Albumin_and_Globulin_Ratio at 0.68963
4.) Total_Protiens is correlated with Albumin at 0.7831

The remaining chemicals in the correlation table yielded either a slightly positive or negative correlation.

2.5 Principal components

```
## Importance of components:
##                           PC1   PC2   PC3   PC4    PC5    PC6    PC7    PC8
## Standard deviation     1.666 1.424 1.170 1.030 0.9584 0.8963 0.8130 0.4511
## Proportion of Variance 0.278 0.203 0.137 0.106 0.0919 0.0803 0.0661 0.0203
## Cumulative Proportion  0.278 0.480 0.617 0.723 0.8151 0.8954 0.9616 0.9819
##                            PC9    PC10
## Standard deviation     0.3545 0.23519
## Proportion of Variance 0.0126 0.00553
## Cumulative Proportion  0.9945 1.00000
```

The first 5 principal components account for 81.51% of the variability in the dataset.

2.6 Variable importance

We will use the earth package to determine the important variables.

```
##                                   nsubsets   gcv    rss
## Direct_Bilirubin                         5 100.0  100.0
## Alamine_Aminotransferase                 4  52.2   60.2
## Age                                      3  33.3   44.1
## Alkaline_Phosphotase                     2  23.5   33.8
## Gender-unused                            0   0.0    0.0
## Total_Bilirubin-unused                   0   0.0    0.0
## Aspartate_Aminotransferase-unused        0   0.0    0.0
## Total_Protiens-unused                    0   0.0    0.0
## Albumin-unused                           0   0.0    0.0
## Albumin_and_Globulin_Ratio-unused        0   0.0    0.0
```

The above results show that only the variables Direct_Bilirubin, Alamine_Aminotransferase, Age, and Alkaline_Phosphotase are important.

2.7 Multi-collinearity (Variance Inflation Factor)

We will use the car package to check for multi-collinearity and determine redundant variables.

```
##                          Age                       Gender
##                         1.09                         1.03
##             Total_Bilirubin           Direct_Bilirubin
##                         3.05                         3.29
##       Alkaline_Phosphotase   Alamine_Aminotransferase
##                         1.14                         2.06
## Aspartate_Aminotransferase             Total_Protiens
##                         2.09                         6.16
##                      Albumin Albumin_and_Globulin_Ratio
##                        11.71                         4.21
```

Typically in practice there is a small amount of collinearity among the predictors. As a rule of thumb, a variance inflation factor (VIF) value that exceeds 5 or 10 indicates a problematic amount of collinearity. (Introduction to Statistical Learning 7ed p.101)

Hence, Albumin and Total_Protiens appear to be multi-collinear or redundant as they have VIF values of 11.71 and 6.16, respectively.

2.8 Check for normality

This is to investigate whether the observed sample is from a normal distribution. It is used for assessing whether the sample data are randomly obtained from a normally distributed population. It does not require that the mean or variance of the hypothesized normal distribution be specified in advance.

We will run shapiro test to generate & report p-values.

a.) Null hypothesis H0: Is the sample from a normal distribution?
b.) if p-value >= 0.05 we do not reject the null hypothesis that the data are from normal distribution
c.) if p-value < 0.05 we reject the null hypothesis that the data are from normal distribution

```
##                                     p_values
## Age.p.value                         3.34e-03
## Total_Bilirubin.p.value             2.21e-38
## Direct_Bilirubin.p.value            1.64e-36
## Alkaline_Phosphotase.p.value        6.99e-35
## Alamine_Aminotransferase.p.value    1.91e-41
## Aspartate_Aminotransferase.p.value 2.01e-42
## Total_Protiens.p.value              2.88e-03
## Albumin.p.value                     5.34e-03
## Albumin_and_Globulin_Ratio.p.value 1.31e-13
```

Since the p-values for all the 9 numeric variables are less than 0.05, we reject the null hypothesis that the data are from normal distribution.

2.9 Check for linearity

This is to investigate whether the observed sample is linear, and that the null hypothesis is that the regression model is linear. This test attempts to detect non-linearities when the data is ordered with respect to a specific variable.

We will run Harvey-Collier test for linearity to generate & report p-values.

a.) Null hypothesis H0: Is the regression model correctly specified as linear?
b.) if p-value >= 0.05 we do not reject the null hypothesis of linearity
c.) if p-value < 0.05 we reject the null hypothesis of linearity

```
##
##  Harvey-Collier test
##
## data:  f
## HC = 0.55589326, df = 568, p-value = 0.5785027
```

Since p-value = 0.5785027, which is >= 0.05, we do not reject the null hypothesis of linearity. And since we are not rejecting the null hypothesis of linearity, this likewise means that we could use generalized linear models like logistic regression.

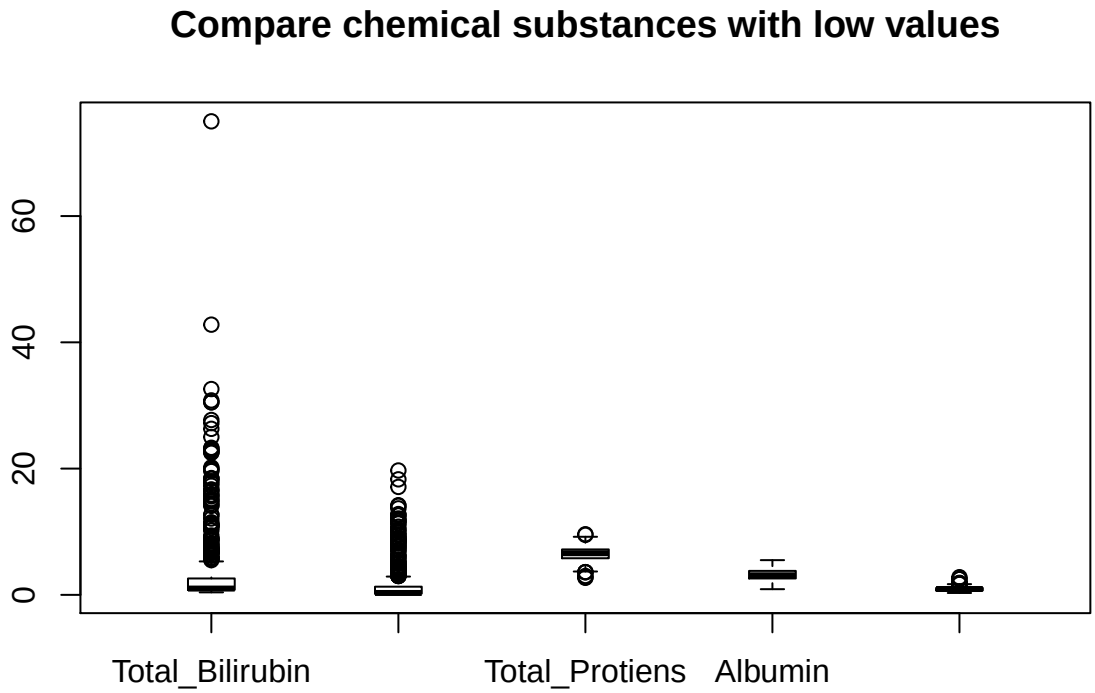
2.10 Check for outliers (chi-squared test)

This is investigate whether the sample data contain outliers. The function chisq.out.test can be used to perform this test and takes the form chisq.out.test(data, variance=1). The parameter variance refers to the known population variance.

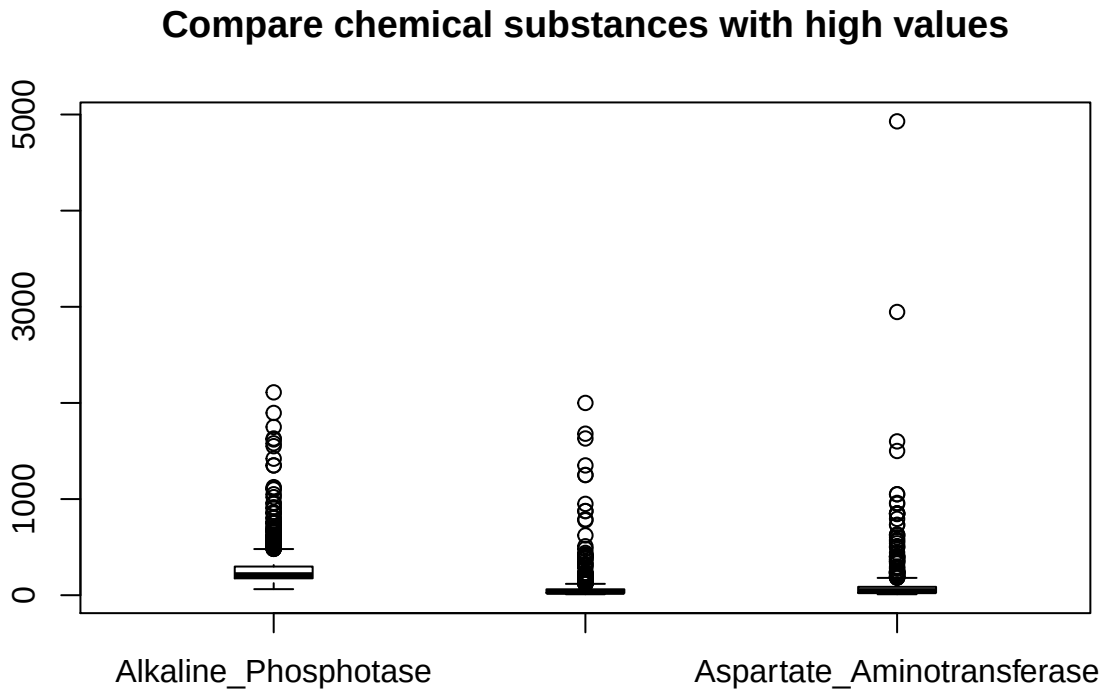
```
##
##  chi-squared test for outlier
##
## data:  reg_residual
## X-squared.116 = 6.5277471, p-value = 0.01062044
## alternative hypothesis: highest value 2.556298918594 is an outlier
```

```
## [1] "The highest value 2.556298918594 is an outlier with a p-value of 0.0106204388918135"
```

Comparative boxplot of chemical substances with low values, namely: Total_Bilirubin, Direct_Bilirubin, Total_Protiens, Albumin, and Albumin_and_Globulin_Ratio



Comparative boxplot of chemical substances with high values, namely: Alkaline_Phosphotase, Alamine_Aminotransferase, and Aspartate_Aminotransferase



3. Methods

The modeling approach for this project is implemented by performing the following 4 general steps:

1.) Splitting the clean liver dataset 70%-30% into train_set and test_set
2.) Choosing and fitting the model on the train_set
3.) Making predictions on the test_set and measuring performance
4.) Selecting the model with the highest specificity

Splitting into train and test datasets

The distribution of liver disease for both the train_set and test_set are as follows:

For the train_set dataset:

```
## disease
##   N   Y
## 289 115
```

```
##         disease
## gender     N   Y
##   Female  61  39
##   Male   228  76
```

For the test_set dataset:

```
## disease
##   N   Y
## 125  50
```

```
##         disease
## gender    N  Y
##   Female 30 10
##   Male   95 40
```

Fitting/training the model

After creating the train_set and test_set datasets, the following models will be built and trained using the train_set:

a.) logistic regression
b.) neural network/deep learning
c.) random forests
d.) gradient boosting machine (GBM)
e.) support vector machine (SVM)
f.) naive bayes
g.) classification tree
h.) C5.0 classification tree
i.) evolutionary classification tree
j.) logistic model-based recursive partitioning
k.) principal components analysis (PCA)
l.) principal components analysis - singular value decomposition (PCA-SVD) method

The first six models from logistic regression to naive bayes indicated above will make use of the h2o package. The h2o library is a scalable open-source machine learning library and as likewise previously mentioned in my Capstone Movielens Project Report, there are benefits of using the h2o package. According to this article from R-bloggers (https://www.r-bloggers.com/5-reasons-to-learn-h2o-for-high-performance-machine-learning/), there are 5 reasons for using h2o:

a.) h2o AutoML automates the machine learning workflow, which includes automatic training and tuning of many models.
b.) Scalable on Local Compute: distributed, in-memory processing speeds up computations
c.) Spark integration & GPU support: the result is 100x faster training than traditional ML
d.) Superior performance: best algorithms, optimized and ensembled: The most popular algorithms are incorporated including GLM, random forest, GBM and more.
e.) Production ready, e.g. docker containers

Making predictions on the test_set and measuring performance

Note that there are 50 out of 175 cases of liver disease in the test_set. This number will be monitored in the confusion matrix as the models are generated. In terms of measuring performance, accuracy, sensitivity, specificity, and precision will be calculated. And all throughout this section, the confusion matrix will be indicated, along with these model performance measures. Importantly, particular emphasis will be on specificity or the true negative rate, because we are interested in the number of true negatives (TN), i.e. the presence of liver disease, being predicted accurately by the models.

The formulas in measuring performance of the models used in this project is found in Section 27.4.4 of the book: https://rafalab.github.io/dsbook/introduction-to-machine-learning.html, namely:

a.) Accuracy = (TruePositives + TrueNegatives)/SampleSize

This is the raw prediction accuracy of the model

b.) Sensitivity = TruePositives/(TruePositives + FalseNegatives)

Sensitivity (or Recall) is the True Positive Rate (TPR) or the proportion of identified positives among the liver disease-positive population (class = 1)

c.) Specificity = TrueNegatives/(TrueNegatives + FalsePositives)

Specificity, which measures the True Negative Rate (TNR), is the proportion of identified negatives among the liver disease-negative population (class = 0)

d.) Precision = TruePositivesP/(TruePositives + FalsePositives)

Precision is the proportion of true positives among all the individuals that have been predicted to have liver disease-positive by the model. This represents the accuracy of a predicted positive outcome.

3.1 Logistic regression (LR) model h2o

```
## [1] "Logistic regression confusion matrix"
```

```
##            Reference
## Prediction  N  Y
##          N 84 17
##          Y 41 33
```

method	accuracy	sensitivity	specificity	precision
h2o logistic regression	0.6685714286	0.672	0.66	0.8316831683

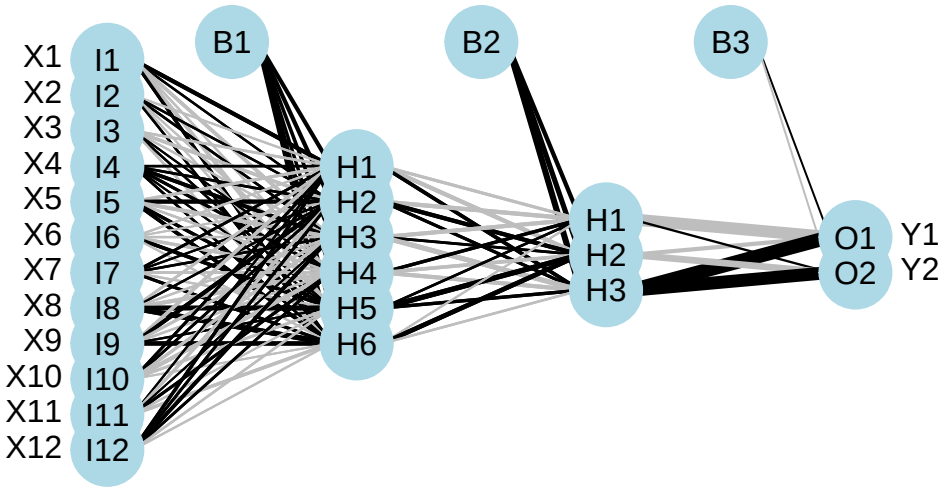
3.2 Neural network (deep learning) model h2o

```
## [1] "Neural network confusion matrix"
```

```
##            Reference
## Prediction  N  Y
##          N 80 17
##          Y 45 33
```

method	accuracy	sensitivity	specificity	precision
h2o logistic regression	0.6685714286	0.672	0.66	0.8316831683
h2o neural network/deep learning (6,3) hidden layers	0.6457142857	0.640	0.66	0.8247422680

Sample representation of the neural network model, with (6, 3) hidden layers, without the actual values of the weights and biases, is shown below. X1..X12 = dataset variables; H1..H6 and H1..H3 = the hidden layers; and Y1/Y2 represent the outputs. B1..B3 are the biases, while the lines represent the synapses/weights.



3.3 Random forests (rf) model h2o

```
## [1] "Random forests confusion matrix"
```

```
##           Reference
## Prediction  N  Y
##          N 67 11
##          Y 58 39
```

method	accuracy	sensitivity	specificity	precision
h2o logistic regression	0.6685714286	0.672	0.66	0.8316831683
h2o neural network/deep learning (6,3) hidden layers	0.6457142857	0.640	0.66	0.8247422680
h2o random forests model	0.6057142857	0.536	0.78	0.8589743590

3.4 Gradient boosting machine (gbm) model h2o

```
## [1] "Gradient boosting machine confusion matrix"
```

```
##           Reference
## Prediction  N  Y
##          N 99 29
##          Y 26 21
```

method	accuracy	sensitivity	specificity	precision
h2o logistic regression	0.6685714286	0.672	0.66	0.8316831683
h2o neural network/deep learning (6,3) hidden layers	0.6457142857	0.640	0.66	0.8247422680
h2o random forests model	0.6057142857	0.536	0.78	0.8589743590
h2o gradient boosting machine (gbm) model	0.6857142857	0.792	0.42	0.7734375000

3.5 Support vector machine (SVM) model h2o

```
## [1] "SVM confusion matrix"
```

```
##           Reference
## Prediction   N   Y
##          N 120  37
##          Y   5  13
```

method	accuracy	sensitivity	specificity	precision
h2o logistic regression	0.6685714286	0.672	0.66	0.8316831683
h2o neural network/deep learning (6,3) hidden layers	0.6457142857	0.640	0.66	0.8247422680
h2o random forests model	0.6057142857	0.536	0.78	0.8589743590
h2o gradient boosting machine (gbm) model	0.6857142857	0.792	0.42	0.7734375000
h2o support vector machine (svm) model	0.7600000000	0.960	0.26	0.7643312102

3.6 Naive bayes (NB) model h2o

```
## [1] "Naive bayes confusion matrix"
```

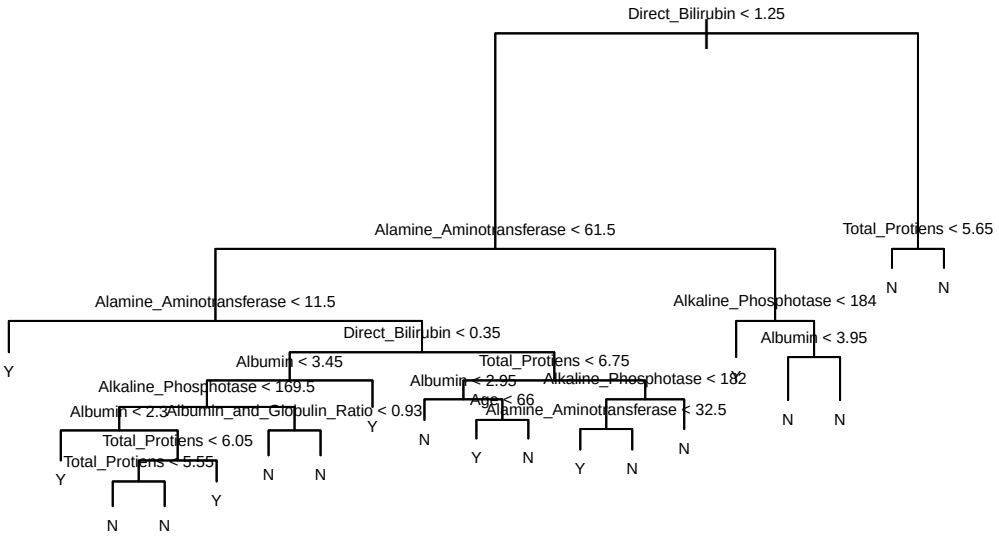
```
##           Reference
## Prediction  N  Y
##          N 60  6
##          Y 65 44
```

method	accuracy	sensitivity	specificity	precision
h2o logistic regression	0.6685714286	0.672	0.66	0.8316831683
h2o neural network/deep learning (6,3) hidden layers	0.6457142857	0.640	0.66	0.8247422680
h2o random forests model	0.6057142857	0.536	0.78	0.8589743590
h2o gradient boosting machine (gbm) model	0.6857142857	0.792	0.42	0.7734375000
h2o support vector machine (svm) model	0.7600000000	0.960	0.26	0.7643312102
h2o naive bayes model	0.5942857143	0.480	0.88	0.9090909091

3.7 Decision tree - classification tree

```
## [1] "Decision tree confusion matrix"
```

```
##           Reference
## Prediction  N  Y
##          N 88 26
##          Y 37 24
```



method	accuracy	sensitivity	specificity	precision
h2o logistic regression	0.6685714286	0.672	0.66	0.8316831683
h2o neural network/deep learning (6,3) hidden layers	0.6457142857	0.640	0.66	0.8247422680
h2o random forests model	0.6057142857	0.536	0.78	0.8589743590
h2o gradient boosting machine (gbm) model	0.6857142857	0.792	0.42	0.7734375000
h2o support vector machine (svm) model	0.7600000000	0.960	0.26	0.7643312102
h2o naive bayes model	0.5942857143	0.480	0.88	0.9090909091
classification tree	0.6400000000	0.704	0.48	0.7719298246

3.8 C5.0 Classification tree

```
## [1] "C5.0 Classification tree important variables:"

##                            Overall
## Aspartate_Aminotransferase  100.00
## Total_Bilirubin              84.41
## Gender                       58.91
## Albumin                      40.10
## Age                          34.90
## Alkaline_Phosphotase         16.58
## Alamine_Aminotransferase     15.84
## Direct_Bilirubin              0.00
## Total_Protiens                0.00
## Albumin_and_Globulin_Ratio    0.00

## [1] "C5.0 Classification tree confusion matrix"
```

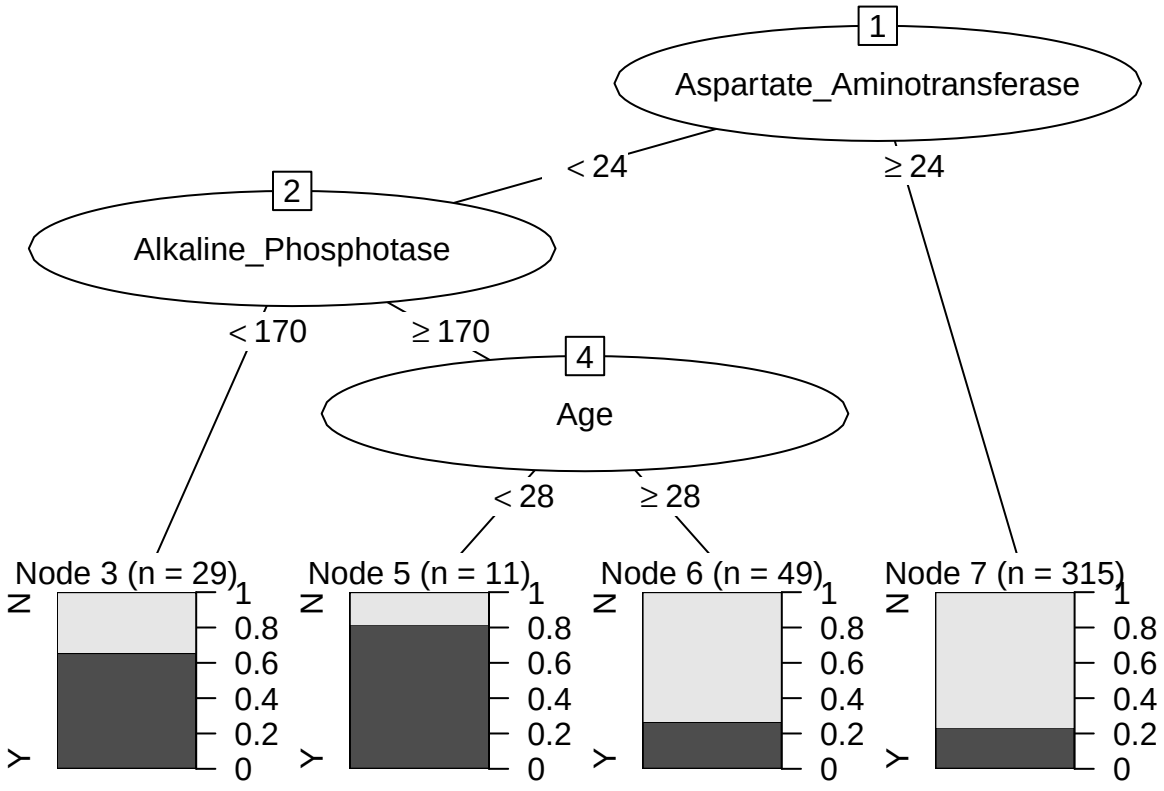
```
##           Reference
## Prediction   N   Y
##          N 113  40
##          Y  12  10
```

method	accuracy	sensitivity	specificity	precision
h2o logistic regression	0.6685714286	0.672	0.66	0.8316831683
h2o neural network/deep learning (6,3) hidden layers	0.6457142857	0.640	0.66	0.8247422680
h2o random forests model	0.6057142857	0.536	0.78	0.8589743590
h2o gradient boosting machine (gbm) model	0.6857142857	0.792	0.42	0.7734375000
h2o support vector machine (svm) model	0.7600000000	0.960	0.26	0.7643312102
h2o naive bayes model	0.5942857143	0.480	0.88	0.9090909091
classification tree	0.6400000000	0.704	0.48	0.7719298246
C5.0 classification tree	0.7028571429	0.904	0.20	0.7385620915

3.9 Evolutionary classification tree

```
## [1] "Evolutionary classification tree confusion matrix"
```

```
##           Reference
## Prediction   N   Y
##          N 115  43
##          Y  10   7
```



method	accuracy	sensitivity	specificity	precision
h2o logistic regression	0.6685714286	0.672	0.66	0.8316831683
h2o neural network/deep learning (6,3) hidden layers	0.6457142857	0.640	0.66	0.8247422680
h2o random forests model	0.6057142857	0.536	0.78	0.8589743590
h2o gradient boosting machine (gbm) model	0.6857142857	0.792	0.42	0.7734375000
h2o support vector machine (svm) model	0.7600000000	0.960	0.26	0.7643312102
h2o naive bayes model	0.5942857143	0.480	0.88	0.9090909091
classification tree	0.6400000000	0.704	0.48	0.7719298246
C5.0 classification tree	0.7028571429	0.904	0.20	0.7385620915
Evolutionary classification tree	0.6971428571	0.920	0.14	0.7278481013

3.10 Logistic model-based recursive partitioning

```
## [1] "Logistic model-based recursive partitioning confusion matrix"

##           Reference
## Prediction   N   Y
##          N 114  43
##          Y  11   7
```

method	accuracy	sensitivity	specificity	precision
h2o logistic regression	0.6685714286	0.672	0.66	0.8316831683
h2o neural network/deep learning (6,3) hidden layers	0.6457142857	0.640	0.66	0.8247422680
h2o random forests model	0.6057142857	0.536	0.78	0.8589743590
h2o gradient boosting machine (gbm) model	0.6857142857	0.792	0.42	0.7734375000
h2o support vector machine (svm) model	0.7600000000	0.960	0.26	0.7643312102
h2o naive bayes model	0.5942857143	0.480	0.88	0.9090909091
classification tree	0.6400000000	0.704	0.48	0.7719298246
C5.0 classification tree	0.7028571429	0.904	0.20	0.7385620915
Evolutionary classification tree	0.6971428571	0.920	0.14	0.7278481013
Logistic model based recursive partitioning	0.6914285714	0.912	0.14	0.7261146497

3.11 Principal components analysis (PCA)

```
## [1] "PCA confusion matrix"

##           Reference
## Prediction  N  Y
##          N 98 34
##          Y 27 16
```

method	accuracy	sensitivity	specificity	precision
h2o logistic regression	0.6685714286	0.672	0.66	0.8316831683
h2o neural network/deep learning (6,3) hidden layers	0.6457142857	0.640	0.66	0.8247422680
h2o random forests model	0.6057142857	0.536	0.78	0.8589743590
h2o gradient boosting machine (gbm) model	0.6857142857	0.792	0.42	0.7734375000
h2o support vector machine (svm) model	0.7600000000	0.960	0.26	0.7643312102
h2o naive bayes model	0.5942857143	0.480	0.88	0.9090909091
classification tree	0.6400000000	0.704	0.48	0.7719298246
C5.0 classification tree	0.7028571429	0.904	0.20	0.7385620915
Evolutionary classification tree	0.6971428571	0.920	0.14	0.7278481013
Logistic model based recursive partitioning	0.6914285714	0.912	0.14	0.7261146497
Principal components analysis (PCA)	0.6514285714	0.784	0.32	0.7424242424

3.12 Principal components analysis - singular value decomposition method (PCA-SVD)

```
##            Reference
## Prediction   N   Y
##          N 103  41
##          Y  22   9
```

method	accuracy	sensitivity	specificity	precision
h2o logistic regression	0.6685714286	0.672	0.66	0.8316831683
h2o neural network/deep learning (6,3) hidden layers	0.6457142857	0.640	0.66	0.8247422680
h2o random forests model	0.6057142857	0.536	0.78	0.8589743590
h2o gradient boosting machine (gbm) model	0.6857142857	0.792	0.42	0.7734375000
h2o support vector machine (svm) model	0.7600000000	0.960	0.26	0.7643312102
h2o naive bayes model	0.5942857143	0.480	0.88	0.9090909091
classification tree	0.6400000000	0.704	0.48	0.7719298246
C5.0 classification tree	0.7028571429	0.904	0.20	0.7385620915
Evolutionary classification tree	0.6971428571	0.920	0.14	0.7278481013
Logistic model based recursive partitioning	0.6914285714	0.912	0.14	0.7261146497
Principal components analysis (PCA)	0.6514285714	0.784	0.32	0.7424242424
PCA - singular value decomposition	0.6400000000	0.824	0.18	0.7152777778

4. Results

Herewith is a summary of the results and brief discussion of the performance of the models:

method	accuracy	sensitivity	specificity	precision
h2o logistic regression	0.6685714286	0.672	0.66	0.8316831683
h2o neural network/deep learning (6,3) hidden layers	0.6457142857	0.640	0.66	0.8247422680
h2o random forests model	0.6057142857	0.536	0.78	0.8589743590
h2o gradient boosting machine (gbm) model	0.6857142857	0.792	0.42	0.7734375000
h2o support vector machine (svm) model	0.7600000000	0.960	0.26	0.7643312102
h2o naive bayes model	0.5942857143	0.480	0.88	0.9090909091
classification tree	0.6400000000	0.704	0.48	0.7719298246
C5.0 classification tree	0.7028571429	0.904	0.20	0.7385620915
Evolutionary classification tree	0.6971428571	0.920	0.14	0.7278481013
Logistic model based recursive partitioning	0.6914285714	0.912	0.14	0.7261146497
Principal components analysis (PCA)	0.6514285714	0.784	0.32	0.7424242424
PCA - singular value decomposition	0.6400000000	0.824	0.18	0.7152777778

h2o models

For the first six h2o models, it is evident that naive bayes model yielded the highest specificity, or true negative rate (TNR), of 0.88. This means that out of the true 50 liver disease cases in the test_set, naive bayes model will accurately predict 44 of those as having liver disease. Unfortunately, it is not very good at predicting non-liver disease due to the model's low sensitivity value of 0.48, likewise known as the true positive rate (TPR). But then as an old addage, it is always better to err on the side of caution and predict the true non-liver disease patients as having the disease so that further examinations may be conducted; rather than to predict the true liver-disease patients as not having the disease. At any rate, naive baye's raw prediction accuracy is at 0.59.

Alternatively, random forests model performed fairly well next to naive bayes which reports specificity of 0.78. This model has sensitivity of 0.536, with about 0.61 raw prediction accuracy. This means that out of the true 50 liver disease cases in the test_set, random forests model will predict about 39 of those as having liver disease.

Classification trees and PCAs

The remaining six models, i.e. classification trees and principal components analysis (PCA), do not seem to perform quite well in this type of dataset. It appears that the highest specificity that could be achieved is 0.48, and this belongs to the classification tree model. This means that out of the true 50 liver disease cases in the test_set, the classification tree model will only predict 24 of those as having liver disease. This value is even less than 50% and may not be acceptable in practice. Nevertheless, the decision trees are good at predicting non-liver disease due to sensitivities in the order of 0.90 and raw accuracies of about 0.70.

5. Conclusion

Based of the foregoing analysis and results, it is evident that the naive bayes model could be used to accurately predict liver disease patients due to its high specificity value of 0.88. This model was determined to be the best algorithm for this type of dataset, with a modest raw prediction accuracy of 0.59.

One of the limitations in predicting liver disease given the variables in this dataset is in knowing the roles and contributions of the various chemical substances and how they interact together. For instance, although the variables Direct_Bilirubin, Alamine_Aminotransferase, Age, and Alkaline_Phosphotase were deemed important as indicated in Section 2.6, it felt inappropriate to simply remove all the other remaining variables and just focus on these four predictors during the modeling process. Perhaps future work might be to further investigate using classification trees or implement regularization algorithms for the liver disease dataset similar to the movielens dataset indicated in Section 33.9 of the Data Science book (https://rafalab.github.io/dsbook/large-datasets.html). This is to understand how the various chemical substances contribute to the presence and absence of liver disease. As well, if time permits, perhaps a brief consultation with a medical expert to understand the variables and validate the modeling results would be very helpful.

Nevertheless, in conclusion, the naive bayes model was found to be the best algorithm in predicting liver disease based of available variables in this dataset.