Developing predictive and explanatory models for diagnosing liver disease in patients.

G.Corke
School of Computing, Engineering and Mathematics, Western Sydney University, Parramatta, Australia
gavan.corke@gmail.com

This paper builds a logistic regression model and decision tree to predict liver disease in patients. The dataset was obtained from the UCI machine learning repository. The dataset was cleaned and transformed into an 80/20 training test split. A correlation matrix was used to determine if there was collinearity among variables and then particular variables were selected to be a part of the candidate models. The overall accuracy of the final logistic regression model was 71%. The final accuracy of the classification tree was 68%. To improve future model building on the same dataset or on similar future datasets, principal component analysis could be used as a way to do feature selection. Using L1 or L2 regularization is also suggested to reduce variance in the model and improve overall generalization of the model. Taking an algorithmic approach to selecting the decision threshold for logistic regression such as finding the optimal threshold using a ROC or precision-recall curve can be used for future analysis. Implementing an ensemble technique like random forests is also suggested instead of a classification tree to improve overall accuracy of the models.

I. Introduction

Liver disease is one of the most prevalent medical conditions in the world. Approximately, 2 million people per year die worldwide from liver disease.[1] There has been multiple factors and reasons for an increase in liver disease: inadequate diets, excessive consumption of alcohol and intake of drugs.[2] As of 2019, 2 billion people consume alcohol and roughly 75 million have alcohol-related disorders and are thus at risk of developing liver disease associated with alcohol consumption.[3] Therefore it is pertinent that diagnosis and treatment is swift and as efficient as possible. Automating the process of diagnosing liver disease and medical issues in general can have an immense impact in the developing world, where adequate medical resources can be sparse. Even in the developed world where there is sufficient medical infrastructure, having a predictive algorithm that can effectively predict liver disease reduces the burden on doctors and enables them to make more informed decisions.[4]

The purpose of this report is twofold: to analyse and explore the dataset provided by the UCI machine learning repository on Indian Liver Disease patients records collected from North East of Andhra Pradesh, India; and then build two machine learning models: one logistic regression and one decision tree model. [5] The logistic regression model will be selected on the basis of its predictive ability in being able to diagnose liver disease or not, and also on its ability to provide a causal relationship between the predictor variables and the target.[6] The decision tree will also be assessed on its predictive capabilities.

The next section will discuss the nature of the data set and the variables contained in it. Then, a brief analysis of the predictor variables and their importance in understanding liver disease will be articulated. In the third section, the data pre-processing and exploration will be done. In the fourth section, the building of the logistic regression and decision tree model and results. In the last section I will provide recommendations for improvements to the report methodology in the future and a conclusion section.

II. The dataset

The dataset obtained by the UCI ML repository as stated, was Liver Disease Patient records collected from Pradesh, India. The dataset contained 416 positive cases (those who had liver disease) and 167 negative cases (those who didn't have liver disease). The data contains 142 female and 441 male patient records. Patients who were older than 89 were all assigned the age of 90.[7]

The variables in the dataset are included in table 2.1[8]

Table 2.1 – Variations in the UCI ML Dataset

Variable(s)	Type	Unit
Age	int	Years
Gender	factor	Sex
Total_Bilirubin	num	mg/dL
Direct_Bilirubin	num	mg/dL
Alkaline_Phosphotase	int	IU/L

[1] H Asrani et al. "Burden of liver diseases in the world." *in Journal of Hepatology*, 70, 2018, p 151.

[2] B Ramana, et al. *A Critical Study of Selected Classification Algorithms for Liver Disease Diagnosis* in International Journal of Database Management Systems, *3(2)*, (2011), *101–114.*

[3] Asrani et al. p 1

[4] Fridman, L, *Jeremy Howard: fast.ai Deep Learning Courses and Research Artificial Intelligence (AI)* Podcast, [Online Video], 2019, https://www.youtube.com/watch?v=J6XcP4JOHmk , (accessed 10 September, 2019).

[5] Dua, D. and Graff, C. (2019). *UCI Machine Learning Repository* [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science.

[6] See Breiman, L. *Statistical Modeling: The Two Cultures (with comments and a rejoinder by the author*). Statistical Science, 16(3), (2001), 199–231.

[7] Kaggle, *Indian patient liver records*, [website], 2017, https://www.kaggle.com/uciml/indian-liver-patient-records, (accessed 9 September, 2019).

[8] Proteins is misspelt "protiens" in the dataset and will not be changed in the analysis.

Alamine_Aminotranserase	int	IU/L
Aspartate_Aminotransferase	int	IU/L
Total_Protiens	num	g/dL
Albumin	num	g/dL
Albumin_and_Globulin_Ratio	num	A/G
Dataset	int	Disease or Not.

As the table 2.1 shows, there were 11 variables included in the dataset. The "Dataset" variable was the target variable which stated whether the patient was diagnosed with liver disease as determined by experts. The other variables besides gender and age were chemical compounds found in the human body, which, given the medical literature are assumed to be related to liver disease.[9]

Bilirubin is a yellow-orange compound that is produced when the liver breaks down red blood cells.[10] There are two types of bilirubin: conjugated (direct) and unconjugated (indirect). [11] Direct bilirubin is water soluble and made from the liver using unconjugated bilirubin, whilst unconjugated is not soluble.[12] Total bilirubin is an estimation of the total sum of direct and indirect bilirubin in the body. High detected levels of bilirubin have been associated with liver disease.[13]

Alkaline phosphatase (ALP) is an enzyme found in the body. The highest concentrations can be found in cells inside the liver and the bone.[14] Elevated levels of ALP can be caused most commonly by liver disease. An ALP test is used to measure the amount of ALP in a patient's blood as elevated levels are usually a symptom of liver damage, since ALP leaks into the bloodstream from the liver when this is the case.[15]

Alamine aminotransferase and aspartate aminotransferase (ALT and AST respectively) are enzymes found in the body. Both have high concentrations in the liver and kidneys. ALT and AST are usually both measured concurrently to screen for liver damage and or monitor liver disease.[16]

Albumin is the main protein found in blood plasma.[17] Albumin is made by the liver; it is responsible for keeping fluid in the bloodstream so that it doesn't leak into other tissues.[18] Low albumin levels can be a marker for kidney or liver disease, including cirrhosis.[19]

The variable **Total_Protiens** is a measurement of the total amount of albumin and globulin in a patient's body. Globulins are another protein that play an important role in the body's immune system.[20]

The albumin and globulin ratio (A/G) is a ratio of the levels of albumin and globulin in the body. A ratio of close to 1 indicates normal health. However, if the ratio is too low, meaning that there is more globulin than albumin, this can indicate cirrhosis, which is a form of liver disease.[21]

Given the nature of the chemical compound variables, we can show they are all viable candidates to be put into a predictive model to predict liver disease, as these are all in some way or another markers of liver disease, and will form the basis for our machine learning models.

III. DATA PREPROCESSING AND EXPLORATION.

3.1 Data Cleaning

The Indian Liver Disease data set was uploaded and checked first for any NA or missing values. It was found that there were 4 NA values in the **Albumin_and_Globulin_Ratio** column. The subsequent records that included these NA values were omitted.[22]

The column name **Dataset** was renamed to **Liver_Disease** so that it would be easier to identify the target variable and identify its true nature. The target variable **Liver_Disease** column had a binary classification value of 1 and 2, where 1 was assigned to patients with liver disease and 2 without. This was modified so that the 1 indicated liver disease and 0 indicated no liver disease.

For the gender variable, the output was either male or female, given as a factor type variable, where 1 was female and 2 was male. This was also changed so male was 1 and female was 0. All the other variables in the data were numeric data types and Age and Gender were integer variables.

Before data exploration and model building, the dataset was divided into a train and test set using an 80/20 split. The reason that the data was split into train and test sets before data exploration was so that there wouldn't be any data leakage.[23] If one uses all the data, even in a descriptive analysis or exploration, these may be factored into our model building, which would affect the integrity of the trained model

[9] Kaggle (2019).

[10] University of Michigan Health, *Bilirubin* [website], 2019, https://www.uofmhealth.org/health-library/hw3474, (accessed 9 September, 2019).

[11] BC and UC for short.

[12] University of Michigan Health (2019).

[13] Murali, A.R & W Carey, *Liver Test Interpretation -Approach to the Patient with Liver Disease: A Guide to Commonly Used Liver Tests* [website], 2017, http://www.clevelandclinicmeded.com/medicalpubs/diseasemanagement/hepatology/guide-to-common-liver-tests/, (accessed 20 September, 2019).

[14] Lab Tests Online, *Alkaline Phosphatase* [website], 2019, https://labtestsonline.org/tests/alkaline-phosphatase-alp/ , (accessed 15 September, 2019)

[15] MedlinePlus, *Alkaline Phosphatase* [website], 2019, https://medlineplus.gov/lab-tests/alkaline-phosphatase/, (accessed 15 September, 2019).

[16] Orlewicz, M.S & E Vovchuck, *Alanine Aminotransferase*, [website], 2014, https://emedicine.medscape.com/article/2087247-overview, (accessed 15 September, 2019).

[17] Farrugia, A., Albumin Usage in Clinical Medicine: Tradition or Therapeutic? *Transfusion Medicine Reviews, 24(1),* (2010), *p 53*

[18] MedlinePlus, *Albumin Blood Test* [website], 2019, https://medlineplus.gov/lab-tests/albumin-blood-test/, (accessed 15 September, 2019).

[19] Medicineplus, *Albumin* (2019).

[20] Slightham, C., *Total Proteins* [website], 2016, https://www.healthline.com/health/total-protein#proteins, (accessed 11 September, 2019).

[21] Slightham, C. (2019)

[22] There are other approaches to dealing with missing values, see James, G, D Witten, T Hastie, & R Tibshirani, *An introduction to statistical learning*. in, 7th ed., New York, Springer, 2017.

[23] Brownlee, J., *Data Leakage in Machine Learning* [website], 2016, https://machinelearningmastery.com/data-leakage-machine-learning/, (accessed 10 September, 2019).

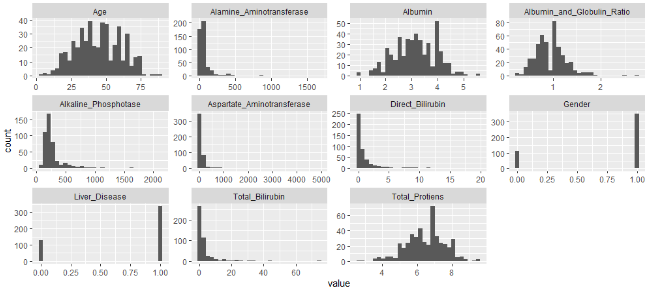
on the test set. The training data was randomly sampled from the whole dataset and the remaining 20% of the data was turned into the test dataset.

3.2 Data Exploration

This section will involve an exploratory analysis of the data. This is important for multiple reasons. One such reason is that it enables us to determine what variables are important for feature selection. It also enables us to see the relationship between the variables and the target variable. In this case, we want to see how the age, gender and the chemical compound variables relate to the diagnosis of liver disease. The analysis will be of the training dataset so as to not have data leakage.

Firstly, we can look at the summary statistics of the variables and their distributions to give us a holistic perspective on the nature of the data. The distribution of the data is given in figure 1.

Figure 1 – Distribution of Data

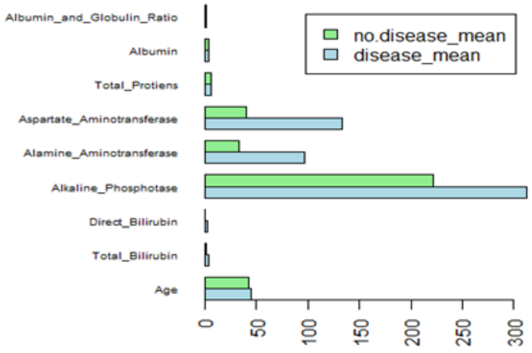


Using the **summary()** function, the summary statistics of the variables were given, shown in the appendix. From the data, the mean for the predictor variables, separated by patients who had liver disease versus those who did not, was aggregated, (see table 2). A bar plot representation of the table is shown in figure 2.

Table 2 – Mean of variables by disease and no disease

Variables	Disease (Mean)	NoDisease (Mean)
Age	45.40	41.94
Total_Bilirubin	4.03	1.17
Direct_Bilirubin	1.84	0.41
Alkaline_Phosphotase	312.99	221.98
Alamine_ Aminotransferase	96.33	33.63
Aspartate_ Aminotransferase	133.37	40.35
Total_Protiens	6.48	6.48
Albumin	3.09	3.32
Albumin_and_Globulin_Ratio	0.92	1.04

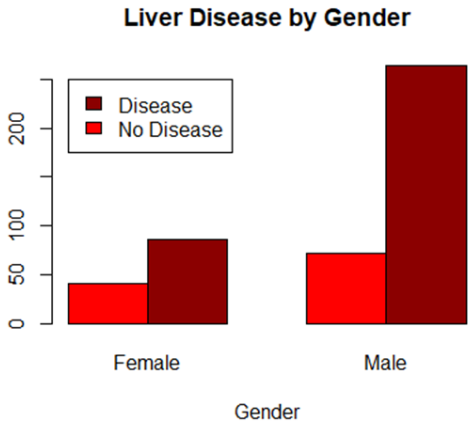
Figure 2 – Bar plot representation of means.



As we can see from the bar plot and table, asapartate aminotranserase, alamine aminotransferase, direct and total bilirubin and alkaline phosphotase are higher on average for those who have liver disease versus those who do not. There is also some small indication that the average age of those with liver disease is slightly higher than those without the disease.

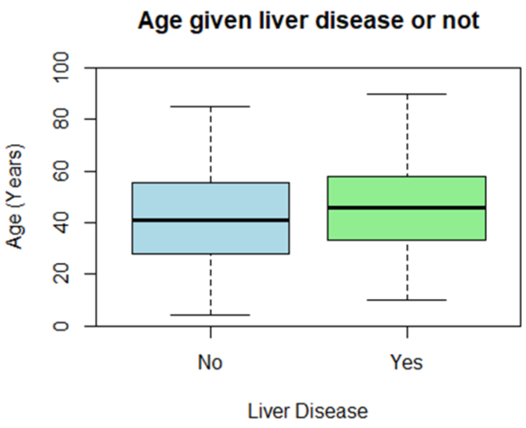
The number of patients diagnosed with liver disease in the training set is 336 and no liver disease is 127 (see appendix 3). The number of patients with and without liver disease categorized by gender is as follows: for males, 264 have liver disease whilst 86 do not, for females, 72 have liver disease 41 do not. These are shown in figures 3 and 4 respectively.

Figure 3 – Disease by gender



To see the age variable more clearly, a boxplot was plotted that shows the distribution of ages given liver disease or not.

Figure 4 – Distribution of Age by diagnosis



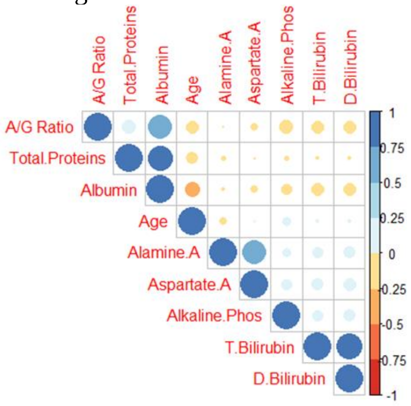
3.3 *Feature Selection*

Feature selection is an important process in building an effective model. This can be done automatically or manually. For the logistic regression model, the features will be selected manually; however, for the classification decision tree, this will not be needed, as this process is automatically taken care of. In this manual selection process, a correlation matrix will be used to determine the correlation between the individual predictors. This is so that we can reduce or remove the effects of collinearity.

Collinearity is a phenomenon that occurs when two or more predictor variables are closely related or correlated. This is problematic as it can be hard to separate how each individual variable is related to the response, and consequently, reduces the accuracy of the estimates of the regression coefficients. A correlation matrix will enable us to detect which predictors are correlated, and thus remove one of the correlated predictors from our analysis.

A correlation matrix was generated to show the relationship between the predictor variables and the correlative strength between them. The cut-off for determining whether or not there was a strong relationship between two variables was if there was a Pearson correlation score equal to or outside the interval +/- 0.7. The matrix identified the pairs of interest: direct and total bilirubin, alamine and aspartate aminotransferase, total proteins and albumin and lastly, albumin and A/G ratio.

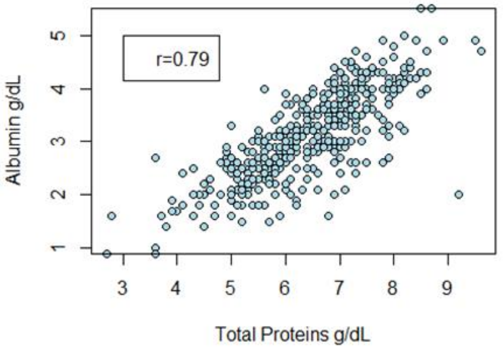
Figure 6 – Correlation Matrix

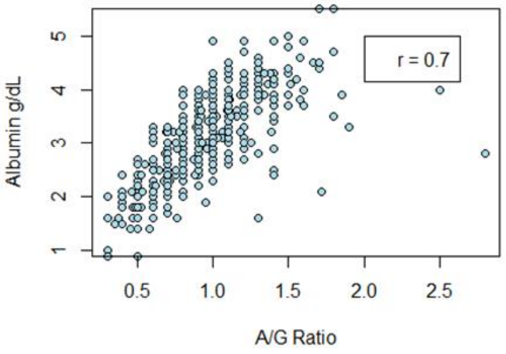


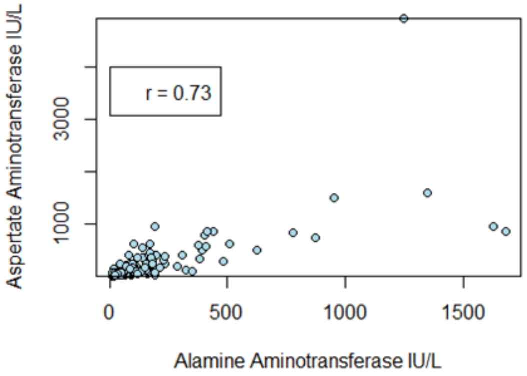
The predictors that were selected to be included into the model building process were: **Albumin_and_Globulin_Ratio Total_Protiens , Age, Alamine_Aminotransferase Alkaline_Phosphotase** and **Total_Bilirubin.** A scatter plot for the correlated variables are shown in figures 7-10.

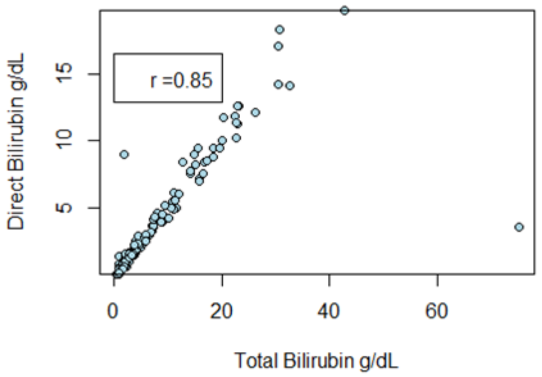
In the next section, we will build the logistic regression and decision tree models, which will be used to predict liver disease in patients.

Figures 7-10









4 MODEL BUILDING

4.1 Logistic Regression

The theoretical justification for the logistic regression model was twofold: (1) to build a model that would successfully predict whether or not a patient had liver disease, but (2) also to be interpretable. Building a ‘black box’ model where the variables were not to be understood in relation to each other would be problematic, since it would be harder to understand how the variables impact the model in any significant way.[24]

To satisfy the second condition, only coefficients of predictors that were statistically significant would be included in any model that would be evaluated via K fold cross validation. This is the operational test for us to make causative claims about the variables in the model. If the model had all statistically significant variable coefficients, then we could say that these variables directly affect whether or not a patient had liver disease with some justification.[25]

To satisfy condition (1) was more complicated. Depending on how we assess the overall accuracy of a model will change how we view the models. For example, in the case of logistic regression, one way to determine the predictive power of a model is to look at the misclassification rate. By reducing the misclassification rate, we thereby reduce the model’s error. Therefore, improving accuracy is an important measure. However, there are issues associated with this approach given the dataset. In the case of unbalanced datasets, sometimes it is easy to get a high accuracy or low misclassification rate if the true values are skewed to one class.

In the case of the liver disease dataset in question, the non-split data has 416 patients with liver disease versus 167 without. Therefore, just predicting a positive case for every value would give an accuracy of 71%. In addition, depending on what we think is important given domain knowledge of liver disease diagnosis, our performance metric will be affected. For example, is reducing the number of false positives (false diagnosis of liver disease) more important than false negatives (false diagnosis of no liver disease)?

In this report, we will ultimately test the performance of a model on minimizing misclassification rate, but with some consideration for its false positive rate and true positive rate given everything else being all equal.

4.1.2 Results

Four models were constructed which all had statistically significant coefficients (see Appendix). These can be seen in the appendix. The model that had the lowest cross validation error was the final model to be used on the test set. The cross-validation error metric was misclassification rate. A cut-off threshold of 0.6 was given instead of 0.5 since the dataset was skewed towards positive classes. The resulting model and its performance on the test set is shown below.

The model that had the lowest cross validation error was:

$$z = log_e\left(\frac{P}{1-P}\right) = \beta + \beta_1(TotalBilirubin) + \beta_2(AlamineAminotransferase) + \beta_3(Age) + \beta_4\,(TotalBilirubin * AlamineAminotransferase) \quad (1)$$

Where:

$$P(Liver\ Disease|z) = \frac{1}{(1+e^{-z})} \qquad (2)$$

The cross validation had a K-fold of 10. The misclassification rate was 0.156 averaged over all the K folds. In the test set, the misclassification rate of the final model was 0.293.

The final models’ parameters trained over the entire training data set was:

$$z = log_e\left(\frac{P}{1-P}\right) = -1.018 + 0.360\,x_1 + 0.017x_2 + 0.016x_3 - 0.001x_4 \quad (3)$$

Where:

$$P(Liver\ Disease|z) = \frac{1}{(1+e^{-1.018+0.360\,x_1+0.017x_2+0.016x_3-0.001x_4})} \; (4)$$

The final model’s performance on the test set is given in table 3 as a confusion matrix. The number of false positives predicted were 20, for a false positive rate of 0.23. The number of true positives 64 for a true positive rate of 0.82. The overall accuracy was 0.71 or 71%.

[24] To see more issues with interpretability and predictive ability of models, see Molnar, C., *Interpretable machine learning: A guide for making black box models explainable* [website], https://christophm.github.io/interpretable-ml-book/ (accessed 11 September, 2019).

[25] There are of course, problems with assuming significant P-values associated with predictors in the model entail causality, but for all intents and purposes, this will be our metric to make causal claims about the model. See James et. al. (2017).

	NoDisease (Actual)	Disease (Actual)
No Disease	18	14
Disease	20	64

Table 3 (Confusion matrix for test set)

The interpretation of the final logistic regression model can be articulated as follows:

- $Total\ Bilirubin = 0.360$: Increasing x_1 by 1 increases the log-odds by 0.360, so the odds that a patient have liver disease increases by a factor of $e^{0.360}$.
- $AlamineAminotransferase = 0.017$: increasing x_2 by 1 increases the log-odds by 0.017, so the odds that a patient has liver disease increases by a factor of $e^{0.017}$.
- $Age = 0.016$: increasing x_3 by 1 increases the log-odds by 0.016, so the odds that a patient has liver disease increases by a factor of $e^{0.016}$.
- $TotalBilirubin * AlamineAminotransferase = -0.001$: increasing x_3 by 1 decreases the log-odds by -0.016, so the odds that a patient has liver disease decreases by a factor of $e^{-0.001}$. This term is an interaction term.

4.2 Decision Trees

The second machine learning algorithm that will be used to predict liver disease in patients is decision trees. In this case, they will be classification trees, given that the response variable is a categorical variable.

The response variable **Liver_Disease** will be converted into a character variable where 1 is transformed to "Yes" and 0 transformed to "No". This will make the interpretability of the decision tree clearer when looking at the diagram after building the models.

Unlike logistic regression, features do not have to be manually removed before analysis as the algorithm itself will select the important features to use in the tree automatically. Therefore, we can use all the variables in the building of the classification trees.

4.2.2 Results

Firstly, the classification tree was built on the training data. The number of terminal nodes in the tree was 24. Using the **summary()** function we can see that the training residual mean deviance was 0.73 and the misclassification error rate was 0.2117 where 98 of the response variables were misclassified on the training data.

On the test data however, the misclassification rate was 0.345. The Accuracy was 0.655. The true positive rate was 0.87, and the false positive rate was 0.31.

	No (Actual)	Yes (Actual)
No	8	10
Yes	30	68

Table 4 (unpruned tree test set)

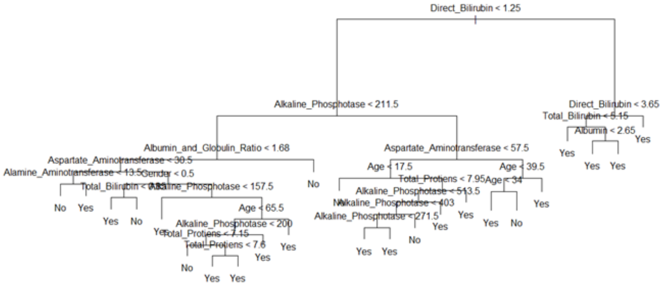

Direct_Bilirubin < 1.25
Alkaline_Phosphotase < 211.5
Direct_Bilirubin < 3.65
Total_Bilirubin < 5.15
Albumin < 2.65
Albumin_and_Globulin_Ratio < 1.68
Aspartate_Aminotransferase < 57.5
Aspartate_Aminotransferase < 30.5
Alamine_Aminotransferase < 13.5
Gender < 0.5
Total_Bilirubin < 0.85
Alkaline_Phosphotase < 157.5
Age < 65.5
Alkaline_Phosphotase < 200
Total_Protiens < 7.15
Total_Protiens < 7.6
Age < 17.5
Age < 39.5
Total_Protiens < 7.95
Age < 34
Alkaline_Phosphotase < 513.5
Alkaline_Phosphotase < 403
Alkaline_Phosphotase < 271.5
No
Yes

Figure 11 (Unpruned classification tree)

As figure 11 shows, the best split occurs using the **Direct_Bilirubin** variable to divide patients who have liver disease versus those who do not. That is why the **Direct_Bilirubin** is the root node in the tree.

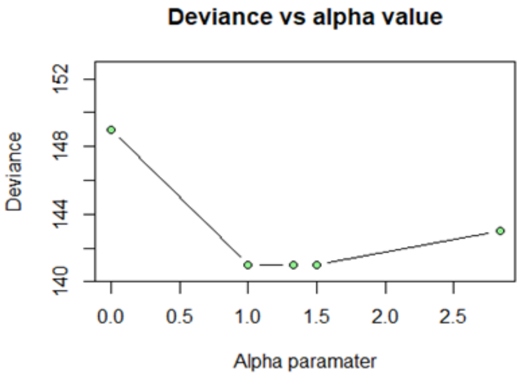
The next step in the decision tree building process is to use a method called 'pruning'. This is to stop the current classification tree from overfitting the data and not being able to adequately perform well on the test set data. In this pruning process, the ideal goal is to 'select a subtree that leads to the lowest test error rate'.[26] This is done by finding all the subtrees as a function of α. For every α there is a given subtree that minimizes the cost function. The α chosen for the final model is whichever one minimizes the lowest average error in cross validation.[27]

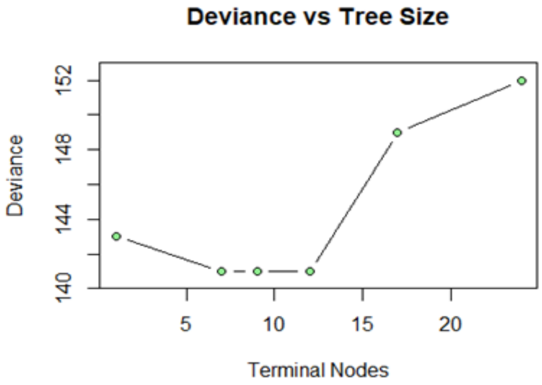
K fold cross validation was run to prune the tree using the **cv.tree()** function. The parameter FUN was given the value **prune.misclass()**, so that the misclassification rate is the error that we are trying to minimize in our cross validation and pruning process.

[26] James, p 308. Of course in practice this would be burdensome and produce too many subtrees.

[27] James, p 308.

Figure 12 and 13

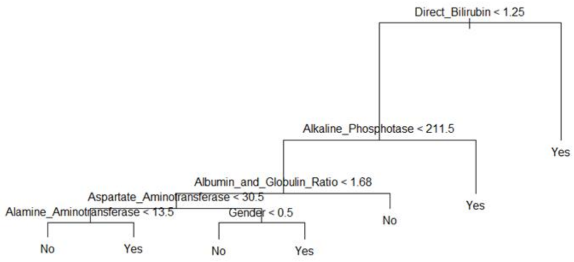




As shown in figures 12 and 13, the lowest deviance was 141, given by α = 1.5 and resulted in the best subtree with 7 terminal nodes. The final pruned tree is shown in figure 14. The final variables used in the construction of the pruned tree were:

- **Albumin_and_Globulin_Ratio,**
- **Gender,**
- **Direct_Bilirubin,**
- **Alamine_Aminotransferase,**
- **Alkaline_Phosphotase**
- **Alamine_Aminotransferase.**

Figure 14 – (Pruned Tree)


Direct_Bilirubin < 1.25
Alkaline_Phosphotase < 211.5
Yes
Albumin_and_Globulin_Ratio < 1.68
Yes
Aspartate_Aminotransferase < 30.5
Alamine_Aminotransferase < 13.5
Gender < 0.5
No
No
Yes
No
Yes

The performance of the pruned tree on the test set can be evaluated from the confusion matrix shown as table 5.

	No (Actual)	Yes (Actual)
No	3	2
Yes	35	76

Table 5 – (Unpruned tree test results)

The misclassification rate for the pruned tree was 0.32 and accuracy 0.68. The true positive rate 0.97 and the false positive rate 0.31. Therefore, we can see that the pruned tree performed better or equal on these three important metrics. It had a lower misclassification rate (and thus higher accuracy). It also had a higher true positive rate, and the false positive rate was the same for both the pruned and unpruned tree.

5 RECOMMENDATIONS

While this report tried to be as comprehensive as possible given the scope of the task, there are various ways in which this report could have been improved upon and as such recommendations will be given for future analysis.

One such way is in variable selection. Instead of the methodology employed in this report, another approach would have been to use principal component analysis. This involves reducing several possibly correlated variables into a smaller set of uncorrelated variables. This would be an alternative approach to the correlation matrix approach used in this report for addressing collinearity.[28]

To deal with the unbalanced dataset, which can be especially problematic for classification problems, one can use resampling techniques in order to increase the frequency of the minority class or reduce the frequency of the majority class.[29]

In choosing the threshold decision boundary in the logistic regression, one can use a ROC or precision-recall curve to optimize the threshold given what parts of the confusion matrix values they want to optimize.

For the logistic regression, using L1 or L2 regularization may have improved overall performance in the test set and reduced overall variance by increasing bias.[30]

Lastly, since the classification tree performed worse than the logistic regression model overall, perhaps an ensemble method such as random forests which utilize decision trees would have improved performance in the future. Another classification method that could have been used for the task is SVM.

[28] See Sharma, A., *Principal Component Analysis in Python* [website], 2019, https://www.datacamp.com/community/tutorials/principal-component-analysis-in-python (accessed 10 September, 2019).

[29] Analytics Vidhya, *How to handle Imbalanced Classification Problems in machine learning?* [Website], 2017, https://www.analyticsvidhya.com/blog/2017/03/imbalanced-classification-problem/

[30] See James et al. p200-220

6 CONCLUSION

A logistic regression model and classification tree were built to detect and diagnose liver disease in patients using the UCI dataset. The dataset was cleaned, and an 80/20 split was used to allocate a random portion of the data as the training set and the remaining 20 percent as the test set. Data exploration was then performed for multiple reasons, including to see the nature of the relationship between the predictors and the response, and to also determine which predictors are important when doing feature selection, especially for building the logistic regression model. It was found that the pairings: direct and total bilirubin, alamine and aspartate aminotransferase, total proteins and albumin, and albumin and A/G ratio had a high correlation with each other and were thus susceptible to collinearity.

The variables :

- **Albumin_and_Globulin_Ratio**
- **Total_Protiens**
- **Age**
- **Alamine_Aminotransferase**
- **Alkaline_ Phosphotase** and
- **Total_Bilirubin**

were decided to form the basis of the logistic regression models, with the final model being shown in equation (3) and (4). The overall accuracy of the logistic regression model was 71% with a false positive rate of 0.23 and a true positive rate of 0.82 on the test set.

For the classification tree, the unpruned tree had 24 terminal nodes and performance on the test set was: 0.345 for the misclassification rate, accuracy was 0.655. The true positive rate was 0.87, and the false positive rate was 0.31.

The pruned tree after using cross validation to select the α that reduced the training error gave a pruned tree with 7 terminal nodes. The misclassification rate for the pruned tree on the test set was 0.32 and accuracy 0.68. The true positive rate 0.97 and the false positive rate 0.31. It performed better on the test set as compared to the unpruned tree.

Overall the logistic regression model had a better accuracy and false positive rate, however the pruned classification tree had a higher true positive rate.

REFERENCES

[1] Analytics Vidhya, *How to Handle Imbalanced Classification Problems in machine learning?* [Website], 2017, https://www.analyticsvidhya.com/blog/2017/03/imbalanced-classification-problem/ (accessed 8 September 2019)

[2] Asrani, SK, H Devarbhavi, J Eaton, & PS Kamath, *"Burden of liver diseases in the world."* in Journal of Hepatology, 70, (2018), 151–171, https://sci-hub.tw/https://www.ncbi.nlm.nih.gov/pubmed/30266282

[3] Breiman, L. *Statistical Modeling: The Two Cultures (with comments and a rejoinder by the author*). Statistical Science, 16(3), (2001), 199–231.

[4] Brownlee, J., *Data Leakage in Machine Learning* [website], 2016, https://machinelearningmastery.com/data-leakage-machine-learning/, (accessed 10 September, 2019).

[5] Dua, D. and Graff, C. *UCI Machine Learning Repository [website], 2019,* [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science.

[6] Fridman, L, *Jeremy Howard: fast.ai Deep Learning Courses and Research | Artificial Intelligence (AI) Podcast,* [Online Video], 2019, https://www.youtube.com/watch?v=J6XcP4JOHmk, (accessed 10 September, 2019).

[7] Farrugia, A., Albumin Usage in Clinical Medicine: Tradition or Therapeutic? *Transfusion Medicine Reviews, 24(1),* (2010), *53–63.*

[8] James, G, D Witten, T Hastie, & R Tibshirani, *An introduction to statistical learning.* in, 7th ed., New York, Springer, *2017.*

[9] Kaggle, *Indian patient liver records,* [website], 2017, https://www.kaggle.com/uciml/indian-liver-patient-records, (accessed 9 September, 2019).

[10] Lab Tests Online, *Alkaline Phosphatase* [website], 2019, https://labtestsonline.org/tests/alkaline-phosphatase-alp/ , (accessed 15 September, 2019)

[11] Molnar, C., *Interpretable machine learning: A guide for making black box models explainable* [website], https://christophm.github.io/interpretable-ml-book/ (accessed 11 September, 2019).

[12] Murali, A.R & W Carey, *Liver Test Interpretation - Approach to the Patient with Liver Disease: A Guide to Commonly Used Liver Tests* [website], 2017, http://www.clevelandclinicmeded.com/medicalpubs/diseasemanagement/hepatology/guide-to-common-liver-tests/, (accessed 20 September, 2019).

[13] MedlinePlus, *Alkaline Phosphatase* [website], 2019, https://medlineplus.gov/lab-tests/alkaline-phosphatase/, (accessed 15 September, 2019).

[14] Orlewicz, M.S & E Vovchuck, *Alanine Aminotransferase,* [website], 2014, https://emedicine.medscape.com/article/2087247-overview, (accessed 15 September, 2019).

[15] MedlinePlus, *Albumin Blood Test* [website], 2019, https://medlineplus.gov/lab-tests/albumin-blood-test/, (accessed 15 September, 2019).

[16] Sharma, A., *Principal Component Analysis in Python* [website], 2019, https://www.datacamp.com/community/tutorials/principal-component-analysis-in-python (accessed 10 September, 2019).

[17] Slightam, C., *Total Proteins* [website], 2016, https://www.healthline.com/health/total-protein#proteins, (accessed 11 September, 2019).

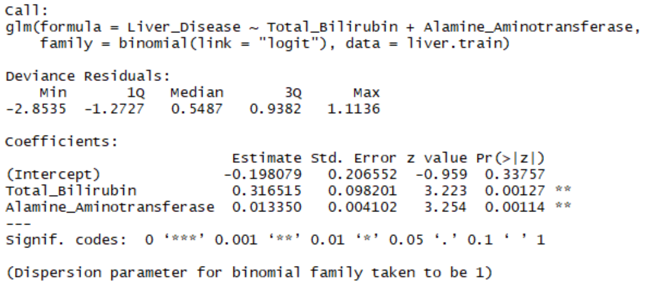
[18] Venkata Ramana, B., Babu, M. S. P., & Venkateswarlu, N. *A Critical Study of Selected Classification Algorithms for Liver Disease Diagnosis* in International Journal of Database Management Systems, *3(2),* (2011), *101–114.*

[19] University of Michigan Health, *Bilirubin* [website], 2019, https://www.uofmhealth.org/health-library/hw3474, (accessed 9 September, 2019).

APPENDIX 1

LOGISTIC REGRESSION MODELS

Model 1



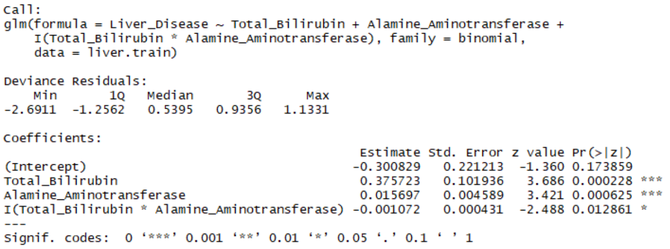
```
Call:
glm(formula = Liver_Disease ~ Total_Bilirubin + Alamine_Aminotransferase,
    family = binomial(link = "logit"), data = liver.train)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.8535  -1.2727   0.5487   0.9382   1.1136

Coefficients:
                          Estimate Std. Error z value Pr(>|z|)
(Intercept)              -0.198079   0.206552  -0.959  0.33757
Total_Bilirubin           0.316515   0.098201   3.223  0.00127 **
Alamine_Aminotransferase  0.013350   0.004102   3.254  0.00114 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)
```

Model 2

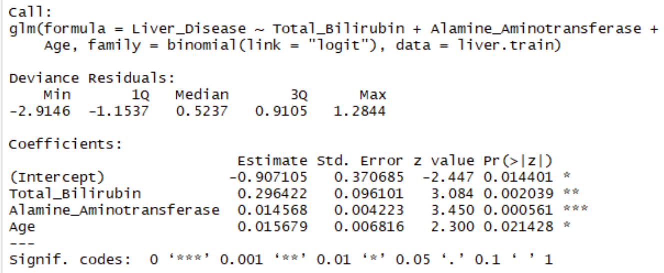


```
Call:
glm(formula = Liver_Disease ~ Total_Bilirubin + Alamine_Aminotransferase +
    I(Total_Bilirubin * Alamine_Aminotransferase), family = binomial,
    data = liver.train)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.6911  -1.2562   0.5395   0.9356   1.1331

Coefficients:
                                               Estimate Std. Error z value Pr(>|z|)
(Intercept)                                   -0.300829   0.221213  -1.360 0.173859
Total_Bilirubin                                0.375723   0.101936   3.686 0.000228 ***
Alamine_Aminotransferase                       0.015697   0.004589   3.421 0.000625 ***
I(Total_Bilirubin * Alamine_Aminotransferase) -0.001072   0.000431  -2.488 0.012861 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Model 3

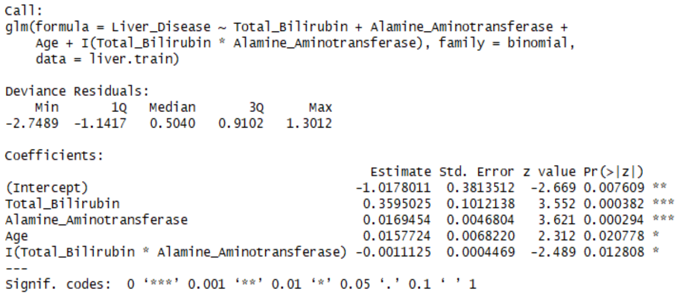


```
Call:
glm(formula = Liver_Disease ~ Total_Bilirubin + Alamine_Aminotransferase +
    Age, family = binomial(link = "logit"), data = liver.train)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.9146  -1.1537   0.5237   0.9105   1.2844

Coefficients:
                          Estimate Std. Error z value Pr(>|z|)
(Intercept)              -0.907105   0.370685  -2.447 0.014401 *
Total_Bilirubin           0.296422   0.096101   3.084 0.002039 **
Alamine_Aminotransferase  0.014568   0.004223   3.450 0.000561 ***
Age                       0.015679   0.006816   2.300 0.021428 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Model 4



```
Call:
glm(formula = Liver_Disease ~ Total_Bilirubin + Alamine_Aminotransferase +
    Age + I(Total_Bilirubin * Alamine_Aminotransferase), family = binomial,
    data = liver.train)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.7489  -1.1417   0.5040   0.9102   1.3012

Coefficients:
                                                Estimate Std. Error z value Pr(>|z|)
(Intercept)                                   -1.0178011  0.3813512  -2.669 0.007609 **
Total_Bilirubin                                0.3595025  0.1012138   3.552 0.000382 ***
Alamine_Aminotransferase                       0.0169454  0.0046804   3.621 0.000294 ***
Age                                            0.0157724  0.0068220   2.312 0.020778 *
I(Total_Bilirubin * Alamine_Aminotransferase) -0.0011125  0.0004469  -2.489 0.012808 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

CROSS VALIDATION ERRORS

A 10 fold cross validation was performed on the 4 models. The cross-validation error metric was the misclassification rate.

The cost function was defined as:

```
cost2 <- function(r ,pi =0) mean(abs(r-pi) > 0.6 )
```

Since the threshold or cut-off boundary was greater than or equal to 0.6 for the classifications of the probabilities.

The model errors are shown below:

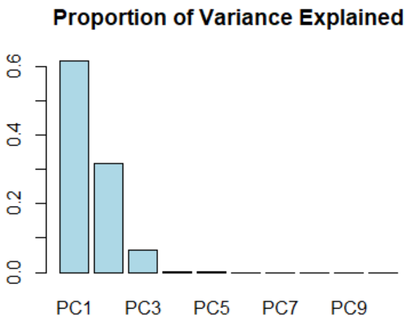
Model 1: 0.1598272
Model 2: 0.1619870
Model 3: 0.1598272
Model 4: 0.1555076

Model 4 had the lowest misclassification rate and was thus the final model chosen to be evaluated on the test set.

APPENDIX 2

PRINCIPAL COMPONENT ANALYSIS

Principal component analysis is an unsupervised learning algorithm that can be used both in variable selection for regression and exploratory data analysis. Here we briefly look at a diagram of what linear combination of variables could be transformed to create principal components that captures most of the original variables.

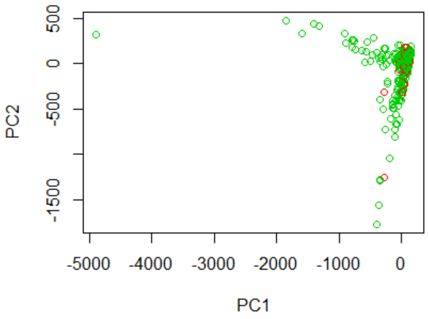


(Appendix Figure 1)

Here we can see the proportion of variance explained by the new principal components. PC1, PC2 and PC3 capture over 90% of the total variance of all the variables.

However, using PC1 and PC2 to plot the class distinctions between those who have liver disease versus those who don’t, doesn’t really capture the difference in a meaningful way as shown below.

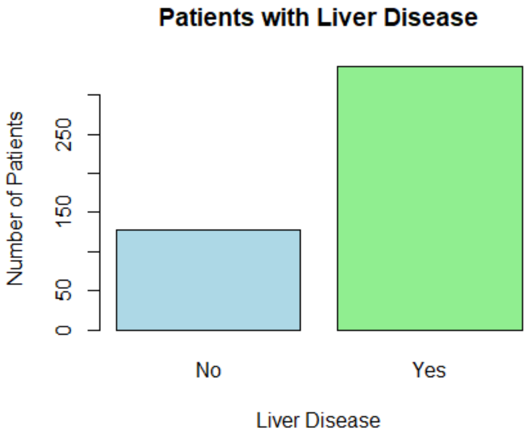
(Appendix Figure 2)



DISTRIBUTION OF LIVER DISEASE

Distribution of those with liver disease versus those without in the training data set.

(Appendix Figure 3)



APPENDIX 3

R CODE

```
library(purrr)

library(tidyr)

library(ggplot2)

library(boot)


liver.data <- read.csv("indian_liver_patient.csv")

# NAMES predictors.

# [1] "Age"                        "Gender"
"Total_Bilirubin"

# [4] "Direct_Bilirubin"
"Alkaline_Phosphotase"
"Alamine_Aminotransferase"

# [7] "Aspartate_Aminotransferase"
"Total_Protiens"             "Albumin"

# [10] "Albumin_and_Globulin_Ratio" "Dataset"

# Dataset variable is whether or not you have
liver disease, yes (1) or no (2).


#1. Age of the patient Yrs.

#2. Gender of the patient M/F

#3. TB Total Bilirubin mg/dL

#4. DB Direct Bilirubin mg/dL

#5. Alkphos Alkaline Phosphotase IU/L

#6. Sgpt Alamine Aminotransferase IU/L

#7. Sgot Aspartate Aminotransferase IU/L

#8. TP Total Protiens g/dL

#9. ALB Albumin g/dL

#10. A/G Ratio Albumin and Globulin Ratio A/G

#11. Selector field used to split the data into
two sets (labelled by the experts)

# ================== #
#     CLEAN DATA     #
# ================== #

# Check for NA values.

clean.liver <- liver.data

which(is.na(clean.liver), arr.ind = TRUE)

# HAD TO REMOVE ROWS WHERE ALBUMIN AND GLOBULIN
RATIO WAS MISSING "NA".


clean.liver <- na.omit(clean.liver)

names(clean.liver)[ncol(clean.liver)] <-
"Liver_Disease"

# Turn response variable into 1 or 0.

clean.liver[, "Liver_Disease"] <-
ifelse(clean.liver$Liver_Disease == 1, 1, 0)

# Male and Female as 1 and 0 respectively.

clean.liver[ , "Gender"] <-
ifelse(clean.liver$Gender == "Male", 1, 0)

# Turn all variables into "numeric"
```

```
#as.numeric(clean.liver$Age)

#as.numeric(clean.liver$Gender)

clean.liver$Age

clean.liver$Gender


# SPLIT INTO TRAINING AND TEST DATA.

# The data has been split 80/20.

set.seed(1)

train_idx <- sample(nrow(clean.liver),
floor(0.8*nrow(clean.liver)), replace = FALSE)

liver.train <- clean.liver[train_idx, ]


# REINDEX

row.names(liver.train) <- 1:nrow(liver.train)

liver.test <- clean.liver[-train_idx, ]

row.names(liver.test) <- 1:nrow(liver.test)


# EXPLAIN THAT AGE FOR PEOPLE OVER 89 IS
CATEGORIZED AS 90.


# ================= #
#  DATA EXPLORATION #
# ================= #


# NOTE REMEMBER THIS IS ALL TRAINING DATA.

# Data Distribution


liver.train %>%

     keep(is.numeric) %>%

     gather() %>%

     ggplot(aes(value)) +

       facet_wrap(~ key, scales = "free") +

       geom_histogram()


# Summary of data (training).

summary(liver.train)


# Create Table

gender_table <- table(liver.train$Gender,
liver.train$Liver_Disease)


# Mean values of variables no disease and disease.

disease <- subset(liver.train, subset =
liver.train$Liver_Disease == 1)[ , -c(2, 11)]

no.disease <- subset(liver.train, subset =
liver.train$Liver_Disease == 0)[ , -c(2, 11)]

disease_mean <- apply(disease,2, FUN = mean)

no.disease_mean <- apply(no.disease, 2, FUN =
mean)

disease_mean <- data.frame(disease_mean)

no.disease_mean <- data.frame(no.disease_mean)

predictors_mean <- cbind(disease_mean,
no.disease_mean)

variables <- data.frame(Variables =
rownames(predictors_mean))

predictors_mean <- cbind(variables,
predictors_mean)

rownames(predictors_mean) <-
c(1:nrow(predictors_mean))

pivot <- predictors_mean[ , c(2,3)]

rownames(pivot) <- predictors_mean[ , 1]

par(mar=c(1,9,2,1))

barplot(t(as.matrix(pivot)), names.arg =
rownames(pivot), xlab = "Value", beside = TRUE,
col = c("lightblue", "lightgreen"),

        horiz = TRUE, las = 2, cex.names = 0.6,
legend = TRUE)

# Distribution of People with Liver Disease vs No
Disease


par(mar=c(5, 4,4,2)+.1)

barplot(table(liver.train$Liver_Disease), main =
"Patients with Liver Disease", names.arg = c("No",
"Yes"),

        ylab = "Number of Patients", xlab = "Liver
Disease", col = c("lightblue", "lightgreen"))


# Create a graph showing the percentage of men vs
women who have liver disease.

barplot(gender_table, main = "Liver Disease by
Gender", xlab = "Gender", col = c("red",
"darkred"), names.arg = c("Female", "Male"),
beside = TRUE)
```

```
legend(x = 1, y = 250, c("Disease", "No Disease"),
fill = c("darkred", "red"))



# Boxplot of distribution of ages with liver
disease versus no liver disease



par(mar=c(6, 4,4,2)+.1)

boxplot(Age~Liver_Disease, data = liver.train,
xlab = "Liver Disease", ylab = "Age (Years)", main
= "Age given liver disease or not", ylim = c(0,
100),

col = c('lightblue','lightgreen'), names = c("No",
"Yes"))



# create graph with liver disease vs not liver
disease given Bilirubin.

plot(liver.train$Total_Bilirubin,
liver.train$Liver_Disease, col =
(liver.train$Liver_Disease+1))



# Shows that there is a threshold from which
somebody will most likely have liver disease.



# CORRELATION MATRIX (HELPS US TO FIND
MULTICOLLINEARITY)

corr_matrix <- liver.train

names(corr_matrix) <- c("Age", "Gender",
"T.Bilirubin", "D.Bilirubin", "Alkaline.Phos",

"Alamine.A", "Aspartate.A", "Total.Proteins",
"Albumin", "A/G Ratio", "Liver.Disease")

cm <- corr_matrix[ , c("Age", "T.Bilirubin",
"D.Bilirubin", "Alkaline.Phos",

                "Alamine.A", "Aspartate.A",
"Total.Proteins", "Albumin", "A/G Ratio")]

cor(cm)

corrplot(cor(cm), type="upper", order="hclust",
col=brewer.pal(n=8, name="RdYlBu"))

# Plots of co-linear variables

plot(liver.train$Total_Protiens,
liver.train$Albumin, pch = 21, bg = 'lightblue',
xlab = "Total Proteins g/dL", ylab = "Albumin
g/dL")

legend(x =3, y =5, legend = "r=0.79")

plot(liver.train$Albumin_and_Globulin_Ratio,
liver.train$Albumin, pch = 21, bg = 'lightblue',
xlab = "A/G Ratio", ylab = "Albumin g/dL")
```

```
legend(x=2, y=5, legend = "r = 0.7")

plot(liver.train$Alamine_Aminotransferase,
liver.train$Aspartate_Aminotransferase, pch = 21,
bg = 'lightblue', xlab = "Alamine Aminotransferase
IU/L", ylab = "Aspertate Aminotransferase IU/L")

legend(x = 0, y = 4000, legend = "r = 0.73")

plot(liver.train$Total_Bilirubin,
liver.train$Direct_Bilirubin, pch = 21, bg =
'lightblue',

xlab = "Total Bilirubin g/dL", ylab = "Direct
Bilirubin g/dL")

legend(x =0, y = 16.5, legend = "r =0.85")



cor(liver.train$Total_Protiens,
liver.train$Albumin)

cor(liver.train$Albumin_and_Globulin_Ratio,
liver.train$Albumin)

cor(liver.train$Alamine_Aminotransferase,
liver.train$Aspartate_Aminotransferase)

cor(liver.train$Total_Bilirubin,
liver.train$Direct_Bilirubin)

# ================================= #
# BUILDING LOGISTIC REGRESSION MODEL #
# ================================= #



# VARIABLES TO KEEP



# Albumin_and_Globulin_Ratio + Total_Protiens +
Age + Alamine_Aminotransferase +
Alkaline_Phosphotase +

# Total_Bilirubin



lr.model7 <- glm(formula =
Liver_Disease~Albumin_and_Globulin_Ratio +
Total_Protiens + Age + Alamine_Aminotransferase +
Alkaline_Phosphotase +Total_Bilirubin, family =
binomial(link = 'logit'), data = liver.train)

lr.model8 <- glm(formula = Liver_Disease ~
Total_Bilirubin + Alamine_Aminotransferase,

                 family = binomial(link =
"logit"), data = liver.train)

lr.model9 <- glm(formula = Liver_Disease ~
Total_Bilirubin + Alamine_Aminotransferase +


I(Total_Bilirubin*Alamine_Aminotransferase),
family = binomial, data = liver.train)
```

```
lr.model10 <- glm(formula =
Liver_Disease~Total_Bilirubin+Alamine_Aminotransfe
rase+Age,

                  family = binomial(link =
"logit"), data = liver.train)

lr.model11 <- glm(formula =
Liver_Disease~Total_Bilirubin+Alamine_Aminotransfe
rase+Age+


I(Total_Bilirubin*Alamine_Aminotransferase),
family = binomial,

                  data = liver.train)

summary(lr.model11)



#=============#
#  K FOLD CV  #
#=============#

set.seed(10)

cost2 <- function(r ,pi =0) mean(abs(r-pi) > 0.6 )

models <- list(lr.model8, lr.model9, lr.model10,
lr.model11)

cv.models2 <- c()

# Threshold 0.6

for(i in 1:length(models)){

  cv.models2[i] <- cv.glm(liver.train,
models[[i]], cost = cost2, K=10)$delta[1]

}

cv.models2

# ============= #
#   PREDICTION  #
# ============= #

prex11 <- predict(lr.model11, newdata =
liver.test, type = 'response')

classx11 <- ifelse(prex11 >= 0.60, 1, 0)

conf_matrix11 <- table(classx11,
liver.test$Liver_Disease)

# ============== #
#     TREES      #
# ============== #

set.seed(1)

disease.train <- ifelse(liver.train$Liver_Disease
== 1, "Yes", "No")

disease.test <- ifelse(liver.test$Liver_Disease ==
1, "Yes", "No")

l.train <- cbind(liver.train, disease.train)
```

```
l.test <- cbind(liver.test, disease.test)


treex <- tree(disease.train~.-Liver_Disease, data
= l.train)

treepred <- predict(treex, l.test, type = 'class')

plot(treex)

text(treex, pretty=0)



# confusion matrix

cm.tree <- table(treepred, l.test$disease.test)



# CV tree

cv.treex <- cv.tree(treex, FUN = prune.misclass)

plot(cv.treex$size, cv.treex$dev, type = 'b', main
= "Deviance vs Tree Size", xlab = "Terminal
Nodes", ylab = "Deviance",  pch =21, bg =
'lightgreen', ylim = c(140, 153))

plot(cv.treex$k, cv.treex$dev, type = 'b', main =
"Deviance vs alpha value", xlab = "Alpha
paramater", ylab= "Deviance", pch = 21,bg =
'lightgreen', ylim = c(140, 153))

prune.treex <- prune.misclass(treex, best =7)

summary(prune.treex)

plot(prune.treex)

text(prune.treex, pretty =0)

# predict with tree given best alpha.

treeprune.predict <- predict(prune.treex, l.test,
type = 'class')

cm.treeCV <- table(treeprune.predict,
l.test$disease.test)


# ==============================#
# PRINCIPAL COMPONENT ANALYSIS
# ============================ #



pca_liver <- prcomp(liver.train[ , -
ncol(liver.train)])

barplot(summary(pca_liver)$importance[2, ], main =
"Proportion of Variance Explained",

        col = "lightblue")

plot(pca_liver$x[ ,1], pca_liver$x[, 2], col =
(liver.train$Liver_Disease+2), xlab = "PC1", ylab
= "PC2")
```