Evaluating the Association between Liver Function Tests, Treatment Options, and its Impact on Patients' Mortality Rate in Hepatitis

IUPUI, Indianapolis IN 46202, USA

Abstract. Hepatitis is the inflammation of the liver most often caused by viruses. But, extended alcohol abuse, medications and secondary effects of other diseases can also result in inflammation of the liver. The main purpose of our study was to understand the treatment options for hepatitis and to evaluate the impact of drugs in biochemical components of liver and how these factors are related to mortality rate in patients with hepatitis. The hepatitis data is collected from UCI Irvine Machine Learning Repository. Analysis was performed using Recursive feature elimination and Logistic Regression. Our logistic regression model gave 91 percent accuracy. The results of our model depicted that, there is an association between liver function tests, treatment options, and mortality rate.

Keywords: Hepatitis, treatment, bio-chemical components, mortality.

1 Introduction

Hepatitis is inflammation of the liver most often caused by viruses, but can also result from extended alcohol abuse, autoimmune diseases which causes immune system to attack healthy tissues, fat which may lead to fatty liver disease, medications, or as a secondary effect of other diseases.[1] It most commonly affects people between the ages of 25 and 44. Hepatitis is usually diagnosed using a liver function test for alanine amino- transferase (ALT), aspartate aminotransferase (AST), alkaline phosphate, and bilirubin. Common treatments for hepatitis include a combination of antiviral medications and steroids.

In this study, we wanted to know if there an association between liver function tests (alkaline phosphatase, SGOT) and other biomarkers such as bilirubin, albumin, PROTIME (prothrombin time) and treatment options (steroids, antivirals) and to analyze their impacts on patients mortality rate based on the treatment options and the levels of biochemical components of the liver. And also to analyze the relationship between individual treatment options, specific enzyme and biochemical levels, symptoms and their impact on mortality rates in hepatitis.

Treatment options for hepatitis and how these drugs are related to difference in the levels of biochemical components of liver should be understood. Ideally, we hoped to develop a model that could predict the best treatment options based on levels of those biochemical components. We also wanted to understand how these biomarkers related to deaths in patients suffering from hepatitis.

Our null hypothesis was that there is no association between treatment, liver function tests, and the mortality rate in hepatitis patients. Our alternative hypothesis expected an association between treatment, liver function tests, and mortality rates in hepatitis patients.

2 Methodology

2.1 Data Collection

Our data came from the UCI Irvine Machine Learning Repository located at https://archive.ics.uci.edu/ml/datasets/Hepatitis. It was downloaded as a comma-separated text file with no headers. An accompanying names files provided additional information about the dataset, including the column headers for the data file. This information was read into python and combined to produce the working dataset. The UCI website is one central database that contains hepatitis specific datasets for multiple variables observed by various scholars in their practice. The Hepatitis database is a significant tool in both clinical practice and for research purposes. A well organized and developed database can act as a reference range for researchers who will want to compare their findings with what exists in data sets to establish consistency of results and any noticeable trends that are taking place about disease infection, presentation, laboratory findings and treatment modalities.

2.2 Data Exploration

Our dataset contains 155 patient records and 20 related attributes which can be logically divided into four categories:

- Class (alive/dead) of patient
- Patient Demographics (Age, Gender)
- Treatment Options (Steroids, Antivirals)
- Biochemical Test Results (Bilirubin, PROTIME, etc.)

To begin, basic descriptive statistics were calculated (Table 1). Then univariate analysis for every feature was done using distribution plots. Examples of these plots can be seen in Figure 1 and Figure 2. These plots helped us uncover insights about our data. As shown in Fig. 1, we found that some of the features including steroid, fatigue, etc. contained unknown values. Also, but for protime, all the other continuous variables were skewed and not normally distributed. The python code used to create these plots can be found in Appendix A.

Fig. 1. Distibution of Categorical Variables

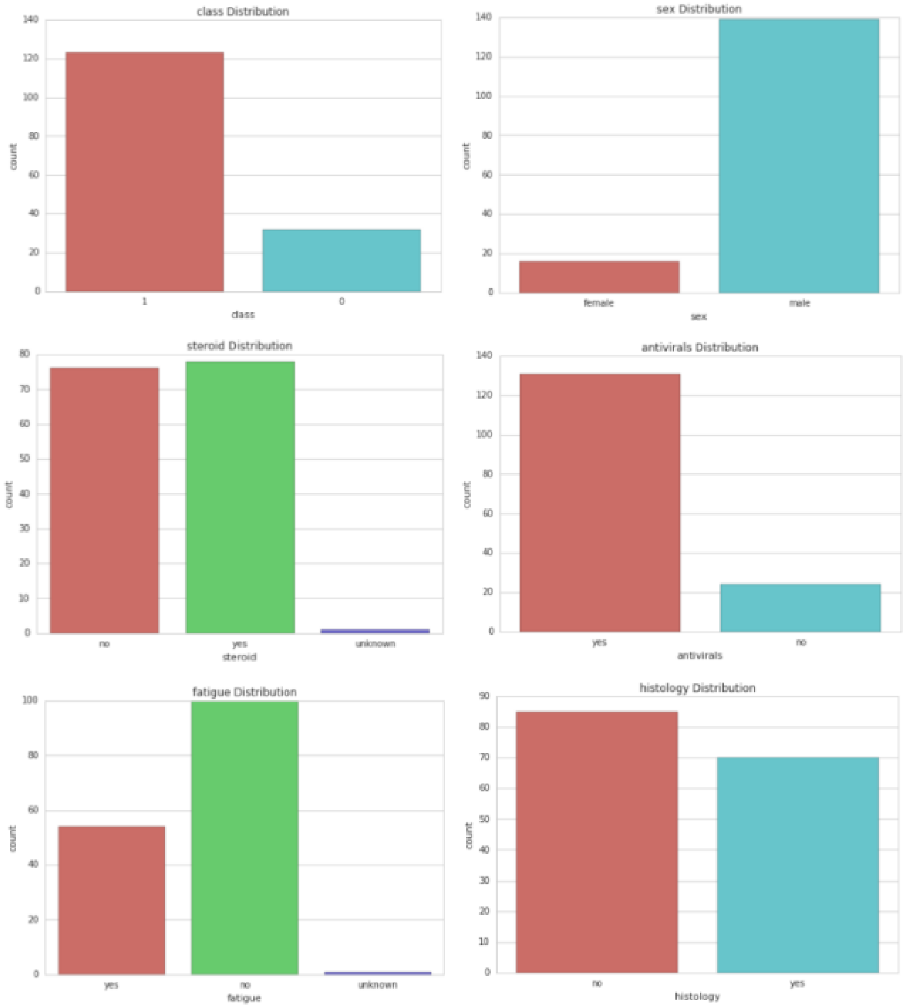


Fig. 2. Distribution of Continuous Variables

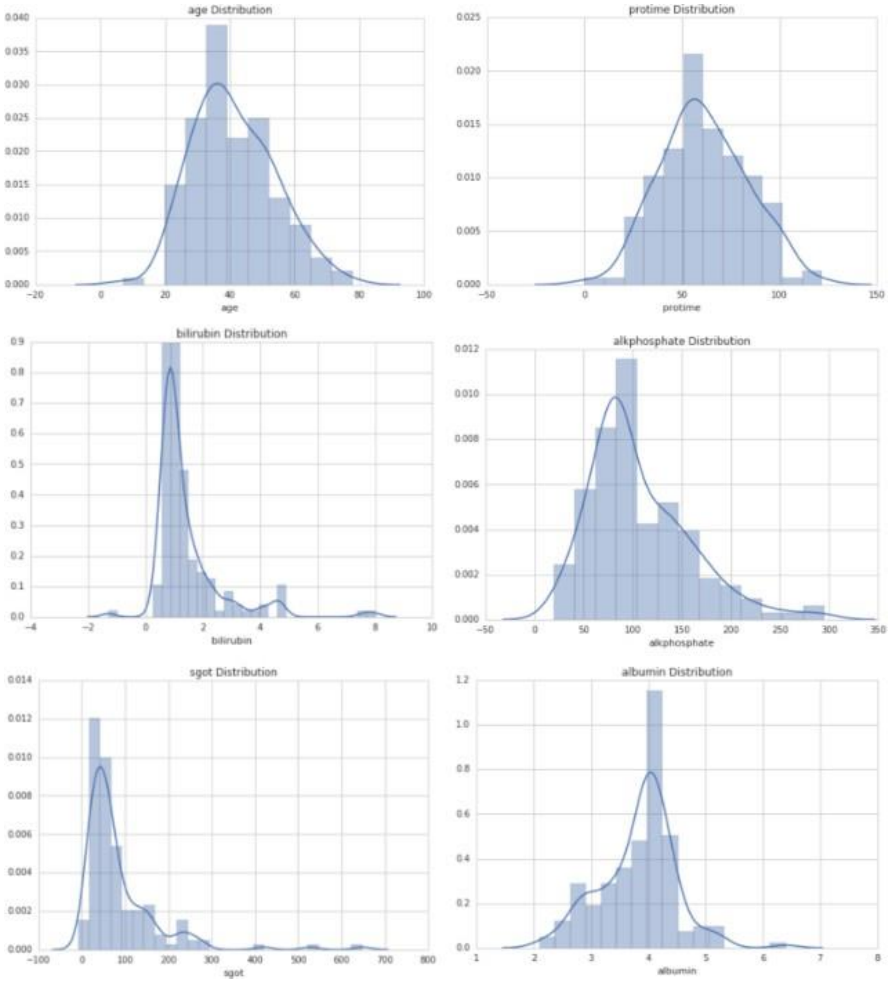


Table 1. Descriptive Statistics

Statistic	age	bilirubin	alkphosphate	sgot	albumin	protime
count	155	149	126	151	139	87
mean	41.2	1.43	105	85.9	3.82	62.1
std	12.6	1.21	51.5	89.7	0.651	22.9
min	7	0.30	26.0	14.0	2.10	0
25%	32	0.7	74.3	31.5	3.4	46
50%	39	1.0	85.0	58.0	4.0	62
75%	50	1.5	132	101	4.2	76.5
max	78	8.0	295	648	6.4	100

Data Exploration Web Application

A data exploration web application was developed in R using the *shiny* package. This app is being hosted on shinyapps.io at https://i501.shinyapps.io/visualizer/. The code used to create the application is available in Appendix B.

The app had three tables. The first tab, Plots, allows the user to look at relationships between the continuous data and categorical data as a box plot. It also allows some basic filter such as only looking at patients who lived, or who took steroids, etc.

The second tab, Mortality, allows the user to explore mortality rates based on a combination of two other factors as a heat map. This was meant to allow one to quickly spot patterns in the data.

The final tab, Classification, allows the user to run 15 different classification models, while altering how the missing values were handled, the split between test and training sets, and which features to include in the model. Tables are provided to see which records were in the training set and which were in the test set for an individual run. It also produces a table of predicted outcome for each model and a model summary table showing the accuracy, of the each model. Additional info boxes allow the user to see the specifics of each model run and the results of individual tests.

2.3 Data Cleaning and Extraction

Handling Missing Values

All the fields containing numeric data had missing values. (See Table 2) Because most classification models cannot handle such data, these values had to be replaced or removed. It was decided by the group with consultation with the professor to replace these missing values with random values from similar distributions.

To that end, normality tests were run on all features containing missing values. If the test showed the data were normally-distributed then missing values

Table 2. Missing Data

Statistic	Missing Count
bilirubin	6
alkphosphate	29
sgot	4
albumin	16
protime	68

were replaced with values from the matching normal distribution, otherwise they were replaced with randomly-generated values. In this dataset, only the protime field was found to be normally-distributed. The other fields; bilirubin, alkphosphate, sgot, and albumin, were replaced with random values from the same field.

Encoding Categorical Variables

A majority of features in our dataset are categorical in nature. Some machine learning algorithms like Logistic Regression do not support categorical variables and therefore transforming them to numeric values becomes inevitable. We chose to use an approach called Dummy Coding or One Hot Encoding. Each category value is converted into a new column and assigned a 1 or 0 (True/False) value to the column. This has the benefit of not weighting a value improperly but does have the downside of adding more columns to the data set [2]. However the new columns that were created for unknown values were dropped. We started with 20 columns and ended up with 33 columns after creating dummies and dropping columns corresponding to unknown values.

2.4 Feature Selection

Feature selection was performed by using Recursive Feature Elimination (RFE). The idea behind RFE is that a model is created including every available feature. That model is tested and either the best or worst feature is removed from the feature set. This process is repeated until all of the features have been removed. Once this process is completed the features are ranked by when they were removed from consideration. The researcher can then select the features to be used based on this ranking. [3] In this analysis this was performed using the *RFE* function from the *sklearn* library. Several logistic regression models with varying number of features were built and the model with 29 features was found to give maximum accuracy.

Boruta Algorithm

A second feature selection algorithm, the Boruta Algorithm, was also attempted but proved difficult to implement in python with our data. It was implemented in *R* with the following code:

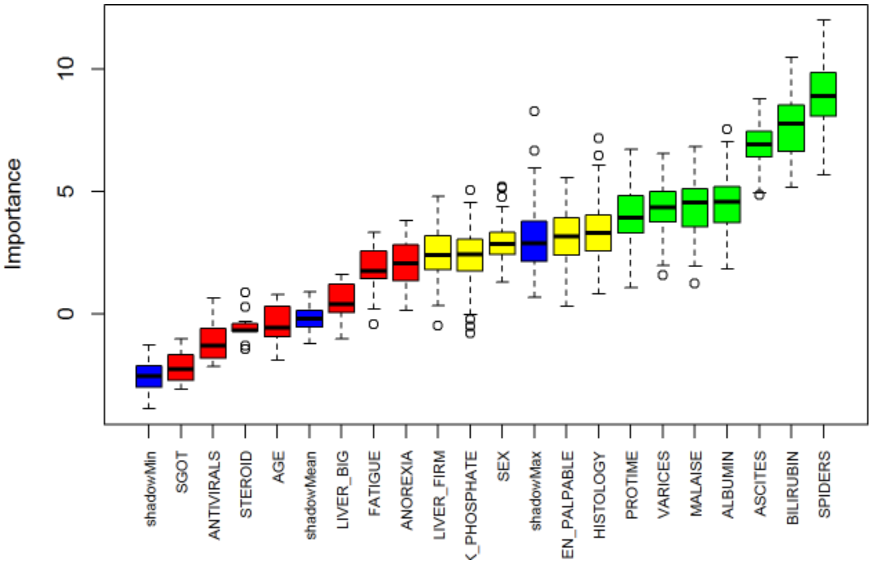
```
library(Boruta)

boruta.train <- Boruta(Class~., data = train)
print(boruta.train)
```

The results of the boruta algorithm were visualized as a boxplot (Figure 3) with the following R code:

```
plot(boruta.train, xlab=””,xaxt = ”n”)
lz <- lapply(1:ncol(boruta.train$ImpHistory), function(i) {
  boruta.train$ImpHistory[
         is.finite(boruta.train$ImpHistory[,i]),i
  ]
})
names(lz) <- colnames(boruta.train$ImpHistory)
Labels <- sort(sapply(lz, median))
axis(side=1,las=2,labels= names(Labels),
      at=1:ncol(boruta.train$ImpHistory),cex.axis=0.7)
```

Fig. 3. Boxplot of Boruta Algorithm Results



This identified 6 features as significant: protime, varices, malaise, albumin, ascites, bilirubin, and spiders. Note: The R analysis was done without creating dummy variables for the categorical data.

2.5 Classiftcation Models

Three classification models were explored during this analysis: Logistic Regression, Support Vector Machines, and K-Nearest Neighbors.

Logistic Regression

Logistic Regression is a machine-learning algorithm which is used to predict the dependent variable. As our dependent variable is dichotomous (contains data as 0(dead) and 1(alive)), we chose to include logistic regression. Logistic regression requires the dependent variable to be binary, and the independent variables are either categorical or continuous data. One of the assumptions of the logistic regression model is that there is no multi-collinearity among the independent variables. The model building process includes splitting the data into training and test datasets in a 70:30 ratio. The model is trained using the training dataset and predicted values are vetted utilizing the test data.

Support Vector Machines

Support vector machines (SVM) is a supervised machine learning algorithm which is best suited for classification or regression problems. Our target variable is 0 (dead) or 1 (alive). We want to divide our data points into two classes and then predict the class into which any new data points belong. Our model generates many hyper planes which are used to classify the data. The classification model is built on the best hyper plane. The best hyper plane is the one with maximum distance (margin) from the nearest data point on either side. The objective of a Linear Support Vector Classifier (SVC) is to fit the data provided, returning a best fit hyper plane that divides, or categorizes, your data. Once we get the hyper plane we feed the independent variables to the classifier to predict the target variable class. Data is randomly split into training and testing datasets in a 70:30 ratio. The classifier is built on the training dataset, and the testing dataset is used to predict the target variable class and vet the model.

K-Nearest Neighbors

K-Nearest Neighbors (KNN) is a supervised machine learning algorithm. This classifier is non-parametric, so it does not make any assumptions about the underlying data distribution. In other words, the model structure is determined from the data. It is an instance-based algorithm which means it memorizes the training data points rather than building the model. That memorized knowledge is used to predict the label of unseen test observations. The KNN algorithm provides the K closest training points for a given unseen test data point. Euclidean distance is used to calculate the distance between the two data points, i.e., Training and Test data points. Based on the conditional probability of each class the input data point is assigned a class with maximum probability.

2.6 Model Validation and Performance Analysis

Basic performance analysis procedure were performed on all constructed models. These procedures included confusion matrices, and calculation of accuracy, precision, recall, and f1-score. In addition, ROC curves and K-folds cross-validation were performed the logistic regression model. The resulting values for various performance metrics across all the models built are shown in Table 3.

Table 3. Performance Metrics Comparison

Model	Accuracy	Recall	Specificity	Precision	F1-Score
Logistic Regression	0.91	0.91	0.57	0.91	0.91
SVM	0.89	0.89	0.42	0.88	0.88
KNN	0.85	0.85	0.16	0.81	0.81

Accuracy Scores

Accuracy score gives the percentage of the number of correct predictions made by total number of predictions made. Table 3 shows the accuracy scores generated for all three models. This score does not incorporate the incorrect predictions made by the model.

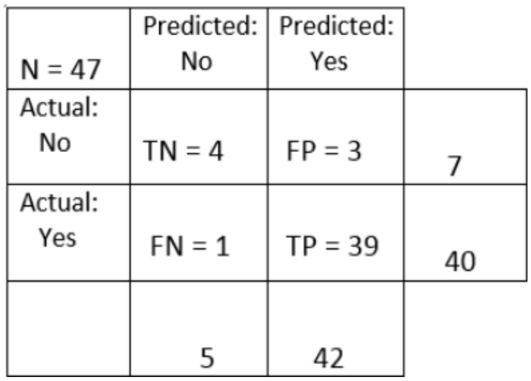
Confusion Matrix

Confusion matrices were generated for all three models. They show the true positives, true negatives, false positives and false negatives. The confusion matrix for logistic regression model is shown in Figure 4. Class 0 (dead) is the negative class and class 1 (alive) is the positive class.

ROC curves

Every point in ROC curve is a pair of sensitivity (true positive rate) and 1-specificity (false positive rate) for a particular threshold. The area under ROC curve is a performance metric that we want to maximize. Fig 4 shows the ROC curve for logistic regression.

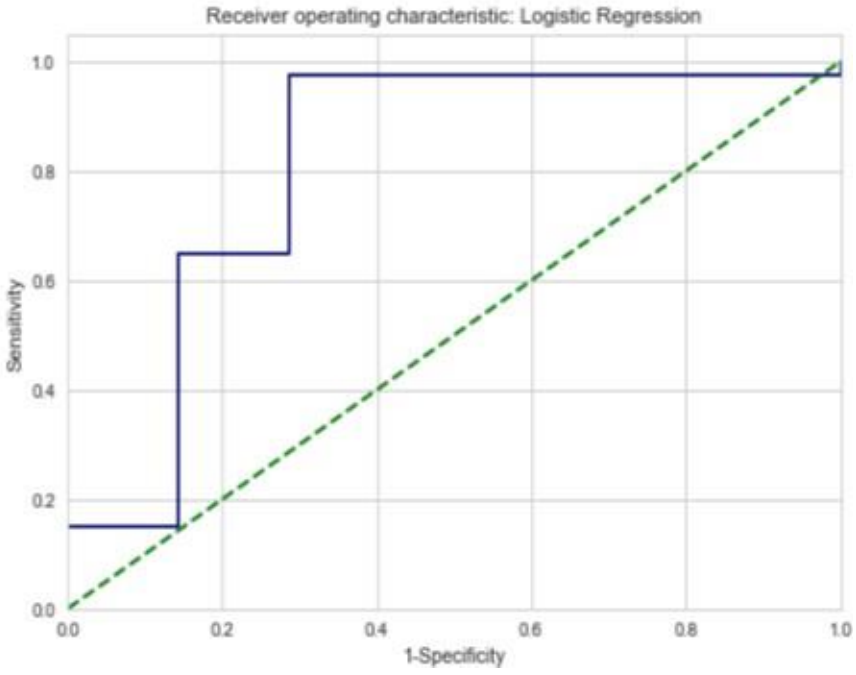
Fig. 4. Confusion Matrix of Logistic Regression Model



N = 47	Predicted: No	Predicted: Yes	
Actual: No	TN = 4	FP = 3	7
Actual: Yes	FN = 1	TP = 39	40
	5	42	

Fig. 5. ROC Curve for Logistic Regression Model

```
Area under ROC: 0.810714285714
```



K-folds Cross Validation

When a model is built to predict the dependent variable by providing the required parameters and tested using test data the model tries to fit too closely to training data and fails to predict anything useful on yet unseen data. This is called overfitting of model. We have used K-fold cross validation to overcome overfitting. The idea is to randomly divide the data into k equal sized parts. We hold out part k, fit the model to the other K-1 parts, and then obtain the predictions for the left-out kth part. This is done in turn for each part k = 1,2,.K and then the average of all the scores is used to measure the accuracy of the model. As our model accuracy and K-fold scores are close enough which indicates the generalization of our model well. Logistic regression model was further validated using 10-fold cross validation to overcome any selection bias problems. sklearn KFold was used to create 10 samples. In 10 iterations, different sample were used every time for test, leaving the rest for training. This yielded an average accuracy of 0.84 for the logistic regression model built.

3 Discussion

A study on hepatitis disease diagnosis using multilayer Neural network with levenberg marquardt training algorithm conducted by M.Serdar Bascil shows the classification accuracy of 91.87% via tenfold cross validation. Another study using novel hybrid method based on support vector machine and simulated annealing (SVM-SA) conducted by Javad Aalimi Sartakhti shows the accuracy of 96.25%. [4] we have applied logistic regression and support vector machines models. Logistic regression included splitting the data into train and test in 70:30 ratio. The model was trained using training dataset and predicted values were vetted using the test data. The same was applied with the SVM model. But, the accuracy varied with slight difference among the two models with 0.91 and 0.83 respectively.

3.1 Findings

Our analysis shows that the null hypothesis should be rejected. There is a relationship between the variables tested. In regression the classifier coefficient describes the weight that each independent variable is having on dependent variable. In regression with multiple independent variables, the coefficient tells you how much the dependent variable is expected to increase when that independent variable increases by one, holding all the other independent variables constant.

The weights of the variables in the below table represents the strength of association between our dependent variable class and independent variables. Since found an association from the results of the table, we are rejecting our Null hypothesis which states that there is no association between laboratory values, treatment options and mortality rate.

Table 4. Performance Metrics Comparison

Weights	variables
0.932	sex_female
0.702	albumin
0.637	liverfirm_no
0.540	spleenpalpable_yes
0.526	spiders_yes
0.534	liverbig_no

3.2 Limitations

The hepatitis dataset was minimal and very old. With only 155 records it was hard to train an accurate model. Also, there were many missing values. For instance, protime had 68 (43%) missing values. Replacing these values with random values made it difficult to get meaningful results from the model.