Liver Disease Prediction Project

Group 7: Mohan Liu, Yimin Yuan, Xinyu Diao

2023-05-09

Introduction

Liver disease is a significant public health concern worldwide, and early detection can greatly improve patient outcomes. This study aims to develop a predictive model to identify liver disease in patients using a dataset collected from India, including 416 with liver disease and 167 without liver disease. The dataset contains 11 variables, including age, gender, and various blood markers, and a special variable named "dataset", it represents the liver disease diagnosis, split the data into two sets (1 = patients with liver disease, 2 = patients with no disease).

Our approach involves comparing different machine learning algorithms, including logistic regression, K-Nearest-Neighbors (KNN), decision tree, random forest, and boosting, to select the model with the highest accuracy. The main conclusions drawn from our study show that certain algorithms perform better in predicting liver disease, which can be useful for medical professionals, contributing to better prevention and management strategies.

Related Work

Previous research has explored different approaches to predict liver disease using machine learning techniques. For instance, Bora et al. (2016) compared the performance of multiple classifiers, including logistic regression, decision trees, and support vector machines, on the same Indian liver patient dataset. They concluded that logistic regression outperformed other methods in terms of accuracy. However, our study extends this work by incorporating additional algorithms such as KNN, random forests, and boosting, providing a more comprehensive comparison and analysis.

Another study by Sathyadevan et al. (2014) investigated the use of artificial neural networks (ANN) to predict liver disease using a different dataset. They reported promising results, but the applicability of their findings to the Indian liver patient dataset remains unclear. Therefore, our study seeks to address this gap by comparing multiple methods on a consistent dataset, ultimately providing a better understanding of the most effective techniques for liver disease prediction.

Methods

To find the best machine learning model for predicting liver disease in this dataset, we tried several different algorithms. Specifically, we used logistic regression, K-nearest neighbors (KNN), decision tree, random forest, and gradient boosting to build predictive models.

The script first fits a binary **logistic regression** model, using 'Dataset' as the target variable and all other variables in the 'train' dataset as predictors. To avoid collinearity, it removes any features with a correlation higher than 0.75. After preprocessing, the model is refitted with the updated dataset. The model's performance is evaluated by generating predictions on the train and test datasets. Predicted probabilities above 0.5 are classified as 1, otherwise as 0. The error rate, or rate of misclassifications, is calculated for both datasets, providing an indication of the model's accuracy.

Then, we employed the **K-nearest-neighbors** (KNN) algorithm as our second model. To ensure that variables can be measured in the same scale in distance calculations, we standardized both the training and test datasets. Then, we used a 5-fold cross validation to find the best k for this data.

After that, we tried a **decision tree** model to do the prediction, because it is easy to build and easy to interpret. Considering the original dataset is imbalanced, which means we have more samples with disease (416) than those without the disease (167), we adjusted the class distribution by upweighting the observations without the disease in the training set using the ovun.sample function.We used the Gini impurity index for splitting criteria here. Also, we applied a 5-fold cross validation to find the best size for the tree and pruned it.

While decision tree models are relatively simple to build, they may not be flexible enough to accurately classify new samples. To improve performance, we applied a **random forest** algorithm, which combines multiple decision trees and takes advantage of their diversity to make more accurate predictions. We chose 300 as the size of the forest because the number of predictors in this dataset is relatively small. Furthermore, variable importance is visualized to identify the most influential predictors.

Boosting is a machine learning algorithm that iteratively builds an ensemble of weak models to create a strong predictive model. In our analysis, we used the boosting algorithm to improve the performance of our predictive model. We set the shrinkage parameter to be 0.2, which controls the learning rate of the algorithm and prevents overfitting, and we set the number of trees to be 5000. We can also obtain a plot of the relative influence of different variables in the model.

For each model, we calculate the training and test errors, as well as the accuracy, which enables us to compare their performance and identify the most effective approach for predicting liver disease in patients.

Data and Experiment setup

The indian liver dataset (https://www.kaggle.com/datasets/uciml/indian-liver-patient-records) is collected from the North East of Andhra Pradesh, India. It contains 11 variables, including two simple demographic variables (age and gender), and various blood markers that may relate with liver disease (for example total bilirubin, alanine aminotransferase), and a special variable named "dataset", which represents the liver disease diagnosis, split the data into two sets (1 = patients with liver disease, 2 = patients with no disease) and it is the response variable for our model.

Firstly, we imported the data, removed missing values, and recode the diagnosis variable to a binary format (0 for no liver disease, 1 for liver disease). Then, we made some descriptive analysis to show the basic information. The dataset consists of 583 patient records, including 441 males and 142 females. Histogram plots of the continuous variables are shown below. From the plots, we can see that the distributions of age, total_proteins, albumin, and albumin and globulin ratio are approximately symmetrical and normally distributed, while the other variables are right-skewed. In addition, the dataset is slightly imbalanced, with 416 individuals diagnosed with liver disease and 167 without.

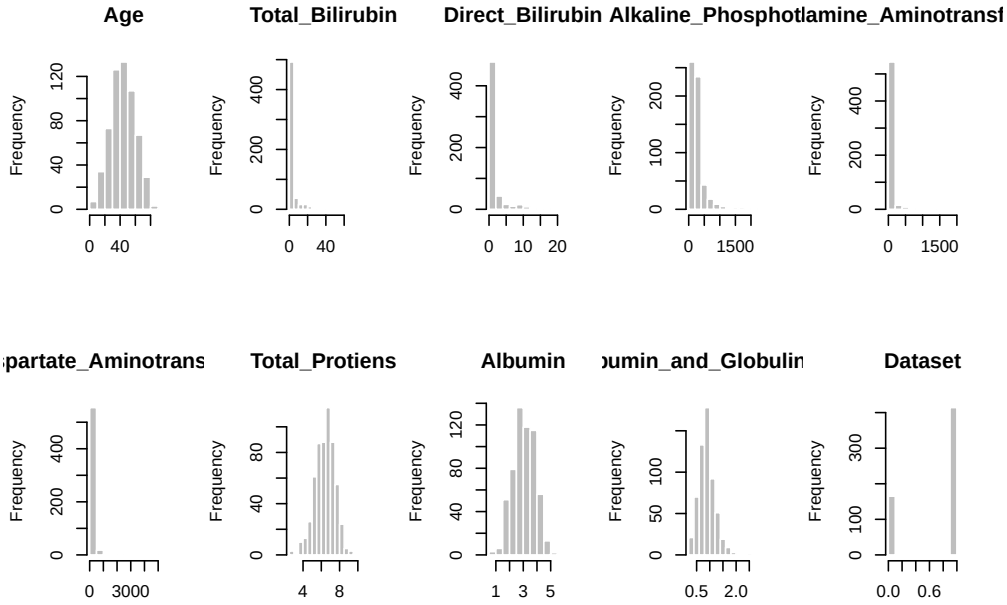

Figure 1: Histogram of continuous varibles

In addition, we performed a correlation analysis to examine the linear relationships between variables. The resulting correlation matrix was visualized using a heat map, allowing us to easily identify strong positive and negative correlations between the variables.

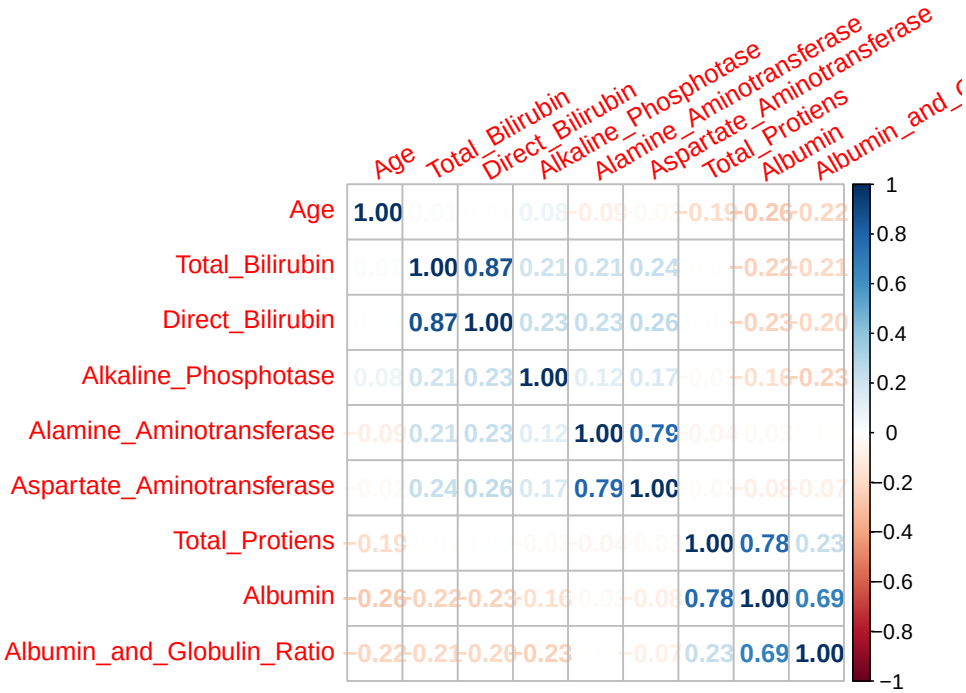

Figure 2: Correlation heat map of continuous variables

To evaluate the performance of our machine learning algorithms, we randomly split the dataset into training (60%) and testing (40%) sets. This allowed us to train our models on a subset of the data and test them on a completely independent set, ensuring that our models were not overfitting to the data. By doing so, we were able to accurately compare the performance of each model and select the best one for predicting liver disease.

Result

1 - Logistic Regression

The logistic regression analysis revealed that the variables age, Total_Bilirubin, and Alamine_Aminotransferase have significant contributions to the model. Although the test error is relatively small, the model exhibits a high false positive rate and poor prediction accuracy for individuals without the disease. Hence, it may not be the optimal choice for liver disease prediction.

Table 1: Coefficients in logistic regression

	Estimate	Std. Error	z value	Pr(>|z|)
(Intercept)	-3.4709170	1.4671136	-2.3658135	0.0179905
Age	0.0192616	0.0083593	2.3042197	0.0212103
GenderMale	0.0432100	0.3034342	0.1424032	0.8867615
Total_Bilirubin	0.0173449	0.0889495	0.1949976	0.8453948
Direct_Bilirubin	0.2288544	0.2426797	0.9430307	0.3456652
Alkaline_Phosphotase	0.0010136	0.0009114	1.1122082	0.2660486
Alamine_Aminotransferase	0.0128429	0.0066417	1.9336916	0.0531511
Aspartate_Aminotransferase	0.0047857	0.0048172	0.9934575	0.3204870
Total_Protiens	0.8720369	0.4070129	2.1425290	0.0321509
Albumin	-1.2912384	0.7886929	-1.6371879	0.1015912
Albumin_and_Globulin_Ratio	0.7797073	1.1709948	0.6658503	0.5055068

```
## [1] "Training error of logistic regression:  0.250720461095101"
```

```
## [1] "Test error of logistic regression:  0.310344827586207"
```

Table 2: Confusion matrix for logistic regression

	Predicted 0	Predicted 1
0	9	61
1	11	151

2 - KNN

In the KNN model, we found the best k in this training data to be 7 after the cross validation. The training error was found to be 0.22, suggestting that the model has good performance on the training data. The test error was found to be 0.35, which indicates that the model has moderate generalization performance. Furthermore, the confusion matrix reveals that the model has good prediction performance on actual patients, with a high true positive rate. However, the false positive rate is found to be high, indicating that a significant number of healthy individuals are misclassified as having the disease.

```
## [1] "Best K: 7"
```

```
## [1] "Training error of KNN:  0.219020172910663"
```

```
## [1] "Test error of KNN:  0.353448275862069"
```

Table 3: Confusion matrix for KNN

	Predicted 0	Predicted 1
0	14	56
1	26	136

3 - Decision Tree

Due to the imbalanced nature of the dataset, the simple decision tree model had a tendency to predict that all individuals had liver disease, since the majority (71%) of the observations did. To address this issue, we balanced the data to get a more equal number of individuals with and without the disease. The results show that, although the training and test error are higher in the decision tree model, the false positive rate is lower than in the previous model. This suggests that decision tree after data balancing is better at correctly identifying individuals without the disease than logistic regression and KNN, even though it may make more errors overall.

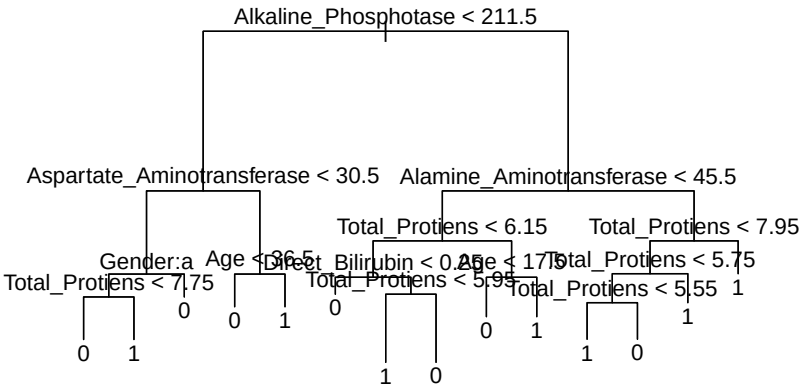

Figure 3: Plot of the final tree

```
## [1] "Training error of decision tree:  0.216138328530259"
```

```
## [1] "Test error of decision tree:  0.392241379310345"
```

Table 4: Confusion matrix for decision tree

	Predicted 0	Predicted 1
0	37	33
1	58	104

4 - Random Forest

After building a random forest model with 300 trees, we observed a significant improvement in the prediction accuracy in both the training and test sets, with a training error of 0 and a test error of 0.31. However, we still noted a high false positive rate in the predictions. Despite this, the random forest model remains the best-performing model among those we evaluated, we will try additional techniques to reduce the false positive rate, such as adjusting the decision threshold or applying further data balancing techniques. Additionally, the variable importance plot indicates that `aspartate_aminotransferase` is the most important variable in the random forest model.

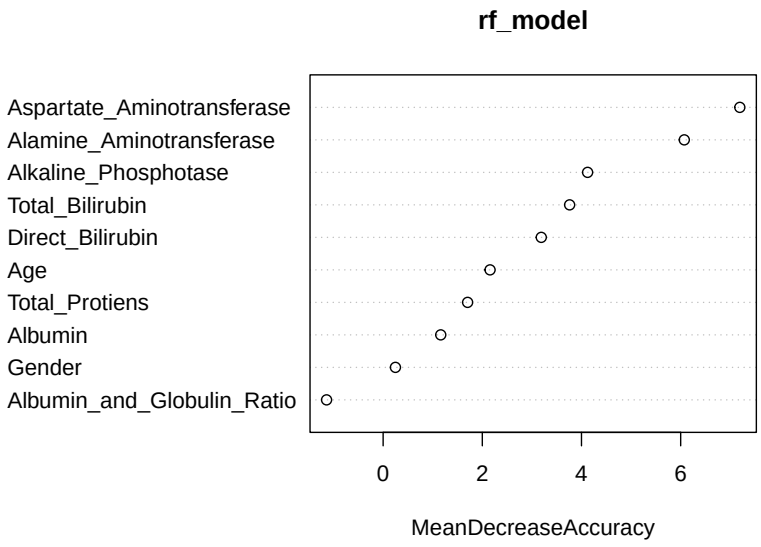

Figure 4: Variable importance plot from random forest

```
## [1] "Training error of random forest:  0"
```

```
## [1] "Test error of random forest:  0.310344827586207"
```

Table 5: Confusion matrix for random forest

	Predicted 0	Predicted 1
0	19	51
1	21	141

5 - Boosting

The relative influence plot generated from the boosting model revealed that the results are consistent with those obtained from the random forest and logistic regression models, where gender had the least influence on liver disease prediction. The model achieved a very low training error of nearly 0, while the test error was 0.34. However, like the other models, the boosting model had a high false negative rate which requires improvement in future iterations.

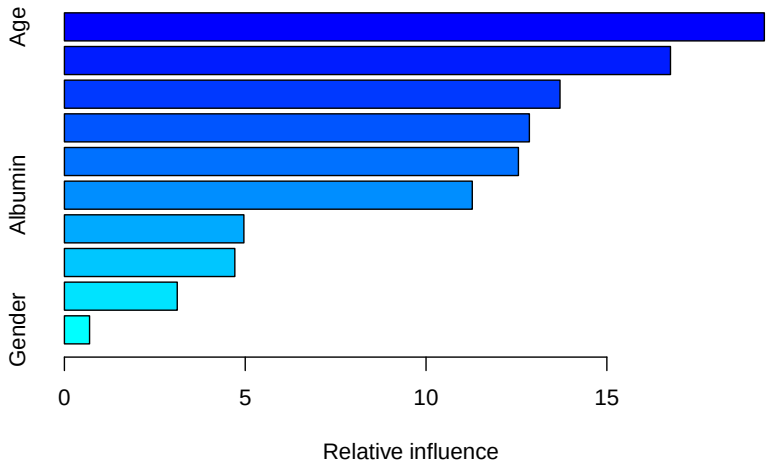

Figure 5: Relative influence of variables in boosting

Table 6: Relative influence of variables in boosting

	rel.inf
Age	19.35
Aspartate_Aminotransferase	16.76
Total_Protiens	13.70
Alkaline_Phosphotase	12.86
Alamine_Aminotransferase	12.55
Albumin	11.28
Albumin_and_Globulin_Ratio	4.96
Total_Bilirubin	4.71
Direct_Bilirubin	3.12
Gender	0.70

```
## [1] "Training error of boosting:  0.0201729106628242"
```

```
## [1] "Test error of boosting:  0.34051724137931"
```

Table 7: Confusion matrix for boosting

	Predicted 0	Predicted 1
0	21	49
1	30	132

Comparison between models

Methods	TrainError	TestError	Accuracy
Logistics regression	0.2507205	0.3103448	0.6896552
KNN	0.2190202	0.3534483	0.6465517
Decision Tree	0.2161383	0.3922414	0.6077586
Random Forest	0.0000000	0.3103448	0.6896552
Boosting	0.0201729	0.3405172	0.6594828

Discussion

By seeing the result of the models shown above, the model fitted by using **Boosting** algorithm has the smallest test error which is around 0.28, followed by the models fitted by using **Logistics regression** and **Random forest** algorithm which are both around 0.31. In our study, accuracy is measured by using **1 − test error**, so for **Boosting** model, the accuracy is around 0.72, and for **Logistic regression** and **Random Forest** models, the accuracy is around 0.69.

Briefly, **Boosting**, **Logistics regression**, and **Random Forest** are the best three models of these five models.

However, there are some limitations to our models. First of all, our dataset is imbalanced. Since there are 416 participants with disease and 167 participants without disease in the dataset, the model may be biased towards the majority class, resulting in poor performance on the minority class. That's why our models predict better on participants with disease than participants without disease. and this can limit the usefulness of the model in real-world applications. Also, the model accuracy is not as good as expected. The accuracy of the **Decision Tree** model is around 0.61, however, if we predict all the participants in the dataset had disease, the accuracy should be around 0.71, so the accuracy of the **Decision Tree** model is even lower than if all the participants are predicted with disease.

For future analysis, since the test error of the **Boosting**, **Logistical regression**, **Random Forest** models are the smallest of these five models, we will mainly focus on these models to do evaluation. In our study, we have 11 variables, including the response variable `"Dataset"`, so there are only 10 predictor variables, which is somehow less. We included all these variables into the models, and we will consider adding the products of the predictor variables to fix new models and measure their test error and accuracy, in order to find out if these new models will be better or not. Finally, we are going to select variables based on the variable importance outputted by **Boosting**, and then use these selected variables to fit new models. We believe that the new models should perform better.