HCV Disease Prediction

Aishwarya Sen, Aishwarya Patange, Manasi Khandekar, Takumi Hayashi

UNI: as6718, aap2239, mk4679, th3000

Columbia University

December 22, 2022

Abstract

Over the past few years, statistical learning has had a significant impact on the medical community. It has provided healthcare workers with new tools for analyzing and interpreting data to make more informed decisions about patient care. It has also aided us by identifying patterns and relationships in the data that may not be immediately evident to humans.

In this paper we have tried to predict if a certain individual has, Hepatitis C, a viral infection of the liver that can lead to serious complications if left untreated. The dataset for this task was obtained from the UCI Database and contains the information for over 600 participants. Using attributes such as participants' age, gender and liver function test results, we aim at building several machine learning models that can accurately detect the presence of Hepatitis as well as classify between the various stages of the disease - Hepatitis, Cirrhosis and Fibrosis. In addition to replicating the results of some papers utilizing this dataset, we experiment with some well known ML models and obtain substantial results which can be utlilized for the timely diagnosis and treatment of Hepatitis C.

Keywords: Hepatitis; Cirrhosis; Fibrosis, Statistical Learning, Decision Tree, Random Forest

1 Introduction

Hepatitis C is a viral disease caused by the Hepatitis C Virus (HCV) which targets the liver of the infected person. It is hard to trace as its incubation period ranges from 2 weeks to 6 months and approximately 80% of patients exhibit mild to no symptoms. Its common symptoms include loss of appetite, abdominal pain and fatigue. It is primarily spread through contact with blood of an infected person. This can occur through various mediums including sharing of infected needles or other equipment, through birth to an infected mother and sometimes even through sharing of personal hygiene items such as razors or toothbrushes. The duration of this disease ranges from a few weeks to even chronic life-threatening liver failure and accompanying illnesses. For some, Hepatitis-C manifests as jaundice-like symptoms. Often the appearance of symptoms indicate advanvced liver disease. If left untreated, Hepatis-C often develops into afflictions such as Cirrhosis and Fibrosis, late stages of Hepatisis C. Chronic stages of Hepatitis C simulates hepatic inflammation which later develops into liver Fibrosis. This can further develop into Cirrhosis, which is a late stage of liver scarring or Fibrosis that can be causes by either Hepatitis or by chronic alcoholism.

The absence of a medically attested vaccination for Hepatitis C, furthers the need for appropriate preventive measures to be taken as this disease is quite widespread. Behaviours promoting its spread such as usage of non-sterile injections for blood transfusion or intake of drugs should be strictly prohibited. Once infected, patients must seek immediate medical assistance as immediate action can cure the disease in around 8 to 12 weeks.

Hence, an efficient and accurate predictive model for the detection of hepatitis C and the classification of its various stages is essential and would enable the infected to avail timely treatment. In this project, we have utilized R in order to replicate the work done in a few existing literatures and created an extensive pipeline using R, including exploratory data analysis, data preprocessing, modeling and explainable AI in an attempt to deal with the detection, classification and treatment of Hepatitis C.

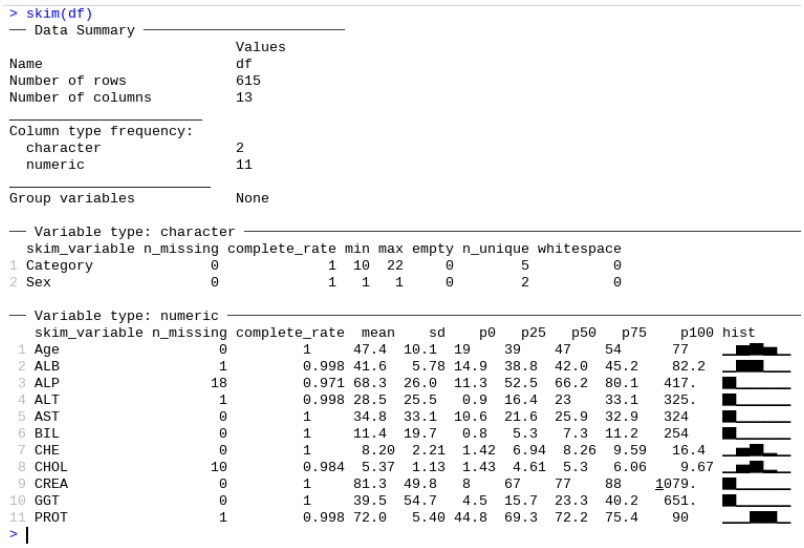
2 Datasets and Related Previous Work

2.1 Dataset Description

The dataset we have chosen - HCV Dataset, contains 615 observations comprising information of blood donors and Hepatitis C patients and is obtained from the UCI Machine Learning Repository Dua and Graff (2017). The features in the dataset are essential in detecting if a patient has hepatitis or not according the the domain experts. This information includes 14 attributes such as demographic data like age and gender and laboratory test results for various liver enzyme levels. The column Category or the diagnosis, is chosen as our target variable and is categorized as:

1. Blood Donor
2. Suspect Blood Donor
3. Hepatitis
4. Fibrosis
5. Cirrhosis

The first two categories are those of normal blood donors (not infected with Hepatitis) and suspected blood donors (those suspected to carry but not confirmed with Hepatitis). The latter three categories are those of the three stages of Hepatitis-C: Hepatitis (preliminary stages of Hepatitis), Fibrosis (advanced stages of Hepatitis) and finally Cirrhosis (chronic Fibrosis).



```
> skim(df)
── Data Summary ────────────────────────
                           Values
Name                       df
Number of rows             615
Number of columns          13
_______________________
Column type frequency:
  character                2
  numeric                  11
________________________
Group variables            None

── Variable type: character ──────────────────────────────────────────────────────
  skim_variable n_missing complete_rate min max empty n_unique whitespace
1 Category              0             1  10  22     0        5          0
2 Sex                   0             1   1   1     0        2          0

── Variable type: numeric ────────────────────────────────────────────────────────
   skim_variable n_missing complete_rate  mean    sd    p0   p25   p50   p75    p100 hist
 1 Age                   0         1     47.4  10.1  19    39    47    54      77   ▂▆▇▅▂
 2 ALB                   1         0.998 41.6   5.78 14.9  38.8  42.0  45.2    82.2 ▁▇▇▁▁
 3 ALP                  18         0.971 68.3  26.0  11.3  52.5  66.2  80.1   417.  ▇▂▁▁▁
 4 ALT                   1         0.998 28.5  25.5   0.9  16.4  23    33.1   325.  ▇▂▁▁▁
 5 AST                   0         1     34.8  33.1  10.6  21.6  25.9  32.9   324   ▇▂▁▁▁
 6 BIL                   0         1     11.4  19.7   0.8   5.3   7.3  11.2   254   ▇▁▁▁▁
 7 CHE                   0         1      8.20  2.21  1.42  6.94  8.26  9.59   16.4 ▁▃▇▃▁
 8 CHOL                 10         0.984  5.37  1.13  1.43  4.61  5.3   6.06    9.67 ▁▃▇▃▁
 9 CREA                  0         1     81.3  49.8   8    67    77    88    1079.  ▇▁▁▁▁
10 GGT                   0         1     39.5  54.7   4.5  15.7  23.3  40.2   651.  ▇▂▁▁▁
11 PROT                  1         0.998 72.0   5.40 44.8  69.3  72.2  75.4    90   ▁▂▅▇▃
>
```

Figure 1: Summary of the dataset using `Skimr`

Additionally, the column 'X' contains the patient ID or number but does not serve much purpose in our analysis and discarded during dataset pre-processing. The column 'age' contains the corresponding age for each participant and these range from the youngest participant with an age of 19 to the oldest ones with an age of 77 years. The mean age was 44 in this study. The column 'Sex' contains the gender of each participant i.e. male or female. All the attributes except 'sex' and 'Category' are numeric in nature. Figure 1 provides a staistical summary of the dataset using Skimr.

The remaining columns contain data on liver function test results for each donor and patient.

- ALB: Amount of albumin in participant's blood
- ALP: Amount of alkaline phosphatase in participant's blood
- ALT: Amount of alanine transaminase in participant's blood
- AST: Amount of aspartate aminotransferase in participant's blood
- BIL: Amount of bilirubin in participant's blood
- CHE: Amount of cholinesterase in participant's blood
- CHOL: Amount of cholesterol in participant's blood
- CREA: Amount of creatinine in participant's blood
- GGT: Amount of gamma-glutamyl transferase in participant's blood

- PROT: Amount of protein in participant's blood

The above tests help monitor the health of the liver by measuring the amounts of certain enzymes and proteins in the blood.
Thus, all these attributes assist in the accurate diagnosis of the presence of Hepatitis C as well as the classification of its various stages.

2.2 Previous Related Works

Hepatitis-C, its various stages of progression, measures taken to prevent it and treatment procedures is an area that has attracted the attention of researchers all over the world. Over the span of several decades, scientists and medical researchers alike have conducted thorough research on this topic, and this has yielded novel methods and breakthroughs which have been documented in the form of research papers. In this section, a few significant papers have been analysed.

Lichtinghagen et al. (2013) studied the significance of Enhanced Liver Fibrosis (ELF) score in the analysis of fibrosis stages in chronic liver disease. The study showed that ELF scores were notably higher in men as compared to women. Other important findings included the high influence rate of age on ELF scores and the identification of 3 cut-off values for the impact of this score on fibrosis. Finally, they conclude that ELF scores can be used to predict moderate stages of fibrosis as well as cirrhosis, while taking into account factors like age and gender.
Hoffmann et al. (2018) utilized machine learning methods such as decision trees to create laboratory diagnostic pathways using the HCV dataset. They demonstrate how adding of measures like ELF scores can enhance the accuracy of classification achieved by the decision trees. Edeh et al. (2022) applies a variety of machine learning and ensemble techniques for the prediction of advanced liver fibrosis in Hepatitis C patients. An accuracy of 94.67% was achieved using the individual models which was increased to 95.59% using an ensemble of various methods.

2.2.1 Replication of Previous Work

In our work, while we have primarily focused on creating our own pipeline by experimenting with a variety of methods including data visualization, data preprocessing, model building and even explainable AI, we have made an attempt to replicate some of the existing literature. We have not carried out an extensive or exact replication of previous papers due to the unavailability

of papers using the exact same dataset or due to the usage of highly complex models. The methods we have replicated are given below: In an attempt to reproduce the method followed by Hoffmann et al. (2018), we have created decision trees for the diagnosis and classification of Hepatitis C. This is discussed in detail in section 3.3.1.

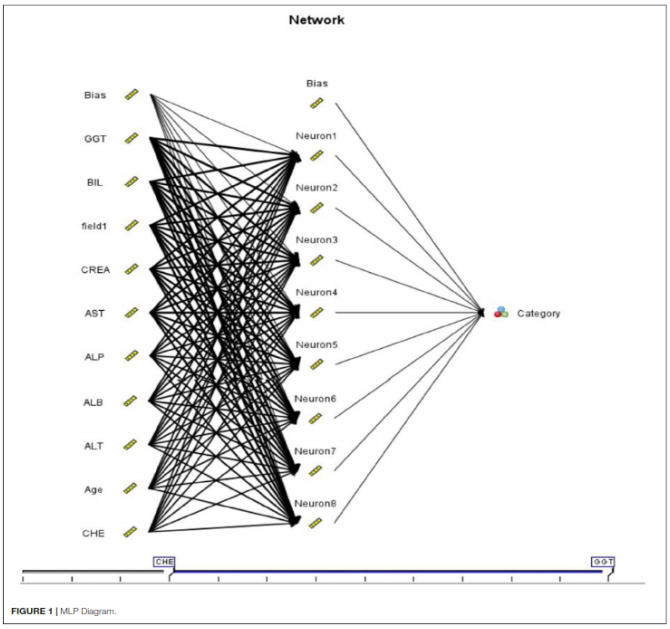


Figure 2: Architecture of MLP from Edeh et al. (2022)

Further, we have replicated the exact architecture of the Multilayer Perceptron (MLP) used by Edeh et al. (2022) and achieved a validation accuracy of 90.4%. We have used Python only for the replication of this model as the implementation on R was complex and resulted in the generation of errors. The architecture from Edeh et al. (2022) is given in Figure 2. We have tried to create a similar multi-layered perceptron in our project. The details are given in section 3.3.2.

3 Methodology of the Authors

An end-to-end pipeline has been created by us for the complete analysis and modeling of the problem statement. This pipeline consists of:

1. Exploratory Data Analysis: A comprehensive analysis using several data visualization techniques in R using the library ggplot and several plots including piechart, violin plot and correlation plot.

2. Data Preprocessing: Here, various steps are carried out in order to obtain a more reliable and clean dataset. These steps include missing value imputation, data standardization and outlier detection and removal.

3. Under modeling, we have attempted to replicate the models used in existing literature as well as carried out our own experimentation with several relevant machine learning models. These include decision tree, random forest and XGBoost.

4. Finally, we have chosen LIME or Local Interpretable Model-agnostic Explanations, a popular method used in explainable AI, in order to elucidate and increase the interpretability of our models and the results obtained.

3.1 Exploratory Data Analysis

Figure 3: Top left: Percentage distribution of HCV Diseases; Top right: Proportion of Gender affected by Different HCV Diseases; Bottom left: Distribution of HCV Diseases Vs Age Groups; Bottom Right: Correlation Plot for the Dataset

Exploratory data analysis is performed to visualize hidden pattern in the vast dataset using statistical graphing and other data visualization techniques. It helps in gathering of useful insights from the dataset which assists in further data processing steps such as preprocessing and feature engineering.

In our project the following analyses were performed in order to understand and obtain relations

between significant features in the dataset.

1. The first chart displays the distribution between the various classes in the target feature i.e. Category. We observe that 86.67% of the data is of uninfected blood donors, 4.88% of patients with Cirrhosis, 3.9% of patients with Hepatitis, 3.41% of those with Fibrosis and just 1.14% of data belonging to Suspect Blood Donors.

2. The second chart shows the distribution of males and females with respect to the various classes in the 'Category' variable. In our dataset, the proportion of males is higher to that of females and this is visualized here for each variable in the target feature as well. from this chart, we observe that a larger proportion of females are suffering from fibrosis.

3. The third chart displays the distribution of 'Category' with respect to the various age groups. In our dataset, there are a higher number of young participants suffering from Hepatitis as compared to the those not infected or those suffering from it's chronic stages.

4. Finally, we have the correlation plot which measures the strength of the relationship between the various variables in the dataset. The highest positive strength exists between ALB and PROT which signifies that patients having high amount of Albumin is their blood are also seen to have a high amount of protein in it.

3.2 Data Preprocessing

3.2.1 Missing Value Imputation

Missing value imputation is the process of replacing missing, null or corrupted values in a dataset with estimates or substituted values. This is an important step in the data preprocessing pipeline as many machine learning algorithms can not handle missing values and may throw errors or produce invalid results if null values are encountered in the data.

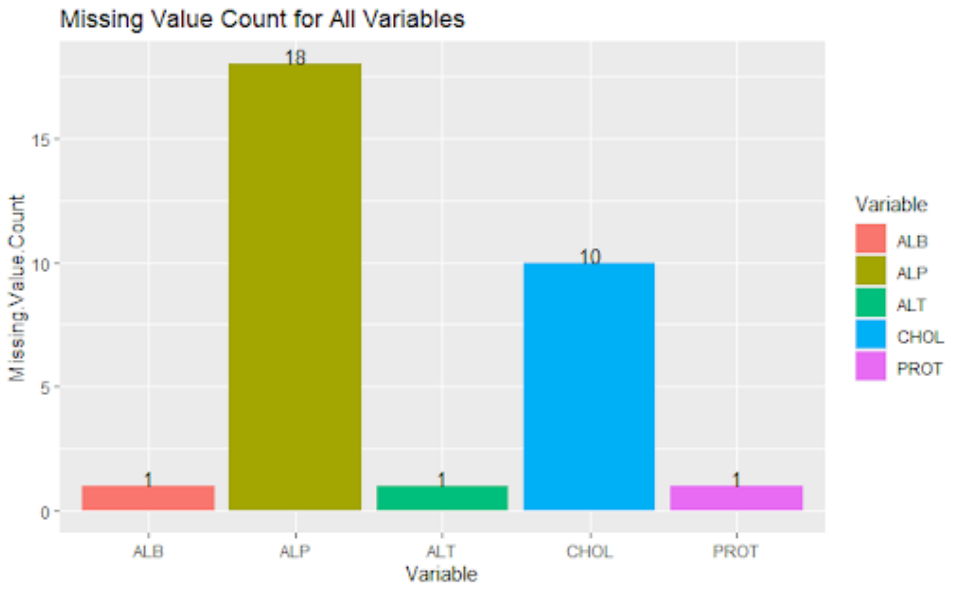


Figure 4: Number of Missing Values

As we can observe in 4, 5 features in our dataset contain missing or null values in their columns. The *ALP* variable contains the maximum number of missing values. Instead of dropping the columns Kwak and Kim (2017) we opted for imputing the missing values as we already had a smaller dataset size. For our usecase, we chose the median value imputation techinque.

We opted to go ahead with the median value imputation for the following two reasons:

- We observed that the some features of the data were heavily skewed or have outliers, and the median is less affected by these types of extreme values.
- In the previous papers Hoffmann et al. (2018), we have observed that decisions trees work better in general on hepatitis prediction datasets Dua and Graff (2017); Nasr et al. (2017). We knew the we won't be using standardization Patro and Sahu (2015) as a feature transformation for the features and median imputation works better when we do not standardize.

3.2.2 Detection and Removal Of Outliers

Outliers are data points that are significantly different from the majority of the data. They can have a negative impact on a model because they can skew the model's predictions. As a result, the model's predictions may not be representative of the majority of the data, and the model's performance may suffer.

Outliers can also cause problems when training machine learning models because they can cause the model to overfit as they can be very influential in determining the model's predictions, and if the outlier is not representative of the overall data, the model may not perform well on new data.

To remove outliers from the dataset, we use the inter-quantile range (IQR) method (Vinutha et al. (2018)). In this method we eliminate the datapoints which are above and below a certain upper and lower bound which is calculated based on the distribution of the variable. The are quantiles calculated from a sample of data, and they are typically denoted by Q_k, where k is the percentile. The sample quantile Q_k is defined as the value in the sample such that $k\%$ of the values in the sample are less than or equal to Q_k. We take the Q_{25} and the Q_{75} quantile values. Then we calculate the inter-quantile range given by:

$$IQR = Q_{75} - Q_{25}$$

Once we have the IQR value, we get the upper and the lower bounds by:

$$\text{Lower Bound} = Q_{25} - 1.5 \times IQR$$

$$\text{Upper Bound} = Q_{75} + 1.5 \times IQR$$

We then eliminate the datapoints which are present above and below these values.

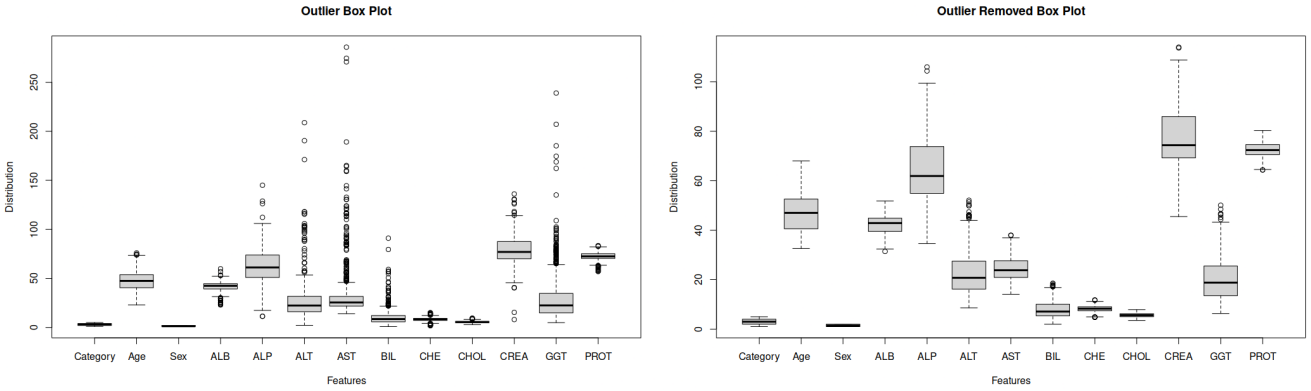


Figure 5: Left: Data before removal of outliers. Right: Data after removal of outliers using IQR technique.

One of the easiest ways to visualize outliers is by the use of boxplot. Figure 5 gives us the distribution of data before and after the removal of outliers using the inter-quantile range

technique.

3.2.3 Data Standardization

We standardization the data (except the response variable) using min-max scaling to scale the data in the range of -3 to 3. In the Figure 6, we can see the range of features after standardization.

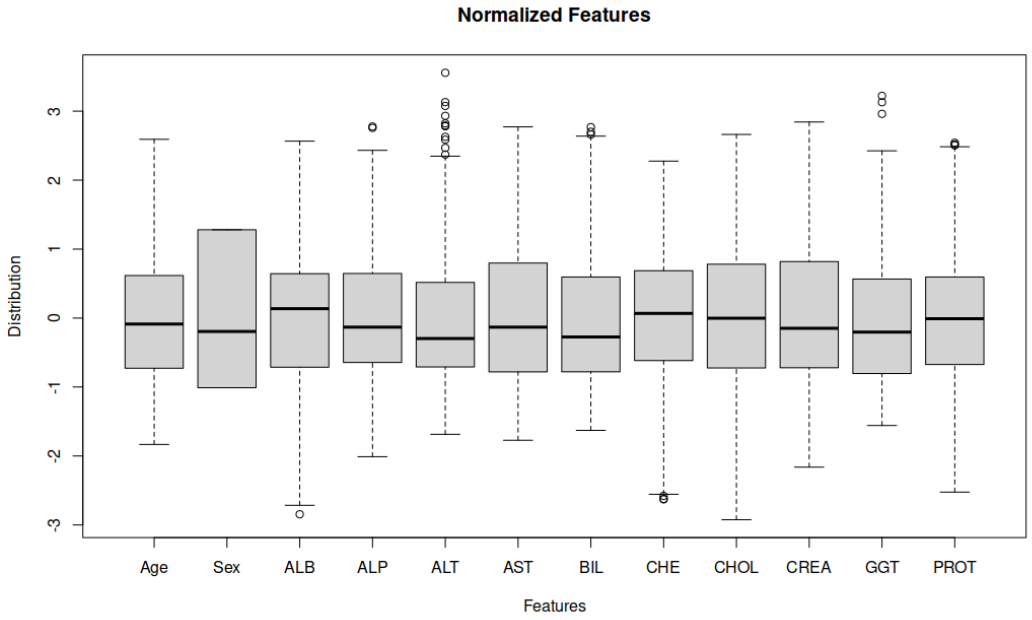


Figure 6: Features After Standardization

3.3 Modeling

We tried out 4 different models : Decision Tree, Multi-layered perceptron, Random Forest and XGBoost.

3.3.1 Decision Tree Classifier

In the decision trees model out of 13 variables available in the dataset, only 8 variables were used to create the decision tree. They are the actual levels of enzymes or proteins in the blood - ALB, ALP, ALT, AST, BIL, CHE, CHOL, CREA, GGT and PROT. The original depth of the tree was 12. The predictions and accuracy obtained on the training data and the test data were as follows:

```
> confusionMatrix(df_train$Category, model.pred_train)
Confusion Matrix and Statistics

          Reference
Prediction  1  2  3  4  5
         1 88  0  1  3  1
         2  3  0  0  1  1
         3  1  0 15  3  2
         4  0  0  6  9  1
         5  0  0  1  1 22

Overall Statistics

               Accuracy : 0.8428
                 95% CI : (0.7767, 0.8956)
    No Information Rate : 0.5786
    P-Value [Acc > NIR] : 7.615e-13

                  Kappa : 0.7406

 Mcnemar's Test P-Value : NA

Statistics by Class:

                     Class: 1 Class: 2 Class: 3 Class: 4 Class: 5
Sensitivity            0.9565       NA  0.65217   0.5294   0.8148
Specificity            0.9254  0.96855  0.95588   0.9507   0.9848
Pos Pred Value         0.9462       NA  0.71429   0.5625   0.9167
Neg Pred Value         0.9394       NA  0.94203   0.9441   0.9630
Prevalence             0.5786  0.00000  0.14465   0.1069   0.1698
Detection Rate         0.5535  0.00000  0.09434   0.0566   0.1384
Detection Prevalence   0.5849  0.03145  0.13208   0.1006   0.1509
Balanced Accuracy      0.9409       NA  0.80403   0.7401   0.8998
```

```
> confusionMatrix(df_test$Category, model.pred)
Confusion Matrix and Statistics

          Reference
Prediction  1  2  3  4  5
         1 24  0  4  1  1
         2  0  0  0  0  2
         3  0  0  0  2  1
         4  0  0  1  1  3
         5  0  0  0  0  6

Overall Statistics

               Accuracy : 0.6739
                 95% CI : (0.5198, 0.8047)
    No Information Rate : 0.5217
    P-Value [Acc > NIR] : 0.02655

                  Kappa : 0.4622

 Mcnemar's Test P-Value : NA

Statistics by Class:

                     Class: 1 Class: 2 Class: 3 Class: 4 Class: 5
Sensitivity            1.0000       NA  0.00000  0.25000   0.4615
Specificity            0.7273  0.95652  0.92683  0.90476   1.0000
Pos Pred Value         0.8000       NA  0.00000  0.20000   1.0000
Neg Pred Value         1.0000       NA  0.88372  0.92683   0.8250
Prevalence             0.5217  0.00000  0.10870  0.08696   0.2826
Detection Rate         0.5217  0.00000  0.00000  0.02174   0.1304
Detection Prevalence   0.6522  0.04348  0.06522  0.10870   0.1304
Balanced Accuracy      0.8636       NA  0.46341  0.57738   0.7308
```

Figure 7: Summary of Decision Tree Model

As can be seen from the figures, the training accuracy is significantly higher than the testing accuracy. This means that the model is overfitting. To overcome this, we pruned the tree. Pruning essentially reduces the tree size by removing the parts of the tree that do not contribute to the decision making. Using cross validation technique we found out the optimal tree size to be 5.

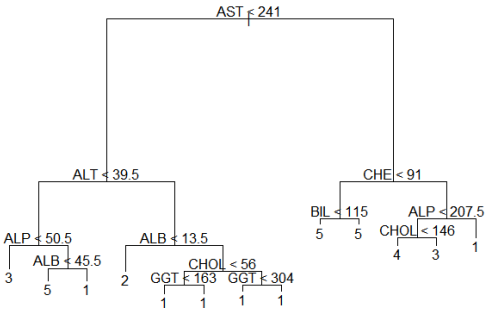


Figure 8: Result of Decision Tree

The overfitting is reduced but as the tree depth decreases the misclassification error rate increases. This is expected to happen since pruning removes certain branches of the tree due to which the tree size decreases.

Table 1: Table of Misclassification Error Rate of Unpruned Tree and Pruned Tree

	Unpruned Tree	Pruned Tree
Misclassification Error Rate	0.04989	0.07592

3.3.2 Multi-Layered Perceptron

As mentioned in section 2.2.1, this model was directly reproduced by us from Edeh et al. (2022). To implement this, we used the `tensorflow-keras` API to create the MLP. It consisted of a input layer, a hidden layer with 8 nodes, and an output layer with 5 nodes. For the data preprocessing of the data for this experiment, we first `OneHotEncoded` the categorical variables (Sex). Then we standardized the numerical variables and also `OneHotEncoded` the target variable as it contained more than 2 classes.

For the optimizer we used the `Adam` optimizer with a learning rate of 0.0001. The loss used was `CategoricalCrossentropy` and the metric we used to evaluate the model was accuracy (as used in the original paper). After training the model for 20 epochs on the dataset, we got a validation accuracy of 90.24%. The loss and accuracy curves for the model are given in Figure 9

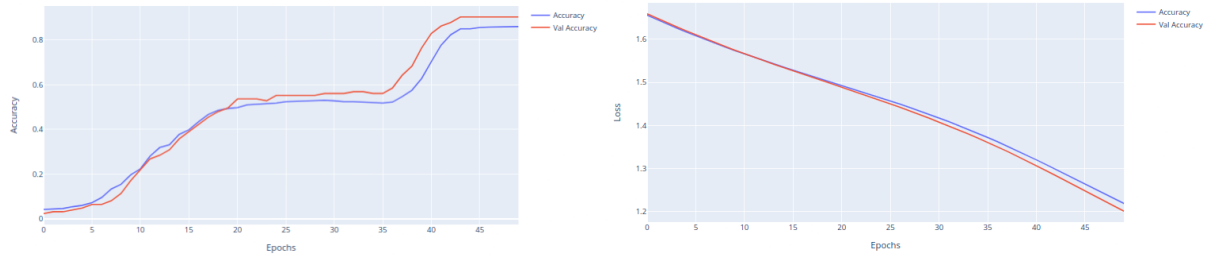


Figure 9: Left: Accuracy Curve of the MLP, Right: Loss Curve of the MLP

3.3.3 Random Forest Classifier

Since the decision tree model performed well on the dataset, we tried the random forest model next. The random forest model or random decision forests is an ensemble method that can be used for classification tasks. It creates a large number of decision trees while training and then outputs the class selected by the most number of trees. For this we used the randomForest library in R. We carried out a random 3 fold cross validation resampling for optimizing the accuracy of the model. A total of 50 trees were used and 2 variables were randomly sampled as candidates at each split. We got an accuracy of 93.5 on the test data using this random forest

model. The model summary obtained was as follows:

```
> confusionMatrix(predictions2, truth_num_fac)
Confusion Matrix and Statistics

          Reference
Prediction   1   2   3   4   5
         1 134   0   4   2   0
         2   0   0   0   0   0
         3   0   0   3   0   0
         4   0   0   1   2   0
         5   0   1   1   1   5

Overall Statistics

               Accuracy : 0.9351
                 95% CI : (0.8838, 0.9684)
    No Information Rate : 0.8701
    P-Value [Acc > NIR] : 0.007322

                  Kappa : 0.684

 Mcnemar's Test P-Value : NA

Statistics by Class:

                     Class: 1 Class: 2 Class: 3 Class: 4 Class: 5
Sensitivity            1.0000 0.000000  0.33333  0.40000  1.00000
Specificity            0.7000 1.000000  1.00000  0.99329  0.97987
Pos Pred Value         0.9571      NaN  1.00000  0.66667  0.62500
Neg Pred Value         1.0000 0.993506  0.96026  0.98013  1.00000
Prevalence             0.8701 0.006494  0.05844  0.03247  0.03247
Detection Rate         0.8701 0.000000  0.01948  0.01299  0.03247
Detection Prevalence   0.9091 0.000000  0.01948  0.01948  0.05195
Balanced Accuracy      0.8500 0.500000  0.66667  0.69664  0.98993
```

```
> model_rf
Random Forest

461 samples
 12 predictor
  5 classes: '1', '2', '3', '4', '5'

No pre-processing
Resampling: Cross-Validated (3 fold)
Summary of sample sizes: 308, 306, 308
Resampling results across tuning parameters:

  mtry  Accuracy   Kappa
  1     0.9045892  0.4648084
  2     0.9132476  0.5711537
  9     0.9132476  0.6254233

Accuracy was used to select the optimal model using the largest value.
The final value used for the model was mtry = 2.
```

Figure 10: Summary of Random Forest

We also plotted the variable importance plot for the trained model.

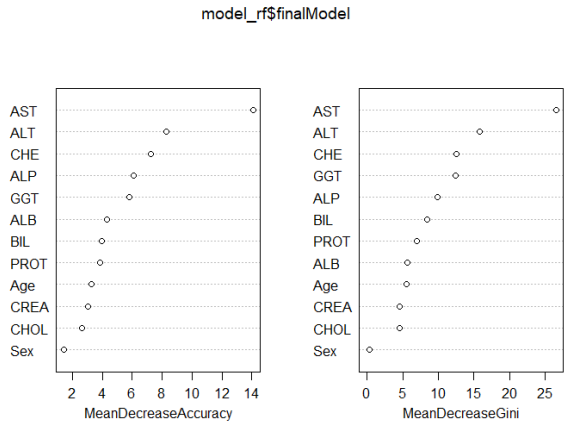


Figure 11: Variable Importance of Random Forest Model

The Mean Decrease Accuracy plot describes by excluding a variable each time, how much accuracy the model will lose. If a variable is necessary for a correct classification then the accuracy will suffer a lot if that variable is excluded from the model. The plot has the variables ordered in decreasing level of importance. The mean decrease in Gini gives a measure of much each variable contributes to the split of nodes in the random forest. A higher value of mean decrease in Gini, indicates a higher importance of that variable. From the plot it can be seen that AST, ALT, CHE are the 3 most important variables while Sex of the individual is the least important variable to make the prediction. We then trained the random forest model using only the 3 most important variables and got the following results:

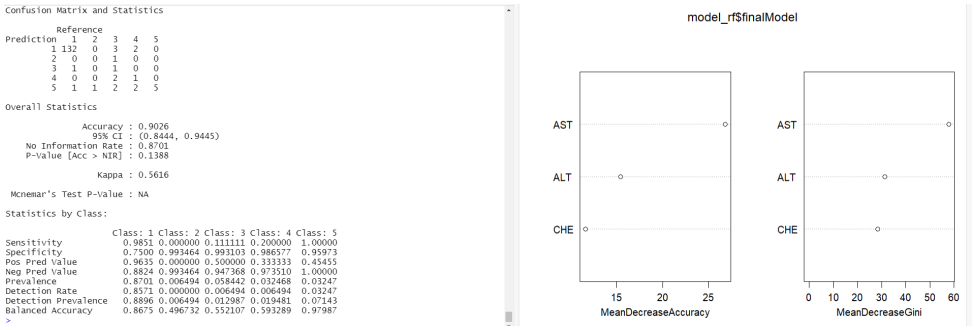


Figure 12: Random Forest model trained on the 3 most important features

3.3.4 XGBoost Classifier

XGBoost is an implementation of gradient boosted decision trees which has an improved speed and performance. It belongs to the boosting algorithms family and is an ensemble decision tree method. Boosting is essentially a sequential process where trees are grown depending on and using information from the previously grown trees. It helps to convert the weak learners that is a learner which is just slighltly better than random guessing into a strong learner. XGBoost belongs to a family of boosting algorithms that convert weak learners into strong learners. A weak learner is one which is slightly better than random guessing. To solve our multiclass classification problem we used gbtree as the booster parameter, so that a tree is grown one after other and attempts to reduce misclassification rate in subsequent iterations. Gamma is value for level regularization to be used, i.e how much should the large coefficients that do not improve the model performance be penalized. Higher the value, higher is the regularization. Max depth parameter controls the depth of the tree, larger the depth more complex is the model. But it also increases the chances of overfitting. There is no standard value for max depth, we used a value of 6 in our model.

```
Confusion Matrix and Statistics

          Reference
Prediction   1   2   3   4   5
         1 134   0   4   2   0
         2   0   0   0   0   0
         3   0   0   3   1   0
         4   0   0   1   1   0
         5   0   1   1   1   5

Overall Statistics

               Accuracy : 0.9286
                 95% CI : (0.8758, 0.9638)
    No Information Rate : 0.8701
    P-Value [Acc > NIR] : 0.01549

                  Kappa : 0.6522

 Mcnemar's Test P-Value : NA

Statistics by Class:

                     Class: 1 Class: 2 Class: 3 Class: 4 Class: 5
Sensitivity            1.0000 0.000000  0.33333 0.200000  1.00000
Specificity            0.7000 1.000000  0.99310 0.993289  0.97987
Pos Pred Value         0.9571      NaN  0.75000 0.500000  0.62500
Neg Pred Value         1.0000 0.993506  0.96000 0.973684  1.00000
Prevalence             0.8701 0.006494  0.05844 0.032468  0.03247
Detection Rate         0.8701 0.000000  0.01948 0.006494  0.03247
Detection Prevalence   0.9091 0.000000  0.02597 0.012987  0.05195
Balanced Accuracy      0.8500 0.500000  0.66322 0.596644  0.98993
```

Figure 13: Summary of XGBoost Model

3.4 Local Interpretable Model-agnostic Explanations (LIME)

The use of local surrogate models to explain black box machine learning predictions is an interpretable technique. A concrete implementation of local surrogate models is proposed in a paper, Local interpretable model-agnostic explanations (LIME) Ribeiro et al. (2016). The surrogate models are trained to approximate the black box model predictions. To explain individual predictions, LIME trains local surrogate models instead of a global surrogate model.

Mathematically, LIME can be expressed as follows,

$$\text{explanation}(x) = \texttt{argmin}_g L(f, g, \pi_x) + \Omega(g)$$

When calculating x, a linear regression model is used to reduce loss L (e.g. mean squared error) at the same time as minimizing model complexity. This metric indicates the degree to which the explanation is close to the original model (such as the linear regression model XGBoost), that is, the linear regression model, $\Omega(g)$. As part of LIME, it is only in practice that the loss part is optimized. G describes all possible explanations. In order to explain an instance x, we use the proximity measure. When creating a linear regression model, it is the user's responsibility to decide how many features to use at one time in the model.

Figure 14 and 15, below show the results of the LIME explanation. A random sample of 10

local regions is used to explain each case. Each section explains whether the variable increases the probability of the variable supporting the local region or decreases the probability that it contradicts it. Based on the Explanation fit, we can determine whether the model adequately explains the local region. The results show that case 4 has the highest likelihood of being attributable out of the 4 observations, with two variables contributing to the high probability, including CHOL and BIL.

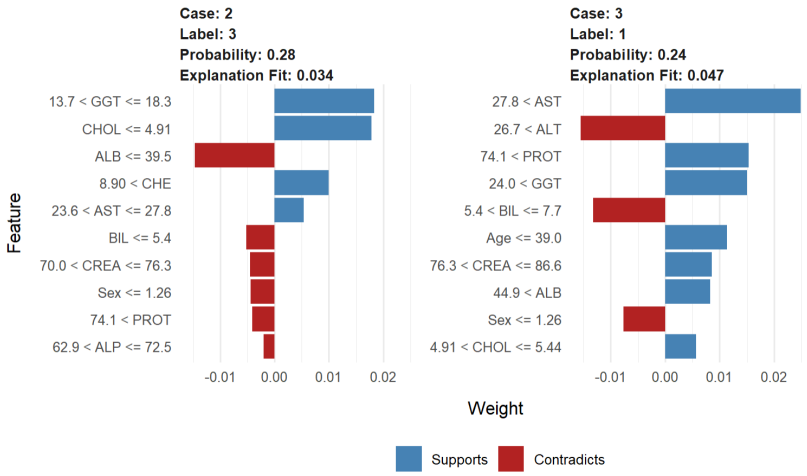


Figure 14: LIME result: Case 2 and 3

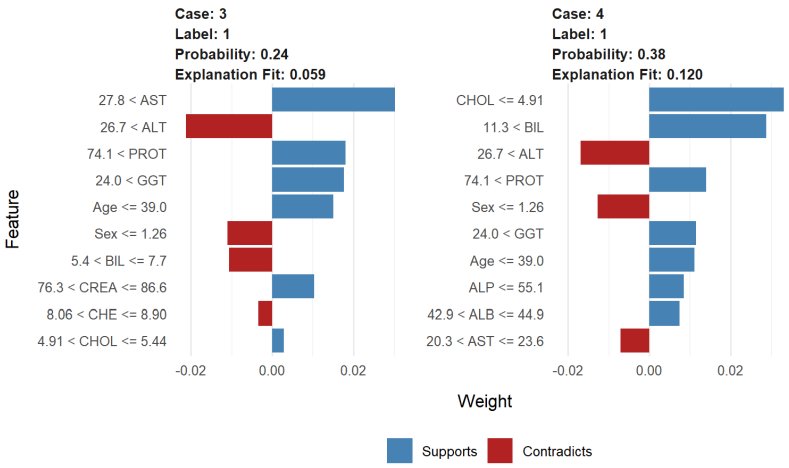


Figure 15: LIME result: Case 3 and 4

4 Results and Discussion

As can be seen from the table below, the accuracies of three types of models (Decision Trees, Random Forest, XGboost) were compared. According to the results of the comparison, the following can be said the Random forest was the most accurate model, 93.51%. As compared to the other two models, the Decision Tree was the least accurate, 67.39%. Decision trees with a large number of splits are more likely to encounter this problem due to over fitting, which contributes to its occurrence.

The random forest classifier performs the best, since instead of relying on a single decision tree, it gives the final prediction based on a majority of votes. The task that we wanted to accomplish was a multiclass classification problem and random forest model does well for such tasks and fits the data with less bias. It also does not require any feature selection as it does not use features which are not useful to split the data.

Table 2: Table of Comparison with 4 Models: Decision tree, Multi-layered Perceptron, Random Forest, and XGBoost

Model	**Accuracy (%)**
Decision Trees	67.39
Multi-Layered Perceptron	90.24
Random Forest	93.51
XGBoost	92.86

5 Conclusion

The implications of the severity of Hepatitis C can be clearly gauged by the sheer widespread and chronic nature of this disease. Thus, its timely diagnosis and treatment is one of the most relevant problems in the medical field. In an attempt to abet in the alleviation of this issue, we propose a pipeline which, using the HCV dataset from UCI database, presents a thorough analysis of its various attributes and machine learning models that help in the prediction of the presence as well as classification between the stages of chronic Hepatitis. Among our models, Random Forest is able to achieve best results with a high accuracy of 93.51%.

References

Dua, D. and Graff, C. (2017). UCI machine learning repository.

Edeh, M. O., Dalal, S., Dhaou, I. B., Agubosim, C. C., Umoke, C. C., Richard-Nnabu, N. E., and Dahiya, N. (2022). Artificial intelligence-based ensemble learning model for prediction of hepatitis c disease. *Frontiers in Public Health*, 10.

Hoffmann, G., Bietenbeck, A., Lichtinghagen, R., and Klawonn, F. (2018). Using machine learning techniques to generate laboratory diagnostic pathways—a case study. *Journal of Laboratory and Precision Medicine*, 3(6).

Kwak, S. and Kim, J. (2017). Statistical data preparation: Management of missing values and outliers. *Korean Journal of Anesthesiology*, 70:407.

Lichtinghagen, R., Pietsch, D., Bantel, H., Manns, M. P., Brand, K., and Bahr, M. J. (2013). The enhanced liver fibrosis (elf) score: Normal values, influence factors and proposed cut-off values. *Journal of Hepatology*, 59(2):236–242.

Nasr, M., El-Bahnasy, K., Hamdy, M., and Kamal, S. M. (2017). A novel model based on non invasive methods for prediction of liver fibrosis. In *2017 13th International Computer Engineering Conference (ICENCO)*, pages 276–281.

Patro, S. G. K. and Sahu, K. K. (2015). Normalization: A preprocessing stage. *CoRR*, abs/1503.06462.

Ribeiro, M. T., Singh, S., and Guestrin, C. (2016). "why should I trust you?": Explaining the predictions of any classifier. *CoRR*, abs/1602.04938.

Vinutha, H. P., Poornima, B., and Sagar, B. M. (2018). Detection of outliers using interquartile range technique from intrusion dataset. In Satapathy, S. C., Tavares, J. M. R., Bhateja, V., and Mohanty, J. R., editors, *Information and Decision Sciences*, pages 511–518, Singapore. Springer Singapore.