# Preventing the Spread of Hepatitis C Through Blood Transfusions

By Silver Monkeys:
Khush Garg
Rithvik Ponugoti
Tino Thoon
Varun Rege

Submitted to
Instructional Staff of IE 49000

And

**Prof. Ana María Estrada Gómez**

**Final Report**

In Partial Fulfillment of the Requirements for
IE 49000- Introduction to Machine Learning and its Applications

Purdue University
School of Industrial Engineering
West Lafayette, IN 47906

May 6, 2022

# Table of Contents

# Problem Statement

Hepatitis C is a virus-borne infection that targets the liver and causes inflammation. Contact with contaminated blood spreads this infection. Hepatitis C is difficult to identify because most people with Hepatitis C don't usually experience symptoms but those who do develop symptoms experience fatigue, nausea, loss of appetite, and yellowing of the eyes and skin. Because most people don't experience symptoms, it is crucial to identify and detect Hepatitis C in blood donations. According to the Mayo Clinic, blood donors usually donate a half liter of blood and in order to be an eligible blood donor one must be in good health, at least 16 years of age, at least 50 kilograms, and able to successfully pass the physical and health-history screening assessments.

The main goal of this project is to distinguish between healthy blood donors and Hepatitis C suspect blood donors. This differentiation is extremely important to prevent the spread of diseases such as Hepatitis C for those in need of blood transfusions. This goal can be broken down into more specific goals as follows:

1. *Accurately identify healthy blood donors (0) from suspect blood donors (0s) solely by analyzing the given attributes in the blood test results*
2. *Summarize any trends seen amongst suspect blood donors and their respective ages or sex*
3. *Minimize the misclassification rate*

The specific success metric to be used to determine whether the goals are reached is by analyzing the confusion matrix. This can be used to calculate values such as accuracy and precision. From this, we can calculate the misclassification rate and a lower rate is ideal to attain the goals of this project. Another metric our team heavily considered is maximizing the trade-off between number of predictors used and the resulting accuracy in the final model.

This project will analyze the HCV data Data Set that contains laboratory values of blood donors and Hepatitis C patients along with their respective demographics. Our solution includes a machine learning model that can accurately predict whether a blood donor is diagnosed with Hepatitis C, by analyzing the blood component measurements. According to the U.S. Department of Health & Human Services there are an estimated 2.4 million people living with hepatitis C in the United States but the actual number may be as high as 4.7 million or as low as 2.5 million. As such, the main benefit of solving this problem is to potentially save several individuals from contracting Hepatitis C through blood transfusion.

# Data Background

The data we will be using in this project comes from the University of California Irvine Machine Learning Repository. The data consists of observations of blood test results from blood donors and hepatitis C patients. The dataset comes in a csv format and consists of 615 observations, each with 14 attributes. The attribute types include laboratory test values of the blood test results as well as the demographic information of the donor/patient in each observation such as age, gender, as well as their hepatitis status. The observations are not associated with time, therefore the data is discrete in nature. The response variable in our data is the "Category" attribute, in which donors/patients are classified into 0 for Blood Donor, 0s for suspect Blood Donors, 1 for Hepatitis, 2 for fibrosis, and 3 for Cirrhosis. Certain observations in this dataset contain N/A values for some of the attributes.

# Methods

## Baseline - Naïve Bayes

Naive Bayes is a type of supervised nonlinear classification technique which belongs to a family of probabilistic classifiers based on the application of Bayes Theorem. This classifier has strong independence assumptions between features in a dataset. The term "Naive" in the name comes from this characteristic, in which the algorithm assumes all feature occurrences are independent of one another. The basis of this technique is Bayes Theorem, $P(A|B) = \frac{P(B|A) * P(A)}{P(B)}$, gives the conditional probability of an event A, given another event B has occurred. In the case of our dataset, we can expand this theorem in order to support our 14 predictors.

In our project, we are employing the Naive Bayes classifier as a baseline model for our dataset, upon which we will compare the results of our subsequent methods in order to gauge improvement in predictive accuracy as well as overall fit. This model is the most appropriate to use as the baseline since it is fast, easy, performs well in binary as well as multi-class classifiers and is the most popular choice for text classification problems.

## Multiclass Logistic Regression

Logistic regression is defined as a process for modeling the probability of a discrete outcome given an input variable. The discrete outcome in the data is called the “Category” attribute wherein donors or patients are classified as healthy blood donors (0) from suspect blood donors (0s) solely by analyzing the given attributes in the blood test results.

To fit our model, the team first had to collapse factor levels into manually defined groups using the fct_collapse function from the forcats library. The team chose to pool all suspect blood donors regardless of their Hepatitis C progress. This would entail that "0s=suspect Blood Donor", "1=Hepatitis", "2=Fibrosis", "3=Cirrhosis" would all be classified as one factor level (2) and that all healthy blood donors are classified as another factor level (1). As a result, the team was able to fit the data to follow the binary logic of logistic regression.

Using the createDataPartition function from the caret package, the data was then split into training and testing samples with the ⅔’s of the data representing the training set and the remainder as the testing set. The team chose multiclass logistic regression as one of the methods because the goal is to classify if a patient has Hepatitis C based on their features.

## Random Forests

Random Forests is a technique part of a family of supervised learning methods known as ensemble learning, in which multiple algorithms are utilized in unison to obtain superior predictive performance than could be achieved by using any of the constituent learning algorithms individually. As the name suggests, this technique operates by constructing a multitude of decision trees during training. While this method is employed in the context of a classification problem in our project, it can also be used for regression tasks as well. The output of a classification approach to random forests comes in the form of the class or category which has been chosen by the most number of decision trees in the ensemble. Random forests solve a problem typical of traditional decision tree models in which models are overfitted to the datasets, resulting in low bias but high variance.

This method is particularly suited for our project and dataset as classification decision trees themselves have several advantages over other classification methods. The algorithm is able to handle both categorical and numerical data, training data does not need extensive preparation such as normalization, the method provides built-in feature selection, and performs well with large datasets.

# Results

## Baseline - Naïve Bayes

```
Naive Bayes Classifier for Discrete Predictors

Call:
naiveBayes.default(x = X, y = Y, laplace = laplace)

A-priori probabilities:
Y
    0=Blood Donor SuspectBloodDonor
        0.8581395         0.1418605

Conditional probabilities:
                    Age
Y                        [,1]      [,2]
  0=Blood Donor     46.54472  9.402751
  SuspectBloodDonor 49.03279 12.424394

                    Sex
Y                          f         m
  0=Blood Donor     0.4092141 0.5907859
  SuspectBloodDonor 0.2622951 0.7377049

                    ALB
Y                        [,1]     [,2]
  0=Blood Donor     42.06341 5.282841
  SuspectBloodDonor 36.70984 8.450951

                    ALP
Y                        [,1]     [,2]
  0=Blood Donor     67.63631 17.95408
  SuspectBloodDonor 72.66569 58.27544

                    ALT
Y                        [,1]     [,2]
  0=Blood Donor     26.75014 14.11922
  SuspectBloodDonor 39.83934 62.90028

                    AST
Y                        [,1]     [,2]
  0=Blood Donor     26.54282 11.51389
  SuspectBloodDonor 84.28689 50.47914

                    BIL
Y                        [,1]      [,2]
  0=Blood Donor      8.544444  6.454888
  SuspectBloodDonor 33.795082 54.386118

                    CHE
Y                        [,1]     [,2]
  0=Blood Donor     8.357913 1.816747
  SuspectBloodDonor 6.827213 3.617163

                    CHOL
Y                        [,1]     [,2]
  0=Blood Donor     5.503990 1.051969
  SuspectBloodDonor 4.571116 1.347657

                    CREA
Y                        [,1]      [,2]
  0=Blood Donor     78.93496  14.72413
  SuspectBloodDonor 99.85902 150.72510
```

Figure 1. Naive Bayes Classifier Model Summary

The team first partitioned the given dataset into training and testing datasets. When the team applied the Naive Bayes classifier model to the training dataset, we obtained the conditional probabilities matrices for each of the factors that also classified them into one of the two outcomes - Healthy blood donor and Suspect blood donor. This model is a probabilistic classifier, which means it predicts one of the two outcomes on the basis of the probability of the factors.
The team then decided to apply this model on our testing data and obtained a confusion matrix, from which we could calculate the accuracy and misclassification rate as summarized below. These values will be used as our baseline comparison model.

| Dataset | Confusion Matrix | Misclassification Rate |
|---|---|---|
| Training | nb.train.class 0=Blood Donor SuspectBloodDonor<br>0=Blood Donor 357 16<br>SuspectBloodDonor 12 45 | **0.06512** |
| Testing | nb.class 0=Blood Donor SuspectBloodDonor<br>0=Blood Donor 160 9<br>SuspectBloodDonor 4 12 | **0.07027** |

Table 1: NB Classifier model success metrics

# Multiclass Logistic Regression

```
Coefficients:
                    Values   Std. Err.
(Intercept)  1.4254128996 4.208501679
Age         -0.0329186712 0.030566530
Sexm        -0.6730437282 0.664934566
ALB         -0.2236487564 0.067887383
ALP         -0.0430752164 0.018385738
ALT         -0.0007218028 0.012568657
AST          0.0596236540 0.017878138
BIL          0.0704648466 0.033291520
CHE          0.1906473860 0.164001104
CHOL        -0.9403414956 0.310757967
CREA         0.0107162637 0.006671109
GGT          0.0442036745 0.009368383
PROT         0.1024769259 0.051891525

Residual Deviance: 97.63988
AIC: 123.6399
```

Figure 2: Multiclass Logistic Regression Model Summary

After running logistic regression on the training set the table above summarizes the results of our model on the training set. To make predictions and test the accuracy of our model the team ran the logistic regression model against the testing set.

Using the predict function and the forward pipe operator(%>%) we were able to determine the accuracy of our model: 95.54%.

The table below regards the confusion matrix and misclassification rate for both training and testing dataset respectively.

| Dataset | Confusion Matrix | Misclassification Rate |
|---|---|---|
| Training | pred<br>1 2<br>1 348 5<br>2 13 47 | **0.04556**<br><<br>0.06512 (Baseline) |
| Testing | pred<br>1 2<br>1 178 2<br>2 7 15 | **0.04663**<br><<br>0.07027 (Baseline) |

Table 2: Multiclass Logistic Regression model success metrics

# Random Forests

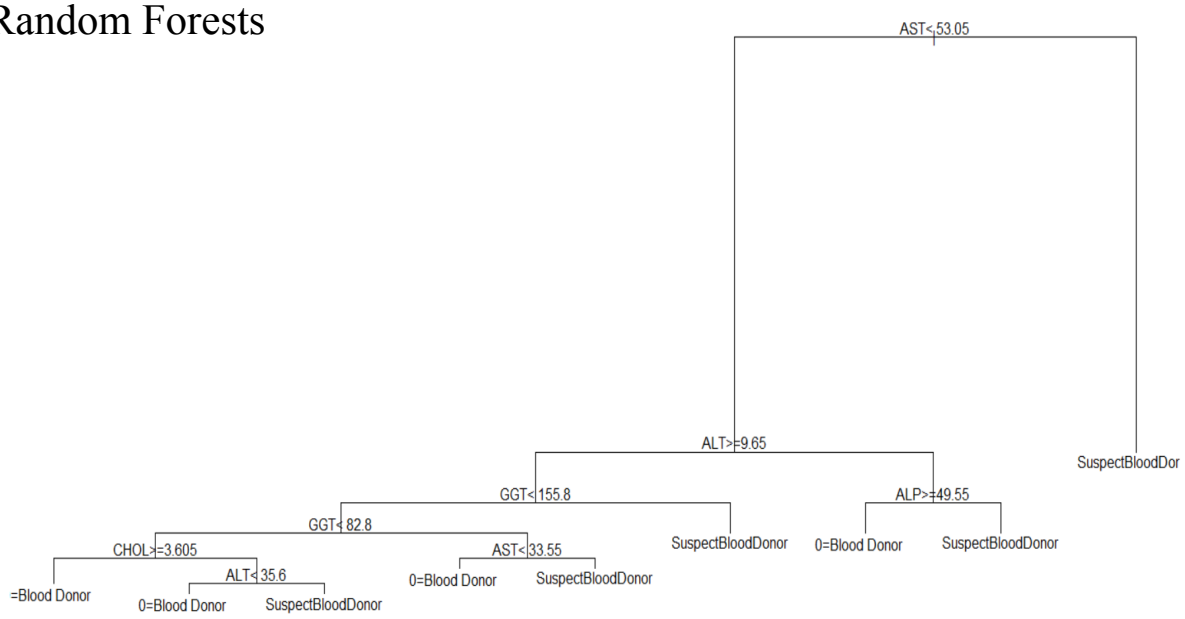


Figure 3: Random Forest Decision Tree

After the team fit a random forest to our training data, we had to initially choose the optimal number of variables to choose at any split. After running a loop and comparing the model performances for values from 1 to 10, the team found that the optimal number of variables to choose at any split for this model is 3. This value is also the rounded value of the square root of the number of predictors in our model. With our optimal model, the team observed an Out Of Bag (OOB) estimate of error rate equalling 3.72%. After the team was satisfied with the model's performance on the training dataset, we went ahead and used the predict function to apply the model to our testing dataset.

We observed a table displaying the importance of each predictor towards the model by analyzing mean decreasing accuracy and mean decreasing Gini value, as seen in Figure 4 and 5 in the Appendix. The table seen below summarizes the success metrics of our model compared to the baseline model. As seen in the confusion matrix for our testing dataset, out of the 164 total healthy blood donors, our model was able to correctly identify 162 of them. This model also had the lowest misclassification rate for the training dataset and hence the team decided to build upon this model to present as our final solution.

| Dataset | Confusion Matrix | Misclassification Rate |
|---|---|---|
| Training | 0=Blood Donor SuspectBloodDonor<br>0=Blood Donor 365 4<br>SuspectBloodDonor 12 49 | **0.03721** < 0.06512 (Baseline) |
| Testing | 0=Blood Donor SuspectBloodDonor<br>0=Blood Donor 162 2<br>SuspectBloodDonor 6 15 | **0.04324** < 0.07027 (Baseline) |

Table 3: Random Forests model success metrics

# Lessons Learned

- Before implementing machine learning, the team first had to truly understand the dataset which led us to further researching about Hepatitis C and its relation to blood transfusions

- We learnt that the current risk of getting Hepatitis C through blood transfusions is as low as 1 in every 125,000 units of blood samples, thanks to extensive selection and screening which were not prevalent before 1992

- Upon further research, the team learnt that few machine learning algorithms do exist but they identify Hepatitis C in health insurance claims data and not blood samples data

- While trying to apply various machine learning algorithms the team learnt to the given data, we realized that we are constrained by the data type and desired outcome

- The team's final model was built using random forests since it had the lowest misclassification rate while using the least number of predictors, in hopes of further eradicating the chances of contracting Hepatitis C through blood transfusions

- While working with the given data, the team explored on how to organize laboratory datasets and pre-process the given data accordingly

- In particular, the team struggled with cleaning the data as a lot of N/A values were present for several blood component measurements in the blood sample data

- Lastly, the team understood the importance of data analysis and the need for machine learning in the health industry through this project

- An area of improvement for the team's model is to further improve accuracy and to better deal with missing values rather than simply using the average value of that predictor