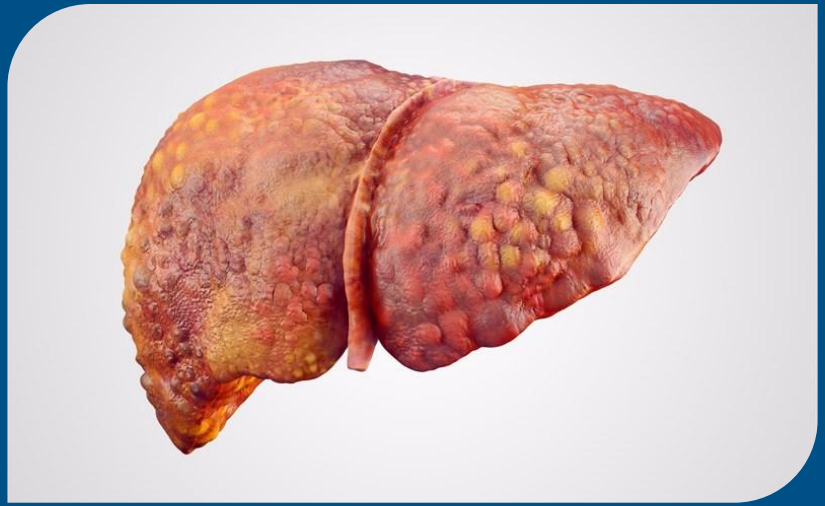
Cirrhosis Patient Survival Prediction

Author: Kevin Hart

Date Completed: 4/21/2025



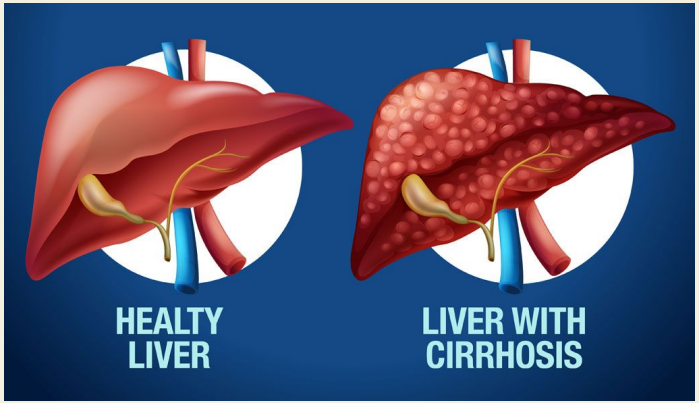
Abstract

This project explores the classification of patient status (death vs. censored) using the cirrhosis dataset. The dataset was cleaned, visualized, and prepared using standard preprocessing techniques, including the imputation of missing values and log-transformation of numeric features. Two machine learning models were applied: a decision tree using rpart() and a Naive Bayes classifier.
Cross-validation was conducted to assess generalizability. While cross-validation suggested a higher complexity parameter (cp = 0.0926) for the decision tree, a smaller cp value of 0.001 actually resulted in better performance on the test set (84.75% accuracy), indicating that the more detailed tree better captured patterns in this dataset. Naive Bayes achieved slightly lower performance (81.36% accuracy), suggesting both models are viable but decision trees may offer a slight edge in this context.

Introduction

Liver cirrhosis is a condition in which the liver is scarred, causing **permanent** damage. As scar tissue replaces healthy tissue, the liver is prevented from working properly and flow of blood is blocked. It is estimated that about **1 in 400** adults in the United States have cirrhosis.

Predicting patient outcomes is important for clinical planning. This project investigates whether clinical and laboratory measurements can predict survival status (death or censored) in patients with cirrhosis. These predictions can be used to inform treatment decisions, prioritize high-risk patients for interventions, and improve overall healthcare resource allocation.



Materials

Dataset from UCI Machine Learning Repository (Link)

- Includes 418 patient records that include 17 features
- Predictor variables include: age, drug, sex, ascites, hepatomegaly, spiders, edema, bilirubin, cholesterol, albumin, urine copper, alkaline phosphatase, SGOT, triglycerides, platelets, prothrombin
- “Status” variable is the outcome variable (“C” = Censored, “D” = Death, “CL” = Censored due to transplant)

Methods

Data Preprocessing:

- **ID** and **N_Days** non-feature columns were removed
- **Missing values** in numeric variables were imputed using the column mean
- The dataset was filtered to **exclude patients with liver transplants** (Status == 'CL') to focus on natural survival outcomes

Naive Bayes:

- Implemented using R's **naiveBayes()** and **predict()**,
- Estimates the prior probability of each status and the conditional probabilities (likelihoods) of each feature given each status

Recursive Partitioning:

- Implemented using R's **rpart(),** which builds a classification tree by repeatedly splitting the data based on the predictor that maximizes the reduction in impurity (measured using Gini index).
- Predictions were made using **predict()** on the test set.

Train-Test Split and Cross-Validation

- An **80/20** stratified train-test split was applied using **initial_split()** from the rsample package to ensure balanced class distribution.
- **10-fold cross-validation** was implemented using **train()** from the caret package
- Numeric features were **log-transformed** using **log1p()** to reduce variance and avoid **zero-probability** issues in Naive Bayes modeling

Model Tuning:

- For the decision tree, the **cp** (complexity parameter) was selected using cross-validation to balance tree complexity and performance
- For Naive Bayes, **kernel estimation** and **Laplace smoothing** (via fL and usekernel) were tuned using grid search to improve generalization and avoid zero-probability issues

Model Evaluation:

- Model performance was evaluated using confusion matrices, overall accuracy, sensitivity, specificity, and Cohen's Kappa, calculated with **confusionMatrix()**

Results!

Data Overview

First 6 Rows

	Status	Drug	Age	Sex	Ascites	Hepatomegaly	Spiders	Edema	Bilirubin	Cholesterol	Albumin
1	D	D-penicillamine	21464	F	Y	Y	Y	Y	14.5	261	2.60
2	C	D-penicillamine	20617	F	N	Y	Y	N	1.1	302	4.14
3	D	D-penicillamine	25594	M	N	N	N	S	1.4	176	3.48
4	D	D-penicillamine	19994	F	N	Y	Y	S	1.8	244	2.54
5	D	Placebo	24201	F	N	Y	N	N	0.8	248	3.98
6	C	Placebo	20284	F	N	Y	N	N	1.0	322	4.09

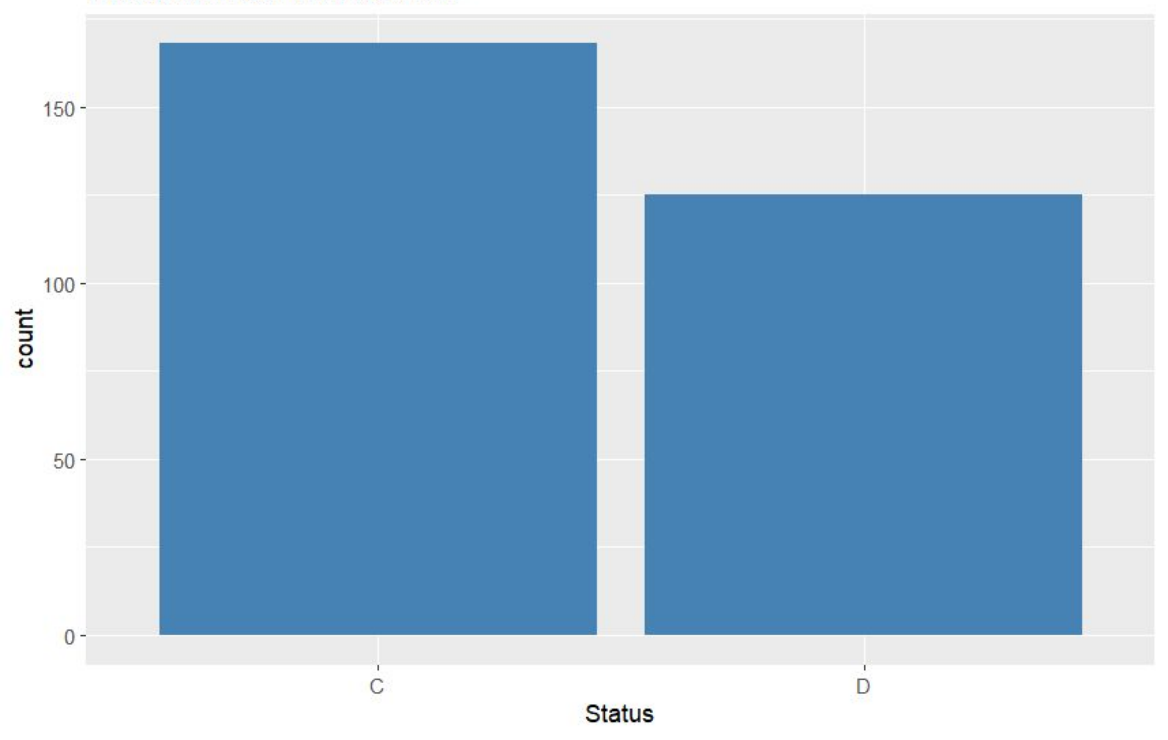
	Copper	Alk_Phos	SGOT	Tryglicerides	Platelets	Prothrombin	Stage
1	156	1718.0	137.95	172	190	12.2	4
2	54	7394.8	113.52	88	221	10.6	3
3	210	516.0	96.10	55	151	12.0	4
4	64	6121.8	60.63	92	183	10.3	4
5	50	944.0	93.00	63	NA	11.0	3
6	52	824.0	60.45	213	204	9.7	3

Basic Statistics

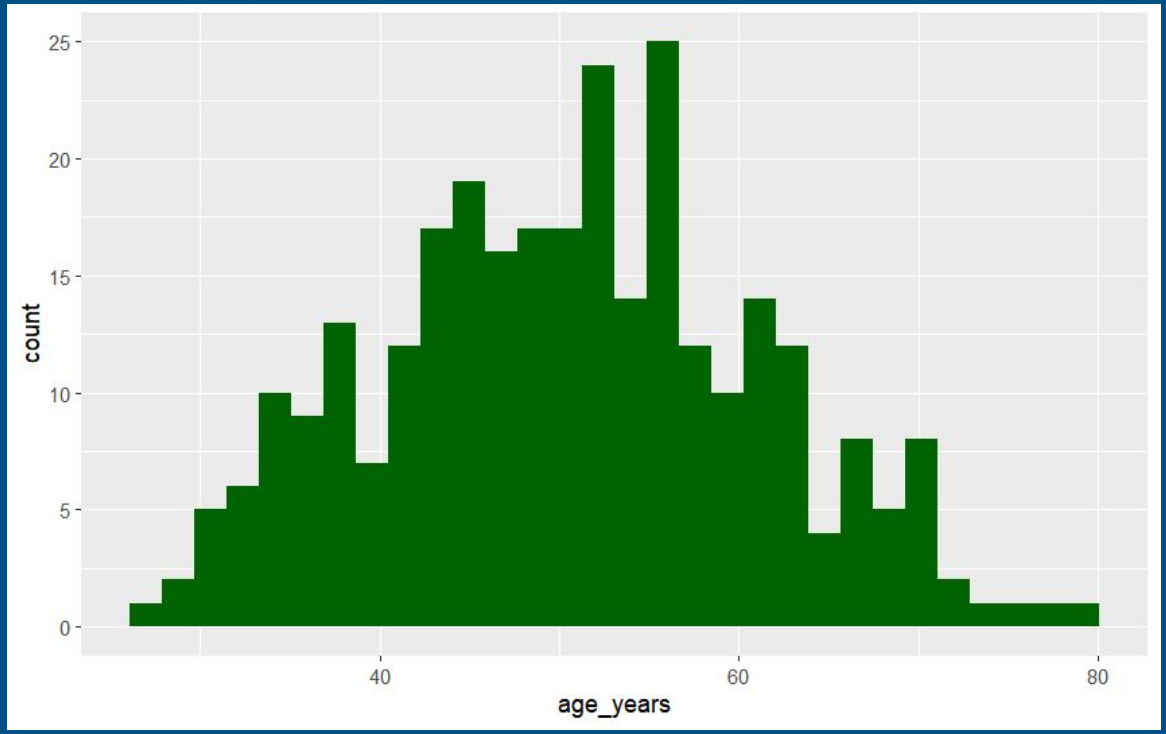
Bilirubin	Cholesterol	Albumin	Copper	Alk_Phos	SGOT
Min. : 0.300	Min. : 120.0	Min. :1.960	Min. : 4.00	Min. : 289	Min. : 26.35
1st Qu.: 0.800	1st Qu.: 248.0	1st Qu.:3.260	1st Qu.: 41.00	1st Qu.: 858	1st Qu.: 79.05
Median : 1.300	Median : 303.0	Median :3.530	Median : 70.00	Median : 1258	Median :111.00
Mean : 3.199	Mean : 364.8	Mean :3.498	Mean : 95.93	Mean : 2012	Mean :122.07
3rd Qu.: 3.300	3rd Qu.: 398.2	3rd Qu.:3.770	3rd Qu.:123.00	3rd Qu.: 2009	3rd Qu.:151.90
Max. :28.000	Max. :1775.0	Max. :4.640	Max. :588.00	Max. :13862	Max. :457.25
	NA's :127		NA's :102	NA's :100	NA's :100

Tryglicerides	Platelets	Prothrombin	Stage
Min. : 44.00	Min. : 62.0	Min. : 9.00	Min. :1.000
1st Qu.: 84.75	1st Qu.:182.2	1st Qu.:10.00	1st Qu.:2.000
Median :108.00	Median :247.0	Median :10.60	Median :3.000
Mean :124.07	Mean :253.6	Mean :10.76	Mean :3.013
3rd Qu.:151.25	3rd Qu.:313.5	3rd Qu.:11.10	3rd Qu.:4.000
Max. :598.00	Max. :721.0	Max. :18.00	Max. :4.000
NA's :129	NA's :11	NA's :2	NA's :6

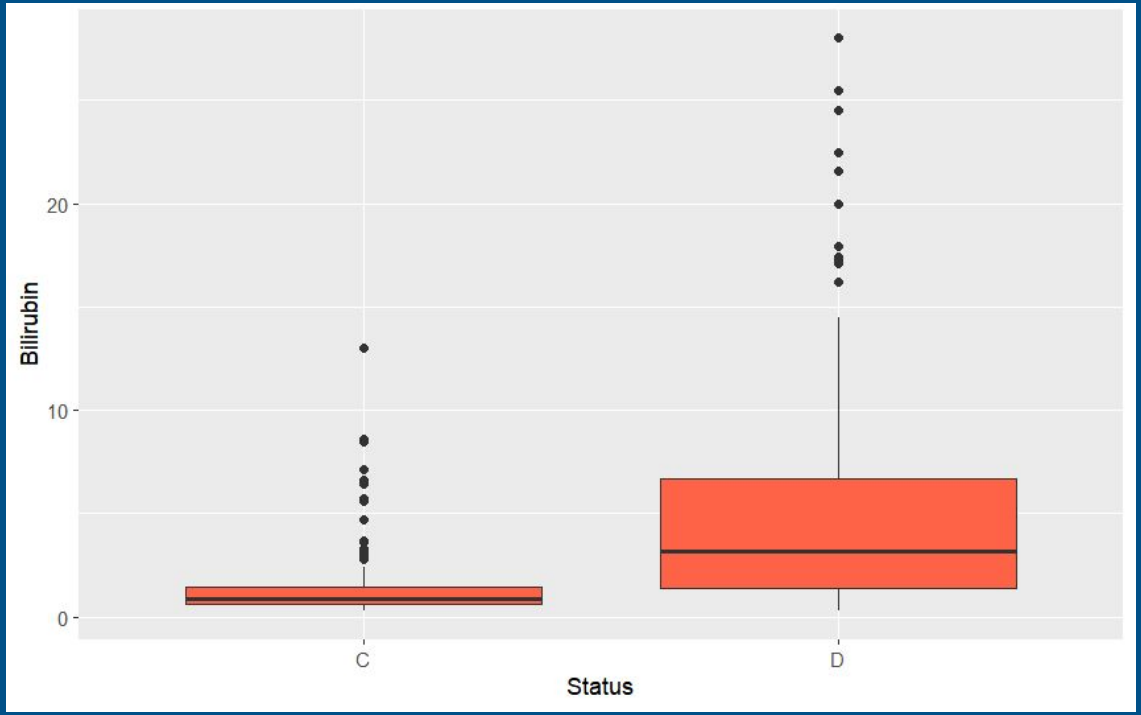
Distribution of Patient Status



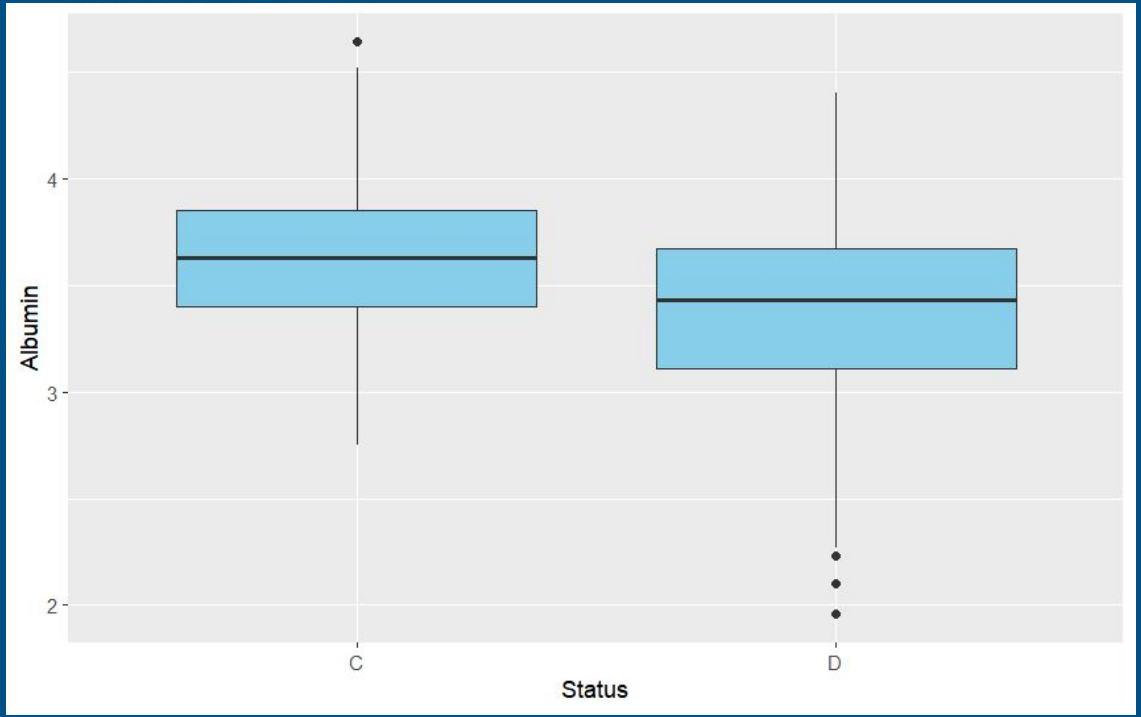
Age Distribution



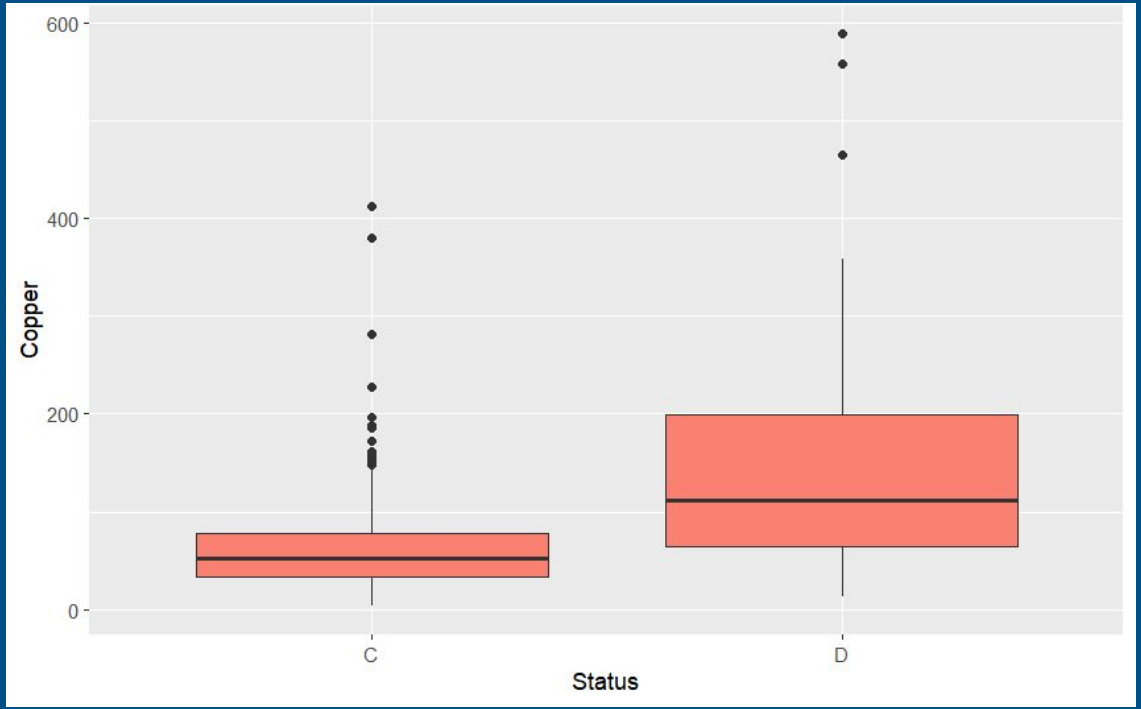
Bilirubin by Patient Status



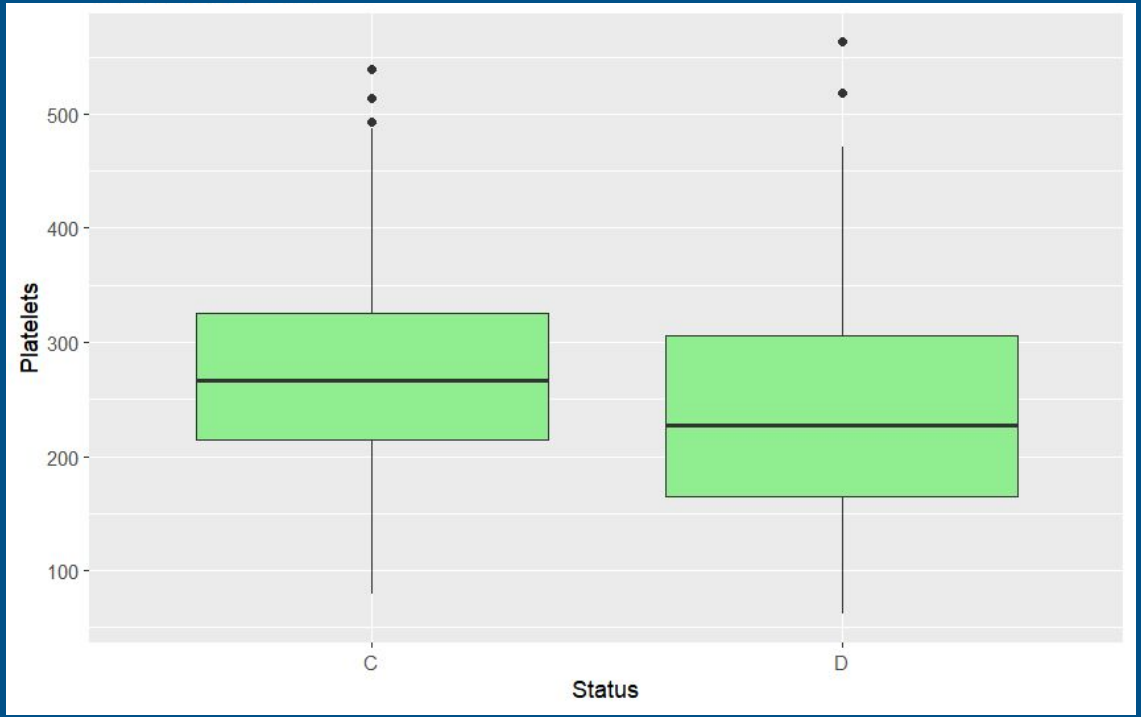
Albumin Levels by Patient Status



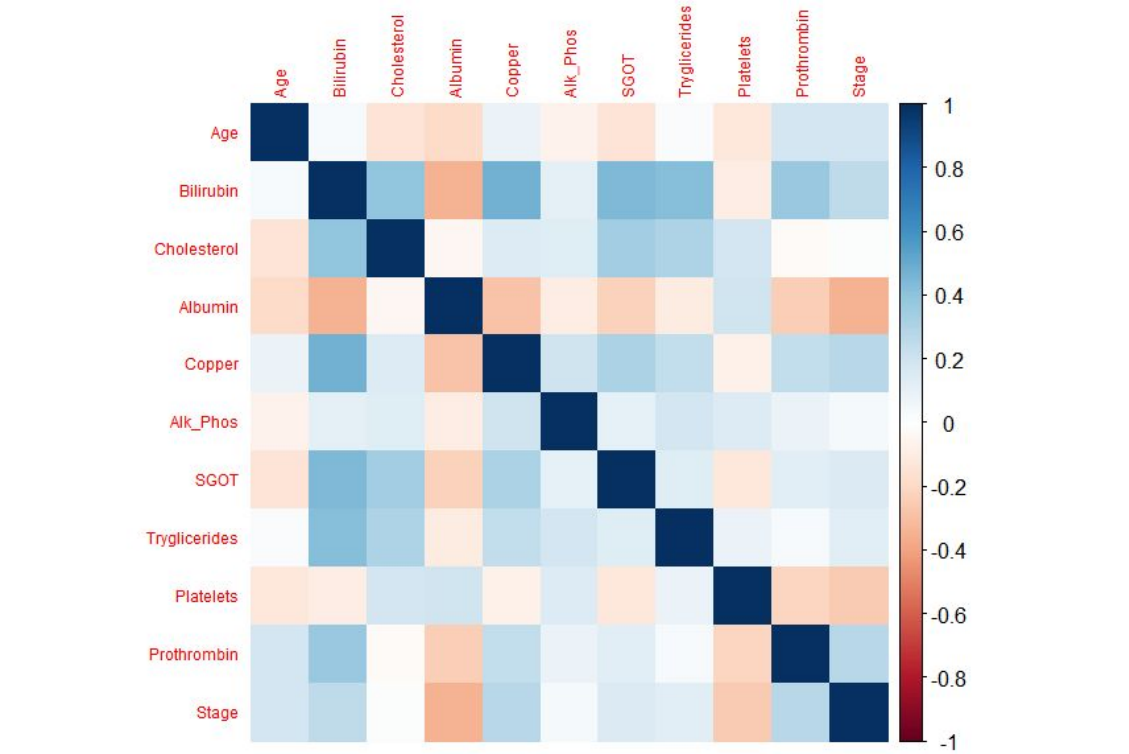
Copper Levels by Patient Status



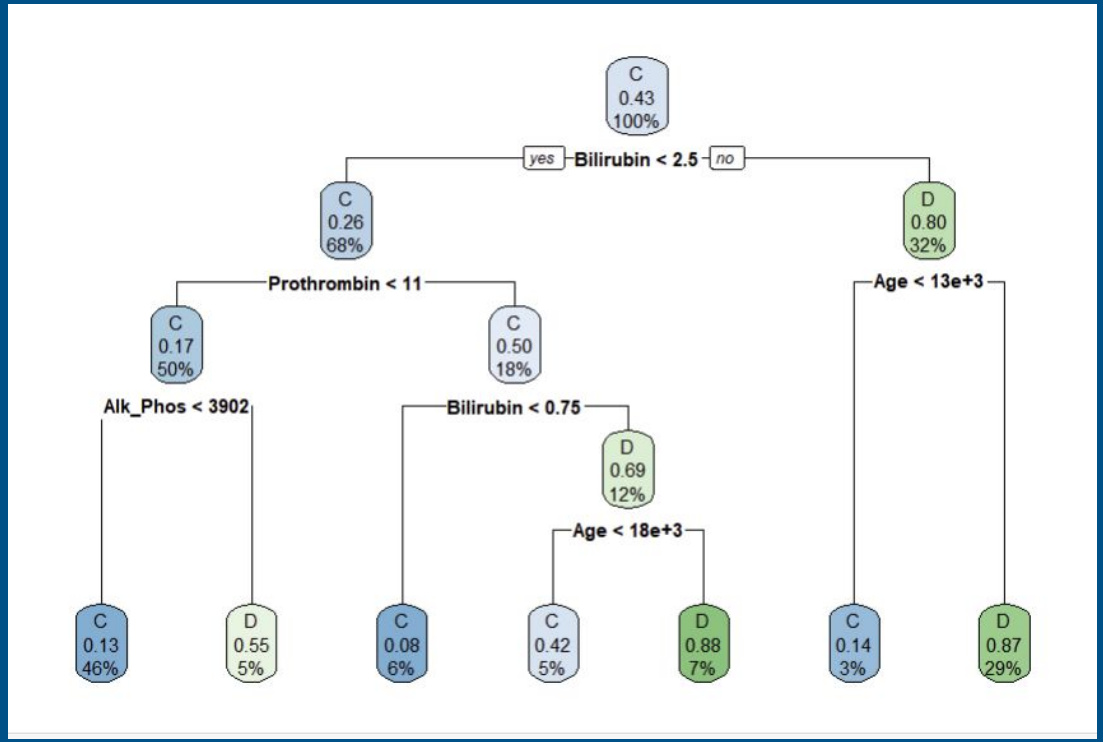
Platelets by Patient Status



Correlation Plot

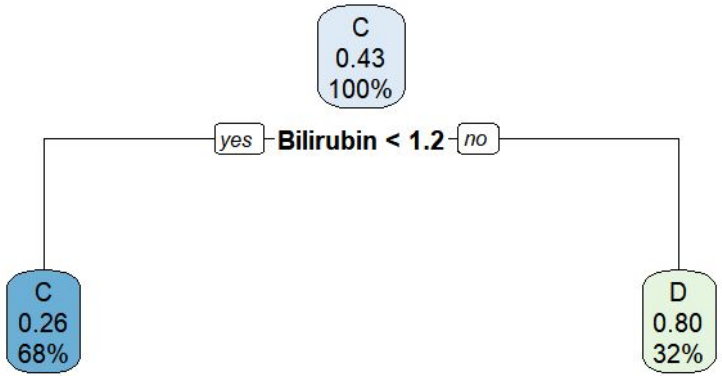


Decision Tree (cp = 0.001)




```
Confusion Matrix and Statistics

          Reference
Prediction  C  D
         C 29  4
         D  5 21

               Accuracy : 0.8475
                 95% CI : (0.7301, 0.9278)
    No Information Rate : 0.5763
    P-Value [Acc > NIR] : 7.736e-06

                  Kappa : 0.6893

 Mcnemar's Test P-Value : 1

            Sensitivity : 0.8529
            Specificity : 0.8400
         Pos Pred Value : 0.8788
         Neg Pred Value : 0.8077
             Prevalence : 0.5763
         Detection Rate : 0.4915
   Detection Prevalence : 0.5593
      Balanced Accuracy : 0.8465

       'Positive' Class : C
```

Decision Tree (cp = 0.09263)




```
Confusion Matrix and Statistics

          Reference
Prediction  C  D
         C 28  9
         D  6 16

               Accuracy : 0.7458
                 95% CI : (0.6156, 0.8502)
    No Information Rate : 0.5763
    P-Value [Acc > NIR] : 0.005209

                  Kappa : 0.471

 Mcnemar's Test P-Value : 0.605577

            Sensitivity : 0.8235
            Specificity : 0.6400
         Pos Pred Value : 0.7568
         Neg Pred Value : 0.7273
             Prevalence : 0.5763
         Detection Rate : 0.4746
   Detection Prevalence : 0.6271
      Balanced Accuracy : 0.7318

       'Positive' Class : C
```

Naive Bayes

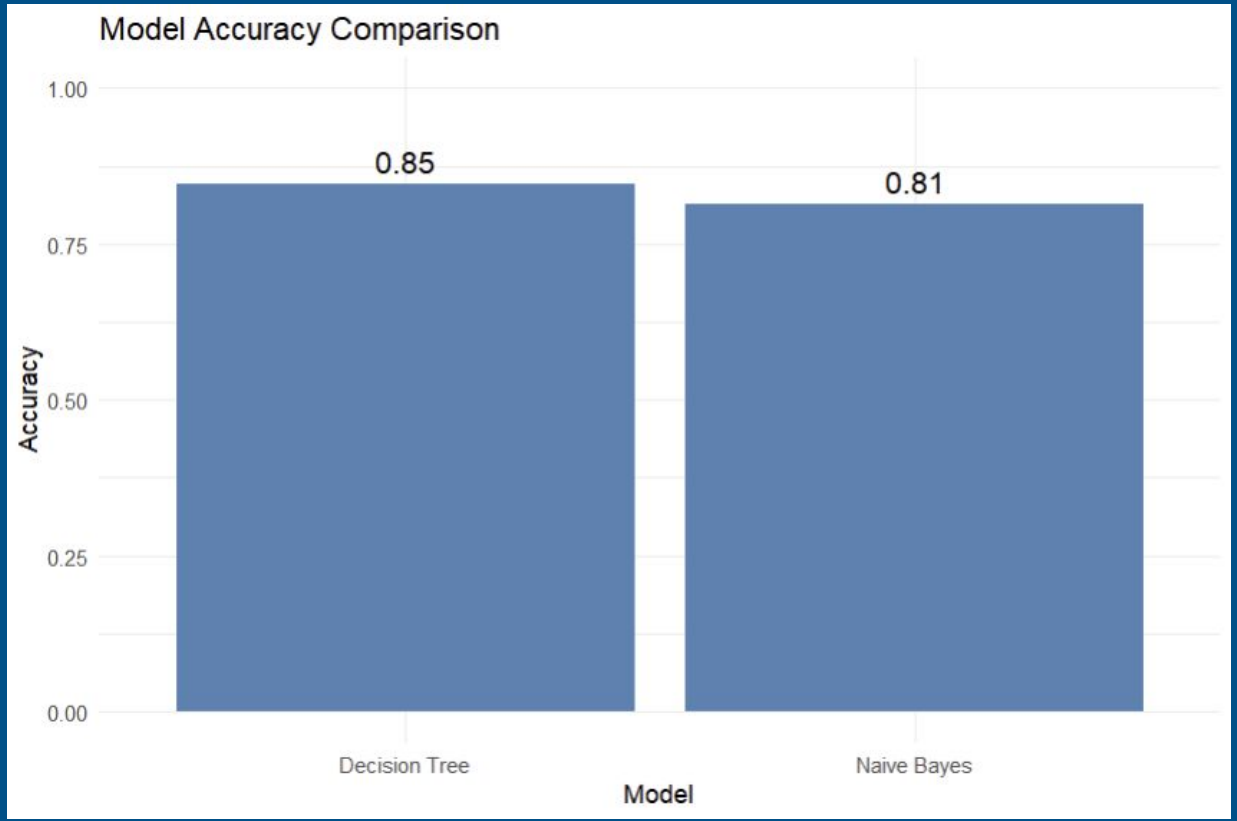
Before Cross Validation

```
Confusion Matrix and Statistics

          Reference
Prediction  C  D
         C 29  6
         D  5 19

               Accuracy : 0.8136
                 95% CI : (0.6909, 0.9031)
    No Information Rate : 0.5763
    P-Value [Acc > NIR] : 0.0001012

                  Kappa : 0.6162

 Mcnemar's Test P-Value : 1.0000000

            Sensitivity : 0.8529
            Specificity : 0.7600
         Pos Pred Value : 0.8286
         Neg Pred Value : 0.7917
             Prevalence : 0.5763
         Detection Rate : 0.4915
   Detection Prevalence : 0.5932
      Balanced Accuracy : 0.8065

       'Positive' Class : C
```

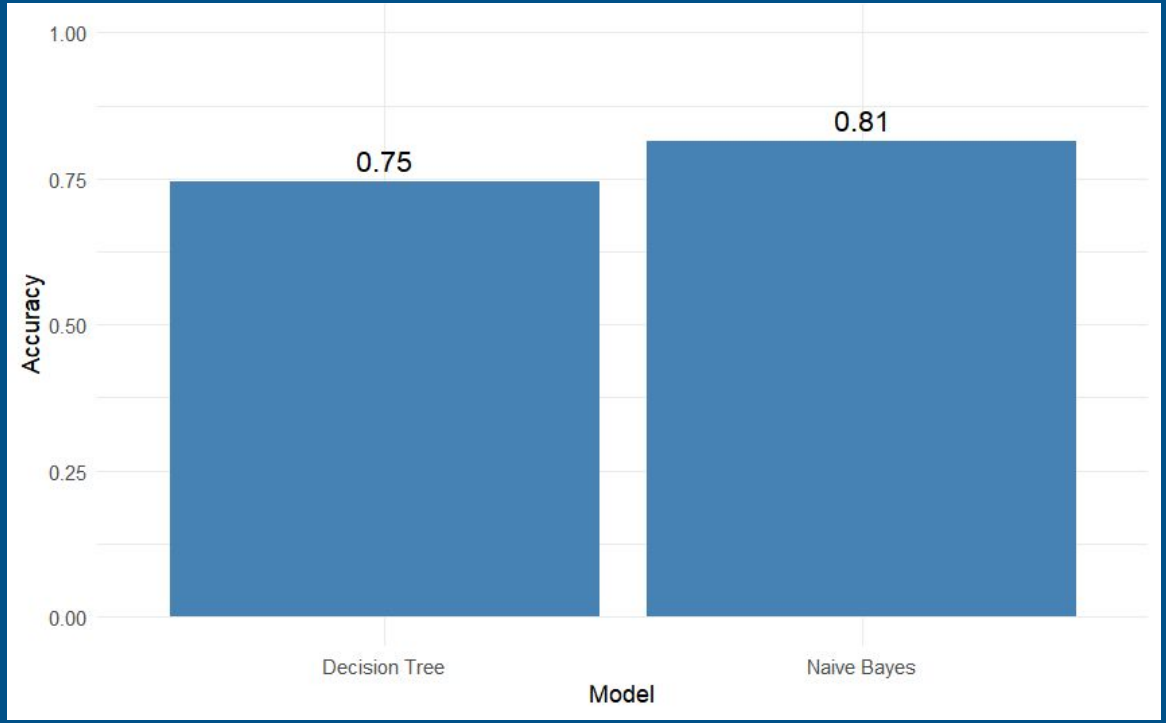
After Cross Validation

```
Confusion Matrix and Statistics

          Reference
Prediction  C  D
         C 30  7
         D  4 18

               Accuracy : 0.8136
                 95% CI : (0.6909, 0.9031)
    No Information Rate : 0.5763
    P-Value [Acc > NIR] : 0.0001012

                  Kappa : 0.6121

 Mcnemar's Test P-Value : 0.5464936

            Sensitivity : 0.8824
            Specificity : 0.7200
         Pos Pred Value : 0.8108
         Neg Pred Value : 0.8182
             Prevalence : 0.5763
         Detection Rate : 0.5085
   Detection Prevalence : 0.6271
      Balanced Accuracy : 0.8012

       'Positive' Class : C
```

Model Accuracy Comparison



Cross Validated Model Accuracy Comparison



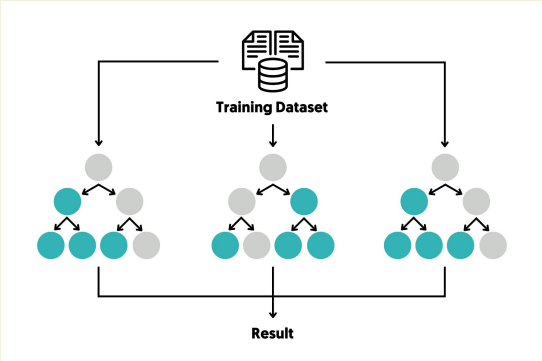
Discussion

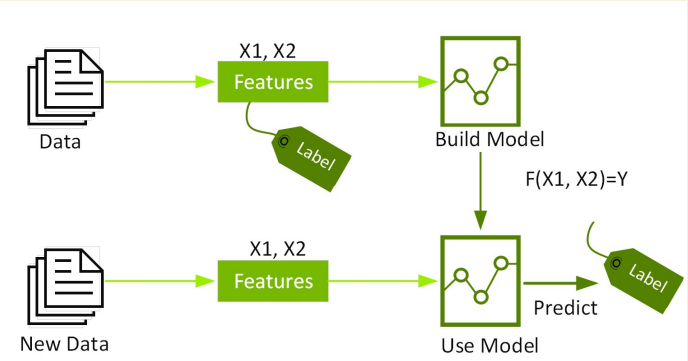
The decision tree and Naive Bayes models were both effective in predicting patient survival status. The decision tree with a manually selected complexity parameter (cp = 0.001) achieved the highest test accuracy of 84.75%, while the Naive Bayes model achieved a slightly lower accuracy of 81.36%. Interestingly, cross-validation suggested a higher cp value (0.0926), but that resulted in lower test performance (74.58%), indicating that the simpler tree may have underfit the data. This highlights that while cross-validation is a powerful tool for general model tuning, it doesn't always guarantee the best result on a specific set.

The decision tree showed balanced performance with a sensitivity of 85.29% and specificity of 84.00%, meaning it was equally effective at correctly identifying both patients who died and those who were censored. Naive Bayes demonstrated a slightly higher sensitivity (88.24%), but lower specificity (72.00%), indicating it was more prone to false positives for the 'death' class.

Discussion (cont.)

These results suggest that the decision tree model captured the complex, possibly nonlinear interactions between variables more effectively than Naive Bayes, which assumes conditional independence among features. Additionally, preprocessing steps like log transformation of skewed numeric variables helped improve both models' performance. Further improvements could involve exploring ensemble methods such as random forests or gradient boosting, which often outperform single decision trees while maintaining interpretability.





Literature Cited

Definition & Facts for Cirrhosis | NIDDK. (n.d.). National Institute of Diabetes and Digestive and Kidney Diseases. https://www.niddk.nih.gov/health-information/liver-disease/cirrhosis/definition-facts

Baki, J. A., & Tapper, E. B. (2019). Contemporary Epidemiology of Cirrhosis. Current Treatment Options in Gastroenterology, 17(2), 244–253. https://doi.org/10.1007/s11938-019-00228-3

UCI Machine Learning Repository. (n.d.). Archive.ics.uci.edu. https://archive.ics.uci.edu/dataset/878/cirrhosis+patient+survival+prediction+dataset-1

Acknowledgments

ChatGPT was used to improve explanations and debugging R code.
I would also like to thank Amrutha Karuturi for their guidance in applying laplace smoothing and log transformations to avoid zero-probability issues in Naive Bayes modeling.

Appendix (R Code)

```r
library(dplyr)
library(ggplot2)
library(rpart)
library(rpart.plot)
library(corrplot)
library(rsample)
library(e1071)
library(caret)
library(tidyr)
library(klaR)

set.seed(12345)
setwd("~/School/data101/finalproject/")
cirrhosis_data <- read.csv("cirrhosis.csv")

# ============================
# 1. Data Loading & Cleaning
# ============================

# remove ID column and N_Days column (not predictors)
cirrhosis_data <- subset(cirrhosis_data, select = -c(ID))
cirrhosis_data <- subset(cirrhosis_data, select = -c(N_Days))

# convert categorical data to factors with proper labels
cirrhosis_data$Status <- as.factor(cirrhosis_data$Status)
cirrhosis_data$Drug <- as.factor(cirrhosis_data$Drug)
cirrhosis_data$Sex <- as.factor(cirrhosis_data$Sex)
cirrhosis_data$Ascites <- as.factor(cirrhosis_data$Ascites)
cirrhosis_data$Hepatomegaly <- as.factor(cirrhosis_data$Hepatomegaly)
cirrhosis_data$Spiders <- as.factor(cirrhosis_data$Spiders)
cirrhosis_data$Edema <- as.factor(cirrhosis_data$Edema)

# remove instances where Status == CL,
# excluding patients with liver transplants
# will focus purely on outcomes of death vs. censored
cirrhosis_data <- cirrhosis_data %>%
  filter(Status != "CL") %>%
  droplevels()

# overview of dataset
str(cirrhosis_data)
summary(cirrhosis_data)
head(cirrhosis_data)
table(cirrhosis_data$Status)

# ensure there are no missing values that need handling
colSums(is.na(cirrhosis_data))

# drop the last 100 rows where there is a large amount of missingness (when Drug == NA)
cirrhosis_data <- cirrhosis_data[!is.na(cirrhosis_data$Drug), ]
colSums(is.na(cirrhosis_data))

# impute missing data using the mean
cirrhosis_data <- cirrhosis_data %>%
  mutate(
    Cholesterol = ifelse(is.na(Cholesterol), mean(Cholesterol, na.rm = TRUE), Cholesterol),
    Tryglicerides = ifelse(is.na(Tryglicerides), mean(Tryglicerides, na.rm = TRUE), Tryglicerides),
    Copper = ifelse(is.na(Copper), mean(Copper, na.rm = TRUE), Copper),
    Platelets = ifelse(is.na(Platelets), mean(Platelets, na.rm = TRUE), Platelets)
  )
colSums(is.na(cirrhosis_data))
table(cirrhosis_data$Status)

# =========
# 2. Plots
# =========

# Status distribution
ggplot(cirrhosis_data, aes(x = Status)) +
  geom_bar(fill = "steelblue") +
  labs(title = "Distribution of Patient Status")

# Age distribution
age_years = cirrhosis_data$Age / 365
ggplot(cirrhosis_data, aes(x = age_years)) +
  geom_histogram(bins = 30, fill = "darkgreen") +
  labs(title = "Age Distribution")
```

Appendix (2)

```r
# Bilirubin vs. Status
ggplot(cirrhosis_data, aes(x = Status, y = Bilirubin)) +
  geom_boxplot(fill = "tomato") +
  labs(title = "Bilirubin by Patient Status")

# Albumin vs. Status
ggplot(cirrhosis_data, aes(x = Status, y = Albumin)) +
  geom_boxplot(fill = "skyblue") +
  labs(title = "Albumin Levels by Status")

# Copper vs. Status
ggplot(cirrhosis_data, aes(x = Status, y = Copper)) +
  geom_boxplot(fill = "salmon") +
  labs(title = "Copper Levels by Status")

# Platelets vs. Status
ggplot(cirrhosis_data, aes(x = Status, y = Platelets)) +
  geom_boxplot(fill = "lightgreen") +
  labs(title = "Platelets by Status")

# ========================
# 3. Correlation Analysis
# ========================

num_vars <- cirrhosis_data[, sapply(cirrhosis_data, is.numeric)]

corrplot(cor(num_vars), method = "color", tl.cex = 0.6)

# ====================
# 4. Train-Test Split
# ====================

split_obj <- initial_split(cirrhosis_data, prop = 0.8, strata = Status)
split_obj

train_data <- training(split_obj)
```

```r
summary(train_data)

test_data <- testing(split_obj)
summary(test_data)

# =============
# 5. Modeling
# =============

# Decision tree
tree_model <- rpart(Status~., data = train_data, method = "class", control=rpart.control(cp=0.001))
rpart.plot(tree_model)
pred_tree <- predict(tree_model, test_data, type = "class")
confusionMatrix(pred_tree, test_data$Status) # Accuracy: 0.8475

# Naive Bayes
nb_model <- naiveBayes(Status~., data = train_data)
pred_nb <- predict(nb_model, test_data)
confusionMatrix(pred_nb, test_data$Status) # Accuracy: 0.8136

# k-fold cross validation, k=10
ctrl <- trainControl(method = "cv", number = 10, savePredictions = TRUE)
tree_cv <- train(Status~., data = train_data, method = "rpart", trControl = ctrl, tuneLength = 20)
print(tree_cv) # best cp value = 0.09263158 with accuracy 0.7442029

nb_cv <- train(Status~., data = train_data, method = "nb", trControl = ctrl, tuneLength = 10)
print(nb_cv)

# adjust grid to avoid probabilty = 0 from cross-validation
grid <- expand.grid(fL = 1, usekernel = FALSE, adjust = c(1))

# since there are still errors, apply log transformation to the numeric variables
num_vars <- c("Age", "Bilirubin", "Cholesterol", "Albumin", "Copper", "Alk_Phos", "SGOT", "Tryglicerides", "Platelets", "Prothrombin", "Stage")
train_data[num_vars] <- log1p(train_data[num_vars])
test_data[num_vars] <- log1p(test_data[num_vars])

# rerun cross-validation again with grid and log transformation
nb_cv_logged <- train(Status~., data = train_data, method = "nb", trControl = ctrl, tuneGrid = grid, tuneLength = 10)
```

Appendix (3)

```
# remake decision tree and naive baiyes with cross validation
new_tree <- rpart(Status~.,data=train_data,method="class",control=rpart.control(cp=0.09263158))
rpart.plot(new_tree)
pred_tree <- predict(new_tree, test_data, type = "class")
confusionMatrix(pred_tree, test_data$Status)

pred_nb_cv <- predict(nb_cv_logged, test_data)
confusionMatrix(pred_nb_cv, test_data$Status)

acc_nb <- confusionMatrix(pred_nb_cv, test_data$Status)$overall["Accuracy"]
acc_tree <- confusionMatrix(pred_tree, test_data$Status)$overall["Accuracy"]

results <- data.frame(Model = c("Naive Bayes", "Decision Tree"),Accuracy = c(acc_nb, acc_tree))

# Bar plot of cross-validated model accuracies
ggplot(results, aes(x = Model, y = Accuracy)) +
  geom_bar(stat = "identity", fill = "steelblue") +
  ylim(0, 1) +
  geom_text(aes(label = sprintf("%.2f", Accuracy)), vjust = -0.5, size = 4.5) +
  labs(title = "Model Accuracy Comparison", y = "Accuracy", x = "Model") +
  theme_minimal()

# cp == 0.001 decision tree accuracy
results$Accuracy[results$Model == "Decision Tree"] <- 0.8475

# Bar plot of best model accuracies
ggplot(results, aes(x = Model, y = Accuracy)) +
  geom_bar(stat = "identity", fill = "steelblue") +
  ylim(0, 1) +
  geom_text(aes(label = sprintf("%.2f", Accuracy)), vjust = -0.5, size = 4.5) +
  labs(title = "Model Accuracy Comparison", y = "Accuracy", x = "Model") +
  theme_minimal()

print(results)
```

Appendix (4) (Output)

```
> library(dplyr)

> library(ggplot2)

> library(rpart)

> library(rpart.plot)

> library(corrplot)

> library(rsample)

> library(e1071)

> library(caret)

> library(tidyr)

> library(klaR)

> set.seed(12345)

> setwd("~/School/data101/finalproject/")

> cirrhosis_data <- read.csv("cirrhosis.csv")

> # ============================
> # 1. Data Loading & Cleaning
> # ============================
>
> # remove ID column and N_Days column (not predic .... [TRUNCATED]

> cirrhosis_data <- subset(cirrhosis_data, select = -c(N_Days))

> # convert categorical data to factors with proper labels
> cirrhosis_data$Status <- as.factor(cirrhosis_data$Status)

> cirrhosis_data$Drug <- as.factor(cirrhosis_data$Drug)
```

```
> cirrhosis_data$Sex <- as.factor(cirrhosis_data$Sex)

> cirrhosis_data$Ascites <- as.factor(cirrhosis_data$Ascites)

> cirrhosis_data$Hepatomegaly <- as.factor(cirrhosis_data$Hepatomegaly)

> cirrhosis_data$Spiders <- as.factor(cirrhosis_data$Spiders)

> cirrhosis_data$Edema <- as.factor(cirrhosis_data$Edema)

> # remove instances where Status == CL,
> # excluding patients with liver transplants
> # will focus purely on outcomes of death vs. censored
> cir .... [TRUNCATED]

> # overview of dataset
> str(cirrhosis_data)
'data.frame':   393 obs. of  18 variables:
 $ Status       : Factor w/ 2 levels "C","D": 2 1 2 2 2 1 2 2 2 2 ...
 $ Drug         : Factor w/ 2 levels "D-penicillamine",..: 1 1 1 1 2 2 2 1 2 2 ...
 $ Age          : int  21464 20617 25594 19994 24201 20284 19379 15526 25772 19619 ...
 $ Sex          : Factor w/ 2 levels "F","M": 1 1 2 1 1 1 1 1 1 1 ...
 $ Ascites      : Factor w/ 2 levels "N","Y": 2 1 1 1 1 1 1 1 2 1 ...
 $ Hepatomegaly : Factor w/ 2 levels "N","Y": 2 2 1 2 2 2 1 1 1 2 ...
 $ Spiders      : Factor w/ 2 levels "N","Y": 2 2 1 2 1 1 1 2 2 2 ...
 $ Edema        : Factor w/ 3 levels "N","S","Y": 3 1 2 2 1 1 1 1 3 1 ...
 $ Bilirubin    : num  14.5 1.1 1.4 1.8 0.8 1 0.3 3.2 12.6 1.4 ...
 $ Cholesterol  : int  261 302 176 244 248 322 280 562 200 259 ...
 $ Albumin      : num  2.6 4.14 3.48 2.54 3.98 4.09 4 3.08 2.74 4.16 ...
 $ Copper       : int  156 54 210 64 50 52 52 79 140 46 ...
 $ Alk_Phos     : num  1718 7395 516 6122 944 ...
 $ SGOT         : num  137.9 113.5 96.1 60.6 93 ...
 $ Tryglicerides: int  172 88 55 92 63 213 189 88 143 79 ...
 $ Platelets    : int  190 221 151 183 NA 204 373 251 302 258 ...
 $ Prothrombin  : num  12.2 10.6 12 10.3 11 9.7 11 11 11.5 12 ...
 $ Stage        : int  4 3 4 4 3 3 3 2 4 4 ...
```

Appendix (5)

```
> table(cirrhosis_data$Status)

  C   D
168 125

> # =========
> # 2. Plots
> # =========
>
> # Status distribution
> ggplot(cirrhosis_data, aes(x = Status)) +
+   geom_bar(fill = "steelblue") +
+ .... [TRUNCATED]

> # Age distribution
> age_years = cirrhosis_data$Age / 365

> ggplot(cirrhosis_data, aes(x = age_years)) +
+   geom_histogram(bins = 30, fill = "darkgreen") +
+   labs(title = "Age Distribution")

> # Bilirubin vs. Status
> ggplot(cirrhosis_data, aes(x = Status, y = Bilirubin)) +
+   geom_boxplot(fill = "tomato") +
+   labs(title = "Bilirubin by ..." ... [TRUNCATED]

> # Albumin vs. Status
> ggplot(cirrhosis_data, aes(x = Status, y = Albumin)) +
+   geom_boxplot(fill = "skyblue") +
+   labs(title = "Albumin Levels  ..." ... [TRUNCATED]

> # Copper vs. Status
> ggplot(cirrhosis_data, aes(x = Status, y = Copper)) +
+   geom_boxplot(fill = "salmon") +
+   labs(title = "Copper Levels by S ..." ... [TRUNCATED]

> # Platelets vs. Status
> ggplot(cirrhosis_data, aes(x = Status, y = Platelets)) +
+   geom_boxplot(fill = "lightgreen") +
+   labs(title = "Platelet ..." ... [TRUNCATED]
```

```
> # ==========================
>
> num_vars <- cirrhosis_data[, sapply(cirrhosis_data, is.nu .... [TRUNCATED]

> corrplot(cor(num_vars), method = "color", tl.cex = 0.6)

> # =====================
> # 4. Train-Test Split
> # =====================
>
> split_obj <- initial_split(cirrhosis_data, prop = 0.8, strata = Statu .... [TRUNCATED]

> split_obj
<Training/Testing/Total>
<234/59/293>

> train_data <- training(split_obj)

> summary(train_data)
 Status               Drug           Age        Sex     Ascites Hepatomegaly Spiders Edema
 C:134   D-penicillamine:117   Min.   : 9598   F:208   N:215   N:122        N:159   N:198
 D:100   Placebo        :117   1st Qu.:15604   M: 26   Y: 19   Y:112        Y: 75   S: 19
                               Median :18414                                        Y: 17
                               Mean   :18390
                               3rd Qu.:20614
                               Max.   :28650
   Bilirubin       Cholesterol        Albumin          Copper          Alk_Phos
 Min.   : 0.300   Min.   : 120.0   Min.   :1.960   Min.   :  4.00   Min.   :  310.0
 1st Qu.: 0.725   1st Qu.: 250.2   1st Qu.:3.333   1st Qu.: 39.00   1st Qu.:  857.2
 Median : 1.300   Median : 309.5   Median :3.550   Median : 68.50   Median : 1214.5
 Mean   : 3.224   Mean   : 355.2   Mean   :3.525   Mean   : 95.54   Mean   : 1940.6
 3rd Qu.: 3.275   3rd Qu.: 374.8   3rd Qu.:3.797   3rd Qu.:121.75   3rd Qu.: 1909.5
 Max.   :28.000   Max.   :1775.0   Max.   :4.640   Max.   :588.00   Max.   :13862.4
      SGOT        Tryglicerides      Platelets      Prothrombin        Stage
 Min.   : 28.38   Min.   : 44.00   Min.   : 70.0   Min.   : 9.00   Min.   :1.000
 1st Qu.: 79.28   1st Qu.: 84.25   1st Qu.:195.0   1st Qu.:10.00   1st Qu.:2.000
 Median :110.05   Median :114.00   Median :253.5   Median :10.60   Median :3.000
 Mean   :120.01   Mean   :121.64   Mean   :257.9   Mean   :10.77   Mean   :3.004
 3rd Qu.:149.31   3rd Qu.:141.50   3rd Qu.:321.8   3rd Qu.:11.20   3rd Qu.:4.000
 Max.   :328.60   Max.   :432.00   Max.   :563.0   Max.   :17.10   Max.   :4.000
```

Appendix (6)

```
> test_data <- testing(split_obj)

> summary(test_data)
 Status              Drug         Age        Sex    Ascites Hepatomegaly Spiders Edema
 C:34   D-penicillamine:31   Min.   :10550   F:52   N:54    N:23         N:49    N:48
 D:25   Placebo        :28   1st Qu.:16002   M: 7   Y: 5    Y:36         Y:10    S: 8
                             Median :18713                                       Y: 3
                             Mean   :18830
                             3rd Qu.:22271
                             Max.   :26259
   Bilirubin       Cholesterol        Albumin          Copper          Alk_Phos
 Min.   : 0.400   Min.   : 174.0   Min.   :2.100   Min.   :  9.00   Min.   :  289.0
 1st Qu.: 0.800   1st Qu.: 263.5   1st Qu.:3.190   1st Qu.: 51.00   1st Qu.:  927.5
 Median : 1.400   Median : 360.0   Median :3.570   Median : 74.00   Median : 1523.0
 Mean   : 3.422   Mean   : 402.6   Mean   :3.487   Mean   : 97.47   Mean   : 2293.7
 3rd Qu.: 5.100   3rd Qu.: 426.5   3rd Qu.:3.795   3rd Qu.:132.00   3rd Qu.: 2145.5
 Max.   :17.900   Max.   :1480.0   Max.   :4.520   Max.   :290.00   Max.   :11552.0
      SGOT        Tryglicerides     Platelets      Prothrombin        Stage
 Min.   : 26.35   Min.   : 46.0   Min.   : 62.0   Min.   : 9.20   Min.   :1.000
 1st Qu.: 80.75   1st Qu.: 97.0   1st Qu.:206.5   1st Qu.:10.00   1st Qu.:3.000
 Median :117.80   Median :118.0   Median :259.5   Median :10.60   Median :3.000
 Mean   :130.21   Mean   :133.7   Mean   :265.8   Mean   :10.66   Mean   :3.068
 3rd Qu.:168.95   3rd Qu.:151.0   3rd Qu.:320.5   3rd Qu.:11.00   3rd Qu.:4.000
 Max.   :457.25   Max.   :598.0   Max.   :539.0   Max.   :13.20   Max.   :4.000

> # ==============
> # 5. Modeling
> # ==============
>
> # Decision tree
> tree_model <- rpart(Status~., data = train_data, method = "class", contro .... [TRUNCATED]

> rpart.plot(tree_model)

> pred_tree <- predict(tree_model, test_data, type = "class")

> confusionMatrix(pred_tree, test_data$Status) # Accuracy: 0.8475
```

```
Confusion Matrix and Statistics

          Reference
Prediction  C  D
         C 29  4
         D  5 21

               Accuracy : 0.8475
                 95% CI : (0.7301, 0.9278)
    No Information Rate : 0.5763
    P-Value [Acc > NIR] : 7.736e-06

                  Kappa : 0.6893

 Mcnemar's Test P-Value : 1

            Sensitivity : 0.8529
            Specificity : 0.8400
         Pos Pred Value : 0.8788
         Neg Pred Value : 0.8077
             Prevalence : 0.5763
         Detection Rate : 0.4915
   Detection Prevalence : 0.5593
      Balanced Accuracy : 0.8465

       'Positive' Class : C


> # Naive Bayes
> nb_model <- naiveBayes(Status~., data = train_data)

> pred_nb <- predict(nb_model, test_data)

> confusionMatrix(pred_nb, test_data$Status) # Accuracy: 0.8136
Confusion Matrix and Statistics

          Reference
Prediction  C  D
         C 29  6
         D  5 19
```

Appendix (7)

```
              Accuracy : 0.8136
                95% CI : (0.6909, 0.9031)
    No Information Rate : 0.5763
    P-Value [Acc > NIR] : 0.0001012

                 Kappa : 0.6162

 Mcnemar's Test P-Value : 1.0000000

            Sensitivity : 0.8529
            Specificity : 0.7600
         Pos Pred Value : 0.8286
         Neg Pred Value : 0.7917
             Prevalence : 0.5763
         Detection Rate : 0.4915
   Detection Prevalence : 0.5932
      Balanced Accuracy : 0.8065

       'Positive' Class : C


> # k-fold cross validation, k=10
> ctrl <- trainControl(method = "cv", number = 10, savePredictions = TRUE)

> tree_cv <- train(Status~., data = train_data, method = "rpart", trControl = ctrl, tuneLength = 20)

> print(tree_cv) # best cp value = 0.09263158 with accuracy 0.7442029
CART

234 samples
 17 predictor
  2 classes: 'C', 'D'

No pre-processing
Resampling: Cross-Validated (10 fold)
Summary of sample sizes: 211, 211, 211, 210, 211, 211, ...
Resampling results across tuning parameters:

  cp          Accuracy   Kappa
  0.00000000  0.7088768  0.4106501
```

```
  0.02315789  0.7262681  0.4412829
  0.04631579  0.7141304  0.4100386
  0.06947368  0.7355072  0.4496489
  0.09263158  0.7442029  0.4657740
  0.11578947  0.7442029  0.4657740
  0.13894737  0.7442029  0.4657740
  0.16210526  0.7442029  0.4657740
  0.18526316  0.7442029  0.4657740
  0.20842105  0.7442029  0.4657740
  0.23157895  0.7442029  0.4657740
  0.25473684  0.7442029  0.4657740
  0.27789474  0.7442029  0.4657740
  0.30105263  0.7442029  0.4657740
  0.32421053  0.7442029  0.4657740
  0.34736842  0.7442029  0.4657740
  0.37052632  0.7442029  0.4657740
  0.39368421  0.7442029  0.4657740
  0.41684211  0.7137681  0.3926222
  0.44000000  0.6159420  0.1423824

Accuracy was used to select the optimal model using the largest value.
The final value used for the model was cp = 0.3936842.

> nb_cv <- train(Status~., data = train_data, method = "nb", trControl = ctrl, tuneLength = 10)

> print(nb_cv)
Naive Bayes

234 samples
 17 predictor
  2 classes: 'C', 'D'

No pre-processing
Resampling: Cross-Validated (10 fold)
Summary of sample sizes: 211, 210, 210, 211, 211, 210, ...
Resampling results across tuning parameters:

  usekernel  Accuracy   Kappa
  FALSE      0.7647192  0.5000096
```

Appendix (8)

```
  usekernel  Accuracy   Kappa
  FALSE      0.7647192  0.5000096
   TRUE      0.7487319  0.4529074

Tuning parameter 'fL' was held constant at a value of 0
Tuning parameter 'adjust' was
 held constant at a value of 1
Accuracy was used to select the optimal model using the largest value.
The final values used for the model were fL = 0, usekernel = FALSE and adjust = 1.

> # adjust grid to avoid probabilty = 0 from cross-validation
> grid <- expand.grid(fL = 1, usekernel = FALSE, adjust = c(1))

> # since there are still errors, apply log transformation to the numeric variables
> num_vars <- c("Age", "Bilirubin", "Cholesterol", "Albumin", "Cop ..." ... [TRUNCATED]

> train_data[num_vars] <- log1p(train_data[num_vars])

> test_data[num_vars] <- log1p(test_data[num_vars])

> # rerun cross-validation again with grid and log transformation
> nb_cv_logged <- train(Status~., data = train_data, method = "nb", trControl = ctrl .... [TRUNCATED]

> # remake decision tree and naive baiyes with cross validation
> new_tree <- rpart(Status~.,data=train_data,method="class",control=rpart.control(cp=0 .... [TRUNCATED]

> rpart.plot(new_tree)

> pred_tree <- predict(new_tree, test_data, type = "class")

> confusionMatrix(pred_tree, test_data$Status)
Confusion Matrix and Statistics

          Reference
Prediction  C  D
         C 28  9
         D  6 16
```

```
               Accuracy : 0.7458
                 95% CI : (0.6156, 0.8502)
    No Information Rate : 0.5763
    P-Value [Acc > NIR] : 0.005209

                  Kappa : 0.471

 Mcnemar's Test P-Value : 0.605577

            Sensitivity : 0.8235
            Specificity : 0.6400
         Pos Pred Value : 0.7568
         Neg Pred Value : 0.7273
             Prevalence : 0.5763
         Detection Rate : 0.4746
   Detection Prevalence : 0.6271
      Balanced Accuracy : 0.7318

       'Positive' Class : C


> pred_nb_cv <- predict(nb_cv_logged, test_data)

> confusionMatrix(pred_nb_cv, test_data$Status)
Confusion Matrix and Statistics

          Reference
Prediction  C  D
         C 30  7
         D  4 18

               Accuracy : 0.8136
                 95% CI : (0.6909, 0.9031)
    No Information Rate : 0.5763
    P-Value [Acc > NIR] : 0.0001012

                  Kappa : 0.6121

 Mcnemar's Test P-Value : 0.5464936
```

Appendix (9)

```
 Mcnemar's Test P-Value : 0.5464936

            Sensitivity : 0.8824
            Specificity : 0.7200
         Pos Pred Value : 0.8108
         Neg Pred Value : 0.8182
             Prevalence : 0.5763
         Detection Rate : 0.5085
   Detection Prevalence : 0.6271
      Balanced Accuracy : 0.8012

       'Positive' Class : C


> acc_nb <- confusionMatrix(pred_nb_cv, test_data$Status)$overall["Accuracy"]

> acc_tree <- confusionMatrix(pred_tree, test_data$Status)$overall["Accuracy"]

> results <- data.frame(Model = c("Naive Bayes", "Decision Tree"),Accuracy = c(acc_nb, acc_tree))

> # Bar plot of cross-validated model accuracies
> ggplot(results, aes(x = Model, y = Accuracy)) +
+   geom_bar(stat = "identity", fill = "steelblue" .... [TRUNCATED]

> # cp == 0.001 decision tree accuracy
> results$Accuracy[results$Model == "Decision Tree"] <- 0.8475

> # Bar plot of best model accuracies
> ggplot(results, aes(x = Model, y = Accuracy)) +
+   geom_bar(stat = "identity", fill = "steelblue") +
+   yli .... [TRUNCATED]

> print(results)
          Model  Accuracy
1   Naive Bayes 0.8135593
2 Decision Tree 0.8475000
```