CIRRHOSIS PATIENT SURVIVAL PREDICTION DATA ANALYSIS

VERÓNICA GAMO PAREJO

ANDREA PRADAS AGUJETAS

JAIME MARTÍN CASADO

10 – 01 – 2025

Machine Learning – Biomedical Engineering

CEU San Pablo University

Abraham Otero

Index

Introduction

1.1 Background and Objective

Cirrhosis is a chronic liver disease characterized by progressive scarring of the liver, often resulting from prolonged damage caused by conditions such as hepatitis or excessive alcohol consumption [1]. This condition can lead to severe complications and, in many cases, mortality. Accurately predicting survival outcomes for patients with cirrhosis is critical to improving clinical decision-making and treatment planning.

The primary goal of this study is to address a classification problem by developing machine learning models that leverage clinical, laboratory and histologic features to predict the survival status of patients with cirrhosis. For the analysis of this dataset, two tools, AI Studio and Python (via Google Collab), will be used. Throughout this document, the methodology and key results found during the analysis will be explained and presented. The code can be accessed through the following link: https://github.com/veronicagamo/Aprendizaje-Autom-tico.git

1.2 Dataset Description

The dataset used in this study comes from a Mayo Clinic research project on primary biliary cirrhosis (PBC) conducted between 1974 and 1984 [2]. It includes data from 418 patients, each represented by the following 20 clinical, laboratory and histologic features:

- **ID**: unique identifier.
- **N_Days**: number of days from registration to the earliest of death, liver transplantation, or study end (July 1986). It is treated as an integer.
- **Status**: is the target variable in this classification problem. It is a categorical variable with the following values: D (death), C (the patient was alive and their data was censored at the end of the study period), CL (censored because they underwent a liver transplant).
- **Drug**: type of drug administered: D-penicillamine or placebo. It is treated as a nominal value.
- **Age**: patient's age in days. It is treated as an integer.
- **Sex**: M (male) or F (female). It is treated as a binominal value.
- **Ascites**: presence of ascites: N (No) or Y (Yes). It is treated as a binominal value.
- **Hepatomegaly**: presence of hepatomegaly: N (No) or Y (Yes). It is treated as a binominal value.
- **Spiders**: presence of spider angiomas: N (No) or Y (Yes). It is treated as a binominal value.

- **Edema**: presence of edema: N = no edema and no diuretic therapy, S = edema present without diuretics or resolved with diuretics. Y = edema despite diuretic therapy. It is treated as a nominal value.
- **Bilirubin**: serum bilirubin in mg/dl. It is treated as an integer.
- **Cholesterol**: serum cholesterol in mg/dl. It is treated as an integer.
- **Albumin**: albumin levels in gm/dl. It is treated as a real value.
- **Copper**: urine copper in µg/day. It is treated as an integer.
- **Alk_Phos**: alkaline phosphatase in U/liter. It is treated as a real value.
- **SGOT**: SGOT levels in U/ml. It is treated as a real value.
- **Triglycerides**: triglycerides in mg/dl. It is treated as an integer.
- **Platelets**: platelet count per cubic ml/1000. It is treated as an integer.
- **Prothrombin**: prothrombin time in seconds. It is treated as a real value.
- **Stage**: histologic stage of disease (1, 2, 3, or 4). It is treated as an ordinal value.

The dataset includes missing values, represented as "NA" to denote the absence of data for specific attributes due to incomplete data collection. To ensure accurate handling of these missing values, the "Declare Missing Values" operator in AI Studio was used, transforming the representation of missing values to "?".

Before proceeding with further analysis, the ID attribute was removed from the dataset using the “Select Attributes” operator in AI Studio, as it does not provide meaningful information or contribute to the analysis. Its inclusion could introduce unnecessary noise or redundancy.

Data Preparation

2.1 Data Quality

The overall quality of the dataset can be considered moderate/poor. In addition to the significant presence of missing values and inconsistencies, the dataset suffers from a low number of instances. It contains 418 instances with no duplicated records, which ensures data uniqueness and reduces redundancy. However, the presence of missing values in 142 rows, where one or more attributes are missing, affects the dataset's completeness and poses challenges for analysis. This represents a significant proportion (approximately 34%) of the dataset, and if not addressed appropriately, it could introduce bias or inaccuracies in the outcomes. For this reason, it is essential to handle the missing values carefully, to preserve the integrity of the analysis and ensure the reliability of the results.

There are six instances in the dataset (with IDs 313, 317, 319, 322, 334, and 337) where the majority of attributes contain missing data for the corresponding patients. According to the

source of the dataset, these patients became untraceable shortly after their diagnosis. These instances will be removed from the dataset using the "Filter Examples" operator in AI Studio, as they provide no usable information for analysis and as it was recommended from the author.

The dataset includes several attributes with no missing values, such as, N_Days, Status, ID, Age, Sex, Edema, Bilirubin, and Albumin. To properly address the missing values, it is crucial to understand the context in which the data was collected. The dataset consists of two distinct groups:

1. Trial participants (312 patients): these patients participated in the randomized placebo-controlled trial, receiving either the drug D-penicillamine or a placebo. As key participants in the clinical study, comprehensive clinical data was consistently collected for this group, ensuring no missing values for the Drug attribute.

   The absence of missing data for Drug in this cohort underscores the rigorous and systematic data collection process associated with the trial. However, other attributes, such as Cholesterol, Cooper, Tryglicerides or Platelets may still exhibit missing values. Since we do not know the reason why those values are missing, if the missingness is due to the structure of the trial (for example, some tests or evaluations were not done because the patient's condition did not meet specific criteria or because they were randomized to the placebo group), the missingness could be categorized as MAR. However, if the missingness is related to the severity of the patient's disease (Status), the missingness could be classified as MNAR.

2. Non-trial participants (100 patients): this group consists of patients who did not participate in the randomized clinical trial. As a result, only basic metrics and survival tracking were recorded, while many clinical attributes (Ascites, Spiders, Hepatomegaly…) and staging information, are missing.

   The lack of detailed data in this group reflects the limited scope of their involvement, as they were not subjected to the rigorous testing and evaluation protocols followed for trial participants. The missing values in this group are predominantly, as the absence of data is directly related to their non-participation in the trial and the associated lack of data collection processes.

Due to the limitations of AI Studio, which allows a maximum of 10 attributes to be represented in graphs, the scatter matrices for both the trial and non-trial groups are presented separately, one for numeric attributes and another for nominal attributes. These scatter matrices were created using Python to provide a comprehensive visual representation. While they are introduced at the beginning of this section for an overview, the conclusions are detailed and

justified in the following sections, supported by additional AI Studio-generated graphics for further clarity.

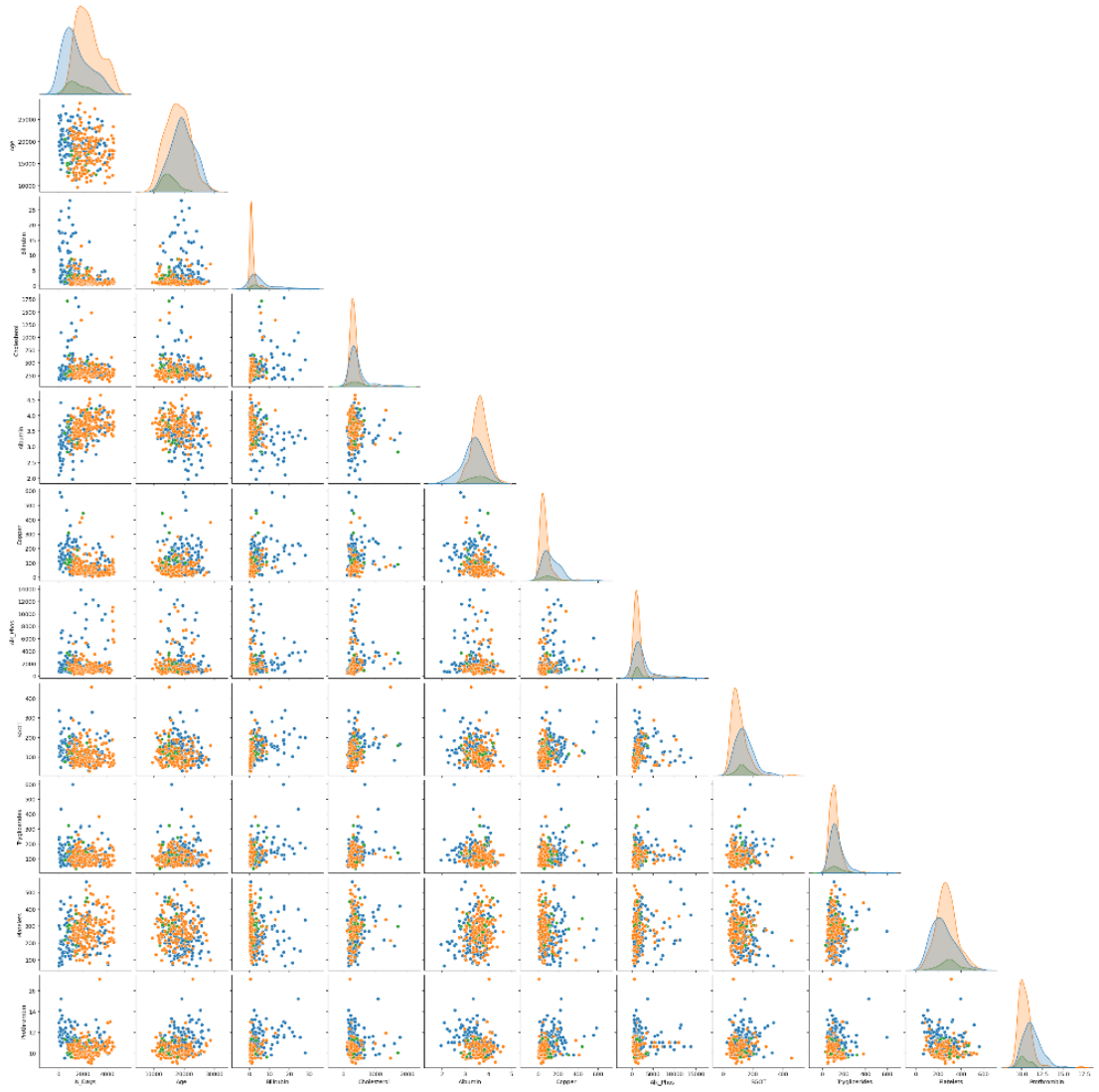


Figure 1. Trial group scatter matrix for numeric attributes.

Figure 2. Trial group scatter matrix for non-numeric attributes.

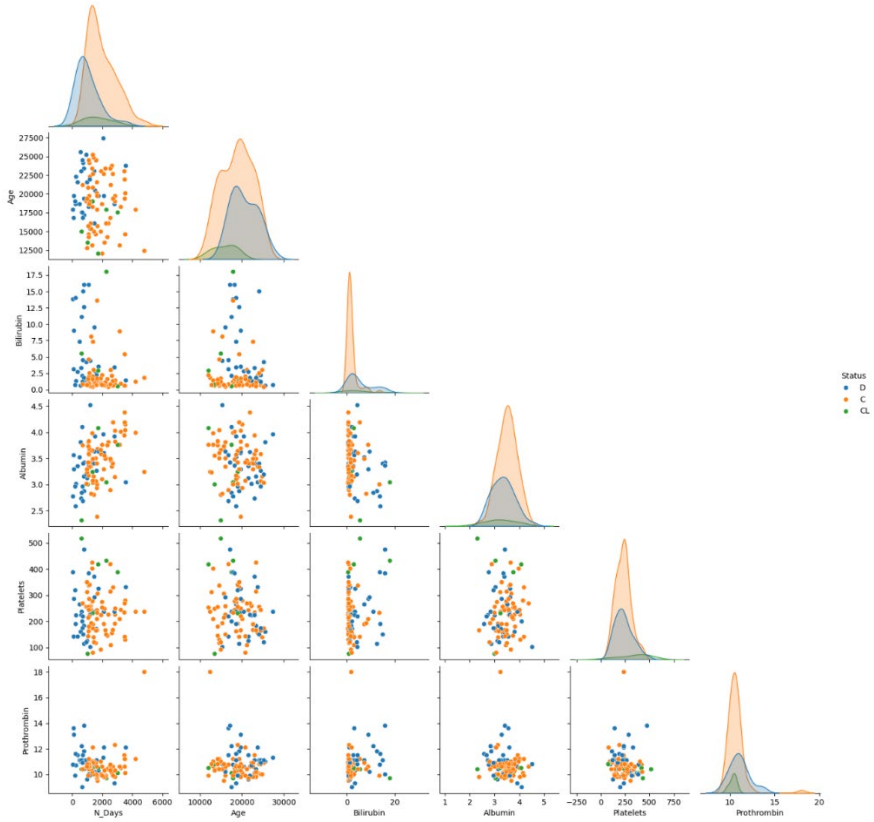


Figure 3. Non-trial group scatter matrix for numeric attributes.

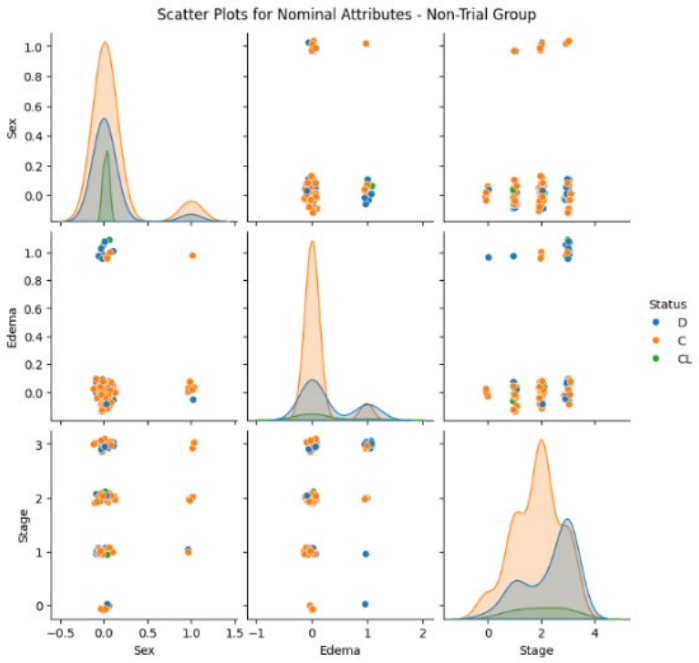


Figure 4. Non-trial group scatter matrix for non-numeric attributes.

2.1.1 HANDLING MISSING VALUES

Due to the significant differences between these two groups, the strategy for handling missing values involves dividing the dataset into two distinct groups: those with the attribute Drug and those without it. This separation acknowledges the fundamental differences in the completeness and quality of data between trial participants and non-trial participants.

From this point forward, we will work on each group independently and finally combine the models. For the trial group, we address missing values for attributes such as Platelets, Copper, Triglycerides, and Cholesterol by replacing them with their respective mean values. This approach fills in missing data points using the average value from the observed data for each attribute, thereby maintaining the dataset's consistency and reducing the likelihood of introducing significant bias during the imputation process. On the other hand, for the non-trial group, attributes with no observed values across any instances (Drug, Ascites, Hepatomegaly, Spiders, Cholesterol, Copper, Triglycerides, Alk Phos, and SGOT) were first removed from the dataset. Subsequently, for the remaining attributes where at least some instances had observed values, missing values were handled using mean imputation (only Platelets and Prothrombin contained missing values). The decision to replace missing values with the mean is appropriate given the numerical nature of these attributes, ensuring a consistent dataset while preserving the overall statistical properties of the data.

2.1.1 OUTLIERS

In both groups, no obvious outliers are observed in the box plots using AI Studio. While some points in the scatter plots appear isolated from the majority of the data, these should not

automatically be classified as outliers. These isolated data points likely correspond to patients with more severe liver disease outcomes, such as death or liver transplantation.

For instance, extreme values such as Cholesterol levels of 1775 mg/dL and 1712 mg/dL, or Alkaline Phosphatase levels reaching 13862.4 IU/L, stand out as exceptionally high but are likely reflective of severe disease states rather than erroneous data. Rather than being dismissed as anomalies, these points reflect the natural variability inherent in the dataset and are essential for understanding the extremes of liver disease progression.

From the perspective of their values, these data points do not appear to be unrealistic or erroneous. Instead, they might be considered extreme real values, representing critical cases within the population. It's important to reflect on the purpose of the analysis when deciding how to handle these extreme cases. For instance, removing individuals with exceptionally high cholesterol or alkaline phosphatase levels might eliminate crucial insights about the progression of severe liver conditions. Therefore, it would be prudent to retain these data points in the model to ensure that the analysis encompasses the full spectrum of disease severity.

If we plot the boxplots in Python (Figure 5 and 6), we can observe that the outliers are represented by circles. This occurs because a different algorithm is used compared to AI Studio. However, as previously mentioned, these points are not erroneous data, and therefore, they will be retained for the analysis.

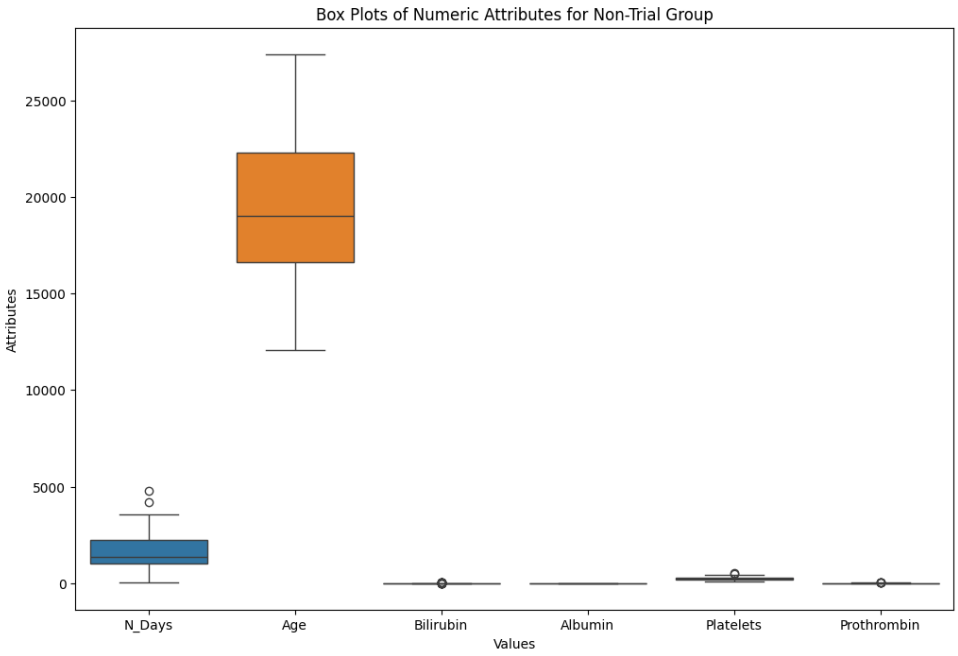


Figure 5. Boxplot non-trial group in Python.

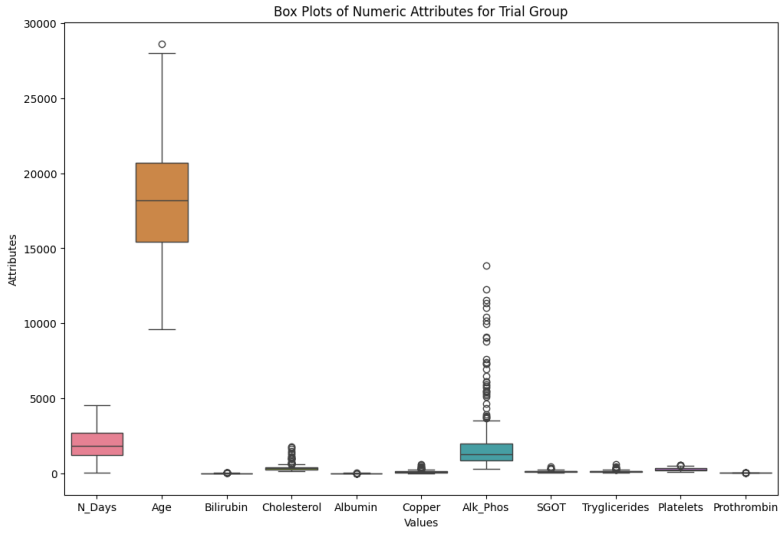


Figure 6. Boxplot trial group in Python.

2.2 Data Exploration

2.2.1 CORRELATION

The degree of association between attributes in the dataset was assessed by computing two separate correlation matrices, one for the trial group and another for the non-trial group, to account for differences in data collection. A high linear correlation between two attributes suggests that as the value of one increase, the other tends to increase as well. Conversely, a low or negative correlation indicates an inverse relationship, where the value of one attribute decreases as the other increases.

Several initially nominal attributes, such as Drug, Sex, Ascites, Hepatomegaly, Spiders, Stage and Edema, were numerized using the to enable their inclusion in the analysis. The matrices were computed using the Altair AI Studio Correlation Matrix Operator, which implements the Pearson correlation coefficient.

In the first correlation matrix (Figure 7), representing the trial group, several moderate positive correlations are evident, reflecting the interplay between clinical attributes associated with liver disease and its progression. A moderate positive correlation is found between Hepatomegaly and Stage (0.467). This association reflects how progressive liver disease often leads to hepatomegaly (enlarged liver) as a compensatory response to increasing liver damage and fibrosis. Similarly, a moderate positive correlation exists between Bilirubin and Copper levels (0.456). This relationship can be explained because a liver dysfunction leads to elevated bilirubin and copper retention in the liver and bloodstream, highlighting the co-accumulation. Another notable positive correlation is observed between Bilirubin and SGOT (0.442). Both markers are indicators of liver damage, with elevated SGOT levels reflecting liver cell injury and increased bilirubin levels resulting from impaired excretion due to liver damage. Additionally, a positive correlation between N_Days and Albumin levels (0.436) suggests that patients with

higher albumin levels tend to have longer survival times. Finally, Bilirubin and Triglycerides are positively correlated (0.418), potentially due to the impact of liver disease on lipid metabolism. Also, several notable negative correlations highlight key relationships between clinical attributes and disease progression. A moderate negative correlation exists between N_Days and Bilirubin levels (-0.442) which indicates that patients with higher bilirubin levels tend to have shorter survival times.

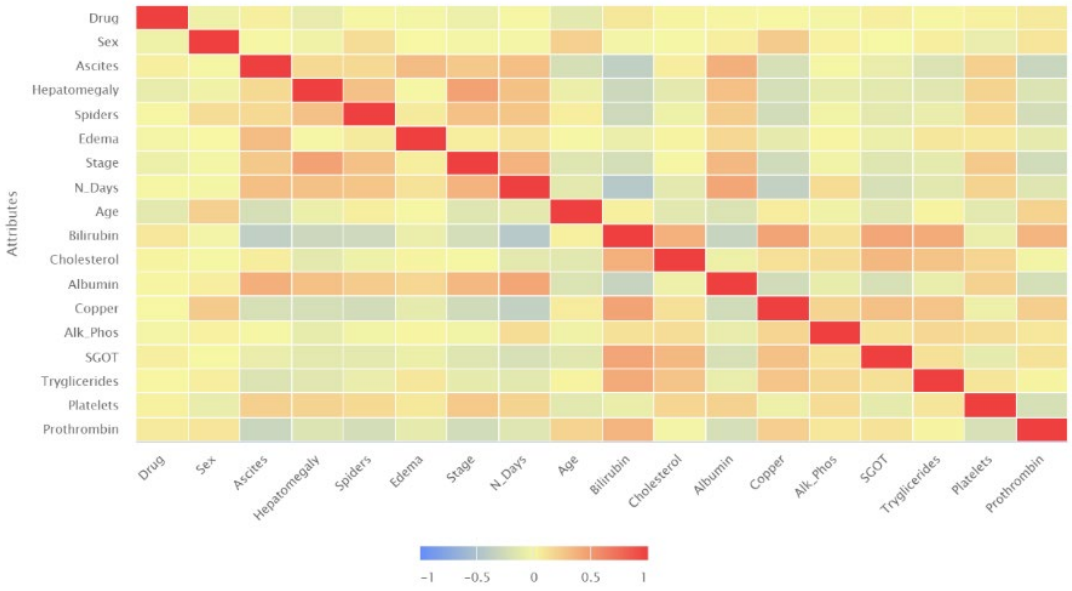


Figure 7. Trial group correlation matrix.

On the other hand, the correlations observed in the non-trial group (Figure 8) are notably weaker compared to those in the trial group, largely due to the limited and incomplete data available for this set. Despite these limitations, the relationships identified within this group remain consistent with established clinical patterns of liver disease progression and align with those observed in the trial group. For example, the moderate positive correlation (0,425) between Albumin levels and N_Days highlights the clinical importance of this liver-synthesized protein as a predictor of better outcomes.

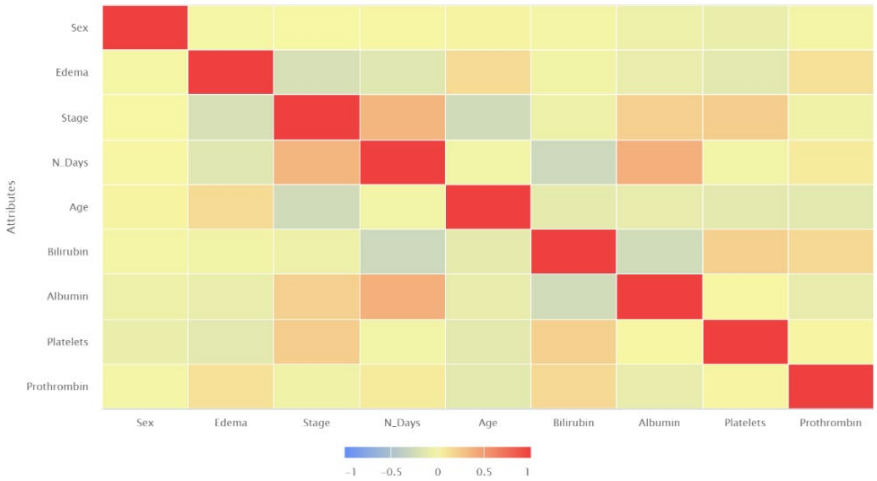


Figure 8. Non-trial group correlation matrix.

Overall, these attributes do not exhibit strong linear relationships in either group. Most scatterplots display a cloud-like shape, with points scattered randomly without noticeable alignment along a straight line. This pattern suggests that the relationships between these variables are likely non-linear. The non-linear behaviour observed in the scatterplots is corroborated by the long tails discussed later, as attributes with prominent skewness and extreme values often deviate from linear trends. This reinforces the observation that linearity is generally absent in these datasets and the variability in the data is better captured through non-linear relationships. It is important to mention that the correlation matrix calculated in Python is consistent with the one obtained from AI Studio, which is why it is not specified separately.

2.2.2 PROBABILITY DISTRIBUTIONS

Normality is assessed by observing the shape of histograms and box plots for the numeric attributes. A normal distribution is characterized by a symmetric bell curve in the histogram, where the mean and median are approximately equal, and the data points are evenly distributed around the centre. In addition, for a distribution to be considered normal, the box plot should exhibit symmetry, with the median line located roughly in the centre of the box. The whiskers on either side should be of approximately equal length, reflecting a balanced spread of data around the centre. Now for each group, only the attributes that most closely approximate normality and those with the most prominent long-tail behaviour are mentioned.

For the non-trial group, most numeric attributes exhibit distributions that deviate from normality. However, some attributes, such as Age and Albumin, display characteristics that approximate normality. The distribution of Age follows a roughly bell-shaped curve, suggesting it is close to normal, with a central tendency around the mean. This is supported by the corresponding box plot, which shows a relatively symmetric spread around the median and relatively equal whiskers on both sides of the median, indicating a balanced spread of the data. Similarly, Albumin exhibits a distribution with a central peak and a relatively symmetric spread, as seen in both the histogram and box plot. The box plot for Albumin further confirms this approximate normality, as the whiskers are roughly equal in length, suggesting that the data is evenly distributed without extreme values skewing the distribution.

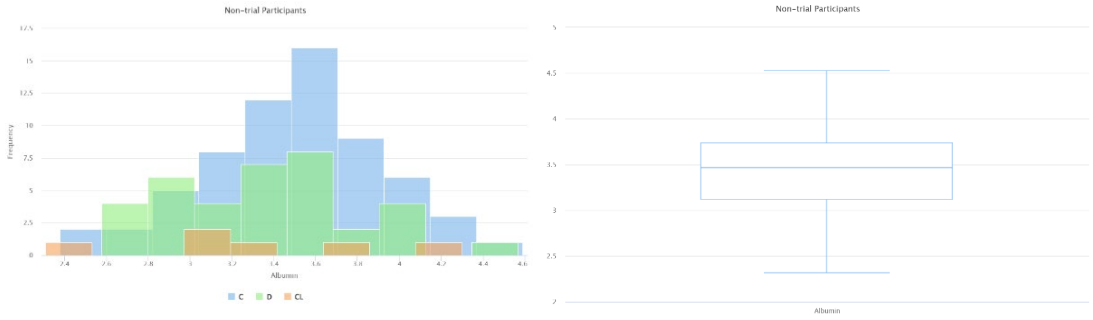


Figure 9. Albumin histogram and boxplot (non-trial).

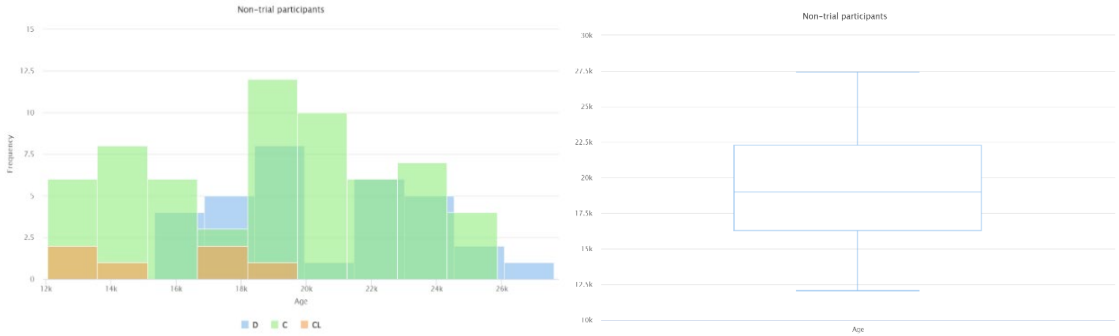


Figure 10. Age histogram and boxplot (non-trial).

These attributes, while not perfectly normal, display visual and statistical characteristics consistent with approximate normality. In contrast, other numeric attributes, such as Bilirubin and Prothrombin, display pronounced right-skewness and long tails, as evidenced by their histograms and box plots, indicating a clear deviation from normality. For Bilirubin, the box plot highlights a significantly extended upper whisker, with a maximum value of 18 mg/dL, far exceeding the upper quartile value of 3.075. This pronounced long tail indicates the presence of a subset of patients with exceptionally high bilirubin levels, often indicative of severe liver dysfunction or advanced disease stages. Similarly, Prothrombin shows a long upper whisker, with a maximum value of 18 compared to an upper quartile value of 11. This right long tail suggests

that some patients have unusually elevated prothrombin levels, which could be linked to advanced disease stages or critical health conditions.

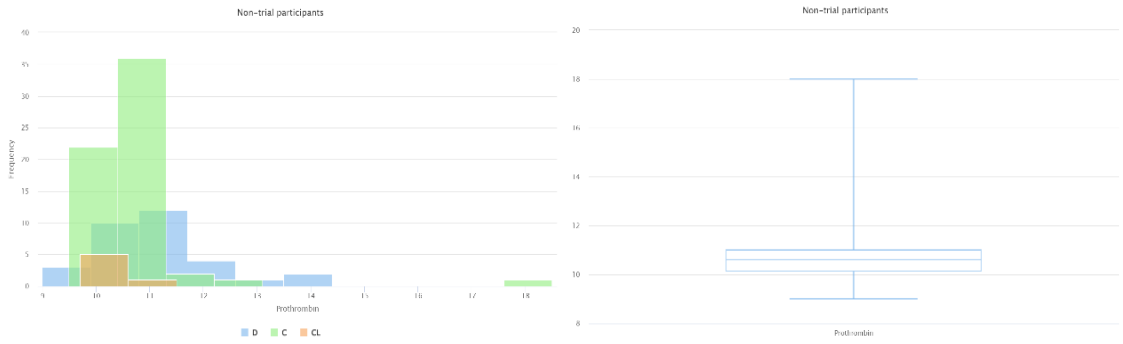


Figure 11. Prothrombin histogram and boxplot (non-trial).

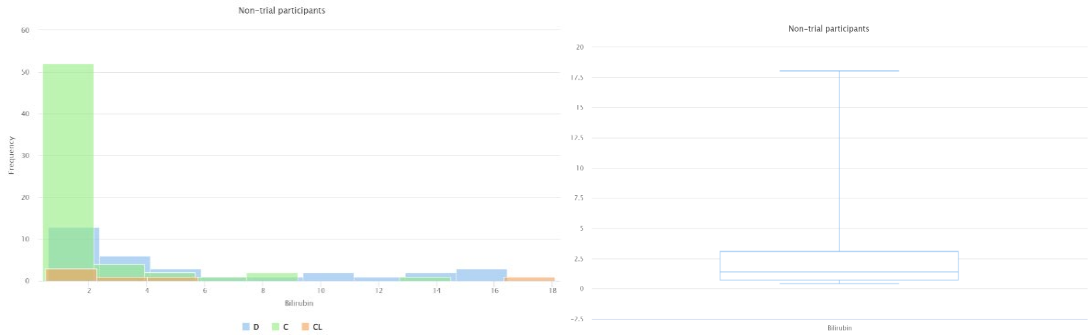


Figure 12. Bilirubin histogram and boxplot (non-trial).

The non-trial group displays marked disparities in the distribution of nominal attributes, reflecting significant class imbalances that may impact subsequent analyses. These imbalances are evident across several key attributes. The Status attribute shows a clear imbalance, with 58.5% of instances falling into the "C" category (62 instances), followed by 30.2% in "D" (32 instances), and only 5.7% in "CL" (6 instances). This disparity highlights an uneven distribution among the groups, with "CL" being particularly underrepresented. In the future, models might struggle to correctly classify instances of "CL". The Sex attribute is similarly skewed, with females making up a vast majority (86.8%, 92 instances), while males represent only 7.5% (8 instances). This imbalance limits the dataset's ability to provide robust insights into gender-specific analyses. Actually, the Edema attribute also reflects notable imbalance, with the "N" category dominating at 80.2% (85 instances), followed by 14.2% in "S" (15 instances). Importantly, there are no instances classified as "Y," leaving this category entirely unrepresented. These class imbalances will be addressed in subsequent analyses to mitigate their potential impact on the results.

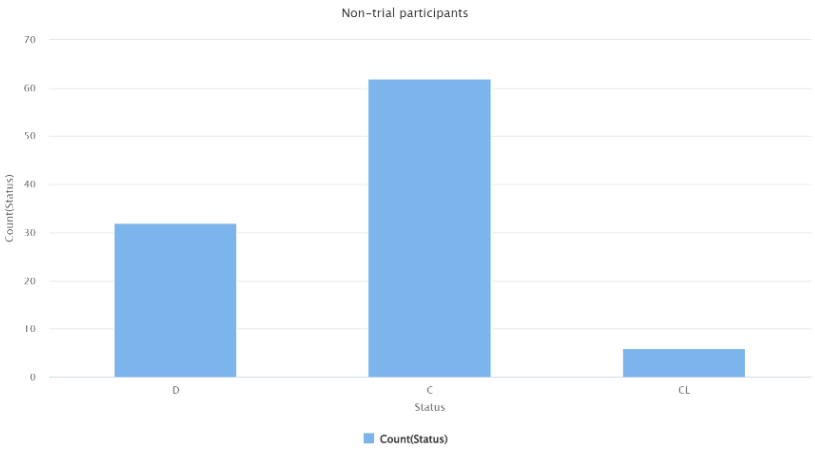


Figure 13. Status bar column (non-trial).

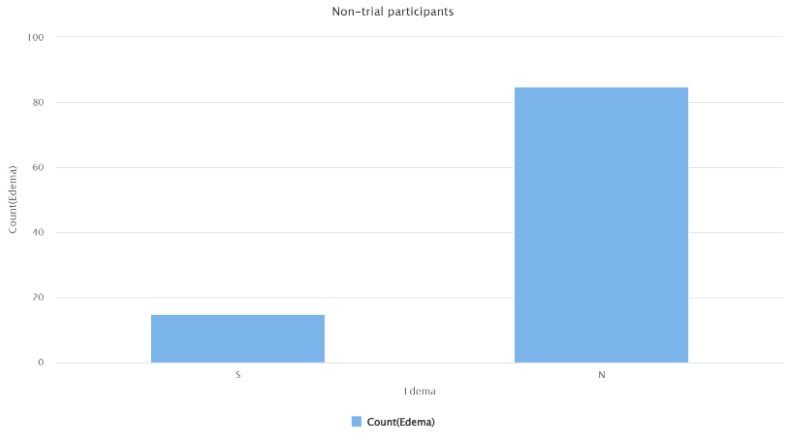


Figure 14. Edema bar column (non-trial).

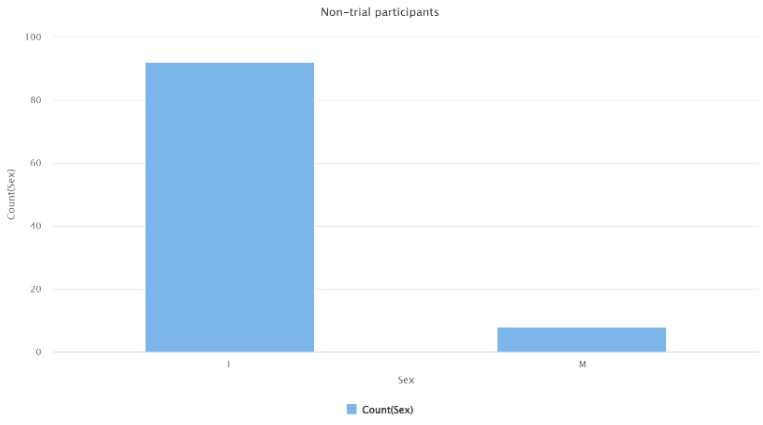


Figure 15. Sex bar column (non-trial).

For the trial participants, the numeric attributes that approximate normality include Age, Albumin, and Platelets. These attributes exhibit roughly symmetric histograms and box plots, supporting their near-normal distributions. The Age attribute demonstrates a bell-shaped curve with most values concentrated around the mean, suggesting an approximately normal distribution. The box plot further reflects this symmetry, with balanced whiskers and minimal

skewness, indicating an even spread of values across its range. Similarly, the Albumin attribute shows a centralized distribution with a pronounced peak around the mean, indicative of near-normality. The histogram reveals a relatively symmetric spread of values, with no extreme skewness observed. The box plot reinforces this observation, showing evenly distributed whiskers and minimal skew. However, slight deviations from perfect symmetry in the tails indicate that the distribution, while close to normality, is not entirely normal. For Platelets, the distribution aligns closely with a bell-shaped curve but exhibits a slight right skew. The histogram shows a subtle extension of higher values to the right, which is reflected in the box plot by a longer whisker on the upper end. Despite this slight skew, the overall distribution remains balanced, with a spread around the central tendency that approximates normality.

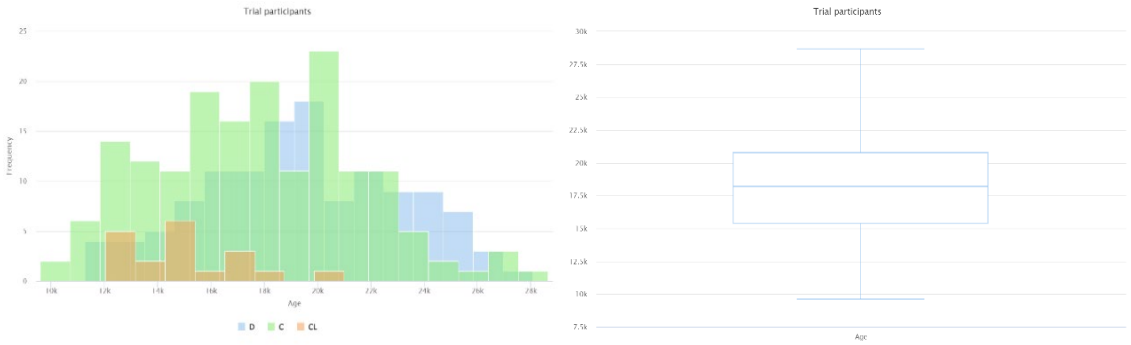


Figure 16. Age histogram and boxplot (trial).

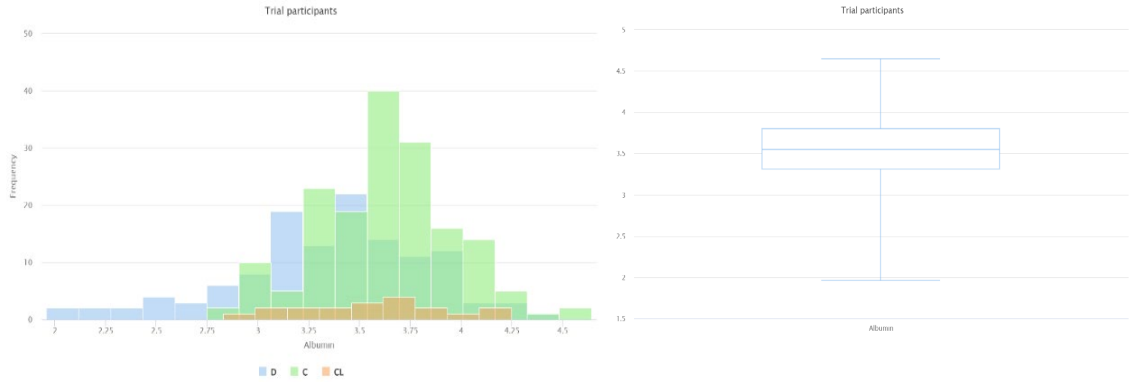


Figure 17. Albumin histogram and boxplot (trial).

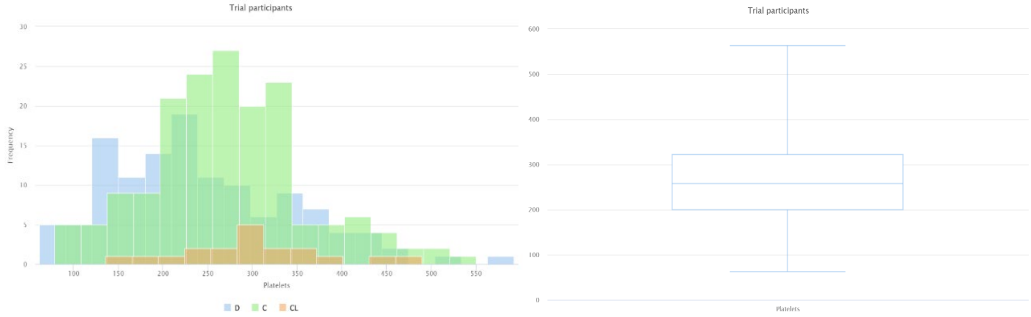


Figure 18. Platelets histogram and boxplot (trial).

In the trial group, the attributes Bilirubin, Cholesterol, Alk_Phos, Copper, Triglycerides, SGOT and Prothrombin exhibit long tails, primarily skewed toward higher values, indicating significant positive skewness. The histogram for Bilirubin shows a concentration of values near the median, with a subset of extremely high values contributing to a pronounced long right tail. Similarly, Cholesterol demonstrates a peak at lower values in the histogram, followed by a gradual decline toward higher values, reflecting a positively skewed distribution. The box plot for Cholesterol confirms this pattern with an extended upper whisker, significantly longer than the lower whisker, indicating the presence of outliers and a long upper tail. Alk_Phos follows a similar trend, with a sharp peak at lower values in the histogram and a tapering decline toward higher values, indicative of a long right tail. Copper's histogram displays a similar pattern of skewness, with most data points in the lower range and a few higher values extending the distribution. The Triglycerides distribution shows a peak near the lower range with a gradual decline forming a long tail. SGOT also exhibits a sharp concentration of values at the lower end, with a long tail extending to higher levels. Finally, Prothrombin, although less pronounced, this variable still shows a slight positive skew, with a concentration around the mean and a gradual tapering off toward higher values.

These patterns suggest that a subset of trial group patients presents abnormally elevated values for these attributes, often associated with advanced stages of liver disease or severe complications in liver function.

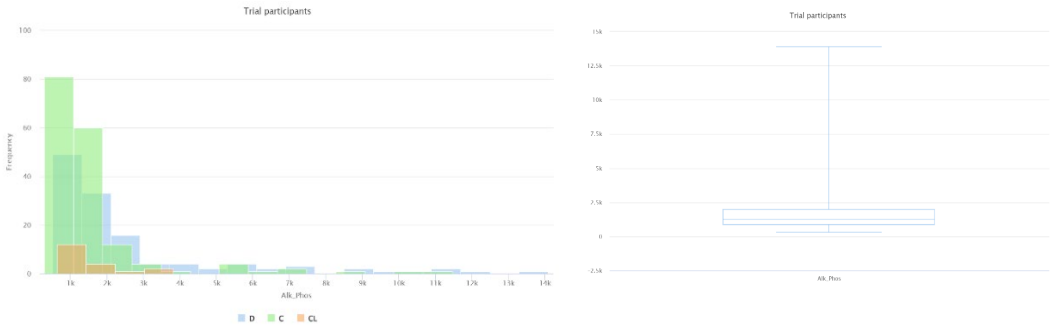


Figure 19. Alk_Phos histogram and boxplot (trial).

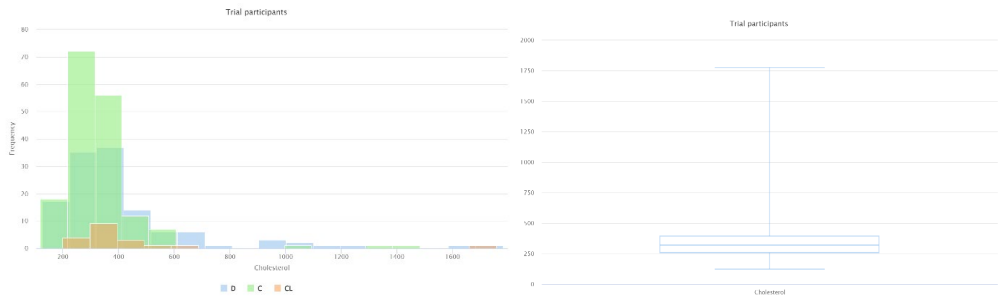


Figure 20. Cholesterol histogram and boxplot (trial).

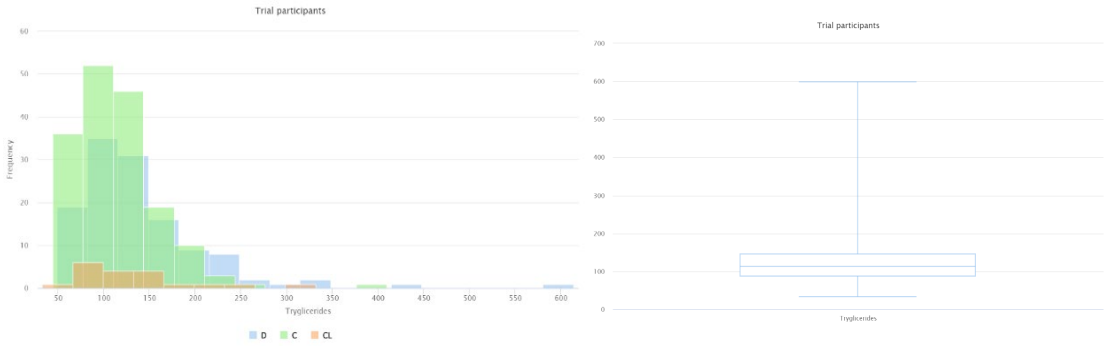


Figure 21. Triglycerides histogram and boxplot (trial).

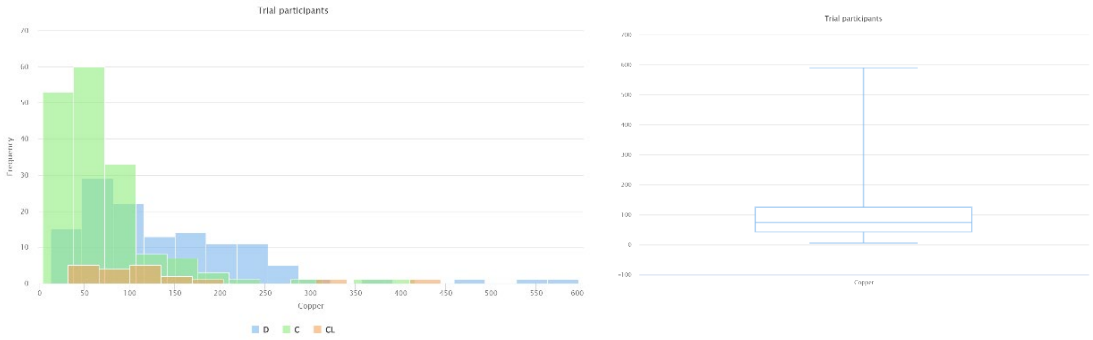


Figure 22. Copper histogram and boxplot (trial).

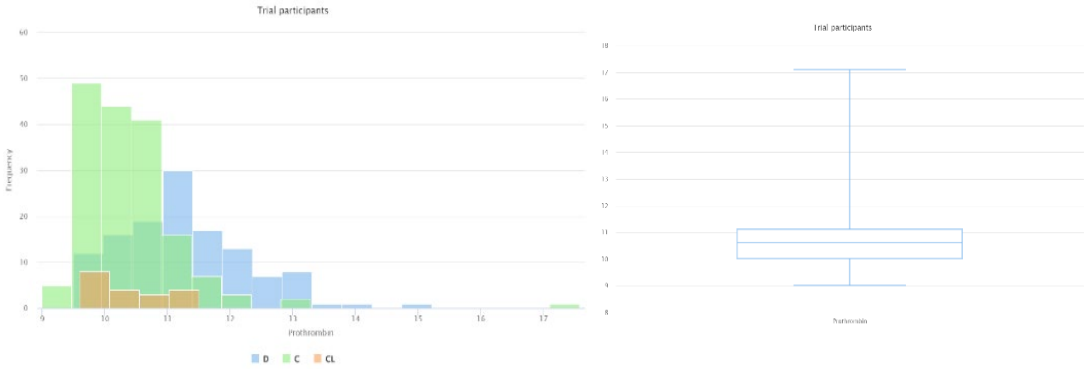


Figure 23. Prothrombin histogram and boxplot (trial).

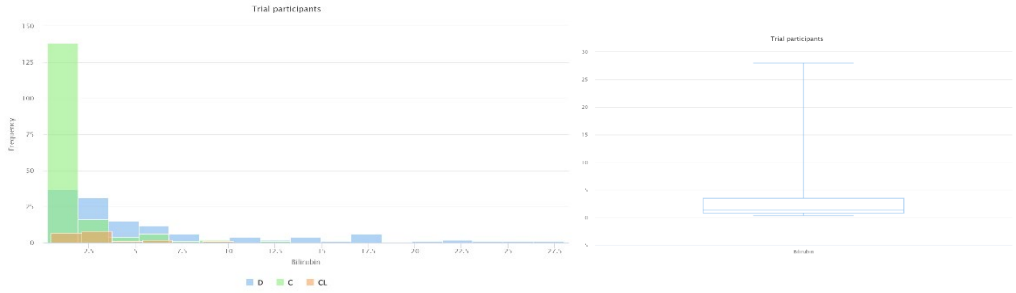


Figure 24. Bilirubin histogram and boxplot (trial).

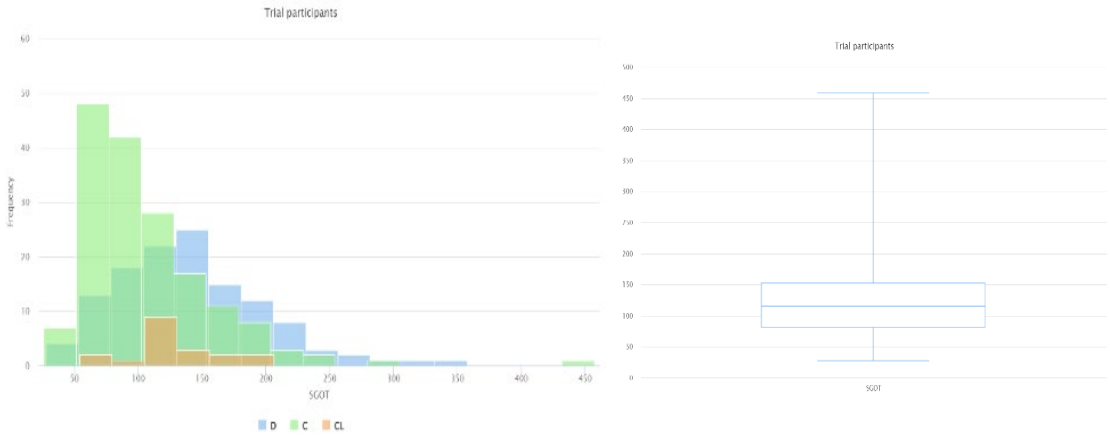


Figure 25. SGOT histogram and boxplot (trial).

In the trial group, there is significant class imbalance observed across several nominal attributes, reflecting disparities in category representation. For the Status attribute, most patients fall under category "C" with 168 instances (approximately 51%), followed by category "D" with 125 instances (around 38%), and only 19 instances (6%) in the "CL" category, highlighting a substantial imbalance. The Sex attribute is heavily skewed toward females, with 276 instances (88%), compared to only 36 male instances (12%). This stark imbalance may affect the reliability of gender-based analyses. For Ascites, most patients are classified as "N" (no ascites) with 288 instances (92%), while only 24 patients (8%) are classified as "Y" (ascites present), showing a strong imbalance in representation. Regarding Edema, 263 instances (84%) indicate "N" (no edema), followed by 29 instances (9%) in category "S" (edema not clinically detectable but present), and only 20 instances (6%) in category "Y" (edema clinically detectable), emphasizing a pronounced skew in favor of "N." The Stage attribute has a more even distribution compared to others, with stage 3 being the most common (120 instances, 38%), followed by stage 4 (109 instances, 35%), stage 2 (67 instances, 21%), and stage 1 (16 instances, 6%). While the representation is less skewed, stage 1 is underrepresented relative to stages 2 ,3 and 4. This class imbalance could impact future analyses and modelling efforts, as underrepresented categories might contribute less to model training or be overshadowed by the majority classes.

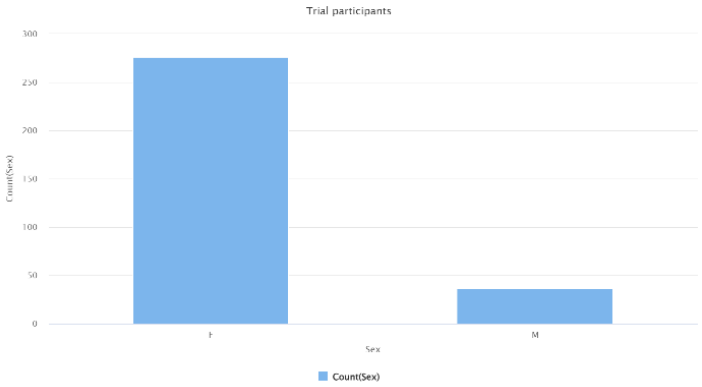


Figure 26. Sex bar column (trial).

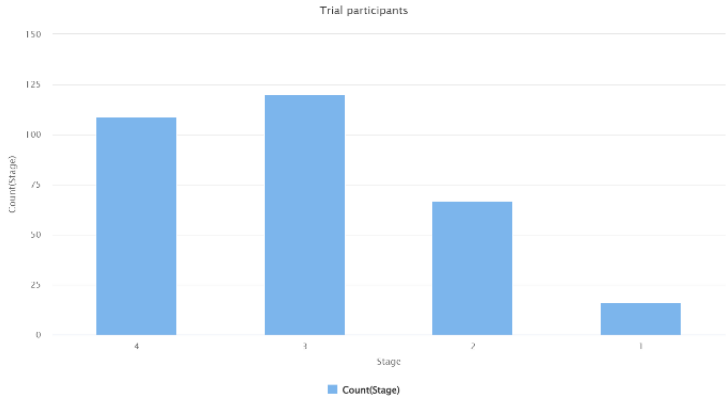


Figure 27. Stage bar column (trial).

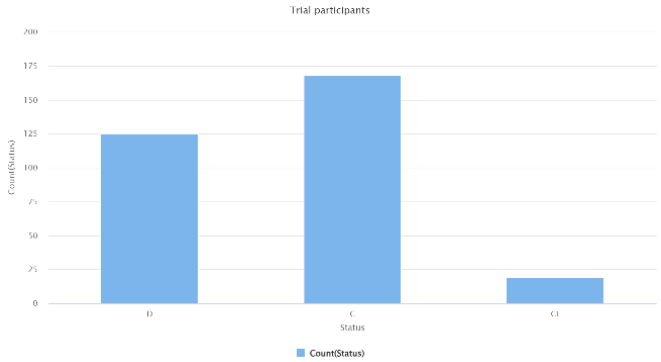


Figure 28. Status bar column (trial).

The overlapping observed in the histograms of both groups, with no noticeable gaps between them, demonstrates that no single attribute distinctly classifies the target variable. This indicates that the Status attribute cannot be reliably separated using individual features in isolation. Instead, the lack of clear boundaries between the classes suggests that a combination of multiple attributes, potentially with nonlinear interactions, will be required to achieve effective classification.

2.2.3 SUMMARY

The dataset presents notable challenges for effective modelling. Numeric attributes such as Age and Albumin approximate normal distributions, with symmetric patterns in histograms and box plots, making them suitable for direct use in models. In contrast, attributes like Bilirubin, Prothrombin, and Cholesterol exhibit significant positive skewness and long tails, reflecting severe clinical conditions. These distributions may distort model performance and require feature transformation techniques explained in later sections. Class imbalances are a critical issue across nominal attributes. For instance, in Status, the "C" category dominates, with "D" and especially "CL" underrepresented, which may bias models toward majority classes.

2.3 Feature Selection

As mentioned before, the non-trial group consists of 100 instances with 10 attributes, while the trial group comprises 312 instances and 19 attributes. For the non-trial group (m=100; n=10), the condition $m \gg n^2$ is not satisfied. Similarly, for the trial group (m=312; n=19) is also not met. This indicates that feature selection (dimensionality reduction) is necessary for both groups to address the imbalance between the number of attributes and instances, ensuring effective modelling. Reducing the number of attributes in both groups will not only enhance computational efficiency but also improve interpretability by focusing on the most relevant features while reducing noise.

Based on the correlation matrices obtained for both the trial and non-trial groups, there are no attributes that exhibit high correlation with each other. This suggests that no features need to be removed solely based on the criterion of multicollinearity.

2.3.1 WEIGHT BY INFORMATION GAIN

Feature selection weighted by information gain uses the same principle as decision trees, where each feature is evaluated as a potential root node. The gain in entropy reduction is computed for every feature, quantifying its contribution to decreasing uncertainty about the target variable. Higher weights signify a substantial contribution to classification and suggest a strong relationship between the feature and the Status.

In the non-trial group, N_Days holds the highest weight (0.317), indicating it is the most significant predictor for the Status attribute. Other features like Bilirubin (0.165), Prothrombin (0.131), and Age (0.120) also demonstrate substantial relevance, suggesting their importance in predicting outcomes. Lower-weight features such as Sex (0.010) and Edema (0.046) offer limited predictive value and may contribute less to the overall classification task.

attribute	weight ↓
N_Days	0.317
Bilirubin	0.165
Prothrombin	0.131
Age	0.120
Platelets	0.100
Stage	0.089
Albumin	0.051
Edema	0.046
Sex	0.010

Figure 29. Attribute weight by information gain (non-trial).

For the trial group, N_Days (0.227) again emerges as the most impactful feature, followed by Bilirubin (0.211) and Prothrombin (0.154). Features like Drug (0.001) and Sex (0.021) rank the lowest, suggesting minimal influence on the classification.

attribute	weight ↓
N_Days	0.227
Bilirubin	0.211
Prothrombin	0.154
Copper	0.142
Albumin	0.087
Ascites	0.087
Hepatomegaly	0.084
SGOT	0.084
Age	0.077
Stage	0.070
Edema	0.069
Alk_Phos	0.063
Platelets	0.047
Cholesterol	0.039
Spiders	0.039
Tryglicerides	0.032
Sex	0.021
Drug	0.001

Figure 30. Attribute weight by information gain (trial).

2.3.2 WEIGHT BY INFORMATION GAIN RATIO

Feature selection weighted by information gain ratio penalizes attributes that produce excessive branching in decision trees, such as IDs, by applying the same entropy-based metrics with adjustments to reduce bias towards overly specific features.

In the non-trial group, the highest weights (0.517) were attributed to features like Bilirubin, Albumin, and Platelets. Features like N_Days, though still significant, have a reduced ranking position in the Weight Information Gain ranking compared to their first position in the Weight Information Gain Ratio ranking due to N_Days having many unique values that result in

decision tree splits creating numerous branches. The attributes with the lowest weights, Edema and Sex, under both methods remain consistent.

attribute	weight ↓
Bilirubin	0.517
Albumin	0.517
Platelets	0.517
N_Days	0.427
Age	0.237
Prothrombin	0.237
Stage	0.096
Edema	0.075
Sex	0.025

Figure 31. Attribute weight by information gain ratio (non-trial).

For the trial group, Triglycerides, N_Days, and Bilirubin emerged as the top features. This demonstrates a shift in focus towards Triglycerides because it may create splits that are more balanced across the decision tree branches, even if the entropy reduction (information gain) is not the highest. The attributes with the lowest weights in the trial group, such as Sex, Drug, and Spiders, continue to rank low under both methods.

attribute	weight ↓
Tryglicerides	0.419
N_Days	0.299
Bilirubin	0.247
Albumin	0.228
Ascites	0.222
Edema	0.201
Prothrombin	0.196
Cholesterol	0.194
Alk_Phos	0.173
Copper	0.166
Platelets	0.163
Age	0.122
Stage	0.107
SGOT	0.092
Hepatomeg...	0.084
Spiders	0.045
Sex	0.040
Drug	0.001

Figure 32. Attribute weight by information gain ratio (trial).

2.3.3 WEIGTH BY CHI SQUARED STATISTICS

Weight by chi-squared statistic measures the relevance of attributes by computing their Chi-squared statistic with respect to the class attribute. It identifies how strongly each attribute is associated with the target variable. Chi-squared assumes that all feature values and the target are categorical and independent.

For the non-trial group, N_Days, Bilirubin, and Platelets have the highest weights, indicating they contribute the most information about the target variable. Their higher scores suggest strong associations with the differences in the Status classes. Albumin, Edema, and Sex have relatively low weights, suggesting weaker relationships with Status.

attribute	weight ↓
N_Days	45.093
Bilirubin	44.572
Platelets	38.878
Albumin	26.511
Age	25.959
Prothrombin	19.193
Stage	14.529
Edema	6.676
Sex	0.892

Figure 33. Attribute weight by chi-squared statistics (non-trial).

For the trial group, N_Days, Bilirubin and Cooper rank highest, which implies their significant role in explaining variations in Status. Cooper's prominence in the chi-squared method suggests it may have a categorical relevance to Status, possibly differentiating patients based on specific frequencies. However, its absence in other methods indicates that its predictive power might depend on interactions with other features. Attributes like Drug, Sex, and Spiders hold lower weights.

attribute	weight ↓
N_Days	98.397
Bilirubin	91.307
Copper	87.273
Prothrombin	75.185
Albumin	52.546
SGOT	47.590
Age	45.342
Stage	39.249
Hepatomegaly	35.656
Edema	33.963
Ascites	33.684
Cholesterol	30.396
Alk_Phos	28.540
Platelets	27.107
Tryglicerides	24.074
Spiders	16.898
Sex	8.936
Drug	0.225

Figure 34. Attribute weight by chi-squared statistics (trial).

2.3.4 WEIGTH BY RELIEF

The weight by relief method emphasizes attributes that effectively differentiate between instances of different classes by evaluating their proximity in feature space. High weights in this context suggest that an attribute can reliably distinguish between close points belonging to different classes, making it a strong predictor. Conversely, negative weights indicate that the attribute might be misleading or irrelevant for classification, as its inclusion could introduce noise or erroneous associations.

In the non-trial group, N_Days, Bilirubin, and Edema have the highest weights. Their strong ranking suggests they are critical in distinguishing between different Status classes. Interestingly, Sex receives a negative weight, indicating it may contribute noise or misleading information to the classification process in this group. This aligns with the idea that it may not have a strong relationship with the target variable in the non-trial group.

attribute	weight ↓
Bilirubin	1.007
Edema	0.806
N_Days	0.561
Age	0.487
Prothrombin	0.371
Albumin	0.349
Platelets	0.130
Stage	0.096
Sex	-0.185

Figure 35. Attribute weight by relief (non-trial).

In the trial group, attributes like Hepatomegaly, N_Days, Drug and Stage dominate the weight rankings.

attribute	weight ↓
Hepatomegaly	0.886
N_Days	0.562
Drug	0.420
Stage	0.339
Spiders	0.301
Bilirubin	0.289
SGOT	0.204
Copper	0.204
Prothrombin	0.194
Age	0.177
Sex	0.109
Alk_Phos	0.094
Ascites	0.080
Cholesterol	0.080
Albumin	0.068
Tryglicerides	0.028
Platelets	-0.008
Edema	-0.113

Figure 36. Attribute weight by relief (trial).

2.3.5 FORWARD SELECTION

The two following tables presented summarize the results of forward selection for both groups, a feature selection method that begins with no features in the model and adds attributes one by one based on their contribution to improving the model's predictive power. The process continues until the addition of new features no longer significantly enhances the model's performance.

In the right table, representing the non-trial group, the attributes Sex and Prothrombin were selected by the forward selection process because they may have a significant contribution

to predicting the target variable. In the left table, for the trial group, only the attribute N_Days was selected.

attribute	weight
Sex	1
Prothrombin	1
Drug	0
Ascites	0
Hepatomegaly	0
Spiders	0
Edema	0
Stage	0
N_Days	0
Age	0
Bilirubin	0
Cholesterol	0
Albumin	0
Copper	0
Alk_Phos	0
SGOT	0
Tryglicerides	0
Platelets	0

attribute	weight
N_Days	1
Sex	0
Edema	0
Stage	0
Age	0
Bilirubin	0
Albumin	0
Platelets	0
Prothrombin	0

Figure 37. Forward selection for trial group (left) and non-trial group (right).

2.3.6 CONCLUSIONS

In both the trial and non-trial groups, the attribute N_Days, although selected by the majority of feature selection methods and consistently receiving the highest weight, should not be included in the predictive model. This attribute directly correlates with the Status, as fewer days in the study are strongly associated with the "Death" class. As a result, its inclusion would make the prediction process redundant, as it essentially encodes the outcome it seeks to predict. Furthermore, from a practical standpoint, N_Days is neither actionable nor meaningful as a feature. Its value is determined by the patient's survival duration during the study, making it a retrospective measure rather than one that can be used at the time of prediction. Including it in the model would therefore rely on information that is unavailable during real-time decision-making, undermining the purpose of predictive modelling. Lastly, the inclusion of N_Days would likely compromise the generalizability of the model. Relying on temporally derived attributes like this may result in a model that performs well on the training dataset but lacks applicability in real-world scenarios. To develop a robust and interpretable predictive model, it is essential to exclude N_Days and focus on features that provide meaningful insights into the target variable without relying on retrospective information.

In addition, for the trial group, the Sex attribute has been removed due to its consistently low weight across most feature selection methods (although selected by the forward selection).

Furthermore, the distribution of classes across sexes does not reveal significant differentiation. While there are generally more females than males in the dataset, the class proportions within each sex are fairly similar. Moreover, removing the Sex attribute helps to address potential issues related to class imbalance, ensuring that future models are less prone to biases or skewed interpretations. Although the Drug attribute also exhibits low weights in most methods except for Relief, its inclusion is retained for now, as the priority is to eliminate attributes with minimal predictive relevance and potentially problematic implications, such as Sex.

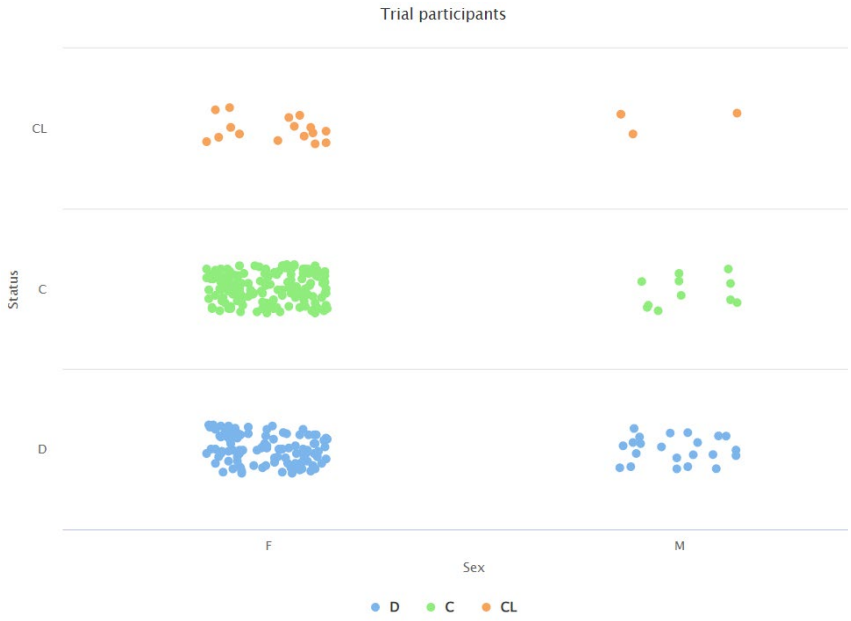


Figure 38. Sex scatter (trial).

With the removal of less relevant attributes, including N_Days and Sex, both the trial and non-trial groups now satisfy the condition $m \gg n^2$. Meeting this condition improves the reliability of the models by reducing the risk of overfitting.

2.4 Feature Transformation

To address the issue of long tails in the data distributions, a non-linear transformation using the natural logarithm has been applied specifically to the attributes identified in Section 2.2.2 as having pronounced long-tail behaviour. This transformation effectively compresses extreme values, reducing skewness and making the distributions more symmetric. By applying this transformation selectively to these attributes, the data becomes more suitable for modelling, particularly for algorithms that perform better under assumptions of normality or reduced variance. Additionally, this approach mitigates the influence of outliers, improving both the robustness and interpretability of the models. Below are the histograms displaying the distributions of the attributes after applying the natural logarithm transformation.

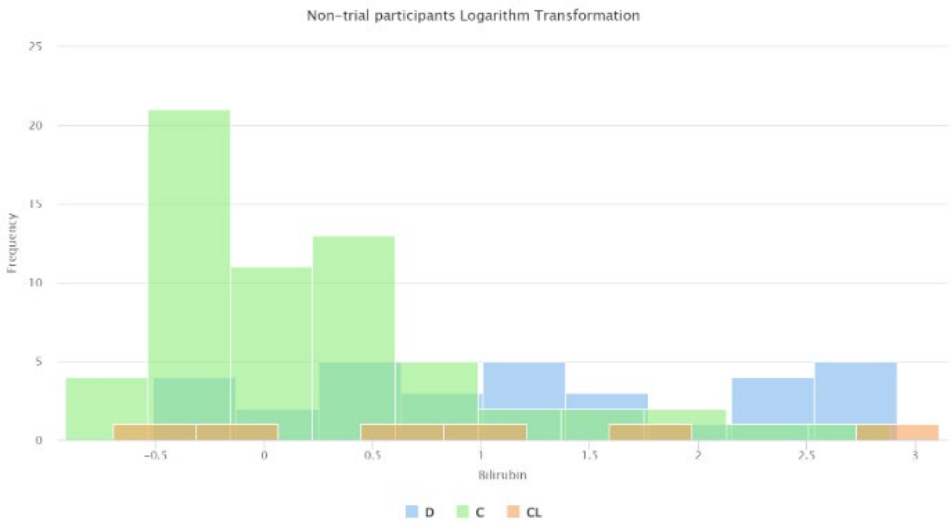


Figure 39. Bilirubin histogram after logarithm transformation (non-trial).

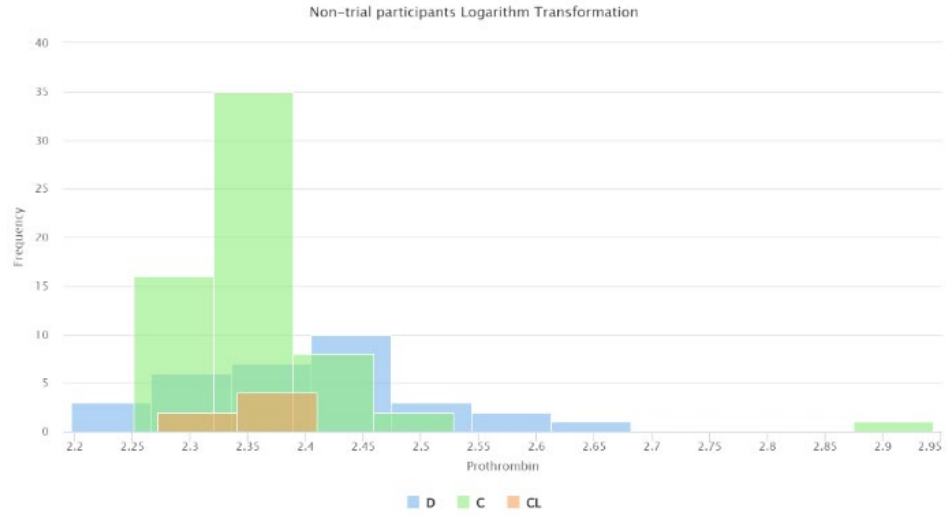


Figure 40. Prothrombin histogram after logarithm transformation (non-trial).

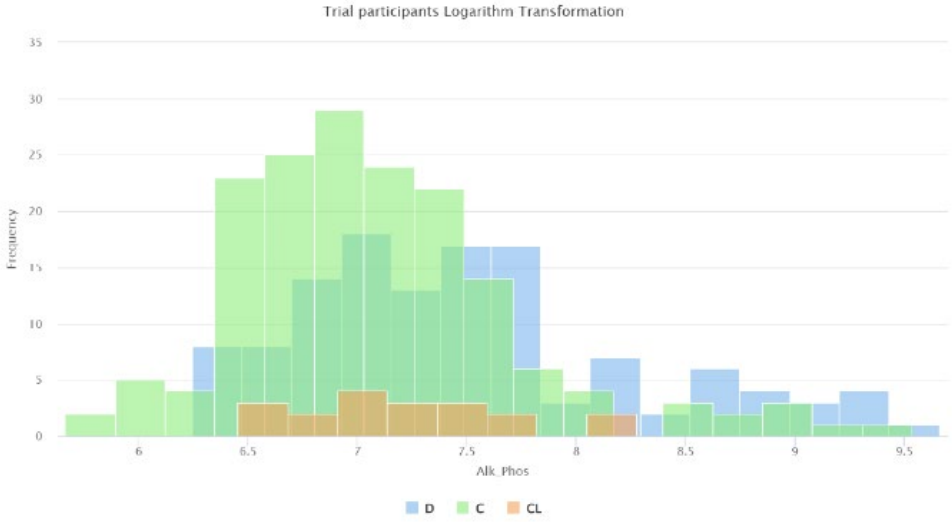


Figure 41. Alk_Phos histogram after logarithm transformation (trial).

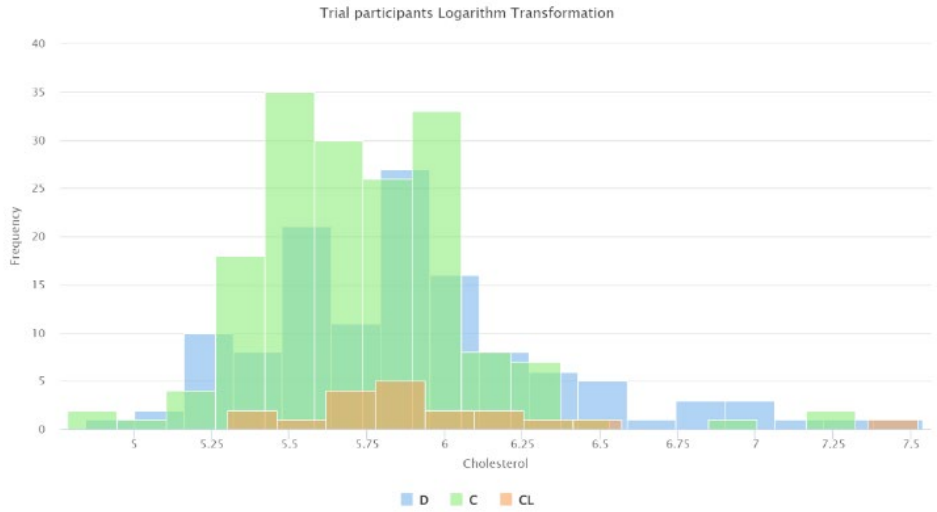


Figure 42. Cholesterol histogram after logarithm transformation (trial).

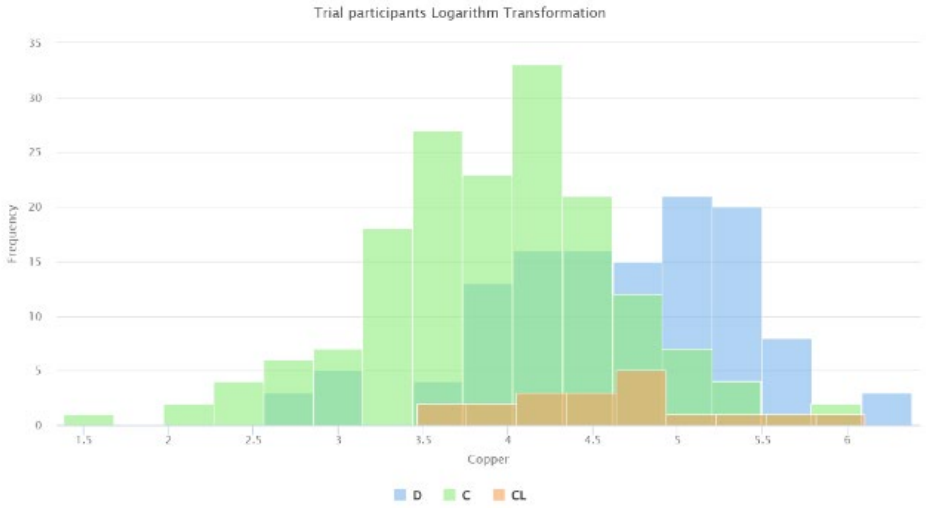


Figure 43. Copper histogram after logarithm transformation (trial).

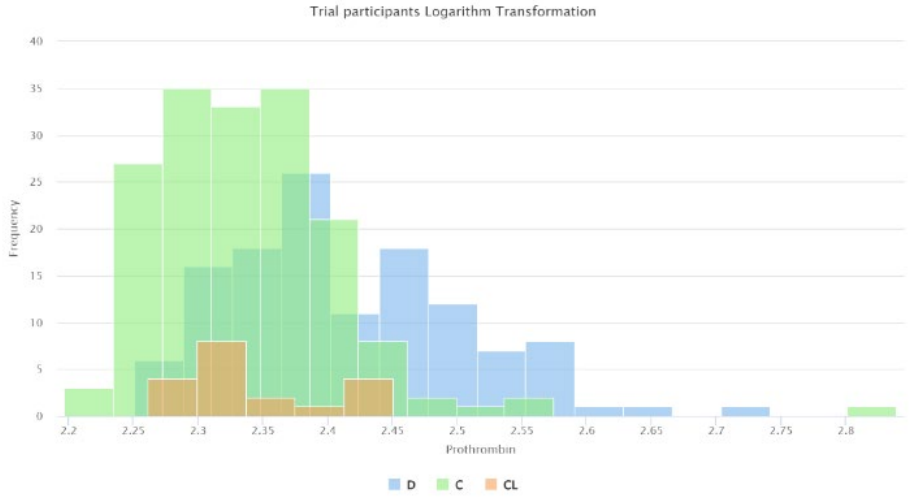


Figure 44. Prothrombin histogram after logarithm transformation (trial).

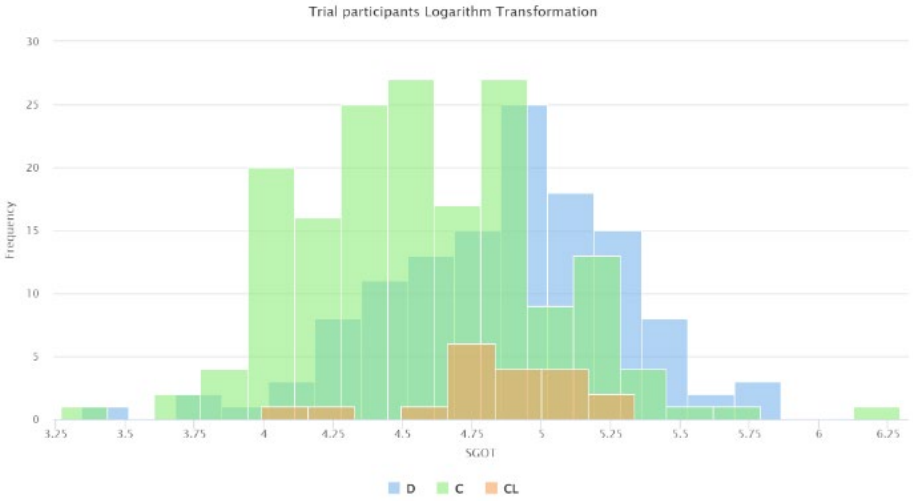


Figure 45. SGOT histogram after logarithm transformation (trial).

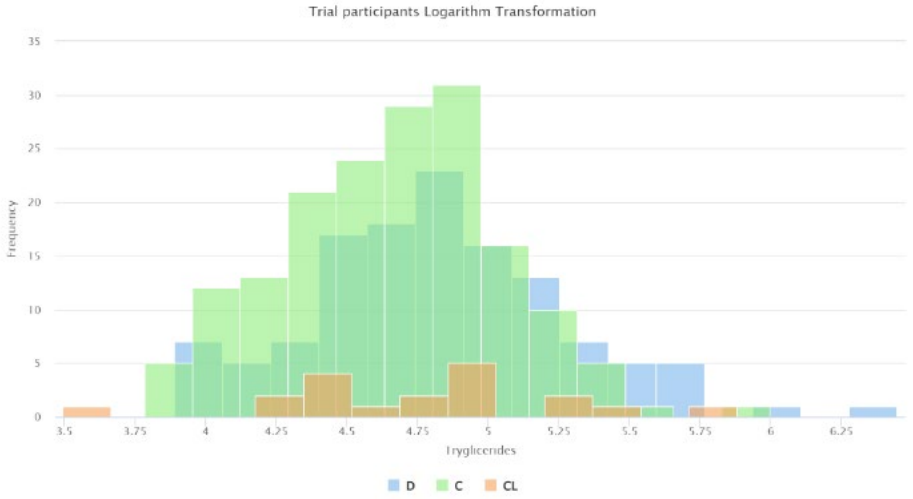


Figure 46. Triglycerides histogram after logarithm transformation (trial).

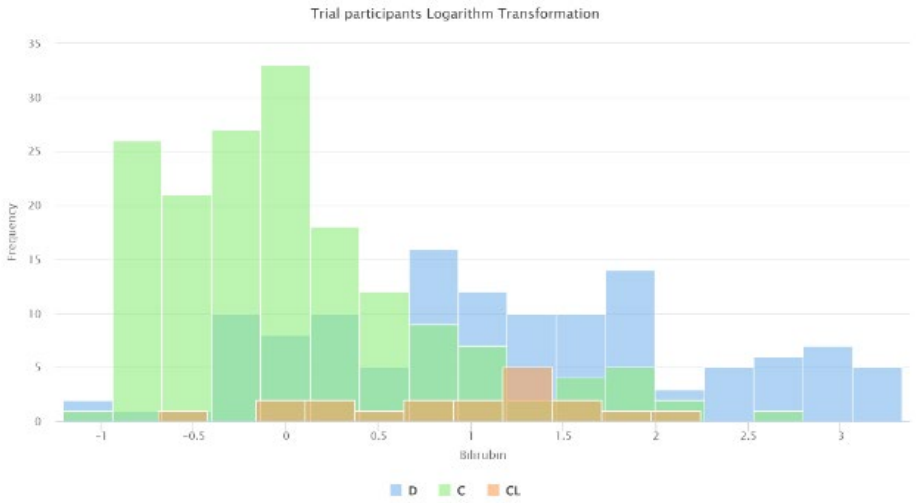


Figure 47. Bilirubin histogram after logarithm transformation (trial).

After applying the natural logarithm transformation, the numeric attributes were normalized to standardize their ranges. Before normalization, the ranges of the numeric attributes varied significantly, with some attributes exhibiting extreme values due to long tails or skewness (e.g., Bilirubin and Prothrombin). The following two figures represent the boxplots of the non-normalized groups, illustrating how the ranges of the attributes are highly diverse. This clearly demonstrates the need for normalization.

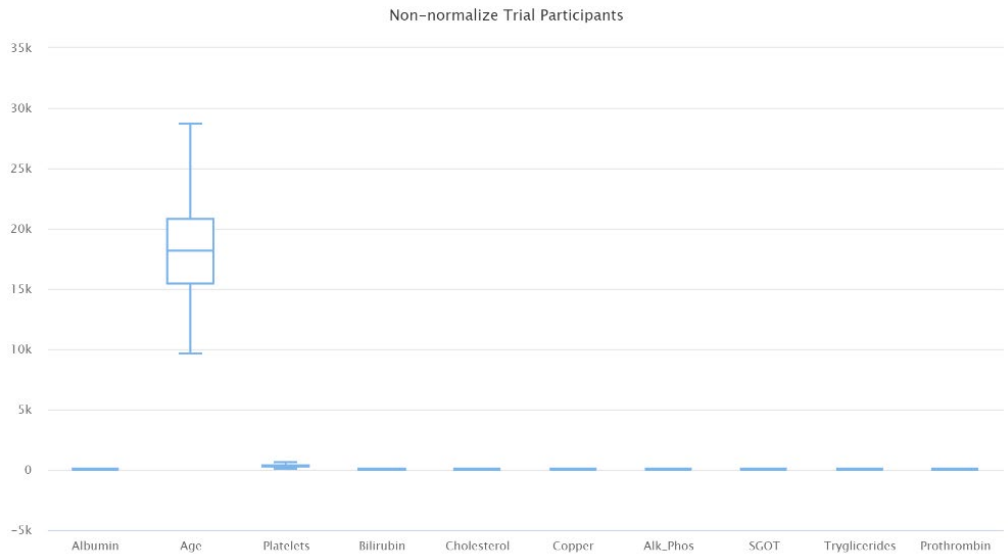


Figure 48. Boxplot trial group.

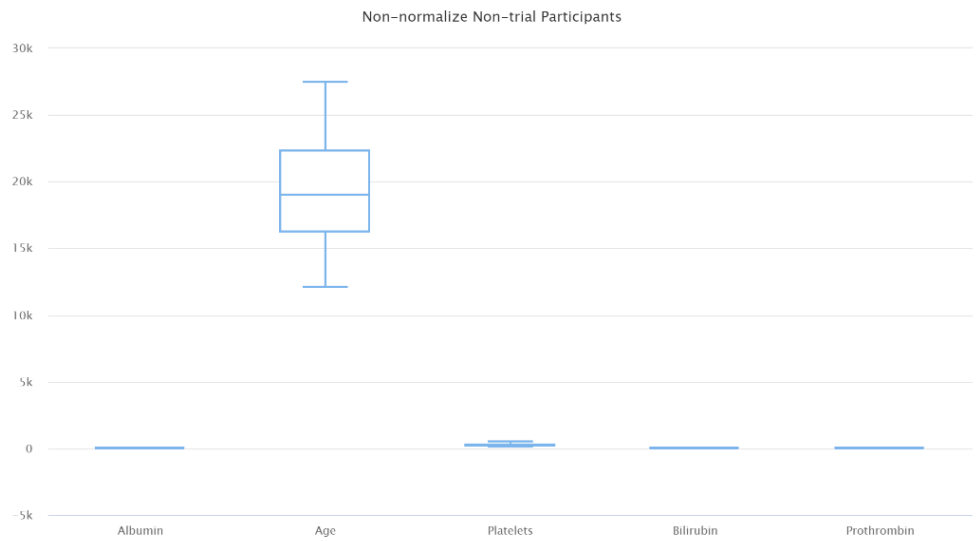


Figure 49. Boxplot non-trial group.

After normalization, the attributes have been rescaled to a consistent range, centred around 0 with standard deviations equal to 1. This standardization eliminates the disparity in scales, making the attributes more comparable and ensuring that no single attribute dominates

the model due to its magnitude. The resulting box plots clearly demonstrate the uniformity in ranges across the normalized attributes for both the trial and non-trial groups.

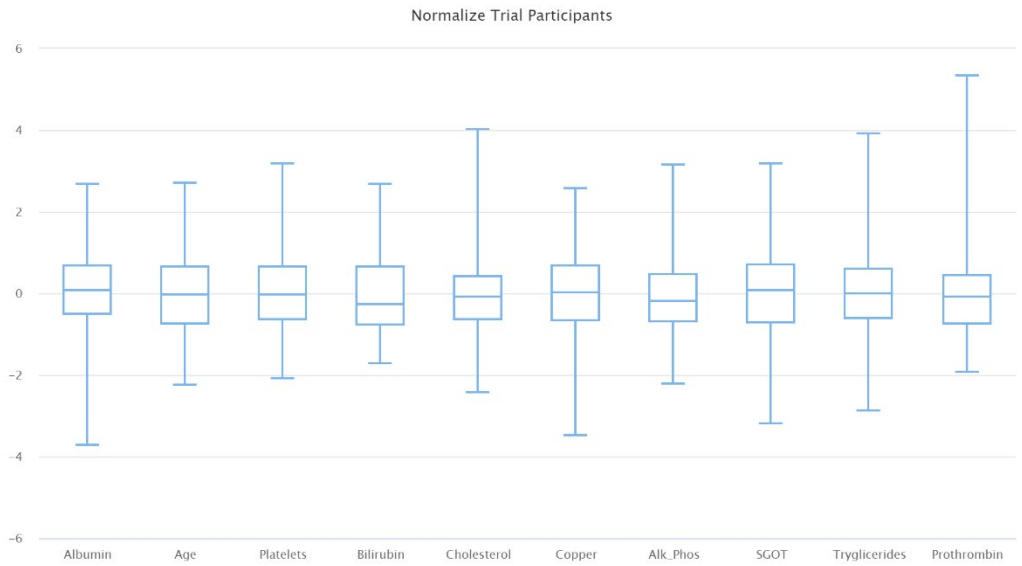


Figure 50. Normalized boxplot trial group.

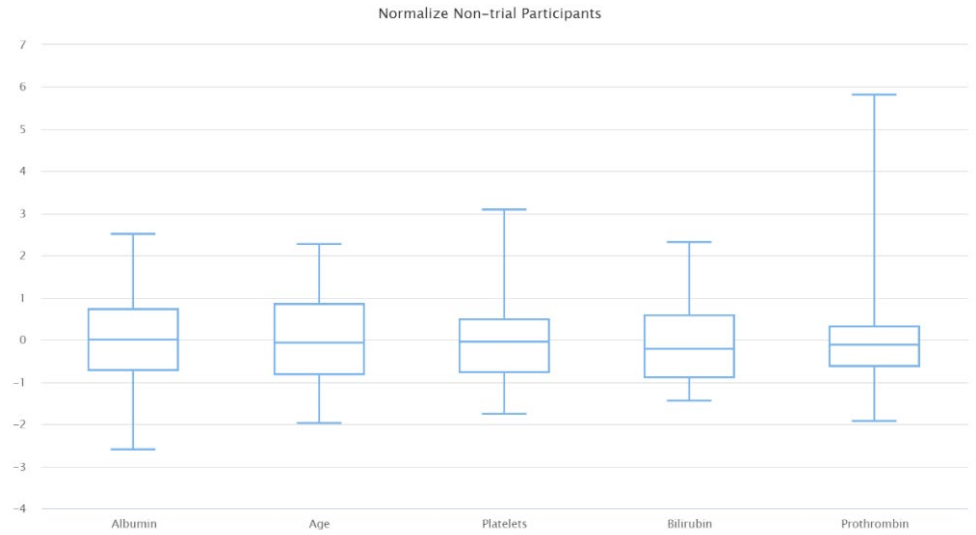


Figure 51. Normalized boxplot non-trial group.

After applying normalization and log transformations to the data, the outcomes of the filter and wrapper feature selection methods remain largely unchanged. Similarly, the correlation structure between attributes is practically preserved, indicating that the transformations, while addressing issues like skewness and scale differences, did not significantly alter the relationships or rankings of features in terms of their relevance to the target variable.

Model Training and Evaluation with Altair AI Studio

For our classification task, different predictive models will be developed to analyse both the normalized and logarithm-transformed datasets. These models will be evaluated for their ability to predict survival outcomes for patients with cirrhosis. The following predictive models will be constructed, which are suitable for the type of data in our cirrhosis dataset:

- Decision tree
- Rule-based model
- Naive Bayes
- Bayesian network
- Neural networks
- Deep learning neural network

3.1 Training Models

Even though the data has been pre-processed separately, training models for each independent group remains essential due to the differences in their characteristics, which can influence model behaviour. However, there is significant potential in merging the data from both groups. The key here is to identify the common attributes between the two groups. Models trained on the trial group specialize in clinical-specific data, while those trained on the non-trial dataset are more focused on what actually happens in a real-world hospital setting and, by combining both, models learn more robust patterns. Therefore, to achieve better results and adapt to the different realities of the data, models will be trained for the separate groups (trial and non-trial), as well as for both combined using the common attributes from both datasets.

A 10-fold cross-validation approach will be applied to train and test the models, ensuring robustness and reducing variability in performance due to random data splits. In this process, each model will be trained under two criteria. The first criterion, accuracy-based training, aims to maximize the overall proportion of correct predictions, while the second, cost-based training, focuses on minimizing errors weighted by a predefined cost matrix to address the clinical importance of different misclassification types. Additionally, the cross-validation provides a confidence interval, offering insight into the variability and reliability of the model's performance metrics across different folds.

The cost matrix used for the Status classification problem is as follows, where the rows represent the predicted classes, and the columns represent the actual classes:

Predicted \ Actual	D (Death)	C (Censored)	CL (Liver Transplant)
D (Death)	0	5	10
C (Censored)	10	0	8
CL (Liver Transplant)	15	5	0

Table 1. Cost matrix.

The cost matrix is based on clinical priorities and potential implications, as no specific economic data regarding misclassifications could be found in scientific literature. These values are therefore arbitrary but reflect the severity of outcomes associated with each misclassification. Misclassifying a censored patient as dead incurs a moderate cost due to incorrect documentation of the outcome. While it does not immediately harm the patient, it may lead to inappropriate statistical conclusions or care mismanagement. Predicting a transplant for a deceased patient incurs a higher penalty, as it could waste resources by unnecessarily reserving a transplant, highlighting a critical misunderstanding of the patient's status. Misclassifying death as censored is a serious error, as it underestimates the severity of the outcome, skews survival analysis, and misrepresents mortality data. Assigning transplant status to a censored patient carries a moderate penalty, reflecting poor judgment but less severity than misclassifying death since the patient is alive and could recover without a transplant. The highest penalty is assigned to misclassifying a deceased patient as needing a transplant, as it wastes critical and scarce resources, representing a significant decision-making error. Misclassifying a censored patient as needing a transplant is considered a moderate error, as it may lead to overtreatment or unnecessary allocation of resources but is less severe than misclassifying death. These costs are designed to prioritize the accurate identification of severe outcomes, reduce impactful errors, and ensure the model aligns with clinical priorities and patient safety considerations, particularly for cases involving death and liver transplants.

3.1.1 TRAINING SEPARATE GROUPS: TRIAL AND NON-TRIAL DATA

A. Accuracy-trained models

Decision tree

In the decision tree model for the trial group, Bilirubin and Prothrombin emerge as the most influential features, driving the primary class splits. These features are pivotal in separating the different classes and significantly contribute to the tree's decisions. By analysing the decision tree, we observe that the tree first utilizes Bilirubin and Prothrombin to make key decisions that help in classifying the instances into the respective classes. After removing N_Days in the preprocessing step, the focus of the model shifts more strongly towards Bilirubin and Prothrombin in the early splits. These features are highly effective in creating clear separations between classes. While other features like Triglycerides show up as important according to Gain Ratio, the tree-building algorithm prioritizes features that provide the greatest immediate

reduction in entropy. As a result, Triglycerides, despite its value for creating balanced splits, does not make it into the final splits of the tree, demonstrating the algorithm's preference for features that provide a more immediate and noticeable reduction in uncertainty.

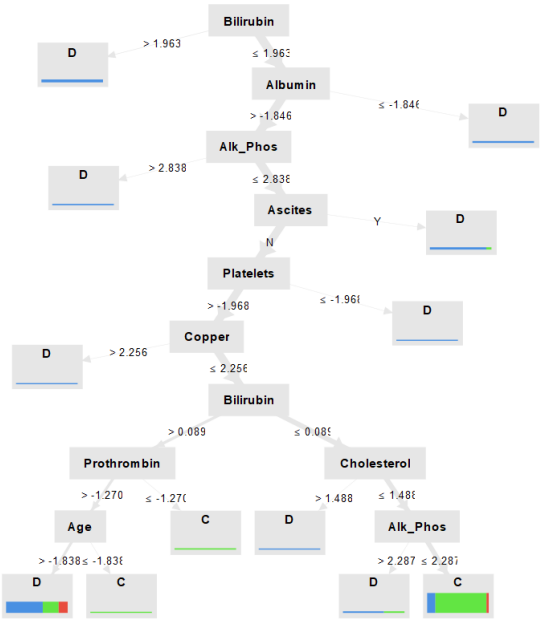


Figure 52. Decision tree trained based on accuracy (trial).

In the non-trial group, the decision tree model illustrates a distinct hierarchy in feature importance with Bilirubin, Albumin, and Platelets dominating the top of the decision splits. This configuration reflects their strong discriminative power in classifying the status of individuals in the non-trial context. Notably, Prothrombin and Age emerge as important features in this group, a trend not observed in the trial group. However, information gain also predicted Prothrombin and Age as important features for classification, aligning with their positions in the decision tree.

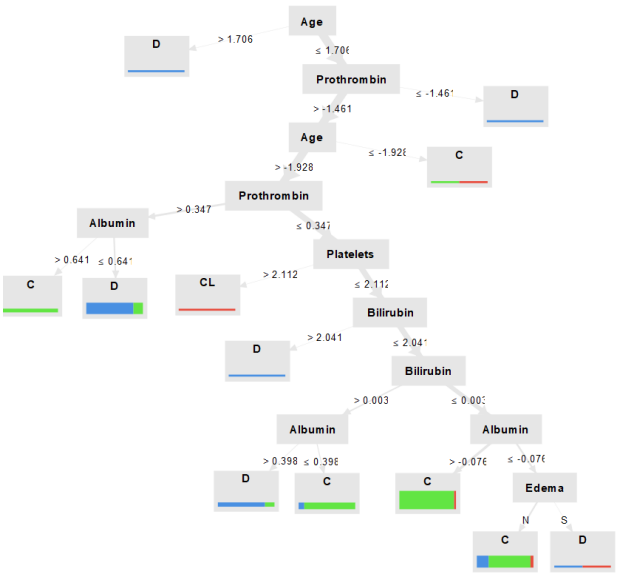


Figure 53. Decision tree trained based on accuracy (non-trial).

Rule-based model

The rule-based model reveals the importance of features like Bilirubin, Platelets, and Albumin in classifying cases for the trial group. The model captures complex, threshold-based relationships and provides a high level of predictive accuracy. Features such as Bilirubin and Platelets appear to play a decisive role in predicting the disease, particularly for distinguishing Censored and Death statuses. These findings align well with the results from the decision tree and other models in which Bilirubin and Platelets were deemed important features.

RuleModel

```
if Bilirubin ≤ -0.016 and Bilirubin ≤ -0.839 and Platelets ≤ 0.195 then C  (0 / 38 / 0)
if Bilirubin ≤ 0.089 and Prothrombin ≤ 0.280 and Alk_Phos ≤ 0.271 and Copper > -1.343 then C  (5 / 63 / 3)
if Prothrombin > 0.280 and Bilirubin > 0.089 and Age > -1.272 then D  (58 / 3 / 1)
if Copper ≤ 0.316 and Age ≤ 0.025 and Cholesterol ≤ -0.031 then C  (1 / 17 / 0)
if Age > 0.025 and Albumin ≤ -0.524 and Platelets ≤ 0.043 then D  (12 / 1 / 0)
if Age > -0.588 and Stage = 3.0 and Spiders = N then D  (16 / 3 / 0)
if Copper ≤ 0.316 and Hepatomegaly = N and Cholesterol > 0.113 then C  (0 / 14 / 2)
if SGOT > 0.688 and SGOT ≤ 1.108 and Albumin > -1.417 then D  (9 / 0 / 0)
if Prothrombin > 0.071 and Bilirubin > -0.035 and Alk_Phos > -0.702 then C  (0 / 7 / 0)
if Age > 0.913 and Platelets ≤ 0.743 then D  (7 / 1 / 0)
if Age > 0.584 then C  (0 / 7 / 0)
if Alk_Phos > 1.413 and Albumin > -0.357 and SGOT ≤ 1.836 then D  (7 / 0 / 0)
if SGOT ≤ -0.068 and Copper > 0.423 then C  (0 / 6 / 0)
if Alk_Phos ≤ -0.507 then CL  (1 / 0 / 5)
if Platelets > 0.606 and Spiders = N then D  (4 / 0 / 1)
if Drug = Placebo and Alk_Phos > -0.244 then C  (0 / 7 / 1)
if Age ≤ -0.813 and Albumin ≤ 0.691 then CL  (0 / 0 / 5)
if Age ≤ 0.457 then D  (5 / 0 / 0)
if Age > 0.565 then C  (0 / 1 / 0)
else CL  (0 / 0 / 0)

correct: 288 out of 311 training examples.
```

Figure 54. Rule-based model trained based on accuracy (trial).

For the non-trial group, the rule-based model highlights Bilirubin, Albumin, and Platelets as key features for classification. The rules show how specific thresholds for these features lead to predictions of either Censored (C) or Death (D). However, it does not effectively classify Transplant (CL), which could be due to the class's low representation in the training data. The model also demonstrates how complex feature interactions (such as Bilirubin and Edema, or Prothrombin and Platelets) can effectively separate the different outcome classes in the non-trial group.

RuleModel

```
if Bilirubin ≤ 0.395 and Edema = N and Albumin > -0.573 then C  (3 / 41 / 2)
if Prothrombin > 0.347 and Platelets > -0.350 then D  (8 / 0 / 0)
if Stage = 3.0 and Platelets ≤ 1.363 then C  (3 / 11 / 0)
else D  (16 / 9 / 3)

correct: 76 out of 96 training examples.
```

Figure 55. Rule-based model trained based on accuracy (non-trial).

Naive Bayes

Not all Gaussian distributions of all the attributes are going to be discussed, only those considered most important for classification.

In the trial group, Bilirubin and Prothrombin emerge as the most influential features for classifying the disease status, particularly for distinguishing class D (Disease) from class C (Censored). These features exhibit clear separation, with Bilirubin being particularly important for predicting class D, as high Bilirubin levels are strongly associated with liver dysfunction. Similarly, Albumin plays a crucial role in differentiating between these classes, with patients in class D showing lower levels of Albumin, which indicates a more severe stage of liver disease.

While Platelets and Prothrombin contribute to the classification, their predictive power is moderate, as their distributions show some overlap between classes, particularly class C and class D, which limits their ability to clearly differentiate between these groups. The Age feature also shows a clear distinction between classes D and C, with the class D being associated with older ages. However, the model faces challenges with class CL (Transplant) due to the limited number of examples and the overlap of features such as Age and Albumin with classes C and D. The smaller differences in feature distributions for class CL hinder the model's ability to predict this class accurately.

Overall, the Gaussian distributions for features like Bilirubin and Albumin display well-defined peaks for each class, facilitating the model's ability to separate the classes effectively. However, the model's ability to predict class CL is diminished due to the scarcity of data and the less distinct feature distributions for this class.

SimpleDistribution

```
Distribution model for label attribute Status


Class D (0.401)
17 distributions

Class C (0.538)
17 distributions

Class CL (0.061)
17 distributions
```

Figure 56. Naive Bayes trained based on accuracy (trial).

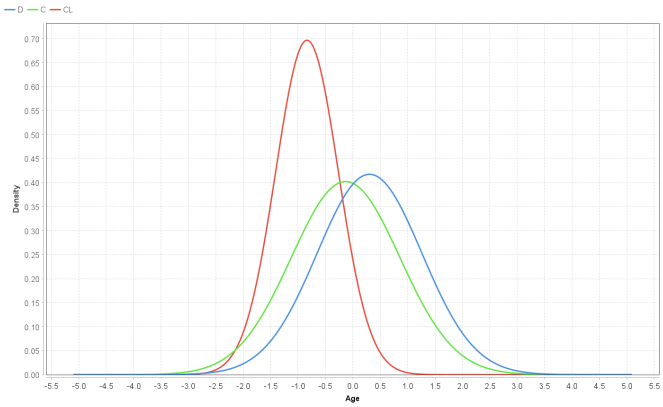


Figure 57. Naive bayes for Age trained based on accuracy (trial).

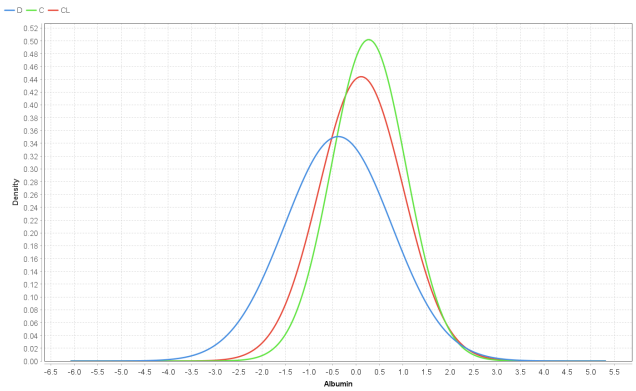


Figure 58. Naive Bayes for Albumin trained based on accuracy (trial).

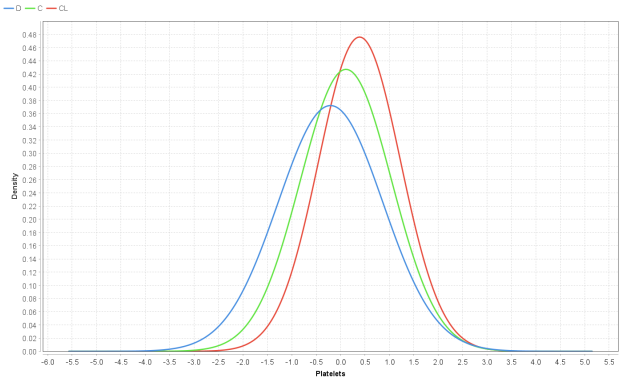


Figure 59. Naive Bayes for Platelets trained based on accuracy (trial).

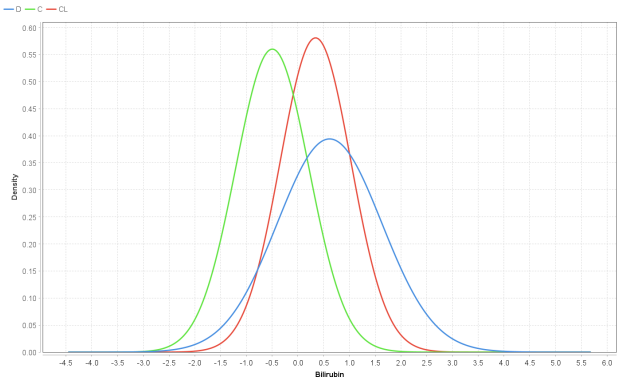


Figure 60. Naive Bayes for Bilirubin trained based on accuracy (trial).

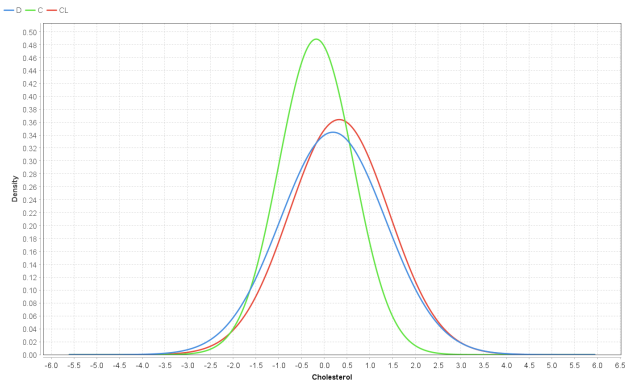


Figure 61. Naive Bayes for Cholesterol trained based on accuracy (trial).

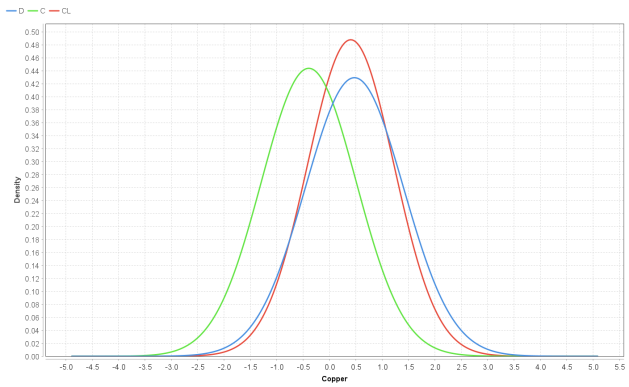


Figure 62. Naive Bayes for Copper trained based on accuracy (trial).

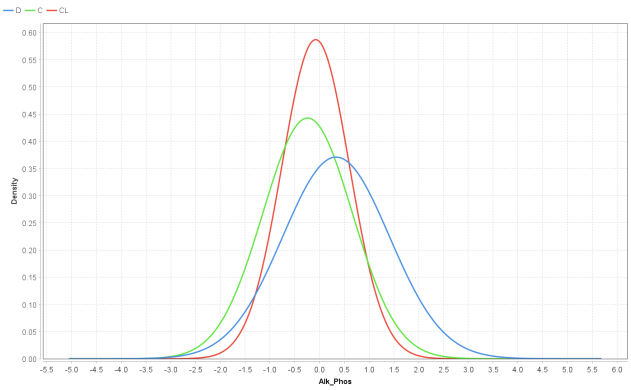


Figure 63. Naive Bayes for Alk_Phos trained based on accuracy (trial).

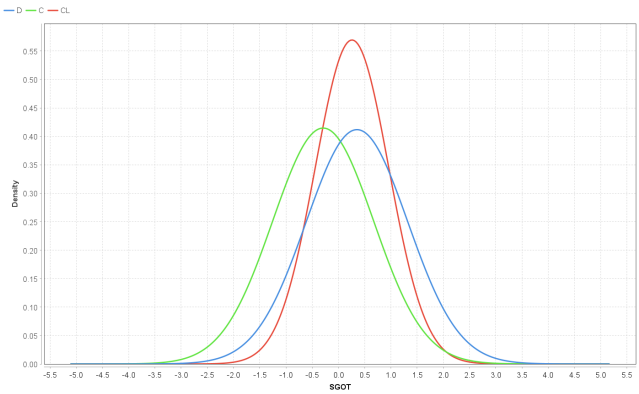


Figure 64. Naive Bayes for SGOT trained based on accuracy (trial).

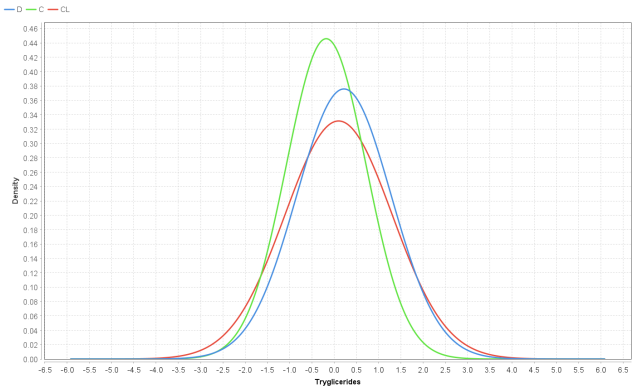


Figure 65. Naive Bayes for Triglycerides trained based on accuracy (trial).

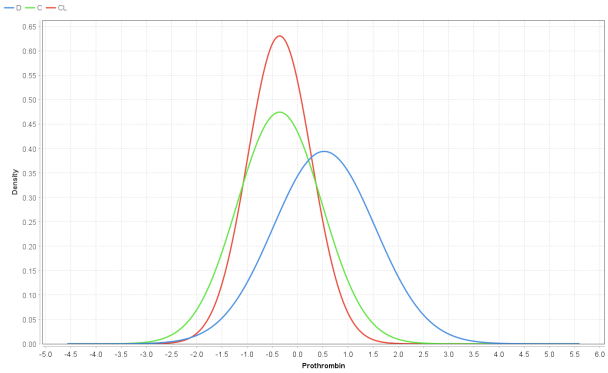


Figure 66. Naive Bayes for Prothrombin trained based on accuracy (trial).

While the Naive Bayes model in the non-trial group demonstrates effective class separation based on key features such as Bilirubin, Albumin, and Prothrombin, some features, such as Platelets and Age, show considerable overlap, reducing their ability to clearly distinguish between certain classes. Despite this, the model benefits from the probabilistic properties of these features, which help it estimate class probabilities effectively, especially for class D. The class CL, however, faces challenges in accurate prediction due to limited data and the overlapping distributions of certain features, such as Age and Albumin, with other classes.

SimpleDistribution

```
Distribution model for label attribute Status


Class D (0.320)
8 distributions

Class C (0.620)
8 distributions

Class CL (0.060)
8 distributions
```

Figure 67. Naive Bayes trained based on accuracy (non-trial).

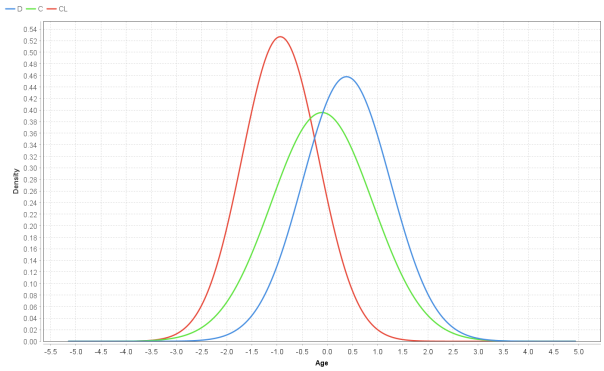


Figure 68. Naive Bayes for Age trained based on accuracy (non-trial).

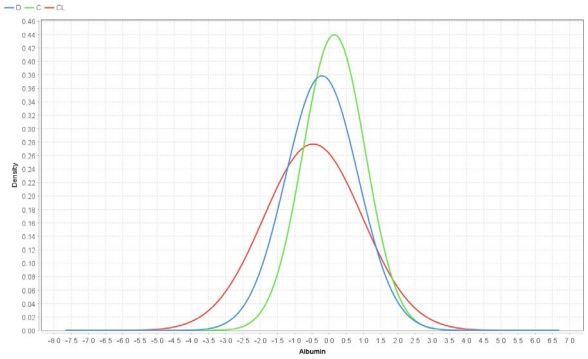


Figure 69. Naive Bayes for Albumin trained based on accuracy (non-trial).

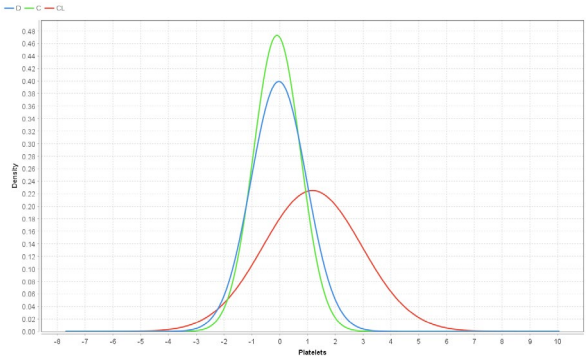


Figure 70. Naive Bayes for Platelets trained based on accuracy (non-trial).

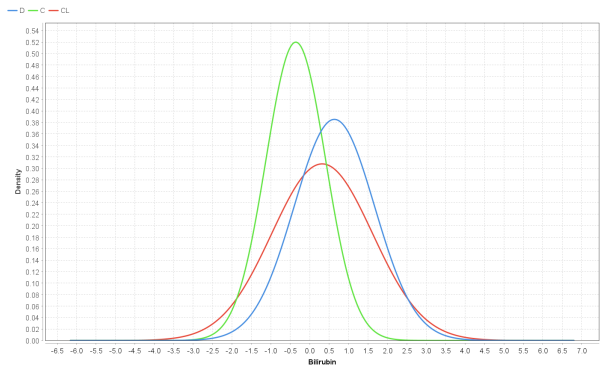


Figure 71. Naive Bayes for Bilirubin trained based on accuracy (non-trial).

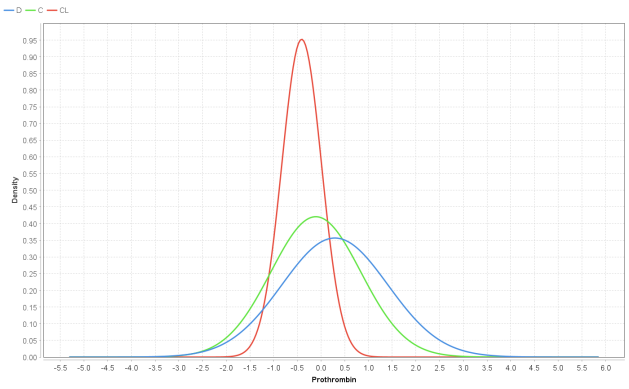


Figure 72. Naive Bayes for Prothrombin trained based on accuracy (non-trial).

Bayesian Network

The Bayesian network model for the trial group highlights the importance of key features such as Age and Albumin. Age influences both SGOT and Platelets, while Albumin is linked to Edema, suggesting that both age and albumin levels are critical for predicting various aspects of cirrhosis disease. Bilirubin emerges as another influential feature with strong connections to SGOT, Copper, and Hepatomegaly, underlining its role in identifying liver dysfunction, particularly through SGOT and Hepatomegaly statuses. Prothrombin and Platelets also play significant roles, with Prothrombin helping to predict the disease stage, reinforcing its importance in liver function assessment.

W-BayesNet

```
Bayes Network Classifier
Using ADTree
#attributes=18 #classindex=17
Network structure (nodes followed by parents)
Age(2): Status SGOT
Albumin(2): Status Edema
Platelets(1): Status Age
Bilirubin(2): Status SGOT
Cholesterol(1): Status Age
Copper(2): Status Bilirubin
Alk_Phos(2): Status Age
SGOT(2): Status
Tryglicerides(1): Status Age
Prothrombin(2): Status Stage
Drug(3): Status Platelets
Sex(2): Status Spiders
Ascites(3): Status Edema
Hepatomegaly(3): Status Bilirubin
Spiders(3): Status Bilirubin
Edema(3): Status Prothrombin
Stage(4): Status Hepatomegaly
Status(3):
LogScore Bayes: -2863.6069451649664
LogScore BDeu: -3236.180158898105
LogScore MDL: -3230.8253808890013
LogScore ENTROPY: -2811.586148178909
LogScore AIC: -2957.586148178909
```

Figure 73. Bayesian network based on accuracy (trial).

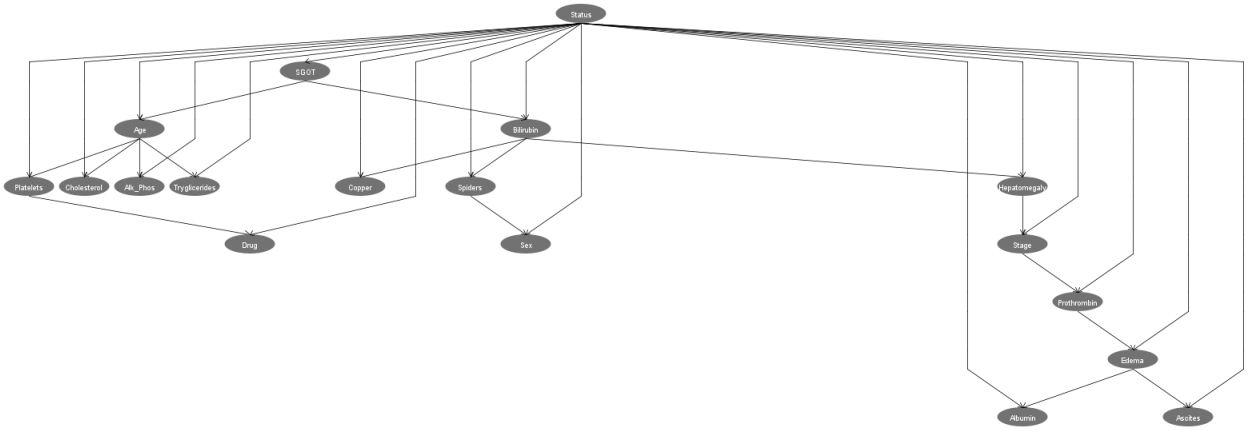


Figure 74. Bayesian network trained based on accuracy (trial).

For the non-trial, the structure of the Bayesian network reveals how the features interconnect to influence the disease status classification. Age significantly impacts Albumin, Platelets, Bilirubin, and Sex, while Prothrombin plays a vital role in determining Stage and, ultimately, the Status. This reflects the complex relationships between features that can be leveraged to predict the disease status accurately.

W-BayesNet

```
Bayes Network Classifier
Using ADTree
#attributes=9 #classindex=8
Network structure (nodes followed by parents)
Age(1): Status Prothrombin
Albumin(1): Status Age
Platelets(1): Status Age
Bilirubin(2): Status Age
Prothrombin(2): Status Stage
Sex(2): Status Age
Edema(3): Status Age
Stage(4): Status
Status(3):
LogScore Bayes: -402.60815873229046
LogScore BDeu: -474.52940386539416
LogScore MDL: -473.3192884944763
LogScore ENTROPY: -392.7288102396847
LogScore AIC: -427.7288102396847
```

Figure 75. Bayesian network based on accuracy (non-trial).

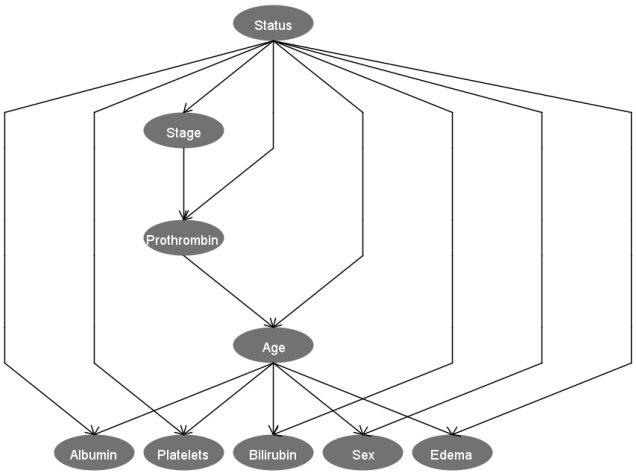


Figure 76. Bayesian network trained based on accuracy (non-trial).

Neural Networks

When training the neural network for both the trial and non-trial groups, two distinct configurations were used for each group: one with a single hidden layer and the other with two hidden layers.

In the non-trial group, the neural network with only one hidden layer reveals the key features influencing the classification of cirrhosis status. Age stands out as a highly influential feature, with significant positive weights in node 1 (0.807) and node 4 (1.452). This highlights the importance of age in distinguishing between the different classes, particularly when differentiating between classes D and C. Albumin also plays a major role, particularly in node 4, where its positive weight of 0.548 suggests that it contributes significantly to determining the severity of liver disease.

Regarding the outputs for each class, class D is predicted based on the significant positive contribution of node 4 (1.442), with smaller negative contributions from node 2 (-1.276) and node 3 (-1.488). Class C is predicted when the weighted sum surpasses a threshold of -2.281, with particularly strong positive influences from nodes 2 (1.134) and 6 (1.908), indicating that these nodes are critical in predicting class C. For class CL, the model faces difficulty in making predictions due to the negative weights from nodes 1 (-1.145) and 2 (-0.196). The threshold for class CL is set at -0.600, reflecting the challenges in classifying this class due to its feature overlap with the other classes and lower discriminatory power of the input features.

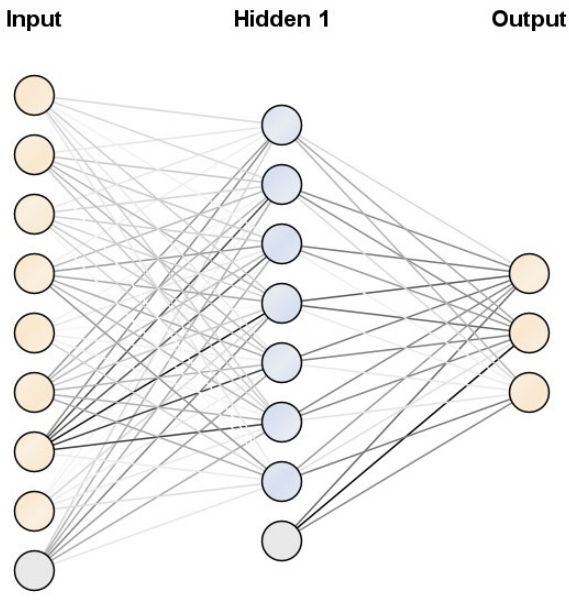


Figure 77. Neural network trained based on accuracy (non-trial).

The use of two hidden layers in the non-trial group (Figure 78) allows for capturing more complex relationships between the input features and the target classes. The outputs show a clear division of influence between the nodes, particularly with the significant contributions from

certain features like Age, Albumin, and Bilirubin. The thresholds for each class suggest that the model can differentiate between disease severity and status but may struggle with underrepresented classes like class CL, which has less clear feature separation.

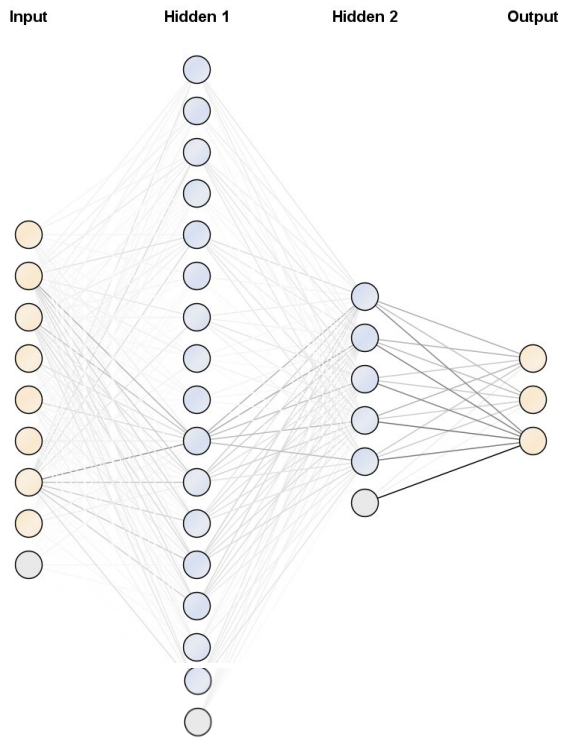


Figure 78. 2-Layer neural network trained based on accuracy (non-trial).

In the neural network designed for the trial group with one hidden layer, Age, Bilirubin, and Prothrombin emerge as prominent predictors effectively distinguishing class D and class C. Their significant weights suggest a robust model capability to differentiate cirrhosis severity. However, the model struggles to classify the Transplant (class CL) due to overlapping features and fewer data instances, reflecting challenges in handling rarer conditions within the dataset.

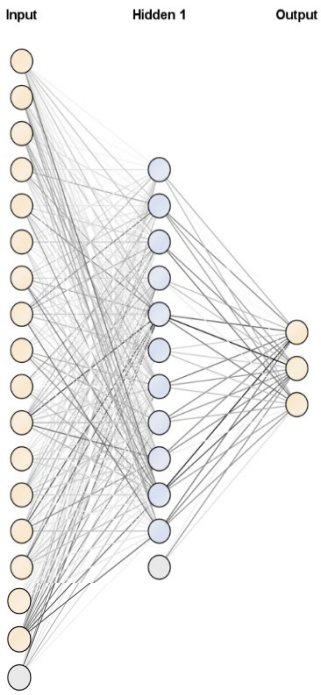


Figure 79. Neural network trained based on accuracy (trial).

In the neural network with two hidden layers for the trial group (Figure 80), the model's complexity allows for improved prediction capabilities compared to a single hidden layer. Key features such as Age, Bilirubin, and Prothrombin remain influential in the classification process, with Age having a particularly high positive weight (2.069) in hidden 1, indicating its significance in distinguishing between the classes, particularly between class D and class C. Bilirubin also plays a crucial role with a high positive weight (3.257) in node 1, suggesting its strong association with class D, especially for more severe liver dysfunction. Hidden layer 2 introduces additional complexity, with features like Prothrombin and SGOT playing significant roles across multiple nodes. The thresholds for classification suggest that while the model is effective at predicting classes D and C, the class CL prediction remains limited.

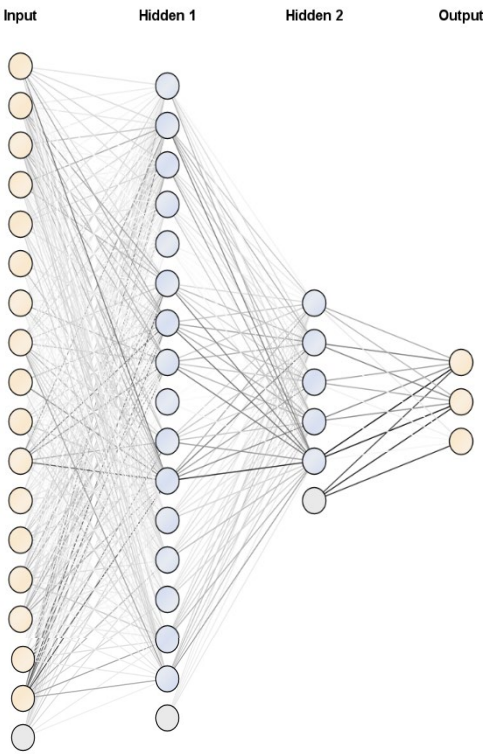


Figure 80. 2-layer neural network trained based on accuracy (trial).

Deep Learning

The deep learning model for the non-trial group provides a good balance of fit, with moderate RMSE, R-squared, and logloss. The model uses a deep learning architecture with an input layer, two hidden layers and a SoftMax output layer. The first hidden layer uses rectifier activation and the second uses the SoftMax function for classification.

Deep Learning Model

```
Model Metrics Type: Multinomial
 Description: Metrics reported on full training frame
 model id: rm-h2o-model-deep_learning_(2)-1136
 frame id: rm-h2o-frame-deep_learning_(2)-1136
 MSE: 0.16618127
 RMSE: 0.40765336
 R^2: 0.4680497
 logloss: 0.5155539
 mean_per_class_error: 0.43055555
 hit ratios: [0.76, 0.99, 1.0]
 AUC: NaN
 pr_auc: NaN
 AUC table: is not computed because it is disabled (model parameter 'auc_type' is set to AUTO or NONE) or due to domain size (maximum is 50 don
 pr_auc table: is not computed because it is disabled (model parameter 'auc_type' is set to AUTO or NONE) or due to domain size (maximum is 50
 CM: Confusion Matrix (Row labels: Actual class; Column labels: Predicted class):
         D   C  CL   Error      Rate
     D  12  20   0  0.6250   20 / 32
     C   0  62   0  0.0000    0 / 62
    CL   0   4   2  0.6667    4 / 6
Totals  12  86   2  0.2400  24 / 100
Status of Neuron Layers (predicting Status, 3-class classification, multinomial distribution, CrossEntropy loss, 323 weights/biases, 8.0 KB, 15
 Layer Units      Type Dropout       L1       L2 Mean Rate Rate RMS Momentum Mean Weight Weight RMS Mean Bias Bias RMS
     1    17     Input  0.00 %
     2    10 Rectifier       0 0.000010 0.000000  0.238135 0.423339 0.000000    0.058125   0.323280  0.602308 0.138206
     3    10 Rectifier       0 0.000010 0.000000  0.001169 0.000886 0.000000   -0.027007   0.333462  0.978740 0.083515
     4     3   Softmax         0.000010 0.000000  0.002850 0.004097 0.000000   -0.298029   0.932457 -0.031598 0.095256
Scoring History:
            Timestamp   Duration Training Speed    Epochs Iterations      Samples Training RMSE Training LogLoss Training r2 Training Classification Error Training AUC Training pr_auc
 2024-12-27 23:36:03  0.000 sec                  0.00000          0     0.000000           NaN              NaN         NaN                           NaN          NaN             NaN
 2024-12-27 23:36:03  0.026 sec 166666 obs/sec  10.00000          1  1000.000000       0.46218          0.70301     0.31623                       0.27000          NaN             NaN
 2024-12-27 23:36:03  0.140 sec 182926 obs/sec 150.00000         15 15000.000000       0.40765          0.51555     0.46805                       0.24000          NaN             NaN

H2O version: 3.42.0.1-rm10.3.1
```

Figure 81. Deep learning trained based on accuracy (non-trial).

The deep learning model for the trial group shows a strong performance, with good fit (R-squared), low RMSE, and improving performance over time, indicating effective learning. The model's use of two hidden layers and the SoftMax activation in the output layer are well-suited for the classification task.

Deep Learning Model

```
Model Metrics Type: Multinomial
 Description: Metrics reported on full training frame
 model id: rm-h2o-model-deep_learning-1261
 frame id: rm-h2o-frame-deep_learning-1261
 MSE: 0.12345278
 RMSE: 0.35135847
 R^2: 0.64331627
 logloss: 0.4133242
 mean_per_class_error: 0.24408521
 hit ratios: [0.84294873, 0.9647436, 1.0]
 AUC: NaN
 pr_auc: NaN
 AUC table: is not computed because it is disabled (model parameter 'auc_type' is set to AUTO or NONE) or due to domain size (maximum is 50 do
 pr_auc table: is not computed because it is disabled (model parameter 'auc_type' is set to AUTO or NONE) or due to domain size (maximum is 50
 CM: Confusion Matrix (Row labels: Actual class; Column labels: Predicted class):
         D    C  CL   Error      Rate
     D  115   10   0  0.0800  10 / 125
     C   26  138   4  0.1786  30 / 168
    CL    4    5  10  0.4737    9 / 19
Totals  145  153  14  0.1571  49 / 312
Status of Neuron Layers (predicting Status, 3-class classification, multinomial distribution, CrossEntropy loss, 533 weights/biases, 11.3 KB,
 Layer Units      Type Dropout       L1       L2 Mean Rate Rate RMS Momentum Mean Weight Weight RMS Mean Bias Bias RMS
     1    38     Input  0.00 %
     2    10 Rectifier       0 0.000010 0.000000  0.291618 0.449637 0.000000   -0.005516   0.238466  0.473549 0.107725
     3    10 Rectifier       0 0.000010 0.000000  0.001308 0.000789 0.000000   -0.007898   0.396627  1.035652 0.108544
     4     3   Softmax         0.000010 0.000000  0.002102 0.002025 0.000000   -0.233336   0.954864 -0.033687 0.099923
 Scoring History:
             Timestamp   Duration Training Speed    Epochs Iterations      Samples Training RMSE Training LogLoss Training r2 Training Classification Error Training AUC Training pr_auc
  2024-12-27 23:36:12  0.000 sec                 0.00000          0     0.000000            NaN              NaN         NaN                        NaN            NaN             NaN
  2024-12-27 23:36:12  0.056 sec 240000 obs/sec  10.00000          1  3120.000000        0.47834          0.71289     0.33892                    0.30128            NaN             NaN
  2024-12-27 23:36:12  0.698 sec  84629 obs/sec 150.00000         15 46800.000000        0.35136          0.41332     0.64332                    0.15705            NaN             NaN

H2O version: 3.42.0.1-rm10.3.1
```

Figure 82. Deep learning trained based on accuracy (trial).

B. Cost-trained models

In this section, the same models will be trained based on cost, and one of the resulting 10 models will be discussed.

Decision tree

The decision tree model based on cost for the non-trial group shows that Platelets are a key feature in the first split of the decision tree, with a threshold of 2.112. Instances with higher platelet levels are classified into the CL class, while the remaining instances proceed to further splits. This indicates that Platelets are crucial in differentiating between various stages, especially in identifying transplant candidates. Prothrombin is important in the second layer of splits, where higher values (above 1.644) are associated with more severe liver dysfunction and classify instances as class D. Lower values of Prothrombin help classify cases into class C or class D. Age and Bilirubin also play essential roles in classifying class D and class C. Age helps refine the classification with thresholds like 1.706 and 2.082, while Bilirubin further aids in distinguishing between these two classes.

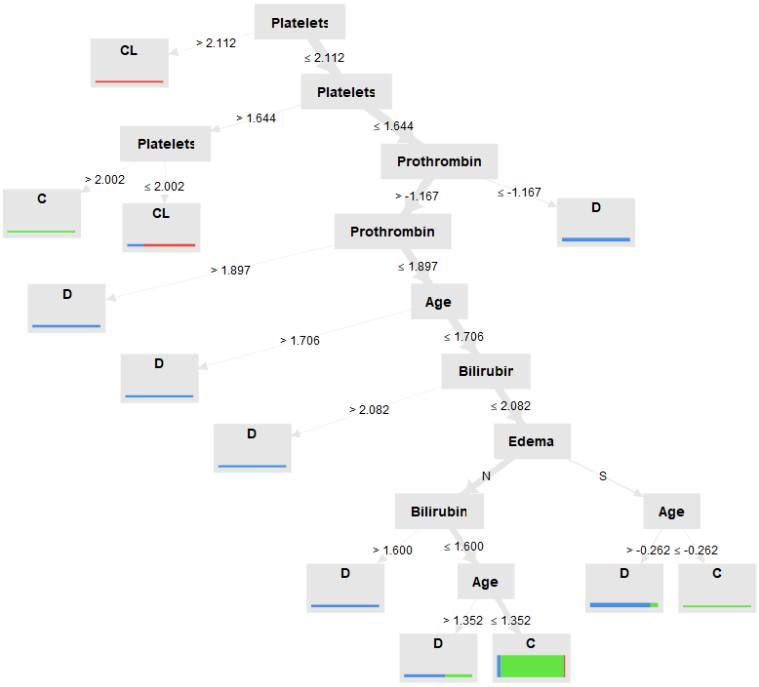


Figure 83. Decision tree trained based on cost (non-trial).

The decision tree for the trial group effectively uses a combination of features, including Triglycerides, Ascites, Prothrombin, Bilirubin, Age, SGOT, and Albumin, to accurately classify patients into their respective classes. In the cost-based decision tree, Triglycerides appear as a prominent feature for classifying class CL, with values greater than 2.557 separating this class from others. In contrast, the accuracy-based model doesn't place as much emphasis on Triglycerides, focusing more on Age, Bilirubin, and Prothrombin to distinguish between the different status.

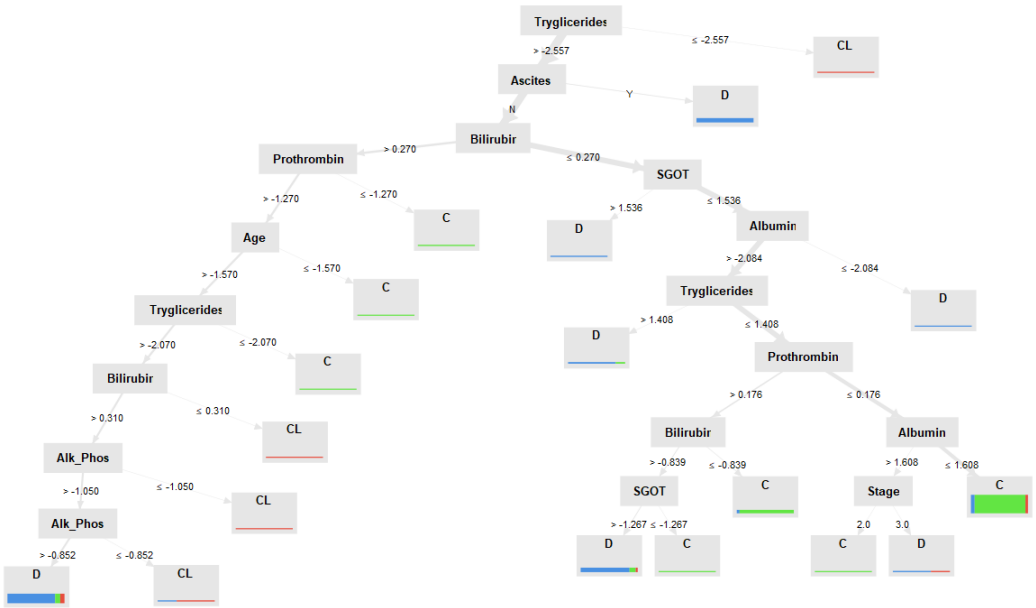


Figure 84. Decision tree trained based on cost (trial).

Rule-based model

The rule-based model for the non-trial group trained using the cost-based approach emphasizes several key features, including Bilirubin, Age, and Albumin, to differentiate between the classes. When compared to the accuracy-based model for the non-trial group, the cost-based decision tree offers more nuanced predictions, especially with respect to class D and class C. In the accuracy-based model, the primary focus is on the correct classification of C and D, with Platelets and Bilirubin playing significant roles in distinguishing these classes. However, the cost-based model adjusts for potential misclassifications, emphasizing features like Albumin and Prothrombin, which further refine the predictions, especially for the D class.

RuleModel

```
if Bilirubin ≤ -0.055 and Age ≤ 0.545 then C  (2 / 37 / 2)
if Bilirubin > 2.041 then D  (6 / 0 / 0)
if Age > 0.394 and Platelets > -0.560 and Bilirubin > -0.597 then C  (0 / 6 / 0)
if Stage = 4.0 and Albumin > -0.758 then D  (12 / 1 / 0)
else C  (9 / 16 / 4)

correct: 77 out of 95 training examples.
```

Figure 85. Rule-based model trained based on cost (non-trial).

The results of this rule-based for trial group are in line with the performance of the other models that also emphasize Bilirubin and Platelets as key predictors. However, compared to the accuracy-trained rule-based model, this one has a slightly higher level of precision in classification. This could be due to the rule-based model's simpler decision-making process, which makes it particularly effective in understanding clear feature thresholds.

RuleModel

```
if Bilirubin ≤ -0.423 and Alk_Phos ≤ 0.027 then C  (4 / 76 / 2)
if Prothrombin > 0.280 and Cholesterol > 0.260 then D  (25 / 0 / 0)
if Copper ≤ 0.168 and Hepatomegaly = N and Albumin ≤ 1.000 then C  (3 / 37 / 0)
if Prothrombin > 0.982 then D  (28 / 1 / 0)
if Bilirubin ≤ 0.613 and Albumin ≤ -0.333 and Copper ≤ 0.573 then C  (1 / 18 / 0)
if Tryglicerides > 0.656 and Cholesterol > 0.058 then D  (23 / 1 / 1)
if Albumin > 0.060 and Platelets ≤ -0.062 and Age ≤ 0.861 then C  (0 / 17 / 1)
if Prothrombin > -0.692 and Alk_Phos > 1.354 then D  (11 / 0 / 0)
if Tryglicerides ≤ -0.908 then C  (0 / 6 / 0)
if Platelets ≤ -0.263 then D  (16 / 4 / 2)
if Age ≤ -0.023 then CL  (1 / 5 / 18)
if Bilirubin ≤ 1.313 then C  (0 / 7 / 0)
else D  (3 / 0 / 0)

correct: 285 out of 311 training examples.
```

Figure 86. Rule-based model trained based on cost (trial).

Naive Bayes

In the Naive Bayes model trained based on cost for the non-trial group, Age shows a younger distribution for class CL, which aligns with the selection criteria for transplant suitability, whereas Age alone does not sufficiently differentiate D from C. Albumin is more effective at predicting disease severity, with lower levels in class D indicating more severe liver dysfunction. Platelets

are also an important feature, particularly for class CL, as higher platelet counts are preferable for transplant eligibility. Bilirubin and Prothrombin distributions clearly separate classes D and C, with higher Bilirubin levels indicating more severe liver disease in class D and normal Prothrombin levels in class C.

SimpleDistribution

```
Distribution model for label attribute Status


Class D (0.310)
8 distributions

Class C (0.630)
8 distributions

Class CL (0.060)
8 distributions
```

Figure 87. Naive Bayes trained based on cost (non-trial).

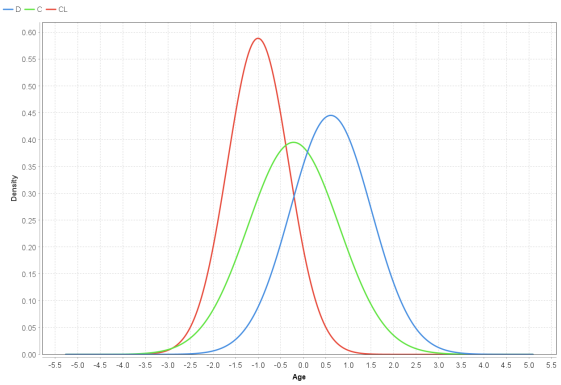


Figure 88. Naive Bayes for Age trained based on cost (non-trial).

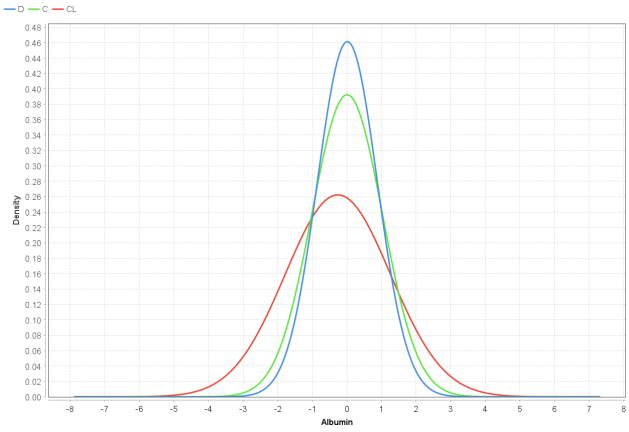


Figure 89. Naive Bayes for Albumin trained based on cost (non-trial).

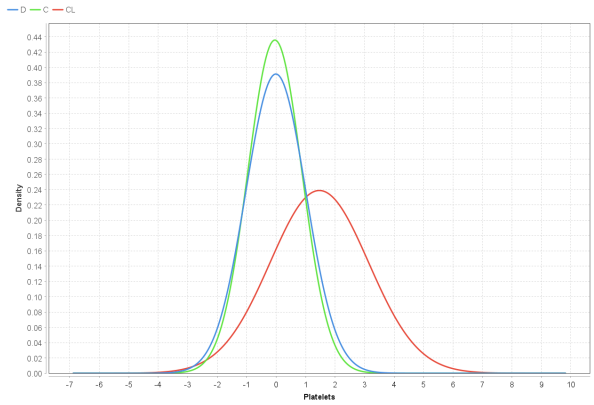


Figure 90. Naive Bayes for Platelets trained based on cost (non-trial).

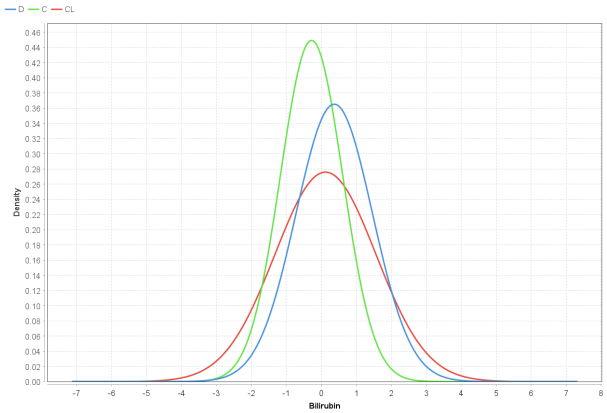


Figure 91. Naive Bayes for Bilirubin trained based on cost (non-trial).

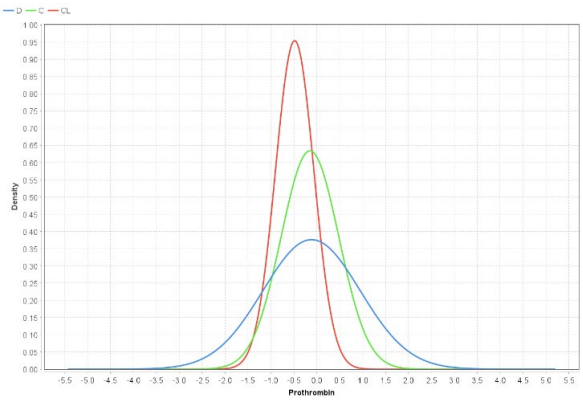


Figure 92. Naive Bayes for Prothrombin trained based on cost (non-trial).

In the Naive Bayes cost-trained model for the trial group, key features such as Age, Albumin, Platelets, and Bilirubin play pivotal roles in distinguishing between the classes. Age shows a significant distinction between class D and class C with class D exhibiting higher peaks in younger patients highlighting age as an important factor in classifying disease severity. Albumin exhibits clear separation between the classes with lower albumin levels associated with class D indicating more severe liver disease compared to class C. Platelets provides useful separation particularly between class D and class CL with higher platelet counts in class CL aligning with the medical standards for transplant eligibility. Bilirubin shows a pronounced contrast with higher

bilirubin levels in class D underscoring its role as a crucial marker for liver dysfunction and more severe disease states.

SimpleDistribution

```
Distribution model for label attribute Status


Class D (0.375)
17 distributions

Class C (0.548)
17 distributions

Class CL (0.077)
17 distributions
```

Figure 93. Naive Bayes trained based on cost (trial).

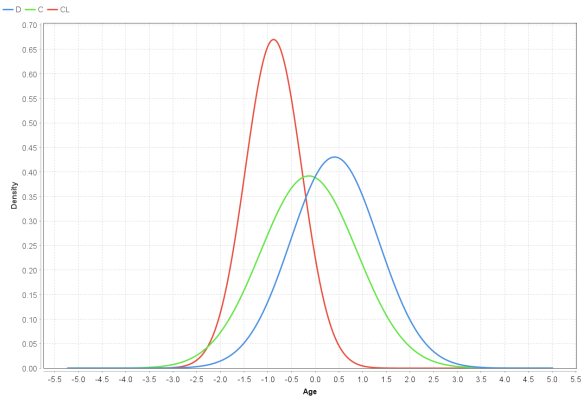


Figure 94. Naive Bayes for Age trained based on cost (trial).

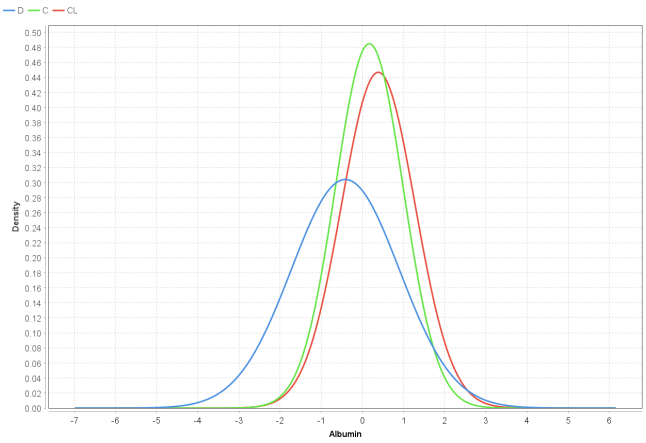


Figure 95. Naive Bayes for Albumin trained based on cost (trial).

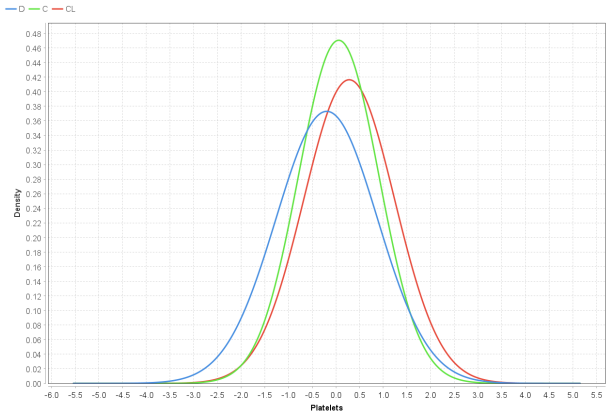


Figure 96. Naive Bayes for Platelets trained based on cost (trial).

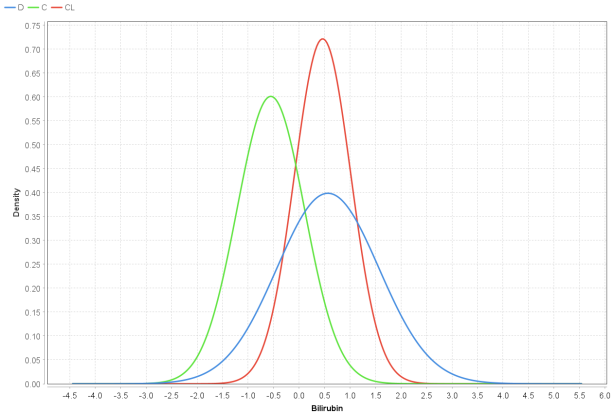


Figure 97. Naive Bayes for Bilirubin trained based on cost (trial).

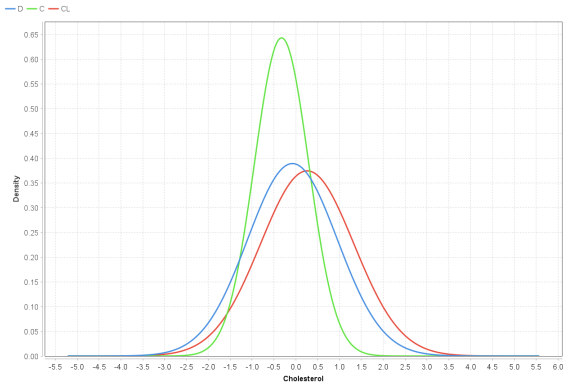


Figure 98. Naive Bayes for Cholesterol trained based on cost (trial).

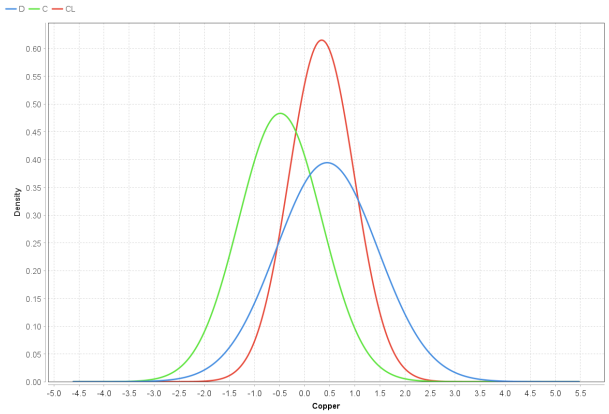


Figure 99. Naive Bayes for Copper trained based on cost (trial).

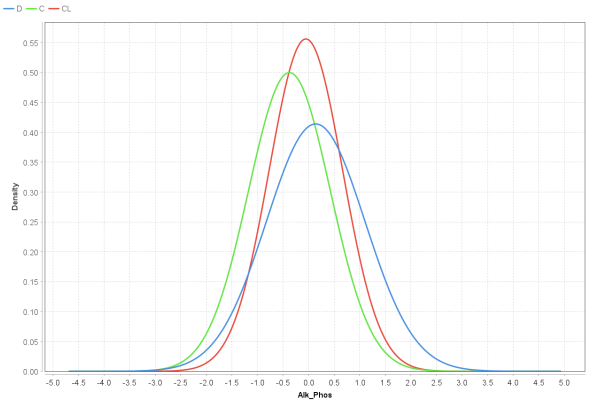


Figure 100. Naive Bayes for Alk_Phos trained based on cost (trial).

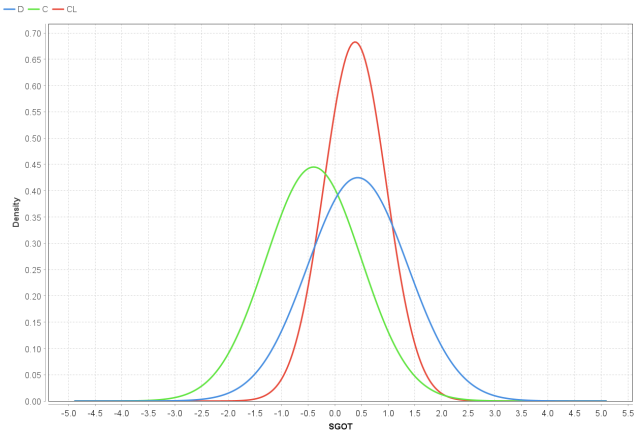


Figure 101. Naive Bayes for SGOT trained based on cost (trial).

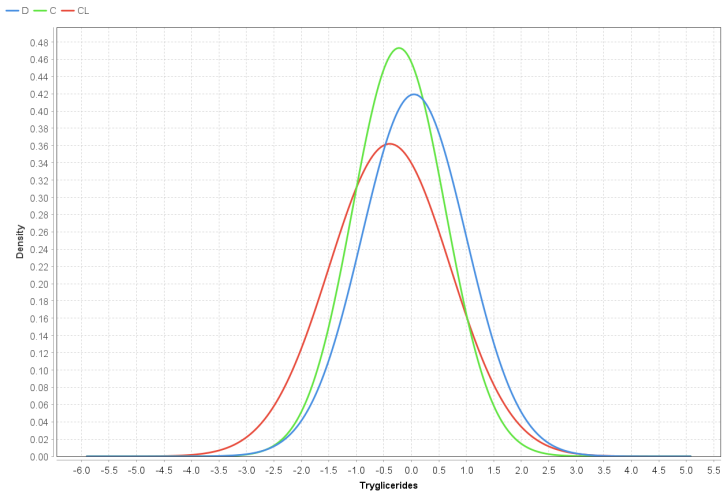


Figure 102. Naive Bayes for Triglycerides trained based on cost (trial).

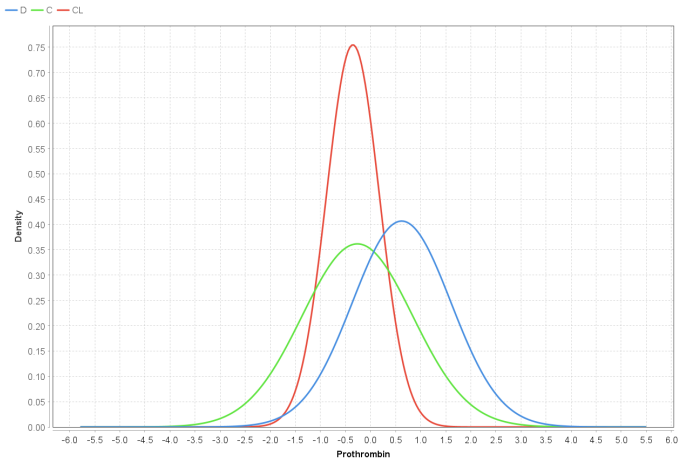


Figure 103. Naive Bayes for Prothrombin trained based on cost (trial).

Bayesian Network

The Bayesian network model for the non-trial group shows a clear structure where Age plays a crucial role in the network, influencing both Albumin and Prothrombin levels. It also plays an essential role in distinguishing between classes, especially when combined with other features like Bilirubin and Platelets. In addition, Bilirubin and Prothrombin also significantly impact class predictions, with Bilirubin being particularly influential for detecting class D.

W-BayesNet

```
Bayes Network Classifier
Using ADTree
#attributes=9 #classindex=8
Network structure (nodes followed by parents)
Age(2): Status Albumin
Albumin(1): Status Prothrombin
Platelets(1): Status Age
Bilirubin(1): Status Age
Prothrombin(2): Status Stage
Sex(2): Status Albumin
Edema(3): Status Stage
Stage(4): Status
Status(3):
LogScore Bayes: -365.8812531340827
LogScore BDeu: -504.0777216064423
LogScore MDL: -493.8242475079401
LogScore ENTROPY: -371.78723757925565
LogScore AIC: -424.78723757925565
```

Figure 104. Bayesian network trained based on cost (non-trial).

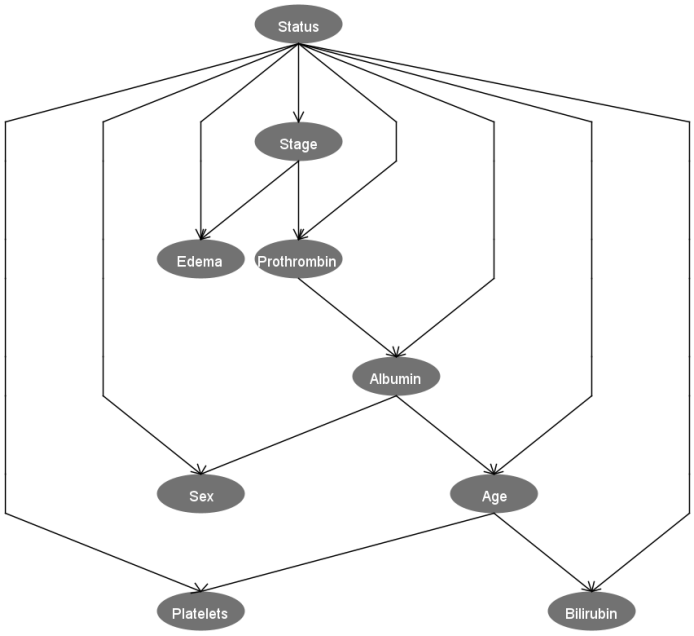


Figure 105. Bayesian network trained based on cost (non-trial).

In the trial group, the model structure is more complex, incorporating additional features like SGOT, Triglycerides, and Hepatomegaly. The feature dependencies highlight the importance of these variables in determining liver status. Prothrombin and Albumin, along with Bilirubin and Age, play central roles in classifying cirrhosis status, while the smaller sample size of the CL class still poses challenges for accurate classification in this category.

W-BayesNet

```
Bayes Network Classifier
Using ADTree
#attributes=18 #classindex=17
Network structure (nodes followed by parents)
Age(2): Status SGOT
Albumin(2): Status Hepatomegaly
Platelets(2): Status Albumin
Bilirubin(3): Status Sex
Cholesterol(1): Status Age
Copper(2): Status Bilirubin
Alk_Phos(2): Status Stage
SGOT(2): Status Bilirubin
Tryglicerides(1): Status Age
Prothrombin(2): Status Stage
Drug(3): Status Age
Sex(2): Status Stage
Ascites(3): Status Edema
Hepatomegaly(3): Status Stage
Spiders(3): Status Sex
Edema(3): Status Albumin
Stage(4): Status
Status(3):
LogScore Bayes: -3043.558850587819
LogScore BDeu: -3502.5842247489654
LogScore MDL: -3493.643225796487
LogScore ENTROPY: -2988.2589452692528
LogScore AIC: -3164.2589452692528
```

Figure 106. Bayesian network trained based on cost (trial).

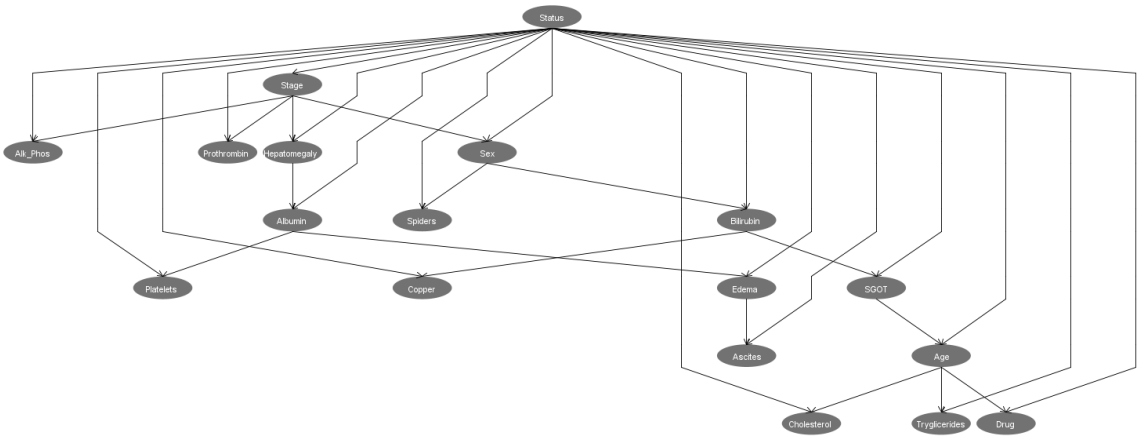


Figure 107. Bayesian network trained based on cost (trial).

Neural Networks

The neural network model for the non-trial group with one hidden layer demonstrates significant feature contributions. Age plays a notable role, particularly in nodes 1, 2, and 4, where it has large negative weights, suggesting that younger patients are more likely to be classified as class D. Albumin and Platelets show moderate influence in the classification, particularly in nodes 1, 2, 5, and 6, where lower levels of these features are correlated with class D, signifying more severe liver dysfunction. Bilirubin is highly influential in differentiating between class D and other classes. In nodes 1, 2, and 4, Bilirubin has strong negative weights, indicating that higher Bilirubin levels are associated with class D. Prothrombin contributes significantly to the network, especially in nodes 4, 5, and 6, where higher levels correlate with better liver function and are

associated with class C. The output layer shows clear thresholds for each class, with class D requiring the sum of weighted values to exceed 3.197, class C needing a value above -3.542, and class CL being more challenging to classify with a threshold of -1.321. Overall, the model's performance is heavily influenced by Bilirubin and Age, with these features providing strong separation between the D and C classes. However, the challenge of accurately predicting class CL persists.

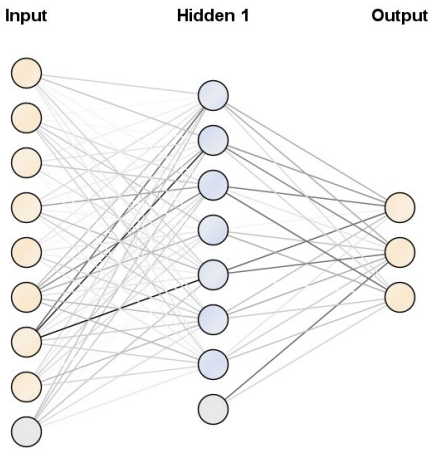


Figure 108. Neural network trained based on cost (non-trial).

The neural network model for the non-trial group with two hidden layers demonstrates the significant role of features like Age, Albumin, Platelets, Bilirubin, and Prothrombin in classifying the statuses. Age is important for distinguishing between D and C, with younger patients more likely to be classified as D. Albumin and Platelets are also influential, with lower levels associated with more severe liver dysfunction in the D class. Bilirubin plays a key role in identifying severe liver damage, strongly correlating with D. Prothrombin helps differentiate between C and D, with higher levels linked to better liver function in the C class. The model's thresholds for classifying the three statuses show how these features contribute to the final classification. However, the challenge remains in accurately predicting class CL.

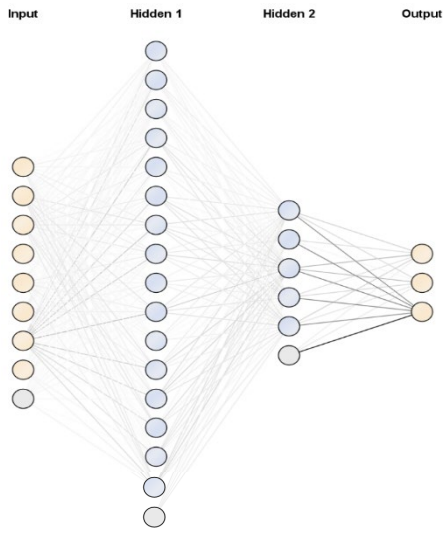


Figure 109. 2-layer neural network trained based on cost (non-trial).

In the neural network model for the trial group with one hidden layer Age is an important feature in classifying between class D and class C with higher negative weights associated with younger patients more likely to fall under the D category. Albumin and Platelets moderate the model's output especially in nodes 2, 4, and 5 where lower levels of these features correlate with class D signalling more severe liver dysfunction. Thresholds for classifying the outputs show that class D requires a weighted sum to exceed 0.469, class C has a threshold of -2.925, and class CL requires a threshold of 0.925 clearly indicating the model's sensitivity in differentiating between the D and CL groups with some overlap. However, classification of class CL (Transplant) remains challenging as evidenced by the higher threshold and overlap in feature distributions.

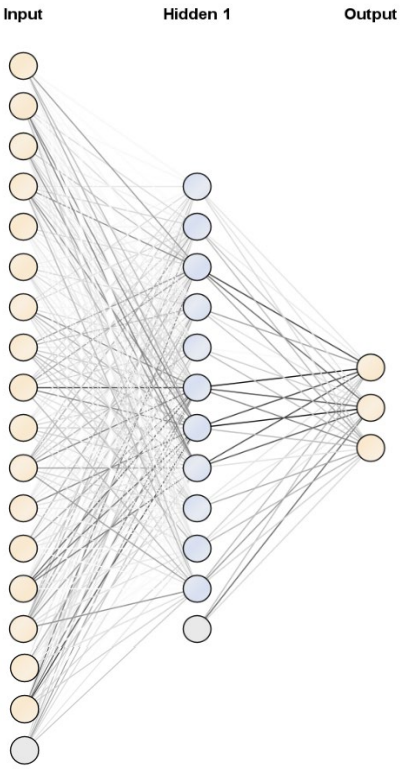


Figure 110. Neural network trained based on cost (trial).

The neural network model for the trial group with two hidden layers captures deeper relationships between the input features such as Age, Albumin, and Bilirubin, and the output classes. In Hidden layer 1, the nodes show significant influences from features like Age, Albumin, Platelets, and SGOT with weights indicating that younger age and lower albumin levels are linked to class D, while higher Platelet levels are linked to class CL. Hidden layer 2 refines decision boundaries further, with specific features showing either positive or negative weights depending on the class, helping the model combine and refine earlier learned representations. Features like Hepatomegaly, Ascites, Albumin, Bilirubin, and SGOT have a significant influence, demonstrating their importance in identifying the class of each instance. Adding a second hidden layer improves the network's ability to classify instances more effectively.

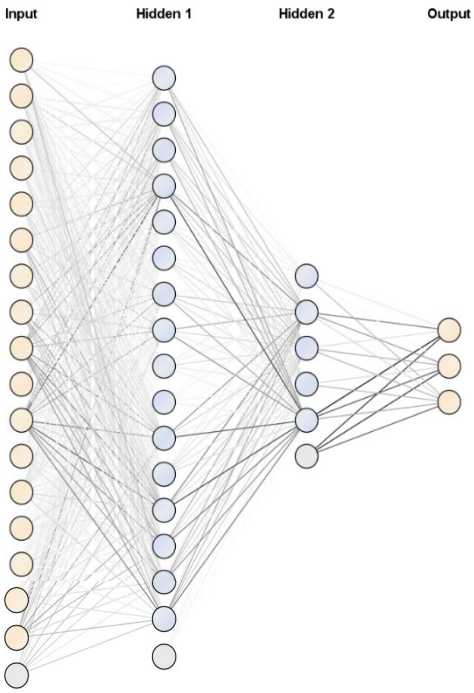


Figure 111. 2-Layer network trained based on cost (trial).

Deep Learning

The deep learning model trained on cost for the non-trial group shows a good predictive capability, as indicated by its R^2 value of 0.83, which suggests a strong correlation between the predicted and actual values. The low RMSE (Root Mean Squared Error) further supports this, as it indicates that the predictions are close to the actual values with minimal deviation. In addition, the model shows a gradual decrease in logloss, a key indicator of model calibration.

Deep Learning Model

```
Model Metrics Type: Multinomial
 Description: Metrics reported on full training frame
 model id: rm-h2o-model-deep_learning_(3)-1151
 frame id: rm-h2o-frame-deep_learning_(3)-1151
 MSE: 0.054172963
 RMSE: 0.23275086
 R^2: 0.82382774
 logloss: 0.17565343
 mean_per_class_error: 0.16487455
 hit ratios: [0.93, 1.0, 1.0]
 AUC: NaN
 pr_auc: NaN
 AUC table: is not computed because it is disabled (model parameter 'auc_type' is set to AUTO or NONE) or due to domain size (maxin
 pr_auc table: is not computed because it is disabled (model parameter 'auc_type' is set to AUTO or NONE) or due to domain size (ma
 CM: Confusion Matrix (Row labels: Actual class; Column labels: Predicted class):
         D   C  CL   Error     Rate
     D  26   5   0  0.1613   5 / 31
     C   0  63   0  0.0000   0 / 63
    CL   0   2   4  0.3333    2 / 6
Totals  26  70   4  0.0700  7 / 100
Status of Neuron Layers (predicting Status, 3-class classification, multinomial distribution, CrossEntropy loss, 323 weights/biases
 Layer Units      Type Dropout       L1       L2 Mean Rate Rate RMS Momentum Mean Weight Weight RMS Mean Bias Bias RMS
     1    17     Input  0.00 %
     2    10 Rectifier       0 0.000010 0.000000  0.238609 0.423126 0.000000     0.013188   0.360276  0.518061 0.156084
     3    10 Rectifier       0 0.000010 0.000000  0.002885 0.003587 0.000000     0.017959   0.400240  1.025230 0.103420
     4     3   Softmax         0.000010 0.000000  0.003504 0.004011 0.000000    -0.046652   0.980658 -0.049902 0.090821
Scoring History:
            Timestamp   Duration Training Speed    Epochs Iterations      Samples Training RMSE Training LogLoss Training r2 Training Classification Error Training AUC Training pr_auc
 2024-12-27 23:36:03  0.000 sec                 0.00000          0     0.000000           NaN              NaN         NaN                         NaN          NaN             NaN
 2024-12-27 23:36:03  0.030 sec 142857 obs/sec  10.00000          1  1000.000000       0.50797          0.84080     0.16088                     0.35000          NaN             NaN
 2024-12-27 23:36:03  0.188 sec  92592 obs/sec 150.00000         15 15000.000000       0.23275          0.17565     0.82383                     0.07000          NaN             NaN

H2O version: 3.42.0.1-rm10.3.1
```

Figure 112. Deep learning trained based on cost (non-trial).

The deep learning model trained on cost for the trial group demonstrates a high R^2 value of 0.83 that indicates that the model is highly effective in explaining the variability in the data. In essence, the model can capture 83% of the variation in disease status. This suggests that the model has learned important patterns within the dataset that help it make accurate predictions.

The low RMSE (Root Mean Squared Error) of 0.2383 further supports the model's accuracy. Additionally, the logloss value of 0.1918 is another critical performance indicator. This model's logloss value suggests that its predicted probabilities are well-calibrated, meaning that the model isn't just classifying the labels correctly but is also producing reliable probabilities that reflect the likelihood of each class. For example, the model can predict not just that a patient is likely to belong to class C but also provide a reliable probability of that prediction, which is essential in clinical decision-making scenarios.

Deep Learning Model

```
Model Metrics Type: Multinomial
 Description: Metrics reported on full training frame
 model id: rm-h2o-model-deep_learning_(4)-1513
 frame id: rm-h2o-frame-deep_learning_(4)-1513
 MSE: 0.056755867
 RMSE: 0.23823489
 R^2: 0.8339092
 logloss: 0.19179587
 mean_per_class_error: 0.07936508
 hit ratios: [0.92948717, 1.0, 1.0]
 AUC: NaN
 pr_auc: NaN
 AUC table: is not computed because it is disabled (model parameter 'auc_type' is set to AUTO or NONE) or due to domain size (maxin
 pr_auc table: is not computed because it is disabled (model parameter 'auc_type' is set to AUTO or NONE) or due to domain size (ma
 CM: Confusion Matrix (Row labels: Actual class; Column labels: Predicted class):
          D    C  CL   Error      Rate
     D  122    4   0  0.0317   4 / 126
     C   15  152   1  0.0952  16 / 168
    CL    1    1  16  0.1111    2 / 18
Totals  138  157  17  0.0705  22 / 312
Status of Neuron Layers (predicting Status, 3-class classification, multinomial distribution, CrossEntropy loss, 533 weights/biases
 Layer Units      Type Dropout       L1       L2 Mean Rate Rate RMS Momentum Mean Weight Weight RMS Mean Bias Bias RMS
     1    38     Input  0.00 %
     2    10 Rectifier       0 0.000010 0.000000  0.294246 0.448569 0.000000   -0.001038   0.295854  0.536469 0.117223
     3    10 Rectifier       0 0.000010 0.000000  0.001777 0.001932 0.000000    0.090472   0.484904  0.996006 0.098301
     4     3   Softmax         0.000010 0.000000  0.002772 0.003146 0.000000   -0.248872   0.944660 -0.027178 0.114021
Scoring History:
             Timestamp   Duration Training Speed    Epochs Iterations      Samples Training RMSE Training LogLoss Training r2 Training Classification Error Training AUC Training pr_auc
 2024-12-27 23:55:41  0.000 sec                  0.00000          0     0.000000           NaN              NaN         NaN                          NaN          NaN             NaN
 2024-12-27 23:55:41  0.349 sec  97500 obs/sec  10.00000          1  3120.000000       0.47889          0.72708     0.32888                      0.30128          NaN             NaN
 2024-12-27 23:55:42  0.711 sec 119693 obs/sec 150.00000         15 46800.000000       0.23823          0.19180     0.83391                      0.07051          NaN             NaN

H2O version: 3.42.0.1-rm10.3.1
```

Figure 113. Deep learning trained based on cost (trial).

3.1.2 MERGED GROUPS: COMBINING TRIAL AND NON-TRIAL DATA

A. Accuracy-based training

Decision Tree

In the decision tree for the merged group, we can observe a mixture of features from both the trial and non-trial data. Bilirubin and Albumin play crucial roles in the initial splits of the decision tree, showing their strong discriminative power in distinguishing between different classes. The model first uses these features to separate the classes, with the structure being similar to the trial group's accuracy-trained decision tree, where Bilirubin was already highlighted as influential.

However, unlike the separate models, this tree appears to incorporate other features (e.g., Platelets and Age), which were more relevant in the non-trial group. The merged model likely benefits from the combination of more varied data sources, which helps the model to generalize better, by using more features effectively across different class boundaries

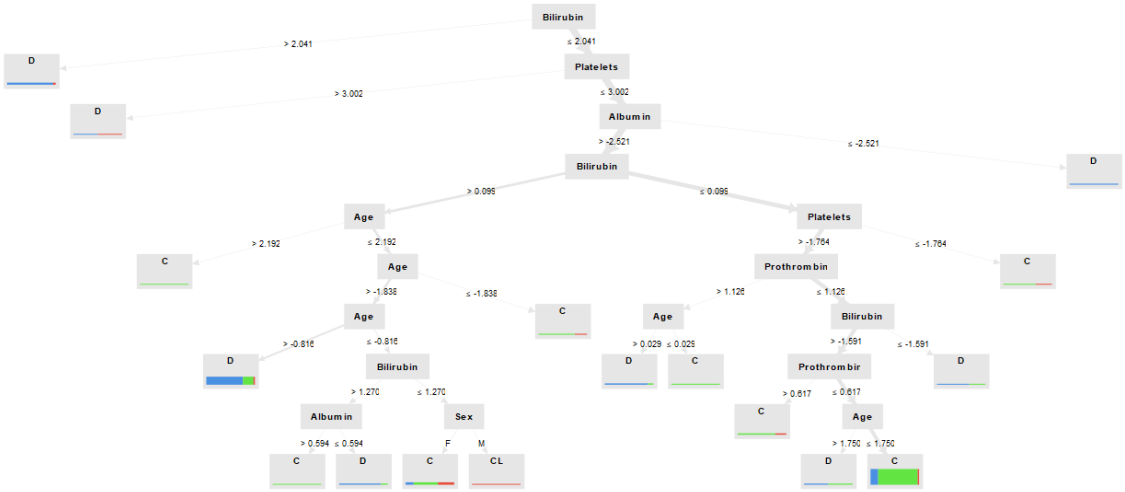


Figure 114. Decision tree trained based on accuracy (combined).

Rule-based Model

The rule-based model for the merged group leverages a diverse set of feature interactions, such as Bilirubin, Albumin, Platelets, and Prothrombin, making it more robust in predicting Censored, Transplant, and Death. By incorporating both clinical trial and hospital data, this model achieves better generalization. The model places significant emphasis on Bilirubin as a threshold feature for classifying cases. Its role is pivotal, as it frequently appears in the decision-making rules for differentiating between different classes.

RuleModel

```
if Bilirubin ≤ 0.099 and Prothrombin ≤ 0.262 and Albumin > -0.455 and Prothrombin ≤ -0.703 then C  (6 / 57 / 1)
if Bilirubin ≤ 0.099 and Bilirubin ≤ -0.554 and Age ≤ 0.025 then C  (3 / 55 / 3)
if Prothrombin > 0.211 and Bilirubin > 0.196 and Age > -0.552 then D  (57 / 6 / 0)
if Bilirubin ≤ 0.538 and Prothrombin ≤ 0.211 and Bilirubin ≤ -0.903 then C  (2 / 22 / 0)
if Bilirubin ≤ 0.554 and Prothrombin ≤ 0.211 and Albumin ≤ 0.589 and Albumin > -0.654 and Bilirubin ≤ 0.310 then C  (4 / 27 / 1)
if Age > -0.816 and Platelets ≤ 0.110 and Age > 1.192 then D  (15 / 1 / 0)
if Age > -0.816 and Bilirubin > 0.089 and Platelets ≤ -0.073 and Bilirubin ≤ 0.422 then D  (8 / 1 / 0)
if Albumin ≤ -0.512 and Prothrombin > -0.142 and Age > -0.162 then D  (10 / 0 / 0)
if Bilirubin ≤ 0.554 and Albumin ≤ 0.274 and Bilirubin > 0.089 then C  (0 / 9 / 0)
if Bilirubin > 0.380 and Age > -0.816 and Prothrombin > -0.249 and Age ≤ -0.560 then D  (8 / 0 / 0)
if Bilirubin ≤ 0.270 and Bilirubin ≤ -0.839 then C  (1 / 8 / 0)
if Bilirubin ≤ 0.270 and Albumin ≤ 1.500 and Albumin > 0.847 then C  (0 / 8 / 0)
if Albumin > -0.748 and Age > -1.269 and Age > 0.991 then D  (5 / 0 / 0)
if Bilirubin > 1.150 and Albumin > -0.853 and Albumin ≤ 0.096 then D  (8 / 1 / 0)
if Platelets ≤ -0.300 and Albumin ≤ -0.226 and Age ≤ 0.394 then C  (1 / 7 / 0)
if Platelets ≤ 0.122 and Platelets > -0.300 and Age > -0.280 then D  (6 / 0 / 0)
if Age ≤ -1.571 and Bilirubin > 0.607 then C  (0 / 6 / 0)
if Platelets ≤ -0.063 and Age ≤ -0.749 and Age > -1.098 then D  (6 / 0 / 0)
if Bilirubin ≤ 0.260 and Age > -0.327 and Platelets > -0.157 then C  (0 / 6 / 1)
if Platelets > 0.353 and Platelets ≤ 0.911 then CL  (2 / 1 / 9)
if Platelets > 0.280 and Platelets ≤ 1.972 and Age > -1.238 then D  (6 / 1 / 0)
if Platelets ≤ -0.831 and Prothrombin > -0.692 and Albumin ≤ -0.012 then C  (1 / 4 / 0)
if Bilirubin > 0.910 and Platelets ≤ 1.631 then C  (0 / 4 / 0)
if Bilirubin > 0.113 and Platelets > 0.195 then CL  (0 / 0 / 5)
if Albumin ≤ 0.095 and Platelets > -1.279 and Bilirubin > -0.336 then D  (5 / 1 / 0)
if Bilirubin > -0.158 and Bilirubin ≤ 0.113 then C  (0 / 3 / 0)
if Platelets ≤ 0.100 and Platelets > -1.394 then CL  (0 / 0 / 5)
if Age > -1.764 and Albumin > 0.393 then D  (3 / 0 / 0)
else C  (0 / 1 / 0)

correct: 373 out of 411 training examples.
```

Figure 115. Rule-based model trained based on accuracy (combined).

Naive Bayes

The Naive Bayes model for the merged group shows that certain features (like Bilirubin and Albumin) are strongly indicative of specific classes (class D and class C), while others (such as Platelets and Prothrombin) are less effective at clearly distinguishing between classes but still contribute to the overall prediction. Class CL faces challenges in prediction due to the overlap in feature distributions with the other two classes, as well as the relatively small number of instances in the training data for this class.

SimpleDistribution

```
Distribution model for label attribute Status


Class D (0.381)
8 distributions

Class C (0.558)
8 distributions

Class CL (0.061)
8 distributions
```

Figure 116. Naive Bayes trained based on accuracy (combined).

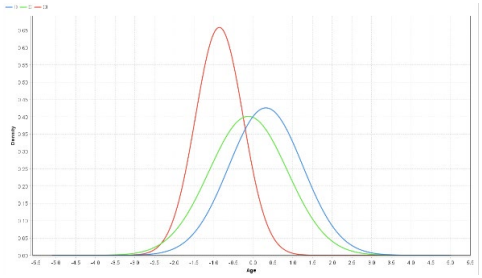


Figure 117. Naive Bayes for Age trained based on accuracy (combined).

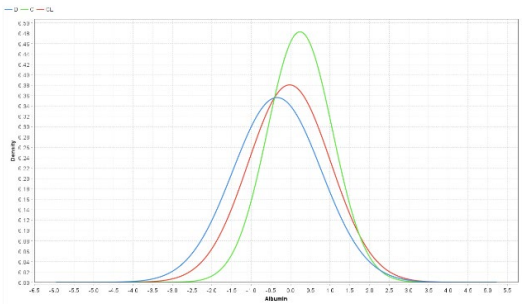


Figure 118. Naive Bayes for Albumin trained based on accuracy (combined).

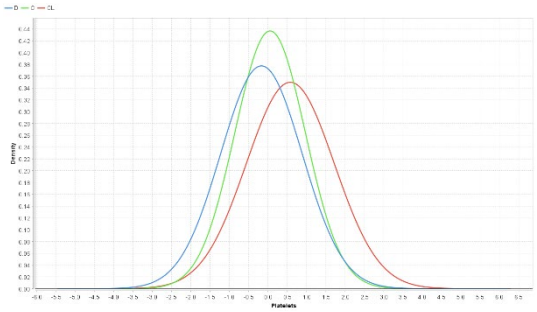


Figure 119. Naive Bayes for Platelets trained based on accuracy (combined).

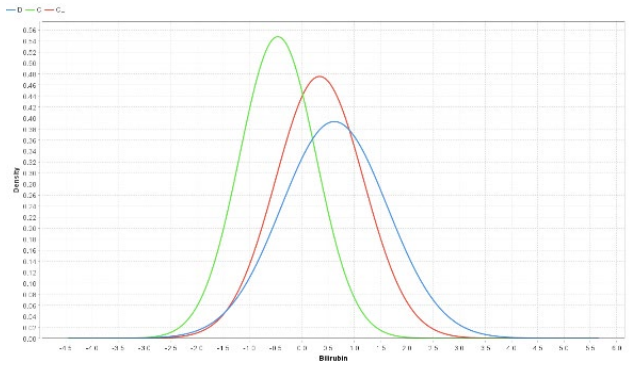


Figure 120. Naive Bayes for Bilirubin trained based on accuracy (combined).

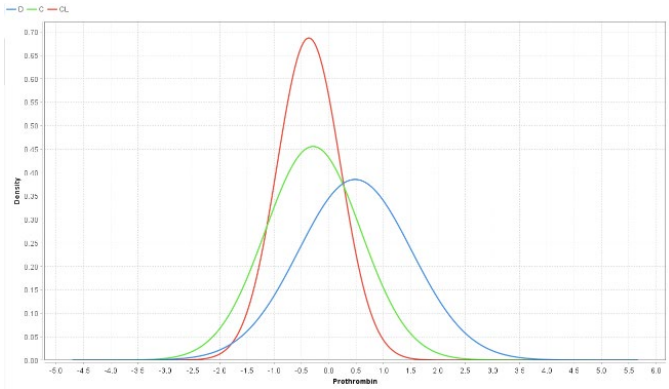


Figure 121. Naive Bayes for Prothrombin trained based on accuracy (combined).

Bayesian Network

Both models, for merged and separated groups, emphasize the importance of Prothrombin, Bilirubin, and Albumin. However, the merged model provides clearer pathways and relationships among these features, potentially offering a more nuanced understanding of how these factors interplay in the progression of cirrhosis. The merged group model's focus on the disease stage through Prothrombin and its impact on Edema and Albumin suggests a strong diagnostic focus on liver function and its complications.

W-BayesNet

```
Bayes Network Classifier
Using ADTree
#attributes=9 #classindex=8
Network structure (nodes followed by parents)
Age(2): Status Platelets
Albumin(2): Status Edema
Platelets(1): Status Albumin
Bilirubin(2): Status Platelets
Prothrombin(2): Status Stage
Sex(2): Status Age
Edema(3): Status Prothrombin
Stage(4): Status
Status(3):
LogScore Bayes: -2151.8061226587597
LogScore BDeu: -2277.1245971764893
LogScore MDL: -2273.6756782884754
LogScore ENTROPY: -2105.087024506689
LogScore AIC: -2161.087024506689
```

Figure 122. Bayesian network trained based on accuracy (combined).

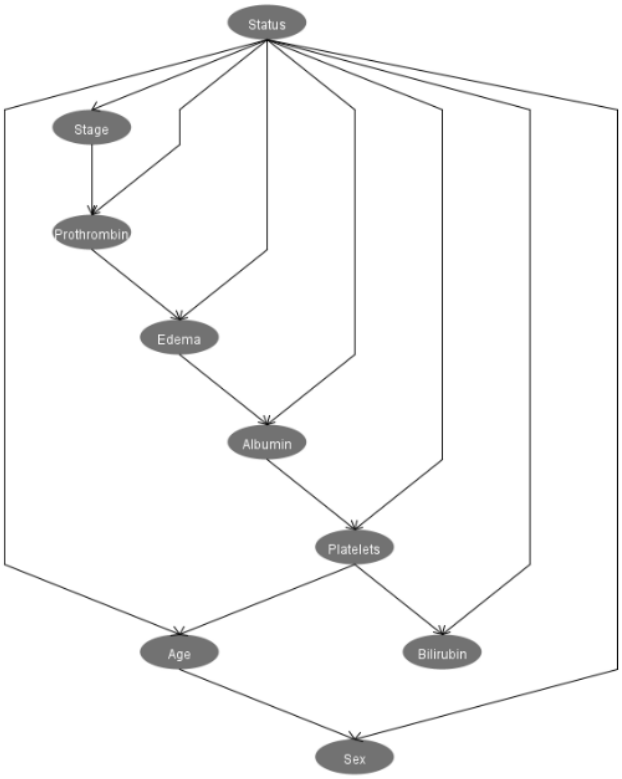


Figure 123. Bayesian network trained based on accuracy (combined).

Neural Networks

For the neural network models of the merged group, two configurations were used again: one with a single hidden layer and another with two hidden layers. These configurations were tested to assess the impact of model complexity on performance.

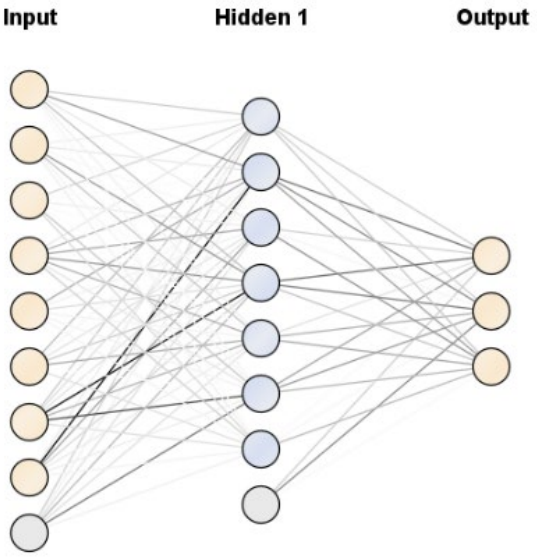


Figure 124. Neural network trained based on accuracy (combined).

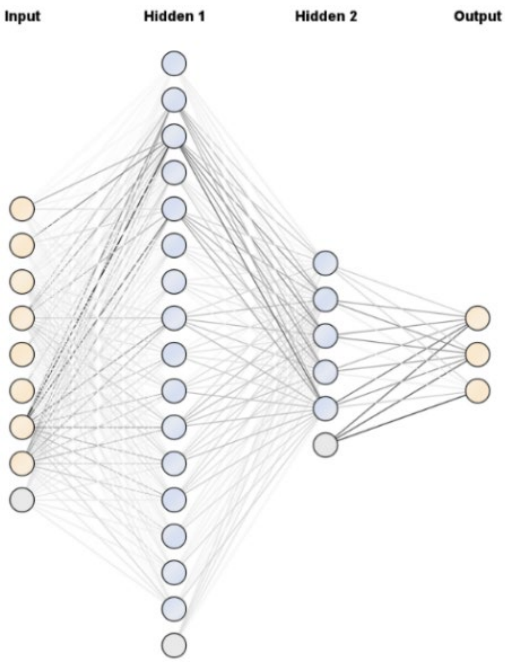


Figure 125. 2-layer neural network trained based on accuracy (combined).

Deep Learning

The scoring history reveals that the model shows consistent improvement over time, indicating that it is effectively learning and adapting to the training data. The model performs well in distinguishing between the Censored (C) and Death (D) classes, which have more defined and separable features. This is evident from the confusion matrix, where a larger number of instances are correctly classified as C and D. However, the model struggles with the Transplant (CL) class.

Deep Learning Model

```
Model Metrics Type: Multinomial
 Description: Metrics reported on full training frame
 model id: rm-h2o-model-deep_learning_(6)-6260
 frame id: rm-h2o-frame-deep_learning_(6)-6260
 MSE: 0.117890455
 RMSE: 0.34335178
 R^2: 0.65234196
 logloss: 0.37394997
 mean_per_class_error: 0.19881104
 hit ratios: [0.83980584, 0.9684466, 1.0]
 AUC: NaN
 pr_auc: NaN
 AUC table: is not computed because it is disabled (model parameter 'auc_type' is set to AUTO or NONE) or due to domain size (maximum is 50 domains).
 pr_auc table: is not computed because it is disabled (model parameter 'auc_type' is set to AUTO or NONE) or due to domain size (maximum is 50 domains).
 CM: Confusion Matrix (Row labels: Actual class; Column labels: Predicted class):
         D    C  CL   Error      Rate
     D  145   11   1  0.0764  12 / 157
     C   38  184   8  0.2000  46 / 230
    CL    7    1  17  0.3200    8 / 25
Totals  190  196  26  0.1602  66 / 412
Status of Neuron Layers (predicting Status, 3-class classification, multinomial distribution, CrossEntropy loss, 1,178 weights/biases, 18.4 KB, 123,600 training samples
 Layer Units      Type Dropout       L1       L2 Mean Rate Rate RMS Momentum Mean Weight Weight RMS Mean Bias Bias RMS
     1    17     Input  0.00 %
     2    25 Rectifier       0 0.000010 0.000000  0.183018 0.387723 0.000000    0.009365   0.352511  0.527482 0.125726
     3    25 Rectifier       0 0.000010 0.000000  0.002445 0.002453 0.000000    0.014856   0.383715  1.028636 0.151893
     4     3   Softmax         0.000010 0.000000  0.005204 0.015385 0.000000   -0.156207   0.606156 -0.122717 0.060850
Scoring History:
            Timestamp   Duration Training Speed    Epochs Iterations       Samples Training RMSE Training LogLoss Training r2 Training Classification Error Training AUC
 2025-01-01 20:55:51  0.000 sec                  0.00000          0      0.000000           NaN              NaN         NaN                           NaN          NaN
 2025-01-01 20:55:51  0.210 sec  65396 obs/sec  10.00000          1   4120.000000       0.43671          0.66568     0.43759                       0.24515          NaN
 2025-01-01 20:55:56  4.551 sec  28167 obs/sec 300.00000         30 123600.000000       0.34335          0.37395     0.65234                       0.16019          NaN
```

Figure 126. Deep learning trained based on accuracy (combined).

B. Cost-based training

For the cost-trained merged group models, there is no specific explanation required for each individual model, as the general trends observed in the accuracy-trained merged group models

largely apply here as well. The overarching conclusion is that combining trial and non-trial data leads to more effective and generalizable models.

Decision Tree

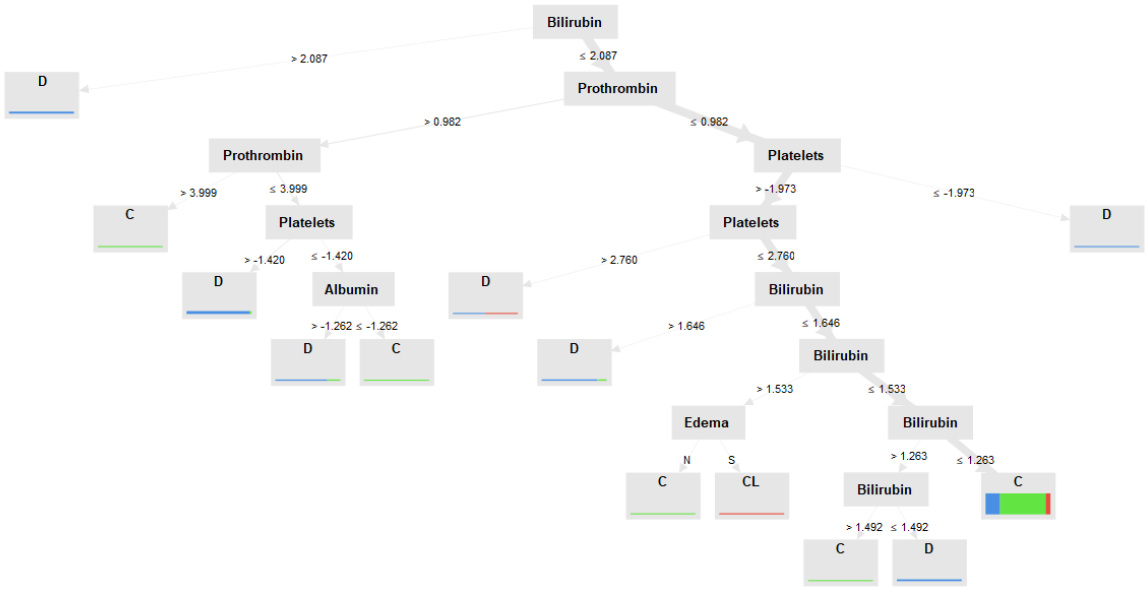


Figure 127. Decision tree trained based on cost (combined).

Rule-based Model

RuleModel

```
if Bilirubin ≤ -0.007 and Bilirubin ≤ -0.554 then C  (12 / 132 / 2)
if Prothrombin > 0.314 then D  (90 / 21 / 3)
if Bilirubin ≤ -0.034 and Albumin > -0.582 and Albumin ≤ 0.655 then C  (1 / 36 / 1)
if Age > -0.465 and Platelets ≤ -0.961 then D  (12 / 1 / 0)
if Bilirubin > 0.642 and Stage = 3.0 and Age > -0.837 then D  (12 / 0 / 0)
if Prothrombin ≤ -0.647 and Bilirubin ≤ 0.523 and Age ≤ -0.063 then C  (1 / 10 / 0)
if Age > 0.884 and Bilirubin > 0.048 then C  (1 / 11 / 0)
if Age > 0.030 and Prothrombin > -0.481 then D  (10 / 0 / 0)
if Platelets ≤ -0.185 then C  (5 / 14 / 3)
if Bilirubin > 0.270 and Albumin > 0.496 then CL  (0 / 0 / 10)
if Bilirubin > 1.408 and Bilirubin ≤ 2.180 then D  (5 / 0 / 0)
if Bilirubin ≤ -0.260 and Albumin ≤ 1.143 then D  (4 / 0 / 0)
else C  (2 / 5 / 5)

correct: 351 out of 409 training examples.
```

Figure 128. Rule-based model trained based on cost (combined).

Naive Bayes

SimpleDistribution

```
Distribution model for label attribute Status


Class D (0.342)
8 distributions

Class C (0.597)
8 distributions

Class CL (0.061)
8 distributions
```

Figure 129. Naive Bayes trained based on cost (combined).

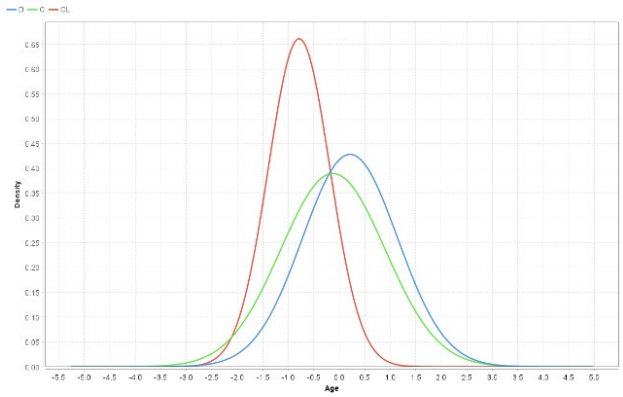


Figure 130. Naive Bayes for Age trained based on cost (combined).

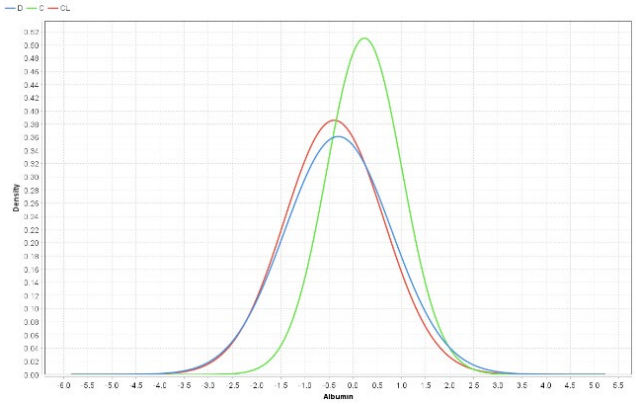


Figure 131. Naive Bayes for Albumin trained based on cost (combined).

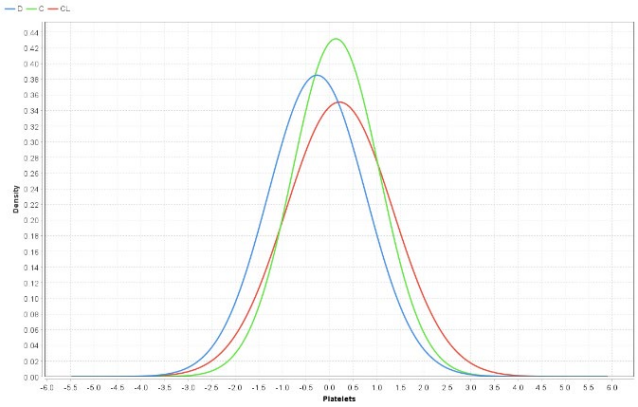


Figure 132. Naive Bayes for Platelets trained based on cost (combined).

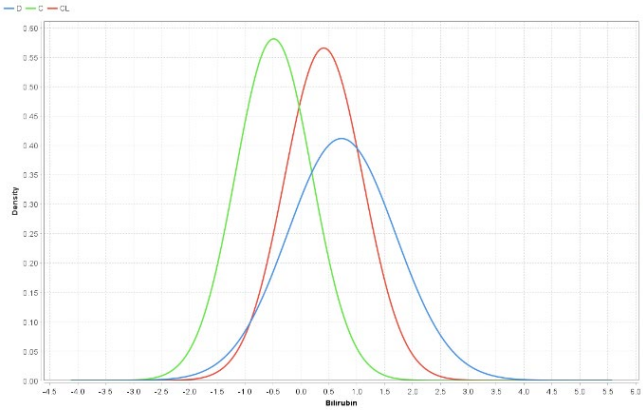


Figure 133. Naive Bayes for Bilirubin trained based on cost (combined).

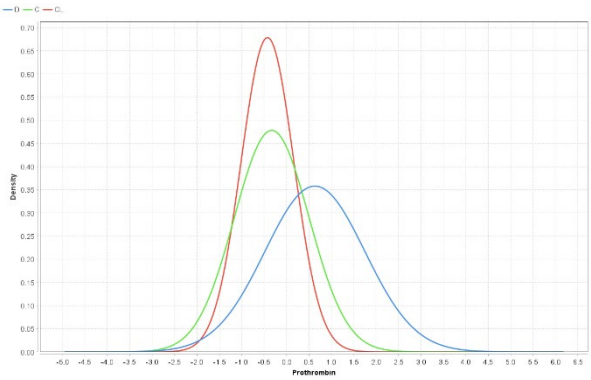


Figure 134. Naive Bayes for Prothrombin trained based on cost (combined).

Bayesian Network

W-BayesNet

```
Bayes Network Classifier
Using ADTree
#attributes=9 #classindex=8
Network structure (nodes followed by parents)
Age(4): Status Bilirubin
Albumin(2): Status Edema
Platelets(2): Status Stage
Bilirubin(3): Status Stage
Prothrombin(2): Status Stage
Sex(2): Status Bilirubin
Edema(3): Status Prothrombin
Stage(4): Status
Status(3):
LogScore Bayes: -2466.964106367172
LogScore BDeu: -2792.5682601685876
LogScore MDL: -2770.332193076955
LogScore ENTROPY: -2421.1128388146826
LogScore AIC: -2537.1128388146826
```

Figure 135. Bayesian network trained based on cost (combined).

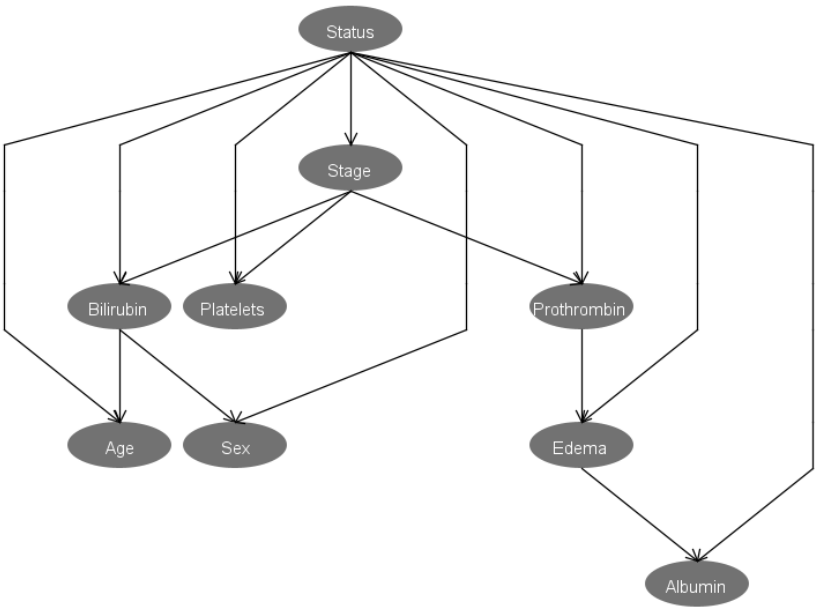


Figure 136. Bayesian network trained based on cost (combined).

Neural Networks

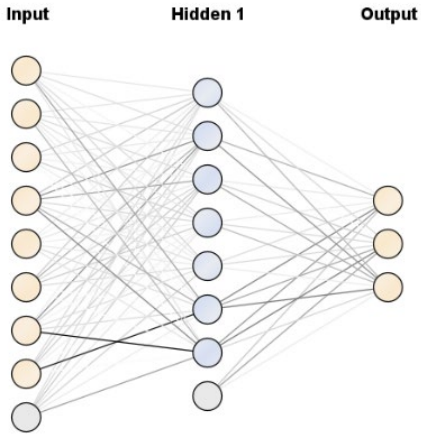


Figure 137. Neural network trained based on cost (combined).

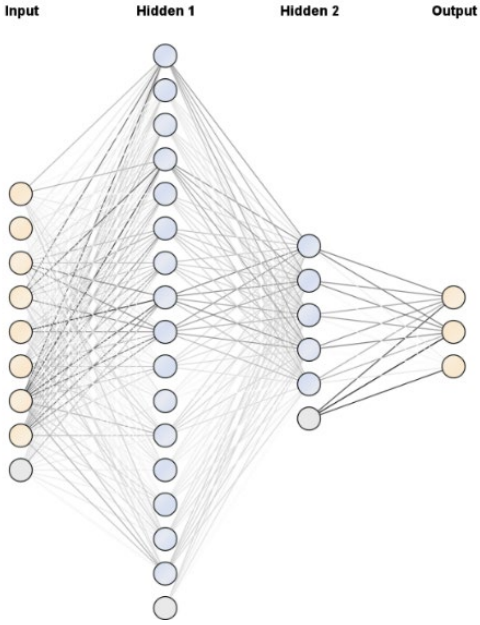


Figure 138. 2-layer neural network trained based on cost (combined).

Deep Learning

Deep Learning Model

```
Model Metrics Type: Multinomial
 Description: Metrics reported on full training frame
 model id: rm-h2o-model-deep_learning_(7)-6272
 frame id: rm-h2o-frame-deep_learning_(7)-6272
 MSE: 0.064928986
 RMSE: 0.25481167
 R^2: 0.80906564
 logloss: 0.20723054
 mean_per_class_error: 0.057068847
 hit ratios: [0.91019416, 1.0, 1.0]
 AUC: NaN
 pr_auc: NaN
 AUC table: is not computed because it is disabled (model parameter 'auc_type' is set to AUTO or NONE) or due to domain size (maximum is 50 domains).
 pr_auc table: is not computed because it is disabled (model parameter 'auc_type' is set to AUTO or NONE) or due to domain size (maximum is 50 domains).
 CM: Confusion Matrix (Row labels: Actual class; Column labels: Predicted class):
          D    C  CL   Error      Rate
     D  159    4   0  0.0245   4 / 163
     C   29  192   4  0.1467  33 / 225
    CL    0    0  24  0.0000    0 / 24
Totals  188  196  28  0.0898  37 / 412
Status of Neuron Layers (predicting Status, 3-class classification, multinomial distribution, CrossEntropy loss, 1,178 weights/biases, 18.4 KB, 123,600 tra
 Layer Units      Type Dropout       L1       L2 Mean Rate Rate RMS Momentum Mean Weight Weight RMS Mean Bias Bias RMS
     1    17     Input  0.00 %
     2    25 Rectifier       0 0.000010 0.000000  0.183023 0.389925 0.000000    0.013910   0.379727  0.473078 0.164185
     3    25 Rectifier       0 0.000010 0.000000  0.003638 0.005469 0.000000    0.031539   0.402783  1.059765 0.245352
     4     3   Softmax         0.000010 0.000000  0.002300 0.004361 0.000000   -0.177160   0.599697 -0.097308 0.077218
Scoring History:
            Timestamp   Duration Training Speed   Epochs Iterations       Samples Training RMSE Training LogLoss Training r2 Training Classification Error
 2025-01-01 20:55:56  0.000 sec                  0.00000          0      0.000000           NaN              NaN         NaN                           NaN
 2025-01-01 20:55:57  1.200 sec   8459 obs/sec  10.00000          1   4120.000000       0.50908          0.81060     0.23788                       0.35680
 2025-01-01 20:56:02  6.838 sec  23976 obs/sec 300.00000         30 123600.000000       0.25481          0.20723     0.80907                       0.08981
```

Figure 139. Deep learning trained based on accuracy (combined).

3.2 Evaluation of Models

The performance of the predictive models will be evaluated based on accuracy, which represents the proportion of correct predictions relative to the total number of predictions, and cost, calculated using the predefined cost matrix that reflects the clinical importance of different types of misclassifications. Additionally, the models will be assessed for interpretability, focusing on how easily the predictions and decision-making processes can be understood by clinicians and stakeholders, and interest and novelty, emphasizing their contribution to advancing the understanding of survival outcomes in cirrhosis while uncovering new insights or confirming existing knowledge. By combining these criteria, the study aims to achieve a balance between accuracy and cost-effectiveness while providing clinically valuable insights into predicting patient outcomes.

3.2.1 TRAINING SEPARATE GROUPS: TRIAL AND NON-TRIAL DATA

A. Accuracy-trained performance

Decision tree

accuracy: 60.00% +/- 10.54% (micro average: 60.00%)

	true D	true C	true CL	class precision
pred. D	8	10	1	42.11%
pred. C	23	52	5	65.00%
pred. CL	1	0	0	0.00%
class recall	25.00%	83.87%	0.00%	

Figure 140. Decision tree accuracy-trained confusion matrix (non-trial).

accuracy: 68.62% +/- 6.32% (micro average: 68.59%)

	true D	true C	true CL	class precision
pred. D	71	24	8	68.93%
pred. C	53	143	11	69.08%
pred. CL	1	1	0	0.00%
class recall	56.80%	85.12%	0.00%	

Figure 141. Decision tree accuracy-trained confusion matrix (trial).

Rule-based model

accuracy: 67.00% +/- 8.23% (micro average: 67.00%)

	true D	true C	true CL	class precision
pred. D	14	9	2	56.00%
pred. C	18	53	4	70.67%
pred. CL	0	0	0	0.00%
class recall	43.75%	85.48%	0.00%	

Figure 142. Rule-based model accuracy-trained confusion matrix (non-trial).

accuracy: 67.65% +/- 9.39% (micro average: 67.63%)

	true D	true C	true CL	class precision
pred. D	80	33	4	68.38%
pred. C	37	129	13	72.07%
pred. CL	8	6	2	12.50%
class recall	64.00%	76.79%	10.53%	

Figure 143. Rule-based model accuracy-trained confusion matrix (trial).

Naive Bayes

accuracy: 67.00% +/- 9.49% (micro average: 67.00%)

	true D	true C	true CL	class precision
pred. D	16	9	1	61.54%
pred. C	13	50	4	74.63%
pred. CL	3	3	1	14.29%
class recall	50.00%	80.65%	16.67%	

Figure 144. Naive Bayes accuracy-trained confusion matrix (non-trial).

accuracy: 73.38% +/- 8.80% (micro average: 73.40%)

	true D	true C	true CL	class precision
pred. D	87	25	6	73.73%
pred. C	29	138	9	78.41%
pred. CL	9	5	4	22.22%
class recall	69.60%	82.14%	21.05%	

Figure 145. Naive Bayes accuracy-trained confusion matrix (trial).

Bayesian Network

accuracy: 63.00% +/- 14.18% (micro average: 63.00%)

	true D	true C	true CL	class precision
pred. D	14	13	3	46.67%
pred. C	18	49	3	70.00%
pred. CL	0	0	0	0.00%
class recall	43.75%	79.03%	0.00%	

Figure 146. Bayesian network accuracy-trained confusion matrix (non-trial).

accuracy: 74.03% +/- 3.58% (micro average: 74.04%)

	true D	true C	true CL	class precision
pred. D	89	25	10	71.77%
pred. C	31	142	9	78.02%
pred. CL	5	1	0	0.00%
class recall	71.20%	84.52%	0.00%	

Figure 147. Bayesian network accuracy-trained confusion matrix (trial).

Neural Networks

accuracy: 66.00% +/- 11.74% (micro average: 66.00%)

	true D	true C	true CL	class precision
pred. D	17	13	2	53.12%
pred. C	15	49	4	72.06%
pred. CL	0	0	0	0.00%
class recall	53.12%	79.03%	0.00%	

Figure 148. Neural network accuracy-trained confusion matrix (non-trial).

accuracy: 64.00% +/- 5.16% (micro average: 64.00%)

	true D	true C	true CL	class precision
pred. D	3	1	0	75.00%
pred. C	29	61	6	63.54%
pred. CL	0	0	0	0.00%
class recall	9.38%	98.39%	0.00%	

Figure 149. 2-layer neural network accuracy-trained confusion matrix (non-trial).

accuracy: 72.40% +/- 7.28% (micro average: 72.44%)

	true D	true C	true CL	class precision
pred. D	88	30	7	70.40%
pred. C	35	137	11	74.86%
pred. CL	2	1	1	25.00%
class recall	70.40%	81.55%	5.26%	

Figure 150. Neural network accuracy-trained confusion matrix (trial).

accuracy: 74.71% +/- 8.53% (micro average: 74.68%)

	true D	true C	true CL	class precision
pred. D	95	30	7	71.97%
pred. C	30	138	12	76.67%
pred. CL	0	0	0	0.00%
class recall	76.00%	82.14%	0.00%	

Figure 151. 2-layer neural network accuracy-trained confusion matrix (trial).

Deep Learning

accuracy: 66.00% +/- 12.65% (micro average: 66.00%)

	true D	true C	true CL	class precision
pred. D	17	12	2	54.84%
pred. C	15	49	4	72.06%
pred. CL	0	1	0	0.00%
class recall	53.12%	79.03%	0.00%	

Figure 152. Deep learning accuracy-trained confusion matrix (non-trial).

accuracy: 69.86% +/- 11.06% (micro average: 69.87%)

	true D	true C	true CL	class precision
pred. D	92	41	8	65.25%
pred. C	29	124	9	76.54%
pred. CL	4	3	2	22.22%
class recall	73.60%	73.81%	10.53%	

Figure 153. Deep learning accuracy-trained confusion matrix (trial).

Conclusions

The following table includes the results from the trained models, displaying the accuracy, cost and confidence intervals for each model.

Model	Accuracy	Cost
Decision tree (non-trial)	60.00% ± 10.54%	3.450 ± 0.669
Decision tree (trial)	68.62% ± 6.32%	2.682 ± 0.704
Rule-based (non-trial)	67.00% ± 8.23%	2.77 ± 0.713
Rule-based (trial)	67.65% ± 9.39%	2.655 ± 0.766
Naive Bayes (non-trial)	67.00% ± 9.49%	2.770 ± 0.759
Naive Bayes (trial)	73.38% ± 8.80%	2.269 ± 0.807
Bayesian network (non-trial)	63.00% ± 14.18%	2.990 ± 1.129
Bayesian network (trial)	74.03% ± 3.58%	2.203 ± 0.461
Neural network (non-trial)	66.00% ± 11.74%	2.670 ± 1.012
Neural network (trial)	72.40% ± 7.28%	2.224 ± 0.630
2-layer neural network (non-trial)	64.00% ± 5.16%	3.430 ± 0.600
2-layer neural network (trial)	74.71% ± 8.53%	1.974 ± 0.654
Deep learning (non-trial)	66.00% ± 12.65%	2.670 ± 0.996
Deep learning (trial)	69.86% ± 11.06%	2.315 ± 0.999

Table 2. Accuracy and cost performance of the different models trained based on accuracy.

Based on the results from the accuracy-based training models, the trial group models show superior performance in terms of both accuracy and cost efficiency when compared to the non-trial group. Among these, the 2-layer neural network emerges as the most effective model, offering a solid balance between accuracy (74.71%) and cost efficiency (1.97). Although its confidence interval is wider, the model's performance is consistent across different folds, demonstrating robustness and practical applicability, particularly in scenarios where both accuracy and cost are important. The Bayesian network also performs well, excelling in accuracy

(74.03%) and stability with smaller confidence intervals. This makes it a reliable choice, especially for applications requiring predictability, although its complexity could limit its interpretability and use in simpler contexts. Naive Bayes is simpler but exhibits more variability in both accuracy and cost, making it less stable and potentially less practical in real-world applications where consistency is crucial.

For the non-trial group, the rule-based model achieves the highest accuracy at 67%, but its wider confidence interval indicates variability in its performance. This suggests a trade-off between performance and stability, which may require additional tuning to ensure more consistent results. Decision trees offer stability in terms of cost, but they show the lowest accuracy (60%) and widest confidence intervals, making them the least effective in this group. The 2-layer neural network (non-trial) performs better than decision tree and rule-based models in terms of accuracy and cost, striking a reasonable balance with stable performance despite the smaller dataset. This model offers a good trade-off between accuracy, cost, and consistency, making it a viable alternative.

B. Cost-trained performance

Decision tree

accuracy: 60.00% +/- 16.33% (micro average: 60.00%)

	true D	true C	true CL	class precision
pred. D	20	20	2	47.62%
pred. C	11	40	4	72.73%
pred. CL	1	2	0	0.00%
class recall	62.50%	64.52%	0.00%	

Figure 154. Decision tree cost-trained confusion matrix (non-trial).

accuracy: 71.45% +/- 6.06% (micro average: 71.47%)

	true D	true C	true CL	class precision
pred. D	102	47	10	64.15%
pred. C	23	121	9	79.08%
pred. CL	0	0	0	0.00%
class recall	81.60%	72.02%	0.00%	

Figure 155. Decision tree cost-trained confusion matrix (trial).

Rule-based model

accuracy: 67.00% +/- 14.94% (micro average: 67.00%)

	true D	true C	true CL	class precision
pred. D	21	16	3	52.50%
pred. C	11	46	3	76.67%
pred. CL	0	0	0	0.00%
class recall	65.62%	74.19%	0.00%	

Figure 156. Rule-based model cost-trained confusion matrix (non-trial).

accuracy: 72.72% +/- 6.70% (micro average: 72.76%)

	true D	true C	true CL	class precision
pred. D	102	43	12	64.97%
pred. C	22	124	6	81.58%
pred. CL	1	1	1	33.33%
class recall	81.60%	73.81%	5.26%	

Figure 157. Rule-based model cost-trained confusion matrix (trial).

Naive Bayes

accuracy: 67.00% +/- 15.67% (micro average: 67.00%)

	true D	true C	true CL	class precision
pred. D	21	16	3	52.50%
pred. C	11	46	3	76.67%
pred. CL	0	0	0	0.00%
class recall	65.62%	74.19%	0.00%	

Figure 158. Naive Bayes cost-trained confusion matrix (non-trial).

accuracy: 75.01% +/- 9.55% (micro average: 75.00%)

	true D	true C	true CL	class precision
pred. D	98	32	10	70.00%
pred. C	24	133	6	81.60%
pred. CL	3	3	3	33.33%
class recall	78.40%	79.17%	15.79%	

Figure 159. Naive Bayes cost-trained confusion matrix (trial).

Bayesian Network

accuracy: 70.00% +/- 14.14% (micro average: 70.00%)

	true D	true C	true CL	class precision
pred. D	23	15	3	56.10%
pred. C	9	47	3	79.66%
pred. CL	0	0	0	0.00%
class recall	71.88%	75.81%	0.00%	

Figure 160. Bayesian network cost-trained confusion matrix (non-trial).

accuracy: 73.34% +/- 8.19% (micro average: 73.40%)

	true D	true C	true CL	class precision
pred. D	102	41	12	65.81%
pred. C	22	127	7	81.41%
pred. CL	1	0	0	0.00%
class recall	81.60%	75.60%	0.00%	

Figure 161. Bayesian network cost-trained confusion matrix (trial).

Neural Networks

accuracy: 70.00% +/- 13.33% (micro average: 70.00%)

	true D	true C	true CL	class precision
pred. D	22	14	2	57.89%
pred. C	10	48	4	77.42%
pred. CL	0	0	0	0.00%
class recall	68.75%	77.42%	0.00%	

Figure 162. Neural network cost-trained confusion matrix (non-trial).

accuracy: 72.00% +/- 10.33% (micro average: 72.00%)

	true D	true C	true CL	class precision
pred. D	22	12	3	59.46%
pred. C	10	50	3	79.37%
pred. CL	0	0	0	0.00%
class recall	68.75%	80.65%	0.00%	

Figure 163. 2-layer neural network cost-trained confusion matrix (non-trial).

accuracy: 74.02% +/- 5.82% (micro average: 74.04%)

	true D	true C	true CL	class precision
pred. D	103	39	9	68.21%
pred. C	22	128	10	80.00%
pred. CL	0	1	0	0.00%
class recall	82.40%	76.19%	0.00%	

Figure 164. Neural network cost-trained confusion matrix (trial).

accuracy: 74.31% +/- 7.25% (micro average: 74.36%)

	true D	true C	true CL	class precision
pred. D	100	36	10	68.49%
pred. C	25	132	9	79.52%
pred. CL	0	0	0	0.00%
class recall	80.00%	78.57%	0.00%	

Figure 165. 2-layer neural network cost-trained confusion matrix (trial).

Deep Learning

accuracy: 70.00% +/- 18.86% (micro average: 70.00%)

	true D	true C	true CL	class precision
pred. D	24	14	2	60.00%
pred. C	8	46	4	79.31%
pred. CL	0	2	0	0.00%
class recall	75.00%	74.19%	0.00%	

Figure 166. Deep learning cost-trained confusion matrix (non-trial).

accuracy: 74.68% +/- 8.10% (micro average: 74.68%)

	true D	true C	true CL	class precision
pred. D	108	44	11	66.26%
pred. C	15	124	7	84.93%
pred. CL	2	0	1	33.33%
class recall	86.40%	73.81%	5.26%	

Figure 167. Deep learning cost-trained confusion matrix (trial).

Conclusions

The following table includes the results from the trained models, displaying the accuracy, cost, and confidence intervals for each model.

Model	Accuracy	Cost
Decision tree (non-trial)	60.00% ± 16.33%	2.870 ± 1.127
Decision tree (trial)	71.45% ± 6.06%	2.044 ± 0.506
Rule-based (non-trial)	67.00% ± 14.94%	2.440 ± 1.069
Rule-based (trial)	72.72% ± 6.70%	1.999 ± 0.476
Naive Bayes (non-trial)	67.00% ± 15.67%	2.440 ± 1.178
Naive Bayes (trial)	75.01% ± 9.55%	1.948 ± 0.635
Bayesian network (non-trial)	70.00% ± 14.14%	2.190 ± 1.141
Bayesian network (trial)	73.34% ± 8.19%	1.198 ± 0.576
Neural network (non-trial)	70.00% ± 13.33%	2.220 ± 0.993
Neural network (trial)	74.02% ± 5.82%	1.892 ± 0.450
2-layer neural network (non-trial)	72.00% ± 10.33%	2.140 ± 0.871
2-layer neural network (trial)	74.31% ± 7.25%	1.932 ± 0.418
Deep learning (non-trial)	70.00% ± 18.86%	2.120 ± 1.346
Deep learning (trial)	74.68% ± 8.10%	1.814 ± 0.591

Table 3. Accuracy and cost performance of different models trained based on cost.

Based on the results from the cost-based training models, the trial group consistently outperforms the non-trial group in both accuracy and cost efficiency. The 2-layer neural network (trial) stands out as the most effective model across both accuracy and cost metrics. Achieving

a high accuracy of 74.31% with a relatively low cost of 1.932 ± 0.418, it demonstrates stable and strong performance. This model's relatively narrow confidence interval indicates that it can be relied upon in environments where both accurate and efficient predictions are needed. The Naive Bayes (trial) model performs remarkably well with an impressive accuracy of 75.01% and a cost of 1.948 ± 0.635. Though it has a wider confidence interval compared to the 2-layer neural network, it still presents a strong case for use, particularly when high accuracy is required. The variability in cost and accuracy may be less of a concern in scenarios where predictive certainty is less critical. The rule-based (trial) and Bayesian network (trial) models also deliver good results. The Bayesian network model, in particular, stands out for its consistency and lower cost (1.198 ± 0.576), making it a reliable choice, especially in data-intensive environments. However, its complexity may limit its use in simpler, faster deployment scenarios, where models like the Naive Bayes or 2-layer neural network are preferred.

While non-trial group models perform reasonably well, they generally lag behind the trial group in both accuracy and cost-efficiency. The decision tree (non-trial) has the lowest accuracy (60.00%) and the widest confidence interval. It struggles to deliver stable results, particularly in smaller datasets like this one. Its performance variability highlights the challenges of working with limited data. The deep learning (non-trial) model achieves 70.00% accuracy with a cost of 2.120 ± 1.346. Although it provides decent performance, its higher cost and lower accuracy compared to models like the Naive Bayes and 2-layer neural network make it less attractive for scenarios that prioritize both accuracy and cost efficiency.

In conclusion, models like the 2-layer neural network (trial) and Naive Bayes (trial) offer the best balance of accuracy and cost efficiency, making them ideal for real-world deployment. Although they are not as simple as the decision tree or rule-based models, their strong performance and relatively easier interpretability compared to the more complex Bayesian network model make them more suitable for a wide range of applications where quick decision-making and model transparency are necessary. These models provide a solid compromise between predictive power and practical usability, especially in environments that require both high performance and efficient resource utilization.

3.2.2 MERGED GROUPS: COMBINING TRIAL AND NON-TRIAL DATA

A. Accuracy-trained performance

Decision Tree

accuracy: 68.19% +/- 7.79% (micro average: 68.20%)

	true D	true C	true CL	class precision
pred. D	99	45	9	64.71%
pred. C	57	182	16	71.37%
pred. CL	1	3	0	0.00%
class recall	63.06%	79.13%	0.00%	

Figure 168. Decision tree accuracy-trained confusion matrix (combined).

Rule-based Model

accuracy: 72.10% +/- 6.77% (micro average: 72.09%)

	true D	true C	true CL	class precision
pred. D	108	39	9	69.23%
pred. C	44	185	12	76.76%
pred. CL	5	6	4	26.67%
class recall	68.79%	80.43%	16.00%	

Figure 169. Rule-based model accuracy-trained confusion matrix (combined).

Naive Bayes

accuracy: 72.08% +/- 8.56% (micro average: 72.09%)

	true D	true C	true CL	class precision
pred. D	100	27	8	74.07%
pred. C	54	196	16	73.68%
pred. CL	3	7	1	9.09%
class recall	63.69%	85.22%	4.00%	

Figure 170. Naive Bayes accuracy-trained confusion matrix (combined).

Bayesian Network

accuracy: 71.59% +/- 6.16% (micro average: 71.60%)

	true D	true C	true CL	class precision
pred. D	105	40	13	66.46%
pred. C	50	189	11	75.60%
pred. CL	2	1	1	25.00%
class recall	66.88%	82.17%	4.00%	

Figure 171. Bayesian network accuracy-trained confusion matrix (combined).

Neural Networks

accuracy: 72.12% +/- 5.66% (micro average: 72.09%)

	true D	true C	true CL	class precision
pred. D	105	36	5	71.92%
pred. C	51	192	20	73.00%
pred. CL	1	2	0	0.00%
class recall	66.88%	83.48%	0.00%	

Figure 172. Neural network accuracy-trained confusion matrix (combined).

accuracy: 73.31% +/- 3.54% (micro average: 73.30%)

	true D	true C	true CL	class precision
pred. D	110	38	9	70.06%
pred. C	47	192	16	75.29%
pred. CL	0	0	0	0.00%
class recall	70.06%	83.48%	0.00%	

Figure 173. 2-layer neural network accuracy-trained confusion matrix (combined).

Deep Learning

accuracy: 67.97% +/- 6.13% (micro average: 67.96%)

	true D	true C	true CL	class precision
pred. D	103	50	10	63.19%
pred. C	48	173	11	74.57%
pred. CL	6	7	4	23.53%
class recall	65.61%	75.22%	16.00%	

Figure 174. Deep learning accuracy-trained confusion matrix (combined).

Conclusions

Model	Accuracy	Cost
Decision tree	68.19 % ± 7.79%	2.533 ± 0.597
Rule-based	72.10% ± 6.75%	2.247± 0.543
Naive Bayes	72.08% ± 8.56%	2.337 ± 0.682
Bayesian network	71.59% ± 6.16%	2.315 ± 0.455
Neural network	72.12% ± 5.66%	2.242 ± 0.474
2-layer neural network	73.31 % ± 3.54%	2.129 ± 0.267
Deep Learning	67.97%±6.13%	2.531 ± 0.531

Table 4. Accuracy and cost performance of different models (combined data) trained based on accuracy.

In general, combining both trial and non-trial data provides better generalization for most models. This combination is valuable because it incorporates both clinical study data (from the trial group) and real-world hospital data (from the non-trial group). Notably, the 2-layer Neural network achieves the highest accuracy in the combined dataset (73.31%), like the performance in the trial dataset alone (74.71%). When comparing the cost performance, the combined group appears more efficient, with 2-layer neural networks achieving the lowest cost of 2.129 compared to 1.974 in the trial-only models, suggesting that the additional dataset provides a more robust model despite slightly higher complexity. Actually, for most models, combining the trial and non-trial data leads to more consistent results, with some models achieving better accuracy and lower cost.

B. Cost-trained performance

Decision Tree

accuracy: 68.94% +/- 6.55% (micro average: 68.93%)

	true D	true C	true CL	class precision
pred. D	113	57	12	62.09%
pred. C	43	171	13	75.33%
pred. CL	1	2	0	0.00%
class recall	71.97%	74.35%	0.00%	

71.97%

Figure 175. Decision tree cost-trained confusion matrix (combined).

Rule-based Model

accuracy: 70.14% +/- 5.30% (micro average: 70.15%)

	true D	true C	true CL	class precision
pred. D	118	58	11	63.10%
pred. C	39	170	13	76.58%
pred. CL	0	2	1	33.33%
class recall	75.16%	73.91%	4.00%	

Figure 176. Rule-based model cost-trained confusion matrix (combined).

Naive Bayes

accuracy: 71.59% +/- 5.47% (micro average: 71.60%)

	true D	true C	true CL	class precision
pred. D	114	47	10	66.67%
pred. C	41	179	13	76.82%
pred. CL	2	4	2	25.00%
class recall	72.61%	77.83%	8.00%	

Figure 177. Naive Bayes cost-trained confusion matrix (combined).

Bayesian Network

accuracy: 71.36% +/- 9.25% (micro average: 71.36%)

	true D	true C	true CL	class precision
pred. D	125	60	14	62.81%
pred. C	31	169	11	80.09%
pred. CL	1	1	0	0.00%
class recall	79.62%	73.48%	0.00%	

Figure 178. Bayesian network accuracy-trained confusion matrix (combined).

Neural Networks

accuracy: 71.85% +/- 4.41% (micro average: 71.84%)

	true D	true C	true CL	class precision
pred. D	121	55	12	64.36%
pred. C	36	175	13	78.12%
pred. CL	0	0	0	0.00%
class recall	77.07%	76.09%	0.00%	

Figure 179. Neural network cost-trained confusion matrix (combined).

accuracy: 71.82% +/- 6.38% (micro average: 71.84%)

	true D	true C	true CL	class precision
pred. D	121	55	13	64.02%
pred. C	36	175	12	78.48%
pred. CL	0	0	0	0.00%
class recall	77.07%	76.09%	0.00%	

Figure 180. 2-layer neural network cost-trained confusion matrix (combined).

Deep Learning

accuracy: 66.96% +/- 5.53% (micro average: 66.99%)

	true D	true C	true CL	class precision
pred. D	117	69	13	58.79%
pred. C	39	156	9	76.47%
pred. CL	1	5	3	33.33%
class recall	74.52%	67.83%	12.00%	

Figure 181. Deep learning cost-trained confusion matrix (combined).

Conclusions

Model	Accuracy	Cost
Decision tree	68.94 % ± 6.55%	2.340 ± 0.356
Rule-based	70.14% ± 5.30%	2.195± 0.407
Naive Bayes	71.59% ± 5.47%	2.183 ± 0.459
Bayesian network	71.36% ± 9.25%	2.082 ± 0.636
Neural network	71.85% ± 4.41%	2.085 ± 0.302
2-layer neural network	71.82 % ± 6.38%	2.092 ± 0.452
Deep Learning	66.96 %±5.53%	2.374 ± 0.520

Table 5. Accuracy and cost performance of different models (combined data) trained based on cost.

For cost-based training, combining the trial and non-trial data results in more stable and reliable performance across various models. While some models show a slight decrease in accuracy, the overall cost of misclassifications is reduced. This suggests that including real-world data helps the models generalize better, balancing the trade-off between accuracy and the cost associated with errors. Thus, combining the trial and non-trial data not only enhances model accuracy and reduces error costs but also ensures the model is better equipped to handle real-world clinical scenarios.

Model Training and Evaluation in Python

In this project, several machine learning models were trained and evaluated using Python, as explained in the Google Collab notebooks of Abraham Otero. Their performance was evaluated based on how well they balanced accuracy with cost, ensuring efficient and reliable

predictions. The models considered include decision trees, Naive Bayes, neural networks with one and two hidden layers, and deep learning models. Bayesian networks were not included as they are not explained in the notebook 4 of training models and deep learning models were not trained based on cost due to time limitations.

To ensure robust and reliable results, cross-validation was employed for model evaluation. The number of folds in k-fold cross-validation was dynamically adjusted based on the number of samples available per class. The number of folds was set to 10 or the smallest class size, whichever was smaller. This helped ensure that each fold had enough data to represent all classes. This methodology ensured that the machine learning models were evaluated in a robust manner, accounting for class imbalance and providing reliable predictions across both accuracy and cost metrics. To understand the implementation of the models, the selection of the number of folds, and the complete code, you can access the detailed explanation and code snippets through the link mentioned in the Introduction section.

4.1 Training Models

4.1.1 TRAINING SEPARATE GROUPS: TRIAL AND NON-TRIAL DATA

A. Accuracy-trained models

Decision tree

The *DecisionTreeClassifier* was trained for both the trial and non-trial groups. For the trial group, 10-fold cross-validation was used. *GridSearchCV* was employed to find the best hyperparameters for the *DecisionTreeClassifier* to maximize accuracy. The optimal hyperparameters found were:

- *max_depth*: 5
- *min_samples_leaf*: 2
- *min_samples_split*: 5

For the non-trial group, 6-fold cross-validation was used due to fewer available samples. Similarly to the trial group, *GridSearchCV* was used to determine the best hyperparameters for the *DecisionTreeClassifier*. The optimal hyperparameters found were:

- *max_depth*: 5
- *min_samples_leaf*: 2
- *min_samples_split*: 2

For each fold in both the trial and non-trial groups, the decision tree is printed. However, only the decision tree for the first fold is provided here. If you wish to view the decision trees

for all folds, you can refer to the code, where the decision tree is visualized and displayed for each fold separately during the cross-validation process.

In the trial group's decision tree (1st fold), Bilirubin and Prothrombin dominate the upper levels of the decision tree, indicating their significant role in splitting the data. Bilirubin serves as the root node, highlighting its importance as a primary predictor. Prothrombin follows closely, appearing as a secondary feature and further refining the classification process. Age plays a supporting role, appearing in the deeper branches of the tree. This suggests that while it is not the primary feature driving the initial splits, it contributes to fine-tuning the classification. Features like SGOT, Platelets, and Albumin appear later in the tree, indicating that they are less influential but still important for refining the final classification in combination with the more dominant features.

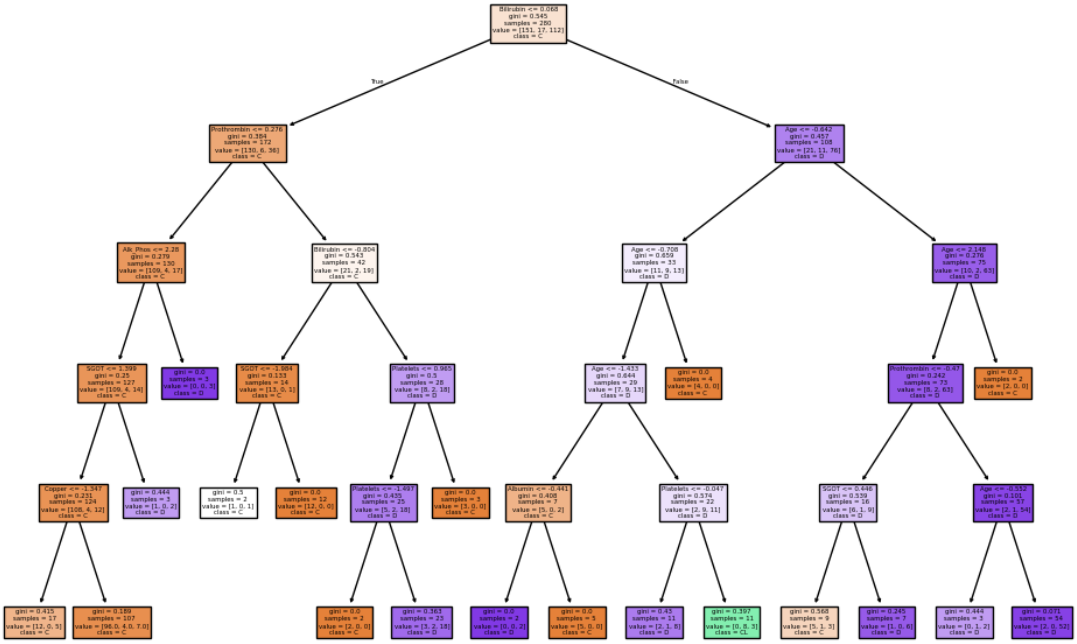


Figure 182. Decision tree trained based on accuracy using Python (trial).

In the non-trial group's decision tree (1st fold), Bilirubin and Prothrombin also serve as important features at the upper levels of the decision tree. Edema appears in the decision tree as an important factor, helping to make distinctions in cases where the Bilirubin levels are within a particular range. It plays a key role in identifying class C patients. Other features, such as Age, Platelets, and Albumin, contribute to the deeper branches of the tree. Age helps further differentiate the C and D classes in more subtle ways.

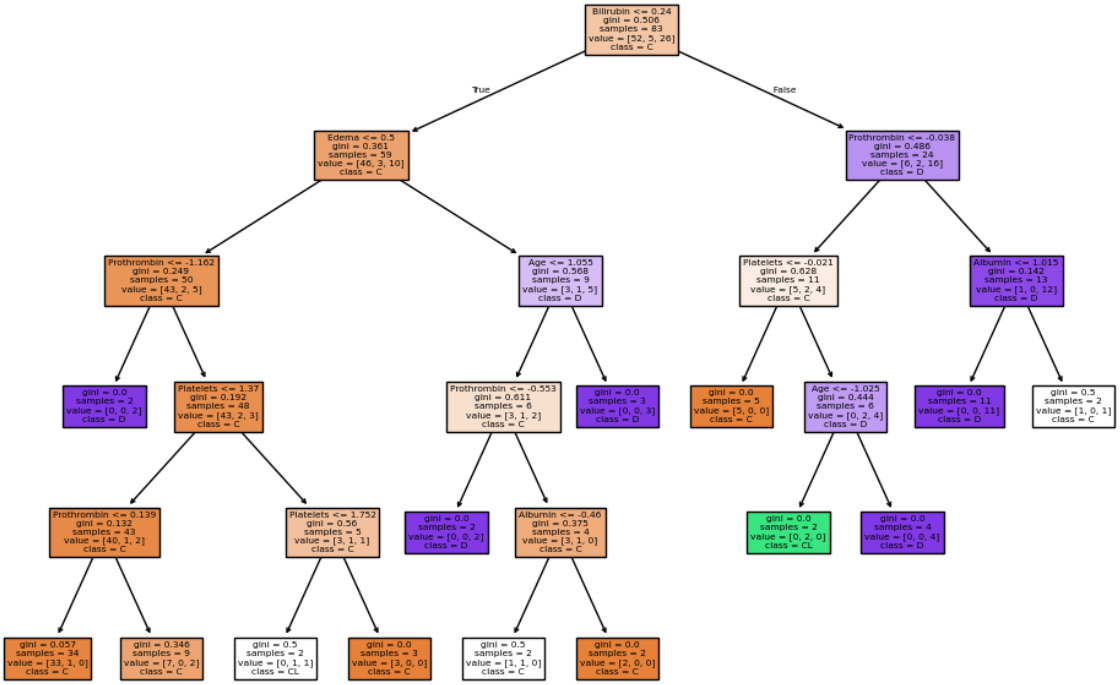


Figure 183. Decision tree trained based on accuracy using Python (non-trial).

Naive Bayes

In this training process, Gaussian Naive Bayes is used, and the hyperparameter *var_smoothing* is optimized with *GridSearchCV*. The *var_smoothing* hyperparameter adds a small value to the variance estimates to avoid division by zero and to enhance the stability of predictions, especially when the model faces very small variance values. The parameter grid for *var_smoothing* explores values on a logarithmic scale ranging from 10^{-9} to 10^{-2}, allowing for fine-tuning of the model's performance in terms of both bias and variance. Since GNB can effectively handle metric data and is easy to implement with minimal computational requirements, it was chosen for this project, especially given that the majority of the features were numeric. Although nominal data was present, it was appropriately transformed. The use of 10-fold cross-validation for the trial group and 6-fold cross-validation for the non-trial group remains unchanged.

Neural Networks

Two different neural networks have been trained. The first model is a multi-layer perceptron (MLP) with a single hidden layer consisting of 1000 neurons. The second model is also an MLP, but it has two hidden layers with varying sizes: 750 neurons in the first hidden layer and 500 neurons in the second. Each configuration allows the models to learn different levels of complexity in the data, with the number of layers and neurons affecting their ability to capture various patterns.

The activation function used for the hidden layers in both models is ReLU (Rectified Linear Unit) due to its ability to handle non-linearity, improve convergence, and mitigate the vanishing gradient problem. The optimizer used for training is Adam. It combines the benefits of two other algorithms (Momentum and RMSProp) and adapts the learning rate for each parameter during training. Alpha is set to 0.0001. This is the L2 regularization parameter, which helps prevent overfitting by penalizing large weights in the model. The learning rate for the optimizer is adaptive, meaning it changes during training based on the model's progress. The learning rate is increased when the model is far from the optimal solution and decreased when the model approaches the solution, improving convergence. Both models will stop after 1000 iterations or once convergence is achieved. Finally, the random state is set to 42.

Deep Learning

The deep learning neural network model is configured with three hidden layers and uses the SoftMax activation function in the output layer. The first hidden layer consists of 64 neurons with ReLU activation, followed by a dropout layer set at 30% to mitigate overfitting by randomly setting 30% of the input units to zero during training, helping the model generalize better. The second hidden layer has 32 neurons and uses the ReLU activation function, with another 30% dropout layer to continue preventing overfitting. The third hidden layer contains 16 neurons and applies ReLU activation, further refining the model's ability to capture intricate patterns. The output layer has a few neurons corresponding to the number of classes in the target variable and uses the SoftMax activation function to output probabilities for each class. The model is compiled using the Adam optimizer with a learning rate of 0.001, which adapts to the parameters during training. Categorical cross-entropy is used as the loss function, making it suitable for this 3-class classification task, comparing predicted probabilities against the actual class labels to measure performance.

B. Cost-trained models

All the models trained based on cost, used the MetaCost approach with the cost matrix defined earlier to trained them not only to maximize accuracy but also to minimize the associated costs of errors, making them more suitable for real-world scenarios where the consequences of misclassifying a patient's condition could be significant.

Decision tree

The image bellow displays the structure of a decision tree trained for the trial group based on cost, using the best parameters obtained from a *GridSearchCV*. This decision tree has a maximum depth of 5, with a minimum of 2 samples per leaf and a requirement of at least 5 samples to split a node. The training process included 10-fold cross-validation to ensure robustness and reduce overfitting.

The decision tree identifies Bilirubin, Prothrombin, and Age as key features driving the classification decisions. These features appear at critical decision points (such as the root and higher-level nodes), indicating their significance in predicting the patient's status. Bilirubin, for instance, plays a central role in the initial split, suggesting its strong influence on classification outcomes.

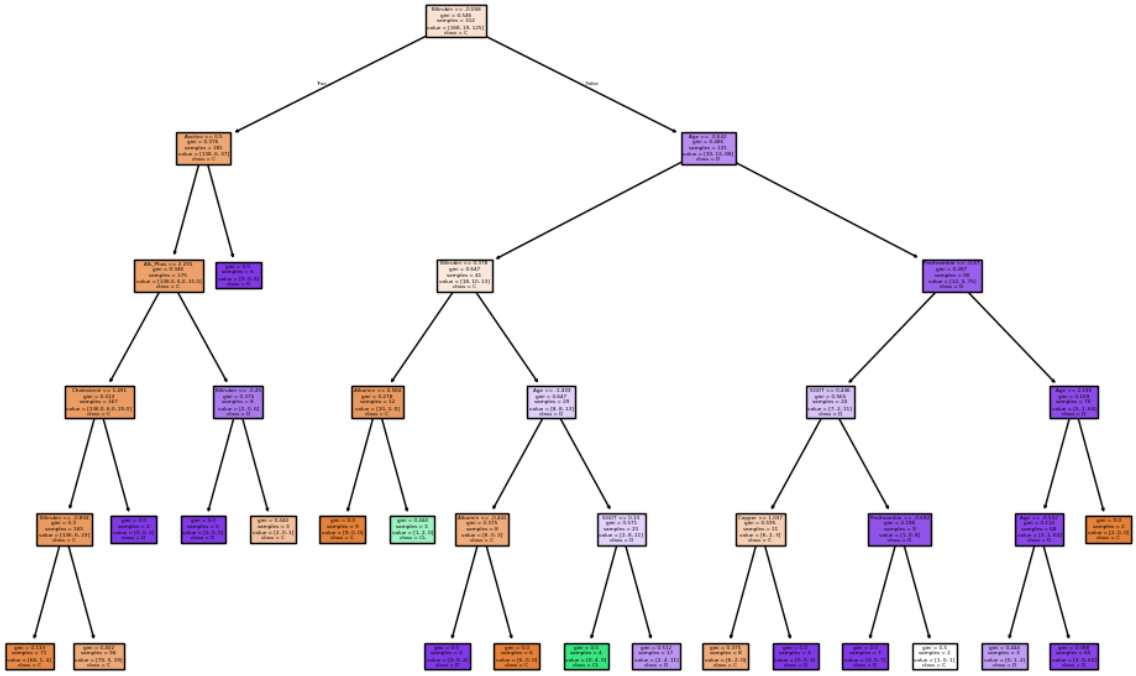


Figure 184. Decision tree trained based on cost using Python (trial).

The decision tree for the non-trial group, with the optimized hyperparameters of max_depth = 10, min_samples_split = 2, and min_samples_leaf = 1, was trained on 6 folds. Features such as Bilirubin, Prothrombin, and Platelets dominate the decision tree's splits, showcasing their importance in determining the class. Specifically, Bilirubin and Prothrombin are located at the root and higher levels of the tree, suggesting their crucial role in classification.

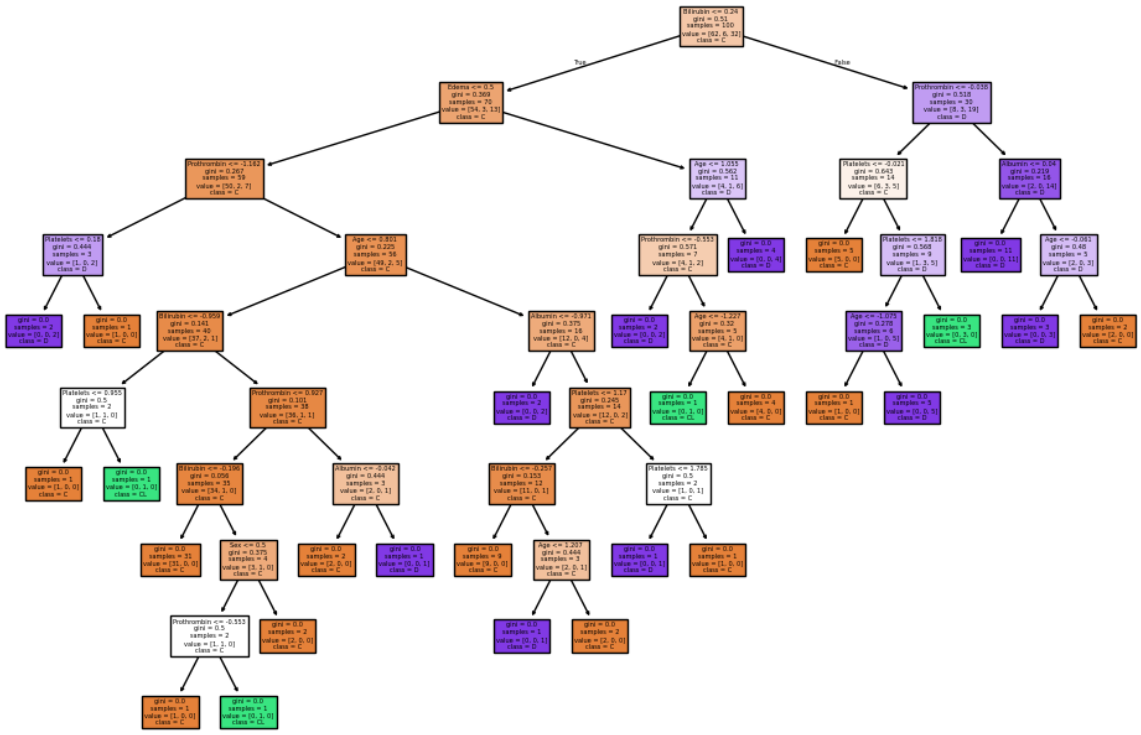


Figure 185. Decision tree trained based on cost using Python (non-trial).

Naive Bayes

For the cost-based training, the same Gaussian Naive Bayes configuration and the same number of folds for each group as in the accuracy-trained models were used, but with MetaCost applied to account for the cost matrix.

Neural Networks

In the cost-based training, two neural networks are retrained using the same configurations as the accuracy-based training. These models maintain the same hyperparameters and configurations as in the accuracy-based training.

4.1.2 MERGED GROUPS: COMBINING TRIAL AND NON-TRIAL DATA

A. Accuracy-trained models

Decision tree

The configuration used for training the decision tree on the combined groups is the same as the one used for the decision tree on the trial group. The decision tree obtained in the first fold is shown above:

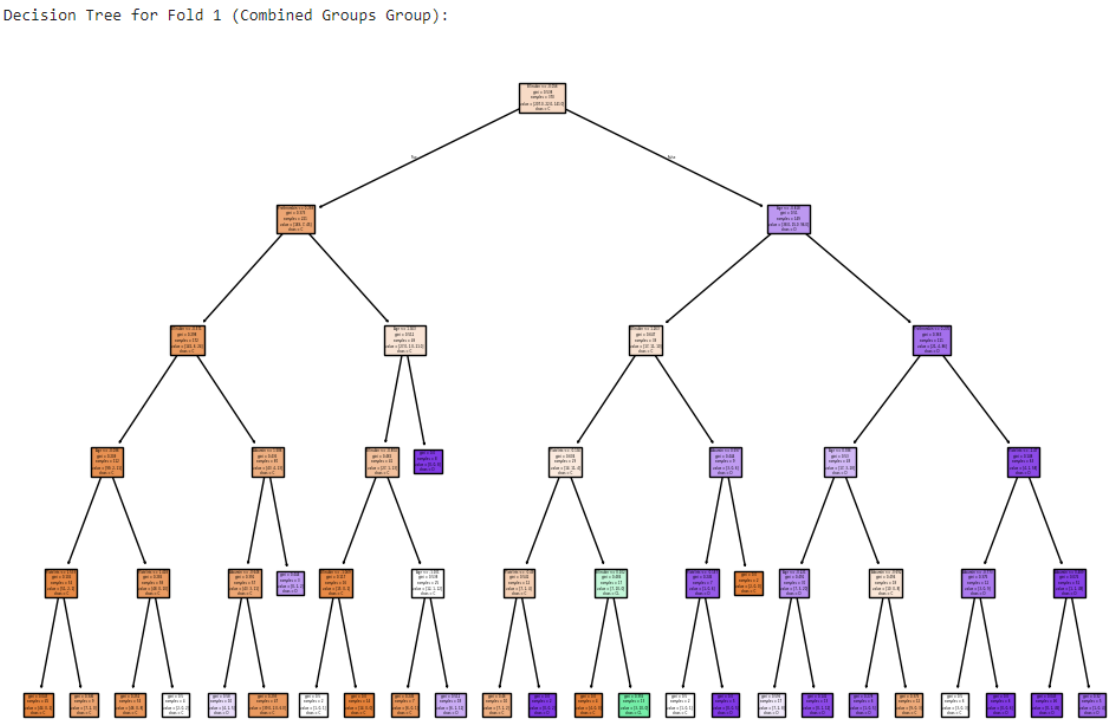


Figure 186. Decision tree trained based on accuracy using Python (combined).

Naive Bayes, Neural Networks and Deep Learning Neural Network

For these three models, the same configuration used for the Trial group has been applied. For the neural networks, two different configurations were used: one with a single hidden layer and the other with two hidden layers.

B. Cost-trained models

Decision tree

The configuration used for training the decision tree on the combined groups is the same as the one used for the cost-trained decision tree on the trial group. The decision tree obtained in the first fold is shown above:

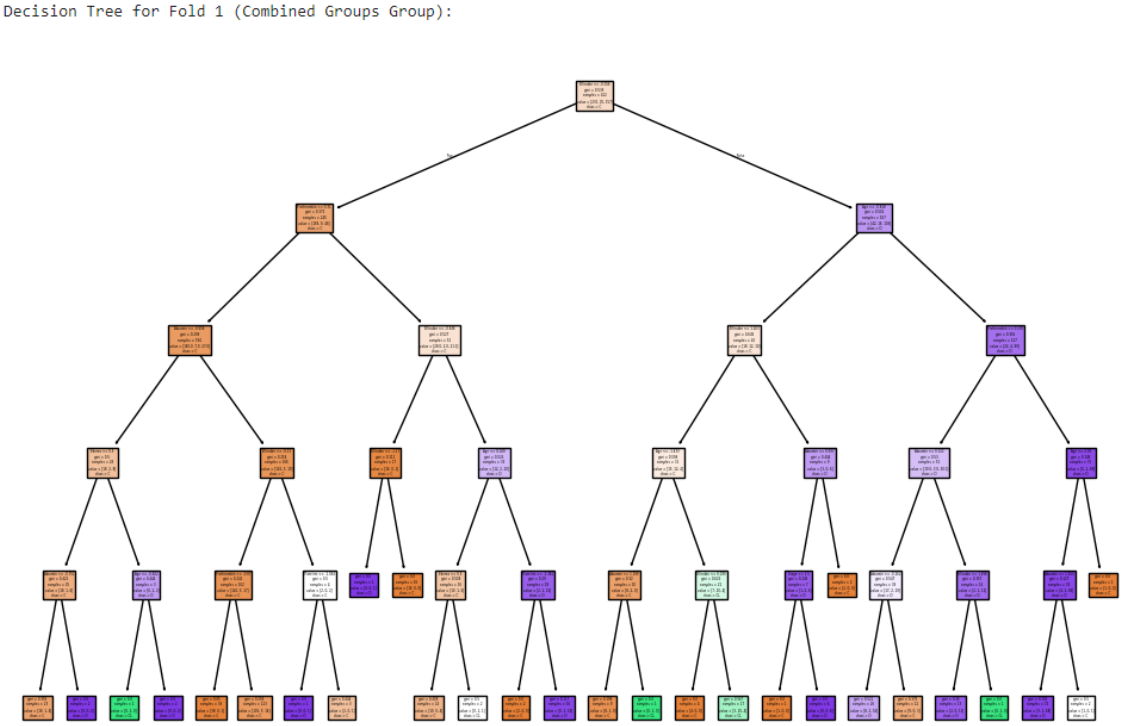


Figure 187. Decision tree trained based on cost using Python (combined).

Naive Bayes and Neural Networks

For these two models, the same configuration used for the Trial group has been applied. For the neural networks, two different configurations were used: one with a single hidden layer and the other with two hidden layers.

4.2 Evaluation of Models

For each model in every fold, the confusion matrix is printed and visualized using *ConfusionMatrixDisplay*. This allows for a clear graphical representation of how the model classifies instances into each of the possible categories. At the end of the evaluation across all folds, the following performance metrics are calculated and printed:

- Mean accuracy is the average of the accuracy scores across all folds. It gives an overall sense of how well the model performed on average across different subsets of the data during cross-validation.
- Standard deviation of accuracy measures the variability or dispersion of accuracy scores across all folds. A high standard deviation indicates that the model's performance is inconsistent across folds, while a low standard deviation suggests stable performance.
- Mean cost is the average of the total costs calculated for each fold. It gives an overall sense of how much misclassification cost the model incurred on average across different subsets of the data.

- Standard deviation of cost measures the variability or dispersion of the cost values across all folds. A higher standard deviation indicates that the model's performance (in terms of cost) is inconsistent across folds, while a lower standard deviation suggests more consistent performance.
- The mean cost per instance indicates the average cost normalized per instance, offering a more granular view of misclassification penalties at the instance level.
- The standard deviation of cost per instance captures the variability of the per-instance cost, showing how stable the cost efficiency is across individual data points.

In following sections, the confusion matrices for each fold are not displayed in full. Instead, only the final results are shown. Additionally, one example of a confusion matrix from a single fold is presented to illustrate the model's performance on a particular subset of the data.

In addition, for the deep learning model, the training history for each fold is displayed, showing the progression of both training and validation metrics. This includes:

- Training loss: represents the error or discrepancy between the model's predictions and the actual values on the training set. A decreasing training loss indicates that the model is improving its fitness to the training data over time.
- Validation loss: reflects the error on the validation set, which is used to evaluate the model's ability to generalize to unseen data. Ideally, the validation loss should decrease as the training progresses, but if it starts to increase, it may indicate overfitting.
- Training accuracy: shows the percentage of correct predictions made by the model on the training set. A steady increase in training accuracy suggests that the model is learning effectively from the data.
- Validation accuracy: represents the percentage of correct predictions on the validation set, providing an indication of how well the model is generalizing to unseen data.

4.2.1 EVALUATING SEPARATE GROUPS: TRIAL AND NON-TRIAL DATA

A. Accuracy-trained models

Decision tree

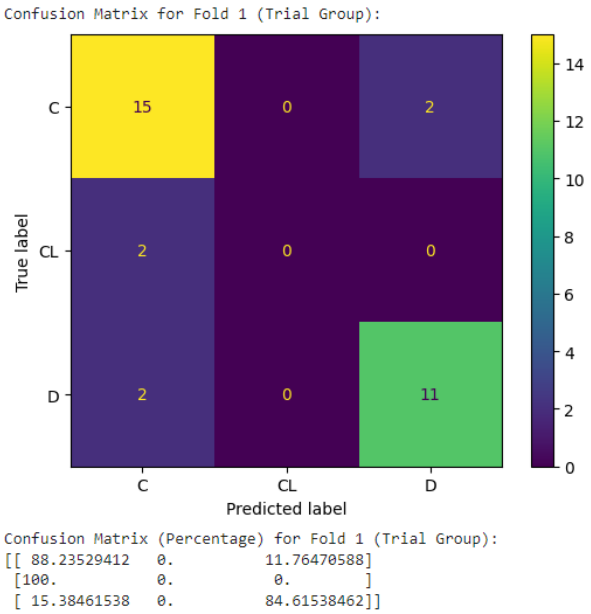


Figure 188. Decision tree confusion matrix trained based on accuracy (trial).

```
Results for Trial Group:
Mean accuracy: 0.7336693548387097
Standard deviation of accuracy: 0.0715453151155573
Mean cost: 70.2
Standard deviation of cost: 17.78651174345324
Mean cost per instance: 2.264516129032258
Standard deviation of cost per instance: 0.5737584433372013
```

Figure 189. Results for decision tree confusion matrix trained based on accuracy (trial).

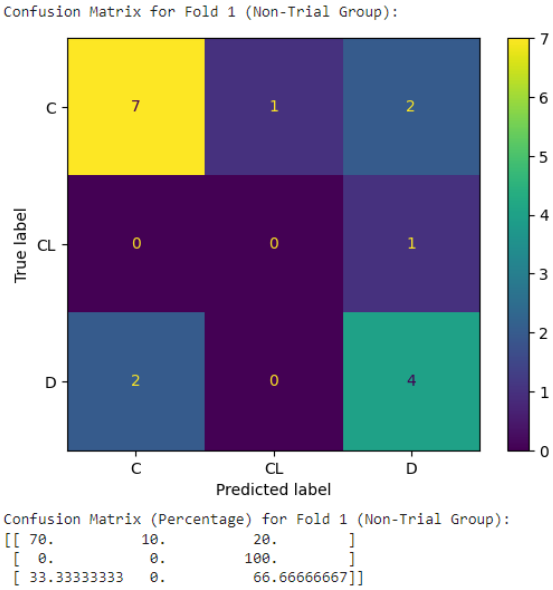
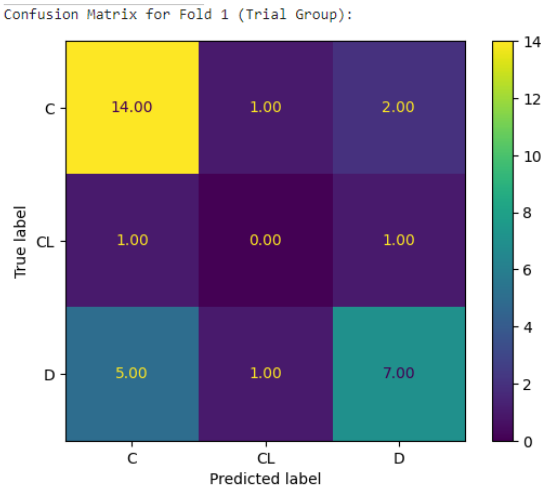


Figure 190. Decision tree confusion matrix trained based on accuracy (non-trial).

```
Results for Non-Trial Group:
Mean accuracy: 0.689950980392157
Standard deviation of accuracy: 0.07512564766663776
Mean cost: 41.5
Standard deviation of cost: 12.065791865158845
Mean cost per instance: 2.59375
Standard deviation of cost per instance: 0.7541119915724278
```

Figure 191. Results for decision tree confusion matrix trained based on accuracy (non-trial).

Naive Bayes

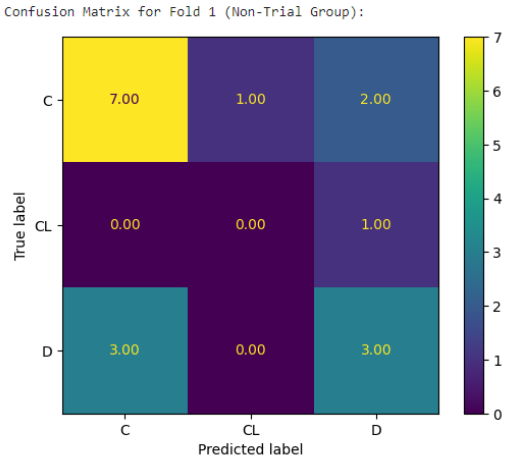



```
Confusion Matrix (Percentage) for Fold 1 (Trial Group):
[[82.35294118  5.88235294 11.76470588]
 [50.          0.         50.        ]
 [38.46153846  7.69230769 53.84615385]]
```

Figure 192. Naive Bayes confusion matrix trained based on accuracy (trial).

```
Results for Trial Group:
Mean accuracy: 0.7147177419354839
Standard deviation of accuracy: 0.0543831887395525
Mean cost: 83.2
Standard deviation of cost: 13.709850473291093
Mean cost per instance: 2.6838709677419357
Standard deviation of cost per instance: 0.44225324107390623
```

Figure 193. Results for Naive Bayes confusion matrix trained based on accuracy (trial).




```
Confusion Matrix (Percentage) for Fold 1 (Non-Trial Group):
[[ 70.  10.  20.]
 [  0.   0. 100.]
 [ 50.   0.  50.]]
```

Figure 194. Naive Bayes confusion matrix trained based on accuracy (non-trial).

```
Results for Non-Trial Group:
Mean accuracy: 0.6795343137254902
Standard deviation of accuracy: 0.07707203227735586
Mean cost: 44.0
Standard deviation of cost: 14.236104336041748
Mean cost per instance: 2.75
Standard deviation of cost per instance: 0.8897565210026093
```

Figure 195. Results for Naive Bayes confusion matrix trained based on accuracy (non-trial).

Neural Networks

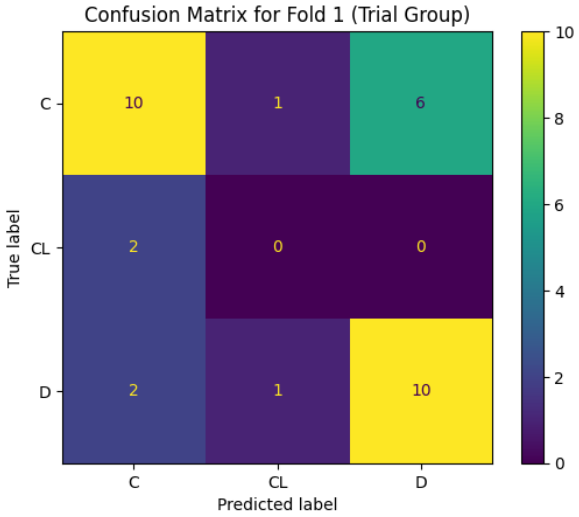


Figure 196. Neural network confusion matrix trained based on accuracy (trial).

```
Results for Trial Group:
Mean accuracy: 0.6827620967741936
Standard deviation of accuracy: 0.06452250787517952
Mean cost: 78.7
Standard deviation of cost: 14.866405079910882
Mean cost per instance: 2.538709677419355
Standard deviation of cost per instance: 0.4795614541906735
```

Figure 197. Results for neural network confusion matrix trained based on accuracy (trial).

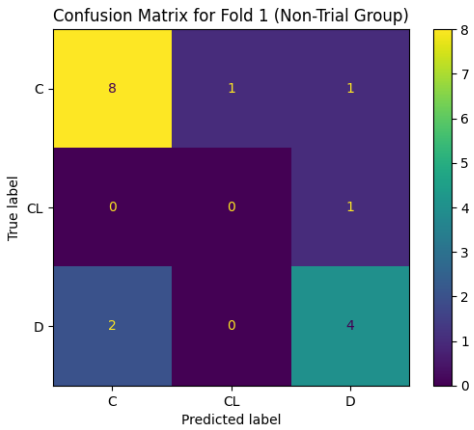


Figure 198. Neural network confusion matrix trained based on accuracy (non-trial).

```
Results for Non-Trial Group:
Mean accuracy: 0.6905637254901961
Standard deviation of accuracy: 0.10685859915055175
Mean cost: 40.833333333333336
Standard deviation of cost: 16.48652648545337
Mean cost per instance: 2.5520833333333335
Standard deviation of cost per instance: 1.0304079053408357
```

Figure 199. Results for neural network confusion matrix trained based on accuracy (non-trial).

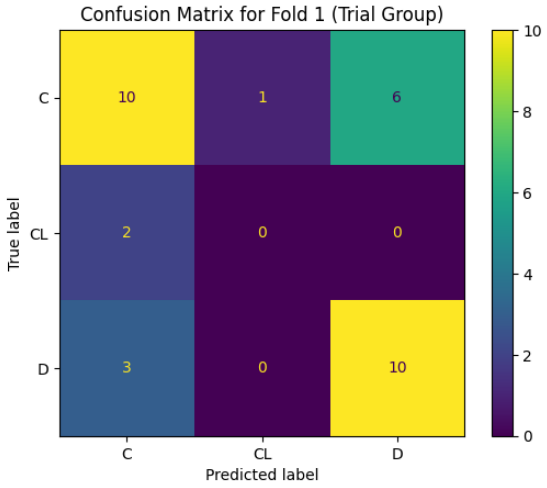


Figure 200. 2-layer neural network confusion matrix trained based on accuracy (trial).

```
Results for Trial Group:
Mean accuracy: 0.6954637096774194
Standard deviation of accuracy: 0.07487205365103884
Mean cost: 75.4
Standard deviation of cost: 18.7840357750937
Mean cost per instance: 2.432258064516129
Standard deviation of cost per instance: 0.6059366379062485
```

Figure 201. Results for 2-layer neural network confusion matrix trained based on accuracy (trial).

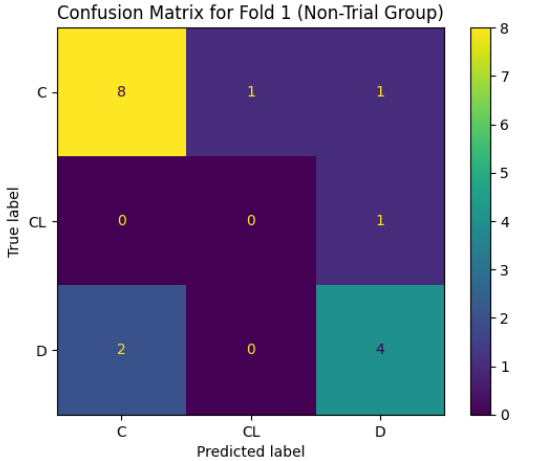


Figure 202. 2-layer neural network confusion matrix trained based on accuracy (non-trial).

```
Results for Non-Trial Group:
Mean accuracy: 0.6905637254901961
Standard deviation of accuracy: 0.10685859915055175
Mean cost: 40.0
Standard deviation of cost: 15.165750888103101
Mean cost per instance: 2.5
Standard deviation of cost per instance: 0.9478594305064438
```

Figure 203. Results for 2-layer neural network confusion matrix trained based on accuracy (non-trial).

Deep Learning

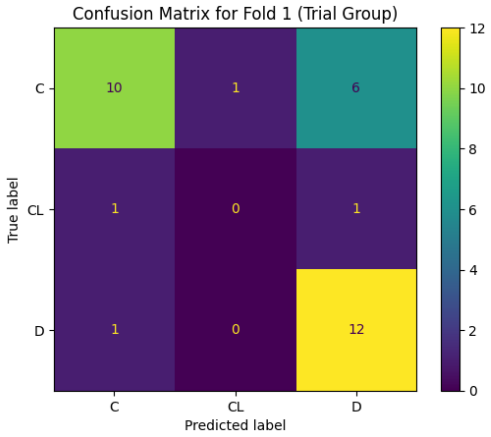


Figure 204. Deep learning confusion matrix trained based on accuracy (trial).

```
Results for Trial Group:
Mean accuracy: 0.7308467741935484
Standard deviation of accuracy: 0.08877036963409574
Mean cost: 60.3
Standard deviation of cost: 23.828764130772708
Mean cost per instance: 1.9451612903225808
Standard deviation of cost per instance: 0.7686698106700873
```

Figure 205. Results for deep learning confusion matrix trained based on accuracy (trial).

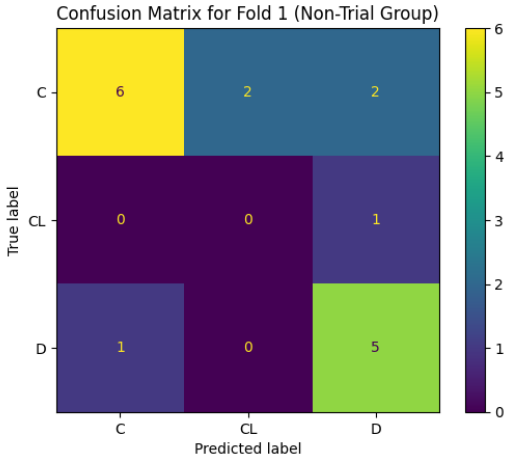


Figure 206. Deep learning confusion matrix trained based on accuracy (non-trial).

```
Results for Non-Trial Group:
Mean accuracy: 0.4889705882352941
Standard deviation of accuracy: 0.09989812428893509
Mean cost: 52.666666666666664
Standard deviation of cost: 12.81492185782739
Mean cost per instance: 3.2916666666666665
Standard deviation of cost per instance: 0.8009326161142118
```

Figure 207. Results for deep learning confusion matrix trained based on accuracy (non-trial).

Conclusions

The following table includes the results from the trained models, displaying the accuracy, cost and confidence intervals for each model.

Model	Accuracy	Cost per instance
Decision tree (non-trial)	68.99% ± 7.512%	2.59 ± 0.75
Decision tree (trial)	73.33% ± 7.154%	2.26 ± 0.57
Naive Bayes (non-trial)	67.95% ± 7.707%	2.75 ± 0.88
Naive Bayes (trial)	71.47% ± 5.438%	2.68 ± 0.44
Neural network (non-trial)	69.05% ± 10.685%	2.55 ± 1.03
Neural network (trial)	68.27% ± 6.452%	2.53 ± 0.47
2-layer neural network (non-trial)	69.05% ± 10.685%	2.5 ± 0.94
2-layer neural network (trial)	69.54% ± 7.487%	2.43 ± 0.60
Deep learning (non-trial)	48.89% ± 9.98%	3.29 ± 0.80
Deep learning (trial)	73.08% ± 8.87%	1.945 ± 0.768

Table 6. Accuracy and cost per instance performance of different models trained based on accuracy in Python.

In terms of accuracy, the trial group models generally perform better than the non-trial group models. Among the trial group models, decision tree and deep learning neural network achieve the highest accuracy values, with deep learning at 73.08% and decision tree at 73.33%. On the other hand, the non-trial group models show lower accuracy, especially deep learning,

which has the lowest accuracy at 48.89%. The other non-trial models, such as decision tree and naive bayes, perform better but still fall short compared to the trial group models.

Cost efficiency is crucial, especially in real-world applications like healthcare, where misclassifications can have severe consequences. Deep learning (trial) is the most cost-efficient model, with the lowest cost per instance of 1.945. Following closely is decision tree (trial) with a cost of 2.26 and 2-layer neural network (trial) with a cost of 2.43. In contrast, the non-trial group has relatively higher costs, especially with deep learning (non-trial) at 3.29, indicating more misclassifications in this model.

Regarding stability and consistency, the trial group models show narrower confidence intervals, which suggest more stable and consistent predictions across folds. For instance, decision tree (trial) and naive bayes (trial) demonstrate relatively narrow confidence intervals, indicating their stability. In comparison, the non-trial group models, especially Naive Bayes and deep learning, exhibit wider confidence intervals, suggesting that the models are less stable and consistent.

Interpretability is a critical factor in medical applications, where understanding why a model made a certain prediction is important. Decision trees are the most interpretable models, as they offer clear, rule-based decision paths. Naive Bayes models are also relatively simple but are based on probabilistic assumptions and are less interpretable than decision trees. Neural networks and deep learning models, while powerful and requiring greater computational resources, are much more complex and operate as black-box models, making them difficult to interpret.

Overall, decision tree (trial) stands out as the most balanced model, offering a good combination of accuracy, cost-efficiency, interpretability, and simplicity. While deep learning (trial) excels in accuracy and cost efficiency, its complexity limits its interpretability and simplicity, making it less desirable in situations where these factors are essential.

B. Cost-trained models

Decision tree

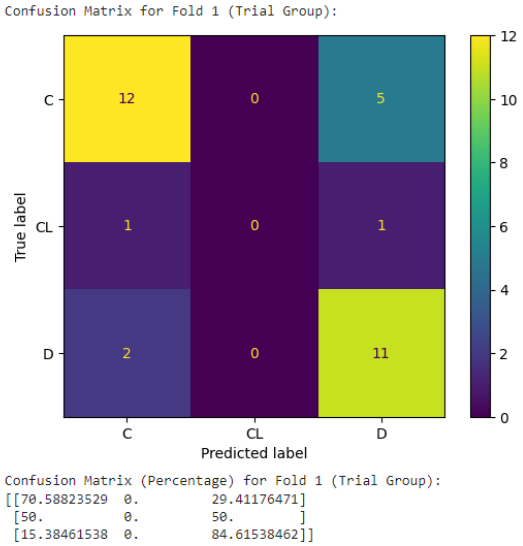


Figure 208. Decision tree confusion matrix trained based on cost (trial).

```
Results for Trial Group:
Mean accuracy: 0.7050403225806452
Standard deviation of accuracy: 0.06447705469953109
Mean cost: 72.5
Standard deviation of cost: 18.5377992221299
Mean cost per instance: 2.338709677419355
Standard deviation of cost per instance: 0.5979935232945129
```

Figure 209. Results for the decision tree confusion matrix trained based on cost (trial).

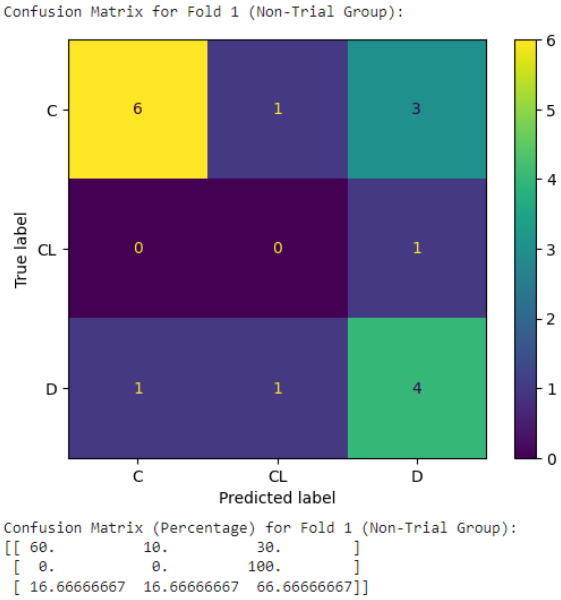


Figure 210. Decision tree confusion matrix trained based on cost (non-trial).

```
Results for Non-Trial Group:
Mean accuracy: 0.5575980392156863
Standard deviation of accuracy: 0.1566232086526713
Mean cost: 57.833333333333336
Standard deviation of cost: 18.451889394374284
Mean cost per instance: 3.6145833333333335
Standard deviation of cost per instance: 1.1532430871483927
```

Figure 211. Results for the decision tree confusion matrix trained based on cost (non-trial).

Naive Bayes

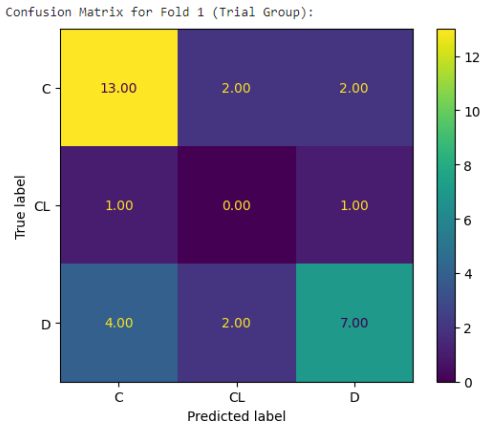


Figure 212. Naive Bayes confusion matrix trained based on cost (trial).

```
Results for Trial Group:
Mean accuracy: 0.7082661290322582
Standard deviation of accuracy: 0.07069716794293024
Mean cost: 80.1
Standard deviation of cost: 16.676030702778167
Mean cost per instance: 2.583870967741935
Standard deviation of cost per instance: 0.5379364742831666
```

Figure 213. Results for the Naive Bayes confusion matrix trained based on cost (trial).

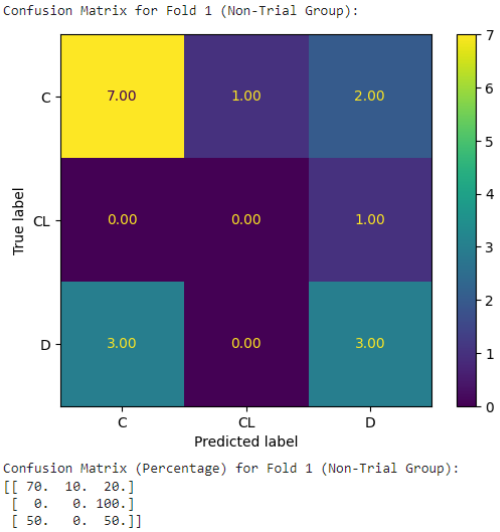


Figure 214. Naive Bayes confusion matrix trained based on cost (non-trial).

```
Results for Non-Trial Group:
Mean accuracy: 0.6599264705882354
Standard deviation of accuracy: 0.04379734627085401
Mean cost: 44.833333333333336
Standard deviation of cost: 9.281103861550568
Mean cost per instance: 2.8020833333333335
Standard deviation of cost per instance: 0.5800689913469105
```

Figure 215. Results for the Naive Bayes confusion matrix trained based on cost (non-trial).

Neural Networks

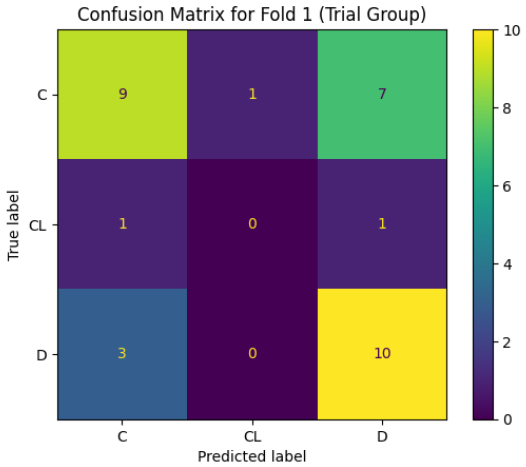


Figure 216. Neural network confusion matrix trained based on cost (trial).

```
Results for Trial Group:
Mean accuracy: 0.6924395161290323
Standard deviation of accuracy: 0.07675926437677663
Mean cost: 75.2
Standard deviation of cost: 16.59397481015323
Mean cost per instance: 2.4258064516129036
Standard deviation of cost per instance: 0.535289510004943
```

Figure 217. Results for the neural network confusion matrix trained based on cost (trial).

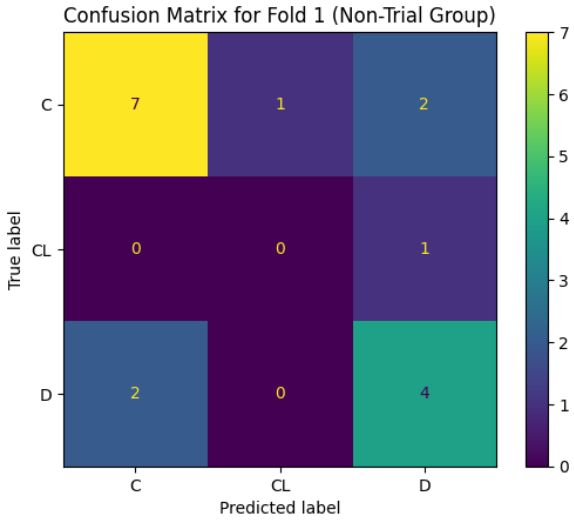


Figure 218. Neural network confusion matrix trained based on cost (non-trial).

```
Results for Non-Trial Group:
Mean accuracy: 0.6617647058823529
Standard deviation of accuracy: 0.08987949288386718
Mean cost: 43.333333333333336
Standard deviation of cost: 12.931443160847214
Mean cost per instance: 2.7083333333333335
Standard deviation of cost per instance: 0.8082151975529509
```

Figure 219. Results for the neural network confusion matrix trained based on cost (non-trial).

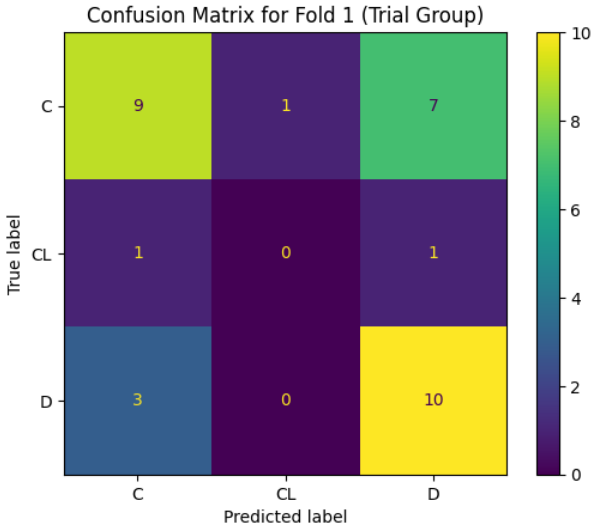


Figure 220. 2-layer neural network confusion matrix trained based on cost (trial).

```
Results for Trial Group:
Mean accuracy: 0.7181451612903226
Standard deviation of accuracy: 0.08842995671749043
Mean cost: 69.3
Standard deviation of cost: 19.401288616996553
Mean cost per instance: 2.235483870967742
Standard deviation of cost per instance: 0.6258480199031147
```

Figure 221. Results for the 2-layer neural network confusion matrix trained based on cost (trial).

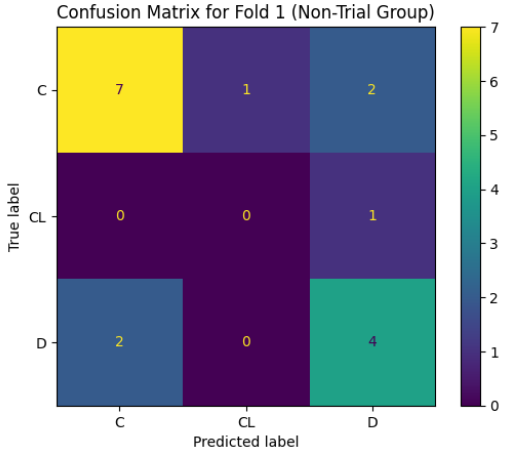


Figure 222. 2-layer neural network confusion matrix trained based on cost (non-trial).

```
Results for Non-Trial Group:
Mean accuracy: 0.6501225490196078
Standard deviation of accuracy: 0.11300621138253514
Mean cost: 45.333333333333336
Standard deviation of cost: 16.659998666133067
Mean cost per instance: 2.8333333333333335
Standard deviation of cost per instance: 1.0412499166333167
```

Figure 223. Results for the 2-layer neural network confusion matrix trained based on cost (non-trial).

Conclusions

The following table includes the results from the trained models, displaying the accuracy, cost, and confidence intervals for each model.

Model	Accuracy	Cost per instance
Decision tree (non-trial)	55.75% ± 15.66%	3.16 ± 1.15
Decision tree (trial)	70.50% ± 6.44%	2.33 ± 0.59
Naive Bayes (non-trial)	65.99% ± 4.37%	2.80± 0.58
Naive Bayes (trial)	70.82 % ± 7.06%	2.58 ± 0.53
Neural network (non-trial)	66.17 % ± 8.98%	2.70 ± 0.80
Neural network (trial)	69.24% ± 7.67%	2.42 ± 0.53
2-layer neural network (non-trial)	65.01% ± 11.30%	2.83 ± 1.04
2-layer neural network (trial)	71.81 % ± 8.84%	2.23 ± 0.62

Table 7. Accuracy and cost per instance performance of different models trained based on cost in Python.

Cost-trained models generally perform better in terms of cost per instance by minimizing misclassifications that could have a significant impact on patient health. However, they tend to sacrifice some accuracy in comparison to accuracy-trained models. In the trial group, cost-trained models show competitive accuracy while maintaining efficient costs, particularly with decision trees and 2-layer neural networks. These models are more cost-efficient, as evidenced by the lower cost per instance, especially for decision trees (2.33 ± 0.59) and 2-layer neural networks (2.23 ± 0.62). On the other hand, accuracy-trained models show better overall accuracy, but this comes at a higher cost. For example, the decision tree (accuracy-trained) has

a higher cost per instance (2.870 ± 1.127), indicating that while the accuracy is better, the misclassification of more expensive classes like death or transplant increases the overall cost.

In the non-trial group, models show lower performance and higher costs when compared to the trial group. For instance, the decision tree has the lowest accuracy (55.75% ± 15.66%) and the highest cost per instance (3.16 ± 1.15), indicating that misclassifications are more frequent. The neural network and 2-layer neural network show similar trends, with performance lagging behind the trial group models and higher costs, particularly in the 2-layer neural network with a cost of 2.83 ± 1.04.

In summary, while cost-trained models may show a slight reduction in accuracy, they are more efficient in reducing costly errors, which is crucial in medical contexts. Accuracy-trained models, although providing higher accuracy, are less cost-efficient and more prone to misclassifications that could have significant consequences in practical scenarios.

4.2.2 MERGED GROUPS: COMBINING TRIAL AND NON-TRIAL DATA

A. Accuracy-trained performance

Decision tree

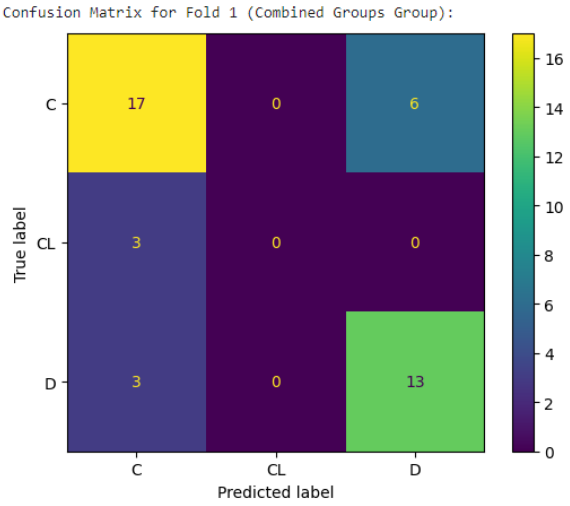


Figure 224. Decision tree confusion matrix trained based on accuracy (combined).

```
Results for Combined Groups Group:
Mean accuracy: 0.6891986062717769
Standard deviation of accuracy: 0.0398128414134793
Mean cost: 105.3
Standard deviation of cost: 18.61209284309532
Mean cost per instance: 2.5682926829268293
Standard deviation of cost per instance: 0.4539534839779346
```

Figure 225. Results for the decision tree confusion matrix trained based on accuracy (combined).

Naive Bayes

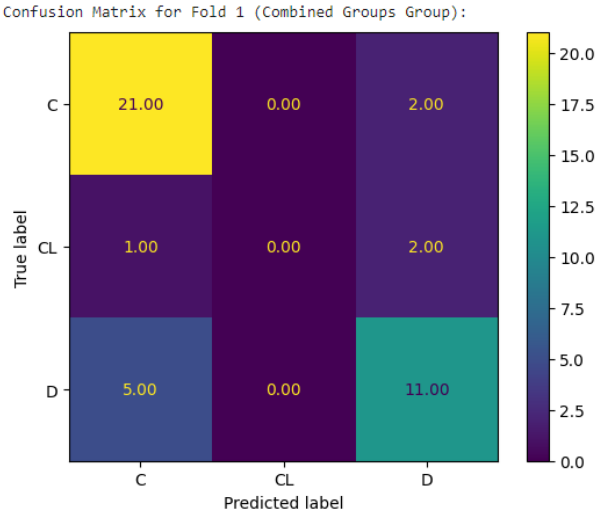


Figure 226. Naive Bayes confusion matrix trained based on accuracy (combined).

```
Results for Combined Groups Group:
Mean accuracy: 0.7085946573751452
Standard deviation of accuracy: 0.03704047443471461
Mean cost: 101.4
Standard deviation of cost: 12.666491226855209
Mean cost per instance: 2.473170731707317
Standard deviation of cost per instance: 0.3089388104111027
```

Figure 227. Results for the Naive Bayes confusion matrix trained based on accuracy (combined).

Neural Networks

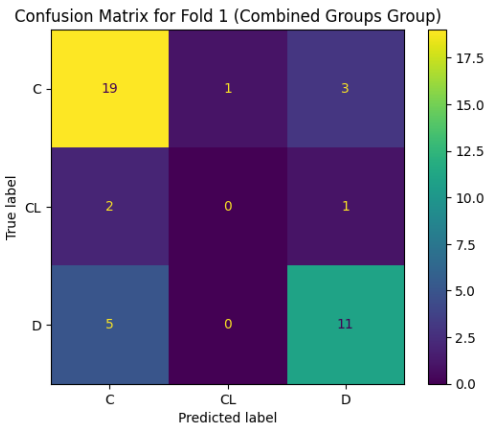


Figure 228. Neural network confusion matrix trained based on accuracy (combined).

```
Results for Combined Groups Group:
Mean accuracy: 0.6747967479674796
Standard deviation of accuracy: 0.045890851900730265
Mean cost: 108.1
Standard deviation of cost: 17.172361514946044
Mean cost per instance: 2.636585365853659
Standard deviation of cost per instance: 0.4188380857303914
```

Figure 229. Results for the neural network confusion matrix trained based on accuracy (combined).

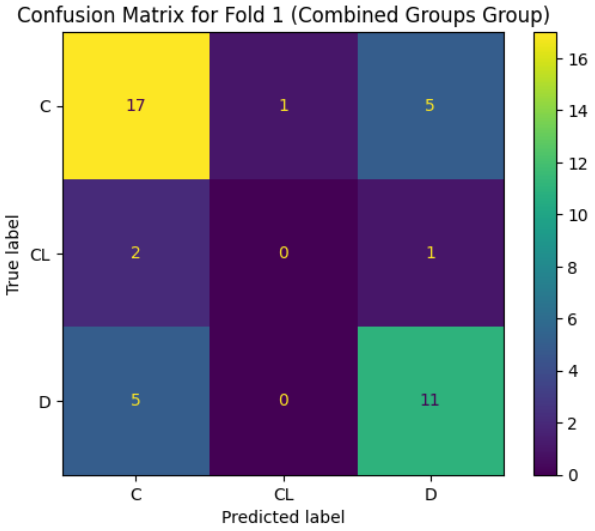


Figure 230. 2-Layer neural network confusion matrix trained based on accuracy (combined).

```
Results for Combined Groups Group:
Mean accuracy: 0.6748548199767711
Standard deviation of accuracy: 0.025636605265672904
Mean cost: 108.3
Standard deviation of cost: 10.178899744078434
Mean cost per instance: 2.6414634146341465
Standard deviation of cost per instance: 0.2482658474165472
```

Figure 231. Results for the 2-layer neural network confusion matrix trained based on accuracy (combined).

Deep Learning

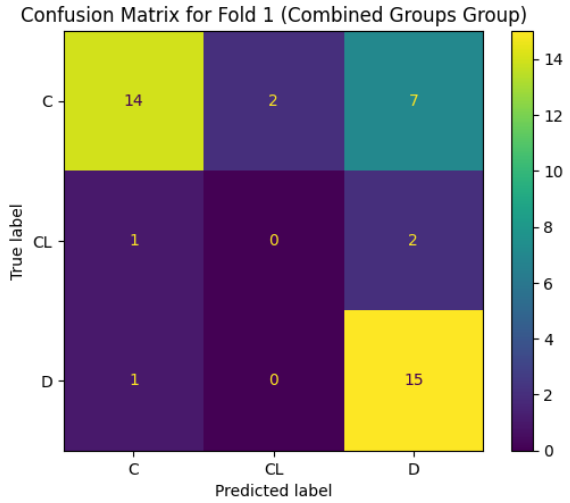


Figure 232. Deep learning confusion matrix trained based on accuracy (combined).

```
Results for Combined Groups Group:
Mean accuracy: 0.7012775842044134
Standard deviation of accuracy: 0.04167466850447221
Mean cost: 89.1
Standard deviation of cost: 17.958006570886425
Mean cost per instance: 2.173170731707317
Standard deviation of cost per instance: 0.4380001602655226
```

Figure 233. Results for the deep learning confusion matrix trained based on accuracy (combined).

Conclusions

Model	Accuracy	Cost per instance
Decision tree	68.91% ± 3.98%	2.56 ± 0.45
Naive Bayes	70.85% ± 3.70%	2.47± 0.30
Neural network	67.47 % ± 4.58%	2.63± 0.41
2-layer neural network	67.48 % ± 2.56%	2.64 ± 0.24
Deep learning	70.12% ± 4.16%	2.17% ± 0.43

Table 8. Accuracy and cost per instance performance of different models (combined data) trained based on accuracy in Python.

When comparing the separated models (using trial and non-trial data separately) with the combined models, several key observations can be made. In terms of accuracy, trial group models, such as decision tree (73.33%) and naive bayes (71.47%), outperform the combined models, which have accuracy values of decision tree (68.91%) and naive bayes (70.85%). Neural networks show similar performance in both separated and combined groups, with neural network (non-trial) achieving 69.05% accuracy and neural network in the combined group reaching 67.47%. The 2-layer neural network (trial) also performs better with 69.05% accuracy compared to the combined model's performance of 67.48%. However, deep learning (combined) shows a marked improvement over non-trial deep learning (48.89%), though it still lags trial-based models in terms of accuracy.

Regarding cost per instance, the combined models generally show a slight improvement in cost efficiency. Deep learning (combined) leads as the most cost-efficient model with a cost of 2.17. Naive bayes (combined) has a cost of 2.47, which is lower than naive bayes (trial) at 2.68. Neural network (combined) has a cost of 2.63, which is slightly higher than neural network (trial) at 2.53, showing some improvement in cost efficiency with the combined data in certain

cases. Similarly, 2-layer neural network (combined) has a cost of 2.64, which is marginally higher than 2.50 for non-trial, but still lower than 2.43 for 2-layer neural network (trial). Both decision tree and naive bayes in the combined group show a balance in terms of accuracy and cost efficiency.

In terms of confidence intervals, the combined models exhibit narrower confidence intervals for accuracy, suggesting more stability in their predictions compared to non-trial models. For example, naive bayes (combined) has a narrower interval of ±3.70%, compared to naive bayes (non-trial), which has ±7.707%. Similarly, decision tree (combined) has an interval of ±3.98%, compared to decision tree (non-trial) with ±7.512%. Neural networks also demonstrate a trend where the combined models show narrower intervals, indicating more consistent results.

In conclusion, while trial group models generally outperform the combined models in terms of accuracy, the combined models tend to offer better cost efficiency, providing a better balance between accuracy and cost. Moreover, the combined models exhibit more stable and consistent performance, as indicated by their narrower confidence intervals, especially when compared to non-trial models.

B. Cost-trained performance

Decision tree

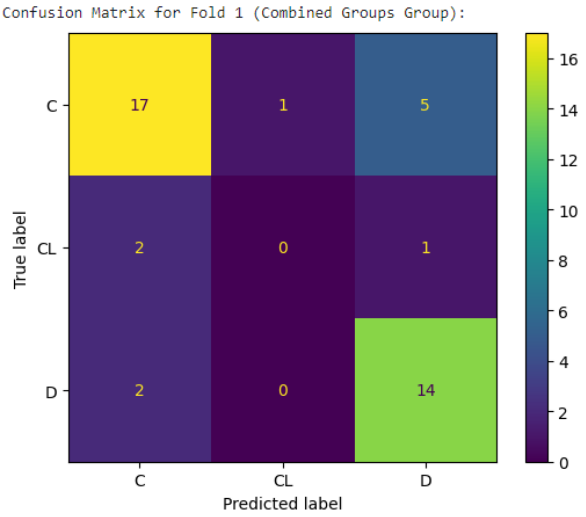


Figure 234. Decision tree confusion matrix trained based on cost (combined).

```
Results for Combined Groups Group:
Mean accuracy: 0.7182346109175377
Standard deviation of accuracy: 0.054494818075674405
Mean cost: 87.3
Standard deviation of cost: 19.20963300013824
Mean cost per instance: 2.129268292682927
Standard deviation of cost per instance: 0.4685276341497131
```

Figure 235. Results for the decision tree confusion matrix trained based on cost (combined).

Naive Bayes

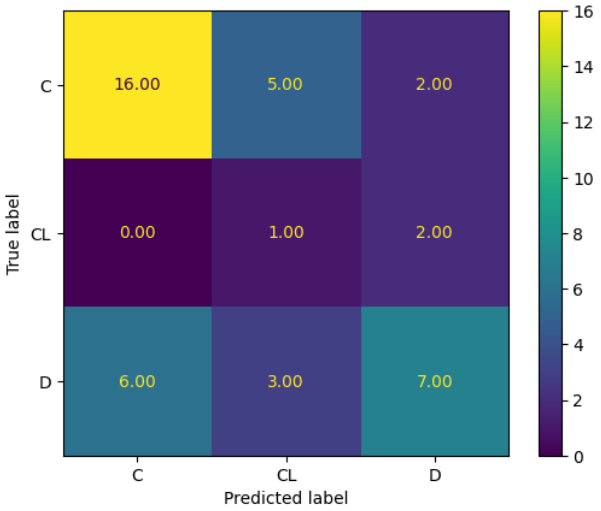


Figure 236. Naive Bayes confusion matrix trained based on cost (combined).

```
--------------------------------------------------
Results for Combined Groups Group:
Mean accuracy: 0.6360046457607432
Standard deviation of accuracy: 0.04519241221597506
Mean cost: 137.8
Standard deviation of cost: 19.4411933790084
Mean cost per instance: 3.3609756097560974
Standard deviation of cost per instance: 0.4741754482684976
```

Figure 237. Results for the Naive Bayes confusion matrix trained based on cost (combined).

Neural Networks

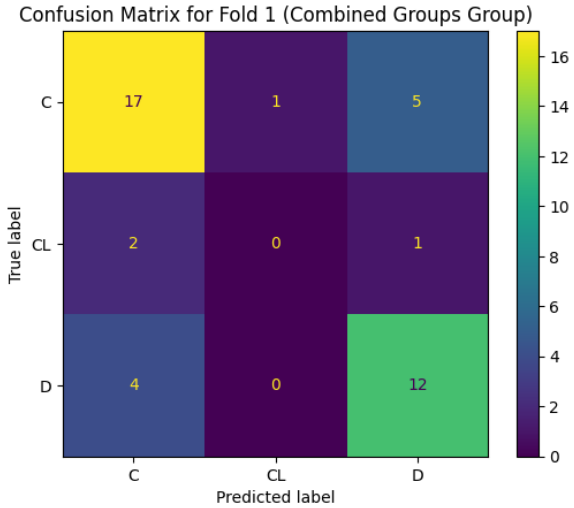


Figure 238. Neural network confusion matrix trained based on cost (combined).

```
Results for Combined Groups Group:
Mean accuracy: 0.6821138211382113
Standard deviation of accuracy: 0.044830086691611246
Mean cost: 103.3
Standard deviation of cost: 11.75627492022877
Mean cost per instance: 2.519512195121951
Standard deviation of cost per instance: 0.2867384126885066
```

Figure 239. Results for the neural network confusion matrix trained based on cost (combined).

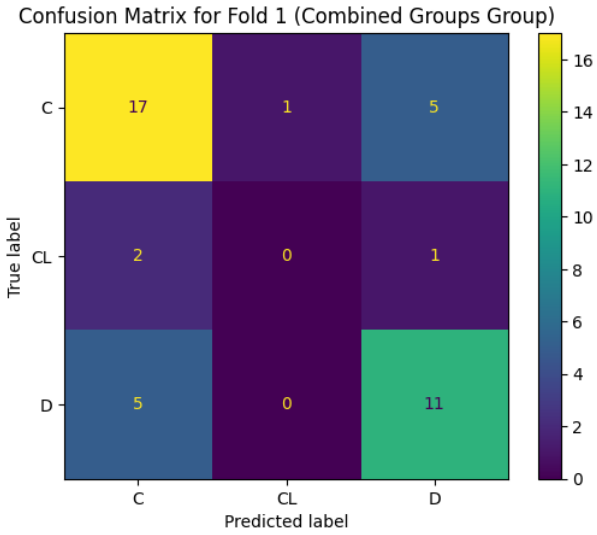


Figure 240. 2-layer neural network confusion matrix trained based on cost (combined).

```
Results for Combined Groups Group:
Mean accuracy: 0.69198606271777
Standard deviation of accuracy: 0.04289096550538194
Mean cost: 99.5
Standard deviation of cost: 15.717824276915684
Mean cost per instance: 2.426829268292683
Standard deviation of cost per instance: 0.3833615677296508
```

Figure 241. Results for the 2-layer neural network confusion matrix trained based on cost (combined).

Conclusions

Model	Accuracy	Cost per instance
Decision tree	71.82% ± 5.44%	2.12 ± 0.46
Naive Bayes	63.6% ± 4.5%	3.36± 0.47
Neural network	68.21 % ± 4.48%	2.51± 0.28
2-layer neural network	69.19% ± 4.28%	2.42 ± 0.38

Table 9. Accuracy and cost per instance performance of different models (combined data) trained based on cost in Python.

For cost-trained models, the trial group models generally outperform the combined models in terms of accuracy. However, the combined models still provide a competitive level of accuracy, with decision tree leading the performance. In addition, the combined models show a more cost-efficient approach, particularly the decision tree, which is the most cost-efficient model overall. However, some models such as Naive have higher costs, which affects the cost-benefit ratio.

Comparison of Altair AI Studio and Python Models

Overall, the results from Altair AI Studio and Python, although slightly different in certain models, do not show significant discrepancies. While Python models tend to provide higher accuracy, especially with trial-based data, Altair AI Studio models offer a better balance in terms of cost-efficiency, consistency and generalization. This makes the combined models from Altair AI Studio more suitable for scenarios where stable performance across different data sources and cost-efficient predictions are prioritized.

Combination of Models

Apparently, but it cannot be confirmed until a statistical test is conducted, cost-trained models tend to be better for predicting cirrhosis in medical contexts, as they help minimize costly errors that could have a significant impact on patient health. Although these models may exhibit a slight reduction in overall accuracy, they are better suited to maximize efficiency by minimizing the costliest errors. Combined models (trial and non-trial) generally outperform models trained separately. By combining both datasets, greater generalization and robustness are achieved, resulting in models with better predictive performance and lower cost in terms of misclassification. This demonstrates that using both types of data (clinical trial data and hospital data) helps improve the models' ability to handle more diverse and real-world scenarios.

To further enhance the performance of these cost-trained models, various model combination approaches will be applied for the combined dataset, including Bagging, AdaBoost, Stacking, and Voting. These ensemble methods combine the strengths of different models to create a more robust system that can handle the inherent variability in the data and reduce overfitting. Due to time constraints, these techniques will not be tested with the trial and non-trial groups independently, nor with accuracy-trained models. Instead, the focus will be on leveraging the models trained with cost-based training and combining the results from the merged groups.

Here are the results obtained for the cost-trained models with AdaBoost and Bagging.

Decision Tree

accuracy: 70.41% +/- 4.19% (micro average: 70.39%)

	true D	true C	true CL	class precision
pred. D	122	62	14	61.62%
pred. C	35	167	10	78.77%
pred. CL	0	1	1	50.00%
class recall	77.71%	72.61%	4.00%	

Figure 242. Decision tree cost-trained confusion matrix (AdaBoost).

accuracy: 72.81% +/- 5.13% (micro average: 72.82%)

	true D	true C	true CL	class precision
pred. D	103	33	9	71.03%
pred. C	54	197	16	73.78%
pred. CL	0	0	0	0.00%
class recall	65.61%	85.65%	0.00%	

Figure 243. Decision tree cost-trained confusion matrix (Bagging).

Rule-based Model

accuracy: 70.43% +/- 7.42% (micro average: 70.39%)

	true D	true C	true CL	class precision
pred. D	118	58	14	62.11%
pred. C	39	171	10	77.73%
pred. CL	0	1	1	50.00%
class recall	75.16%	74.35%	4.00%	

Figure 244. Rule-based model cost-trained confusion matrix (AdaBoost).

accuracy: 70.13% +/- 5.97% (micro average: 70.15%)

	true D	true C	true CL	class precision
pred. D	113	55	14	62.09%
pred. C	44	175	10	76.42%
pred. CL	0	0	1	100.00%
class recall	71.97%	76.09%	4.00%	

Figure 245. Rule-based model cost-trained confusion matrix (Bagging).

Naive Bayes

accuracy: 71.34% +/- 6.60% (micro average: 71.36%)

	true D	true C	true CL	class precision
pred. D	111	48	10	65.68%
pred. C	43	180	12	76.60%
pred. CL	3	2	3	37.50%
class recall	70.70%	78.26%	12.00%	

Figure 246. Naive Bayes model cost-trained confusion matrix (AdaBoost).

accuracy: 71.85% +/- 6.96% (micro average: 71.84%)

	true D	true C	true CL	class precision
pred. D	100	31	8	71.94%
pred. C	54	194	15	73.76%
pred. CL	3	5	2	20.00%
class recall	63.69%	84.35%	8.00%	

Figure 247. Naive Bayes cost-trained confusion matrix (Bagging).

Bayesian Network

accuracy: 69.41% +/- 8.80% (micro average: 69.42%)

	true D	true C	true CL	class precision
pred. D	116	59	10	62.70%
pred. C	41	169	14	75.45%
pred. CL	0	2	1	33.33%
class recall	73.89%	73.48%	4.00%	

Figure 248. Bayesian network cost-trained confusion matrix (AdaBoost).

accuracy: 73.80% +/- 4.17% (micro average: 73.79%)

	true D	true C	true CL	class precision
pred. D	106	33	9	71.62%
pred. C	50	197	15	75.19%
pred. CL	1	0	1	50.00%
class recall	67.52%	85.65%	4.00%	

Figure 249. Bayesian network cost-trained confusion matrix (Bagging).

2- Layer Neural Network

accuracy: 70.15% +/- 6.08% (micro average: 70.15%)

	true D	true C	true CL	class precision
pred. D	120	61	13	61.86%
pred. C	37	169	12	77.52%
pred. CL	0	0	0	0.00%
class recall	76.43%	73.48%	0.00%	

Figure 250. 2-layer neural network cost-trained confusion matrix (AdaBoost).

accuracy: 73.29% +/- 4.64% (micro average: 73.30%)

	true D	true C	true CL	class precision
pred. D	106	34	9	71.14%
pred. C	49	196	16	75.10%
pred. CL	2	0	0	0.00%
class recall	67.52%	85.22%	0.00%	

Figure 251. 2-layer neural network cost-trained confusion matrix (Bagging).

Deep Learning

accuracy: 66.25% +/- 6.20% (micro average: 66.26%)

	true D	true C	true CL	class precision
pred. D	121	75	13	57.89%
pred. C	34	149	9	77.60%
pred. CL	2	6	3	27.27%
class recall	77.07%	64.78%	12.00%	

Figure 252. Deep learning cost-trained confusion matrix (AdaBoost).

accuracy: 71.61% +/- 7.46% (micro average: 71.60%)

	true D	true C	true CL	class precision
pred. D	110	41	11	67.90%
pred. C	45	183	12	76.25%
pred. CL	2	6	2	20.00%
class recall	70.06%	79.57%	8.00%	

Figure 253. Deep learning cost-trained confusion matrix (Bagging).

Model	Accuracy	Cost
Decision tree (AdaBoost)	70.41% ± 4.19%	2.146 ± 0.319
Decision tree (Bagging)	72.81% ± 5.13%	2.241 ± 0.5
Rule-based (AdaBoost)	70.43% ± 7.42%	2.194 ± 0.467
Rule-based (Bagging)	70.13% ± 5.97%	2.269 ± 0.477
Naive Bayes (AdaBoost)	71.34% ± 6.60%	2.236 ± 0.562
Naive Bayes (Bagging)	71.85% ± 6.94%	2.342 ± 0.608
Bayesian Network (AdaBoost)	69.41% ± 8.80%	2.158±0.391
Bayesian Network (Bagging)	73.80% ± 4.17%	2.158±0.391
2-Layer Neural network (AdaBoost)	70.15% ± 6.08%	2.186 ± 0.437
2-Layer Neural network (Bagging)	73.29% ± 4.64%	2.203 ± 0.429
Deep Learning (AdaBoost)	66.25% ± 6.20%	2.372 ± 0.416
Deep Learning (Bagging)	71.61% ± 7.46%	2.234 ± 0.563

Table 10. Accuracy and cost performance of different models trained based on cost.

In general, Bagging improved accuracy for most cost-trained models (e.g., decision tree, Bayesian network, 2-layer neural network, and deep learning) compared to the models without Bagging. However, the trade-off between accuracy and cost exists, as some models saw a slight increase in cost with improved accuracy. The Bayesian network and deep learning models showed notable improvements with Bagging. On the other hand, AdaBoost did not result in significant improvements for any model when compared to their performance with Bagging or without these ensemble methods.

For both Vote and Stacking, the cost-trained models of the merged groups were used (deep learning and 2-layer neural network models include Bagging). The accuracies achieved by the combined models using Vote and Stacking were not the highest, nor were the costs the

lowest. However, it is worth noting that the confidence interval for the Vote model is relatively narrow, indicating that the model's performance is consistent and stable across the different folds. This suggests that while the overall performance may not be the best in terms of accuracy or cost, the model's reliability is higher.

accuracy: 71.86% +/- 4.04% (micro average: 71.84%)

	true D	true C	true CL	class precision
pred. D	115	49	12	65.34%
pred. C	42	181	13	76.69%
pred. CL	0	0	0	0.00%
class recall	73.25%	78.70%	0.00%	

```
Misclassificationcosts: 2.156 +/- 0.306 (micro average: 2.158)
```

Figure 254. Combined models trained based on cost (Vote).

accuracy: 68.48% +/- 7.74% (micro average: 68.45%)

	true D	true C	true CL	class precision
pred. D	117	63	12	60.94%
pred. C	39	163	11	76.53%
pred. CL	1	4	2	28.57%
class recall	74.52%	70.87%	8.00%	

```
Misclassificationcosts: 2.298 +/- 0.572 (micro average: 2.301)
```

Figure 255. Combined models trained based on cost (Stacking).

Statistical Analysis and Final Conclusions

A t-test was performed to statistically analyse and compare the performance of the models across different training setups. The p-value is the key metric in this analysis. A p-value below 0.05 indicates that the model's performance is statistically significantly different from the other model it is being compared to. In other words, if the p-value is less than 0.05, we can reject the null hypothesis (which assumes no difference between the models) and conclude that the performance differences observed are unlikely to be due to chance. Conversely, if the p-value is greater than 0.05, it suggests that the models do not show a statistically significant difference in performance, meaning the variations in their outcomes could be attributed to random chance rather than a true underlying difference between them.

In this section the models will be referred to as numbers. The code to know which number corresponds to each model is the following one (accuracy-trained models):

1. Decision tree trained based on accuracy (non-trial).
2. Rule-based model trained based on accuracy (non-trial).
3. Naive Bayes trained based on accuracy (non-trial).

4. Bayesian network trained based on accuracy (non-trial).
5. Deep learning trained based on accuracy (non-trial).
6. Neural network trained based on accuracy (non-trial).
7. 2-layer neural network trained based on accuracy (non-trial).
8. Vote operator trained based on accuracy (non-trial).
9. Stacking operator trained based on accuracy (non-trial).
10. Decision tree trained based on accuracy (trial).
11. Rule-based model trained based on accuracy (trial).
12. Naive Bayes trained based on accuracy (trial).
13. Bayesian network trained based on accuracy (trial).
14. Deep learning trained based on accuracy (trial).
15. Neural network trained based on accuracy (trial).
16. 2-layer neural network trained based on accuracy (trial).
17. Decision tree trained based on accuracy (combined).
18. Rule-based model trained based on accuracy (combined).
19. Naive Bayes trained based on accuracy (combined).
20. Bayesian network trained based on accuracy (combined).
21. Deep learning trained based on accuracy (combined).
22. Neural network trained based on accuracy (combined).
23. 2-layer neural network trained based on accuracy (combined).

The code to know which number corresponds to each model is the following one (cost-trained models):

1. Decision tree trained based on cost (non-trial).
2. Rule-based model trained based on cost (non-trial).
3. Naive Bayes trained based on cost (non-trial).
4. Bayesian network trained based on cost (non-trial).
5. Deep learning trained based on cost (non-trial).
6. Neural network trained based on cost (non-trial).
7. 2-layer neural network trained based on cost (non-trial).
8. Decision tree trained based on cost (trial).
9. Rule-based model trained based on cost (trial).
10. Naive Bayes trained based on cost (trial).
11. Bayesian network trained based on cost (trial).
12. Deep learning trained based on cost (trial).
13. Neural network trained based on cost (trial).
14. 2-layer neural network trained based on cost (trial).

15. Decision tree trained based on cost (combined).
16. Rule-based model trained based on cost (combined).
17. Naive Bayes trained based on cost (combined).
18. Bayesian network trained based on cost (combined).
19. Deep learning trained based on cost (combined).
20. Neural network trained based on cost (combined).
21. 2-layer neural network trained based on cost (combined).
22. Vote operator trained based on cost (combined).
23. Stacking operator trained based on cost (combined).

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X
	0.580 +/-...	0.720 +/-...	0.670 +/-...	0.660 +/-...	0.690 +/-...	0.700 +/-...	0.580 +/-...	0.610 +/-...	0.680 +/-...	0.680 +/-...	0.664 +/-...	0.741 +/-...	0.741 +/-...	0.653 +/-...	0.715 +/-...	0.730 +/-...	0.701 +/-...	0.677 +/-...	0.716 +/-...	0.723 +/-...	0.702 +/-...	0.726 +/-...	0.713 +/-...
0.580 +/-...		0.064	0.147	0.234	0.159	0.058	1.000	0.600	0.137	0.129	0.158	0.014	0.013	0.219	0.027	0.023	0.049	0.131	0.025	0.022	0.058	0.016	0.034
0.720 +/-...	0.064		0.362	0.328	0.675	0.712	0.016	0.041	0.506	0.490	0.282	0.705	0.697	0.213	0.915	0.853	0.720	0.450	0.940	0.950	0.738	0.908	0.901
0.670 +/-...	0.147	0.362		0.828	0.737	0.424	0.022	0.074	0.824	0.819	0.852	0.070	0.059	0.632	0.169	0.135	0.366	0.862	0.148	0.127	0.418	0.072	0.228
0.660 +/-...	0.234	0.328	0.828		0.647	0.388	0.090	0.241	0.703	0.695	0.929	0.092	0.083	0.880	0.198	0.150	0.348	0.726	0.181	0.154	0.382	0.113	0.237
0.690 +/-...	0.159	0.675	0.737	0.647		0.866	0.075	0.167	0.877	0.870	0.647	0.400	0.391	0.529	0.662	0.509	0.845	0.834	0.639	0.564	0.848	0.519	0.688
0.700 +/-...	0.058	0.712	0.424	0.388	0.866		0.004	0.010	0.656	0.634	0.280	0.261	0.259	0.185	0.643	0.441	0.970	0.571	0.600	0.492	0.967	0.387	0.703
0.580 +/-...	1.000	0.016	0.022	0.090	0.075	0.004		0.342	0.034	0.027	0.017	0.000	0.000	0.040	0.000	0.001	0.002	0.024	0.000	0.000	0.004	0.000	0.001
0.610 +/-...	0.600	0.041	0.074	0.241	0.167	0.010	0.342		0.098	0.079	0.059	0.001	0.000	0.142	0.001	0.002	0.004	0.073	0.000	0.001	0.013	0.000	0.002
0.680 +/-...	0.137	0.506	0.824	0.703	0.877	0.656	0.034	0.098		0.996	0.697	0.188	0.174	0.535	0.396	0.287	0.617	0.951	0.369	0.311	0.640	0.248	0.444
0.680 +/-...	0.129	0.490	0.819	0.695	0.870	0.634	0.027	0.079	0.996		0.683	0.163	0.149	0.514	0.360	0.261	0.590	0.953	0.332	0.278	0.618	0.212	0.413
0.664 +/-...	0.158	0.282	0.852	0.929	0.647	0.280	0.017	0.059	0.697	0.683		0.030	0.022	0.728	0.068	0.071	0.214	0.718	0.054	0.052	0.282	0.018	0.119
0.741 +/-...	0.014	0.705	0.070	0.092	0.400	0.261	0.000	0.001	0.188	0.163	0.030		0.998	0.019	0.414	0.792	0.257	0.128	0.435	0.607	0.317	0.614	0.443
0.741 +/-...	0.013	0.697	0.059	0.083	0.391	0.259	0.000	0.000	0.174	0.149	0.022	0.998		0.014	0.382	0.781	0.230	0.114	0.402	0.584	0.296	0.584	0.418
0.653 +/-...	0.219	0.213	0.632	0.880	0.529	0.185	0.040	0.142	0.535	0.514	0.728	0.019	0.014		0.041	0.046	0.135	0.535	0.033	0.032	0.190	0.011	0.075
0.715 +/-...	0.027	0.915	0.169	0.198	0.662	0.643	0.000	0.001	0.396	0.360	0.068	0.414	0.382	0.041		0.643	0.632	0.295	0.945	0.752	0.694	0.615	0.967
0.730 +/-...	0.023	0.853	0.135	0.150	0.509	0.441	0.001	0.002	0.287	0.261	0.071	0.792	0.781	0.046	0.643		0.426	0.216	0.674	0.844	0.479	0.885	0.651
0.701 +/-...	0.049	0.720	0.366	0.348	0.845	0.970	0.002	0.004	0.617	0.590	0.214	0.257	0.230	0.135	0.632	0.426		0.522	0.583	0.470	0.993	0.344	0.704
0.677 +/-...	0.131	0.450	0.862	0.726	0.834	0.571	0.024	0.073	0.951	0.953	0.718	0.128	0.114	0.535	0.295	0.216	0.522		0.268	0.224	0.558	0.158	0.352
0.716 +/-...	0.025	0.940	0.148	0.181	0.639	0.600	0.000	0.001	0.369	0.332	0.054	0.435	0.402	0.033	0.945	0.674	0.583	0.268		0.793	0.652	0.656	0.921
0.723 +/-...	0.022	0.950	0.127	0.154	0.564	0.492	0.000	0.001	0.311	0.278	0.052	0.607	0.584	0.032	0.752	0.844	0.470	0.224	0.793		0.539	0.922	0.755
0.702 +/-...	0.058	0.738	0.418	0.382	0.848	0.967	0.004	0.013	0.640	0.618	0.282	0.317	0.296	0.190	0.694	0.479	0.993	0.558	0.652	0.539		0.443	0.746
0.726 +/-...	0.016	0.908	0.072	0.113	0.519	0.387	0.000	0.000	0.248	0.212	0.018	0.614	0.584	0.011	0.615	0.885	0.344	0.158	0.656	0.922	0.443		0.650
0.713 +/-...	0.034	0.901	0.228	0.237	0.688	0.703	0.001	0.002	0.444	0.413	0.119	0.443	0.418	0.075	0.967	0.651	0.704	0.352	0.921	0.755	0.746	0.650	

Figure 256. T-test table on accuracy performance of accuracy-trained models.

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X
	0.640 +/-...	0.640 +/-...	0.680 +/-...	0.700 +/-...	0.660 +/-0...	0.670 +/-...	0.660 +/-...	0.692 +/-...	0.702 +/-0...	0.744 +/-...	0.734 +/-0...	0.728 +/-...	0.750 +/-0...	0.760 +/-...	0.687 +/-...	0.701 +/-...	0.726 +/-...	0.723 +/-...	0.704 +/-...	0.716 +/-...	0.701 +/-...	0.719 +/-...	0.680 +/-...
0.640 +/-...		1.000	0.546	0.414	0.822	0.652	0.770	0.383	0.296	0.073	0.113	0.125	0.069	0.059	0.401	0.284	0.136	0.159	0.249	0.201	0.309	0.173	0.480
0.640 +/-...			0.497	0.370	0.810	0.613	0.744	0.311	0.225	0.035	0.065	0.070	0.035	0.031	0.319	0.207	0.080	0.100	0.170	0.137	0.240	0.110	0.403
0.680 +/-...				0.754	0.806	0.859	0.733	0.798	0.638	0.147	0.240	0.271	0.138	0.115	0.870	0.628	0.298	0.347	0.567	0.441	0.657	0.384	0.996
0.700 +/-...					0.647	0.641	0.548	0.890	0.970	0.411	0.537	0.599	0.371	0.308	0.807	0.979	0.631	0.675	0.939	0.777	0.981	0.730	0.706
0.660 +/-...						0.903	1.000	0.673	0.579	0.260	0.328	0.359	0.239	0.204	0.712	0.577	0.376	0.403	0.546	0.461	0.589	0.431	0.788
0.670 +/-...							0.866	0.646	0.501	0.101	0.173	0.193	0.097	0.082	0.696	0.485	0.215	0.256	0.427	0.332	0.521	0.283	0.825
0.660 +/-...								0.529	0.406	0.083	0.139	0.155	0.079	0.057	0.563	0.389	0.172	0.205	0.338	0.267	0.423	0.226	0.675
0.692 +/-...									0.786	0.109	0.232	0.261	0.112	0.097	0.872	0.779	0.301	0.376	0.691	0.509	0.808	0.423	0.703
0.702 +/-...										0.177	0.347	0.399	0.173	0.145	0.623	0.984	0.450	0.536	0.949	0.695	0.985	0.605	0.480
0.744 +/-...											0.717	0.493	0.826	0.623	0.025	0.111	0.456	0.455	0.070	0.334	0.193	0.322	0.017
0.734 +/-...												0.824	0.614	0.474	0.102	0.278	0.770	0.727	0.249	0.573	0.358	0.598	0.071
0.728 +/-...													0.439	0.341	0.098	0.313	0.929	0.863	0.268	0.675	0.412	0.715	0.065
0.750 +/-...														0.783	0.039	0.123	0.409	0.405	0.096	0.307	0.185	0.302	0.027
0.760 +/-...															0.041	0.109	0.320	0.318	0.092	0.246	0.154	0.243	0.030
0.687 +/-...																0.584	0.129	0.208	0.440	0.331	0.657	0.229	0.774
0.701 +/-...																	0.369	0.471	0.917	0.644	0.999	0.537	0.430
0.726 +/-...																		0.927	0.335	0.737	0.460	0.786	0.087
0.723 +/-...																			0.464	0.821	0.541	0.879	0.147
0.704 +/-...																				0.665	0.933	0.533	0.299
0.716 +/-...																					0.692	0.925	0.241
0.701 +/-...																						0.608	0.516
0.719 +/-...																							0.158
0.680 +/-...																							

Figure 257. T-test table on accuracy performance of cost-trained models.

In terms of both accuracy and cost-efficiency, the 2-layer neural network (trial) consistently ranks among the top models. Additionally, decision tree (trial) and Naive Bayes (trial) also perform exceptionally well, showing significant improvements over their non-trial and combined counterparts. In contrast, deep learning (non-trial) consistently demonstrates the weakest performance, with statistically significant differences when compared to most trial-based and combined models.

The t-test results show the following p-values for accuracy:

- The p-values for trial models, particularly those trained on accuracy, are statistically significant ($p < 0.05$), indicating a significant improvement in performance and consistency over non-trial models.
- Combined models generally show promising generalization across diverse datasets, achieving cost efficiencies while maintaining competitive accuracy. The p-values for the combined models indicate that they are not as statistically different from trial models in terms of cost but may show some variance in accuracy (with p-values greater than 0.05 in some comparisons).

In conclusion, models trained on the trial group remain the best choice when accuracy is prioritized, especially for clinical applications. However, combined models strike a valuable balance between cost and accuracy, making them well-suited for real-world applications that require both generalization and efficiency. The p-values support these findings, with most trial models showing significant performance improvements over non-trial models.

References

[1] Mayo Clinic. (n.d.). *Cirrhosis - Symptoms and causes*. Mayo Clinic. Retrieved December 1, 2024, from https://www.mayoclinic.org/diseases-conditions/cirrhosis/symptoms-causes/syc-20351487

[2] Dickson, E., Grambsch, P., Fleming, T., Fisher, L., & Langworthy, A. (1989). Cirrhosis Patient Survival Prediction [Dataset]. UCI Machine Learning Repository. https://doi.org/10.24432/C5R02G.