Liver Cirrhosis Stage Prediction

Authors: Miloš Čuturić (SV11/2020), Luka Đorđević (SV14/2020), Marko Janošević (SV46/2020)

1. Motivation

Liver cirrhosis is a serious chronic disease that leads to permanent liver damage. Timely diagnosis and prediction of the stage of cirrhosis can significantly improve treatment outcomes and the quality of life for patients. Our goal is to utilise machine learning to develop a model that can accurately predict the stage of liver cirrhosis based on clinical and laboratory data.

2. Research questions

The specific problem we aim to address is the prediction of liver cirrhosis stages (1, 2, or 3) based on available medical data. We are using a dataset from the Kaggle platform, which contains 19 columns and over 10,000 rows of data. The dataset includes information on patient demographics, clinical characteristics, and laboratory results [1].

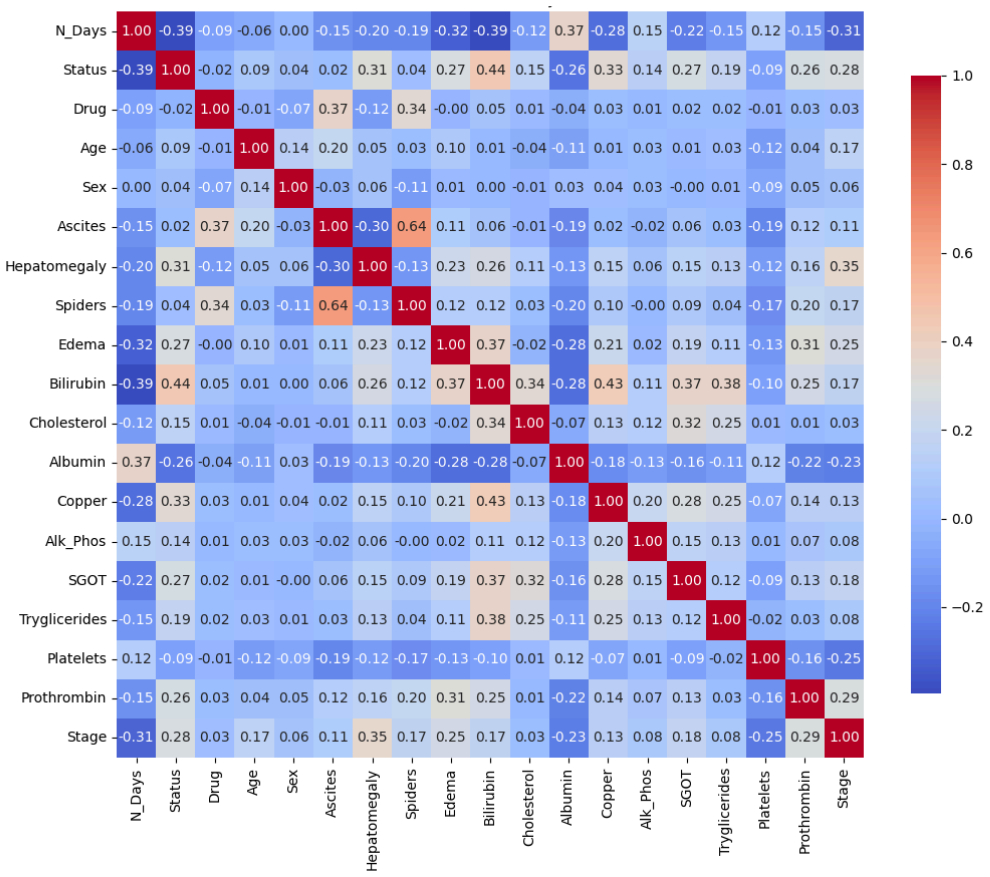
3. Related work

The problem of predicting the stage of liver cirrhosis has been previously explored using various machine-learning techniques and statistical analyses. Commonly used models include logistic regression, support vector machines (SVM), random forests, and various types of artificial neural networks.

4. Methodology

The solution was implemented in the Python programming language using *sckit-learn* [2], *matplotlib* [3] and *seaborn* [4] libraries. Our approach to solving the problem includes the following steps:

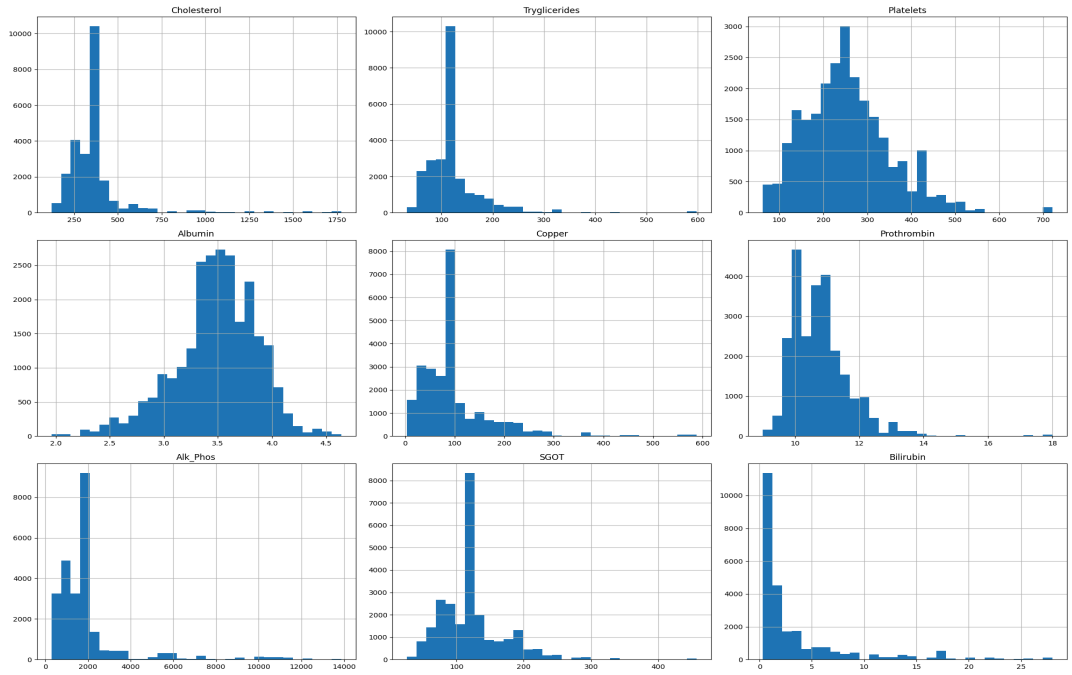
1. **Data Preprocessing:** Conversion of categorical data into numerical values was achieved using the LabelEncoder class from *sckit-learn*. Label encoding was chosen as the optimal encoding strategy, because of the high dimensionality of the problem, as well as the fact that, in some of the encoded features, the order matters. Normalisation was done using *StandardScaler* class, which uses z-score normalisation to scale the data.
2. **Dimensionality reduction:** As the problem's dimensionality is high, the first step of optimization was to reduce said dimensionality. This was attempted using the *PCA* class from *sckit-learn*, which uses Singular Value Decomposition of the data to project it to a lower dimensional space. However, after reevaluating the model, its performance was not improved. After further investigation, it was noticed that the correlation between features was negligible, as can be seen from the picture below:



Picture 1 - Correlation matrix

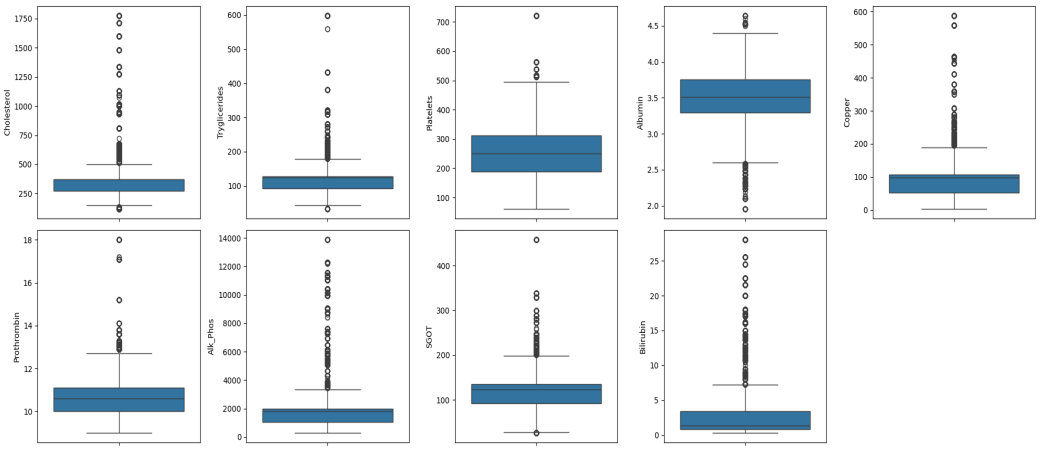
The biggest correlation coefficient was 0.64 between “Spiders” and “Ascites”. This, however, had no major effect on the model's performance.

3. **Outlier Removal:** While analysing the distributions of the features, it was noticed that there are potential outliers in the dataset:



Picture 2 - Data distribution plots

This was confirmed by looking at the box plots of the relevant features:



Picture 3 - Data box plots

However, when the outliers were removed, and the model reevaluated, its performance was actually worse. This is due to the fact that, in medical data, outliers can represent rare but clinically significant cases that are crucial for accurate diagnosis and prediction. For example, extremely high values in certain laboratory tests may indicate a serious condition that is important for predicting the stage of cirrhosis.

4. **Data Splitting:** The dataset is split into training and test sets in an 80:20 ratio.
5. **Model Selection:** We tested four different classification models: Random Forest [5], Bagging [6], Gradient Boosting [7] and Logistic Regression [8].
6. **Hyperparameter Optimization:** Using GridSearchCV for each model to find the best combination of hyperparameters through 5-fold cross-validation.

5. Discussion

The main chosen metric of the evaluation was the **micro f1 score** [9]. Micro f1 score was chosen, because the classes were relatively well balanced:

Stage	Number of samples
1	8265
2	8441
3	8294

Table 1 - Class samples

The table below shows the performances of each model used in the classification. Along with the micro f1 score, included metrics are **precision**, **accuracy** and **recall**.

Model	Accuracy	Precision	Recall	Micro f1
Random forest	0.9552	0.9552	0.9552	0.9552
Bagging	0.9588	0.9588	0.9588	0.9588
Boosting	0.9632	0.9632	0.9632	0.9632
Logistic regression	0.6002	0.6002	0.6002	0.6002

Table 2 - Model performance

6. References

[1] *Kaggle dataset* (url)
[2] *Sckit-learn documentation* (url)
[3] *Matplotlib documentation* (url)
[4] *Seaborn documentation* (url)
[5] *Random forest classifier* (url)
[6] *Bagging classifier* (url)
[7] *Boosting classifier* (url)
[8] *Logistic regression classifier* (url)
[9] *F1 score documentation* (url)