1 Introduction

This project aims to classify the status of patients with Primary Biliary Cirrhosis(PBC), a type of liver disease which gradually gets worse over time and the causes of which are unknown, into 1 of 3 classes. The classes are: alive which will be referred to as 'C', alive and received a liver transplant which will be referred to as 'CL' and died which will be referred to as 'D'. To do this we used a synthetic dataset created from a deep-learning model[1] to train our models. The deep-learning model was trained on data from a 1984 Mayo Clinic clinical trial[2]. The trial tested the therapeutic effectiveness of the drug 'D-penicillamine' on patients with PBC. This report will detail the trends observed in data exploration (Section 2), possible methods of imputation in cases of missing data (Section 3), treatment of unbalanced classes (Section 4), the range of models attempted and their results (Section 5) and our conclusions and recommendations (Sections 6 and 7).

2 Data Exploration

The synthetic data set (referred to as the dataset from here on) had the vitals of 13175 patients, 7905 were used as the training set and 5270 as the test set. The data available for each patient were: *ID, N_Days, Drug, Age, Sex, Ascites, Hepatomegaly, Spiders, Edema, Bilirubin, Cholesterol, Albumin, Copper, Alk_Phos, SGOT, Trygliceride*s, *Platelets, Prothrombin, Stage* and *Status*; descriptions of which can be found in (Appendix A). For each numerical feature, we split the range of the data into equal bins. In each bin, we calculate the fraction of patients of each Status C, CL, and D and plot that fraction to observe any trends between the feature and the label Status as well as the values they occur at (an example can be seen in Figure 1b).

The trends observed in numerical features include:

- Bilirubin levels $>$ 1.4 mg/dl indicate a higher likelihood of death
- *Cholesterol* levels $>$ 200 mg/dl indicate a higher likelihood of death.
- *Albumin* levels $>$ 3.2 gm/dl indicate a lower likelihood of death
- *Copper* levels $>$ 90 ug/day indicate a higher likelihood of death
- *SGOT* levels $>$ 130 U/ml indicate a higher likelihood of death
- *Platelets* levels $>$ 140 indicate a lower likelihood of death
- *Prothrombin* values $>$ 11s indicate a higher likelihood of death
- Patients in Stages 1-3 have a higher likelihood of survival.
  - Patients in Stage 4 have the highest likelihood of obtaining a transplant: 4% compared to patients in Stage 3: 3.6%.

In Figure 2, the distribution of the different classes in each categorical feature is shown. From this, we can see features *Sex, Ascites, Edema* and *Status* are very unbalanced and Spiders is slightly unbalanced. Because of this *Sex, Ascites, Edema* and *Spiders* might not be useful to include in the final model however due to *Status* being the label of our dataset, the feature we are trying to predict, we consider 2 methods to remedy the imbalance in Section 4. The remaining features *Drug* and *Hepatomegaly* are balanced. However, when performing a chi-squared test on the categorical features, *Drug* produces the lowest score at 5.88 and *Hepatomegaly* produces the highest at 1243.5. From this, we infer that the relationship between *Drug* and *Status* is not significant. This significance of a feature when predicting *Status* will be tested further by including all the features in the final model and assessing the features' importance.

[1]Walter Reade, Ashley Chow. (2023). Multi-Class Prediction of Cirrhosis Outcomes. Kaggle. https://kaggle.com/competitions/playground-series-s3e26

[2]Dickson, E. R., Fleming, T. R., Wiesner, R. H., Baldus, W. P., Fleming, C. R., Ludwig, J., & McCall, J. T. (1985). Trial of penicillamine in advanced primary biliary cirrhosis. New England Journal of Medicine, 312(16), 1011–1015. https://doi.org/10.1056/nejm198504183121602

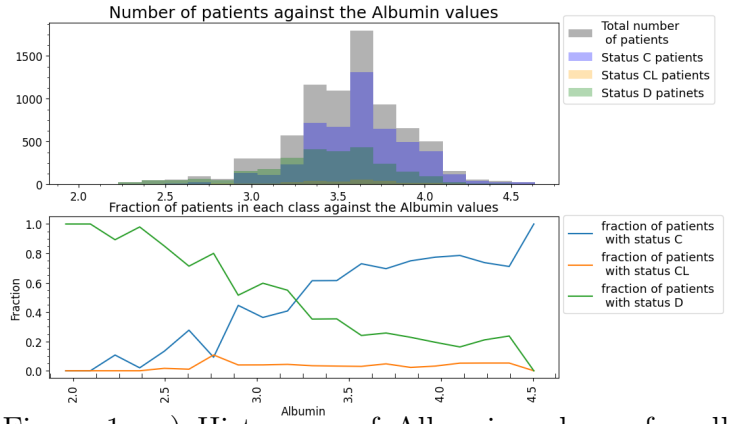

Figure 1: a) Histogram of Albumin values: for all patients in grey, for patients with Status C in blue, for patients with Status CL in yellow and for patients with Status D in green. b) Fraction of Status C, CL and D patients in each bin in blue, yellow and green respectively.

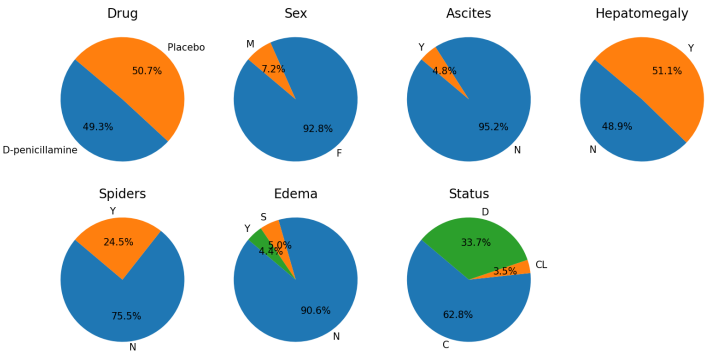

Figure 2: Pie chart of all categorical features in the dataset and their class balances.

3 Imputation Methods

In this report, we also test 3 methods of imputation: Simple Imputer, Iterative Imputer and KNN Imputer to compare with the original data and find which method performs the best at retrieving the original data. We test 2 methods of Simple Imputation. For categorical data, both methods replaced the missing values with the most common value of that feature. For the numerical data, the first method used the mean value as a replacement and the second used the median value. The Iterative Imputer method works by modelling each feature as a function of all the other features making predictions using a regressor as to what the missing value might be. Finally, the KNN Imputer works by considering the nearest neighbours around the observation with a missing value and taking the average of the neighbour's values for that feature. Our dataset has no missing values so we first create a dataset with missing values by randomly removing a percentage of the dataset. Simple imputer and Iterative Imputer performed best at 5% of the data removed and KNN Imputer performed best at 10% of the data removed with 3 nearest neighbours. To compare the methods against each other, the imputed datasets from all the methods were then tested on a decision tree model with 2 parameters: criterion='log_loss' and min_samples_split=100 to ensure consistency. We found that the Iterative Imputer performed the best. Iterative Imputer and KNN Imputer both had similar accuracy however Iterative Imputer allowed for a better classification of the minority class 'CL' while the dataset from KNN imputer led to 0 predictions for the minority class seen in Figure 3.

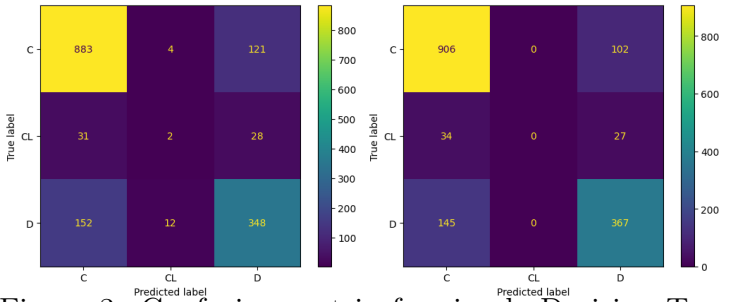

Figure 3: Confusion matrix for simple Decision Tree model using imputed data from an: Iterative Imputer(left) and KNN Imputer(right).

4 Unbalanced Classes

As mentioned in Section 2 the label *Status* is unbalanced. Here we consider 2 possible solutions to remedy the imbalance when training the model: weight adjustment and SMOTE. The method of weight adjustment is to adjust the sample distribution during the training process by assigning different weights to samples of different categories. This method showed a high accuracy (weighted average of 0.76). However, the recognition of minority class CL was poor, and its precision, recall and f1-score were all very low (0.33, 0.01 and 0.02 respectively), which indicates that the model had obvious difficulties in identifying minority class CL.

The SMOTE method increases the number of minority class samples by generating synthetic minority class samples, which can better simulate the distribution of the minority class. The model showed an ac-

curacy comparable to the weights method (weighted average of 0.74) and it also showed a higher recognition ability of the category CL(precision, recall and f1-score were 0.12, 0.41 and 0.19 respectively).
Therefore, although the weight-adjusted method was slightly better than SMOTE in terms of overall accuracy, SMOTE performs better when dealing with extremely imbalanced data sets.

5 Assess Models

We attempted 6 models: Support Vector Machine classifier(SVC), Gradient boosting classifier(GBC), Random Forest classifier, Gaussian Naive Bayes, Logistic Regressor and XGBoost classifier. The first 5 models were made using the `scikit-learn` library and the XBGoost classifier was from the `xgb` library. The final parameters of each can be found in Appendix B.

5.1 SVC

SVCs work by coming up with a boundary between observations that try to split the classes up as accurately as possible. The SVC trained here uses a radial basis function(RBF) to decide these boundaries. RBF calculates the 'similarity' of 2 observations by considering their distances in feature space. The SVC model was trained on the oversampled SMOTE data and initially predicted most observations as Status 'C' with a *logloss* score of 5.996. Attempts at tuning the model reduced the score to 2.11 and increased the prediction of the class 'D' but it also reduced the correct classification of the minority class 'CL' by 50%.

5.2 GBC

We used the SMOTE method to oversample the training data when using this model. This model initially also classified most of the observations into class C, with a *logloss* of 0.519. We then used a random search method to adjust the hyperparameters and applied the best hyperparameters to the validation set for verification, which reduced the *logloss* to 0.518. Other metrics such as weighted average precision, recall, and F1 score also improved.

5.3 Random Forest Classifier

Another one of the models used to classify patients was a Random Forest classifier. Forest classifiers work by building many decision trees that each individually try to sort the data into a classification. The forest then classifies the individual into a group based on which classification was chosen the most by the decision trees generated. The forest classifier has multiple hyperparameters that were manually tuned to find the best-performing model. The best-performing forest classifier model had a *logloss* of 0.506 and an accuracy of 81%.

5.4 Gaussian Naive Bayes

Considering our data involves predicting certain disease conditions in the medical field, the Gaussian Naive Bayes model was attempted as it requires the data features to be continuous variables, assuming they follow a Gaussian distribution. During the training on the dataset, it can be observed that the model performs relatively well on Status C patients, with a recall of 0.91 and a precision of 0.75. However, for the D category, the recall is lower at 0.44, though the precision is higher at 0.81. In contrast, the CL category shows poor performance with a recall of 0.08 and a precision of 0.05. The model's *logloss* was 2.59, which indicates that this model is not very suitable for our data. The possible reason might be that one of the premises of Gaussian Naive Bayes is the need for data features to conform to a Gaussian normal distribution, but some features in our data do not meet this precondition. Moreover, in our data, it cannot be ensured that each feature is independent of one another, which could also affect the accuracy of the Gaussian model. Therefore, this classifier is only attempted and not selected as the best model.

5.5 Logistic Regression Classifier

The logistic regression classifier is used to predict the outcome of patients. It looks at various features and learns how each factor affects the chances of each possible patient outcome. After the model is trained, it then estimates these probabilities for patients in the validation set. This model achieved a correct prediction rate of 65%, meaning it accurately predicted the patient's outcome 65 times out of 100. It was best at

forecasting Status D patients and less precise for the C and CL classes. The *logloss* value was 0.81.

5.6 XGBoost

Regarding the XGBoost model, in addition to similarities with other models, such as good accuracy, recall rates, and F1 scores for the C and D categories, the XGBoost model has a higher recall rate for the CL category compared to the Gaussian Naive Bayes, with an accuracy of 80%. The F1 score results indicate that the model also performs well in balancing recall and accuracy. With a *logloss* score of 0.53, the model had a closer approximation between predicted probabilities and the actual classes. For this model, we have utilized hyperparameter tuning and cross-validation to find the optimal model configuration, resulting in a *logloss* score of 0.50.
Therefore, from the 6 models tested XGBoost performed the best and thus was used as the final model. It was applied to the test set and submitted to Kaggle for a final score of 0.46593 seen in Figure 4.

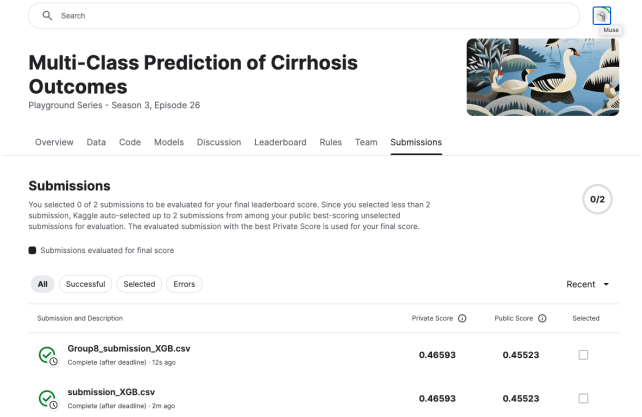

Figure 4: Final Kaggle score on the test set

6 Recommendations

As mentioned earlier, the initial dataset originates from a clinical trial testing the effectiveness of D-penicillamine. By considering the initial data exploration, specifically the chi-squared value of *Drug* we can see that the feature *Drug* has little correlation with a patient's *Status*. Furthermore, by considering the feature importance values seen in Figure 5, we conclude again that *Drug* has little importance in classifying a patient's *Status*. From this, we can infer that D-penicillamine is ineffective in helping treat PBC similar to the initial paper for the clinical trial[3] as well as a subsequent trial by Gong. Y et al[4].

Another observation obtained when considering the feature importance is the split between categorical and numerical features. The least important numerical feature (*Platelets*) is still 4 times more important than the most important nominal categorical feature(*Spiders*). This could be due to the imbalance discussed in Section 2 as the least imbalanced features(*Hepatomegaly, Spiders* and *Drug*) had higher importance scores than the most imbalanced categorical features. The split between numeric and categorical features indicates that categorical features are less significant in the model and further tests are required to evaluate model performances without them.

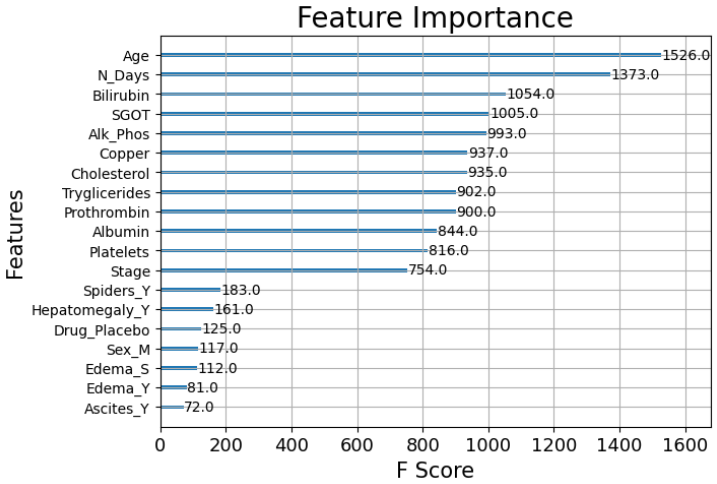

Figure 5: Feature Importance of the final XGB model

7 Conclusion

In summary, our study emphasises comprehensive data exploration, effective imputation techniques, and appropriate methods to balance the data and select appropriate models to predict the results of the test set. In particular, the XGBoost model showed good performance in classifying PBC patient status, with the lowest *logloss* of 0.501 among all selected models. Compared to the baseline estimator, which predicts the patient is in *Status* C for every observation and thus has an accuracy of 63%, our model is more accu-

[3]Dickson, E. R., Fleming, T. R., Wiesner, R. H., Baldus, W. P., Fleming, C. R., Ludwig, J., & McCall, J. T. (n. 2)

[4]Gong, Y., Frederiksen, S., & Gluud, C. (2004). D-penicillamine for primary biliary cirrhosis. Cochrane Database of Systematic Reviews. doi:10.1002/14651858.cd004789

rate with an accuracy of 80% and therefore we believe it can reasonably predict the status of a patient with PBC. A limitation of the models is the use of *N_days*. As seen in Figure 4 it is the 2nd most important feature in our model and as such heavily impacts the predictions. However, it limits the future use cases of the model as this data is not available for patients who are newly admitted. Because of this, we recommend our model be used for patients who have already been admitted for several days. The patient's doctor can then check the resulting prediction of the patient's Status before deciding on a course of action.

A Description of Features

Table 1: Description of features included in the dataset

Feature	Description	Values\Types
ID	Unique Patient Identifier	Integer values between 0 and 13174
N_Days	Number of Days from joining the study to first of death, liver transplant or end of the study	Integer values between 41 and 4795
Drug	Was the patient given D-penicillamine or a placebo	'D-penicillamine' or 'Placebo'
Age	Patient's age in days	Integer values between 9598 and 28650
Sex	Patient's Sex	'M' or 'F'
Ascites	Accumulation of excess fluid in the abdomen	'Y' or 'N'
Hepatomegaly	Enlargement of the liver more than normal	'Y' or 'N'
Spiders	Dilation of blood vessels found under the skin	'Y' or 'N'
Edema	Swelling caused by fluid collection in tissue	'Y', 'S' or 'N'[e]
Bilirubin	created during the breakdown of aged or abnormal red blood cells	Continuous value [mg/dl]
Cholesterol	type of lipid found in the blood	Continuous integer value [mg/dl]
Albumin	A family of proteins made by the liver	Continuous value [gm/dl]
Copper	Metal carried by a protein called ceruloplasmin	Continuous integer value [ug/day]
Alk_Phos	Alkaline phosphatase, an enzyme	Continuous value [U/liter]
SGOT	An enzyme, serves as a marker for liver function	Continuous value [U/ml]
Tryglicerides	A type of lipid found in the blood, serves different purposes from cholesterol	Continuous integer value
Platelets	Cells that bind together to form clots	Continuous integer value [x10-3/mm3]
Prothrombin	Prothrombin time, how long it takes for a clot to form in a blood sample	Continuous value [s]
Stage	Severity of liver disease	1, 2, 3 or 4 where 1 is least severe
Status	Whether the patient is alive, dead or got a liver transplant by the end of N_days.	'C': alive, 'CL': alive and received a transplant or 'D': dead

[e] 'Y' if present and not solved by diuretic therapy, 'S' if present and no diuretic therapy or if solved by diuretic therapy, 'N' if not present

B Model performance and parameters

Table 2: Models attempted, their best *logloss* Scores and the parameters used

Model	LogLoss	Parameters
XGBoost	0.50	objective='multi:softprob', num_class= 3, eval_metric='mlogloss',learning_rate=0.1,max_depth=6, subsample=0.8,colsample_bytree=0.8,random_state=42
Random Forest Classifier	0.506	criterion="log_loss", min_samples_split=5, max_depth=30, min_samples_leaf=2, n_estimators=200
Gradient Boosting	0.52	learning_rate=0.1, max_depth=8, min_samples_leaf=12, min_samples_split=9, n_estimator=120
Gaussian NB	0.7	var_smoothing= 1.0
Logistic Regression	0.81	multi_class='multinomial', max_iter=1000
SVM	2.1	C= 8, gamma= 0.5, kernel= 'rbf'