# ML in Computational Biology Assignment2

Marina Thalassini Filippidou

24/04/23

**Abstract**
In this paper we use the nested cross validation method in order to fine tune and test the perfomance of 5 machine learning algorithms, namely Support Vector Machines, Linear Regression, Gaussian Naive Bayes and Linear Discriminant Analysis, on the Hepatitis C dataset. After fine tuning the algorithms we select the one with the best perfomance on the dataset and we evaluate several perfomance metrics on the dataset.

## Introduction

Hepatitis C is a liver infection caused by the hepatitis C virus (HCV). Hepatitis C is spread through contact with blood from an infected person.Today, most people become infected with the hepatitis C virus by sharing needles or other equipment used to prepare and inject drugs.Chronic hepatitis C can result in serious, even life-threatening health problems like cirrhosis and liver cancer.Thus it is crucial for blood donors to be tested before donating blood to prevent the spreading of the virus. But is there a way to infer if a person has Hepatits or not using specific personal medical information without the need to perform a Hepatitis C test? In this paper we atempt to optimise a model for infering the eligibility of a person for donating blood based on information about their age, sex and several laboratory data.

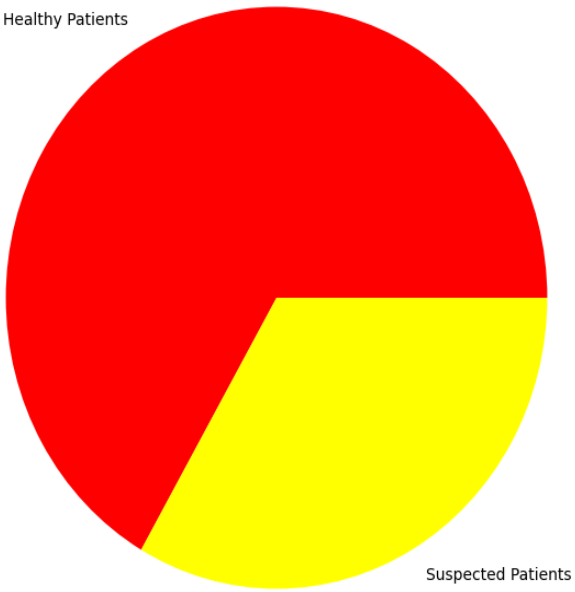
## Methods

**Dataset description** The Hepatitis C dataset is an openly available dataset which consist of 14 attributes. All attributes except Category and Sex are numerical. Although in the dataset that we were given all the Cate-

gorical data were tranformed to numerical. The laboratory data are the attributes 3-12. The target attribute for classification is label (blood donors(1) vs. Hepatitis C(0)).

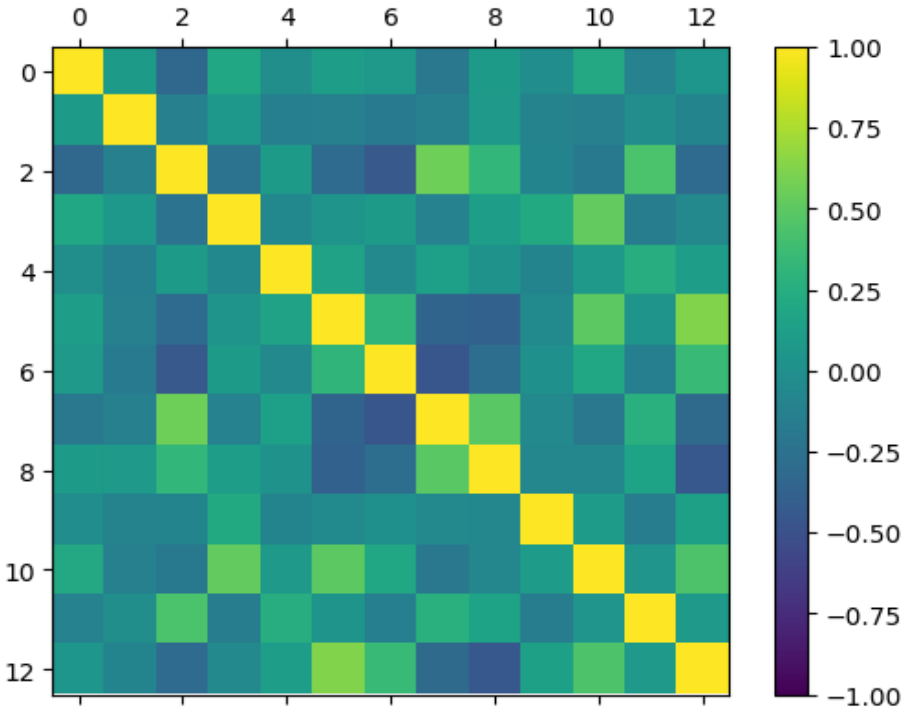
1) Age (in years)
2) Sex (f,m)
3) ALB
4) ALP
5) ALT
6) AST
7) BIL
8) CHE
9) CHOL
10) CREA
11) GGT
12) PROT
13) label (diagnosis) (values: '0=Blood Donor','1=Hepatitis')

```
Total Suspected Patients : 136
Total Healthy Patients : 68
```

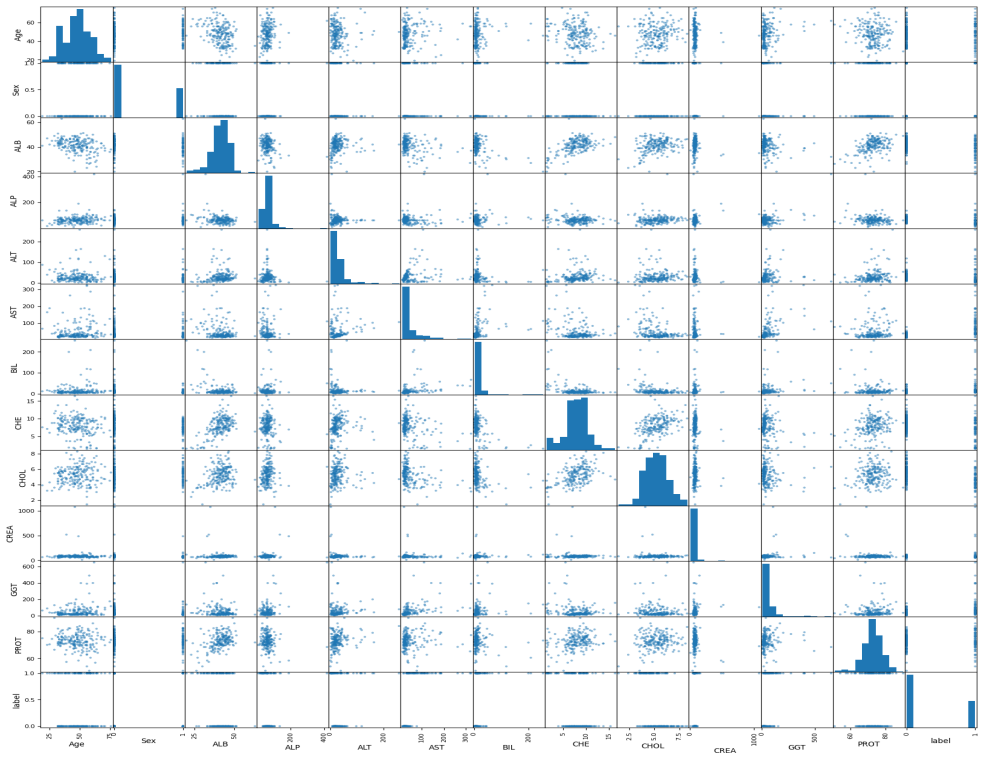



By printing a pie chart of the data we observe that the two classes are im-

balanced. Thus we should be careful not to use evaluation metrics that do not comply with imbalanced datasets sich as accuracy.
We can also print a heatmap of the correlation to see if there are any strongly correlated features.

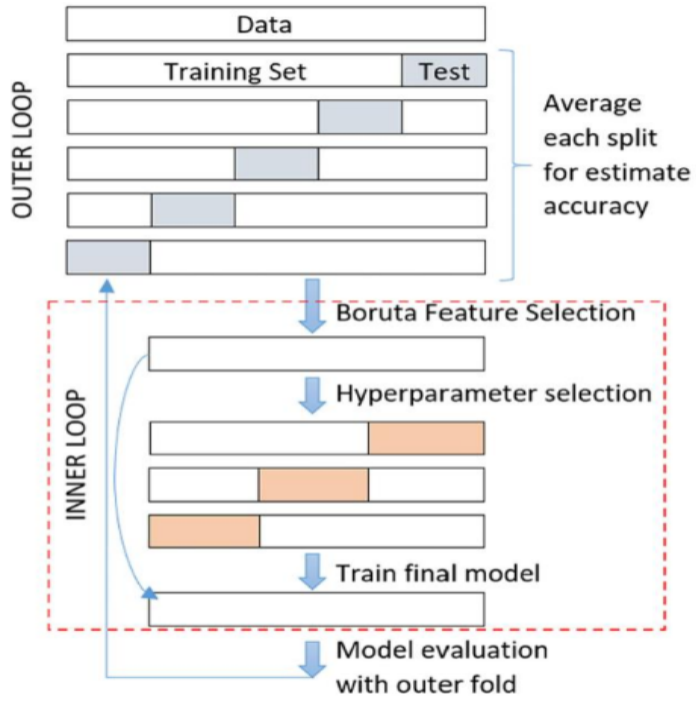



As we can observe the correlation is not significantly big (in the range -0.5-0.5). Thus we can continue our analysis treating the features as independent.
We can use pandas.plotting to plot a scatter matrix of our features.



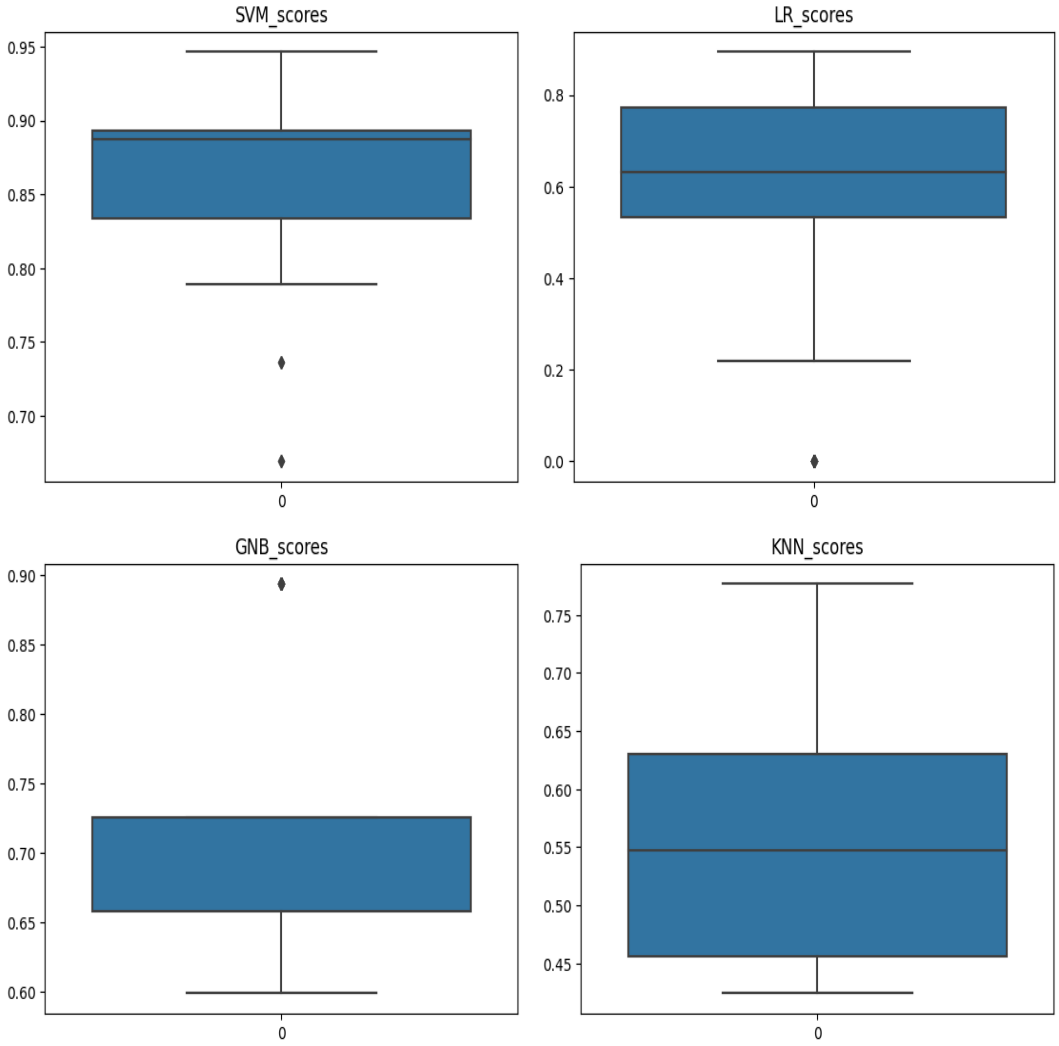

We verify that there are no nan values in the dataset. Additionally we can verify that all the data have numerical values. As for the preprossecing we embed a minmax prerpocessing step inside our nested CV functions but this will be discussed in the next paragraph.

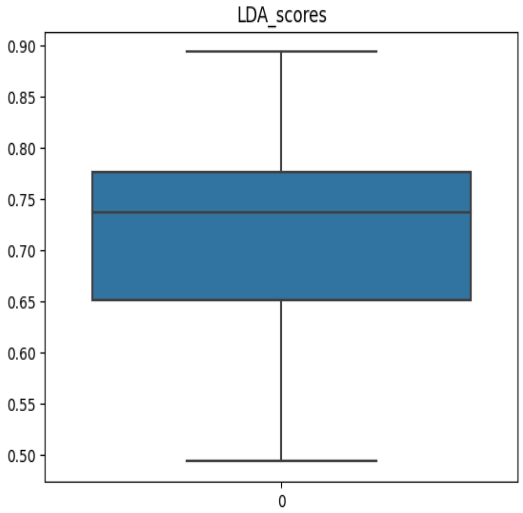
**Pipeline structure**




In order to tune the hyperparameters of each of the five algorithms we construct for each one of them a function which applies nested Cross Validation with inner loop folds K=5 and inner loop folds L=3. We use the Stratified cross validation algorithm by sklearn in order to account for our unbalanced dataset. We add a minmax normalization step inside each fuction as a preprocessing step. In each of the inner loops repetition we use optunas optimize function with 100 number of trials to optimize the hyperparamaters. We select the Tree-structured Parzen Estimator algorithm for sampling hyperparameters. For the training and model selection we used the Mathews Correlation Coefficient (MCC). In the outer loop we evaluate the perfomance of the optimized algorithm using again the MCC. The function returns a list of 5 MCC values. Then we test all our algorithms runing the functions for the whole dataset 10 times and we evaluate the mean of the 50 MCC values produced. We also plot a boxplot of the 50 values for each algorithms.
These are the mean MCC values of all the 50 trials for each algorithm:
Support Vector Machines MCC Score: 0.8567259002528494
Linear Regression MCC Score: 0.5501159538783542
K-Nearest Neighbours MCC Score: 0.5646210879189877

Linear Discriminant Analysis MCC Score: 0.7174646111869111
Gaussian Naive Bayes MCC Score: 0.7199876871393086
Below we can see the 5 boxplots.






As we observe the SVM algorithm has the best perfomance in the Hepatitis C dataset.

After finding the winner algorithm using the nested CV, we use the whole dataset and a five fold cross validation to determine the "final-model".We end up with the following hyperparamters
Optimized parameters: 'C': 8, 'kernel': 'rbf', 'degree': 1, 'gamma': 'scale'

**Results and Discussion**

We create and evaluate a model with the optimized hyperparametes with a simple 10 fold cross-validation using the confusion matrix, Sensitivity/Recall, Specificity, PPV/Precision, NPV, F1-score, False positive rate and True positive rate metrics. The results are the following.
2X2 confusion matrix:'['TP', 135, 'FP', 6, 'FN', 1, 'TN', 62]
Sensitivity/Recall:'0.993
Specificity:0.912
PPV/Precision:0.957
NPV:0.984
F1-score:0.975
False positive rate:0.088
True positive rate:0.993
We also evaluated the perfomance of the model with the best mcc score of the

previous 50 trials(Optimized parameters: 'C': 8, 'kernel': 'rbf', 'degree': 1, 'gamma': 'scale'). An interesting finding is that the two models had exactly the same scores and thus we conclude that the model can achieve the same scores for different hyperparameter combinations.

For the feature selection step we used the MRMR method for selecting 5 and 7 optimal features. We found through trial and error that 7 selected features yield the best scores. We rerun the hyperparameter optimization nested CV(5) step and end up with the following scores for the model.
For the 5 features:
2X2 confusion matrix:'TP', 98, 'FP', 4, 'FN', 3, 'TN', 99 Sensitivity/Recall:0.97
Specificity:0.961
PPV/Precision:0.961
NPV:0.971
F1-score:0.966
False positive rate:0.039
True positive rate:0.97
For the 7 features:
2X2 confusion matrix:'TP', 101, 'FP', 3, 'FN', 0, 'TN', 100
Sensitivity/Recall:1.0
Specificity:0.971
PPV/Precision: 0.971
NPV: 1.0
F1-score: 0.985
False positive rate:0.029
True positive rate:1.0

We observe that in the case of the 5 selected features the Specificity and Precission have increased though the rest of the metrics have deteriorated. In case of the 7 selected features we observe that all the scores have imporved. We conclude that feature selection can be a powerful tool but we may need domain specific knowledge in order to select the right number of features.

**Conclusions**

In our analysis we evaluate 5 algorithms on the Hepatitis C dataset. We tune their hyperparameters using nested Cross Validation and we compare their results on 10 trials of the dataset. We observe that the SVM algorithm

had the best perfomance. We then optimized it's hyperparametes again to attain the final model. We then used feature selection to improve the perfomance of our model.
The limitations of our research is the computational time needed to perform the hyperparameter tuning with cross validation. Although optuna is reported to perform better in terms of computational time compared to scikitlearn GridSearch, the whole 50 trials for the 5 algorithms required almost one hour to run.

**References**

data: https://archive.ics.uci.edu/ml/datasets/HCV+data
hepatits C: https://www.cdc.gov/hepatitis/hcv/index.htm
hyperparameter-tuning: https://datagy.io/python-optuna/
https://towardsdatascience.com/exploring-optuna-a-hyper-parameter-framework-using-logistic-regression-84bd622cd3a5
MRMR: https://towardsdatascience.com/mrmr-explained-exactly-how-you-wished-someone-explained-to-you-9cf4ed27458b
optuna: https://optuna.readthedocs.io/en/stable/index.html