Hepatitis C Machine Learning Pipeline Technical Report

Psallidas Kyriakos

[1]Department of Computer Science and Telecommunications, National and Kapodistrian University of Athens

Abstract

This report presents a machine-learning pipeline developed for hepatitis C virus (HCV) prediction. The aim is to assess the performance of different classification algorithms in the task at hand and reliably select and optimize the hyperparameters of the best-performing one. The baseline pipeline designed for HCV prediction consists of three main stages: data preprocessing, model evaluation, and final model selection and tuning. The model evaluation stage utilizes a developed custom class called ***ClassifierCV****, compatible with the sci-kit learn API that performs classifier algorithm evaluation using nested cross-validation trials and Bayesian hyperparameter tuning with optuna. The performance of several classification algorithms was evaluated, including linear, rbf, sigmoid, poly-support vector machines (SVMs), K nearest neighbours (KNN), linear discriminant analysis (LDA), Gaussian naive Bayes (GNB) and lasso,ridge, elastic logistic regression (LR). The results indicate that SVMs with a liner kernel performed best on the 12-D feature space.*

Introduction

Hepatitis C virus (HCV) is a viral infection that results in liver inflammation. While some HCV infections may be short-term, asymptomatic, and not life-threatening, the majority (approximately 70%) progress to chronic infection, which progressively increases the risk of developing cirrhosis [1], a serious and life-threatening condition characterized by the formation of scar tissue in the liver. Due to the crucial importance of early detection and/or clinical stage of HCV, Machine learning algorithms that are able to uncover patterns in laboratory data that may not be apparent to humans are a valuable non-invasive tool to assist medical personnel.

In this context, the aim of this report is to present in a machine-learning pipeline developed for HCV prediction to assess the performance of different classification algorithms in the task at hand and reliably select and optimize the hyperparameters of the best performing one.

Methodologies

Data-set Description

This study utilizes a dataset consisting of 204 samples, comprising data on hepatitis C patients and healthy blood donors. Each sample corresponds to a patient's ID, and 12 features are available for each donor. To visualize the data-set, a t-SNE projection based on the minimization of the Kullback-Leibler divergence between the estimated t-distribution kernel density joint probability function of the data points in the original high-dimensional space and a lower-dimensional space was utilized (figure 2(a)). The first two features of the dataset capture the age and sex of the donor, while the remaining features represent the levels of various molecules obtained from blood chemistry results, specifically from liver function tests as presented in (table 1). Concerning the target labels, the blood donors are binary classified into healthy (label=0), or HCV positive (label=1). It's important to note that as with most medical datasets, the samples for the labels are imbalanced with the HCV-positive class being the minority class and accounting for 50% of the number of samples of the majority HCV-negative class.

Table 1: *Features*

Feature	Category
Age	Discrete
Sex	Binary
ALB (albumin)	Continuous
ALP (Alkaline phosphatase)	Continuous
ALT (Alanine transaminase)	Continuous
AST (Aspartate transaminase)	Continuous
BIL (Bilirubin)	Continuous
CHE (Cholinesterase)	Continuous
CHOL (Cholesterol)	Continuous
CREA (Creatinine)	Continuous
GGT (Gamma-glutamyltransferase)	Continuous
PROT (Total Protein)	Continuous

Many of the laboratory results featured in the data set are measured in different scales, either units per liter (U/L), grams per deciliter (g/dl) or milligrams per deciliter (mg/dL). [2]. Additionally, the majority of them include outliers, observations above or below 150% of the interquartile range (IQR) from the first

and the third quartile respectively (figure 1).

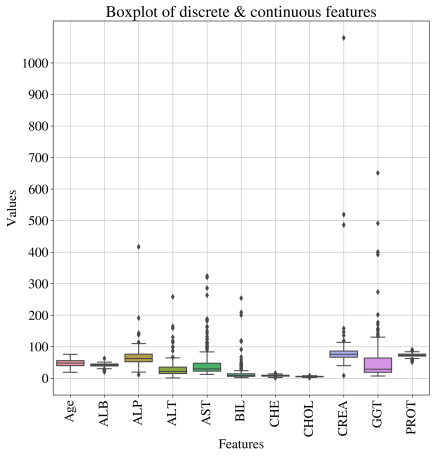


Figure 1: *Box-plot of the un-processed features excluding binary features*

Base Pipeline

Machine learning pipelines provide several significant advantages, such as reproducibility, scalability, and efficiency, by presenting a concise and interpretable way to run machine learning methods. The baseline pipeline designed for Hepatitis C prediction consists of three main stages: data preprocessing, model evaluation, final Model Selection & tuning.

Data Preprocessing with the inclusion of **outlier removal** was investigated, but it was deemed unsuitable because it results in losing 69% of the HCV-positive class and only 12.5% of the HCV-negative class. This observation is reasonable in the context of medical laboratory data, where a large deviation from the normal range is often indicative of pathology. Regarding **feature scaling**, both standardization and normalization methods were examined, but ultimately normalization was chosen as illustrated in the "Preprocessing" block in (figure 3). This decision was made because normalization bounds the feature values between $0-1$, which, when combined with the binary values in the sex feature $\in \{0,1\}$, effectively differentiates between the sexes for classification models. This clear distinction is evident in (figures 2(b) and 2(c)). Of course, this places significantly more weight on one specific feature and can introduce artificial bias, thus should be reasonably justified. The logic behind this choice is the fact that most often laboratory result expected levels differ between males and females. This claim is validated specifically for liver function test in [2].

For Model Evaluation , a custom class called *ClassifierCV* was created. This class is fully compatible with the sci-kit learn API and is designed to perform classifier algorithm evaluation using nested cross-validation trials and Bayesian hyper-parameter tuning inside the inner cross-validation loop with the optuna library [3, 4]. The "Model Evaluation" block in (figure 3) showcases this approach. By combining nested cross-validation and Bayesian hyper-parameter tuning, *ClassifierCV* enables a comprehensive evaluation of the model's performance and ensures that the model is not over-fitting on a specific train/test split. *ClassifierCV* accepts several classifier categories, including K nearest neighbours (KNN), Gaussian naive Bayes (GNB), lasso logistic regression (LR), ridge LR, elastic LR, linear discriminant analysis (LDA), polynomial kernel support vector machines (SVM), radial basis function kernel SVM, sigmoid kernel SVM, and linear kernel SVM. The F_{beta} score is used as the metric to be maximized by optuna's objective function inside the inner cross-validation loop and the model with the hyperparameters that produced the highest F_{beta} score is evaluated in the outer cross-validation loop each time.

$$F_{beta} = (1 + beta^2)\frac{precision \times recall}{(beta^2 \times precision) + recall}$$

The $beta(deault = 1)$ parameter is a class argument and can be set by the user. For Hepatitis C prediction the most important task is to minimize the number of blood donors that are HCV positive but are classified as HCV negative (FN). For this reason we have set $beta = 2$ to prioritize a better $Recall = \frac{TP}{TP+FN}$ over $Precision = \frac{TP}{TP+FP}$. Another important parameter of *ClassifierCV* is the class weight, which determines the weight assigned to each class in applicable algorithms such as LR and SVM. By default, the weights are set to equal $\{1:1, 0:1\}$. In the case of the Hepatitis C prediction pipeline, the weight for the minority HCV-positive class is set to double compared to the negative class $\{1:2, 0:1\}$ to introduce further cost sensitive learning to the imbalanced dataset. The remaining parameters correspond to the number of cross-validation trials, number of outer stratified cross-validation splits and number of inner cross-validation trials. We have utilized 10 uniquely seeded trials of 5 outer and 3 inner splits, for a total of 50 nested cross-validation loops per classifier evaluated. For each one of the 50 loops The best classifier parameters from the inner loop and the F_1, F_{beta}, *ROC AUC*,*Balanced Accuracy*, *Mathews Coefficient*, *Precision*, *Recall*, *Specificity*, *Negative Predictive Values* are collected and stored in a data-frame.

Final Model Selection & Tuning is accomplished by picking the classifier algorithm with the highest *Matthews correlation* (MCC) mean score across the

50 nested cross validation loops.

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}}$$

The MCC metric is chosen as the selection criterion because it yields high scores only when the classifier accurately predicts both positive and negative data instances. This makes it insensitive to class balance. MCC ranges between -1 and +1, where -1 and +1 denote perfect misclassification and perfect classification, respectively. A score of 0 indicates the classifier is performing randomly [5].

Once the optimal classifier algorithm is identified, it is fitted to the entire dataset's features X. Bayesian hyperparameter tuning is performed using a non-nested 5-fold cross-validation loop of *ClassifierCV* to select the best hyperparameters for the classifier with optuna.

Feature Selection Methods

To select the top five features and assess the selections impact on the performance of the baseline pipeline, we utilized two methods that assign importance to features with respect to the target label. The first method involves selecting the five features with the highest **ANOVA F-score**, which measures the ratio of the variance of each feature between class labels to the variance within each feature group. As ANOVA F-score relies on linear relationships, we also employed a second selection method based on entropy, which does not assume any specific relationship. This method involves calculating the **Mutual Information** between each feature X and label Y, using the formula

$$I(X;Y) = H(X) - H(X|Y)$$

Mutual Information measures the reduction in uncertainty about the label by knowing each of the features.

Results & Discussion

Classifier algorithm evaluation

The results from the model evaluation pipeline stage, presented in (figure 4), show that linear-SVC classifiers outperformed other classifiers in binary classification between healthy blood donors and HVC-positive donors. They achieved the highest mean and lowest standard deviation across all classification metric scores, except for specificity, as demonstrated in (figure 4(a) and Figure 4(b)). Specifically, linear SVC classifiers achieved a mean MCC score of 0.829 with a standard deviation of 0.101 over 50 cv loops. This represents an improvement of $\approx 14\%$ in mean performance compared to the GNB baseline classifier, which had a mean MCC score of 0.717. However, the improvement in classifier prediction standard deviation was minor, as it remained similar between the linear-SVC and the baseline classifier ($0.101 \sim 0.108$).

Regarding the lower specificity of linear-SVC, they achieved a mean score of 0.956, which is slightly lower than the scores of KNN and LDA classifiers, which were 0.965 and 0.979, respectively. This indicates that the linear SVC is slightly less effective at correctly identifying HVC-healthy donors. However, its mean recall score of 0.860 is by far the highest (second-best elastic LR: 0.841), making it the most capable classifier of minimizing the misclassification of HVC-positive patients as healthy, the most crucial task that a model deployed to assist in the diagnostic setting should perform.

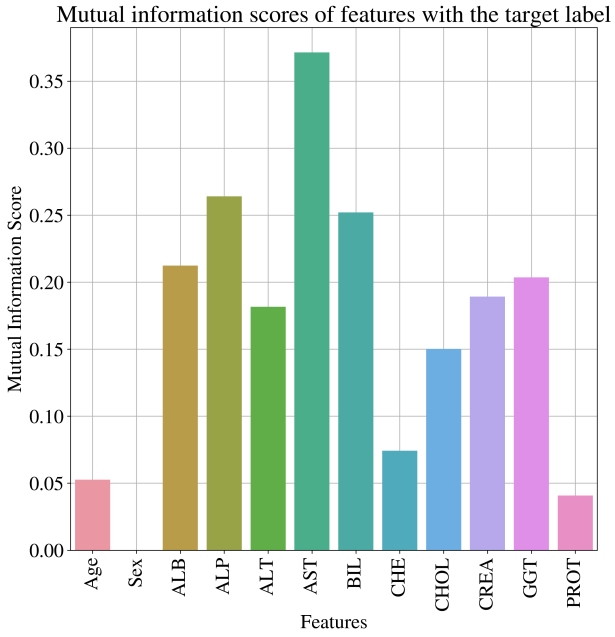
Hyper-parameter distributions

The stored best parameters for each of the 50 cross-validation loops identified by optuna in the model selection pipeline stage, were utilized to estimate the probability density function of the regularization parameter C for both Support Vector Classifier (SVC) and Logistic Regression (LR) classifiers in the context of HVC prediction. It is worth noting that smaller values of C in both cases result in reduced fixation of the model on the training set, at the cost of increased bias. Therefore, choosing an appropriate value for C is crucial for achieving a balanced trade-off between variance and bias. The distributions of the parameter C are illustrated in (figure 5(a)), for the majority of SVC and LR classifiers the mean of the distributions lies between $10 < mean < 100$ with an original parameter space of 0.001 to 100.

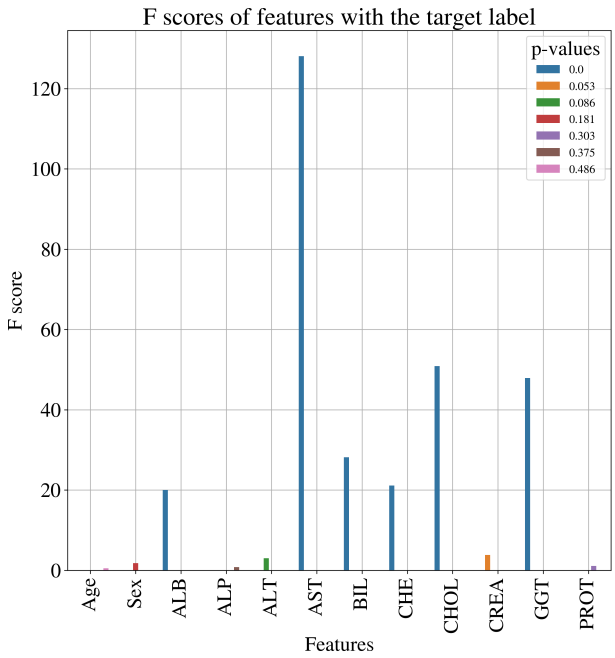
For non-linear SVC classifiers, it is important to consider the hyper-parameter *gamma* in conjunction with the distribution of C. *gamma* controls the shape of the decision boundary and acts as an inverse regularization strength parameter similar to C. We present the joint distributions of $C, gamma$ for non-linear SVC in (figure 5(b)) which showcase clearly their negative covariance.

Results from feature selection

Based on the ANOVA F-score, the top five features in descending order are AST, CHOL, GGT, BIL, and CHE. The feature ALB has a slightly lower score than CHE and is therefore not selected. The rest of the features have a score close to zero, as shown in (figure **??**). On the other hand, the top five features selected based on the Mutual Information (MI) score are AST, ALP, BIL, ALB, and GGT, again in descending order. Unlike the ANOVA F-score selection,now with the



lack of assumption about specific relationships (e.g. linear) the remaining features such as CHOL, CREA, and ALT seem to carry a significant amount of information about the class label (figure **??**) . Running



the pipeline on the top five features selected based on mutual information score the best classifier algorithm is SVC with polynomial kernels with a mean MCC score of 0.802, an $\approx 3.3\%$ decrease over the top score of linear SVC classifiers without feature selection. Other classification metrics (excluding recall with a $> 1\%$ increase) are also lower and especially precision $= 0.862$ suffers a $\approx 6\%$ decrease over the original precision of 0.909. This illustrates that in this feature space classifiers have a harder time predicting True cases of HCV virus. Another important observation is the fact that the GNB and KNN classifiers are the only models to benefit in performance $\approx 3.5\%$ and $\approx 19\%$ respectively (figure 6(a)). KNN performs much worse in higher dimensions due to the curse of dimensionality. GNB on the other hand may have a better score in this feature space because this features hold better to the Independence assumption than the total of 12 features. Regarding the classification performance after the selection of the top five features based on ANOVA F-score it's substantially worse. The best classifier is ridge logistic regression with an MCC score of only 0.791. Additionally, the recall scores of all the classifiers are below 0.779 (figure 6(b)). Consequently it's evident that neither selection method improves the performance of the top model, however, Mutual information is much more promising and the inclusion of more features than five might have provided better performance than the original.

Conclusions

In the context of HCV prediction, we developed a pipeline to evaluate and select machine learning models for binary classification. Our pipeline includes a custom-built class called *ClassifierCV* that handles imbalanced datasets and weighted recall and precision. We used 50 nested cross-validation loops to assess feature scaling methods and feature selection methods. We found that normalization was the best feature scaling method for the HVC dataset, and a linear-SVC model had the highest mean MCC score of 0.829. However, reducing the feature space to five features using ANOVA F-score and Mutual Information did not improve the classification performance of the best-selected model by the pipeline. Therefore, we recommend considering other feature selection methods and/or increasing the number of top features. Currently, *ClassifierCV* can only evaluate one classifier per class call using nested cross-validation. To improve the pipeline, we suggest allowing *ClassifierCV* to read classifiers from a list internally, eliminating the need for looping and providing a cleaner and more concise pipeline. Regarding imbalanced datasets, the cost-sensitive learning method we used is only one of several methods to deal with them. We recommend utilizing additional methods, such as oversampling with SMOTE, to extend the base pipeline presented in this report.

References

[1] *Hepatitis C*. en. URL: https://www.who.int/news-room/fact-sheets/detail/hepatitis-c (visited on 04/18/2023).

[2] BR Thapa and Anuj Walia. "Liver function tests and their interpretation". In: *The Indian Journal of Pediatrics* 74 (2007), pp. 663–671.

[3] Takuya Akiba et al. "Optuna: A next-generation hyperparameter optimization framework". In: *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*. 2019, pp. 2623–2631.

[4] Fabian Pedregosa et al. "Scikit-learn: Machine learning in Python". In: *the Journal of machine Learning research* 12 (2011), pp. 2825–2830.

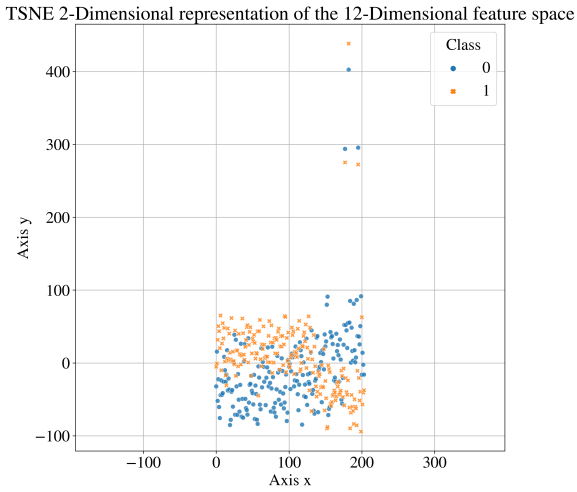
[5] Davide Chicco and Giuseppe Jurman. "The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation". In: *BMC genomics* 21 (2020), pp. 1–13.

Figures

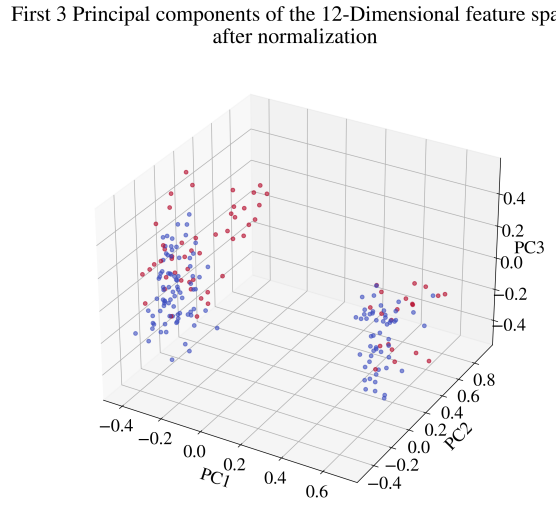
The figures referenced in the methods & results sections of the report are showcased on the following pages: $6 - 10$.

Code Availability

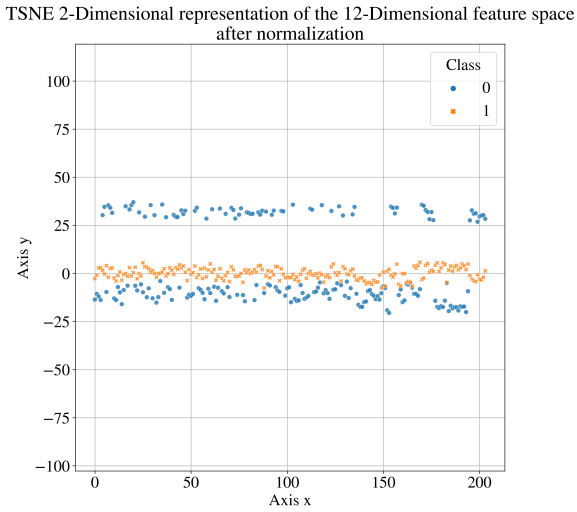
The code to reproduce the results of the paper and/or extend the base pipeline is available in this GitHub repository.



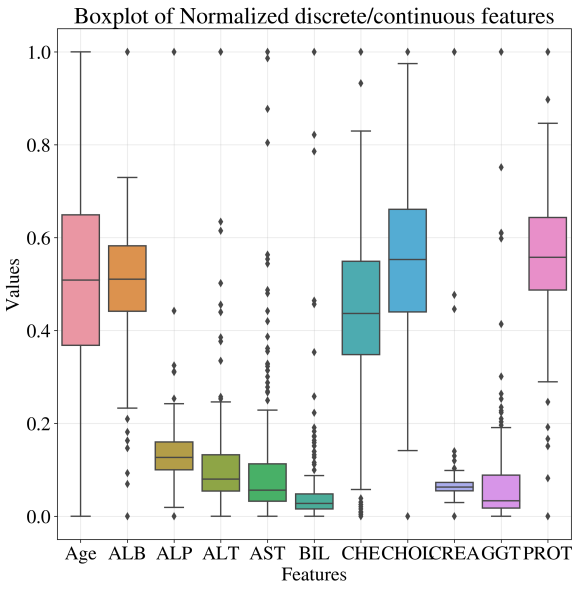
(a) TSNE visualization of the original data-set



(b) First 3 principal components (PCs) of the normalized data-set



(c) TSNE visualization of the normalized data-set



(d) Boxplot of the normalized data-set excluding binary features

Figure 2: *Hepatitis C exploratory data analysis findings*

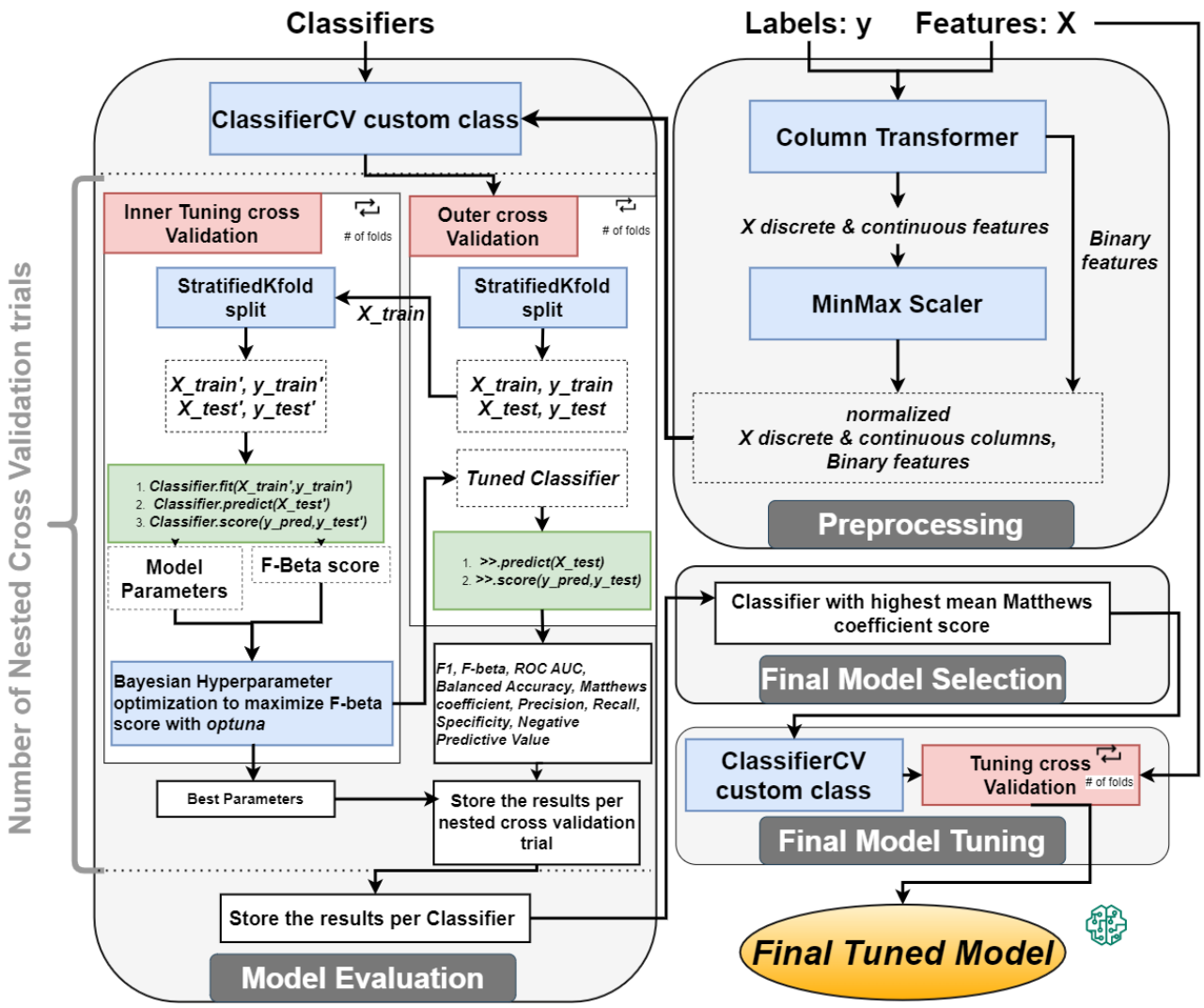
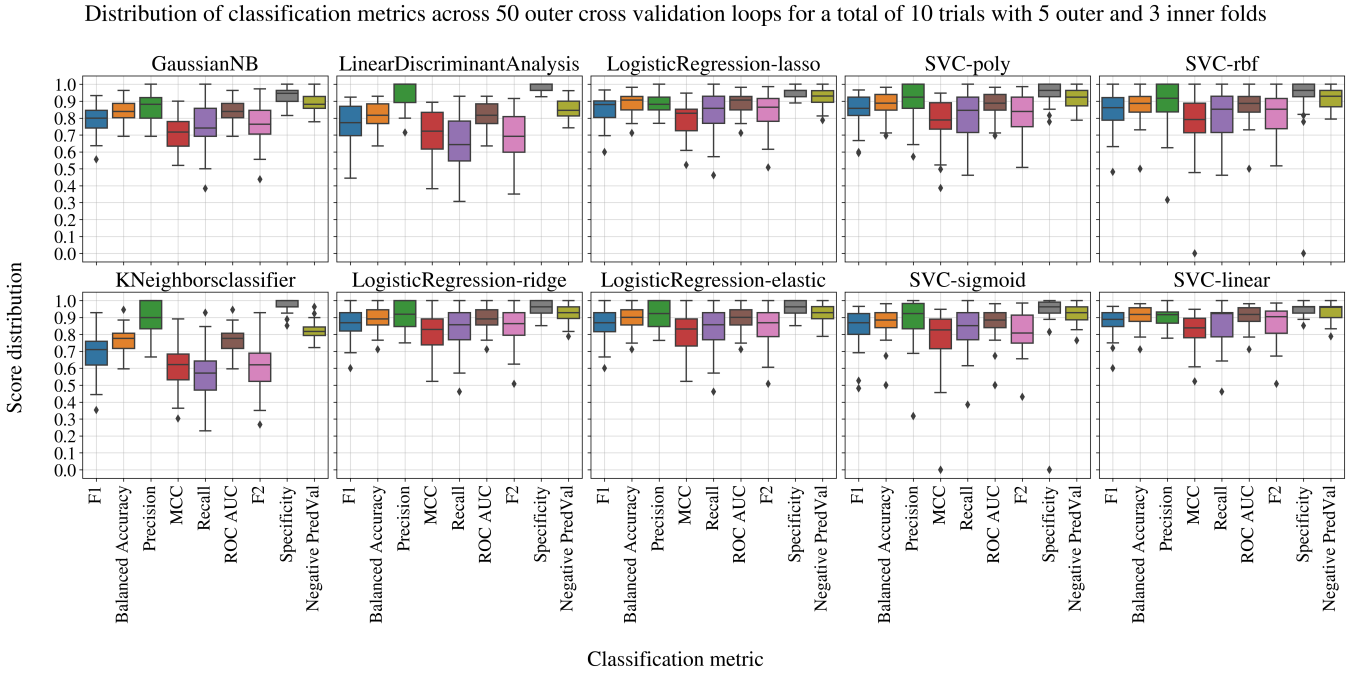
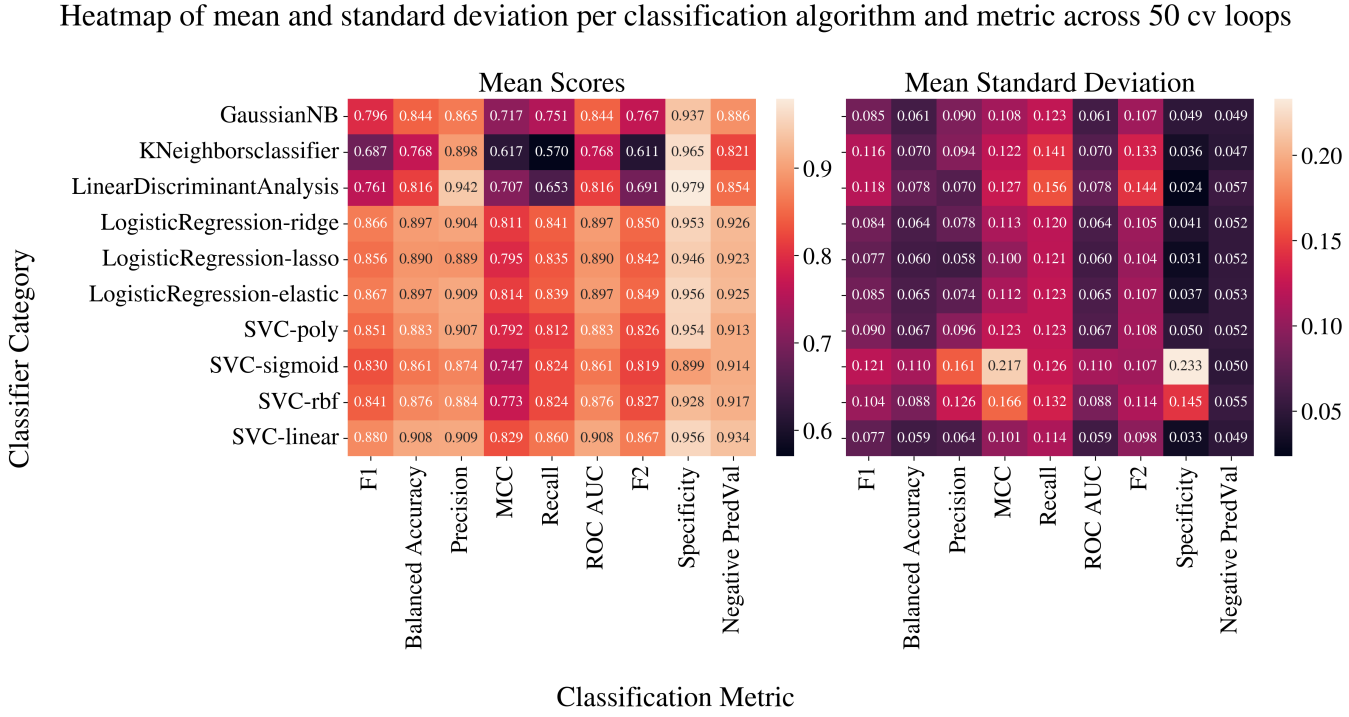


Figure 3: *baseline Hepatitis C machine learning Pipeline*

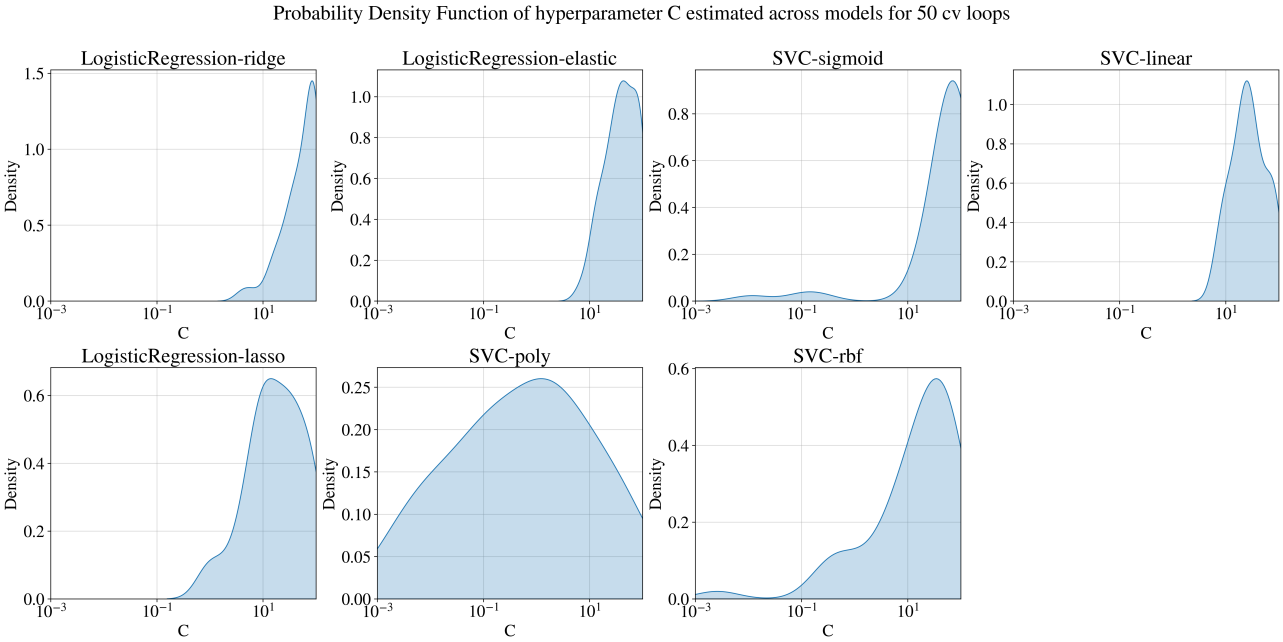


(a) Score distribution for each classifier categories after 50 outer cross-validation loops

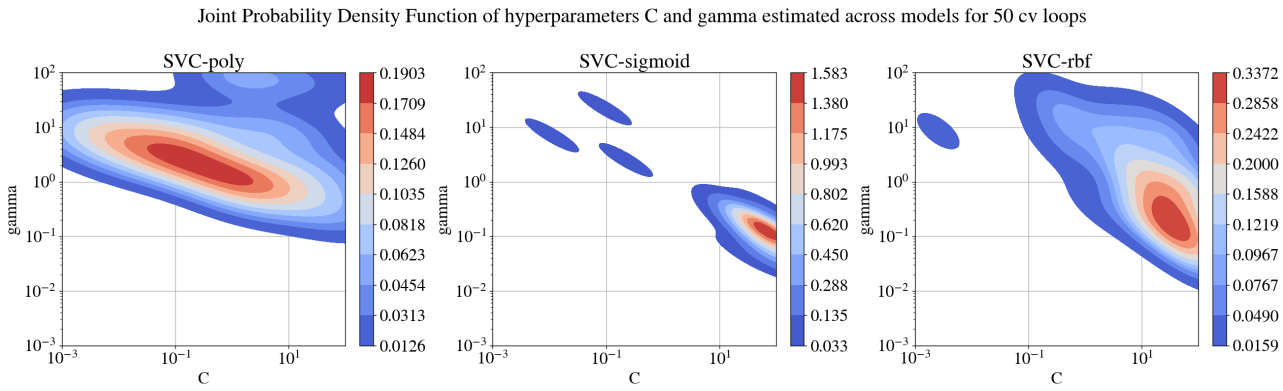


(b) Exact score mean and standard deviation for each classifier categories after 50 outer cross-validation loops

Figure 4: *Results from the model evaluation pipeline stage*

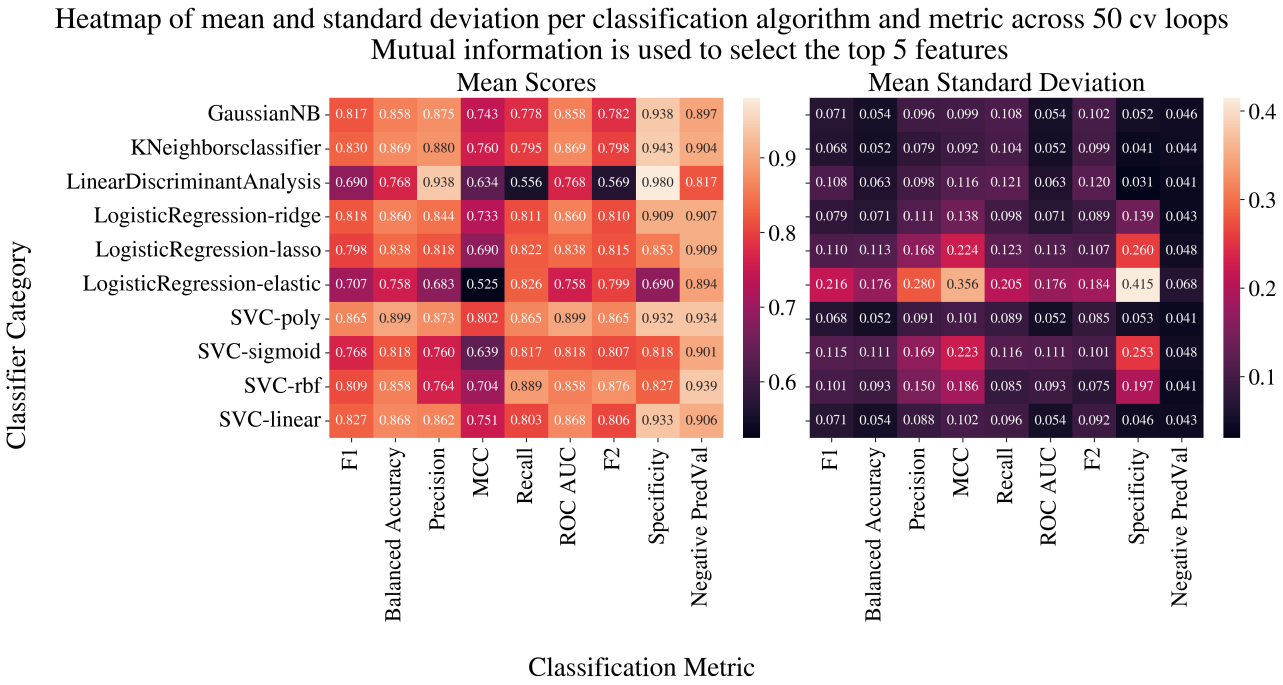


(a) Hyper-parameter C distribution for all applicable classifiers

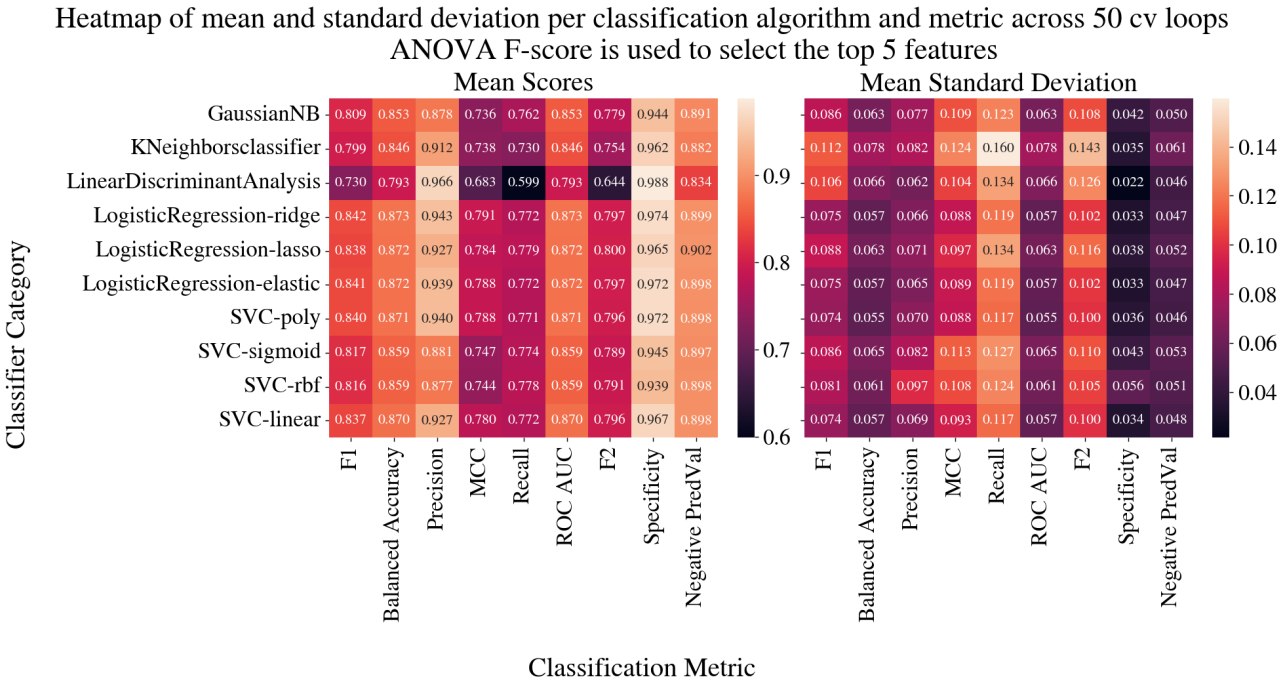


(b) Hyper-parameter C, gamma joint distribution for all applicable classifiers

Figure 5: *Hyper-parameter distribution per classifier algorithm across the 50 cv loops*



(a) Classification score based on the MI selected features



(b) Classification score based on the ANOVA F-score selected features

Figure 6: *Results from the feature selection performance*