Assignment 2: *Nested CV on Hepatitis C dataset*

Konstantinos Mexis

Department of Process Engineering, NTUA, Iroon Politechniou 6 Zografou, Athens, Greece

Abstract

Hepatitis C infection is an insidious disease and early diagnosis of the disease is not possible in many cases. Early diagnosis and treatment have an important place for the disease. The diagnosis of the disease can be made with the use of machine learning methods. In this study, a complete ML pipeline was created in order to classify future patients as Hepatis C patients or healthy blood donors. For this task, we built a *nested Cross Validation (nCV)* pipeline to compare systematically the expected performance of multiple classification algorithms on unseen data. The pipeline in this study can be used as an alternative method in the diagnosis of Hepatitis C-related diseases.

**Keywords**: Hepatitis C; machine learning; nested cross validation;

* 1. Introduction

Hepatitis C is a viral infection that causes the inflammation of the liver. It is caused by the hepatitis C virus (HCV), which is one of the most important global health problems in the world. Worldwide, 350,000 people out of 185 million patients infected with HCV die from diseases caused by HCV. HCV infections pose serious problems on a global scale. Since there is no vaccine yet to prevent HCV infection, it is essential to prevent infection. Early detection of patients and people at risk is critical to prevent the spread of HCV infection. The contribution of this study is to create a complete ML pipeline to process the collected data and classify successfully future patients.

* 1. Methods
     1. Dataset description

The data set used in this study is a part of the HCV data set in the UCI Machine Learning Repository. The given dataset consists of 204 rows and 12 features. The features are described in the following table:

|  |  |  |
| --- | --- | --- |
| **Feature** | **Description** | **Data type** |
| Age | Numerically it is the age value in years | int64 |
| Sex | Female=1, Male=0 | int64 |
| ALB | Numerical value of laboratory test data | float64 |
| ALP | Numerical value of laboratory test data | float64 |
| ALT | Numerical value of laboratory test data | float64 |
| AST | Numerical value of laboratory test data | float64 |
| BIL | Numerical value of laboratory test data | float64 |
| CHE | Numerical value of laboratory test data | float64 |
| CHOL | Numerical value of laboratory test data | float64 |
| CREA | Numerical value of laboratory test data | float64 |
| GGT | Numerical value of laboratory test data | float64 |
| PROT | Numerical value of laboratory test data | float64 |

The target variable is the *label* column. ‘label=1’ corresponds to a Hepatitis C patient (positive class) and ‘label=0’ corresponds to a healthy blood donor (negative class).

* + 1. Data pre-processing

Data pre-processing is very important for the correct operation and high performance of AI algorithms. In the given dataset, there was no missing data. As a result, no data imputation techniques were used in this study.

As a preprocessing step, we performed data normalization on all the numerical features, meaning all the dataset features except the categorical feature ‘Sex’. Data normalization ensures that each feature is on the same scale, reducing the impact of outliers and extreme values. Normalization helps improve the performance and reliability of a machine learning model. In this study, we used the *Standardization Scaling,* subtracting the mean of each observation and diving by the standard deviation.

* + 1. Pipeline

In this study, the expected performance of the following classification algorithms was examined in the problem of diagnosis of Hepatitis C patients:

|  |
| --- |
| **Classification algorithm** |
| Logistic Regression (LR) |
| Gaussian Naïve Bayes (GNB) |
| K-Nearest Neighbors (kNN) |
| Linear Discriminant Analysis (LDA) |
| Support Vector Machines (SVM) |

In order to select the best machine learning model, we built a nested Cross Validation pipeline to systematically compare the performance of the mentioned algorithms.

First, we performed a simple 3-fold Cross Validation to get an idea of the performance of the algorithms. In this step, we used the default settings of the classificators, meaning no hyperparameter tuning was performed. For the evaluation of the algorithms, we used the Mathews Correlation Coefficient (MCC).

Hyperparameter tuning is important in any machine learning model training activity. The hyperparameters of a model cannot be determined from the given datasets through the learning process. However, they are very crucial to control the learning process itself. The performance of a model on a dataset significantly depends on the proper tuning, i.e., finding the best combination of the model hyperparameters. Different techniques are available for hyperparameter optimization. In this study, we implemented the **Randomized Search**. We performed hyperparameters tuning using a Randomized Search Cross Validation, to get a better idea of the performance of the algorithms. For training and model selection we used the F1 score.

Finally, we performed a **nested Cross Validation**, using a K=5 fold for the outer loop and a L=3 folds for the inner loop. Nested cross-validation is an approach to model hyperparameter optimization and model selection that attempts to overcome the problem of overfitting the training dataset. The procedure involves treating model hyperparameter optimization as part of the model itself and evaluating it within the broader k-fold cross-validation procedure for evaluating models for comparison and selection. For the evaluation of the algorithms (outer loop), we used the Mathews Correlation Coefficient (MCC). For training and model selection (inner loop) we used the F1 score.

After finding the winner algorithm using nested CV, we used the whole dataset and a simple 5-fold Cross Validation to determine the *final model* that will be deployed in th field. To find the optimal set of hyperparameters for that model, we defined a grid and performed a Randomized Search. For training and model selection we used the Mathews Correlation Coefficient (MCC).

* 1. Results and Discussion

In order to get an idea of the models performance, we implemented a simple 3-fold Cross Validation on the whole dataset.

* 1. Conclusion

References