HW1 Report
Daniel Shemesh 211388251 & Shir Turgeman 315047324

Link to repository: https://github.com/DanielShemesh/hw1_094295

Executive summary

The project aims to predict whether a patient in intensive care suffers from sepsis, a life-threatening condition caused by an infection, six hours before he/she is diagnosed, based on clinical data about their medical condition over time. The project a dataset of TSV files that represent patients, each containing a column named SepsisLabel. The value in this column is 1 if the row is up to six hours before the identification of sepsis and 0 otherwise. If the patient did not suffer from sepsis at all, the column contains only zeros. A patient is labeled as having sepsis if there is a row where SepsisLabel=1. The project also processes the tables so that the input to the prediction model does not contain rows that contain data after the first row where SepsisLabel=1 and does not contain the column SepsisLabel.

We used four different machine learning algorithms to train and test classification models on this patients' dataset. We optimized the hyperparameters of each algorithm and evaluated the performance of each model using the F1 score. The best model was XGBClassifier with an F1 score of 0.75 on the test set. We also analyzed the importance of different features and the limitations of our approach.

Exploratory Data Analysis

a. The dataset consists of data collected from patients. The data is stored in pipe-separated value (PSV) files, one file per patient. Each file contains a header row with the names of 41 features, followed by one or more rows of data. Each row represents an hourly interval of a patient's stay in the ICU. The features include vital signs, laboratory tests, and demographics. The last feature is SepsisLabel, which indicates whether the patient had sepsis at that hour (1) or not (0). Sepsis is a life-threatening condition that occurs when the body's response to infection causes tissue damage, organ failure, or death.

As we mentioned earlier, the goal of the analysis is to identify patients who are at risk of developing sepsis and provide early intervention. Therefore, the SepsisLabel feature is the target variable for the predictive modeling task. The dataset is imbalanced, as only a minority of the rows have SepsisLabel equal to 1.

c. There are different methods to handle missing data, such as deleting, imputing, or ignoring them. The choice of method depends on the type and pattern of missingness, the amount and distribution of missing data, and the objective of the analysis.

In this case, we have used the forward fill method to impute missing values in each patient file. This method replaces missing values with the last observed value in the same column. This method assumes that the data is ordered by time and that the missing values are random or intermittent.

Feature Engineering

We decided to use all 40 features except for SepsisLabel as input for our prediction model, as we assumed that they all have some predictive power for sepsis risk.

Prediction

We used four different machine learning algorithms to build prediction models: Logistic Regression (LR), Random Forest Classifier (RFC), Support Vector Classifier (SVC), and XGBoost Classifier (XGB). We used Optuna to optimize the hyperparameters of each algorithm by maximizing the F1 score.

Hyperparameter selection, regularization: We used the Optuna library to tune the hyperparameters by maximizing the F1 score on the test set. We also used regularization techniques such as L1 & L2 penalty and early stopping.

We trained each algorithm on the train set and evaluated it on the test set using the F1 metric, with the goal of not only finding the best parameters, but the best algorithm out of the four.

After comparing the four algorithms, we concluded that the best one is XGBoost and it also converged the fastest (less parameter tuning steps).

Summary & Discussion

In this project, we aimed to predict sepsis risk in intensive care patients using machine learning models. We used a dataset of clinical data and performed exploratory data analysis, feature engineering, and prediction tasks. We used four different machine learning algorithms and optimized their hyperparameters. We evaluated their performance using F1 score on a test set. We found that XGB was the best model with an F1 score of 0.75 on the test set.