COSC 4557/5557 Practical Machine Learning Spring 2024
Auto Sklearn Fail
Submitted by: Iqbal Khatoon

Introduction

This report investigates the application of Auto-sklearn to a specific task—predicting the quality of red wine based on its physicochemical properties. Auto-sklearn was run on a dataset for 5 minutes, and its performance was compared with a default parameter Random Forest Classifier from scikit-learn. The dataset used is from OpenML (ID: 40691).

Wine Quality Dataset (Red)

This exercise employs the "winequality-red" dataset, containing information on different attributes of red wines. These attributes encompass fixed acidity, volatile acidity, citric acid, residual sugar, chlorides, free sulfur dioxide, total sulfur dioxide, density, pH, sulphates, and alcohol content. The dataset consists of eleven features. The target variable, denoted as "quality," assesses the wine's quality on a scale ranging from zero to ten (0-10). Based on our data analysis on the provided data set, we can ascertain that the dataset contains a total of 1599 entries, each with non-null values across all features and the label. This implies that there are no missing values present in the dataset, which is a positive aspect for our analysis. Furthermore, upon inspecting the data types assigned to each column, we observe that all features have been appropriately assigned the 'float64' data type, indicating numerical values. Similarly, the label class 'quality' comprises integer values exclusively, aligning with its assigned 'integer' data type.

Initial Results

When we ran AutoSklean on a wine quality dataset for 5 minutes, its performance with a default parameter Random Forest Classifier from scikit-learn and with Autosklearn is as follows:

- Random Forest Accuracy: 0.67
- Auto-sklearn Accuracy (initial run): 0.65

Observations

The initial run of Auto-sklearn resulted in a slightly lower accuracy than a simple Random Forest Classifier. This unexpected result prompts an investigation into potential reasons and possible improvements.
We analyzed few issues as mentioned below:

- Short Training Time: Auto-sklearn's initial setting allowed only 300 seconds (5 minutes) for training. This constraint may be too restrictive, giving Auto-sklearn insufficient time to thoroughly explore the model space and tune hyperparameters.
- Data Preprocessing and Encoding: The dataset contains categorical data that Auto-sklearn attempts to fit directly, leading to warnings about potential data type preservation issues. This could affect the learning process, as improper handling of categorical data might lead to inefficient model training.
- Default Resampling Strategy: The default resampling strategy might not be the most efficient for this particular dataset, potentially leading to suboptimal model performance.

Experiments and Adjustments

Several modifications were tested to improve Auto-sklearn's performance:

- Increased Training Time: Doubling the training time to 600 seconds did not yield a significant improvement. The accuracy remained approximately the same, indicating that simply increasing the time without adjusting other parameters may not be effective.
- Changing Resampling Strategy to Cross-Validation: Implementing a 5-fold cross-validation strategy improved the performance slightly, achieving an accuracy of 0.68. This suggests that a more robust validation strategy helps in achieving a more generalizable model.
- Ensemble Size and Configuration Tuning: Modifying the ensemble size and other configurations provided no significant improvements, with accuracies hovering around 0.655 to 0.68. This suggests that ensemble adjustments alone are not sufficient to substantially enhance performance.
- Utilization of All CPU Cores: Setting n_jobs = -1 to utilize all available CPU cores did not result in accuracy improvements, indicating that computational power was not a limiting factor in this scenario.