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, Experiments and Adjustments
Data preprocessing
Initially, the quality of the data will be assessed. Given that the goal is to categorize wine quality using feature columns, the analysis will primarily concentrate on three key aspects:

1. Missing values
2. Skewness in data distribution
3. Imbalance in data

Issues found with the dataset:
The dataset does not have missing values. However, the dataset has two main issues:

1. skewed explanatory variables (features) with outliers
2. imbalanced class

Dataset Balancing and Preprocessing Strategy

The dataset is characterized by a significant class imbalance, with minority classes 3, 4, and 8 having ratios of less than 0.05 compared to the majority class. To address this, I employed the Synthetic Minority Oversampling Technique (SMOTE) to augment the minority classes, coupled with undersampling using Tomek links. The goal of oversampling is to achieve a balanced dataset, while undersampling aims to mitigate the influence of outliers on the overall data.

Preprocessing with Scalers

Analysis of the various scalers showed that although the power transformer was most effective in managing outliers, it did not preserve the variance of some categories adequately. Consequently, I opted for the robust scaler for its efficacy in preprocessing. Nevertheless, to fully leverage the capabilities of autosklearn, I decided against applying any scaler during the autosklearn training phase, given autosklearn's robust internal data preprocessing mechanisms.

Addressing Imbalance in Class Distribution

To tackle the issue of imbalanced classes, I initially oversampled the minority classes to achieve a ratio of approximately 1:3 with the majority class. This ratio was chosen to minimize the introduction of noise and bias that could stem from excessive oversampling, particularly given the scant number of samples in the minority classes. This approach is designed to enhance the model's long-term generalizability. Following the oversampling, I applied undersampling to further align the majority and minority classes and reduce the impact of outliers. This dual approach utilized SMOTE in conjunction with Tomek links (SMOTETomek), striking a balance between class equality and data integrity.

Results and Discussion

Our evaluations reveal that the autosklearn model slightly outperforms the standard RandomForest (RF) model. Extensive testing with the autosklearn package was undertaken to understand the underlying reasons. Primarily, a resampling strategy was integrated, and a 5-fold cross-validation was implemented to enhance model generalizability and reduce the likelihood of overfitting. The performance of the autosklearn ensemble models suggests that tree-based methods are highly effective, which also accounts for the strong baseline performance of the RF model.

Furthermore, an oversampling method was tested, which equalized the number of observations across minority and majority classes. Although this approach yielded high accuracy in test datasets, concerns about its reliability led to a decision to apply oversampling at reduced rates to improve generalizability.

In summary, two major challenges were addressed: 1) skewed features with outliers were managed using autosklearn's internal data preprocessing tools, and 2) class imbalance issues were tackled using a combination of over- and undersampling techniques, specifically with SMOTETomek. These measures have collectively enhanced the autosklearn's performance beyond that of the conventional RF model.

Final accuracies

RF Accuracy nwithout scaler = 0.7893617021276595
RF Accuracy after applying scaler = 0.7957446808510639
AutoML Accuracy 0.8063829787234043

Reference:

[1] https://www.kdnuggets.com/2017/06/7-techniques-handle-imbalanced-data.html