Practical Machine Learning: AutoSklearn Fail

Russell Todd[1]

[1]University of Wyoming

1 Introduction

In this exercise, I will be looking at an application of AutoSklearn on the Red Wine dataset. When run, the AutoSklearn model scores lower than a default RandomForestClassifier (RFC). I will investigate what the problem is and how it can be corrected such that the AutoSklearn model will have its performance improved above the default RFC.

2 Problem Exploration

The famous Red Wine Quality dataset is comprised of 12 features. The first 11 are measurements of various physical qualities of each wine (fixed acidity, citric acid, residual sugar, pH, alcohol, etc.) and the 12th is the output variable which is a score (0-10) denoting the quality of each wine. There are 1,599 entries (wines) in the dataset and no entries are missing data for any feature. All features are positive numeric values.

The Red Wine Quality dataset is quite imbalanced with relatively few entries for low quality and high quality wines as can be seen in figure 1 below. I believe that the imbalance of the dataset is what is stopping AutoSklearn from performing well.

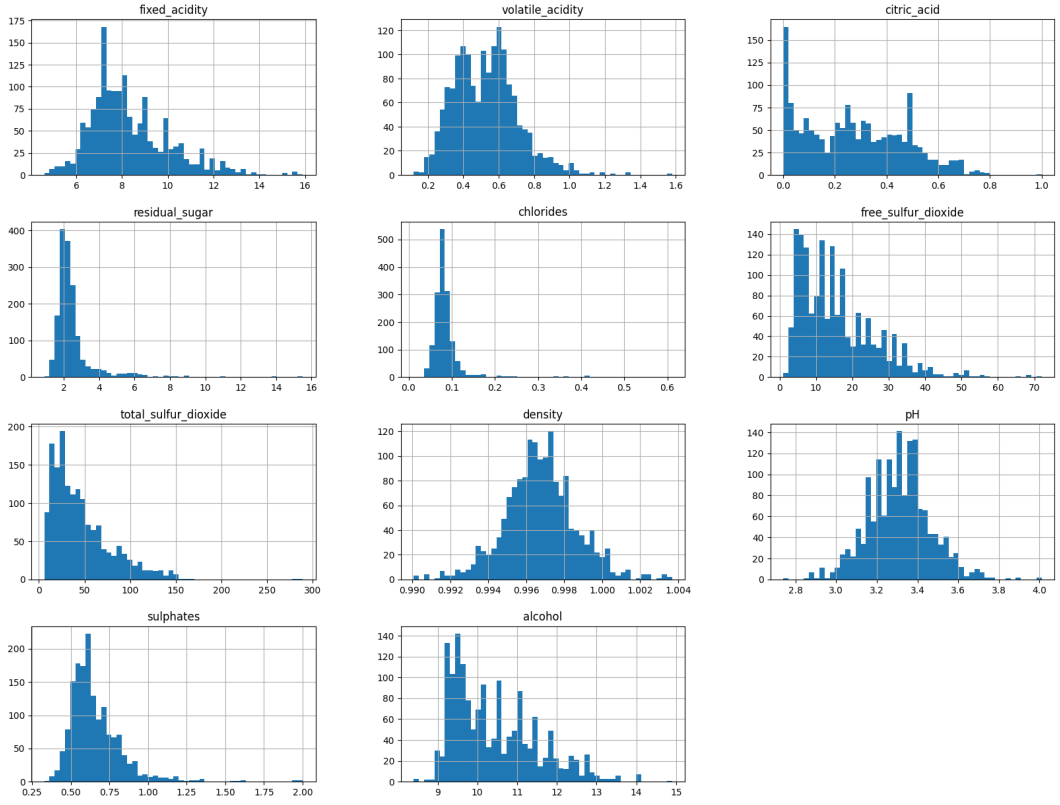

Figure 1: Unbalanced Red Wine Frequency Plot

3 Solution Attempt

Since my hypothesis is that the imbalance is the main issue, I will try to correct that by using SMOTE (Synthetic Minority Over-sampling Technique) to generate more instances of those low and high quality wines in the dataset. In figure 2 you can see how the frequency plot has changed after SMOTE was applied to the dataset. The general shape of the distribution is very similar to the original, but is smoothed a bit.

Additionally, I attempted to find a solution that precluded data preprocessing to see if the built-in options provided within the autoML process could achieve similar results. Therefore, I also did an iteration of the autoML where I utilized the metrics.balanced_accuracy_score from auto-sklearn as the scoring metric while also utilizing the cross validation resampling method with shuffling and 5 folds. As part of this, nested resampling was done as part of the cv resampling process. This iteration was run for a full hour, whereas the previous attempt was run for 5 minutes. In the results I refer to this run as AutoML2.

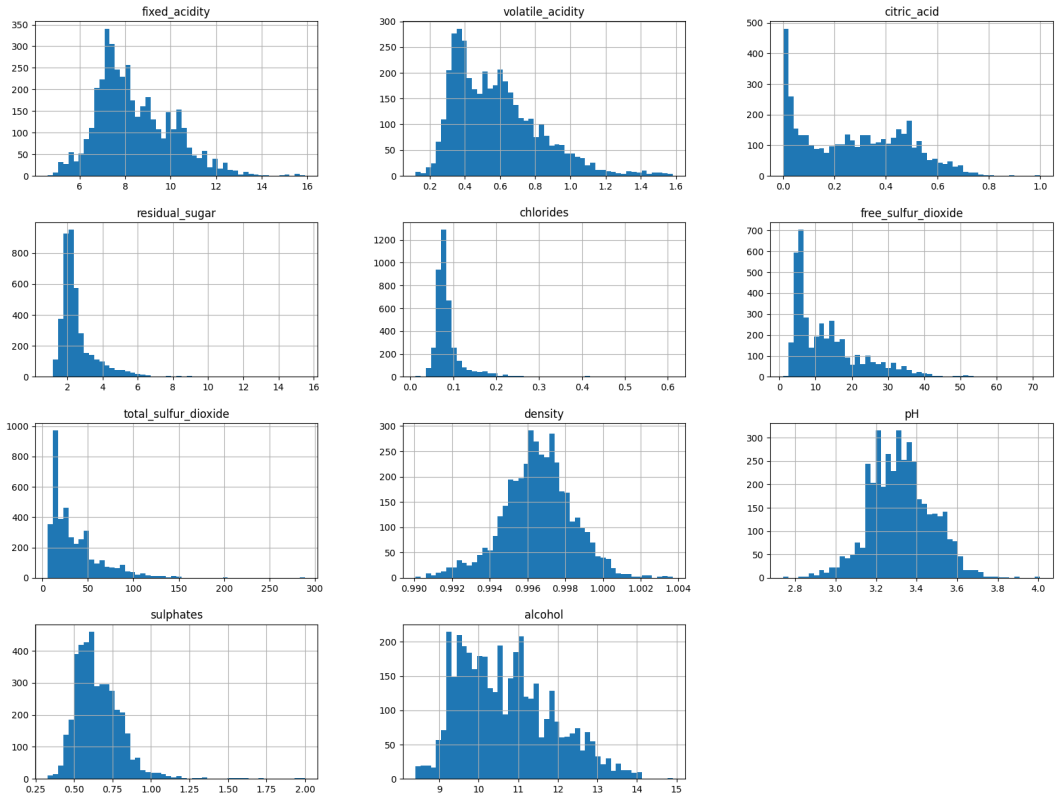

Figure 2: Frequency Plot after SMOTE

4 Solution Results

The solution was successful in raising the performance of the autoSklearn model as can be seen in the accuracy scores shown below:

However, the AutoML2 run did not succeed in outperforming the RandomForest so I ran it again but with the usual accuracy_score metric and named it AutoML3. This run did outperform the RandomForest but not as dramatically as when SMOTE was used as preprocessing.

```
RF Accuracy: 0.67
AutoML Accuracy (original): 0.66
```

```
AutoML Accuracy: 0.8287671232876712
AutoML2 Accuracy: 0.63
AutoML3 Accuracy: 0.675
```

The AutoSklearn model now shows a significant improvement both over its previous score and the score achieved by the default Random Forest Classifier, but does necessitate preprocessing decided upon by a human user.