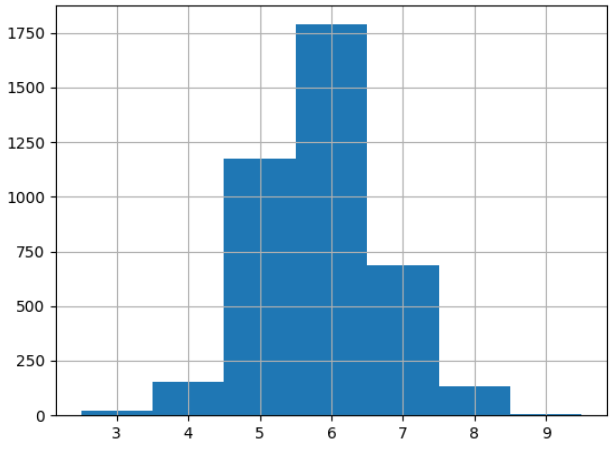
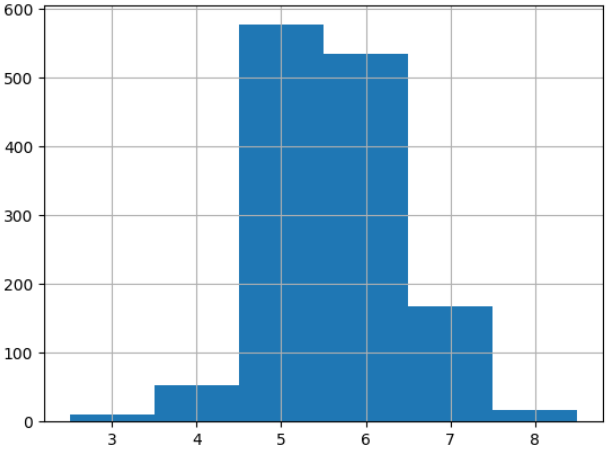
Problem

The goal of this project is to produce a web app which estimates the quality of a "Vinho Verde" wine with a given set of physiochemical properties. The user will have the ability to set the values of these properties as well as choose whether they want a red or white wine. The outcome should be predicted as a number on a 1-10 scale (this maybe be a discrete integer scale or continuous one depending on which provides the better user experience).

Data

I have continued to work with the dataset from the University of California Irvine Machine Learning Repository which was originally collected by Dr. Paulo Cortez at the University of Minho. It must be noted that my data is heavily imbalanced and is not ideal, as can be seen in the histograms below (white left, red right). Specifically note how the extremes have nearly no data points.





In pre-processing, I remove all duplicate rows (this has already been done in the above graphs), reducing the white dataset to 3961 entries and the red dataset to 1359. I am currently using a 70/30 split for creating the train and test sets.

I am currently trying to use all available features as the discussion for the dataset suggests that the predictions are far more reliant on the relationship of the properties to each other rather than being related to any single variable. To determine whether all the features are affecting the predictions, we can look at feature importance arrays.

With the following feature list:

['fixed acidity', 'volatile acidity', 'citric acid', 'residual sugar', 'chlorides', 'free sulfur dioxide', 'total sulfur dioxide', 'density', 'pH', 'sulphates', 'alcohol']

We see the following feature importance values for

White:

[0.0652104,1 0.12328078, 0.05705787, 0.06318535, 0.05997773, 0.11850914, 0.06493333, 0.05437223, 0.07234969, 0.05450575, 0.26661772]

and

Red:

[0.05071501, 0.12751734, 0.04407684, 0.06257668, 0.06221539, 0.04630209, 0.07988127, 0.05199071, 0.06001097, 0.13466593, 0.28004776]

Based on these values, certain properties like ‘citric acid’ and ‘chlorides’ could be excluded as they have a relatively small importance, however in my tests the removal simply reduced accuracy and so I saw no solid reason for their exclusion.

Model

It seems possible that this model is not the best choice for this dataset. As will be discussed in the results, the predictions include very few extreme values; while the large class imbalance present in my dataset is almost certainly to blame, it is also likely that the robustness to outliers which random forests has[1] is reducing the effect of the few extreme training entries on the model. It is also highly likely that I will continue to battle for extreme values as I implement this data into my final product, as the extrapolative abilities of random forest models are fairly limited due to the nature of the decision trees they are based on.

For these initial tests, I relied heavily on scikit-learn. For the completed project, I hope to implement my own random forest regressor, however I chose to use a library implementation for the time being in order to produce short-term, proof of concept results. This approach also provided me with some easily accessible hyper-parameters which I could use to tune the accuracy of the model. The effects of these changes, however, were relatively limited. Through some trial and error I found I got the my best accuracy when I set n_estimators=500 and min_samples_split=5, but these optimizations only really increased the accuracy by 2% or 3%. I tuned the these hyper-parameters for both red and white wine models simultaneously, for the final project I would likely want to separate these tunings to achieve the best possible results on each model.

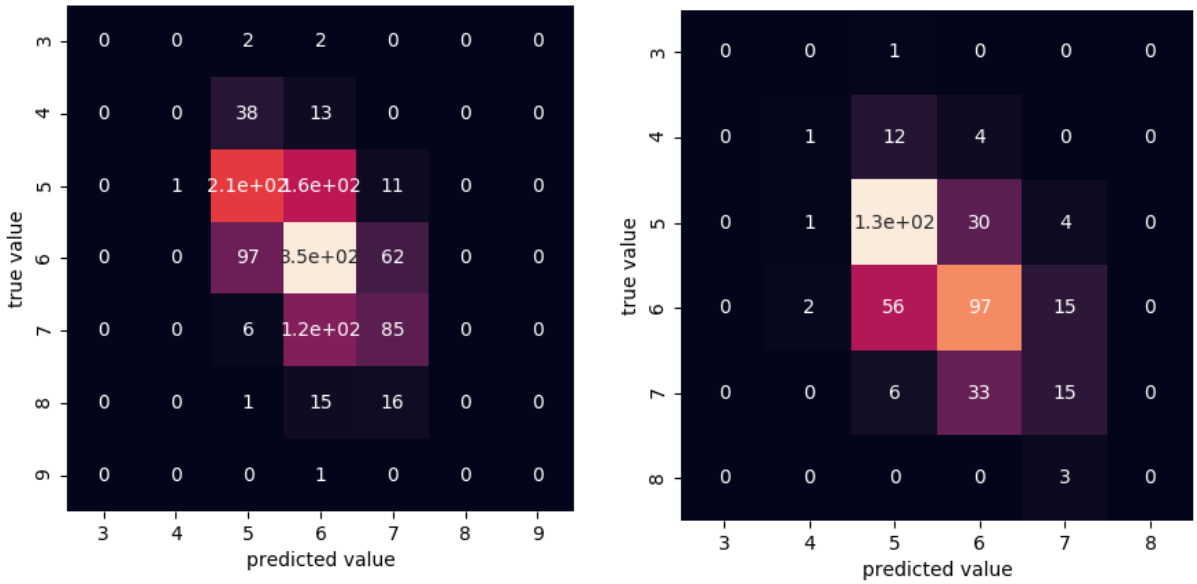
Results

Initial results aren’t particularly promising. The training accuracy is quite high (~92% for white set, ~91% for), but the test accuracy is relatively low (~54% and ~64% for white and red respectively). I figured my model might have been overfitting, however as I altered hyper-parameters the accuracies tended to increase/decrease to approximately the same degree. I used f1 scores as a metric of precision, these followed a similar trend to the accuracy where the training values were high (~93 for both), while the test values were lower (57/65). A note which must considered along with the data that has just been presented is that all predicted values are rounded to the nearest whole number as the regressors returns continuous outputs.

---

[1] D. (2016, December 30). Dimensionless’s Answer to “Why are tree-based models robust to outliers?”. Quora. Retrieved February 21, 2019, from https://www.quora.com/Why-are-tree-based-models-robust-to-outliers/answer/Dimensionless-1

A brief look at the confusion matrices (left is white, right is red) reveals what the model is learning from this data. Due to the great imbalance in the data, it is a pretty safe bet that most of the wines will be a 5 or 6, as such the model predicts almost exclusively these values



It seems the model is not entirely useless, however it struggles to predict anything beyond average-rated wines; it is entirely possible that this will create a final product that has very limited capabilities.

Next steps

I must consider whether or not I'm going to have to take an entirely different approach to this project. My current method is obviously not producing particularly exciting results, and there seems to be little to nothing I can do about the issues in my dataset. It is possible I would end up with a more useable tool if I were change my targets to a pair or short series of classes giving more general rankings to the wine, however in terms of a final product this would be far less interesting. Might it even be prudent to search for an entirely new dataset and start from scratch? I am unsure as to what my best option and so it is here that I ask for some guidance from the MAIS execs.