Machine Learning Engineer Nanodegree

Capstone Report

Al Piran

May 31, 2018

I. Definition

Project Overview

Being a wine enthusiast I've always been curious as to what makes one wine better than the other. Is it just a matter of opinion or are there characteristics that are common between wines that are considered to be high quality. Making wine is both an art and science. The process of going from grapes to an actual bottle of wine is complex and involves many variables that affect the quality of the wine. Those variables can be measured and tweaked, but the ultimate judge of quality is the human palette. This projects intends to use machine learning and the Wine Quality Dataset from UCI in order to show what objective factors in the wine making process have the biggest effect on the subjective quality of wine.

There are multiple studies that have analyzed the same dataset, 2 of which are cited below.

- Cortez(2008): https://pdfs.semanticscholar.org/bebb/83a340b77917acc0a59d55ddab066e9c1acf.pdf
- Er(2016): http://www.asosindex.com/cache/articles/the-classification-of-white-wine-and-red-wine-according-to-their-physicochemical-qualities-f169542.pdf

Cortez uses 3 regression techniques: multiple regression(MR), multilayered perceptron neural network(NN) and support vector machines (SVM), for his analysis, and an accuracy of 62% using the SVM. Er on the other hand uses 3 classification techniques: SVM, k-nearest neighbours and Random Forest(RF), and achieves 71% precision using RF.

Problem Statement

The subjective quality assessment that wine experts perform has a lot riding on it. A rating above seven on a scale of 10 in a magazine such as wine spectator can have a big effect on the price and appeal of the product. But can this assessment be understood in terms of several measurable objective variables? If so to what extent can this be done? and of the 11 variables that we have in this dataset which are the 2 or 3 most important ones for being able to predict the subjective quality assessment.

Hence we want to predict the subjective quality rating based on the objective chemical features of a wine. This will be done by applying 2 machine learning models to the data and evaluating each using metrics that are described below. The chosen models are Linear Regression and Random Forest. Linear Regression will be used as a benchmark since it's the simplest and fastest model and a good baseline to compare further results. Random Forest on the other hand is a more sophisticated ensemble model that is expected to produce a better result. Both models are described in more detail below in the Analysis section.

Metrics

R2 score is the most common metric used for evaluating regression models and will be used to evaluate both the benchmark and the solution model. R2 score explains how well the independent variables, in our case the measurable chemical features explain the variability in the dependent variable, the quality rating. This is a value between 0 and 1 for not explaining at all and fully explaining respectively. The mean squared error which is the other useful metric for regression problems will also use used as a metric for confirmation. Hence if both the R2 is higher and the mean squared error gets lower, it can be ascertained that the model is better.

R2 is defined as follows:
R2 = (Explained Variarion)/ (Total Variation)
= SSR / SST

$$= \sum(f_i – \text{mean}(y)) / \sum(y_{i\,–}\, \text{mean}(y))$$

where:
SSR: Sum of Squares Regression,
SST: Sum of Squares Total,
f_i : the predicted value
y_i: the data value

II. Analysis

Data Exploration

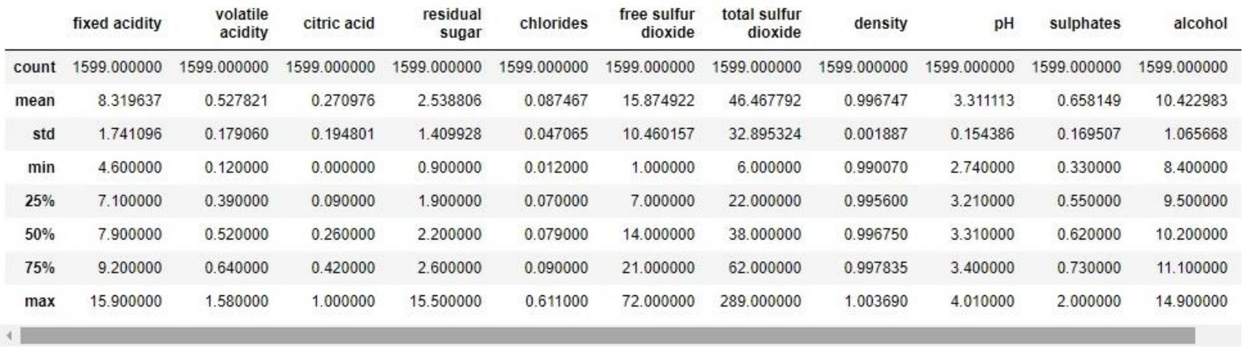
The UCI red wine dataset will be used which is hosted below at UMass.

http://mlr.cs.umass.edu/ml/machine-learning-databases/wine-quality/winequality-red.csv

The following are the features of red wine available in the above dataset:

0 - fixed acidity: acids that not evaporate readily
1 - volatile acidity: the amount of acetic acid in wine, if high can give wine an unpleasant, vinegary taste.
2 - citric acid: adds ‘freshness’ and flavor to wines
3 - residual sugar: the amount of sugar remaining after fermentation stops, usually between 1 and 9 g/liter for red wines
4 - chlorides: one of the main minerals (salts) present in wine
5 - free sulfur dioxide: used as a preservative because of its anti-oxidative and anti-microbial properties in **wine**, but also as a cleaning agent for barrels and winery facilities
6 - total sulfur dioxide: amount of free and bound forms of S02
7 - density: the density of wine (g/cc) is close to 1, the density of water
8 - pH: most red wines have a pH of 3.3 to 3.6.
9 - sulphates: a wine additive which can contribute to sulfur dioxide gas (S02) levels,
10 - alcohol: the percent alcohol content of the wine
11 – quality: output variable subjective evaluation (0-10)

Table 1: Basic statistics about the data in tabular and graphical form:



	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcohol
count	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000
mean	8.319637	0.527821	0.270976	2.538806	0.087467	15.874922	46.467792	0.996747	3.311113	0.658149	10.422983
std	1.741096	0.179060	0.194801	1.409928	0.047065	10.460157	32.895324	0.001887	0.154386	0.169507	1.065668
min	4.600000	0.120000	0.000000	0.900000	0.012000	1.000000	6.000000	0.990070	2.740000	0.330000	8.400000
25%	7.100000	0.390000	0.090000	1.900000	0.070000	7.000000	22.000000	0.995600	3.210000	0.550000	9.500000
50%	7.900000	0.520000	0.260000	2.200000	0.079000	14.000000	38.000000	0.996750	3.310000	0.620000	10.200000
75%	9.200000	0.640000	0.420000	2.600000	0.090000	21.000000	62.000000	0.997835	3.400000	0.730000	11.100000
max	15.900000	1.580000	1.000000	15.500000	0.611000	72.000000	289.000000	1.003690	4.010000	2.000000	14.900000

Looking at Table 1, we can see that the min/max values of the different features have very different ranges. Hence some preprocessing will be needed before machine learning takes place. The details of preprocessing will be explained in the preprocessing section below.

Algorithms and Techniques

The 2 ML algorithms that will be used in this study are: Linear Regression and Random Forest.

Linear Regression: Linear regression consists of finding the best-fitting straight line through the data points. It is perhaps the simplest method for finding a relationship between 2 variables.

The best fitting line has the equation: Y = bX + a,

where:
$$a = \frac{(\sum y)(\sum x^2) - (\sum x)(\sum x y)}{n(\sum x^2) - (\sum x)^2}$$
$$b = \frac{n(\sum x y) - (\sum x)(\sum y)}{n(\sum x^2) - (\sum x)^2}$$

In Python Linear Regression is used in the following way:

```
from sklearn.linear_model import LinearRegression
```

Random Forest: This is an ensemble method as mentioned. Ensembles are divide-and-conquer algorithms used to improve performance. The main principle behind ensemble methods is that a group of "weak learners" can come together to form a "strong learner". Random Forest is an ensemble method based on decision trees and tries to solve the problem of overfitting in deep decision trees. It does so by classifying decision trees on many sub-samples of the dataset and finds the average result from those many decision trees. While the predictions of a single tree are highly sensitive to noise in its training set, the average of many trees is not, as long as the trees are not correlated.

In Python Random Forest Regressor is used in the following way:

```
from sklearn.ensemble import RandomForestRegressor
```

Benchmark

Linear Regression was chosen as a benchmark because it is simple and intuitive. This benchmark was evaluated using the metrics R2 score and mean squared error and served as a baseline of performance for the rest of the project.

Table 2: Benchmark Results

Benchmark Model	**R2 Score**	**Mean Squared Error**
Linear Regression	0.41	0.46

III. Methodology

Data Preprocessing

As can be seen in both the Table 1 data and the box plots from Figs 1 and 2 below, there are 2 issues with this data that need to be addressed before ML will be effective. First, the different features of the data have very different ranges of values, 0.27 to 1.00 for citric acid and 6 to 289 for total sulfur dioxide for example. Second, the data has outliers that need to be removed. For ML to be effective we want all features to have equal weight in terms of how much they influence the output variable in the model. Hence we'll have a 2 step data preprocessing using standardization and removing outliers to correct the 2 issues.

Step 1: Standardization

Standardization is the process of transforming data such that it has a mean of zero and standard deviation of one. In python we used the preprocessing.StandardScaler() on each feature to achieve this. As can be seen by comparing **Fig.1**, the raw data and **Fig.2**, standardized data, the former has a range of 0 to almost 300, while the latter goes from -4 to 12. The features are now much more equal in terms of their weight and influence.

Step 2: Outlier Removal

There are various ways to detect outliers. The method used here was with IQR or Inter Quartile Range. IQR is the range where the middle 50% of the data lie, ie between the 25th and 75th percentiles. The IQR method considers any data point more than 1.5 IQRs below the first quartile or above the third quartile of the data to be an outlier.

The full dataset contains 1599 data points, after removal of the outliers we have 1179 data points, hence 420 data points (1599-1179) were outliers. We can see in **Fig.3** that once outliers were removed, the range of data goes from about -3 to 3, hence the features are more uniform and the data ready for the actual ML step.

Fig. 1: Box plot of raw data

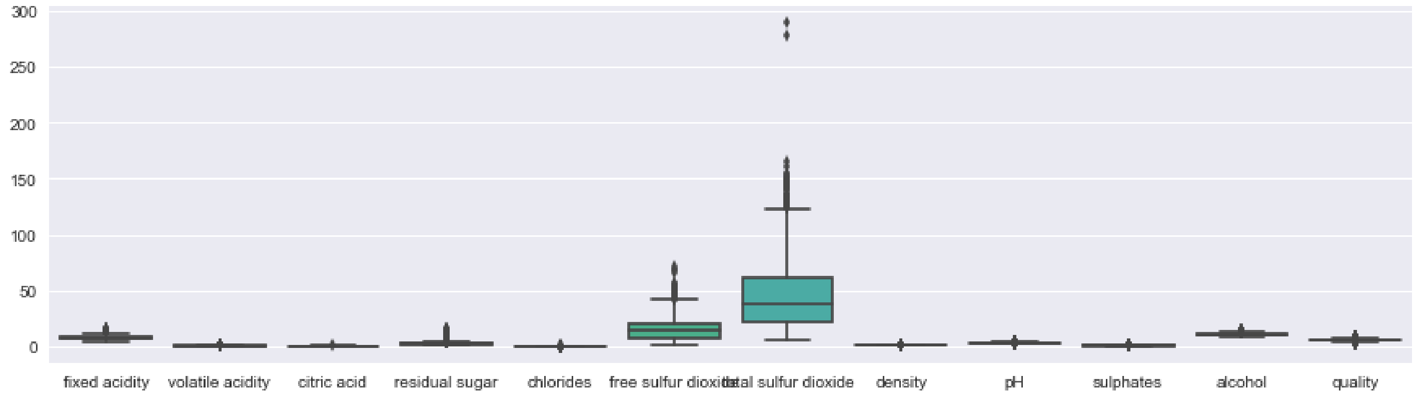


Fig 2: Box plot of standardized data

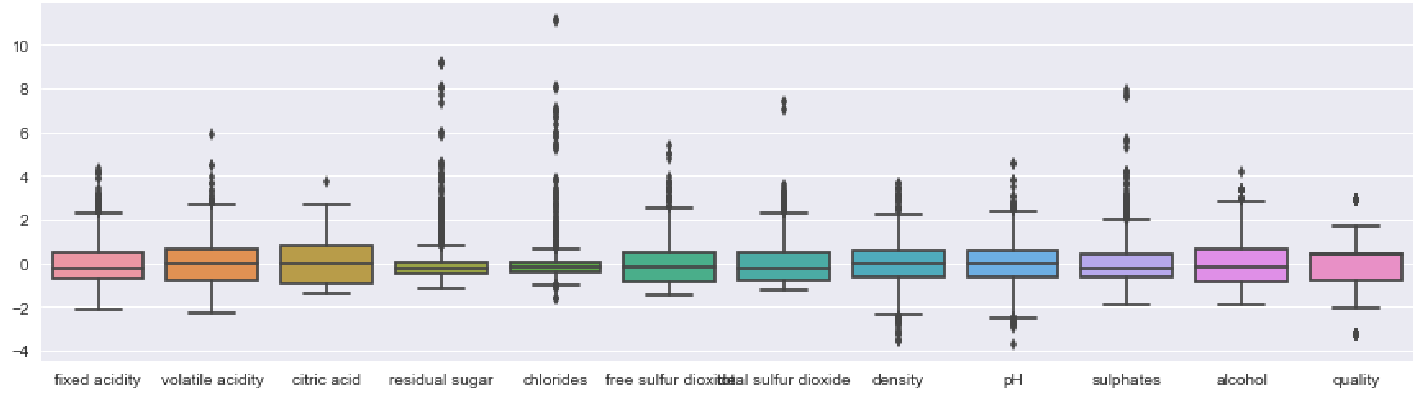
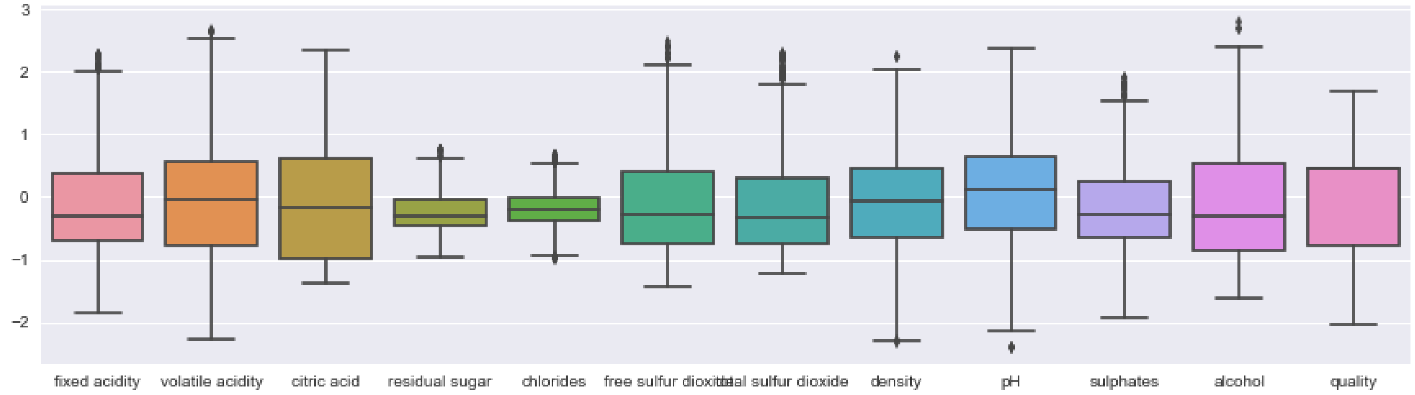


Fig 3: Box plot of standardized data after removing outliers



Implementation

1. The dataset was loaded from the following url (http://mlr.cs.umass.edu/ml/machine-learning-databases/wine-quality/winequality-red.csv). The dataset can be found as a csv file which can be downloaded, however using a url is more robust since it won't require the use of an extra data file that has to be available for the code to work.

2. The data was found to have varying ranges among the different features, hence standardization was used to prepare the data for ML. Standardization was chosen as opposed to normalization, since in general it performs better for regression ML algorithms.
3. After standardization the data was found to have outliers among most features. Hence the IQR method, explained above, was applied to remove outliers from all features.
4. The Benchmark, Linear Regression was run to establish a baseline of comparison for the solution algorithm.
5. The solution model Random Forest Regressor was run. The hyperparameter n_estimators which is the number of trees in the forest was varied from 25 to 100 and 200 with a very small improvement in results. At n_estimators=100 an R2=0.52 was achieved and that was not exceeded with n-estimators=200.
6. The challenge with hyperparameters is that it can be difficult to know over what ranges to experiment with each hyperparameter inside GridSearch. Since results did not improve with the first attempt, a more thourgh GridSearch with more parameters was performed. With the following parameter grid:

```
param_grid = {
   'n_estimators': [int(x) for x in np.linspace(start = 50, stop = 450, num = 5)],
   'max_depth': [int(x) for x in np.linspace(10, 110, num = 11)],
   'min_samples_split': [2, 5, 10],
   'min_samples_leaf': [1, 2, 4],
   'bootstrap' : [True, False],
   'max_features': ['auto', 'sqrt', 'log2']
}
```

However the results didn't improve and remained at: R2=0.52 and error=0.38

7. Another challenge was finding the best way to measure the robustness of the solution algorithm. There does not appear to be a generally agreed upon method to determine robustness in the literature. After much research it was decided that the standard deviation of the test score given by the cv_results object of GridSearchCV is a good measure for this particular application.
8. Due to GridSearchCV being extremely slow on the computer that was used, RandomSearchCV was used in its place. RandomSearchCV gives similar results to GridSearchCV however it does so much faster. Moreover it has the same cv_results object, hence the same information about the mean and standard deviation of test scores was obtained.
9. The average of the 10 values for the mean and standard deviation of the test score from RandomSearchCV was calculated. This gave us an estimate of the robustness of the algorithm.
10. Feature Importance was used to find the most important features that are responsible for predicting the target.

Refinement

Hyperparameters were tuned in 2 steps. n_estimators, which is the number of trees in the forest is the most important of the parameters and was tuned first by starting with 25 and going up to 100 trees. Once the optimum number of trees was found, grid search was used in order to determine the optimum model. A 2nd more thorough Hyperparameter tuning was done as described in the above section with more parameters and wider parameter ranges.

IV. Results

Model Evaluation and Validation

The full code of how the results were achieved is in the wine.ipynb file, however the following are the main results. Note that all R2 score and Mean Squared Errors (MSE) are with respect to the Test Set of the data.

1. The Benchmark, Linear Regression method clearly showed that yes, there is a relationship between the subjective quality of a wine and the objective features of that wine that are examined in this study. (The specific features are listed above in the data exploration section.)
2. Linear Regression had a R2 score=0.41 and an MSE =0.46 . These are benchmark metrics that other models can be compared to.
3. The solution model Random Forest had an R2 score=0.52 and MSE =0.38 using the hyperparameters below. This is good news we have both a higher R2 and a lower MSE.

Hyperparameters producing the final result were as follows:
n_estimators: 350,
min_samples_split: 5
min_samples_leaf: 1
max_features: sqrt
max_depth: 80
bootstrap: False

4. The only hyperparameter that had a big effect on results was n_estimator=100 or higher. The other hyperparameters described above (#3) at default would result in the same R2 score=0.52.
5. The Average of the 10 mean test scores was: 0.42.
6. The Average of the 10 standard deviation of the mean test scores was: 0.07
7. Feature Importance showed that the 3 main features that help predict the target are: volatile acidity, sulphates and alcohol.

Table 3: Summary of results for Regression (R2 and MSE of test set):

Model	Parameters	R2 score	Mean Square Error
Linear Regression	none	0.41	0.46
Random Forest	n_estimators=25	0.51	0.38
Random Forest	n_estimators=100	0.52	0.38
Random Forest	n_estimators=200	0.52	0.38
Random Forest	n_estimators: 50 to 450 max_depth: 10 to 110 min_samples_split: 2, 5, 10 min_samples_leaf: 1, 2, 4 bootstrap: True, False, max_features: auto, sqrt, log2	0.52	0.38

Table 4: Summary of results for Feature Importance:

3 top features responsible for predicting the target:	Alcohol	Sulphates	Volatile Acidity

Justification

The question posed at the outset was the following: Is there a relation between the subjective quality score of a wine and the objective, measurable features available in this dataset? The Benchmark Linear Regression model already answered the question with a clear “Yes”. Further to that the Random Forest model improved on that by reaching an R2 score of 0.52 as opposed to 0.41 of the benchmark.

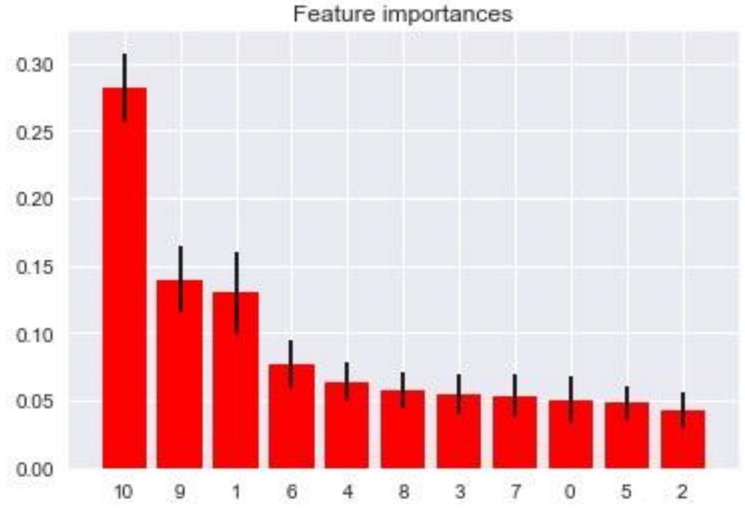
As far as the robustness of the above result, we have a mean test score of 0.42 with a standard deviation of 0.07 from the RandomSearchCV. This means that given different data 67% of the time the mean test score will be between 0.35 and 0.49. This is at best moderately robust and perhaps could be better. In my reference #9 below I indicate the sklearn page that suggests GridSearchCV and RandomSearchCV produce similar results, but RandomSearchCV is much faster. The speed advantage of RnadomSearch has been confirmed, however due to slow hardware whether GridSearch can produce better results that are more robust has not been confirmed.

The second question posed was: Which are the features most responsible for predicting the target? To answer this question Feature Importance was used and it showed that the 3 key features in order of most to least importance are: Alcohol, Sulphates and Volatile Acidity.

V. Conclusion

Free-Form Visualization

Fig. 4: Best Predictors of the target using Feature Importances



The feature numbers above correspond to the following once again:
0 - fixed acidity: acids that not evaporate readily
1 - volatile acidity: the amount of acetic acid in wine, if high can give wine an unpleasant, vinegary taste.
2 - citric acid: adds ‘freshness’ and flavor to wines
3 - residual sugar: the amount of sugar remaining after fermentation stops, usually between 1 and 9 g/liter for red wines
4 - chlorides: one of the main minerals (salts) present in wine
5 - free sulfur dioxide: used as a preservative because of its anti-oxidative and anti-microbial properties in wine, but also as a cleaning agent for barrels and winery facilities
6 - total sulfur dioxide: amount of free and bound forms of S02
7 - density: the density of wine (g/cc) is close to 1, the density of water
8 - pH: most red wines have a pH of 3.3 to 3.6.
9 - sulphates: a wine additive which can contribute to sulfur dioxide gas (S02) levels,
10 - alcohol: the percent alcohol content of the wine
11 – quality: output variable subjective evaluation (0-10)

Reflection and improvement

It has surely been ascertained in this study that the subjective quality rating for wines do have a basis in objective science. The measurable chemical features of wine in this dataset can predict the quality rating with an R2 score of 0.52 when a Random Forest Regressor is used. Moreover. 3 features in particular help most with this prediction those are: Volatile Acidity, Sulphats and Alcohol.

There were several challenges during this project. The original version of the project did not use any data preprocessing, however after some reflection and experimentation it was deemed useful and gave better results. How to preprocess was also not clear at the outset, but researching the given algorithms, standardization and outlier removal were chosen as the best course of action in this context. It was a surprise that tuning the many hyperparameters described above using RandomSearch did not result in a better R2 value, and as long as n_estimators was above 100, an R2 value of 0.52 would have been achieved.

Finally regarding the robustness of the result, there are several issues. One is that there doesn't seem to be any standard way to measure robustness based on my research. It seems highly dependent on the problem and since RandomSearch was used in this case, it seemed reasonable to use the standard deviation of the mean test score as a measure of robustness. The second issue is: what value for the above mentioned standard deviation is considered robust? Again this is a question that is highly dependent on the problem and its application. For the purpose of this project namely establishing a link between the subjective quality of a wine and its chemical components, it is most probably sufficient.

There are many ways in which this study could be improved:

1. Other red wine data sets might be available, and it would be interesting to see if the results from those would be any different.
2. This data set could have been segmented in a variety of ways. For example looking at wines with high alcohol content or low sulphate content only. Perhaps some of these segments may have higher predictive power than the data set as a whole
3. In this study only no neural nets were used, perhaps a neural net model could bring better results.

References

1. https://archive.ics.uci.edu/ml/datasets/wine
2. https://machinelearningmastery.com/metrics-evaluate-machine-learning-algorithms-python/
3. https://machinelearningmastery.com/feature-selection-machine-learning-python/
4. https://medium.com/click-bait/wine-quality-prediction-using-machine-learning-59c88a826789
5. https://winefolly-wpengine.netdna-ssl.com/wp-content/uploads/2016/04/wine-sweetness-chart-wine-folly1.jpg#large
6. https://winobrothers.com/2011/10/11/sulfur-dioxide-so2-in-wine/
7. https://towardsdatascience.com/hyperparameter-tuning-the-random-forest-in-python-using-scikit-learn-28d2aa77dd74
8. http://scikit-learn.org/stable/auto_examples/ensemble/plot_forest_importances.html
9. http://scikit-learn.org/stable/auto_examples/model_selection/plot_randomized_search.html