Predicting wine quality from its chemical properties

Marcin Kolaszewski

Data Science Initiative, Brown University

github.com/marklin36/1030-final-project.git

December 2019

1 Introduction

This project aims to develop a model that predicts white wine quality based on its chemical properties. Hence, the target variable is wine quality.

The problem can be viewed as both regression or classification depending whether we view the wine quality as a continuous or discrete feature. We will approach the problem as a classification task because usually people use a discrete scale to describe how much they like something.

The problem is interesting because it attempts to explore the objective characteristics of wine that people enjoy. If it turns out that perceived wine quality can be predicted from its chemical properties, the results might be impactful for wine producers (who will gain some insight how to produce better wines) and also for consumers (wishing to maximize their wine satisfaction). On the other hand, if it is the case that people's wine preference are independent from it's chemical properties then it would be an interesting result from a psychological perspective.

The dataset used in this project comes from the UCL Machine Learning Repository. It contains 4898 data points and 11 features and it is rather well documented, please see the winequaility.names file for the original data set description. All features are numerical but they lack units of measurement. From the ranges of these feature we can infer the following units:

Feature	Units
fixed acidity	standard ph units
volatile acidity	g/L
citric acid	g/L
residual sugar	g/L
chlorides	g/L
free sulfur dioxide	ppm
total sulfur dioxide	ppm
density	g/L
sulphates	g/L
pH	stand ph units
alcohol	% by volume
quality	1-12 scale, 12 is the best

Table 1: Features and corresponding units

This data set has been used in multiple public studies including the following three:

- Modeling wine preferences by data mining from physicochemical properties [1] – Academic study that used regression and support vector machine algorithms to predict the wine quality from the given features. The authors reported that the use of SVM outperformed regression and neural networks.
- Project Report:-Red Wine Quality Analysis Final [2] – This study uses analogical data set for red wine with the same features and comparable sample size. The author's explanatory data analysis employs regression on some engineered features.
- Exploratory Data Analysis on Red Wine Quality [3] – This study is also uses the red wine data set from the example above. The author's data analysis focuses on use of statistical techniques, and it does not use regression or SVMs in order to predict the wine quality.

1.1 Preprocessing

Since the data was already in tidy format, it had only numerical features and no missing values we only needed to apply a suitable scaling to each feature. Since all features have rather small and predictable range, we applied min-max scaler to all of them.

As we can see in the figure 1, the data set is highly imbalanced, it contains mostly average wines (labels equal to 5, 6 and 7). For example, members of classes $1, 2, 10, 11, 12$ were not present in the data set and members of classes 3 and 9 made less than 1% of the data set. That is why, we decided to group the labels into the following 5 classes:

- Class 1: label less than 5
- Class 2: label equal to 5
- Class 3: label equal to 6
- Class 4: label equal to 7
- Class 5: label greater than 7

So with the redefined classes, label 1 corresponds to the wine of the worst quality and label 5 corresponds to the wine of best quality.

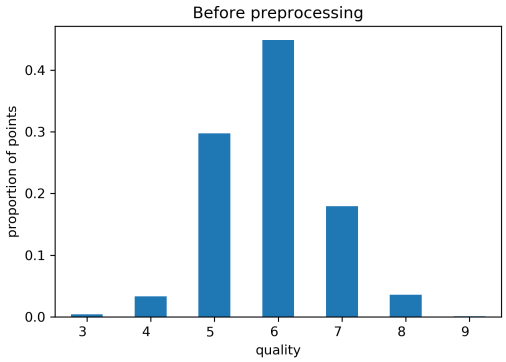


Figure 1: Dataset balance before preprocessing

Hence we achieved more balanced data set in which each class has balance greater than 3%.

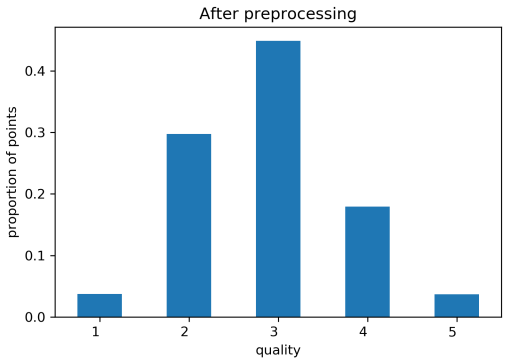
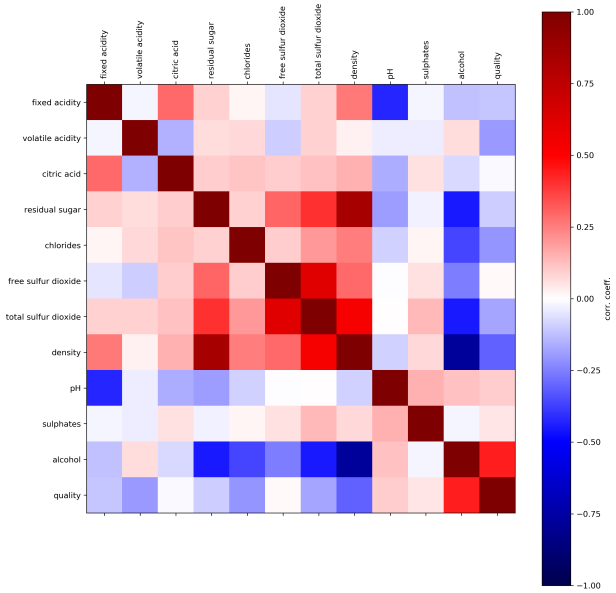


Figure 2: Dataset balance after preprocessing

2 EDA

In order to identify the most predictive features, we computed the correlation matrix (figure 3) and correlation of each feature with the label.



Feature	Correlation
density	-0.319747
chlorides	-0.210518
volatile acidity	-0.195547
total sulfur dioxide	-0.173394
fixed acidity	-0.112149
residual sugar	-0.098631
citric acid	-0.009790
free sulfur dioxide	0.013386
sulphates	0.054161
pH	0.100084
alcohol	0.440394

Table 2: Feature correlation with label

Figure 3: Correlation matrix

As we can see in the table 2, alcohol is the feature that is the most strongly correlated with quality. This is further demonstrated in the figure 4 which depicts the distribution of alcohol for each class. The correlation of alcohol with quality is also visible in the violin plot of alcohol and quality (figure 5).

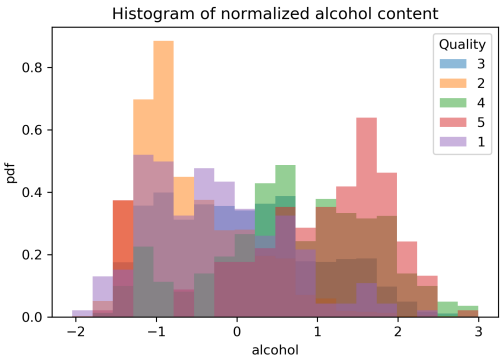


Figure 4: Distribution of alcohol for each class

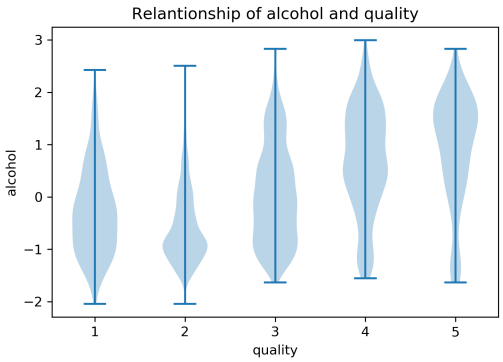


Figure 5: Violin plot of alcohol and quality

Another feature that is strongly correlated with quality is density, however, the predictive power of this feature is probably no greater than of alcohol because density if almost solely determined by the alcohol content.

The third feature with the highest absolute value of the correlation coefficient, is the chlorides content. In the Figure 5, we can see the most of the highest quality wines have low chlorides content.

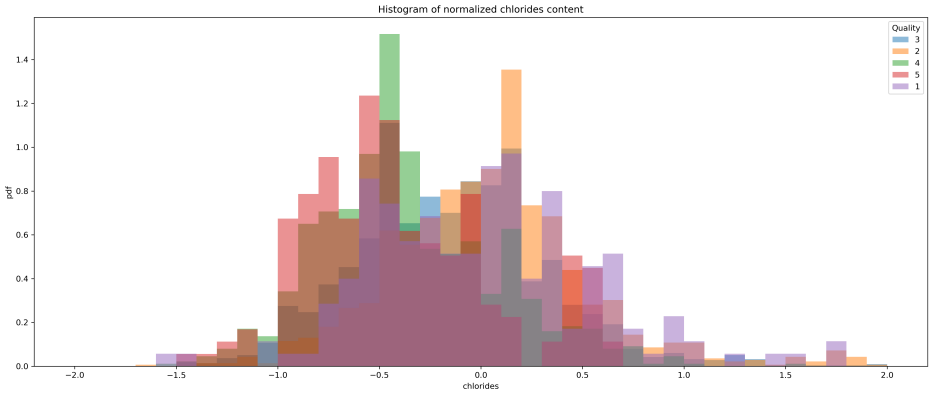


Figure 6: Distribution of chlorides for each class

2.1 Dimensionality reduction

In order to gain some insights into the structure of the data set, we used dimensionality reduction to visualize it in two and three dimensions. In figure 7 and figure 8 we can see the scatter plots of the the first two and three principal components, respectively. In both cases, we can see that the data is far from being linearly separable, however, there are some regions of the scatter plots dominated by one of the classes.

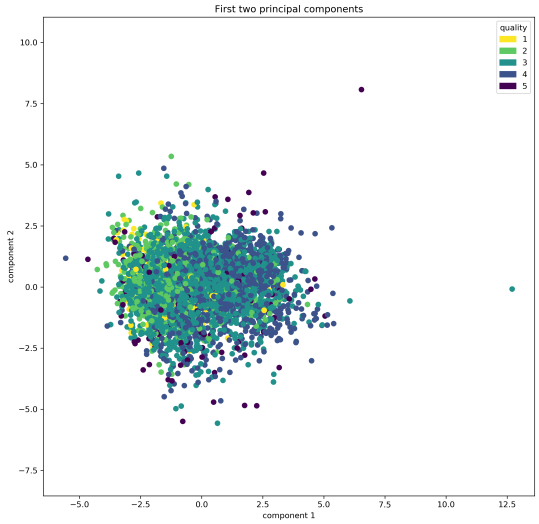


Figure 7: PCA with 2 components

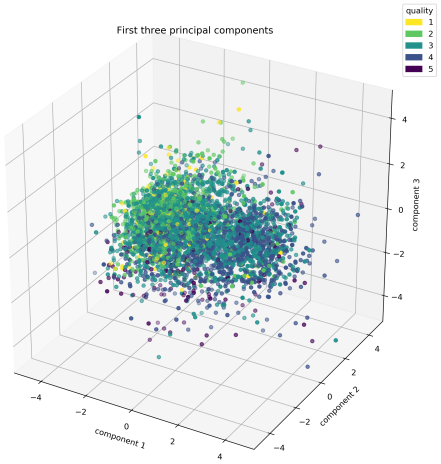


Figure 8: PCA with 3 components

Similar results are obtained if we apply t-distributed stochastic neighbor embedding algorithm. In figure 9 and figure 10 we can see that to some extent points of the same class are clustered together.

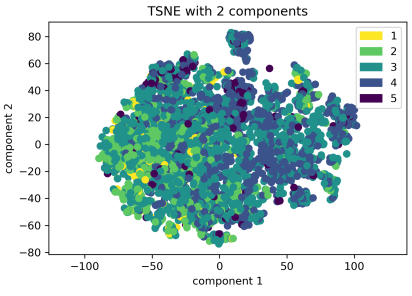


Figure 9: TSNE with 2 components

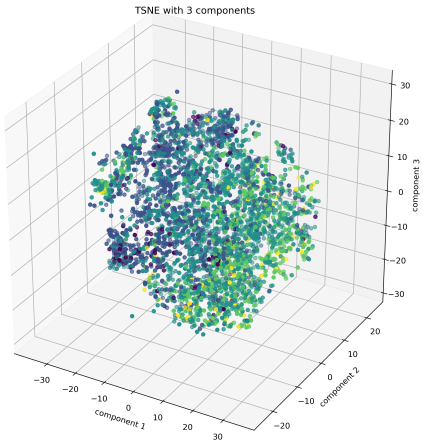


Figure 10: TSNE with 3 components

3 Methods

This section discussed the ML pipeline developed for this project.

3.1 Performance metric

Since we approach the problem as a classification task, not a regression, we use accuracy (percentage of correctly classified points) as our main performance metric. Note that we transformed labels, so now the data set is no longer imbalanced so accuracy is a suitable choice of performance metric.

3.2 Cross validation

We split our data into two sets, training (80%) and test (20%) set, and then we split training set into 5 CV sets. For each considered model, we use these CV sets to tune model's hyper-parameters. Note that the we fit the standard scaler separately for each CV set to avoid data leakage. Then, having chosen optimal hyper-parameters, we record model's accuracy on the test data.

3.3 Uncertainty

In order to measure uncertainty due to this splitting and use of non-deterministic ML methods (notably random forests), for each model we repeat the procedure 8 times (with 8 different random states) and we report the model's mean test score and standard deviation of these scores.

3.4 Considered models

In order to find the model that achieves the greatest accuracy, we considered three different classifiers: k nearest neighbor classifier (KNN), support-vector machine (SVM) and random forests (RF).

More specifically, we used KNN with neighbors weighted by their distances and we tuned the number of nearest neighbors. We have tried all values of n (number of neighbors) in $\{1, 3, 4, 10, 30, 50\}$ and in the process of k-fold cross validation we have found $n^* = 30$ to be the optimal value of this hyperparameter.
In case of SVM, we used radial basis function kernel and we tuned two hyperparameters: C (regularization parameter) and γ (kernel bandwidth). We have found the values $\gamma^* = 1$ and $C^* = 1$ to be optimal. Note that these chosen values of the hyperparameters are not on the edges of the considered ranges.

As our third model, we used random forests. We have tuned hyperparameters: n (number of trees) and d (max depth). In the process of k-fold cross validation, we have found the optimal values of these hyperparameters to be $n^* = 300$ and $d^* = 12$. Note that choosing max depth to be 12 basically means we don't restrict max depth therefore it is not a concern that we chose a hyperparameter value which on the edge of the considered range.

Please see table 3 for a detailed summary of the used models, hyperparameters and obtained accuracies.

Model	μ_{test}	σ_{test}	Hyperparameters	Range of hyperparameters	Optimal hyperparameter
KNN	0.681	0.0142	n neighbors	$n \in \{1, 3, 5, 10, 30, 50\}$	$n^* = 30$
SVM	0.661	0.0161	γ, C	$\gamma \in \{0.1, 1, 5, 50\}$ $C \in \{10^{-4}, 10^{-3}, 1, 10\}$	$\gamma^* = 1$, $C^* = 1$
RF	0.669	0.0146	n (estimators) d (max_depth)	$n \in \{100, 200, 300, 400\}$ $d \in \{8, 10, 12\}$	$n^* = 300$, $d^* = 12$

Table 3: Summary of the considered models

4 Results

This sections attempts to analyze the obtained results, which includes comparing the developed models and identifying the most important features.

4.1 Assessing used models

The baseline accuracy B of the data set is equal to 0.449. As we can see in *table* 4, all considered models achieved significantly better accuracy then the baseline.

Model	μ_{test}	σ_{test}	$(\mu_{test} - B)/\sigma_{test}$
KNN	0.681	0.0142	16.3
RF	0.669	0.0146	15.1
SVM	0.661	0.0161	13.3

Table 4: Summary of the obtained accuracies

KNN algorithm achieved the best accuracy ($\mu_{test} = 0.681$), however, this score is within two standard deviations from the accuracies obtained by random forests or SVM. Therefore, since KNN achieved the highest accuracy, we will mainly focus on this model in the further analysis but other models are also worth further investigation, especially if a speed of classification is a factor.

4.2 Confusion matrix

The figure 11 depicts the confusion matrix for KNN model evaluated on the test data. Even though the achieved accuracy of 0.68 is relatively low, we can see that when model missclassifies a point, it usually assigns a label that is close to a true label.

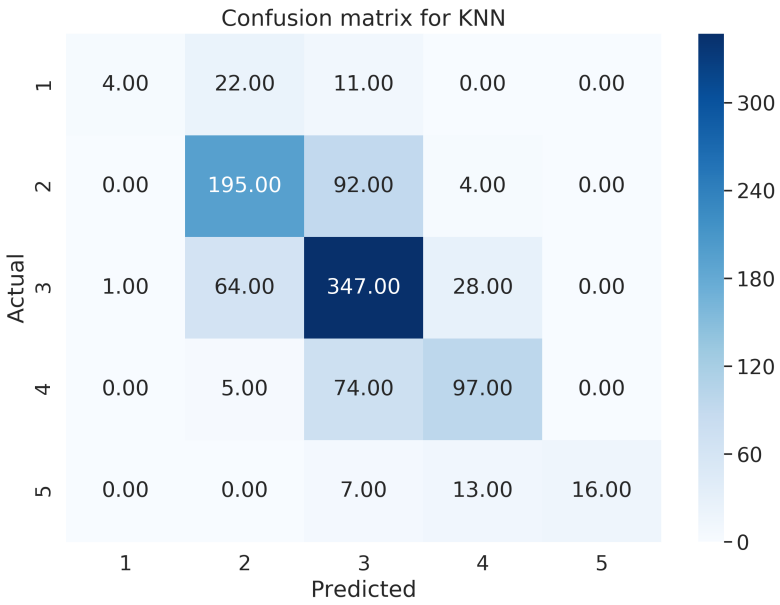


Figure 11: Confusion matrix for KNN algorithm evaluated on test data

4.3 Feature importance

In order to evaluate global feature importance, the permutation feature importance was calculated for KNN (figure 12) and random forests (figure 13). As it was suggested the exploratory data analysis, alcohol content seems to be the most important feature determining the quality of wine.

Furthermore, both models suggest that second most important feature is the volatile acidity. Rather surprisingly, random forests seem to assign greater weight to this feature than KNN. This results is consistent with the fact that volatile acidity is strongly negatively correlated with the wine quality ($\rho = -0.2$).

Even though chlorides were strongly negatively correlated with quality ($\rho = -0.21$), the permutation test does not identify it as a very important feature.

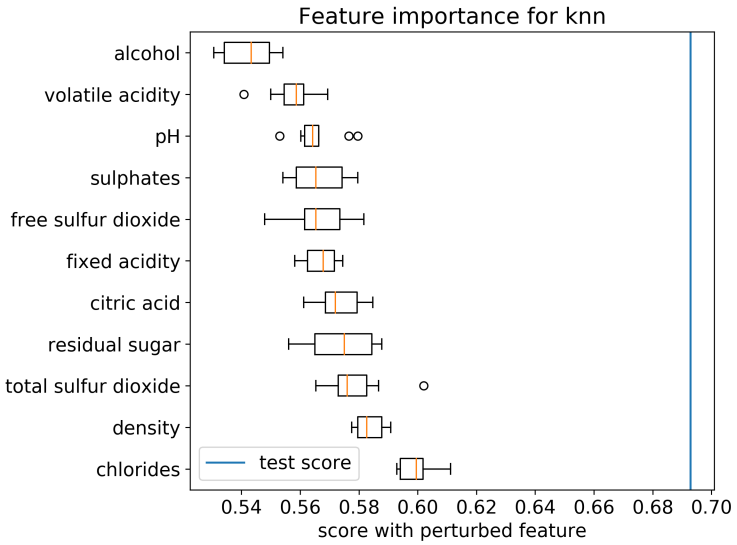


Figure 12: Permutation feature importance KNN ($n = 30$, points weighted by distances)

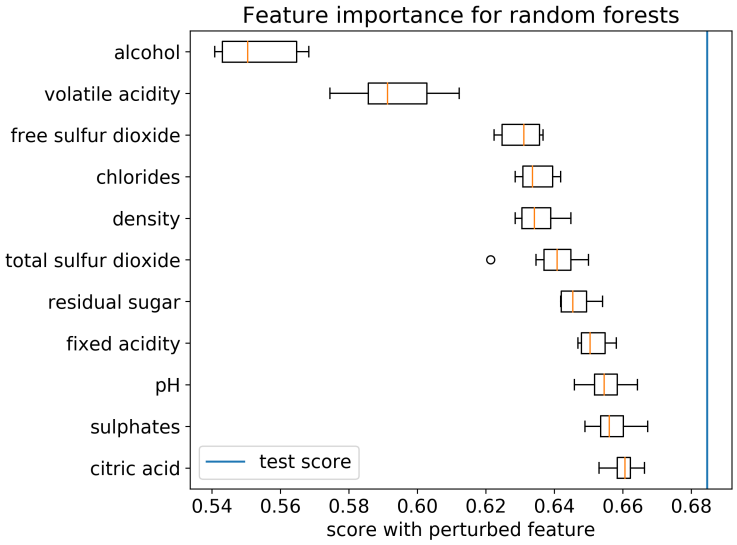


Figure 13: Permutation feature importance RF ($n = 300,\ d = 12$)

4.4 Interpretation

Since all three considered models achieved accuracies significantly better then the baseline, it is very unlikely that the labels (wine quality) were assigned at random due to purely subjective taste of the wine experts.

This suggests that there is some level of objectivity in determining the quality of wine.

The presented results also have some implications for the stakeholders: wine producers and wine consumers (experts). Both group of stakeholders can be guided by the fact that alcohol and the volatile acidity are strong predictors of wine quality.

In case of wine producers, it is essential to mention that the correlation of higher alcohol content with the perceived wine quality does not necessarily imply the causation. That is, artificially adding more alcohol to wine, will not necessarily increase its quality because there might be some confounding variables at play. However, if the wine producers do not change their approach to wine making under influence of this report, alcohol content will be a strong predictor of quality of newly developed wines. Hence the result can be useful while deciding whether to start a mass production of a new wine.

Note these implications for stakeholders are valid for wines which are drawn from the same distribution as the wines from our data set.

5 Outlook

This sections aims to discuss possible improvements and weaknesses of this project.

5.1 Possible improvements

In order to improve the classification accuracy, one can attempt to combine the developed models into one voting classifier or other ensemble methods. Alternatively, developing new classifiers like multi-layer perceptron (deep learning) might also yield better results.

In order to improve interpretability, one could investigate local feature importance and different metrics of global feature importance. Furthermore, it would be interesting to investigate the feature importance of a classification task restricted to labels 3 and 4.

5.2 Weaknesses

The most significant weakness of our modelling approach is lack of feature engineering. It is very likely that in-depth knowledge of wines would result in engineering new features and hence obtaining better accuracy (without developing new models).

Another weakness of our approach is the performance metric that penalizes each missclassification equally. Probably the considered stakeholders would like to penalize classification of 4 as 1 more as classification of 4 as 3. One metric that can achieve this is mean squared loss.

5.3 Other approaches

Possibly the project would be more impactful if we approached the problem as a regression problem. This would possibly make interpretability more straightforward and we would avoid an arbitrary regrouping of labels.

References

[1] Paulo Cortez, António Cerdeira, Fernando Almeida, Telmo Matos, José Reis, *Modeling wine preferences by data mining from physicochemical properties*, Decision Support Systems, 2009.

[2] Ayan Ghosh, *Red Wine Quality Analysis Final*, rpubs.com, 2017.

[3] Jiayi Liao, *Exploratory Data Analysis on Red Wine Quality*, rpubs.com, 2019.