Machine Learning Approaches Towards Red Wine Quality Prediction

Miken Guo, Chuxuan He, Jialong Liu
Washington University in St. Louis
ESE417: Introduction to Machine Learning and Pattern Classification
Dr. Jinsong Zhang

December 18, 2023

1 Introduction

Machine learning includes a variety of techniques that enable machines to learn from data and identify patterns. Its algorithms can be categorized into supervised, unsupervised, and reinforcement learning, each with meaningful applications. The application of machine learning has revolutionized many fields, including image and speech recognition, predictive modeling of business operations and even prediction of wine quality.

There have been attempts to use data mining or machine learning methods to predict wine quality based on physicochemical data since 1991 (A. Asuncion, 2007), but usually limited to small datasets. From May 2004 to February 2007, (Cortez et al., 2009) performed common physicochemical tests on vinho verde, a category of wine produced uniquely in the northwestern region of Portugal, and built a dataset of records of 1599 red wine samples and 4898 white wine samples. They labeled the dataset by inviting at least three assessors to evaluate and grade wine quality, and the final score is the median of the grades.

Our goal in this project is to apply common machine learning techniques to the red wine quality dataset, explore the relationship between physicochemical characteristics and the human sensory quality of red wine, and build reliable and interpretable models to predict wine quality based on the insights we gain.

In our project, we adopt three machine learning methods to perform the multi-class classification task, namely Artificial Neural Network, Random Forest and Support Vector Machine. All models give satisfying performance on the task while Random Forest achieves the highest accuracy on the dataset. In addition, we also gain some insight on which variables are most important in determining the sensory quality of red wines.

2 Exploratory Analysis of the dataset

The Red Wine Quality dataset(Cortez et al., 2009), obtained from UCI Machine Learning Repository, contains several chemical and physical attributes of red wines. It contains 1599 records and 12 features including 11 input features in float data type and the target feature quality in integer data type, and there are no null values. The dataset is structured in a clean tabular format which enables systematic analysis and predicts the quality of the wine based on these features. The physicochemical meaning of features are listed in Table 1(Swafford, 2021):

We confirm that all distribution of the features are regular. To better explore the information in the dataset, we generate box plots that describe the relationship between each feature and the quality. (Figure 1) To explore more about the target label, a barplot (Figure 2) was generated, which describes the distribution of quality levels.

Features	Description
Fixed acidity	Most acids involved with wine or fixed or nonvolatile
Volatile acidity	The amount of acetic acid in wine
Citric acid	Found in small quantities, citric acid can add 'freshness' and flavor to wines
Residual sugar	The amount of sugar remaining after fermentation stops
Chlorides	The amount of salt in the wine
Free sulfur dioxide	The free form of SO2 exists in equilibrium between molecular SO2 and bisulfite-ion
Total sulfur dioxide	Amount of free and bound forms of SO2
Density	Density of wine depends on the percent alcohol and sugar content
pH	Describes how acidic or basic a wine
Sulphates	A wine additive which can contribute to sulfur dioxide gas (SO2) levels
Alcohol	The percent alcohol content of the wine

Table 1: Description of features contained in dataset.

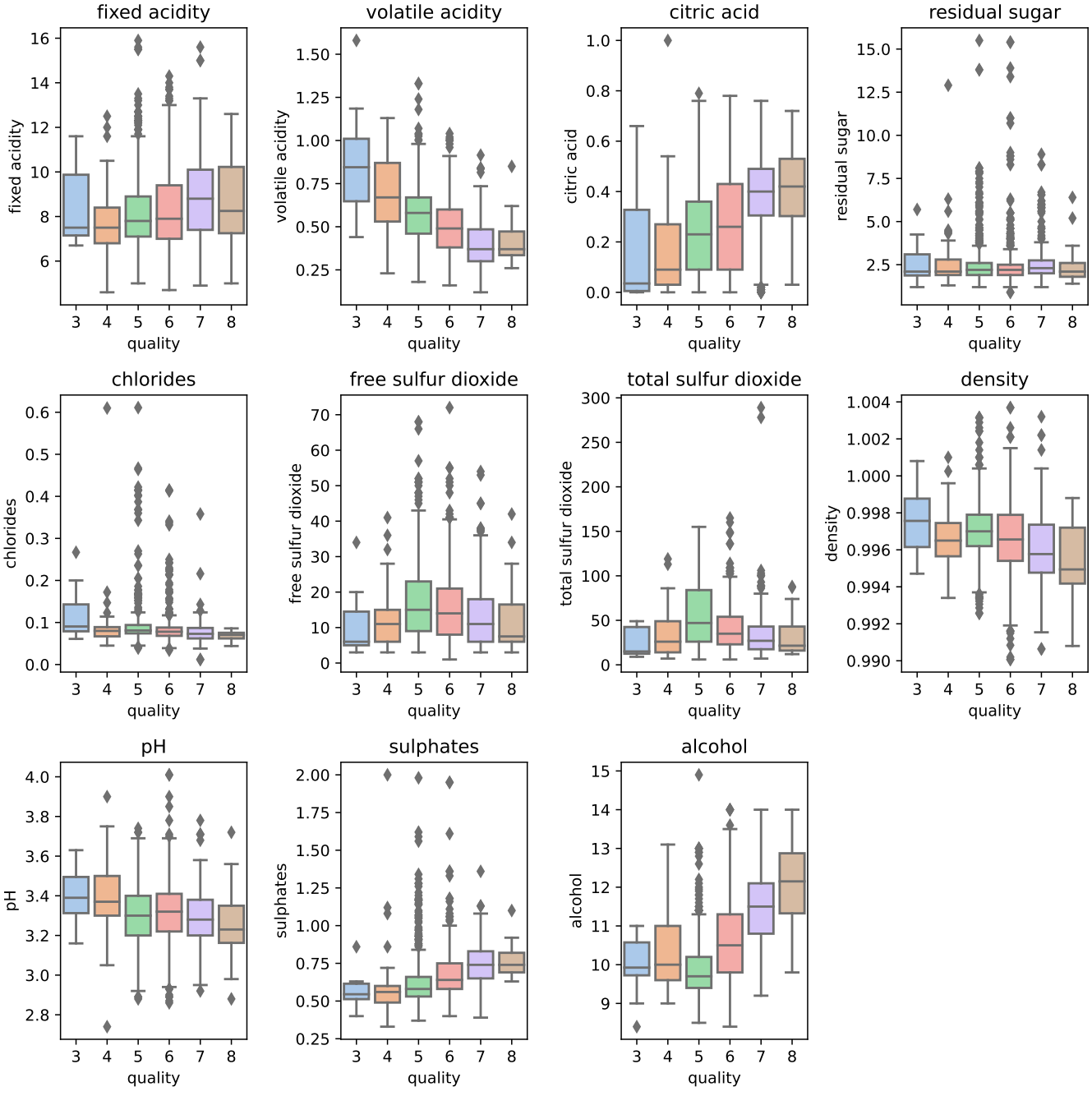


Figure 1: box plots between each feature and the quality label

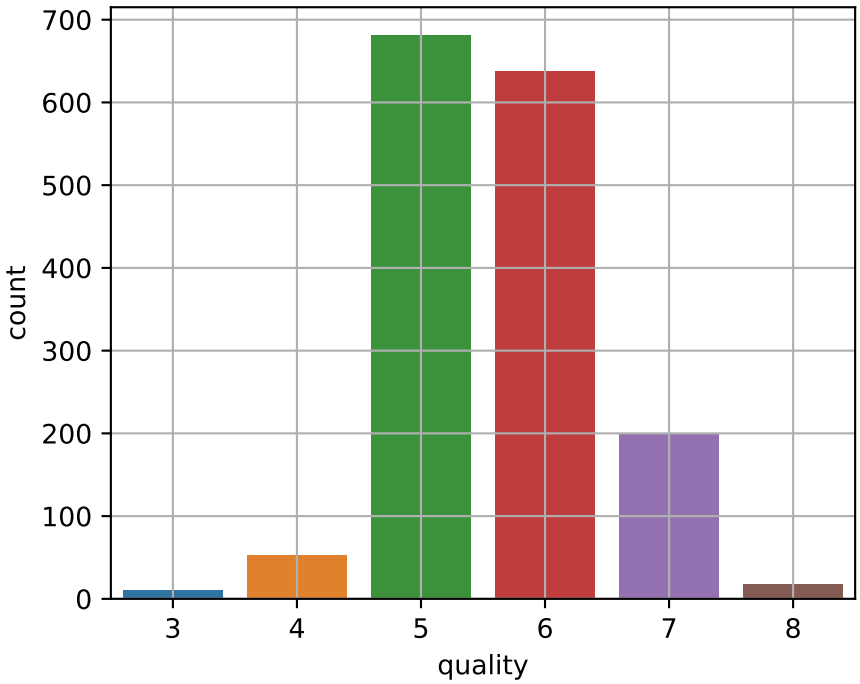


Figure 2: bar chart of target value 'quality'

From the bar chart, the distribution of quality level roughly follows the normal distribution. Level 5 and level 6 account for a relatively large proportion of about 82.5%, while other levels account for about 17.5%.Through data visualization, we can better understand the association of each feature with quality and the distribution of quality, which can help us better apply the model to make predictions.

3 Methods

In this section we will introduce the feature engineering techniques and models used in our project, as well as the hyper parameter tuning procedure.

3.1 Feature Engineering

(C. and V., 2001) mentioned that there is an interaction effect between free SO_2 quantity in a wine and the pH of the wine called molecular SO_2 that affects the tasting of wine, as effectiveness of the SO2 to protect the wine from oxidation is dependant upon the pH. Molecular SO_2 can be calculated by the following formula:

$$Molecular \quad SO_2 = Free \quad SO_2/(1 + 10^{pH} - 1.81)$$

We calculate Molecular SO_2 and include it as a feature in our model. Some suggest discarding free SO_2 feature after adding Molecular SO_2, but we decide to keep it in our feature set and fit our model with 12 features.

3.2 Models

3.2.1 Random Forest

Random Forest(Breiman, 2001) is an ensemble machine learning method that fits a number of decision trees and uses the average or majority vote of each tree's prediction as output to control over-fitting, reduce model variance and improve the predictive accuracy. To increase the variance between each tree, Random Forest uses bootstrap methods to generate sub-samples to fit each tree and consider

only a randomly selected subset of features at each split of a tree. In our project, we use the scikit-learn package to implement the Random Forest algorithm.

3.2.2 Artificial Neural Networks

Artificial Neural Network(ANN)(Hinton, 1990) is one of the most important computational machine learning models inspired by the structure and function of the human brain(Wikipedia, 2023). ANN implements complex information processing and learning through connections and information transfer between nodes, and updating weights through backpropagation. In this project, we use the multi-layer perceptron model to implement the Artificial Neural Network. Multi-layer perceptron model is a type of artificial neural network with multiple layers, including an input layer, hidden layers, and an output layer. In the scikit-learn package, MLPClassifier implements a multi-layer perceptron model which uses backpropagation to update the model.

3.2.3 Support Vector Machine

The Support Vector Machine (SVM)(Platt et al., 1999) is a versatile supervised machine learning model, mostly used for classification tasks. It functions by finding the optimal hyperplane that best separates different classes in the feature space. In our project, we employ the SVM algorithm provided by the scikit-learn package.

For the SVM model, we use the 'one-versus-one' method implemented by SVC function and particularly focus on tuning key hyper parameters to enhance its performance.

3.3 Hyper Parameter Tuning

To obtain optimal performance of each model, we propose some hyper parameter space for each model and exhaustively search over the space for the best hyper parameter setting using a 3-fold cross validation on the training set. The procedure is implemented by scikit-learn package's GridSearchCV function. The detail of our hyper parameter choice in each model is listed in the appendix.

4 Results and Analysis

To evaluate the performance of three models, we did a train-test split of ratio 7:3, train our model on the training set and evaluate on the test set. We first calculate the performance metric of each models on the test set and on each class in a one-verse-all way. The metric we choose is accuracy for the whole test set and precision, recall, F1-score for each class, which are common metrics used in machine learning literature. We also use weighted average to compute the last three metrics for the whole test set.The performance of our models on the test set is listed in table below:

	Model	RF	ANN	SVM
Quality	Metrics			
3	Precision	0	0	0
	Recall	0	0	0
	F1-Score	0	0	0
4	Precision	0	0	0
	Recall	0	0	0
	F1-Score	0	0	0
5	Precision	69.74	64.9	63.35
	Recall	80.92	74.81	77.86
	F1-Score	74.91	69.5	69.86
6	Precision	61.43	53.85	59.06
	Recall	65.65	58.78	57.25
	F1-Score	63.47	56.2	58.14
7	Precision	57.14	36	46.67
	Recall	36.36	20.45	31.82
	F1-Score	44.44	26.09	37.84
8	Precision	0	0	0
	Recall	0	0	0
	F1-Score	0	0	0
Total	Accuracy	65	57.5	59.69
Weighted Average	Precision	61.55	53.56	56.53
	Recall	65	57.5	59.69
	F1-Score	62.76	55.05	57.6

Table 2: The performance of Random Forest, Artifical Neural Network, SVM on each class and the whole test set.

Overall, random forest model has the highest performance in all metric over the whole test set as well as each class. We also notice that all models give poor performance on the minority classes, i.e fail to predict samples with extremely bad (quality 3-4) or extremely good (quality 8) tastings. This is probably due to the severe imbalance of the classes.

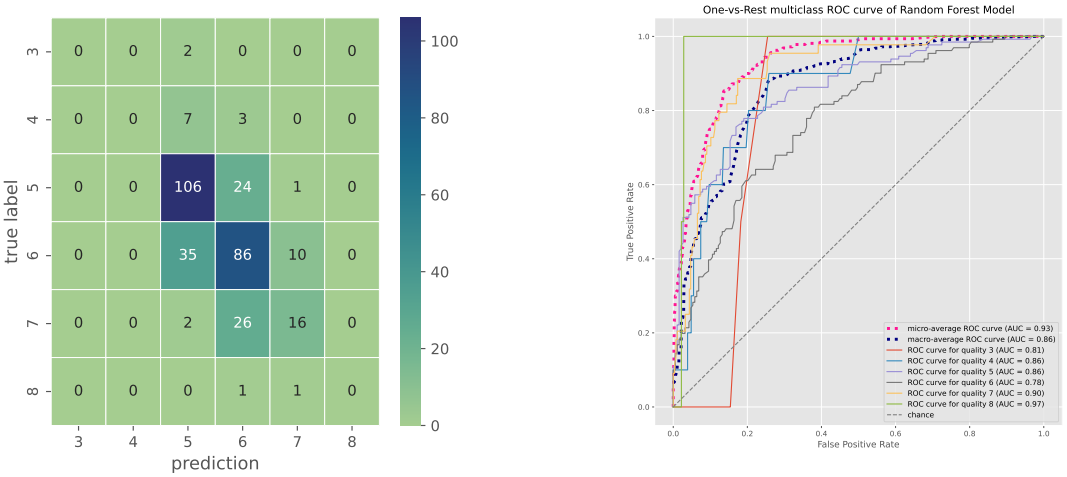
Similar to (Cortez et al., 2009), we assume that the difference of taste between two adjacent classes, e.g a wine with quality label 5 and a wine with quality label 6, can be omitted. Based on this idea, we evaluate the performance of three models with a tolerance value of 1, i.e a sample of quality 4 is considered correctly classified if our model predicts it to be of quality either 3,4 or 5; but considered wrongly classified if our model predicts it to be of other quality. The performance of models with tolerance is listed in the table below:

	Quality	3	4	5	6	7	8	Total
Model	Accuracy							
RF		0	70	99.23	100	100	100	98.125
ANN		0	80	97.71	100	100	100	97.8125
SVM		0	70	97.71	100	100	100	97.5

Table 3: Accuracy of model prediction with tolerance on each class and the whole test set.

From Table 2 we can draw two conclusions: Firstly, all models have nearly perfect accuracy when we tolerate classifying a sample into its adjacent class. This confirms our guess that wines with similar qualities share similar physicochemical properties, and models built on these physicochemical features have robustness to misjudge or personal taste of assessors. Secondly, Random Forest is still has the best performance when we evaluate with tolerance, so in the rest part of the section, we will focus on Random Forest Model when interpreting the results.

The confusion matrix and ROC curve of Random Forest model on the test set is as follows:



(a) confusion matrix (b) ROC curve

Figure 3: Detailed performance of Random Forest Model

The result is satisfactory for a multi-class classification problem as we note that large elements in the confusion matrix are mainly on the diagonal and the ROC curve are significantly dominating random classification. Furthermore, we achieve AUROC score ranging from 0.78 to 0.97 for each class, and average AUROC for test set are 0.93 (micro) and 0.86 (macro) respectively, which indicate good classification performance.

We want to further confirm which variables are most crucial in determining the sensory quality of red wine. In order to gain more insight, we perform a permutation feature importance test on the Random Forest model. Each time we shuffle one feature while preserving the other features and investigate the decrease in model performance. The top 6 most important features are shown in the figure below:

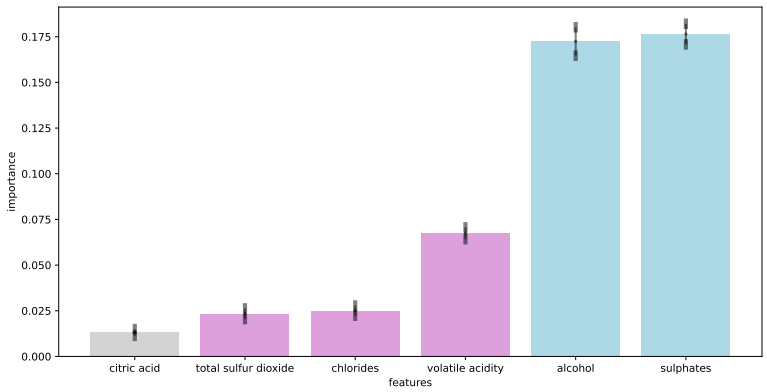


Figure 4: top 6 most important features and their importance

According to the figure, alcohol and sulfates are the most crucial feature in affecting the quality of red wine, and after them are several compounds like acids and chlorides. This agrees with our knowledge that red wines higher alcohol content typically have a fuller body and sulfates help stabilize the color and flavour of wine.

5 Conclusions

In our study, we employed three distinct machine learning algorithms: Random Forest, Artificial Neural Network, and Support Vector Machine, to execute a multi-class classification job over a dataset

about red wine quality. The models achieved respective accuracies of 65%, 57.5%, and 59.69%. Nonetheless, upon assessing the efficacy of the three models with a tolerance level of 1, there is a significant enhancement in accuracy, reaching 98.125%, 97.8125%, and 97.5%, respectively. In conclusion, the Random Forest model outperforms with an initial accuracy of 65%, which further escalates to 98.125% when the tolerance is considered. We furtherly plotted the confusion matrix and ROC curve of the Random Forest model. We found that misclassified are mostly considered as its adjacent classes and sulfates, alcohol are two most decisive factors affecting the quality of red wine.

In general, we are successful in implementing a machine learning model that is effective in wine quality prediction, and we gained deeper insight about how wine's basic physicochemical properties affects the overall quality of a wine. For future studies, a dataset with a more even distribution of samples across all classes could prove to be more valuable, or we can explore if certain data augumentation techniques like SMOTE in imbalanced learning field can be of use.

Contributions of each member:
Miken Guo: Introduction, Random Forest model, Results and Analysis
Chuxuan He: Analysis of the dataset, ANN model, Conclusion
Jialong Liu: Support Vector Machine, Conclusion

6 Appendix

6.1 Experiment Environment

Environment	Specifics and Versions
System	Linux 6.1.58 (Google Colab)
Python	3.10.12
numpy	1.23.5
pandas	1.5.3
matplotlib	3.7.1
seaborn	0.12.2
scikit-learn	1.2.2

Table 4: Specifics of the system, programming language and packages used in our project.

6.2 Hyper Parameter Setting

Model	Hyper parameters
Random Forest	class weights: balanced subsample, criterion: gini, max_features: 4, n_estimators: 700
ANN	activation: relu, alpha: 0.1, hidden_layer_sizes: (200, 100), learning_rate_init: 0.01
SVM	C: 3, gamma: scale, kernel: rbf

Table 5: Hyper parameter settings for each model; hyper parameters not specificed here are set to default values of scikit-learn.

References

A. Asuncion, D. Newman. (2007). *UCI Machine Learning Repository.* http://www.ics.uci.edu/ mlearn/MLRepo
Breiman, Leo (2001). "Random forests". In: *Machine learning* 45, pp. 5–32.
C., Delfini and Formica J V. (2001). *Wine microbiology: science and technology.* CRC Press.
Cortez, Paulo et al. (2009). "Modeling wine preferences by data mining from physicochemical properties". In: *Decision support systems* 47.4, pp. 547–553.

Hinton, Geoffrey E (1990). "Connectionist learning procedures". In: *Machine learning*. Elsevier, pp. 555–610.

Platt, John et al. (1999). "Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods". In: *Advances in large margin classifiers* 10.3, pp. 61–74.

Swafford, Crystal (2021). *Red Wine Quality Analysis*. https://rpubs.com/cswaff7/775970.

Wikipedia (2023). *Neuromorphic engineering*. https://en.wikipedia.org/wiki/Neuromorphic$_e$*ngineering*.