Determining Wine Quality Test with Machine Learning

Ryan Tracey
500824573
ryan.tracey@ryerson.ca
Toronto Ontario, Canada

Jeremy Ng
500882192
jeremy.ng@ryerson.ca
Toronto Ontario, Canada

Dingqi Liu
500840427
dingqi.liu@ryerson.ca
Toronto, Ontario, Canada

Abstract -- Wine has been a delicious beverage for humanity for thousands of years. There are numerous wines, and the quality differentiates from one to another. Wine experts determine the quality by physical properties such as the taste, the smell, and the texture. However, certain chemical properties might have a great effect on the quality. The Kaggle website has a dataset of wine properties with corresponding quality scores determined by wine experts. Using the dataset and programming tools, we trained various machine learning models using supervised classification and regression learning algorithms. After a series of experiments and analyses, we noticed that linear algorithms generally have better accuracy than others, meaning that wine quality data is linearly separable. In the end, we are able to predict a numerical wine quality score based on the chemical properties of the wine. Our best models have an accuracy of around 60% and a small loss. The result shows that there is a correlation between wine quality and its chemical properties.

I. Introduction

When it comes to wine, determining the quality is generally an extremely subjective thing, but there are certain chemical characteristics of wine that generally impress wine experts and critics. The four main categories that determine wine quality are environmental (sunlight, temperature, soil, etc.), the species of grape, viticultural practices (how the winegrower influences the growing grape) and enological practices (oxygen, sulfur dioxide and oak). [4] The final result of these four categories produces a certain chemical composition for the wine which is what our dataset contains, and is what we have used to train a variety of different machine learning models to try to assess which machine learning algorithms perform best when it comes to predicting wine quality.

Due to the nature of how wine quality is judged, it presented a few unique problems for us to attempt to overcome. Because wine quality is judged by a mixture of subjective opinion of taste and objective factors relating to how it was made, it became challenging to get high accuracy ratings with our models as the features were not a perfect mathematical or logical representation of the quality rating of the wine. We incorporated a number of techniques to attempt to overcome this fact to see if there was enough correlation between the chemical compound of a wine and its quality.

The problem at hand is considered a multiclass problem, having the wine quality label be represented by a value from 1 to 10. We decided to experiment and create models using 3 different categories of algorithms to see which yielded the best results and examine why. We used regression algorithms like multinomial logistic and linear regression many different classifiers including SVM, naive bayes and random forest classification and finally a muli-layer perceptron neural network and our findings were not what we expected. Some of the techniques we used to further improve the dataset were outlier removal, standard scalars and cross validation, and to improve the models we used bias, variance and overfitting analysis, as well as hyper parameter optimization for our Neural Network, all of which will be explored deeper further in the report.

II. Problem Statement

When judging wine, everyone may have different opinions. It is difficult to give a particular bottle of wine a definitive score due to the nature of subjectivity. However we can determine what combination of chemical attributes can potentially make a great wine.

Our main problem is "What machine learning algorithms can accurately predict the quality of the wine?" The goal is to analyze two given datasets of red wine and white wine [2] using various machine learning techniques including regression, neural networks and classification. We can determine the best algorithm using its success rate as a metric for accurately predicting the quality of the wine from a scale of 1 to 10.

A sub problem that we can solve is to compare the chemical attributes of both red wine and white wine, that make them receive high quality scores.

III. Dataset

The two datasets [2] used in this report include the chemical compound of 1500 red wine samples and 4900 white wine samples. The datasets were collected in 2009 of different red and white wines that were of the Portugueses “Vinho Verde” wine. The datasets include the physiological composition of each wine as features and a quality rating from 1-10 as the label. The features included in the dataset are as follows: fixed acidity, volatile acidity, citric acid, residual sugar, chlorides, free sulfur dioxide, total sulfur dioxide, density, pH, sulphates, and alcohol. Since the output label is discrete we considered this to be a classification problem but also used regression algorithms and neural networks on the datasets. Looking at the labels of the dataset it is clear that the dataset is not balanced, as the vast majority of wine is rated at a 5/10 or close to it. This can be viewed in figure 1 (Red wine) and figure 2 (red wine) below.

Figure 1. Red Wine dataset

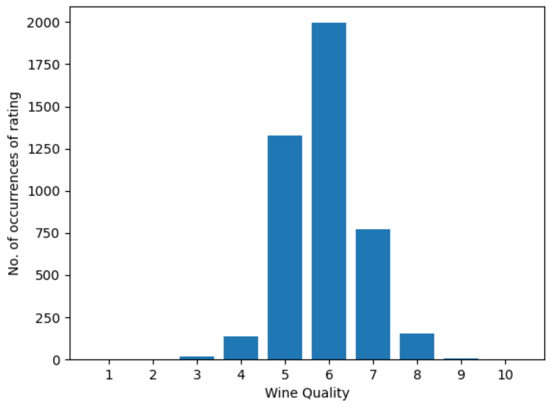
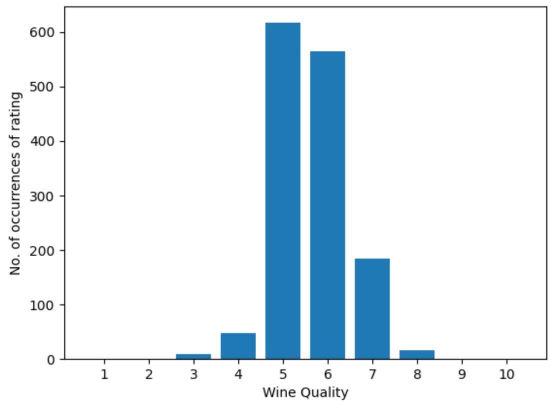


Figure 2. White Wine dataset



For preprocessing of the dataset we used two different methods to attempt to improve our results. The first was Standard Scaling to standardize features by subtracting the mean and scaling to unit variance. After visualizing the data we noticed there were a number of outliers in different inputs so the other preprocessing method used was to remove outliers using the Winsorization method to detect them. More on these preprocessing methods in the Analysis section.

IV. Methodology

A. Feature extraction and preprocessing

The datasets we have are provided by Keggle in two .csv files, each contain features for red wine and white wine respectively. We use the "pandas" Python package to import all the data for red wine and white wine separately, and explicitly set the "quality" column as the labels of the data.

During the importing phase, we use the train_test_split function of Scikit-Learn to split and shuffle the data. For the parameters, we use a random_state of 0, which is a seed that controls the shuffling applied to the data set. Using the same random_state can have a reproducible output across multiple function calls.

B. Methods and models

Since we are expecting specific output values, we only used supervised machine learning algorithms. The implementation is done through Python with Pandas, Numpy, and Scikit-Learn libraries.

We primarily focused on classification machine learning algorithms, including Logistic Regression, Naive Bayes, K-Nearest Neighbors, and Support Vector Machines. For logistic regression, since it only supports binary classification, we converted the labels into two classes in pre-processing. Wines with a score greater than 6 are considered good, and the rest is considered bad.

Even though we are dealing with a classification problem, yet the labels are numerical values, so we have the option to use regression machine learning algorithms as well. We performed the linear regression algorithm on the data, and then in post-processing, the continuous values are rounded into the nearest integers.

In the end, as the previous methods did not give optimal results, we decided to implement the deep learning Multi-Layer Perceptron machine learning model by Scikit-Learn to further improve our predictions. We tried a number of different combinations of hyperparameters, but in the end, we kept one hidden layer of 100 neurons and relu for activation.

C. Evaluation metrics

For evaluating, comparing and choosing the best model, we considered the overall accuracy, average loss, confusion matrix, and cross-entropy loss. We use overall accuracy to compare how many predictions are exactly the same as the true labels. We use the cross-entropy loss to see the numerical differences between the true labels and the predictions. In addition, we generated confusion matrices using Scikit-Learn. By using the confusion matrix, we are able to visualize how many predictions are successful and where the false predictions lie. We also consider the average loss although we are dealing with a classification problem. Since the labels of our data are integers, we have the option of using average loss to find the numerical differences between our predictions and the true labels. We use the average loss as an addition to the cross-entropy loss to monitor the numerical differences between the predictions and the true labels.

By using the combination of the overall accuracy, cross-entropy loss, confusion matrix, and average loss, we are able to have a clear understanding of the performance of our machine learning models.

V. Initial Results

Figure 3. Table showing the initial results of all algorithms on Red and White wine datasets.

	Red	**White**
Linear Regression	0.623	0.49
Multinomial Logistic Regression	0.621	0.517
Naive Bayes	0.548	0.449
KNN	0.471	0.465
SVM	0.504	0.431
RFC	0.688	0.669
MLP	0.588	0.497

In figure 3, we can see the results provided from various algorithms. The highest success rate is the random forest classifier. Both linear regression and logistic regression models seem to work best for the red wine dataset at 62.3% and 62.1% respectively. However, Naive Bayes, K-Nearest Neighbours and Support Vector Machines seem to have the worst performances, averaging around 50%. The accuracy for the white dataset across all models except for RFC are low, ranging from 43.1% to 51.7%. Moving forward, we can apply optimizations to both dataset and machine learning implementations in order to increase the accuracy.

VI. Results And Discussions

A. Removing Outliers

One issue that was noted while getting the initial results was that some of the inputs had significant outliers that could distort the results by affecting the standard deviation and mean values of the dataset. Shown in the figures below is the distribution of values for residual sugar (figure 4) and chlorides (figure 5). As you can see there are significant outliers on the top end of values in both features, there were similar outliers in the chlorides and volatile acidity features. We used a Winsorizing algorithm [7] to transform the data to remove extreme outliers from the top end of these features so it would not influence the machine learning models. This optimization unfortunately had slightly positive results but overall the improvement was minimal.

Figure 4. Distribution of residual sugar.

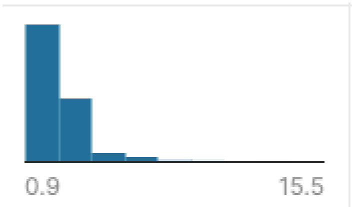
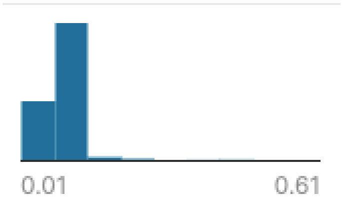


Figure 5. Distribution of chlorides



B. Bias and Variance Analysis

Next we took a look at the bias and variance of our algorithms to try and determine if there was any under/overfitting for our models. Figure 6 which is available below shows the bias and variance of one of each type of algorithm after our initial results, specifically Linear Regression, SVM, and MLP. Upon analysis of the results, our machine learning models on this dataset have a relatively high bias but a very low variance which tells us that there is underfitting in our models. This implies that the model is unable to fully get the pattern of the dataset. Unfortunately this could be caused by a lack of data, considering there are only 1500 red wine inputs and 4900 white wine inputs. So collecting more data for these models could be a good solution going forward to reduce underfitting to allow our algorithms to more accurately model the data.

Figure 6.

Algorithm	**Bias**	**Variance**
Linear Regression	0.5398	0.0299
SVM	0.5143	0.0531
MLP Neural Network	0.4541	0.2107

C. Standard Scalars to dataset

We normalized our dataset by applying standard scalers from scikit-learn [6]. Standardization helps for our algorithms to converge faster. Most noticeably, it sped up logistic regression. However, it does not seem to impact our results.

D. Cross validation

We applied 3 fold cross validation on all algorithms. Unfortunately, the only noticeable improvement is from our logistic regression with a 5% increase in accuracy [Figure 6]. The cross validation score for linear regression in both red and white wine datasets were significantly cut in half [Figure 6].

E. Hyperparameters optimization for NN

A number of different combinations of hyper-parameters have been used to train the MLP model. The GridSearch function by Scikit-Learn is used to tune the hyper-parameters. However, tuning the activation methods, the number of layers, and the number of neurons per layer did not have a noticeable consistent improvement on the predictions.

F. Overfitting

Multiple evaluation metrics are calculated and considered to identify possible overfitting, including accuracy, cross-entropy loss, confusion matrix, and regression loss. The prediction is run on both the train data and the validation data to detect and prevent overfitting.

Figure 7. Table showing results of all algorithms after applying optimizations.

	Red	**Red-CV score**	**White**	**White-CV score**
Linear Regression	0.617	0.314	0.49	0.234
Multinomial Logistic Regression	0.598	0.65	0.528	0.531
Naive Bayes	0.538	0.552	0.459	0.416
KNN	0.569	0.56	0.533	0.486
SVM	0.681	0.627	0.622	0.551
RFC	0.652	0.642	0.652	0.57
MLP	0.59	0.585	0.548	0.489

VII. Implementation And Code

The project was done on Github [1] with the three members of the project writing and committing code to create the finished product. The dataset [2] was obtained from Kaggle, and there is a projects section on the Kaggle site that details many different projects that others have attempted with this dataset that were consulted at various points to attempt to overcome the challenges while working on this dataset. Also much of the course material from the lectures was used to implement the algorithms and also tweak the code and investigate certain issues we were having to try to achieve an optimal result. Different wine books and websites were consulted to understand more about our features and labels and all were cited accordingly in the references section of this report.

To implement the machine learning algorithms Scikit [6] was used on the dataset. Using scikit we were able to create an object from a specific machine learning class in scikit and feed it certain parameters and run functions like train and fit to achieve the models used in the assignment. Matplotlib was also used to plot the different figures that were used in this report as well as throughout the assignment to visualize what was going on with our data and code.

VIII. Conclusions

The initial training and predictions were unsuccessful due to the incorrect use of algorithms, unbalanced dataset, and lack of data. The labels of the data are clustered around five and six, and there are very few low and high scores. In addition, the lack of data may also have caused a high bias and low accuracy in the beginning. We were able to improve the models by removing outliers, applying standard scalars, cross-validation, and analyzing evaluation metrics. We noticed that linear machine learning algorithms such as linear regression and logistic regression perform well in comparison to others. This tells us that our data is linearly separable. In the end, we are able to have an accuracy of around 60% for our best models. By analyzing the loss and confusion matrix, we noticed that the loss is relatively small, and the false predictions are mostly adjacent to the true labels. Since the wine quality scores are a series of integers and can be ranked, we believe that our models are sufficient for making wine quality score predictions.

IIX. References

1. Tracey, R., Ng, J. and Liu, D., 2021. [online] cps803-project. Available at: <https://github.com/KingJeremyNg/cps803-project> [Accessed 11 December 2021].Cortez, P.,

2. Kumar, V. Prediction of quality of Wine. (2017). Kaggle [Data file]. Retrieved October 4, 2021, from https://www.kaggle.com/vishalyo990/prediction-of-quality-of-wine/notebook.

3. Weka 3: Machine Learning Software in Java. Weka 3 - Data Mining with Open Source Machine Learning Software in Java. (n.d.). Retrieved October 4, 2021, from https://www.cs.waikato.ac.nz/ml/weka/.

4. Reynolds, A., 2010. Managing wine quality. Cambridge: Woodhead Publishing, pp.107-133.

5. Cerdeira, A., Almeida, F., Matos, T., & Reis, J. (2009). Modeling wine preferences by data mining from physicochemical properties. Decision Support Systems, 47(4), 547-553. doi:10.1016/j.dss.2009.05.016

6. Scikit-learn.org. 2021. scikit-learn: machine learning in Python. [online] Available at: <https://scikit-learn.org/stable/index.html> [Accessed 11 December 2021].

7. Statistics How To. 2021. Winsorize: Definition, Examples in Easy Steps. [online] Available at: <https://www.statisticshowto.com/winsorize/> [Accessed 11 December 2021].