The Key to Good Wine Quality

Kaelin Facun
Computer Science
San Diego State University
San Diego, USA
kfacun4429@sdsu.edu

Alisa Sriphet
Computer Science
San Diego State University
San Diego, USA
asriphet7210@sdsu.edu

Stella Wong
Computer Science
San Diego State University
San Diego, USA
swong3087@sdsu.edu

Abstract

In this project, we would like to define the most important features that contribute to good wine quality and also predict a wine's quality. We would like to look into the reasons as to why certain features may increase or decrease wine quality. To do this, we will be analyzing three datasets containing wines and their various features. We plan to use multiple machine learning techniques, including Support Vector Machines and Neural Networks, in order to classify the wines and reach some accurate conclusions.

Index Terms

Machine learning, neural networks, support vector machines, wine industry

I. Introduction

Machine learning allows us to find patterns within datasets, and we plan to use these techniques to accurately predict a wine's quality. We will train several machine learning models on three wine quality datasets: UCI Red Wine Quality (RWQ), UCI White Wine Quality (WWQ), and Kaggle Wine Quality (KWQ). First, we will use a Support Vector Machine (SVM) to perform linear regression (LR) on the data points. Then, we will train a Neural Network (NN) on the same data points and subsequently compare the results.

II. Dataset Analysis

We are studying three datasets covering wine quality. The key features include fixed acidity, volatile acidity, citric acid, residual sugar, chlorides, free sulfur dioxide, total sulfur dioxide, density, pH, sulfates, alcohol, and quality. We'll compare each key feature and its relationship with the quality of the wine.

Our group will correlate the key features of the wine and their quality scores. Then we will use the data to predict the wine quality based on a given key feature.

The RWQ dataset has 1599 data points, the WWQ dataset has 4898 data points, and the KWQ dataset has 1143 data points. All data sets include the same features, all of which are numerical.

A. Label

Our label is the predicted quality of the wine, which will be represented as a number between a possible range of 3 (worst) to 8 (best).

B. Features

- Fixed acidity: Fixed acids in the wine that affect the overall flavor.
- Volatile acidity: These are the gaseous acids and contribute a vinegar taste and smell to wine if not properly handled.
- Citric acid: Citrus fruit acid that contributes tanginess.
- Residual sugar: Leftover unfermented sugar.
- Chlorides: The wine's salt content.
- Free sulfur dioxide: Known as sulfites; too much of it gives wine an undesirable taste and smell.
- Total sulfur dioxide: Preservative that keeps wine fresh and kills bacteria.
- Density: The conversion rate of sugar to alcohol; sweeter wines have a higher density due to the sugar.
- pH: How acidic or basic the wine is on a scale of 0 (not acidic) to 14 (very acidic).
- Sulfates: The amount of mineral salts in the wine.
- Alcohol: The wine's alcohol percentage.
- Quality: Variable we want to accurately predict that ranges from whole numerical scores of 3 to 8.

As for preprocessing steps, we will be standardizing the feature values to fit within a numerical range of 0 to 10.

III. Model Selection

With our model selection, our group is using the SVM model to perform LR on the input, and we will also be using a NN model to compare different wine qualities based on their key features.

A. Support Vector Machine

An SVM is a supervised machine learning algorithm that groups data into multiple classes with a hyperplane. We want to maximize the distance between the hyperplane and all the classes in order to avoid overfitting (when a particular solution is too specific and not general enough to work on other datasets). In our scenario, because the data is not necessarily linearly separable, our SVM will be a nonlinear one.

1) Equations:

We'll be using the Gaussian Kernel for our nonlinear datasets.

$$K(x_i, x_j) = exp(-\frac{\|x_i - x_j\|^2}{2\sigma^2})$$

We'll be using Hinge loss to train our models.

$$Loss(h(x^{(i)}, y^{(i)})) = \max(1 - h_\theta(x^{(i)})y^{(i)}, 0)$$

B. Neural Network

A neural network is a model that uses layers, feature vectors, weights (level or priority), and labels in order to make a prediction. The layers have nodes / neurons that act as variables of functions, ultimately contributing to the prediction. Neural networks are typically used in more complicated problems.

- $x^{(i)}$ = vectors of format (feature$_1$, feature$_2$, ...), where i is a specific row in the dataset;
  RWQ example:
  $x^{(i)}$ = (fixed acidity, volatile acidity, citric acid, residual sugar, chlorides, free sulfur dioxide, total sulfur dioxide, density, pH, sulphates, alcohol);

$$x^{(0)} = (7.4, 0.70, 0.00, 1.9, 0.076, 11.0, 34.0, 0.9978, 3.51, 0.56, 9.4) \quad [1]$$

- $y^{(i)}$ = scalar value from quality column, where i is a specific row in the dataset;
  KWQ example: $y^{(0)} = 5$ [2]

We will decide what the weights should be depending on how important the features are. Features with greater priority and influence on a good quality wine will have larger weights.

1) Equations:

Features:

$$A_i^{(1)} = f^{(1)}(w_i^{(1)T}x + w_{0,i}^{(1)})$$

where A is the layer, f is the function, w is the weight vector for the ith feature, and x is the initial input vector.

$$A^{\text{\# of layers}} = NN(x; W, W_0)$$

C. Model Comparison

We decided on the SVM model for multi-class classification to compare wine quality scores from 3-8 based on their key features. This will result in 6 different classes in our model, one for each quality value. Additionally, our SVM will have soft decision boundaries, allowing the classification to be more lenient. Since wine quality is not as crucial as medical scenarios where being overly cautious would be ideal, if there are slight discrepancies, consequences would be minor. Furthermore, since our data is likely to have noise and general imperfections, using soft margins will help prevent overfitting or being overly affected by outliers. With our SVM and our NN, we should be able to predict a wine's quality accurately.

Since neural networks are better for highly complicated datasets, such as image classification, we hypothesize that an SVM will be better for our project since our values are more clearly and simply defined. In addition, our data will likely have a noticeable relationship between the features and the wine quality.

We will first train our models on each dataset individually, splitting the data into training and testing sets. Then, we plan to see if a RWQ model, for instance, will work similarly on WWQ or KWQ. The WWQ dataset has more data points than the the other datasets, so this should be a good indication as to whether our model is general and accurate enough for predicting any wine quality.

IV. Evaluation Metrics

A. Goal

Predict the quality of wine based on the features, and compare the output to the true quality values in the datasets.

B. Metrics

- Accuracy: This is the ratio of overall correct wine quality predictions that our models develop.
- Recall: This is the ratio of true positive values that are predicted correctly. Since we are doing multi-class classification in this project, the true positive values will be correctly predicted quality values for a certain class (let's say a quality of 5), and the false positive values will be predictions of 5 for qualities that are not 5.
- Precision: This represents the accuracy of our model's positive predictions.
- F1 score: This uses precision and recall in order to better define the overall accuracy of our models.

C. Plan for Complexity Evaluation

To evaluate the complexity of our SVM, we will use test datasets to test our model's generalization ability. To evaluate the complexity of our NN, we will track the number of layers in our model, with more being more complex. We will also keep track of the number of weights for our feature vectors as well as the number of nodes / neurons in each layer. Further, we will pay attention to the time and space complexity of the models. With the calculated point metrics, we will be able to see how well our models perform.

V. Model Implementation

A. Support Vector Machine

To implement our support vector machines, first we imported the various necessary libraries. This includes pandas, which creates a data frame to display the data, numpy for mathematical functions, and sklearn to help with processing and analyzing the data. Next, we loaded in our data for training and testing. We created an array of the features as input and an array to hold the labels. For our labels, we used the values from the quality column in each dataset. This resulted in 6 different labels, representing each of the classes in our multi-class classification.

Using the SVM provided by sklearn, we made our SVM with a Gaussian kernel as our data is not linearly separable. Next, we trained the model on our training data then had the model make predictions based on the test data. After comparison between the test results for UCI's red wine dataset and our predictions, the SVM provided an accuracy of 50.94%, precision of 34.32%, recall of 20.98%, and F1 of 19.33%. Some classes were not predicted in the model, which is why these values are relatively low. Since the precision is higher than the recall, that means that the SVM model has a tendency to be more careful with making positive predictions. Both the UCI white wine dataset and the Kaggle dataset performed similarly.

B. Neural Network

For the UCI Red Wine dataset, we created two neural networks. For the UCI White Wine dataset, we created five neural networks. For the Kaggle dataset, we created two neural networks. We followed the same steps for each. For the inputs and labels, we used the same ones from our SVM implementations. Then, we implemented an instance of the MLP classifier from sklearn.

For each of the neural network implementations, we experimented with different amounts of hidden layers and neurons per layer in order to see which would result in better metrics. For example, for the UCI White Wine dataset, we implemented five different neural networks. First, we created a neural network with 1 hidden layer composed of 11 neurons. We chose 11 neurons because the dataset has 11 features. Subsequently, we implemented the following neural networks: 1 hidden layer with 11 features * 5 = 55 neurons, 1 hidden layer with 11 features * 10 = 110 neurons, 3 hidden layers with 11 features * 3 = 33 neurons each, and 5 hidden layers with 11 features * 5 = 55 neurons each.

After creating each model, we trained it on the training data and then had it create predictions. Using the predictions, we then calculated the evaluation metrics.

VI. Baseline Results

A. UCI Red Wine Dataset

	Accuracy	Precision	Recall	F1
SVM	50.94%	34.32%	20.98%	19.33%
NN (3x33)	55.62%	26.44%	26.15%	26.00%
NN (1x11)	54.69%	24.54%	23.10%	21.90%

For the RWQ dataset, we achieved an accuracy of 50.94% in our SVM implementation and an accuracy of 55.62% in our first NN implementation. They are very similar, which is not what we expected. We thought a NN implementation would result in too much overfitting, as NNs are typically used for larger, more complicated datasets. In the NN for the RWQ dataset, we used 3 hidden layers with 33 neurons each, which happened to be not much different from a NN with only 1 hidden layer composed of 11 neurons—the latter resulting in an accuracy of 54.69%. The pattern seems to be that increasing the number of layers and neurons in the NN for the RWQ dataset slightly increases the accuracy. The precision metric for SVM was 34.32%, while the precision metrics for both neural network implementations were 26.44% and 24.54% respectively. As increasing the neural network complexity only slightly increased the accuracy, the same can be said for the precision metric. SVM and NN perform about the same for the recall metric. The recall metric for SVM was 20.98%, while the metrics for the neural networks were 26.15% and 23.10%. Both SVM and NN also perform similarly for the F1 score, with SVM having a score of 19.33% and NNs having scores of 26.00% and 21.90%. This makes sense because the accuracy metric is similar for both classifiers. Overall, an SVM and an NN with decent complexity will perform similarly on the red wine dataset.

B. UCI White Wine Dataset

	Accuracy	Precision	Recall	F1
SVM	44.29%	14.84%	17.30%	12.81%
NN (1x11)	48.67%	22.49%	22.45%	19.22%
NN (1x55)	46.73%	50.01%	23.95%	23.57%
NN (1x110)	50.00%	45.83%	26.21%	27.58%
NN (3x33)	48.27%	44.90%	25.71%	27.55%
NN (5x55)	50.31%	42.42%	27.60%	29.57%

For the WWQ dataset, the SVM implementation resulted in an accuracy of 44.29%. On the other hand, the NN with 1 hidden layer composed of 11 neurons computed pretty quickly, and it resulted in an accuracy of 48.67%. The 1 hidden layer 55 neurons NN ran for a bit more time and resulted in a slightly lower accuracy of 46.73%. The 1 hidden layer 110 neurons NN ran for a similar amount of time, and the accuracy went up to 50.00%. The NN with 3 hidden layers 33 neurons each had an accuracy of 48.27%. Finally, the NN with 5 hidden layers 55 neurons each had an accuracy of 50.31%. From these statistics, we can infer that a NN with an arbitrary number of hidden layers and neurons will perform with similar accuracy for a dataset of this size (4898 records). This could be because this particular dataset is not overly complex. The NN implementations had similar results for most of the other metrics. However, the SVM implementation performed slightly worse for the other metrics compared to the NN implementations. Perhaps the complexity of this WWQ dataset is more suited for a NN implementation. Overall, NNs perform slightly better for the UCI White Wine dataset.

C. Kaggle Wine Quality Dataset

	Accuracy	Precision	Recall	F1
SVM	56.33%	44.76%	26.89%	25.11%
NN (2x22)	65.94%	37.42%	36.64%	36.77%
NN (10x110)	58.95%	33.99%	36.72%	35.01%

For the KWQ dataset, the accuracy from the SVM implementation was 56.33%. The accuracy from the 2-layer NN implementation was 65.94%, while the accuracy from the 10-layer neural network implementation was 58.95%. We can see that using more layers and more neurons for this dataset is worse for the accuracy metric. The SVM has slightly better precision, while the NNs have slightly better recall and F1 scores. This means that the SVM is better at predicting positive predictions correctly, while the NN might be making more incorrect positive predictions. However, with a better F1 score, this means that the NN is slightly better overall, which is supported by the higher accuracy. Thus, a simple NN is desirable for the KWQ dataset.

D. Overall Takeaways

While the statistics for the different classification methods are very similar overall, the main difference is that the computation time of the neural network is slightly longer. This could be a reason to use SVM over NN in some situations. In other situations, NN might be better suited due to the complexity of the dataset. The overall similarity between SVM and NN performance might be due to the fact that these datasets are not too complex.

VII. Challenges

For the SVM, we had trouble deciding which classification we wanted to use, whether it'd be binary classification or multi-classification, as our wine quality scores were strictly quantitative from values 3 through 8. We eventually settled on multi-classification.

We didn't face too much difficulty in training the neural network models. The computation time was not too lengthy for any of the neural networks. This might be due to the fact that our datasets are not overly complicated or large (UCI Red Wine: 1599 records, UCI White Wine: 4898 records, Kaggle: 1143). Even the Kaggle neural network with 10 hidden layers of 110 neurons each did not run for too long.

Initially, we had implemented our models using labels of 0 and 1 (splitting the wine quality values in the middle), which resulted in pretty good metrics. Once we had switched to multi-class classification implementations, all of the metrics went down because not all the classes were being predicted. This is something we are still working on improving. We plan to experiment and alter the different parameters in our models in hopes of achieving better evaluation metrics.

VIII. Model Comparison

A. Varying Regularization Parameters

Red Wine SVM:

Regularization Parameter	Accuracy	Precision	Recall	F1	Runtime
1.0	50.94%	34.32%	20.98%	19.33%	0.54s
5.0	50.94%	33.66%	21.02%	19.04%	0.40s
0.01	40.62%	6.77%	16.67%	9.63%	0.29s

For testing purposes, we adjusted the regularization parameter C to see how it would affect our outcomes. As a default, we used C = 1.0, which led to overall the most balanced results. However, the test parameter with 5.0, which would cause the model to lean towards overfitting as it would penalize misclassification more, had the next best results. For both the initial regularization parameter and the test parameter with 5.0, they had the same accuracy of 50.94% and many of their other metrics were within 1% of each other. For many of the other metrics, such as recall, F1 and run time, the test of 5.0 actually had slightly better results, with only precision having a slightly worse performance. However, because it is overfitting, it is less generalizable for any future data. Meanwhile, the test parameter of 0.01 was generally worse across the board, with only the run time being better. Many of the other metrics ranged from a 10% to 28% deduction. This lower C value contributes to having higher regularization and having a wider, softer margin. While it generally has worse performance, it is much more general and would likely perform similarly when given new data.

Comparing the initial parameter (C = 1.0) red wine SVM to the red wine neural networks, the performance is fairly comparable in most categories with the exception of run time, in which the SVM was significantly faster, completing in almost a quarter of the time of the neural network. Both neural networks (3 hidden layers with 33 neurons each and 1 hidden layer with 11 neurons) had a higher accuracy than the SVM by roughly 4%. They additionally both had higher recall and F1 by around 2-7%. However, the precision was worse than the SVM by neary 10% for 1 hidden layer, and 8% for the three hidden layers. Because the performances between the two were fairly comparable, the SVM would be the better choice in this scenario because of its significantly faster run time, and increased interpretability. Because of the way neural networks are structured, it is essentially a black block, causing it to be less interpretable than the SVM.

Meanwhile, for the white wine data set, the neural networks far out performed the SVM. The neural networks had better performance by anywhere from 4% (accuracy) to 30% (precision) and only had an increased run time of approximately 1.5 seconds. This change could be based on the difference in the structure of the data set. While the interpretability for the SVM would still be better than the neural network, because every other category was bested by the neural network, the neural network would be better for this data.

B. Cross-testing Model Implementations

For the purposes of further analysis, we also cross-tested our model implementations between the three different datasets.

1) Cross-tested Support Vector Machines:

Trained Data Model/Test Dataset

	RWQ/WWQ	RWQ/KWQ	KWQ/RWQ
Accuracy	29.49%	56.77%	50.94%
Precision	10.47%	44.19%	35.04%
Recall	16.50%	27.12%	20.96%
F1 Score	7.79%	25.56%	19.00%
Runtime	0.21s	0.12s	0.09s

With a trained SVM (with a default regularization hyperparameter of 1.0) on the UCI Red Wine dataset, we tested it on the UCI White Wine dataset and the Kaggle dataset. The RWQ SVM model performed significantly worse on the WWQ data, as shown in the above figure. The accuracy came out to be 29.49%, in comparison to the accuracy of 44.29% when using the WWQ model on the WWQ testing data. The other metrics we calculated—precision, recall, and F1 score—also decreased greatly. This is likely because red wines and white wines place more significance on different features (e.g. residual sugar, chlorides, alcohol, etc.). When tested on the KWQ testing data, however, the RWQ SVM model performed about the same as our previous dataset-specific models, with an accuracy of 56.77%. For comparison, the KWQ model on the KWQ dataset resulted in an accuracy of 56.33%. The remaining metrics also ended up being similar, which suggests that the RWQ model works similarly as the KWQ model. Then, we tested our SVM that was trained on the KWQ dataset against the RWQ dataset. This also resulted in similar metrics: cross-tested accuracy of 50.94% vs dataset-specific accuracy of 50.94%, precision of 35.04% vs 34.32%, recall of 20.96% vs 20.98%, and F1 score of 19.00% vs 19.33%. Thus, it seems any SVM trained on the UCI Red Wine dataset will perform similarly on the Kaggle dataset, and likewise for any SVM trained on the Kaggle dataset against the UCI Red Wine dataset. Based on how the RWQ model behaved on the WWQ data, we can also infer that the KWQ model will perform worse on the WWQ data.

2) Cross-tested Neural Networks:

- Trained Data Model/Test Dataset
- RWQ NN: 3 hidden layers, 33 neurons each
- KWQ NN: 2 hidden layers, 22 neurons each

	RWQ/WWQ	RWQ/KWQ	KWQ/RWQ
Accuracy	42.35%	66.81%	59.38%
Precision	25.29%	37.95%	28.63%
Recall	22.34%	38.13%	27.96%
F1 Score	19.44%	37.87%	27.88%
Runtime	2.29s	2.29s	1.87s

When cross-testing neural networks, we chose to test the best-performing neural networks from the ones we had implemented. First, we tested a RWQ-trained NN (3 hidden layers with 33 neurons each) on the WWQ data. As with the SVM implementations, a model that was trained on the RWQ data before being tested on the WWQ data performs worse on the WWQ data. The cross-tested model resulted in an accuracy of 42.35%, whereas the dataset-specific model with the same hidden layer numerical parameters resulted in an accuracy of 48.27%. The other metrics are also lower for the cross-tested model, with precision being 25.29% vs 44.90%, recall being 22.34% vs 25.71%, and F1 score being 19.44% vs 27.55%. As we hypothesized earlier, we think this is because red wines and white wines prioritize different features when it comes to producing high-quality wine. On the other hand, the RWQ NN performed similarly as the KWQ NN (2 hidden layers with 22 neurons each) on the KWQ data. The metrics (RWQ NN vs KWQ NN) are as follows: accuracy of 66.81% vs 65.94%, precision of 37.95% vs 37.42%, recall of 36.64%, and F1 score of 37.87% vs 36.77%. Even though the numerical hidden layer implementations are slightly different, both models perform about the same. Then, we tested a KWQ-trained NN (2 hidden layers with 22 neurons each) on the RWQ data. Ordered by KWQ NN vs RWQ NN, this resulted in an accuracy of 59.38%

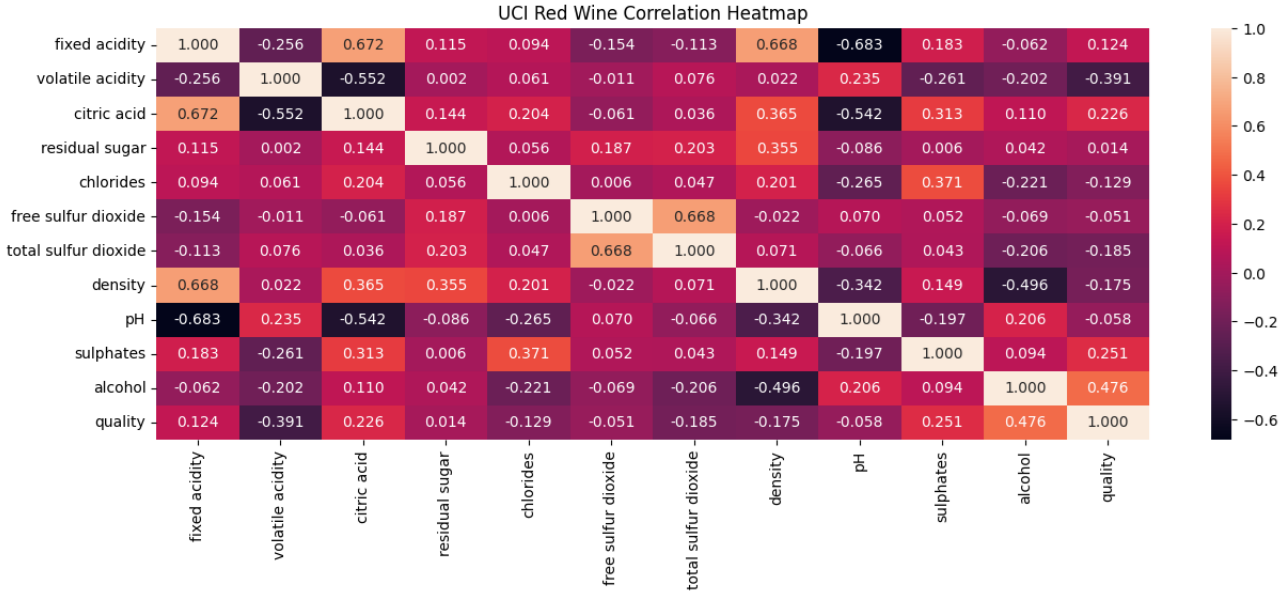
vs 55.62%, a precision of 28.63% vs 26.44%, a recall of 27.96% vs 26.15%, and an F1 score of 27.88% vs 26.00%. As with the SVM implementations, we observe a similar pattern; the RWQ NN models and KWQ NN models perform very similarly on each other's datasets. We suspect this is due to the fact that both the RWQ and KWQ datasets focus on red wine qualities, whereas the WWQ dataset focuses on white wine qualities.

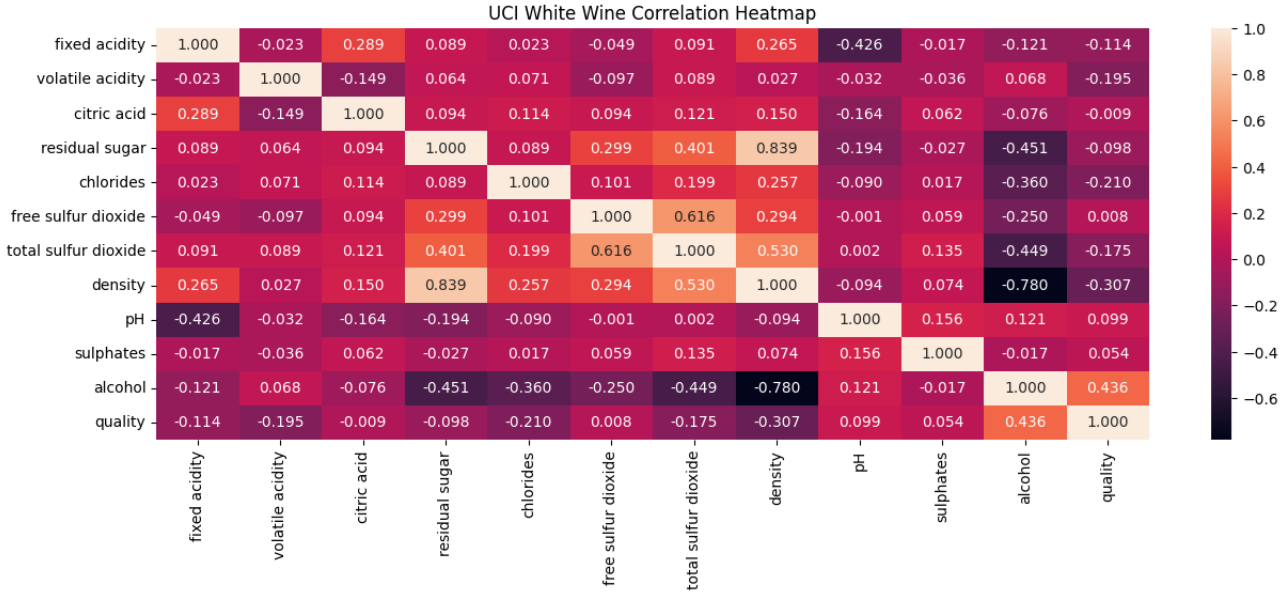
Cross-tested training times:

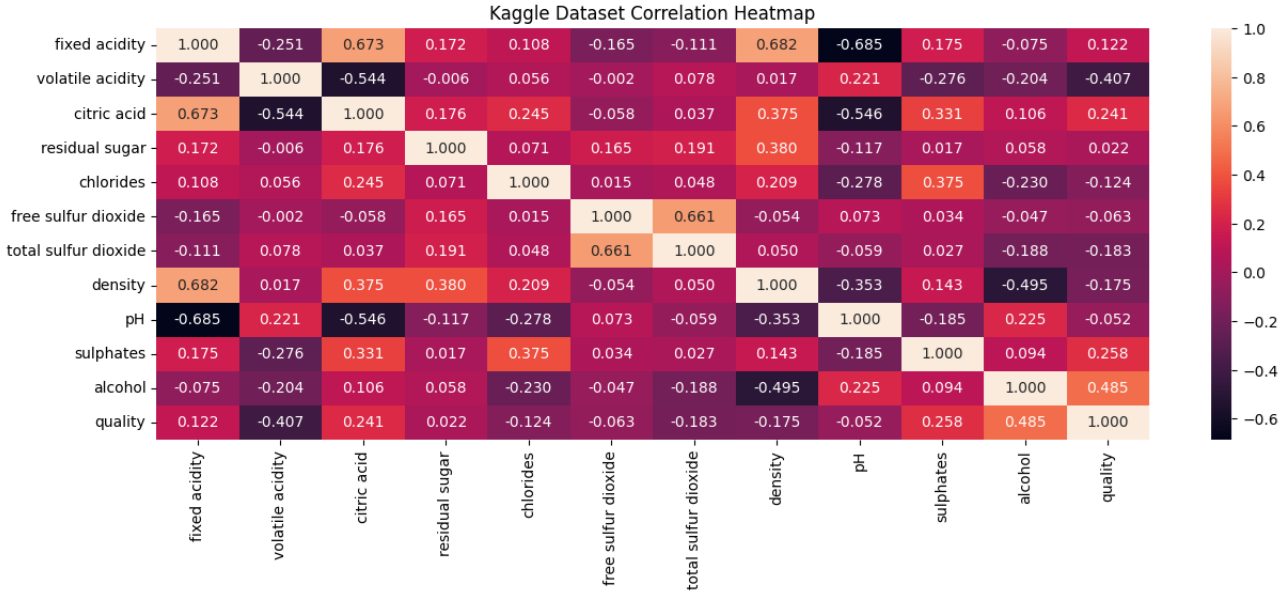
SVM training time	NN training time
0.21 seconds	2.29 seconds
0.12 seconds	2.29 seconds
0.09 seconds	1.87 seconds

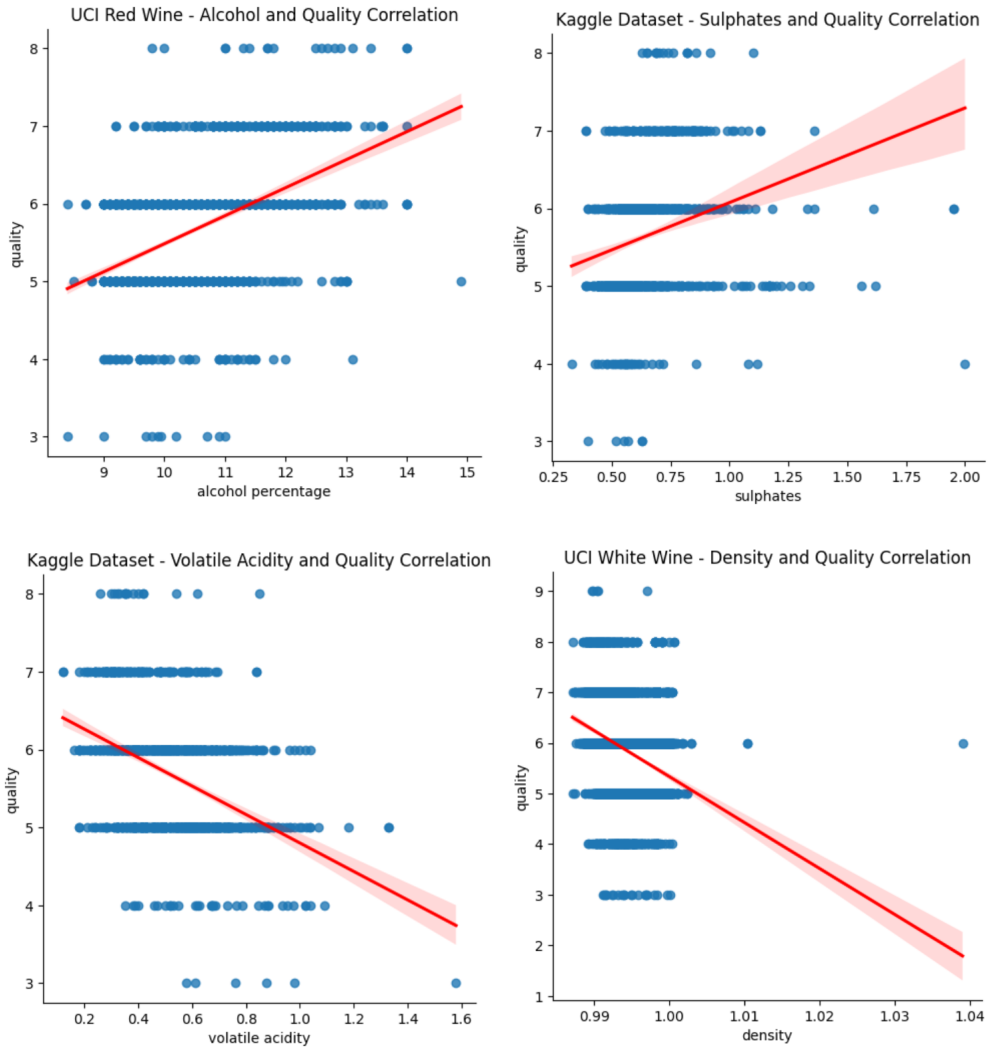
As for the training time, in this cross-testing portion of our project, the neural networks required more computational time to run. In the above table, some of the training times from an instance of cross-testing models are listed. The time varies slightly with each execution of the code, but generally, the neural network computational time is greater than the support vector machine computational time by a factor of 10-20. With a much larger dataset, this training time difference might be inefficient and undesirable for NN implementations, especially if the results between the SVM implementations and the NN implementations are not too different across the different datasets.

As for model interpretability, a NN implementation with multiple layers could add unnecessary complexity for the datasets we are using in this project.




UCI White Wine Correlation Heatmap
fixed acidity 1.000 -0.023 0.289 0.089 0.023 -0.049 0.091 0.265 -0.426 -0.017 -0.121 -0.114
volatile acidity -0.023 1.000 -0.149 0.064 0.071 -0.097 0.089 0.027 -0.032 -0.036 0.068 -0.195
citric acid 0.289 -0.149 1.000 0.094 0.114 0.094 0.121 0.150 -0.164 0.062 -0.076 -0.009
residual sugar 0.089 0.064 0.094 1.000 0.089 0.299 0.401 0.839 -0.194 -0.027 -0.451 -0.098
chlorides 0.023 0.071 0.114 0.089 1.000 0.101 0.199 0.257 -0.090 0.017 -0.360 -0.210
free sulfur dioxide -0.049 -0.097 0.094 0.299 0.101 1.000 0.616 0.294 -0.001 0.059 -0.250 0.008
total sulfur dioxide 0.091 0.089 0.121 0.401 0.199 0.616 1.000 0.530 0.002 0.135 -0.449 -0.175
density 0.265 0.027 0.150 0.839 0.257 0.294 0.530 1.000 -0.094 0.074 -0.780 -0.307
pH -0.426 -0.032 -0.164 -0.194 -0.090 -0.001 0.002 -0.094 1.000 0.156 0.121 0.099
sulphates -0.017 -0.036 0.062 -0.027 0.017 0.059 0.135 0.074 0.156 1.000 -0.017 0.054
alcohol -0.121 0.068 -0.076 -0.451 -0.360 -0.250 -0.449 -0.780 0.121 -0.017 1.000 0.436
quality -0.114 -0.195 -0.009 -0.098 -0.210 0.008 -0.175 -0.307 0.099 0.054 0.436 1.000
fixed acidity
volatile acidity
citric acid
residual sugar
chlorides
free sulfur dioxide
total sulfur dioxide
density
pH
sulphates
alcohol
quality
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4
−0.6


Kaggle Dataset Correlation Heatmap
fixed acidity 1.000 -0.251 0.673 0.172 0.108 -0.165 -0.111 0.682 -0.685 0.175 -0.075 0.122
volatile acidity -0.251 1.000 -0.544 -0.006 0.056 -0.002 0.078 0.017 0.221 -0.276 -0.204 -0.407
citric acid 0.673 -0.544 1.000 0.176 0.245 -0.058 0.037 0.375 -0.546 0.331 0.106 0.241
residual sugar 0.172 -0.006 0.176 1.000 0.071 0.165 0.191 0.380 -0.117 0.017 0.058 0.022
chlorides 0.108 0.056 0.245 0.071 1.000 0.015 0.048 0.209 -0.278 0.375 -0.230 -0.124
free sulfur dioxide -0.165 -0.002 -0.058 0.165 0.015 1.000 0.661 -0.054 0.073 0.034 -0.047 -0.063
total sulfur dioxide -0.111 0.078 0.037 0.191 0.048 0.661 1.000 0.050 -0.059 0.027 -0.188 -0.183
density 0.682 0.017 0.375 0.380 0.209 -0.054 0.050 1.000 -0.353 0.143 -0.495 -0.175
pH -0.685 0.221 -0.546 -0.117 -0.278 0.073 -0.059 -0.353 1.000 -0.185 0.225 -0.052
sulphates 0.175 -0.276 0.331 0.017 0.375 0.034 0.027 0.143 -0.185 1.000 0.094 0.258
alcohol -0.075 -0.204 0.106 0.058 -0.230 -0.047 -0.188 -0.495 0.225 0.094 1.000 0.485
quality 0.122 -0.407 0.241 0.022 -0.124 -0.063 -0.183 -0.175 -0.052 0.258 0.485 1.000
fixed acidity
volatile acidity
citric acid
residual sugar
chlorides
free sulfur dioxide
total sulfur dioxide
density
pH
sulphates
alcohol
quality
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4
−0.6



Additionally, we created some correlation heatmaps to analyze the relationships between the various features in the datasets. We used the Pearson correlation coefficient to measure potential linear relationships. The primary feature to focus on in these heatmaps is the quality feature, since this is our target variable. In all three datasets and their corresponding heatmaps, the alcohol feature has the highest positive linear correlation with wine quality, averaging 0.466. This demonstrates medium strength of correlation. Some other notable features that have some linear influence on red wine quality are citric acid and sulphates, with correlation coefficient values averaging 0.2335 and 0.2545, respectively. Although there is some positive correlation between these features and wine quality for red wines, the same can't be said for white wines, who display almost no linear correlation between these features and wine quality. A notable feature that shows low negative linear correlation for red wines is volatile acidity, with an average correlation coefficient of -0.399. White wines, on the other hand, place slightly more significance on density and chlorides, with negative correlation coefficient values of -0.307 and -0.210, respectively. All of the remaining features, across the three datasets, display little to no linear correlation in regards to overall wine quality. Included above are some scatter plots with regression lines that display some of these correlations.

IX. Conclusions

Throughout our Wine Quality Prediction project, the comparison between SVM and neural network models highlights key trade-offs in performance, interpretability, and efficiency across the datasets. For the red wine datasets, the SVM model demonstrated faster runtimes and greater interpretability, making it a strong choice despite slightly lower accuracy and recall compared to the neural network models. Conversely, for the white wine dataset, neural networks consistently outperformed SVMs in almost every metric, indicating their suitability for datasets with more complex relationships between features and quality.

Cross-testing revealed that models trained on red wine datasets performed better on each other than on the white wine dataset, emphasizing the differing feature priorities between red and white wine quality predictions. This aligns with real-world

practices, as red and white wines differ in their production and aging processes, leading to variations in quality-grading criteria.

While neural networks provide higher accuracy and recall for some datasets, their computational complexity and reduced interpretability pose challenges. Neural networks rely on multiple layers of computations, making them harder to train and understand compared to SVMs. In contrast, SVMs, with their simpler decision boundaries and faster runtimes, are more suitable for tasks requiring transparency and efficiency. Ultimately, SVMs are a strong choice for wine quality prediction where interpretability and speed are crucial, while neural networks are better suited for datasets where complexity and accuracy take priority.

X. Team Member Contributions

Kaelin Facun implemented the evaluation metric computations for the neural networks and wrote the README. Alisa Sriphet implemented the support vector machines, their respective evaluation metrics, and training time calculations for all the models. Stella Wong implemented the neural networks, cross-tested the models between the datasets, and created the visuals (correlation heatmaps and scatter plots with regression lines). We all worked on the written report together.

References

[1] P. Cortez, A. Cerdeira, F. Almeida, T. Matos, and J. Reis. "Wine Quality," UCI Machine Learning Repository, 2009. [Online]. Available: https://doi.org/10.24432/C56S3T.

[2] M. Yasser. "Wine Quality Dataset," Kaggle, 2022. [Online]. Available: https://www.kaggle.com/datasets/yasserh/wine-quality-dataset.