Wine quality data analysis with Neural Networks

- **Dataset**

  The Wine Quality (red) dataset in https://archive.ics.uci.edu/datasets is used in this project [1]. The dataset is related to red variant of the Portuguese “Vinho Verde” wine. It consists of various physicochemical properties of red wine and their associated quality ratings.

- **Objective**

  The goal of this analysis is to model wine quality based on physicochemical tests using a Neural Network model.

- **Input Variables**

  The wine quality is analyzed based on 11 variables,

  1. Fixed Acidity: The fixed acidity of the wine.
  2. Volatile Acidity: The volatile acidity of the wine.
  3. Citric Acid: The citric acid content of the wine.
  4. Residual Sugar: The residual sugar content of the wine.
  5. Chlorides: The chloride content of the wine.
  6. Free Sulfur Dioxide: The free sulfur dioxide content of the wine.
  7. Total Sulfur Dioxide: The total sulfur dioxide content of the wine.
  8. Density: The density of the wine.
  9. pH: The pH value of the wine.
  10. Sulphates: The sulphate content of the wine.
  11. Alcohol: The alcohol content of the wine.

- **Output variable**

  Quality: The quality rating of the wine scored between 3 and 8, where score of 8 representing the highest quality.

0.1 Exploratory data analysis and data prepossessing

The dataset have no missing values.

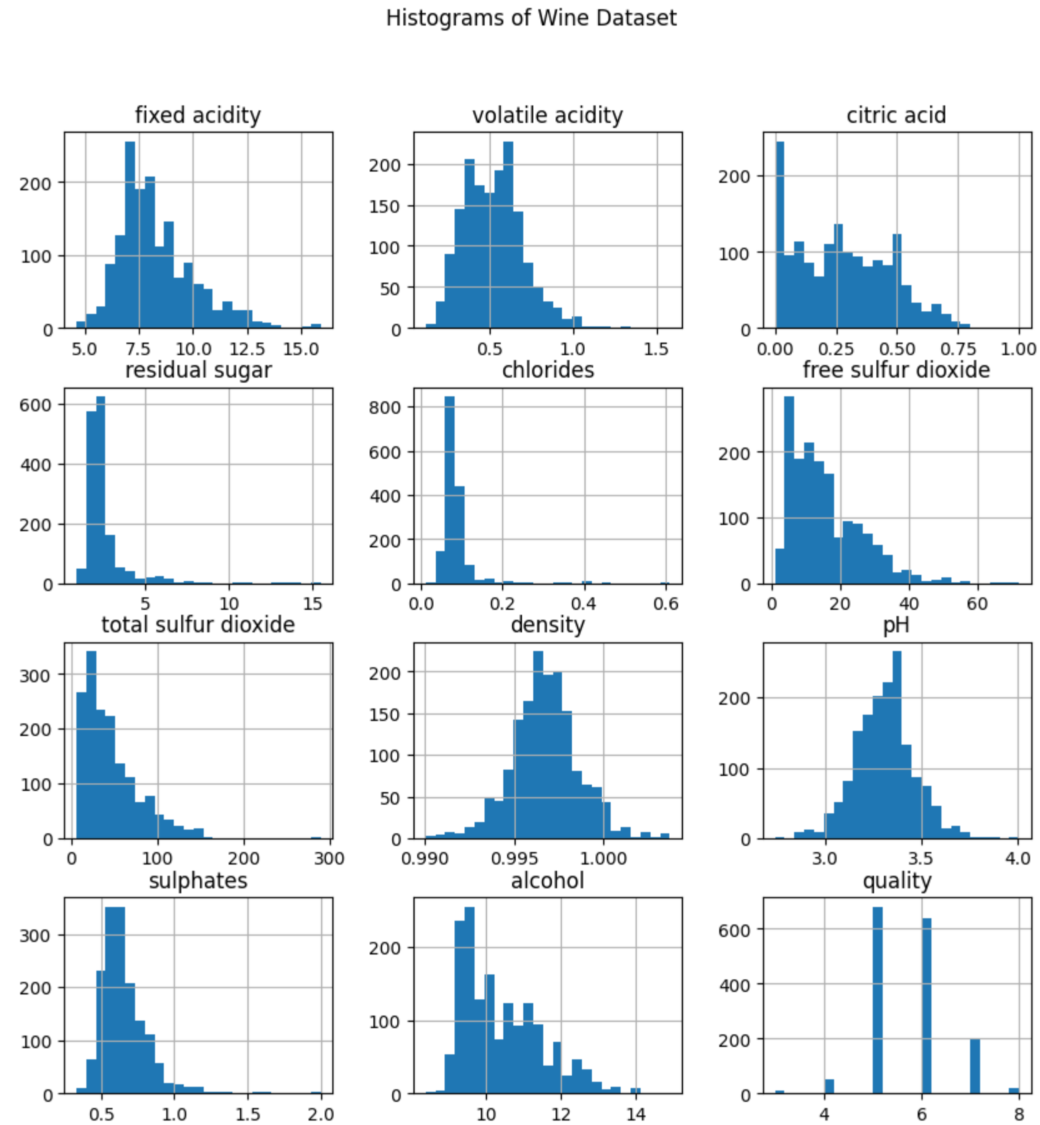

Figure 1: Histrograms of attributes

Figure 1 shows the distributions of all the variables. It can be seen that most of the variables are normally distributed with some skewness. Also it can be seen that the variables are having different scales. Therefore, all the variables are standardized and scaled.

The class conditional distributions of the predictors were examined via boxplots (Figure 2). It can be seen that the class conditional distributions for almost all predictors differ across the response classes and therefore, these predictors are useful to predict 'quality'. Also 'quality' is a good distribution for all other features as we can observe that quality is a distribution which is available in every feature. So, we can use the variable 'quality' to distinguish between these features.

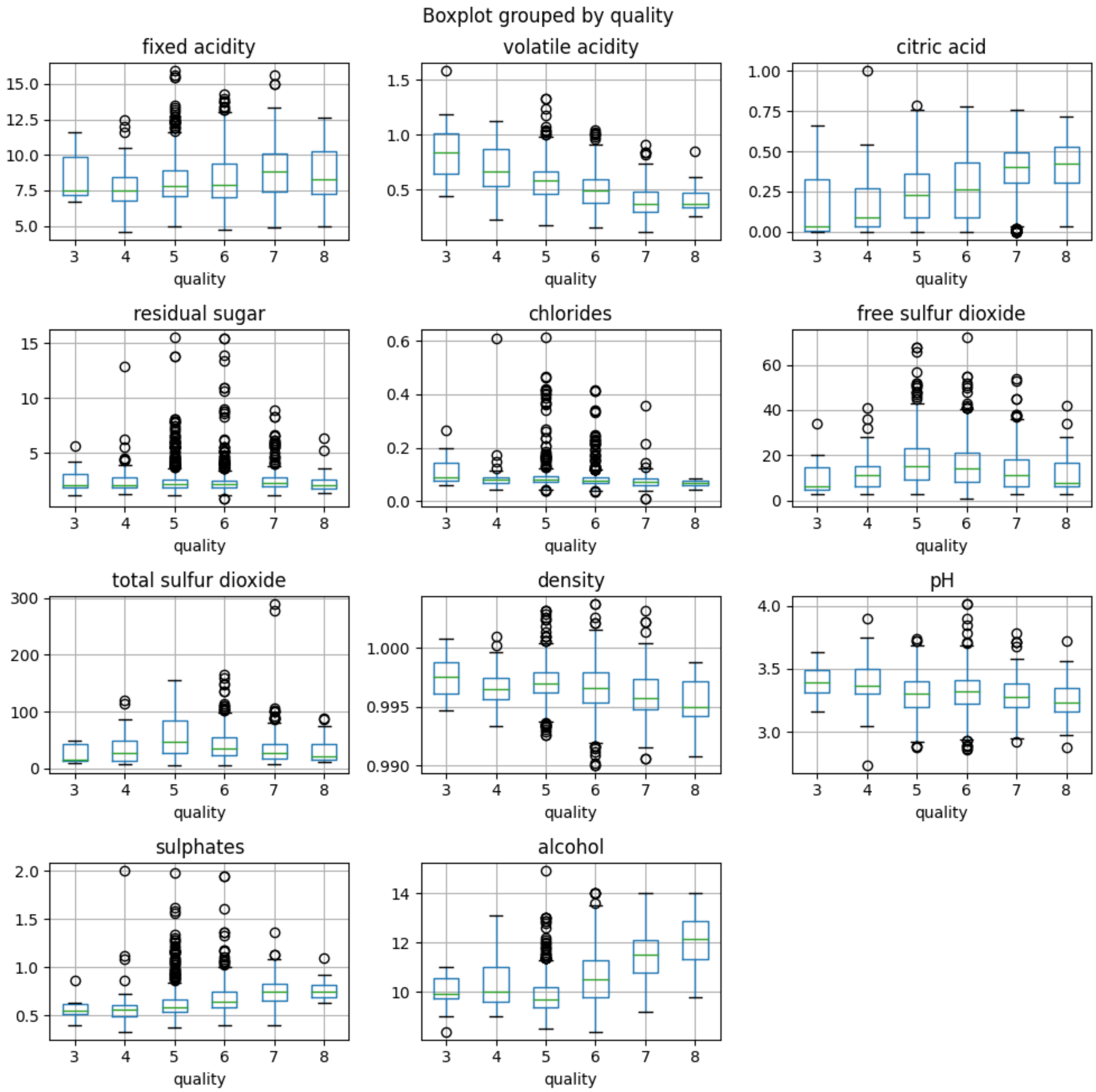

Figure 2: Class conditionals Boxplots

A heat map (figure 3) was drawn to analyze the correlation between wine quality and other predictor variables. It can be seen that the variable 'Alcohol' is moderately correlated to quality and them followed by 'Sulphates' and 'Volatile acidity'. It's likely that these variables are also the most important features in categorizing the 'quality' of wine. Also it can be seen that 'fixed acidity' is highly positively correlated with 'Citric acid' and 'density' and highly negatively correlated with 'pH'.

A Random Forest feature selection method (figure 4) was performed to see which variables are relevant and most important when predicting the 'quality'. Variables 'alcohol' seems to be the most important predictor followed by 'sulfates' and 'volatile acidity'. This supports the conclusions drawn from figure 3 as well. We decided to keep all the variables as data set is large enough.

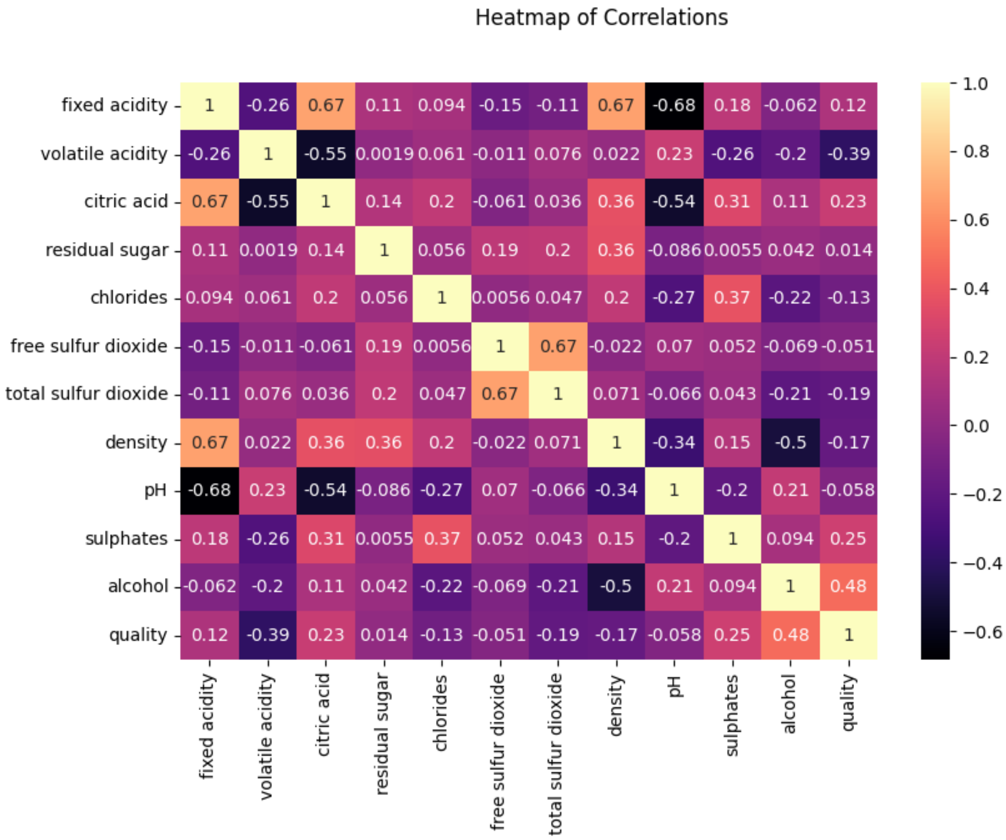

Figure 3: Heatmap of attributes

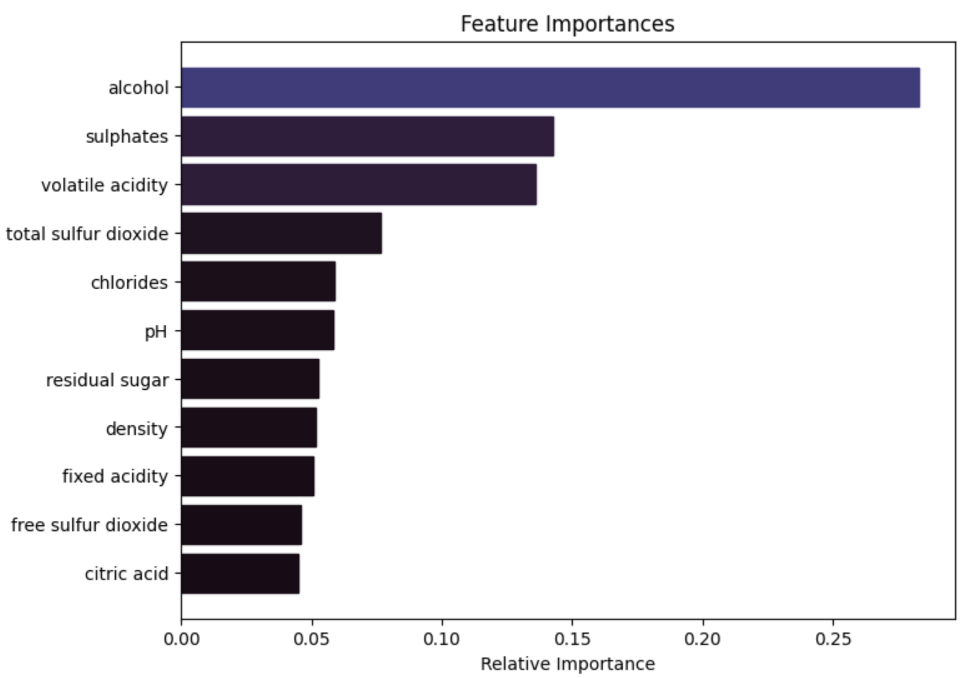

Figure 4: Feature importances

0.2 Training Neural Network

A neural network is built with an input layer with 11 neurons, 2 hidden layers with 16 and 8 neurons respectively and an output layer with 6 output units. Instead of depending only on the basic gradient descent algorithm, gradient descent with momentum optimizer is used to update the weights [2].

$$w_{ji} = w_{ji}^{old} - \alpha m_t$$

where

$$m_t = \beta m_t^{old} + (1-\beta)\frac{\partial E_d}{\partial w_{ji}}$$

and β is considered as its default value, 0.9.

	activation_function	learning_rate	epochs	train_accuracy	test_accuracy
0	sigmoid	0.1	100	0.387803	0.44375
1	sigmoid	0.1	500	0.426896	0.421875
2	sigmoid	0.1	1000	0.426896	0.421875
3	sigmoid	0.05	100	0.13448	0.084375
4	sigmoid	0.05	500	0.387803	0.44375
5	sigmoid	0.05	1000	0.426896	0.421875
6	sigmoid	0.01	100	0.576231	0.584375
7	sigmoid	0.01	500	0.387803	0.44375
8	sigmoid	0.01	1000	0.55903	0.578125
9	tanh	0.1	100	0.00625489	0.00625
10	tanh	0.1	500	0.00625489	0.00625
11	tanh	0.1	1000	0.00625489	0.00625
12	tanh	0.05	100	0.00625489	0.00625
13	tanh	0.05	500	0.00625489	0.00625
14	tanh	0.05	1000	0.00625489	0.00625
15	tanh	0.01	100	0.00625489	0.00625
16	tanh	0.01	500	0.00625489	0.00625
17	tanh	0.01	1000	0.00625489	0.00625
18	relu	0.1	100	0.00625489	0.00625
19	relu	0.1	500	0.00625489	0.00625
20	relu	0.1	1000	0.00625489	0.00625
21	relu	0.05	100	0.00625489	0.00625
22	relu	0.05	500	0.00625489	0.00625
23	relu	0.05	1000	0.00625489	0.00625
24	relu	0.01	100	0.00625489	0.00625
25	relu	0.01	500	0.00625489	0.00625
26	relu	0.01	1000	0.00625489	0.00625

Table 1: Training and Test Accuracies with the optimizer

	activation_function	learning_rate	epochs	train_accuracy	test_accuracy
0	sigmoid	0.1	100	0.387803	0.44375
1	sigmoid	0.1	500	0.681001	0.646875
2	sigmoid	0.1	1000	0.42846	0.421875
3	sigmoid	0.05	100	0.129789	0.15625
4	sigmoid	0.05	500	0.426896	0.421875
5	sigmoid	0.05	1000	0.562158	0.4875
6	sigmoid	0.01	100	0.600469	0.5625
7	sigmoid	0.01	500	0.666145	0.640625
8	sigmoid	0.01	1000	0.731822	0.609375
9	tanh	0.1	100	0.095387	0.1
10	tanh	0.1	500	0.00625489	0.00625
11	tanh	0.1	1000	0.0883503	0.090625
12	tanh	0.05	100	0.18061	0.16875
13	tanh	0.05	500	0.150899	0.165625
14	tanh	0.05	1000	0.131353	0.1
15	tanh	0.01	100	0.46208	0.496875
16	tanh	0.01	500	0.364347	0.384375
17	tanh	0.01	1000	0.36122	0.36875
18	relu	0.1	100	0.00625489	0.00625
19	relu	0.1	500	0.00625489	0.00625
20	relu	0.1	1000	0.00625489	0.00625
21	relu	0.05	100	0.00625489	0.00625
22	relu	0.05	500	0.00625489	0.00625
23	relu	0.05	1000	0.00625489	0.00625
24	relu	0.01	100	0.00625489	0.00625
25	relu	0.01	500	0.00625489	0.00625
26	relu	0.01	1000	0.00625489	0.00625

Table 2: Training and Test Accuracies without the optimizer

The Neural network is trained on 80% of data and tested on 20% of data under 3 different activation functions *sigmoid*, *tanh* and *ReLu*. Also the model trained under different combinations of epochs and the learning rates and results obtained are summarized in table 1 and table 2.

Table 1 shows the results obtained when gradient descent with momentum optimizer is applied whereas table 2 shows the results obtained when basic gradient descent (without momentum optimizer) is applied.

In both the cases it can be seen that the neural network performs better when *sigmoid* activation is used as in both those cases the training accuracy and the test accuracy is higher. But for this dataset, the results obtained when basic gradient descent (without momentum optimizer) is applied is better than when gradient descent with momentum optimizer is applied. In both these cases the best learning accuracy as well as the test accuracy was obtained with the *sigmoid* activation where the learning rate is 0.01 and the number of epochs is 1000.

References

[1] P. Cortez, A. Cerdeira, F. Almeida, T. Matos and J. Reis, "Modeling wine preferences by data mining from physicochemical properties". Decision support systems, vol. 47, pp. 547–553, https://archive.ics.uci.edu/dataset/186/wine+quality.

[2] R. Karim, "10 Stochastic Gradient Descent Optimisation Algorithms + Cheatsheet". Towards Data Science, https://towardsdatascience.com/10-gradient-descent-optimisation-algorithms-86989510b5e9.