Assignment 2 – Backpropagation

Viraj Bangari 10186046

[Viraj.bangari@queensu.ca](mailto:Viraj.bangari@queensu.ca)

# Instructions and Code

Use the command make all to compile and run executable to train and validate the neural network. The files MulticlassNeuralNetwork.cpp/.hpp contain the source code, and main.cpp contains reading the files and creating and output. The file preprocessing.ipynb contains a jupyter notebook that shows the preprocessing steps. The file training.csv contains the training data, and testing.csv contains the testing data.

# Data preprocessing and Split

The raw data contains 11 features and three possible classification values based on quality level. The frequency of the classification values were plotted on a histogram.

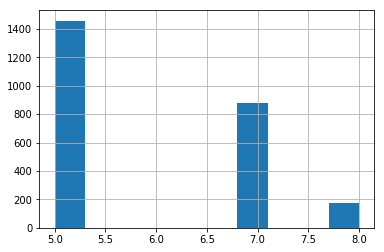


Figure 1 - Histogram of quality and number of occurrences

The data was sampled so that there was an even distribution of each class.

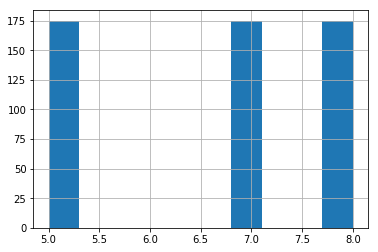
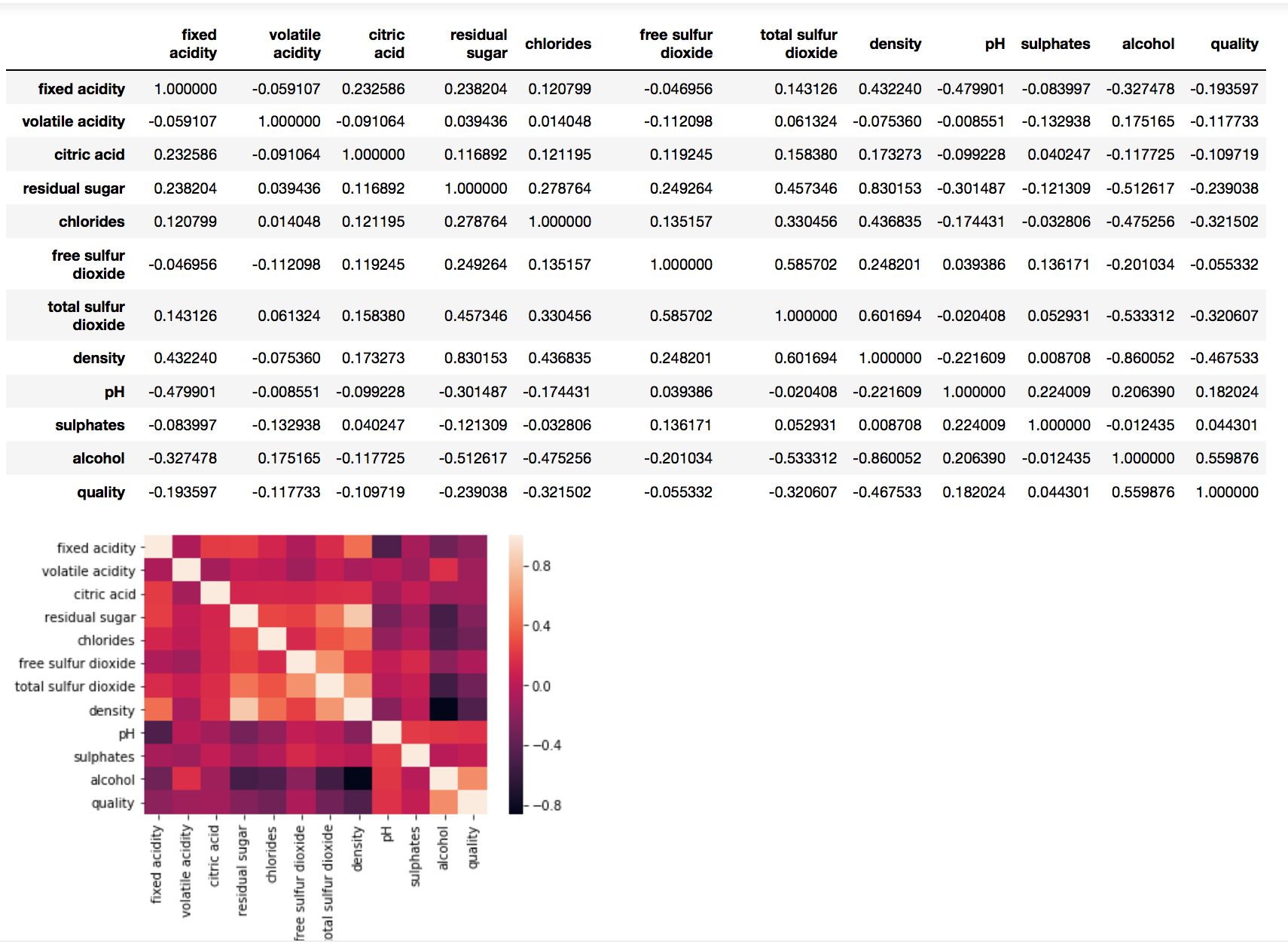
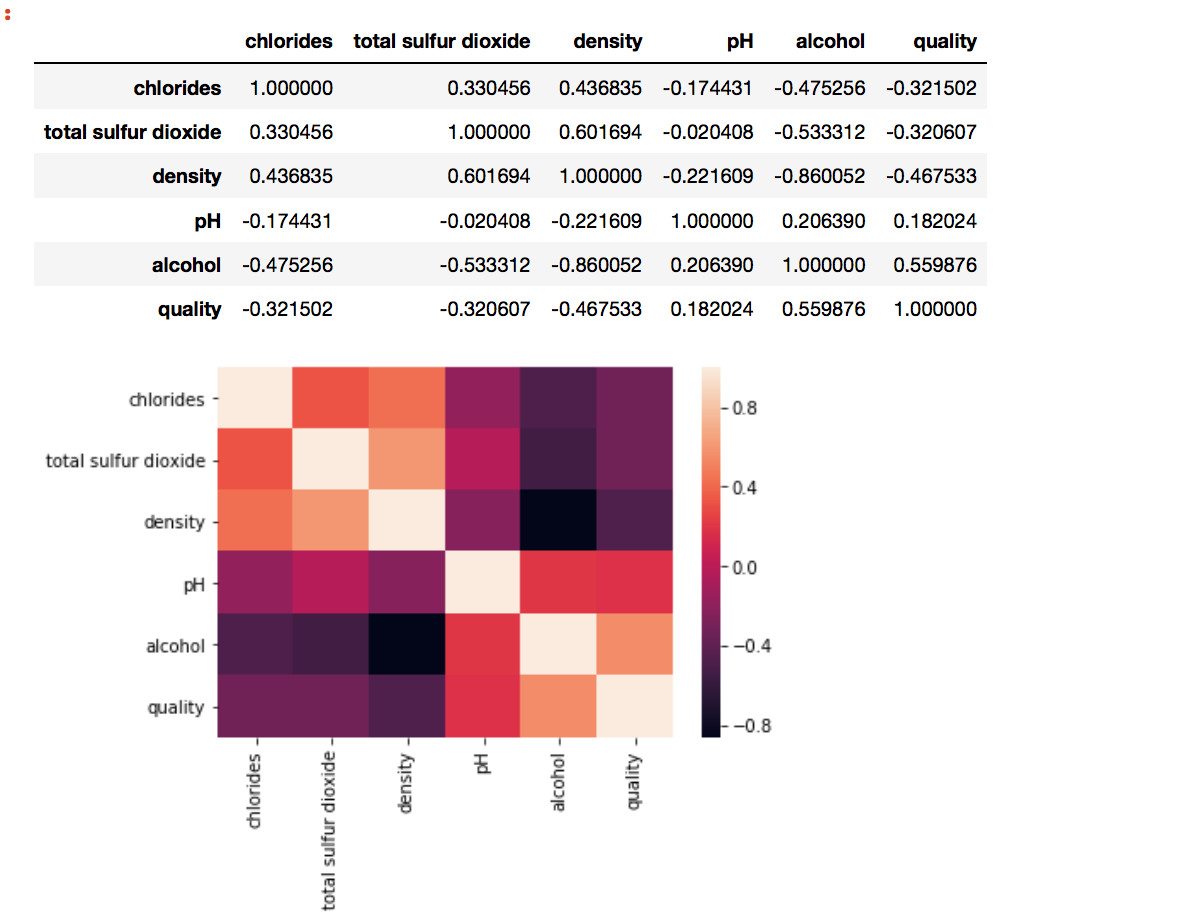


Figure 2 - Histogram of evenly sampled wine data

A correlation matrix was made with all the evenly sampled data.



The features that had a low correlation with quality were dropped from the dataset. The features removed were citric acid, fixed acidity, volatile acidity, sulphates and free sulphur dioxides. Residual sugars dropped because it had a very high correlation with alcohol, meaning it was redundant.



Note, that some of the features left are still correlated with each other. This is an issue with the data set, since not all variables are completely independent of each other. After dropping the column, all the data was normalized from 0 to 1 using min-max. The top 70% of the data was used for training, 15% was used for validation and the remaining 15% was used for testing. The data was shuffled before the split to ensure that the data distribution was even.

# Neural Network and Results

The NN model built has 5 inputs nodes, a hidden layer with 4 nodes (one for each feature except density since it correlated with the other features), a hidden layer with 10 nodes and an output layer of 3 nodes. The node output function was a sigmoid, and its termination condition was to stop after 100 epochs with a batch size of 30. The learning rate is 0.05 and the momentum parameter is 0.1. The epsilon threshold was 0.2.

The initial weights for the NN were:

Layer: 0

Inputs: 6 Outputs: 4

-0.21 -0.01 -0.65 -0.62 -0.57 -0.36 -0.51 -0.38 -0.59 -0.15 -0.06 -0.04 -0.53 -0.99 -0.48 -0.1 -0.46 -0.8 -0.45 -0.16 -0.45 -0.53 -0.16 -0.28

Layer: 1

Inputs: 5 Outputs: 10

-0.71 -0.01 -0.05 -0.07 -0.7 -0.39 -0.44 -0.36 -0.54 -0.38 -0.47 -0.66 -0.12 -0.95 -0.48 -0.98 -0.49 -0.28 -1 -0.88 -0.07 -0.57 -0.25 -0.49 0 -0.22 -0.71 -0.91 -0.34 0 -0.32 -0.05 -0.67 -0.57 -0.97 -0.74 -0.84 -0.96 -0.35 -0.63 -0.51 -0.68 -0.18 -1 -0.16 0 -0.88 -0.14 -0.8 -0.62

Layer: 2

Inputs: 11 Outputs: 3

-0.39 -0.01 -0.96 -1 -0.54 -0.66 -0.01 -0.85 -0.68 -0.12 -1 -0.49 -0.28 -0.95 -0.14 -0.5 -0.79 -0.47 -0.05 -0.57 -0.67 -0.27 -0.85 -0.37 -0.63 -0.32 -0.07 -0.96 -0.18 -0.92 -0.93 -0.18 -0.65

After training, the weights were adjusted to:

Layer: 0

Inputs: 6 Outputs: 4

-1.36591 -0.669949 -1.59106 -0.165411 2.86216 -2.7524 -4.32153 -1.38162 -1.12296 2.03611 11.0653 -5.54641 -1.82854 -1.61024 -0.363665 0.420453 3.14593 -2.84571 -4.23026 -1.12345 -0.941896 1.89726 11.0486 -5.7618

Layer: 1

Inputs: 5 Outputs: 10

-1.18358 -1.02606 -0.297763 -1.11638 -1.99181 -0.500961 -1.97622 -0.478173 -2.08818 -1.15003 -0.0635111 -2.52897 -0.209047 -2.74265 -0.956086 -0.725591 -1.85406 -0.190005 -2.38455 -1.07724 -0.0418034 -2.65282 -0.492068 -2.44963 -0.933282 -0.146037 -2.41301 -1.05598 -2.01067 -1.07199 -0.476883 -1.3511 -0.848554 -1.86537 -1.40462 -0.70046 -2.00118 -0.903707 -1.56985 -1.19898 -0.33514 -2.26996 -0.203588 -2.59708 -0.971507 0.215425 -2.71408 -0.28603 -2.5745 -0.96086

Layer: 2

Inputs: 11 Outputs: 3

-1.22532 -1.48606 -1.82901 -1.66086 -1.65412 -1.58759 -1.35123 -1.55504 -1.70212 -1.76123 0.086156 0.17781 -0.0732146 -0.873151 0.074731 -0.626095 -0.563746 -0.0793897 0.196594 -0.412188 -0.843524 -0.137475 0.964904 1.84642 1.86386 1.80316 1.89505 1.86511 1.68475 1.67543 1.83965 1.90045 -2.41082

The following confusion matrices and precision/recall values were generated for each class:

n = 105

TN FP

FN TP

Class 0 (Quality = 5)

55 11

21 18

Precision: 0.62069 Recall: 0.461538

Class 1 (Quality = 7)

21 9

33 56

Precision: 0.861538 Recall: 0.629213

Class 2 (Quality = 8)

33 52

31 7

Precision: 0.118644 Recall: 0.184211

There seems to be overfitting between class 1 and class 2. This is because class 1 tends to have a very high variance. As an example, this was the data distribution for alcohol (normalized). This can be seen with the high precision for class 1, but relatively lower recall.

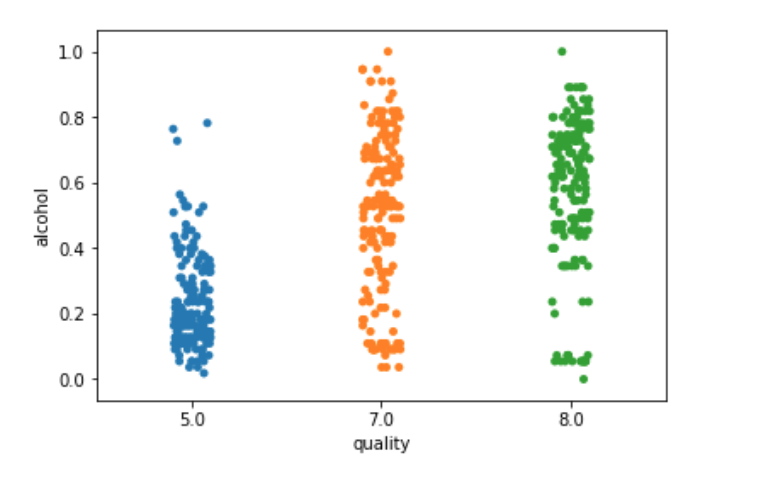


Figure 3 - Data distribution for alcohol and quality

# List of predicted vs expected values (for training data)

See training\_results.txt