Sequential Neural Network to Predict Wine Quality

Christopher Brantner

Abstract

The use of neural networks to analyze data sets has widespread use for predicting outcomes of an independent feature. In this report, we will analyze two data sets of white and red wine quality given eleven dependent features. These features were ran through a sequential neural network to make predictions upon the wine quality using the Keras framework and TensorFlow. The neural network was then modeled again with dropout between layers. The difference between these two models will be analyzed in the following paper.

Introduction

Data analysis is a fundamental problem in the world today, and artificial intelligence offers a variety of solutions to this problem. Deep learning, a sub-field in machine learning and AI, relies on the technique of neural networks to attempt to reach a solution to this problem. Neural networks are models that were designed to act in a similar (albeit grossly simplified) version of neurons in brains. Broadly, these networks are composed of nodes (known as *neurons*) which take a linear combination of dependent variables, or features, along with coefficients (*weights*) and an intercept (*bias*) and apply different functions to these terms. In the simplest case, a neural network is comprised of an input layer, a hidden layer, and an output later. The goal of analysis is to train the model using a subset of data to make accurate predictions upon new values. Some metrics to gauge efficacy of the model are precision and recall, along with various other metrics. The capability of the neural network is often correlated to its overall size. A complex neural network, with more neurons and hidden layers, can apply many more parameters through the functions, allowing for more fine tuned analysis of data. We will rely on this fact in our exploration of features which effect wine quality. A sequential neural network is a neural network whose layers are sequential; the input layer is fed to each subsequent hidden layer which is then fed to the output layer. We will utilize dense layers with a rectifier activation function. An activation function is one which is applied to each node, the range of the previous as its domain, and is important in "training" the model. The activation function used in this application was a rectified linear unit, or ReLU, defined as follows:

$$f(x) := \max(0, x) = \begin{cases} x & \text{if } x > 0, \\ 0 & \text{otherwise} \end{cases}$$

An important problem with data analysis using neural networks is known as overfitting. Networks are trained on one data set and then predict on another (test set). Overfitting can occur when the model is fit too tightly to the training set, leaving a wide margin of error when analyzing the never-before-seen set. To compensate for this, we will investigate using a method known as dropout, a regularization method. Regularization is meant to prevent complex co-adaptions of the model on the training set, and are generally categorized into two groups: strong dilution, and weak dilution. Dropout, the method we will use, is a type of weak dilution. In the dropout technique, a row of weights are dropped from the vector matrix being analyzed. This can be thought of as "pruning" the weights, and prevents one feature from having too much of an effect on the model, which results in overfitting.

Methodologies

Overview of Wine Data Sets

We will be conducting analysis on two data sets, white wine and red wine. Each data set consists of 11 independent features and one dependent feature. The goal of analysis will be to predict the dependent feature given only the independent features. The dependent feature of interest is wine quality, given on a scale from three to eight. The independent features are as follows:

- Fixed acidity
- Volatile acidity
- Citric acid
- Residual sugar
- Chlorides
- Free sulfur dioxide
- Total sulfur dioxide
- Density
- pH
- Sulphates
- Alcohol

Each data set contains 1599 entries, and each will be split 70% training and 30% testing.

Descriptive Statistical Analysis of Data Sets

In processing data for analysis, interpretation of general spread and distribution of the set is important, as well as identifying any problematic aspects of the data. Much of data science work is spent in so-called "preprocessing" of data. This involves formatting the set into a usable format, as well as removing any potential outliers or missing data (*NaN values*) from the set. The following figure shows the spread of each feature of interest, and they appear roughly normally distributed (as to be expected with a large sample size). This will be important to consider when computing descriptive statistics of the sets. The final histogram is empty for formatting purposes only.

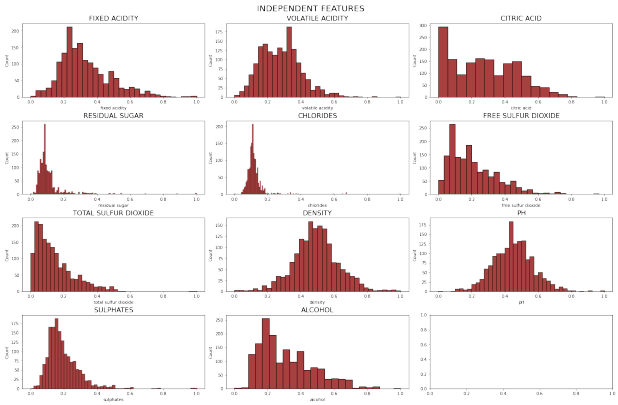

Figure 1: *Distributions for red wine features*

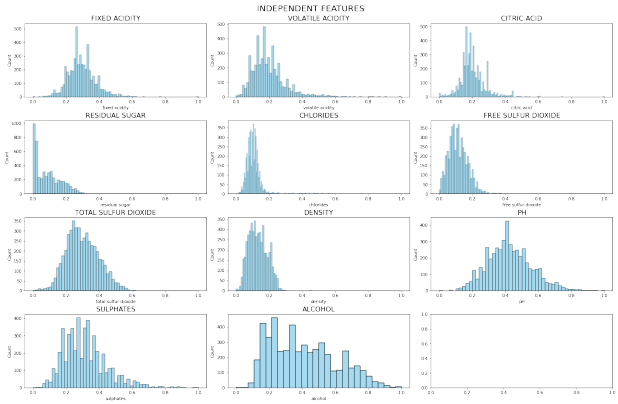

Figure 2: *Distributions for white wine features*

As can be seen in 1, 2, the features are roughly normal, and so descriptive statistics using mean and standard deviation will be useful to analyze. The following tables provide some meaningful insight as to the values of the data sets.

The complete table showed that the data was roughly normally distributed. Next, analysis of correlation between features was completed, in the form of a heatmap, to see which values were closely related. This helps to gauge which of the independed features may be related.

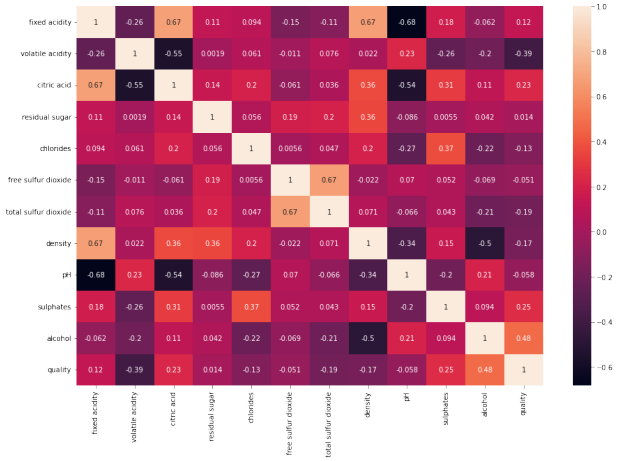

Figure 3: *A heatmap for correlation of features in the red wine data set*

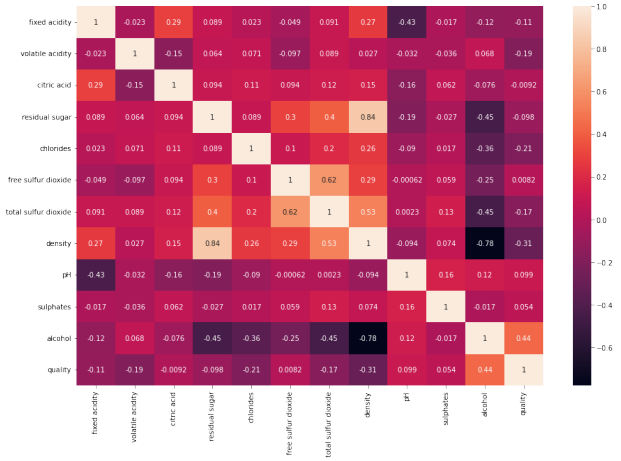

Figure 4: *A heatmap for correlation of features in the white wine data set*

In this case, lighter values indicate features which are more closely correlated. From this, we can see that fixed acidity and citric acid are closely correlated, as well as free sulfur dioxide and total sulfur dioxide. From this, we likely could remove one of these features as they would have a similar effect on the output predictions. However, we will leave them in for this analysis.

Further Preprocessing

Now that preliminary preprocessing has been completed, since no large outliers were found in the data set and they appear to follow a normal distribution,

	Fixed Acidity	Volatile Acidity	Citric Acid	Residual Sugar	Chlorides	Free sulfur dioxide	Total Sulfur D
mean	0.33	0.28	0.27	0.11	0.13	0.21	
std	0.15	0.12	0.19	0.10	0.08	0.15	
min	0.00	0.00	0.00	0.00	0.00	0.00	
25%	0.22	0.18	0.09	0.07	0.10	0.08	
50%	0.29	0.27	0.26	0.09	0.11	0.18	
75%	0.41	0.36	0.42	0.12	0.13	0.28	
max	1.00	1.00	1.00	1.00	1.00	1.00	

Table 1: *Descriptive stats for the red wine data set*

	Fixed Acidity	Volatile Acidity	Citric Acid	Residual Sugar	Chlorides	Free sulfur dioxide	Total Sulfur D
mean	0.29	0.19	0.20	0.09	0.11	0.12	
std	0.08	0.10	0.07	0.08	0.06	0.06	
min	0.00	0.00	0.00	0.00	0.00	0.00	
25%	0.24	0.13	0.16	0.02	0.08	0.07	
50%	0.29	0.18	0.19	0.07	0.10	0.11	
75%	0.34	0.24	0.23	0.14	0.12	0.15	
max	1.00	1.00	1.00	1.00	1.00	1.00	

Table 2: *Descriptive stats for the white wine data set*

the independent features were then normalized. Normalization of data is usually a necessary precursor for analysis via neural networks. Normalization is a broad method of reducing feature values to be within a specified range (usually 0 to 1 or -1 to 1). Normalization and standardization methods include batch normalization, weighted-batch normalization, Min-Max scaling, among many others. In this, we will use Min-Max scaling, which scales the values down to be within $[0, 1]$. This step is important to make the features comparable to one another. We also then performed our split with 70% training and 30% testing data.

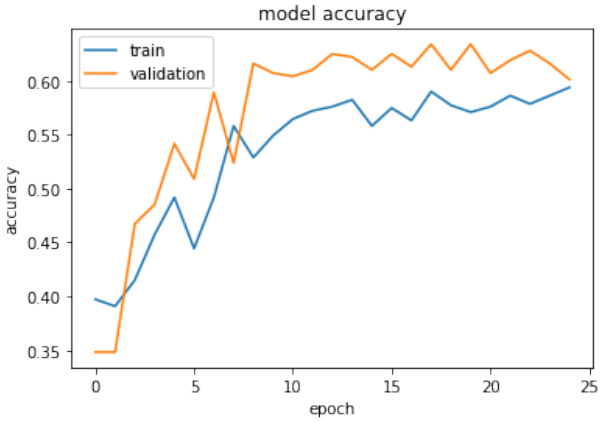

Figure 5: *Red wine accuracy v. 25 epochs*

Results

Two sequential neural network models were then constructed using Keras framework and Tensorflow. Each training set was fitted to each model, and trained for $25, 50, 100$, and 200 epochs. The batch size used for all was 20. Resulting metrics of accuracy and loss vs. epochs were calculated, and are listed in 5,7,6,8.

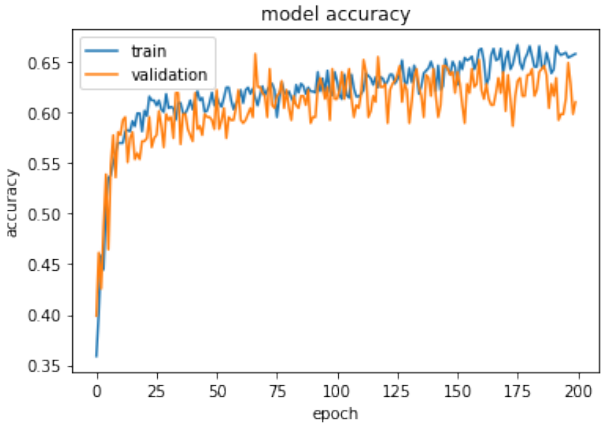

Figure 6: *Red wine accuracy v. 200 epochs*

Red wine accuracy and loss

As apparent, increasing the epochs from 25 up to 200 results in increased accuracy, but a divergence in the validation loss with respect to the training loss, indicating that 200 may be too many epochs to train on in this instance. Better results were seen with 50 and 100 epochs, indicating that the "sweet spot" for training epochs lies somewhere in this range. The results were identical for the white wine set, and will be omitted to reduce redundancy. These results are from a model using six dense hidden layers with $60, 50, 40, 30$, and 10 layers, respectively. A classifica-

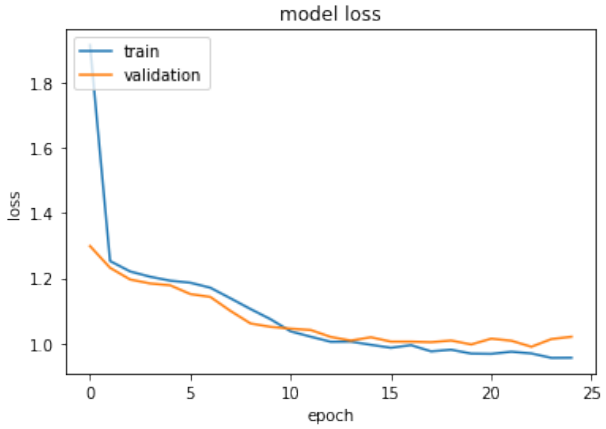

Figure 7: *Red wine loss v. 25 epochs*

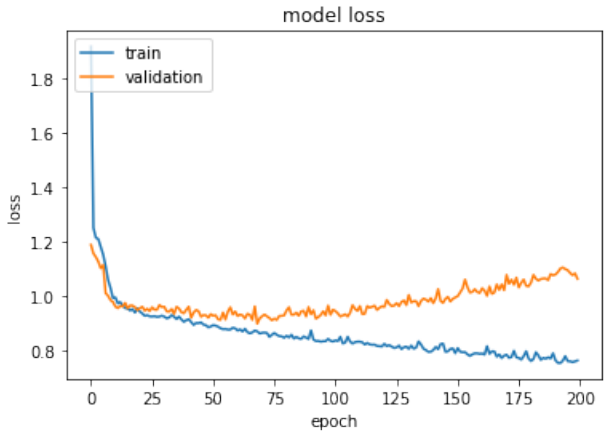

Figure 8: *Red wine loss v. 200 epochs*

tion table was then constructed to analyze the precision, recall, f1, and support metrics for 50 epochs with this model. The results are listed in 3 below. The confusion matrix, from which the precision and recall are calculated from, was

$$\begin{bmatrix} 0 & 0 & 2 & 2 & 0 & 0 \\ 1 & 2 & 11 & 5 & 1 & 0 \\ 0 & 3 & 147 & 52 & 6 & 0 \\ 2 & 7 & 34 & 99 & 28 & 0 \\ 0 & 0 & 9 & 17 & 46 & 1 \\ 0 & 0 & 0 & 1 & 4 & 0 \end{bmatrix}$$

Results of classification was very similar for the white wine data set with 50 epochs was as follows,

The white wine confusion matrix was:

$$\begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 2 & 9 & 20 & 6 & 1 & 1 & 0 \\ 2 & 18 & 276 & 98 & 23 & 3 & 2 \\ 2 & 13 & 145 & 423 & 95 & 12 & 1 \\ 3 & 9 & 14 & 85 & 138 & 11 & 0 \\ 0 & 2 & 2 & 13 & 20 & 20 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Next, a modified version of the model used previously was constructed. In this construction, the dropout technique was employed, dropping 20% of weights between layers. Again, results between data sets were very similar, and so only the red wine data set will be display for brevity. Both sets were again trained for 25, 50, 100, and 200 epochs, and the resulting accuracy, loss versus epochs are as follows in

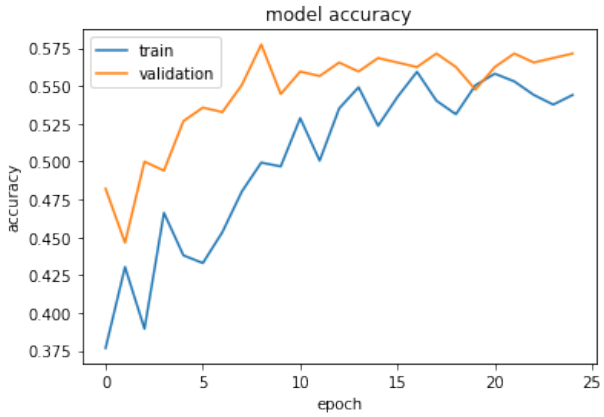

Figure 9: *Red wine dropout model accuracy v. 25 epochs*

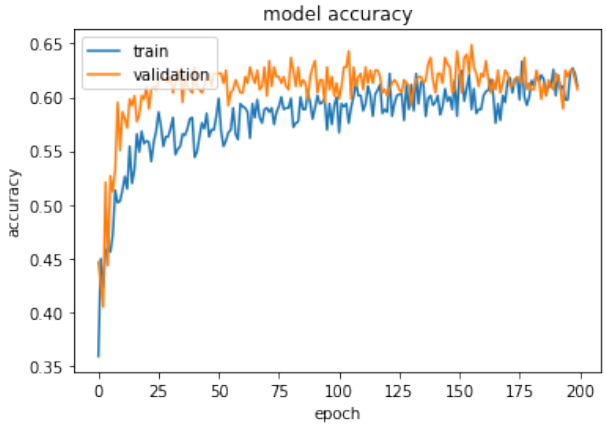

Figure 10: *Red wine dropout model accuracy v. 200 epochs*

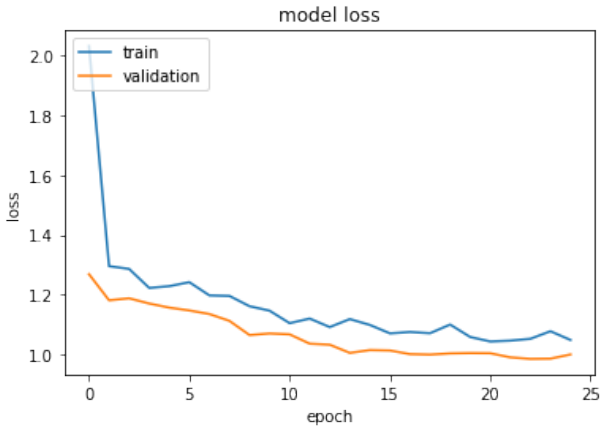

Figure 11: *Red wine dropout model loss v. 25 epochs*

	Precision	Recall	f1-score	Support
3	0.000000	0.000000	0.000000	3.000000
4	0.066667	0.055556	0.060606	18.000000
5	0.681818	0.627907	0.653753	215.000000
6	0.562189	0.638418	0.597884	177.000000
7	0.649123	0.616667	0.632479	60.000000
8	0.500000	0.428571	0.461538	7.000000
Accuracy	0.602083	0.602083	0.602083	0.602083
Macro avg	0.409966	0.394520	0.401043	480.000000
Weighted avg	0.603637	0.602083	0.601360	480.000000

Table 3: *Classification table for red data set using* 50 *epochs*

	Precision	Recall	f1-score	Support
3	0.000000	0.000000	0.000000	8.000000
4	0.288889	0.245283	0.265306	53.000000
5	0.617117	0.612975	0.615039	447.000000
6	0.631988	0.620427	0.626154	656.000000
7	0.577778	0.595420	0.586466	262.000000
8	0.338462	0.500000	0.403670	44.000000
accuracy	0.593197	0.593197	0.593197	0.593197
macro avg	0.409039	0.429018	0.416106	1470.000000
weighted avg	0.593208	0.593197	0.592623	1470.000000

Table 4: *Classification table for white data set using* 50 *epochs*

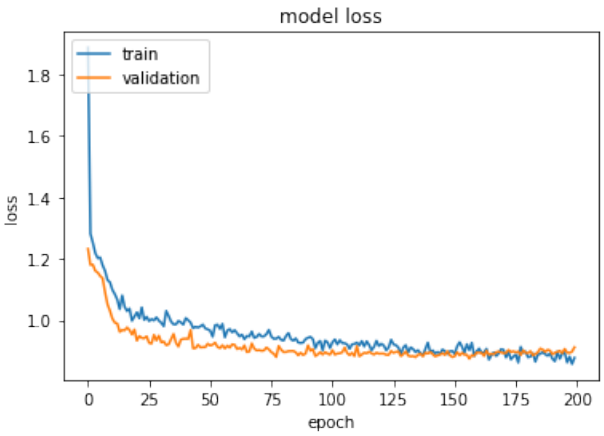

Figure 12: *Red wine dropout model loss v. 200 epochs*

From these, it is evident that the dropout technique was effective in reducing overfitting of the model, as there was less divergence from the training and validation sets.

Conclusion

In this, we used sequential neural networks to analyze data sets of 11 features of white and red wine to predict quality. We used preprocessing techniques of descriptive statistics, normalization/standardization, and built a model to be trained. Upon reflection of training vs. validation metrics upon the model results, overfitting was detected. To combat this, we used the weak dilution method of dropout to remove some of the weights used in the original model. Overfitting was also seen as when a large amount of epochs were used, and so the ideal number of epochs in this scenario was somewhere in the 50 to 100 range, as that was when we saw the least overfitting. Results from the dropout model improved, as testing data closely matched training data in both accuracy and loss metrics. The precision and recall had also improved in this new model. Overall, we can see that the dropout method can be a useful tool in improving a sequential neural network model's prediction capabilities.