HW2 - Wine Quality Data

Ryan King

Introduction

For this assignment, I chose to use the Wine Quality Data from the UC Irvine Machine Learning Repository which containted 11 different features/variables. Each wine in the data set was given a quality score ranging from 1 to 10. I decided to make a neural network and logistic regression model to use the other variables in the data set as predictors to predict the quality of a wine.

Analysis

During the exploratoty data analysis, I dug deep to understand the characteristics of the variables. Pictured below are tables displaying the summary statistics of each variable in the data set. I used the code provided by the UCI Machine Learning Repository to load in and fetch the data set. I stored all the predictor variables in one variable called predictors and stored quality in its own variable called predict. I then had to z score all my variables to put all my variables on a similar scale for measuring against each other since their values were all over. I also split the data into training and testing sets so that I could build my neural network model and logistic regression model using them.

	fixed_acidity	volatile_acidity	citric_acid	residual_sugar
count	4898.000000	4898.000000	4898.000000	4898.000000
mean	6.854788	0.278241	0.334192	6.391415
std	0.843868	0.100795	0.121020	5.072058
min	3.800000	0.080000	0.000000	0.600000
25%	6.300000	0.210000	0.270000	1.700000
50%	6.800000	0.260000	0.320000	5.200000
75%	7.300000	0.320000	0.390000	9.900000
max	14.200000	1.100000	1.660000	65.800000

	chlorides	free_sulfur_dioxide	total_sulfur_dioxide	density
count	4898.000000	4898.000000	4898.000000	4898.000000
mean	0.045772	35.308085	138.360657	0.994027
std	0.021848	17.007137	42.498065	0.002991
min	0.009000	2.000000	9.000000	0.987110
25%	0.036000	23.000000	108.000000	0.991723
50%	0.043000	34.000000	134.000000	0.993740
75%	0.050000	46.000000	167.000000	0.996100
max	0.346000	289.000000	440.000000	1.038980

	pH	sulphates	alcohol	quality
count	4898.000000	4898.000000	4898.000000	4898.000000
mean	3.188267	0.489847	10.514267	5.877909
std	0.151001	0.114126	1.230621	0.885639
min	2.720000	0.220000	8.000000	3.000000
25%	3.090000	0.410000	9.500000	5.000000
50%	3.180000	0.470000	10.400000	6.000000
75%	3.280000	0.550000	11.400000	6.000000
max	3.820000	1.080000	14.200000	9.000000

Methods

The primary model was a neural network, built by using TensorFlow and Keras. A neural network is like a complex connection of computations that learn and develop from the data it is given to make predictions. The neural network I created had multiple layers that contained many nodes that take in input data, complete calculations and move the results along through the network. I experimented with different layer amounts, changing input layer node amounts, and trying different activation methods but landed on the ones listed below. I started off my model with an input layer of 256 nodes, followed by two hidden layers that contained 128 and 64 nodes. All layers had a ReLu activation function to help give our model better results. My model's output layer had 10 nodes and used a softmax activation, which is useful for this case because the quality of wine was a classification problem. Throughout the model, I also implemented dropout regularization to help prevent overfitting. This process randomly ignores certain nodes in the network throughout the training which makes it so the model cannot over rely on certain connections between nodes or patterns in the network. Finally, I printed metrics like train/test loss, train/test accuracy, and train/test mean absolute error to see how my neural network performed.

I built a Logistic Regression model to compare my neural network to. I used the same predictors and outcome as I did in the neural network. The logistic regression model uses the

relationship of the predictors to predict the value of the outcome which in this case was wine quality. I also printed out the train/test accuracy and train/test mean absolute error of my logistic regression model so that I could compare it's performance to the neural networks.

Results

Neural Network Metrics: - Train Accuracy: 56.13% - Train Mean Absolute Error (MAE): 0.115 - Test Accuracy: 59.80% - Test MAE: 0.114

Logistic Regression Metrics - Train Accuracy: 53.37% - Train MAE: 0.522 - Test Accuracy: 55.51% - Test MAE: 0.511

In evaluating the performance between the neural network and logistic regression model, the metrics displayed some notable differences. The neural network displayed a much better accuracy on the training and testing datasets which shows the networks ability to correctly classify wines. When evaluating model precision, the neural network produced drastically lower MAE values of 0.115 for training and 0.114 for the testing set, which means it has a much higher predictive precision compared to the logistic regression models MAE values of 0.522 and 0.511. When the logistic regression model misclassifies instances, the predicted values usually are further from the actual values. This happens because of its higher MAE score.

The neural network did do better than the logistic regression when it comes to this wine quality data. It produced better performance metrics, having a better accuracy and much better mean absolute error value. The logistic regression model is much more simple, more interpretability, and less computationally expensive than the neural network. These reasons plus the fact that the neural network's accuracy was not much higher lead me to believe that the neural network is not needed for this case involving wine. The wine industry may not have the technical knowledge like other industries so using a simpler model like the Logistic Regression would be beneficial due to its easy interpretability. It is a model like requires less computational resources and is much quicker to train and test than the neural network.

Reflection

During this assignment, I had some difficulties testing out the different values while creating my neural network. The real-world data presented certain challenges like choosing the right architecture for my model, and making sure that the data was properly z scored for training. One of the biggest hurdles to overcome was exploring the different number of layers and nodes within each layer to see how they would impact my models performance. A challenge towards the end of the assignment was deciding between a neural network that had slightly better performance metrics and a logistic regression model that was cost less computationally and had less complexity. Weighing the models against each other gave me an opportunity to work

on a real world scenario and make a decision based on my findings. After completing this assignment, I feel more comfortable with neural networks and have a better understanding of how they work. In the future, I would ask more questions about the assignment so that I understand completely what is expected of my work.