Compare classification performance of different machine-learning algorithms

Glenn Guan

Introduction:

Machine-Learning as a growing field of study has proven its ability to perform a variety of different tasks. From identifying pictures to understanding human languages. Entrepreneurs from different markets are focused on the development and use of machine-learning to fuel their growth and increase their market shares. One common use of machine learning is to predict results based on inputs. This is also the topic this research focuses on.

Description of the dataset:

The dataset chosen for this research is publicly accessible provided by UCI Machine Learning Repository. Dataset is called “winequality-white” which is related to white variants of the Portuguese “Vinho Verde” wine (Paulo Cortez, 2009).

The dataset contains 4898 data entry. Each data entry contains 12 features. the Total number of data points is 58776 and the dimension of this dataset is 4898 by 12.

Below are the labels and explanation of each feature :

1. Fixed acidity: Amount of tartaric acid
2. Volatile acidity: Amount of acetic acid
3. Citric acid: Small quantity, add “freshness “ and flavor
4. Residual sugar: Amount of sugar remaining after fermentation stops.
5. Chlorides: Amount of salt
6. Free sulfur dioxide: The amount of free form of SO2 exists.
7. Total sulfur dioxide: Amount of free and bound forms of SO2
8. Density: Density of the wine
9. pH: How acidic and basic of the wine
10. Sulphates: Wine additive acts as antimicrobial and antioxidant
11. Alcohol: Alcohol content of the wine
12. Quality: Score between 0 and 10

(by Daria Alekseeva)

In this research, the first 11 features will be used to predict the last feature with is the quality.

Pre-processing of data:

A dataset with this size requires a carefully exploratory analysis. This will help identify outlier and provide a basic understanding of the data. Both of these things are important for algorithm selection and parameter choice.

In this research, I am using the python library pandas to import the data file. The first step is to identify any missing values in the entire dataset. After running some python script, the result is all false, which indicates no null values in the data.

The next step is to identify outliers, python 'pandas.dataframe' module provides an easy way to check the data. Based on the result, residual sugar, total sulfur dioxide, and free sulfur dioxide all show potential for containing outlier data. The mean value for residual sugar is 6.39, the minimum value is 0.6 and the maximum value is 65.8. The mean value for total sulfur dioxide is 138.36, the minimum value is 9 and the maximum value is 440. The mean value for free sulfur dioxide is 35.31, the minimum value is 2 and the maximum value is 289.

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcohol	quality
count	4898.000000	4898.000000	4898.000000	4898.000000	4898.000000	4898.000000	4898.000000	4898.000000	4898.000000	4898.000000	4898.000000	4898.000000
mean	6.854788	0.278241	0.334192	6.391415	0.045772	35.308085	138.360657	0.994027	3.188267	0.489847	10.514267	5.877909
std	0.843868	0.100795	0.121020	5.072058	0.021848	17.007137	42.498065	0.002991	0.151001	0.114126	1.230621	0.885639
min	3.800000	0.080000	0.000000	0.600000	0.009000	2.000000	9.000000	0.987110	2.720000	0.220000	8.000000	3.000000
25%	6.300000	0.210000	0.270000	1.700000	0.036000	23.000000	108.000000	0.991723	3.090000	0.410000	9.500000	5.000000
50%	6.800000	0.260000	0.320000	5.200000	0.043000	34.000000	134.000000	0.993740	3.180000	0.470000	10.400000	6.000000
75%	7.300000	0.320000	0.390000	9.900000	0.050000	46.000000	167.000000	0.996100	3.280000	0.550000	11.400000	6.000000
max	14.200000	1.100000	1.660000	65.800000	0.346000	289.000000	440.000000	1.038980	3.820000	1.080000	14.200000	9.000000

Multiple ways can be used to identify and eliminate outliers from the dataset. Z-score is the method that I am going to use in this research. The mathematical interpretation of the Z-score is to subtract the mean of the population from the original data points and divide it by the standard deviation. This measures how far away (how many standard deviations away) from the data point to the mean value. Any points that are too far away will be considered an outlier and eliminated. Based on the most common analysis, 99% of values have a Z-score between -3 and 3 are considered as 'good' dataset(Adam Hayes, 2020). In this research, the parameter for Z-score will be set to 3.

After applying the Z-score function to remove outliers, the new dataset now contains 4487 entry which is approximately 91.6% of the original dataset.

After eliminating the dataset, the next step is to divide the total data into a training set and a validation set. In this research, 90% of the data are randomly selected to be training data, and the remaining 10% serve as validation sets.

Algorithms used:

Algorithms selected for this research is based on both requirement and previous experience with machine-learning algorithms. The first algorithm is the Linear regression. Linear regression is the most commonly used classification algorithm. The logic behind this algorithm is to mathematically find a line that represents the boundary between output values. The second algorithm selected for this research is K-nearest neighbors. The logic behind this algorithm is to find a K number of data points in the training sample and calculate the distances between the data point. The last algorithm is Neural Network. The idea behind the neural network is to find the correct hidden weights between each node, also known as neurons. The process of finding hidden weights by knowing the output value and the input value is called backpropagation.

How to compare different algorithms:

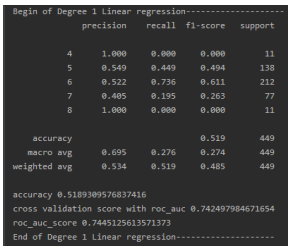
The way to compare the performance of each algorithm depends on the average accuracy of 10 folds cross-validation results.

How to choose parameters for linear regression :

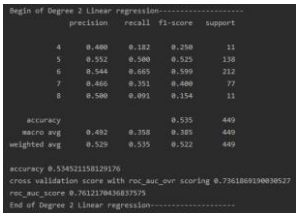
The parameter chosen to be modified with linear regression is the degree of polynomial functions. The boundary (which is the line) that differentiate the result will change based on the degree. The degree can not go without a limit, higher degree will likely result in overfitting of the data.

Results of each test in linear regression:

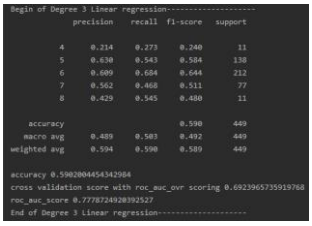
Degree 1 polynomial: Average accuracy is 74.45%



Degree 2 polynomial: Average accuracy is 76.12%



Degree 3 polynomial: Average accuracy is 77.79%



With the degree of polynomial function change from 1 to 3, the overall accuracy after averaging the result of 10 folds cross-validation changed from 74% to 77%.

How to choose parameters for K-nearest neighbor:

K-nearest neighbor(KNN) algorithm's performance depends on the choice of the value of K. In this research, I calculated the accuracy for K values from 1 to 20.

The result indicates that the K value should be 19 with an accuracy score of 0.52085.

Results of KNN:

K = 19 : Average accuracy is 80.04%

```
------------------- begin k = 19
              precision    recall  f1-score   support

           4      1.000     0.000     0.000        11
           5      0.559     0.551     0.555       138
           6      0.580     0.703     0.635       212
           7      0.536     0.390     0.451        77
           8      1.000     0.000     0.000        11

    accuracy                          0.568       449
   macro avg      0.735     0.329     0.328       449
weighted avg      0.586     0.568     0.548       449

cross validation score 0.5208508988227808
cross validation score with roc_auc 0.7089859400420266
roc_auc_score 0.8003801500338842
------------------- end k = 19
```

The K value selected for this research is the best result-producing K values from the initial process. The overall performance is better than Linear Regression.

How to choose parameters for the neural network:

Parameters that cause dramatic differences between different neural network models include the number of hidden layers and the number of hidden nodes. An increasing number of hidden layers and hidden nodes will dramatically impact the calculation time.

In this research, I also checked one additional factor which is the activation function.

```
1 layer, 100 hidden nodes, logistic activation function
              precision    recall  f1-score   support

           4      0.143     0.091     0.111        11
           5      0.631     0.558     0.592       138
           6      0.607     0.670     0.637       212
           7      0.560     0.545     0.553        77
           8      0.636     0.636     0.636        11

    accuracy                          0.599       449
   macro avg      0.515     0.500     0.506       449
weighted avg      0.596     0.599     0.596       449

accuracy 0.5991091314031181
cross validation score with roc_auc_ovr scoring 0.7162488385229351
roc_auc_score 0.8072161882051644
End Scenario 1
```

```
1 layer, 100 hidden nodes, RELU activation function
              precision    recall  f1-score   support

           4      0.250     0.091     0.133        11
           5      0.627     0.536     0.578       138
           6      0.567     0.698     0.626       212
           7      0.548     0.442     0.489        77
           8      0.250     0.091     0.133        11

    accuracy                          0.575       449
   macro avg      0.449     0.372     0.392       449
weighted avg      0.567     0.575     0.564       449

accuracy 0.5746102449888641
cross validation score with roc_auc_ovr scoring 0.7260757883908457
roc_auc_score 0.8019246662829753
End Scenario 2
```

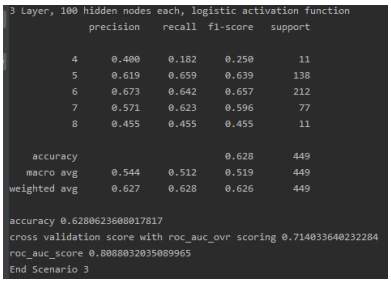
The result shows a slightly better performance with the logistic activation function than the ReLU activation function.

Results of each test in the neural network

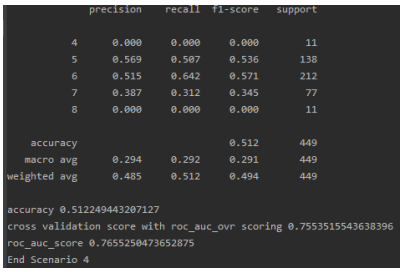
1 layer, 100 hidden nodes: Average accuracy is 80.72%

```
1 layer, 100 hidden nodes, logistic activation function
              precision    recall  f1-score   support

           4      0.143     0.091     0.111        11
           5      0.631     0.558     0.592       138
           6      0.607     0.670     0.637       212
           7      0.560     0.545     0.553        77
           8      0.636     0.636     0.636        11

    accuracy                          0.599       449
   macro avg      0.515     0.500     0.506       449
weighted avg      0.596     0.599     0.596       449

accuracy 0.5991091314031181
cross validation score with roc_auc_ovr scoring 0.7162488385229351
roc_auc_score 0.8072161882051644
End Scenario 1
```

3 layers, 100 hidden nodes each: Average accuracy is 80.88%



3 layers, 10 hidden nodes each: Average accuracy is 76.55%



The result shows 3 layers with 100 hidden nodes each have a better performance.

Strength and limitations of each method used:

Linear regression classifier, compared to other algorithms evaluated in this research, uses the least amount of time and resources. The downside is it is mostly used to classify data into a binary condition such as True or False, Yes or No, etc. When the data has multiple classes, it tends to produce an inaccurate result.

K-Nearest Neighbor classifier also has the advantage of less time and resources consuming. The downside is that process time significantly increases when datasets increase.

Neural Network classifier: It has the advantage of process non-linear data with a large number of inputs, and the prediction can be fast once trained. The downside is unclear how much each feature affects the outcome, time, and resource-consuming, and depends a lot on training data.

Conclusion:

The neural network algorithm has the best result in predicting the result with the dataset selected for this research. On the other hand, if we take computation time and resources used during the evaluation of each algorithm, the neural network may not be the go-to solution for every dataset. In this research, the neural network takes 3-4 times longer time and memory to do the classification, but the result does not show a dramatic difference. Factors such as time, resource availability, computation power, complexity should all play their role in the design and selection of machine learning algorithms.

References

1. P. Cortez, A. Cerdeira, F. Almeida, T. Matos and J. Reis.
   Modeling wine preferences by data mining from physicochemical properties. In Decision Support Systems, Elsevier, 47(4):547-553, 2009.
   http://archive.ics.uci.edu/ml/datasets/Wine+Quality
2. Cortez, P., Cerdeira, A., Almeida, F., Matos, T., & Reis, J. (2009). Modeling wine preferences by data mining from physicochemical properties. *Decision Support Systems, 47*(4), 547-553. doi:10.1016/j.dss.2009.05.016. http://www3.dsi.uminho.pt/pcortez/wine5.pdf
3. Hayes, A. (2020, September 23). What a Z-Score Tells Us. Retrieved December 13, 2020, from https://www.investopedia.com/terms/z/zscore.asp
4. Alekseeva, D. (n.d.). Red and White Wine Quality. Retrieved December 12, 2020, from https://rstudio-pubs-static.s3.amazonaws.com/57835_c4ace81da9dc45438ad0c286bcbb4224.html

Link to GitHub repository

https://github.com/guang16/Machine-Learning