Wine Quality Analysis – White wine - dataset

The following analysis refers to the Red Wine Quality dataset, a public dataset that contains physical-chemical data on red wines and their quality. Data were collected from northern Portugal and has just under 5000 observations. The dataset can be downloaded from https://archive.ics.uci.edu/ml/machine-learning-databases/wine-quality/ (winequality-white.csv). An artificial neural network and a multiple linear regression model done in python were implemented for the dataset analysis.

The goal of the study is to implement an artificial neural network and a multiple linear regression model and evaluate their differences, for example, the accuracy of the models.

Since a study has already been done on predicting the quality of red wine in my other post, I will not go into detail about each feature and its impact on the model. I'll leave the link of the post at the end of this text. for you to read if you want. However, I will make a brief introduction to the dataset and its possible problem.

Table 1 - Dataset

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcohol	quality
0	7.0	0.27	0.36	20.7	0.045	45.0	170.0	1.00100	3.00	0.45	8.8	6
1	6.3	0.30	0.34	1.6	0.049	14.0	132.0	0.99400	3.30	0.49	9.5	6
2	8.1	0.28	0.40	6.9	0.050	30.0	97.0	0.99510	3.26	0.44	10.1	6
3	7.2	0.23	0.32	8.5	0.058	47.0	186.0	0.99560	3.19	0.40	9.9	6
4	7.2	0.23	0.32	8.5	0.058	47.0	186.0	0.99560	3.19	0.40	9.9	6
...	...	...	...	...	...	...	...	...	...	...	...	...
4893	6.2	0.21	0.29	1.6	0.039	24.0	92.0	0.99114	3.27	0.50	11.2	6
4894	6.6	0.32	0.36	8.0	0.047	57.0	168.0	0.99490	3.15	0.46	9.6	5
4895	6.5	0.24	0.19	1.2	0.041	30.0	111.0	0.99254	2.99	0.46	9.4	6
4896	5.5	0.29	0.30	1.1	0.022	20.0	110.0	0.98869	3.34	0.38	12.8	7
4897	6.0	0.21	0.38	0.8	0.020	22.0	98.0	0.98941	3.26	0.32	11.8	6

Source: The Author

In table 1, we see that the features are the same as the ones seen in the post about predicting the quality of red wine, so we can expect a very similar behavior between the 2 datasets.

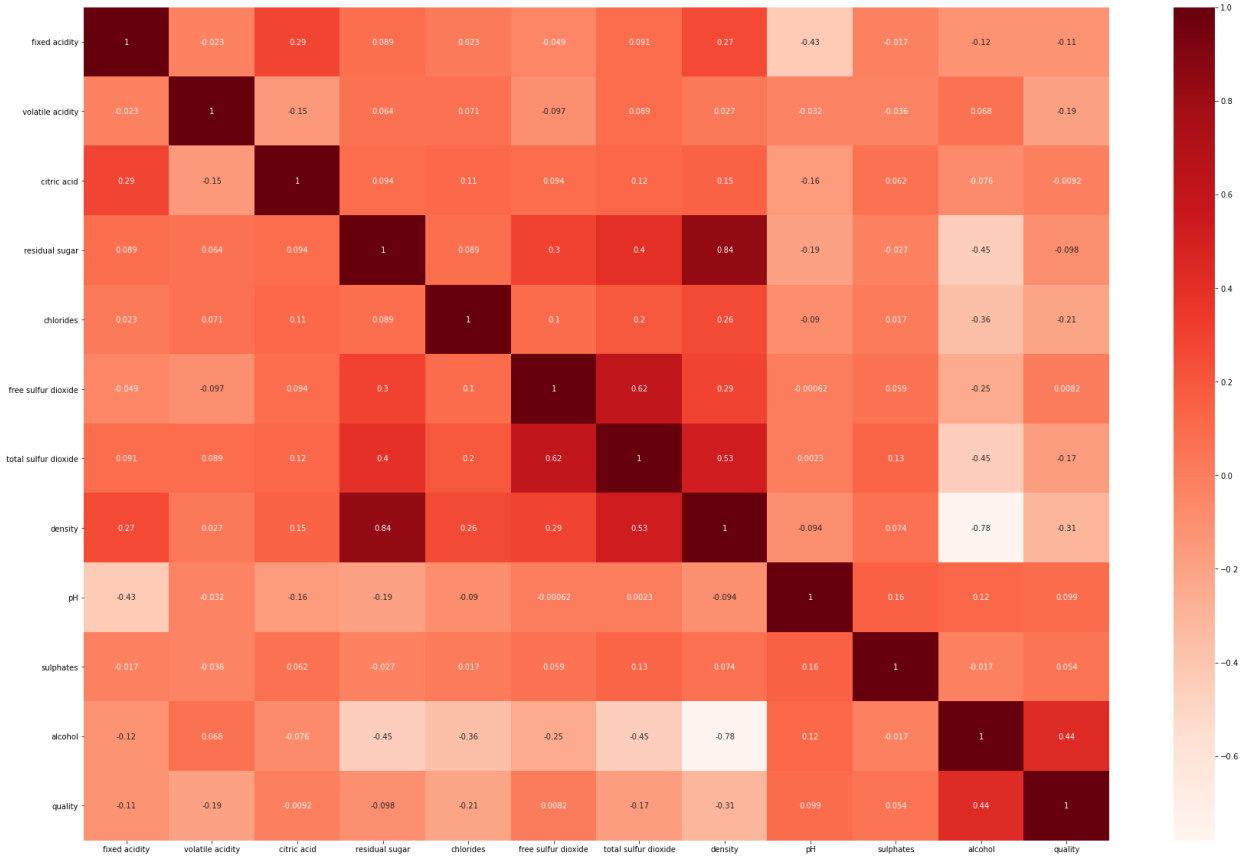

Figure 1 - Heatmap - white wine

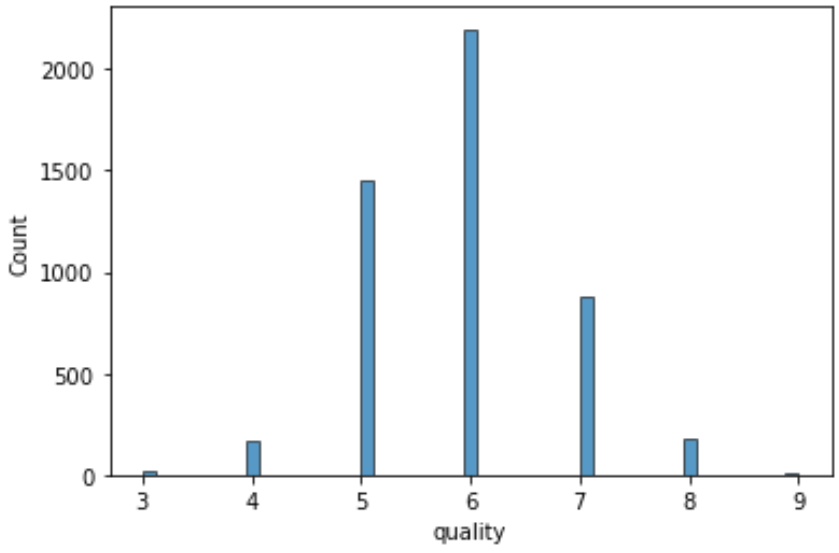

Figure 2 - Histogram of quality – white wine

We see that in Figure 1, the *same features* that present a high positive and negative correlation with the target*(quality)*are the same as in the red wine post. Just like the heatmap*,* we have the same problem as an unbalanced dataset, as seen in Figure 2, so we can expect a loss of accuracy *in* the result. More information in the post about red wine.

In this post, the study was done with artificial neural networks to predict the quality of white wine and compared it with the predicting power of a *multiple linear regression algorithm.*

The first step to building an artificial neural network is to pre-process the data. In it, the inputs were normalized for better performance of the algorithms and the outputdata was transformed to a *ndarray so* that we could use the data in tensorflow. Then the dataset *was* divided into 3 parts, one for training, one for validation, and the rest for the final accuracy *tests.* I chose to divide it into 80% for training, 10% for validation and the rest for testing. I choose this way to try to avoid overfitting the models. Finally, we saved these 3 datasets in .npz files to be used by tensorflow.

To create the artificial neural network, we will use the *tensorflow library,* the most used today. First, we load the data into the variables and then we define the number of inputs, in this case, there are 11, because there are 11 features and the number of outputs, in this case, are 10, because we have values of notes up to 9. For a first test, we make the number of hidden *layers* equal to 100. This first model (I) will have 2 layers reLU type activation function and the output layer of type *softmax,* because we have aclassification problem with more than 2 output types. To try to avoid *overfitting,* we have a *batchsize* equal to 100 and the number of *epochs* equal to 50. I also set a stop function if the algorithm detects that the model is losing accuracy *because* of overfitting. In addition to these settings, hyperparameters have *been* changed to get the best possible model. Another model (II) was made with 50 hidden *layers,* 3 layers with reLU activation functions and 3 dropout *layers.* The third model (III) was made with 50 hidden *layers,* 3 layers with reLU activation functions and 2 dropout layers. Finally, a multiple linear *regression* (IV) model was made for comparison with the other models.

Table 2 – Model results

I	II	III	IV
Test Accuracy (%)	*Test Accuracy* (%)	*Test Accuracy* (%)	*Test Accuracy* (%)
56,62	56,01	58,04	23,70

Source: The Author

In table 2, we see the results of similar models I, II, and III, showing that even adding and removing layers with activation and *dropout functions,* the neural network was not able to improve the final result. The result of model IV was not good when compared to the other models. More details on the reasons for this result of the model IV you can find in my post about red wine.

The artificial neural network is much more capable of achieving better results compared to the *machine learning model.* "Bad" results were expected for the models, because the *dataset is* extremely unbalanced, but, the most interesting thing, is to see how the complexity of artificial neural networks can extract a result far superior to the machine learning *model,* demonstrating its superiority in the aspect of *accuracy*.

All the code you find on my Github:
https://github.com/williamausenka/ML_estudos_de_caso/tree/main/Quality%20Red_Wine%20-%20MLR

I hope you enjoyed it.
Be sure to comment on what you thought of the analysis

Thanks, and until the next post!

https://www.boradedatascience.com/post/an%C3%A1lise-do-wine-quality-red-wine-dataset