Wine Quality Analysis

Hirley Dayan Lourenço da Silva *and* Marcia Maria Parmigiani Martins

Unicamp Data Science - Supervised Learning Module

Logistic regression is used to describe data and to explain the relationship between one dependent binary variable and one or more independent variables. This work explores Logistic Regression techniques for creating a model for classifying the quality of wines based on a few chemical features.

Logistic Regression | glm | glmnet

Introduction. This report contains an analysis of **WineQuality** datasets, using a machine learning pipeline for exploring the datasets and proposing a logistic classifier model for identifying good and bad wines based on a few features.

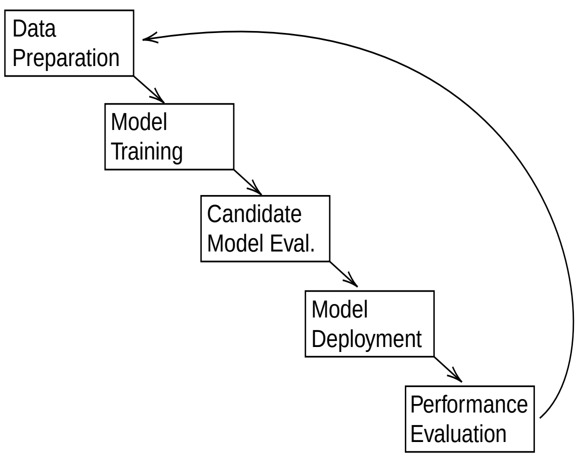

Fig. 1. Machine Learning Pipeline

Data Preparation. There are **3898** samples in the **training** dataset and **1299** samples in the **validation** dataset. The **validation** dataset represents **25%** of the total available data (both **training** and **validation** datasets). The **testing** dataset contains **1300** samples. A few samples of the **training** dataset is shown in the **Table 1**.

	1	2	3	4	5	6
fixed.acidity	7.10	6.00	7.90	6.20	7.00	7.00
volatile.acidity	0.33	0.39	0.18	0.28	0.50	0.31
citric.acid	0.30	0.17	0.49	0.51	0.25	0.31
residual.sugar	3.30	12.00	5.20	7.90	2.00	9.10
chlorides	0.03	0.05	0.05	0.06	0.07	0.04
free.sulfur.dioxide	30.00	65.00	36.00	49.00	3.00	45.00
total.sulfur.dioxide	102.00	246.00	157.00	206.00	22.00	140.00
density	0.99	1.00	1.00	1.00	1.00	0.99
pH	3.08	3.15	3.18	3.18	3.25	2.98
sulphates	0.31	0.38	0.48	0.52	0.63	0.31
alcohol	12.30	9.00	10.60	9.40	9.20	12.00
quality	1.00	0.00	0.00	0.00	0.00	1.00

Table 1. Training dataset

Training, **validation** and **test** datasets contains the following number of incomplete samples:

- **Training**: **0** incomplete samples
- **Validation**: **0** incomplete samples
- **Testing**: **0** incomplete samples

As shown above, there are **no** incomplete cases in the **training**, **validation** and **testing** datasets.

	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
fixed.acidity	3.80	6.40	6.90	7.19	7.60	15.90
volatile.acidity	0.08	0.23	0.29	0.34	0.40	1.58
citric.acid	0.00	0.24	0.31	0.32	0.39	1.66
residual.sugar	0.60	1.80	3.00	5.42	8.00	65.80
chlorides	0.01	0.04	0.05	0.06	0.06	0.61
free.sulfur.dioxide	1.00	17.00	29.00	30.64	41.00	146.50
total.sulfur.dioxide	6.00	76.25	118.00	115.33	155.00	366.50
density	0.99	0.99	0.99	0.99	1.00	1.04
pH	2.74	3.11	3.21	3.22	3.32	4.01
sulphates	0.23	0.43	0.51	0.53	0.60	2.00
alcohol	8.00	9.50	10.30	10.47	11.30	14.90
quality	0.00	0.00	0.00	0.20	0.00	1.00

Table 2. Training dataset overview (without normalization)

Table 2 and **Table 3** present an overview of the **training** dataset before and after normalization. For normalizing the datasets it was used the **min-max** normalization, as follows:

$$x' = \frac{x - min(x)}{max(x) - min(x)} \tag{1}$$

	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
fixed.acidity	0.00	0.21	0.26	0.28	0.31	1.00
volatile.acidity	0.00	0.10	0.14	0.17	0.21	1.00
citric.acid	0.00	0.14	0.19	0.19	0.23	1.00
residual.sugar	0.00	0.02	0.04	0.07	0.11	1.00
chlorides	0.00	0.04	0.06	0.07	0.09	1.00
free.sulfur.dioxide	0.00	0.11	0.19	0.20	0.27	1.00
total.sulfur.dioxide	0.00	0.19	0.31	0.30	0.41	1.00
density	0.00	0.10	0.15	0.15	0.19	1.00
pH	0.00	0.29	0.37	0.38	0.46	1.00
sulphates	0.00	0.11	0.16	0.17	0.21	1.00
alcohol	0.00	0.22	0.33	0.36	0.48	1.00
quality	0.00	0.00	0.00	0.20	0.00	1.00

Table 3. Training dataset overview (normalized)

The box plot in the **Figure 2** gives a good overview of the data distribution of the **training** dataset before the removal of the outliers. For the removal of the outliers, which are values that differ considerably from the majority of a set of data, different techniques are available. In this study, outlier removal was performed using the *capping* technique,by replacing values outside the $1.5 * IQR$

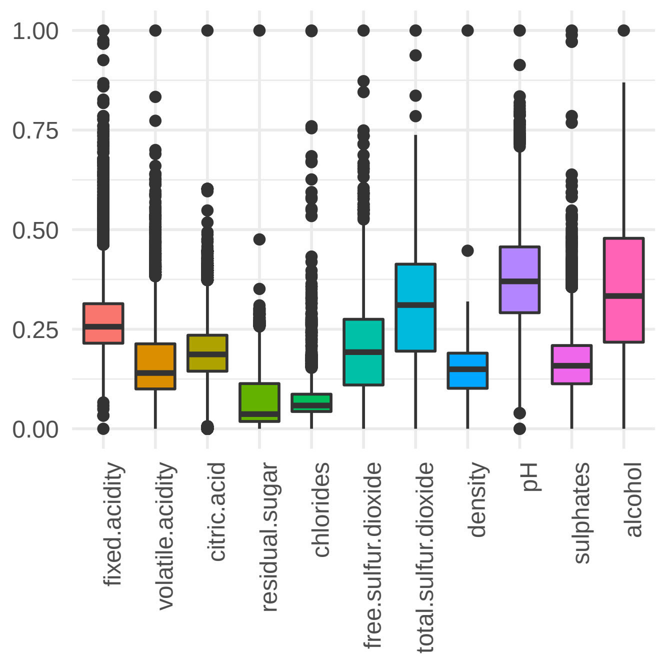

Fig. 2. Data distribuition analysis

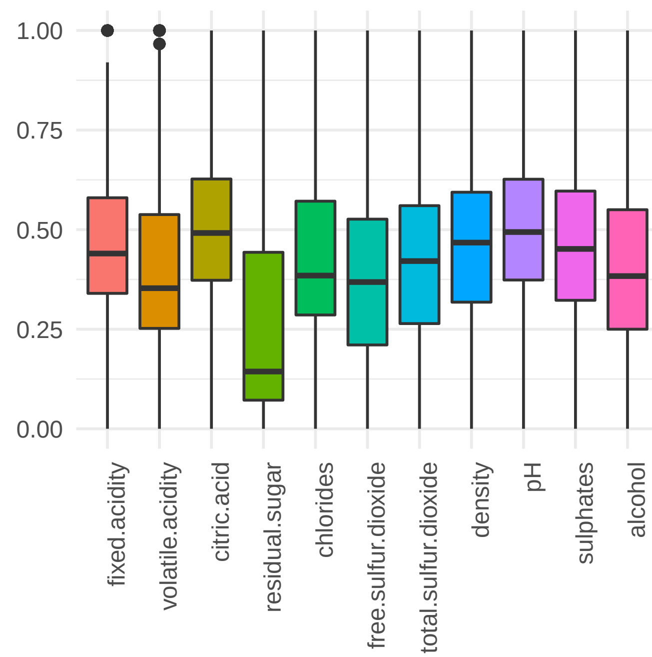

Fig. 3. Data distribuition analysis after capping

limits, with the lower limit replaced by the **5th** percentile and the bigger limit replaced by the **95th** percentile.
The lower and upper limits are calculated by the equations:

$$IQR = Q3 - Q1 \tag{2}$$
$$Lower = Q1 - 1.5 * IQR \tag{3}$$
$$Upper = Q3 + 1.5 * IQR \tag{4}$$

Once the limits are calculated, outliers below the **5th** percentile and above the **95th** percentile are replaced by both ***Lower*** and ***Upper*** limits, respectively.
After the **capping** of the outliers, the new data distribution is shown in **Figure 3**.
According to the quality, the datasets are balanced as follows:

- **Tranining**: **19.78%** of good wine samples
- **Validation**: **64.36%** of good wine samples
- **Test**: **61.38%** of good wine samples

Model Training, Deployment and Evaluation. For training the model, quadratic and cubic functions were created with the dataset features. Additionally, the ***SMOTE*** technique was used for dealing with the imbalanced datasets.

Formula	BACC	F1	Good Wine Perc
1	0.7041	0.6721	66.67
1	0.6836	0.5902	57.15
2	0.7115	0.6804	66.67
2	0.6894	0.5981	57.15
2	0.6573	0.5199	50.00
3	0.7246	0.6941	66.67
3	0.6867	0.6054	57.15
3	0.6692	0.5517	50.00

Table 4. Cross analysis matrix (without penalty terms)

For fitting the model, the `glm` function was used for tuning the predefined functions, with the **training** dataset re-balancing during validation. The result of the analysis can be found in **Table 4**. The **confusion matrix** of the predicted wine quality against the actual classification, when using the **testing** dataset, can be found in the **Table 5**. The performance during the tests is also shown by the **ROC** curve in **Figure 4**.

	0	1
0	464	450
1	38	348

Table 5. Confusion matrix (without penalty terms)

The function `glmnet` was also used for fitting the model, and during the evaluation of the quadratic and cubic functions, penalties were applied by changing the value of `lambda` parameter. Similarly to the previous model fitting by `glm`, the **training** dataset was also re-balanced during the training. The result of that analysis can be found in **Table 6**. For avoiding spending space in this report, the **Table 6** does not bring all collected values in the training. Only the values with **Balanced Accuracy (BACC)** bigger than **70%** are shown. The **confusion matrix** for the predicted wine quality against the actual quality, in the the **testing** dataset, can be found in **Table 7**. The performance during the tests of the classifier is also shown by the **ROC** curve in **Figure 5**.

Final Conclusions. The logistic regression classifier with a generalized linear model with penalization, by means of the function `glmnet`, has big potential for bringing good results due to its fine-tuning parameters. Despite all the potentials of the `glmnet`, this study showed that the `glm` classifier performs well, giving almost similar results as the `glmnet` classifier. The dataset features were not enough for creating a good classifier, in both cases evaluated, and the addition of 3rd-degree components was necessary to increase the performance during training and validation. Addition-

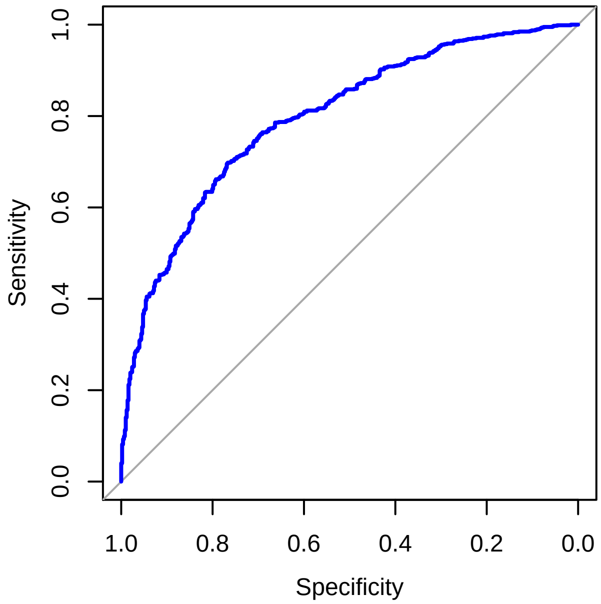

Fig. 4. ROC (without penalty terms)

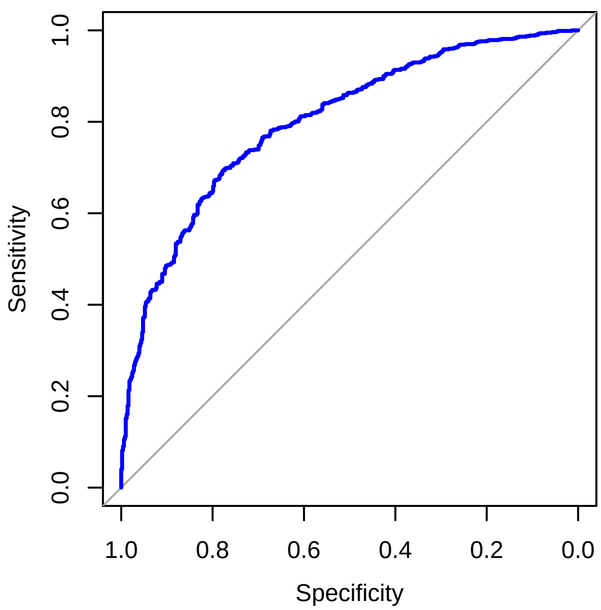

Fig. 5. ROC (with penalty terms)

Formula	Alpha	Lambda	BACC	F1	Good Wine Perc
1	0	0.0	0.7041	0.6721	66.67
1	0	0.0	0.7041	0.6721	66.67
1	0	0.0	0.7041	0.6721	66.67
1	0	0.0	0.7041	0.6721	66.67
1	0	0.0	0.7041	0.6721	66.67
1	0	0.0	0.7063	0.6731	66.67
1	0	0.0	0.7100	0.6804	66.67
1	0	0.1	0.7160	0.7033	66.67
2	0	0.0	0.7115	0.6804	66.67
2	0	0.0	0.7115	0.6804	66.67
2	0	0.0	0.7115	0.6804	66.67
2	0	0.0	0.7115	0.6804	66.67
2	0	0.0	0.7126	0.6809	66.67
2	0	0.0	0.7127	0.6761	66.67
2	0	0.0	0.7059	0.6687	66.67
2	0	0.1	0.7131	0.6866	66.67
3	0	0.0	0.7212	0.6911	66.67
3	0	0.0	0.7212	0.6911	66.67
3	0	0.0	0.7224	0.6930	66.67
3	0	0.0	0.7192	0.6915	66.67
3	0	0.0	0.7215	0.6878	66.67
3	0	0.0	0.7160	0.6776	66.67
3	0	0.0	0.7096	0.6697	66.67
3	0	0.1	0.7121	0.6814	66.67
3	0	1.0	0.7022	0.7282	66.67

Table 6. Cross analysis matrix (with penalty terms)

	0	1
0	478	527
1	24	271

Table 7. Confision matrix (with penalty terms)

ally, the tuning of the classifier with a different proportion of wine classes showed that a better performance can be achieved by having more good wine samples in the training dataset.