Wine prediction using Classic Machine Learning Algorithms

Mariana Ramrez Duque
Computer Engineering department
Universidad EAFIT

I. Introduction

Machine Learning can be define as a branch of Artificial Intelligence that uses algorithms that can learn from data without relying on rules-based programming[2]. In this paper we use some classic Machine Learning algorithms to predict the quality of wine.

II. Data Exploration

A. Data Quality

This data set contains 11 input features, also called independent variables, which are:

- fixed acidity
- volatile acidity
- citric acid
- residual sugar
- chlorides
- free sulfur dioxide
- total sulfur dioxide
- density
- pH
- sulphates
- alcohol

All of whom share the same float64 data type and whose combination define the output or dependent variable, quality, of type int64 that ranges from 1 to 10. 1 meaning that a wine has the worst quality and 10 meaning it has the best quality.

By looking at the shape of the data set we can conclude it is has a small number of examples, only 4200 entries. However, this examples don't contain any null or corrupt values making it a neat and well structured data set.

B. Class distribution

In order to do a binary classification we define a threshold of 5 so that if the quality variable is greater than 5 then the wine has a good quality. Doing this the output variable becomes 0 (bad) or 1(good).
By analyzing the distribution of this two output classes we get that 33% of the output variables in the data set are 0 and 67% are 1,showing that the classes are not balanced

III. Data partitioning

In our prediction of wine quality it is essential to include training and testing sets. The training set is used for building a model that can predict new data and the testing set aims at measuring the accuracy of the prediction made by the model [7,8].It is common to use part of the training set to test, tune and compare various models, such set is called the validation set.

We divide our data set into training set and testing set, or, training set, validation set and testing set based on the number of hyperparameters in each model.

A. Training set and testing set

We divide our data set only into training and testing in the Lineal Regression model due the small number of hyperparameters that we can tune. We partitioned the training set using 70% (2940) of the data and testing set using 30% (1260) of the data

B. Training set, validation set and testing set

For the other three machine learning algorithms we use training set, validation set and testing test since they all contain a high number of hyperparameters to tune, thus, we do not wish to overfit or bias the testing set, instead, it is better to have another another set to tune and test the hyperparameters

We partitioned the data set into 60% (2520) for the training set, 20% (840) for the validation set and also 20%(840) for the testing set

IV. Data analysis and feature selection

Understanding well the features is an essential part for getting a high accuracy in any prediction technique. In this section we will focus on understanding well the data and removing any unnecessary features that may be redundant and even damaging our performance

The importance of feature selection resides in the fact that there are several feature which are lineally dependent with other features, making them nothing an extension of the other essential features.Therefore there is a need to extract the most important and relevant feature to get the most effective predicting modelling[4]

Guyon at al. [5, p.1] stated that "The objective of variable selection is three-fold: improving the prediction performance of the predictors, providing faster and more cost-effective predictors, and providing a better understanding of the underlying process that generated the data."

A. Correlation matrix

A correlation analysis is a statistical method used to denote the strength of the relationships between two, numerical and continuous, variables.If there exist a correlation between two variables, it means that when there is a systematic, there

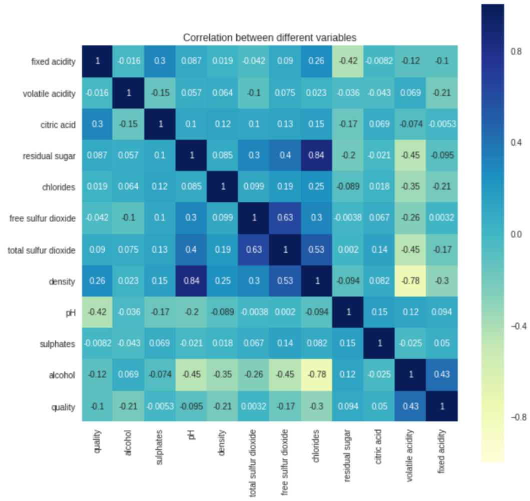

Fig. 1. Correlation matrix of features in the wine data set

is also a systematic change in the other variable.[3] The correlation coefficient can go from -1, inverse proportionality, to 1, positive or direct proportionality. A correlation coefficient of 0 means that there is no correlation between the two variables.

In Fig. 1. we see that the correlation matrix is symmetric, this happens because the correlation of variable a with variable b is the same as the correlation of variable b with variable a. In addition, the diagonal of the matrix is one since a correlation of a variable with itself is always one. We can see that the correlation between the pH and the chlorides is high (.84), as well as the correlation between the chlorides and the alcohol(.76) and the correlation between the pH and total sulfur dioxide(.63). One may think that it can be good to remove, for example, the alcohol variable since it has a lineal dependency with chlorides, free sulfur dioxide and pH, therefore it can be represented by them, we will see that later.

B. Filter method for Feature selection

Filter methods rely on the basics characteristics of the training data to select the best features with independence of any machine learning algorithm. This makes them faster than other methods and better at generalizing.[6]

The filter method that we are going to implement is a Chi-squared statistical test for non-negative features to see which features have a stronger relationship with the output variable. By doing this we get that the features with a higher relationship with quality are: total sulfur dioxide,residual sugar, alcohol, volatile acidity, fixed acidity, chlorides, sulphates, pH, free sulfur dioxide, citric acid, density in the respective order. We decided to remove the 6 features with a weaker relationship to the dependent variable, reducing the number of features from 11 to 5.

C. Wrapper method for feature selection

The wrapper method uses the prediction performance of a given learning algorithm to define the usefulness of a subset of features. In order to address the search of all possible variables subsets one can use any of the two greedy search strategies: forward selection and backward elimination. Forward elimination consists in progressively incorporating variables into larger and larger subsets while backward elimination is based in starting with a set of all the variables and progressively eliminating the least promising ones [5].

In this case we use backward elimination in each machine learning algorithm that we are using to get the importance of all the features and then proceed to evaluate the model accuracy for all binary classification models and the Root Mean Square Error for Linear Regression.

Machine Learning Algorithm	Feature importance from high to low using backwards elimination
Linear Regression	density, chlorides, volatile acidity, sulphates,alcohol, citric acid, pH, residual sugar, fixed acidity, free sulfur dioxide, total sulfur dioxide
Logistic Regression	volatile acidity, chlorides, sulphates, density, alcohol, pH, fixed acidity, residual sugar, citric acid, free sulfur dioxide, total sulfur dioxide
Decision Tree	free sulfur dioxide, alcohol, pH, volatile acidity, total sulfur dioxide, chlorides, citric acid, residual sugar, sulphates, density, fixed acidity
Random Forest	density, free sulfur dioxide, alcohol, volatile acidity, total sulfur dioxide, residual sugar, chlorides, citric acid, pH, fixed acidity and sulphates

TABLE I

FEATURE IMPORTANCE FROM HIGH TO LOW USING BACKWARDS ELIMINATION

As shown in I the most important features in the wrapper method change from one learning algorithm to another, however, volatile acidity and alcohol remain in the top five most relevant features in all the machine learning algorithms. This implies that these two features are useful to the output variable no matter the predictor used, the filter method confirms that, as it also includes this two variables in its top five most relevant features. It is important to highlight that the correlation between alcohol and volatile acidity is low (0.069) which mean that they both contribute to the output variable in a different way

Taking into account the previously discussed importance of the alcohol, one can also see that although alcohol has a correlation with chlorides, free sulfur and dioxide, as seen in the correlation matrix, it is not a good idea to remove it due its usefulness to the output variable

Fig. 2. shows the performance of the four machine learning algorithms based on the number of features, ranked from more important to least important, according to the wrapper method. At the beginning, the performance of all the algorithms tend to increase significantly as the number of features increases, this changes when the fifth feature is included and the performance start to raise gradually. One can then

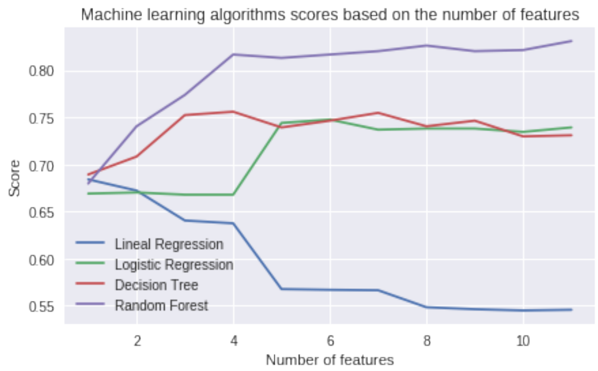

Fig. 2. Machine learning algorithms scores based on the number of features

conclude that, apart from the Decision Tree classifier, all algorithms tend to get better with a higher number of input variables. However, the five most relevant features give the most useful information to the algorithm and the least two relevant features serve little or nothing to the model.

	Linear Regression (RMSE)	Logistic Regression (accuracy)	Decision Tree (accuracy)	Random Forest (accuracy)
Number Variables	9	6	4	11
Score	0.546	0.748	0.756	0.831

TABLE II

SCORE USING OPTIMAL AMOUNT OF FEATURES

The number of optimal feature depend on each independent algorithm which signifies that the way each predictor uses the data is different, even though some preprocessing steps that are independent of the learning algorithm can be done to achieve better results than those obtained when no feature selection is applied.

V. BINARY CLASSIFICATIONS MODELS

We retrain all the machine learning models by using the optimal number of features shown in Table II

A. *Hyperparameter tunning*

Hyperparameter tunning is an essential part of the majority of machine learning algorithms, choosing their right values may be the difference between an algorithm having moderate or an state of the art performance [9].

In this section, we focused on getting the best value of the hyperparameters for each of the binary classification model, we do this by proving different values and measuring the accuracy in the validation set.

1) Logistic Regression: One of the main things to define when implementing a Logistic Regression algorithm is the threshold, this is, the value that determines whether a probability is assigned to the true or the false class.

The importance of the threshold relies in its control over the precision and recall trade off. We define precision as the proportion of predicted positive that was correct and recall as the proportion of true positives that was correctly predicted. The formulas for them are:

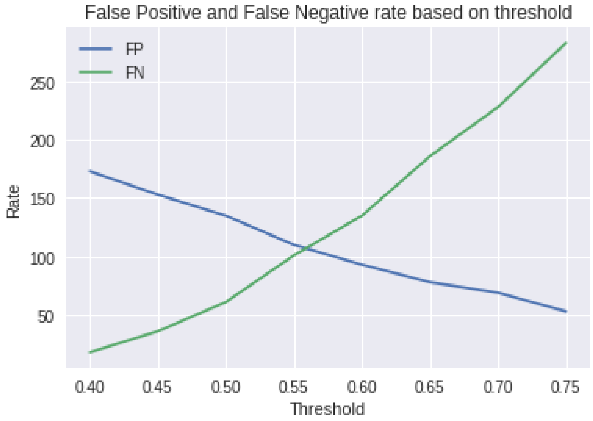

Fig. 3. False Positive and False Negative rate based on threshold

$$precision = \frac{TP}{TP + FP}$$

$$recall = \frac{TP}{TP + FN}$$

The decision between which one to give more importance depends on the specifications of the problem. In our case, as wine producers and sellers we care more about reducing the number of false positives, that is, we want to minimize the number of bad wines that are predicted as good ones. Thus, we want to maintain our precision high.

Treshold	TN	FP	FN	TP	Accuracy
.4	77	173	18	572	0.77
.45	97	153	36	554	0.775
.5	115	135	61	529	0,767
.55	140	110	101	489	0,749
.6	157	93	135	455	0,729
.65	172	78	186	404	0.686
.7	181	69	228	362	0.646
.75	197	53	283	307	0.6

TABLE III

FP FN AND ACCURACY OF LOGISTIC REGRESSION WITH DIFFERENT THRESHOLDS

We set a number of possible threshold values and see the accuracy as well rate of False Positives and False Negative with each threshold. As shown in Fig. 3. the rate of FN and FP goes in opposite direction as the threshold changes. As said earlier we want to minimize our False Positives, however, we should maintain a balance between False Positive and False Negative, not damaging the overall accuracy. We choose a threshold of 0.6, because, as seen in Table III, it reduces the number of False Postives from 135 in 0.5 to 93. However, it stills maintains a good overall accuracy of the model(0.729), giving more importance to minimizing the FP but keeping the FN reduced as well.

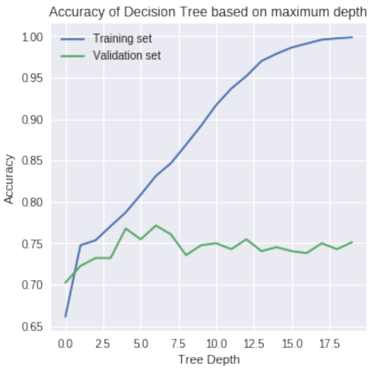

Fig. 4. Accuracy of Decision Tree based on maximum depth

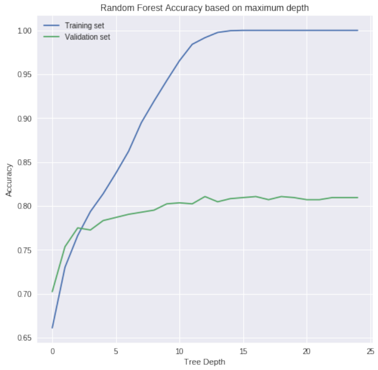

Fig. 5. Random Forest Accuracy based on maximum depth

2) Decision Tree: One of the main problems to face when implementing Decision Trees is that they tend to overfitt. When we compare the training and the validation set we get that the trainning accuracy is 0.9996 and the validation accuracy is of 0.7476, therefore, we confirm that our model is overfitting. One thing to do when facing this problem is to use regularization, we use the max_depth parameter as a regularization technique.

Fig. 4. shows that after a depth of 6 the decision tree accuracy does not improve, therefore, we apply a max_depth of 6 and get a training score of 0.787 and a validation score 0.768. Our model is no longer overfitting and its accuracy has improve slightly

3) Random Forest: Random forest is based in assembling a set of Decision Trees, using the bagging method, which affirms that a set of learning models increases the overall result.
As one may expect it also has the problem of overfitting. By evaluating our model we get that the training score is of 1 and the validation score is of 0.808, thus, it has overffited. We once again use the max_depth parameter as a method for regularization.

Fig. 5. illustrates the different accuracy of the random forest algorithm based on the maximum depth, it is clear that after the 13 layer the model does not improve its performance, thus, we set the maximum depth to 13. Applying this regularization technique our training score changes to 0.992 and out validation score changes to 0.811. Even though the difference between the training and the test is still large, it has reduce significantly and our model accuracy has increase lightly

B. Model Evaluation

In this section we evaluate each model performance by evaluating different measurement metrics on data that the model has not seen before.

1) Logistic Regression: As seen before, we care to minimize the number of False Positive, thus, keeping a high precision. Fig. 6. shows the confusion matrix of the this model, in addition, we achieved a precision of 0.7749, a recall of 0.875 and an accuracy of 0.748

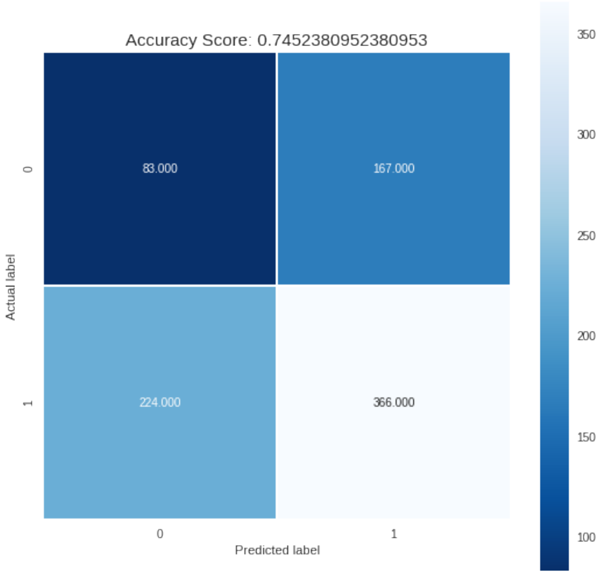

Fig. 6. Logistic Regression confusion matrix

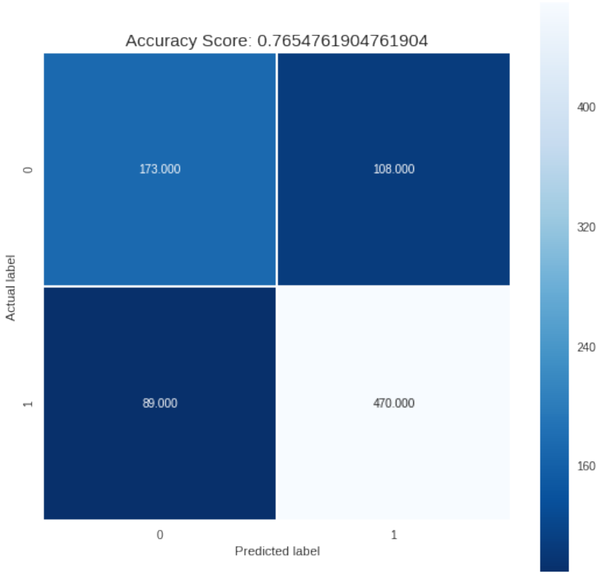

Fig. 7. Decision tree confusion matrix

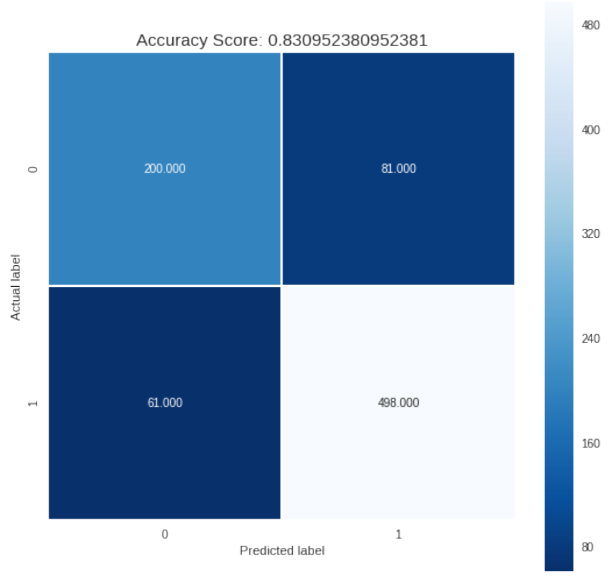

Fig. 8. Random Forest confusion matrix

2) Decision Tree: The decision tree achieved an accuracy of 0.765, a precision of 0.813 and a recall of 0.8407. The confusion matrix can be seen in Fig. 7.

C. Random Forest

The Random Forest model achieved a 0.831 accuracy, a precision of 0.85 and a recall of 0.89. The confusion matrix can be seen in Fig. 8.

One can clearly see that as the model complexity increases, so the performance of the algorithms.

VI. Linear Regression Model

This model is the simplest model we implement, it is based in finding the optimal parameters that best fit the data, minimizing the distance between the examples and the predictions.

In order to measure the model performance we use the Mean Square Error (MSE)

$$RMSE = \frac{1}{2m}\sum_{i=1}^{m}(\hat{y} - y)^2$$

We get a RMSE of 0.5463, which is still high, however, it is still a reasonable performance for the simplity of the algorithm.

In the same way, we used the determination coefficient r^2 that allow us to measure how close are the predictions from the regression line. In other words, it is the percentage of the dependent variable variance that can be explained collectively by the independent variables

We get a R2 of 0.285, a low value, which means that only 28.5% of the variance of the wine quality can be predicted by the independent variables in the model. This makes sense since the model is way to simple to be able to predict with high accuracy.

VII. Conclusions

The results of this paper shows the importance of different steps in the implementation of machine learning algorithms. It is important to do feature selection in the majority of machine learning algorithms, having in mind that the way in which each model treats the data is different, thus, it may be a good idea to select the features using a method that takes into account the algorithm. However, there are some methods that are independent from the algorithm that can give even better results than those obtain with no feature selection.

Additionally, choosing the right values of the hyperparameter changes the performance of the algorithm and should be considered a crucial step in the majority of machine learning models.

Furthermore, we saw how the complexity of the model affected the its performance. In general, the more complex an algorithm is the better performance it gets.

References

[1] G. O. Young, Synthetic structure of industrial plastics (Book style with paper title and editor), in Plastics, 2nd ed. vol. 3, J. Peters, Ed. New York: McGraw-Hill, 1964, pp. 1564.

[2] W.-K. Chen, Linear Networks and Systems (Book style). Belmont, CA: Wadsworth, 1993, pp. 123135.

[3] Correlation Analysis - Market Research", Djs-research.co.uk, 2019. [Online]. Available: https://www.djsresearch.co.uk/glossary/item/correlation-analysis-market-research.

[4] Beginner's Guide to Feature Selection in Python, DataCamp Community. [Online]. Available: https://www.datacamp.com/community/tutorials/feature-selection-python. [Accessed: 26-Mar-2019].

[5] I. Guyon and A. Elisseeff, An Introduction to Variable and Feature Selection, Journal of Machine Learning Research, vol. 3, Mar. 2003.

[6] N. Snchez-Maroo, A. Amparo Alonso-Betanzos, and M. Tombilla-Sanromn, Filter methods for feature selection. A comparative study

[7] H. Liu and M. Cocea, Semi-random partitioning of data into training and test sets in granular computing context, Granular Computing, vol. 2, no. 4, pp. 357386, 2017.

[8] H. Liu, S.-M. Chen, and M. Cocea, Subclass-based semi-random data partitioning for improving sample representativeness, Information Sciences, vol. 478, pp. 208221, 2019.

[9] M. Wistuba, N. Schilling, and L. Schmidt-Thieme, Sequential Model-Free Hyperparameter Tuning, 2015 IEEE International Conference on Data Mining, 2015.