Wine Quality and Taste Classification Using Machine Learning Models

Ankit Meena

IIT Delhi

ce1190223@iitd.ac.in

Deepak Kumar Jangir

IIT Delhi

bb1190018@iitd.ac.in

Abstract

With the recent developments in the field of machine learning, it is now possible to analyse large bulks of data from practical fields and draw out patterns and features which are required for better understanding of the producs. This paper demonstrates the use of various machine learning techniques and models to assess and predict the quality of different wines.

Introduction:

We are living in 21st century, with advancement in technology and innovation standard of life also changed. With the recent developments in the field of data science and machine learning, and the vast amounts of data being available, it has now become possible for industries to understand and adapt according to the needs of their customers with a data driven approach. Data is now becoming one of the most valuable assets for these companies which helps them compete within the market and improve the product quality.

Over the recent decade, as the demand for wine has increased, wine industries are now looking for better alternatives in terms of composition and quality of the wine produced. Wines compose of several chemical ingredients. These ingredients and chemicals differ based on conditions and purpose. These, in turn affect both the quality and taste of the wine produced. We will use these chemical compounds as our features to train our machine learning algorithm. Then we will predict the quality and rating of a wine. Our prediction will be based on different chemical composition of wine.

Literature survey:

Several attempts have been made to integrate machine learning into wine feature selection and classification in the past. Beltran [[i]] proposed an algorithm to classify wine data sets based on genetic algorithms and aroma chromatograms, Er and atasoy [ii] also proposed an approach based on support vector machine (SVM) classifier for classifying the quality of wine. Several other attempts based on various techniques such as central clustering method, random forest method, etc. have been made for the purpose of classification based on features of wines.

Problem Statement

We wish to find the different correlations between various features that affect the quality of a wine and it's classification as good or bad wine. We train machine learning algorithmic models for the prediction of wine quality (as good and bad) based on remaining 11 features and then compute the precision, f1-score and support of our predictions made by various models. So, for the wine dataset, 'quality' will be the only dependent variable and remaining all 11 features will be independent variable. trying to find out the best fit to this prediction function F(X) by choosing different type of models, fitting to our wine data, and comparing the precision, f1-scores, and supports for the predictions for various models.

Data collection:

The wine dataset we've used is publicly available for access from open-source data base of UCI machine learning repository [iii] provided by the kaggle community. This contains information about 1599 samples of white wines and 4898 samples of red wines. Which include 11 input parameters based on chemical component of the wine and 1 output parameter quality review rating between 0 to 10 given by human expert.

We have total 11 features for our wine dataset which are:

1. Fixed acidity, Volatile Acidity, Citric Acid, Chlorides, Free SO2, Total SO2, Density, PH values, Sulphate, Alcohol

Importing the data set:

we first import our data and check for any missing values in the columns.

Data visualization:

Bivariate analysis: Graphical visualization and analysis between two variables to determine the empirical relation between them. One of these two variables will be dependent variable ('quality') and second one will be independent variable. We are going to use two in-build python libraries (matplotlib.pyplot as plt and seaborn as sns) and their in-build methods

Bivariate Analysis between	**Result**
Fixed Acidity and quality	No direct relation
Alcohol and quality	Amt. of alcohol increases with quality
Citric acid and quality	Mean of citric acid increases with quality

So, by the bivariate analysis we can see that for high quality of wine volatile acidity should decrease but citric acid and alcohol should increase. we also observed that the citric acid concentration is proportional to the quality of wine.

Feeding Data To Various Models.

The data was first splitted into training and test data by a ratio of 70:30 from the wine dataset. Then, scaling and preprocessing (of mean, variances) was done so as to transform the data to a suitable form to feed to inbuilt libraries of various classification models.

Before moving to the mathematical terminologies of models used, defined below are some terms which are used to describe the results.

Confusion matrix:

$$\begin{array}{ccc} 2x2\ confisuion\ matrix & positive\ prediction & negative\ prediction \\ positive\ case & true\ positive & false\ negative \\ negative\ class & false\ positive & true\ negative \end{array}$$

Precision:

$$precision = \frac{true\ positive}{true\ positive + false\ negative}$$

Recall:

$$recall = \frac{true\ positive}{true\ positive + false\ positive}$$

F-measure:

$$f - score = \frac{2 * precision * recall}{precision + recall}$$

Mathematical Terminologies of various models used:

1. **Logistic regression model:** So, we have 4547 training examples with 11 features (independent variables) and 1 output variable (quality) in our training data.

   $feature\ vector\ X = \{x_1, x_2, x_3, \dots \dots \dots ., x_{11}\}$

   $$training\ data = S = \{X^1, X^2, \dots \dots \dots, X^{4547}\}$$

   Aim is to predict the quality of the wine such that

   $$y = \begin{cases} 0, & h_\theta(x) < 0.5 \\ 1, & h_\theta(x) \geq 0.5 \end{cases}$$

   Here $h_\theta(x)$ is a sigmoid function. Which is defined as

   $$h_\theta(x) = \frac{1}{1 + e^{-\theta^T X}}$$

   $$\theta^T = [\theta_1, \theta_2, \theta_3, \dots \dots ., \theta_{11}]^T$$

   Now, cost function for logistic regression is

$$J(\theta) = \frac{1}{4547} \sum_{i=1}^{4547} cost(h_\theta\left(x^{(i)}\right), y^{(i)})$$

$$cost(h_\theta(x), y) = \begin{cases} -\log(h_\theta(x)), & y = 1 \\ -\log(1 - h_\theta(x)), & y = 0 \end{cases}$$

So, by combining both cost functions for y = 1and y =0 we get

$$J(\theta) = \frac{1}{4547} \sum_{i=1}^{4547} [y^{(i)} * \left(-\log\left(h_\theta\left(x^{(i)}\right)\right)\right) + (1 - y^{(i)}) * (- - \log\left(1 - h_\theta(x)\right))]$$

Now, we require the values of the θ for which the cost function is minimum. So, to fulfill this, gradient descent is used where the following iteration for each parameter θ choosing any random values of θ and $\propto$.

$$\theta_j = \theta_j - \propto \frac{d\,J(\theta)}{d\theta_j}$$

This will optimize the θ. This value of θ we will put in sigmoid function and then on a new input feature vector we will predict our values.

2. **Stochastic gradient descent:** we want a balance between goodness of the gradient descent (values of initial θ) and speed (depend up on $\propto$). This is achieved by running gradient descent on a small number of random data points. running gradient descent on multiple data points at a same time. This is similar to logistic regression except we are running gradient descent for only one data point, we are running it on multiple data points in parallel.

3. **Support vector machine:**
Target- To predict the quality of the wine such that

$$y = \begin{cases} 0, & \sum_{i=1}^{11} \theta_i^T f_i < 0 \\ 1, & \sum_{i=1}^{11} \theta_i^T f_i \geq 0 \end{cases}$$

Here $h_\theta(x)$ is a sigmoid function. Which is defined as

$$f_i = similarity\left(x, l^{(i)}\right) = \exp\left(\frac{-\left\|x - l^{(i)}\right\|}{2 * \sigma^2}\right)$$

$$\theta^T = [\theta_1, \theta_2, \theta_3, \ldots\ldots., \theta_{11}]^T$$

$l^{(i)}$
$= landmark\ data\ points\ near\ the\ boundary\ achieved\ by\ visiualization\ of\ data$

$$\sigma = variance$$

overall cost function is

$$J(\theta) = \sum_{i=1}^{4547} [y^{(i)} * cost_1(\theta_i^T f_i) + \left(1 - y^{(i)}\right) cost_0(\theta_i^T f_i)] + \frac{1}{2}\sum_{j=1}^{m} \theta_j^2$$

The model uses the same optimization technique as gradient descent to find θ for which cost function is minimum.

4. **Decision tree classifier:**
define Root node = training data = S.
Now, we compute Information gain (IG) and Gini index for every feature vector x_i.

$$IG(x_i) = E(S) - P(y = 1|x_i) * E(x_i > constant)$$
$$E(A) = \ -P(A) * \log_2(P(A))$$

Where $IG(x_i)$ is information gain of x_i
E(S) = entropy of training data
And $E(x_i > constant)$ is entropy of x_i , when x_i is greater than a particular value.
And $P(A)$ is frequentist probability of A.

$$Gini\ index = 1 - \sum_{i=1}^{11} P_i^2$$

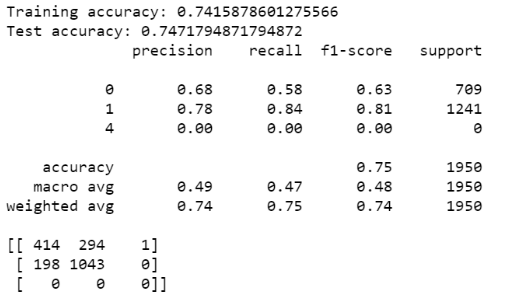
Feature for which Information gain is maximum and Gini index is minimum is the basis of splitting at root node.
splitting our data in the same pattern up to left with only one feature. Then we fill do a final split into two leaf nodes. So, at every decision we are getting branch like structure with a root (node) on the top. It looks like s upside down tree. So, when we give a new feature vector this classify it and give a corresponding leaf node. Which contain the quality of the wine.

5. **Random forest classifier:** the only difference between random forest and decision tree classifier is that decision tree uses training data as root node but random forest classifier makes multiple samples of the training data and run multiple decision trees. Then to split the data at node we using ensemble method on these different trees. And decision (feature) with maximum votes is selected for splitting of the training data.
Root nodes = $S_1, S_2, \ldots \ldots \ldots, S_n\ is\ subsets\ of\ S.$
So, by using n random samples of S we create n weak decision tree. Formation of decision tree is exactly same as we did in decision tree classifier. Then by using ensemble method we choose the right decision.

Performance comparison of different classifiers used:

Conclusion:

Random Forest Classifier showed the highest Test Accuracy of 82.05% and decision Tree Classifier performed the poorest with an test accuracy of 72.56%.



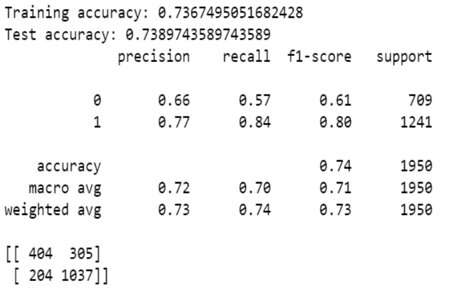
```
Training accuracy: 0.7415878601275566
Test accuracy: 0.7471794871794872
              precision    recall  f1-score   support

           0       0.68      0.58      0.63       709
           1       0.78      0.84      0.81      1241
           4       0.00      0.00      0.00         0

    accuracy                           0.75      1950
   macro avg       0.49      0.47      0.48      1950
weighted avg       0.74      0.75      0.74      1950

[[ 414  294    1]
 [ 198 1043    0]
 [   0    0    0]]
```

Figure 1 logistic regression



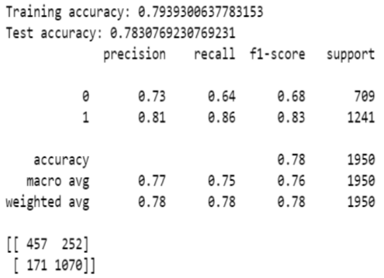
```
Training accuracy: 0.7367495051682428
Test accuracy: 0.7389743589743589
              precision    recall  f1-score   support

           0       0.66      0.57      0.61       709
           1       0.77      0.84      0.80      1241

    accuracy                           0.74      1950
   macro avg       0.72      0.70      0.71      1950
weighted avg       0.73      0.74      0.73      1950

[[ 404  305]
 [ 204 1037]]
```

Figure 2 stochastic gradient descent



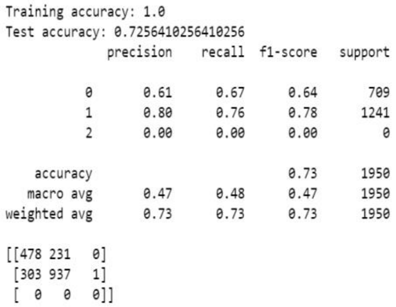
```
Training accuracy: 0.7939300637783153
Test accuracy: 0.7830769230769231
              precision    recall  f1-score   support

           0       0.73      0.64      0.68       709
           1       0.81      0.86      0.83      1241

    accuracy                           0.78      1950
   macro avg       0.77      0.75      0.76      1950
weighted avg       0.78      0.78      0.78      1950

[[ 457  252]
 [ 171 1070]]
```

Figure 3 support vector machine



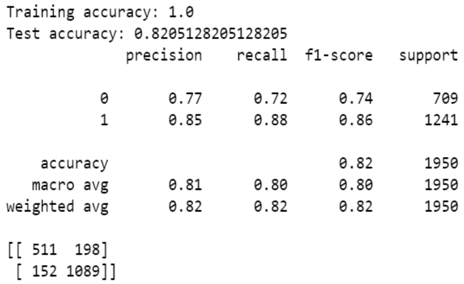
```
Training accuracy: 1.0
Test accuracy: 0.7256410256410256
              precision    recall  f1-score   support

           0       0.61      0.67      0.64       709
           1       0.80      0.76      0.78      1241
           2       0.00      0.00      0.00         0

    accuracy                           0.73      1950
   macro avg       0.47      0.48      0.47      1950
weighted avg       0.73      0.73      0.73      1950

[[478 231   0]
 [303 937   1]
 [  0   0   0]]
```

Figure 4 decision tree classifier



```
Training accuracy: 1.0
Test accuracy: 0.8205128205128205
              precision    recall  f1-score   support

           0       0.77      0.72      0.74       709
           1       0.85      0.88      0.86      1241

    accuracy                           0.82      1950
   macro avg       0.81      0.80      0.80      1950
weighted avg       0.82      0.82      0.82      1950

[[ 511  198]
 [ 152 1089]]
```

Figure 5 random forest classifier

Based on the bivariate analysis, wines with high alcohol and low volatile acidity is appreciated and given high rating.

we use different data visualization techniques to find the characteristics of the dataset. use. We also calculated the Precision, Recall, Support and F1-score of given algorithms to evaluate the model accuracy.

REFERENCES

[i] N. H. Beltran, M. A. Duarte- MErmound, V. A. S. classification using volatile organic compounds data obtained with a fast GC analyzer," Instrum. Measurement, IEEE Trans., 57: 2421-2436, 2008.

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.

[ii] Yesim Er*1, Ayten Atasoy1. "The Classification of White Wine and Red Wine According to their Physiochemical Qualities", ISSN 2147-6992147-6499, 3rd September 2016

[iii] online link for database