Classification of Wine Quality with Imbalanced Data

Gongzhu Hu, Tan Xi, Faraz Mohammed
Department of Computer Science, Central Michigan University, USA
{hu1g, xi1t, moham1f}@cmich.edu

Huaikou Miao
School of Computer Engineering and Science, Shanghai University, China
hkmiao@shu.edu.cn

***Abstract*—We propose a data analysis approach to classify wine into different quality categories. A data set of white wines of 4898 observations obtained from the *Minho* region in Portugal was used in our analysis. As the occurrence of events in the data set was imbalanced with about 93% of the observations are from one category, we applied the Synthetic Minority Over-Sampling Technique (SMOTE) to over sample the minority class. The balanced data was used to model a classifier that categorizes a wine into three categories as high quality, normal quality, and poor quality. Three different classification techniques were used: decision tree, adaptive boosting (AdaBoost), and random forest. Our experiments show that the random forest technique seems to produce the desired results with the least percentage of error.**

***Index Terms*—classification, imbalanced data, SMOTE, wine quality**

I. Introduction

A lot research have been done on wine quality that are mostly based empirical studies in the wine industry. Quality of wines is not easy to define and there are many factors that influence the perceived quality. These factors include intrinsic characteristics (visual, taste, smell), environmental (climate, region, site) and management practices (viticultural practice), as well as physicochemical ingredients (acid, pH, etc.).

In addition to the research in the food industry, machine learning techniques have also been applied to classification of wine quality. The purpose of using machine learning methods, as in many other applications, is to build models from data of known class labels to predict the quality of a wine.

We endeavor to build models to classify different wines into quality categories based a data set of 4898 instances. Three models were built and tested on the data set. The purpose of devising such a categorization is to aid wine-makers in providing a better as well as resource efficient end product. The *quality* variable is determined by several factors (variables). The analysis would give a clearer idea to wine-makers as to which variables influence the quality the most and what tweaks could be made to attain more desirable results.

One of the problems this particular data set presents is that the classes are imbalanced — 93% of the 4898 instances are from one class and only 7 % are from the other class. The models built based on this data set may be over-fitting in favor of the majority class. We applied the Synthetic Minority Over-Sampling Technique (SMOTE) to over overcome this problem in the data preparation stage.

The data analysis software system R was used throughout the entire processing in our work, from data preparation to model building to classification. We conducted classification experiments that showed that the random forest model outperformed the other two models in terms of classification errors.

II. Related Work

We shall briefly discuss related work in data analysis of wine quality and techniques dealing with imbalanced data.

A. Data Analysis of Wine Quality

The consumption of wine has been increasing over the years, particularly the wines of high quality. The quality of wine is commonly assessed by expert tasters who make their judgment based on various sensory factors, such as color, taste, and odor. However, measurement based approaches commonly use the components in the wines such as acid, pH, and level of sugar [6].

A lot of research have been done to assess wine quality using physicochemical data, but many of them are based on small sample sizes. In [12] pattern recognition approaches (clustering, principle component analysis, nearest neighbors, etc.) were applied to classify wines from Galicia (northwestern Spain) between several different brands using a data set of 42 white wines. Principle component analysis for wine classification according to geographical region was reported in [10]. The data set used in their study contains 33 greek wines with physicochemical variables. A study of 2-stage classification (principle component and clustering) from 24 industrial fermentations of a particular type of wine was given in [17] that try to detect undesirable fermentation behavior.

Several more recent work used data mining techniques to classify the quality of wines using a larger physicochemical data set, which is also the data set we used in our study. Cortez and his colleagues [5] built models using multiple regression, support vector machine, and neural networks. They developed

978-1-4673-8075-1/16/$31.00 ©2016 IEEE

a computational procedure that performs simultaneous variable and model selection. Their support vector machine achieved desirable results, "outperforming the multiple regression and neural network methods." Their model is useful to support the oenologist wine tasting evaluations and improve wine production. Appalasamy et al. [1] applied two classification algorithms, decision tree and Naïve Bayes and compared results with the one in [5].

Our study presented in this paper is similar to the work in [1], [5], but our work is different in two aspects: (a) we pre-processed the imbalanced data using SMOTE, and (b) we build different models, as shown in Table I.

TABLE I
SUMMARY OF DIFFERENCES BETWEEN STUDIES

		Work		
		[1]	[5]	Ours
Handle Imbalance Data				✓
Algorithm	Regression		✓	
	Support vector machine		✓	
	Neural network		✓	
	Decision tree	✓		✓
	Naïve Bayes	✓		
	AdaBoost			✓
	Random forest			✓

B. Handling Imbalanced Data

The problem of imbalanced data refers to the situation that the proportions of the data records in the data set are very imbalanced among classes. In the class with more data points is commonly called *majority* class while the one with fewer data points is called *minority* class. For example, a sample of 4898 data records in our study contains 4535 normal quality wines and 363 high/low quality wines. This sample is likely considered imbalanced as there are 12.5 times more records in the majority class than the ones in the minority class. Analysis using imbalanced data often cause bias and have high misclassification errors. However, the imbalanced data problem is quite common in some applications, such as a male-female imbalanced sample that may occur when data are obtained from engineering schools or nursing schools.

Various methods have been proposed to handle imbalanced data [7]. Data-level approaches solve this problem mostly by re-sampling the data space, either under-sampling or over-sampling. Under-sampling is to remove instances from the majority class randomly or using some heuristics, such as removing noisy and borderline instances [15], removing redundant instances [11], and using Neighborhood cleaning rule [13]. Over-sampling approaches replicate instances of the minority class randomly or create new instances using heuristics. Synthetic Minority Over-Sampling Technique (SMOTE) [4] is a commonly used over-sampling method that generate new minority class instances by interpolating data instances in certain neighborhood. Several variations of SMOTE were also proposed such as borderline-SMOTE [8].

The imbalanced class distribution problem can also be handled at algorithm level. For example, an inductive bias toward small classes can be considered in the learning algorithms. Since we use SMOTE to handle imbalanced data problem, we shall not discuss algorithm level solutions here.

III. METHODOLOGY

A. Data Preparation

The data set contains 4898 instances of white wines from the UCI Machine Learning Repository [16]. There are 11 physicochemical (inputs) variables that influence the quality of wines, as shown in Table II.

Tartaric acid, citric acid and malic acid are present in wine, while generally, ascorbic, sorbic and sulfurous acids are added during wine making. Residual sugar determines the *sweetness* of a wine. Although it is not the only factor which determines the sweetness, its still plays a major role in determining the taste of a wine. Alcohol in wine, is a by-product of yeast metabolism.

The *quality* variable in the data set varies from 3 to 9 with 3 being the poorest quality, while 9 denoting the highest quality. Interestingly, values 1, 2, and 10 do not exist. The values of the quality variable and their "class labels" are:

- Low Quality: 3, 4
- Normal: 5, 6, 7
- High Quality: 8, 9

The distribution of the instances in the data set on quality value is shown in Fig. 1(a) and on quality class is shown in Fig. 1(b).

The events are clearly not equally dispersed. We see that for quality "Normal", which pertains to values 5, 6 and, 7, has the highest occurrence. Among 4898 instances, these occur 4535 times. Hence these could be regarded as *Normal* occurrences. On the other hand, there are only 183 instances of 'Low' quality (values 3 and 4) and 180 instances of 'High' quality (values 8 and 9). We call these instances of low and high quality *Rare* occurrences.

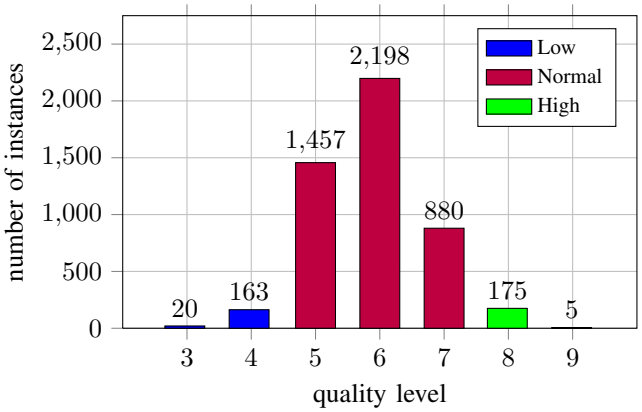
B. SMOTE

The Synthetic Minority Oversampling Technique (SMOTE) is an over-sampling method to address the problem of imbalanced distribution of data. The basic idea is to re-sample the data space to create more synthetic points of the rare class.

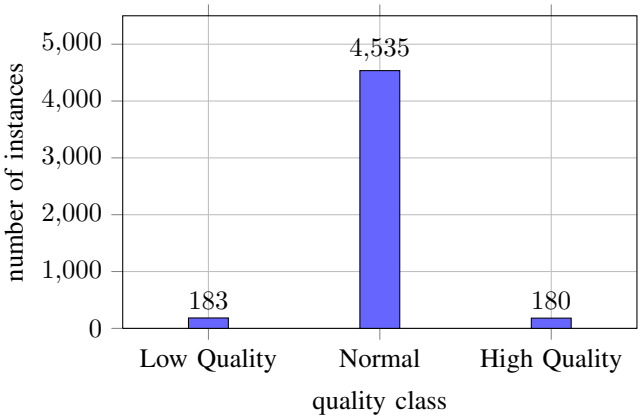
Here is how the basic SMOTE algorithm works. Let D be the data set, $R \subset D$ be the set of minority class. The algorithm iterates through for each $r \in R$, identifies the k nearest neighbor N of r based on some distance measure, where k is a parameter. Then, the algorithm randomly picks a point $n \in N$, interpolating between r and n to create a synthetic point p. The new point p is to be added to the minority class. This is illustrated in Fig. 2, where 2(a) shows the two classes (minority class has only one point) and 2(b) shows the 4 nearest neighbors of the minority point and the newly generated synthetic point by SMOTE.

TABLE II
PHYSICOCHEMICAL VARIABLES IN RED AND WHITE WINES [16]

Variable	Description
Fixed acidity	Acidity refers to the "fresh, tart and sour" attributes of a wine [2].
Volatile acidity	Organic acids that are more volatile or more easily vaporized than fixed acid.
Citric acid	One of the nonvolatile acid present in wine that can add "freshness" to wine.
Residual sugar	Amount of sugar after the fermentation process.
Chlorides	Amount of salt in the wine.
Free sulfur dioxide	Amount of free form of S02 preventing microbial growth and the oxidation of wine.
Total sulfur dioxide	Amount of free and bound forms of sulfur dioxide gas (S02).
Density	density of water depending on the percent alcohol and sugar content.
pH	level (0–14) of acidity or alkalinity of a solution. Most wines are between 3.3–3.7.
Sulphates	A wine additive which can contribute to sulfur dioxide gas (S02) levels.
Alcohol	Percent of alcohol content of the wine.

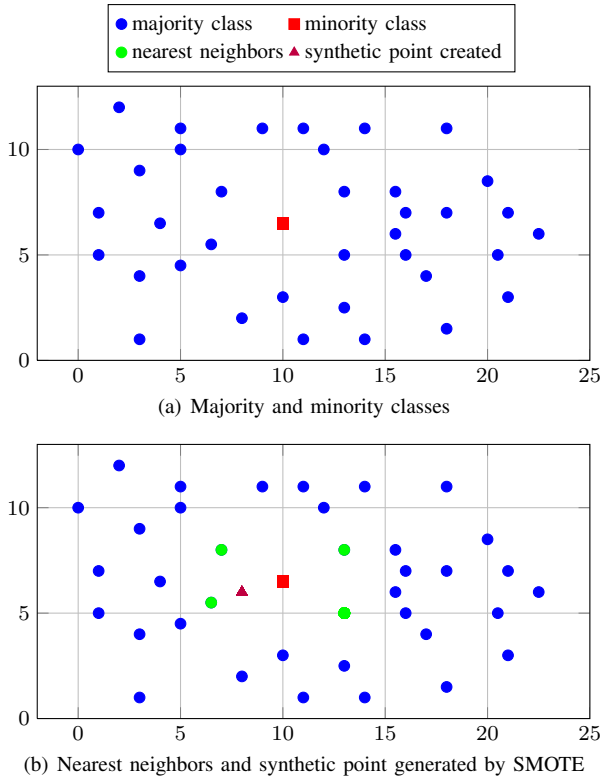


(a) Quality distribution



(b) Targte distribution

Fig. 1. Data distribution on quality and classes



(a) Majority and minority classes

(b) Nearest neighbors and synthetic point generated by SMOTE

Fig. 2. SMOTE process steps

There are several variations of the SMOTE algorithm, some of which consider heuristics and/or use different distance metrics in determining the nearest neighbors.

The **R** code to apply the SMOTE algorithm is quite simple as shown below.

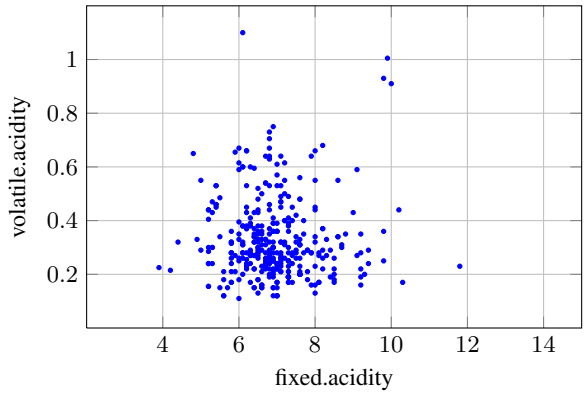
```
%[frame=single]
table(wine_data $ Target)
wine <- wine_data
names(wine)
library(DMwR)
form <- formula(Target ~.)
SMOTEData <- SMOTE(form, wine, perc.over = 600,
                   k = 5, perc.under = 100)
table(SMOTEData $ Target)
write.csv(SMOTEData, file = "SMOTE_WINE.csv",
                     row.names = FALSE)
```

In this code, the `wine_data` (categorized into normal and rare classes as the `Target`) is fed to the `SMOTE` algorithm with the over-sampling parameter 600 for generating cases of the minority class (rare), under-sampling parameter 100 to select cases from the majority class (normal) for each generated minority class case, and the size of nearest neighbors being 5. The result is then written to a file as the new balanced

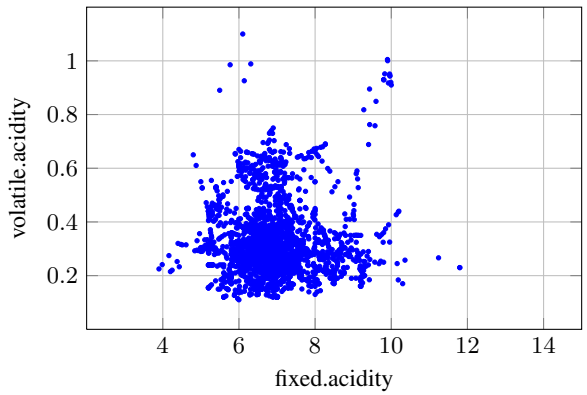
data set for classification.

The normal and rare cases in the original data set were 4535 and 363; and the numbers became 2178 and 2541 after SMOTE algorithm was applied.

Just for illustration, the data points of the rare class are plotted on two variables shown in Fig. 3. It is seen that many more cases were generated after over-sampling while maintaining the same data distribution.



(a) Original data



(b) After applying SMOTE

Fig. 3. Data set (fixed.acidity vs volatile.acidity) before and after applying SMOTE

C. Models

Three machine learning methods were used in our study. A model was built using each method and applied to the original imbalanced data and the balanced data after SMOTE. The basic ideas of the three methods are given here.

Decision Tree:

Decision trees are a method of plotting all the possible outcomes of an event, using a tree-like graph system. Decision trees can be used to obtain a possible set of consequences, and are advantageous if the variables are not inter-related or inter-linked.

AdaBoost:

AdaBoost is short for "Adaptive Boosting". It's a machine learning algorithm used in conjunction with other learning algorithms to improve their performance [9].

Random Forest:

Random Forest is an *Ensemble* learning method, used for classification [3]. Ensemble methods utilize multiple learning algorithms to obtain better predictive performance that can not be obtained by using a simple learning algorithm [14].

These are among the most commonly used methods for classification tasks. Other methods can also be used, as did in some previous work mentioned in Section II.

IV. Experimental Results

A. Results of Error Rates and ROC

We used System **R** for data preparation and analysis. The three classification methods (Decision Tree, AdaBoost, Random Forest) were applied on data before and after applying the SMOTE algorithm to balanced the data. For each model, we used 75% of the data instances for training, 15% for validation, and 15% for testing.

For each run, we calculated some of the standard performance measures (statistics) to evaluate the performance of the algorithms, including error rate, specificity, and ROC values. Classification results have four possible outcomes: true positive (TP), true negative (TN), false negative (FN), and false positive (FP), as given in Table III.

TABLE III
Terms used to define performance measures

Test Outcome	**Standard of Truth**	
	Positive	Negative
Positive	TP	FP
Negative	FN	TN

The measures commonly used to describe the tests are

$$\begin{aligned} Sensitivity &= TP\,/\,(TP+FN) \\ Specificity &= TN\,/\,(TN+FP) \\ Accuracy &= (TN+TP)\,/\,(TN+TP+FN+FP) \\ ErrorRate &= 1-Accuracy \end{aligned}$$

The classification results before and after applying the SMOTE algorithm are given in Table IV(a) and Table IV(b), respectively.

The results show that Random Forest has the lowest average error rate among the three methods. Since Random Forest had the lowest error rate, we ran the Random Forest algorithm with 100–600 trees to see how the error rate is affected by the number of trees. The results on the balanced data set is shown in Fig. 4. It appears that the error rates dramatically decreased from a few trees to about 50 trees, and stabled at about 4% for rare cases and about 10% for normal cases after 200 trees. The OOB (out-of-bag) measures are between the two.

The results in Table IV also show that the error rates of Decision Tree and AdaBoost increased significantly on the rebalanced data set after SMOTE was applied. This is because the models built on the original date were over-fitted with bias to the majority class, and most of the instances in the

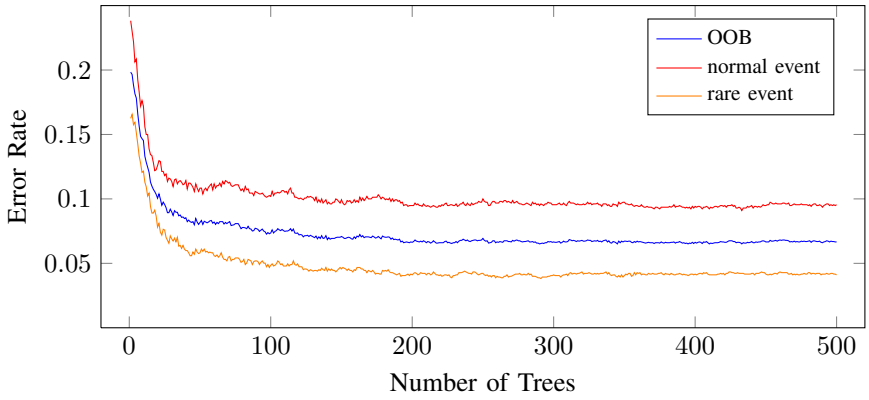


Fig. 4. Rate of Errors with the random forest model

TABLE IV
CLASSIFICATION RESULTS

(a) Before applying SMOTE

Models	**Error Rate**	**Specificity** $(TN\,/\,N)$	**ROC**
Decision tree	7.2%	0/53	0.50
AdaBoost	6.8%	4/53	0.83
Random Forest	5.4%	17/53	0.88

(b) After applying SMOTE

Models	**Error Rate**	**Specificity** $(TN\,/\,N)$	**ROC**
Decision tree	28.7%	276/388	0.73
AdaBoost	12.4%	350/388	0.93
Random Forest	4.7%	373/388	0.99

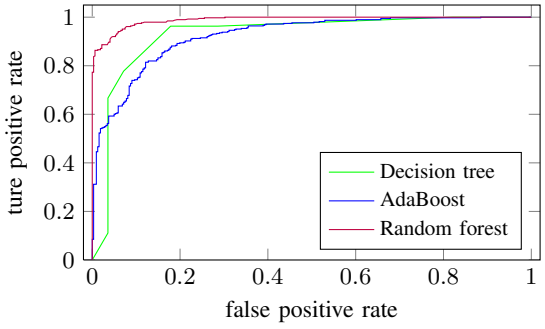
testing set are also in the majority class that resulted in higher accuracy (lower error rate). The accuracy would be 100% if all instances in the data set are in the same class, which is the extreme case of unbalanced distribution. Such accuracy is not a good measure for model's performance. A better metric is the ROC value, which is defined as $ROC = TPR\,/\,FPR$, where TPR is the true positive rate and FPR is the false positive rate, and $TPR = Sensitivity$ and $FPR = 1 - Specificity$.

As seen in Table IV, the average ROC values improved quite a lot after the data set was balanced. The ROC charts are shown in Fig. 5 at different false positive rates.

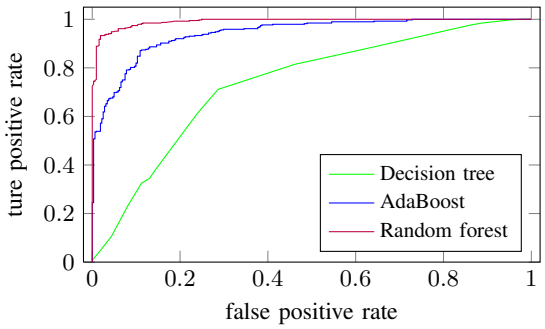
To compare our results with the work of others, particularly the work in [1] and [5] that used the same data set, we converted the confusion matrices (that are 7×7 for seven quality classes) into 3×3 matrices (for high, normal, and low quality classes) shown in Table V.

TABLE V
BINARY CONFUSION MATRICES CONVERTED FROM [1] AND [5]

	Prediction in [1]			Prediction in [5]		
	Low	Normal	High	Low	Normal	High
Low	148	126	0	18	166	0
Normal	82	4365	11	10	4516	9
High	0	165	12	0	122	58



(a) before applying Smote



(b) after applying Smote

Fig. 5. ROC curves for AdaBoost and Random Forest models

This gives accuracy of 92.0% in [1] (Decision Tree) and 93.7% in [5] (Support Vector Machine), respectively, or error rate of 8.0% and 6.3%. The comparison of the error rates on the same data set (without re-balancing) are given in Table VI.

TABLE VI
COMPARISON OF ERROR RATES

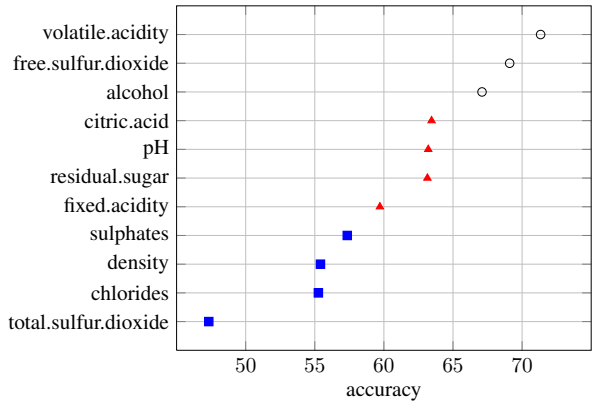
		Prediction		
		[1]	[5]	Ours
Algorithm	Support vector machine		6.3%	
	Decision tree	8.0%		7.2%
	AdaBoost			6.8%
	Random forest			5.4%

Note that this comparison may be a bit misleading because all used existing algorithms; neither of these studies (including ours) created their own algorithms. The differences of the error rates were caused by the particular implementations of the algorithms, rather than by the methods used in our analysis. For example, the Decision Tree used in [1] was ID3 in the software tool Weka, whereas we used the Decision Tree in system **R**.

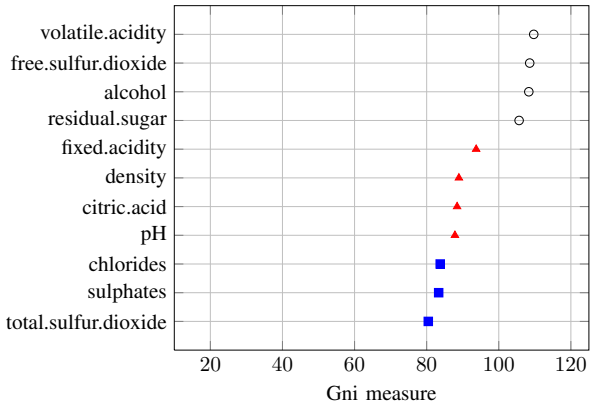
The point we want to make is that our study considers classification accuracy for high, normal, and low wine quality instead of 3–9 quality levels as did in [1] and [5]. For this reason, the over-fitting problem of imbalanced classes becomes important.

B. Importance of Variables

We also calculated the mean decrease accuracy and mean decrease Gini values for all the 11 variables to identify those variables that have the lowest values so that they are considered to play an important role in determining the quality of a wine. The results are given in Fig. 6, where 6(a) shows the mean decrease accuracy and 6(b) shows the mean decrease Gini.



(a) mean decrease accuracy



(b) mean decrease Gini

Fig. 6. Importance of the variables

The results indicate that *volatile acidity*, *free sulfur dioxide* and *alcohol* have the highest values, while *total sulfur dioxide* plays the least role in determining the quality of a wine.

V. Conclusion and Future Work

We presented an analysis of wine quality on the data set of 4898 white wines using three classification methods. Since we were only interested in distinguishing between high, normal, and poor qualities, the instances in the data set of 7 quality levels were grouped into three classes. This grouping resulted in imbalanced classes that might cause over-fitting problem. The Synthetic Minority Oversampling Technique (SMOTE) was used to efficiently solve the problems created by imbalanced data. In conjunction with SMOTE, Random Forest gave us the best results in our experiments in terms of error rates as well as ROC values.

We are currently working on applying other classification algorithms to the data set, including Bayes and neural network. In addition, we plan to apply variations of SMOTE to imbalanced data sets in other application domains.

References

[1] P Appalasamy, A Mustapha, N. D. Rizal, F Johari, and A. F. Mansor. Classification-based data mining approach for quality control in wine production. *Journal of Applied Sciences*, 12(6):598–601, 2012.

[2] R. B. Beelman and J. F. Gallander. Wine deacidification. *Advances in Food Research*, 25:1–53, 1979.

[3] Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[4] Nitesh V Chawla, Kevin W Bowyer, Lawrence O Hall, and W Philip Kegelmeyer. SMOTE: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, 16:321–357, 2002.

[5] Paulo Cortez, António Cerdeira, Fernando Almeida, Telmo Matos, and José Reis. Modeling wine preferences by data mining from physico-chemical properties. *Decision Support Systems*, 47(4):547–553, 2009.

[6] Susan E Ebeler. Linking flavor chemistry to sensory analysis of wine. In *Flavor Chemistry*, pages 409–421. Springer, 1999.

[7] Qiong Gu, Zhihua Cai, Li Zhu, and Bo Huang. Data mining on imbalanced data sets. In *Proceedings of International Conference on Advanced Computer Theory and Engineering*, pages 1020–1024. IEEE, 2008.

[8] Hui Han, Wen-Yuan Wang, and Bing-Huan Mao. Borderline-SMOTE: a new over-sampling method in imbalanced data sets learning. In *Advances in intelligent computing*, pages 878–887. Springer, 2005.

[9] Trevor Hastie, Robert Tibshirani, Jerome Friedman, and James Franklin. The elements of statistical learning: data mining, inference and prediction. *The Mathematical Intelligencer*, 27(2):83–85, 2005.

[10] S Kallithraka, IS Arvanitoyannis, P Kefalas, A El-Zajouli, E Soufleros, and E Psarra. Instrumental and sensory analysis of greek wines; implementation of principal component analysis (pca) for classification according to geographical origin. *Food Chemistry*, 73(4):501–514, 2001.

[11] Miroslav Kubat, Stan Matwin, et al. Addressing the curse of imbalanced training sets: one-sided selection. In *Proceedings of the 14th International Conference on Machine Learning*, volume 97, pages 179–186. Nashville, USA, 1997.

[12] Maria J Latorre, Carmen Garcia-Jares, Bernard Medina, and Carlos Herrero. Pattern recognition analysis applied to classification of wines from galicia (northwestern spain) with certified brand of origin. *Journal of Agricultural and Food Chemistry*, 42(7):1451–1455, 1994.

[13] Jorma Laurikkala. *Improving identification of difficult small classes by balancing class distribution*. Springer, 2001.

[14] Lior Rokach. Ensemble-based classifiers. *Artificial Intelligence Review*, 33(1-2):1–39, 2010.

[15] Ivan Tomek. Two modifications of CNN. *IEEE Trans. Syst. Man Cybern.*, 6:769–772, 1976.

[16] UCI Machine Learning Repository. Wine quality data set. https://archive.ics.uci.edu/ml/datasets/Wine+Quality.

[17] Alejandra Urtubia, J Ricardo Pérez-Correa, Alvaro Soto, and Philippo Pszczólkowski. Using data mining techniques to predict industrial wine problem fermentations. *Food Control*, 18(12):1512–1517, 2007.