# Course: Symbolic Learning
# EM-DMKM

Dominik Cygalski

May 3, 2012

## 1 Choice of the dataset

The name of the chosen dataset is "Wine Quality" and it has been obtained from UCI Machine Learning Repository[1]. It was introduced firstly by P. Cortez [3] and aimed to predict the quality of the wine based on its physiochemical properties. Although different data mining experiments had been performed on wine data[2] this was the first large dataset introduced in this area and has been even subject to a data mining competition[3]. In fact, it consists of two sets of observations: about white wines and about red wines that were collected from May 2004 to February 2007 by computerized system during the certification process. Described wines come from the Portuguese region of *vinho verde*[4].

Original motivation of study of the wine dataset was to support various aspects of wine industry, which is important due to the significant growth of this market in recent years. Not only can automatic classification tools ease certification process (e.g. by making it more cost-efficient) but thay can also help

[1] http://archive.ics.uci.edu/ml/datasets/Wine+Quality/
[2] For example on famous Wine dataset http://archive.ics.uci.edu/ml/datasets/Wine/
[3] http://www.crowdanalytix.com/contests/cheers-predict-wine-quality/
[4] http://www.vinhoverde.pt/en/

performing different marketing activities – e.g. "stratify wines such as premium brands" [3]. Finally, thorough data analysis can lead to the conclusions that can be further utilized by wine-makers in the process of production.

### 1.1 Previous works

In [3] three algorithms were applied to classify white wines with encouraging results. In that work, Support Vector Machine outperformed Neural Networks and Multiple Regression with overall accuracies of 64.3% (for Tolerance=0.5) and 86.8% (for Tolerance=1.0). It will be interesting to see if it is difficult to achieve similar results for the **Neural Networks** as authors of that paper and also check the performance and compare results of other algorithms on the dataset – **K-Nearest Neighbours** and **Decision Trees** will be tested.

### 1.2 Document outline

The document has the following structure. In Section 2 more detailed description of the dataset is provided. Section 3 focuses on tuning parameters of three algorithms based on simple metrics. Later, in Section 4 other metrics such as kappa and accuracy of classification for different tolerance levels are computed so that comparison between algorithms and other publications is possible.

## 2 Data description

| **Attribute** | **Min** | **Mean** | **Max** | ***units*** |
|---:|---|---|---|---|
| fixed acidity | 3.80 | 6.85 | 14.2 | $g(tartaric\ acid)/dm^3$ |
| volatile acidity | 0.08 | 0.28 | 1.10 | $g(acetic\ acid)/dm^3$ |
| citric acid | 0.00 | 0.33 | 1.66 | $g/dm^3$ |
| residual sugar | 0.60 | 6.40 | 65.8 | $g/dm^3$ |
| chlorides | 0.01 | 0.05 | 0.35 | $g(sodium\ chloride)/dm^3$ |
| free sulfur dioxide | 2.00 | 35.3 | 289 | $mg/dm^3$ |
| total sulfur dioxide | 9.00 | 138 | 440 | $mg/dm^3$ |
| density | 0.99 | 0.99 | 1.04 | $g/cm^3$ |
| pH | 2.72 | 3.19 | 3.82 | — |
| sulphates | 0.22 | 0.49 | 1.08 | $g(potassium\ sulphate)/dm^3$ |
| alcohol | 8.00 | 10.5 | 14.2 | *% vol.* |

Table 1: Descriptive statistics of the variables.

This work will focus only on the white wines dataset which contains 4898 observations with no missing values. Eleven continuous numerical attributes

describe standard physiochemical properties of the wine that were measured in the laboratory. Names of those attributes are self-explanatory and their basics statistics are presented in the Table 1.

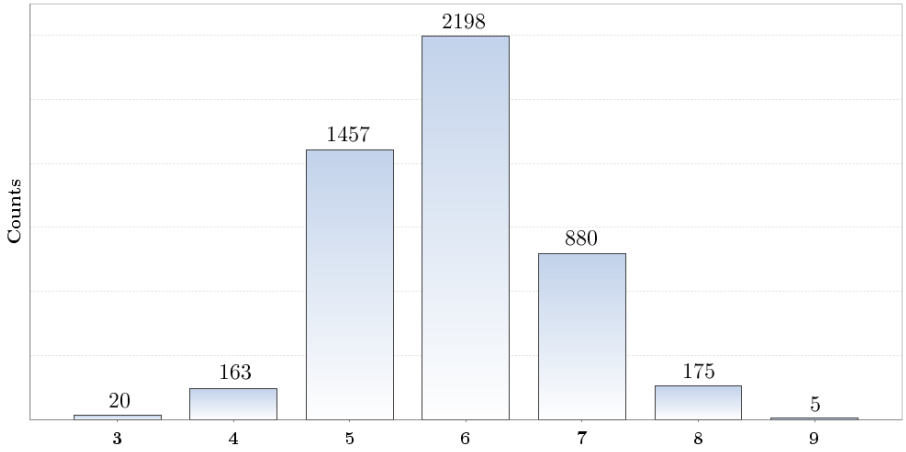


Figure 1: Distribution of Quality class in the dataset.

Wine Quality class is the attribute to be discriminated. It was obtained in the following way. Each sample was evaluated by at least three sensory assessors, who used blind tests to estimate wine quality and the median of their assessments was taken as a final value of the attribute [3]. Possible marks range from 0 – very bad to 10 – excellent. The distribution of the class attribute is presented in Figure 1. Individuals with values 0, 1, 2 and 10 are not present in the dataset. It can be observed that this is a unimodal distribution with mode of value 6. With the values of skewness=0.156 measured by third standardized moment and kurtosis=0.217 measured by the forth standardized moment distribution is slightly positively skewed and more peaked when compared to the normal distribution. Positive kurtosis means that the tails of the distributions are lighter – they contain few observations. In fact, class 3 and 9 together account for less than 1% of all observations which may make it very difficult for studied in subsequent sections algorithms to learn how to discriminate those classes. Disproportion between number of observations of average and extreme classes is depicted in Figure 2.

Normed Principal Component Analysis was performed on all the continuous variables and its results has been plotted in Figure 3. Quality class was projected as a categorical supplementary variable and also plotted on the same plan. It can be observed that variables are rather weakly correlated. The strongest correlated pairs are: *Density* and *Residual sugar* with coefficient 0.84; *Density* and *Alcohol* with coefficient -0.78. Those are the only pairs for which correlation

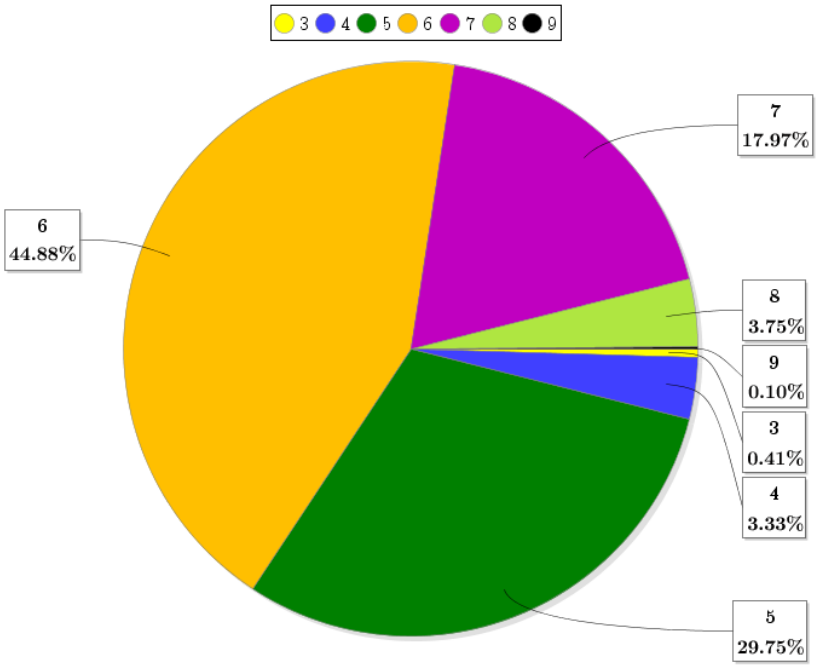


Figure 2: Distribution of Quality class in the dataset.

coefficient exceed 0.65 in absolute value. This suggest that attribute selection models could be applied to potentially exclude correlated variables. Since only two pairs are significantly correlated and applying robust attribute selection methods is beyond the scope of this work I decide to keep all variables for further analysis. By looking at the projection of the Quality class in the first factorial plane which explains 43.75% of the total inertia it is possible to observe that the first axis oppose good quality wines with poor and average quality ones. Better quality wines, which are placed on the left side, are rather associated with higher alcohol content. This observation accord with the experts knowledge in this field [3] and can be later verified with the behaviour of the algorithms.

# 3 Model selection

In this section choice of the best model for each of three algorithms will be made (parameter tuning). For choosing the best model parameters data will be split into training and testing set, both containing 50% of observations. Following simple measures will be computed on the test set: Mean absolute error and Root mean squared error. Those are the default measures given by the output of WEKA software that is used to conduct the experiment. Since they measure error the goal is to minimize them. It is important to emphasize

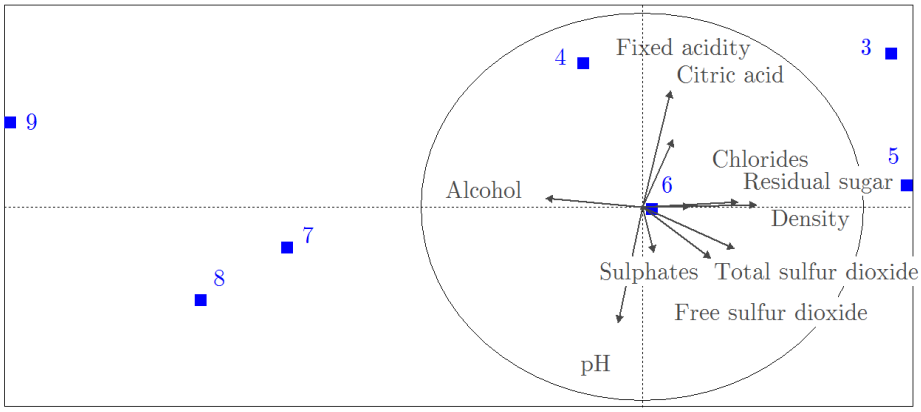


Figure 3: First factorial plan of Principal Component Analysis.

that unlike the Decision Tree which assign classes directly Neural Network and K-Nearest Neighbour algorithms predict class variable as a numerical continuous variable that should be further interpreted in order to be assigned to the final discreet class. This interpretation will be made later in Section 4 and now model selection will be based solely on mentioned measures. For instance, printout of the algorithm may look as follows:

| Instance ($i$) | actual class value ($a_i$) | predicted ($p_i$) | error ($er_i$) |
|---|---|---|---|
| 1 | 7 | 7.000 | 0.000 |
| 2 | 6 | 5.331 | 0.331 |

And for those values with $n$ number of instances metrics are computed as follows:

$$\text{Mean absolute error (mae)} \qquad \frac{\sum_{i=1}^{n} |er_i|}{n} \tag{1}$$

$$\text{Root mean squared error (rmse)} \qquad \sqrt{\frac{\sum_{i=1}^{n} er_i^2}{n}} \tag{2}$$

### 3.1 Neural network

Neural Networks are mathematical models that have been inspired by the natural structure of neurons observed in human brain. WEKA offers the most common type of Neural Network – Multilayer Perceptron. It consists of multiple layers of nodes in a directed graph, with each layer fully connected to the next one [4]. This is a classifier that uses backpropagation to classify instances. All nodes except for the output node utilize sigmoid function. Output nodes in

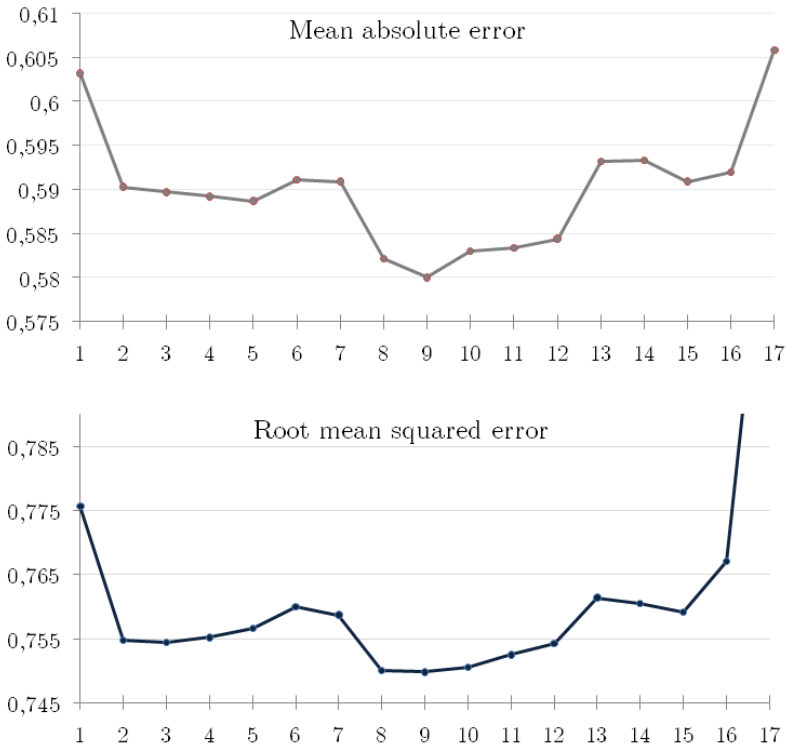


Figure 4: Error rates for different network topologies.

case of predicting numeric class become unthresholded linear units[5]. Various parameters that determine performance of this classifier have been experimented:

- Normalization of attributes and class – generally better performance was achieved with normalized attributes and class (so that they were from range -1 to 1) than non-normalized. Normalization does not influence result interpretation as the results are scaled back to their original range.
- Learning rate and momentum – are two parameters that influence learning process (i.e. optimization algorithm). Learning rate is the amount the weights are updated in each iteration. Momentum allows a change to the weights to persist for a number of adjustment cycles. Experiments with different values of those parameters did not give good results – mea around 0.7 and rmse around 0.9. Improvement was observed when decaying learning rate was used – this means that learning rate decreases while learning the model. Finally, values were set to learning rate=0.9 and momentum=0.1.

[5]WEKA documentation.

- Network topology – one of the most important factors in designing the Neural Network as too simple structure may not fully grasp underlying data dependencies and too complex structure may overfit the data [3]. Figure 4 presents plots of error rates (vertical axis) depending on the number of nodes in the single hidden layer of the network (horizontal axis) with all other parameters set as discussed above. The smallest error is obtained for 9 nodes with value of mae=0.58 and rmse=0.7498. Experiments with two hidden layers did not offer better results, hence the topology with one hidden layer of nine nodes will be used in subsequent sections.

## 3.2 K-nearest neighbours

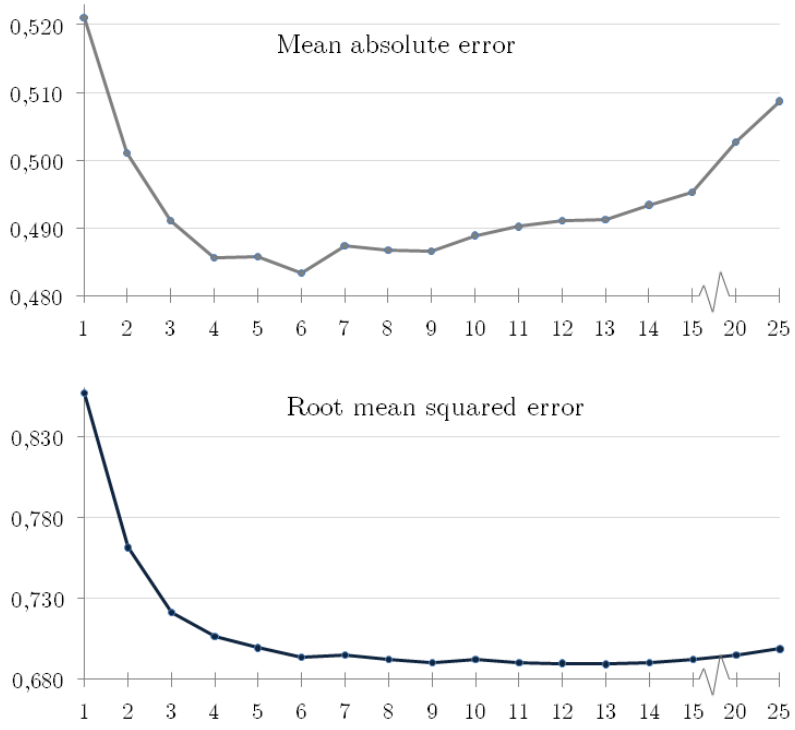


Figure 5: Error rates for different values of k parameter (k on horizontal axis).

K-nearest neighbours is an example of instance-based learning. That is a family of methods that use only specific instances in making predictions rather than building explicit abstractions like for example decision trees [1]. They classify new instance based on the information obtained from the most similar points following the assumption that *similar instances have similar classifications*. Important notion for this algorithm is a notion of similarity measure.

WEKA uses Euclidean distance as a default and normalize all attributes before making predictions[6]. Learning process for this model amounts only to storing all learning examples and all computation is performed when prediction for a new individual is calculated. Main parameters that determine performance of this classifier are:

- Distance weighting – significant improvement in performance was observed when 'votes' of the nearest neighbours are weighted with 1/distance weights. This means that closer neighbours have greater impact than further ones on deciding class for a new observation.
- K – main parameter that decide how many nearest neighbours participate in making decision on classification of the new point. In Figure 5 plots of error rates are presented depending on k. In this case choice of the best model is not that straightforward as in Neural Network case. The smallest errors are mae=0.4833 for k=6 and rmse=0.6892 for k=13. We can observe that rmse increases at slower rate than mae after reaching minimum. I decide to keep simpler model with k=6 and errors: mae=0.4833 and rmse=0.6937.

### 3.3 Decision tree

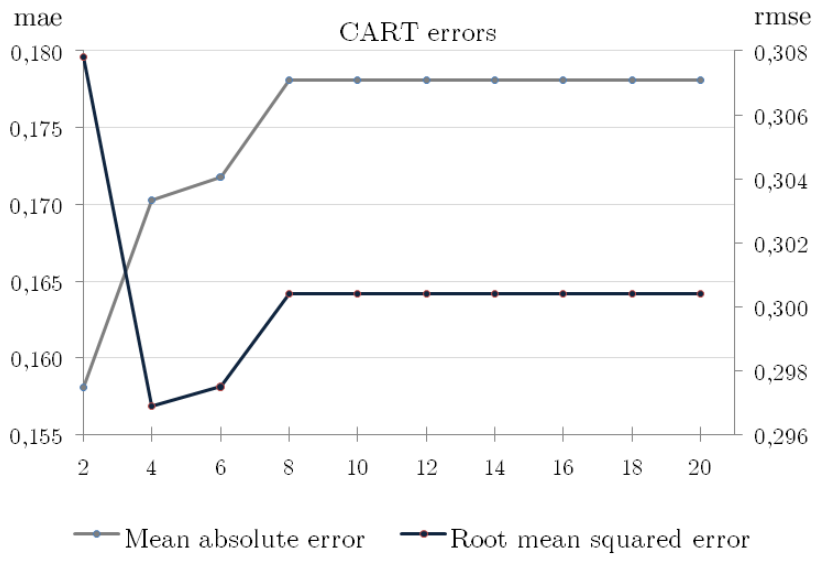


Figure 6: Error rates for CART tree.

CART algorithm will be used and its WEKA implementation under a name

[6]https://list.scms.waikato.ac.nz/pipermail/wekalist/2008-June/039956.html

of SimpleCart[7]. It was developed by four statisticians in the 80s and stands for Classification and Regression Tree [2]. The main idea of this approach is building overgrown tree using binary splits and then pruning it. While making choice gini index is considered as the criterion. The advantage of this algorithm is that it handles numeric variables well.

Main parameter for adjusting algorithm's performance in WEKA is the minimal number of instances at the terminal nodes ($m$). Errors have been plotted in Figure 6. They are not very sensitive to the changes of the value of this parameter (on the horizontal axis). In this case to decide between choosing value 2 or 4 of $m$ one additional measure is introduced. Tree complexity can be measured by counting numbers of splits. For $m = 2$ tree has 256 splits whereas for $m = 4$ only 49 splits are performed. Choice of value $m = 4$ seems to be a reasonable compromise because it still minimizes rmse and offers less complex tree.

# 4 Evaluation

To be able to compare results with those presented in [3] the same metrics are computed and very similar testing strategy is applied. Instead of running 20 times 5-fold cross validation as authors of [3] did, only one run is performed for each algorithm with parameters selected in Section 3. Thus obtained results will not be presented as the confidence intervals although still they will be comparable. Three computed metrics are:

- **MAD** – stands for Mean Absolute Deviation and is the same measure as Mean Absolute Error used in Section 3 for choosing parameters of the models.
- **Accuracy** – accuracy is the simple measure of percentage of correctly classified instances. Because two of chosen classifiers predict class as a numerical continuous variable at this point one must decide when the prediction is considered correct or incorrect. For this reason Tolerance ($T$) parameter is introduced. Accuracy will be measured for three values of tolerance: 0.25, 0.5 and 1. It means that an instance is considered as correctly classified if the absolute value of the difference between prediction and actual class is less or equal $T$. For example, instance with actual value 6 and prediction 5.70 is considered correctly classified for $T = 0.5$ but not for $T = 0.25$. Accuracy given by CART tree classifier cannot be expressed in terms of different $T$ levels because this method directly assign new individuals into discreet classes (it does not predict continuous variable). Its result would be compared best with other algorithms' accuracy at level of 0.5 because in this case most individuals can be assign exactly to one class (apart from rare cases of predictions such as 1.5, 2.5, etc. which can be correctly classified to two classes).

[7] https://list.scms.waikato.ac.nz/pipermail/wekalist/2009-September/045726.html

- **Kappa** – metric is computed for the $T = 0.5$. This statistic measures the accuracy compared with the random classifier for which it takes value of 0% [3]. The higher the value of this metric the more accurate the classifier. Kappa is computed from confusion matrix using following formula[8]:

$$\text{Kappa} = \frac{n \sum_{i=1}^{r} x_{ii} - \sum_{i=1}^{r} (x_{i\bullet} \cdot x_{\bullet i})}{n^2 - \sum_{i=1}^{r} (x_{i\bullet} \cdot x_{\bullet i})} \tag{3}$$

  where $n$ is the total number on instances; $r$ is the number of columns/rows of confusion matrix; $x_{ii}$ is the number of observations in column $i$ and row $i$; $x_{i\bullet}$ is the sum of the observations in the row $i$ and $x_{\bullet i}$ is the sum of the observations in column $i$.

| | NN Cortez | NN | Tree | KNN | SVM Cortez |
|---|---|---|---|---|---|
| MAD | 0.58±0.0 | 0.574 | **0.163** | 0.425 | 0.45±0.0 |
| $\text{Accuracy}_{T=0.25}$ (%) | 26.5±0.3 | 27.01 | — | **50.22** | **50.2±1.1** |
| $\text{Accuracy}_{T=0.5}$ (%) | 52.6±0.3 | 52.65 | 54.59 | **66.80** | 64.3±0.4 |
| $\text{Accuracy}_{T=1}$ (%) | 84.7±0.1 | 84.93 | — | **88.73** | 86.8±0.2 |
| $\text{Kappa}_{T=0.5}$ (%) | 23.5±0.6 | 22.91 | 28.68 | **49.03** | 43.4±0.4 |

Table 2: Results of classification.

All results are presented in the Table 2. Results from [3] can be found in columns *NN Cortez* and *SVM Cortez*. Best values of the metrics are in bold (MAD should be minimized and the rest should be maximized). All algorithms perform better than random classifier which is indicated by the positive values of Kappa. Among three algorithms applied in this work the on with the best performance is K-Nearest Neighbours (KNN). It has the best values of 4 out of 5 metrics used for the comparison. For MAD, Tree has the best value what can be explained by the fact that it is predicting class as a discreet categorical variable and not as a continuous numerical one. Comparing KNN with the best classifier of [3] – Support Vector Machine (SVM) one can observe that KNN slightly outperform SVM.

Almost identical performance is observed for the Neural Network tested in this paper and in [3] – values of $\text{Kappa}_{T=0.5}$ and $\text{Accuracy}_{T=0.5}$ for NN are inside confidence intervals of NN Cortez and the remaining three measures are a little better for NN model. This may be explained by the fact that different topologies of the network were used. In [3], exploited network was much simpler and had on average 2.1 hidden nodes whereas the one tested in this study has 9 nodes.

[8] http://www.nrcan.gc.ca/earth-sciences/glossary/index_e.php?id=2967

In the Table 3 confusion matrix for the KNN with Tolerance=0.5 is presented and precision of classification is calculated. Precision measures the percentage of correctly classified predictions within a certain class. Really high results are achieved for extreme classes with precision over 90% for class 8 and 89.29% for class 4 which means that this model can be successfully used for making predictions. As expected, no instances were classified into classes 3 or 9. This is because observations from these classes account for less than 1% of the dataset which is not sufficient for algorithms to learn how to discriminate them.

| | | **Predicted class** | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| **Actual class** | 3 | **0** | 1 | 10 | 8 | 1 | 0 | 0 |
| | 4 | 0 | **25** | 76 | 56 | 6 | 0 | 0 |
| | 5 | 0 | 1 | **974** | 451 | 31 | 0 | 0 |
| | 6 | 0 | 1 | 302 | **1657** | 236 | 2 | 0 |
| | 7 | 0 | 0 | 20 | 307 | **548** | 5 | 0 |
| | 8 | 0 | 0 | 0 | 47 | 60 | **68** | 0 |
| | 9 | 0 | 0 | 1 | 0 | 4 | 0 | **0** |
| $\text{Precision}_{T=0.5}$ | | — | 89,29% | 70,43% | 65,60% | 61,85% | 90,67% | — |

Table 3: Confusion matrix for KNN and Tolerance=0.5

Decision Tree built using CART algorithm was ranked at second place by accuracy measure and Kappa statistic with scores respectively 54.59% and 29.68%. It is interesting to analyse the structure of the tree as it may give some insights into variables significance. The first split is made depending on *alcohol level* and this attribute is used seven times to split the tree what seems to confirm that this variable is important and that the hypothesis from Section 2 that better quality wines are often associated with higher content of alcohol may be valid. Contrarily, tree uses *sulphates* variable only once, *citric acid level* twice, *density* three time and does not use *residual sugar level* at all to decide on splits what can imply that those variables are less significant in discriminating between different quality classes. This is a interesting conclusion because in [3] different interpretation of the variables significance was obtained for SVM model. Similarly, *alcohol* was considered the most influential and *density* and *citric acid level* less important. However, *sulphates* were considered second important variable and *residual sugar level* was moderately important.

# 5 Conclusion

In this study three classification algorithms were tested on the wine dataset. Firstly, best parameters for each algorithm were selected using two metrics: Mean Absolute Error and Root Mean Squared Error calculated on the test set obtained by 50% split of data. Then, for the selected models one run of 5-fold cross validation was applied and measures such as accuracy and Kappa were computed so that comparison between different algorithms was possible. Neural Networks successfully recreated results obtained by Neural Network in [3]. The best classifier was K-Nearest Neighbours with value of K=6 and it slightly outperformed SVM presented in [3] with overall accuracy of 66.80% (Tolerance=0.5) and 88.73% (Tolerance=1). Decision tree built using CART algorithm was ranked at second place. Obtained results are promising and tested models could be exploited further in the wine industry for various pourposes such as improving and automating certification process or supporting marketing decisions which becomes more and more important as the market is growing.

# References


[1] David W. Aha, Dennis Kibler, and Marc K. Albert. Instance-based learning algorithms. *Machine Learning*, 6:37–66, 1991. 10.1023/A:1022689900470.

[2] Leo Breiman, Jerome H. Friedman, Richard A. Olshen, and Charles J. Stone. *Classification and Regression Trees*. Wadsworth International Group, Belmont, California, 1984.

[3] Paulo Cortez, Juliana Teixeira, António Cerdeira, Fernando Almeida, Telmo Matos, and José Reis. Using data mining for wine quality assessment. In *Proceedings of the 12th International Conference on Discovery Science*, DS '09, pages 66–79, Berlin, Heidelberg, 2009. Springer-Verlag.

[4] Nigel Williams, Sebastian Z, and Grenville Armitage. Evaluating machine learning algorithms for automated network application identification. Technical report, 2006.