Are oenologists outdated?

A machine learning approach to predict human wine taste

Università degli Studi di Milano-Bicocca, January 2020
Machine Learning Project, Team 31

Emanuele Artioli[1)], **Maddalena Baldo**[2)], **Giacomo De Gobbi**[3)], **Daniele Monterisi**[4)], **Davide Vercesi**[5)]

[1)] 00000 CdLM Data Science, Università degli Studi di Milano-Bicocca
[2)] 00000 CdLM Data Science, Università degli Studi di Milano-Bicocca
[3)] 00000 CdLM Data Science, Università degli Studi di Milano-Bicocca
[4)] 00000 CdLM Data Science, Università degli Studi di Milano-Bicocca
[5)] 00000 CdLM Data Science, Università degli Studi di Milano-Bicocca

Abstract

Wine classification is a difficult task since taste is the least understood of the human senses [2]. The aim of this work is to establish whether it is possible to implement an automatic system of wine quality classification, in order to support decision-making processes; we propose a data mining approach to predict human wine taste preferences based on physicochemical properties from wine analyses. The question to be answered is: is it possible to predict wine quality effectively and automatically?

Contents

List of Figures

List of Tables

1 Introduction and scope

Once viewed as a luxury good, wine is nowadays increasingly enjoyed by a wider range of consumers. To support its growth, the wine industry is investing in new technologies for both wine making and selling processes[1].
In this study a predictive modeling of wine quality and performance evaluation of different classification models is presented in order to use a model that classifies as correctly as possible the quality of Portoguese "Vinho Verde" red wine, this analysis based on analytical data that are easily available at the wine certification step. Wine certification is generally assessed by physicochemical and sensory tests [4] including determination of density, alcohol or pH values, while sensory tests rely mainly on human experts. This data is available and contains valuable information such as trends and patterns, which can be used to improve decision making and optimize chances of success [6]. Wine certification and quality assessment are key elements within this context. Certification prevents the illegal adulteration of wines (to safeguard human health) and assures quality for the wine market. Quality evaluation is often part of the certification process and can be used to improve wine making (by identifying the most influential factors) and to stratify wines such as premium brands (useful for setting prices) [1]. Such models are useful to support the oenologist wine tasting evaluations and improve wine production. Furthermore, similar techniques can help in target marketing by modeling consumer tastes from niche markets.

2 Data exploration

Data obtained from a dataset containing information about the quality of Portuguese red wine "Vinho Verde" a unique product from the Minho (northwest) region of Portugal. Medium in alcohol, is it particularly appreciated due to its freshness (specially in the summer). This wine accounts for 15% of the total Portuguese production [7].
The dataset is available on the Kaggle Platform [5] and it is made of 1599 records each with the following 12 features:

Input variables, based on physicochemical tests:

1. **Fixed acidity** (g/dm^3): most acids involved with wine or fixed or nonvolatile (do not evaporate readily) includes acids that do not vaporize when boiling and are present in musts and wines. Its determination is aimed at assessing the stability and preservability of wines;
2. **Volatile acidity** (g/dm^3): the amount of acetic acid in wine, which if too high can lead to an unpleasant, vinegar taste;
3. **Citric acid** (g/dm^3): found in small quantities, citric acid can add 'freshness' and flavor to wines;
4. **Residual sugar** (g/dm^3): the amount of sugar remaining after fermentation, it's rare to find wines with less than 1 *gram/liter* and wines with greater than 45 *grams/liter* are considered sweet;
5. **Chlorides** (g/dm^3): the amount of salt in the wine;
6. **Free sulfur dioxide** (mg/dm^3): the free form of SO_2 exists in equilibrium between molecular SO_2 (as a dissolved gas) and bisulfite ion, it prevents microbial growth and the oxidation of wine;
7. **Total sulfur dioxide** (mg/dm^3): amount of free and bound forms of SO_2, in low concentrations SO_2 is mostly undetectable in wine, but at free SO_2 concentrations over 50 ppm, SO_2 becomes evident in the smell and taste of wine;
8. **Density** (g/dm^3): is the ratio between the density of the wine (or must) and the density of water at the same temperature;
9. **pH**: describes how acidic or basic a wine is on a scale from 0 (very acidic) to 14 (very basic), most wines are between 3-4 on the pH scale;
10. **Sulphates** (g/dm^3): a wine additive which can contribute to sulfur dioxide gas (SO_2) levels, wich acts as an antimicrobial and antioxidant;
11. **Alcohol** (% vol.): the percentage of alcohol content of the wine.

Output variable, based on sensory data:

12. **Quality**: output variable based on sensory data[1], ranges from 0 to 10.

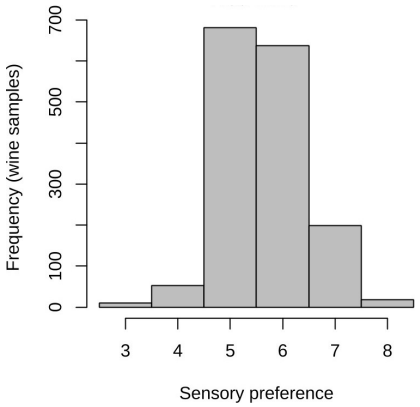

Figure 1: Histogram of the target variable: *quality*

[1]Median of at least three evaluations made by wine experts, each expert graded the wine quality between 0 (very bad) and 10 (excellent).

Attribute	Min	Max	Mean
fixed acidity (g/dm^3)	4.6	15.9	8.3
volatile acidity (g/dm^3)	0.1	1.6	0.5
citric acid (g/dm^3)	0.0	1.0	0.3
residual sugar (g/dm^3)	0.9	15.5	2.5
chlorides (g/dm^3)	0.01	0.61	0.08
free s. dioxide (mg/dm^3)	1	72	14
total s. dioxide (mg/dm^3)	6	289	46
density (g/dm^3)	0.990	1.004	0.996
pH	2.7	4.0	3.3
sulphates (g/dm^3)	0.3	2.0	0.7
alcohol (% vol.)	8.4	14.9	10.4

Table 1: The physicochemical data (input variables), and its corresponding statistics.

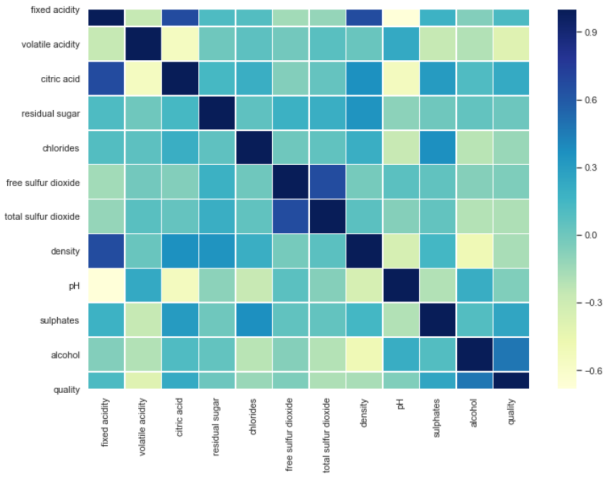

Figure 2: Dataset correlation matrix.

The treatment of attributes that show correlation is explained in the Section 6.1.2.

3 Preprocessing

3.1 Binarization of Target Variable

Quality domain (as presented) ranges between 0 and 10. Its distribution is comprised of a majority of central values, with narrow tails. Furthermore, as shown in Figure 1, the extreme values (lower than 3 and higher than 8) are not represented, therefore the dataset domain actually ranges between 3 and 8.
To lower the dispersion of the attribute, *Quality* was binarized in a new column ***Category***, setting its value as "**good**" if the value of quality was higher than or equal to 7, and "**not good**" if value was lower than 7; as suggested in the description of the dataset by the user who made data available.
This technique has been implemented in Knime through the RuleEngine node which creates a division into two classes according to the rule:

Category	Records
good (≥ 7)	217
not good (≤ 6)	1382

Table 2: Counting values by class (*good, not good*).

The aim is to be able to predict the quality of the wine according to these two classes.

3.2 Outliers Management

After exploring the domain of each attribute it has been noticed that there are outliers, i.e. values higher than the upper whisker (Q3 + 1.5 * IQR) or lower than the lower whisker (Q1 - 1.5 * IQR).

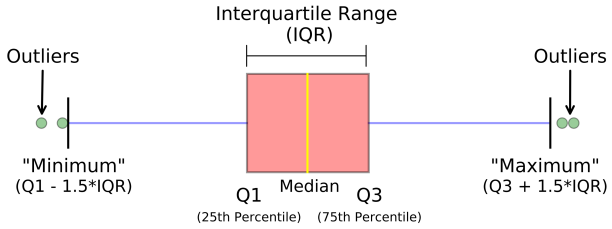

Figure 3: Outliers.

In order to avoid eliminating them and risking a significant loss of information, given the already small size of the dataset, the out-of-range values were replaced with the corresponding upper ("maximum") or lower whisker ("minimum").

4 Classification

4.1 Partitioning

The preprocessed dataset was partitioned in a training and a test sets using the stratified sampling method on the attribute *Category*. The training set consisted of 75% of the original dataset, while the test set consisted of the remaining 25%.

Original Dataset	Training Set	Test Set
(100%) 1599	(75%) 1199	(25%) 400

Table 3: Partitioning of the dataset.

4.2 Class Imbalance Problem

The training set presents the class inbalance problem: the "good" class is underrepresented, the model will suffer in training and will not be accurate.
In this context, many classification learning algorithms have low predictive accuracy for the infrequent class[8].

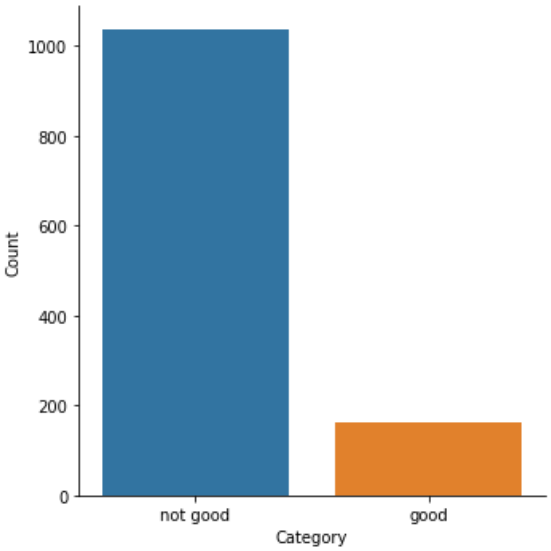

Figure 4: Class imbalance problem on traning set.

4.2.1 SMOTE Technique

The approach we used is to reduce the class imbalance oversampling in the training set using the SMOTE (Synthetic Minority Over-sampling) technique [9], implemented in Knime with the SMOTE node. We decided to choose the SMOTE instead of a normal oversampling (taking the same record multiple times) because it is less prone to the overfitting phenomenon. After splitting the original dataset (see Section 4.1), perform oversampling on the training set only and test on the original data test set. The algorithm, starting from the real instances already present in the dataset and represented in minority, creates new records obtained from the observation of the k nearest neighbours of the considered instance. The parameter k has been set to 5 because we have noticed clear improvements in the classification results of the instances belonging to the high class.

4.3 Classification Models

In this work, several classification techniques have been applied, inherent to Machine Learning; the aim is to identify the most suitable technique to try to predict the category of wine by using the available data.

- **Random Forest**: Random forests are a combination of tree predictors such that each tree depends on the values of a random vector sampled independently with the same distribution for all trees in the forest. So, this method is to grow an ensemble of trees and let them vote for the most popular class. Firstly, in order to grow these ensemble trees, features are selected randomly that govern the growth of each tree in the ensemble. Secondly, at each node, the split is selected at random from among the best splits. Last, random training set, bagging is used in tandem with random feature selection [10].

- **J48**: J48 is an open source Java implementation of the C4.5 algorithm in the Weka data mining tool. C4.5 is a program that creates a decision tree based on a set of labeled input data. The decision trees generated by C4.5 can be used for classification, and for this reason [11].

- **Decision Tree**: is defined as a classification procedure that recursively partitions a data set into smaller subdivisions on the basis of a set of tests defined at each branch (or node) in the tree. The tree is composed of a root node (formed from all of the data), a set of internal nodes (splits), and a set of terminal nodes (leaves). Each node in a decision tree has only one parent node and two or more descendant nodes [12].

- **Naive Bayes**: the naive Bayes classifier greatly simplify learning by assuming that features are independent given class. Although independence is generally a poor assumption, in practice naive Bayes often competes well with more sophisticated classifiers [13].

- **Logistic Regression** is one of the regression analysis approaches which are used to predict an outcome when the dependent variable is categorical (binary variable) [14].

- **Multilayer Perceptron**: consists of a system of simple interconnected neurons, or node which is a model representing a non linear mapping between an input vector and an output vector. The nodes are connected by weights and output signals which are a function of the sum of the inputs to the node modified by a simple non linear transfer [15].

- **SMO**: Sequential Minimal Optimization (SMO) is an algorithm to efficiently solve the optimization problem that emerges during the training of a support vector machine. This implementation replaces missing values, transforms nominal attributes into binaries and finally normalizes all explanatory variables [16].

5 Performance Evaluation

Chosen the model, it is important to validate it. To do this we use some validation criteria that rely on the confusion matrix:

	Predicted Negative	**Predicted Positive**
Actual Negative	TN	FP
Actual Positive	FN	TP

Figure 5: Confusion matrix.

Given TN (TP) as true negative (true positive) or the portion of negative (positive) class correctly predicted and FN (FP) as false negative (false positive) or the portion of negative (positive) class erroneously predicted. Through these quantities it is possible to calculate indices that allow to evaluate the performance of the model.

- **Precision**: Percentage of observations that are actually positive in the group of the predicted positive class.
$$\frac{TP}{TP + FP}$$
- **Recall**: Percentage of positive observations that are correctly predicted by the model, which is summarized in the ability of the classifier to correctly assign the real class to each instance.
$$\frac{TP}{TP + FN}$$
- **F-measure (F1)**: Harmonic mean between Recall and Precision. A high value indicates that both Recall and Precision take high values.
$$\frac{2 * R * P}{R + P}$$
- **Accuracy**: Percentage of positive and negative observations correctly predicted.
$$\frac{TP + TN}{TP + FP + TN + FN}$$

6 Validation

Each classification model has been validated using Holdout and k-Fold-Cross-Validation techniques.

6.1 Holdout

The first approach used is the Holdout method which is based on partitioning the dataset into two separate subsets following a stratified sampling procedure (75% for train and 25% for test).
Below are presented the Precision, Recall and F-measure measurements for both *good* and *not good* classes, and a table showing the Accuracy values.

Method		**Prec.**	**Recall**	**F1**
Random Forest	good	0.451	0.685	0.544
	not good	0.947	0.870	0.907
J48	good	0.539	0.667	0.529
	not good	0.943	0.867	0.904
Decision Tree	good	0.400	0.556	0.465
	not good	0.926	0.870	0.897
Naive Bayes	good	0.328	0.741	0.455
	not good	0.950	0.763	0.846
Logistic Regression	good	0.357	0.759	0.485
	not good	0.954	0.786	0.862
Multilayer Perceptron	good	0.417	0.741	0.553
	not good	0.954	0.838	0.892
SMO	good	0.536	0.685	0.602
	not good	0.949	0.908	0.928

Table 4: Holdout: Precision, Recall and F-measure values.

The results that appear in Tables 4 and 5 show that SMO is the best classifier with the highest values of Precision, Recall, F-mesaure and Accuracy. We note that the classifiers in general are very good at predicting the class of not good wines compared to good ones.
The advantage of Holdout method is that it is usually preferable to the residual method and takes no longer to compute. However, its evaluation can have a high variance. The evaluation may depend heavily on which data points end up in the training set and which end up in the test set, and thus the evaluation may be significantly different depending on how the division is made.

Method	Accuracy
Random Forest	0.86
J48	0.82
Decision Tree	0.83
Naive Bayes	0.76
Logistic Regression	0.79
Multilayer Perceptron	0.82
SMO	0.89

Table 5: Holdout: accuracy values.

6.1.1 Overfitting Check

After identifying the classification model that achieved the best performance (SMO), it was decided to check for overfitting. The phenomenon of overfitting occurs when a classification model adapts too well to the train set data (compared to the test set data) memorizing various peculiarities of the training data rather than finding a general predictive rule [17]. In addition, the previous SMOTE technique used may lead to an overfitting condition.

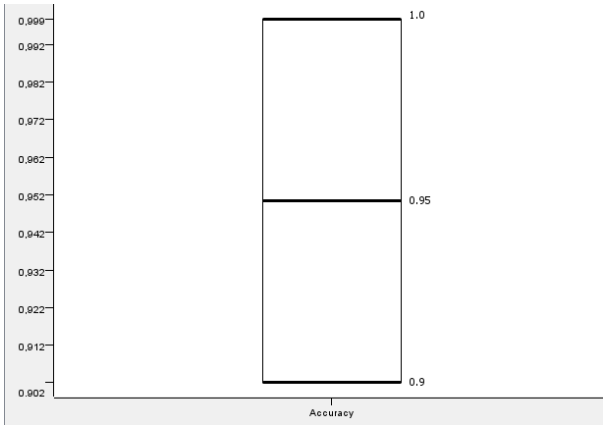

Figure 6: Holdout overfitting.

The plot shows the range of accuracy values obtained on the train set (1) and the test set (0.9). Although the overfitting problem is present, the difference between the accuracy obtained on the train and the test set is less than 0.1. The model adapts too well to the train set data but also performs very well on the test set. Overfitting is present, but it's acceptable.
The overfitting phenomenon can be determined by datasets that have too many parameters compared to the attributes that are really relevant.

As seen before, the correlation matrix (Figure 2.) shows a correlation between some variables: *fixed acidity* positively correlated with *density* (0.7) and *citric acid* (0.7), and negatively correlated with *pH* (-0.7); *free sulfur dioxide* positively correlated with *total sulfur dioxide* (0.7); and *volatil acidity* negatively correlated with *citric acid* (-0.6). Observing this correlation we tried to implement a reduction in dimensionality with the Feature Selection operation through the node AttributeSelectedClassifier, which selected these attributes: *volatile acidity, cictic acid, free sulfur dioxide, density, sulphates, alchol, category.* The classifiers were trained with the filtered dataset, but with lower performance than the classifiers trained on the original dataset. This may be due to the fact that no particular attribute contributes to the quality of the wine.
It was decided not to operate any Feature Selection procedure.

6.2 K-Fold Cross Validation

K-fold Cross Validation is one way to improve over the Holdout method and a powerful preventative measure against overfitting. The data set is divided into k subsets, and the holdout method is repeated k times. Each time, one of the k subsets is used as the test set and the other *k-1* subsets are put together to form a training set. Then the average error across all k trials is computed.
The advantage of this method is that it matters less how the data gets divided and achieves good performance on a reduced dataset; the variance of the resulting estimate is reduced as k is increased, and in our case k is set to 10.
Below are presented the Precision, Recall and F-measure measurements for both *good* and *not good* classes, and a table showing the Accuracy values.

Method	Accuracy
Random Forest	0.90
J48	0.89
Decision Tree	0.89
Naive Bayes	0.85
Logistic Regression	0.87
Multilayer Perceptron	0.89
SMO	0.92

Table 6: k-Fold-Cross-Validation: accuracy values.

Method		**Prec.**	**Recall**	**F1**
Random Forest	good	0.650	0.618	0.634
	not good	0.940	0.948	0.944
J48	good	0.594	0.479	0.531
	not good	0.921	0.949	0.934
Decision Tree	good	0.594	0.599	0.596
	not good	0.937	0.936	0.936
Naive Bayes	good	0.471	0.668	0.552
	not good	0.944	0.882	0.912
Logistic Regression	good	0.556	0.341	0.423
	not good	0.902	0.957	0.929
Multilayer Perceptron	good	0.619	0.516	0.563
	not good	0.926	0.950	0.938
SMO	good	0.946	0.401	0.563
	not good	0.914	0.996	0.953

Table 7: k-Fold-Cross-Validation: Precision, Recall and F-measure values.

The results that appear in Tables 6 and 7 show that SMO is still the best classifier with the highest values of Precision, Recall, F-mesaure and Accuracy. We note a significant improvement compared to the Holdout validation method, Accuracy increases in all classification models by a good percentage.

7 Conclusions

In recent years, the interest in wine has increased, leading to growth of the wine industry. As a consequence, companies are investing in new technologies to improve wine production and selling [1]. This work aims at the prediction of wine preferences from physicochemical properties tests that are available at the wine quality certification step. A dataset is accessible which contains red wine samples from the portoguese wine "Vihno Verde". Our study showed that it is possible to predict the quality of the red wine "Vihno Verde" (*good, not good*) with a 92% Accuracy given by the SMO classification model which proved to be the best model, both with Holdout and k-Fold Cross Validation. For our data set, SMO adapts well to the data and can predict wine quality just as well, another very reliable model for prediction is Random Forest with 90% of Accuracy. With k-Fold Cross Validation we have a significant increase in all performance measures. No feature selection was necessary, but instead outliers have to be treated and the class imbalance problem has to be solved with the SMOTE technique. The question we were looking for answers shown a positive outcome and we hope that in the years to come this machine learning methodology will become a way to increase and enrich the wine industry. The algorithm could be useful, for example, to a wine producer or a wine seller who wants to rely on the model to see how "good" his wine is. The quality of the wine will be assessed by the producer/seller faster and without requiring the help on an expert. It is important to correctly classify in such a way that the reputation, the price to be applied and therefore the profit are rightly influenced.

References

[1] 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.

[2] D. Smith, R. Margolskee, Making sense of taste, Scientific American, Special issue 16 (3) (2006) 84–92.

[3] I. Walsh, G. Pollastri, and S. C. E. Tosatto, "Correct machine learning on protein sequences: A peer-reviewing perspective," Brief. Bioinform., vol. 17, no. 5, pp. 831–840, 2016, doi: 10.1093/bib/bbv082.

[4] S. Ebeler, Flavor Chemistry — Thirty Years of Progress, Kluwer Academic Publishers, 1999, pp. 409–422, chapter Linking flavour chemistry to sensory analysis of wine.

[5] Kaggle (January 2020). Red Wine Quality Dataset. Retrieved from: https://www.kaggle.com/uciml/red-wine-quality-cortez-et-al-2009winequality-red.csv

[6] E. Turban, R. Sharda, J. Aronson, D. King, Business Intelligence, A Managerial Approach, Prentice-Hall, 2007.

[7] CVRVV. Portuguese Wine — Vinho Verde. Comissão de Viticultura da Região dos Vinhos Verdes (CVRVV), http://www.vinhoverde.pt, July 2008.

[8] Ling C.X., Sheng V.S. (2011) Class Imbalance Problem. In: Sammut C., Webb G.I. (eds) Encyclopedia of Machine Learning. Springer, Boston, MA.

[9] Blagus, R., Lusa, L. SMOTE for high-dimensional class-imbalanced data. BMC Bioinformatics 14, 106 (2013).

[10] H. Zhang, F. Hamprecht, and A. Amann, "Report about VOCs Dataset's Analysis based on randomForests Method," Proc. - Eighth Int. Conf. High-Performance Comput. Asia-Pacific Reg. HPC Asia 2005, vol. 2005, pp. 603–607, 2005, doi: 10.1109/HPCASIA.2005.85.

[11] J. Gholap and J. Gholap, "Performance Tuning of J48 Algorithm for Prediction of Soil Fertility," vol. 2, no. 8, 2013.

[12] C. E. Brodley and M. A. Friedl, "Decision tree classification of land cover from remotely sensed data," Remote Sens. Environ., vol. 61, no. 3, pp. 399–409, 1997, doi: 10.1016/S0034-4257(97)00049-7.

[13] E. P. F. Lee, J. Lozeille, P. Soldán, S. E. Daire, J. M. Dyke, and T. G. Wright, "An empirical study of the naive Bayes classifie," Phys. Chem. Chem. Phys., vol. 3, no. 22, pp. 4863–4869, 2001, doi: 10.1039/b104835j.

[14] A. El-Koka, K. H. Cha, and D. K. Kang, "Regularization parameter tuning optimization approach in logistic regression," Int. Conf. Adv. Commun. Technol. ICACT, pp. 13–18, 2013.

[15] M. W. Gardner and S. R. Dorling, "Artificial neural networks (the multilayer perceptron) - a review of applications in the atmospheric sciences," Atmos. Environ., vol. 32, no. 14–15, pp. 2627–2636, 1998, doi: 10.1016/S1352-2310(97)00447-0.

[16] Y. Z. Liu, H. X. Yao, W. Gao, and D. Bin Zhao, "Single sequential minimal optimization: An improved SVMs training algorithm," 2005 Int. Conf. Mach. Learn. Cybern. ICMLC 2005, no. August, pp. 4360–4364, 2005, doi: 10.1109/icmlc.2005.1527705.

[17] T. Dietterich, "Overfitting and Undercomputing in Machine Learning," pp. 2–3.