A Comparison of Naïve Bayes and Random Forest Applied to the Red Wine Dataset

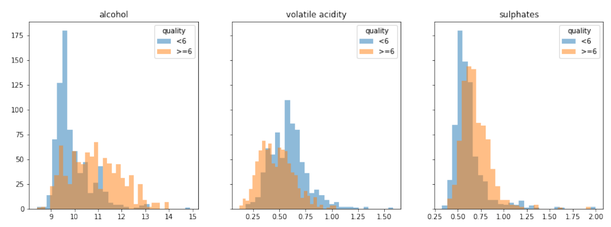
Chikaze Mori

Brief description and motivation of the problem

- We will apply Naïve Bayes and Random Forests to the red wine dataset, and compare the performance in a multi class classification problem which predicts qualities of red wine based on physicochemical tests.
- We will also contrast our results to the results obtained by a previous study by S. Kumar, K. Agrawal and N. Mandan (2020) [1].
- As wine is increasingly enjoyed by a wider range of consumers, wine quality assessment is a key element within this context since it prevents the illegal adulteration of wines and assures quality for the wine market [2].

Initial analysis of the data set including basic statistics

- Dataset: Wine Quality Datasets from UCI [2]
- The dataset is separated into train and test sets with the size of 0.7 and 0.3 respectively.
- The original dataset has 12 attributes - 11 numeric (ratio), 1 categorical (ordinal).
- To understand the dataset better, the basic statistics are grouped by its wine quality (<6 or >=6).
- The basic statistics for each column are calculated and showed by the table on the right.
- Normalised histograms show noticeable differences in some variables between wine qualities, Such as alcohol, volatile acidity, and sulphates. (e.g. alcohol is right-skewed in better quality of red wine but is more normally distributed in worse quality of red wine)



	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcohol	quality
mean(<6)	8.14	0.59	0.24	2.54	0.09	16.57	54.65	1.00	3.31	0.62	9.93	4.90
mean(>=6)	8.47	0.47	0.30	2.54	0.08	15.27	39.35	1.00	3.31	0.69	10.86	6.27
std(<6)	1.57	0.18	0.18	1.39	0.06	10.89	36.72	0.00	0.15	0.18	0.76	0.34
std(>=6)	1.86	0.16	0.20	1.42	0.04	10.04	27.25	0.00	0.15	0.16	1.11	0.49
skew(<6)	1.32	0.81	0.61	4.34	5.48	1.21	0.80	0.46	0.06	3.08	1.68	-3.68
skew(>=6)	0.74	0.54	0.08	4.71	4.98	1.27	2.64	0.08	0.31	2.18	0.37	1.53

Brief summary of the two ML models with their pros and cons

Naïve Bayes (NB)

- A conditional probability model based on the Bayes theorem
- The simplest form of Bayesian network, in which all attributes are independent given the value of the class variable [3].
- Predicts membership probabilities for every class, and the class that has maximum probability will be chosen (Maximum A Posteriori).
- When the dataset is continuous, it assumes that probabilities of class membership for each attributes distributed according to a normal distribution (Gaussian Naïve Bayes).

Pros

- It is simple, easy to understand, and works fast.
- Its competitive performance in classification is surprisingly good [3].
- Could perform better than other models when the attributes are independent.
- Only requires small number of data.

Cons

- When the training samples are not adequate, probability estimation method will inevitably suffer from the zero-frequency problem [4]. Laplace-estimate and M-estimate are the main smoothing techniques used to avoid this problem.
- It assumes that attributes are independent although it is almost impossible to get a completely independent attributes in real life. However, NB performs quite well in practice even when strong attribute dependencies are present [5].
- The probability estimation is poor even when the classification is still correct [3].

Random Forests (RF)

- A combination of decision trees
- Each tree depends on the values of a random vector sampled independently and with the same distribution for all trees in the forest [6].
- Once generated, each tree casts a unit vote for the most popular class based on the mean or the mode of the classes.
- First introduced by Tin Kam Ho [7], and later developed by Leo Breiman [6].

Pros

- Outperforms Decision Trees by reducing from individual trees without increasing bias.
- Less likely overfits [6].
- Outperforms most other classification techniques on most problems [8].
- Feature importances can be inspected after modelling.
- Can be implemented with parallel computing.

Cons

- It takes time to train on a big dataset.
- It outperforms Decision trees but its accuracy is lower than Boosted trees [6].
- Not suitable for regression problems.
- Does not perform well on smaller datasets [6].

Hypothesis statement

- We expect RF to perform better than NB since RF is considered as one of the methods that gives the best performance and NB is not competitive with those [8].
- Also previous empirical studies [1] [9] show that RF outperforms NB on the red wine dataset.
- On their study, RF gives an error of 34.2% and NB gives an error of 44.1%.
- Since the default parameters are not optimal in random forest and naive Bayes [9], we need to tune hyperparameters.

Description of the choice of training and evaluation methodology

- Split the dataset into a train set (70%) and test set (30%) and train on the train set that has 1120 data points to allow comparability with our reference paper.
- For each model, we tune its hyperparameters to find optimum values using a grid search.
- Also tune hyperparameters for a NB model that its train set is applied PCA as it might improve performance of NB [10].
- Use 10-fold cross validation to prevent overfitting and estimate the classification error.
- The error measure used was classification error.

Choice of parameters and experimental results

Naïve Bayes (NB)

Parameters

Ran a Grid Search on the two hyperparameters, type of data distributions used for modelling, kernel or normal (DistributionNames), and kernel smoothing window width when the data distribution type is kernel (Width). Also conducted this process of hyperparameter tuning on the dataset that the number of its variables is reduced by PCA to see if PCA improves performance of NB as Fan claimed [10].

Experimental Results

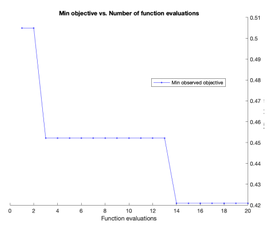
- The best model used 5 attributes that were produced by PCA, kernel density function, and the optimised width of 1.2805e-04.
- The best model among models that did not use PCA also used kernel density function.
- The classification errors on the 10 fold cross validation were compatible between models with PCA and without PCA, while the model with PCA performed better on the test set.
- Hyperparameter tuning for the best model evaluated 20 objectives and took 7.9 seconds, and all other models were tuned similarly.
- The confusion matrix showed that the model predicted the classes of 5, 6, and 7 well, while it predicted the classes of 4 poorly and did not predict the classes of 3 and 8 at all.
- Regardless of PCA, all the models performed significantly worse than any Random Forest models.

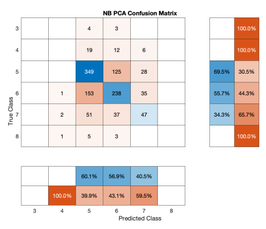
Naïve Bayes classification error

Train	43.4%
Validation	43.9%
Test	46.1%

NB (PCA) classification error

Train	42.1%
Validation	43.3%
Test	43.8%





Random Forests (RF)

Parameters

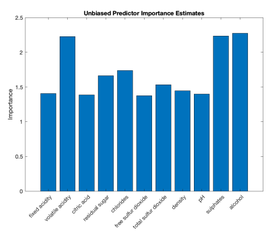
Ran a Grid Search on the two hyperparameters, number of trees in the ensemble (NumLearningCycle), number of attributes to sample at each node (NumVariablesToSample) since they are the two important parameters of Random Forests [11].

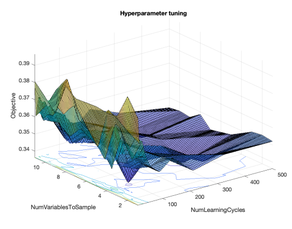
Experimental Results

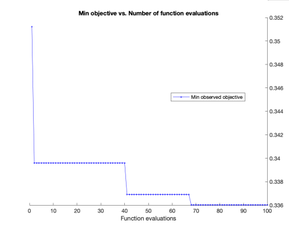
- The best model in the grid search used 324 trees, and 5 variables.
- Hyperparameter tuning for the best model evaluated 100 objectives and took 1391.2 seconds, and all other models were tuned similarly.
- The confusion matrix showed that the model predicted the classes of 5, 6, and 7 well, while it predicted the classes of 3 and 4 poorly and did not predict the class of 8 at all.
- Unbiased predictor importance estimates showed that alcohol was the most important attribute followed by sulphates and volatile acidity, which is consistent with our initial analysis.
- The time taken to built a model increased as it used more trees and variables.
- Its performance significantly improved at first as the number of trees increased from 0.
- It did not perform well when the number of variables is too small.

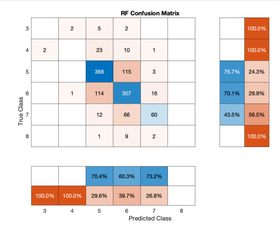
Random Forest classification error

Train	33.6%
Validation	34.3%
Test	36.8%









Analysis and critical evaluation of results

- Our experiments consistently showed that RF outperformed NB on the dataset as any of the RF models outperformed any of the NB models in any phase, which supports our hypothesis motivated by Caruana et al [8].
- The best NB model showed high bias and low variance and they are indicated by the small difference of errors between the train and test sets, while the best RF model showed a relatively big difference of errors between the train and test sets and seemed overfitting. In fact, the classification error of our best RF model on the train set is lower compared to the previous empirical study by Kumar et al [1], and the error on the test set is higher.
- The better performance of RF indicates RF's ability of reducing bias, which is consistent with the study by Breiman [6].
- The best NB model performed better than the previous study [1]. It showed that data preprocessing can improve the performance of NB, and it is consistent with the study by Fan at el [10]. This is because NB assumes that attributes are independent and PCA reduced dependencies between attributes.
- Both models completely failed to predict minor classes, 3, 4, and 8. Since this seemed to be caused by the imbalance of the dataset, it was reasonable to apply SMOTE to the dataset. However, balancing the dataset by applying SMOTE was not effective in this case. It even led worse performances for both models (see the Intermediate Results section of the supplementary material). Considering the extreme imbalance of the dataset, it is possible that simply removing the minor classes could have been a fair way for comparison and even improve the models.
- Hyperparameter tuning took much longer for RF than NB, 1391.2 seconds, 23.2 minutes and 7.9 seconds respectively. It indicates that RF is capable of being trained more computationally, while NB is simpler.
- The best NB model did not predict the classes of 3 and 8 at all, which is likely considered as a zero-frequency problem. Hence it was preferable if smoothing techniques such as Laplace-estimate and M-estimate would have been applied to the NB models as suggested by Wu at el [4].

Lessons learned and future work

- NB is a model that could be built much faster than RF, which indicates that NB does not require much of hyperparameter tuning, while RF does. However, NB only achieves limited performances compared to RF, and data preprocessing techniques such as PCA are applicable for NB to improve its performances.
- NB is expected to perform better if PCA is applied to its dataset and the number of attributes and classes is bigger [10]. Also it is shown that NB performs well when PCA is applied to its dataset. Hence, it might improve the performance of NB more if PCA and SMOTE are both applied so the attributes are independent and SMOTE allows all classes to be well predicted.
- According to Fan et al [10], CC-ICA (Class Conditional Independent Component Analysis) improves performances of NB better than PCA.
- Even though number of trees in the ensemble and number of attributes to sample at each node are the two important hyperparameters of Random Forests [11], it is considerable to see the performance of RF when minimum number of leaf node observations is also optimised. It is also considerable to apply PCA to the dataset for RF.

References

1. S. Kumar, K. Agrawal and N. Mandan, "Red Wine Quality Prediction Using Machine Learning Techniques," 2020 International Conference on Computer Communication and Informatics (ICCCI), Coimbatore, India, 2020, pp. 1-6, doi: 10.1109/ICCCI48352.2020.9104095.
2. P. Cortez, A. Cerdeira, F. Almeida, T. Matos and J. Reis. Modelling wine preferences by data mining from physicochemical properties. In *Decision Support Systems*, Elsevier, 47(4):547-553, 2009.
3. Zhang, Harry. (2004). The Optimality of Naive Bayes. Proceedings of the Seventeenth International Florida Artificial Intelligence Research Society Conference, FLAIRS 2004. 2.
4. Wu, J. and CAI, Z., 2014. A naive Bayes probability estimation model based on self-adaptive differential evolution. Journal of Intelligent Information Systems, 42(3), pp. 671-694.
5. Domingos, P., & Pazzani, M. (1996). Beyond Independence: Conditions for the Optimality of the Simple Bayesian Classifier. ICML.
6. Breiman, L. Random Forests. Machine Learning 45, 5–32 (2001)
7. Tin Kam Ho, "Random decision forests," Proceedings of 3rd International Conference on Document Analysis and Recognition, Montreal, Quebec, Canada, 1995, pp. 278-282 vol.1, doi: 10.1109/ICDAR.1995.598994.
8. Caruana, Rich & Niculescu-Mizil, Alexandru. (2006). An Empirical Comparison of Supervised Learning Algorithms. Proceedings of the 23rd international conference on Machine learning - ICML '06. 2006. 161-168. 10.1145/1143844.1143865.
9. Yue Jiang, Bojan Cukic, and Tim Menzies. 2008. Can data transformation help in the detection of fault-prone modules? In Proceedings of the 2008 workshop on Defects in large software systems (DEFECTS '08). Association for Computing Machinery, New York, NY, USA, 16–20.
10. Fan, L. and Poh, K. L. (2007) 'A Comparative Study of PCA, ICA and Class-Conditional ICA for Naïve Bayes Classifier', in Computational and Ambient Intelligence. International Work-Conference on Artificial Neural Networks, Springer, Berlin, Heidelberg, pp. 16–22. doi: 10.1007/978-3-540-73007-1_3.
11. T. M. Khoshgoftaar, M. Golawala and J. V. Hulse, "An Empirical Study of Learning from Imbalanced Data Using Random Forest," 19th IEEE International Conference on Tools with Artificial Intelligence(ICTAI 2007), Patras, 2007, pp. 310-317, doi: 10.1109/ICTAI.2007.46.