Wine Quality Classification: Naïve Bayes & Random Forest Model Comparison

Rudra Grover

Rajetharan Krishnakumar

City University, MSc Data Science 2019

Brief description and motivation of the problem

- To use Naïve Bayes (NB) and Random Forest (RF) machine learning models to predict the quality of wine based on physicochemical properties as close to the assessment of a wine sommelier.
- To compare, contrast and critically evaluate NB and RF in order to understand the mechanism behind the respective machine learning processes.
- To evaluate results with existing literature using the same dataset: Appalasamy et al. 2012 [29](Naïve Bayes) ; Yesim and Ayten (2016) [27] (Random Forest)

Initial analysis of the data set

- The datasets are extracted from UCI Machine Learning Repository [1]. The original data is sourced from Cortez et al., 2009 [2].
- 2 separate datasets: red wine (1599 samples) and white wine (4898 samples).
- 11 numerical predictor variables relating to the physicochemical properties of the wine and 1 categorical predictor variable for the colour of the wine.
- 1 numerical response variable scoring the wine quality through sensory assessors.
- The correlation heatmap shows weak coefficients between the predictor variables and the target variable. This could negatively impact the overall accuracy of the model.
- Low collinearity is noticed, other than features volatile acidity—fixed acidity and free sulphur dioxide and total sulphur dioxide which is expected.
- The feature distributions of the train and test datasets are similar.
- The summary statistics show predictors with varying magnitudes and scales, thus normalization should be utile.
- Classes are imbalanced; quality ratings vary from 3 to 8, but 5th-90th percentile lie between 5 and 6. There are few extreme ratings.

Summary Statistics

	fixed_acidity	volatile_acidity	citric_acid	residual_sugar	chlorides	free_sulfur_dioxide	total_sulfur_dioxide	density	ph	sulphates	alcohol	color	quality
count	2400.000000	2400.000000	2400.000000	2400.000000	2400.000000	2400.000000	2400.000000	2400.000000	2400.000000	2400.000000	2400.000000	2400.000000	2400.000000
mean	7.591000	0.402377	0.301129	4.508771	0.066408	25.667292	92.935000	0.995390	3.251433	0.572404	10.485556	0.500833	5.767083
std	1.553837	0.191296	0.162905	4.264832	0.040188	16.837639	59.792032	0.002865	0.163200	0.170423	1.159702	0.500104	0.842494
min	4.600000	0.085000	0.000000	0.600000	0.009000	1.000000	6.000000	0.987110	2.720000	0.220000	8.400000	0.000000	3.000000
1%	5.100000	0.130000	0.000000	1.000000	0.021000	3.000000	9.990000	0.989000	2.900000	0.300000	8.700000	0.000000	4.000000
5%	5.795000	0.170000	0.010000	1.200000	0.030000	5.000000	14.000000	0.990200	2.999500	0.360000	9.000000	0.000000	5.000000
10%	6.000000	0.200000	0.060000	1.400000	0.034000	6.000000	19.000000	0.991200	3.049000	0.390000	9.200000	0.000000	5.000000
25%	6.600000	0.250000	0.210000	1.900000	0.043000	12.000000	38.000000	0.993400	3.140000	0.460000	9.500000	0.000000	5.000000
50%	7.200000	0.357500	0.300000	2.400000	0.059000	23.000000	90.000000	0.995785	3.245000	0.550000	10.300000	1.000000	6.000000
75%	8.200000	0.530000	0.400000	6.100000	0.080000	36.000000	138.000000	0.997400	3.360000	0.650000	11.300000	1.000000	6.000000
90%	9.800000	0.665000	0.490000	11.705000	0.096000	50.000000	176.000000	0.998700	3.460000	0.780000	12.200000	1.000000	7.000000
95%	10.700000	0.750000	0.570000	14.302500	0.115000	57.000000	195.000000	0.999700	3.530000	0.860000	12.700000	1.000000	7.000000
99%	12.800000	0.960050	0.710100	18.150000	0.222040	70.010000	231.030000	1.001103	3.680000	1.110200	13.500000	1.000000	8.000000
max	15.900000	1.580000	1.660000	31.600000	0.467000	131.000000	366.500000	1.010300	4.010000	1.980000	14.900000	1.000000	8.000000

Table 1

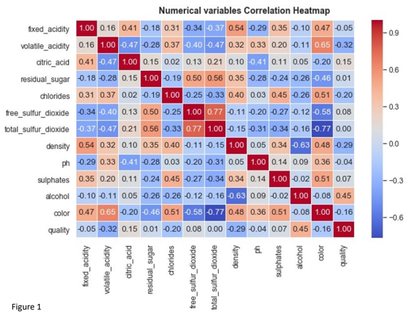

Figure 1

Summary of Naïve Bayes & Random Forest

Naïve Bayes (NB)

- It is a generative probabilistic algorithm based on Bayes' theorem, and uses posterior probability for classifying new instances.
- The classifier assumes that features are conditionally independent (naïve) and identically distributed (i.i.d). [3]
- The prior distribution is not easily determinable, and so it is optimised through an iterative selection or background knowledge.[12]
- A marginal likelihood distribution is created for each class by fitting the observed data onto assumed data distributions.
- The posterior probability for new instances is calculated using the product of prior distribution and corresponding likelihood probability estimates. The classifier selects the most probable class for each observation using a decision rule like the maximum a posteriori (MAP).

Pros

- Simple, interpretable and fast running time in comparison to other classification algorithms—following Occam's razor principle.[4][9]
- It is known to exhibit good performance with categorical data.[11]
- Performs well with a small dataset/observations even with many features as it is able to overcome the curse of dimensionality using the conditional independence assumption.[5]
- Performs well for multi-class classification and avoids over-fitting if the training sample is representative.[6]
- For continuous data, it is able to handle missing values as the marginal distribution can still be estimated from the remaining data.[8]
- In reality most data features do not hold the assumption of conditional independence. However, the classification is still proved to be effective even if the assumption does not hold.[9]

Cons

- For discrete/categorical data, it requires smoothing [7] or omission of likelihood probabilities [6] to prevent zero frequency problem.
- Posterior probability estimates may not be accurate as the conditional independence does not often hold.[10]
- The performance for a regression task is restricted by the strong conditional independence assumption.[11]
- Larger datasets could have a higher variability, and so the accuracy rate of NB could reduce in comparison to discriminative models.[5]
- It is a high bias algorithm that would not be efficient in recognising complex patterns in order to adjust to different datasets.[13]

Random Forest (RF)

- It is a discriminative model, that employs an ensemble classifier to build multiple decision trees that vote for a class (predict a value for regression).[14]
- The modal class vote determines the classification prediction (Mean/average value returned by the predictor for regression).
- RF randomly samples vectors independently from the training set, but maintains the training set distributions across trees.[14]
- The model sometimes relies on *bagging* [15] i.e. Bootstrap Aggregation proposed my Breiman (1994). It creates multiple bootstrap samples of the training data, without replacement, and uses these sets to train the trees in the Random Forest.[15][17]
- At each split of the tree a random number of features is sampled, that then grows the tree on a bagged training set.[14]
- RF proves accurate when the correlation between the trees is minimized and the strength of individual trees is maximized. [14][26]

Pros

- Breiman (2001) demonstrates the tendency of random forest generalization error to converge as more trees are added; thus, random forests avoid overfitting even if more trees are added.[14]
- Is an ensemble classifier that shows better performances than individual classifiers, which often suffer from high variance (Eg. Decision trees). Random forests instead result in a lower variance model.[16][17]
- Feature importance for Random Forests can be computed using ' out-of-bag error', which adds explainability to the model. [14][18][19]
- Generalization accuracy can also be estimated using the aforementioned 'out-of-bag estimates' that is found while training the model. This can be faster than ascertaining generalization accuracy than methods such as k fold cross validation. [21][22]
- Random Forests have robust performance even with presence of outlier data.[20]
- Require little preprocessing since it is insensitive to magnitudes of feature values.[25]
- Since decision trees within the Random Forest model can be trained at the same time, the model is naturally parallelisable.[23]

Cons

- As number of trees gets larger, the memory required by the model increases significantly, increasing overall run time. [24]
- Although the model is not prone to overfit in terms of generalization error, it is not immune to overfit in terms of difference of train and test error.
- The model does not supply parameter estimates like those given by logistic regression.

Hypothesis statement

- Both literature results are based on separate analysis on red and white wine. This research is using a combined dataset of equal random observations.
- Overall accuracy of results is expected to be lower as both type of wines depend on the same chemical properties but the score is subjective. Therefore, the combination could add noise to the models.
- RF is expected to have a better performance than NB as shown in the results of the existing literatures. This is because its ensemble nature, and robustness w.r.t imbalanced data, outliers.
- NB scores will be affected more from preprocessing steps such as normalization and PCA than RF scores.[14]
- RF is expected to be easier to tune due to the relative insensitivity of its hyperparameters in accordance with the model's accuracy.[14]
- NB is expected to have a faster running time as RF has to build many decision trees.

Training and evaluation methodology

- Select and combine randomly red (1500) and white (1500) wine observations into a single dataset and shuffle.
- Use 10-fold stratified cross validation to prevent overfitting (In the case of NB).
- Normalise the data using z-score if found to improve accuracy.
- Adjust ML model parameters to find the optimal model using grid-search.
- Perform PCA on the data and select the best performing model.
- Compare the performance of the optimised models using 80/20% train/test split:
  - Evaluate using train/test error, running time, average weighted precision, recall, F1 score and average AUC.
  - Evaluate ROC curve using one v all approach.

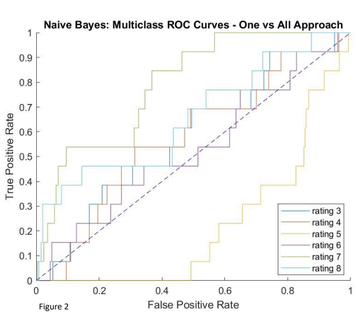

Figure 2

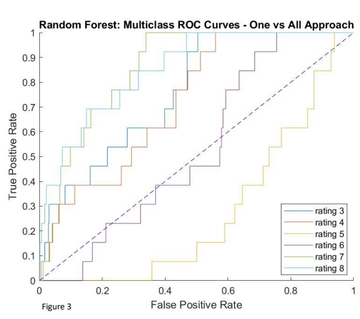

Figure 3

Naïve Bayes	Final Metrics	Random Forest
0.5983	Train MSE	0.2729
0.6550	Test MSE	0.4967
0.1459	Train Time (s)	62.9890
0.1908	Test Time (s)	62.2348
0.5195	Precision	0.7030
0.5351	Recall/Accuracy	0.6183
0.5141	F1 Score	0.6483
0.5231	AUC	0.6051

Table 2

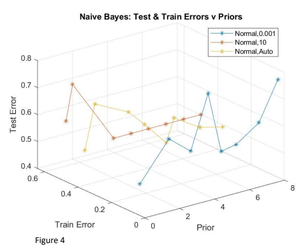

Figure 4

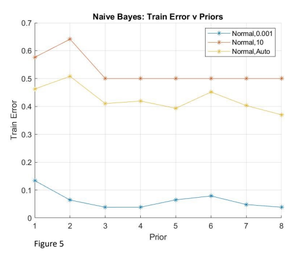

Figure 5

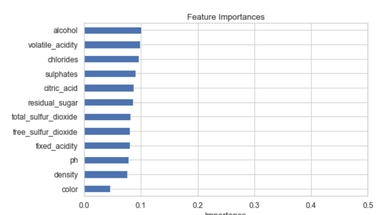

Figure 6

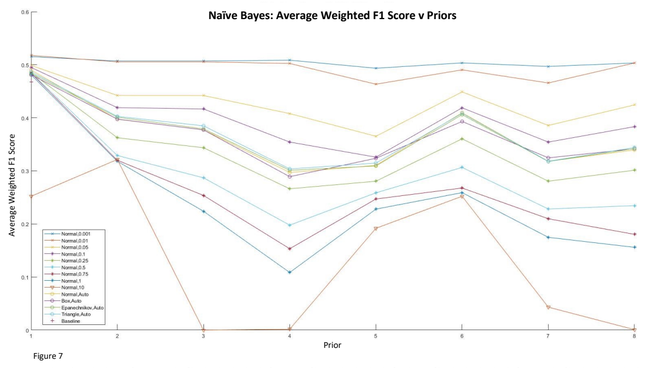

Figure 7

Prior 1 = Sample , Prior 2 = Uniform, Prior 3 = [0.5 0.1 0.1 0.1 0.1 0.1], Prior 4 = [0.1 0.5 0.1 0.1 0.1 0.1], Prior 5 = [0.1 0.1 0.5 0.1 0.1 0.1],....Prior 8 = [0.1 0.1 0.1 0.1 0.1 0.5]

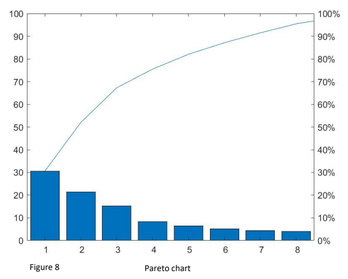

Figure 8

Choice of parameters and experimental results analysis

Naïve Bayes (NB)

- Normalization and performing PCA does improve model accuracy.
- The priors and marginal distributions were varied to explain and optimise the model performance.
- The first prior set was sample priors based on the observed class frequency. The second prior set was uniform. Thereafter, different sets were used where each set had a unique class being assigned with a high prior probability.
- All the features are numerical data, therefore the marginal distribution variety consisted of normal and kernel distributions. The kernel distribution combinations included selecting different smoother type and widths. These widths were set both manually in different magnitudes and as an automatic in-built optimisation.
- A baseline model was also created using sample priors and normal (Gaussian) distribution for the marginal distribution. This normal distribution involves fitting the data by calculating the mean and standard deviation for each class.
- The prior is very influential in the performance of the model as shown on Figure 7. The sample prior performs the best, and better than uniform prior in terms of accuracy and average weighted F1 score for all the different types of marginal distribution. This difference is especially noticeable with the accuracy. The same trend can be observed after performing the PCA, although the accuracy is mainly improved for combinations with an automatic selection of kernel widths. In the cases of high single class prior combinations, the highest accuracy also tends to be for the cases where that prior is closest to the highest sample prior.
- The test error was consistently high for cases of uniform prior irrespective of the marginal distribution or whether PCA has been performed, as shown on Figure 4.
- There is a high correlation of both accuracy and F1 score being inversely proportional to the kernel width irrespective of the prior as shown on Figure 7. The same trend can also be observed after performing the PCA. The type of kernel did not vary the performance of the models much.
- The parameters leading to the optimised NB model includes sample priors and kernel smoothing with an automatic width optimisation. The final results can be seen on Table 2.

Random Forest (RF)

For the random forest, two main hyperparameters and two others were varied and tested. These refer to number of trees in the forest and the number of features to sample at each node. The lesser hyperparameters varied were the depth of the trees and the minimum samples at which the tree would split. Performance is expected to improve as number of trees is increased and features sampled decreases.

- Grid search shows best results are achieved on test as the number of trees reaches 200, and number of features sampled is lowered to 2, corresponding with Breiman's hypothesis about the same. [14]
- Results of the best model are contained in Table 2. Noteworthy, is the highest achieved F1 score of 0.6483 on the test set and MSE of 0.4967.
- Normalization does improve model accuracy, and achieves a more accurate feature importance chart (Figure 6).
- PCA does not appreciably improve results, thus the best model does not use PCA.
- The model does overfit the data, as can be seen from the increase in Mean Squared Error from training (0.2729) to testing (0.4967) in Table 2.
- Although training and testing time increase for larger number of trees, the time taken is not inordinate (Eg.100 trees~1:00 minutes 300 trees ~2:20 minutes Table 2) and is more or less linear. For a larger dataset with higher dimensionality, time may become a constraint.
- Within this imbalanced multiclass problem, it can be noticed from the confusion matrices that the medium quality ratings are of greatest frequencies and represent a majority of classifications. Due to the same, there is a high rate of misclassification for the medium quality ratings (as is evidenced from the ROC curves).

Evaluation of results

- The reason for the initial prior having a significant impact on the NB model performance is likely to be due to the small dataset. NB algorithm uses the posterior as the next prior after each observation, and therefore due to the small size the classifier is unable to sufficiently deviate away from the initial prior. The fact that the test error was consistently high for uniform prior shows that high dependency on the initial prior for small datasets.
- The kernel width for NB determines the extent to which the data is fitted when using kernel density estimation for probability density functions. Figure 4 and 5 shows the test and train error for very small(Normal,0.001), auto (Normal, Auto) and large (Normal,10) kernel widths. It can be seen that a very small width leads to overfitting of the data as the test error is high even though the training error is low. On the other hand, a large width consistently shows high training error which represents over-smoothing.
- RF outperformed NB on the basis of accuracy (RF 0.62 vs NB 0.54) and F1 score (RF 0.64 vs NB 0.51) on the test sets . This validates our hypothesis that the ensemble nature of RF outcompetes the individual NB classifier. Our findings are also consistent with the literature that reports similar primacy of RF over NB over a majority of datasets and across the literature. [28]
- It is to be noted that the overall accuracies of both models are quite poor. The results for each model type NB and RF in the literature, correspond to data where white and red wine qualities are predicted separately[27][29] . Nonetheless our results closely mirror these. The low scores are likely due to the weak correlations between predictors and the target as well as the possibly noisy dataset. It is corroborated by the RF feature importance chart (Figure 6), with all features falling below 0.12 feature importance, which is very low.
- The high difference between train and test scores for RF shows that it is overfitting compared to NB. This could also be explained by the small size of the dataset.
- On the other hand, RF achieves overall higher accuracy scores than NB because of its high variance and low bias model that is able to approximate a more complex distribution than that of NB, which is only capable of approximating simpler hypothesis functions.
- Accuracy scores of NB were more significantly affected by preprocessing steps of normalization and PCA. This is in line with the hypothesis that RF was more insensitive to preprocessing; the many decision trees in the forest are obtained by collection of many partition rules, which doesn't change with scaling (trees see ranks as opposed to values in the features). Thus, monotonic transformations like normalization have little effect on RF accuracies.
- NB also showed greater sensitivity to hyperparameter tuning than RF. The choice of initial priors and likelihood distributions had a much larger effect on NB – error rates were significantly higher than the best NB model. The RF models on the other hand saw milder improvements than RF; initial accuracy was not much lower than the best RF model, and carefully increasing the number of trees and reducing number of features mildly improved the RF model.
- The average AUC value is higher for RF than NB, which also shows RF is a better model. The one v all approach ROC curve shows that RF is able to predict more classes better than NB.
- The NB model was far more computationally efficient than the RF model. The time to build individual decision trees is similar to the time for running a NB model. Since the RF model had to build a given number (200) of decision trees and then let those vote for classes, the RF model takes significantly more time than the NB model.

Lessons learned and future works

- Accuracies achieved with both the models was in line with those found in the literature, with the models in the literature achieving slightly higher scores. This gap could be bridged by performing various preprocessing steps:
  - Feature removal—Features that have been shown to have little or no influence on the model (from Figure 6) could be removed (Eg. Wine colour).
  - SMOTE—For unbalanced class problems like the one we tackled where the minority class is underrepresented, an oversampling technique called SMOTE is recommended [28]. It adds bias to the minority classes by providing more examples of the minority class, thus allowing more coverage for the minority class. For our problem, this should significantly improve the models ability to classify very low, and very high wine qualities since they are currently underrepresented in our data.
- Binning the target variable into low, medium and high quality likely improves the model scores, and works better for the task at hand. Actual consumers may find low, medium and high as good enough ratings for picking a wine, as opposed to a larger scale or being able to differentiate between a rating 5 and rating 6.
- Using other sophisticated models might achieve better accuracy, specifically neural network models.

References

[1] Wine Quality Data Set. *UCI Machine Learning Repository* [accessed on 26 Oct 2018]. URL: https://archive.ics.uci.edu/ml/datasets/Wine+Quality
[2] Cortez, P., Cerdeira, A., Almeida, F., Matos, T., & Reis, J. (2009). Modeling wine preferences by data mining from physicochemical properties. Decision Support Systems, 47(4), 547-553 . URL https://www.sciencedirect.com/science/article/pii/S0167923609001377?via%3Dihub
[3] Hand, D. J., & Yu, K. (2001). Idiot's Bayes—not so stupid after all? International statistical review, 69(3), 385-398.
[4] Kaviani, Pouria & Dhotre, Sunita. (2017). Short Survey on Naive Bayes Algorithm. International Journal of Advance Research in Computer Science and Management. 04.
[5] Sordo, M., & Zeng, Q. (2005, November). On sample size and classification accuracy: a performance comparison. In International Symposium on Biological and Medical Data Analysis (pp. 193-201). Springer, Berlin, Heidelberg.
[6] Farid, D. M., Rahman, M. M., & Al-Mamuny, M. A. (2014, May). Efficient and scalable multi-class classification using naive Bayes tree. In 2014 International Conference on Informatics, Electronics & Vision (ICIEV) (pp. 1-4). IEEE.
[7] Kikuchi, M., Yoshida, M., Okabe, M., & Umemura, K. (2015, August). Confidence interval of probability estimator of Laplace smoothing. In 2015 2nd International Conference on Advanced Informatics: Concepts, Theory and Applications (ICAICTA) (pp. 1-6). IEEE.
[8] Zhang, Y., Xin, Y., Li, Q., Ma, J., Li, S., Lv, X., & Lv, W. (2017). Empirical study of seven data mining algorithms on different characteristics of datasets for biomedical classification applications. Biomedical engineering online, 16(1), 125.
[9] Domingos, P., & Pazzani, M. (1997). On the optimality of the simple Bayesian classifier under zero-one loss. Machine learning, 29(2-3), 103-130.
[10] Zhang, H. (2004). The optimality of naive Bayes. AA, 1(2), 3.
[11] Frank, E., Trigg, L., Holmes, G., & Witten, I. H. (2000). Naive Bayes for regression. Machine Learning, 41(1), 5-25.
[12] Mitchell machine learning book
[13] Brain, D., & Webb, G. (1999). On the effect of data set size on bias and variance in classification learning. In Proceedings of the Fourth Australian Knowledge Acquisition Workshop, University of New South Wales (pp. 117-128).
[14] Breiman, L. (2001). Random forests. *Machine learning*, 45(1), 5-32.
[15] Breiman, L. (1996). Bagging predictors. *Machine learning*, *24*(2), 123-140.
[16] Dietterich, T. G. (2002). Ensemble learning. *The handbook of brain theory and neural networks*, *2*, 110-125.
[17] Kocev, D., Vens, C., Struyf, J., & Džeroski, S. (2007, September). Ensembles of multi-objective decision trees. In *European conference on machine learning* (pp. 624-631). Springer, Berlin, Heidelberg.
[18] Altmann, A., Toloşi, L., Sander, O., & Lengauer, T. (2010). Permutation importance: a corrected feature importance measure. *Bioinformatics*, *26*(10), 1340-1347.
[19] Menze, B. H., Kelm, B. M., Masuch, R., Himmelreich, U., Bachert, P., Petrich, W., & Hamprecht, F. A. (2009). A comparison of random forest and its Gini importance with standard chemometric methods for the feature selection and classification of spectral data. *BMC bioinformatics*, *10*(1), 213.
[20] Gislason, P. O., Benediktsson, J. A., & Sveinsson, J. R. (2006). Random forests for land cover classification. *Pattern Recognition Letters*, *27*(4), 294-300.
[21] Breiman, L. (1996). Out-of-bag estimation.
[22] Bylander, T. (2002). Estimating generalization error on two-class datasets using out-of-bag estimates. *Machine learning*, *48*(1-3), 287-297.
[23] Chen, J., Li, K., Tang, Z., Bilal, K., Yu, S., Weng, C., & Li, K. (2016). A parallel random forest algorithm for big data in a spark cloud computing environment. *IEEE Transactions on Parallel and Distributed Systems*, *28*(4), 919-933.
[24] Liaw, A., & Wiener, M. (2002). Classification and regression by randomForest. *R news*, *2*(3), 18-22.
[25] Liu, M., Wang, M., Wang, J., & Li, D. (2013). Comparison of random forest, support vector machine and back propagation neural network for electronic tongue data classification: Application to the recognition of orange beverage and Chinese vinegar. *Sensors and Actuators B: Chemical*, *177*, 970-980.
[26] Bernard, S., Heutte, L., & Adam, S. (2010, August). A study of strength and correlation in random forests. In *International Conference on Intelligent Computing* (pp. 186-191). Springer, Berlin, Heidelberg.
[27] Er, Y., & Atasoy, A. (2016). The classification of white wine and red wine according to their physicochemical qualities. International Journal of Intelligent Systems and Applications in Engineering, 23-26.
[28] Caruna, R., Karampatziakis, N., & Yessenalina, A. (2008). An empirical evaluation of supervised learning in high dimensions. In Proceedings of the 25th International Conference on Machine Learning.
[29] Appalasamy, P., Mustapha, A., Rizal, N. D., Johari, F., & Mansor, A. F. (2012). Classification-based Data Mining Approach for Quality Control in Wine Production. *Journal of Applied Sciences*, *12*(6), 598-601.