PREDICTIVE ANALYTICS

Syndicate Task #4 – Predicting Wine Quality

Syndicate 8

Tony Trinh (1099433)
Alvin Lee (1169514)
Varun Sharma (959620)
Jonno Lindsay (1026965

Table of Contents

1. Purpose 2
2. Results & Analysis 2
2.1. Support Vector Machine (SVM) 2
2.2. Bagging 2
2.3. Random Forest 3
2.4. Boosting 3
3. Limitations 3
4. Conclusion & Recommendation 4
Appendix 5
Appendix 1 – SVM Fitted vs Actual 5
Appendix 2 – VIP for Random Forest (300 trees, 9 mtry, 5 nodes) 6
Appendix 3 – Boosting & impact on loss functions 7

1. Purpose

The purpose of this report is to explore different ensemble methods, namely Bagging, Random Forest, Boosting and Support Vector Machine (SVM). These methods are built on a combination of previous (base) methods of predicting red wine quality scores.

The report will elaborate how well these new models can explain the relationship between the physiochemical properties and red wine quality score, as compared to the base models. It then will discuss some limitations and how winemakers can best use the new models to maximise profit.

2. Results & Analysis

	RMSE	MAE	MAPE	MASE	WineMakerLoss
Linear	0.5859	0.4650	8.6633	0.7294	84.6165
Stepwise	0.5872	0.4646	8.6563	0.7288	86.0358
Nonlinear	0.5835	0.4608	8.6200	0.7228	86.7770
RegTree	0.6290	0.4739	8.9016	0.7433	98.7678
NN_1_5_seed#321	0.6158	0.4858	9.0150	0.7620	81.6527
kMeans	0.6607	0.5316	9.9400	0.8339	150.0844
knn	0.6092	0.4929	9.3244	0.7732	102.4079
Bag_20	0.5669	0.4470	8.3721	0.7012	82.5532
Bag_100	0.5636	0.4460	8.3519	0.6996	82.7377
Bag_200	0.5639	0.4457	8.3460	0.6991	80.6104
RF_200_9_5	0.5052	0.3735	7.0499	0.5858	66.0805
RF_300_9_5	0.5046	0.3735	7.0427	0.5859	65.7197
RF_700_2_20	0.5291	0.4123	7.7614	0.6468	77.0576
SVM_linear	0.5886	0.4587	8.5195	0.7195	83.9995
SVM_polynomial	0.8273	0.5374	9.9873	0.8430	121.6862
SVM_radial	0.5850	0.4385	8.1612	0.6878	79.1204
Boost_0.01	0.6499	0.5441	10.2159	0.8535	110.9334
Boost_0.99	0.5861	0.4651	8.6642	0.7295	84.6063
Boost_0.98	0.5861	0.4651	8.6642	0.7295	85.7257

Figure 1: Prediction Model Output Summary

2.1. Support Vector Machine (SVM)

The SVM models were trained utilizing four different kernels (linear, radial, polynomial & sigmoid) though sigmoid is not included due to its poor performance. The main takeaway from the SVM models can be seen in *Appendix 1.* Out of the remaining three, the model with polynomial kernels performed the worst. Its actual vs fitted graph shows the model tends to overpredict when the actual score is between 4 to 6, yet underpredict when the actual score is 7. This pattern adversely affects SVM_polynomial's predictive performance in terms of the specific winemakers loss function.

The linear and radial functions tend to predict values below 7 which is why these models perform slightly better when assessing against the winemakers loss function. The linear model tends to overpredict lower values (3-4) and the radial model does a slightly better job at compensating for this which is why it performs best out of the three.

2.2. Bagging

Several bagging models were run with an improving predictive power as the number of bags were increased with the best model using bags=200. The top three important variables were 'Va', 'Alc' and 'Sulphates', and these were the same variables identified by the regression tree as being important.

The bagging models improved all loss function metrics compared to the regression tree, due to reducing the variance of the regression tree. Although bagging models predict better than regression

trees, it is less explainable, and a winemaker would struggle in determining *why* certain wines score the way they do or explain the impact each variable has on a final quality score.

2.3. Random Forest

Multiple permutations of random forest models were run, with trees ranging from 100 to 1000, mtry values between 2 to 10 and node sizes between 2 to 20. Generally speaking, the winemaker loss functions tended to improve with increasing mtry values and adverse performance if node size was greater than 5. In all the different simulations, it appeared that the tree size did not follow any rules. The best random forest model contained 300 trees.

Although the random forest is an extension of the bagging models, there are differences in the variable importance order. 'Alc' followed by 'Sul' are the most important variables followed equally by 'VA' and 'TSD' (Appendix 2).

2.4. Boosting

Boosting was run using nu values ranging from 0.01 to 0.99 with the expectation that the lower the nu value, the better the predictive performance. However, this was not the case and all loss functions improved as the nu values approached 0.99. Appendix 3 plots provide some insight as to why this is occurring. At the slowest rates of learning, the boosting algorithm only included 'Alc' and 'VA' as being important variables, improving the loss function by approximately 0.75% and 0.26% respectively.
The boosting models with nu=0.99 selected nearly all variables at least once through the model construction. It also identified that 'Alc' and 'VA' were the most impactful variables reducing the loss function by approximately 0.70% and 0.25% respectively, even though they were not necessarily selected the most. The selection of other variables and their ability to improve the loss function also highlighted that other variables have a part to play in determining quality score and that just 'Alc' and 'VA' alone are insufficient.

	Intercept	FA	VA	CA	RS	Ch	FSD	TSD	Density	pH	Sulphates	Alc
Multi Linear Regression Coefficients	14.900	0.0124	-1.0210	-0.1296	0.0062	-2.0590	0.0041	-0.0036	-10.2400	-0.5783	0.8651	0.2908
Boosting_0.99 Coefficients	7.480	0.0102	-1.0235	-0.1294	0.0053	-2.0556	0.0041	-0.0036	-8.4220	-0.5922	0.8595	0.2929

Given the boosting is improving on the multi-linear regression model, at high learning rates the coefficients in the table above seem to indicate similar relationships and magnitudes between wine properties and quality score.

3. Limitations

The common weakness across most of the ensemble methods, except for Glmboost, are their inability to extrapolate predictions outside of the observed range and to provide a causal relationship or an easy way to interpret their results. Furthermore, the dataset only contains quality scores between 3 & 8, which intensify the aforementioned limitation of the models. That said, ensemble models fare better than their respective base models in several areas.

Firstly, bagging improved prediction stability, as mentioned in section 2.2, and reduced the chance of over fitting as compared to single regression tree (base model) by averaging the predictions from multiple trees. However, there is no guarantee that all training data points will be used.

It is also worth noting that Random Forest (RF) shares the same weaknesses with bagging, with more randomness in terms of how trees are constructed. This may mean better performance with some

(e.g. 300 tree, 9 mtry and 5 nodes as depicted in figure 1) but not necessarily across all configurations (full list of 400 Random Forest models are available upon request).

On the other hand, Glmboost could not outperform multi-linear regression, as per figure 1, so it seems boosting a series of component-wise univariate linear models, aka "weak learners", may not be a good option in this instance. Though Glmboost produced a set of coefficients for interpretation, these values change dramatically as nu changes so they should be taken with a grain of salt. For example, coefficients of CH & pH increased almost 3 times between nu = 0.05 and nu=0.1 (below).

	Intercept	FA	VA	Ch	TSD	pH	Sulphates	Alc
Nu = 0.05	-2.1980	0.0018	-0.9759	-0.4479	-0.0013	-0.1054	0.4634	0.2715
Nu = 0.1	-1.6453	0.0036	-0.9878	-1.4038	-0.0021	-0.3353	0.6911	0.2879

Lastly, SVM models with radial & linear kernel, despite fair predictive power, still run the risk of over fitting due to their nature, especially in this case where quality score data does not reflect the population.

4. Conclusion & Recommendation

The winemaker's ultimate objective is to maximise profit, which is achieved by pricing and advertising the product according to its quality. The challenge faced with most of the proposed models is their inability to extrapolate beyond the ranges of the data (3 to 8). As mentioned in report #3, this extrapolation issue still occurs in neural networks, regression trees, knn and k-mean models and as a result, strategies to predict wines that may fall out of these ranges must be employed.

Higher quality wines will generate exponentially increasing profits and it is crucial in being able to identify these higher quality wines. As such, one recommendation for winemakers is to do a two-step approach when predicting wine scores:

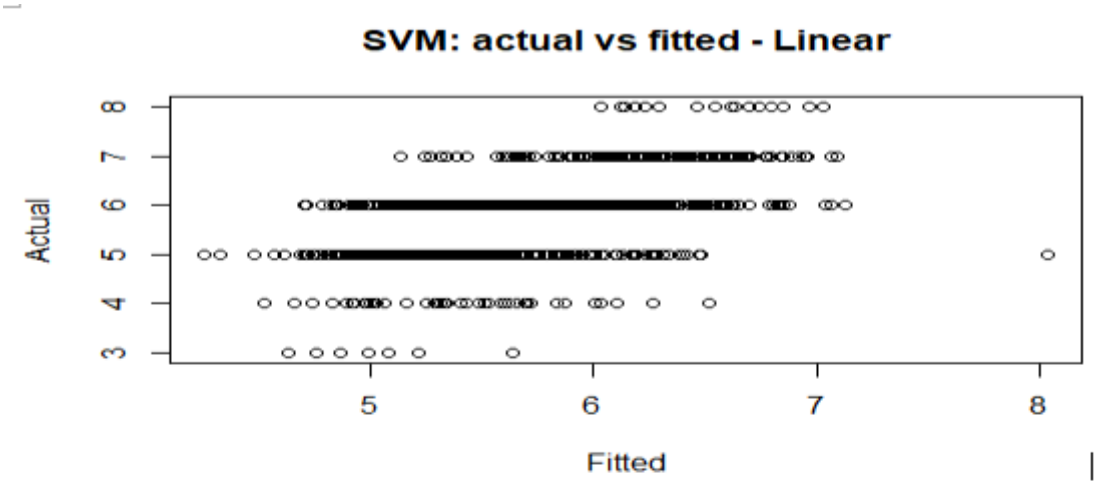
1. All wines should be run against the multilinear regression model as this model is able to extrapolate beyond the bounds of the input data. For wines with a quality score ≥ 8, it is recommended to use industry reports and expert opinion.
2. For wines with a quality score of ≤ 8, use the random tree model with trees=300, mtry=9.

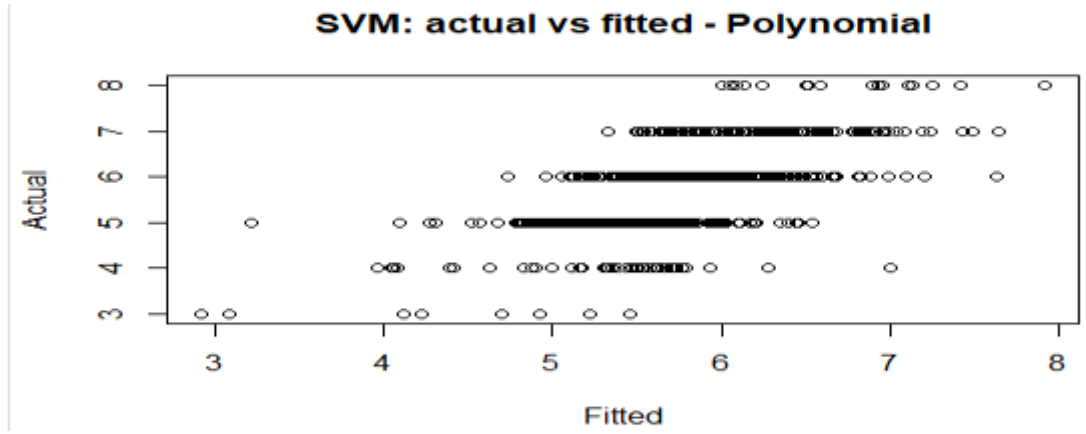
This means that if there is a wine with superb physiochemical properties, it has a chance of being firstly identified by the linear model. Furthermore, solely relying on the outlined predictive models is short-sighted. There are many non-physiochemical variables that impact wine quality score i.e. altitude, humidity, soil quality, weather conditions etc. Therefore, we implore the winemaker to draw upon their experience and understanding of other micro-factors, to subjectively assign a quality score to each wine which there are indicators of being defined as high quality as it is crucial to not miss out on being able to market and sell these premium wines. For wines which score ≤ 8, the random forest model should be able to predict very well.

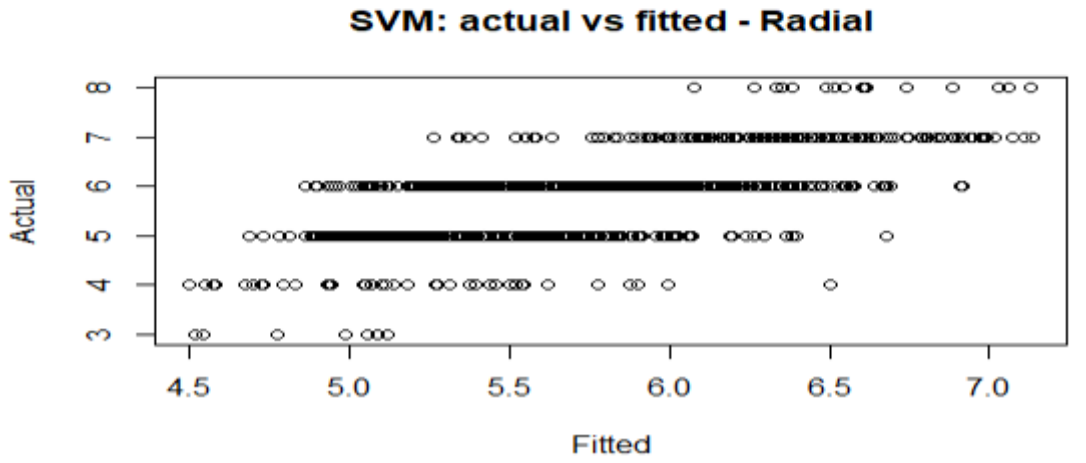
Together, this two-step approach will help the winemaker to assess wine quality across the score spectrum and make informed decisions to optimise their pricing and advertising strategies.

Appendix

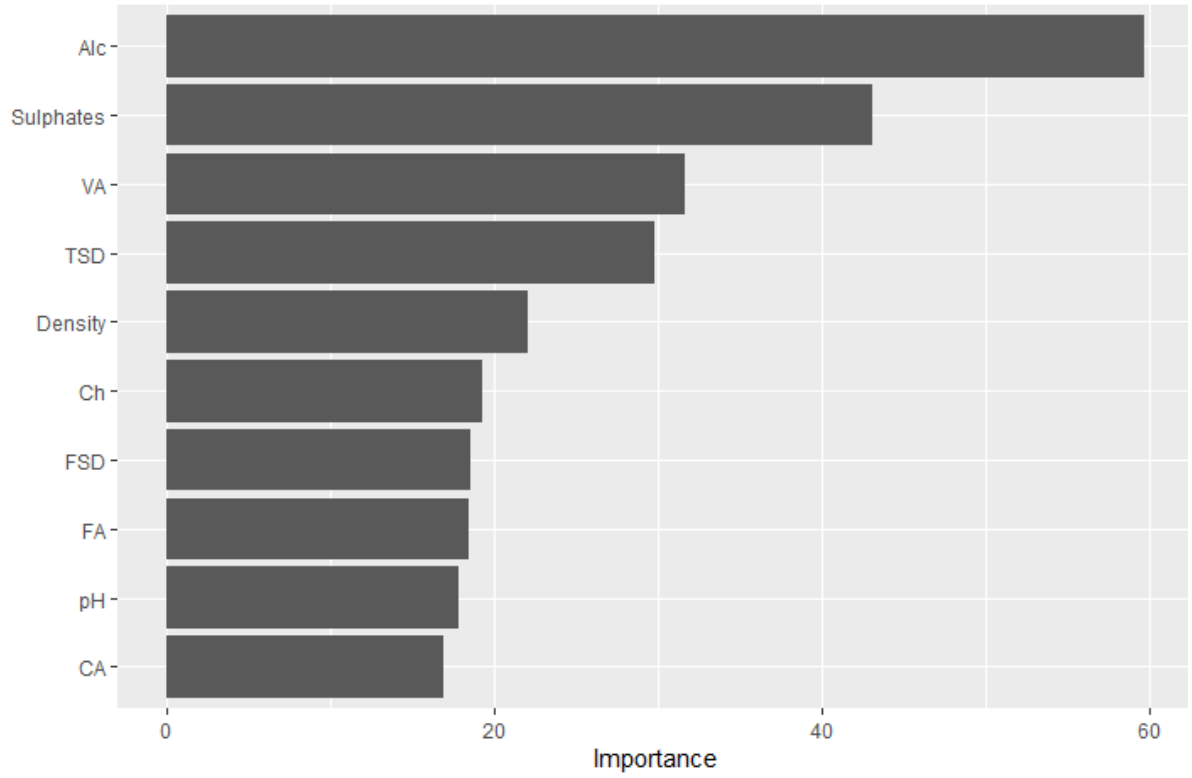
Appendix 1 – SVM Fitted vs Actual







Appendix 2 – VIP for Random Forest (300 trees, 9 mtry, 5 nodes)



Appendix 3 – Boosting & impact on loss functions

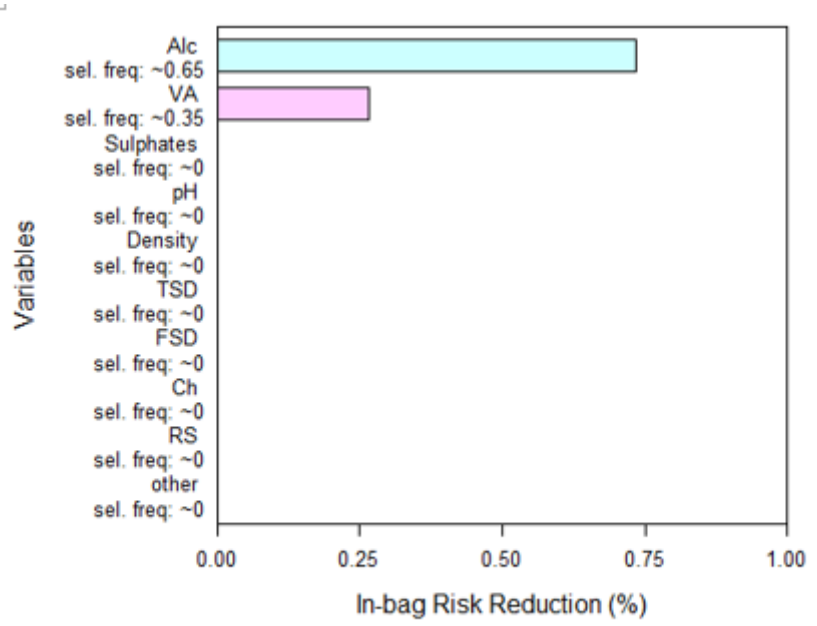


Table Above: nu=0.01

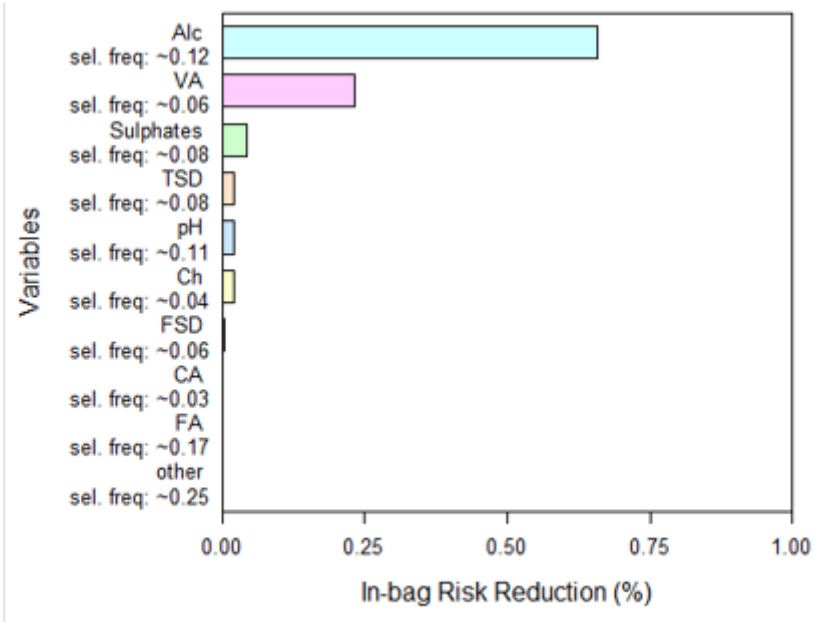


Table Above: nu=0.99