AY 2019/20 Semester 2

BUILDING A GENERALISABLE RED WINE QUALITY CLASSIFIER

BT2101 DECISION MAKING METHODS & TOOLS
PROJECT REPORT

Prepared by:
Axel Lau (A0197376L)
Sushmit Sharma (A0201698Y)
Tom Joju (A0200047Y)
Christopher Liew (A0189987U)

Special Thanks:
Dr. SungYong Um
(Information Systems & Analytics)

TABLE OF CONTENTS

1. BUSINESS PROBLEM

2. EXPLORATORY DATA ANALYSIS

3. INTERACTION TERMS

4. DATA PREPROCESSING

5. HYPOTHESES

TABLE OF CONTENTS

6. MODEL SELECTION & EVALUATION

7. MAIN MODEL: RANDOM FORESTS

8. EVALUATION MODEL: SVM GAUSSIAN RBF

9. COMPARING MAIN VS. EVALUATION MODEL

TABLE OF CONTENTS

1. Problem Description

We are a team of data scientists from the E & J Gallo wine company investigating the factors which determine the quality of *Vinho Verde* red wine. This will be done by applying data driven methods when analysing the reviews of established wine critics and the chemical compositions of the respective red wines.

Through the analyses of the red wine quality dataset, our primary aim is to build a classification model to predict the quality of a given sample of red wine using its characteristics. It will predict the categorical target variable that is the quality of red wines. (i.e. Red wines are scored with increasing merit on a scale of 0 to 10, with the worst red wines having quality = 0 and the best quality = 10).

An effective model would help our clients assess the quality of their new or existing red wines, before they reach the market. Such knowledge is invaluable in helping them focus their marketing and production efforts, according to our model's predictions. Additionally, our model would give our clients an idea of how they could tune the characteristics of their existing red wines to improve their quality. Essentially, we want to create a quicker, more accurate and cheaper method for wine makers to predict how their products will do in the market.

Overview of Project

The goal of our project is to predict a binary target variable (Good Quality = 1 or Other Quality = 0). This report will explain our entire process, starting from data exploration to the evaluation of the final models chosen.

We will first use Exploratory Data Analysis, ANOVA and existing domain knowledge to identify the key characteristics that might be crucial to building an accurate and robust predictive model. We will then formulate our hypotheses using these key characteristics and test them using our pipeline models.

We will attempt several models and narrow down to one main and one evaluation model. The justification, along with an explanation of the results we observed will be provided as well. Finally, we will proceed to compare both models and discuss any limitations and areas for improvement.

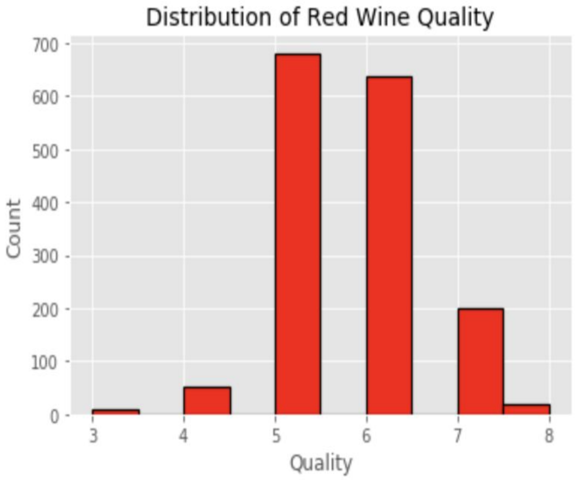
Description of the Red Wine Quality dataset

The red wine quality dataset consists of *12 variables* and *1599 observations* obtained from the UCI Machine Learning Repository. Each observation belongs to a red wine sample from Portugal which has been scored based on wine tasting evaluations, giving it its quality feature. The features observed were *fixed acidity, volatile acidity, citric acid, residual sugar, chlorides, free sulfur dioxide, total sulfur dioxide, density, pH, sulfates and alcohol*.

2. Exploratory Data Analysis

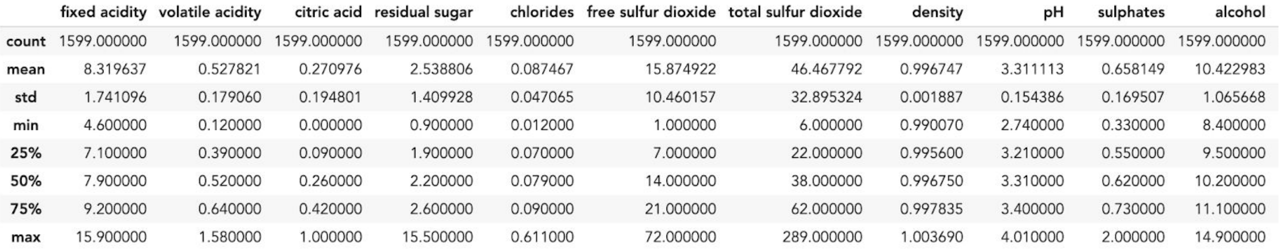
We observed that all 1599 rows across all 12 columns contained no missing data. Hence no imputation of missing data is necessary and we can continue with further data exploration.

Exploring the Quality target variable



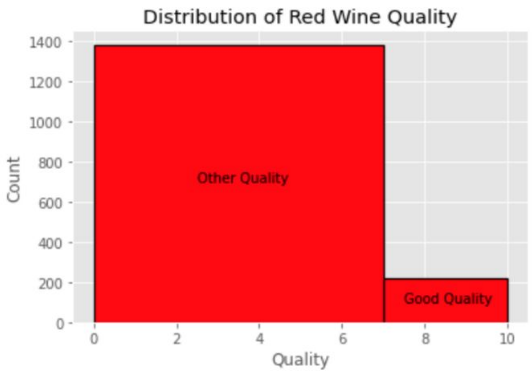
Given the categorical nature of the quality target variable and the fact that we are most concerned with high quality red wines, we dichotomised our quality variable into a binary target variable ***good_quality***, where **quality < 7 (*Other Quality*) & quality ≥ 7 (*Good Quality*)**. Furthermore, this would give us a greater number of samples for each target class, enabling our classification models to learn better.

Exploring the numerical features



	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcohol
count	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000
mean	8.319637	0.527821	0.270976	2.538806	0.087467	15.874922	46.467792	0.996747	3.311113	0.658149	10.422983
std	1.741096	0.179060	0.194801	1.409928	0.047065	10.460157	32.895324	0.001887	0.154386	0.169507	1.065668
min	4.600000	0.120000	0.000000	0.900000	0.012000	1.000000	6.000000	0.990070	2.740000	0.330000	8.400000
25%	7.100000	0.390000	0.090000	1.900000	0.070000	7.000000	22.000000	0.995600	3.210000	0.550000	9.500000
50%	7.900000	0.520000	0.260000	2.200000	0.079000	14.000000	38.000000	0.996750	3.310000	0.620000	10.200000
75%	9.200000	0.640000	0.420000	2.600000	0.090000	21.000000	62.000000	0.997835	3.400000	0.730000	11.100000
max	15.900000	1.580000	1.000000	15.500000	0.611000	72.000000	289.000000	1.003690	4.010000	2.000000	14.900000

We observed that most features are approximately normally distributed, with the exception of *residual sugars, chlorides, free sulfur dioxide, total sulfur dioxide* and *sulphates*. However, it has to be noted that the two target classes are not well separated by the features.



Plausible Imbalanced Classification

Furthermore, as observed, there is an imbalance in the number of samples for red wines of *Other Quality* and those of *Good Quality* . This imbalance might lead to an eventual *accuracy paradox* as our classification algorithm could predict *Other Quality* with an ~86.4% accuracy using the underlying distribution. (13.6% of all samples are Good Quality; 86.4% of all samples are of Other Quality)

Implications of an Imbalanced Dataset

Data imbalance can be a major downside when we use various classification techniques as classifiers operate on the basic assumption that the size of each target class is balanced[1]. However, given our imbalanced data set, our classifiers may become biased towards the majority class (*Other Quality*). In the case of our dataset, our models could correctly classify all "*Other Quality*" red wines and would obtain a relatively high 86.4% accuracy rate.

This led us to two major implications for our project. The first affected the metric we used to evaluate our models. Due to the imbalanced nature of our dataset, we had to use metrics other than *accuracy* to judge our models. Another implication was on the models we used. Many models have an underlying assumption that the input data is balanced and may not perform well on imbalanced data, such as SVM and Random Forest[2]. These models may need certain hyperparameters to provide more reliable results. As such, we had to keep these considerations in mind when we chose our models and evaluated the output.

3. Interaction Terms

Total Acidity Interaction Term

If a wine is too tart the acidity is too high. If a wine is soft and flabby the acidity is too low. To determine if we've gotten our acidity pegged at a reasonable quantity, we have to measure titratable and volatile acid concentrations to determine what the total amount of acid present is. Thus, to account for total acidity, we created an interaction term of *volatile acidity + fixed acidity* called ***Total Acidity.***

The pH * sulphates interaction term

Wines with higher pH require more sulfites to protect them from oxidation because it decreases the sulfites' effectiveness. With higher pH wines, there is a recommended amount of sulfur one can add to make a wine microbiologically stable. Considering this, we have included a ***pH * sulphates*** interaction term.

[1] Solving an Imbalanced Dataset, *Towards Data Science*, 2019
[2] https://machinelearningmastery.com/what-is-imbalanced-classification/

4. Data Preprocessing

Handling of Missing Values & Outliers

Given that our dataset contains no missing data, no imputation of missing data was necessary. We defined any values exceeding the interquartile range[3] to be outliers. However, with generalisability being key we retained the outliers and traded higher bias for lower variance.

Feature Selection with ANOVA F-Value

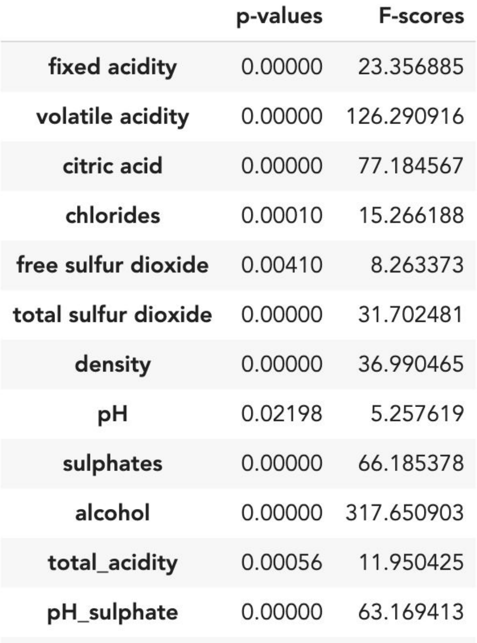
$$F - value = \frac{Variance\ between\ groups}{Variance\ within\ groups}$$

Given that all of our features are *numerical* and we have a *binary target variable*, we decided on the ANOVA F-value as our feature selection method. The F-value examines if, when we group a numerical feature by the target variable (E.g. *Alcohol* by *Good Quality* & *Other Quality*), the means for each group are significantly different.

Thus, the larger the F-value, the more likely the groups have different means and the feature is useful for our classification models in discriminating between *Good & Other Quality*.

After computing the F-values for each feature, we computed their respective p-values and selected features with p-values ≤ 5%. (Refer to table) The feature *residual sugar* was not selected.

Classification models without embedded feature selection methods like Logistic Regression, SVM & Naive Bayes will benefit from manual feature selection, which reduces overfitting, speeds up algorithmic runtime and may improve accuracy.



	p-values	F-scores
fixed acidity	0.00000	23.356885
volatile acidity	0.00000	126.290916
citric acid	0.00000	77.184567
chlorides	0.00010	15.266188
free sulfur dioxide	0.00410	8.263373
total sulfur dioxide	0.00000	31.702481
density	0.00000	36.990465
pH	0.02198	5.257619
sulphates	0.00000	66.185378
alcohol	0.00000	317.650903
total_acidity	0.00056	11.950425
pH_sulphate	0.00000	63.169413

Train-Test Split & Data Leakage Minimisation

Following feature selection, we split our dataset for training and testing using the *holdout validation method*. The dataset was subsequently split into 75% for *training* and 25% for *testing*. Furthermore, by splitting the dataset prior to feature scaling, we eliminated the risk of *data leakage* from the testing to the training set which would have been detrimental to the predictive power of our classification models on new test data.

[3] Interquartile Range: Lower Fence: Q1-(1.5*IQR) and Upper Fence Q3+(1.5*IQR*)*

Feature Scaling

Having kept our outliers to minimise variance, we used the *RobustScaler* which scales a feature vector by subtracting the median and dividing by the interquartile range (set to *75% - 25%* quartiles). This allowed us to minimise the effect of outliers on our scaled values. Thus, our classification models which used euclidean distances or gradient descent to optimise their decision boundaries (e.g. SVM), were able to converge more quickly. This in turn allowed us to develop models which are better suited for real-world applicability, especially as our firm's repository of red wine data scales up in the future. We also applied row-wise *Standard Scaling* to centre the distribution of our features to that of a normal distribution. This was to allow our models to better discriminate between our target classes.

5. Hypotheses

Following our data preprocessing pipeline, we decided on 3 key hypotheses to be tested based on both statistical merit (*ANOVA F-scores*) and the domain knowledge of wine experts. They are:

1. *Alcohol content* in red wines is a statistically significant feature in determining red wine quality.
2. *Volatile acidity* is a statistically significant feature in determining red wine quality.
3. *Citric acid* content is a statistically significant feature in determining red wine quality.

The 3 features being tested in our hypotheses had the highest *ANOVA F-scores*, indicating higher variance *between groups* vs. *variance within groups*. Thus, they are statistically useful features in discriminating between *Good Quality* and *Other Quality* red wines.

Furthermore, based on the knowledge of wine experts, higher *alcohol* content was found to have significant consumer appeal as it gives rise to rich and ripe flavours, whilst *citric acid* is generally added to give red wines a "fresh" flavour. On the other hand the presence of *volatile acidity* in any appreciable quantity is an indicator of spoilage in wines. Thus, expert knowledge has thoroughly informed our choices of hypotheses for our investigation.

Our hypotheses served as evaluative indicators for our final operational classification model, given that any robust model that we seek to create should employ the *3* features highlighted in our *3* key hypotheses which should be statistically significant by its estimates. As such, we assessed our models based on the hypotheses we have proposed.

6. Model Selection & Evaluation

Model development & deployment

In our *red wine* classification project, we have selected and applied the following classification models to our *red wine* dataset: *1) Logistic Regression, 2) Linear SVM, 3) Gaussian SVM, 4) Naive Bayes, 5) Random Forest.*

To optimise the performance of our models, we used the GridSearchCV model selection algorithm to tune our models' hyperparameters and selected the best model based on *macro averaged F1-Scores* rather than *accuracy or F1*. Our rationale for this will be explained below.

Metrics for Evaluation

Accuracy = (TP + TN)/ All Samples

Accuracy would not be a useful metric for evaluating our model given our aim of measuring how well our models classify *Good Quality* red wines. This is due to the fact that the proportion of *Other Quality* is significantly higher, leading to a skewing of our accuracy result.

Recall = (TP) / (TP + FN)

We selected *recall* as a key metric for model evaluation because *false negative*s are undesirable for our business problem. This is because we do not want to leave out any *Good Quality red wines* and the features which determine its Good Quality. This will maximize our clients' profits.

Precision = (TP) / (TP + FP)

Precision will help us to gauge the '*error rate'* of a classification model, in terms of the rate at which they might predict *false positives.* This is undesirable as we do not want to classify *Other Quality* red wines as *Good Quality* and in turn harm the reputation of our firm. Thus, precision was another key metric for model evaluation.

F1-Harmonic Mean

$$F_1 = 2 * \frac{precision * recall}{precision + recall}$$

With *Recall & Precision* being key evaluation metrics as outlined above, we decided to use the *F1-score* as an evaluation metric as it takes into account both *Recall & Precision* with equal weight. Specifically, we used the *macro averaged F1-score* as a key evaluation metric.

$$F1_{class1} + F1_{class2} + \cdot\cdot\cdot + F1_{classN}$$

The *macro averaged F1-score* calculates *F1-scores* for each of our classes before summing them up. This penalised models which did not perform well with our minority *Good Quality* red wine class and accounted for our underlying class imbalance.

Area Under Receiver Operating Characteristic

The AUROC takes on values from 0 to 1, with 0 meaning that the model does not discriminate & separate our target classes at all and 1 meaning that the model can perfectly discriminate & separate our target classes. Thus, we selected models with high AUROC.

Preliminary Results

The preliminary results from fitting our models with the training data are summarised below:

Model	Recall (Good Quality)	F1	F1 Macro	AUROC
Logistic Regression	0.78	0.52	0.70	0.77
SVM Linear	0.87	0.74	0.63	0.87
SVM Gaussian RBF	0.51	0.92	0.77	0.92
Naive Bayes	0.66	0.84	0.73	0.86
Random Forest	0.64	0.62	0.79	0.92

Assessing Each Model

Logistic Regression

Logistic Regression has many merits as a binary classifier. Firstly, it is very easy to implement, and has a relatively low time complexity. Despite our imbalanced dataset, Logistic Regression is able to balance class weights using *"class_weight"* hyperparameter. Thus, upon balancing, its *recall* improved significantly from 0.21 to 0.78.

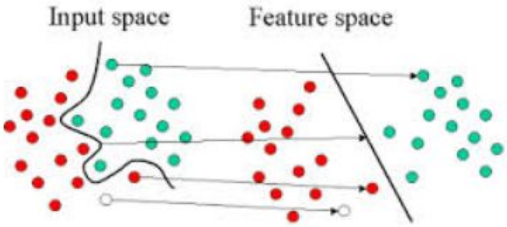
However, Logistic Regression is limited by its binary outcome variables. This means that when we convert a continuous feature such as wine *quality* (from 0 to 10) into a binary categorical variable such as *Good Quality* and *Other Quality*, we sacrifice the precision of the dataset. Furthermore, it is imperative that observations are *i.i.d.*[4] as non-independent features lead to larger emphasis being placed on them. To combat this, we did use ANOVA during our preprocessing to free our data from the issue of multicollinearity. The absence of linear features affected the performance of logistic regression as well, leading to a comparatively low AUROC and *macro-averaged F1-scor*e. Lastly, logistic regression models appear to have more predictive power than they actually do as a result of sampling bias. This means that they might be overfitted to our dataset and might not be that applicable to other red wine sample data. While this is minimised by calculating the 10-fold cross-validated F1-score, it is still a factor that affects Logistic Regression more significantly as compared to other models.

[4] *i.i.d. : Independently & Identically Distributed*

Thus, we decided to eliminate this model, primarily because its simplicity and possible overfitting could lead to a lack of accuracy in predicting wine quality. Hence, it would not be a suitable model for us to use to predict wine quality.

Support Vector Machine (Linear)

A linear Support Vector Machine (SVM) can be used for both classification and regression challenges. Each data item is plotted in a n-dimensional space, where n the number of features we have. SVM then finds the hyper-plane that differentiates the two classes the best. A margin is the distance between the support vectors of each class and the hyperplane. SVM essentially chooses the hyperplane that maximises this margin. In a linear SVM, the algorithm assumes linear separability for each data point, and simply seeks to maximise the distance between the plane and the point.



The assumption of linear separability is often restrictive and produces a less accurate fit for the model. Moreover, being able to utilize transformation kernels in data allows for a better fit for the data points. This is a drawback, as linear SVM is not expected to perform as well with low-dimensional datasets such as ours as it is harder to linearly separate the data points due to the fewer dimensions. Hence, we did not expect our data to perform as well for this model. However, it outperformed the Gaussian SVM in its *recall* score. A possible reason for this is that the data points tend to be more distinct in a linear SVM as compared to the Gaussian RBF SVM. This can allow the model to predict the *Good Quality* wine with a higher accuracy, explaining the better *recall* score. Additionally, it suggests that our data points may be linearly separable to a sufficient extent to explore linear classification methods.

Some merits of Linear SVM include the fact that it attempts to find the best hyperplane that separates the classes, and this reduces the risk of error on the data, while logistic regression does not. Logistic Regression can have different decision boundaries with different weights that are near the optimal point. This could explain the higher AUROC and recall scores. SVM also has a L2 Regularization feature, giving it good generalization capabilities to reduce overfitting. Lastly, the SVM model accounts for the imbalanced dataset by adjusting the Cost parameter for each target class based on the proportion of observations for each target class.

However, this model requires a long training time for large datasets. Although this is not specifically an issue for our dataset, it can affect the performance when using datasets containing more than 1500 observations. A poor *F1-Score* for Linear kernel SVM is hence, not surprising due to the fact that variables for *Good Quality* red wines are within a certain range and vary in accordance to other features such as *acidity* and as such cannot be linearly separated. Therefore, we decided to reject the Linear SVM model as well, given the non-linearity of our red wine dataset and its low dimensionality.

Naive Bayes Classifier

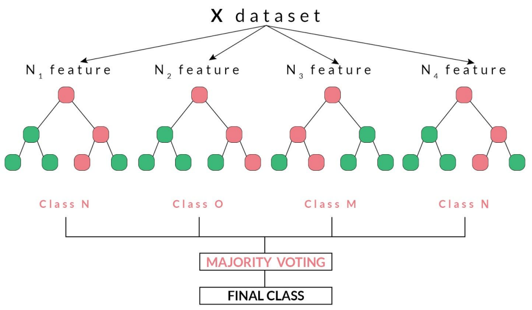
The Naive Bayes classifier is based on a generative probability model used for various classification problems. An advantage of the Naive Bayes classifier is that it works well with small datasets like ours. Furthermore, Naive Bayes can also handle irrelevant features by placing a higher weightage on more important features. This is probably the reason behind its relatively high *macro-averaged F1-Score* and *AUROC.*

However, the Naive Bayes' model assumes that all variables are completely independent of one another, which is almost impossible to fulfil. While multicollinearity is minimised through ANOVA feature selection, it is still insufficient in ensuring complete independence amongst features, as none of our features have Pearson's correlation coefficients of 0 in relation to other features (e.g. volatile acidity and citric acid have a correlation of -0.55). This is because our features are linked to each other. Hence, we decided to reject this model in spite of its decent performance as it will not be generalizable to other wine samples.

7. Main Model: The Random Forest Classifier

Intuition and rationale

The random forest classifier is an ensemble method learning method which taps into the "wisdom of the crowd" in order to create a robust classification model out of weak learners like decision trees, which are prone to overfitting. Random forest circumvents the weaknesses of decision trees by firstly, bootstrapping the dataset and building uncorrelated decision trees out of these random subsets and secondly aggregating the predictions of the forest of decision trees through 'hard voting'. Consequently, this allows our random forest classifier to be less prone to overfitting and thus have lower variance without introducing bias, overcoming the issue of the bias-variance tradeoff.



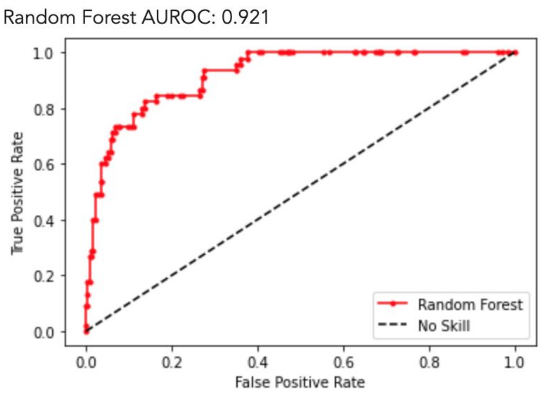
Our rationale for choosing the random forest classifier foremost lies in its performance relative to the other classification models that we have tried. Given that minimising both false positives and false negatives are a priority, *recall* and *precision* are integral. Thus, random forest was selected as it produced the highest *macro-averaged F1-Score* of 0.787 and *AUROC* of 0.921, indicating its balanced performance in recall and precision and high discriminating power.

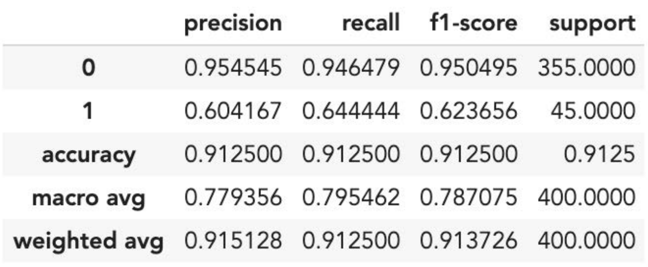
Furthermore, random forests have been empirically shown to scale and adapt well with increasingly large datasets. This is demonstrated by random forest's ability to handle high dimensionality without overfitting whilst remaining consistent in its predictive accuracy, even when a dataset has missing data.[5] Lastly, given the inherent imbalance in our target classes of Good Quality and Other Quality, random forests' "*class_weight*" hyperparameter is useful in dealing with the imbalance.

Application & Results

In order to yield an optimal *macro averaged F1-score* whilst being cautious about overfitting, we engaged in hyperparameter tuning of the following parameters: 1) *n_estimators*, 2) *max_depth*, 3) *class_weight*. By controlling the number of trees in our forest and the vertical depth of each tree, it allowed us to prevent our classifier from learning relations specific to individual observations. Additionally, by tuning class_weight, we can handle the imbalances in target class and prevent the model from succumbing to an accuracy paradox. We did not feature scale or engage in any feature selection for the random forest model as it is 1) Insensitive to euclidean distances and 2) Engages in its own 'auto' feature selection through splitting its nodes based on information gain.

Hyperparameter tuning was applied in a two step process with the more efficient Randomised Search tuning algorithm being applied first to find candidate hyperparameters before applying the more exhaustive Grid Search tuning algorithm. Our final hyperparameter results are as follows: **{'class_weight': 'balanced', 'max_depth': 10, 'n_estimators': 800}**. We also used *GINI impurity* to measure information gain, which is computationally less expensive relative to the logarithmic *Entropy*. Our results are as follows:





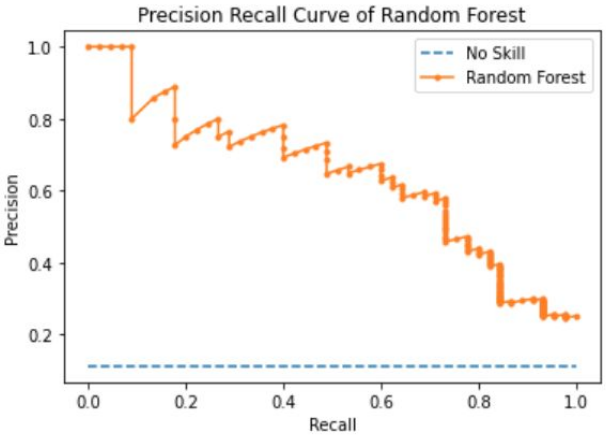
	precision	recall	f1-score	support
0	0.954545	0.946479	0.950495	355.0000
1	0.604167	0.644444	0.623656	45.0000
accuracy	0.912500	0.912500	0.912500	0.9125
macro avg	0.779356	0.795462	0.787075	400.0000
weighted avg	0.915128	0.912500	0.913726	400.0000

[5] Tang F, Ishwaran H. Random Forest Missing Data Algorithms. *Stat Anal Data Min*. 2017

Explanation of Results

Since our business problem is centered around identifying and classifying *Good Quality* wines, we will specifically evaluate the model based on its performance on the *Good Quality* class. Random forests' results for precision and recall is relatively average as compared to the other classification models we have tested. The reason for this might be due to the use of *shallow trees*, with a '*max_depth'* of 10 in our random forest classifier. This further minimises variance and increases the generalisability of our classifier.

The random forest yielded the highest *macro averaged F1-Score* amongst all the models we have tested at 0.787. The reason for this lies in its relatively linear precision recall curve for *Good Quality* red wines as show below:



Thus, we can observe that the tradeoff between *precision* and *recall* is fairly balanced as compared to the other classification models which have shown curves that skew to recall, exhibiting an "L-shaped" precision recall curve. Therefore, random forests offers us the best balance between precision and recall out of all models and consequently produces the best *macro averaged F1-Score*. Hence, it is suitable for our Good Quality red wine business problem.

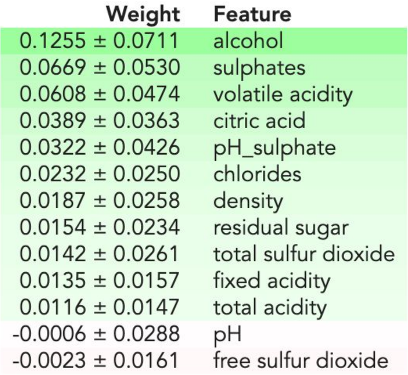
The *AUROC* of random forest was surprisingly high at 0.921 as compared to the other methods. However, we deduced that the *AUROC* might be positively biased, and is thus an over-optimistic estimation of the discriminatory power of random forest. This is as the ROC curve is based on the true positive and false positive rates of a classifier, which do not depend on the distribution of our target class.[6] Thus, due to the large skew to Other quality in our dataset, the *AUROC* is inflated. As such, we propose using the less biased *macro averaged F1-Score* and the *precision-recall* curve of good quality wines to evaluate random forests, as it more accurately evaluates the predictive ability of random forests.

[6] Tom Fawcett ROC Graphs: Notes and Practical Considerations for Researchers, *HP Labs*, 2004

Hypotheses Testing & Feature Importance:

Given the nonlinearity of the random forest classification model, we are unable to use Standard Errors and a t-test to confirm or reject our hypotheses. However, random forest offers an impurity based feature importance score (MDI[7]) which indicates the importance of a feature in classifying a dataset. However, MDI suffers from being computed on statistics derived from our training set, thus resulting in high importances for features which are not actually predictive of the target variable as long as random forests have the capacity to use them to overfit on the data.[8]

To overcome the biases of *GINI feature importance* (MDI), we decided on using *Permutation Importance*, which computes feature importances by shuffling the observations within the feature so as to simulate random noise. Thus, if it is an informative feature, the score of the model would systematically decrease, otherwise, an uninformative feature would produce a random decrease or increase in score when permuted.[9] In our permutation importance test, we scored the model based on *macro averaged F1*, our main evaluation metric. The results are summarised below:



Weight	Feature
0.1255 ± 0.0711	alcohol
0.0669 ± 0.0530	sulphates
0.0608 ± 0.0474	volatile acidity
0.0389 ± 0.0363	citric acid
0.0322 ± 0.0426	pH_sulphate
0.0232 ± 0.0250	chlorides
0.0187 ± 0.0258	density
0.0154 ± 0.0234	residual sugar
0.0142 ± 0.0261	total sulfur dioxide
0.0135 ± 0.0157	fixed acidity
0.0116 ± 0.0147	total acidity
-0.0006 ± 0.0288	pH
-0.0023 ± 0.0161	free sulfur dioxide

Once again, the features of alcohol, volatile acidity and citric acid are amongst the top five most important features, given by their feature importance (Weight in the table). Thus, this further confirms our earlier assessments of our hypotheses.

We can confirm the statistical significance of our hypotheses by conducting the following hypothesis test which uses the following formula for normalised importance (Z-Score):

$$\frac{VI(\mathbf{x}_j)}{\frac{\hat{\sigma}}{\sqrt{ntree}}} = z_j$$

- ***VI(x$_j$)*** - Feature Importance (Weight)
- $\hat{\sigma}$ - Estimated Standard Deviation of Variable Importance
- ***ntree*** - No. of trees used in our classifier (800)

[7] MDI: Mean Decrease in Impurity, *The Mathematics Behind Decision Trees & Random Forests*, 2018

[8] https://scikit-learn.org/stable/modules/permutation_importance.html

[9] Testing Variable Importance in Random Forests, Strobl & Zeileis, *lifestat*, 2008

Since, each value of VI(xj) is computed using our 800 independent trees which were constructed using bootstrapped samples, thus by Central Limit Theorem, the mean of VI(xj) follows a normal distribution with a standard error of $\sigma/\sqrt{ntree}$. As such, we can test our 3 key hypotheses at the 5% significance level using VI(xj) under the null hypothesis of zero importance. Since $Z_j \sim N(0,1)$, we will use the standard normal distribution table to obtain critical values.

1. Alcohol content in red wines is a statistically significant feature in determining red wine quality.

$$Z_{\text{alcohol}} = 49.925 > Z_{0.05} = 1.645$$

2. Volatile acidity is a statistically significant factor in determining red wine quality.

$$Z_{\text{volatile acidity}} = 36.280 > Z_{0.05} = 1.645$$

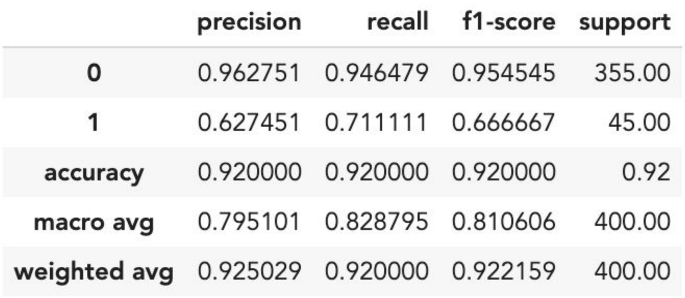
3. Citric acid content is a statistically significant factor in determining red wine quality.

$$Z_{\text{citric acid}} = 30.310 > Z_{0.05} = 1.645$$

Thus, under a one-sided hypothesis test, we know that alcohol, volatile acidity and citric acid have statistically significant positive feature importances and are important in discriminating between Good and Other quality wines. However, we are unable to isolate feature importance by target class, hence we cannot explicitly determine if the importance of any of the above features is specifically significant to the classification of Good Quality & Other Quality red wines.

Finally the negative feature importances (weights) of *free sulfur dioxide* and *pH* indicates that the *macro averaged F1-Score* for our random forest classifier actually increased when these features were removed. Additionally, *free sulfur dioxide* and *pH* both have Z-scores of -4.041 and 0.589 respectively and are thus not statistically significant features in terms of importance. Hence, in order to further optimise our model we will proceed to drop them from our training data and evaluate its results.

Refined Random Forest Classifier & Results



	precision	recall	f1-score	support
0	0.962751	0.946479	0.954545	355.00
1	0.627451	0.711111	0.666667	45.00
accuracy	0.920000	0.920000	0.920000	0.92
macro avg	0.795101	0.828795	0.810606	400.00
weighted avg	0.925029	0.920000	0.922159	400.00

From the results of our refined random forest classifier (without pH and free sulfur dioxide), we observed a comparatively significant increase in almost all evaluation metrics as compared to our previous random forest model. Most notably, the refined model produced a much higher *recall* of 0.711 versus the previous 0.644 and *precision* of 0.627 versus the previous 0.604. Consequently raising its *macro averaged F1-Score* to 0.811 as compared to its previous 0.787. This has a significant benefit to our business problem, as it reduces our false negative rate by 6.6% and false positive rate by 2.3%.

Limitations

The key disadvantage of random forest is in its lack of interpretability. Unlike an individual decision tree, we are unable to easily visualise and interpret all the splitting criterions for each node across all 800 of our trees. Therefore, while robust and powerful, random forest remains largely a black box method that is difficult to tune beyond its available hyperparameters.

8. Evaluation Model: Gaussian Kernel SVM

Intuition & Rationale

The SVM classifier with the Gaussian RBF kernel is simply a weighted linear combination of the kernel function computed between a data point and each of the support vectors. Gaussian kernels have infinite dimensional feature space. The idea is to gain linear separation by mapping the data to a higher dimensional space when the data can't be separated by a linear function, but can be separated by a quadratic one. Through SVM, we aim to maximise the margin of separation around the hyperplane between our 2 classes of wine.

Comparing the overall results, Linear SVM has a better *recall* (0.87 vs 0.51) but Gaussian SVM excels at the other 3 metrics. The higher *macro averaged F1-score* for Gaussian SVM (0.77 vs 0.63) is especially important as it takes into account both *precision* and *recall*, both of which are important metrics in ensuring that our classifier is equally applicable to other types of red wines in the market. Therefore, we have decided to make use of the Gaussian kernel SVM based on its performance metrics.

Moreover, Gaussian RBF kernels tend to yield good performance under general smoothness assumptions and should be considered especially if no additional knowledge of the data is available. Datasets with a large number of features tend to be computationally harder to train using Gaussian RBF kernel with little to no gain in classification accuracy. It is recommended to use Gaussian SVM when the number of features is small (13) and the size of the dataset is intermediate (< 10,000).

Additionally, classifying between "Good" and "Other" quality wines is a nonlinear problem. Based on wine experts' opinions, each individual variable such as Alcohol, Citric Acid has an optimal range. Given the various permutations of features that could lead to "Good" quality wine, it may be hard for a single dimension hyperplane to differentiate between "Good" and "Other" quality wine. Therefore, for the above reasons, we have decided to opt for the Gaussian SVM instead of linear SVM as our evaluation model.

Application & Results

SVM classifiers trained on an imbalanced dataset can produce suboptimal models which are biased towards the majority class and have low performance on the minority class due to the positive instances lying further away from the "ideal" boundary than the negative instances.

Dealing with Class Imbalance

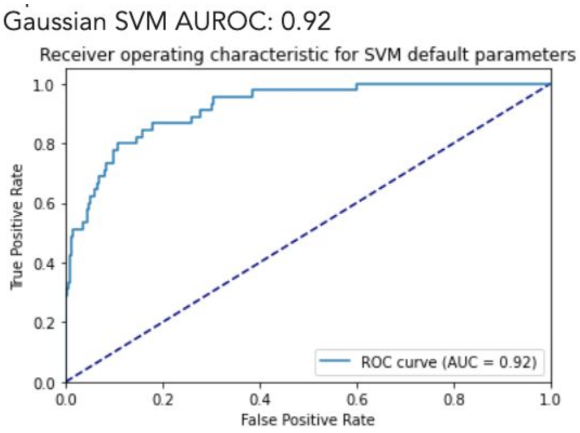
In order to overcome the problem of having an imbalanced dataset, where "Good Quality" wines make up 13.6% of our dataset, we have decided to make use of the *class_weight='balanced'* parameter in the SVM module in the *sklearn* package. This parameter helps to rescale the C parameter to ensure that the size of points is proportional to its weight, which means that the classifier puts more emphasis on accurately predicting the outcome of "Good Quality" wine and this mitigates the effect of reduced sample size of "Good Quality" wines.

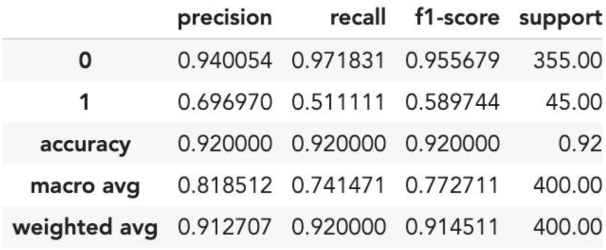
Hyperparameter Tuning

Grid search was used to locate optimal values of C and gamma for our Gaussian SVM model. We obtained optimised parameters as such - {*C*: 1.0, *Gamma*: 0.1}.

C is the penalty parameter, which represents misclassification or error term. It informs the SVM model of the amount of error tolerable. For large values of C, the optimizer will choose a smaller-margin hyperplane if that hyperplane does a better job at correctly classifying our training data. Conversely, a very small value of C will cause the optimizer to look for a larger-margin separating hyperplane, even if that hyperplane causes more misclassification.

Gamma is the influence of each point on the decision boundary. With low gamma, points far away from the plausible separation line are considered in its calculation of the separation line. Conversely, high gamma means the points close to the plausible separation line are considered in its calculation of the separation line.





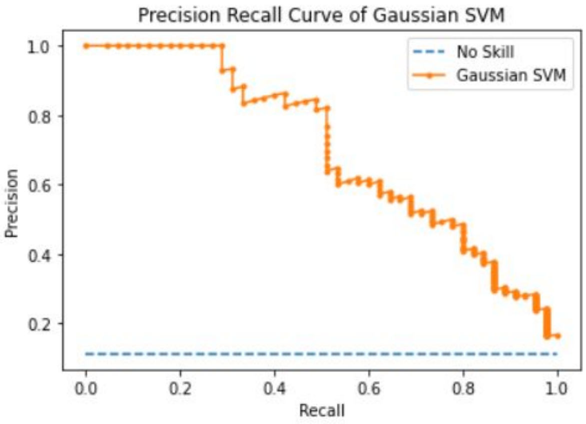
	precision	recall	f1-score	support
0	0.940054	0.971831	0.955679	355.00
1	0.696970	0.511111	0.589744	45.00
accuracy	0.920000	0.920000	0.920000	0.92
macro avg	0.818512	0.741471	0.772711	400.00
weighted avg	0.912707	0.920000	0.914511	400.00

Explanation of Results

Looking at the results obtained using Gaussian SVM, we can observe that its *recall* score for "*Good Quality"* red wines is average. To potentially improve on this, we could increase the value of *C* in our model, which would make the hyperplane between points more defined, but sacrifices on the generalisability of the model. As mentioned earlier, the lower recall score could be attributed to the fact that the data points are less distinctly separated in a Gaussian RBF SVM as compared to the Linear SVM. As a result, the model may inaccurately predict and miss out on some "Good Quality" wines.

We have chosen the *Macro-Averaged F1-score* to be a gauge of our model's suitability as it takes both false positives and false negatives into account. This is especially useful for dealing with unbalanced datasets like ours. The Gaussian SVM classifier is able to find a proper separating plane between the 2 classes of wine, thus it is able to strike a balance between *precision* and *recall*, leading to a higher *Micro-Averaged F1-Score* of 0.773.

The *AUROC* score of Gaussian SVM (0.92) is higher than the *AUROC* score for Linear SVM (0.87). It is equivalent to the probability that a randomly chosen "Good Quality" sample is ranked higher than a randomly chosen "Other Quality' wine. The *AUROC* score for both SVM models is already considered high, but the higher score for Gaussian SVM indicates that a higher dimensional hyperplane is more suitable for separating the "Good Quality" wines.



The results obtained for the precision recall curve is similar to what we have obtained for the Random Forests method where it achieves a good balance between precision and recall and consequently produces the best macro averaged *F1-Score.* Hence, it is suitable for our Good Quality red wine business problem.

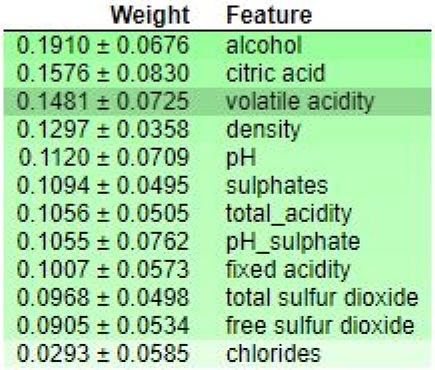
K-fold cross validation is a strong indicator of our model's generalisability. The dataset is split into K subsets and multiple models are trained on these subsets. This ensures that the results that we have obtained are generalisable to most red wines.

F1	**F1 Macro**	**Precision**	**Recall (Good Quality)**
0.528	0.732	0.676	0.441

From the results obtained from *K-fold cross validation* with 10 folds, it seems that our Gaussian kernel SVM has low variance due to the similarity of the *F1 macro score* and *precision* score when compared to our original results from test data. Research has shown that the complexity of hypothesis space defined by Gaussian kernels is inversely proportional to the variance.

If the function that determines the margin of separation class is very simple (high bias, low variance), it will not be able to change rapidly to classify the different quality of red wines. On the other hand, if the function class is very complex (low bias, high variance), it can rapidly change in the small region, fitting the training data accurately, but this does not bode well for applying this model to other datasets due to the possibility of overfitting.

Hypothesis Testing & Feature Importance



Weight	Feature
0.1910 ± 0.0676	alcohol
0.1576 ± 0.0830	citric acid
0.1481 ± 0.0725	volatile acidity
0.1297 ± 0.0358	density
0.1120 ± 0.0709	pH
0.1094 ± 0.0495	sulphates
0.1056 ± 0.0505	total_acidity
0.1055 ± 0.0762	pH_sulphate
0.1007 ± 0.0573	fixed acidity
0.0968 ± 0.0498	total sulfur dioxide
0.0905 ± 0.0534	free sulfur dioxide
0.0293 ± 0.0585	chlorides

The permutation importance scores were larger than 0 for all 3 of the variables in our hypotheses (*citric acid, alcohol and volatile acidity*). This suggests that all 3 features are important in determining the SVM's model's classification power and helps it to classify "*Good*" vs "*Other*" quality wines. The weight of the metric indicates that dropping *citric acid, alcohol* and *volatile acidity* individually results in the biggest drop in the predictive power of the model, which proves that they are significant factors in determining the quality of wine.

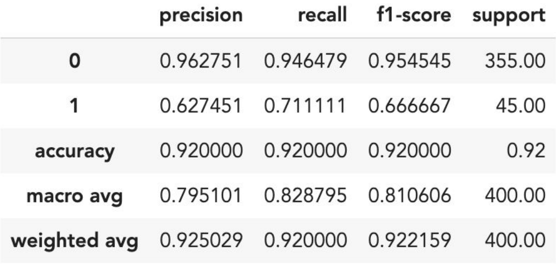
It was interesting to note that when we ran the permutation importance for Gaussian SVM, none of the features were dropped, indicating that all of our features used are important. This was consistent with literature review suggesting that permutation importance is more effective on Random Forests as compared to Gaussian SVM. This is backed up by our results, where running the permutation importance on this model did not affect it significantly.

Limitations

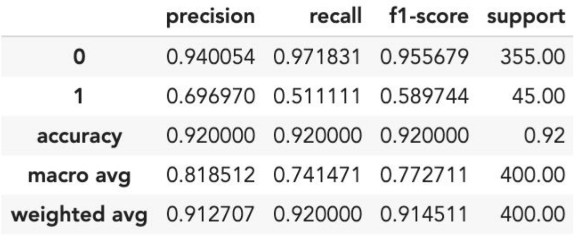
A common disadvantage of non-parametric techniques such as SVMs is the lack of transparency of results. SVMs cannot represent the score of all features as a simple parametric function of wine quality, since its dimension may be very high. The lack of transparency for SVM also prevents us from including business logic in feature engineering since we are unable to exactly know how it will affect the model. Additionally, it is a non-probabilistic method, thus we are unable to know if the probability that each classification is correct. Finally, without adequate scientific knowledge of how the different proportions of chemicals in the wine interact with one another, it may be tricky to find the appropriate kernel to use for SVM.

9. Comparing the Main & Evaluation Models

Test Results



	precision	recall	f1-score	support
0	0.962751	0.946479	0.954545	355.00
1	0.627451	0.711111	0.666667	45.00
accuracy	0.920000	0.920000	0.920000	0.92
macro avg	0.795101	0.828795	0.810606	400.00
weighted avg	0.925029	0.920000	0.922159	400.00



	precision	recall	f1-score	support
0	0.940054	0.971831	0.955679	355.00
1	0.696970	0.511111	0.589744	45.00
accuracy	0.920000	0.920000	0.920000	0.92
macro avg	0.818512	0.741471	0.772711	400.00
weighted avg	0.912707	0.920000	0.914511	400.00

Based on our comparison of the results above, our selection of Random Forest as our main model is justified given its superior performance in most of our evaluation metrics (i.e. *recall, macro averaged F1*). Thus, the Random Forest classifier is better suited to our current *red wine quality* dataset and our business needs of accurately identifying as many *Good Quality* red wines without having too many false positives amongst them.

Cross Validation Scores & Time Complexity

To evaluate and compare the generalisability of our main and evaluation models, we fitted both models on the red wine data using *K-fold cross validation* with 10 folds. The results are as follows:

Evaluation Metric	**Random Forest**	**Gaussian SVM**
F1	0.473	0.528
F1 Macro	0.700	0.732
Precision	0.625	0.676
Recall (Good Quality)	0.436	0.441
Time Complexity	O(n^2)	O(n^3)

Based on the results, it is clear that Gaussian SVM performs better in all key evaluation metrics and is thus more generalizable than Random Forest. Therefore, we have concluded that Random Forest would be better applied to an existing repository of red wine data that contains data from either a specific vineyard or family of red wines, given its smaller time complexity. On the other hand, SVM Gaussian RBF would be more useful in classifying data from a wider range of sources, thus it could be applied to multi market analysis or the classification of a diverse set of red wine samples since it will already be pre-trained on a larger existing dataset.

10. Areas for Improvement

The most pivotal area for improvement would be to expand our dataset, this will help us mitigate existing class imbalance and help our models grow in predictive power with more information. Another area for improvement is perhaps collecting a series of wine data over time, as tastes and preferences change over time and hence, a constantly updated model that is able to predict wine quality over time would be extremely useful for wine companies.

11. Conclusion

Addressing the Hypotheses

We began our investigation with the idea of building an effective red wine quality classifier. To achieve this, we first used ANOVA to determine statistically significant features that would inform our model. After which, we coupled these results with domain knowledge sourced from red wine experts in order to come up with 3 key hypotheses. These hypotheses were later used to confirm the validity of our assumptions about specific red wine characteristics and validate the performances of our models. From which, we concluded that the features used in our hypotheses were statistically significant factors in determining red wine quality. This conclusion was made using *Permutation Importance* and their computed Z-scores. However, we were unable to conclude if there is a direct relationship between these features and "*Good Quality*" red wines. Nonetheless, we managed to fulfil our project aim of building a predictive model to determine the quality of red wines.

Potential Business Applications

The more generalizable Gaussian SVM model will be useful for a client that has a global market. This is because our model will be better at handling the diverse tastes and preferences of the global market. This model will form more robust predictions that will be better generalised to the client's global market. Our Random Forest model will be used for the client that has a localized market, concentrated in a specific country or a few regions. The model can then provide predictions that will be less generalizable, but more impactful for this type of client.

With our chosen final models, our clients can input wine samples and their characteristics from their current or new wine products. They can then get a sensing of how well their wine products will do as our models classify them as either "*Good Quality*" or "*Other Quality*". With this information, our clients would be able to better focus their advertising or production efforts. For example, they could expand production and market the "*Good Quality*" wines with greater intensity. At the same time, they could use our models' assistance to adjust the significant features (which we have identified through the hypotheses that have been accepted) to create a new wine product that will perform better in that market. Our models present a quicker, cheaper and more accurate method for wine companies to predict the wine quality.