

# From Orchard to Vineyard: Evaluating Machine Learning Algorithms for Apple and Wine Quality Prediction

Marcus Tan  
mtan75@gatech.edu

**Abstract** — This paper delves into the academic discourse surrounding the evaluation of five machine learning algorithms across two distinct datasets. The paper encompasses a thorough examination of each dataset, including its inherent characteristics, unique features, and formulated hypotheses. Subsequently, the study explores the performance of the datasets under various algorithms, leading to a comparative analysis of these methodologies.

## 1 INTRODUCTION

In recent years, there has been a notable surge in the acknowledgment and utilization of machine learning and deep learning methods, paralleling the advancements in computational capabilities (Smith, Johnson, & Williams, 2020). This paper seeks to contribute to this evolving field by conducting research on five specific algorithms (Decision Trees, Boosted Trees, Neural Networks, K-Nearest Neighbor, Support Vector Machine) applied to two datasets, each varying in three distinct characteristics (1. Collinearity, 2. Outliers, 3. Imbalanced Multi-Class vs Balanced Binary Class).

This paper unfolds in three sections: (1) an examination of the dataset, encompassing its nature, peculiarities, and associated hypotheses; (2), an exploration of the dataset's performance across diverse algorithms; and (3), a comparative analysis of these algorithms, and concluding on whether the hypothesis is accurate.

## 2 DATA

### 2.1 Dataset Introduction

**[Dataset #1: Wine Quality]** The first dataset used is the "[winequality-white.csv](#)" (Cortez, Cerdeira, Almeida, Matos, & Reis, 2009) dataset which contains information about white wines, specifically their chemical properties and quality ratings. Each row in the dataset represents a different white wine, and the columns include various attributes such as acidity levels, residual sugar, alcohol content, and quality.

**[Dataset #2: Apple Quality]** The second dataset is the "[apple quality.csv](#)" (Elgiriye withana, 2023) dataset which is a more recent dataset on Kaggle that contains different characteristics of an apple such as the size, weight, sweetness in order to determine whether an apple is a good or bad quality.

### 2.2 Interesting Characteristics of the 2 Datasets

Qualitatively, this dataset was selected because of a genuine fascination with food, particularly intrigued by the opportunity to explore how produce from orchards and vineyards manifests distinct characteristics defining its quality. The overlap of certain columns, such as acidity and the target variable - quality, in both datasets adds a unique qualitative layer to the investigation.

Quantitatively, several intriguing characteristics have surfaced, providing valuable insights for our algorithm analysis:

**[Correlation Patterns]** In the wine quality dataset, noticeable collinearity exists between variables. For instance, a strong positive correlation between density and residual sugar (Pearson correlation: 0.83), while density exhibits a significant negative correlation with alcohol (-0.78). In contrast, the apple quality dataset displays fewer correlated variables, with the most notable correlation between sweetness and size (Pearson correlation: -0.32).

**[Outliers]** Wine quality exhibits a higher prevalence of outliers compared to the apple quality dataset. Notably, columns such as chlorides demonstrate a pronounced outlier condition, affecting 1-3% of the dataset. Outliers are defined as data points falling outside the 1.5 Interquartile Range. Conversely, the majority of features in the apple quality dataset lie within this range, with less than 1% of data points identified as outliers across all columns.

**[Imbalance and Class Distribution]** Wine quality is characterized by a high degree of imbalance due to its multi-class nature, contrasting with the balanced binary classification in the apple quality dataset. Notably, the multi-class distribution in wine quality is uneven, with 92% of data points falling into quality levels 5, 6, or 7 & remaining 8% is split between classes 3, 4, 8, 9. The exploration will delve into understanding how algorithms respond to this imbalanced distribution.

**[Similarities for Controlled Analysis]** To mitigate potential confounding effects, it's crucial to highlight certain similarities in the analysis. Both datasets share similar dimensions, with approximately 4,000 rows and 8-11 features. Moreover, standardization has been applied to ensure uniform weightage across variables in subsequent modeling. Models are also applied Stratified K-Fold (5) to obtain the results on the training/validation sets. Stratified K-fold has been used to ensure that imbalanced classes from wine quality will have an equal chance of appearing in all the different folds of the dataset.

### **2.3 Hypothesis – How do we define best and which algorithm is hypothesized to perform the best?**

While various metrics could define a "best" algorithm, such as efficiency or accuracy or wall clock times, in this case, we define "best" as the algorithm that performs optimally in terms of recall. Given the imbalanced nature of the wine quality dataset and the uneven distribution of multi-class data points, recall is chosen as the metric to emphasize the model's ability to correctly identify instances of different quality levels.

Given that neural networks are inherently able to capture intricate patterns and non-linear relationships within complex datasets, this report hypothesize that neural networks will outperform decision trees, boosted decision trees, support vector machines (SVM), and k-nearest neighbors (KNN) in this analysis due to their capacity for handling data plagued with outliers and imbalanced multiclass issues. This report hypothesizes that neural networks, are well-suited to discern the nuanced and multi-faceted characteristics inherent in the wine and apple quality datasets.

### 3 PERFORMANCE OF MODELS

#### 3.1 Decision Trees

To assess the efficacy of decision trees in managing datasets characterized by imbalanced multi-class, outliers, and multicollinearity, we will experiment with different hyperparameters, specifically adjusting variables like (1) the maximum depth of the tree and (2) the maximum number of leaf nodes.

	Wine Quality	Apple Quality
Max Depth		
Max Number of Nodes		
Learning Curve	<p>Note: Grid Search optimal (depth: 17, leaf_nodes: 142)</p>	<p>Note: Grid Search optimal (depth: 12, leaf_nodes: 152)</p>

**[Observation]** From the above chart, there are 2 key observations: (1) max depth and max number of nodes tend to plateau at higher depth and higher number of tree nodes and (2) learning curves indicate that apple quality requires larger number of dataset in order to achieve higher accuracy.

**[Analysis]** Upon reviewing scikit-learns documentations that uses CART algorithm (based off Quinlan’s C4.5 model), the observations can be attributed to the model complexity limitations of decision trees. Decision trees rely on a greedy approach to maximize information gain at each node, risk suboptimal splits by prioritizing local optimization over global data structure. Additionally, these models encounter limitations in capturing complex relationships, where increased node or tree depth may yield minimal performance gains, resulting in the reason for plateaus observed in the decision tree diagrams above.

### 3.2 Boosting

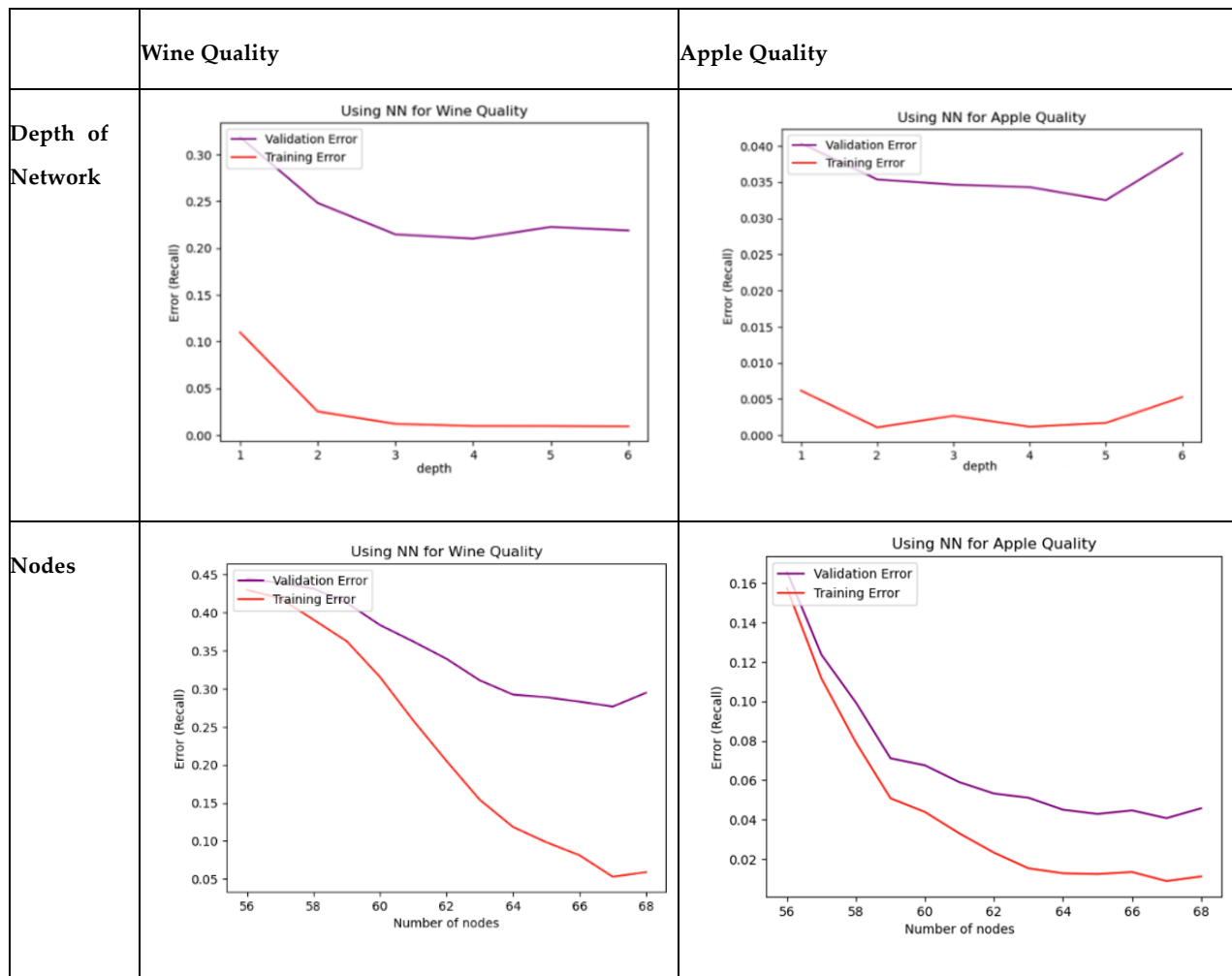
For boosting, this paper relies on Adaboost to combine multiple “weak learners” to obtain a more effective model. The 2 hyperparameters shown in the validation curve below include: (1) number of trees (or number of weak learners), and (2) max depth of trees.

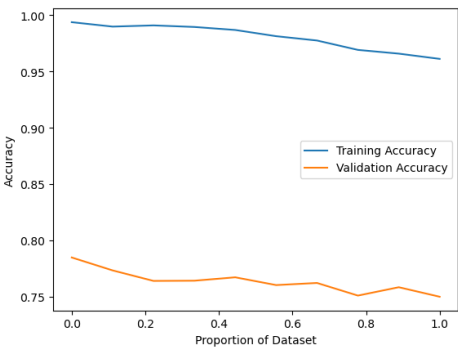
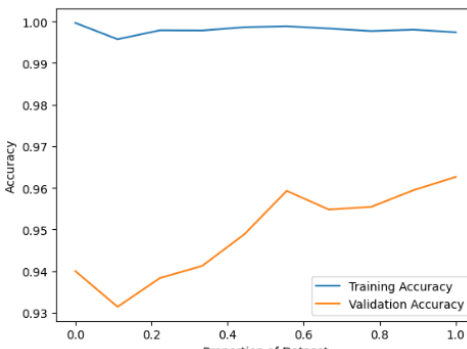
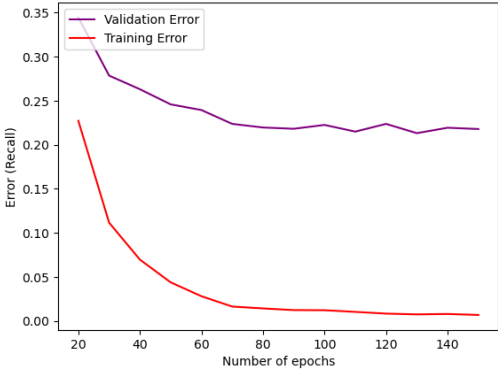

	Wine Quality	Apple Quality
Number of Trees		<p>Note: Validation error appears to be a straight line but there are in fact some minimal changes in error in validation curve.</p>
Max Depth of Trees		
Learning Curve	<p>Note: Grid Search optimal (depth: 10, n_estimator: 850)</p>	<p>Note: Grid Search optimal (depth: 6, n_estimator: 15)</p>

**[Observations]** In the above diagrams, we observe 3 key takeaways: (1) Wine quality requires more trees (~600 trees) before the incremental error plateaus, (2) Minimal overfitting is seen in both the datasets and (3) both dataset requires around similar number of training (1000 – 1500) examples before validation score plateaus.

**[Analysis]** Given the characteristics of wine quality (collinearity, outliers, imbalanced multiclass), it is reasonable to expect that a higher number of trees is necessary when implementing AdaBoost for wine quality as compared to apple quality, particularly due to its sensitivity to noise and outliers. This stems from AdaBoost's utilization of an exponential loss function (Tanha, Abdi, Samadi, Razzaghi, & Mohammad, 2020). As the algorithm progresses, it focuses more on the challenging data points, allowing it to adapt and learn from mistakes without excessively fitting to the noise. Additionally, as observed in the charts above, AdaBoost is also less susceptible to overfitting as it places higher emphasis on instances that are misclassified by the previous weak learners in the ensemble, resulting in the overfitting observations seen in the graphs above.

### 3.3 Neural Networks



	Wine Quality	Apple Quality
Learning Curve	 <p>Note: Grid Search optimal parameters (layers: 64 nodes x 3 layers, epochs: 150, Optimizer: Adam(LR: 0.01), Sigmoid Activation)</p>	 <p>Note: Grid Search optimal parameters (layers: 64 nodes x 3 layers, epochs: 100, Optimizer: Adam (LR: 0.01), Sigmoid Activation)</p>
Epoch	 <p>Using NN for Wine Quality</p>	 <p>Using NN for Apple Quality</p>

**[Observation]** There are 2 notable observations from the above table: (1) neural networks tend to overfit at higher nodes and depth and overfit at approximately the same number of node/depth, (2) both datasets tend to see better results with the Adam optimizer.

**[Analysis]** The 2 observations seen above are within expectations. At higher nodes and depths, neural networks, especially those with many parameters (nodes and weights), can memorize the training data rather than learning general patterns. This can lead to poor generalization to unseen data, resulting to overfitting as we've seen above with higher nodes and depth.

Additionally, Adam optimizers is often favored and performs well in practice due to its adaptive learning rates and momentum-like behavior. It is suitable for a wide range of tasks and is less sensitive to the choice of hyperparameters. Adam dynamically adjusts the learning rates for each parameter during training. It combines ideas from RMSprop and Momentum, using moving averages of both the gradients and their squared values, resulting in the reason why Adam optimizers tend to escape from local optima and provide better results.

### 3.4 K Nearest Neighbor

In K-Nearest Neighbor, we will look at varying 2 key hyperparameters: (1) the number of neighbors and (2) the distance metric selected.

	Wine Quality	Apple Quality
<b>K – Number of Neighbors with Euclidean Metric</b>		
<b>Distance Metric (Manhattan)</b>		
<b>Learning Curve</b>	<p>Note: Grid Search optimal (weight: distance, metric: manhattan, n_neighbor = 25 ,algorithm = kd_tree)</p>	<p>Note: Grid Search optimal (weight: distance, metric: euclidean, n_neighbor = 4 ,algorithm = kd_tree)</p>

**[Observation]** Three notable observations emerged during the design of the validation and learning curves. (1) it is evident that wine quality is more susceptible to overfitting compared to apple quality, (2) the optimal value for K (a parameter, possibly related to neighbors in K-nearest neighbor algorithms) for apple quality is lower than that for wine quality (3), the learning curves for the training dataset approach a value close to 1, indicating a high level of performance.

**[Analysis]** According to Muhr et al., the observations align with the typical K-nearest neighbors (KNN) equation:

$$KNN(x) = \frac{1}{k} \sum_{i=1}^k d^{(i)}(x, X), \text{ where } x \in X \text{ and } d^{(i)}(x, X) \text{ is the distance between } x \text{ and its } i\text{th nearest neighbor in } X.$$

Per the equation above, when dealing with outliers such as in the case of wine quality, a larger number of neighbors becomes essential to mitigate this impact (Wine Quality: 25 neighbors vs Apple quality 4 neighbors). Notably, the equation highlights the significance of the distance metric in predicting the target variable, where closer datapoints contribute higher weightage to the prediction (Muhr, Affenzeller, & Kung, 2023).

### 3.5 SVM

To evaluate SVM on the 2 datasets, this paper varied the following hyper parameters: (1) type of Kernels (Polynomial and RBF) and (2) different degrees for polynomial kernels.

	Wine Quality	Apple Quality
Polynomial Kernel and different degrees		
RBF Kernel with Varying C		
Learning Curve	<p>Note: Grid Search optimal (kernel: rbf, c (regularization parameter): 2)</p>	<p>Note: Grid Search optimal (kernel: rbf, c (regularization parameter): 4)</p>



**[Observations]** Several observations can be derived from the charts above: (1) using RBF Kernels and varying the regularization parameter “c” can create validation curves with lower error rates, (2) using polynomial kernels at higher degree results in overfitting, (3) learning curves for SVM at optimized functions hint at the need for more data to generate better predictions.

**[Analysis]** Generally, the reasons why RBF kernels perform better than polynomial kernels (such as in the case highlighted here) is due to the nature of their influence decay with behavior (similar to K nearest neighbor, with distance as the weightage!). This can be illustrated through the formula for RBF kernels:

$K(x, y) = e^{\frac{(x-y)^2}{2\sigma}}$  where x and y are input vectors representing datapoints in the features space and  $\sigma$  is the parameter that controls the kernel function. As the distance between the datapoints widens, the result approaches zero and RBF kernels give lesser weights to datapoints further away. This helps to capture complex decision boundaries such as those in the dataset where polynomial curves are not able to accomplish.

## 4 SYNTHESIS – WHICH MODEL PERFORMED BEST? WHY?

### 4.1 Best Model Without Transformations

In summary, the utilization of grid search with cross-validation allowed us to pinpoint optimized parameters for each of the 5 models that were employed in the earlier learning curves. Each set of these refined parameters was then input into the respective models, yielding the outcomes presented below, including recall scores and corresponding wall clock times.

	WINE QUALITY VALIDATION SET – RECALL SCORE	WINE QUALITY WALL CLOCK TIMES (TRAIN AND TEST)	APPLE QUALITY VALIDATION SET – RECALL SCORE	APPLE QUALITY WALL CLOCK TIMES (TRAIN AND TEST)
DECISION TREES	0.535	0.22 seconds	0.811	0.27s seconds
ADABOOST	0.657	122.17 seconds	0.89	66.86 seconds
NEURAL NETWORK	0.76	88 seconds	0.97	42 seconds
KNN	0.646	1.6 seconds	0.90	0.2 seconds
SVM	0.566	5.58 seconds	0.89	1.39 seconds

Judging purely based on the recall score of the validation set’s recall score, Neural Network model performed the best for Wine Quality and Apple Quality dataset. There are primarily 3 reasons why the dataset performed better with Neural Networks.

**[Reason 1: Adaptability to Imbalanced Distributions]** Neural networks can be configured to handle imbalanced class distributions effectively. Neural networks classify highly imbalanced by considering the unit gradient direction of positive and negative classes (Huang, Sang, Sun, & Lv, 2022). Additionally, techniques like class weighting or specialized loss functions in NNs enable them to focus on the minority class, improving their performance in imbalanced multi-class scenarios.

**[Reason 2: Capacity to learn from Collinear Features]** Neural networks are also capable of working with collinear features because they rely on non-linear activation function and allow for regularization through dropouts. This adaptability is beneficial when dealing with collinear features, such as in the case of wine quality.

**[Reason 3: Robustness to Outliers]** Neural networks are generally robust to outliers, especially when trained on large datasets with outliers less than 15% (Khamis, Ismail, Haron, & Mohammed, 2005). The non-linear activation functions and the ability to learn from a diverse range of examples help NNs handle outliers more effectively compared to some other algorithms, like SVM or KNN.

#### 4.2 Best Model – Balancing Datasets

To further our analysis, we further investigated how nullifying characteristics in the wine quality dataset – such as (1) reducing outliers and (2) transforming target variable into binary classes – impacts algorithmic performance. We aimed to determine if these changes yield similar, better, or worse results and identify which algorithm benefits most from the data transformations.

	DECISION TREES	ADABOOST	NEURAL NETWORK	KNN	SVM
WINE QUALITY VALIDATION SET – RECALL SCORE	0.71	0.78	0.86	0.79	0.74
IMPROVEMENT FROM PREVIOUS OPTIMAL MULTI CLASS RECALL SCORE	$0.71 - 0.535 = 0.175$	$0.78 - 0.657 = 0.123$	$0.86 - 0.76 = 0.1$	$0.79 - 0.646 = 0.144$	$0.74 - 0.566 = 0.174$

**[Observations]** As observed in the above table, we observe that most of the algorithms improved by at least 10%. In some algorithms such as Decision Trees, where the initial performance on multiclass is not performing as well, we see larger improvements of up to 17.5%.

**[Analysis]** These findings are expected, as it is commonly acknowledged that accurately assigning a new instance to one of multiple classes becomes more challenging with an increasing number of classes. The underlying challenge lies in the notion that employing the learning algorithm on each decision boundary separately is more effective than applying the same learning algorithm concurrently to multiple decision boundaries (Del Moral, Nowaczyk, & Pashami, 2023). By converting the problem into a binary classification problem, we will simplify the decision boundary for the algorithm to identify, which makes it easier for us to get higher recall score as we have seen in the table above.

## 5 CONCLUSION

In summary, this report explored a classification problem across two datasets differing in collinearity, outliers, and imbalanced multi-class scenarios. Neural networks emerged as the most effective algorithm, showcasing adaptability to imbalanced distributions, capacity to handle collinear features, and robustness to outliers. Additionally, the transformation of a multiclass dataset into binary classes improved the performance of all algorithms in binary classification. While the focus was on recall as a metric, the report would also like to emphasize the importance of considering factors such as explainability and training time, urging data scientists to make informed trade-offs when choosing an appropriate model for their specific use case.