Wine Classification - Supervised Learning Approach

Kenny Bartel (CS 5033) [1] **Tanner Benbrook (CS 4033)** [2]

1. Project Domain

1.1. Overview

The semester project, focused on Supervised Learning (SL), aims to develop a classification system to categorize wines into three quality groups—Low ([0,5)), Medium ([5,7)), and High ([7,10])—based on a numerical quality scale from 0.0 to 10.0. Our team hypothesizes that supervised learning techniques can train a model to accurately classify wine quality using the UCI Wine Quality dataset, targeting an overall accuracy of 75%. The project seeks to compare multiple SL algorithms to achieve reliable and insightful results, utilizing training and test datasets for learning and evaluation.

1.2. Approach

The project addresses predicting wine quality based on chemical properties, such as alcohol level, acidity, and wine color, using the UCI Wine Quality dataset, which includes 6,497 samples with 11 features and quality scores from 0 to 10. We were tasked with grouping these scores into three categorical bins: Low (0 to <5), Medium (5 to <7), and High (7 to 10). Accurate classification offers practical benefits for quality control in wine production and consumer decision-making. The challenge lies in mapping numerical features to qualitative categories. The team will implement supervised learning algorithms, including Decision Trees, k-Nearest Neighbors (k-NN), Gaussian Naive Bayes, and Random Forest. The UCI Wine dataset will be divided into training (80%) and testing (20%) sets. The performance of these algorithms will be compared, and our results will be benchmarked against existing implementations to determine the most effective approach.

2. Hypotheses

Our hypotheses are the following listed below:

1. Naive Bayes will perform the worst of all classification algorithms.

2. All models will exceed an accuracy of 75%.

3. Decision Trees will correctly classify wines with medium quality with higher accuracy than wines with low or high quality, due to the imbalanced dataset containing mostly medium quality wine scores.

4. A Weighted K-NN approach would significantly outperform an unweighted K-NN approach

3. Experiments

3.1. Hypothesis 1

Hypothesis 1 posited that Naive-Bayes would perform the worst of all algorithms. The accuracy of each implementation is shown below.

Algorithm	Min	Average	Max
Naive-Bayes	61.38%	67.77%	63.39%
Random-Forest	79.08%	81.10%	83.08%
Weighted 7-NN	76.85%	77.85%	78.60%
Decision Tree	68.34%	70.80%	71.93%

Table 1. Minimum, maximum, and average accuracies of each SL model based on our wine dataset.

Based on the data from Table 1, we can see that Naive-Bayes had the worst performance. This under-performance is due to several factors. First, the class imbalance in the dataset caused the model to over-predict the ”low” class, as it dominates the data. Second, the Naive Bayes assumption of feature independence was violated because features like ”free sulfur dioxide” and ”total sulfur dioxide” are highly correlated. Third, some features are heavily skewed and don’t follow a Gaussian distribution, which further impacts the model’s effectiveness.

3.2. Hypothesis 2

Based on the results in Table 1, we can see that Naive-Bayes did not reach the 75% threshold. Naive Bayes fell short of the 75% accuracy threshold due to the dataset's class imbalance, correlated features, non-Gaussian feature distributions, and lack of stratification in the train-test splits. These issues violated the model's core assumptions, leading to an average accuracy of 67.77%.

3.3. Hypothesis 3

For this hypothesis, we developed a confusion matrix to determine which classification that our decision tree model struggles the most with. This was done by calculating the true label of each data instance in our test data, as well as the corresponding predicted label from the decision tree classification.

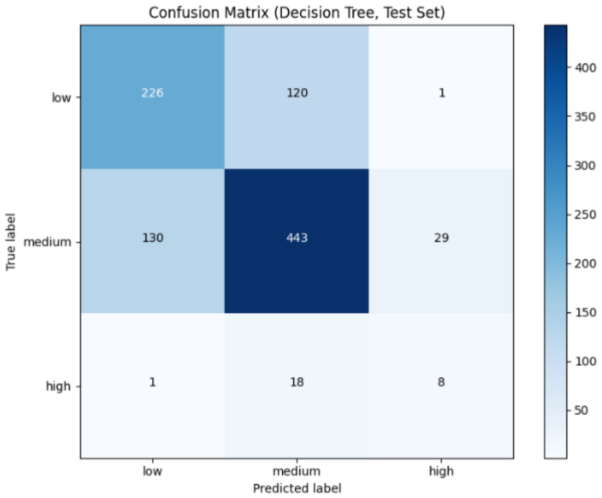

Figure 1. Confusion Matrix of true labels vs predicted labels for each class

Based on figure 1, we can clearly see the data imbalance of the medium class having the most instances, followed by the low class, and the high class having a significantly lower amount of instances. As expected given this information, wines that fall into the medium class are classified with the highest percentage of all classes by our decision tree model. Given this fact, we conclude that hypothesis 3 was correct. The accuracy of predictions for each class can be seen in the table below.

Algorithm	Low	Medium	High
Decision Tree	65.13%	73.59%	29.63%

Table 2. Percentage of correctly identified instances for each class in the Decision Tree model based on the confusion matrix from the test set.

3.4. Hypothesis 4

Hypothesis 4 posited that a weighted K-NN approach would significantly outperform an unweighted K-NN approach. To facilitate this comparison, the experiment utilized a tuned parameter $k = 7$, with 80% of the dataset allocated for training and 20% for testing. To determine these values, our experiment began with a 1-NN approach, subsequently incrementing k by 2 for each iteration, up to 13-NN. This tuning process revealed that 7-NN yielded the optimal results for the wine dataset, as both the smallest and largest k values within the range of odd numbers [1,13] produced noticeably inferior outcomes compared to 7-NN. To ensure each run featured a distinct split for training and testing data, our experiment employed NumPy's `np.random.seed()` and `np.random.permutation()` functions. As evidenced Table 3 shown below, the initial hypothesis was confirmed by the collected data

Algorithm	Min	Max	Average
Weighted 7-NN	76.85%	80.08%	78.60%
Unweighted 7-NN	69.85%	73.69%	71.73%

Table 3. Minimum, maximum, and average wine classification percentages of Weighted 7-NN vs Unweighted 7-NN for the UCI Wine Quality Dataset over 10 runs per algorithm

4. Literature Review

4.1. Naive-Bayes

McCallum and Nigam (1998) analyzed Naive-Bayes' application to imbalanced datasets in a seminal paper published in the *AAAI-98 Workshop on Learning for Text Categorization*. They proposed adjusting class priors and handling feature dependencies to improve Naive Bayes performance, achieving competitive results in text classification with unbalanced classes. In my wine quality experiments, Naive Bayes achieved an average accuracy of 67.77% falling short of the 75% target. This aligns with McCallum and Nigam's findings that Naive-Bayes struggles with imbalanced data. Their suggestion to adjust priors could improve the model's performance by better balancing the class predictions.

4.2. Random Forest

Breiman (2001) introduced the algorithm in a foundational paper published in *Machine Learning*, demonstrating its robustness to noise and ability to handle correlated features in classification tasks. Breiman showed that Random Forests reduce overfitting by combining decision trees with bagging and feature randomness, achieving high accuracy in complex datasets. In my experiments, Random Forest achieved an average accuracy of 81.10%. This aligns with Breiman's findings, as Random Forest effectively handled the dataset's correlated features and non-linear relationships.

4.3. Comparative Analysis of Decision Tree Classification Algorithms

Priyam et al. (2013) conducted a comparative analysis of decision tree algorithms (ID3, C4.5, CART, SLIQ, SPRINT) in the context of educational data mining, focusing on predicting student performance. Published in the International Journal of Current Engineering and Technology, the study found that C4.5 outperforms others on small datasets due to its pruning and ability to handle continuous attributes, while SPRINT excels for large datasets due to its scalability. In this project, we implemented a custom decision tree classifier, closely resembling CART. This implementation uses binary splitting and information gain (via entropy) to select splits, as seen in a typical CART approach, but lacks C4.5s gain ratio or SPRINTs scalability features. Also, our model struggled significantly with the "high" quality class (29.63% accuracy, Table 2), reflecting Priyam et al.'s observation that class imbalance challenges decision trees, supporting Hypothesis 3 that Decision Trees perform better on the majority "medium" class.

4.4. Class Confidence Weighted kNN Algorithms for Imbalanced Data Sets

Liu and Chawla (2011) introduced Class Confidence Weighted (CCW) k-NN algorithms to address class imbalance, published in the Proceedings of the 15th European Conference on Principles of Data Mining and Knowledge Discovery (ECML PKDD). They propose weighting k-NN prototypes using the likelihood of attribute values given class labels $(p(xi|yi))$, transforming the decision rule from prior to posterior probabilities. Experiments on imbalanced datasets showed CCW-kNN outperforming traditional methods, achieving an AUC-PR of 0.035 on the Appetency dataset (k = 1) compared to 0.022 for unweighted kNN. In our project, Weighted 7-NN, using inverse distance weighting, achieved 78.60% average accuracy versus 71.73% for unweighted k-NN (Table 3), supporting Hypothesis 4. However, Liu and Chawla highlight that inverse distance weighting is less effective in dense, imbalanced regions, as seen in the UCI Wine datasets minority "high" class. Applying CCW could further improve performance by correcting this bias, aligning with future work to optimize weighting strategies.

4.5. A Classification Approach with Different Feature Sets to Predict the Quality of Different Types of Wine using Machine Learning Techniques

Aich et al. (2019) explored wine quality prediction on the UCI Wine Quality dataset using decision tree-based classifiers, including RPART (CART), with feature selection via PCA and RFE. Presented at the International Conference on Advanced Communications Technology (ICACT), their RPART model achieved an accuracy of approximately 76% (inferred from figures), while Random Forest with RFE reached 94.51% for red wine and 97.79% for white wine. In our project, a custom CART-like decision tree classifier achieved an average accuracy of 70.80% (Table 2), comparable to Aich et al.'s RPART performance, slightly underperforming in comparison. However, both models struggled with the "high" quality class due to class imbalance, supporting Hypothesis 3 that Decision Trees perform better on the majority "medium" class. Aich et al.'s use of RFE suggests a potential improvement for this project, aligning with our potential future work to address class imbalance.

5. Novelty

Given the significant imbalance of the UCI wine dataset used (especially for the "high quality" class), we decided to focus on attempting to fix this issue. For this, we introduced new data that was specifically under this high quality class. Once we generated this new data (150 instances), and combined the data with the existing UCI data, the performance of our existing weighted 7-NN and Decision Tree implementations was benchmarked a second time. This was done in the hope that performance would increase as our classifiers will become more familiar with the high quality wines through training. In addition, benchmarking was done over the course of ten runs for each classifier. The updated confusion matrix for our decision tree implementation, as well as a table of performance for both classification models given the new dataset can be seen below.

Algorithm	Min	Max	Average
Weighted 7-NN	78.63%	81.47%	80.07%
Decision Trees	70.46%	73.08%	71.92%

Table 4. Minimum, maximum, and average wine classification percentages of Weighted 7-NN and Decision Trees for the UCI Wine Quality Dataset over 10 runs per algorithm (with new data instances with "high quality" scores introduced)

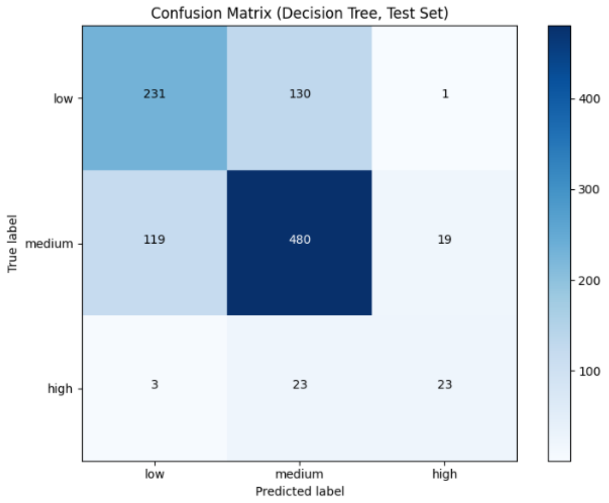

Figure 2. Confusion Matrix of true labels vs predicted labels for each class given new data

By comparing our newly generated table 4 with tables 3 and 2 done on the original dataset, we can see some minor improvements to both classification models. For weighted 7-NN, we see roughly a 2% increase in classification accuracy. In regards to our decision tree model however, we see a vast improvement in the classification of high quality wines. Based on the confusion matrix shown in figure 2, our model correctly classifies approximately 53.5% of these data instances. This is nearly a two times improvement from our decision tree trained on the original UCI wine dataset, as the classification for high quality wines was only 29.63%. Overall, we would say that the introduction of these new data instances was beneficial to both the weighted 7-NN and decision tree classification models, and would expect these models to improve in accuracy if there was more balanced data.

6. Conclusion/Future Work

In this project, we evaluated four supervised learning algorithms—Naive Bayes, Random Forest, Weighted 7-NN, and Decision Trees—for classifying wine quality into three categories: low, medium, and high, using the UCI Wine Quality dataset. Our experiments confirmed Hypothesis 1, as Naive Bayes performed the worst with an average accuracy of 67.77%, due to class imbalance, correlated features, and non-Gaussian distributions, aligning with McCallum and Nigam's findings. Hypothesis 2 was not met, as Naive Bayes fell short of the 75% accuracy threshold. Random Forest's superior performance supports Breiman's claims of robustness to noise and correlated features. Hypothesis 3 was validated, with Decision Trees achieving the highest accuracy for the medium class (73.59%) due to the dataset's imbalance. Finally, Hypothesis 4 was confirmed, as Weighted 7-NN (78.60%) outperformed unweighted 7-NN (71.73%), highlighting the benefit of distance-based weighting.

For future work, we propose addressing the class imbalance by applying techniques like SMOTE to oversample the underrepresented high-quality class, potentially improving performance across all models, especially Naive Bayes. Additionally, we could enhance Naive Bayes by using PCA to decorrelate features or applying log transformations to skewed features like residual sugar, aligning better with its Gaussian assumptions. Implementing stratified train-test splits would ensure consistent class distributions, reducing variability in results. Exploring ensemble methods, such as combining Random Forest with boosting techniques like AdaBoost, could further boost accuracy. Lastly, tuning hyperparameters for Decision Trees (e.g., tree depth) and K-NN (e.g., distance metrics) may yield better performance, providing deeper insights into the dataset's structure.

7. Contributions

Kenny implemented weighted/unweighted k-NN, as well as decision tree Supervised Learning methods (hypotheses 3 and 4). In addition, the literature review seen in sections 4.3, 4.4, and 4.5 of this article respectively, with a comparison of weighted/unweighted k-NN (hypothesis 4) and a comparison of custom CART decision tree implementation against the decision tree model Aich et al. (2019) created for the same UCI Wine dataset. Lastly, section 5 (Novelty) was written by Kenny as well.

Tanner implemented naive bayes as well as random forest Supervised Learning methods (hyptotheses 1 and 2). In addition, the literature review seen in sections 4.1 and 4.2.

8. References

Andrew McCallum and Kamal Nigam. A Comparison of Event Models for Naive Bayes Text Classification. In AAAI-98 Workshop on Learning for Text Categorization, pages 41–48, 1998.

Leo Breiman. Random Forests. Machine Learning, 45(1):5–32, 2001.

Priyam, A., Abhijeet, Gupta, R., Rathee, A., & Srivastava, S. (2013). Comparative Analysis of Decision Tree Classification Algorithms. International Journal of Current Engineering and Technology, 3(2), 334–337

Liu, W., & Chawla, S. (2011). Class Confidence Weighted kNN Algorithms for Imbalanced Data Sets. In Machine Learning and Knowledge Discovery in Databases, pp. 345–360.

Aich, S., Al-Absi, A. A., Hui, K. L., Lee, J. T., & Sain, M. (2019). A Classification Approach with Different Feature Sets to Predict the Quality of Different Types of Wine using Machine Learning Techniques. In International Conference on Advanced Communications Technology (ICACT), pp. 139–143