ISSN: 1829-667X / E-ISSN: 2460-9021
DOI:10.31515/telematika.v21i2. 13007

Performance Evaluation of Multiple Machine Learning Models for Wine Quality Prediction

Evaluasi Kinerja *Multiple* Model *Machine Learning* untuk Prediksi Kualitas Wine

Dedik Fabiyanto[1], Yan Rianto[2]

[1,2] S2 Ilmu Komputer, Fakultas Teknologi Informasi, Universitas Nusa Mandiri, Indonesia

[1*]14220030@nusamandiri.ac.id, [2]yan.yrt@nusamandiri.ac.id
**: Penulis korenspondensi* (*corresponding author*)

Informasi Artikel
Received: December 2023
Revised: January 2024
Accepted: January 2024
Published: February 2024

Abstract

Research utilizing a dataset from the UCI repository evaluated the predictive accuracy of nine machine learning models for wine quality. The models employed include Logistic Regression, K-Nearest Neighbor (KNN), Decision Tree, Support Vector Machine (SVM), Random Forest, XGBoost, LightGBM, CatBoost, and Gradient Boosting. The dataset comprises 1,599 samples with 12 chemical parameters. Data preprocessing, including oversampling, normalization, standardization, and seeding, was performed to enhance model performance.

The study's findings indicate that the models with the highest accuracy values were LightGBM (87.80%), CatBoost (86.60%), and Random Forest (85.70%). A voting classifier combining these three models achieved an accuracy of 87.29%. Further analysis using a confusion matrix demonstrated that this combined model effectively predicts the "Good" and "Not Good" classes.

In conclusion, the combination of LightGBM, CatBoost, and Random Forest models proves to be an effective approach for predicting wine quality based on chemical parameters, with an accuracy value of 87.29%.

Keywords: wine quality, voting classifier, model evaluation
Kata kunci: kualitas wine, voting classifier, evaluasi model

Abstrak

Penelitian mengevaluasi kinerja sembilan model machine learning dalam memprediksi kualitas wine menggunakan dataset dari repositori UCI telah dilakukan. Model machine learning yang digunakan adalah Logistic Regression, K-Nearest Neighbor (KNN), Decision Tree, Support Vector Machine (SVM), Random Forest, XGBoost, LightGBM, CatBoost, dan Gradient Boosting. Dataset wine yang digunakan terdiri dari 1.599 sampel dengan 12 parameter

ISSN: 1829-667X / E-ISSN: 2460-9021
DOI:10.31515/telematika.v21i2. 13007

kimia. Pra-pemrosesan data termasuk pengaturan seed, oversampling, normalisasi, dan standarisasi dilakukan untuk meningkatkan kinerja multi model machine learning. Hasil penelitian menunjukkan bahwa model LightGBM, CatBoost, dan Random Forest memberikan akurasi tertinggi dengan masing-msing nilai akurasi secara berturut-turut sebesar 87,80%, 86,60%, dan 85,70%. Dengan menggunakan voting classifier yang menggabungkan ketiga model ini, akurasi sebesar 87,29% berhasil dicapai. Analisis lebih lanjut menggunakan *confusion matrix* menunjukkan bahwa model kombinasi ini memiliki performa yang baik dalam memprediksi kelas "*Good*" dan "*Not Good*". Penelitian ini menyimpulkan bahwa kombinasi model *Light GBM*, *Cat Boost*, dan *Random Forest* adalah pendekatan yang efektif untuk memprediksi kualitas anggur berdasarkan parameter kimia dengan nilai akurasi 87,29%.

1. Introduction

Wine is an alcoholic refreshment made by aging grapes and other natural products. The generation preparation includes yeast maturing the characteristic sugars within the natural product, changing them into liquor and carbon dioxide (CO2). Wine quality is impacted by different variables, including grape assortment, maturation strategies, capacity conditions, and the wine's age [1]. Wine quality is crucial in the alcoholic beverage industry, directly impacting consumer satisfaction and market price. Experts typically perform quality assessments using sensory methods, which require experience and are subjective [2]. while these conventional methods have proven effective, they are time-consuming and costly. Therefore, a more efficient and objective approach to wine quality assessment is needed. Advances in technology, particularly in Artificial Intelligence (AI) and machine learning, offer opportunities to develop more efficient and objective approaches for assessing wine quality. Traditional approaches to predicting wine quality use statistical methods such as linear regression or discriminant analysis. However, with advancements in machine learning, particularly deep learning, there is an opportunity to improve prediction accuracy. Deep learning models, such as Deep Neural Networks (DNN) and Convolutional Neural Networks (CNN), have shown success in various predictive applications [3]. These approaches aim to reduce the subjectivity associated with human assessment and improve consistency in determining wine quality. Machine learning algorithms, like deep learning, enable more in-depth and precise analysis of the physicochemical data related to wine [4].

The primary journal referenced in this research discusses wine quality prediction using machine learning algorithms. The data originates from a public dataset that includes various chemical components in wine. Researchers conducted data analysis and visualization, applying several machine learning algorithms, including Random Forest, XGBoost, and Decision Tree, to predict wine quality. The findings indicate that the Random Forest model has the best predictive accuracy at 66.8%, followed by XGBoost at 60.1% and Decision Tree at 59.5%. The researchers

concluded that these machine learning models perform well in predicting high-quality wine but less so for low-quality wine [5].

Previous studies have explored the use of machine learning methods for wine quality prediction. Research by Jeffrey A. Clarin compared the performance of several regression algorithms in predicting white wine quality using a dataset from the UCI Machine Learning Repository and implemented using WEKA. The study found that the Random Forest algorithm provided the best performance with a correlation coefficient of $r = 0.7459$. Among the input variables, alcohol and acidity remained significantly correlated with the prediction model performance, with values of $r = 0.44$, and $r = -0.391$ respectively [6].

Another think about utilizing a machine learning technique to look at 1,599 wine tests, each containing 11 input parameters, to recognize the factors with the foremost noteworthy effect on by and large wine quality. The utilization of direct relapse models in this think about appeared that liquor and causticity were the essential components influencing wine quality. Furthermore, warm maps were utilized to display the connections among these factors. Assist investigation utilized box plots and three-dimensional scramble plots to strengthen the conclusions inferred from the straight relapse demonstration, giving more particular experiences into the factors that have the most noteworthy impact on wine quality [4].

Other investigations compared the execution of a few relapse models and combinations of relapse and gathering models in anticipating wine quality utilizing the wine quality dataset from the UCI Machine Learning Store. This dataset comprises white and ruddy Vinho Verde wines from northern Portugal, with 6,497 tests. Sometime recently preparing the models, the dataset experienced suitable preprocessing steps to guarantee information quality and consistency. Five relapse algorithms Linear Relapse (LR), Arbitrary Timberland Regressor (RF), Bolster Vector Relapse (SVR), Choice Tree Regressor (DT), and Multi-layer Perceptron Regressor (MLP)—were prepared and tried on the dataset. Furthermore, expectations from these person relapse models were combined with four outfit models XGB Regressor (XGB), AdaBoost Regressor (ABR), Stowing Regressor (BR), and Slope Boosting Regressor (GRB). They come about demonstrated that among person models, Arbitrary Woodland (RF) appeared the finest execution, with the most reduced MAE, MSE, and RMSE values and the most noteworthy R^2 score. This recommends that RF is more suited to the ruddy wine quality dataset compared to other relapse models. In any case, combining Irregular Woodland with Sacking Regressor (RF and BR) outflanked the person models, appearing with lower mistakes and generally higher R^2 scores [7].

2. Metode/Perancangan

This research employs quantitative methods with multiple machine-learning models. The nine machine learning models used are Logistic Regression, K-Nearest Neighbor (KNN), Decision Tree, Support Vector Machine (SVM), Random Forest, XGBoost, LightGBM, CatBoost, and Gradient Boosting. The selection of these nine models provides a broad spectrum, allowing for a comprehensive evaluation and precise accuracy comparison. This approach also helps identify which model best fits the data characteristics. The overall research steps are shown in **Figure 1**.

ISSN: 1829-667X / E-ISSN: 2460-9021
DOI:10.31515/telematika.v21i2. 13007

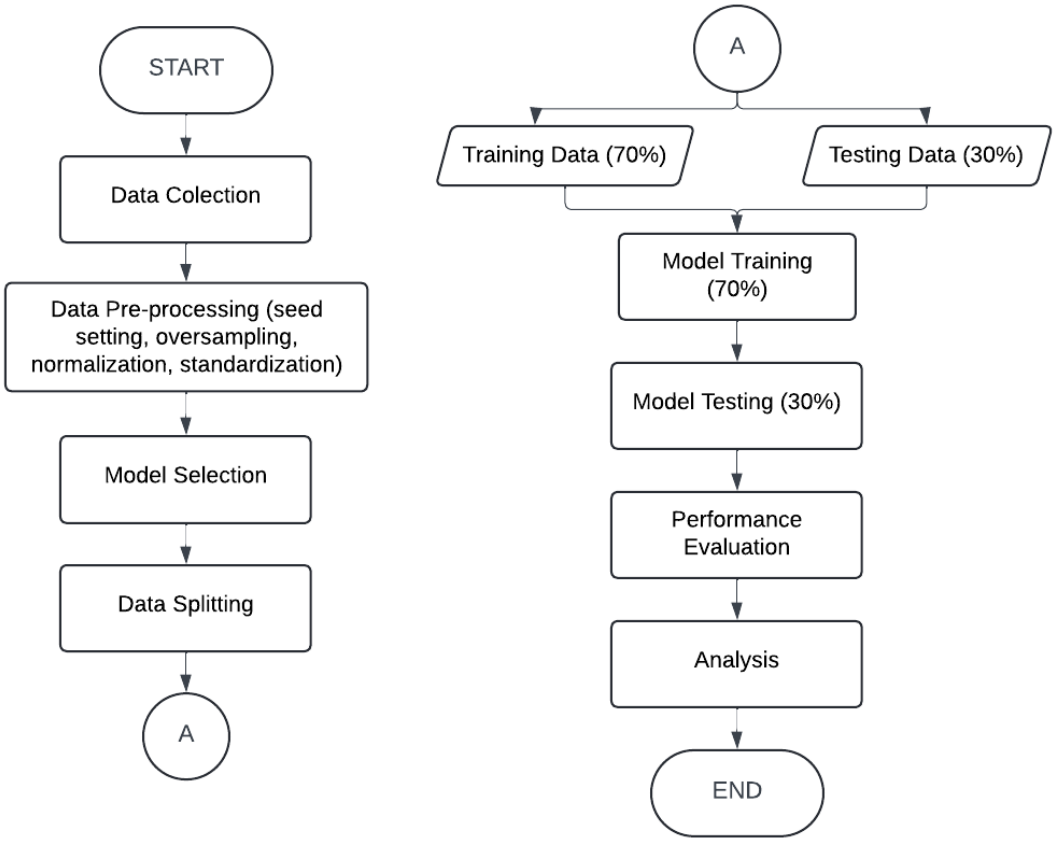

Figure 1. Research Flowchart

2.1. Data Collection

The dataset used is the wine quality dataset available in the UCI machine learning repository [8]. This dataset consists of 1,599 wine samples with 12 chemical parameters and their quality labels. The 12 parameters used are shown in **Table 1**.

Table 1. Chemical Parameters of Wine [9]

No.	Parameter	Information
1.	*Fixed acidity*	The total amount of non-volatile acids in wine, mainly tartaric, malic, and citric acids
2.	*Volatile acidity*	The total amount of volatile acids, primarily acetic acid
3.	*Citric acid*	Natural acid found in citrus fruits
4.	*Residual sugar*	The amount of sugar remaining after the completion of alcoholic fermentation
5.	*Chlorides*	The content of chloride ions in wine
6.	*Free sulfur dioxide* (SO2)	The amount of free sulfur dioxide available in wine
7.	*Total sulfur dioxide* (SO2)	The total amount of sulfur dioxide, including both free and bound forms
8.	*Density*	The density of wine, often correlated with alcohol and sugar content

No.	Parameter	Information
9.	pH	The acidity or alkalinity level of wine
10.	*Sulphates*	Sulfur compounds, such as potassium sulfate, found in wine
11.	*Alcohol*	The alcohol content in wine, usually measured as a percentage of volume
12.	*Quality*	Wine quality rating, usually on a numerical scale (e.g., 0-10)

Twelve chemical parameters were tested on 1,599 wine samples with varying values, as shown in **Table 2**, which provides an example of the dataset used in this study.

Table 2. Example of Dataset and Parameters Used

fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcohol	quality
7.4	0.7	0.0	1.9	0.076	11.0	34.0	0.9978	3.51	0.56	9.4	5
7.8	0.88	0.0	2.6	0.098	25.0	67.0	0.9968	3.2	0.68	9.8	5
7.8	0.76	0.04	2.3	0.092	15.0	54.0	0.997	3.26	0.65	9.8	5
11.2	0.28	0.56	1.9	0.075	17.0	60.0	0.998	3.16	0.58	9.8	6
7.4	0.7	0.0	1.9	0.076	11.0	34.0	0.9978	3.51	0.56	9.4	5
7.4	0.66	0.0	1.8	0.075	13.0	40.0	0.9978	3.51	0.56	9.4	5
7.9	0.6	0.06	1.6	0.069	15.0	59.0	0.9964	3.3	0.46	9.4	5

2.2. Data Preprocessing

Data quality significantly influences the outcome of the model built, hence preprocessing steps are necessary. The steps in data preprocessing for this study include seed setting, oversampling, normalization, and standardization. Each step aims to prepare the data so that the model can learn more effectively and produce accurate predictions. The first step, seed setting, involves setting the initial value for the random number generator. Seed setting ensures that data splitting, random sample selection, and other random processes can be repeated with the same results each time training and testing are run, thus validating the results consistently [10]. The next step, oversampling, addresses the issue of class imbalance in the dataset. Oversampling techniques, such as the Synthetic Minority Over-sampling Technique (SMOTE), have been proven effective in improving model performance on imbalanced datasets. Oversampling helps the model learn better about the features of the minority class, thereby improving the model's performance in predicting that class [11].

Normalization is the process of changing the scale of features in the data so that they have the same range, usually between 0 and 1. Normalization helps in accelerating the convergence of algorithms, reducing variability in data, and allowing the model to learn more efficiently from the data, thus improving prediction accuracy. The normalization method used in this study is Min-Max Scaling [12]. The final process is standardization, which transforms the distribution of features to have a mean of 0 and a standard deviation of 1. Unlike normalization, which adjusts the data scale to a specific range, standardization ensures that the data has a uniform

distribution. This is very useful in algorithms such as Support Vector Machines (SVM) that assume the data is normally distributed. With standardization, features with different scales can be treated equally by the model, potentially improving overall model performance [13].

2.3. Model Selection

Multiple machine-learning models were used to compare the performance of various classifiers in the modeling process. The models used in this study are shown in **Table 3**.

Table 3. Research Models Used

No.	Research Model	Python Code
1.	*Logistic Regression*	*lr = LogisticRegression(max_iter=500, n_jobs=-1, random_state=SEED)*
2.	*K-Nearest Neighbors (KNN)*	*knn = KNeighborsClassifier()*
3.	*Decision Tree*	*dt = DecisionTreeClassifier(random_state=SEED)*
4.	*Support Vector Classifier (SVC)*	*svc = SVC(random_state=SEED)*
5.	*Random Forest*	*rf = RandomForestClassifier(random_state=SEED)*
6.	*Extreme Gradient Boosting (XGBoost)*	*xgbc = xgb.XGBClassifier(random_state=SEED)*
7.	*LightGBM*	*lgbmc = lgbm.LGBMClassifier(random_state=SEED)*
8.	*CatBoost*	*cbc = cb.CatBoostClassifier(random_state=SEED, verbose=False)*
9.	*Gradient Boosting*	*gbc = GradientBoostingClassifier(random_state=SEED)*

2.4. Dataset Splitting

The dataset consisting of 1,599 samples was then split into training and testing data with a composition of 70% and 30%. The training data was used to train the predetermined models, and the testing data was used to further test or evaluate model performance.

2.5. Model Performance Evaluation

Model performance evaluation was conducted by comparing the accuracy values of each model. The three models with the highest accuracy values were then combined using the voting classifier method.

2.6. Analysis

Subsequently, an analysis of the model performance evaluation results was conducted through the confusion matrix. The confusion matrix provides an overview of the model's prediction distribution and informs about the performance of the classification model by comparing predicted values with actual values from the test data. The information includes the True Positive (TP) value, which is the number of positive cases correctly predicted by the model, meaning the model accurately identifies positive cases. True Negative (TN) is the number of negative cases correctly predicted by the model, meaning the model accurately identifies

ISSN: 1829-667X / E-ISSN: 2460-9021
DOI:10.31515/telematika.v21i2. 13007

negative cases. False Positive (FP) is the number of negative cases incorrectly predicted as positive by the model. False Negative (FN) is the number of positive cases incorrectly predicted as negative by the model [14]. From these four values, further evaluation metrics such as accuracy, precision, recall, and F1-score can be calculated [15].

Accuracy is the total percentage of correct predictions out of all predictions made by the model. It is calculated as shown in **Equation (1)** and **Equation (2)**.

$$Accuracy = \left(\frac{TP+TN}{TP+FP+FN+TN}\right) \quad (1)$$

Accuracy for more than one class:

$$Accuracy = \frac{True\ Positive}{Jumlah\ sampel} \quad (2)$$

Precision shows the percentage of positive cases correctly predicted out of all positive predictions made by the model. It is calculated as shown in **Equation (3)**.

$$Precision = \left(\frac{TP}{TP+FP}\right) \quad (3)$$

Recall (Sensitivity or True Positive Rate) is the percentage of positive cases correctly identified by the model out of all actual positive cases. It is calculated as shown in **Equation (4)**.

$$Recall = \left(\frac{TP}{TP+FN}\right) \quad (4)$$

F1-score is the harmonic mean of precision and recall. The F1-score provides a balance between these two metrics and is useful when there is a class imbalance. It is calculated as shown in **Equation (5)**.

$$F1 - Score = \frac{2*(Recall*Precission)}{(Recall+Precission)} \quad (5)$$

3. Results and Discussion

The distribution of wine quality in the dataset on a scale of 0-10 is shown in **Table 4**.

Table 4. Frequency Distribution of Wine Quality

Quality	Frequency
5	681
6	638
7	199
4	53
8	18
3	10

ISSN: 1829-667X / E-ISSN: 2460-9021
DOI:10.31515/telematika.v21i2. 13007

From **Table 3**, it is evident that the wine quality with the highest number of samples is 5 with 681 samples, followed by 6 with 638 samples. Meanwhile, the qualities with the least number of samples are 3 and 8, with 10 and 18 samples, respectively. The frequency distribution of wine quality can be seen in **Figure 2**.

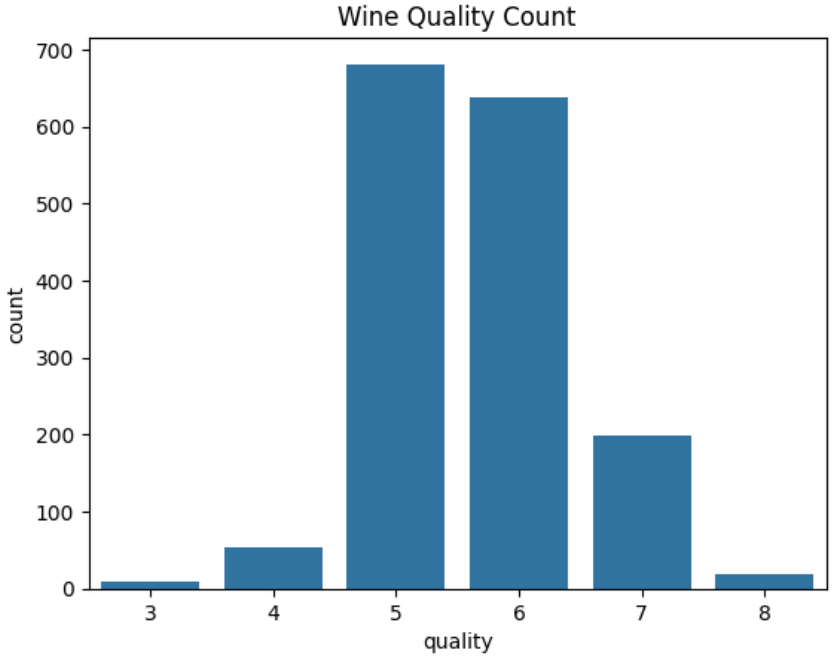

Figure 2. Frequency Distribution of Wine Quality

The researcher further examined the correlation of two variables with wine quality, namely alcohol content and fixed acidity. The correlation between alcohol content and wine quality is shown in **Figure 3**.

ISSN: 1829-667X / E-ISSN: 2460-9021
DOI:10.31515/telematika.v21i2. 13007

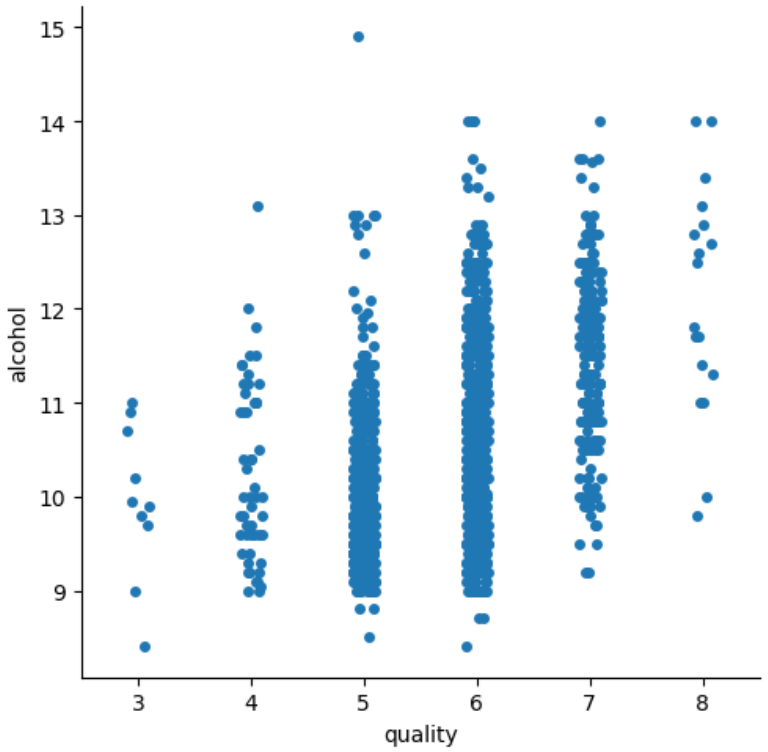

Figure 3. Correlation Between Wine Quality and Alcohol Content

Each dot in the scatter plot represents a wine sample in the dataset. The distribution pattern suggests that wines with lower quality (scores 3 and 4) tend to have varying alcohol content, but generally in the lower range (around 9-11%). Wines with medium quality (scores 5 and 6) show a denser distribution with varying alcohol content, but there is a significant concentration in the 9-11% range for quality 5 and 10-12% for quality 6. Higher quality wines (scores 7 and 8) tend to have higher alcohol content overall. For quality 7, alcohol content is often in the 10-13% range, and for quality 8, despite the small sample size, alcohol content tends to be higher and more variable. The positive correlation observed suggests that higher-quality wines tend to have higher alcohol content. This is seen from the rightward distribution of dots (higher quality), with alcohol content tending to increase. The plot also shows considerable variability in alcohol content for each quality level, especially for qualities 5 and 6, which show denser and broader spread.

The second variable reviewed is shown in **Figure 4**, which is the correlation between fixed acidity and wine quality. The distribution shows that wines with quality scores of 5, 6, and 7 have higher data point densities, indicating these quality scores are more common in the dataset. In contrast, quality scores of 3, 4, and 8 have fewer data points, indicating these scores are less common. There is no strong linear relationship between fixed acidity and wine quality. Fixed acidity values for wines with quality scores of 5, 6, and 7 show a wide range, from around 6 to over 14.

ISSN: 1829-667X / E-ISSN: 2460-9021
DOI:10.31515/telematika.v21i2. 13007

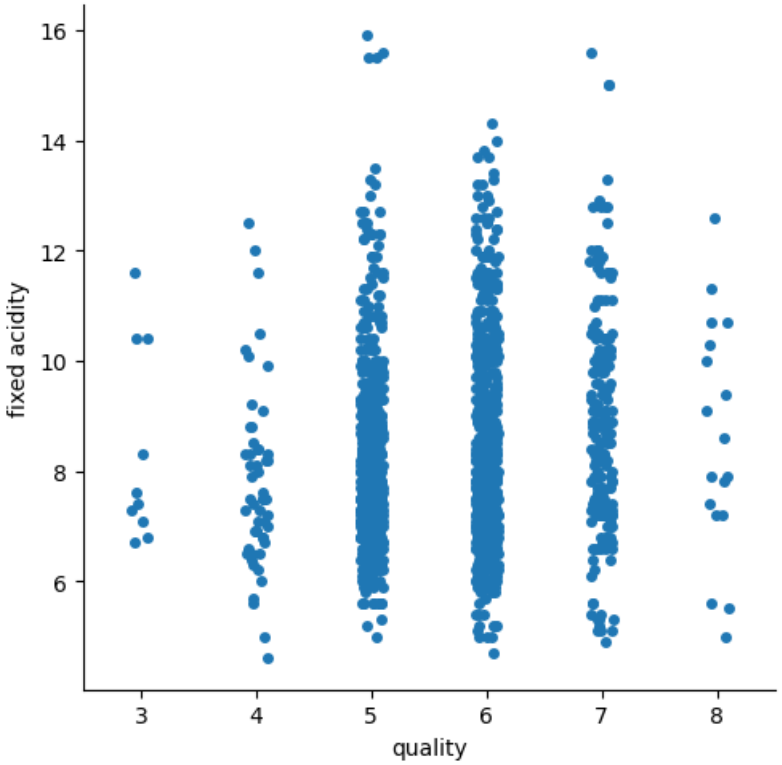

Figure 4. Correlation Between Wine Quality and Fixed Acidity

The highest fixed acidity observed is around 16, which appears in wines with a quality score of 5. Wines with a quality score of 8 have lower fixed acidity, primarily ranging between 6 and 10. The correlation between wine quality and other variables in the dataset is shown through a heatmap in **Figure 5**.

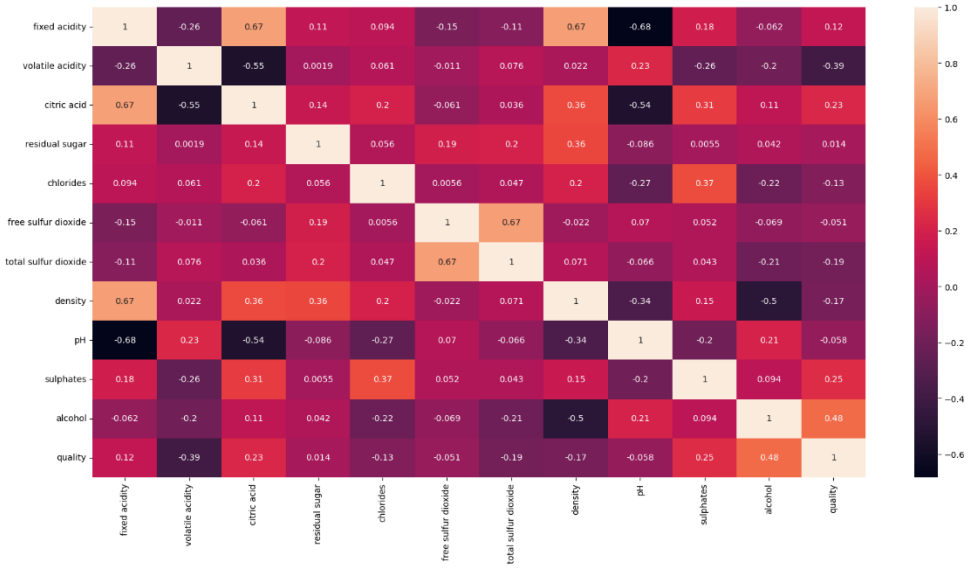

Figure 5. Heatmap of Wine Quality Correlation with Dataset Variables

ISSN: 1829-667X / E-ISSN: 2460-9021
DOI:10.31515/telematika.v21i2. 13007

The heatmap appears relationship values extending from -1 to 1. A esteem of 1 shows a idealize positive relationship, meaning that as one trait increments, the other trait too increments relatively. A esteem of -1 demonstrates a culminate negative relationship, meaning that as one property increments, the other quality diminishes relatively. A esteem of shows no relationship between the two traits. Based on the color elucidation, lighter colors (toward white) show more grounded relationships (both positive and negative), whereas darker colors (toward dark) show weaker or no relationship.

The correlation between variables shows a strong positive correlation between fixed acidity and density (0.67) and between citric acid and fixed acidity (0.67). Total sulfur dioxide and free sulfur dioxide also show a very strong positive correlation (0.67). The anticipated strong correlation between alcohol and quality turned out to be moderately strong based on the correlation heatmap, with a value of 0.48. Additionally, the correlations between volatile acidity and quality, and between citric acid and pH, show moderately strong negative correlations, with values of -0.39 and -0.54 respectively.

3.1. Modeling

The researcher compared the accuracy of models before and after normalization and standardization, as shown in **Table 5** and **Figure 6**.

Table 5. Model Accuracy Comparison

Model	Accuracy		
	Before Normalization and Standardization	After Normalization	After Standardization
Logistic Regression	56.7%	58,90%	58,70%
K-Nearest Neighbors (KNN)	69.6%	75,10%	76,70%
Support Vector Classifier (SVC)	42.7%	72,90%	*77,00%*
Random Forest	85.7%	85,30%	85,60%
Decision Tree	78.1%	78,70%	79,00%
Extreme Gradient Boosting (XGBoost)	77.5%	77,80%	77,80%
LightGBM	87.8%	87,80%	86,40%
CatBoost	86.4%	86,60%	86,60%
Gradient Boosting	81.8%	82,00%	82,00%

ISSN: 1829-667X / E-ISSN: 2460-9021
DOI:10.31515/telematika.v21i2. 13007

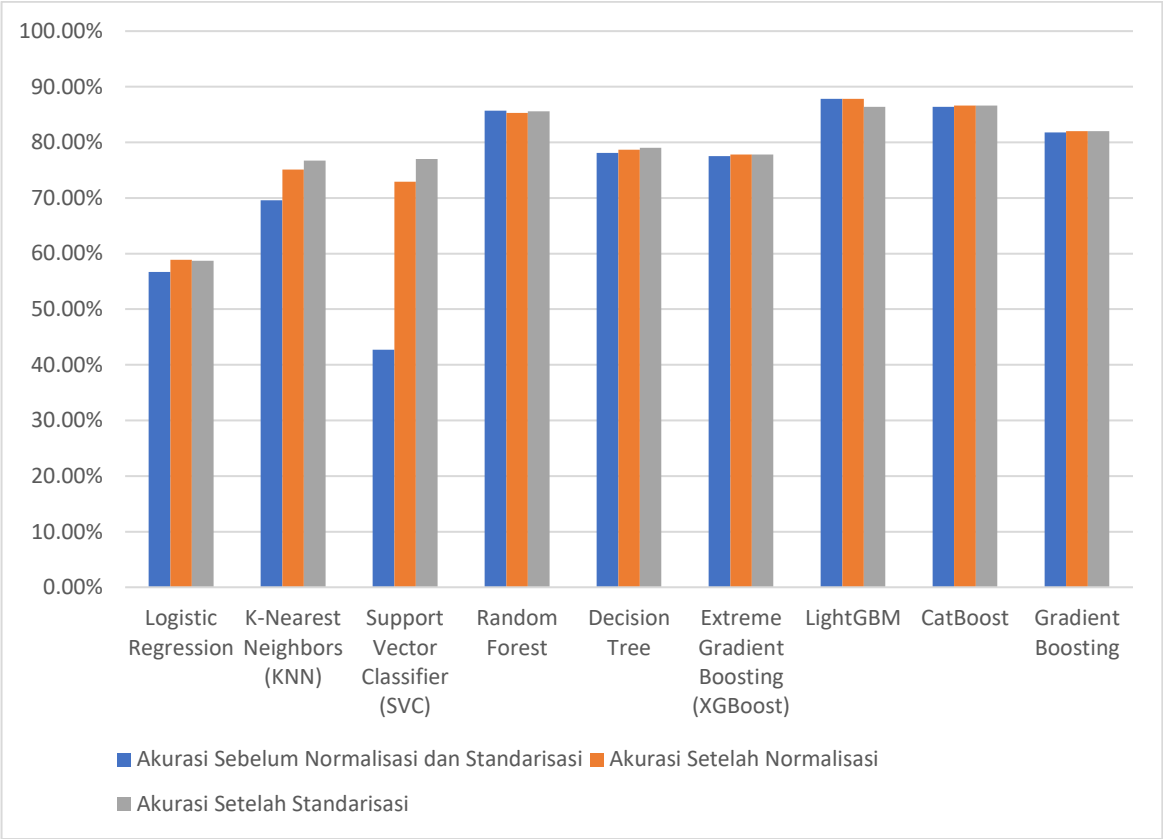

Figure 6. Model Accuracy Comparison

The table and figure above show that the three models with the highest accuracy are LightGBM, CatBoost, and Random Forest with accuracy values of 87.80%, 86.60%, and 85.70%, respectively. To achieve better accuracy, the researcher combined the models with the highest accuracy (LightGBM, CatBoost, and Random Forest) into a voting classifier. The accuracy of the voting classifier model is 87.29%. This shows that the combination of LightGBM, CatBoost, and Random Forest models provides good results.

3.2. Model Analysis

The perplexity network utilized to assess the execution of the voting classifier demonstrate is appeared in Figure 7. Both the real and anticipated values comprise of two classes: "Good" and "Not Good". Each cell within the network appears the number of forecasts that drop into a certain category. The network has the structure (0, 0): Anticipated "Good" and real esteem "Good" (TP), (0, 1): Anticipated "Great" and real esteem "Not Good" (FP), (1, 0): Anticipated "Not Good" and real esteem "Good" (FN), and (1, 1): Anticipated "Not Good" and real esteem "Not Good" (TN).

True Positive **(TP):** According to the matrix, there are 132, 128, 115, 95, 111, and 133 cases where the model correctly predicted "good" as "good".

True Negative **(TN):** The model correctly identified "not good" as "not good" in 0 (or close to zero) cases.

ISSN: 1829-667X / E-ISSN: 2460-9021
DOI:10.31515/telematika.v21i2. 13007

***False Positive* (FP)**: There are 1, 2, 3, 1, and 1 cases where the model incorrectly predicted "not good" as "good".

***False Negative* (FN):** There are 1, 2, 7, 8, 35, 12, and 4 cases where the model incorrectly predicted "good" as "not good".

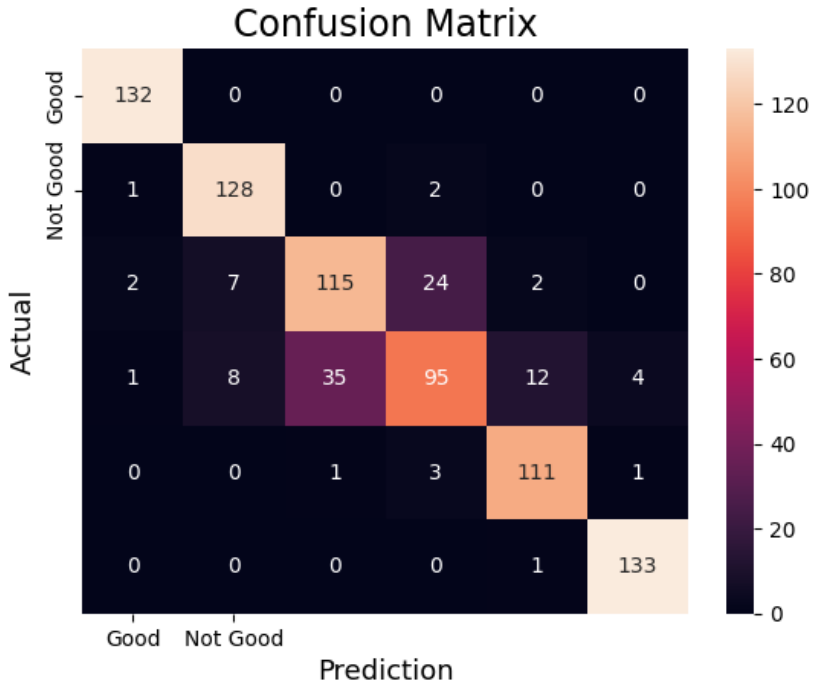

Figure 7. Confusion Matrix of Voting Classifier Model

The combined classification model shows good performance in predicting the "Good" class. The precision for this class is 0.6567, which means that about 65.67% of all predictions stating "Good" are correct. The recall for the "Good" class is 0.9635, indicating that the model successfully identifies about 96.35% of all actual "Good" cases. The F1-score for the "Good" class is 0.7836, which is a harmonic measure of precision and recall, showing a good balance between them despite some false positives.

The model shows excellent performance in predicting the "Not Good" class. The precision for this class is 1.0, meaning all predictions stating "Not Good" are correct without any errors. The recall for the "Not Good" class is 0.9568, indicating that the model can identify about 95.68% of all actual "Not Good" cases. The F1-score for the "Not Good" class is 0.978, indicating an excellent balance between precision and recall, with very few prediction errors. This is comprehensively shown in **Table 6**.

Table 6. Model Performance Comparison

Class	Precision	Recall	F1-Score
Good	0.6567	0.9635	0.7836
Not Good	1.0	0.9568	0.978

4. Conclusion and Suggestions

his research has successfully evaluated the performance of nine machine learning models in predicting wine quality using a wine quality dataset from the UCI machine learning repository. The models used are Logistic Regression, K-Nearest Neighbor (KNN), Decision Tree, Support Vector Machine (SVM), Random Forest, XGBoost, LightGBM, CatBoost, and Gradient Boosting. Through a series of data preprocessing steps such as seed setting, oversampling, normalization, and standardization, this study aims to prepare the data for more effective model learning and accurate predictions.

The results show that data normalization and standardization positively impact the accuracy of most models, with some models showing significant accuracy improvements after normalization and standardization. The three models with the highest accuracy are LightGBM, CatBoost, and Random Forest with accuracy values of 87.80%, 86.60%, and 85.70%, respectively. Combining these three models in a voting classifier resulted in an accuracy of 87.29%, showing that model combination can yield good results.

Further analysis through the confusion matrix and evaluation metrics of precision, recall, and F1-score shows that the voting classifier model has excellent performance in predicting both "Good" and "Not Good" classes. Precision, recall, and F1-score for the "Not Good" class reach very high values, each being 1.0, 0.9568, and 0.978, respectively, indicating an excellent balance between precision and recall with very few prediction errors. For the "Good" class, the precision, recall, and F1-score are 0.6567, 0.9635, and 0.7836, respectively, showing that the model also performs well in predicting this class despite some false positives.

Overall, this research shows that with proper data preprocessing and the appropriate selection of machine learning models, accurate and reliable models for predicting wine quality can be obtained. The voting classifier model combining LightGBM, CatBoost, and Random Forest proves to be an effective approach in this study, providing optimal results in wine quality classification.

References

[1] R. Zhu, “Chemical Change and Quality Control in Winemaking,” *Scientific and Social Research,* vol. 4, no. 7, pp. 62-67, 14 Juli 2022.

[2] M. H. Shahrajabian dan W. Sun, “Assessment of Wine Quality, Traceability and Detection of Grapes Wine, Detection of Harmful Substances in Alcohol and Liquor Composition Analysis,” *Letters in Drug Design & Discovery,* vol. 21 (8), no. Doi: 10.2174/1570180820666230228115450, pp. 1377-1399, Juni 2024.

[3] L. Le, P. N. Hurtado, I. Lawrence, Q. Tian dan B. Chen, “Applying Neural Networks in Wineinformatics with the New Computational Wine Wheel,” *Fermentation*, vol. 9 (7), no. Doi: 10.3390/fermentation9070629, pp. 629-629., 2023.

[4] J. Dong, “Red Wine Quality Analysis based on Machine Learning Techniques,” *Highlights in Science, Engineering and Technology,* vol. 49, no. Doi: 10.54097/hset.v49i.8506, pp. 208-213, 2023.

[5] C. Zeng, J. Fang, Q. Yang, C. Xiang, Z. Zhao dan Y. Lei, "Wine quality grade data analysis and prediction based on multiple machine learning algorithms," dalam *Proceedings of the 2nd International Conference on Mechatronics and Smart Systems,* 2024.

[6] J. A. Clarin, "Comparison of the Performance of Several Regression Algorithms in Predicting the Quality of White Wine in WEKA," *International Journal of Emerging Technology and Advanced Engineering,* vol. 12 (07), no. Doi: 10.46338/ijetae0722_03 , pp. 20-26, 3 Juli 2022.

[7] A. K., "Regression Modeling Approaches for Red Wine Quality Prediction: Individual and Ensemble," *International Journal for Research in Applied Science & Engineering Technology (IJRASET),* vol. 11, no. Doi: doi.org/10.22214/ijraset.2023.54363, pp. 3621-3627, Juni 2023.

[8] N. Pourmoradi, "Red Wine Quality," Kaggle, 2023. [Online]. Available: https://www.kaggle.com/code/nimapourmoradi/red-wine-quality/input. [Diakses 21 Juni 2024].

[9] R. S. Jackson, Wine Science: Principles and Applications (3rd Edition), Burlington: Academic Press, 2008.

[10] A. Géron, Hands-on Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems (2nd edition), N. Tache, Penyunt., Sebastopol: O'Reilly Media, Inc, 2019.

[11] Ridwan, E. H. Hermaliani dan M. Ernawati, "Penerapan Metode SMOTE Untuk Mengatasi Imbalanced Data Pada," *Computer Science (CO-SCIENCE),* vol. 4 (1), no. E-ISSN: 2774-9711, pp. 80-88, Januari 2024.

[12] D. A. Nasution, H. H. Khotimah dan N. Chamidah, "Perbandingan Normalisasi Data untuk Klasifikasi Wine Menggunakan Algoritma K-NN," *CESS (Journal of Computer Engineering System and Science),* vol. 4 (1), pp. 78-82, Januari 2019.

[13] S. Ioffe dan C. Szegedy, "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift," dalam *Proceedings of the 32nd International Conference on Machine Learning, PMLR,* 2015.

[14] K. S. Nugroho, "Confusion Matrix untuk Evaluasi Model pada Supervised Learning," 13 November 2019. [Online]. Available: https://ksnugroho.medium.com/confusion-matrix-untuk-evaluasi-model-pada-unsupervised-machine-learning-bc4b1ae9ae3f. [Diakses 23 Juni 2024].

[15] S. Raschka dan V. Mirjalili, Python Machine Learning, Burningham: Packt Publishing Ltd., 2019.