Proceedings of CONF-MLA 2026 Symposium: Learning and Decision Making in Multi Agent Software Systems
DOI: 10.54254/2755-2721/2026.BA36266

Wine Quality Prediction by Integrating SQL-Based Feature Engineering and Random Forest Modeling

Youyi Huang

Department of E-commerce, South China University of Technology, Guangzhou, China
202466860234@mail.scut.edu.cn

Abstract. Traditional wine quality assessment based on tasters is costly and subjectively inconsistent, making it difficult to meet the needs of automated quality control for large volumes of data. This study imports the 1,143-record WineQT dataset into MySQL, checks data quality, and constructs two SQL-derived features before six-class modeling. The initial row-wise split gave the full-feature random forest an accuracy of 0.7118 and a macro-F1 of 0.3415. A later audit, however, identified 125 repeated rows. Among these, 43 test records had identical counterparts in the training set. Keeping identical observations in the same group reduced accuracy to 0.5721 and macro-F1 to 0.2713, providing a more conservative estimate. In repeated stratified group five-fold cross-validation, the original features achieved 0.5984±0.0300 accuracy and 0.2793±0.0218 macro-F1; adding the SQL-derived features produced no improvement. Class weighting raised macro-F1 but lowered accuracy and weighted-F1. SQL therefore contributes mainly through consistent data checks, feature definitions, and a more traceable evaluation process.

Keywords: Wine quality prediction, SQL-based feature engineering, database governance, random forest, model interpretability

1. Introduction

Wine quality affects product grading, pricing, and production decisions. Sensory evaluation can consider aroma, taste, and finish, but it becomes costly and less consistent when many samples are involved. Physicochemical measurements, including acidity, alcohol, and sulfur dioxide, therefore provide a useful basis for computer-assisted screening and quality control.

Machine learning has been widely used for wine quality prediction. Cortez et al. developed a public dataset with 11 physicochemical variables, providing a reproducible basis for later experiments [1]. Jain et al. reported good results from random forest and extreme gradient boosting, while Bhardwaj et al. showed that sample structure can affect model performance [2, 3]. Zaza et al. examined class imbalance and identified alcohol as an important variable [4]. Chen's benchmark study also highlighted the need to prevent data leakage and use stratified cross-validation [5]. These studies have improved dataset construction, model comparison, and experimental practice, but the recognition of minority quality grades and the traceability of data processing remain open problems.

Most studies use prepared data files directly and give less attention to database-level checks and feature management. Although this approach may be sufficient for a single experiment, scattered

© 2026 The Authors. This is an open access article distributed under the terms of the Creative Commons Attribution License 4.0 (https://creativecommons.org/licenses/by/4.0/).

DOI: 10.54254/2755-2721/2026.BA36266

scripts make cleaning rules and feature definitions harder to trace and reuse when data sources expand or several users work on the same project. In this paper, the WineQT data are imported into MySQL. SQL is used to check missing values, duplicate record identifiers, repeated observations, and score ranges, while a database view generates two derived features. Different feature sets and classification models are then compared. The main questions are whether the SQL-derived features provide additional predictive information and how random forest performs on imbalanced data. The database is mainly used to keep data checks and feature definitions consistent, while the machine-learning models perform the classification task. The resulting workflow provides a reproducible reference for later studies that combine database processing with wine-quality modeling.

2. Methods

2.1. Data source and database construction

This study uses the WineQT dataset, compiled and released on Kaggle, as the experimental dataset. It is derived from publicly available wine quality data and contains 1,143 red wine samples [6]. Each sample contains several physicochemical variables and a quality score, with scores ranging mainly from 3 to 8. Sample identifiers are used only to distinguish records and are not included as model inputs. Model training uses the physicochemical and derived features, with quality score as the classification target.

In terms of data organization, this paper first imports the raw data into a MySQL database and establishes data tables corresponding to the original fields. This data table uses the sample number as the primary key, mainly to ensure that each record can be uniquely identified. Compared with directly reading the data file, a database table can more clearly preserve field types and data structures, and it also facilitates the unified checking of missing values, duplicate values, and abnormal scores before modeling.

2.2. Data preprocessing and SQL-based feature engineering

SQL checks found no missing values, duplicate identifiers, or scores outside 3–8. Comparison of the 11 physicochemical variables and quality label identified 125 repeated rows, so the 1,143 records formed 1,018 independent predictor groups. The rows were retained to preserve the original dataset, but identical observations were assigned to the same group during the stricter evaluation. This prevents the same physicochemical profile from appearing in both training and test sets.

Two composite features are constructed from the original physicochemical variables. The first is the alcohol-density index, which represents the ratio between alcohol content and density. Because density is jointly affected by alcohol, sugar, and other dissolved substances, this index provides a simplified representation of wine structure beyond alcohol content alone. It is calculated as follows:

$$\text{Alcohol} - \text{density index} = \frac{\text{Alcohol}}{Density} \tag{1}$$

The second feature is the acidity stability index, defined as the difference between fixed acidity and volatile acidity. Fixed acidity is related to the basic acid structure of wine, whereas excessive volatile acidity may produce pungent odors. Their difference therefore provides a simplified description of acidity status. It is calculated as follows:

$$Acidity\ stability\ index = Fixed\ acidity - Volatile\ acidity \tag{2}$$

DOI: 10.54254/2755-2721/2026.BA36266

2.3. Model development and evaluation

Four models were compared: decision tree, logistic regression, k-nearest neighbors (KNN), and random forest. Random forest follows Breiman's method [7]. Its performance was first tested with the original 11 variables and then with the two SQL-derived features. The full set was used for the four-model comparison.

The initial experiment used an 80/20 row-wise stratified split (random_state=42). After the duplicate audit, an approximately 80/20 stratified group split was added as the primary evaluation: records with the same 11 original variables remained together, and no group appeared in both sets. Logistic regression and KNN used standardized inputs. Accuracy, macro-F1, weighted-F1, the confusion matrix, and random forest feature importance were reported. The models were implemented using scikit-learn [8]. Macro-F1 was emphasized because it gives equal weight to each quality class [9].

Repeated stratified group five-fold cross-validation was also performed five times. Means and standard deviations were reported, and default and balanced class-weight random forests were compared. Figure 1 summarizes the workflow.

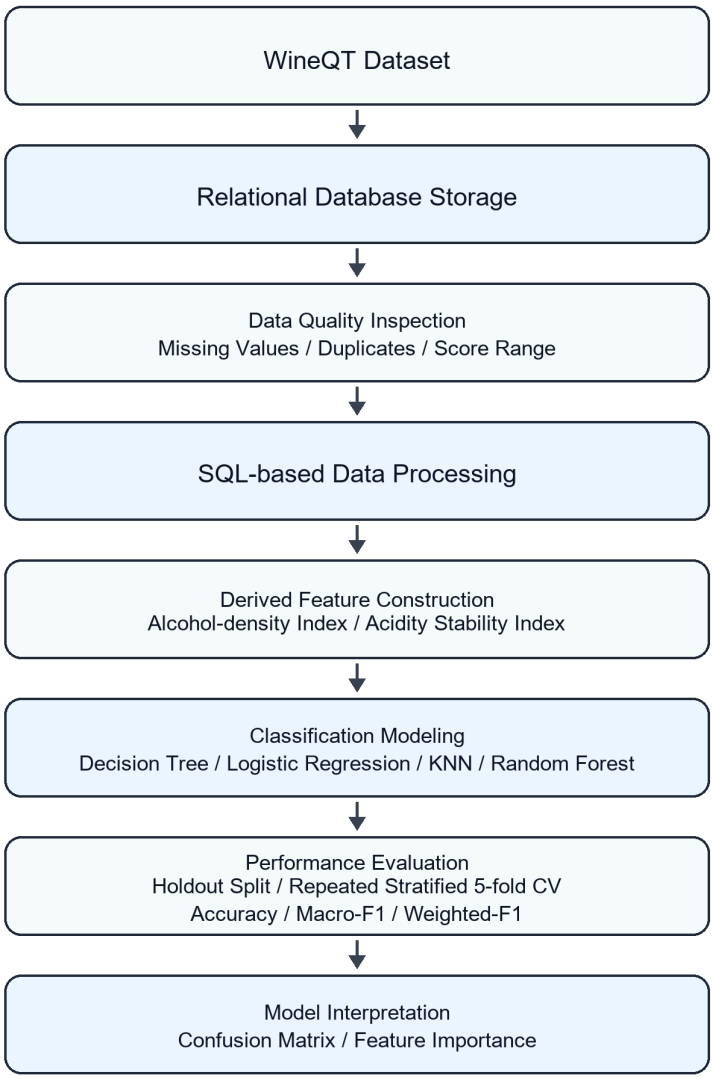

Figure 1. Modeling workflow used in this study (photo/picture credit: original)

DOI: 10.54254/2755-2721/2026.BA36266

3. Results and data analysis

3.1. Sample distribution and class imbalance

The dataset is imbalanced: scores 5 and 6 contain 483 and 462 records, while scores 3, 4, and 8 contain only 6, 33, and 16. The original distribution was retained. Macro-F1 and weighted-F1 were reported with accuracy, and the group-aware split kept identical observations together.

3.2. Correlation analysis of physicochemical variables

Table 1 presents the Pearson correlations between the physicochemical variables and quality scores. Alcohol has the strongest positive correlation with quality at 0.485. The alcohol-density index has a very similar value of 0.484, suggesting that this derived feature mainly retains information already contained in alcohol. Sulphates and citric acid also show positive correlations, while volatile acidity has the strongest negative correlation at −0.407. Since correlation only describes linear relationships, these results are considered together with the classification results and feature importance.

Table 1. Pearson correlations with wine quality

Feature	Correlation	Feature	Correlation
Alcohol	0.485	pH	−0.053
Alcohol-density index	0.484	Free sulfur dioxide	−0.063
Sulphates	0.258	Chlorides	−0.124
Citric acid	0.241	Density	−0.175
Acidity stability index	0.159	Total sulfur dioxide	−0.183
Fixed acidity	0.122	Volatile acidity	−0.407
Residual sugar	0.022		

3.3. Effect of SQL-based feature engineering

Random forest was evaluated with the original 11 variables and with the two SQL-derived features under both the initial row-wise split and the stricter group-aware split. Model parameters and random_state were held constant.

Table 2. Random forest performance under row-wise and group-aware holdout splits

Split method	Feature set	Accuracy	Macro-F1	Weighted-F1
Row-wise	Original 11	0.7118	0.3454	0.6893
Row-wise	Original + SQL	0.7118	0.3415	0.6883
Group-aware	Original 11	0.5939	0.2882	0.5781
Group-aware	Original + SQL	0.5721	0.2713	0.5536

Under the row-wise split, accuracy was 0.7118 with both feature sets, while macro-F1 changed from 0.3454 to 0.3415 (Table 2). However, 43 of the 229 test records had identical training counterparts. After these observations were kept within groups, accuracy was 0.5939 with the

DOI: 10.54254/2755-2721/2026.BA36266

original features and 0.5721 with the full set; macro-F1 was 0.2882 and 0.2713. The initial result is therefore potentially optimistic, and the SQL-derived features did not improve performance under either split.

The lower group-aware result does not remove the database contribution. SQL-based audits identified these repeated observations. Database tables and views kept field names, validation rules, and feature-calculation logic consistent. Data governance and feature predictive value should therefore be evaluated separately.

Table 3. Results of repeated stratified group five-fold cross-validation and class weighting

Feature Set	Class Weight	Accuracy	Macro-F1	Weighted-F1
Original 11 Features	None	0.5984 ± 0.0300	0.2793 ± 0.0218	0.5776 ± 0.0315
Original 11 Features	Balanced	0.5627 ± 0.0286	0.3090 ± 0.0465	0.5578 ± 0.0284
Original + SQL-Derived Features	None	0.5906 ± 0.0292	0.2756 ± 0.0201	0.5708 ± 0.0294
Original + SQL-Derived Features	Balanced	0.5708 ± 0.0303	0.3011 ± 0.0356	0.5635 ± 0.0308

Table 3 reports repeated stratified group five-fold cross-validation. With the original features, random forest achieved 0.5984±0.0300 accuracy, 0.2793±0.0218 macro-F1, and 0.5776±0.0315 weighted-F1. The full feature set gave 0.5906±0.0292, 0.2756±0.0201, and 0.5708±0.0294. This supports the conclusion that the SQL-derived features did not add predictive information.

Balanced class weights increased macro-F1 from 0.2793 to 0.3090 with the original features, but accuracy fell from 0.5984 to 0.5627 and weighted-F1 from 0.5776 to 0.5578. The full feature set showed the same trade-off.

3.4. Evaluation metrics and random forest classification results

Accuracy can hide weak minority-class recognition, so macro-precision, macro-recall, macro-F1, and weighted-F1 were examined together [9].

Table 4. Overall performance of the random forest model

Metric	Value
Accuracy	0.5721
Macro-Precision	0.2910
Macro-Recall	0.2671
Macro-F1	0.2713
Weighted-F1	0.5536

As Table 4, random forest achieved 0.5721 accuracy, 0.2910 macro-precision, 0.2671 macro-recall, 0.2713 macro-F1, and 0.5536 weighted-F1. The large gap between macro-F1 and weighted-F1 shows that performance is concentrated in the larger classes.

DOI: 10.54254/2755-2721/2026.BA36266

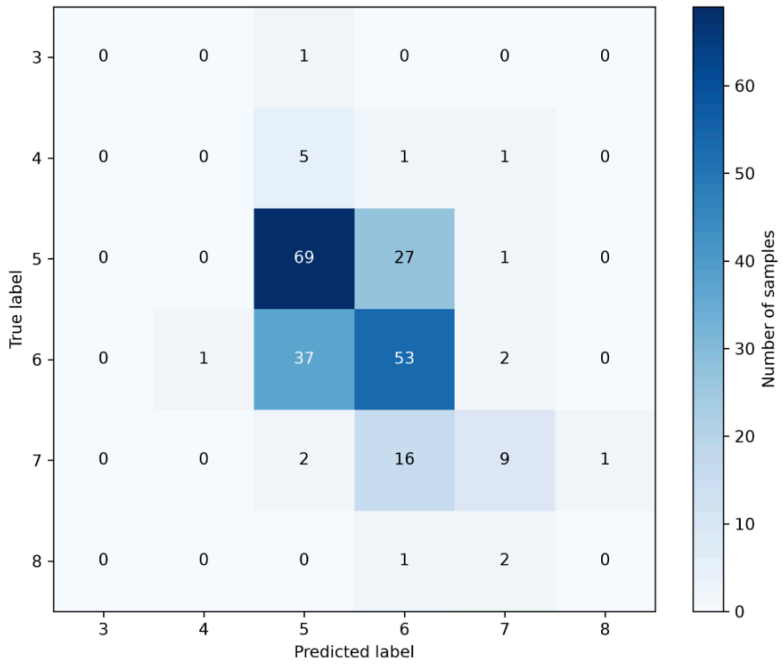

Figure 2. Confusion matrix for the random forest model (photo/picture credit: original)

Figure 2 shows that the model mainly recognized scores 5, 6, and 7; none of the samples scored 3, 4, or 8 were correctly classified. Most errors occurred between adjacent grades. Similar minority-class difficulties were reported by Chen [5].

3.5. Comparison of classification performance across models

Table 5 compares the four models on the same group-aware test set using accuracy, macro-F1, and weighted-F1.

Table 5. Accuracy, macro-F1, and weighted-F1 across models

Model	Accuracy	Macro-F1	Weighted-F1
Decision Tree	0.5459	0.3017	0.5435
Logistic Regression	0.5895	0.2609	0.5705
KNN	0.4498	0.2205	0.4398
Random Forest	0.5721	0.2713	0.5536

Logistic regression had the highest accuracy (0.5895) and weighted-F1 (0.5705), followed by random forest at 0.5721 and 0.5536. Decision tree produced the highest macro-F1 at 0.3017.

Random forest did not lead every metric, although it outperformed KNN and provided a nonlinear model for interpretation. Its macro-F1 of 0.2713 still indicates weak minority-class recognition.

3.6. Feature importance and model interpretation

Figure 3 presents impurity-based feature importance from the group-aware random forest.

DOI: 10.54254/2755-2721/2026.BA36266

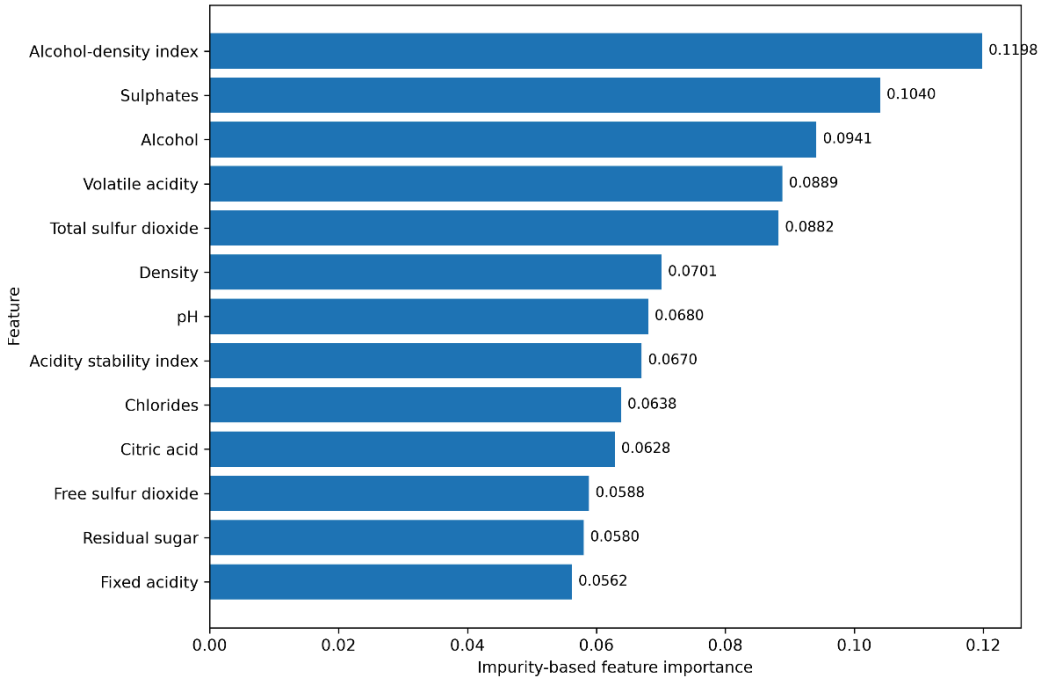

Figure 3. Random forest feature importance ranking (photo/picture credit: original)

The alcohol-density index ranked first at 0.1198. Its correlation with quality (0.484) is almost identical to alcohol (0.485), suggesting that it mainly reformulates existing information. Since it did not improve prediction, its high importance only shows frequent use by this model, not new information or causality. Feature rankings can also change with data processing and model choice [4].

4. Discussion and future work

The initial 0.7118 accuracy was obtained from a conventional row-wise split. The database audit later showed that 43 test records had identical training counterparts. Group-aware splitting reduced accuracy to 0.5721, and repeated group cross-validation gave 0.5906±0.0292 for the full feature set. The difference is not a deterioration of the trained algorithm; it shows that row-wise splitting gave an optimistic estimate when repeated observations crossed sets.

MySQL is used to standardize processing rather than directly raise accuracy. It preserves field definitions, supports checks for missing values, duplicate identifiers, repeated observations, and score ranges, and calculates both derived features through one view. This makes the cleaning rules and feature provenance easier to review and reuse. Adding the two derived features failed to improve either evaluation: row-wise macro-F1 changed from 0.3454 to 0.3415, while group-aware macro-F1 changed from 0.2882 to 0.2713. Class weighting improved macro-F1 but reduced accuracy and weighted-F1. SQL's value in this study is therefore process consistency and the detection of evaluation risk, not a guaranteed prediction gain.

Three limitations remain. First, the sample is small, imbalanced, and contains repeated profiles; grouping prevents cross-set overlap but leaves few independent minority examples [4]. Second, the two derived features are simple and largely overlap with existing variables; richer chemical features may be more useful [3]. Third, the data omit cultivation, climate, fermentation, and storage conditions, which may provide complementary information [10].

DOI: 10.54254/2755-2721/2026.BA36266

Future work should test resampling and cost-sensitive methods, design features from enological knowledge, evaluate independent production batches, and examine the ordered nature of quality scores. Feature-importance stability should also be checked across validation folds.

5. Conclusion

This paper combines MySQL data management, SQL feature construction, and random forest modeling. The initial row-wise split produced 0.7118 accuracy, but repeated profiles crossed the training-test boundary. Group-aware evaluation reduced the more reliable estimate to 0.5721 accuracy and 0.2713 macro-F1. The two derived features did not improve prediction, while balanced weights traded overall performance for a higher macro-F1. SQL nevertheless made the validation rules and feature calculations traceable and helped identify the optimistic initial result. Further work requires more independent minority-class samples and external data.

References

[1] Cortez, P., Cerdeira, A., Almeida, F., et al. (2009). Modeling wine preferences by data mining from physicochemical properties. *Decision Support Systems, 47*(4), 547–553. https://doi.org/10.1016/j.dss.2009.05.016

[2] Jain, K., Kaushik, K., Gupta, S. K., et al. (2023). Machine learning-based predictive modelling for the enhancement of wine quality. *Scientific Reports, 13*, Article 17042. https://doi.org/10.1038/s41598-023-44111-9

[3] Bhardwaj, P., Tiwari, P., Olejar, K., et al. (2022). A machine learning application in wine quality prediction. *Machine Learning with Applications, 8*, Article 100261. https://doi.org/10.1016/j.mlwa.2022.100261

[4] Zaza, S., Atemkeng, M., & Hamlomo, S. (2023). Wine feature importance and quality prediction: A comparative study of machine learning algorithms with unbalanced data. *arXiv*. https://arxiv.org/abs/2310.01584

[5] Chen, Z. (2025). Wine quality prediction with ensemble trees: A unified, leak-free comparative study. *arXiv*. https://arxiv.org/abs/2506.06327

[6] Yasser, H. M. (2022). Wine quality dataset [Data set]. *Kaggle*. https://www.kaggle.com/datasets/yasserh/wine-quality-dataset

[7] Breiman, L. (2001). Random forests. *Machine Learning, 45*(1), 5–32. https://doi.org/10.1023/A:1010933404324

[8] Pedregosa, F., Varoquaux, G., Gramfort, A., et al. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research, 12*, 2825–2830.

[9] He, H., & Garcia, E. A. (2009). Learning from imbalanced data. *IEEE Transactions on Knowledge and Data Engineering, 21*(9), 1263–1284. https://doi.org/10.1109/TKDE.2008.239

[10] Kulasiri, D., Somin, S., & Kumara Pathirannahalage, S. (2024). A machine learning pipeline for predicting Pinot Noir wine quality from viticulture data: Development and implementation. *Foods, 13*(19), Article 3091. https://doi.org/10.3390/foods13193091