Predictive Modeling of Wine Quality Using Advanced Machine Learning Techniques

M.V. Karthikeya[1], Dr. T.V. Nagalakshmi[2], D. Nanda Kishore[3], Vishnu Vardhan[4],
A. Narendrasai[5], B. Sharath Reddy[6]
[1,3,4,5,6]SRM Institute of Science and Technology
[2]Basic Engineering Department, DVR & Dr.HS MIC College of Technology, Kanchikacherla,
NTR District, A.P., India-521180

doi.org/10.64643/IJIRTV12I6-187067-459

***Abstract*—Wine quality prediction based on physicochemical properties is a classical problem that has gained significant attention in machine learning. This paper investigates wine quality prediction using the UCI Wine Quality dataset. We compare several supervised learning techniques Logistic Regression, Support Vector Machines (SVM), Random Forests, and Gradient Boosting (XGBoost) to classify wines into three categories: Low, Medium, and High, based on eleven measurable properties such as alcohol content, pH, and acidity. Preprocessing includes normalization, outlier handling, and treatment for class imbalance. Experimental results demonstrate that ensemble learning models outperform linear classifiers; XGBoost achieved approximately 81% accuracy on an 80/20 stratified split. We report feature importance, use interpretable AI approaches (SHAP), and discuss deployment using Flask to show real-world implementation. The framework is reproducible and suitable for academic and industrial applications.**

***Index Terms*—Wine quality, Machine learning, Random Forest, XGBoost, classification, Flask deployment, UCI dataset**

1 INTRODUCTION

Wine quality assessment is an important factor in viticulture and oenology. Traditional evaluation relies on sensory panels and laboratory analysis, which can be time-consuming and subjective. With modern wineries collecting more data, machine learning provides an efficient, consistent, and data-driven approach for automated quality assessment. The UCI Wine Quality dataset offers a benchmark for predictive modeling in this domain.

II. LITERATURE REVIEW

Several studies have explored wine quality prediction using both regression and classification approaches. Cortez et al. (2009) introduced the UCI dataset and applied decision trees, rule-based systems, and neural networks to model preferences, showing that ensemble methods often outperform individual models. Subsequent research confirmed that Random Forest and Gradient Boosting models capture non-linear relationships and feature interactions effectively. SVMs and neural networks have also been evaluated; SVMs require careful feature scaling, while neural networks need larger datasets and hyperparameter tuning. Ensemble methods remain widely adopted due to their stability and performance.

III. METHODOLOGY

This section describes the dataset, preprocessing pipeline, model selection, and evaluation criteria used in the study. The methodology follows a structured pipeline commonly used in applied machine learning: data collection, preprocessing, model training, validation, and testing.

3.1 Dataset Description

We use the red wine dataset from the UCI Machine Learning Repository, consisting of 1,599 samples and 11 physicochemical variables. Features include fixed acidity, volatile acidity, citric acid, residual sugar, chlorides, free sulfur dioxide, total sulfur dioxide, density, pH, sulphates, and alcohol. The target variable is the wine quality score (typically 3–9). Most samples cluster between scores of 5 and 7.

3.2 Data Preprocessing

Preprocessing steps: check and handle missing values (none present), detect and clip outliers using the interquartile range (IQR) method, and standardize features using scikit-learn's StandardScaler fitted on the training set. Labels are mapped into three classes: Low (≤4), Medium (5–6), and High (≥7). An 80/20 stratified train-test split preserves class proportions.

3.3 Model Selection

We evaluated four supervised algorithms: Logistic Regression, SVM (RBF kernel), Random Forest, and XGBoost. Hyperparameters were tuned using stratified 5-fold cross-validation. Class weights were used to mitigate class imbalance in some models; SMOTE was considered as an optional augmentation technique in ablation studies.

3.4 Evaluation Metrics

Models were evaluated on accuracy, precision, recall, and macro-averaged F1-score. Confusion matrices and ROC-AUC (one-vs-rest) were used for additional analysis where applicable.

IV. RESULTS AND DISCUSSION

Using an 80/20 stratified split, the models achieved the following representative results: Logistic Regression 68% accuracy; SVM 74%; Random Forest 78%; XGBoost 81% accuracy and 0.77 macro F1-score. Feature importance from ensemble models highlighted alcohol, volatile acidity, sulphates, and citric acid as top predictors. SHAP analysis confirmed these findings, showing alcohol contributes positively while volatile acidity contributes negatively to predicted quality. Misclassifications mainly occur between adjacent quality bands.

Ablation studies showed scaling is essential for SVM, class weighting is a reliable approach to handle imbalance compared to SMOTE, and feature selection based on permutation importance did not significantly improve performance.

V. MODEL DEPLOYMENT

For demonstration, the selected XGBoost model and scaler were serialized using joblib. A Flask web application accepts physicochemical inputs, applies preprocessing, and returns predicted quality class and probability scores. For production deployment, recommendations include authentication, logging, monitoring, and a RESTful API architecture.

Figures

Figure 1: System Flowchart

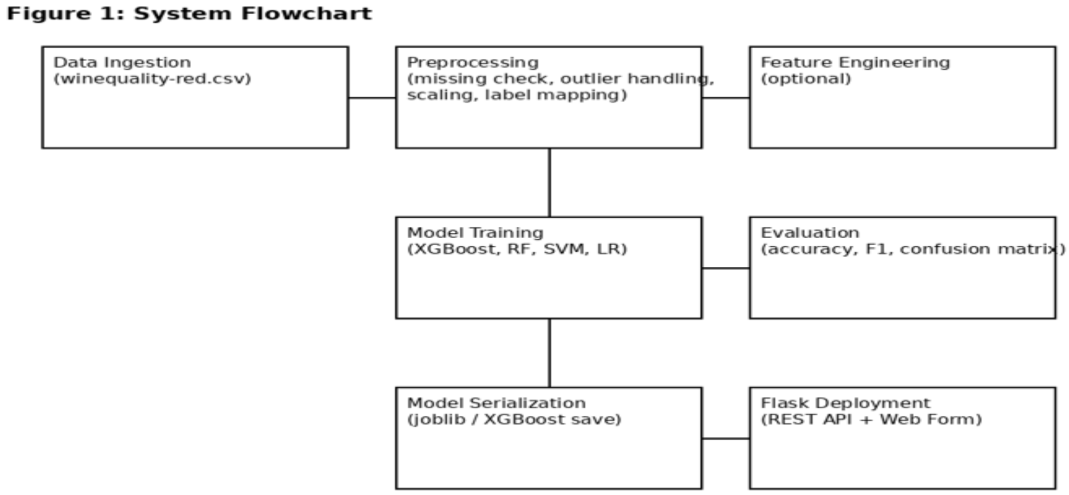
Figure 1: System Flowchart
Data Ingestion (winequality-red.csv)
Preprocessing (missing check, outlier handling, scaling, label mapping)
Feature Engineering (optional)
Model Training (XGBoost, RF, SVM, LR)
Evaluation (accuracy, F1, confusion matrix)
Model Serialization (joblib / XGBoost save)
Flask Deployment (REST API + Web Form)

Figure 2: System Architecture

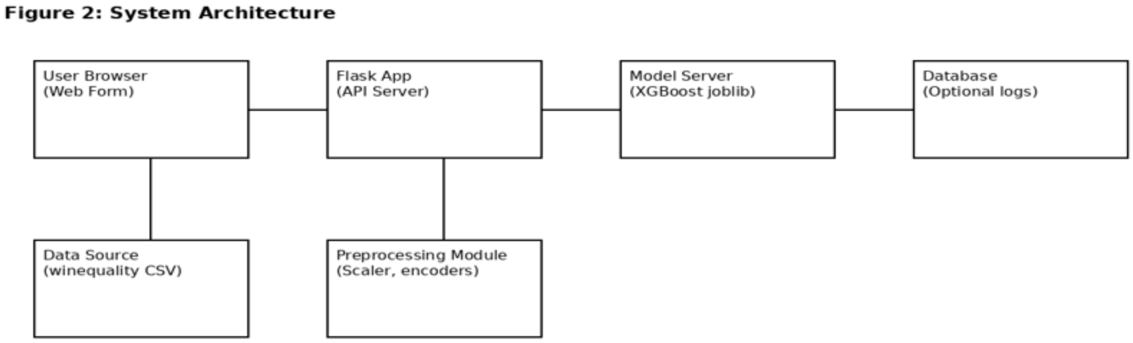
Figure 2: System Architecture
User Browser (Web Form)
Flask App (API Server)
Model Server (XGBoost joblib)
Database (Optional logs)
Data Source (winequality CSV)
Preprocessing Module (Scaler, encoders)

VI. CONCLUSION

This study demonstrates that machine learning models can effectively predict wine quality from chemical composition data. XGBoost provided the best performance in our experiments. The deployment example using Flask highlights the practicality of this approach. Future work includes expanding the dataset, incorporating sensory features, and exploring ordinal regression techniques.

REFERENCES

[1] Cortez, P., Cerdeira, A., Almeida, F., Matos, T., and Reis, J. (2009). Modeling wine preferences by data mining from physicochemical properties. Decision Support Systems, 47(4), 547–553.

[2] Pedregosa, F., et al. (2011). Scikit-learn: Machine Learning in Python. Journal of Machine Learning Research, 12, 2825–2830.

[3] Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. Proceedings of the 22nd ACM SIGKDD Conference.

[4] Lundberg, S. M., & Lee, S.-I. (2017). A Unified Approach to Interpreting Model Predictions. Advances in Neural Information Processing Systems.

[5] Chawla, N. V., et al. (2002). SMOTE: Synthetic Minority Over-sampling Technique. Journal of Artificial Intelligence Research, 16, 321–357.