WINE QUALITY PREDICTION USING MACHINE LEARNING

Mohammed Abdul Rahman | DA-DS | March

Abdul Rahman Mohammed

Developed: September 2025

---

1. Project Overview

This project explores multiple machine learning algorithms to predict the quality of wine based on its chemical properties. It includes: - Data analysis - Model building - Evaluation - Hyperparameter tuning for optimal performance

Objective: To compare the performance of various regression and classification algorithms and determine the best-performing model for predicting wine quality.

Algorithms Implemented: - Linear Regression - Logistic Regression - K-Nearest Neighbors (KNN) - Naive Bayes (Gaussian, Multinomial, Bernoulli) - Decision Tree Regressor - Support Vector Machine (SVM) - Hyperparameter Tuning using GridSearchCV

---

2. Libraries & Dataset

Python Libraries Used: - pandas, numpy, seaborn, matplotlib - sklearn (model_selection, metrics, preprocessing, linear_model, neighbors, naive_bayes, tree, svm)

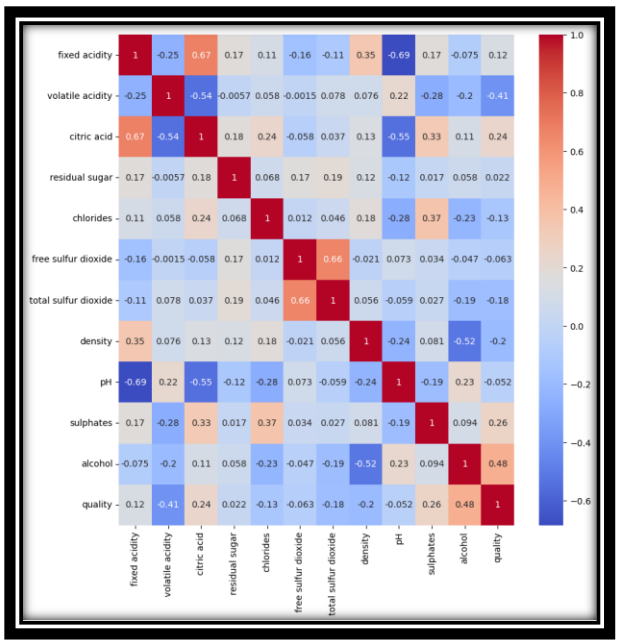
Dataset: Wine_Quality.csv - Total Rows: 1599 - Features: 12 numeric + 1 target (`quality`) - Target Variable: quality (score 0–10) - ID column removed before processing

---

3. Exploratory Data Analysis (EDA)

- Checked for missing values
- Rounded numeric columns for uniformity
- Identified correlations between features
- Strong correlation observed between alcohol, volatile acidity, and quality

Visualizations: - Correlation Heatmap



---

4. Machine Learning Models

4.1 Linear Regression

- Model used to predict wine quality
- **Performance Metrics:**
  - Mean Squared Error (MSE)
  - Root Mean Squared Error (RMSE)
  - Mean Absolute Error (MAE)
  - R2 Score

```
Mean Square Error:  0.37849868720036056
Root Mean Square Error:  0.6152224696809769
Mean Absolute Error:  0.47560665313054873
R2 Score:  0.3198255892168427
```

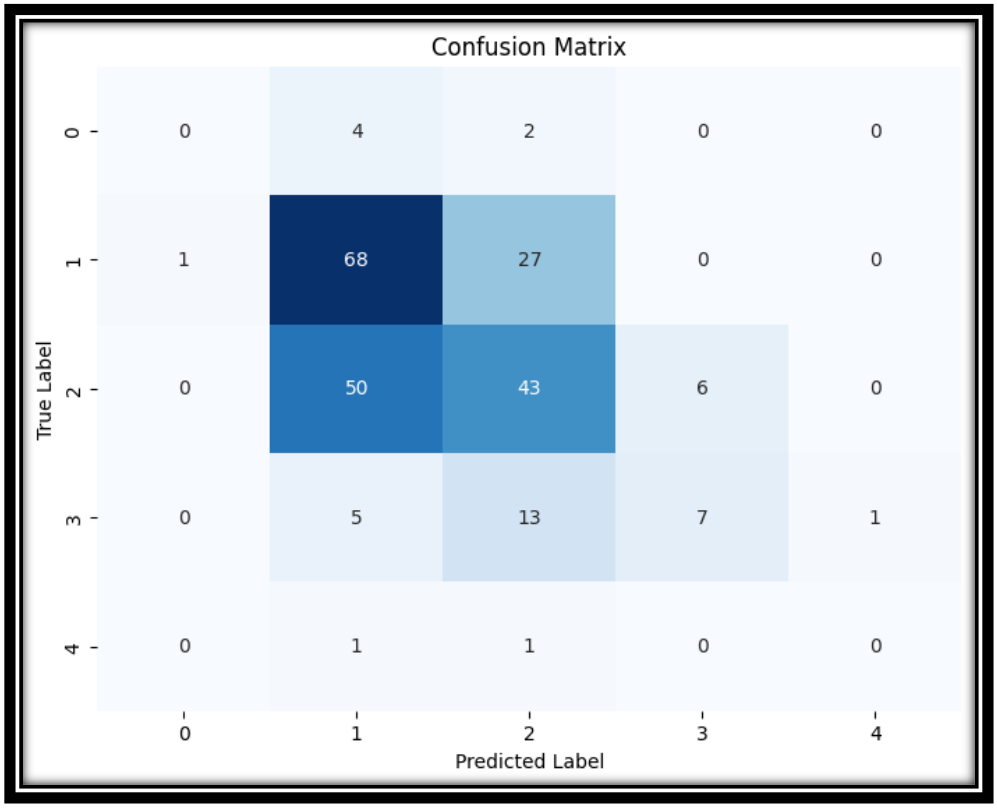
4.2 Logistic Regression

- Used for classification of wine quality categories
- **Evaluated via:**
  - Accuracy
  - Confusion Matrix
  - Classification Report

```
Accuracy Score: 0.6069868995633187

Confusion Matrix:
[[ 0  4  2  0  0]
 [ 0 73 22  1  0]
 [ 0 34 64  1  0]
 [ 0  3 21  2  0]
 [ 0  0  1  1  0]]

Classification Report:
              precision    recall  f1-score   support

           4       0.00      0.00      0.00         6
           5       0.64      0.76      0.70        96
           6       0.58      0.65      0.61        99
           7       0.40      0.08      0.13        26
           8       0.00      0.00      0.00         2

    accuracy                           0.61       229
   macro avg       0.32      0.30      0.29       229
weighted avg       0.57      0.61      0.57       229
```

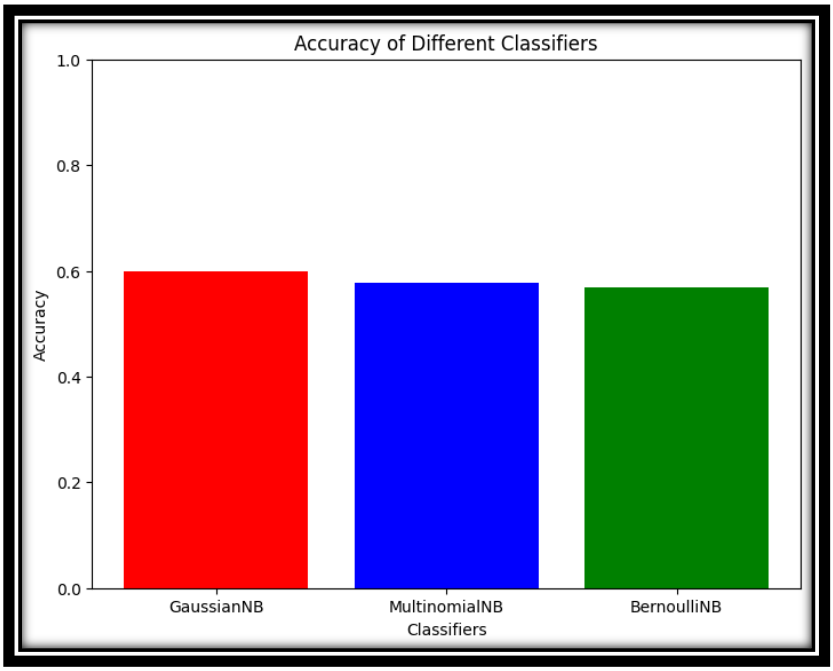
4.3 K-Nearest Neighbors (KNN)

- Distance-based classification
- Default n_neighbors=5
- Confusion matrix visualized



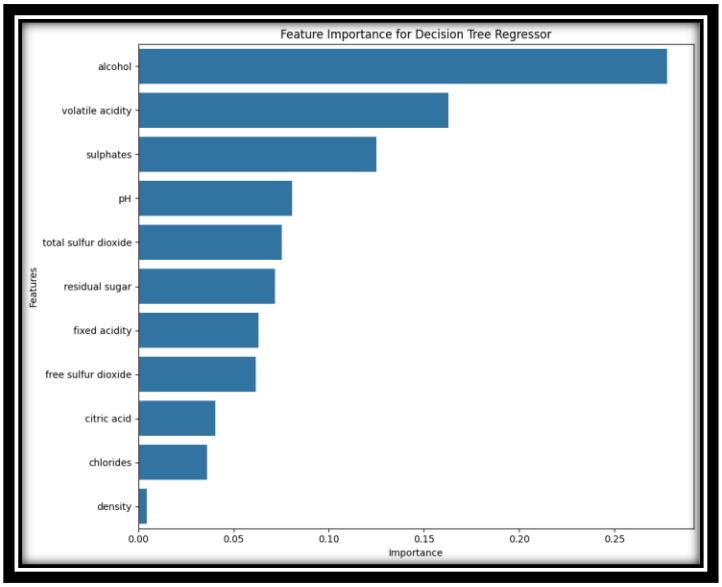
4.4 Naive Bayes Classifiers

- Implemented GaussianNB, MultinomialNB, BernoulliNB
- Standardization and preprocessing applied as needed
- Accuracy comparison visualized in a bar chart



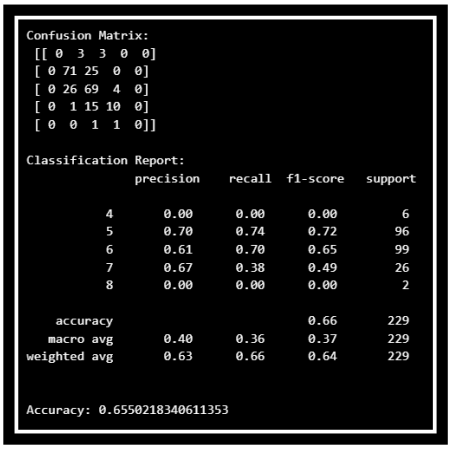
4.5 Decision Tree Regressor

- Predicted wine quality
- Feature importance analyzed and plotted
- Top features identified impacting wine quality



4.6 Support Vector Machine (SVM)

- Used SVC with RBF kernel
- Data scaled using StandardScaler
- Performance evaluated via confusion matrix and classification report

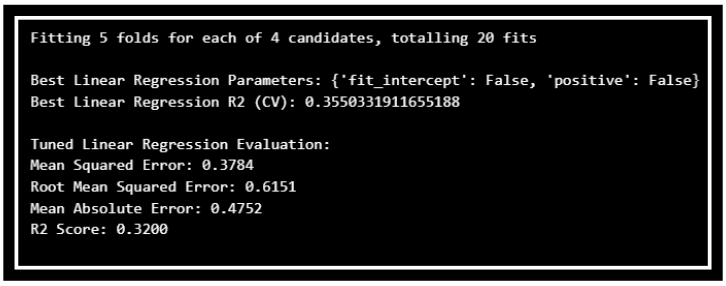


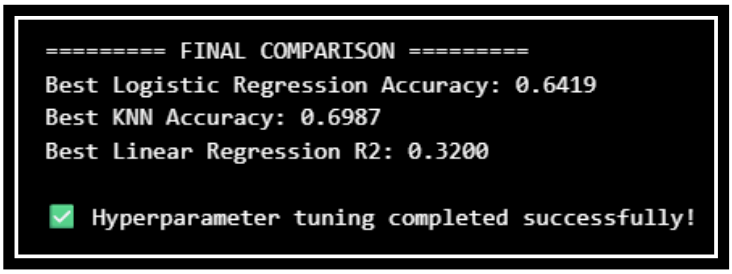
---

5. Hyperparameter Tuning

- Performed GridSearchCV for:
  - Logistic Regression (`C`, `solver`, `penalty`)
  - KNN (`n_neighbors`, `weights`, `metric`)
  - Linear Regression (`fit_intercept`, `positive`)
- 5-fold cross-validation applied

Best Parameters & Performance: - Logistic Regression: `best_params_`, CV accuracy `best_score_` - KNN: `best_params_`, CV accuracy `best_score_` - Linear Regression: `best_params_`, CV R2 `best_score_`





6. Final Model Comparison

Model	Metric	Score
Logistic Regression	Accuracy	0.6419
KNN	Accuracy	0.6987
Linear Regression	R2	0.3200

Conclusion: - Best performing model identified based on accuracy/R2 - Machine learning pipeline from data preprocessing → model building → evaluation → hyperparameter tuning successfully demonstrated

---

7. Future Enhancements

- Apply ensemble models like Random Forest, XGBoost
- Feature scaling & selection for improved performance
- Test on larger, more diverse datasets
- Deploy as web-based prediction application
