Ensemble Learning on the Red Wine Quality Dataset

This project applies ensemble machine learning methods to predict wine quality using physicochemical properties from the UCI Red Wine Quality dataset. The dataset is downloaded directly from Kaggle using kagglehub, and several ensemble models are trained and evaluated to compare their performance on this regression task.

Project Overview

The purpose of this project is to demonstrate the application of supervised learning and ensemble techniques on a real-world dataset. The workflow includes data acquisition, preprocessing, model development, evaluation, and result visualization.

Key Steps

- Dataset Acquisition: Retrieved automatically from Kaggle using the kagglehub library.

- Exploratory Data Analysis: Reviewed structure, summary statistics, missing values, and distribution of the target variable.

- Data Preprocessing:

- Removed missing values

- Performed outlier clipping using percentile thresholds

- Added a derived feature (sulfur_dioxide_ratio)

- Model Development: Implemented four ensemble models:

- Random Forest Regressor

- Gradient Boosting Regressor

- Bagging Regressor (with Random Forest as the base estimator)

- Voting Regressor

- Evaluation Metrics:

- Mean Squared Error (MSE)

- Root Mean Squared Error (RMSE)

- R^2 Score

- Visualizations:

- Model comparison bar charts

- Feature importance plot

- Distribution of wine quality values

Requirements

Install the required Python packages:

pip install kagglehub pandas numpy scikit-learn matplotlib

You must also have a Kaggle account and a valid kaggle.json API token stored at ~/.kaggle/kaggle.json. The notebook assumes authentication is already configured.

How to Run the Project

1. Download or clone this repository.

2. Open the notebook file (project3_red_wine_ensemble.ipynb).

3. Run all cells in order:

- The dataset will download automatically

- EDA, preprocessing, model training, and evaluation will run sequentially

4. Review the output tables and visualizations to compare model performance.

The notebook is compatible with Jupyter Notebook, VS Code, and Google Colab.

Results Summary

The ensemble models showed the following general performance trends:

- Random Forest and Gradient Boosting produced the strongest results, achieving the lowest RMSE and highest R^2 values.

- The Voting Regressor performed well but did not consistently outperform its strongest individual components.

- The Bagging Regressor was stable but generally weaker than boosting-based methods.

- Feature importance highlighted key predictors such as alcohol content, volatile acidity, and sulfur dioxide-related features.

Tree-based ensemble methods performed effectively on this dataset due to their ability to model nonlinear relationships and feature interactions.

Rationale for Using Random Forest for Feature Importance

Random Forest includes a built-in and reliable method for estimating feature importance. Because it averages importance scores across many decision trees, it produces stable and interpretable results. Other ensemble methods used in this project, including Gradient Boosting, Bagging, and Voting Regressors, either do not provide consistent feature importance measures or do not support them at all. Therefore, Random Forest was the most appropriate choice for visualizing feature importance.

Possible Extensions

This project can be expanded by:

- Performing hyperparameter tuning with GridSearchCV or RandomizedSearchCV
- Adding additional engineered features
- Testing advanced ensemble methods such as XGBoost, LightGBM, or CatBoost
- Reframing the task as a classification problem by grouping quality ratings into categories