Project — Wine Quality Prediction (Red wine)

Task: Predict wine quality (score 0–10) from physicochemical tests. Do both a **regression** (predict the numeric quality) and a **classification** (label wines as good/bad).

Dataset (official): Wine Quality dataset (red wine). The dataset includes 11 physicochemical features and a quality score (0–10). Use the red wine CSV for this exercise.

Learning goals

- EDA and visualization
- Cleaning & preprocessing (outliers, missing values)
- Regression vs classification problem framing
- Training multiple models and cross-validation
- Model evaluation with appropriate metrics (MAE/RMSE/R^2 for regression; accuracy/precision/recall/F1/ROC AUC for classification)
- Feature importance and interpretation
- Packaging model (joblib) and short report

Step-by-step project guide (what you should do)

1. **Download dataset**
2. **Problem definition**
    - Regression: predict quality (numeric).
    - Classification: define good wine as quality >= 7 (or choose another threshold) → binary target.
3. **Exploratory Data Analysis (EDA)**
    - Show head(), dtypes, value counts of quality.
    - Distributions, boxplots for numeric features, correlation matrix (heatmap).

- Check for missing values and outliers.

4. **Preprocessing**
    - No categorical features here — all features numeric.
    - Handle missing values (if any).
    - Optionally standardize features (StandardScaler) — especially required for models like LogisticRegression / SVM.
    - Optionally apply log transform to skewed features.

5. **Train/test split**
    - Use train_test_split (e.g. 80/20) or 5-fold cross-validation.

6. **Baselines**
    - Regression baseline: predict mean quality.
    - Classification baseline: predict majority class.

7. **Models to train**
    - Regression: LinearRegression, RandomForestRegressor, GradientBoostingRegressor (or XGBoost/LightGBM).
    - Classification: LogisticRegression, RandomForestClassifier, GradientBoostingClassifier.

8. **Hyperparameter tuning**
    - Use GridSearchCV or RandomizedSearchCV with cross-validation.

9. **Evaluation**
    - Regression: MAE, RMSE, R^2 on test set.

- Classification: Accuracy, Precision, Recall, F1, ROC AUC, confusion matrix; show calibration if needed.

10. **Interpretation**
    - Feature importances for tree models.
    - Coefficients for linear models.
    - Optionally: SHAP values or partial dependence plots.

11. **Report & presentation**
    - Summarize EDA, chosen preprocessing, best models, metrics, and a paragraph on limitations (e.g., dataset size, imbalanced quality classes).

12. **(Optional) Deployment**
    - Create a small Streamlit app allowing users to input physicochemical values and get a prediction.

Grading / Comparison checklist

- EDA completeness: histograms, correlation analysis, and clear description — 20 pts
- Preprocessing & justification (scaling, imputation) — 20 pts
- Models trained & hyperparameter attempts — 20 pts
- Correct evaluation & clear reporting of metrics — 20 pts
- Interpretation (feature importances) & discussion of limitations — 15 pts
- (Optional) Deployment or Streamlit demo — 5–10 bonus pts