WineGuard AI: Machine Learning for Wine Quality Prediction

Syed Armghan Ahmad

May 2025

1 Introduction

WineGuard AI is an end-to-end machine learning pipeline that predicts red wine quality based on physicochemical features, such as acidity and alcohol content. Utilizing the UCI Wine Quality dataset, it features a modular architecture, MLOps integration with MLflow and DagsHub, and a Flask web application for user predictions. Hosted at `https://github.com/SyedArmghanAhmad/End-to-End-Machine-Learning-Project-with-MLOps-MLflow-`, WineGuard AI offers a production-ready solution for winemaking quality control.

2 The Challenge

Winemaking relies on consistent quality assessment, but manual tasting is subjective and time-consuming. The challenge was to automate wine quality prediction (scores 0–10) using 11 measurable features, enabling wineries to enhance efficiency. The solution required a scalable pipeline, MLOps for experiment tracking, and a user-friendly interface.

3 Our Solution

WineGuard AI implements a five-stage pipeline:

1. **Data Ingestion**: Downloads and extracts the UCI Wine Quality dataset.
2. **Data Validation**: Verifies dataset schema against `schema.yaml`.
3. **Data Transformation**: Splits data into training (75%) and testing (25%) sets.
4. **Model Training**: Trains an ElasticNet regression model.
5. **Model Evaluation**: Logs metrics to MLflow and DagsHub.

A Flask app delivers predictions, and Docker ensures deployment flexibility.

4 Implementation

The modular `src/mlProject/` structure includes:

- `components/data_ingestion.py`: Downloads `winequality-red.csv` using `urllib.request` and extracts it with `zipfile`.
- `components/data_validation.py`: Ensures columns match `schema.yaml`.
- `components/data_transformation.py`: Splits data using `sklearn.model_selection.train_test_`
- `components/model_trainer.py`: Trains ElasticNet, saves with `joblib`.
- `components/model_evaluation.py`: Logs RMSE, MAE, R^2 to MLflow.
- `pipeline/*.py`: Executes stages (e.g., `stage_01_data_ingestion.py`).
- `config/configuration.py`: Loads `config.yaml` with `ConfigurationManager`.
- `entity/config_entity.py`: Defines dataclasses (e.g., `DataIngestionConfig`).
- `utils/common.py`: Utilities for YAML and file operations.
- `app.py`: Flask app for predictions via `pipeline/prediction.py`.
- `config.yaml`: Pipeline settings.

Dataclasses ensure structured configs, and `setup.py` supports package installation.

5 Results

The ElasticNet model achieved the following test set performance, logged to MLflow:

Metric	**Value**
RMSE	0.720
MAE	0.567
R^2	0.233

Table 1: WineGuard AI performance metrics.

6 Future Work

Future improvements include:

- Advanced models (e.g., XGBoost) to improve R^2.
- Feature engineering (e.g., scaling).

- AWS EC2 deployment.
- Fixing a `data_validation.py` bug (status file overwriting).
- Expanding `test.py` with unit tests.

7 Contact Information

- **LinkedIn**: https://pk.linkedin.com/in/armaghan-ahmad
- **GitHub**: https://github.com/SyedArmghanAhmad

8 Conclusion

WineGuard AI delivers a robust ML pipeline for wine quality prediction, with MLOps integration and a Flask interface. Its modular design ensures scalability and production readiness, with potential for further optimization.