Linear Regression Analysis Report

Predicting Wine Quality Using OLS and SGD

Contents

Project and Dataset Selection

For this project, I selected the Wine Quality dataset from the UCI Machine Learning Repository. The dataset combines two related datasets on red and white variants of Portuguese “Vinho Verde” wine. It contains 6,497 total observations, with 4,898 white wines and 1,599 red wines.

Each observation includes 11 physicochemical attributes (such as acidity, residual sugar, chlorides, sulphates, alcohol, etc.), along with the categorical attribute color (red or white). The wines are evaluated using the target variable quality, which is a sensory score assigned by human tasters on a scale from 0 to 10 (in practice, most ratings fall between 3 and 9).

The goal of this project is to build regression models that predict wine quality from its chemical and categorical attributes.

Regression Model Building

Pre-Processing

The dataset was loaded into a Pandas DataFrame by combining both red and white wine files provided in the UCI repository. An additional column, *color*, was added to differentiate between red and white wines when the two datasets were merged. A new DataFrame, *wine,* was created, containing both datasets of red and white wine.

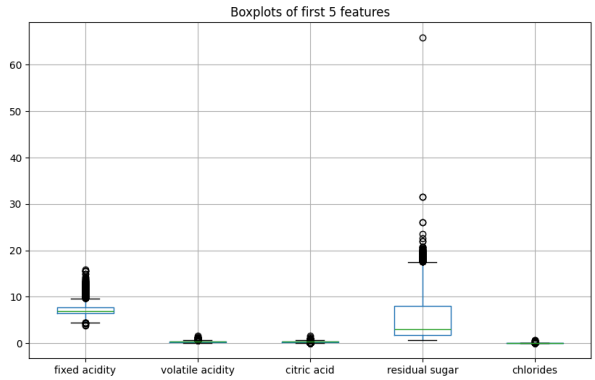
To ensure data consistency, I first checked for any null values using both .isna().sum() and .info(). Since .isna().sum() returned all zeroes, and in .info() every column’s Non-Null Count equals the number of rows, both methods confirmed that there were no null values or missing values in the dataset.

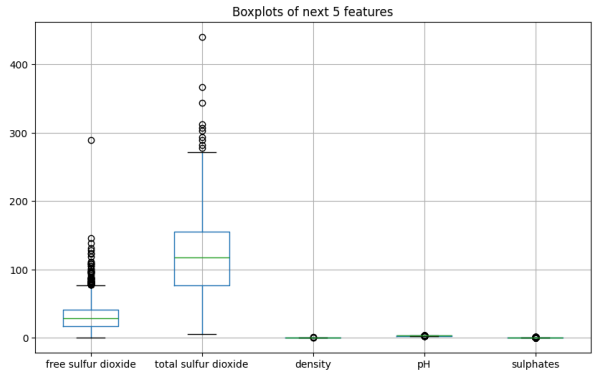
I then checked for any inconsistent data using .describe(). This showed that ranges are mostly fine. However, there were some suspicious extremes:

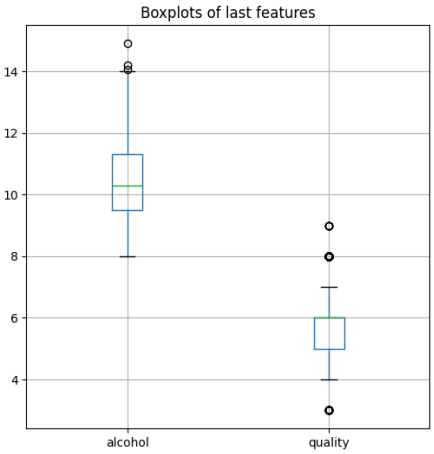
- residual sugar: max = 65.8, likely a dessert wine or an outlier.
- chlorides: max = 0.611, unusually salty.
- free SO2: up to 289, also up to 440. Possible outliers or measurement differences.
- sulphates: max = 2.0, high compared to the norm.

These are extreme values that can be potential outliers or data quality issues.

Next, I visualized the features using boxplots. Boxplots revealed many observations outside the whiskers, especially in residual sugar, free sulfur dioxide, total sulfur dioxide, sulphates, and fixed acidity. These may represent valid extreme cases (like sweet wines or wines with high preservatives) but could also indicate outliers worth further analysis. The variables with many outliers will be flagged as potential candidates for removal, and we will compare model performance with and without them.





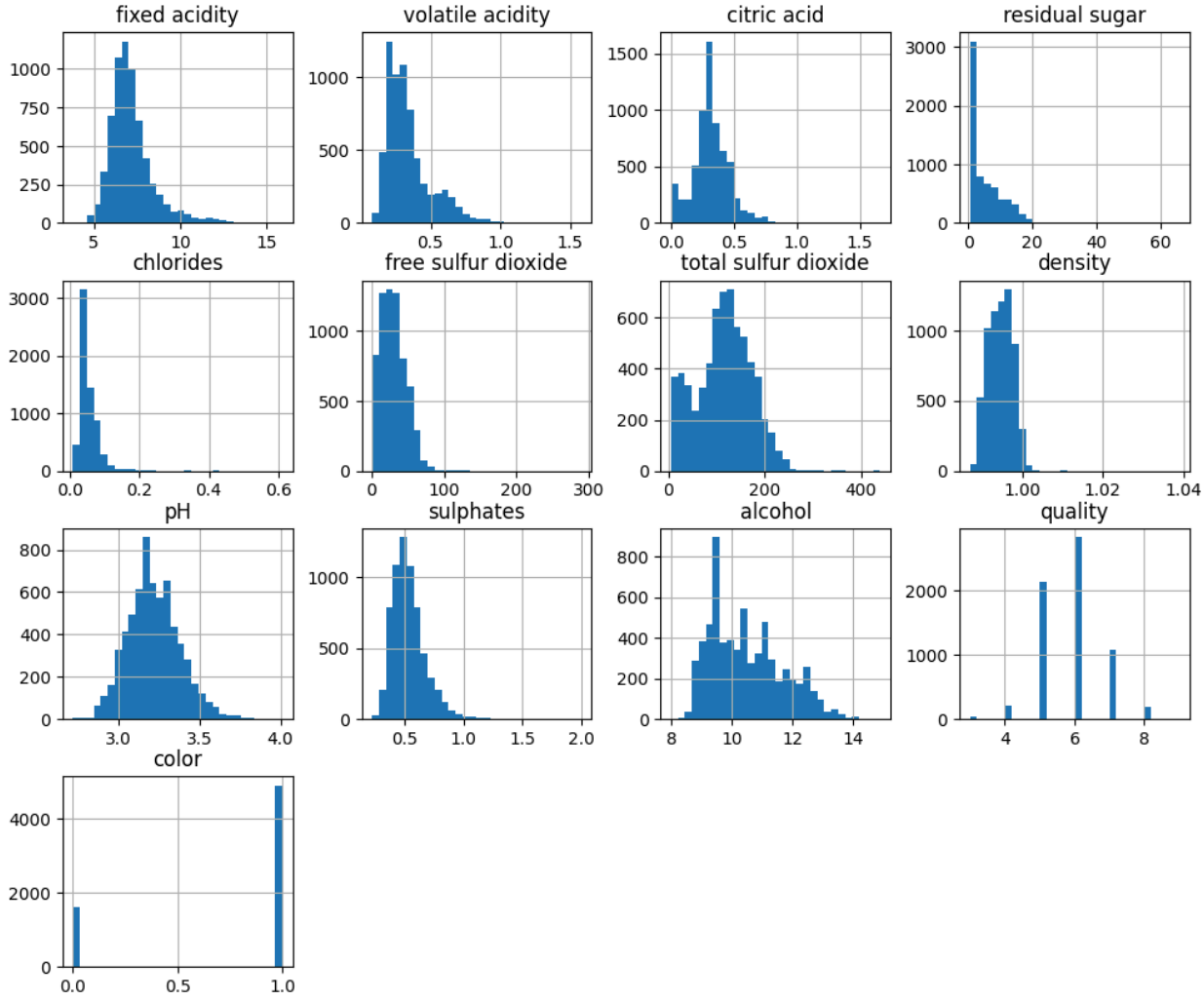


After this, I examined each attribute and the target variable. The dataset contains 11 attributes describing the composition of wines, plus one categorical attribute *color* (red or white), and the target variable *quality*.

- fixed acidity: concentration of non-volatile acids
- volatile acidity: acetic acid content; high values lead to vinegar taste
- citric acid: adds freshness to wine, higher values may indicate fruity character
- residual sugar: amount of sugar left after fermentation. Higher values -> sweeter wine
- chlorides: salt content in the wine
- free sulfur dioxide: SO2 not bound to other compounds; acts as an antimicrobial
- total sulfur dioxide: total SO_2 (free + bound), used as a preservative
- density: density of the wine, related to sugar and alcohol content
- pH: measure of acidity/basicity of the wine
- sulphates: potassium sulphate level, can contribute to wine preservation and flavor
- alcohol: percentage of alcohol by volume
- color: categorical feature (red = 0, white = 1
- quality (target variable): sensory score assigned by human tasters, on a scale from 0–10 (in this dataset, mostly 3–9)

The only categorical feature was *color*. This was converted into numeric form using label encoding, where red = 0 and white = 1. All other attributes were already numeric and required no conversion.

Then, to examine the distribution of the attributes, I generated histograms for each feature and also calculated their skewness values using the .skew() function. The plots showed that most variables are not normally distributed. Several, such as residual sugar, sulphates, chlorides, and the acidity measures, displayed heavy right skew, with a concentration of wines at lower values and a long tail of extreme higher values. Others, such as pH, density, and alcohol, appeared closer to symmetric. The target variable quality is discrete (scored 3–9), so it does not follow a normal distribution.



Then, to quantify these observations, I calculated skewness values using the .skew() function. Most attributes had positive skewness, confirming the right-skewed shape seen in the histograms. Features like residual sugar, sulphates, and chlorides had especially high skew values, while pH, density, and alcohol were closer to zero, indicating more balanced distributions.

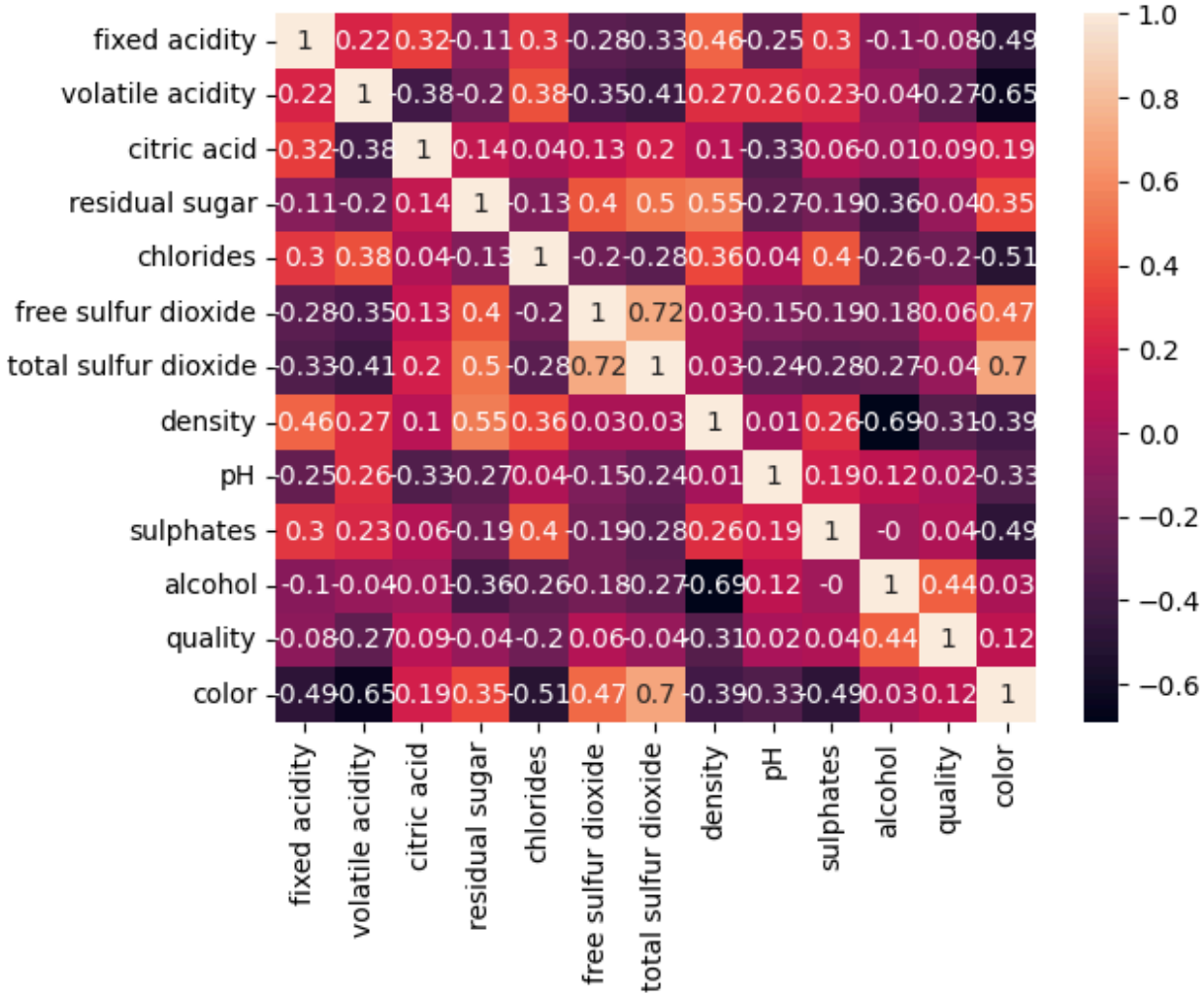
	0
fixed acidity	1.723290
volatile acidity	1.495097
citric acid	0.471731
residual sugar	1.435404
chlorides	5.399828
free sulfur dioxide	1.220066
total sulfur dioxide	-0.001177
density	0.503602
pH	0.386839
sulphates	1.797270
alcohol	0.565718
quality	0.189623
color	-1.179095

The skewness values confirmed the patterns observed in the histograms. Attributes such as residual sugar, sulphates, and chlorides had high positive skewness, reinforcing the visual impression of long right tails and extreme outliers. In contrast, features like pH, density, and alcohol showed skewness values closer to zero, supporting their more balanced, symmetric distributions. This numerical analysis validates the visual findings and provides a quantitative measure of the degree of skew in each attribute.

Next, both standardization and normalization were applied to the wine dataset.

- Standardization transformed each attribute to have mean 0 and standard deviation 1, ensuring all features contribute equally regardless of their original scale.
- Normalization then scaled each attribute into the [0,1] range, which is especially useful for models that rely on distance measures or require bounded inputs. While neither transformation changes the underlying distribution (attributes remain skewed and outliers remain present), these preprocessing steps make the dataset more suitable for machine learning by placing all features on comparable scales.

To examine the relationships among the variables, I calculated the correlation matrix with all the attributes and visualized it using a heatmap. This provided both a numerical and visual overview of how features relate to each other and to the target variable *quality.*



Correlation analysis shows that the strongest positive correlation with wine quality is alcohol content, suggesting that higher alcohol wines are generally rated higher. Volatile acidity, density, and chlorides are negatively correlated with quality, meaning higher values in these attributes are associated with lower rated wines. Other features, such as citric acid, pH, residual sugar, and sulphates, exhibit only weak correlations with quality, indicating limited direct influence. Overall, alcohol content appears to be the most important attribute in predicting wine quality.

Model Construction

Baseline SGDRegressor Model

After selecting alcohol, volatile acidity, and density as the top predictors, I split the data into training (80%) and testing (20%) sets. I then trained a baseline SGDRegressor with default settings (1000 iterations, tolerance of 1e-3). The model learned positive weights for alcohol and density, and a negative weight for volatile acidity, which matches the earlier correlation analysis. The intercept was about 5.81, and the test R^2 score was roughly 0.25, meaning the model explained about 25% of the variation in wine quality.

Hyperparameter Tuning with SGDRegressor

To improve the baseline model, I tuned the main hyperparameters of the SGDRegressor: the loss function, regularization penalty, number of iterations, and learning rate schedule (with different initial learning rates). I tracked performance using MSE, MAE, and R^2 on the test set. For the loss function, the standard squared error worked best. Switching to the Huber loss actually hurt performance, dropping the R^2 to around 0.23. This suggests that outliers were not a big issue in this dataset.

For the regularization penalty, I tried L1, L2, and ElasticNet. All three gave nearly the same results, with R^2 values hovering around 0.25, so the choice of penalty didn't really matter here. When I increased the maximum iterations from 1000 to 5000, the results didn't change. That showed the model was already converging well before 5000 passes.

The biggest differences came from the learning rate schedule and eta0. A very small initial learning rate (eta0 = 0.001) combined with the "invscaling" schedule gave the best result, with an R^2 of about 0.254. The "adaptive" schedule was also steady across different eta0 values, while "constant" with a high eta0 (0.1) performed the worst, dropping the R^2 to around 0.19. This makes sense because too high of a step size causes the model to overshoot instead of gradually converging.

In the end, the best combination was:
squared_error loss, L2 penalty, 2000 iterations, and an "invscaling" learning rate with eta0 = 0.001. This setup gave me the best R^2 (≈0.254), which is only a small improvement over the baseline.

Overall, the tuning showed that the model is most sensitive to the learning rate settings. The relatively low R^2 values also highlight that the chemical features in the dataset can only explain part of wine quality, and other factors (like human taste preferences) likely play an important role.

Ordinary Least Squares Regression

I trained an OLS regression using alcohol, volatile acidity, and density as predictors of wine quality. The model achieved an R^2 of 0.271 and an adjusted R^2 of 0.270, meaning it explained about 27% of the variation in wine quality—slightly higher than the tuned SGD model. All three predictors were statistically significant ($p < 0.001$). Alcohol had the strongest positive effect on quality (coef = 0.468), while volatile acidity had a significant negative effect (coef = -0.250). Density also showed a smaller but positive effect (coef = 0.122). The F-statistic (642.8, $p < 0.001$) confirmed that the model as a whole was highly significant.

Comparison of Models

Both models captured similar patterns in the data, with alcohol as the strongest positive predictor of quality and volatile acidity as a negative one. The SGDRegressor, after tuning, achieved an R^2 of about 0.254, while the OLS regression performed slightly better with an R^2 of 0.271. The difference is small, but OLS has the advantage of providing detailed statistical diagnostics such as p-values and confidence intervals, which confirmed the significance of all three predictors. In contrast, SGD required careful tuning of hyperparameters (learning rate, iterations, and penalty type) to achieve reasonable performance. Overall, both models highlight the same key factors driving wine quality, but OLS offers more interpretability, while SGD demonstrates flexibility and scalability for larger datasets.

Conclusion

This project demonstrated how regression techniques can be applied to predict wine quality using physicochemical properties. Both SGDRegressor and OLS produced similar results, explaining around one-quarter of the variation in wine quality. Alcohol consistently emerged as the strongest positive predictor, while volatile acidity negatively affected quality, and density contributed more modestly. Although the models leave much variance unexplained (likely due to the subjective nature of human taste), they provide valuable insight into which measurable features influence wine ratings.