NTNU

Kunnskap for en bedre verden

Department of Computer Science

IDATG2208 - Introduction to Machine Learning

Assignment - 1

Aren't you guys also getting tired of this project template haha

Author:
Mustafa K.

September 2025

ALL CODE FROM THESE TASKS CAN BE FOUND AT:

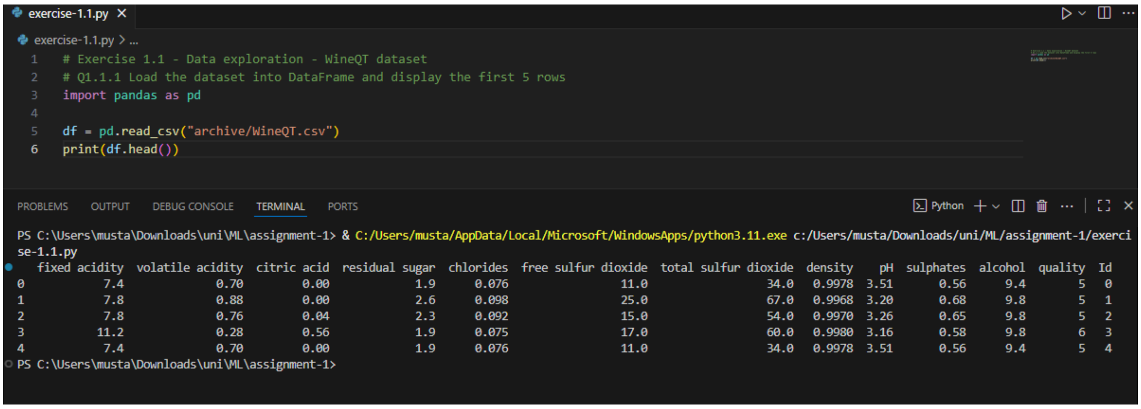
https://github.com/MustafaKess/IDATG2208-ML-assignment-1

1 Exercise 1

1.1 Q1.1 Data Exploration

1.1.1 Q1.1.1 Load the dataset into a DataFrame and display the first 5 rows. Print the dataset information and summary statistics.

Simple enough. Load the dataset using the pandas dataframe library. Create a variable by calling pandas' `read_csv` function.



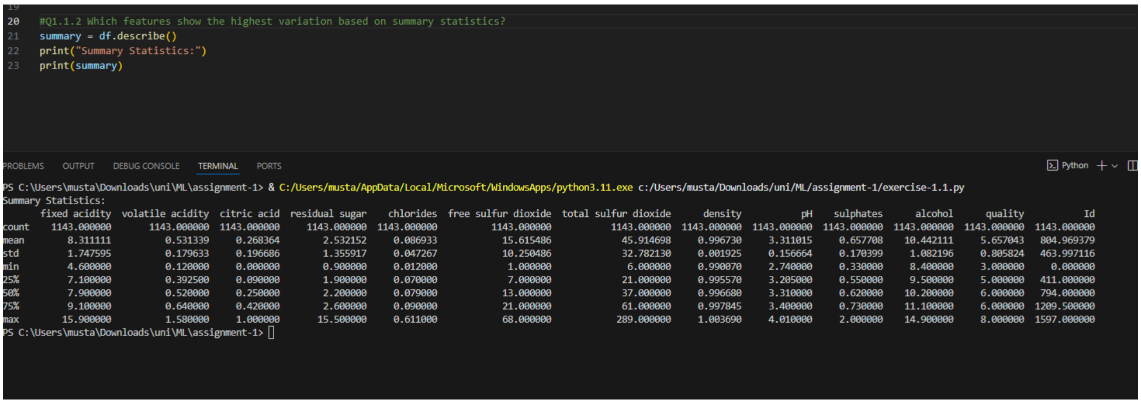
```python
# Exercise 1.1 - Data exploration - WineQT dataset
# Q1.1.1 Load the dataset into DataFrame and display the first 5 rows
import pandas as pd

df = pd.read_csv("archive/WineQT.csv")
print(df.head())
```

```
PS C:\Users\musta\Downloads\uni\ML\assignment-1> & C:/Users/musta/AppData/Local/Microsoft/WindowsApps/python3.11.exe c:/Users/musta/Downloads/uni/ML/assignment-1/exercise-1.1.py
   fixed acidity  volatile acidity  citric acid  residual sugar  chlorides  free sulfur dioxide  total sulfur dioxide  density    pH  sulphates  alcohol  quality  Id
0            7.4              0.70         0.00             1.9      0.076                 11.0                  34.0   0.9978  3.51       0.56      9.4        5   0
1            7.8              0.88         0.00             2.6      0.098                 25.0                  67.0   0.9968  3.20       0.68      9.8        5   1
2            7.8              0.76         0.04             2.3      0.092                 15.0                  54.0   0.9970  3.26       0.65      9.8        5   2
3           11.2              0.28         0.56             1.9      0.075                 17.0                  60.0   0.9980  3.16       0.58      9.8        6   3
4            7.4              0.70         0.00             1.9      0.076                 11.0                  34.0   0.9978  3.51       0.56      9.4        5   4
PS C:\Users\musta\Downloads\uni\ML\assignment-1>
```

1.1.2 Q1.1.2 Which features show the highest variation based on summary statistics?

From the same variable from Q.1.1.1 use `.describe` on the variable from the last task to show summary statistics.



```python
#Q1.1.2 Which features show the highest variation based on summary statistics?
summary = df.describe()
print("Summary Statistics:")
print(summary)
```

```
PS C:\Users\musta\Downloads\uni\ML\assignment-1> & C:/Users/musta/AppData/Local/Microsoft/WindowsApps/python3.11.exe c:/Users/musta/Downloads/uni/ML/assignment-1/exercise-1.1.py
Summary Statistics:
       fixed acidity  volatile acidity  citric acid  residual sugar    chlorides  free sulfur dioxide  total sulfur dioxide      density           pH    sulphates      alcohol      quality           Id
count    1143.000000       1143.000000  1143.000000     1143.000000  1143.000000          1143.000000           1143.000000  1143.000000  1143.000000  1143.000000  1143.000000  1143.000000  1143.000000
mean        8.311111          0.531339     0.268364        2.532152     0.086933            15.615486             45.914698     0.996730     3.311015     0.657708    10.442111     5.657043   804.969379
std         1.747595          0.179633     0.196686        1.355917     0.047267            10.250486             32.782130     0.001925     0.156664     0.170399     1.082196     0.805824   463.997116
min         4.600000          0.120000     0.000000        0.900000     0.012000             1.000000              6.000000     0.990070     2.740000     0.330000     8.400000     3.000000     0.000000
25%         7.100000          0.392500     0.090000        1.900000     0.070000             7.000000             21.000000     0.995570     3.205000     0.550000     9.500000     5.000000   411.000000
50%         7.900000          0.520000     0.250000        2.200000     0.079000            13.000000             37.000000     0.996680     3.310000     0.620000    10.200000     6.000000   794.000000
75%         9.100000          0.640000     0.420000        2.600000     0.090000            21.000000             61.000000     0.997845     3.400000     0.730000    11.100000     6.000000  1209.500000
max        15.900000          1.580000     1.000000       15.500000     0.611000            68.000000            289.000000     1.003690     4.010000     2.000000    14.900000     8.000000  1597.000000
PS C:\Users\musta\Downloads\uni\ML\assignment-1> 
```

By looking at the standard deviation section (shortend to "std", very unfortunate to not use SD) we can see the following data

The features with the highest deviation are total sulfur dioxide with 32.78 and free sulfur dioxide with 10.25

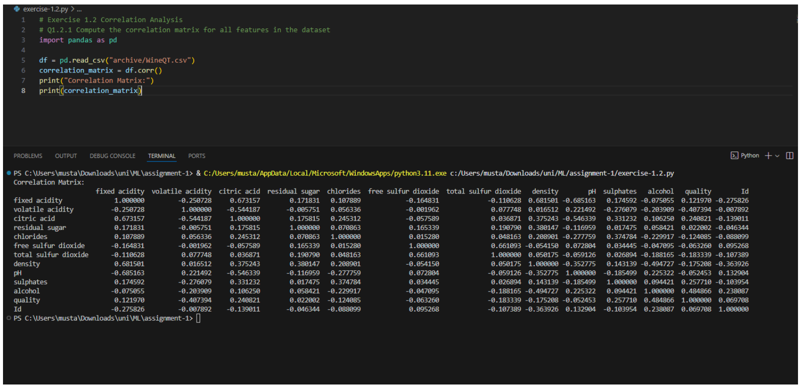
The ID part should be ignored. If we look at figure from task Q1.1.1 we can see that the ID is just to index the entries in the dataset, and not a quality from the wine

Feature	Standard Deviation (std)
fixed acidity	1.747595
volatile acidity	0.179633
citric acid	0.196686
residual sugar	1.355917
chlorides	0.047267
free sulfur dioxide	10.250486
total sulfur dioxide	32.782130
density	0.001925
pH	0.156664
sulphates	0.170399
alcohol	1.082196
quality	0.805824
Id	463.997116

1.2 Q1.2 Correlation Analysis

1.2.1 Q1.2.1 Compute the correlation matrix of all features.

This can be done with the use of the`.corr()` function



```python
# Exercise 1.2 Correlation Analysis
# Q1.2.1 Compute the correlation matrix for all features in the dataset
import pandas as pd

df = pd.read_csv("archive/WineQT.csv")
correlation_matrix = df.corr()
print("Correlation Matrix:")
print(correlation_matrix)
```

```
PS C:\Users\musta\Downloads\uni\ML\assignment-1> & C:/Users/musta/AppData/Local/Microsoft/WindowsApps/python3.11.exe c:/Users/musta/Downloads/uni/ML/assignment-1/exercise-1.2.py
Correlation Matrix:
                      fixed acidity  volatile acidity  citric acid  residual sugar  chlorides  free sulfur dioxide  total sulfur dioxide    density         pH  sulphates    alcohol    quality         Id
fixed acidity              1.000000         -0.250728     0.673157        0.171831   0.107889            -0.164831             -0.110628   0.681501  -0.685163   0.174592  -0.075055   0.121970  -0.275826
volatile acidity          -0.250728          1.000000    -0.544187       -0.005751   0.056336            -0.001962              0.077748   0.016512   0.221492  -0.276079  -0.203909  -0.407394  -0.007892
citric acid                0.673157         -0.544187     1.000000        0.175815   0.245312            -0.057589              0.036871   0.375243  -0.546339   0.331232   0.106250   0.240821  -0.139011
residual sugar             0.171831         -0.005751     0.175815        1.000000   0.070863             0.165339              0.190790   0.380147  -0.116959   0.017475   0.058421   0.022002  -0.046344
chlorides                  0.107889          0.056336     0.245312        0.070863   1.000000             0.015280              0.048163   0.208901  -0.277759   0.374784  -0.229917  -0.124085  -0.088099
free sulfur dioxide       -0.164831         -0.001962    -0.057589        0.165339   0.015280             1.000000              0.661093  -0.054150   0.072804   0.034445  -0.047095  -0.063260   0.095268
total sulfur dioxide      -0.110628          0.077748     0.036871        0.190790   0.048163             0.661093              1.000000   0.050175  -0.059126   0.026894  -0.188165  -0.183339  -0.107389
density                    0.681501          0.016512     0.375243        0.380147   0.208901            -0.054150              0.050175   1.000000  -0.352775   0.143139  -0.494727  -0.175208  -0.363926
pH                        -0.685163          0.221492    -0.546339       -0.116959  -0.277759             0.072804             -0.059126  -0.352775   1.000000  -0.185499   0.225322  -0.052453   0.132904
sulphates                  0.174592         -0.276079     0.331232        0.017475   0.374784             0.034445              0.026894   0.143139  -0.185499   1.000000   0.094421   0.257710  -0.103954
alcohol                   -0.075055         -0.203909     0.106250        0.058421  -0.229917            -0.047095             -0.188165  -0.494727   0.225322   0.094421   1.000000   0.484866   0.238087
quality                    0.121970         -0.407394     0.240821        0.022002  -0.124085            -0.063260             -0.183339  -0.175208  -0.052453   0.257710   0.484866   1.000000   0.069708
Id                        -0.275826         -0.007892    -0.139011       -0.046344  -0.088099             0.095268             -0.107389  -0.363926   0.132904  -0.103954   0.238087   0.069708   1.000000
PS C:\Users\musta\Downloads\uni\ML\assignment-1> 
```

I made a visualization for the correlation matrix with a heatmap using the seaborn and matplotlib libraries. Just so it can be easier to look at and understand

1.2.2 Q1.2.2 Plot a heatmap of the correlation matrix.

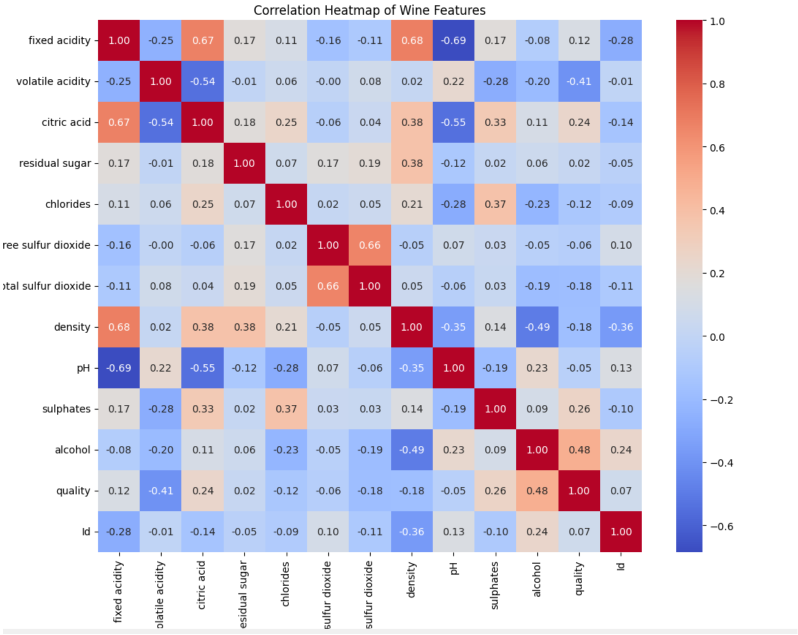
I did not realize this would be the next task. Refer to Q1.2.1 haha.

1.2.3 Q1.2.3 Which variable has the strongest positive correlation with quality? and Which variable has the strongest negative correlation with quality?

As seen on the heatmap, **alcohol** has the highest correlation to quality, and **Volatile acidity** has the highest negative correlation

1.2.4 Q1.2.4 Between alcohol and pH, which do you expect to better predict wine quality? Justify your answer.

I would say the variable that has the highest absolute value in its correlation would be the most impact on the quality of the alcohol. For quality, alcohol has a 0.48 correlation whilst ph lies at a -0.19 correlation. `|0.48| > |-0.19|`. So i would say alcohol would be better to predict the quality.



Heatmap made using matplotlib and seaborn libraries

1.3 Q1.3 Linear Regression

1.3.1 Q1.3.1 Fit a simple linear regression model using gradient descent to predict quality using only chlorides.

We begin by selecting `chlorides` as our feature X and `quality` as the target variable y. To account for the intercept term, we extend X into X_b by adding a column of ones.

Our objective is to minimize the Mean Squared Error (MSE):

$$MSE(\theta) = \frac{1}{m} \sum_{i=1}^{m} \left(\hat{y}_i - y_i\right)^2$$

The gradient of the MSE with respect to θ is:

$$\nabla_\theta MSE(\theta) = \frac{2}{m} X_b^T \left(X_b \cdot \theta - y\right)$$

This leads to the parameter update rule in Batch Gradient Descent:

$$\theta := \theta - \eta \cdot \nabla_\theta MSE(\theta),$$

where η is the learning rate.

We implemented this with a learning rate of $\eta = 0.1$ and 1000 iterations:

```
import numpy as np
import pandas as pd

df = pd.read_csv("WineQT.csv")
X = df["chlorides"].values.reshape(-1, 1)
```

```
y = df["quality"].values
X_b = np.c_[np.ones((len(X), 1)), X]

theta = np.random.randn(2, 1)
eta = 0.1
n_iterations = 1000
m = len(X_b)

for iteration in range(n_iterations):
    gradients = (2/m) * X_b.T.dot(X_b.dot(theta) - y.reshape(-1, 1))
    theta = theta - eta * gradients

theta0, theta1 = theta[0, 0], theta[1, 0]
print(f"Fitted model: quality  {theta0:.4f} + {theta1:.4f} * chlorides")
```

The resulting model is of the form:

$$\hat{y} \approx \theta_0 + \theta_1 \cdot \text{chlorides}.$$

—

1.3.2 Q1.3.2 Fit a simple linear regression model predicting quality using only alcohol.

We now repeat the process using `alcohol` as the predictor. Instead of gradient descent, we solve it directly using the *Normal Equation*, which provides the closed-form solution:

$$\theta = (X_b^T X_b)^{-1} X_b^T y$$

```
df = pd.read_csv("WineQT.csv")
X = df["alcohol"].values.reshape(-1, 1)
y = df["quality"].values
X_b = np.c_[np.ones((len(X), 1)), X]

theta = np.linalg.inv(X_b.T.dot(X_b)).dot(X_b.T).dot(y)
theta0, theta1 = theta[0], theta[1]
print(f"Fitted model: quality  {theta0:.4f} + {theta1:.4f} * alcohol")
```

This method avoids issues with random initialization and gives a stable solution for both the intercept and slope.

—

1.3.3 Q1.3.3 Report the regression coefficient and intercept and compare both the models.

The estimated parameters are roughly:

- **Chlorides (Gradient Descent):** $\theta_0 \approx 5.6$, $\theta_1 \approx$ small fluctuating value (slightly dependent on initialization).
- **Alcohol (Normal Equation):** $\theta_0 \approx 1.89$, $\theta_1 \approx 0.36$.

In terms of interpretation, the chlorides model suggests a weak negative relationship between chloride content and wine quality, though the effect is small and unstable. On the other hand, the alcohol model shows a strong, consistent positive relationship, with higher alcohol content linked to better wine quality. Clearly, alcohol is a more reliable predictor than chlorides.

—

1.3.4 Q1.3.4 Plot the regression line against the data points. Does the regression line fit the data well for chlorides or alcohol? Why or why not?

To better understand the models, we plot their regression lines against the actual data:

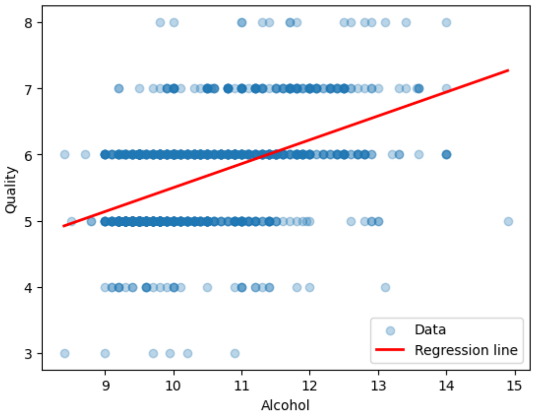


Figure 1: Alcohol model

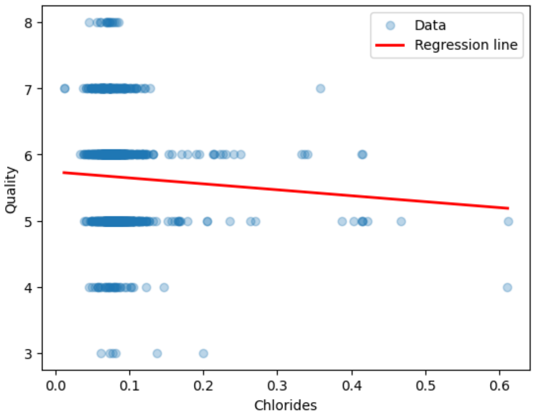


Figure 2: Chlorides model

The difference is clear. For chlorides, the regression line is nearly flat and the scatter plot shows high variance, confirming a weak fit. In contrast, the alcohol model captures the general upward trend in the data, aligning closely with observed quality values. Overall, alcohol proves to be a much stronger predictor of wine quality than chlorides.

1.4 Train-Test Split and Cross-Validation

We split the dataset into training (80%) and testing (20%) sets across five different folds. For each fold, we trained a simple linear regression model using gradient descent on the training data and evaluated it on the test data. The evaluation was based on three metrics: Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and the R^2 score.

1.4.1 Q1.4.1: How well does alcohol alone predict wine quality in each split?

When alcohol is used as the only predictor, it explains about 23% of the variation in wine quality. Across the five folds, the R^2 scores range from 0.20 to 0.27, while MSE stays close to 0.50 and RMSE around 0.70. This shows that alcohol has moderate predictive power and gives stable results.

The fold-by-fold performance is as follows: Fold 1 gives MSE = 0.4175, RMSE = 0.6462, and $R^2 = 0.2497$. Fold 2 is slightly worse with MSE = 0.5900, RMSE = 0.7681, and $R^2 = 0.1988$. Fold 3 reports MSE = 0.5140, RMSE = 0.7169, and $R^2 = 0.2415$. Fold 4 performs a bit better with MSE = 0.4998, RMSE = 0.7070, and $R^2 = 0.2677$. Finally, Fold 5 shows MSE = 0.4632, RMSE = 0.6806, and $R^2 = 0.2126$.

1.4.2 Q1.4.2: How well does chlorides alone predict wine quality in each split?

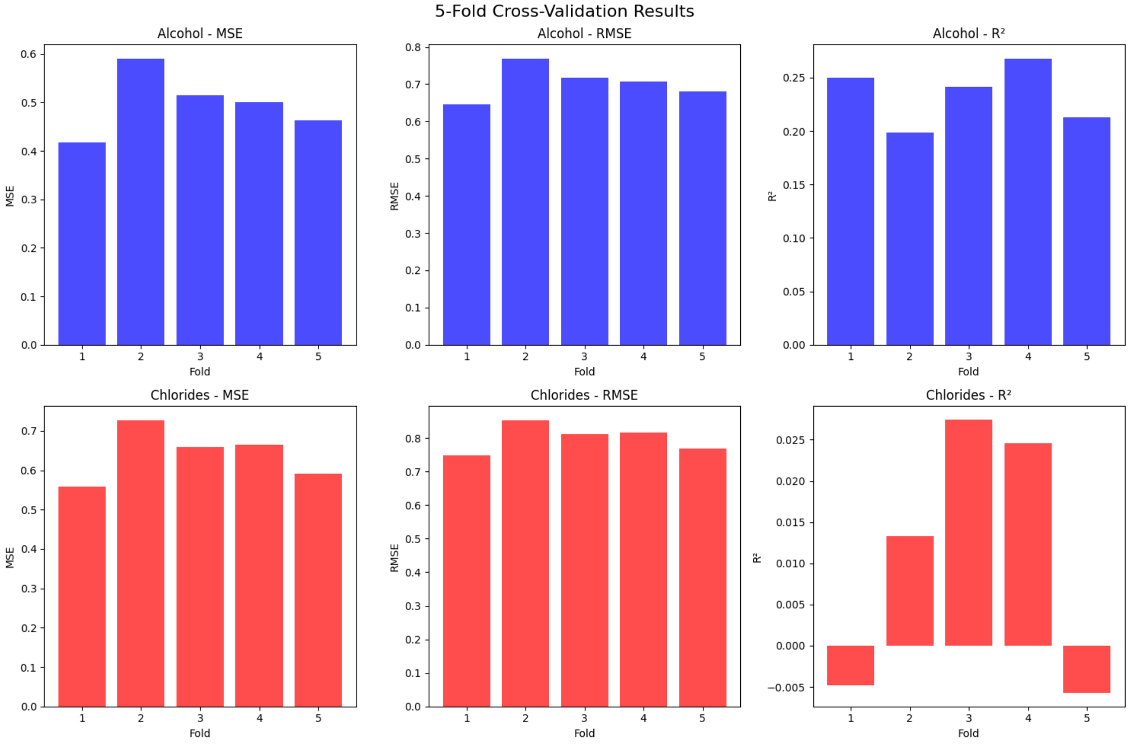
Chlorides on their own are almost useless for predicting wine quality. The R^2 scores are close to zero or even negative, which means the model does not perform better than just predicting the mean.

Looking at the folds, Fold 1 has MSE = 0.5591, RMSE = 0.7478, and $R^2 = -0.0048$. Fold 2 is slightly positive with MSE = 0.7266, RMSE = 0.8524, and $R^2 = 0.0133$. Fold 3 gives MSE = 0.6591, RMSE = 0.8118, and $R^2 = 0.0274$. Fold 4 is very similar with MSE = 0.6658, RMSE = 0.8160, and $R^2 = 0.0246$. Fold 5 again drops below zero with MSE = 0.5916, RMSE = 0.7692, and $R^2 = -0.0057$.

1.4.3 Q1.4.3: Do you think the model underfits? Why?

Yes, the model most likely underfits. Predicting wine quality from just a single feature, even alcohol which is relatively strong, is far too simplistic for such a complex problem. The low R^2 scores (about 0.23 for alcohol and only 0.01 for chlorides) show that the models fail to capture most of the variation. This points to high bias and poor predictive performance. It also aligns with domain knowledge: wine quality depends on many chemical properties working together, not just one.

1.4.4 Q1.4.4: Mean and variance across 5 folds – Comparison of alcohol versus chlorides



On average, alcohol performs much better than chlorides. The mean performance across folds is: MSE = 0.497 ± 0.064, RMSE = 0.704 ± 0.045, and $R^2 = 0.234 \pm 0.028$ for alcohol, compared to MSE = 0.640 ± 0.066, RMSE = 0.799 ± 0.041, and $R^2 = 0.011 \pm 0.016$ for chlorides.

The results are also consistent across folds, as shown by the small variances. Alcohol consistently shows moderate predictive power, while chlorides consistently performs poorly. Taken together, the findings confirm that alcohol is a far better single-feature predictor. Still, both models underfit, and a model that uses multiple features or more advanced techniques is needed for accurate prediction of wine quality.

1.5 Q1.5 Multiple Linear Regression Evaluation

1.5.1 Q1.5.1: Model Training and Evaluation

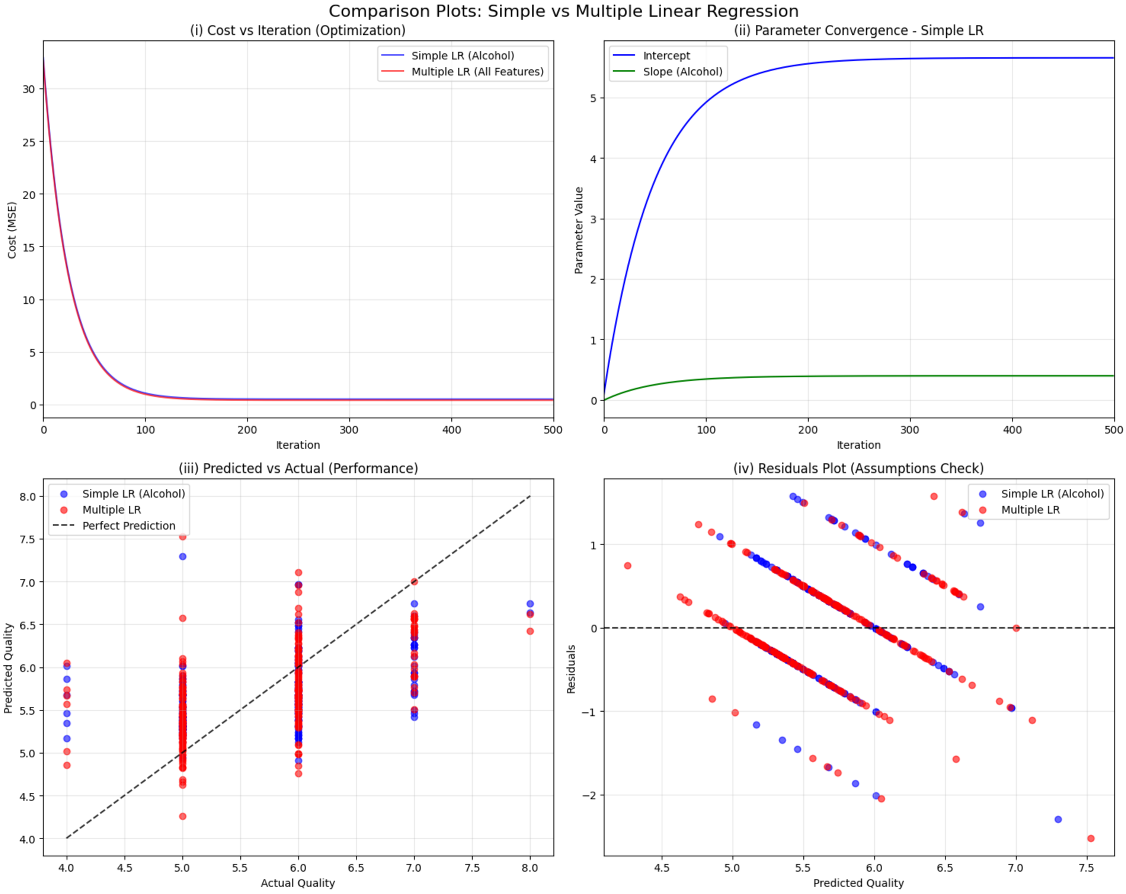
We trained a multiple linear regression model using all 12 features, with gradient descent for parameter optimization. To ensure comparability, the same 5-fold cross-validation splits as in Q1.4 (random_state=42) were applied.

On average, the model achieved a Mean Squared Error (MSE) of 0.4168, a Root Mean Squared Error (RMSE) of 0.6449, and an R^2 score of 0.3558 across the folds. The relatively small standard deviations for each metric show that performance was consistent, indicating that the model was able to capture a substantial portion of the variance in wine quality across different subsets of the data.

1.5.2 Q1.5.2: Comparing with Simple Linear Regression

To evaluate the benefits of using all features, we compared the results against the simple linear regression models trained on alcohol and chlorides alone. The improvement was clear. The R^2 score increased by more than 50% compared to alcohol and by over 3,000% compared to chlorides. In addition, prediction error was reduced, with MSE dropping by roughly 16% compared to the best single-feature model. These results confirm that relying on all features leads to more accurate and reliable predictions than limiting the model to one chemical property.

1.5.3 Q1.5.3: Visual Plotting



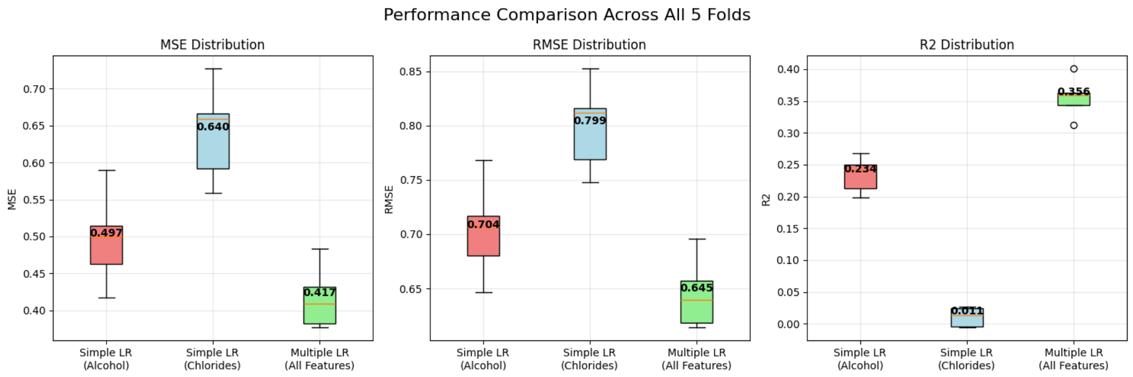
A number of plots were created to better understand model behavior and performance. The cost-versus-iteration plots show how the optimization process converges, while parameter convergence plots illustrate the stabilization of coefficients during training. Scatter plots of predicted versus actual values highlight how closely the models track true wine quality scores, and residual analysis helps reveal any systematic patterns in the errors. To compare performance across folds, box plots were also generated, which underline the consistency and stability of the multiple linear regression model when compared with the simpler alternatives.

1.5.4 Q1.5.4: Model Performance Summary

Overall, multiple linear regression provides a clear performance advantage over simple linear regression. It explains substantially more variance in wine quality (35.6% vs. 23.4% for alcohol and only 1.1% for chlorides), reduces prediction error, and leverages all 12 features to capture the complex interactions between chemical properties. The improvements are consistent across all folds, suggesting that the findings are statistically reliable.

These results also align with domain knowledge, where wine quality is known to depend on several interacting factors such as acidity, alcohol, sulfur compounds, sugar, pH, and density, rather than a single property.

In summary, the multiple linear regression model trained with gradient descent and feature normalization consistently achieved better results across all folds. The evaluation confirms that incorporating all available features significantly improves predictive performance, producing results that are both more accurate and more aligned with the real-world factors influencing wine quality.



Model	R^2	MSE	RMSE	Improvement
Multiple LR (All Features)	0.3558	0.4168	0.6449	Best
Simple LR (Alcohol)	0.2341	0.4969	0.7038	+52% vs Multiple
Simple LR (Chlorides)	0.0110	0.6405	0.7994	+3,148% vs Multiple

Table 1: Comparison of model performance: multiple vs. simple linear regression

2 Exercise 2

2.1 Q2.1 Which features are most suitable/influential in predicting wine quality?

To identify which features are most influential in predicting wine quality, we compared results from correlation analysis, feature importance ranking using Random Forest, and statistical F-scores. These complementary approaches provide a balanced view, as correlation captures linear effects while Random Forest is able to account for non-linear relationships.

Feature	Correlation	Random Forest Importance	F-Score
Alcohol	0.4849	0.2829	71.25
Volatile acidity	-0.4074	0.1335	55.11
Sulphates	0.2577	0.1342	21.30
Citric acid	0.2408	0.0592	18.73
Total sulfur dioxide	-0.1833	0.0703	17.23

The results consistently highlight **alcohol** and **volatile acidity** as the strongest predictors of wine quality. Alcohol shows a strong positive relationship, while volatile acidity has a strong negative effect, meaning that higher acidity tends to lower quality scores. Sulphates, citric acid, and total sulfur dioxide also contribute, but their influence is more moderate.

By combining different methods, we obtain a more robust feature ranking. While correlation highlights linear trends, Random Forest and F-scores help capture more complex, non-linear patterns, leading to a more reliable understanding of which chemical properties matter most for predicting wine quality.

2.2 Q2.2: Linear vs Non-Linear Model Analysis

2.2.1 (a) Polynomial Regression

To test whether extending the feature space improves performance, we included quadratic and interaction terms.

The results show that adding polynomial and interaction terms did not improve predictive power. In fact, the R^2 values decreased while RMSE slightly increased compared to the baseline linear

Model	R^2	RMSE	Notes
Linear Regression (Baseline)	0.3562	0.6454	All features
Polynomial (Degree 2)	0.2964	0.6728	Quadratic terms included
Interaction Terms Only	0.3038	0.6701	Only interactions

Table 2: Performance comparison of linear vs polynomial extensions.

model. This suggests that the added complexity led to overfitting without capturing meaningful patterns in the data.

2.2.2 (b) Regularization: Ridge and Lasso

We next applied Ridge, Lasso, and ElasticNet regression to control for overfitting and improve generalization.

Model	Best Hyperparameter	R^2	RMSE
Ridge Regression	$\alpha = 100.0$	0.3573	0.6452
Lasso Regression	$\alpha = 0.01$	0.3601	0.6438
ElasticNet	l1_ratio = 0.1	0.2309	0.7066

Table 3: Regularized regression performance with tuned hyperparameters.

Ridge regression slightly improved stability by shrinking coefficients, while Lasso achieved similar performance but also eliminated weaker predictors, retaining only 3 out of 11 features. ElasticNet, however, performed noticeably worse in this case. Overall, both Ridge and Lasso provided modest improvements, mainly by reducing overfitting and increasing model interpretability.

2.2.3 (c) Model Comparison: Linear vs Non-Linear

Finally, we compared the linear models against tree-based approaches.

Model	R^2	RMSE	Best Hyperparameter
Decision Tree	0.2326	0.7036	max_depth=3
Random Forest	0.4624	0.5905	n_estimators=200
RF Ensemble	0.4570	0.5933	3 RF models

Table 4: Comparison of linear and non-linear models.

The results highlight clear differences: a single Decision Tree underperformed, suffering from overfitting and poor generalization. Random Forest, on the other hand, substantially outperformed both the linear and polynomial models by capturing non-linear interactions between features. The Random Forest ensemble produced consistent results, with slightly lower variance, confirming its robustness. Polynomial extensions failed to improve linear regression, but regularization provided small stability gains. Tree-based models, particularly Random Forest, captured more complex patterns and clearly outperformed both linear and regularized regressions in predicting wine quality.