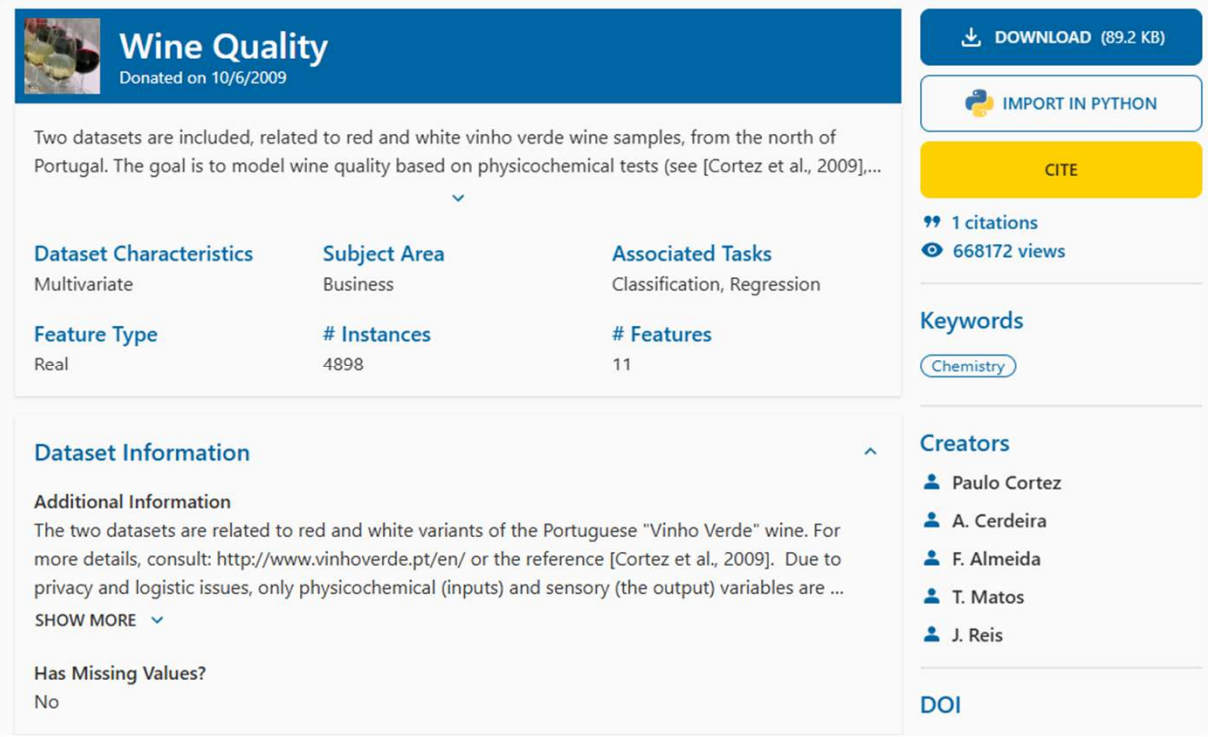
Hands on Supervised & Unsupervised Machine Learning

I Putu Dody Lesmana

Department of Computer Engineering
Institut Teknologi Sepuluh Nopember
2025

Supervised Learning

Dataset: https://archive.ics.uci.edu/dataset/186/wine+quality



Wine Quality

Donated on 10/6/2009

Two datasets are included, related to red and white vinho verde wine samples, from the north of Portugal. The goal is to model wine quality based on physicochemical tests (see [Cortez et al., 2009],...

Dataset Characteristics	Subject Area	Associated Tasks
Multivariate	Business	Classification, Regression
Feature Type	**# Instances**	**# Features**
Real	4898	11

Dataset Information

Additional Information

The two datasets are related to red and white variants of the Portuguese "Vinho Verde" wine. For more details, consult: http://www.vinhoverde.pt/en/ or the reference [Cortez et al., 2009]. Due to privacy and logistic issues, only physicochemical (inputs) and sensory (the output) variables are ...

SHOW MORE

Has Missing Values?

No

DOWNLOAD (89.2 KB)

IMPORT IN PYTHON

CITE

1 citations

668172 views

Keywords

Chemistry

Creators

Paulo Cortez

A. Cerdeira

F. Almeida

T. Matos

J. Reis

DOI

https://archive.ics.uci.edu/ml/machine-learning-databases/wine-quality/winequality-red.csv

```
"fixed acidity";"volatile acidity";"citric acid";"residual sugar";"chlorides";"free sulfur dioxide";"total sulfur dioxide";"density";"pH";"sulphates";"alcohol";"quality"
7.4;0.7;0;1.9;0.076;11;34;0.9978;3.51;0.56;9.4;5
7.8;0.88;0;2.6;0.098;25;67;0.9968;3.2;0.68;9.8;5
7.8;0.76;0.04;2.3;0.092;15;54;0.997;3.26;0.65;9.8;5
11.2;0.28;0.56;1.9;0.075;17;60;0.998;3.16;0.58;9.8;6
7.4;0.7;0;1.9;0.076;11;34;0.9978;3.51;0.56;9.4;5
7.4;0.66;0;1.8;0.075;13;40;0.9978;3.51;0.56;9.4;5
7.9;0.6;0.06;1.6;0.069;15;59;0.9964;3.3;0.46;9.4;5
7.3;0.65;0;1.2;0.065;15;21;0.9946;3.39;0.47;10;7
7.8;0.58;0.02;2;0.073;9;18;0.9968;3.36;0.57;9.5;7
7.5;0.5;0.36;6.1;0.071;17;102;0.9978;3.35;0.8;10.5;5
6.7;0.58;0.08;1.8;0.097;15;65;0.9959;3.28;0.54;9.2;5
7.5;0.5;0.36;6.1;0.071;17;102;0.9978;3.35;0.8;10.5;5
5.6;0.615;0;1.6;0.089;16;59;0.9943;3.58;0.52;9.9;5
7.8;0.61;0.29;1.6;0.114;9;29;0.9974;3.26;1.56;9.1;5
8.9;0.62;0.18;3.8;0.176;52;145;0.9986;3.16;0.88;9.2;5
8.9;0.62;0.19;3.9;0.17;51;148;0.9986;3.17;0.93;9.2;5
8.5;0.28;0.56;1.8;0.092;35;103;0.9969;3.3;0.75;10.5;7
8.1;0.56;0.28;1.7;0.368;16;56;0.9968;3.11;1.28;9.3;5
7.4;0.59;0.08;4.4;0.086;6;29;0.9974;3.38;0.5;9;4
```

GitHub: https://github.com/joiceamirahlesmana/Machine-Learning

MACHINE LEARNING PIPELINE DEMO - WINE QUALITY DATASET

The pipeline includes:

A. Data loading
B. Exploratory Data Analysis (EDA)
C. Feature engineering
D. Train-test splitting
E. Scaling / preprocessing
F. Training 3 different ML models:
   - ❖ SVM (Classification)
   - ❖ Linear Regression (Regression)
   - ❖ Random Forest (Classification)
G. Model evaluation

The goal is to compare regression vs. binary classification approaches using the same dataset.

DATA LOADING

```python
# ==============================================================
# 1. IMPORT LIBRARIES
# ==============================================================
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.svm import SVC
from sklearn.metrics import accuracy_score, classification_report
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, r2_score
from sklearn.ensemble import RandomForestClassifier
```

```python
# ==============================================================
# 2. LOAD DATASET
# ==============================================================
# Dataset source:
# https://archive.ics.uci.edu/ml/machine-learning-databases/wine-qual
url = "https://archive.ics.uci.edu/ml/machine-learning-databases/wine

# Load dataset (semicolon-separated CSV)
data = pd.read_csv(url, sep=';')

print("\n=== Sample Data ===")
print(data.head())

print("\n=== Dataset Info ===")
print(data.info())

print("\n=== Statistical Summary ===")
print(data.describe())
```

```
=== Sample Data ===
   fixed acidity  volatile acidity  citric acid  residual sugar  chlorides  \
0            7.4              0.70         0.00             1.9      0.076
1            7.8              0.88         0.00             2.6      0.098
2            7.8              0.76         0.04             2.3      0.092
3           11.2              0.28         0.56             1.9      0.075
4            7.4              0.70         0.00             1.9      0.076

   free sulfur dioxide  total sulfur dioxide  density    pH  sulphates  \
0                 11.0                  34.0   0.9978  3.51       0.56
1                 25.0                  67.0   0.9968  3.20       0.68
2                 15.0                  54.0   0.9970  3.26       0.65
3                 17.0                  60.0   0.9980  3.16       0.58
4                 11.0                  34.0   0.9978  3.51       0.56

   alcohol  quality
0      9.4        5
1      9.8        5
2      9.8        5
3      9.8        6
4      9.4        5
```

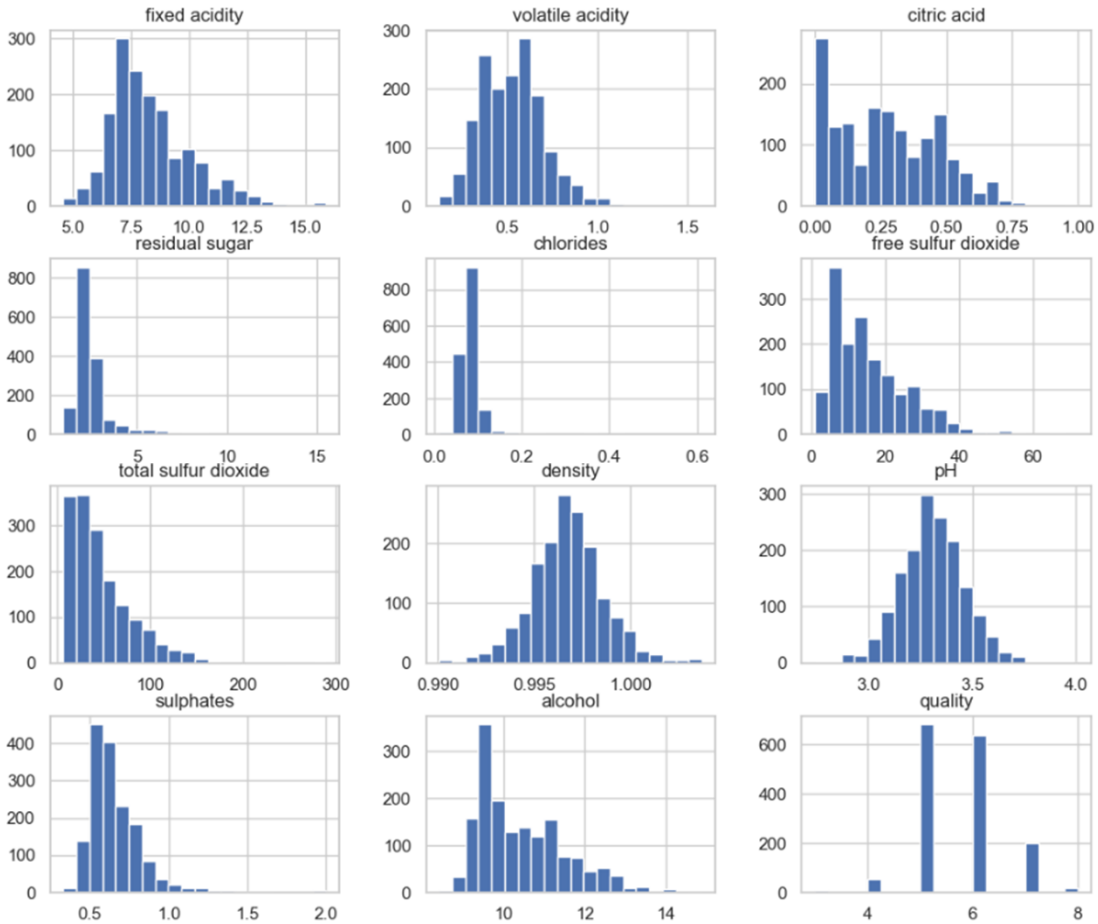
```
=== Dataset Info ===
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1599 entries, 0 to 1598
Data columns (total 12 columns):
 #   Column                Non-Null Count  Dtype
---  ------                --------------  -----
 0   fixed acidity         1599 non-null   float64
 1   volatile acidity      1599 non-null   float64
 2   citric acid           1599 non-null   float64
 3   residual sugar        1599 non-null   float64
 4   chlorides             1599 non-null   float64
 5   free sulfur dioxide   1599 non-null   float64
 6   total sulfur dioxide  1599 non-null   float64
 7   density               1599 non-null   float64
 8   pH                    1599 non-null   float64
 9   sulphates             1599 non-null   float64
 10  alcohol               1599 non-null   float64
 11  quality               1599 non-null   int64
dtypes: float64(11), int64(1)
memory usage: 150.0 KB
None
```

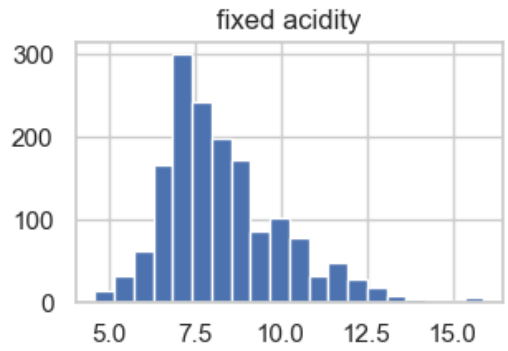
```
=== Statistical Summary ===
       fixed acidity  volatile acidity  citric acid  residual sugar  \
count    1599.000000       1599.000000  1599.000000     1599.000000
mean        8.319637          0.527821     0.270976        2.538806
std         1.741096          0.179060     0.194801        1.409928
min         4.600000          0.120000     0.000000        0.900000
25%         7.100000          0.390000     0.090000        1.900000
50%         7.900000          0.520000     0.260000        2.200000
75%         9.200000          0.640000     0.420000        2.600000
max        15.900000          1.580000     1.000000       15.500000

         chlorides  free sulfur dioxide  total sulfur dioxide      density  \
count  1599.000000          1599.000000           1599.000000  1599.000000
mean      0.087467            15.874922             46.467792     0.996747
std       0.047065            10.460157             32.895324     0.001887
min       0.012000             1.000000              6.000000     0.990070
25%       0.070000             7.000000             22.000000     0.995600
50%       0.079000            14.000000             38.000000     0.996750
75%       0.090000            21.000000             62.000000     0.997835
max       0.611000            72.000000            289.000000     1.003690

                pH    sulphates      alcohol      quality
count  1599.000000  1599.000000  1599.000000  1599.000000
mean      3.311113     0.658149    10.422983     5.636023
std       0.154386     0.169507     1.065668     0.807569
min       2.740000     0.330000     8.400000     3.000000
25%       3.210000     0.550000     9.500000     5.000000
50%       3.310000     0.620000    10.200000     6.000000
75%       3.400000     0.730000    11.100000     6.000000
max       4.010000     2.000000    14.900000     8.000000
```

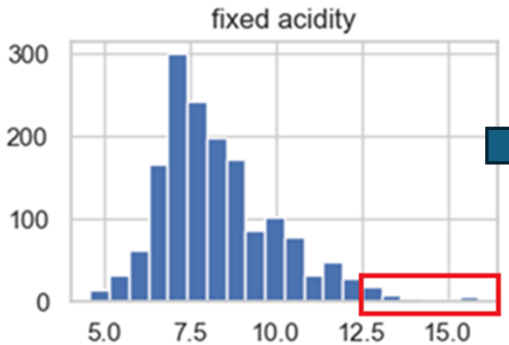
EXPLORATORY DATA ANALYSIS (EDA) & VISUALIZATION (1)

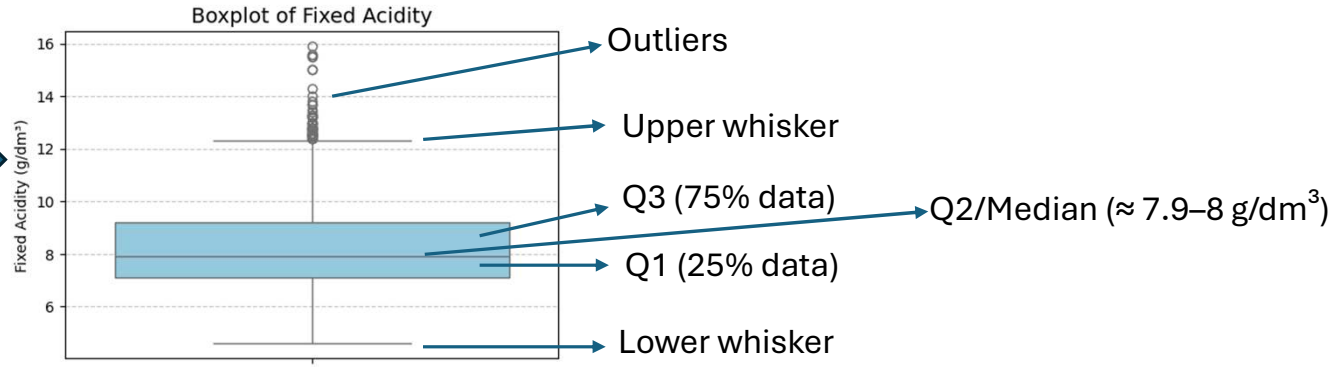




How to analyze this data?
What information can be obtained from "fixed acidity histogram" ?

EXPLORATORY DATA ANALYSIS (EDA) & VISUALIZATION (2)

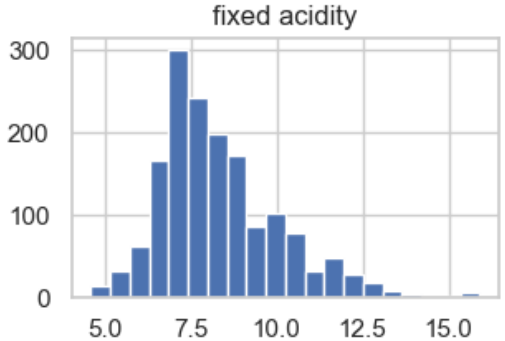


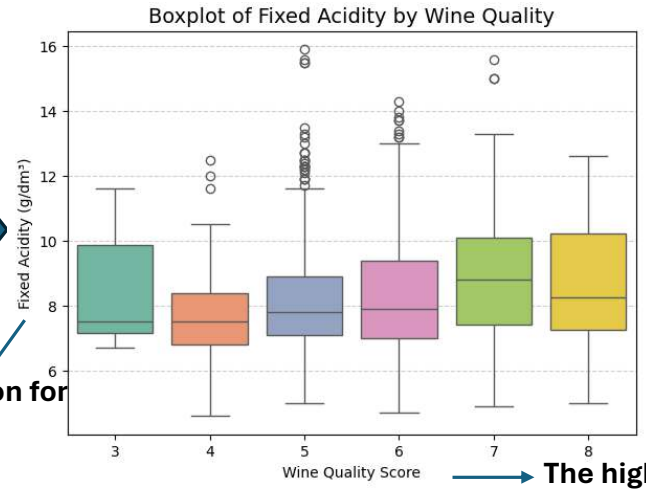


- The distribution resembles a normal (bell-shaped) curve, but is slightly right-skewed
- Values: ~4.0 ≤ fixed acidity values ≤ ~15.5
- The highest frequency occurs around: ≈ 7.0-8.0
- Data Density: 200–300 samples in the central bins
- Outliers: Present on the right side (≥ 12) → very low frequency

  outliers should be examined further before model training
- Outliers: wines with **very high acidity** (unusual chemical composition), or possible measurement errors or anomalies
- Contextual Interpretation (Wine Domain):
  - ✓ Fixed acidity refers to non-volatile acids (such as tartaric acid) that determine the freshness and stability of wine.
  - ✓ Most red wines in this dataset have fixed acidity levels between 7–9 g/dm^3, indicating balanced and typical acidity levels.
  - ✓ The higher outliers (above 12) might represent wines that are sharper or excessively acidic, potentially affecting their overall quality.

EXPLORATORY DATA ANALYSIS (EDA) & VISUALIZATION (3)





Non-volatile acid concentration for each quality group

The higher the score, the better the wine quality

- Most wines have similar fixed acidity levels around **7–8 g/dm^3**, but higher-quality wines (scores 7–8) tend to show slightly higher or more stable acidity than lower-quality wines (scores 3–5)
- Wines with medium quality (scores 5–6) show the widest variation in fixed acidity, while wines with very low or very high quality (scores 3 or 8) display more consistent acidity levels.
- High-quality wines (scores 7–8) tend to have well-maintained acidity — not too low (which would make them taste flat) and not too high (which would make them overly sharp).
- Some wine samples show outliers with fixed acidity above 12 g/dm^3, indicating unusual fermentation conditions or extreme chemical compositions.
- There is no strong linear correlation between fixed acidity and wine quality, but higher-quality wines (scores 7–8) tend to have more stable and balanced acidity levels.

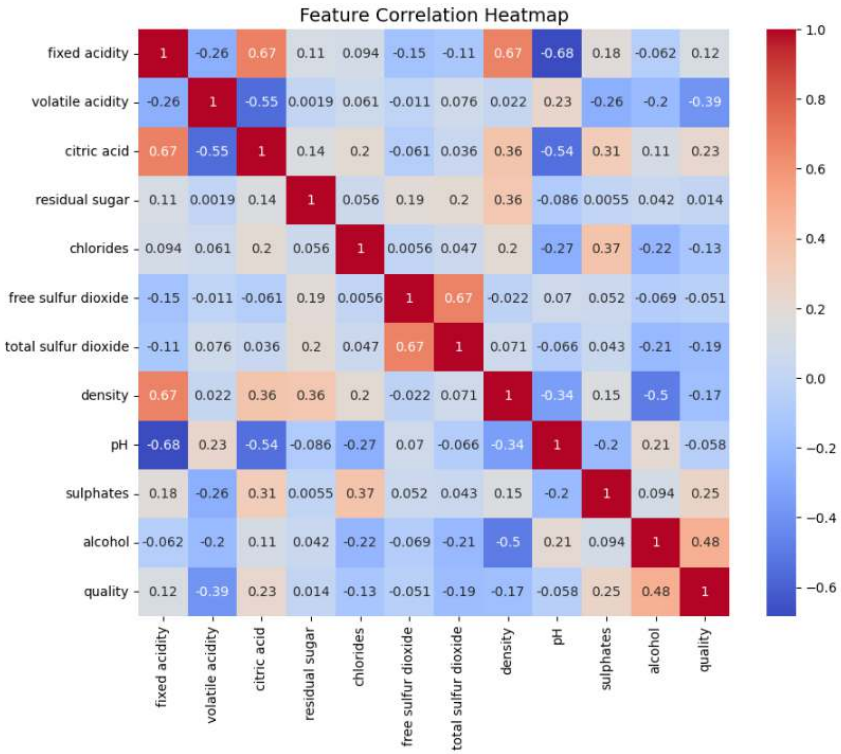
PEARSON CORRELATION

- A statistical measure that shows how strong and in what direction the linear relationship is between two numerical variables.
- The Pearson correlation coefficient is represented by the letter r, and its value ranges from **–1 to +1**:
  - **+1:** Perfect positive relationship (both variables increase together)
  - **0:** No linear relationship (changes in one variable do not affect the other)
  - **-1:** Perfect negative relationship (when one variable increases, the other decreases)
- **Function:** 1) identify which features have the strongest influence on the target variable (e.g., quality), 2) detect multicollinearity, a condition where two features are highly correlated, which can confuse or destabilize a model.
- **Formula:**

$$r = \frac{\mathrm{Cov}(X, Y)}{\sigma_X \cdot \sigma_Y}$$

  The correlation value is the covariance between X and Y divided by the product of their standard deviations.
- **Examples:**
  - alcohol and quality have r = +0.48 → the higher the alcohol content, the better the wine quality.
  - volatile acidity and quality have r = –0.39 → the higher the volatile acidity, the lower the wine quality.

EXPLORATORY DATA ANALYSIS (EDA) & VISUALIZATION (4)



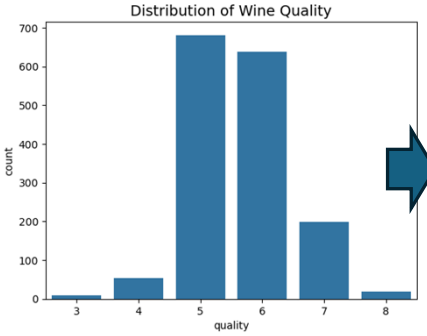
- This heatmap shows the linear relationship (correlation) between every pair of numerical variables.
- **Red:** positive correlation, **Blue:** negative correlation, values near 0 → no strong linear relationship
- **Strong Positive Correlations:** fixed acidity ↔ citric acid (+0.67), free sulfur dioxide ↔ total sulfur dioxide (+0.67), alcohol ↔ quality (+0.48)
- **Strong Negative Correlations:** fixed acidity ↔ pH (–0.68), volatile acidity ↔ quality (–0.39)
- **Weak Correlations:** residual sugar, chlorides, and density show weak correlations with quality (around 0.1–0.2) → This means these features have little direct influence on wine quality ratings.
- **Most influential factors for wine quality:** Alcohol (strong positive), Volatile acidity (strong negative)
- **Supporting factors:** Citric acid, sulphates, and fixed acidity.
- **Less relevant factors:** Residual sugar, chlorides, density.

FEATURE ENGINEERING, TRAIN-TEST SPLITTING, & SCALING / PREPROCESSING

```python
# ================================================================
# 4. FEATURE ENGINEERING (BINARY CLASS TARGET)
# ================================================================
# Convert the numeric quality label into a binary class:
# 1 = good quality (quality > 5), 0 = bad quality (quality <= 5)
data['quality_class'] = [1 if q > 5 else 0 for q in data['quality']]

print("\n=== Class Distribution (0 = bad, 1 = good) ===")
print(data['quality_class'].value_counts())
```

```
=== Class Distribution (0 = bad, 1 = good) ===
quality_class
1    855
0    744
Name: count, dtype: int64
```




```python
# ================================================================
# 5. SPLIT FEATURES & TARGETS
# ================================================================
# X (features), y_classification (binary), y_regression (continuous)
X = data.drop(['quality', 'quality_class'], axis=1)
y_classification = data['quality_class']
y_regression = data['quality']

# Train-test split (80% train, 20% test)
X_train, X_test, y_train_cls, y_test_cls = train_test_split(
    X, y_classification, test_size=0.2, random_state=42
)
_, _, y_train_reg, y_test_reg = train_test_split(
    X, y_regression, test_size=0.2, random_state=42
)
```

```python
# ================================================================
# 6. FEATURE SCALING (ESSENTIAL FOR SVM & LINEAR REGRESSION)
# ================================================================
# StandardScaler standardizes features by removing the mean and scaling to unit variance
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)  # fit on training data
X_test_scaled = scaler.transform(X_test)        # apply same transformation to test data
```

TRAINING & MODEL EVALUATION

		Predicted Class: Positive	Predicted Class: Negative	
Actual Class	Positive	True Positive (TP)	False Negative (FN) Type II Error	**Sensitivity** $\frac{TP}{(TP+FN)}$
	Negative	False Positive (FP) Type I Error	True Negative (TN)	**Specificity** $\frac{TN}{(TN+FP)}$
		Precision $\frac{TP}{(TP+FP)}$	**Negative Predictive Value** $\frac{TN}{(TN+FN)}$	**Accuracy** $\frac{TP+TN}{(TP+TN+FP+FN)}$

TP (True Positive): Correctly predicted positives

TN (True Negative): Correctly predicted negatives

FP (False Positive): Incorrectly predicted as positive

FN (False Negative): Incorrectly predicted as negative

```
--- Support Vector Machine (Classification) ---
Accuracy: 0.7719
Classification Report:
              precision    recall  f1-score   support

           0       0.73      0.77      0.75       141
           1       0.81      0.77      0.79       179

    accuracy                           0.77       320
   macro avg       0.77      0.77      0.77       320
weighted avg       0.77      0.77      0.77       320
```

```
--- Random Forest (Classification) ---
Accuracy: 0.7906
Classification Report:
              precision    recall  f1-score   support

           0       0.76      0.76      0.76       141
           1       0.81      0.82      0.81       179

    accuracy                           0.79       320
   macro avg       0.79      0.79      0.79       320
weighted avg       0.79      0.79      0.79       320
```

1. Mean Squared Error (MSE)

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

- **Meaning:**
  MSE measures how far the model's predictions ($\hat{y}_i$) are from the actual values (y_i).
  It calculates the **average of the squared differences** between predicted and true values.
- **Interpretation:**
  - The smaller the MSE, the better the model fits the data.
  - In your output, **MSE = 0.3900**, meaning that, on average, the squared difference between predicted and actual values is **0.39**.
  - However, MSE is **scale-dependent**, so it should be compared relative to the data range.

---

2. R-squared (R^2 Score)

$$R^2 = 1 - \frac{\sum(y_i - \hat{y}_i)^2}{\sum(y_i - \bar{y})^2}$$

- **Meaning:**
  R^2 measures **how much of the variation in the target variable** can be explained by the model's input features.
- **Range:**
  - $R^2 = 1.0$ → Perfect prediction
  - $R^2 = 0.0$ → Model performs no better than predicting the mean
  - $R^2 < 0$ → Model performs worse than a simple average
- **Interpretation for your result:**
  - $R^2 = 0.4032$ means **the model explains about 40.32% of the variance** in the wine quality scores.
  - The remaining 59.68% of variability is due to other factors not captured by the model.
  - This indicates a **moderate linear relationship** — not bad, but not highly accurate either.

```
--- Linear Regression (Regression) ---
Mean Squared Error (MSE): 0.3900
R-squared (R2 Score): 0.4032
```

UNSUPERVISED LEARNING – K-MEANS

The K-Means Algorithm

The algorithm follows an iterative process to find the optimal cluster assignments. It primarily consists of two main steps that are repeated until convergence.

🎯 **Goal:** To minimize the **Within-Cluster Sum of Squares (WCSS)**, which is the sum of the squared distances between each data point and its assigned cluster's centroid.

Step 1: Initialization

First, you must choose the number of clusters, **K**. Then, **K** initial **centroids** are chosen from the data points. A common method is to select them randomly.

- Let $C = \{c_1, c_2, \ldots, c_k\}$ be the set of initial centroids.

Step 2: Assignment Step

For each data point in the dataset, calculate its distance to every one of the K centroids. The data point is then assigned to the cluster of the closest centroid. The most commonly used distance metric is the squared Euclidean distance.

- **Formula:** Each data point x_i is assigned to a cluster S_j based on the following rule, which finds the centroid c_j that minimizes the squared distance:

$$\text{Assign } x_i \text{ to cluster } S_j, \text{ where } j = \arg\min_{j \in \{1,\ldots,k\}} \|x_i - c_j\|^2$$

Here, $\|x_i - c_j\|^2$ represents the squared Euclidean distance between data point x_i and centroid c_j.

Step 3: Update Step

After all data points have been assigned to clusters, the position of each centroid is recalculated. The new position for a centroid is the **mean** (or average) of all the data points that were assigned to its cluster.

- **Formula:** The new centroid c_j for cluster S_j is calculated as:

$$c_j = \frac{1}{|S_j|} \sum_{x_i \in S_j} x_i$$

Where $|S_j|$ is the total number of data points in cluster S_j.

Step 4: Repetition 🔄

Steps 2 (Assignment) and 3 (Update) are repeated until a stopping condition is met.

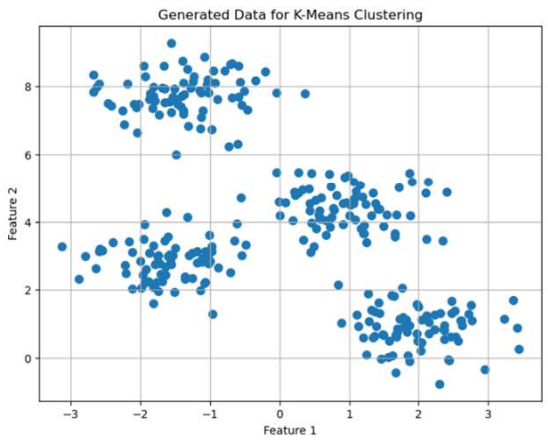
Convergence (Stopping Condition)

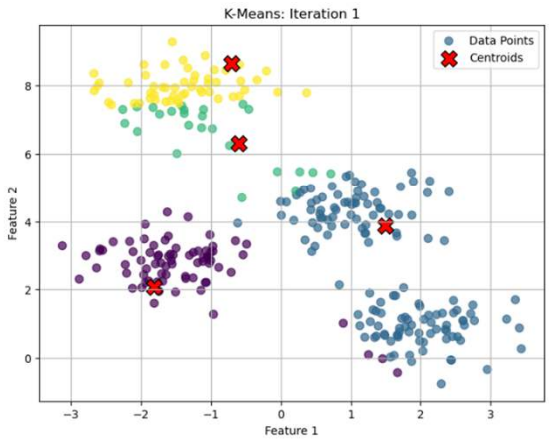
The iterative process stops when the algorithm converges, which typically happens when one of the following conditions is met:

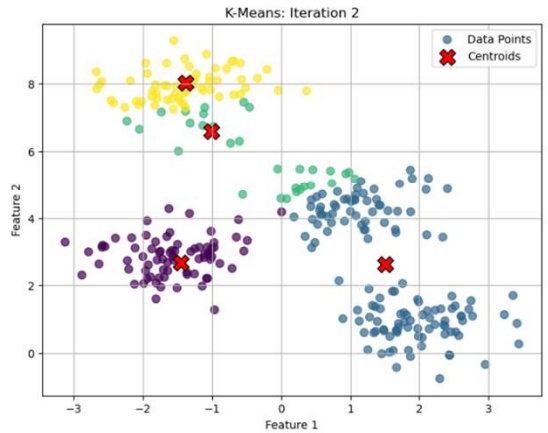
1. **The centroids no longer move:** The positions of the centroids remain the same after an update step.
2. **Cluster assignments do not change:** No data points switch clusters between iterations.
3. **Maximum iterations reached:** The algorithm stops after a pre-defined number of iterations to prevent it from running indefinitely.

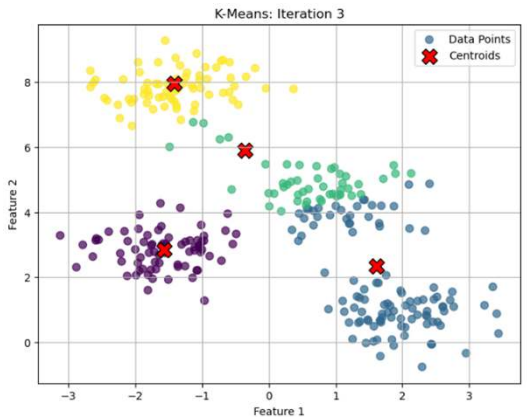
K-MEANS TRAINING

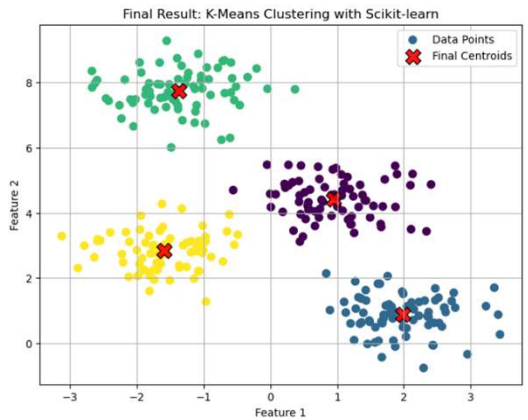
GitHub: https://github.com/joiceamirahlesmana/Machine-Learning











ASSIGNMENTS

1. Find a dataset with tabular data (from Kaggle, UCI Machine Learning Repository, etc.) and perform classification using Random Forest, Support Vector Machine (SVM), and Logistic Regression (LR).Next, conduct a detailed analysis following the step-by-step procedures that have been previously explained.
2. Please watch the following video: https://youtu.be/0jOlZpFFxCE?si=UomNaSsBXQeErGxS

   Provide a detailed explanation of FAISS (Facebook AI Similarity Search), focusing on IndexFlatL2 and IndexIVF (GitHub source code available) as discussed in the video. Explore in depth how these two indexing methods work internally, including the clustering mechanism used in FAISS for efficient similarity search. Include an experimental demonstration that you performed to support your analysis—this may involve extending or modifying the GitHub source code to show practical understanding and results.

Please email the complete answer to: dody.lesmana@gmail.com with the subject: “Hands on Supervised_Unsupervised Learning – Student Registration Number”.

The assignment must be submitted before 23.59pm, 14 November 2025.

Thank You