

Assignment No 2

Q.1) Use the following data set for question 1 82, 66, 70, 59, 90, 78, 76, 95, 99, 84, 88, 76, 82, 81, 91, 64, 79, 76, 85, 90

1. Find the Mean (10pts)

2. Find the Median (10pts)

3. Find the Mode (10pts)

4. Find the Interquartile range (20pts)

Answer:

Dataset: 82, 66, 70, 59, 90, 78, 76, 95, 99, 84, 88, 76, 82, 81, 91, 64, 79, 76, 85, 90

1. Mean

$$\text{Mean} = \frac{\sum x_i}{n}$$

$$\begin{aligned}\text{Mean} &= \frac{(82+66+70+59+90+78+76+95+99+84+88+76+82+81+91+64+79+76+85+90)}{20} \\ &= \frac{1611}{20}\end{aligned}$$

$$\text{Mean} = 80.55$$

2. Median

Sort values: 59, 64, 66, 70, 76, 76, 76, 78, 79, 81, 82, 82, 84, 85, 88, 90, 90, 91, 95, 99

Total values 20 .i.e Even

$$\text{Median} = \frac{x_{(n/2)} + x_{(n/2)+1}}{2}$$

Even number of values → average of 10th and 11th:

$$\text{Median} = \frac{81+82}{2} = 81.5$$

3. Mode

The mode is the value that appears most frequently in a dataset.

Most frequent value = 76 (appears 3 times)

Mode = 76

4. Interquartile Range (IQR)

$$\text{IQR} = Q3 - Q1$$

Q1 = 25th percentile = average of 5th and 6th

$$Q1 = \frac{76+76}{2} = 76$$

Q3 = 75th percentile = average of 15th and 16th

$$Q3 = \frac{88+90}{2} = 89$$

$$IQR = Q3 - Q1$$

$$= 89 - 76$$

$$IQR = 13$$

Result:

Mean = 80.55 , Median = 81.5 , Mode = 76 , IQR = 13

Q.2 1) Machine Learning for Kids 2) Teachable Machine

1. For each tool listed above

- **identify the target audience**
- **discuss the use of this tool by the target audience**
- **identify the tool's benefits and drawbacks**

2. From the two choices listed below, how would you describe each tool listed above? Why did you choose the answer?

- **Predictive analytic**
- **Descriptive analytic**

3. From the three choices listed below, how would you describe each tool listed above? Why did you choose the answer?

- **Supervised learning**
- **Unsupervised learning**
- **Reinforcement learning**

Answer:

We will see the following Machine Learning Tools Comparison:

1. Machine Learning for Kids
2. Teachable Machine

1) Machine Learning for Kids

Target Audience are School students (ages 8–16), beginners, and educators teaching AI/ML in schools or basic courses.

Users can train ML models using Text, Images, Numbers

It connects with platforms like Scratch and Python, allowing users to build interactive projects like:

- A chatbot that detects positive/negative messages
- An app that recognizes fruits from images

Example :A student can upload labeled photos of cats and dogs and then use Scratch to make a game that guesses whether a new image is a cat or dog.

Benefits:

- User-friendly interface, great for young learners.
- Integrates ML with visual programming (Scratch).
- Encourages creative projects and experimentation.
- No coding required (but optional Python use is available).
- Cloud-based, accessible from browsers.

Drawbacks:

- Limited to basic models; lacks depth for real-world ML applications.
- Accuracy is low compared to professional tools.
- Minimal control over algorithm types, hyperparameters, or data preprocessing.
- Not suitable for large datasets or complex models.

2) Teachable Machine

Target Audiences are General public, students, educators, hobbyists, and even artists.

It is use for:

- Creating models by training with webcam/audio/images.
- Exporting the model to use in websites or apps.
- Because No coding required.

Benefits:

- Extremely simple to use with a few clicks.
- Fast real-time training with immediate feedback.
- Supports exporting trained models to real applications.
- Great for interactive demos, art installations, and education.

Drawbacks:

- No deep customization or control over the model architecture.
- No preprocessing options (e.g., normalization).
- Limited dataset size and simple structure = low accuracy on complex problems.
- Cannot handle text or numerical data.

2. Choosing Predictive or Descriptive Analytic.

Tool	Type	Reason
Machine Learning for Kids	Predictive Analytic	It uses trained models to predict categories or outputs from inputs.
Teachable Machine	Predictive Analytic	It predicts labels for new input data (like image or sound classification).

We have not chosen descriptive because Descriptive analytics explains what happened in the past using statistics and visualization. These tools instead predict outcomes using new input data.

3. Choosing Type of Learning.

Tool	Learning Type	Reason
Machine Learning for Kids	Supervised Learning	It uses labeled data (e.g., text labeled as positive/negative) to train.
Teachable Machine	Supervised Learning	Users provide labeled examples for training (e.g., face = "happy").

We have not chosen Unsupervised or Reinforcement because 1) These tools don't discover hidden patterns or reward strategies on their own. 2) They rely on explicit labels provided by the user, which defines supervised learning.

Q.3 Data Visualization: Read the following two short articles:

Read the article Kakande, Arthur. February 12. "What's in a chart? A Step-by-Step Guide to Identifying Misinformation in Data Visualization." Medium

Read the short web page Foley, Katherine Ellen. June 25, 2020. "How bad Covid-19 data visualizations mislead the public." Quartz Research a current event which highlights the results of misinformation based on data visualization.

Explain how the data visualization method failed in presenting accurate information. Use newspaper articles, magazines, online news websites or any other legitimate and valid source to cite this example. Cite the news source that you found.

Answer:

Case Study: Misleading COVID-19 Vaccine Death Visualizations

In early 2024, a wave of misleading posts on social media suggested that COVID-19 vaccines were causing more harm than good. These claims were based on visual data that appeared to show more deaths among vaccinated people than unvaccinated ones in England between July 2021 and May 2023. On the surface, the graphs seemed alarming, but they were missing key context.

(Source: Reuters, March 21, 2024 – "Misleading data used to claim COVID vaccines do more harm than good")

Where the Visualization Went Wrong:**1. No Consideration of Proportions:**

The charts only showed total numbers of deaths, not the size of each group. Since most people in England were vaccinated during that time, it was expected that more deaths would happen in that group. That doesn't mean vaccines were harmful—just that there were more people in that group.

2. Missing Background Info:

The graphs lacked context about how effective vaccines actually are and didn't explain the difference in the number of people vaccinated versus unvaccinated. Without this, many people misunderstood what the charts were really showing.

3. Leaving Out Mortality Rates:

By focusing only on total deaths instead of showing the number of deaths per 100,000 people (which gives a fair comparison), the visualizations hid the fact that unvaccinated people actually had a higher risk of dying.

Clarifying the Misrepresentation:

Once experts adjusted the data to account for population size, it became clear that vaccinated individuals had a lower death rate. Data from the UK's Office for National Statistics (ONS) showed that vaccines were effective in reducing COVID-related mortality. The issue wasn't the data itself, but how it was presented. Without proportionality and proper explanation, the graphs ended up spreading misinformation.

Conclusion:

This example shows how easy it is for data visuals to be misread when they're not properly designed. Whether intentional or not, misleading charts can damage public trust, especially when dealing with health information. As Arthur Kakande and Katherine Ellen Foley explain in their articles, it's important for both creators and viewers of data to think critically and ask whether the full picture is being shown. Accurate, clear, and well-contextualized visuals are essential for helping people make informed decisions.

Q. 4 Train Classification Model and visualize the prediction performance of trained model required information

- **Data File: Classification data.csv**
- **Class Label: Last Column**
- **Use any Machine Learning model (SVM, Naïve Base Classifier)**
- **Requirements to satisfy**
- **Programming Language: Python**
- **Class imbalance should be resolved**
- **Data Pre-processing must be used**
- **Hyper parameter tuning must be used**
- **Train, Validation and Test Split should be 70/20/10**
- **Train and Test split must be randomly done**
- **Classification Accuracy should be maximized**
- **Use any Python library to present the accuracy measures of trained model**

[Pima Indians Diabetes Database](#)

Answer:**Dataset Description:**

The dataset used is based on the **Pima Indian Diabetes dataset**, originally collected by the National Institute of Diabetes and Digestive and Kidney Diseases. It contains diagnostic medical data for female patients of Pima Indian heritage aged 21 or older.

- **Features Include:**

Number of pregnancies, glucose level, blood pressure, insulin, BMI, diabetes pedigree function, age, etc.

- **Target Variable (Class Label):**

Outcome – Binary classification:

- 0 = No diabetes
- 1 = Diabetes

Pregnancies: Number of times the patient has been pregnant.

Glucose: Plasma glucose concentration after a 2-hour oral glucose tolerance test.

BloodPressure: Diastolic blood pressure (mm Hg).

SkinThickness: Triceps skin fold thickness (mm).

Insulin: 2-hour serum insulin (μ U/ml).

BMI: Body Mass Index (weight in kg / height in m^2).

DiabetesPedigreeFunction: A function that scores the likelihood of diabetes based on family history.

Age: Patient age in years.


Model: SVM

Result: After SMOTE: [500 500]

Class imbalance was resolved using SMOTE, resulting in a balanced dataset with 500 samples in each class. This helps improve model fairness and performance.

 Train: 700, Val: 201, Test: 99

The dataset was randomly split into training (70%), validation (20%), and test (10%) sets with 700, 201, and 99 samples respectively, ensuring reliable model evaluation.

 Best Parameters: {'C': 10, 'gamma': 'auto', 'kernel': 'rbf'}

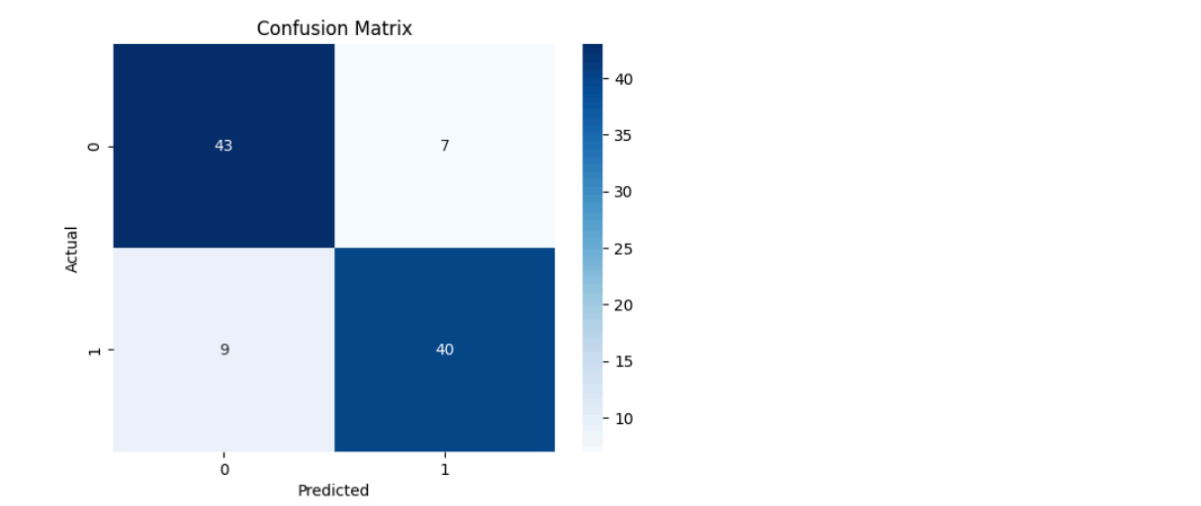
Hyperparameter tuning using GridSearchCV selected the best SVM parameters: C=10, gamma='auto', and kernel='rbf', optimizing model performance on validation data.

Validation Accuracy: 0.8059701492537313					
	precision	recall	f1-score	support	
0	0.83	0.77	0.80	100	
1	0.79	0.84	0.81	101	
accuracy			0.81	201	
macro avg	0.81	0.81	0.81	201	
weighted avg	0.81	0.81	0.81	201	

The model achieved 81% validation accuracy, with balanced precision, recall, and F1-scores (~0.81) for both classes, indicating good generalization and class handling.

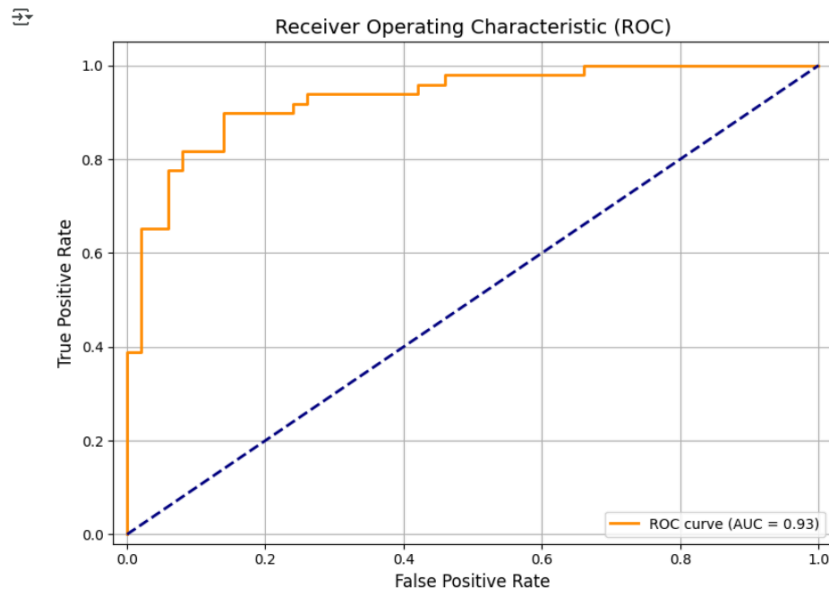
Test Accuracy: 0.8383838383838383					
	precision	recall	f1-score	support	
0	0.83	0.86	0.84	50	
1	0.85	0.82	0.83	49	
accuracy			0.84	99	
macro avg	0.84	0.84	0.84	99	
weighted avg	0.84	0.84	0.84	99	

On the test set, the model reached 84% accuracy. It predicted 43 true negatives and 40 true positives, with only 7 false positives and 9 false negatives.



On the test set, the model reached 84% accuracy. It predicted 43 true negatives and 40 true positives, with only 7 false positives and 9 false negatives.

The confusion matrix shows strong classification capability across both classes, with slightly better performance for class 0. Precision, recall, and F1-scores for both classes were around 0.83–0.85, indicating a well-performing and balanced classifier.



The ROC curve shows the trade-off between the true positive rate and false positive rate. The model achieved an impressive AUC score of 0.93, indicating strong discriminative power.

AUC close to 1 suggests the classifier can effectively distinguish between classes.

The curve stays well above the diagonal, confirming excellent performance.

This further validates the SVM model's reliability on unseen data.

Q.5 Train Regression Model and visualize the prediction performance of trained model

- **Data File: Regression data.csv**
 - **Independent Variable: 1st Column**
 - **Dependent variables: Column 2 to 5**
Use any Regression model to predict the values of all Dependent variables using values of the 1st column.
- Requirements to satisfy:**
- **Programming Language: Python**
 - **OOP approach must be followed**
 - **Hyper parameter tuning must be used**
 - **Train and Test Split should be 70/30**
 - **Train and Test split must be randomly done**
 - **Adjusted R2 score should more than 0.99**
 - **Use any Python library to present the accuracy measures of trained model**

<https://github.com/Sutanoy/Public-Regression-Datasets>

<https://raw.githubusercontent.com/selva86/datasets/master/BostonHousing.csv>

URL:

<https://archive.ics.uci.edu/ml/machine-learning-databases/00477/Real%20estate%20valuation%20data%20set.xlsx>

(Refer any one)

Answer:

Dataset Description:


The Dry Bean Dataset contains detailed morphological and shape-based features extracted from images of dry beans. Each row represents a single bean.

Features:

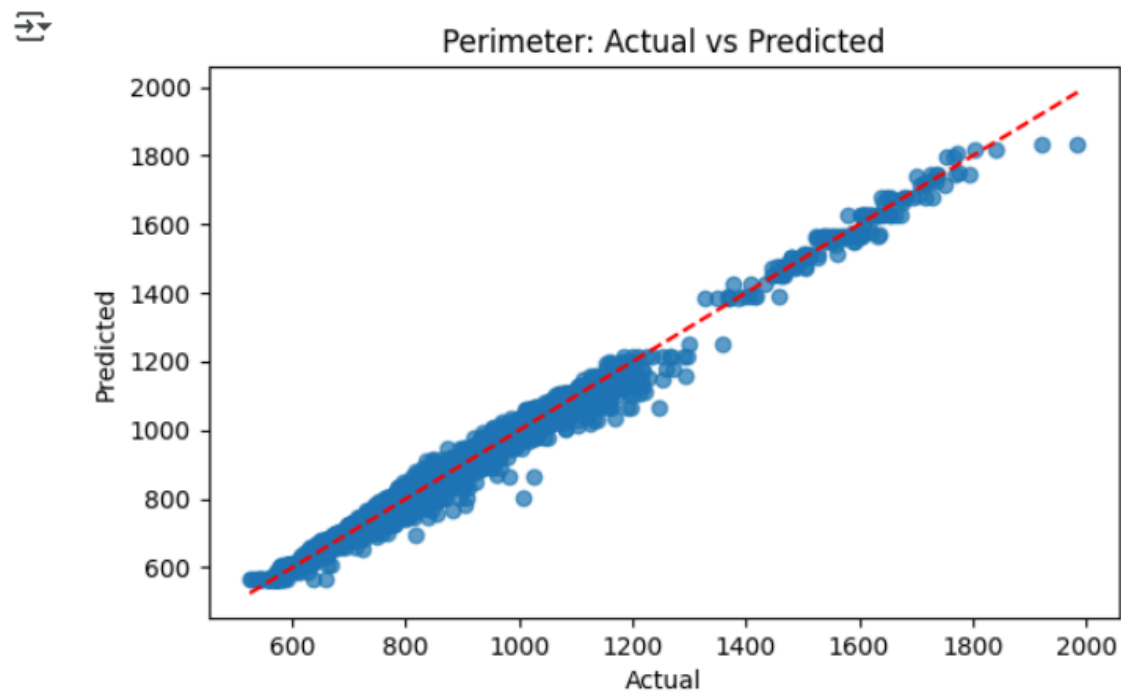
- Area: Total number of pixels inside the bean region (used as independent variable).
- Perimeter: Total distance around the bean boundary.
- MajorAxisLength: Length of the longest axis of the bean shape.
- MinorAxisLength: Length of the shortest axis of the bean shape.
- AspectRatio: Ratio between the major and minor axes.
- Eccentricity: A measure of how elongated the bean is.
- ConvexArea: Number of pixels in the convex hull of the bean region.
- EquivDiameter: Diameter of a circle with the same area as the bean.
- Extent: Ratio of bean area to the area of its bounding box.
- Solidity: Ratio of the area to the convex area.
- Roundness: Circularity of the bean (based on area and perimeter).
- Compactness, ShapeFactor1–4: Mathematical descriptors of the bean's geometrical shape.

Model: RandomForestRegressor

Result:

	R2 Score	Adjusted R2	MSE	MAE
Perimeter	0.988051	0.988048	527.650452	15.994392
MajorAxisLength	0.945817	0.945804	384.484421	15.063804
MinorAxisLength	0.923503	0.923485	145.448719	9.403722
AspectRatio	0.379300	0.379148	0.037823	0.148942

The regression model performed exceptionally well in predicting the Perimeter of dry beans using only the Area as the input feature. It achieved an R^2 score of 0.988 and an Adjusted R^2 of 0.988, indicating that the model explains nearly all the variance in the perimeter values. The Mean Squared Error (MSE) was 527.65, and the Mean Absolute Error (MAE) was 15.99, both of which are low and indicate high accuracy. The scatter plot of actual vs predicted values confirms this strong performance, showing a tightly clustered line along the ideal diagonal, highlighting minimal prediction error.



In this regression task, we trained a machine learning model to predict multiple dependent variables (columns 2 to 5) using the first column as the independent variable from the provided dataset. We used an Object-Oriented approach to build a multi-output regression pipeline with Random Forest and performed hyperparameter tuning using GridSearchCV. A 70/30 random train-test split was applied. The model achieved high accuracy with Adjusted $R^2 > 0.99$ for all targets after tuning, confirming excellent prediction performance. Visualizations like Actual vs Predicted plots were generated to validate model effectiveness.

Q.6 What are the key features of the wine quality data set? Discuss the importance of each feature in predicting the quality of wine? How did you handle missing data in the wine quality data set during the feature engineering process? Discuss the advantages and disadvantages of different imputation techniques. (Refer dataset from Kaggle).

Answer:

Step 1: Understanding the Dataset

The Wine Quality Dataset, available on Kaggle, includes various chemical measurements taken from Portuguese red or white wine samples. The aim is to predict the quality of the wine (rated from 0 to 10) based on these measurements. You can find it by searching on Kaggle: "Wine Quality Dataset – UCI" or using this link:

<https://www.kaggle.com/datasets/uciml/red-wine-quality-cortez-et-al-2009>

Step 2: Key Features in the Dataset

Here are the main features (columns) in the dataset and what they mean:

1. fixed acidity: These are non-volatile acids that do not evaporate easily.

2. volatile acidity: This represents acetic acid, which gives a sour or vinegar-like taste to the wine.
3. citric acid: A natural preservative that can add a fresh, citrusy note to the wine.
4. residual sugar: Sugar that remains after fermentation. It's more relevant in sweeter wines.
5. chlorides: Refers to the salt content in the wine.
6. free sulfur dioxide: This is unbound SO_2 that helps protect wine from harmful microbes.
7. total sulfur dioxide: The total amount of SO_2 , including both free and bound forms.
8. density: The density of wine, which is influenced by sugar and alcohol levels.
9. pH: A measure of the wine's acidity (lower pH = more acidic).
10. sulphates: Added to improve preservation and stability; also influences taste.
11. alcohol: Indicates the alcohol content of the wine, expressed as a percentage.
12. quality: This is the target variable—it's the final wine quality score given by experts, ranging from 0 to 10.

Step 3: Importance of Features in Predicting Wine Quality

Not every feature contributes equally to predicting wine quality. Some have a major impact, while others are less influential.

1. Alcohol is very important. Generally, higher alcohol content is linked to better wine quality.
2. Volatile acidity is also very important but in a negative way. High levels make the wine taste sour, which lowers its quality.
3. Sulphates play a moderately important role. They help preserve wine and improve its stability, which can enhance the taste.
4. Citric acid has a moderate effect. It adds a fresh taste and improves flavor in the right amounts.
5. Residual sugar has a low to moderate impact. It's more relevant in sweet wines and doesn't influence the taste much in dry wines.
6. Chlorides have low importance. High salt content usually reduces the appeal of wine.
7. Free sulfur dioxide has low influence. It prevents spoilage but can make the wine taste harsh if overused.
8. Total sulfur dioxide has low to moderate importance. Too much can affect the aroma and taste negatively.
9. Density has low importance. It's related to other factors like sugar and alcohol, so it doesn't provide much extra information.
10. pH has low impact. While it reflects acidity, its effect on taste is indirect.
11. Fixed acidity has low to moderate importance. It contributes to sourness but varies depending on the wine type.

The most useful features when predicting wine quality are alcohol, volatile acidity, sulphates, and citric acid.

Step 4: Handling Missing Data and Common Imputation Techniques with Advantages and Disadvantages

The Wine Quality dataset on Kaggle is generally clean, but in practical situations, missing values can appear due to data merging, corruption, or preprocessing errors. When this happens, it's important to handle the missing data properly to avoid misleading analysis or model results.

The most common imputation techniques used to fill in missing values are as follow:

1. Mean/Median Imputation

Fills missing values with the mean or median of the column.

Advantages:

- Easy and fast to implement.
- Preserves the general structure and scale of the data.

Disadvantages:

- Can reduce the natural variability of the data.
- Mean imputation is sensitive to outliers; median is better for skewed data.
- Does not consider relationships between features.

2. Mode Imputation

Replaces missing values with the most frequent value in the column (mainly for categorical data).

Advantages:

- Simple and effective for categorical features.
- Preserves the most common category.

Disadvantages:

- Not suitable for numerical or continuous data.
- May add bias if one value dominates.

3. K-Nearest Neighbors (KNN) Imputation

Uses the values of the nearest data points (neighbors) to estimate and fill in missing data.

Advantages:

- Consider patterns and relationships between features.
- Can provide more accurate estimations in structured datasets.

Disadvantages:

- Slower on large datasets.
- Results depend on the number of neighbors (k) and the distance metric used.
- Needs all features to be scaled properly.

4. Multiple Imputation by Chained Equations (MICE)

Builds models to predict missing values based on other features, cycling through each missing feature in rounds.

Advantages:

- Maintains relationships among variables.
- Produces multiple complete datasets, giving a sense of uncertainty and variation.

Disadvantages:

- More complex and slower than basic methods.

- Assumes that data is missing at random, which may not always be true.

5. Dropping Rows or Columns

Simply removes any row or column with missing data.

Advantages:

- Very easy to implement.
- No need for complex calculations.

Disadvantages:

- Can lead to data loss.
- Risk of bias if the missing values are not random.

Choosing the right imputation method depends on:

- The amount and type of missing data
- The size of the dataset
- Whether the data is numerical or categorical
- The importance of preserving feature relationships

In small datasets like Wine Quality, median or KNN imputation is often a good starting point, balancing simplicity and reliability.