AIDS-I 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

Find the Mean, Median, Mode, and Interquartile Range (IQR).

Ans:

Step 1: Sort the Data
Sorted Data: 59, 64, 66, 70, 76, 76, 76, 78, 79, 81, 82, 82, 84, 85, 88, 90, 90, 91, 95, 99

Step 2: Mean
Mean formula: Mean = Sum of all values / Number of values
Sum = 1621
Number of values = 20
Mean = 1621 / 20 = 81.05

Step 3: Median
Median is the average of the 10th and 11th values in the sorted data.
10th value = 81, 11th value = 82
Median = (81 + 82) / 2 = 81.5

Step 4: Mode
The number that appears most frequently is 76 (3 times).
Mode = 76

Step 5: Interquartile Range (IQR)
Q1 is the median of the first half (1st to 10th values):
Q1 = (76 + 76) / 2 = 76
Q3 is the median of the second half (11th to 20th values):
Q3 = (88 + 90) / 2 = 89
IQR = Q3 - Q1 = 89 - 76 = 13

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

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.

Ans:

Both tools are designed to democratize machine learning by lowering the barrier to entry, making it accessible to non-experts. Let's analyze each.

Machine Learning for Kids:

Target Audience: This tool primarily targets school-aged children (roughly ages 8–16), as well as educators and parents who want to introduce foundational AI and machine learning concepts in an educational setting. It's ideal for those with little to no prior technical knowledge.
Use by Target Audience: Children use the tool to create simple machine learning models through interactive, visual interfaces like Scratch or basic Python. They might classify text (e.g., positive/negative sentiment) or images (e.g., cats/dogs) by providing examples and letting the tool learn patterns. Educators integrate it into lesson plans to teach computational thinking, logic, and AI literacy, often as part of STEM curricula. The focus is on experiential learning, where users build intuition about how machines "learn" from data.

Benefits:
Extremely user-friendly, with no coding required for basic use, making it approachable for beginners.
Promotes creativity and critical thinking by allowing users to experiment with real-world problems (e.g., sorting, prediction).
Provides a safe, controlled environment for learning complex concepts like algorithms and data patterns.

Drawbacks:
Limited in scope; it's not designed for advanced machine learning tasks or large-scale applications.
Performance can be inconsistent if the training data is small or poorly curated, which might confuse learners.
Primarily educational, not suitable for professional or production-level machine learning projects.

Teachable Machine:

Target Audience: This tool targets a broader audience, including beginners, hobbyists, artists, teachers, and even small-scale developers who want to experiment with machine learning without coding expertise. It's particularly appealing to those interested in quick prototyping or creative projects.
Use by Target Audience: Users train models directly via a web interface using their webcam, microphone, or uploaded files (e.g., images, sounds, poses). For example, an artist might train a model to recognize different gestures for an interactive installation, while a teacher might use it to demonstrate object recognition in class. The tool allows real-time feedback, enabling users to see how their model performs and adjust training data accordingly. Models can be exported for use in larger projects, like web applications.

Benefits:
No coding knowledge is needed, making it highly accessible and fast to use.
Offers real-time training and testing, which is engaging and allows for immediate iteration.
Supports a variety of input types (images, sounds, poses), broadening its applicability.
Easy export options (e.g., to TensorFlow.js) allow users to integrate models into other projects.

Drawbacks:
Limited in handling complex datasets or advanced machine learning techniques, restricting its use to simple applications.
Reliant on user hardware and internet connectivity, which can affect performance.
Not ideal for large-scale or production environments due to its simplicity and lack of customization.

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

Predictive Analytic
Descriptive Analytic

Ans: Both Machine Learning for Kids and Teachable Machine are best described as predictive analytic tools. To understand this, consider the fundamental purpose of each tool: they enable users to train models on historical or labeled data (inputs paired with outputs) and then use those models to predict outcomes for new, unseen data. This is the essence of predictive analytics—using past data to forecast future events or classify new inputs.

For instance, in Machine Learning for Kids, a child might train a model to predict whether a piece of text is “happy” or “sad” based on examples they provide. Similarly, in Teachable Machine, a user might train a model to predict whether a webcam image shows a “cat” or a “dog.” In both cases, the tools are not merely summarizing past data (which would be descriptive analytics) but are focused on making forward-looking predictions. This distinction arises from their design: they emphasize model training for classification or regression tasks rather than data exploration or visualization. Hence, predictive analytics is the appropriate category.

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

Ans: Both Machine Learning for Kids and Teachable Machine are examples of supervised learning tools.Consider the learning paradigm: supervised learning involves training a model on a dataset where the inputs are paired with correct outputs (labels). The model learns to map inputs to outputs by minimizing the error between its predictions and the true labels.

In both tools, users provide labeled examples—such as tagging images as “cat” or “dog” in Teachable Machine, or labeling text as “positive” or “negative” in Machine Learning for Kids. The tools then use these labeled datasets to train models that can generalize to new, unlabeled data. This process contrasts with unsupervised learning, which finds patterns in data without labels (e.g., clustering), and reinforcement learning, which involves an agent learning through trial and error to maximize a reward. Since both tools rely on pre-labeled data and focus on prediction based on those labels, they clearly fall under supervised learning.

Q.3 Data Visualization:

Read the following two short articles:

Kakande, Arthur. February 12. “What’s in a chart? A Step-by-Step guide to Identifying Misinformation in Data Visualization.” Medium
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.

Ans:

In April 2024, social media platforms saw a rise in climate change sceptics circulating claims suggesting that Arctic sea ice was not decreasing, and therefore, global warming concerns were exaggerated. These assertions were based on a data visualization comparing sea ice extent on January 8, 2004, and January 8, 2024. At first glance, the chart appeared to show minimal difference, implying that the Arctic ice had remained stable or even improved.

Source: Reuters – *Climate change sceptics use misleading Arctic ice data to make case*, April 25, 2024

How the Data Visualization Misled:

1. **Cherry-Picked Dates:**
   The chart selected two isolated data points 20 years apart, ignoring seasonal and year-to-year variability. This selective comparison masked the long-term downward trend in Arctic sea ice levels.

2. **Lack of Context:**
   The visual did not reference broader satellite data, which consistently show a significant decline in Arctic sea ice since 1979. Without context, viewers could not assess whether the snapshots were representative or anomalous.

3. **Misleading Visual Design:**
   The comparison relied on images and graphs that visually downplayed differences.

Small but significant changes in ice area were portrayed as negligible, potentially swaying public opinion toward climate skepticism.

Clarifying the Misrepresentation:

When viewed through long-term, continuous data provided by the National Snow and Ice Data Center (NSIDC) and NASA, it becomes clear that Arctic sea ice is in persistent decline. Experts pointed out that visualizations relying on two points in time can be deeply misleading without showing trends or averages. The broader dataset reveals shrinking ice coverage, later seasonal freezing, and earlier melting—clear signs of a warming planet.

Conclusion:

This case highlights the critical importance of providing full context, consistent scales, and representative timelines in data visualization. Misleading graphics can fuel public misunderstanding, especially in high-stakes issues like climate change. It is the responsibility of data communicators to ensure their work conveys the true story—not just one convenient frame of it.

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]

```
Validation Accuracy: 70.0 %
Test Accuracy: 75.0 %

Classification Report (Test Set):
              precision    recall  f1-score   support

           0       0.70      0.88      0.78        50
           1       0.84      0.62      0.71        50

    accuracy                           0.75       100
   macro avg       0.77      0.75      0.75       100
weighted avg       0.77      0.75      0.75       100
```

The model performs better at detecting non-diabetic patients.

- **Explanation:** This indicates that the model has a higher success rate in correctly identifying individuals without diabetes (class 0) compared to those with diabetes (class 1). This could be due to a bias toward the majority class or better-defined features for non-diabetic cases.

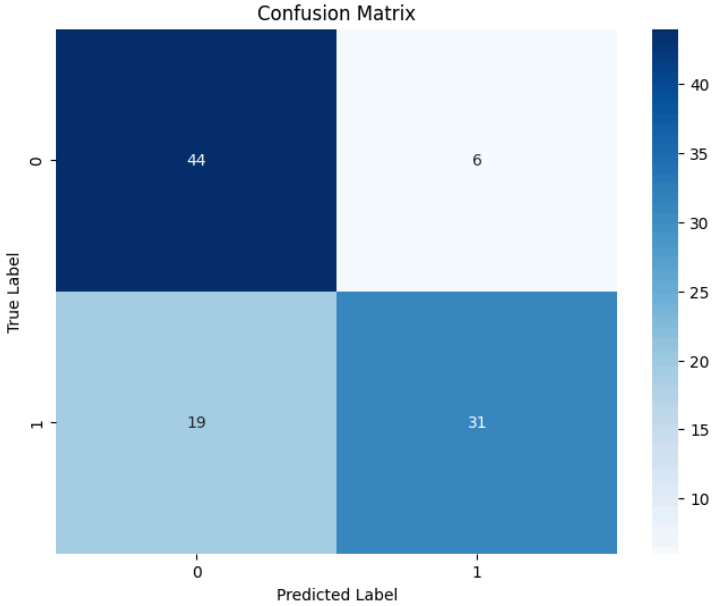
Recall for Class 1 (diabetic) is relatively low at 62%, meaning it's missing many positive cases.

- **Explanation:** Recall (also known as sensitivity or true positive rate) measures the proportion of actual positive cases (diabetic patients) that the model correctly identifies. A recall of 62% for class 1 means that 38% of diabetic cases are missed (false negatives), which is significant in a medical context where detecting all positive cases is critical. This suggests the model struggles to generalize to diabetic patients.

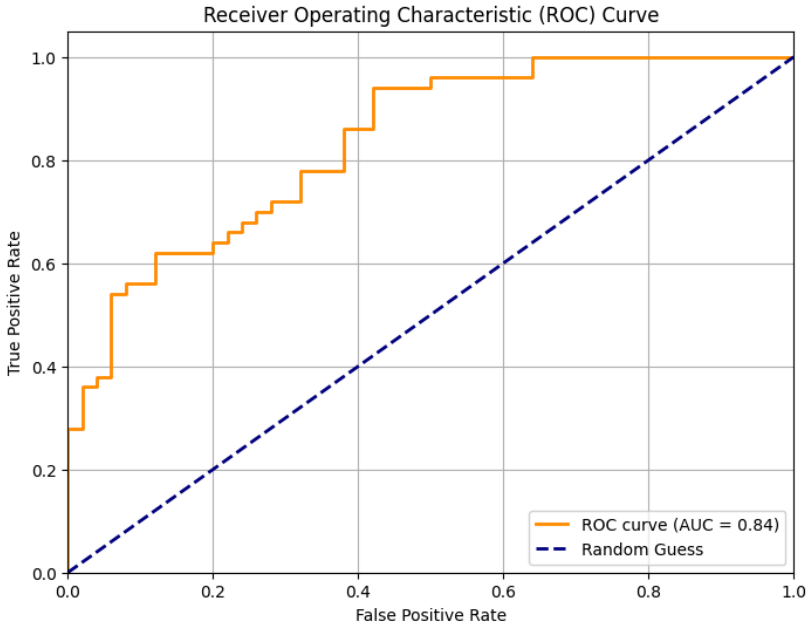
This is a common issue with imbalanced datasets, even after SMOTE.

- **Explanation:** Class imbalance occurs when one class (e.g., non-diabetic) has significantly more instances than the other (diabetic). SMOTE (Synthetic Minority Over-sampling Technique) generates synthetic samples to balance the classes, but the low recall indicates that the imbalance or feature overlap may still affect performance, possibly due to insufficient feature differentiation or model limitations.

The Naive Bayes classifier achieved a validation accuracy of 70.0% and a test accuracy of 75.0%, with an ROC AUC score of 0.84, indicating moderate to good performance in distinguishing between diabetic and non-diabetic patients.



The confusion matrix reveals that out of 100 test cases (50 per class), the model correctly identified 31 diabetic cases while misclassifying 19, which is critical in healthcare applications. Despite applying class balancing techniques, the model still shows bias toward the majority class.



The ROC curve plots the True Positive Rate (Sensitivity) against the False Positive Rate (1 - Specificity) at various threshold levels. It helps evaluate how well the model distinguishes

between the positive (diabetic) and negative (non-diabetic) classes. The orange curve represents the model's performance across different classification thresholds. The dashed diagonal line is the baseline (random guessing) — a model performing no better than chance would follow this line.

- **Explanation:**
    - **True Positive Rate (TPR or Sensitivity)** is the recall, or the proportion of actual positives correctly identified.
    - **False Positive Rate (FPR)** is the proportion of negatives incorrectly classified as positive (FP / (FP + TN)).
    - The ROC curve visualizes this trade-off, with the orange curve showing the model's performance. The baseline (diagonal) represents random guessing (AUC = 0.5), and deviation above it indicates predictive power.

The Area Under the Curve (AUC) is 0.84, which is quite good. It indicates that there's an 84% chance that the model will correctly distinguish a randomly chosen diabetic patient from a non-diabetic one.

- **Explanation:** AUC quantifies the overall ability of the model to discriminate between classes. An AUC of 0.84 is well above random (0.5) and approaches excellent (0.9+), suggesting the model is effective but could be optimized further, especially for recall.

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 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

Predictive Modeling and Evaluation

Dataset Overview

The dataset contains customer demographic and behavioral attributes. Below are the first few rows:

```
First few rows:
   CustomerID  Gender  Age  Annual Income (k$)  Spending Score (1-100)
0           1    Male   19                  15                      39
1           2    Male   21                  15                      81
2           3  Female   20                  16                       6
3           4  Female   23                  16                      77
4           5  Female   31                  17                      40
```

Data Quality

No missing values were found in any of the columns:

```
Missing values:
CustomerID                0
Gender                    0
Age                       0
Annual Income (k$)        0
Spending Score (1-100)    0
dtype: int64
```

Model Training and Performance

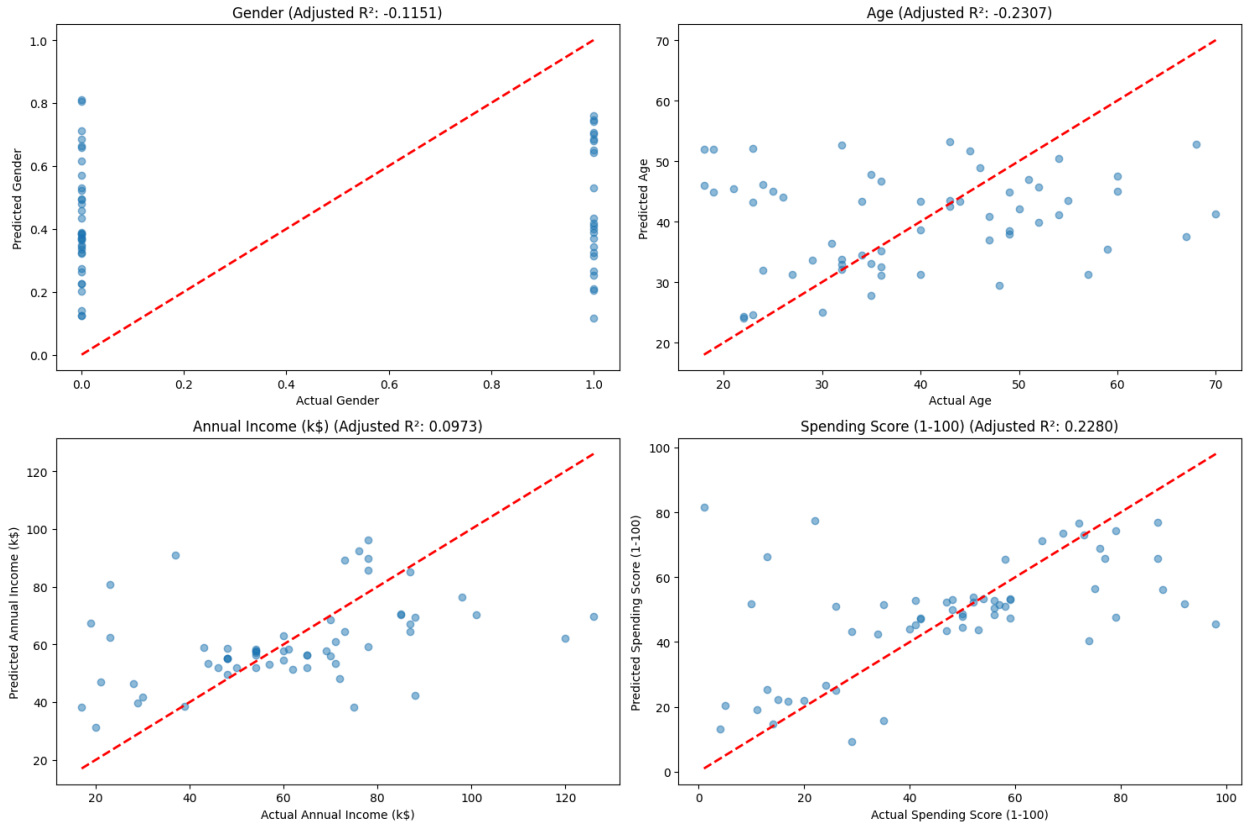
Random Forest Regressors were trained to predict each feature using the others. Hyperparameter tuning was performed using GridSearchCV to identify the best model configuration.

- **Gender**
    - **Best Parameters**: `{'max_depth': 10, 'min_samples_split': 10, 'n_estimators': 50}`
    - **Adjusted R^2 Score**: **-0.1151**
    - Interpretation: The model performed poorly in predicting gender, indicating that the other features are not informative enough to distinguish gender.
- **Age**
    - **Best Parameters**: `{'max_depth': 10, 'min_samples_split': 10, 'n_estimators': 200}`
    - **Adjusted R^2 Score**: **-0.2307**
    - Interpretation: Age prediction was also poor, suggesting a weak relationship between age and the other features used.
- **Annual Income (k$)**
    - **Best Parameters**: `{'max_depth': 10, 'min_samples_split': 10, 'n_estimators': 50}`
    - **Adjusted R^2 Score**: **0.0973**
    - Interpretation: The model shows limited ability to predict income, indicating weak linear correlations.
- **Spending Score (1–100)**
    - **Best Parameters**: `{'max_depth': 10, 'min_samples_split': 5, 'n_estimators': 50}`
    - **Adjusted R^2 Score**: **0.2280**
    - Interpretation: Spending score had the best predictability among all features, though still modest. This suggests a slightly stronger relationship with other

inputs.

Visualization of Predictions

The following figure shows the actual vs. predicted values for each feature along with the red dashed line representing the ideal prediction line (i.e., perfect correlation):



Conclusion

In this study, we applied machine learning techniques to explore the predictability of customer demographic and behavioral attributes using a Random Forest Regressor. The primary goal was to determine whether features such as Gender, Age, Annual Income, and Spending Score could be accurately predicted from one another.

We began by analyzing the dataset, which was clean and free from missing values. Each feature was individually treated as a target variable, and the rest were used as predictors. Hyperparameter tuning via GridSearchCV was employed to optimize the model for each prediction task.

The outcomes varied across features:

- **Gender and Age** yielded negative adjusted R^2 scores, indicating that the model performed worse than a simple mean-based prediction. This suggests weak or non-existent relationships between these features and the others, or that they are inherently harder to predict due to categorical nature (in the case of Gender) or noise (in the case of Age).

- **Annual Income** had a slightly positive adjusted R^2 (0.0973), reflecting some predictive potential, albeit limited.

- **Spending Score** showed the highest performance with an adjusted R^2 of 0.2280, indicating a modest but more promising relationship with the input features.

What Could Be Improved

- **Feature Engineering**: Including more relevant features such as occupation, education level, or marital status could significantly improve predictive power.

- **Categorical Treatment**: Gender was treated numerically, which may not have been optimal. A classification approach could be more appropriate.

- **Model Choice**: While Random Forest is robust, experimenting with other models like Gradient Boosting, Support Vector Machines, or Neural Networks might yield better results.

- **Dimensionality**: With only a few features, the model lacked complexity. More diverse and informative data could enable better learning.

The results underline an important insight in machine learning — not all problems are equally predictable. Some features may inherently carry more noise or less correlation with others. While this experiment provided valuable experience in supervised learning, it also highlighted the critical role of data richness and thoughtful model selection in achieving meaningful predictions.

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).

Ans:

Key Features and Their Importance:

Fixed Acidity: This measures non-volatile acids (e.g., tartaric, malic) that contribute to the wine's total acidity. High fixed acidity can enhance tartness and preservation but too much can make the wine overly sharp. It's crucial for predicting quality because it affects taste balance and microbial stability.

Volatile Acidity: This reflects volatile acids like acetic acid, which can impart a vinegar-like flavor if too high. Low levels are desirable for quality, as excessive volatile acidity is a defect. It's a key predictor because it directly impacts sensory appeal.

Citric Acid: Adds freshness and flavor, often enhancing complexity. Its presence can improve quality by balancing other acids, but too much can make the wine overly tart. It's important for fine-tuning taste profiles.

Residual Sugar: The sugar left after fermentation influences sweetness, body, and mouthfeel. Low residual sugar is typical for dry wines, while higher levels suit sweeter styles. It's critical for quality prediction as it affects consumer preference and style classification.

Chlorides: Salt content can influence taste and preservation. High chloride levels might make the wine taste salty or bitter, negatively impacting quality. It's a minor but relevant factor in overall balance.

Free Sulfur Dioxide and Total Sulfur Dioxide: These measure antioxidant and antimicrobial agents. Free SO_2 protects against oxidation, while total SO_2 includes bound forms. Both are vital for quality, as they prevent spoilage, but excessive levels can create off-odors.

Density: Related to alcohol and sugar content, density affects the wine's body and weight on the palate. It's important for predicting quality as it correlates with perceived richness and structure.

pH: Indicates acidity/basicity, affecting microbial stability and aging potential. Wines with stable pH (typically 3–4) are less prone to spoilage and better rated for quality.

Sulphates: Contribute to SO2 levels and can enhance aroma and preservation. Higher sulphates often correlate with better quality due to improved stability, but balance is key.

Alcohol: A major quality determinant, as it influences body, flavor intensity, and mouthfeel. Higher alcohol content often correlates with better ratings, but balance with other components is essential.

Each feature interacts with others, and their combined effect determines whether a wine is balanced, stable, and appealing, which drives quality scores.

Handling Missing Data in the Wine Quality Dataset:

Deletion: Remove rows or columns with missing values. This is simple but can lead to data loss, which is undesirable if the dataset is small or the missingness is significant.
Imputation with Mean or Median: Replace missing values with the mean (average) or median (middle value) of the feature. This preserves the dataset size but assumes the missing values are randomly distributed and similar to the observed data.
Mode Imputation (for Categorical Data): For categorical features (if any), use the most frequent category. This is straightforward but can overrepresent certain categories.
Advanced Techniques (e.g., KNN): Use machine learning methods like K-Nearest Neighbors (KNN) to impute based on similar data points. These are more sophisticated but computationally intensive.

Advantages and Disadvantages of Imputation Techniques:

Mean/Median Imputation:
Advantages: Easy to implement, preserves sample size, and works well for normally distributed numerical data. Mean is sensitive to outliers, so median is often preferred for descentness.
Disadvantages: Can distort variance, ignore data relationships, and assume missing values are similar to observed ones, which may not hold true.
Mode Imputation:
Advantages: Simple and effective for categorical data, maintaining the most common category without introducing new values.
Disadvantages: Can skew distributions by overrepresenting the mode, especially if the mode is not representative of the missing data.
KNN Imputation:

Advantages: Accounts for relationships between features, potentially providing more accurate imputations by using nearby data points.
Disadvantages: Computationally expensive, sensitive to the choice of neighbors (k) and distance metrics, and less effective with high-dimensional data.