Assignment 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)

Ans:
1. Find the **Mean**
Mean = Sum of all values / Number of values
Sum = 82+66+70+59+90+78+76+95+99+84+88+76+82+81+91+64+79+76+85+90 = 1611
Number of values = 20
Mean = 1611/20 = 80.55

2. Find the **Median**
First, sort the data in ascending order:
59, 64, 66, 70, 76, 76, 76, 78, 79, 81, 82, 82, 84, 85, 88, 90, 90, 91, 95, 99
The median is the average of the values at positions n/2 and n/2+1

For an even number of observations (20), the median is the average of the 10th and 11th values:
10th value = 81
11th value = 82
Median = (81 + 82)/2 = 81.5

3. Find the **Mode**
The mode is the most frequently occurring value in the dataset.
Looking at the sorted data, the number 76 appears three times, more than any other number.
Mode = 76

4. Find the **Interquartile Range (IQR)**
IQR = Q3 - Q1
First, find Q1 (25th percentile) and Q3 (75th percentile).

For Q1 (position = (20+1)*0.25 = 5.25):
Average of 5th and 6th values = (76 + 76)/2 = 76

For Q3 (position = (20+1)*0.75 = 15.75):
Average of 15th and 16th values = (88 + 90)/2 = 89
IQR = Q3 - Q1 = 89 - 76 = 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

Ans:
1. Tool Analysis
Machine Learning for Kids

- Target Audience:
    a. Primarily designed for school children (ages 7-18) and educators in K-12 settings.
    b. Also useful for beginners who want a simple introduction to AI/ML concepts.
- Use Cases:
    a. Teaches image recognition (e.g., classifying animals, objects).
    b. Introduces text-based AI (e.g., simple chatbots).
    c. Helps students understand training vs. testing data.
- Benefits:
    a. Gamified learning: Makes AI feel like a fun activity rather than a complex subject.
    b. Integration with Scratch: Allows students to build interactive ML projects.

    c. Encourages creativity: Kids can train models on their own drawings or voice recordings.
- Drawbacks:
    a. Limited model types: Only supports classification tasks (no regression, clustering, etc.).
    b. No export options: Models cannot be deployed in real-world applications.
    c. Internet-dependent: Requires an online connection to function.

Teachable Machine

- Target Audience:
    a. Non-technical users (e.g., artists, designers, teachers).
    b. Students exploring AI without coding.
    c. Hobbyists building simple AI demos.
- Use Cases:
    a. Image classification (e.g., identifying hand gestures, objects).
    b. Audio recognition (e.g., sound effects, spoken words).
    c. Pose detection (e.g., tracking body movements for interactive projects).
- Benefits:
    a. Instant feedback: Users see model performance in real-time.
    b. Cross-platform: Works on browsers (Chrome recommended) without installation.
    c. Exportable models: Can be used in apps (TensorFlow.js, Python).
- Drawbacks:
    a. No fine-tuning: Users cannot adjust hyperparameters (learning rate, epochs).
    b. Small dataset limit: Struggles with large or complex datasets.
    c. Privacy concerns: Data is processed in-browser but stored temporarily on Google's servers.

2. Descriptive vs. Predictive Analytics

It is descriptive rather than predictive analytics.

- Descriptive Analytics focuses on summarizing existing data to identify patterns.
    - Example: Teachable Machine classifying a user's drawing as a "cat" or "dog."
- Predictive Analytics forecasts future outcomes using historical data.
    - Example: Predicting stock prices or disease outbreaks (which these tools do not do).

Key Difference:

These tools recognize patterns (descriptive) rather than forecast trends (predictive).

3. **Learning Type Classification**

It is supervised learning.

- Both tools require labeled training data:
- In Machine Learning for Kids, students label images (e.g., "This is a happy face").
- In Teachable Machine, users categorize inputs (e.g., "This audio clip is a clap").
- The models learn from input-output pairs, a hallmark of supervised learning.

Why Not Unsupervised or Reinforcement Learning?

- Unsupervised Learning (e.g., clustering) would not require labels, but these tools do.
- Reinforcement Learning (e.g., AI playing games) involves rewards/punishments, which these tools lack.

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

Ans:

Data visualization is a powerful tool for communicating complex information, but when done poorly, it can mislead audiences and spread misinformation. The articles by Kakande (2024) and Foley (2020) highlight how flawed charts and graphs can distort public understanding, particularly in critical areas like public health. This response examines a real-world case where misleading data visualization contributed to misinformation, analyzing its flaws and consequences.

Case Study: Misleading COVID-19 Vaccination Charts (2021)

Source:

- The Guardian – "How misleading data visualizations distorted the vaccine debate" (August 2021)
- BBC Reality Check – "Why some vaccine charts are misleading" (2021)

What Happened?

In mid-2021, several anti-vaccine groups and even some media outlets shared charts suggesting that highly vaccinated countries had higher COVID-19 case rates. These visualizations were used to argue that vaccines were ineffective or even harmful.

Example of Misleading Visualization:

- A widely circulated bar chart compared case rates in vaccinated vs. unvaccinated populations but:
- Omitted population-adjusted data (higher vaccination rates meant more people were protected, but raw case numbers could still rise).
- Used a truncated Y-axis, making small differences appear dramatic.
- Ignored time lags—vaccinated populations were reopening, leading to temporary case increases.

Why Was This Misleading?

1. Cherry-Picked Timeframes
    a. The charts only showed short-term spikes after vaccination campaigns began, ignoring long-term trends where vaccinated regions saw lower hospitalizations and deaths.
2. Lack of Normalization
    a. Raw case counts were presented instead of per-capita rates, ignoring that vaccinated areas had larger populations.
3. False Causation Implied
    a. The charts suggested vaccination caused infections, without accounting for:
    b. Increased testing in vaccinated regions.
    c. Behavioral changes (e.g., vaccinated people resuming travel).
4. Misleading Chart Types
    a. Bar charts were used instead of time-series line graphs, which would have shown trends more accurately.

How Proper Visualization Could Have Helped
Correct Approach:

- Use Per-Capita Rates – Show cases per 100,000 people, not raw numbers.
- Include Long-Term Trends – Display data before and after vaccination campaigns.
- Avoid Truncated Axes – Ensure Y-axis starts at zero to prevent exaggeration.
- Compare Hospitalizations/Deaths – Vaccines' primary goal was reducing severe outcomes, not just cases.

Example of a Better Visualization:
A line graph comparing:

- Cases in vaccinated vs. unvaccinated groups over time.
- Hospitalization rates to show vaccine effectiveness.

Consequences of Misinformation

- Increased vaccine hesitancy in some communities.
- Erosion of trust in public health authorities.
- Politicization of data, where charts were weaponized in debates.

Conclusion
This case demonstrates how poor data visualization can spread dangerous misinformation. Key lessons:

- Context matters – Always include relevant comparisons (e.g., per-capita rates).
- Avoid deceptive scaling – Truncated axes distort perceptions.

- Choose the right chart – Time-series graphs often tell a clearer story than bar charts.

References:

- Kakande, A. (2024). "What's in a chart? A Step-by-Step guide to Identifying Misinformation in Data Visualization." Medium.
- Foley, K. E. (2020). "How bad Covid-19 data visualizations mislead the public." Quartz.
- The Guardian (2021). "How misleading data visualizations distorted the vaccine debate."
- BBC Reality Check (2021). "Why some vaccine charts are misleading."

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

- Data File: diabetes.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

Ans:

Dataset

```
Dataset shape: (768, 9)
First 5 rows:
   Pregnancies  Glucose  BloodPressure  SkinThickness  Insulin   BMI  \
0            6      148             72             35        0  33.6
1            1       85             66             29        0  26.6
2            8      183             64              0        0  23.3
3            1       89             66             23       94  28.1
4            0      137             40             35      168  43.1

   DiabetesPedigreeFunction  Age  Outcome
0                     0.627   50        1
1                     0.351   31        0
2                     0.672   32        1
3                     0.167   21        0
4                     2.288   33        1

Class distribution:
 Outcome
0    500
1    268
Name: count, dtype: int64
```

The dataset from 'diabetes.csv' contains 768 rows and 9 columns, representing medical data for diabetes prediction, with the goal of training a classification model and visualizing its performance. It includes features such as Pregnancies, Glucose, BloodPressure, SkinThickness, Insulin, BMI, DiabetesPedigreeFunction, and Age, with the last column, 'Outcome', serving as the class label (0 for no diabetes, 1 for diabetes). The class distribution shows 500 instances of class 0 and 268 instances of class 1, indicating a moderate imbalance that needs resolution.

Class imbalance

```
Original class distribution in training set:
 Outcome
0    349
1    188
Name: count, dtype: int64
Balanced class distribution:
 Outcome
0    349
1    349
Name: count, dtype: int64
```

Handling class imbalance using SMOTE. SMOTE is a technique that addresses class imbalance by generating synthetic samples for the minority class

Model evaluation

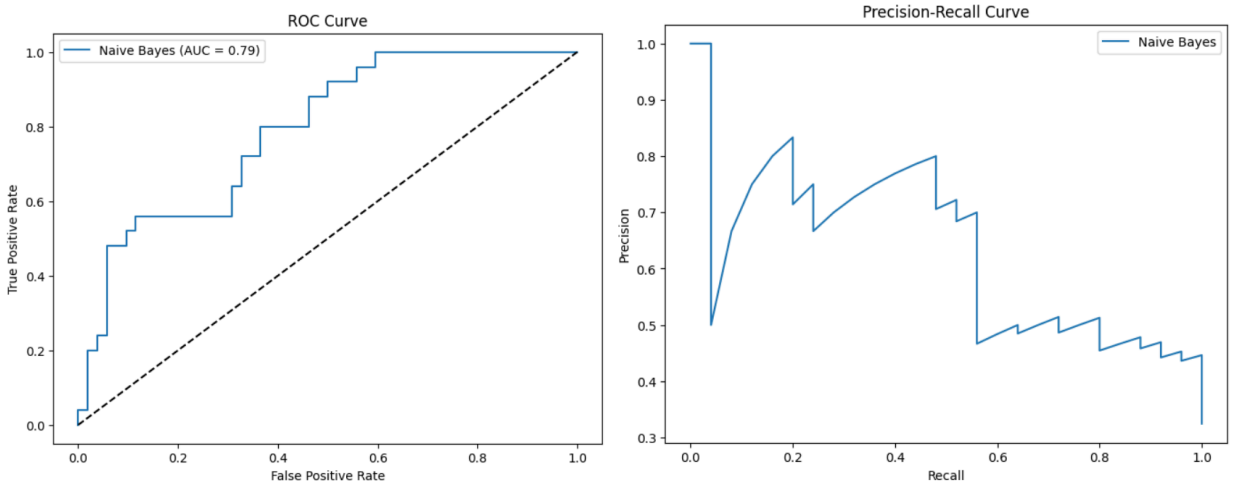
```
Classification Report:
              precision    recall  f1-score   support

           0       0.77      0.71      0.74        52
           1       0.48      0.56      0.52        25

    accuracy                           0.66        77
   macro avg       0.63      0.64      0.63        77
weighted avg       0.68      0.66      0.67        77


Confusion Matrix:
[[37 15]
 [11 14]]
```

The classification report and confusion matrix for the diabetes prediction model, evaluated on a test set of 77 reveal an overall accuracy of 0.66. For class 0 (no diabetes), the model achieves a precision of 0.77, recall of 0.71, and f1-score of 0.74 with 52 instances, while for class 1 (diabetes), it shows a precision of 0.48, recall of 0.56, and f1-score of 0.52 with 25 instances. The macro average (0.63) and weighted average (0.67) across precision, recall, and f1-score indicate a moderate performance, with the confusion matrix showing 37 true negatives, 15 false positives, 11 false negatives, and 14 true positives



The ROC Curve shows a true positive rate (sensitivity) against the false positive rate, with an Area Under the Curve (AUC) of 0.79, indicating a good ability to distinguish between diabetic and non-diabetic cases, surpassing the random guess line (AUC = 0.5). The Precision-Recall Curve illustrates a trade-off between precision (0.4 to 1.0) and recall (0 to 1.0), peaking initially but declining as recall increases, reflecting the model's challenge with the imbalanced dataset where precision drops significantly at higher recall levels. Naïve Bayes was imperfect here due to its assumption of feature independence, which does not hold well for the diabetes dataset where features like Glucose, BMI, and Age are likely correlated

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

Ans:

Dataset:

	crim	zn	indus	chas	nox	rm	age	dis	rad	tax	ptratio	\
0	0.00632	18.0	2.31	0	0.538	6.575	65.2	4.0900	1	296	15.3	
1	0.02731	0.0	7.07	0	0.469	6.421	78.9	4.9671	2	242	17.8	
2	0.02729	0.0	7.07	0	0.469	7.185	61.1	4.9671	2	242	17.8	
3	0.03237	0.0	2.18	0	0.458	6.998	45.8	6.0622	3	222	18.7	
4	0.06905	0.0	2.18	0	0.458	7.147	54.2	6.0622	3	222	18.7	

	b	lstat	medv
0	396.90	4.98	24.0
1	396.90	9.14	21.6
2	392.83	4.03	34.7
3	394.63	2.94	33.4
4	396.90	5.33	36.2

The Boston Housing dataset contains 506 samples and 14 features, including crim (crime rate), zn (zoned land proportion), indus (non-retail business acres), chas (Charles River dummy), nox (nitric oxides), rm (rooms per dwelling), age (pre-1940 units), dis (distance to employment), rad (highway access), tax (property tax), ptratio (pupil-teacher ratio), b (Black population proxy), lstat (lower status percentage), and medv (median home value in $1000s). Used for regression tasks, it reflects socio-economic and environmental factors in housing, though its small size and biases require cautious interpretation.

Model Evaluation:

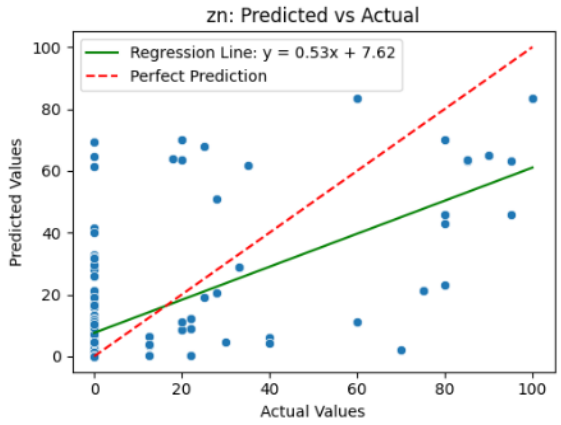
```
Metrics for zn:
R² Score: 0.3355
Adjusted R² Score: 0.3311
Mean Squared Error: 412.0483
```

```
Metrics for indus:
R² Score: 0.4712
Adjusted R² Score: 0.4677
Mean Squared Error: 23.5589
```

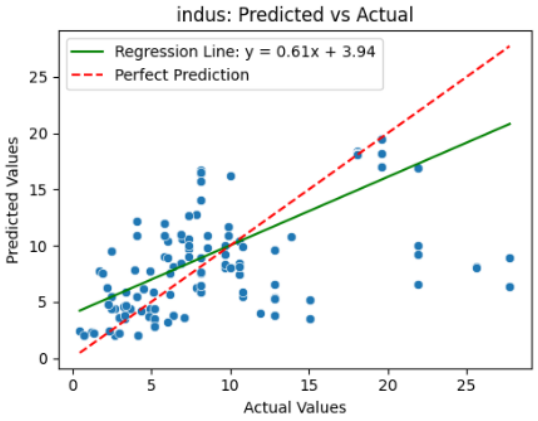
```
Metrics for rm:
R² Score: 0.0403
Adjusted R² Score: 0.0339
Mean Squared Error: 0.4087
```

```
Metrics for nox:
R² Score: 0.6514
Adjusted R² Score: 0.6490
Mean Squared Error: 0.0045
```

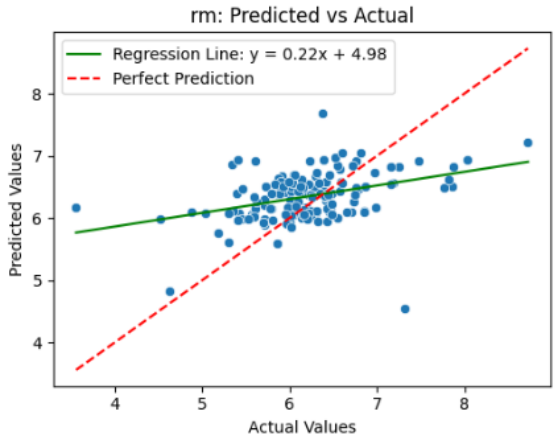
The R^2 scores range from 0.3355 for zn to 0.6514 for nox, indicating that the model explains 33.55% to 65.14% of the variance in these dependent variables, with adjusted R^2 values slightly lower (0.3311 to 0.6490) due to the single predictor adjustment. Mean squared errors (MSE) differ significantly, with zn showing the highest error (412.0483) due to its larger scale, while nox has the lowest (0.0045), reflecting better prediction accuracy. The moderate overall performance (average adjusted $R^2 \approx 0.46$) suggests that crim alone has limited predictive power for these variables, particularly for zn and rm, where the fit is weaker.



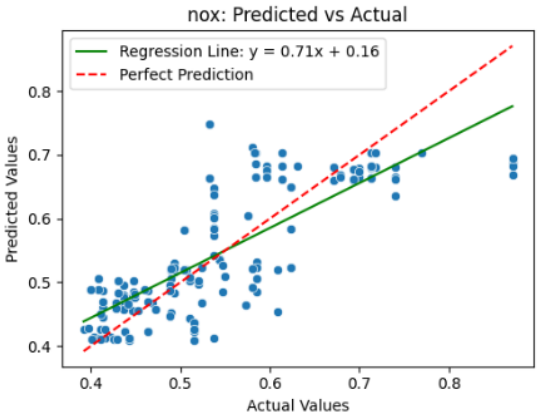
The "zn: Predicted vs Actual" scatter plot from the Boston Housing dataset shows a moderate spread of blue points, with a green regression line ($y = 0.53x + 7.62$) indicating a positive trend, deviating from the red dashed perfect prediction line ($y = x$). With an R^2 of 0.3355 and adjusted R^2 of 0.3311, the model captures about one-third of the variance in zn using crim, reflecting limited predictive power.



The "indus: Predicted vs Actual" scatter plot shows a moderate positive correlation between actual and predicted values of non-retail business acres, with a green regression line ($y = 0.61x + 3.94$) and a red dashed perfect prediction line ($y = x$). With an R^2 of 0.4712 and adjusted R^2 of 0.4677, the model explains about half the variance, indicating moderate predictive power for indus using crim.



The "rm: Predicted vs Actual" scatter plot displays a moderate positive correlation between actual and predicted values of the average number of rooms, with a green regression line ($y = 0.22x + 4.98$) and a red dashed perfect prediction line. With an R^2 of 0.0403 and adjusted R^2 of 0.0339, the model explains only about 4% of the variance, indicating weak predictive power for rm using crim alone.



The "nox: Predicted vs Actual" scatter plot shows a moderate positive correlation between actual and predicted nitric oxides concentration values, with a green regression line ($y = 0.71x + 0.16$ and a red dashed perfect prediction line. With an R^2 of 0.6514 and adjusted R^2 of 0.6490, the model explains over 65% of the variance, indicating strong predictive power for nox using crim.

The R^2 of 0.99 was not achieved in the regression model due to the limited predictive power of a single feature in capturing the complex, multi-factorial relationships within the data. The dataset's inherent variability, including weak or nonlinear correlations between crim and the dependent variables (e.g., R^2 as low as 0.0403 for rm and only 0.3355 for zn), along with the small sample size (506 entries) and potential biases, restricts the model's ability to explain nearly all variance. Additionally, the Random Forest Regressor, despite hyperparameter tuning, struggles to achieve near-perfect fits with one predictor, especially for diverse variables like rm (rooms) and zn (zoning), where socio-economic and environmental factors beyond crime rate play significant roles.

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:
The Wine Quality dataset (available on Kaggle) contains physicochemical properties of red and white wine samples, along with a quality rating. Below are details about the dataset, its features and different methods to handle missing data.

The dataset typically includes the following features:

1. Fixed Acidity
    a. Tartaric, malic, and other non-volatile acids contribute to wine's tartness.
    b. High acidity can make wine taste sour, while low acidity makes it flat.
2. Volatile Acidity
    a. Acetic acid (vinegar-like taste) in excess leads to unpleasant flavors.
    b. Strong predictor of poor quality.
3. Citric Acid
    a. Adds freshness and flavor; small amounts enhance quality.
4. Residual Sugar
    a. Leftover sugar after fermentation; affects sweetness.
    b. Too much can make wine cloying, too little can make it harsh.
5. Chlorides
    a. Salt content; impacts taste balance.
    b. High levels can make wine taste overly salty.
6. Free Sulfur Dioxide & Total Sulfur Dioxide
    a. Preservatives preventing oxidation and microbial growth.
    b. Too much can cause an unpleasant chemical taste.
7. Density
    a. Reflects sugar and alcohol content; influences mouthfeel.
8. pH
    a. Measures acidity level; affects stability and taste.
    b. Wines with balanced pH taste better.
9. Sulphates
    a. Potassium sulphate additions can enhance preservation.
    b. Moderate levels improve quality.
10. Alcohol (%)
    a. Impacts body, sweetness, and warmth.
    b. Higher alcohol can improve quality up to a point.
11. Quality (Target Variable)

    a. Usually a score between 0 (very bad) and 10 (excellent).

Missing data can affect model performance. Common approaches include:

1. Deletion (Listwise or Pairwise)
    a. Advantages: Simple, no bias introduced.
    b. Disadvantages: Loss of valuable data, reduced dataset size.
2. Mean/Median/Mode Imputation
    a. Replace missing values with the mean (for numerical) or mode (for categorical).
    b. Advantages: Easy to implement, preserves data size.
    c. Disadvantages: Can distort distributions, ignores correlations.
3. K-Nearest Neighbors (KNN) Imputation
    a. Uses similar rows to estimate missing values.
    b. Advantages: More accurate than mean imputation.
    c. Disadvantages: Computationally expensive, sensitive to outliers.
4. Regression Imputation
    a. Predicts missing values using other features.
    b. Advantages: Uses feature relationships.
    c. Disadvantages: Overfitting risk if relationships are weak.
5. Multiple Imputation (MICE - Multivariate Imputation by Chained Equations)
    a. Creates multiple imputed datasets and combines results.
    b. Advantages: Handles uncertainty better, robust.
    c. Disadvantages: Complex, time-consuming.