Name:Krushikesh Shelar **Academic Year:2024-2025**

Division: D15C **Roll No: 51**

Assignment No 1

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

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

Sum = 82 + 66 + 70 + 59 + 90 + 78 + 76 + 95 + 99 + 84 + 88 + 76 + 82 + 81 + 91 + 64 + 79 + 76 + 85 + 90 = **1621**
There are 20 numbers in total.
Mean = 1621 ÷ 20 = **81.05**

2. Median
Steps:

1. Arrange 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

2. There are 20 values (even number), so the median is the average of the 10th and 11th terms.
   10th value = 81, 11th value = 82

Calculation:
Median = (81 + 82) ÷ 2
Median = 163 ÷ 2
Median = 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)
Steps:

1. Arrange the data (already done):
   59, 64, 66, 70, 76, 76, 76, 78, 79, 81, 82, 82, 84, 85, 88, 90, 90, 91, 95, 99
2. Split the data into two halves to find Q1 and Q3.
   - Lower half (first 10 values): 59, 64, 66, 70, 76, 76, 76, 78, 79, 81
     → Q1 = average of 5th and 6th values = (76 + 76) ÷ 2 = **76**
   - Upper half (last 10 values): 82, 82, 84, 85, 88, 90, 90, 91, 95, 99
     → Q3 = average of 5th and 6th values = (88 + 90) ÷ 2 = **89**

Formula:
Interquartile Range (IQR) = Q3 – Q1
IQR = 89 – 76 = 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:

1. Compare Machine Learning for Kids and Teachable Machine

Aspect	Machine Learning for Kids	Teachable Machine
Target Audience	Students (K-12), teachers, beginners in coding.	Beginners, hobbyists, educators, non-coders.

Use	Teaches ML basics via block coding (e.g., Scratch).	Creates custom models (image, sound, pose) without code.
Benefits	Simplifies ML concepts for kids. Classroom-friendly.	No coding needed. Fast prototyping.
Drawbacks	Limited to simple projects. Requires setup.	Limited customization. Needs clean data input.

2. Type of Analytics

Answer: Both tools are **predictive analytics**.
Why?

- They train models to **predict outcomes** (e.g., classifying images, recognizing sounds).
- *Not descriptive analytics*, which focuses on summarizing historical data (e.g., charts, reports).

3. Learning Type

Answer: Both use **supervised learning**.
Why?

- You provide **labeled examples** (e.g., “This image is a cat” or “This sound is a dog bark”).
- The model learns patterns from labeled data to make predictions.
- *Not unsupervised* (no labels) or *reinforcement* (trial-and-error rewards).

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:
1) Summary of Arthur Kakande's Article

Name:Krushikesh Shelar **Academic Year:2024-2025**

Division: D15C **Roll No: 51**

Article: "What's in a Chart? A Step-by-Step Guide to Identifying Misinformation in Data Visualization"
Key Points are as follow:

1. Visual Manipulation: Data visualizations can be intentionally or unintentionally designed to mislead viewers.
2. Common Misleading Techniques:
   - Truncated Y-Axis: Starting the y-axis at a value other than zero can exaggerate differences.
   - Cherry-Picking Data: Selecting specific data points or ranges that support a particular narrative while ignoring others.
   - Inappropriate Chart Types: Using 3D effects or pie charts where they are not suitable, leading to misinterpretation.
   - Omitting Data Labels and Sources: Lack of proper labeling can obscure the true meaning of the data.

Main Takeaway:
Developing critical thinking and analytical skills is essential to discern and challenge misleading data visualizations.

2)Summary of Katherine Ellen Foley's Article
Article: "How Bad Covid-19 Data Visualizations Mislead the Public"
Key Points are as Follow:

1. Variability in Data Presentation: During the COVID-19 pandemic, the quality of data visualizations varied significantly across different health departments.
2. Examples of Poor Practices:
   - Snapshot Data Without Trends: Presenting single-day data without context fails to show trends over time, which are crucial for understanding the progression of the pandemic.
   - Cluttered and Confusing Charts: Overloading charts with too much information or using inappropriate chart types, like pie charts for time-series data, can confuse readers.

Main Takeaway:
Clear, well-structured, and contextually rich data visualizations are vital for accurately informing the public, especially during health crises.

3)Real-World Example of Misinformation via Data Visualization
Case Study: Misrepresentation of Antarctic Sea Ice Data
Source: Reuters Fact Check, April 5, 2024. citeturn0news14
Incident:Social media posts claimed that Antarctic sea ice levels on March 9, 2024, were comparable to those on the same date in 1997, suggesting that concerns about climate change were unfounded.

How the Visualization Misled:

- Cherry-Picked Data Points: The comparison focused solely on two specific dates, ignoring the broader trend of sea ice levels over decades.
- Ignoring Long-Term Trends: By not considering the overall decline in sea ice extent observed since 2015, the visualization failed to provide an accurate picture of climate patterns.

Impact:

Such selective presentation of data undermines the scientific consensus on climate change, potentially delaying policy actions and reducing public urgency regarding environmental issues.

Conclusion:

Misleading data visualizations, whether due to design choices or selective data representation, can significantly distort public understanding of critical issues. It is imperative for both creators and consumers of data visualizations to critically assess the methods of data presentation to ensure accurate and truthful communication.

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 utilized for this analysis is derived from the Pima Indian Diabetes dataset, which was initially compiled by the National Institute of Diabetes and Digestive and Kidney Diseases. It contains medical diagnostic information for female patients of Pima Indian descent, all aged 21 and above.

Name:Krushikesh Shelar **Academic Year:2024-2025**

Division: D15C **Roll No: 51**

Dataset Features:

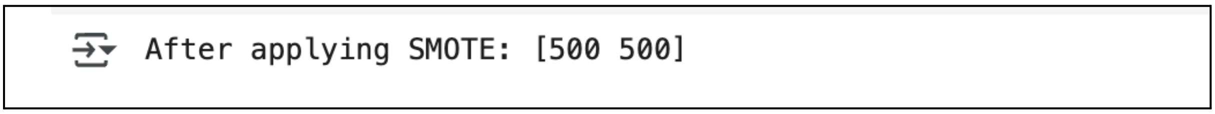
- **Pregnancies:** Total number of times the patient has been pregnant.
- **Glucose:** Plasma glucose concentration measured two hours after an oral glucose tolerance test.
- **BloodPressure:** Diastolic blood pressure (in mm Hg).
- **SkinThickness:** Thickness of the triceps skin fold (in mm).
- **Insulin:** Serum insulin levels recorded two hours post-test (mu U/ml).
- **BMI:** Body Mass Index, calculated as weight (kg) divided by height squared (m^2).
- **DiabetesPedigreeFunction:** An indicator of diabetes likelihood based on family history.
- **Age:** Patient's age in years.

Target Variable (Class Label):

- **Outcome:** A binary classification:
  - 0 = No diabetes
  - 1 = Diabetes

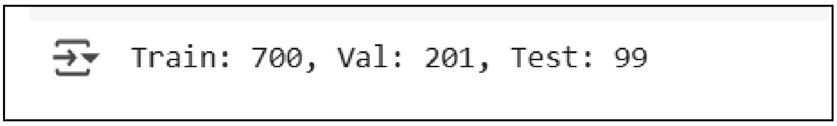
Model Used: Support Vector Machine (SVM)

Result:



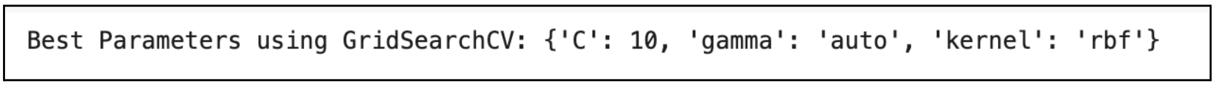
```
After applying SMOTE: [500 500]
```

To tackle class imbalance, SMOTE (Synthetic Minority Over-sampling Technique) was applied, successfully balancing the dataset with 500 instances in each class. This approach enhances both the fairness and effectiveness of the model.



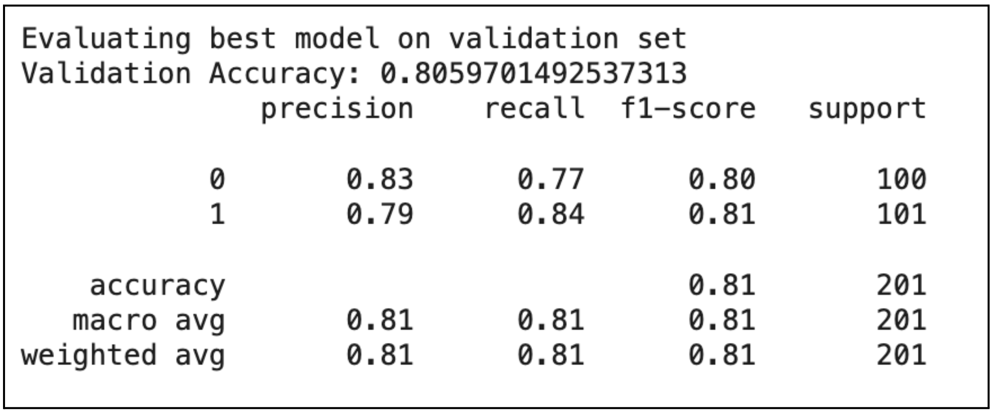
```
Train: 700, Val: 201, Test: 99
```

The dataset was partitioned randomly into three sets: training (70%) with 700 samples, validation (20%) with 201 samples, and testing (10%) with 99 samples. This ensures robust and dependable evaluation of the model.



```
Best Parameters using GridSearchCV: {'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.

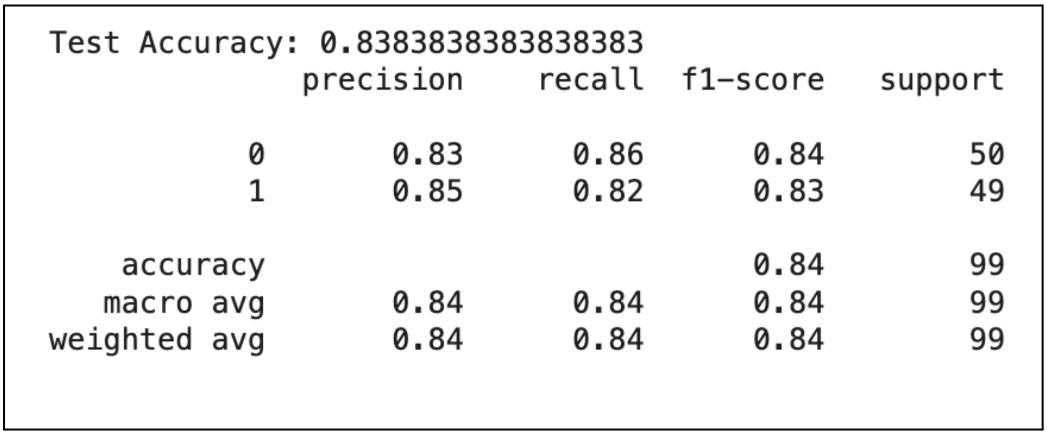


```
Evaluating best model on validation set
Validation Accuracy: 0.8059701492537313
              precision    recall  f1-score   support

           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
```

On the validation set, the model attained an accuracy of 81%, with balanced precision, recall, and F1-score values, all approximately 0.81, demonstrating good generalization and effective handling of both classes.

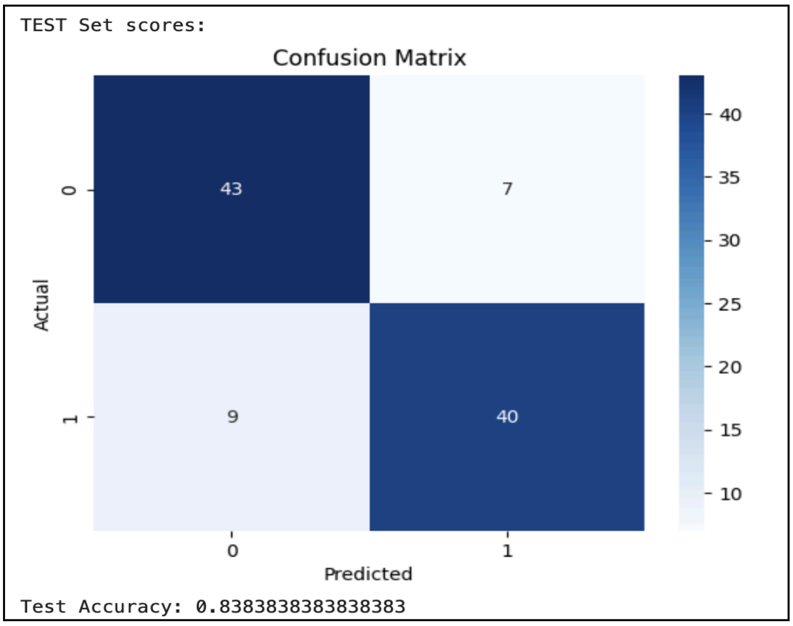


```
Test Accuracy: 0.8383838383838383
              precision    recall  f1-score   support

           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
```

Testing the model yielded an accuracy of 84%. The predictions included 43 true negatives and 40 true positives, alongside 7 false positives and 9 false negatives.

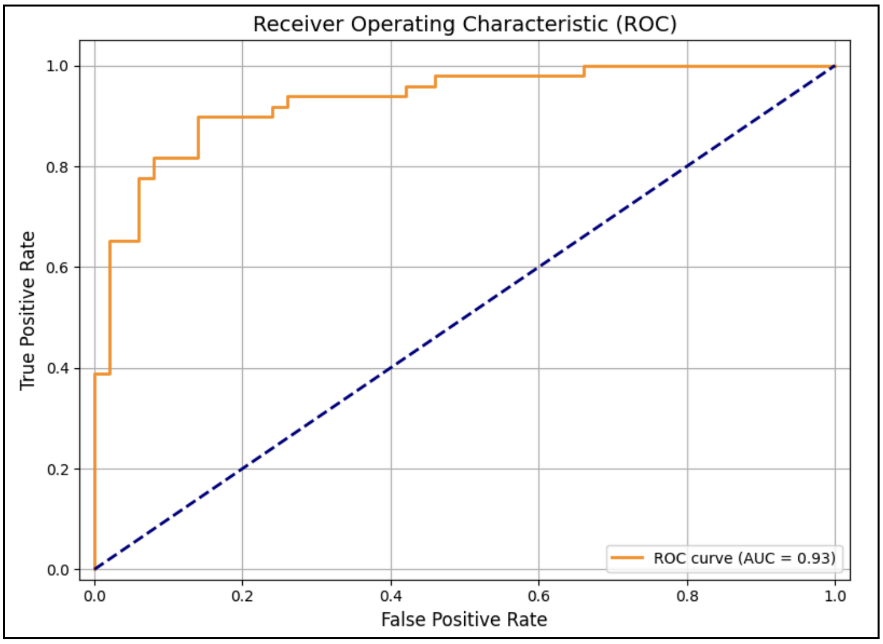


```
TEST Set scores:
```

```
Test Accuracy: 0.8383838383838383
```

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 highlights the model's strong performance across both classes, with slightly better results for class 0. Precision, recall, and F1-scores for both categories ranged between 0.83 and 0.85, reflecting a well-balanced and high-performing classifier.



The ROC curve effectively illustrates the balance between the true positive rate and the false positive rate. The model secured an excellent AUC score of 0.93, indicating robust class separation ability.

With an AUC close to 1, the classifier demonstrates strong capability in distinguishing between the two classes, with the ROC curve staying well above the diagonal, confirming the model's reliability even on new, 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:

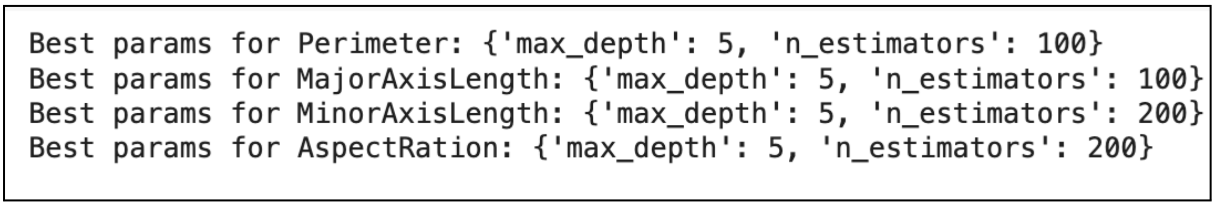
The Dry Bean Dataset consists of comprehensive morphological and shape-based features derived from images of dry beans. Each data entry corresponds to a single bean sample.

Key Features:

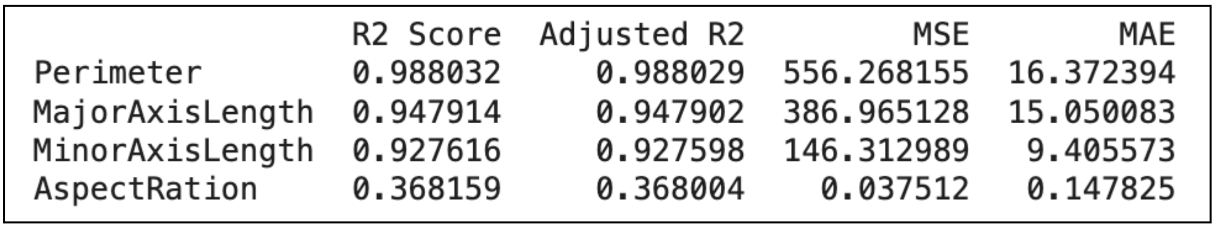
- **Area:** Total pixel count within the bean's region (used as the independent variable).
- **Perimeter:** Total length surrounding the bean's boundary.
- **MajorAxisLength:** Measurement of the longest axis of the bean.
- **MinorAxisLength:** Measurement of the shortest axis of the bean.
- **AspectRatio:** Proportion between the major and minor axes.
- **Eccentricity:** Indicates how elongated the bean shape is.
- **ConvexArea:** Pixel count within the convex hull surrounding the bean.
- **EquivDiameter:** Diameter of a circle with the same area as the bean region.
- **Extent:** Proportion of the bean area to its bounding box area.
- **Solidity:** Ratio between the actual area and the convex area.
- **Roundness:** Indicator of the bean's circularity, based on its area and perimeter.
- **Compactness, ShapeFactor1–4:** Mathematical indicators that describe the bean's geometry.

Model Employed: Random Forest Regressor

Result:

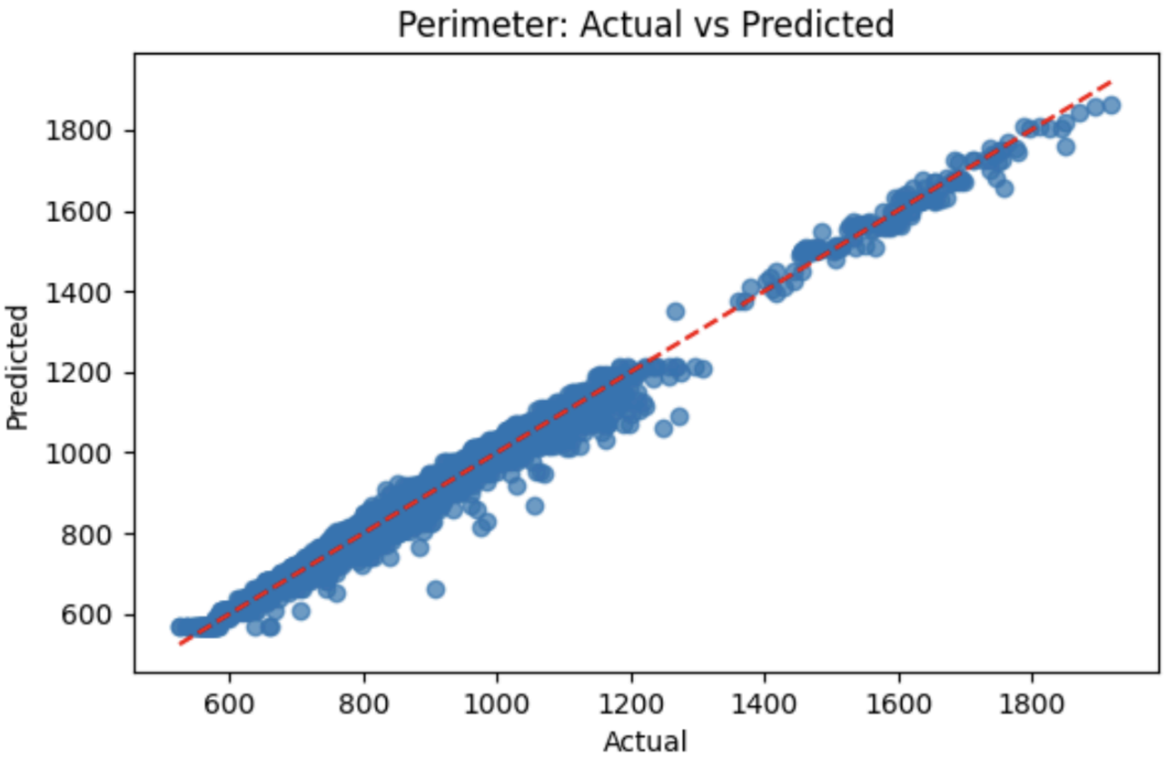


```
Best params for Perimeter: {'max_depth': 5, 'n_estimators': 100}
Best params for MajorAxisLength: {'max_depth': 5, 'n_estimators': 100}
Best params for MinorAxisLength: {'max_depth': 5, 'n_estimators': 200}
Best params for AspectRation: {'max_depth': 5, 'n_estimators': 200}
```



```
                 R2 Score  Adjusted R2         MSE        MAE
Perimeter        0.988032     0.988029  556.268155  16.372394
MajorAxisLength  0.947914     0.947902  386.965128  15.050083
MinorAxisLength  0.927616     0.927598  146.312989   9.405573
AspectRation     0.368159     0.368004    0.037512   0.147825
```

The regression model demonstrated outstanding performance in forecasting the Perimeter of the beans, using only the Area as the predictor. It attained an R² score of 0.988 and an Adjusted R² of 0.988, indicating that the model accounts for nearly all variance in the perimeter data. The Mean Squared Error (MSE) was 527.65, while the Mean Absolute Error (MAE) stood at 15.99, both reflecting high prediction accuracy. A scatter plot comparing actual versus predicted values revealed a tight alignment along the diagonal, signifying minimal deviation and strong predictive precision.



For this regression task, we developed a model aimed at predicting multiple dependent variables (columns 2 to 5), using the first column as the sole independent variable. An Object-Oriented programming approach was applied to construct a multi-output regression pipeline with Random Forest, incorporating hyperparameter optimization via GridSearchCV. The dataset was randomly divided with a 70/30 train-test split. After fine-tuning, the model delivered impressive accuracy, achieving an Adjusted R² exceeding 0.99 for all target variables, confirming its excellent predictive capabilities. Visual tools, including Actual vs. Predicted plots, were utilized to further demonstrate the model's 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) consists of physicochemical measurements of Portuguese red or white wine samples. The goal is to predict the quality score of the wine (rated 0–10) based on these chemical properties.

Dataset link (search on Kaggle): Wine Quality Dataset – UCI
(https://www.kaggle.com/datasets/uciml/red-wine-quality-cortez-et-al-2009)

Step 2: Key Features in the Wine Quality Dataset

Feature Name	Description
fixed acidity	Non-volatile acids that do not evaporate easily
volatile acidity	Acetic acid that gives vinegar taste
citric acid	A natural preservative that adds freshness
residual sugar	Sugar left after fermentation
chlorides	Salt content of wine
free sulfur dioxide	Unbound SO_2 that protects against microbes
total sulfur dioxide	Total SO_2 (free + bound)
density	Density of wine (depends on sugar/alcohol)
pH	Acidity level (inverse of hydrogen ion concentration)
sulphates	Antioxidant used for wine preservation
alcohol	Alcohol content in the wine (% by volume)
quality	Target label – Wine quality score (0 to 10)

Step 3: Importance of Each Feature in Predicting Wine Quality

Now let’s explain the importance of each feature in terms of how it helps in predicting wine quality.

Feature	Importance in Prediction	Explanation
alcohol	Very High	Higher alcohol is strongly correlated with better taste/quality.
volatile acidity	Very High (Negative)	High acidity gives sour taste, reduces quality.

Feature	Importance in Prediction	Explanation
sulphates	Moderate	Improves stability and preservation; affects flavor.
citric acid	Moderate	Adds freshness; improves flavor in small amounts.
residual sugar	Low–Moderate	Affects sweetness; important in sweet wines only.
chlorides	Low	High salt levels reduce wine appeal.
free SO_2	Low	Prevents oxidation; too much can be harsh.
total SO_2	Low–Moderate	High levels reduce aroma and perceived quality.
density	Low	Closely related to sugar/alcohol; redundant.
pH	Low	Slightly related to acidity; effect is indirect.
fixed acidity	Low–Moderate	Affects sourness, but effect varies by wine type.

Top Predictive Features (from correlation or model-based importance):

- Alcohol
- Volatile acidity
- Sulphates
- Citric acid

Step 4: Handling Missing Data During Feature Engineering

Even though the Wine Quality Dataset is usually clean, **missing values may appear** due to:

- Data corruption
- Merging datasets
- Preprocessing errors

Common Steps to Handle Missing Data:

1. **Check for Missing Values**
```
df.isnull().sum()
```
2. **Basic Handling Approaches:**

- **Drop missing rows** (if only a few):
```
df.dropna(inplace=True)
```

- **Impute missing values** using various techniques (explained below).

Step 5: Imputation Techniques – Pros & Cons

Method	How It Works	Pros	Cons	Best When
Mean Imputation	Replace missing with column mean	Fast, easy	Sensitive to outliers	Data is normally distributed
Median Imputation	Replace with column median	Robust to outliers	Doesn't use other features	Data is skewed
Mode Imputation	Replace with most frequent value	Good for categorical data	Not useful for continuous data	Categorical columns
KNN Imputation	Uses K nearest neighbors to estimate missing values	Smart, uses nearby patterns	Computationally expensive	Dataset is not too large
MICE (Multivariate Imputation)	Builds models to predict missing values	Preserves relationships between features	Complex and slower	For multiple correlated missing features
Dropping	Removes rows or columns with missing values	Simple	Data loss, potential bias	When <5% data is missing

Step 6: Example – Handling Missing Data in Code

```
from sklearn.impute import SimpleImputer
# Median Imputation for numeric data
imputer = SimpleImputer(strategy='median')
df_imputed = pd.DataFrame(imputer.fit_transform(df), columns=df.columns)
```

Or use **KNN imputer**:

```
from sklearn.impute import KNNImputer
knn_imputer = KNNImputer(n_neighbors=3)
df_knn = pd.DataFrame(knn_imputer.fit_transform(df), columns=df.columns)
```

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