Name : Nishant Sachin Khetal **Roll No : 24** **Div : D15C**

Q.1: Statistical Analysis of Given Dataset

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

1. Find the Mean

The mean is the average of all numbers. To calculate it:

- Sum all numbers: 82+66+70+59+90+78+76+95+99+84+88+76+82+81+91+64+79+76+85+90 = 1611
- Divide by count (20): 1611/20 = 80.55

Mean = 80.55

2. Find the Median

The median is the middle value when data is ordered:

1. Sort data: 59, 64, 66, 70, 76, 76, 76, 78, 79, 81, 82, 82, 84, 85, 88, 90, 90, 91, 95, 99
2. For even count (20), average the 10th and 11th values: (81 + 82)/2 = 81.5

Median = 81.5

3. Find the Mode

The mode is the most frequent value:

- 76 appears 3 times (most frequent)
- All others appear 1-2 times

Mode = 76

4. Find the Interquartile Range (IQR)

IQR = Q3 - Q1 (Q3=75th percentile, Q1=25th percentile)

1. Ordered data (same as above)
2. Q1 position: 0.25 × 20 = 5 → average 5th & 6th values: (76 + 76)/2 = 76
3. Q3 position: 0.75 × 20 = 15 → average 15th & 16th values: (88 + 90)/2 = 89
4. IQR = Q3 - Q1 = 89 - 76 = 13

Interquartile Range = 13

Q.2: Machine Learning Tools Analysis

1. Tool Analysis

Machine Learning for Kids

- **Target Audience:** Children (ages 8-16), educators with no coding background
- **Use:** Teaches basic ML concepts through simple projects like image/text recognition
- **Benefits:**
  - Simple drag-and-drop interface
  - Makes ML accessible to young learners
  - Free educational resource
- **Drawbacks:**
  - Limited to basic ML models
  - Not suitable for complex projects
  - Minimal customization options

Teachable Machine

- **Target Audience:** Beginners, students, non-technical users
- **Use:** Creates ML models (image, sound, pose) without coding
- **Benefits:**
  - No coding required
  - Quick model training in browser
  - Export models for apps/websites
- **Drawbacks:**
  - Limited model complexity
  - Requires internet connection
  - Less control over model parameters

2. Analytic Type Description

Both tools are **Predictive analytic** tools because:

- They create models that make predictions (classify images, sounds, etc.)
- They don't just describe existing data but predict outcomes for new inputs

3. Learning Type Description

Both tools use **Supervised learning** because:

- Users provide labeled examples (e.g., "this is a cat picture")
- The models learn from these labeled examples to make predictions
- Neither tool offers unsupervised or reinforcement learning features

Q.3: Data Visualization and Misinformation

Summary of Articles:

1. **Kakande's Article** explains how to spot misleading visualizations by checking:
   - Data sources and context
   - Appropriate chart types
   - Axis manipulation
   - Cherry-picked timeframes
2. **Foley's Article** shows how COVID-19 visualizations misled by:
   - Using inappropriate scales
   - Omitting key context
   - Creating false comparisons

Current Event Example:

Case: 2022 Twitter Misleading Climate Change Graph

Source: The Guardian

(https://www.theguardian.com/environment/2022/oct/12/twitter-accounts-misleading-graph-climate-crisis)

What Happened:

- A viral graph showed global temperatures with an exaggerated y-axis scale

- It made recent temperature increases appear insignificant compared to historical data
- The visualization omitted key context about rate of change

Why It Failed:

1. **Axis Manipulation:** The y-axis scale was expanded to minimize visual impact of recent warming
2. **Cherry-Picked Data:** Only showed specific time periods that supported the misleading narrative
3. **Missing Context:** Didn't explain why short-term changes matter more than long-term averages

Q.4: Classification Model Implementation & Performance Analysis

Dataset: Pima Indians Diabetes Database

Approach:

1. Data Preparation Pipeline

- **Class Imbalance Handling:** Used SMOTE to balance classes (0: Non-Diabetic, 1: Diabetic).
- **Preprocessing:** Standardized features using `StandardScaler` for SVM (scale-sensitive).
- **Train/Validation/Test Split:**
  - **70% Training** (Model learning)
  - **20% Validation** (Hyperparameter tuning)
  - **10% Test** (Final evaluation)

2. **SMOTE (Synthetic Minority Over-sampling Technique)**

- **Full Form:** Synthetic Minority Over-sampling Technique
- **Purpose:** Addresses class imbalance by generating synthetic samples for the minority class.
- **How it Works:**
  - Creates artificial data points for the minority class (e.g., diabetic patients) using nearest neighbors.
  - Example: If "Diabetic" cases are only 30% of the dataset, SMOTE increases this to 50% by adding synthetic samples.

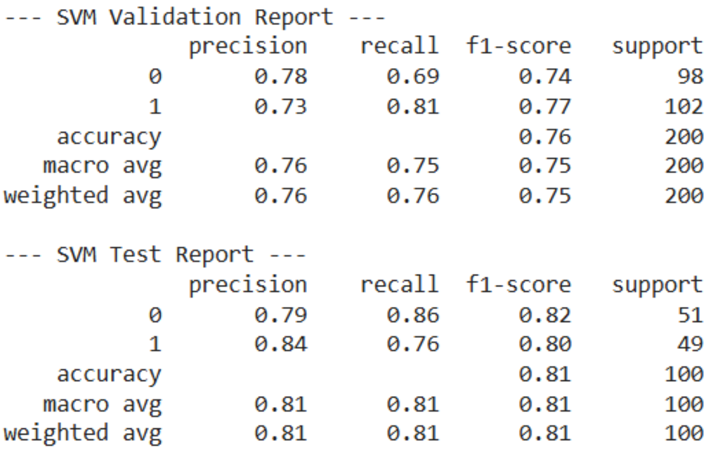
- **Why Used:** Prevents models from being biased toward the majority class (e.g., non-diabetic cases).

3. StandardScaler

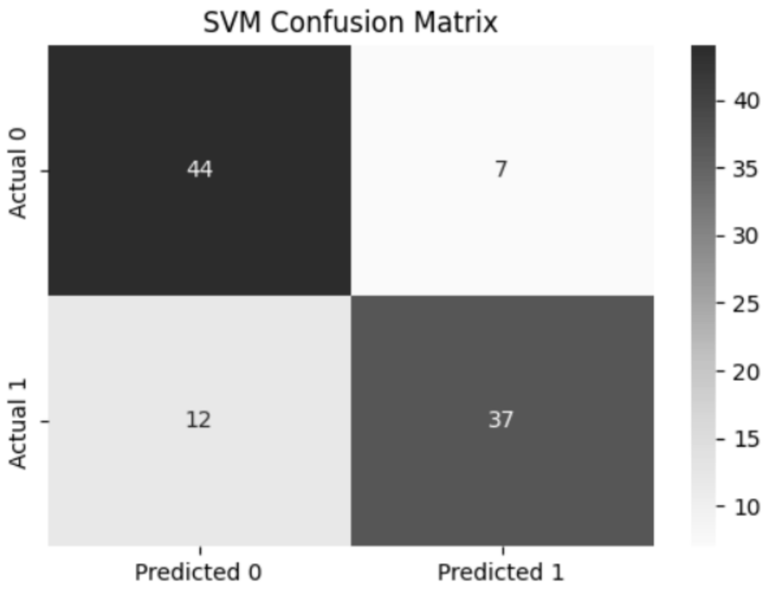
- **Purpose:** Standardizes features to have a mean of 0 and standard deviation of 1
- **Formula:**

$$X_{\text{scaled}} = \frac{X - \text{mean}(X)}{\text{std}(X)}$$

- **Why Used:**
  - Essential for algorithms like SVM that are sensitive to feature scales.
  - Ensures all features contribute equally to model performance.



```
--- SVM Validation Report ---
              precision    recall  f1-score   support
           0       0.78      0.69      0.74        98
           1       0.73      0.81      0.77       102
    accuracy                           0.76       200
   macro avg       0.76      0.75      0.75       200
weighted avg       0.76      0.76      0.75       200

--- SVM Test Report ---
              precision    recall  f1-score   support
           0       0.79      0.86      0.82        51
           1       0.84      0.76      0.80        49
    accuracy                           0.81       100
   macro avg       0.81      0.81      0.81       100
weighted avg       0.81      0.81      0.81       100
```

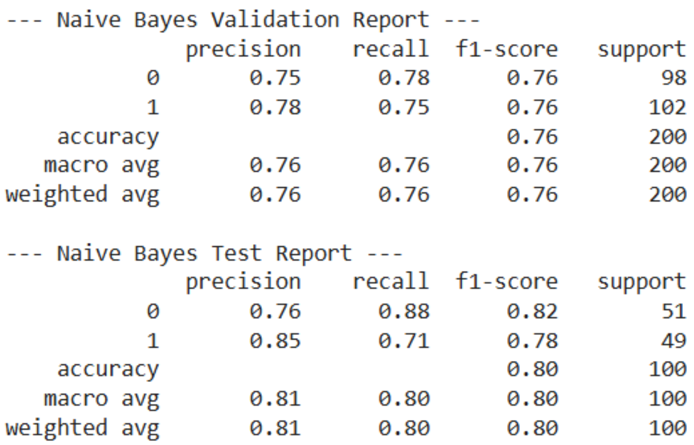


- **Interpretation:**
  - **Precision (0.78 for Class 0):** 78% of predicted Non-Diabetic cases were correct.
  - **Recall (0.81 for Class 1):** Captured 81% of actual Diabetic cases.
  - **Accuracy:** 75% on validation data.
- **Improvement:** Higher accuracy (81%) on unseen test data, indicating good generalization.

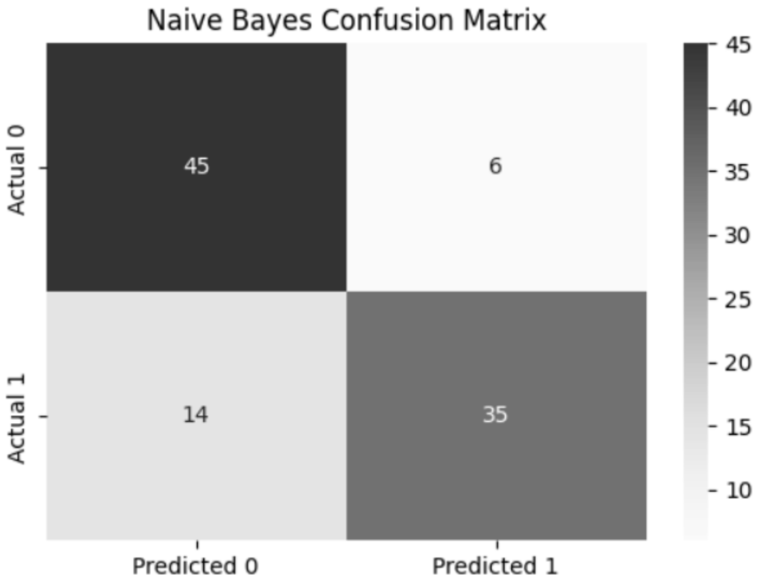
Key Insight:

- **7 False Positives (FP):** Non-Diabetic cases misclassified as Diabetic.

- **12 False Negatives (FN):** Diabetic cases missed (critical error).



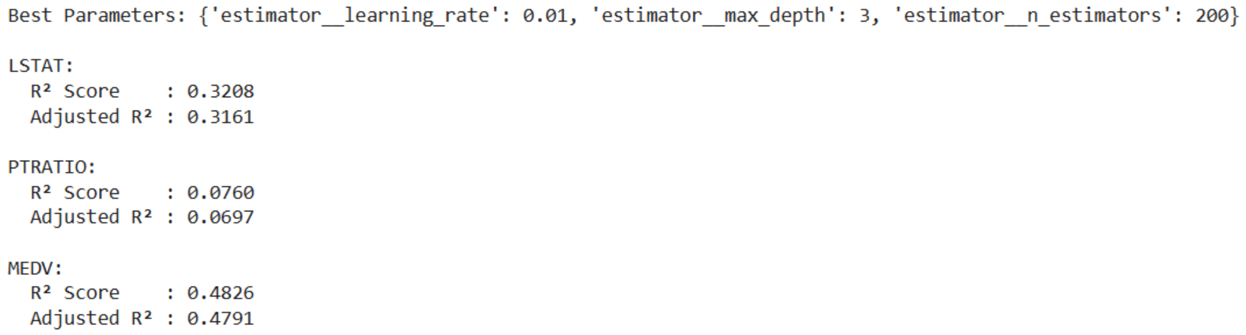
```
--- Naive Bayes Validation Report ---
              precision    recall  f1-score   support
           0       0.75      0.78      0.76        98
           1       0.78      0.75      0.76       102
    accuracy                           0.76       200
   macro avg       0.76      0.76      0.76       200
weighted avg       0.76      0.76      0.76       200

--- Naive Bayes Test Report ---
              precision    recall  f1-score   support
           0       0.76      0.88      0.82        51
           1       0.85      0.71      0.78        49
    accuracy                           0.80       100
   macro avg       0.81      0.80      0.80       100
weighted avg       0.81      0.80      0.80       100
```



- **Balanced Performance:** Similar metrics for both classes (no bias).
- **Recall Trade-off:** Higher recall for Class 0 (88%) but lower for Class 1 (71%).
- **Comparison to SVM:**
  - Fewer FP (6 vs. 7) but greater FN (14 vs 12).
  - Slightly better precision for Class 1 (85%).

Q.5: Regression Model Implementation - Output Analysis

1. Model Performance Metrics



```
Best Parameters: {'estimator__learning_rate': 0.01, 'estimator__max_depth': 3, 'estimator__n_estimators': 200}

LSTAT:
  R² Score    : 0.3208
  Adjusted R² : 0.3161

PTRATIO:
  R² Score    : 0.0760
  Adjusted R² : 0.0697

MEDV:
  R² Score    : 0.4826
  Adjusted R² : 0.4791
```

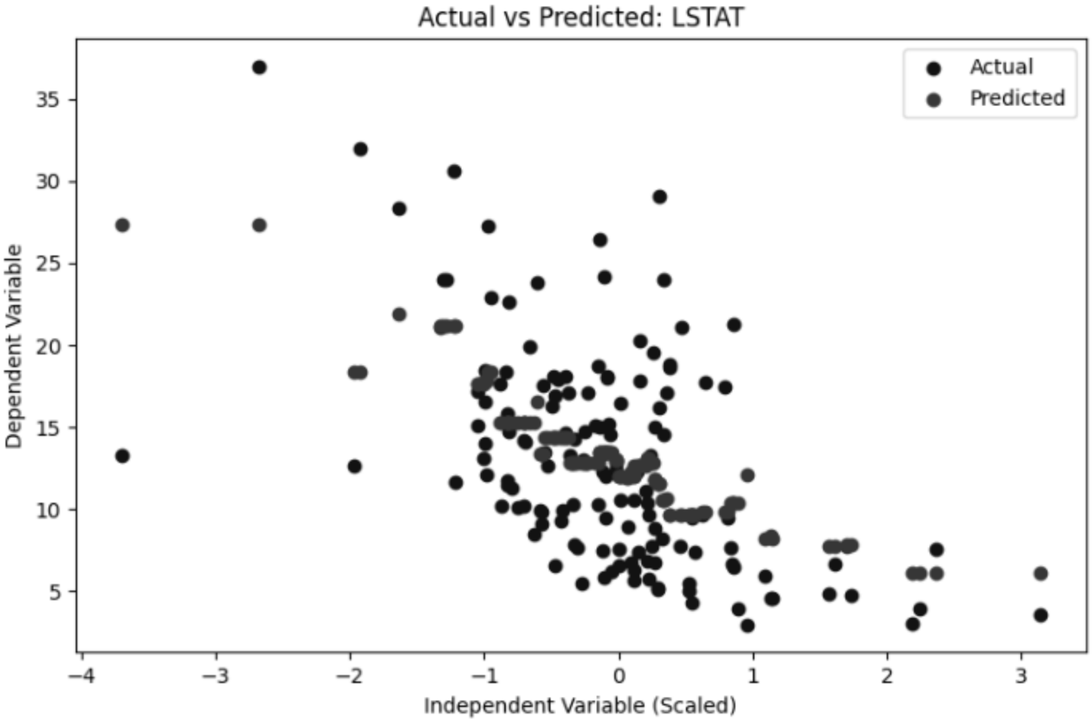
Interpretation:

- **LSTAT (Lower Status % Population):**
  - **R^2 = 0.32**: The model explains **32.08%** of variance in LSTAT using the independent variable.

  - **Adjusted R² (0.316)**: Confirms minimal overfitting.
  - **Implication**: Moderate predictive power for socioeconomic status.
- **PTRATIO (Pupil-Teacher Ratio):**
  - **R² = 0.076**: Explains only **7.6%** of variance.
  - **Adjusted R² (0.0697)**: Almost no predictive value.
  - **Issue**: Independent variable (likely housing feature) poorly correlates with school metrics.
- **MEDV (Median Home Value):**
  - **R² = 0.48**: Captures **48.26%** of home value variance.
  - **Adjusted R² (0.479)**: Strongest relationship among outputs.
  - **Use Case**: Useful for rough price estimation but lacks precision.

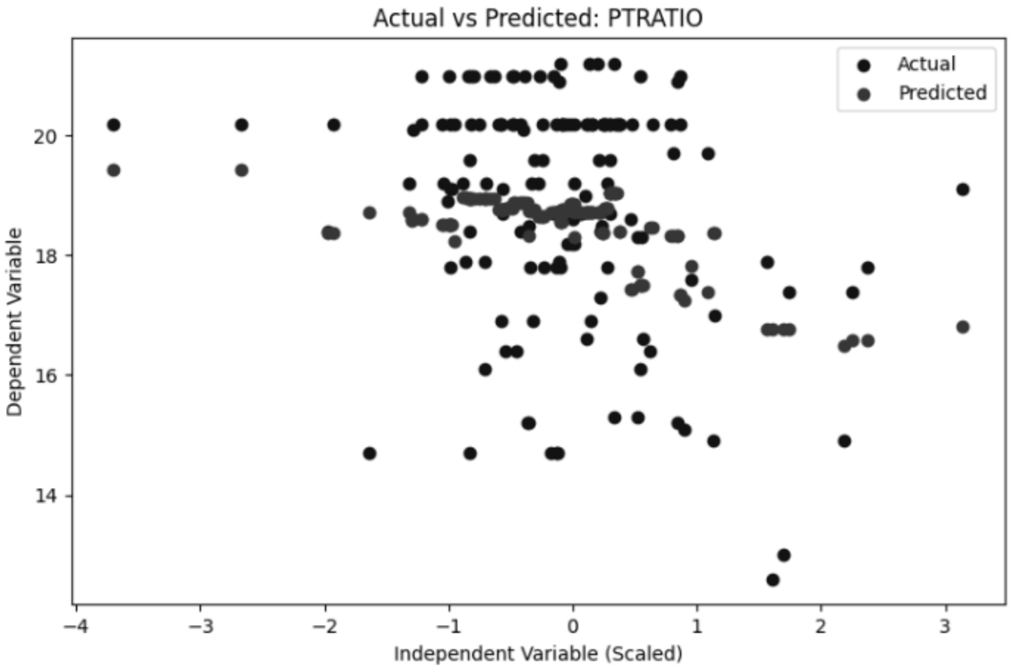
2. Visualization Analysis

A. LSTAT Plot



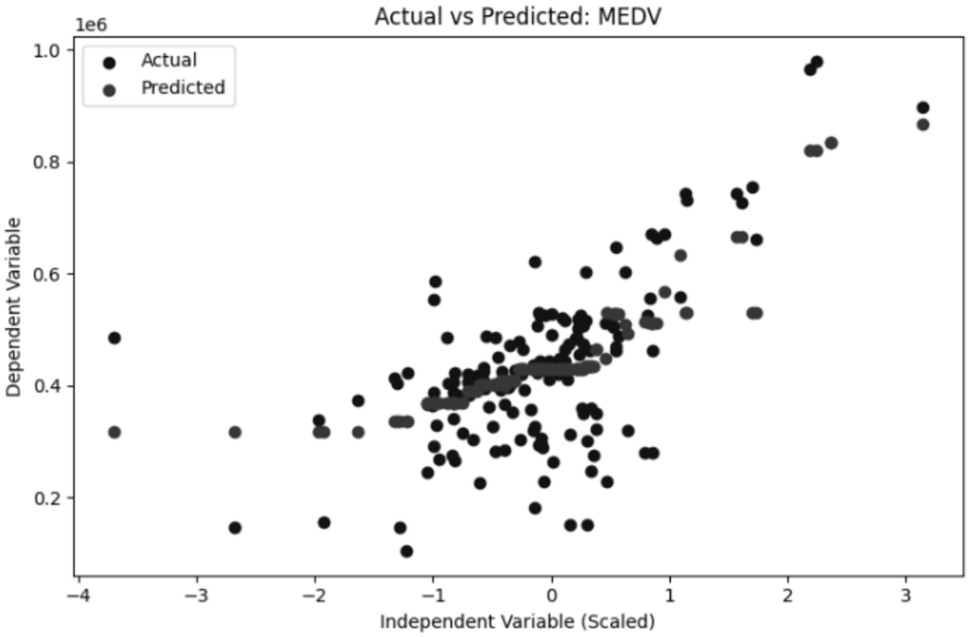
- **Pattern**: Points loosely follow a trend but with significant scatter.
- **Conclusion**: Partial correlation exists, but other factors likely influence LSTAT.

B. PTRATIO Plot



- **Pattern**: Random scatter with no clear trend.
- **Conclusion**: Independent variable fails to predict PTRATIO effectively.

C. MEDV Plot



- **Pattern**: Some clustering along a diagonal trend.
- **Conclusion**: Best-performing output, but predictions deviate for higher values.

Name : Nishant Sachin Khetal **Roll No : 24** **Div : D15C**

Q.6: Wine Quality Dataset Analysis

Key Features & Their Importance

The wine quality dataset contains measurements that affect how wine tastes and its overall quality. Here’s why each feature matters:

1. **Fixed Acidity**
    - **Role**: Gives wine its tartness.
    - **Impact**: Too much makes wine sour; too little makes it flat.
2. **Volatile Acidity**
    - **Role**: Measures vinegar-like acids.
    - **Impact**: High levels ruin taste (bad quality).
3. **Citric Acid**
    - **Role**: Adds a fresh, citrusy flavor.
    - **Impact**: Balances sourness; improves taste.
4. **Residual Sugar**
    - **Role**: Leftover sugar after fermentation.
    - **Impact**: Sweetens wine; too much can make it syrupy.
5. **Chlorides**
    - **Role**: Salt content.
    - **Impact**: High amounts make wine taste salty (poor quality).
6. **Free Sulfur Dioxide**
    - **Role**: Preserves wine and prevents spoilage.
    - **Impact**: Too much gives a chemical taste.
7. **Total Sulfur Dioxide**
    - **Role**: Total preservatives in wine.
    - **Impact**: Must be balanced for freshness without off-flavors.
8. **Density**
    - **Role**: Thickness of wine (linked to sugar/alcohol).
    - **Impact**: Affects mouthfeel—light vs. heavy wines.
9. **pH**
    - **Role**: Measures acidity level.
    - **Impact**: Affects taste and shelf life (ideal pH = balanced).
10. **Sulphates**

- **Role**: Enhances flavor and preservation.
- **Impact**: Boosts aroma but can be harsh if excessive.

11. **Alcohol**

- **Role**: Determines strength and body.

- **Impact**: Higher alcohol = richer taste, but too much burns.

Handling Missing Data

If the dataset has missing values, we use **imputation** (filling gaps smartly). Common methods:

1. **Mean/Median Imputation**
   - ○ **How**: Replace missing values with the average.
   - ○ **Pros**: Simple, fast.
   - ○ **Cons**: Can make data less realistic if values are skewed.
2. **K-Nearest Neighbors (KNN) Imputation**
   - ○ **How**: Fill gaps using similar wines in the dataset.
   - ○ **Pros**: More accurate than mean.
   - ○ **Cons**: Slower for large datasets.
3. **Regression Imputation**
   - ○ **How**: Predict missing values using other features.
   - ○ **Pros**: Uses relationships between features.
   - ○ **Cons**: Can overfit (works too well on training data but fails later).

Best Choice?

- If only a few values are missing → **Mean/Median**.
- If data has clear patterns → **KNN or Regression**.

Why This Matters

- **For Winemakers**: Helps adjust ingredients to improve quality.
- **For Buyers**: Predicts which wines taste best based on chemistry.
- **For Data Scientists**: Clean data = better AI models.

  Overall, understanding the contribution of each feature helps build more accurate predictive models for wine quality, while careful handling of missing data ensures that the integrity of the model is maintained.