Assignment

- Submission Format (.zip to CyberCampus)
  - Python .py file (clean and commented)
  - Matplotlib figures (screen)
  - Inline or markdown-style explanations for each section (word)

Assignment (Clean It, Fix It, Scale It)

- Assignment: Data Preprocessing & Feature Scaling (Hands-On)

- Learning Objectives
  - By completing this assignment, you will:
  - Handle missing values using basic and advanced methods
  - Detect and treat outliers using statistical techniques
  - Apply different feature scaling methods
  - Understand when and why to use each scaling technique
  - Visualize data before and after transformation

Assignment (Clean It, Fix It, Scale It)

- Data Setup
  - You'll generate synthetic data using NumPy — no file loading needed.

```
import numpy as np

np.random.seed(42)
n = 150

# Synthetic features
age = np.random.normal(40, 10, n)
income = np.random.normal(60000, 15000, n)
purchases = np.random.exponential(300, n)
clicks = np.random.poisson(5, n)

# Inject missing values
income[5] = np.nan
purchases[10] = np.nan

# Inject outliers
income[7] = 300000
purchases[3] = 5000
```

Assignment (Clean It, Fix It, Scale It)

- Missing Value Handling
  - Detect which features contain missing values.
  - Fill missing values using:
    - Mean for income
    - Median for purchases
  - Print both original and filled values for confirmation.

- Outlier Detection & Handling
  - Use the IQR method to detect outliers in:
    - income
    - purchases
  - Print outlier values and their indices.
  - Replace them with the nearest non-outlier value or clip them.

Assignment (Clean It, Fix It, Scale It)

- Feature Scaling
  - Apply the following scaling methods:
  - a. Min-Max Scaling for age
  - b. Z-score Standardization for income
  - c. Log Transformation for purchases
  - d. Robust Scaling for income
  - e. Vector Normalization for [age, income, clicks] as a feature vector

- For each method:
  - Print the transformed values (first 5 entries)
  - Explain why that method is or isn’t appropriate for the given feature

- Visualization
  - Use matplotlib to plot:
  - Histogram of purchases before and after log transform
  - Box plot of income before and after robust scaling

Assignment: Categorical Encoding (Hands-On)

- “From Strings to Vectors: Encoding Categorical Data”

- Learning Objectives
  - Identify nominal vs ordinal categorical variables
  - Apply label encoding, manual ordinal encoding, and manual one-hot encoding using NumPy
  - Understand when each method is appropriate

- Synthetic Dataset Setup
  - You’ll simulate a small dataset:

```
import numpy as np

# Categorical variables
colors = np.array(["Red", "Green", "Blue", "Green", "Red", "Blue"])
sizes = np.array(["Small", "Medium", "Large", "Small", "Large", "Medium"])
brands = np.array(["Nike", "Adidas", "Puma", "Nike", "Puma", "Adidas"])
```

Assignment: Categorical Encoding (Hands-On)

- 1 Label Encoding
  - Write code to convert brands into numeric labels:
  - "Nike" → 0
  - “Adidas" → 1
  - "Puma" → 2
  - ➡ Use np.unique() to get sorted unique values, then loop to encode.

- 2 Ordinal Encoding
  - Encode sizes based on order:
  - "Small" → 1
  - "Medium" → 2
  - "Large" → 3
  - ➡ Use a manual mapping with a dictionary.

- 3 One-Hot Encoding
  - One-hot encode colors using np.unique() and a loop.
  - Output should be a 6x3 array where each row represents a color.

Assignment: Categorical Encoding (Hands-On)

- 4 Print a final feature matrix combining:
  - One-hot encoded colors
  - Ordinal encoded sizes (as one column)
  - Label encoded brands (as one column)
  - ➜ Final shape should be 6x(3 + 1 + 1) = 6x5

- Short Reflection Questions (in comments)
  - Why is one-hot encoding better for colors than label encoding?
  - Why is ordinal encoding okay for sizes?

Assignment: Feature Selection & Preprocessing

- Apply a variety of data preprocessing and feature selection techniques to a real-world dataset and analyze which features are most useful for classification.

- Dataset
  - Use the built-in Breast Cancer Wisconsin dataset from sklearn.datasets.

- Part 1: Data Preparation
  - Load the dataset and display:
  - Number of samples
  - Feature names
  - Target class distribution
  - Normalize all feature values using MinMaxScaler.

Assignment: Feature Selection & Preprocessing

- Part 2: Feature Selection Techniques
- A. Chi-Square Test
  - Apply SelectKBest with chi2 to score all features.
  - Plot a bar chart of Chi-Square scores.
  - Identify the top 5 features.
- B. Lasso Regression
  - Use Lasso(alpha=0.01) to fit the scaled data.
  - Print out the coefficients.
  - Identify which features are selected (non-zero).
- C. Tree-Based Model
  - Train ExtraTreesClassifier on the same data.
  - Plot feature importances.
  - Identify the top 5 most important features.

Assignment: Feature Selection & Preprocessing

- Part 3: Comparison & Reflection

- Answer the following:
  - Which features were selected consistently across methods?
  - Did any method eliminate features that another considered important?
  - Which method do you think is most trustworthy for this task, and why?