Machine Learning: Student Independent Tasks - 1 (Lectures 1-3)

Course: CSCI3234 — School of IT and Engineering

Overview. This document contains independent practical tasks for the *Machine Learning* course, aligned with Lectures 1–3.
Each task is designed for a Jupyter/Colab workflow with Python (pandas, numpy, matplotlib/seaborn).

Submission & Deliverables

- A well-commented **Jupyter notebook** showing data loading, code, and outputs exported **in PDF**.
- **Markdown** cells under each result explaining the interpretation (what was done and why).
- All plots must include titles, axis labels, and short captions.
- Different datasets may be selected for each task, depending on which one is most appropriate for the given objective.

1 Level A — Easy

1.1 A1. Data Types & Attributes (0.5)

Goal Correctly identify and classify attributes.

Instructions

1. Choose a public dataset.
2. For each attribute, classify it as **nominal**, **ordinal**, **discrete**, or **continuous**.
3. Justify each classification in 1–2 sentences.

1.2 A2. Descriptive Statistics (0.5)

Goal Compute central tendency and dispersion; visualize outliers.

Instructions

1. Select all numeric columns.
2. Compute: **mean**, **median**, **mode**, **range**, **variance**, **standard deviation**, **IQR**.
3. Create a **boxplot**; mark suspected outliers and list their values.

1.3 A3. Handling Missing Data (0.75)

Goal Apply multiple imputation strategies and compare effects.

Instructions

1. Introduce missing values artificially (e.g., 5% random NaN) or use a dataset with missing values.
2. Apply three imputation methods:
   - Constant value,
   - Mean/Median/Mode,
   - Predictive imputation (e.g., regression or KNN).
3. Compare mean and standard deviation before and after imputation.
4. Briefly discuss advantages and disadvantages of each method.

2 Level B — Medium

2.1 B1. Normalization & Standardization (0.75)

Goal Compare scaling methods.

Instructions

1. Select at least two numeric features with different scales.
2. Apply:
   - Min–max normalization to [0,1],
   - Z-score standardization.
3. Plot distributions before and after scaling.
4. Explain when each method is preferable.

2.2 B2. Feature Creation & Discretization (0.75)

Goal Engineer useful features. Each task below (B2 task) independent and applies for different features.

Instructions

1. Create one new feature using domain logic.
2. Apply discretization using (and after compare bin counts and boundaries):
   - Equal-width binning,
   - Equal-frequency binning.
3. Convert categorical variables using one-hot encoding.
4. Discuss how these transformations may impact ML models.

3 Additional Analytical Tasks

3.1 D1. Measures of Similarity and Dissimilarity (0.5)

Goal Understand distance and similarity metrics. Try to write calculation using Numpy (in low code).

Instructions

1. Calculate **Euclidean distance** between two numeric vectors:
2. Compute **Jaccard similarity** between two sets:
3. Calculate between two numeric vectors:
   - Cosine similarity,
   - Pearson correlation coefficient.
4. Write a short explanation of each metric.
5. Compare numerical results and explain why they differ.

3.2 D2. Dimensionality Reduction (PCA) (0.75)

Goal Apply Principal Component Analysis manually using NumPy.

Instructions

1. Standardize features.
2. Compute the covariance matrix.
3. Find eigenvalues and eigenvectors.
4. Select the two largest eigenvalues.
5. Project data onto the corresponding eigenvectors.
6. Plot the 2D result using matplotlib.
7. Briefly explain:
   - What each step means,
   - What eigenvalues represent,
   - What the PCA visualization shows.
8. Compare the original data structure with the reduced 2D projection.

4 Level C — Difficult

4.1 C1. Advanced Sampling & Data Quality Analysis (1.5)

Goal Evaluate representativeness under different sampling schemes.

Instructions

1. Select a dataset $\geq$ 50MB.
2. Create:
   (a) Simple Random Sample (10%),
   (b) Stratified sample,
   (c) Cluster sample (justify).
3. Compute descriptive statistics (mean, median, variance, IQR).
4. Visualize distributions vs. full dataset.

5. Write a short analytical report discussing:
    - Which sampling method preserved distributions best,
    - Impact on outlier detection,
    - Recommendations for real-world projects.

Total: 6.0 points