Task 2: Core ML Foundations (4 Parts)

Objective

Understand and implement the core steps of a machine learning pipeline. You will independently select a meaningful dataset, prepare it, implement one supervised and one unsupervised algorithm, and evaluate results. Reference resources will be provided for learning.

Allowed tools & constraints

- **scikit-learn: allowed for dataset loading, preprocessing helpers, and baseline comparison only**
- **Core supervised and unsupervised algorithms must be implemented by you**
- **Notebook must run top-to-bottom**
- **If subsampling is used, it must be justified**

Dataset requirement

- **You must search and select your own dataset (Kaggle, UCI, data.gov, etc.)**
- **Dataset must be real-world and meaningful, not toy**
- **Dataset size: manageable on a laptop**
- **Clearly cite the dataset source in the README**
- **All work for this task must be pushed to a GitHub repository, and daily commits / updates will be monitored to track consistent progress**

Part 1: Problem framing & EDA

Deliverables: short text + plots + observations

Required Text

- **Dataset description and source**
- **Problem type (regression / classification / clustering)**

- Target variable and key input features
- One-line success criterion

EDA

- Dataset shape and dtypes
- Missing-value summary
- Basic statistics (mean / median / std)
- Three plots:
  - Target distribution
  - Feature relationship (correlation heatmap or 2-feature scatter)
  - Outlier check (boxplot or histogram)

Observations to mention

- What the target distribution indicates (skew, imbalance, spread)
- Any strong feature relationships or lack thereof
- Presence of outliers or scale differences and their possible impact

Part 2: Preprocessing & data split

Deliverables: functions, processed arrays, observations

Required

- Implement preprocessing steps:
  - Missing-value handling
  - Categorical encoding
  - Feature scaling
- Implement your own `train_test_split(X, y, seed)`

Observations to mention

- Why chosen imputation and encoding methods suit this dataset
- Whether scaling changes feature dominance
- One concrete data-leakage risk in this dataset and how it was avoided

---

Part 3: Supervised learning (from scratch + baseline)

Deliverables: implementation, metrics, observations

Required

- Implement one supervised algorithm from scratch:

  Example:

  - Linear Regression or
  - k-Nearest Neighbors or
  - Logistic Regression
  - Or any of your choice
- Implement `fit()` and `predict()`
- Evaluate using self-implemented metrics
- Train an equivalent scikit-learn model as baseline and compare results

Observations to mention

- Effect of algorithm assumptions on results
- Sensitivity to parameters (e.g., k, learning rate)
- Where and why errors occur (at least one concrete example)

Part 4: Unsupervised learning & reflection

Deliverables: implementation, plots, observations

Required

- **Implement one unsupervised algorithm from scratch:**

  Example:

  - **K-Means or**
  - **PCA**
  - **Or any of your choice**
- **Produce required visualizations**

Observations to mention

- **Structure revealed by clustering or dimensionality reduction**
- **Alignment (or lack thereof) with known labels, if available**
- **How unsupervised insights could help supervised modeling**

Reflection

- **Two failure cases of the supervised model**
- **Three questions you expect in the 1:1 discussion (with short answers)**

Final submission

- `task2_<yourname>.ipynb` **- runnable notebook**
- `README.md` **- dataset source, problem statement, execution steps, short summary**
- **Optional** `utils.py`
- **3-slide PDF or 3-minute screencast**

- **Task is sequential and required for the upcoming tasks.**
- **Project completion entirely with AI is not allowed - violations mean redo.**
- **Project Review & doubt-clearing meets will be conducted.**