Assignment 1: From Dirty Data to Predictive Models

Due Date: Monday, October 6, 2025 at 12:00 AM (midnight, Eastern Time)
Policy: https://columbia-coms4995.github.io/aml-fall2025/#policies

📃 Submission

- A runnable Jupyter/Colab Notebook (.ipynb).
- A PDF report (5+ pages w/ clear structure).

🎯 Objective

The goal of this assignment is to guide you through the process of building an end-to-end flow to turn data to predictive models, using what you have learned so far. You will:

- Clean and transform a real dataset that contains missing values, noisy data, and categorical variables.
- Engineer meaningful features to improve model performance.
- Train and compare 2 different modeling approaches:
  - Generative: Naive Bayes
  - Discriminative: Linear Regression
- Evaluate your models using appropriate metrics and visualizations.
- Reflect on your workflow and disclose any AI tools used.

By the end of this assignment, you shall have hands-on experience with data preprocessing, feature engineering, model training, basic evaluation, and interpretation.

📂 Datasets

Choose one of the following datasets. Both datasets contain real-world challenges such as missing values and categorical variables.

Titanic Survival Dataset

Task: Predict whether a passenger survived or not.

Dataset link: Kaggle Titanic Dataset

Notes: Requires a Kaggle account.

Heart Disease Prediction Dataset

Task: Predict whether a patient has heart disease based on medical attributes.

Dataset link: UCI Heart Disease Dataset

🛠️ Steps

To resolve this supervised learning task, your assignment will consist of the following steps:

1. Data Cleaning
    a. Handle missing values (imputation, dropping, or flagging).
    b. Address noisy or inconsistent values.
    c. Justify your choices (why impute vs. drop?).

2. Feature Engineering
    a. Apply appropriate transformations (normalization, standardization, log-scaling).
    b. Encode categorical variables (one-hot encoding or other methods).
    c. Optionally, construct new features (e.g., ratios, group statistics).

3. Model Training
    a. Train a Naive Bayes classifier (BernoulliNB or GaussianNB depending on your dataset), and make sure to apply Laplace smoothing (add-α).
        i. You should experiment with at least 2 different values of alpha (e.g., 1.0 vs. 0.01) and compare the results.
    b. Train a Linear Regression model.
        i. Although Linear Regression is typically used for regression tasks, in this assignment we will also apply it to a binary classification problem with a probability threshold as 0.5.
        ii. Reminder: To prevent overfitting and to explore the effect of regularization, you are encouraged to also try:
            1. Ridge Regression (L2 regularization)
            2. LASSO Regression (L1 regularization)
    c. Ensure fair comparison (same train/test split).

4. Model Evaluation
    a. [Required] Accuracy + Confusion Matrix. [Encouraged] Precision, Recall, F1-score. [Bonus] ROC + AUC
    b. Include at least one visualization (confusion matrix heatmap, ROC curve).
    c. Explain what happened when you used smoothing vs. without smoothing.

5. Report Writing (5-7 pages)
    a. Introduction: problem definition + dataset description.
    b. Data Cleaning: your steps, reasoning, and before/after examples.
    c. Feature Engineering: transformations and encodings applied.
    d. Model Comparison: training setup, evaluation results.
    e. Discussion: interpretation of results and limitations.
    f. AI Tool Usage Disclosure:
        i. List AI tools used (e.g., ChatGPT, Gemini, Claude).
        ii. Describe what these tools contributed (e.g., generating starter code, visualization, debugging).
        iii. Clarify which parts were your own contribution.

🔑 Pointers & Hints

To help you get started as the 1st assignment:

- Libraries: `scikit-learn`, `pandas`, `numpy`, `matplotlib/seaborn`.
- Functions:
  - `train_test_split` for splitting data.
  - `classification_report`, `confusion_matrix` for evaluation.
  - `cross_val_score` for validation (optional).
- Models:
  - `sklearn.naive_bayes.BernoulliNB`
  - `sklearn.naive_bayes.GaussianNB`
  - `sklearn.linear_model.LinearRegression`
- Visualization:
  - Use `seaborn.heatmap` for confusion matrix.
  - [Optional] Use `sklearn.metrics.roc_curve` and `auc` for ROC plots.

📊 Grading Rubric

Category	Percentage	Description
Data Cleaning & Transformation	20	Clear handling of missing, noisy, and inconsistent data with justification.
Feature Engineering	20	Correct application of transformations and encodings; creativity in constructing new features.
Model Implementation	20	Properly trained Naive Bayes and Linear Regression; fair comparison setup.
Evaluation & Visualization	15	Correct use of metrics; meaningful visualizations included.
Discussion & Interpretation	15	Thoughtful analysis of results, strengths/limitations, and clear explanations.
AI Tool Usage Disclosure	10	Transparent description of AI tool usage and personal contributions.

💰Bonus (up to +10%)

Bonus points are only awarded for well-explained analysis, NOT just running extra code.

- Discuss how the choice of regularization (and the parameter `alpha`) affects model performance and feature importance.
- Observe which coefficients are shrunk to zero and discuss what this means for feature importance in LASSO.
- Reflect on AI vs. human-generated code and insights.