Indiana University
DSCI-D 321: Data Representation (FA25)
Muazzam Siddiqui

Assignment 6
Nov 21, 2025

Data Preparation for Machine Learning

Important Information:

- Assigned: Nov 21, 2025
- Deadline: Dec 11, 2025 at 11:59 PM EST
- This is a 100 pt assignment, worth 7.5% of your grade.
- Late penalty (10%) deadline: There is no late submission for this assignment.

Submission Instructions

To turn in your assignment, you will need to upload to Canvas the following files:

1. `lastname_firstname_a6.ipynb` with Python code and plots and findings

Generative AI

AI is allowed for this assignment, but think of it as a learning assistant rather than a solution provider. Use it to explore concepts, get hints, and debug your code, but let the actual problem solving come from you.

While it's possible to generate a full solution quickly, that won't help you build the skills you will need. The real value of these assignments is in preparing you for the exams, where you will tackle similar problems without AI support. By engaging deeply with the work now, you'll be setting yourself up for success later.

Objectives:

In this assignment, you will evaluate the effectiveness of different data preprocessing techniques intrinsically and extrinsically (on downstream tasks)

- Compare imputation methods (`drop` attribute, mean, median, mode)
- Compare categorical encoders (label, one-hot, binary, target with OOF) and scalers (min-max, z-score, log, Box-Cox, sigmoid, max-abs, unit).
- Use proper experiment protocol (no leakage, pipelines, nested CV).
- Analyze results with summary statistics and simple statistical tests.

Task 1. Implement preprocessing methods (30 marks)

Implement OR use built-in functions for:

1. **Numeric imputation**: `drop` attribute (column removal), mean (numeric only), median (numeric/ordinal), mode. (10 marks)
2. **Categorical encoders**: Label, one-hot, binary, target encoding with out-of-fold (OOF) safe scheme. Provide an OOF implementation or use category encoders with OOF. (10 marks)
3. **Scaling/normalization**: min-max, z-score, log (handle zeros), box-cox, sigmoid, max-abs, unit-length. (10 marks)

Task 2. Experiment Pipeline & Evaluation (50 marks)

For this assignment, all necessary data is provided in a zipped folder containing the `generated_datasets` directory. You must ensure this directory is accessible to your notebook code. This folder contains 78 CSV files and a JSON file named `metadata.json`. The 78 CSV files are composed of 6 original CLEAN base datasets and 72 derived datasets where missing values (MCAR, MAR, MNAR at 10% and 30% rates across two random seeds) have been intentionally introduced only into the original 80% training data split. The `metadata.json file` is required by the `main_experiment_loop` in the notebook to identify the experimental parameters (dataset name, missingness type, rate, and seed) corresponding to each CSV file, allowing the experiment to iterate through all scenarios correctly.

1. Build `scikit-learn` pipelines that fit-transforms only on training folds (use `ColumnTransformer`). (6 marks)

2. **Missing value imputation evaluation**: For each training dataset with missing values,

   - Evaluate intrinsic imputation quality (RMSE for numeric; accuracy for categorical) by masking a separate holdout of observed values. (8 marks)
   - Evaluate extrinsic model performance (Classification: Accuracy + ROC-AUC; Regression: RMSE + R^2) using two model families: Logistic/Linear regression, and Neural Networks(NN), against the testing dataset. (12 marks)

3. **Category encoder evaluation**: For each original training dataset,

   - Evaluate extrinsic model performance (Classification: Accuracy + ROC-AUC; Regression: RMSE + R^2) using two model families: Logistic/Linear regression, and Neural Networks(NN), against the testing dataset. (12 marks)

4. **Scaling/normalization evaluation**: For each original training dataset,

   - Evaluate extrinsic model performance (Classification: Accuracy + ROC-AUC; Regression: RMSE + R^2) using two model families: Logistic/Linear regression, and Neural Networks(NN), against the testing dataset. (12 marks)

Task 3. Analysis & Visualization (20 marks)

1. Compare imputation methods across missingness & datasets & model. (5 marks)
2. Compare category encoding methods across datasets and models. (5 marks)
3. Compare scaling/normalization methods across datasets and models. (5 marks)
4. Summarize key findings: which techniques are robust, interactions with model type, surprising results. (5 marks)