Responsible Machine Learning
Assignment 1
10 points

November 5, 2026

In Assignment 1, you will train an explainable machine learning (ML) model and make predictions on the test data. You may use AI tools to build your model and any associated code, documentation, or other artifacts. Ensure Python 3 and Git are installed.

1 Download Data.

Download Home Mortgage Disclosure Act (HMDA) data as zip files from this folder in the class repository. The folder includes two data files:

- `hmda_train_preprocessed.zip` – Zipped CSV HMDA *labeled* training data.
- `hmda_test_preprocessed.zip` – Zipped CSV HMDA *unlabeled* test data.

Later you will score the unlabeled test data with your models and submit these scores as part of your assignment deliverable.

2 Load and Explore Data.

Load the data into modeling software. Training data contains 160338 rows and 23 columns. Test data contains 19831 rows and 22 columns. The features to use for Assignment 1 are as follows:

- `high_priced`: Binary target, whether (1) or not (0) the annual percentage rate (APR) charged for a mortgage is 150 basis points (1.5%) or more above a survey-based estimate of similar mortgages. (High-priced mortgages are legal, but somewhat punitive to borrowers. High-priced mortgages often fall on the shoulders of minority homeowners and are one of many issues that perpetuate a massive disparity in overall wealth between different demographic groups in the US.)
- `conforming`: Binary numeric input, whether the mortgage conforms to normal standards (1), or whether the loan is different (0), e.g., jumbo, HELOC, reverse mortgage, etc.
- `debt_to_income_ratio_std`: Numeric input, standardized debt-to-income ratio for mortgage applicants.
- `debt_to_income_ratio_missing`: Binary numeric input, missing marker (1) for `debt_to_income_ratio_std`.
- `income_std`: Numeric input, standardized income for mortgage applicants.
- `loan_amount_std`: Numeric input, standardized amount of the mortgage for applicants.
- `intro_rate_period_std`: Numeric input, standardized introductory rate period for mortgage applicants.
- `loan_to_value_ratio_std`: Numeric input, ratio of the mortgage size to the value of the property for mortgage applicants.
- `no_intro_rate_period_std`: Binary numeric input, whether a mortgage does not include an introductory rate period.

- `property_value_std`: Numeric input, value of the mortgaged property.
- `term_360`: Binary numeric input, whether the mortgage is a standard 360-month mortgage (1) or a different type of mortgage (0).

This data contains no major quality issues, so no preprocessing is required. Familiarize yourself with the data using basic exploration techniques. You may optionally try to improve your model with feature engineering or other preprocessing approaches.

Training data should be used to create training and validation partitions. Use the training and validation data to develop and evaluate your model.

3 Train an Explainable Model

Train an explainable model using the training data. Use validation data to guide model development and apply appropriate practices such as reproducibility and model selection. Then use the final model to make output-probability predictions for every row of the test data for submission. You are encouraged to use AI tools to build your model and associated artifacts. Your approach should support explanation, discrimination testing and remediation, and model debugging in coming weeks.

4 Submit Code and Predictions

Your deliverable for this assignment has two parts. Each part is worth 5 points.

- Check in your code and associated artifacts to a public GitHub repository, or a private repository shared with the instructor (`jphall663`). Code should be available as a commented script, Jupyter notebook, R Markdown document, or other polished and professional format.
- Create a prediction file with output probabilities for each row of the test data. The file should have one column named `phat`, and should follow the `<personal_or_group_identifier>_<model_type>.csv` naming convention. See the example submission file.

Submit your code link and prediction file to the class evaluation server at a TBD URL by the deadline below. The class evaluation server will rank submissions using the cross-validated ranking method described in the model evaluation notebook. It will also use an LLM-as-judge to assess whether the submitted model is explainable. The five points for the prediction deliverable will be based on the ranking. You will have opportunities to improve your rank each week of the class.

Your deliverables are due TBD.