Lab Report #5: Classification & Ensemble Methods
Course: RSM338H1S, Winter 2026

Instructions:

- This assignment may be completed in **groups of up to 3 students**. Group members may be from either section. If you work in a group, submit one report with all names listed.
- Submit your lab report as a **PDF** to Crowdmark via Quercus. Export your Jupyter Notebook to PDF before uploading. Only one group member should upload the submission, being sure to select their group mates at the time of submission.
- There is no page limit, but be concise. A good report is thorough but not padded.

Marking:

- **75%** — Coding and results (correct implementation, complete answers to all parts, appropriate choice of methods, accurate numerical output, properly formatted tables and figures)
- **25%** — Overall quality (clear and professional writing, thoughtful interpretation of results, demonstrated understanding of the underlying concepts, logical flow and narrative structure)

Writing Expectations: Your report should read as a **coherent narrative**, not just code with scattered comments. Use section headers to indicate which problem you're working on. Before each code block, briefly explain what you are about to do and why. After results appear, interpret what you see. A reader should be able to understand your analysis even if they skipped the code cells.

You may use AI coding assistants (ChatGPT, Copilot, Claude, etc.) to help write code, but you must be able to explain what every line does. The text you write around the code is what demonstrates your understanding. You are ultimately responsible for your own work. **If you use an AI tool, you must disclose this in a note at the end of your report. Mention which tool you used, which tasks you asked it to complete, and discuss your (dis)satisfaction with its assistance.**

Assignment: You are an analyst at a consumer lending platform. The risk team wants to build a model that predicts whether a borrower will default on their loan, using information available at the time of application. Your job is to evaluate several classification methods—from simple baselines to ensemble models—and recommend one for deployment.

Data: You are provided with two files:

- `lending_clubFull_Data_Set.xlsx` — 25,000 Lending Club loans with 135 columns. This is the raw data as it comes from the platform.
- `lendingclub_datadictionary.xlsx` — A data dictionary describing each column.

The data is messy. Many columns have missing values, some are irrelevant (e.g., IDs, URLs, free-text descriptions), and the target variable `loan_status` contains multiple categories that you will need to recode into a binary outcome. This is intentional—real-world data requires cleaning before modelling.

Problem 1: Data Preparation

Before you can fit any model, you need to turn the raw data into something usable. Document every decision you make and explain your reasoning.

Tasks:

(a) **Define the target variable.** The `loan_status` column contains several categories (e.g., “Fully Paid”, “Charged Off”, “Current”, “Late”, etc.). You need a binary outcome: did the borrower default or not? Decide which categories count as default and which count as repaid. Report the number of loans remaining and the class balance.

(b) **Select features.** You have 135 columns. Many are useless (IDs, dates, free text), many are redundant, and many would not be available at the time a loan is approved (e.g., payment history after origination). Select a set of features that (i) would realistically be available *before* approving the loan, and (ii) are plausibly related to default risk. You do not need to use all of them—aim for a manageable set. List the features you chose and briefly justify your selections.

(c) **Handle missing values and data types.** Some columns have missing values, some numeric columns may need type conversion, and some categorical columns need encoding. Describe what you did and why. Report the final dimensions of your cleaned dataset.

(d) **Train/test split and standardization.** Split into training and test sets (80/20) using `train_test_split`. Standardize your numeric features using `StandardScaler` fit on the training set only.

Problem 2: Baseline Classifiers

Fit the following four classifiers on the training data and evaluate each on the test set:

1. **Logistic Regression**
   `sklearn.linear_model.LogisticRegression`
2. **Linear Discriminant Analysis**
   `sklearn.discriminant_analysis.LinearDiscriminantAnalysis`
3. ***K*-Nearest Neighbours**
   `sklearn.neighbors.KNeighborsClassifier`
   Use cross-validation on the training set to choose K. You decide how many folds to use—justify your choice. Try several values and report which one you selected and why.
4. **Decision Tree**
   `sklearn.tree.DecisionTreeClassifier`
   Pre-prune the tree. Use cross-validation to select your hyperparameters. Justify your choice of fold count.

Tasks:

(a) For each classifier, report its **accuracy** on the test set. Also record the **wall-clock time** each model takes to train (use Python's `time` module). Present your results in a table.

(b) For each classifier, display the **confusion matrix** on the test set. Briefly explain what the false positives and false negatives represent in this lending context, and which type of error is more costly from the lender's perspective.

(c) Given the nature of this dataset, is accuracy alone a sufficient measure of model performance for this problem? Think back to the other evaluation metrics we discussed in lecture. Which metric or combination of metrics would you use to evaluate these classifiers from the lender's perspective? Justify your choice, add the metric(s) to your results table, and use them throughout the rest of this report.

Problem 3: Ensemble Methods

Now fit three ensemble classifiers on the same training data:

1. **Random Forest**
   `sklearn.ensemble.RandomForestClassifier`
2. **AdaBoost**
   `sklearn.ensemble.AdaBoostClassifier`
3. **XGBoost**
   `xgboost.XGBClassifier`

Tasks:

(a) Each of these models has several hyperparameters that need tuning. For each model, explain which hyperparameters you chose to tune, what values you searched over, and why. Use cross-validation to select the best combination and report your results.

(b) Evaluate all three models on the test set using the metric(s) you chose in Problem 2(c), along with accuracy and training time. Add these results to your summary table from Problem 2 so that all seven classifiers are compared side by side.

(c) For the Random Forest, set `oob_score=True` when fitting the model. Compare the out-of-bag performance to the test set performance. What does this comparison tell you about how well the model generalizes?

(d) For all three ensemble models, produce a **feature importance** plot using each model's built-in `feature_importances_` attribute. Which features matter most? Does this align with your economic intuition about what predicts loan default?

(e) Do the ensemble methods improve over the baseline classifiers from Problem 2? By how much? Is the improvement large enough to justify the added model complexity? In your answer, discuss the trade-offs that come with more complex models from a business perspective.

Problem 4: Model Selection and Deployment

(a) Plot the **ROC curves** for all seven classifiers on a single figure. Include the diagonal reference line (random classifier). Which model has the best AUC? Is the ranking consistent with your summary table?

(b) Suppose the lending platform estimates that the cost of a **false negative** (approving a borrower who defaults) is 5 times the cost of a **false positive** (rejecting a borrower who would have repaid). Using your best model, experiment with the **classification threshold**: instead of predicting default when $P(\text{default} \mid \mathbf{x}) > 0.5$, try thresholds of 0.3, 0.4, 0.5, and 0.6. For each threshold, report the number of false positives and false negatives on the test set. Which threshold best reflects the 5:1 cost ratio? Explain your reasoning.

(c) **Final recommendation.** Which model and threshold would you recommend deploying? Write this as a memo to the risk team. Your recommendation should be well-reasoned and draw on evidence from your full analysis.