Machine Learning—Assignment 3

Publication date: 14/06/2025

Due date: 23/07/2025

Data

Data source: Google Drive Link

On this assignment, you will work on three datasets. They are:

Forest Cover Type—A dataset about classifying the type of trees according to various features (feature information is in the link). This dataset has been filtered to only include some labels and fewer observations. Do not download the dataset from Kaggle, use the one we provided from the Google Drive link above.

MNIST Digits—A dataset consisting of handwritten digits, represented as 28X28 grayscale pixel values. We have filtered the dataset to only include even digits.

Synthetic Data (Bonus)—A high-dimensional data synthesized for the bonus section.

IMPORTANT: The data is not currently split into train/test/validation sets. You need to split the data into three sets with a 80/10/10 % split (train, test and validation respectively), and train your model with the train set only. Hyperparameter tuning should only be done on the **validation** set, and once you find the best parameters, evaluate your model on the **test** set. This is true for all three sets.

There is no need to implement the models/algorithms you use in this assignment. You may use scikit-learn or any other library, with explanations and annotations.

Requirements

Part 1—Forest Cover Type (50 pts)

Section A - Data Exploration & Visualization (5 pts)

Explore the data using tables, visualizations, and other relevant methods.

- Plots should have an informative main title, axis labels and a legend (if needed).
- For each plot or table, provide a short description of **key observations**. Make sure to only include content which would be **meaningful/informative**.
- The visualizations should be detailed and cover all relevant aspects of the data.
- The visualizations should highlight any interesting patterns or trends that can be observed in the data, as well as key statistics such as the mean, mode etc. of each feature.
- Include **exactly 5** visualizations using **at least 3** different plot types. Each visualization should include a short explanation of what it shows and how it helps you understand the data or supports later analysis. Avoid adding visualizations that don't add value.

The goal of this section is to get insights on the data which may or may not be relevant for the following sections.

Section B - Data Pre-processing (10 pts)

Before starting this section, please read Sections C and D. Understanding the overall goal first might help guide better preprocessing decisions.

Apply different methods of pre-processing to the data in order to prepare it for the models you wish to apply in the next sections.

- Perform feature engineering on the data similarly to Assignment 2. Generate at least 3 new features. Explain the features you generated.

- Perform other pre-processing steps as we discussed in class (Imputation, transformation etc.). Explain your choices.

Section C - Classification and Clustering (15 pts)

Classification

Use at least **three** different machine learning models we have studied in class to predict the label of each observation (1, 2 or 3).

- The implementation must include parameter tuning.
- Report a suitable measure to evaluate the performance of each model.
- Present the models' results in a plot.
- Compare the results of the different models, discuss them.

Clustering

- Use **two** different clustering algorithms and attempt to cluster the data **before dimensionality reduction**. Each cluster should represent the label of the observations in it. Evaluate the clusters you created using several evaluation methods of your choice—explain which metric or method you used and what information it gives.
- Attempt to visualize the clusters you found, **again before dimensionality reduction**. How well did the visualization go? If the clusters are unclear—why?

Section D - PCA (20 pts)

Perform Principal Component Analysis on the data. Retain the number of Principal Components (PCs) that explain >80% of the variance (you can modify this value as you wish—explain why if you choose to do so). Then perform the same tasks as Section C with the same models/algorithms. Compare the results of the lower-dimension data to the results of Section C. Discuss differences in performance and clustering—why was it better/worse?

Part 2—MNIST (50 pts)

Section A - Visualization (5 pts)

- Visualize several of the observations, so that the digit is clearly visible. Visualize at least two of each digit.
- Plot a heatmap of the data, showing the average value of each pixel (feature) to see which pixels are the most “popular”.
    - Use `.reshape(28, 28)` on the averaged data so the heatmap would show as a 28 X 28 “image”.
    - Explain why this visualization is useful for this dataset.

Section B - Classification (10 pts)

Use **two** machine learning models we have studied in class to train and classify the data (the digits/labels are 0, 2, 4, 6, 8), similar to Part 1 Section C:

- The implementation must include parameter tuning.
- Report a suitable measure to evaluate the performance of each model.
- Present the models’ results in a plot.
- Compare the results of the different models, discuss them.

For this section, log the training time and predict time for each model for comparison later.

Section C - PCA and Feature Importance/Selection (25 pts)

- In this section, perform PCA in the same manner as Part 1 Section D. Log the training and predict time (**not** the time it takes to perform PCA).
- After you run the models on the reduced-dimension data, report on the 5 most important features and 5 least important, from the PC loadings (the "amount" each feature contributes to the variance each PC explains). Take care to verify your process is correct—we have not gone over it technically in class.
- Choose a value for the variable `drop_percent`. This value would set the number of features to drop, whose importance (the total variance they explain) is up to `drop_percent`. For example, if drop_percent = 0.1, then we will drop the "worst" features which in total explain up to 10% of the variance across all PCs.
- Report on what features those are and the total number you will drop. Do these features make sense to drop?
- Drop those features from the **original data**. Then perform section B and C again (run models on the data after dropping features before and after dimensionality reduction). No need to calculate feature importance and selection again. In total you should have 4 runs (for each model) across sections B and C.
- Compare the results (evaluation metrics and runtime) across all 4 runs and discuss them. Why were some better/worse? Did they match your expectations?

Section D - t-SNE (10 pts)

Visualize the data using t-SNE in 2D or 3D. Color each observation of each label in a different color. Are clusters apparent in the visualization (If not, try different hyper-parameters)?
Why would trying to visualize clusters without dimensionality reduction, like in Part 1 Section C, be difficult?

Part 3 - Bonus (25 pts)

- In this part, you will work on synthetic data (computer-generated without any semantic meaning). The data is high-dimensional, consisting of 120 features (f0–f119), a label (0 or 1) and is_outlier describing if the observation is an outlier (1) or not (0). There are 100 total outliers.

- **Important**—Some of the features are useless, and some are highly correlated with others.
- Train a model of your choice on the data, and predict the class on a test set. Evaluate and report your findings.
- After doing so, run feature importance analysis in any manner of your choice. Drop features you deem unimportant. Take care to manage highly correlated features correctly.
- Train the same model again on the data without those features and compare the results to the previous run.
- Try to identify outliers in any manner of your choice (finding extremes, visualizing PCA etc.). Report the accuracy and recall of the outliers you found using the is_outlier label.

Guidelines

Please read the following section carefully before submitting the assignment.

Coding Guidelines

- Use familiar packages with explicit explanations.
- If you have installed any libraries beyond those presented in the exercises, please specify this in the report.
- The code should run without warnings or errors.
- Good documentation is **critical**.
- Indicate the exercise sections in the code as well.
- Use meaningful variable names.
- Do not use reserved words.
- Use constants where possible.

Submission Guidelines

- The assignment should be submitted in pairs (only one submission).

- You are required to submit two files including all the sections. One in **.ipynb** format and one in **.html** format. **Both files should also include the program's outputs**. In addition, you are required to upload a **PDF** file of the presentation you prepared.
- The files' names should be of the form: **ML_HW2_ID1_ID2**.
- Assignments submitted late will receive a penalty of **3 points** for each day, up to one week. Later submissions will not be accepted.

Grading

You can get more than 100 points for the exercise. The exercise will be graded according to correctness, clarity, efficiency of implementation, elegance of implementation.

Self-learning

As we mentioned at the beginning of the course, self-learning is an important part of the course. Treat all sources of information carefully and critically.
Usage of LLMs is allowed, but reference it where it was used.
You can and should consult with other students in the course, but each pair must write their own work.
It is reasonable to assume that not all results and algorithms will be identical.

Questions and Reception hours

- Please post your questions on the exercise forum in Moodle, after you have read the previous posts. Professional questions sent by email will not be answered.
- If you want to schedule a reception hour with one of the instructors - please send your questions by email in advance.
- In any other case (personal questions, request for an extension with a justified reason, etc.) please email the instructor.

Good luck