

# Final Project

CS-UY 4563 - Introduction to Machine Learning

## 1. Overview

In this project you will partner with **one classmate** to apply machine learning methods learned in this course to a dataset of your choice. Your work will culminate in a presentation and a project submission that includes a write-up, code, and (if relevant) the dataset or a link.

Your tasks:

- Choose a dataset and clearly formulate a learning task (classification or regression). As a rough guideline, your dataset should have **at least 200 training examples** and **at least 10 features**.
- Perform exploratory data analysis and feature preprocessing.
- Train and evaluate multiple models, compare results, and reflect.
- Present your findings to the class at the end of the semester.

## 2. Submission Requirements

**Due:** Evening before the first presentation session (exact date TBA).

You must submit on **Gradescope**:

- Presentation slides (PDF or PowerPoint).
- Project write-up (PDF only). Other formats will incur a 5% deduction.
- Executed Jupyter Notebook with visible output and code. Failure to comply with this format may result in deductions or a requirement to re-submit.
- If using custom dataset, upload it to Gradescope (or provide a GitHub link, if necessary)

### 3. Project Phases and Write-Up Structure

#### A. Introduction

- Describe your chosen dataset: source, size, features, and target.
- State your machine-learning task: what are you predicting, and why it is interesting.

#### B. Exploratory & Unsupervised Analysis

- *Visualize feature distributions* (e.g., histograms, density plots), *relationships with the target*, and *correlation matrix* to show relationships between features
- Apply unsupervised methods (for example: PCA for dimensionality reduction or K-means clustering to uncover structure).
- Document and interpret any patterns or structure you discover — if you found none, still describe what you attempted.
- *Preprocess your data:* handle missing values, scale features, consider other transformations (e.g., log, power for skewed data).

#### C. Supervised Modeling

Train at least **three distinct learning models**<sup>1</sup> discussed in the class (i.e., Linear Regression, Logistic Regression, KNN, Neural Networks, CNN).<sup>2</sup> You may use your own implementation (from homework or developed independently) or libraries (*scikit-learn*, or *Pytorch*, or *Keras*).

For each approach,<sup>3</sup> you must:

- Try it on the data after preprocessing.
- Apply at least **three feature transformations** that map the data into a new feature space (Z-space). The feature transformations should increase the model complexity or help to find a better set of basis functions,<sup>4</sup>
- For each transformation (and non-transformed data) try at least 6 different regularization values per approach.

---

<sup>1</sup>You can turn a regression task into a classification task by binning, or for the same dataset, select a different feature as the target for your model.

<sup>2</sup>If you wish to use a model not discussed in class, you must discuss it with me first, or you will not receive any points for that model. If we have time and cover SVM, you may use that as well.

<sup>3</sup>Even if you get a very high accuracy, you are still required to perform these feature transformations to see what happens.

<sup>4</sup>While scaling is a feature transformation, it is considered a preprocessing step for this project and does not count toward the required feature transformations.

Note this means you will train at least:

3 approaches  $\times$  (3 transformations + 1 untransformed)  $\times$  6 hyperparameter settings = 72 models

**Be thoughtful in your choices.** For each feature transformation or regularization value you try, briefly explain why, e.g., “Model was overfitting, so I increased regularization,” or “Training error was high, so I added features.”

**Some additional clarifications/ comments are posted on EdStem.**

## D. Table of Results

- Provide a **table** with *training* accuracy and *validation* metrics for every model. Include results for the different parameter settings (e.g., different regularization values).
  - For classification include metrics such as precision/recall.
  - For regression models, report metrics like MSE,  $R^2$ . For example, suppose you’re using Ridge Regression and manipulating the value of  $\lambda$ . In that case, your table should contain the training and validation accuracy for every lambda value you used.
- **Plot/Graph** and analyze how performance metrics (like accuracy, precision, recall, MSE) change with different feature transformations, hyperparameters (e.g. regularization settings, learning rate).

## E. Conclusion and Analytical Discussion

- Interpret your findings: Which approach, feature transformations, or regularization helped? Why?
- Explain key findings. Provide a **chart** of your key findings.
- Highlight the impact of feature transformations, regularization, and other hyperparameters on the model’s performance. Refer to the graphs provided in earlier sections to support your analysis. Focus on interpreting:
  - Whether the models overfit or underfit the data.
  - How bias and variance affect performance, and which parameter choices helped achieve better generalization.
  - Reflect on limitations: What would you try next or change?

## 4. Presentation Notes

- You and your partner will give a 6-8 minute presentation to the class.

- Presentations will be held during the last 2 or 3 class periods and during the final exam period for this class. You will be assigned a day for your presentation. If we run out of time the day you are to present your project, you will present the next day reserved for presentations.
- **Attendance during all presentations is required.** A part of your project grade will be based on your attendance for everyone else's presentation.

## Important Notes on Academic Integrity

- Your submission will undergo plagiarism checks.
- If we suspect you of cheating, you will receive 0 for your final project grade. See the syllabus for additional penalties that may be applied.

## Dataset Resources

Below are some resources where you can search for datasets. You are free to use these resources, look elsewhere, or *create your own dataset*.

- <https://www.kaggle.com/competitions>
- <https://www.openml.org/>
- <https://paperswithcode.com/datasets>
- <https://registry.opendata.aws/>
- <https://dataportals.org/>
- [https://en.wikipedia.org/wiki/List\\_of\\_datasets\\_for\\_machine-learning\\_research](https://en.wikipedia.org/wiki/List_of_datasets_for_machine-learning_research)
- <https://www.reddit.com/r/datasets/>
- <https://www.quora.com/Where-can-I-find-large-datasets-open-to-the-public>

## Modifications

- If you have a project idea that doesn't satisfy all the requirements mentioned above, please inform me, and we can discuss its viability as your final project.
- If you use techniques not covered in class, you must demonstrate your understanding of these ideas.

## Brightspace Submissions Guidelines

- **Dataset and Partner:** Submit the link to your chosen dataset and your partner's name by November 10th
- **Final Submissions:** Upload your presentation slides, project write-up, and code to Gradescope by the evening before the first scheduled presentation. The exact date will be announced once the total number of projects is confirmed. (I expect the due date to be December 2nd or December 7th.)<sup>5</sup>

## Potential Challenges and Resources

As you work with your dataset, you may encounter specific challenges that require additional techniques or tools. Below are some topics and resources that might be useful. Please explore these topics further through online research.

- **Feature Reduction:** Consider using PCA (which will be covered in class). If you choose to use SelectKBest from scikit-learn, you must understand why it works before you use it.
- **Creating Synthetic Examples:** When using SMOTE or other methods to generate synthetic data, ensure that only real data is used in the validation and test sets.
  - If using synthetic data, make sure your validation set and test set mirrors the true class proportions from the original dataset. A balanced test set for naturally unbalanced data can give misleading impressions of your model's real-world performance. For more details, see: Handling Imbalanced Classes
- **Working with Time Series Data:** For insights on working with time series data, visit: NIST Handbook on Time Series.  
For learning how to use  $K$ -fold cross validation with time series, visit: .
- **Unbalanced Datasets:**
  - <https://www.sciencedirect.com/science/article/pii/S0957417424009849>
  - <https://machinelearningmastery.com/tour-of-evaluation-metrics-for-imbalanced-classes/>
- **Handling Missing Feature Values:**
  - See Lecture 16 at Stanford STATS 306B
  - Techniques to Handle Missing Data Values

---

<sup>5</sup>Top presentation gets a 8 pt bonus, second best gets a 6 pts bonus and third best 4 point bonus (rated by course assistants and myself)

- How to Handle Missing Data in Python
- Statistical Imputation for Missing Data

- **Multiclass Classification:**

- Understanding Softmax in Multiclass Classification
- Precision and Recall for Multiclass Metrics

- **Optimizers for Neural Networks:** You may use Adam or other optimizers for training neural networks.

- **Centering Image Data with Bounding Boxes:** If you are working with *image data*, you are allowed to use *bounding boxes* to center the objects in your images. You can use libraries like *OpenCV* ('cv2').

## Tips

*Don't forget to scale your data* as part of preprocessing.<sup>6</sup> Be sure to document any modifications you made, including the *scaling or normalization techniques* you applied.

Don't confuse *validation* and *test* data.

The following resource might be helpful: CS229: Practical Machine Learning Advice  
Please stick to topics we discussed in class or those mentioned above.

---

<sup>6</sup>Log scaling may be useful for certain features.