CMSE 492 Final Project

Learning Goals and Purpose

This course is project-based, and the final project serves as your capstone demonstration of applied machine learning skills—you can think of it as your "final exam," since there are no traditional exams in this course.

The project serves several important goals. One of the primary goals is to ensure that you have something substantial to show at the end of this course. You will create a complete end-to-end machine learning project that is accessible from your personal GitHub account, demonstrating your ability to tackle real-world problems with machine learning. This project will showcase your skills in data analysis, model development, critical thinking about model selection, and technical communication—all essential skills for careers in data science and machine learning.

In addition to building technical machine learning skills, you will develop excellent communication abilities through writing a comprehensive technical report. You can link to your project repository from your CV and use your CMSE 492 project in future job interviews, graduate school applications, or when discussing your work with colleagues and collaborators.

Nearly all of the work on the project is done by you outside of class. You know already that you will be **tempted to procrastinate**, so now is the time to think about how to trick yourself so that you don't do that. I will help you by providing checkpoints and milestones throughout the semester. I highly recommend using a calendar and putting milestones for this course into it with early reminders before due dates. Start exploring datasets and brainstorming project ideas early—the best projects come from genuine curiosity and steady, consistent effort.

Dataset Selection

You are free to choose any dataset that interests you for this project, with the following requirements and restrictions:

Allowed Sources

- Kaggle datasets (excluding overused competition datasets)
- UCI Machine Learning Repository
- Government open data portals (data.gov, census data, etc.)
- Research institution datasets
- Industry-specific datasets (finance, healthcare, climate, etc.)
- Datasets you collect or compile yourself (with proper documentation)

Prohibited Datasets

You **may not** use commonly available toy datasets that are frequently used in tutorials and examples, including but not limited to:

Datasets built into common libraries:

- Any dataset that can be loaded directly from scikit-learn (`load_iris()`, `load_wine()`, `load_diabetes()`, `load_breast_cancer()`, `fetch_california_housing()`, etc.)
- Any dataset built into seaborn (`tips`, `titanic`, `penguins`, `diamonds`, etc.)
- Any dataset built into statsmodels, tensorflow.datasets, or other common ML libraries

Overused benchmark datasets:

- MNIST handwritten digits
- Fashion-MNIST
- CIFAR-10/CIFAR-100
- Iris flower dataset
- Boston Housing (deprecated)
- Wine quality dataset
- Titanic survival dataset
- Any dataset from Kaggle's "Getting Started" competitions

Dataset Requirements

- **Minimum size:** At least 1,000 samples (rows)
- **Sufficient complexity:** At least 5 features, with meaningful relationships to explore
- **Real-world applicability:** The dataset should address a genuine problem or question
- **Proper documentation:** You must be able to clearly describe the data's origin, collection method, and meaning of each feature

Note: Choosing an interesting, well-suited dataset is part of the learning experience in applied machine learning. The dataset you choose will significantly impact the quality of your analysis and the insights you can derive.

Written Report

Your **written report** is a narrative that explains what you did, why you did it, and how you went about it. The report must contain the following meaningful sections so that the instructor can follow the logical flow of your work:

The report should be written in LaTeX and a template will be shared with you. You are not allowed to change the settings of the template and you only need to fill the sections with your narrative. You can use Overleaf to complete the report. If you don't know how to use Overleaf follow the instructions provided in the document in D2L. You can create a free account with the MSU email.

Report Sections

Background and Motivation

Describe the problem/question you are attempting to answer. This section must answer the following questions:

- Why is this problem/question important?
- Who cares about this problem/question being solved/answered?
- What are the consequences of solving this problem/answering this question?

- What has been done so far to address this problem/question?
- State very clearly what the desired outcome is. How can Machine Learning (ML) help achieve your goal and/or solve your problem?

Data Description

Describe your data and any issues there might be. This section should have a clear answer to all these questions:

- **Origins of the data:** This does not mean "I got the data from Kaggle". Instead, you should read the description and metadata of the dataset and report that. For example: "The MNIST dataset consists of 60,000 images of handwritten digits written by 500 high school students in Bethesda. The dataset was originally assembled by the US Census Bureau in the 1990s."
- How many columns and rows do you have?
- What type of data is there? Numerical, categorical, time series, geographical?
- Are there missing values? What do you think is the missingness mechanism? Pattern? How did you arrive at this conclusion?
- Is the dataset balanced? What technique are you going to use to balance the dataset?
- Show some statistics of the data: correlations, univariate and bivariate distributions, ranges of the data, outliers

Preprocessing

Describe the preprocessing steps and why you are doing these steps:

- **Splitting:** How are you going to split the data and why you chose it. Stratified splitting, random splitting, time series splitting? Recall that the splitting should happen before you do any EDA.
- Scaling, Transformation, Encoding, Imputation
- Feature engineering techniques. For example: "We used K-means clustering to create 5 clusters of the CA districts", or "We created polynomials up to degree 10 for all the features"

Machine Learning Task and Objective

This section focuses on the machine learning aspect of the project. This section should have clear answers to the following questions:

- Describe why we need ML and how humans or current methods fail at this task.
- What type of ML task is this?
  - **Supervised:** Regression or Classification
    - Interpolation, Extrapolation, Binary, Multiclass, Multi-label, Multi-output
  - **Unsupervised:** Dimensionality Reduction or Clustering
  - **Reinforcement Learning:** Value-based, Policy-based, Actor-critic, Policy-learning

Models

Describe the machine learning models you will compare. If you use deep neural networks, you must compare with other methods so that you can demonstrate the power and convergence of deep learning. This section should include completely different algorithms (e.g., decision trees versus logistic regression).

The point is that you cannot know the power of a method if you don't compare it to other methods. (Who knows, maybe guessing is better than your ML algorithm!)

- Describe the models you are going to use and how they will be evaluated. You need at least three models in increasing order of complexity. For example: Regression task → Linear Regression with polynomial features and L2 regularizer, Gradient Boosted Random Forest, Deep Neural Network.
- Describe the neural network architecture and why you chose this one.
- Describe the regularization and hyperparameter tuning procedures if any (Linear regression does not need any hypertuning)

Training Methodology

For each model describe how training is performed, write down the equation for the loss function, and any technique used to track the learning of your model and avoid over- and under-fitting. This subsection must include plots of the learning curves or other metrics used to track the learning process. In addition, this section should include *hyperparameter tuning*, *unsupervised learning techniques*, *cross validation*, *bootstrapping*, etc.

This section must contain a table with these columns:

Model	Parameters	Hyperparameters	Loss Function	Regularization

Metrics

Clearly define the metrics you will be using to evaluate the performance. How do you know that your model is doing well, e.g., RMSE, MSE, F1 Score, precision, recall? Explain your choice.

Results and Model Comparison

Compare the different algorithms using the metrics defined above. Compare the algorithms on their difficulty in training (time and hardware resources). Explain your choice of best algorithm for the task. Explain why some models perform better than others and/or why all the models are not performing well. This section must include:

- Tables with the training and inference time of each model
- Tables comparing the metric scores of the models

Model Interpretation

Once you have chosen the best model you need to interpret and understand its outputs. Feature importance and RFE, SHAP values.

Conclusion

Summarize what you have done. Which is the best algorithm for the task and why? Did your algorithm achieve the desired score? In addition, describe what went wrong and how you think you could solve the issues in the future.

Submission Guidelines

- Submit your written report as a PDF to D2L
- Include the link to your GitHub repository in D2L
- Ensure your GitHub repository includes:
  - All code used for data preprocessing, training, and evaluation
  - A README.md file with instructions on how to run your code
  - Requirements.txt or environment.yml file for dependencies
  - Clear documentation and comments in your code

Due Date: TBD

Presentation Date: See course schedule (Week 15: December 2 and 4)