ECE133A Winter 2025
Final Project Deliverables

February 3, 2025

Description Please read Chapters 13, 14, 15, and 18 before starting the project. This will help you speed up the process and understand the requirements described here better. The final project tests your ability to apply the algorithms and tools you learned from class to some real-world dataset. It is intended to be open-ended, meaning while you have general constraints on what you should deliver, you are free to choose the dataset and the type of models you use for a prediction task of your choice. You should write your codes in either Python or Matlab, and use consistent formatting for your reports, for example, it's easy to accomplish in LaTeX. You should form a team of 2 or 3 students, and you can look for potential teammates on the class piazza. The final project weighs 50% of your final grade. The reports are due at 11:59pm on the due dates.

Deliverables and timeline

1. **Description of the data and data cleaning.** The report for this part is due February 21, 2025 and is worth 5% of the total 50%.
   (a) Pick a dataset that catches your interest and you can perform some sort of prediction on. You might find the following databases useful:
      - UCI Machine Learning Repository for datasets categorized by the type of prediction you can perform on them;
      - DataHub.io for continuously updated datasets like stock prices;
      - kaggle for a variety of datasets;
      - Earth Data for data compiled by NASA;
      - Pew Research Center for datasets used in Social Sciences;
      - You can also scrape your own data or obtain it in some other way.

   (b) Decide on what type of prediction you will carry out: either classification or model fitting (regression). Please, note that "regression" in the textbook refers to a specific type of linear model fitting, but, in general, it can refer to any model fitting.

   (c) Clean up your dataset by identifying and removing irrelevant and noisy entries, etc. and compile it in a useful format like a matrix.

   (d) Dataset requirements: once you clean it, you should have a matrix with at least 10,000 rows (samples) and 30 columns (features).

   (e) Discuss the preliminary trajectory your project is going to take and come up with a plan in your report.

   (f) Provide any code you wrote up as well (you can provide a link to the code if you create a Github repository for the project or a Google Collab page).

2. **Understanding the structure of the dataset**. The report for this part is due February 28, 2025 and is worth 10% of the total 50%.
   (a) Standardize your raw features. Please, refer to Chapters 3 (3.3) and 13 (13.3.1) for more information. Report the mean and standard deviation of every feature across the samples.

(b) Perform k-means clustering described in Chapter 4 for different values of k. Report the best k you could find.

(c) Perform Singular Value Decomposition (SVD) to your standardized dataset. Report the features with the highest singular values. (For your reference, please read carefully the following webpage: `https://towardsdatascience.com/understanding-singular-value-decomposition-and-its-application-in-data-science-388a54be95d`)

(d) Construct the correlation matrix/table for the feature vectors. If A is the correlation matrix, then $A_{i,j}$ is the correlation coefficient (Chapter 3.4) between the columns i and j of the original (standardized) dataset. Report on the highly correlated features.

(e) Please write a report showing your results, discussion and provide your code.

3. **Model Evaluation and Prediction.** The report for this part is due March 9, 2025 and is worth 25% of the total 50%. Please read through Chapters 13, 14, 15 and 18 before you start this step.

(a) Start with a basic linear model (the features are only standardized and are used as they are without any transformations) regardless if you are doing the least-squares data fitting (Chapter 13) or the least-squares classification (Chapter 14) task. Evaluate the initial model using cross-validation and report the RMS error. Make sure to save the model parameters for each fold of the cross-validation.

(b) Perform feature engineering. Come up with more interesting feature mappings or basis functions for the linear least-squares data fitting or the linear least-squares classifier. For example, try out a stratified model (Chapter 13.3.2) using the results of the k-means clustering you did earlier. Choose the best model using cross-validation and report the RMS error. Make sure to save the model parameters for each fold of the cross-validation. If you are doing classification, report the confusion matrix for the best model you found.

(c) Separate the training and testing set. Read through Chapter 15.4, and add regularization to your best model from the previous step. Choose the best regularization parameter from evaluating the models on the test set and report the RMS error. Now, revise your feature engineering step, since regularization allows one to incorporate more interesting basis functions while avoiding overfitting. Report if regularization helped your model to be more accurate and general. Make sure to save the norm of the best model parameters. If you are doing classification, report the confusion matrix for the best model you found.

(d) Try one non-linear data fitting (Chapter 18.4) or nonlinear least-squares classifier (Chapter 18.5). You can do some research to find out what non-linear models are usually used for your dataset and the type of prediction you are doing. For example, there might be an underlying nonlinear physical model to predict the temperature in a certain area based on past temperatures. Evaluate your model using cross-validation and report the RMS error. Make sure to save the model parameters for each fold of the cross-validation. If you can't find any non-linear model, you can use orthogonal distance regression (Chapter 18.4). If you are doing classification, report the confusion matrix for the best model you found.

(e) Please write a report showing your results, discussion and provide your code. You can use the best models you found in 3(a), 3(b), 3(c) and 3(d) to predict on some unseen data and compare the predicted outcomes.

4. **Complexity analysis.** This final part is due March 18, 2025 and is worth 10% of the total 50%.

(a) Compute the number of flops required to evaluate the best models of 3(a), 3(b), 3(c) and 3(d). Start counting the flops from the basis functions computations through validation. Compare the complexity of each model against their RMS errors.

(b) Study the numerical stability of the best models from 3(a), 3(b), 3(c) and 3(d) by examining the model parameters. For example, take a look at Table 13.5 where for each fold of the 5-fold cross-validation, model parameters are saved. Then, they examine if those parameters are close to one another. You can check that by looking at the mean and standard deviation of each parameter across the folds. For part 3(c), you can just comment on the value of the norm of the model parameters.

(c) Write a final report combining all the previous parts and draw appropriate conclusions.