Final Project

DS 4021 – Instructor: Javier Rasero; **TAs:** Jeffrey Zhang, Ethan Meidinger
Due: See Canvas
Submission format: GitHub repository (submitted by link in Canvas)

Group Assignment

General Description: Submit to Canvas the link to your group's GitHub repository for this project. This link should be **the same** that you created for the prerequisite assignment, and will contain all materials related to your final project, including Jupyter notebooks, documentation, figures, and a final report summarizing the results.

We will use GitHub's commit history to verify submission times. Any commits after the deadline will be considered late, and the final project **will receive no credit.**

Why am I doing this? The goal of the final project is to put into practice everything you have learned throughout this course. Using the same dataset and research question that you selected previously, you will perform exploratory visualization, descriptive statistics, and the training and optimization of four predictive models: a linear model with penalization, a Support Vector Machine, a neural network implemented in PyTorch, and an ensemble method. Proper validation through cross-validation, model comparison, and final evaluation on the held-out test set are required.

What will we do? Your group will complete a full machine learning analysis on your selected dataset and write a concise report presenting the main results and findings. Work should be divided among team members.

How will this be graded? This final project is worth 100 points.

- The notebooks portion is worth 80 points.
- The final report is worth 20 points.

See the table below for a detailed description of where the points come from.

Important: Although only the notebooks and the final report count toward the grade, your repository must include all required files and folders **outlined below** in order to receive full credit.

Formatting	● One Github Repository (submitted via link on canvas) ● The top level page of the repository should contain: ○ A README.md file (which auto displays) ○ A NOTEBOOKS folder ○ A Data Folder ○ A OUTPUT folder ○ A final_report.pdf file
README.md	● Goal: The README helps others quickly understand your project. Keep it short, clear, and written for someone who has never seen your files before. ● Use markdown headers to divide content. ● Make an H2 (##) section explaining the contents of the repository It must include two simple sections:

	● Section 1: Software and platform section List the main tools needed to run your notebooks. For example: ● Python version ● Jupyter notebooks ● Packages (pandas, numpy, scikit-learn, matplotlib, seaborn, PyTorch) ● Operating system (Mac, Windows, or Linux) ● Section 2: A Map of your documentation. In this section, you should provide an outline or tree illustrating the hierarchy of folders and subfolders contained in your Project Folder, and listing the files stored in each folder or subfolder.
NOTEBOOKS folder	● Goal: This folder contains all the jupyter notebooks for your project. These are the required notebooks you will need to upload: 1. **Descriptive Analysis Notebook (10 points):** ○ Exploratory visualizations ○ Summary statistics 2. **Models optimization and training Notebook (s) (60 points)** Must include complete training and optimization of: ○ A Penalized (Ridge, Lasso or ElasticNet) linear model (Linear Regression or Logistic Regression). ○ Support Vector Machine ○ Ensemble model (e.g. Random Forest or Gradient Boosting) ○ Neural network implemented in PyTorch You may use one combined notebook or separate notebooks for each model. **REMEMBER**: For the optimization and training stage, **you must not use the test set** you put aside in the prerequisite Final Project assignment. Using the test set for tuning or training will result in **major penalties**. 3. **Final Test-Set Evaluation Notebook (10 points)** ○ Select the best model (**and by best, this also includes its optimal hyperparameters**) from the optimization phase and fit it to the **entire training set**. ○ Evaluate it on the **test set** you had initially put aside. ○ Report final results. **REMEMBER**: Here is where you **must use** the test set you put aside in the prerequisite Final Project assignment. Not evaluating on the correct test set will result in **zero points** for this part. **Notebook Requirements** ● Use cross-validation to tune your models. ● Try different hyperparameter combinations.

	● Embed preprocessing inside the cross-validation to avoid any data leakage. You may accomplish this using pipelines. Failing to do this will incur in **penalties**. ● Assess and report the final model's performance using more than one metric. e.g. Accuracy and confusion matrix for *classification*, Coefficient of determination and Mean Square Error for *regression*. ● Cross-validation and hyperparameter tuning must be clearly documented. ● Notebooks must run top-to-bottom without errors. ● Code must be accompanied by explanatory text. ● Figures must be readable and well labeled. ● Random seeds must be set whenever possible for reproducibility. Scikit-learn objects typically include random_state as an argument for this. For Pytorch, you may use *torch.manual_seed.*
Data folder	● This is the **same folder** where you originally uploaded the test set. ● For the final project, this folder should also contain the training set. ● If any of these pieces of data are too large for GitHub, upload it to a platform such as Google Drive and include a link with appropriate download permissions.
OUTPUT folder	● Goal: Store all outputs generated during the project. For example: Figures, Tables, any materials used for your final presentation, etc.
final_report.pdf	● Goal: to summarize everything into a final report file. ● A PDF file 1 or 2 pages long. ● **20 points** towards the final grade. **Structure** ● One introductory paragraph with ○ Research question. You may reinstate or refine the same research question you used for the prerequisite document. Remember that your research question needs to be testable. For example, something like: *"which factors contribute most to...", "which features are the most predictive…"* might now be directly addressable for some models (e.g. Neural Networks or SVM with non-linear Kernel). A good and testable research question could just be: *"Can we successfully predict [SOMETHING]?"* "*What model best predicts [SOMETHING]?"* ○ Motivation ● One or two methods paragraphs ○ Dataset description. ○ Summary of any data cleaning/curation you needed to perform. ○ Outline each model used, including the preprocessing steps each needed to include. ○ Cross-validation strategy. ○ Metrics used to assess model performance. ● One–two results paragraphs ○ Key findings and model results. ○ **Include one table** showing the cross-validated performance of each best model ○ Report the performance for the selected best model on the test set, i.e. the piece of data you prepare for this at the very beginning of the project.

	● One–two discussion paragraphs ○ Interpretation of results ○ Why some models may perform better than others ○ Limitations of the project