NSYSU Statistical Learning and Data Mining–Fall 2026

Final Project

Goal

The primary objective of this course is to equip you with the ability to implement statistical learning techniques to solve real-world application challenges, as well as to provide a solid foundation for those interested in pursuing research in statistical learning or machine learning domains. The final project is designed as a stepping stone towards achieving these goals. It will serve as a platform for you to immerse yourself in a particular dataset, to thoroughly comprehend its intricacies, and to employ your skills in Exploratory Data Analysis (EDA) and predictive modeling to extract meaningful insights. This project is your chance to showcase your analytical prowess and the practical application of the concepts learned throughout the course!

Topics

The project topics for your final assignment are categorized into two primary streams:

1. **Application-oriented**. This may be the preferred choice for most. Find a dataset or a real-world issue that captures your interest, and use the statistical learning tools we've discussed in class to explore and understand it. Begin with Exploratory Data Analysis (EDA), which is like detective work to uncover what the data is telling you, often revealing patterns, anomalies, or relationships. Your dataset might be a bit untidy, so you'll likely need to clean it up and organize it before you can fully analyze it. After working through the data, your goal is to come up with some clear insights or findings about it. For instance, if you're looking at a dataset of local weather patterns, you might end up predicting rainfall amounts or identifying climate trends. Open-government and other real-world data are especially encouraged over popular, heavily analyzed datasets.
2. **Research-oriented**. If you're drawn to deep exploration, this is your route. Choose a statistical method or concept we've covered, like Generalized Linear Models (GLM), and aim to understand it deeply. You might review the original research papers or existing open-source code, and then try to recreate the algorithm in a simplified form. Once you fully understand it, you could even attempt to improve it or adapt it in a new way. Alternatively, you could compare various approaches to a common data problem, like how different algorithms perform in ensemble methods or explain the differences in tackling Multiclass, Multioutput, or Multilabel classification problems. If this is your project choice, make sure to thoroughly read and analyze related research to inform your project. Note that a working reimplementation alone is not enough: explain why the method works, where it fails, and how your work differs from existing code.

Your project should be based on something that excites you, whether it's a specific type of data, a question you want to answer, or a method you wish to master. Your work **must be original** for this class, but it can build upon what you've learned from your past projects. It's fine if others in the class choose similar topics, as long as each person or group works independently. This way, even with the same starting point, each project can take a unique direction based on your individual analysis and perspective. AI tools (e.g., ChatGPT, Claude Code, Codex, OpenCode) are encouraged, but their use must be disclosed (see Acknowledgements and AI Use below), and your team remains responsible for every number, figure, and claim.

The following sources are suggested to search for datasets (none is required)

政府資料開放平台
內政部不動產實價登錄
主計總處總體統計資料庫
中央銀行統計資料庫
中央氣象署開放資料平臺
Twinkle Hub (MCP-based search and summary queries of Taiwan government data; the free plan returns at most 10 rows per query, so use it to find and screen datasets, then download the original files from the source agency)
FRED
Our World in Data
Kaggle
Google dataset search

A guide (in Chinese) to what these sources can and cannot do with AI coding agents such as Claude Code, Codex, and OpenCode, how to connect them, and what to check when using AI for analysis: https://phonchi.github.io/statlearning-selfstudy/project/
The following packages are suggested for looking for algorithms that you are interested in or you can use when you are exploring your dataset

- https://intro-stat-learning.github.io/ISLP/
- https://scikit-learn.org/stable/
- https://www.statsmodels.org/stable/index.html
- https://rasbt.github.io/mlxtend/
- https://github.com/scikit-learn-contrib/scikit-learn-contrib/blob/master/README.md
- https://github.com/rapidsai/cuml (GPU only; optional)

If you have your own ideas for a project or feel unsure about how to begin, don't hesitate to reach out. You're welcome to drop by during the office hours of your Teaching Assistant or mine. We're here to listen to your thoughts and provide guidance. Whether you need help refining your idea or finding a starting point, we're eager to assist you in launching a project that's both exciting and rewarding.

Example Workflow

1. **Application-oriented Project**
   - ✓ Dataset Selection:
     - Choose a dataset from a suggested list or any other reputable source. Aim for less common datasets to avoid repeating analyses. If it's a popular dataset, focus on a distinct question or uncovering unique insights.
     - Avoid datasets that are extensively covered in well-known books or resources, as these have likely been thoroughly examined.
     - Record the source, license, and retrieval date, and keep a raw copy of the data before cleaning. For Kaggle or other secondary datasets, trace the data back to its original source.
   - ✓ Data Preprocessing and EDA:
     - Clean and prepare your data for analysis.
     - Conduct exploratory data analysis using unsupervised methods or create various plots to deeply understand the characteristics of your data.
   - ✓ Model Development:
     - Build a predictive model and identify key features within the dataset.
     - Use statistical methods to argue the importance of one feature over another.
   - ✓ Model Comparison:
     - Compare the performance of different models, ensuring you have selected the right parameters (for example, through cross-validation).
     - Consider statistical hypothesis testing to compare model performances.
     - Establish simple models like logistic or linear regression as baselines for comparison.
     - Avoid data leakage: fit preprocessing steps on the training data only, and use splits suited to time-ordered or grouped observations.
   - ✓ Discovery Reporting:
     - Document and report on your findings and the insights gained from your analysis.

2. **Research-oriented Project**
   - ✓ Method Selection:
     - Choose one or more statistical learning methods that pique your interest.
   - ✓ In-depth Understanding:
     - Delve into the chosen method(s) by reviewing academic papers or studying the source code.
   - ✓ Method Development:
     - Build the method from the ground up or streamline existing code to grasp its core principles.
     - Apply your method to various datasets to test its efficacy.
   - ✓ Algorithm Comparison:

- If you are summarizing, compare your algorithm's performance on synthetic or real-world datasets.
- Document the outcomes of these comparisons.

✓ Algorithm Innovation:

- Once you have a solid understanding, consider developing a variation of the algorithm.
- For example, with ridge regression, experiment with different regularization strengths for different coefficients or constrain coefficients to approach a specific value rather than zero, and work out the solution.
- Tailor your variant to different application scenarios based on your understanding and research.

Grading policy and Deliverables (30%)

1. **Final presentation (15%)**
   - Schedule: December 24 (9:10 AM - 12:00 PM)
   - Format: Each team will deliver a 12-minute presentation, followed by a 3-minute Q&A session.
   - Criteria: Presentations will be assessed based on the problem and data novelty, clarity of delivery, alignment with course content, and technical depth of the project.
   - Scoring Breakdown: The final presentation score combines Teaching Assistant's grades (7.5%), and the instructor's grade (7.5%).
2. **Final Report, Slides, and Code (15%)**
   - Deadline: December 26, 11:59 PM
   - Report Structure: Your report should include the following sections:
     - Abstract
     - Introduction and Related Work
     - Dataset Description
     - Methods
     - Experiments and Results
     - Discussion
     - Conclusion and Future Work
     - Individual Contributions
     - References
   - Details: Even if your project results are inconclusive or negative, you should still report your process, explain what was attempted, and provide a thoughtful discussion. Follow the format outlined here. Your report **should not exceed 10 pages**, inclusive of appendices and figures. Use **A4 or 8.5 x 11-inch paper size with a minimum 10pt font.** Additional pages allowed for references only. Verify all AI-assisted results and references; unverified

numbers or fabricated citations are your responsibility.

- Acknowledgements and AI Use: If you received advice or assistance, you must fully recognize these contributions. In your report, briefly state which AI tools you used and for what (e.g., finding data, coding, writing), or state that none were used; conversation logs are not required. Also include any attribution your data source requires.
- Team Contributions: **Detail each member's specific contributions to the project**.
- Code Submission: Submit the code used to produce your results. The code must run from start to finish and reproduce the numbers and figures in your report; include brief run instructions, package versions (e.g., requirements.txt), and fixed random seeds. Python is preferred unless another language is more appropriate for the task. Provide either a link to your code repository or a zip file. Do not include external libraries or API keys in your submission. Include the data if it is small; otherwise provide the source, retrieval date, and a download link or script. Reproducing your results should not require an API key.
- Final Slides Submission: Submit the slide used for your final presentation along with your report. Your slides should also include a brief AI-use statement (one slide or a corner of a slide).
- Evaluation: The report will be assessed on problem and data novelty, clarity, course relevance, code correctness and reproducibility, and technical merit.
- Scoring: The report's grade will be the sum of the Teaching Assistant's (7.5%) and the instructor's (7.5%) evaluations.