CIS 5200, Machine Learning, Fall 2022
Final Project

November 1, 2022

1 Project Description

1.1 Overview

For the CIS 5200 final project, groups of **2-3 students** will use machine learning algorithms to address a ”real world” ML problem using real data. The project will typically involve understanding the problem and data involved and selecting and applying multiple machine learning algorithms appropriate to the problem. You may also wish to develop new ML techniques or variations of existing techniques. Many groups will be formed within the pod, for those doing a cross-pod project, please select one pod as the home pod for project.

The process will be:

1. **Find a problem and appropriate data**. You are free to pursue whatever application domain interests you. However, you must be able to find a suitable dataset, and the scope of your problem must be realistically solvable within the project time-frame (roughly five weeks). We recommend the use of public datasets; we have compiled a list as a starting point.
2. **Look up related research and ideas**. What similar problems have people tried to solve? What is the current state of the art? What kind of data are typically used in your application domain? What machine learning methods to uses? It is fine to pursue a topic that already has existing solutions; but mention them in your report, and explore variations on the published methods.
3. **Formulate the problem as a machine learning problem.** Looking across all of the techniques we have learned so far in the course, what would be most appropriate for your problem? Is it a supervised, semi-supervised, or unsupervised task? Is it regression, classification, or clustering? Are there unique properties of your data that require additional consideration or preprocessing?
4. **Choose which machine learning methods you will implement.** You should compare multiple methods for tackling your problem (typically 4 or 5), including a “baseline method”. For example, in a regression task, your baseline method could be ridge regression with comparisons to random forests and gradient tree boosting. You should have one method that is novel: a non-standard training loss, transfer learning, regularization, or such. Be sure to think about what properties are important for your problem, such as model representation, computation speed in training and testing, memory, etc. to justify your method choice.
5. **Decide how you will evaluate your methods.** What performance measures are most appropriate for your model? How will you ensure your models are not overfit to the data? Plan out your evaluation

pipeline before you start working with the data: e.g., set aside a test set that you will not look at until all models are built, use k-fold cross validation for model selection, consider class imbalance (in the case of classification), etc.

6. **Make a project plan.** With your data, problem formulation, modeling choices, and evaluation framework in hand, make a timeline of your planned work, with details on what each team member will do. Be sure to budget time for data processing and preparing your final report.

1.2 Logistics and Evaluation

There are no restrictions on packages that can be used, so long as proper attribution is made. Milestones of the project are as follows:

- Project proposal **due November 15th, 11:59 pm**
- Project checkpoint **due November 29th, 11:59 pm**
- Jupyter notebook and final report **due December 13th, 11:59 pm**

2 Project Deliverables [100 points]

2.1 Project proposal

[**5 points**] As the main purpose of this proposal is to receive feedback from the instructors on your project, you will get full credit on it if you give all the information listed below. Your proposal should be submitted as a pdf (see the provided project template) and have sections for:

- **Group members** and **Team name**
- **Home pod**
- **Motivation**: Briefly introduce the problem you are planning to work on
- **Data set** Briefly describe your proposed dataset (what is n, p, sets of features, ...) Provide a **link** to the data.
- **Related Work**: Include **at least one citation** (does not have to be a publication, can be a website or blog post) and one paragraph description of prior work related to your project.
- **Problem Formulation**: Describe how you will frame your problem as a machine learning task.
- **Methods**: List the methods you plan on using or improving upon.
- **Evaluation**: How do you intend to evaluate your methods? Be precise about the loss function.
- **Project plan**: Provide a rough timeline of your project work schedule, including which team members are responsible for what portions of the project.

Please only submit one proposal per group on Gradescope, making sure to add other group members to the submission.

2.2 Project checkpoint

[5 points] The project checkpoint should be a draft of your final project report only (no notebook), graded only for being done. Its goal is to encourage you to think about the report structure early. Note that it is okay for this to be incomplete (e.g. "we plan on discussing the performance comparison of model X with model Y..."), so long as you give a plan or brief outline of how you will write each section of the report. See the project report section below for more details.

2.3 Project report and notebook

The Jupyter notebook should be styled in the form of a brief blog post describing and visualizing some aspect of the data you are working with, and then showing how you implemented one of your machine learning algorithms.(this is probably also the basis for the presentation in your pod.)

The final report will be structured like a conference paper summarizing your results. **The reports should be range from 5 to 10 pages, including references. We will ignore any page beyond the page limit in your PDF (do not add a cover page). However, it doesn't mean you should make the content too small or layout too crowded. All the texts, plots and tables must be readable to the naked eye (i.e., without zooming in).** The grading structure is outlined below:

Jupyter notebook

- **[5 points] Data exploration**. Provide a brief overview of your data and at least one visualization exploring some aspect of it. (Note that this can overlap with your report.)
- **[5 points] Model walk-through**. Give a brief code walk-through of the implementation of one of your models, including aspects such as hyperparameter selection, visualization of train/test curves, and final performance. The intent is to provide a self-contained guide on how to implement that particular model for your data. (This, too can overlap with your report.)
- **[4 points] Code submission.** Submit both your notebook and your project source code via Gradescope.

Report Sections (5-10 pages, including references)

- **[3 points] Abstract.**
  - Give a brief 1 paragraph summary of your project and findings
- **[6 points] Motivation.**
  - Introduce and describe your problem
  - Discuss why you find your problem domain interesting and why you think machine learning is suitable to solve it
- **[6 points] Related work.**
  - Describe related work, pros/cons of their approaches
  - Include references to any work cited: references should be on a new page at the end of the report
- **[9 points] Dataset.**

  - Describe the dataset used (what is n? what is p?), including a link to the dataset source
  - Provide relevant summary statistics, and visualizations (if appropriate)
  - Describe any data pre-processing that was needed

- **[6 points] Problem Formulation.**
  - Describe how you framed your problem as a machine learning task
  - Include justification for your framing, such as your choice of loss function, model representation, feature engineering, etc.

- **[12 points] Methods.**
  - Describe your baseline method and why you chose it to be your baseline
  - Describe your other method(s) and why you chose them for your particular problem
  - Describe your implementation approach and cite any packages when appropriate

- **[12 points] Experiments and Results**
  - Describe your experimental/evaluation framework so that it is reproducible (e.g. hyperparameter initializations, optimization methods, etc.)
  - Describe your performance metrics and why you feel they are appropriate for your problem
  - Report in a single table a summary of your performance results across all of your models (e.g. train/test accuracy for classification)
  - When appropriate, include figures visualizing your evaluation process and results (e.g. training curves, AUC, final learned clusters, etc.)

- **[12 points] Conclusion and Discussion.**
  - Summarize your findings and any model comparison conclusions from your results
  - Discuss some qualitative interpretations of your results. Do your models reveal any insight to the problem you are exploring?
  - Discuss any lesson learned from the overall project process. What did you try that helped? What did you try that didn't help?
  - Identify potential opportunities for future research/extensions to your project

- **[10 points] Overall evaluation.** In addition to the criteria provided above, we will evaluate projects based on:
  - **Technical quality.** Is the problem well-formulated? Are the results sensible? Are there any major gaps in the authors' approach or understanding?
  - **Novelty.** Is this a new problem or a unique take on an existing problem? Did it involve some interesting data or solution approach?
  - **Clarity.** Is the report well-written? Are the results presented and/or visualized in a clear manner?