CS 4641: Machine Learning
Final Project: Experimental Design for Supervised Learning

Collaboration

All of the assignments in this class are **individual work only**. You're welcome to *discuss* the math or high level concepts behind the algorithms you're being asked to implement, but you are **not allowed to share code or answers to the written questions**. We will be checking, manually and with automated tools, for any violations of this policy, so please **do not share solutions or code**.

Introduction

This project is different from the homeworks you've worked on previously in the semester. There is no skeleton code file or list of questions to answer. You are welcome to use libraries which implement the algorithms you'll be applying. But the primary difference is that this project is much more open ended. The purpose is for you to apply what you've learned throughout this semester to a problem that interests you, and report on what you find. The remainder of this document outlines a basic report structure and some required components, but the bulk of the work will be in designing, running, and interpreting the results of experiments which compare the performance of several machine learning algorithms on a dataset of your choice. There is no "correct" result, because what you find will depend entirely on the problem you're trying to solve and the experiments you design. What we will be looking for in grading this assignment is whether you include everything we ask you to, whether your experimental methodology is well designed and described, and whether your conclusions are supported by the results of your experiments.

1 Description

Pick one dataset that is interesting from a ML perspective, and important to you. There should be a supervised learning signal available (e.g. class labels or regression targets), and the dataset should be *non-trivial*. Non-triviality is subjective, but here are some heuristics:

- There should be at least 1000 **datapoints**.
- There should be at least 5 **features** for each datapoint.
- The datapoints should have a non-trivial **distribution**. For example, linearly separable data is not very interesting.
- If in doubt, ask yourself, "Will the TAs agree that this interesting?"
- For the purposes of this assignment, datasets we have used in previous homeworks won't be considered non-trivial.

Using *any* implementation (yours, library, code off the internet) compare the performance of the following algorithms:

If you are doing regression:

- Kernel Ridge Regression with a non-linear kernel[1]
- k-Neighbors Regression [2]
- Neural Network with at least 2 hidden layers (preferably more)

If you are doing classification:

- Random Forests with Bagging[3] or Boosting[4]
- Support Vector Machines with a non-linear kernel
- Neural Network with at least 2 hidden layers (preferably more)

2 Analysis

Your report should be a PDF with a maximum of 15 pages or 5000 words (whichever is shorter). It should include the following sections.

Introduction to the dataset Explain what the data consist of, what the underlying supervised learning problem is, and why it's important to you. Explain the metric(s) that you will use to measure performance.

Description of the algorithms Briefly describe each of the algorithms you will be testing. Identify the hyperparameters that need to be tuned in the next section, and explain how they are related to the complexity of the hypothesis class.

Tuning hyperparameters Describe a set of experiments for tuning the relevant hyperparameters for each method using cross-validation (or another appropriate tuning technique). Include figures (graphs and/or tables) showing the performance of each algorithm in terms of the metric(s) introduced in the first section. Justify the claim that the data has a non-trivial distribution. Report the final hyperparameters that you chose, the performance the chosen hyperparameters achieve, and how much computation time was necessary to run each experiment.

Comparing algorithm performance Describe a set of experiments for comparing the performance of each tuned model with the others on a held-out test set (unseen during tuning). Be sure that your results include confidence intervals or other appropriate measures of statistical significance. Based on the results, draw a conclusion about which algorithm works best and why. Support your assertions with figures.

Conclusion Compare the algorithms based on metrics other than performance (e.g. training time, number of hyperparameters). Use this, along with performance, to make a claim about what algorithm you would use in practice on real world data from the same domain.

Acknowledgements List any external sources you used to help you with this assignment. This includes software libraries, other students, and websites. When in doubt, cite it!

Submission format

Please submit **one** zipfile in the format `final-GTusername.zip`, replacing “GTusername” with your username (for example “final-bhrolenok3.zip”). This zipfile should include:

- `report.pdf` A PDF with your written analysis. See “Analysis” above. To maintain some anonymity when grading, please do not put your name on the PDF itself.

[1] `https://scikit-learn.org/stable/modules/kernel_ridge.html`

[2] `https://scikit-learn.org/stable/modules/neighbors.html#regression`

[3] `https://scikit-learn.org/stable/modules/ensemble.html#forests-of-randomized-trees`

[4] `https://scikit-learn.org/stable/modules/ensemble.html#adaboost`

- README A text file that explains how to reproduce your experiments. See "Reproducibility" below.
- Anything else needed to reproduce your experiments (code, raw data, preprocessed data, etc.).

3 Reproducibility

Reproducibility is an important part of experimental design. Accordingly, the expectations for reproducibility for this project are higher than they were for the previous assignments.

Obtaining the data

If at all possible, please include your dataset(s) in your zip file. The Canvas file size limit is 500 MB, so try to choose datasets that are smaller than this. If uploading your dataset is not possible, please explain in your README how to obtain the data. Try to include a URL. If even this is not possible (e.g. your dataset is protected by privacy or intellectual property laws), please obtain prior permission from the instructor.

Re-running your experiments

Your README should also explain how, starting from the raw data and with only a handful of terminal commands, we can re-generate all of the graphs and tables in your report.pdf. In particular, it should explain

- how to install software dependencies
- what terminal commands to run to re-generate your graphs and figures.

Remember, this should require as input only the raw data (e.g. loading pre-trained regression parameters from a file is not allowed).

Appendix

Extra credit

At their own discretion the TAs may offer a few points of extra credit for exceptionally good reports. Some qualities that make a good report include:

- Conciseness: A short report will get *more* extra credit than a long one.
- Formatting: A well-formatted PDF will make the reader well-disposed toward you.

Examples of interesting datasets

You may use any dataset(s) that you like for this assignment, but here are some ideas to get you started.

- https://www.dataquest.io/blog/free-datasets-for-projects/
- https://www.cooldatasets.com/
- https://github.com/awesomedata/awesome-public-datasets