CPSC 483: Spring 2022

Class Project Instructions

CALIFORNIA STATE UNIVERSITY FULLERTON

Paper Due: May 01 at 11:59 PM on Canvas

For Graduate Students ● **Option 1: (Mandatory**) You are recommended to submit a paper for the class project. (Team Size 2) Sample paper
For Undergraduate Students ● **Option 1: (Recommended**) You are recommended to submit a paper for the class project. (Team Size 2-3) ● **Option 2: (Mandatory**) You can required to submit a research poster for the class project (Team Size 2) ● Please note you will either submit Option 1 **OR** Option 2. Sample poster

1. Description

Your class project counts for 25 % of your final grade. The goal of the Project is to give you experience in using real-world data sets and researching machine learning. You are expected to work in groups of 2. There are multiple ways to do this project:

- **Health Domain - Cancer (**Research Component Involved) You can choose a topic from the list provided below. You don't have to limit yourself to the exact project listed below, a similar idea would be just fine.
  - Machine learning applications in cancer prognosis and prediction: Reading
  - Feature selection and classification in breast cancer prediction using IoT and machine learning: Reading
  - Classification of Breast Cancer Risk Factors Using Several Resampling Approaches: Reading
  - Cancer Prognosis: Remove mammography data from BCSC data
  - Breast cancer type classification using machine learning: Reading
- Pick one of the ongoing contests on **kaggle.com**, download the data and develop the best model you can. The website will provide you with the training data set which will include the initial set of features, a framework for testing. You can also check out the leaderboard from time to time to compare your performance with other teams working on the same contest. Choosing this option

will let you focus on developing better features, expanding the training data, choosing appropriate models and tuning them, and building an ensemble from a collection of models.

- **New Domain**: Identify an interesting problem, collect data, design a

  feature representation, apply several machine learning algorithms (being careful not to train on test data), and analyze the results.
- **Algorithm Development:** Develop and evaluate a new machine learning algorithm, representation, regularizer, optimization method, etc. It is hard to do this well, since most of the easy and obvious ideas have been tried already.
- **More Ideas:**
  - Stanford ML projects
  - Stanford ML project ideas
  - Hilary Mason's data links
  - UCI ML repository
  - Kaggle ML competitions
  - scikit-learn
  - Weka

If you want help picking a project, feel free to ask me questions. It’s best if you already have some idea of what you want to do.

2. Methods and Results

For an application paper, you should evaluate and justify the choices

you made. Here are some questions to think about:

- How did you formulate your problem as a machine learning problem?
- How did you select your data? How much data do you have? What cleaning or pre-processing did you perform for the data? (For some problems, you can get creative about integrating data from multiple sources, or be able to handle noisy labels.)
- Is this problem best posed as classification? Regression? Clustering? Ranking? Probability estimation?
- What features did you select and why?
- What algorithms did you use? (You should almost always use more than one, in order to have a comparison.)
- What baselines did you use (if proposing a novel algorithm or feature set or problem formulation)?
- How did you set up the training/tuning/testing data? Did you do cross-validation? How did you tune the parameters?
- How do you choose to measure performance? Accuracy? Learning curves? ROC curves? Confusion matrix? Precision/recall/F1 measure? Running time?
- Which algorithm performs best? Can you determine why that algorithm works best?

You do not need to implement everything yourself. Scikit-learn and Weka are popular open-source toolkits that already include many common classifiers. Other popular open-source tools include LIBLINEAR (for linear models), LIBSVM (for SVMs), and Keras (for neural networks).

Please do follow the scientific method. Develop appropriate experiments to validate or refute your hypothesis, as well as to provide more insight. For example, which feature representation worked best? Which classifiers or combinations of classifiers worked best? Why do you think this is? What evidence do you have for this Explanation?

An accuracy with no explanation is not interesting. An explanation of how you obtained that accuracy, what worked and what didn't, and what you learned is more interesting. Please include some quantitative measures, in tables, charts, and graphs.

This work should demonstrate that you understand how to apply machine learning to a real problem (for an application paper) or how to develop and evaluate novel algorithms (for an algorithms paper).

Negative results are acceptable. If you get a negative result, explore what happened and why. Not enough data? Overfitting? Bad features? Noisy labels? Different distribution at test time? Explore what led to the poor results and try to determine if that could be overcome.

3. Writing

All papers are expected to be clearly written with a good structure. I will hold graduate students to a higher standard of formal, technical writing and analysis of experimental results. This project should be doable by a two-team person, so I expect that larger groups will have correspondingly more experiments and more analysis.

Many machine learning papers use an outline similar to the following outline:

1. **Abstract**: Summarize the entire paper (including results) in 50-250 words.
2. **Introduction**: Identify the problem you're trying to solve, describe why it's important, and outline the key method or strategy that you will use to solve it.
3. **Background**: Describe the technologies or ideas that you will build on your method. For an application paper, this could simply be a detailed description of the problem you're trying to solve. For an algorithm paper, this could be the machine learning methods that you're extending.
4. **Methods**: Describe your approach to solving the problem. This should contain your key contributions.
5. **Experiments**: Evaluate your approach experimentally. Describe your methods in enough detail that another researcher could replicate them. How well does your method work? Does your method outperform reasonable baselines? How does your method compare to simplified versions of your method? What kinds of errors remain? What interesting things do you learn from your experiments?

Tables of results are useful, but charts and figures are often better. This can also be integrated with the methods, so that each aspect of the model is evaluated as it is introduced (e.g., feature selection, classifier selection, ensemble construction).

6. **Conclusion**: Summarize your contributions and discuss future work (50-500 words).
7. **References**: Works that you cite in the body of your paper. You may use any standard citation style as long as it is consistent. I recommend that you use a structure similar to this one, unless you have a good reason.

I do not require perfect English, but I greatly appreciate clear writing. Your methods should be described clearly enough to replicate your results. Your conclusions should be supported by evidence. Your arguments should follow logically. Each paragraph should discuss a single idea. If you're having trouble, there is writing tutoring available on campus for all students.

Learning to write a good technical paper is an extremely valuable skill in both graduate school and industry. Writing well is very difficult, even for experienced writers, but it does get easier with practice.

4. Project Proposal (This is not required but recommended)

In order to give you early feedback on your ideas, please send a 1-page proposal to me, Neda and Nino as soon as you have an idea of what you want to do. Feedback will be given in the order that projects are received, so sooner is better.

You only need to send one proposal per group. A good proposal should describe the problem you are trying to solve and your ideas for how to solve it. What data will you use? What features will you use? What algorithms will you try? What metrics will you apply? For Kaggle problems, some of these questions are already answered by the problem specification. However, you will still need to consider ways to modify the training data or features, ways to combine different models into an ensemble, how you plan to do parameter tuning for all of the algorithms you want to apply, etc. (Simply applying standard machine learning algorithms to the default Kaggle representation is not an acceptable project.) The more details you provide, the better feedback we can give. If you submit a proposal, Neda, Nino and I will give you feedback that will be helpful as you work on your final project. For example, we may be able to help identify if the problem you're trying to solve is too difficult or doesn't really count as a machine learning problem. We may also be able to suggest alternate ideas and approaches, or relevant background reading that could help.

5. Grading

The paper will be graded on a 25-point scale.

Paper/Poster - 10 points. Paper/Poster should be clearly written and structured. Use tables, figures, and other visualizations as appropriate.

Description of methods and presentation of results should showcase the scientific method -- make it clear what you're evaluating, how you're evaluating it, and what the result is. Discussion and analysis of the results.

Methods - 10 points. Select appropriate feature representations, classifiers, ensembles, etc. For the most points, evaluate a variety of methods, select methods that are a good fit to the particular problem you're working on, and, if possible, include your own novel ideas about how to prepare the data or train the algorithms. Use proper experimental procedures for parameter tuning and classifier evaluation.

In-class Presentation - 5 points. Present your project to the class. Slides are recommended but not required, as long as you can clearly communicate your methods and your results.

Feel free to ask for help: We are happy to provide feedback on your project plans to help you succeed. For example, we may be able to suggest useful features, classifiers, or tuning methods.

Acknowledgements: Large portions of this document draw from the document authored by Professor Daniel Lowd.