MSAN 630
Advanced Machine Learning
Final Class Project

Summary
The purpose of the project is to give you the opportunity to work in a team and synthesize the skills, techniques, and models you have learned in MSAN 630 and MSAN 621 and work with a "real- world" data set and problem of your choice. In particular, the aim of this project is to encourage you to:

- Assess the machine learning technical skills and conceptual knowledge you have learned and choose the appropriate techniques to create a predictive or descriptive model.
- Apply your knowledge of the machine learning packages from Python and/or R to support your modeling process; and
- Present your model, reasoning, and conclusions in a short paper and presentation.

Project Rules. The rules for this project are (at least) the following:

- Unless we have arranged otherwise, groups will be assigned at random. Group assignments will be posted on Canvas.
- You should choose the type of machine learning task/problem (classification, regression, recommendation, …) and/or domain (text, images, financial) that you are interested in before choosing a dataset.
- You must use a dataset that you have identified, documented (if needed), and downloaded yourself. You are welcome to explore multiple datasets for a given problem.
- Both your problem and dataset must be approved by me; see milestones below.
- You will generally want to avoid most of the data sets on the UC Irvine site, though most non-tutorial Kaggle competitions are fair game (and some of the harder tutorials are good too).
- Your project must deploy in **some way** techniques from machine learning; in other words, it should involve the experimental application and evaluation of at least one algorithm from the ML literature to at least one dataset. Sample projects are given below. You are not restricted to algorithms we have covered in class.
- The presentation will be on the last day of class, parameters to be specified.
- The paper must be the equivalent of 6-8 pages long, single-spaced, including space for pictures, charts, tables, and diagrams—but not for appendices.
- The paper must be typeset in LaTeX.
- You must ultimately turn over your raw (and transformed or cleaned) data to me in case I decide to replicate all or part of your analysis. (Email me a compressed file with all materials related to your analysis, including the scripts, SAS files, raw data files, transformed data files, the paper, etc., before you consider yourself done with your project, or email me a pointer to your github repo containing same)

About working in teams: in order to try to fairly grade the teams, each team member will provide a "peer assessment" for the members of the team, including yourself. These will be entirely confidential.

Timeline –

- Project proposal – Wednesday, Feb 18 (details on Canvas)
- Progress report (details coming)
- Final project presentation (details coming) – Friday, March 13
- Final project report and supporting code (details coming) – Sunday, March 15

Detailed project guidelines and resources
You will be going through the following steps:

- deciding on what you want to achieve with machine learning
- identifying one or more data sets and problem domain
- choosing appropriate methods and algorithms
- implementing and testing your methods
- evaluating your techniques on your data sets
- reporting conclusions in a paper and in-class presentation

Two types of projects will be worth an "A" if done right (in both cases you **must** also evaluate what you do on some data!):

1. Design a novel machine learning technique (could be an adaptation of one you know or read about)
2. Implement a non-trivial machine learning technique that was not implemented by you in this class or a prior one

Here are some other typical types of projects; the more ambitious, the better!

- Focus on the data and a particular task (e.g., prediction, clustering) and use multiple different algorithms from the literature to address this task. The focus here might be more on the data set and the task, than on the algorithms.
- Focus on the algorithms: compare one or more baseline algorithms with a new recently-published algorithm - the focus here might be to see if the new algorithm really works as well as described in the paper where it is published
- Develop a new algorithm/method and implement it and apply it to a task and data set – it's important here to evaluate how your approach compares to existing baseline methods (at least one).
- Another example would be to take a well-known technique (such as Naïve Bayes or hierarchical clustering) and carefully evaluate and compare a variety of different ways for fitting such models to data, evaluating perhaps on multiple data sets rather than just on one.
- Select one or more algorithms for detailed study. Test the algorithms on various datasets. Try to determine their relative strengths and weaknesses, and how the algorithms perform under varying conditions, such as a varying number of features. Come up with ways of extending or combining the algorithms, and test these ideas experimentally. The goal of such a project might be to find an "off-the-shelf" algorithm giving the best performance on a range of datasets (where performance is measured in one of the ways discussed in class, such as test accuracy). Alternatively, a similar but different kind of project might focus on computational issues and how to make one particular algorithm as fast as possible without sacrificing performance.
- Study a particular application domain, such as classification of visual images, clustering of email messages, or automatic recommendation of movie titles. Consider a number of algorithms for your problem, and determine which seems to perform the best. Or, design and test a learning model that is especially appropriate to your problem. Think about the issues that are most relevant to your specific problem; for instance, what features are most appropriate, what independence assumptions are reasonable, and how do the algorithms you are using fit or not fit this problem?

Note that a critically important part of any project is that you conduct careful empirical evaluations (for example by using appropriate train/test partitions) of different approaches. Also note that your project grade will not necessarily be correlated with the accuracy of your results: for example, a project developing a classification algorithm that achieves high accuracy on some classification task could receive a low score if you do not provide any insight into why the algorithm did well. Conversely, a project on classification with relatively low accuracies could get a very high score if you can provide insight and understanding into why the accuracies are not high.

Grading Criteria – This project, including the paper & presentation, is worth 35% of your grade in the class. Here is how I will break down that 35% (percent of the 35%!)

- (5%) Proposal
- (10%) Progress report (short written update)
- (3%) Appropriateness & Ambitiousness of project: did you choose an interesting project and dataset, set reasonable goals, choose an appropriate (set of) algorithm(s) to apply to the data, and do something cool with it all, leveraging the skills you've learned in and outside of class?
- (5%) EDA: Smart data acquisition, exploration, & preprocessing
- (20%) Implementation: implementing substantial & correct algorithmic code, or working with an algorithm or tool kit that is difficult to apply; complexity of evaluative code
- (12%) Experimental results: using proper evaluation criteria and methodology such as evaluating on data other than the training set, using cross-validation and/or a validation set for parameter setting, etc.
- (25%) Paper: Did you write a clearly, well-organized, and complete paper?
- (20%) Presentation: Did you explain your project in a way that your classmates could understand? Did you show what you did using good displays of your data, algorithm outlines, and/or experimental results?

Other Project ideas:

- This list of ideas from a Stanford AI class has a lot of projects related to Data Mining: http://stanford.edu/~cpiech/cs221/homework/finalProject.html#ideas
- The ACM SIGKDD has a ``KDD Cup'' competition each year which might have problems & data of interest: http://www.kdnuggets.com/datasets/kddcup.html
- Competitions in Kaggle: https://www.kaggle.com/
- Text-analysis projects
    - The CONLL Conference has a ``shared task'' each year; check out the list here if you want to explore any
- Sentiment analysis datasets
    - http://www.cs.cornell.edu/People/pabo/movie-review-data/
    - http://mpqa.cs.pitt.edu/ (I have some code to read the opinion corpus & subjectivity lexicon; you may use it if you want to work with this data)

Sources of data that could be the seed of ideas

- http://rs.io/2014/05/29/list-of-data-sets.html
- http://www.kdnuggets.com/datasets/index.html
- http://www.statsci.org/datasets.html (which of course lists Kaggle!)
- http://www.quora.com/Where-can-I-find-large-datasets-open-to-the-public

Hints

- These slides (by Andrew Ng at Stanford) may help you as you start to get results for your chosen project: http://cs229.stanford.edu/materials/ML-advice.pdf