CS-UY 4563 Project

The CS-UY 4563 project is an *applied machine learning* project. Students will:

1. Find or collect a data set.
2. Ask a question (or two) about the data set which can possibly be answered with machine learning.
3. Apply tools and techniques learned in the class to answering that question.

Basic Info

- Work in groups of **two**.
- 20% of final course grade.
- Deliverables: Project proposal (2 pages), project report (4 pages, or longer if needed), presentation to class (5 minutes), GitHub page with all code and documentation.
- Any data set or topic is allowed. You can use publically available data or data you collect yourself (e.g. via web-scraping). However, you should not reproduce an analysis that has already been done! Even if the question you ask has been asked before, you should take a new approach to solving it.

Important Dates + Deliverables

Start looking for data and topics early!!

4/1, Project Topic Chosen. Email cmusco@nyu.edu with group members, proposed project topic, and any preliminary ideas on data sources.

4/2, 4/6-4/8, Meeting Held. Schedule mandatory 10-15 min meeting with me to discuss project, possible data sources, and possible approaches. Please sign-up for a slot on this spreadsheet.

4/13, Project Proposal Due. 2 page proposal containing:

- Description of proposed project, discussion of questions you seek to answer, description of potential approach and methods you will try. Discussion of any prior work on the dataset or question.
- How will you evaluate the performance of any methods you try? Loss, error rate, precision, recall, reconstruction error?
- Discussion of baselines that will be considered (linear regression, k-nearest neighbor classification, outputing most popular class label, etc.)
- List of all datasets that will be used, including links.
- References to any relevant papers, libraries, repository, etc. that will be important for your project.

5/11, Project Report Due. 4 page report containing:

- Description of project and questions addressed (these will likely evolve over the course of the project). Discussion of what methods were used, how they were used, and what motivated your choices. You will likely discuss things like data preprocessing, feature selection, feature extraction and transformation, regularization, model optimization, model selection and cross validation, etc.
- Discussion of final model performance, including a full comparison to simple baseline methods. You will likely want to include plots, tables, or other figures, but do so judiciously. We don't need to see every experiment run.
- A conclusion describing what you would have done with more time: are there modifications of your approach worth trying? Other questions to address? Different data sets or features?

Finding Data

- There are *tons* of free data sets available online. This KDnuggets page is a wonderful starting point with lots of links and inspiration.
- In you are interested in *social network data* take a look at the Stanford SNAP project.
- Lots of interesting data can be scraped from the internet -- e.g. financial data, text data, images, etc. If you go in the direction of collecting your own data, *make sure you have enough time to invest*. We are grading you on what you do with the data, not how you collect it.
- ... more to be added

Tips

- Choose a data set and that interests you. It will be more fun that way. Center your project around a question or two that you think would be interesting to answer using data. Talk to me or a TA early if your team is struggling!
- **Look at the data.** Look at the data in table form, plot different features against each other, run PCA and plot, etc. Spend time getting to know your data set before running it through any ML algorithms. You will notice lots of things: missing data, outliers, features which are clearly useless, errors in how the data was imported/processed, etc. Re-examine data after doing any preprocessing, feature extraction, cleaning, etc.
- **Start simple.** Try the simplest methods we know before moving onto more complex approaches. Simple linear regression, logistic regression, naive bayes, or k-nearest neighbors for supervised problems, PCA for dimensionality reduction, etc. These methods will give you baseline results to improve on.
- **Start small.** Large datasets are slow to process! When writing and debugging code, test on a small subset of your data. This goes for everything from preprocessing, to initial model implementation. Don't run on the full data set until you are sure your code is working and bug free.
- **Sanity Check.** Is the error you achieved for a regression problem better than what would have been acheived by predicting every value to be the mean of the training data? Is your classification error better than what you would get from predicting 0 every time?