Machine Learning for Data Science (CS 4786)

Competition 1

Due on 5th May, 2016

The first in-class Kaggle competition for our class puts us in the exciting world of on-line entrepreneurship. We have a large set of projects from a number of different categories. Approximately half of these projects were successfully funded, while others failed to reach their goal. The information we were able to collect about these projects is: The backers network: a large bipartite graph indicating which donors gave money in support of which projects. (graph.csv) The project description: A sparse, censored representation based on the text from the project's description page. (description.csv)

The social media response and project status over time: A dense representation that contains information about the attention the project got on Twitter over time, as well as the project's growth. (social_and_evolution.csv)

Examples of 3 successful and 3 unsuccessful projects for Task 1. These are identified with their 0-indexed ids corresponding to the rows in the matrices above. Examples of 3 projects in each category for Task 2. These are identified with their 0-indexed ids corresponding to the rows in the matrices above.

- Task 1: (Kaggle: `https://inclass.kaggle.com/c/competition-1-cs4786sp16`) Predict which projects were successful and which were not. Submit this as a CSV of 1829 index and binary (0 or 1) label pairs, one for each project. A label of 1 means that you predict the project was successfully funded, while a label of 0 means the project failed to raise the funds needed. The first column is the index (starting from 0) and the second column is the predicted label. Example submission: success_test.csv
- Task 2: (Kaggle: `https://inclass.kaggle.com/c/competition-1-cs4786sp16-task-2`) Predict the category of each project. Submit this as a CSV of 1829 index and category label (0-12) pairs, one for each project. Example submission: category_test.csv

Collaboration and academic integrity policy

Students may discuss and exchange ideas with students not in their group, but only at the conceptual level. We distinguish between ?merely? violating the rules for a given assignment and violating academic integrity. To violate the latter is to commit fraud by claiming credit for someone else?s work. For this assignment, an example of the former would be getting

detailed feedback on your approach from person X who is not in your group but stating in your homework that X was the source of that particular answer. You would cross the line into fraud if you did not mention X. The worst-case outcome for the former is a grade penalty; the worst-case scenario in the latter is academic-integrity hearing procedures.

The way to avoid violating academic integrity is to always document any portions of work you submit that are due to or influenced by other sources, even if those sources weren?t permitted by the rules.

Deliverables Instructions

The competition has been hosted on Kaggle. At the end of the competition you will be required to submit two data files described in the two challenges as a zip files and the code for your best-performing submission. More importantly you need to also submit a writeup/report (?writeup.pdf?) of the things you tried and why you tried them (irrespective of whether they worked or failed). The writeup will count for at least as much of the grade as the empirical results of the final cluster assignments you submit.

Here are a few other remarks:

1. The report must be at least 5 pages long and not more than 15 pages, but with font sizes, spacing, etc., we just expect you to do something reasonable. Include all of your names, netids, and the name of your Kaggle team in your writeup.

2. Include visualizations of both successful and unsuccessful trials and tell us how these visualizations helped make your design choices.

3. Make a note of all successful and unsuccessful methods you tried. Explain why you made the choices you made and why you expected them to work. Take a shot at explaining why the less successful ones were in fact not so successful.

4. Organize your writeup into sections where each section (and its title) corresponds to a particular method.

5. You are certainly encouraged to try methods not covered in class or develop your own methods for the problem! If you use methods other than ones covered in class, do compare the performance both empirically and conceptually with (a reasonable choice of) methods covered in class.

6. If you turn in a draft a week before the deadline we will provide some comments as feedback on your report.

7. The zipfile of code you submit on the final (not the preliminary) CMS "Assignment" needs to include a README.txt file that explains how we can run your code. Include

in the README the exact values of any parameters you set to achieve your best result. The code should be "standalone" in the sense you should include any extra modules, libraries, etc. that you wouldn't expect our standard installs of R, Matlab, python, numpy, etc. to necessarily have.

8. Include in your writeup the values of any parameters you set for your best results. These values should also be in your README file.