Data Mining & Statistical Learning

Statistics 640 / 444 • Fall 2015 • Competition Description, Rules & Grading

Timeline:

Contest Opens	12:00am September 8
Progress Report I	5:15pm October 1
Progress Report II	5:15pm November 3
Contest Closes	5:59pm November 30
In-Class Contest Presentations	December 1 & 3
Final Report Due	5:00pm December 4

Competition Website:
`https://inclass.kaggle.com/c/rice-stat-640-444`
(Note: You must create an account on Kaggle using your @rice.edu email to access the competition.)

Contest Description:

For many users, the idea of sorting through hundreds of thousands of online dating profiles to find potential matches seems daunting. Instead, it would be great to have an automated system that recommends profiles of other users that a user will like. One way to accomplish this is to build a recommendation system that predicts the profiles a user is likely to enjoy based upon the user's past ratings of other profiles. To build such a recommender system, we will be working with a small subset of profile rating data from the Czech dating site `http://libimseti.cz/`.

The training data you are given consists of 3,279,759 ratings of 10,000 profiles by 10,000 users, yielding about 97% missing ratings. The ratings are integers between 1 and 10 with 10 being the best. Each user in the data set has rated at least 25 profiles. Your objective is to use the available ratings for each user to predict the missing ratings, thus recommending dating profiles that the user will enjoy. This type of problem goes by many names in different fields: "recommender system", "collaborative filtering", "matrix completion", or "missing data imputation". Beyond the ratings, you are also given the gender of each user. No additional information may be used to build your predictive models. Further information on the data and its format are available from the website in the README file.

Numerical evaluation for the competition will be based on the root-mean-squared-error (RMSE) between your predicted profile ratings and a subset of true ratings. The true ratings used for evaluation purposes are observed ratings that have been strategically deleted (by a secret algorithm!) from the training ratings matrix you are given. Typically, recommender systems fill in ALL of the missing ratings, and you can certainly build a model to do so. To ensure that the computation and upload time is manageable, however, we will focus on predicting a small subset of 500,000 of the missing ratings. More details on this subset are given in the README file available from the website.

Additionally, evaluation for the public (query set) and private (test set) leaderboards will be done using separate (secret!) subsets of the 500,000 ratings you are to predict. The public leaderboard corresponding to results on the query set will be made available in real time throughout the competition. The winner of the competition will be determined as the team with the best RMSE on the private leaderboard, which will remain hidden throughout the competition and only revealed at the competition's conclusion. Thus, if you overfit to the query set used to determine the public leaderboard, you may perform poorly on the test solution set.

Data for this competition has been made available from Vaclav Petricek: Lukas Brozovsky and Vaclav Petricek, "Recommender System for Online Dating Service", In Proceedings of Conference Znalosti, 2007. Thank you!

Contest Rules:

1. Students may work as individuals or in pairs.
2. Individuals may merge into a team up until October 1, 2015.
3. If a team consists of one student taking Stat 640 and the other taking Stat 444, the team will be graded according to the rubric for Stat 640.
4. A maximum of 2 submissions is permitted per day.
5. No outside information related to the data or other data is permitted.
6. You may use methods and software published by others. If other software packages are used, they must be free and publicly available. All outside methods and software used must be properly cited in the final report.
7. You must submit the code used to produce your final entry to the TAs as an executable script or file. Code will be checked for hard-coding and to ensure that it produces the results of the final entry.

Grading Overview:

Performance in Competition	25%
Innovation (640) or Interesting Findings (444)	25%
Learning Algorithms	25%
Progress & Final Reports	25%

The winning teams, defined as the teams with the best accuracy on the private leaderboard, one each from 640 and 444, will automatically receive an A+. The teams with the most innovative solution (640) or with the most interesting finding (444) will automatically receive an A+.

Requirements for Progress Reports:
Each progress report should be a one page document for each team summarizing and reflecting upon your progress thus far in the competition. The report should contain one graphic or table that summarizes your results. This includes both your internal training and predicted test error as well as the error rate for each of your submissions. (Thus, you should keep track of the performance of each base learner you fit and each submission you make.) The progress report should also address the following questions: What learning algorithms have you tried? Reflect upon the performance of the algorithms. Which ones worked well? Which performed poorly? Why? Have you innovated (640)? If so, how? Have you found anything interesting? If so what, and what methods did you use to find this? What are the future directions in which you would like to go?

Requirements for Final Report:
One final competition report per team (not to exceed 6 pages for 444 or 8 pages for 640) should summarize your experience throughout the competition and reflect upon what you learned. The final report should contain the following sections:

1. Overview. A paragraph overview of your approach to the competition and overall performance.
2. Base Learners. A brief summary of all the base learners tried and how they performed. Reflect upon their performance. How did you tune each method? How did you assess the error rate of each method? What can you conclude from your results?
3. Ensembles. A brief summary of the ensemble building methods tried and how they performed. Reflect upon their performance. What can you conclude from your results?
4. Model Selection & Assessment. How did you internally assess your training and testing error? Were your error rates optimistic or pessimistic for the leaderboard? Were your error rates different between the private and public test sets? Reflect upon your findings.

5. Best Scoring Model. Describe in detail your best performing method in terms of prediction error.

6. Innovation (640 only). How did you innovate? Be sure to place your innovation in the context of the statistics and machine learning literature.

7. Interesting Findings. Did you find anything interesting? What method did you use to find this? Reflect upon your findings.

8. What I / We Learned. What did you learn from the competition? What was the most challenging aspect? Was there anything unexpected? What would you do differently in the future?

Additionally, each report should contain the following:

- At least one graphic which summarizes your progress throughout the competition.
- Team acknowledgments. Acknowledge and clearly delineate the specific contributions of each team member.
- References or websites for publicly available software used.
- (640 only) Literature Cited. You should place your best model, innovations, and interesting findings in the context of the statistics and machine learning literature.

Code for Final Entry:
An electronic copy of the code (as an executable file or script) used to produce your best performing entry should be e-mailed to the TAs by **5pm on December 4, 2015**.

Electronic Copy of Competition Progress:
An electronic copy detailing your progress throughout the competition should be e-mailed to the TAs by 5pm on December 4, 2015. This file should be entitled "[your-team-name]_progress_stat640-444-2015.csv" and should contain the following columns: (Date, Training.Error, Test.Error, Leaderboard.Error), entitled precisely as specified. The date should be in the following format: (MM-DD), and should reflect the date on which the entry was submitted. The training and test error should be from your internal calculations to ensure there is no overfitting. The rows of this matrix should be for each individual method tried, not simply the ones submitted as entries to Kaggle. Please leave the Leaderboard.Error column blank for entries that you did not submit.

In-Class Competition Presentations:
In class on December 1 and 3, each team will be given 4 minutes to present their best performing method, most interesting finding, and/or innovative or creative solution. After each presentation, students will rate how interesting / innovative the finding or solution is. The ratings will be used in part to determine the two teams, one from 444 and one from 640, with the most interesting finding or innovative solutions.

Competition Grading:

Criterion	% Grade	Developing ($\leq$ B-)	Progress (B - B+)	Competent (A- - A)	Exemplary (A+)
Performance in Competition	25%	Bottom 25%	25% - 50%	50% - 90%	top 10%
Innovation (640 only)	15%	Used standard, black-box methods and techniques.	Trivial innovations and straightforward extensions.	Innovation that make an impact on overall performance.	Publication worthy innovation.
Interesting Findings	10% (640) or 25% (444)	Trivial inference and observations.	Findings that could be found by trivial analysis or provide simple insight into the data.	Findings that are interesting and provide new insight into the data.	Publication worthy findings.
Base Learners	10%	Limited set of base learners fit.	Fit several base learners but did not go beyond black-box methods.	Fit several base learners and went beyond black-box methods.	Fit several base learners in a way that substantially improved performance.
Model Selection & Assessment	10%	Major overfitting. Limited model tuning. Naive model assessment.	Some Overfitting. Models not well tuned. Flaws in algorithm assessment.	Minor Overfitting. Minor flaws in assessment. Well tuned models.	No Overfitting. Well tuned models. Correct model assessment procedures.
Ensemble Building	5%	Used black-box ensemble methods.	Naive ensemble assembly methods. Flaws in ensemble weighting schemes.	Solid ensemble building. Correct weighting schemes.	Innovative ensemble building. Weighting schemes that improved performance.
Progress, Final & Oral Reports	25%	Unclear graphics. Grammar and spelling mistakes. No reflection. Organizational flaws.	Limited or trivial reflection on performance of methods. Organizational flaws. Content unclear. Minor grammar mistakes; no spelling errors.	Good reflection on methods. Content mostly clear. Good graphics. Clear organization. Grammar and spelling correct.	Articulate, compelling and insightful. Superb reflection on methods. Well organized. Convincing graphics.

Examples of Innovation: Developed new statistical learning methods or algorithms; Extended existing methods or algorithms to improve performance; Built an ensemble in a novel manner; Combined methods or algorithms in a new way; Developed new computational strategies for fitting algorithms; Developed novel processing techniques that improved performance.

Examples of Interesting Findings: Found interesting associations between users and/or profiles; Found interesting groups of users and/or profiles; Found unusual aspects of the data or interesting outliers; Novel insight into the data that improved predictive performance.

Research Computing Accounts on DAVinCI:
To support the computing needs of this competition, each student registered for the course has been given an account on DAVinCI which runs a Linux operating system. Information on how to log in to DAVinCI and help pages can be found here: `http://www.rcsg.rice.edu/davinci/`. If you are auditing the class and would like an account on DAVinCI, please contact the TAs. Thank you Rice Research Computing Support Group!

Thank you Kaggle In Class!