C S 488/508 Introduction to Data Mining
Project: utilize data mining techniques to solve a real problem

1 Objective

In this project, you are required to perform meaningful data mining analysis in real-world applications. Through this project, you should be able to achieve the following objectives:

- Identify data mining problems by exploring different applications,
- Analyze data to get meaningful knowledge by using data mining techniques,
- Understand the steps to perform data analysis tasks using data mining techniques.

This is a **group project**. See below for group requirements.

2 Requirements

2.1 General requirements

- Group requirements
  - You should form a group consisting of 2 or 3 people to work on this part of the project (i.e., the maximum number of students in a group is three).
  - Each group just needs to submit one copy of your program. Points will be deducted if multiple copies are submitted.
- Design an interesting problem that comes from real-world applications and design solutions to solve the problem. The problem can be totally new (from your own investigation) or from other sources (e.g., a kaggle competition problem, a problem defined in a recent research article). The problem can also be part of a problem that you are currently working on (e.g., towards your PhD dissertation, or Master's project, or Master's thesis).
  (1) **Motivation**: Clearly describe the applications/motivations of the problem. If there are related works on this, briefly explain the related works.
  (2) **Problem**: Clearly define the problem and explain the specific analysis tasks.
  (3) **Solution**: Design reasonable solutions to solve the problem by utilizing the techniques that you have learned and making use of other related tools.
  (4) **Data**: Obtain proper datasets to test your solution. You can use self-created datasets or publicly available datasets. If you utilize existing datasets, they had better be reasonably big in size (e.g., with more than 10K instances). If you create your own datasets (e.g., by writing script to crawl data), your dataset may not be huge in size (e.g., with several hundreds of instances).
  (5) **Analysis**: Properly analyze the performance of your solution.

3 Detailed instructions

Your project will be submitted in five stages.

- **Stage 1** (3 points): Form a team. Create a `team.txt` file with student names. Submit this file.
- **Stage 2** (10 points): Formulate an interesting problem from real world with a strong motivation. Figure out what data mining tasks you may want to perform. Put your motivations, problem definitions and possible data mining tasks to your report. If there are related works on this, briefly explain the related works. This partial report should be about 1 page. Note that this report can be updated in later stages.
  Create team work files (for the team and for each individual).

- (1 point) The whole team needs to update the `team.txt` to write down the work allocation of the different team members in this stage.
- (1 point) Each team member needs to create a `PeerEvaluation_<yourname>.txt` to include a peer evaluation to your team members. (Details about this file see Section 3.1)

Submission:

- For the team, one student as a team representative needs to submit the report and the `team.txt` file.
- Each student needs to submit `PeerEvaluation_<yourname>.txt`.

Grading:

(1) (4 points) motivations and related works
(2) (4 points) problem definition and specific analysis tasks
(3) (1 point) `team.txt`
(4) (1 point) `PeerEvaluation_<yourname>.txt`

- **Stage 3** (10 points): Collect large data sets that can be used to test your problem solution. You can use existing datasets or write code to collect new datasets. You may want to provide analysis about the datasets, e.g., providing statistics about the data, plotting information about the datasets, and argue why the datasets are reasonable to conduct the analysis. You may want to collect 2 or more datasets so that you can thoroughly test your code in later tasks. Add descriptions about the datasets to your report. Add your proposed solutions to your report, continue to refine your motivations and problem definition, and update your report; Start to write code to do basic analysis. Update team work files.
  - The whole team needs to update `team.txt` to write down the work allocation of the different team members in this stage.
  - Each team member needs to update `PeerEvaluation_<yourname>.txt`.

  Submission:

  - For the whole team, one student as a team representative needs to submit the updated report, code, the links to your datasets, and the `team.txt` file.
  - Each student needs to submit `PeerEvaluation_<yourname>.txt`.

  Grading:

  (1) (1 point) updated problem definitions and possible data mining tasks
  (2) (5 points) datasets and code to analyze the datasets
  (3) (2 point) initial solution
  (4) (1 point) `team.txt`
  (5) (1 point) `PeerEvaluation_<yourname>.txt`

- **Stage 4** (10 points): Implement and improve your solutions (extensive programming!). Conduct experiments to examine your solutions. Your initial code from the previous stage may not generate good solutions. In this stage, you can explore other options to write other code, or improve your code to generate solutions with better performance. You need to test your code using your collected data. You also need to analyze your results and make improvements to your solutions if the results are not good. Continue to refine your report; finish at least half of the code (approximately); have some preliminary results from at least one dataset. Update team work files.
  - The whole team needs to update `team.txt` to write down the work allocation of the different team members in this stage.
  - Each team member needs to update `PeerEvaluation_<yourname>.txt`.

  Submission:

  - For the whole team, one student as a team representative needs to submit the updated report, code, and links to your datasets, and the `team.txt` file.
  - Each student needs to submit `PeerEvaluation_<yourname>.txt`.

Grading:

(1) (1 point) updated problem definitions and possible data mining tasks
(2) (2 points) updated datasets and code to analyze the datasets
(3) (5 point) updated solution and solution code
(4) (1 point) `team.txt`
(5) (1 point) `PeerEvaluation_<yourname>.txt`

- **Stage 5** (67 points): Finish your solution and your algorithm analysis; finish your report. Your report should explain all your work. The report should have an extensive analysis of the results. The analysis should consist of both effectiveness and efficiency.
  Submission and grading:
  (1) (62 points) For the whole team, a student as a team representative needs to submit the following content:
    (a) (3 points) A README file `readme.txt` with (i) how the code base is organized, (ii) the commands to run your code, (iii) names and versions of all the packages/libraries, and (iv) data set information: if you use existing datasets to test your solutions, the readme file should include the links to the existing datasets.
    (b) (40 points) Code base containing your code for crawling the data (if there is any), preprocessing the data, solving the problem, analyzing your solutions, plotting analysis figures, etc. The data had better come from two sources.
    (c) (17 points) A 3-5 page report, `report.pdf`, to include the above content (motivations, problem definition, solution explanation, data description, result analysis).
      (i) (5 points) Motivations and problem definition: clearly define a meaningful problem. The scores you get depend on how meaningful and hard the problem is.
      (ii) (5 points) Solution description: describe the methods you explored and implemented. There is no need to explain all the theory behind the methods if we already discussed those in our lectures.
      (iii) (5 points) Analysis of experimental results: your analysis should contain (1) the statistics of your datasets, (2) how you get your datasets, (3) experimental setting, (4) the effectiveness of your solution, and (5) the efficiency (running time) of your solution.
      (iv) (2 points) English (correct grammar, logical sentences, etc.) (Deduct 0.5 points for each error, maximum deduction 5 points);
    (d) (2 points) The `team.txt` to write down the work allocation of the different team members.
  (2) (5 points) Each student needs to submit `PeerEvaluation_<yourname>.txt`.

3.1 Instructions on peer evaluation

Your peer evaluation (`PeerEvaluation_<yourname>.txt`) should include an overall score (1 to 5, with 1 being poorest and 5 being best) and a justification for your score. In the justification, you may want to comment on the following several aspects.

- Communication effectiveness (whether your team members attend meetings regularly, and reply emails or direct messages promptly)
- Work effectiveness (whether your team members put effort to finish his/her allocated work on time; whether your team members contribute significantly in team discussions)
- Attitude (whether your team members treat other people in the team professionally, and have a cooperative and supportive attitude)

This file can be as short as 1 short paragraph or as long as one page.

3.2 Code requirements

- Your Python code should be written for **Python version 3.5.2 or higher**.
- Please write proper **comments** in your code to help the instructor and teaching assistants to understand it.
- You can use any libraries/packages offered in a language to conduct the above work.
- Please properly organize your Python code (e.g., create proper classes, modules).

3.3 Format requirements of the report

- For LaTeX users: please use `sample-sigconf.tex` from the ACM article template. Additional information about formatting and style files is available online at `https://www.acm.org/publications/proceedings-template`.
- For word users: Margin at each of the top, bottom, left, and right sides is 1.0 inch; double columns; The font size is 10pt; font type is Times New Roman. Single line space.

4 Grading criteria

- **ZERO point will be given if your code does not work. Please do not submit code that you did not test and make sure it works.**
- The score allocation is put beside each stage.
- You score in each project stage will be **weighted/adjusted** based on the work allocation and peer evaluation results.
- If one stage is submitted late, the late penalty (details see course syllabus) will be applied to the portion of that stage.
- For Stages 1-4, you are given feedback about your project. You need to address those feedback comments in a later stage.