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

1 Objective

In this group project, you are required to perform meaningful data mining analysis. Through this project, you should be able to achieve the following objectives:

- Discover 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.

2 Requirements

2.1 General 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.
- You need to finish the tasks stated in three stages. NOTE that the due dates of the three stages are different (see *Section 3: Submissions* for the submission details for different stages).

2.2 Detailed instructions

- Task 1: Define an interesting data mining problem from real world. Figure out what data mining tasks you may want to perform.
- Task 2: 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.
- Task 3: Design and implement (using Python, or R, or Java, or C++) your solutions. You need to generate workable code to generate some solution, which may not show the best performance.
- Task 4: Conduct experiments and improve your solutions. Your code from Task 3 may not generate good solutions. In this task, 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.
- Task 5: Write a report to explain all your work. The report should have an analysis of the results. The analysis should consist of both effectiveness and efficiency. (The detailed requirements of the report see below).

You can use any libraries/packages offered in a language to conduct the above work.

2.3 Report format requirements

- The report can be 4-10 pages;
- The page size is A4;
- The margin at each of the top, bottom, left, and right sides is 1.0 inch;

- The font size is 11pt; font type is Times New Roman; and
- The line space is single line.

3 Submissions

- Stage 1 (proposal): submit the partially completed report for Task 1 with (i) good motivations, (ii) problem definition, and (iii) possible data mining tasks. This partial report should be about 1 page.
- Stage 2 (mid-term): submit the partially completed report and code with information for Tasks 1, 2, and 3. In particular, you need to submit (i) the source code to collect data and (ii) the datasets. You need to include solution source code (which do not need to be completed). The report can be updated with the new information.
- Stage 3 (final): submit the fully completed report and the code with information for all the tasks. In the report, you need to report experimental setting and the results and provide informative analysis of he results. You need to update your code for Tasks 2 and 3. You can also update your datasets.

4 Grading criteria

- (40 pts) Source code and datasets
  - (10 pts) Create or choose at least two proper datasets.
  - (30 pts) Correctly implemented algorithms. In particular, your algorithms should be able to be run in any Computer Science server (i.e., openSUSE Linux system).
    - For C/C++ code, you need to provide a Makefile using which your code can be compiled in a batch.
    - For Java code, you need to provide a build.xml using which the ant tool can compile your code and create a jar file for your program. You also need to provide the other jar files that you use.
    - You are required to put all the commands that you test your algorithms into a command file `project_commands.txt`.
- (50 pts) Report
  - (15 pts) Problem definition: clearly define a meaningful data mining problem. The scores you get depend on how meaningful and hard the problem is.
  - (15 pts) 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.
  - (15 pts) 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.
  - (5 pts) English (correct grammar, logical sentences, etc.) (Deduct 0.5 points for each error, maximum deduction 5 points);
- (10 pts) Team work
  - (5 pts) Create a README.txt to write down the work allocation of the different team members.
  - (5 pts) Each team member needs to create a `PeerEvaluation_<yourname>.txt` to include a peer evaluation to your team members. Your peer evaluation 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)

- You final project score will be **weighted/adjusted** based on the work allocation and peer evaluation results.
- The project will be graded after everything is submitted (i.e., the deadline of the last stage). If one stage is submitted late, the late penalty (details see course syllabus) will be applied to the portion of that stage.