IRDS: Instructions for Mini-Project

Autumn 2016

The mini-project is an open-ended project in which you should compare a number of data science methods on a real data set or a realistic scenario. You choose which types of methods and which data sets you wish to compare. For example, you may choose to compare methods in machine learning, statistics, optimization, natural language processing, computer vision, databases — anything that falls within the broad remit of "data science". Or you might compare different methods for visualization / exploratory data analysis, and examine which methods are most useful for extracting knowledge from the data set.

Essentially, your project should involve

1. Exploring the data set to determine which methods and features are likely to work well

2. Choosing some methods that might work well on this task, based on your exploration of the data set or of previous methods that have been tried on this task.

3. Evaluating the results of the different methods on the task. Based on your results, are you confident which method is best? What have you learned about the original problem?

Because the definition of the project is deliberately vague, here is an example to give you a sense of how much effort is expected. You might choose a data set, split it into a training and test set, visualize some of the features that are most closely associated with the class label, and compare several different classification algorithms on the data set, choosing their tuning parameters carefully (e.g., by cross validation).

Your project need not follow this template. For example, you might instead choose to compare different methods of feature selection or different hand-built choices of features, rather than focusing on the choice of classification algorithm. Additionally, you might choose to generate learning curves to find out the extent to which the performance of the classifier degrades as a function of the size of the training set. Or you might try to find additional information from the Internet that you can use as features. If you are extremely ambitious, you might think about whether you could use more advanced ML techniques, but then you would need to be sure that you have a good existing implementation available — you will not have time to implement and debug something complex as part of this project. Note that this list is intended as a source of ideas to choose from; you are certainly NOT expected to do all of those things in one project.

It is also not necessary that your project use machine learning. For example, if you are interested in learning more about databases, you might compare the performance of different types of data processing systems on a range of practical queries. Whatever topic you choose, your project should involve empirical exploration, comparison, and evaluation of methods in a realistic practical setting.

Essentially, you are expected to use your insight and imagination to try something that you think will perform well on the task. If your ideas don't work out, that is OK, as long as they are reasonable and properly evaluated.

Projects will typically include some amount of exploratory data analysis, using graphics or summary statistics, as appropriate. This you help you decide which methods or features to use.

Overall I would expect the mini-project to take you around 40 hours work.

Choosing Your Project. A list of potential data sets and projects is given on the IRDS web page. You may choose one of these, or propose your own project if you like. If you wish to propose your own project, please discuss it with me, so that we can make sure it is feasible in time that you have available.

Existing Software. You are not expected to implement your own learning methods or to develop new methods, although you are allowed to do so if you wish. You may use any existing software that you like. In your report, you should be clear about what software you used from others, and what you did yourself as part of the project.

Comparison to Previous Work. You are not expected to match the best published performance on your data set in the amount of time that you have available. However, a good project will discuss: How do your numbers compare to the numbers in the previous work? Is simply comparing the numerical results fair? (There are various reasons why it might not be, and that's OK.) What are the most important things you would do next if you had time?

Report Structure

The report should be around 6 pages in length of single spaced text. The report should describe what the problem was, what you did, why you made the decisions that you made, and what happened. The exact structure of your report will depend on your specific project, but the following headings are likely to be useful.

- Overview of the task
- Analytic goals (see below)
- Data preparation
- Exploratory data analysis
- Methods used
- Results, evaluation
- Conclusions

Analytic Goals

Whether you are working in industry as a data scientist, or if you are carrying out research that is more applied, an important skill is communicating the results of your analysis to the "owner" of a data set in such a way that they can understand and use the information that you have discovered from the data.

So, as part of the final report, I'd like to ask you to imagine that you are actually carrying out this analysis as part of a "real job" in industry or the public sector. Think about: Who are the end users who care about this data set? Why did they collect it? What questions are likely to be of interest to them? What background are they likely to have? What kind of information are they likely to be looking for? What information are they likely to treat as "common sense" based on their experience? (Presenting evidence from the data that is confirmed by their experience is a good way to build trust.)

Then write a section of the report that explains the answers to these questions and how they have impacted the analyses that you have done. Explain how you would summarize the main messages of your report to the end users of this information, and which of the figures in your report you would rely on to do so.

It may seem artificial to need to imagine this, because in real life, you could just ask them, but in fact nothing could be more practical than an exercise of empathy, that is, of trying to build your own independent understanding of the end user's point of view. This is because to do good applied work, you need to understand both the details of what is and is not possible using current analytic techniques (that is what you know as a data scientist) and what end questions are most of interest in the domain (this is what your users know). The more you understand of the other side, the easier it is for you to focus the analysis on the questions that are most important, and sometimes to suggest that the data might be able to help with questions that others didn't realize that it could.

Presentation

You should prepare a five minute presentation based on your project, to be presented in class. The audience should be an imaginary person who is an expert in your data set, but not in machine learning or data science methods, who wants to understand the results of your study. The actual audience for your presentation will be the other students and lecturer in the class (i.e., not imaginary), but we will try to pretend to be this imaginary person for the purposes of this exercise.

Your presentation should summarize succinctly the data set and analytic task that you are trying to solve, any relevant details about the data set that arose in your exploratory data analysis, what methods you applied to the data, and what the major results were. You will not be able to explain all of the details of this in five minutes, but you should be able to give a high-level summary.

We will have all of the talks in a single 90-minute session, with 5 minutes for each talk, 3 minutes for questions from the audience, and one minute for changeover between speakers. Speaker order will be decided randomly.

How much technical detail to present here is a difficult tradeoff. On the one hand, if you say nothing about what methods that you have used, then your audience will not have any background as to how far to trust your conclusions. On the other hand, if you go into technical details, you will obscure your main message. You want to be somewhere in between these two extremes, but unfortunately I suspect that the only way to get a sense of what the right tradeoff is by experience. (And the right tradeoff will depend strongly on your audience.)

You will notice that the presentations occur *earlier* than the deadline for the written report. Although this is primarily due to scheduling constraints, it does have the good effect that you can use feedback from the presentation to help you final written report (which does after all account for most of the mark).

Schedule

Choosing your project: **By 4pm on the deadline listed on the course web site**, please send an email to the instructor and TA that says which project and data set you would like to work on. If you choose one of the suggested projects from the web site, then just give the title. If you are choosing a data set not on the list, then please write a paragraph or two that explains the data set and task, similar to the descriptions on the web site. It would be good to chat with me first before writing this description.

Interim report: By **4 pm on the deadline listed on the course web site** you must email the instructor a text description of your progress so far and your plans for completion of the mini-project by the final deadline. A short paragraph is fine. You should discuss what comparisons you want to run. This report will not form part of your numerical mark for the course. The goal of interim report is to make sure that your project has the right scope and that you are on track.

Tutorials: The project will be supported by tutorial meetings during the CDT Tea and Biscuits. These will be informal groups in which you can discuss your projects with your fellow students, and share ideas and advice.

Final due date: Evaluation of the work on the mini project will be by a written report. This is due by manual submission to ITO by **4pm on on the deadline listed on the course web site**.

Presentation: Finally, you will present your work in a five-minute talk during class time to share the results more generally. The date for this will be announced and posted on the course web site.

Late penalties: The policy of the School of Informatics is that no late submissions are allowed except on valid ground agreed a priori with the year organiser.

Marking

The project will be marked 80% based on the written report, and 20% based on your presentation.

The marking criteria for the written report include the appropriateness of the methods chosen, quality of the analysis, the quality of the evaluation, the amount of work, and the quality of the explanation of the report (both text and graphics).

Your presentation will be marked primarily on how effective it is on helping people to understand the main ideas of your project. It is unnecessary to spend inordinate time on the design of your slides. "Simple but clear" should be your goal.

A guide to the letter marks are:

A Well explained description of points above plus extra achievement at understanding or analysis of results. Clear explanations, evidence of creative or deeper thought will contribute to a higher grade.
B Well explained description of points above.
C Good description of points above but significant deficiencies.
D Evidence that the student has gained some understanding, but not addressed that specified task properly.
E/F/G serious error or slack work.

Policy on Collaboration and Plagiarism

I encourage you to discuss your projects with other students in order to share ideas and ask questions. Indeed, we will have tutorial groups that are specifically designed for you to do so.

That said, the work and the writing that you present should be your own. For more information about the School Plagiarism policy, see `http://www.inf.ed.ac.uk/admin/ITO/DivisionalGuidelinesPlagiarism.html`.