EE 660 **Project FAQs and Tips** Fall 2020

Jenkins v1.0, 10/16/2020

FAQs

Question 1: What is an appropriate workload or difficulty level for a project?

Answer:
First, there are no specific (quantitative) requirements on difficulty level - it will be assessed on the basis of the entire project. The project can involve many aspects: for example preprocessing, feature selection or dimensionality adjustment, issues like dealing with unbalanced data, trying various models to get good performance, feature sets that include disparate feature types, etc. A good project might be difficult in some aspects and easy in other aspects; or might be moderate in a number of aspects.

The project should be more difficult and involved than a computer homework problem. For example, if the dataset has all features of similar type, all numeric and ready to use, with no missing data; it's relatively small with a small number of informative features; you use mostly or entirely library routines and so have minimal coding; and good results are easily obtained with most any hypothesis set, then it's too easy.

Here are some examples of what can make the dataset or problem more challenging, in each of 6 aspects:

- Preprocessing: missing data, noisy data, categorical data, other issues;
- Features: disparate feature types to deal with or needs significant dimensionality adjustment; or filtering out of possibly irrelevant features;
- Models: it takes significant effort to get good performance;
- Performance evaluation: the problem requires alternate performance measures, and/or has significantly unbalanced data
- Approaches and data preparation: no one has used the dataset in an ML problem before.
- Coding: doing a lot of your own coding because of need for non-standard algorithms, nonstandard use, or compute-time efficiency

If your dataset has some issues like these in a couple of aspects (or a particularly difficult issue in one aspect), it is probably about right for an individual project. If it has issues like these in more aspects, then it's probably more appropriate for a team project.

Note that the time and effort the project takes you should be about the same whether it's a team project or individual project. But a team has more people working on it so will make more progress.

Note that some topics or datasets can have a difficulty level or workload that you can control. For example, an easy version of the problem might involve using a portion of the features to predict one thing. A more difficult version might involve more features (such that the additional features involve issues that the first set of features didn't have - e.g., noisier, more missing data, more reformatting required, etc.); and/or a more difficult version might predict something that is more difficult to get good performance on. This type of project can be good because you can adapt the difficulty or workload as you go. But it's not necessary, and many datasets might not be amenable to varying the difficulty level.

Another option is to use 2 datasets, maybe one that is pretty easy and one that is more challenging.

Question 2: How will performance of my machine-learning system be graded?

Answer:
We do not grade on an absolute scale. Some datasets get a very high performance easily, and others take a lot of work to get small performance gains. We take this into account, as follows:

(i) We compare your results to other results - first to your baseline(s), and consider what your choices for baseline(s) were; and second, if relevant, to any known works from others (e.g., published in the literature or posted online), considering the system you are comparing to (e.g., we may not expect your results to match or beat published works, because most published works are based on a lot more time and effort than is available for a class project).

(ii) The work you did, which can indicate how easy or difficult it is to get results of a certain performance level.

The performance of your final system (graded as described above) will contribute about 15% of your total project grade (exact weight to be determined, but will be close to 15%).

Question 3: What are the criteria by which our project will be graded?

Answer:
Please see the Project Assignment, page 5, "Grading Criteria". It lists 7 criteria; each criterion will be scored as 0-100, and your total project score will be a weighted sum of each criterion score. The typical weight for a criterion will be in the range 15% - 20%.

Question 4: Can we use algorithms or methods that are not covered in class? Can we use Deep learning algorithms?

Answer:
You can use algorithms that are out of the scope of EE660, but at least 50% of project work must be methods covered in class.

Deep learning algorithms are not the focus in this class. Datasets most suited to deep learning methods are images, natural languages, videos, etc. But datasets suited for methods covered in class are commonly numerical or categorical. You may use deep learning as feature extraction method. But machine learning algorithms and analysis should be the focus of your project.

Tips

1. **Include at least one baseline system, although it is recommended to use 2 baseline systems (one trivial and one nontrivial)**. This should be done for any machine learning project or system development, including this course project.

   There are 2 types of baseline systems: trivial and nontrivial. Include at least a trivial baseline system with your course project; its purpose is to check if your system has actually learned anything. Examples of a trivial baseline system include:

   (i) For a 2-class pattern classification problem in which performance is measured by percent correct classification: always decide the majority class.
   (ii) For a regression system in which performance is measured by MSE: always output the mean of the training-set outputs, $\bar{y}_{Tr}$.

   The trivial-system performance can be calculated analytically or numerically, or both (your choice).

   You can choose something different for your own trivial baseline system; the idea is a system that is trivial (doesn't really involve machine learning), but given that it is trivial, it outputs a reasonable best guess.

   Nontrivial baseline systems include:

   (i) System that uses some machine learning algorithm, but maybe is a first reasonable try, or a first system that does better than the trivial baseline system. In this case, your goal would be to do significantly better that the nontrivial baseline. For examples, refer to Discussion 8.
   (ii) System you have found published in the literature or posted online. In this case, the nontrivial baseline is more of a benchmark that you would compare your result to.

2. **Online tutorial or demo machine learning systems are generally not allowed for your project topic.** Feel free to try these as a potentially worthwhile learning or practice experience, but using such a demo or tutorial example that you have copied from elsewhere, for your project, won't get you any points. Additionally, if you don't adequately cite the source and show (in your report and code) what was copied or used from the source, you could be committing plagiarism which would cost you substantial penalty.