## Term Project guidelines and dates

You will be randomly assigned to **groups of four to five** students for the term project to analyze and develop a machine learning solution using (preferably) business-related data.

**Dates**:

1. **Project outline due Oct 23**. 2-3 pages describing the problem, data available, some possible approaches you will consider to address the problem, and a short list of references. (need not be fully flushed out, more for a sanity check).
2. **In-class presentation** of project results, Late Nov/early Dec, approx 15-20 mins per group.
3. **Blog due by midnight, Dec 12h**, via Canvas. One submission per group. You are also asked to  submit supplementary materials (code, referenced papers) via a pointer to the appropriate URL/dropbox/github/.. location(s). See instructions in the “blog” document.
4. Blogs are usually posted in a publicly accessible location, however if you don’t want to make the blog public, that is OK with me so long as I have access and thus can read it.
5. **Peer-evaluation survey, due Dec 12th.** Done via canvas.

## Project presentation schedule

Project groups, title and schedule will be available on Canvas when ready. Guidelines for your in-class presentation and for the content of  report and the criteria for its evaluation are uploaded into Modules - Projects

## Project topics

The project should be centered around some predictive modeling problem that involves the use of a reasonably large dataset(s). In the process, if you invent new techniques/algorithms or processes, or make inferences that are useful and not done before, of course that is an added bonus, though this is not common.

An .xls with pointers to projects done by a previous class can be found via Canvas.

Two common types of projects are:

### Type I: Based on a Competition or other Real-World Large Datasets

Such as those available on **Kaggle**.

**Type II: Based on Type of Analysis or Application Domain**

Current hot topics include:

1. Concept drift: Dealing with changing data statistics over time, e.g. see this and this

2. Video recommendation (e.g. tiktok style). Modern recommendation systems that involve deep learning (go beyond simple collaborative filtering) and work on very large/complex data, e.g. see video recommendation datasets.

3. Pretty much anything to deal with LLMs and other foundation models.

**Note:**

1. **The project should be doable within a couple of months, but also non-trivial: at the very least it should involve a large (say "rows" times "columns" > 100K) data set**. (for grayscale images column size is # pixels in an image). Remember that your class presentation is public, however your class report is not, and I (and the TA) can sign NDAs if need be in order to work with you on such a project and to evaluate it.

2. Policy on Code reuse: Nowadays a large number of pretrained models are publicly available. Also many forums, including Kaggle, post a lot of code. Powerful pre-trained models are most notably available for text analytics, one reason (besides the fact that there is a separate course on this topic) that pure text analytics projects are discouraged.

   If your project involves use of pre-trained model or code posted by others, you should explicitly mention (and specify which portions were reused/copied code vs. what is a new contribution from your team) and provide links to the source(s). If software used to detected copied code flags your submission and you have not declared the source(s) that this software discovers, you can be heavily penalized.

**Team Grading Policy**: **Grading**: This project is a critical part of the course, and a significant factor in determining your grade. By default, all team members will receive the same score for their project. To ensure that this is fair, each member needs to submit a “peer assessment” form along with the final report. If these forms suggest HIGHLY imbalanced contributions, then I will need to look at the issue in more detail and most likely have a meeting with all the members together to mediate and come up with a fair score distribution.