**Data Analytics Pipeline**

It's at the intersection of engineering, data science, and project management, so there's quite a bit of each in here. Some topics are probably extensive for a lecture, so there may be a subset of the bullet points under each during the lecture.

1. A brief history of machine learning
   1. Develop basic familiarity with models and scaling
   2. give some context for where models are implemented in production applications
   3. Emphasize cost (dev + maintenance + workflow/human issues) vs. value of deliverables
   4. cover some basic model zoology
2. Using data to generate value
   1. Prioritization of analytics vs. machine learning
   2. Examples from popular data science blogs
   3. Massive value from processes as simple as ranking data (NYC budget anomaly; parking from ben wellington)
   4. Changing user experience through AB testing and design vs. algorithms
   5. Using algorithms to create a new product experience
3. Dashboards and reporting
   1. (review) Value can come from simple analytics tasks
   2. Provide more information to power actions/decisions
   3. You must integrate with workflow
   4. Speed of delivery vs. complexity of the product
   5. Scope and incrementalism (basic MVP -> mature product)
4. Team structure and the development process
   1. Where do you fit in to the team?
   2. Data Scientists vs. Data Engineers vs. Engineers
   3. Collaboration with product and stakeholders
   4. Agile development
   5. Resource constraints and stakeholder expectations: why we start minimal
5. Service-oriented architectures and data science
   1. Benefits of service oriented architecture
      1. good abstraction gives modularity, and a path for growth
      2. spaghetti and glue code, and how to avoid it
   2. Basic architectures for machine learning
      1. batch vs. online computing
      2. caching vs. real-time models
      3. model maintenance
   3. Model performance
      1. parallelism and horizontal scaling
      2. speed of development vs. runtime (python vs. C++/Go vs. etc.)
      3. Services in the critical path (e.g. loading during the pageview!)
      4. Sources of latency
6. Minimum viable products and feature creep
   1. Example of services and MVPs for services
   2. Good abstraction (e.g. distinct services with nice APIs) helps you move fast
   3. Where do features come from
      1. things that look cool
      2. stakeholder requests
      3. collaborator requests
   4. Solutions
      1. keep services simple
      2. define your priorities
      3. pay off technical debt while adding features
      4. keep refactoring costs in mind when giving resource estimates
7. Architectures for machine learning systems
   1. Examples of some real systems
      1. basic data pipelines (NSQ, kafka)
      2. dashboard backend (pipeline -> online statistics service -> database. database -> dashboard API)
      3. feature extraction and tagging (queuereader -> ML service -> feature data store via an API)
      4. neural net architecture (batch model training -> model save. model load -> real-time API with load balancer)
   2. focus on covering batch vs. online processing, and real-time computation vs. result caching.
8. Data logistics problems and machine learning
   1. Data locality for batch computing systems (cost/benefit of HDFS vs. remote data stores and cluster maintenance)
   2. sampling for online systems that match events (e.g. clicks and impressions) across time
   3. Data siloing and how to avoid it (e.g. publish to stream on db insert)
9. Model logistics and performance
   1. separating model training and loading
   2. example: Netflix's H0ll0w package
   3. writing a basic model logistics library (cross-validate, save, load)
   4. parallelizing online training
10. Cloud computing frameworks
    1. AWS and google cloud
    2. cluster computing
    3. data storage
    4. lambdas
    5. basics of cluster management
       1. fixing a version
       2. auto-terminate
       3. access credentials
       4. read-only data stores
11. Third party services and APIs
    1. work within your org's expertise, and outsource some services
    2. when to outsource
    3. how to validate a service provider
       1. product support guarantees
       2. reverse compatibility
       3. comparing performances
12. AB testing machine learning products
    1. Basic testing frameworks and test design
    2. dependent sample issues
    3. sources of bias (especially selection effects)

**COURSE GRADING**

|  |  |
| --- | --- |
| **Percentage**  **30%** | **Homework Assignment(s)**: There will be 5 homework assignments. Each assignment will be worth 6% of the final grade. |
| **20%** | **Midterm**: The midterm is one class period, open book, computer permitted. The midterm will cover all material discussed in the course up to the week before the exam. |
| **20%** | **Final**: The final exam is scheduled at the normal final exam time for this class period. The final is open book, computer permitted. The final is cumulative and will cover all material discussed in the course. |
| **30%** | **Course Project(s)** |

|  |
| --- |
| **ASSIGNMENT POLICY** |
| All work is due by the date and time specified in the respective assignment. It is much better to submit partially complete homework on time and get partial credit for your work than to submit late homework for no credit. Assignments submitted after the respective deadlines when they are due are considered late. Late assignments will not be accepted unless there is a valid medical or family condition with appropriate documentation submitted to the instructor. |
| **GRADING POLICY** |
| If you disagree with any grade, submit your grievance via email documenting the merits of your case. |
| **COLLABORATION / COPYING POLICY** |
| We encourage you to help one another in understanding the concepts and principles needed to do the assignments for this class. However, what you turn in must be your own, or for group projects, your group's own work. Copying any part of other people's code, solution sets, or from any other sources is strictly prohibited. The assignments must be the work of the students turning them in. Anyone found violating the class collaboration policy will be subject to the School’s Dean’s Disciplinary process.  You must explicitly cite ALL sources of information that you reference as part of your assignment submissions. For each citation, you should describe how that source was referenced. Failing to cite your sources used to inform your work will violate the policy. The University Library has resources for help with providing references and citations for your work.  Note that you are responsible to ensure the integrity of your own work. Do not leave copies of your assignments lying around for others to have access to. |
| **DISABILITY SERVICES** |
| Disability Services facilitates equal access for students with disabilities by coordinating accommodations and support services, cultivating a campus culture that is sensitive and responsive to the needs of students.  Students seeking accommodations or support services from Disability Services are required to [register with the office](http://health.columbia.edu/getting-care/register-disability-services). If you are interested in pursuing an evaluation for a learning disability, please visit the [referrals and other campus resources page](http://health.columbia.edu/students/university-and-affiliate-referrals-disability-concerns). |

