Final Project Guideline
CISC 5950 - Big Data Programming

Goal: The goal of the final project is to get hands-on experience with a specific Big Data technology and to communicate this knowledge to others.

Background

New technologies emerge every day and existing technologies change rapidly. As a result, technologists rely on other practitioners to learn new methodologies and tools. One way they do this is through tutorials. Tutorials written by professionals often give users a set of practical tips on how to get up and running with a new technology quickly. For instance, a given tutorial might describe how to install and configure a technology such as Hadoop, and then go on to describe how to write your first set of batch computations using the MapReduce programming paradigm.

For this final project you will work in teams to learn a Big Data technology and then create a tutorial for teaching other members of the class. Not only will you get practical experience learning a technology as you would in an industry setting, but you'll have a polished product to include in your portfolio for potential employers at the end of the semester.

Format

The class will be split into groups based on students' interests and background (i.e. analyst vs. engineer). Each group will be responsible for working together to create a final product (outlined below in the *Deliverable* section). The group will give a 20 minute presentation to the rest of the class during Finals.

Deliverable

Each group is responsible for delivering:

1. A blog post style tutorial that
    a. presents a high level overview of the technology. Give readers background information on the tool. What problem does the technology solve? How was the technology developed? Was it developed at a company or in academia? Basically, provide useful information that explains why anyone would use this Big Data technology.
    b. Walks users through creating a sample application with code. This code should also live in a separate set of files (see the following point).

2. A working set of code that (at the very least)
    a. Installs the technology.
    b. Configures the environment.
    c. Performs some task. This task is highly dependent on the technology itself.

    This code should live within a GitHub repository and be publically available.

Examples

Here are a few links to some tutorials I've found online. This list is **by no means exhaustive**. There are many, many tutorials online covering all sorts of things from web development to artificial intelligence algorithms.

- https://machinelearningmastery.com/machine-learning-in-python-step-by-step/
- https://marcobonzanini.com/2015/10/24/building-data-pipelines-with-python-and-luigi/
- https://blog.miguelgrinberg.com/post/the-flask-mega-tutorial-part-i-hello-world

Grading

Although I haven't worked out the grading rubric entirely, students can expect that projects will be graded based on:

- Clarity of the exposition: Does your tutorial make sense? If I had never used this technology before, would I walk away with an understanding of what the technology is useful for? Would I be able to write a basic program with the technology?

- Organization and Documentation: Is your code carefully explained? Does it make sense? Does it run if I follow your directions exactly?

- Relevance of the Sample Application: The sample code written should perform some task that would actually be performed using that technology. For example, the "Hello,World" of the Hadoop world is to write a batch job that performs a word count. Don't try to write a tutorial on how to use Hadoop to create a web application.