Massive Data Analytics Final Project

Objectives

This project seeks to give you the opportunity to design and conduct a full big data analysis using the tools from this course. This is a common activity for professional data scientists. You will do this as small teams of 2-3 people to give you the experience of collaborative data science. In this project you will:

1) Define the questions you seek to answer and the model you would like to build
2) Implement a big-data solution to help find those answers and build the model
3) Prepare a final presentation of the analysis for your classmates
4) Write a written report describing your work

Milestones & Due Dates

1. Teams Assigned - February 17th
2. Define Analysis / Modeling Objectives - March 17th
3. White Paper Rough-Draft - April 28th
4. Final Presentation & White Paper Due - May 12th

2 and 3 will not be graded but to ensure progress each team should submit these to us by the due dates written

Dataset

We will continue to use the Cord-19 dataset which we have been getting comfortable with throughout the semester. Now you have the opportunity to do a full analysis and exploration on your own.

Instructions for accessing the CORD-19 dataset can be found here:

https://registry.opendata.aws/cord-19/

Defining Analysis / Modeling Objectives

It is often helpful to describe up-front what you're trying to achieve with the work. When doing Data Science whether on big-data or small data, there are usually two main components: exploratory analysis and modeling. Exploratory analysis helps you to understand more about

the data at hand. It can answer targeted questions or bring clarity to a broader problem domain. Modeling seeks to use the data to construct a mathematical representation of the relationships. This can be used for prediction of unseen data or inferences about populations from samples of data. Modeling requires exploratory analysis up-front.

What we're expecting here is that you've thought through:

1) The problem you're trying to solve or the question you're trying to answer
2) What tools and techniques you'll need to solve that question
3) What assumptions you're making up front about the data and the problem

White Paper Expectations

The white paper should contain the following elements:

- Names
- Introduction
- Code (you should list your code files here, and these must be in your submitted repository with appropriate comments.)
- Methods section:
    - How you cleaned, prepared the dataset with samples of intermediate data
    - Tools you used for analyzing the dataset and the justification (tools, models, etc.)
    - How you modeled the dataset, what techniques did you use and why?
    - Did you have a hypothesis that you were trying to prove?
    - Or did you just visualize the dataset?
- Results section:
    - What you found.
    - How you validated what you found (if you validated what you found)
- Future work
    - What would you do differently?
    - What new questions do you have?

Final Presentation Expectations

On the last day of class each team will have 30 minutes to present their work to the rest of the class. Cover the main topics from the white paper. Use the experience you have gained with the paper presentations to make the material accessible and engaging. Think about what you learned from the project. What was surprising to you. What new questions do you have now that you didn't have at the beginning.

Final Deliverables

1. 30 Minute presentation to your classmates
2. “White-paper” describing your analysis & infrastructure
3. Code used to conduct analysis

Grading

While the project is open-ended, there are some parameters and guidelines to help plan and organize your approach. The project needs to have, at a minimum, the following:

Exploratory Analysis (80 points): Explore, assess and visualize the data. Aggregate, count, and summarize. Create graphs, tables, etc and explain your findings in writing. Clean data if necessary.

Model (80 points): Build any type of model you feel is appropriate and meaningful. You can perform any type of supervised or unsupervised approach. You must have evaluation metrics for supervised approaches and/or visualizations for unsupervised learning approaches. You are welcome to try different modeling techniques that you are comfortable with.

White Paper (20 points)

Presentation (20 Points)

Extra Credit (up to 20 pts): You can run multiple models or use different and reasonable feature combinations and conduct an error analysis to get the extra credit.

The project will be graded holistically with the following rubric:

- Grade of A:
    - Writeup covers all areas above
    - Language is clear, figures support research/investigation
    - There is discussion on specifics of the analysis, and analysis decisions are justified
    - Properly formatted
    - Presentation is clear and engaging
- Grade of B:
    - One major deficiency and/or
    - Writeup and/or analysis is missing significant discussion/justification around analysis performed and/or
    - Minor flaws in layout/presentation of analysis and/or
    - Presentation is hard to follow, or sloppy

- Grade of C:
    - Two or more deficient areas and/or
    - Major flaws in layout/presentation of analysis
    - For the purposes of grading, a deficiency can mean any of the following:
        - Instructions are not followed
        - There are more files in your repository than need to be
        - Missing sections of the writeup
        - Poor and sloppy writing and/or presentation
        - Many spelling and grammatical errors
        - Code is not documented with comments
        - Missing model performance metrics
        - Doing an analysis and/or model just for the sake of doing it, without thinking through and providing justification

Submitting the Project

The files to be submitted for the project are:

- Project.txt is your project writeup file.
- Instance-metadata.json
- Any code/notebook file referenced in the code section of your writeup.
    - We should be able to follow the code with the writeup.

General Recommendations

- Use spot pricing for your cluster. With m4.xlarge machines, each machine costs $0.20/hr per machine time, plus the EMR fee of $0.06 for a total of $0.26 per machine per hour. You can save money with spot pricing (just on machine time, not on EMR cost.) Keep track of your spend!
- Refer to the PySpark Documentation (https://spark.apache.org/docs/latest/api/python/index.html) and Spark Documentation (https://spark.apache.org/docs/latest/)
- Consider saving intermediate datasets in your S3 buckets, in Apache Parquet format.
- Consider saving a model object in S3 after you train it, especially if training takes a while.
    - To save a model object, use the following code:
        - model.save("s3://[[your-s3-ucket]]/model_location/")
- When creating the Machine Learning pipelines, you may want to try it first on a small sample of your training data to make sure the pipelines work as planned. To create a tiny DataFrame, use the limit method: df.limit(100) (this creates a small DataFrame with the first 100 rows from df.)
- If you need to re-start your Jupyter notebook for any reason, make sure you close the Spark connection first before restarting the kernel. To do this, type either sc.stop() or spark.stop() in a cell. If you don't do this, YARN will not release resources previously allocated.