DSCI 551 – Fall 2020

Project Guideline

Data management and exploration for X

You are free to choose a topic of interest X. You can get idea from the areas and interests (collected from your survey) posted. You can also find data sets from Kaggle and some other web sites (google "data science data sets"), e.g., https://www.springboard.com/blog/free-public-data-sets-data-science-project/

You can find inspiration from many online resources (e.g., example projects):

- https://www.ischool.berkeley.edu/programs/mids/capstone
- https://www.datascience.columbia.edu/2020-data-science-institute-student-capstone-projects
- https://datasciencedegree.wisconsin.edu/capstone-projects/

Data sets (please feel free to share yours on Piazza)

- https://www.kaggle.com/c/home-depot-product-search-relevance (this data set has a data quality problem/typos/misspellings, especially among user searches).
- https://www.kaggle.com/aestheteaman01/harp-healthcare-availability-resource-prediction (this covid-19 data set may have missing value problems)
- Social data - Harvard Dataverse https://dataverse.harvard.edu/ (it has Covid-19 data among many others.)
- Online networks/social media - SNAP http://snap.stanford.edu/
- Education data – DataShop @ CMU https://pslcdatashop.web.cmu.edu/
- General – Academic Torrents https://academictorrents.com/
- UCI Machine Learning Repository https://archive.ics.uci.edu/ml/index.php

You may also find a sample project (on keyword-driven exploration of data in database) from previous semester helpful:

- https://youtu.be/mpHjROTbUx4

Please avoid using the data sets that have already been extensively used and explored (IMDB, MovieLends, NBA scores, FIFA, sample databases from MySQL/SQL Server, etc.)

Note also that X can also be a topic on computer science/IT. For example, you can think about how to improve software/tools you are using, e.g., analyze the server logs (MySQL server, web server, MongoDB server, etc.), etc.

Please focus on the **data management** aspects of project, including:

- Data storage (cloud data storage, e.g., Amazon S3, Firebase storage)
- Data modeling (relations, JSON, graph, XML, etc.)
- Data types (handling different data types: time series, images, audio, texts, etc.)
- Data cleaning (including entity resolution/record linkage)
- Data transformation (e.g., JSON into relational table, first + last name => full name)
- Data integration (e.g., integrating Home Depot products with reviews and ratings from Amazon; integrating job postings of multiple web sites; and developing a unified search interface over data in MySQL (e.g., products) /MongoDB (sales data)/Firebase (customer reviews /etc.)
- Data retrieval (e.g., developing a form-based interface to allow users search for JSON in MongoDB)
- Data aggregation/exploration (e.g., derive statistics/summary from data set and allow users to explore them, similar to that in OLAP).
- Feature extraction (e.g., extracting features/new data/metadata from images and storing them in database for data retrieval and processing) Note that the focus should not be on machine learning aspect, e.g., how to select a good feature, but on how to allow users to explore/search data using the extracted feature.
- Parallel data processing (e.g., using Spark to perform data cleaning/transformation, feature extraction, data aggregation, etc.)
- Scalable data search/exploration/processing: what if the data set you are using is 1000 times larger (GBs or even TBs), does your solution still work?

Here are general requirements that every team should meet:

- Use at least one database (relational or NoSQL or cloud) to store the data (raw and derived) used for the project.
- Use Spark to perform at least one data wrangling/processing/analytics tasks: data cleaning, data transformation, feature extraction, data aggregation, etc. You should store the results in the database and allow user to search/explore. For example, take COVID-19 case data, you can precompute the distribution of cases by states, countries, continents, etc.
  You are required to use Spark dataframes/RDD API (instead of SQL) so that you gain insights on how Spark works internally.
  In the proposal stage, please identify the area in the project that can benefit from parallel processing.
- Develop an intuitive interface for **searching** and **exploring** data in the database. (If you are not familiar with Web app development and have only one person in the team, it is acceptable that the interface only displays data without search function. You can take Firebase Web API sample as a starting point.) Exploration can be done using keyword search and present users ways to further explore the search results, e.g., using faceted

interface (e.g., explore sales data by region, product category, etc.), or see the sample project mentioned earlier.
The interface may be Web browser based or mobile app.

Additional requirements for 2-person projects (in additional to general requirements):

- You need to use at least two databases, one relational (e.g., MySQL) and the other either NoSQL or cloud data storages (e.g., S3 or Firebase storage).
- Your interface should have search function (i.e., take input from users), and should allow users to explore all the data in the databases (raw and derived) without using visualization tools. The exploration should involve the computation and display of summary information about the data (e.g., group-by and aggregation).

Additional requirements for 3-person team (in addition to 2-person requirements):

- NoSQL database you are using should be a cloud-hosted database, e.g., Firebase, MongoDB on the cloud, etc.
- You need to find a realistic use case where data from different databases need to be integrated meaningfully for display in the interface. For example, point-of-interest data in MongoDB with demographic data in MySQL; stock market data in MySQL and twitter data in Firebase; etc.

Note that depending on your background and skills, you are free to implement machine learning, data visualization, NLP for the project (e.g., you are expanding this project for other courses you are taking); but these cannot be used to satisfy the above basic requirements. However, you can store the analytics results in databases and allow users to search in the interface to satisfy the data exploration requirements. You can also identify the area that Spark can help and implement it using your own algorithms (instead of simply application of existing ML algorithms in Spark for example). For example, write a parallel algorithm for analyzing texts (recall WordCount example shown in class).

Team formation: Typically, you find group members from the same class, unless you have a very compelling reason not to do so. Individual project is ok. You may also use Piazza to help you find group members. To do so, it will be productive if you introduce your background, interests, potential project ideas, etc. Collaboration among students from different disciplines are strongly encouraged.

Change of topic: Topic change after proposal is permitted only once and handled on a case-by-case basis. No more changes are permitted after the midterm progress report.

Bonus points: We may give truly excellent projects extra credit, typically up to 20% of project scores, based on your feedback and teaching staff's discretion. We will also take into consideration the diversity of your team and how you have helped each other, during our grading and considerations of extra credit. To receive extra credit, we typically ask you to set up a website that shows case your app and a video introduction to the project.

The project consists of 3 phases: proposal, midterm report, final report & demo. The total point of the project is 100, broken down as follows.

- Proposal: 10 points
- Midterm report: 10 points
- Final report: 20 points
- Demo: 10 points
- Project implementation: 50 points

Proposal (1-2 pages):

Your proposal should include the following content. Please also prepare 3-5 slides for a short presentation (1-2 minutes) of your project idea. All members should be present.

- Project title
- Project description
- Data sets
- Data problems to be addressed (data cleaning, transformation, integration, aggregation, etc)
- Databases to be used and how to use them
- Team members, background and skills.
- Milestones and timelines.

Midterm progress report (3-4 pages):

- Provide a checklist showing the items in your timeline and the status on each time (complete, on-going, etc.).
- Provide a screenshot of the components you have completed.
- Are you on track to achieve your milestones?
- Any challenges you have encountered? Any helps that you will need?
- Any other things you think should be reported in the midterm?

Final report (5-10 pages):

It should be a comprehensive report. You may include the contents from your proposal and midterm report, with changes to reflect the final implementation of your project. The final report should have the following parts.

- Project title
- Project description
- System design/architecture
- Data problems that have been addressed, evaluation, and conclusion.
- How did you meet the requirements?
- Team members and what each member has done

Final Demo:

- Demo of your app (10-minute) will be done in the last week.
- Show the working of each component of your system.
- All group members should be present at the presentations.

Deliverables:

Your phase & final reports and project codes (in a sharable link e.g., at Google drive).