TERM PROJECT

This document describes teaming, topics, proposal, grading criteria, and final project report submission guidelines related to the term project. Everyone must do a term project.

Teaming: A team may consist of up to **3 members, except the last team**. A solo project can be approved when there is a good reason.

- **Project grading consists of the following components:**
  - Teaming & Proposal: 20%, Presentation & Final Report: 80%
- **Important dates:**
  - ✓ **Teaming**: Due by the end of Week 3 (11:59PM of April 23, 2023)
  - ✓ **Proposal**: Due by the end of Week 5 (11:59PM of May 7, 2023)
  - ✓ **Project PPT**: Due by Monday of Exam week (11:59PM of June 12, 2023)
  - ✓ **Project Presentation**:
    - Exam week Class time: 6:30-9:20PM of June 12, 2023 and
    - Exam week Office hour: 8:00-9:00PM of June 13, 2023
  - ✓ **Project Report**: Due on Friday of the Exam week (11:59PM, June 16, 2023)
  - ✓ **Project Peer Evaluation Form: Friday,** 11:59PM of Exam week (June 16, 2023)

- *Each team must post their team members and topics by Sunday midnight of Week 3.*
- *If your name is not shown by the due date, I will randomly assign you to a topic to a left-over topic with **zero grade to Teaming score**.*
- **Project proposal**: Each team must submit a proposal by the above due date.
- ***All teams will need to make a presentation in exam week, using a power point file***. I will set up a presentation session using ZOOM.
- *See the end of this file for the submission guideline.*

Topics:

Here are several recommended topics, but not limited to the following topics. If you prefer to work on a different topic, contact me via email before you finalize the topic for the topic and scope.

An important aspect you need to consider when selecting a project topic is that your project should have some **meaningful "data" aspects**. The "data" aspect could include any of the following, but not limited to, ETL, data preprocessing, data transformation, data integration, dealing with unstructured data and manipulation, graph data representation, real-time aspects, data warehousing, data storage, use of a Cloud, use of different big data formats (such as Parquet, Avro, etc), reading data from multiple sources or systems, data virtualization, data visualization, explorative data analysis (before machine learning), outlier detection, data/outcome interpretation, big data analytics lifecycle, solving business problems through the output, SQL, PL/SQL, using data-related tools (NoSQL databases, NewSQL databases, ETL tools, SQL for non-relational or distributed systems, PyMongo, PySpark, etc.)

In the past, some students performed excellent ML projects developing multiple ML models and showing the results, but there was not much on "data" aspects. The project used a clean dataset downloaded from a Web site. The data was "clean" meaning that it had no missing data; no data preprocessing/transformation was necessary; no corelated variables among the variables. There

were a few data visualization showing the relationships between the variables for EDA (but very minor), but that's all. This kind of project may be an excellent project in an ML/DS class, but not an excellent project in an advanced DB class. So, I recommend that when you pick up a ML project using Spark or Tensorflow, choose a dataset that needs some "data" aspects mentioned above such as data preprocessing, explorative data analysis, connecting your output to business problems, etc. **These "data" aspects must be included during your presentation and the final documentation**. Without the data aspects, your project will not merit a high grade.

If you are interested in a few choices but need my advice in deciding a topic, send me an email with the top 3 choices with a short description of your study goal or career goal as well as your background. I will recommend one topic that could best suit your goal. Send me an email (song@drexel.edu) with your email heading of ***INFO607 project****.*

A. Tools and Implementation

This category requires an implementation for hands-on experience.

(1) PySpark

Spark is a parallel computing framework that supports in-memory computing for analytics, machine learning, SQL, databases, and multiple language interface. Experiment with Spark Streaming, Spark Machine Learning, or Spark GraphX. You may experiment with a small application in Spark. There are many resources available on Spark at Google. Use Google Colab for easy implementation. Choose one focus from one or two of the following sub-topics:

a) **Spark ML:** In this topic, you are focus on Spark ML
b) **Spark GraphX:** In this topic, you are focus on Spark GraphX
c) **Spark Streaming:** In this topic, you are focus on Spark Streaming

(2) PyMongo

MongoDB is the most widely used document-oriented JSON-based NoSQL database. Study how MongoDB is used with Python and what they can do together. Experiment with a small data set.

(3) Cassandra

Cassandra is the most widely used columnar NoSQL database. Study its architecture, modeling, and learn its SQL-like query language with CRUD operations, aggregation, indexing, etc. Experiment with a small application in any domain such as stock data, log data, etc.

(4) **Neo4J**

Neo4j is the most widely used open-sourced graph database. Install it, perform CRUD operations and experiment with its Cypher query language. Compare with MySQL performance.

(5) Tensorflow

Tensorflow is a framework for machine learning, especially deep learning applications, produced by Google. Experiment its functionality, usability, and capacity with a case study.

(6) A Cloud-based DW system (BigQuery, AWS Redshift, Snowflake, MS Azure)

Experiment with one of the cloud-based data warehouse systems that supports analytics.

(7) DBT (Data Build Tool)
DBT is an ETL for big data. Try at *https://www.getdbt.com/*.

(7) Airflow
Airflow is an ETL for big data. Try at https://airflow.apache.org/

(8) BigML (bigml.com) or Dataiku (dataiku.com)
BigML (or Dataiku) is a cloud-based automated machine learning tool. It implements various machine learning algorithms and executes with clicks, but without coding. You should explore multiple ML techniques clustering, text analysis, deep learning. There are many tutorials on Youtube.

(9) RapidMiner
RapidMiner is a popular industry-strength data mining tool. Learn it. Experiment with popular examples such as Titanic Data set and or Diabetics data set.

(10) Data Lake/Data Lakehouse Technology
Experiment with implementing a data lake or a data lakehouse. Try Databricks.com.

(11) ChatGPT for BigData Projects
Experiment with ChatGPT for managing and implementing Big Data projects.

(12) **SLAM SQL** (https://slam-for-sql.software.informer.com/) or **Studio 3T** (https://studio3t.com/mongodb-tools/#featureSqlQuery) is known to provide SQL interface for MongoDB. Both are open-sourced. Download and experiment with them and report your experience!

(13) SQL for Data Scientists. SQL is the most important DB language and has many features we could not cover during 605 and 606 classes. Also different vendors might provide other types of advanced functions useful for ETL, Querying and reporting purposes. So, in this topic, you are studying various advanced SQL functions of Oracle, MySQL, Postgres, and SQL Server. Even you may explore new commands that can handle JSON, XML, or multimedia data types.

(14) AWS Hands-on
Experiment with AWS system. Any system is fine.

(15) NewSQL databases
Experiment with any NewSQL database system. Any system is fine.

(16) Your own topic
Email me with your idea for the topic and scope.

B. Case Study/Survey
Teams in this category perform a specific case study. Each team should produce a comprehensive report with about a 20+ page report with a presentation file.

(1) Datafication

Study on various datafication cases. Study their project, business goals, applications, tools, techniques, data elements collected, storage, analysis of the collected data, results, etc.

(2) Your own topic on a case study or a specific topic.

a. Database technologies used in Amazon
b. Covid 19 Data Management and Analytics
c. Data Management and Analytics for Smart Aging (Health)

Term Project Grading Criteria:

Your project will be graded by the following four criteria

(a) *Technical significance*: The contents must have some technical substances, not just concepts.
(b) *Clarity*: The contents must be presented with principles, specific examples, diagrams, and tables whenever possible, so that the ideas can be easily understood.
(c) *Up-to-date*: The contents must be up-to-date with proper references to existing materials.
(d) *Structure and style*: The report must be well organized

- Use proper chapter and section headings
- Discussion style (Introduction-Main-Conclusion)
- Looks (page numbers, grammar, spelling, stapling or binding)
- Citation (*cite properly in the body* of the report, in addition to the list of references)

Proposal

Your proposal should be 3-4 pages in length. Your proposal should have Goal, Context, Scope (**IN-scope** that will be included in your project and **OUT-Scope** that will not be included), Project Plan (how you will proceed to learn the topic), SW/HW involved in the project, Experimentation plan (what data set to use, which techniques to learn, etc, if the project includes implementation), Deliverables, and Reference (at the time of writing the proposal).

- ***Submit a word file, not PDF***, for my easy commenting.
- ***I will create a link to submit the proposal later.***

Model Review, Data Set to use & Incremental Documentation

- If your project involves creation of a database schema, I will review and comment on them without any penalty. Possible artifacts I can review include entity-relationship diagram, any UML diagrams (use case diagram, class diagram, sequence diagram), data warehouse schema such as Star schema or snowflake schema, and relational schema.
- The best method for documentation is *incremental documentation*. **DO NOT WAIT UNTIL YOU COMPLETE YOUR SYSTEM**. *It is very easy to forget the fresh idea you had by the time you finish your project*. Write whenever you have some material that you need to include in your final report.
- If you are in doubt in correctness of your final deliverables, send them to me for my review before you submit the final version to be graded.

Project Presentation and Documentation

Each team should present in the class using a PPT file and then submit a Word file that elaborates the PPT file and describing the detailed technical aspects of the project.
The PPT file should include essential components of the project, which includes project goals, business questions, data used, any data quality and preprocessing issues, main methods employed, results, interpretation of the results, and significance of the project, lessons learned, and references. Add slide numbers.

The Word documentation should elaborate the contents of the PPT file with additional technical details such as data set, installation guidelines, system architecture, implementation aspects, full code or parts of the code if too long, test guideline, remaining issues, etc. Your report should be comprehensive by consolidating all the materials into a single Word file including detailed explanation, diagrams, programs or codes, input, output, testing, etc. That is, copy and paste all the diagrams and programs into the final report as a stand-alone documentation. Organize them in a reviewer-friendly manner. Add page numbers.

- Show me what your team has done, how you have done, what you got as the results, how you tested, how complicated your work is, how important your work is, how I can replicate your work by reading your documentation, etc. Include as many screenshots, tables, and diagrams as possible.
- Your report should include a step-by-step instruction on how to replicate your project. That is, if your project was implemented in a Cloud, you should give me URL, ID and PW that I can log in and a direction on how to use them and what to test. If your system is a standalone or an add-on system, I would like to have a step-by-step instruction on how to install your system/module, and test them on my laptop to appreciate your work.

The typical Project Documentation for an implementation/tool category could be:

a. Goals and scope of the study
b. Setting up the computing environment
c. System architecture
d. Explanation on the modules
e. Data profile (source of data, metadata, data quality, data distribution)
f. Data Transformation (how to read, how the data are pre-processed, and transformed)
g. Method employed
h. System implementation (explanation on data structures such as data frames and interesting codes)
i. Results and Discussion
j. Testing (How I can replicate your experiments)
k. References
l. Appendix

Project Deliverable and Submission

All the project teams will submit a soft copy to Blackboard. I will create a submission link in Week 11 Assignment folder:

(1) Create **one zip file** that includes **the report file in Word, the data set used** (if your project involved an experimentation), **power point file**, or any other related files (e.g, Visio diagram files, program codes, screen shots, output) used in the project.
(2) Make sure to include what data set you used and the source of the data set.
(3) Use the naming convention of **607-23S-P-yourLastName_or_team_name-acronym-of-your-project.zip (e.g., 607-23S-PySpark.zip).** That is, by reading your zip file name, I should be able to easily identify your term and project topic.
(4) Submit project and peer evaluation form to Blackboard Assignment folder of Week 10. Your PEF also should have the naming convention of **607_23S_PEF_yourLastName_ProjectName (e.g., 607_23S_PEF_Song_PySpark.doc).**

- Note that:
  - It is mandatory for your team to create a powerpoint file based on your report and present during the exam week.
  - If you use any diagram, graphics or tables, cite them in both word file and powerpoint file.
  - All members of each team must participate in presentation. Sone online students may be excused if not possible at all. The presentation time could be about 15 min, depending on the number of teams in the class.
  - Every team member must also submit a Peer Evaluation Form (PEF) that evaluates other team members. This form must be submitted within 24 hours of submitting the final project report. PEF link will be created in Week 10 Assignment folder.
  - If you do not submit a PEF, you may get a lower grade than your team members.