16

Project 3 Overview

Requirements

For Project 3, you will work with your group on one of two different track options: data visualizations or data engineering.

Data Visualization Track

For this track, your group will tell a story using data visualizations. Here are the specific requirements:

1. Your project must include visualizations. The visualizations can be created with:
   - Python (e.g. Matplotlib, Pandas plotting, hvplot)
   - JavaScript (e.g. Plotly or Leaflet)
   - A Python or JavaScript visualization library that was not covered in class
2. Data must be stored in and extracted from at least one database (PostgreSQL, MongoDB, SQLite, etc).
3. Your project must include at least one JavaScript OR Python library that we did not cover.
4. Your project must be powered by a dataset with at least 100 records.
5. Your project must include some level of user-driven interaction, such as:
   - HTML menus, dropdowns, and/or textboxes to display JavaScript-powered visualizations
   - Flask backend with interactive API routes that serve back Python or JavaScript created plots
   - Visualizations created from user-selected filtered data, which could be powered by:
     - JavaScript libraries
     - Python in Jupyter Notebook
     - Command-line Python scripts that save visualizations locally

   Remember: You have learned how to filter data in Pandas, JavaScript, SQL, SQLAlchemy, and MongoDB.
6. If possible, your final visualization should ideally include at least three views.
7. Your GitHub repo must include a README.md with an outline of the project including:

  - An overview of the project and its purpose
  - Instructions on how to use and interact with the project
  - At least one paragraph summarizing efforts for ethical considerations made in the project
  - References for the data source(s)
  - References for any code used that is not your own

Data Engineering Track

For this track, your group will follow data engineering processes. Here are the specific requirements:

1. Data must be stored in a SQL or NoSQL database (PostgreSQL, MongoDB, SQLite, etc) and the database must include at least two tables (SQL) or collections (NoSQL).
2. The database must contain at least 100 records.
3. Your project must use ETL workflows to ingest data into the database (i.e. the data should not be exactly the same as the original source; it should have been transformed in some way).
4. Your project must include a method for reading data from the database and displaying it for future use, such as:
   - Pandas DataFrame
   - Flask API with JSON output
5. Your project must use one additional library not covered in class related to data engineering. Consider libraries for data streaming, cloud, data pipelines, or data validation.
6. Your GitHub repo must include a README.md with an outline of the project including:
   - An overview of the project and its purpose
   - Instructions on how to use and interact with the project
   - Documentation of the database used and why (e.g. benefits of SQL or NoSQL for this project)
   - ETL workflow with diagrams or ERD
   - At least one paragraph summarizing efforts for ethical considerations made in the project
   - References for the data source(s)
   - References for any code used that is not your own
7. OPTIONAL: add user-driven interaction, either before or after the ETL process. e.g.:
   - BEFORE: provide a menu of options for the user to narrow the range of data being extracted from a data source (e.g. API or CSV file, where fields are known in advance).
   - AFTER: Once the data is stored in the database, add user capability to extract filtered data from the database prior to loading it in a Pandas DataFrame or a JSON output from a Flask API.

For this project, you can focus your efforts within a specific industry, as detailed in the following examples.

Finance

Tracking market data is crucial for equity traders. Not all traders code and are able to create custom-tailored visualizations. What's the best way for them to get what they need for success?

One option is offered by the **Wall Street Journal** **(https://www.wsj.com/market-data)** . Their website offers a dashboarding tool providing a high-level view of market performance.

This highly interactive tool allows users to easily explore stocks, bonds, currencies, and commodities.

- Users of all skill levels can use the data.
- Visualizations help make the data easier to understand.
- Multiple views are available for customized content.

Healthcare

Imagine: Vacation time is coming up, and so is flu season. Trying to plan a road trip across the United States while keeping everyone's health in mind can be tricky.

Using the FluView dashboard provided by the CDC, users can easily confirm which areas to avoid.

Different interactive features include:

- An overall view of the United States, or customizable view (state by state)
- Historic and current cases
- A chart showing the count of cases, broken down by strain

With this, data are delivered quickly and navigated through with ease.

Custom

We've only specified healthcare and finance, but any industry can benefit from data visualization. Consider the following example of weather tracking.

While on the way to work one morning, you notice dark clouds on the horizon. You don't remember hearing about a storm coming in, but this looks ominous.

A quick visit to Weather Underground's Dashboard helps illuminate the situation.

Updated with live data, you can view a live map as well as specific conditions such as temperature, pressure, and even feed from a live webcam.

The data delivery is up-to-date and seamless, making it easy to understand current conditions without digging too deeply.

Working with Your Group

When working on an online group project, it's crucial to meet with your group and communicate regularly. Plan for significant collaboration time outside of class. The following tips can help you make the most of your time:

- Decide how you're going to communicate with your group members when you begin. Create a Slack channel, exchange phone numbers, and ensure that the group knows each group member's available working hours.
- Set up an agile project by using **GitHub Projects** (https://docs.github.com/en/issues/planning-and-tracking-with-projects/learning-about-projects/quickstart-for-projects) so that your group can track tasks.
- Create internal milestones to ensure that your group is on track. Set due dates for these milestones so that you have a timeline for completing the project. Some of these milestones might include:
    - Project ideation
    - Data fetching/API integration
    - Data analysis
    - Testing
    - Creating documentation
    - Creating the presentation

Since this is a two-week project, make sure that you have done at least half of your project by the end of the first week in order to stay on track.

Although you will divide the work among the group members, it's essential to collaborate and communicate while working on different parts of the project. Be sure to check in with your teammates regularly and offer support.

Support and Resources

Your instructional team will provide support during classes and office hours. You will also have access to learning assistants and tutors to help you with topics as needed. Make sure to take advantage of these resources as you collaborate with your group on this first project.

Data Visualization Track Requirements (75 points)

Data and Delivery (20 points)

- The dataset contains at least 100 unique records. (5 points)
- A database is used to house the data (SQL, MongoDB, SQLite, etc.). (5 points)
- The GitHub repo has a README.md that includes the following: (10 points)
    - An overview of the project and its purpose
    - Instructions on how to use and interact with the project
    - At least one paragraph summarizing efforts for ethical considerations made in the project
    - References for the data source(s)
    - References for any code used that is not your own

Visualizations (25 points)

- A minimum of three unique views present the data. (10 points)
- The visualizations are presented in a clear, digestible manner. (5 points)
- The data story is easy to interpret for users of all levels. (10 points)

Usability (30 points)

- The script, notebook, or webpage created to showcase data visualizations runs without error. (10 points)
- A Python or JavaScript library not shown in class is used in the project. (10 points)
- The project includes some level of user-driven interaction, conforming to one of the following designs: (10 points)
  - HTML menus, dropdowns, and/or textboxes to display JavaScript-powered visualizations
  - Flask backend with interactive API routes that serve back Python or JavaScript created plots
  - Visualizations created from user-selected filtered data

Data Engineering Track Requirements (75 points)

Database Design (40 points)

- The project uses ETL workflows to ingest data into the database. (10 points)
- The original dataset(s) are transformed prior to storing it in the database. (5 points)
- A database is used to house the data (SQL, MongoDB, SQLite, etc.). (5 points)
- The database has at least two tables (SQL) or collections (NoSQL). (5 points)
- The project documents the choice of the database used and why. (5 points)
- The project includes documentation of the ETL workflow with diagrams or ERD. (10 points)

Data and Delivery (35 points)

- The database contains at least 100 unique records. (5 points)
- The project uses one additional library not covered in class related to data engineering. (10 points)
- The project includes a method for reading data from the database and displaying it for future use, such as: (10 points)
  - Pandas DataFrame
  - Flask API with JSON output
- The GitHub repo has a README.md that includes the following: (10 points)
  - An overview of the project and its purpose

  - Instructions on how to use and interact with the project
  - At least one paragraph summarizing efforts for ethical considerations made in the project
  - References for the data source(s)
  - References for any code used that is not your own

Both Track Requirements

Group Presentation (25 points)

- All group members speak during the presentation. (5 points)
- The content is relevant to the project. (5 points)
- The presentation maintains audience interest. (5 points)
- Content, transitions, and conclusions flow smoothly within any time restrictions. (10 points)

Project Guidelines

The following project guidelines focus on teamwork, your project proposal, data sources, and data cleanup and analysis.

Collaborating with Your Team

Remember that these projects are a group effort. The experience of close collaboration will create better project outcomes and help you in your future careers. Specifically, you'll learn collaborative workflows that will enable you to approach and solve complex problems. Working in groups allows you to work smart and dream big. Take advantage!

Project Proposal

Before you start writing any code, your group should outline the scope and purpose of your project. This will help provide direction and safeguard against **scope creep** (the tendency for projects to become more complex after work begins).

The proposal is essentially a brief summary of your interests and intent. Be sure to include the following details:

- The kind of data you'd like to work with and the field you're interested in (finance, healthcare surveys, etc.)
- The questions you'll ask of the data
- Possible source for the data

Use the following example for guidance:

"The aim of our project is to uncover patterns in credit card fraud. We'll examine relationships between transaction types and location, purchase prices and times of day, purchase trends over the course of a year, and other related relationships derived from the data."

Finding Data

Once your group has written a proposal, it's time to start searching for data. We recommend the following curated sources of high-quality data:

- **data.world** (https://www.data.world)
- **Kaggle** (https://www.kaggle.com)
- **Data.gov** (https://www.data.gov)
- **Awesome Public Datasets** (https://github.com/awesomedata/awesome-public-datasets)
- **Public-APIs** (https://github.com/n0shake/Public-APIs)
- **Awesome API** (https://github.com/Kikobeats/awesome-api)
- **Medium API List** (https://benjamin-libor.medium.com/a-curated-collection-of-over-150-apis-to-build-great-products-fdcfa0f361bc)

IMPORTANT

Whenever you use a dataset or create a new dataset based on other sources (such as existing datasets or information scraped from websites), make sure to use the following guidelines:

1. Check for copyright protections, and make sure that the way you plan to use this dataset is within the bounds of fair use.
2. Document how you intend to use this dataset now and in the future. Find any licenses or terms of use associated with the dataset, and review them to confirm that your intended use is in compliance.
3. Investigate how the dataset was collected. Identify any indicators that the data was obtained from a source that the compilers were not authorized to access.

You'll likely have to adjust your project plan as you explore the available data. That's okay! This is all part of the process. Just make sure that everyone in the group is aligned on the project's goals as you make changes.

Make sure that your datasets are not too large for your personal computer. Big datasets are difficult to manage locally, so consider using data subsets or different datasets altogether.

Data Cleanup and Analysis

Now that you've picked your data, it's time to tackle development and analysis. This is where the fun starts!

The analysis process can be broken into two broad phases: (1) exploration and cleanup, and (2) analysis.

As you've learned, you'll need to explore, clean, and reformat your data before you can begin answering your research questions. We recommend keeping track of these exploration and cleanup steps in a dedicated Jupyter notebook to keep you organized and make it easier to present your work later.

After you've cleaned your data and are ready to start crunching numbers, you should track your work in a Jupyter notebook dedicated specifically to analysis. We recommend focusing your analysis on multiple techniques, such as aggregation, correlation, comparison, summary statistics, sentiment analysis, and time-series analysis. Don't forget to include plots during both the exploration and analysis phases. Creating plots along the way can reveal insights and interesting trends in the data that you might not notice if you wait until you're preparing for your presentation. Presentation requirements will be further explained in the next module.

Presentation Day

It's crucial that you find time to rehearse before presentation day.

On the day of your presentation, each member of your group is required to submit the URL of your GitHub repository for grading.

NOTE

> Projects are requirements for graduation. While you are allowed to miss up to two Challenge assignments and still earn your certificate, projects cannot be skipped.