Final Project

Data Engineering

Goal of final project

- Develop a full data engineering pipeline on real-world data
- ETL
  - Extract data from an online source
  - Do transformations needed in python / pandas / pyspark / sql
  - Load into new SQL database
- Defined goal of what the transformed data look like
  - Weekly sales aggregates by region
  - Percentages of votes per hour
- All done in a colab notebook or local jupyter notebook
- A short video (5 min) presentation that describe what you have done.

Defining end goal of data

- You should have a clear end goal of what you want your data to look like
  - E.g. weekly measures of x, y, and z by region, state, etc
  - Measures should be clearly defined at the start of the notebook
- Some level of normalization to reduce redundancy
  - E.g. Customers and sales table
- Once you figure out this you can determine the steps you need to take with your raw data to get there
  - This ‘roadmap’ needs to be in the notebook
- **DON’T** just start with raw data and figure it out as you go!

Extract - data requirements

- Data don’t have to be huge
  - But minimum of a thousand rows
- Need to **integrate at least 2 sources of data**
- Data need to be individual observations
  - Single sale, single click, single call, etc
- Extract must be from *stable* online source
  - Can host on your google drive
  - Can put on your own Amazon S3 (must make public!)
  - Can be linked to *stable* source you don’t own
- Must provide link to data source in notebook

Data Collection

At least on of your sources of data must be collected from the data exchange techniques that we have discussed:

- RestAPI or it’s python interface like YelpAPI or Spotify
- Or extracting data from web-pages such as WikiPedia page

Data Collection Example

Assume you use want to study Computer Science Academic Scholars based on their influential citation provided by https://www.semanticscholar.org and then study the influence of institutions based on their scholars.

- here is an example: https://www.semanticscholar.org/author/Yoshua-Bengio/1751762

Semantic Scholar provides an API to collect any scholars information including its highly influential citation number.

However, you need the list of CS Academic Scholars and their institution

- which you can find in csrankings.csv here https://github.com/emeryberger/CSrankings

Data Collection Example - Cont

Now you have two sources of informaiton one from semantic scholars API and other from csrankings.csv.

For each scholars listed in csrankings you have to pull off the information from semantics scholar.

However, you will face data mismatch, inconsistency in the names, etc, that you have to handle in the data cleaning.

Extract - potential operations

- Joining CSV files
- Column names
- Separators
- Filling NA values
- All, some, other, or none depending on data and exploration
- **Can’t be pre-cleaned**
  - E.g. kaggle data

Transform

- Whatever steps are needed to make your data meet the end format.
- Some level of cleaning
- Filtering
- Aggregating
- Datatype conversions
- Etc
- **You must explore your data after import to both figure out what to do and then demonstrate you methods were successful.**

Load

- Data need to be in a SQL relational database
- You can use either local database or AWS like we did in class
- If you are using local MySQL, you have to submit all sql files such that we can reproduce the database.

Transform

- You need to normalize your data to some level
- E.g. user info should be in a different table than ones about actions each user made
- Need primary key to form relations

Your notebook

- This should be done fully in google colab or local jupyter
- Be sure to structure into sections
  - Add text cells with basic formatting
    - # = large text, ## medium text, ### small
- Code MUST be annotated
  - Use # in code blocks to describe goal of cell
- Exploration and checking must be included along with description of what your exploration told you
- We should be able to run start to finish without errors
  - Exception - we won't push data back to your database (but we'll check the code)
  - When you submit the notebook make sure that all cell have outputs

Where to get data?

- Data.gov - https://catalog.data.gov/dataset
- Amazon open data - https://registry.opendata.aws/
- Data.world - https://data.world/
- Github awesome open data - https://github.com/awesomedata/awesome-public-datasets
- Data is plural - https://docs.google.com/spreadsheets/d/1wZhPLMCHKJvwOkP4juclhjFgqIY8fQFMemwKL2c64vk/edit#gid=0
- WikiPedia
- Public Rest API: https://github.com/public-apis/public-apis

Where else to get data?

- Lots of API’s… spotify, nyt, twitter, etc
  - Make your own dataset and toss on S3
- Simple google searches
  - “Votes by state by batch csv”
  - “MLB individual statistics by game”
- Don’t be afraid of JSON format :)

Extra Credit Opportunity: Streamlit App

What is Streamlit?

- An open-source Python library for building interactive, web-based applications.
- Designed specifically for data scientists, engineers, and machine learning practitioners.
- No need for front-end programming knowledge—focus entirely on Python.

Why Use Streamlit in Data Engineering?

- Quick prototyping of data pipelines and visualization dashboards.
- Turn Python scripts into shareable web apps effortlessly.
- Integrates seamlessly with popular Python libraries like Pandas, NumPy, and Matplotlib.

Key Features

- Minimal code: Build apps in minutes.
- Live code reloading: See changes instantly while coding.
- Interactive widgets: Sliders, checkboxes, file uploads, and more.

How to get the extra credit:

- Create full working streamlit app for your project
- Create the app code within your project notebook
- Include the screenshots of the app in your report

Capabilities of Streamlit

https://docs.streamlit.io/develop/api-reference

Data Display

- Render tables and dataframes (`st.dataframe`, `st.table`).
- Display metrics (`st.metric`) for key performance indicators.

Visualization

- Plot with libraries like Matplotlib, Plotly, and Altair.
- Built-in support for static and interactive charts.

User Input

- Widgets for interactive filtering and parameter adjustments:
  - Sliders, text inputs, dropdowns, and date pickers.

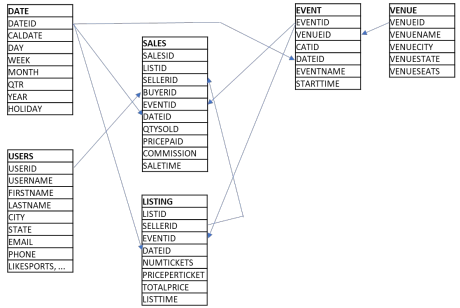
Example Use Cases in Data Engineering

- Interactive ETL (Extract, Transform, Load) pipelines.
- Monitoring dashboards for data pipelines and jobs.
- Visual data exploration for big datasets.

Report

You must submit A PDF report along with your Notebook that contains:

- Data
  - List of data sources you will use (links to them)
  - What data are available at each source
  - What data you used and how you integrated it
  - What was the data cleaning challenges
- RDB schema (full detailed schema of your database)
  - What tables does it contain
  - Relationships (keys) between tables
  - Want an actual diagram
- Queries and plots
  - Make sure you have couple of queries (at least 5) to show the use-cases of your data.
  - Add some simple plots to show trends/behavior, similar to what we did for movie ratings.



Video Presentation

The video must address:

- Motivation and goal
- Data collection
    - Challenges
- Data cleaning and transformation
- Data Schema
- Final queries and outcome
- Please create a separate presentation slides (do not go over your notebook in the presentation)

The video must be at most 5 minutes.

Submission

- Upload your video presentation to Youtube. It must be public, but you can change the visibility to accessible only via link
- Create directory: yourname_final_project
- What goes inside it:
  - Your notebook (.ipynb)
  - Your SQL files to replicate database if it is not in notebook
  - Your report
  - video.txt that includes the link to your video presentation
- Zip and submit