DSCI-560: Data Science Practicum Laboratory Assignment 6
Instructor: Young Cho, Ph.D.

Oil Wells Data Wrangling

This lab focuses on text extraction, web scraping, data preprocessing, and visualization from scanned PDF files. You assigned tasks include writing and running scripts that efficiently and effectively collect and organize data from PDFs and create a web interface to visualize the collected information. You will work with your team on pdf text extraction in this lab. Additionally, you'll preprocess the data to remove missing values, fetch additional data from the web, and store them in a database.

Part 1: Data Collection, Wrangling, and Munging

1) Initial Setup

You may use any available tools and scripts including OCRMYPDF, PyPDF, PyTesseract, requests/selenium, beautifulsoup4, mySQL database for this assignment.

For the assignment, you must use Python scripts and other useful tools in the Linux environment. (Document any setup steps/requirements for running your scripts in your submitted document).

Do not spend too much time on the installation and setup, and invest your time in exploring the concepts and improvising your submission.

2) Data Collection / Storage

For this task, you will focus on creating the database tables, parsing PDF files, and collecting information from given websites.

For this set of assignments, we would focus on information related to oil wells, such as their physical location and specifications, and create a webpage in Part 2 to plot this information on maps and visualize the collected data.

3) PDF Extraction

We will be using this Drive Folder for the Assignment

      https://drive.google.com/drive/u/4/folders/12g-bhOylyaMoLF5djocnAeZHBx-gsxgY

The above folder has different PDFs of scanned images of different oil wells and information/specifications related to them.

Download a copy of the folder to your local machine. Your task is to write a Python script to iterate over all the PDFs in the folder, extract the information from the PDFs, and store it in your database tables.

All PDFs will have well-specific information and stimulation data (how much proppant chemical was injected after drilling).

Operator: Oasis Petroleum LLC   Well Name: Kline Federal 5300 31-18 6B   API: 33-053-06057

Enseco Job#: S15072-02   Job Type: MWD D&I   County, State: McKenzie County, N. Dakota

Well Surface Hole Location (SHL): Lot 3, Sec. 18, T153N, R100W (2457' FSL & 238 FWL)

Latitude: 48° 04' 27.510 N   Longitude: 103° 36' 11.380 W   Datum: Nad 83

Figure 1: Relevant data include API#, longitude, latitude, well name & number, address, and any relevant fields.

Well Specific Stimulations

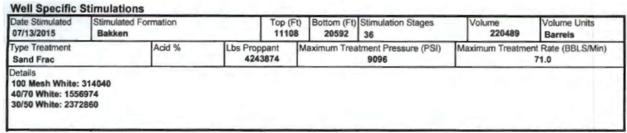| Date Stimulated 07/13/2015	Stimulated Formation Bakken		Top (Ft) 11108	Bottom (Ft) 20592	Stimulation Stages 36	Volume 220489	Volume Units Barrels
Type Treatment Sand Frac		Acid %	Lbs Proppant 4243874	Maximum Treatment Pressure (PSI) 9096		Maximum Treatment Rate (BBLS/Min) 71.0	
Details 100 Mesh White: 314040 40/70 White: 1556974 30/50 White: 2372860							

<div align="center">Figure 2: For Stimulation Data, extract all the fields mentioned in the snapshot above.</div>

4) Additional Web Scraped Information

In This part, we will use the API# and well name extracted above to gather additional information related to the wells from Internet sources.

Your task here is to iterate over each row of the database, and for each database entry, use the API# and well name to make a search query on this page: https://www.drillingedge.com/search

Once you get the search results, you must open the well page and gather information about the well status, type, closest city, and barrels of oil and gas produced.

The required fields you need to scrape are highlighted in yellow in the example snapshot below.
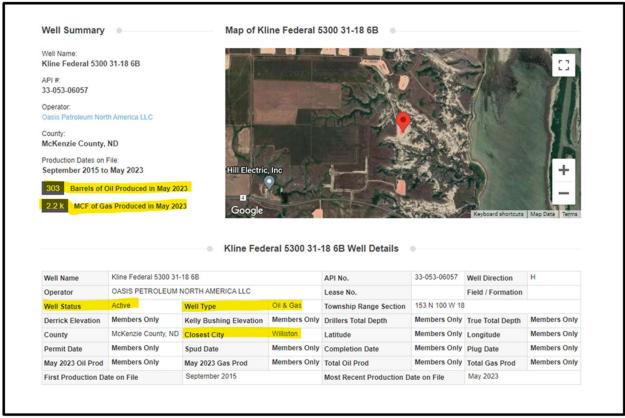

<div align="center">Figure 3: Search results on drillingedge.com</div>

The extracted information should be appended as additional fields to the existing entries in the database.

5) Data Preprocessing

Preprocess the data by removing HTML tags, special characters, and irrelevant information before storing it in the database. Transform the data into a suitable format for analysis, such as converting timestamps. Replace any missing data with 0 /or N/A for web-scraped information and text extracted from PDF files.

Part 2: Web Access and Visualization

This part focuses on providing you with experience in tools that allow you to present your results from Part 1. You will work with your team to associate and plot the data you extracted in the previous lab. You will create map layers to plot the position of the wells and represent the additional information we collected along with it.

1) Initial Setup

We will use web-related tools and platforms, as well as the MySQL database, for this assignment. Please use a Linux environment. (Make sure to document any setup steps/requirements for running your scripts in the document you submit)
Pace yourself so that you can explore the higher-level concepts to improve the quality of your submission.

2) Webpage and Mapping
For this task, you will focus on creating map layers to plot the wells and represent the additional information we collected along with it.
Set up an Apache web server and create a webpage with a section for the map to be displayed.
Integrate maps for your data using libraries (You may use any map APIs and platforms available, including OpenLayers, as mentioned in the sample links below, or consider alternatives like Leaflet, etc.)

>   https://openlayers.org/en/latest/examples/overlay.html

Configure the map to display a base map and set the initial view. Then, push pins (markers) will be added to the map to indicate the locations of the wells. Each push pin should correspond to a well location. This can be done differently, depending on the platform that you decide to use. Ensure that the coordinates (latitude and longitude) from your database are used to position the push pins accurately.
Implement a mechanism that displays a popup window with detailed information when a push pin is clicked.

>   https://openlayers.org/en/latest/examples/popup.html

The popup should display all the data related to the well, including well-specific information, well-specific simulation data, and any additional data collected from the website. Thoroughly test your web page to ensure the map, push pins, and popups work correctly. Make sure that the data is presented neatly and easily read within the popups. Document your code and provide instructions on how to use your web application.

References

- Extracting text from PDFs using PyPDF2: https://automatetheboringstuff.com/chapter13/
- OCRMYPDF: https://github.com/ocrmypdf/OCRmyPDF
- Understanding PyTesseract: https://nanonets.com/blog/ocr-with-tesseract/
- Introduction to OpenLayers: https://www.youtube.com/watch?v=DqzJ6pwSwWk
- Node.js for MySQL: https://www.w3schools.com/nodejs/nodejs_mysql.asp

Team Discussions

Your team is expected to meet in-person / virtually each day of the week and discuss the assignment progress & next steps. Document and compile minutes of all meetings in a separate file called 'meeting_notes_<team_name>.pdf'

Submission

Make one submission per team. Each team must submit all the code files for the working solution, a readme document containing information for running the code in PDF format, and a document that outlines the minutes of all team meetings in PDF format.

For the demo video submission, prepare one video per team that demonstrates the entire working solution, explains how the data tables were loaded, demonstrates query results, and talks about the design decisions made along with reasoning for the same. Also, include details about how your team preprocessed the data. Please include the team's name and the names of all members in the video.

There will be a 50% penalty for all late submissions.