**SI507 FINAL PROJECT PROPOSAL (REVISION)**

REASONS FOR REVISION:

I managed to set up an OAuth 2.0 authorization and after an initial exploration of the data I’d be able to pull from LinkedIn, I realized that in order to access users’ full profile (that I would need for my final project), I’d have to apply for extra authorization with LinkedIn. Unfortunately, I don’t believe that LinkedIn would grant me extra permissions for a school project; therefore, I have decided to move in a different direction with my project.

DATA SOURCES & CHALLENGE SCORE:

I am a firm believer in adopting pets from local animal shelters rather than buying a dog from a bredder. During my search for my current dog, I spent a lot of time on Petfinder and the Humane Society of Huron Valley’s website searching for the perfect dog to adopt. For my final project, I’d like to create a program that allows a user to search through Petfinder’s records of adoptable dogs.

My intention is first scrape Wikipedia’s “List of dog breeds” to generate a database of possible dog breeds that a user can search for. From there, using PetFinder’s API (which requires an API Key), my program will request records of adoptable dogs.

The challenge score for this revised project is meets the minimum of 8. I earn 4 points for scraping the single Wikipedia page and then an additional 4 points for requesting data from a web API that requires an API key.

DATA ACCESS AND STORAGE:

I will cache all the scraped and requested data as well as placed them in a database with the following tables:

* Dog Breeds
* Shelter Information
  + Name, Location
* Pet Records
  + Name, Breed ID, Shelter ID, Age ID, Sex ID, Size ID
* Age Information
* Sex Information
* Size Information

DATA PROCESSING & TESTING:

Most likely I’ll be creating/caching dictionaries about each pet which will then be used to populate my database.

I will create unit tests to verify that each cached record has the information I need to populate my data base and unit tests that will ensure that my SQL queries pull the correct information.

DATA PRESENTATION:

At the present moment, I intend to have to use Flask to display some tables of data that includes information about the pets they’ve queried about. Potentially, the pet records can be sorted by age or location and I might had some functionality that allows them to view the shelter locations via Plotly.

**SI507 FINAL PROJECT PROPOSAL**

DATA SOURCES & CHALLENGE SCORE:

As part of the Michigan Athletics Career Center, I’d like to create a program for the organization that allows a user to view professional information about former student-athlete alumni.

My intention is to extract data from the Linkedin API (<https://developer.linkedin.com/>)—which requires authentication with OAuth 2.0—for members who have listed the University of Michigan and “Student Athlete” (as one of their past positions) on their profile. From there, I intended to compiled the following information:

* Basic Profile:
  + id: The unique identifying value for the member
  + first-name: The member’s first name
  + last-name: The member’s last name
  + industry: The (current) industry the member belongs to
  + site-standard-profile-request: The URL to the member’s authenticated profiled on LinkedIn
* Position:
  + title: The title of the member’s current position
* Company:
  + name: The name of the company
* Education:
  + field-of-study: The field that the member studied in
  + degree: The degree received

Using each of the member’s LinkedIn profile URL, I will scrape their profile page for the company’s name and location. I will use that data to extract GPS coordinates from Google Places API (<https://developers.google.com/places/web-service/search>) to map my scatter plot in Plotly.

The challenge score is 10. I earn 6 points for using a Web API I haven’t used before that requires OAuth (LinkedIn) and 4 points for scraping a new single page (LinkedIn Profile).

DATA ACCESS & STORAGE:

I will cache all the requested data from LinkedIn as well as placed them in a database with the following tables:

* Alumni Information
  + AlumniID, First Name, Last Name, IndustryID, PositionID, CompanyID, LocationID
* Company
  + CompanyID, Name, LocationID, Latitude, Longitude
* Location
  + LocationID, City, State
* Position
  + PositionID, Name, CompanyID
* Industry
  + IndustryID, Name

DATA PROCESSING:

I’ll be creating dictionaries of information about each industry and nested in there is information about each member.

DATA PROCESSING:

I’ll be creating unit tests to verify that each member is an alumni of the University of Michigan; that my database is correctly constructed, and that the results I require for my data presentation are also correctly constructed and formatted.

DATA PRESENTATION:

1. The user will has two options:
   1. They can view two types of graphs:
      1. A pie chart that communicates how much of the total alumni in the database are in a specific industry
      2. A bar chart that conveys how much of the total alumni in the database earned a specific degree
   2. Enter the industry (from a list) that they are interested in.
2. The user will view a table that displays all the first and last name and current company of the student-athlete alumni.
3. The user has two options:
   1. If they click on a alumna/alumnus, it will lead the user to another table that displays the first and last name, position title, company name, and LinkedIn profile URL of that alumna/alumnus.
   2. If they choose “Map”, the program will display to the user a map scatter plot of all the company locations with alumni in that particular industry.

CONCERNS:

I am little anxious about using the LinkedIn API. I know that I can get location information about each company through the LinkedIn API but since I needed to add to my challenge score, I decided to scrape for the company’s name and location via the alumni’s profile page. I worry that since not all alumni have a public profile and that their profiles are only accessible if you are logged into LinkedIn, if I will even be able to access their profile page.

If such is the case, I plan on scrapping the above idea and maybe designing a program that scrapes data about the different sports teams from the mgoblue.com website, creates a database with information about each sport’s roster and schedule, and allowing the user to interact with that data.