DAV 6100 Information Architectures Final Project (Spring 2022)

Katz School Alumni Data Analytics in AWS

Group members: Atreish Ramlakhan, Mahlet Melese, Shichao Zhou, Yuxiao Shen

1. Motivation

The need for communicating and maintaining a secure exclusive and private social network with alumni exists for Yeshiva University’s Katz School of Science and Health. Since this is a relatively small yet prestigious school, we can create a system where students keep in contact with each other via a medium that only they can access. Their Linkedin accounts and current job title can be edited at any time, and we can infer their salary. It allows the students to maintain all the data related to their courses, groups they may be associated with career interests, skill sets, and potential internship opportunities with alumni. Using a connected set of information from Registrar, admissions, alumni, and LinkedIn datasets students can manage a profile and see their relationships. This can also help build a network between the current students and the alumni. By signaling their interests and identifying alumni with a trajectory that appeals to the student, they can model their experience on an alumni.

2. Data source: Alumnus dummy data, LinkedIn, and salary.com

Students dummy data

We randomly generated students data because we were unable retrieve data from Katz school due to privacy regulations.

In the students dummy data, we have 'Student ID', 'First Name', 'Last Name', 'Katz School Major', 'GPA', 'Graduation Semester', 'Country of Origin', 'Languages', 'Undergraduate Major', 'Age at Graduation', 'Years of Experience', 'Location', 'Job Decription', and 12 columns for each courses that students have taken.

Challenges:

We need to match the student's native language with student's country of origin. We randomly populates students native language according to their birth country in certain percentages to mimic the actually student's demographic. For example, we set 78% of our students in the US speaks English as their first language and 22% of our students in the US speaks Spanish as their first language.

We have list of core courses for Artificial Intelligence and Data Analytics and Visualization. And common electives for both of the majors. We randomly assigned their core courses according to their major along with electives. Later on we discovered put students courses taken in a single column is not friendly when we are creating Neptune, therefore we extracted them into 12 columns for each courses.

LinkedIn

We web scraped the data on our four teammates' LinkedIn profile. In the LinedIn dataset, we have 'Full Name', 'Location', 'Most Recent Company', 'Job Title', and 'Company Url'.

Challenges:

We discovered that for some reason LinkedIn changed their HTML structure which resulted in error when we tried to ran the code on Jupyter Notebook.

Salary.com

We utilized a web scraper to extract salary information from www.salary.com to enrich our students dataset. In the salary.com dataset, we have 'Title', 'Location', 'Description', and 5 columns of salary percentiles on 10%, 25%, 50% (medium), 75%, and 90%, respectively.
(Ref: https://github.com/israel-dryer/Salary-Dot-Com-Scraper)

3. S3 and AWS Lambda

3.1 S3

In the s3 bucket, we stored Lambda layers' packages and CSV files triggered through the AWS Lambda function.

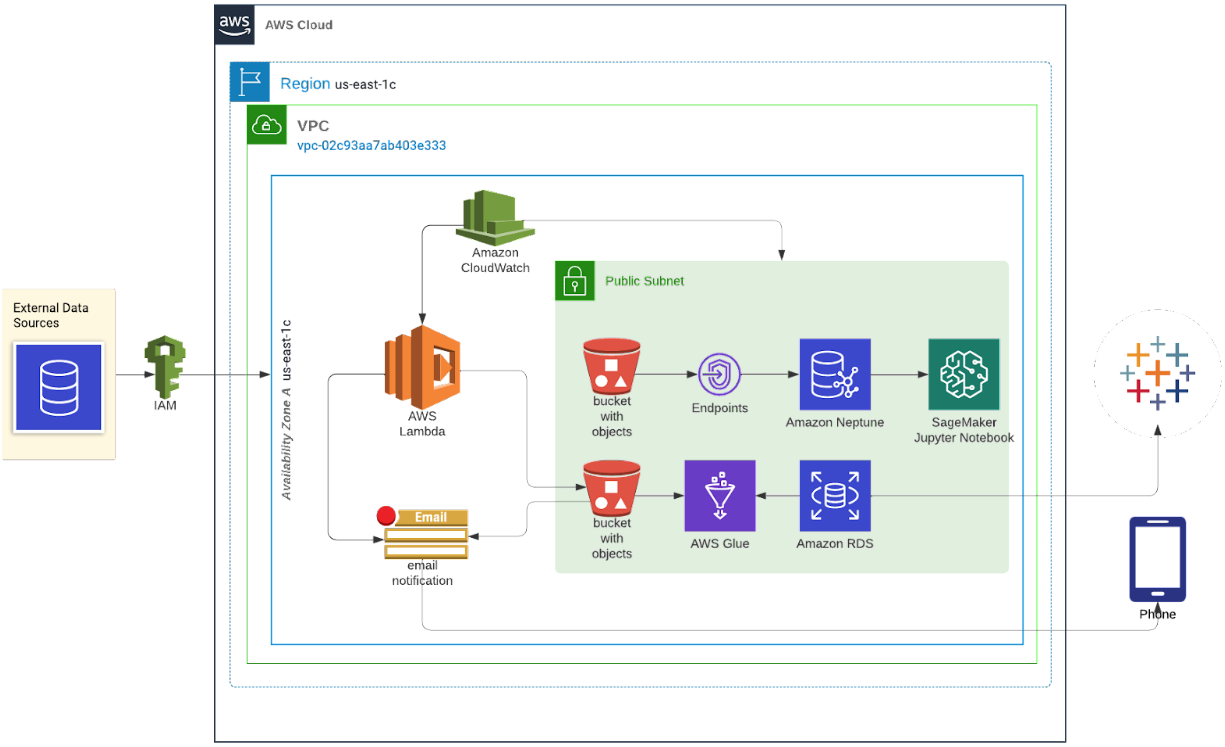
Challenges and Solutions:

- Timeout Limit: Generally, Lambda function invocation default time is 3 secs. So, we need to update the default time before triggering the lambda function. Also, the Lambda function can last up to 15 minutes. Though this will be more than enough, some long-running processes may require more extended periods.
- Package Size Limit: The deployment package can only have up to 250 MB. In our case, the capacity of the relevant selenium driver is too large to operate the script. So we modified the script and loaded it in memory during runtime execution.

- Incompatible Dependencies: AWS Lambda does not include Pandas/NumPy Python libraries by default. When we installed the dependencies in the source code directory and used Python 3.9 in the runtime setting, we got the error with NumPy-1.19 missing required NumPy dependencies. We updated the runtime setting from 3.9 to 3.8 then it worked.

  Lambda benefits:
- It makes our scripts more structured so that the code can be reused. In this way, our team's workflow can be more streamlined and efficient.

3.2 AWS Infrastructure



4. Neptune

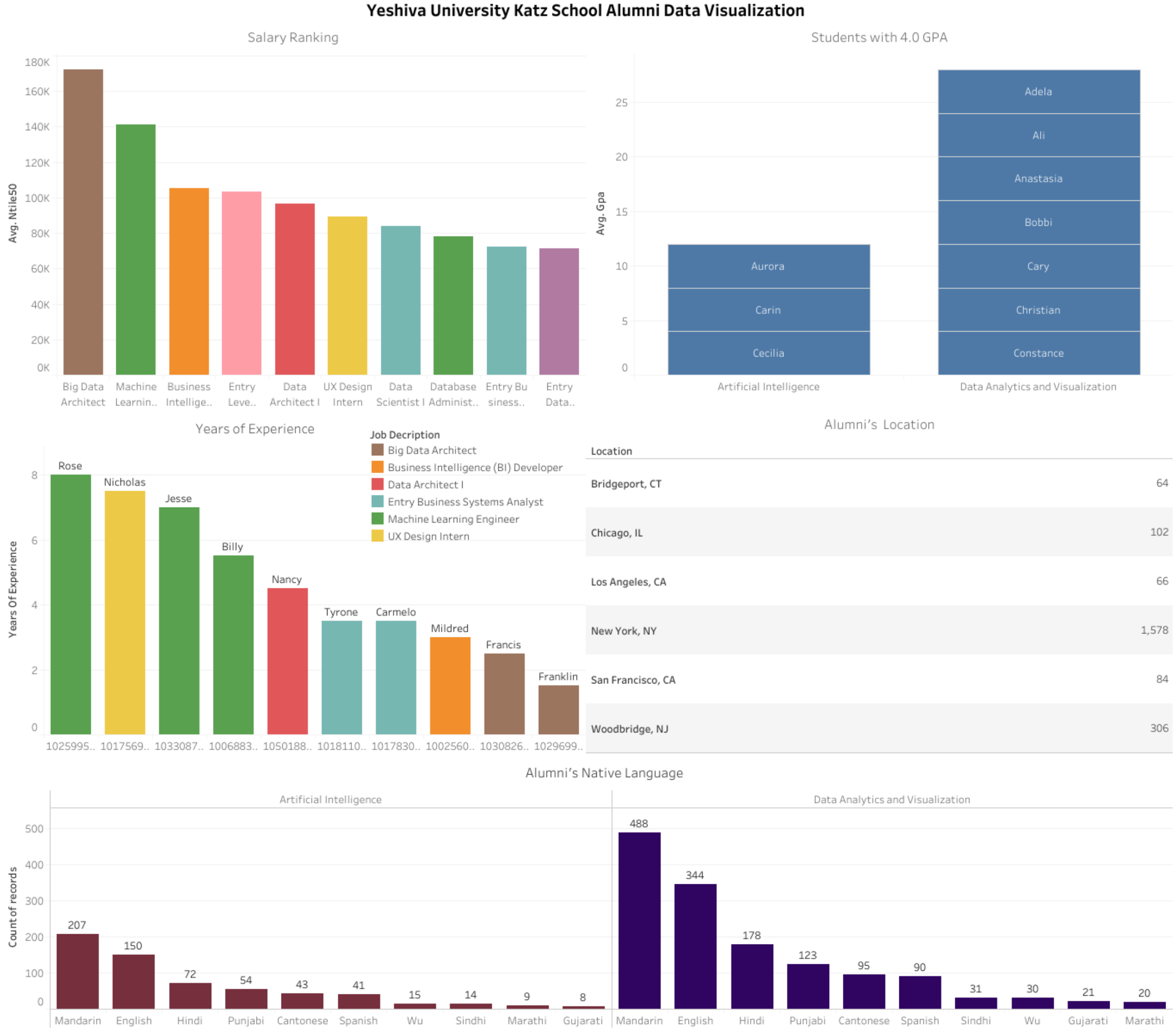
Graph databases, like Amazon Neptune, are purpose-built to store and navigate relationships. They have advantages over relational databases for use cases like social networking and recommendation engines where we need to create relationships between data and quickly query these relationships.

There are a number of challenges to building these types of applications using a relational database. We would need multiple tables with multiple foreign keys. SQL queries to navigate this data would require nested queries and complex joins that quickly become unwieldy, and the queries would not perform well as our data size grows over time.

Neptune uses graph structures such as nodes (data entities), edges (relationships), and properties to represent and store data. The relationships are stored as first order citizens of the data model. This allows data in nodes to be directly linked, dramatically improving the performance of queries that navigate relationships in the data. Neptune’s interactive performance at scale effectively enables a broad set of graph use cases.(Ref: https://docs.aws.amazon.com/neptune/latest/userguide/intro.html)
We describe an arbitrary domain as a connected graph of nodes and relationships with properties and labels. We formalize our entities a bit and match expected syntax for relationship types to create the node/relationship view for the property graph model.

Matrix - match node and relationship format of property graph mode

5. Visualization & Analysis in Tableau



Yeshiva University Katz School Alumni Data Visualization

Location	
Bridgeport, CT	64
Chicago, IL	102
Los Angeles, CA	66
New York, NY	1,578
San Francisco, CA	84
Woodbridge, NJ	306

6. Conclusion

We fully take advantage of the data and transform them into usefully insight about alumnus career, this also can better serve our prospective students.

7. References

Web scraping Salary.com https://github.com/israel-dryer/Salary-Dot-Com-Scraper