Data Analytics

Job Recommender Using kmeans Classifier

Ahmad Salim AHMADI

February, 2023

INTRODUCTION

While the overall job market is improving, it's crucial to acknowledge the impact of the COVID-19 pandemic. It caused a **disruptive shock** across various sectors, leading to temporary closures, job losses, and shifts in hiring trends. The overall job market has seen positive trends in recent months, with unemployment rates decreasing and job openings increasing across various sectors. However, it's important to note that this varies depending on the specific industry and location. The tech sector and certain niche areas continue to show strong growth, while other sectors are recovering at a slower pace.

Within the broader job market, the data-related field continues to thrive. The rising importance of data in business decision-making has fueled a significant demand for professionals who can collect, analyze, and interpret data to generate insights. This has led to a highly competitive job market for data professionals.

Finding the right data job can be tough. With so many options and things to consider, it's easy to feel lost. That's where this “job recommender” tool comes in! I've used kmeans classifier to help myself and others to find their perfect data job by matching their skills and goals to the best opportunities out there. Think of it like a smart friend who knows all about data jobs and can connect you with the right one in a flash.

As previously mentioned this tool uses KMeans classifier, which is sort of like a clever way to group things together based on how similar they are. In this case, it groups data jobs together based on skills and experience you put as input to the model. Then, it matches your profile to the most fitting group of jobs, showing you the ones that are most likely to be a good fit for you.

In this report, we'll delve deeper into the technical aspects of the KMeans classifier, analyze its effectiveness in recommending data jobs, and discuss potential future developments. This report will include the whole process

So, instead of spending hours searching through hundreds of listings, you can use this tool to quickly find the data jobs that are perfect for you!

Data Sources

1- Web Scraping

Data source for this project mainly relied on web scraping, with the primary data source being the prominent job portal www.indeed.com. Indeed stands out as a leading global job search platform that effectively connects millions of job seekers with a wide array of employment opportunities. With an impressive user base exceeding 300 million monthly visitors and strong collaborations with over 3 million employers, it serves as a main platform for job seekers worldwide. Users can easily locate jobs based on titles, companies, locations, or skill sets, and refine their search using filters such as industry, salary range, company size, and more.

Initially, I attempted to extract data using Python's BeautifulSoup library. However, it became apparent that www.indeed.com had implemented measures to prevent any form of automated login. Despite exploring various workarounds, my efforts proved futile. As a result, I turned to the Selenium library, which allowed me to successfully retrieve the necessary data for my model.

Data i scraped from indeed consisted:

- Job title
- URL
- Job ID
- Hiring Company
- Location
- Posted Date
- Salary

	job_title	url	job_id	company	location	additional_info	posted_date	full_url	salary
0	Board Certified Behavior Analyst (FT BCBA)	https://www.indeed.com/rc/clk?jk=ebd87097db3af...	job_ebd87097db3afc8e	Autism Spectrum Therapies	Pomona, CA 91766	Board Certified Behavior Analyst (FT BCBA)	Posted\nJust posted	https://www.indeed.com/rc/clk?jk=ebd87097db3af...	71,250–82,000 a year
1	Data Analyst - Health, Principal	https://www.indeed.com/rc/clk?jk=96b9457e9bc11...	job_96b9457e9bc1131e	Blue Shield of California	Woodland Hills, CA 91367	Data Analyst - Health, Principal	Posted\nToday	https://www.indeed.com/rc/clk?jk=96b9457e9bc11...	136,400–204,600 a year
2	nCino Business Analyst Senior	https://www.indeed.com/rc/clk?jk=f1bee190e67cb...	job_f1bee190e67cbf0f	City National Bank	Los Angeles, CA 90071	nCino Business Analyst Senior	Posted\nToday	https://www.indeed.com/rc/clk?jk=f1bee190e67cb...	92,114–156,880 a year

In an effort to collect job descriptions along with previous code (snapshot mentioned above), I encountered numerous instances of missing values during the initial scraping process. To address this issue, I had to extract job descriptions separately and in a separate attempt. Consequently, my second scraping endeavor involved:

- Job ID
- Job Descriptions

	job_id	job_description
0	job_96b9457e9bc1131e	JOB DESCRIPTION\nYour Role\nThe Network Perfor...
1	job_f1bee190e67cbf0f	Overview:\nNCINO BUSINESS ANALYST SENIOR\n\nWH...
2	job_ebd87097db3afc8e	Overview:\nWe're looking for...\n\nBright,\n\n...

2- API

In the API data collection part I tried to fetch some statistical data related to job market from the world bank website. The World Bank Public API unlocks a treasure trove of data, empowering you to explore and analyze various global trends and indicators. However its open for everyone but specially below categories benefit from it extensively:

- **Researchers and analysts:** Gain deeper insights for data-driven decision-making and research projects.
- **Data journalists and storytellers:** Craft compelling narratives using global data visualizations.
- **Students and curious minds:** Expand your knowledge and understanding of the world through interactive exploration.

- **Developers and innovators:** Build data-driven applications and tools with ready-to-use indicators.

. The statistical data obtained from the World Bank website through an API illustrates the trajectory of the unemployment rate in the United States. Notably, there was a surge in unemployment in 2020, likely attributed to the effects of the Covid-19 pandemic. Subsequently, the unemployment rate gradually declined after the peak of the pandemic. During the process of gathering statistical data from the World Bank website, I employed a World Bank API wrapper, simplifying and streamlining the task.

date	Unemployment, total (% of total labor force) (USA)
2022	2.321
2021	3.576
2020	5.591
2019	2.380
2018	2.450
2017	2.686
2016	2.908
2015	3.032
2014	3.691
2013	4.322
2012	4.741
2011	5.136

Data Cleaning and Exploratory Data Analysis

As previously mentioned, the primary data source involves web scraping, introducing challenges such as inconsistencies and noise in the collected data. To address this, separate scraping codes are executed for each job title and location (state/city), and the results are then concatenated. The snapshot below illustrates the combined data gathered for job titles Data Analyst, Data Engineer, and Data Scientist in Los Angeles.

```
# LOS ANGELES DATA SETS

dfladajd = pd.read_csv('../ironhack_final_project/los_angeles_da_jd.csv')
dflada = pd.read_csv('../ironhack_final_project/los_angeles_da.csv')

dfladejd = pd.read_csv('../ironhack_final_project/los_angeles_de_jd.csv')
dflade = pd.read_csv('../ironhack_final_project/los_angeles_de.csv')

dfladsjd = pd.read_csv('../ironhack_final_project/los_angeles_ds_jd.csv')
dflads = pd.read_csv('../ironhack_final_project/los_angeles_ds.csv')

losAngeles = pd.concat([dflada, dflade, dflads])
losAngeles_jd = pd.concat([dfladajd, dfladejd, dfladsjd])


losAngeles.shape, losAngeles_jd.shape
```

```
((906, 9), (804, 2))
```

Before inserting our data into the database, I had to go through the following data cleaning steps to ensure its usability for further analysis:

- Handling duplicates
- Handling null values
- Handle date formats
- Filtering values
- Renaming columns
- Creating columns
- Dropping columns when necessary
- Dropping rows when necessary
- Verifying that the datatypes in the DataFrame were accurate, if not, handle conversion
- String formatting (strip, replace methods)
- Concatenation for joining multiple DataFrames
- Indexing

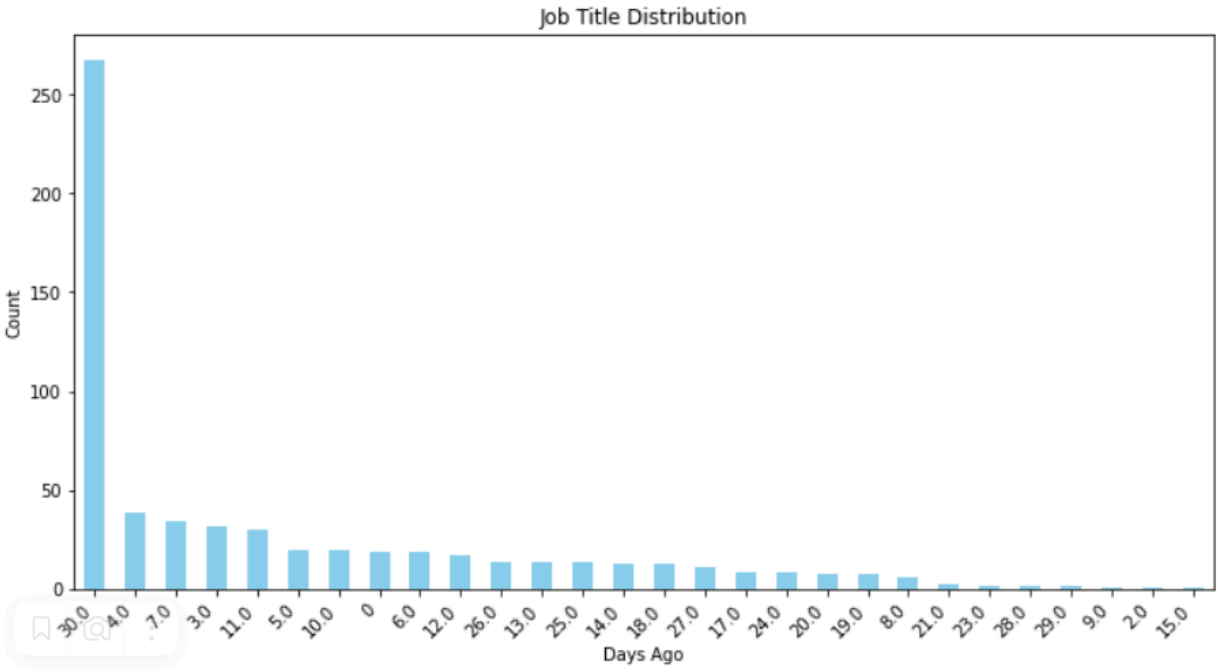
As seen below, our dataset for Los Angeles underwent the following changes after the cleaning process:

```
# Convert posted 15 days ago to: 15
losAngeles['days_ago'] = losAngeles['posted_date'].str.extract(r'(\d+)').astype(float)

# Display the DataFrame with the new 'days_ago' column
losAngeles[['posted_date', 'days_ago']].head(20)

# Convert those posted today to '0'
losAngeles['days_ago'] = losAngeles['days_ago'].fillna('0')
losAngeles['days_ago'].isna().sum()
losAngeles.shape
```

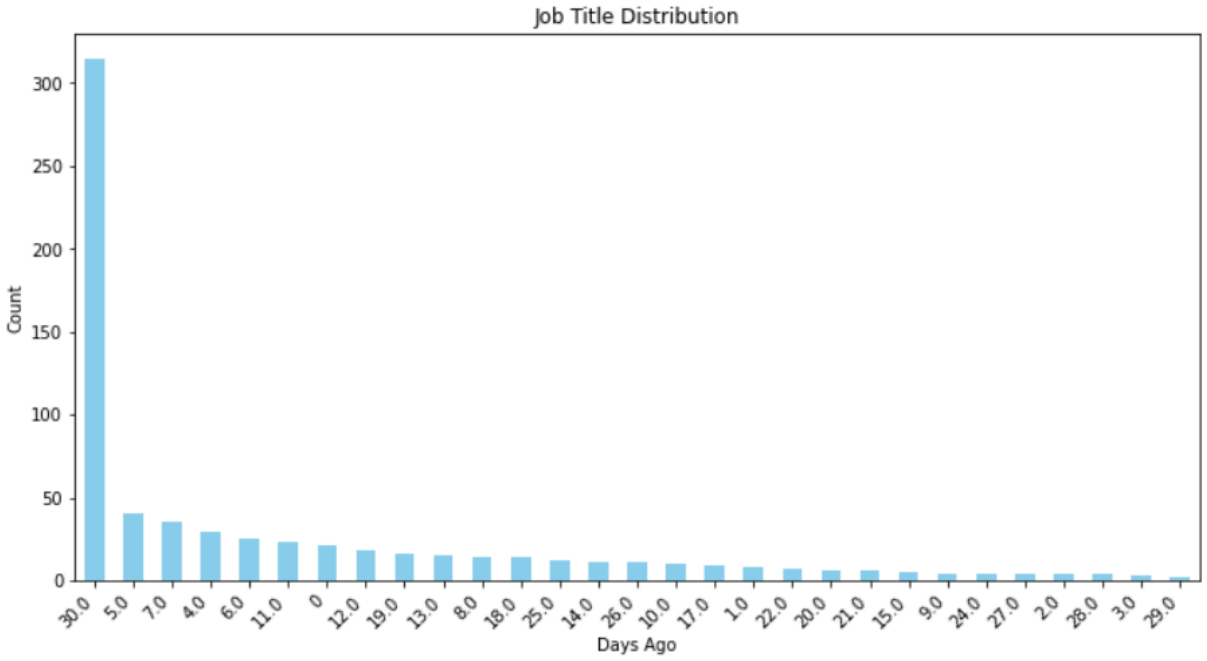
```
(628, 11)
```

For the initial phase of Exploratory Data Analysis, we have conducted some preliminary analyses. The primary analysis is scheduled for tomorrow, incorporating data from additional states that are currently being scraped. Presently, the dataset for Los Angeles and Ohio states are prepared, while datasets for Arizona and other states are still undergoing processing. Focusing on the available Los Angeles and Ohio datasets, the following chart illustrates the distribution of jobs based on the number of days (ago) they were posted.

Job distribution for Los Angeles

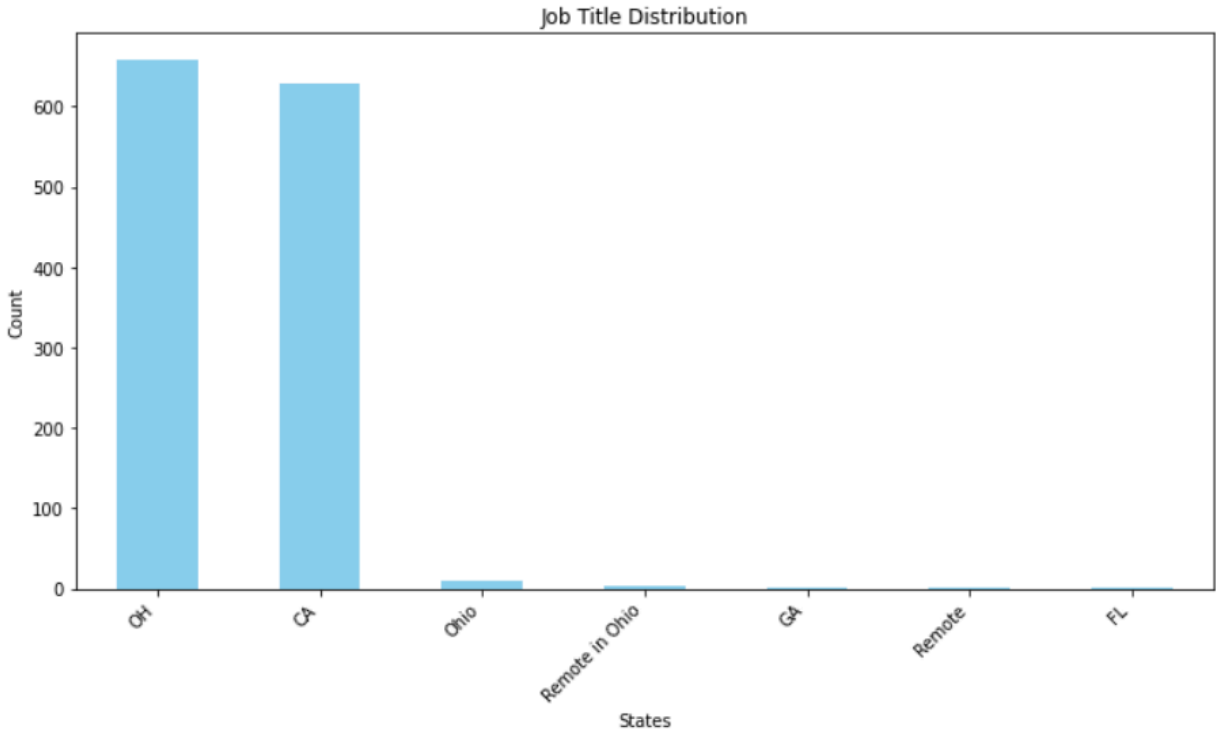


Job distribution for Ohio



The bar plot below indicates that there are still instances where states are represented in different formats than their standard abbreviations, such as LA and OH. This implies that these outliers need to be converted into the correct format.

Job Distribution by State



Our primary dataset, which will be used for our model, is the second dataset containing job IDs and job descriptions. Job descriptions encompass a wealth of information about the posted job, including the required skills. After cleaning our second dataset for Los Angeles and Ohio, we now possess a refined dataset with over 1200 records. Focusing on the "DATA" field, I've extracted the necessary skills from each data-related job's description. This skill data will serve as the foundation for training my model, allowing it to recommend jobs based on input skills.

Below are the most important skills that are usually needed for a position as Data Engineer, Data Analyst or Data Scientist. So I have fetched these skills from job description part of my data set and converted them into one hot encoding as below:

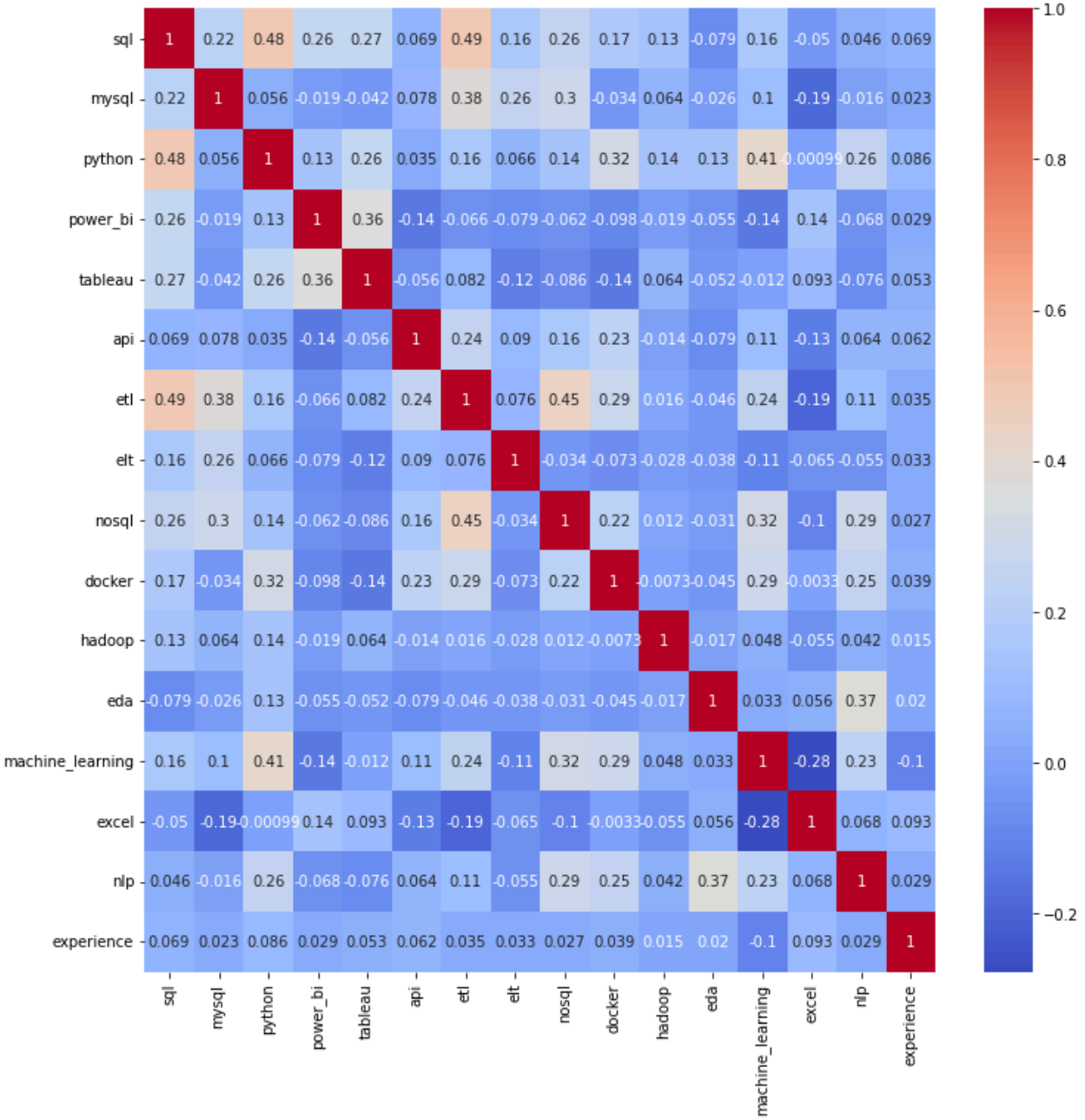
- SQL
- MySQL
- Python
- Power BI
- Tableau
- API
- ETL
- ELT
- NoSQL
- Docker
- Hadoop
- EDA (Exploratory Data Analysis)
- Machine Learning
- Excel
- NLP (Natural Language Processing)
- Experience

	job_id	job_description	sql	mysql	python	power_bi	tableau	api	etl	elt	nosql	docker	hadoop	eda	machine_learning	excel
0	job_96b9457e9bc1131e	JOB DESCRIPTION\nYour Role\nThe Network Perfor...	0	0	0	0	1	0	0	0	0	0	0	0	0	1
1	job_f1bee190e67cbf0f	Overview:\nNCINO BUSINESS ANALYST SENIOR\n\nWH...	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	job_ebd87097db3afc8e	Overview:\nWe're looking for...\n\nBright,\n\n...	0	0	0	0	0	1	0	0	0	0	0	0	0	1

The provided snapshot represents the dataset for training my classifier. Using binary input, where skills are denoted as 0 or 1, the classifier will then suggest a selection of jobs along with their titles and URLs for potential applications.

I believe an analysis without a heatmap would be incomplete. The heatmap below illustrates the correlation between the skills extracted from job descriptions. Notably, there is a high positive correlation between SQL and Python, NoSQL and ETL, Python and Machine Learning, which aligns with expectations. Additionally, a high negative correlation is observed between Machine Learning and Excel, as well as ETL and Excel, which is consistent with logical expectations.

Correlation Heatmap



Database type Selection

Relational Database vs Non-relational Database:

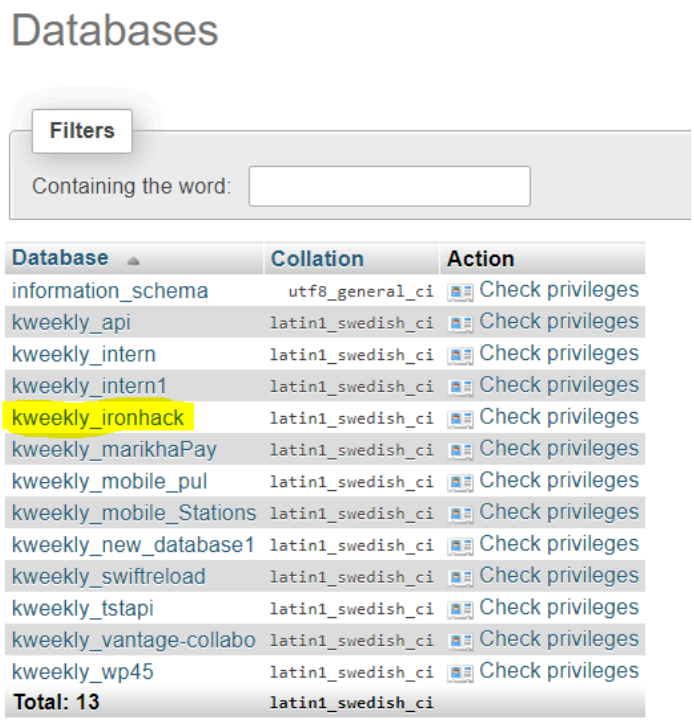
Like languages for data, SQL in relational databases is like a clear, structured dialect, perfect for precise queries and complex joins. No-SQL in non-relational databases offers diverse dialects, each suited to specific data types, making it ideal for exploring and managing unstructured information. Below table showcase the main difference between aforementioned databases.

Feature	Relational Database	Non-relational Database
Data Structure	Tables with rows and columns	Flexible, document-like structures
Schema	Strict and predefined	Less structured, data defines the schema
Querying	Uses SQL	Uses various non-SQL languages

Since my data is neatly organized into connected tables, like a well-structured spreadsheet, with clear relationships and defined connections (think: parent-child!), a relational database feels like the perfect fit. This kind of database will help me keep data tidy and consistent (no pesky duplicates!), let me easily play around with it, and most importantly, unlock the power of SQL. With SQL, I can ask super specific questions by connecting information from different tables, like a data detective with superpowers!

Database Creation

As illustrated below, I utilized the PhpMyAdmin portal available in my cloud server cPanel to create the database.



But for creating I opted for the command-line interface (CLI) to establish the tables on my server for data storage.

Below I created a table for storing job_id and job_description which is my main dataset for training my classifier.

```
mysql> CREATE TABLE job_description (
    -> job_id VARCHAR(255) PRIMARY KEY,
    -> job_description TEXT
    -> );
Query OK, 0 rows affected (0.02 sec)

mysql>
```

Table creation to store all other information related to job announcement.

```
mysql> CREATE TABLE job_announcement (
    -> job_id VARCHAR(255) PRIMARY KEY,
    -> job_title VARCHAR(255),
    -> url VARCHAR(255),
    -> company VARCHAR(255),
    -> location VARCHAR(255),
    -> additional_info VARCHAR(255),
    -> posted_date VARCHAR(255),
    -> full_url VARCHAR(255),
    -> job_details TEXT,
    -> days_ago INT(11)
    -> );
Query OK, 0 rows affected (0.02 sec)

mysql>
```

I also created a table to store my cleaned and one hot encoded data as below

```
mysql> CREATE TABLE job_skills (
    ->     job_id VARCHAR(255) PRIMARY KEY,
    ->     sql_skill INT,
    ->     mysql INT,
    ->     python INT,
    ->     power_bi INT,
    ->     tableau INT,
    ->     api INT,
    ->     etl INT,
    ->     elt INT,
    ->     nosql INT,
    ->     docker INT,
    ->     hadoop INT,
    ->     eda INT,
    ->     machine_learning INT,
    ->     excel INT,
    ->     nlp INT,
    ->     experience INT
    -> );
Query OK, 0 rows affected (0.02 sec)

mysql>
```

Finally, I established an additional table to store job_id and their respective states, facilitating data aggregation in subsequent analyses.

```
mysql> CREATE TABLE job_states (
    -> job_id varchar(255) PRIMARY KEY,
    -> state varchar(24)
    -> );
Query OK, 0 rows affected (0.02 sec)

mysql> show tables;
+----------------------------+
| Tables_in_kweekly_ironhack |
+----------------------------+
| job_announcement           |
| job_description            |
| job_skills                 |
| job_states                 |
+----------------------------+
4 rows in set (0.00 sec)

mysql>
```

Some SQL Queries

Fetch data from job_announcement table based on job_id:

INPUT

```sql
SELECT ja.job_title, ja.location, ja.days_ago, ja.url
FROM job_announcement ja
WHERE ja.job_id = 'job_25748514b21609ea'
ORDER BY ja.days_ago
```

OUTPUT

job_title	location	days_ago	url
Business Analyst, BizOps	Seattle, WA	0	https://www.indeed.com/rc/clk?jk=25748514b21609ea&...

Incase you want to check the skills needed for one specific job based on its job_id:

INPUT:

```sql
SELECT js.* FROM job_announcement ja
INNER JOIN job_skills js ON ja.job_id = js.job_id
WHERE ja.job_id = 'job_0059fe53d545b67f'
ORDER BY ja.days_ago
```

OUTPUT:

job_id	sql_skill	mysql	python	power_bi	tableau	api	etl	elt
job_0059fe53d545b67f	0	0	0	0	0	1	0	0

nosql	docker	hadoop	eda	machine_learning	excel	nlp	experience
0	*NULL*	0	0	0	1	*NULL*	1

Or incase you want to check skills of multiple job_id:

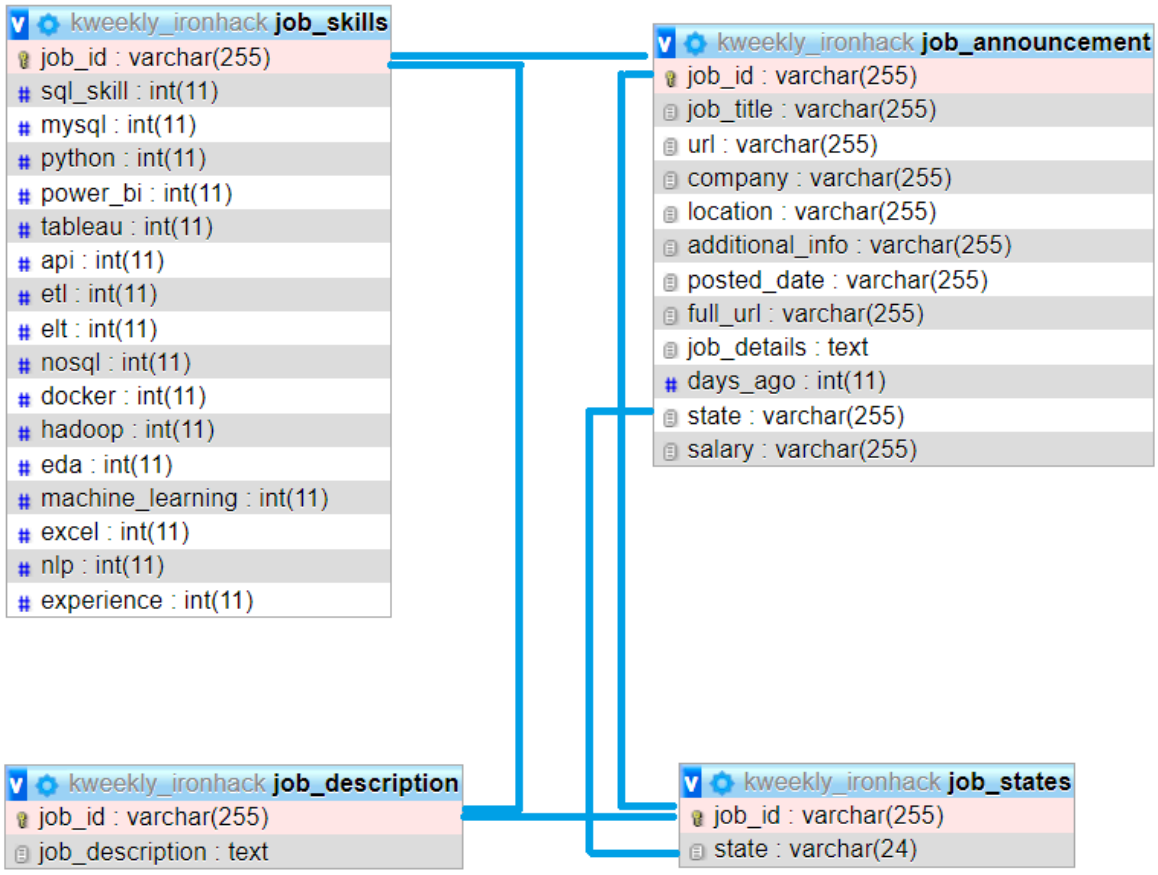
INPUT:

```sql
SELECT js.*
FROM (
    SELECT ja.job_id
    FROM job_announcement ja
    ORDER BY ja.days_ago
    LIMIT 10
) AS subquery
INNER JOIN job_announcement ja ON subquery.job_id = ja.job_id
INNER JOIN job_skills js ON ja.job_id = js.job_id
ORDER BY ja.days_ago;
```

OUTPUT:

job_id	sql_skill	mysql	python	power_bi	tableau	api	etl	elt	nosql	docker	hadoop	eda	machine_learning	excel	nlp	experience
job_0bed1b8854972d18	1	0	0	0	0	0	0	0	0	*NULL*	0	0	0	1	*NULL*	1
job_dee1fcafe0414654	1	0	1	1	1	0	0	0	0	*NULL*	0	0	0	1	*NULL*	1
job_a54a8da2ffdd899a	0	0	0	0	0	0	0	0	0	*NULL*	0	0	0	1	*NULL*	1
job_da1160728cb6150f	0	0	0	0	0	0	0	0	0	*NULL*	0	0	0	1	*NULL*	1
job_b914e766cb3ea6b6	0	0	0	0	0	0	0	0	0	*NULL*	0	0	0	1	*NULL*	1
job_ae91acbb904a0c41	1	0	0	0	1	0	0	0	0	*NULL*	0	0	0	1	*NULL*	1
job_52ec907574cd9f69	1	0	0	0	1	0	0	0	0	*NULL*	0	0	0	1	*NULL*	1
job_0876c18d798074a4	0	0	0	0	0	0	0	0	0	*NULL*	0	1	0	1	*NULL*	1
job_39201a50bde44eb3	1	0	0	0	1	0	0	0	0	*NULL*	0	0	0	1	*NULL*	1
job_a277ee11720e6e37	0	0	0	1	0	0	0	0	0	*NULL*	0	0	0	1	*NULL*	1

Entity Relationship Diagram (ERD)

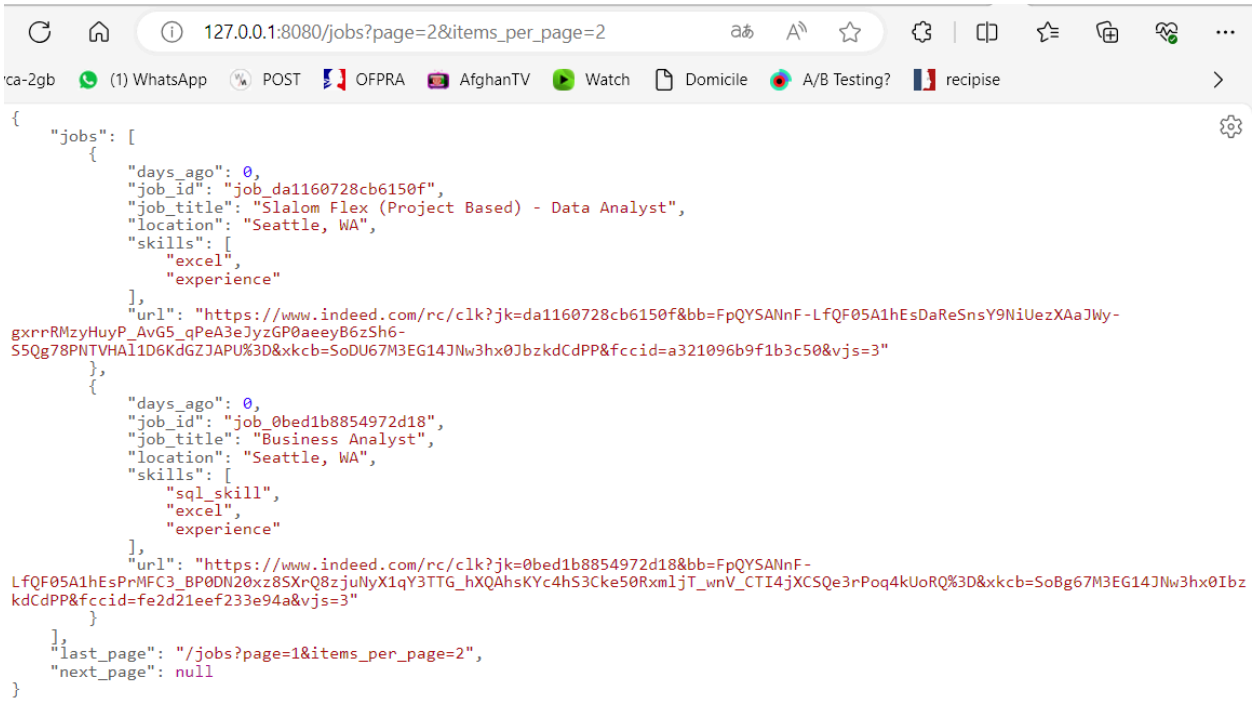


Exposing Data via API

Now its the time to expose these data using REST API. i have got 2 end points as below:

1. /jobs
2. /jobs/job_id

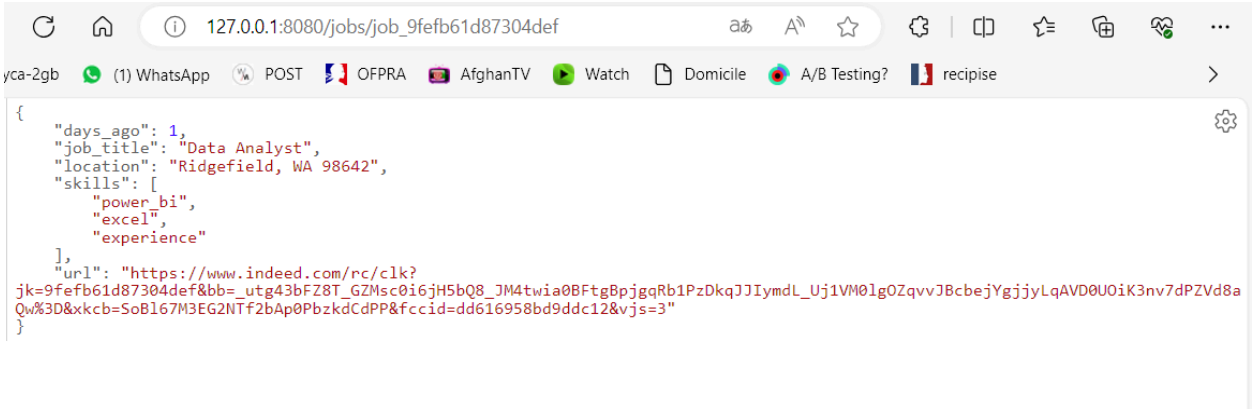
For the /jobs endpoint the data is exposed as follow:




```
{
    "jobs": [
        {
            "days_ago": 0,
            "job_id": "job_da1160728cb6150f",
            "job_title": "Slalom Flex (Project Based) - Data Analyst",
            "location": "Seattle, WA",
            "skills": [
                "excel",
                "experience"
            ],
            "url": "https://www.indeed.com/rc/clk?jk=da1160728cb6150f&bb=FpQYSANnF-LfQF05A1hEsDaReSnsY9NiUezXAaJWy-
gxrrRMzyHuyP_AvG5_qPeA3eJyzGP0aeeyB6zSh6-
S5Qg78PNTVHAl1D6KdGZJAPU%3D&xkcb=SoDU67M3EG14JNw3hx0JbzkdCdPP&fccid=a321096b9f1b3c50&vjs=3"
        },
        {
            "days_ago": 0,
            "job_id": "job_0bed1b8854972d18",
            "job_title": "Business Analyst",
            "location": "Seattle, WA",
            "skills": [
                "sql_skill",
                "excel",
                "experience"
            ],
            "url": "https://www.indeed.com/rc/clk?jk=0bed1b8854972d18&bb=FpQYSANnF-
LfQF05A1hEsPrMFC3_BP0DN20xz8SXrQ8zjuNyX1qY3TTG_hXQAhsKYc4hS3Cke50RxmljT_wnV_CTI4jXCSQe3rPoq4kUoRQ%3D&xkcb=SoBg67M3EG14JNw3hx0Ibz
kdCdPP&fccid=fe2d21eef233e94a&vjs=3"
        }
    ],
    "last_page": "/jobs?page=1&items_per_page=2",
    "next_page": null
}
```


As depicted above, you have the flexibility to specify the page number and the number of job items displayed per page in the URL. This presents job details such as title, ID, location, posting date, and required skills for each job.

However, if you wish to retrieve details for a specific job, you can utilize the second endpoint by providing the job_id, and the response will be as follows:




```
{
    "days_ago": 1,
    "job_title": "Data Analyst",
    "location": "Ridgefield, WA 98642",
    "skills": [
        "power_bi",
        "excel",
        "experience"
    ],
    "url": "https://www.indeed.com/rc/clk?
jk=9fefb61d87304def&bb=_utg43bFZ8T_GZMsc0i6jH5bQ8_JM4twia0BFtgBpjgqRb1PzDkqJJIymdL_Uj1VM0lgOZqvvJBcbejYgjjyLqAVD0UOiK3nv7dPZVd8a
Qw%3D&xkcb=SoBl67M3EG2NTf2bAp0PbzkdCdPP&fccid=dd616958bd9ddc12&vjs=3"
}
```