SPRING 2024 BIG DATA: CSGY 6513-D

- Name:

Jiaxin Dong, Chenchen Guo, Mohammed Zakriah Ibrahim

- NetID:

jd5226, cg4421, mi2471

Project Detail

This project proposes to utilize a comprehensive dataset containing 1.3 million job listings scraped from LinkedIn, augmented with detailed job skills information, to gain insights into the current job market trends, identify skill gaps, and develop a job recommendation system. The dataset, a rich source of information on job titles, industries, companies, and required skills, offers an unprecedented opportunity to analyze and address the needs of the modern workforce.

OBJECTIVE

- **Exploratory Data Analysis (EDA) on Job Market Data:** Perform a comprehensive exploratory data analysis on job market datasets to uncover underlying patterns, detect anomalies, and gain insights into the job market dynamics. This topic could cover visualization of data distributions, identification of key variables influencing job market trends, and preliminary assessments of data quality and structure.

OBJECTIVE

- **Industry and Job Title Trends**: Identify emerging trends in job titles and industries, spotlighting growth sectors and roles that are becoming more prevalent. This will help job seekers and educational institutions tailor their focus towards areas of future demand.
- **Skills Mapping**: Leverage the skills data within the dataset to map out the most in-demand skills across different sectors. This analysis will identify core competencies sought after by employers, facilitating a better alignment between job seekers' skill sets and market needs.
- **Job Title and Skills Relationship Exploration**: Investigate the relationship between job titles and required skills to uncover the specific competencies that are critical for success in various roles. This will help job seekers focus their skill development efforts more effectively.

OBJECTIVE

- **Job Market Analysis:** Conduct a thorough analysis of the job market across various dimensions, including industries, job titles, geographical locations, and company types. This will provide a clear picture of the current employment landscape, highlighting areas of high demand and potential oversaturation.
- **Job Recommendation System Development**: Develop a sophisticated job recommendation system with NLP or any other machine learning algorithms that matches job seekers with potential job listings based on their profiles, previous experience, and skill sets. This system will aim to streamline the job search process and increase the chances of successful employment.

METHODOLOGY & TECHNOLOGY

Data Cleaning, Statistical Analysis and Search Algorithms:

- Python 3.10+
- Pandas
- PySpark
- Jupyter Notebook

Visualization and GUI:

- Matplotlib
- Seaborn
- Streamlit/Gradio

Data source:
1.3M Linkedin Jobs & skills
Dataset size: 2GB
Number of Record: 1,296k

1.3M Linkedin Jobs & Skills (2024)

Scraped Jobs from Linkedin. Augmented with Job Skills

Link:
https://www.kaggle.com/datasets/asaniczka/1-3m-linkedin-jobs-and-skills-2024/data

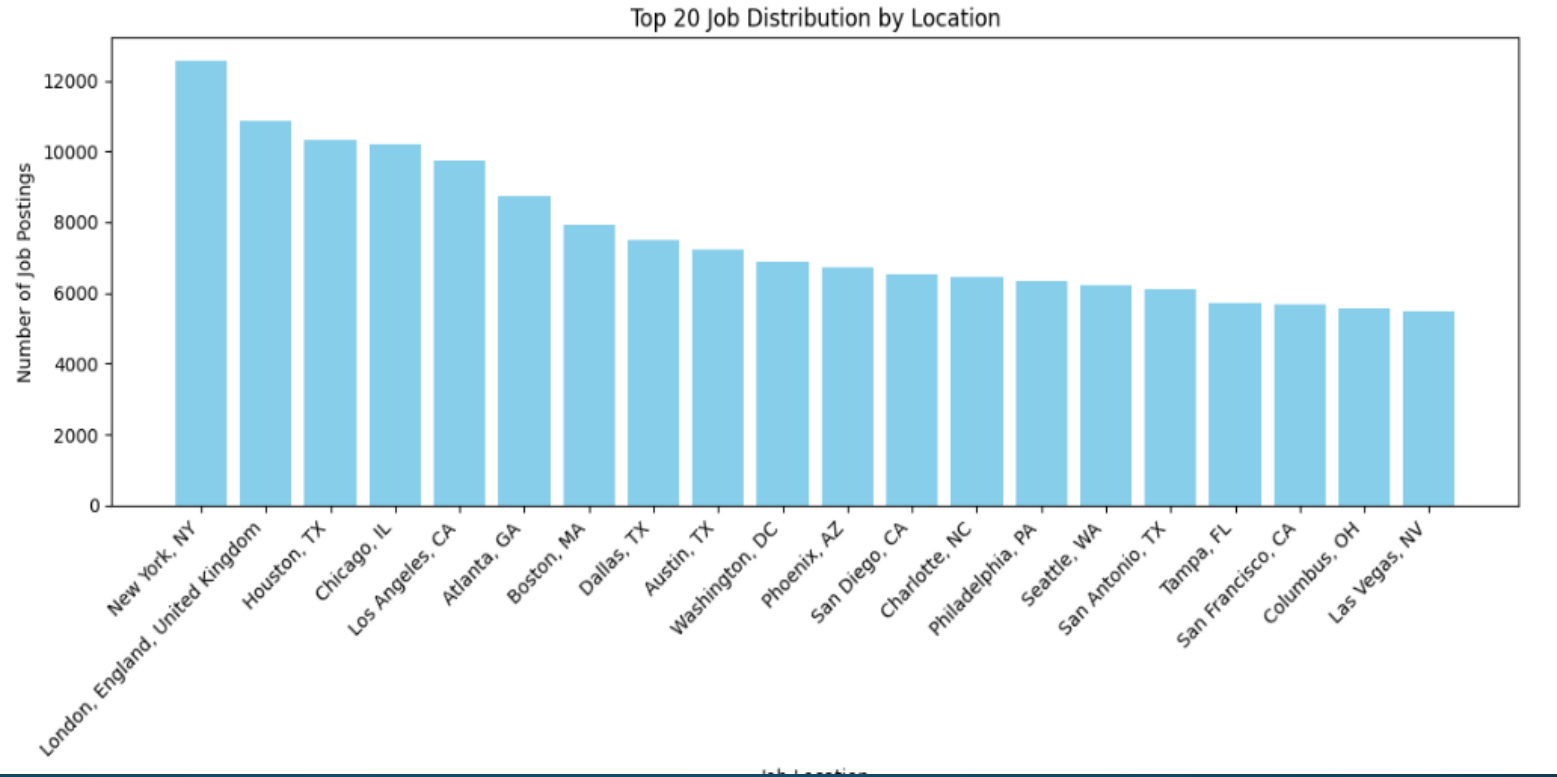
Link to project:
https://github.com/JD5226/BigData/tree/main/Project

PYSPARK

1、Job analysis of location distribution

job_location	count
New York, NY	12579
London, England, ...	10878
Houston, TX	10332
Chicago, IL	10187
Los Angeles, CA	9736
Atlanta, GA	8738
Boston, MA	7924
Dallas, TX	7514
Austin, TX	7235
Washington, DC	6869
Phoenix, AZ	6722
San Diego, CA	6532
Charlotte, NC	6470
Philadelphia, PA	6326
Seattle, WA	6235
San Antonio, TX	6102
Tampa, FL	5701
San Francisco, CA	5684
Columbus, OH	5552
Las Vegas, NV	5468

only showing top 20 rows

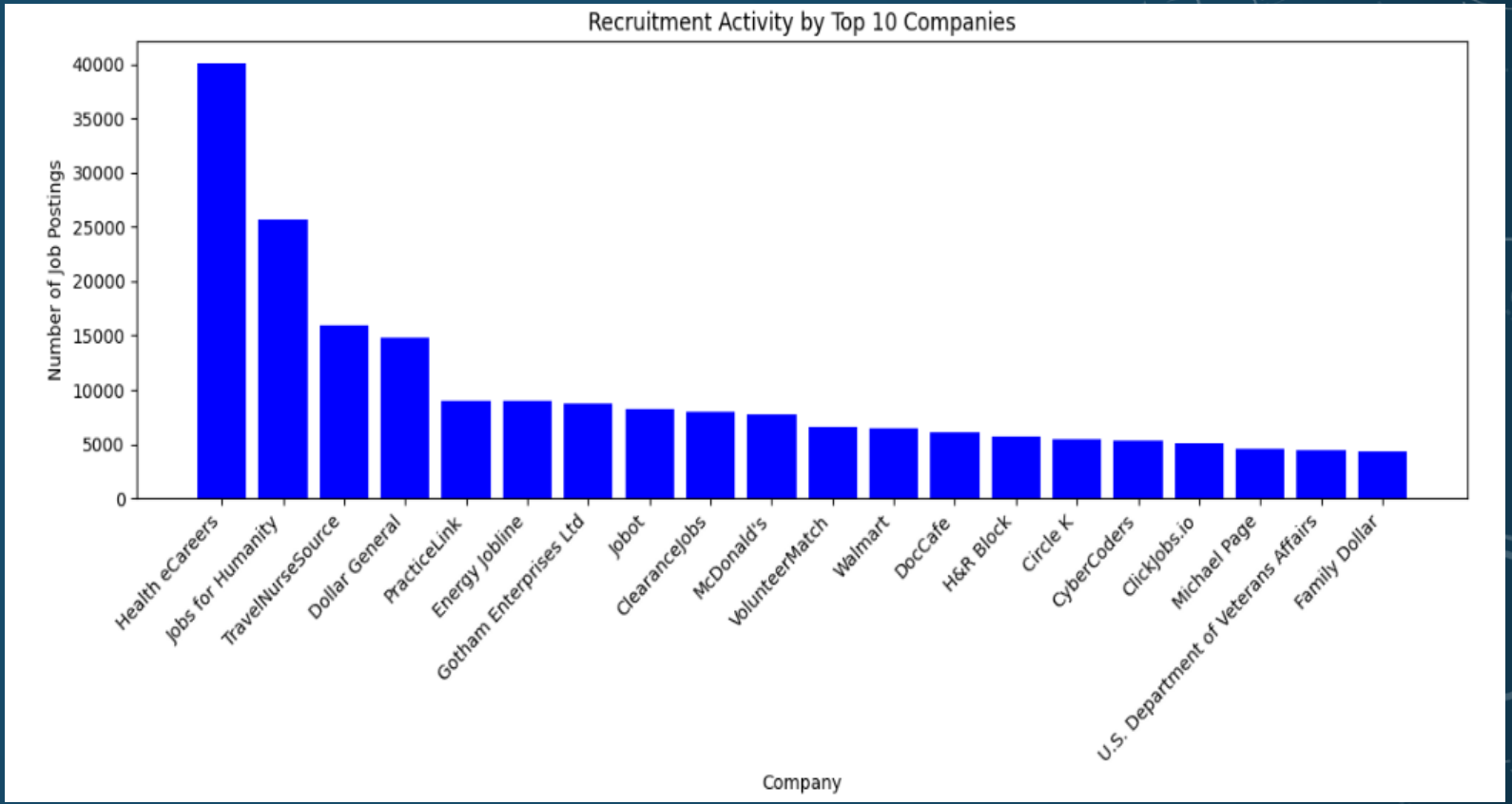


PYSPARK

2、Analysis of company recruitment activities:

company	count
Health eCareers	40047
Jobs for Humanity	25629
TravelNurseSource	15997
Dollar General	14775
PracticeLink	9043
Energy Jobline	8987
Gotham Enterprises Ltd	8700
Jobot	8264
ClearanceJobs	8015
McDonald's	7742
VolunteerMatch	6653
Walmart	6455
DocCafe	6026
H&R Block	5668
Circle K	5493
CyberCoders	5273
ClickJobs.io	5105
Michael Page	4560
U.S. Department of Veterans Affairs	4473
Family Dollar	4349

only showing top 20 rows

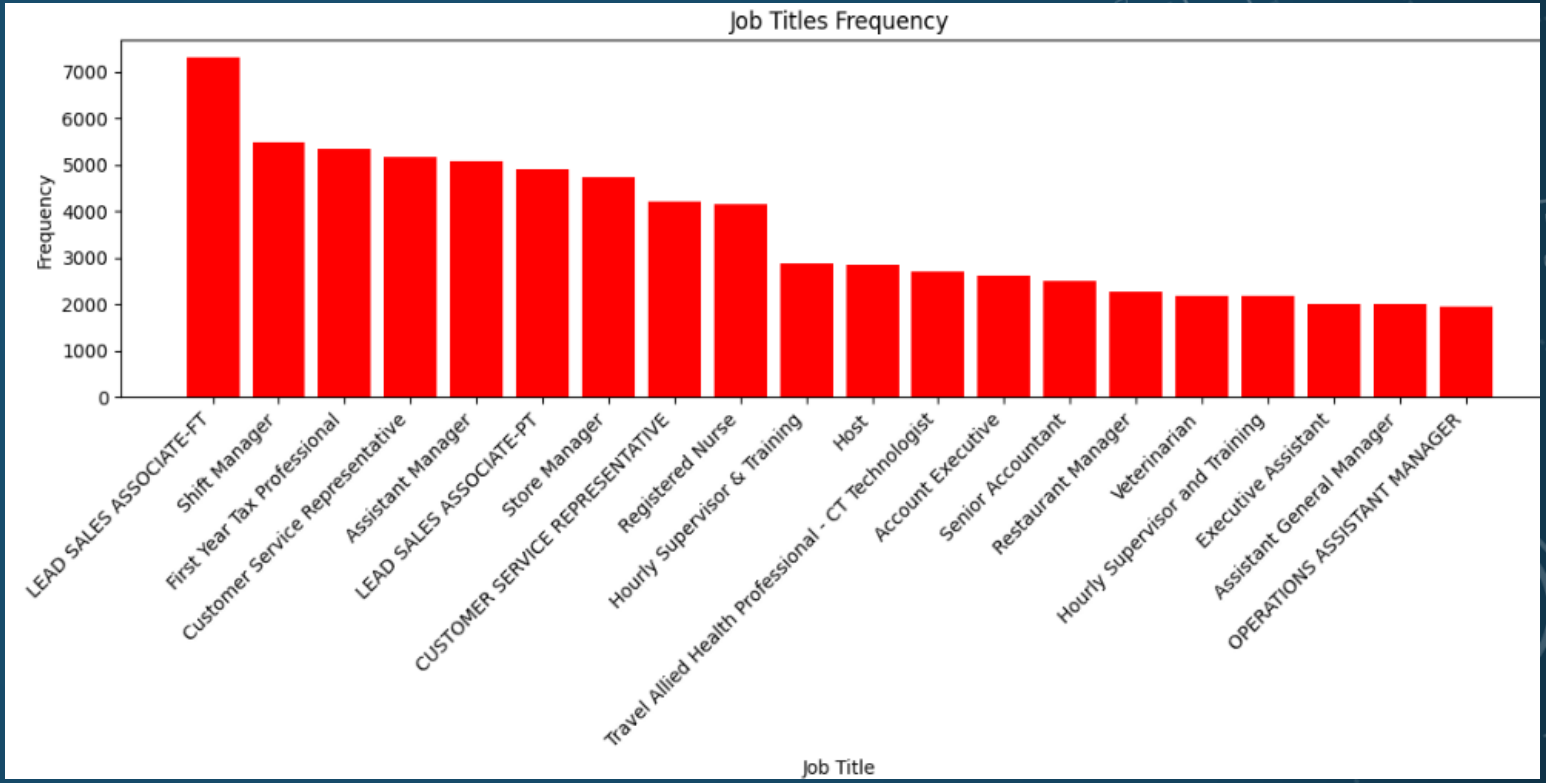


PYSPARK

3、Job Title Frequency Analysis:

job_title	count
LEAD SALES ASSOCIATE-FT	7315
Shift Manager	5500
First Year Tax Professional	5351
Customer Service Representative	5165
Assistant Manager	5067
LEAD SALES ASSOCIATE-PT	4911
Store Manager	4739
CUSTOMER SERVICE REPRESENTATIVE	4214
Registered Nurse	4142
Hourly Supervisor & Training	2883
Host	2861
Travel Allied Health Professional - CT Technologist	2717
Account Executive	2614
Senior Accountant	2497
Restaurant Manager	2280
Veterinarian	2194
Hourly Supervisor and Training	2179
Executive Assistant	2021
Assistant General Manager	1998
OPERATIONS ASSISTANT MANAGER	1960

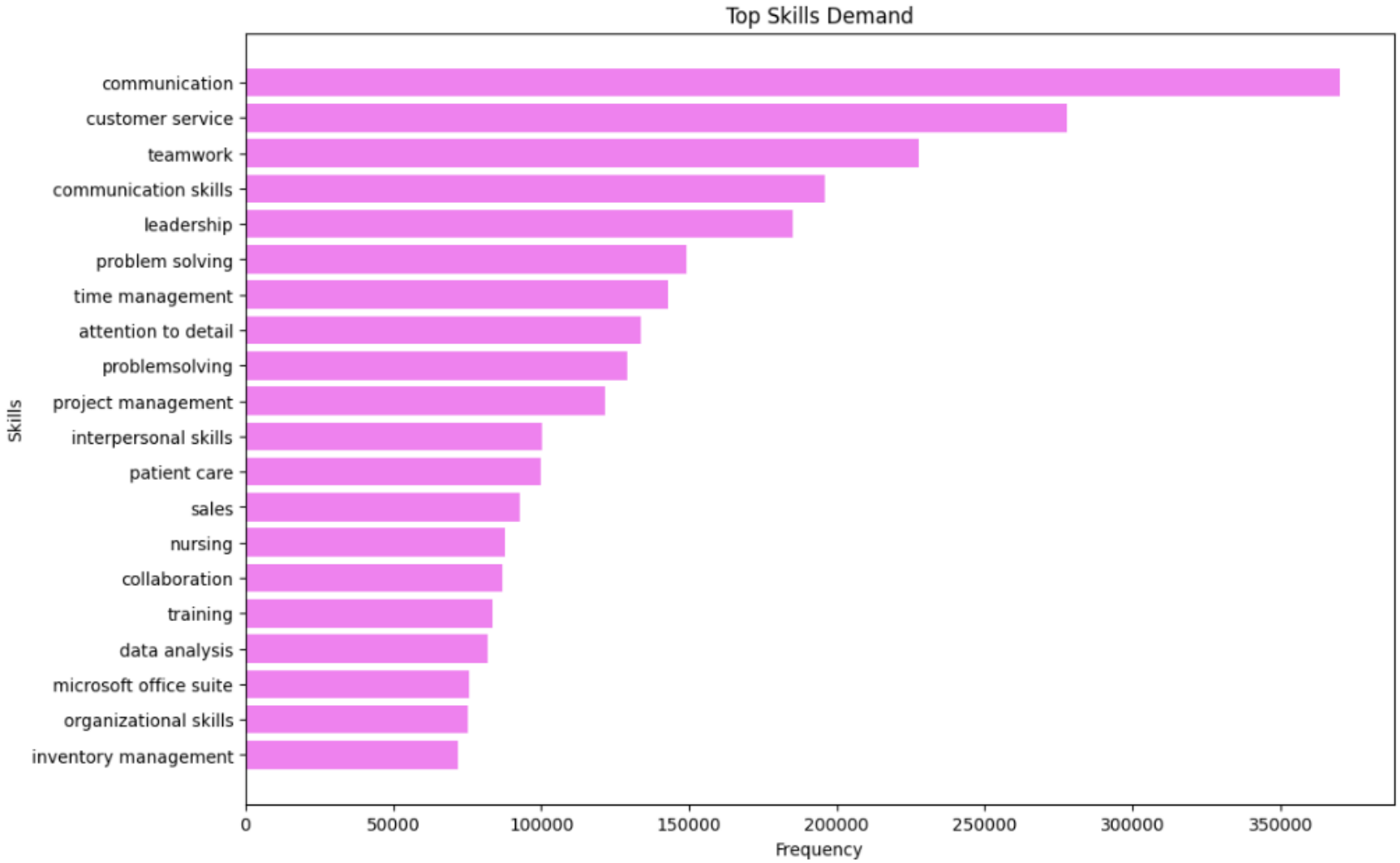
only showing top 20 rows



PYSPARK

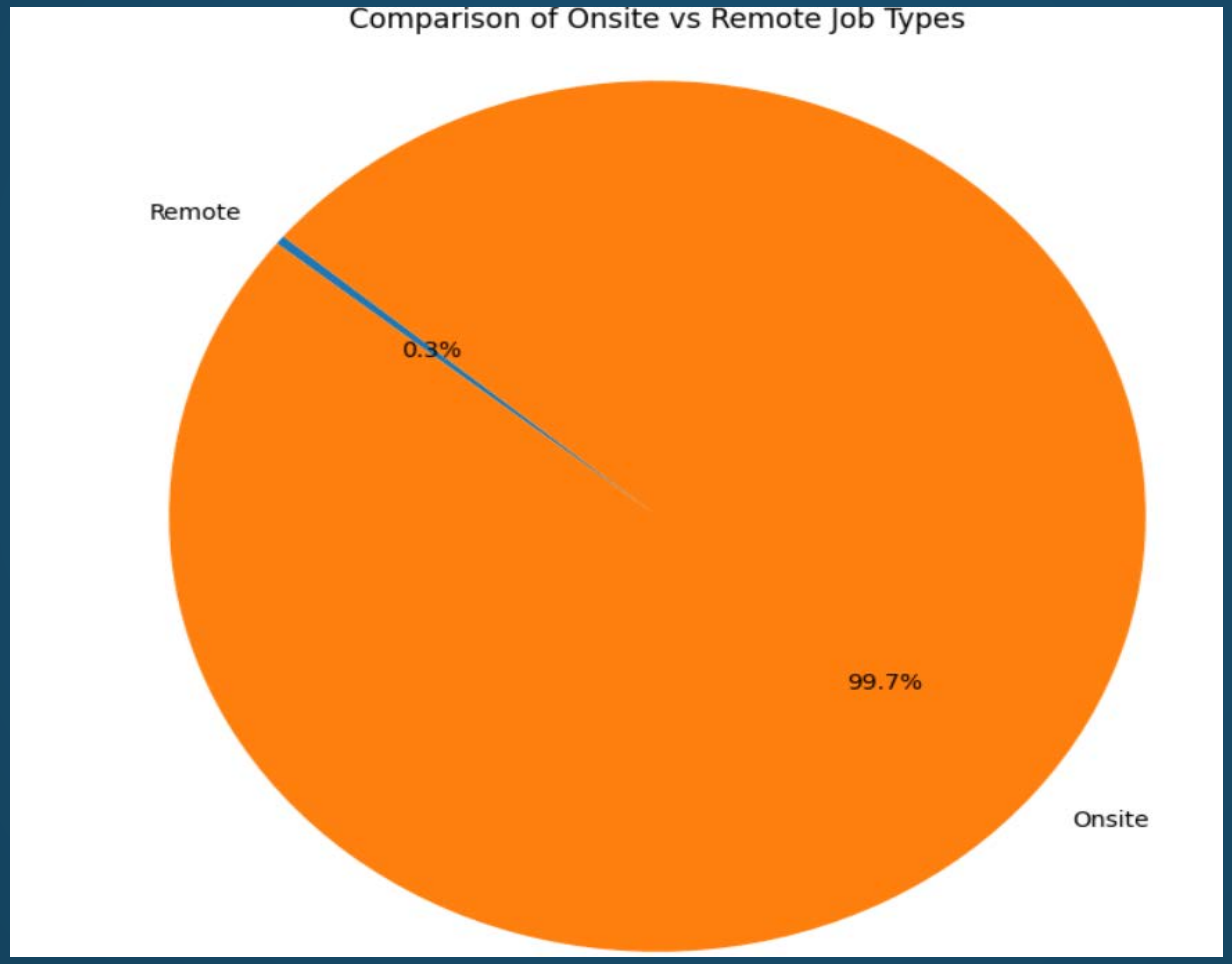
4、Skill Demand Analysis:

```
communication: 370020
customer service: 278012
teamwork: 227535
communication skills: 195820
leadership: 185134
problem solving: 148987
time management: 142861
attention to detail: 133916
problemsolving: 129293
project management: 121515
interpersonal skills: 100218
patient care: 99906
sales: 92977
nursing: 87945
collaboration: 87080
training: 83638
data analysis: 81945
microsoft office suite: 75507
organizational skills: 75245
inventory management: 71899
```



PYSPARK

5、Job Type Analysis:

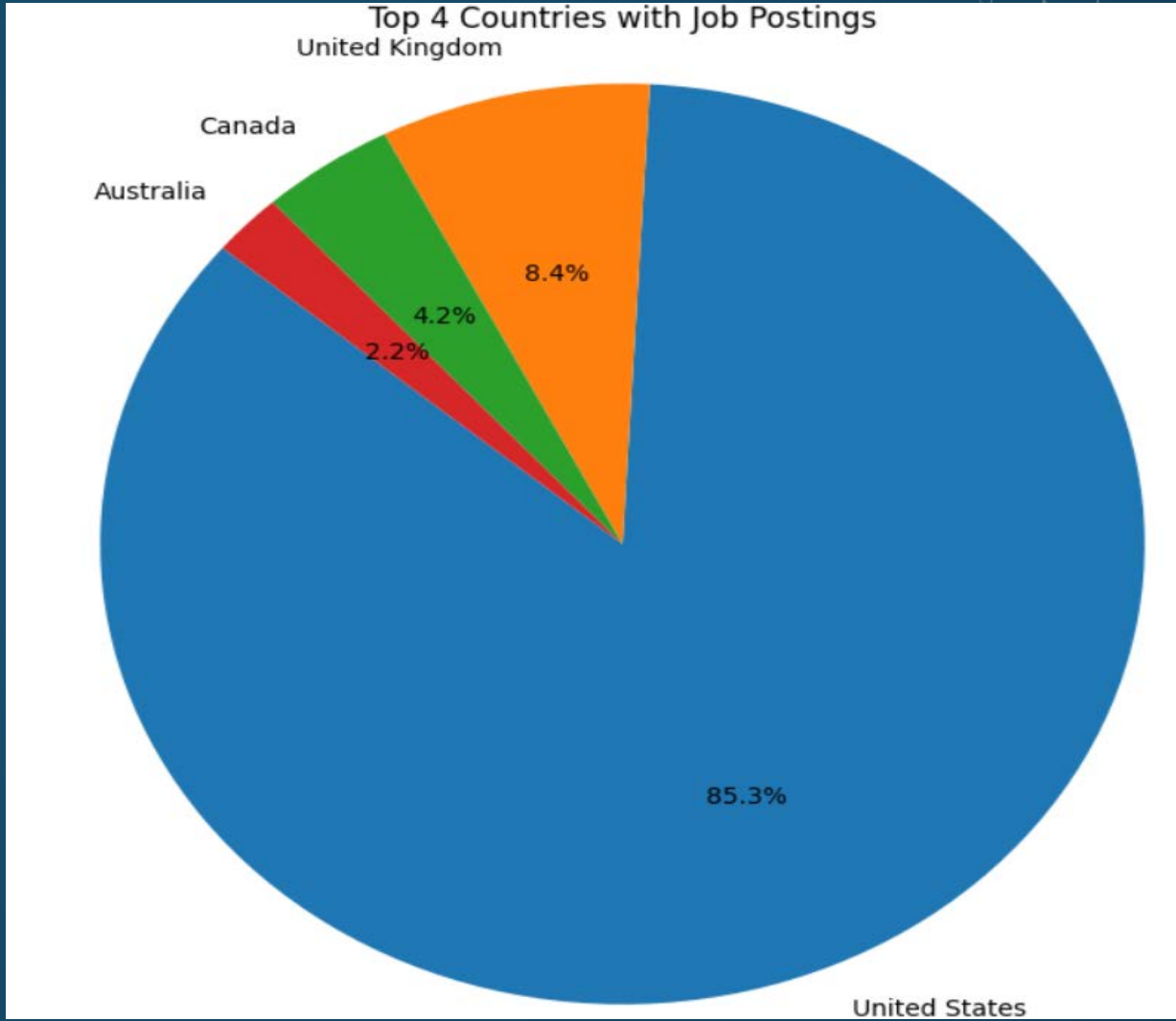


PYSPARK

6、Geographical Distribution of Job Postings:

search_country	count
United States	1105364
United Kingdom	108487
Canada	53903
Australia	28540
Akron	7
Fayetteville	2
Columbus	2
Alexandria	1
Oceanside	1
Clinical Therapist	1
North Chicago	1
Beverly	1
Layton	1
Pittsfield	1
Hollywood	1
Fort Walton Beach	1
Nashville	1
Garland	1
Arkansas	1
Chandler	1

only showing top 20 rows



Data Analysis with PySpark: The PySpark module performs extensive data mining tasks. It includes analyzing job distribution across different locations, company recruitment activities, and job title frequencies. This component utilizes Spark sessions to load, process, and visualize data from CSV files, providing insights into job demand, industry trends, and the popularity of various job titles. It enables data-driven decision-making for job seekers and recruiters by identifying key trends and demands in the job market.

UTILITY FUNCTIONS:

1. load_skills(dir)

```
def load_skills(dir):
    skill_tag_list = []
    with open(dir,'r') as file:
        skill_tag_list = json.load(file)['skill_tags']

    return skill_tag_list
```

This function loads a list of skill tags from a specified JSON file.

- **Parameter**: **dir -** Path to the JSON file.
- **Implementation**:
  - Opens the specified file.
  - Uses **json.load** to read data from the file.
  - Extracts the value associated with the **"skill_tags"** key from the JSON object (a list of skills).
- **Return Value**: A list containing skill tags.

UTILITY FUNCTIONS:

2. do_search_simple(df, search_items)

```python
def do_search_simple(df,search_items):
    def score_row(row, search_terms):
        score = 0
        for column, search_value in search_terms.items():
            if column=='job_skills':
                for skill in search_value:
                    if skill.lower() in str(row[column]).lower():
                        score += 1
            elif search_value.lower() in str(row[column]).lower():
                score += 1
        return score

    df['score'] = df.apply(score_row, axis=1, args=(search_items,))

    df_searched = df[df['score']>0]

    # Sort the DataFrame based on the score
    sorted_df = df_searched.sort_values(by='score', ascending=False)

    return sorted_df[['job_link','job_title','company','job_location','job_type']]
```

This function filters data within a DataFrame based on given search criteria and calculates a matching score for each row.

- **Parameters**:
  - **df** - DataFrame.
  - **search_items** - A dictionary of search criteria, where keys are column names and values are the search terms.
- **Implementation**:
  - Defines an inner function **score_row**, which calculates a score for each row of the DataFrame based on how well it matches the search terms. If the column is "job_skills", it checks for each skill in the list if it is present in the row.
  - Applies **score_row** across the DataFrame to compute scores for each row and adds them as a new column.
  - Filters rows with a score greater than zero and returns them sorted by score.
- **Return Value**: A filtered and sorted DataFrame containing only specific columns.

UTILITY FUNCTIONS:

3. paginate_dataframe(df, page_size, page_num)

```
# search page setting
def paginate_dataframe(df, page_size, page_num):
    start_index = page_size * (page_num - 1)
    end_index = start_index + page_size
    return df.iloc[start_index:end_index]
```

This function paginates a DataFrame.

- **Parameters**:
  - **df** - DataFrame.
  - **page_size** - Number of records per page.
  - **page_num** - Page number.
- **Implementation**:
  - Calculates the start and end indices based on the page number and page size.
  - Uses **iloc** to slice the DataFrame according to the index range.
- **Return Value**: A sliced DataFrame representing the content of the specified page.

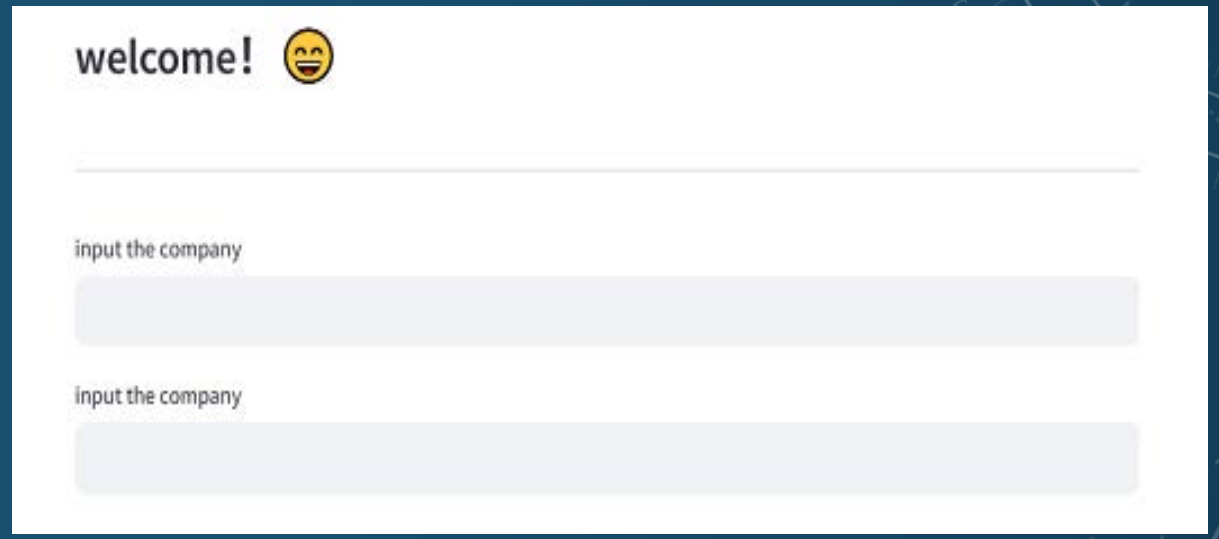
Utility Functions: The Utils module contains functions essential for data manipulation and search functionality within the application. It includes functions for loading skill tags from JSON files, filtering DataFrame based on search criteria, and paginating results. These utilities are crucial for handling data efficiently and ensuring that the web application can perform searches and display results effectively.

WEB

Streamlit Configuration

Page

```
# page_title
st.set_page_config(page_title='job select')
# header
st.header(':red[select] :green[ your] :blue[ job]')
# subheader
st.subheader('welcome! :smile:')
st.divider()  # Draws a horizontal rule
#search
st.text_input("input the company",key="company")
st.text_input("input the job title",key="title")
```



WEB

Search and Results Display

```
if 'result' in st.session_state:
    paged_data = paginate_dataframe(st.session_state['result'], page_size, st.session_state['page_num'])
    # Display the current page of entries
    st.write(f"Displaying page {st.session_state['page_num']} of {len(st.session_state['result']) // page_size
    st.dataframe(paged_data)

    # Add navigation buttons
    col1, col2 = st.columns(2)
    with col1:
        prev_button = st.button("Previous", key="prev")
    with col2:
        next_button = st.button("Next", key="next")

    if prev_button:
        if st.session_state['page_num'] > 1:
            st.session_state['page_num'] -= 1

    if next_button:
        if st.session_state['page_num'] * page_size < len(st.session_state['result']):
            st.session_state['page_num'] += 1
```

```
#button
if st.button("Submit"):
    search_items = {
        'state':st.session_state.state,
        'job_skills':st.session_state.skills,
        'company':st.session_state.company,
        'job_title':st.session_state.title
    }

    result = do_search_simple(df,search_items)
    st.session_state['result'] = result  # Store result in session state
    st.session_state['page_num'] = 1  # Reset to first page
```

Submit

Displaying page 4 of 129636

	job_link	job_title
30	https://www.linkedin.com/jobs/view/cheese-specialist-at-safeway-3742784971	Cheese Specialist
31	https://www.linkedin.com/jobs/view/rn-at-bon-secours-mercy-health-3781018201	RN
32	https://www.linkedin.com/jobs/view/paes-schools-counselor-sy-23-24-at-aztec-mun	PAES Schools Counselo
33	https://www.linkedin.com/jobs/view/sales-lead-slpt-lane-bryant-at-lane-bryant-378	Sales Lead (SLPT) - Lan
34	https://www.linkedin.com/jobs/view/retail-district-manager-unassigned-at-dollar-ge	RETAIL DISTRICT MANA
35	https://www.linkedin.com/jobs/view/asset-wealth-management-%E2%80%93-regul.	Asset Wealth Managem
36	https://www.linkedin.com/jobs/view/travel-rn-med-surg-at-rnnetwork-3802701170	Travel RN - Med Surg
37	https://www.linkedin.com/jobs/view/sr-experience-design-manager-learn-and-help-	Sr Experience Design M
38	https://ca.linkedin.com/jobs/view/coordonnateur-diversit%C3%A9-%C3%A9quit%C	Coordonnateur, Diversi
39	https://www.linkedin.com/jobs/view/assistant-salon-manager-cornelius-gateway-at-	Assistant Salon Manage

Previous Next

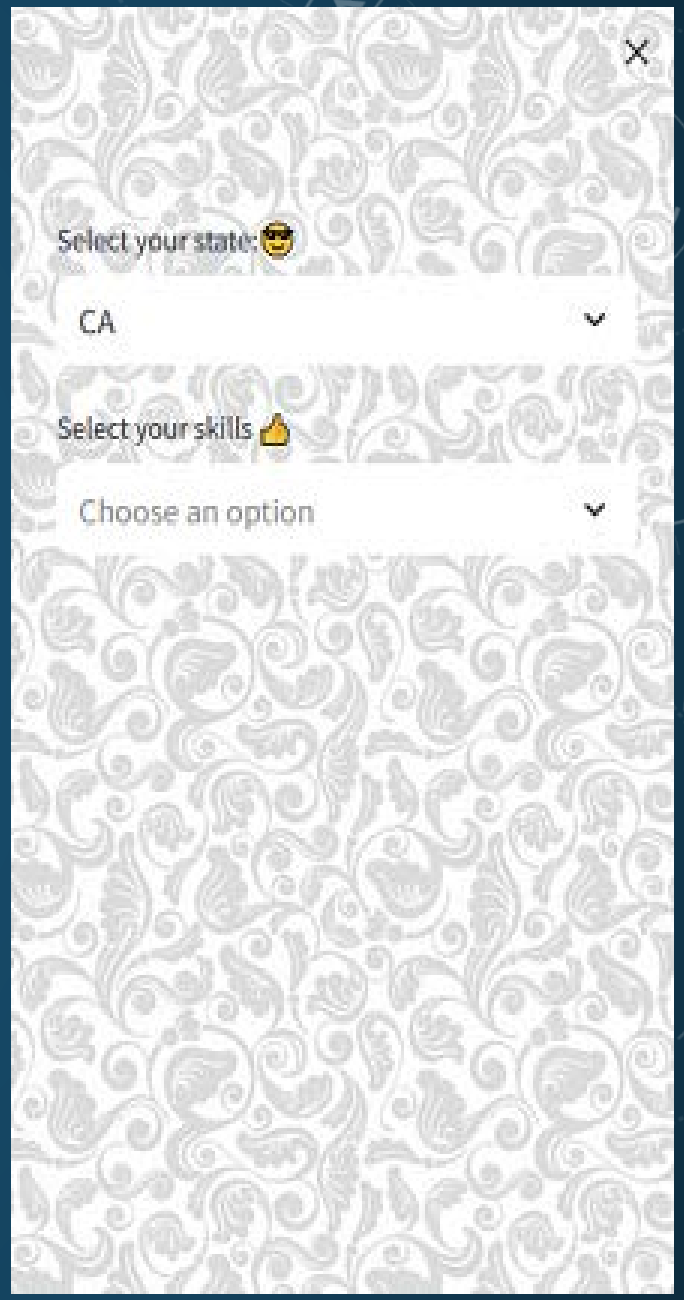
WEB

Sidebar Background:

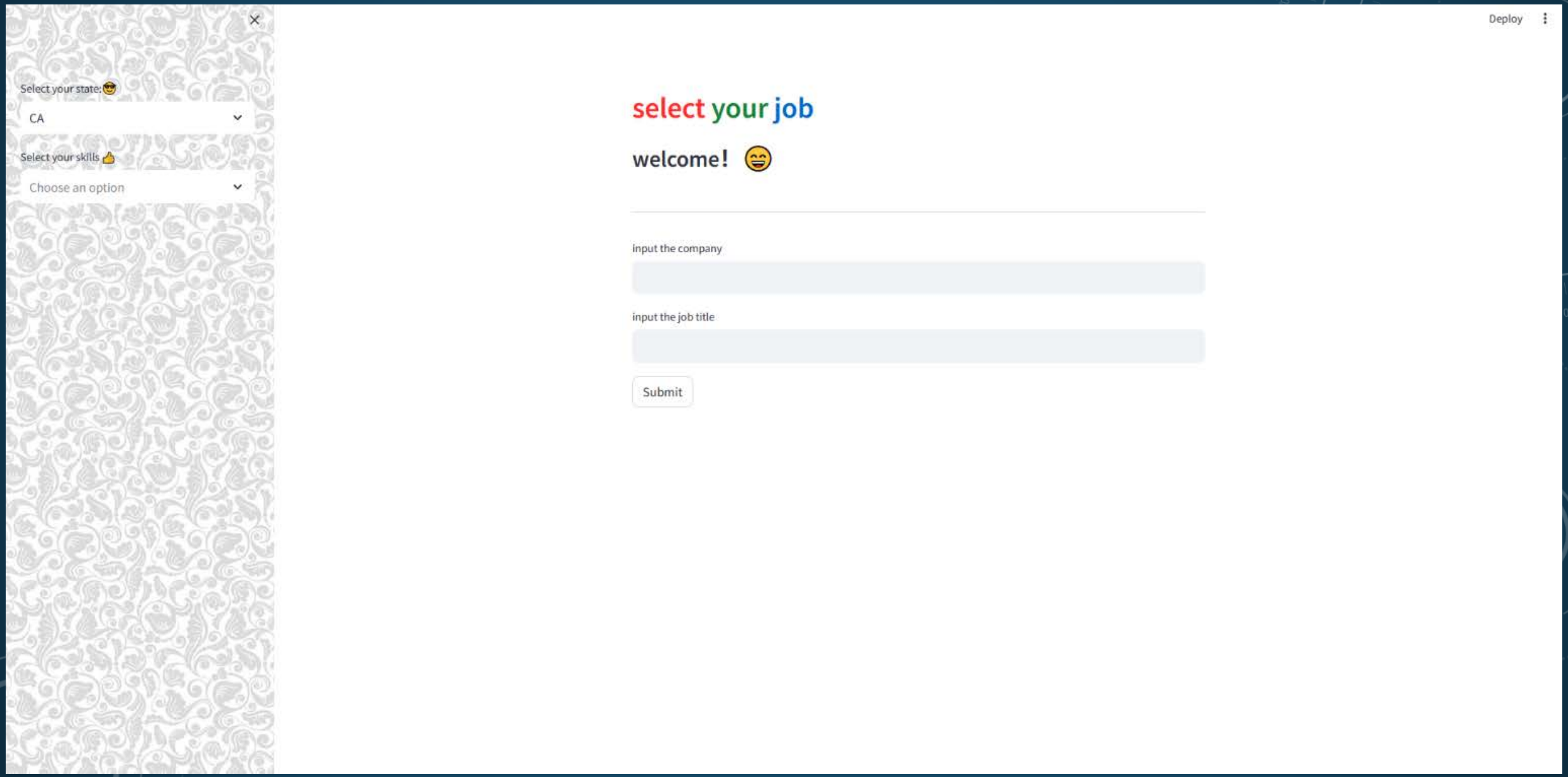
Configuration

```python
##back ground
def sidebar_bg(side_bg):

    side_bg_ext = 'png'

    st.markdown(
        f"""
        <style>
        [data-testid="stSidebar"] > div:first-child {{
            background: url({base64.b64encode(open(side_bg, "rb").read()).decode()}
        }}
        </style>
        """,
        unsafe_allow_html=True,
        )

#use
sidebar_bg('./pics/background.jpg')
```

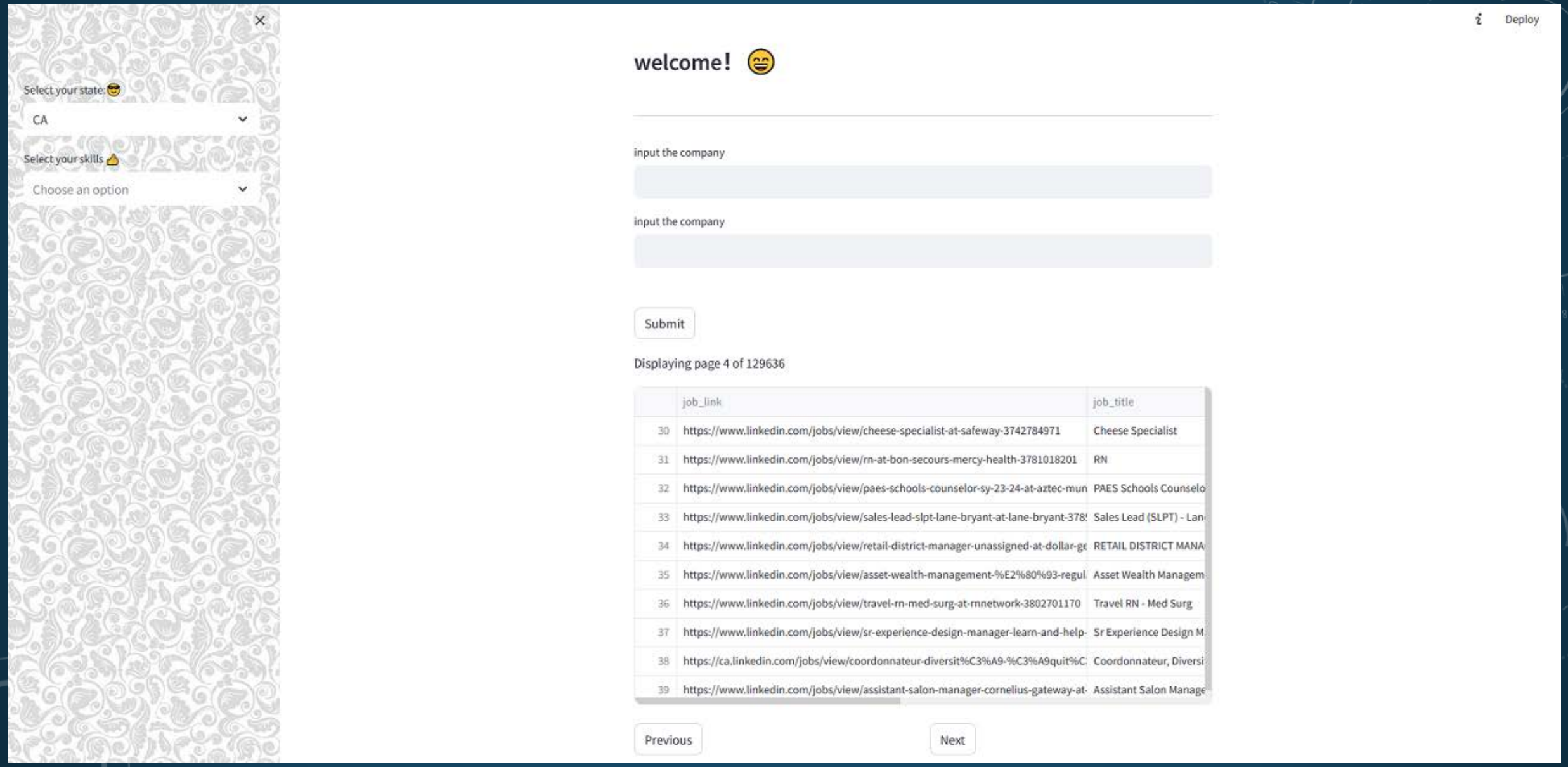
```python
###sidebar
##state
add_selectbox = st.sidebar.selectbox(
    'Select your state::sunglasses:',
    states,
    key='state'
)

##skills
# list
#options = ['Python', 'JAVA', 'C++'] # debug only
#multiselect
selected_options = st.sidebar.multiselect(
    'Select your skills     :thumbsup: ',
    options,
    default=[],  # Set default selected options (an empty list means there are no default selected options)
    key='skills'
)
```



WEBSITE


Deploy
Select your state:😎
CA
Select your skills👍
Choose an option
select your job
welcome! 😄
input the company
input the job title
Submit

WEBSITE


Deploy
welcome! 😄
Select your state: 😎
CA
Select your skills 👍
Choose an option
input the company
input the company
Submit
Displaying page 4 of 129636
job_link
job_title
30 https://www.linkedin.com/jobs/view/cheese-specialist-at-safeway-3742784971 Cheese Specialist
31 https://www.linkedin.com/jobs/view/rn-at-bon-secours-mercy-health-3781018201 RN
32 https://www.linkedin.com/jobs/view/paes-schools-counselor-sy-23-24-at-aztec-mun PAES Schools Counselo
33 https://www.linkedin.com/jobs/view/sales-lead-slpt-lane-bryant-at-lane-bryant-378 Sales Lead (SLPT) - Lan
34 https://www.linkedin.com/jobs/view/retail-district-manager-unassigned-at-dollar-ge RETAIL DISTRICT MANA
35 https://www.linkedin.com/jobs/view/asset-wealth-management-%E2%80%93-regul Asset Wealth Managem
36 https://www.linkedin.com/jobs/view/travel-rn-med-surg-at-rmnetwork-3802701170 Travel RN - Med Surg
37 https://www.linkedin.com/jobs/view/sr-experience-design-manager-learn-and-help- Sr Experience Design M
38 https://ca.linkedin.com/jobs/view/coordonnateur-diversit%C3%A9-%C3%A9quit%C Coordonnateur, Diversi
39 https://www.linkedin.com/jobs/view/assistant-salon-manager-cornelius-gateway-at- Assistant Salon Manage
Previous
Next

Web Application with Streamlit: The web application leverages Streamlit to provide an interactive user interface for job search. It allows users to input search criteria such as job title, company, skills, and location, and uses the utility functions to filter and display relevant job postings. The application also includes features for navigating through search results, selecting job skills from a dynamic list, and customizing the sidebar background.

CONCLUSION AND LESSON LEARNED

Conclusion

This project integrates three components—data analysis using PySpark, utility functions, and a web application built with Streamlit—to create a comprehensive job search and analysis platform. Overall, this project showcases the integration of backend data processing with a user-friendly frontend interface, making it a powerful tool for job seekers and analysts. The combination of PySpark for heavy-duty data processing, Python for utilities, and Streamlit for web deployment creates a versatile platform that addresses various aspects of job market analysis and search functionality.

lesson learned

Integration of Technologies: Each technology plays a crucial role, with PySpark handling large-scale data operations, Python ensuring functional versatility, and Streamlit providing a user-friendly interface

Data-Driven Insights: Utilizing PySpark to analyze and visualize data has underscored the value of data-driven decision making in real-world applications.

User-Centric Design: The use of Streamlit to create an interactive web application emphasizes the importance of user-centric design.

Modularity and Scalability: By structuring the utility functions and data processing separately from the web interface, the project is made more modular, which enhances scalability and maintainability.

THANK YOU!