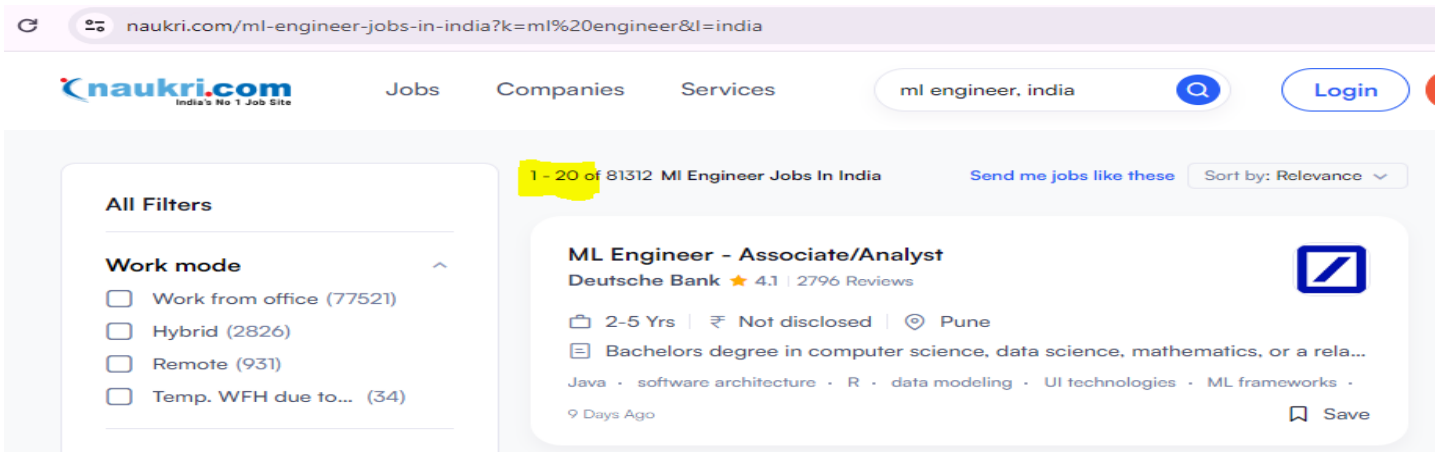
Project - Job Market Analysis

Analysis of job market data to explore the open positions related to the jobs in the field of data or machine learning in a particular region.

A. Data Collection Method:

Sample Size:

Performed web scraping on job portal **"Naukri.com"** using Python, Beautiful Soup and Selenium for data collection and analysis. After searching for job postings for "ML Engineer" in India, the "Naukri.com" website shows 20 job postings at a time.



In order to get 100 samples, I had to web scrape in total 5 pages that accounted to 5*20=100 samples. Hence the sample size consisted of **100 job postings** related to "ML Engineer" positions in India.

Process of Data Collection:

Beautiful Soup in Python was used as a web scraping technique to extract job postings from "Naukri.com."

```
import requests
from bs4 import BeautifulSoup
url_1="https://www.naukri.com/ml-engineer-jobs-in-india?k=ml+engineer&l=india"
url_2="https://www.naukri.com/ml-engineer-jobs-in-india-2"
url_3="https://www.naukri.com/ml-engineer-jobs-in-india-3"
url_4="https://www.naukri.com/ml-engineer-jobs-in-india-4"
url_5="https://www.naukri.com/ml-engineer-jobs-in-india-5"
res1=requests.get(url_1)
print(res1)
```

```
<Response [200]>
```

The above screenshot has HTTP response code 200 meaning success response for the 1st page with 20 job postings. Remaining 4 pages also had 200 response codes. However, the response object returned by the HTTP request of requests library doesn't have HTML tags. Kindly refer to below screenshot to see the problem:

```
soup1=BeautifulSoup(res1.text,'html.parser')
print(soup1)
```

```
4.6v14.3h-10.8V200.8z"></path><circle class="styles_sc__swoosh-2__8uCcv" cx="187.7" cy="224.2" r="4.6"></circle><path class
="styles_sc__swoosh-3__V7EJf" d="M163.1 251.5v-11.3h2.3v11.3H163.1z"></path><path class="styles_sc__swoosh-3__V7EJf" d="M175
251.5h-2.2v-4.2c0-0.9 0-1.5-0.1-1.7 -0.1-0.3-0.2-0.5-0.4-0.6 -0.2-0.1-0.5-0.2-0.8-0.2 -0.4 0-0.7 0.1-1 0.3 -0.3 0.2-0.5 0.5-
0.6 0.8 -0.1 0.3-0.2 1-0.2 1.9v3.7h-2.2v-8.2h2v1.2c0.7-0.9 1.6-1.4 2.7-1.4 0.5 0 0.9 0.1 1.3 0.3 0.4 0.2 0.7 0.4 0.9 0.7 0.2
0.3 0.3 0.6 0.4 0.9 0.1 0.3 0.1 0.8 0.1 1.5V251.5z"></path><path class="styles_sc__swoosh-3__V7EJf" d="M184.7 251.5h-2v-1.2c-
0.3 0.5-0.7 0.8-1.2 1 -0.5 0.2-0.9 0.3-1.4 0.3 -0.9 0-1.7-0.4-2.4-1.1 -0.7-0.8-1-1.8-1-3.2 0-1.4 0.3-2.4 1-3.2 0.7-0.7 1.5-1.
1 2.5-1.1 0.9 0 1.7 0.4 2.4 1.1v-4.1h2.2V251.5zM178.9 247.2c0 0.9 0.1 1.5 0.4 1.9 0.3 0.6 0.8 0.8 1.5 0.8 0.5 0 0.9-0.2 1.3-
0.6 0.4-0.4 0.5-1.1 0.5-1.9 0-0.9-0.2-1.6-0.5-2 -0.3-0.4-0.8-0.6-1.3-0.6 -0.5 0-0.9 0.2-1.3 0.6C179.1 245.8 178.9 246.4 178.9
247.2z"></path><path class="styles_sc__swoosh-3__V7EJf" d="M186.8 242.2v-2h2.2v2H186.8zM186.8 251.5v-8.2h2.2v8.2H186.8z"></pa
th><path class="styles_sc__swoosh-3__V7EJf" d="M192.8 245.8l-2-0.4c0.2-0.8 0.6-1.4 1.1-1.8 0.5-0.4 1.3-0.6 2.4-0.6 1 0 1.7 0.
1 2.2 0.3 0.5 0.2 0.8 0.5 1 0.9 0.2 0.4 0.3 1 0.3 1.9l0 2.5c0 0.7 0 1.2 0.1 1.6 0.1 0.3 0.2 0.7 0.4 1.1h-2.1c-0.1-0.1-0.1-0.4
-0.2-0.6 0-0.1-0.1-0.2-0.1-0.3 -0.4 0.4-0.8 0.6-1.2 0.8 -0.4 0.2-0.9 0.3-1.3 0.3 -0.8 0-1.5-0.2-2-0.7 -0.5-0.5-0.7-1-0.7-1.7
```

So, to get the HTML source, l used the selenium library of Python which is generally used for automation. I used a Chrome web driver.

```
from selenium import webdriver
from selenium.webdriver.chrome.service import Service
import time
import pprint

# Specify the full path to the ChromeDriver executable
chrome_service = Service('chromedriver.exe')

# Create Chrome instance with options and service
chrome_browser = webdriver.Chrome(service=chrome_service)
chrome_browser.maximize_window()
#Opens chrome browser and redirects to the link provided in the get method.
chrome_browser.get(url_1)
time.sleep(5)

soup1 = BeautifulSoup(chrome_browser.page_source,'html.parser')
print(soup1.prettify())
```

Similar steps were followed for the remaining 4 pages and each page source was extracted. The output had HTML tags, kindly refer to screenshot below:

```
            <label class="styles_chkLbl__n2x09" for="chk-Data Science &amp;amp; Machine Learning-glbl_qcrc-">
             <i class="ni-icon-unchecked">
             </i>
             <p class="styles_txtLbl__Pl_48">
              <span class="styles_ellipsis__cvWP1 styles_filterLabel__jRP04" title="Data Science &amp;amp; Machine Learning">
               Data Science &amp;amp; Machine Learning
              </span>
              <span class="styles_filterCount__Xn_PG">
               (1823)
              </span>
             </p>
            </label>
           </div>
           <a class="styles_read-more-link__DU4hQ" id="glbl_qcrc">
            <span>
             View More
            </span>
           </a>
          </div>
```

```
chrome_browser.get(url_2)
time.sleep(5)
```

After fetching the page source of each targeted page, I extracted the following information:
- Title of the job
- Location of the company (including remote)
- Hiring/ Recruiting company name
- Skill set required for the respective job position by each company.

In order to extract the above listed information, various HTML appropriate tags with class were used to acquire HTML content associated with the targeted page and data wrangling was performed on the collected data to make it more useful for analysis and data visualization.

Try and except conditions were used for each to maintain consistency and also to handle “TypeError = NoneType” fields. Filters were used to group similar types of title and location. Case-sensitive same or similar data were grouped and then stored in the list, this is part of cleaning the data which helps in data visualization. The final results of each list were used to create Pandas Dataframe and named ‘job_data’. There were no missing or duplicate values.

```
import pandas as pd
# Create a DataFrame to store the extracted data
job_data = pd.DataFrame({
    "Job Title": job_titles,
    "Location": job_locations,
    "Company": company,
    "Key Skills": key_skills
})

job_data.shape
```

```
(100, 4)
```

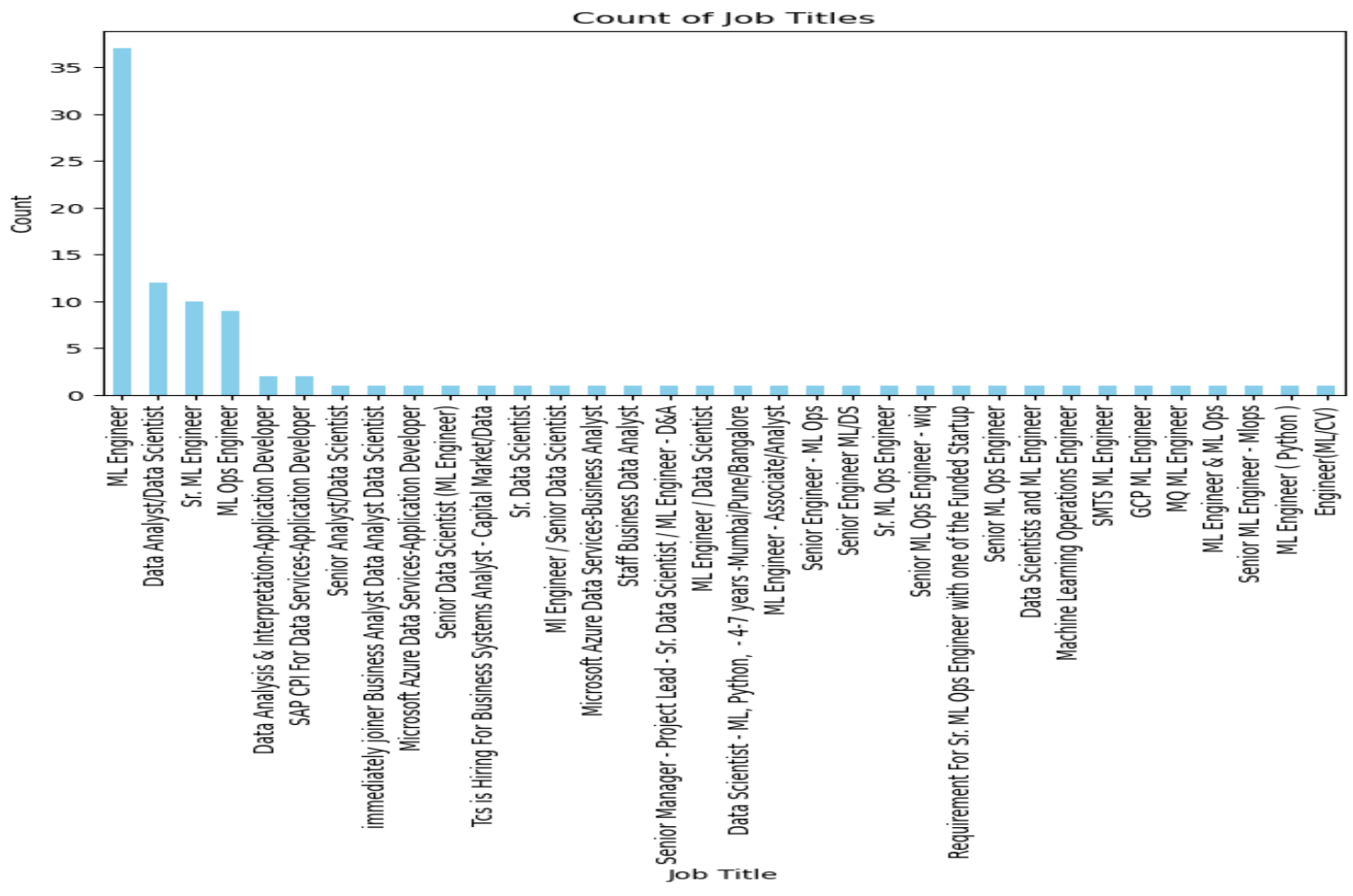
```
# Print the DataFrame
job_data.head()
```

	Job Title	Location	Company	Key Skills
0	ML Engineer - Associate/Analyst	Pune	Deutsche Bank	Java,software architecture,R,data modeling,UI ...
1	Lead ML Engineer	Pune	Xoriant	Computer science,data science,Data modeling,An...
2	ML Engineer (Python)	Bangalore/Bengaluru	RandomT	machine learning,predictive modelling,python,d...
3	ML Engineer	Hybrid - Hyderabad(Gachibowli)	Space Multimedia	GenAI,NLP,ML engineer,LLM,MI,Tensorflow,AI,Com...
4	ML Engineer	Hyderabad/Secunderabad, Telangana	OSI Digital	machine learning,python,tensorflow,Azure,nlp,c...

B. Market Data Visualization

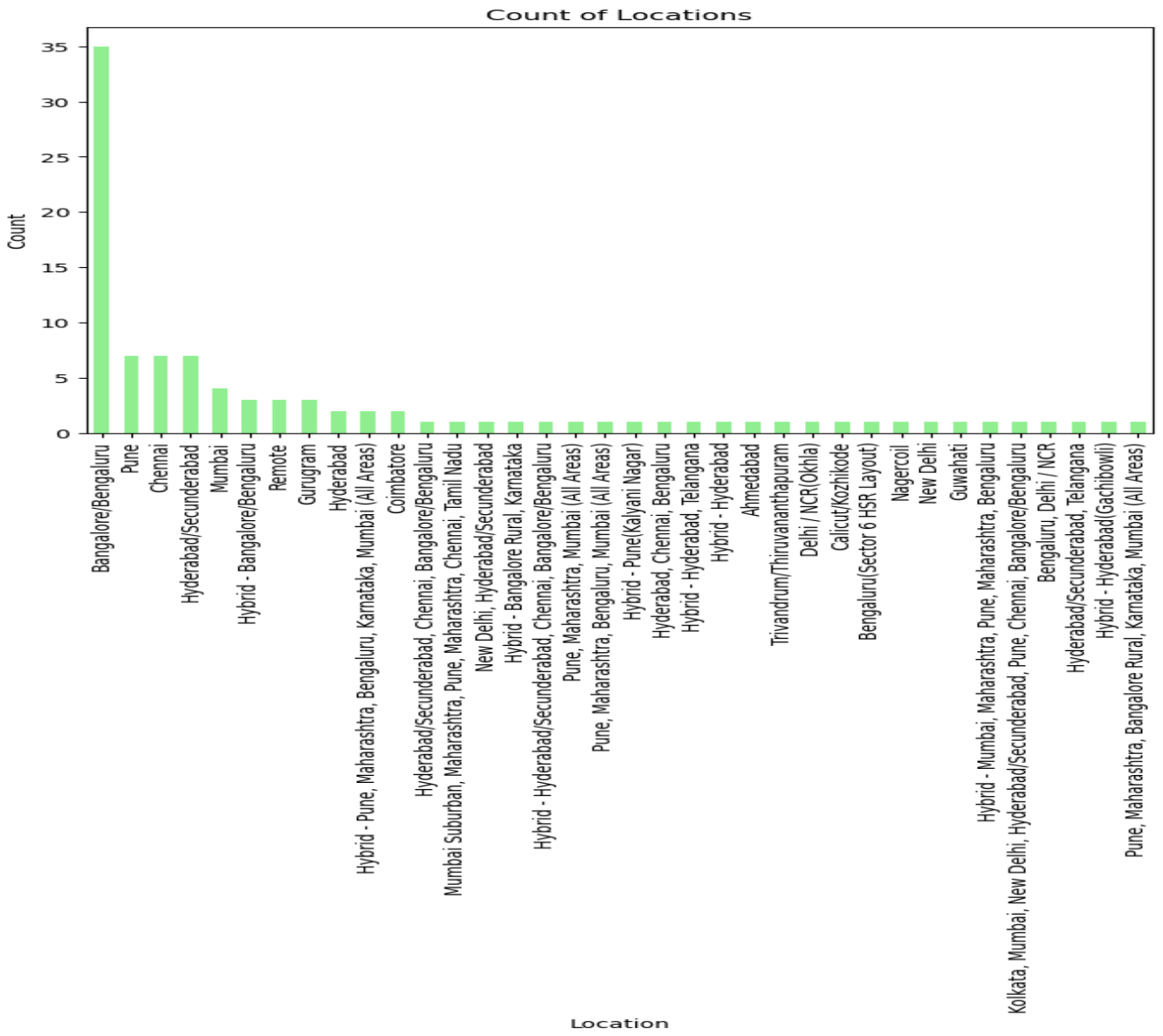
Univariate analysis:

- **Data visualization for job title**: The bar chart shows that maximum number of job titles obtained were ‘ML Engineer’ as queried. The next title that was mostly found was ‘Data Analyst/ Data Scientist’. Kept the least found titles to highlight that different companies use different job posting titles for ‘ML engineer’ job roles.

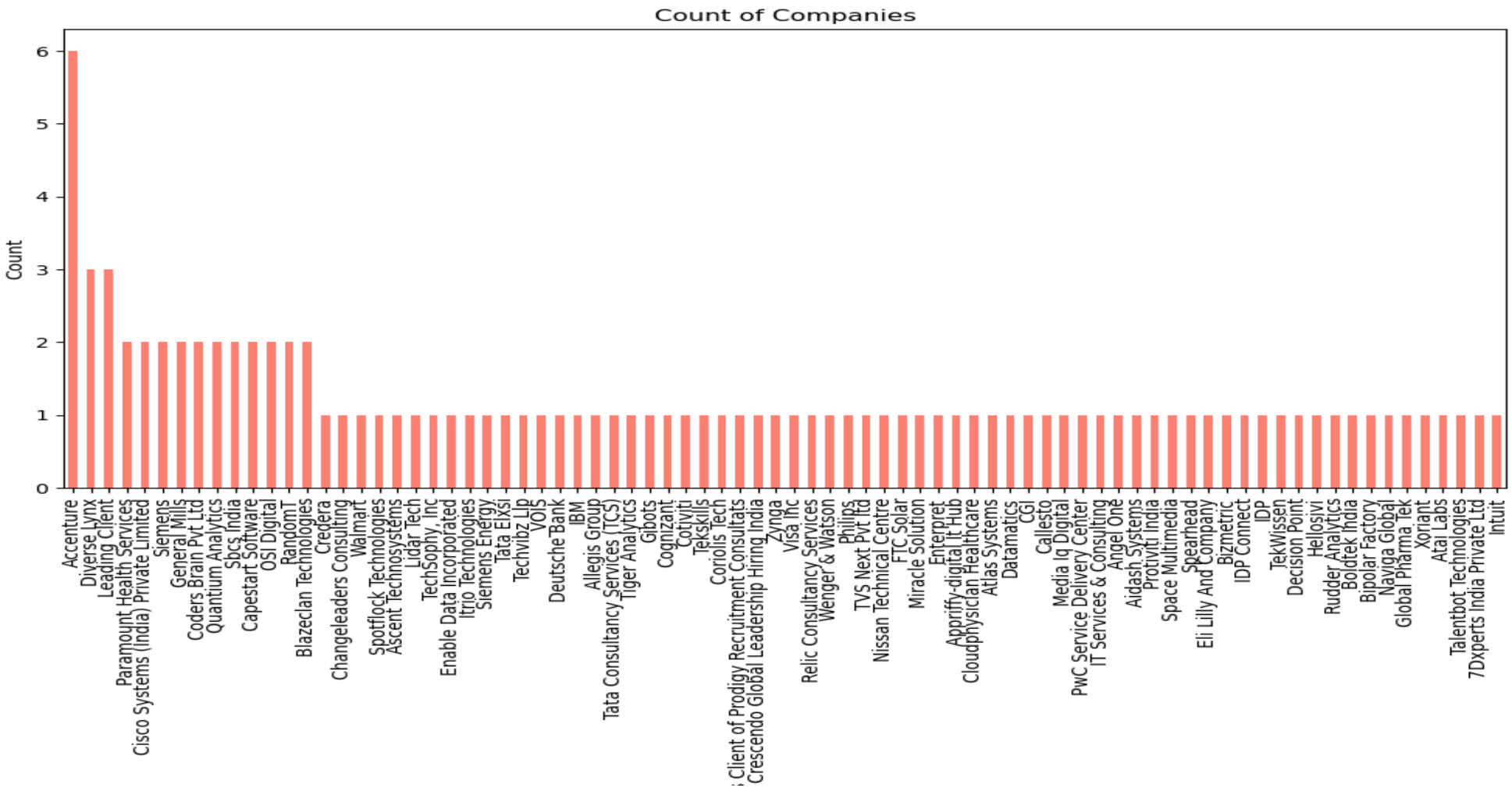


- **Data visualization for company location:** The below table and bar chart shows that the majority of ‘ML Engineer’ hiring companies are located in Bangalore/ Bengaluru. There are some job positions which are flexible in terms of locations. The prominent ones are Pune, Mumbai, Chennai and Hyderabad.

	Location	Count
1	Bangalore/Bengaluru	35
5	Chennai	7
30	Pune	7
21	Hyderabad/Secunderabad	7
25	Mumbai	4



- **Data visualization for different companies hiring/recruiting ML engineers:** “Accenture” is hiring or recruiting maximum number of ‘ML Engineers’ amongst the 100 samples, followed by Diverse Lynx and Leading Client.

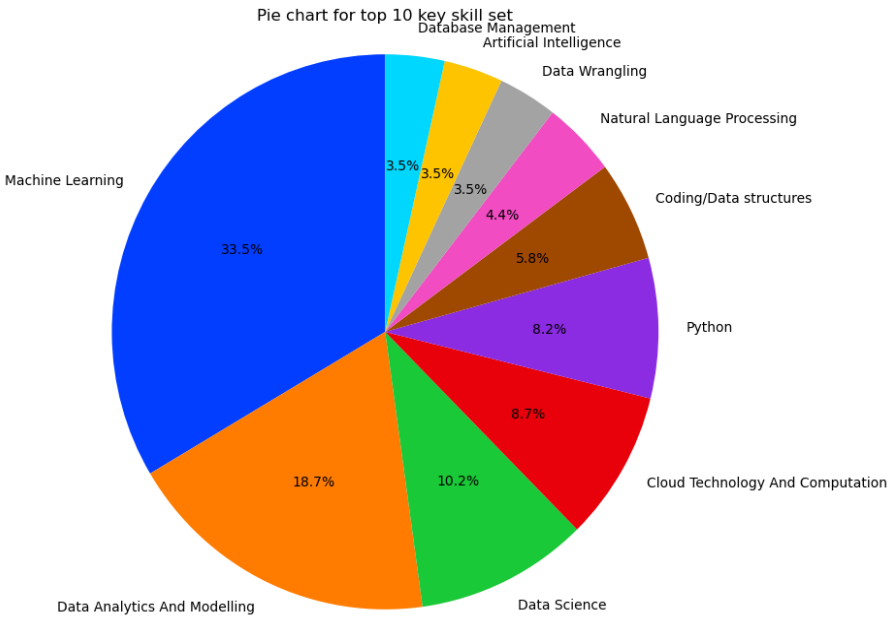


- **Data visualization for key skills required for most of the companies:** In the job_data data frame it can be noticed that column “Key skills” have multiple skills in each row. To clean that data and maintain the frequency of each skill set, created one data frame that holds different unique skills as key and its count as value and sorted them in descending order by their count value.

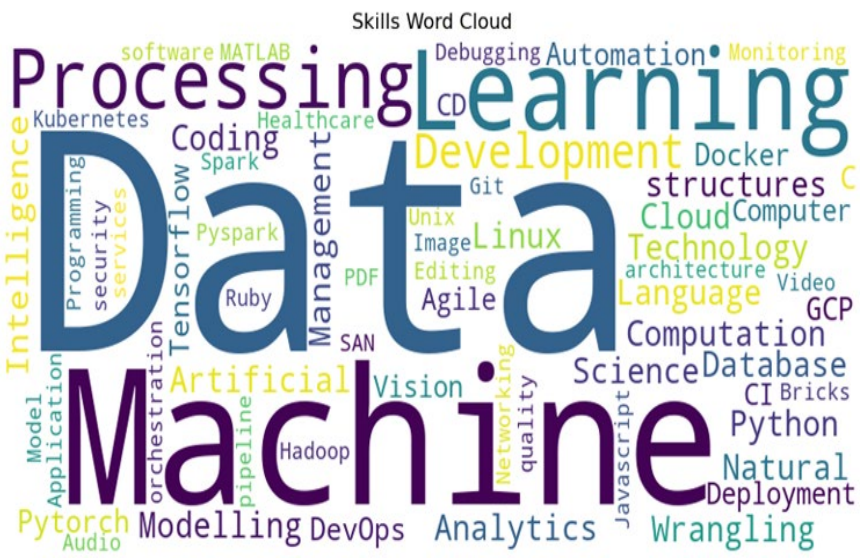
	Skill	Frequency
1	Machine Learning	115
13	Data Analytics And Modelling	64
17	Data Science	35
29	Cloud Technology And Computation	30
10	Python	28
...	...	...
139	LLM	1
140	object detection	1
141	Basic	1
142	Senior	1
241	Gaming	1

242 rows × 2 columns

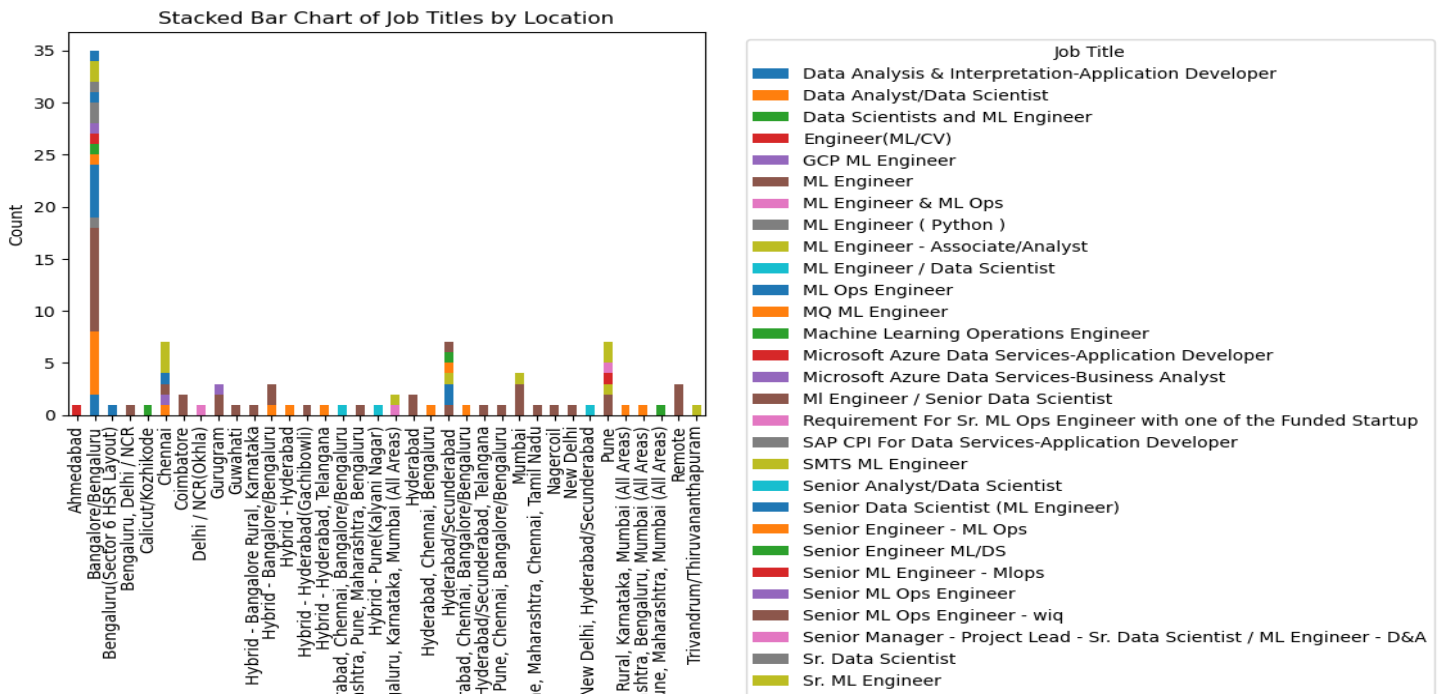
As there are 242 unique skills required, filtered top 10 unique skills that are required by the majority of the companies. Using pie chart for data visualization and better understanding of the most required skill set for the ML Engineer job role:



Word cloud for the varied skills for "ML Engineer" job positions:



Bivariate analysis: The below stacked bar chart highlights job titles with respect to the different company locations and helps in understanding which location gives diverse job roles pertaining to the "ML Engineer" role. The diverse job titles are highlighted in the legend section which focuses on different seniority levels for the respective role and the chart shows their bifurcation as per locations.



C. Your Ideal Job:

Job Title: ML Engineer

Key technical skills required:

- Strong understanding of machine learning algorithms, ML frameworks and its application.
- Proficiency in data analytics and modeling.
- In depth knowledge of cloud technologies and computation.
- Most in demand programming language is Python.
- Extensive knowledge of data science concepts and its application
- Superb analytical and problem-solving abilities.

Key non-technical skills:

- Great communication and collaboration skills.
- Excellent time management and organizational abilities.