Insights into Market Dynamics: Exploring Job Openings, Applicant Trends

Mounica Seelam - S559228
Harshitha Chowdary Alapati - S561420
Swarupa Jinne - S558943
Supriya Bodapati - S562040

April 2024

1 Project Idea

The purpose of this project is to use PySpark to analyze workforce data and extract insights on industry trends, job types, employee demographics, and company popularity. The idea is to develop a comprehensive data analytics platform that integrates job openings data, applicant trends, and corporate profiles to provide insights into workforce dynamics. Utilize advanced data mining techniques to identify patterns, predict future job demand, and assess the competitiveness of companies in attracting talent, aiding both job seekers and recruiters in making informed decisions.

2 Technology Summary

PySpark, Python, Jupyter Notebooks

3 Architecture Diagram

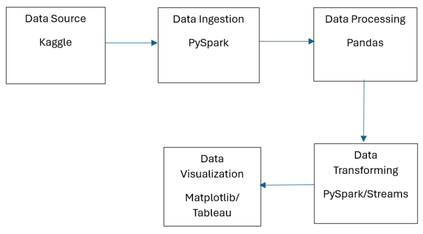


Figure 1: Architecture.

4 Architecture Summary

Data source: Kaggle - The dataset is sourced from Kaggle, a popular platform for datasets and data science competitions.

Data Ingestion: PySpark is used for data ingestion, enabling the loading of the Kaggle dataset into a PySpark Data Frame.

Data Preprocessing: Data preprocessing is performed using the Pandas library, which allows for handling missing values, cleaning, and transforming data before moving it into PySpark.

Data Transformation: PySpark used for data transformation, can be used for more complex stream-based transformations.

Data Visualization: Matplotlib is utilized for data visualization, providing a versatile library for creating static, animated, and interactive plots.

5 Project Goals

Goal 1: How many employees have full-time and contract-based jobs?

- The goal is to count the number of employees with full-time and contract-based jobs, providing a clear understanding of the workforce composition in terms of employment types. This applies volume in big data 5 V's.

Goal 2: How many employees have different types of jobs(Ex – It, Hardware)?

- This goal aims to quantify the number of workers associated with various job kinds. The analysis's findings provide support for strategic choices about workforce planning and talent management.
- The query is useful for summarizing the distribution of job occurrences across different designations in the job-cleanData dataset. It provides insights into the frequency of each job designation and can be valuable for workforce analysis and planning.
- This query applies the 'Volume' from 5 big data V's. The query is concerned with analyzing and aggregating data related to job occurrences, and the volume of job records in the dataset plays a significant role in the processing requirements and overall performance.

Goal 3: Employees with different states and different types of Jobs

- This goal provides a holistic view of the workforce by examining the distribution of employees across different states and job types. The outcomes of this analysis support strategic decisions related to regional workforce planning and the alignment of skills with organizational needs.

- This above query is useful for summarizing and analyzing the distribution of employees across different states and job designations in the "job-cleanData" dataset. It provides insights into workforce distribution and can be valuable for regional and role-based analysis.
- This query applies the 'Variety' from 5 bigdata V's. The query is working with data that has variety in terms of different states and job designations, and it highlights the importance of handling diverse data types in the big data context.

Goal 4: How many applicants does a company have?

- This goal shows to provide a quantitative measure of the interest and engagement the company receives from potential candidates. The results contribute to recruitment analytics and decision-making processes for optimizing talent acquisition strategies.
- Volume is the most applicable V as the goal involves quantifying the number of applicants for a company. The larger the number of applicants, the greater the volume of data that needs to be processed and analyzed to determine the total count.

Goal 5: How many employees have different designations in the It industry for different states?

- This goal aims to quantify and understand the distribution of employees within the Information Technology (IT) industry across various states based on their job designations.
- This query is to provide insights into the distribution of employees across different states and job designations specifically within the "IT Services and IT Consulting" industry. It's useful for analyzing workforce composition and patterns within a particular sector of the organization.
- This query applies the 'Variety' from 5 bigdata V's. The query involves working with structured data representing diverse job information within a specific industry, emphasizing the need to handle various data types and sources effectively.

Goal 6: How many employees are working for onsite and remote jobs?

- This goal describes identifying the number of employees in onsite and remote job roles to gain insights into the workforce distribution based on work locations.
- Variety is applicable as the goal involves categorizing employees based on their work locations, which can be considered a categorical variable.

Handling and analyzing this variety in work location data is essential for providing insights into the distribution of employees based on their onsite or remote work status.

Goal 7: Compare the LinkedIn followers for each company in the dataset to identify trends and popular companies.

- The goal is to analyze and compare the number of LinkedIn followers for each company in the dataset. The aim is to identify trends and recognize popular companies based on the size of their LinkedIn followership.
- This query extracts the maximum LinkedIn followers for each unique name in the ”job-cleanData” dataset, ordering the results by the maximum follower count in ascending order and displaying the outcome.
- This query applies the 'Volume' from 5 big data V's. The query is concerned with analyzing data, and the size of the dataset (number of records) can impact the computational resources required for processing.

Goal 8: Calculate the number of applicants based on job levels.

- The goal is to quantify and analyze the number of job applicants categorized by different job levels. This analysis aims to provide insights into the distribution of applicants across various levels within the organization.
- This query selects the ”designation” column from the ”job-cleanData” dataset and counts the total number of applicants for each unique job designation. The results are then grouped by designation, and the count is displayed. This query provides a summary of the applicant count for each job designation in the dataset.
- This query applies the 'Volume' from 5 bigdata V's. The query is concerned with analyzing a substantial volume of data related to job designations and applicant counts.

Goal 9: Cities with the highest number of jobs.

- The goal is to identify and rank cities based on the number of job opportunities available, highlighting those with the highest job counts.
- The query counts and displays the number of employees in each city from the ”job-cleanData” dataset. The results are ordered in descending order based on employee count, showing the city with the highest number of employees first.
- This query applies the 'Volume' from 5 bigdata V's. The volume of data influences the efficiency of the analysis and the resources required for processing.

Goal 10: Designations among companies with the highest LinkedIn followers.

- This goal is to assess and compare the diversity of job levels within each company by counting the occurrences of different job levels. The analysis aims to provide insights into the distribution of roles across companies in the dataset.
- The query counts the occurrences of different job levels for each company in the "job-cleanData" dataset. The results provide a summary of the distribution of job levels within each company.
- In this query, the "Variety" aspect is applicable as it deals with different job levels within each company. The query analyzes a variety of job-level data for each company in the dataset.

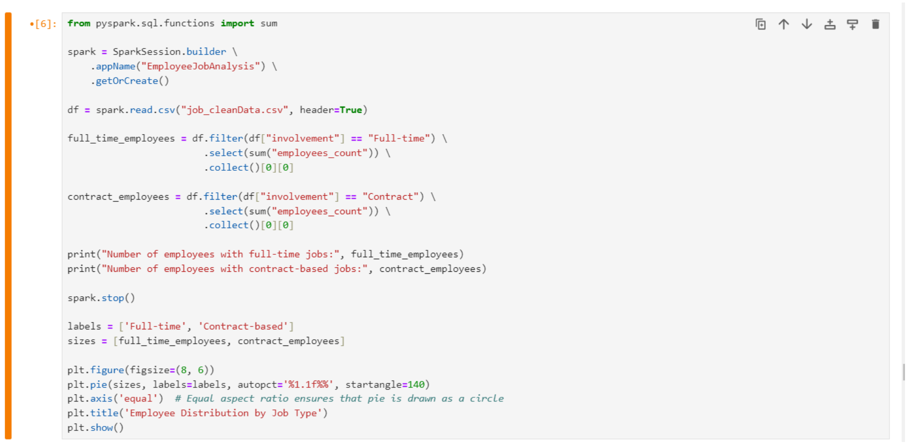
6 Project Description

First, a dataset from Kaggle is loaded into a PySpark Data Frame. Data is then represented and manipulated in a distributed fashion using the PySpark SQL or Data Frame API. To extract the needed insights from the dataset, several procedures and queries are carried out after the data frame is created.
Matplotlib, an effective Python data visualization toolkit, is used to visualize the results that were collected. With Matplotlib, you can create graphs, charts, and other visual representations of data that offer a thorough and illuminating view of the material that has been studied. This multi-step procedure enables a comprehensive approach to data analysis and interpretation by combining the advantages of Matplotlib for efficient data visualization and PySpark for data manipulation.

7 Results Summary

Goal 1: How many employees have full-time and contract-based jobs?

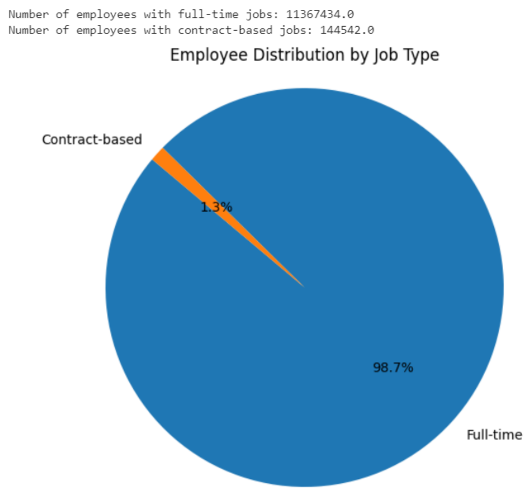
- **Implementation**



```
•[6]: from pyspark.sql.functions import sum

      spark = SparkSession.builder \
          .appName("EmployeeJobAnalysis") \
          .getOrCreate()

      df = spark.read.csv("job_cleanData.csv", header=True)

      full_time_employees = df.filter(df["involvement"] == "Full-time") \
                              .select(sum("employees_count")) \
                              .collect()[0][0]

      contract_employees = df.filter(df["involvement"] == "Contract") \
                              .select(sum("employees_count")) \
                              .collect()[0][0]

      print("Number of employees with full-time jobs:", full_time_employees)
      print("Number of employees with contract-based jobs:", contract_employees)

      spark.stop()

      labels = ['Full-time', 'Contract-based']
      sizes = [full_time_employees, contract_employees]

      plt.figure(figsize=(8, 6))
      plt.pie(sizes, labels=labels, autopct='%1.1f%%', startangle=140)
      plt.axis('equal')  # Equal aspect ratio ensures that pie is drawn as a circle
      plt.title('Employee Distribution by Job Type')
      plt.show()
```

Figure 2: Goal 1 implementation.

- **output**

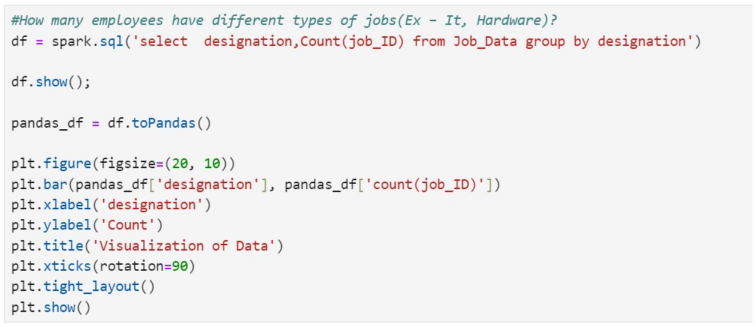


```
Number of employees with full-time jobs: 11367434.0
Number of employees with contract-based jobs: 144542.0
```

Figure 3: Goal 1 output.

Goal 2: How many employees have different types of jobs?

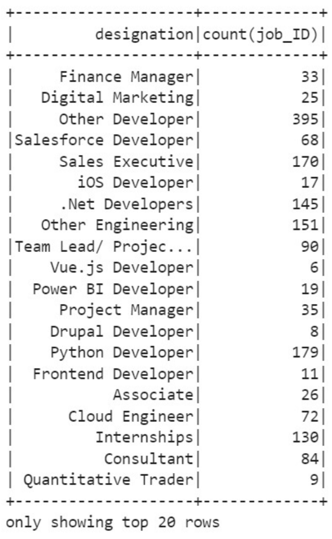
- **Implementation**



```
#How many employees have different types of jobs(Ex - It, Hardware)?
df = spark.sql('select  designation,Count(job_ID) from Job_Data group by designation')

df.show();

pandas_df = df.toPandas()

plt.figure(figsize=(20, 10))
plt.bar(pandas_df['designation'], pandas_df['count(job_ID)'])
plt.xlabel('designation')
plt.ylabel('Count')
plt.title('Visualization of Data')
plt.xticks(rotation=90)
plt.tight_layout()
plt.show()
```

Figure 4: Goal 2 implementation.

- **output**



```
+--------------------+-------------+
|         designation|count(job_ID)|
+--------------------+-------------+
|     Finance Manager|           33|
|   Digital Marketing|           25|
|     Other Developer|          395|
|Salesforce Developer|           68|
|     Sales Executive|          170|
|       iOS Developer|           17|
|     .Net Developers|          145|
|   Other Engineering|          151|
|Team Lead/ Projec...|           90|
|    Vue.js Developer|            6|
|  Power BI Developer|           19|
|     Project Manager|           35|
|    Drupal Developer|            8|
|    Python Developer|          179|
|  Frontend Developer|           11|
|           Associate|           26|
|      Cloud Engineer|           72|
|         Internships|          130|
|          Consultant|           84|
| Quantitative Trader|            9|
+--------------------+-------------+
only showing top 20 rows
```

Figure 5: Goal 2 output.

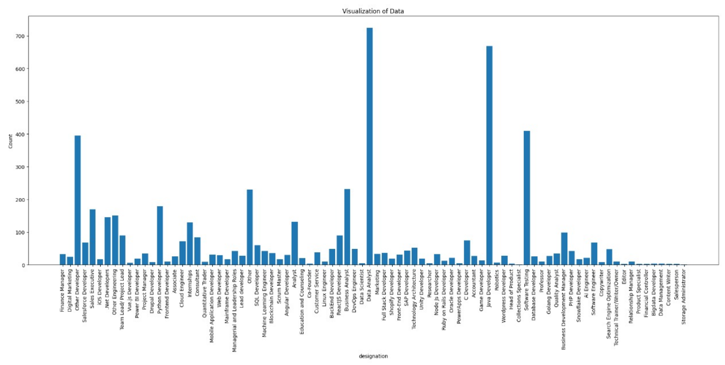
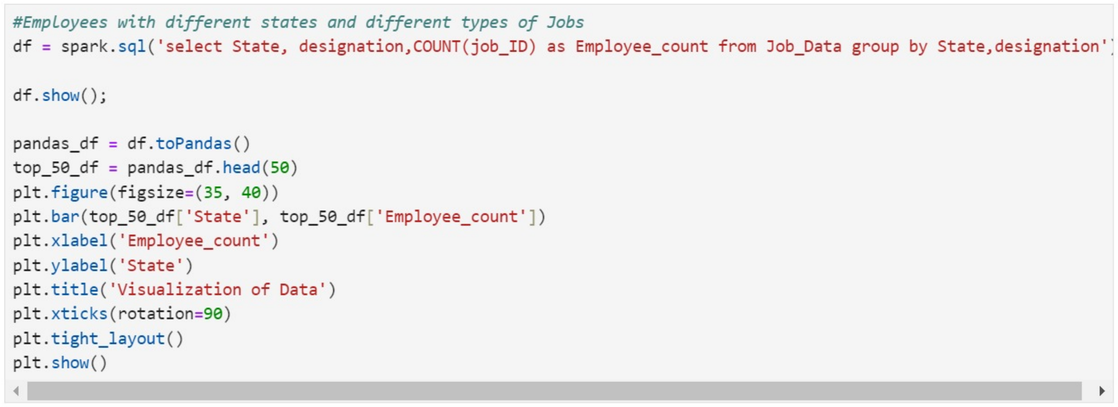


Figure 6: Goal 2 visualization.

Goal 3: Employees with different states and different types of Jobs

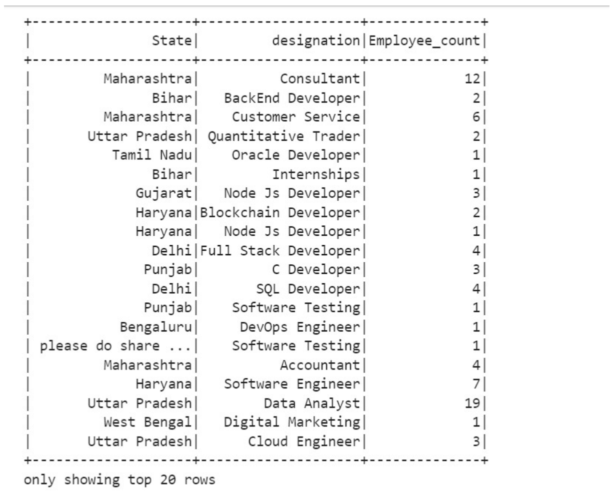
- **Implementation**



```
#Employees with different states and different types of Jobs
df = spark.sql('select State, designation,COUNT(job_ID) as Employee_count from Job_Data group by State,designation'

df.show();

pandas_df = df.toPandas()
top_50_df = pandas_df.head(50)
plt.figure(figsize=(35, 40))
plt.bar(top_50_df['State'], top_50_df['Employee_count'])
plt.xlabel('Employee_count')
plt.ylabel('State')
plt.title('Visualization of Data')
plt.xticks(rotation=90)
plt.tight_layout()
plt.show()
```

Figure 7: Goal 3 implementation.

- **output**



```
+--------------------+--------------------+--------------+
|               State|         designation|Employee_count|
+--------------------+--------------------+--------------+
|         Maharashtra|          Consultant|            12|
|               Bihar|   BackEnd Developer|             2|
|         Maharashtra|    Customer Service|             6|
|       Uttar Pradesh| Quantitative Trader|             2|
|          Tamil Nadu|    Oracle Developer|             1|
|               Bihar|         Internships|             1|
|             Gujarat|   Node Js Developer|             3|
|             Haryana|Blockchain Developer|             2|
|             Haryana|   Node Js Developer|             1|
|               Delhi|Full Stack Developer|             4|
|              Punjab|         C Developer|             3|
|               Delhi|       SQL Developer|             4|
|              Punjab|    Software Testing|             1|
|           Bengaluru|     DevOps Engineer|             1|
| please do share ...|    Software Testing|             1|
|         Maharashtra|          Accountant|             4|
|             Haryana|   Software Engineer|             7|
|       Uttar Pradesh|        Data Analyst|            19|
|         West Bengal|   Digital Marketing|             1|
|       Uttar Pradesh|      Cloud Engineer|             3|
+--------------------+--------------------+--------------+
only showing top 20 rows
```

Figure 8: Goal 3 output.

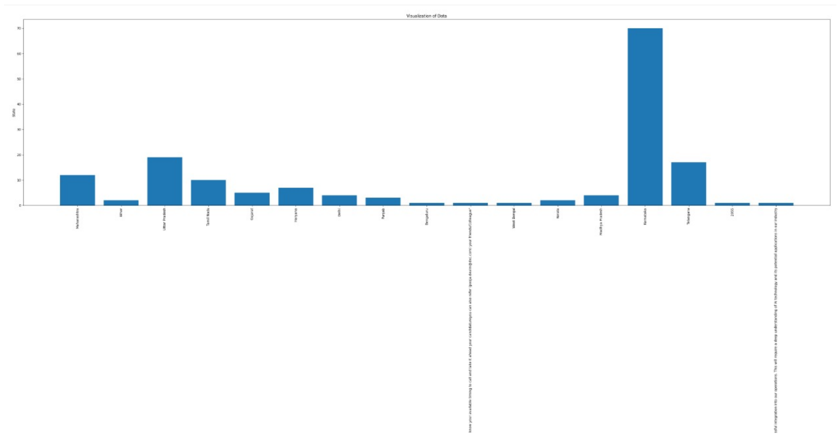
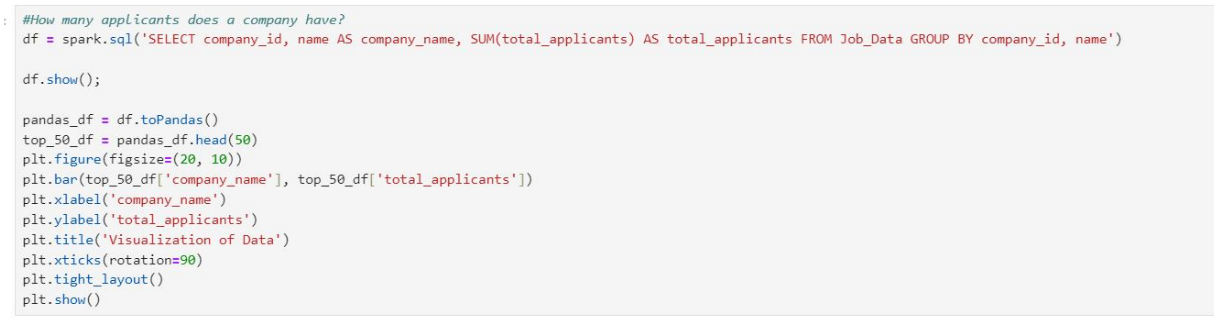


Figure 9: Goal 3 visualization.

Goal 4: How many applicants does a company have?

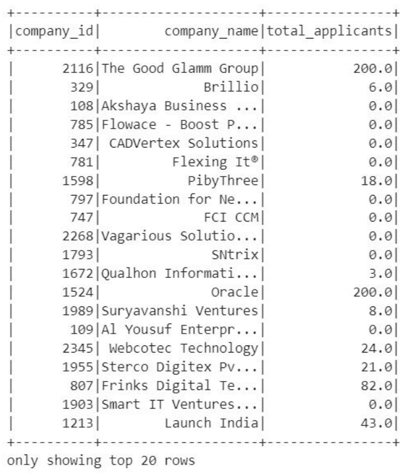
- **Implementation**



```
#How many applicants does a company have?
df = spark.sql('SELECT company_id, name AS company_name, SUM(total_applicants) AS total_applicants FROM Job_Data GROUP BY company_id, name')

df.show();

pandas_df = df.toPandas()
top_50_df = pandas_df.head(50)
plt.figure(figsize=(20, 10))
plt.bar(top_50_df['company_name'], top_50_df['total_applicants'])
plt.xlabel('company_name')
plt.ylabel('total_applicants')
plt.title('Visualization of Data')
plt.xticks(rotation=90)
plt.tight_layout()
plt.show()
```

Figure 10: Goal 4 implementation.

- **output**



```
+----------+-------------------+----------------+
|company_id|       company_name|total_applicants|
+----------+-------------------+----------------+
|      2116|The Good Glamm Group|           200.0|
|       329|            Brillio|             6.0|
|       108|Akshaya Business ...|             0.0|
|       785|Flowace - Boost P...|             0.0|
|       347| CADVertex Solutions|             0.0|
|       781|         Flexing It®|             0.0|
|      1598|          PibyThree|            18.0|
|       797|Foundation for Ne...|             0.0|
|       747|            FCI CCM|             0.0|
|      2268|Vagarious Solutio...|             0.0|
|      1793|             SNtrix|             0.0|
|      1672|Qualhon Informati...|             3.0|
|      1524|             Oracle|           200.0|
|      1989|Suryavanshi Ventures|             8.0|
|       109|Al Yousuf Enterpr...|             0.0|
|      2345| Webcotec Technology|            24.0|
|      1955|Sterco Digitex Pv...|            21.0|
|       807|Frinks Digital Te...|            82.0|
|      1903|Smart IT Ventures...|             0.0|
|      1213|       Launch India|            43.0|
+----------+-------------------+----------------+
only showing top 20 rows
```

Figure 11: Goal 4 output.

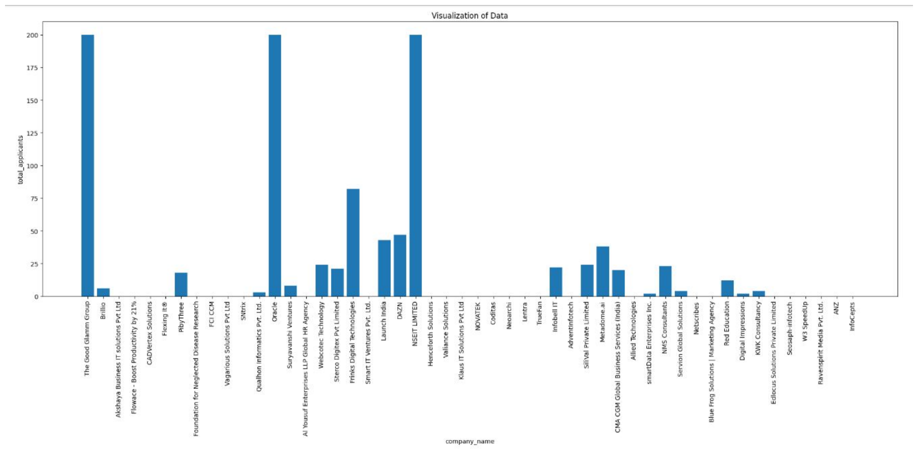
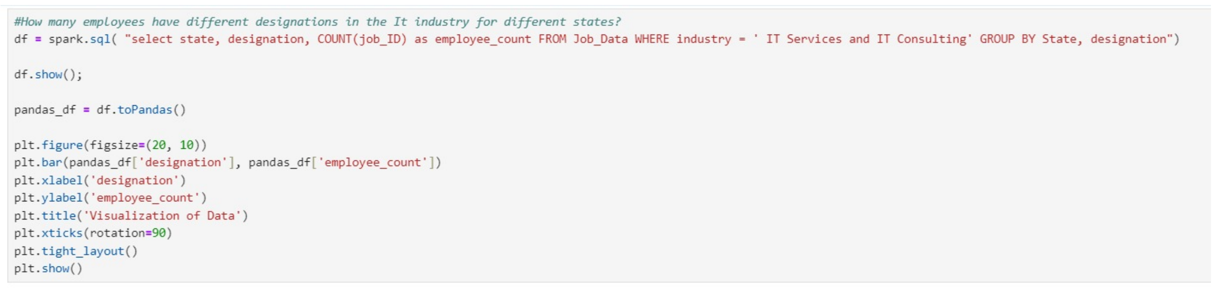


Figure 12: Goal 4 visualization.

Goal 5: How many employees have different designations in the It industry for different states?

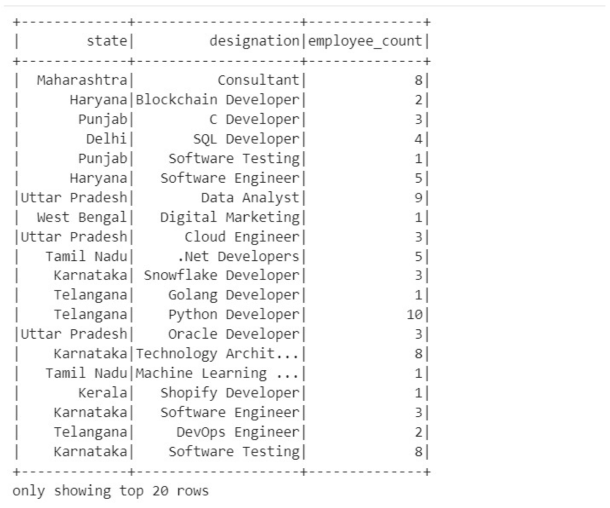
- **Implementation**



```
#How many employees have different designations in the It industry for different states?
df = spark.sql( "select state, designation, COUNT(job_ID) as employee_count FROM Job_Data WHERE industry = ' IT Services and IT Consulting' GROUP BY State, designation")

df.show();

pandas_df = df.toPandas()

plt.figure(figsize=(20, 10))
plt.bar(pandas_df['designation'], pandas_df['employee_count'])
plt.xlabel('designation')
plt.ylabel('employee_count')
plt.title('Visualization of Data')
plt.xticks(rotation=90)
plt.tight_layout()
plt.show()
```

Figure 13: Goal 5 implementation.

- **output**



```
+-------------+--------------------+--------------+
|        state|         designation|employee_count|
+-------------+--------------------+--------------+
|  Maharashtra|          Consultant|             8|
|      Haryana|Blockchain Developer|             2|
|       Punjab|         C Developer|             3|
|        Delhi|       SQL Developer|             4|
|       Punjab|    Software Testing|             1|
|      Haryana|   Software Engineer|             5|
|Uttar Pradesh|        Data Analyst|             9|
|  West Bengal|   Digital Marketing|             1|
|Uttar Pradesh|      Cloud Engineer|             3|
|   Tamil Nadu|     .Net Developers|             5|
|    Karnataka| Snowflake Developer|             3|
|    Telangana|    Golang Developer|             1|
|    Telangana|    Python Developer|            10|
|Uttar Pradesh|    Oracle Developer|             3|
|    Karnataka|Technology Archit...|             8|
|   Tamil Nadu|Machine Learning ...|             1|
|       Kerala|   Shopify Developer|             1|
|    Karnataka|   Software Engineer|             3|
|    Telangana|     DevOps Engineer|             2|
|    Karnataka|    Software Testing|             8|
+-------------+--------------------+--------------+
only showing top 20 rows
```

Figure 14: Goal 5 output.

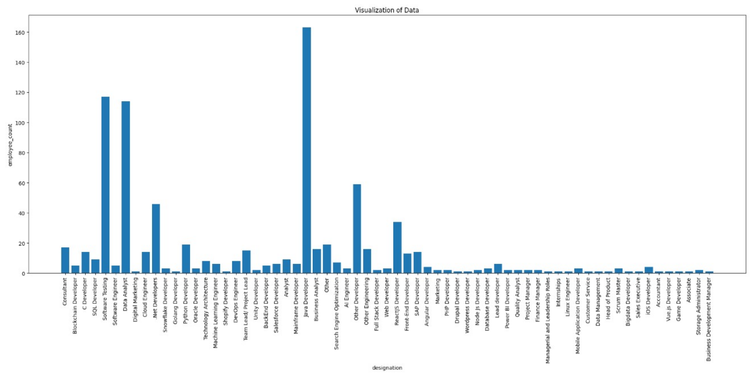
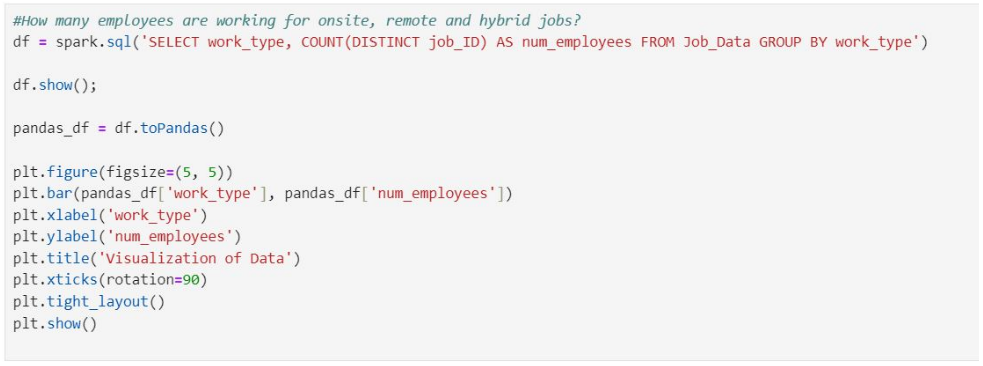


Figure 15: Goal 5 visualization.

Goal 6: How many employees are working for onsite and remote jobs?

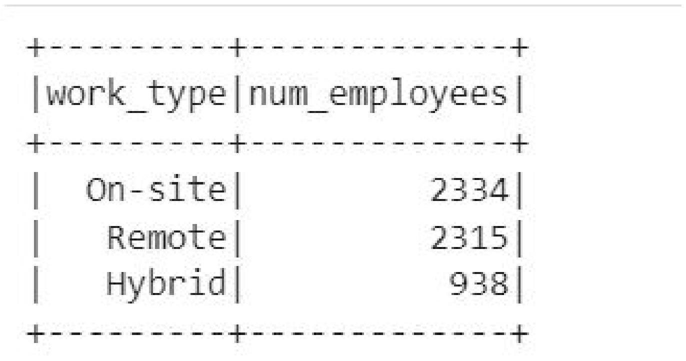
- **Implementation**



```
#How many employees are working for onsite, remote and hybrid jobs?
df = spark.sql('SELECT work_type, COUNT(DISTINCT job_ID) AS num_employees FROM Job_Data GROUP BY work_type')

df.show();

pandas_df = df.toPandas()

plt.figure(figsize=(5, 5))
plt.bar(pandas_df['work_type'], pandas_df['num_employees'])
plt.xlabel('work_type')
plt.ylabel('num_employees')
plt.title('Visualization of Data')
plt.xticks(rotation=90)
plt.tight_layout()
plt.show()
```

Figure 16: Goal 6 implementation.

- **output**



```
+---------+-------------+
|work_type|num_employees|
+---------+-------------+
|  On-site|         2334|
|   Remote|         2315|
|   Hybrid|          938|
+---------+-------------+
```

Figure 17: Goal 6 output.

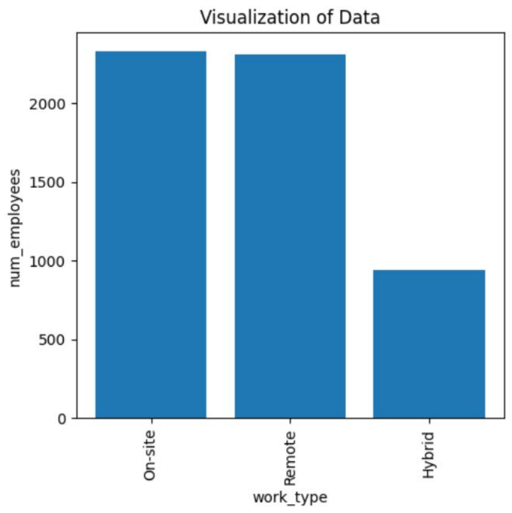
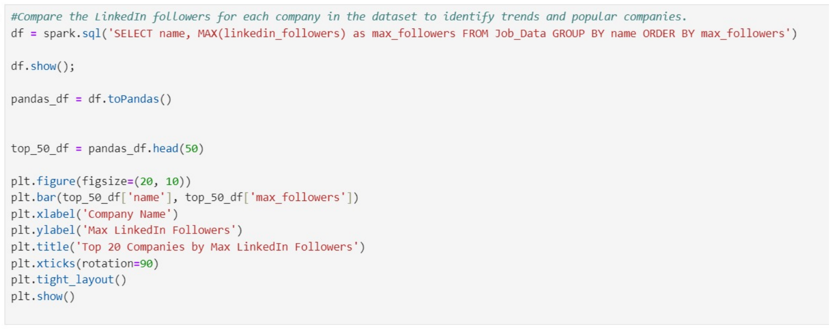


Figure 18: Goal 6 visualization.

Goal 7: Compare the LinkedIn followers for each company in the dataset to identify trends and popular companies.

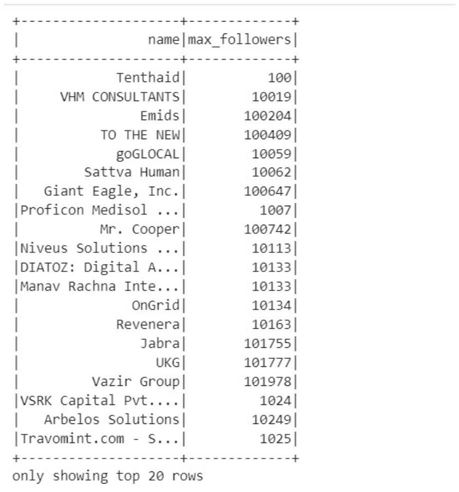
- **Implementation**



```
#Compare the LinkedIn followers for each company in the dataset to identify trends and popular companies.
df = spark.sql('SELECT name, MAX(linkedin_followers) as max_followers FROM Job_Data GROUP BY name ORDER BY max_followers')

df.show();

pandas_df = df.toPandas()


top_50_df = pandas_df.head(50)

plt.figure(figsize=(20, 10))
plt.bar(top_50_df['name'], top_50_df['max_followers'])
plt.xlabel('Company Name')
plt.ylabel('Max LinkedIn Followers')
plt.title('Top 20 Companies by Max LinkedIn Followers')
plt.xticks(rotation=90)
plt.tight_layout()
plt.show()
```

Figure 19: Goal 7 implementation.

- **output**



```
+--------------------+------------+
|                name|max_followers|
+--------------------+------------+
|            Tenthaid|         100|
|     VHM CONSULTANTS|       10019|
|               Emids|      100204|
|          TO THE NEW|      100409|
|            goGLOCAL|       10059|
|        Sattva Human|       10062|
|   Giant Eagle, Inc.|      100647|
|Proficon Medisol ...|        1007|
|          Mr. Cooper|      100742|
|Niveus Solutions ...|       10113|
|DIATOZ: Digital A...|       10133|
|Manav Rachna Inte...|       10133|
|              OnGrid|       10134|
|            Revenera|       10163|
|               Jabra|      101755|
|                 UKG|      101777|
|         Vazir Group|      101978|
|VSRK Capital Pvt....|        1024|
|   Arbelos Solutions|       10249|
|Travomint.com - S...|        1025|
+--------------------+------------+
only showing top 20 rows
```

Figure 20: Goal 7 output.

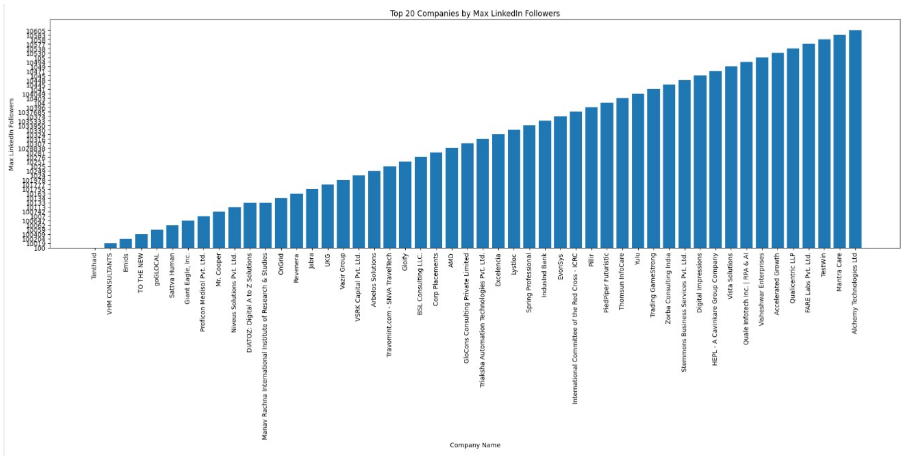
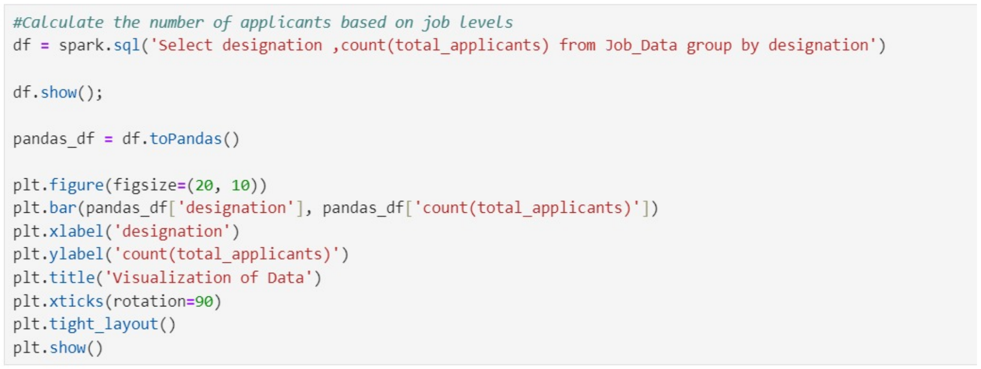


Figure 21: Goal 7 visualization.

Goal 8: Calculate the number of applicants based on job levels.

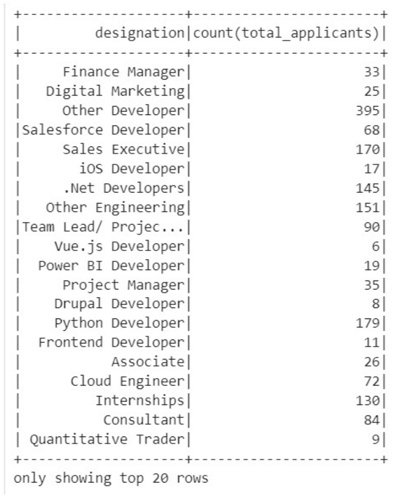
- **Implementation**



```
#Calculate the number of applicants based on job levels
df = spark.sql('Select designation ,count(total_applicants) from Job_Data group by designation')

df.show();

pandas_df = df.toPandas()

plt.figure(figsize=(20, 10))
plt.bar(pandas_df['designation'], pandas_df['count(total_applicants)'])
plt.xlabel('designation')
plt.ylabel('count(total_applicants)')
plt.title('Visualization of Data')
plt.xticks(rotation=90)
plt.tight_layout()
plt.show()
```

Figure 22: Goal 8 implementation.

- **output**



```
+--------------------+----------------------+
|         designation|count(total_applicants)|
+--------------------+----------------------+
|     Finance Manager|                    33|
|   Digital Marketing|                    25|
|     Other Developer|                   395|
|Salesforce Developer|                    68|
|     Sales Executive|                   170|
|       iOS Developer|                    17|
|      .Net Developers|                  145|
|   Other Engineering|                   151|
|Team Lead/ Projec...|                    90|
|    Vue.js Developer|                     6|
|  Power BI Developer|                    19|
|     Project Manager|                    35|
|    Drupal Developer|                     8|
|    Python Developer|                   179|
|  Frontend Developer|                    11|
|           Associate|                    26|
|      Cloud Engineer|                    72|
|         Internships|                   130|
|          Consultant|                    84|
|  Quantitative Trader|                    9|
+--------------------+----------------------+
only showing top 20 rows
```

Figure 23: Goal 8 output.

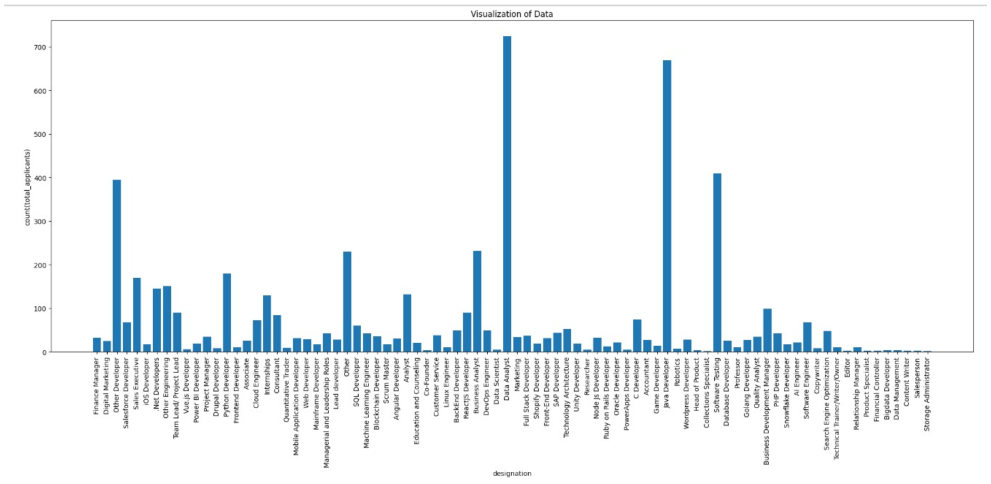
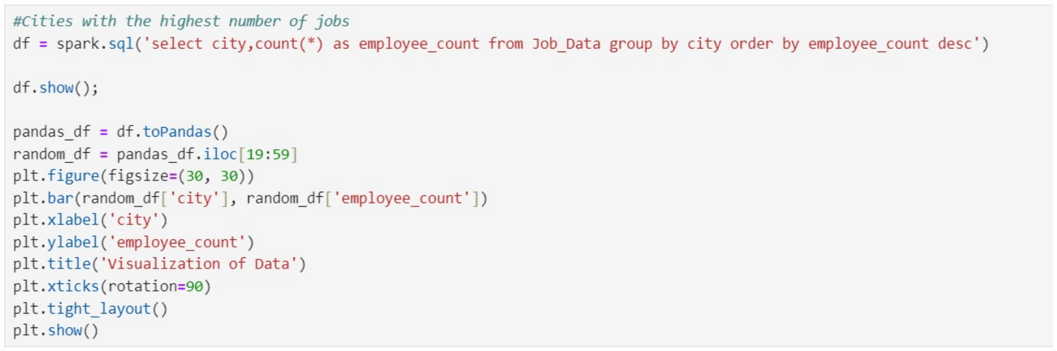


Figure 24: Goal 8 visualization.

Goal 9: Cities with the highest number of jobs.

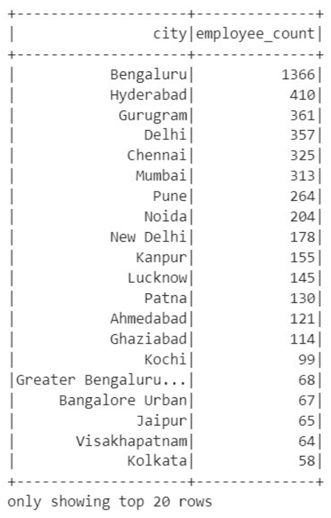
- **Implementation**



```
#Cities with the highest number of jobs
df = spark.sql('select city,count(*) as employee_count from Job_Data group by city order by employee_count desc')

df.show();

pandas_df = df.toPandas()
random_df = pandas_df.iloc[19:59]
plt.figure(figsize=(30, 30))
plt.bar(random_df['city'], random_df['employee_count'])
plt.xlabel('city')
plt.ylabel('employee_count')
plt.title('Visualization of Data')
plt.xticks(rotation=90)
plt.tight_layout()
plt.show()
```

Figure 25: Goal 9 implementation.

- **output**



```
+-------------------+-------------+
|               city|employee_count|
+-------------------+-------------+
|          Bengaluru|         1366|
|          Hyderabad|          410|
|           Gurugram|          361|
|              Delhi|          357|
|            Chennai|          325|
|             Mumbai|          313|
|               Pune|          264|
|              Noida|          204|
|          New Delhi|          178|
|             Kanpur|          155|
|            Lucknow|          145|
|              Patna|          130|
|          Ahmedabad|          121|
|          Ghaziabad|          114|
|              Kochi|           99|
|Greater Bengaluru...|           68|
|     Bangalore Urban|           67|
|             Jaipur|           65|
|       Visakhapatnam|           64|
|            Kolkata|           58|
+-------------------+-------------+
only showing top 20 rows
```

Figure 26: Goal 9 output.

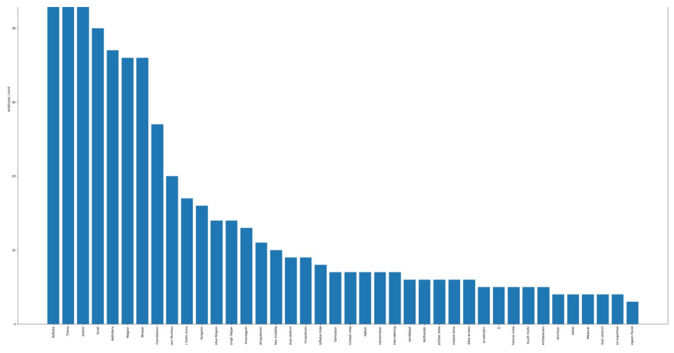
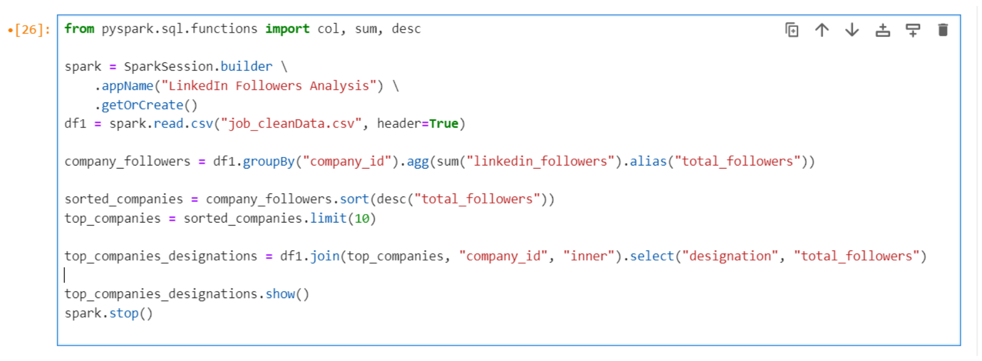


Figure 27: Goal 9 visualization.

Goal 10: Designations among companies with the highest LinkedIn followers.

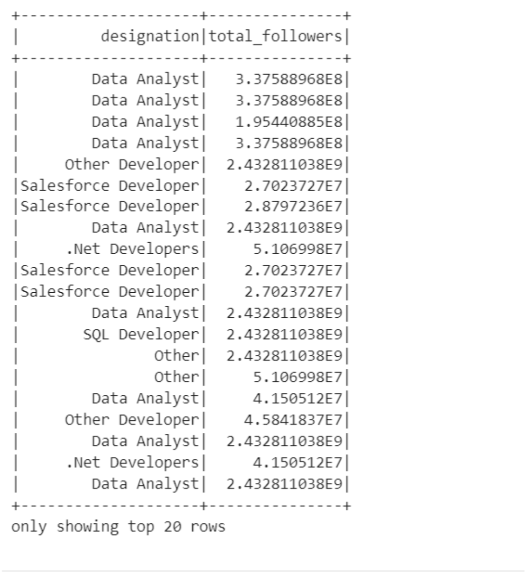
- **Implementation**



```
•[26]: from pyspark.sql.functions import col, sum, desc

spark = SparkSession.builder \
    .appName("LinkedIn Followers Analysis") \
    .getOrCreate()
df1 = spark.read.csv("job_cleanData.csv", header=True)

company_followers = df1.groupBy("company_id").agg(sum("linkedin_followers").alias("total_followers"))

sorted_companies = company_followers.sort(desc("total_followers"))
top_companies = sorted_companies.limit(10)

top_companies_designations = df1.join(top_companies, "company_id", "inner").select("designation", "total_followers")

top_companies_designations.show()
spark.stop()
```

Figure 28: Goal 10 implementation.

- **output**



```
+--------------------+---------------+
|         designation|total_followers|
+--------------------+---------------+
|        Data Analyst|   3.37588968E8|
|        Data Analyst|   3.37588968E8|
|        Data Analyst|   1.95440885E8|
|        Data Analyst|   3.37588968E8|
|     Other Developer| 2.432811038E9|
|Salesforce Developer|    2.7023727E7|
|Salesforce Developer|    2.8797236E7|
|        Data Analyst| 2.432811038E9|
|      .Net Developers|     5.106998E7|
|Salesforce Developer|    2.7023727E7|
|Salesforce Developer|    2.7023727E7|
|        Data Analyst| 2.432811038E9|
|       SQL Developer| 2.432811038E9|
|               Other| 2.432811038E9|
|               Other|     5.106998E7|
|        Data Analyst|     4.150512E7|
|     Other Developer|    4.5841837E7|
|        Data Analyst| 2.432811038E9|
|      .Net Developers|     4.150512E7|
|        Data Analyst| 2.432811038E9|
+--------------------+---------------+
only showing top 20 rows
```

Figure 29: Goal 10 output.

8 Conclusion

The Workforce Dynamics Explorer project, employing PySpark provided actionable insights into industry trends, job types, demographics, and company popularity. Uncovering nuances in job openings, applicant trends, and corporate profiles, the project facilitates informed HR and talent acquisition decision-making.
It enhances strategic workforce planning, refines recruitment strategies, and improves company positioning. Despite challenges in data quality and privacy, the project's impact is substantial, fostering organizational adaptability. Continuous monitoring and exploration of advanced analytics will further refine predictive capabilities for continued relevance in the dynamic workforce landscape.

9 Citations

- GitHub
- Dataset