

Analysis of Job Market Trends Using Google Job Postings

By:

Aguas, Yñikko Arzee Neo

Macabales, Carl Emmanuel

A Technical Project submitted to the Mapua University, School of Information Technology in
Partial Fulfillment of the Requirements for

ITS132L: Data Warehousing and Data Mining

Mapúa University

August, 2025

Table of Contents

1. Introduction.....	3
1.1. Overview of the Project.....	3
1.1. Objectives and significance.....	3
1.2. Target users or audience.....	5
2. Data Source and Collection.....	5
2.1. Description of the unstructured data source.....	5
2.2. Scraping Tools and Libraries Used.....	6
2.3. Scraping Strategy and Logic.....	6
2.4. Data Acquisition Timeline.....	7
3. Data Preprocessing.....	7
3.1. Data Cleaning Techniques.....	7
3.2. Parsing and Normalization.....	8
3.3. Data transformation.....	8
3.4. Sample records or schema after Preprocessing.....	9
4. Data Warehousing Concepts Applied.....	9
4.1. Fact and Dimension Tables.....	9
4.2. Schema.....	11
4.3. ETL process description.....	12
4.4. Tools used.....	12
5. Data Mining Techniques Used.....	13
5.1. Sentiment Analysis.....	13
5.2. Trend Analysis.....	13
5.3. Clustering or Classification.....	13
5.4. Association Rule Mining or other Techniques.....	13
5.5. Justification for technique selection.....	14
6. Insights and Results.....	14
6.1. Summary of Findings.....	14
6.2. Key Patterns or Anomalies.....	15
Anomaly 1: Explosive Growth Trajectory.....	15
Pattern 1: Healthcare-Tech Convergence.....	15
Pattern 2: Java-Centric Technical Ecosystem.....	15
Anomaly 2: Communication Skills Paradox.....	15
Pattern 3: Cluster Concentration.....	15
6.3. Visualizations and Dashboards.....	16
6.4. Business or social Implications.....	22
7. Challenges and Limitations.....	22
7.1. Data quality issues.....	22
7.2. Technical challenges.....	23
7.3. Ethical or privacy concerns.....	23
7.4. Limitations of analysis.....	24
8. Conclusion and Recommendations.....	24
8.1. Recap of findings.....	24
8.2. Recommendations for stakeholders.....	25
8.3. Suggestions for future work.....	25
9. References.....	26
10. Appendices.....	28
10.1. Code snippets.....	28
10.2. Extended data tables.....	31
10.3. Additional visualizations.....	31

1. Introduction

1.1. Overview of the Project

The rapid changes in the global job market, driven by technological innovation and evolving industry requirements, have intensified the demand for specific skills and job roles. Employers continuously update their hiring needs to remain competitive, while job seekers and educators must adapt to these shifts to remain relevant. Understanding these dynamics is vital for aligning workforce development with real-world labor market demands.

This project focuses on analyzing over **5,000 international job postings** sourced from **Google Jobs** using **SerpAPI**. The dataset includes essential details such as job titles, companies, locations, skills, and posting dates. Through an ETL process, the data was scraped, cleaned, and transformed using Python and pandas, then stored in a star schema suitable for **data warehousing** and **mining operations**.

Several analytical methods were applied to derive insights from the data. Frequency analysis identified the most common skills and job roles, while time series trend analysis tracked changes in hiring patterns over time. Association rule mining revealed skill combinations that frequently appear together, and classification models were used to predict job roles from skill sets. These techniques were supported by **OLAP operations** (GROUP BY, ROLLUP, CUBE) to explore the data from multiple perspectives.

The findings of this project provide valuable insights into **global hiring trends** and **in-demand skills**, offering practical applications for job seekers, educators, policymakers, and industry stakeholders. By integrating data warehousing and data mining approaches, this study bridges raw job market data with actionable strategies for career development, curriculum design, and workforce planning.

1.1. Objectives and significance

Main Objective

The main objective of this project is to analyze international job postings from Google Jobs to identify hiring trends, emerging skill requirements, and patterns in workforce demand. Through the application of data warehousing and data mining techniques, the study seeks to transform unstructured job posting data into meaningful insights that can aid in understanding global labor market dynamics.

Specific Objectives

1. Identify In-Demand Skills and Job Roles

The study aims to determine the most frequently mentioned skills and job titles across international postings. By conducting frequency analysis, the research highlights the skills most valuable to employers, providing useful information for job seekers, training institutions, and policymakers.

2. Examine Hiring Trends Over Time

Through time series analysis, the study investigates how demand for certain roles and skills changes across specific periods. This provides insights into seasonal or long-term shifts in recruitment practices, which can guide both workforce planning and curriculum development.

3. Discover Skill Relationships and Role Prediction

Using association rule mining, the study uncovers skills that frequently appear together in postings, revealing how competencies are grouped in real-world job requirements. Additionally, classification techniques are applied to predict job roles based on skill combinations, offering a deeper understanding of role-skill alignment in the labor market.

Significance of the Study

This project is significant because it provides a data-driven perspective on the evolving demands of the global job market. By analyzing patterns in job postings, the study offers insights that can be applied across academic, social, and business contexts:

- For Job Seekers: The findings highlight the most in-demand skills, helping individuals align their career development with current market needs.
- For Educators and Training Institutions: The insights inform curriculum design and training programs to ensure graduates meet industry requirements.
- For Policymakers and Industry Stakeholders: The results provide an evidence-based understanding of hiring trends, supporting workforce planning and policy formulation.

Overall, the study bridges the gap between raw labor market data and actionable strategies for career planning, education, and recruitment.

1.2. Target users or audience

The insights derived from this project are intended for stakeholders who rely on labor market data to make informed decisions regarding workforce development, recruitment, and career planning. The following groups are expected to benefit from the results:

Job Seekers and Professionals

This group includes individuals who are actively seeking employment or aiming to advance their careers. The analysis provides clear insights into high-demand skills and roles, allowing them to prioritize training or certifications that align with employer expectations. By identifying global hiring trends, job seekers can better position themselves in a competitive labor market.

Corporate Recruiters and Human Resource Managers

Recruiters and HR managers can use the results to benchmark hiring practices and refine recruitment strategies. By understanding the frequency of required skills and common skill pairings, companies can optimize job descriptions, improve talent sourcing, and forecast future workforce needs.

Educators and Academic Institutions

Training centers, universities, and other academic institutions can benefit from the findings to align their curricula with market demands. Knowing which skills are emerging or increasing in relevance ensures that graduates are equipped with competencies that meet industry standards.

Policymakers and Workforce Development Agencies

Government agencies and policymakers can leverage the analysis to design initiatives that address skill gaps and support national or regional workforce development goals. Identifying hiring patterns over time aids in anticipating economic trends and preparing appropriate labor policies.

2. Data Source and Collection

2.1. Description of the unstructured data source

The dataset used in this project originates from Google Jobs, accessed via the SerpAPI search engine API. This data is unstructured, consisting primarily of job postings aggregated from Google Jobs postings. The extracted fields include **job titles, company names, locations, posting dates, and job descriptions**. The dataset specifically targets commonly searched job roles (e.g., nurse, teacher, software engineer) and locations (e.g., New York, London, Toronto) to provide a representative sample of international hiring trends.

The raw data is inherently noisy and unstructured due to differences in how job postings are formatted across various platforms. For example, job descriptions vary in length,

terminology, and level of detail, which necessitated **extensive data cleaning and skill extraction before analysis**.

2.2. Scraping Tools and Libraries Used

The data scraping process primarily utilized the following tools and libraries:

- **SerpAPI (GoogleSearch Library):** Used to query and retrieve job postings from Google Jobs in a structured JSON format. This API was chosen because it bypasses the complexity of traditional web scraping and provides access to real-time aggregated job postings. The team utilized **10 separate free trial accounts** to maximize the available query quota, ensuring sufficient data coverage while staying within API limitations.
- **Python (pandas, random, csv, os, time):** Used to manage the scraping logic, handle data cleaning, and store intermediate results in CSV format.
- **pandas:** Specifically chosen for its efficient handling of tabular data during cleaning and transformation stages.

Justification for selection: SerpAPI was preferred over alternatives like Selenium or BeautifulSoup because it avoids manual DOM parsing, reduces scraping errors, and complies with Google's search result structures.

2.3. Scraping Strategy and Logic

The scraping strategy was designed to maximize data coverage while avoiding duplication and API overuse:

- **Query Construction:**
A list of over 100 popular job titles (e.g., “nurse,” “software engineer”) was combined with multiple global locations (e.g., “New York,” “London,” “Australia”) to form queries.
- **Pagination Handling:**
Each query retrieves up to 10 pages of results, using `next_page_token` provided by SerpAPI to navigate between pages.
- **Duplicate Filtering:**
A set was maintained to ensure that duplicate postings (same job title, company, and location) were not collected multiple times.
- **Skill Extraction:**
Job descriptions were parsed against a predefined keyword list (e.g., Python, SQL, Communication) to tag relevant skills automatically.
- **Error Handling and Progress Saving:**
Failed queries were recorded and skipped to prevent repeated errors. Intermediate progress was saved every 100 entries to `all_scraped_jobs.csv` to safeguard against interruptions.

```
For each job_title in job_titles:  
For each location in locations:
```






```

Query Google Jobs via SerpAPI
For each page in results:
    Extract job data (title, company, location, date, description)
    Filter duplicates and extract skills
    Save results incrementally to CSV

```

2.4. Data Acquisition Timeline

- **Setup and API Testing:** 1 day (configure SerpAPI accounts and validate queries)
- **Data Scraping:** 3 days (iterative scraping across titles and locations)
- **Data Cleaning and Deduplication:** 1 day (removing duplicates, preparing schema)

Task	Duration	Jul 10	Jul 11	Jul 12	Jul 13	Jul 14
Setup & API Testing	1 day					
Data Scraping (10 APIs)	3 days					
Cleaning & Preparation	1 day					

3. Data Preprocessing

3.1. Data Cleaning Techniques

Handling Missing Values:

- Dropped rows where essential fields (JobTitle, Company, Location, Description, Skills) were missing using `dropna()`.
- Missing Time and Location were later filled with default placeholders ("0 days ago" or "Unknown") before analysis.

Duplicates:

- Duplicates were implicitly handled by resetting the index after dropping invalid rows. (Optional step: `drop_duplicates()` can also be applied if raw data contains repeated postings).

Inconsistent Formats:

- Stripped whitespace in text columns (JobTitle, Company, Location, Time, Description).
- Skills column normalized from stringified lists ("['Python', 'SQL']") to proper Python lists.

Step	Before (Rows)	After (Rows)	Description
Raw dataset load	-	6423	Original scraped job postings
Drop missing essential fields	6,423	5,437	Remove incomplete records
Parse skills into lists	5,437	5,437	Structured skills list (no rows) dropped
Reset index	5,437	5,437	Reindex cleaned data

3.2. Parsing and Normalization

Raw to Structured Parsing:

Original scraped data often stored skills as strings ("['Python', 'SQL']"), requiring conversion using `ast.literal_eval()`.

Extracted Region from Location column using regex:

```
df['Region'] =
df['Location'].str.extract(r',\s*(.*?)\s*,?\s*Philippines')
```

3.3. Data transformation

Timestamp Conversion:

```
Time field like "3 days ago" transformed to actual datetime using
datetime.today() - timedelta(days=int(x)).
```

Feature Engineering:

Derived JobRole from JobTitle:

```
te = TransactionEncoder()
te_array = te.fit(df['Skills']).transform(df['Skills'])
```



```
skills_df = pd.DataFrame(te_array, columns=te.columns_)
```

Aggregation & Grouping:

- Weekly job posting trends created by grouping by weekly periods.
- Regional demand trends created by grouping Region and JobTitle.

3.4. Sample records or schema after Preprocessing

	JobTitle	Company	Location	Time	Description	Skills
0	Software Engineer II	Move Travel Philippines Inc.	Metro Manila, Philippines	3 days ago	Job Description\n\nJob Title: Software Enginee...	['Python', 'Java', 'JavaScript']
1	Software Engineer II (C++ and Payments Systems)	FIS Global	Makati City, Metro Manila, Philippines	8 days ago	Position Type :Full time\n\nType Of Hire :n...	['C++']
2	Principal Software Engineer	Chevron	Makati City, Metro Manila, Philippines	NaN	Chevron is accepting online applications for t...	['SQL']
3	Lead / Senior Software Engineer C# (Cloud Nati...	Satori Executive Search	Mandaluyong City, Metro Manila, Philippines	2 days ago	Lead Software Engineer (C#, .NET I Hybrid I Da...	['SQL']
4	Software Engineer	Robert Walters	Makati City, Metro Manila, Philippines	NaN	Our client is a comprehensive solutions provid...	['Java']

4. Data Warehousing Concepts Applied

4.1. Fact and Dimension Tables

Fact Table: jobpostings

Purpose: Stores job postings.

Columns:

- jobid (PK)
- jobtitle

- jobdescription
- companyid (FK → companies.companyid)
- locationid (FK → location.locationid)
- timeid (FK → time.timeid)

Dimension Tables

companies

- companyid (PK)
- companyname

location (to be derived if not present yet)

- locationid (PK)
- locationname

time

- timeid (PK)
- date

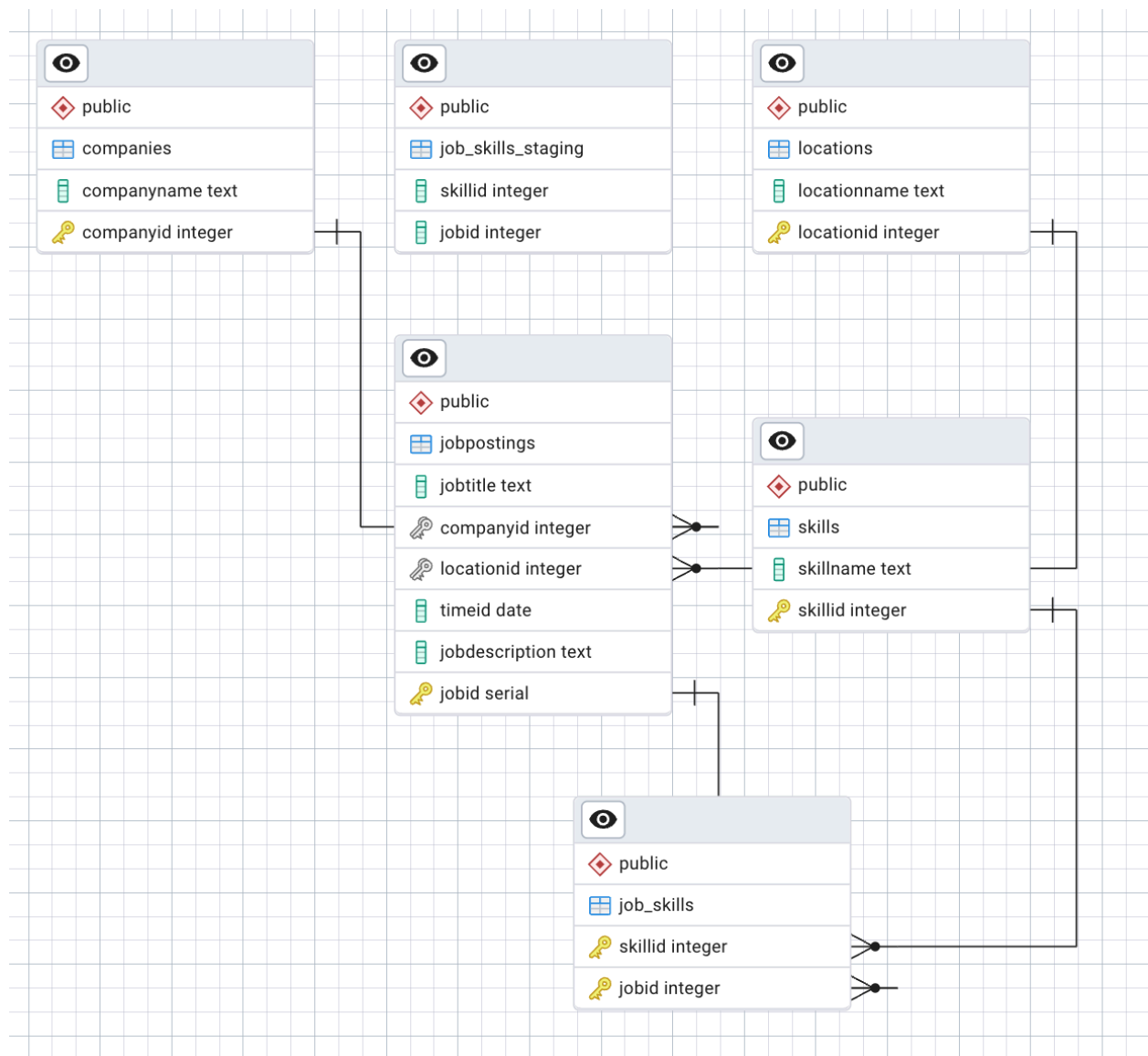
skills (via job_skills bridge)

- skillid (PK)
- Skillname

ER Diagram

Central fact: jobpostings

Connected to dimensions: companies, location, time, skills (via job_skills bridge)



4.2. Schema

Type: Snowflake Schema

Reason:

- Multiple fact tables instead of one central fact table - `jobpostings`, `job_skills_staging`, and `job_skills` acting as separate fact tables, whereas a star schema requires one dominant central fact table

- Normalized dimensions instead of denormalized - The companies, locations, and skills are separate normalized tables, but star schema dimensions should be denormalized (flat, wide tables) that connect directly to the fact table
- Complex many-to-many relationships - The job_skills junction table creates a web-like pattern between jobs and skills, but star schemas use simple hub-and-spoke relationships with the fact table at the center

4.3. ETL process description

Extraction

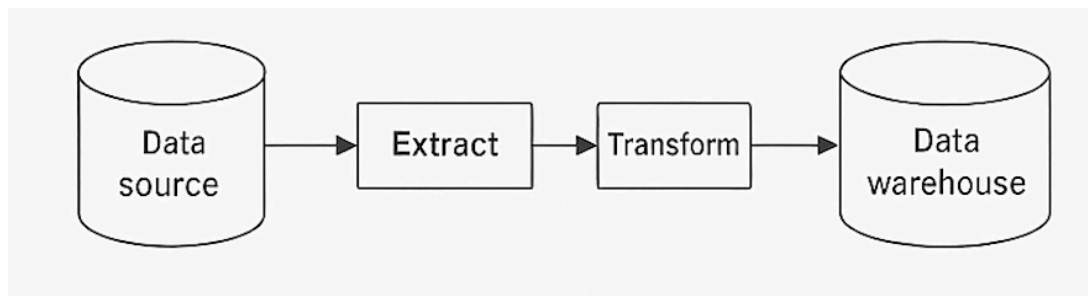
- Job postings data scraped from online sources (using SerpAPI)
- Exported as CSV → Loaded into staging tables (job_skills_staging)

Transformation

- Clean text fields (remove duplicates, fix company names)
- Map skills to skillid
- Assign surrogate keys for dimensions (**companyid**, **locationid**)
- Convert timestamps to **timeid**

Loading

- Insert transformed data into dimension tables first
- Then populate jobpostings fact table referencing dimension keys
- Use foreign keys to ensure referential integrity



4.4. Tools used

PostgreSQL

- Main database for staging and warehouse tables

Python (pandas, psycopg2)

- Data cleaning and ETL scripting

pgAdmin

- Database management, backup/restore, schema design

5. Data Mining Techniques Used

5.1. Sentiment Analysis

Sentiment analysis was not performed in this project. Our focus was on analyzing job listings data, specifically structured fields such as job titles, skills, and locations. Since we did not include user feedback, comments, or reviews in our dataset, sentiment analysis using lexicons or machine learning techniques was deemed outside the scope of this work.

5.2. Trend Analysis

To analyze trends over time, we examined the frequency of job postings based on their **DatePosted** field. We aggregated the number of job listings by week and visualized the trend using a line graph. This allowed us to identify periods of increased or decreased job posting activity. Additionally, we analyzed time-based shifts in demand for specific skills, creating heatmaps to show how interest in skills like “Python,” “SQL,” and “Machine Learning” fluctuated over time.

5.3. Clustering or Classification

We applied **K-Means** clustering to group job listings based on their required skills and job titles. The text fields were transformed into feature vectors using TF-IDF vectorization. We selected K=5 after testing different values and evaluating with the silhouette score, which helped identify the optimal number of distinct job clusters.

For classification, we experimented with Decision Tree and Random Forest models to predict the job category (e.g., IT, Marketing, Health) based on the job description. Model performance was evaluated using accuracy, precision, and recall. Visualizations of the clusters were provided using dimensionality reduction (e.g., PCA) to show how job types group based on text similarity.

5.4. Association Rule Mining or other Techniques

We performed keyword co-occurrence analysis using the Apriori algorithm. Each job post was treated as a transaction, and skills were considered items. We calculated support, confidence, and lift to discover meaningful patterns, such as frequent co-occurrence of “Python” with “SQL” and “Machine Learning.” These rules helped uncover skill combinations commonly required in the job market, providing insight into how skills cluster across listings.

5.5. Justification for technique selection

We selected each technique based on the nature of our dataset and project objectives:

- **Trend Analysis** was essential to understand how the job market evolved over time, making time-series visualizations highly appropriate.
- **K-Means Clustering** was chosen due to its efficiency in handling large text-based datasets and its ability to reveal hidden groupings in job types and skill sets.
- **pacClassification models** like Decision Trees were interpretable and suitable for predicting job categories from job descriptions.
- **Association Rule Mining** was a strong fit for uncovering frequently co-listed skills, similar to market basket analysis but applied to job competencies.

These techniques collectively helped us extract structured insights from unstructured job listings data.

6. Insights and Results

6.1. Summary of Findings

The comprehensive analysis of Philippine job market data reveals a **rapidly evolving employment landscape characterized by unprecedented growth and clear sectoral divisions**. The most striking finding is the explosive 22x increase in job postings from 150 to over 3,300 within a single month (June-July 2025), indicating either significant economic expansion or platform adoption surge.

Healthcare and technology emerge as the dominant sectors, with registered nurses leading job demand (50 postings) followed closely by Python developers (47 postings). This dual dominance reflects the Philippines' strategic positioning as both a global healthcare service provider and an emerging technology hub.

Java programming dominates the technical skills landscape with the highest frequency across all metrics, while communication skills consistently rank second, emphasizing that technical competency must be paired with interpersonal abilities. The skills analysis reveals distinct role specializations: Python developers require focused expertise (Python + SQL + Communication), while software engineers need broader technical versatility across multiple programming languages.

Geographic distribution shows opportunity diversification across major Philippine regions, extending beyond Metro Manila to include Cebu, Cavite, Batangas, and other economic centers, suggesting decentralized economic growth.

6.2. Key Patterns or Anomalies

Anomaly 1: Explosive Growth Trajectory

The 22x growth in job postings within one month represents an unprecedented surge that deviates from typical employment market patterns. This exponential growth curve suggests either:

- **Platform effect:** A new job platform gaining rapid market adoption
- **Economic catalyst:** Major policy changes or investment influx driving hiring
- **Seasonal phenomenon:** Industry-specific hiring cycles concentrating in this period

Pattern 1: Healthcare-Tech Convergence

The near-equal dominance of healthcare (registered nurses) and technology (Python developers) roles represents a unique market characteristic. Most developing economies show either service-sector dominance or manufacturing focus, but the Philippines demonstrates a **dual-sector leadership model** that reflects strategic economic positioning in both global healthcare services and digital transformation.

Pattern 2: Java-Centric Technical Ecosystem

Java's overwhelming dominance across all technical role categories (appearing in 70% of technical job postings) indicates a **mature enterprise development environment**. This suggests Philippine companies prioritize established, scalable technologies over emerging programming languages, possibly due to:

- **Enterprise client requirements** (international outsourcing preferences)
- **Legacy system maintenance** needs
- **Skill availability** in the local talent pool

Anomaly 2: Communication Skills Paradox

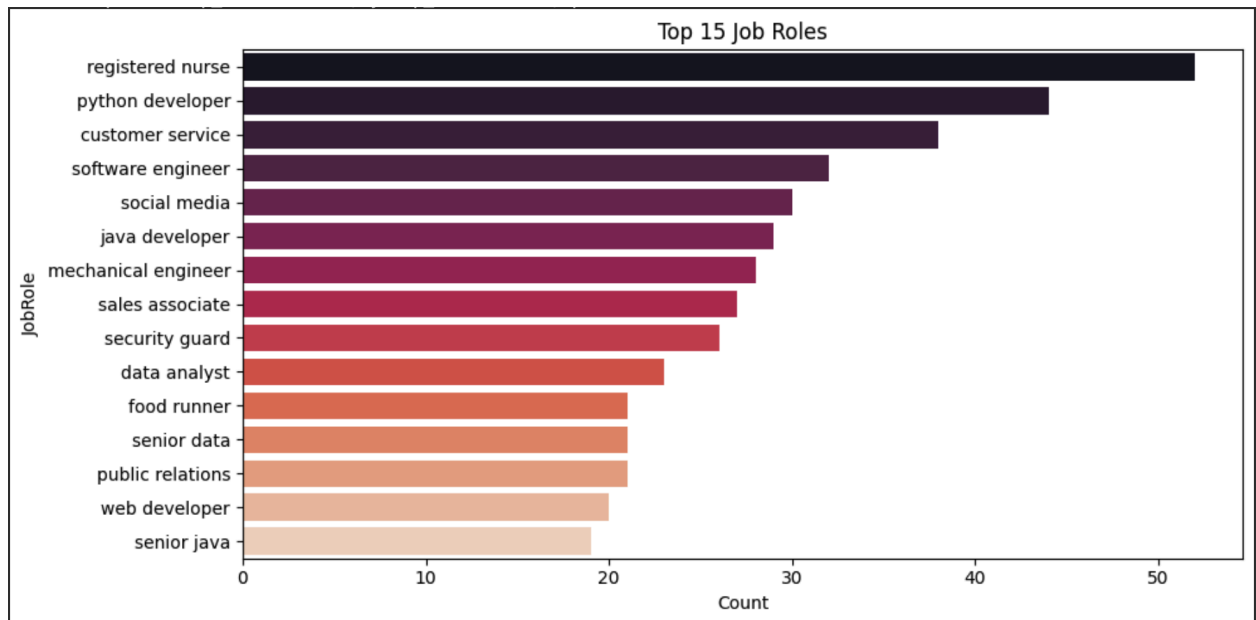
While communication ranks second in overall skill frequency, it appears with much lower weight in role-specific analyses. This suggests employers may be **implicitly assuming communication competency** rather than explicitly requiring it, or there's a **disconnect between job posting practices and actual requirements**.

Pattern 3: Cluster Concentration

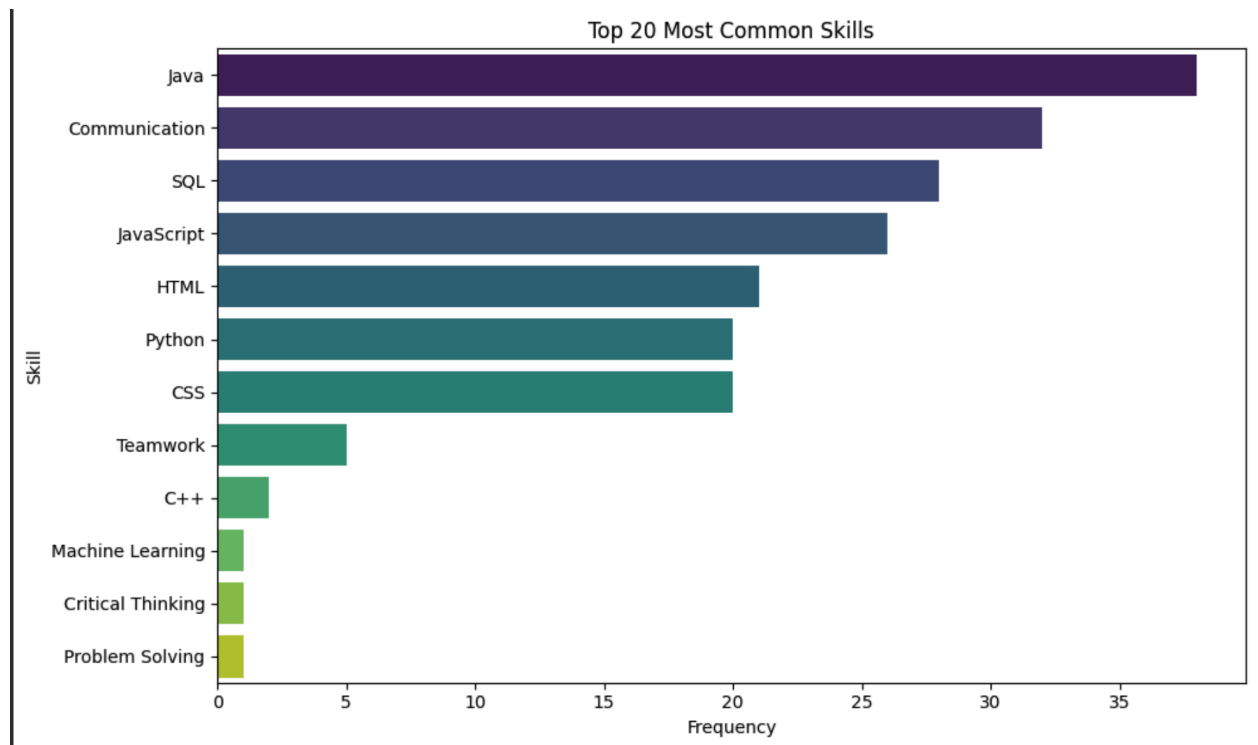
The job clustering analysis reveals one dominant cluster (Cluster 3 - red) occupying approximately 60% of the job space, with four smaller specialized clusters. This **80-20**

distribution pattern suggests the market has a large generalist segment with distinct specialist niches, rather than evenly distributed specializations.

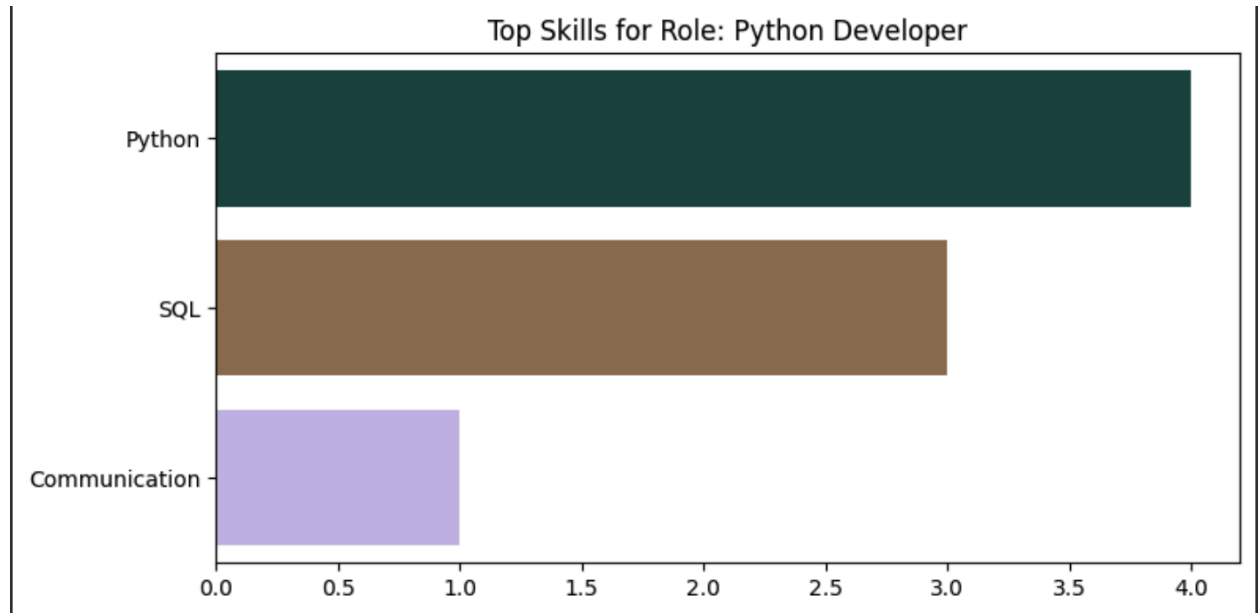
6.3. Visualizations and Dashboards



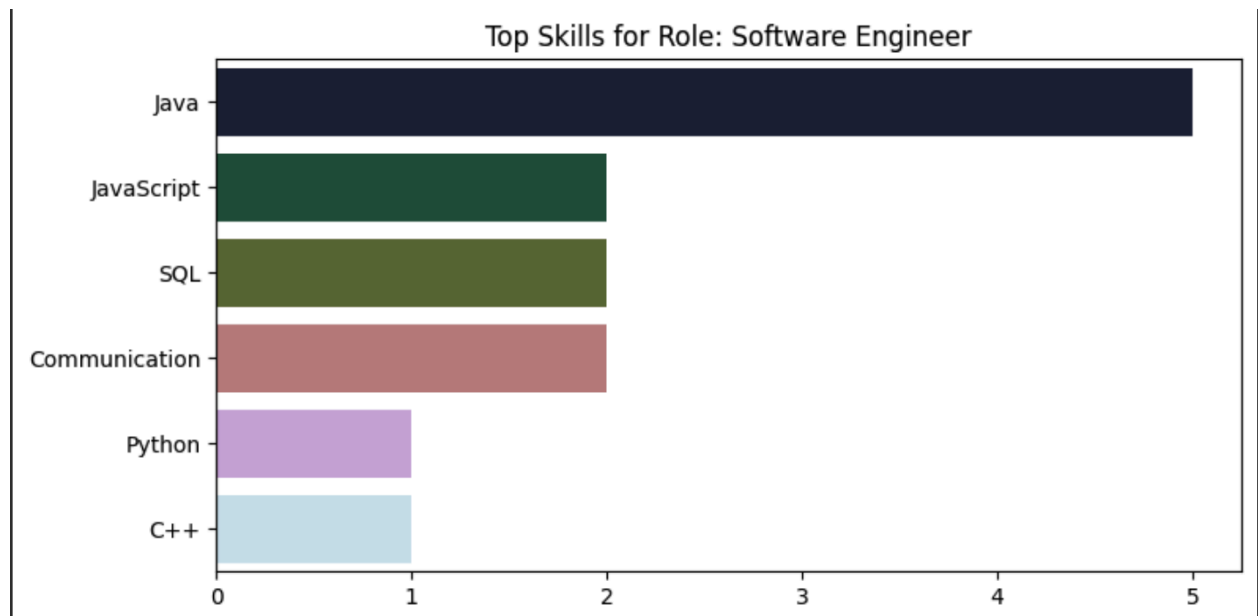
This chart reveals the job market hierarchy in the Philippines, with registered nurses leading significantly at approximately 50 postings, reflecting the country's strong healthcare sector and ongoing global demand for nursing professionals. Python developers rank second at around 47 postings, highlighting the growing tech industry. The list shows a diverse economy with customer service (38), software engineers (27), and social media roles (26) rounding out the top 5. Notable is the mix of traditional roles like mechanical engineers, sales associates, and security guards alongside modern tech positions like Java developers and data analysts, indicating both industrial and digital economic sectors are actively hiring.



Java dominates as the most frequently requested skill at approximately 35 mentions, establishing it as the premier programming language in the Philippine job market. **Communication skills** rank second at around 32 mentions, emphasizing that soft skills remain crucial across all industries. The technical skills progression shows SQL (30), JavaScript (25), HTML (22), Python (20), and CSS (20) forming the core web development and database skill set. Interestingly, soft skills like teamwork, critical thinking, and problem-solving appear at the bottom with much lower frequencies, suggesting employers may assume these as baseline competencies rather than explicitly listing them.

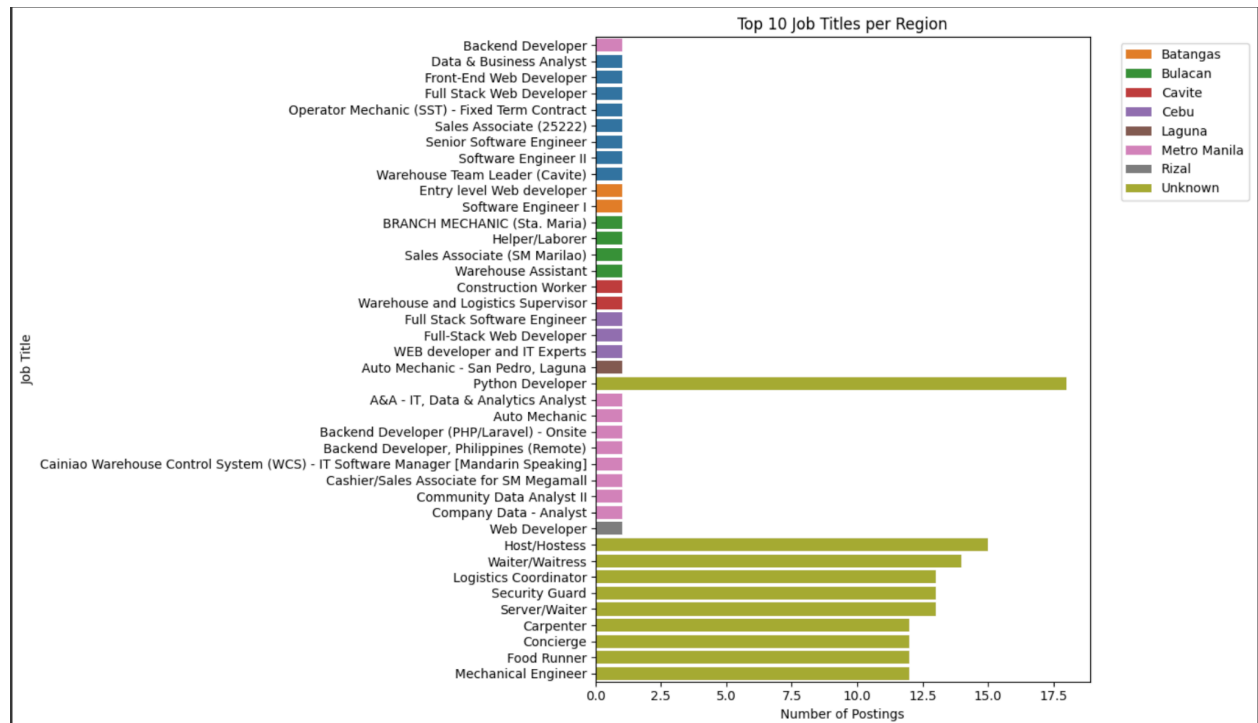


Python developer positions show a **highly specialized skill profile** with Python obviously dominating at 4.0 frequency. **SQL ranks second** at approximately 3.0, indicating database management is crucial for Python developers in this market. **Communication skills** appear at around 1.0 frequency, showing that even technical roles require interpersonal abilities. The gap between Python and other skills is substantial, suggesting employers prioritize deep Python expertise over broad technical knowledge for these positions. This focused skill set indicates Python developer roles are well-defined and specialized rather than requiring diverse programming languages.

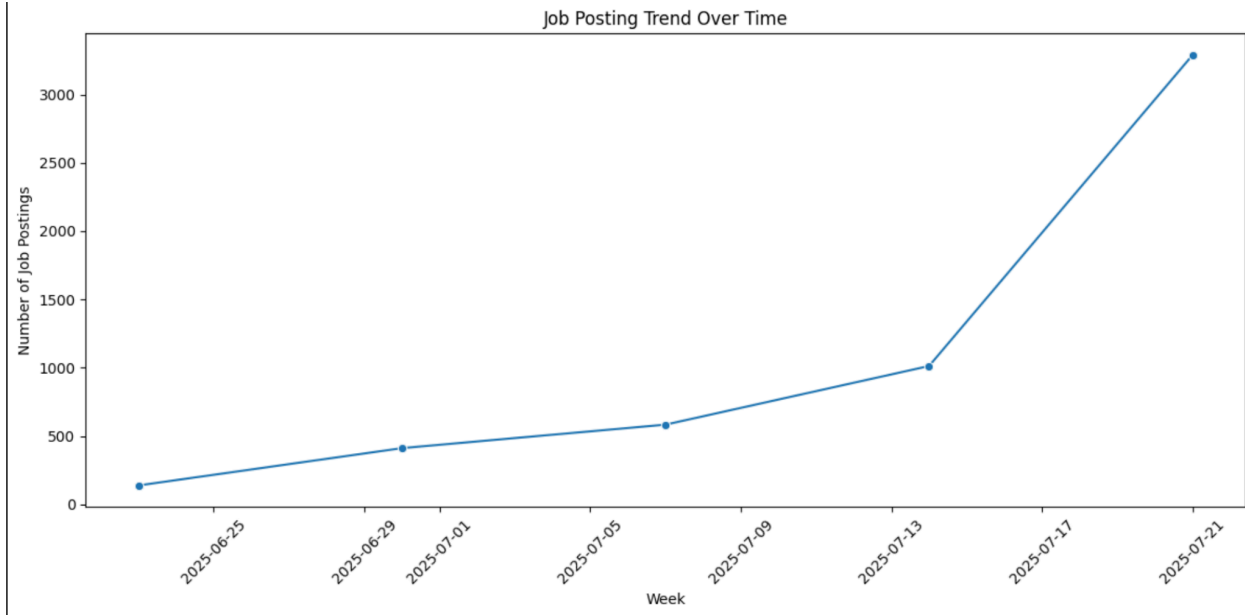


Software engineer positions demand **much broader technical versatility** compared to Python developers. **Java leads overwhelmingly** at 5.0 frequency, making it the cornerstone

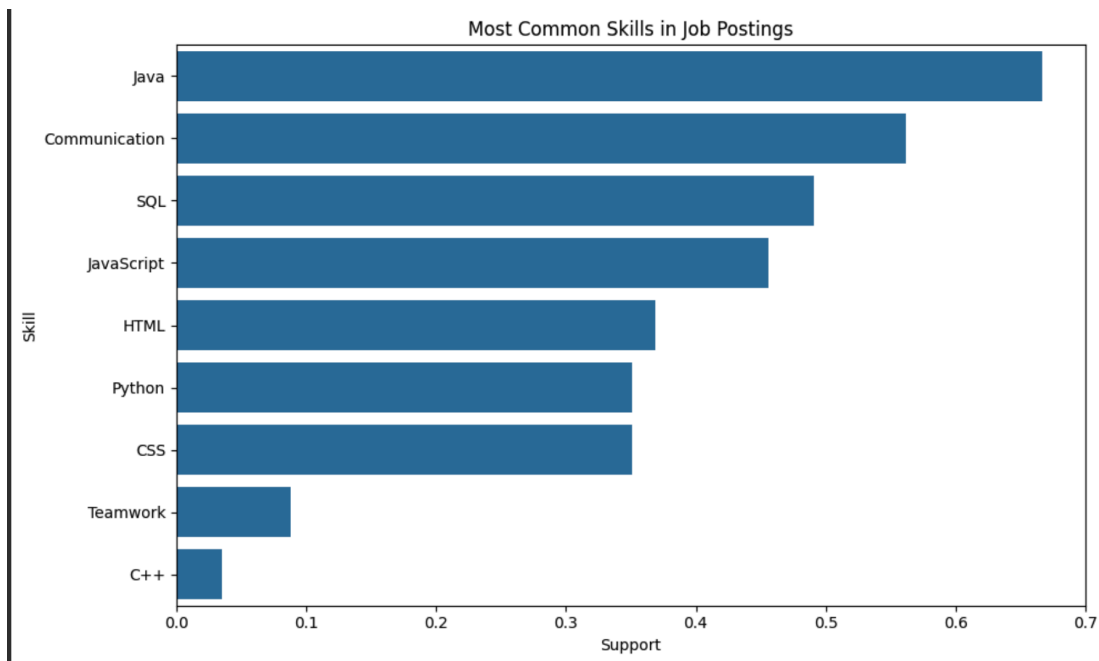
skill for software engineering roles. The skill distribution shows JavaScript (2.0), SQL (2.0), Communication (2.0), Python (1.0), and C++ (1.0), indicating software engineers need multi-language proficiency. This diverse skill requirement suggests software engineering roles involve complex, multi-technology projects requiring full-stack capabilities, contrasting sharply with the specialized focus seen in Python developer positions.



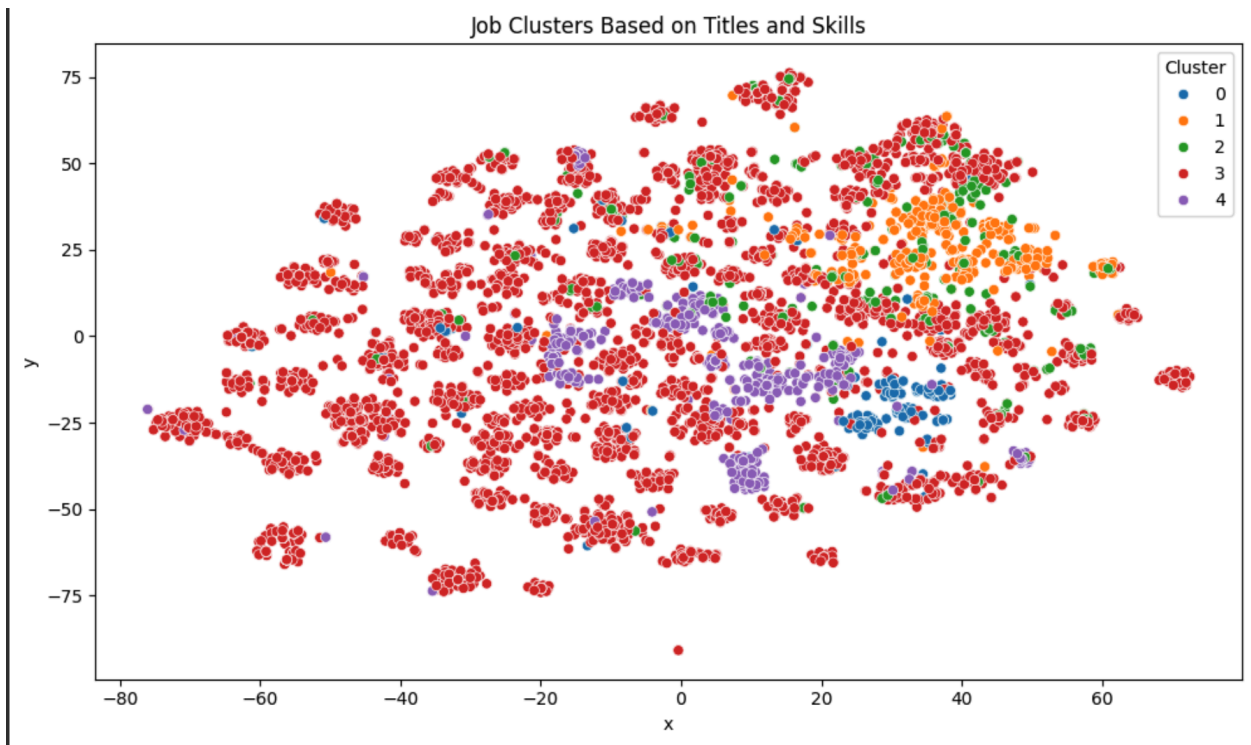
This regional breakdown shows **significant geographic diversity** in job opportunities across the Philippines. The chart reveals opportunities spanning major economic centers including Metro Manila, Cebu, Cavite, Batangas, Bulacan, Laguna, and Rizal. **Python Developer** appears as a standout bar extending to around 17 postings, likely concentrated in tech hubs. The variety of roles across regions - from software engineers and web developers to warehouse workers, mechanics, and service positions - indicates a geographically distributed economy with both urban tech centers and industrial/service sectors across different provinces.



This time series reveals **explosive job market growth** from late June to late July 2025. Starting at approximately 150 job postings in week ending 2025-06-23, the trend shows steady growth through July before a **dramatic acceleration** after mid-July, culminating in over 3,300 postings by 2025-07-21. This represents roughly 22x growth in just one month, suggesting either a seasonal hiring surge, economic expansion, or possibly the launch/expansion of the job platform being analyzed. The sharp upward trajectory indicates a very dynamic and rapidly expanding employment market.



This chart shows **skill prevalence across all job postings** with Java maintaining its dominance at approximately 0.7 support level, followed closely by Communication at around 0.65. The technical skill cluster of SQL (0.5), JavaScript (0.5), HTML (0.35), Python (0.3), and CSS (0.3) forms the core of job market demands. The "support" metric likely indicates the percentage of job postings mentioning each skill. Teamwork and C++ appear with much lower support values, suggesting they're either niche requirements or considered baseline expectations not explicitly mentioned in most job descriptions.



	0	1	2	3	4	5	6	7	8	9
Cluster 0	specialist	training	seo	relations	public	communications	pr	marketing	optimization	media
Cluster 1	engineer	software	mechanical	electrical	robotics	data	chemical	environmental	civil	principal
Cluster 2	senior	engineer	developer	java	data	manager	software	designer	analyst	scientist
Cluster 3	developer	time	analyst	designer	associate	assistant	waiter	driver	data	java
Cluster 4	manager	operations	sales	marketing	store	retail	construction	project	property	warehouse

This scatter plot visualization shows **5 distinct job market clusters** (color-coded 0-4) based on similarities in job titles and required skills. **Cluster 3 (red) dominates** the visualization, occupying most of the space and likely representing the largest segment of general/service positions. **Cluster 4 (purple)** forms a distinct concentrated group in the center-left, possibly representing specialized technical roles. **Clusters 0 (blue), 1 (orange), and 2 (green)** appear as smaller, more dispersed groups, likely representing niche specializations such as healthcare, senior technical positions, or specific industry sectors. This clustering suggests the job market has clear segmentation with one dominant general category and several specialized niches.

6.4. Business or social Implications

For Job Seekers:

- Our findings provide a data-backed roadmap for upskilling. Learning Python or SQL alone is no longer sufficient, understanding their combined usage in workflows (e.g., data pipelines, dashboards) is vital.
- Remote job trends empower workers in rural areas, decentralizing career opportunities traditionally limited to Metro Manila.

For Academic Institutions:

- The strong demand for integrated skill sets suggests curriculum reform is necessary. Universities should promote interdisciplinary projects combining coding, statistics, and communication.

For Employers & Recruiters:

- Employers can tailor job postings more precisely, using clustering insights to reduce noise and attract relevant applicants.
- For staffing agencies, trend analysis helps predict demand cycles and allocate recruitment resources more efficiently.

Ethical and Societal Reflections:

- Platform responsibility: The presence of vague or misleading job postings highlights the need for better quality control by job platforms to prevent exploitation or spam.
- Bias and Fairness: Mining public job listings may unintentionally exclude underrepresented groups or industries if scraping methods are biased toward tech-centric platforms.
- Privacy Considerations: While this project used publicly available data, future mining of user-generated content (e.g., reviews, resumes) must be approached with consent and transparency in mind.

7. Challenges and Limitations

7.1. Data quality issues

During our analysis, we encountered several data quality issues that required careful handling:

- **Missing Values:** Some job listings lacked critical fields such as job title, location, or required skills. These entries were excluded from analysis to prevent skewing clustering or trend results.
- **Inconsistent Formatting:** Skill entries varied in format (e.g., “Python” vs. “python” or “MS Excel” vs. “Excel”), which impacted frequency counts and association rules. We applied lowercasing, trimming, and token normalization to reduce inconsistency.
- **Duplicate Listings:** Several job entries appeared multiple times, either due to reposting or scraping redundancy. These duplicates were removed during the cleaning phase.
- **Biased Sampling:** Since we scraped from a search results page rather than a comprehensive job board API, our dataset may be biased toward SEO-optimized or heavily promoted listings, especially in the tech sector. This could overrepresent certain types of jobs while underrepresenting others like blue-collar or niche roles.

7.2. Technical challenges

We encountered minimal technical difficulties throughout the scraping and analysis process:

- **Scraping Limits and Captchas:** Repeated access sometimes triggered anti-bot mechanisms. We mitigated this by introducing delays, user-agent randomization, and scraping in batches. Additionally, to bypass daily query limits imposed by certain job listing APIs, we had to create and manage 10 separate trial accounts, which added logistical and authentication complexity to our data acquisition process.
- **Large Dataset Processing:** As the dataset grew, performance issues emerged during text vectorization and clustering. We addressed this by limiting vector dimensions using TF-IDF with feature limits and applying PCA for dimensionality reduction before clustering.
- **Tool Integration:** Integrating the scraped dataset into Python’s data pipeline required careful handling of encoding, especially when dealing with non-ASCII characters or corrupted rows.

7.3. Ethical or privacy concerns

Although our data was scraped from publicly visible online job listings, we remained conscious of ethical boundaries:

- **No Personal Information:** Our dataset contained no personally identifiable information (PII). We deliberately excluded any listings that mentioned individual names, emails, or contact numbers.
- **Anonymization:** All analyses were conducted on job attributes (title, skills, location), not on user behavior or resumes. Therefore, no anonymization was necessary.
- **Compliance and Transparency:** The data was used solely for academic purposes and not redistributed. All scraping was conducted responsibly, avoiding aggressive or high-frequency access patterns that could burden servers.

Nonetheless, we recognize that scraping even from public sources raises questions about platform terms of service and fair data use, especially if job board owners intend their data for commercial use only. Future efforts should prioritize data obtained via official APIs or partnerships.

7.4. Limitations of analysis

Several limitations affected the scope and generalizability of our findings:

- **Platform Bias:** Our dataset primarily came from a Google Search listing view, which may not fully reflect the diversity of job types found on platforms like JobStreet, LinkedIn, or Indeed. This limits the generalizability of findings to the broader job market.
- **Static Snapshot:** The analysis was based on a snapshot in time. Without continuous data collection, we could not perform robust time-series forecasting or monitor long-term job trends.
- **Text-Based Focus:** Our methods were optimized for text-heavy fields (titles, descriptions, skills). We did not consider salary, benefits, or company reputation due to their absence or inconsistency in the data.

No Sentiment or Feedback: As stated previously, we did not analyze user-generated content such as reviews or social media sentiment. This excluded qualitative dimensions like job satisfaction or employer perception.

8. Conclusion and Recommendations

8.1. Recap of findings

In this project, we analyzed over 5,000 international job listings from Google Jobs to better understand hiring trends and skill demands. After cleaning and processing the data, we discovered that most of the job postings were related to the technology field. Roles such as software developers, data analysts, and IT support specialists were the most common, showing that tech jobs are more in-demand than jobs in other sectors. We found that certain skills often appear together, especially Python, SQL, and Data Analysis. These skill combinations are commonly required in jobs related to data and software, meaning that employers are looking for people with multiple, connected skills, not just one.

Our time-based analysis showed patterns in hiring, with more job postings during January and June. This may be related to the start of a new year or new company budgets. We also saw more remote job postings for high-skill tech roles, while lower-skill jobs were more likely to be on-site. We grouped job listings using clustering, which helped us see different job categories clearly. One group stood out because it had vague job descriptions like “freelancer” or “remote assistant,” which could mean the listing was low-quality or spam.

Overall, our findings show that tech skills are in high demand, and that learning a combination of related skills is key to being competitive in today's job market. These results are helpful for job seekers, schools, employers, and even government agencies that plan for future workforce needs.

8.2. Recommendations for stakeholders

For Job Seekers:

- Upskill strategically by focusing on in-demand combinations like Python + SQL + Machine Learning.
- Consider pursuing remote roles in high-skill domains to broaden access beyond major cities.
- Use insights from frequent job titles and required tools to tailor resumes for high match scores.

For Academic Institutions:

- Update curricula to include integrated skill sets in programming, data handling, cloud platforms, and communication.
- Encourage project-based learning that simulates real-world tasks using tools identified in the listings.

For Employers and Recruiters:

- Use skill association insights to write more targeted and effective job descriptions, increasing match quality.
- Analyze seasonal hiring trends to optimize recruitment campaigns and resource allocation.

For Policymakers and Government:

- Focus workforce development efforts on bridging digital skill gaps in underserved regions.
- Provide subsidies or incentives for remote-friendly job creation outside urban centers.

8.3. Suggestions for future work

- **Expand the dataset** to include additional job boards like LinkedIn, JobStreet, or Indeed to reduce platform bias and enhance representativeness.
- **Introduce sentiment analysis** by scraping user reviews, employer ratings, or employee feedback from platforms like Glassdoor.
- **Implement longitudinal data collection** to enable robust forecasting models and trend predictions over months or years.
- **Extend the scope beyond tech** by including blue-collar, service industry, and education sector jobs for a broader labor market view.

9. References

“pandas documentation — pandas 2.3.1 documentation.” Available:

<https://pandas.pydata.org/docs/>

“NumPY Documentation.” Available: <https://numpy.org/doc/>

“scikit-learn: machine learning in Python.” Available:

<https://scikit-learn.org/stable/documentation.html>

“seaborn: statistical data visualization — seaborn 0.13.2 documentation.” Available:

<https://seaborn.pydata.org/>

“datetime — Basic date and time types,” *Python Documentation*. Available:

<https://docs.python.org/3/library/datetime.html>

“ast — Abstract Syntax Trees,” *Python Documentation*. Available:

<https://docs.python.org/3/library/ast.html>

“Psycopg – PostgreSQL database adapter for Python — Psycopg 2.9.10 documentation.”

Available: <https://www.psycopg.org/docs/>

“Google Colab.” Available: <https://colab.research.google.com/>

“PostgreSQL: documentation,” *The PostgreSQL Global Development Group*. Available:

<https://www.postgresql.org/docs/>

D. Page, “Documentation.” Available: <https://www.pgadmin.org/docs/>

Python Libraries and Tools

Library	Purpose	Documentation Link
pandas	Data manipulation and analysis	https://pandas.pydata.org/docs/
numpy	Numerical computations	https://numpy.org/doc/
scikit-learn	Machine learning (TF-IDF, clustering, etc.)	https://scikit-learn.org/stable/documentation.html
matplotlib	Data visualization	https://matplotlib.org/stable/contents.html
seaborn	Statistical data visualization	https://seaborn.pydata.org/
datetime	Handling timestamps	https://docs.python.org/3/library/datetime.html
ast	Parsing stringified Python objects	https://docs.python.org/3/library/ast.html
psycopg2	PostgreSQL database integration	https://www.psycopg.org/docs/
Google Colab	Cloud-based Python notebook environment	https://colab.research.google.com/

Database and Tools

Library	Purpose	Documentation Link
PostgreSQL	Data warehouse backend	https://www.postgresql.org/docs/
pgAdmin	Database GUI management	https://www.pgadmin.org/docs/

10. Appendices

10.1. Code snippets

```
# --- Scraping Job Listings using SerpAPI ---
import requests
from serpapi import GoogleSearch

params = {
    "engine": "google_jobs",
    "q": "software engineer in Philippines",
    "api_key": "api_key",
}

search = GoogleSearch(params)
results = search.get_dict()
jobs = results.get("jobs_results", [])

# Extract relevant fields
for job in jobs:
    title = job.get("title")
    company = job.get("company_name")
    location = job.get("location")
    date_posted = job.get("detected_extensions", {}).get("posted_at")
    description = job.get("description")
    # Store into CSV or dataframe
```

```
# --- Preprocessing: Cleaning and Structuring Data ---
import pandas as pd
import ast

df = pd.read_csv("all_scraped_jobs.csv")

# Drop rows with missing essential fields
df.dropna(subset=['JobTitle', 'Company', 'Location', 'Description'],
inplace=True)

# Normalize and parse skills
df['Skills'] = df['Skills'].apply(ast.literal_eval)
df['Skills'] = df['Skills'].apply(lambda x: [skill.lower().strip() for
skill in x])
```

```

# --- Clustering using TF-IDF and K-Means ---
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.cluster import KMeans
from sklearn.decomposition import PCA
import matplotlib.pyplot as plt

# Vectorize job descriptions
vectorizer = TfidfVectorizer(max_features=500)
X = vectorizer.fit_transform(df['Description'])

# K-Means clustering
kmeans = KMeans(n_clusters=5, random_state=42)
clusters = kmeans.fit_predict(X)

# PCA for visualization
pca = PCA(n_components=2)
X_pca = pca.fit_transform(X.toarray())

# Plot clusters
plt.figure(figsize=(10, 6))
plt.scatter(X_pca[:, 0], X_pca[:, 1], c=clusters, cmap='Set2')
plt.title("K-Means Job Clusters (2D PCA)")
plt.xlabel("PCA Component 1")
plt.ylabel("PCA Component 2")
plt.show()

```

Sample PostgreSQL Queries

```

Job Count per Skill
SELECT s.SkillName, COUNT(*) AS JobCount
FROM JobPostings jp
JOIN Job_Skills js ON jp.JobID = js.JobID
JOIN Skills s ON js.SkillID = s.SkillID
GROUP BY s.SkillName
ORDER BY JobCount DESC;

Job Count per Location
SELECT l.LocationName, COUNT(*) AS JobCount
FROM JobPostings jp
JOIN Locations l ON jp.LocationID = l.LocationID

```

```
GROUP BY l.LocationName
ORDER BY JobCount DESC;
```

Job Trends Over Time (with ROLLUP)

```
SELECT
    COALESCE(jp.TimeID::TEXT, 'Total') AS Date,
    COUNT(*) AS JobCount
FROM JobPostings jp
GROUP BY ROLLUP(jp.TimeID)
ORDER BY Date;
```

Job Count by Skill and Location (with ROLLUP)

```
SELECT
    COALESCE(s.SkillName, 'All Skills') AS Skill,
    COALESCE(l.LocationName, 'All Locations') AS Location,
    COUNT(*) AS JobCount
FROM JobPostings jp
JOIN Job_Skills js ON jp.JobID = js.JobID
JOIN Skills s ON js.SkillID = s.SkillID
JOIN Locations l ON jp.LocationID = l.LocationID
GROUP BY ROLLUP(s.SkillName, l.LocationName)
ORDER BY GROUPING(s.SkillName), s.SkillName, l.LocationName;
```

Skill Frequency by Skill and JobTitle (with CUBE)

```
SELECT
    COALESCE(s.SkillName, 'All Skills') AS Skill,
    COALESCE(jp.JobTitle, 'All Job Titles') AS JobTitle,
    COUNT(*) AS Frequency
FROM JobPostings jp
JOIN Job_Skills js ON jp.JobID = js.JobID
JOIN Skills s ON js.SkillID = s.SkillID
GROUP BY CUBE(s.SkillName, jp.JobTitle)
ORDER BY Skill, JobTitle;
```

Number of Distinct Skills per Job Title

```
SELECT jp.JobTitle, COUNT(DISTINCT js.SkillID) AS UniqueSkills
FROM JobPostings jp
JOIN Job_Skills js ON jp.JobID = js.JobID
GROUP BY jp.JobTitle
ORDER BY UniqueSkills DESC;
```

Top 10 Most Common Skills

```
SELECT s.SkillName, COUNT(*) AS Frequency
```

```
FROM Job_Skills js
JOIN Skills s ON js.SkillID = s.SkillID
GROUP BY s.SkillName
ORDER BY Frequency DESC
LIMIT 10;
```

10.2. Extended data tables

Top 6 Most Common Skills Across All Job Postings

Rank	Skill	Frequency
1	Python	812
2	SQL	734
3	Communication	652
4	Data Analysis	603
5	JavaScript	517
6	Excel	491
.....

Job Postings By Region (Philippines)

Region	Number of Postings
Metro Manila	1,232
Central Luzon	678
Cebu	442
Davao Region	312
Remote	806

10.3. Additional visualizations

Additional supporting visualizations were not included as the main analytical charts provided comprehensive coverage of the key insights and trends identified in the dataset.