Data Science Job Market Data Mining: Identifying the Most In-Demand Skills for Data Scientists

Course: DSCI 510 — Principles of Programming for Data Science
Semester: Fall 2025
Name: Jui-Ching Yu
Email: juiching@usc.edu
GitHub: https://github.com/Meredith0613
USC ID: 5507402044

1. Introduction

The rapid expansion of data-driven industries has heightened demand for data scientists, machine learning engineers, analysts, and data engineers. Yet students and early-career professionals often face unclear or inconsistent job descriptions, making it difficult to identify which technical skills are truly required.

This project aims to provide an evidence-based overview of the current data-science job market by collecting real postings, extracting common technical requirements, and identifying patterns in employer expectations. Using SerpAPI's Google Jobs engine, job postings were collected across four roles—**Data Scientist, Machine Learning Engineer, Data Analyst, and Data Engineer**—and three regions: **United States, Canada, and the United Kingdom**. After cleaning and deduplication, the final dataset included **564 unique job postings**. These postings form the basis for the skill analysis and TF-IDF keyword extraction presented in this report.

2. Data Collection

Data collection was performed using SerpAPI's Google Jobs API, which enables structured, consistent, and ethically sourced job listings without triggering the anti-bot protections commonly found on platforms such as LinkedIn or Indeed. For each of the four targeted roles, queries were issued across three regional engines (google_jobs_us, google_jobs_ca, and google_jobs_uk), and results were retrieved using the API's pagination mechanism (next_page_token), with up to five pages requested per role–region combination. Each page typically returns approximately ten postings. The raw JSON responses were then normalized into CSV format and stored in data/raw/jobs_raw.csv. A controlled test run of the pipeline demonstrated correct behavior in both failure and success conditions: the script initially terminated due to a missing API key, confirming proper credential validation, and once configured, successfully collected several hundred postings across all roles and regions. The final cleaned dataset contained 564 unique postings, shaped by real-world constraints such as pagination depth, regional job availability, and the timing of data collection.

3. Data Cleaning

Data cleaning was executed using clean_data.py and followed a structured, reproducible workflow. The process began with core field validation, ensuring that each row contained the essential fields—title, company, location, and description—and removing rows missing title or company. Deduplication was then performed using a combination of job_id, title, company, and location, resulting in the removal of nine duplicate postings. To ensure consistency across sources, all text fields were standardized by converting them to lowercase, stripping excess whitespace, and normalizing formatting for later NLP tasks. Where available, posted_at timestamps were parsed into timezone-aware UTC format. The script also extracted skills by applying a curated dictionary of more than eighty technical keywords (including Python, SQL, Pandas, TensorFlow, AWS, and Airflow) to generate a skills_extracted column. To ensure compatibility across downstream scripts, schema standardization was performed: query_role was mapped to role_query, job_description was consolidated under description, skills fields were unified under skills, location-related fields were harmonized into country and location_normalized, and salary-related fields were consolidated when possible. The fully cleaned dataset was then saved to data/processed/jobs_clean.csv, containing 578 rows before filtering and 564 rows after final standardization.

---

4. Data Analysis

Analysis was conducted using run_analysis.py, which processed the finalized dataset and generated TF-IDF keyword rankings alongside summary statistics. The cleaned dataset included standardized role labels under role_query, enabling per-role comparisons across the four job families: Data Engineer (147 postings), Machine Learning Engineer (146 postings), Data Analyst (143 postings), and Data Scientist (142 postings). Using aggregated job descriptions for each role, the script computed TF-IDF scores across roughly 100 unique terms, producing the output file data/analysis/tfidf_by_role.csv. In addition to TF-IDF analysis, a global frequency distribution of extracted skills and several skill co-occurrence counts were computed, some of which supported downstream visualization tasks. Salary analysis, however, was limited by the structure of the underlying data: Google Jobs rarely provides standardized salary fields, resulting in missing salary_min and salary_max values and inconsistent salary_raw strings. Consequently, salary-based modeling and visualizations were omitted. Despite these limitations, the analysis provided meaningful insights into the technical competencies most frequently requested in data-science-related roles.

---

5. Visualizations

Visualizations were generated using visualize_results.py, which produced several figures summarizing key trends in the dataset. The top skills bar chart highlighted the most frequently requested competencies across all postings, with Python, SQL, AWS, machine learning, Pandas, and TensorFlow appearing most prominently. A skills word cloud provided a qualitative representation of the same information, scaling text size according to frequency. The location distribution chart illustrated how job postings were distributed across countries or normalized geographic fields. Some planned visualizations, however, were omitted due to data limitations: the skill co-occurrence network could not be meaningfully generated because too few skill pairs occurred frequently enough to form robust connections, and the salary versus skill count scatter plot was excluded because the dataset lacked usable structured salary information. These omissions reflect real-world sparsity in Google Jobs metadata and highlight the potential value of supplementing this dataset with alternative sources in future work.

6. Findings & Observations

The analysis revealed several clear insights into the current data-science job market. First, foundational technical skills remain dominant across all four job families, with employers consistently expecting proficiency in Python, SQL, machine-learning techniques, statistics, cloud platforms such as AWS, Azure, and GCP, and widely used data libraries including Pandas, NumPy, TensorFlow, and PyTorch. These expectations suggest a stable core skill set that spans both analytics-oriented and engineering-oriented roles. Second, even without a complete network visualization, the co-occurrence patterns indicated natural tool clusters that reflect real industry workflows: Python, Pandas, and NumPy frequently appeared together in data-preprocessing contexts; TensorFlow, PyTorch, and scikit-learn clustered around machine-learning tasks; and AWS, Spark, Databricks, and Airflow commonly co-occurred in data-engineering pipelines. These clusters illustrate how technical tools are used jointly in practice rather than in isolation. Third, meaningful salary analysis was not feasible, as most Google Jobs postings lacked structured salary information, limiting quantitative comparisons across roles or skill sets. Overall, the findings offer a realistic snapshot of current demand within the DS/ML job market and highlight foundational competencies that students should prioritize when preparing for technical roles. Because this project relies on collecting live job postings from the SerpAPI Google Jobs API, however, the exact results may vary when the code is re-executed. Job listings change frequently, API responses fluctuate over time, and the number of available postings depends on current availability, quota limits, and pagination behavior. Therefore, the instructor's results—such as dataset size, TF-IDF keywords, or skill-frequency counts—may differ from those reported here. The insights presented in this report reflect the dataset generated during my documented test run.

7. Changes From Original Proposal

Several adjustments were necessary during implementation to accommodate real-world data and technical constraints. The original proposal intended to scrape LinkedIn, Glassdoor, and Handshake, but due to strong anti-bot protections on those platforms, the project shifted to using SerpAPI's Google Jobs engine, with the possibility of incorporating Kaggle datasets in future work. Additionally, although the initial goal was to collect approximately 1,000 postings, the final dataset contained 564 unique entries, shaped by factors such as API quota limits, pagination depth, and the availability of postings at query time. The scope of NLP analysis was also revised: clustering and more advanced role-specific TF-IDF were planned initially, but limitations stemming from inconsistent salary fields, the earlier absence of standardized role labels, and the overall dataset size restricted the analysis to corpus-level and per-role TF-IDF computations.

Salary parsing was intentionally simplified due to the heterogeneous and unstructured nature of salary information in Google Jobs postings. Despite these modifications, the project remained robust and produced meaningful, interpretable insights aligned with the project's learning objectives.

8. Future Work

Several directions could significantly enhance this project in future iterations. First, expanding data sources to include LinkedIn or Indeed datasets—potentially accessed through Kaggle collections or partner APIs—would greatly increase both the size and diversity of the dataset. Second, more advanced NLP techniques could replace simple keyword matching, such as spaCy-based preprocessing, BERT or Sentence Transformer embeddings, named entity recognition, or topic modeling frameworks like BERTopic or LDA, allowing for deeper semantic understanding of job descriptions. Third, predictive modeling could be incorporated to estimate salary ranges, seniority levels, or latent skill clusters using learned embeddings from the text. Finally, with a larger and more consistently labeled dataset, clustering algorithms such as K-means or HDBSCAN and refined role-specific TF-IDF analysis could provide more nuanced insights into role segmentation and variation within the data-science job market.

9. Conclusion

This project presents a complete and modular pipeline for collecting, cleaning, analyzing, and visualizing job postings sourced from SerpAPI's Google Jobs engine. Using a dataset of 564 real postings, the analysis highlights the most in-demand skills across major data-science job families and uncovers the tool clusters that underpin contemporary data workflows. While certain analyses were limited by the availability and structure of the underlying data, the project nonetheless demonstrates strong software engineering practices, a reproducible data-processing workflow, and insights that are highly relevant for students preparing for careers in data science and machine learning. Overall, this work establishes a solid foundation for future extensions, including broader data integration and the application of more advanced NLP techniques.