Winter
2023-24

Data Collection Lab

Final Project Report

Tal Peer **Yoav Sgan Cohen** **Itay Bachar**

Technion - Israel Institute of Technology

Table of Contents

Introduction

In this project, we aimed to develop an AI-powered Profile Analyzer tool that empowers LinkedIn members to make informed decisions about their career paths and skill development strategies.

Our research question: ***How can AI be leveraged to provide personalized recommendations for skill development and career advancement on LinkedIn?***

Our main interest was to develop tools for understanding and developing a roadmap of people on LinkedIn, based on the LinkedIn-people dataset and additional datasets, which play a crucial part in jobs on LinkedIn but were not part of the original dataset.
Hence, it obligated us to use Data integration techniques such as entity matching methods for integrating multiple data sources into one, where the new table index is the position attribute after processing using various Natural Language Processing (from here will be referred as NLP) methods, as will be discussed later. Also, basic machine learning methods such as feature selection, cross-validation (for choosing the "best" metric for our entities), and dimensional reduction were employed.

In our initial proposal, we suggested a form of RAG - creating a roadmap that would eventually be stored on a data store platform named Vector Store. This platform would be utilized by an LMM agent to retrieve information and generate a textual summary based on a constant prompt and the data from the retrieved table. Additionally, a dashboard would be developed to enhance the user experience. All these aspects would utilize the Langchain package. Unfortunately, the last methodology didn't fit into the project's time window, so we decided to focus on the Entity Matching process, setting the foundation for various implementation ideas. The Demo app (appendix) represents a basic implementation of this concept. Given that our project primarily deals with Entity Matching, we merged the sections of Data Collection and Integration with Data Analysis.

Data Collection and Integration

Data Collection

1. LinkedIn Profiles Dataset (Profiles) - Provided as course material, this dataset is collected from LinkedIn user profiles, including professional experiences, education, skills, and more.
2. 1.3 million LinkedIn Jobs & Skills (Job Postings) - A dataset listed on Kaggle that contains over 1.3 million job listings scraped from LinkedIn job postings in the year 2024. In collaboration with the profile's dataset, we integrated this dataset to extract relevant skills, education, and experience associated with each position.
3. O*Net 27.2 Data (EM Titles) - The latest version of the O*Net database, providing descriptions of job-related attributes. This resource enabled us to obtain clean job titles for entity matching between the two datasets.

Job Posting Dataset – Quick overview

We opted to utilize job posting data from LinkedIn to gain insights into emerging job markets and the required skills. Leveraging LinkedIn's vast job posting database, we extracted information like job titles, descriptions, and necessary skills. Integrating this dataset with our existing resources enabled us to offer valuable recommendations to users aiming to acquire the skills needed for their desired positions. As technology transforms the known framework for positions, traditional metrics like position, previous experience and degrees may become less relevant. Skills provide another

perspective for understanding job requirements, which reflect the evolving nature of work. We base this assumption on the "Skills, not job titles, are the new metric for the labour market "
this dataset, used by LinkedIn themselves in many project, (e.g. "LinkedIn - Future of Skills" article), which highly emphasizes the importance of skills in the present job searching paradigm.

- ***Table Scheme for the relevant columns we extracted from each dataset for our project can be find in the appendix section.***

Preprocessing

To gather meaningful insights on the jobs themselves and to later join these insights with data gathered from the Job Posting dataset we applied basic processing to Profiles' columns so that it would follow the following criteria:

- non null or null equivalent position - this is important because we wanted to gather data only for users who are currently working in a position.
- non null or null equivalent experience duration - to match Job Posting's data about typical required experience for the position.
- non null or null equivalent education degree - to match Job Posting's data about required education certification for the position.

Choosing job title subset to focus on

To stay within the project timeframe while incorporating Gemini to help extract experience and education requirement data from "job_summary" in Job Postings, we had to find a subset of positions to focus on from the large pool of possible job titles (~54,000).

Furthermore, we needed a good way to choose titles that will have a high probability of showing up in both datasets.

Solution

- Sample from both tables.
- Use string matching algorithm.
- Find Intersection: the top 500 "match_positions" from each dataset are compared to find common positions that appear in both datasets.

This process results in approximately 250 job titles that are likely to be present in both datasets.

Data Analysis & Feature Engineering

Upon further inspection of the data, we identified key problems that needed to be addressed before we could finalize the features of each table:

Profiles dataset

- position - the position column showcases significant diversity, with 123,446 unique positions out of 142,966 entries. Furthermore, because the original column is string and free text in LinkedIn, each position can be written in different ways like the position ".Net Developer": ".NET Developer - Clean Code, Agile Delivery", ".Net Developer at ISDH" and more.
- experience_titles - includes a wide range of freeform text, some of which may not be relevant to the position. This can lead to noise in the data, making it difficult to accurately assess the user's qualifications for a particular position.

- experience_duration - as it is represented as a string of the form "1 year 6 months" leaving the strings in their current form poses a challenge for data analysis due to their lack of standardization, making numerical calculations difficult. Moreover, comparing the two dataset's values in this column revealed a significant disparity. After calculating the mean duration for each position, we found that for 225 out of 229 positions, the mean duration in the profile's dataset exceeded that of the skills dataset. We suspect this discrepancy is due to the profile's dataset including durations of non-relevant experience items with regards to the position, whereas the skills dataset does not.
- education_degree - the education_degree column contains a total of 60,902 unique education degree values out of 142,966 entries. yet again, very high variety with many values seen only once. In addition, the same degree can be written in certain ways, for example: "Bachelor of Arts - BA", "Bachelor's degree", "B.A", "BA", "Bachelor's" etc.

Job Postings dataset

- Position - mirroring the difficulties encountered in the profile's dataset, the position column faces a comparable challenge: a multitude of unique positions (12,861) compared to the total entries (21,482). This inconsistency in job titles complicates data analysis, possibly leading to inaccuracies in the insights derived from the dataset.

Feature Engineering

Profiles dataset

- Position - to address the uniqueness and the freeform text issues, we will use a predefined list of common positions using string matching (details provided in the next section).
- experience_title - to capture the entirety of a user's career experience, we integrated an additional dataset from O*Net, which provides a hierarchical structure of job titles. We then filtered out non-relevant experiences and eliminated any experiences that match the position itself. This process enabled us to retain only those experiences that fall within the same class of job titles as the position excluding the position itself. This left us with only relevant experience items.
- durations - converting the string values to floating-point numbers (e.g., 1.5 or 6.583) to enable numerical calculations and standardization for easier data analysis.
  While filtering out non relevant experience items, we produce two new columns: "total_years_in_position" and "total_years_relevant_experience".
  "total_years_relevant_experience" is the sum of the durations of the relevant experience for the current position, and "total_years_in_position" is the sum of all durations mapped by the string matching algorithm to the same job title.
- Education_degree - to tackle the variety of the values, we created a set list of common education degrees and mapped each value to this list, ensuring consistency and facilitating meaningful insights from the dataset.

Job Postings dataset

- Position - to address the uniqueness and the freeform text issues, we will use a predefined list of common positions using string matching (details provided in the next section).

Following extensive analysis, feature selection, and data transformation on both datasets, we unified the profiles and skills datasets using the position column and combined the rich information from both sources to provide a comprehensive overview of each position. This dataset serves as a valuable resource for gaining insights into job roles and is presented along with other data.

AI Methodologies

Entity Matching - Matcher

In the realm of Data Integration and NLP, comparing and matching strings is a common and crucial task. However, due to variations in spelling, word order, and minor differences of how people define their position title on LinkedIn, exact string matching may not always yield accurate results.
In our project, we explored different methods like Gemini. However, due to time constraints, we had to abandon it. We also attempted to clone HuggingFace models for embeddings but were constrained by resource limitations. Ultimately, we found that testing various fuzzy string-matching algorithms yielded the best results.

Fuzzy string-matching algorithms provide valuable tools for comparing and measuring the similarity between strings. These algorithms are particularly useful when dealing with variations in spelling, word order, or minor differences that are highly relevant for project data. This capability enables us to express rules that are challenging to clearly define, resulting in a pipeline with resilience to data imperfections.

Each step of our EM pipeline is based on a Python package named "TheFuzz," which offers several different algorithms (also called ratios or scorers) for calculating match accuracy. All the scorers use Levenshtein Distance to calculate differences between strings. We used cross-validation techniques on a sample dataset and chose the best scorer based on accuracy – the "Token Set Ratio".

```
Best matches for EXPORT SPECIALIST at DHL Express Nederland:
token_set_ratio
     - Export Specialist - score of 100 %
token_sort_ratio
     - Seed and Fertilizer Specialist - score of 64 %
partial_ratio
     - Export Specialist - score of 100 %
partial_token_sort_ratio
     - Lawn Specialist - score of 93 %
q_ratio
     - Health Support Specialist (HSS) - score of 58 %
default(Wratio)
     - Export Specialist - score of 90 %
```

The algorithm tokenizes both input strings, removes duplicate tokens, and calculates the similarity score based on the intersection and union of the token sets. It captures the essence of the strings' content rather than their specific order. Token Set Ratio is effective for scenarios where the word arrangement might vary but the overall content remains similar – a quality particularly relevant for LinkedIn position titles.

Generative AI & Data Extraction

To extract relevant education and relevant experience items mentioned in Job Posting's job summary column we incorporated the use of Google's generative AI model, Gemini, using a structured function call. Gemini was instructed to single out education requirements and experience requirements if present in the job summary text, in the same form as in the Profiles dataset as a structured Json-like object to allow us to process its results with the same logic we used for Profiles.

Evaluation and Results

Entity matching requires a way to determine whether two entities are alike enough to represent the same real-world entity. As mentioned in previous section, we used a fuzzy string matcher by specifying how the similarity between two entities is computed (the metric to use) and consequently setting the threshold according to the results. As with any other ML model, we tested several available matching scorers, and the best results were in favor of "partial_token_set_ratio" with threshold of 90%.

Due to its pairwise matching nature, class imbalance is a distinguishing property of EM, compared to regular classification tasks. The “privileged” positions are likely to be over-represented and minority groups under-represented in the chosen set. As a result, a lower (true and false) positive match rate is expected for minority groups and they end up receiving information, which is based on fewer samples than other, resulting in biased results.

We examined the distribution of match results in both tables, assuming that the percentage of value counts for each position in the entire dataset should be similar. Any deviation from this expectation may indicate differences between the original LinkedIn Profiles dataset and the integrated dataset. We performed this analysis using the Kolmogorov-Smirnov test available in the SciPy package. The Kolmogorov-Smirnov test is a statistical method used to compare two probability distributions, measuring the similarity between their cumulative distribution functions (CDFs).

A low p-value implies robust evidence against the null hypothesis, indicating significant divergence between the distributions. With a p-value below the conventional significance threshold of 0.05, we reject the null hypothesis, suggesting distinct distributions. This outcome could be attributed to the disparity between LinkedIn profile data, derived from individual profiles, and the integrated dataset sourced from LinkedIn job descriptions. Positions such as “Cook” and “Nurse” may be infrequent among LinkedIn users but prevalent in job postings.

```
from scipy.stats import ks_2samp

statistic, p_value = ks_2samp(position_pct_skills.values.flatten(),
                              position_pct_profiles.values.flatten())

print("Kolmogorov-Smirnov test statistic:", statistic)
print("P-value:", p_value)
Executed at 2024.04.10 20:27:48 in 17ms

Kolmogorov-Smirnov test statistic: 0.1414640662570131
P-value: 0.025587653259881535
```

Our project result is a table with positions as keys, where each field was calculated with some form of aggregation (sum or count), based on the table resulted from the EM process. (More in the appendix)

Limitations and Reflection

We encountered a series of challenges and limitations that offered valuable learning opportunities:

- Limited computational resources led to longer processing times, with tasks divided into 38 chunks and each taking approximately 90 minutes, resulting in over 2 days of waiting time. This forced us to prioritize project milestones and abandon familiar approaches like embeddings and vector space search.
- Methodological complexity arose from the abundance of available methodologies, requiring extensive research to find the most suitable approach. Integration of new concepts, such as vector stores, presented implementation challenges due to their unavailability within the Databricks platform and the need for paid subscriptions for many resources and models.
- Language model evaluation lacked universally accepted metrics, leading to a trade-off between optimizing the entity matching pipeline and time investment. Additionally, bias persisted in the extraction of the top 200 most common position titles, impacting the mapping of experience titles and influencing aggregated fields.
- Challenges in processing the Education section due to a lack of a matching dataset resulted in categorizing education levels into predefined categories. While effective, implementing this approach earlier could have saved time.

Project specific

- Language models are statistical models, and there is currently no universally accepted set of evaluation metrics for assessing the robustness of LLMs, and there exists a trade-off between optimizing our entity matching pipeline (for example, improving entity matches stages) and the associated time investment.
- The extraction of the top 200 most common position titles is subject to bias. As explained before, we conducted 38 batches of sampling and set a high threshold for match accuracy at 90%. As a result, the bias persisted throughout the processing pipeline, impacting the mapping of experience titles to current position titles, and subsequently influencing any aggregated field.

- Challenges arose in processing the Education section due to a lack of a matching dataset. Consequently, we opted to categorize education levels into predefined categories (e.g., BA, BSc, MSc, etc.), which provided limited informativeness. However, we leveraged this field to illustrate the distribution of education levels for each position by 'sum' aggregation by position groups. While this approach proved effective, implementing it earlier in the process could have saved valuable time.

Conclusions

In summary, our project aimed to leverage AI to provide personalized recommendations for skill development and career advancement on LinkedIn. We explored various AI methodologies, including data integration, natural language processing (NLP), and entity matching, which culminated in the development of a user interactive dashboard. Despite encountering limitations that prompted us to explore different alternatives from our original plan, our results lay a foundation for future research in this field. Our entity matching framework offers valuable insights and practical experience, particularly in implementing our original idea of RAG using Langchain Agent & Vector store and LLMs, which may be highly relevant for next semester's lab project. These findings suggest the potential of our approach to assist LinkedIn members in making informed career decisions. Moving forward, continued research and development efforts could further enhance the effectiveness of AI-driven tools for career guidance on LinkedIn.

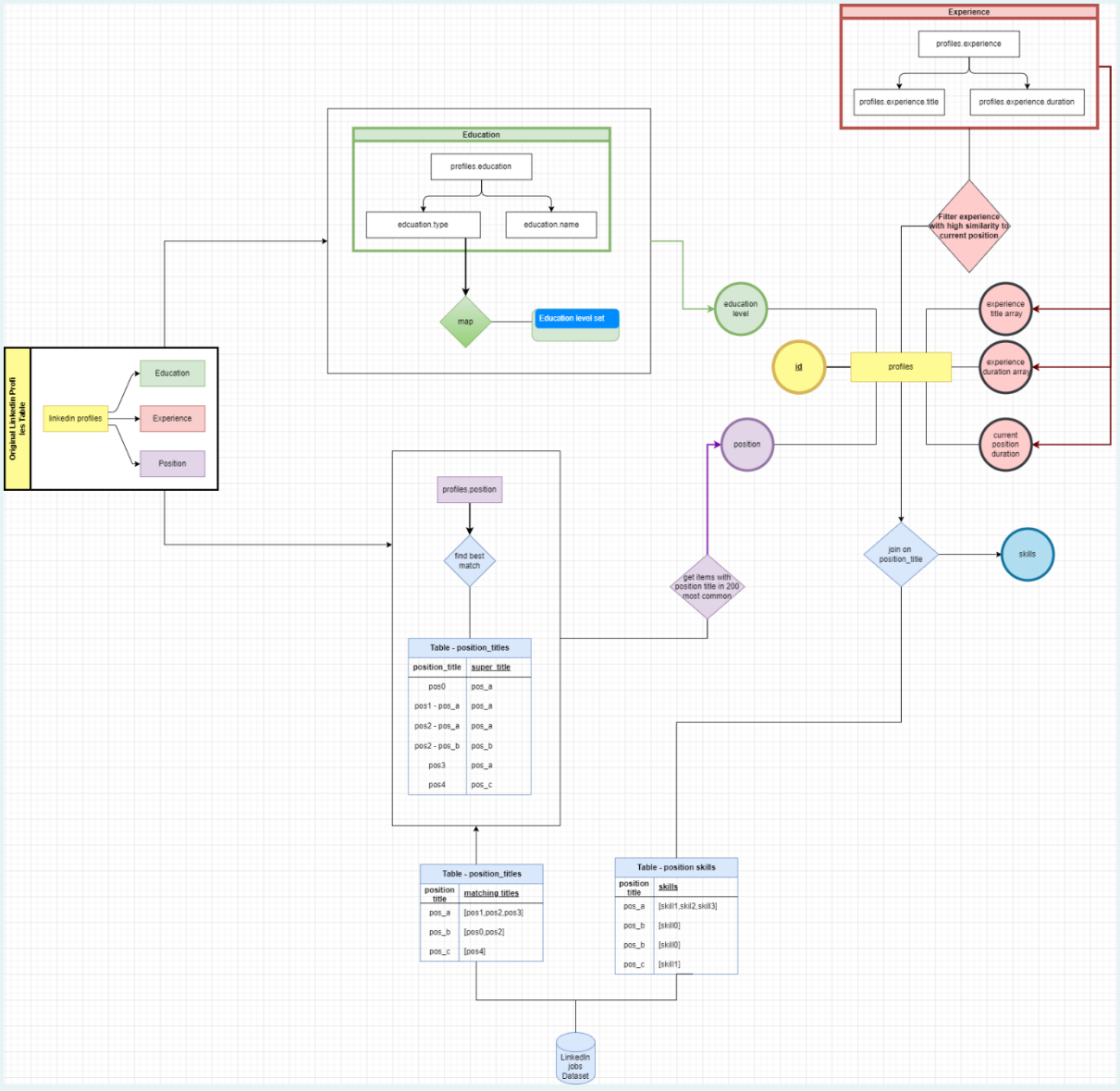
Appendix

Project GitHub Repository https://github.com/TalPeer-github/DataCollectionLab

Project Results Demo App https://datacollectionlab-pmnez6c6fd3cimjg5oqtov.streamlit.app/

Final Table Results Path https://docs.google.com/spreadsheets/d/1BxiQPrdBafpNvLgVd5lB23mrt7zr2jmD6GzHP9LIeNI/edit?usp=sharing

Entity Matching Data Integration Chart (Illustration, not formal)



Entity Match Result table

```
entity_match_results.head(5)
```

Executed at 2024.04.11 21:41:41 in 130ms

5 rows × 6 columns pd.DataFrame

	position	skills	education	original_titles	years_of_experience	position_count
2	Executive Assistant	{'Customer Service': 7.0, 'Communication ski…	{'MA': 24.0, 'BA': 289.0, …	[Executive Assistant, Executive Assista…	[6.91, 9.0, 6.34, 6.2…	975
0	Facilitator	{'Teamwork': 9.0, 'Communication': 15.0}	{'MA': 85.0, 'BA': 216.0, …	[IEP Facilitator, Instructional Designe…	[9.0, 4.42, 4.08, 23.…	624
3	Advisor	{'Sales': 138.0, 'Customer Service': 67.0, '…	{'MA': 149.0, 'BA': 758.0,…	[OD Advisor at Vaya Health, Membership …	[0.0, 8.35, 0.0, 2.84…	2835
1	Project Manager (PM)	{'Customer Service': 6.0, 'Analytical Skills…	{'MA': 106.0, 'BA': 647.0,…	[Project Manager, Physician | Prior Cli…	[8.75, 7.91, 6.0, 18.…	3506
4	Math Teacher	{'Analytical Skills': 1.0, 'Communication sk…	{'MA': 27.0, 'BA': 51.0, '…	[Upper School Science & Math Teacher at…	[7.17, 4.42, 13.17, 7…	236

Gemini Prompt

10

Python

```
prompt = """
Please extract education and experience requirements from the following job posting description following these guidelines:
required_education_item_array is an array of the education requirement items that appear in the job posting description.
all required education mentioned should be extracted and each should appear seperatly in an item containing the following two properties:
    1. type_of_education - string of the type of education. guidelines specific to this type are:
        a. for any non academic education requirement use \"Non-Academic\", including high school
        b. for academic degrees use abbreviations when possible (e.g. \"BS\" instead of \"Bachelor of Science\")
        c. for certificate programmes use \"Certification\"
        d. for diplomas use \"Diploma\" (not high school diplomas)
        e. in any other case or when not sure, extract type \"Other\"
    2. education_in_field - string of the field of education related to type_of_education mentioned above. guidelines specific to this type are:
        a. omit the type from this field as it should be exracted seperatly to the previous field of this item.
        b. use the most general and consice term for the topic of study (e.g. \"Psychology\" instead of \"Behavioral and Clinical Psychological Analysis\")
required_experience_item_array is an array of the experience requirement items that appear in the job posting description.
all required experience mentioned should be extracted and each should appear seperatly in an item containing the following two properties:
    1. experience_in_field - string of the field title or previous position title in which the experience requirement is. guidelines specific to this type are:
        a. use the most general and consice term for the field
        b. if no field or past position is mentioned, extract \"Any\"
    2. minimal_years_of_experience_in_the_field - string of float of the number of years required in the field mentioned above. guidelines specific to this type are:
        a. if time ranges of experience in field are mentioned, extract the lower number (e.g. \"5 to 10 years of experience\" should be \"5.0\", \"7+ years\" should be \"7.0\")
        b. if number of months is mentioned like in \"4 years and 3 months required\", write it as \"4.25\".
        c. if no time related to the experience field is mentioned, write null here.

the job posting description:

"""
```

11

```
result = model.generate_content(prompt + example_description)
```

Titles embeddings results (Try)

3/31/2024 (1s)

14

Python

```
result_df.display()#(truncate=100)
```

(1) Spark Jobs

Table

New result table: OFF

	token	word_embeddings
1	Aeronautics	[-0.24439, -0.58204, 0.10801, -0.091287, -0.094922, -1.1694, -0.47062, 0.26347, -0.51878, 0.56591, 0.38424, -1.2956, 0.17525, 0.042586, -0.99407, -0.83589, 1.1568, 0.88653, 0.20237, -0.1321, -0.17115, 0.10746, -0.017649, -0.31125, 0.36437, -0.2724, -0.25501, 0.64495, -0.61257, -0.29852, -1.1557, 0.27756, -1.2178, 0.0078743, -1.0532, 0.71714, -0.33161, 1.0952, 0.1971, 0.38325, -0.51085, -0.30643, -0.60067, 0.32267, -0.40605, 0.039975, 0.73819, 0.96776, 0.0076974, 0.27197, 0.51555, -0.41369, -0.15076, 0.62077, 0.15567, -0.014375, 0.34509, 0.48512, 1.0388, 0.38331, 0.77563, 0.015549, 0.32254, 0.60092, -0.68299, 0.085444, -0.37597, 0.99719, 0.24001, 1.026, 0.48564, 0.28998, 0.07144, -0.14181, -0.12203, -0.25814, 0.47992, 0.43393, -1.1385, -0.029604, -0.39208, 0.027464, 0.27923, 0.32304, -0.43328, 0.57522, 0.71057, 0.53406, 0.37855, -0.92006, 0.77301, -0.28471, -0.37209, 0.42132, 0.11947, 0.66997, 0.11122, 0.011175, -0.37362, -1.2024]
2	Commission	[0.42907, -0.12272, 0.055479, -0.55187, 0.58443, -0.821, -0.40999, 1.5811, -0.017978, -0.66577, -0.32984, -0.092401, 0.054901, -0.082338, -0.091538, 0.079559, 0.59988, -0.17559, -1.0329, -0.047256, -0.86077, -0.34112, 0.45233, -0.52813, -0.5802, -0.70218, -0.39732, 0.072254, -0.31329, -0.034317, 0.44618, 0.047347, -0.35127, -0.12262, -0.44323, -0.11218, 0.21866, 0.12695, 0.02078, -0.020919, -1.0779, -0.70357, 0.22384, -0.87452, 0.040906, -0.42818, -0.19158, -0.49534, -0.86176, -0.80143, 0.77955, -0.34689, 0.10145, 0.72466, -0.8401, -1.5338, -0.11041, -0.045025, 2.3297, 0.66514, -0.28847, -0.60296, -0.20418, 0.24521, 0.55566, -0.26053, -1.0823, 1.0548, 0.57733, 0.43143, 0.5137, 0.063484, -0.024913, -0.42154, 0.11953, -0.56791, -0.70634, 0.36085, -1.6184, -0.4882, 0.44096, -0.35807, 0.46516, 0.24303, -1.0255, -0.28699, 0.02411, 0.0026135, 0.27219, -0.63039, 0.33668, -0.85397, -0.86422, 0.41496, 0.55116, 0.48124, -0.34899, -0.14697, 0.22617, -0.18687]
	Director	[0.26554, -0.78286, -0.088447, -1.0131, 0.99533, -0.94081, -0.3687, 0.33595, -0.66131, 0.01866, -0.10583, -0.50757, 0.43957, 0.11668, 0.0068358, -0.16246, 0.83221, -0.029842, -0.58107, 0.59797, -0.34653, 0.33801, 0.083349, -0.33689, -0.15555, 0.3037,

2,380 rows | Truncated data | 0.93 seconds runtime

Refreshed 11 days ago

Additional Evaluation phases

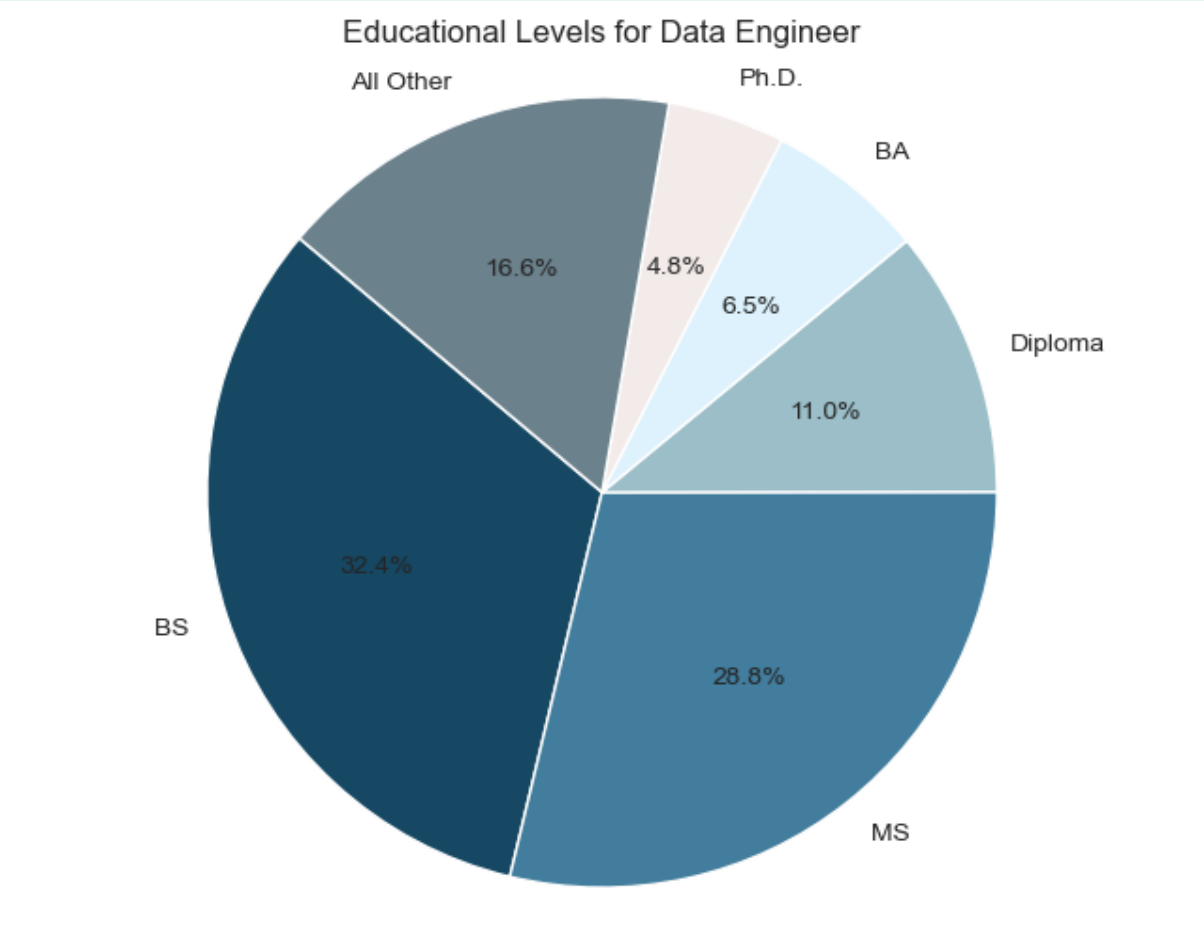
As you can observe, it is very likely that even after the matching to the most common titles as intersection from both tables, some positions still didn't make it (skills extraction process resulted with no skills):

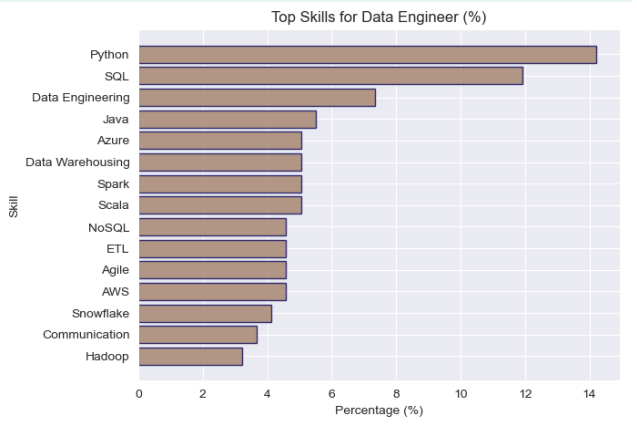
```
unavailable_skills_positions = entity_match_results[entity_match_results['skills'] == {}].position.tolist()
print(f"Skills for positions: \n{unavailable_skills_positions}\nAre not available.")
```

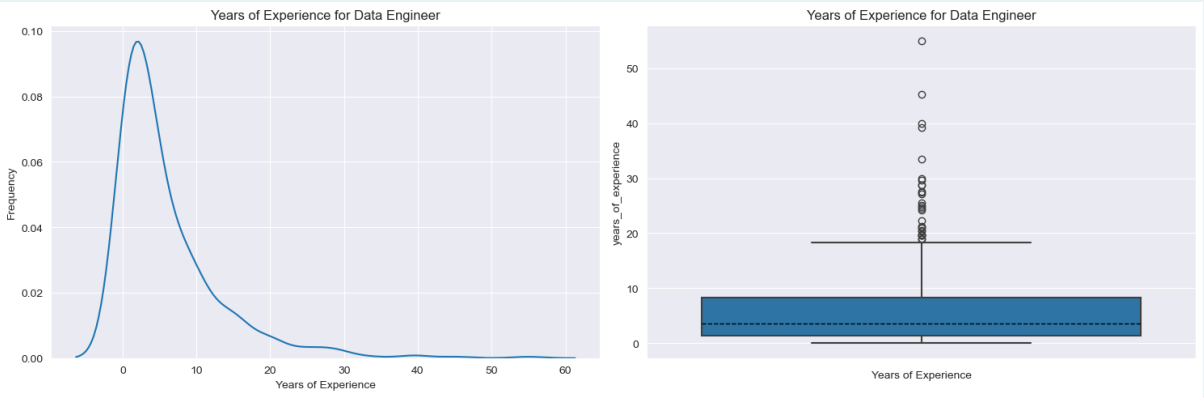
Executed at 2024.04.11 21:48:23 in 100ms

```
Skills for positions:
['Health Director', 'Medical Director', 'Regional Director', 'Clinical Director', 'Academic Advisor', 'Tax Administrative Assistant', 'Executive Director', 'City Engineer', 'Finance
Director', 'Bioinformatics Scientist', 'Client Advisor', 'Statistician', 'Full Stack Software Engineer']
Are not available.
```

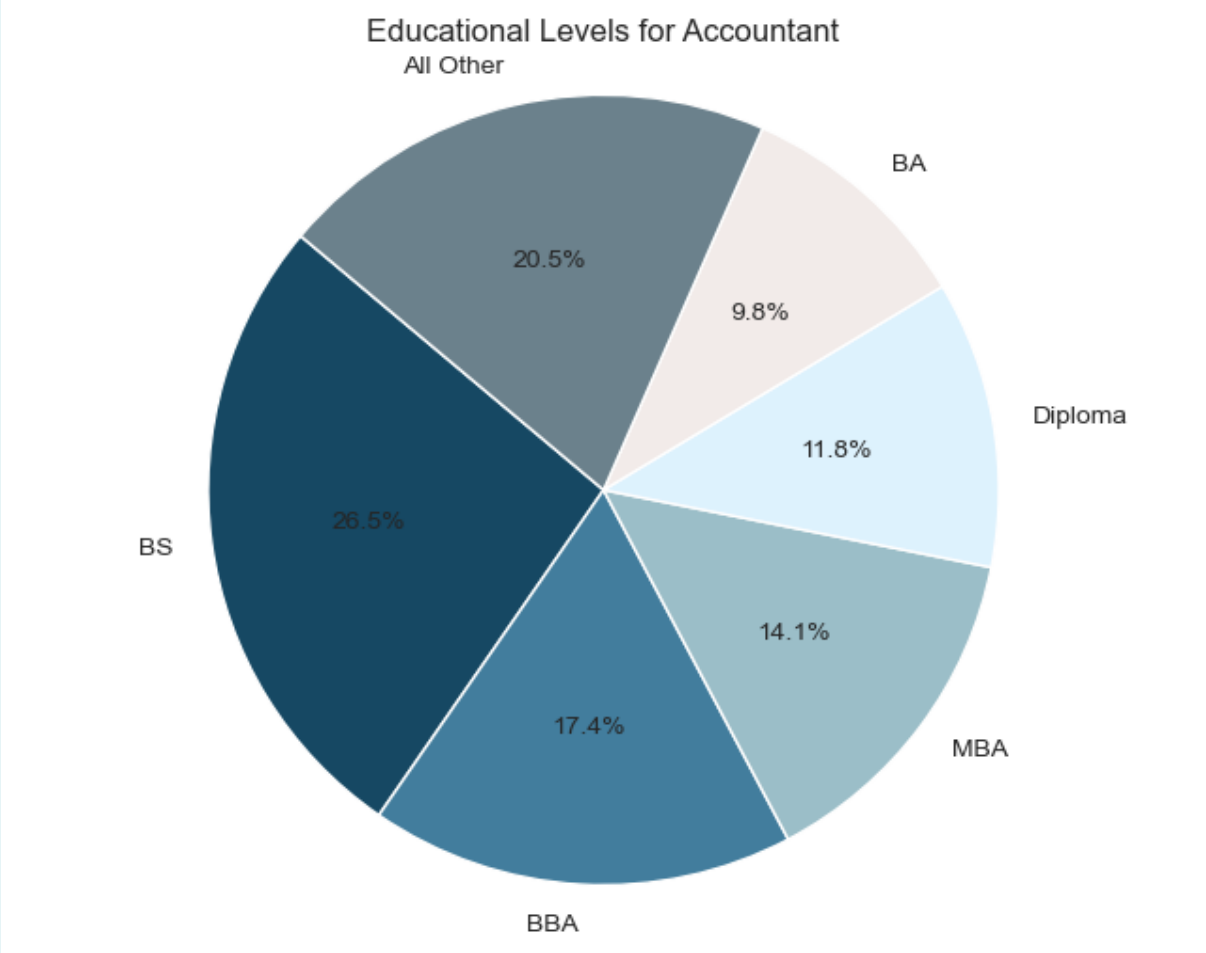
Data Engineer Plots

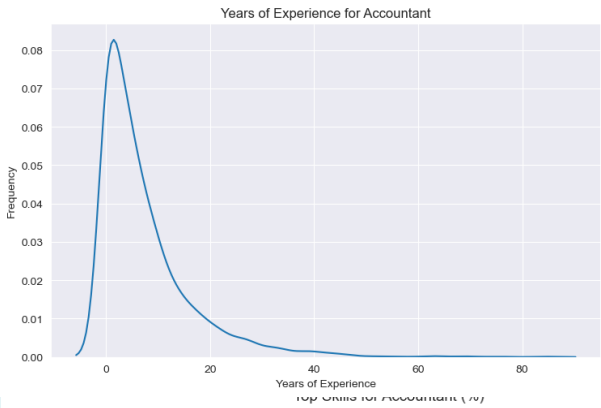


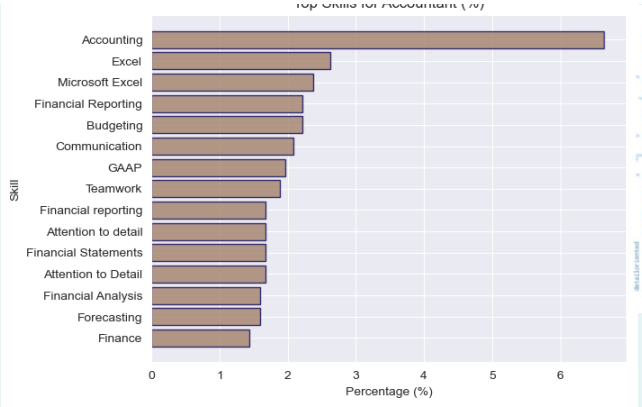



Years of Experience for Data Engineer
Frequency
0.10
0.08
0.06
0.04
0.02
0.00
0
10
20
30
40
50
60
Years of Experience
Years of Experience for Data Engineer
years_of_experience
50
40
30
20
10
0
Years of Experience

Accountant Plots







Dataset	Column	Column Items (for array columns)	Description
Profiles	position	-	current position of the linkedin user
	experience	titles	title of previous occupation
		duration_short	duration of previous occupation in text form (e.g “1 years 5 months”)
	education	degree	type of previously attained education certificate (e.g. PhD)
		field	field of study in previously attained certificate
Job Postings	job_title	-	job title displayed in the job posting
	job_skills	-	relevant skills mentioned in the job posting
	job_summary	-	freeform text of job description, requirements, responsibilities,..
EM Titles	Title	-	class of jobs (e.g. “Chief Executives”)
	Alternate Title	-	specific job titles (e.g. “Aeronautics Commission Director”)

Kolmogorov–Smirnov test

The test statistic of Kolmogorov–Smirnov calculates the maximum difference between the two CDFs, with the null hypothesis assuming identical distributions. If the test statistic exceeds a critical value based on a chosen significance level, the null hypothesis is rejected, indicating significant differences between the distributions. Notably, this test is non-parametric and does not rely on assumptions about the underlying distributions.

References

1. "Skills, not job titles, are the new metric for the labour market " , Jian Lu, 2019
2. "How we mapped the "skills genome" of emerging jobs", Zhichun Jenny Ying, LinkedIn Engineering Blog, 2019
3. "LinkedIn - Future of Skills",
4. "Skills – LinkedIn Data", "THE WORLD BANK" Data Catalog
5. "Understanding Fuzzy String Matching" , Chandra Prakash Bathula, Medium, 2023
6. "Levenshtein Distance" , Wikipedia
7. TheFuzz - Documentation on GitHub
8. "Kolmogorov–Smirnov test", Wikipedia