**Abstract**

Looking for jobs or internships seems a task of its own and the search is no longer based on sole fulfillment of the required job skills, but a lot of networking and recommendations is involved around too. The amount of work involved in finding the correct job builds a great amount of anxiety among the job seekers and the recruiters who want the right talent for their company.

There are two concerns that are to be addressed here. First, matching the job seekers with the right employers and second, provide guidance to aspiring job seekers on the skills that are in demand so that they can build them to stay relevant in the job market.

The job providers and job seekers form a large amount of data which provides for many interesting trends for analysis and interpretation to make the most of data available.

With the data currently available from the seekers and providers, these pitfalls can be fixed. The presence of information on job skills, salaries and user tendencies in many existing websites such as Indeed, LinkedIn, Glassdoor etc. can be utilized to match people to positions which may seem simply impossible without using AI to analyze data.

The jobs database would be a one stop solution to reduce the job search and talent acquisition stress levels. Artificial intelligence (AI) and machine learning can be utilized for complex task of matching work to talent so that it is efficient and less resume spamming.

Domains for Job Database:

1. Finance

2. Health Care

3. Digital Marketing

4. Engineering – Computer vision

5. E-commerce

**Introduction**

This project is aimed to create a list of companies with their link and websites including the job postings. The project will be dealing with two concerns First, matching the job seekers with the right employers and second, provide guidance to aspiring job seekers on the skills that are in demand so that they can build them to stay relevant in the job market. For this study, the data is scraped from www.indeed.com a job listing site for different companies. The relevant variables that will be scrapped due to the nature of the site include: Job Location, Job description, Salary. After collecting the data, data cleaning will be conducted and then the data frame is converted into SQL table to perform use cases, views and Function. Also, we created interface by python switching.

The main goal of this project is to have a list of at least 1500 companies with their website and media pages links for domains such as Finance, Health Care, Digital Marketing, Engineering – Computer vision and E-commerce. Apart from that, we will also have the list of relevant skills required for a domain.

**Methodology**

1. **Scrapped the Data from Indeed.com**

The very first step we followed is scraping data from Indeed.com by using the python module Beautiful Soup. Since the site is relatively well structured and a simple text page hence easy to find the relevant data. Below is the sample view of scrapped data:



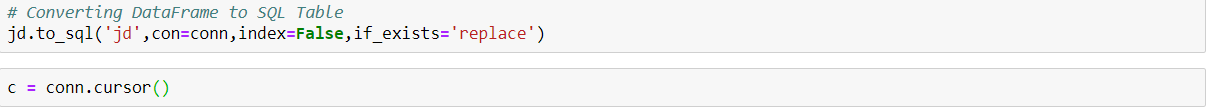
Later we cleaned the data and stored the data in the form of CSV file:



Once the scrapping and cleaning of data is done, we came up with all the below columns:

1. City: Where the job vacancy is available
2. Job: The title of job such as Business Intelligence engineer, Quantitive Analyst manymore
3. Salary: The annual amount of salary
4. Company: The name of company who is hiring
5. Expertise: The field/ skill or area of expertise for a Job posting
6. Recruiter: Name of the hiring manager for a Job Posting

In the next step, we converted the data frame into SQL table.

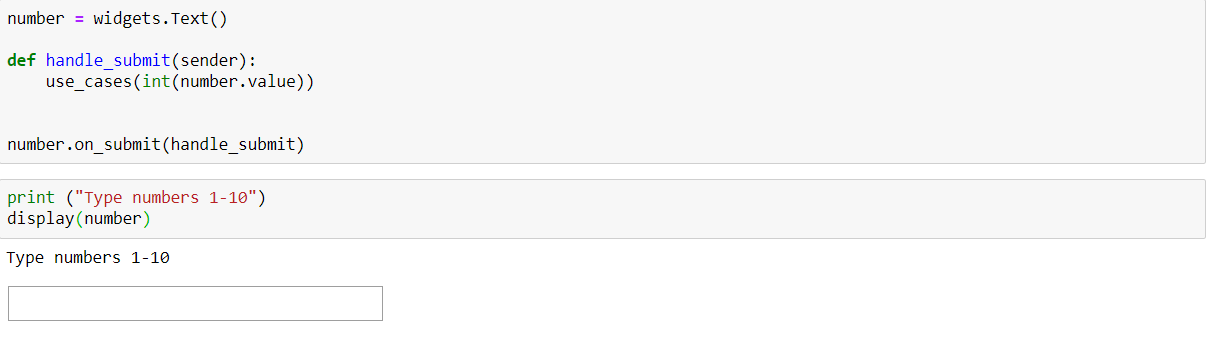


After converting the data frame into SQL table, we performed ten use cases on the Data frame, with the help of creating an interface by python switching. One of the use case we performed are below:

If someone wants to figure out the number of Jobs available according the city, so the first use case is based on the approach of a city like, if you put New York, you will get all the jobs where city is New York and we assigned it as 1. So, when you put 1 in the box below, it will display the result showing all the jobs from New York. Likewise, we did for all the 10 use cases

Use-Cases: 10 use cases have been covered.

1. All jobs in New York
2. All jobs with six figure or more salary
3. All jobs in Seattle and Los Angeles .
4. All jobs with the keyword data
5. All jobs by the recruiter Sophia Corde
6. All jobs related to Data
7. All jobs in New York with at least 100,000$ salary
8. All Analyst jobs in New York
9. All Engineering jobs with at least 100,000$ salary
10. All jobs by any recruiter named John

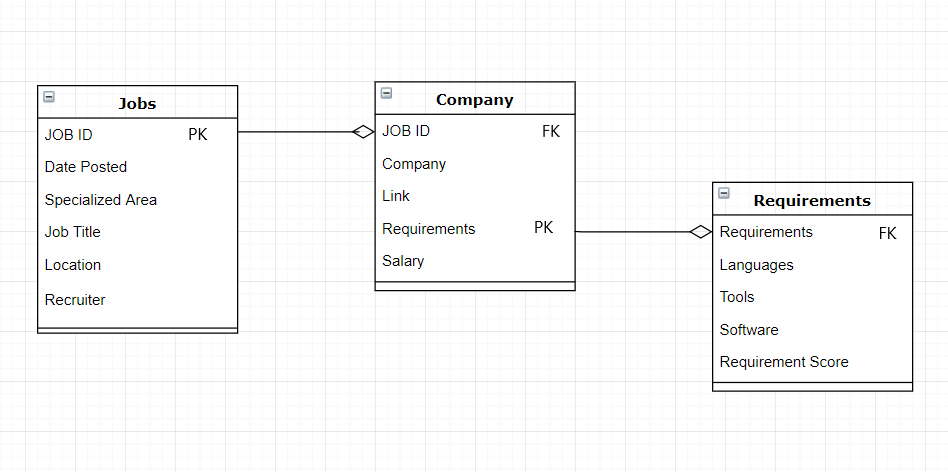


So, we created the interface such that, if you put any number between 1- 10, you will get the result corresponding to that use case.

**Word Cloud**: The next step we did is for the most relevant skills we created a word cloud that looks like:



**ER- diagram**



**Results**

By the end of this project we have a list of 1500 companies with City: Where the job vacancy is available Job: The title of job such as Business Intelligence engineer, Quantitative Analyst many more, Salary: The annual amount of salary Company: The name of company who is hiring, Expertise: The field/ skill or area of expertise for a Job posting, Recruiter: Name of the hiring manager for a Job Posting. Apart from that, we will also have the list of relevant skills required for a domain.

**References**

<https://github.com/skunkworksneu/Projects/blob/master/Jobs%20Database.pdf>

<https://stackoverflow.com/questions/41792471/installing-wordcloud-using-jupyter-notebook>

<https://www.kaggle.com/adiljadoon/word-cloud-with-python>

<https://stackoverflow.com/questions/46253288/scrape-with-correct-character-encoding-python-requests-beautifulsoup>

<https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.get_dummies.html>

<https://towardsdatascience.com/the-dummys-guide-to-creating-dummy-variables-f21faddb1d40>

<https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.cross_val_score.html>

<https://scikitlearn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html>

<https://machinelearningmastery.com/arima-for-time-series-forecasting-with-python/>