JOBS DATABASE PROJECT
DMDD
Prof. Nik Brown

PARTH SURESH BHANUSHALI (NUID: 001409809)

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.

Introduction

There are various sites that have number of posting of jobs for different domain that are posted on site. These jobs are random for different post and different city and have no proper format to these jobs. Aim of creating this database is to scrape jobs for a specific domain, in this case finance, to make it easier to read and to find for the dream job. The user can search for the jobs with a variety of use cases depending on the salary that is offered, or location they want to work in some city (location like Boston). Also type of job that is offered that are Internship, Part Time, Full Time or Contract base. This code uses beautiful soup to extract data from Indeed.

ER Diagram

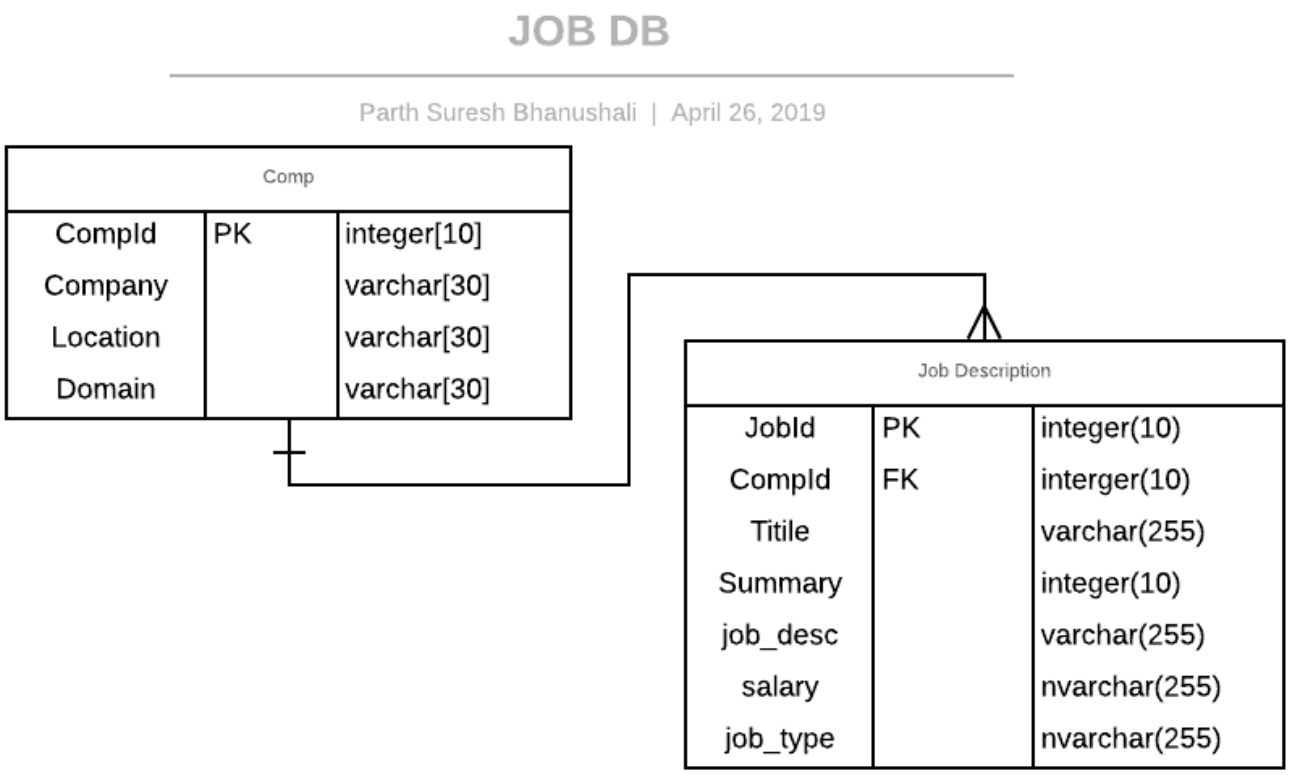

Table 1: Comp_details
Primary key is CompID

Table 2: Job_details
Primary key is JobID
CompID key is Foreign key

Code

Step 1: Importing all the required libraries. Installing the library in the anaconda command prompt.

Importing libraries

In [2]:
```
from bs4 import BeautifulSoup
import requests
import re
import pandas as pd
from nltk import bigrams
from nltk.corpus import stopwords
from nltk.stem import WordNetLemmatizer
from nltk.tokenize import word_tokenize
import string
import matplotlib as mlt
import matplotlib.pyplot as plt
%matplotlib inline

from subprocess import check_output
from wordcloud import WordCloud, STOPWORDS
```

In [3]:
```
import urllib3
```

Step 2: Declaring the URL of the source

Declaring the source and prettifying to make HTML readable (Indeed)

In [4]:
```
source = requests.get('https://www.indeed.com/q-finance-l-Boston,-MA-jobs.html').text
```

In [5]:
```
soup = BeautifulSoup(source, 'lxml')
```

In [6]:
```
print(soup.prettify())
```

```
<!DOCTYPE html>
<html dir="ltr" lang="en">
 <head>
  <meta content="text/html;charset=utf-8" http-equiv="content-type"/>
  <script src="/s/f2cb3a7/en_US.js" type="text/javascript">
  </script>
  <script>
   !function(n){function r(n){for(var r=a,t=n.length;t;)r=33*r^n.charCodeAt(--t);return r>>>0}var t=this['indeed.i18n.localeD
ata'],e=t['']||{},a=e.salt;if(e.hasOwnProperty('salt'))for(var i in n)t[function(n){var t=r(n);return e.hasOwnProperty('id_le
ngth')&&(t=String(t).substring(0,e.id_length)),t}(i)]=n[i];else for(var i in n)t[i]=[null].concat(n[i])}({"indeedapply_serp_l
abel":["Apply instantly"]});
  </script>
  <link href="/s/97464e7/jobsearch_all.css" rel="stylesheet" type="text/css"/>
  <link href="http://rss.indeed.com/rss?q=finance&amp;l=Boston%2C+MA" rel="alternate" title="Finance Jobs, Employment in Bost
on, MA" type="application/rss+xml"/>
  <link href="/m/jobs?q=finance&amp;l=Boston%2C+MA" media="only screen and (max-width: 640px)" rel="alternate"/>
  <link href="/m/jobs?q=finance&amp;l=Boston%2C+MA" media="handheld" rel="alternate"/>
  <script type="text/javascript">
   if (typeof window['closureReadyCallbacks'] == 'undefined') {
        window['closureReadyCallbacks'] = [];
```

Step 3: Inspecting the elements to extract data from the html page after the data is prettyfied.

Inspecting elements of the HTML to scrape tags and links

In [7]:
```
indeed_jobs_page = soup.find('div', class_="jobsearch-SerpJobCard") #defining class of the page to be extracted
```

In [8]:
```
len(indeed_jobs_page) # checking length of jobs on the page
```

Out[8]: 11

In [9]:
```
url = f"https://www.indeed.com/jobs?q=finance&l=MA"
link = requests.get(url)
page = BeautifulSoup(link.content, 'html.parser')
```

In [10]:
```
x = page.select('.jobsearch-SerpJobCard')
```

In [11]:
```
x[0].select('.jobtitle')[0].text.strip() #class of job titles
```

Out[11]: 'Finance Assistant'

In [12]:
```
x[0].select('.company')[0].text.strip() #class of company
```

Out[12]: 'Grassroots Campaigns'

In [13]:
```
x[0].select('.location')[0].text #class of location
```

Out[13]: 'Boston, MA'

Step 4: Creating loop for company and job details table and storing the file to CSV.

Creating a loop to extract all companies and create dataframe.

In [25]:
```
page_extract = page.select('.jobsearch-JobComponent-description')
```

In [30]:
```
company = []
location = []
start = 0
for i in range(25):
        url = f"https://www.indeed.com/jobs?q=finance&l=Boston%2C+MA&start={start}"
        link = requests.get(url)
        page = BeautifulSoup(link.content, 'html.parser')
        start += 10
        for block in page.select('.jobsearch-SerpJobCard'):
            company.append(block.select('.company')[0].text.strip())
            location.append(block.select('.location')[0].text)
#creating loop for company table
```

In [32]:
```
title = []
summary=[]
salary = []
extract_text = []
#urltext = []
job_desc = []
start = 0
for i in range(50):
        url = f"https://www.indeed.com/jobs?q=finance&l=Boston%2C+MA&start={start}"
        link = requests.get(url)
        page = BeautifulSoup(link.content, 'html.parser')
        start += 10
        for block in page.select('.jobsearch-SerpJobCard'):
            title.append(block.select('.turnstileLink')[0].text.strip())
            #urltext.append(block.select('.turnstileLink')[0]['href'])
            summary_url = "https://www.indeed.com" + block.select('.turnstileLink')[0]['href']
            summary_page = requests.get(summary_url)
```

Step5: In this step, I filtered all the data for all the null values. Stored the final csv file.

```
In [55]: indeedjob_filter = pd.read_csv('indeedjob14.csv')

In [ ]: indeedjob_filter.isnull()

In [ ]: indeedjob_filter.isnull().sum()

In [ ]: indeedjobs = indeedjob_filter.dropna()

In [ ]: indeedjobs.isnull().sum()

In [ ]: indeedjobs.to_csv('indeedjobs1.csv',index=False)

In [ ]: indeedjob14.csv=indeedjobs1.dropna()
```

I also tried to extract data using n grams. I had written a code which can be used to extract data in better format. I Also wrote code for word cloud but could not implement due to lack of time in this project.

In [ ]:
```
def process_text(text):

        text = text.lower()
        text = text.replace(',', ' ')
        text = text.replace('/', ' ')
        text = text.replace('(', ' ')
        text = text.replace(')', ' ')
        text = text.replace('.', ' ')
        text = text.replace('[', ' ')
        text = text.replace(']', ' ')
        text = text.replace('"', ' ')

 # Convert text string to a list of words
        return text.split()


def generate_ngrams(words_list, n):
    ngrams_list = []

    for num in range(0, len(words_list)):
        ngram = ' '.join(words_list[num:num + n])
        ngrams_list.append(ngram)

    return ngrams_list

"""
if __name__ == '__main__':
    words_list = process_text(text)
    unigrams = generate_ngrams(words_list, 1)
    bigrams = generate_ngrams(words_list, 2)
    trigrams = generate_ngrams(words_list, 3)
"""
```

In [ ]:
```
def data_cleaning(combined_text):
    text = combined_text.lower() #converts everything to lower case charecters
    text = re.sub('[%s]' % re.escape(string.punctuation), '', text)#replaces all the punctuations with none
    text = re.sub('\w*\d\w*', '', text) #removes all the numbers
    text = text.replace('\n','') #replaces all the tab charecters
    text = text.replace('\t','')#replaces all the tab charecters
    text = [i for i in word_tokenize(text) if i not in stop_words] #removing stop words
    lemmatizer = WordNetLemmatizer()
    text = [lemmatizer.lemmatize(word) for word in text] #Lemmantizing every word

    return text
```

In [ ]:
```
stop_words = stopwords.words('english') + list(string.punctuation) #List of stop words and punctuations
```

In [ ]:
```
generate_ngrams(data_cleaning(y[0].text), 5)
```

In [ ]:
```
import matplotlib as mpl
```

In [50]:
```
generate_ngrams(data_cleaning(y[0].text), 3)
```

Out[50]:
```
['yearthis position requires',
 'position requires intelligent',
 'requires intelligent flexible',
 'intelligent flexible person',
 'flexible person willing',
 'person willing learn',
 'willing learn new',
 'learn new task',
 'new task problem',
 'task problem solve',
 'problem solve advance',
 'solve advance mission',
 'advance mission makeawish®',
 'mission makeawish® massachusetts',
 'makeawish® massachusetts rhode',
 'massachusetts rhode island',
 'rhode island generalist',
 'island generalist role',
 'generalist role responsible',
 'role responsible performing',
 'responsible performing routine',
 'performing routine accounting',
 'routine accounting benefit',
 'accounting benefit administration',
 'benefit administration operational',
 'administration operational function',
 'operational function support',
 'function support finance',
 'support finance operation',
 'finance operation department',
 'operation department organization',
```

In [ ]:
```
def wordcloud_draw(data, key, color='black'):
    words = ' '.join(data)
    cleaned_word = " ".join([word for word in words.split()
                             if 'http' not in word
                                 and not word.startswith('@')
                                 and not word.startswith('#')
                                 and word != 'RT'
                             ])
    wordcloud = WordCloud(stopwords=STOPWORDS,
                      background_color=color,
                      width=2500,
                      height=2000
                     ).generate(cleaned_word)
    plt.figure(1,figsize=(13, 13))
    plt.imshow(wordcloud)
    plt.axis('off')
    plt.title('Wordcloud of key "{}"'.format(key))
    plt.savefig('{}.png'.format(key))
    plt.close()
```

In [ ]:
```
for key, value in job_desc.items():
    num_Descriptions = len(value)
    if (num_Descriptions >= 50):
        num_Descriptions
        wordcloud_draw(value["Description"], key, 'white')
```

In [ ]:
```
df = pd.read_csv('indeedjobs_no_rep.csv')
```

In [ ]:
```
df.shape
```

In [ ]:
```
stopwords = set(STOPWORDS)

def mywordcloud(data, title = None):
    wordcloud = WordCloud(
        background_color='white',
        stopwords=stopwords,
        max_words=200,
        max_font_size=40,
        scale=3,
        random_state=1
    ).generate(str(data))

    fig = plt.figure(1, figsize=(20, 20))
    plt.axis('off')
    if title:
        fig.subtitle(title, forsize=20)
        #fig.subplots_adjust(top=2,3)

    plt.imshow(wordcloud)
    plt.show()

mywordcloud(df["job_desc"].dropna())
```

Creating use cases using SSMS

```
-------------creating table for Companies----------------------


Drop table comp_details

Create Table Comp_details(
    CompID INTEGER PRIMARY KEY NOT NULL,
       company nvarchar(MAX),
       location_job nvarchar(MAX),
       domain nvarchar(MAX),
);

Select * from Comp_details
```

```
-------------creating table for JOBS----------------------

Drop table Job_details

Create Table Job_details(
    JobID INTEGER PRIMARY KEY NOT NULL,
       CompID INTEGER references Comp_details(CompID),
       title nvarchar(max),
       summary nvarchar (max),
       job_desc nvarchar(max),
       salary nvarchar(max),
       job_type nvarchar(max),
);

Alter Table Job_details
       Add FOREIGN KEY (CompID) REFERENCES Comp_details(CompID);

Select * from Job_details


----------------------------------USE CASES-----------------------------------------


--1. Stored procedure for companies with analyst postions. #as I was interested only
analyst positions in Finance domain

drop procedure analyst

create procedure analyst
as
select *
from Job_details
where title like '%Analyst%';
GO

EXEC analyst
```

	JobID	CompID	title	summary	job_desc	salary	job_type
1	4	103	Entry Level Business Analyst	Atlantic Trans Trading seekin...	$70,000 - $75,000 ...	$70,000 - $75,000 a year	Full time
2	13	112	Financial Analyst	The Financial Analyst is resp...	Company: Private ...	$53,000 - $75,000 a year (Indeed ...	Full time
3	15	114	Business Analyst with Capital...	5+ years' work experience a...	$45 - $58 an hourC...	$45 - $58 an hour	Contract
4	16	115	Analyst, Finance Rotational ...	Analyst, Finance Rotational ...	Analyst, Finance R...	$61,000 - $82,000 a year (Indeed ...	Full time
5	19	118	Sr Analyst, Program and Proj...	Provides support for activitie...	$54,000 a yearPro...	$54,000 a year	Part time
6	20	119	Junior Financial Analyst	3 years of Operations, finan...	$40,000 - $60,000 ...	$40,000 - $60,000 a year	Part time
7	27	126	Sales Solutions Analyst	Candidate will come from an...	$90,000 - $100,00...	$90,000 - $100,000 a year	Contract
8	30	129	Financial Analyst	The Financial Analyst will su...	$25 - $30 an hourC...	$25 - $30 an hour	Contract
9	31	130	Financial Analyst I	Analyst is responsible for:. S...	Analyst is responsi...	$41,000 - $58,000 a year (Indeed ...	Part Time
10	34	133	Budget and Policy Analyst	In support of state and feder...	$62,530 - $90,570 ...	$62,530 - $90,570 a year	Full time

```
--2. Jobs that are offered on contract basis #to search for jobs on contract basis

create procedure Contracttype as
select cd.company, cd.location_job, jb.title, jb.salary, jb.job_type from Comp_details as
cd
inner join Job_details as jb on
cd.CompID=jb.CompID where  jb.job_type='Contract';
Go

exec Contracttype

drop view Type_job
```

	company	location_job	title	salary	job_type
1	Wework Solutions Inc	Boston, MA	Business Analyst with Capital Markets	$45 - $58 an hour	Contract
2	Matrss	Boston, MA	Sales Solutions Analyst	$90,000 - $100,000 a year	Contract
3	Lightning Asset Managem...	Lexington, MA	Chief Investment Officer	$40,000 - $120,000 a year	Contract
4	Roessel Joy	Cambridge, MA	Financial Asisstant	$23 - $25 an hour	Contract
5	Andover Personnel	Peabody, MA	Financial Analyst	$25 - $30 an hour	Contract
6	JOCRF	Boston, MA 02116 (South End area)	Payroll and A/P Administrator PT	$32 - $34 an hour	Contract
7	Roessel Joy	Roslindale, MA	Junior Accountant	$22 - $25 an hour	Contract
8	General Indemnity Group	Boston, MA	Accounts Payable/Receivable Assis...	$15 - $25 an hour	Contract
9	Accenture	Boston, MA 02199 (Back Bay-Beaco...	Finance & Risk - ERP Digital Financ...	$92,000 - $136,000 a year (In...	Contract

```
--3. View for jobs that are offered for full time. #to find number of full time jobs
offered

create view fullTime(job_type,jobID)
AS SELECT job_type,COUNT(*)
FROM Job_details
WHERE job_type like '%Full time%' GROUP BY job_type;

select * from fullTime;


SELECT * FROM C.CompId CID
LEFT OUTER JOIN C.location_job CLOC ON CID
FROM Comp_details AS C, Job_details AS J
INNER JOIN Comp_details ON C.CompID=J.CompID;
```

	job_type	jobID
1	Full time	54

```
--4. Jobs that are offer intermediate level of salary (around $50000)? #to find jobs with
salary in and around range of $50000

drop procedure more_salary

create procedure more_salary
as
select cd.company, cd.location_job, jb.title, jb.salary, jb.job_type from Comp_details as
cd
inner join Job_details as jb on
cd.CompID=jb.CompID where  jb.salary like '$5%';
go

EXEC more_salary
```

	company	location_job	title	salary	job_type
1	Make A Wish Massachusetts a...	Boston, MA 02110 (Central area)	Coordinator, Finance and Operati...	$50,000 - $60,000 a year	Full time
2	QABASA	Boston, MA	Manual Tester/QA - Entry Level	$55,000 - $60,000 a year	Full time
3	Mission Wealth	Boston, MA	Client Advisor Associate	$55,000 - $80,000 a year	Full time
4	xNexvenco	Boston, MA	Financial Analyst	$53,000 - $75,000 a year (Indeed...	Full time
5	BNY Mellon	Everett, MA 02149	Sr Analyst, Program and Project ...	$54,000 a year	Part time
6	JDJ Family Office Services	Boston, MA 02109 (Central area)	Staff Accountant - Entry Level	$57,000 - $74,000 a year (Indeed...	Full time
7	Beacon Dental Health Manage...	Boston, MA 02108 (Back Bay-Bea...	Staff Accountant	$52,000 - $60,000 a year	Full time
8	UMASS	Cambridge, MA 02142 (East Camb...	Assistant Director for Financial Ed...	$54,954 - $68,000 a year	Part time
9	Cybereason	Boston, MA 02116 (South End area)	Financial Analyst	$59,000 - $83,000 a year (Indeed...	Full time

```
--5. view for Jobs in around Boston area. # to search for jobs in Boston Location

create view Boston_Strong as
select * from Comp_details as cd
where cd.location_job='Boston, MA';

select * from Boston_Strong
```

	CompID	company	location_job	domain
1	101	McAdam Financially Advanced	Boston, MA	Finance
2	103	Atlantictranstrading	Boston, MA	Finance
3	104	QABASA	Boston, MA	Finance
4	105	Siharum Advisors, LLC	Boston, MA	Finance
5	109	Mission Wealth	Boston, MA	Finance
6	110	Boston Planning & Development Agency	Boston, MA	Finance
7	112	xNexvenco	Boston, MA	Finance
8	114	Wework Solutions Inc	Boston, MA	Finance
9	124	Boston University	Boston, MA	Finance
10	125	Adidas	Boston, MA	Finance

Conclusion

We were able to create a database for finance domain from indeed using beautiful soup of 100 job postings. Use cases are made using the MICROSOFT SQL DATABASE SERVER to be able to search for job as per required parameters.

Reference

1. https://nycdatascience.com/blog/student-works/project-3-web-scraping-company-data-from-indeed-com-and-dice-com/
2. https://stackoverflow.com/
3. https://code.likeagirl.io/how-to-use-python-to-remove-or-modify-empty-values-in-a-csv-dataset-34426c816347
4. https://github.com/indeedlabs/indeed-python
5. https://github.com/robagwe/kick-off-web-scraping-python-selenium-beautifulsoup
6. https://github.com/
7. https://realpython.com/python-data-cleaning-numpy-pandas/