COSC329 project report

Zhenghua Chen

My project is to analyse the required skills for Data Scientist and Software Developer in Canada.

Scraping

1. prepare

Before scraping, I firstly looked at the indeed website in Canada https://ca.indeed.com to get a general idea. After I searched in the box, the web page will show pages of results and each result has a link which leads to that specific job page.

My general method is to go into the search page first, get all the result links on that page, read each job's description through each link and then go to the next page, do the same thing until I collect enough posts.

2. pipeline

I used python package requests and BeautifulSoup for completing this pipeline. All the links in the search page were identified by looking for all the a (link) item under the div item with id "mosaic-provider-jobcards". After a posting link is accessed, its job description was identified by the div item with id: "jobDescriptionText" and class "jobsearch-jobDescriptionText". The next page on the search page was found by ul item with class "pagination-list" and the link item with "aria-label":"Next" inside the class.

Some links lead to the company's own website which don't have the exact class as above, but it's only a few posts. In that case I output a link of this job and the job description can be manually extracted later.

I make the process sleep for 3 seconds before scraping each page to avoid high frequency. A random user agent from a list was used in requests header to keep from being blocked by the website.

After I scraped one job posting, I wrote it directly to a csv row with job title, location, description and link for later use.

3. results

Though the code can finish successfully with no error, the jobs I got were far less from the number of search results showed on the website. I only got 1114 Software Developer job postings and 939 Data Scientist job postings from indeed, while the total jobs would be over 10000. The scraped jobs were also not in the same order as I saw on the web page. I checked the csv file and it looked good with correct information scraped. I didn't figure out why this happened so I left this as a unfinished feature of my project.

Analysis

1. prepare

Before processing the csv file with all the job information, I searched online for effective methods. The most popular methods for extracting text features online were bag-of-words and tfidf vectorization. I adopted the python package sklearn.feature_extraction.text and the useful tools CountVectorizer and TfidfVectorizer to try on my work. I also decided to use the package RegEx for cleaning and nltk.wordpunct_tokenize for tokenizing the strings. I used the lemmatize function from the package textblob for lemmatization and English srop words from nltk.corpus for removing unrelated words. Pandas.DataFrame and numpy were also utilized for dealing with large amount of data.

2. methods

The first step overall was to read the data from csv file and organize them in the pandas dataframe. I removed the rows with NaN and the did the general cleaning by removing special characters, digits and underscores. Then all the texts were converted to lowercase.

I developed two methods for extracting features. The first one is to use tfidf-vectorizor from the package directly.

In this method I first cleaned data again by removing non-english words, removing stop words, lemmatization and dropping words less than 2 chars. When I was developing and testing on the workflow, I noticed that there were still a lot of noise in the text, so I added an input of list which declares the extra stop words that need to be removed. Then the cleaned text string was converted to word list and was put into the TfidfVectorizer with max_df=2 and min_df=0.5. After running the function fit_transform on the corpus, all the features were extracted and their tfidf scores were listed for each job. Then I get the top n features by averaging the tfidf scores for each feature and ranking them.

The second method was more of my own techniques. I found out that only using tfidf vectorizor and basic cleaning did not work very well since there were still a large number of less-important features about the company, general description or other noise. To get a more precise outcome, I believed that the texts need to be further preprocessed. Based on our searching habits, I created a customized searching method to look for the section in the job posting named “requirements”, “qualifications” or “skills”. Though a lot of jobs do not have these section names for skills but have “who you are”, “what you’ll need to do”, around 5/7 of all the posts contained these noticeable keywords. Thus, I tried to extract only a proportion of words following these key words. Notice that because of web scraping, sometimes these keywords were concatenated with other words such as “ ……a sentence endSkills you need….”. In this way we should search for the substring instead of the tokenized words.

Since I didn’t want to clean too much to before finding the patterns for this approach, I just removed the stop words and lemmatized the text so that “skills” would all become “skill” for easier detection. Then I extracted 80 words after each occurrence of the keywords ["requirement", "qualification", "skill"] for each text. If none of the keywords appeared, I searched for the keyword substrings in the text string and extracted the following 640 chars. Finally, the extra stop words were removed and a list of words were gained from the extracted string. Then the tfidf and count vectorizors were applied on these words.

Finally the top 20 features of each method were showed and I calculated the overlap of them using bhattacharyya_coefficient function from distance package. A heatmap was plotted to show the final comparison.

3. tests and results

Data Scientist:

For method 1:

```
[('data', 0.4745309723037524), ('experience', 0.2562409372411891), ('learn
', 0.21741967975425838), ('business', 0.19430043251834614), ('team', 0.185
43229194463734), ('machine', 0.1529591042978648), ('machine learn', 0.1484
5198480721547), ('analytics', 0.13411077451607425), ('science', 0.12741647
448115886), ('build', 0.11699428945000243), ('analysis', 0.105933550523799
4), ('engineer', 0.09744002242886084), ('new', 0.09501028884584822), ('kno
wledge', 0.09391096055039992), ('data science', 0.09095950302422869), ('he
lp', 0.08879748154398966), ('design', 0.08771769283194432), ('support', 0.
08477634817727388), ('technical', 0.08282301184291198), ('understand', 0.0
8174505460120045)]
```

For method 2:

Count vectorization:

```
[('data', 7.4185303514377), ('experience', 5.567625133120341), ('business
', 2.617678381256656), ('learn', 2.5228966986155483), ('team', 2.252396166
134185), ('science', 1.8519701810436635), ('machine', 1.516506922257721),
('machine learn', 1.4494142705005324), ('model', 1.428115015974441), ('ana
lytics', 1.3258785942492013), ('knowledge', 1.3194888178913737), ('analysi
s', 1.2523961661341854), ('python', 1.1448349307774228), ('process', 1.106
496272630458), ('engineer', 1.066027689030884), ('problem', 1.042598509052
1832), ('degree', 1.012779552715655), ('project', 1.0021299254526093), ('u
nderstand', 0.9957401490947817), ('data science', 0.9850905218317358)]
```

TFIDF vectorization:

```
[('data', 0.2877633101224421), ('experience', 0.2756532150624001), ('learn
', 0.15325665146712186), ('business', 0.14650920636848752), ('team', 0.144
2797654502848), ('science', 0.1101214687427899), ('knowledge', 0.100525090
21414582), ('machine', 0.09537498590419481), ('model', 0.0930319414542899
6), ('machine learn', 0.09281667674616376), ('analytics', 0.08670057399632
608), ('python', 0.08591469951180035), ('process', 0.0827930727525265), ('
engineer', 0.08192012619924995), ('analysis', 0.07932254545507077), ('prob
lem', 0.07841315525590559), ('project', 0.07561435962042432), ('understand
', 0.07076068422663422), ('degree', 0.06712962710563769), ('data science',
 0.06681374071601766)]
```

From results above, we can see the most important feature is data, which is apparent for this job. Experience is also very important for any industry work in common sense. It also contains “business”, “machine learn”, “model”, “analytics” and “python”, which are all must-have skills to be a data scientist. We can also conclude that most of the jobs of data scientist are related to business company. It is obvious since most business companies need to process and analyse the huge amount of data behind customer behaviors to improve their service correspondingly. Machine learning is especially popular among all the techniques due to its power of identifying, predicting and extracting. A degree is desired for this position too.

Software Developer:

For method 1:

```
[('experience', 0.36315990797403175), ('team', 0.2899107897934766), ('desi
gn', 0.1989588877278291), ('test', 0.16643072740503162), ('code', 0.160402
6579337243), ('knowledge', 0.15305727672595665), ('build', 0.1491454108772
4317), ('technical', 0.13604905620153687), ('technology', 0.13468307185121
287), ('environment', 0.1340344319246207), ('er', 0.13389282520859083), ('
computer', 0.13382761218268285), ('company', 0.13219381676187777), ('job',
 0.125825006138966), ('full', 0.12391009289097059), ('look', 0.11351757572
75613)]
```

For method 2:

Count vectorization:

```
[('experience', 4.693140794223827), ('software', 3.1958483754512637), ('de
sign', 2.1796028880866425), ('team', 2.032490974729242), ('test', 1.789711
19133574), ('application', 1.6525270758122743), ('knowledge', 1.4801444043
3213), ('technology', 1.3005415162454874), ('system', 1.2851985559566788),
 ('technical', 1.2382671480144405), ('code', 1.227436823104693), ('compute
r', 1.167870036101083), ('programm', 0.990072202166065), ('understand', 0.
9575812274368231), ('environment', 0.9467509025270758), ('engineer', 0.914
2599277978339), ('web', 0.9106498194945848), ('process', 0.84205776173285
2), ('communication', 0.8312274368231047), ('degree', 0.8185920577617328)]
```

TFIDF vectorization:

```
[('experience', 0.2790188128574318), ('software', 0.1771664861509763), ('t
eam', 0.14653905103455686), ('design', 0.12570037361211184), ('knowledge',
 0.11043349699422986), ('test', 0.11042675687065157), ('application', 0.10
563755812287254), ('technology', 0.1031428028807942), ('code', 0.092641828
75744362), ('technical', 0.08866975924485232), ('computer', 0.085730861581
9803), ('environment', 0.08293039167910356), ('understand', 0.081914925835
01561), ('web', 0.08173344515569282), ('system', 0.08120392553333655), ('p
rogramm', 0.07673117613265693), ('engineer', 0.07405795043957615), ('commu
nication', 0.0728312294025619), ('degree', 0.07221857784491101), ('process
', 0.06680401883645198)]
```

From the above results about software developer, we can see that the most important feature is experience instead of data. That’s the biggest difference from data scientist jobs since developers don’t need to deal with a large size of data. Their jobs are mostly about: “software”, “design”, “test”, “application”, “code”, “web”, “system” and “program”. So, it’s not surprising to see that these are the required skills.

To compare the preprocessing techniques of the two methods, I further checked the extracted text before vectorization of a random data scientist job to see the difference.

```
Data Scientist checking...
After cleanning using method 1:
challenge company organization role collaborate platform global collaborat
ive present problem goal speed knowledge capital technology scientist help
 equity python computer page job support keep growth complex structured ev
ery acumen better passion think analytical make per combine degree everyon
e aggregate paced create raw intelligence field active analyze presentatio
n research industry eye extract business algebra data preferred learn full
 team knack private engineer across information innovative rapidly revolut
```

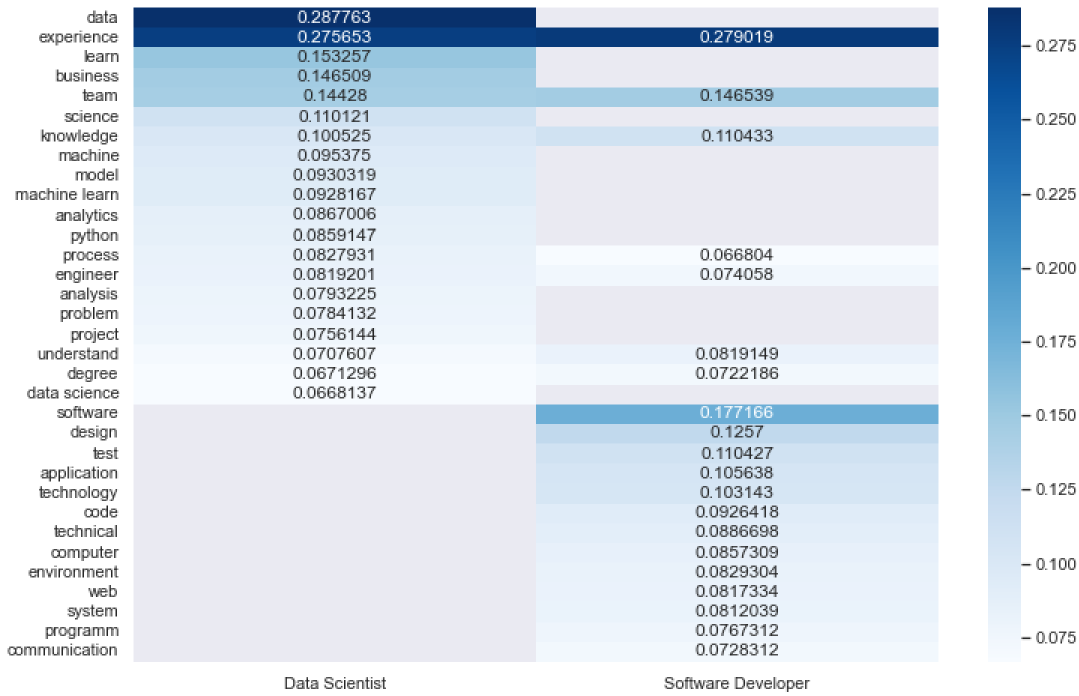
```
ionize culture based compensation math aptitude predictive experience ecos
ystem collection tech undertake vision large discover competitive propriet
ary graduate type understand relevant communication innovation highly iden
tify opportunity mind modern well day statistic access rely backed proven
north efficiency machine propose ensemble earn funded neural build quantit
ative visualization tableau market globally excellent science environment
mean unique fast valuable grow look

After cleanning using method 2:
ba challenge company organization global tensorflow us collaborative kera
solv problem knowledge capital scientist equity python computer job suppor
t growth complex every leadershipcome acumen better think analytical degre
e evolv paced intelligence field active presentation dealmaker industry al
gebra scikit business data preferred learn sql team engineer innovative ra
pidly revolutionize culture based compensation math aptitude experience bs
c champion competitive graduate understand relevant communication model op
portunity frame nlp mind well modern day statistic backed proven net machi
ne earn funded neural quantitative tableau bert u market e excellent scien
ce environment g canadian fast grow
```

We can see that the first preprocessed text had more general information like “collaborate”, “analytical”, “communication” which are required by most of the industries. The second cleaned text, however, revealed more specific skills that a data scientist should have, such as “tensorflow”, “kera”, “scikit”, “sql”, “nlp”, “bert”… They both contained “python”, “math”, “computer”, so these can be considered as the fundamental skills for both jobs.

Then the Bhattacharyya coefficient was calculated as 0. 8261 which indicated a large overlap between the two jobs. This makes sense since they are all computer science related jobs and have a lot of required skills in common.

The following heatmap further explores the overlapping features between the two jobs. The most important feature of software developer jobs, “experience”, is also the second important feature of data scientist jobs. This together with other 6 common features result in a high Bhattacharyya coefficient.


data
experience
learn
business
team
science
knowledge
machine
model
machine learn
analytics
python
process
engineer
analysis
problem
project
understand
degree
data science
software
design
test
application
technology
code
technical
computer
environment
web
system
programm
communication
0.287763
0.275653
0.279019
0.153257
0.146509
0.14428
0.146539
0.110121
0.100525
0.110433
0.095375
0.0930319
0.0928167
0.0867006
0.0859147
0.0827931
0.066804
0.0819201
0.074058
0.0793225
0.0784132
0.0756144
0.0707607
0.0819149
0.0671296
0.0722186
0.0668137
0.177166
0.1257
0.110427
0.105638
0.103143
0.0926418
0.0886698
0.0857309
0.0829304
0.0817334
0.0812039
0.0767312
0.0728312
Data Scientist
Software Developer
0.275
0.250
0.225
0.200
0.175
0.150
0.125
0.100
0.075