ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
KHOA CÔNG NGHỆ THÔNG TIN

NHÓM 3

BÁO CÁO ĐỒ ÁN MÔN LẬP TRÌNH KHOA HỌC DỮ LIỆU – THỰC HÀNH

NGÀNH: KHOA HỌC DỮ LIỆU

Thành phố Hồ Chí Minh – 2022

ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
KHOA CÔNG NGHỆ THÔNG TIN

NHÓM 3

BÁO CÁO ĐỒ ÁN MÔN LẬP TRÌNH KHOA HỌC DỮ LIỆU – THỰC HÀNH

| Đề tài |

THỰC HIỆN MỘT QUY TRÌNH KHOA HỌC DỮ LIỆU VỚI TẬP DỮ LIỆU VỀ VIỆC LÀM IT Ở VIỆT NAM

NGÀNH: KHOA HỌC DỮ LIỆU

Thành phố Hồ Chí Minh - 2020

CÁC THÀNH VIÊN TRONG NHÓM:

- PHẠM PHÚ HOÀNG SƠN – 20120366
- HÀ XUÂN TRƯỜNG - 20120391
- NGUYỄN HOÀNG VIỆT – 20120402
- TRẦN MINH QUANG – 20120559

Mục lục

1. Giới thiệu sơ bộ về đồ án

Đồ án lần này được thực hiện để nâng cao khả năng thực hiện một quy trình khoa học dữ liệu. Đồ án tổng hợp lại tất cả kiến thức và kĩ năng đã được học ở môn Lập trình khoa học dữ liệu giúp cho sinh viên có thể ôn tập và cũng cố kiến thức cũng như nâng cao khả năng của bản thân.

2. Giới thiệu về thu thập dữ liệu

Tập dữ liệu của đồ án lần này được lấy trên kaggle gồm có 2 file là jobs.csv và companies.csv, vì là tập dữ liệu thu thập về việc làm trên website itviec.com nên tập dữ liệu này vẫn còn là tập dữ liệu chưa được cấp phép, tuy nhiên chúng ta chỉ sử dụng dữ liệu này cho việc thực hiện một đồ án trong môi trường đại học và không sử dụng để cho mục đích kiếm tiền nên nhóm chúng em sẽ sử dụng tập dữ liệu này cho đồ án.

Tập dữ liệu được thu thập bằng cách sử dụng thư viện scrapy để cào dữ liệu tại trang tìm kiếm việc làm của website itviec.com sau đó lưu vào 2 file csv và được đăng tải lên kaggle.

Dữ liệu của 2 file sẽ được lưu vào trong 2 dataframe là df_job và df_companies.

Nhóm chúng em đã tải tập dữ liệu này về trên kaggle và sử dụng nó.

3. Khám phá dữ liệu và tiền xử lí dữ liệu

1. Dataframe df_job

Dữ liệu có 7 cột và 1412 dòng

```
[3]
#   YOUR CODE HEAR
num_rows = df_job.shape[0]
num_cols = df_job.shape[1]
print(num_cols)
print(num_rows)
```

```
7
1412
```

Mỗi dòng ở đây thể hiện cho một việc làm mà nhà tuyển dụng đặt ra.

Các cột thể hiện những mô tả về công việc của nhà tuyển dụng:

- job_id: Loại công việc

- copany_id: Loại công ti
- job_name: Tên công việc
- taglist: Danh sách từ khóa liên quan đến công việc
- location: Địa điểm làm việc
- three_reasons: Lợi ích khi làm việc tại đây
- description: Mô tả thêm về công việc

Các kiểu dữ liệu trong tập này hoàn toàn là dạng chuỗi, một số chỗ là NaN nên sẽ được gán là kiểu float trong pandas. Áp dụng phương thức type() để kiểm tra từng ô dữ liệu trong một cột rồi lưu lại vào một set()

```
# YOUR CODE HERE
def open_object_dtype(s):
    dtypes = set()
    s.apply(lambda x : dtypes.add(type(x)))
    return dtypes

for i in df_job.columns:
    print(i," ", open_object_dtype(df_job[i]))
```

```
job_id   {<class 'str'>}
company_id   {<class 'str'>}
job_name   {<class 'str'>}
taglist   {<class 'str'>}
location   {<class 'str'>}
three_reasons   {<class 'str'>, <class 'float'>}
description   {<class 'str'>}
```

Chúng ta có 2 vấn đề cần xử lí với kiểu dữ liệu này:

- Thay thế các NaN sang dạng chuỗi rỗng
- Vì description là dạng chuỗi được phân cách bởi các kí tự xuống dòng và tab nên ta lưu từng giá trị riêng vào một list để dễ sử dụng

Đổi giá trị NaN sang chuỗi rỗng.

```
df_job['three_reasons'] = df_job['three_reasons'].replace([np.nan],'')
```

Lưu giá trị của cột description vào list.

+ Code + Markdown

```
if type(df_job['description'].at[0]) != list:
    df_job['description'] = df_job['description'].replace(r'\n','',regex=True)

    df_job['description'] = df_job['description'].apply(lambda x: x.split('\r'))

    df_job['description'] = df_job['description'].apply(lambda x: list(filter(None, x)))
```

Do là dữ liệu dạng chuỗi, ta tìm kiếm các đặc trưng của nó về phân bố như cách đã được học, ta kiểm tra số giá trị null của một cột, số lượng các giá trị khác nhau và các giá trị đó là gì.

```
missing_ratio = (df.isnull().sum() / df.shape[0]) * 100

num_diff_vals = df.nunique()
temp = df
temp = temp.dropna()
diff_vals = set(temp)
```

Ở đây là các đặc trưng của những cột dạng categorical

	job_id	company_id	job_name	taglist	location	three_reasons	description
missing_ratio	0.0	0.0	0.0	0.0	0.0	0.0	0.0
num_diff_vals	1411	682	1359	923	696	493	1409
diff_vals	{cmc-sai-gon-technology-solution:database_engineer_oracle_mysql, fpt-softwaresoftware_data_engi...	{aurecle-information-technology-ait, infodation-vietnam, cmc-sai-gon-technology-solution, đăng ...	{Remote Senior Devops (AWS), System Engineer (Linux, Networking), Mid-Senior Unity Developer (C#...	{ JavaScript Python .NET Fresher Accepted , Angular Agile JavaScript , DevOps Linu...	{Sala City, Floor 1, 28-32 Nguyen Co Thach Street, An Loi Dong Ward, District 2, HCM City., Dist...	{, Mức lương cạnh tranh, xứng đáng với năng lực\r\nLàm việc cùng các chuyên gia hàng đầu của tập...	{IKMS Labs is the startup incubation arm of IKMS Technology. Our team consists of technology entr...

2. Dataframe df_companies

Dữ liệu có 14 cột và 2041 dòng

```
num_rows, num_cols = df_companies.shape
print(num_cols)
print(num_rows)
```
✓ 0.4s

```
14
2041
```

Mỗi dòng ở đây thể hiện cho một việc làm mà nhà tuyển dụng đặt ra.

Các cột thể hiện những mô tả về công việc của nhà tuyển dụng:

- company_id: ID của tông ty
- company_name: Tên công ty
- average_rating: Đánh giá trung bình
- num_review: Số lượng đánh giá - city: Địa điểm làm việc
- type: Loại công ty (outsourcing hoặc product)
- num_employee: Số lượng nhân viên
- country: Đất nước
- working_day: Các ngày làm việc trong tuần
- OT: Thông tin về việc công ty có OT hay không?
- overview: Tổng quan về công ty
- expertise: Chuyên môn
- benifit: Phúc lợi khi làm việc
- logo_link: logo link

Các kiểu dữ liệu trong tập này hầu hết là dạng chuỗi, chỉ có average_rating là kiểu float trong pandas. Ngoài ra những cột còn lại có float là những ô NaN. Áp dụng phương thức type() để kiểm tra từng ô dữ liệu trong một cột rồi lưu lại vào một set()

```python
def open_object_dtype(s):
    dtypes = set()
    s.apply(lambda x : dtypes.add(type(x)))
    return dtypes

for i in df_companies.columns:
    print(i," ", open_object_dtype(df_companies[i]))
```

✓ 0.4s

```
company_id   {<class 'str'>}
company_name   {<class 'str'>}
average_rating   {<class 'float'>}
num_review   {<class 'str'>}
city   {<class 'str'>}
type   {<class 'str'>}
num_employee   {<class 'str'>}
country   {<class 'str'>}
working_day   {<class 'str'>, <class 'float'>}
OT   {<class 'str'>, <class 'float'>}
overview   {<class 'str'>, <class 'float'>}
expertise   {<class 'str'>, <class 'float'>}
benifit   {<class 'str'>, <class 'float'>}
logo_link   {<class 'str'>}
```

Chúng ta có những vấn đề cần xử lí với kiểu dữ liệu này:

- Cột ‘working_day’, `OT`, `overview`, `expertise`, `benifit` có những giá trị là float, đây có thể là những giá trị NaN nên hãy chuyển nó sang dạng chuỗi rỗng để tiện cho việc sử dụng sau này.
- Cột `num_review` đang có kiểu dữ liệu str, ta chuyển sang dạng int bằng cách chỉ lấy số lượng review
- Cột `num_employee` có dạng str ta chuyển đổi như sau:
- Các dòng có giá trị 1000+ ta đưa về 1001
- Các dòng có giá trị 501-1000 ta đưa về 10000 - Các dòng có giá trị 1-50 ta đưa về 50 - Các dòng có giá trị 51-150 ta đưa về 150 - Các dòng có giá trị 151-300 ta đưa về 300 - Các dòng có giá trị 301-500 ta đưa về 500

Ta tiến hành xử lý cột num_employee

```
df_companies.num_employee.unique()
```

✓ 0.4s

```
array(['1000+', '501-1000', '1-50', '51-150', '151-300', '301-500'],
      dtype=object)
```

```
def transform_age(age_type):
    if age_type == '1000+':
        k = 1001
    elif age_type == '501-1000':
        k = 1000
    elif age_type == '1-50':
        k = 50
    elif age_type == '51-150':
        k = 150
    elif age_type == '151-300':
        k = 300
    elif age_type == '301-500':
        k = 500
    else:
        k = 0
    return k

df_companies["num_employee"] = df_companies["num_employee"].apply(lambda x : transform_age(x))
```

✓ 0.4s

Tiếp tục xử lý dữ liệu cột num_review

```
def extract_number(x):
    k = re.findall(r'\d+', x)
    if len(k) == 0:
        return 0
    else:
        return k[0]

df_companies['num_review'] = df_companies['num_review'].apply(lambda x : extract_number(x))
```

✓ 0.4s

```
df_companies['num_review'] = df_companies.num_review.astype('int')
```

✓ 0.4s

Với các cột có kiểu dữ liệu số, ta sẽ tính:

- Tỉ lệ % (từ 0 đến 100) các giá trị thiếu
- Giá trị min
- Giá trị lower quartile (phân vị 25)
- Giá trị median (phân vị 50)
- Giá trị upper quartile (phân vị 75)
- Giá trị max Lưu kết quả vào DataFrame `num_col_info_df`, trong đó:
- Tên của các cột là tên của các cột số trong `survey_df`
- Tên của các dòng là: "missing_ratio", "min", "lower_quartile", "median", "upper_quartile", "max" Để dễ nhìn, tất cả các giá trị bạn đều làm tròn với 1 chữ số thập phân bằng phương thức `.round(1)`.

```
numerics = ['int16', 'int32', 'int64', 'float16', 'float32', 'float64']
new_df = df_companies.select_dtypes(include=numerics)

missing_ratio = (new_df.isna().sum()*100/num_rows).round(1)

missing_ratio = missing_ratio.to_frame().T

df = new_df.describe().round(1)

min = df.iloc[[3]]

lower_quartile = df.iloc[[4]]

median = df.iloc[[5]]

upper_quartile = df.iloc[[6]]

max = df.iloc[[7]]
frames = [missing_ratio, min, lower_quartile, median, upper_quartile, max]

num_col_info_df = pd.concat(frames)

num_col_info_df = num_col_info_df.rename(index={0: 'missing_ratio', '25%': 'lower_quartile', '50%': 'median', '75%':'upper_quartile'})

num_col_info_df
```

	average_rating	num_review	num_employee
missing_ratio	72.4	0.0	0.0
min	2.0	0.0	50.0
lower_quartile	3.6	0.0	50.0
median	4.0	0.0	50.0
upper_quartile	4.4	3.0	150.0
max	5.0	1486.0	1001.0

Với các cột có kiểu dữ liệu không phải số, ta sẽ tính:

- Tỉ lệ % (từ 0 đến 100) các giá trị thiếu
- Số lượng các giá trị
- Tỉ lệ % (từ 0 đến 100) của mỗi giá trị được sort theo tỉ lệ % giảm dần

Lưu kết quả vào DataFrame `cat_col_info_df`, trong đó:

- Tên của các cột là tên của các cột không phải số trong `df_companies`
- Tên của các dòng là: "missing_ratio", "num_values", "value_ratios"

Để dễ nhìn, tất cả các giá trị đều được làm tròn với 1 chữ số thập phân bằng phương thức `.round(1)`.

```
list_cat_col = ['company_id', 'company_name', 'type', 'country', 'working_day','OT']

df = df_companies[list_cat_col].copy()

missing_ratio = (df.isnull().sum()*100/num_rows).round(1)
missing_ratio = missing_ratio.to_frame().T
missing_ratio = missing_ratio.rename(index={0: 'missing_ratio'})

num_values = df.nunique()
num_values = num_values.to_frame().T
num_values = num_values.rename(index={0: 'num_values'})

value_ratios = df.apply(lambda x: dict((x.value_counts(normalize = True)*100).round(1)))
for val in value_ratios.values:
    val = dict(sorted(val.items(), key = lambda x: x[1], reverse = True))
value_ratios = value_ratios.to_frame().T
value_ratios = value_ratios.rename(index={0: 'value_ratios'})

frames = [missing_ratio, num_values, value_ratios]
cat_col_info_df = pd.concat(frames)

cat_col_info_df
```

	company_id	company_name	type	country	working_day	OT
missing_ratio	0.0	0.0	0.0	0.0	0.7	8.6
num_values	2041	2041	2	39	2	4
value_ratios	{'kms-technology': 0.0, 'viet-money-jsc': 0.0, 'toss-viva-republica-vietnam': 0.0, 'hyperion360'...	{'KMS Technology': 0.0, 'Viet Money Jsc': 0.0, 'Toss (Viva Republica Vietnam)': 0.0, 'Hyperion36...	{'Product': 75.2, 'Outsourcing': 24.8}	{'Vietnam': 58.9, 'Japan': 9.6, 'Singapore': 7.9, 'United States': 6.8, 'Korea, Republic of': 4....	{'Monday - Friday': 91.6, 'Monday - Saturday': 8.4}	{'No OT': 85.2, 'Extra salary for OT': 12.1, 'Extra days off for OT': 2.0, 'OT included in base ...

4. Đặt câu hỏi và trả lời

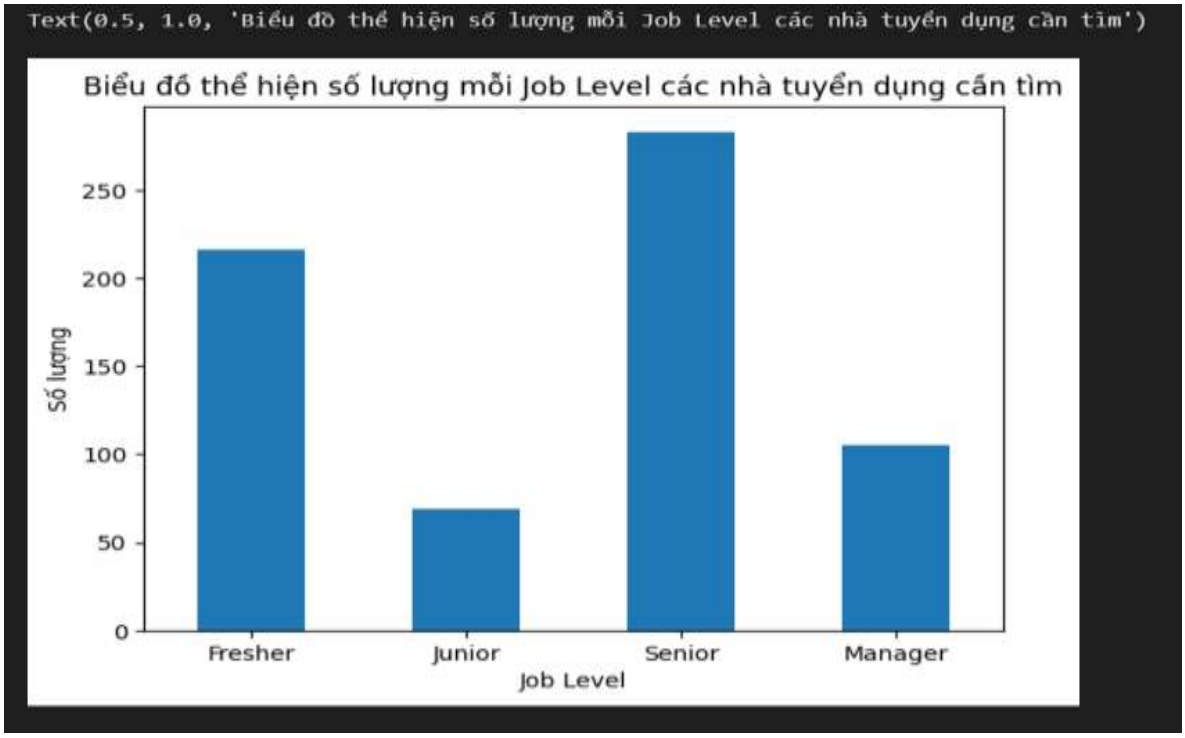
Dựa vào dữ liệu đã khám phá, chúng ta có thể đặt ra một số câu hỏi như sau:

a. **Câu hỏi 1:** Hiện nay nhà tuyển dụng tìm kiếm 4 job level chính là: Fresher, Junior, Senior, Manager. Vậy hãy đếm số lượng của từng loại job level trên và vẽ một biểu đồ cột giúp chúng ta so sánh.

*Tiền xử lý để trả lời câu hỏi: Do các job level không được liệt kê thành các thuộc tính riêng biệt mà chỉ được đính kèm theo tên công việc và chức vụ cần tuyển dụng của các công ty nên ta phải lọc ra các job level từ cột job_name cũng như cột taglist. Kết quả sau khi lọc:

```
Fresher    216
Junior      69
Senior     283
Manager    105
dtype: int64
```

Từ kết quả trên ta trực quan bằng biểu đồ cột:



*Trả lời: Các nhà tuyển dụng cần level Senior nhiều nhất và Junior là ít nhất. Bên cạnh đó, số lượng công ti cần level Junior cũng khá cao.

*Ý nghĩa của câu hỏi: Xác định được những đối tượng nhân lực mà các công ti IT đang có xu hướng tuyển dụng.

b. **Câu hỏi 2:** Tìm ra 10 công nghệ/framework/ngôn ngữ lập trình được nhiều công ti tuyển dụng đề ra và săn đón nhiều nhất.

*Tiền xử lý để trả lời câu hỏi: Trước tiên hãy xem danh sách các tag được tuyển dụng trong df_job và số lượng công ti tuyển dụng của mỗi tag.

```
QA QC              169
English            152
Tester             169
Project Manager     54
Agile               85
                   ...
Games               24
Bridge Engineer      1
Magento              7
Japanese             2
Cocos                1
Length: 79, dtype: int64
```

Có rất nhiều tag không đề cập đến công nghệ mà chỉ là vị trí tuyển dụng, yêu cầu về ngôn ngữ,... Chúng ta cần phải lọc bớt đi những tag này. Sau khi lọc, chúng ta sẽ còn lại những tag liên quan đến công nghệ như sau:

```
Index(['Agile', 'NodeJS', 'React Native', 'Java', 'JavaScript', 'Spring',
       'ReactJS', 'Ruby on Rails', 'AWS', 'Database', 'SQL', '.NET', 'C#',
       'ASP.NET', 'Scrum', 'Python', 'C++', 'Linux', 'Embedded', 'C language',
       'OOP', 'Django', 'JQuery', 'Cloud', 'Kotlin', 'Blockchain', 'Android',
       'iOS', 'DevOps', 'Swift', 'ERP', 'Oracle', 'System Engineer',
       'Networking', 'MySQL', 'Scala', 'PHP', 'Laravel', 'Angular', 'VueJS',
       'Sharepoint', 'UI-UX', 'PostgreSql', 'IT Support', 'Objective C',
       'Golang', 'Xamarin', 'CSS', 'HTML5', 'SAP', 'Flutter', 'NoSQL', 'JSON',
       'MVC', 'J2EE', 'Ruby', 'Unity', 'Drupal', 'Wordpress', 'AngularJS',
       'Hybrid', 'Games', 'Bridge Engineer', 'Magento', 'Cocos'],
      dtype='object')
```

*Trả lời: 10 công nghệ được tuyển dụng nhiều nhất.

```
tech_sr = tech_sr.sort_values(ascending=False)
tech_sr.head(10)
```

```
Java          265
JavaScript    244
ReactJS       164
.NET          144
Python        140
SQL           133
NodeJS        129
PHP           125
C#            108
MySQL         102
dtype: int64
```

*Ý nghĩa của việc trả lời câu hỏi: tìm ra các công nghệ/ngôn ngữ lập trình/framework được tuyển dụng nhiều nhất, có xu hướng phát triển trong tương lai để chuẩn bị các kiến thức về những công nghệ này để "bắt kịp" xu hướng.

c. **Câu hỏi 3:** Hãy tìm những quận huyện ở HCM và HN trong df_job, sau đó lưu vào 2 dict là *hcm* và *hn* với keys là tên huyện, values là số lượng việc làm tuyển dụng ở quận huyện đó. Sau đó vẽ một barchart để thể hiện dict này.

*Tiền xử lý để trả lời câu hỏi: Hiển thị các địa chỉ thuộc HCM vào dict *hcm*:

```
{'District 3': 94,
 'Tan Binh': 199,
 'District 4': 42,
 'District 1': 214,
 'Binh Thanh': 101,
 'District 2': 40,
 'District 7': 62,
 'District 10': 68,
 'District 9': 10,
 'Phu Nhuan': 58,
 'Thu Duc': 11,
 'Other': 18,
 'District 5': 20,
 'District 12': 26,
 'Thu Duc City': 14,
 'District 11': 11,
 'Tan Phu': 7,
 'Go Vap': 7,
 'District 8': 1,
 'NA': 1,
 'HCM': 1,
 'Ho Chi Minh City': 1}
```

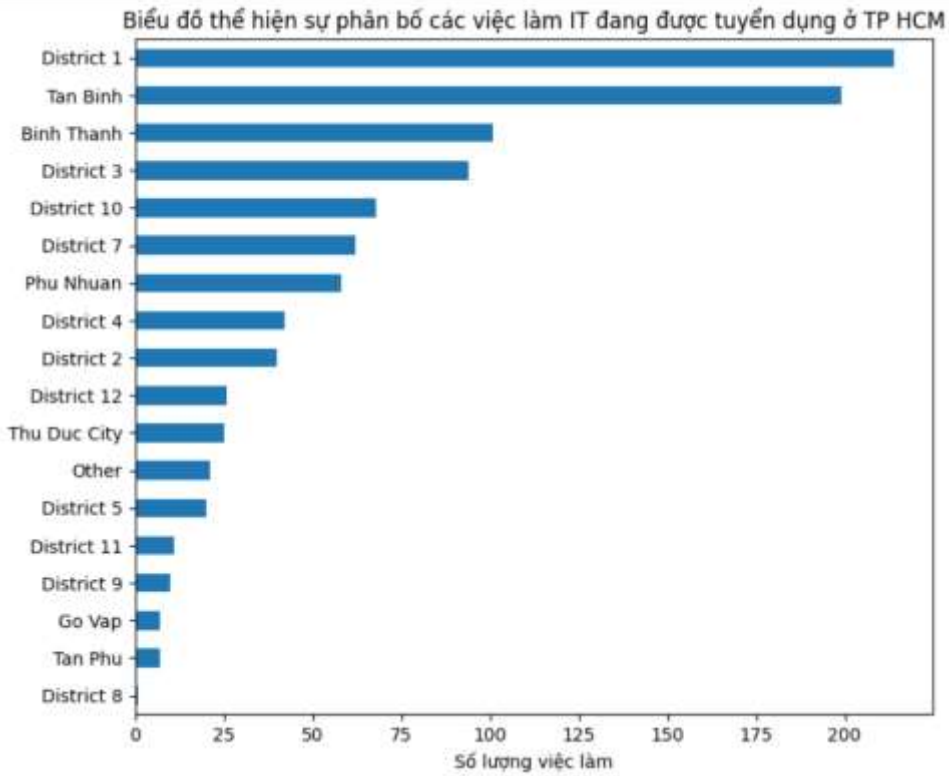
Có các dữ liệu chưa "hợp lí" (có vẻ như các công ti đó đã nhập thông tin chưa đầy đủ hoặc không chính xác) -> gom vào Other:

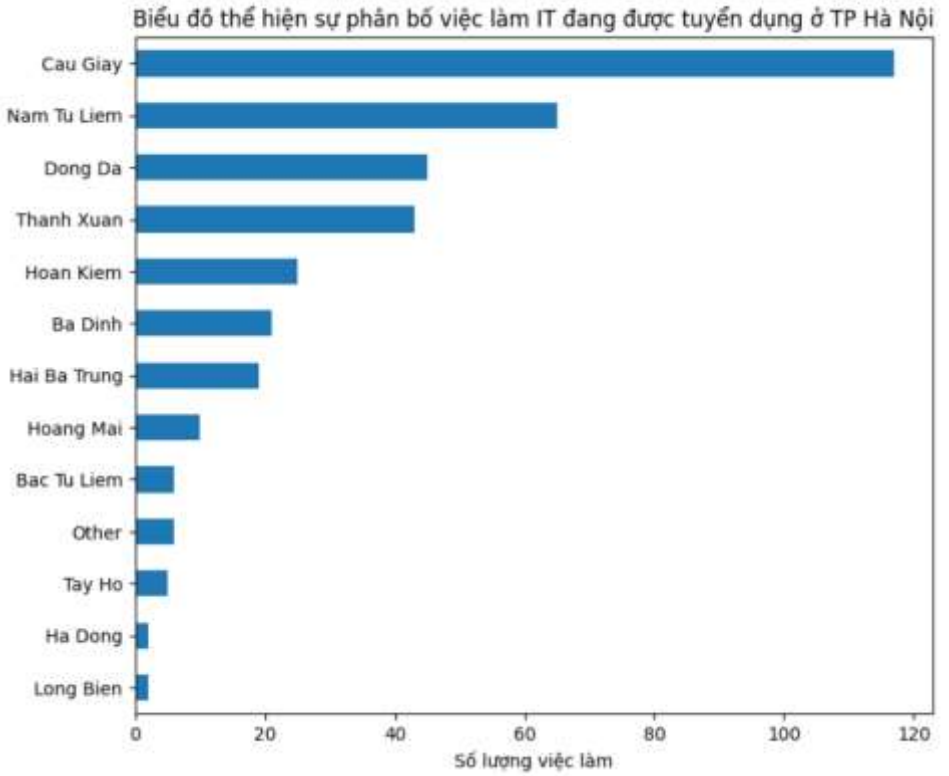
```
{'District 3': 94,
 'Tan Binh': 199,
 'District 4': 42,
 'District 1': 214,
 'Binh Thanh': 101,
 'District 2': 40,
 'District 7': 62,
 'District 10': 68,
 'District 9': 10,
 'Phu Nhuan': 58,
 'Other': 21,
 'District 5': 20,
 'District 12': 26,
 'Thu Duc City': 25,
 'District 11': 11,
 'Tan Phu': 7,
 'Go Vap': 7,
 'District 8': 1}
```

Tương tự với dict *hn*:

```
{'Dong Da': 45,
 'Other': 6,
 'Hoan Kiem': 25,
 'Cau Giay': 117,
 'Nam Tu Liem': 65,
 'Thanh Xuan': 43,
 'Ba Dinh': 21,
 'Hai Ba Trung': 19,
 'Long Bien': 2,
 'Bac Tu Liem': 6,
 'Hoang Mai': 10,
 'Tay Ho': 5,
 'Ha Dong': 2}
```

Trực quan:


Biểu đồ thể hiện sự phân bố các việc làm IT đang được tuyển dụng ở TP HCM
District 1
Tan Binh
Binh Thanh
District 3
District 10
District 7
Phu Nhuan
District 4
District 2
District 12
Thu Duc City
Other
District 5
District 11
District 9
Go Vap
Tan Phu
District 8
0
25
50
75
100
125
150
175
200
Số lượng việc làm


Biểu đồ thể hiện sự phân bố việc làm IT đang được tuyển dụng ở TP Hà Nội
Cau Giay
Nam Tu Liem
Dong Da
Thanh Xuan
Hoan Kiem
Ba Dinh
Hai Ba Trung
Hoang Mai
Bac Tu Liem
Other
Tay Ho
Ha Dong
Long Bien
0
20
40
60
80
100
120
Số lượng việc làm

*Trả lời:

- TPHCM: Quận 1 (District 1) và quận Tân Bình là 2 quận có nhiều việc làm IT đang được tuyển dụng nhất.
- Hà Nội: Quận Cầu Giấy là quận có nhiều việc làm IT đang được tuyển dụng nhất.

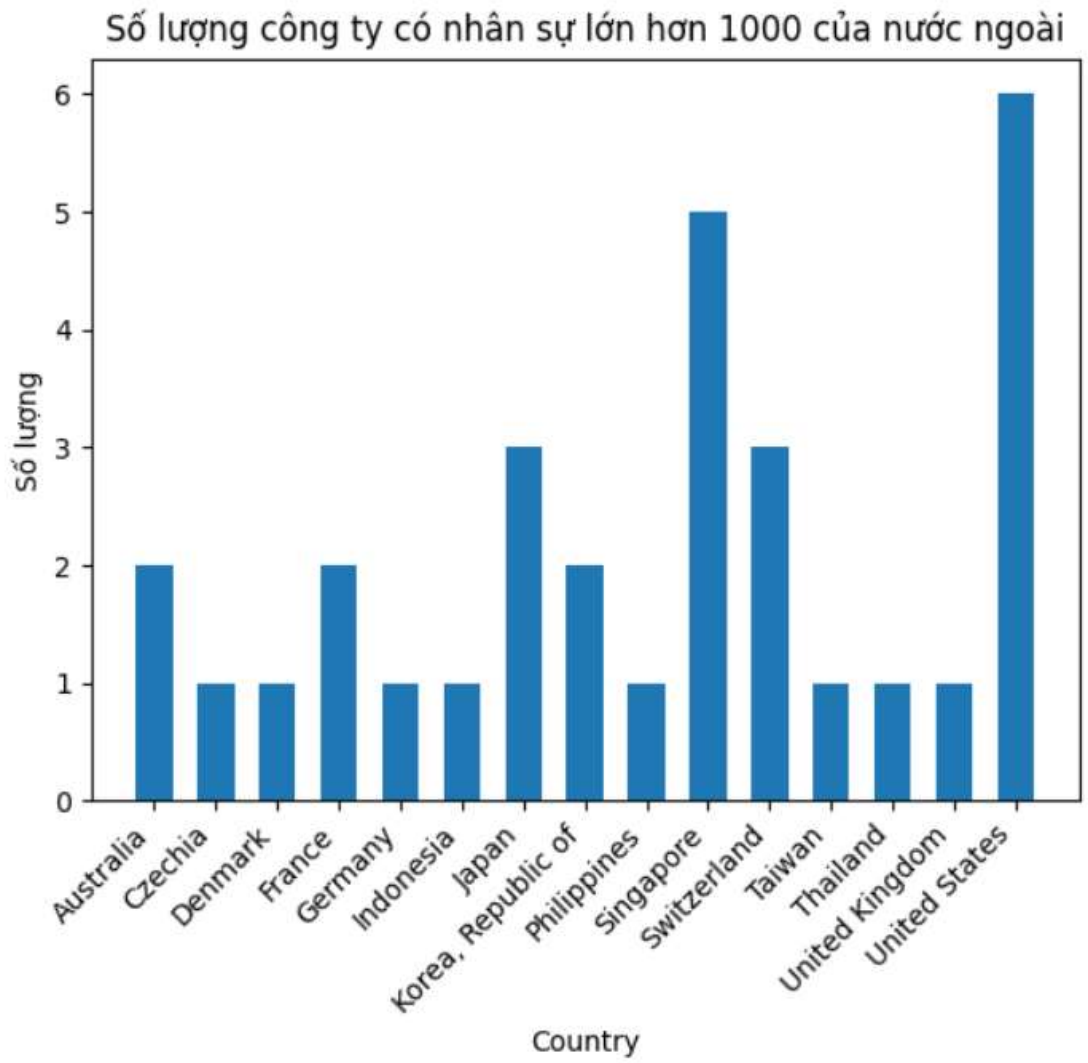
*Ý nghĩa của việc trả lời câu hỏi: phục vụ cho việc tra cứu việc làm theo vị trí địa lí.

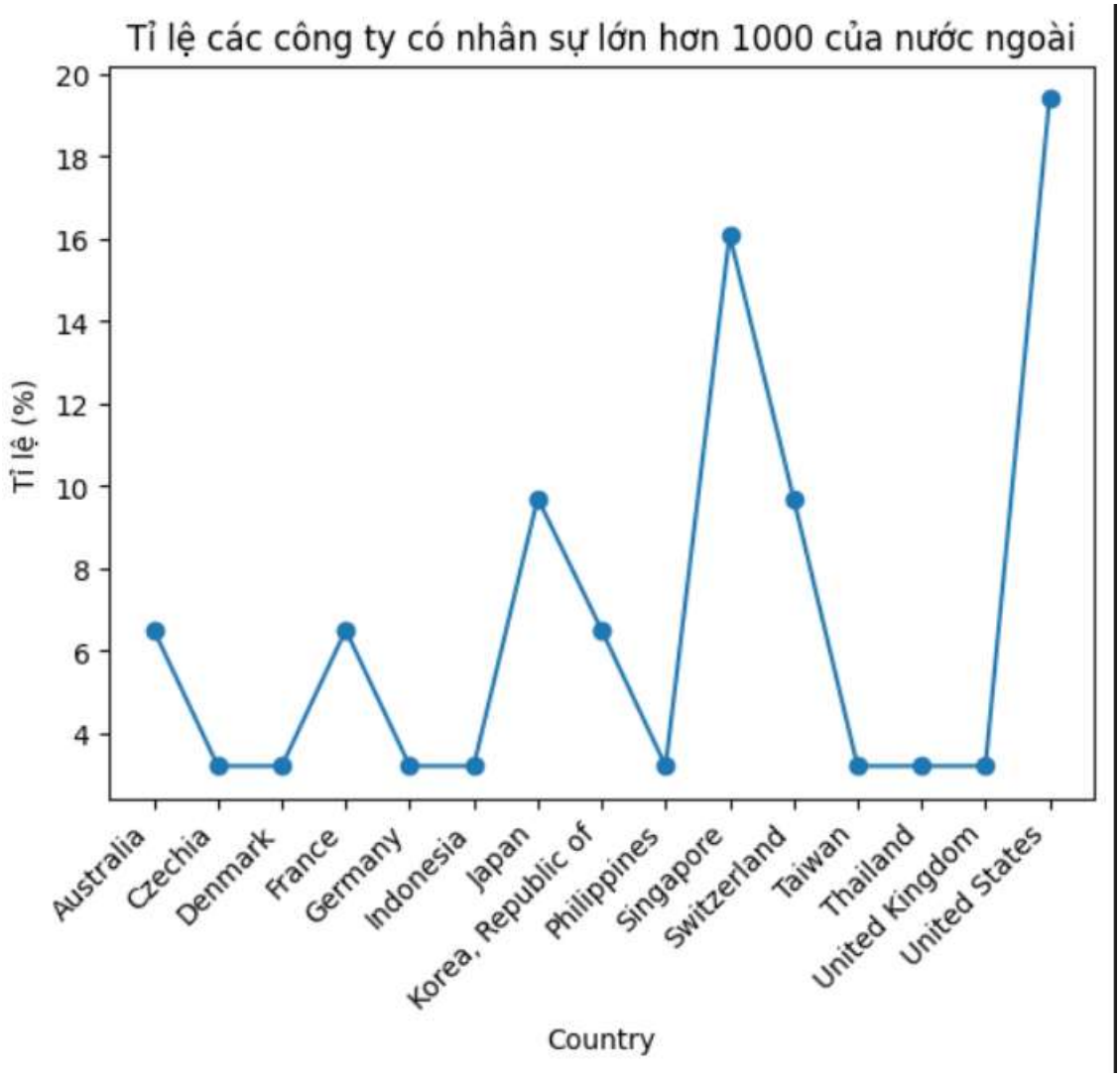
d. **Câu hỏi 4**: Những công ty nước ngoài có số nhân viên trên 1000 thuộc những nước nào, số lượng, tỉ lệ ra sao và tỉ lệ Outsourcing cho từng nước?

* Tiền xử lí để trả lời câu hỏi: lưu các thông tin vào một dataframe country_df

country	num_companies	ratio_num_companies	ratio_oursourcing
Australia	2	6.5	50.0
Czechia	1	3.2	0.0
Denmark	1	3.2	100.0
France	2	6.5	50.0
Germany	1	3.2	0.0
Indonesia	1	3.2	0.0
Japan	3	9.7	33.3
Korea, Republic of	2	6.5	0.0
Philippines	1	3.2	0.0
Singapore	5	16.1	20.0
Switzerland	3	9.7	33.3
Taiwan	1	3.2	0.0
Thailand	1	3.2	0.0
United Kingdom	1	3.2	100.0
United States	6	19.4	66.7

Trực quan:


Số lượng công ty có nhân sự lớn hơn 1000 của nước ngoài
Số lượng
0
1
2
3
4
5
6
Australia
Czechia
Denmark
France
Germany
Indonesia
Japan
Korea, Republic of
Philippines
Singapore
Switzerland
Taiwan
Thailand
United Kingdom
United States
Country



*Trả lời

- Có 15 quốc gia có tổng số lượng công ti lớn hơn 1000 nhân sự ở tập dữ liệu này
- Trong số đó Mỹ và Singapore là 2 quốc gia có số lượng công ti góp mặt là lớn nhất, và đây cũng là 2 thị trường tiềm năng nhất về mặt tuyển dụng nhân sự.

*Ý nghĩa: cho chúng ta nắm được số lượng những công ty có thể xem là bigtech của một quốc gia là bao nhiêu, và cũng cho ta nắm được xem thị trường tuyển dụng nào là đông đảo nhất với một thị trường outsourecing như Việt Nam.