Nhóm 15

Báo cáo đồ án thực hành NMKHDL

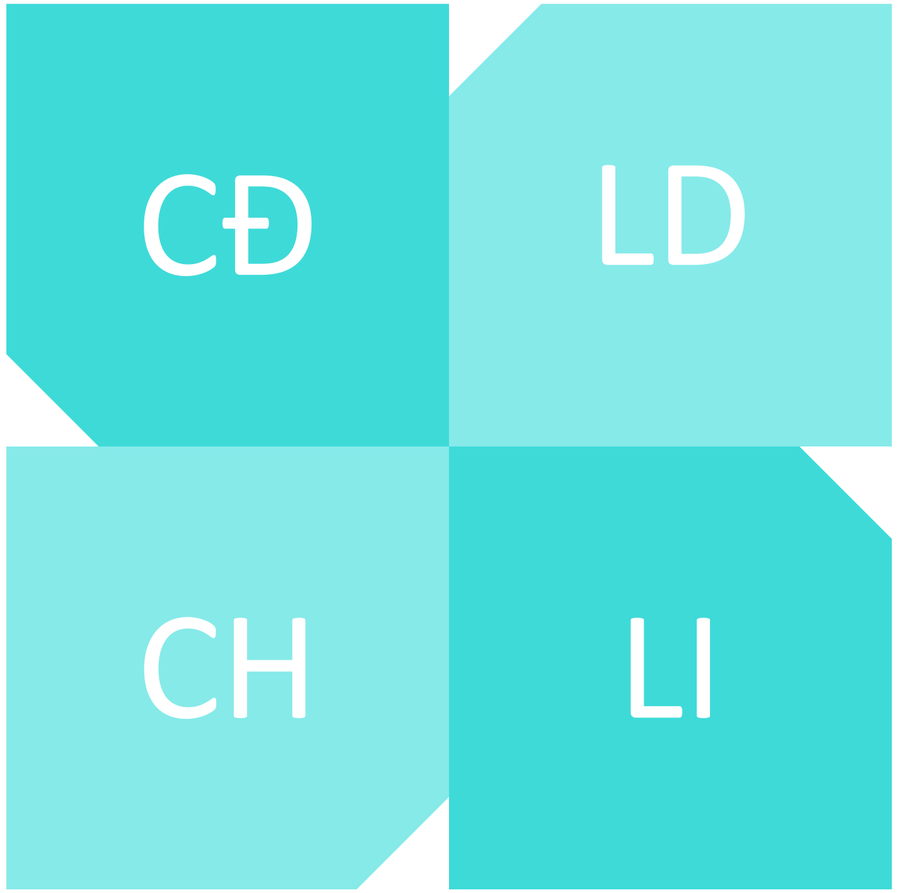
01	18120278	Phạm Hoàng Nam Anh
02	20120146	Nguyễn Thị Châu Ngọc
03	20120549	Lê Hoàng Phúc
04	20120587	Nguyễn Hoàng Thịnh

Mục lục

Giới thiệu đồ án

Chủ đề:

Phân tích thị trường lao động ở Việt Nam.



Link Dataset:

https://www.vietnamworks.com/tim-viec-lam/tat-ca-viec-lam.

Câu hỏi:

Dự đoán số lượng ứng viên đăng ký một công việc dựa trên các đặc trưng của công việc?

Lợi ích

Đem lại các thông tin cần thiết cho những người muốn tìm kiếm, tìm hiểu về việc làm.

01. Thu thập, tiền xử lý dữ liệu

Quy trình thu thập, tiền xử lý dữ liệu

Thu thập dữ liệu

Cách tiếp cận sử dụng kĩ thuật crawling:
- robots.txt.
- Lazy-loading.

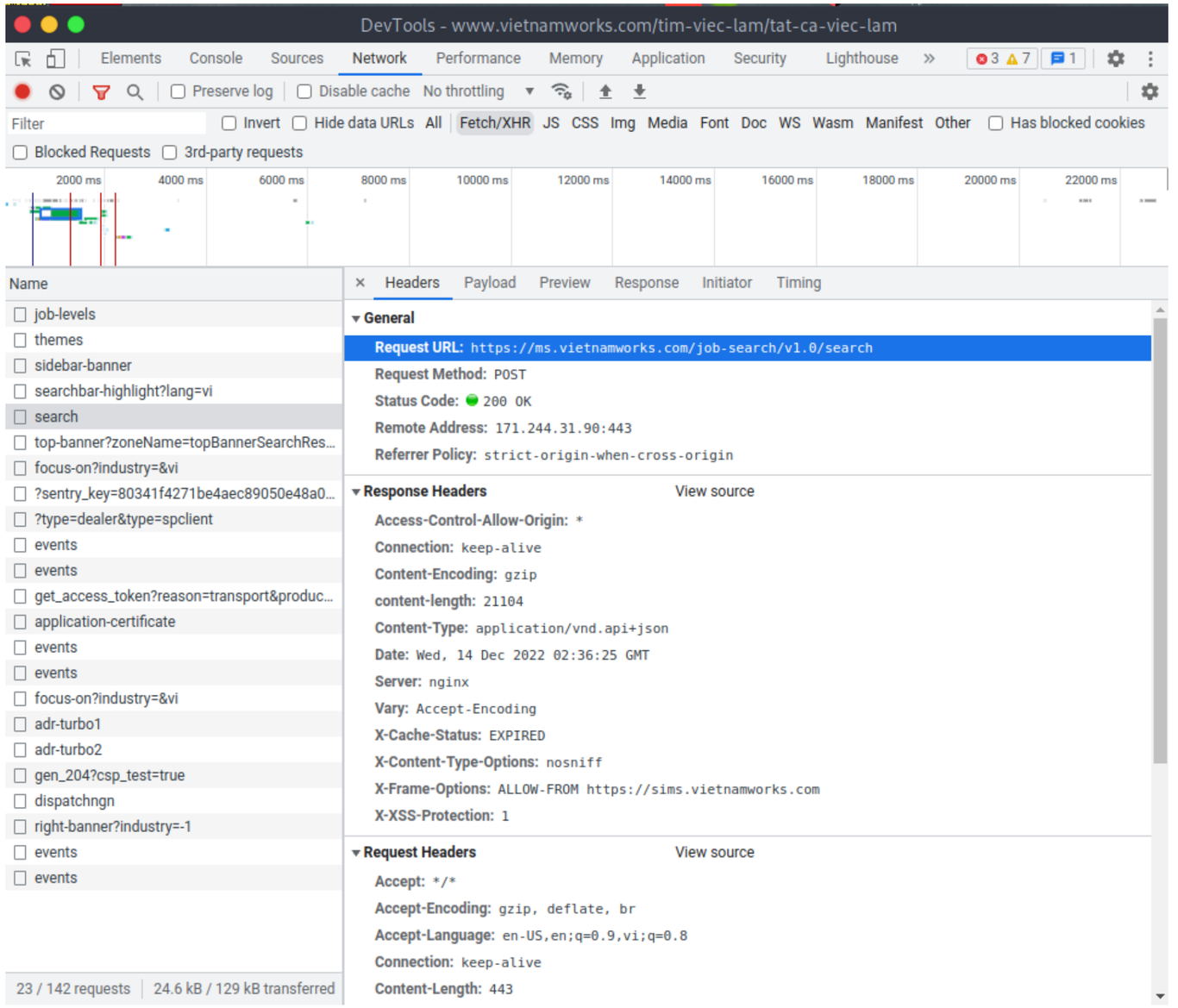
```
User-agent: *

Disallow: /my-profile
Disallow: /my-profile/
Disallow: /ho-so
Disallow: /ho-so/

Disallow: /my-career-center
Disallow: /my-career-center/
Disallow: /quan-ly-nghe-nghiep
Disallow: /quan-ly-nghe-nghiep/

Disallow: /dang-nhap/?*
Disallow: /login/?*

# Block all robots from restricted areas
Disallow: /jobseekers/apply_online.php?*
Disallow: /viec-lam/nop-ho-so-truc-tuyen/
Disallow: /jobs/apply-job-online/
Disallow: /jobseekers/apply_on_oneclick.php

# Block all robots from internal actions
Disallow: /jobseekers/jobdetail_print.php?*
Disallow: /jobseekers/open_authenticate.php?*
Disallow: /jobseekers/checkAuthenticate.php*
Disallow: /company/preview/*

# Block all robots from AJAX actions
Disallow: /jobseekers/ajax.php?*

# Block all robots from advertisement
Disallow: /vclick/index.php?*
Disallow: /wow-cv/render/

Sitemap: https://www.vietnamworks.com/sitemap/sitemap.xml
```

Thu thập dữ liệu

Sử dụng HTTP Request:

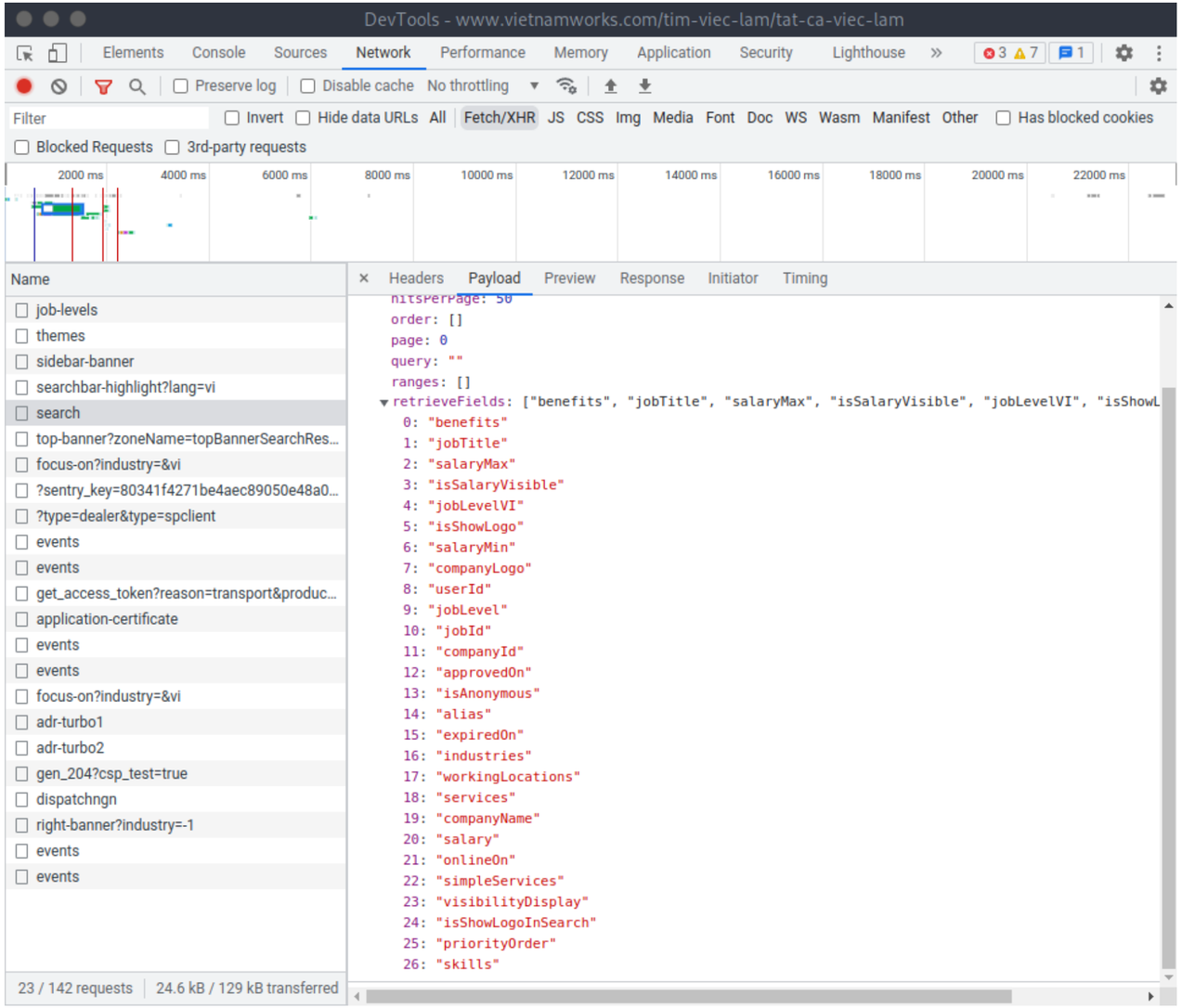
Dùng DevTools để inspect trang web, dò các http request đến server của trang web để tìm API.

Trang không yêu cầu các phương thức bảo mật như access token hay user login.



Tiền xử lý dữ liệu

API cho phép truy cập nhiều thông tin không cần thiết nên việc tiếp theo là ta cần loại bớt những thông tin thừa như jobUrl, contactName,... và format lại dữ liệu về đúng kiểu dữ liệu có thể sử dụng



Số dòng và cột của dữ liệu

```
num_rows = data_df.shape[0]
num_cols = data_df.shape[1]
print(num_rows)
print(num_cols)
```

```
8216
36
```

Tiền xử lý dữ liệu

Mã hóa dữ liệu cột companySize.

- "Less Than 10" → (0,10)
- "Over 50,000" → (50000,100000)
- Từ "10-24" → (10,24)
- Từ "25-99" → (25,99)
- Từ "100-499" → (100,499)
- Từ "500-999" → (500,999)
- Từ "1,000-4,999" → (1000,4999)
- Từ "5,000-9,999" → (5000,9999)
- Từ "10,000-19,999" → (10000,19999)
- Từ "20,000-49,999" → (20000,49999)
- Không có dữ liệu → NaN

Tiền xử lý dữ liệu

Tìm hiểu sự phân bố giá trị với dữ liệu dạng numeric và tìm ra các giá trị không hợp lệ.

Số lượng các giá trị không hợp lệ là khá ít.
=> Xóa các giá trị này không làm ảnh hưởng đến tổng thể dữ liệu.

```
: print(count_invalid_salary(data_df))
  print(count_invalid_salaryMin(data_df))
  print(count_invalid_salaryMax(data_df))
```

```
0
133
133
```

Tiền xử lý dữ liệu

Tìm hiểu sự phân bố giá trị với dữ liệu dạng categorical: typeWorkingId, skills, benefits, workingLocations, industries.

Tiền xử lý dữ liệu

Thêm dữ liệu typeWorkingName.

- Với giá trị là 1 tại cột "typeWorkingId", ta có giá trị tương ứng tại cột "typeWorkingName" là **Full-time**
- Với giá trị là 2 tại cột "typeWorkingId", ta có giá trị tương ứng tại cột "typeWorkingName" là **Part-time**
- Với giá trị là 3 tại cột "typeWorkingId", ta có giá trị tương ứng tại cột "typeWorkingName" là **Internship**
- Với giá trị là 4 tại cột "typeWorkingId", ta có giá trị tương ứng tại cột "typeWorkingName" là **Online jobs**
- Với giá trị là 5 tại cột "typeWorkingId", ta có giá trị tương ứng tại cột "typeWorkingName" là **Freelancer**
- Với giá trị là 6 tại cột "typeWorkingId", ta có giá trị tương ứng tại cột "typeWorkingName" là **Seasonal**
- Với giá trị là 7 tại cột "typeWorkingId", ta có giá trị tương ứng tại cột "typeWorkingName" là **Other**

Tiền xử lý dữ liệu

Ở các cột skills, benefits, workingLocations, industries, thực hiện việc chọn giá trị điển hình cho mỗi phần tử trong mảng để đưa ra kết quả dễ quan sát hơn.

02. Trực quan hóa dữ liệu

Quy trình trực quan hóa dữ liệu

Trực quan hóa dữ liệu

01 Các câu hỏi xung quanh về vấn đề tuyển dụng

02 Tìm hiểu về những top 20 thú vị

03 Nơi tập trung nhiều công việc và thu nhập bình quân

04 Lĩnh vực có lượt đăng tải công việc và số lượng ứng viên đăng kí nhiều nhất

05 Top những kĩ năng cần thiết cho các công việc hot hiện tại

Trực quan hóa dữ liệu

01. Các câu hỏi xung quanh về vấn đề tuyển dụng

01 Hình thức làm việc nào được ưa chuộng trong các công việc tuyển dụng.

02 Các công việc thường cung cấp những phúc lợi nào cho người ứng tuyển.

03 Các công ty tuyển dụng thường có quy mô như thế nào.

04 Các công việc tuyển dụng thường yêu cầu với cấp bậc nào.

05 Các công việc tuyển dụng thường có mức lương dao động như thế nào.

Trực quan hóa dữ liệu

01. Các câu hỏi xung quanh về vấn đề tuyển dụng

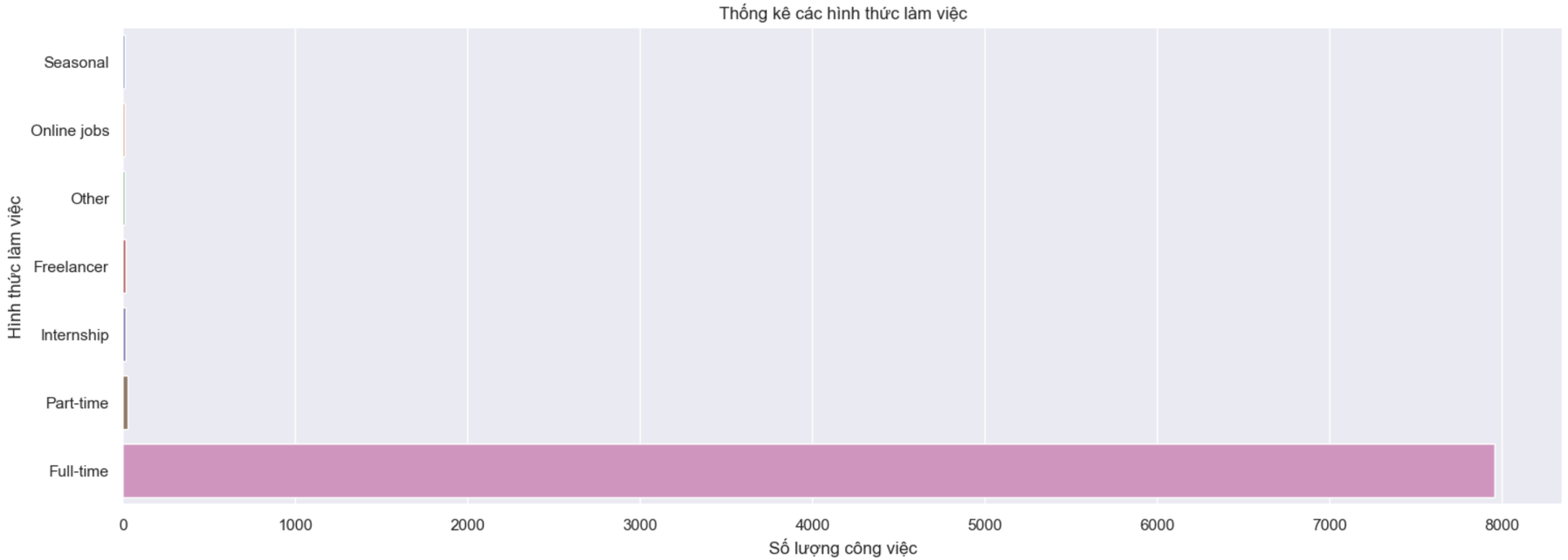
Lợi ích khi trả lời câu hỏi?

- Có được những thông tin về loại hình công việc, quy mô các công ty, lợi ích có được, cấp bậc tuyển dụng, mức lương cho các công việc để từ đó có được sự lựa chọn công việc phù hợp với nhu cầu, khả năng của bản thân.

- Đồng thời có những hiểu biết khái quát về cơ cấu, xu hướng của thị trường việc làm hiện nay từ đó có được những định hướng cho công việc tương lai.

Trực quan hóa dữ liệu

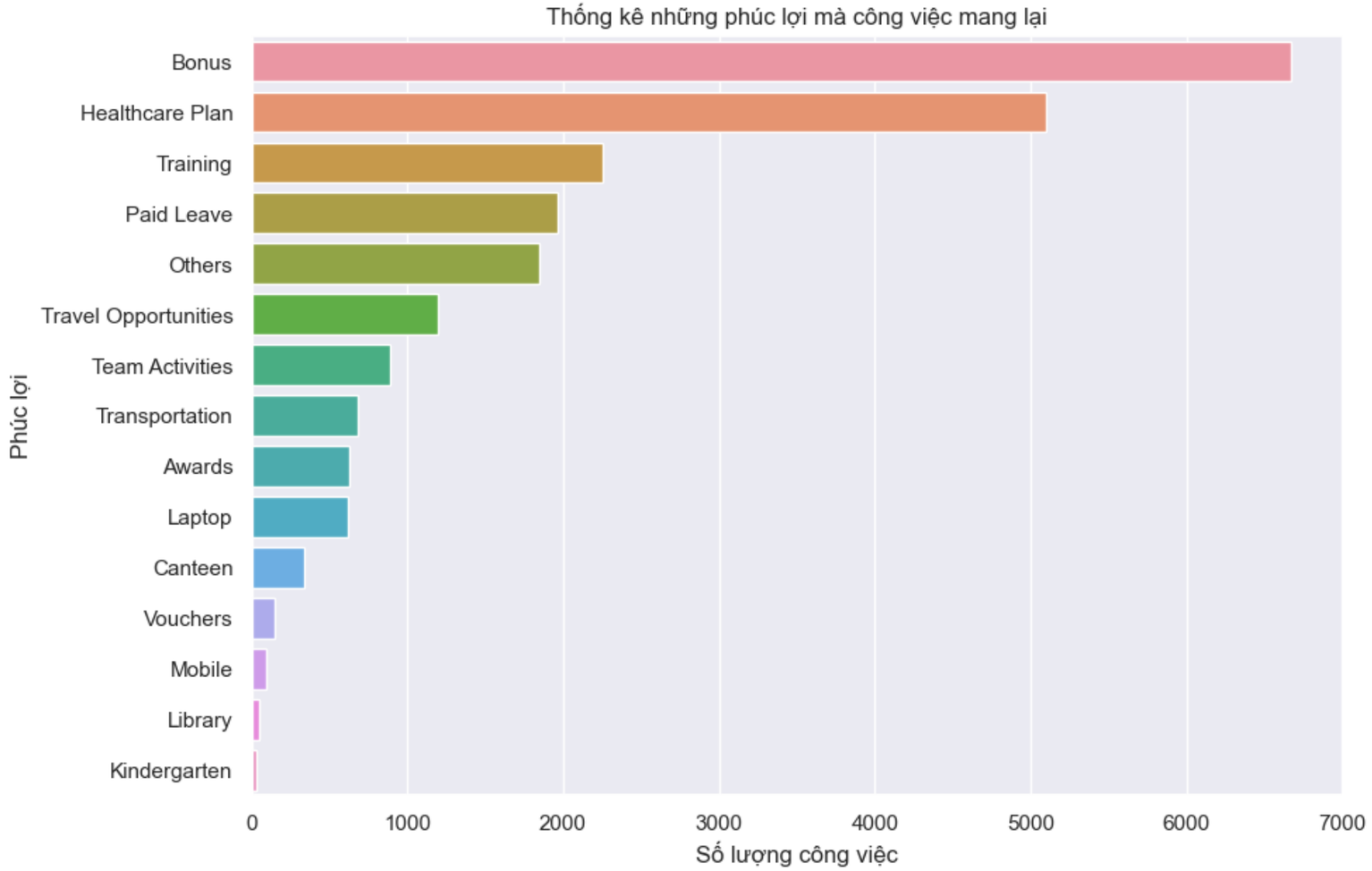
Hình thức làm việc nào được ưa chuộng trong các công việc tuyển dụng.



- Các công việc tuyển dụng thường ưa chuộng các công việc Full-time và có nhu cầu ít hơn với những hình thức làm việc còn lại.

Trực quan hóa dữ liệu

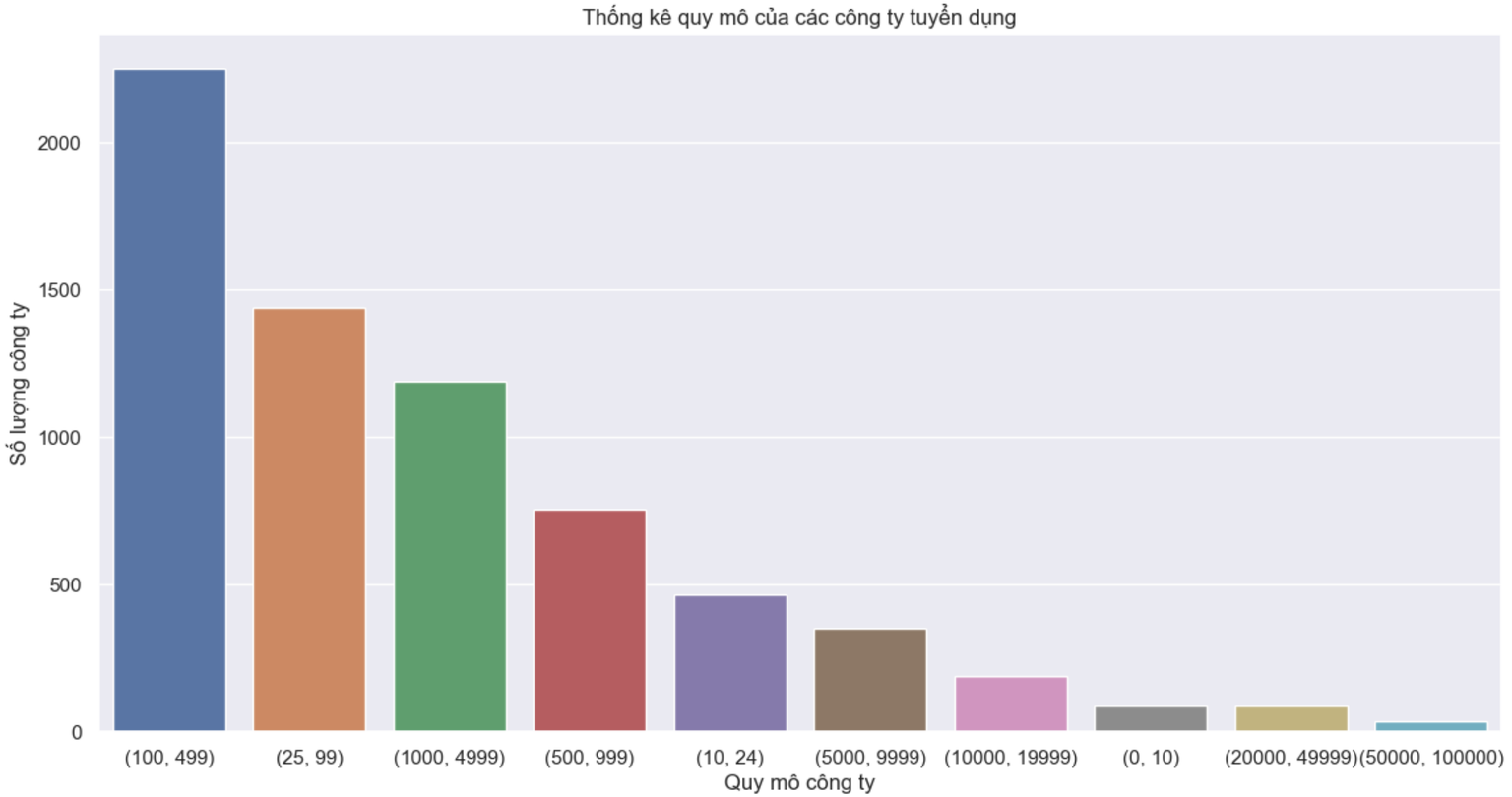
Các công việc thường cung cấp phúc lợi gì cho người ứng tuyển



- Các công việc tuyển dụng thường cung cấp bonus cho người ứng tuyển, bên cạnh đó là Healthcare Plan, Training..

Trực quan hóa dữ liệu

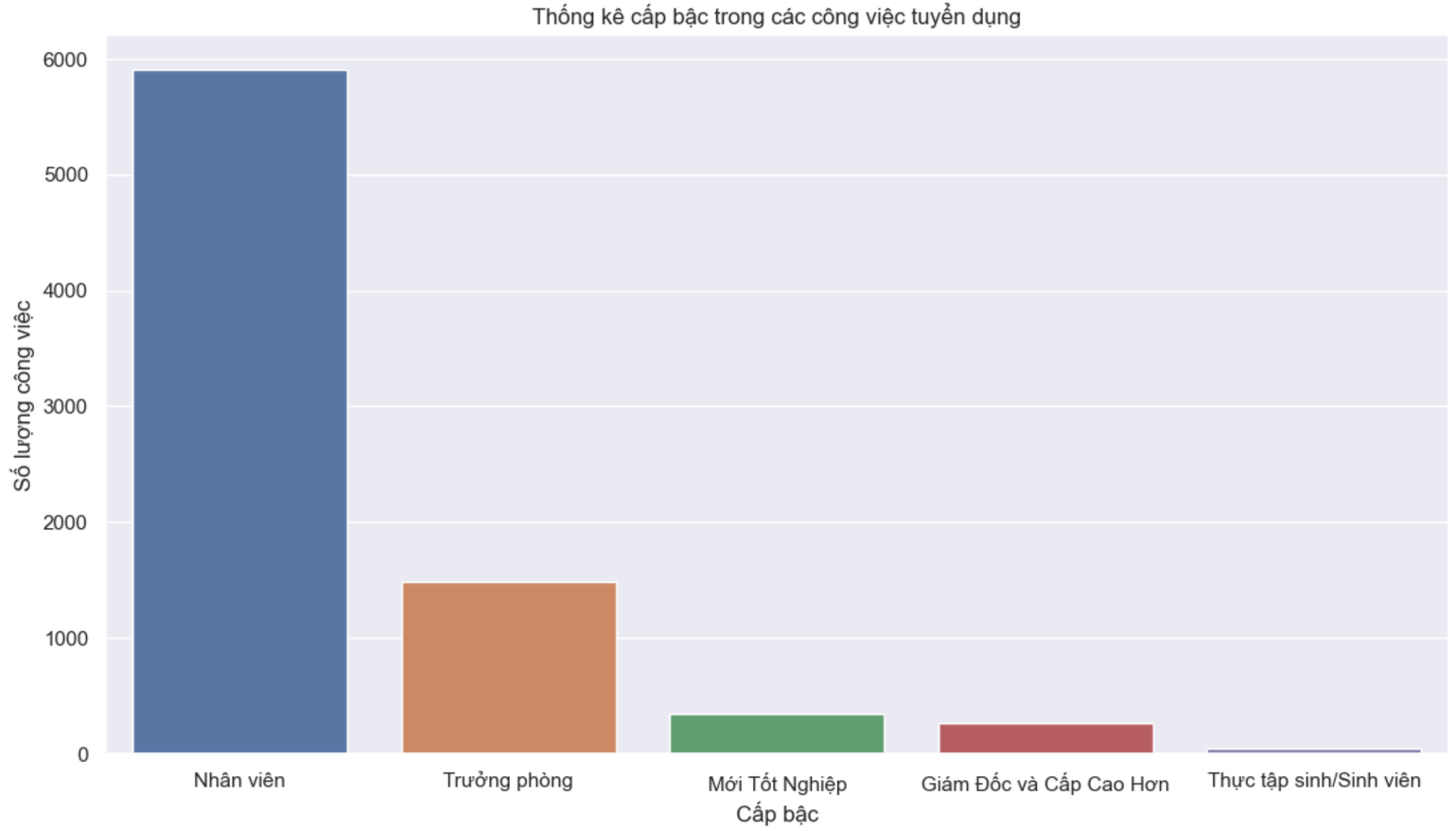
Các công ty tuyển dụng thường có quy mô như thế nào.



- Các công ty tuyển dụng chủ yếu có quy mô từ 100 - 499 và chỉ một số ít có quy mô 50000-100000.

Trực quan hóa dữ liệu

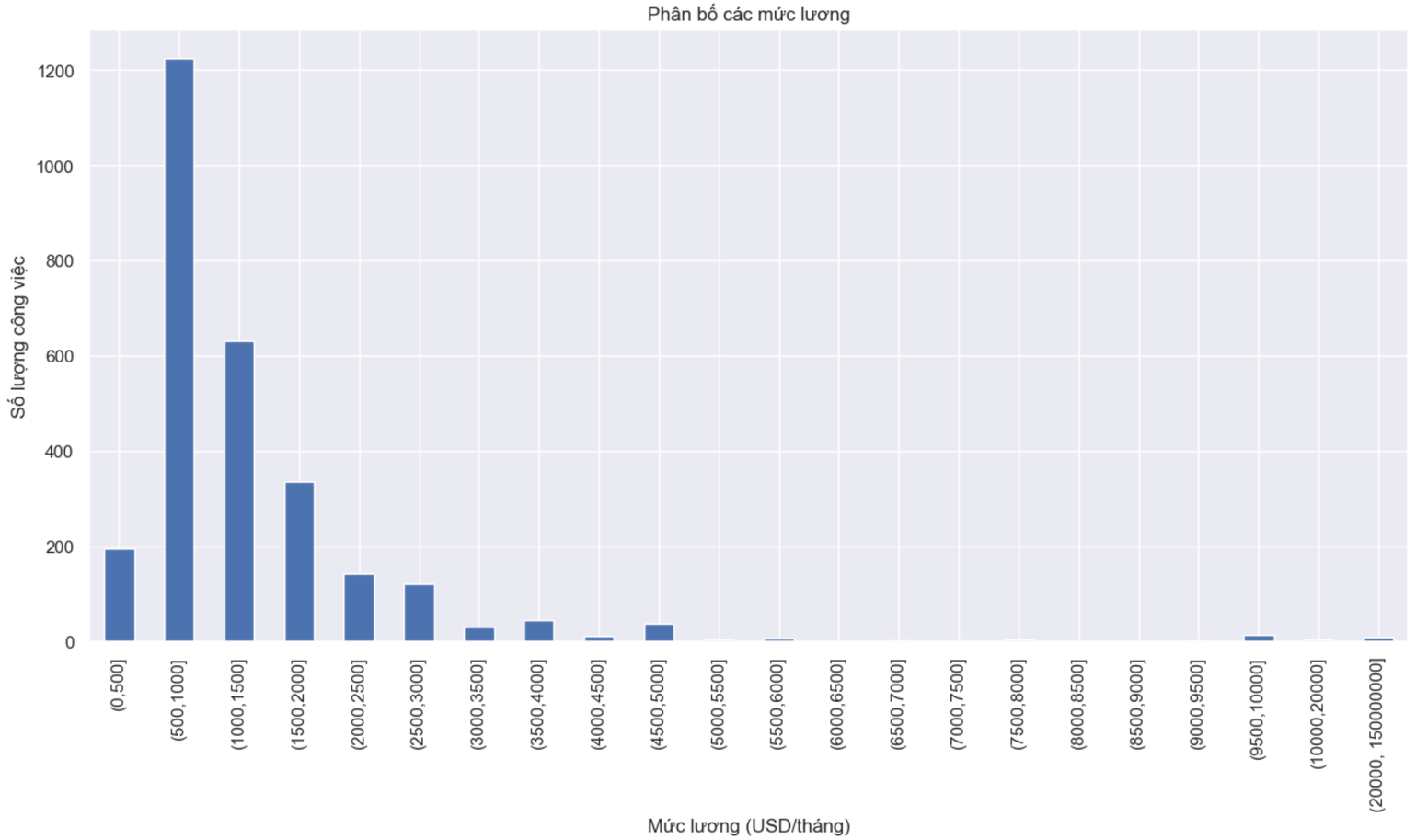
Các công việc tuyển dụng thường yêu cầu với cấp bậc nào.



- Các công ty tuyển dụng hầu như ở cấp bậc nhân viên, và thấp nhất đối với thực tập sinh/sinh viên.

Trực quan hóa dữ liệu

Các công việc tuyển dụng thường có mức lương dao động như thế nào.



- Mức lương chủ yếu của các công việc tuyển dụng dao động từ 500-1000 USD là chủ yếu.

Trực quan hóa dữ liệu

02. Tìm hiểu về những Top 20 thú vị

01 Top 20 những công ty có nhu cầu tuyển dụng cao.

02 Top 20 những ngành nghề được ưa chuộng trong tuyển dụng.

03 Top 20 kĩ năng được ưa chuộng trong tuyển dụng.

04 Top 20 địa điểm có lượt tuyển dụng cao.

Trực quan hóa dữ liệu

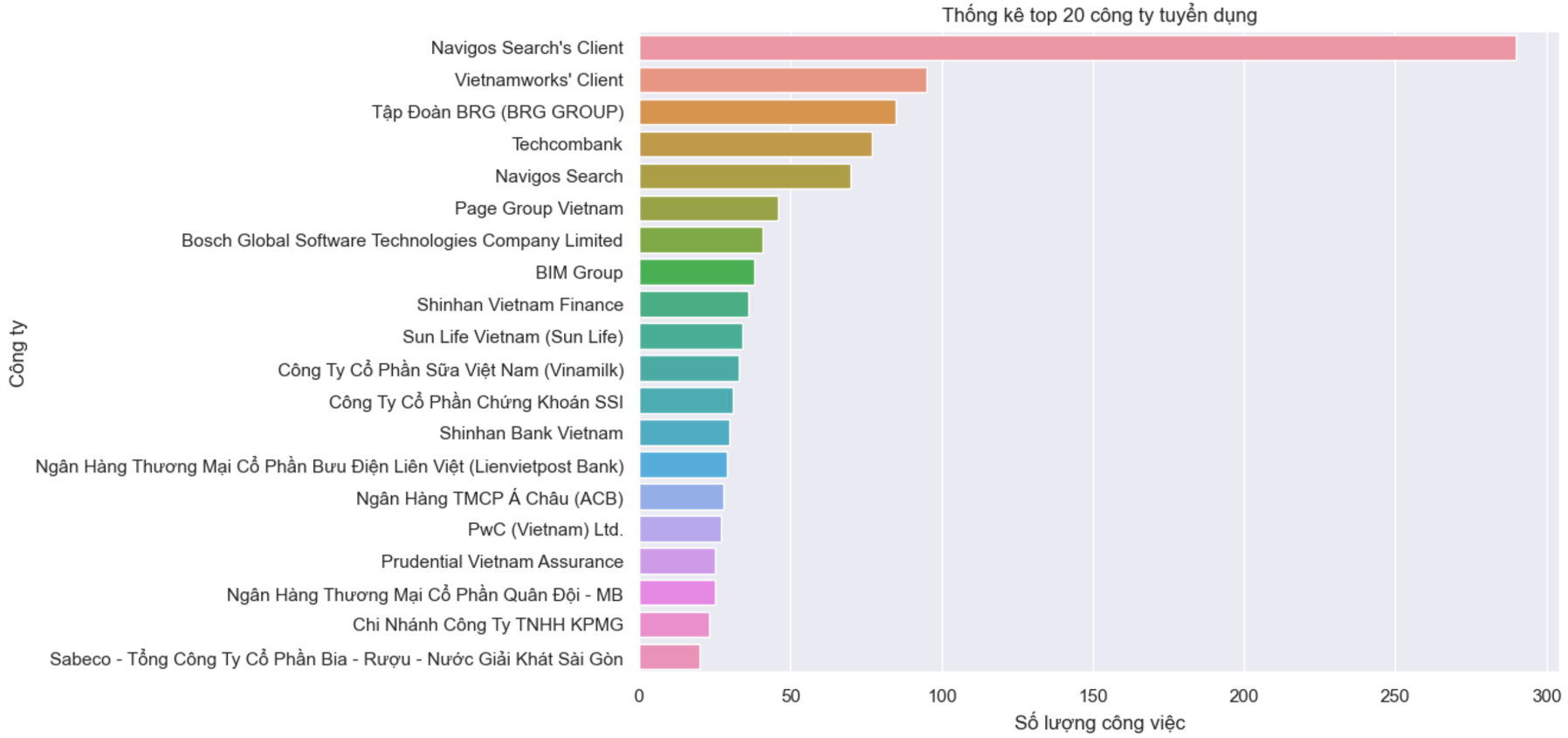
02. Tìm hiểu về những Top 20 thú vị

Lợi ích khi trả lời câu hỏi?

- Có được những thông tin về công ty, ngành nghề, địa điểm, kĩ năng được ưa chuộng để nắm bắt được những xu hướng tuyển dụng và từ đó có được những định hướng cho công việc tương lai.

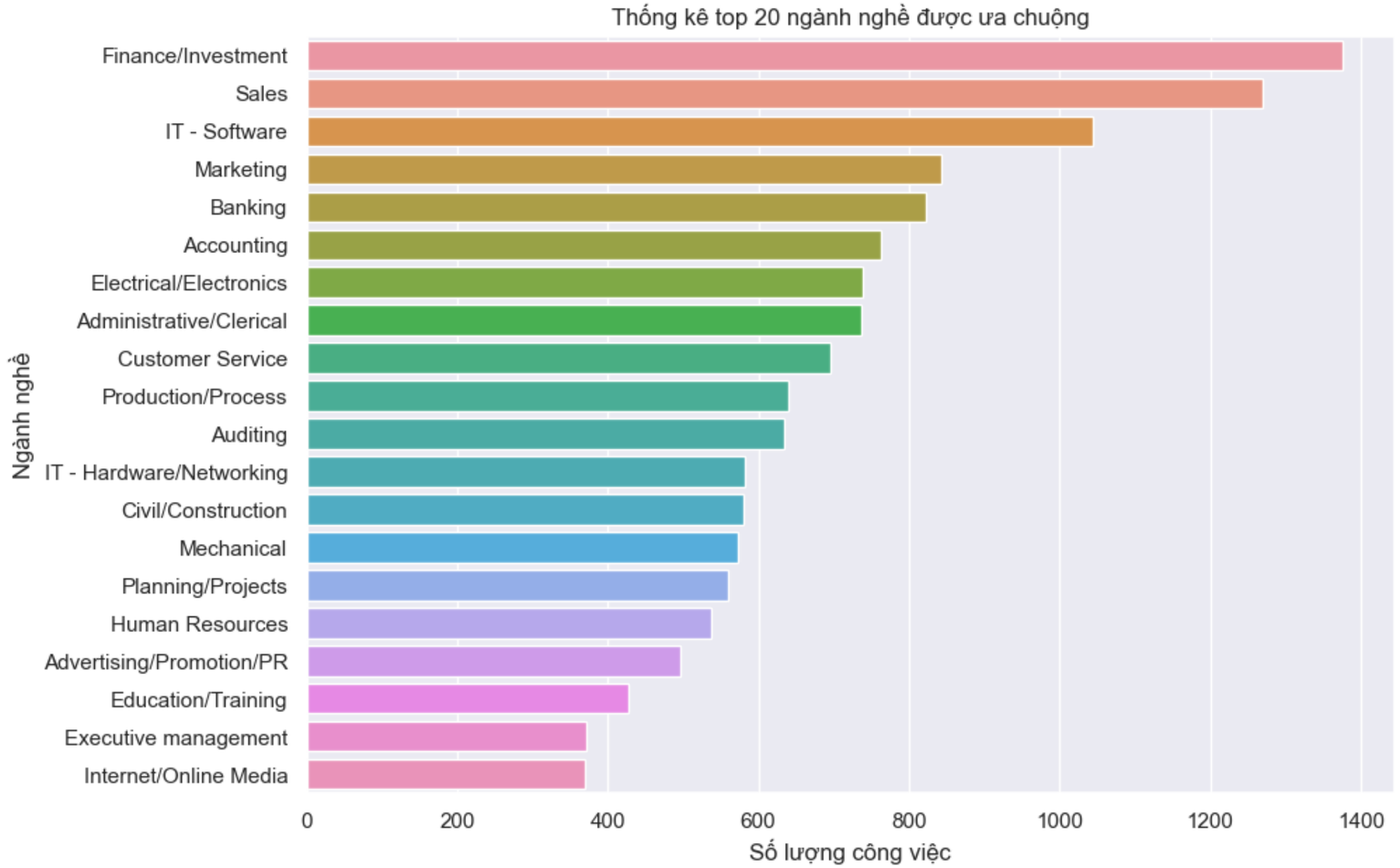
Trực quan hóa dữ liệu

Top 20 những công ty có nhu cầu tuyển dụng cao.



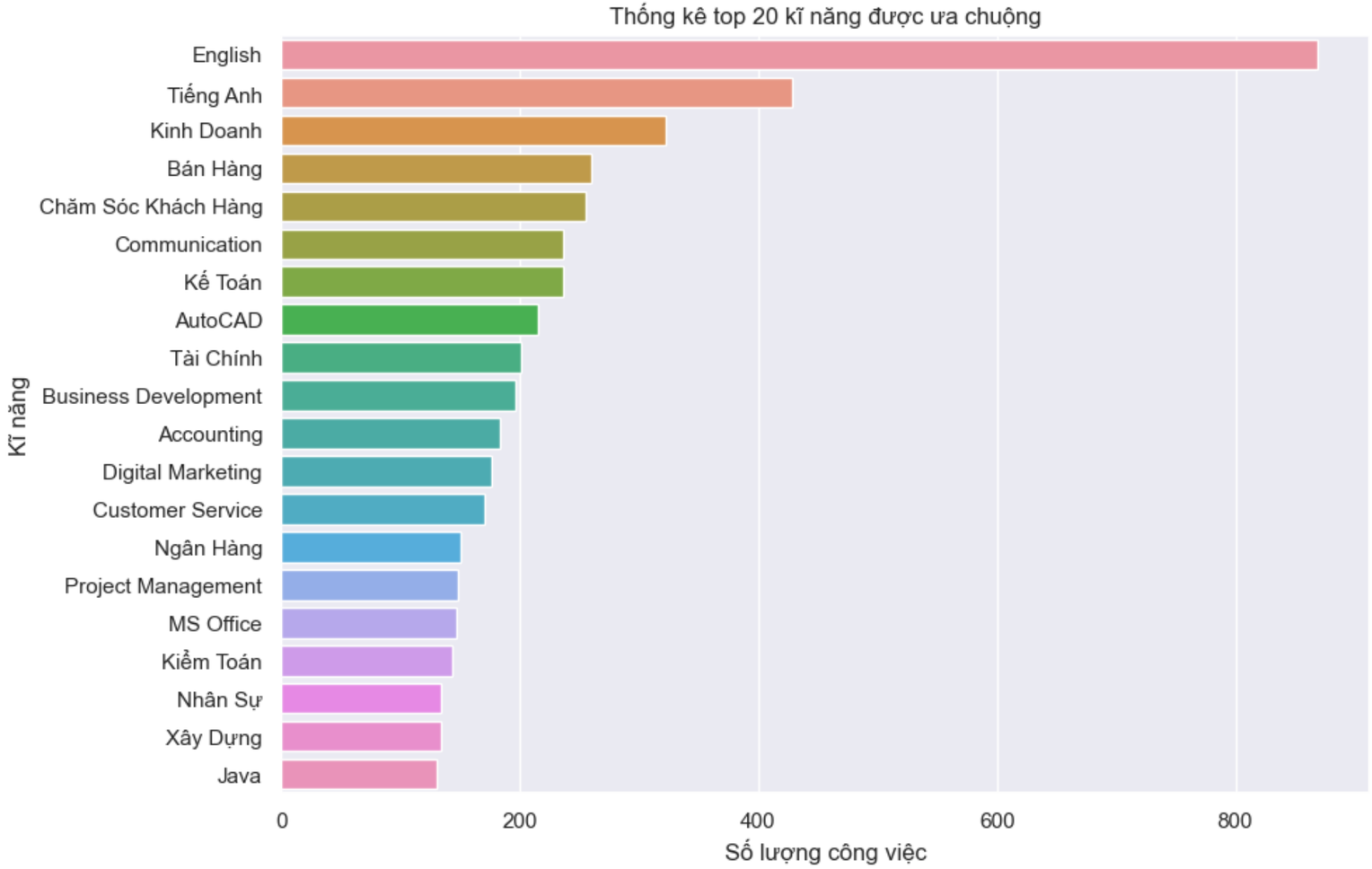
Trực quan hóa dữ liệu

Top 20 những ngành nghề được ưa chuộng trong tuyển dụng.



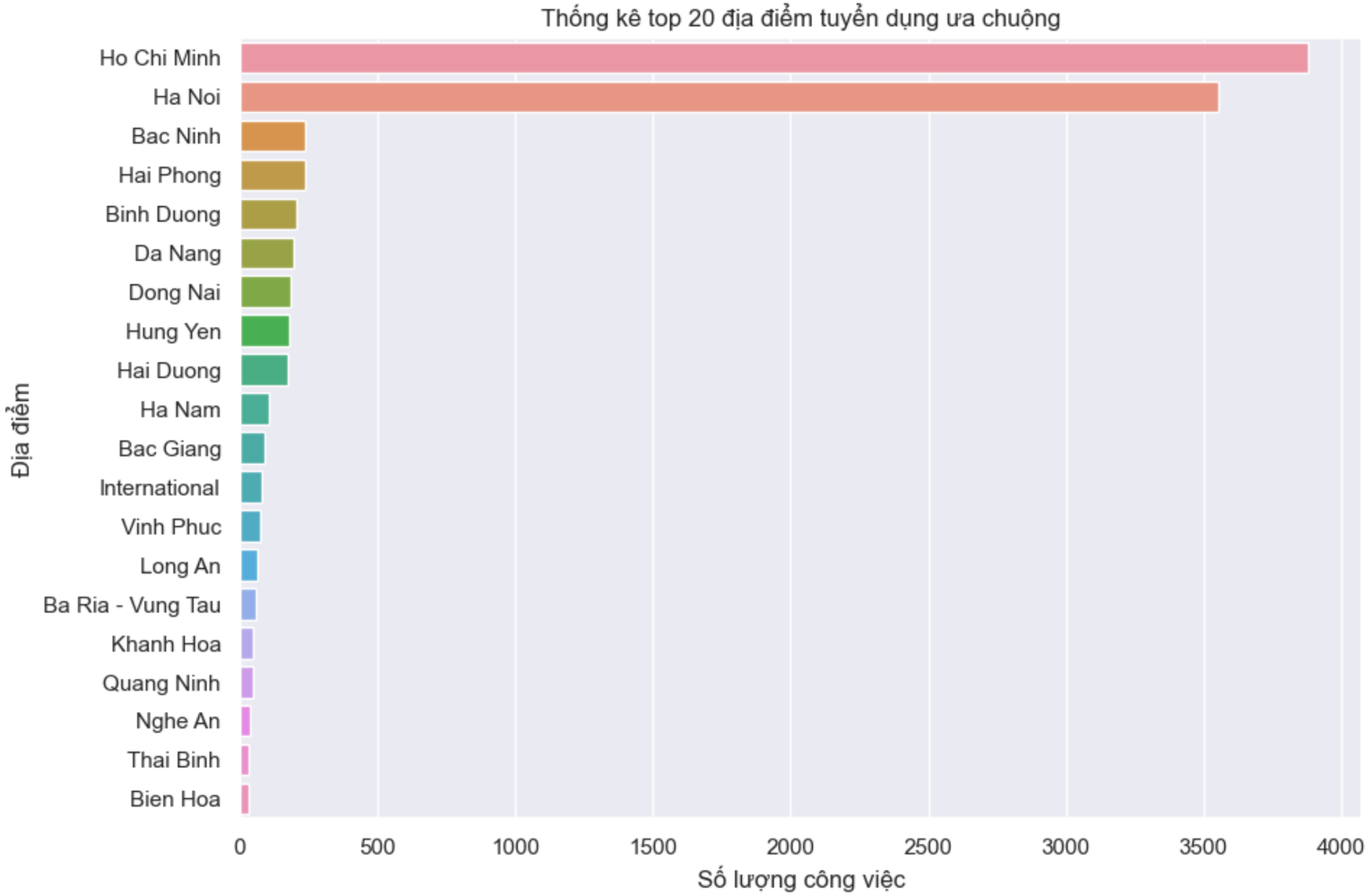
Trực quan hóa dữ liệu

Top 20 kĩ năng được ưa chuộng trong tuyển dụng.



Trực quan hóa dữ liệu

Top 20 địa điểm có lượt tuyển dụng cao



Trực quan hóa dữ liệu

03. Nơi tập trung nhiều công việc và thu nhập bình quân

Đối với sinh viên, người đi làm

- ❑ Xác định được những địa điểm có nhiều cơ hội việc làm.
- ❑ Cân nhắc tiếp tục làm ở địa phương hay tìm một nơi khác có nhiều thuận lợi hơn.

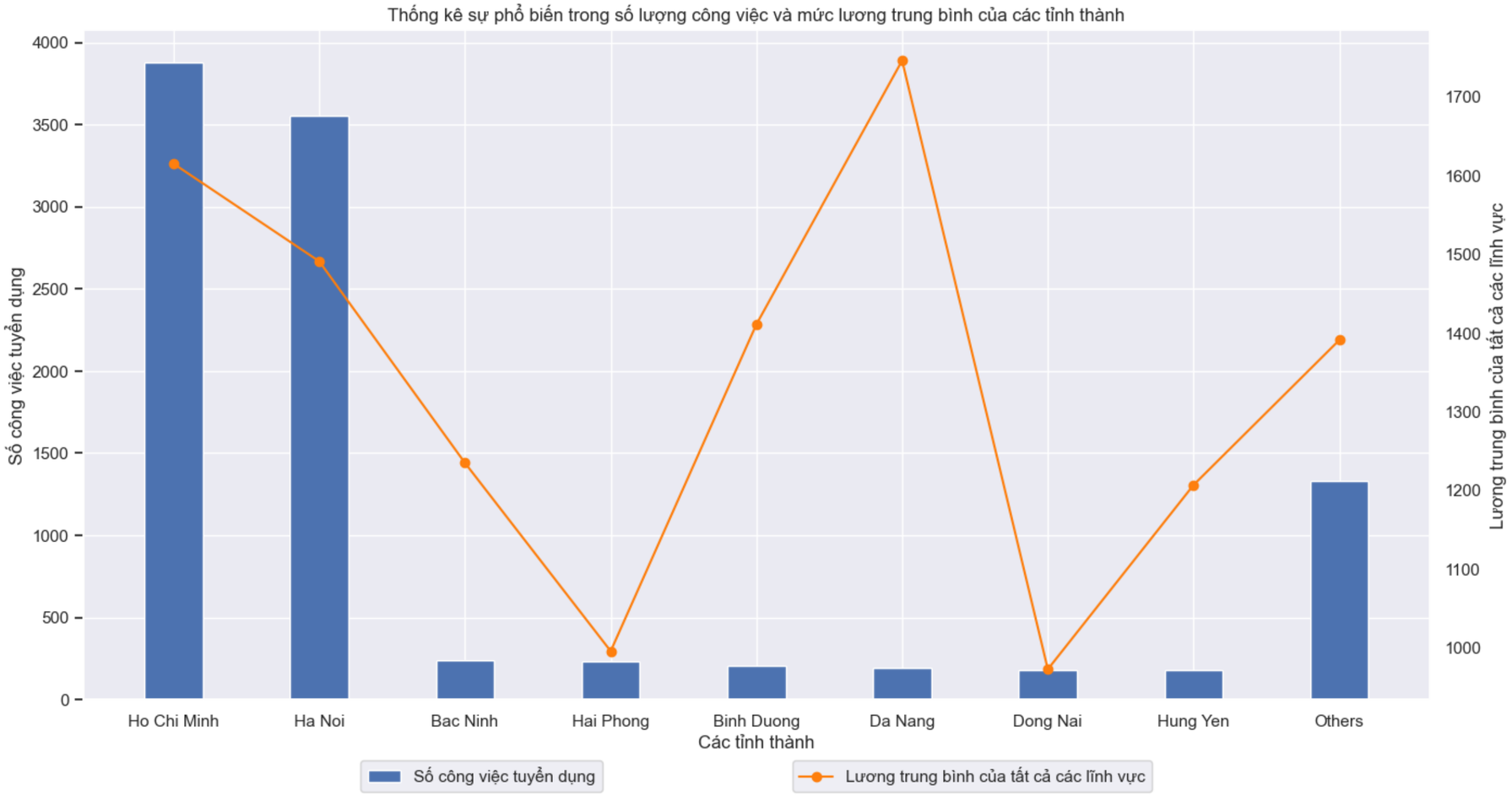
Đối với nhà tuyển dụng, nhà đầu tư

- ❑ Nắm bắt nơi tập trung lực lượng lao động dồi dào. Mở rộng quy mô ở những tỉnh thành khác tiềm năng.

Lợi ích khi trả lời câu hỏi?

Trực quan hóa dữ liệu

03. Nơi tập trung nhiều công việc và thu nhập bình quân



Trực quan hóa dữ liệu

03. Nơi tập trung nhiều công việc và thu nhập bình quân

Trả lời câu hỏi

- Hồ Chí Minh và Hà Nội vẫn là nơi có số lượng công việc nhiều nhất. Đây là những nơi tập trung của nhiều dân cư, trường học, đô thị. Được tạo nhiều điều kiện thuận lợi để phát triển từ chính quyền địa phương.

- Đà Nẵng, Hồ Chí Minh, Hà Nội, là nhưng nơi có mức lương trung bình cao nhất trong số những nơi có job nhiều. Đặc biệt Đà Nẵng dù có ít công việc hơn song được chi trả vô cùng hậu hĩnh. Nguyên nhân lớn nhất về sự khác biệt trong cơ cấu kinh tế của mỗi khu vực.

Trực quan hóa dữ liệu

04. Lĩnh vực có lượt đăng tải công việc và số lượng ứng viên đăng kí nhiều nhất

Lợi ích khi trả lời câu hỏi?

Đối với sinh viên, người đi làm

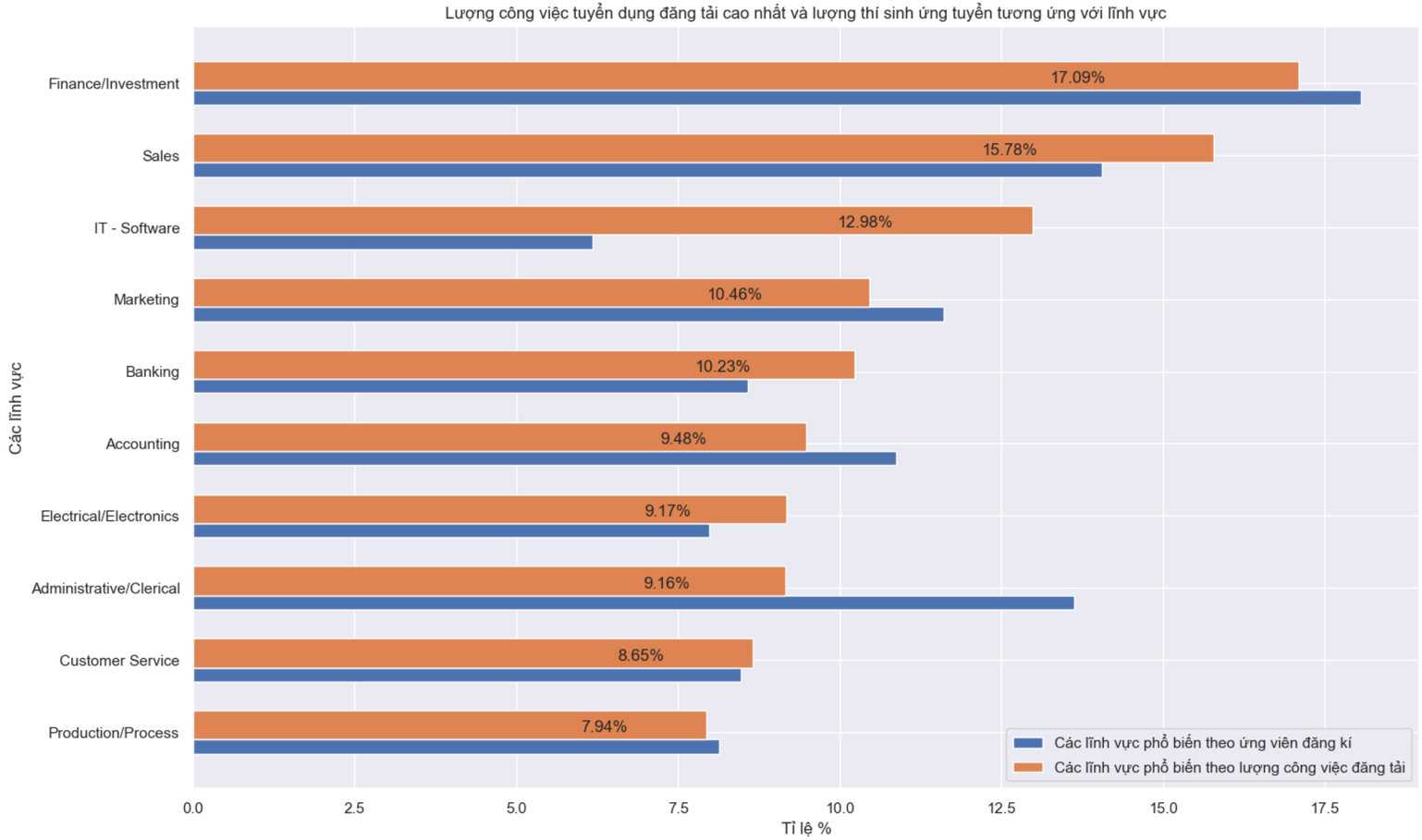
- ❑ Xác định được xu hướng công việc tại thời hiện tại đang tập trung ở những lĩnh vực nào.
- ❑ Vạch ra những định hướng nghề nghiệp, chiến lược học tập, làm việc cho bản thân.

Đối với nhà tuyển dụng, nhà đầu tư

- ❑ Mở các khóa đào tạo nghề, cam kết có việc làm sau khi hoàn tất quá trình đào tạo.
- ❑ Mở rộng hợp tác với các trường đại học hàng đầu đào tạo các ngành nghề phù hợp nhu cầu của công ty.

Trực quan hóa dữ liệu

04. Lĩnh vực có lượt đăng tải công việc và số lượng ứng viên đăng kí nhiều nhất



Tham khảo từ "Báo cáo thị trường IT Việt Nam – Tech Hiring 2022" - TopDev
https://topdev.vn/page/vietnam-it-market-reports

Trực quan hóa dữ liệu

04. Lĩnh vực có lượt đăng tải công việc và số lượng ứng viên đăng kí nhiều nhất

Trả lời câu hỏi

- Các ngành thuộc lĩnh vực kinh tế (Finance/Investment,Sales,...), công nghệ thông tin (IT-Software), kĩ thuật (Electical/Electronic) có số lượng công việc tuyển dụng được đăng tải nhiều nhất. Phản ánh xu thế chung của thế giới hiện nay.
- Hầu hết phần trăm các ngành có lượng công việc đăng tải cao, thì tương ứng với đó số lượng ứng viên đăng kí cũng tăng theo tương ứng. Nhưng ở lĩnh vực IT-Software, lượng ứng tuyển còn hơi thấp so với nhu cầu.

Trực quan hóa dữ liệu

05. Top những kĩ năng cần thiết cho các công việc hot hiện tại

Đối với sinh viên, người đi làm

- ❑ Trau dồi những kiến thức, kĩ năng cần thiết cho các lĩnh vực quan tâm.

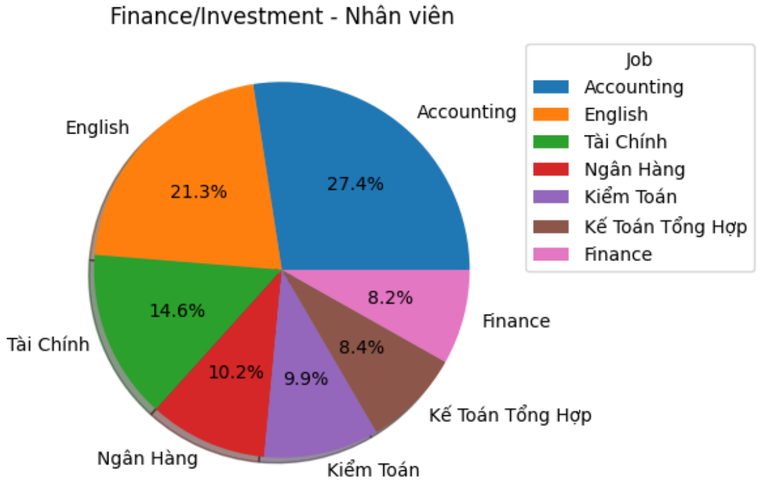
Đối với các trường đại học, trung tâm dạy nghề

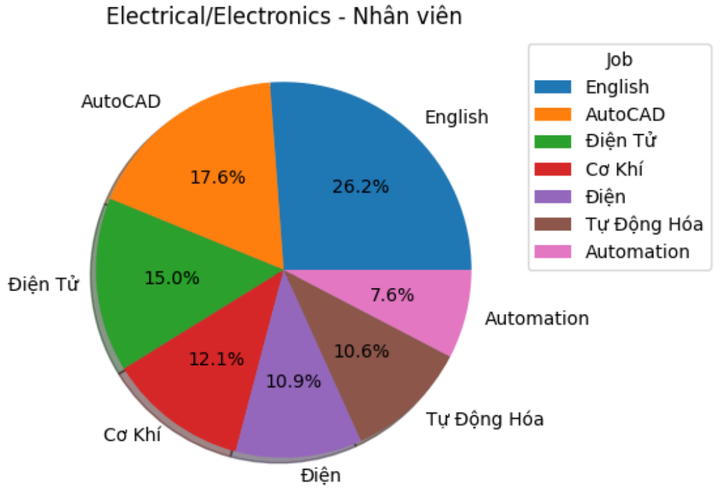
- ❑ Giảng dạy cho sinh viên về các kỹ năng phù hợp với xu hướng hiện tại.
- ❑ Đảm bảo các sinh viên ra trường phải được trang bị những kiến thức nền tảng thật tốt để đáp ứng các yêu cầu của doanh nghiệp đặc biệt những ngành hot, có sự cạnh tranh cao.

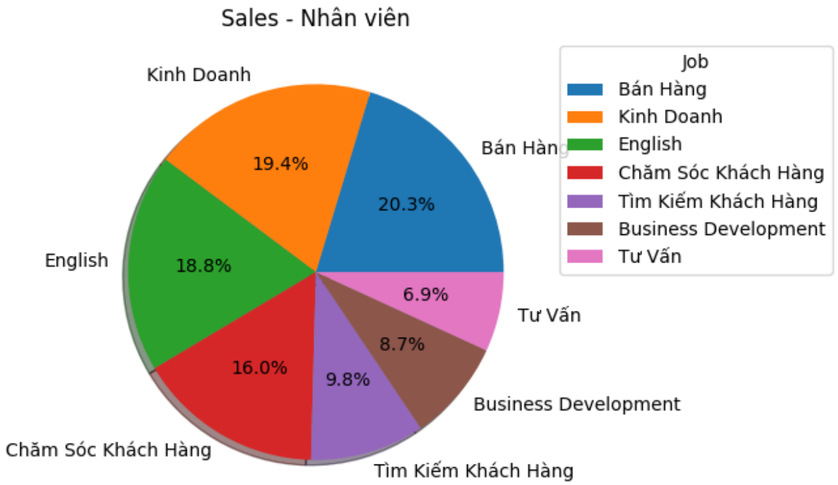
Lợi ích khi trả lời câu hỏi?

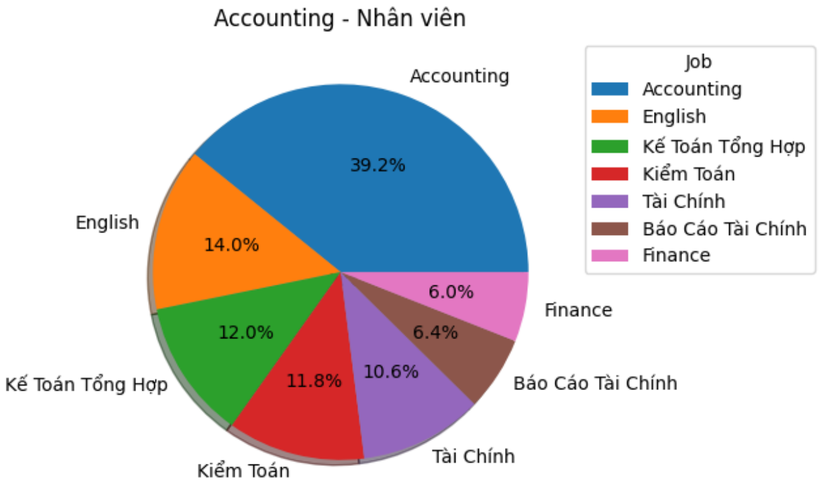
Trực quan hóa dữ liệu

05. Top những kĩ năng cần thiết cho các công việc hot hiện tại



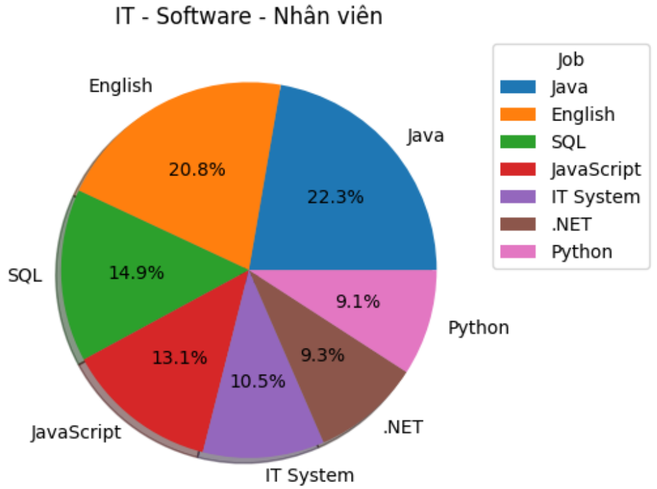


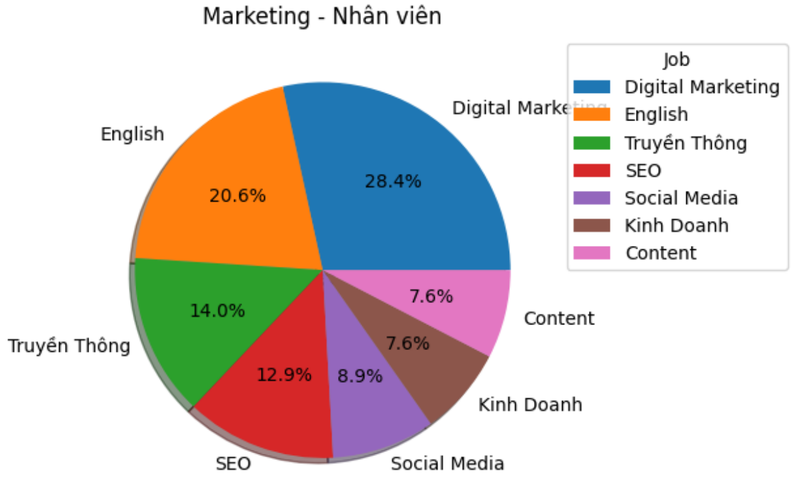


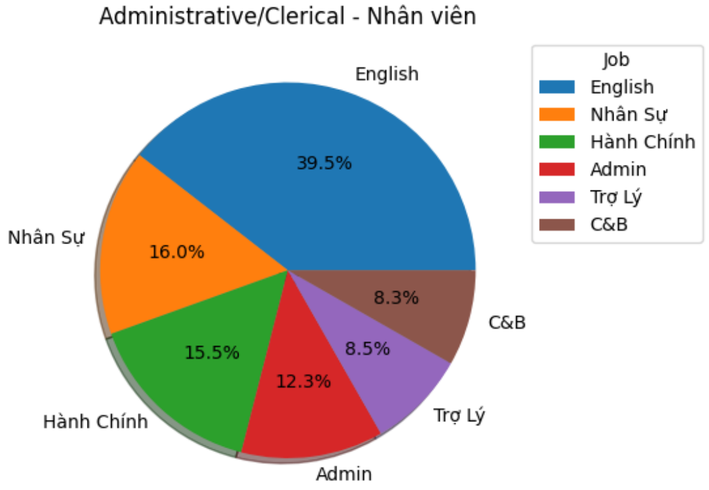


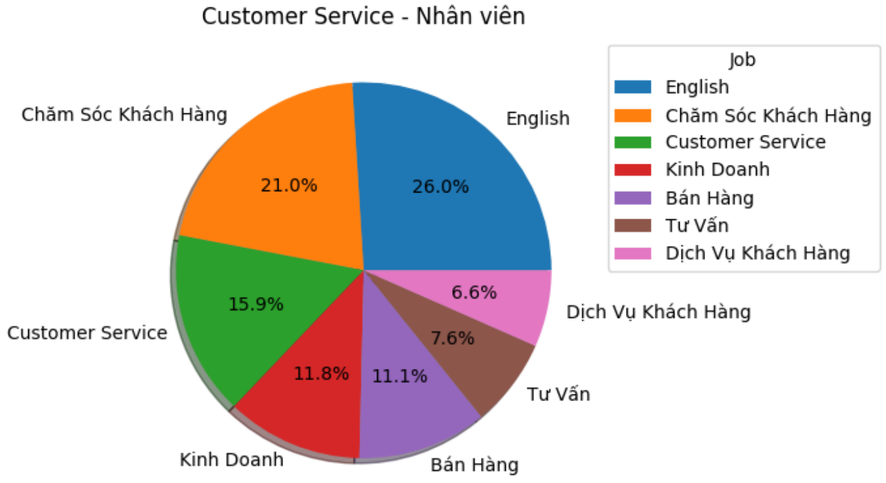
Trực quan hóa dữ liệu

05. Top những kĩ năng cần thiết cho các công việc hot hiện tại









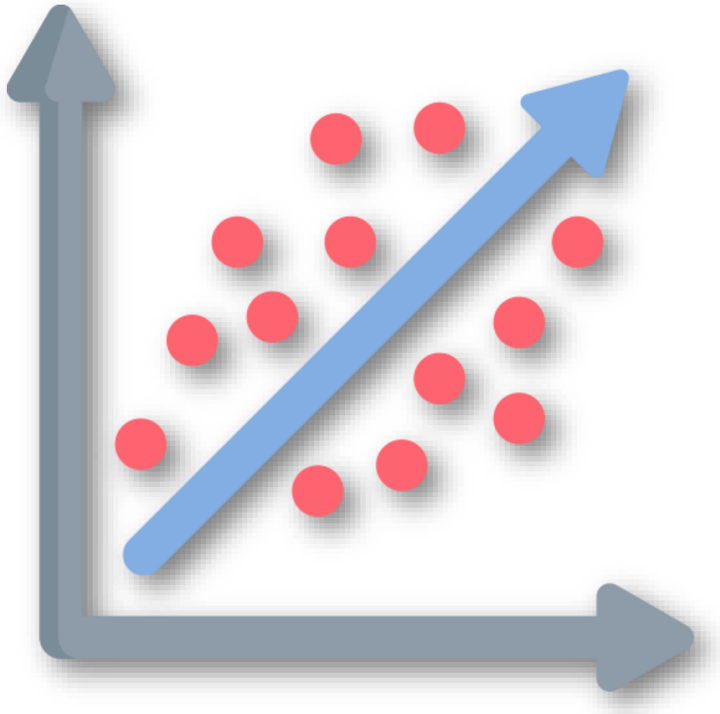
Trực quan hóa dữ liệu

05. Top những kĩ năng cần thiết cho các công việc hot hiện tại

Trả lời câu hỏi

- Tất cả các công việc đếu yêu cầu trình độ English là cao nhất. Điều này phản ánh rõ tình trạng thị trường ở Việt Nam hiện tại đang muốn mở rộng ra nước ngoài và tiếng anh là công cụ chính.

- Việc nhìn vào cơ cấu biểu đồ này, có thể biết được những kỹ năng mà công việc yêu cầu để dễ dàng tiếp cận công việc này hơn theo đúng hướng của xu hướng công việc hiện tại.



03. Xây dựng mô hình

Quy trình xây dựng mô hình

Tiền xử lý trước khi xây dựng mô hình

Trước khi xây dựng mô hình, ta cần xử lý dữ liệu để có dạng thích hợp mới có thể tiến hành mô hình hóa.

Ở đây, ta có tất cả 38 cột, song ta sẽ bỏ qua các cột quá đặc thù khó có thể khai thác như jobId, jobTitle, userId, companyLogo, alias,...

Những cột chúng ta sẽ lấy là: numOfApplications, salary, skills, industries, workingLocations, jobLevel, companySize, approvedOn, typeWorkingName

Tiền xử lý trước khi xây dựng mô hình

- Kiểm tra các giá trị thiếu

```
application_df.isna().sum()
```

```
numOfApplications        0
salary                   0
skills                   0
industries               0
workingLocations         0
jobLevel                 0
companySize           1189
approvedOn               0
typeWorkingName          0
dtype: int64
```

- Ở cột companySize, dữ liệu bị thiếu khá nhiều => Điền thêm dữ liệu cho các giá trị cột này (Chuyển về dạng số và lấy trung bình, sau đó điền trung vị)

```
#Chuyển dữ liệu từ dạng object sang numberic
application_df['companySize'] = application_df['companySize'].apply(lambda x : math.ceil(np.mean([int(size) for size in str(x).replace('(','').replace(')','')
.split(',')])) if not isinstance(x, float) else np.nan)

#Bổ sung các giá trị thiếu
median_size = math.floor(application_df['companySize'].median())
application_df['companySize'] = application_df['companySize'].fillna(median_size)
```

Tiền xử lý trước khi xây dựng mô hình

- Cột salary ở đây có rất nhiều giá trị 0 => Cần bổ sung bằng giá trị trung vị ở cột này

```
median_salary = math.floor(application_df['salary'].loc[application_df['salary'] > 0].median())
application_df['salary'] = application_df['salary'].replace(to_replace = 0, value = median_salary)
```

- Đối với approvedOn, ta sẽ chuyển đổi các dữ liệu này về dạng số (hiệu timestamp) với ý nghĩa là số giờ (h) đã đăng tải

```
newest_date = application_df['approvedOn'].max()
application_df['approvedOn'] = application_df['approvedOn'].apply(lambda x : round((datetime.datetime
.fromisoformat(newest_date).timestamp() - datetime.datetime.fromisoformat(x).timestamp())/3600,0))
```

Tiền xử lý trước khi xây dựng mô hình

- Chuyển đổi dữ liệu ở các cột có 1 giá trị dạng categorical về dạng numeric (OneHotEncoder)

```
ohe = OneHotEncoder()
encoder_df = pd.DataFrame(ohe.fit_transform(application_df[['jobLevel']]).toarray())
ohe.categories_[0]

ohe1_df = pd.DataFrame(ohe.fit_transform(application_df[['jobLevel']]).toarray(),columns=ohe.categories_)
ohe2_df = pd.DataFrame(ohe.fit_transform(application_df[['typeWorkingName']]).toarray(),columns=ohe.categories_)
application_df = pd.concat([application_df,ohe1_df.iloc[:,1:],ohe2_df.iloc[:,1:]],axis=1)
application_df = application_df.drop(['jobLevel','typeWorkingName'], axis=1)
```

- Với những cột có nhiều hơn 1 giá trị, ta sẽ sử dụng MultiLabelBinarizer

```
mlb = MultiLabelBinarizer()
mlb1_df = pd.DataFrame(mlb.fit_transform(application_df['skills'].apply(lambda x : x.split(';'))),columns=mlb.classes_)
mlb2_df = pd.DataFrame(mlb.fit_transform(application_df['industries'].apply(lambda x : x.split(';'))),columns=mlb.classes_)
mlb3_df = pd.DataFrame(mlb.fit_transform(application_df['workingLocations'].apply(lambda x : x.split(';'))),columns=mlb.classes_)

application_df = pd.concat([application_df,mlb1_df,mlb2_df,mlb3_df],axis=1)
application_df = application_df.drop(['skills','industries','workingLocations'], axis=1)
```

Tiền xử lý trước khi xây dựng mô hình

❑ Feature Selection for Numeric

```
numeric_df = application_df[['numOfApplications','salary','companySize','approvedOn']].copy()
corr = numeric_df.corr()
corr
```

Vì các cột **salary**, **companySize** có corr thấp (không có ý nghĩa cho việc xây dựng mô hình), nên ta sẽ loại bỏ các cột này

```
application_df = application_df.drop(['salary','companySize'], axis=1)
```

❑ Feature Selection for Categorical

Ở đây nhóm sẽ sử dụng **Chi-Square** để lấy ra các đặc trưng có mức độ liên quan cao đối với các cột có kiểu là `Categorical` mà ta đã encode về dạng số trước đó

```
categorical_df = application_df.drop(['numOfApplications', 'approvedOn'], axis=1)
chi_scores = chi2(categorical_df, application_df['numOfApplications'])
chi_scores
```

Ta sẽ loại bỏ đi những cột mà có p-value > 0.05 (Tức có ít liên quan đến với việc dự đoán của bài toán)

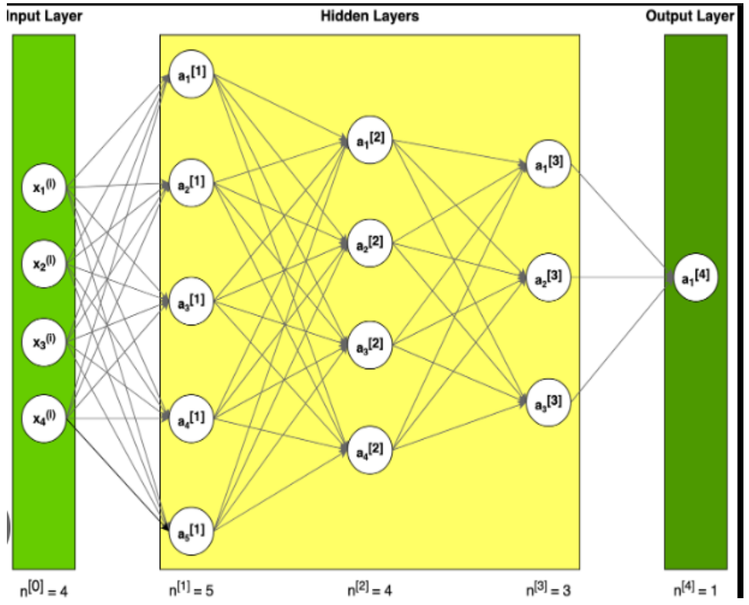
+ Code + Markdown

```
lower_importance_columns = p_values[p_values.values > 0.05].index.values
application_df = application_df.drop(columns=lower_importance_columns, axis=1)
```

Các mô hình đã chọn

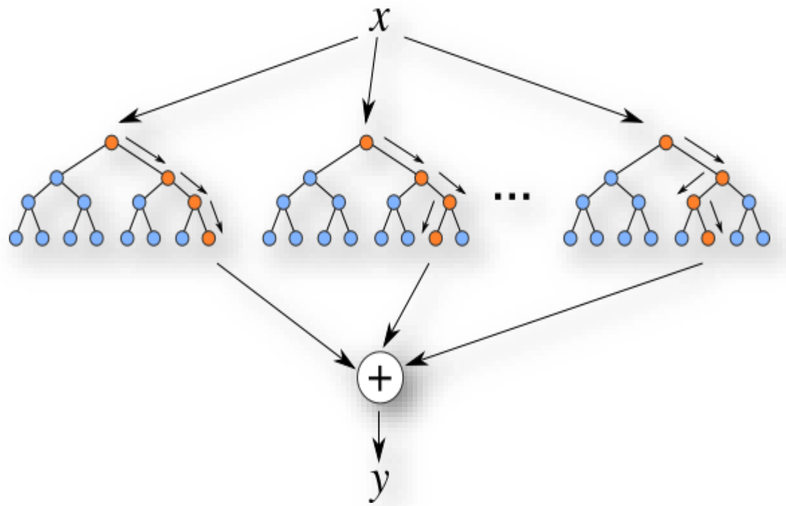
Neuron networks

Thuật toán:
Multiple layer perception Regressor



RandomForest

Thuật toán:
Random Forest Regressor



Các bước đầu khi xây dựng mô hình

01 Chọn các tập dữ liệu (independent variable) X và tập giá trị (dependent variable) Y cần thiết cho việc mô hình bài toán đang xét

```
y = application_df['numOfApplications']
X = application_df.drop(['numOfApplications'],axis = 1)
```

Các bước đầu khi xây dựng mô hình

02 Chia tập bộ dữ liệu thành các tập train (tập huấn luyện),tập test (Tập kiểm tra) và tập validation (Tập kiểm định) (80:10:10)

Tập train

Là tập dữ liệu được sử dụng để huấn luyện mô hình. Các thuật toán học máy sẽ học các mô hình từ tập huấn luyện này. Việc học sẽ khác nhau tùy thuộc vào thuật toán và mô hình sử dụng

Tập test

Là tập giá trị được sử dụng để đánh giá độ chính xác hoặc sai số của mô hình dự đoán đã được huấn luyện. Ta cần so sánh với tập test để đánh giá khả năng tổng quát hóa với những dữ liệu không chỉ đã học mà còn với những dữ liệu mới, chưa gặp trước đó.

Tập validation

Cung cấp các đánh giá công bằng về sự phù hợp của mô hình trên tập dữ liệu huấn luyện trong quá trình huấn luyện. Validation set có chức năng như một sự kết hợp: nó vừa là dữ liệu huấn luyện được sử dụng để thử nghiệm, nhưng không phải là một phần của quá trình huấn luyện cấp thấp cũng không phải là một phần của thử nghiệm cuối cùng. Nó là một bước chung gian cho phép lựa chọn mô hình phù hợp.

Các bước đầu khi xây dựng mô hình

03 Scale dữ liệu (Feature scaling)

- Đây là một phương pháp được dùng để scale (điều chỉnh) phạm vi của các giá trị để làm cho phù hợp với các giá trị đặc trưng trong tập dữ liệu, đặc biệt đối với những tập có phạm vi chênh lệch lớn.
- Có 2 cách phổ biến để scale dữ liệu đó là Normalization và Standardization. Trong đó Normalization sẽ scale khoảng dữ liệu bất kì về 0 -> 1, còn Standardization sẽ scale dữ liệu về một phân bố trong đó giá trị trung bình của các quan sát là 0 và độ lệch chuẩn là 1.

Ở bài toán này, ta sẽ scale các dữ liệu ở các cột salary, companySize, approvedOn bằng phương pháp Standardization bằng StandardScaler (thư viện có sẵn của scikit-learn)

```
sc=StandardScaler()
X1_train[['approvedOn']] = sc.fit_transform(X1_train[['approvedOn']])
X1_val[['approvedOn']] = sc.fit_transform(X1_val[['approvedOn']])
X1_test[['approvedOn']] = sc.fit_transform(X1_test[['approvedOn']])
```

Xây dựng mô hình – Mô hình neuron net

- Sử dụng MLPRegressor trong module neural_network của sklearn
- Mô hình MLPRegressor sẽ tối ưu hóa được lỗi bình phương (Square Error)

Thiết lập cho thuật toán:

- hidden_layer_sizes: Số lượng neuron trong lớp ẩn thứ i.
- random_state: Xác định việc tạo số ngẫu nhiên cho trọng số và khởi tạo độ lệch, phân tách thử nghiệm huấn luyện nếu sử dụng tính năng dừng sớm
- max_iter: Số lần lặp tối đa
- early_stopping: Chức năng dừng sớm
- verbose: In ra quá trình

```
mlpregressor = MLPRegressor(hidden_layer_sizes=(256, 512, 512, 256, ),random_state=0, max_iter=500, early_stopping=True, verbose=1)

model_1 = mlpregressor.fit(X1_train,y1_train)
```

Huấn luyện & dự đoán
Mô hình neuron_net

Code huấn luyện - dự đoán

```
y1_val_pred = model_1.predict(X1_val)
```

```
comparison_df = pd.DataFrame({'Actual': y1_val,'Predicted': y1_val_pred})
comparison_df
```

Kết quả

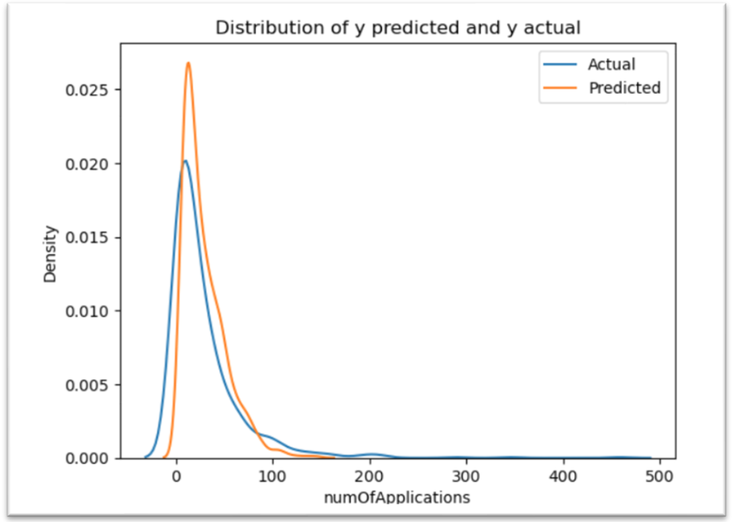
	Actual	Predicted
5878	47	33.829479
593	3	15.493029
2823	23	39.156752
5392	9	17.771997
3464	8	12.035661
...	...	...
4700	35	11.532030
7472	31	56.124741
5260	11	27.864803
2359	54	87.786244
2519	38	16.098290

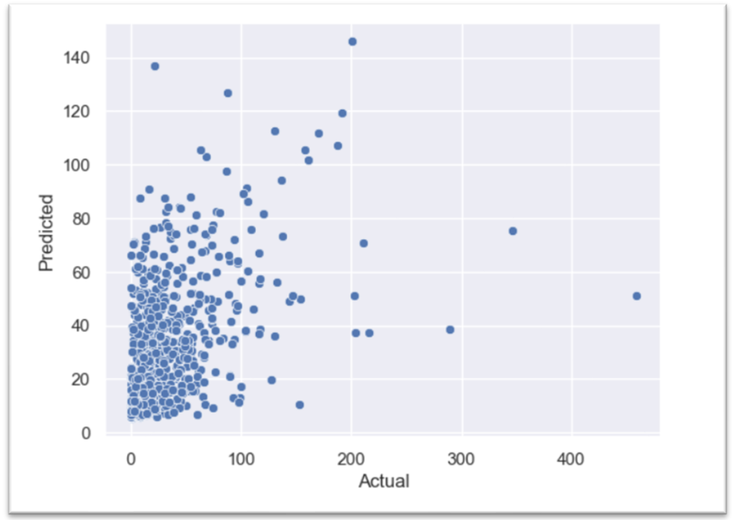
Kết quả hồi quy- Mô hình neuron_net

Code vẽ minh họa

```
plt.title("Distribution of y predicted and y actual")
ax1=sns.kdeplot(y1_val, label = 'Actual')
sns.kdeplot(y1_val_pred, ax=ax1, label = 'Predicted')
ax1.legend(loc="upper right")
plt.show()
```

```
sns.set_theme()
sns.scatterplot(x = y1_val,y = y1_val_pred)
plt.xlabel('Actual')
plt.ylabel('Predicted')
plt.show()
```





Xây dựng mô hình – Mô hình RandomForest

- Sử dụng RandomForestRegression trong module RandomForest của sklearn

Thiết lập cho thuật toán:

- n_estimators: Số cây trong rừng
- random_state: Kiểm soát cả tính ngẫu nhiên của việc khởi động các mẫu được sử dụng khi xây dựng cây

```
model_2 = RandomForestRegressor(n_estimators = 100, random_state = 0)
model_2.fit(X2_train, y2_train)
```

Huấn luyện & dự đoán
Mô hình RandomForest

Code huấn luyện - dự đoán

```
y2_val_pred = model_2.predict(X2_val)
```

```
comparison_df = pd.DataFrame({'Actual': y2_val,'Predicted': y2_val_pred})
comparison_df
```

Kết quả

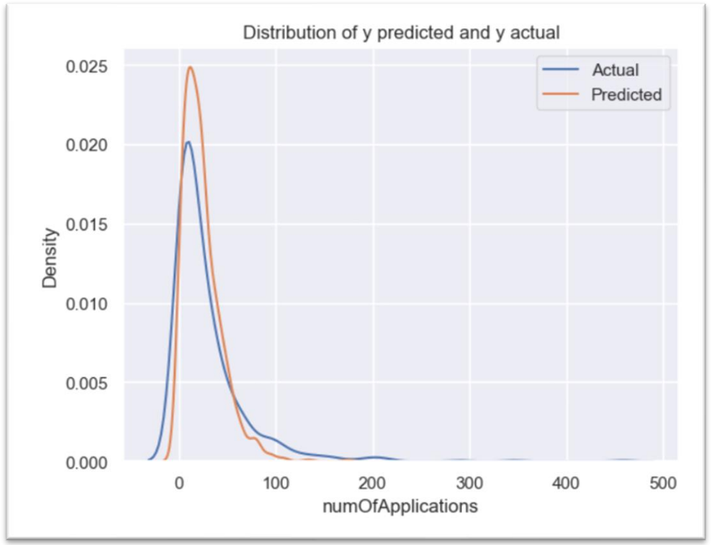
	Actual	Predicted
5878	47	26.087167
593	3	5.504667
2823	23	46.782000
5392	9	19.840000
3464	8	4.870905
...	...	...
4700	35	10.415833
7472	31	76.780000
5260	11	10.216548
2359	54	37.626000
2519	38	16.770000

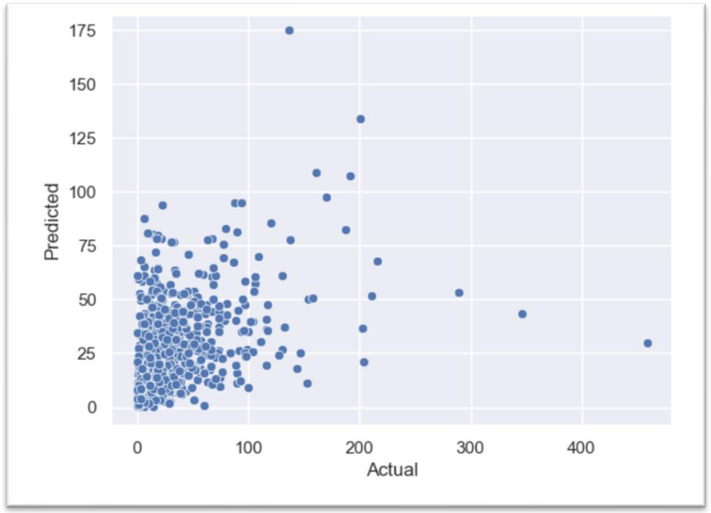
Kết quả hồi quy- Mô hình RandomForest

Code vẽ minh họa

```
plt.title("Distribution of y predicted and y actual")
ax1=sns.kdeplot(y2_val, label = 'Actual')
sns.kdeplot(y2_val_pred, ax=ax1, label = 'Predicted')
ax1.legend(loc="upper right")
plt.show()
```

```
sns.set_theme()
sns.scatterplot(x = y2_val,y = y2_val_pred)
plt.xlabel('Actual')
plt.ylabel('Predicted')
plt.show()
```





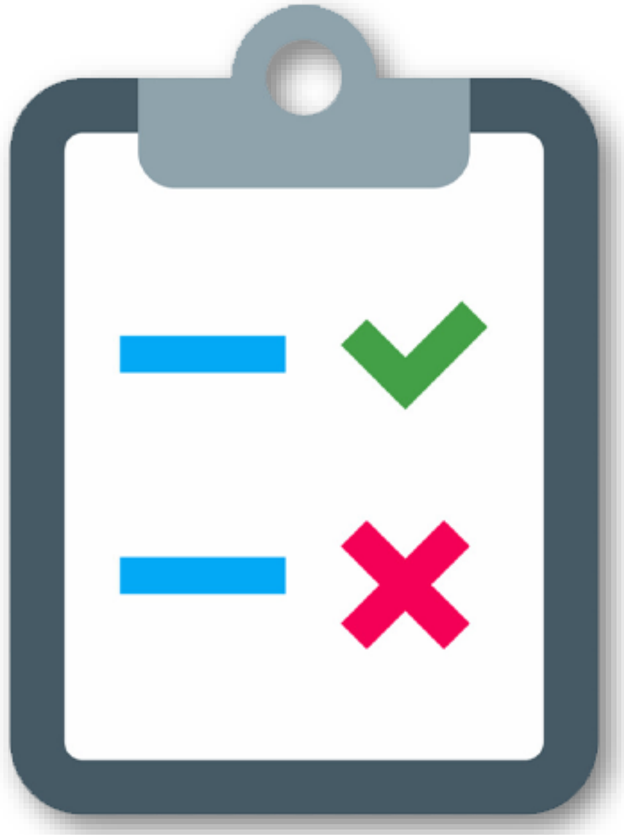
Kết luận

- Cả 2 thuật toán sử dụng đều có độ chính xác khá thấp, có thể là do thuật toán sử dụng chưa phù hợp và cũng chưa tìm được số lớp ẩn (hidden_layer_sizes) và số cây trong rừng (n_estimators) chưa được tối ưu.
- Ngoài ra thì cũng có thể do dữ liệu còn bị nhiễu khá nhiều khi cột salary có rất nhiều giá trị 0, companySize còn quá nhiều giá trị thiếu,...

=> Vì vậy cần tìm nhiều thuật toán tối ưu hơn để huấn luyện và dự đoán

04. Đánh giá mô hình

Quy trình đánh giá mô hình



Đánh giá mô hình

01 | R^2 Score , the coefficient of determination

02 | Mean Absolute Error (MAE)

Đánh giá mô hình

R^2 Score

- Đánh giá hiệu xuất mô hình máy học dựa trên hồi quy.
- Là thước đo xem các mẫu không nhìn thấy có khả năng được mô hình dự đoán tốt như thế nào, thông qua tỷ lệ phương sai được giải thích.
- Giá trị R^2 dao động từ 0 đến 1 và có thể âm.

Mean Absolute Error (MAE)

- Đánh giá hiệu xuất mô hình máy học dựa trên hồi quy.
- Sai số trung bình tuyệt đối (Mean absolute error) đo lường mức độ trung bình của các lỗi trong một tập hợp các dự đoán, mà không xem xét hướng của chúng.

Đánh giá mô hình

R^2 Score

- Công thức

$$R^2(y, \hat{y}) = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2}$$

Trong đó:

n là số điểm dữ liệu

$\hat{y}_i$ là giá trị dự đoán

y_i là giá trị thực.

Mean Absolute Error (MAE)

- Công thức

$$\text{MAE} = \frac{\sum_{i=1}^{n}|x_i - y_i|}{n}$$

Trong đó:

n là số điểm dữ liệu

x_i là giá trị thực

y_i là giá trị dự đoán.

Đánh giá mô hình

R^2 Score

❑ Cài đặt

```
def R2_Score(y_test,y_pred):
    rss = np.sum(np.square(y_test - y_pred))
    tss = np.sum(np.square(y_test - np.mean(y_test)))
    return 1 - (rss/tss)
```

Mean Absolute Error (MAE)

❑ Cài đặt

```
def mae(y_true, y_pred):
    return sum(abs(y_true - y_pred)) / len(y_true)
```

Với:

y_test, y_true: giá trị trong tập test

y_pred: giá trị trong tập dự đoán

Đánh giá mô hình

Multiple Layer Perception Regression

Multiple Layer Perception Regression

```
print('R Square (R^2)              : ',R2_Score(y1_test,y1_pred))
print('R Square (R^2) with sklearn: ', r2_score(y1_test,y1_pred))
```

```
R Square (R^2)              :  0.27308814722925734
R Square (R^2) with sklearn:  0.27308814722925734
```

```
print('Mean Absolute Error (MAE)              : ', mae(y1_test,y1_pred))
print('Mean Absolute Error (MAE) with sklearn: ', mean_absolute_error(y1_test,y1_pred))
```

```
Mean Absolute Error (MAE)              :  19.10189764272016
Mean Absolute Error (MAE) with sklearn:  19.101897642720164
```

Random Forest Regression

Random Forest Regression

```
print('R Square (R^2)              : ',R2_Score(y2_test,y2_pred))
print('R Square (R^2) with sklearn: ', r2_score(y2_test,y2_pred))
```

```
R Square (R^2)              :  0.16967481780481375
R Square (R^2) with sklearn:  0.16967481780481375
```

```
print('Mean Absolute Error (MAE)              : ', mae(y2_test,y2_pred))
print('Mean Absolute Error (MAE) with sklearn: ', mean_absolute_error(y2_test,y2_pred))
```

```
Mean Absolute Error (MAE)              :  18.94059971318819
Mean Absolute Error (MAE) with sklearn:  18.940599713188202
```

Đánh giá mô hình

Tổng kết:

01 Cả hai mô hình đều chưa thực sự tối ưu đối với bộ dữ liệu.

02 Nguyên nhân phần lớn đến từ bộ dữ liệu ta thu thập có thể chưa đủ lớn.

03 Có thể bị thiếu đối với những thông tin đầu vào ảnh hưởng đến dự đoán.

04 Tập dữ liệu bị thiếu nhiều dẫn đến việc bổ sung dữ liệu ảnh hưởng kết quả dự đoán của mô hình.

Tổng kết

Thực hành được quy trình của một bài toán Khoa học dữ liệu:

01 Thu thập, tiền xử lý dữ liệu:
- Sử dụng HTTP Requests để crawl data.
- Các kỹ năng trong việc khám phá và tiền dữ liệu

02 Trực quan hóa dữ liệu:
- Kĩ năng đặt và trả lời câu hỏi sau đó trực quan nó bằng biểu đồ

03 Xây dựng mô hình:
- Biết sử dụng cơ bản 2 thuật toán MLPRegressor và RandomForestRegressor để dự đoán

04 Đánh giá mô hình:
- Biết được các kỹ năng cơ bản để đánh giá mô hình

Tài liệu tham khảo

01 Đánh giá model trong machine learning: https://viblo.asia/p/danh-gia-model-trong-machine-learing-RnB5pAq7KPG

02 Metrics and scoring: https://scikit-learn.org/stable/modules/model_evaluation.html

03 RandomForestRegressor: https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestRegressor.html

04 MLPRegressor: https://scikit-learn.org/stable/modules/generated/sklearn.neural_network.MLPRegressor.html#sklearn.neural_network.MLPRegressor

Thanks for listening & watching!!!