Phân tích trao đổi ngăn xếp

bởi David VanHeeswijk

Trong dự án này, chúng ta sẽ thảo luận về các chủ đề hay nhất để viết về Khoa học dữ liệu. Chúng ta sẽ khám phá trang web Stack Exchange để khám phá điều mà những ngư ời quan tâm đến Khoa học dữ liệu thấy thú vị nhất hoặc thách thức nhất liên quan đến chủ đề Khoa học dữ liệu.

Stack Exchange là một trang web đư ợc thiết kế để cung cấp một diễn đàn cho mọi ngư ời đặt câu hỏi về các chủ đề khác nhau thuộc nhiều lĩnh vực quan tâm và học tập, bao gồm Khoa học dữ liệu, Toán học, Tôn giáo, Kinh tế, v.v.

Khi xem chủ đề Khoa học dữ liệu trong Stack Exchange, chúng tôi thấy nhiều câu hỏi thuộc nhiều khía cạnh của Khoa học dữ liệu. Điều này bao gồm các câu hỏi như :

- Tôi nên thay thế bao nhiêu lớp trong CNN học chuyển tiếp?
- Độ phức tạp thời gian của mạng thần kinh giai đoạn học tập/thử nghiệm là gì?
- Cấu trúc thư a thớt cục bộ tối ư u của mạng tầm nhìn tích chập là gì?

Những gì chúng tôi muốn biết

Đối với dự án này, chúng tôi sẽ khám phá các câu hỏi đã đư ợc hỏi trong Stack Exchange nhằm cố gắng xem chủ đề nào phổ biến nhất. Cụ thể, chúng ta sẽ khám phá hai lĩnh vực chính của dữ liệu:

- Tổng số lần một chủ đề đư ợc gắn thẻ cho một câu hỏi
- Tổng số lần một chủ đề là một phần của câu hỏi đư ợc xem

Hai lĩnh vực chính này sẽ cho chúng ta ý tư ởng tốt về những câu hỏi nào đang đư ợc hỏi nhiều nhất trên Stack Exchange.

Đang tải dữ liệu

Trong 1]:

```
nhập gấu trúc dư ới dạng
pd nhập matplotlib.pyplot dư ới dạng plt
nhập seaborn dư ới dạng sns
nhập numpy dư ới dạng np

%matplotlib nội tuyến
```

Trong 2]:

```
df = pd.read_csv("2019_questions.csv", parse_dates=["CreationDate"])
```

Bây giờ chúng tôi đã tạo DataFrame của mình, bây giờ chúng tôi sẽ khám phá dữ liệu để sửa những gì cần sửa.

Làm sạch dữ liệu

Trong 3]:

```
df.info()
```

```
<lớp 'pandas.core.frame.DataFrame'>
RangeIndex: 8839 mục, 0 đến 8838
Các cột dữ liệu (tổng cộng 7 cột):
 # Cột Non-Null Count Dtype
--- ------ -------------- -----
 0 ID 8839 int64 không null
 1 CreationDate 8839 non-null datetime64[ns]
 2 điểm 8839 int64 không null
 3 ViewCount 8839 non-null int64
 4 Tags 5 8839 đối tượng không null
 AnswerCount 6 8839 không null int64
 FavoriteCount 1407 non-null float64 dtypes: datetime64[ns]
(1), float64(1), int64(4), object(1)
sử dụng bộ nhớ: 483,5+ KB
```

Chúng ta có thể thấy rằng cột FavoriteCount có rất nhiều giá trị null, chỉ có 1407 trong tổng số 8839 của chúng ta có một giá trị.

Chúng tôi cũng thấy rằng cột Thẻ được liệt kê dưới dạng một đối tượng, bây giờ chúng tôi sẽ khám phá sâu hơn để xem liệu chúng ta có thể thay đổi các loại giá trị hay không.

Cuối cùng, chúng tôi muốn tập trung vào cột thẻ cũng như mức độ phổ biến khác, vì vậy chúng tôi sẽ thực hiện như sau Các bước làm sạch dữ liệu:

- Điền các giá trị còn thiếu vào FavoriteCount bằng 0 vì các giá trị còn thiếu cho biết rằng câu hỏi không phải là bình chọn.
- Chuyển đổi FavoriteCount thành int
- Chuyển đổi chuỗi Thẻ thành định dạng dễ đọc hơn

Trong [4]:

```
df.fillna(value={"FavoriteCount":0}, inplace=True)
df['FavoriteCount'] = df['FavoriteCount'].astype(int)
```

Trong [5]:

```
df['Tags'].head(5)
```

Hết[5]:

```
0 <máy học><khai thác dữ liệu>
      <machine-learning><regression><linear-regressi...
1 2 <python><chuỗi thời gian><dự báo><dự báo>
3 <máy học><scikit-learning><pca>
      <dataset><bigdata><data><speech-to-text>
4 Tên: Thẻ, dtype: đối tượng
```

Vì có nhiều thẻ được liệt kê trong cột này, mỗi thẻ là một chuỗi nên chúng tôi có thể tạo một danh sách cho mỗi câu hỏi của các thẻ, để chúng tôi có thể dễ dàng đánh giá tần suất sử dụng bất kỳ thẻ nhất định nào.

Trong [6]:

```
df['Tags'] = df['Tags'].str.replace('><',',').str.replace('<', '')\
.str.replace('>', '')\
.str.split(',')
in(df["Tags"].sample(5))
```

```
7262              [phân loại, đa lớp-phân loại]
6307      [học máy, python, lựa chọn tính năng, ...
6551                                   [mất chức năng, softmax]
6917      [máy học, mô hình dự đoán, độ chính xác]
5604      [trăn, mô hình dự đoán, kỹ sư  tính năng...
Tên: Thẻ, dtype: đối tư ợng
```

Phân tích dữ liệu

Thẻ đư ợc sử dụng nhiều nhất và đư ợc xem nhiều nhất

Chúng tôi sẽ tập trung vào Thẻ để xác định:

- Đếm số lần mỗi thẻ đư ợc sử dụng.
- Đếm số lần mỗi thẻ đư ợc xem.
- Tạo trực quan hóa cho các thẻ trên cùng của mỗi kết quả trên

Trong [7]:

```
in(df.sample(5))
```

```
             Nhận dạng              Ngày tạo Điểm Lư ợt xem \
2247 57978 2019-08-21 17:59:44                 0            28
5703 63249 2019-11-16 13:20:13                 1            42
2016 57851 2019-08-20 08:36:15 2944
58932 2019-09-09 22:13:04                      0 0         19 332
3973 49436 2019-04-16 23:01:11                 0            27

                                                      Thẻ Trả lờiĐếm \
2247                                        [orange, orange3]                  0
5703 [phân cụm, dữ liệu, trực quan hóa, đồ họa thông tin] [bộ dữ               2
ảnh,                    liệu, làm sạch dữ liệu, đào tạo] 2016 [máy             1
2944 cnn, hàm mất mát] [hồi quy tuyến tính, tư ơ ng quan]                      1
3973                                                                           1

        Đếm yêu thích
2247                0
5703                1
2016                0
2944                0
3973                0
```

Trong [8]:

```
tag_count = dict()

cho các thẻ trong df["Tags"]:
      cho thẻ trong thẻ:
            nếu thẻ trong tag_count:
                  tag_count[thẻ] += 1
            khác:
                  tag_count[thẻ] = 1
```

Trong [9]:

```
tag_count = pd.DataFrame.from_dict(tag_count, orient='index')
tag_count.rename(columns={0:"Tag_Count"},inplace=True)
tag_count.sort_values(by="Tag_Count",ascending=False, inplace=True)
top_ten_tags = tag_count.head(10)
top_ten_tags
```

Hết[9]:

	Tag_Count
máy học	2693
con trăn	1814
học kĩ càng	1220
mạng lưới thần kinh	1055
máy ảnh	935
phân loại	685
dòng chảy căng	584
scikit-học	540
nlp	493
cnn	489

Chúng tôi vừa hiển thị tổng số cho mỗi thẻ trong khung dữ liệu câu hỏi của chúng tôi. Chúng tôi thấy, sau khi xem xét 10 thẻ được sử dụng nhiều nhất:

- Học máy là chủ đề được gắn thẻ nhiều nhất với tỷ suất lợi nhuận lớn, có hơn 1000 thẻ hơn tất cả trừ tất cả một thẻ khác
- Python đứng thứ hai, tiếp theo là Deep-Learning và Neural-Networks

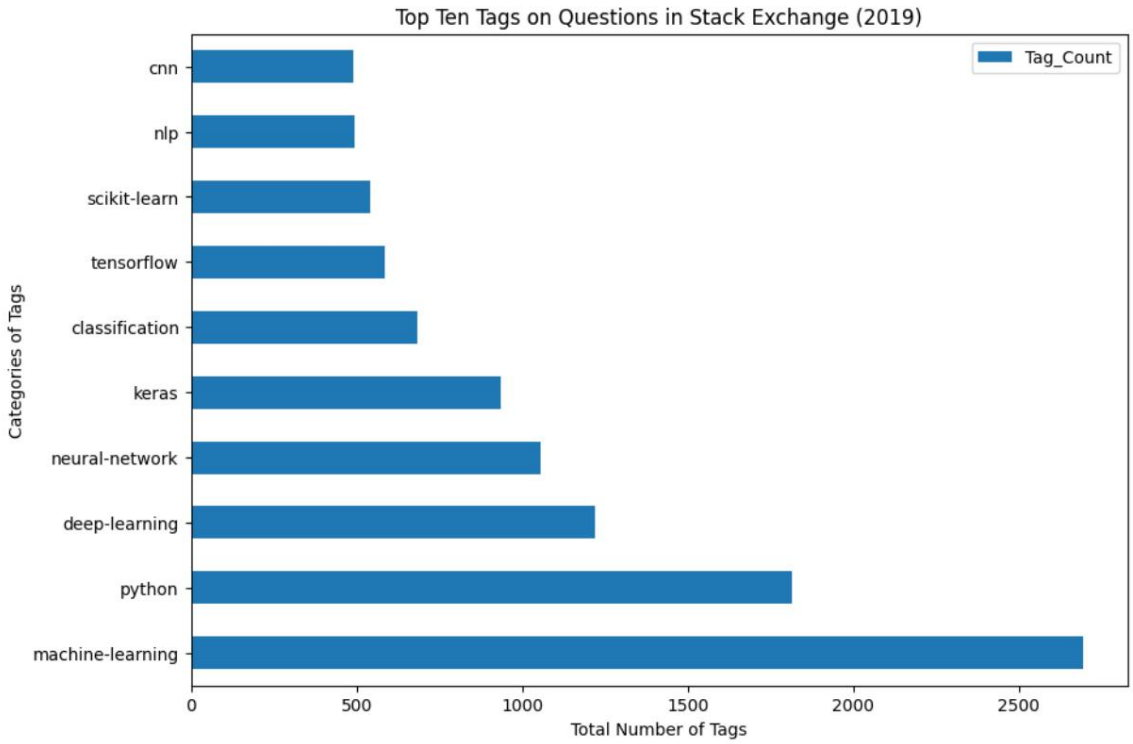
Để trực quan hóa thông tin này, chúng tôi sẽ vẽ các giá trị cùng nhau bằng biểu đồ thanh.

Trong [10]:

```
top_ten_tags.plot(kind='barh', figsize=(10,7)) plt.xlabel("Tổng
số thẻ") plt.ylabel("Danh mục thẻ")
plt.title(" Mười thẻ hàng đầu cho các câu
hỏi trong Trao đổi ngăn xếp (2019)")
```

Hết[10]:

Văn bản (0,5, 1,0, 'Mười thẻ hàng đầu cho câu hỏi trong Stack Exchange (2019)')



Đồ họa ở trên hiển thị rõ ràng hơn số lượng thẻ so với các đối tượng riêng lẻ.

Bây giờ chúng ta có thể xem danh mục nào có nhiều lượt xem nhất và xem liệu có điều gì thú vị để phân tích không thông tin đó.

Trong [11]:

```
view_count = dict()

cho chỉ mục, hàng trong df.iterrows(): cho
     thẻ trong hàng["Tags"]:
          nếu thẻ trong view_count:
               view_count[tag] += row["ViewCount"] khác:

               view_count[tag] = row["ViewCount"]

view_count = pd.DataFrame.from_dict(view_count, orient="index")
```

Trong [12]:

```
view_count.rename(columns={0: "View_Count"}, inplace=True)
view_count.sort_values(by="View_Count", Ascending=False, inplace=True) top_ten_views =
view_count.head(10)
```

Trong [13]:

```
top_ten_views
```

Hết[13]:

	Lư ợng xem
con trăn	537585
máy học	388499
máy ảnh	268608
học kĩ càng	233628
gấu trúc	201787
mạng lư ới thần kinh	185367
scikit-học	128110
dòng chảy căng	121369
phân loại	104457
khung dữ liệu	89352

Chúng tôi thấy một số điểm khác biệt khi xem nhanh tổng số lư ợt xem, bao gồm:

- Python là danh mục đư ợc xem nhiều nhất, mặc dù nó đư ợc gắn thẻ cao thứ hai và nó đã chuyển sang những nơ i có máy học
- Deep Learning và Neural Networks đều đứng sau Keras về tổng số lư ợt xem, mặc dù cả hai đều đư ợc gắn thẻ thư ơ ng xuyên hơ n. Pandas cũng có nhiều lư ợt xem hơ n Neural Network

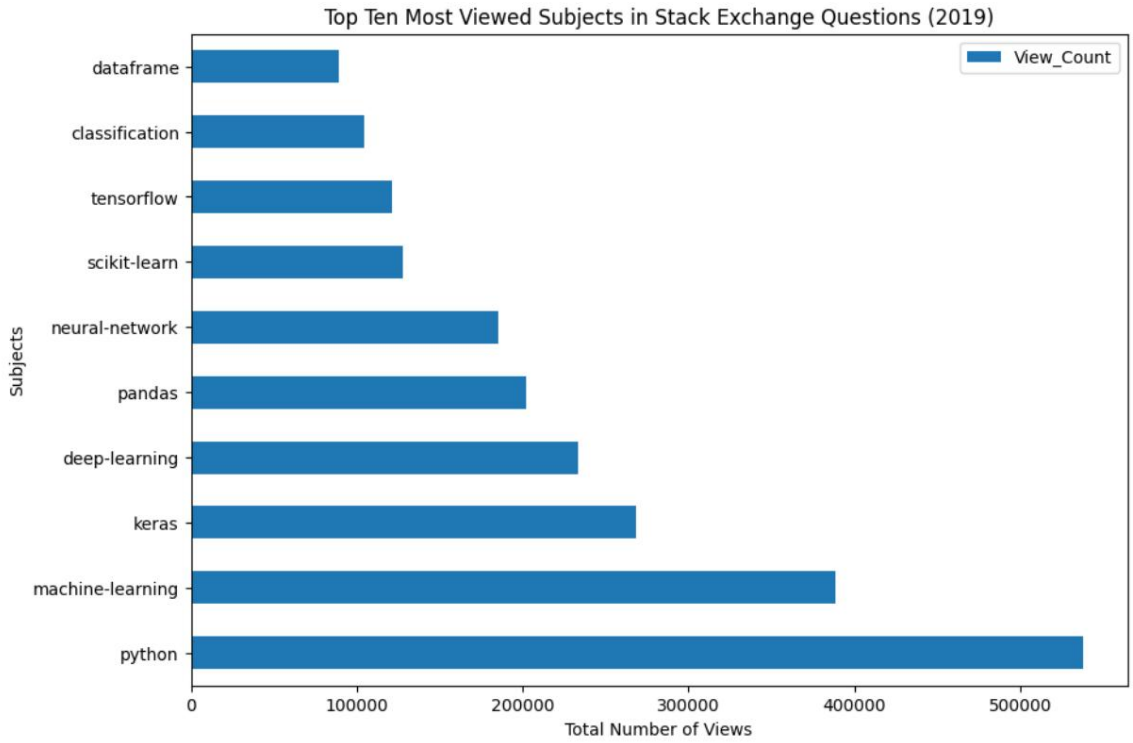
Bây giờ chúng ta sẽ trực quan hóa dữ liệu này, như chúng ta đã làm trư ớc đây, để hiểu rõ hơ n về sự khác biệt.

Trong [14]:

```
top_ten_views.plot(kind="barh", figsize=(10,7)) plt.xlabel('Tổng
số lượt xem') plt.ylabel("Chủ đề")
plt.title("Mười chủ đề được
xem nhiều nhất trong Stack Exchange Câu Hỏi (2019)")
```

Hết[14]:

Văn bản (0,5, 1,0, 'Mười chủ đề được xem nhiều nhất trong câu hỏi trao đổi ngăn xếp (2019)')



Chúng ta thấy rằng biểu đồ trông tương tự như biểu đồ đầu tiên, với hai đối tượng cao nhất về lượt xem cao hơn đáng kể so với các đối tượng còn lại.

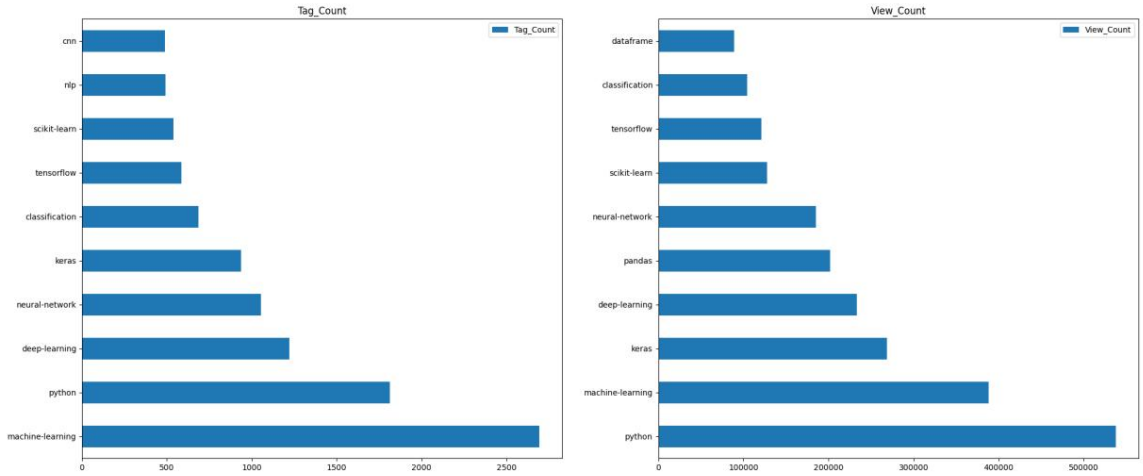
Nếu đặt hai biểu đồ cạnh nhau, chúng ta có thể thấy rõ hơn cách các lượt xem và thẻ xếp chồng lên nhau.

Trong [15]:

```
fig,axes = plt.subplots(nrows=1,ncols=2)
fig.set_size_inches((24, 10))
top_ten_tags.plot(kind="barh", ax=axes[0], subplots=True)

top_ten_views.plot(kind="barh", ax=axes[1], subplots=True)
```

Hết[15]:

```
mảng([<Axes: title={'center': 'View_Count'}>], dtype=object)
```



Những gì chúng ta thấy từ hai biểu đồ trên là có một mối quan hệ khá rõ ràng giữa các chủ đề được gắn thẻ thường xuyên hơn trong các câu hỏi và những chủ đề nhận được nhiều lượt xem hơn. Điều này có ý nghĩa, vì số lượng thẻ cho câu hỏi càng cao thì khả năng chủ đề sẽ được xem càng cao. Tuy nhiên, điều thú vị là Machine Learning được gắn thẻ trong hơn 800 câu hỏi nhiều hơn Python, nhưng các câu hỏi với python đã được xem 15.000
nhiều lần hơn.

Kết quả

Điều chúng tôi cũng nhận thấy là các chủ đề như Mạng thần kinh và Học sâu đều được gắn thẻ thường xuyên, giống như Học máy. Hóa ra là tất cả các chủ đề này đều có liên quan với nhau, điều này có thể chiếm tổng số thẻ cao hơn cho Học máy. Có khả năng là mỗi khi một trong số Mạng thần kinh hoặc Học sâu được gắn thẻ, thì Học máy cũng được gắn thẻ.

Tương tự, Keras có liên quan đến cả Python, Mạng thần kinh và Học sâu, nghĩa là tất cả chúng đều có khả năng được gắn thẻ cùng nhau thường xuyên. Vì các chủ đề có thẻ Python được xem thường xuyên, Keras có khả năng cũng được gắn thẻ cho nhiều câu hỏi trong số này, dẫn đến giá trị cao hơn cho các lượt xem của nó so với vị trí của nó trong các thẻ.

Điều chúng tôi nhận thấy là các chủ đề về Mạng thần kinh và Học sâu dường như rất phổ biến vào lúc này.

Bây giờ, chúng ta sẽ xem xét liệu Deep Learning vẫn tồn tại hay chỉ phổ biến ở thời điểm hiện tại.

Deep Learning, một "mốt nhất thời" hay "ở đây để ở lại"?

Trong [16]:

```
all_q = pd.read_csv('all_questions.csv',parse_dates=["CreationDate"])
```

Trong [17]:

```
all_q.head()
```

Hết[17]:

	Id	Ngày thành lập	thẻ
0	12954	2016-07-23 07:05:30	<python><clustering><unsupervised-learning>
1	12956	2016-07-23 09:46:42	<học sâu><rnn><chuẩn hóa><định mức hàng loạt...
2	12958	2016-07-23 12:34:34	<phân loại><phân cụm><thống kê><quý...
3	12959	2016-07-23 21:02:17	<machine-learning><markov- process><âm thanh-nhận...
4	12960	2016-07-23 22:33:04	<khai thác văn bản><khai thác tính năng><văn bản>

Giống như trước đây, chúng tôi sẽ phải dọn dẹp cột Thẻ một chút để có thể phân biệt các thẻ riêng lẻ.

Trong [18]:

```
all_q['Tags'] = all_q['Tags'].str.replace('><',',').str.replace('<', ' ')\ .str.replace('>', ' ')
 \ .str.split(',')
```

Bây giờ chúng tôi sẽ cố gắng lọc các thẻ của mình để chỉ những thẻ liên quan đến Deep Learning. Điều này bao gồm ["lstm", "cnn", "scikit-learning", "tensorflow", "keras", "neural-network", "deep-learning"].

Để làm điều này, chúng tôi sẽ tạo một phần tách trong các thẻ và tạo một cột mới trong khung dữ liệu của chúng tôi.

Trong 19]:

```
dl_set = ["lstm", "cnn", "scikit-learning", "tensorflow", "keras", "neural-network",
             "học sâu"] def
deep_learning(tags): cho thẻ
     trong thẻ: nếu thẻ
           trong dl_set: trả về 1

           khác:
                trả về 0
```

Trong 20]:

```
all_q["Deep_Learning"] = all_q["Tags"].apply(deep_learning)
all_q.head()
```

Hết[20]:

	Ngày thành lập		Thẻ Deep_Learning
0	12954	2016-07-23 07:05:30 [trăn, phân cụm, học không giám sát]	0
1	12956	2016-07-23 09:46:42 [học sâu, rnn, chuẩn hóa, định mức hàng loạt...	1
2	12958	2016-07-23 12:34:34 [phân loại, phân cụm, thống kê, missi...	0
3	12959	2016-07-23 21:02:17 [học máy, xử lý markov, nhận dạng âm thanh...	0
4	12960	2016-07-23 22:33:04 [khai thác văn bản, trích xuất tính năng, văn bản]	0

Tiếp theo, chúng tôi sẽ phân chia thông tin dựa trên quý của mỗi năm, bắt đầu từ bây giờ và ngược lại đúng giờ. Chúng ta nên lưu ý rằng quý của các mục nhập gần đây nhất có thể không đầy đủ, vì vậy để điều này, chúng tôi sẽ kết thúc thông tin của chúng tôi kể từ năm 2019.

Trong [21]:

```
all_q = all_q[all_q["CreationDate"].dt.year < 2020]
```

Trong [22]:

```
quý chắc chắn (thời gian):
    năm = str(time.year)
    quý = str(((thời gian.tháng-1))//3+1)
    năm trở lại +"Q"+quý

all_q["Quý"] = all_q["Ngày tạo"].apply(quý)
all_q.head()
```

Hết[22]:

	Ngày thành lập		Thẻ Deep_Learning	quý
0	12954	23-07-2016 07:05:30	[trăn, phân cụm, học không giám sát]	0 2016Q3
1	12956	23-07-2016 09:46:42	[học sâu, rnn, chuẩn hóa, hàng loạt định mức...	1 Quý 3 năm 2016
2	12958	23-07-2016 12:34:34	[phân loại, phân cụm, thống kê, cô...	0 2016Q3
3	12959	23-07-2016 21:02:17	[máy học, quy trình markov, nhận dạng âm thanh...	0 2016Q3
4	12960	23-07-2016 22:33:04	[khai thác văn bản, trích xuất tính năng, văn bản]	0 2016Q3

Mức độ phổ biến hàng quý của Deep Learning

Trong [23]:

```
câu hỏi_per_quý = all_q.groupby("Quý").agg({"Deep_Learning":['sum','size']})
câu hỏi_per_quý.head()
```

Hết[23]:

	Học kĩ càng	
	kích thư ớc tổng	
Một phần tư		
2014Q2	3	157
2014Q3	4	189
2014Q4	4	214
2015Q1	5	190
2015Q2	6	284

Trong [24]:

```
câu hỏi_per_quý.columns = ['DL_Questions', 'Total_Questions']
câu hỏi_per_quý.head()
```

Hết[24]:

	DL_Câu hỏi	Total_Câu hỏi
Một phần tư		
2014Q2	3	157
2014Q3	4	189
2014Q4	4	214
2015Q1	5	190
2015Q2	6	284

Trong [25]:

```
câu hỏi_per_quý['DL_percentage'] = câu hỏi_per_quý['DL_Questions']/câu hỏi_
câu hỏi_per_quý.head(10)
```

Hết[25]:

Một phần tư	DL_Câu hỏi	Tổng_Câu hỏi	DL_percentage
2014Q2	3	157	0,019108
2014Q3	4	189	0,021164
2014Q4	4	214	0,018692
2015Q1	5	190	0,026316
2015Q2	6	284	0,021127
2015Q3	13	311	0,041801
2015Q4	19	382	0,049738
2016Q1	38	516	0,073643
2016Q2	45	517	0,087041
2016Q3	70	584	0,119863

mã ở trên: question_per_quý['DL_percentage'] =
câu hỏi_per_quý['DL_Questions']/câu hỏi_per_quý["Total_Questions"]

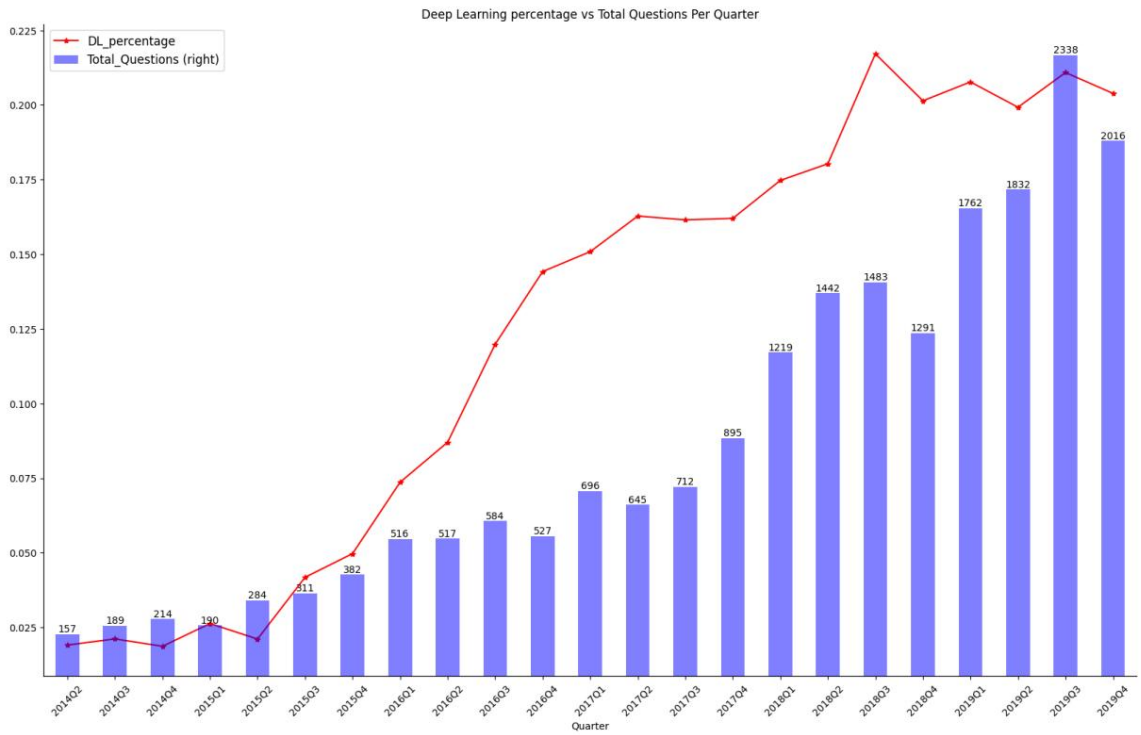
Trong [26]:

```
câu hỏi_per_quý.reset_index(inplace=True)
```

Trực quan hóa dữ liệu

Trong [27]:

```
ax1 = question_per_ quarter.plot(x="Quarter",y="DL_percentage",kind="line", linestyle = ax2 = question_per_
quarter.plot(x="Quarter",y="Total_Questions", kind="bar", ax=ax1, giây

cho idx, t trong question_per_quý["Total_Questions"].items(): ax2.text(idx, t,
     str(t), ha="center", va="bottom") xlims = ax1.get_xlim()


ax1.get_legend().remove()

xử lý1, nhãn1 = ax1.get_legend_handles_labels() xử lý2, nhãn2 =
ax2.get_legend_handles_labels() ax1.legend(handles=handles1 +
handles2, nhãn=nhãn1 + nhãn2, loc="phía trên bên
              trái", prop={"size": 12} )


cho ax in (ax1,ax2): cho
     loc in ("top", "right"):
          ax.spines[loc].set_visible(False)
          ax.tick_params(right=False, labelright = False)

plt.title(" Tỷ lệ phần trăm Deep Learning so với Tổng số câu hỏi mỗi quý")

plt.show()
```



Suy nghĩ cuối cùng

Biểu đồ trên cho thấy mức độ phổ biến tăng rõ rệt bắt đầu từ giữa năm 2015, với tỷ lệ phần trăm câu hỏi được hỏi tăng đều đặn cho đến giữa năm 2018, trong đó tỷ lệ này ở mức khoảng 21% hoặc tất cả các câu hỏi được hỏi.

Tôi tin rằng sự gia tăng mức độ phổ biến có liên quan nhiều hơn đến chủ đề tương đối mới và do đó có nhiều chỗ để khám phá hơn. Kể từ năm 2018, cứ 5 câu hỏi thì nó vẫn giữ nguyên 1 câu hỏi, cho thấy rằng vẫn còn rất nhiều việc phải làm trong lĩnh vực này bởi những cá nhân muốn nghiên cứu hoặc làm việc trong lĩnh vực Học sâu.

Tôi tin rằng Deep Learning sẽ duy trì mức độ phổ biến tương tự trong một vài năm nữa, sau đó sẽ giảm xuống nếu có điều gì đó khả thi hơn xuất hiện hoặc tăng hơn nữa nếu có những cải tiến lớn trong lĩnh vực này.

TRONG [ ]: