Tin tặc ¶

Hacker News là một trang web đư ợc thành lập bởi vư ờn ư ơ m khởi nghiệp Y Combinator, nơ i các câu chuyện do ngư ời dùng gửi (đư ợc gọi là "bài đăng") nhận phiếu bầu và nhận xét, tư ơ ng tự như reddit. Hacker News cực kỳ phổ biến trong giới công nghệ và khởi nghiệp, và kết quả là các bài đăng lọt vào đầu danh sách Hacker News có thể nhận đư ợc hàng trăm nghìn khách truy cập.

Dữ liệu

Dữ liệu nguồn cho nghiên cứu này có thể đư ợc tìm thấy ở đây (https://www.kaggle.com/hacker-news/hacker news-posts). Nó chứa gần 300.000 hàng, mỗi hàng đại diện cho một bài đăng. Dữ liệu là của năm 2016. Tuy nhiên, đối với nghiên cứu này, chúng tôi sử dụng phiên bản đã giảm xuống còn khoảng 20.000 hàng bằng cách xóa tất cả các nội dung gửi không nhận đư ợc bất kỳ nhận xét nào, sau đó lấy mẫu ngẫu nhiên từ các nội dung gửi còn lại. Tệp này do Dataquest chuẩn bị và có thể tải xuống từ đây (https://app.dataquest.io/m/356/guided-project%3A-exploring hacker-news-posts/1/introduction).

Hãy để chúng tôi bắt đầu với việc đọc dữ liệu và hiển thị hàng tiêu đề và một mẫu nhỏ.

- id: mã định danh duy nhất từ Hacker News cho tiêu đề bài
- đăng: tiêu đề của bài
- đăng url: URL mà bài đăng liên kết đến, nếu bài đăng có URL
- num_points: số điểm mà bài đăng có đư ợc, đư ợc tính bằng - - tổng số số lư ợng upvote trừ tổng số downvote num_comments: số lư ợng nhận xét về
- bài viết tác giả: tên ngư ời dùng của ngư ời đã gửi bài
- viết created_at: ngày và thời gian gửi bài viết
-

- Trung bình Ask HN hay Show HN nhận đư ợc nhiều comment hơ n?

- Trung bình các bài đăng đư ợc tạo vào một thời điểm nhất định có nhận đư ợc nhiều bình luận hơ n không?

Mở tập dữ liệu

Trong [1]:
```
từ trình đọc nhập csv # nhập mô-đun CSV vào
open_file=open('hacker_news.csv', encoding='UTF-8')
read_file=reader(open_file)
hacker_news_full=danh sách(read_file)                    # Toàn bộ tập dữ liệu trong Li

header=hacker_news_full[0]                               # Lấy tiêu đề của d
hn_full=hacker_news_full[1:] print("                              # Toàn bộ tập dữ liệu
Số hàng trong tập dữ liệu: ", len(hn_full) )
```

```
Số hàng trong tập dữ liệu: 20100
```

Chức năng tìm Chỉ mục cùng với Tên cột

Trong [58]:
```
bộ chỉ mục def (data_header, variable_name):                    # Nhập tiêu đề tập dữ liệu
    chỉ số = 0
    print("Chỉ mục của {}".format(tên_biến)) cho tên_cột          # in biến n
    trong dataset_header: print(index, ' index+=1                 # tìm từng tên cột
                          : ', tên cột)


    in('\n')

bộ chỉ mục (tiêu đề, " Dữ liệu mới của Hacker")
```

```
Chỉ mục dữ liệu mới của Hacker
0 : id
1 : tiêu đề
2 : url
3 : số_điểm
4 : số_bình_luận
5 : tác giả
6 : created_at
```

"Được rồi, vậy là có 'num-points' ở chỉ mục 3 và 'num_comments' ở chỉ mục 4. Bây giờ tôi sẽ tìm xem có bao nhiêu hàng trong tập hợp dữ liệu này thỏa mãn các điều kiện để có nhận xét và điểm."

Xóa các hàng không có nhận xét và điểm

Trong [3]:
```
hn=[]                                                    # Tạo một danh sách trống để lưu trữ

cho hàng trong hn_full:                                  # Lặp qua từng hàng o
     comments=int(row[4])
     points=int(row[3]) nếu
     comments >0 và point>0: hn.append(row)              # Áp dụng điều kiện để tạo


print("Độ dài của tập dữ liệu mới là ", len(hn))
```

```
Độ dài của tập dữ liệu mới là 20100
```

Trong [4]:
```
def explorer_data(tập dữ liệu, bắt đầu, kết thúc, rows_and_columns=False):
     dataset_slice = dataset[start:end] cho hàng
     trong dataset_slice: print(row)
          print('\n')


     if rows_and_columns:
          print('Số hàng:', len(tập dữ liệu)) print('Số
          cột:', len(tập dữ liệu[0]))
```

Trong [5]:
```
in (tiêu đề)
```

```
['id', 'title', 'url', 'num_points', 'num_comments', 'author', 'created_at']
```

Tạo danh sách riêng biệt cho các bài đăng Hỏi, Hiển thị và Khác

Hàm startedwith() sẽ được sử dụng để tìm xem chuỗi có bắt đầu với lát chuỗi đã cho hay không. Ở đây trong trường hợp này, nếu nó bắt đầu bằng 'hỏi hn' hoặc 'hiển thị hn'. Để chuẩn hóa chuỗi, nó sẽ được chuyển đổi thành chữ thường nhỏ hơn bằng cách sử dụng chức năng thấp hơn.

Trong [6]:
```
ask_posts = []
show_posts = []
other_posts = []

cho hàng ở hn:
    title = row[1].lower() if
    title.startswith('ask hn'):
        ask_posts.append(row) elif
    title.startswith('show hn'):
        show_posts.append(row) khác:

        other_posts.append(hàng)
```

Trong [7]:
```
explorer_data(ask_posts,0,2,True)
```

['12296411', 'Hỏi HN: Làm cách nào để cải thiện trang web cá nhân của tôi?', '', '2', '6', 'ah medbaracat', '8/16/2016 9:55']

['10610020', 'Hỏi HN: Tôi có phải là ngư ời duy nhất bị xúc phạm bởi Twitter tắt tính năng sha re không?', '', '28', '29', 'tkfx', '22/11/2015 13:43']

Số hàng: 1744
Số cột: 7

Trong [8]:
```
print("No. of Ask posts", len(ask_posts))                                    # in ra num print("No.
of Show posts", len(show_posts)) # in ra nu print("No. of Other posts", len(other_posts)) # in ra n
print("Tổng thời lư ợng tin tức Hacker ", len(hn)) # kiểm tra bản in("Total",len(ask_posts)+len(show_posts)
+len(other_posts)) # so sánh wi
```

Số bài hỏi 1744 Số bài hiển thị 1162 Số bài khác 17194
Tổng thời lư ợng tin tức Hacker 20100 Tổng 20100

Tìm số bình luận trung bình và điểm

Trong [18]:
```
total_ask_comments = 0 cho
hàng trong ask_posts:
    num_comments = int(row[4])
    total_ask_comments += num_comments
avg_ask_comments = total_ask_comments / len(ask_posts)


print("Tổng số bình luận cho các bài viết Hỏi: ", total_ask_comments)
print('Số lư ợng bình luận trung bình cho các bài viết Hỏi: {:.4f}'.format(avg_ask_comments))
```

Tổng số bình luận cho bài hỏi: 24483 Số bình luận trung bình cho bài hỏi: 14,0384

Trong [19]:
```
total_show_comments = 0
cho hàng trong show_posts:
      num_comments = int(row[4])
      total_show_comments += num_comments
avg_show_comments = total_show_comments / len(show_posts)

print("Tổng số bình luận cho Show Posts: ", total_show_comments) print('Số bình
luận trung bình cho Show posts: {:.4f}'.format(avg_show_comments)
```

```
Tổng số bình luận cho Hiển thị bài viết: 11988
Số bình luận trung bình cho Hiển thị bài viết: 10,3167
```

Có vẻ như trung bình các bài đăng 'hỏi' nhận được nhiều bình luận hơn các bài đăng 'hiển thị'.

Để phân tích xem các thời điểm cụ thể trong ngày có thu hút nhiều bình luận hơn hay không, chúng tôi sẽ tiếp tục với các bài đăng "hỏi" này. Chúng tôi có thể kiểm tra thời gian nào là thời gian tốt nhất để đăng bài? Ví dụ: nếu có nhiều người dùng trực tuyến hơn tại một thời điểm cụ thể, thì sẽ có nhiều lượt ủng hộ hơn trong thời gian đó. Vì vậy, nếu chúng ta có thể tìm ra thời điểm tốt nhất để đăng bài, điều đó sẽ rất tuyệt.

Tìm số lượng bài hỏi và bình luận theo giờ Tạo

Trong [13]:
```
nhập ngày giờ dưới dạng dt                    # nhập mô-đun ngày giờ để làm việc với ngày & # Tạo danh sách
chứa thời gian tạo và số nhận xét (ask-p result_list = [] #a danh sách các danh sách cho hàng trong
ask_posts: created_at = row[6] num_comments = int( hàng[4])
result_list.append([created_at,
      num_comments])



in(result_list[0:3])
```

```
[['16/8/2016 9:55', 6], ['22/11/2015 13:43', 29], ['2/5/2016 10:14', 1]]
```

Trong [14]:
```
# Xây dựng bảng tần suất cho số lượng bài đăng và số lượng bình luận count_by_hour ={}
comments_by_hour = {}                  # số lượng bài đăng mỗi giờ trong ngày
cho hàng trong result_list:             # số lượng bình luận mỗi giờ trong ngày

     created_at = dt.datetime.strptime(row[0], '%m/%d/%Y %H:%M') created_at_hour
     = created_at.strftime('%H') nếu created_at_hour theo
     số_theo_giờ:
          counts_by_hour[created_at_hour] += 1
          comments_by_hour[created_at_hour] += row[1] other:

          counts_by_hour[created_at_hour] = 1
          comments_by_hour[created_at_hour] = row[1]

print("Đây là Số lượng bài viết trong mỗi giờ \n \n", counts_by_hour) print("\n Đây là
Số lượng bình luận trong mỗi giờ \n \n", comments_by_hour
```

Đây là Số lượng bài viết trong mỗi giờ

{'09': 45, '13': 85, '10': 59, '14': 107, '16': 108, '23': 68, '12': 73, '1 7': 100, '15': 116, '21': 109, '20': 80, '02': 58, '18': 109, '03': 54, '05': 46, '19': 110, '01 ': 60, '22': 71, '08': 48, '04': 47, '00': 55, '06': 44, '0 7': 34, '11': 58}

Đây là Số lượng bình luận trong mỗi giờ

{'09': 251, '13': 1253, '10': 793, '14': 1416, '16': 1814, '23': 543, '12': 687, '17': 1146, ' 15': 4477, '21': 1745, '20': 1722, '02': 1381, '18': 1439, '03': 421, '05': 464, '19': 1188, '01' : 683, '22': 479, '08': 492, '04': 337, '00': 447, '06': 397, '07': 267, '11': 641}

Phương pháp thứ hai của nhiệm vụ trên:

Trong [21]:
```
# Tạo từ điển với Giá trị là Giờ làm Khóa và Số Nhận xét.


nhập ngày giờ dư ới dạng dt             # nhập mô-đun ngày giờ để làm việc với ngày &



hourly_comment={}                          # Làm trống từ điển để lư u trữ số lư ợng bình luận
hourly_post= {}                             # Từ điển trống để lư u trữ số lư ợng bài viết

cho hàng trong ask_posts:                    # Lặp qua từng hàng của ask post li
     time_string = row[6]                     # chỉ định hàng[6], "đư ợc tạo tại" cho thời điểm g
     comments_number = int(row[4])             # Gán giá trị nguyên của số o


          # Chuyển đổi thời gian đã cho trong Chuỗi thành đối tư ợng ngày giờ. Thời gian lấy mẫu là g
          # Đây là định dạng tháng/ngày/4 chữ số Năm Giờ: Phút --> %m


     convert_time = dt.datetime.strptime(time_string,"%m/%d/%Y %H:%M")
     giờ_posted = convert_time.strftime("%H") # Gán đối tư ợng ngày giờ


          # Tạo bảng tần số bằng Từ điển

     nếu giờ_đăng trong giờ_bình luận: bình luận                          # nếu giờ_đăng là
          hàng giờ[giờ_đăng] += số_bình luận hàng giờ_đăng[giờ_đăng]       # Cộng commen
          += 1                                                               # Thêm 1 vào số

     khác:                                                                  # nếu nó không phải là pres
          hourly_comment[hour_posted] = comments_number                    # Gán giá trị
          hourly_post[hour_posted] = 1                                   # Gán giá trị như


print("Đây là Số lư ợng bài viết trong mỗi giờ \n \n", hourly_post)
print("\n Đây là Số bình luận trong mỗi giờ \n \n", hourly_comment)
```

```
Đây là Số lư ợng bài viết trong mỗi giờ

 {'09': 45, '13': 85, '10': 59, '14': 107, '16': 108, '23': 68, '12': 73, '1
7': 100, '15': 116, '21': 109, '20': 80, '02': 58, '18': 109, '03': 54, '05':
46, '19': 110, '01': 60, '22': 71, '08': 48, '04': 47, '00': 55, '06': 44, '0
7': 34, '11': 58}

 Đây là Số lư ợng bình luận trong mỗi giờ

 {'09': 251, '13': 1253, '10': 793, '14': 1416, '16': 1814, '23': 543, '12':
687, '17': 1146, '15': 4477, '21': 1745, '20': 1722, '02': 1381, '18': 1439,
'03': 421, '05': 464, '19': 1188, '01': 683, '22': 479, '08': 492, '04': 337,
'00': 447, '06': 397, '07': 267, '11': 641}
```

Bây giờ tôi có hai từ điển với số lư ợng bình luận và số lư ợng bài đăng đối với mỗi giờ, tôi có thể ngay lập tức tìm thấy số lư ợng bình luận trung bình đư ợc đăng trong mỗi giờ.

Tìm Nhận xét Trung bình cho mỗi bài đăng trong Mỗi Giờ

Một danh sách sẽ đư ợc tạo để liệt kê giờ và số lư ợng bài đăng trung bình trong mỗi giờ. Để dễ sắp xếp, chúng tôi sẽ giữ cột đầu tiên là số lư ợng bài đăng trung bình mỗi giờ.

Trong [69]:
```
# Tạo bảng chứa số giờ trong ngày và số bình luận trung bình của comm avg_comments_by_hour = []
avg_by_hour_rev = []


trong giờ, tính bằng counts_by_hour.items():
     avg_comments = comments_by_hour[hour] / đếm
     avg_comments_by_hour.append([giờ, avg_comments])
     avg_by_hour_rev.append([avg_comments, giờ])

# Sắp xếp danh sách (trên phần tử đầu tiên, là giờ trong ngày)
avg_comments_by_hour.sort()

# In kết quả ra =
"Trong giờ {}, số bình luận trung bình trên mỗi bài đăng là {:.2f}" cho hàng trong
avg_comments_by_hour: print
     (output.format(row[0], row[1]))
```

```
Đối với giờ 00 số bình luận trung bình trên mỗi bài đăng là 8,13 Đối với
giờ 01 số bình luận trung bình trên mỗi bài đăng là 11,38 Đối với giờ 02 số
bình luận trung bình trên mỗi bài đăng là 23,81 Đối với giờ 03 số bình luận
trung bình trên mỗi bài đăng là 7,80 Đối với giờ 00 04 số lư ợng bình luận
trung bình trên mỗi bài đăng là 7,17 Đối với giờ 05 số lư ợng bình luận
trung bình trên mỗi bài đăng là 10,09 Đối với giờ 06 số lư ợng bình luận
trung bình trên mỗi bài đăng là 9,02 Đối với giờ 07 số lư ợng bình luận
trung bình trên mỗi bài đăng là 7,85 Đối với giờ 08 số lư ợng bình luận
trung bình trên mỗi bài đăng là 7,85 số lư ợng bình luận trung bình trên
mỗi bài đăng là 10,25 Đối với giờ 09 số lư ợng bình luận trung bình trên
mỗi bài đăng là 5,58 Đối với giờ 10 số lư ợng bình luận trung bình trên mỗi
bài đăng là 13,44 Đối với giờ 11 số lư ợng bình luận trung bình trên mỗi
bài đăng là 11,05 Đối với giờ 12 số lư ợng bình luận số bình luận trên mỗi
bài đăng là 9,41 Đối với giờ 13 số bình luận trung bình trên mỗi bài đăng
là 14,74 Đối với giờ 14 số bình luận trung bình trên mỗi bài đăng là 13,23
Đối với giờ 15 số bình luận trung bình trên mỗi bài đăng là 38,59 Đối với
giờ 16 số bình luận trung bình mỗi bài đăng là 16,80 Trong giờ thứ 17, số
lư ợng bình luận trung bình trên mỗi bài đăng là 11,46 Trong giờ thứ 18, số
lư ợng bình luận trung bình trên mỗi bài đăng là 13,20 F bfti 10 80
     h      ngày 19                                    t
```

Sắp xếp các bình luận trung bình trên mỗi bài đăng mỗi giờ

Bây giờ tôi có một danh sách các bình luận trung bình cho mỗi bài đăng trong mỗi giờ, tôi có thể sắp xếp danh sách đó để xem số lư ợng bình luận đư ợc thực hiện nhiều nhất trên mỗi bài đăng vào giờ nào. Để sắp xếp danh sách, bây giờ tôi sẽ sử dụng hàm sorted() theo thứ tự giảm dần.

Trong [70]:
```
sorted_avg = đã sắp xếp(avg_by_hour_rev, reverse=True)
print ("Số trung bình đư ợc sắp xếp cho mỗi bài đăng mỗi giờ ở đây \n")
sorted_avg
```

```
Trung bình đư ợc sắp xếp trên mỗi bài đăng mỗi giờ ở đây


Ra[70]: [[38.5948275862069, '15'],
 [23.810344827586206, '02'],
 [21.525, '20'],
 [16.796296296296298, '16'],
 [16.009174311926607, '21'] ,
 [14.741176470588234,' 13'],
 [13.440677966101696, '10'],
 [13.233644859813085, '14'],
 [13.20183486238532, '18'], [11.46,
 '17'],
 [11.3833333333333333, '01'],
 [11.051724137931034, '11' ], [10.8,
 '19'], [10.25,
 '08'],
 [10.08695652173913, '05'],
 [9.41095890410959, '12'], [9
 022727272727273 '06']
```

Làm báo cáo ngắn

Từ danh sách này, tôi có thể nói rằng 3 giờ chiều, 2 giờ sáng, 8 giờ tối là 3 giờ tốt nhất để đăng để tạo ra nhiều bình luận nhất theo dữ liệu này. Tiếp theo, chúng tôi sẽ viết một số mã để tạo một loại báo cáo. Vì vậy, tôi sẽ sử dụng một số kỹ thuật định dạng.

Trong [72]:
```
# Tạo danh sách đư ợc sắp xếp theo số lư ợng bình luận trung bình thay vì swap_avg_by_hour = []
cho hàng trong
avg_comments_by_hour:
     swap_avg_by_hour.append([row[1], row[0]])
in(swap_avg_by_hour)

# Đã tạo phiên bản đư ợc sắp xếp của danh sách này
sorted_swap = sorted(swap_avg_by_hour, reverse = True)

# Hiển thị kết quả print('\n
5 giờ hàng đầu cho bình luận về bài đăng đư ợc hỏi', '\n') output =
"{}: {:.2f} bình luận trung bình trên mỗi bài đăng" cho hàng
trong sorted_swap[:5]:
     thetime = dt.datetime.strptime(str(row[1]), '%H') thetime =
     thetime.strftime('%H:%M') print
     ( output.format(thetime,row[0] ))
```

```
[[8.127272727272727, '00'], [11.383333333333333, '01'], [23.810344827586206, '02'], [7.796296296296297,
 '03'], [7.170212765957447, '04'], [10.086956521739 13, '05'], [9.022727272727273, '06'],
[7.852941176470588, '07'], [10.25, '0 8'], [5.5777777777777775, '09'], [13.440677966101696, '10'],
[11.05 1724137931 034, '11'], [9.41095890410959, '12'], [14.741176470588234, '13'], [13.2336448
59813085, '14'], [38.5948275862069, '15'], [16.796296296296298,' 16'], [11.4 6, '17'],
[13.20183486238532, '18'], [10.8, '19'], [21.525, '20'], [16.009174 311926607, '21'],
[6.746478873239437, '22'], [7.985294117647059, '23']]]
```

5 giờ hàng đầu cho các bài đăng hỏi Bình luận

15:00: 38,59 bình luận trung bình mỗi bài đăng
02:00: 23,81 bình luận trung bình mỗi bài đăng
20:00: 21,52 bình luận trung bình mỗi bài đăng
16:00: 16,80 bình luận trung bình mỗi bài đăng
21:00: 16,01 bình luận trung bình mỗi bài đăng

phư ơ ng pháp thứ hai:

Trong [23]:
```
cho hàng trong sorted_avg[:5]:                                                    # lặp lại

    time = dt.datetime.strptime(row[1],"%H").strftime("%I %p")                    # chuyển đổi
        # %I hiển thị Giờ ở định dạng 12 giờ, %p hiển thị AM hoặc PM

    trung bình = hàng[0]                                                          # Chỉ định

        # In bằng phư ơ ng pháp .format. Đối số this_time đư ợc lư u trữ tại thời điểm # this_avg đư ợc
        định dạng có 2 dấu thập phân

    print(" Nếu bạn đăng vào {this_time}, bạn có cơ  hội nhận đư ợc {this_avg
            .format(this_time=time,this_avg=avg))


    # In kết luận của chúng tôi

print("Vì vậy, thời điểm tốt nhất để đăng bài để có đư ợc lực kéo tốt là {} EST".format(
    dt.datetime.strptime(sorted_avg[0][1],"%H").strftime("%I %p")))
```

Nếu bạn đăng lúc 03 giờ chiều, bạn có cơ hội nhận đư ợc 38,59

Nếu bạn đăng lúc 02 giờ sáng, bạn có cơ hội nhận đư ợc 23,81

Nếu bạn đăng lúc 08 giờ tối, bạn có cơ hội nhận đư ợc 21,52

Nếu bạn đăng lúc 04 giờ chiều, bạn có cơ hội nhận đư ợc 16,80

Nếu bạn đăng lúc 09:00, bạn có cơ hội nhận đư ợc 16.01

Vì vậy, thời điểm tốt nhất để đăng để có đư ợc lực kéo tốt là 03 PM EST

Kết luận nhỏ

Tôi nghĩ rằng chúng tôi đã đi đến kết luận về thời gian để đăng. Từ dữ liệu chúng tôi đã phân tích, đó là dữ liệu đư ợc thu thập trong năm 2016, chúng tôi đã thu thập một tập hợp con dữ liệu có nhận xét và điểm.

Từ bộ dữ liệu đó, tôi có thể nói với bạn rằng việc đăng bài trong danh mục Hỏi HN có thể tạo ra một tư ơ ng tác dẫn đến số lư ợng bình luận nhiều hơ n.

Như ng nếu bạn muốn có thêm bình luận ngay lập tức, tôi đã liệt kê một số thời điểm tốt hơ n để đăng. 3 giờ chiều, 2 giờ sáng, 8 giờ tối, 4 giờ chiều và 9 giờ tối là những thời điểm đó. Trong số này 3 giờ chiều là thời gian tốt nhất theo chúng tôi phân tích với 38,59 là bình luận trung bình trên mỗi bài đăng mỗi giờ.

Tóm lại, nếu bạn muốn trở nên nổi tiếng trên Hacker News, bạn cần tìm những bài viết bạn có thể gửi trên Ask HN Category và đăng chúng lúc 3 giờ chiều EST

Chúng tôi sẽ làm việc với dữ liệu này nhiều hơ n một chút và cố gắng tìm xem có bất kỳ kết nối nào với các tác giả, nếu ai đó đang làm tốt hơ n những ngư ời khác

Mối quan hệ giữa Tác giả và số lượng bình luận

Trong lúc đó, tôi thực sự quan tâm đến nó và bắt đầu đào sâu để tìm ra một số loại tương quan giữa các tác giả và số lượng bình luận. Vì vậy, tôi đã tạo ra một từ điển để tìm phân chia bình luận giữa các tác giả.

Trong [36]:
```
# Tạo Tác giả - Phân phối nhận xét

tác giả={}                                          # Tạo một từ điển trống để lưu trữ
cho hàng trong ask_posts:                            # lặp qua ask_posts
     tên=hàng[5]                                      # chỉ định tên của các tác giả
     nhận xét=int(hàng[4])                            # chỉ định số lượng bình luận
     nếu tên trong tác giả:                           # Kiểm tra xem có tên tác giả không i
          tác giả[tên] += nhận xét khác:              # nếu có thêm số c

                                                      # nếu không
          tác giả[tên] = bình luận                    # chỉ định số bình luận đầu tiên


author_list=[]                                      # Tạo danh sách với tác giả n

cho tên trong tác giả:                               # Lặp lại thông qua tác giả di
     author_list.append([tác giả[tên], tên])         # nối thêm danh sách rỗng với


sorted_list=sorted(author_list, reverse=True)       # Sắp xếp danh sách xuống dần
print(sorted_list[:10])                             # In 10 au đầu tiên
```

```
[[3046, 'whoishiring'], [868, 'mod50ack'], [691, 'boren_ave11'], [531, 'schap
pim'], [520, 'sama'], [489, 'prmph'], [383, 'Apocryphon'], [309, 'curiousga
l'], [284, 'mikemajzoub'], [258, 'philippnagel']]
```

Tìm tiêu đề và nhận xét tương ứng

Bây giờ chúng tôi có danh sách các tác giả có số lượng bình luận cao nhất, hãy để tôi đi qua từng tác giả và tìm ít nhất 10 tiêu đề bài đăng của họ và nhận xét tương ứng mà họ nhận được.

5 tác giả đầu tiên sẽ được phân tích để xem có thể suy luận được điều gì từ đó không.

Tại [65]:
```
# In chi tiết tác giả có bình luận cao nhất

đếm = 0

cho hàng trong ask_posts: if (row[5]=='whoishiring' and count<11): count+=1 # Lặp đi lặp lại # Phát hiện

        print( "Tiêu đề :",hàng[1], "\n", "Số bình luận: ", hàng[4]) #Printing
```

```
Tiêu đề : Hỏi HN: Ai muốn làm thuê? (Tháng 6 năm 2016)
  Số bình luận: 250
Tiêu đề : Hỏi HN: Freelancer? Tìm kiếm freelancer? (Tháng 12 năm 2015)
  Số bình luận: 93
Tiêu đề : Hỏi HN: Ai đang thuê? (Tháng 9 năm 2016)
  Số bình luận: 910
Tiêu đề : Hỏi HN: Ai muốn làm thuê? (tháng 8 năm 2016)
  Số bình luận: 118
Tiêu đề : Hỏi HN: Freelancer? Tìm kiếm freelancer? (Tháng 9 năm 2016)
  Số bình luận: 85
Tiêu đề : Hỏi HN: Ai đang thuê? (tháng 8 năm 2016)
  Số bình luận: 947
Tiêu đề : Hỏi HN: Ai muốn làm thuê? (tháng 4 năm 2016)
  Số bình luận: 283
Tiêu đề : Hỏi HN: Freelancer? Tìm kiếm freelancer? (Tháng 11 năm 2015)
  Số bình luận: 158
Tiêu đề : Hỏi HN: Ai muốn làm thuê? (tháng 3 năm 2016)
  Số bình luận: 202
```

Tại [59]:
```
# Tác giả in ấn có bình luận cao thứ 2

đếm = 0

cho hàng trong ask_posts:
      nếu (hàng[5]=='mod50ack'):
            đếm+=1
            print( "Tiêu đề :",hàng[1], "\n", "Số bình luận: ", hàng[4])
```

```
Tiêu đề : Hỏi HN: Công cụ tốt nhất bạn từng sử dụng mà không tồn tại nữa
nốt Rê?
  Số bình luận: 868
```

Trong [45]:
```
# Tác giả in ấn có bình luận cao thứ 3

đếm=0 cho

hàng trong ask_posts:


    if (row[5]=='boren_ave11' and count<11): count+=1 print( "Tiêu
        đề :",row[1],

        "\n", "Số bình luận: ", row[4])
```

```
Tiêu đề : Hỏi HN: Bạn kiếm được bao nhiêu ở Amazon? Đây là số tiền tôi kiếm được tại Am
azon
  Số bình luận: 691
```

Tại [63]:
```
# Tác giả in ấn có bình luận cao thứ 4

đếm=0 cho

hàng trong ask_posts:


    nếu (hàng[5]=='schappim' và đếm<11):
        đếm+=1

        print( "Tiêu đề :",hàng[1], "\n", "Số bình luận: ", hàng[4])
```

```
Title : Hỏi HN: Bạn được tặng cuốn sách gì?
  Số bình luận: 514 Tiêu đề : Hỏi

HN: Port Apple's Swift sang ESP8266 có khả thi không?
  Số bình luận: 17
```

Trong [47]:
```
# Tác giả in ấn có bình luận cao thứ 5

đếm=0 cho

hàng trong ask_posts:


    if (row[5]=='sama' and count<11): count+=1
        print( "Tiêu

        đề :",row[1], "\n", "Số bình luận: ", row[4])
```

```
Tiêu đề : Hỏi HN: Chúng ta nên tài trợ gì cho YC Research?
  Số bình luận: 520
```

Thêm kết luận

Từ phân tích này, tôi hiểu rằng bằng cách liên tục đăng các câu hỏi liên quan đến Tuyển dụng, tác giả 'whoishiring' đã nhận được số lượng bình luận hàng đầu. Vì vậy, nói về Quy trình tuyển dụng có thể là một ý tưởng hay. Nhưng sau khi xem qua cổng Hacker News, tôi hiểu rằng đây là một bài đăng định kỳ được tạo bởi nhóm đằng sau Hacker News để hỗ trợ quá trình tuyển dụng. Vì vậy, có lẽ đây không phải là nơi bạn tôi nên tập trung.

Bài đăng được bình luận cao thứ hai hỏi về Công cụ, bài thứ ba về kiếm lợi nhuận từ Amazon, bài thứ tư về sách được tặng và bài thứ năm tài trợ cho YC Research. Vì vậy, có một vài điều tôi có thể suy ra từ những chủ đề này.

- Đặt một câu hỏi đơn giản mà mọi người có thể liên quan và mọi người sẽ trả lời, chẳng hạn như về một công cụ lỗi thời. Có thể nói về MS Paint và một số người có thể hoài cổ trong khi

những người khác nói về nó là hoàn toàn lỗi thời.

- Hỏi về điều gì đó có thể khiến mọi người tò mò, chẳng hạn như thu nhập trực tuyến của ai đó từ Amazon hoặc blog, v.v. Hỏi về những
- điều mà mọi người có quan điểm nào đó, chẳng hạn như bạn sẽ kể gì về cuốn sách được tặng làm quà của mình.
- Hỏi về bất kỳ chủ đề mở nào trong việc tài trợ, có thể tạo ra một cuộc thảo luận hấp dẫn.

Nếu tìm được những chủ đề đơn giản, chân thực nhưng gây tò mò, đôi khi gây tranh cãi, có thể tạo cảm xúc cho con người, đồng thời có mối liên hệ với trải nghiệm cuộc sống, tôi nghĩ họ có thể tạo một số bài đăng phổ biến trên kênh Hacker News. Đó là những gì tôi có thể suy ra từ bộ dữ liệu này.

Bây giờ tôi phải viết tất cả những điều này trong một email và gửi cho bạn tôi. Thật là một ngày chủ nhật! Thật là một ngày vui vẻ! Tất cả là nhờ bộ dữ liệu Hacker News!

TRONG [ ]: