Basics

May 30, 2023

1 Analysig Posts in Hacker News Website

In this analysis, we will look at a sampled data set of 20,000 posts in the website Hacker News that received at least one comment and that relate to either ask HN or show HN.

[1]:
```
from csv import reader
%ls
hn = list(reader(open("hacker_news.csv")))
for row in hn[:5]:
    print(row)
```

```
Basics.ipynb*  hacker_news.csv*
['id', 'title', 'url', 'num_points', 'num_comments', 'author', 'created_at']
['12224879', 'Interactive Dynamic Video',
'http://www.interactivedynamicvideo.com/', '386', '52', 'ne0phyte', '8/4/2016
11:52']
['10975351', 'How to Use Open Source and Shut the Fuck Up at the Same Time',
'http://hueniverse.com/2016/01/26/how-to-use-open-source-and-shut-the-fuck-up-
at-the-same-time/', '39', '10', 'josep2', '1/26/2016 19:30']
['11964716', "Florida DJs May Face Felony for April Fools' Water Joke",
'http://www.thewire.com/entertainment/2013/04/florida-djs-april-fools-water-
joke/63798/', '2', '1', 'vezycash', '6/23/2016 22:20']
['11919867', 'Technology ventures: From Idea to Enterprise',
'https://www.amazon.com/Technology-Ventures-Enterprise-Thomas-
Byers/dp/0073523429', '3', '1', 'hswarna', '6/17/2016 0:01']
```

[2]:
```
header = hn[0]
hn = hn[1:]
print("Header: ")
print(header)
print("\nFirst five rows: ")
for row in hn[:5]:
    print(row)
```

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

First five rows:
['12224879', 'Interactive Dynamic Video',
```

```
'http://www.interactivedynamicvideo.com/', '386', '52', 'ne0phyte', '8/4/2016
11:52']
['10975351', 'How to Use Open Source and Shut the Fuck Up at the Same Time',
'http://hueniverse.com/2016/01/26/how-to-use-open-source-and-shut-the-fuck-up-
at-the-same-time/', '39', '10', 'josep2', '1/26/2016 19:30']
['11964716', "Florida DJs May Face Felony for April Fools' Water Joke",
'http://www.thewire.com/entertainment/2013/04/florida-djs-april-fools-water-
joke/63798/', '2', '1', 'vezycash', '6/23/2016 22:20']
['11919867', 'Technology ventures: From Idea to Enterprise',
'https://www.amazon.com/Technology-Ventures-Enterprise-Thomas-
Byers/dp/0073523429', '3', '1', 'hswarna', '6/17/2016 0:01']
['10301696', 'Note by Note: The Making of Steinway L1037 (2007)',
'http://www.nytimes.com/2007/11/07/movies/07stein.html?_r=0', '8', '2',
'walterbell', '9/30/2015 4:12']
```

[3]:
```
for row in hn[:5]:
    print(row)
```

```
['12224879', 'Interactive Dynamic Video',
'http://www.interactivedynamicvideo.com/', '386', '52', 'ne0phyte', '8/4/2016
11:52']
['10975351', 'How to Use Open Source and Shut the Fuck Up at the Same Time',
'http://hueniverse.com/2016/01/26/how-to-use-open-source-and-shut-the-fuck-up-
at-the-same-time/', '39', '10', 'josep2', '1/26/2016 19:30']
['11964716', "Florida DJs May Face Felony for April Fools' Water Joke",
'http://www.thewire.com/entertainment/2013/04/florida-djs-april-fools-water-
joke/63798/', '2', '1', 'vezycash', '6/23/2016 22:20']
['11919867', 'Technology ventures: From Idea to Enterprise',
'https://www.amazon.com/Technology-Ventures-Enterprise-Thomas-
Byers/dp/0073523429', '3', '1', 'hswarna', '6/17/2016 0:01']
['10301696', 'Note by Note: The Making of Steinway L1037 (2007)',
'http://www.nytimes.com/2007/11/07/movies/07stein.html?_r=0', '8', '2',
'walterbell', '9/30/2015 4:12']
```

[4]:
```
ask_posts = []
show_posts = []
other_posts = []
```

[5]:
```
for row in hn:
    title = row[1]
    title = title.lower()
    if title.startswith("ask hn"):
        ask_posts.append(row)
    elif title.startswith("show hn"):
        show_posts.append(row)
    else:
        other_posts.append(row)
```

[6]:
```
print("Number of ask posts: ",len(ask_posts))
print("Number of show posts: ", len(show_posts))
print("Number of other posts: ", len(other_posts))
print("Sum: ",len(ask_posts)+len(show_posts)+len(other_posts))
print("Total number of posts: ",len(hn))
```

```
Number of ask posts:  1744
Number of show posts:  1162
Number of other posts:  17194
Sum:  20100
Total number of posts:  20100
```

[7]:
```
total_ask_comments = 0
for row in ask_posts:
    total_ask_comments += int(row[4])
print("Total number of comments in ask posts: ",total_ask_comments)
avg_ask_comments = total_ask_comments / len(ask_posts)
print("Average number of comments in ask posts: ", avg_ask_comments)
```

```
Total number of comments in ask posts:  24483
Average number of comments in ask posts:  14.038417431192661
```

[8]:
```
total_show_comments = 0
for row in show_posts:
    total_show_comments += int(row[4])
print("Total number of comments in show posts: ",total_show_comments)
avg_show_comments = total_show_comments / len(show_posts)
print("Average number of comments in show posts: ", avg_show_comments)
```

```
Total number of comments in show posts:  11988
Average number of comments in show posts:  10.31669535283993
```

As we can see, the average number of comments in ask posts is larger than that in show posts.

Because ask posts receive more comments, we will analyse those posts further and neglect the show posts.

[9]:
```
import datetime as dt
result_list = []
for row in ask_posts:
    result_list.append([row[6],int(row[4])])
```

[10]:
```
counts_by_hour = {}
comments_by_hour = {}
for row in result_list:
    time_comment_created = dt.datetime.strptime(row[0],"%m/%d/%Y %H:%M")
    hour_comment_created = time_comment_created.strftime("%H")
    if hour_comment_created not in counts_by_hour:
        counts_by_hour[hour_comment_created] = 1
```

```
        comments_by_hour[hour_comment_created] = row[1]
    else:
        counts_by_hour[hour_comment_created] += 1
        comments_by_hour[hour_comment_created] += row[1]
```

[11]:
```
print("Counts by hour: ")
for key,value in counts_by_hour.items():
    print("Hour: ", key, ",Number of posts: ",value)
print("Comments by hour: ")
for key,value in comments_by_hour.items():
    print("Hour ", key, ",Number of comments: ",value)
```

```
Counts by hour:
Hour:  09 ,Number of posts:  45
Hour:  13 ,Number of posts:  85
Hour:  10 ,Number of posts:  59
Hour:  14 ,Number of posts:  107
Hour:  16 ,Number of posts:  108
Hour:  23 ,Number of posts:  68
Hour:  12 ,Number of posts:  73
Hour:  17 ,Number of posts:  100
Hour:  15 ,Number of posts:  116
Hour:  21 ,Number of posts:  109
Hour:  20 ,Number of posts:  80
Hour:  02 ,Number of posts:  58
Hour:  18 ,Number of posts:  109
Hour:  03 ,Number of posts:  54
Hour:  05 ,Number of posts:  46
Hour:  19 ,Number of posts:  110
Hour:  01 ,Number of posts:  60
Hour:  22 ,Number of posts:  71
Hour:  08 ,Number of posts:  48
Hour:  04 ,Number of posts:  47
Hour:  00 ,Number of posts:  55
Hour:  06 ,Number of posts:  44
Hour:  07 ,Number of posts:  34
Hour:  11 ,Number of posts:  58
Comments by hour:
Hour  09 ,Number of comments:  251
Hour  13 ,Number of comments:  1253
Hour  10 ,Number of comments:  793
Hour  14 ,Number of comments:  1416
Hour  16 ,Number of comments:  1814
Hour  23 ,Number of comments:  543
Hour  12 ,Number of comments:  687
Hour  17 ,Number of comments:  1146
Hour  15 ,Number of comments:  4477
```

```
Hour  21 ,Number of comments:  1745
Hour  20 ,Number of comments:  1722
Hour  02 ,Number of comments:  1381
Hour  18 ,Number of comments:  1439
Hour  03 ,Number of comments:  421
Hour  05 ,Number of comments:  464
Hour  19 ,Number of comments:  1188
Hour  01 ,Number of comments:  683
Hour  22 ,Number of comments:  479
Hour  08 ,Number of comments:  492
Hour  04 ,Number of comments:  337
Hour  00 ,Number of comments:  447
Hour  06 ,Number of comments:  397
Hour  07 ,Number of comments:  267
Hour  11 ,Number of comments:  641
```

[12]:
```
avg_by_hour = []
for key in counts_by_hour:
    avg_by_hour.append([key,comments_by_hour[key]/counts_by_hour[key]])
```

[13]:
```
print("hour, avg number of comments per hour")
for row in avg_by_hour:
    print(row[0],", ",row[1])
```

```
hour, avg number of comments per hour
09 ,  5.5777777777777775
13 ,  14.741176470588234
10 ,  13.440677966101696
14 ,  13.233644859813085
16 ,  16.796296296296298
23 ,  7.985294117647059
12 ,  9.41095890410959
17 ,  11.46
15 ,  38.5948275862069
21 ,  16.009174311926607
20 ,  21.525
02 ,  23.810344827586206
18 ,  13.20183486238532
03 ,  7.796296296296297
05 ,  10.08695652173913
19 ,  10.8
01 ,  11.383333333333333
22 ,  6.746478873239437
08 ,  10.25
04 ,  7.170212765957447
00 ,  8.127272727272727
06 ,  9.022727272727273
07 ,  7.852941176470588
```

```
11 ,  11.051724137931034
```

[15]:
```
swap_avg_by_hour = []
for row in avg_by_hour:
    swap_avg_by_hour.append([row[1],row[0]])
print(swap_avg_by_hour)
```

```
[[5.5777777777777775, '09'], [14.741176470588234, '13'], [13.440677966101696,
'10'], [13.233644859813085, '14'], [16.796296296296298, '16'],
[7.985294117647059, '23'], [9.41095890410959, '12'], [11.46, '17'],
[38.5948275862069, '15'], [16.009174311926607, '21'], [21.525, '20'],
[23.810344827586206, '02'], [13.20183486238532, '18'], [7.796296296296297,
'03'], [10.08695652173913, '05'], [10.8, '19'], [11.383333333333333, '01'],
[6.746478873239437, '22'], [10.25, '08'], [7.170212765957447, '04'],
[8.127272727272727, '00'], [9.022727272727273, '06'], [7.852941176470588, '07'],
[11.051724137931034, '11']]
```

[18]:
```
sorted_swap = sorted(swap_avg_by_hour,reverse = True)
for row in sorted_swap:
    print(row)
```

```
[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']
[11.383333333333333, '01']
[11.051724137931034, '11']
[10.8, '19']
[10.25, '08']
[10.08695652173913, '05']
[9.41095890410959, '12']
[9.022727272727273, '06']
[8.127272727272727, '00']
[7.985294117647059, '23']
[7.852941176470588, '07']
[7.796296296296297, '03']
[7.170212765957447, '04']
[6.746478873239437, '22']
[5.5777777777777775, '09']
```

Top 5 Hours for Ask Posts Comments

[19]:
```
for row in sorted_swap[:5]:
    print(row)
```

```
[38.5948275862069, '15']
[23.810344827586206, '02']
[21.525, '20']
[16.796296296296298, '16']
[16.009174311926607, '21']
```

[27]:
```
for row in sorted_swap[:5]:
    print(str.format("{hour}:00: {n_comm:.2f} average comments per post",hour =␣
 ↪row[1],n_comm = row[0]))
```

```
15:00: 38.59 average comments per post
02:00: 23.81 average comments per post
20:00: 21.52 average comments per post
16:00: 16.80 average comments per post
21:00: 16.01 average comments per post
```

From the data shown above, at 15 hours, or 3pm eastern time or 1pm mountain time, one should create a post to have a higher chance of receiving comments.