Exploring Trends in Hacker News Posts

February 8, 2021

1 Exploring Trends in Hacker News Posts

Hacker News is an online platform started by Y Combinator, and is extremely popular in circles of the tech community. Two main types of posts are often added:

- "Ask HN" posts, which are composed of a question directed towards the community
- "Show HN" posts, which often seek to publicize a product, project, or an interesting development

I will analyze a data set from Hacker News (from roughly September 2015 - September 26, 2016) in order to discern which kinds of posts get the most engagement (can be found here: https://www.kaggle.com/hacker-news/hacker-news-posts). Additionally, I will analyze whether posting at certain times invites more engagement – all times are in EST.

[1]:
```
from csv import reader
open_file = open('/Users/natasharavinand/Downloads/my_datasets/Projects/
 ↪HN_posts_year_to_Sep_26_2016.csv')
read_file = reader(open_file)
hn = list(read_file)
print(hn[:5])
```

```
[['id', 'title', 'url', 'num_points', 'num_comments', 'author', 'created_at'],
['12579008', 'You have two days to comment if you want stem cells to be
classified as your own',
'http://www.regulations.gov/document?D=FDA-2015-D-3719-0018', '1', '0',
'altstar', '9/26/2016 3:26'], ['12579005', 'SQLAR  the SQLite Archiver',
'https://www.sqlite.org/sqlar/doc/trunk/README.md', '1', '0', 'blacksqr',
'9/26/2016 3:24'], ['12578997', 'What if we just printed a flatscreen television
on the side of our boxes?', 'https://medium.com/vanmoof/our-secrets-
out-f21c1f03fdc8#.ietxmez43', '1', '0', 'pavel_lishin', '9/26/2016 3:19'],
['12578989', 'algorithmic music',
'http://cacm.acm.org/magazines/2011/7/109891-algorithmic-composition/fulltext',
'1', '0', 'poindontcare', '9/26/2016 3:16']]
```

[2]:
```
headers = hn[0]
hn = hn[1:]
print(headers)
print(hn[:5])
```

```
['id', 'title', 'url', 'num_points', 'num_comments', 'author', 'created_at']
[['12579008', 'You have two days to comment if you want stem cells to be
classified as your own',
'http://www.regulations.gov/document?D=FDA-2015-D-3719-0018', '1', '0',
'altstar', '9/26/2016 3:26'], ['12579005', 'SQLAR  the SQLite Archiver',
'https://www.sqlite.org/sqlar/doc/trunk/README.md', '1', '0', 'blacksqr',
'9/26/2016 3:24'], ['12578997', 'What if we just printed a flatscreen television
on the side of our boxes?', 'https://medium.com/vanmoof/our-secrets-
out-f21c1f03fdc8#.ietxmez43', '1', '0', 'pavel_lishin', '9/26/2016 3:19'],
['12578989', 'algorithmic music',
'http://cacm.acm.org/magazines/2011/7/109891-algorithmic-composition/fulltext',
'1', '0', 'poindontcare', '9/26/2016 3:16'], ['12578979', 'How the Data Vault
Enables the Next-Gen Data Warehouse and Data Lake',
'https://www.talend.com/blog/2016/05/12/talend-and-Â\x93the-data-vaultÂ\x94',
'1', '0', 'markgainor1', '9/26/2016 3:14']]
```

1.1 Do "Ask HN" or "Show HN" Posts Receive More Comments?

[3]:
```
ask_posts = []
show_posts = []
other_posts = []

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

print(len(ask_posts))
print(len(show_posts))
print(len(other_posts))
```

```
9139
10158
273822
```

[4]:
```
total_ask_comments = 0

for post in ask_posts:
    n_comments = int(post[4])
    total_ask_comments += n_comments

avg_ask_comments = total_ask_comments / len(ask_posts)
```

```
print(avg_ask_comments)

total_show_comments = 0

for post in show_posts:
    n_comments = int(post[4])
    total_show_comments += n_comments

avg_show_comments = total_show_comments / len(show_posts)

print(avg_show_comments)
```

```
10.393478498741656
4.886099625910612
```

We see above that "Ask HN" posts tend to receive more comments on average (average comment count of 10 versus approximately 5).

1.2 Do Ask Posts at Certain Times Receive More Engagement?

We'll focus our remaining analysis on ask posts to determine whether asks posts created at a certain time are more likely to attract comments. I will:

- Calculate the amount of ask posts created in each hour of the day, along with the number of comments received.
- Calculate the average number of comments ask posts receive by hour created.

[5]:
```
import datetime as dt

result_list = []

for post in ask_posts:
    created_at = post[6]
    comments = int(post[4])
    post_list = [created_at, comments]
    result_list.append(post_list)

counts_by_hour = {}
comments_by_hour = {}

for result in result_list:
    fmt = '%m/%d/%Y %H:%M'
    dt_obj = dt.datetime.strptime(result[0], fmt)
    hour = dt_obj.strftime("%H")
    if hour in counts_by_hour:
        counts_by_hour[hour] += 1
        comments_by_hour[hour] += int(result[1])
    else:
```

```
        counts_by_hour[hour] = 1
        comments_by_hour[hour] = int(result[1])
```

[6]:
```
avg_by_hour = []

for hour in comments_by_hour:
    avg_by_hour.append([hour, comments_by_hour[hour]/counts_by_hour[hour]])

print(avg_by_hour)
```

```
[['02', 11.137546468401487], ['01', 7.407801418439717], ['22',
8.804177545691905], ['21', 8.687258687258687], ['19', 7.163043478260869], ['17',
9.449744463373083], ['15', 28.676470588235293], ['14', 9.692007797270955],
['13', 16.31756756756757], ['11', 8.96474358974359], ['10', 10.684397163120567],
['09', 6.653153153153153], ['07', 7.013274336283186], ['03', 7.948339483394834],
['23', 6.696793002915452], ['20', 8.749019607843136], ['16', 7.713298791018998],
['08', 9.190661478599221], ['00', 7.5647840531561465], ['18', 7.94299674267101],
['12', 12.380116959064328], ['04', 9.7119341563786], ['06', 6.782051282051282],
['05', 8.794258373205741]]
```

[7]:
```
swap_avg_by_hour = []

for row in avg_by_hour:
    swap_avg_by_hour.append([row[1], row[0]])

sorted_swap = sorted(swap_avg_by_hour, reverse=True)

print("Top 5 Hours for Ask Posts Comments")
fmt = "{hour}: {comments:.2f} average comments per post"

for row in sorted_swap[:5]:
    dt_obj = dt.datetime.strptime(row[1], "%H")
    hour = dt_obj.strftime("%H:%M")
    string = fmt.format(hour = hour, comments = float(row[0]))
    print(string)
```

```
Top 5 Hours for Ask Posts Comments
15:00: 28.68 average comments per post
13:00: 16.32 average comments per post
12:00: 12.38 average comments per post
02:00: 11.14 average comments per post
10:00: 10.68 average comments per post
```

This short analysis shows us the best time (all times in EST) for posting – to garner the most comments – is 3 PM, followed by 1 PM, followed by 12 AM.