Data Science

Syed Ali Irtaza

06/13/2020

Hacker news is a popular site where technology related news are posted and people are allowed to like and commnet on these posts. The data has been reduced from 300,000 rows to approximately 20,000 rows by removing all submissions that did not receive any comments, and then randomly sampling from the remaining submissions. In this project we will compare two different types of posts.

Ask HN; post to ask questions to Hacker News community. Show HN; to share a project with Hacker News community.

Introduction

Read in the data

In [1]:
```
import csv

file = open('hackers_news.csv')
hn = list(csv.reader(file))
hn[:5]
```

Out[1]:
```
[['id', 'title', 'url', 'num_points', 'num_comments', 'author', 'crea
ted_at'],
 ['12579008',
  'You have two days to comment if you want stem cells to be classifi
ed 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-compositio
n/fulltext',
  '1',
  '0',
  'poindontcare',
  '9/26/2016 3:16']]
```

Removing header

In [2]:
```
headers = hn[0]
hn = hn[1:]
print(headers)
print("\n")
print(hn[:5])
```

```
['id', 'title', 'url', 'num_points', 'num_comments', 'author', 'creat
ed_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=FD
A-2015-D-3719-0018', '1', '0', 'altstar', '9/26/2016 3:26'], ['125790
05', '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 ou
r 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-alg
orithmic-composition/fulltext', '1', '0', 'poindontcare', '9/26/2016
3:16'], ['12578979', 'How the Data Vault Enables the Next-Gen Data Wa
rehouse and Data Lake', 'https://www.talend.com/blog/2016/05/12/talen
d-and-Â\x93the-data-vaultÂ\x94', '1', '0', 'markgainor1', '9/26/2016
3:14']]
```

Creating new lists of list containing Ask HN and Show HN

We will extract data into two different lists. One containing Ask HN and other Show HN.

In [3]:
```
ask_posts = []
show_posts = []
other_posts = []
for row in hn:
    title = row[1]
    if title.lower().startswith("ask hn"):
        ask_posts.append(row)

    elif title.lower().startswith("show hn"):
        show_posts.append(row)

    else:
        other_posts.append(row)

print("The ask hn posts are = ", len(ask_posts))
print("The show hn posts are = ", len(show_posts))
print("The other posts are = ",  len(other_posts))
```

```
The ask hn posts are =  9139
The show hn posts are =  10158
The other posts are =  273822
```

Average comments on each type of post

Ask HN

Average comments under Ask hn posts

In [4]:
```
total_ask_comment = 0
for row in ask_posts:
    total_ask_comment += int(row[4])

avg_ask_comments = total_ask_comment / len(ask_posts)
print("The average comments of ask hn posts is = ", avg_ask_comments)
```

```
The average comments of ask hn posts is =  10.393478498741656
```

Show HN

In [5]:
```
total_show_comment = 0
for row in show_posts:
    total_show_comment += int(row[4])

avg_show_comments = total_show_comment / len(show_posts)
print("The average comments of show hn post is = ", avg_show_comments)
```

```
The average comments of show hn post is =  4.886099625910612
```

According to our calculations, Ask HN received approximately 10 comments and Show HN received approximately 4 comments. Ask posts are more likely to receive comments.

Amount of Ask posts and comment created in each hour of the day

In [10]:
```
import datetime as dt
result_list = []

for row in ask_posts:
    result_list.append([row[6], int(row[4])])


counts_by_hour = {}
comments_by_hour = {}

for row in result_list:
    date = row[0]
    comment = row[1]
    time = dt.datetime.strptime(date, "%m/%d/%Y %H:%M")
    hour = time.strftime("%H")
    if hour not in counts_by_hour:
        counts_by_hour[hour] = 1
        comments_by_hour[hour] = comment
    else:
        counts_by_hour[hour] += 1
        comments_by_hour[hour] += comment

counts_by_hour
```

Out[10]:
```
{'02': 269,
 '01': 282,
 '22': 383,
 '21': 518,
 '19': 552,
 '17': 587,
 '15': 646,
 '14': 513,
 '13': 444,
 '11': 312,
 '10': 282,
 '09': 222,
 '07': 226,
 '03': 271,
 '23': 343,
 '20': 510,
 '16': 579,
 '08': 257,
 '00': 301,
 '18': 614,
 '12': 342,
 '04': 243,
 '06': 234,
 '05': 209}
```

In [11]:
```
comments_by_hour
```

Out[11]:
```
{'02': 2996,
 '01': 2089,
 '22': 3372,
 '21': 4500,
 '19': 3954,
 '17': 5547,
 '15': 18525,
 '14': 4972,
 '13': 7245,
 '11': 2797,
 '10': 3013,
 '09': 1477,
 '07': 1585,
 '03': 2154,
 '23': 2297,
 '20': 4462,
 '16': 4466,
 '08': 2362,
 '00': 2277,
 '18': 4877,
 '12': 4234,
 '04': 2360,
 '06': 1587,
 '05': 1838}
```

Average number of comments per post for posts created during each hour of the day.

In [7]:
```
avg_by_hour = []

for row in comments_by_hour:
    avg_by_hour.append([row, comments_by_hour[row] / counts_by_hour[ro

avg_by_hour
```

Out[7]:
```
[['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]]
```

Sorting

In [12]:
```
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)

sorted_swap
```

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

In [9]:
```
print("Top 5 Hours of Ask Posts Comments \n")

for row in sorted_swap[:5]:
    print(
    "{}:00 {:.2f} average comment per post.".format(row[1], row[0]))
```

```
Top 5 Hours of Ask Posts Comments

15:00 28.68 average comment per post.
13:00 16.32 average comment per post.
12:00 12.38 average comment per post.
02:00 11.14 average comment per post.
10:00 10.68 average comment per post.
```

15:00 is the hour in which Ask posts receives most comments with an average of 28.68 comments per post. When we convert 15:00 24 hour clock to 12 hour clock, it is 3:00 pm.

Conclusion

In this project, we analyzed Hacker News posts to determine which type of post and at what time does it receive the most comments. According to our analysis, in order to get the maximum number of comment on a post, it is recommended to post an Ask post between 15:00 to 16:00 (3:00pm- 4:00pm est).

Note: The data was reduced from 300,000 rows to approximately 20,000 rows by removing all submissions that did not receive any comments.