DATAx New York City - Data Science in the real world. Use KD200 to save

A Beginner's Guide to Tweet Analytics with Pandas

Previous post
Next post

http likes 223

Like 79 Share 79 Tweet Share 159

Tags: Pandas, Python, Twitter

Unlike a lot of other tutorials which often pull from the real-time Twitter API, we will be using the downloadable Twitter Analytics data, and most of what we do will be done in Pandas.

Want to code in Python, Java or R?
Try SAS Viya free for 14 days

By Matthew Mayo, KDnuggets.

Twitter provides access to analytics for all of its users, but I am assuming relatively few vanilla tweeples pay much attention to its existence. There are a variety of other services which can help perform tweet and audience analytics, and further analysis such as that related to geographic and natural language processing, but when paired with some simple Python, the Twitter-supplied data can be incredibly useful.

This is a simple guide to getting your hands a bit dirty doing analysis on your own in Python. Unlike a lot of other tutorials which often pull from the real-time Twitter API, we will be using the downloadable Twitter Analytics data, and most of what we do will be done in Pandas.

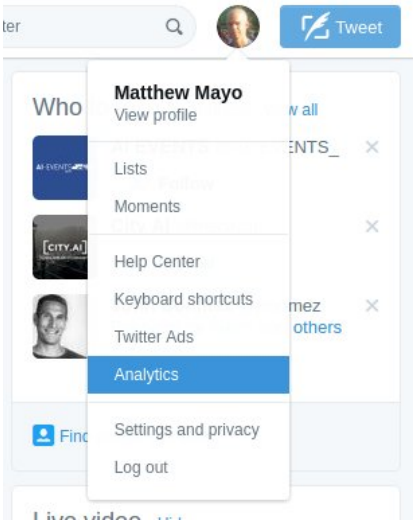
Before we get started, let's get the obligatory imports out of the way.

```
1  import pandas as pd
2  import string
```

twitter-analytics-0.py hosted with ❤ by GitHub view raw

Get and Inspect the Data

First we need the data. This part is easy enough; go to Twitter, click on the upper right menu (your profile pic), select **Analytics**, choose the **Tweets** tab along the top, use the date range pickers to select a time period, and choose **Export Data**. It doesn't matter how much data you use; our simple example will work with any amount. I chose the default, **Past 28 Days**.



Get that Twitter analytics data.

Once we have the CSV file, we will want to load it into a Pandas DataFrame for analysis.

```
1  def load_tweets(tweet_file):
2
3      """ Load and process a Twitter analytics data file """
4
5      # Read tweet data (obtained from Twitter Analytics)
6      tweet_df = pd.read_csv(tweet_file)
```

```
    # Drop irrelevant columns
    tweet_df = tweet_df.drop(tweet_df.columns[[13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36,37,38,39]],

    return tweet_df
```

twitter-analytics-1.py hosted with ❤ by GitHub view raw

Don't mind all the dropped columns; while a lot of what is there is useful for our analysis -- tweet text, time, impressions, retweets, etc. -- many are not -- all the **promoted** things -- and so we will just omit them from the start.

As we would with any data analysis project, next we have a look at the data.

```
tweet_df = load_tweets('tweets.csv')
tweet_df.head()
```

twitter-analytics-2.py hosted with ❤ by GitHub view raw

	Tweet id	Tweet permalink	Tweet text	time	impressions	engagements	engagement rate	retweets	replies	likes	user profile clicks	url clicks	hashtag clicks

Running:

```
len(tweet_df.index)
```

twitter-analytics-3.py hosted with ❤ by GitHub view raw

tells us that I have tweeted a measly 95 times in the past 4 weeks. Not a very large dataset, and we probably would not want to make any inferences based on our findings, but a good enough toy set to start out with.

Let's see what useful analytics we can pull out of this.

Basic Tweet Stats

So, given what data is shown in the output of running `head()` on the dataset above, and having a rough intuition of what tweet metrics would be useful, we will grab the following stats:

- **Retweets** - Mean RTs per tweet & top 5 RTed tweets
- **Likes** - Mean likes per tweet & top 5 liked tweets
- **Impressions** - Mean impressions per tweet & top 5 tweets with most impressions

```
# Total tweets
print 'Total tweets this period:', len(tweet_df.index), '\n'

# Retweets
tweet_df = tweet_df.sort_values(by='retweets', ascending=False)
tweet_df = tweet_df.reset_index(drop=True)
print 'Mean retweets:', round(tweet_df['retweets'].mean(),2), '\n'
print 'Top 5 RTed tweets:'
print '------------------'
for i in range(5):
    print tweet_df['Tweet text'].ix[i], '-', tweet_df['retweets'].ix[i]
print '\n'

# Likes
tweet_df = tweet_df.sort_values(by='likes', ascending=False)
tweet_df = tweet_df.reset_index(drop=True)
print 'Mean likes:', round(tweet_df['likes'].mean(),2), '\n'
print 'Top 5 liked tweets:'
print '-------------------'
for i in range(5):
```

```
    print tweet_df['Tweet text'].ix[i], '-', tweet_df['likes'].ix[i]
print '\n'

# Impressions
tweet_df = tweet_df.sort_values(by='impressions', ascending=False)
tweet_df = tweet_df.reset_index(drop=True)
print 'Mean impressions:', round(tweet_df['impressions'].mean(),2), '\n'
print 'Top 5 tweets with most impressions:'
print '-----------------------------------'
for i in range(5):
    print tweet_df['Tweet text'].ix[i], '-', tweet_df['impressions'].ix[i]
```

twitter-analytics-5.py hosted with ❤ by GitHub view raw

```
Total tweets this period: 95

Mean retweets: 1.72

Top 5 RTed tweets:
------------------
A PyTorch IPython Notebook tutorial on #deeplearning, with an emphasis on #NaturalLanguageProcessing https://t.co/bxiBD42T7I
On the Origin of #DeepLearning https://t.co/oe7r43HHVS #NeuralNetworks #arxiv https://t.co/BIcba61FR9  -  25
7 MORE Steps to Mastering #MachineLearning With #Python https://t.co/5yAjeUpCfS https://t.co/juWH0rQaNR  -  16
Every Intro to #DataScience Course on the Internet, Ranked https://t.co/rQG7Higk6b https://t.co/b6VveKfJxD  -  8
Pandas & Seaborn - A guide to handle & visualize #data elegantly @tryolabs https://t.co/LPq2q8k1i1 #Python #dataviz https:/

Mean likes: 3.17

Top 5 liked tweets:
-------------------
A PyTorch IPython Notebook tutorial on #deeplearning, with an emphasis on #NaturalLanguageProcessing https://t.co/bxiBD42T7I
7 MORE Steps to Mastering #MachineLearning With #Python https://t.co/5yAjeUpCfS https://t.co/juWH0rQaNR  -  37
On the Origin of #DeepLearning https://t.co/oe7r43HHVS #NeuralNetworks #arxiv https://t.co/BIcba61FR9  -  37
Pandas & Seaborn - A guide to handle & visualize #data elegantly @tryolabs https://t.co/LPq2q8k1i1 #Python #dataviz https:/
I've been reposted on @YhatHQ - The Current State of Automated #MachineLearning https://t.co/ggAW1Hrmxk https://t.co/8030HhA

Mean impressions: 674.39

Top 5 tweets with most impressions:
-----------------------------------
On the Origin of #DeepLearning https://t.co/oe7r43HHVS #NeuralNetworks #arxiv https://t.co/BIcba61FR9 - 6409
A PyTorch IPython Notebook tutorial on #deeplearning, with an emphasis on #NaturalLanguageProcessing https://t.co/bxiBD42T7I
7 MORE Steps to Mastering #MachineLearning With #Python https://t.co/5yAjeUpCfS https://t.co/juWH0rQaNR - 4374
I've been reposted on @YhatHQ - The Current State of Automated #MachineLearning https://t.co/ggAW1Hrmxk https://t.co/8030HhA
Pandas & Seaborn - A guide to handle & visualize #data elegantly @tryolabs https://t.co/LPq2q8k1i1 #Python #dataviz https:/
```

I won't bother with any analysis of these metrics. Needless to say, I should step my social media game up.

Top #Hashtags and @Mentions

It's no secret that hashtags play an important role in Twitter, and mentions can also help grow your network and influence. Together they help put the 'social' in social networking, transforming platforms like Twitter from passive experiences to very active ones. With that, getting a handle on the most social aspect of this social network can be a helpful endeavour.

```
# Hashtags & mentions
tag_dict = {}
mention_dict = {}

for i in tweet_df.index:
    tweet_text = tweet_df.ix[i]['Tweet text']
    tweet = tweet_text.lower()
    tweet_tokenized = tweet.split()

    for word in tweet_tokenized:
        # Hashtags - tokenize and build dict of tag counts
```

```
        if (word[0:1] == '#' and len(word) > 1):
            key = word.translate(string.maketrans("",""), string.punctuation)
            if key in tag_dict:
                tag_dict[key] += 1
            else:
                tag_dict[key] = 1

        # Mentions - tokenize and build dict of mention counts
        if (word[0:1] == '@' and len(word) > 1):
            key = word.translate(string.maketrans("",""), string.punctuation)
            if key in mention_dict:
                mention_dict[key] += 1
            else:
                mention_dict[key] = 1

# The 10 most popular tags and counts
top_tags = dict(sorted(tag_dict.iteritems(), key=operator.itemgetter(1), reverse=True)[:10])
top_tags_sorted = sorted(top_tags.items(), key=lambda x: x[1])[::-1]
print 'Top 10 hashtags:'
print '----------------'
for tag in top_tags_sorted:
    print tag[0], '-', str(tag[1])

# The 10 most popular mentions and counts
top_mentions = dict(sorted(mention_dict.iteritems(), key=operator.itemgetter(1), reverse=True)[:10])
top_mentions_sorted = sorted(top_mentions.items(), key=lambda x: x[1])[::-1]
print '\nTop 10 mentions:'
print '----------------'
for mention in top_mentions_sorted:
    print mention[0], '-', str(mention[1])
```

twitter-analytics-4.py hosted with ❤ by GitHub view raw

```
Top 10 hashtags:
----------------
machinelearning - 21
deeplearning - 16
python - 15
datascience - 10
neuralnetworks - 10
ai - 5
data - 4
datascientist - 3
tensorflow - 3
rstats - 3

Top 10 mentions:
----------------
kdnuggets - 3
francescoai - 2
jakevdp - 2
quora - 2
yhathq - 2
noahmp - 1
udacity - 1
clavitolo - 1
monkeylearn - 1
nicholashould - 1
```

Putting aside some evident bumps like punctuation being removed from tweet text prior to checking Twitter handles (this could be a problem if you have tweeps named both Francesco_AI and FrancescoAI), this works and is at least relatively Pythonic (though I'm sure it could be more so).

Time-series Analysis

Finally, let's have a look at some very basic temporal data. We will check mean impressions for tweets based -- independently -- on both the hour of day and day of week that they are tweeted. I caution (once gain) that this is based on very little data, and so nothing useful will likely be gleaned. However, given much larger amounts of tweet data, entire social media campaigns are planned.

While this is based on impressions, it could just as reasonably (and easily changed to) be based on engagements, or RTs, or whatever else you pleased. Working in advertising, and promoting tweets? Maybe you are more interested in some of those **promotion*** metrics we hacked off the dataset at the start.

We have to convert the Twitter supplied date field to a legitimate Python datetime object, bin the data based on which hourly slot it falls into, identify days of week, and then capture this data in a couple of additional columns in the DataFrame, which we will pillage for stats afterward.

```
# Time-series impressions (DOW, HOD, etc) (0 = Sunday... 6 = Saturday)
gmt_offset = -4

# Create proper datetime column, apply local GMT offset
tweet_df['ts'] = pd.to_datetime(tweet_df['time'])
tweet_df['ts'] = tweet_df.ts + pd.to_timedelta(gmt_offset, unit='h')

# Add hour of day and day of week columns
tweet_df['hod'] = [t.hour for t in tweet_df.ts]
tweet_df['dow'] = [t.dayofweek for t in tweet_df.ts]

hod_dict = {}
hod_count = {}
dow_dict = {}
dow_count = {}
weekday_dict = {0: 'Mon', 1: 'Tue', 2: 'Wed', 3: 'Thu', 4: 'Fri', 5: 'Sat', 6: 'Sun'}

# Process tweets, collect stats
for i in tweet_df.index:
    hod = tweet_df.ix[i]['hod']
    dow = tweet_df.ix[i]['dow']
    imp = tweet_df.ix[i]['impressions']

    if hod in hod_dict:
        hod_dict[hod] += int(imp)
        hod_count[hod] += 1
    else:
        hod_dict[hod] = int(imp)
        hod_count[hod] = 1

    if dow in dow_dict:
        dow_dict[dow] += int(imp)
        dow_count[dow] += 1
    else:
        dow_dict[dow] = int(imp)
        dow_count[dow] = 1

print 'Average impressions per tweet by hour tweeted:'
print '----------------------------------------------'
for hod in hod_dict:
    print hod, '-', hod+1, ':', hod_dict[hod]/hod_count[hod], '=>', hod_count[hod], 'tweets'

print '\nAverage impressions per tweet by day of week tweeted:'
print '-----------------------------------------------------'
for dow in dow_dict:
    print weekday_dict[dow], ':', dow_dict[dow]/dow_count[dow], '=>', dow_count[dow], ' tweets'
```

twitter-analytics-6.py hosted with ❤ by GitHub view raw

```
Average impressions per tweet by hour tweeted:
-----------------------------------------------
0 - 1 : 141 => 1 tweets
9 - 10 : 445 => 10 tweets
10 - 11 : 611 => 9 tweets
11 - 12 : 1319 => 10 tweets
12 - 13 : 528 => 10 tweets
13 - 14 : 448 => 11 tweets
14 - 15 : 464 => 16 tweets
15 - 16 : 763 => 8 tweets
17 - 18 : 634 => 9 tweets
18 - 19 : 1306 => 8 tweets
19 - 20 : 454 => 1 tweets
21 - 22 : 186 => 1 tweets
23 - 24 : 208 => 1 tweets

Average impressions per tweet by day of week tweeted:
-----------------------------------------------------
Mon : 475 => 20  tweets
Tue : 568 => 18  tweets
Wed : 1418 => 18  tweets
Thu : 545 => 17  tweets
Fri : 432 => 22  tweets
```

It seems I tweet at rather consistent times of day. It also seems that my Wednesday tweets, 11 AM tweets, and 6 PM tweets are my bread and butter. Of course, this is based on 95 tweets, and so is meaningless and inconclusive. However, after performing these same steps on some considerably larger sets of data, some interesting trends have been observed which may help lead to business decisions. All from some simple Python.

While not earth-shattering, our simple Pandas-based Twitter analytics code is enough to get us thinking about how we may better use social media. Applied to the right data, elementary scripts can be quite powerful.

Related:

- Mining Twitter Data with Python Part 1: Collecting Data
- Pandas Cheat Sheet: Data Science and Data Wrangling in Python
- Tidying Data in Python

Previous post
Next post

Top Stories Past 30 Days

Most Popular

1. **The Death of Big Data and the Emergence of the Multi-Cloud Era**
2. **Top 13 Skills To Become a Rockstar Data Scientist**
3. **Top 10 Data Science Leaders You Should Follow**
4. **Top 10 Best Podcasts on AI, Analytics, Data Science, Machine Learning**
5. **5 Probability Distributions Every Data Scientist Should Know**
6. **Convolutional Neural Networks: A Python Tutorial Using TensorFlow and Keras**
7. **What 70% of Data Science Learners Do Wrong**

Most Shared

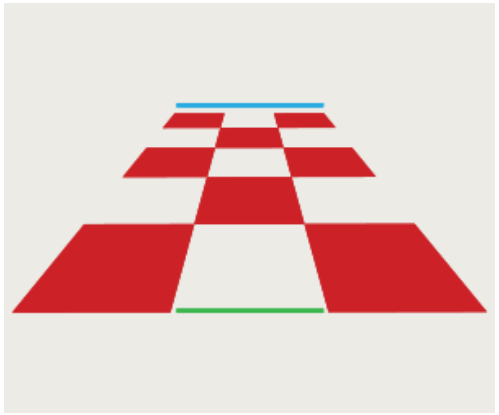
1. **The Death of Big Data and the Emergence of the Multi-Cloud Era**
2. **What's wrong with the approach to Data Science?**
3. **Training a Neural Network to Write Like Lovecraft**
4. **Top 13 Skills To Become a Rockstar Data Scientist**
5. **Convolutional Neural Networks: A Python Tutorial Using TensorFlow and Keras**
6. **Bayesian deep learning and near-term quantum computers: A cautionary tale in quantum machine learning**
7. **Top 10 Best Podcasts on AI, Analytics, Data Science, Machine Learning**

Latest News

- How to better manage your data science team's wor...
- Top Stories, Jul 29 – Aug 4: Top 10 Best Podcasts...
- [video] Introduction to Generative Adversarial Networks...
- Machine Learning is Happening Now: A Survey of Organiza...
- Getting Started With Data Science

- South Dakota State University: Data Visualization Devel...

KNIME Fall Summit 2019
Nov 5-8, Austin
Use code KDNUGGETS for 10% off



How you view data changes how you view business decisions. Take the next step. Get MSBA at NYU Stern.

Top Stories
Last Week

Most Popular

1. **Top 10 Best Podcasts on AI, Analytics, Data Science, Machine Learning**

2. **What 70% of Data Science Learners Do Wrong**
3. **Top 13 Skills To Become a Rockstar Data Scientist**
4. **Five Command Line Tools for Data Science**
5. **How a simple mix of object-oriented programming can sharpen your deep learning prototype**
6. **Top Certificates and Certifications in Analytics, Data Science, Machine Learning and AI**
7. **Convolutional Neural Networks: A Python Tutorial Using TensorFlow and Keras**

Most Shared

1. **Top 10 Best Podcasts on AI, Analytics, Data Science, Machine Learning**
2. **What 70% of Data Science Learners Do Wrong**
3. **GPU Accelerated Data Analytics & Machine Learning**
4. **Understanding Tensor Processing Units**
5. **Ten more random useful things in R you may not know about**
6. **Easily Deploy Deep Learning Models in Production**
7. **Pytorch Cheat Sheet for Beginners and Udacity Deep Learning Nanodegree**

More Recent Stories

- South Dakota State University: Data Visualization Developer an...
- Statistical Thinking for Industrial Problem Solving (STIPS) &#...
- Pytorch Cheat Sheet for Beginners and Udacity Deep Learning Na...
- GPU Accelerated Data Analytics & Machine Learning
- What 70% of Data Science Learners Do Wrong
- Easily Deploy Deep Learning Models in Production
- Opening Black Boxes: How to leverage Explainable Machine Learning
- How a simple mix of object-oriented programming can sharpen yo...
- A 2019 Guide to Object Detection
- Top tweets, Jul 24-30: Nothing but NumPy: Understanding and...
- Are We Ready to Partner With Machines?
  Data Science Salon ...
- Can we trust AutoML to go on full autopilot?
- Five Command Line Tools for Data Science
- Ten more random useful things in R you may not know about
- KDnuggets 19:n28, Jul 31: Top 13 Skills To Become a Rocksta...
- A Data Science Playbook for explainable ML/xAI
- Understanding Tensor Processing Units
- P-values Explained By Data Scientist
- Here's how you can accelerate your Data Science on GPU
- Monash University: Lecturer / Sr Lecturer – Blockchain [...

© 2019 KDnuggets. About KDnuggets. Privacy policy. Terms of Service

Subscribe to KDnuggets News

X