Making smarter apps with ML

Acquire data using Beautiful Soup

Beautiful Soup provides a few simple methods and Pythonic idioms for navigating, searching, and modifying a parse tree: a toolkit for dissecting a document and extracting what you need. It doesn't take much code to write an application

In [5]:

```
import get_stock_news as getter
import pprint as pp

source_url = "https://www.google.com/finance/company_news?q=NYSE%3ATWTR&ei=aXN
VWfnGBd3a2AaHhKq4Dg"

headlines = getter.get_from_url(source_url)
pp.pprint(headlines)
```

```
[u"The Simple Reason I Still Won't Buy Twitter\xa0Inc.",
 u"Twitter Inc's Brain Drain\xa0Continues",
 u'Twitter Inc (TWTR) Stock Remains Hopelessly\xa0Overvalued',
 u'\nChecking the Overall Picture for Twitter, Inc. (TWTR)',
 u'\nEPS for Twitter Inc (TWTR) Expected At $-0.11',
 u'Better Buy: Twitter, Inc. vs\xa0Google',
 u"Ackman Joins Twitter, Follows Trump, Bezos, 'Hamilton'\xa0Creator",
 u'Better Buy: Twitter, Inc. vs.\xa0Baidu',
 u'Twitter: The Turning Revenue\xa0Trend',
 u"\nDon't Buy Twitter Inc (TWTR) Stock, Buy a 1900% No-Brainer Instead",
 u'\nWhy The Rally In Twitter Inc (TWTR) Stock Could Continue For Now',
 u'Twitter will partner with IGN to live stream San Diego\xa0Comic-Con',
 u'\nIGN Entertainment and Twitter Partner on Global Live Stream at San Diego
 ...',
 u'Twitter: Take This Opportunity To\xa0Sell',
 u'Why Twitter Inc (TWTR) Stock Is Still Stuck in the\xa0Mud',
 u'\n2',
 u'\n3',
 u'\n4',
 u'\n5',
 u'\n6',
 u'\n7',
 u'\n8',
 u'\n9',
 u'\n10',
 u'Next']
```

Analyzing Text with Textblob and NLTK

In [6]:
```
from textblob import TextBlob

first_sent = TextBlob("Jupyter notebooks are a great way to demo code.")
print(first_sent.sentiment)
```

```
Sentiment(polarity=0.8, subjectivity=0.75)
```

In [7]:
```
neg_sent = TextBlob("I am unsure about quality of sentiment analysis purely ba
sed on rules.")
print(neg_sent.sentiment)
```

```
Sentiment(polarity=0.21428571428571427, subjectivity=0.5)
```

In [8]:
```
for line in headlines:
    blob_line = TextBlob(line)
    print(blob_line.sentiment)
```

```
Sentiment(polarity=0.0, subjectivity=0.35714285714285715)
Sentiment(polarity=0.0, subjectivity=0.0)
Sentiment(polarity=0.0, subjectivity=0.0)
Sentiment(polarity=0.0, subjectivity=0.0)
Sentiment(polarity=-0.1, subjectivity=0.4)
Sentiment(polarity=0.5, subjectivity=0.5)
Sentiment(polarity=0.0, subjectivity=0.0)
Sentiment(polarity=0.5, subjectivity=0.5)
Sentiment(polarity=0.0, subjectivity=0.0)
Sentiment(polarity=0.0, subjectivity=0.0)
Sentiment(polarity=0.0, subjectivity=0.0)
Sentiment(polarity=0.13636363636363635, subjectivity=0.5)
Sentiment(polarity=0.06818181818181818, subjectivity=0.25)
Sentiment(polarity=0.0, subjectivity=0.0)
Sentiment(polarity=0.0, subjectivity=0.0)
Sentiment(polarity=0.0, subjectivity=0.0)
Sentiment(polarity=0.0, subjectivity=0.0)
Sentiment(polarity=0.0, subjectivity=0.0)
Sentiment(polarity=0.0, subjectivity=0.0)
Sentiment(polarity=0.0, subjectivity=0.0)
Sentiment(polarity=0.0, subjectivity=0.0)
Sentiment(polarity=0.0, subjectivity=0.0)
Sentiment(polarity=0.0, subjectivity=0.0)
Sentiment(polarity=0.0, subjectivity=0.0)
Sentiment(polarity=0.0, subjectivity=0.0)
```

Hmm that's not quite enough

Not much sentiment in those headlines eh. Lets try and train a model to get a little more of these headlines.

In [9]:

```
import stock_news_classifier as sncl

cl = sncl.train_nb_v1()
```

```
Most Informative Features
            contains(Is) = True             sell : buy    =      3.5 : 1.0
            contains(to) = True              buy : sell   =      2.9 : 1.0
           contains(n't) = True             sell : hold   =      2.7 : 1.0
           contains(For) = True             hold : buy    =      2.2 : 1.0
         contains(Stock) = True             sell : hold   =      2.0 : 1.0
          contains(NYSE) = True             hold : sell   =      1.8 : 1.0
            contains(in) = True              buy : hold   =      1.8 : 1.0
           contains(Inc) = False             buy : sell   =      1.7 : 1.0
            contains(An) = True             sell : hold   =      1.6 : 1.0
         contains(Short) = True             sell : hold   =      1.6 : 1.0
None
('Accuracy', 0.6470588235294118)
```

Lets try the model on some more data

In [10]:
```
fb_news_url = "https://www.google.com/finance/company_news?q=NASDAQ%3AFB&ei=TY
RVWcCOEN3a2AaHhKq4Dg"
fb_headlines = getter.get_from_url(fb_news_url)

for line in fb_headlines:
    print(line, cl.classify(line))
```

```
(u'Facebook, Inc. Nears 2 Billion\xa0Users', u'hold')
(u'\nCEO Zuckerberg tweaks Facebook mission to focus on groups', u'buy')
(u'\nFacebook Fine Tunes Mission to Focus on Groups', u'buy')
(u"Facebook, Inc. Hits 2 Billion Users -- But It Won't Stop\xa0There", u'hol
d')
(u'\nFacebook Inc Hits Another Milestone. Should You Buy FB Stock?', u'buy')
(u'\nFacebook hits 2 billion-user mark, doubling in size since 2012', u'buy')
(u"Facebook Inc, Apple Inc And Amazon.com Inc - Today's Technical Stock Tradi
ng\xa0Ideas", u'hold')
(u'Facebook Says Internet Drone Lands Successfully on Second\xa0Test', u'hol
d')
(u'\nFacebook Inc (FB) Aquila Internet Drone Completes Successful Test Fligh
t', u'buy')
(u"\nFacebook Inc (NASDAQ:FB)'s Aquila drone completed its second flight",
 u'buy')
(u'Better Buy: Facebook, Inc. vs. Line\xa0Corp', u'buy')
(u'Facebook, Amazon, Netflix and Apple Get Thrashed as Broad Tech Sector Sell
off\xa0...', u'hold')
(u'Do Apple and Facebook have what it takes to succeed in\xa0TV?', u'buy')
(u'3 Stocks to Watch on Wednesday: Facebook Inc (FB), AeroVironment, Inc. (AV
AV\xa0...', u'buy')
(u'The Next Facebook Inc (FB) Stock Driver? Small\xa0Businesses.', u'buy')
(u'Better Buy: Facebook, Inc. vs.\xa0Google', u'buy')
(u'\n2', u'hold')
(u'\n3', u'hold')
(u'\n4', u'hold')
(u'\n5', u'hold')
(u'\n6', u'hold')
(u'\n7', u'hold')
(u'\n8', u'hold')
(u'\n9', u'hold')
(u'\n10', u'hold')
(u'Next', u'hold')
```

Lemme tize it

Cleaning up training and text to be classified generally improves accuracy. Typical cleanup includes

- stop word removal
- lemmatization
- case conversion

In [11]:

```
print(sncl.preprocessing("To be or not to be those are the questions?"))
print(sncl.preprocessing("The quick brown foxes jumped over the fences"))
```

```
question ?
quick brown fox jumped fence
```

Same model with cleaner text

In [12]:

```
cl = sncl.train_nb_v2()
```

```
Most Informative Features
        contains(needle) = True             sell : buy    =      2.7 : 1.0
           contains(buy) = True              buy : hold   =      2.0 : 1.0
           contains(hit) = True             sell : hold   =      2.0 : 1.0
      contains(interest) = True             sell : hold   =      2.0 : 1.0
         contains(stock) = True             sell : hold   =      2.0 : 1.0
         contains(alert) = True             sell : hold   =      2.0 : 1.0
         contains(short) = True             sell : hold   =      2.0 : 1.0
          contains(nyse) = True             hold : buy    =      1.9 : 1.0
        contains(moving) = True             sell : buy    =      1.6 : 1.0
          contains(vetr) = True             sell : buy    =      1.6 : 1.0
None
('Accuracy', 0.625)
```

In [13]:
```
for line in fb_headlines:
    print(line, cl.classify(sncl.preprocessing(line)))
```

```
(u'Facebook, Inc. Nears 2 Billion\xa0Users', u'hold')
(u'\nCEO Zuckerberg tweaks Facebook mission to focus on groups', u'hold')
(u'\nFacebook Fine Tunes Mission to Focus on Groups', u'hold')
(u"Facebook, Inc. Hits 2 Billion Users -- But It Won't Stop\xa0There", u'hol
d')
(u'\nFacebook Inc Hits Another Milestone. Should You Buy FB Stock?', u'buy')
(u'\nFacebook hits 2 billion-user mark, doubling in size since 2012', u'hol
d')
(u"Facebook Inc, Apple Inc And Amazon.com Inc - Today's Technical Stock Tradi
ng\xa0Ideas", u'hold')
(u'Facebook Says Internet Drone Lands Successfully on Second\xa0Test', u'hol
d')
(u'\nFacebook Inc (FB) Aquila Internet Drone Completes Successful Test Fligh
t', u'buy')
(u"\nFacebook Inc (NASDAQ:FB)'s Aquila drone completed its second flight",
 u'buy')
(u'Better Buy: Facebook, Inc. vs. Line\xa0Corp', u'buy')
(u'Facebook, Amazon, Netflix and Apple Get Thrashed as Broad Tech Sector Sell
off\xa0...', u'hold')
(u'Do Apple and Facebook have what it takes to succeed in\xa0TV?', u'hold')
(u'3 Stocks to Watch on Wednesday: Facebook Inc (FB), AeroVironment, Inc. (AV
AV\xa0...', u'buy')
(u'The Next Facebook Inc (FB) Stock Driver? Small\xa0Businesses.', u'buy')
(u'Better Buy: Facebook, Inc. vs.\xa0Google', u'buy')
(u'\n2', u'hold')
(u'\n3', u'hold')
(u'\n4', u'buy')
(u'\n5', u'hold')
(u'\n6', u'hold')
(u'\n7', u'hold')
(u'\n8', u'hold')
(u'\n9', u'hold')
(u'\n10', u'hold')
(u'Next', u'hold')
```

Decision Tree instead of Naives Bayes

In [14]:
```
cl = sncl.train_dtree_v2()
```

```
('Accuracy', 1.0)
```

Sometimes perfect is not a good thing

Ok so the model above turns out to have an accuracy of 1.0 which means it classified everything in the test data set correctly. Normally this would be a good thing. But in the early stages of training an ML model a perfect or very high score is a sure sign that something is wrong. High scores early on mean that the model is 'overfitting'. Which means it is picking up on some random noise that might be present equally in the training and test data set vs. picking up on true signals that matter. There are many pausible causes for this but a good place to start is to check your training & test data-set and increase the volume and variety of one or both. Ok that said lets see how the 'perfect' model does on real data set.

In [15]:
```
for line in fb_headlines:
    print(line, cl.classify(sncl.preprocessing(line)))
```

```
(u'Facebook, Inc. Nears 2 Billion\xa0Users', u'buy')
(u'\nCEO Zuckerberg tweaks Facebook mission to focus on groups', u'buy')
(u'\nFacebook Fine Tunes Mission to Focus on Groups', u'buy')
(u"Facebook, Inc. Hits 2 Billion Users -- But It Won't Stop\xa0There", u'hol
d')
(u'\nFacebook Inc Hits Another Milestone. Should You Buy FB Stock?', u'hold')
(u'\nFacebook hits 2 billion-user mark, doubling in size since 2012', u'hol
d')
(u"Facebook Inc, Apple Inc And Amazon.com Inc - Today's Technical Stock Tradi
ng\xa0Ideas", u'hold')
(u'Facebook Says Internet Drone Lands Successfully on Second\xa0Test', u'bu
y')
(u'\nFacebook Inc (FB) Aquila Internet Drone Completes Successful Test Fligh
t', u'buy')
(u"\nFacebook Inc (NASDAQ:FB)'s Aquila drone completed its second flight",
 u'buy')
(u'Better Buy: Facebook, Inc. vs. Line\xa0Corp', u'buy')
(u'Facebook, Amazon, Netflix and Apple Get Thrashed as Broad Tech Sector Sell
off\xa0...', u'buy')
(u'Do Apple and Facebook have what it takes to succeed in\xa0TV?', u'buy')
(u'3 Stocks to Watch on Wednesday: Facebook Inc (FB), AeroVironment, Inc. (AV
AV\xa0...', u'buy')
(u'The Next Facebook Inc (FB) Stock Driver? Small\xa0Businesses.', u'buy')
(u'Better Buy: Facebook, Inc. vs.\xa0Google', u'buy')
(u'\n2', u'buy')
(u'\n3', u'buy')
(u'\n4', u'buy')
(u'\n5', u'buy')
(u'\n6', u'buy')
(u'\n7', u'buy')
(u'\n8', u'buy')
(u'\n9', u'buy')
(u'\n10', u'buy')
(u'Next', u'buy')
```

Conclusion

So the neither model is quite at the point where you can bet even $10 on it. Which is to be expected since it is operating on a very small training data set (< 100 samples) and the classification model is using default parameters. Lets head back to the PPT to see what steps can help improve the model

In [ ]: