In [56]:
```
import json
from matplotlib import pyplot as plt
from collections import defaultdict
from sklearn import linear_model
import numpy
import random
import gzip
```

In [3]:
```
f = open("fantasy_10000.json")
dataset = []
for l in f:
    dataset.append(json.loads(l))
```

In [4]:
```
### Question 1
```

In [8]:
```
dataset[0]
```

Out[8]:
```
{'book_id': '18245960',
 'date_added': 'Sun Jul 30 07:44:10 -0700 2017',
 'date_updated': 'Wed Aug 30 00:00:26 -0700 2017',
 'n_comments': 1,
 'n_votes': 28,
 'rating': 5,
 'read_at': 'Sat Aug 26 12:05:52 -0700 2017',
 'review_id': 'dfdbb7b0eb5a7e4c26d59a937e2e5feb',
 'review_text': 'This is a special book. It started slow for about the
first third, then in the middle third it started to get interesting, th
en the last third blew my mind. This is what I love about good science
fiction - it pushes your thinking about where things can go. \n It is a
2015 Hugo winner, and translated from its original Chinese, which made
it interesting in just a different way from most things I\'ve read. For
instance the intermixing of Chinese revolutionary history - how they ke
pt accusing people of being "reactionaries", etc. \n It is a book about
science, and aliens. The science described in the book is impressive -
its a book grounded in physics and pretty accurate as far as I could te
ll. Though when it got to folding protons into 8 dimensions I think he
was just making stuff up - interesting to think about though. \n But wh
at would happen if our SETI stations received a message - if we found s
omeone was out there - and the person monitoring and answering the sign
al on our side was disillusioned? That part of the book was a bit dark
- I would like to think human reaction to discovering alien civilizatio
n that is hostile would be more like Enders Game where we would band to
gether. \n I did like how the book unveiled the Trisolaran culture thro
ugh the game. It was a smart way to build empathy with them and also un
derstand what they\'ve gone through across so many centuries. And who k
now a 3 body problem was an unsolvable math problem? But I still don\'t
get who made the game - maybe that will come in the next book. \n I lov
ed this quote: \n "In the long history of scientific progress, how many
protons have been smashed apart in accelerators by physicists? How many
neutrons and electrons? Probably no fewer than a hundred million. Every
collision was probably the end of the civilizations and intelligences i
n a microcosmos. In fact, even in nature, the destruction of universes
must be happening at every second--for example, through the decay of ne
utrons. Also, a high-energy cosmic ray entering the atmosphere may dest
roy thousands of such miniature universes...."',
 'started_at': 'Tue Aug 15 13:23:18 -0700 2017',
 'user_id': '8842281e1d1347389f2ab93d60773d4d'}
```

In [9]:
```
ratingCounts = defaultdict(int)
for d in dataset:
    ratingCounts[d['rating']] += 1
```

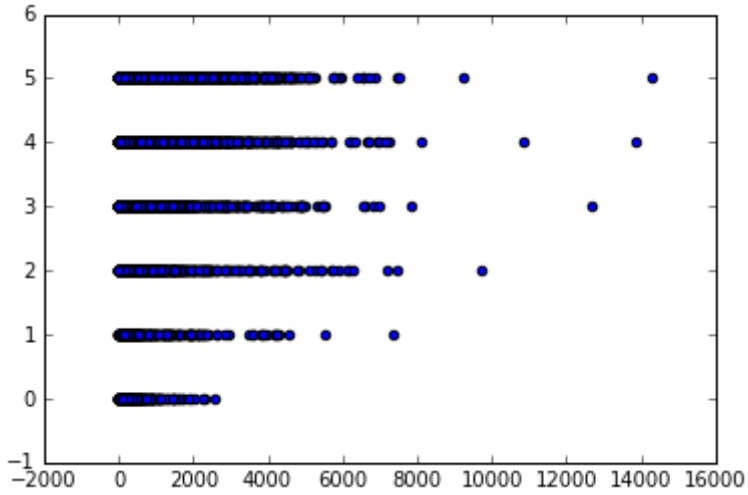
In [10]:
```
ratingCounts
```

Out[10]:
```
defaultdict(int, {0: 326, 1: 286, 2: 778, 3: 2113, 4: 3265, 5: 3232})
```

In [12]:
```
X = []
Y = []
for d in dataset:
    X.append(len(d['review_text']))
    Y.append(d['rating'])
```

In [16]:
```
plt.scatter(X,Y)
```

Out[16]:
```
<matplotlib.collections.PathCollection at 0x7f22210e0ba8>
```




In [17]:
```
### Question 2
```

In [19]:
```
def feature(datum):
    return [1] + [len(datum['review_text'])]
```

In [20]:
```
X = [feature(d) for d in dataset]
Y = [d['rating'] for d in dataset]
```

In [23]:
```
theta,residuals,rank,s = numpy.linalg.lstsq(X,Y)
```

```
/usr/local/lib/python3.5/dist-packages/ipykernel_launcher.py:1: FutureW
arning: `rcond` parameter will change to the default of machine precisi
on times ``max(M, N)`` where M and N are the input matrix dimensions.
To use the future default and silence this warning we advise to pass `r
cond=None`, to keep using the old, explicitly pass `rcond=-1`.
  """Entry point for launching an IPython kernel.
```

In [22]:
```
theta
```

Out[22]:
```
array([3.68568136e+00, 6.87371675e-05])
```

In [24]:
```
residuals
```

Out[24]:
```
array([15522.08662236])
```

In [26]:
```
#MSE
residuals[0] / len(Y)
```

Out[26]:
```
1.5522086622355318
```

In [27]:
```
### Question 3
```

In [28]:
```
def feature(datum):
    return [1] + [len(datum['review_text']), datum['n_comments']]
```

In [29]:
```
X = [feature(d) for d in dataset]
Y = [d['rating'] for d in dataset]
```

In [30]:
```
theta,residuals,rank,s = numpy.linalg.lstsq(X,Y)
```

```
/usr/local/lib/python3.5/dist-packages/ipykernel_launcher.py:1: FutureW
arning: `rcond` parameter will change to the default of machine precisi
on times ``max(M, N)`` where M and N are the input matrix dimensions.
To use the future default and silence this warning we advise to pass `r
cond=None`, to keep using the old, explicitly pass `rcond=-1`.
  """Entry point for launching an IPython kernel.
```

In [31]:
```
theta
```

Out[31]:
```
array([ 3.68916737e+00,  7.58407490e-05, -3.27928935e-02])
```

In [32]:
```
residuals[0] / len(Y)
```

Out[32]:
```
1.5498351692774528
```

In [33]:
```
### Question 4
```

In [34]:
```
def feature(datum, deg, norm):
    feat = [1]
    for i in range(1, deg + 1):
        feat.append((len(datum['review_text'])/norm)**i)
    return feat
```

In [35]:
```
norm = max([len(d['review_text']) for d in dataset])
```

In [36]:
```
norm
```

Out[36]:
```
14306
```

In [38]:
```
for i in range(1,6):
    X = [feature(d, i, norm) for d in dataset]
    Y = [d['rating'] for d in dataset]
    theta,residuals,rank,s = numpy.linalg.lstsq(X,Y)
    print("Degree " + str(i) + ":")
    print("  theta = " + str(theta))
    print("  MSE = " + str(residuals[0] / len(Y)))
```

```
Degree 1:
  theta = [3.68568136 0.98335392]
  MSE = 1.5522086622355318
Degree 2:
  theta = [ 3.65975869  1.8395413  -2.62503319]
  MSE = 1.5506567696339304
Degree 3:
  theta = [ 3.63659658  2.8884065  -8.48042966  6.12504475]
  MSE = 1.5497985323805634
Degree 4:
  theta = [  3.64736873   2.20419719  -1.80763945 -11.6451833   12.2184
4408]
  MSE = 1.5496291324524694
Degree 5:
  theta = [  3.6441158    2.47396326  -5.65441081   5.55309592 -15.9463
7484
  14.68100179]
  MSE = 1.5496142023298645

/usr/local/lib/python3.5/dist-packages/ipykernel_launcher.py:4: FutureW
arning: `rcond` parameter will change to the default of machine precisi
on times ``max(M, N)`` where M and N are the input matrix dimensions.
To use the future default and silence this warning we advise to pass `r
cond=None`, to keep using the old, explicitly pass `rcond=-1`.
  after removing the cwd from sys.path.
```

In [48]:
```
### Question 5
```

In [49]:
```
random.shuffle(dataset)
```

In [53]:
```
for i in range(1,6):
    X = [feature(d, i, norm) for d in dataset]
    Y = [d['rating'] for d in dataset]

    Xtrain = X[:len(X)//2]
    Xtest = X[len(X)//2:]
    Ytrain = Y[:len(Y)//2]
    Ytest = Y[len(Y)//2:]

    mod = linear_model.LinearRegression()
    mod.fit(Xtrain,Ytrain)
    Ypred = mod.predict(Xtest)
    MSE = sum([(yp - yt)**2 for (yp,yt) in zip(Ypred, Ytest)]) / len(Yte
st)
    print("Degree " + str(i) + ":")
    print("  test MSE = " + str(MSE))
```

```
Degree 1:
  test MSE = 1.5128199073717468
Degree 2:
  test MSE = 1.5141927174685252
Degree 3:
  test MSE = 1.5120767457172632
Degree 4:
  test MSE = 1.512084912578448
Degree 5:
  test MSE = 1.5112100377630662
```

In [54]:
```
### Question 7
```

In [60]:
```
f = open("beer_50000.json")
dataset = []
for l in f:
    if 'user/gender' in l:
        dataset.append(eval(l))
```

In [61]:
```
len(dataset)
```

Out[61]:
```
20403
```

In [63]:
```
X = [[1, len(d['review/text'])] for d in dataset]
y = [d['user/gender'] == 'Female' for d in dataset]
```

In [65]:
```
mod = linear_model.LogisticRegression()
mod.fit(X,y)
predictions = mod.predict(X) # Binary vector of predictions
correct = predictions == y # Binary vector indicating which predictions
 were correct
print(sum(correct) / len(correct))
```

```
0.9849041807577317
```

In [66]:
```
TP = [a and b for (a,b) in zip(predictions,y)] # etc.
sum(TP) / len(TP)
```

Out[66]: 0.0

In [67]:
```
mod = linear_model.LogisticRegression(class_weight='balanced')
mod.fit(X,y)
predictions = mod.predict(X) # Binary vector of predictions
correct = predictions == y # Binary vector indicating which predictions
 were correct
print(sum(correct) / len(correct))
```

```
0.4225849139832378
```

In [68]:
```
TP = [a and b for (a,b) in zip(predictions,y)] # etc.
sum(TP) / len(TP)
```

Out[68]: 0.00975346762730971