

```
In [2]: from sklearn.datasets import fetch_20newsgroups
data = fetch_20newsgroups()
```

```
In [3]: import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

```
In [6]: data.target_names
```

```
Out[6]: ['alt.atheism',
'comp.graphics',
'comp.os.ms-windows.misc',
'comp.sys.ibm.pc.hardware',
'comp.sys.mac.hardware',
'comp.windows.x',
'misc.forsale',
'rec.autos',
'rec.motorcycles',
'rec.sport.baseball',
'rec.sport.hockey',
'sci.crypt',
'sci.electronics',
'sci.med',
'sci.space',
'soc.religion.christian',
'talk.politics.guns',
'talk.politics.mideast',
'talk.politics.misc',
'talk.religion.misc']
```

```
In [7]: categories = ['alt.atheism',
'comp.graphics',
'comp.os.ms-windows.misc',
'comp.sys.ibm.pc.hardware',
'comp.sys.mac.hardware',
'comp.windows.x',
'misc.forsale',
'rec.autos',
'rec.motorcycles',
'rec.sport.baseball',
'rec.sport.hockey',
```

```
'sci.crypt',
'sci.electronics',
'sci.med',
'sci.space',
'soc.religion.christian',
'talk.politics.guns',
'talk.politics.mideast',
'talk.politics.misc',
'talk.religion.misc']
train = fetch_20newsgroups(subset="train",categories=categories)
test = fetch_20newsgroups(subset="test",categories=categories)
print(train.data[5])
```

From: dfo@vttoulu.tko.vtt.fi (Foxvog Douglas)
Subject: Re: Rewording the Second Amendment (ideas)
Organization: VTT
Lines: 58

In article <lr1eul\$4t@transfer.stratus.com> cdt@sw.stratus.com (C. D. Tavares) writes:
>In article <1993Apr20.083057.16899@ousrvr.oulu.fi>, dfo@vttoulu.tko.vtt.fi (Foxvog Douglas) writes:
>> In article <lqv87v\$4j3@transfer.stratus.com> cdt@sw.stratus.com (C. D. Tavares) writes:
>> >In article <C5n3GI.F8F@ulowell.ulowell.edu>, jrutledg@cs.ulowell.edu (John Lawrence Rutledge) writes:
>
>> >> The massive destructive power of many modern weapons, makes the
>> >> cost of an accidental or crimial usage of these weapons to great.
>> >> The weapons of mass destruction need to be in the control of
>> >> the government only. Individual access would result in the
>> >> needless deaths of millions. This makes the right of the people
>> >> to keep and bear many modern weapons non-existant.

>> >Thanks for stating where you're coming from. Needless to say, I
>> >disagree on every count.

>> You believe that individuals should have the right to own weapons of
>> mass destruction? I find it hard to believe that you would support a
>> neighbor's right to keep nuclear weapons, biological weapons, and nerve
>> gas on his/her property.

>> If we cannot even agree on keeping weapons of mass destruction out of
>> the hands of individuals, can there be any hope for us?

>I don't sign any blank checks.

Of course. The term must be rigidly defined in any bill.

>When Doug Foxvog says "weapons of mass destruction," he means CBW and
>nukes. When Sarah Brady says "weapons of mass destruction" she means
>Street Sweeper shotguns and semi-automatic SKS rifles.

I doubt she uses this term for that. You are using a quote allegedly
from her, can you back it up?

>When John
>Lawrence Rutledge says "weapons of mass destruction," and then immediately
>follows it with:

>>> The US has thousands of people killed each year by handguns,
>>> this number can easily be reduced by putting reasonable restrictions
>>> on them.

>...what does Rutledge mean by the term?

I read the article as presenting first an argument about weapons of mass
destruction (as commonly understood) and then switching to other topics.
The first point evidently was to show that not all weapons should be
allowed, and then the later analysis was, given this understanding, to
consider another class.

>cdt@rocket.sw.stratus.com --If you believe that I speak for my company,
>OR cdt@vos.stratus.com write today for my special Investors' Packet...

--
doug foxvog
douglas.foxvog@vtt.fi

```
In [10]: print(len(train.data))  
11314
```

```
In [11]: print(len(test.data))  
7532
```

```
In [13]: print(train.target[0:10])  
[ 7  4  4  1 14 16 13  3  2  4]
```

```
In [14]: from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.naive_bayes import MultinomialNB
from sklearn.pipeline import make_pipeline
model = make_pipeline(TfidfVectorizer(),MultinomialNB())
model.fit(train.data,train.target)
y_pred = model.predict(test.data)
```

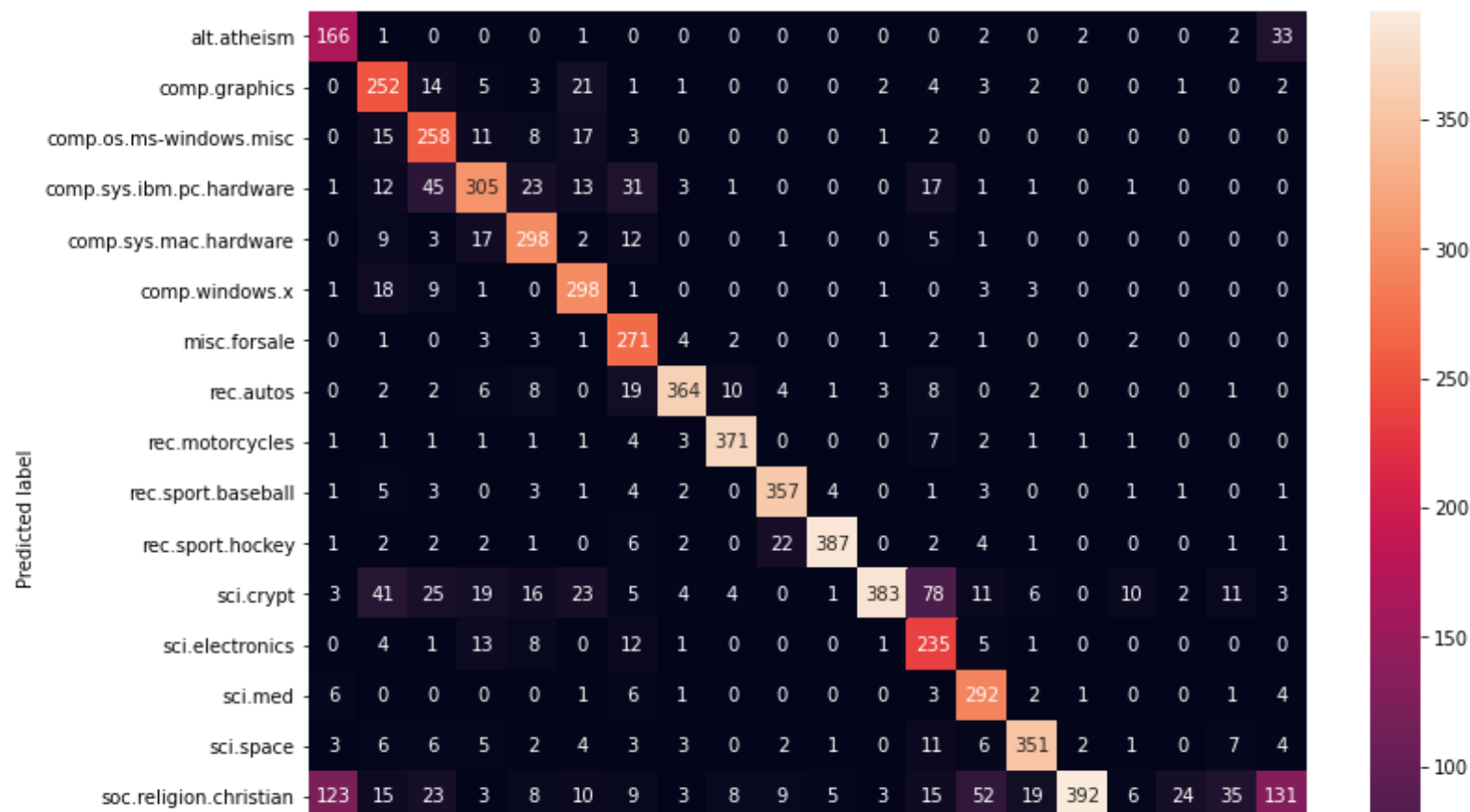
```
In [17]: from sklearn.metrics import confusion_matrix
cm = confusion_matrix(test.target,y_pred)
print(cm)
```

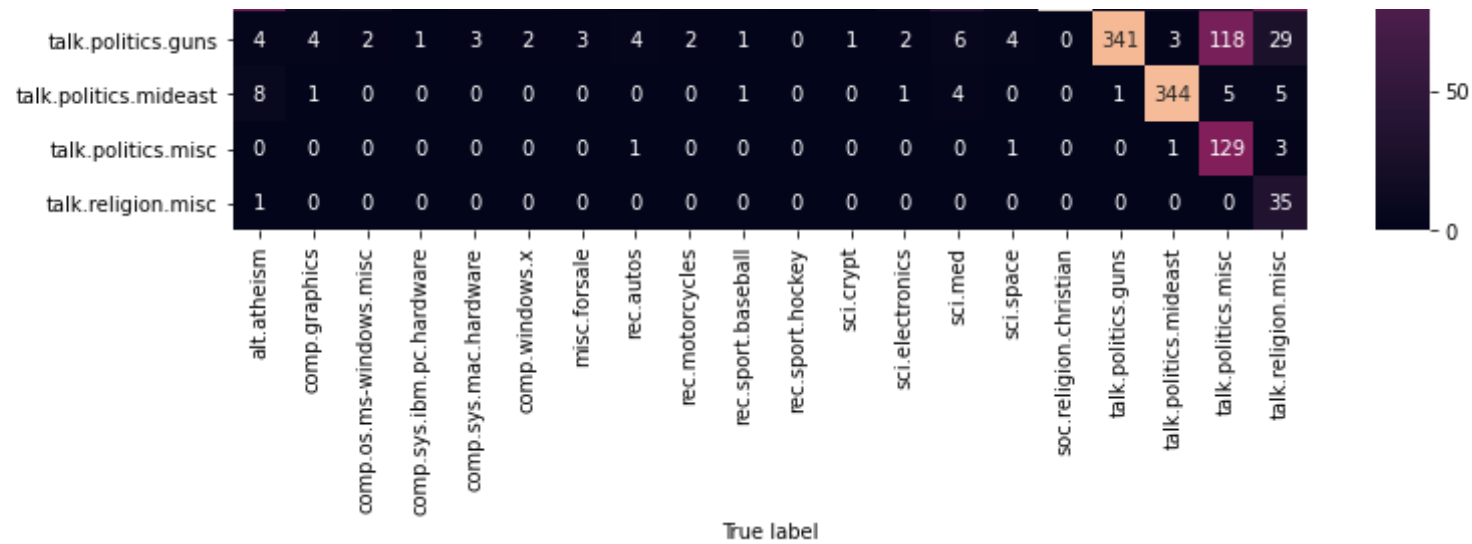
```
[[166  0  0  1  0  1  0  0  1  1  1  3  0  6  3 123  4  8
  0  1]
 [ 1 252 15 12  9 18  1  2  1  5  2 41  4  0  6 15  4  1
  0  0]
 [ 0 14 258 45  3  9  0  2  1  3  2 25  1  0  6 23  2  0
  0  0]
 [ 0  5 11 305 17  1  3  6  1  0  2 19 13  0  5  3  1  0
  0  0]
 [ 0  3  8 23 298  0  3  8  1  3  1 16  8  0  2  8  3  0
  0  0]
 [ 1 21 17 13  2 298  1  0  1  1  0 23  0  1  4 10  2  0
  0  0]
 [ 0  1  3 31 12  1 271 19  4  4  6  5 12  6  3  9  3  0
  0  0]
 [ 0  1  0  3  0  0  4 364  3  2  2  4  1  1  3  3  4  0
  1  0]
 [ 0  0  0  1  0  0  2 10 371  0  0  4  0  0  0  8  2  0
  0  0]
 [ 0  0  0  0  1  0  0  4  0 357 22  0  0  0  2  9  1  1
  0  0]
 [ 0  0  0  0  0  0  0  1  0  4 387  1  0  0  1  5  0  0
  0  0]
 [ 0  2  1  0  0  1  1  3  0  0  0 383  1  0  0  3  1  0
  0  0]
 [ 0  4  2 17  5  0  2  8  7  1  2 78 235  3 11 15  2  1
  0  0]
 [ 2  3  0  1  1  3  1  0  2  3  4 11  5 292  6 52  6  4
  0  0]
 [ 0  2  0  1  0  3  0  2  1  0  1  6  1  2 351 19  4  0
  1  0]
 [ 2  0  0  0  0  0  0  0  1  0  0  0  0  1  2 392  0  0
  0  0]
```

```
[ 0  0  0  1  0  0  2  0  1  1  0 10  0  0  1  6 341  1
 0  0]
[ 0  1  0  0  0  0  0  0  0  1  0  2  0  0  0 24  3 344
 1  0]
[ 2  0  0  0  0  0  0  1  0  0  1 11  0  1  7 35 118  5
129 0]
[33  2  0  0  0  0  0  0  0  1  1  3  0  4  4 131 29  5
 3 35]]
```

```
In [20]: plt.figure(figsize=(12,10))
sns.heatmap(cm.T,annot=True,fmt="d",xticklabels=train.target_names,yticklabels=train.target_names)
plt.xlabel("True label")
plt.ylabel("Predicted label")
```

Out[20]: Text(86.99999999999999, 0.5, 'Predicted label')





```
In [21]: def predict_category(string_input,train=train,model=model):
         pred = model.predict([string_input])
         return train.target_names[pred[0]]
```

```
In [22]: predict_category("BMW is better tha Audi")
```

```
Out[22]: 'rec.autos'
```

```
In [23]: predict_category("President of India")
```

```
Out[23]: 'talk.politics.misc'
```

```
In [24]: predict_category("Sending to space")
```

```
Out[24]: 'sci.space'
```

```
In [ ]:
```