Document Classification

Points: 0 Rank: 129204

Your Document Classification submission got 98.53 points. Share Post

Try the next challenge | Try a Random Challenge

Problem | Submissions | Leaderboard | Editorial

You have been given a stack of documents that have already been processed and some that have not. Your task is to classify these documents into one of eight categories: [1,2,3,...8]. You look at the specifications on what a document must contain and are baffled by the jargon. However, you notice that you already have a large amount of documents which have already been correctly categorize (training data). You decide to use Machine Learning on this data in order to categorize the uncategorized documents.

Training Data

In order to figure out what category each document should fall under you will base it on the categories of the documents in the "trainingdata.txt" file. This file will be included with your program at runtime and will be named "trainingdata.txt".

The file is formatted as follows:

The first line contains the number of lines that will follow.

Each following line will contain a number (1-8), which is the category number. The number will be followed by a space then some space seperated words which is the processed document.

Input

The first line in the input file will contain T the number of documents. T lines will follow each containing a series of space seperated words which represents the processed document.

Output

For each document output a number between 1-8 which you believe this document should be categorized as.

Sample Input

3
This is a document
this is another document
documents are seperated by newlines

Sample Output

1
4
8

Scoring

Your score for this challenge will be 100* (#correctly categorized - #incorrectly categorized)/(T).

Change Theme Language Python 3

```
import numpy as np

l = [
    ('Business means risk!', 1),
    ("This is a document", 1),
    ("this is another document", 4),
    ("documents are separated by newlines", 8)
]
```

```
13
14  def load_data(filename):
15      with open(filename, 'r') as data_file:
16          s = int(data_file.readline())
17          X = np.zeros(s, dtype=np.object_)
18          Y = np.zeros(s, dtype=np.int_)
19          for i, line in enumerate(data_file):
20              ind = line.index(' ')
21              if ind == -1:
22                  raise ValueError('invalid input file')
23              targ = int(line[:ind])
24              words = line[ind + 1:]
25              X[i] = words
26              Y[i] = targ
27      return X, Y
28
29  # Load training data
30  X, Y = load_data('trainingdata.txt')
31
32  c = pipeline.Pipeline([
33      ('vect', text.TfidfVectorizer(
34          stop_words='english', ngram_range=(1, 1), min_df=4,
35          strip_accents='ascii', lowercase=True)),
36      ('clf', linear_model.SGDClassifier(class_weight='balanced'))
37  ])
38  model = c.fit(X, Y)
39
40  t = list(line for line in sys.stdin)[1:]
41
42  for y, x in zip(model.predict(t), t):
43      for pattern, targ in l:
44          if pattern in x:
```

Line: 49 Col: 1

Upload Code as File Test against custom input Run Code Submit Code

Congratulations

You solved this challenge. Would you like to challenge your friends?

Next Challenge

Test case 0

Test case 1

Test case 2

Test case 3

Compiler Message

Success

Hidden Test Case

Unlock this testcase for 5 hackos.

Unlock