MOVIE REVIEWS CLASSIFICATION

Team members

Kusuma.P Supritha.N A.Swathi Apoorva.RK

Abstract

The entertainment industry requires new and better ways to target specific users with certain features. This project is exploring the possibility of classifying the movie review corpus as positive or negative to help enable make better decisions for target users, using data mining techniques.

The sample data set given has files with movie review comments which are already tagged as positive or negative based on the review comments. Create a machine learning model or classifier using naïve bayes technique that can be used to classify new movie reviews.

Requirements

1. Feature extraction
2. Identify the training, validation and test data
3. Create and train the model using training data
4. Validate the model using validation data
5. Verify the model with test data
6. Calculate the accuracy of classification

Design

This project is based on text classification. In text classification, we are given a description **d Ɛ X** of a document, where **X** is the document space; and a fixed set of classes C = { c1, c2, … ,cn}. Classes are also called categories or labels. Typically, the document space X is some type of high-dimensional space, and the classes are human defined for the needs of an application. We are given a training set of labelled documents or text D. In this project the classes C = {Positive, Negative} and the document space is the movie review

Code

```
#!/usr/bin/python


import os, sys
#import shutil
import nltk
import random
file_paths1=[]
file_paths=[]
count = {}


DIR = r"C:\Users\vijeth\AppData\Local\Programs\Python\Python35-32\pos"
for root,directories,files in os.walk(DIR):
   for filename in files:
     filepath=os.path.join(root,filename)
     file_paths.append(filepath)


all_words=[]
lnames=[]
lpos=[[[],'pos']]
for p in file_paths:
  lnames=open(p,'r').read().split()
  lpos.append([lnames,'pos'])
  for w in lnames:
    all_words.append(w)
#print(lpos[1])



DIR1 = r"C:\Users\vijeth\AppData\Local\Programs\Python\Python35-32\neg"
for root,directories,files in os.walk(DIR1):
  for filename in files:
   filepath1=os.path.join(root,filename)
   file_paths1.append(filepath1)

for q in file_paths1:
  lnames=open(q,'r').read().split()
  lpos.append([lnames,'neg'])
  for w in lnames:
    all_words.append(w)
#print(lpos[1])
```

```
random.shuffle(lpos)
print(len(all_words))
#print(lpos)

word_features=list(all_words)[:2000]

#print(all_words)
def document_features(document):
    document_words = set(document)
    features = {}
    for word in word_features:
        features['contains(%s)' % word] = (word in document_words)
    return features

featuresets=[(document_features(d),c) for (d,c) in lpos]
train_set,test_set=featuresets[5:],featuresets[:5]

classifier = nltk.NaiveBayesClassifier.train(train_set)
print (nltk.classify.accuracy(classifier, test_set))
classifier.show_most_informative_features(5)
```

Conclusions and challenges

In this study, an experiment is conducted on Movie Review dataset. The Naive Bayes classifier is used to train dataset. Movie review is classified by using Naive Bayes, Naive Bayes Neural classifier. Accuracy of sentiment analysis is increased by proposed system from dependence and independence assumptions among features.

References

https://docs.python.org/3/tutorial/inputoutput.html.

http://www.nltk.org/book

http://www.stackoverflow.com