# CS 590 Machine Learning

# Homework 02

# (Ashwini Kulkarni)

**Part I**

Let’s consider we have 9 files,

Get their url, store it in variable and fetch data from pages using urllib to stored actual data from raw files.

Let’s consider there are 10 categories from w1, w2,w3,…w10 (w1…w10 are attributes)

>>> categories = [w1, w2,w3..,w3]

We can now load the list of files matching above categories

>>> from sklearn.datasets import fetch\_urldata

>>> wiki\_array = fetch\_urldata (subset='wiki',

... categories=categories, shuffle=True, random\_state=42)

Setting targets, lets consider we have either sports or flower domain so target will be either Sports or Flowers.

>>> wiki\_array.target\_names =[‘Sports’, ‘flowers’]

Getting target value for each document using,

>>> for t in wiki\_array.target[:10]:

... print(wiki\_array.target\_names[t])

Also using fit\_transform from Sklearn we can get vector form representation of document w.r.t. all categories .(Output like [[1 1 0 1 …………………..])

>>> from sklearn.feature\_extraction.text import CountVectorizer

>>> vectorizer = CountVectorizer(binary=True, stop\_words='english')

>>> print vectorizer.fit\_transform(wiki\_array.data).todense()

So if we have **less** features we can just get idea by looking their vector format, which one are similar.

But for doing more automated,

We can define function

def cosine\_sim(text1, text2):

... tfidf = vectorizer.fit\_transform([text1, text2])

... return ((tfidf \* tfidf.T).A)[0,1]

If both documents are exact same then it will return 1 else it will calculate similarity factor 1 to 0.



**Part II**

Understanding Naïve base algorithm for this problem

**Using Naïve Bayes Algorithm,**

Let consider we have 9 no. of html document with out of them 5 are sports related, and other are flower related.

10 categories such as w1, w2, w3… w10.

So each document we can represent with vector of size 10.

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Doc. | W1 | W2 | W3 | W4 | W5 | W6 | W7 | W8 | W9 | W10 | Class |
| 1 | 1 |  |  |  |  | 1 |  | 1 |  | 1 | A |
| 2 |  | 1 |  | 1 |  |  |  |  |  |  | B |
| 3 | 1 |  |  |  | 1 | 1 |  |  | 1 |  | A |
| 4 | 1 |  |  |  | 1 |  |  | 1 | 1 | 1 | A |
| 5 | 1 |  |  |  |  | 1 |  | 1 |  |  | A |
| 6 |  | 1 |  | 1 |  |  |  | 1 | 1 |  | B |
| 7 |  |  | 1 |  |  |  |  | 1 | 1 |  | B |
| 8 |  |  |  |  |  | 1 | 1 | 1 |  | 1 | B |
| 9 | 1 |  |  |  |  | 1 |  | 1 |  | 1 | A |

Note:-

Document related to Sports Classified as ‘A’

Document related to Flower Classified as ‘B’

P(A)= 5/9 = 0.55

P(B)=4/9 = 0.44

Computing e word likelihoods using for class(A)

P(w1|A),P(w2|A),…P(w10|A)

Using formula,

P(w|A)= Nk + 1 / N+(total no of words)

Nk  is how many time w is occur.

N are no. of words in class(A) occur (i.e. 20)

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Doc. | W1 | W2 | W3 | W4 | W5 | W6 | W7 | W8 | W9 | W10 | Class |
| 1 | 1 |  |  |  |  | 1 |  | 1 |  | 1 | A |
| 3 | 1 |  |  |  | 1 | 1 |  |  | 1 |  | A |
| 4 | 1 |  |  |  | 1 |  |  | 1 | 1 | 1 | A |
| 5 | 1 |  |  |  |  | 1 |  | 1 |  |  | A |
| 9 | 1 |  |  |  |  | 1 |  | 1 |  | 1 | A |

Calculations:

P(w1|A)= 5 +1 /20+10 =6/30= 0.2

P(w2|A) = P(w3|A) = P(w4|A) =P(w7|A)= 0+1 /20+10=1/30=0.03

P(w5|A) = P(w9|A)= 2+1/20+10 =3/30 = 0.1

P(w6|A)= 4+1/20+10 =5/30 = 0.167

P(w8|A)=4+1/20+10 =5/30 =0.167

P(w10|A)=3+1/20+10=4/30 =0.13

Similarly, likelihood of each word for class(B)

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Doc. | W1 | W2 | W3 | W4 | W5 | W6 | W7 | W8 | W9 | W10 | Class |
| 2 |  | 1 |  | 1 |  |  |  |  |  |  | B |
| 6 |  | 1 |  | 1 |  |  |  | 1 | 1 |  | B |
| 7 |  |  | 1 |  |  |  |  | 1 | 1 |  | B |
| 8 |  |  |  |  |  | 1 | 1 | 1 |  | 1 | B |

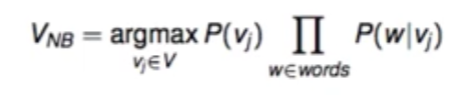
P(w1|B) =P(w5|B) =0+1/13+10 = 1/23 = 0.043

P(w2|B)=P(w4|B) = P(w9|B)=2+1/13+10 =3/23 = 0.13

P(w3|B) =P(w6|B) =P(w7|B)=P(w10|B) =1+1/13+10 =2/23 = 0.086

P(w8|B)=3+1/13+10 =4/23 =0.17

Now lets classify new document with following formula for Naïve Bayes,



Where V is class

**Scenario I**

Lets condsider, new document have words w1, w3, w8, w10

So Calculating value for class A

VA = P(A) P(w1|A) P(w3|A) P(w8|A) P(w10|A)

= (0.55) \* (0.2) \*(0.03) \*(0.167) \*(0.13)

= 7.164 \* 10-5

VB =P(b) P(w1|B) P(w3|B) P(w8|B) P(w10|B)

= (0.44) \*(0.043) \*(0.086) \* (0.17) \* (0.086)

= 2.37 \* 10-5

Max(VA,VB) is VA so new document belong to class A i.e. its related to Sports. This is way to find out similarities between documents.

**Scenario II**

For New Document we may need to add new categories, So there we will need to add new category and calculate probability for that, we can consider least probability and calculate VA  and VB .

Coding Walkthrough =>



Pros:

* It is easy and fast to predict class of test data set. It also perform well in multi class prediction
* When assumption of independence holds, a Naive Bayes classifier performs better compare to other models like logistic regression and you need less training data.
* It performs well in case of categorical input variables compared to numerical variable(s). For numerical variable, normal distribution is assumed (bell curve, which is a strong assumption).

Cons:

* If categorical variable has a category (in test data set), which was not observed in training data set, then model will assign a 0 (zero) probability and will be unable to make a prediction. This is often known as “Zero Frequency”. To solve this, we can use the smoothing technique. One of the simplest smoothing techniques is called Laplace estimation.
* On the other side naive Bayes is also known as a bad estimator, so the probability outputs from predict probability are not to be taken too seriously.
* Another limitation of Naive Bayes is the assumption of independent predictors. In real life, it is almost impossible that we get a set of predictors which are completely independent.