**Homework 3**

**Karan Ashar – kashar**

**Contents**

**1: Create Word Features**

**2: Train Classifier 1 (BOW)**

**3: Run Classifier 1 on our data**

**4: Classifier 2**

**5: BernoulliNB Classifier**

**6: Summary Table**

**1: Create Word Features:**

I first imported the sentence\_polarity corpus. It consisted of sentences which were labeled as ‘pos’ or ‘neg’. We used this labeled data to train our models so that later on we could use the model to classify our unlabeled Amazon review text file. I extracted these sentences in a list called document. I randomly shuffled the document. I then extracted the words from this document and stored it in the list called word\_features. I selected most common 2000 words. I had started off with 3000 words, but the time taken to process the review text file was very long. By reducing the words, I could increase the processing speed by 30%.

**2: Train Classifier 1:**

I then defined a function document\_features which was nothing but the ‘Bag of Words’. This function returned features which were nothing but a dictionary which consisted of the word as a key and True or False as a value indicating the presence of the word. We then create a featuresets list which consisted of the features along with the category which they belong to.

I then divided the featuresets into training and testing sets in a ratio nearly equal to 91:9.

I used the nltk.NaiveBayesClassifier as my classification model. I trained the Classifier 1 on the training set and tested it on the test set. It resulted in an accuracy of 74%

**3: Run Classifier 1 on our data:**

Now the task was to apply my unlabeled data on the model. I imported my data and tokenized it based on the sentences because in the end we need two lists of positive and negative sentences. These sentences were stored in a list called ‘token’. I ran a loop iterating through the ‘token’ list.

Each sentence was first word tokenized because our word\_features function accepts a word tokenized list. I then used the classifier function to classify my sentence. The output of the classify function is one of the labels i.e ‘pos’ or ‘neg’. The output was stored in a variable ‘a’. I then checked the value in ‘a’. If it was ‘pos’ then it would be appended to the ‘p’ list else, it would be appended to the ‘n’ list. ‘p’ stood for Positive and ‘n’ for negative.

I then checked for the length of the two lists and found the negative list to be more than the positive list.

**4: Classifier 2:**

The second feature I used was detecting negation words. I created a list of many negation words like ’not’, ’no’, ‘never’, ‘hardly’ etc.

I created a NOT\_featuresets from the labeled document we had. I then divided the NOT\_featuresets into training and testing sets in a similar ratio as the previous step.

I then trained Classifier 2 using the NOT\_featuresets using the training set and then tested it on the testing set. I got an accuracy of 58.5%. This is very low.

I imported the stopwords list and the time package. The time package was to get an idea of the amount of time the classification task took.

This time I decided to process the data first. I thought this would improve the speed of classification because irrelevant words would not be used. So, after word tokenizing my sentence I filtered out Stop words and non-alpha words. I then used the classify function to classify the sentences. The classification task took me nearly 261 minutes.

I found the number of positive sentences to be more than the negative sentences by a huge number. This is because it has a low accuracy compared to our previous model.

**5: BernoulliNB:**

I then decided to explore one of the sklearn classifiers to check for accuracy. I imported the BernoulliNB from sklearn.naive\_bayes. It showed an accuracy of 73% on the first feature set and an accuracy of 58.5% on the second feature set.

**6: Summary Table:**

|  |  |
| --- | --- |
| **Positive Sentences** | **Negative Sentences** |
| 'This is a great tutu and at a really great price.'  'A++\r\nI bought this for my 4 yr old daughter for dance class, she wore it today for the first time and the teacher thought it was adorable.',  'Made well and cute on the girls.',  "So far, she's using it to play out her Cinderella dreams but I am sure we'll be able to use it for a recital sometime soon.",  'Great tutu for a great price.',  'It is well built and we hope she gets lots of wear out of it.', | 'REFUSES to make good on purchase...... Real creeps.',  'Company is a rip off.',  'Clearly plenty of room to grow.',  'Thank you Halo Heaven great product for Little Girls.',  'Not cheap materia!',  'It will be too short and small for older girls.',  'The product itself is five stars, however the description is faulty.', |

|  |  |
| --- | --- |
| **Method** | **Accuracy** |
| **Naïve-Bayes**  **‘BOW’** | **74%** |
| **Naïve-Bayes**  **NOT\_Features** | **58.5%** |
| **BernoulliNB**  **‘BOW’** | **73%** |
| **BernoulliNB**  **NOT\_Features** | **58.5%** |