A PROJECT REPORT ON

Sentiment Analysis on Text Restaurant Reviews

Submitted by

Pranav Kalikate

What is NLP (Natural Language Processing)?

NLP is a subfield of computer science and artificial intelligence concerned with interactions between computers and human (natural) languages. It is used to apply machine learning algorithms to text and speech. We can use NLP to create a system like speech recognition**,** document summarization**,** machine translation**,** spam detection**,** named entity recognition**,** question answering, autocomplete, predictive typing**.**

NLTK Library

Natural language toolkit contains a suite of text processing libraries for classification, tokenization, stemming, tagging, parsing, and semantic reasoning.

Basics of NLP

- **Sentence Tokenization (nltk.sent_tokenize)**
  Sentence tokenization (also called sentence segmentation) is the problem of dividing a string of written language into its component sentences.
- **Word Tokenization (nltk.word_tokenize)**
  Word tokenization (also called word segmentation) is the problem of dividing a string of written language into its component words
- **Text Lemmatization and Stemming**
  The goal of both stemming and lemmatization is to reduce inflectional forms and sometimes derivationally related forms of a word to a common base form.
  Stemming usually refers to a process that chops of the ends of words and Lemmatization remove inflectional endings only and return the base or dictionary form of a word, which is known as the lemma.
- **Stop words**
  Stop words are words which are filtered out before or after processing of text. When applying machine learning to text, these words can add a lot of noise. That's why we want to remove these irrelevant words. There is no single universal list of stop words. The list of the stop words can change depending on your application.
- **Regex**
  A regular expression, regex is a sequence of characters that define a search pattern. We can use regex to apply additional Filtering to our text. For example, we can remove all the non-words characters like punctuations.
- **Bag-of-words**
  The bag-of-words model is a popular and simple feature extraction technique used when we work with text. It describes the occurrence of each word within a document.
  Machine learning algorithms cannot work with raw text directly, we need to convert the text into vectors of numbers. This is called feature extraction. Any information about the order or structure of words is discarded. That's why it's called a bag of words.
- **CountVectorizer**
  CountVectorizer class from the sklearn library is used to create a vocabulary of known words or Bag of words model.
  The words are then scored to convert each raw text into a vector of numbers. After that, we can use these vectors as input for a machine learning model. The simplest scoring method is to mark the presence of words with 1 for present and 0 for absence.
- **Sparse Vectors**
  The scored vectors which have a lot of zeros are called sparse vectors.

Problem Description

Here I have a dataset ‘Restaurant Reviews’ which consists of the reviews written by the customers about the restaurant and I will try to build a Machine learning model to predict if the review is positive or negative. The algorithm which I will built will be very well applicable to other kinds of text, for example it can be applied on books to predict the genre of the book.

First ten rows of the dataset:

dataset - DataFrame

Index	Review	Liked
0	Wow... Loved this place.	1
1	Crust is not good.	0
2	Not tasty and the texture was just nasty.	0
3	Stopped by during the late May bank holiday off Rick Steve recommendation and loved it.	1
4	The selection on the menu was great and so were the prices.	1
5	Now I am getting angry and I want my damn pho.	0
6	Honeslty it didn't taste THAT fresh.)	0
7	The potatoes were like rubber and you could tell they had been made up ahead of time being kept under a warmer.	0
8	The fries were great too.	1
9	A great touch.	1
10	Service was very prompt.	1

In the dataset we have 1000 such rows of observations which I will be using to train the model and test its performance.

Steps in Building NLP:

- Importing the libraries
- Cleaning the text
- Create a Bag of words model
- Apply machine learning model (classification) onto Bag of words model.

Building a NLP

- **Importing the libraries and dataset**

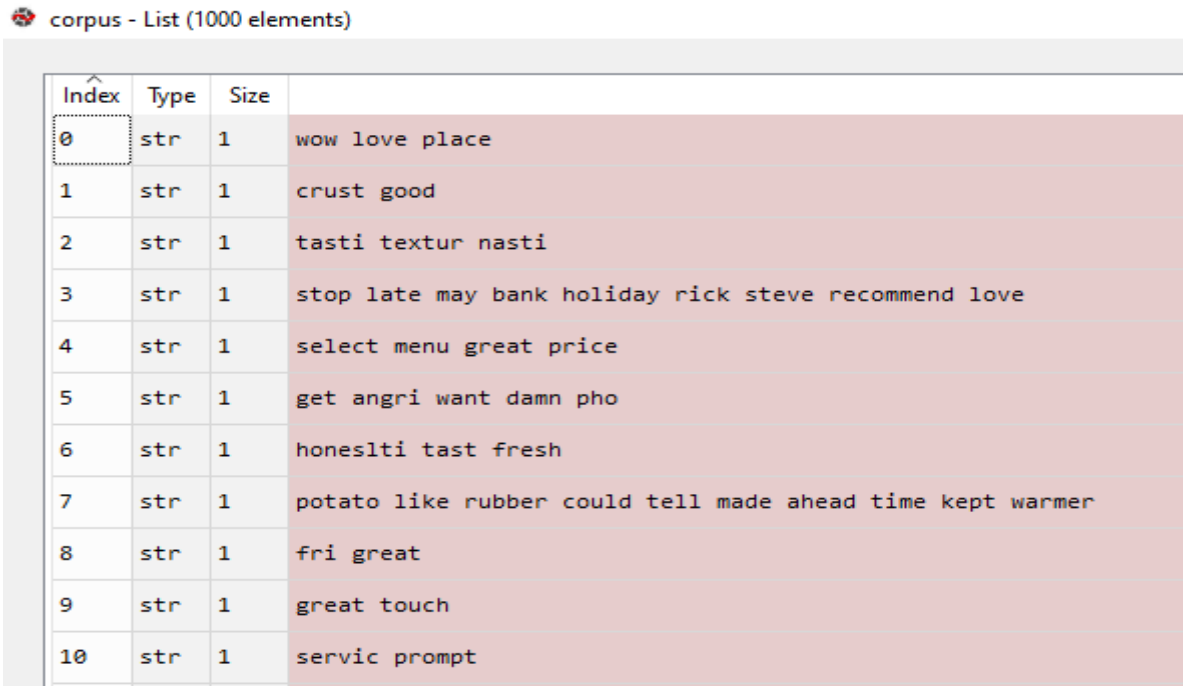
```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd

dataset = pd.read_csv('Restaurant_Reviews.tsv', delimiter = '\t', quoting = 3)
```

Quoting=3 will take care of double quotes.

- **Cleaning the texts to get rid of unnecessary words**

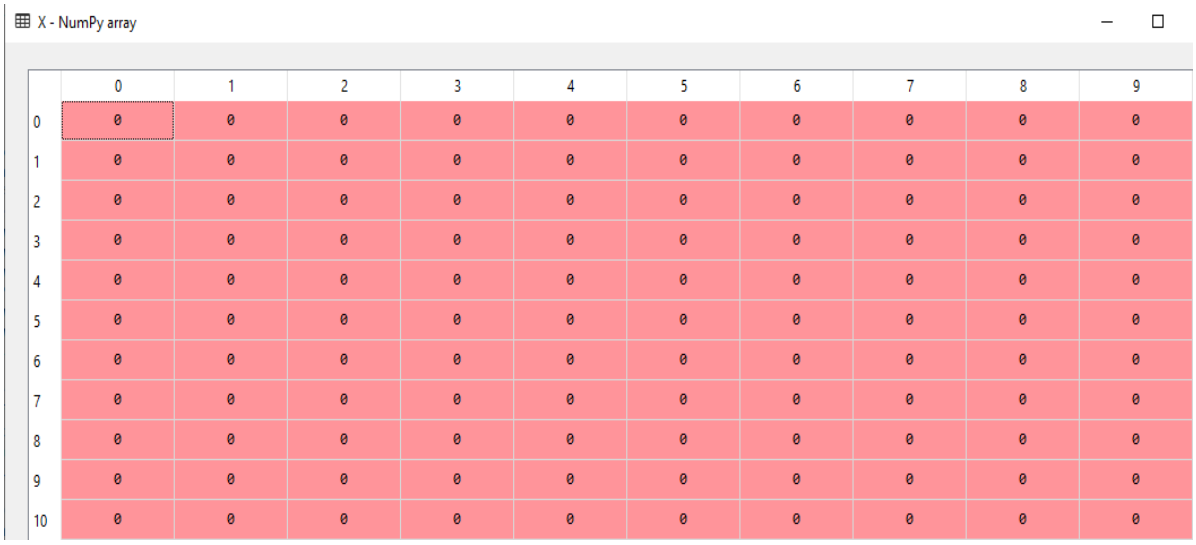
```
import re
import nltk
nltk.download('stopwords')
from nltk.corpus import stopwords
from nltk.stem.porter import PorterStemmer
corpus = []                                  #corpus is a collection of cleaned text
for i in range(0, 1000):                     #1000 observations
    review = re.sub('[^a-zA-Z]',' ', dataset['Review'][i])
    review = review.lower()
    review = review.split()
    ps = PorterStemmer()
    review = [ps.stem(word) for word in review if not word in set(stopwords.words('english'))]
    review = ' '.join(review)
    corpus.append(review)
```



Index	Type	Size	
0	str	1	wow love place
1	str	1	crust good
2	str	1	tasti textur nasti
3	str	1	stop late may bank holiday rick steve recommend love
4	str	1	select menu great price
5	str	1	get angri want damn pho
6	str	1	honeslti tast fresh
7	str	1	potato like rubber could tell made ahead time kept warmer
8	str	1	fri great
9	str	1	great touch
10	str	1	servic prompt

- **Creating the Bag of Words model**

```
from sklearn.feature_extraction.text import CountVectorizer
cv = CountVectorizer(max_features = 1500)
X = cv.fit_transform(corpus).toarray()        #Sparse matrix
y = dataset.iloc[:, 1].values
```



	0	1	2	3	4	5	6	7	8	9
0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0

- **Splitting the dataset into the Training set and Test set**

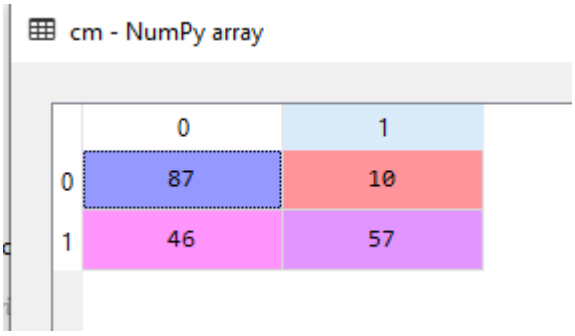
```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.20, random_state = 0)
```

- **Fitting Random Forest Classification to the Training set**

```
from sklearn.ensemble import RandomForestClassifier
classifier = RandomForestClassifier(n_estimators = 10, criterion = 'entropy', random_state = 0)
classifier.fit(X_train, y_train)
#out of all classifiers, Random forest works best
```

- **Predicting the Test set results and making the Confusion Matrix**

```
y_pred = classifier.predict(X_test)
from sklearn.metrics import confusion_matrix
cm = confusion_matrix(y_test, y_pred)
```



	0	1
0	87	10
1	46	57

Accuracy on test set = 72 %.