LP2.

Assignment 4

Date of Completion:- 24.9.2020

Date of Submission:- 29.9.2020

Title:- Text analysis.

Problem Statement:- Consider a suitable text dataset. Remove stop words, apply stemming and feature selection ~~techn~~ techniques to represent documents as vectors. Classify documents and evaluate precision recall.

Learning Objectives:- To understand the process of stemming, calculating precision and recall.

Learning Outcomes:- Students will be able to understand stop words, stemming, feature selection techniques and calculate precision and recall.

Software / Hardware requirements:- ~~NLTK~~ ~~NLTK~~ NLTK data / packages, python, Anaconda IDE.

Theory:-

Stop Words

1) The most commonly used words in a language.
2) These words are filtered out before or after the

natural language data (text) is processed.

3) They do not add much meaning to a sentence.
   Eg. as, the, at, a etc.

Stemming:-

1) A process of producing morphological variant of the root/base word.
2) A word is reduced to its root form.
   Eg loved → love
   played → play
   eating → eat.

Precision and Recall:-

1) precision = $\frac{\text{Number of correct triples}}{\text{Number of triples retrieved}} = \frac{T_p}{T_p + f_p}$ $T_p \Rightarrow$ true positive, $F_p \Rightarrow$ false positive
2) recall = $\frac{\text{Number of correct triples}}{\text{Number of triples in gold set}} = \frac{T_p}{T_p + f_n}$ $f_n \Rightarrow$ false negative

3) A measure of success of prediction when classes are very imbalance.
4) Precision is measure of result relevancy
5) Recall is measure of how many truly relevant results are returned.

Algorithm:-

1. Import python different packages numpy, pandas, nltk (natural language toolkit), re (RegEx)
2. read the dataset.
3. remove the stopwords and perform stemming on the words from the dataset and form a

corpus (dataset with no stop word and stemmed words)
4. Prepare or vectorize the words. dataset (feature extraction)
5. Split the data in training and test set.
6. Fit classifier model on the dataset (Naive Bayes)
7. Predict value on the test data.
8. Build a confusion matrix
9. Calculate precision and recall.

Conclusion:-

Dataset Used:- Restaurant Reviews.

Conclusion:- Thus I have completed this assignment and understood how to calculate precision recall and analyze text.

CODE

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

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

# Cleaning the texts
import re
import nltk
nltk.download('stopwords')
from nltk.corpus import stopwords
from nltk.stem.porter import PorterStemmer
corpus = []
for i in range(0, 1000):
    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)

# Creating the Bag of Words model
from sklearn.feature_extraction.text import CountVectorizer
cv = CountVectorizer(max_features = 1500)
X = cv.fit_transform(corpus).toarray()
y = dataset.iloc[:, 1].values

# 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 Naive Bayes to the Training set
from sklearn.naive_bayes import GaussianNB
classifier = GaussianNB()
classifier.fit(X_train, y_train)
```

```
# Predicting the Test set results
y_pred = classifier.predict(X_test)

# Making the Confusion Matrix
from sklearn.metrics import confusion_matrix
cm = confusion_matrix(y_test, y_pred)

recall = np.diag(cm) / np.sum(cm, axis = 1)
precision = np.diag(cm) / np.sum(cm, axis = 0)

print('Precision of the model = ',precision[0])
print('Recall of the model = ',recall[0])
```

OUTPUT

Precision of the model = 0.8208955223880597

Recall of the model = 0.5670103092783505