Manas Jha

RA1911003010643

Aim: Implementation of NLP programs

Problem Formulation- Solving a dataset using NLP.

Problem Statement- Using NLP, create a spam classifier to determine if a particular SMS is spam or not.

Algorithm-

Language is broken down into pieces or tokens that machines can understand. SMS gathered from various sources will be tokenized and then evaluated and classed as spam or not, as in this problem. After modifying the data with NLP, the classification problem is handled with Nave Bayes. The Nave Bayes classifiers are a set of Bayes' Theorem-based classification methods. It is a family of algorithms that share a similar idea, namely that each pair of characteristics being categorized is independent of the others.

Code-

In [ ]:
```
import pandas as pd
```

In [ ]:
```
messages = pd.read_csv('SMSSpamCollection', sep='\t',
                           names=["label", "message"])
```

In [ ]:
```
messages
```

Out[ ]:

	label	message
0	ham	Go until jurong point, crazy.. Available only ...
1	ham	Ok lar... Joking wif u oni...
2	spam	Free entry in 2 a wkly comp to win FA Cup fina...
3	ham	U dun say so early hor... U c already then say...
4	ham	Nah I don't think he goes to usf, he lives aro...
...	...	...
5567	spam	This is the 2nd time we have tried 2 contact u...
5568	ham	Will ü b going to esplanade fr home?
5569	ham	Pity, * was in mood for that. So...any other s...
5570	ham	The guy did some bitching but I acted like i'd...
5571	ham	Rofl. Its true to its name

5572 rows × 2 columns

In [ ]:
```
import re
import nltk
nltk.download('stopwords')
```

```
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
```

Out[ ]:
```
True
```

In [ ]:
```
from nltk.corpus import stopwords
from nltk.stem.porter import PorterStemmer
ps = PorterStemmer()
corpus = []
```

In [ ]:
```
for i in range(0, len(messages)):
    review = re.sub('[^a-zA-Z]', ' ', messages['message'][i])
    review = review.lower()
    review = review.split()

    review = [ps.stem(word) for word in review if not word in stopwords.words('english')]
    review = ' '.join(review)
    corpus.append(review)
```

In [ ]:
```
corpus
```

In [ ]:
```
from sklearn.feature_extraction.text import CountVectorizer
cv = CountVectorizer(max_features=5000)
X = cv.fit_transform(corpus).toarray()
```

In [ ]:
```
len(X)
```

Out[ ]:
```
5572
```

In [ ]:
```
y = pd.get_dummies(messages['label'])
y =y.iloc[:,1].values
y
```

Out[ ]:
```
array([0, 0, 1, ..., 0, 0, 0], dtype=uint8)
```

In [ ]:
```
y
print(len(y))
```

```
5572
```

In [ ]:
```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.2, random_state = 0)
```

In [ ]:
```
from sklearn.naive_bayes import MultinomialNB
spam_detect_model = MultinomialNB().fit(X_train, y_train)
```

In [ ]:
```
y_pred = spam_detect_model.predict(X_test)
```

In [ ]:
```
from sklearn.metrics import confusion_matrix
confusion_m = confusion_matrix(y_test, y_pred)
confusion_m
```

Out[ ]:
```
array([[946,   9],
       [  8, 152]])
```

In [ ]:
```
from sklearn.metrics import accuracy_score
accuracy_score(y_test, y_pred)
```

Out[ ]:
```
0.9847533632286996
```

Result- Hence NLP is implemented to solve a SMS spam classifier.