SpamText_NB_and_DT

February 2022

SpamText NB and DT

[ ]:
```
!pip install tensorflow
```

[ ]:
```
import pandas as pd
from tensorflow.keras.preprocessing.text import Tokenizer
from tensorflow.keras.preprocessing.sequence import pad_sequences
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.preprocessing import LabelEncoder

stopwords = [ "a", "about", "above", "after", "again", "against", "all", "am",␣
 ↪"an", "and", "any", "are", "as", "at", "be", "because", "been", "before",␣
 ↪"being", "below", "between", "both", "but", "by", "could", "did", "do",␣
 ↪"does", "doing", "down", "during", "each", "few", "for", "from", "further",␣
 ↪"had", "has", "have", "having", "he", "he'd", "he'll", "he's", "her", "here",␣
 ↪"here's", "hers", "herself", "him", "himself", "his", "how", "how's", "i",␣
 ↪"i'd", "i'll", "i'm", "i've", "if", "in", "into", "is", "it", "it's", "its",␣
 ↪"itself", "let's", "me", "more", "most", "my", "myself", "nor", "of", "on",␣
 ↪"once", "only", "or", "other", "ought", "our", "ours", "ourselves", "out",␣
 ↪"over", "own", "same", "she", "she'd", "she'll", "she's", "should", "so",␣
 ↪"some", "such", "than", "that", "that's", "the", "their", "theirs", "them",␣
 ↪"themselves", "then", "there", "there's", "these", "they", "they'd",␣
 ↪"they'll", "they're", "they've", "this", "those", "through", "to", "too",␣
 ↪"under", "until", "up", "very", "was", "we", "we'd", "we'll", "we're",␣
 ↪"we've", "were", "what", "what's", "when", "when's", "where", "where's",␣
 ↪"which", "while", "who", "who's", "whom", "why", "why's", "with", "would",␣
 ↪"you", "you'd", "you'll", "you're", "you've", "your", "yours", "yourself",␣
 ↪"yourselves" ]
```

[ ]:
```
datasets = pd.read_csv('spam1.csv')
print("\nData :\n",datasets)
print("\nData statistics\n",datasets.info())
```

```
Data :
        v1                                                 v2
0    spam  Free entry in 2 a wkly comp to win FA Cup fina...
1    spam  FreeMsg Hey there darling it's been 3 week's n...
```

```
2    spam  WINNER!! As a valued network customer you have...
3    spam  Had your mobile 11 months or more? U R entitle...
4    spam  SIX chances to win CASH! From 100 to 20,000 po...
..    ...                                                ...
508  spam  This is the 2nd time we have tried 2 contact u...
509   ham               Will _ b going to esplanade fr home?
510   ham  Pity, * was in mood for that. So...any other s...
511   ham  The guy did some bitching but I acted like i'd...
512   ham                         Rofl. Its true to its name

[513 rows x 2 columns]
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 513 entries, 0 to 512
Data columns (total 2 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   v1      513 non-null    object
 1   v2      513 non-null    object
dtypes: object(2)
memory usage: 8.1+ KB

Data statistics
 None
```

##Analysis

To analyze the text data, we have to turn the words into numerical numbers. We have multiple choices to accomplish this step:

1) Binary Term Frequency : count presence(1) or absence(0) for term in document

2) Bag of Words Frequency: captures the frequency of term in document

3) Term Frequency:

4) TFIDF :

In this way, if a term appears frequently in a document, it's important; if a term appears in many documents, it's not a unique identifier.

Word2Vec.

[ ]:

```

```

#Next we use CountVectorizer:

More Details and example at:

https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.CountVectorizer.html

[ ]:

```
#Import scikit-learn metrics module for accuracy calculation
from sklearn import metrics
from sklearn.metrics import precision_score
from sklearn.metrics import recall_score
```

Naive Bayes

[ ]:

```

```

Decision Tree

[ ]:

```

```

Exercise: Try this on full spam.csv file and bigram matching instead of unigram matching