```
pip install nltk
```

```
Requirement already satisfied: nltk in /usr/local/lib/python3.7/dist-packages (3.2.5)
Requirement already satisfied: six in /usr/local/lib/python3.7/dist-packages (from nltk
```

```
import nltk


nltk.download('wordnet')
```

```
[nltk_data] Downloading package wordnet to /root/nltk_data...
[nltk_data]   Unzipping corpora/wordnet.zip.
True
```

```
#loading dataset


import pandas as pd


dt = pd.read_csv("spam.csv", encoding = 'windows-1252')


dt.head(10)
```

	type	text
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...
5	spam	FreeMsg Hey there darling it's been 3 week's n...
6	ham	Even my brother is not like to speak with me. ...
7	ham	As per your request 'Melle Melle (Oru Minnamin...
8	spam	WINNER!! As a valued network customer you have...
9	spam	Had your mobile 11 months or more? U R entitle...

```
dt.fillna({'spam':-1}, inplace=True)
dt.replace({'spam':{'ham':0, 'spam':1}}, inplace=True)


dt.head()
```

	type	text
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...

```
dt['Spam'] = dt['type'].map( {'Spam' : 1 , 'ham' : 0} ).astype(int)
```

```
dt = dt.dropna()
dt['spam'] = dt['type'].map({'spam' : 1, 'ham' : 0}).astype('int64')
```

```
dt.head()
```

	type	text	spam
0	ham	Go until jurong point, crazy.. Available only ...	0
1	ham	Ok lar... Joking wif u oni...	0
2	spam	Free entry in 2 a wkly comp to win FA Cup fina...	1
3	ham	U dun say so early hor... U c already then say...	0
4	ham	Nah I don't think he goes to usf, he lives aro...	0

```
print("COLUMNS IN THE GIVEN DATA: ")
for col in dt.columns:
  print(col)
```

```
COLUMNS IN THE GIVEN DATA:
type
text
spam
```

```
t=len(dt['type'])
print("NO OF ROWS IN REVIEW COLUMN:", t)
```

```
NO OF ROWS IN REVIEW COLUMN: 116
```

```
#tokenization
```

```
dt['text'][4]
```

```
     'Nah I don't think he goes to usf, he lives around here though'

def tokenizer(text):
    return text.split()


dt['text'] = dt['text'].apply(tokenizer)


dt['text'][4]

     ['Nah',
      'I',
      "don't",
      'think',
      'he',
      'goes',
      'to',
      'usf,',
      'he',
      'lives',
      'around',
      'here',
      'though']


#stemming
dt['text'][4]

     ['Nah',
      'I',
      "don't",
      'think',
      'he',
      'goes',
      'to',
      'usf,',
      'he',
      'lives',
      'around',
      'here',
      'though']


from nltk.stem.snowball import SnowballStemmer
porter = SnowballStemmer("english", ignore_stopwords = False)


def stem_it(text):
  return [porter.stem(word) for word in text]


dt['text'] = dt['text'].apply(stem_it)


dt['text'][4]
```

```
['nah',
 'i',
 "don't",
 'think',
 'he',
 'goe',
 'to',
 'usf,',
 'he',
 'live',
 'around',
 'here',
 'though']
```

```
#Lemmitization
```

```
from nltk.stem import WordNetLemmatizer
lemmatizer = WordNetLemmatizer()
```

```
def lemmit_it(text):
  return [lemmatizer.lemmatize(word , pos = 'a') for word in text]
```

```
dt['text'] = dt['text'].apply(lemmit_it)
```

```
dt['text'][4]
```

```
['nah',
 'i',
 "don't",
 'think',
 'he',
 'goe',
 'to',
 'usf,',
 'he',
 'live',
 'around',
 'here',
 'though']
```

```
from nltk.corpus import stopwords
stop_words = stopwords.words("english")
```

```
nltk.download('stopwords')
```

```
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data]   Unzipping corpora/stopwords.zip.
True
```

```
def stop_it(text):
  review = [word for word in text if not word in stop_words]
  return review


dt['text'] = dt['text'].apply(stop_it)


dt['text'][4]
```

```
['nah', 'think', 'goe', 'usf,', 'live', 'around', 'though']
```

```
dt.head()
```

	type	text	spam
0	ham	[go, jurong, point,, crazy.., avail, onli, bug...	0
1	ham	[ok, lar..., joke, wif, u, oni...]	0
2	spam	[free, entri, 2, wkli, comp, win, fa, cup, fin...	1
3	ham	[u, dun, say, earli, hor..., u, c, alreadi, sa...	0
4	ham	[nah, think, goe, usf,, live, around, though]	0

```
dt['text'] = dt['text'].apply(' '.join)


dt.head()
```

	type	text	spam
0	ham	go jurong point, crazy.. avail onli bugi n gre...	0
1	ham	ok lar... joke wif u oni...	0
2	spam	free entri 2 wkli comp win fa cup final tkts 2...	1
3	ham	u dun say earli hor... u c alreadi say...	0
4	ham	nah think goe usf, live around though	0

```
from sklearn.feature_extraction.text import TfidfVectorizer
tfidf=TfidfVectorizer()
y=dt.spam.values
x=tfidf.fit_transform(dt['text'])


from sklearn.model_selection import train_test_split
x_train,x_text,y_train,y_text=train_test_split(x,y,random_state=1,test_size=0.2,shuffle=False
```

```
from sklearn.linear_model import LogisticRegression
clf=LogisticRegression()
clf.fit(x_train,y_train)
y_pred=clf.predict(x_text)


from sklearn.metrics import accuracy_score
acc_log = accuracy_score(y_pred,y_text)*100
print ("accuracy:",acc_log )
```

```
accuracy: 87.5
```

New Section