Попов Илья Андреевич ИУ5-23М

In [15]:
```
import spacy
import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split, cross_val_score
from sklearn.naive_bayes import MultinomialNB
from sklearn.svm import LinearSVC
from sklearn.pipeline import Pipeline
from sklearn.metrics import accuracy_score, balanced_accuracy_score
from sklearn.feature_extraction.text import CountVectorizer, TfidfVectorizer
```

In [2]:
```
raw_data = pd.read_csv('SPAM text message 20170820 - Data.csv')
```

In [3]:
```
raw_data.head()
```

Out[3]:

	Category	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...

In [4]:
```
raw_data.shape
```

Out[4]:
```
(5572, 2)
```

In [5]:
```
vocab_list = raw_data['Message'].tolist()
vocab_list[:10]
```

Out[5]:
```
['Go until jurong point, crazy.. Available only in bugis n great world la e buffet... Cine there got amore wat.
..',
 'Ok lar... Joking wif u oni...',
 "Free entry in 2 a wkly comp to win FA Cup final tkts 21st May 2005. Text FA to 87121 to receive entry questio
n(std txt rate)T&C's apply 08452810075over18's",
 'U dun say so early hor... U c already then say...',
 "Nah I don't think he goes to usf, he lives around here though",
 "FreeMsg Hey there darling it's been 3 week's now and no word back! I'd like some fun you up for it still? Tb
ok! XxX std chgs to send, £1.50 to rcv",
 'Even my brother is not like to speak with me. They treat me like aids patent.',
 "As per your request 'Melle Melle (Oru Minnaminunginte Nurungu Vettam)' has been set as your callertune for al
l Callers. Press *9 to copy your friends Callertune",
 'WINNER!! As a valued network customer you have been selected to receivea £900 prize reward! To claim call 090
61701461. Claim code KL341. Valid 12 hours only.',
 'Had your mobile 11 months or more? U R entitled to Update to the latest colour mobiles with camera for Free!
Call The Mobile Update Co FREE on 08002986030']
```

In [6]:
```
#Векторизация CountVectorizer
cv = CountVectorizer()
cv.fit(vocab_list)
cv_corpusVocab = cv.vocabulary_
# Количество признаков
len(cv_corpusVocab)
```

Out[6]:
```
8709
```

In [7]:
```
for i in list(cv_corpusVocab)[1:10]:
    print('{}={}'.format(i, cv_corpusVocab[i]))
```

```
until=8080
jurong=4370
point=5954
crazy=2334
available=1313
only=5567
in=4110
bugis=1763
great=3651
```

In [8]:

```
cv_test_features = cv.transform(vocab_list)
cv_test_features.shape
```

Out[8]:

```
(5572, 8709)
```

In [9]:

```
cv.get_feature_names()[2000:2020]
```

```
c:\users\ilya\appdata\local\programs\python\python37\lib\site-packages\sklearn\utils\deprecation.py:87: FutureW
arning: Function get_feature_names is deprecated; get_feature_names is deprecated in 1.0 and will be removed in
1.2. Please use get_feature_names_out instead.
  warnings.warn(msg, category=FutureWarning)
```

Out[9]:

```
['chef',
 'chennai',
 'cheque',
 'cherish',
 'cherthala',
 'chess',
 'chest',
 'chex',
 'cheyyamo',
 'chez',
 'chg',
 'chgs',
 'chic',
 'chick',
 'chicken',
 'chickened',
 'chief',
 'chik',
 'chikku',
 'child']
```

In [11]:

```
tfidfv = TfidfVectorizer()
tfidf_features = tfidfv.fit_transform(vocab_list)
tfidf_features.shape
```

Out[11]:

```
(5572, 8709)
```

In [17]:

```
tfidfv.get_feature_names()[2000:2020]
```

Out[17]:

```
['chef',
 'chennai',
 'cheque',
 'cherish',
 'cherthala',
 'chess',
 'chest',
 'chex',
 'cheyyamo',
 'chez',
 'chg',
 'chgs',
 'chic',
 'chick',
 'chicken',
 'chickened',
 'chief',
 'chik',
 'chikku',
 'child']
```

In [23]:

```
#Векторизация: CountVectorizer; Классификация MultinomialNB
pipeline1 = Pipeline([("vectorizer", cv), ("classifier", MultinomialNB())])
score = cross_val_score(pipeline1, raw_data['Message'], raw_data['Category'], scoring='accuracy', cv=3).mean()
print('Accuracy = {}'.format(score))
```

```
Accuracy = 0.9854630284966029
```

In [24]:

```
#Векторизация: TfidfVectorizer; Классификация MultinomialNB
```

In [24]:

```
#Векторизация: TfidfVectorizer; Классификация MultinomialNB
pipeline1 = Pipeline([("vectorizer", tfidfv), ("classifier", MultinomialNB())])
score = cross_val_score(pipeline1, raw_data['Message'], raw_data['Category'], scoring='accuracy', cv=3).mean()
print('Accuracy = {}'.format(score))
```

```
Accuracy = 0.9547742528730302
```

In [25]:

```
#Векторизация: CountVectorizer; Классификация LinearSVC
pipeline1 = Pipeline([("vectorizer", cv), ("classifier", LinearSVC())])
score = cross_val_score(pipeline1, raw_data['Message'], raw_data['Category'], scoring='accuracy', cv=3).mean()
print('Accuracy = {}'.format(score))
```

```
Accuracy = 0.9834887108563705
```

In [26]:

```
#Векторизация: TfidfVectorizer; Классификация LinearSVC
pipeline1 = Pipeline([("vectorizer", tfidfv), ("classifier", LinearSVC())])
score = cross_val_score(pipeline1, raw_data['Message'], raw_data['Category'], scoring='accuracy', cv=3).mean()
print('Accuracy = {}'.format(score))
```

```
Accuracy = 0.9847454109867356
```

Все комбинации показали очень хороший результат, с минимальной разницей в точности лучшей комбинацией стала CountVectorizer + MultinomialNB

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js