Рубежный контроль по дисциплине "Методы машинного обучения" №2.

Выполнил: Громоздов Д.Р.; группа ИУ5-23М

In [3]:
```
import pandas as pd
from sklearn.feature_extraction.text import CountVectorizer, TfidfVectorizer
from sklearn.svm import LinearSVC
from sklearn.naive_bayes import MultinomialNB
from sklearn.model_selection import cross_val_score
```

Возьмём датасет, содержащий классификацию сообщений на спам (spam) и важные сообщения (ham):

In [4]:
```
data = pd.read_csv('datasets/spam_classifier.csv', sep=",")
```

In [5]:
```
data.shape
```

Out[5]:(5572, 2)

In [6]:
```
data.head()
```

Out[6]:

	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...

Убедимся, что в используемых нами данных не будет пропусков, удалив их с помощью dropna(). Пропусков во взятом для рубежного контроля наборе данных не оказалось:

In [7]:
```
data.dropna()
```

Out[7]:

	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...
...	...	...
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

Зададим переменную, содержащую значения классов:

In [10]:
```
target = data["Category"].values
target
```

Out[10]:array(['ham', 'ham', 'spam', ..., 'ham', 'ham', 'ham'], dtype=object)

Создадим векторизацию текста сообщений на основе модели CountVectoriser:

In [9]:
```
countv = CountVectorizer()
countv_features = countv.fit_transform(data["Message"])
countv_features
```

Out[9]:<5572x8709 sparse matrix of type '<class 'numpy.int64'>'
with 74098 stored elements in Compressed Sparse Row format>

Посмотрим, какие результаты нам дают модели анализа характера (спам/не спам) текста при этом типе векторизации:

In [26]:
```
score_count_svc = cross_val_score(LinearSVC(), countv_features, target, scoring='accuracy', cv=3).mean()

print('Модель векторизации - Countvectorizer, \nМодель классификации - LinearSVC, \n3начение accuracy = {}'.format(score_count_svc))
```

Модель векторизации - Countvectorizer,
Модель классификации - LinearSVC,
3начение accuracy = 0.9834887108563705

In [28]:
```
score_count_mnb = cross_val_score(MultinomialNB(), countv_features, target, scoring='accuracy', cv=3).mean()
```

```
print('Модель векторизации - Countvectorizer, \nМодель классификации - MultinomialNB, \n3начение accuracy = {}'.format(score_count_mnb))
```

Модель векторизации - Countvectorizer,
Модель классификации - MultinomialNB,
3начение accuracy = 0.9793604779788613
Создадим векторизацию текста сообщений на основе модели TfidfVectorizer:

In [32]:
```
tfidfv = TfidfVectorizer()
tfidfv_features = tfidfv.fit_transform(data["Message"])
tfidfv_features
```

Out[32]:<5572x8709 sparse matrix of type '<class 'numpy.float64'>'
with 74098 stored elements in Compressed Sparse Row format>
Посмотрим, какие результаты нам дают модели анализа характера (спам/не спам) текста при данном типе векторизации:

In [33]:
```
score_tfidf_svc = cross_val_score(LinearSVC(), tfidfv_features, target, scoring='accuracy', cv=3).mean()

print('Модель векторизации - TfidfVectorizer, \nМодель классификации - LinearSVC, \n3начение accuracy = {}'.format(score_tfidf_svc))
```

Модель векторизации - TfidfVectorizer,
Модель классификации - LinearSVC,
3начение accuracy = 0.9806172747190152
In [34]:
```
score_tfidf_mnb = cross_val_score(MultinomialNB(), tfidfv_features, target, scoring='accuracy', cv=3).mean()

print('Модель векторизации - TfidfVectorizer, \nМодель классификации - MultinomialNB, \n3начение accuracy = {}'.format(score_tfidf_mnb))
```

Модель векторизации - TfidfVectorizer,
Модель классификации - MultinomialNB,
3начение accuracy = 0.9542356533014753
In [35]:
```
max_score = max(score_count_svc, score_count_mnb, score_tfidf_svc, score_tfidf_mnb)
print('Наилучшее значение accuracy = {}'.format(max_score))
```

Наилучшее значение accuracy = 0.9834887108563705
Лучший результат на данном наборе данных показала модель классификации LinearSVC в сочетании с векторизацией CountVectorizer.