Рубежный контроль №2 по курсу «Методы машинного обучения»

Кузьмин Роман, ИУ5-25М

Вариант задания

Группа	Классификатор №1	Классификатор №2
ИУ5-25М	SVC	LogisticRegression

Импорт библиотек

In [3]:
```
from sklearn.feature_extraction.text import CountVectorizer, TfidfVectorize
r
from sklearn.model_selection import train_test_split
from sklearn.svm import SVC
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score
import pandas as pd
import time
```

Необходимо решить задачу классификации текстов на основе любого выбранного Вами датасета (кроме примера, который рассматривался в лекции). Классификация может быть бинарной или многоклассовой. Целевой признак из выбранного Вами датасета может иметь любой физический смысл, примером является задача анализа тональности текста.

Необходимо сформировать два варианта векторизации признаков - на основе CountVectorizer и на основе TfidfVectorizer.

Для каждого метода необходимо оценить качество классификации. Сделайте вывод о том, какой вариант векторизации признаков в паре с каким классификатором показал лучшее качество.

Датасет: Spam Emails (https://www.kaggle.com/datasets/abdallahwagih/spam-emails). Содержит спам и не-спам емейлы

In [5]:
```
# Загрузка данных
df = pd.read_csv("spam.csv")

# Заменяем целевую переменную класса на числовое значение
df.Category = df.Category.apply(lambda x: 1 if x == 'spam' else 0)

df.head()
```

Out[5]:

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

In [6]:
```
df.shape
```

Out[6]:
```
(5572, 2)
```

In [7]:
```
X_train, X_test, y_train, y_test = train_test_split(df['Message'], df['Category'], test_size=0.2, random_state=42)
```

Сформировать два варианта векторизации признаков

In [8]:
```
count_vectorizer = CountVectorizer()
X_train_count = count_vectorizer.fit_transform(X_train)
```

In [9]:
```
tfidf_vectorizer = TfidfVectorizer()
X_train_tfidf = tfidf_vectorizer.fit_transform(X_train)
```

In [15]:
```
X_test_tfidf = tfidf_vectorizer.transform(X_test)
X_test_count = count_vectorizer.transform(X_test)
```

Решение задачи классификации текстов

SVC

In [18]:
```
svc_classifier_tfidf = SVC()
svc_classifier_count = SVC()
```

In [19]:
```
svc_classifier_tfidf.fit(X_train_tfidf, y_train)
svc_classifier_count.fit(X_train_count, y_train)
```

Out[19]:
```
SVC
SVC()
```

In [20]:
```
svc_accuracy = svc_classifier_tfidf.score(X_test_tfidf, y_test)
print('Точность SVC + TFIDF:', svc_accuracy)
svc_accuracy = svc_classifier_count.score(X_test_count, y_test)
print('Точность SVC + Count:', svc_accuracy)
```

```
Точность SVC + TFIDF: 0.989237668161435
Точность SVC + Count: 0.9847533632286996
```

LogisticRegression

In [21]:
```
lr_classifier_tfidf = LogisticRegression()
lr_classifier_count = LogisticRegression()
```

In [22]:
```
lr_classifier_tfidf.fit(X_train_tfidf, y_train)
lr_classifier_count.fit(X_train_count, y_train)
```

Out[22]:
```
LogisticRegression
LogisticRegression()
```

In [23]:
```
lr_accuracy = lr_classifier_tfidf.score(X_test_tfidf, y_test)
print('Точность LogReg + TFIDF:', lr_accuracy)
lr_accuracy = lr_classifier_count.score(X_test_count, y_test)
print('Точность LogReg + Count:', lr_accuracy)
```

```
Точность LogReg + TFIDF: 0.9748878923766816
Точность LogReg + Count: 0.9865470852017937
```

Вывод

Лучшие результаты классификации спама были получены с использованием метода векторизации TFIDFVectorizer и классификатором SVC. TFIDF лучше для классификации спама, так как он вылавливает важные слова в предложениях, а это легко помогает определить спам. SVC имеет нелинейную границу между классами и поэтому может хорошо разделять сложные классы.