1. **Постановка задачи**

Классификация письменных документов.

Применить 3 классификатора библиотеки scikit-learn для классификации документов, имеющихся в полученном файле.

2. **Классы документов**

№ класса	Класс	Число документов в классе
1	‘автомобиль’	249
2	‘здоровье’	157
3	‘культура’	358
4	‘наука’	227
5	‘недвижимость’	98
6	‘политика’	600
7	‘происшествие’	436
8	‘реклама’	94
9	‘семья’	101
10	‘спорт’	373
11	‘страна’	146
12	‘техника’	289
13	‘экономика’	272
Общее	**13**	**3400**

3. **Порядок предварительной обработки корпуса**

- С помощью команды `re.sub(r'[^А-я #\n]', '', data)` удаляем всё кроме русских слов, решётки и знака переноса строки
- С помощью команды `data.lower()` приводим к ниженему регистру
- Далее `data.split('\n')` делит большую строку на мелкие строки – документы, `nltk.word_tokenize(txt[i])` – токенезация,

```
morph = pymorphy2.MorphAnalyzer()
txt[i] = [morph.parse(word)[0].normal_form for word in txt[i] if word not in stop_words]
```

– отбор и лемматизация (нормализация) только тех слов, которые не являются стоп-словами.

Стоп-слова – это часто используемые слова, которые не вносят никакой дополнительной информации в текст.

Собрали стоп-слова из двух библиотек:

```
from nltk.corpus import stopwords
from stop_words import get_stop_words
stop_words = list(stopwords.words('russian'))
print(len(stop_words)) # 151
stop_words_ = list(get_stop_words('russian'))
print(len(stop_words_)) # 421
stop_words.extend(stop_words_)
```

```
print(len(stop_words)) # 572
```
Примеры стоп-слов:
[…,
'но',
'да',
'ты',
'к',
'у',
'же',
'вы',
'за',
'бы',
'по',
'только',
'ее',
'мне',
'было',
'вот',
'от',
'меня',
'еще',
'нет',
'о',
'из',
'ему',
'теперь',
'когда',
'даже',
'ну',
'вдруг',
…]

- Разделим метки и данные.

4. Размер словаря корпуса до и после предварительной обработки

Размер словаря до удаления иностранных слов, цифр и посторонних символов	Размер словаря после удаления иностранных слов, цифр и посторонних символов	Размер словаря после удаления верхних регистров и стоп-слов
84687	81286	34989

5. Примененные классификаторы и их параметры

Классификатор	Сокращённо	Время обучения, с	Параметры
KNeighborsClassifier	(KNN) - Метод K-ближайших соседей	0.044	n_neighbors=12
PassiveAggressiveClassifier		110.694	по умолчанию
LinearDiscriminantAnalysis	(LDA) - Линейный дискриминантный анализ	40.555	по умолчанию

KNeighbors – объект присваивается тому классу, который является наиболее распространённым среди k соседей данного элемента, классы которых уже известны

PassiveAggressiveClassifier – алгоритм реагирует агрессивно (меняет свои веса) на неверно классифицированные примеры и остается пассивным (не изменяет веса) в случае правильной классификации

LinearDiscriminantAnalysis – строит линейные комбинации признаков для максимизации разделения между классами данных. Алгоритм использует информацию о разбросе между классами и внутри классов для определения оптимальной гиперплоскости, разделяющей классы в признаковом пространстве

6. Точность классификации документов обучающего и проверочного множеств (общая и по классам)

KNeighbors

Train

	precision	recall	f1-score	support
автомобиль	0.9118	0.3316	0.4863	187
здоровье	0.5455	0.0508	0.0930	118
культура	0.8344	0.4888	0.6165	268
наука	0.9429	0.5824	0.7200	170
недвижимость	1.0000	0.2192	0.3596	73
политика	0.9077	0.6778	0.7761	450
происшествие	0.8982	0.6208	0.7342	327
реклама	0.8000	0.0563	0.1053	71
семья	0.4412	0.1974	0.2727	76
спорт	0.9014	0.6857	0.7789	280
страна	0.8000	0.1468	0.2481	109
техника	0.1667	0.9585	0.2840	217
экономика	0.8198	0.4461	0.5778	204
accuracy			0.5286	2550
macro avg	0.7669	0.4202	0.4656	2550
weighted avg	0.7950	0.5286	0.5717	2550

Val

	precision	recall	f1-score	support
автомобиль	0.7500	0.2903	0.4186	62
здоровье	0.0000	0.0000	0.0000	39
культура	0.8780	0.4000	0.5496	90
наука	0.9118	0.5439	0.6813	57
недвижимость	0.6667	0.0800	0.1429	25
политика	0.9519	0.6600	0.7795	150
происшествие	0.8361	0.4679	0.6000	109
реклама	0.0000	0.0000	0.0000	23

семья	0.6250	0.2000	0.3030	25
спорт	0.8929	0.5376	0.6711	93
страна	0.8333	0.1351	0.2326	37
техника	0.1468	0.9722	0.2550	72
экономика	0.6923	0.2647	0.3830	68
accuracy			0.4529	850
macro avg	0.6296	0.3501	0.3859	850
weighted avg	0.7238	0.4529	0.4978	850

LDA

Train

	precision	recall	f1-score	support
автомобиль	1.0000	0.9893	0.9946	187
здоровье	1.0000	0.9831	0.9915	118
культура	0.9158	0.9739	0.9439	268
наука	0.9011	0.9647	0.9318	170
недвижимость	1.0000	0.9178	0.9571	73
политика	1.0000	0.9933	0.9967	450
происшествие	0.9969	0.9817	0.9892	327
реклама	1.0000	0.9859	0.9929	71
семья	0.9859	0.9211	0.9524	76
спорт	0.9786	0.9821	0.9804	280
страна	0.9464	0.9725	0.9593	109
техника	0.9953	0.9724	0.9837	217
экономика	0.9950	0.9755	0.9851	204
accuracy			0.9773	2550
macro avg	0.9781	0.9702	0.9737	2550
weighted avg	0.9783	0.9773	0.9775	2550

Val

	precision	recall	f1-score	support
автомобиль	0.9375	0.4839	0.6383	62
здоровье	0.8421	0.4103	0.5517	39
культура	0.5122	0.2333	0.3206	90
наука	0.2072	0.4035	0.2738	57
недвижимость	0.0921	0.2800	0.1386	25
политика	0.6311	0.5133	0.5662	150
происшествие	0.2857	0.4404	0.3466	109
реклама	0.5000	0.0435	0.0800	23
семья	0.0833	0.2400	0.1237	25
спорт	0.7273	0.2581	0.3810	93
страна	0.2800	0.1892	0.2258	37
техника	0.7692	0.1389	0.2353	72
экономика	0.1691	0.3382	0.2255	68
accuracy			0.3447	850
macro avg	0.4644	0.3056	0.3159	850
weighted avg	0.5123	0.3447	0.3679	850

PassiveAggressive

Train

	precision	recall	f1-score	support
автомобиль	1.0000	1.0000	1.0000	187
здоровье	1.0000	1.0000	1.0000	118
культура	1.0000	1.0000	1.0000	268
наука	1.0000	1.0000	1.0000	170

недвижимость	1.0000	1.0000	1.0000	73
политика	0.9978	1.0000	0.9989	450
происшествие	1.0000	1.0000	1.0000	327
реклама	1.0000	1.0000	1.0000	71
семья	1.0000	1.0000	1.0000	76
спорт	1.0000	1.0000	1.0000	280
страна	1.0000	0.9908	0.9954	109
техника	1.0000	1.0000	1.0000	217
экономика	1.0000	1.0000	1.0000	204
accuracy			0.9996	2550
macro avg	0.9998	0.9993	0.9996	2550
weighted avg	0.9996	0.9996	0.9996	2550

Val

	precision	recall	f1-score	support
автомобиль	0.8871	0.8871	0.8871	62
здоровье	0.7674	0.8462	0.8049	39
культура	0.8182	0.9000	0.8571	90
наука	0.9153	0.9474	0.9310	57
недвижимость	0.6071	0.6800	0.6415	25
политика	0.9178	0.8933	0.9054	150
происшествие	0.9266	0.9266	0.9266	109
реклама	0.9333	0.6087	0.7368	23
семья	0.7692	0.8000	0.7843	25
спорт	0.9684	0.9892	0.9787	93
страна	0.4286	0.4054	0.4167	37
техника	0.8841	0.8472	0.8652	72
экономика	0.8594	0.8088	0.8333	68
accuracy			0.8612	850
macro avg	0.8217	0.8108	0.8130	850
weighted avg	0.8627	0.8612	0.8605	850

7. Вывод

Классификатор KNN оказался наиболее быстр, но наименее точен.

Классификатор LDA обучался дольше, при этом случилось переобучение.

Классификтор PA оказался наиболее медленным, при этом на тесте показал высокую точность.