АНТИПЛАГИАТ
ОБНАРУЖЕНИЕ ЗАИМСТВОВАНИЙ

Отчет о проверке на заимствования №1

Автор: Содиков Фарход Фирдавсович
Проверяющий: Пользователь для API (galkinala@fa.ru / ID: 2588)
Организация: Финансовый университет при Правительстве РФ

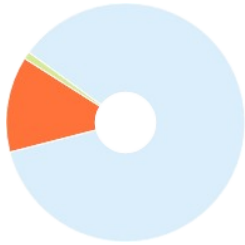
Отчет предоставлен сервисом «Антиплагиат» - http://fa.antiplagiat.ru

ИНФОРМАЦИЯ О ДОКУМЕНТЕ

№ документа: 1134668
Начало загрузки: 28.04.2022 01:07:33
Длительность загрузки: 00:00:07
Имя исходного файла: ПИ19-1_Содиков_Фарход_Пояснительная_записка_с_ссылкой_на_исходный_код.docx
Название документа: ПИ19-1_Содиков_Фарход_Пояснительная_записка_с_ссылкой_на_исходный_код.docx
Размер текста: 1 кБ
Символов в тексте: 20776
Слов в тексте: 2379
Число предложений: 173

ИНФОРМАЦИЯ ОБ ОТЧЕТЕ

Последний готовый отчет (ред.)
Начало проверки: 28.04.2022 01:07:40
Длительность проверки: 00:00:12
Комментарии: не указано
Поиск с учетом редактирования: да
Модули поиска: ИПС Адилет, Библиография, Сводная коллекция ЭБС, Интернет Плюс, Сводная коллекция РГБ, Цитирование, Переводные заимствования (RuEn), Переводные заимствования по eLIBRARY.RU (EnRu), Переводные заимствования по Интернету (EnRu), Переводные заимствования издательства Wiley (RuEn), eLIBRARY.RU, Модуль поиска "ФУ", СПС ГАРАНТ, Медицина, Диссертации НББ, Перефразирования по eLIBRARY.RU, Перефразирования по Интернету, Перефразирования по коллекции издательства Wiley, Патенты СССР, РФ, СНГ, СМИ России и СНГ, Шаблонные фразы, Кольцо вузов, Издательство Wiley, Переводные заимствования



ЗАИМСТВОВАНИЯ	САМОЦИТИРОВАНИЯ	ЦИТИРОВАНИЯ	ОРИГИНАЛЬНОСТЬ
12,83%	0%	1,13%	86,04%

Заимствования — доля всех найденных текстовых пересечений, за исключением тех, которые система отнесла к цитированиям, по отношению к общему объему документа.
Самоцитирования — доля фрагментов текста проверяемого документа, совпадающий или почти совпадающий с фрагментом текста источника, автором или соавтором которого является автор проверяемого документа, по отношению к общему объему документа.
Цитирования — доля текстовых пересечений, которые не являются авторскими, но система посчитала их использование корректным, по отношению к общему объему документа. Сюда относятся оформленные по ГОСТу цитаты; общеупотребительные выражения; фрагменты текста, найденные в источниках из коллекций нормативно-правовой документации.
Текстовое пересечение — фрагмент текста проверяемого документа, совпадающий или почти совпадающий с фрагментом текста источника.
Источник — документ, проиндексированный в системе и содержащийся в модуле поиска, по которому проводится проверка.
Оригинальность — доля фрагментов текста проверяемого документа, не обнаруженных ни в одном источнике, по которым шла проверка, по отношению к общему объему документа.
Заимствования, самоцитирования, цитирования и оригинальность являются отдельными показателями и в сумме дают 100%, что соответствует всему тексту проверяемого документа.
Обращаем Ваше внимание, что система находит текстовые пересечения проверяемого документа с проиндексированными в системе текстовыми источниками. При этом система является вспомогательным инструментом, определение корректности и правомерности заимствований или цитирований, а также авторства текстовых фрагментов проверяемого документа остается в компетенции проверяющего.

№	Доля в отчете	Доля в тексте	Источник	Актуален на	Модуль поиска	Блоков в отчете	Блоков в тексте
[01]	2,36%	3,35%	Агасандян А.М..pdf	15 Дек 2020	Модуль поиска "ФУ"	2	3
[02]	2,96%	2,96%	Lebedenko_diplom	10 Июн 2021	Кольцо вузов	3	3
[03]	0%	2,6%	Панкратова_ЕС_ВКР.docx	24 Мая 2020	Кольцо вузов	0	3
[04]	0%	2,56%	Валитова ПМ18-4 «Машинное обучение в задачах распознавания текста».pdf	17 Дек 2020	Модуль поиска "ФУ"	0	3
[05]	0%	2,28%	Шмыглёв_181462_ПМ18-4_Машинное_обучение_в_задачах_визуализации_информации_Пояснительная_записка.pdf	13 Дек 2020	Модуль поиска "ФУ"	0	1
[06]	0%	2,24%	Записка_к_курсовой_работе.pdf	15 Дек 2020	Модуль поиска "ФУ"	0	1
[07]	0,53%	2,13%	Бадараева.pdf	02 Дек 2020	Модуль поиска "ФУ"	1	2
[08]	2,13%	2,13%	18062021108	18 Июн 2021	Кольцо вузов	1	1
[09]	0%	2,06%	ВКР_Панкратова_ЕС_3.pdf	03 Июн 2020	Кольцо вузов	0	2
[10]	0%	1,97%	Машинное обучение в задачах извлечения знаний из аудиоданных.pdf	04 Дек 2020	Модуль поиска "ФУ"	0	1
[11]	0,99%	1,91%	ТЕСТИРОВАНИЕ МЕТОДОВ МАШИННОГО ОБУЧЕНИЯ В ЗАДАЧЕ КЛАССИФИКАЦИИ HTTP ЗАПРОСОВ С ПРИМЕНЕНИЕ ТЕХНОЛОГИИ TF-IDF. http://elibrary.ru	11 Фев 2020	Перефразирования по eLIBRARY.RU	1	2
[12]	0%	1,85%	Курсовая работа.docx	16 Дек 2020	Модуль поиска "ФУ"	0	1
[13]	0%	1,74%	kursovaya_Repin.docx	08 Дек 2020	Модуль поиска "ФУ"	0	1
[14]	0%	1,74%	Курсовая_работа_Кузьменко.docx	12 Дек 2020	Модуль поиска "ФУ"	0	1

[15]	1,61%	1,61%	Отчет_Алимбаев_Надиржан.pdf	11 Июн 2020	Кольцо вузов	1	1
[16]	0%	1,6%	Курсовая Финал.pdf	22 Дек 2020	Модуль поиска "ФУ"	0	1
[17]	0,79%	1,19%	dvoryashin_d_a_analiz-i-prognozirovanie-zaemshchika-rossiyskogo-roznichnogo-banka.docx	20 Мая 2019	Кольцо вузов	1	2
[18]	0,11%	1,15%	38703-Машинное обучение на языке Python с использованием SciKitLearn	09 Июл 2020	Кольцо вузов	1	2
[19]	1,13%	1,13%	не указано	13 Янв 2022	Шаблонные фразы	4	4
[20]	0%	0,96%	http://www.fa.ru/fil/penza/org/chair/mion/education/Documents/Uchpos_Analyzdann1_bkl_17.pdf http://fa.ru	17 Апр 2022	Интернет Плюс	0	1
[21]	0%	0,89%	Вопросы теории и практики налогообложения: сборник научных статей https://e.lanbook.com	22 Янв 2020	Сводная коллекция ЭБС	0	1
[22]	0%	0,83%	Проблемы экономической безопасности России в условиях геополитического кризиса и санкционного давления западных стран https://e.lanbook.com	22 Янв 2020	Сводная коллекция ЭБС	0	1
[23]	0%	0,83%	Совершенствование технологий и инструментов в развитии бизнеса: сборник научно-исследовательских работ https://e.lanbook.com	22 Янв 2020	Сводная коллекция ЭБС	0	1
[24]	0%	0,83%	Выписка из протокола | Социальная сеть Pandia.ru http://pandia.ru	30 Сен 2021	Интернет Плюс	0	1
[25]	0%	0,83%	Образец формы отзыва на выпускную квалификационную (бакалаврскую) работу http://mydocx.ru	раньше 2011	Интернет Плюс	0	1
[26]	0,77%	0,77%	dissertatsia	26 Июн 2020	Кольцо вузов	1	1
[27]	0%	0,76%	Реализация SVM и ядра SVM с помощью Python Scikit-Learn - pythobyte.com https://pythobyte.com	17 Мая 2021	Интернет Плюс	0	1
[28]	0%	0,76%	ВКР_Гнатюк_И.pdf	10 Июн 2020	Кольцо вузов	0	1
[29]	0%	0,71%	Вопросы теории и практики налогообложения: сборник научных статей https://e.lanbook.com	22 Янв 2020	Сводная коллекция ЭБС	0	1
[30]	0%	0,71%	Финансовый университет http://fa.ru	04 Янв 2016	Интернет Плюс	0	1
[31]	0%	0,71%	Постановление Правительства РФ от 14.07.2010 N 510 "О федеральном государственном образовательном бюджетном учреждении высшего образования "Финансовый университет при Правительстве Российской Федерации" (с изменениями и дополнениями) (1/3) http://base.garant.ru	раньше 2011	Интернет Плюс	0	1
[32]	0%	0,71%	Сборник официальных документов и материалов 4/2016 http://studentlibrary.ru	20 Дек 2016	Медицина	0	1
[33]	0%	0,71%	Управление прибылью в акционерных обществах региона: теория и практика. Книга 1 http://studentlibrary.ru	19 Дек 2016	Медицина	0	1
[34]	0%	0,71%	Ресурс: Новое прочтение и геоэкономическое измерение экспортного потенциала http://studentlibrary.ru	19 Дек 2016	Медицина	0	1
[35]	0%	0,38%	Скачать этот файл PDF (1/2) https://vestnik.susu.ru	20 Мар 2020	Интернет Плюс	0	1
[36]	0%	0,38%	Analysis of monthly and daily profiles of DHW use in apartment blocks in Norway https://e3s-conferences.org	27 Апр 2022	Интернет Плюс	0	1
[37]	0%	0,38%	Download this PDF file https://vestnik.susu.ru	16 Ноя 2021	Интернет Плюс	0	1
[38]	0%	0,38%	Analysis of monthly and daily profiles of DHW use in apartment blocks in Norway https://e3s-conferences.org	27 Апр 2022	Интернет Плюс	0	1
[39]	0,33%	0,33%	Научный электронный журнал "Меридиан" http://meridian-journal.ru	05 Мая 2020	Интернет Плюс	1	1
[40]	0%	0,29%	Dropout in Neural Networks Simulates the Paradoxical Effects of Deep Brain Stimulation on Memory https://frontiersin.org	27 Апр 2022	Интернет Плюс	0	1
[41]	0%	0,29%	http://www.hjphd.iit.uni-miskolc.hu/SH/courses.pdf http://hjphd.iit.uni-miskolc.hu	10 Мар 2022	Интернет Плюс	0	1
[42]	0%	0,29%	The EMNIST Dataset | NIST https://nist.gov	16 Апр 2022	Интернет Плюс	0	1
[43]	0%	0,29%	не указано	13 Янв 2022	Цитирование	0	1
[44]	0%	0,26%	30 книг о Big Data и анализе данных |DataSides http://ru.datasides.com	30 Апр 2020	Интернет Плюс	0	1
[45]	0%	0,26%	GitHub - dwelcaslu/machine-learning-books https://github.com	29 Мая 2021	Интернет Плюс	0	1
[46]	0%	0,26%	5 книг, с которыми анализ данных и Scala до безобразия просты - Как научиться программировать? https://webstudio-uwk.ru	10 Дек 2020	Интернет Плюс	0	1

[47]	0,02%	0,25%	Оценка моделей прогнозирования на основе данных о спросе | Статья в журнале «Молодой ученый» https://moluch.ru	20 Окт 2020	Интернет Плюс	1	1
[48]	0%	0,25%	Оценка моделей прогнозирования на основе данных о спросе | Статья в журнале «Молодой ученый» https://moluch.ru	13 Окт 2020	Интернет Плюс	0	1
[49]	0,21%	0,21%	Диссертация https://hse.ru	30 Дек 2017	Интернет Плюс	1	1

Федеральное государственное образовательное бюджетное учреждение высшего образования
«Финансовый университет при Правительстве Российской Федерации»
(Финансовый университет)

Департамент анализа данных и машинного обучения

Пояснительная записка к курсовой работе по дисциплине «Технологии анализа данных и машинного обучения» на тему:

«Машинное обучение в задачах классификации текстов»

Выполнил:
Студент группы ПИ19-1
Содиков Фарход Фирдавсович

Научный руководитель:
Доцент, Кандидат педагогических наук
Никитин Пётр Владимирович

Москва
2022

Содержание

ВВЕДЕНИЕ

Основная идея курсового проекта, в целом машинного обучения, состоит в том, чтобы машина (компьютерная) не просто использовала заранее написанный алгоритмический код, а сама обучалась решению поставленной задачи. Главной и единственной средой разработки в создании машинного обучения является Jupyter Notebook, с помощью которого сразу можно видеть результат выполнения кода или одного из его фрагментов. В данном проекте единственным языком программирования является Python, потому что он является одним из тех языков, который легко и просто решает проблемы, связанные с машинным обучением и анализом больших данных. Также этот язык имеет множество различных библиотек, некоторые из которых используется в поставленной задаче.

<u>В разработку машинного обучения входит:</u>

- Классификация данных
- Кластеризация данных
- Выявление аномалий (в нашей задаче не является обязательным пунктом)
- Снижение размерности (в нашей задаче не является обязательным пунктом)
- Предсказание событий (регрессия)

В наше время искусственный интеллект значительно развивается и всё больше IT-компаний нуждаются в создании машин, которые обучают самих себя для автоматизации работ не только в IT-сфере, но и в других отраслях.

К концу работы должен быть готовый машинный код, который проанализирует небольшой набор данных (датасет) и обучит себя эмпирическим методом с использованием различных библиотек.

Постановка задачи

Перед стартом реализации нашей программной части работы нужно понять, какая задача стоит перед нами.

Три этапа реализации:

1. Получение разновидных текстов из различных источников. Можно брать готовые датасеты.
2. Загрузка всех собранных данных или датасета.
3. Получение результата с максимально высшей точностью.

Давайте определимся с такими терминами как: машинное обучение, глубокое обучение, нейронные сети и искусственный интеллект. А как они связаны между собой? Как они отличаются друг от друга?

И машинное обучение, и глубокое обучение, и нейронные сети – все это области искусственного интеллекта. Однако глубокое обучение на самом деле является частью машинного обучения, а нейронная сеть – частью глубокого обучения.

Для понимания и освоения вышеперечисленных терминов давайте дадим им определения простыми словами:

- *Нейронные сети (искусственные)* – это математические модели. Их работоспособность схожа с нервными клетками животных. Нейронные сети лежат в основе алгоритмов глубокого обучения и являются подмножеством машинного обучения. В состав искусственных нейронных сетей входит входной слой, выходной слой и парочку скрытых слоёв. Искусственные нейроны, или узлы, связаны друг с другом и имеют соответствующие веса и пороговые значения. Активация узла происходит, если выходные данные любого отдельного узла превышают указанное пороговое значение. Затем отправляются данные на следующий уровень сети. Иначе данные не отправляются на следующий уровень сети.

- *Машинное обучение* — это сфера информатики и искусственного интеллекта (ИИ), которая фокусируется на использовании данных и алгоритмов для имитации способа обучения людей, постепенно повышая его точность.

После того как мы поняли основную идею курсового проекта, этапы реализации поставленной задачи и основные термины мы можем перейти к описанию датасета.

Реализация

Описание датасета

В нашей задаче мы будем использовать готовый датасет «The 20 newsgroups text dataset», в котором, как видно по названию, содержится 20 наборов данных, которые в свою очередь содержат более 18000 новостных постов по 20 различным темам. Данные разделены на два подмножества: одна для обучения, а другая для тестирования или для оценки производительности.

Рисунок с кодом загрузки датасета:

Загрузка датасета

```
# Обучающая выборка
newsgroups_train = fetch_20newsgroups(subset='train', remove=('headers', 'footers', 'quotes'))
# Тестовая выборка
newsgroups_test = fetch_20newsgroups(subset='test', remove=('headers', 'footers', 'quotes'))
```

Среда разработки не может показать нам все наши новости и предупреждает о превышении скорости ввода данных, так как датасет имеет большой объём данных. Поэтому сервер среды временно прекращает отправку выходных данных.

Рисунок с предупреждением о большом объёме данных в датасете:

Исследование данных

```
print(newsgroups_train)
```

```
IOPub data rate exceeded.
The notebook server will temporarily stop sending output
to the client in order to avoid crashing it.
To change this limit, set the config variable
`--NotebookApp.iopub_data_rate_limit`.

Current values:
NotebookApp.iopub_data_rate_limit=1000000.0 (bytes/sec)
NotebookApp.rate_limit_window=3.0 (secs)
```

```
print(newsgroups_test)
```

```
IOPub data rate exceeded.
The notebook server will temporarily stop sending output
to the client in order to avoid crashing it.
To change this limit, set the config variable
`--NotebookApp.iopub_data_rate_limit`.

Current values:
NotebookApp.iopub_data_rate_limit=1000000.0 (bytes/sec)
NotebookApp.rate_limit_window=3.0 (secs)
```

Чтобы увидеть хоть какое-то описание воспользуемся функцией .DESCR(), которая выводит базовую описательную статистику, включая метки переменных.

Рисунок с описательной статистики:

```
print(newsgroups_train.DESCR)
```

```
.. _20newsgroups_dataset:

The 20 newsgroups text dataset
------------------------------

The 20 newsgroups dataset comprises around 18000 newsgroups posts on
20 topics split in two subsets: one for training (or development)
and the other one for testing (or for performance evaluation). The split
between the train and test set is based upon a messages posted before
and after a specific date.

This module contains two loaders. The first one,
:func:`sklearn.datasets.fetch_20newsgroups`,
returns a list of the raw texts that can be fed to text feature
extractors such as :class:`~sklearn.feature_extraction.text.CountVectorizer`
with custom parameters so as to extract feature vectors.
The second one, :func:`sklearn.datasets.fetch_20newsgroups_vectorized`,
returns ready-to-use features, i.e., it is not necessary to use a feature
extractor.
```

```
print(newsgroups_test.DESCR)
```

```
.. _20newsgroups_dataset:

The 20 newsgroups text dataset
------------------------------

The 20 newsgroups dataset comprises around 18000 newsgroups posts on
20 topics split in two subsets: one for training (or development)
and the other one for testing (or for performance evaluation). The split
between the train and test set is based upon a messages posted before
and after a specific date.

This module contains two loaders. The first one,
:func:`sklearn.datasets.fetch_20newsgroups`,
returns a list of the raw texts that can be fed to text feature
extractors such as :class:`~sklearn.feature_extraction.text.CountVectorizer`
with custom parameters so as to extract feature vectors.
The second one, :func:`sklearn.datasets.fetch_20newsgroups_vectorized`,
returns ready-to-use features, i.e., it is not necessary to use a feature
extractor.
```

Также можно посмотреть один из примеров постов. Например, возьмем случайную группу под нулевым индексом и пост под нулевым индексом, то есть первую в группе. Рисунок с примером поста:

```
print(f'''
Группа: {newsgroups_train.target[0]}
Данные: {newsgroups_train.data[0]}
''')
```

```
Группа: 7
Данные: I was wondering if anyone out there could enlighten me on this car I saw
the other day. It was a 2-door sports car, looked to be from the late 60s/
early 70s. It was called a Bricklin. The doors were really small. In addition,
the front bumper was separate from the rest of the body. This is
all I know. If anyone can tellme a model name, engine specs, years
of production, where this car is made, history, or whatever info you
have on this funky looking car, please e-mail.
```

Рисунок с ещё одним примером поста:

```
print(f'''
Группа: {newsgroups_train.target[2]}
Данные: {newsgroups_train.data[2]}
''')
```

```
Группа: 4
Данные: well folks, my mac plus finally gave up the ghost this weekend after
starting life as a 512k way back in 1985.  sooo, i'm in the market for a
new machine a bit sooner than i intended to be...

i'm looking into picking up a powerbook 160 or maybe 180 and have a bunch
of questions that (hopefully) somebody can answer:

* does anybody know any dirt on when the next round of powerbook
introductions are expected?  i'd heard the 185c was supposed to make an
appearence "this summer" but haven't heard anymore on it - and since i
don't have access to macleak, i was wondering if anybody out there had
more info...

* has anybody heard rumors about price drops to the powerbook line like the
ones the duo's just went through recently?

* what's the impression of the display on the 180?  i could probably swing
a 180 if i got the 80Mb disk rather than the 120, but i don't really have
a feel for how much "better" the display is (yea, it looks great in the
store, but is that all "wow" or is it really that good?).  could i solicit
some opinions of people who use the 160 and 180 day-to-day on if its worth
taking the disk size and money hit to get the active display?  (i realize
this is a real subjective question, but i've only played around with the
machines in a computer store breifly and figured the opinions of somebody
who actually uses the machine daily might prove helpful).

* how well does hellcats perform?  ;)

thanks a bunch in advance for any info - if you could email, i'll post a
summary (news reading time is at a premium with finals just around the
corner... :( )
--
Tom Willis  \  twillis@ecn.purdue.edu    \    Purdue Electrical Engineering
```

Проведём описательный анализ датасета:

```
arr = newsgroups_train.target
size = len(arr)
mean = np.mean(arr)
median = np.median(arr)
min = np.amin(arr)
max = np.amax(arr)
rng = np.ptp(arr)
var = np.var(arr)
std = np.std(arr)

print(f'''
Описательный анализ. Выборка train target.
Массив: {arr}
Размер: {size}
Минимум: {min}
Максимум: {max}
Средняя: {mean}
Медиана: {median}
Диапазон: {rng}
Дисперсия: {var}
Стандартное отклонение = {std}
''')

counter = Counter(arr)
groups, counts = list(counter.keys()), list(counter.values())
indexes = range(len(groups))
plt.bar(indexes, counts, align="center")
plt.xticks(indexes, groups)
plt.title("Гистограмма распределения частот. Выборка train target")
plt.ylabel("Группа train target")
plt.xlabel("Кол-во в выборке")
plt.show()
```

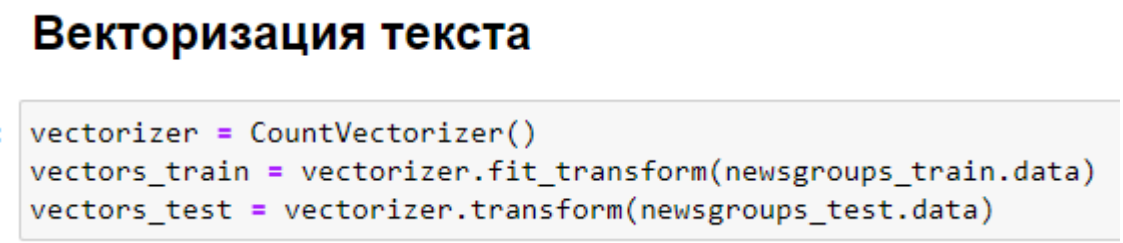
```
Описательный анализ. Выборка train target.
Массив: [7 4 4 ... 3 1 8]
Размер: 11314
Минимум: 0
Максимум: 19
Средняя: 9.29299982322786
Медиана: 9.0
Диапазон: 19
Дисперсия: 30.941108855046814
Стандартное отклонение = 5.562473267805142
```

Как можно заметить размер выборки равен 11314. Остальные из 18000 данных находятся в подмножестве newsgroup_test, то есть в тестовом. Проверим это изменив первую строчку на «arr=newsgroups_test.target»:

```
Описательный анализ. Выборка train target.
Массив: [ 7  5  0 ...  9  6 15]
Размер: 7532
Минимум: 0
Максимум: 19
Средняя: 9.293414763674987
Медиана: 9.0
Диапазон: 19
Дисперсия: 30.94603735691427
Стандартное отклонение = 5.56291626369787
```

Векторизация текста

Для векторизации текста используем функцию CountVectorizer:



CountVectorizer нужен нам для того, чтоб машина могла понимать символы и слова. Но так как мы имеем дело с текстовыми данными, нам нужно предоставлять их в цифрах, чтоб они были понятны машине. С помощью этого метода мы преобразуем текст в числовые данные.

Построение моделей

Для начала создадим общую функцию для создания моделей. Входной данной является классификатор. Функция будет выводить:

1. Размер выборки
2. Время обучения
3. Accuracy
4. Precision
5. Recall
6. F1 метрика
7. График кривой обучения
8. Матрица ошибок

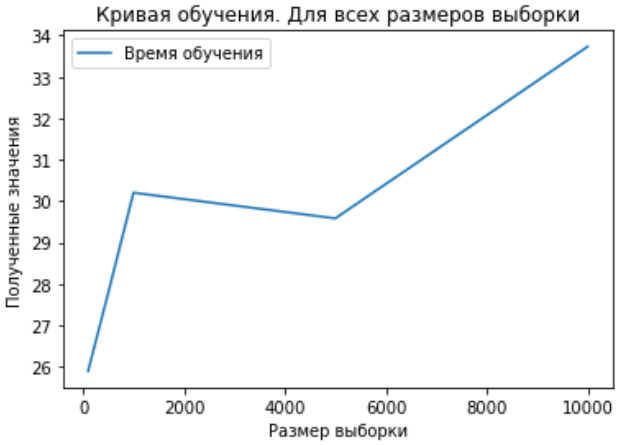
Рисунок с функцией process():

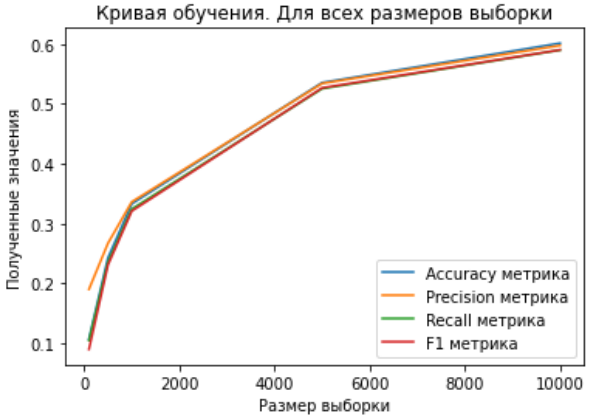
```python
def process(cls):
  sizes = [100, 500, 1000, 5000, 10000]
  Ts, As, Ps, Rs, Fs = [], [], [], [], []
  for S in sizes:
    start = time.time()
    cls.fit(vectors_train[:S], newsgroups_train.target[:S])
    end = time.time()
    T = round(end - start, 2)
    pred = cls.predict(vectors_test)
    A = metrics.accuracy_score(newsgroups_test.target, pred)
    P = metrics.precision_score(newsgroups_test.target, pred, average="macro")
    R = metrics.recall_score(newsgroups_test.target, pred, average="macro")
    F =  metrics.f1_score(newsgroups_test.target, pred, average="macro")
    print("Размер выборки:", S, "элементов")
    print("Время обучения:", T, "секунд")
    print("Accuracy метрика:", A)
    print("Precision метрика:", P)
    print("Recall метрика:", R)
    print("F1 метрика:", F)
    print()
    Ts.append(T)
    As.append(A)
    Ps.append(P)
    Rs.append(R)
    Fs.append(F)

  plt.plot(sizes, Ts, label="Время обучения")
  plt.title("Кривая обучения. Для всех размеров выборки")
  plt.ylabel("Полученные значения")
  plt.xlabel("Размер выборки")
  plt.legend()
  plt.show()

  plt.plot(sizes, As, label="Accuracy метрика")
  plt.plot(sizes, Ps, label="Precision метрика")
  plt.plot(sizes, Rs, label="Recall метрика")
  plt.plot(sizes, Fs, label="F1 метрика")
  plt.title("Кривая обучения. Для всех размеров выборки")
  plt.ylabel("Полученные значения")
  plt.xlabel("Размер выборки")
  plt.legend()
  plt.show()

  sns.heatmap(metrics.confusion_matrix(newsgroups_test.target, pred), annot=True)
  plt.title("Матрица ошибок. Размер выборки 10000")
  plt.ylabel("Действительные значения")
  plt.xlabel("Предсказанные значения")
  plt.show()
```

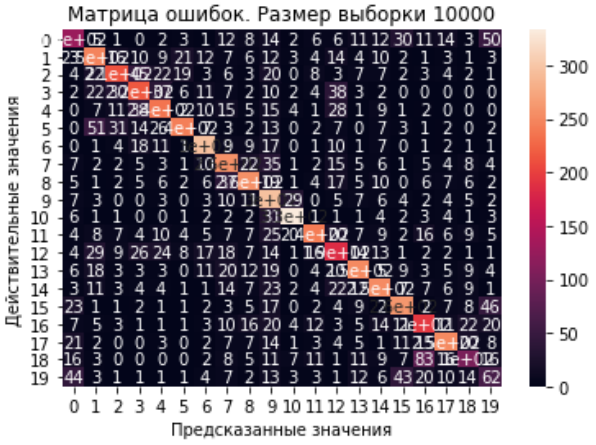
Логистическая регрессия (LogisticRegression)

```
process(cls = LogisticRegression())
```

```
Размер выборки: 100 элементов
Время обучения: 25.89 секунд
Accuracy метрика: 0.10621348911311737
Precision метрика: 0.18972990026410932
Recall метрика: 0.10410648140804454
F1 метрика: 0.08924794199159744

Размер выборки: 500 элементов
Время обучения: 27.78 секунд
Accuracy метрика: 0.242299522039299
Precision метрика: 0.26685459640207415
Recall метрика: 0.23552692330562802
F1 метрика: 0.2306386517402231

Размер выборки: 1000 элементов
Время обучения: 30.2 секунд
Accuracy метрика: 0.3328465215082315
Precision метрика: 0.3360787021396109
Recall метрика: 0.32403039192935335
F1 метрика: 0.32071765336379116

Размер выборки: 5000 элементов
Время обучения: 29.58 секунд
Accuracy метрика: 0.5357142857142857
Precision метрика: 0.5343628467180175
Recall метрика: 0.5256185383435513
F1 метрика: 0.5269034230819498

Размер выборки: 10000 элементов
Время обучения: 33.73 секунд
Accuracy метрика: 0.6016994158258099
Precision метрика: 0.5978868529081882
Recall метрика: 0.5902299469146306
F1 метрика: 0.5905503224940134
```



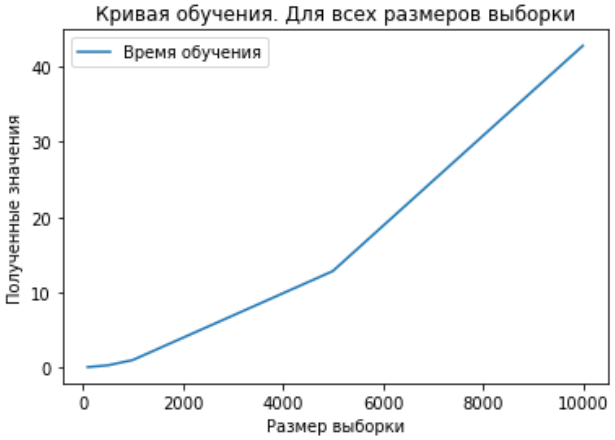


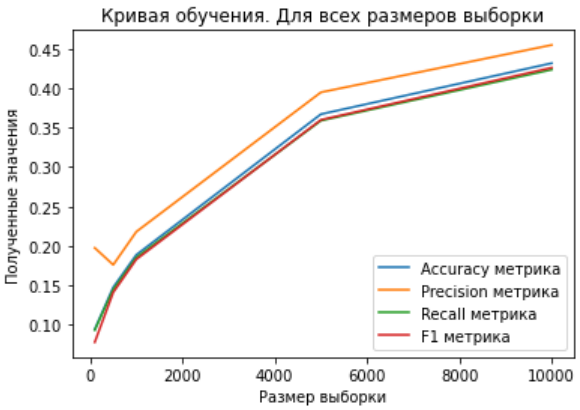


Метод опорных векторов с линейным ядром (SVC)

```
process(cls = SVC(kernel='linear'))
```

```
Размер выборки: 100 элементов
Время обучения: 0.02 секунд
Accuracy метрика: 0.09386617100371747
Precision метрика: 0.19734917375443467
Recall метрика: 0.09297358178917556
F1 метрика: 0.07751912921875245

Размер выборки: 500 элементов
Время обучения: 0.25 секунд
Accuracy метрика: 0.14723844928305896
Precision метрика: 0.17581710166787995
Recall метрика: 0.14453730707821638
F1 метрика: 0.14081039083262473

Размер выборки: 1000 элементов
Время обучения: 0.95 секунд
Accuracy метрика: 0.18852894317578334
Precision метрика: 0.21788762478954665
Recall метрика: 0.18520747404429144
F1 метрика: 0.182795406618042

Размер выборки: 5000 элементов
Время обучения: 12.79 секунд
Accuracy метрика: 0.3672331386086033
Precision метрика: 0.3950283299152616
Recall метрика: 0.3590524242101238
F1 метрика: 0.3600209218170143

Размер выборки: 10000 элементов
Время обучения: 42.8 секунд
Accuracy метрика: 0.4320233669676049
Precision метрика: 0.4551721262920491
Recall метрика: 0.42381899989150373
F1 метрика: 0.4262801419670943
```

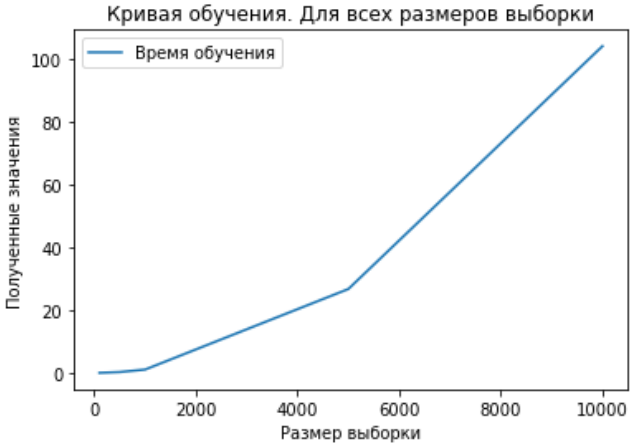


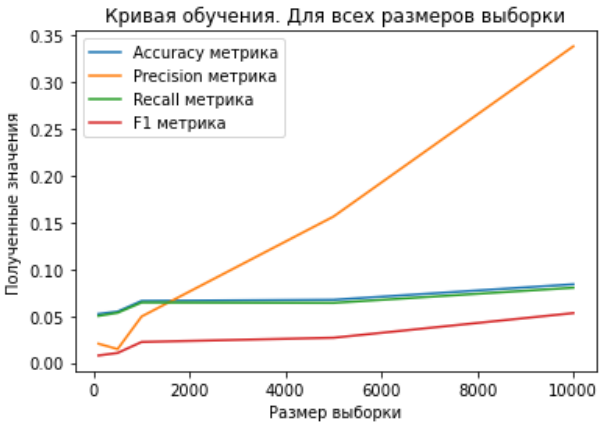


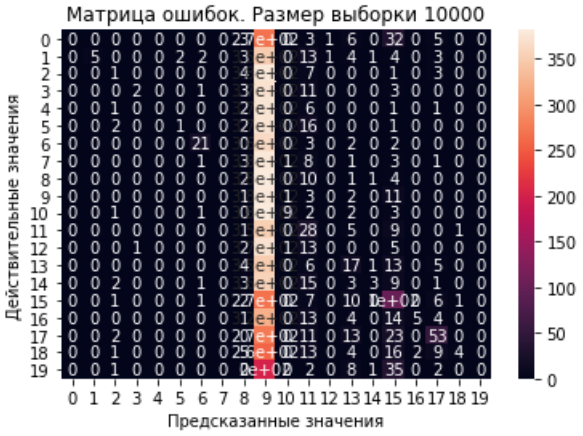
Метод опорных векторов с гауссовым ядром (SVC)

```
process(cls = SVC(kernel='rbf'))
```

```
Размер выборки: 100 элементов
Время обучения: 0.02 секунд
Accuracy метрика: 0.052177376526818905
Precision метрика: 0.020288809061534872
Recall метрика: 0.050062451638363946
F1 метрика: 0.007680664705923969

Размер выборки: 500 элементов
Время обучения: 0.27 секунд
Accuracy метрика: 0.05469994689325544
Precision метрика: 0.01452285970884939
Recall метрика: 0.053384751233257734
F1 метрика: 0.010383854101683533

Размер выборки: 1000 элементов
Время обучения: 1.06 секунд
Accuracy метрика: 0.06611789697291556
Precision метрика: 0.04942526960272105
Recall метрика: 0.06437875308834694
F1 метрика: 0.022152855810613666

Размер выборки: 5000 элементов
Время обучения: 26.72 секунд
Accuracy метрика: 0.06731279872543813
Precision метрика: 0.15640476305872522
Recall метрика: 0.06397245848426322
F1 метрика: 0.02666421595640337

Размер выборки: 10000 элементов
Время обучения: 104.07 секунд
Accuracy метрика: 0.08390865639936272
Precision метрика: 0.3386439605034669
Recall метрика: 0.0802120884091797
F1 метрика: 0.0530461432106776
```



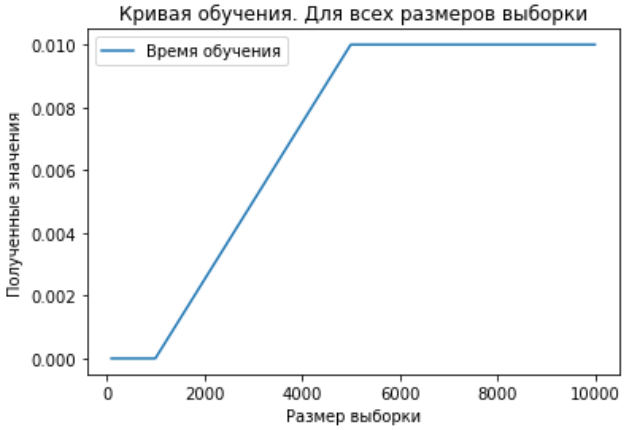


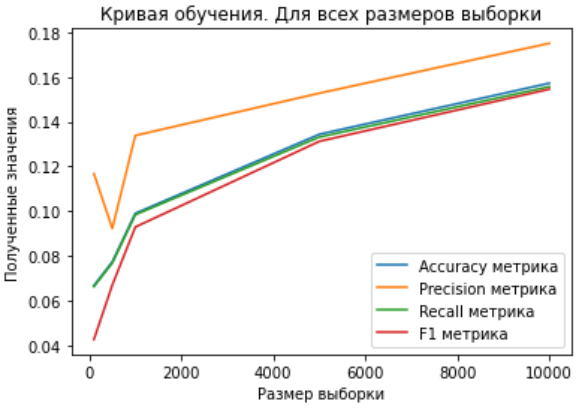


Метод k ближайших соседей (KNeighborsClassifier)

```
process(cls = KNeighborsClassifier())
```

```
Размер выборки: 100 элементов
Время обучения: 0.0 секунд
Accuracy метрика: 0.06678173127987254
Precision метрика: 0.11674743739962463
Recall метрика: 0.066358603369721
F1 метрика: 0.04271346995551904

Размер выборки: 500 элементов
Время обучения: 0.0 секунд
Accuracy метрика: 0.07740308019118428
Precision метрика: 0.09228915641464898
Recall метрика: 0.07691974217276061
F1 метрика: 0.06706284883419339

Размер выборки: 1000 элементов
Время обучения: 0.0 секунд
Accuracy метрика: 0.09904407859798195
Precision метрика: 0.13381843915986413
Recall метрика: 0.09845238803059816
F1 метрика: 0.09293966333306201

Размер выборки: 5000 элементов
Время обучения: 0.01 секунд
Accuracy метрика: 0.13436006372809348
Precision метрика: 0.15271443676747531
Recall метрика: 0.1331199501485711
F1 метрика: 0.13120865154335565

Размер выборки: 10000 элементов
Время обучения: 0.01 секунд
Accuracy метрика: 0.1571959638874137
Precision метрика: 0.17501808977954642
Recall метрика: 0.15563265757442213
F1 метрика: 0.1545128906500431
```

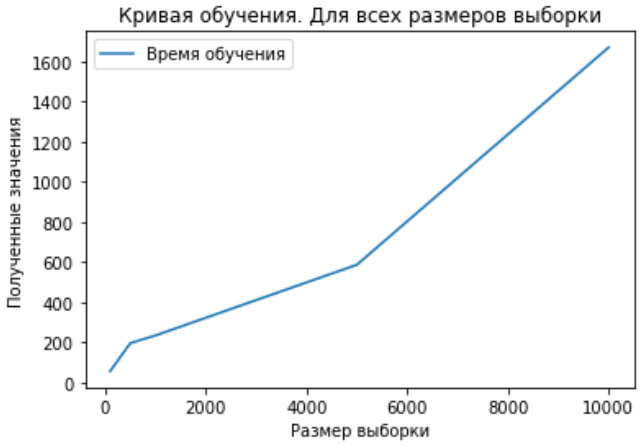


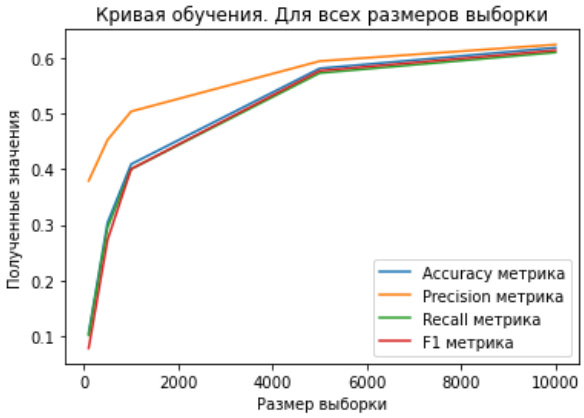


Многослойный перцептрон (MLPClassifier)

```
process(cls = MLPClassifier())
```

```
Размер выборки: 100 элементов
Время обучения: 54.35 секунд
Accuracy метрика: 0.104753053637812
Precision метрика: 0.3789321049796335
Recall метрика: 0.10119848156508995
F1 метрика: 0.0777880470070207

Размер выборки: 500 элементов
Время обучения: 193.78 секунд
Accuracy метрика: 0.30363781200212425
Precision метрика: 0.45256550918734917
Recall метрика: 0.29500844729567005
F1 метрика: 0.2724135725999503

Размер выборки: 1000 элементов
Время обучения: 232.18 секунд
Accuracy метрика: 0.40892193308550184
Precision метрика: 0.5038331730880199
Recall метрика: 0.40005199904111277
F1 метрика: 0.40013081399715367

Размер выборки: 5000 элементов
Время обучения: 584.6 секунд
Accuracy метрика: 0.5812533191715348
Precision метрика: 0.5943389071924553
Recall метрика: 0.5728935216576623
F1 метрика: 0.5770051500634981

Размер выборки: 10000 элементов
Время обучения: 1668.08 секунд
Accuracy метрика: 0.6184280403611259
Precision метрика: 0.6240983601497266
Recall метрика: 0.6100437828963479
F1 метрика: 0.6137083302638002
```

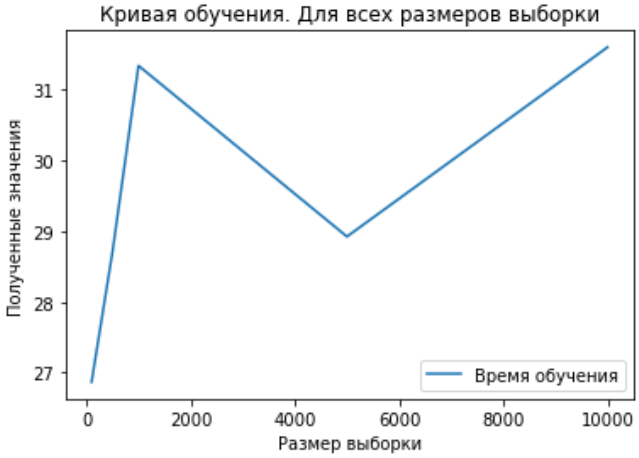


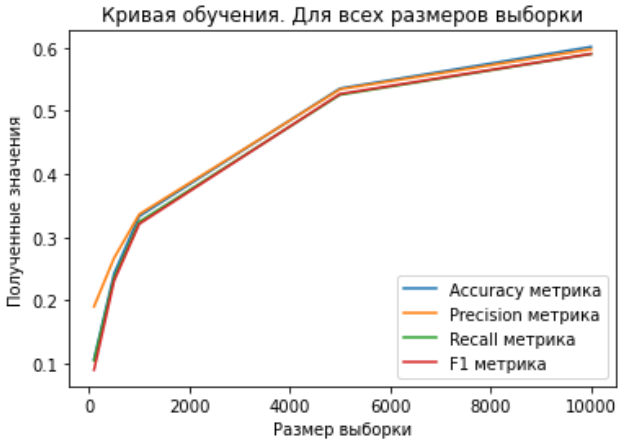


Логистическая регрессия (LogisticRegression)

```
process(cls = LogisticRegression())
```

```
Размер выборки: 100 элементов
Время обучения: 26.87 секунд
Accuracy метрика: 0.10621348911311737
Precision метрика: 0.18972990026410932
Recall метрика: 0.10410648140804454
F1 метрика: 0.08924794199159744

Размер выборки: 500 элементов
Время обучения: 28.7 секунд
Accuracy метрика: 0.242299522039299
Precision метрика: 0.26685459640207415
Recall метрика: 0.23552692330562802
F1 метрика: 0.2306386517402231

Размер выборки: 1000 элементов
Время обучения: 31.33 секунд
Accuracy метрика: 0.3328465215082315
Precision метрика: 0.3360787021396109
Recall метрика: 0.32403039192935335
F1 метрика: 0.32071765336379116

Размер выборки: 5000 элементов
Время обучения: 28.92 секунд
Accuracy метрика: 0.5357142857142857
Precision метрика: 0.5343628467180175
Recall метрика: 0.5256185383435513
F1 метрика: 0.5269034230819498

Размер выборки: 10000 элементов
Время обучения: 31.59 секунд
Accuracy метрика: 0.6016994158258099
Precision метрика: 0.5978868529081882
Recall метрика: 0.5902299469146306
F1 метрика: 0.5905503224940134
```

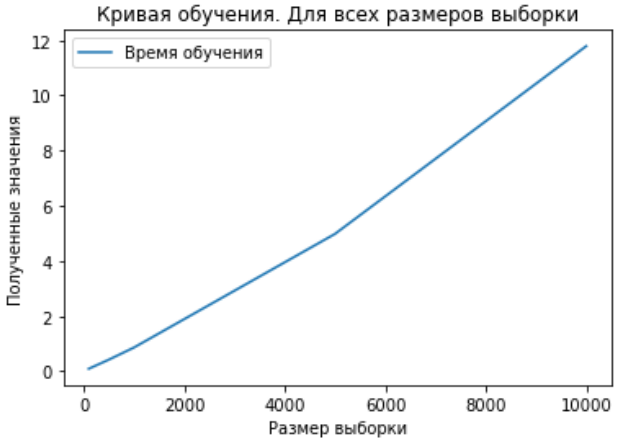


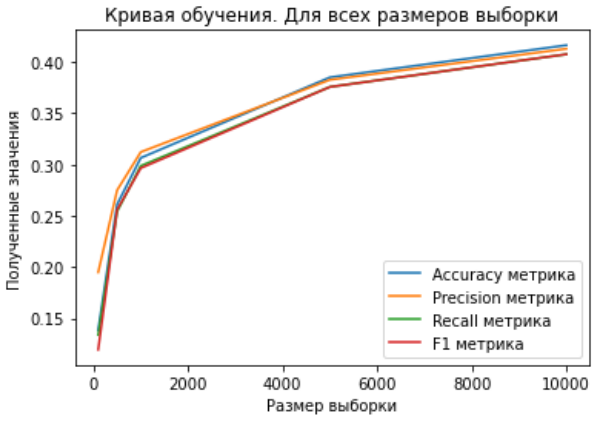


Классификатор дерева решений (DecisionTreeClassifier)

```
process(cls = DecisionTreeClassifier())
```

```
Размер выборки: 100 элементов
Время обучения: 0.09 секунд
Accuracy метрика: 0.13847583643122677
Precision метрика: 0.1948501058735874
Recall метрика: 0.13375878022517965
F1 метрика: 0.11936401160926276

Размер выборки: 500 элементов
Время обучения: 0.42 секунд
Accuracy метрика: 0.2611524163568773
Precision метрика: 0.2749046852522786
Recall метрика: 0.2541514208822009
F1 метрика: 0.2558282390420837

Размер выборки: 1000 элементов
Время обучения: 0.86 секунд
Accuracy метрика: 0.3064259160913436
Precision метрика: 0.31212946493330884
Recall метрика: 0.2986447434703295
F1 метрика: 0.2964378828800439

Размер выборки: 5000 элементов
Время обучения: 4.98 секунд
Accuracy метрика: 0.38502389803505044
Precision метрика: 0.3825978314653541
Recall метрика: 0.3757129325063251
F1 метрика: 0.37564515469076654

Размер выборки: 10000 элементов
Время обучения: 11.79 секунд
Accuracy метрика: 0.4163568773234201
Precision метрика: 0.4128478285510903
Recall метрика: 0.40749738455557616
F1 метрика: 0.40752224966636347
```

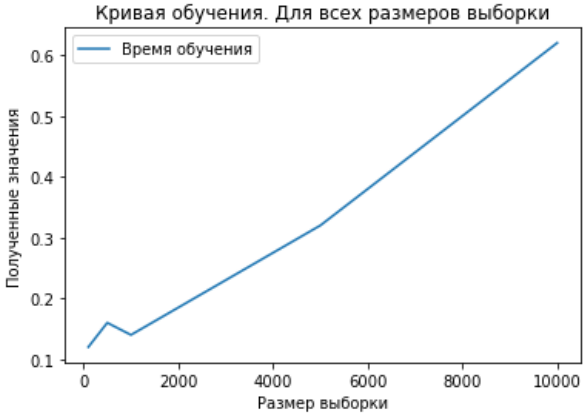


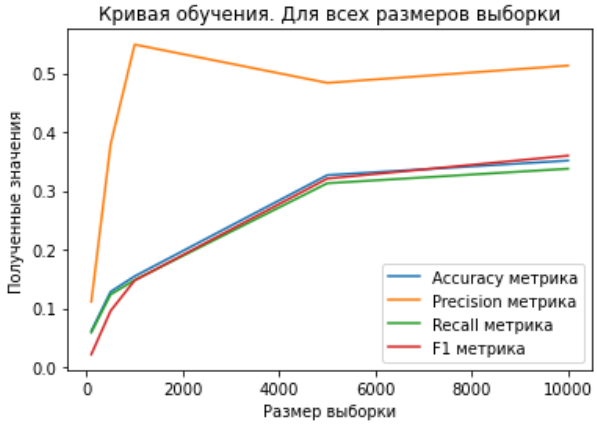


Классификатор «Случайный лес» (RandomForestClassifier)

```
process(cls = RandomForestClassifier(max_depth=2))
```

```
Размер выборки: 100 элементов
Время обучения: 0.12 секунд
Accuracy метрика: 0.06133828996282528
Precision метрика: 0.11119758959174024
Recall метрика: 0.058976284377005486
F1 метрика: 0.02166448992317794

Размер выборки: 500 элементов
Время обучения: 0.16 секунд
Accuracy метрика: 0.12812002124269783
Precision метрика: 0.379017079663289
Recall метрика: 0.123415423777473733
F1 метрика: 0.0955518702713603

Размер выборки: 1000 элементов
Время обучения: 0.14 секунд
Accuracy метрика: 0.15454062665958576
Precision метрика: 0.5487744392553647
Recall метрика: 0.1480213994226963
F1 метрика: 0.1474082863346105

Размер выборки: 5000 элементов
Время обучения: 0.32 секунд
Accuracy метрика: 0.3266064790228359
Precision метрика: 0.4835431970870939
Recall метрика: 0.312781328316703
F1 метрика: 0.3207010426060667

Размер выборки: 10000 элементов
Время обучения: 0.62 секунд
Accuracy метрика: 0.3513011152416357
Precision метрика: 0.5129633400804334
Recall метрика: 0.3374486292039528
F1 метрика: 0.3596986849285833
```

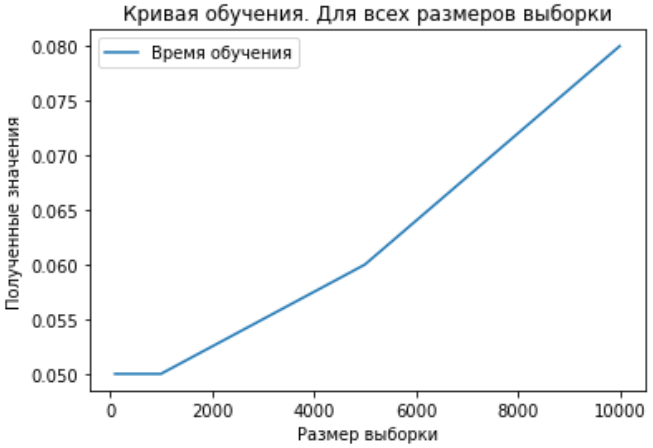


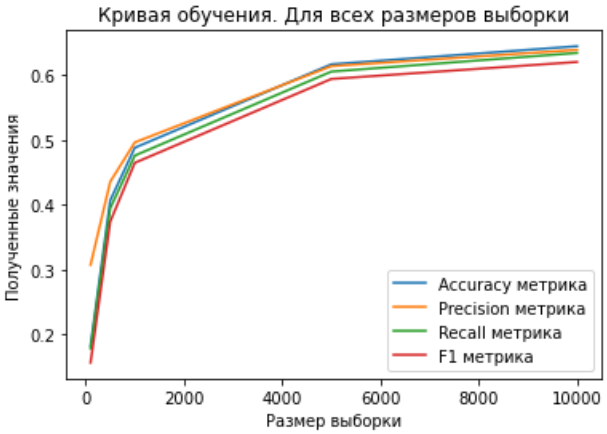


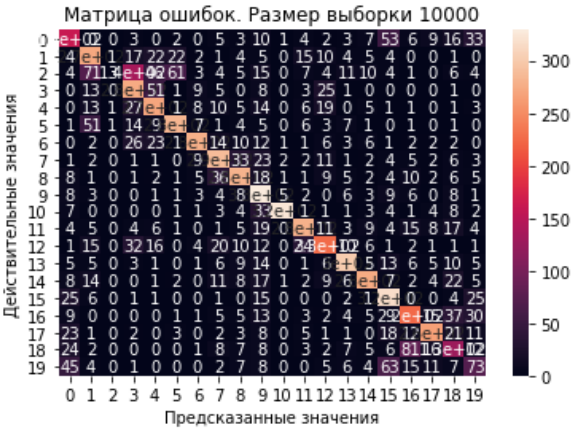
Наивный байесовский метод (MultinominalNB)

```
process(cls = MultinomialNB(alpha=.01))
```

```
Размер выборки: 100 элементов
Время обучения: 0.05 секунд
Accuracy метрика: 0.18228890069038767
Precision метрика: 0.3074740746712149
Recall метрика: 0.17841951811224727
F1 метрика: 0.1565500325752669

Размер выборки: 500 элементов
Время обучения: 0.05 секунд
Accuracy метрика: 0.40693043016463093
Precision метрика: 0.43563497918196503
Recall метрика: 0.39490365529546817
F1 метрика: 0.3740830410639945

Размер выборки: 1000 элементов
Время обучения: 0.05 секунд
Accuracy метрика: 0.4881837493361657
Precision метрика: 0.4964638681003527
Recall метрика: 0.47606808989901717
F1 метрика: 0.46485501381060823

Размер выборки: 5000 элементов
Время обучения: 0.06 секунд
Accuracy метрика: 0.6169676048858205
Precision метрика: 0.6141090565487991
Recall метрика: 0.6057015815101486
F1 метрика: 0.5943179199318414

Размер выборки: 10000 элементов
Время обучения: 0.08 секунд
Accuracy метрика: 0.6448486457780138
Precision метрика: 0.6391689801413923
Recall метрика: 0.6344615449246059
F1 метрика: 0.620531683848988
```







Вывод о моделях

Наиболее хорошо себя показали модели:

1. Наивный байесовский метод (MultinomialNB)
2. Логистическая регрессия (LogisticRegression)
3. Многослойный перцептрон (MLPClassifier)

Самая быстрая и точная модель:

1. Наивный байесовский метод (MultinomialNB)

Модель с самым долгим обучением:

1. Многослойный перцептрон (MLPClassifier)

Инструменты реализации

Среда разработки Jupyter Notebook

Jupyter Notebook – это среда разработки, где пользователь сразу может видеть выполнение кода или его отдельные фрагменты. Одной из главных отличий этой среды от других является возможность разбиения программного кода на куски с любым порядком выполнения. То есть вы можете создать класс или функцию и сразу же её проверить без запуска и выполнения всего программного кода. Также можно отдельно смотреть содержимое загруженных файлов и отдельно обработать данные файла. Этот функционал помогает достичь поставленных целей, так как, если мне захочется построить график прямо в середине кода я смогу просто запустить функцию и увидеть результат. Команда jupyter notebook создала свой ноутбук в облаке. Теперь пользователи могут пользоваться им через браузеры.

Язык программирования Python

Язык программирования Python – высокоуровневый ЯП общего назначения, который направлен на повышение производительности разработчика и читабельности кода. Этот язык имеет множество парадигм программирования, такие как объектно-ориентированное (ООП), императивное, структурное, функциональное и другие. У языка есть свободная лицензия, позволяющая использовать ЯП без ограничений в любых разработках.

Python хорошо подходит под выполнения математических задач, научных расчётов, а также для машинного обучения и анализа больших объёмов данных. Следующей причиной, по которой мы используем этот язык, является наличие большого количества различных библиотек для разработки приложений, а также анализа, обработки, визуализации данных, но основными пакетами в разработке машинного обучения и

анализа данных являются следующие библиотеки: NumPy, Pandas, Sklearn, Matplotlib, Seaborn, а также time, collections, itertools, warnings.

Библиотека NumPy

NumPy – это одна из популярных библиотек с открытым исходным кодом языка Python, которая помогает добавить поддержку огромных многомерных массивов и матриц, а также высокоуровневых и математических функций. Одним из основных объектов NumPy является одномерный массив.

Если реализовывать математические алгоритмы на чистом Python, то зачастую ваш программный код работает медленнее, чем на других компилируемых языках, поэтому NumPy помогает оптимизировать работу с многомерными массивами. NumPy написан на языке C.

Библиотека Pandas

Pandas – это высокоуровневая библиотека языка Python для анализа и обработки данных. Данная библиотека является самой продвинутой и перспективной в работе с данными. Самыми основными структурами данного пакета являются Series и DataFrame, которые нужны для манипуляции числовыми таблицами или временными рядами. Наименование Pandas происходит от термина «панельные данные». Pandas находится под лицензией BSD, которая позволяет свободно пользоваться данной библиотекой.

Возможности библиотеки:

- Манипуляция индексированными массивами двумерных массивов.
- Совмещение данных и обработка информации
- Объединение и слияние наборов данных

Библиотека Scikit Learn

Scikit-learn – это также один из популярнейших и широко использующих пакетов для машинного обучения и Data Science. C

помощью этого ПО можно выполнять большое количество операций. Scikit-learn имеет большое количество встроенных готовых алгоритмов. Эта библиотека имеет одну из лучших документаций о своих встроенных алгоритмах классах, методах и функциях.

Scikit-learn поддерживает:

- Выбор моделей
- Классификации
- Регрессии
- Предварительную обработку данных
- Кластерный анализ
- Уменьшение размерности

Scikit-learn не поддерживает:

- Нейронные сети
- Обучение с ассоциативными правилами
- Самоорганизующиеся карты (Кохонена сети)
- Обучения с подкреплением (Reinforcement learning)

У пакета есть свои наборы данных, с помощью которых можно тестировать свои модели. Так же как и вышеперечисленные библиотеки, Scikit-learn имеет открытый исходный код. Он бесплатный и лицензирован под BSD, как и Pandas-пакет.

Алгоритмы классификации:

- *LogisticRegression* (LogReg) – это один из главных алгоритмов данной библиотеки. Используется для отнесения исследований к дискретному набору классов. Эта регрессия преобразует свой вывод с помощью Сигмоида для возращения значения вероятности, которое в следствие может округлиться в сторону одного из дискретных классов.

- *KNeighborsClassifier* – это одна из контролируемых моделей машинного обучения. Модель учится на наборе помоченных данных. Сначала модель получает набор входных объектов и выходных значений. После приёма данных модель обучается на них, чтобы узнать, как сопоставить выходные данные с желаемыми выходными данными. Это, в свою очередь, нужно для того, чтоб модель могла делать прогнозы на невидимых данных. Работает она просто. В первую очередь модель берет ближайшие помеченные точки, глядя на «k». Затем происходит присвоение меток для большинства точек рядом с «k». Например, если k = 20, а пятнадцать из точек жёлтые, а пять остальных голубые, то рассматриваемая точка превратится в жёлтый, так как желтый цвет является большинством.
- *DecisionTreeClassifier* – это древоподобная структура, напоминающая блок-схему, в которой внутренний узел представляет функцию или атрибут. Ветвь представляет правило принятия решения, а каждый конечный узел - результат. Самый верхний узел – корневой узел. Он учится распределять на основе значения атрибута, рекурсивно разбивая дерево и вызывая тем самым рекурсивное разбиение. Это похоже на блок-схему, которая очень легко воспроизводит мысли на человеческом уровне. Именно поэтому деревья решений легко интерпретировать и понять.
- *SVM* (машина опорных векторов) – это один из алгоритмов классификации с контролируемым машинным обучением. Её результаты хороши, поэтому этот алгоритм является одним из популярных. SVM отличен от других алгоритмов классификации тем, что он выбирает границу принятия решения, которая максимизирует расстояние от ближайших

точек данных всех классов. SVM находит самую оптимальную границу принятия решения, то есть ту, которая имеет максимальный запас от близлежащих точек всех классов. Близлежащие точки от границы принятия решения, которые максимизируют расстояние между границей принятия решения и точками, называются опорными векторами.[8]

- *MLP* (Многослойный перцептрон) – это форма искусственной нейронной сети с прямого распределения, которая сопоставляет входные наборы данных с набором соответствующих выходных данных. MLP состоит из трёх слоев: входного, выходного и скрытого. Каждый слой полностью соединен со следующим. Узлами слоев являются нейроны с нелинейными функциями активации, за исключением узлов входного слоя.
- *Наивные байесовские оценки* — это вероятностные оценки, которые основаны на теореме Байеса, которая гласит, что между объектами существует сильная независимость. Байесовская теорема помогает узнать вероятность наступления событий на основе некоторого предварительного знания условий, которые могут быть связаны с событием. Наивные байесовские классификаторы[26] довольно хорошо работают для приложений классификации документов и фильтрации нежелательной почты. Он требует небольшого объема обучающих данных для настройки с учетом вероятностей для теоремы Байеса и поэтому работает довольно быстро. Scikit-Learn предоставляет список из 4 наивных байесовских оценок: 1) BernoulliNB - представляет собой классификатор, основанный на данных, представляющих собой многомерные распределения Бернулли.

2) GaussianNB - представляет собой классификатор, основанный на предположении, что вероятность признаков является гауссовым распределением.

3) ComplementNB - представляет собой классификатор, который использует дополнение каждого класса для вычисления весов модели. Это стандартный вариант многочленного наивного Байеса, который хорошо подходит для несбалансированных задач классификации классов.

4) MultinomialNB - представляет собой классификатор, который подходит для многомерно распределенных данных.

- *RandomForestClassifier* – это один из алгоритмов работы с учителем. Зачастую его используют для регрессии и для классификации. Этот алгоритм является одним из простых в использовании. Random Forest создаёт деревья решений для случайных sample-data. Затем делает прогноз от каждого дерева и в итоге выбирает лучший с помощью голосования. Бонусом является то, что он предоставляет мерку важности признаков. Случайный лес основан на алгоритме Борута, который определяет наиболее важные и значимые признаки датасета.

  Этапы работы случайного леса:

  1. Создание случайной выборки из датасета.
  2. Построение деревьев для каждой выборки и получение результатов.
  3. Голосование за каждый полученный прогноз.
  4. Окончательным результатом будет являться ветвь с наибольшим количеством голосов.

Библиотека Matplotlib

Matplotlib – это обширная библиотека для создания статических, анимированных и интерактивных визуализаций на Python. Matplotlib делает простые вещи простыми, а сложные возможными.

Возможности Matplotlib:

- Создание интерактивных фигур, которые можно масштабировать, панорамировать и обновлять.
- Экспорт во многие форматы файлов.
- Богатый набор сторонних пакетов, созданных на основе Matplotlib.
- Создание графиков качества публикации.
- Настройка визуального стиля и макета.
- Встраивание в JupyterLab и графические пользовательские интерфейсы.

ЗАКЛЮЧЕНИЕ

Поставленные задачи были реализованы в выполненной работе. Найден датасет с различными постами, содержащие текст, разбитый на различные темы и виды. Загрузка и исследование данных из датасета (приведены примеры текстов). Проведена векторизация текста, которая помогает машине конвертировать буквы в цифры для чтения и понимая слов в тексте. Создано девять различных моделей начиная с самых простых и быстрых заканчивая с самых сложных и долгих. Для моделей брались разные значения размера выборки (100, 500, 1000, 5000, 10000 элементов), но строились они с размером 10000. Время обучения у классификаторов были разные. Некоторые обучались за считанные секунды и даже доли секунд, но эффективность в некоторых моделях была минимальна. Другие более сложные модели обучались очень долго по несколько десяток минут, но гораздо эффективнее. Конечно, среди всех моделей есть те, что имеют среднюю эффективность и среднее время обучения.

Для улучшения качества обучения нашей машины стоит увеличить объём выборки и иметь большое количество объёмов данных.

Я думаю, направление классификации текстов нужно в наше время и его стоит развивать, так как существует много новостных сервисов, социальных сетей и СМИ, которые выпускают огромное количество текста, который можно исследовать и обучать машину для автоматизации сервисов.

ИСТОЧНИКИ

1. L.P. Coelho, W. Richert. Building machine learning system with Python – Packt Publishing, 2013. – 290 с.

2. J. Grus, Data Science from Scratch: First Principles with Python – O'Reilly Media, Inc, 2015. – 330 с.

3. C. Albon, Machine Learning with Python Cookbook – O'Reilly Media, Inc, 2018 – 366 с.

4. Aurelien Geron, Hands-on Machine Learning with Sckit-Learn, Keras and Tensorflow, 2nd Edition – O'Reilly Media, Inc, 2019. – 600 с.

5. The EMNIST Dataset [Электронный ресурс] // nist.gov. 28.03.2019. URL: https://www.nist.gov/itl/products-and-services/emnist-dataset

6. Getting started with EasyOCR for Optical Character Recognition [Электронный ресурс] // pyimagesearch.com. 14.09.2020. URL: https://pyimagesearch.com/2020/09/14/getting-started-with-easyocr-for-optical-character-recognition/

7. How to OCR with Tesseract, OpenCV and Python [Электронный ресурс] // nanonets.com. 10.02.2022. URL: https:/nanonets.com/blog/ocr-with-tesseract/ (https://nanonets.com/blog/ocr-with-tesseract/)

Свёрточные нейронные сети [Электронный ресурс] // habr.com. 31.01.2018. URL: https://habr.com/ru/post/348000/.

ИСХОДНЫЙ КОД ПРОГРАММЫ

Ссылка на GitHub-репозиторий: https://github.com/ffrrwwrr/Machine-learning