УДК 004.912

**МУЛЬТИКЛАССОВАЯ КЛАССИФИКАЦИЯ КОРПУСА СМЕШАННЫХ ТЕКСТОВ
АЛГОРИТМАМИ МАШИННОГО ОБУЧЕНИЯ**

Е.В. Гузеев

Научный руководитель: доцент, к.ф.-м.н. М.Е. Семёнов

Национальный исследовательский Томский политехнический университет,

Россия, г. Томск, пр. Ленина, 30, 634050

E-mail: evg39@tpu.ru

MULTI-CLASSIFICATION OF MIXED TEXTS CORPUS BY MACHINE LEARNING ALGORITHMS

E.V. Guzeev

Scientific Supervisor: Assoc. Prof., Ph.D. M.E. Semenov

Tomsk Polytechnic University, Russia, Tomsk, Lenin str., 30, 634050

E-mail: evg39@tpu.ru

Abstract. *This paper compares different machine learning algorithms for multiclass classification of mixed texts corpus. The F1-score was used as a quality metric for the algorithm comparison. The algorithms with acceptable quality for the corpus of texts were selected in the process. The dataset includes 7863 rows and 4 features, the gradient boosting showed the best result based on metric $F1=0.771$.*

Введение. Для решения задач бинарной классификации разработаны алгоритмы машинного обучения, к сожалению, в общем случае эти алгоритмы невозможно обобщить на случай мультиклассовой классификации, более того заранее невозможно определить какой алгоритм будет наилучшим в смысле качества классификации. Кратко приведем описание алгоритмов классификации, которые будем использовать в работе.

Логистическая регрессия при замене сигмоидной функции на функцию softmax может использоваться для задач мультиклассовой классификации. Алгоритм ближайших соседей также справляется с такими типами задач. Для этого необходимо отыскать k -ближайших соседей данных для входа X и вернуть класс, к которому принадлежит больше всего данных среди k [1, 2]. Если предположение о независимости признаков не нарушается, можно использовать полиномиальный наивный байесовский классификатор, который предполагает, что каждый класс имеет свой набор параметров. В алгоритме вычисляется вероятность появления класса в выборке. Если рассмотреть алгоритмы на базе решающих деревьев, то хорошо подходят алгоритм случайного леса и градиентного бустинга. Случайный лес состоит из нескольких деревьев решений, каждое из которых обучается на случайной подвыборке данных. Каждое дерево строится путем разбиения на две или более подгруппы на основе определенного признака. Главная идея градиентного бустинга заключается в том, что каждый следующее решающее дерево, учитывает ошибки предыдущих деревьев и пытается исправить их. Таким образом постепенно улучшается качество предсказаний [2, 3].

Цель работы – проведение сравнительного анализа качества алгоритмов машинного обучения для задачи мультиклассовой классификации. Для достижения цели мы поставили и последовательно решили следующие задачи:

1. Сбор (парсинг) исходных данных.
2. Предварительная обработка данных.
3. Подготовка данных для классификации.
4. Выбор метрики качества для классификации данных.
5. Выбор наилучшего алгоритма на основе метрики качества.

Экспериментальная часть. В качестве исходных данных мы использовали набор данных, который собрали на основе данных открытых групп в социальной сети ВК. Для тестирования различных алгоритмов мы сформулировали следующую гипотезу «Пользователь, который интересуется прикладной информатикой частично состоит в сообществах социальной сети по интересам». Эта гипотеза была проверена, на основе группировки и сортировки неуникальных данных, по которым видно, что люди состоят в сообществах так или иначе, связанных с программированием или математикой. Для сбора данных был написан парсер на основе VK API. Для каждого пользователя `user_id` собраны данные по 10 группам, в которых он состоит. Полученная таблица данных включает 7863 строки и 4 столбца: `description` (описание группы), `name` (название группы), `activity` (общая тематика группы), `group_id` (идентификатор группы). Исходные данные имели 54,5 % уникальных данных, после удаления дубликатов осталось 4365 строк. Далее мы провели разметку данных на 4 класса: 0 – группа не имеет отношения, а 3 – группа имеет прямое отношение к интересующей целевой аудитории. Распределение по классам (в долях): 0 – 0,52; 1 – 0,37; 2– 0,073; 3 – 0,037. Пример данных после предварительной обработки приведен на рисунке 1.

	description	activity	name	screen_name	target
0	Присылайте свои анекдоты в "Предложите новость...	Юмор	Азохней! Бесплатные щикарные анекдоты	azohen_vey	1
1	Познавательные мемы, широкий охват тем, уникал...	Юмор	Мемуары ценителей научных мемов	sciencemem	1
2	На Хабр Q&A можно получить ответ на вопрос по ...	Программирование	Хабр Q&A	toster_ru	4
3	Официальный канал сервиса РЕК Инвестиции для т...	Финансы	РЕК Инвестиции	rbc_investments	3
4	Помогаем новичкам стать веб-разработчиками, а ...	Программирование	Хекслет	hexlet	4

Рис. 1. Пример исходных данных

Отметим, что выборка не сбалансированная: мажоритарный класс в 14 раз больше миноритарного класса, поэтому мы выбрали для оценки качества классификации – метрику F_1 , которая представляет собой гармоническое среднее точности и полноты [1-3]:

$$F_1 = \frac{2 \cdot \text{точность} \cdot \text{полнота}}{\text{точность} + \text{полнота}}$$

Размеченный набор данных мы преобразовали в мешок слов, а затем применили частотный анализ текста. Для реализации мешка слов были проделаны следующие действия:

1. Удалили в тексте стоп-слова, которые не несут никакой смысловой нагрузки.
2. Разбили текстовые данные в документах на слова, привели все слова к исходной словоформе (процесс лемматизации и токенизации).
3. Сформировали словарь всех слов в корпусе документа. Провели расчет встречаемости каждого слова.
4. Создали векторное пространство из текстовых данных в виде разреженной матрицы.

Для формирования набора признаков данные были преобразованы в векторное TFIDF пространство. Это пространство может оценить важность слова в рамках конкретного документа: слову присваивается больший вес, если оно чаще встречается в документе, но при этом редко встречается в остальных документах корпуса. Если слово часто появляется в конкретном документе, но при этом редко встречается в остальных документах, оно вероятно, будет описывать содержимое данного документа лучше [1]. Таблица с преобразованным векторным пространством имеет 4365 строк и 23464 столбца.

	5789	5878	6098	6552	6686	7396	7629	7637	7761	7864	...	20614	20798	21046	21439	21804
0	0.070666	0.138949	0.539146	0.11321	0.074291	0.11321	0.132464	0.179032	0.138949	0.057835	...	0.093175	0.063489	0.11321	0.116779	0.072068
1	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	...	0.000000	0.000000	0.000000	0.000000	0.000000
2	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.207997	...	0.000000	0.000000	0.000000	0.000000	0.194389

<4365x23464 sparse matrix of type '<class 'numpy.float64''>
with 115993 stored elements in Compressed Sparse Row format>

Рис. 2. Пример данных в векторном TFIDF пространстве

Результаты. В результате работы для каждого алгоритма были рассчитаны метрики точности и полноты и на основе них был проведен расчет метрики F_1 (таблица 1). Для каждой модели были подобраны гиперпараметры для оптимизации метрики F_1 .

Таблица 1

Результаты сравнения качества работы алгоритмов

Алгоритм	Метрика F_1
Логистическая регрессия	0,729
Случайный лес	0,709
Наивный Байес	0,706
Ближайшие соседи	0,726
Градиентный бустинг (CatBOOST)	0,771

Заключение. В результате проведенного исследования показано, что наиболее успешно с задачей мультиклассовой классификации справились алгоритмы градиентного бустинга, логистическая регрессия и алгоритм ближайших соседей. В связи с тем, что задача классификации – это поиск вероятности отношения к тому или иному классу, а значит результаты можно улучшить путем построения ансамбля моделей.

СПИСОК ЛИТЕРАТУРЫ

1. Груздев А. В. Предварительная подготовка данных в Python: Том 1. Инструменты и валидация. – М.: ДМК Пресс, 2023. – 816 с.
2. Бурков А. Машинное обучение без лишних слов. – СПб.: Питер, 2020. – 192 с.
3. Мэрфи К.П. Вероятностное машинное обучение: введение / пер. с англ. А. А. Слинкина. – М.: ДМК Пресс, 2022. – 940 с.
4. Бурков А. Инженерия машинного обучения / пер. с англ. А. А. Слинкина. – М.: ДМК Пресс, 2022. – 306 с.
5. Васильев Ю. Обработка естественного языка. Python и spaCy на практике. — СПб.: Питер, 2021. – 256 с.