АНТИПЛАГИАТ
ОБНАРУЖЕНИЕ ЗАИМСТВОВАНИЙ

Отчет о проверке на заимствования №1

Автор: Пестов Егор Владимирович
Проверяющий: Пользователь для API (galkinala@fa.ru / ID: 2588)
Организация: Финансовый университет при Правительстве РФ

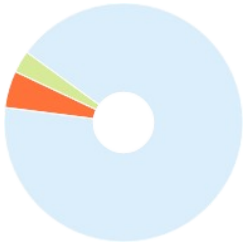
Отчет предоставлен сервисом «Антиплагиат» - http://fa.antiplagiat.ru

ИНФОРМАЦИЯ О ДОКУМЕНТЕ

№ документа: 1107941
Начало загрузки: 05.04.2022 16:55:29
Длительность загрузки: 00:00:14
Имя исходного файла: ВКР.docx
Название документа: ВКР.docx
Размер текста: 1 кБ
Символов в тексте: 43360
Слов в тексте: 5385
Число предложений: 271

ИНФОРМАЦИЯ ОБ ОТЧЕТЕ

Последний готовый отчет (ред.)
Начало проверки: 05.04.2022 16:55:43
Длительность проверки: 00:00:24
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
5,24%	0%	3,22%	91,54%

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	Комментарии
[01]	1,28%	2,34%	Баданина НД (171067)[ПМ17-1] Классификация текстов методами машинного обучения.pdf	05 Июн 2021	Модуль поиска "ФУ"	4	3	
[02]	0,2%	2,32%	ВКР_Васильев.docx	11 Мая 2021	Модуль поиска "ФУ"	1	3	
[03]	0%	2,12%	ВКР_Сальков_ДД_ПМ17_3_Разработк... ат_бота_для_рекомендательной.docx	06 Июн 2021	Модуль поиска "ФУ"	0	2	
[04]	0%	2,12%	Лафицкова АБД19-1м Магистерская ВКР.pdf	19 Мая 2021	Модуль поиска "ФУ"	0	2	
[05]	0%	2,12%	ВКР Хруцкий.docx	30 Мая 2021	Модуль поиска "ФУ"	0	2	
[06]	2,11%	2,11%	не указано	13 Янв 2022	Библиография	1	1	
[07]	0,46%	1,86%	ВКР_Шамырканова_РА_ПИ19-1м.pdf	21 Мая 2021	Модуль поиска "ФУ"	1	2	
[08]	0%	1,86%	ВКР_Бахматов АВ_ПМ17-2 Классификация транзакционных данных по расширенному описанию.pdf	05 Июн 2021	Модуль поиска "ФУ"	0	2	
[09]	0%	1,78%	Гирфанов ПМ 17-3 Разработка рекомендательной системы в области финансовый рынков на основе предиктивной аналитики больших данных.docx	10 Июн 2021	Модуль поиска "ФУ"	0	3	
[10]	1,31%	1,31%	ДИССЕРТАЦИЯ v3.docx	25 Июн 2020	Кольцо вузов	2	2	
[11]	0%	1,27%	направленность программы магистратуры "Анализ больших данных и машинное обучение в экономике и финансах" Методические рекомендации по подготовке и защите ВКР	13 Авг 2020	Интернет Плюс	0	5	

			http://fa.ru				
[12]	0%	1,21%	профиль "Аналитическое и информационное обеспечение финансово-экономической деятельности" http://fa.ru	31 Мар 2020	Интернет Плюс	0	4
[13]	0%	1,21%	Методические_рекомендации_ВКР_ПМиИ.pdf http://fa.ru	20 Авг 2019	Интернет Плюс	0	4
[14]	0%	1,21%	профиль "Аналитическое и информационное обеспечение финансово-экономической деятельности" http://fa.ru	30 Сен 2020	Интернет Плюс	0	4
[15]	0%	1,18%	ЧижоваИА(181469)[ПМ18-4]«Машинное обучение в задачах классификации текстов».pdf	11 Дек 2020	Модуль поиска "ФУ"	0	1
[16]	0%	1,13%	направленность программы магистратуры "Интеллектуальные информационные технологии в экономике и финансах" Методические рекомендации по подготовке и защите ВКР http://fa.ru	13 Авг 2020	Интернет Плюс	0	4
[17]	1,11%	1,11%	не указано	13 Янв 2022	Шаблонные фразы	11	11
[18]	0%	1,03%	Методические_рекомендации_ВКР_ПИ.pdf http://fa.ru	20 Авг 2019	Интернет Плюс	0	3
[19]	0%	1,03%	Программа ГИА ПИ, КИС, 2016.pdf http://fa.ru	13 Авг 2020	Интернет Плюс	0	3
[20]	0%	0,94%	http://www.fa.ru/org/dep/findata/SiteAssets/Pages/bak/%D0%9C%D0%B5%D1%82%D0%BE%D0%B4%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%B8%D0%B5_%D1%80%D0%B5%D0%BA%D0%BE%D0%BC%D0%B5%D0%BD%D0%B4%D0%B0%D1%86%D0%B8%D0%B8_%D0%92%D0%9A%D0%A0_%D0%9F%D0%9C%D0%B8%D0%98.pdf http://fa.ru	15 Мар 2020	Интернет Плюс	0	2
[21]	0%	0,94%	http://www.fa.ru/org/dep/findata/SiteAssets/Pages/bak/%D0%9C%D0%B5%D1%82%D0%BE%D0%B4%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%B8%D0%B5_%D1%80%D0%B5%D0%BA%D0%BE%D0%BC%D0%B5%D0%BD%D0%B4%D0%B0%D1%86%D0%B8%D0%B8_%D0%92%D0%9A%D0%A0_%D0%9F%D0%98.pdf http://fa.ru	27 Окт 2020	Интернет Плюс	0	2
[22]	0%	0,74%	10-Баран_БИ17-3_ВКР.pdf	24 Мая 2021	Модуль поиска "ФУ"	0	1
[23]	0%	0,72%	профиль "Налоги и налогообложение" http://fa.ru	13 Авг 2020	Интернет Плюс	0	3
[24]	0%	0,72%	направленность программы магистратуры "Бизнес-аналитика" http://fa.ru	13 Авг 2020	Интернет Плюс	0	3
[25]	0%	0,69%	направленность программы магистратуры "Современное банковское дело и модели управления" (заочная форма обучения) http://fa.ru	13 Авг 2020	Интернет Плюс	0	3
[26]	0%	0,6%	http://www.fa.ru/org/div/umoop/Documents/%D0%9F%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D1%8B%20%D0%93%D0%AD_%D0%93%D0%98%D0%90/%D0%92%D1%8B%D0%BF%D1%83%D1%81%D0%BA%202022/%D0%93%D0%98%D0%90_%D0%9C%D0%B0%D0%B3%D0%B8%D1%81%D1%82%D1%80%D0%B0%D1%82%D1%83... http://fa.ru	10 Янв 2022	Интернет Плюс	0	1
[27]	0,26%	0,55%	DILAF: A framework for distributed analysis of large-scale system logs for anomaly detection https://doi.org	28 Фев 2019	Издательство Wiley	2	5
[28]	0,53%	0,53%	robin_i_yu_detektirovanie-negativnogo-kontenta-v-seti-internet-s-primeneniem-mashinnogo-obucheniya.docx	19 Мая 2020	Кольцо вузов	3	3
[29]	0,28%	0,47%	Igor_Krivulec	09 Июн 2017	Кольцо вузов	1	2
[30]	0,47%	0,47%	1.9. Наивные методы Байеса - scikit-learn https://scikit-learn.ru	29 Ноя 2021	Интернет Плюс	1	1

[31]	0,45%	0,45%	Как обучить Word2vec на русскоязычных twitter-постах с Python https://python-school.ru	24 Ноя 2020	Интернет Плюс	3	3	
[32]	0%	0,45%	Types of taxes http://revolution.allbest.ru	08 Янв 2018	Переводные заимствования (RuEn)	0	1	
[33]	0%	0,42%	Влияние плана BEPS ОЭСР на современный международный бизнес https://nauchkor.ru	06 Июл 2020	Интернет Плюс	0	1	
[34]	0%	0,4%	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	
[35]	0%	0,4%	Решения ученого совета от 20.09.2016 http://fa.ru	29 Янв 2017	Перефразирования по Интернету	0	1	
[36]	0%	0,39%	Экономическая и финансовая эффективность проведения спортивных мероприятий https://knowledge.allbest.ru	05 Апр 2022	Интернет Плюс	0	2	
[37]	0%	0,38%	Отделение прикладной математики и информатики/471ПМ Чусовлянов Дмитрий Сергеевич Машинное обучение для определения тональности и классификации текстов на несколько классов 381982526de5c	06 Июн 2014	Кольцо вузов	0	1	
[38]	0%	0,34%	БУХГАЛТЕРСКИЙ УЧЁТ: современные вызовы, приоритеты и пути развития. Т. 2 https://book.ru	03 Июл 2017	Сводная коллекция ЭБС	0	1	
[39]	0%	0,34%	Бухгалтерский учет:современные вызовы, приоритеты и пути развития. Том 4 https://book.ru	03 Июл 2017	Сводная коллекция ЭБС	0	1	
[40]	0%	0,34%	Вопросы теории и практики налогообложения: сборник научных статей https://e.lanbook.com	22 Янв 2020	Сводная коллекция ЭБС	0	1	
[41]	0%	0,34%	Сборник официальных документов и материалов 4/2016 http://studentlibrary.ru	20 Дек 2016	Медицина	0	1	
[42]	0%	0,34%	Управление прибылью в акционерных обществах региона: теория и практика. Книга 1 http://studentlibrary.ru	19 Дек 2016	Медицина	0	1	
[43]	0%	0,34%	Ресурс: Новое прочтение и геоэкономическое измерение экспортного потенциала http://studentlibrary.ru	19 Дек 2016	Медицина	0	1	
[44]	0%	0,31%	Страницы - Бакалавриат http://fa.ru	20 Дек 2020	Интернет Плюс	0	1	
[45]	0%	0,24%	Список вузов Москвы с экзаменами егэ "информатика и ИКТ, математика, русский язык" https://vuzoteka.ru	20 Дек 2020	Интернет Плюс	0	1	
[46]	0%	0,23%	СРАВНИТЕЛЬНЫЙ АНАЛИЗ МЕТОДОВ МАШИННОГО ОБУЧЕНИЯ ДЛЯ РЕШЕНИЯ ЗАДАЧИ КЛАССИФИКАЦИИ ДОКУМЕНТОВ НАУЧНО-ОБРАЗОВАТЕЛЬНОГО УЧРЕЖДЕНИЯ. http://elibrary.ru	14 Янв 2020	eLIBRARY.RU	0	1	
[47]	0%	0,22%	neganova_e_a_razrabotka-metodiki-povysheniya-kachestva-modeley-mashinnogo-obucheniya-na-osnovanii-metodov-subgrou.pdf	23 Мая 2020	Кольцо вузов	0	1	
[48]	0%	0,2%	https://official.satbayev.university/download/document/8839/%D0%92%D0%95%D0%A1%D0%A2%D0%9D%D0%98%D0%9A-2019%20%E2%84%962.pdf https://official.satbayev.university	16 Фев 2022	Интернет Плюс	0	1	Источник исключен. Причина: Маленький процент пересечения.
[49]	0%	0,2%	Методика автоматического определения аутентичности данных, представленных в текстах на естественном языке. http://elibrary.ru	24 Янв 2020	eLIBRARY.RU	0	1	Источник исключен. Причина: Маленький процент пересечения.
[50]	0%	0,2%	171912_m1-IFST21_2019_1	15 Фев 2022	Кольцо вузов	0	1	Источник исключен. Причина: Маленький процент пересечения.
[51]	0%	0,2%	180683_m2-IFST21_2020_1	10 Фев 2022	Кольцо вузов	0	1	Источник исключен. Причина: Маленький процент пересечения.
			A privacy-preserving density peak					Источник исключен.

[52]	0%	0,17%	clustering algorithm in cloud computing https://doi.org	10 Июн 2020	Издательство Wiley	0	1	Причина: Маленький процент пересечения.
[53]	0%	0,15%	Corporate social responsibility “glocalisation”: Evidence from the international construction business https://doi.org	31 Мар 2020	Издательство Wiley	0	2	Источник исключен. Причина: Маленький процент пересечения.
[54]	0%	0,15%	Диссертация на тему «Методы и модели анализа данных в управлении образовательным процессом образовательной организации МВД России», скачать бесплатно автореферат по специальности ВАК РФ 05.13.10 - Управление в социальных и экономических системах https://dissercat.com	11 Дек 2019	Интернет Плюс	0	1	Источник исключен. Причина: Маленький процент пересечения.
[55]	0%	0,15%	https://mvd.ru/upload/site120/folder_page/011/790/378/Dissertatsiya_Kuznetsova.pdf https://mvd.ru	25 Июн 2020	Интернет Плюс	0	1	Источник исключен. Причина: Маленький процент пересечения.
[56]	0%	0,15%	https://xn--b1aew.xn--p1ai/upload/site120/folder_page/011/790/378/Dissertatsiya_Kuznetsova.pdf https://xn--b1aew.xn--p1ai	05 Апр 2022	Интернет Плюс	0	1	Источник исключен. Причина: Маленький процент пересечения.
[57]	0%	0,14%	не указано	13 Янв 2022	Цитирование	0	1	Источник исключен. Причина: Маленький процент пересечения.

Федеральное государственное образовательное бюджетное учреждение высшего образования

«Финансовый университет при Правительстве Российской Федерации»

(Финансовый университет)

Факультет прикладной математики и информационных технологий

Департамент анализа данных и машинного обучения

Выпускная квалификационная работа

на тему: «**Машинное обучение для анализа тональности текстов**»

Направление подготовки 01.03.02 «Прикладная математика и информатика»,
Профиль «Анализ данных и принятие решений в экономике и финансах»

Выполнил студент группы ПМ18-3

Пестов Егор Владимирович

Руководитель д.т.н., доцент, профессор

Судаков Владимир Анатольевич

_____

ВКР соответствует предъявляемым требованиям

Руководитель Департамента
д.э.н., профессор

_____ В.И. Соловьев

«__» _____20__ г.

Москва 2022

Содержание

ВВЕДЕНИЕ

В условиях динамичного развития онлайн-сервисов и социальных сетей пользователи Интернета получили обширный спектр возможностей, среди которых числится и возможность свободно, но в рамках разумного, выражать своё мнение. Это могут быть различные отзывы на продукты питания, товары и услуги, художественные произведения (фильмы, картины, книги), а также просто точки зрения и высказывания обычных людей по любому другому поводу. Логично, что сформулированное по 17кому поводу мнение может оказаться как позитивным, так и негативным. Здесь берёт начало термин – тональность. Людям, по психологической специфике, важно иметь платформу для выражения своих мыслей, а также быть услышанными. И социальные сети или агрегаторы отзывов – лучшее место для удовлетворения данной потребности.

Даже на первый взгляд вполне очевидно предполагать, что ручная обработка, к примеру, комментариев в соцсетях довольно затруднительна, а в общем смысле попросту невозможна. Количество публикуемых пользователями комментариев исчисляется сотнями тысяч, и это далеко не предел. Во многом, именно по этой причине стали развиваться такие сферы цифрового сегмента, как «Opinion Mining» или «Sentiment Analysis». Всё это вплотную связано с машинным обучением, которое предлагает методы анализа общественного мнения с использованием словарей тональности.

Главной и основополагающей задачей здесь можно назвать классификацию высказанных мнений на два (позитивные и негативные) класса. В некоторых особо углублённых процессах классификация может быть обширнее – не всё делится на чёрное и белое, а поэтому инициализируется более, чем два основных класса; существуют нейтральные и смешанные мнения, к которым необходимо применять более продуманную и человеко-ориентированную методологию. Также в задачах можно выделить выдержку мнений из исходных текстов с целью нахождения в них субъективной составляющей – ключевой аспект человеческого видения.

Исследование и анализ отзывов пользователей помогает узнать отношение клиентской базы к многим нюансам – усреднённый процент отзывов формирует рейтинг, на основе которого другие люди принимают решение, стоит ли им пользоваться услугами данного сервиса или компании. Негативные и позитивные мнения в качестве альтернативных точек зре17я,17 взаимно дополняют друг друга. Таким образом, можно прийти к выводу, что сфера определения тональности текстов весьма актуальна и важна в современном мире, где люди полагаются на мнения других, и порой на их основе строят своё собственное. И это важный элемент не только для потребителей товаров и услуг, но и для руководителей компаний и предприятий, которым тоже важно учитывать обратную связь людей, ради которых и существует их прямая деятельность.

Но с другой стороны, если отстраниться от принципа полезности изучения чужих мнений, не стоит забывать и про то, что в остальных случаях, например, во время обсуждения острых политических тем в социальных сетях – чтение чужих комментариев может стать причиной для расстройства. Все мы прекрасно понимаем современные реалии, где в сети можно повсеместно нарваться на вопиющий негатив, безосновательные оскорбления и в принципе на хамское отношение от совершенно чужих и незнакомых людей. За последние годы разработчики соцсетей встали перед дилеммой – оставлять свои платформы местом с нерушимым принципом свободы слова или же подвергать негативные мнения людей цензуре с целью предотвращения кибербуллинга и прочих видов интернет-травли. Поскольку вопросы психологического здоровья в обществе поднимаются всё чаще, а отношение к этой проблеме меняется в лучшую сторону, это довольно актуальный и злободневный нюанс на повестке дня.

Именно поэтому, имея в виду изложенные тезисы, особенно важно с особым вниманием относиться к сфере «Sentiment Analysis» и задачам определения тональности текстов.

Исследования в данной работе проводились на основе реальных данных, предоставленных источником LinisCrowd.

Глава 1. Понятие тональности текстов и задачи её анализа.

Прежде всего имеет смысл уточнить определение самого понятия анализа тональности текстов. Оно подразумевает собой совокупность методов, использующихся в целях автоматизации выявления в текстах эмоциональной окраски и субъективности мнений авторов по отношению к спектру различных объектов, о которых говорят авторы.

Под тональностью можно понимать и просто эмоциональную оценку, которая выражается в письменной речи. Стоит отметить, что тональность текста можно определить по трём ключевым факторам:

- тональной оценке (эмоциональное отношение автора к объекту);
- объекту тональности (сам автор мнения);
- субъекту тональности (то, о чём высказывает своё мнение автор).

Под целями работы стоит понимать анализ, подбор оптимальной методологии, программную реализацию и сравнение результатов работы используемых методов.

1.1. Проблематика в анализе тональности текстов.

Рассуждая о проблематике рассматриваемой задачи, необходимо понимать, что сами по себе мнения людей субъективны, а значит, субъективны и методы определения эмоциональной оценки текстов. Людям свойственно по-разному реагировать на одинаковые вещи и иметь противоположные мнения. Помимо этого, вольно написанные пользователями тексты нельзя отнести к структурированным объектам информации – из-за этого с ними не так легко работать. Манера изложения мыслей у каждого человека индивидуальна; в речи может присутствовать юмор, сарказм и элементарные опечатки – компьютер всё это попросту не поймёт. Выходя за лингвистические рамки, стоит также принять во внимание тот факт, что методы, разработанные для одного языка, могут быть неприменимы для другого.

Каждая отдельная задача в сфере анализа тональности текстов подразумевает собой хорошо подобранный и выверенный подход к предметной области. Одни и те же слова из разных областей могут иметь разную эмоциональную окраску. Например, в сфере киноиндустрии «очень *страшный* фильм» в отзыве на картину жанра «ужасы» является позитивным мнением, но «очень *страшный* район» в жилищной сфере говорит о негативном отношении к упомянутому объекту.

1.2. Обзор и постановка задачи классификации текстов

Для написания работы были установлены следующие задачи:

- Предварительный анализ и описательная статистика подобранных данных для машинного обучения;
- Обзор методов анализа мнений и их последующей классификации;

- Исследование операций по обработке естественного языка;
- Построение словаря, содержащего все слова из исходного набора данных;
- Обучение нескольких моделей-классификаторов тональности;
- Оценка результатов проведённого обучения;
- Эксперименты на реальных данных, проверка эффективности методов;
- Анализ текстов при помощи нейронных сетей;
- Подбор оптимальных методов.

Глава 2. Машинное обучение для определения тональности текстов.

Говоря о компьютеризированной реализации анализа тональности текстов, нельзя не подумать о машинном обучении. Наиболее распространённые его классы: обучение на размеченных данных и обучение с учителем. К обоим этим классам относятся процессы, в которых, имея конечное число примеров, необходимо научить компьютер прогнозировать определённую величину для конкретного объекта. В задачах автоматизированной классификации текстов используются заранее размеченные корпусы данных – именно на них обучаются модели перед грядущим использованием.

Но существуют и другие способы определения тональности текстов – например те, которые базируются на работе со словарями оценочной лексики (словарями тональности). Существует три основных типа таких способа:

- экспертный;
- основанный на текстовых коллекциях;
- основанный на тезаурусах (словарях).

Экспертный метод подразумевает собой составление словаря экспертами (людьми) вручную. По понятным причинам этот способ сильно отличается от других: он очень трудоёмкий, поскольку исключается фактор автоматизации процесса, а также в нём велика вероятность отсутствия специальной лексики из определённых предметных областей – все их попросту не охватить. Но есть и преимущества: итоговый продукт окажется словарём с присвоенной ему тональностью высокого качества – такой словарь хоть и будет составлен медленно, но исключит в себе множественные погрешности, которые мог бы допустить компьютер, а человек их бы не заметил в силу огромного количества обработанных данных.

Способ, основывающийся на текстах, по которым в итоге составляются словари, подразумевает статистический анализ размеченных текстовых данных. Подобные коллекции текстов имеют отношение к тем предметным областям, в которых и составляются словари. Данный подход уменьшает шанс отсутствия уникальных терминов из заданной предметной области. Но стоит помнить, что в таком случае качество итогового словаря тональности прямо зависит от качества изначально размеченных текстов.

Наконец, при способе, основанном на тезаурусах, полученный на вход список слов пополняется за счёт привлечения новых словарей. К таким дополнительным словарям можно отнести коллекции позитивно окрашенных слов, негативно окрашенных, антонимов, синонимов и т.д. Недостатком данного метода является сложность в вопросах сохранения и корректного поддержания связей составляемых словарей с изначально заданной предметной областью.

Проанализировав преимущества и недостатки методов анализа тональности текстов, основанных на машинном обучении или на словарях тональности, а также приняв во внимание тему выпускной квалификационной работы, было принято решение сделать выбор в пользу первого метода – основанного на машинном обучении. В качестве альтернативы словарям тональности в следующей главе будет обзорно рассмотрена работа модуля Word2Vec – совокупности моделей, которые предназначены для векторного представления слов на естественном языке.

2.1. Импортирование библиотек и описательная статистика.

С целью приступить к описанию датасета на экран были выведены первые и последние строки таблицы – это позволило убедиться, что набор данных успешно считался и готов к работе.

```
[ ]  1  DS.head()
```

	Comment	Rate
0	Не рациональная системность, а интуитивный поз...	0
1	Когда возникнут трудности, они тебе не помогут...	0
2	Кривая национализация это политический компром...	-1
3	Такой вид биологического оружия не действует н...	-2
4	В Эль-Кусейре /к западу от Хомса/ сирийские по...	0

```
[ ]  1  DS.tail()
```

	Comment	Rate
32432	Это помогло Соединенным Штатам прорубить окно ...	0
32433	Она уже вернулась на УИК. Член ПСГ Строгин от ...	-2
32434	Всего было две линии обороны: в первую входили...	-1
32435	5. Западный образ жизни несовместим с выживани...	-2
32436	Оригинал взят у в Светлая память - защитнику О...	0

Таблица 1. Первые и последние строки датафрейма

У нас имеются два атрибута:

- **«Comment»** – категориальный признак, в котором хранятся все многочисленные мнения пользователей социальных сетей;
- **«Rate»** – численный признак, который хранит оценки тональности мнения. Также данный атрибут является целевой переменной в нашей задаче.

Продолжая изучать атрибут «Rate» более подробно, было выяснено, что диапазон его значений варьируется в промежутке [-2;2], где отрицательные числа (-1;-2) обозначают негативную оценку, ноль (0) – нейтральную, а положительные (1;2) – позитивную оценку соответствующего мнения. Для наглядности была построена круговая диаграмма, которая показала соотношение этих трёх типов тональности.

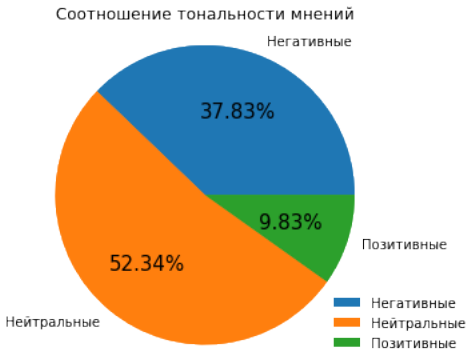


Диаграмма 1. Соотношение тональности мнений (%)

В процессе описания дата-сета на экран ноутбука были выведены следующие информационные сообщения:

- Количество точек данных (измерений) в наборе;
- Количество полей данных (атрибутов);
- Названия всех атрибутов в наборе данных;
- Общая информация о дата-сете;
- Общая описательная статистика дата-сета;
- Тип данных каждого поля, шкала каждого поля;
- Количество уникальных значений для каждого атрибута;
- Количество отсутствующих значений для каждого поля.

Количество уникальных значений для каждого атрибута:

```
[ ]  1  DS.nunique()

Comment    20381
Rate           6
dtype: int64
```

Таблица 2. Количество уникальных значений для каждого атрибута

Описательная статистика показала, что атрибут «Rate» хранит 6 различных переменных. Но так быть не должно, ведь ранее речь шла о диапазоне [-2;2], в котором находятся лишь 5 чисел. Это значит, что в данных затесались лишние значения, от которых на следующем этапе столбец будет очищен.

Количество отсутствующих значений для каждого поля:

```
[ ]  1  len(DS) - DS.count()

Comment    31
Rate       13
dtype: int64
```

То же самое, но в процентном представлении:

```
[ ]  1  (DS.isna().sum(axis=0)/len(DS)).sort_values()

Rate       0.000401
Comment    0.000956
dtype: float64
```

Таблица 3. Количество отсутствующих значений для каждого поля (также в %)

Помимо этого, имеет смысл очистить данные и от отсутствующих значений, пусть их в нашем датасете и немного – в обоих атрибутах менее 1%.

2.2. Предобработка данных.

Прежде чем передать данные в работу моделей машинного обучения, необходимо обработать и очистить их. Очевидно, что «грязные» и необработанные данные могут содержать искажения и пропущенные значения – это ненадёжно, поскольку способно привести к крайне неверным результатам по итогам моделирования. Но безосновательно удалять что-либо тоже неправильно. Именно поэтому сначала набор данных надо изучить.

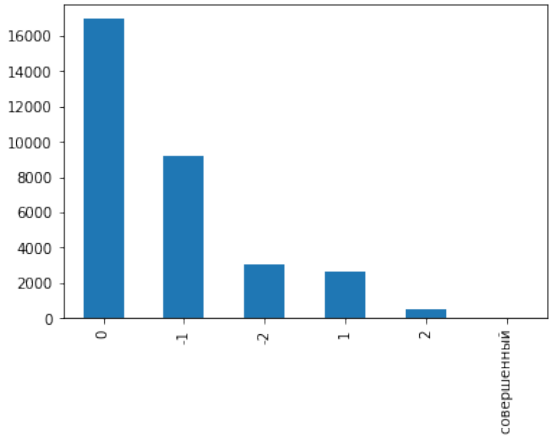


Диаграмма 2. Подсчёт значений в атрибуте «Rate»

Учитывая наше предыдущее наблюдение, появление буквенного значения в численном атрибуте не стало сюрпризом. Более того, из-за наличия в «Rate» данного мусора, столбец попросту не может считаться численным – на данный момент его тип данных заявлен программой как «object» (т.е. содержит несколько разных типов).

Поэтому перед нами встаёт необходимость очистить датафрейм от этого. Привычнее, если вид целевой переменной будет качественным бинарным. Для этого в том же порядке избавимся от тех текстов, чья тональность была оценена как нейтральная (Rate = 0). Отныне «0» будет обозначать негативную тональность. Значения «-2» и «-1» объединены, так же, как и «1» и «2».

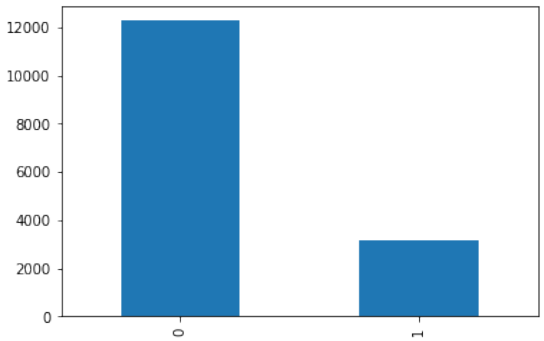


Диаграмма 3. Подсчёт значений в атрибуте «Rate» после объединения и очистки

Был создан индикатор для признаков с пропущенными данными во всём датасете (0 – первый столбец «Comment», 1 – второй столбец «Rate»), а затем на основе индикатора построена гистограмма, которая показала, что отсутствующие значения присутствуют лишь в атрибуте «Comment».

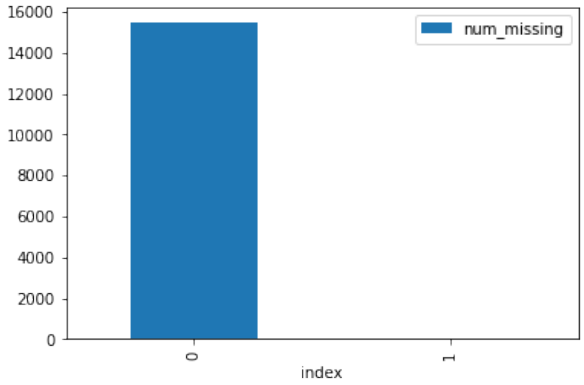


Диаграмма 4. Количество отсутствующих значений в наборе данных

```
Количество строк в датасете до очистки:  15465
Суммарное количество отсутсвующих значений (Nan) в датасете:  14
Таким образом, после очистки пропущенных значений в датасете должно остаться 15451 строк
```

Таблица 4. Прогнозные подсчёты количества строк после процесса очистки данных

При помощи функции drop() из библиотеки pandas из набора данных были удалены те строки, в которых было обнаружено более одного пропуска. Прогнозные подсчёты (Таблица 4) сошлись с реальным положением дел (Таблица 5).

```
1  ind_missing = DS[DS['num_missing'] > 0].index # если количество пропусков больше нуля
2  DS_less_missing_rows = DS.drop(ind_missing, axis=0,inplace=True)
3  DS
```

	Comment	Rate	Comment_ismissing	Rate_ismissing	num_missing
2	Кривая национализация это политический компром...	0.0	False	False	0
3	Такой вид биологического оружия не действует н...	0.0	False	False	0
6	бактериофобия, верминофобия, вермифобия, гельм...	0.0	False	False	0
8	Президент Сирии также ответил на обвинения в а...	0.0	False	False	0
12	Великий писатель беспрестанно доказывал несост...	1.0	False	False	0
...	...	...	...	...	...
32426	Сейчас говорят, что собираются снова переподчи...	0.0	False	False	0
32429	Жертвами атак боевиков стали четыре военнослуж...	0.0	False	False	0
32433	Она уже вернулась на УИК. Член ПСГ Строгин от ...	0.0	False	False	0
32434	Всего было две линии обороны: в первую входили...	0.0	False	False	0
32435	5. Западный образ жизни несовместим с выживани...	0.0	False	False	0

15451 rows × 5 columns

Таблица 5. Информация о размерности датасета и первые/последние его строки после очистки

После успешно проведённого процесса очистки исходных данных от пропусков и мусора логические атрибуты Comment_ismissing, Rate_ismissing и num_missing были удалены при помощи функции del. Атрибуты Comment и Rate сохранены в формате списков.

Стоит заметить, что сфера работы находится в границах Natural Language Processing (NLP) — одного из направлений искусственного интеллекта, которое работает с анализом, пониманием и генерацией живых языков с целью взаимодействовать с компьютерами, используя естественные языки вместо компьютерных.

На следующем этапе предобработки необходимо провести векторизацию текстовых данных. Тем не менее, перед преобразованием текстовых единиц в числа, необходимо их обработать специальными способами, которые предлагают Python-библиотеки «pymorphy2» и «NLTK». Среди них можно перечислить удаление стоп-слов, стемминг и лемматизацию.

```
!pip install pymorphy2
```

```
Requirement already satisfied: pymorphy2 in /usr/local/lib/python3.7/dist-packages (0.9.1)
Requirement already satisfied: docopt>=0.6 in /usr/local/lib/python3.7/dist-packages (from pymorphy2) (0.6.2)
Requirement already satisfied: dawg-python>=0.7.1 in /usr/local/lib/python3.7/dist-packages (from pymorphy2) (0.7.2)
Requirement already satisfied: pymorphy2-dicts-ru<3.0,>=2.4 in /usr/local/lib/python3.7/dist-packages (from pymorphy2)
```

```
pip install pymorphy2[fast]
```

```
Requirement already satisfied: pymorphy2[fast] in /usr/local/lib/python3.7/dist-packages (0.9.1)
Requirement already satisfied: pymorphy2-dicts-ru<3.0,>=2.4 in /usr/local/lib/python3.7/dist-packages (from pymorphy2[f
Requirement already satisfied: dawg-python>=0.7.1 in /usr/local/lib/python3.7/dist-packages (from pymorphy2[fast]) (0.7
Requirement already satisfied: docopt>=0.6 in /usr/local/lib/python3.7/dist-packages (from pymorphy2[fast]) (0.6.2)
Requirement already satisfied: DAWG>=0.8 in /usr/local/lib/python3.7/dist-packages (from pymorphy2[fast]) (0.8.0)
```

```
pip install -U pymorphy2-dicts-ru
```

```
Requirement already satisfied: pymorphy2-dicts-ru in /usr/local/lib/python3.7/dist-packages (2.4.417127.4579844)
```

Таблица 6. Информационные сообщения об успешной установке библиотеки pymorphy2

```
import pymorphy2
morph = pymorphy2.MorphAnalyzer()
```

```
import os
import requests
from pathlib import Path
import nltk
from nltk import sent_tokenize, word_tokenize, regexp_tokenize
from nltk.corpus import stopwords
import pymorphy2
from collections import Counter
```

```
import nltk
nltk.download('punkt')
```

```
[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data]   Package punkt is already up-to-date!
True
```

Таблица 7. Установка и импорт библиотек для NLP-операций

Имеет смысл рассмотреть по отдельности и реализовать NLP-операции на наших данных.

1. **Исключение стоп-слов из текстовых данных.**

Всегда бывает так, что некоторых слов в тексте больше, чем каких-то других. Речь идёт о таких словах, которые встречаются почти в каждом предложении

и в общем-то не несут информационной нагрузки. Подобные текстовые единицы – шум, который именуется стоп-словами.

Была написана функция get_text, которая предназначена для чтения текстовых данных – в нашем случае это список стоп-слов со специального ресурса на GitHub, а также все предложения из исходного датасета.

с	будет	был	вам	ведь	вместе
а	будете	была	вами	везде	вне
алло	будешь	были	вас	вернуться	вниз
без	будто	было	ваш	весь	внизу
белый	буду	быть	ваша	вечер	во
близко	будут	в	ваше	взгляд	вода
более	будь	важная	ваши	взять	война
больше	бы	важное	вверх	вид	вокруг
большой	бывает	важные	вдали	видел	вон
будем	бывь	важный	вдруг	видеть	вообще

Таблица 8. Фрагмент списка стоп-слов

При необходимости список стоп-слов можно дополнять любыми другими словами, которые покажутся лишними в процессе работы. Так как объект stopwords_ru является списком, то в него свободно можно добавлять (а также удалять из него) слова при помощи функции append().

2. **Стеммизация (стемминг).**

Ещё со школы все мы помним и знаем, насколько русский язык обширен и богат с точки зрения морфологии. К примеру, прилагательные "замечательн**ый**" и "замечательн**ая**" - слова идентичного смысла, но различной морфологической структуры (просклонены по двум разным родам). Но в машинном обучении вопрос минимизации данных и их очищения всегда стоит довольно остро; поэтому имеет смысл привести подобные слова к одной форме для уменьшения размерности.

Одним из способов сделать это можно назвать стемминг (stemming). Его суть заключается в отделении от слова окончаний, тем самым оставляя от слов лишь основу. В Python-библиотеке «NLTK» для этого существует метод «Snowball Stemmer» с поддержкой русского языка.

Слово	Окончание	Стем
замечательный	-ый	замечательн
база	-а	баз

Таблица 9. Пример работы стемминга

3. **Лемматизация:**

Метод лемматизации позволяет приводить слова к своей начальной морфологической форме. Например, "вижу", "видят" имеют начальную форму "видеть". С реализацией данного метода в Python нам поможет библиотека pymorphy2, которая является специально написанным для морфологического анализа русского языка инструментом.

Существует метод «parse», который возвращает список обозначающих некоторые важные грамматические формы анализируемых нами слов. Рассмотрим подробнее атрибуты данного метода:

- tag – морфологический разбор слова. Определяет часть речи, вид, переходность, число, лицо, время и наклонение;
- normal_form — инфинитив слова, его изначальная форма;
- score — правильность морфологического разбора в процентах.

По понятным причинам нам больше всех нужен атрибут normal_form. Так как объекты метода сортируются в порядке убывания значения score, то нам необходимо взять самый первый элемент (под нулевым индексом).

```
1  morph.parse("видят")[0].normal_form
```

```
'видеть'
```

```
1  morph.parse("вижу")[0].normal_form
```

```
'видеть'
```

```
1  morph.parse("видим")[0].normal_form
```

```
'видеть'
```

Таблица 10. Пример работы лемматизации

В результате принято решение отказаться от стемминга в пользу лемматизации. Последняя является более предпочтительным способом морфологического анализа. В отличие от стеммизации она не отрубает от слов окончания, превращая эти самые слова в бессмысленные в лингвистическом смысле объекты. Специальный инструмент, который работает на основе словарей русского языка, подходит для нашей работы больше, поскольку позволяет получить одно слово из разных его форм, в то же время не травмируя исходное слово.

4. **Нормализация.**

Для осуществления нормализации исходного текста была написана функция remove_special_characters, которая на основе специальных паттернов очистила строки в атрибуте «Comment» от ссылок, специальных символов, повторяющихся символов, чисел и английских букв (так как наши данные преимущественно русскоязычные и изначально ориентированы на русский язык). Также текст был целиком приведен к нижнему регистру.

```python
def remove_special_characters(text, remove_digits=True):
    urlPattern        = r"((http://)[^ ]*|(https://)[^ ]*|( www\.)[^ ]*)"
    sequencePattern   = r"(.)\1\1+"
    seqReplacePattern = r"\1\1"
    text = text.lower()
    # Заменяем все ссылки на 'URL':
    text = re.sub(urlPattern,' URL', text)
    # Заменяем 3 одинаковых символа подряд на 2:
    text = re.sub(sequencePattern, seqReplacePattern, text)
    pattern=r'[^а-яА-ЯёЁa-zA-Z0-9]'
    pattern_1=r'[A-Za-z0-9_]'
    text=re.sub(pattern,' ',text)
    text=re.sub(pattern_1,' ',text)
    return text
```

Таблица 11. Функция remove_special_characters

После применённой к обработанному тексту токенезации он стал пригодным для перевода в числовую форму (векторизации). Был выведен список из 20 самых часто встречающихся слов в датасете, чтобы дальше продолжить извлечение признаков.

```
[('боязнь', 29054),
 ('боевик', 6349),
 ('район', 5917),
 ('сирийский', 4540),
 ('террорист', 4455),
 ('сирия', 4356),
 ('армия', 3996),
 ('военный', 3716),
 ('оружие', 3662),
 ('уничтожить', 3582),
 ('группировка', 3284),
 ('провинция', 2696),
 ('результат', 2620),
 ('российский', 2510),
 ('являться', 2432),
 ('государство', 2394),
 ('дамаск', 2324),
 ('число', 2198),
 ('находиться', 2142),
 ('войско', 2091)]
```

Таблица 12. Самые часто встречающиеся в датасете слова и их частота

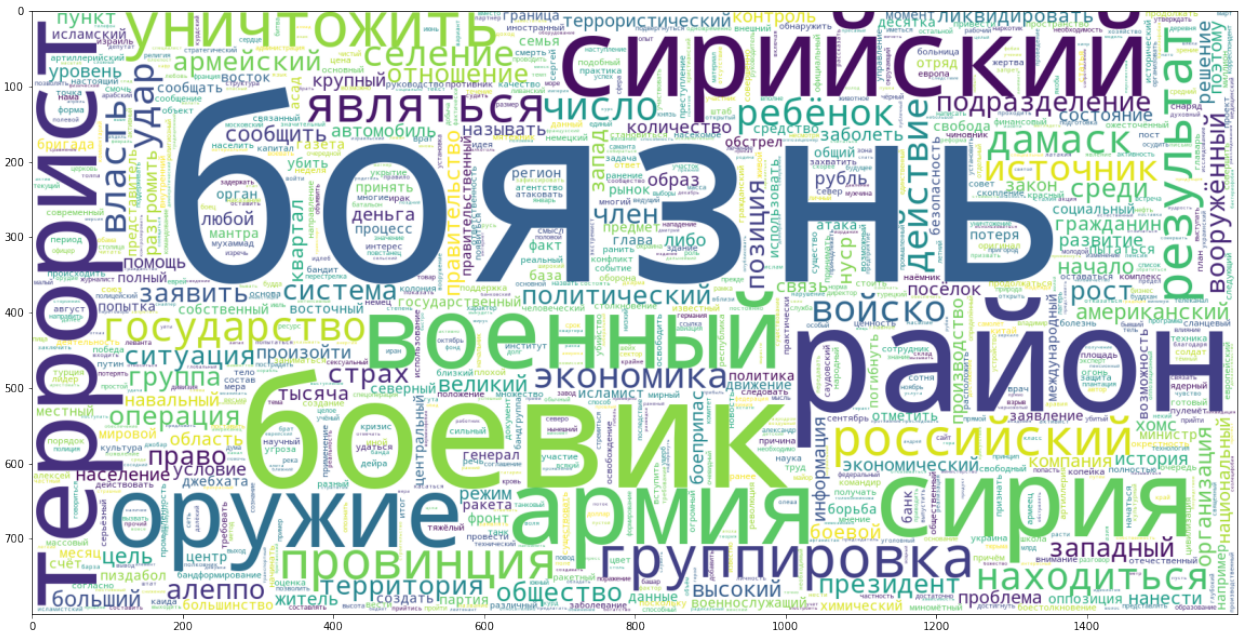


Диаграмма 5. Облако с самыми часто встречающимися в датасете словами

2.3. Разделение набора данных на обучающую и тестовую выборки.
Машинное обучение по своему определению является процессом нахождения особых свойств выборки данных и применения этих свойств уже на новых данных. Поэтому, исходя из общепринятой практики оценки алгоритма в машинном обучении, на данном этапе работы было проведено разбиение данных на два отдельных набора. Первый — обучающая выборка, на которой будут изучаться свойства данных, а второй — тестовая выборка, на которой выделенные свойства будут тестироваться.

В процессе разделения был использован случайный метод, который к тому же работает по умолчанию. Прочие методы было решено не использовать в силу специфики задачи.

Разделили набор данных на обучающую и тестовую выборки с помощью метода train_test_split из библиотеки sklearn. Параметр test_size = 0,2 внутри функции указывает процент данных, которые должны быть сохранены для тестирования. Обычно это соотношения около 80/20 или 70/30. Параметр random_state = 0 – это начальное значение (номер) случайного числа. Он необходим в процессе повторных тестов, где будет гарантированно получено случайное число с таким же значением.

Разбиение данных на две этих выборки выглядело следующим образом:

```
1  X_train, X_test, y_train, y_test = train_test_split(processedtext_done_1, Rate, test_size = 0.2, random_state = 0)
2  print(f'Разбиение данных прошло успешно.')
```

```
Разбиение данных прошло успешно.
```

Таблица 13. Разбиение данных на обучающую и тестовую выборки

Далее была проведена векторизация данных при помощи TF-IDF Vectorizer.

Данный инструментарий по определению — подход из сферы статистики, использующийся в решении задач по оцениванию контекстной важности слова внутри документа/корпуса слов, участвующих в анализе. Основной тезис данного подхода гласит, что вес конкретного слова пропорционален частоте его употребления в тексте, а также обратно пропорционален частоте употребления данного слова во всей коллекции текстов.

TF-IDF определяет важность каждого слова с целью проанализировать текстовый документ или любой другой обрабатываемый текст в датасете. Разобраться в работе данного инструмента можно на следующем практическом примере: допустим, у нас имеется набор данных, состоящий из сочинений школьников на тему "Мой дом". В таком датасете слово "дом" может появляться множество раз; это часто встречающееся слово в сравнении с другими словами в датасете. Набор данных содержит и другие слова. Например, "квартира", "комнаты" и т.д., но они уже встречаются реже, а поэтому их частота встречаемости ниже. Следовательно, они несут в себе больше информации, нежели слово "дом". Такова логика инструмента TF-IDF.

Векторизация при помощи "TF-IDF Vectorizer" конвертирует набор необработанных текстовых данных в матрицу "TF-IDF features". Vectorizer обычно обучается только лишь на наборе данных X_train.

```
1  vectoriser = TfidfVectorizer(ngram_range=(1,2), max_features=560000)
2  vectoriser.fit(X_train)
3  print(f'Векторизация прошла успешно.')
4  print('Количество слов: ', len(vectoriser.get_feature_names()))
```

```
Векторизация прошла успешно.
/usr/local/lib/python3.7/dist-packages/sklearn/utils/deprecation.py:87
  warnings.warn(msg, category=FutureWarning)
Количество слов:  513985
```

Таблица 14. Процесс векторизации данных

2.4. Обучение моделей для решения выбранной задачи.

Создадим 9 моделей для решения нашей задачи по анализу тональности текстов.

- Алгоритм наивного байесовского классификатора: метод Бернулли (BernoulliNB);
- Метод Опорных Векторов (LinearSVC);
- Логистическая регрессия (Logistic Regression);
- Мешок сл27 (Bag of Words) и Модель Term Frequency-Inverse Document Frequency (TFIDF);
- Оснащённые стохастическим градиентным спуском модели Bag of Words и TFIDF;
- Дерево решений и случайный лес (Decision Tree Classifier и Random Forest Classifier).

Оценочной метрикой выберем показатель accuracy — долю правильных ответов алгоритма. Для оценки эффективности работы моделей также будем рассматривать метрики precision (точность) и recall (полнота). Под precision принято понимать объекты, названные классификатором положительными, и оставшиеся действительно положительными даже после проверки. Recall демонстрирует конкретную часть положительных объектов из всех обнаруженных алгоритмом и при этом задействованных в модели. Чтобы получить максимально подробную информацию по работе модели, имеет смысл выводить классификационные отчёты **(classification_report)**.

И перед переходом к анализу метрик необходимо использовать **confusion matrix** (матрицу ошибок), которая имеет следующий вид:

	$y = 1$	$y = 0$
$\hat{y} = 1$	True Positive (TP)	False Positive (FP)
$\hat{y} = 0$	False Negative (FN)	True Negative (TN)

Таблица 15. Условный вид матрицы ошибок (confusion matrix)

Высокие показатели precision и recall говорят о том, что построенная модель эффективна, а её качество её работы оценивается как высокое. Не менее важной метрикой является f-мера, которая объединяет данные о точности и полноте алгоритма, представляя собой гармоническое среднее между ними. Значение этой метрики зависит от значений recall и precision. Например, если они обе стремятся к нулю, тогда значение f-меры будет делать то же самое.

Значения f-меры рассчитываются на основе confusion matrix. Формулы метрик имеют вид:

$$Precision = \frac{TP}{TP+FP}$$

$$Recall = \frac{TP}{TP+FN}$$

$$F = \frac{2*Precision*Recall}{Precision+Recall}$$

Для подробного и визуализированного вывода классификационного отчёта и матрицы ошибок была написана функция model_Evaluate. Также внутри неё осуществлялись процессы предсказания значений для обучающей выборки.

2.4.1. Алгоритм наивного байесовского классификатора (BernoulliNB)

Наивные методы Байеса являются комплексом алгоритмов машинного обучения, которые основываются на теореме Байеса с неким предположением о том, что существует условная независимость между всеми парами характеристик при определённом значении переменной класса, которое было задано в программе ранее.

Было решено использовать метод BernoulliNB, который реализует простые байесовские алгоритмы обучения и классификации данных, тем самым распределяя их в соответствии с многомерным распределением Бернулли. Предполагается, что каждая из функций является двоичной переменной. Соответственно, данный класс требует, чтобы образцы были представлены как векторы признаков с двоичными значениями; если переданы данные любого другого ти[30] BernoulliNB экземпляр может преобразовать свой ввод в двоичную форму.

Решающее правило для BernoulliNB выглядит следующим образом:

$$P(x_i \mid y) = P(i \mid y)\, x_i + (1 - P(i \mid y))(1 - x_i)$$

Преимущества:

- высокий показатель скорости функционирования алгоритма;
- нетрудная программная реализация;
- доступная интерпретируемость результатов работы модели.

Недостатки:

- неудовлетворительный показатель качества классификации;
- отсутствие учёта зависимости результатов классификации от сочетания признаков.

2.4.2. Метод Опорных Векторов (LinearSVC)

Метод Опорных Векторов[10] является линейным алгоритмом, который используется в решении задач классификации и регрессии. Этот метод имеет широкое распространение в вопросах машинного обучения, поскольку он используется в решении задач линейного и нелинейного плана. Концепция работы опорных векторов заключается в том, что алгоритмом создаётся линия/гиперплоскость, с помощью которых данные разделяются на классы.

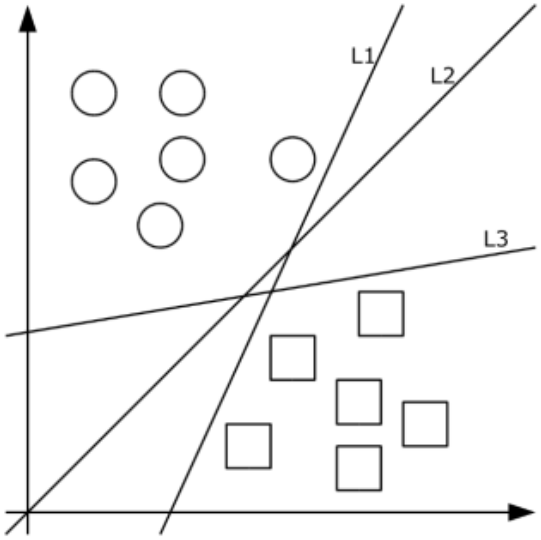


Рисунок 1 – Классифицирующие разделяющие прямые метода «SVM»

Преимущества:

- Метод является эффективным при использовании в пространствах большого размера и в случаях, когда число измерений оказывается выше числа образцов.
- Из соображения эффективности в плане использования памяти, алгоритм использует в себе подмножество обучающих точек в функции принятия решений;
- Алгоритм универсален в использовании в задачах различного толка.

Недостатки:

- параметры метода трудно интерпретировать;
- модель довольно неустойчива к выбросам в исходных наборах данных.

2.4.3. Логистическая регрессия (Logistic Regression)

Логистическая регрессия является моделью машинного обучения, которая используется для решения задач классификации, что делает её очередным подходящим нашей работе кандидатом.

Задачи, в которых применяется метод логистической регрессии, должны иметь строго две категории. Иными словами, должны быть задачами двоичной классификации. Разным категориям назначаются 0 и 1 соответственно.

В таких методах подбирается особый вектор коэффициентов $b = (w1,w2,\ldots,wn)^T$, позже он используется для осуществления процесса классификации:

$$f\ x = 11+\exp(-x,b) = \sigma(\ x,b\ )$$

Преимущества:

- модель выступает одним из наиболее эффективных методов;
- алгоритм оснащён довольно простой программной реализацией.

Недостатки:

- трудно интерпретируются параметры модели;
- алгоритм неустойчив по отношению к выбросам в исходных наборах данных.

2.4.4. Мешок слов (Bag of Words) и модель Term Frequency-Inverse Document Frequency (TFIDF)

«Мешок слов» является упрощенным представлением, которое используется в процессе обработки естественного языка (NLP) и при поиске информации. В данной модели текстовые данные представляются как мультимножества слов, из которых они состоят; порядок текстовых единиц и грамматика не играют тут роли, но множественность сохраняется.

Метод обычно используется в задачах классификации документов, где частота появления каждого слова используется как признак для обучения исходного классификатора. Он довольно успешно применяется в решении задач моделирования языка и классификации документов различных типов.

В TF-IDF наибольший вес имеют те слова, которые чаще всего встречаются в пределах одного текста, и которые реже всего встречается в остальных документах.

Также была предпринята попытка модернизировать данные методы при помощи стохастического градиентного спуска – таким образом, класс «SGDClassifier» реализовал нетрудный процесс обучения, который поддерживал функции потерь и штрафы за классификацию. Данная процедура оказалась эквивалентна линейному SVM (методу опорных векторов), который мы уже рассматривали ранее.

2.4.5. Decision Tree Classifier и Random Forest Classifier

Дерево решений является методом прогнозного моделирования, которое применяется во множестве предметных областей различного толка. Обычно они строятся с помощью подхода, определяющего способы разделения набора данных на основе разных признаков и условий. Деревья решений – это непараметрический метод обучения с учителем, который используется в задачах регрессии и классификации. Цель данного метода состоит в том, чтобы создать модель, которая предскажет значение целевой переменной на основе изучения простых правил принятия решений, полученных из характеристик данных.

Преимущества:

- довольно простая реализация алгоритма;

- легко интерпретируемые результаты работы.

Недостатки:

- алгоритм неустойчив к выбросам в исходных наборах данных;
- для получения наиболее точных результатов, необходима работа с большими объёмами данных.

Случайный лес — один из самых универсальных методов классификации. Его главное преимущество в том, что он применяется в весьма широком спектре практических задач. Помимо этого, существуют особые Random Forest модели, которые используются для решения задач классификации, регрессии и кластеризации. Это как раз попадает под контекст нашей задачи анализа тональности текстов.

Глава 3. Эксперименты и сравнение методов.

3.1. Выбор наиболее перспективной модели для решения задачи.

Для того, чтобы выбрать наиболее перспективную модель для решения поставленной задачи по анализу тональности текстов, сначала необходимо провести апробацию всех построенных моделей на тестовой выборке. По итогам тестов будет построена таблица с оценками качества работы моделей на основе нескольких оценочных метрик (precision, recall, f1-score, accuracy).

3.1.1. Апробация моделей на тестовой выборке.

Модель 1: Алгоритм наивного байесовского классификатора: метод Бернулли (BernoulliNB).

	precision	recall	f1-score	support
0.0	0.80	1.00	0.89	2468
1.0	0.80	0.01	0.01	623
accuracy			0.80	3091
macro avg	0.80	0.50	0.45	3091
weighted avg	0.80	0.80	0.71	3091

Таблица 16. Классификационный отчёт по модели BernoulliNB

Диаграмма 6. Матрица ошибок для модели BernoulliNB

По итогу метрика accuracy по первой модели (BernoulliNB) показала себя хорошо, дав 80% точности.

Модель 2: Метод Опорных Векторов (LinearSVC).

	precision	recall	f1-score	support
0.0	0.98	1.00	0.99	2468
1.0	0.99	0.93	0.96	623
accuracy			0.98	3091
macro avg	0.98	0.96	0.97	3091
weighted avg	0.98	0.98	0.98	3091

Таблица 17. Классификационный отчёт по модели LinearSVC

Диаграмма 7. Матрица ошибок для модели LinearSVC

По итогу метрика accuracy по второй модели (LinearSVC) показала себя почти идеально, дав 98% точности.

Модель 3: Логистическая регрессия (Logistic Regression).

	precision	recall	f1-score	support
0.0	0.86	1.00	0.92	2468
1.0	0.99	0.35	0.52	623
accuracy			0.87	3091
macro avg	0.93	0.68	0.72	3091
weighted avg	0.89	0.87	0.84	3091

Таблица 16. Классификационный отчёт по модели Logistic Regression

Диаграмма 8. Матрица ошибок для модели Logistic Regression

По итогу метрика accuracy по третьей модели (LinearSVC) показала себя несколько хуже предыдущей (второй модели), дав 87% точности.

Модель 4: Мешок 28ов (Bag of Words) и **Модель 5**: Term Frequency-Inverse Document Frequency (TFIDF).

	precision	recall	f1-score	support
Positive	0.89	0.91	0.90	2772
Negative	0.08	0.07	0.07	319
accuracy			0.82	3091
macro avg	0.49	0.49	0.49	3091
weighted avg	0.81	0.82	0.82	3091

Таблица 18. Классификационный отчёт по модели Bag of Words

Диаграмма 9. Матрица ошибок для модели Bag of Words

	precision	recall	f1-score	support
Positive	0.90	1.00	0.95	2772
Negative	0.00	0.00	0.00	319
accuracy			0.90	3091
macro avg	0.45	0.50	0.47	3091
weighted avg	0.80	0.90	0.85	3091

Таблица 19. Классификационный отчёт по модели TFIDF

Диаграмма 10. Матрица ошибок для модели TFIDF

Модель 6 и **Модель 7**: Оснащённые стохастическим градиентным спуском Bag of Words и TF-IDF.

	precision	recall	f1-score	support
Positive	0.89	0.91	0.90	2772
Negative	0.07	0.06	0.07	319
accuracy			0.82	3091
macro avg	0.48	0.48	0.48	3091
weighted avg	0.81	0.82	0.81	3091

Таблица 20. Классификационный отчёт по модели SMV_Bag of Words

	precision	recall	f1-score	support
Positive	0.90	1.00	0.95	2772
Negative	1.00	0.00	0.01	319
accuracy			0.90	3091
macro avg	0.95	0.50	0.48	3091
weighted avg	0.91	0.90	0.85	3091

Таблица 21. Классификационный отчёт по модели SMV_TFIDF

Диаграмма 11. Матрица ошибок для модели SMV_Bag of Words

Диаграмма 12. Матрица ошибок для модели SMV_TFIDF

Оснащённые стохастическим градиентным спуском модели Bag of Words и TF-IDF уверенно показали себя на тестовой выборке:

- Метрика accuracy в случае SVM_Bag of Words: 82% точности
- Метрика accuracy в случае SVM_TF-IDF: 90% точности

Модель 8: Дерево решений (Decision Tree Classifier) и **Модель 9**: Случайный лес (Random Forest Classifier).

```
Accuracy of DecisionTreeClassifieris 0.80944678097703
              precision    recall  f1-score   support

         0.0       0.89      0.87      0.88      2535
         1.0       0.47      0.53      0.50       556

    accuracy                           0.81      3091
   macro avg       0.68      0.70      0.69      3091
weighted avg       0.82      0.81      0.81      3091

[[   1  318]
 [   0 2772]]
```

Таблица 22. Классификационный отчёт и матрица ошибок по модели Decision Tree Classifier

```
Accuracy of RandomForestClassifieris 0.8330637334196053
              precision    recall  f1-score   support

         0.0       0.94      0.86      0.90      2678
         1.0       0.42      0.63      0.50       413

    accuracy                           0.83      3091
   macro avg       0.68      0.75      0.70      3091
weighted avg       0.87      0.83      0.85      3091

[[   1  318]
 [   0 2772]]
```

Таблица 23. Классификационный отчёт и матрица ошибок по модели Random Forest Classifier

Оба классификатора показали себя хорошо на тестовой выборке:

- Метрика accuracy в случае DecisionTree: 81% точности;
- Метрика accuracy в случае RandomForest: 83% точности.

3.1.2. Анализ и сравнение результатов

Как говорилось ранее, для наглядности была построена таблица с оценками точности каждой модели. Это помогло определиться с лучшими и худшими результатами.

	Полное название модели	Рабочее название модели	Точность (precision)	Полнота (recall)	F-мера (f1-score)	Общая точность предсказания (accuracy)	Общая точность предсказания (accuracy)[%, округлённая]
0	Алгоритм наивного байесовского классификатора: метод Бернулли	BernoulliNB	0.800000	0.800000	0.710000	0.799418	80.000000
1	Метод Опорных Векторов	LinearSVC	0.980000	0.980000	0.980000	0.982206	98.000000
2	Логистическая регрессия	LogisticRegression	0.890000	0.870000	0.840000	0.874474	87.000000
3	Мешок слов	Bag of Words	0.810000	0.820000	0.820000	0.826917	83.000000
4	Term Frequency-Inverse Document Frequency	TFIDF	0.800000	0.900000	0.850000	0.896797	90.000000
5	Мешок слов + стохастический градиентный спуск	SVM_Bag of Words	0.810000	0.820000	0.810000	0.819799	82.000000
6	Term Frequency-Inverse Document Frequency + стохастический градиентный спуск	SVM_TFIDF	0.910000	0.900000	0.850000	0.897121	90.000000
7	Дерево решений	Decision Tree Classifier	0.820000	0.810000	0.810000	0.807506	81.000000
8	Случайный лес	Random Forest Classifier	0.870000	0.830000	0.850000	0.830476	83.000000

Таблица 24. Результаты работы построенных моделей

- Лучший результат показал Метод Опорных Векторов(LinearSVC): **98%** точности;

- Самый низкий результат показал Алгоритм наивного байесовского классификатора (метод Бернулли BernoulliNB): **80%** точности.

3.1.3. Анализ слов при помощи Word2Vec.

Данная совокупность моделей на основе нейронных сетей функционирует, принимая на вход большой набор текстовых данных и сопоставляя каждому слову в нём вектор; тем самым в качестве вывода выдавая координаты слов. Первым делом генерируется словарь текстового корпуса, затем осуществляется машинное обучение с процессом вычисления векторного представления слов (word embedding). Оно базируется на контекстной близости, поскольку слова, которые встречаются рядом в тексте с другими конкретными словами (с похожим значением), по логике модели могут иметь близкие по косинусному расстоянию векторы.

Аргументы модели Word2Vec:

- min_count — нижний порог частоты встречаемости слова;
- window — размер контекстного окна;
- size — размер векторного представления слова.
- negative — количество слов вне контекста для учёта в обучении.
- alpha — начальное значение, которое используется в алгоритме обратного распространения ошибки (Backpropogation).
- min_alpha — нижняя граница learning_rate, на которое может опуститься модель в процессе обучения.
- sg —1 для Skip-gram; 0 для CBOW.

Главные алгоритмы, на основе которых реализуется обучение модели Word2Vec, это Continuous BoW (непрерывный мешок слов) и Skip-gram. CBoW осуществляет предсказания текущего слова на основе его контекста. Skip-gram, в свою очередь, использует текущее слово с целью предугадать окружающие его слова. Порядок слов 2нтекста не оказывает влияния на результат ни в одном из этих алгоритмов.

```
w2v_model = Word2Vec(
    min_count=10,
    window=2,
    size=300,
    negative=10,
    alpha=0.03,
    min_alpha=0.0007,
    sample=6e-5,
    sg=1)
```

Таблица 25. Значения аргументов в модели W2V

После обучения модели на 30 эпохах стало возможным оценить результаты. Так как каждое слово было представлно вектором, слова (векторы) можно сравнить друг с другом. Для сравнения в библиотеке Gensim используется косинусный коэффициент (Cosine similarity).

У модели Word2vec в качестве атрибута существует объект wv, который и содержит векторное представление слов. У этого объекта есть методы для получения мер схожестей слов. Для примера можно вызвать список слов, которые по контексту ближе всего к слову “сирия”:

```
1   w2v_model.wv.most_similar(positive=["сирия"])
```

```
[('сирийский', 0.5738059878349304),
 ('ирак', 0.5021064281463623),
 ('асад', 0.4718540906906128),
 ('боевик', 0.4643007516860962),
 ('дамаск', 0.4640951156616211),
 ('сша', 0.4587487280368805),
 ('башар', 0.45228761434555054),
 ('страна', 0.4428333044052124),
 ('маяанна', 0.4293001890182495),
 ('оппозиция', 0.4250986576080322)]
```

Таблица 26. Схожие по контексту слова для «сирия»

Чем больше косинусный коэффициент, тем выше значение контекстной близости слов. Можно заметить, что образованное от слова "сирия" прилагательное "сирийский" – самое похожее на исходное слово (по факту – прилагательное, образованное от данного существительного). Также среди похожих слов нашлись соседствующий с Сирией Ирак, имя/фамилия президента страны, а также города/районы данного государства.

```
1   w2v_model.wv.most_similar(positive=["страна"])[:3]
```

```
[('россия', 0.5004262328147888),
 ('сша', 0.4614945650100708),
 ('сирия', 0.44504767656326294)]
```

Таблица 27. Схожие по контексту слова для «страна»

Существует возможность вывести и самое близкое по контексту слово из заданного списка к другой заданной текстовой единице. Это осуществляется при помощи метода most_similar_to_given:

```
1   w2v_model.wv.most_similar_to_given("сирия", ["война", "бумага", "правда"])
```

```
'война'
```

Таблица 28. Выбор самого похожего слова на «сирия» из списка других слов

Слово “сирия” из всего списка наиболее близко к слову “война” (так как контекст датасета – политические комментарии пользователей по поводу военных действий в Сирии (2016 год).

Получаем косинусный коэффициент, применив функцию similarity:

```
1   w2v_model.wv.similarity("башар", "асад")
```

```
0.8433391
```

```
1   w2v_model.wv.similarity("террорист", "боевик")
```

```
0.67645717
```

Таблица 29. Работа similarity

Далее рассмотрим на диаграмме контекста слов, насколько далеко располагается «война» от слов «голод», «смерть», «победа»:

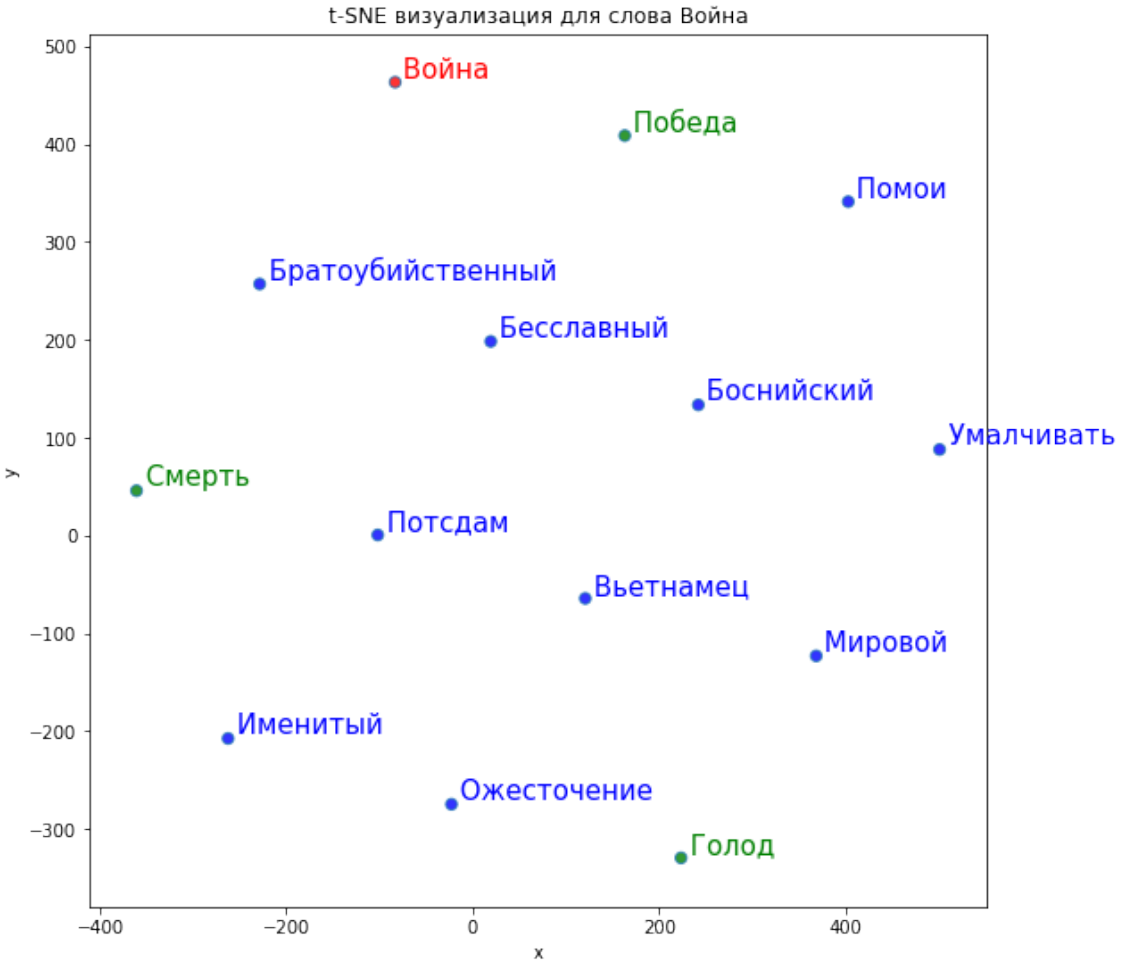


Диаграмма 13. Близость слов по контексту

Выбор лишнего по контексту слова в списке осуществляется при помощи атрибута doesnt_match:

```
1   w2v_model.wv.doesnt_match("россия сша европа сирия боязнь".split())
```

```
/usr/local/lib/python3.7/dist-packages/gensim/models/keyedvectors.py:8
  vectors = vstack(self.word_vec(word, use_norm=True) for word in usec
'боязнь'
```

Таблица 30. Выбор контекстно-лишнего слова

Размерность словаря, созданного на основе датасета:

```
1   w2v_model.wv.syn0.shape
```

```
/usr/local/lib/python3.7/dist-packages/ipykernel_
  """Entry point for launching an IPython kernel.
(15896, 300)
```

Таблица 31. Размерность словаря

Полученные векторные представления слов позволили вычислить «семантическое расстояние» между словами естественного языка. Так были найдены похожие по значению слова и дана оценка их похожести – на её основе можно оценить и саму работу модели как удовлетворительную. Word2vec осуществил прогноз на основании контекстной близости данных слов. Поскольку данный метод основан на обучении простой искусственной нейронной сети, то для более эффективной его работы в процессе обучения необходимо использовать более крупные корпусы текстовых данных. Используя модель на нашем довольно обширном датасете, можно быть уверенными в высоком качестве предсказаний.

3.2. Выводы и результаты

Отсортированная по убыванию относительно метрики accuracy таблица с результатами работы построенных моделей:

Полное название модели	Рабочее название модели	Точность (precision)	Полнота (recall)	F-мера (f1-score)	Общая точность предсказания (accuracy)	Общая точность предсказания (accuracy)[%, округлённая]
Метод Опорных Векторов	LinearSVC	0.98	0.98	0.98	0.982206	98.0
Term Frequency-Inverse Document Frequency + ст...	SGD-TFIDF	0.91	0.90	0.85	0.897121	90.0
Term Frequency-Inverse Document Frequency	TFIDF	0.80	0.90	0.85	0.896797	90.0
Логистическая регрессия	LogisticRegression	0.89	0.87	0.84	0.874474	87.0
Случайный лес	Random Forest Classifier	0.87	0.83	0.85	0.830476	83.0
Мешок слов	Bag of Words (BoW)	0.81	0.82	0.82	0.826917	83.0
Мешок слов + стохастический градиентный спуск	SGD-BoW	0.81	0.82	0.81	0.819799	82.0
Дерево решений	Decision Tree Classifier	0.82	0.81	0.81	0.807506	81.0
Алгоритм наивного байесовского классификатора:...	BernoulliNB	0.80	0.80	0.71	0.799418	80.0

Таблица 32. Итоговая таблица с метриками моделей

В принципе все методы показали хорошие результаты - процент точности варьируется от 80 до 98, поэтому ни одну из моделей нельзя назвать неудачной. Тем не менее, каждая из них имеет свои преимущества и недостатки. Оптимальной моделью для решения задач анализа тональности текстов был выбран Метод Опорных Векторов.

Также стоит отметить, что в процессе работы и подробного изучения темы анализа тональности текстов было выявлено два основных типа моделей для решения задач подобного типа. Во-первых, это модели, которые предсказывают оценку эмоциональной окраски в рамках исходного набора данных – они более точные, но в то же время ограничиваются своей предметной областью. Во-вторых, существуют модели, базирующиеся на нейронных сетях – такие как полноформатный Word2Vec, способный давать непосредственную оценку тональности тем текстовым единицам, которые находятся вне исходного набора данных. Как правило, такие модели обучаются на корпусах текстов определённого языка, а их работа по итогам моделирования на основе ограниченных предметной областью текстовых данных не может расцениваться как корректная.

ЗАКЛЮЧЕНИЕ

Несмотря на масштабные и быстрорастущие темпы научно-технического прогресса в IT-сфере, на сегодняшний день нет какого-то определённого и самого близкого к идеалу алгоритма анализа тональности текстов. Многие аспекты упираются в границы предметных областей, которые не так просто стереть, если речь идёт об автоматизации задач подобного типа. Их реализация внутри социальных сетей, где миллионами пользователей ведутся активные дискуссии на всевозможные темы, представляется крайне трудной задачей. Невозможно принять во внимание все существующие прогнозы событий, когда речь идёт о коммуникации людей.

Именно поэтому методы фильтрации негативных мнений если и вводятся, то лишь на экспериментальном уровне, а для процесса модерирования всё ещё требуется контроль человека, который выполняет анализ и наблюдение в ручном режиме. С другой стороны, классификация тональности требует тщательного подбора технологий нормализации текстовых данных, прежде чем начнётся процесс машинного обучения. По этой причине данной работе были исследованы и реализованы операции из сферы Natural Language Processing (обработки естественного языка): удаление стоп-слов, токенизация, нормализация, стемминг и лемматизация. Далее были задействованы 9 различных моделей машинного обучения, в которых на исходных данных компьютер обучился распознавать негативные и позитивные комментарии – оценка точности работы этих моделей варьировалась от 80 до 98%, что говорило об успешных результатах обучения.

По результатам апробации созданных моделей были отобраны самые оптимальные в контексте решаемой задачи: Метод Опорных Векторов, который показал наилучший результат, а также модели TF-IDF (стандартная и оснащённая стохастическим градиентным спуском). Метрика точности этих моделей начиналась с отметки в 90%.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ И ИНТЕРНЕТ-РЕСУРСОВ

● Интернет-ресурсы:
1 http://www.linis-crowd.org/

2 https://proglib.io/p/lyublyu-i-nenavizhu-analiz-emocionalnoy-okraski-teksta-s-pomoshchyu-python-2020-11-13

3 https://python-school.ru/blog/nlp-text-preprocessing/

4 https://www.kaggle.com/stoicstatic/twitter-sentiment-analysis-for-beginners/notebook

5 https://scikit-learn.ru/1-9-naive-bayes/

6 https://scikit-learn.ru/1-4-support-vector-machines/

● Основная литература:

7 М.В. Коротеев. Об основных задачах дескриптивного анализа данных.

8 М.В. Коротеев. Учебное пособие по дисциплине “Анализ данных и машинное обучение” - 2018.

● Дополнительная литература:

9 L.P. Coelho, W. Richert. Building machine learning systems with Python - 2015

10 W. McKiney. Pandas: powerful Python data analysis toolkit - 2016

11 Классический курс по машинному обучению

12 Выполненные лабораторные работы по предмету

13 Материалы на «GitHub» и «Stack Overflow»