ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО
ОБРАЗОВАНИЯ
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ
«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»

Московский институт электроники и математики им. А.Н. Тихонова

Пехтерев Денис Олегович, группа МСКМ-181

АНАЛИЗ ТОНАЛЬНОСТИ ТЕКСТА МЕТОДАМИ МАШИННОГО ОБУЧЕНИЯ

Междисциплинарная курсовая работа
по направлению 01.04.04 Прикладная математика

студента образовательной программы магистратуры
«Суперкомпьютерное моделирование в науке и инженерии»

Студент _____ Пехтерев Д.О.

Научный руководитель
к.ф-м.н., доцент Артамонов С.Ю

Москва 2019

Содержание

Аннотация

Было проведено исследование предыдущих работ на тему анализ тональности методами машинного обучения, для определения того, какие методы являются современными, какие их особенности уже были выявлены, и для определения вектора развития данной области ([3] - [9]).

В расчётной составляющей произведено изучение алгоритма по анализу тональности текстовых данных, где был рассмотрен и использован метод регуляризованной логистической регрессии. Для выделения признаков были рассмотрены три различных метода: Мешок слов (от англ. BoW), TF-IDF и N-граммы, среди которых был выбран TF-IDF. При этом был написан алгоритм, который из заранее заданного диапазона коэффициентов регуляризации подбирает и использует для классификации параметр, при котором модель имеет максимальное качество.

Кроме этого, были визуализированы результаты работы логистической регрессии, связанные с весами коэффициентов: было построено облако слов, анализируя которое можно выявить какие слова ассоциируются с положительными эмоциями у потребителей, а какие с отрицательными. Из-за простой формы подачи материалы, таким анализом смогут пользоваться люди, не имеющих специальных знаний в машинному обучении, что является положительным фактором внедрения такой модели в компании.

1. Введение

Мы живём в то время, когда интернет есть в каждом нашем устройстве, независимо от того, где мы находимся и чем занимаемся. Что касается России, то к началу 2019 года количество интернет-пользователей среди населения 16+ стало равно 90 миллионов человек, что есть 75,4% взрослого населения страны [1]. Проникая во всё более новые отрасли нашей жизни, интернет стал наполняться огромным количеством информации, генерируемой потребителями продуктов. Стоит заметить, что почти у каждого конкурентноспособного бренда есть как минимум свой сайт или форум, поэтому клиенты имеют возможность расписать свой личный опыт использования того или иного продукта. Если смотреть на этот процесс со стороны успешного функционирования бизнеса, то одна из его важнейших частей – понимание клиента и его потребностей. Приблизительно 95% потребителей в возрасте от 18 до 34 лет читают отзывы на продукт или услугу перед покупкой, из них 91% доверяют онлайн отзывам так же, как и личным рекомендациям [2].

1.1. Поставновка цели и задачи

Цель работы: применение современных средств прикладной математики для анализа данных.

Задача работы: анализ социальной активности потребителей в интернете, касающейся тональности оставленных ими отзывов.

Основная задача: анализ тональности отзывов методами машинного обучения.

Подходы: машинное обучение.

Используемый метод: метод регуляризованной логистической регрессии.

1.2. Обсуждение постановки задачи

Обычно перед компанией стоит задача по мониторингу реакции клиента на:

- Выпуски новых продуктов (запуск новых сервисов);
- Внесение изменений в существующие продукты (услуги);
- Внедрённые или готовящихся к внедрению инициатив, связанных с обслуживанием клиентов;

- Запуск новых рекламных компаний.

Наиболее же важным для компаний является определение того, какие их действия вызывают положительный отклик у клиентов, а какие отрицательный.

Зачастую, количество отзывов по конкретным продуктам (услугам) может превышать 10000 записей в день. Данная цифра включает в себя также и сообщения, поступающие в службу поддержки, которые необходимо сразу же сортировать и классифицировать, ведь от этого зависит опыт использования продукта клиентом и его мнение о том, на сколько дорог он как клиент для компании. Одной из задач, которая стоит перед компаниями сейчас – это задача по определению тональности текста (в пер. с англ. sentiment analysis).

Сейчас решением данной задачи на Российском рынке занимаются два крупных сервиса: YouScan и Brand Analytics. С одной стороны, эти системы кажутся абсолютно схожими – они решают одинаковые задачи и дают примерно одинаковый анализ. Главной же проблемой данных решений является их цена и ограниченный функционал. Рассмотрим ценообразование, которое используется сервисами YouScan и Brand Analytics по минимальной стоимости:

- Standard , YouScan – 35 000 руб / мес:
  - за каждые 5 тем в месяц;
  - 100 000 упоминаний в каждой теме.
- Стартовый плюс , Brand Analytics – 35 000 руб / мес:
  - сообщений в аккаунте за месяц: 20 000 сообщений;
  - количество тем: 5;
  - автоматическая разметка тематик (тегов на тему): 30[1].

Как мы видим, сервисы по своей сути предлагают одинаковый пул услуг: ограниченный набор тем (вы можете отслеживать только 5 запросов) и имеют ограничение на количество упоминаний (другими словами, за эти деньги больше указанного выше лимита вы увидеть не сможете). Большим и достаточно развитым бизнесам не составит труда купить подписку подороже и отслеживать свои товары на ежедневной основе,

[1]например, бывают тэги: головная боль, кожный зуд и т.д.

получая при этом достаточно хороший сервис. Однако здесь оказывается незанятой другая ниша рынка – компании, имеющие достаточно много продуктов (брендов 50), но не выделяющих большие бюджеты на анализ отзывов, оставляемых относительно их продуктов. Такой анализ был бы полезен продакт-менеджерам [2] для того, чтобы отслеживать реакцию потребителей и смотреть на то, как они воспринимают/потребляют продукт, и с какими сложностями они сталкиваются.

К таким компаниям относятся:

- Фармацевтические компании;
- Медицинские клиники;
- Маркетплейсы[3].

Спрос на сервисы, связанные с решением задачи по определению тональности также есть у компаний, обладающих развитым каналом обратной связи. Например, когда клиент может сразу написать отзыв об указанной услуге/товаре и не поставить при этом оценку.

2. Обзор литературы

В этой части работы будут проанализированы статьи других авторов, напрямую связанные с анализом тональности текста и анализом текста, содержащегося в отзывах о покупках на различных сайтах.

В обзорной статье [3] авторы изучили основные алгоритмы как по выделению признаков, так и по построению моделей, способных определять тональность текста. Говоря об анализе тональности, авторы обращаются к рисунку 1, который описывает существующие и использующиеся сейчас техники по анализу тональности текста.

Также авторы рассмотрели основные методы по извлечению признаков из текста:

1. Методы, в основе которых лежит измерение корреляции между словами и классами:

[2] Люди, возглавляющие все активности, связанные с продуктом (маркетологи)

[3] На англ. online marketplace, что в переводе на русский язык есть онлайн-магазин электронной торговли

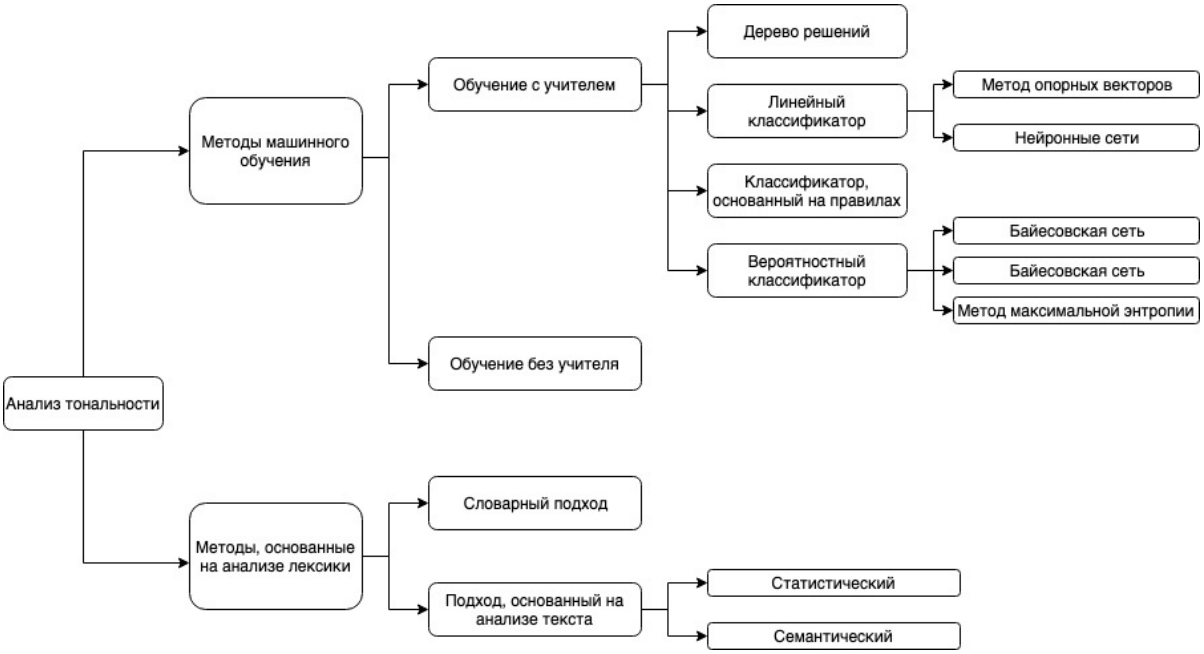

Рис. 1. Техники по анализу тональности текста (на основе оригинального рисунка 2 из статьи [3])

- Метод поточечной взаимной информации (от англ. Point-wise Mutual Information (PMI)): метод основан на введении и использовании меры, которая показывает контекстную связь между признаками и классами. Пример использования: метод использовался в работе Yu и Wu [4], где авторы при помощи данного метода расширяли первоначальный набор слов, который был собран из небольшого набора текстов, посвященных новостям с фондового рынка. Результаты их модели превзошли результаты других разновидностей моделей PMI;

- Хи-квадрат (от англ. Chi-square (χ^2)): метод также отвечает за поиск величины связи конкретного слова и класса, однако лучше PMI в том, что внутри него используются нормализованные значения, соответственно, эти значения больше подходят к терминам в той же категории. Пример использования: в результате исследования Фаном и Чангом [5] контекстной рекламы, ими было показано, что их метод может выделить рекламные объявления, которые непосредственно коррелируют с интересами блогера. Для выделения признаков в работе использовался метод хи-квадрат, тогда как для модели классификации использовался метод опорных векторов, изображённый на рисунке 1.

2. Методы, суть которых заключается в уменьшении размерности данных:
   - Латентно-семантический анализ (от англ. Latent Semantic Indexing (LSI)).
3. Методы, основанные на статистическом подходе:
   - Hidden Markov Model (HMM);
   - Latent Dirichlet Allocation (LDA).
4. Другие методы:
   - Метод Мешка слов (от англ. Bag of Words (BoW)).

Авторы в [6] провели исследование по анализу тональности наборов данных, в состав которых входили: Cornell Movie Dataset, Twitter Dataset, Amazon Products Dataset и IMDB Dataset, используя различные методы машинного обучения. В работе рассматривались следующие методы: Naive Bayes (NB), стохастический градиентный спуск (Stochastic gradient descent - SGD), метод опорных векторов (support vector machine - SVM), пассивный агрессивный метод (Passive Aggressive - PA), максимальной энтропии (Maximum Entropy), Adaptive Boosting (AdaBoost), Multinomial Naive Bayes (MNB), Bernoulli Naive Bayes (BNB), Ridge Regression (RR) и Logistic Regression (LR). В результате использования данных методов вместе с методами по извлечению признаков из наборов данных: TF-IDF и N-Грамм (юниграмма, биграмма и триграмма), было получено, что использование алгоритмов PA и RR по сравнению с остальными методами дают одни из самых высоких результатов классификации: от 87% до 99.96%, полученный при использовании метода PA. Также было выделено, что метод логистической регрессии и метод опорных векторов также давали приемлемые результаты: 87.56% и 85.76% соответственно.

Авторы в [7] проводят анализ тональности текста (бинарная классификация), используя набор данных, состоящий из отзывов, оставленных пользователями на сайте Amazon об электрических товарах (Kindle, DVD и других) в период с февраля 2012 года по июль 2017 года.

Для извлечения признаков из данных использовался метод TF-IDF, а для обучения модели были рассмотрены методы машинного обучения: логистическая регрессия и 3 разновидности Naive Bayes классификатора:

- Обычный метод Naive Bayes;
- Multinomial Naive Bayes;
- Bernoulli Naive Bayes.

В итоге были получены следующие результаты: Multinomial Naive Bayes - 92.87%, Bernoulli Naive Bayes - 92.35% и Логистическая регрессия - 93.34%.

Те же самые авторы в [8] изучали применение методов по анализу тональности текста, используя библиотеки языка программирования R по машинному обучению применительно к отзывам покупателей о приобретённой еде на сайте Amazon. В работе описан алгоритм, при помощи которого проводилась классификация отзывов, а также приведены облаки слов, которые имели больший вес в модели при определении тональности.

Ещё одна работа по анализу тональности отзывов с сайта Amazon является работа [9], где авторы предлагают алгоритм по обработке отзывов, призванный разбить оставленные пользователями отзывы на две категории:

- Отзывы о представленном сервисе;
- Отзывы о характеристиках продукта.

Далее авторами используется собственный алгоритм, который проводит анализ отзывов на предмет того, является ли отзыв прямым или косвенным. Например: “Камера не лучше, чем у iPhone 4s” - косвенный отзыв. Такие фразы из предложений сравниваются с выделенными вручную фразами, по итогу чего и происходит классификация каждого отзыва. Также интересна обработка отрицаний в статье авторов, когда алгоритм встречает в отзыве фразу “не хороший товар”, то сначала алгоритм анализирует тональность слова хороший (стоит также отметить, что он ищет следующую связку: частицу, обозначающую отрицание и прилагательное). После чего, при помощи словаря тональности определяется тональность прилагательного и, так как ещё есть частица, означающая отрицание, то данной фразе присваивается противоположная тональность.

Алгоритм работы с отзывами, после извлечения набора фраза из предложений, может быть представлен следующим образом:

- Все фразы проверяются на то, являются ли они косвенными или прямыми относительно продукта;
- Проверяются на наличие с частицей “не”.

3. Этапы работы по определению тональности

3.1. Нормализация и токенизация текста

Нормализация текста подразумевает приведение всего текста к одному виду, её также можно разделить на следующие шаги:

- Удаление знаков препинания;
- Удаление лишних пробелов;
- Приведение всех слов к нижнему регистру.

Токенизация - разделение всех предложений в наборе данных на слова. В английском языке, также как и в русском это происходит при помощи разделения предложений по пробелам.

3.2. Обработка слов в тексте

Под обработкой слов в тексте понимается первоначальная обработка слов с целью их приведения к базовой форме.

Для этого принято использовать два метода:

- Лемматизация (на англ. Lemmatization);
- Стемминг (на англ. Stemming).

Целью двух алгоритмов является приведение слов к их первоначальной форме. Это делается для того, чтобы в дальнейшем при векторизации текста, одни и те же слова кодировались одинаково.

Например:

- Слова "на фрукте, фруктовый, у фруктов, фрукты"будут приведены к одинаковому виду: "фрукт";

- Предложение, как "я иду на работу с утра"будет приведено к виду: "я идти работа с утро".

Стемминг - процесс, заключающийся в отбрасывании от слова приставок и суффиксов, с целью как можно точнее выделить основу слова. Стоит отметить, что такая основа слова не всегда будет совпадать с общепринятым корнем слова.

Например, для русского языка, слово "играл" будет приведено к виду: "игра" , также как и для английского, слово: "played"будет приведено к виду "play".

Одним из самых знаменитых стеммеров[4] для английского языка является стеммер разработанный Мартином Портером [10]. Алгоритм состоит из 5 этапов обработки слов, которые применяются друг за другом. Суть алгоритма заключается в том, что при обработке каждого слова смотрится его суффикс, который сравнивается с заранее заданными в базе суффиксов: если находится совпадение, то суффикс отбрасывается.

Преимуществом является то, что не нужно держать базу основ всех существующих в английском языке слов.

Недостатком же является то, что алгоритм может обрезать часть слова, которая несёт смысл, например: "кроватью"будет преобразовано в "крова".

Лемматизация - процесс по обработке слов в тексте, суть которого заключается в том, чтобы использовать словарные формы слов (леммы), удалив при этом лишние приставки и суффиксы [11].

Отличием лемматизации от стемминга является то, что процесс лемматизации полностью выстроен на поиске начальной (словарной) формы слова, что может достигаться за счёт поиска части речи и других преобразований слов.

Помимо широкого использования вышеперечисленных алгоритмов в задачах по машинному обучению, они также применяются в большом количестве других областей:

- Системы индексации в поисковых системах. Индексация в поисковых системах - процесс обработки сайта (сведений с сайта) и добавление этих сведений роботом в базу данных. Делается это с целью оптимизации поиска информации;

- SEO[5] - поисковое продвижение сайта, которое включает в себя по-

[4]Конкретная реализация метода Стемминг

[5]Search Engine Optimization, поисковая оптимизация)

вышение соответствия страниц поисковому запросу, оптимизации содержания и т.д.;

- Поисковые системы - веб-сервисы, созданные для поиска информации в интернете.

3.3. Векторизация отзывов

При анализе тех или иных текстов методами машинного обучения, текстовые данные требуется перевести в числовой формат. Для этого существует определённый набор методик, таких как:

- Метод "Мешок слов"(BoW, bag-of-words) [12];

  Метод переводит набор данных в матричный формат, где каждая строка является отдельным предложением или отдельным отзывом, тогда как столбец отвечает за 1 слово в этом предложении/отзыве. Элементами же матрицы являются частоты, с которыми данные слова встречаются в предложении. Всего же формируется количество признаков равное количеству уникальных слов в наборе данных.

  Рассмотрим, например, следующие предложения: "Эта была отличная затея"и "Эта затея стоила сил". Разбивая на слова, получаем: 'эта', 'была', 'отличная', 'затея' и 'эта', 'затея', 'стоила', 'сил'. Уникальными словами здесь являются: 'эта', 'была', 'отличная', 'затея', 'стоила', 'сил' - количество признаков равно 6.

  Применяя метод "Мешок слов"получаем два вектора:

  – $[1, 1, 1, 1, 0, 0]$ - для предложения "Эта была отличная затея"
  – $[1, 0, 0, 1, 1, 1]$ - для предложения "Эта затея стоила сил"

- Метод TF-IDF (TF — term frequency, IDF — inverse document frequency).

  Метод, суть которого состоит в оценке важности слова как в контексте рассматриваемого экземпляра документа, так и относительно других документов, находящихся в коллекции. Зачастую метод TF-IDF также используется для оценки схожести текстов [13];

  Алгоритм помогает избавиться от предлогов, союзов и других частей речи, которые часто встречаются в любом тексте. Например, в русском языке: "это" ,"и" , "а" , "или" , а в английском языке: "is" , "a" , "the" и др.

Частота слов (TF) вычисляется как отношение количества вхождений конкретного слова в документе к общему количеству слов в этом документе и записывается в следующем виде:

$$tf(t,d) = \frac{n_t}{\sum_k (n_k)}, \tag{1}$$

где n_t - количество вхождений слова t в документ n, $\sum_k (n_k)$ - общее число слов в документе n.

Обратная частота документа (IDF) - отношение общего количества документов в коллекции к числу документов, где присутствует данное слово и записывается в виде:

$$idf(t) = \ln \frac{1+n}{1+df(t)} + 1, \tag{2}$$

где n отвечает за общее число документов, входящих в набор данных, тогда как $df(t)$ - количество документов, которые содержат слово t.

В конечном итоге, метод принимает следующий вид [14]:

$$tf - idf(t,d) = tf(t,d) \times idf(t). \tag{3}$$

Также в большинстве случаев, после того как выполнено преобразование TF-IDF принято полученные вектора нормализовать, используя Евклидову норму, имеющую вид:

$$v_{norm} = \frac{v}{||v||_2} = \frac{v}{\sqrt{{v_1}^2 + {v_2}^2 + \cdots + {v_n}^2}}. \tag{4}$$

Теперь рассмотрим пример применения данного метода на следующих предложениях:

– "Тебя знаю точно" ;

– "Знаю ли тебя" ;

– "Знаю точно себя" .

Словарь уникальных слов будет состоять из 5 слов и иметь следующий вид: ’знаю’, ’ли’, ’себя’, ’тебя’, ’точно’.

Рассмотрим первое предложение: "Тебя знаю точно" .

Начнём со слова 'тебя': $n = 3; df(t)_{term1} = 2;$

$$\text{idf}(t)_{term1} = \ln \frac{1+n}{1+df(t)} + 1 = \ln(4/3) + 1 = 1.2877. \quad (5)$$

После нормализации Евклидовой нормой, величина будет иметь вид:

$$\frac{(\ln(4/3)+1)}{((\ln(1)+1)^2 + (\ln(4/3)+1)^2 + (\ln(4/3)+1)^2)^{(1/2)}} = 0.6198. \quad (6)$$

Итоговое значение слова 'тебя' будет равно значению слова 'точно', так как встречается два раза в наборе данных и является одним из трёх слов в каждом предложении.

По аналогии, подставляя корректные значения в формулы выше, получаем следующие значения, которые удобно записываются в виде матрицы. При расположении слов в таком порядке: 'знаю', 'ли', 'себя', 'тебя', 'точно', для трёх предложений выше получается следующая матрица:

$$\begin{bmatrix} 0.48133417 & 0 & 0 & 0.61980538 & 0.61980538 \\ 0.42544054 & 0.72033345 & 0 & 0.54783215 & 0 \\ 0.42544054 & 0 & 0.72033345 & 0 & 0.54783215 \end{bmatrix}$$

- N-граммы (N-grams).

  Метод, основанный на методе Мешка слов, однако отличается тем, что данный метод позволяет учесть порядок слов в предложении, что является полезным свойством, особенно при анализе таких предложений как: "Продукт плох, совсем не качественный"или "Продукт качественный, совсем не плохой" [15]. Данные предложения будут иметь одинаковое числовое представление при их обработке методом Мешка слов. Для такого рода ситуаций подходит метод n-граммы, так как позволяет захватить контекст использования данного слова, учитывая не только количество отдельных слов (токенов), но и пар слов или словосочетаний состоящих из 3-ёх слов.

3.4. Логистическая регрессия

Для решения данной задачи бинарной классификации будет использоваться логистическая регрессия, которая является одним из методов

построения линейного классификатора и даёт оценивать апостериорные вероятности принадлежности объектов классам.

Логистическая регрессия строит регрессионную модель, которая может предсказывать вероятности того, что запись относится к классу 1. Как линейная регрессия использует линейную функцию, так логистическая регрессия основана на функции вида сигмоида, которая записывается следующей формулой:

$$\sigma(z) = \frac{1}{1 + e^{(-x)}}. \tag{7}$$

И имеет вид, изображённый на рисунке 2:

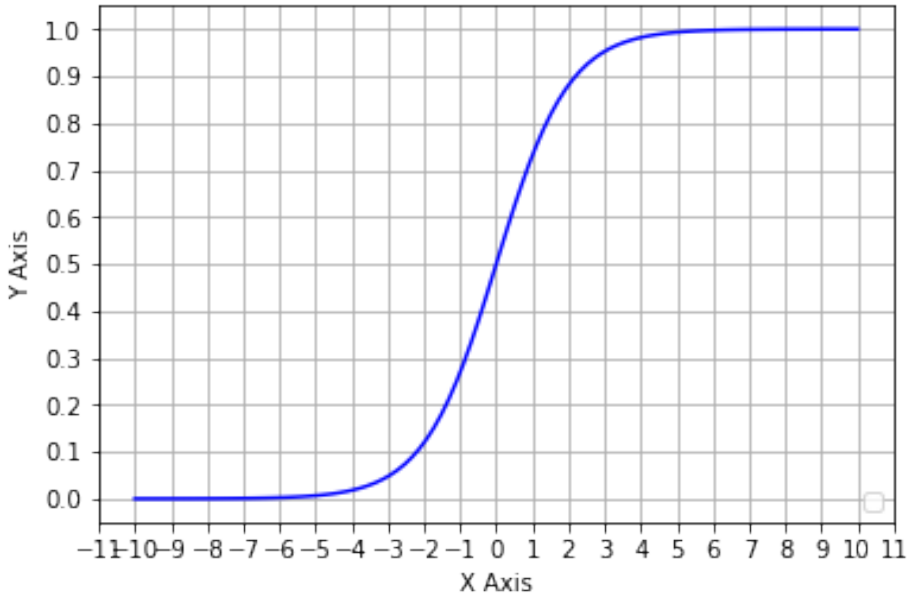

Рис. 2. Логистическая функция (сигмоида).

Моделью классификации логистическая регрессия становится только после введения порогового значения. На самом деле, подбор правильного порогового значения зависит от значений метрик: точность и полнота.

В зависимости от количества категорий, которые необходимо классифицировать, логистическая регрессия может иметь вид:

- Биноминальная: целевая переменная принимает только два значения, например: "да-нет" , "выиграет-проиграет" ;
- Мультиномиальная: целевая переменная принимает больше двух значений, например: категории товаров - "категория А" ,"категория Б" ,"категория В" ;

- Порядковая: целевая переменная принимает значения, принадлежащие порядковой шкале, например: "неудовлетворительно" , "удовлетворительно" ,"хорошо" ,"отлично" .

Мною будет рассмотрена именно биноминальная логистическая регрессия. Задача, решаемая данным типом логистической регресии, может быть записана в виде:

$$y = \begin{cases} 1, & \text{положительная тональность} \\ -1, & \text{отрицательная тональность} \end{cases}$$

Имеем обучающую выборку вида: $X^m = (x_i, y_i)_{i=1}^m$, где $x_i \in R^n$. В логистической регрессии также используется линейный алгоритм классификации, который имеет вид:

$$a(x, w) = sign(\sum_{j=1}^{n} w_j * f_j(x) - w_0) = sign\langle x, w\rangle, \tag{8}$$

где:

- w_j - вес j-го признака;
- w_0 - порог принятия решения;
- $w = (w_0, w_1, ..., w_n)$ - вектор весов;
- $\langle x, w\rangle$ - скалярное произведение имеющихся признаков на вектор весов.

Заметим, что на данном этапе искусственно введён нулевой признак: $f_0(x) = -1$ [16].

Задачей обучения является настройка весов по имеющимся данным. В логистической регрессии для этой цели решается задача минимизации эмпирического риска, который имеет следующую функцию потерь:

$$Q(w) = \sum_{i=1}^{m} ln(1 + exp(-y_i * \langle x_i, w\rangle)) \to \min_w. \tag{9}$$

После того, как вектор весов w найден, модель готова к вычислению классификации для конкретного объекта x. Кроме того, для объекта x возможно оценивать вероятности принадлежности этого класса к конкретным классам, используя формулу вида:

$$P(y|x) = \frac{1}{1 + e^{(-y*\langle x,w\rangle)}}. \tag{10}$$

Для решения данной задачи я использовал пакет sklearn, в котором содержится функция LogisticRegression [17].

Мною использовалась штрафная функция типа $L2$, так как она позволяет избавиться от проблем мультиколлинеарности. При этом, в пакете существует три варианта регуляризации логистической регрессии [18]:

- L1 регуляризация имеет эффект отбора признаков, так как обнуляет веса неинформативных признаков:

$$\min_{w,c} \|w\|_1 + C \sum_{i=1}^{n} \log(\exp(-y_i(X_i^T w + c)) + 1), \tag{11}$$

  где:

  - $\|w\|_1 = \sum_{i=0}^{n} |w_i|$;
  - X_i^T - транспонированная матрица признаков;
  - C - коэффициент регуляризации: чем меньше значение, тем сильнее регуляризация.

- L2 регуляризация отвечает за решение проблему мультиколлинеарности, сокращает веса линейно зависимых признаков:

$$\min_{w,c} \frac{1}{2} w^T w + C \sum_{i=1}^{n} \log(\exp(-y_i(X_i^T w + c)) + 1). \tag{12}$$

- Elastic-Net регуляризация является комбинацией двух регуляризаций (L1 и L2):

$$\min_{w,c} \frac{1-\rho}{2} w^T w + \rho \|w\|_1 + C \sum_{i=1}^{n} \log(\exp(-y_i(X_i^T w + c)) + 1), \tag{13}$$

  где коэффициент p отвечает за силу l_1 регуляризации и l_2 регуляризации.

3.5. Оценка точности классификации

Исследователями выделяется большое количество оценки точности классификации.

К ним относятся:

- BaseRate - базовая доля правильных ответов. Показатель демонстрирует минимум, который должна превышать метрика Accuracy.

$$BaseRate = \text{argmax}\frac{1}{l}\sum_{i=1}^{l}[y_o = y_i], \tag{14}$$

где:

  - y_o - объекты определённого класса;
  - argmax - выбирает класс с максимальным количеством элементов;
  - y_i - результат работы алгоритма.

- Accuracy - суть показателя заключается в том, что при исследовании точности модели мы смотрим на долю правильных ответов, которая может быть записана в следующем виде:

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN}, \tag{15}$$

где:

  - TP (True Positive) - количество истинно-положительных результатов, когда заданные объекты класса 1 равны результату алгоритма 1;
  - TN (True Negative) - количество истинно-отрицательных результатов, когда заданные объекты класса 0 равны результату алгоритма 0;
  - FP (False Positive) - количество ложно-положительных результатов, заданные объекты класса 0 равны результату алгоритма 1;
  - FN (False Negative) - количество ложно-отрицательных результатов, заданные объекты класса 1 равны результату алгоритма 0.

- Precision (точность алгоритма в пределах рассматриваемого класса) – это доля объектов действительно принадлежащих рассматриваемому классу относительно всех объектов, которые алгоритм отнесла к этому классу.

  Метрика Precision выражается следующей формулой:

$$Precision = \frac{TP}{TP + FP}. \tag{16}$$

- Recall (полнота алгоритма) – это доля найденных классификатором объектов, принадлежащих определённому классу относительно всех объектов этого класса.

  Метрика Recall выражается следующей формулой:

$$Recall = \frac{TP}{TP + FN}. \tag{17}$$

  Как видно из формул выше, точность и полнота характеризует разные стороны определения точности классификатора, именно поэтому на эти метрики принято смотреть в тандеме.

- Матрица ошибок (confusion matrix, матрица неточностей) - представляет из себя матрицу размером M на M, где M - отвечает за количество классов. В случае бинарной классификации матрица имеет размер 2 на 2, однако такие матрицы распространено строить, если количество классов > 2. Столбцы матрицы относятся к истинным (заданным) ответам, строки же к результатам алгоритма. Отсюда следует, что чем большие числа стоят на диагоналях данной матрицы, тем лучше работает классификатор.

- F-мера – метрика, представляющая гармоническое среднее между точностью и полнотой. Другими словами, она объединяет два показателя в один, давая исследователю общее представление о качестве работы классификатора.

  Существует два вида F-меры:

  – Если двум метрикам - полноте и точности отдаётся одинаковое предпочтение, то есть максимизируются оба этих значения, то F-мера называется сбалансированной и принимает следующий вид:

$$F = 2\frac{Precision \times Recall}{Precision + Recall}. \tag{18}$$

– Также есть возможность придать различный вес точности и полноте, тогда F-мера принимает следующий вид:

$$F = \left(\alpha^2 + 1\right) \frac{Precision \times Recall}{\alpha^2 Precision + Recall}. \tag{19}$$

, где: если мы хотим улучшить метрику точности, то нужно задавать α в диапазоне $0 < \alpha < 1$, если же мы хотим улучшить метрику полноты, то нужно задавать α в диапазоне $1 < \alpha$.

Используя сбалансированную F-меру удобно быстро принимать решения о качестве алгоритма, так как здесь учитываются две основные метрики, используемые при бинарной классификации: полнота и точность.

- ROC – кривая (Receiver Operating Characteristic, иногда называют кривой ошибок) - служит для визуализации результата работы алгоритма, тогда как для определения качества используют площадь под кривой AUC (Area Under the Curve).

  Для этого понадобятся следующие формулы:

  – FPR - доля ложно-положительных результатов относительно количества объектов, принадлежащих нулевому классу. Откладывается по оси X:

  $$FPR = \frac{FP}{FP + TN}. \tag{20}$$

  – TPR - доля истинно-положительных результатов относительно количества объектов, принадлежащих первому классу. Откладывается по оси Y:

  $$TPR = \frac{TP}{TP + FN}. \tag{21}$$

4. Описание наборов данных

В данной работе использовался следующий набор данных: отзывы на категории товаров, представленные в разных категориях на сайте

Amazon в период с мая 1996 по июль 2014 года. Набор данных был собран в файлы по категориям профессором Стэнфордского университета Джулианом Макаули [19].

Были взяты наборы данных по следующим категориям:

1. Отзывы на товары категории "Книги";

   Всего отзывов в категории: 8 898 041 записей.

2. Отзывы на товары категории "Электроника";

   Всего отзывов в категории: 1 689 188 записей.

3. Отзывы на товары категории "Кино и Телевидение";

   Всего отзывов в категории: 1 697 533 записей.

4. Отзывы на товары категории "CD-диски";

   Всего отзывов в категории: 1 097 592 записей.

5. Отзывы на товары категории "Здоровье и личная гигиена";

   Всего отзывов в категории: 346 355 записей.

Наборы данных были представлены в одинаковом формате JSON[6] и состояли из следующих колонок:

- reviewerID – уникальный идентификатор пользователя;
- asin – уникальный идентификатор продукта;
- reviewerName – имя человека, оставившего отзыв;
- helpful – оценка полезности отзыва;
- reviewText – текст отзыва;
- overall – рейтинг, оставленный пользователем (от 1 до 5);
- summary – краткий итог отзыва;
- unixReviewTime/reviewTime – время, когда отзыв был оставлен пользователем.

[6]JavaScript Object Notation - текстовый формат обмена данными, основанный на JavaScript.

Из этих столбцов для решаемой задачи будут интересны только reviewText - сам отзыв, оставленный посетителем сайта и overall - оценка пользователя, которая ставится в соответствии с его отзывом.

Итак, обработав отзывы в формате json и переведя их в формат pandas[7] - DataFrame, мы получили 5 наборов данных. Для начала будем рассматривать в каждом из этих наборов данных максимальное количество записей - 1 млн.

4.1. Обработка данных

Первым шагом в данном упражнении переведём оценки из формата 1-5 в 2 класса, которые понадобятся нам для классификации (0/1), для этого воспользуемся следующей логикой:

- при overall $\in$ (1,2) присвоим значение 0, что будет означать, что отзывы данной категории имеют отрицательную тональность;
- при overall $\in$ (3) не будем присваивать никакого значения, так как мы будем рассматривать задачу бинарной классификации;
- при overall $\in$ (4,5) присвоим значение 1, так как оставляя отзывы с такими оценками, мы можем быть уверены, что пользователь был удовлетворён своей покупкой, а значит отзыв имеет положительную тональность.

Имеем 5 наборов данных, состоящих из отзывов пользователей и их оценок. Так как в задачах машинного обучения важно иметь сбалансированные выборки, а в разных категориях товаров разное количество положительных и отрицательных отзывов, то была написана функция, которая считает в каждом наборе данных количество положительных и отрицательных отзывов, берёт наименьшее из них и обрезает класс, где отзывов больше, тем самым получается равное количество положительных и отрицательных отзывов в выборке.

[7]Pandas - библиотека языка Python для работы с наборами данных

Таблица 1. Количество отзывов по классам тональности

Название категории	Положительные	Нейтральные	Отрицательные
"Книги"	7 203 909	955 189	738 943
"Кино и Телевидение"	1 289 602	201 302	206 629
"Электроника"	1 356 067	142 257	190 864
"CD-диски"	903 002	101 824	92 766
"Здоровье и личная гигиена"	279 801	33 254	33 300

В таблице 2 представлено количество отзывов, разбитое по классам тональности.

5. Проведение эксперимента

5.1. Подготовка набора данных

Как и было описано выше, для корректной работы алгоритмов машинного обучения с текстовыми данными необходимо привести наборы данных к одному формату.

В работе использовался программный язык Python и следующие готовые библиотеки:

- json - для чтения файлов формата json;
- pandas - для удобной работы с наборами данных;
- re - для удаления лишних символов;
- nltk - для обработки текста;
- sklearn - для использования метода логистической регрессии;
- numpy - для работы с матрицами;
- matplotlib - для построения графиков.

Для этого была написана функция, которая:

- очищает данные от знаков препинания;
- приводит слова к нижнему регистру;
- формирует одинаковое количество отзывов в 2 классах.

5.2. Подбор коэффициента регуляризации C

Подбор коэффициента регуляризации C осуществлялся путём определения качества модели на тестовой выборке, что продемонстрировано на рис.3.

Из рисунка 3 можно сделать вывод, что реализованная модель выдаёт лучший результат при коэффициенте регуляризации: $C = 5$. Данное наблюдение может использоваться в дальнейшем для того, чтобы использовать модель с заранее заданным параметром.

Рисунок 4 демонстрирует зависимость времени тренировки модели от коэффициента регуляризации C. Отсюда также видно, что модель

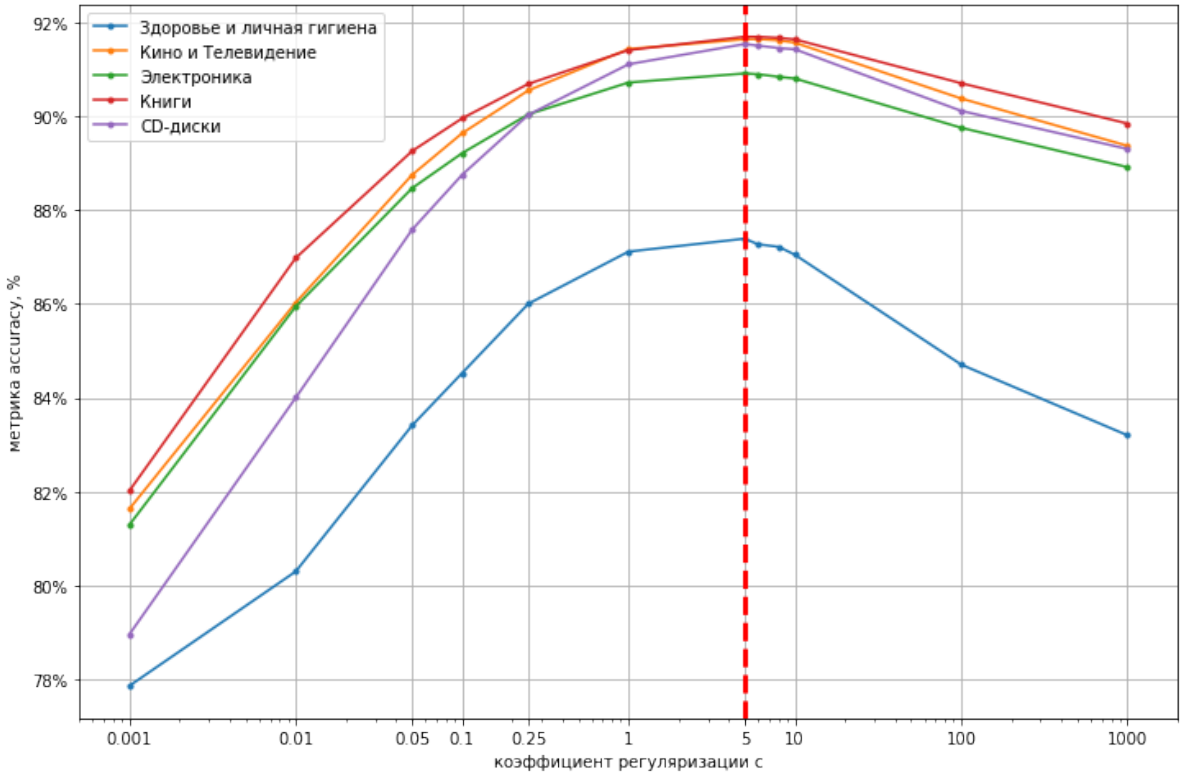

Рис. 3. Зависимость качества классификации от коэффициента C

обучается быстрее всего на категории "Здоровья и личная гигиена" , что объясняется тем, что набор данных по данной категории является самым маленьким.

5.3. Результаты

Результаты работы классификатора представлены в таблице 2.

Таблица 2. Качество работы классификатора

Название категории	Accuracy
"Книги"	91.69%
"Кино и Телевидение"	91.64%
"Электроника"	90.91%
"CD-диски"	91.54%
"Здоровье и личная гигиена"	87.39%

Здесь мы можем увидеть результаты работы классификатора, реа-

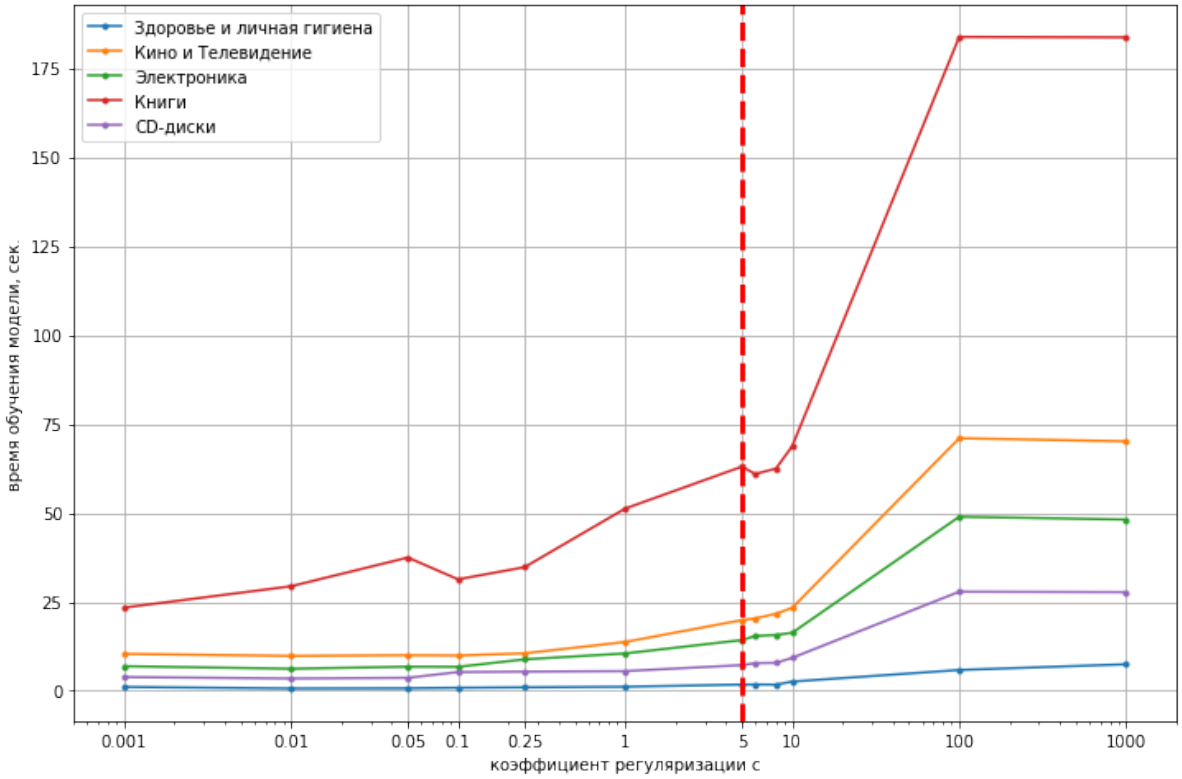

Рис. 4. Зависимость времени работы классификатора от коэффициента C

лизованного с использованием регуляризованной логистической регрессии. Так как отзывов по категории товаров: "Здоровье и личная гигиена" было меньше всего, то можем наблюдать не такие высокие результаты работы классификатора, как по остальным категориям.

На рисунке 5 изображено облако слов, реализованное в программе Tableau Public, для категории: "Кино и Телевидение" . В положительных словах мы видим слова типа: "best" , "wonderful" , "excellent" , "awesome" и т.д., что показывает, как люди обычно выражают положительные эмоции о товарах в этой категории. Также на рисунке видно, какие весовые коэффициенты получили слова в модели: начиная от слова "great" с коэффициентом 11.33 и заканчивая словом "outstanding" с коэффициентом 6.59.

Что касается отрицательных слов, то здесь присутствуют такие слова как: "poorly" , "terrible" , "awful" , "lacklustre" и т.д. От весового коэффициента зависит как размер слова, так и его цвет, что можно увидеть на рисунке 5.

Рис. 5. Облако слов для категории: "Кино и Телевидение"

Рисунок 6 изображает комбинированное облако слов, которое включает в себя и положительные и отрицательные слова. Синим цветом выделены положительные слова, тогда как красным - отрицательные.

Рис. 6. Облако слов для категории: "Электроника"

6. Заключение

Анализ социальной активности потребителей в интернете, касающейся тональности оставленных ими отзывов является актуальной задачей, что видно по росту количества научных работ, посвященных этой теме. Кроме того, в этом есть и прямая бизнес потребность, так как каждая компания хочет знать, что об её продуктах/услугах думают потребители.

В работе было проведено исследование предыдущих работ для определения того, какие методы являются современными, какие их особенности уже были выявлены, а также для определения вектора развития данной области. Далее были детально рассмотрены основные этапы работы по определению тональности:

- Нормализация и токенизация текста;
- Обработка слов в тексте;
- Векторизация отзывов;
- Логистическая регрессия;
- Оценка точности классификации.

Данные, используемые в работе были собраны пользователем Julian McAuley с сайта Amazon за период: с мая 1996 по июль 2014 года, где были выбраны различные категории товаров, такие как: Книги, Здоровье и личная гигиена, Кино и Телевидение, Электроника и CD-диски.

В расчётной составляющей произведено изучение алгоритма по анализу тональности текстовых данных, где был рассмотрен и использован метод регуляризованной логистической регрессии. При этом был написан алгоритм, который из заранее заданного диапазона коэффициентов регуляризации подбирает и использует для классификации параметр C, при котором модель имеет максимальное качество. Исходя из графика видно, что для разных наборов данных оптимальным являлся параметр $C = 5$. Используя это наблюдение, можем сделать вывод, что введя такую модель в полноценную систему, отвечающую за классификацию отзывов, можно сразу поставить параметр $C = 5$, для того, чтобы сократить время на его подбор.

Кроме этого были визуализированы результаты работы логистической регрессии, связанные с весами коэффициентов: было построено

облако слов, анализируя которое можно выявить какие слова ассоциируются с положительными эмоциями у потребителей, а какие с отрицательными. Из-за простой формы подачи материалы, таким анализом смогут пользоваться люди, не имеющих специальных знаний в машинному обучении, что является положительным фактором внедрения такой модели в компании.

Список литературы / References

[1] Сергей Орлов, "Журнал о современных тенологиях", *КОМПЬЮТЕРРА*, 16 января 2019.

[2] Rosie Murphy, "Local consumer review survey 2018", *Веб-сайт*, 7 декабря 2018.

[3] Medhat, Walaa and Hassan, Ahmed and Korashy, Hoda, "Sentiment analysis algorithms and applications: A survey", *Ain Shams engineering journal*, **5** (2014), 1093 – 1113.

[4] Yu Liang-Chih, Wu Jheng-Long, Chang Pei-Chann, Chu Hsuan-Shou, "Using a contextual entropy model to expand emotion words and their intensity for the sentiment classification of stock market news", *Knowl-Based Syst*, **41** (2013), 89 – 97.

[5] Teng-Kai Fan, Chia-Hui Chang, "Blogger-Centric Contextual Advertising", *Expert Systems with Applications*, **38**:3 (2011), 1777 – 1788.

[6] Gamal, Donia and Alfonse, Marco and M El-Horbaty, El-Sayed and M Salem, Abdel-Badeeh, "Analysis of Machine Learning Algorithms for Opinion Mining in Different Domains", *Machine Learning and Knowledge Extraction*, **1** (2019), 224 – 234.

[7] Sasikala P, L.Mary Immaculate Sheela, "Sentiment Analysis and Prediction of Online Reviews with Empty Ratings", *International Journal of Applied Engineering Research*, **13** (2018), 11525 – 11531.

[8] Sasikala P, L.Mary Immaculate Sheela, "Sentiment Analysis of Online Food Reviews using Customer Ratings", *International Journal of Pure and Applied Mathematics*, **119** (2018), 3509 – 3514.

[9] Bhatt, Aashutosh and Patel, Ankit and Chheda, Harsh and Gawande, Kiran, "Amazon Review Classification and Sentiment Analysis", *International Journal of Computer Science and Information Technologies*, **6** (2015), 5107 – 5110.

[10] Porter, Martin F, "An algorithm for suffix stripping", *Program*, **3** (1980), 130 – 137.

[11] Halácsy, Péter and Trón, V, "Benefits of deep NLP-based Lemmatization for Information Retrieval.", *CLEF (Working Notes)*, 2006.

[12] McClure, Nick, "TensorFlow machine learning cookbook", 2017, 185 – 187.

[13] Huang, Cheng-Hui and Yin, Jian and Hou, Fang, "A text similarity measurement combining word semantic information with TF-IDF method", *Jisuanji Xuebao(Chinese Journal of Computers)*, **34** (2011), 856 – 864.

[14] Scikit-learn, TfidfTransformer https://scikit-learn.org/stable/modules/feature_extraction.html.

[15] Müller, Andreas C and Guido, Sarah and others, "Introduction to machine learning with Python: a guide for data scientists", 2016, 339–341.

[16] Логистическая регрессия, определения https://www.machinelearning.ru.

[17] Scikit-learn, Logistic regression Python https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html#sklearn.linear_model.LogisticRegression.

[18] Scikit-learn, Logistic regression Description https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression.

[19] Amazon product data, files http://jmcauley.ucsd.edu/data/amazon/.