САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

КАФЕДРА ТЕХНОЛОГИИ ПРОГРАММИРОВАНИЯ

Шепелев Павел Леонидович

Выпускная квалификационная работа бакалавра

Методы анализа тональности отзывов пользователей ресторанов

Направление 01.03.02

Прикладная математика и информатика

Научный руководитель,
канд. техн. наук,
доцент
Блеканов И.С.

Рецензент,
старший преподаватель,
Давыденко А.А.

Санкт-Петербург

2020

Содержание

Введение

Распространение интернета в современном мире позволило многим видам бизнеса полностью или частично перейти в онлайн. Коммуникация между бизнесом и клиентом в настоящее время зачастую происходит с использованием интернет технологий. Ежедневно тысячи людей заказывают товары в интернет магазинах, бронируют столики в ресторанах через приложения, а также оставляют отзывы на товары и услуги. Наличие отзывов на товары и услуги повышает доверие клиентов к бизнесу и способствует увеличению продаж. Добавление отзывов на страницу товара в интернет магазине может увеличить его продажи более чем в 1,4 раза [1]. Кроме того отзывы пользователей повышают количество уникального контента на сайте, что способствует продвижению сайта в поисковых системах.

Успех бизнеса напрямую зависит от его способности удовлетворять желания и потребности клиентов. Это утверждение верно как для производителей товаров, так и для предприятий, оказывающих услуги. Например, уровень качества сервиса и еды в ресторанах оказывает непосредственное влияние не только на удовлетворенность клиентов, но и на число постоянных клиентов [2].

Анализ тональности отзывов является полезным инструментом для бизнеса, так как позволяет автоматически получать мнения пользователей о товаре или услуге для анализа их качества и сравнения его с конкурентами. Популярные рекомендательные системы, такие как Yelp, Tripadvisor, Foursquare позволяют пользователям помимо отзывов ставить оценки бизнесу, и эта оценка может являться показателем значения тональности соответствующего отзыва. Однако если пользователь не оставил отзыв на одном из таких сайтов, владельцы бизнеса могут попросить клиента дать обратную связь через email рассылку. При больших объемах обратной связи

возникает необходимость автоматической классификации полученных отзывов.

При покупке новых товаров и заказе новых услуг люди очень сильно полагаются на отзывы о товарах. Согласно опросу BrightLocal [3] положительные отзывы увеличивают вероятность использования услуги у 91% респондентов, в то время как 82% опрошенных отпугнут отрицательные отзывы. Использование автоматического анализа тональности в рекомендательных системах позволяет пользователям не знакомым с товаром или услугой узнать, что думают о данном товаре или услуге те, кто уже их использовал без необходимости в чтении всех отзывов.

В данной работе будет продемонстрировано сравнение алгоритмов машинного обучения в задаче классификации русскоязычных отзывов на рестораны, а также некоторые методы улучшения качества классификации.

Постановка задачи

Целью данной работы является разработка метода автоматического определения тональности русскоязычных отзывов на рестораны. Каждый отзыв должен быть классифицирован как положительный, нейтральный или отрицательный. Качество работы алгоритма, оцениваемое F-мерой должно быть не меньше 60%.

Для достижения поставленной в работе цели были поставлены следующие задачи:

1. Написание программы для сбора отзывов. Формирование корпуса данных.
2. Предобработка данных.
3. Построение векторных моделей отзывов.
4. Реализация некоторых алгоритмов машинного обучения.
5. Решение проблемы несбалансированных классов.
6. Реализация системы перекрестной проверки алгоритмов и векторных моделей. Выбор алгоритма и векторной модели с лучшими показателями F-меры.

Обзор литературы

Анализ тональности текста представляет собой задачу классификации текста, основанную на эмоциональном отношении автора текста к описываемому объекту или явлению. Методы анализа тональности текста можно разделить на две группы:

1. Методы, основанные на словарях и лексиконах
2. Методы, основанные на машинном обучении

Методы, основанные на словарях и лексиконах, требуют наличия словаря, в котором словам присваивается тональная оценка. Тональная окраска текста вычисляется как функция от значений тональности всех слов текста, которые присутствуют в словаре.

Тональный словарь английского языка SentiWordNet [4] был составлен на основе словаря WordNet, где каждому синсету из базы WordNet были поставлены в соответствие три числа: позитивность, негативность и объективность.

Для русского языка в настоящий момент существует несколько словарей тональности. Словарь RuSentiLex [5] составлен на основе твитов и новостных статей. Для каждого слова в словаре указывается полярность слова (позитивная, негативная или нейтральная), источник тональности (прямо выраженная оценка, эмоция или коннотация). Кроме того, словарь содержит тональные различия между значениями многозначных слов.

Словарь SentiRusColl составлен из словосочетаний от 2 до 5 слов, которые определяют тональность текстов из 10 тем. Kotelnikov и Kotelnikova [6] использовали объединение словарей RuSentiLex и SentiRusColl, что позволило превзойти F-меру классификации, показанную этими словарями по отдельности.

Недостатком методов основанных на словарях является привязанность конкретного словаря к языку и предметной области. Кроме того составление

словаря вручную экспертом лингвистом требует значительных трудовых и временных затрат.

Методы машинного обучения более универсальны, так как могут быть применены для анализа тональности текстов на любом языке и для любой предметной области. Наиболее часто исследователи прибегают к методам обучения с учителем.

Pang et al. [7] исследовали эффективность методов машинного обучения в задаче тональной классификации отзывов на фильмы. Исследователи сравнивали следующие классификаторы: наивный байесовский классификатор, метод максимальной энтропии и метод опорных векторов. Кроме того, каждый классификатор был обучен на разных наборах признаков: униграммы, биграммы, униграммы и теги частей речи, прилагательные. Наилучший результат показал метод опорных векторов, обученный на униграммах.

Yu et al. [8] построили модель для определения тональных слов в отзывах на рестораны из датасета Yelp. Применение метода опорных векторов для определения полярности слов позволило получить списки положительных и отрицательных характеристик ресторанов различной кухни.

Sharif et al. [9] провели сравнение качества работы наивного байесовского классификатора, дерева решений и случайного леса в задаче классификации отзывов на рестораны на бенгальском языке. Наилучшую точность на кросс-валидации в поставленной задаче показал наивный байесовский классификатор.

Недостатком методов машинного обучения с учителем является необходимость в размеченном корпусе данных. В случае отсутствия размеченных данных применяются методы обучения без учителя.

Turney [10] предложил метод оценки семантической ориентации фраз, содержащих имена прилагательные и наречия. Семантической ориентация фразы определяется путем вычисления взаимной информации с положительным словом (“excellent”) и отрицательным словом (“poor”). Тональность фразы считается положительной, если положительна средняя семантическая ориентация входящих в нее слов.

Putri et al. [11] исследовали задачу классификации отзывов на туристическом сайте. В качестве решения был предложен метод обучения без учителя - латентное размещение Дирихле.

В последние годы лучшие результаты в сентимент анализе показали предобученные нейросетевые модели BERT [12] и XLNet [13]. Данные модели являются нейросетями архитектуры Transformer. Эти модели были обучены решать несложные задачи на огромном объеме данных, после чего предобученные модели используются для решения других, более специфических задач [14].

Sousa et al. [15] провели эксперименты по предсказанию значений индекса Dow Jones в зависимости от тональности новостей из различных источников за соответствующий день. Модель BERT значительно превзошла по точности наивный байесовский классификатор и метод опорных векторов, а также сверточную нейросеть TextCNN.

Gong et al. [16] провели эксперимент для задачи классификации отзывов на фильмы с использованием классификаторов основанных на нейронных сетях. Авторами статьи была предложена модель BroXLNet, которая улучшила контекстное представление сети BLS [17] путем поиска дополнительных признаков. Модель BroXLNet показала более высокую точность классификации, чем BERT и XLNet на датасете SST-2.

Большинство исследований решают задачу классификации текста на уровне документа или предложения. Более сложной задачей является аспектный анализ тональности. Эта задача разбивается на две подзадачи [18]: выделение аспектов и оценка тональности терминов относящихся к каждому аспекту.

Rybakov и Malafeev [19] решают задачу аспектного анализа тональности отзывов на отели на русском языке. Для векторного представления слов использовалась модель word2vec. Для каждой аспектной категории вручную был составлен словарь изначальных терминов, который впоследствии дополнялся наиболее близкими словами по косинусному сходству. Похожим образом выбирались тональные слова для классов “positive” и “negative”.

Nombre et al. [20] предложили свое решение задачи аспектного анализа тональности отзывов на товары. Для выделения аспектов использовался метод CRF, обученный на вручную размеченной выборке. Для классификации по тональности использовался метод опорных векторов.

Глава 1 Формирование корпуса данных

1.1 Корпус данных

Корпус данных состоит из 22258 русскоязычных отзывов на рестораны Москвы и Санкт-Петербурга, оставленных на сайте restoclub.ru, а также оценок от 1.0 до 10.0, поставленных пользователями соответствующим ресторанам. Для получения данных с сайта была написана программа для парсинга этого сайта.

В работе была поставлена задача классификации отзывов на 3 класса: положительный, отрицательный и нейтральный. Для разделения корпуса данных на классы каждому отзыву в соответствие была поставлена метка ‘1’ для положительных отзывов (оценка ≥ 7.0), ‘-1’ для отрицательных отзывов (оценка ≤ 4.0) и ‘0’ для нейтральных отзывов (4.0< оценка <7.0).

Анализ тональности может применяться как на уровне предложений, так и на уровне текстов целиком. Один отзыв может содержать предложения с разной эмоциональной окраской, но оценку пользователь ставит одну на весь отзыв. Поэтому в данной работе было решено оценивать тональность текстов целиком.

1.2 Предобработка данных

Предобработка данных является важным и необходимым этапом решения задачи анализа данных. Целью предобработки является приведение данных к более удобному виду для решения поставленной задачи.

Первым этапом предобработки текста является токенизация. Суть токенизации текста заключается в делении текста на более мелкие смысловые единицы – токены. Минимальной смысловой единицей текста является слово. Деление текстов на слова по пробелам между словами, не является оптимальным, так как слова могут разделять знаки препинания, перенос строки, табуляция и др. Поэтому лучше использовать набор правил

токенизации, который может меняться в зависимости от языка. Знаки препинания после токенизации удаляются, так как они не несут нужной информации. Также удаляются лишние пробелы, символы переноса строки и табуляции.

Одно и то же слово, написанное с заглавной буквы и со строчной не эквивалентны друг другу, но чаще всего имеют одинаковый смысл. Поэтому все слова были приведены к строчному регистру.

В русском языке служебные части речи: предлоги, частицы и союзы служат для связи слов в предложении, но при этом сами по себе не имеют смысла. Они часто встречаются в отзывах любой тональности и являются шумом, плохо влияющим на качество классификации. Такие слова называют стоп-словами, их также удаляют на этапе предобработки. Кроме стоп-слов из отзывов были удалены гиперссылки и числа. Слова на латинице в отзывах обычно обозначают название ресторанов и блюд, и не являются важными признаками, поэтому также удаляются. Среди собранных отзывов было небольшое количество отзывов на английском языке, которые были удалены.

Следующим шагом предобработки является нормализация слов. Слова в русском языке могут иметь различные формы. Например, имена существительные могут стоять в разных падежах, глаголы имеют различные времена и так далее. Задача нормализации – преобразование различных форм слова к нормальной форме. Существует два способа нормализации слов: стемминг и лемматизация. В русском языке форма слова чаще всего образуется с помощью окончаний. Стемминг представляет собой процесс получение основы слова путем отсечения окончаний и суффиксов. Полученная основа слова будет одинаковой для всех форм слова. Лемматизация – процесс приведения слов к первоначальной словарной форме. В отличие от стемминга, чаще всего основанного на некоторых эвристиках, лемматизация реализуется с помощью словарей и

морфологических анализаторов. В данной работе нормальные формы всех слов корпуса данных были получены с помощью лемматизации.

На этапе предобработки была решена проблема отрицания слов. В русском языке для обозначения отрицания используется частица “не”. Удаление этой частицы как стоп-слова может изменить тональность предложения на противоположную, что негативно скажется на качестве классификации. Для того чтобы верно распознавать отрицания, конструкции вида “не слово” были преобразованы в “неслово”.

Слова не являются единственным способом выражения эмоций в тексте. В настоящее время широкое распространение в интернете получили специальные символы – эмодзи. Эмодзи описывают не только настроение и эмоции, но также предметы и явления. В настоящее время словарь эмодзи насчитывает 3304 символа [21]. Многие исследователи используют эмодзи в качестве признаков в задаче анализа тональности. Le Compte и Chen [22] показали, что добавление эмодзи к словесным признакам улучшило качество классификации твитов для наивного байесовского классификатора и метода опорных векторов. Wankhede et al. [23] использовали выделение эмодзи вместе с исправлением опечаток для повышения качества классификации твитов. Эмодзи могут быть хорошими признаками для определения тональности текста, поэтому было реализовано извлечение эмодзи из слов. На рисунке 1 показан результат предобработки одного из отзывов.

```
Выражаю огромную благодарность владельцу и коллективу ресторана Бархат! Я
выбрала Бархат для празднования своего юбилея и не прогадала! Потрясающе
вкусная еда, внимательный персонал, очень уютная и тёплая атмосфера! Все
23 человека гостей остались довольны! 😍 Спасибо! До новых встреч!!!


выражать огромный благодарность владелец коллектив ресторан бархат
выбрать бархат празднование свой юбилей непрогадать потрясающе вкусный
еда внимательный персонал очень уютный теплый атмосфера весь человек
гость остаться довольный 😍 спасибо новый встреча
```

Рисунок 1. Пример отзыва до и после предобработки.

1.3 Построение векторного пространства признаков

Алгоритмы машинного обучения предназначены для работы с числовыми данными, поэтому необходимо некоторым образом преобразовать текст отзыва в числовой вектор признаков.

Модель bag-of-words является моделью упрощенного представления текстового документа. Каждый отзыв представляется в виде неупорядоченного набора слов. Для получения числовых векторов признаков из текста существует несколько мер. Мера TF (term frequency) – частота вхождений слова в отзыв. Вес слова t из отзыва $d \in D$ рассчитывается по формуле:

$$TF(t,d) = \frac{n_t}{n_d}$$

где n_t-число вхождений слова t в отзыв d, n_d- общее число слов в отзыве d. Из определения меры TF следует, что чем чаще слово встречается, тем больше его вес. Но если определенное слово часто встречается во всех отзывах коллекции, оно не должно иметь важности при классификации. Этот факт учитывается в мере TF – IDF.

TF – IDF - статистическая мера, используемая для оценки важности слова в контексте документа, являющегося частью корпуса [24]. TF – IDF является произведением TF и IDF (inverse document frequency) [25]. Вес слова t из отзыва $d \in D$ рассчитывается по формуле:

$$TF - IDF(t,d,D) = \frac{n_t}{n_d} \times log \frac{|D|}{|\{d_i \in D | t \in d_i\}|}$$

В отличие от меры TF, мера TF – IDF уменьшает вес наиболее частых слов, за счет множителя IDF. Таким образом, наибольший вес будут иметь слова с высокой частотой употребления в данном отзыве и редко встречающиеся в других отзывах.

Одним из недостатков модели bag-of-words является тот факт, что при использовании этой модели не учитывается информация о взаимном расположении слов в тексте, которая влияет на смысл текста. Использование словесных n-грамм позволяет увеличить пространство признаков, а также учитывать порядок слов.

N-граммы – это последовательности из n подряд идущих слов. В данной работе для представления отзывов были выбраны униграммы, биграммы, триграммы и их комбинации. Использование n-грамм более высокого порядка уменьшает вероятность встретить конкретный термин в других отзывах, что влечет за собой разреженность данных. Такая модель будет более склонна к переобучению. Недостатком использования n-грамм является также большое увеличение памяти для их хранения.

1.4 Проблема несбалансированных данных

Из 22258 отзывов в корпусе данных 17854 (80,2%) положительных, 2381 (10,7%) нейтральных и 2023 (9,1%) отрицательных отзывов. Наличие такой несбалансированности данных может привести к неточности классификации.

Методы решения данной проблемы можно поделить на три типа [26]:

1. Алгоритмические методы
2. Методы с учетом издержек классификации
3. Методы на уровне данных

Алгоритмические методы и методы с учетом издержек классификации корректируют метод классификации, в то время как методы на уровне данных изменяют распределение классов данных в обучающем множестве. Методы на основе данных являются более универсальными, так как могут применяться совместно с любыми алгоритмами, а также считаются одними из наиболее эффективных способов решения проблемы несбалансированных классов [27].

Одним из методов на основе данных является сэмплинг обучающей выборки. Различают две стратегии сэмплинга [28]:

1. Undersampling – удаление некоторого количества элементов мажоритарного класса.
2. Oversampling – увеличение количества элементов миноритарного класса.

В силу небольшого размера корпуса данных для решения поставленной задачи больше подходит стратегия oversampling. Увеличение числа элементов миноритарного класса может достигаться путём копирования экземпляров этого класса или путем создания новых примеров необходимого класса. Для реализации был выбран oversampling алгоритм SMOTE [29]. Этот алгоритм создает искусственные элементы миноритарного класса следующим образом:

1. Находится разность d между векторами признаков соседних элементов a и b этого класса, найденных методом k–ближайших соседей

$$d = X_b - X_a$$

2. Формируется вектор нового элемента с:

$$X_c = X_a + i\,d$$

где i – случайное число, $i \in (0,1)$.

На рисунке 2 показана графическая интерпретация алгоритма SMOTE.

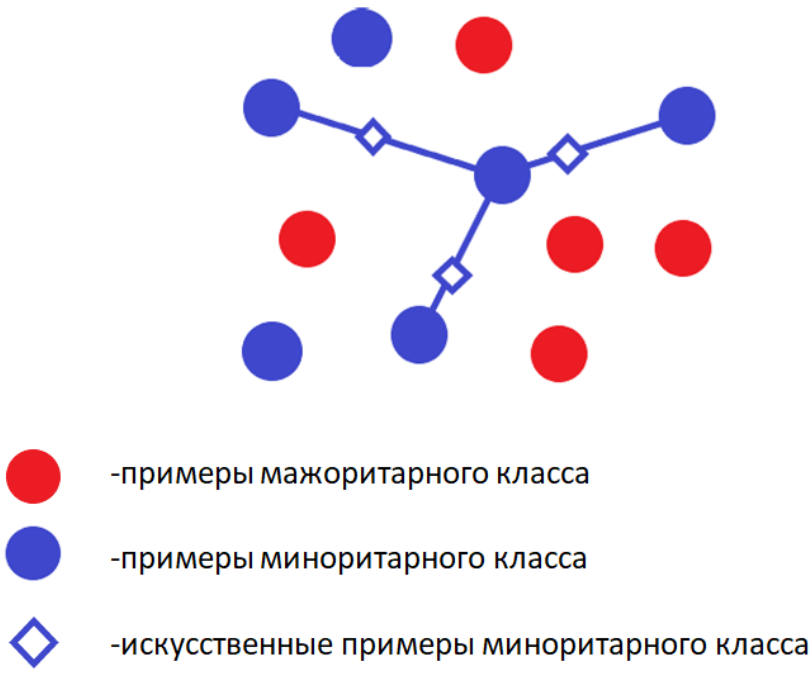


Рисунок 2. Результат применения алгоритма SMOTE.

Применив алгоритм SMOTE к объектам классов нейтральных и отрицательных отзывов, можно улучшить общее качество классификации.

Глава 2 Алгоритмы классификации

2.1 Наивный байесовский классификатор

Наивный байесовский классификатор – вероятностный алгоритм классификации, основанный на теореме Байеса. Для каждого отзыва d из корпуса отзывов классификатор вычисляет класс $\tilde{c}$, к которому наиболее вероятно принадлежит этот отзыв [30]:

$$\tilde{c} = \underset{c \epsilon C}{\operatorname{argmax}} P(c|d) = \underset{c \epsilon C}{\operatorname{argmax}} P(d|c)P(c) \qquad (1)$$

где $P(c|d)$ - вероятность того, что отзыв d принадлежит классу $c \epsilon C$, $P(d|c)$ – вероятность появления отзыва d в классе c. $P(c)$ – априорная вероятность класса c, определяемая формулой $P(c) = \frac{D_c}{D}$, где D_c - число отзывов принадлежащих классу c, D- общее число отзывов.

Наивный байесовский классификатор делает два допущения:

1) отзыв d представляется в виде множества слов $f_1, f_2, \dots, f_n$, при этом порядок слов не учитывается. Такая модель получила название “мешок слов”.

2) Вероятности слов $P(f_i|c),\ i = \overline{1,n}$ являются независимыми. Из этого предположения можно вывести следующую формулу для вычисления вероятности отзыва d в классе c:

$$P(d|c) = P(f_1, f_2, \dots, f_n|c) = \prod_{i=1}^{n} P(f_i|c)$$

В итоге формула (1) будет иметь следующий вид:

$$c_{NB} = \underset{c \epsilon C}{\operatorname{argmax}} P(c) \prod_{i=1}^{n} P(f_i|c) \qquad (2)$$

При больших размерах текстов под знаком произведения в формуле (2) возникает большое количество малых по значению множителей. Для того

чтобы избежать проблемы арифметического переполнения снизу, используется следующая формула:

$$c_{NB} = \underset{c \epsilon C}{\operatorname{argmax}} \log P(c) \sum_{i=1}^{n} \log P(f_i|c)$$

2.1 Логистическая регрессия

Логистическая регрессия – метод построения линейного классификатора. На вход алгоритму подается набор данных $(x^{(i)}, y^{(i)})$. Пусть $x = (x_1, x_1, \ldots, x_n)$ - вектор признаков, y – переменная, которую нужно классифицировать как объект класса $c \epsilon C$, K -число классов. Для этого найдем вероятность

$$P(y = c|x) = \frac{e^{w_c \cdot x + b_c}}{\sum_{j=1}^{n} e^{w_j \cdot x + b_j}}$$

где $w = (w_1, w_2, \ldots, w_n)$ -вектор весов признаков, $b = (b_1, b_2, \ldots, b_n)$ - смещение [30]. Функцию в правой части уравнения называют softmax-функцией. Она используется вместо сигмоидной функции в задаче многоклассовой классификации.

Функция потерь рассчитывается как:

$$L_{CE} = \begin{cases} -\sum_{k=1}^{K} log \frac{e^{w_k \cdot x + b_k}}{\sum_{j=1}^{K} e^{w_j \cdot x + b_j}}, & если\ y = k \\ 0, & если\ y \neq k \end{cases}$$

Вектор весов признаков w находится путем минимизации функции потерь L_{CE}.

2.3 Метод стохастического градиента

Метод стохастического градиента применяется для подбора вектора весов в линейном классификаторе. Пусть $X^l = (x_i, y_i)\,{}_{i=1}^{l}$ - обучающая выборка, $y_i = y^*(x_i)$. Нужно найти алгоритм $a(x, w)$, аппроксимирующий y^* вида [31]:

$$a(x, w) = sign(\sum_{j=1}^{n} w_j x^j - w_0)$$

Для этого методом градиентного спуска решается оптимизационная задача

$$Q(w) = \sum_{i=1}^{n} L(a(x_i, w), y_i) \to \min_w$$

где $L(a, y)$ -заданная функция потерь. На каждой итерации вектор весов изменяется в направлении антиградиента $w_{k+1} = w_k - \eta \nabla Q(w)$.

2.4 AdaBoost классификатор

Алгоритм AdaBoost в процессе обучения строит композиции базовых алгоритмов (в данной работе деревьев решений), причем каждый следующий классификатор строится по объектам, неверно классифицированным предыдущим классификатором. Для многоклассовой классификации используется модификация AdaBoost - алгоритм SAMME [32]. Пусть $(x_i, c_i)\,{}_{i=1}^{n}$ – обучающее множество, Нужно найти алгоритм $C(x)$. Сначала каждому x_i из обучающего множества ставится в соответствие вес $w_i = \frac{1}{n}$. На каждом m – ом шаге на взвешенных данных тренируется классификатор $T^m(x)$. Далее считается коэффициент взвешенного голосования:

$$a^{(m)} = log\frac{1 - err^{(m)}}{err^{(m)}} + \log(K - 1)$$

где $err^{(m)} = \frac{\sum_{j=1}^{n} w_j}{\sum_{i=1}^{n} w_i}, c_j \neq T^m(x_j)$ - взвешенная ошибка, K -число классов. После этого присваивается новое значение весам $w_i \leftarrow w_i \cdot exp(a^{(m)}), i = \overline{1,n}$.

После нормировки весов объектов $w_0 = \sum_{j=1}^{n} w_j$, $w_i \leftarrow \frac{w_i}{w_0}, i = \overline{1,n}$, алгоритм переходит на следующий шаг. Через M шагов алгоритм возвращает наиболее вероятный класс:

$$C(x) = arg\max_k \sum_{m=1}^{M} a^{(m)}$$

2.5 Метод опорных векторов

Основной идеей метода опорных векторов является построение гиперплоскости, разделяющей объекты на два класса оптимальным образом. Пусть $X^l = (x_i, y_i)\ _{i=1}^{l}$- обучающая выборка, $y_i = y^*(x_i), X = \mathbb{R}^n, Y = \{-1, 1\}$. Нужно найти алгоритм $a: X \to Y$, аппроксимирующий y^* [33]:

$$a(x) = sign(\sum_{j=1}^{n} w_j x^j - w_0) = sign((w, x) - w_0)$$

Разделяющая гиперплоскость описывается уравнением $(w, x) = w_0$. Задача классификатора состоит в выборе таких параметров w и w_0, чтобы максимизировать расстояние до каждого из классов. Если выборка линейно разделима, решается оптимизационная задача:

$$\begin{cases} (w, w) \to \min \\ y_i\big((w, x_i) - w_0\big) \geq 1, i = \overline{1,l} \end{cases}$$

В случае линейно неразделимой выборки осуществляется переход от исходного пространства признаков X к пространству более высокой

размерности H, в котором выборка окажется линейно разделимой. В этом случае классификатор примет вид:

$$a(x) = sign((w, \psi(x)) - w_0), \psi: X \to H$$

В задаче n- классовой классификации методом опорных векторов решается n задач бинарной классификации “один против всех”.

Глава 3 Программная реализация и анализ результатов

В качестве инструмента реализации программ использовался язык Python 3 в среде разработки Jupyter Notebook. Для предобработки отзывов использовался лемматизатор pymorphy2, а также библиотека nltk. Для реализации алгоритмов машинного обучения использовалась библиотека Scikit-learn.

3.1 Парсинг данных

Парсинг- процесс сбора информации с веб-сайтов, автоматизированный с помощью специальной программы. Доступ к сайтам осуществляется с помощью протокола HTTP. Структура веб-страницы определяется кодом на языке разметки HTML. Содержимое веб-страницы структурировано с помощью html-тегов.

Работа программы для парсинга начинается с отправки HTTP запроса на сайт. Запросы в программе были реализованы с помощью библиотеки requests. В результате GET запроса программа возвращает код веб страницы в формате html. Библиотека Beautiful soup позволяет преобразовать HTML разметку в дерево синтаксического разбора [34]. Использование этой библиотеки позволяет получить данные внутри html-тегов, даже в случае неправильной разметки на сайте. Перед использованием парсера необходимо преобразовать html код страницы в формат строки. Для получения необходимой информации нужно определить параметры тех элементов разметки веб-страницы, внутри которых находятся нужные данные.

Программа для парсинга сайта включает в себя два модуля: parse_restaurants и parse_reviews. Модуль parse_restaurants осуществляет сбор названий ресторанов и ссылок на страницы ресторанов из базового url адреса:

https://www.restoclub.ru/msk/search/?expertChoice=false&averageBill%5B%5D=-800&averageBill%5B%5D=800-1500&averageBill%5B%5D=1500-2000&averageBill%5B%5D=2000-3000&averageBill%5B%5D=3000-

В базовом url адресе были заданы параметры отбирающие все группы ресторанов по среднему чеку, что позволило отобразить ссылки на все страницы ресторанов.

После того как был получен список ссылок на страницы ресторанов, модуль parse_reviews осуществляет парсинг отзывов с каждой страницы из полученного списка. Со страницы ресторана выделяется html-элемент, содержащий отзыв, после чего текст отзыва и оценка добавляются в коллекцию данных.

3.2 Кросс-валидация

Кросс-валидация – процедура эмпирического оценивания обобщающей способности алгоритмов, обучаемых по прецедентам [35]. Простейшим вариантом кросс-валидации является валидация на отложенных данных. Для этого весь корпус данных разбивается на два непересекающихся множества: обучающее и отложенное. Алгоритм обучается на обучающем множестве, качество классификации измеряется на отложенном множестве. Преимуществом валидации на отложенных данных является то, что модель обучается один раз, что экономит время и вычислительные мощности. Недостатком данного способа валидации является зависимость от способа разбиения данных. Валидация на отложенных данных подходит для задач с большим объемом данных. При малых размерах данных отложенная выборка может оказаться нерепрезентативной [36]. Кроме того снижается объем данных, доступных для обучения, что является критичным фактором в случае небольшого датасета.

Для уменьшения зависимости от разбиения данных применяется k-fold кросс-валидация. Корпус данных разбивается на k непересекающихся блоков. На каждой из k итераций проверка алгоритмов производится на одном из блоков, а обучение на оставшихся k-1 блоках. В качестве результата рассчитывается средняя ошибка по всем k разбиениям. Недостатком этого способа валидации являются значительные временные и вычислительные затраты, так как алгоритм приходится обучать k раз.

Для определения лучшей модели классификации была проведена процедура кросс-валидации по 5 блокам. Схема разбиения данных на обучающее и тестовое множество представлена рисунке 3.

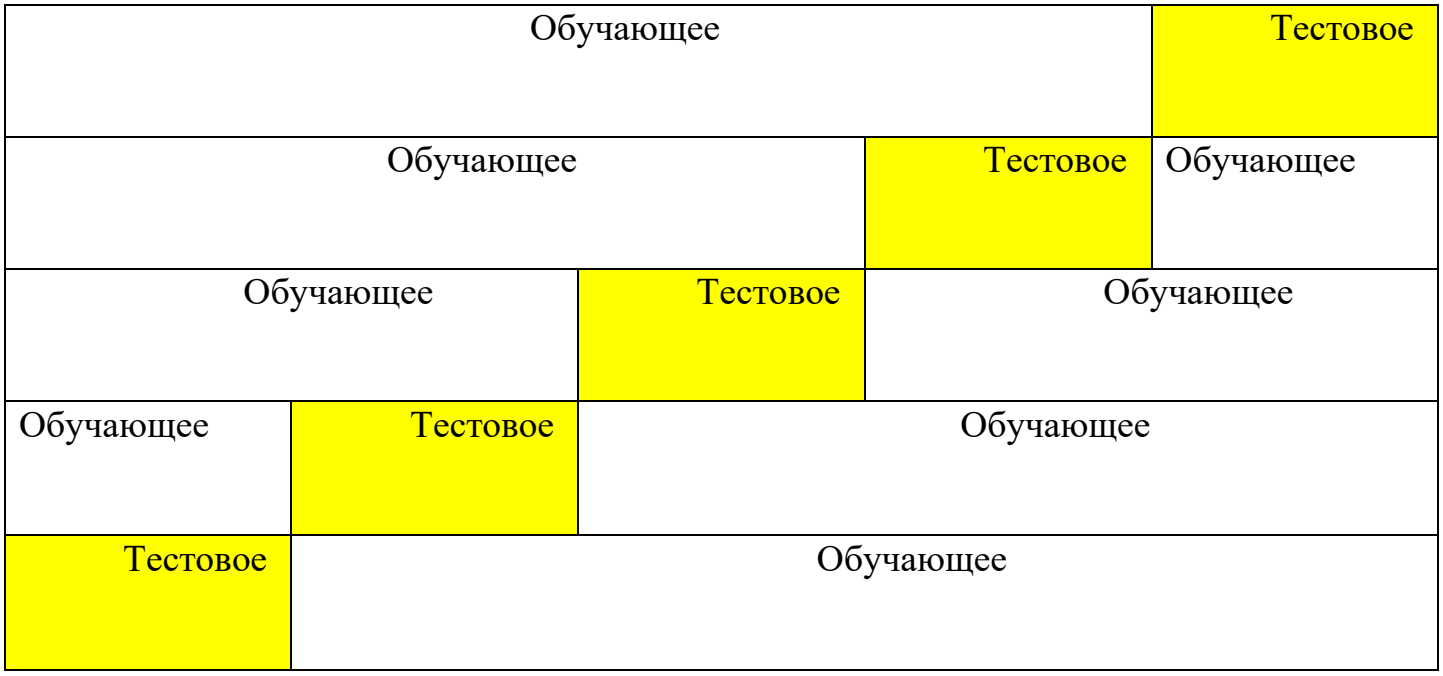


Рисунок 3. Схема кросс-валидации.

Для каждого блока производилось обучение на алгоритмах: наивный байесовский классификатор, логистическая регрессия, AdaBoost классификатор, метод стохастического градиента и метод опорных векторов. Также каждый алгоритм был обучен на следующем наборе n-грамм: 1-грамм, 2-грамм, 3-грамм, 1-грамм + 2-грамм, 2-грамм + 3-грамм, 1-грамм + 2-грамм + 3-грамм. Кроме того каждый алгоритм отдельно обучались на исходной

обучающей выборке и на обучающей выборке, дополненной искусственно созданными с помощью алгоритма SMOTE объектами классов ‘нейтральный’ и ‘отрицательный’ до размеров класса ‘положительный’ из этой выборки

3.3 Методы оценки качества классификации

Для сравнения алгоритмов необходимо ввести метрики оценки качества классификации. Популярной и легко интерпретируемой метрикой является метрика Accuracy:

$$Accuracy = \frac{P}{N}$$

где P - число верно предсказанных отзывов, N -общее число отзывов тестового множества. Данная метрика не подходит для оценки качества классификаторов, обученных на несбалансированных данных. Например, если построить для нашего корпуса данных классификатор, предсказывающий каждый отзыв как положительный, то значение метрики Accuracy равнялось бы 80,2%, хотя очевидно, что такой классификатор не является эффективным. Поэтому в поставленной задаче используются альтернативные метрики.

В задачах бинарной классификации выделяют следующие метрики качества классификации:

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

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

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

где TP- истинно-положительные значения, FP- ложно-положительные значения, TN- истинно-отрицательные значения, FN-ложно-отрицательные значения.

В данной работе рассматривается задача многоклассовой классификации, поэтому необходимо определить данные метрики для случая трех классов. В таблице 1 обозначены метрики Precision, Recall и F-score для задач бинарной классификации “один против всех”. Например:

$$R_1 = \frac{TP}{TP + FN}$$

где TP – количество верно предсказанных объектов класса “положительный”, FN – количество объектов класса “положительный”, ошибочно предсказанных как объекты объединённого класса “нейтральный или негативный”.

Класс	Precision	Recall	F-score
положительный	P_1	R_1	F_1
нейтральный	P_2	R_2	F_2
отрицательный	P_3	R_3	F_3

Таблица 1. Метрики в задаче трехклассовой классификации

В задаче многоклассовой классификации существует несколько способов определения F-меры. В данной работе в качестве метрики качества классификации будет взята macro F-score, определяемая формулой:

$$F - score = \frac{F_1 + F_2 + F_3}{3}$$

где $F_i = 2\frac{P_i * R_i}{P_i + R_i}, i = 1,2,3.$

3.4 Результаты

Для оценки качества моделей была проведена процедура 5-кратной кросс-валидации. В таблице 2 представлены три лучшие модели на каждом блоке кросс-валидации.

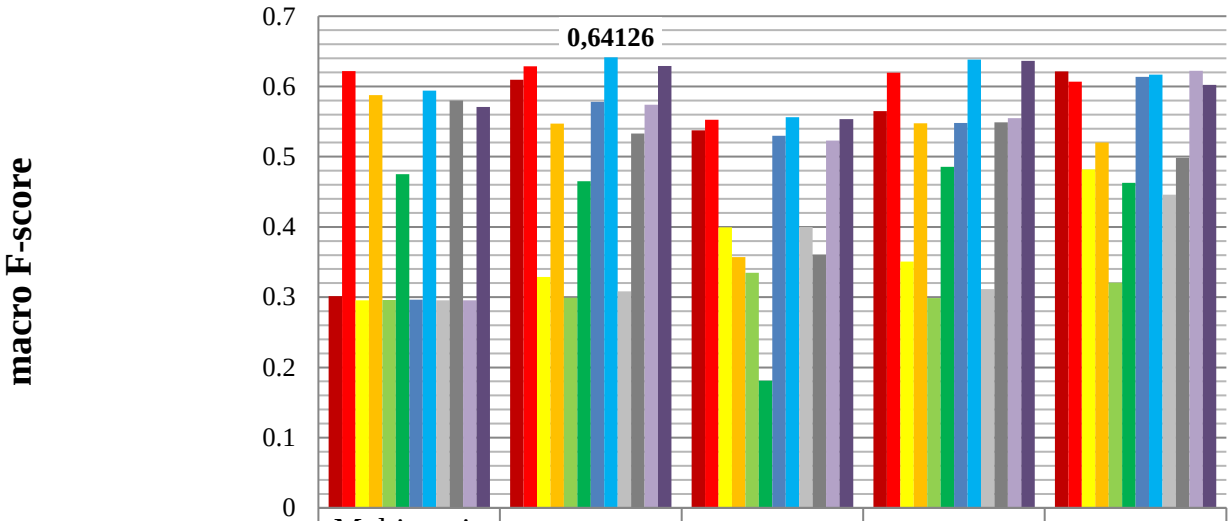
N fold		1 место		2 место		3 место	
		Модель	F-score	Модель	F-score	Модель	F-score
0	Алгоритм	SVM	0,6244	SGD Classifier	0,6174	SVM	0,6162
	N- граммы	1-грамм		1-грамм + 2-грам + 3-грамм		1-грамм + 2-грам + 3-грамм	
	SMOTE	нет		да		нет	
1	Алгоритм	SGD Classifier	0,6756	Logistic Regression	0,6756	SGD Classifier	0,6722
	N- граммы	1-грамм + 2-грамм		1-грамм + 2-грамм		1-грамм + 2-грам + 3-грамм	
	SMOTE	да		да		да	
2	Алгоритм	Logistic Regression	0,6288	Logistic Regression	0,6249	SGD Classifier	0,6216
	N- граммы	1- грамм + 2-грамм + 3-грамм		1-грамм + 2-грамм		1-грамм + 2-грам + 3-грамм	
	SMOTE	да		да		да	
3	Алгоритм	SGD Classifier	0,65	Logistic Regression	0,6445	SGD Classifier	0,6411
	N- граммы	1-грамм + 2-грамм		1-грамм + 2-грамм		1-грамм + 2-грам + 3-грамм	
	SMOTE	да		да		да	
4	Алгоритм	Logistic Regression	0,6655	Multinomial Naive Bayes	0,6642	SGD Classifier	0,6585

	N- граммы	1-грамм		1-грамм		1-грамм	
	SMOTE	да		да		да	

Таблица 2. Результаты кросс-валидации. Три лучших модели на каждом блоке.

На всех блоках кросс-валидации кроме “0” три лучшие модели были улучшены с помощью применения алгоритма SMOTE. Наибольшее значение F-меры =0,6756 было достигнуто на блоке “2” с использованием метода стохастического градиента, обученного на комбинации униграмм и биграмм.

Средние значения F- меры на кросс-валидации для моделей представлены в таблице 3.



	Multinomial Naïve Bayes	Logistic Regression	AdaBoost Classifier	SGD Classifier	SVM
1 gramm	0.30154	0.6096	0.5373	0.56504	0.62128
1 gramm SMOTE	0.62204	0.62848	0.55274	0.6194	0.60688
2 gramm	0.29564	0.32896	0.39944	0.35046	0.48192
2 gramm SMOTE	0.58754	0.54694	0.3572	0.54764	0.52024
3 gramm	0.29612	0.2991	0.3347	0.29868	0.32074
3 gramm SMOTE	0.47524	0.4649	0.18128	0.48578	0.4629
1+2 gramm	0.2966	0.57828	0.52984	0.54814	0.61382
1+2 gramm SMOTE	0.59396	0.64126	0.55606	0.63824	0.61664
2+3 gramm	0.29564	0.30812	0.39968	0.31138	0.44574
2+3 gramm SMOTE	0.58008	0.53288	0.36002	0.5487	0.49848
1+2+3 gramm	0.29564	0.57384	0.52298	0.55506	0.6225
1+2+3 gramm SMOTE	0.57066	0.62904	0.55334	0.63646	0.60236

Таблица 3. Результаты кросс-валидации.

Для всех алгоритмов, кроме наивного байесовского классификатора, лучшими признаками оказались комбинации 1-грамм + 2-грамм, 1-грамм + 2-грамм + 3-грамм, а также униграммы. Для всех алгоритмов, кроме метода опорных векторов, применение алгоритма SMOTE дало заметное улучшение качества классификации для данных наборов n-грамм.

Наивный байесовский классификатор показал низкую эффективность на несбалансированных данных. Значение F-меры около 0,3 для классификатора означает, что алгоритм почти всегда предсказывает мажоритарный класс (положительный) и практически никогда остальные классы. Использование алгоритма SMOTE оказалось необходимым для корректной работы наивного байесовского классификатора.

Биграммы, триграммы и их комбинации показали худшие результаты, что можно объяснить низкой частотой словосочетаний в отзывах.

Лучшее качество классификации F-score = 0,64126 показал алгоритм логистическая регрессия. Увеличение количества признаков за счет использования биграмм и использование алгоритма SMOTE для решения проблемы несбалансированных классов данных увеличили точность базового алгоритма логистической регрессии на 0,032.

Выводы

По результатам работы для решения задачи автоматического определения тональности русскоязычных отзывов на рестораны была выбран алгоритм логистическая регрессия, обученный на комбинации униграмм и биграмм и улучшенный с помощью алгоритма SMOTE. В ходе выполнения работы были решены следующие задачи:

1. Написана программа для парсинга отзывов. Сформирован корпус русскоязычных отзывов на рестораны.
2. Данные корпуса были предобработаны для повышения качества классификации.
3. Построены векторные модели отзывов на основе меры TF-IDF и набора униграмм, биграмм, триграмм и их комбинаций.
4. Реализованы следующие алгоритмы машинного обучения: наивный байесовский классификатор, логистическая регрессия, AdaBoost классификатор, метод стохастического градиента и метод опорных векторов.
5. Решена проблема несбалансированных классов данных с помощью применения алгоритма SMOTE к обучающей выборке.
6. Реализована система перекрестной проверки алгоритмов и векторных моделей. Выбран алгоритм логистическая регрессия, показавший лучшее качество классификации F-score = 0,64126.

Причиной невысокой точности полученной модели может выступать несовершенство обучающих данных, в частности из-за схожести нейтральных отзывов с положительными. Для улучшения качества данных можно собрать дополнительный объем нейтральных отзывов, а также провести маркировку датасета вручную.

Заключение

В результате данной работы была разработана система выбора моделей для задачи многоклассовой классификации текстов. Для решения задачи автоматического определения тональности русскоязычных отзывов на рестораны был выбран алгоритм логистическая регрессия. Качество классификации было улучшено с помощью добавления биграмм и применения алгоритма SMOTE к обучающему множеству.

В качестве направления для дальнейших исследований может выступать анализ тональности отзывов на уровне аспектов. Например, для отзывов на рестораны имеет смысл узнать тональность аспектов: еда, интерьер и сервис.

Список литературы

1. Askalidis, Y., and Malthouse, E. C. (2016). vv RecSys 2016 - Proceedings of the 10th ACM Conference on Recommender Systems, 155-158. Association for Computing Machinery, Inc.
2. Al-Tit, Ahmad. (2015). The Effect of Service and Food Quality on Customer Satisfaction and Hence Customer Retention. Asian Social Science. 11. 129-139. 10.5539/ass.v11n23p129.
3. https://www.brightlocal.com/research/local-consumer-review-survey/
4. Baccianella, Stefano & Esuli, Andrea & Sebastiani, Fabrizio. (2010). SentiWordNet 3.0: An Enhanced Lexical Resource for Sentiment Analysis and Opinion Mining. Proceedings of LREC. 10.
5. Loukachevitch, N.,Levchik,A.: Creating a generalRussian sentiment lexicon. In: Proceedings of Language Resources and Evaluation Conference, LREC 2016, pp. 1171–1176 (2016)
6. Kotelnikova A.V., Kotelnikov E.V. SentiRusColl: Russian Collocation Lexicon for Sentiment Analysis // 8th conference on Artificial Intelligence and Natural Language (AINL-2019). Tartu, Estonia.
7. Pang, Bo & Lee, Lillian & Vaithyanathan, Shivakumar. (2002). Thumbs up? Sentiment Classification Using Machine Learning Techniques. EMNLP. 10. 10.3115/1118693.1118704.
8. Yu, Boya & Zhou, Jiaxu & Zhang, Yi & Cao, Yunong. (2017). Identifying Restaurant Features via Sentiment Analysis on Yelp Reviews.
9. O. Sharif, M. M. Hoque and E. Hossain, "Sentiment Analysis of Bengali Texts on Online Restaurant Reviews Using Multinomial Naïve Bayes," 2019 1st International Conference on Advances in Science, Engineering and Robotics Technology (ICASERT), Dhaka, Bangladesh, 2019, pp. 1-6, doi: 10.1109/ICASERT.2019.8934655.

10. Turney, Peter. (2002). Thumbs Up or Thumbs Down? Semantic Orientation Applied to Unsupervised Classification of Reviews. Computing Research Repository - CORR. 417-424. 10.3115/1073083.1073153.
11. Putri, Indiati & Kusumaningrum, Retno. (2017). Latent Dirichlet Allocation (LDA) for Sentiment Analysis Toward Tourism Review in Indonesia. Journal of Physics: Conference Series. 801. 012073. 10.1088/1742-6596/801/1/012073.
12. Devlin, Jacob & Chang, Ming-Wei & Lee, Kenton & Toutanova, Kristina. (2018). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding.
13. Yang, Zhilin et al. “XLNet: Generalized Autoregressive Pretraining for Language Understanding.” NeurIPS (2019).
14. https://habr.com/ru/company/ods/blog/458928/
15. M. G. Sousa, K. Sakiyama, L. d. S. Rodrigues, P. H. Moraes, E. R. Fernandes and E. T. Matsubara, "BERT for Stock Market Sentiment Analysis," 2019 IEEE 31st International Conference on Tools with Artificial Intelligence (ICTAI), Portland, OR, USA, 2019, pp. 1597-1601, doi: 10.1109/ICTAI.2019.00231.
16. X. Gong, J. Jin and T. Zhang, "Sentiment Analysis Using Autoregressive Language Modeling and Broad Learning System," 2019 IEEE International Conference on Bioinformatics and Biomedicine (BIBM), San Diego, CA, USA, 2019, pp. 1130-1134, doi: 10.1109/BIBM47256.2019.8983025.
17. C. P. Chen and Z. Liu, “Broad learning system: an effective and efficient incremental learning system without the need for deep architecture,” IEEE transactions on neural networks and learning systems, vol. 29, no. 1, pp. 10–24, 2018.
18. Bing Liu. Sentiment Analysis and Opinion Mining, Morgan & Claypool Publishers, May 2012. , pp. 58-60.

19. V.Rybakov, A.Malafeev, “Aspect-Based Sentiment Analysis of Russian Hotel Reviews“, Supplementary Proceedings of the 7th International Conference on Analysis of Images, Social Networks and Texts (AIST-SUP 2018), Moscow, Russia, July 5-7, 2018, pp. 75-84
20. Nobre, Guilherme et al. “BooViews : Aspect-based Sentiment Analysis on Product Reviews combining SVM and CRF in Portuguese.” (2016).
21. https://emojipedia.org/faq/
22. T. LeCompte and J. Chen, "Sentiment Analysis of Tweets Including Emoji Data," 2017 International Conference on Computational Science and Computational Intelligence (CSCI), Las Vegas, NV, 2017, pp. 793-798, doi: 10.1109/CSCI.2017.137.
23. S. Wankhede, R. Patil, S. Sonawane and P. A. Save, "Data Preprocessing for Efficient Sentimental Analysis," 2018 Second International Conference on Inventive Communication and Computational Technologies (ICICCT), Coimbatore, 2018, pp. 723-726, doi: 10.1109/ICICCT.2018.8473277.
24. https://ru.wikipedia.org/wiki/TF-IDF
25. Jones K. S. A statistical interpretation of term specificity and its application in retrieval (англ.) // Journal of Documentation : журнал. — MCB University: MCB University Press, 2004. — Vol. 60, no. 5. — P. 493-502. — ISSN 0022-0418.
26. Lango, Mateusz. (2019). Tackling the Problem of Class Imbalance in Multi-class Sentiment Classification: An Experimental Study. Foundations of Computing and Decision Sciences. 44. 151-178. 10.2478/fcds-2019-0009.
27. Galar, M., Fernandez, A., Barrenechea, E., Bustince, H., Herrera, F.: A reviewon ensembles for the class imbalance problem: bagging-, boosting-, and hybrid-based approaches. IEEE Transactions on

Systems, Man, and Cybernetics, Part C(Applications and Reviews), 42(4), 463-484 (2012)

28. https://basegroup.ru/community/articles/imbalance-datasets
29. Nitesh V. Chawla, Kevin W. Bowyer, Lawrence O. Hall, and W. Philip Kegelmeyer. 2002. SMOTE: synthetic minority over-sampling technique. J. Artif. Int. Res. 16, 1 (June 2002), 321–357.
30. Jurafsky, Daniel & Martin, James. (2019). Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition, pp. 56 – 92.
31. К. В. Воронцов. Машинное обучение (курс лекций) http://www.machinelearning.ru/wiki/images/5/53/Voron-ML-Lin-SG.pdf
32. J. Zhu, H. Zou, S. Rosset, T. Hastie. “Multi-class AdaBoost”, 2009.
33. К. В. Воронцов. Лекции по методу опорных векторов. 2007.
34. Документация Beautiful Soup http://wiki.python.su/Документации/BeautifulSoup
35. Воронцов К. В. Комбинаторный подход к оценке качества обучаемых алгоритмов. — Математические вопросы кибернетики / Под ред. О. Б. Лупанов. — М.: Физматлит, 2004. — Т. 13. — С. 5–36
36. Е. Соколов. Семинары по методам выбора моделей. 2014. http://www.machinelearning.ru/wiki/images/a/af/Sem06_model_selection.pdf