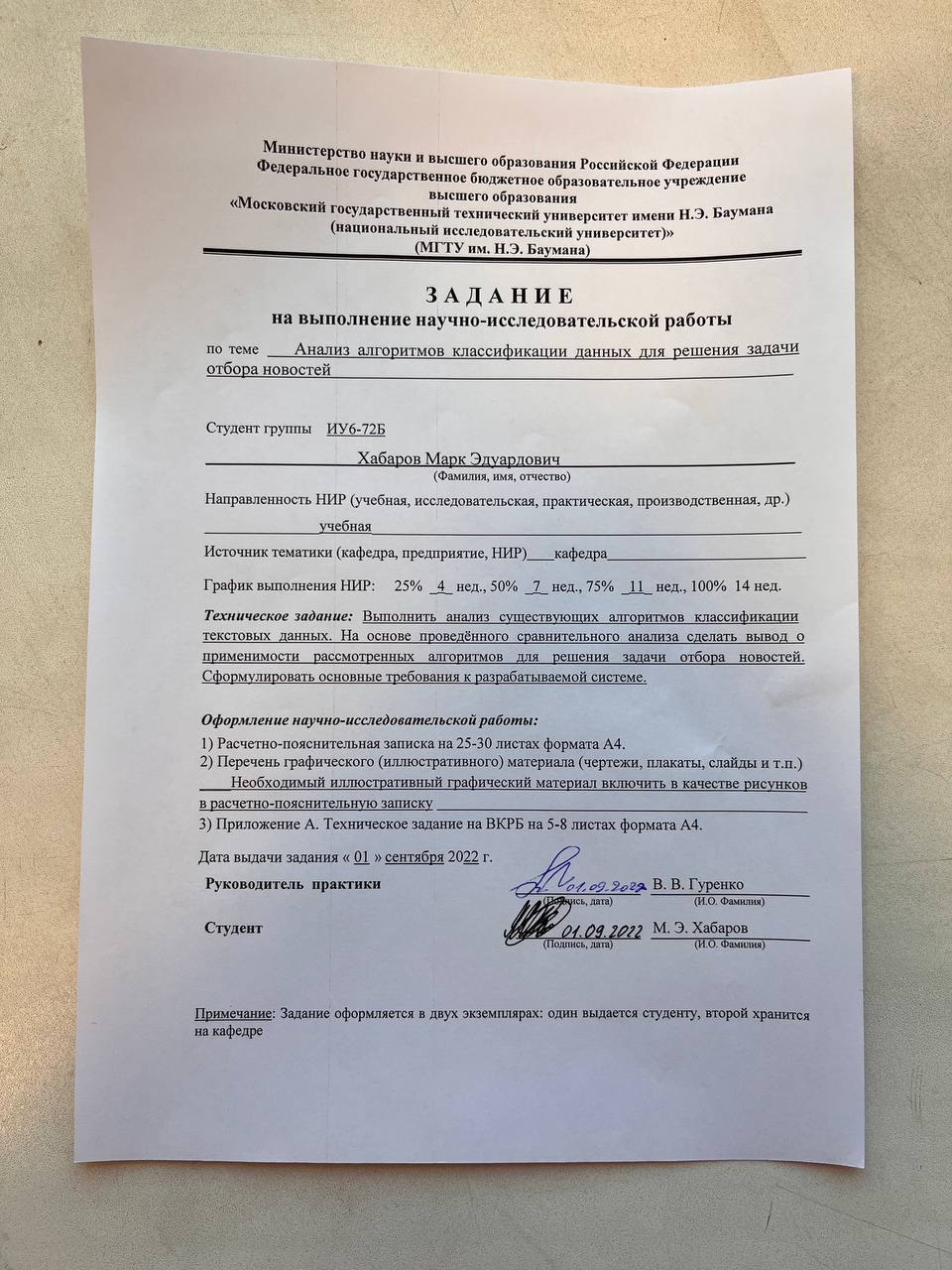


****

**РЕФЕРАТ**

НИР 33 с., 6 рис., 12 источников.

АЛГОРИТМЫ КлассификациИ, ОТБОР НОВОСТей, Классификация, ФЕЙК, ПРЕДОБРАБОТКА ДАННЫХ, СЛУЧАЙНЫЙ ЛЕС, ГРадиентное улучшение, Машинное обучение

В работе рассматриваются подходы к решению задачи отбора новостей с помощью алгоритмов классификации данных. Описан алгоритм разделения данных на коллекции с их последующей предобработкой, исследован метод векторизации текста, предназначенный для оценивания значимости слова в текстовом документе на основе данных о всём наборе документов.

Целью работы является изучение и последующее сравнительный анализ алгоритмов классификации текстовых данных применительно к решению задачи отбора новостей. Данная цель достигается рассмотрением различных методов классификации данных и выявлением их преимущественных качеств, способствующих решению поставленной задачи.

В результате проведённого исследования выбраны методы классификации, которые предлагается использовать для решения задачи отбора новостей, а также были выработаны требования к разрабатываемой системе.

**СОДЕРЖАНИЕ**

Термины и определения4

Введение5

1 Постановка задачи автоматического отбора новостей7

1.1 Математическая постановка задачи классификации текстов7

1.2 Разделение данных на коллекции8

2 Предварительная обработка и классификация новостных данных10

2.1 Общий подход к предобработке данных10

2.2 Аналитический обзор методов классификации данных12

2.2.1 Классификация методом k-ближайших соседей12

2.2.2 Классификация с помощью логистической регрессии14

2.2.3 Классификация методом случайного леса19

2.2.4 Классификация с помощью градиентного улучшения22

2.3 Подход к оценке качества работы классификатора24

2.4 Сравнительный анализ методов классификации28

3 Выработка требований к разрабатываемой системе30

Заключение31

Список использованных источников32

Приложение А. Техническое задание на выпускную квалификационную работу бакалавра (черновик)

**Определения и сокращения**

Гиперпараметр –характеристика алгоритма классификации, которая фиксируется заранее и не изменяется в процессе всего обучения.

Машинное обучение (от англ. Machine Learning, ML – машинное обучение) – методики анализа данных, основанные на использовании математических моделей и позволяющие аналитической системе обучаться в ходе решения множества схожих задач.

Фейк (от англ. fake – подделка, фальсификация, обман) – наименование изменённой, искажённой или заведомо ложной информации, представленной в виде текста, фото или видеофайлов.

NLP (от англ. Natural Language Processing – обработка естественного языка) – подраздел информатики, направленный на изучение методов анализа и синтеза естественного (человеческого) языка.

TF-IDF (от англ. Term Frequency - Inverse Document Frequency – частота слова – инверсия частоты слова в документах) – статистическая мера, используемая для оценки важности слова в контексте документа, являющегося частью коллекции документов.

**Введение**

Мировая глобализация, постоянное развитие информационных технологий и увеличение роли онлайн-СМИ, а также неуклонное возрастание пользователей Всемирной паутины приводят к тому, что люди всё чаще отдают предпочтение использованию новых социальных медиа ресурсов. С их помощью стало проще взаимодействовать с публикой, угадывать ее предпочтения и узнавать, что ей нравится. Непрерывный рост тех, кто предпочитает брать информацию из соцсетей и чатов, связан с удобством их использования, ведь они позволяют в любой момент времени получать оперативную информацию о произошедших событиях.

Новостные статьи интересны в любом возрасте, а понятие «новость» фигурирует практически в каждом разговоре, который мы можем услышать в повседневной жизни. Однако насколько сильно увеличивается популярность новых источников информации, настолько же сильно происходит уменьшение достоверности публикуемого материала. Всё чаще тиражируемая информация оказывается целенаправленно ложной и её становится всё сложнее разглядеть среди мощного информационного потока, ведь её появление в медиа-пространстве неслучайно. С помощью откровенно ложных новостей, которые принято называть фейковыми, активно продвигаются необходимые авторам сведения и осуществляется манипулирование общественным мнением.

Чтобы решить проблему фейковых новостей, важно сначала понять, что это такое. Под фейковыми новостями понимают контент, созданный с целью ввести читателей в заблуждение и тем самым осуществлять манипулирование общественным мнением [1].

Также не стоит забывать о таком понятии, как феномен постправды, когда люди верят в то, во что хотят верить, а непрерывный информационный поток лишь усиливает этот эффект, вынуждая многих людей верить не доказанным фактам, а наиболее популярной информации. Тем самым подтверждаются выводы исследований, которые были проведены учеными разных стран и показали, что люди плохо распознают фальшивые новости. Лишь 14% респондентов смогли определить фейковую информацию с вероятностью выше обычной случайности [2].

Актуальность выбранной темы заключается в том, что проблема фейковых новостей привлекла большое внимание ещё в 2007 году, когда была опубликована разоблачительная информация о всемирно известных издательствах, которые не брезговали фабрикацией фактов. Однако угроза сфальсифицированной информации остаётся открытой в информационной повестке и в настоящее время.

**1 Постановка задачи автоматического отбора новостей**

В рамках данной работы рассматривается задача автоматического отбора новостей, осуществляемого с помощью алгоритма классификации данных.

**1.1 Математическая постановка задачи классификации текстов**

Основываясь на различных научных работах [1], можно выделить одну из важнейших отличительных черт подобных новостных статей – это лексико-стилистические особенности текста, то есть броские заголовки, оценочная лексика, частое использование эмоционально окрашенных слов и сатиры, кроме того, допускаются опечатки.

Также для того чтобы найти пути решения поставленной задачи классификации, необходимо определить, что объединяет все фейковые новости. Все они предназначены для манипулирования эмоциями людей с целью управления обществом. Именно поэтому модели обработки естественного языка (NLP) и машинного обучения могут выявлять закономерности в них.

Классификация данных – задача разделения набора данных на классы путём назначения метки каждому конкретному наблюдению из предметной области [3]. Задача классификации текстовой информации, которая осуществляется на основе семантического (смыслового) анализа текста, является одной из самых сложных задач NLP и компьютерной лингвистики. Вся сложность заключается в том, чтобы «научить» модель как можно более правильно трактовать смысл, который передаёт автор текста.

Под классификатором понимают алгоритм, соотносящий некие входные данные с одним или несколькими классами. В отличие от алгоритмов кластеризации, эти классы должны быть определены заранее [3].

С целью формализованного подхода к решению поставленной задачи необходимо разработать математическую модель её предметной области.

Пусть – конечное множество новостей, – конечное множество категорий, Ф: – неизвестная целевая функция, которая для каждой пары <> определяет, соответствуют ли они друг другу, то есть принадлежит ли новость категории (1 или True) или нет (0 или False). [3]

Задача классификации состоит в построении классификатора Ф’: , максимально близкого к Ф.

Также важно упомянуть, что в рамках данной работы рассматривается задача «точной» классификации, то есть каждая новость относится только к одной категории.

**1.2 Разделение данных на коллекции**

Машинное обучение основывается на начальной коллекции новостей , где . При этом значение целевой функции Ф известно для каждой пары . Зачастую, для построения классификатора, Y разделяют на две непересекающиеся коллекции [3]:

* тренировочная (train-and-validation) . Коллекция новостей, с помощью которой создаётся классификатор Ф’. Ф’ обучается контролируемо (индуктивно);
* тестовая (test) . Коллекция новостей, на которой тестируется эффективность построенного классификатора. Каждая новость из тестовой коллекции подаётся на вход созданному классификатору, затем результат классификатора Ф’() сравнивается с известным значением Ф(). Чем больше количество таких совпадений, тем выше эффективность классификатора Ф’.

Следует отметить, что, для повышения эффективности классификатора, Y необходимо разбивать на три части, выделяя дополнительно «проверочную» (validation) коллекцию, с помощью которой будет осуществляться контроль процесса создания классификатора. Данная коллекция предназначена для обеспечения оптимизации настройки параметров.

После разделения Y на коллекции важно рассмотреть возможную проблему несбалансированности классов применительно к тренировочной выборке, ведь в подобной ситуации становится невозможно оценить действительную эффективность работы созданного классификатора.

Под несбалансированными (skewed) классами понимают такие классы, доли объектов которых существенно различаются [4].

Существует два основных способа решения задачи искажённых классов:

1. увеличение примеров миноритарного класса путём случайного дублирования его образцов;
2. сокращение числа образцов мажоритарного класса путём их исключения из коллекции.

Проблема первого подхода заключается в том, что он приводит к чрезмерному обучению, поскольку классификатор обучается на одних и тех же примерах. Поэтому, несмотря на то, что при удалении образцов мажоритарного класса теряются данные, которые могли бы быть использованы для повышения точности классификатора, второй способ является более предпочтительным к использованию.

**2 Предварительная обработка и классификация новостных данных**

**2.1 Общий подход к предобработке данных**

Ранее было упомянуто, что классификация текстов осуществляется на основе смыслового анализа информации. Чтобы извлечь из поступившего на вход объекта его семантическое значение, классификаторы, как правило, векторизуют текст, представляя его в виде вектора в пространстве признаков [5].

Для очистки текстового документа от возможных шумов, аномалий и прочих особенностей, которые могут помешать построению достоверного вектора, производится предобработка текста. Предобработка включает в себя следующие этапы [5]:

* токенизация – процесс разбиения текста на более мелкие части (токены), например, слова или предложения;
* удаление стоп-слов, то есть слов, которые могут иметь высокую частоту встречаемости в человеческом общении, но не придают никакой дополнительной ценности вектору и при этом добавляют много шума;
* приведение к нормальной форме. На этом шаге все формы слова приводятся к единому представлению.

Существует два основных подхода к нормализации:

1. стемминг – подход, при котором выделяется основа слова, дополнив которую можно получить слова-потомки. Данный подход считается устаревшим, так как обладает сравнительно высоким процентом некорректной нормализации, а его скорость работы оставляет желать лучшего;
2. лематизация – подход, разработанный с целью устранения недостатков стемминга путём использования структуры данных, в которой все формы слова связываются с его простейшей формой (леммой).

Дополнительно необходимо отметить, что может возникнуть ситуация, когда нельзя удалять стоп-слова, так как они или полезны для оценки семантики текста, или вовсе изменяют его смысл. По этой причине рекомендуется крайне осторожно подходить к выбору списка стоп-слов и, при возможности, полностью отказываться от данного этапа предобработки.

После предобработки текстовых данных необходимо выбрать алгоритм векторизации текста. Существует два вида подобных алгоритмов [5]:

* основанные на сопоставлении каждого слова в тексте с числовой характеристикой важности этого же слова;
* основанные на нейронных сетях.

Однако, следует учитывать, что в рамках поставленной задачи у алгоритмов, основанных на нейросетевой архитектуре, существует ряд недостатков, вынуждающих отказаться от их использования:

1. требуется определённая версия нейронной сети, предобученная на использование определённого естественного языка в рамках конкретной области с целью минимизировать количество ошибок семантического анализа;
2. в случае небольшого количества данных для обучения, подобные модели показывают более низкую точность, чем базовые алгоритмы без нейросетевой архитектуры.

Таким образом, было принято решения выбрать один из самых распространённых алгоритмов первого вида, основанный на TF-IDF.

TF-IDF предназначен для оценивания значимости слова в документе (тексте) на основе данных о всей коллекции документов. Первым компонентом этой меры является TF (Term Frequency) – количество вхождений терма (слова) в отношении к общему число слов в документе (тексте). TF определяется по следующей формуле [6]:

где – количество раз, которое термин встретился в документе, *S* – общее количество элементов в документе*.*

Второй компонент IDF (Inverse Document Frequency) представляет собой обратную частоту, с которой терм встречается в документах. IDF определяется по следующему соотношению [6]:

где – общий объём документов, *N* – количество документов, в которых встречается этот термин.

Результирующая формула TF-IDF выглядит следующим образом [6]:

.

Важно упомянуть, что TF-IDF решает дилемму об удалении стоп-слов за счёт уменьшения веса слов, которые часто используются в тексте и не несут в себе полезной нагрузки для модели.

В результате векторизации предварительно обработанных данных, текст представляется в виде мультимножества термов (слов) . Каждому терму сопоставлен некоторый вес – числовая характеристика важности этого слова в тексте . То есть, каждый текст имеет вид вектора весов его термов: .

**2.2 Аналитический обзор методов классификации данных**

В данной работе рассмотрены следующие методы классификации текста, а именно: k-ближайших соседей (k-nearest neighbours, KNN), логистическая регрессия (LogisticRegression, LR), случайный лес (RandomForest, RF) и градиентное улучшение (XGBoost) [7].

**2.2.1 Классификация методом k-ближайших соседей**

Метод k-ближайших соседей (KNN) является базовым, простым и одновременно одним из самых популярных методов классификации данных. После получения на вход какого-то объекта, алгоритм вычисляет расстояние до каждого из уже обработанных объектов и отбирает k из них, расстояние до которых минимально. Класс нового объекта – это класс, наиболее часто встречающийся среди k ближайших соседей. После чего, этот объект добавляется в выборку и в дальнейшем будет использоваться для классификации новых данных.

Следует отметить, что в качестве метрики расстояния чаще всего используются частные случаи метрики Минковского, формула которой выглядит следующим образом:

где – расстояние между точками, – показатель Минковского, – размерность пространства.

Возможные частные случаи метрики Минковского:

1. евклидова метрика . Задаёт расстояние, как длину прямой, соединяющий две заданные точки;
2. манхэттенская метрика . Является минимальной длиной пути из точки “x” в точку “y” при условии, что можно двигаться только параллельно осям координат;
3. метрика Чебышева . Предназначена для выбора наибольшего из расстояний между векторами по каждой координате.

С учётом того, что в решаемой задаче производится бинарная классификация, то логичнее всего выбрать евклидову метрику.

Также важной особенностью метода k-ближайших соседей является тот факт, что для корректной работы алгоритма k должно принадлежать множеству нечётных натуральных чисел.

*Преимущества метода:*

1. простота реализации;
2. ленивое обучение (Lazy Learning). То есть метод не требует обучения перед стартом работы, что делает KNN намного быстрее других алгоритмов классификации.

*Недостатки метода:*

1. при сравнительно высокой эффективности алгоритм имеет высокую стоимость прогнозирования для больших объёмов данных;
2. необходимо методом перебора подбирать оптимальное значение параметра k.

**2.2.2 Классификация с помощью логистической регрессии**

Логистическая регрессия позволяет оценивать апостериорные вероятности принадлежности объектов классам. Данный алгоритм применяется для прогнозирования вероятности возникновения некоторого события по значениям множества признаков. Основная идея заключается в том, что пространство исходных значений может быть разделено линейной границей на две соответствующих классам области. Эта граница задается в зависимости от имеющихся исходных данных и обучающего алгоритма.

В формульном виде логистическую регрессию можно представить следующим образом:

где – результат работы классификатора, – функция активации «сигмоида», – количество весов, характеризующих классифицируемый объект.

То есть сначала, взяв скалярное произведение между вектором весов и вектором признаков, получается число в интервале . Далее, к этому числу применяется функция активации «сигмоида», которая определяется по следующей формуле:

Задача сигмоиды брать число из и переводить его в число в интервале , характеризующее вероятность отнесения объекта к положительному классу. График функции активации представлен на рисунке 1.

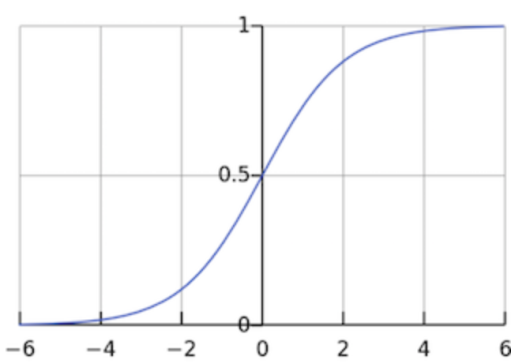


Рисунок 1 – Функция активации

Для корректной работы модели необходимо настроить значения вектора весов, которые подбираются за счёт оптимизации функции ошибки (loss-function),то есть за счёт нахождения таких “w”, чтобы .

Функция ошибки – это функция, предназначенная для сравнения реальных ответов (y) и ответов, предсказанных моделью () [8].

Как уже было сказано выше, в случае логистической регрессии результатом работы классификатора является число в диапазоне , представляющее собой вероятность отнесения объекта к положительному классу. Соответственно, в качестве функции ошибки необходимо применять функцию, указывающую то, насколько близка вероятность предсказания к соответствующему истинному значению. Такая функция называется логарифмической потерей (Log Loss) и имеет вид:

где *m* – длина выборки (количество элементов в обучающей коллекции).

Для оптимизации функции ошибки используется метод градиентного спуска. Под градиентным спуском понимается метод нахождения локального минимума функции ошибки. Основная идея метода – осуществление оптимизации функции в направлении наискорейшего спуска (антиградиента).

Эвристика градиентного спуска:

Пусть существует начальное значение функции ошибки, с которого начинается оптимизация. Необходимо найти направление антиградиента. Это реализуется с помощью определения частичных производных функции по каждому из весов в данной точке. Полученные производные показывают то, насколько каждый из параметров влияет на эту функцию, чем больше модуль производной, тем больше влияние этого веса. Соответственно, анализируя полученные производные, можно определять, как изменить каждый из весов.

Если производная отрицательная, то при увеличении веса функция уменьшается, и мы можем осуществить увеличение веса. Если производная положительная, то при уменьшении веса функция уменьшается, и мы можем осуществить уменьшение веса. Однако, необходимо контролировать величину изменения весов, так как если одномоментно осуществлять их значительное увеличение или уменьшение, то это может привести к постоянному изменению знака производной и алгоритм не сойдётся (то есть на бесконечном промежутке времени функция не будет оптимизирована). Если же одномоментное изменение весов будет слишком маленьким, то потребуется огромное количество итераций для оптимизации.

Поэтому, для контролирования уменьшений/увеличений весов вводится понятие скорости обучения (learning rate)**. –** это гиперпараметр, определяющий порядок того, как осуществляется корректировка весов с учётом функции ошибки в градиентном спуске.

Также важным гиперпараметром градиентного спуска является эпоха, представляющая собой одну итерацию, за которую вся обучающая выборка прошла через классификатор один раз, что привело к изменению весов [9].

Градиент функции в точке с координатами можно представить в виде формулы:

*(,…, ) =*

Необходимо отметить, что алгоритм базового градиентного спуска не лишён минусов, ведь на каждой эпохе требуется выполнить потенциально дорогую операцию вычисления градиента по всей выборке. Также в процессе оптимизации существует проблема застревания в локальных минимумах.

Поэтому всё чаще стандартному градиентному спуску предпочитают стохастический градиентный спуск (SGD). Он отличается от базового алгоритма тем, что обучающая коллекция разбивается на пакеты-батчи (от англ. batch). То есть, перед началом каждой эпохи происходит случайное перемешивание обучающей коллекции, после чего из неё подряд берутся «B» объектов, где «B» – определённый заранее размер пакетов. На основе извлечённого батча происходит определения функции ошибки L и обновление весов модели. Потом извлекается следующий пакет и происходит очередной перерасчёт весов. Так продолжается до момента, пока не закончится обучающая выборка. При этом функция ошибки приобретает вид следующей формулы:

где *В* – размер пакетов (батчей).

Таким образом, при использовании SGD шаг (изменение весов) происходит не один раз в эпоху, а несколько раз (в зависимости от количества пакетов, на которые был разбит исходный датасет).

Различия поиска минимума с помощью базового и стохастического градиентных спусков представлены на рисунке 2.

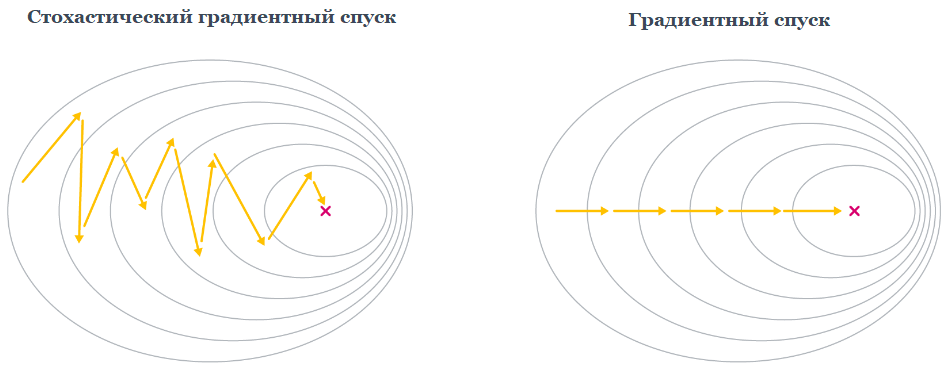


Рисунок 2 – Схождение базового градиентного спуска и SGD

Исходя из вышесказанного, можно сформулировать плюсы использования стохастического градиентного спуска:

* частота обновления параметров (весов) выше, чем в простом градиентном спуске, что позволяет обеспечить более надёжную сходимость, избегая локальных минимумов;
* скорость сходимости SGD намного менее зависима от выбора хорошей точки старта, чем у базового алгоритма;
* эффективное использование оперативной памяти, ведь при использовании пакетирования исчезает необходимость держать в памяти одновременно всю обучающую выборку;
* SGD является случайным по своей природе, в отличие от детерминированного базового градиентного спуска;
* процесс конвергенции происходит быстрее, чем у стандартного алгоритма.

*Преимущества метода:*

1. простота реализации;
2. обладает низкой склонностью к переобучению;
3. низкие требования к ресурсам в процессе обучения;
4. имеет относительно высокую скорость обучения в сравнении с остальными алгоритмами классификации.

*Недостатки метода:*

1. выбросы в данных могут сильно исказить результат работы;
2. поддерживает только линейные решения, что в определённых ситуациях может дать самые низкие метрики среди всех алгоритмов классификации.

**2.2.3 Классификация методом случайного леса**

Случайный лес (random forest) представляет собой универсальный алгоритм классификации и регрессии, суть которого состоит в использовании ансамбля деревьев принятия решений [10].

Дерево принятия решений (decision tree) – это инструмент интеллектуального анализа данных предсказательной аналитики, который помогает в решении задач классификации и регрессии. Дерево решений представляет собой иерархическую древовидную структуру, состоящую из правил вида «Если …, то …» [10].

Одной из главных проблем машинного обучения является ситуация, когда алгоритм представляет из себя “чёрный ящик”, внутреннее устройство и механизм работы которого сложны и неизвестны. При этом, для снижения рисков ошибочной работы алгоритма, в процессе решения бизнес-задач часто требуется увидеть то, чем руководствуется модель при принятии тех или иных решений. Именно для преодоления подобных ситуаций предназначены решающие деревья.

Главным требованием к дереву принятия решений является исчислимость условий и их простота. То есть условия в узлах должны быть унифицированы и иметь вид:

,

где с – некоторая константа, – вектор признаков.

Пример дерева решений с тремя листьями и глубиной 2 представлен на рисунке 3.

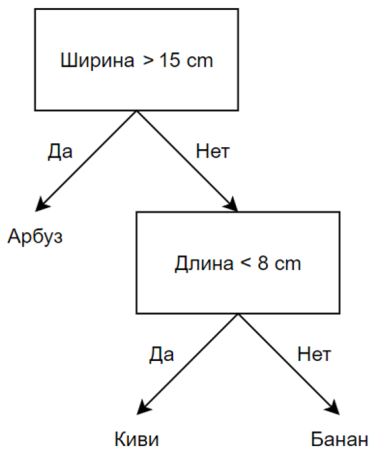


Рисунок 3 – Пример дерева решений

Оценка качества того, как узел решающего дерева разделяет данные на множества, осуществляется с помощью функции ошибки, зависящей от i-го признака, константы “c” и – множества всех данных, поступивших на вход. Такая функция ошибки определяется по следующему соотношению:

где – левое подмножество, – правое подмножество, *I* – критерий информативности*.*

Под критерием информативности понимается метрика, которая оценивает качество распределения целевой переменной среди объектов множества. Чем она ниже, лучше качество распределения.

В задаче классификации, где лист представляет собой разбиение множества по классам, информативность считается, как критерий (загрязнение) Джини, форма которого выглядит следующим образом:

,

где – доля объектов класса k в выборке Х.

Критерий Джини зависит от доли объектов класса k в выборке X, которая определяется по следующей формуле:

Таким образом, обучение дерева решений происходит по узлам, то есть модель, спускаясь от корня в глубину, будет для каждого узла по очереди, руководствуясь величиной , подбирать оптимальное условие вида .

Однако если не ограничивать глубину обучения, то деревья решений очень легко переобучаются, так как будут обучаться до того момента, пока окончательно не разделят все классы. Если же ограничить глубину на недостаточном значении, то качество классификации резко ухудшается.

Поэтому, для повышения эффективности работы классификатора, принято использовать именно алгоритм Random forest, то есть набор деревьев, которые объединены в одну модель на основе бутстреппинга (bootstraping, самозагрузка).

Суть начальной агрегации заключается в том, что на вход случайному лесу подаётся обучающая выборка размера N. После чего, каждое дерево из ансамбля моделей получает для обучения коллекцию, которая представляет собой случайную выборку размера N (с заменой) из первоначального датасета.

Также важной отличительной чертой случайного леса является случайность признаков для каждого из деревьев. То есть, если в обычном дереве для каждого узла подбирается оптимальное условие по наиболее подходящему параметру, то в random forest каждое дерево может выбирать наиболее подходящий параметр не из всех доступных, а только из их случайного подмножества. Это увеличивает диверсификацию и уменьшает корреляцию между деревьями.

После обучения всех деревьев выбор финального класса происходит по методу голосования (voting method), то есть выбирается тот класс, который был предсказан большинством деревьев.

*Преимущества метода:*

1. простота реализации;
2. низкие затраты ресурсов на обучение деревьев из ансамбля, так как это возможно осуществлять параллельно;
3. высокая скорость обучения из-за параллелизма;
4. низкая корреляция между обученными деревьями, то есть деревья “защищают” друг друга от своих индивидуальных ошибок;
5. обладает нелинейной границей принятия решения, что способствует принятию моделью нестандартных решений, а также позволяет с высокой точностью разделять данные на классы.

*Недостатки метода:*

1. обладает высокой склонностью к переобучению;
2. проблема выбора количества деревьев в ансамбле, которое позволит получить максимальную точность классификатора.

**2.2.4 Классификация с помощью градиентного улучшения**

Градиентное улучшение (gradient boosting) представляет собой алгоритм классификации и регрессии, реализованный на основе идеи «продолжай учиться» с помощью ансамбля деревьев принятия решений. Суть этой идеи заключается в том, что каждая последующая модель ансамбля обучается на неточностях предыдущей, пытаясь их исправить [11].

Для начала работы алгоритма классификации неслучайного леса необходимо получить одинаковое для всех экземпляров первичное (базовое) предсказание (Prediction), определяемое по следующему соотношению:

где – количество экземпляров позитивного класса в обучающей выборке, – количество экземпляров негативного класса в обучающей выборке.

После этого на основе полученного базового предсказания определяется вероятность (Probability) отнесения каждого из экземпляров к положительному классу по формуле. Вероятность определяется следующим образом:

*Probability =*

Далее для каждого экземпляра данных из обучающей выборки производится взвешивание, то есть определяется разница реальной вероятности (для положительного класса = 1, а для отрицательного = 0) и вероятности Probability.

Полученные взвешенные данные подаются на вход дереву решений, на основе результатов работы которого осуществляется перерасчёт весов.

Далее алгоритм повторяется, и веса будут подаваться на вход новым деревьям до тех пор, пока ошибка обучения не опустится ниже определённого порога.

Пример схемы градиентного улучшения, основанного на ансамбле из 3-х моделей, представлен на рисунке 4.

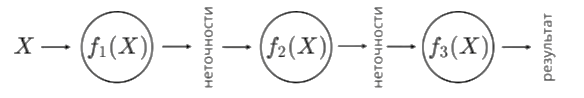


Рисунок 4 – Пример схемы градиентного улучшения

*Преимущества метода:*

1. простота реализации;
2. в данном методе, в отличие от случайного леса, не обучается большое количество примерно одинаковых деревьев, а идёт уклон именно на ошибки предыдущей модели. Поэтому точность (эффективность) таких моделей выше.

*Недостатки метода:*

1. достаточно высокие затраты ресурсов и времени на обучение;
2. в отличие от случайного леса, модели бустинга сильно переобучаются, так как random forest благодаря механизму бутстреппинга может хорошо справляться с выбросами в данных.

**2.3 Подход к оценке качества работы классификатора**

Прежде чем приступить к выбору классификатора, необходимо определить набор метрик, по которым будет происходить оценивание эффек-

тивности той или иной модели классификации.

Метрика – это качественный или количественный показатель, отражающий уровень успешности классификатора [12]. Наиболее распространённой метрикой является аккуратность (Accuracy), представляющая собой долю правильных ответов и определяющаяся по формуле:

Однако метрика Accuracy даёт ошибочную оценку в ситуации несбалансированных классов, методы решения которой описывались в п.1.2.

Поэтому, с целью обеспечения корректности оценки модели, несмотря на необходимость недопущения искажённости классов, вводятся дополнительные метрики.

Первая из них – матрица ошибок, вид которой показан на рисунке 5. В строках матрицы отображается “фактический класс”, а в столбцах – “прогнозируемый класс”.

Из этой матрицы следует четыре показателя [12]:

* истинно-положительный (TP) – количество прогнозов, когда классификатор верно спрогнозировал положительный класс;
* истинно-отрицательный (TN) – количество прогнозов, когда классификатор верно спрогнозировал отрицательный класс;
* ложноположительный (FP) – количество прогнозов, когда классификатор ошибочно спрогнозировал положительный класс (ошибка первого рода);
* ложноотрицательный (FN) – количество прогнозов, когда классификатор ошибочно спрогнозировал отрицательный класс (ошибка второго рода).

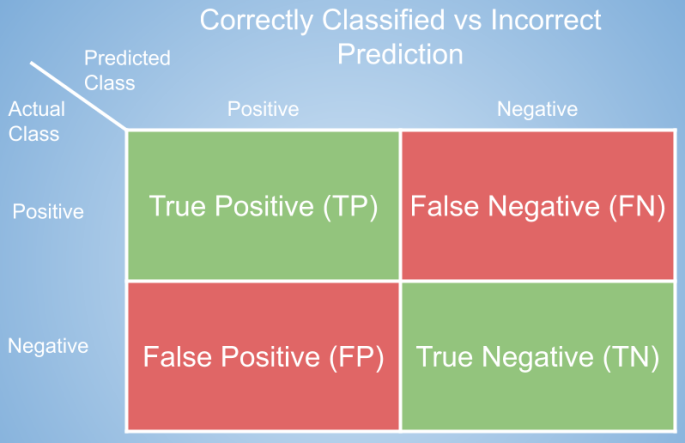


Рисунок 5 – Матрица ошибок

На основе этих четырёх показателей формируются метрики:

* точность (Precision) **–** эта метрика связана с ошибкой первого рода и показывает долю реальных положительных документов среди всех, которые классификатор счёл положительными. Формула точности выглядит следующим образом [12]:
* охват/полнота (Recall) **–** эта метрика связана с ошибкой второго рода и показывает долю положительных документов, которые нашёл классификатор, относительно всех документов этого класса. Охват определяется по следующему соотношению [12]:

Несмотря на то, что для полноты картины качества работы классификатора используются обе метрики, необходимо иметь в виду, что их важность различается в зависимости от решаемой задачи. Так, при отборе новостей охват имеет более высокую значимость, так как показывает, какую часть из действительно фейковых новостей в тестовой коллекции модель определила верно (в идеале модель должна охватить все фальсифицированные новости). А точность показывает то, сколько среди тех новостей, которые модель определила, как фейк, на самом деле фальшивые. Проще говоря, при решении задачи отбора новостей будет лучше, если классификатор пометит истинную новость как фейк, чем наоборот.

Также для оценки работы классификатора часто используют метрику F1, которая по своей сути является средним гармоническим точности и охвата и предназначена для их контроля и поиска оптимального баланса между ними. Метрика F1 определяется по следующей формуле [12]:

Ещё одной важной метрикой, с помощью которой визуализируют результат работы классификатора на тестовой выборке, является ROC-AUC (площадь под ROC-кривой).

ROC-кривая – это график, позволяющий оценить качество бинарной классификации и отображающий соотношение между долей объектов от общего количества носителей признака, верно классифицированных как несущие признак, и долей объектов от общего количества объектов, не несущих признака, ошибочно классифицированных как несущие признак [12]. Пример ROC-кривой представлен на рисунке 6.

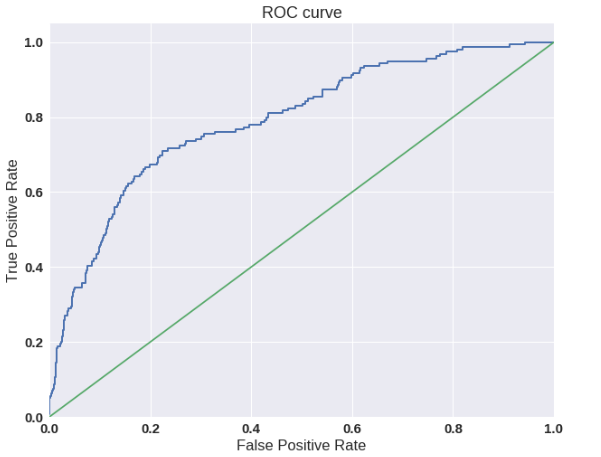


Рисунок 6 – Пример ROC-кривой

Осями ROC-кривой являются упомянутый ранее охват, который ещё называют истинно положительной ставкой (True Positive Rate, TPR), и ложноположительная ставка (False Positive Rate, FPR), которая показывает, какую долю из объектов отрицательного класса алгоритм предсказал неверно. Формула ложноположительной ставки выглядит следующим образом [12]:

На основании определений ставок TRP и FPR можно утверждать:

1. в идеальном случае, когда классификатор работает безошибочно (TPR = 1, FPR = 0), мы получаем площадь под кривой, равную единице;
2. в случае, когда классификатор выдаёт правильный ответ с обычной случайности, площадь будет стремиться к 0.5, а ROC-кривая выродится в прямую под углом 45°, так как алгоритм будет выдавать примерно одинаковое количество TP и FP прогнозов;
3. в случае, если была допущена ошибка в процессе обучения, информация в обучающей коллекции была испорчена или же предобработка данных была выполнена некорректно, площадь будет стремиться к нулю.

Вместе с ROC-кривой принято использовать J-статистику Юдена, также называемую индексом Юдена [13]. Он представляет собой разницу между истинно положительной ставкой и ложноположительной ставкой и определяется по следующему соотношению:

По сути, индекс Юдена можно описать как вертикальное расстояние от прямой, проходящей под углом 45°, и точкой на ROC-кривой. То есть J-статистика показывает то, насколько ROC-кривая близка к идеалу, и чем больше индекс, тем лучше работает классификатор.

**2.4 Сравнительный анализ методов классификации**

В результате изучения и анализа методов классификации данных можно сделать следующие выводы об их применимости для решения задачи отбора новостей.

1. Метод k-ближайших соседей обладает необходимостью подбирать оптимальное значение параметра k, что увеличивает риск влияния человеческого фактора на эффективность обучения модели. Также метод k-ближайших соседей является самым худшим вариантом для прогнозирования в реальном времени, когда важна скорость выдачи результата пользователю, так как при увеличении количества данных возрастает время прогнозирования.

2. Логистическая регрессия предполагает отсутствие выбросов в данных, которые могут сильно исказить результат работы. Кроме того, логистическая регрессия поддерживает только линейную границу решения, поэтому есть риск возникновения ситуации, когда из-за линейной неразделяемости данных будет получен высокий уровень ошибки модели.

3. Метод случайного леса является одним из самых мощных и точных классификаторов, который одинаково хорошо справляется как с линейными, так и с нелинейными задачами. Также случайный лес обладает высокой скоростью обучения за счёт параллелизма и хорошо справляется с выбросами в данных за счёт низкой корреляции деревьев ансамбля. Однако данная модель обладает высокой склонностью к переобучению, для борьбы с которым необходимо вручную задавать количество деревьев в ансамбле и их максимальную глубину.

4. Модель градиентного улучшения по своей эффективности не уступает случайному лесу, но обладает большей склонностью к переобучению за счёт выбросов данных. Также градиентное улучшение требует больше ресурсов для обучения, нежели алгоритм случайного леса. Однако для устранения большинства недостатков неслучайного леса можно использовать библиотеку XGBoost (экстремальное градиентное улучшение), которая призвана бороться с переобучением за счёт регуляризации, а также предусматривает возможность распараллеливания в процессе обучения.

Таким образом, для отбора новостей целесообразно выполнить обучение классификаторов случайного леса и градиентного улучшения (XGBoost). В процессе обучения для каждой модели в рамках решаемой задачи следует получить метрики оценки качества работы, описанные в разделе 2.3.

Окончательный выбор классификатора необходимо осуществлять на основании полученных показателей эффективности моделей.

**3 Выработка требований к разрабатываемой системе**

На основании проведенного анализа можно сформулировать основные требования к разрабатываемой рекомендательной системе:

* должна быть реализована функция получения текста новости от пользователя;
* должен быть реализован общий подход к предобработке полученной информации, включающий в себя токенизацию, удаление стоп-слов, приведение к нормальной форме, а также векторизацию с помощью алгоритма TF-IDF;
* анализ полученной от пользователя новости должен осуществляться одним из двух методов: случайный лес или градиентное улучшение. При этом окончательный выбор модели необходимо осуществить на основании полученных в процессе практической реализации показателей эффективности: аккуратность, точность, охват, F1, индекс Юдена;
* система должна выполнять функцию рекомендации, можно ли доверять полученному контенту, выдавая пользователю оценку истинности или ложности с вероятностью отнесения информации к одному из классов;
* необходима функция поддержки русского языка.

Предметной областью разрабатываемой системы являются новостные статьи из области медицины, а потенциальными потребителями – пользователи, объект интереса которых представляет собой новость в сфере медицины, в истинности которой уверенности нет.

Для работы с системой пользователь должен обладать базовыми навыками использования мессенджера Telegram – кроссплатформенного приложения для мгновенного обмена сообщениями.

Для проектирования классификатора, представляющего собой основу рекомендательной системы, необходим набор данных на русском языке, который должен иметь вид [текст новости: истинность], где истинность принимает значения 0 (ложная новость) или 1 (истинная новость).

**Заключение**

В ходе научно-исследовательской работы были проанализированы основные алгоритмы, применимые для классификации текстовых данных. На основе выявленных преимуществ и недостатков каждого из классификаторов были выбраны те, которые лучше всего подходят для решения задачи отбора новостей. Для сравнения отобранных классификаторов с целью выявления того из них, который наиболее подходит для решения поставленной задачи, были выбраны метрики оценки качества работы классификатора.

В результате выполнения работы сформулированы требования к разрабатываемой системе.

**Список использованных источников**

1. Зуйкина К.Л., Соколова Д.В., Особенности идентификации фейковых новостей молодёжной аудиторией [Электронный ресурс]. URL: https://cyberleninka.ru/article/n/osobennosti-identifikatsii-feykovyh-novostey-molodezhnoy-auditoriey/viewer (дата обращения 09.12.2022).

2. Fake News on Social Media: People Believe What They Want to Believe When it Makes No Sense at All [Электронный ресурс]. URL: https://papers.ssrn.com/sol3/papers.cfm?abstract\_id=3269541 (дата обращения 09.12.2022).

3. Лифшиц Ю., Автоматическая классификация текстов // Лекция №6 курса Алгоритмы для Интернета [Электронный ресурс]. URL: https://logic.pdmi.ras.ru/~yura/internet/06ianote.pdf (дата обращения 09.12.2022).

4. Дисбаланс классов [Электронный ресурс]. URL: <https://alexanderdyakonov.wordpress.com/2021/05/27/imbalance/> (дата обращения 10.12.2022).

5. Текстовая классификация с TF-IDF, LSTM и BERT [Электронный ресурс]. URL: https://medium.com/@claude.feldges/text-classification-with-tf-idf-lstm-bert-a-quantitative-comparison-b8409b556cb3 (дата обращения 10.12.2022).

6. Текстовая классификация с использованием TF-IDF [Электронный ресурс]. URL: https://medium.com/swlh/text-classification-using-tf-idf-7404e75565b8 (дата обращения 11.12.2022).

7. Классификация в машинном обучении [Электронный ресурс]. URL: https://www.helenkapatsa.ru/klassifikatsiia/ (дата обращения 12.12.2022).

8. Функция потерь (Loss Function) [Электронный ресурс]. URL: https://www.helenkapatsa.ru/funktsiia-potieri/ (дата обращения 12.12.2022).

9. Эпоха (Epoch) [Электронный ресурс]. URL: https://www.helenkapatsa.ru/epokha/ (дата обращения 13.12.2022).

10. Случайный лес (Random Forest) [Электронный ресурс]. URL: https://www.helenkapatsa.ru/sluchainyi-lies/ (дата обращения 13.12.2022).

11. Градиентный бустинг (GB) [Электронный ресурс]. URL: https://www.helenkapatsa.ru/gradiientnyi-bustingh-2/ (дата обращения 14.12.2022).

12. Метрики в задачах машинного обучения [Электронный ресурс]. URL: https://habr.com/ru/company/ods/blog/328372/ (дата обращения 15.12.2022).

13. Корнеенков А.А., Рязанцев С.В., Вяземская Е.Э. Вычисление и интерпретация показателей информативности диагностических медицинских технологий. Медицинский Совет. 2019 [Электронный ресурс]. URL: https://www.med-sovet.pro/jour/article/view/4387 (дата обращения 16.12.2022).