



Министерство науки и высшего образования Российской Федерации  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«Московский государственный технический университет  
имени Н.Э. Баумана  
(национальный исследовательский университет)  
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ «Информатика и системы управления»

КАФЕДРА «Программное обеспечение ЭВМ и информационные технологии»

## РАСЧЕТНО-ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

*К НАУЧНО-ИССЛЕДОВАТЕЛЬСКОЙ РАБОТЕ*

*НА ТЕМУ:*

*«Анализ основных методов решения задачи  
классификации текстов»*

Студент ИУ7И-576  
(Группа)

Я. Тэмүүжин  
(Подпись, дата)

(И.О.Фамилия)

Руководитель

К. А. Кивва  
(Подпись, дата)

(И.О.Фамилия)

2022 г.

**Министерство науки и высшего образования Российской Федерации  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования**

**«Московский государственный технический университет имени Н.Э. Баумана  
(национальный исследовательский университет)  
(МГТУ им. Н.Э. Баумана)**

---

УТВЕРЖДАЮ

Заведующий кафедрой ИУ-7

И. В. Рудаков  
« \_\_\_\_\_ » \_\_\_\_\_ 20 \_\_\_\_ г.

## **З А Д А Н И Е на выполнение научно-исследовательской работы**

по теме Анализ основных методов решения задачи классификации текстов

Студент группы ИУ7И-576

Тэмүүжин Янжинлхам

(Фамилия, имя, отчество)

Направленность НИР (учебная, исследовательская, практическая, производственная, др.)

учебная

Источник тематики (кафедра, предприятие, НИР) НИР

График выполнения НИР: 25% к 4 нед., 50% к 7 нед., 75% к 11 нед., 100% к 14 нед.

**Техническое задание Рассмотреть этапы процесса классификации текстов. Изучить существующие подходы и основные методы решения задачи классификации текстов.**

### ***Оформление научно-исследовательской работы:***

Расчетно-пояснительная записка на 15-25 листах формата А4.

Перечень графического (илюстративного) материала (чертежи, плакаты, слайды и т.п.)

Презентация на 8-10 слайдах.

Дата выдачи задания « \_\_\_\_\_ » \_\_\_\_\_ 20 \_\_\_\_ г.

**Руководитель НИР**

\_\_\_\_\_  
(Подпись, дата)

**К. А. Кивва**

(И.О.Фамилия)

**Студент**

\_\_\_\_\_  
(Подпись, дата)

**Я. Тэмүүжин**

(И.О.Фамилия)

**Примечание:** Задание оформляется в двух экземплярах: один выдается студенту, второй хранится на кафедре.

# Оглавление

<b>Введение</b>	<b>2</b>
<b>1 Анализ предметной области</b>	<b>3</b>
1.1 Предобработка документов . . . . .	5
1.1.1 Извлечение термов (Term extraction) . . . . .	5
1.1.2 Взвешивание термов (Term weighting) . . . . .	5
1.2 Индексация . . . . .	6
1.3 Отбор термов (Feature Selection) . . . . .	8
1.4 Вывод . . . . .	9
<b>2 Классификация существующих методов</b>	<b>11</b>
2.1 Системы на основе правил (Rule-based systems) . . . . .	11
2.2 Системы на основе машинного обучения (Supervised machine learning based systems) . . . . .	12
2.2.1 Метод k ближайших соседей (K-Nearest Neighbor, KNN) .	12
2.2.2 Логистическая регрессия (Logistic Regression, LR) . . .	13
2.2.3 Метод Наивного Байеса (Naive Bayes, NB) . . . . .	16
2.2.4 Метод опорных векторов (Support Vector Machines, SVM)	18
2.2.5 Метод деревьев решений (Decision Tree, DT) . . . . .	20
2.2.6 Случайный лес (Random Forest, RF) . . . . .	24
2.2.7 Сравнение алгоритмов машинного обучения . . . . .	25
2.3 Гибридные системы . . . . .	26
2.4 Рассмотрение экспериментов . . . . .	27
<b>Заключение</b>	<b>32</b>
<b>Литература</b>	<b>33</b>

# Введение

Огромное количество информации скапливается в многочисленных текстовых базах, хранящихся в персональных компьютерах, в локальных и глобальных сетях. Рядовому пользователю становится все сложнее работать с большими объемами данных. Чтение объемных текстов, ручной поиск и анализ нужной информации в гигантских массивах текстовых данных малоэффективны. Для решения данной проблемы и автоматизации процессов получило развитие направление обработки естественного языка (Natural Language Processing), решающее задачи информационного поиска (information retrieval), машинного перевода (machine translation), извлечения информации (information extraction), классификации текстов (text classification) и др. [1].

Классификация текстов – одна из важнейших задач обработки естественного языка. Это процесс классификации текстовых строк или документов по различным категориям в зависимости от содержания строк. Классификация текстов имеет множество приложений, таких как анализ тональности, маркировка тем, обнаружение спама и обнаружение намерений.

**Цель данной научно-исследовательской работы** – провести обзор основных методов решения задачи классификации текстов.

Для достижения поставленной цели необходимо решить следующие задачи:

- изучить предметную область;
- изучить этапы процесса классификации текстов;
- изучить существующие подходы к решению задач классификации текстов;
- изучить основные методы решения задачи классификации текстов.

# 1 Анализ предметной области

Классификация текстов является важной частью обработки естественного языка (Natural Language Processing, NLP) – общее направление искусственного интеллекта и математической лингвистики, изучающее проблемы компьютерного анализа и синтеза текстов на естественных языках.

Текущие исследования классификации текстов направлены на улучшение качества представления текста и разработку высококачественных классификаторов. Процесс классификации текстов делится на следующие этапы [2].

1. Предобработка документов.
  - (a) Извлечение термов.
  - (b) Взвешивание термов.
2. Индексация документов.
3. Отбор термов.
4. Построение и обучение классификатора.
5. Классификация новых текстов обученным классификатором.

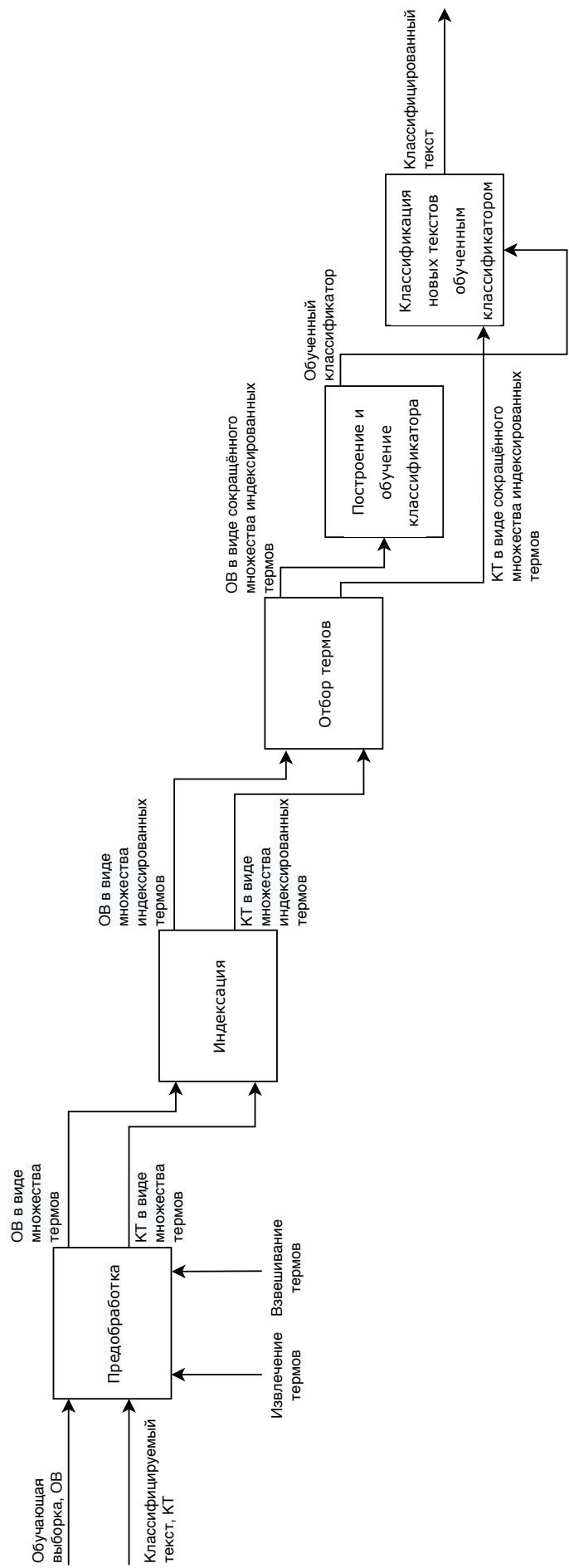


Рисунок 1.1 – Этапы процесса классификации текстов

## 1.1 Предобработка документов

### 1.1.1 Извлечение термов (Term extraction)

Извлечение термов, или извлечение признаков – это процесс разбиения текста на более простые объекты (термы) и выбора наиболее значимых термов. Результат данного процесса – это множество термов, которое дальше используется для индексации и получения весовых характеристик документа. Основными методами извлечения термов являются следующие.

1. **Токенизация** – это метод предварительной обработки, который разбивает поток текста на слова, фразы, символы или другие значимые элементы, называемые токенами [3].
2. **Удаление стоп-слов.**

Стоп-слова – это слова, не несущие какой-либо самостоятельной смысловой нагрузки. К стоп-словам относятся предлоги, союзы и местоимения. В целях уменьшения размерности пространства термов индексатор удаляет их при анализе.

3. **Лемматизация** – это процесс приведения слов к леммам, т. е. нормальным словесным формам [4]. В русском языке нормальными формами считаются:

- для существительных и прилагательных – именительный падеж, единственное число;
- для глаголов, причастий и деепричастий – глагол в неопределенный форме [5].

4. **Стемминг** – это отбрасывание изменяемых частей слов, главным образом, окончаний.

### 1.1.2 Взвешивание термов (Term weighting)

Взвешивание термов – это процесс определения значимости терма для выбранного документа. Существуют несколько способов определения веса при-

знаков документа. Наиболее распространенный – вычисление функции TF-IDF (Term Frequency-Inverse Document Frequency). Его основная идея состоит в том, чтобы больший вес получали слова с высокой частотой в пределах конкретного документа и с низкой частотой употреблений в других документах.

Вычисляется частота терма  $TF$  (term frequency) – оценка важности слова в пределах одного документа  $d$  по формуле

$$TF = \frac{n_{t,d}}{n_d}, \quad (1.1)$$

где  $n_{t,d}$  – количество употреблений слова  $t$  в документе  $d$ ,  $n_d$  – общее число слов в документе  $d$ .

Обратная частота документа  $IDF$  (inverse document frequency) – инверсия частоты, с которой слово встречается в документах коллекции. IDF уменьшает вес общеупотребительных слов, вычисляется по формуле

$$IDF = \log \left( \frac{|D|}{D_t} \right), \quad (1.2)$$

где  $|D|$  – общее количество документов в коллекции,  $D_t$  – количество всех документов, в которых встречается слово  $t$ .

Итоговый вес терма в документе относительно всей коллекции документов вычисляется по формуле

$$V_{t,d} = TF \cdot IDF. \quad (1.3)$$

Следует отметить, что по формуле оценивается значимость терма только с точки зрения частоты вхождения в документ, без учета порядка следования термов в документе и их лексической сочетаемости [6].

## 1.2 Индексация

Индексация документов – это построение некоторой числовой модели текста, которая переводит текст в удобное для дальнейшей обработки представление.

К основным моделям индексации относятся следующие.

### 1. *N*-граммы.

Объединение в группы N-грамм – это процесс объединения нескольких последовательных слов в одну группу, которую так же называют N-граммой. В таком случае, каждая N-грамма, рассматривается как самостоятельный терм документа.

### **Преимущества:**

- учитывается порядок слов;
- просто реализуемый.

### **Недостатки:**

- неустойчивость по отношению к выбросам в исходных данных;
- занимает большой объём памяти при больших  $n$ .

## **2. *Bag-of-words (BoW)*.**

Модель «мешка слов» позволяет представить документ в виде многомерного вектора слов и их весов в документе. Другими словами, каждый документ – это вектор в многомерном пространстве, координаты которого соответствуют номерам слов, а значения координат – значениям весов. В качестве веса часто рассматривается количество вхождений слова в данный документ.

### **Преимущества:**

- прост для понимания и реализации;
- занимает меньше места в памяти по сравнению с N-грамм.

### **Недостатки:**

- не учитывает порядок и семантику слов [7].

## **3. *Word2vec*.**

Word2vec принимает большой текстовый корпус в качестве входных данных и сопоставляет каждому слову вектор, выдавая координаты слов на выходе. Сначала он генерирует словарь корпуса, а затем вычисляет векторное представление слов, «обучаясь» на входных текстах. Векторное

представление основывается на контекстной близости: слова, встречающиеся в тексте рядом с одинаковыми словами (а следовательно, имеющие схожий смысл), будут иметь близкие (по косинусному расстоянию) векторы.

### **Преимущества:**

- изучение семантики и грамматической информации с учетом контекста слов;
- вектор занимает меньше места в памяти по сравнению с другими моделями.

### **Недостатки:**

- не обрабатывает слова, не входящие в словарь, то есть не встречающиеся в обучающей выборке [8].

## **1.3 Отбор термов (Feature Selection)**

Для эффективной работы классификатора часто прибегают к сокращению числа используемых признаков (термов). За счет уменьшения размерности пространства термов (Dimensionality Reduction) можно сократить вычислительные затраты, также снизить эффект переобучения – явление, при котором классификатор ориентируется на случайные или ошибочные характеристики обучающих данных, а не на важные и значимые [3].

К основным методам уменьшения размерности векторов относятся следующие.

1. *Метод главных компонент (Principal Component Analysis, PCA)* является простейшим методом уменьшения размерности в данных. Идея метода заключается в поиске в исходном пространстве гиперплоскости заданной размерности с последующим проектированием выборки на данную гиперплоскость. При этом выбирается та гиперплоскость, ошибка проектирования данных на которую является минимальной в смысле суммы квадратов отклонений.

### **Преимущества:**

- удаляет коррелированные признаки текста;
- высокое качество работы с меньшим количеством выборок на класс.

#### **Недостатки:**

- необходимость стандартизации данных;
- независимые переменные становятся менее интерпретируемыми [9].

2. *Линейный дискриминантный анализ (Linear Discriminant Analysis, LDA)* В отличие от метода PCA, в LDA производится попытка максимизировать разброс проекций векторов различных классов, минимизируя одновременно разброс проекций внутри классов.

#### **Преимущества:**

- использует все признаки для создания новой оси, максимизирует расстояние между классами двух переменных;
- высокая скорость работы и легкая интерпретируемость;
- высокое качество работы с большим количеством выборок на класс.

#### **Недостатки:**

- необходимость стандартизации данных [10].

## **1.4 Вывод**

В общем случае тексты и документы представляют собой неструктурированные наборы данных. Однако, для их эффективной и точной классификации неструктурированные текстовые последовательности должны быть преобразованы в структурированное пространство признаков при использовании математического моделирования. Во-первых, данные необходимо очистить, чтобы исключить ненужные символы и слова. Для этого используются методы извлечения термов такие, как токенизация, удаление стоп-слов, лемматизация и стемминг. После очистки данных для построения некоторой

числовой модели текста применяются методы взвешивание термов. Наиболее распространенным способом нахождения весов термов является вычисление функции TF-IDF. Самой эффективной моделью индексации является Word2vec. Среди способов уменьшения размерности пространства термов, в случае небольшого количества выборок на класс PCA работает лучше, а с большим набором данных, имеющим несколько классов, лучше работает LDA.

В данном разделе были описаны вышеперечисленные методы предварительной обработки текстов, их преимущества и недостатки.

## 2 Классификация существующих методов

Существует множество подходов к автоматической классификации текстов. Можно выделить 3 основные типы систем:

- системы на основе правил;
- системы на основе машинного обучения с обучением;
- гибридные системы.

### 2.1 Системы на основе правил (Rule-based systems)

Системы, основанные на правилах, классифицируют текст используя набор созданных вручную лингвистических правил. Эти правила предписывают системе использовать семантически релевантные элементы текста для определения релевантных категорий на основе его содержания. Каждое правило состоит из шаблона и прогнозируемой категории [11].

К примеру, чтобы разделить новостные статьи на две группы – Спорт и Политика – во-первых, нужно определить два списка слов, которые характеризуют каждую группу (например, слова, связанные со спортом, как футбол, баскетбол, Леброн Джеймс, Spartak и т. д., и слова, связанные с политикой, такие как правительство, Дональд Трамп, Путин и т.д.).

Затем, чтобы классифицировать новый входящий текст, нужно подсчитать количество слов, связанных со спортом, которые появляются в тексте, и сделать то же самое для слов, связанных с политикой. Если количество появлений слов, связанных со спортом, превышает количество слов, связанных с политикой, то текст классифицируется как «Спорт» и наоборот.

Например, эта система, основанная на правилах, классифицирует заголовок *«Когда первая игра Леброна Джеймса с "Лейкерс"?»* как «Спорт», так как найден один терм, связанный со спортом (Леброн Джеймс), и не нашлось термов, связанных с политикой.

Системы, основанные на правилах, понятны человеку и со временем могут быть улучшены. Но этот подход имеет некоторые недостатки. Во-первых, эти системы требуют глубоких знаний предметной области. Кроме того, они отнимают много времени, поскольку создание правил для сложной системы может

быть довольно сложной задачей и обычно требует большого объема анализа и тестирования. Системы, основанные на правилах, также сложно поддерживать и плохо масштабировать, учитывая, что добавление новых правил может повлиять на результаты ранее существовавших правил [3].

## 2.2 Системы на основе машинного обучения (Supervised machine learning based systems)

Большинство задач классификации текстов решается с помощью машинного обучения с учителем (Supervised Machine Learning, SML). Обучение с учителем использует обучающую выборку (Training Set) для обучения моделей получению правильного результата. Обучающая выборка данных включает в себя входные и правильные выходные данные, которые позволяют модели обучаться с течением времени.

В задачах классификации текстов с помощью SML обучающая выборка – набор из  $N$  документов, каждый из которых был вручную помечен классом:  $(d_1, c_1), \dots, (d_N, c_N)$ . При этом задача SML – разработать классификатор, способный найти класс  $c$ , к которому принадлежит документ  $d$ .

Среди алгоритмов SML, применяемых в рамках классификации текстовых данных, общераспространенными являются классификатор Роше, метод k ближайших соседей, логистическая регрессия, наивный байесовский классификатор, метод опорных векторов, деревья решений, случайный лес.

### 2.2.1 Метод k ближайших соседей (K-Nearest Neighbor, KNN)

Для нахождении категории, соответствующей документу  $d$ , классификатор сравнивает  $d$  со всеми документами из обучающей выборки  $T$ , то есть для каждого  $d_T \in T$  вычисляется расстояние  $\rho(d_T, d)$  – косинус угла между векторами признаков:

$$\rho(d_T, d) = \cos(d_T, d). \quad (2.1)$$

Далее из обучающей выборки выбираются  $k$  документов, ближайших к  $d$ . Согласно методу  $kNN$ , документ  $d$  считается принадлежащим тому классу,

который является наиболее распространенным среди соседей данного документа.

#### **Преимущества метода:**

- относительно простая программная реализация алгоритма;
- устойчивость алгоритма к линейно неразделимым данным.

#### **Недостатки метода:**

- большая длительность работы из-за необходимости полного перебора обучающей выборки;
- относительно низкое качество классификации;
- высокая зависимость результатов классификации от выбранной метрики.

kNN относительно плохо работает с длинными текстами по сравнению с короткими. В практике часто считается алгоритмом низкого качества, но из-за высокой интерпретируемости и простой реализации иногда используется в задачах классификации более формальных и структурированных документов [5].

### **2.2.2 Логистическая регрессия (Logistic Regression, LR)**

#### 1. Бинарная логистическая регрессия.

Для вектора признаков  $\vec{x} = \{x_1, x_2, \dots, x_n\}$  документа  $x$ , вывод классификатора  $y$  может принимать одно из двух значений – 1 (если документ принадлежит классу), 0 (иначе). Таким образом, задача классификатора сводится к нахождению вероятности того, что документ  $x$  принадлежит классу:

$$P(y = 1|x). \quad (2.2)$$

LR решает эту задачу, изучая вектор **весов (weights)**  $\vec{w}$  и **смещение (bias term)**  $b$  из обучающей выборки. Вес  $w_i$  признака  $x_i$  представляет,

насколько важен этот признак для решения о классификации, и может быть положительным или отрицательным.

После нахождения весов признаков  $w_i$  и смещения  $b$  на этапе обучения с помощью кросс-энтропийной потери и стохастического градиентного спуска, классификатор умножает каждый  $x_i$  на его вес  $w_i$ , суммирует взвешенные признаки и добавляет смещения  $b$ . Полученное число  $z$  выражает взвешенную сумму признаков для класса.

$$z = \sum_{i=1}^n w_i \cdot x_i + b = \vec{w} \cdot \vec{x} + b \quad (2.3)$$

Чтобы получить вывод классификатора в виде вероятности,  $z$  передается в логистическую функцию (сигмоида)  $\sigma(z)$ :

$$\sigma(z) = \frac{1}{1 + \exp(-z)} \quad (2.4)$$

Таким образом, при бинарной LR, вероятность того, что документ  $x$  принадлежит классу  $P(y = 1|x)$  и вероятность того, что документ  $x$  принадлежит другому классу  $P(y = 0|x)$  находятся по следующим формулам:

$$P(y = 1|x) = \sigma(\vec{w} \cdot \vec{x} + b) = \frac{1}{1 + \exp(-(\vec{w} \cdot \vec{x} + b))} \quad (2.5)$$

$$\begin{aligned} P(y = 0|x) &= 1 - \sigma(\vec{w} \cdot \vec{x} + b) = 1 - \frac{1}{1 + \exp(-(\vec{w} \cdot \vec{x} + b))} = \\ &= \frac{\exp(-(\vec{w} \cdot \vec{x} + b))}{1 + \exp(-(\vec{w} \cdot \vec{x} + b))} \end{aligned} \quad (2.6)$$

## 2. Мультиномиальная логистическая регрессия.

Задача мультиномиальной LR – нахождение класса  $k$  из множества классов  $K$ , к которому принадлежит документ  $x$ . Выводом классификатора  $y$  будет вектор из  $|K|$  элементов. Если документ принадлежит классу  $c_i$ , то  $y_i = 1$  и  $y_j = 0 \forall j = c$ .

Для вычисления вероятности мультиномиальная LR использует обобщение сигмоиды, называемое функцией **softmax**.

Для  $i$ -ого члена вектора  $\vec{z} = \{z_1, z_2, \dots, z_K\}$  размерности  $|K|$  *softmax* определяется как:

$$\text{softmax}(z_i) = \frac{\exp(z_i)}{\sum_{j=1}^K \exp(z_j)}, 1 \leq i \leq K. \quad (2.7)$$

*softmax* вектора  $\vec{z}$  находится по формуле 2.8.

$$\text{softmax}(\vec{z}) = \left[ \frac{\exp(z_1)}{\sum_{i=1}^K \exp(z_i)}, \frac{\exp(z_2)}{\sum_{i=1}^K \exp(z_i)}, \dots, \frac{\exp(z_K)}{\sum_{i=1}^K \exp(z_i)} \right] \quad (2.8)$$

Таким образом, вероятность принадлежности документа  $x$   $k$ -ому классу находится по формуле 2.9, используя весовый вектор  $w_k$  и смещения  $b_k$   $k$ -ого класса [7].

$$P(y_k = 1|x) = \frac{\exp(\vec{w}_k \cdot \vec{x} + b_k)}{\sum_{j=1}^K \exp(\vec{w}_j \cdot \vec{x} + b_j)} \quad (2.9)$$

### Преимущества метода:

- является одним из наиболее качественных;
- поддерживает инкрементное обучение.

### Недостатки метода:

- сложная интерпретируемость параметров алгоритма;
- неустойчивость по отношению к выбросам в исходных данных.

LR является одним из самых качественных алгоритмов классификации документов благодаря своей способности прозрачно изучать важность отдельных признаков документа. В практике часто используется в задачах классификации академических, научных или формальных документов. В задачах бинарной классификации или определения тональности документа LR работает лучше по сравнению с другими алгоритмами из-за простоты расчета вероятности принадлежности сигмоидной функцией [5].

### 2.2.3 Метод Наивного Байеса (Naive Bayes, NB)

Для документа  $d$  классификатор из всех  $C$  классов возвращает класс  $c_*$ , который имеет максимальную апостериорную вероятность для данного документа.

$$c_* = \operatorname{argmax}_{c \in C} P(c|d) \quad (2.10)$$

Для вычисления значений  $P(c|d)$  используется теорема Байеса, которая позволяет разбить любую условную вероятность на три другие вероятности:

$$P(c|d) = \frac{P(c) \cdot P(d|c)}{P(d)}. \quad (2.11)$$

Если представить документ  $d$  в виде множества признаков  $f_1, f_2, \dots, f_n$ , то формула 2.10 примет вид:

$$c_* = \operatorname{argmax}_{c \in C} P(c|d) = \operatorname{argmax}_{c \in C} \frac{P(c) \cdot P(d|c)}{P(d)} = \operatorname{argmax}_{c \in C} \frac{P(c) \cdot P(f_1, f_2, \dots, f_n|c)}{P(d)}, \quad (2.12)$$

где  $P(c)$  – априорная вероятность того, что документ относится к классу  $c$ ,  $P(f_1, f_2, \dots, f_n|c)$  – вероятность найти документ  $d = \{f_1, f_2, \dots, f_n\}$  в классе  $c$ ,  $P(d)$  – вероятность того, что документ можно представить в виде множества признаков  $d = \{f_1, f_2, \dots, f_n\}$ .

Так как  $P(d)$  является константой для всех  $c \in C$ ,

$$c_* = \operatorname{argmax}_{c \in C} P(c) \cdot P(f_1, f_2, \dots, f_n|c). \quad (2.13)$$

Вычисление  $P(f_1, f_2, \dots, f_n|c)$  затруднительно из-за большого количества признаков, поэтому делают "наивное" предположение о том, что признаки  $f_1, f_2, \dots, f_n$ , рассматриваемые как случайные величины, независимы в совокупности, то есть любые два признака независимы друг от друга:

$$P(f_1, f_2, \dots, f_n|c) = P(f_1|c) \cdot P(f_2|c) \cdot \dots \cdot P(f_n|c). \quad (2.14)$$

Таким образом, формула нахождения класса документа, состоящего из слов  $w_1, w_2, \dots, w_n$ , наивным байесовским классификатором, выглядит следующим образом:

$$c_{NB} = \operatorname{argmax}_{c \in C} P(c) \cdot \prod_{i \in n} P(w_i | c). \quad (2.15)$$

$P(c)$  – априорная вероятность того, что документ относится к классу  $c$ , находится, используя классическое определение вероятности:

$$P(c) = \frac{N_c}{N_{doc}}, \quad (2.16)$$

где  $N_c$  – количество документов в обучающей выборке с классом  $c$ ,  $N_{doc}$  – количество всех документов.

$P(w_i | c)$  – вероятность принадлежности слова  $w_i$  к классу  $c$ , находится по формуле 2.17.

$$P(w_i | c) = \frac{\operatorname{count}(w_i, c)}{\sum_{w \in D} \operatorname{count}(w, c)}, \quad (2.17)$$

где числитель – частота повторения этого слова в классе  $c$ , знаменатель – сумма частот повторения каждого слова всего документа в классе  $c$ .

В случае когда слово или токен документа не встречается в рассматриваемом классе, но присутствует в обучающей выборке (принадлежит какому-то другому классу), в алгоритме  $NB$  нулевая вероятность этого слова приведет к нулевой вероятности всего документа. Чтобы избежать этой проблемы, используется метод **сглаживания Лапласа** (Laplace Smoothing, Add-one Smoothing), который заключается в прибавлении единицы к частоте каждого слова:

$$P(w_i | c) = \frac{\operatorname{count}(w_i, c) + 1}{\sum_{w \in D} (\operatorname{count}(w, c) + 1)} = \frac{\operatorname{count}(w_i, c) + 1}{(\sum_{w \in D} \operatorname{count}(w, c)) + |D|} \quad (2.18)$$

### Преимущества метода:

- высокая скорость работы;
- поддерживает инкрементное обучение;
- имеет относительно простую программную реализацию алгоритма;
- возможность работы с небольшим набором данных для обучения.

### **Недостатки метода:**

- неспособность учитывать зависимость результата классификации от сочетания признаков.

В практике NB, не имея возможности учитывать взаимодействие признаков и делая наивное предположение, что все признаки независимы, неожиданно хорошо работает для большого количества задач классификации текстов [7].

### **2.2.4 Метод опорных векторов (Support Vector Machines, SVM)**

Основная идея метода SVM – поиск разделяющей гиперплоскости, максимально удаленной от любой точки обучающих данных. По обеим сторонам разделяющей гиперплоскости строятся две параллельных гиперплоскости. Алгоритм основан на допущении, что чем больше разница или расстояние между этими параллельными гиперплоскостями, тем меньше будет средняя ошибка классификатора [12].

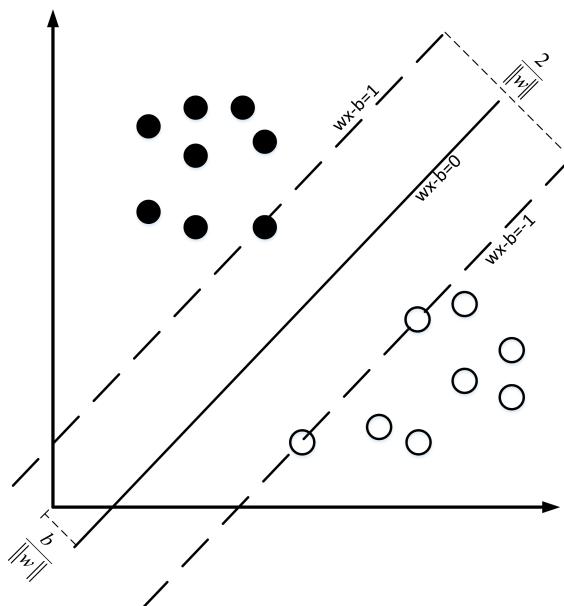


Рисунок 2.1 – Разделяющая гиперплоскость в методе SVM [13].

Разделяющая гиперплоскость имеет вид:

$$\vec{w}x - b = 0, \quad (2.19)$$

где  $w$  – вектор нормали к разделяющей гиперплоскости,  $\frac{b}{\|w\|}$  – расстояние от гиперплоскости до начала координат.

Ближайшие к гиперплоскости точки называются опорными векторами. Параллельные гиперплоскости можно описать следующим уравнением:

$$\vec{w}x - b = |1|. \quad (2.20)$$

- В случае когда обучающая выборка линейно разделима, то строится разделяющая гиперплоскость так, чтобы по одну сторону от неё лежали все точки одного класса, а по другую – все точки другого класса. При классификации объекта учитывается, с какой стороны прямой он находится в пространстве признаков.

Гиперплоскости выбираются таким образом, чтобы между ними не лежала ни одна точка обучающей выборки и затем максимизировать расстояние между гиперплоскостями, которое равно  $\frac{2}{\|w\|}$ . Таким образом, задача алгоритма – минимизировать  $\|w\|$ .

Задача минимизации  $\|w\|$  по теореме Каруша-Куна-Таккера (Karush–Kuhn–Tucker conditions, ККТ) эквивалентна двойственной задаче поиска седловой точки функции Лагранжа.

- В случае когда обучающая выборка линейно неразделима, то есть точки, принадлежащие разным классам, нельзя разделить с помощью гиперплоскости, необходимо перейти от исходного пространства признаков документов к новому, в котором обучающая выборка окажется линейно разделимой. Для этого каждое скалярное произведение необходимо заменить на некоторую функцию, отвечающую определенным требованиям. Эту функцию называют ядром. Замена скалярного произведения функцией-ядром позволяет перейти к другому пространству признаков, где данные уже будут линейно разделимы. Наиболее распространёнными ядрами являются радиальная базисная функция, сигмоид, однородное и неоднородное полиномиальное ядро.

- В случае многоклассовой классификации, задача разбивается на несколько задач бинарной классификации. Общераспространенными являются подходы "One-versus-All" и "One-versus-One".
  - В подходе "One-versus-All" находится гиперплоскость, которая разделяет объекты одного класса от всех остальных, и выбирается класс, гиперплоскость которого находится на наибольшем расстоянии от его объектов.
  - В подходе "One-versus-One" находится гиперплоскость для разделения каждого двух классов, пренебрегая объектами других классов, и выбирается класс, которого выбрали большинство классификаторов [5].

#### **Преимущества метода:**

- один из наиболее качественных методов;
- возможность работы с небольшим набором данных для обучения;

#### **Недостатки метода:**

- сложная реализация;
- работает плохо при больших данных;
- неустойчивость по отношению к выбросам в исходных данных [14].

SVM достаточно хорошо работает для некоторых конкретных задач, таких как классификация журналистских и медицинских статей, которые обычно содержат низкую степень неформальности. Поскольку SVM изначально разработан для бинарной классификации, в подобных задачах работает с высокой достоверностью (accuracy – отношение числа правильно классифицированных объектов к общему числу объектов) [5].

### **2.2.5 Метод деревьев решений (Decision Tree, DT)**

Деревом решений называют ациклический граф, по которому производится классификация объектов (документов), описанных набором признаков. Каждый узел дерева содержит условие ветвления по одному из признаков

[4]. В процессе классификации осуществляются последовательные переходы от одного узла к другому в соответствии со значениями признаков объекта. Разбиение заканчивается тогда, когда достигнут один из листьев (конечных узлов) дерева, то есть в подмножестве оказываются лишь объекты из одного класса. Значение этого листа определит класс, которому принадлежит рассматриваемый объект.

Алгоритм построения дерева решений состоит из следующих шагов.

1. Создается первый узел дерева, в который входят все документы, каждый из которых представлен признаками  $X_1, X_2, \dots, X_n$ .
2. Для текущего узла дерева выбираются наиболее подходящий признак  $X_i$  и его значение  $x_{ia}$ .
3. На основе выбранного признака и его значения производится разделение обучающей выборки на две части.
4. Продолжить разбиение пока не рассмотрены все признаки или на текущем узле объекты принадлежат разным классам [15].

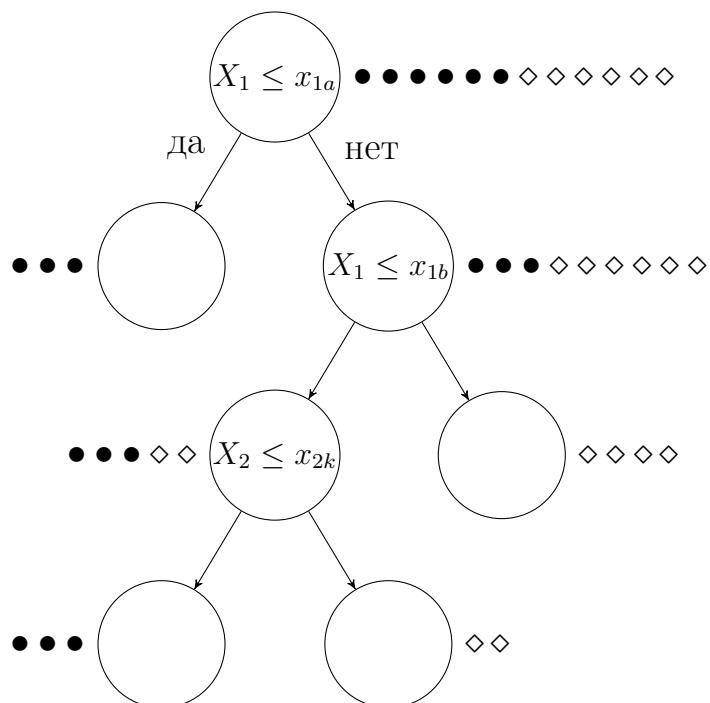


Рисунок 2.2 – Пример дерева решений бинарной классификации объектов, описанных признаками  $X_1, X_2$ .

**Выбор условий разбиения.**

Для выбора подходящего признака на практике применяют различные критерии, наиболее популярным из которых стал теоретико-информационный. Теоретико-информационный критерий основан на понятиях **энтропии** (Entropy,  $E$ ) и **прироста информации** (Information Gain,  $IG$ ).

Энтропия является мерой неоднородности (примеси) или неопределенности подмножества по представленным в нём классам, вычисляется следующей формулой:

$$E = - \sum_{i=1}^K p_i \log_2(p_i), \quad (2.21)$$

где  $p_i$  - вероятность того, что случайно выбранный элемент множества принадлежит  $i$ -тому классу,  $K$  - количество классов, к которым могут принадлежать элементы множества.

На примере, представленном на рисунке 2.2, энтропия корневого состояния равна  $E = -0.5 \log_2(0.5) - 0.5 \log_2(0.5) = 1$ . Корневое состояние имеет самую высокую примесь или неопределенность, соответственно, его энтропия принимает значение 1 – максимально возможное значение энтропии. Энтропия конечного узла, где в подмножестве оказываются объекты только одного класса, например в узле с элементами  $\diamond\diamond$ , равна  $E = -1 \cdot \log_2(1) = 0$ , что и является минимальным возможным значением энтропии.

Прирост информации при разбиении выборки по условию  $Q$  равен разнице энтропии родительского узла и объединенной энтропии дочерних узлов:

$$IG(Q) = E_{parent} - \sum_{i=1}^q \frac{N_i}{N} E_i, \quad (2.22)$$

где  $E_{parent}$  - энтропия родительского узла,  $E_i$  - энтропия  $i$ -того дочернего узла,  $q$  – число дочерних узлов после разбиения,  $N_i$  – число элементов выборки  $i$ -того дочернего узла,  $N$  - число элементов родительского узла [16].

Таким образом, лучшим условием разбиения будет тот, который обеспечит максимальное снижение энтропии или максимальный прирост информации результирующего подмножества относительно родительского.

Модель сравнивает каждое возможное условием разбиения и выбирает тот, который максимизирует прирост информации, то есть он проходит через все признаки объекта и их значения для поиска наилучшего признака и соответствующего значения.

## **Критерий остановки алгоритма.**

В результате работы алгоритма возможно будет построено дерево, в котором для каждого объекта обучающей выборки будет создан отдельный лист. Очевидно, что такое дерево окажется бесполезным, поскольку оно будет переобученным — каждому объекту будет соответствовать свой уникальный набор условий, актуальный только для данного объекта.

Решением проблемы является принудительная остановка построения дерева, пока оно не стало переобученным. Для этого разработаны следующие подходы.

- *Ранняя остановка* — алгоритм будет остановлен, как только будет достигнуто заданное значение некоторого критерия, например процентной доли правильно распознанных примеров.
- *Ограничение глубины дерева* — задание максимального числа разбиений в ветвях, по достижении которого обучение останавливается.
- *Задание минимально допустимого числа примеров в узле* — запретить алгоритму создавать узлы с числом примеров меньше заданного.

## **Преимущества метода:**

- высокая достоверность и скорость работы;
- позволяет работать с большим объемом информации без специальной предобработки данных.

## **Недостатки метода:**

- неустойчивость по отношению к выбросам в исходных данных;
- возможность получения переобученного дерева [17].

В отличие от kNN, LR и SVM метод деревьев решений считается устойчивым к шуму, потому что его стратегии сокращения позволяют избежать переобучения зашумленных данных. В практике используется для классификации документов различных областей.

## 2.2.6 Случайный лес (Random Forest, RF)

Основная идея алгоритма RF заключается в использовании большого набора (ансамбля) деревьев решений, созданных на случайному подвыборке данных. Набор таких деревьев-классификаторов образует лес.

Алгоритм обучения классификатора для RF применяет общую технику **Бутстрэп-агрегирование** (Bootstrap Aggregating) или **Бэггинг** (Bagging), состоит из следующих шагов.

- Пусть обучающая выборка состоит из  $N$  объектов, размерность пространства признаков равна  $M$ , размер выбранного набора признаков равен  $m$ .
- 1. *Bootstrapping* – генерация случайной повторной подвыборки из обучающей выборки. Размер подвыборки равен  $N$ , то есть совпадает с размером исходной обучающей выборки. Некоторые объекты попадут в неё несколько раз. *Bootstrapping* гарантирует, что для каждого дерева используется разные данные, поэтому в некотором смысле это помогает модели быть менее чувствительной к исходным обучающим данным.
- 2. Построение дерева решений, классифицирующее образцы данной подвыборки.
  - *Random feature selection* – в ходе создания очередного узла дерева выбирается случайный набор признаков, на основе которых производится разбиение. Обычно, размер выбранного набора признаков  $m \approx \sqrt{M}$ . Случайный выбор признаков помогает уменьшить корреляцию между деревьями.
  - Выбор наилучшего из этих  $m$  признаков может осуществляться различными способами, как критерий Джини, критерий прироста информации.
  - Дерево строится до полного исчерпания подвыборки. Некоторые деревья будут обучены менее важным признакам, поэтому они будут давать плохие прогнозы, но некоторые деревья также будут давать плохие прогнозы в противоположном направлении, поэтому они будут сбалансиированы.

3. Классификация объектов проводится путём голосования: каждое дерево относит классифицируемый объект к одному из классов, выбирается класс, за который проголосовало наибольшее число деревьев. Этот процесс объединения результатов всех деревьев называется агрегацией (*Aggregation*) [18].

#### **Преимущества метода:**

- высокая достоверность (при увеличении количества деревьев);
- способность эффективно обрабатывать данные с большим числом признаков и классов;
- устойчивость по отношению к выбросам в исходных данных.

#### **Недостатки метода:**

- сложная интерпретируемость;
- низкая скорость при построении большого количества деревьев [19].

Являясь наиболее качественным алгоритмом, RF широко используется для классификации документов из разных областей, благодаря своей устойчивости к выбросам в исходных данных. В случаях, когда потребность в высокой вычислительной мощности и ресурсах не важна, RF является наилучшим алгоритмом машинного обучения для классификации документов.

### **2.2.7 Сравнение алгоритмов машинного обучения**

Таблица 2.1 Сравнение алгоритмов машинного обучения в задачах классификации текстов

Критерии	kNN	LR	NB	SVM	DT	RF
Достоверность	низкая	высокая	средняя	средняя	высокая	высокая
Скорость	низкая	средняя	высокая	средняя	средняя	низкая
Устойчивость к шуму	нет	нет	да	нет	да	да
Интерпретируемость	лёгкая	сложная	лёгкая	сложная	лёгкая	сложная

## **2.3 Гибридные системы**

Гибридные системы сочетают базовый классификатор, обученный с использованием машинного обучения, с системой, основанной на правилах, которые используются для дальнейшего улучшения результатов. Такие системы можно легко строить, добавив определенные правила для тех конфликтующих тегов, которые не были правильно смоделированы базовым классификатором [3].

## 2.4 Рассмотрение экспериментов

В статье [20] сравнивается достоверность различных методов классификации в 12 разных наборах данных, охватывающих различные размеры выборки, социальные сети и платформы электронной коммерции, корпоративные блоги.

Таблица 2.2 Данные, используемые в эксперименте статьи [20].

ID	Классифицируемые документы	КС	РНД	РПП	Классы
AMT	заголовки отзывов о продуктах на Amazon	5	3000	239	2 (+, -)
AMR	отзывы продуктов на Amazon	82	3000	3374	2 (+, -)
IMD	обзоры фильмов, опубликованные на сайте IMDB	15	1000	557	2 (+, -)
YEL	отзывы о ресторанах на сайте Yelp	11	1000	480	2 (+, -)
FBK	комментарии на Facebook	13	3000	549	3 (+, -, ~)
CBC	комментарии в корпоративных блогах	36	2942	1274	3 (+, -, ~)
TWS	посты в Twitter	10	3000	349	3 (+, -, ~)
YTU	комментарии Youtube	17	1000	624	2 (реклама, коммуникация пользователей)
SMS	текстовые сообщения, предоставляемые компаниями Almeida, Gomez Hidalgo	19	1000	861	2 (реклама, коммуникация пользователей)
ROT	обзоры фильмов, опубликованные на сайте Rotten Tomatoes	22	3000	855	2 (субъективный, объективный)
CBP	посты в корпоративных блогах	344	1000	10170	3 (выс., сред., низ. оценка повествования)
TWC	посты в Twitter	10	3000	358	3 (эмоция, информация, их комбинация)

Данные, используемые в эксперименте, описаны в таблице 2.2, где КС –

среднее количество слов в документе, РНД – размер набора данных (количество документов), РПП – размерность пространства признаков, (+, -, ~) – тональные оценки: позитивная, негативная, нейтральная.

На рисунке 2.3 проиллюстрированы результаты достоверности методов kNN (метод k ближайших соседей), NB (наивный байесовский алгоритм), RF (случайный лес), DICT (4 метода на основе правил), ANN (искусственные нейронные сети).

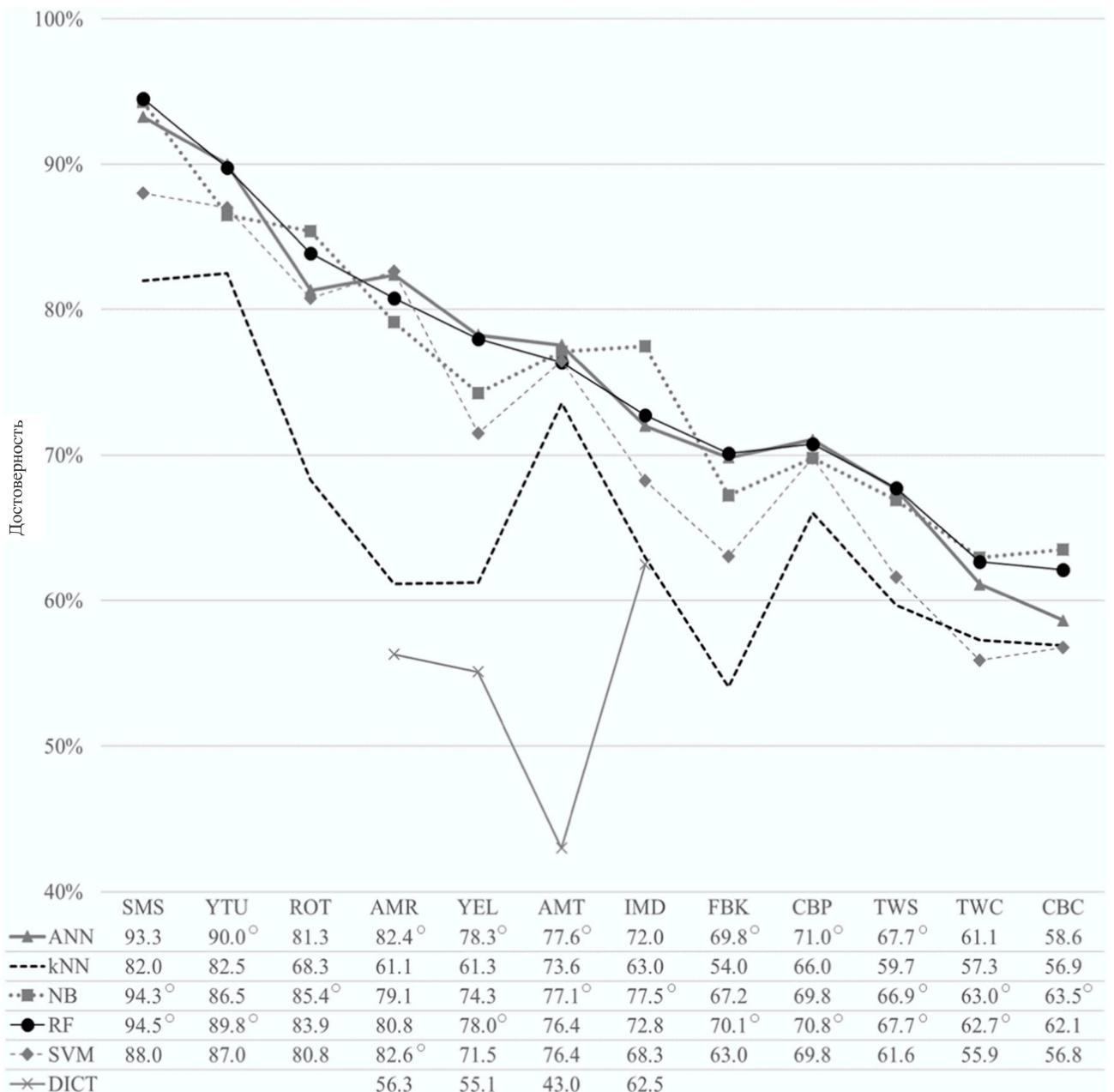


Рисунок 2.3 – Результаты эксперимента из статьи [20]

Видно, что максимальные достоверности при решении самой простой за-

дачи классификации (бинарная классификация коротких текстовых сообщений (SMS) с достоверностью 94,5 %) и самой сложной задачи (классификация тональности пользовательских комментариев к постам корпоративного блога (CBC) с достоверностью 63,5%).) сильно различаются. Это доказывает, что достоверность алгоритмов классификации текста сильно зависит от конкретной задачи и наборов данных.

Методы на основе правил отстают от всех методов машинного обучения с учителем, особенно при классификации коротких текстов.

Во всех различных контекстах ANN, RF и NB последовательно достигают высочайшей достоверности, а kNN почти всегда достигает самой низкой достоверности среди методов машинного обучения.

Пять самых низких показателей достоверности получены при задачах классификации FBK, СВР, TWS, TWC, и CBC. Это объясняется тем, что все эти наборы документов имеют высокий уровень шума, а также над ними выполняется 3-классовая классификация. Соответственно, SVM, являясь неустойчивым по отношению к выбросам данных и слабоработающим при многоклассовых классификациях, при этих данных имеет низкую достоверность.

kNN относительно плохо работает с длинными текстами обзоров Amazon (AMR) по сравнению с короткими заголовками обзоров Amazon (AMT).

В статье [4] проводится эксперимент по классификации 3000 учебно-методических документов научно-образовательного учреждения, используя различные методы машинного обучения. В качестве признака классификации использовалось "наименование документа" (то есть его категория: служебная записка, заявление, рабочая программа, лекционный материал и т. д.). В эксперименте сравниваются достоверности классификации до и после предварительной обработки данных. В качестве предварительной обработки применялись методы извлечения, как токенизация, удаление стоп-слов и лемматизация. Также сокращалось число используемых признаков (термов).

Выполняя предобработку документов, были достигнуты положительные результаты как по времени обучения классификатора (вплоть до трехкратного улучшения показателя), так и по достоверности его работы (прирост от 5 до 20%).

Полученные результаты подтверждают, что LR и DT хорошо работают

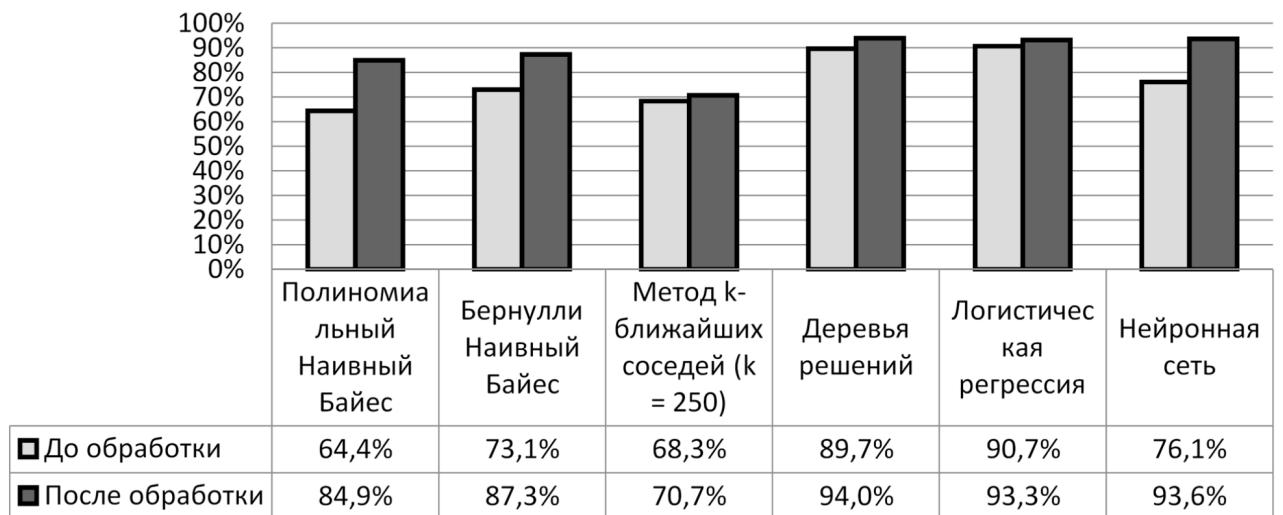


Рисунок 2.4 – Результаты эксперимента из статьи [4] – сравнение достоверности классификации.

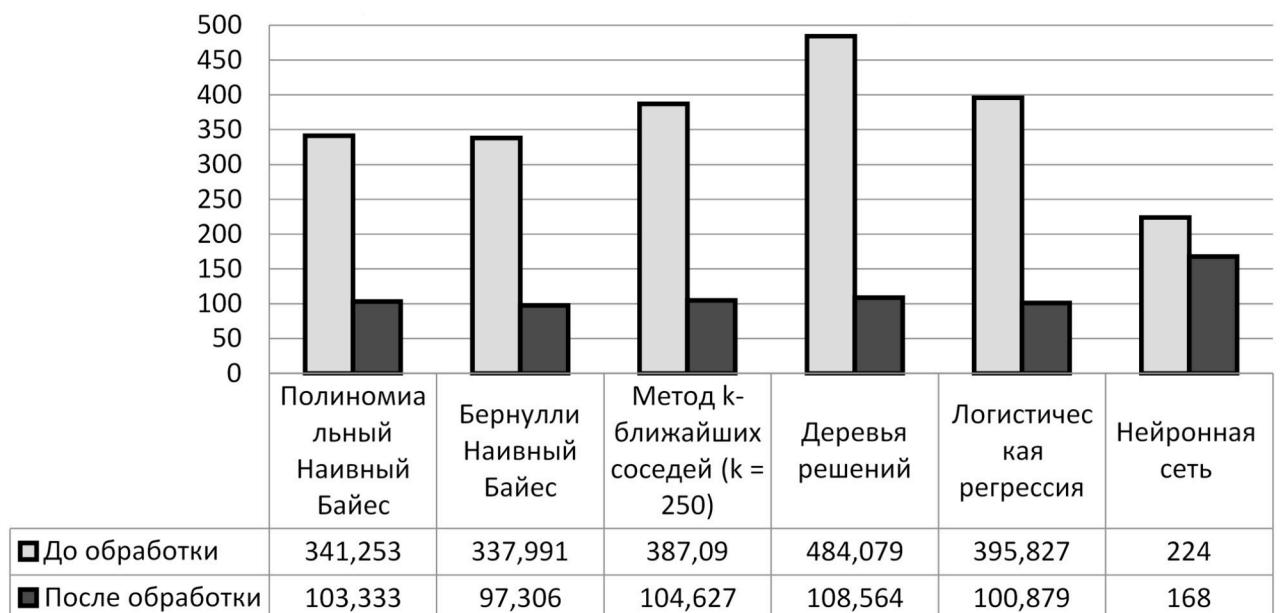


Рисунок 2.5 – Результаты эксперимента из статьи [4] – сравнение времени обучения классификатора.

при классификации документов с низким уровнем шума, а kNN плохо работает с длинными текстами.

Несмотря на его высокую достоверность, у DT время обучения классификатора оказалось самым высоким.

Две версии алгоритма NB оказались относительно быстрыми по сравнению с kNN, DT и LR, что и было предсказуемо.

# **Заключение**

В ходе выполнения данной работы были рассмотрены этапы процесса классификации текстов: предобработка и индексация документов, отбор термов. Также были изучены существующие подходы и основные методы решения задачи классификации текстов. Среди существующих подходов к классификации текстов было уделено внимание системам на основе машинного обучения.

На основе проделанной работы можно сделать вывод о том, что алгоритмом с самой высокой достоверностью при решении задач классификации различных документов является случайный лес, а самым быстрыми являются логистическая регрессия и алгоритм наивного байеса. Метод k ближайших соседей – самый слабый среди всех остальных рассмотренных алгоритмов машинного обучения. Для документов с высоким уровнем шума эффективно работают алгоритм наивного байеса, деревья решений и случайный лес.

# Литература

1. Юсупова Н.И., Богданова Д.Р., Бойко М.В. Алгоритмическое и программное обеспечение для анализа тональности текстовых сообщений с использованием машинного обучения // Вестник Уфимского государственного авиационного технического университета. 2012. С. 91–99.
2. Ikonomakis M., Kotsiantis S., Tampakas V. Text Classification Using Machine Learning Techniques // WSEAS TRANSACTIONS on COMPUTERS. 2005. Т. 4, № 8. С. 966–974.
3. Kamran Kowsari, Meimandi Jafari. Text Classification Algorithms: A Survey // Information. 2019. Т. 10. – Режим доступа: <https://www.mdpi.com/2078-2489/10/4/150>.
4. Краснянский М.Н., Обухов А.Д. Сравнительный анализ методов машинного обучения для решения задачи классификации документов научно-образовательного учреждения // Тамбовский государственный технический университет. 2018. August. С. 174–181.
5. Manning Christopher D., Raghavan Prabhakar, Schütze Hinrich. An Introduction to Information Retrieval. Cambridge University Press, 2009. С. 329–340. – Режим доступа: <https://nlp.stanford.edu/IR-book/pdf/irbookonlinereading.pdf>.
6. Zhang Xiang, Zhao Junbo, LeCun Yann. Character-level Convolutional Networks for Text Classification // Cornell University. 2016.
7. Jurafsky Dan, Martin James H. Speech and Language Processing. 3 изд. New Jersey: Prentice Hall series in Artificial Intellegence, 2022. – Режим доступа: <https://web.stanford.edu/~jurafsky/slp3/>.
8. Mikolov Tomas, Chen Kai. Efficient Estimation of Word Representations in Vector Space // Cornell University. 2013.
9. Hyvärinen Aapo, Karhunen Juha, Oja Erkki. Independent component analysis // New York: Wiley-Interscience. 2001.

10. Веторв Д.П., Журавлёв Ю.И. Уменьшение размерности описания данных: метод главных компонент. Математические основы теории прогнозирования // Московский государственный университет имени М. В. Ломоносова. 2011.
11. Aubaid Asmaa M., Mishra Alok. A Rule-Based Approach to Embedding Techniques for Text Document Classification // Applied Sciences. 2020. № 10.
12. Patra Anuradha, Singh Divakar. A Survey Report on Text Classification with Different Term Weighing Methods and Comparison between Classification Algorithms // International Journal of Computer Applications (0975 – 8887). 2013. August. Т. 75, № 7.
13. Shalaginov Andrii, Dehghantanha Ali, Banin Sergii. Cyber Threat Intelligence, Machine Learning Aided Static Malware Analysis: A Survey and Tutorial. 2018. С. 7–45.
14. Barber David. Bayesian Reasoning and Machine Learning. Cambridge University Press, 2020.
15. Classification and Regression Trees / Leo Breiman, Jerome Friedman, Charles J. Stone [и др.]. Chapman and Hall, 1984.
16. Quinlan J. Ross. C4.5: Programs for Machine Learning (Morgan Kaufmann Series in Machine Learning). Morgan Kaufmann, 1992.
17. Murthy Sreerama K. Automatic Construction of Decision Trees from Data: A Multi-Disciplinary Survey // Siemens Corporate Research, Princeton. 1997.
18. Ho Tin Kam. Random Decision Forests // Proceedings of the 3rd International Conference on Document Analysis and Recognition. 1995. c. 278–282. – Режим доступа: <https://web.archive.org/web/20181105063147/http://ect.bell-labs.com/who/tkh/publications/papers/odt.pdf>.
19. Breiman Leo, Cutler Adele. Random Forests // Machine Learning Journal. 2001. Т. 45, № 1. с. 5—32. – Режим доступа: [https://www.stat.berkeley.edu/~breiman/RandomForests/cc\\_home.htm](https://www.stat.berkeley.edu/~breiman/RandomForests/cc_home.htm).

20. Hartmann Jochen, Huppertz Juliana. Comparing automated text classification methods // International Journal of Research in Marketing. 2019. Т. 20-38, № 36.
21. Дмитриев Е.А., Мясников В.В. Сравнение алгоритмов описания комплекснозначного поля градиента цифровых изображений с использованием линейных методов снижения размерности // Компьютерная оптика. 2018. С. 822–828.
22. Баданина Н.Д., Судаков В.А. Модели машинного обучения для классификации отзывов о банках // Препринты ИПМ им. М.В.Келдыша. 2021. Т. 14, № 50. – Режим доступа: <https://doi.org/10.20948/prepr-2021-50>.
23. Text categorization using Rocchio algorithm and random forest algorithm / S. Thamarai Selvi, P. Karthikeyan, A. Vincent [и др.] // 2016 Eighth International Conference on Advanced Computing (ICoAC). 2016. С. 7–12.
24. Akinsola Jet. Supervised Machine Learning Algorithms: Classification and Comparison // International Journal of Computer Trends and Technology (IJCTT). 2017. Т. 48, № 3.