МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования
«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ Н. Г. ЧЕРНЫШЕВСКОГО»

Кафедра теории функций и стохастического анализа

РАЗРАБОТКА ИНФОРМАЦИОННОЙ СИСТЕМЫ ДЛЯ АНАЛИЗА НАУЧНЫХ ТЕКСТОВ НА РУССКОМ ЯЗЫКЕ В КОНТЕКСТЕ МАШИННОГО ОБУЧЕНИЯ

АВТОРЕФЕРАТ БАКАЛАВРСКОЙ РАБОТЫ

Студентки 4 курса 451 группы
направления 38.03.05 — Бизнес-информатика

механико-математического факультета
Носкиной Анастасии Викторовны

Научный руководитель
к. ф.-м. н., доцент _____ Д. В. Мельничук

Заведующий кафедрой
д. ф.-м. н., доцент _____ С. П. Сидоров

Саратов 2025

ВВЕДЕНИЕ

Актуальность темы. В связи с растущим объемом научной информации и разнообразием научных областей, с которыми сталкиваются современные исследователи, возникает острая необходимость в создании эффективных инструментов для углубленного анализа и исследования публикаций. При постоянном увеличении количества научных статей и разнообразии тематик, традиционные методы ручного исследования информации не в состоянии удовлетворить потребности ученых, которым требуется быстрый и качественный доступ к необходимым материалам.

Данная работа представляет интерес, поскольку выявление похожих публикаций становится особенно важным, и, в результате, не только упрощается процесс исследования, но и появляется возможность более глубоко понимять изучаемые темы. Исследователи, в свою очередь, могут быстрее находить связи между различными работами, что, соответственно, ускоряет процесс написания новых научных статей и позволяет формировать новые идеи и гипотезы на основе имеющихся данных.

Целью бакалаврской работы является проведение анализа научных статей и разработка информационной сиситемы посредством использования машинного обучения для упрощения задачи research-анализа пользователями электронных библиотек при написании новых работ, а именно, выявление категории науки, к которой относится предоставленное пользователем наименование статьи, а также поиск наиболее схожих статей для дальнейшего обращения к ним, что помогает эффективно и быстро ориентироваться в понимании новой исследуемой тематики.

Объект исследования – публикации, взятые из открытой научной электронной библиотеки CyberLeninka по кампьютерным и информационным, химическим, математическим, физическим разделам наук.

Предмет исследования – методы и алгоритмы машинного обучения, используемые для выявления схожести научных публикаций.

Для достижения поставленной в работе цели необходимо решить следующие **задачи**:

— Проведение анализа существующих научных статей с использованием

языка программирования Python;

— Изучение уровня схожести наименований научных работ с предполагаемым названием новой статьи;
— Выявление тематики, к которой можно определить введенное наименование для новой научной работы;
— Изучение методов машинного обучения для посмотрения моделей, определяющих принадлежность рассматриваемых публикаций к определенной тематике и нахождение косинусого сходства введенного текста к существующим наименованиям статей;
— Проведение сравнительного анализа работы моделей машинного обучения с использованием метрик качества;
— Разработка пользовательского функционала для информационной системы с использованием Python библиотеки Streamlit.

Практическая значимость проводимого исследования состоит в том, что на основании построенных моделей машинного обучения в контексте задачи анализа научных статей и преобразовании используемых моделей в информационную систему, проведение исследовательского анализа при написании новой публикации занимает наименьшее количество времени, и, соответственно, повышает эффективность выявления значимости разрабатываемого исследования в определенном временном диапазоне.

Структура и содержание бакалаврской работы. Работа состоит из введения, пяти разделов, заключения, списка использованных источников, содержащего 22 наименования, и четырех приложений. Общий объем работы состовляет 48 страниц, включая 6 таблиц и 13 графических представлений.

ОСНОВНОЕ СОДЕРЖАНИЕ РАБОТЫ

Во **введении** обосновывается актуальность темы работы, формулируется ее цель и соответствующие решаемые задачи, а также отмечается практическая значимость полученных результатов.

В **первом разделе** приводится сбор данных путем парсинга и исследование выбранной области с помощью обработки и преобразования набора данных в графические представления для наилучшего представления информации о них путем использования библиотек языка программирования Python, таких, как Pandas, NumPy, NLTK, Simplemma, Matplotlib, WordCloud.

Рассматриваются используемые методы машинного обучения для решения задачи классификации по принадлежности введенного наименования статьи к определенной тематике и поиска схожих статей с помощью нахождения косинусного сходства между входным текстом и заголовками статей из общего набора данных.

Для исследования были выделены наиболее популярные модели на аналогичных задачах классификации текста, обученные на одном и том же корпусе научных текстов на русском языке, собранных путем парсинга.

1. Дерево решений (DecisionTreeClassifier) является одним из наиболее популярных методов машинного обучения, используемых на задачах классификации и регрессии. Данный алгоритм работает путем разбиения данных на подмножества, которые зависят от входных признаков, формируя древовидную структуру, где внутренний узел – решение (на основе признака), ветвь – результат решения, конечный узел – конечное решение или классификация;
2. Метод k-ближайших соседей (KNeighborsClassifier) является непараметрическим алгоритмом машинного обучения, в котором при классификации присваивается дискретная метка класса новой точке данных относительно оценок соседних точек, что способствует определению близкого расположения точек со схожими признаками;
3. Наивный байесовский классификатор (MultinomialNB) относится к семейству вероятностных алгоритмов, которые основываются на теореме Байеса для расчета вероятности принадлежности заданного образца к

каждому из возможных классов и относит образец к классу с наибольшей вероятностью;

4. Градиентный бустинг (CatBoostClassifier) позволяет строить аддитивную функцию в виде суммы деревьев решений итерационно, по аналогии с методом градиентного спуска, где под градиентным спуском понимается численный метод нахождения локального минимума или максимума функции с помощью движения вдоль градиента (сингулярное значение, которое сообщает наклон функции в точке);
5. Логистическая регрессия (LogisticRegression) – классический инструмент для решения задач классификации и регрессии. В данном случае алгоритм используется для задачи многоклассовой классификации, который возвращает оценку вероятности принадлежности точки данных к определенному положительному классу;
6. Метод опорных векторов (SVM) – контролируемый алгоритм машинного обучения, который использует поиск наилучшей гиперплоскости между разными классами в пространстве признаков, где под гиперплоскостью понимается граница принятия решения при разделении на классы.

Для выявления косинустного сходства производилась векторизация данных. Косинусное сходство (cosine similarity) — это метрика, используемая для измерения степени сходства между двумя векторами в пространстве. Оно основано на вычислении косинуса угла между двумя векторами и применяется во многих областях, включая обработку естественного языка, машинное обучение и системы рекомендаций. Формула для вычисления косинустного сходства выглядит следующим образом:

$$\text{cosine similarity} = \cos(\Theta) = \frac{A \cdot B}{\|A\|\|B\|} = \frac{\sum_{i=1}^{n} A_i \cdot B_i}{\sqrt{\sum_{i=1}^{n} (A_i)^2} \cdot \sqrt{\sum_{i=1}^{n} (B_i)^2}},$$

где

$A \cdot B$ — скалярное произведение векторов A и B,

$\|A\|$ и $\|B\|$ — длины (нормы) векторов A и B.

Здесь под скалярным произведением векторов понимаем операцию, которая принимает два вектора и возвращает скаляр (число).

Для двух векторов $A = [A_1, A_2, \ldots, A_n]$ и $B = [B_1, B_2, \ldots, B_n]$ скалярное произведение определяется как:

$$A \cdot B = A_1 \cdot B_1 + A_2 \cdot B_2 + \ldots + A_n \cdot B_n = \sum_{i=1}^{n} A_i \cdot B_i,$$

где n — количество элементов в векторах.

Скалярное произведение также может быть выражено через угол θ между двумя векторами:

$$A \cdot B = \|A\|\|B\| \cos(\theta),$$

где

$\|A\|$ и $\|B\|$ — длины (нормы) векторов A и B,

θ — угол между векторами.

Далее рассматриваются метрики оценки и сравнения качества работы моделей такие, как Precision, Recall, F1-score и Accuracy.

Пусть TP – это истинно положительный объект, TN – это истинно отрицательный объект, FP – это ложноположительный объект, FN – это ложноотрицательный объект.

Метрика Precision, или же точность, формально определяется как доля всех положительно предсказанных объектов по отношению ко всем истинно положительным и ложноположительным объектам. Относительно исследуемой области данная метрика позволяет показать долю правильно классифицированных статей к определённой категории науки относительно всех статей, принадлежащих этой категории науки. При меньшем количестве ложноположительных срабатываний метрика показывает наилучший результат.

Формула для вычисления метрики Precision выглядит следующим образом:

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

Метрика Recall, или же полнота, определяется как доля действительно правильно найденных положительных предсказаний по отношению ко всем

истинно положительным и ложноотрицательным объектам. В исследуемой области при классификации метрика показывает долю правильно предсказанных категорий ко всем правильно определённым категориям. В результате, чем меньше ложноотрицательных срабатываний, тем выше точность.

Формула для вычисления метрики Recall выглядит следующим образом:

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

F1-мера является средним гармоническим значением пары Precision-Recall и позволяет упростить сравнение работы моделей для большого количества классов, показывая качество одним числом.

Формула для вычисления F1-меры выглядит следующим образом:

$$F1\text{-score} = \frac{2}{\frac{1}{\text{Recall}} + \frac{1}{\text{Precision}}} = \frac{2 \cdot (\text{Recall} \cdot \text{Precision})}{\text{Recall} + \text{Precision}} = \frac{TP}{TP + \frac{(FP+FN)}{2}}.$$

Accuracy показывает долю правильных классификаций и позволяет быстро определить, насколько хорошо модель справляется с правильными прогнозами, характеризуя качество модели.

Формула для вычисления метрики Accuracy выглядит следующим образом:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}.$$

Во **втором разделе** производится проведение анализа тектовых данных и выявление основных зависимостей и закономерностей в них.

В **третьем разделе** производится обработка текстовых данных, включающая в себя удаление дубликатов, пустых значений, стоп-слов, а также лемматизацию, токенизацию и векторизацию текста для дальшейшего обучения моделей.

В **четвертом разделе** проводится сравнительный анализ результатов в таблицах по работе каждой модели. На поставленной задаче анализа научных статей и разработки инструмента посредством использования машинного обучения для упрощения задачи research-анализа пользователями электронных библиотек, наилучшим образом на классификации по тематике проявила себя модель MultinomialNB, которая показала достаточно хорошие результа-

ты на выбранном массиве данных. Данный результат может быть обусловлен тем, что модель обеспечивает быстрое развертывание, обладает стабильностью на разнородных данных и прозрачность работы, что может быть критично для пользователей, которым нужен простой, но надежный инструмент для research-анализа. В таблицах, представленных ниже, номера классов 0, 1, 2, 3 соответствуют компьютерным и информационным, математическим, физическим и химическим классам наук.

Таблица 1 – Результаты исследования на всем объеме данных на модели DecisionTreeClassifier

Класс	Precision	Recall	F1-score	Accuracy
0	0.61	0.65	0.63	0.61
1	0.50	0.57	0.53	
2	0.58	0.52	0.54	
3	0.78	0.70	0.73	

Таблица 2 – Результаты исследования на всем объеме данных на модели KNeighborsClassifier

Класс	Precision	Recall	F1-score	Accuracy
0	0.73	0.87	0.79	0.77
1	0.70	0.66	0.68	
2	0.76	0.69	0.72	
3	0.90	0.84	0.87	

Таблица 3 – Результаты исследования на всем объеме данных на модели MultinomialNB

Класс	Precision	Recall	F1-score	Accuracy
0	0.76	0.90	0.82	0.81
1	0.76	0.70	0.73	
2	0.80	0.77	0.79	
3	0.93	0.87	0.90	

Таблица 4 – Результаты исследования на всем объеме данных на модели CatBoostClassifier

Класс	Precision	Recall	F1-score	Accuracy
0	0.75	0.75	0.75	0.72
1	0.67	0.67	0.67	
2	0.63	0.69	0.66	
3	0.84	0.76	0.80	

Таблица 5 – Результаты исследования на всем объеме данных на модели LogisticRegression

Класс	Precision	Recall	F1-score	Accuracy
0	0.79	0.81	0.80	0.79
1	0.71	0.70	0.71	
2	0.77	0.77	0.77	
3	0.90	0.88	0.89	

Таблица 6 – Результаты исследования на всем объеме данных на модели SVM

Класс	Precision	Recall	F1-score	Accuracy
0	0.79	0.83	0.81	0.79
1	0.72	0.70	0.71	
2	0.76	0.77	0.76	
3	0.91	0.87	0.89	

В **пятом разделе** представлена информация по применению косинусного сходства к поставленной задачи, а также описаны особенности использования этого метода для поиска схожих научных работ.

В **шестом разделе** представлена информация о современном Python-фреймворке Streamlit для создания веб-интерфейсов, используемого для визуализации и интерактивного представления результатов выполнения поставленных задач в контексте анализа научных статей. В рамках бакалаврской работы Streamlit используется как платформа для создания пользовательского интерфейса информационной системы.

Архитектура системы включает в себя следующие компоненты:

— Представление всего набора данных для ознакомления с ним;
— Поле для ввода наименования статьи;
— Область предварительного просмотра содержимого;
— Кнопки управления процессом анализа;
— Визуализация облака слов по введенному наименованию статьи.

К модулям анализа относятся:

— Классификатор категорий науки;
— Система поиска похожих статей;
— Построение облака слов по наименованию статьи для улучшенной визуализации.

В рамках данной работы Streamlit обеспечивает преимущества для представления пользовательского функционала информационной системы такие, как быструю разработку интерфейса, встроенную поддержку интерактивных элементов управления, простую интеграцию с библиотеками машинного обучения, а также возможность развертывания как локального, так и облачного решения.

В **заключении** приведены результаты бакалаврской работы.

ОСНОВНЫЕ РЕЗУЛЬТАТЫ

1. Проведен анализ существующих научных статей с использованием языка программирования Python;
2. Изучен уровнь схожести наименований научных работ с предполагаемым названием новой статьи;
3. Проведено выявление тематик, к которым можно определить введенное наименование для новой научной работы;
4. Изучены и использованы на практике методы машинного обучения для посмотрения моделей, определяющих принадлежность рассматриваемых публикаций к определенной тематике и нахождение косинусого сходства введенного текста к существующим наименованиям статей;
5. Проведен сравнительный анализ работы моделей машинного обучения с использованием метрик качества;
6. Разработан пользовательский функционал для информационной системы по проведению исследовательского анализа.