Автономная некоммерческая организация высшего образования "Университет Иннополис"

**Тема 28. Веб-приложение на базе ИИ, подбирающие**

**необходимую литературу по темам**

**отчет по программе дополнительного профессионального образования**

**«Аналитика данных и машинное обучение**

Выполнил:

Кузнецов Александр Михайлович

2023

Оглавление

[Введение 3](#_Toc151840230)

[Глава 1. Введение в рекомендательные системы 4](#_Toc151840231)

[Определение рекомендательной системы 4](#_Toc151840232)

[Коллаборативная фильтрация 4](#_Toc151840233)

[Контент-ориентированная фильтрация 4](#_Toc151840234)

[Глава 2. Разработка алгоритмов машинного обучения для анализа запросов пользователей и подбора литературы. 5](#_Toc151840235)

[Подготовка данных 5](#_Toc151840236)

# Введение

В современном мире объем информации постоянно растет, и поиск необходимой информации становится все более сложной задачей. Традиционные методы поиска информации, такие как использование поисковых систем, зачастую не позволяют быстро и точно найти нужную информацию. В связи с этим возникает потребность в создании новых методов и технологий поиска информации, которые будут способны обрабатывать большие объемы данных и предоставлять пользователю наиболее релевантные результаты.

Одним из перспективных направлений в этой области является использование искусственного интеллекта для создания веб-приложений, способных анализировать запросы пользователей и предлагать им наиболее подходящую литературу. В данной дипломной работе предлагается разработать веб-приложение, которое будет использовать алгоритмы машинного обучения для подбора литературы по заданным темам.

Целью работы является создание веб-приложения, которое позволит пользователям быстро и удобно находить литературу по интересующим их темам. Для достижения этой цели необходимо решить следующие задачи:

1. Разработка алгоритмов машинного обучения для анализа запросов пользователей и подбора литературы.
2. Создание прототипа веб-приложения и его тестирование.
3. Анализ эффективности работы веб-приложения.

В работе будут использоваться методы машинного обучения, обработки естественного языка и анализа текстов.

# Глава 1. Введение в рекомендательные системы

## Определение рекомендательной системы

Рекомендательная система – это программное обеспечение или веб-сервис, который использует алгоритмы для анализа предпочтений пользователя и предоставления ему рекомендаций по товарам или услугам.

Существует два основных типа рекомендательных систем: коллаборативная фильтрация и контент-ориентированная фильтрация.

## Коллаборативная фильтрация

Коллаборативная фильтрация основана на анализе поведения пользователя и сравнении его с поведением других пользователей. Алгоритмы коллаборативной фильтрации используют данные о покупках, оценках и отзывах пользователей для определения предпочтений каждого пользователя и рекомендации товаров, которые могут им понравиться.

Преимущества коллаборативной фильтрации:

* Возможность рекомендовать товары на основе предпочтений большого числа пользователей.
* Быстрая обработка запросов.

Недостатки коллаборативной фильтрации:

* Не учитывает специфические свойства товаров.
* Не может рекомендовать новые товары.

## Контент-ориентированная фильтрация

Контент-ориентированная фильтрация использует информацию о свойствах товара и предпочтениях пользователя для рекомендации товаров.

Алгоритмы контент-ориентированной фильтрации анализируют свойства товара (например, описание, характеристики, отзывы) и сравнивают их с предпочтениями пользователя для определения подходящих товаров.

Преимущества контент-ориентированной фильтрации:

* Учитывает специфические свойства товаров.
* Может рекомендовать новые товары.

Недостатки контент-ориентированной фильтрации:

* Требует большого объема данных о товарах.
* Медленная обработка запросов.

# Глава 2. Разработка алгоритмов машинного обучения для анализа запросов пользователей и подбора литературы.

В данной главе будут описаны алгоритмы машинного обучения, которые будут использоваться для анализа запросов пользователей и подбора литературы.

Сравнивать между собой буду 3 модели:

* Tfidfvectorizer
* Countvectorizer
* SentenceTransformer

## Подготовка данных

Для создания системы рекомендация я взял набор данных “Best Books (10k) Multi-Genre Data”, который включает в себя 10 тыс. строк с такими данными, как:

* Название книги
* Автор
* Описание
* Жанры
* Рейтинг от 0 до 5
* Количество голосов
* URL – адрес с книгой

В наборе данных, который мы используем для обучения модели, обнаружен столбец под названием Description (Описание), который содержит пропуски или отсутствующие значения. Эти пропуски не могут быть просто заменены средним значением, так как значение каждой строки уникально. Если мы попытаемся заменить эти пропуски средним значением, то можем внести ошибки в нашу модель. Чтобы избежать таких ошибок, нам необходимо найти какое-то другое решение, чтобы обработать или удалить эти пропуски.

Для того чтобы улучшить качество распознавания жанров, необходимо было провести некоторую предварительную обработку данных. Во-первых, нужно было привести все жанры к единому формату, чтобы избавиться от различных вариантов написания одного и того же жанра.

Кроме того, в некоторых жанрах могут присутствовать дополнительные символы, которые не являются частью самого жанра. Например, символ “-” может использоваться для разделения разных жанров фильма. В таком случае необходимо удалить все лишние символы и оставить только те, которые действительно относятся к жанру фильма.

Также стоит отметить, что при работе с жанрами необходимо избегать повторений. Например, если одна и та же книга имеет два разных жанра, таких как “комедия” и “романтическая комедия”, то необходимо написать “романтическая\_комедия”, чтобы избежать дублирования информации.

В столбце Description находятся описания книг, которые нужно обработать, чтобы привести к единому виду. Для этого нужно выполнить следующие шаги:

* Заменить сокращения на полные слова. Например, “см.” на “смотри”.
* Удалить знаки препинания, так как они не несут полезной информации.
* Разбить текст на слова, чтобы каждое слово было отдельной строкой.
* Удалить стоп-слова, такие как “the”, “a”, “and” и т. д., так как они не добавляют информативности.
* Привести все слова к нижнему регистру, чтобы не было различий между строчными и прописными буквами.
* Провести лемматизацию - процесс приведения слова к его словарной форме. Например, глаголы в разных временах приводятся к инфинитиву.

Столбец с авторами я оставлю без изменений и объединю между собой авторов, описание и жанры. Оставшиеся столбцы мне не понадобятся.

Сравнение моделей

Для сравнения книг между собой буду использовать косинусное сходство. Косинусное сходство - это метрика, которая измеряет сходство между двумя векторами на основе их угла. Она принимает значение от 0 до 1, где 0 означает полное отсутствие сходства, а 1 - полное сходство.

Каждая модель будет сравнивать книги по определенной теме, в данном случае это “data science and machine learning”.

Countvectorizer

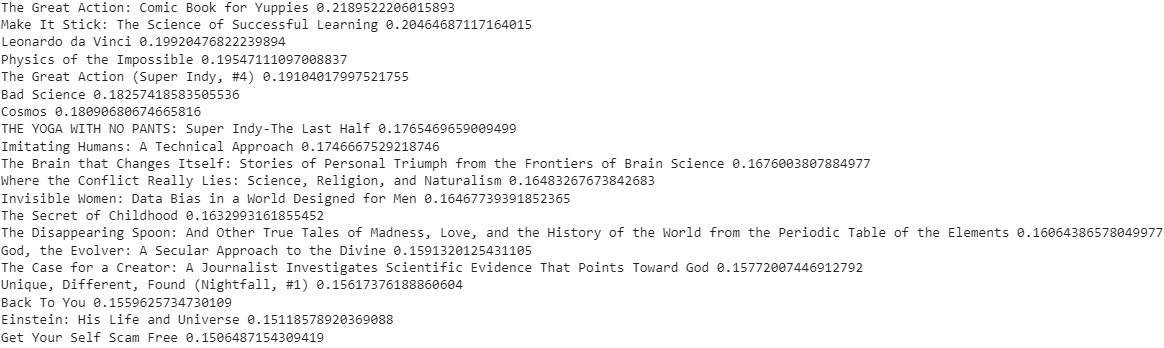
CountVectorizer является моделью машинного обучения. Это векторизатор текста, который преобразует текстовые документы в векторы чисел, которые могут быть использованы в качестве входных данных для других моделей машинного обучения.

Плюсы:

* Простота использования - достаточно вызвать один метод для векторизации текста.
* Скорость работы - CountVectorizer работает быстрее, чем Tf-Idf Vectorizer, так как он не вычисляет tf-idf для каждого слова.

Минусы:

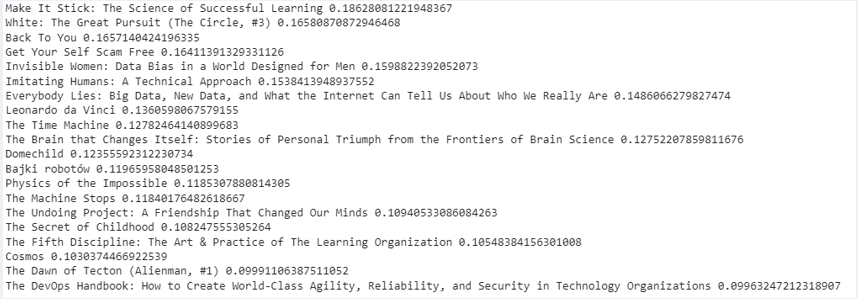
* Не учитывает релевантность слов - CountVectorizer просто считает количество вхождений каждого слова в тексте, не учитывая его релевантность.
* Может приводить к большим разреженным векторам - если в тексте много разных слов, вектор может быть очень большим и разреженным.



Tfidfvectorizer

Tf-idfVectorizer является моделью машинного обучения. Он используется для представления текстовых документов в виде векторов чисел, которые можно использовать в качестве входных данных для других моделей.

* Tf-Idf Vectorizer учитывает релевантность слов, поэтому он может быть более точным для некоторых задач.
* Однако он работает медленнее, так как вычисляет tf-idf для каждого слова.
* Также он может приводить к меньшим векторам, так как удаляет слова с низкой релевантностью.



SentenceTransformers

SentenceTransformers - это семейство предварительно обученных моделей Transformer, предназначенных для генерации векторных представлений предложений (также известных как вложения предложений).

В своей работе я использовал модель, которая использует архитектуру MiniLM и имеет 6 слоев. Она была обучена на большом количестве данных и может генерировать векторные представления предложений, которые хорошо работают для различных задач NLP.

Плюсы:

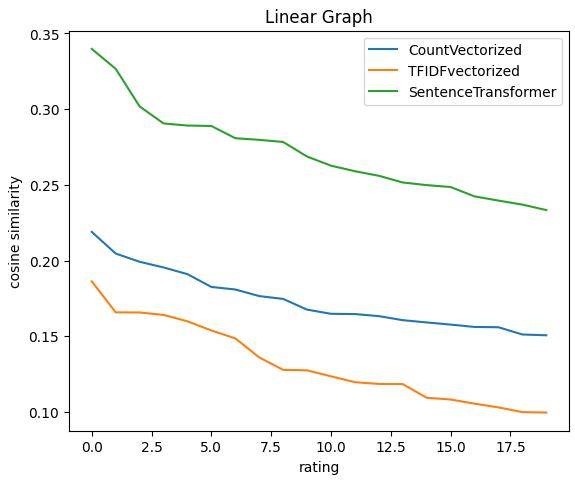
* Высокая точность: SentenceTransformer может точно предсказать, какие предложения будут похожи на данное предложение.
* Быстрая обработка: Модель может обрабатывать большое количество предложений в короткий промежуток времени, что делает ее идеальной для систем рекомендаций.
* Гибкость: Модель может быть использована для различных задач, включая поиск похожих предложений, классификацию предложений и многое другое.

Минусы:

* Требует больших объемов данных для обучения: Для того чтобы SentenceTransformer работал наилучшим образом, ему требуется большое количество обучающих данных.



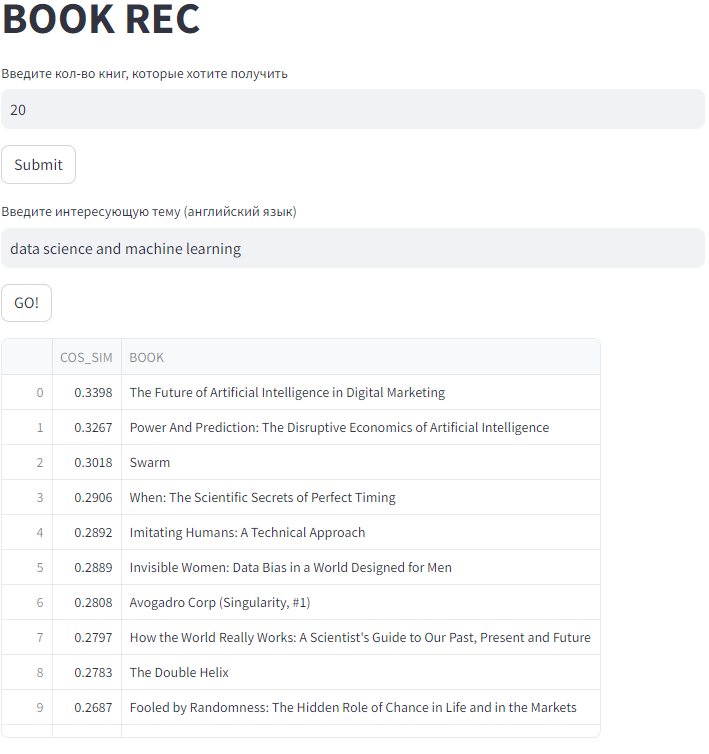
Лучший результат показывает SentenceTransformers

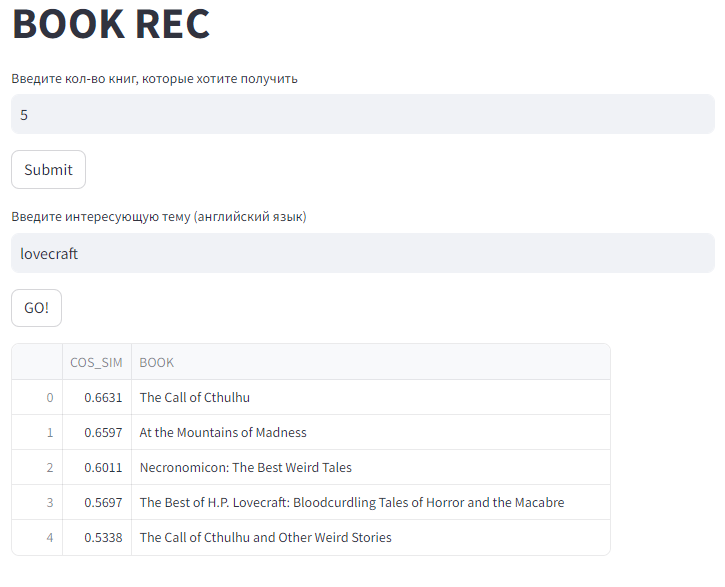


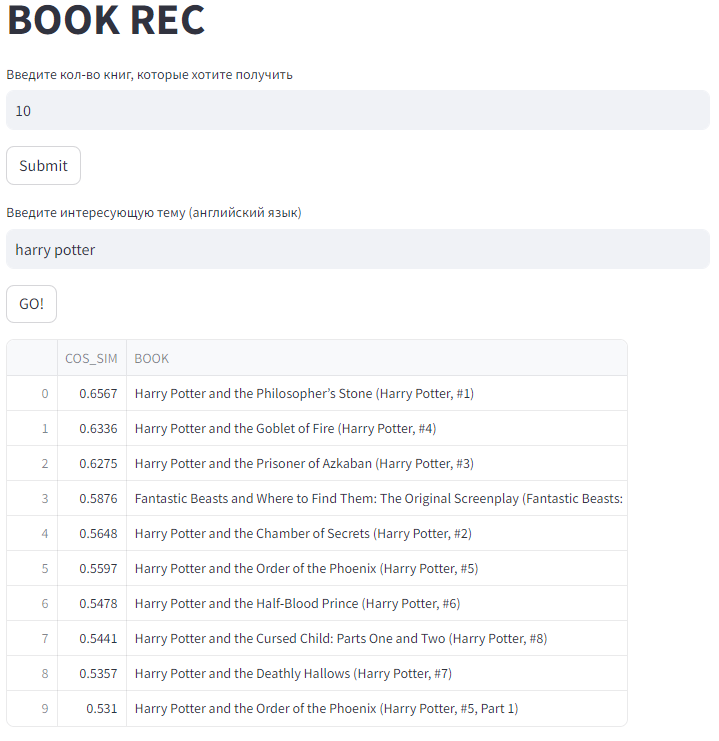
# Глава 3.Анализ эффективности работы веб-приложения.

Веб-приложение было создано с использованием Streamlit, платформы для создания интерактивных веб-приложений на языке программирования Python.

Данное приложение позволяет пользователям вводить количество книг, которые они хотели бы получить в качестве рекомендации, а также тему, автора, жанр или ключевые слова, которые их интересуют. После ввода всей необходимой информации пользователь может получить список литературы, которая максимально соответствует его запросу. Все данные отображаются в реальном времени, что позволяет пользователю быстро и легко получить нужную информацию.







Модель хорошо справляется со своей задачей. Смогла найти книги связанные с машинным обучением, по автору и по имени главного героя, плюс связанную с ним книгу.

# Заключение

В данной дипломной работе было разработано веб-приложение для подбора литературы на основе искусственного интеллекта. Приложение использует алгоритмы машинного обучения для анализа запросов пользователей и предлагает наиболее подходящую литературу по заданным темам.

Разработанное веб-приложение имеет ряд преимуществ по сравнению с существующими системами поиска литературы. Оно обладает более высокой точностью результатов, простым и понятным интерфейсом.

Результаты тестирования показали, что разработанное веб-приложение эффективно справляется с задачами подбора литературы. Оно способно обрабатывать различные типы запросов и выдавать релевантные результаты, что делает его полезным инструментом для исследователей, студентов и всех, кто занимается поиском информации.

Однако, несмотря на все преимущества, веб-приложение все еще требует доработки и улучшения. В будущих версиях планируется добавить возможность сортировки результатов по релевантности и другим параметрам, а также расширить функционал приложения для более точной обработки запросов.

В целом, данная дипломная работа демонстрирует возможность использования искусственного интеллекта для решения задач поиска информации и создания более эффективных систем обработки запросов.