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»

Отчёт по лабораторной работе №3 по дисциплине "Проектирование рекомендательных систем"

Тема Алгоритмы контентной фильтрации

Студент Варламова Е. А.

Группа ИУ7-33М

Оценка (баллы) _____

Преподаватели Быстрицкая А.Ю.

Москва — 2024 г.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

Цель работы – изучить TF-IDF и LDA.
Для достижения поставленной цели потребуется:

— привести описание алгоритмов;

— привести описание используемых для исследования данных;

— привести зависимости скорости и точности работы алгоритмов от объёма данных.

1 | Аналитический раздел

1.1 TF-IDF

TF-IDF (Term Frequency-Inverse Document Frequency) – это статистическая мера, используемая в информационном поиске и анализе текста для оценки важности слова в документе относительно всей коллекции документов. Эта мера может быть полезной и в рекомендательных системах для оценки сходства между элементами и пользователями. [1]

TF – частота слова, отношение числа вхождений некоторого слова к общему числу слов документа, так оценивается важность слова t_i в пределах отдельного документа:

$$tf(t,d) = \frac{n_t}{\sum_k n_k}, \tag{1.1}$$

где [15mm]

n_t число вхождений слова t в документ;

$\sum_k n_k$ общее количество слов в данном документе.

IDF – обратная частота документа, инверсия частоты, с которой некоторое слово встречается в документах коллекции.

$$IDF(t,D) = log\frac{|D|}{|\{d_i \in D | t \in d_i\}|}, \tag{1.2}$$

где [15mm]

$|D|$ число документов в коллекции;

$|\{d_i \in D | t \in d_i\}|$ число документов из коллекции D, в которых встречается t (когда $n_t \neq 0$).

Данная мера может быть использована в рекомендательных системах для:

— Представления контента, такого как текстовые описания товаров, фильмов или музыкальных треков; каждый объект (например, товар) будет представлен его описанием-вектором, в котором каждое слово представлено его

TF-IDF весом, что позволит понимать, какие слова играют важную роль в этом описании – выделить “тэги”;

— Определения сходства элементов и пользователя через косинусное сходство между векторами; элементы, чьи векторы более похожи на вектор пользователя, могут быть ему рекомендованы;

— Улучшения рекомендаций путем подсчета весовых коэффициентов для слов или фраз в профилях пользователей; если пользователь часто взаимодействует с элементами, содержащими определенные ключевые слова, то можно увеличить вес для этих слов в профиле пользователя;

— Модификации; TF-IDF может быть использован вместе с другими методами рекомендации, например, с коллаборативной фильтрацией, для улучшения точности и разнообразия рекомендаций.

При этом TF-IDF имеет некоторые ограничения: он не учитывает контекст слов и не способен обрабатывать синонимы. [1]

1.2 LDA

LDA (Latent Dirichlet Allocation) – это статистическая модель, используемая в анализе текстовых данных для выявления скрытых тем в коллекции документов. Данная модель предполагает, что каждый документ в коллекции создается путем комбинирования нескольких тем, и каждая тема представляет собой распределение слов. [2]

Данная модель может быть использована в рекомендательных системах для [2]:

— Извлечения тематических профилей – путем применения LDA к текстовым данным, можно извлечь тематические профили для каждого объекта, которые представляют собой вероятностные распределения тем в каждом элементе;

— Рекомендаций на основе тем – при наличии профилей объекта и пользователя, можно измерить сходство между темами и рекомендовать объекты, которые имеют близкие тематические профили к профилю пользователя;

— Разнообразия рекомендаций – LDA может помочь в улучшении разнообразия, так как модель позволяет контролировать количество тем;

— Персонализации – модель может быть адаптирована к поведению конкретного пользователя, чтобы улучшить качество рекомендаций.

2 | Конструкторский раздел

2.1 Kaggle IMDB dataset: movies

В качестве источника данных был взят датасет, располагающийся в свободном доступе на веб-сайте Kaggle. Датасет IMDB включает в себя описание фильмов и их жанр.

2.2 Предобработка данных

Для предобработки были проведены:

1. токенизация с приведением всего к нижнему регистру ;
2. удаление стоп-слов и знаков препинания;
3. лемматизация для уменьшения количества слов (удаление суффиксов и окончаний).

3 | Технологический раздел

3.1 Средства реализации

В качестве используемого был выбран язык программирования Python [3]. Данный выбор обусловлен следующими факторами:

— Большое количество исчерпывающей документации;

— Широкий выбор доступных библиотек для разработки;

— Простота синтаксиса языка и высокая скорость разработки.

При написании программного продукта использовалась среда разработки Visual Studio Code. Данный выбор обусловлен тем, что данная среда распространяется по свободной лицензии, поставляется для конечного пользователя с открытым исходным кодом, а также имеет большое число расширений, ускоряющих разработку.

3.2 Библиотеки

При анализе и обработке датасета, а также для решения поставленных задач использовались библиотеки:

— pandas;

— numpy;

— matplotlib [4];

— sklearn [5].

Данные библиотеки позволили полностью покрыть спектр потребностей при выполнении работы.

4 | Исследовательский раздел

4.1 Условия исследований

Исследование проводилось на персональном вычислительной машине со следующими характеристиками:

— процессор Intel Core i5,

— операционная система MacOS Big Sur

— 8 Гб оперативной памяти.

Временные затраты определялись с использованием библиотеки time.

На рисунке 4.1 представлено сравнение 4 алгоритмов в зависимости от количества документов (строк в таблице, где каждая строка – это описание фильма и его жанр) по времени работы и точности кластеризации:

— LDA из библиотеки sklearn;

— LDA, разработанный по описанию;

— TF-IDF из библиотеки sklearn + KMEANS из библиотеки sklearn;

— TF-IDF, разработанный по описанию, + KMEANS из библиотеки sklearn.

Видно, что LDA (собственный и библиотечный) работают дольше TF-IDF + KMEANS. При этом точность кластеризации у всех алгоритмов относительно невысокая (до 0.2).

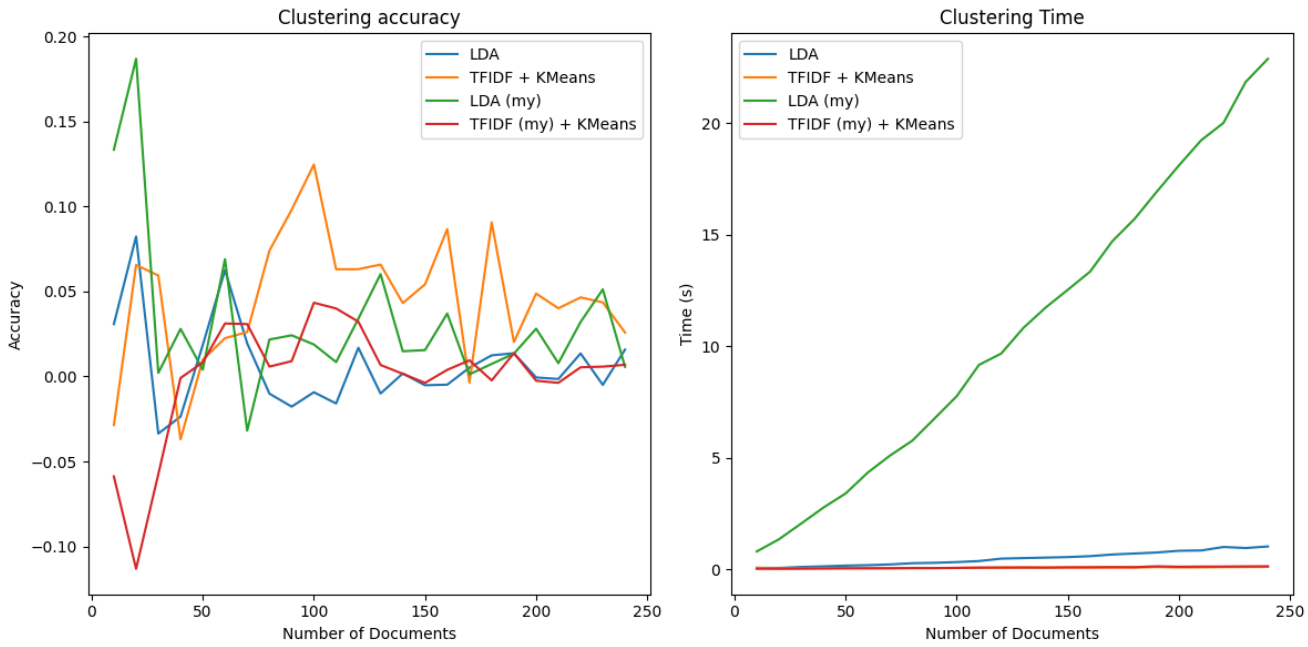

Рис. 4.1: Сравнение алгоритмов

ЗАКЛЮЧЕНИЕ

В ходе выполнения работы были изучены TF-IDF и LDA.
Для достижения поставленной цели были решены задачи:

— приведено описание алгоритмов;

— приведено описание используемых для исследования данных;

— приведены зависимости скорости и точности работы алгоритмов от объёма данных.

Проведенные исследования показали, что LDA (собственный и библиотечный) работают дольше TF-IDF + KMEANS. При этом точность кластеризации у всех алгоритмов относительно невысокая (до 0.2).

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. *Rajaraman A. U. J.* Data Mining. — 2011. — С. 1—17.
2. *Blei M. Ng Y. J. I.* Latent Dirichlet Allocation // Journal of Machine Learning Research. — 2003. — № 3.
3. Python official page [Электронный ресурс]. — Режим доступа: `https://www.python.org/` (дата обращения 10.05.2023).
4. Matplotlib official page [Электронный ресурс]. — Режим доступа: `https://matplotlib.org/` (дата обращения 10.05.2023).
5. Scikit-learn official page [Электронный ресурс]. — Режим доступа: `https://scikit-learn.org/stable/` (дата обращения 10.05.2023).