**Докумнентация за препоръчителна система за филми базирана на съдържание**

Проекта и документацията са изготвени от:

Матрин Иванов

Ф.Н. 319222015

Дисциплина: „ИЗД“

Специалност: Анализ на Данни

E-mail: martin03m2000@gmail.com

**Съдържание:**

1. Теория
   1. Информация
   2. Използвани алгоритми
   3. Инструменти за реализания на алгоритмите
2. Пример
3. Експеримент

**1. Теория**

1.1 Информация

Проектът представлява система за препоръчване на филми, базирана на съдържанието, предлага филми на потребителите въз основа на характеристиките на филмите. Системата използва характеристики на филмите като жанр, режисьор, актьори, ключови думи за сюжета или дори специфични теми, за да подбере филми.

Някои от предимствата на препоръчителна система за филми, базирана на съдържанието:

* **Няма нужда от данните на потребителите**
* **Ефективна дори при нови или не толкова популярни филми** ( стига да имат добре описана инф.)
* **Не бива засегната от „cold-start“ проблем**

1.2 Използвани алгоритми

**Портър Стемър ( Porter Stemmer )**

* Опростяване на думи чрез премахване на суфикси, за да се получи основната им форма
* Работи само на английски език
* Работи с поредица от предварително дефинирани правила в пет стъпки
  + Всяка стъпка има набор от суфикси се преработван или отстраняват

**Count Vectorization**

* Преобразува колекция от текстови документи в цифрово представяне
* Как работи:
  + Разделяне на текста на думи (токени)
  + Изготвяне на списък с уникални думи от всички документи
  + За всеки документ се брои колко пъти се появява всяка дума от речника
  + Създава се матрица, в която всеки ред е документ, всяка колона е дума от речника, а стойностите са броят на думите ( Document-Term Matrix )

**Косинусова Сходимост**

* Използва се за изчисляване на сходството между два вектора ( в нашият случай използваме матрицата от CountVectorization)
* Формула: cos(θ) = A . B / ( ||A|| . ||B|| )
* Варира от -1 (напълно противоположен) до 1 (идентичен)
* Отчита само посоката на векторите, а не големината ( полезно при сравняване на документи с различни дължини)

**2. Пример**

2.1 Основни стъпки по задачата

* Разархивиране и прочитане на данните
* Преработване на данните
* Векторизиране на избраните ключови данни
* Изчисляване на косинусовата сходимост между данните
* Създаване на фукция, която препоръчва спямо подадено заглавие на филм

2.2 Описание на данните

Данните са разделени на 2 файла

* tmdb\_5000\_movies.csv – съдържа описанието и характеристиките на филмите
  + Важни за нас колони: 'movie\_id', 'title', 'overview', 'genres', 'keywords'
* tmdb\_5000\_credits.csv – съдържа инф. за акьорите и екира, направили определен филм
  + Важни за нас колони: 'cast','crew'

2.3 Преработка на данните

1. Обединяване на двете таблици по заглавието на филма
2. Премахване на редове със празни стойности
3. Някои от колоните съхраняват информаци под JSON файл формат, затова ги десереализираме и взимаме само важните за нас данни
   1. От колоните 'genres' и 'keywords' взимаме свойствата който имат име 'name'
   2. От колоната 'cast' взимаме имената на първите трима актьори
   3. От колоната 'crew' взимаме името на режисьора
4. Накрая обединяваме всички колони, без 'movie\_id' и 'title' под една обща колона 'tags'

2.4 Моделиране на данните

1. Използване на стемер за опростяване на думите в 'tags' колоната
2. Прилагане на CountVectorization
3. Изчисляване на косинусовата сходимост между филмите
4. Създаване на фукцията за препоръчване