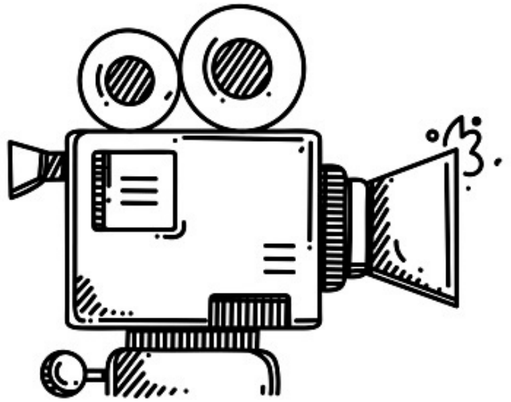
Movie Recommendation System

Habib Ja’far Nuur



Yogyakarta,30 Mei 2024

Contents

Introduction

Dalam era digital saat ini industri perfilman telah mengalami perkembangan pesat yang didorong oleh kemajuan teknologi informasi. Layanan streaming seperti Netflix dan Disney+ menjadi salah satu pilihan dalam menonton film. Hal ini menjadi masalah bagi pengguna untuk menemukan film yang sesuai dengan preferensi mereka.

Di Perlukan Sistem Rekomendasi

Pengguna

Pengguna dapat menemukan konten yang sama dengan preferensi mereka

Penyedia Layanan

Dengan meningkatkan kepuasan pengguna, layanan streaming dapat mempertahankan pelanggannya dan meningkatkan engagement.

Contents

Term Frequency

all	can	day	do	feel	fun	hate	learn
0	0	0	0	0	2	0	1
1	1	1	1	0	0	0	0
0	0	0	0	1	0	1	0

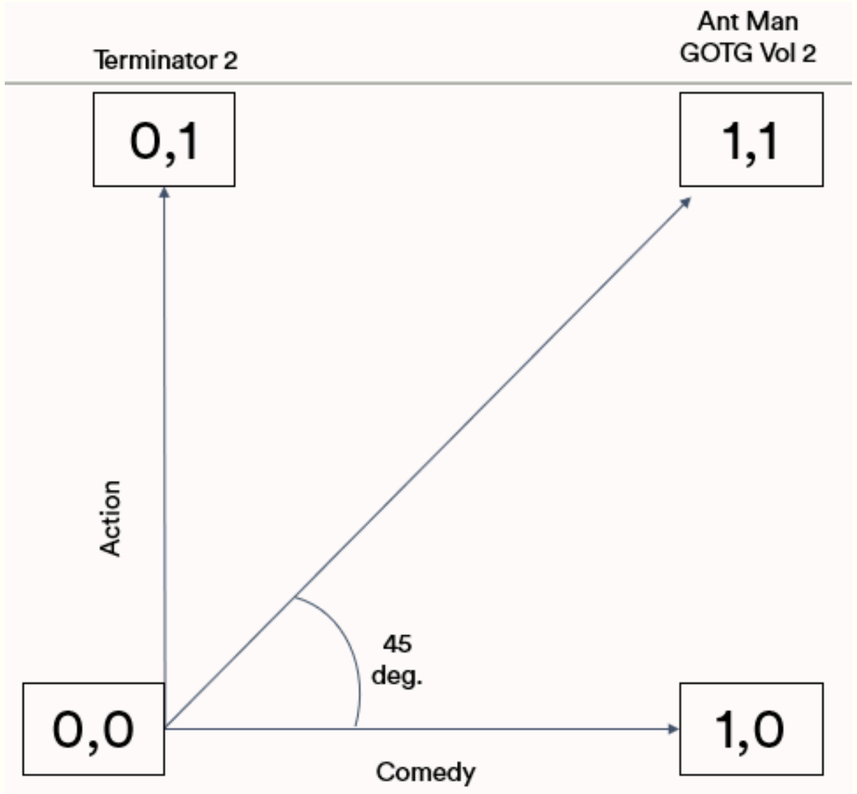
Term Frequency adalah mmetode yang digunakan untuk mengubah kata menjadi angka dengan menghitung frequency kemunculan kata pada setiap document

Movie_recomendation_system.ipynb

```
1  from sklearn.feature_extraction.text import CountVectorizer
2
3  tf =CountVectorizer()
4  tf_matrix = tf.fit_transform(df['listed_in'])
5  tf_matrix.shape
```

Snipped

Cosine Similarity



Melakukan pengukuran kemiripan/similarity berdasarkan kata yang telah dilakukan term frequency

Movie_recomendation_system.ipynb

```
1   from sklearn.metrics.pairwise import linear_kernel
2
3   similarity = linear_kernel(tf_matrix, tf_matrix)
```

Snipped

Recommendation Class

Movie_recomendation_system.ipynb

```python
def recommendations(title, similarity=similarity):
    # Get the index of the movie that matches the title
    try:
        idx = df[df['title'] == title].index[0]
        # mendapatkan similarity
        sim_score = list(enumerate(similarity[idx]))

        # mengurutkan film berdasarkan similarity
        sim_score = sorted(sim_score, key=lambda x: x[1], reverse=True)

        # mendapatkan 5 film yang memiliki similarty
        sim_score = sim_score[1:6]

        # mendapatkan index dari movie
        index_movie = [i[0] for i in sim_score]

        # mengembalikan 5 film yang memiliki similarity
        return df.iloc[index_movie][['title', 'type', 'country','listed_in']]

    except IndexError:
        print(f"Title '{title}' not found in the dataset.")
        return []
```

Snipped

Membuat Class/fungsi untuk melakukan rekomendasi dengan berdasarkan similarity score

Output

```
recommendations('Dick Johnson Is Dead')
```

	title	type	country	listed_in
16	Europe's Most Dangerous Man: Otto Skorzeny in ...	Movie	NaN	Documentaries, International Movies
45	My Heroes Were Cowboys	Movie	NaN	Documentaries
68	Schumacher	Movie	NaN	Documentaries, International Movies, Sports Mo...
88	Blood Brothers: Malcolm X & Muhammad Ali	Movie	NaN	Documentaries, Sports Movies
91	The Women and the Murderer	Movie	France	Documentaries, International Movies

Sistem memberikan 5 recomendasi film berdasarkan similarity dari genre film

Contents

Conclusions

Conclusion 01

sistem rekomendasi ini dapat memangkas waktu pengguna dalam mencari film baru berdasarkan persamaan preferensi

Conclusion 02

Sistem Rekomendasi dapat meningkatkan engagement bagi penyedia layanan streaming film

Conclusion 03

film-film yang kurang populer memiliki peluang lebih besar untuk ditemukan oleh pengguna yang mungkin memiliki preferensi genre yang sama

Thank you

https://medium.com/@habibjafar08