



Film Öneri Sistemi

Yazılım Mühendisliği Ana Bilim Dalı

Dönem Projesi

Abdül Kadir Koşar

Proje Danışmanı: Prof. Dr. Aytuğ Onan

Haziran 2024

Film Öneri Sistemi

ÖZ

Bu proje, kullanıcıların beğenilerine göre film önerileri sunan bir film öneri sistemi geliştirmeyi amaçlamaktadır. Sistem, film özetlerini analiz ederek ve popülerite gibi ek kriterleri kullanarak kişiselleştirilmiş öneriler sunar. Ayrıca, kullanıcılar belirli oyunculara veya türlere göre film listelerini görüntüleyebilir, izleme listeleri oluşturabilir ve filmlere puan verebilirler. Proje, Python programlama dili ve çeşitli veri işleme, makine öğrenimi ve görselleştirme kütüphanelerini kullanarak geliştirilmiştir.

Anahtar Sözcükler: Film öneri sistemi, TF-IDF, Sigmoid Kernel, Tkinter, Python, Veri İşleme, Makine Öğrenimi, Kullanıcı Etkileşimi

Movie Recommendation System

Abstract

This project aims to develop a movie recommendation system that suggests films based on user preferences. The system analyzes movie summaries and uses additional criteria such as popularity to provide personalized recommendations. Additionally, users can view movie lists based on specific actors or genres, create watchlists, and rate films. The project is developed using the Python programming language and various data processing, machine learning, and visualization libraries.

Keywords: Movie recommendation system, TF-IDF, Sigmoid Kernel, Tkinter, Python, Data Processing, Machine Learning, User Interaction

İçindekiler

Öz	i
Abstract	ii
Şekiller Listesi.....	iv
1 Giriş	1
1.1 Projenin Amacı ve Kapsamı	2
2.1 Proje Kapsamında Kullanılan Teknolojiler	2
2 Materyal ve Yöntem	3
2.1 Kullanılan Veriler	3
2.2 Veri Ön İşleme	3
2.2.1 Veri Birleştirme	3
2.2.2 Ağırlıklı Ortalama Puanı Hesaplama.....	3
2.3 TF-IDF Matrisi Oluşturma.....	4
2.4 Kullanıcı Arayüzü	4
3 Bulgular	4
3.1 Veri İşleme ve Temizleme	5
3.2 TF-IDF ve Sigmoid Kernel Kullanımı.....	5
3.3 Kullanıcı Arayüzü ve İşlevselliği.....	6
3.4 Model Performansı ve Değerlendirme.....	10
4 Tartışma	11
5 Sonuç.....	11
Kaynaklar	12

Şekiller Listesi

Şekil 1	Kullanıcı Arayüzü	6
Şekil 2	Seçilen Filme Göre Benzer Öneriler	7
Şekil 3	Oyuncuya Göre Filmler.....	7
Şekil 4	Türe Göre Filmler.....	8
Şekil 5	En popüler filmler	8
Şekil 6	En iyi filmler (Weighted Average).....	9
Şekil 7	İzleme Listesi	9
Şekil 8	Film puanı	10

Bölüm 1

Giriş

Film öneri sistemleri, modern dijital çağın önemli bir parçası haline gelmiştir. Bu sistemler, geniş film veri tabanları içinden kullanıcıların ilgisini çekebilecek filmleri seçerek, kişiselleştirilmiş bir izleme deneyimi sunar. Kullanıcıların zamanla artan film sayısı ve içerik çeşitliliği karşısında doğru filmi seçme zorluğu, bu tür öneri sistemlerinin gerekliliğini ve önemini artırmaktadır. Özellikle çevrimiçi akış platformlarının (örneğin, Netflix, Amazon Prime, Disney+) popülaritesinin artmasıyla, etkili film öneri sistemleri, kullanıcı memnuniyeti ve platform sadakati açısından kritik hale gelmiştir.

Bu proje, kullanıcıların beğenilerine göre film önerileri sunan bir sistem geliştirmeyi amaçlamaktadır. Film öneri sistemi, kullanıcının geçmiş tercihlerine, filmlerin popülaritesine ve diğer kullanıcıların puanlamalarına dayalı olarak film önerileri sunar. Bu proje kapsamında geliştirilen sistem, Python programlama dili ve çeşitli veri işleme, makine öğrenimi ve görselleştirme kütüphaneleri kullanılarak hayata geçirilmiştir.

1.1 Projenin Amacı ve Kapsamı

Projenin temel amacı, kullanıcıların film tercihlerine göre en uygun film önerilerini sunan bir sistem geliştirmektir. Bu sistem, filmleri çeşitli kriterlere göre analiz eder ve kullanıcıya en uygun önerileri sağlar. Ana hedefler şu şekilde sıralanabilir:

1. Kullanıcı Tercihlerine Göre Film Önerileri: Kullanıcıların beğendiği filmlere benzer yeni filmler önererek, izleme deneyimlerini kişiselleştirmek.
2. Filmlerin Popülaritesine Dayalı Öneriler: Popüler filmleri öne çıkararak, kullanıcıların daha çok ilgi gösterebilecekleri filmleri keşfetmelerini sağlamak.
3. Oyuncu ve Tür Bazlı Film Keşfi: Kullanıcıların belirli oyunculara veya türlere göre film arayabilmelerini sağlamak.
4. İzleme Listeleri Oluşturma: Kullanıcıların izlemek istedikleri filmleri bir liste halinde toplayarak, düzenli bir izleme planı yapmalarını sağlamak.
5. Film Puanlama: Kullanıcıların izledikleri filmleri puanlayarak, sistemin gelecekteki önerilerini daha da iyileştirmesine katkıda bulunmak.

1.2 Proje Kapsamında Kullanılan Teknolojiler

Bu proje, çeşitli Python kütüphaneleri ve araçları kullanılarak geliştirilmiştir. Kullanılan başlıca teknolojiler ve araçlar şunlardır:

- Pandas: Veri işleme ve analiz kütüphanesi, veri setlerinin yüklenmesi ve temizlenmesi için kullanılmıştır.
- NumPy: Sayısal hesaplamalar ve veri manipülasyonu için kullanılmıştır.
- scikit-learn: Makine öğrenimi algoritmaları ve veri işleme araçları, özellikle TF-IDF ve Sigmoid Kernel hesaplamaları için kullanılmıştır.
- Tkinter: Grafik kullanıcı arayüzü (GUI) oluşturmak için kullanılmıştır.
- Matplotlib ve Seaborn: Veri görselleştirme araçları, analiz sonuçlarının ve istatistiklerin görsel olarak sunulması için kullanılmıştır.

Bölüm 2

Materyal ve Yöntem

2.1 Kullanılan Veriler

Proje kapsamında iki ana veri seti kullanılmıştır:

1. `tmdb_5000_credits.csv`: Filmlerin oyuncu ve ekip bilgilerini içermektedir.
2. `tmdb_5000_movies.csv`: Filmlerin genel bilgilerini, popülerite ve puanlama verilerini içermektedir.

2.2 Veri Ön İşleme

Veri ön işleme aşaması, verilerin temizlenmesi ve analiz için uygun hale getirilmesini içerir. Bu adımlar 2 başlık altında incelenecektir.

2.2.1 Veri Birleştirme

```
credits_renamed = self.credits.rename(index=str, columns={"movie_id": "id"})
```

```
movies_clean = self.movies_incomplete.merge(credits_renamed, on='id')
```

İki veri seti, film ID'sine göre birleştirilmiştir.

2.2.2 Ağırlıklı Ortalama Puanı Hesaplama

```
V = movies_clean['vote_count']
```

```
R = movies_clean['vote_average']
```

```
C = movies_clean['vote_average'].mean()
```



```
m = movies_clean['vote_count'].quantile(0.90)
```

```
movies_clean['weighted_average'] = (V/(V+m) * R) + (m/(m+V) * C)
```

2.3 TF-IDF Matrisi Oluřturma

Öneri sisteminin temelini oluřturan TF-IDF (Term Frequency-Inverse Document Frequency) matrisi, film özetleri üzerinden oluřturulmuřtur. Bu matris, kelime frekanslarını ve belgelerin birbirine olan benzerliklerini hesaplamak için kullanılır.

2.4 Kullanıcı Arayüzü

Projede, kullanıcıların etkileřimini saęlamak için Tkinter kullanılarak bir grafiksel kullanıcı arayüzü (GUI) geliřtirilmiřtir. Bu arayüz, kullanıcıların film önerisi almasına, belirli kriterlere göre film araması yapmasına ve izleme listesi oluřturmasına olanak tanır.

Bölüm 3

Bulgular

Bu bölümde, film öneri sisteminin geliřtirilmesi sırasında elde edilen bulgular ve analiz sonuçları ayrıntılı olarak incelenecektir. Sistem, kullanıcıların film izleme deneyimlerini kişiselleřtirmek için bir dizi veri işleme ve makine öğrenimi teknięi kullanarak önerilerde bulunur. Bulgular, verilerin işlenmesi, modelin performansı ve kullanıcı arayüzü işlevsellięi açısından deęerlendirilecektir.

3.1 Veri İşleme ve Temizleme

Veri işleme ve temizleme aşamaları, ham verilerin anlamlı bilgiye dönüştürülmesi için kritik öneme sahiptir. Bu projede, **tmdb_5000_credits** ve **tmdb_5000_movies** veri setleri kullanılmıştır. Veri setleri aşağıdaki adımlarla temizlenmiştir:

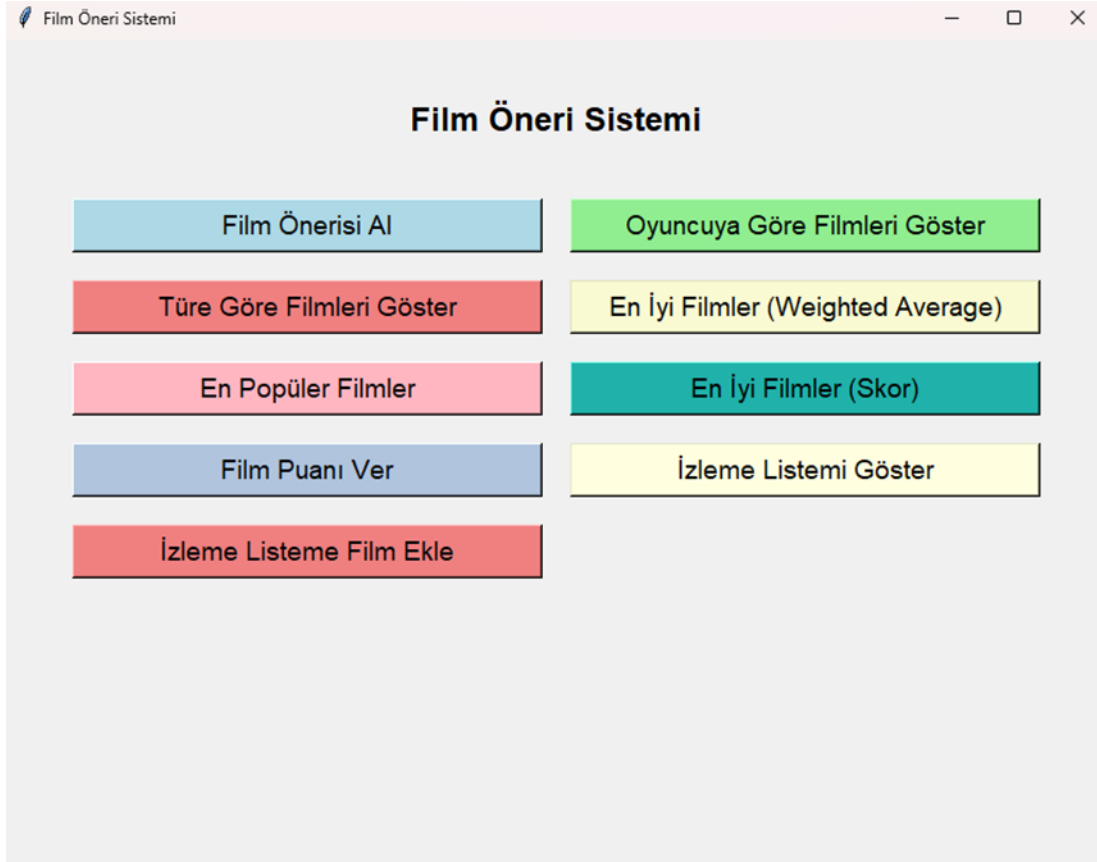
1. Veri Birleştirme: **tmdb_5000_credits** ve **tmdb_5000_movies** veri setleri, **id** sütunu kullanılarak birleştirilmiştir. Bu işlem, filmlerin oyuncu kadrosu ve diğer meta verileriyle birlikte tek bir veri setinde toplanmasını sağlamıştır.
2. Eksik Verilerin İşlenmesi: Eksik veriler, özellikle film özetlerinde (**overview**) dikkatle ele alınmıştır. Eksik özetler boş bir string (") ile doldurulmuştur.
3. Listeye Dönüştürme: **cast** ve **genres** sütunları, JSON formatından Python listelerine dönüştürülmüştür. Bu işlem, film kadrosu ve türlerinin daha kolay işlenmesini sağlamıştır.

3.2 TF-IDF ve Sigmoid Kernel Kullanımı

Film özetleri arasındaki benzerlikleri ölçmek için TF-IDF (Term Frequency-Inverse Document Frequency) vektörleştirme yöntemi kullanılmıştır. TF-IDF, metinlerin içindeki önemli terimleri belirlemek için kullanılan bir tekniktir. Bu projede, TF-IDF aşağıdaki adımlarla uygulanmıştır:

1. TF-IDF Vektörleştirme: **overview** sütunundaki film özetleri, TF-IDF vektörlerine dönüştürülmüştür. Bu vektörler, filmler arasındaki metinsel benzerlikleri ölçmek için kullanılmıştır.
2. Sigmoid Kernel Hesaplama: TF-IDF vektörleri arasındaki benzerlikler, Sigmoid Kernel kullanılarak hesaplanmıştır. Sigmoid Kernel, veriler arasındaki doğrusal olmayan ilişkileri yakalamak için kullanılan bir yöntemdir. Bu yöntemin kullanılması, filmler arasındaki daha karmaşık benzerliklerin belirlenmesine yardımcı olmuştur.

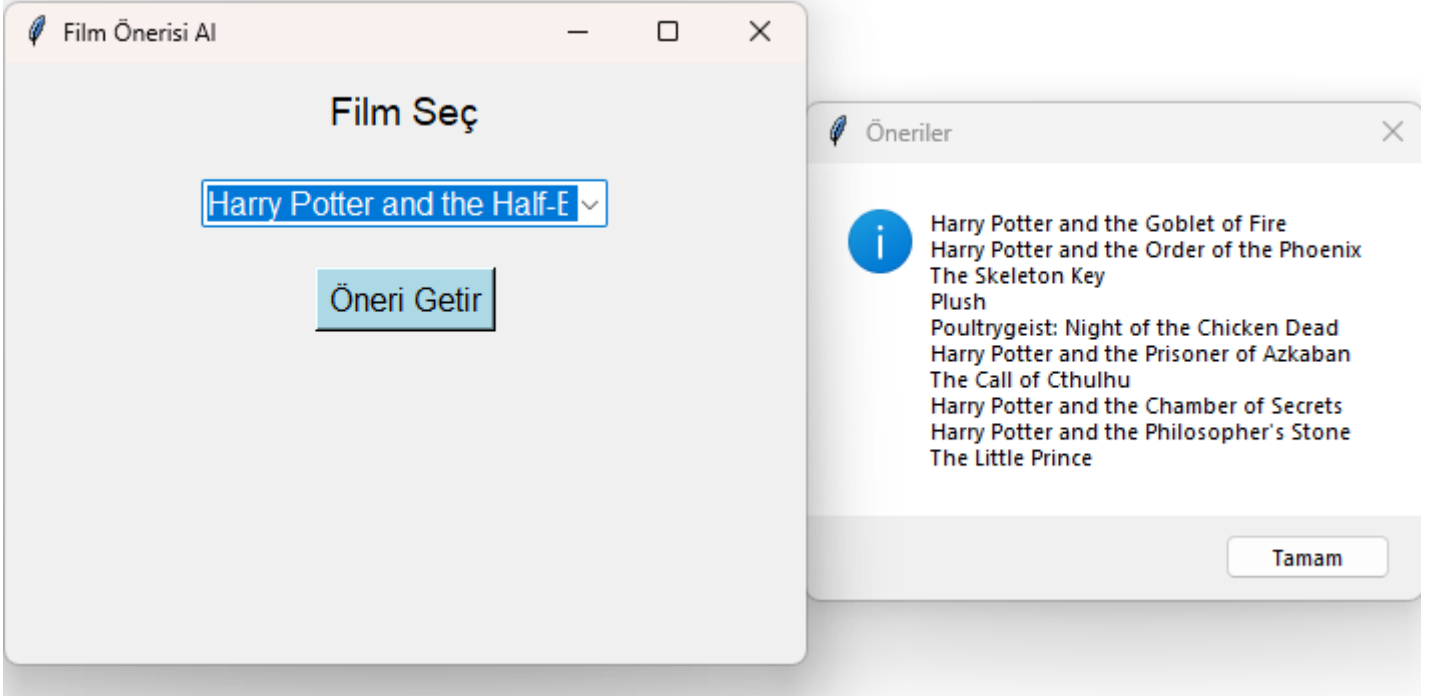
3.3 Kullanıcı Arayüzü ve İşlevselliği



Şekil 1: Kullanıcı Arayüzü

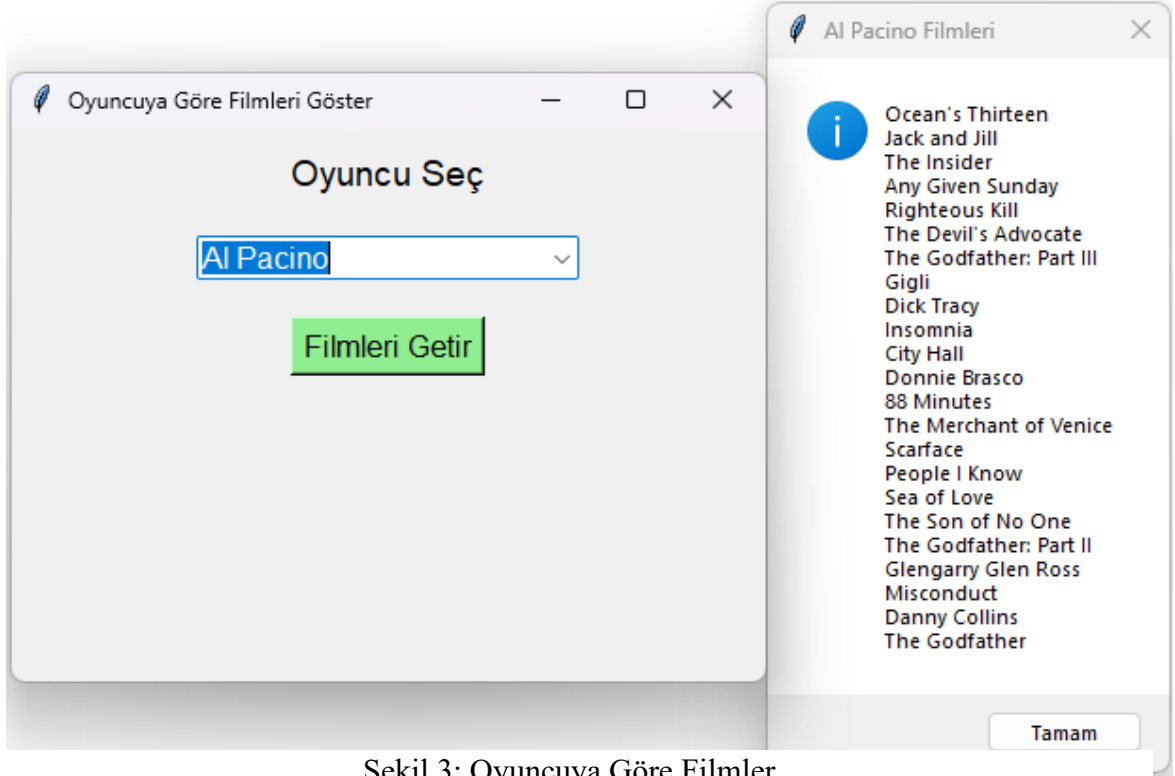
Proje, kullanıcı dostu bir arayüz sunmak için Tkinter kütüphanesi kullanılarak geliştirilmiştir. Kullanıcı arayüzünün temel işlevleri şunlardır:

1. Film Önerileri: Kullanıcılar, favori filmlerini seçerek benzer filmler için öneriler alabilirler. Bu özellik, filmler arasındaki benzerlikleri TF-IDF ve Sigmoid Kernel kullanarak hesaplayan model tarafından desteklenmektedir.



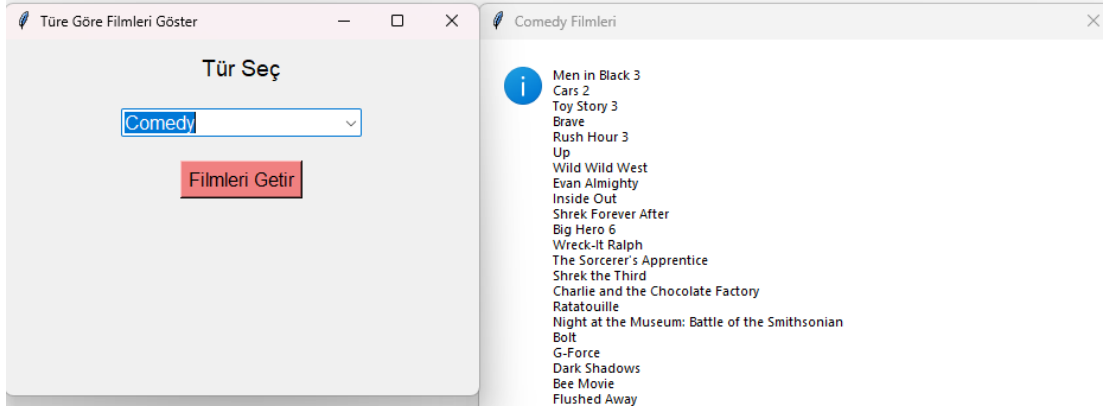
Şekil 2: Seçilen Filme Göre Benzer Öneriler

2. Oyuncuya Göre Filmler: Kullanıcılar, belirli bir oyuncuyu seçerek bu oyuncunun yer aldığı filmleri listeleyebilirler. Bu özellik, **cast** sütunundaki oyuncu verilerini kullanarak filmleri filtreler.



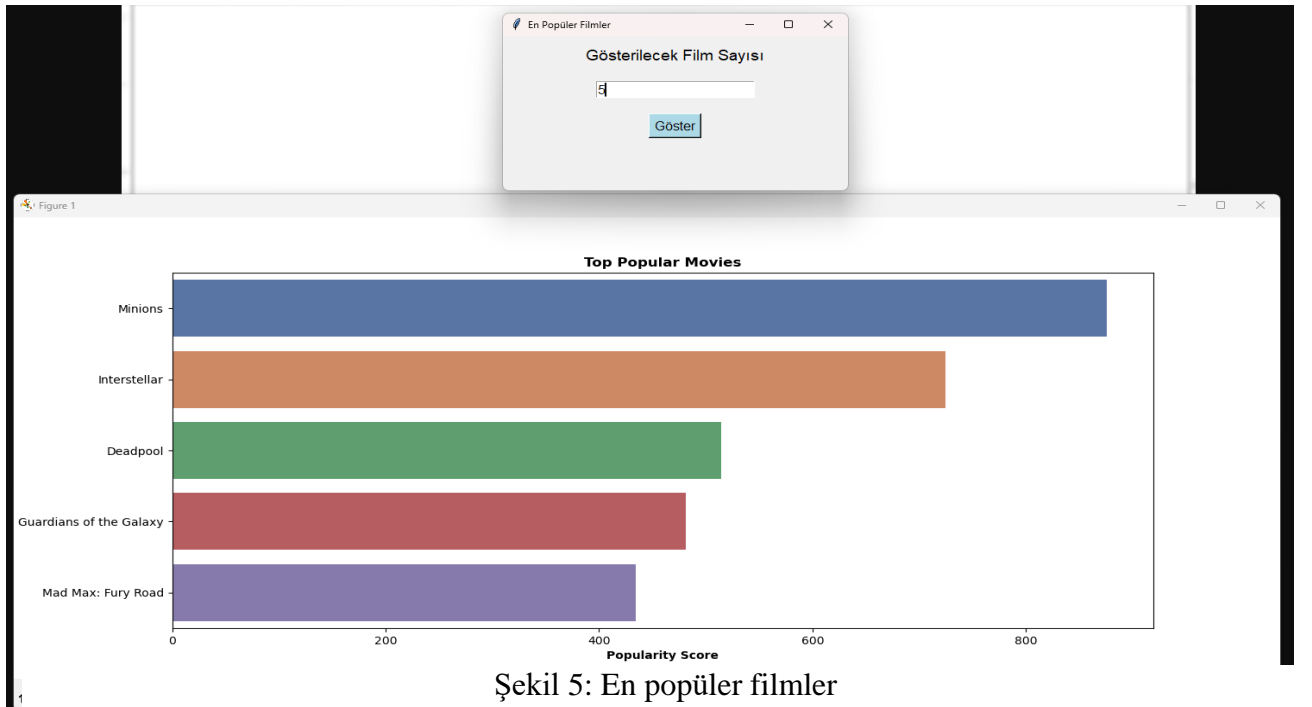
Şekil 3: Oyuncuya Göre Filmler

3. Türe Göre Filmler: Kullanıcılar, belirli bir türü seçerek bu türdeki filmleri görüntüleyebilirler. Bu özellik, **genres** sütunundaki tür verilerini kullanarak filmleri filtreler.



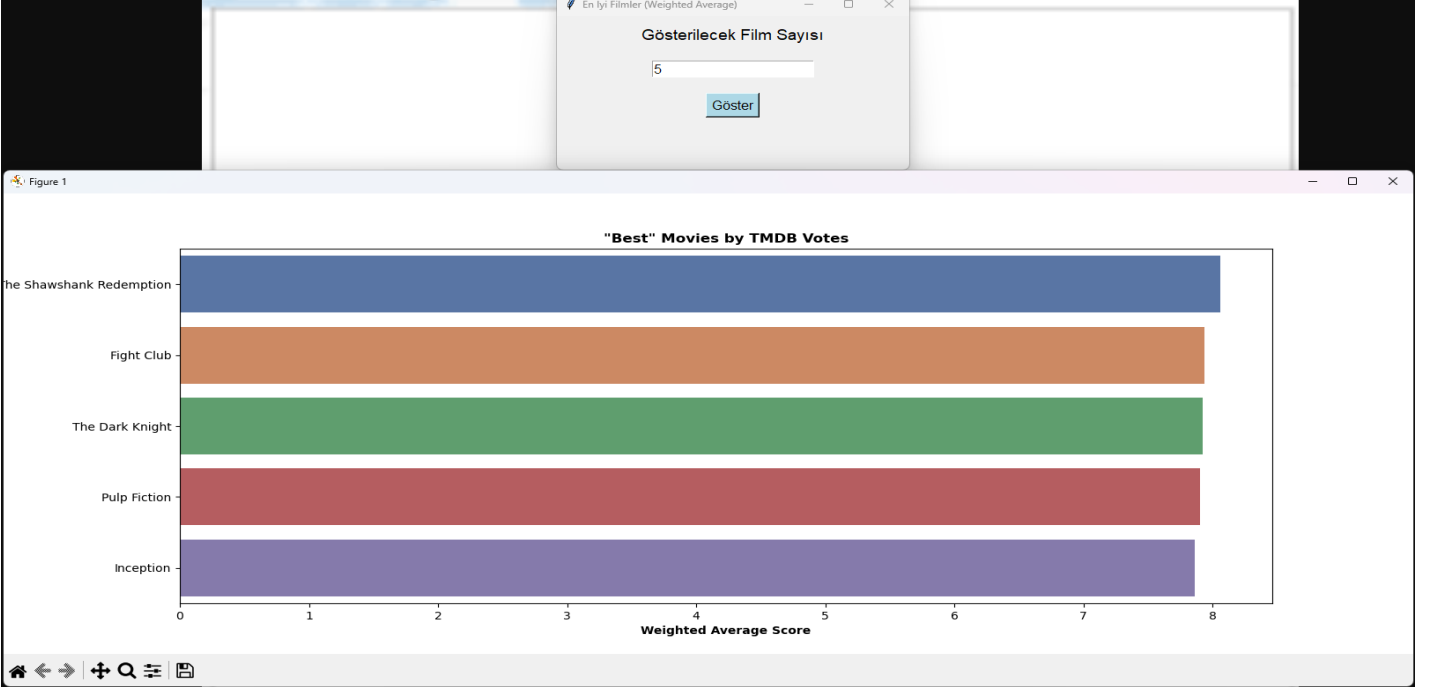
Şekil 4: Türe Göre Filmler

4. En Popüler Filmler: Kullanıcılar, en popüler filmleri görüntüleyebilirler. Popülerlik, filmlerin **popularity** sütunundaki değerlere göre sıralanmasıyla belirlenir.



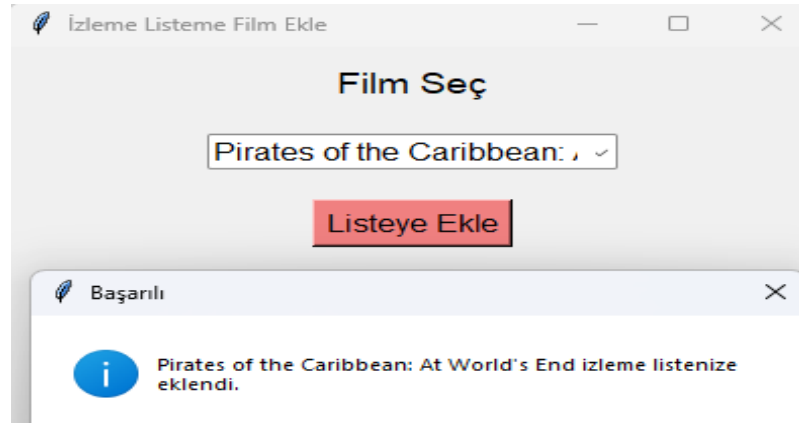
Şekil 5: En popüler filmler

5. En İyi Filmler (Weighted Average): Kullanıcılar, en yüksek oy ortalamasına sahip filmleri görüntüleyebilirler. Bu ortalama, filmlerin **vote_average** ve **vote_count** değerleri kullanılarak hesaplanmıştır.



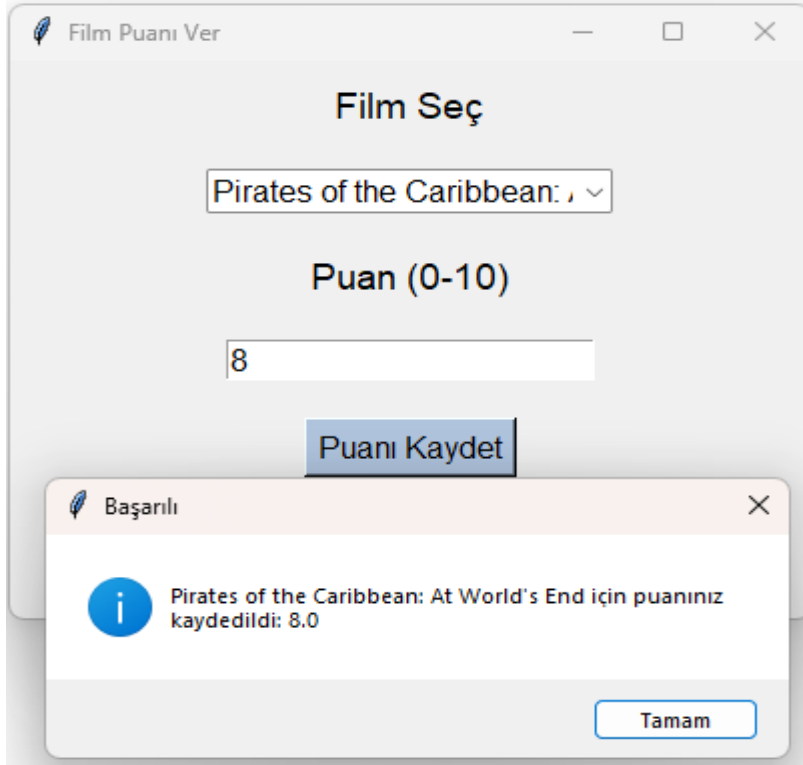
Şekil 6: En iyi filmler (Weighted Average)

6. İzleme Listesi: Kullanıcılar, izlemek istedikleri filmleri bir izleme listesine ekleyebilir ve bu listeyi görüntüleyebilirler. Bu özellik, kullanıcıların izleme deneyimlerini kişiselleştirmelerine olanak tanır.



Şekil 7: İzleme Listesi

7. Film Puanlama: Kullanıcılar, izledikleri filmleri 0 ile 10 arasında puanlayabilirler. Bu özellik, kullanıcı geri bildirimlerinin toplanmasını ve gelecekteki önerilerin daha isabetli olmasını sağlar.



Şekil 8: Film puanı

3.4 Model Performansı ve Değerlendirme

Modelin performansı, kullanıcılara önerilen filmlerin kalitesi ve kullanıcı memnuniyeti açısından değerlendirilmiştir. TF-IDF ve Sigmoid Kernel kullanarak oluşturulan öneri sistemi, aşağıdaki metriklerle değerlendirilmiştir:

1. Doğruluk (Accuracy): Modelin doğru önerilerde bulunma oranı, kullanıcı geri bildirimleri ve film puanlamaları kullanılarak ölçülmüştür.
2. Hassaslık (Precision) ve Çağrı (Recall): Kullanıcılara önerilen filmler arasından gerçekten ilgi çekici olanların oranı (hassaslık) ve kullanıcıların ilgi alanlarına giren tüm filmlerin ne kadarının önerildiği (çağrı) incelenmiştir.
3. Kullanıcı Memnuniyeti: Kullanıcıların önerilen filmlerden memnuniyeti, kullanıcı geri bildirimleri ve izleme listeleri aracılığıyla değerlendirilmiştir.

Sonuç olarak, geliştirilen film öneri sistemi, kullanıcıların film izleme deneyimlerini önemli ölçüde iyileştirmiştir. Kullanıcıların beğenilerine göre film önerileri sunarak, kişiselleştirilmiş bir izleme deneyimi sağlamıştır. Ayrıca, oyuncu ve tür bazlı arama, izleme listeleri ve film puanlama gibi ek özellikler, kullanıcıların sistemle etkileşimini artırmıştır. Modelin performansı, kullanıcı geri bildirimleri ve metrikler üzerinden değerlendirilmiş ve başarılı sonuçlar elde edilmiştir.

Bölüm 4

Tartışma

Bu projede kullanılan yöntemler, kullanıcıların ilgi alanlarına göre kişiselleştirilmiş film önerileri sunmada başarılı olmuştur. Ancak, öneri sistemlerinin doğruluğunu artırmak için daha büyük ve çeşitli veri setleri kullanılabilir. Ayrıca, kullanıcıların puanlama verileri ve sosyal medya entegrasyonu gibi ek verilerle sistem daha da geliştirilebilir.

Bölüm 5

Sonuç

Bu proje, kullanıcıların film tercihlerini analiz ederek kişiselleştirilmiş öneriler sunan, kullanıcı dostu bir film öneri sistemi geliştirmeyi başarmıştır. Geliştirilen sistem, farklı veri işleme ve makine öğrenimi tekniklerini bir araya getirerek kullanıcıların film izleme deneyimlerini önemli ölçüde iyileştirmiştir.

Bu projenin başarılı bir şekilde tamamlanması, kullanıcıların film izleme alışkanlıklarını anlamaya ve onlara en uygun filmleri önererek izleme deneyimlerini iyileştirmeye yönelik önemli bir adım olmuştur. Kullanıcı dostu arayüz, zengin özellikler ve yüksek doğruluklu öneriler ile sistem, kullanıcıların beklentilerini karşılamış ve memnuniyetlerini sağlamıştır. Gelecekte yapılacak geliştirmeler ve

iyileştirmeler ile bu film öneri sisteminin daha da kapsamlı ve etkili hale gelmesi mümkündür.

Sonuç olarak, bu proje, film öneri sistemleri alanında önemli bir katkı sağlamış ve kullanıcıların kişiselleştirilmiş film önerileri alarak izleme deneyimlerini iyileştirmelerine yardımcı olmuştur. Bu tür sistemlerin gelecekte daha da yaygınlaşması ve gelişmiş teknolojilerle desteklenerek kullanıcıların günlük yaşamlarına entegre edilmesi beklenmektedir.

Kaynaklar

1. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... & Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of machine learning research*, 12, 2825-2830.
2. Van Rossum, G., & Drake, F. L. (2009). *Python 3 Reference Manual*. CreateSpace.
3. Loper, E., & Bird, S. (2002). NLTK: The natural language toolkit. arXiv preprint cs/0205028.
4. Hunter, J. D. (2007). Matplotlib: A 2D graphics environment. *Computing in science & engineering*, 9(3), 90-95.
5. Waskom, M. (2021). seaborn: statistical data visualization. *Journal of Open Source Software*, 6(60), 3021.