YAZILIM GELİŞTİRME I DERSİ
PROJE RAPORU

İzzet Esener-210229048
Yazılım Mühendisliği
Kocaeli Üniversitesi
Kocaeli, Türkiye

1.Giriş

Bu projede, kullanıcıların film izleme ve beğeni eğilimlerini analiz ederek genel ve kişiye özel önerilerde bulunmayı hedefleyen bir film öneri sistemi tasarlanmıştır. MovieLens 20M veri seti kullanılarak gerçekleştirilen bu proje, birliktelik kuralları kullanarak diğer kullanıcıların beğenilerine göre popüler veya kişinin kendi beğenilerine göre film önerme amacını taşımaktadır.

Projenin Amacı

Bu projenin amacı, büyük bir kullanıcı kitlesinin film izleme ve beğeni eğilimlerini analiz ederek kullanıcılara daha kişisel, anlamlı ve ilgi çekici film önerileri sunacak bir öneri sistemi geliştirmektir. Proje, veri seti olarak MovieLens 20M veri setini kullanarak, kullanıcıların izlediği filmler ve bu filmler hakkında yaptığı değerlendirmeler üzerinden çıkarımlar yapmayı hedefler. Bu büyük veri seti, kullanıcıların film beğenileri, izleme tercihleri ve yorumlarını içerir ve kişiselleştirilmiş öneriler sunmak için yeterli bilgi sağlar.

Proje kapsamında, gerekli yerlerde **Apriori algoritması** kullanılarak birliktelik kurallarına göre film öneri sistemi oluşturulacaktır. Bu algoritma, özellikle büyük veri kümelerinde sık görülen kalıpları ve ilişki kurallarını keşfetmek için etkili bir yöntemdir. Kullanıcıların izlediği ve beğendiği filmler arasında kurulan bu ilişki kuralları, gelecekte izlemek isteyebilecekleri diğer filmleri tahmin etmemizi sağlar. Apriori algoritması, bu ilişkileri analiz etmek ve güçlü birliktelik kuralları çıkararak öneri sistemini güçlendirmek için önemli bir araçtır.

2.Materyal ve Yöntem

2.1-MovieLens 20M Dataset

Projede kullanılan MovieLens 20M veri seti, film derecelendirmeleri ve filmle ilgili diğer verileri içeren, MovieLens tarafından yayınlanmış büyük bir veri kümesidir. İçerisinde 20 milyondan fazla derecelendirme kaydı vardır.

MovieLens 20M Veri Setinin İçeriği:

1. **movies.csv**:
   - ➢ Filmler hakkında temel bilgileri içerir.
   - ➢ **Alanlar**:
     - movieId: Film ID'si
     - title: Film adı
     - genres: Film türleri
2. **ratings.csv**:
   - ➢ Kullanıcıların filmlere verdikleri puanları içerir.
   - ➢ **Alanlar**:
     - userId: Kullanıcı ID'si
     - movieId: Film ID'si
     - rating: Kullanıcının filme verdiği puan (0.5 - 5 arasında)
     - timestamp: Derecelendirmenin yapıldığı tarih ve saat
3. **tags.csv**:
   - ➢ Kullanıcılar tarafından filmlere eklenen etiketleri içerir.
   - ➢ **Alanlar**:
     - userId: Kullanıcı ID'si
     - movieId: Film ID'si
     - tag: Kullanıcı tarafından filme eklenen etiket
     - timestamp: Etiketin eklendiği tarih ve saat
4. **genome-scores.csv**:
   - ➢ Filmler için, önceden tanımlanmış etiketlerin uygunluk puanlarını içerir.
   - ➢ **Alanlar**:

- movieId: Film ID'si
- tagId: Etiket ID'si
- relevance: Etiketin filme uygunluk puanı (0 - 1 arasında)

5. **genome-tags.csv**:
   - ➢ Önceden tanımlanmış etiketleri ve etiket ID'lerini içerir.
   - ➢ **Alanlar**:
     - tagId: Etiket ID'si
     - tag: Etiketin kendisi

2.2-Apriori Algoritması

Apriori algoritması, büyük veri kümeleri içinde sık görülen öğe gruplarını (itemsets) ve bu gruplar arasındaki ilişkileri bulmak için kullanılan bir **birliktelik kuralı** (association rule learning) algoritmasıdır. Özellikle alışveriş sepeti analizinde yaygın olarak kullanılır. Algoritma, belirli bir eşik değeri (support threshold) üzerinde gerçekleşen öğe kombinasyonlarını belirleyerek güçlü birliktelik kurallarını çıkarır.

Apriori algoritması üç ana aşamadan oluşur:

1. **Sık Öğe Kümelerinin Belirlenmesi**: Algoritmanın ilk aşamasında, veri kümesinde belirli bir destek değerine (support) sahip olan öğe kümeleri belirlenir. Bu destek değeri, öğenin tüm veri kümesindeki görülme sıklığını ifade eder. İlk olarak tekil öğeler analiz edilir ve belirlenen destek değerini geçenler sık öğe olarak işaretlenir. Ardından, bu sık öğeler birleştirilerek daha büyük öğe kümeleri oluşturulur.
2. **Birliktelik Kurallarının Çıkarımı**: Algoritmanın ikinci aşamasında, sık öğe kümelerinden birliktelik kuralları çıkarılır. Birliktelik kuralları, bir öğe kümesinin başka bir öğe kümesi ile ilişkili olma olasılığını belirler. Bu ilişkiler, "güven" (confidence) değeri ile ölçülür. Örneğin, A filmini izleyen kullanıcıların B filmini de izleme olasılığı, A ve B arasındaki bir birliktelik kuralı olarak ifade edilir.
3. **Kuralların Filtrelenmesi**: Son aşamada, belirli bir minimum güven değerini karşılamayan kurallar elenir. Bu sayede yalnızca anlamlı ve güçlü ilişkilere sahip olan kurallar analizde tutulur. Destek ve güven değerleri ile birlikte diğer değerlendirme ölçütleri kullanılarak öneri sisteminin doğruluğu artırılabilir.

2.3-Veri Temizleme ve Filtreleme

Projenin başarılı ve doğru bir biçimde çalışabilmesi için bazı eksik veya eşleşmeyen verilerin temizlenmesi ve düzenlenmesi gerekiyor. Bunun için bazı csv dosyalarının kullanıma göre düzenlenmesi gerekmekteir. Kullanıma göre csv dosyaları birden fazla farklı filtrelemelere göre yeni csv dosyaları oluşturulmaktadır.

Tag.csv:

tag.csv dosyasında userId ve MovieId leri olup tag değerleri null olan toplam 16 adet satır vardır. Bu satırlar belirlenip silinmiştir. Ardından daha doğru bir öneri sistemi için 3 dan az tag değeri vermiş kullanıcılar silinmiştir.

Eski csv dosyaları	Eski Değer	Yeni csv dosyaları	Yeni Değer
Tag.csv	7801	cleaned_tag.csv	5002

Movie.csv

movie.csv dosyasında herhangi bir null değer yok fakat rating.csv dosyası ile karşılaştırıldığında eşleşmeyen 534 adet film var. Yani kullanıcılardan hiçbiri bu filmlere rating puanı vermemiş. Bu yüzden bu 534 adet film **movie.csv** dosyasından silinmiştir.

Eski csv dosyaları	Eski Değe	Yeni csv dosyaları	Yeni Değer
movie.csv	2727	filtered_movies	26745
filtered_movi	2674	top_10_genres_movies.	23341
filtered_movi	2674	MoreThan40Movies.cs	10193

Rating.csv

rating.csv dosyasında herhangi bir null değer bulunmamaktadır. Bu csv yi filtrelemek için 300 filmden az filme rating puanı veren kullanıcılar silindi. Popüler-Türe göre kısmında bu csv kullanıldı diğer kısımlarda filtrelenmemiş kullanıldı. Sebebi popüler olanlarda fazla film izleyenleri seçmek, veriyi küçülterek daha hızlı çalışmasını sağlamak ve bellek tasarrufu yapmak.

Eski csv dosyaları	Eski Değer	Yeni csv dosyaları	Yeni Değer
rating.csv	13849	filtered_rating1.csv	26745
rating.csv	13849	filtered_rating2.csv	

2.4-Yöntem

Bu bölümde projede seçeneklere göre film öneri sistemlerinin hangi yöntemle yapıldığı anlatılmaktadır. Bu yöntemler [ek-1] deki proje kılavuzundaki “Kişiselleştirilmiş Öneriler” ve “Birliktelik Kurallarına Dayalı Öneriler” bölümleri dikkate alınarak yapılmıştır.

2.4.1-Kişisel Beğenilere Göre Öneriler

Kişi-Tür: Bu bölümde Seçilen kullanıcıya göre, seçilen türde kişiselleştirilmiş film önerisinde bulunulmaktadır. Bunun için **ClassKisiTür.py** dosyasından **KisiTür** sınıfı kullanılır. Sınıf 2 adet csv dosyası ve 3 adet fonksiyondan oluşmaktadır.

Kullanılan csv dosyaları:
top_10_genres_movies.csv , filtered_ratings1.csv

Fonksiyonlar: get_movies_by_genre, get_high_rated_movies_by_user, get_recommendations.

get_movies_by_genre: Seçilen türdeki filmleri filtreler.

get_high_rated_movies_by_user: Seçilen kullanıcının seçilen türdeki izlediği filmlerden yüksek rating puanı (3.5 ve üzeri) verdiği filmleri seçer.
get_recommendations: Kullanıcının yüksek puan (3.5 ve üzeri) verdiği filmleri izleyen diğer kullanıcılara bakar. Bu kullanıcıların seçilen türde izlediği ve yüksek rating puanı(3.5 ve üzeri) verdiği filmleri seçer ve öneri yapar.

Kişi İsim: Bu bölümde seçilen kullanıcıya göre, seçilen film isminde kişiselleştirilmiş film önerisinde bulunulmaktadır. Bunun için **ClassKisiİsim.py** sınıfındaki **KisiİsIm** sınıfı kullanılmaktadır.

Kullanılan csv dosyaları:
filtered_movies.csv , cleaned_tag.csv, genome_scores.csv, genome_tags.csv

Fonksiyonlar:
get_movie_recommendation_by_user_tag: Bu fonksiyon seçilen kullanıcının izlemiş olduğu filmlere bakar. Sonrasında seçilen filme yapmış olduğu tag etiketini cleaned_tag.csv dosyasından alır. Daha sonra bu tagin Id sini genome_tags.csv dosyasından alır. En son genome_scores.csv dosyasından bu tag etiketiyle relevance değeri yüksek olan(0.6 ve üzeri) filmleri önerir.

2.4.2-Birliktelik Kurallarına Dayalı Öneriler

Popüler-Film İsmi: Bu bölümde seçilen filme göre birliktelik kuralları kullanılarak film önerisinde bulunulmaktadır. Bunun için **ClassPopİsim.py** dosyasındaki **PopIsim** sınıfı kullanılmaktadır.

Kullanılan csv dosyaları:
TagMovie.csv , cleaned_tag.csv, genome_scores.csv.

Fonksiyonlar: get_top_tags, get_similar_movies, generate_user_movie_matrix, get_recommendations,

get_top_tags: Bu fonksiyon seçilen film ile ilgili yapılmış taglerden relevance değeri en yüksek 15 tag etiketini alır.

get_similar_movies: Bu fonksiyon elde edilen tag değerleri ile relevance değeri yüksek olan (0.7 ve daha üstü) filmleri seçer.

generate_user_movie_matrix: Bu metod, önerilecek filmleri değerlendirmek için gerekli olan kullanıcı-film matrisini oluşturur. **cleaned_tag.csv** ve **TagMovie.csv** dosyalarını kullanarak apriori algoritmasında kullanmak için pivot tablo oluşturur. Oluştururken izlenen filmler için 1, izlenmeyenler için 0 yazar.

get_recommendations:

- get_similar_movies metodunu çağırarak, kullanıcıya önerilecek benzer film ID'lerini (similar_movie_ids) alır.

- generate_user_movie_matrix metodunu çağırarak, benzer filmler ve kullanıcılar arasındaki ilişkiyi gösteren bir kullanıcı-film matrisi (user_movie_pivot) elde eder.

Apriori ve İlişkisel Kurallar:

- **Apriori Algoritması**: Kullanıcı-film matrisine Apriori algoritmasını uygulayarak, sıkça bir arada izlenen film kümelerini (frequent_itemsets) bulur.
- **İlişkisel Kurallar**: Apriori algoritmasından elde edilen küme verileri üzerinden, güven (confidence) değeri 0.5'in üzerinde olan ilişki kurallarını (rules) hesaplar. Bu kurallar, izlenen filmler arasında ilişki olup olmadığını anlamak için kullanılır.

Öneri Listesi:

- similar_movie_ids kümesinden önerilecek filmleri bulmak için, rules'ta antecedents (öncüller) kısmında benzer filmlerden en az birinin geçtiği kuralları filtreler.
- consequents (sonuçlar) kısmından, recommended_movie_ids kümesine önerilecek film ID'lerini ekler.

- Son olarak, recommended_movie_ids kümesindeki film ID'leri üzerinden movies_df'den seçme yaparak öneri listesini oluşturur ve kullanıcıya film başlığı (title) ve türleriyle (genres) birlikte öneri listesi olarak döndürür.

Popüler Tür: Bu bölümde seçilen film türüne göre film önerisi yapmaktadır. Bunun için **ClassPopülerTür.py** dosyasındaki **PopulerGenre** sınıfı kullanılmaktadır.

Kullanılan csv dosyaları: filtered_ratings2.csv, MoreThan50Movies.csv

Fonksiyonlar: get_popular_genre_movies.csv, get_users_who_watched_genre_movies, create_user_movie_pivot, generate_recommendations, get_recommended_movies

get_populer_genre_movies:

- **Tür Filtreleme**: selected_genre ile belirtilen türe sahip filmleri movies_df'de arar ve genre_movies DataFrame'inde saklar.

• **Minimum Ortalama Puan Filtreleme**: ratings_df'de her film için ortalama puan hesaplar ve bu puan, min_rating değerinden yüksek olan filmleri seçer.

    • popular_genre_movies DataFrame'i, hem belirli bir türe sahip hem de minimum puanın üzerinde olan filmleri içerir.

get_users_who_watched_genre_movies:

Bu metod, popular_genre_movies'deki popüler türe ait filmleri izleyen kullanıcıların listesini çıkarır.

- ratings_df'de, popüler türe ait filmler için puan veren kullanıcıları seçer.
- Bu kullanıcıların benzersiz ID'lerini (userId) döndürür.

create_user_movie_pivot:

Bu metot, popüler türe ait filmleri izleyen kullanıcılar için bir kullanıcı-film pivot tablosu oluşturur.

- ratings_df üzerinden, ilgili kullanıcıların popüler türe ait filmleri izleyip izlemediklerini gösteren bir pivot tablo (user_movie_pivot) oluşturur.
- Bu tabloyu, film izlenmişse True, izlenmemişse False olarak gösteren bir boolean veri tipine dönüştürür.
- Pivot tablo, kullanıcıların popüler türe ait filmleri izleme durumunu gösterir ve Apriori algoritması için hazırlanır.

generate_recommendations:

Apriori ve İlişkilendirme Kuralları:

- user_movie_pivot tablosunda Apriori algoritmasını çalıştırarak, sıkça birlikte izlenen film kümelerini (frequent_itemsets) bulur.
- Bu kümelerden, güven (confidence) değeri belirtilen min_confidence (varsayılan olarak 0.5) eşik değerinden yüksek olan ilişkilendirme kurallarını (rules) elde eder.

Önerilecek Filmler:

- Belirli bir güven değerinin üzerindeki kuralların consequents (sonuçlar) kısmında yer alan film ID'lerini recommended_movie_ids kümesine ekler.
- Bu küme, önerilmesi muhtemel olan filmlerin ID'lerini içerir ve öneri listesi oluşturmak için kullanılır.

get_recommended_movies:

recommended_movie_ids kümesindeki film ID'leri üzerinden movies_df'den seçme yaparak önerilen filmleri alır.

- Film ID, başlık (title), ve tür bilgileriyle (genres) birlikte önerilen filmleri recommended_movies DataFrame'i olarak döndürür.

3-Değerlendirme Ölçütleri

Bu projede **Apriori algoritması** ve değerlendirme ölçütlerinden **support (destek)** ve **confidence (güven)**, kullanıcıların beğendiği veya izlediği filmlerden yola çıkarak diğer kullanıcılar için film önerileri oluşturmak amacıyla kullanılıyor. İşte bu iki ölçütün projede nasıl kullanıldığına dair detaylar:

1. Support (Destek Değeri)

Projede, **support** değeri belirli bir film kümesinin kullanıcılar arasında birlikte izlenme sıklığını ifade eder. Bu projede, generate_recommendations fonksiyonunda **min_support** parametresi aracılığıyla belirlenmiştir. Support değeri, kullanıcıların birlikte izlediği film gruplarının ne kadar popüler olduğunu ölçer.

- **Popüler - Türe Göre Kullanımı:**
  PopulerGenre sınıfında, generate_recommendations fonksiyonu apriori algoritmasını kullanarak sık izlenen film kombinasyonlarını belirler. min_support=0.4 gibi bir eşik değeri ayarlanarak, yalnızca bu orandan fazla izlenmiş film kombinasyonları analiz edilir. Bu sayede, düşük izlenme oranına sahip kombinasyonlar elenir ve yalnızca kullanıcılar arasında yaygın olarak izlenen film grupları öneri sürecine dahil edilir.
- **Popüler - İsme Göre Kullanımı:**
  PopIsim sınıfında, get_recommendations fonksiyonu ile min_support=0.01 eşik değeri kullanılarak apriori algoritması uygulanır. Burada, user_movie_pivot tablosunda benzer filmleri izleyen kullanıcıların ortak film tercihlerinin yaygınlığı analiz edilir. Bu düşük destek değeri, nadir ancak anlamlı izleme örüntülerini de tespit etmeye yardımcı olur.

2. Confidence (Güven)

Projede **confidence** değeri, bir film grubunun izlenmiş olması durumunda diğer bir filmin de izlenmiş olma olasılığını ölçer. Bu, önerilen filmin ne kadar anlamlı olduğunu gösterir. Yani, birlikte izlenen filmler arasındaki ilişkilerin gücünü ifade eder.

- **Popüler - Türe Göre Kullanımı:**

PopulerGenre sınıfında, apriori algoritmasıyla oluşturulan sık film kümelerinden association_rules fonksiyonu ile öneri kuralları üretilir. min_confidence=0.5 gibi bir güven eşiği belirlenir ve yalnızca bu eşiği geçen kurallar analiz edilir. Bu sayede, yalnızca anlamlı ve güçlü ilişkiler önerilere dahil edilir ve kullanıcının belirli bir türdeki tercihleri analiz edilerek öneri yapılır.

- **Popüler - İsme Göre Kullanımı:**
  PopIsim sınıfında, generate_user_movie_matrix fonksiyonu ile kullanıcı ve film ilişkileri bir pivot tabloya dönüştürülür. Bu tablo üzerinde apriori algoritması çalıştırılarak min_confidence=0.5 eşiğine göre güvenilir kurallar belirlenir. İlgili kurallar filtrelenerek, relevant_rules içerisindeki güçlü ilişkilere göre öneriler çıkarılır. Bu, aynı türden izlenme örüntülerine dayalı olarak anlamlı film önerileri oluşturur.

Bu proje sayesinde, kullanıcıların geçmiş izleme tercihleri analiz edilerek en çok izlenen ve güvenilir ilişkiler taşıyan filmler öneri olarak sunulur. **Support** değeri, birlikte izlenen film gruplarının seçimini sağlarken, **confidence** değeri ise önerinin güvenilirliğini sağlayarak anlamlı öneriler sunar.

4-Sonuç

Projede her bir öneri şekli için 4 adet Class yapısı kullanılmıştır. Bu classlardan arayüz dosyasında nesne oluşturarak bu classlar ve içindeki fonksiyonlar öneri sistemi için kullanılmıştır. Veri temizleme dosyasında gerekli csv düzenlemelri yapılmış ve öneri siteminde kriterlere göre farklı csv dosyaları kullanılmıştır.

Arayüz [1] deki “ 3.5. Öneri Sistemi Tasarımı” bölümündeki “Arayüz (GUI) Tasarımı” kısmında verilen bilgilere göre yapılmıştır. **Apriori Algoritması’nın** uygulanması bu bölümdeki:

- **“Popüler Film Önerileri” yazmalı ve bu seçenek seçildiğinde birliktelik kurallarına dayalı öneriler yöntemini kullanarak sistemin önerilerde bulunulması istenmektedir. Birinci alandaki ikinci seçenekte “Kişiselleştirilmiş Film Önerileri” yazmalı ve bu seçenek seçildiğinde kişiselleştirilmiş öneriler yöntemini kullanarak sistemin önerilerde bulunulması istenmektedir.**

- kısmına göre yapılmıştır.

5-Kaynakça

[1] YAZILIM GELİŞTİRME I DERSİ PROJE I Bilgi Metni

[2] https://www.kaggle.com/datasets/grouplens/movielens-20m-dataset

[3]https://www.youtube.com/watch?v=034t73eSfOc&t=151s&pp=ygUcYW5sYcWfxLFsxLFyIGVrb25pbWkgYXByaW9yaQ%3D%3D

[4]https://www.youtube.com/watch?v=zDaIGpHNCDk&t=307s&pp=ygUXYW5sYcWfxLFsxLFyIGVrb25pbWkgOTY%3D

[5] https://medium.com/@ekrem.hatipoglu/machine-learning-association-rule-mining-birliktelik-kural-%C3%A7%C4%B1kar%C4%B1m%C4%B1-apriori-algorithm-4326b8f224c3