BÜYÜK VERİ PROJESİ

Ahmet YALÇIN
Bilgisayar Mühendisliği
TOBB Ekonomi ve Teknoloji Üniversitesi
Ankara, Türkiye
ahmetyalcin@etu.edu.tr

Efe Barkın Köse
Bilgisayar Mühendisliği
TOBB Ekonomi ve Teknoloji Üniversitesi
Ankara, Türkiye
efebarkinnn@gmail.com

***Özetçe*—Bu çalışmada, kullanıcılara kişiselleştirilmiş film önerileri sunmak için işbirlikçi filtreleme ve içerik tabanlı filtreleme tekniklerini birleştiren hibrit bir film öneri sistemi geliştirilmiştir. Sistem, Apache Spark üzerinde çalışan ALS (Alternating Least Squares) algoritması ve TF-IDF vektörleştirme tekniklerini kullanarak, hem kullanıcı davranışlarını hem de film içeriklerini analiz etmektedir. MovieLens 25M veri seti üzerinde gerçekleştirilen deneyler, hibrit yaklaşımın tek başına işbirlikçi veya içerik tabanlı filtrelemeye göre daha iyi sonuçlar verdiğini göstermiştir.**

***Anahtar Kelimeler*—Büyük Veri, Apache Spark, ALS**

I. GİRİŞ

Projenin Github linkine gitmek için buraya tıklayın.

Film öneri sistemleri, kullanıcıların tercihlerini analiz ederek onlara ilgi duyabilecekleri yeni içerikler sunmayı amaçlar. Günümüzde Netflix, Amazon Prime ve Disney+ gibi platformların başarısında, etkili öneri sistemlerinin rolü büyüktür. Bu sistemler, kullanıcı memnuniyetini artırırken, platformların içerik keşfedilebilirliğini de geliştirmektedir.

Bu projede, kullanıcılara daha doğru ve çeşitli öneriler sunmak için işbirlikçi filtreleme ve içerik tabanlı filtreleme yaklaşımlarını birleştiren hibrit bir öneri sistemi geliştirilmiştir. İşbirlikçi filtreleme, benzer kullanıcıların davranışlarını analiz ederken, içerik tabanlı filtreleme film özellikleri ve metadatalarını kullanır. Bu iki yaklaşımın birleştirilmesi, "soğuk başlangıç" problemi gibi her iki yaklaşımın da zayıf yönlerini telafi etmeyi amaçlamaktadır.

II. VERİ

Projede kullanılan Veri seti için buraya tıklayın.

A. Veri Kaynakları ve Yapısı

1) MovieLens Veri Seti:

- Bu projede, GroupLens Research tarafından sağlanan MovieLens 25M veri seti kullanılmıştır. Bu veri seti, 162,000 kullanıcının 62,000 film hakkında 25 milyon derecelendirmesini içermektedir. Veri seti ratings.csv, movies.csv, tags.csv, genome-scores.csv ve genome-tags.csv dosyalarından oluşmaktadır.

2) Veri Boyutları:

- Ratings.csv dosyası 25 milyon satır ve 650 MB boyutundadır. Movies.csv dosyası 62,423 film kaydı içermektedir. Tags.csv dosyası 1 milyondan fazla kullanıcı etiketi içerir. Genome-scores.csv dosyası 11.7 milyon film-etiket ilişki skoru içerir. Genome-tags.csv dosyası 1,128 benzersiz etiket içerir.

B. Veri Ön İşleme Adımları

1) Veri Doğrulama ve Temizleme:

- Veri yükleme sürecinde, her veri dosyası için sütun varlığı kontrolü, veri tipi doğrulaması ve tekrarlanan kayıtların temizlenmesi gibi adımlar uygulanmıştır. Özellikle ratings.csv dosyasında userId ve movieId alanları tamsayıya, rating alanı ise ondalık sayıya dönüştürülmüştür.

2) Eksik Değer İşleme:

- Veri setindeki eksik değerler, veri türüne göre farklı stratejilerle ele alınmıştır. Sayısal sütunlardaki eksik değerler ortalama değer ile, kategorik sütunlardaki eksik değerler ise en sık görülen değer ile doldurulmuştur.

3) Özellik Mühendisliği:

- Film türleri (genres) sütunu, her film için birden fazla tür içeren bir dize olarak saklanmaktadır. Bu veri, türler dizisi olarak ayrıştırılmış ve one-hot encoding uygulanmıştır. Derecelendirme zaman damgaları (timestamp), Unix zaman damgası formatından okunabilir tarih formatına dönüştürülmüştür. Film başlıklarından yıl bilgisi çıkarılarak ayrı bir sütun oluşturulmuştur. Ayrıca başlıklar, yıl bilgisi olmadan temizlenmiş bir formatta saklanmıştır.

C. Veri Keşfi ve Analizi

1) Derecelendirme Dağılımı:

- Kullanıcı derecelendirmelerinin dağılımı analiz edilmiştir. Derecelendirmelerin 0.5 ile 5.0 arasında değiştiği ve en yaygın derecelendirmenin 4.0 olduğu gözlemlenmiştir.

2) Film Türü Analizi:

- Film türlerinin dağılımı ve popülerliği analiz edilmiştir. Drama, Komedi ve Aksiyon en popüler türler olarak öne çıkmıştır. Ayrıca, her türün ortalama derecelendirmesi hesaplanarak, hangi türlerin daha yüksek puan aldığı belirlenmiştir.

3) Kullanıcı Aktivite Analizi:

- Kullanıcıların derecelendirme davranışları analiz edilmiştir. Kullanıcı başına derecelendirme sayısı ve ortalama derecelendirme hesaplanmıştır. Bazı kullanıcıların binlerce film derecelendirdiği, bazılarının ise sadece birkaç film derecelendirdiği gözlemlenmiştir.

4) Zaman Serisi Analizi:

- Derecelendirmelerin zaman içindeki dağılımı analiz edilmiştir. Yıllara ve aylara göre derecelendirme sayıları hesaplanarak, kullanıcı aktivitesinin zamansal değişimi incelenmiştir.

D. Veri Örnekleme ve Bölme

1) Eğitim ve Test Veri Setleri:

- Modellerin eğitimi ve değerlendirilmesi için veri seti, eğitim (%80) ve test (%20) olarak bölünmüştür. Bu bölünme, modellerin genelleme yeteneğini değerlendirmek için kullanılmıştır.

2) Veri Örnekleme:

- Büyük veri setiyle çalışırken, hesaplama kaynaklarını verimli kullanmak için veri örnekleme uygulanmıştır. Orijinal veri setinin %20'si alınarak, daha hızlı model eğitimi ve değerlendirmesi sağlanmıştır.

E. Veri Kalitesi ve Güvenlik Önlemleri

1) Veri Doğrulama Mekanizmaları:

- Veri yükleme sürecinde, dosya varlığı ve erişilebilirliği kontrol edilmiştir. Eksik veya bozuk dosyalar için uygun hata mesajları oluşturulmuştur.

2) Hata Yönetimi:

- Veri işleme sürecinde oluşabilecek hatalar için kapsamlı hata yönetimi uygulanmıştır. Hata durumunda, uygun günlük (log) mesajları oluşturulmuş ve alternatif çözümler uygulanmıştır.

III. KULLANILAN TEKNOLOJİLER

A. Apache Spark / PySpark

Büyük veri setlerinin paralel ve dağıtık olarak işlenmesi için kullanılmıştır. Özellikle Alternating Least Squares (ALS) algoritmasının ölçeklenebilir şekilde eğitilmesi ve veri ön işleme adımlarında Spark DataFrame yapısından yararlanılmıştır.

B. Flask

Projeye web tabanlı bir kullanıcı arayüzü ve RESTful API endpointleri kazandırmak için kullanılmıştır. Kullanıcıların öneri sistemine erişimi ve sonuçların görselleştirilmesi Flask ile sağlanmıştır.

C. scikit-learn

İçerik tabanlı öneri algoritmalarında, metin verisinin işlenmesi ve TF-IDF (Term Frequency-Inverse Document Frequency) vektörleştirme işlemleri için kullanılmıştır. Ayrıca, cosine similarity fonksiyonu ile benzerlik hesaplamaları yapılmıştır.

IV. KULANILAN MODELLER

A. Alternating Least Squares (ALS)

ALS (Alternating Least Squares), özellikle büyük ölçekli ve seyrek kullanıcı-ürün etkileşim matrislerinde öneri yapmak için yaygın olarak kullanılan bir işbirlikçi filtreleme algoritmasıdır. Bu projede PySpark MLlib'in ALS implementasyonu kullanılmıştır.

ALS algoritması, kullanıcıların geçmişteki film puanlamalarını temel alarak, kullanıcı ve film vektörlerini öğrenir. Amaç, kullanıcıların puanlama eğilimlerini ve filmler arasındaki ilişkileri düşük boyutlu bir uzayda temsil etmektir. Model, kullanıcı-film matrisini iki düşük boyutlu matrisin çarpımı olarak yaklaşıklar ve bilinmeyen puanlamaları tahmin eder.

- Veri Girdisi: Model, kullanıcı kimliği (userId), film kimliği (movieId) ve puan (rating) sütunlarını içeren bir veri seti ile eğitilmiştir.
- Model Eğitimi: ModelTrainer sınıfı, hiperparametre optimizasyonu için grid search ve cross-validation (çapraz doğrulama) yöntemlerini kullanır. Parametreler arasında latent faktör sayısı, regularization katsayısı ve iterasyon sayısı bulunur.
- Soğuk Başlangıç Problemi: Modelin tahminlerinde güvenilirlik sağlamak için coldStartStrategy olarak "drop" seçeneği kullanılmıştır. Böylece modeli hiç görmemiş kullanıcı/film çiftleri değerlendirme sırasında dışlanır.
- Değerlendirme Metrikleri: Modelin başarısı RMSE (Root Mean Squared Error), MAE (Mean Absolute Error) ve R2 (R-kare) gibi metriklerle ölçülür.
- Tahmin ve Öneri: Eğitim tamamlandıktan sonra, belirli bir kullanıcı için en yüksek tahmin puanına sahip filmler öneri olarak sunulur.

B. İçerik Tabanlı Filtreleme Modeli

Filmlerin içerik özelliklerine (başlık, tür, etiket, vb.) dayalı öneriler sunan bu modelde, TF-IDF vektörleştirici ve cosine similarity metriği kullanılmıştır. Kullanıcının daha önce beğendiği filmlerle içerik açısından benzer olan filmler, metin madenciliği teknikleriyle belirlenerek öneri olarak sunulmuştur.

V. SONUÇ

Bu projede, film öneri sistemleri alanında yaygın olarak kullanılan iki farklı yaklaşım — işbirlikçi filtreleme (ALS) ve içerik tabanlı filtreleme — bir arada uygulanarak kapsamlı bir öneri platformu geliştirilmiştir. Büyük veri işleme için Apache Spark altyapısı ve kullanıcı dostu bir web arayüzü için Flask framework'ü tercih edilmiştir. Proje kapsamında, hem

kullanıcıların geçmiş etkileşimlerine dayalı kişiselleştirilmiş öneriler sunulmuş, hem de içerik benzerliklerine göre alternatif öneriler sağlanmıştır.

Sonuç olarak, geliştirilen bu öneri sistemi; veri bilimi, makine öğrenimi ve yazılım mühendisliğinin bir araya getirildiği, gerçek dünyada uygulanabilir ve kullanıcı deneyimini ön planda tutan çözüm sistemi tasarlanmıştır.

KAYNAKLAR

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

[2] https://spark.apache.org/