

# Film Öneri Sistemi

Esra DİNÇ

Kocaeli Üniversitesi

Kocaeli, Türkiye

235112013@koceli.edu.tr

**Öz—** Bu projede, veri analizi ve model eğitimi için PySpark kullanılarak bir film öneri sistemi geliştirildi. İlk olarak, kullanıcıların film tercihlerini içeren veri seti PySpark'ın güçlü yetenekleriyle işlenerek veri analizi gerçekleştirildi. Veri ön işleme aşamalarında, eksik, istenmeyen türde olan ve istenmeyen karakterler içeren veriler ele alınarak temiz bir veri seti oluşturuldu. Kullanılacak olan veri seti eğitim ve test seti olarak ikiye ayrıldı. Daha sonra, ALS (Alternating Least Squares) algoritmasıyla model eğitildi. Bu süreçte, çapraz doğrulama ile modelin güvenilirliğini değerlendirildi ve ardından test seti üzerinde modelin başarısı ölçüldü. Son olarak, eğitilen model kullanılarak kullanıcılara kişiselleştirilmiş film önerilerinde bulunularak proje tamamlandı.

**Anahtar kelimeler:** Film önerisi, PySpark, Python, ALS (Alternating Least Squares), Çapraz doğrulama

## I. GİRİŞ

Film öneri sistemleri, bireylerin geniş bir film ve dizi içeriği arasından seçim yapmalarına yardımcı olmak amacıyla kullanılır. Kişiselleştirilmiş içerik deneyimi sunarak, kullanıcılara izleme geçmişleri, tercihleri ve beğenileri üzerinden özel öneriler sunarlar. Bu sistemler, zaman tasarrufu sağlar ve kullanıcıları popüler içeriklerle sınırlamaz; aksine, benzer profillere sahip kullanıcıların tercihleri ve çeşitli kategorilerdeki önerilerle kişisel keşif fırsatları sunarlar. Film öneri sistemleri, film ve dizi platformlarından online alışveriş sitelerine, sosyal medya platformlarından müzik ve kitap platformlarına kadar birçok alanda kullanılarak geniş bir içerik yelpazesi sunarak çeşitliliği artırır.

Bu proje, PySpark makine öğrenimi tekniklerini kullanarak kullanıcılara daha önceden izledikleri ve oy verdikleri filmlerden ve bu oylardan yola çıkarak daha önceden izlemediği fakat ilgi alanı olan ve izleyebileceği film önerileri sunar.

PySpark<sup>[1]</sup>, Büyük veri kapsamında Spark ile çalışabilmeyi sağlayan bir kütüphanedir. Büyük veri ekosisteminin, özellikle Spark tarafına bakan yönleriyle, SQL işlemleri, makine öğrenmesi uygulamaları, veri ön işleme işlemleri, modelleme gibi çok çeşitli ihtiyaçlar kapsamında çözüm sunar. Bu projede PySpark'ın kullanılması, büyük ölçekli veri kümeleri üzerinde etkili ve hızlı bir şekilde çalışabilen bir platforma ihtiyaç duyulmasından kaynaklanmaktadır.

Büyük veri setleri, geleneksel veri işleme araçlarıyla işlenmekte zorluklar yaşayabilir ve işlem süreleri uzayabilir. Ancak, PySpark'ın dağıtılmış işleme yetenekleri sayesinde, büyük veri setleri paralel olarak işlenir ve işlem süreleri önemli ölçüde azaltılır. Bu özellik, film öneri sistemimizin daha geniş ve karmaşık veri setlerinde dahi hızlı ve ölçeklenebilir çözümler sunabilmesine olanak tanır.

Öte yandan, ALS (Alternatif En Küçük Kareler) algoritması gibi makine öğrenimi algoritmalarının

uygulanması için PySpark, paralel hesaplama yetenekleriyle bu algoritmaların büyük veri setlerinde etkili bir şekilde çalışmasını sağlar. Bu da film öneri sisteminin daha kesin ve kişiselleştirilmiş öneriler sunmasına olanak tanır.

Alternating Least Squares(ALS)<sup>[2]</sup>, her bir yineleme için orijinal verilerimizin çarpanlara ayrılmış temsiline daha yakın ve daha yakın gelmeye çalıştığımız yinelemeli bir optimizasyon(iterative optimization process) sürecidir. Büyük ölçekli veri setlerindeki eksik değerleri doldurmak ve kullanıcıların ürünlere olan tercihlerini tahmin etmek amacıyla kullanılan bir matris faktörizasyon algoritmasıdır. ALS algoritması, kullanıcı ve ürün matrislerini birbirine yakın bir şekilde eşleştirmek için iteratif bir optimizasyon süreci uygular. Bu projede ALS algoritması, PySpark'ın içinde yer alan pyspark.ml.recommendation. ALS sınıfı kullanılarak implemente edildi. Ayrıca, modelin performansını değerlendirmek için çapraz doğrulama (cross-validation) tekniği kullanıldı. Çapraz doğrulama, modelin genelleştirilebilirliğini ve performansını test etmek için veriyi farklı alt kümelerde eğitip değerlendirmeyi sağlar. Bu yaklaşım, projede kullanılan ALS modelinin güvenilirliğini ve etkinliğini artırmaya yönelik bir strateji olarak tercih edildi.

## II. YÖNTEM

### A. Geliştirme Ortamı

Bu çalışma, Python dilini ve PySpark'ın 3.5 versiyonunu içeren bir geliştirme ortamı olarak Visual Studio Code'u (Vs Code) kullanarak film öneri sistemi geliştirmiştir. PySpark, büyük film veri setleri üzerinde etkili bir performans sergileyebilen bir araçtır. Dağıtık hesaplama yetenekleri, paralel işleme avantajları ve Python dilinin kullanım kolaylığı, projenin geliştirilmesine hız ve ölçeklenebilirlik katmıştır. Bu kombinasyon, film öneri sistemi gibi karmaşık görevlerde daha etkili çözümler üretebilmemize olanak tanımaktadır.

### B. Veri Seti

Veri seti olarak "Movierating"<sup>[3]</sup> kullanılmıştır. Veri seti, kullanıcıları, kullanıcıların izledikleri filmleri ve bu filmlere verdikleri oyları içermektedir. İçerisinde yaklaşık 100.000 örnek bulundurmaktadır. Verilen oylar, 1 ile 5 arasında değer alır, bu da kullanıcıların filmlere verdikleri derecelendirmelerin hassasiyetini gösterir. Kullanıcı benzersiz anahtarları, her bir kullanıcıya farklı olacak biçimde tanımlayarak kişiselleştirilmiş öneri sistemlerinin geliştirilmesine olanak tanır. Bu zengin veri seti, büyük veri analizi, öneri sistemleri ve kişiselleştirilmiş içerik önerileri gibi uygulamalarda kullanılarak kullanıcıların tercih ve beğenilerini anlamak ve öngörmek için değerli bir kaynak sunar. Bu veri seti, film endüstrisinde ve kullanıcı deneyimi odaklı platformlarda kullanılacak çeşitli analiz ve modelleme çalışmalarına ilham verici veriler sağlamaktadır.

Kullanılan veri seti içerisinde bir örnek Şekil 1'deki gibidir.

```
405,"Death in the Garden (Mort en ce jardin, La) (1956)",1
782,Ripe (1996),2
592,Ed's Next Move (1996),4
181,Ed's Next Move (1996),1
655,Ed's Next Move (1996),3
655,Two Friends (1986),3
787,Men of Means (1998),3
206,Men of Means (1998),1
655,"Niagara, Niagara (1997)",2
932,Spirits of the Dead (Tre passi nel delirio) (1968),4
405,Spirits of the Dead (Tre passi nel delirio) (1968),1
405,"Glass Shield, The (1994)",1
456,"Glass Shield, The (1994)",3
```

Şekil 1. Veri setinden örnekler

SparkSession oluşturuldu ve veri Dataframe'e atıldı. Yapılan bu işlemler Şekil 2'de gösterildiği gibidir.

```
spark = SparkSession.builder.appName('recommender_system').getOrCreate()
df=spark.read.csv('movie_ratings_df.csv',inferSchema=True,header=True)
```

Şekil 2. SparkSession oluşturma

### C. Veri Analizi

Veri setinin içerdiği satır ve sütun sayısı ve içerdiği verilerin türleri incelenerek veri analiz edildi. Analiz kodları ve çıktılar Şekil 3 ve Şekil 4'te verilmiştir.

```
print("Dataset {d} satır ve {s} sütundan oluşmaktadır.".format(df.count(),len(df.columns)))
print("Dataframe Schema:")
df.printSchema()
```

Şekil 3. Veri analizi kod parçası

```
Dataset 100010 satır ve 3 sütundan oluşmaktadır.
Dataframe Schema:
root
 |-- userId: integer (nullable = true)
 |-- title: string (nullable = true)
 |-- rating: integer (nullable = true)
```

Şekil 4. Veri analizi çıktısı

### D. Veri Ön İşleme

Büyük veri, genellikle hacmi, çeşitliliği ve hızı yüksek olan veri setlerini ifade eder. Büyük veri analitiği, bu veri setlerinden anlamlı bilgiler çıkarmak ve öngörülerde bulunmak için kullanılır. Ancak, bu büyük veri kütlelerini etkili bir şekilde kullanabilmek için öncelikle veri ön işleme adımının tamamlanması gerekmektedir.

Veri ön işleme<sup>[4]</sup>, büyük veri setlerindeki karmaşıklığı azaltmayı, veri setini temizlemeyi ve anlamlı bir şekilde kullanılabilir hale getirmeyi amaçlar.

- 1- Analitik Performansın Artırılması: Veri ön işleme, büyük veri setlerindeki karmaşıklığı azaltır ve analitik modellerin daha hızlı çalışmasını sağlar. Bu da analiz süreçlerinin daha verimli ve etkili olmasını sağlar.
- 2- Doğruluk ve Güvenilirlik: Temizlenmiş ve düzenlenmiş bir veri seti, analitik sonuçların doğruluğunu artırır. Bu da karar alma süreçlerinde daha güvenilir sonuçlar elde edilmesine katkı sağlar.

- 3- Ölçeklenebilirlik: Büyük veri setleri genellikle ölçeklenebilir özelliklere sahiptir. Veri ön işleme, bu ölçeklenebilirlikle başa çıkabilmek için önemlidir ve analizlerin büyüklükleri arttıkça performansın korunmasına yardımcı olur.

Projede yapılan ön işleme adımları şu şekildedir:

- 1- Veri setinde null değer içeren satırlar veri setinden çıkartıldı. Şekil 5'te bahsi geçen işlemler yapılmaktadır.

```
df = df.dropna(subset=['userId', 'rating', 'title'])
```

Şekil 5. Eksik değerleri veri setinden çıkaran kod parçası

Fonksiyonda geçen anahtar kelimelerin açıklamaları aşağıdaki gibidir:

- **Subset:** Herhangi bir sütunda eksik değer kontrolü yapar.Eksik değerler, PySpark DataFrame'deki NaN (Not a Number) veya NULL değerleridir.
  - **Dropna:** Bu fonksiyon eksik değerleri içeren satırları kaldırmak için kullanılır.
- 2- Veri seti içerisinde tekrar eden satırlar veri setinden çıkartıldı. Bahsi geçen işlemi yapan kod parçası Şekil 6'daki gibidir.

```
duplicate_count= df.count() - df.dropDuplicates(['userId', 'title', 'rating']).count()
```

Şekil 6. Tekrar eden satırları silen kod parçası

- 3- Opsiyonel olarak verinin daha düzgün ve temiz gözükmesi için tüm film isimlerinin ilk harflerinin büyük harfle başlanması sağlandı. Şekil 7'de işlemi yapan kod parçası bulunmaktadır.

```
title_initcap_df = df.withColumn("title", initcap(col("title")))
```

Şekil 7. Büyük harfe çevirme işlemi yapan kod parçası

- 4- Film isimlerinin içerisindeki gereksiz karakterler ve noktalama işaretleri kaldırıldı. Şekil 8'de işlemi yapan kod parçası bulunmaktadır.

```
regex_new_df = trimmed_new_df.withColumn("title", regexp_replace(col("title"), "[^a-zA-Z0-9\\s]", ""))
```

Şekil 8. Gereksiz karakterleri silen kod parçası

- 5- Verilen oylar incelenerek 5'ten büyük olan oy değerleri 5'e eşitlendi. Bu işlem Şekil 9'daki kod ile yapılmıştır.

```
cleaned_df = regex_new_df.withColumn("rating", when(col("rating") > 5, 5).otherwise(col("rating")))
```

Şekil 9. Oyları normalize etme işlemi yapan kod parçası

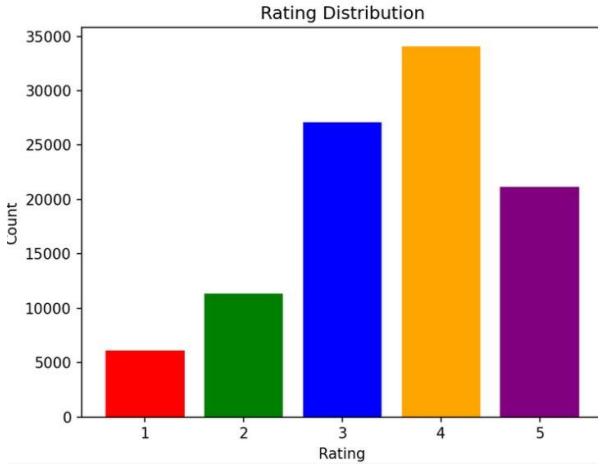
- 6- Film isimlerini içeren sütunda kategorik değerlerin sayısal değerlere indexlenmesi işlemi yapıldı. Elde edilen sonuçlar "title\_new" sütunu oluşturularak bu sütuna yazıldı. Bu işlemi yapan kod parçası Şekil 10'daki gibidir.

```
stringIndexer = StringIndexer(inputCol='title', outputCol='title_new')
model = stringIndexer.fit(df)
indexed = model.transform(df)
```

Şekil 10. İndeksleme kod parçası

### E. Veri görselleştirme

Veri seti temizlendi ve sonuç olarak kullanılacak veri seti görselleştirilerek 1-5 arası verilen oyların dağılımı incelendi. Şekil 11'deki gibi veri görselleştirmesi yapılmıştır.



Şekil 11. Veri görselleştirme

İncelemeler sonucunda 4 puan alan filmlerin sayısının diğer oylara oranla daha fazla olduğu anlaşılmış oldu. Bunun iyi bir sonuç olduğu 4 ve 5 puanlarının fazla olmasının daha fazla ve daha doğru filmler önerilebileceği sonucu çıkartıldı.

### F. Eğitim Ve Test Veri Seti

Veri temizleme sonucu oluşan yeni veri setinin %80 i eğitim ve %20 si test seti olarak kullanılmak üzere bölünmüştür. Şekil 12'de gösterilmiştir.

```
train, test = indexed.randomSplit([0.8,0.2])
```

Şekil 12. Eğitim ve Test veri seti

### G. Model Eğitimi ve Hiperparametre Ayarları

Als algoritması ve cross-validation kullanılarak model eğitime başlandı. İlk adım, Alternating Least Squares (ALS) algoritmasının kullanılmasıdır. Bu algoritmanın kullanımına dair belirlenen hiperparametreler şunlardır:

- maxIter (Maksimum İterasyon Sayısı):** ALS algoritmasının kaç iterasyon boyunca çalışacağını belirler. Bu, modelin ne kadar süreyle eğitileceğini ve optimal parametreleri öğrenmek için geçen iterasyon sayısını kontrol eder. Projede bu parametre 10 olarak belirlendi.
- regParam (Düzenleme Parametresi):** ALS algoritmasının aşırı öğrenmeyi önlemek için kullanılan bir düzenleme parametresidir. Küçük bir regParam değeri, modelin daha fazla esnek olmasına ve eğitim verilerine daha fazla uymasına izin verirken, büyük bir değer aşırı öğrenmeyi azaltabilir. Projede regParam değerleri [0.01, 0.1] olarak belirlenmiştir.
- rank (Faktör Sayısı):** ALS algoritmasının kullanacağı faktör sayısını belirler. Faktör sayısı, kullanıcının ve ürünün özelliklerini temsil eden vektörlerin boyutunu ifade eder.

Fazla faktör sayısı modelin karmaşık hale gelmesine neden olabilir, bu nedenle optimum bir değer seçilmelidir. Projede çeşitli değerler denendikten sonra en optimal sonucu vermiş olan [12, 27] değerleri kullanılmıştır.

- coldStartStrategy:** Bu parametre, daha önce görülmemiş kullanıcılar veya ürünler ile nasıl başa çıkılacağını belirler. "drop" seçeneği, bu durumları eğitim sırasında dışlamayı tercih eder.

Sonraki adımda ise cross-validation yöntemi kullanılarak modelin performansının değerlendirilmesi sağlanmaktadır. Cross-validation<sup>[5]</sup>, veri kümesini belirli bir sayıda katmana böler ve her katmanı sırayla modelin eğitim ve test süreçlerinde kullanır. Bu yöntem, modelin genel performansını daha güvenilir bir şekilde değerlendirmek için kullanılır.

- numFolds (Katman Sayısı):** Cross-validation işlemi sırasında kaç katmanın kullanılacağını belirler. Her bir katman, modelin bir kısmının test edilmesi ve geri kalan kısmının eğitilmesi için kullanılır. Bu projede 5 katmanlı bir çapraz doğrulama yapılmıştır.
- revaluator (Değerlendirici):** Cross-validation sürecinde modelin performansını ölçen metrikleri belirler. Bu projede, hata ölçümü olarak "rmse" (root mean squared error) kullanılmıştır.

Bu şekilde belirlenen hiperparametreler, ALS algoritması ve cross-validation yönteminin etkili bir şekilde uygulanmasını sağlamak için seçilmiştir. Bu ayarlar, modelin eğitim sürecinde optimum performansı elde etmeye yönelik stratejileri temsil eder.

Belirlenen hiperparametrelerin ardından, model eğitimi ve çapraz doğrulama işlemine geçildi. Bu adım, Spark MLlib kütüphanesinde bulunan CrossValidator sınıfı ile gerçekleştirildi.

ALS algoritması ve belirlenen hiperparametre seti üzerinde çapraz doğrulama işlemi Şekil 13'teki gibi başlatılır. crossvalidation nesnesi, ALS algoritmasını, belirlenen hiperparametre kombinasyonlarını ve değerlendiriciyi içerir. fit(train) fonksiyonu, eğitim veri kümesi (train) üzerinde çapraz doğrulama işlemini başlatır.

```
def cross_validation(train):
    k = 5
    alsalg = ALS(maxIter=10,
                  regParam=0.01,
                  userCol='userid',
                  itemCol='title_new',
                  ratingCol='rating',
                  nonnegative=True,
                  coldStartStrategy="drop")
    paramgrid = ParamGridBuilder().\
        addGrid(alsalg.rank, [12, 27]).\
        addGrid(alsalg.regParam, [0.01, 0.1]).\
        addGrid(alsalg.maxIter, [8]).build()
    crossvalidation = CrossValidator(estimator=alsalg,
                                     estimatorParamMaps=paramgrid,
                                     evaluator=RegressionEvaluator(
                                         metricName="rmse",
                                         predictionCol="prediction",
                                         labelCol="rating"),
                                     numFolds=k,
                                     collectSubModels=True)
    crossvalmodel = crossvalidation.fit(train)
    return crossvalmodel
cv_model = cross_validation(train)
```

Şekil 13. Çapraz Doğrulama kod parçası ve hiperparametreler

Bu işlem sırasında, veri kümesi belirli bir sayıda katmana bölünür ve her bir katman sırayla test için ayrılırken geri kalan kısım eğitim için kullanılır. Bu sayede modelin performansı, farklı veri kesitleri üzerinde değerlendirilir ve genel bir performans ölçütü elde edilir.

Sonuç olarak, *crossvalmodel* değişkeni, çapraz doğrulama işlemi tamamlandıktan sonra en iyi performansa sahip olan modeli içerir. Bu model, belirlenen hiperparametre kombinasyonlarına dayanarak veri seti üzerindeki kullanıcı ve ürün ilişkilerini öğrenmiş ve bu ilişkileri kullanarak önerilerde bulunabilecek bir durumda bulunmaktadır.

#### H. Çapraz Doğrulama Sonuçlarının Özeti

Çapraz doğrulama işlemi tamamlandıktan sonra elde edilen modellerin özet bilgileri Şekil 14'te görüntülendi. Her bir çapraz doğrulama katmanına ait modellerin ve bu modellere ait özet bilgilerin görsel bir şekilde incelenmesi, her bir modelin performansının daha anlaşılır bir şekilde değerlendirilmesini sağlar.

Ayrıca, bu sürecin sonucunda en iyi model belirlendi ve bu modelin özeti ekrana yazdırıldı. Bu adım, çapraz doğrulama işleminin genel başarısını değerlendirmek adına önemli bir aşamayı temsil eder. Her bir çapraz doğrulama katmanındaki modellerin performansının, ortalama RMSE değerleri üzerinden kontrol edilmesi, çapraz doğrulama işleminin sağladığı değerlendirmenin güvenilirliğini artırır.

Bu değerlendirme, modelin öğrenme sürecinde belirlenen hiperparametrelerin etkinliğini ve genel performansını anlamamıza yardımcı olur. Elde edilen en iyi model, çapraz doğrulama işleminin başarısını yansıtan bir ölçüt olarak ön plana çıkar ve modelin öğrenme yeteneğini en iyi şekilde temsil eder.

```
-----1-----
1
Model özet:ALSModel: uid=ALS_698cb3f5d3f9, rank=12
2
Model özet:ALSModel: uid=ALS_698cb3f5d3f9, rank=12
3
Model özet:ALSModel: uid=ALS_698cb3f5d3f9, rank=27
4
Model özet:ALSModel: uid=ALS_698cb3f5d3f9, rank=27
-----
```

Şekil 14. Çapraz doğrulama sonuçlarının özetinden bir kesit

#### İ. Tahmin Değerlerinin Hesaplanması

Model eğitimi tamamlandıktan sonra, oluşturulan çapraz doğrulama modeli (*cv\_model*) kullanılarak test veri kümesi üzerinde tahmin değerleri hesaplandı. Bu adım, modelin öğrendiği ilişkileri kullanarak, henüz görülmemiş veri noktaları için rating tahminlerinde bulunmayı içerir.

Şekil 15'teki kod satırı, *cv\_model* çapraz doğrulama modelini kullanarak test veri kümesindeki kullanıcı ve ürün kombinasyonları için rating tahminlerini oluşturur. Elde edilen *predicted\_ratings* DataFrame'i, her bir gözlemin gerçek rating değerleriyle birlikte, modelin tarafından öngörülen rating değerlerini içerir.

```
predicted_ratings=cv_model.transform(test)
print("tahminler hesaplandı. Bazı sonuçlar:")
predicted_ratings.show(5)
```

Şekil 15. Tahmini değerleri hesaplayan kod parçası

Ardından, bu tahmin değerleri ekrana yazdırılarak bazı sonuçlar incelendi. Bu sonuçlar Şekil 16'daki gibidir.

```
tahminler hesaplandı. Bazı sonuçlar:
+-----+-----+-----+-----+-----+
|userId|      title|rating|title_new|prediction|
+-----+-----+-----+-----+-----+
| 1| 101 Dalmatians 1996| 2| 308.0| 2.4549627|
| 1| 12 Angry Men 1957| 5| 261.0| 4.178587|
| 1| Batman Forever 1995| 1| 297.0| 2.3174531|
| 1|Bedknobs And Broo...| 2| 556.0| 2.3577664|
| 1| Belle De Jour 1967| 3| 732.0| 3.8846118|
+-----+-----+-----+-----+-----+
```

Şekil 16. Tahmin sonuçları

Bu kod bloğu, oluşturulan tahminlerin ilk beş gözlemini ekrana yazdırır. Bu sayede, modelin ne kadar başarılı tahminlerde bulunduğunu anlamak ve modelin performansını görsel olarak değerlendirmek mümkün olur.

Bu aşama, çapraz doğrulama modelinin gerçek dünya verileri üzerindeki performansını değerlendirmek adına önemli bir adımdır. Elde edilen tahmin değerleri, modelin eğitim sürecinde öğrendiği bilgileri nasıl uyguladığını ve kullanıcı-ürün ilişkilerini ne kadar doğru tahmin ettiğini gösterir.

#### J. Model Performansının Değerlendirilmesi: RMSE Metriği

Test veri kümesi üzerinde modelin performansını değerlendirmek için kullanılan bir metrik olan RMSE (Root Mean Squared Error) metriği, tahmin edilen değerlerle gerçek değerler arasındaki hata miktarını ölçer. Bu metriğin hesaplanması sonucu Şekil 17'de gösterildiği gibidir.

```
Rmse 1 : 1.073
Rmse 2 : 0.939
Rmse 3 : 1.120
Rmse 4 : 0.933
```

Şekil 17. RMSE değerleri

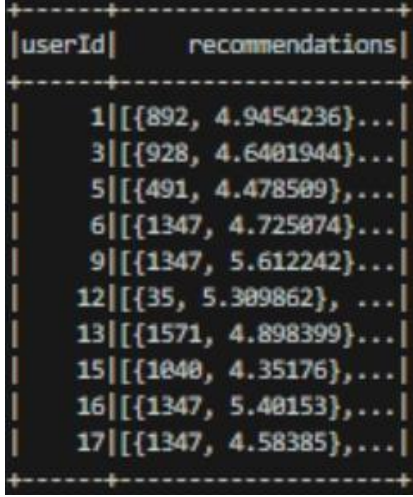
Sonuç olarak elde edilen RMSE değeri, modelin test veri kümesi üzerinde ne kadar doğru tahminler yaptığını gösterir. Düşük bir RMSE değeri, modelin yüksek doğrulukta tahminler yaptığını ifade ederken, yüksek bir RMSE değeri modelin performansında iyileştirmeler yapılması gerektiğini gösterebilir.



Bu aşama, modelin gerçek dünya verileri üzerindeki performansını değerlendirmek ve geliştirmek adına kritik bir adımdır.

### III. SONUÇLAR

Çapraz doğrulama sonuçlarına göre belirlenen en iyi modelin, tüm kullanıcılara yönelik olarak her bir kullanıcı için 10 öneri üretip bu öneriler kontrol edildi. Bu öneriler, kullanıcılara ilgilerine uygun olan ürünleri tavsiye etmek amacıyla model tarafından hesaplanmıştır. Şekil 18’de on kullanıcı için önerilen filmler gösterilmiştir.

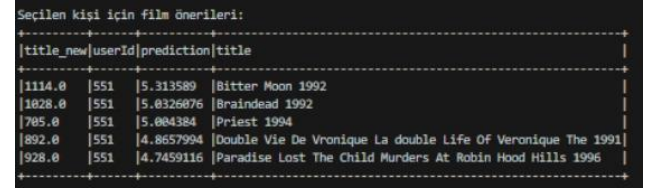


userId	recommendations
1	[[{892, 4.9454236}, ...]
3	[[{928, 4.6401944}, ...]
5	[[{491, 4.478509}, ...]
6	[[{1347, 4.725074}, ...]
9	[[{1347, 5.612242}, ...]
12	[[{35, 5.309862}, ...]
13	[[{1571, 4.898399}, ...]
15	[[{1040, 4.35176}, ...]
16	[[{1347, 5.40153}, ...]
17	[[{1347, 4.58385}, ...]

Şekil 18. On kullanıcı için film önerisi

Daha sonra kullanıcının daha önce izlediği filmler belirlenerek tüm filmlerden izlemiş olduğu filmler çıkartıldı ve kullanıcının daha önce izlemediği filmlerden oluşan bir veri seti elde edildi. Çapraz doğrulama modeli kullanılarak bu izlenmemiş filmler üzerinden tahminler

yapıldı ve kullanıcıya bu tahminler üzerinden öneriler sunuldu. Şekil 19’da seçilen kullanıcı 5 adet film önerilmiştir.



title	new	userId	prediction	title
Bitter Moon	1992	551	5.313589	
Braindead	1992	551	5.0326076	
Priest	1994	551	5.004384	
Double Vie De Veronique La double Life Of Veronique	The 1991	551	4.8657994	
Paradise Lost The Child Murders At Robin Hood Hills	1996	551	4.7459116	

Şekil 19. Seçili kullanıcı için önerilen 5 film

Sonuç olarak öneri listesi, kullanıcının ilgi alanlarına en uygun olan filmleri içerir. Bu sayede, öneri sistemi kullanıcının film tercihlerini anlamaya ve ona özel öneriler sunmaya yönelik etkili bir araç olarak kullanılabilir.

### REFERENCES

- [1] Medium, “Apache Spark ile Veri Entegrasyonu” <https://medium.com/baybaynakit/apache-spark-ile-veri-entegrasyonu-4e2897694ae1>
- [2] Medium, “Yapay Zeka ile Tavsiye Sistemleri Yazı Dizisi: 2— Alternating Least Squares(ALS) Metodu” <https://medium.com/@mbburabak/yapay-zeka-ile-tavsiye-sistemleri-yazi-dizisi-2-alternating-least-squares-als-metodu>
- [3] Kaggle, “Movierating” [https://www.kaggle.com/datasets/tientd95/movierating/data?select=movie\\_ratings\\_df.csv](https://www.kaggle.com/datasets/tientd95/movierating/data?select=movie_ratings_df.csv)
- [4] Medium, “Data Preprocessing(Veri Ön İşleme)” <https://medium.com/@ilkbahamaz/data-preprocessing-veri-on-isleme-85236484f913>
- [5] Miraç Öztürk, “Çapraz Doğrulama Teknikleri” <https://miracozturk.com/capraz-dogrulama-teknikleri-cross-validation>