DISNEY PLUS KULLANICILARI İÇİN İÇERİK ÖNERİ ARACI

THE CONTENT RECOMMENDATION TOOL FOR DISNEY PLUS USER

Mert Can GÖNEN

Bilgisayar Mühendisliği Bölümü
TOBB Ekonomi ve Teknoloji Üniversitesi, Ankara
mgonen@etu.edu.tr

Özetçe

Bu çalışmada Kaggle üzerinde bulunan "Disney+ Movies and TV Shows" veri seti kullanılarak, kullanıcılar için mümkün olduğunca tercih ettikleri içerikle benzerliklere sahip içerikler tavsiye eden bir yapay öğrenme modeli sunulmuştur. Kullanıcının girdi olarak verdiği içerikle en çok benzer özelliğe sahip 10 içerik kullanıcıya çıktı olarak sunulmuştur. Veri setini özetlemek için Python programlama diline ait gerekli kütüphaneler ve bu kütüphanelerin fonksiyonları kullanılmıştır.

Abstract

In this paper, using the "Disney+ Movies and TV Shows" dataset on Kaggle, a machine learning model is presented that recommends content that has similarities with the content user prefers as much as possible. The first 10 content that has the most similar features with the content given by the user as input are presented to the user as output. The necessary libraries of the Python programming language and the functions of these libraries were used to summarize the dataset.

1. Giriş

Bu çalışmadaki motivasyon ve amaç, Disney+ platformu üzerinde tüketilecek içerik arayan kullanıcılara, tercih ettikleri içeriklere mümkün oldukça benzer içerikler sunmaktır. Çözüme kavuşturulmak istenen problem ise bir regresyon problemidir. Content-based filtering kullanılmıştır. Kullanıcının tercih ettiği içeriğe göre diğer içeriklerin benzerlik oranı değişkenlik göstermektedir. Örnek vermek gerekirse; bir kullanıcı A içeriğini beğeniyor ve buna benzer içerikleri öğrenmek istiyor. B içeriğinin A'ya benzerliği (cosine benzerliği) %85 iken, C içeriğinin A'ya benzerliği %10 oluyor. Ancak başka bir kullanıcı D içeriğini beğeniyor. B ve C içeriklerinin D'ye benzerliği sırasıyla %10, %90 olabilir. Aynı zamanda benzer içerikleri ararken "evet, hayır" gibi kesin değişkenler kullanamıyoruz. Bu sebeplerden ötürü problemimiz bir regresyon problemi olarak adlandırılabilir. Bu çalışmadaki hedef başarım metrikleri için, Count Vectorizer ve Tfid Vectorizer metotlarında benzerlik (similarity score) puanının 0.5'e mümkün oldukça yakın olmasıdır. Çünkü benzerlik için bir metot uygularken ele alacağımız özellikler içeriğe ait; tür, oyuncu kadrosu ve açıklama verileri. Bunların çok yüksek oranda benzerliğe sahip olabilmesi için neredeyse tüm verilerin aynı olması gerekir. Bu da içeriklerin aynı olması anlamına geliyor. Böyle bir durum söz konusu olamaz.

2. Literatür Araştırması

Önceki dönemlerde farklı okullarda da dönem projesi/ödev olarak yapılan bu modellemede, modeli oluşturan kişiler/ekipler Count Vectorizer ve TfidVectorizer benzeri yapıları kullanarak modellerini oluşturmuşlar.

3. Veri Seti, Veri Özellikleri ve Öznitelikleri

- Kullanılan veri setinin linki:

  https://www.kaggle.com/datasets/shivamb/disney-movies-and-tv-shows

- Veri kümesi:

  show_id = sıralı olarak otomatik atanmış string id'ler (herhangi bir anlam içermiyor)

  type = içeriğin türünü (film/dizi) belirten string

  title = içeriğin ismini belirten string

  director = içeriğin yönetmenini belirten string

  cast = içerikte rol alan aktörleri ve aktrisleri belirten string

  country = filmin ait olduğu ülkeyi belirten string

  date_added = içeriğin platforma eklendiği tarihi belirten string

  release_year = içeriğin vizyona giriş tarihi

  rating = içeriğin izleyici kategorisini (15+, 13+ gibi) belirten string

  duration = içeriğin süresini belirten string

  listed_in = içeriğin ait olduğu türü belirten string

  description = filme ait kısa bir tanımlama metni

- Önişleme aşamaları

  Uygulama aşamasında daha temiz ve kullanılabilir bir veri setine sahip olmak için veri seti bazı önişleme aşamalarından geçirildi.

  a) Exploratory Data Analysis (EDA) aşamasında ve content-based filtering aşamasında kullanılmayacak olan **show_id**, **country**, **date_added** ve **rating** sütunları veri setinden atıldı. **Director** ve **release_year** gibi sütunlar Exploratory Data Analysis (EDA) aşamasında kullanılacağı için veri setinden dışarı atılmadı.

  b) Modellemeye başlamadan önce duplicate verilerin atılması kararlaştırılmıştı. Ancak veri setinde herhangi bir veriye birden çok kez rastlanmadı.

  c) Modellemede Count Vectorizer ve Tfid Vectorizer matrislerini oluştururken kullanacağımız title, cast, listed_in, description sütunlarında bulunan boş alanları "NA" string'i ile dolduruldu. Bunun sebebi metadata soup oluşturmadan önce kullanılacak tüm verilere "clean_text" metodu uygulanmasıdır.

  d) listed_in adı altında bulunan sütunun ismini daha anlaşılabilir olması için **"genre"** olarak değiştirdik.

  e) Veri setinin bu işlemlerden sonraki halini **Ekler** altında **Ek0.jpg** olarak görebilirsiniz.

- Veri setinde bulunan veriler nominal türde verilerdir. Herhangi bir sayısal, karşılaştırılabilir bir değer bulunmamaktadır. Karşılaştırılabilir hale getirmek çok mümkün değil. Ancak model olarak matrisler oluşturup onlar üzerinde

benzerliğe göre column'daki verileri arttıracağımız için (metadata soup'da geçen kelimenin count sayısını +1 yapmak vs.) karşılaştırma bu verilerin bir araya getirilip benzerlik skoru oluşturmasıyla ve bu skorların karşılaştırılmasıyla yapılacak.

- Veri setinde bulunan bu nominal veriler sırasız verilerdir. Aralarında sıralama yapmanın sonuca herhangi bir etkisi bulunmamaktadır. Ancak benzerlik skorlarını oluşturduktan sonra verileri benzerlik skoru yüksekten düşüğe göre sıralanmıştır.

- Veri dağılımı dikkatle incelersek veri seti içerisindeki içeriklerin %73'ünü filmler oluştururken, %27'sini diziler oluşturuyor. Grafiği **Ekler** altında **Ek1.png** olarak bulabilirsiniz.

- İçeriklerin vizyona giriş tarihlerindeki dağılıma bakarsak özellikle 2000'li yılların başından günümüze kadar çok ciddi bir artış söz konusu. Bu verilere ait grafiği **Ekler** altında **Ek2.png** olarak bulabilirsiniz.

- Count Vectorizer'ı oluşturmak için kullandığımız veri türleri arasında yer alan "**director**" sütununda en fazla yer alan verilere/yönetmenlere ait grafiği **Ekler** altında **Ek3.png** olarak bulabilirsiniz.

- Count Vectorizer'ı oluşturmak için kullandığımız veri türleri arasında yer alan "**cast**" sütununda en fazla yer alan aktörlere ait grafiği **Ekler** altında **Ek4.png** olarak bulabilirsiniz.

- Count Vectorizer'ı oluşturmak için kullandığımız veri türleri arasında yer alan "**genre**" sütununda en fazla yer alan türlere ait grafiği **Ekler** altında **Ek5.png** olarak bulabilirsiniz.

- Veri setinde bulunan veriler imbalanced haldeler. Ancak elimizde bulunan içeriklerin türe ve cast'e göre genel olarak[1] çok düzensiz olmadığı için herhangi bir balance işlemi gerçekleştirilmedi.

- Veriler birden fazla sınıflara ait, her içerik ayrı genre/genre'lere ait. Her içeriğin cast'ında birden fazla aktör/aktris bulunmaktadır. Düz metin verileridir.

- Veriler üzerinde normalizasyon uygulanmadı. Çünkü normalizasyın uygulanacak bir türde değiller.

4. Kullanılan Modeller ve Test Sonuçları

Bu çalışma özelinde 2 adet model kullanıldı. Bunlardan ilki Count Vectorizer. Bir diğeri ise Tfid Vectorizer. İkisine ait çıktılar ve birbirlerine karşı karşılaştırılmaları raporun ilerleyen kısmında hem sözel hem de görsel olarak (**Ekler** kısmında bulunan grafikler) sunuldu.

- Öncelikle eldeki veriler önişlemeden geçirildi.

- Ardından ilk olarak Count Vectorizer'e dahil edilecek olan öznitelikler clean text haline getirildi.

örnek:

[1] Birkaç adet genre ya da aktör dışında diğer veriler arasında çok uçuk farklar bulunmamaktadır.

action-adventure,family,sciencefiction

- Ardından clean text haline getirilen özniteliklerden metadata_soup oluşturuldu.

- Metadata_soup ile birlikte cosine matrix oluşturuldu ve bu cosine matrix'le birlikte cosine similarity hesaplandı.

- İlk olarak test verilerinden biri olan herhangi bir içeriğin **title** özniteliği, data, filmlere ait indexler ve cosine similarity ile birlikte parametre olarak **get_recommendations_new** fonksiyonuna veriliyor.

- Bu metot benzerlik skoru en yüksek olan 10 içeriği skorlarla birlikte return ediyor.

- Modeli eğitmek için veri setinin %80'i, test etmek için ise %20'si kullanıldı.

- Count Vectorizer modelinin kullanılma sebepleri;

    - Tamamen text içeren veri setleri için kullanışın çok ideal olması.
    - Verileri herhangi bir ölçeklemeye sokmadan ya da label'a ait yapmadan, tamamen seçilen özniteliklere göre benzerlik skorlarını oluşturması.
    - Öneri sistemlerinde çok yaygın olarak kullanılan bir model olması.
    - İmplementasyonunun kolay olması ve iyi sonuçlar vermesi.

- **<u>Count Vectorizer Sonuçları</u>**

    - Test için ilk olarak, modeli test etmek için ayrılan verilerden olan ve "Marvel Studios' Captain America: The Winter Soldier" içeriği seçildi. Bu teste ait sonuçları görsel olarak **Ekler** altında **Ek6-CV.png** olarak bulabilirsiniz. Sözel olarak değerlendirmek gerekirse, önerilen içeriklerin benzerlik skorları sırasıyla (0.70 – 0.53 – 0.53 – 0.48 – 0.46 – 0.46 – 0.46 – 0.44 – 0.37 – 0.36) olarak gözlemlendi. Modeli eğitmeden önceki beklentimiz benzerlik oranının 0.50 civarlarında olmasıydı. Bu sonuç beklentilerimizin de üstünde çıktı.

    - Bu 10 adet öneriye ait benzerlik skorlarının ortalaması ise 0.474 olarak karşımıza çıktı. 10 öneriye ait ortalama benzerlik skorunun 0.50'ye yakın olması beklentilerimize yakın bir karşılık oldu.

    - Bu 10 adet öneriye ait benzerlik skorlarının ortalamasının değişimine bakarsak;
        1 öneride => 0.70
        2 öneride => 0.61
        3 öneride => 0.57
        4 öneride => 0.55
        5 öneride => 0.54
        6 öneride => 0.52
        7 öneride => 0.51
        8 öneride => 0.50
        9 öneride => 0.49
        10 öneride => 0.47

mean değerine sahip oluyor. Bu teste ait sonuçları görsel olarak **Ekler** altında **Ek7-CV.png** olarak bulabilirsiniz.

- **<u>Tfid Vectorizer Sonuçları</u>**

    - İkinci model olarak Tfid Vectorizer yöntemini modelimizi eğitmek için

kullandık. Burada da input olarak Count Vectorizer modelinin sonuçlarını hesaplarken yararlandığımız "Marvel Studios' Captain America: The Winter Soldier" içeriğini input olarak öneri sistemine verdim. Bu teste ait sonuçları görsel olarak **Ekler** altında **Ek6-Tfid.png** olarak bulabilirsiniz. Sözel olarak değerlendirmek gerekirse, önerilen içeriklerin benzerlik skorları sırasıyla (0.68 – 0.53 – 0.49 – 0.46 – 0.42 – 0.40 – 0.30 – 0.30 – 0.27 – 0.26) olarak gözlemlendi. Modeli eğitmeden önceki beklentimiz benzerlik oranının 0.50 civarlarında olmasıydı. Bu sonuçta beklentilerimizin de üzerine çıkılan noktalar oldu.

- Bu 10 adet öneriye ait benzerlik skorlarının ortalaması ise 0.416 olarak karşımıza çıktı. 10 öneriye ait ortalama benzerlik skorunun 0.50'ye yakın olması beklentilerimize yakın bir karşılık oldu.

- Bu 10 adet öneriye ait benzerlik skorlarının ortalamasının değişimine bakarsak;
  1 öneride => 0.68
  2 öneride => 0.61
  3 öneride => 0.57
  4 öneride => 0.54
  5 öneride => 0.52
  6 öneride => 0.50
  7 öneride => 0.47
  8 öneride => 0.45
  9 öneride => 0.43
  10 öneride => 0.41

mean değerine sahip oluyor. Bu teste ait sonuçları görsel olarak **Ekler** altında **Ek7-Tfid.png** olarak bulabilirsiniz.

- Aynı adımları farklı bir test verisi için tekrar izlersek aşağıdaki maddelerde bulunan sonuçlara ulaşıyoruz.

- **<u>Count Vectorizer Sonuçları</u>**

  - Test için ilk olarak, modeli test etmek için ayrılan verilerden olan ve "Ice Age: A Mammoth Christmas" içeriği seçildi. Bu teste ait sonuçları görsel olarak **Ekler** altında **Ek8-CV.png** olarak bulabilirsiniz. Sözel olarak değerlendirmek gerekirse, önerilen içeriklerin benzerlik skorları sırasıyla (0.78 – 0.57 – 0.51 – 0.45 – 0.45 – 0.45 – 0.45 – 0.41 – 0.41 – 0.41) olarak gözlemlendi. Modeli eğitmeden önceki beklentimiz benzerlik oranının 0.50 civarlarında olmasıydı. Bu sonuç beklentilerimizin de üstünde çıktı.

  - Bu 10 adet öneriye ait benzerlik skorlarının ortalaması ise 0.487 olarak karşımıza çıktı. 10 öneriye ait ortalama benzerlik skorunun 0.50'ye yakın olması beklentilerimize yakın bir karşılık oldu.

  - Bu 10 adet öneriye ait benzerlik skorlarının ortalamasının değişimine bakarsak;
    1 öneride => 0.78
    2 öneride => 0.68
    3 öneride => 0.62
    4 öneride => 0.58
    5 öneride => 0.55
    6 öneride => 0.53
    7 öneride => 0.52
    8 öneride => 0.51
    9 öneride => 0.49
    10 öneride => 0.48

mean değerine sahip oluyor. Bu teste ait sonuçları görsel olarak **Ekler** altında **Ek9-CV.png** olarak bulabilirsiniz.

- **Tfid Vectorizer Sonuçları**

  - Test için ilk olarak, modeli test etmek için ayrılan verilerden olan ve "Ice Age: A Mammoth Christmas" içeriği seçildi. Bu teste ait sonuçları görsel olarak **Ekler** altında **Ek8-Tfid.png** olarak bulabilirsiniz. Sözel olarak değerlendirmek gerekirse, önerilen içeriklerin benzerlik skorları sırasıyla (0.37 – 0.32 – 0.31 – 0.30 – 0.23 – 0.22 – 0.21 – 0.20 – 0.19 – 0.17) olarak gözlemlendi. Modeli eğitmeden önceki beklentimiz benzerlik oranının 0.50 civarlarında olmasıydı. Bu sonuç beklentilerimizi de karşılamadı.

  - Bu 10 adet öneriye ait benzerlik skorlarının ortalaması ise 0.254 olarak karşımıza çıktı. 10 öneriye ait ortalama benzerlik skorunun 0.50'ye yakın olmaması modelin beklentileri karşılamamasına sebep oldu.

  - Bu 10 adet öneriye ait benzerlik skorlarının ortalamasının değişimine bakarsak;
    1 öneride => 0.37
    2 öneride => 0.34
    3 öneride => 0.33
    4 öneride => 0.32
    5 öneride => 0.31
    6 öneride => 0.29
    7 öneride => 0.28
    8 öneride => 0.27
    9 öneride => 0.26
    10 öneride => 0.25

mean değerine sahip oluyor. Bu teste ait sonuçları görsel olarak **Ekler** altında **Ek9-Tfid.png** olarak bulabilirsiniz.

- Çıkan sonuçlara baktığımız zaman Count Vectorizer hem sayısal olarak (benzerlik skoruna göre) hem de mantıksal olarak Tfid Vectorizer yöntemine göre çok daha iyi sonuçlar veriyor. Bu tür öneri sistemlerin de Count Vectorizer kullanmak, Tfid Vectorizer kullanmaktan çok daha avantajlı.

5. Sonuçlar

- Bu çalışmada yapılanları kısaca özetlemek gerekirse, tamamen text yapısından oluşan ve herhangi bir şekilde sayısal bir normalizasyona giremeyip sayısal veriye dönüşemeyen özniteliklerle birlikte, Count Vectorizer ve Tfid Vectorizer metotlarını kullanarak bir içerik öneri aracı oluşturuldu.

- Çalışmada Count Vectorizer, Tfid Vectorizer yapılarını öğrenmemin yanı sıra, Cosine similarity hesaplamalarını da net bir biçimde anladım. Text olup da sayısal verilere dönüşmeyecek (en azından öznitelik ölçekleme kısmında manuel olarak herhangi bir şey yapılmadı) verilerle neler yapılabileceğini öğrendim.

- Bu çalışma özelinde Tfid Vectorizer metodunun node'lar ve graph'lar kullanılarak yapılan bir yolu daha vardı. Ancak onu koduma bir türlü implement edemedim.

- Veri seti ve model için herhangi bir loss, accuracy, precision veya recall

hesaplaması yapılamıyor. Çünkü modeller benzerlik skoru üzerine çalışıyor. Herhangi bir şekilde 0/1 veya Evet/Hayır gibi cevaplar üretmiyor. Bu yüzden sonuçlar karşılaştırılırken kullanıcıya önerilen içeriğin model üzerinde Count Vectorizer ve Tfid Vectorizer kullanıldığında oluşan benzerlik (similarity) skoru kullanılmıştır.

DEMO: https://youtu.be/T67ag8gRSkQ