Gişe Verilerine Göre Hasılat Tahmini

Veri Toplama ve Veri Ön İşleme Uygulamaları

Miray Gürbüz
Bilişim Sistemleri Mühendisliği
Kocaeli Üniversitesi
Kocaeli, Türkiye
221307031

Zeynep Yılmaz
Bilişim Sistemleri Mühendisliği
Kocaeli Üniversitesi
Kocaeli, Türkiye
221307012

***Özet*—Bu raporda Yazılım Geliştirme Laboratuvarı – I dersinin projesinin veri toplama ve veri ön işleme aşamaları ayrıntılı olarak ele alınmıştır. Box Office Mojo web sitesinden 2009 – 2024 yılları arasında kaydedilmiş 15 yıllık günlük gişe verileri web kazıma yöntemiyle toplanmıştır. Veri toplama Python dilinde Selenium ve BeautifulSoup4 kullanılarak gerçekleştirilmiştir. Veri ön işleme aşamasında verideki eksik bilgiler doldurulmuş; zaman serisinin yeniden örneklenmesi, trend ve mevsimsellik bileşenlerinin ayrıştırılması, otokorelasyon durumunun incelenmesi adımları uygulanmıştır. Bu adımlar sonrasında çıkartılan sonuçlar doğrultusunda özellik mühendisliği yapılarak projenin ikinci aşamasındaki modelleme adımında yarar sağlayacak yeni sütunlar eklenmiştir.**

Anahtar Kavramlar—zaman serisi, web kazıma, veri ön işleme, yeniden örnekleme, trend, mevsimsellik, otokorelasyon, özellik mühendisliği

I. GİRİŞ

Zaman serileri, belirli bir veri ögesinin zaman içinde düzenli aralıklarla ölçülmesiyle elde edilen gözlemlerden oluşur[1]. Proje kapsamında günlük olarak ölçülen gişe verileri kullanılarak hasılat tahmini yapmak amaçlanmaktadır. Projenin bu aşamasında veri toplama ve veri ön işleme adımları gerçekleştirilmiştir.

II. VERİ TOPLAMA

A. *Veri Kaynağı Seçme*

Veri kaynağı seçilirken verilerin düzenli aralıklarla kayıt altına alınmış olması, eksik verilerin olmaması ve kaynağın güvenilirliği önemlidir. Tüm bunlardan dolayı Amazon şirketi olan IMDb'ye 2008 yılından beri bağlı olan Box Office Mojo[2] web sitesi kaynak olarak seçilmiş ve sitedeki Daily sayfasındaki günlük verilerden yararlanılmıştır.

B. *Web Kazıma Aşaması*

Box Office Mojo web sitesinden 2009-2024 yılları arasında kaydedilmiş 5781 günlük gişe verisi çekilmiştir. Selenium ve BeautifulSoup4 kullanılarak 15 yıla ait 15 adet sayfanın her birindeki günlük satır kayıtları çekildikten sonra, BeautifulSoup4 ve Requests kütüphaneleri kullanılarak 5781 güne ait 5781 adet sayfanın her birinden o gün vizyonda olan filmlerin toplam hasılat değerleri toplanmış ve toplanmış olan günlük toplam hasılat değeri çekilmiştir. Bu işlem concurrent.futures kütüphanesindeki ThreadPoolExecutor[3] sınıfı kullanılarak paralel olarak gerçekleştirilmiştir.

C. *Veri Sütunlarının İçeriği*

- *url:* O günün bireysel sayfasının URL bilgisini içerir.
- *date:* O güne ait tarihi içerir.
- *day_of_year:* Yılın kaçıncı günü olduğunu belirtir.
- *releases:* O gün vizyonda olan film sayısını gösterir.
- *top_1_gross:* O gün hasılat bakımından birinci olan filmin günlük hasılatını belirtir.
- *top_1_release:* O gün hasılat bakımından birinci olan filmin adını içerir.
- *top_10_gross:* O gün hasılat bakımından ilk on film arasına giren filmlerin günlük toplam hasılatını gösterir.
- *total_gross:* O gün vizyonda olan tüm filmlerin günlük toplam hasılatını belirtir.

III. VERİ ÖN İŞLEME

Veri ön işleme aşamasında; tarih sütunun formatının belirlenmesi ve indeks olarak seçilmesi, zaman değişkenlerinin eklenmesi, zaman serisinin yeniden örneklenmesi, trend ve mevsimsellik bileşenlerinin ayrıştırılması, otokorelasyon, özellik mühendisliği, veri analizi ve verilerin normalizasyonu adımları uygulanmıştır.

A. *Zaman Serisinin Yeniden Örneklenmesi*

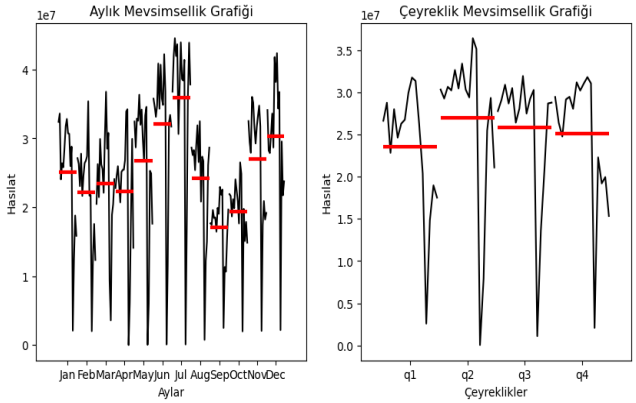
Zaman serisi yeniden örneklenerek verilerin tutarlı bir sıklıkta dağıtılması sağlanılır. Yeniden örnekleme orijinal verinin zaman frekansını değiştirerek gerçekleştirilir.[4] İki adet yeniden örnekleme yöntemi vardır:

- *Yukarı Örnekleme:* Zaman frekansını arttırır. Örneğin; günlerden saatlere ayırmak. Bu sayede veri sayısı artar.
- *Aşağı Örnekleme:* Zaman frekansını azaltır. Örneğin; günlerden haftalara ayırmak. Bu sayede veri sayısı azalır ve daha yumuşak bir gözlem yapılabilir.

Proje kapsamında çekilen veriler günlük frekansta olduğundan aşağı örnekleme tercih edilmiştir. Günlük veri; haftalık, aylık, çeyreklik ve yıllık olarak yeniden örneklenmiştir.

B. Trend ve Mevsimsellik Bileşenleri

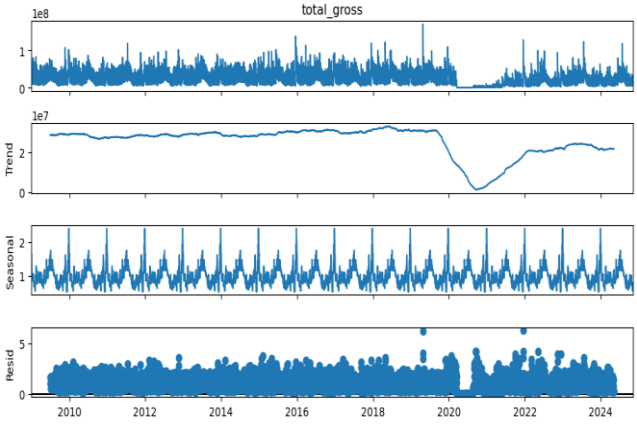
- *Trend:* Verideki uzun vadede gözlemlenen genel hareketi ifade eder.
- *Mevsimsellik:* Verinin belirli bir periyodik döngüde değişen ve bu değişikliklerin önceden tahmin edilebilir olduğu durumu ifade eder.



Görsel 1: Aylık ve Çeyreklik Mevsimsellik Grafiği

Yukarıdaki grafikler incelendiğinde veride aylık ve çeyreklik olarak mevsimsellik gözlemlenmektedir. Kırmızı çizgiler toplam hasılat verisinin ilgili ay ve çeyrekliklerdeki ortalama değerlerini gösterirken, siyah çizgiler toplam hasılat verisinin yıllar boyu olan değişimini gösterir. Kırmızı çizgilerin, yani ortalama değerlerin artıp azalıyor olması mevsimselliğe işarettir.

1) Trend ve Mevsimsellik Bileşenlerinin Ayrıştırılması



Görsel 2: Mevsimsel Ayrışım Grafiği

Statsmodels kütüphanesinin seasonal_decompose fonksiyonu kullanılarak toplam hasılat sütununun trend ve mevsimsellik bileşenleri ayrıştırılmıştır. Verinin yıllık döngü yaptığı varsayılmış ve bileşenler multiplicative model ile ayrıştırılmıştır.

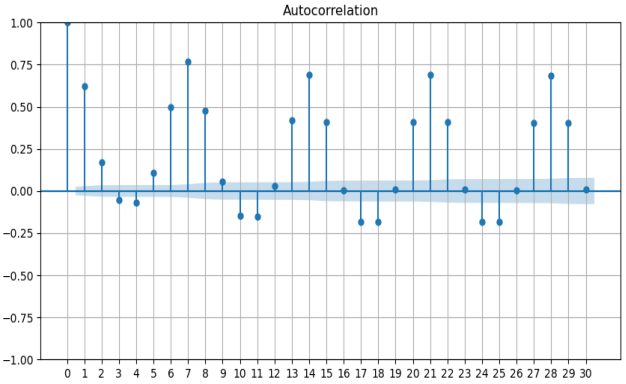
Grafik incelendiğinde trend bileşeninin 2020 sonrasında pandemi etkisiyle düşüş yaşamış olabileceği yorumu yapılabilmektedir. Mevsimsellik grafiğinde örüntü gözlemlenmesi verinin mevsimselliğinin olduğuna işaret etmektedir.

C. Otokorelasyon

Otokorelasyon, zaman serilerinde serinin gecikmeli değerleriyle şu anki değerleri arasındaki ilişkiyi ölçer. Değerlerin kendi kendisiyle olan korelasyonunu gösterir.

Otokorelasyon katsayısı 1 ile -1 arasında değişiklik gösterir:

- *1'e yakın değerler:* Pozitif otokorelasyon anlamına gelir. Önceki değerlerle benzer yönde bir değişim eğilimi gösterir.
- *0'a yakın değerler:* Otokorelasyonun zayıf olduğunu gösterir. Geçmiş verilerin mevcut veriler üzerindeki etkisi yok sayılır.
- *-1'e yakın değerler:* Negatif otokorelasyon anlamına gelir. Önceki değerlerin tersine hareket eder.



Görsel 3: Otokorelasyon Grafiği

Yukarıdaki grafik, otokorelasyonun gözlenmesi için Statsmodels'in ACF fonksiyonuyla 30 gecikmeye kadar oluşturulan otokorelasyon grafiğidir.

Grafiğe bakıldığı zaman otokorelasyon değerlerinin 7 günlük aralıklarla yükseldiği gözlemlenmektedir. Buradan yola çıkılarak günlük toplam hasılat değerlerinin *haftalık* olarak mevsimsellik gösterdiği şeklinde bir yorum yapılabilir.

D. Özellik Mühendisliği

- *Özellik:* Verideki gözlemler hakkında bilgi veren her bir değişken veya özniteliktir.
- *Özellik mühendisliği:* Ham veriyi modellemede kullanılabilecek özelliklere dönüştürmek için seçme, manipüle etme ve dönüştürme sürecidir[5].

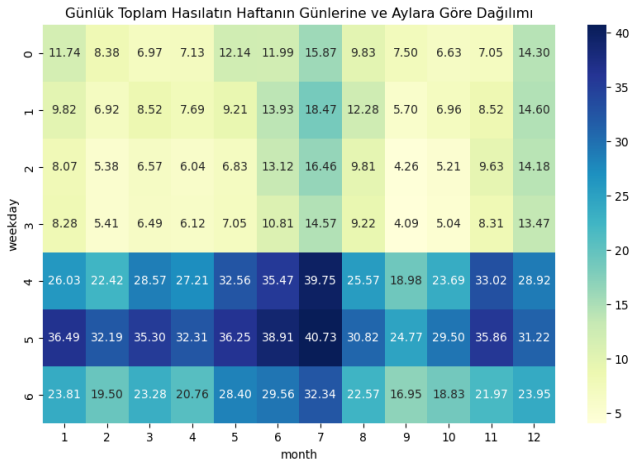
Özellik mühendisliği adımında veri setine; 7 günlük gecikmeli toplam hasılat değeri, 7 ve 30 günlük toplam hasılat hareketli ortalamaları, günlük getirileri, ilgili günün resmi tatil olup olmadığı gibi bilgileri içeren sütunlar eklenmiştir. Döngüsel sütunlara sinüs ve kosinüs dönüşümleri uygulanmıştır.

E. Veri Analizi

Bu adımda günlük toplam hasılat verileri üzerinde çeşitli analizler yapılarak filmlerin hasılat performansları gözlemlenmiştir.

1) Günlük Toplam Hasılatın Haftanın Günlerine ve Aylara Göre Dağılımı

Seaborn kütüphanesi kullanılarak haftanın günlerine ve aylara göre günlük toplam hasılat verilerinin dağılımı görselleştirilmiştir. Isı haritası grafiği ile elde edilen sonuçlar, hangi günlerin ve ayların daha yüksek gelir sağladığını ortaya koymuştur.

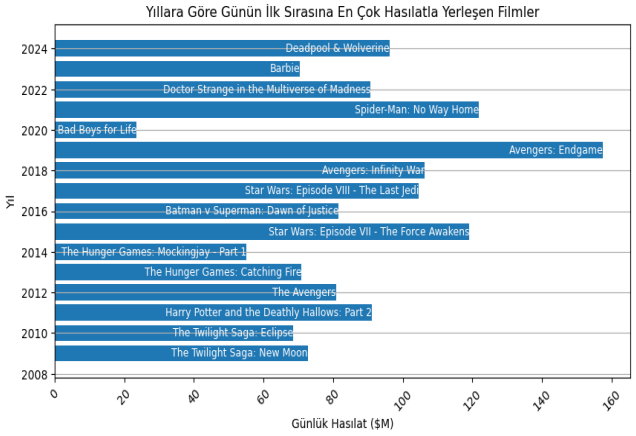


Görsel 4: Günlük Toplam Hasılatın Dağılım Grafiği

Grafikte toplam hasılat değerinin en yüksek olduğu ayın temmuz, günün ise cumartesi olduğu gözlemlenmektedir.

2) Yıllara Göre Günün İlk Sırasına En Çok Hasılatla Yerleşen Filmler

Yıllara göre her günün en yüksek hasılatını elde eden filmler incelenmiş ve yıllık bazda hangi filmlerin daha yüksek hasılatlarla zirveye yerleştiği analiz edilmiştir.



Görsel 5: Yıllara Göre İlk Sıraya Yerleşen Filmler Grafiği

Grafikte 2009-2024 yılları arasında günün ilk sırasına en çok hasılatla yerleşen filmler listelenmiştir. Bu bağlamda en başarılı yılın 2019, filmin ise *Avengers: Endgame* olduğu yorumu yapılabilmektedir.

F. Normalizasyon

- *Normalizasyon:* Veri değerlerini ortak bir ölçeğe veya değer dağılımına hizalayan veri dönüştürme işlemidir[6].

Min-maks ölçekleme normalizasyon işlemidir. Değerler 0 ile 1 arasında yeniden örneklenir[7]. Min-maks ölçekleme formülü şudur:

$$X_{normalize} = \frac{X - X_{min}}{X_{max} - X_{min}}$$
$$X_{ölçeklenmiş} = X_{normalize} \times (maks - min) + min$$

[8]

Sklearn kütüphanesi MinMaxScaler fonksiyonu ile nümerik değer taşıyan sütunlar normalize edilmiş ve normalize edilmiş veri seti kaydedilmiştir.

REFERANSLAR

[1] "Time Series Analysis: The Basics", https://www.abs.gov.au/websitedbs/D3310114.nsf/home/Time+Series+Analysis:+The+Basics Erişim Tarihi: [10.11.2024]

[2] https://www.boxofficemojo.com/ Erişim Tarihi: [04.11.2024]

[3] https://docs.python.org/3/library/concurrent.futures.html Erişim Tarihi: [10.11.2024]

[4] "How to Resample Time Series Data in Python?", https://www.geeksforgeeks.org/how-to-resample-time-series-data-in-python/ Erişim Tarihi: [10.11.2024]

[5] "Feature Engineering Explained", https://builtin.com/articles/feature-engineering Erişim Tarihi: [12.11.2024]

[6] "Normalization", https://c3.ai/glossary/data-science/normalization/ Erişim Tarihi: [12.11.2024]

[7] https://www.datacamp.com/tutorial/normalization-in-machine-learning Erişim Tarihi: [11.10.2024]

[8] https://scikit-learn.org/1.5/modules/generated/sklearn.preprocessing.MinMaxScaler.html Erişim Tarihi: [12.11.2024]