

**T.C.  
SAKARYA ÜNİVERSİTESİ  
FEN BİLİMLERİ ENSTİTÜSÜ**

**BİR PERAKENDE FİRMASINDA METİN BENZERLİĞİ VE  
TEKİL DEĞER AYRIŞIMI ALGORİTMASI TABANLI ÜRÜN  
ÖNERİ SİSTEMİNİN OLUŞTURULMASI**

**YÜKSEK LİSANS TEZİ**

**Bilal ERDEMİR**

**Enstitü Anabilim Dalı : ENDÜSTRİ MÜHENDİSLİĞİ**

**Tez Danışmanı : Doç. Dr. Alper KİRAZ**

**Haziran 2022**

**T.C.  
SAKARYA ÜNİVERSİTESİ  
FEN BİLİMLERİ ENSTİTÜSÜ**

**BİR PERAKENDE FİRMASINDA METİN BENZERLİĞİ VE  
TEKİL DEĞER AYRIŞIMI ALGORİTMASI TABANLI ÜRÜN  
ÖNERİ SİSTEMİNİN OLUŞTURULMASI**

**YÜKSEK LİSANS TEZİ**

**Bilal ERDEMİR**

**Enstitü Anabilim Dalı : ENDÜSTRİ MÜHENDİSLİĞİ**

**Bu tez 27.06.2022 tarihinde aşağıdaki jüri tarafından oybirliği / oyçokluğu ile kabul edilmiştir.**

**Doç. Dr.  
Alper KİRAZ  
Jüri Başkanı**

**Dr. Öğr. Üyesi  
Merve Cengiz Toklu  
Üye**

**Dr. Öğr. Üyesi  
Çağatay Teke  
Üye**

## **BEYAN**

Tez içindeki tüm verilerin akademik kurallar çerçevesinde tarafımdan elde edildiğini, görsel ve yazılı tüm bilgi ve sonuçların akademik ve etik kurallara uygun şekilde sunulduğunu, kullanılan verilerde herhangi bir tahrifat yapılmadığını, başkalarının eserlerinden yararlanılması durumunda bilimsel normlara uygun olarak atıfta bulunulduğunu, tezde yer alan verilerin bu üniversite veya başka bir üniversitede herhangi bir tez çalışmasında kullanılmadığını beyan ederim.

Bilal ERDEMİR

27.06.2022

## TEŞEKKÜR

Yüksek lisans eğitimim boyunca değerli bilgi ve deneyimlerinden yararlandığım, her konuda bilgi ve desteğini almaktan çekinmediğim, araştırmanın planlanmasından yazılmasına kadar tüm aşamalarında yardımlarını esirgemeyen, teşvik eden, aynı titizlikte beni yönlendiren değerli danışman hocam Doç. Dr. Alper KİRAZ'a teşekkürlerimi sunarım.

Hayatımın her anında sevgisini hissettiğim sevgili annem Suna ERDEMİR'e, bana güvenen, destek olan değerli babam Ahmet ERDEMİR'e, tez yazma sürecinde zaman ve gerekli imkanları oluşturarak işle beraber bu süreci yönetmemi sağlayan değerli büyüğüm Cana Hande UĞUR'a ve bu zorlu süreçte benden desteğini hiçbir zaman esirgemeyen değerli abim Doğan ADIGÜZEL'e teşekkür ederim.

## İÇİNDEKİLER

|                                                |      |
|------------------------------------------------|------|
| TEŞEKKÜR .....                                 | i    |
| İÇİNDEKİLER .....                              | ii   |
| SİMGELER VE KISALTMALAR LİSTESİ .....          | v    |
| ŞEKİLLER LİSTESİ .....                         | vi   |
| TABLolar LİSTESİ .....                         | vii  |
| ÖZET .....                                     | viii |
| SUMMARY .....                                  | ix   |
|                                                |      |
| BÖLÜM 1.                                       |      |
| GİRİŞ .....                                    | 1    |
|                                                |      |
| BÖLÜM 2.                                       |      |
| KAYNAK ARAŞTIRMASI .....                       | 4    |
|                                                |      |
| BÖLÜM 3.                                       |      |
| MATERYAL VE METOD .....                        | 10   |
| 3.1. Metin Benzerliği Algoritmaları.....       | 10   |
| 3.1.1. Kosinüs benzerliği .....                | 10   |
| 3.1.2. Monge Elkan mesafesi .....              | 11   |
| 3.1.3. Bag mesafesi .....                      | 11   |
| 3.1.4. Jaccard benzerlik katsayısı .....       | 12   |
| 3.1.5. Sorensen Dice benzerlik katsayısı ..... | 12   |
| 3.1.6. Twersky indeksi .....                   | 13   |
| 3.1.7. Levenshtein mesafesi .....              | 13   |
| 3.1.8. Needleman Wunsch mesafesi .....         | 14   |

|                                             |    |
|---------------------------------------------|----|
| 3.1.9. Jaro – Winkler mesafesi .....        | 14 |
| 3.1.10. Hamming mesafesi .....              | 15 |
| 3.1.11. Smith-Waterman mesafesi .....       | 15 |
| 3.1.12. Ratcliff/Obershelp benzerliği ..... | 16 |
| 3.2. Tekil Değer Ayrışımı Algoritması ..... | 16 |
| 3.3. K-Ortalamlar Algoritması .....         | 18 |

## BÖLÜM 4.

|                                                                         |    |
|-------------------------------------------------------------------------|----|
| UYGULAMA .....                                                          | 20 |
| 4.1. Verinin Elde Edilmesi .....                                        | 21 |
| 4.2. Yöntem .....                                                       | 22 |
| 4.2.1. Kullanılan araç-gereçler .....                                   | 22 |
| 4.2.2. Veri ön işleme (veri manipülasyonu) .....                        | 23 |
| 4.2.2.1. Ürün gruplarının düzenlenmesi.....                             | 23 |
| 4.3. Segment ve Fonksiyonların Oluşturulması .....                      | 24 |
| 4.3.1. Renk fonksiyonu tanımlaması .....                                | 24 |
| 4.3.2. K-Ortalamlar ile fiyat segmentasyonun oluşturulması .....        | 25 |
| 4.3.3. Ürünlere ait marka fonksiyonunun tanımlanması.....               | 26 |
| 4.4. Text Similarity Algoritması .....                                  | 26 |
| 4.5. TDA (Tekil Değer Ayrışımı) Algoritmasının Uygulanması .....        | 29 |
| 4.6. TDA ve Segmentlere Göre Tavsiye Ürünlerin Belirlenmesi .....       | 31 |
| 4.7. GTM ile Google Analytics’e İlgili Alan Bilgilerinin Aktarımı ..... | 32 |
| 4.8. Çıktılarının Web Sitesi ve Mobil Uygulamaya Aktarılması .....      | 33 |

## BÖLÜM 5.

|                                                                          |    |
|--------------------------------------------------------------------------|----|
| ARAŞTIRMA BULGULARI .....                                                | 35 |
| 5.1. Kullanıcı Ekranlarının Karşılaştırılması .....                      | 35 |
| 5.2. Öneri Sistemi Performansının Google Analytics ile Kıyaslanması .... | 35 |
| 5.3. Öneri Sisteminin Satışa Olan Etkisinin Yüzdesel Değişimi .....      | 37 |

## BÖLÜM 6.

TARTIŞMA VE SONUÇ ..... 41

KAYNAKLAR ..... 43

ÖZGEÇMİŞ ..... 47



## SİMGELER VE KISALTMALAR LİSTESİ

|     |                                                               |
|-----|---------------------------------------------------------------|
| GTM | : Google Tag Manager                                          |
| İF  | : İşbirlikçi Filtreleme                                       |
| KEA | : Küçük ev aletleri                                           |
| RFM | : Recency (güncellik), Frequency (sıklık), Monetary (parasal) |
| SSS | : Satış sonrası servis                                        |
| TDA | : Tekil Değer Ayrışımı                                        |

## ŞEKİLLER LİSTESİ

|                                                                                     |    |
|-------------------------------------------------------------------------------------|----|
| Şekil 2.1. Öneri Sistemi Yöntemleri.....                                            | 9  |
| Şekil 4.1. Uygulama Akış Şeması .....                                               | 20 |
| Şekil 4.2. Kullanılacak verilere ait Dax sorgusu ve Python kodlarıyla çekilmesi ... | 21 |
| Şekil 4.3. Renk bilgisi fonksiyon tanımlama .....                                   | 25 |
| Şekil 4.4. K-Ortalamalar algoritması ile fiyat segmentasyonunun yapılması .....     | 26 |
| Şekil 4.5. Text Similarity uygulaması için liste oluşturulması .....                | 27 |
| Şekil 4.6. Metin Benzerliği Algoritmasının Uygulanması .....                        | 28 |
| Şekil 4.7. TDA Algoritmasının Uygulanması .....                                     | 30 |
| Şekil 4.8. TDA ve segmentlere göre ürün önerileri alanının doldurulması.....        | 31 |
| Şekil 4.9. Tava Ürün grubuna yönelik düzenleme örneği .....                         | 32 |
| Şekil 4.10. Tamamlayıcı ürün önerileri alanı görüntülenme Tag konfigürasyonu .      | 33 |
| Şekil 4.11. Tamamlayıcı ürün önerileri alanı tıklanma Tag konfigürasyonu .....      | 33 |
| Şekil 4.12. Ürün ID'lerinin liste formatına çevrilmesi .....                        | 34 |
| Şekil 5.1. Mevcut sistemin ürün önerileri .....                                     | 35 |
| Şekil 5.2. Geliştirilen sistemin ürün önerileri (fiyat segmentasyonu odaklı) .....  | 36 |
| Şekil 5.3. Geliştirilen sistemin ürün önerileri (ürün ilişkileri odaklı) .....      | 36 |
| Şekil 5.4. Geliştirilen sistemin ürün önerileri (ürün boyutları odaklı) .....       | 37 |
| Şekil 5.5. Google Analytics görüntülenme ve tıklanma sayıları karşılaştırması ..... | 38 |
| Şekil 5.6. “Clicked” sayılarının bir önceki aya göre günlük değişimi .....          | 38 |

## TABLÖLAR LİSTESİ

|                                                           |    |
|-----------------------------------------------------------|----|
| Tablo 2.1. Öneri sistemleri literatür taraması .....      | 4  |
| Tablo 4.1. Veri yanımlamaları .....                       | 22 |
| Tablo 4.2. Ürün önerilerinin ID olarak düzenlenmesi ..... | 34 |



## ÖZET

Anahtar kelimeler: Ürün öneri sistemi, Metin Benzerliği, Tekil Değer Ayrışımı, e-ticaret

Günümüzde şirketlerin dijital dönüşüm kapsamında yaptığı çalışmalar özellikle pandemi sonrası tüketiciden gelen talebin artmasıyla giderek hız kazanmıştır. Bu bağlamda, E-Ticaret alanında web site ve mobil uygulamalarda müşteriye en uygun ürün önerilerinin sunulması, müşteri ihtiyaçlarının karşılanması ve şirketlerin satış hedeflerinin gerçekleştirilmesi için ürün öneri sistemlerine ihtiyaç duyulmuştur.

Bu alanda yapılan çalışmalar, ürün tipi ve çeşitliliği değişkenlik gösterdiği için çoğu zaman mantıksız ve yanlış ürün önerilerinin tüketicilere sunulmasına yol açtığı, farklı sitelerin ürün önerileri incelendiğinde açıkça görülmüştür. Bu çalışmada, şirketin ürün ve veri yapısına göre en uygun şekilde veri manipülasyonun gerçekleştirilmesi, özelleştirilmiş fonksiyonların yazılması, Metin Benzerliği ve Tekil Değer Ayrışımı algoritmasına dahil olarak en uygun tamamlayıcı ürünlerin gösterilmesi sağlanmıştır. En uygun algoritmanın geliştirilerek yazılımı tamamlanmıştır. Elde edilen ürün önerileri web sitesi ve mobil uygulamada canlıya alınmıştır. Sonuçları Google Analytics üzerinden ve Python kodlarıyla ayrı ayrı gözlemlenmiştir.

Yapılan çalışma sonucunda geliştirilen ürün öneri sisteminin mevcut sisteme kıyasla set halinde satılan ürünlerde %7,62, tekli olarak satılan ürünlerde ise %11,2 oranında daha iyi performans gösterdiği gözlemlenmiştir. Ayrıca ilgili alanın web site ve mobil uygulamada görüntülenme sayısı %7,17, tıklanma sayısı %28,93 artış göstermiştir. Ürün önerilerinin mevcut duruma kıyasla daha mantıklı ve tamamlayıcı olarak daha ilişkili ürün önerileri sunduğu tespit edilmiştir ve ilgili alanlar farklı zamanlarda incelenerek gözlemlenmiştir.

# **Establishing of A Product Recommendation System Based on Text Similarity and Singular Value Decomposition Algorithm in A Retail Company**

## **SUMMARY**

Keywords: Product recommendation engine, Text Similarity, Singular Value Decomposition, e-commerce

Due to rising client demand following the pandemic, organizations' initiatives in the field of digital transformation are gaining pace nowadays. Product recommendation systems were necessary for this context to deliver the most relevant product choices to clients, meet their expectations, and accomplish the businesses' sales targets through the e-Commerce website and mobile applications.

Examining the product suggestions of numerous websites reveals that study on this subject frequently results in the presentation of illogical and erroneous product recommendations to visitors, given the variety and type of things accessible. This study looked into data manipulation, building custom functions, Text Similarity, and Singular Value Decomposition. The algorithm displays the best complementary commodities. The software was completed by developing the best algorithm. The product proposals were brought to life through the use of a website and a mobile application. Individual observations were made using Google Analytics and Python scripts.

The study found that the created product recommendation system beat the current system by 7.62 per cent in items sold as a group and by 11.2 per cent in products sold separately. Furthermore, the number of views on the website and mobile application for the relevant area climbed by 7.17 per cent, and the number of clicks increased by 28.93 per cent. It has been decided that the product suggestions provide more logical and complementary product recommendations than the existing situation, and the relevant regions have been monitored and reviewed at various times.

## BÖLÜM 1. GİRİŞ

Günümüzde tüketim ihtiyaçlarının artmasına yönelik olarak perakende sektöründe faaliyet gösteren firmalar, ürün çeşitliliğini ve faaliyet alanlarını genişleterek müşteriye daha çok alternatif sunmak ve satış hedeflerini yükseltmek istemektedir. Özellikle pandemi sonrası tüketici davranışlarının dünyada e-ticaret alanına yönelmesiyle birlikte firmalar gelişen teknolojik gelişmeler neticesinde dijital dönüşüm kapsamında yatırımlarına ağırlık vermektedir. Bu alanda en öne çıkan konulardan birini de ürün öneri sistemleri oluşturmaktadır.

İnternetin her geçen gün yaygınlaşmasıyla kullanıcılara ait verilerden oluşan büyük veri ambarlarından değerli bilgilerin çıkarılması için analiz çalışmaları yapılmış ve öneri sistemleri 1990'lı yılların ortalarından itibaren veri teknolojileri içerisinde farklı bir başlık olarak incelenmeye başlanmıştır [1].

Öneri sistemleri e-ticaret sitelerinden satın alınacak bir ürün, sosyal medya kanallarında izlenecek bir video, film, dizi, haber, kitap veya müzik gibi çok farklı alanlarda kullanıcıların ihtiyaçlarını karşılamak için zaman ve fayda sağlamayı hedefleyen, belirlenmiş kullanıcıların en çok ilgisini çekebilecek nesnelerin önerilmesini sağlayan yazılım araçları ve teknikleridir [2]. Bu bağlamda öneri sistemlerinin en yaygın olarak kullanıldığı ve üzerinde yoğun çalışmaların yapıldığı Netflix [3], Amazon [4], Youtube [5] ve Spotify [6] gibi platformlar öne çıkmaktadır. Bu platformlar çok sayıda kullanıcıdan beslenen büyük veri yapılarıyla çalıştığından dolayı başarı oranının daha yüksek olarak gerçekleştiği bilinmektedir. Ürün satışı yapmak isteyen çoğu perakende firmalarında ise veri büyüklüğünün az olması, ürün çeşitliliğinin artması ve firma yapılarından kaynaklı olarak ürün öneri sistemlerinin mantıksız ve ilgili olmayan ürün önerileri sunduğu gözlemlenmiştir.

Öneri sistemleri siteye trafik çekmek, müşterilere ilgili ve tamamlayıcı ürünlerin gösterimini gerçekleştirmek, satışların ve ürün görüntülenme sayılarının artmasını sağlamak, kullanıcılara en uygun kişiselleştirilmiş seçenekleri sunarak karar vermeyi kolaylaştırmak gibi avantajları sunmaktadır. Buna karşın ürün veya hizmet önerilerinin yapılırken bulunduğu sektör ve ilgi alanına göre değişen dinamiklere uyum sağlayamama, soğuk başlangıç, veri yetersizliği ve seyrek olma durumu, yeterli sürede çalışamama gibi dezavantajları da bulunmaktadır.

Bu kapsamda öneri sistemleri ile ilgili yapılan çalışmalar incelenmiştir. Genel olarak farklı algoritmaların performans değerleri karşılaştırılarak birinin seçilmesi ve bu doğrultuda ürün önerilerinin gerçekleştirildiği gözlemlenmiştir. Uygulandığı sektör, algoritma tercihleri ve veri yapılarının uygunluğu göz önünde bulundurularak hangi yöntemlerin daha iyi sonuç verdiği araştırılmıştır. Özellikle ürün çeşitliliğinin son derece fazla ve değişken olduğu züccaciye sektörüne ait bir çalışmaya rastlanmamıştır. Metin benzerliği ve Tekil Değer Ayrışımı algoritması yaklaşımının birlikte kullanıldığı herhangi bir ürün öneri sistemi de literatürde yer almamaktadır. Bu bağlamda tercih edilen metotlar ve uygulandığı sektör bakımından, geliştirilen ürün öneri sisteminin özgünlüğü ortaya konulmuştur.

Geliştirilen sistemde algoritmaların firmaya ait mevcut veri yapılarına uygunluğu göz önünde bulundurulmuştur ve ürün önerileri yapılırken öncelikli olarak metin benzerliği ve tekil değer ayrışımına ek olarak k-ortalamlar, özelleştirilmiş fonksiyonlar gibi çok yönlü bir ürün önerileri geliştirilmiştir

Yapılan çalışmanın amacı, perakende sektöründe faaliyet gösteren bir firmaya ait mevcut durumda bulunan ürün öneri sisteminde hatalı, eksik ve ilgili olmayan ürün önerilerinin gözlemlenmesi sonucu geliştirilerek firmanın satışlarını artırmak, kullanıcılara en uygun tamamlayıcı ürünleri sunmak ve buna bağlı olarak müşteri memnuniyetini sağlamak, geri bildirimlere göre düzenlenebilir esnek bir yapı kazandırılarak uygulamanın sürdürülebilirliği gerçekleştirmektir. Perakende sektöründe önemli bir yer tutan züccaciye alanında yapılmış olan ilk çalışma niteliğini taşıyarak bu alanda yapılacak çalışmalara öncülük etmesi hedeflenmiştir.

Bu çalışmada, züccaciye, ev tekstili, elektrikli ev aletleri ve dekoratif kategorilerinde hizmet veren bir perakende firması için ürün öneri sistemi geliştirilmiştir. Gerçek veri üzerinde çalışılarak veri ön işleme işlemleri tamamlanmıştır. Ürün öneri sisteminde kullanılacak algoritmalara uygun format elde edilmesi için gerekli tanımlamalar yapılmıştır. Elde edilen ürün önerilerini GTM ile uygulanabilirliği sağlanmıştır. Program çıktıları mevcut durumda bulunan tamamlayıcı ürün öneri sistemine göre karşılaştırılması yapılarak firmaya satış bakımından etkisi test edilmiştir. Tekli satılan ve set halinde satılan ürünler için ürün öneri sisteminin performansı ayrı olarak test edilmiştir. Kullanıcı ekranlarında nasıl bir değişim olduğu gözlemlenmiştir ve firmaya ait mobil uygulama, web sitesi gibi platformlarda tıklanma ve görüntülenme sayılarındaki karşılaştırma sonuçları gözlemlenmiştir.

## BÖLÜM 2. KAYNAK ARAŞTIRMASI

Ürün öneri sistemlerine dair literatür incelemesi Tablo 2.1.'de gösterildiği gibi zaman sıralamasına göre çalışma isimleri, kullanılan yöntemleri ve çalışma alanları incelenerek verilmiştir.

Tablo 2.1. Öneri sistemleri literatür taraması

| Yıl  | Yazar                 | Çalışma Türü | Çalışma Adı                                                                                              | Kullanılan Yöntem                                            | Çalışma Alanı                  |
|------|-----------------------|--------------|----------------------------------------------------------------------------------------------------------|--------------------------------------------------------------|--------------------------------|
| 2014 | Choi, Kim (2019)      | Makale       | An auxiliary recommendation system for repetitively purchasing items in E-commerce                       | Kullanıcı Tabanlı Filtreleme, Ürün Tabanlı Filtreleme        | E-ticaret, Tekrarlı Ürünler    |
| 2017 | Alharthi ve ark. 2017 | Makale       | A survey of book recommender systems                                                                     | İF, İçerik Tabanlı Filtreleme                                | Kitap Öneri                    |
| 2018 | Yıldız (2018)         | Tez          | Müşteri Segmentasyonu Odaklı Bütünleşik Bir Ürün Öneri Sistemi: Moda Perakendesi Sektöründe Bir Uygulama | RFM, Sepet Analizi                                           | Moda                           |
| 2018 | Batmaz (2018)         | Tez          | Yükseköğretimde Öneri Sistemlerine Dayalı Ders Seçme Modeli                                              | NASA-TLX                                                     | Ders Seçimi                    |
| 2019 | Abdullah (2019)       | Tez          | E-ticaret için ürün tavsiye sistem geliştirmesi                                                          | İF, Apriori, Jaccard, Kosinüs Benzerliği, Pearson Benzerliği | İnşaat                         |
| 2019 | Güler (2019)          | Tez          | Öneri Sistemleri ve E-Ticarete Öneri Sistemlerinin Kullanımı                                             | İçerik Tabanlı Filtreleme, İF                                | Öneri Sistemleri Karşılaştırma |

Tablo 2.1. (Devam)

| Yıl  | Yazar                           | Çalışma Türü | Çalışma Adı                                                                               | Kullanılan Yöntem                           | Çalışma Alanı |
|------|---------------------------------|--------------|-------------------------------------------------------------------------------------------|---------------------------------------------|---------------|
| 2020 | Erdoğan (2020)                  | Tez          | Banka Verileriyle Ürün Öneri Sistemi                                                      | Apriori, İF                                 | Banka         |
| 2021 | Uçar (2021)                     | Tez          | E-Ticaret Siteleri için Bulanık Mantık ve Makine Öğrenmesi Tabanlı bir Öneri Sistemi      | Bulanık Mantık, K-Ortalamalar, C4.5         | Kitap         |
| 2021 | Dundar, Görgülü Kalkisim (2021) | Makale       | Kıyafet Öneri Sistemi için Giyim Metaverilerine Dayalı Temsil Öğrenimi                    | Word2Vec, Doc2Vec                           | Giyim         |
| 2021 | Yalçın (2021)                   | Makale       | İşbirlikçi Filtreleme Algoritmalarının Çok-Beğenilen Ürünlere Yönelik Yanlılığı           | UserkNN, ItemkNN, TDA, TDA++, Co-Clustering | İF'nin etkisi |
| 2021 | Bozkurt ve Acı (2021)           | Makale       | Öneri Algoritmalarının Film Önerme Problemi Üzerinde Karşılaştırılması: MovieLens Örneği. | TDA, İF, Kosinüs Benzerliği                 | Film          |

Bozkurt, Acı (2021) MovieLens veri seti kullanılarak film önerilerinin tekil değer ayrışımı, işbirlikçi filtreleme ve kosinüs benzerliği yöntemleri kullanarak 3 farklı metodun karşılaştırılması yapılmıştır. Kullanıcılara öneriler hakkında düşünceler sorulduğunda en iyi sonuçları tekil değer ayrışımı ile elde edilirken yapılan çalışma sonucunda işbirlikçi filtrelemenin daha başarılı olduğu gözlemlenmiştir [7]. İncelenen bu çalışma neticesinde TDA'nın avantajları açıkça görülmüştür ve geliştirilen ürün öneri sistemi için korelasyon değerlerinin önemi anlaşılmıştır.

Yalçın (2021), Ürün profilleri ayrı bir yaklaşımla, beğenilme dereceleriyle incelenmiştir. İşbirlikçi filtreleme algoritmalarının en çok tercih edilen ürünlere yönelik yanlılık durumu araştırılmıştır. 3 farklı kategoriden 9 önemli işbirlikçi filtreleme algoritmasını kullanarak iki reel dünya veri kümesi üzerinde farklı deneyler gerçekleştirilmiş. Yapılan deneyler sonucunda neredeyse bütün algoritmaların çok

beğenilen ürünlere yanlılığın yüksek olduğunu ve kaliteli öneriler üretmede TDA, TDA++ gibi matris çarpanlarına ayırma tabanlı algoritmaların başarılı olduğu tespit edilmiştir [8]. İncelenen bu çalışmada, Tekil Değer Ayrışımının algoritmalar arasında en iyi ürün önerileri sunduğu gözlemlenmiştir. Geliştirilen ürün öneri sisteminde müşteriler satın aldığı ürünü beğenmiştir kabulü ile ilerlenmiştir. Bu açıdan yapılan çalışmaya katkıda bulunmuştur.

Dundar, Görgülü Kalkisim (2021), Çalışmalarında giyim metaverilerine dayalı temsil öğrenimi ile kıyafet öneri sistemi geliştirmiştir. Rastgele yürüyüş ve Skipgram yöntemlerini kullanarak ürünler arası benzerliği modelleyen metaveriler üzerinden yeni bir yaklaşımda bulunmuştur. Yöntemin sınıflandırma başarı performansını ölçmek için Rastgele Orman (RF) kullanılmış olup F-ölçüm (Mikro) %76,72 bulunmuştur. Karşılaştırma için kullanılan yöntemlerde F-ölçüm (Mikro) değerleri Word2Vec için %62,92 ve Doc2Vec için %60,74 olarak sınıflandırma başarısı göstermiştir [9]. Perakende sektöründe giyim alanında yapılan bu çalışma incelenerek özellikle ev tekstili kategorisinde yapılacak ürün önerilerine katkı sağlanmıştır.

Choi, Kim (2014) Çalışmalarında e-ticaret siteleri içinde müşterilere ait tekrarlı satın alma işlemlerine yönelik bir öneri sistemi geliştirmişlerdir. Kullanıcılardan elde edilen veriler geliştirilen öneri sisteminde kullanılmıştır. Veri setinde yaklaşık 5000 kullanıcı ve 4000 ürün kullanılarak öneri sisteminde performans analizleri yapılmıştır. Sonuç olarak kullanıcı tabanlı filtreleme tekniğinin ürün tabanlı filtreleme tekniğinden daha iyi performans gösterdiği tespit edilmiştir [10]. Çalışma incelenerek ilk olarak kullanıcı tabanlı ilerlenmesi planlanmıştır. Burada kullanıcılara ait büyük bir veri üzerinden hesaplamalarda eksiklikler tespit edildiği için ürün içeriklerini baz alarak ilerleme kararı alınmıştır.

Alharthi ve ark. 2017 yılında kitap öneri sistemlerine ilişkin bir literatür çalışması yapmıştır. Literatürde kullanılan veri setleri ile öneri sistemleri yöntemlerini araştırmıştır. Yapılan çalışmada işbirlikçi filtreleme ve içerik tabanlı yöntemler ön plana çıkmıştır. INEX Book Track, Book-Crossing veriseti, LibraryThing, LitRec, Amazon reviews, Goodreads ve Project Gutenberg gibi veri setlerinin kitap öneri

sistemi için ön plana çıktığını tespit etmiş bu veriler hakkında bilgilere değinilmiştir [11].

Uçar (2021), E-ticaret sitelerine yönelik makine öğrenmesi ve bulanık tabanlı bir öneri sistemi geliştirmiştir. Farklı sitelerde görüntülenen ve satın alınan kitap verileri ile desteksiz öğrenme yöntemiyle kümeleme yapılmıştır. Alt kümeleri K-Ortalamalar algoritması kullanarak kümeleme işlemlerinden elde etmiştir. Ürün önerilerinin geliştirmesinde bulanık model uygulamanın avantajlarını ve bu modelin kurallarının belirlenmesinde karar ağacı metodunun önemini göstermiştir [12].

Yıldız (2018), Moda perakendesi alanında müşteri segmentasyonu temelli bir ürün öneri sistemi geliştirmiştir. RFM analizi ile edilen skorlar ile müşteri bilgileri birleştirilerek k-ortalamlar tekniğiyle kümeleme işlemi yapılmıştır. Elde edilen 3 kümeye birliktelik analizi yapılarak hangi kategorileri birlikte aldığı tespit edilmiş ve ürün önerileri yapılmıştır. Geliştirilen sistemin mevcut sisteme göre hem f1 hem de ortalama satış değeri bakımından pozitif etkisi gözlemlenmiştir [13].

Çalışmalar göstermiştir ki fiyat segmentasyonu oluşturmada k-ortalamlar tekniğinin kullanılarak kümeleme işlemlerin yapılması öneri sisteminde amaçlanan sonuçlara ulaşmak için en uygun yöntemdir [12-13].

Erdoğan (2018), Bankacılık sektöründe müşteri deneyimini artırmaya yönelik hibrit bir ürün öneri sistemi geliştirmiştir. Bir bankadaki müşteri verilerini kullanarak geliştirdiği hibrit sistemin her bir müşteri için test setindeki toplam doğruluk oranını %67,8 bulurken WEKA Apriori algoritmasında %37,3 olarak hesaplamıştır. Ayrıca oluşturduğu hibrit sistemde %71,3 hassasiyet, %71,2 f1-score oranı ve %77,4 hatırlatma oranı yakalarken WEKA Apriori algoritmasına bu değerleri %67,5 hassasiyet, %64,5 f1-score oranı ve 62,6 hatırlatma oranı olarak daha düşük bulmuştur [14].

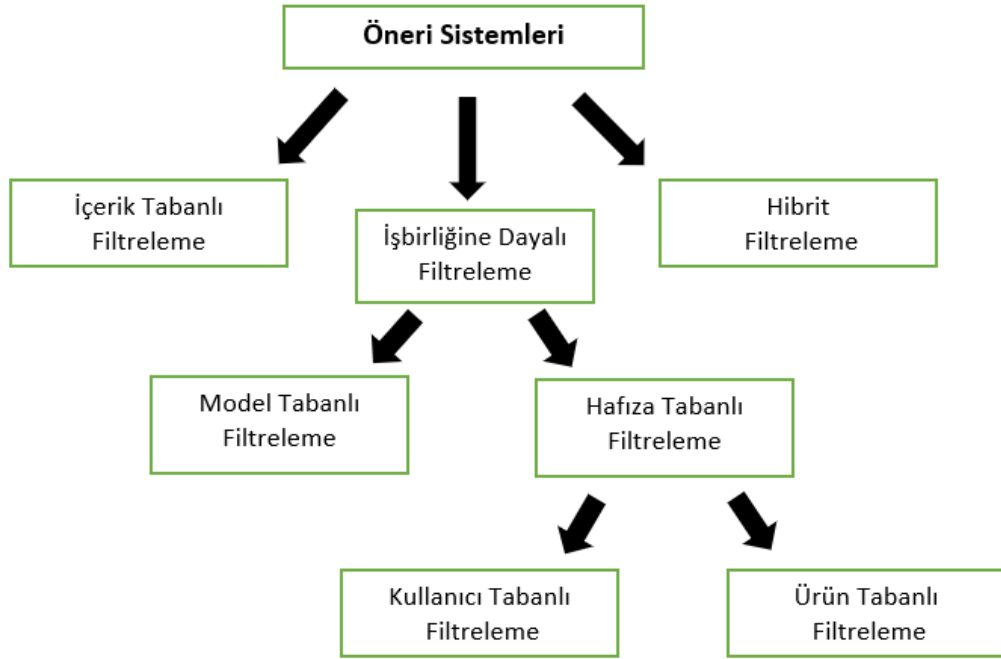
Abdullah (2019), yaptığı tez çalışmasında e-ticaret için ürün öneri sistemi geliştirmiştir. Veri setinde 1023 ürün ve yarım milyon satış bilgilerini içermektedir.

İşbirlikçi filtreleme, FP-Büyüme, Apriori algoritması ve sık örüntü madenciliği üzerine çalışmalar yapmıştır. Çalışmasında benzerlik hesaplama yöntemlerinden Jaccard Benzerliği, Kosinüs ve Pearson kullanmıştır. Sonuçta en iyi performans veren yöntemlerin Jaccard benzerliği ve FPBüyüme algoritmasının olduğu gözlemlenmiştir [15].

Güler (2019), Çalışmasında öneri sistemlerine genel bir bakış açısıyla yaklaşmıştır. Öneri sistemlerinde kullanılan yöntemler, karşılaşılan problemler ve bu sistemlerin uygulamaları ve karşılaştırmaları üzerinde durmuştur. Öneri sistemleri ile ilgili araştırma yapan kullanıcılar için etkileşimlerde ilk 10'a girme oranlarını hibrit yöntem %53, içerik bazlı yöntem %52, işbirlikçi yöntem %45 olarak tespit ederken ilk 5'e girme oranlarını hibrit yöntem %43, içerik bazlı yöntem 41 ve işbirlikçi yöntem için %33 olarak hesaplamıştır [16]. Bu çalışmanın incelenmesi neticesinde hibrit ve işbirlikçi yöntemlerin birbirine çok yakın sonuç değerlerine sahip olduğu gözlenmiştir ve bir sonraki çalışmada hibrit yöntem ile de karşılaştırmalı sonuç değerlerine ulaşılması planlanmıştır.

Öneri sistemleri, alınması düşünülen ürünler için aynı veya benzer ürünleri önceden almış başka müşterilerin yorum, değerlendirme ve oylarının da yardımıyla alternatif seçenekler veya tamamlayıcı başka ürünler önererek yapılacak alışverişi çok kolaylaştırmaktadır. Bu sayede zamandan tasarruf edilmesi, ihtiyaca daha yakın ürünlerin incelenmesi sağlanmış olur [16].

Ürün veya içerik öneri sistemlerindeki temel hedef, kullanıcıların beğenisini öngörüp onlara bu beğenileri neticesinde öneriler sunmaktır. Öneri sistemlerinde öneri sunulacak alana, elde edilen veri yapısına ve hedefine göre tercih edilen metotlar birkaç ana temel üzerinde ilerlemektedir. Bu metotların ve algoritmaların türleri ve birbirleri ile olan ilişkilerini gösteren, öneri sistemleri yapısı geçmiş çalışmalardan derlenerek akış diyagramı halinde Şekil 2.1.'de verilmiştir.



Şekil 2.1. Öneri Sistemi Yöntemleri

Yapılan çalışmada işbirliğine dayalı filtreleme yöntemleri içerisinde bulunan kullanıcı tabanlı filtreleme yöntemi Tekil Değer Ayrışımı (TDA) kullanılarak ürün öneri tablosunun doğrudan etkileyen korelasyon değerleri hesaplanmıştır. Literatür incelemesi yapıldığında ürün tabanlı öneri sistemlerinde özellikle TDA'nın en iyi performans değerlerine sahip olduğu açıkça gözlemlenmiştir. TDA'nın geleneksel işbirlikçi algoritmalarından daha iyi sonuçlar verdiği, veri matrisindeki kısıt seyreklik problemi, kümeleme algoritmaları için benzerliklerin hesaplanmasında ve uygun komşulukların belirlenmesindeki sorunların giderilmesinde uygulanan algoritmalar arasında TDA'nın en iyi sonuçları verdiği tespit edilmiştir [17].

## BÖLÜM 3. MATERYAL VE YÖNTEM

### 3.1. Metin Benzerliği Algoritmaları

Metin benzerliği algoritmaları yapılan işlemlerin niteliklerine göre sınıflandırılmaktadır.

Belirteç Bazlı Benzerlik Sınıfı'nda beklenen girdi, tam dizelerden ziyade bir dizi ayıraçtır. Amaç mesafe ölçülecek her 2 sette de benzer ayıraçları bulmaktır. Ortak belirteç sayısı arttıkça, setler arasındaki benzerlik de artar. Kosinüs Benzerliği, Monge Elkan Uzaklığı ve Bag Uzaklığı, Jaccard Benzerlik Katsayısı ve Sorensen Dice Benzerlik Katsayısı ve Tversky İndeksi bu kategori içerisinde yer almaktadır.

Uzaklık Düzenleme Bazlı Benzerlik Sınıfı'nda ise Levenshtein, Jaro-Winkler, Smith Waterman, Hamming ve Needleman Wunsch Mesafeleri bulunmaktadır. Amaç bir dizeyi diğer dizeye dönüştürmek için değiştirme, ekleme, silme vb. gerekli işlem sayısının hesaplanmasıdır. Yapılan işlem sayısı arttıkça iki dize arasındaki benzerlik daha az gerçekleşmektedir

Diziliş Tabanlı Benzerlik Sınıfı'nda Ratcliff/Obershelp Benzerliği yer almaktadır. İki dize arasındaki ortak alt dizelerin bir faktörüdür. Amaç her 2 dizede bulunan en uzun diziyi bulmaktır. Bu dizelerden fazla bulunması durumunda benzerlik puanı daha yüksek olmaktadır [18].

#### 3.1.1. Kosinüs benzerliği

Kosinüs Benzerliği, bir iç çözüm uzayının iki vektörü arasındaki benzerliği ölçmektedir. İki vektörün kabaca aynı yönü gösterip göstermediğini tespit

edilmektedir. İki vektör arasındaki açının kosinüsü ile ölçülmektedir. Genel olarak metin analizinde belge benzerliğini ölçmek için kullanılmaktadır. Bir belge, her biri belirli anahtar kelime gibi bir kelimenin ya da belgede bulunan cümlelerin frekansını kayıt eden binlerce nitelik ile temsil edilebilmektedir. Buna bağlı olarak, her belge, terim frekansı vektörü olarak isimlendirilen bir nesneyle temsil edilmektedir. Terim frekans vektörleri tipik olarak oldukça uzun ve çok sayıda 0 değerine sahip olduğu için seyrek bir yapıdadır. Bu yapıları kullanan uygulamalar arasında bilgi erişimi, gen özellik haritalaması, metin belgesi kümelenmesi ve biyolojik taksonomi bulunmaktadır. Kosinüs benzerliği, belgeleri karşılaştırmak ya da belirlenmiş bir sorgu kelimesi vektörüne göre bir belge sıralamasını vermek amacıyla uygulanabilecek bir benzerlik ölçüsüdür. Karşılaştırma için  $a$  ve  $b$  iki vektör olsun. Kosinüs ölçüsü benzerlik fonksiyonu olarak kullanılarak aşağıda (Denklem 3.1) gösterildiği gibi formülize edilmektedir [19].

$$sim(a, b) = \frac{a \cdot b}{\|a\| \|b\|} \quad (3.1)$$

### 3.1.2. Monge Elkan mesafesi

Monge-Elkan, her bir belirteç için en iyi eşleşmeyi bulan, belirteçlere ve iç benzerlik işlevine dayanan genel bir metin dizesi karşılaştırma yöntemidir. Bu bağlamda belirteçler, çoğu insan dilinde olduğu gibi kelimelere bölünmüş karakter dizileridir. Algoritma, tipik metin alanlarının özyinelemeli yapısını kullanmaktadır. İki dize, aynı atomik dize ise veya biri diğerini kısalttıysa, derece 1 ile eşleşir. Aksi takdirde, eşleşme derecesi 0'dır [20]. Aşağıda bu durum (Denklem 3.2) formülize edilmiştir.

$$sim_{MongeElkan}(A, B) = \frac{1}{|A|} \sum_{i=1}^{|A|} \{ \max_{j=1}^{|B|} sim'(a_i, b_j) \} \quad (3.2)$$

### 3.1.3. Bag mesafesi

Bag mesafe algoritması,  $x$  dizesindeki ve  $y$  dizesindeki bir karakter ile benzer bulunmayan bir şekilde eşleştirilemeyen her bir karakteri numaralandırmaktadır ve bu

işlemin tersi de geçerli olmaktadır. İki değerden maksimum olanı Bag mesafesi olarak kabul edilmektedir. Bag mesafesi Levenshtein mesafesine bir üst sınır koymaktadır ve bunun hızlı bir yaklaşımı olarak önerilmiştir [21].

#### 3.1.4. Jaccard benzerlik katsayısı

Jaccard benzerlik katsayısı, örnek kümelerin benzerliğini ve çeşitliliğini ölçümlemek amacıyla kullanılan bir istatistik yöntemidir. Jaccard katsayısı sonlu örnek kümeleri arasındaki benzerlik durumunu ölçmektedir ve kesişim boyutunun örnek kümelerin birleşiminin boyutuna bölünmesiyle tanımlanmaktadır. Bu yöntemle ait formülizasyon (Denklem 3.3 ve Denklem 3.4) aşağıda gösterilmiştir:

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|} = \frac{|A \cap B|}{|A| + |B| - |A \cap B|} \quad (3.3)$$

Sonucun 1'e eşit olması iki kümenin de eş olduğu anlamına gelmektedir. Jaccard Mesafesi, 1'den Jaccard benzerlik katsayısının çıkarılması ile bulunmaktadır [22].

$$dj(A, B) = 1 - J(A, B) = \frac{|A \cup B| - |A \cap B|}{|A \cup B|} \quad (3.4)$$

#### 3.1.5. Sorensen Dice benzerlik katsayısı

Sorensen – Dice katsayısı, iki örneğin benzerliğini ölçmek için kullanılan bir istatistiktir. Bağımsız olarak sırasıyla 1948 ve 1945'te yayınlanan botanikçiler Thorvald Sørensen ve Lee Raymond Dice tarafından geliştirilmiştir. Aşağıda (Denklem 3.5) formülize edilmiştir [23].

$$DSC = \frac{2|X \cap Y|}{|X| + |Y|} \quad (3.5)$$

### 3.1.6. Tversky indeksi

İsmi Amos Tversky'den alan Tversky indeksi, bir dizeler üzerindeki asimetrik bir benzerlik ölçüsüdür. Sorensen-Dice ve Jaccard benzerlik katsayısının bir genellemesi olarak görülebilmektedir. Formülize edilmiş (Denklem 3.6) durumu aşağıda gösterilmiştir:

$$S(X, Y) = \frac{|X \cap Y|}{|X \cap Y| + \alpha|X - Y| + \beta|Y - X|} \quad (3.6)$$

X ve Y kümeleri için Tversky indeksi, 0 ile 1 arasında bir sayıdır. Burada  $\alpha, \beta \geq 0$  Tversky indeksinin parametreleridir. Formülde  $\alpha = \beta = 1$  ise Jaccard katsayısı;  $\alpha = \beta = 0.5$  ise Sorensen-Dice katsayısı elde edilir. X'i prototip ve Y'yi varyant olarak düşünür isek,  $\alpha$  prototipin ağırlığına karşılık gelir iken,  $\beta$  varyantın ağırlığına karşılık gelmektedir Tversky indeksi  $\alpha + \beta = 1$  olarak formülize edilebilmektedir [24].

### 3.1.7. Levenshtein mesafesi

Levenshtein uzaklığı iki dize arasındaki benzerliğin bir ölçüsüdür. Algoritma, a dizesini b dizesine dönüştürmek için gerekli minimum değişiklik sayısı olarak tanımlanır. Bu, a dizesine bir karakter ekleyerek, silerek veya değiştirilerek yapılır. Levenshtein mesafesi ne kadar küçük ise, dizeler de bir o kadar benzemektedir [25]. Formülasyonu (Denklem 3.7) aşağıda gösterildiği gibidir:

$$\begin{aligned} & \max(i, j) \text{ lev}_{a,b}(i, j) \quad , \text{ eğer } \min(i, j) = 0 \\ & \text{lev}(a, b) = \min \begin{cases} \text{lev}_{a,b}(i-1, j) + 1 \\ \text{lev}_{a,b}(i, j-1) + 1 \\ \text{lev}_{a,b}(i-1, j-1) + 1_{(a_i \neq b_j)} \end{cases} \quad , \text{ diğ er} \end{aligned} \quad (3.7)$$

Levenshtein uzaklığının hesaplanması için, her kelimedenden geçmeli ve en az + 1 kullanılmalıdır.

(i-1, j): sol kutu anlamına gelir (silme)

(i, j-1): üst kutu anlamına gelir (ekleme)

(i-1, j-1): sol üst kutu anlamına gelir (yer değiştirme) [26].

Yapılan tez çalışmasında metin benzerliği algoritmaları içerisinde bulunan uzaklık düzenleme bazlı benzerlik sınıfına ait Levenshtein Mesafesi kullanılmıştır. Bu algoritmanın kullanılmasındaki amaç kullanılacak veride çok sayıda yanlış girilmiş ve düzenlenmesi gereken dizelerinin bulunması, ürün ismi içerisinde yer alan, istenilen nitelik olan model ailelerini ön plana çıkarabilmesi, uzman görüşleri sonrası bazı özellikler için ceza puanı uygulayarak uygun ürün önerilerin gösteriminde yüksek fayda sağlamasıdır.

### 3.1.8. Needleman Wunsch mesafesi

Saul B. Needleman ve Christian D. Wunsch, Needleman-Wunsch algoritması olarak da isimlendirilen, sekans hizalaması için sezgisel bir homoloji algoritmasını 1970 yılında önermiştir. Hesaplama adımlarını gerektiren (ve hizalanan iki dizinin uzunlukları olan) küresel bir hizalama algoritmasıdır. Küresel hizalamayı göstermek amacı ile bir matrisin yinelemeli hesaplamasını kullanmaktadır. Karakter dizisinin sırasını dikkate aldığından dolayı dize kıyaslaması yapmak için en iyisidir. Diziler arasında en uygun hizalama çözümünü bulmaktadır. Olumsuz yönlerinden biri diğer yöntemlere göre hizalamayı yapmanın daha fazla zaman alması, bundan kaynaklı olarak da performansının azalmasıdır [27].

### 3.1.9. Jaro – Winkler mesafesi

Jaro algoritması veri bağlantı sistemlerinde, ad eşleşmesi için genel olarak kullanılmaktadır. Yer değiştirme, silme ve ekleme işlemlerini açıklamaktadır. Bu algoritma, ortak karakterlerin daha uzun dizinin yarısında kalan karakterleri (c sayısını) kabul ederek yer değiştirme sayısını (t) hesaplar. Bir benzerlik ölçümü olarak formülasyonu (Denklem 3.8) aşağıdaki gibi verilmektedir:

$$sim_{jaro}(s1, s2) = \frac{1}{3} \left( \frac{c}{|s1|} + \frac{c}{|s2|} + \frac{c-t}{c3} \right) \quad (3.8)$$

Winkler algoritması ise genel olarak adların başında daha az hatanın meydana geldiğini tespit eden yapay çalışmalara dayanan fikirler uygulayarak, Jaro algoritmasını geliştirmektedir. Winkler algoritması bu sebepten dolayı birinci karakterleri kabul etmek amacıyla Jaro benzerlik ölçüsünü artırmaktadır Formülasyonu (Denklem 3.9) aşağıda gösterildiği gibidir [28]:

“s”, kabul edilen karakter sayısıdır.

$$sim_{Winkler}(s1, s2) = sim_{Jaro}(s1, s2) + \frac{s}{10} (1 - sim_{Jaro}(s1, s2)) \quad (3.9)$$

### 3.1.10. Hamming mesafesi

Bir dizeyi diğer bir dizeye dönüştürmek için gerekli minimum konum değiştirme adedini veya bir dizeyi diğer bir dizeye dönüştürmüş olabilecek minimum hata adedini ölçmektedir. Daha genel bir bağlamda, Hamming mesafesi, iki dizi arasındaki düzenleme uzaklığını ölçümlemek için kullanılan farklı dize metriklerinden birisidir. Adını Amerikalı matematikçi Richard Hamming'den almaktadır [29].

### 3.1.11. Smith-Waterman mesafesi

Needleman-Wunsch algoritması kullanılımlarını takip eden on yıl içerisinde Sankoff, Reichert, Beyer ve diğerleri gen dizilerini analiz etmek amacıyla alternatif sezgisel algoritmalar geliştirmiştir. 1976'da Waterman ve arkadaşları orijinal ölçüm sistemine boşluklar kavramını eklemiştir. 1981'de Smith ve Waterman, yerel hizalamayı hesaplamak için Smith-Waterman algoritmasını yayınlamıştır. Büyük uzunlukta iki dizeyi hizalamak için: Smith-Waterman algoritması optimize edilmiştir.

Needleman-Wunsch algoritmasından en büyük farkları aşağıda maddeler halinde verilmiştir:

- Traceback prosedürü en yüksek puan alan matris hücresinden başlar ve sıfır puan alan bir hücreye ulaşılan kadar devam eder.

- En iyi yerel hizalamaları vurgulamak amacıyla negatif puanlama matris hücreleri sıfıra ayarlanır [30].

### 3.1.12. Ratcliff/Obershelp benzerliği

Ratcliff / Obershelp benzerliği algoritması 1983 yılında John W. Ratcliff ve John A. Obershelp tarafından tanıtılmıştır. Eğitim yazılımı endüstrisi üzerinde Bu algoritmanın etkisi olmuştur. Önceki dönemlerde, çoğunlukla eğitim yazılımı konusunda sadece çoktan seçmeli testler sunulmuştur çünkü girilen verilerin işlenmesi ve kontrol edilmesi için kullanıcı tarafından yazılan cevap algoritmaları gerekmektedir. Örneğin, 18. hanedanın Mısır firavunun kim olduğu sorusu için, Tutankhamun, Tutenkhamun, Tutankhamen, Tutankhamon'un cevapları doğru olarak düşünülmelidir. Ayrıca, bir kullanıcı çift “m” girmiş veya başka tür bir yanlış yazmış olabilir. Ratcliff / Obershelp algoritması bu sorunun çözülmesine yardımcı oldu. Jaro-Winkler mesafe algoritması gibi Ratcliff / Obershelp, 0 ile 1 arasında bir değer döndürür; burada 1, verilen iki dize için tam bir eşleşmedir. Aşağıda (Denklem 3.10) formülize edilmiştir.

$$DRO = \frac{2 * Km}{|S1| + |S2|} \quad (3.10)$$

Burada  $Km$  eşleşen harf sayısıdır [31].

### 3.2. Tekil Değer Ayırışımı Algoritması

Tekil Değer Ayırıştırma (TDA), karmaşık bir matrisi daha basit matrislerin çarpımı şeklinde gösterilmesini sağlayan bir işlemdir. Bu işlemlerde en çok kullanılan algoritmalar, TDA, QR Ayırışımı, Alt-Üst Ayırıştırma, CS Ayırışımı, QLP Ayırışımı, Pivot QR Ayırışımı ve Pivot QLP Ayırışımı'dır [32].

Özellikle matrisin köşegenleştirilmesi ile sıkı bir ilişkisi olan TDA, tüm matris ayırışım algoritmalarının başında gelerek, özellikle regresyon analizinde önemli bir yer tutmaktadır [33]

$$X = \begin{bmatrix} 3.01 & 0.01 & -2.99 \\ 2.99 & -0.01 & -3.01 \\ 2.00 & -4.00 & 2.00 \end{bmatrix} \text{ olsun.} \quad (3.11)$$

X matrisi aşağıdaki (Denklem 3.12, (Denklem 3.13, (Denklem 3.14) gibi çözümlenebilir [34]:

$$X = \begin{bmatrix} 3 & 0 & -3 \\ 3 & 0 & -3 \\ 0 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 2 & -4 & 2 \end{bmatrix} + \begin{bmatrix} 0.01 & 0.01 & 0.01 \\ -0.01 & -0.01 & -0.01 \\ 0 & 0 & 0 \end{bmatrix} \quad (3.12)$$

$$= \delta \begin{bmatrix} 1/2 & 0 & -1/2 \\ 1/2 & 0 & -1/2 \\ 0 & 0 & 0 \end{bmatrix} + 2\sqrt{6} \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1/\sqrt{6} & -2/\sqrt{6} & 1/\sqrt{6} \end{bmatrix} \\ + \sqrt{6}/100 \begin{bmatrix} 1/\sqrt{6} & 1/\sqrt{6} & 1/\sqrt{6} \\ -1/\sqrt{6} & -1/\sqrt{6} & -1/\sqrt{6} \\ 0 & 0 & 0 \end{bmatrix} \quad (3.13)$$

$$= \delta \begin{bmatrix} 1/\sqrt{2} \\ 1/\sqrt{2} \\ 0 \end{bmatrix} [1/\sqrt{2} \ 0 \ -1/\sqrt{2}] + 2\sqrt{6} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} [1/\sqrt{6} \ -2/\sqrt{6} \ 1/\sqrt{6}] \\ + \sqrt{6}/100 \begin{bmatrix} 1/\sqrt{2} \\ -1/\sqrt{2} \\ 0 \end{bmatrix} [1/\sqrt{3} \ 1/\sqrt{3} \ 1/\sqrt{3}] \quad (3.14)$$

$$= \theta_1 U_1 V_1 + \theta_2 U_2 V_2 + \theta_3 U_3 V_3 = \sum \theta_i U_i V_i^T \quad (3.15)$$

$$= \begin{bmatrix} 1/\sqrt{2} & 0 & 1/\sqrt{2} \\ 1/\sqrt{2} & 0 & -1/\sqrt{2} \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 6 & 0 & 0 \\ 6 & 2\sqrt{6} & 0 \\ 0 & 0 & \sqrt{6}/100 \end{bmatrix} \begin{bmatrix} 1/\sqrt{2} & 0 & -1/\sqrt{2} \\ 1/\sqrt{6} & -2/\sqrt{6} & 1/\sqrt{6} \\ 1/\sqrt{3} & 1/\sqrt{3} & 1/\sqrt{3} \end{bmatrix} = U \Theta V^T \quad (3.16)$$

Yukarıdaki denklem eşitliğinden görülmektedir ki, (Denklem 3.15)'deki  $U_i$  (Denklem 3.16)'daki  $U$  ortonormal matrisinin sütunudur. (Denklem 3.15) 'deki  $V^T$  ise, (Denklem 3.16)'daki  $V^T$  ortonormal matrisinin satırıdır.  $\Theta_i$  ise,  $\Theta$  köşegen matrisinin köşegenindeki öğesidir. (Denklem 3.15) ve (Denklem 3.16) ise,  $X$  matrisinin tekil

değer ayrışımının (TDA) açıklanmasıdır. Sadece buradaki X değil, her türlü matris denklem (Denklem 3.16) gibi çözümlenebilmektedir.

### 3.3. K-Ortalamlar Algoritması

K-Ortalamlar algoritması, devam eden her aşamada kümelerin yenilendiği ve ulaşabileceği en optimum sonuç değerlerine kadar süregelen bir algoritma çeşididir. Bu alanda etkili ve ilk tercih edilen algoritmalarından biridir. Bölümlemeli diğer algoritmalar K-Ortalamlar algoritmasına benzemektedir veya bu algoritmadan esinlenmiştir.

K-Ortalamlar algoritmasındaki k, önceden belirlenmiş olan ve aranan küme adedini ifade eder. k adet küme merkezi keyfi olarak algoritmanın başlangıç seviyesinde seçilir. Seçilen bu noktalar prototip olmak üzere isimlendirilmektedir. k adet prototip ve her bir küme olacak şekilde kümeleme analizinin başlangıcındaki durum 8 numaralı eşitlikte (Denklem 3.17) gösterildiği gibidir.

$$W_i = \{w_{il} \mid w_{il} \in \{1, \dots, k\}, l \in \{1, \dots, m\}\} \quad (3.17)$$

Algoritma girdileri, aşamaları ve elde edilen çıktı aşağıda gösterildiği gibi açıklanabilir:

Girdiler:

k: belirlenen küme adedi

D: mevcut verilere dair adımlar:

Keyfi olarak belirlenen k adet elemanın küme merkezi ( $s_1, s_2, \dots, s_k$ ) olarak belirlenmesi

Her bir elemanın en yakın olduğu  $s_i$  'nin kümesine atanması

Kümelere ait  $s_1, s_2, \dots, s_k$  değerlerinin yeniden hesaplanması

Kümede farklılık bulunmayan dek birinci aşamadan itibaren ilerlenmesi.

Adımların tamamlanmasından sonra değişiklik bulunmuyorsa durulması gerekmektedir. Çıktılar: m adet küme biçiminde olacaktır [35]. Yapılan çalışmada fiyatlarına göre 3 adet küme oluşturulmuştur:

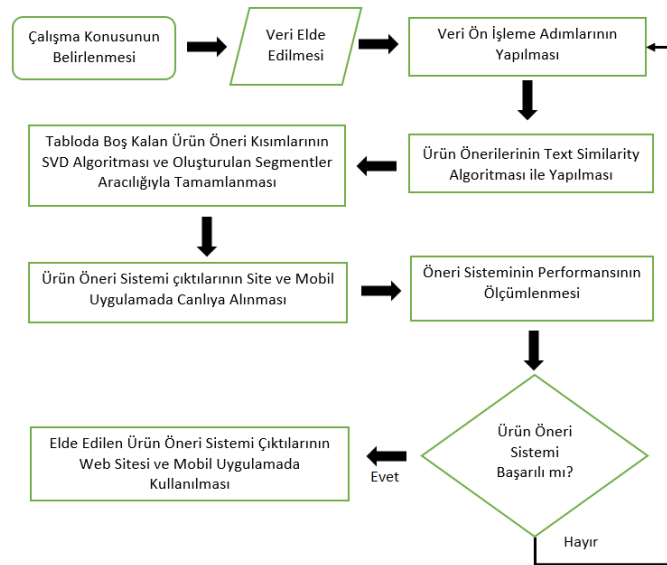
- Good
- Better
- Best



## BÖLÜM 4. UYGULAMA

Bu kısımda oluşturulan ürün öneri sistemi için alınan verilerin nasıl elde edildiğine ve analiz çalışmalarının nasıl ve hangi yöntemle yapıldığına dair bilgiler verilecektir.

Çalışmada züccaciye, elektrikli ev aletleri, dekoratif ürünler ve ev tekstili alanında satış faaliyeti gösteren bir firmaya ait 8 aylık veriler baz alınmıştır. Öncelikle veri ön işleme adımları tamamlanmış ve bu kapsamda özelleştirilmiş fonksiyonlar yazılmıştır. Ürünlerin ilk adımda Metin Benzerliği algoritmasına dahil olarak uygun ürün önerileri bulunmuştur. Sonraki adımda TDA algoritmasına dahil olarak ürün öneri tablosu tamamlanmıştır ve elde edilen çıktılar csv formatında alınarak Google Tag Manager kodlarıyla sisteme eklenmiştir. Google Analytics üzerinden sonuçları gözlemlenmiştir. Ayrıca Python kullanılarak Excel çıktısı üzerinden satış miktarları baz alınarak, her bir ürün için önerilen 3 ürünün mevcut duruma kıyasla değişim yüzdeleri hesaplanarak geliştirilen ürün öneri sisteminin etkileri hesaplanmıştır. Şekil 4.1.'de çalışma adımlarının akışını gösteren şema verilmiştir.



Şekil 4.1. Uygulama Akış Şeması

#### 4.1. Verinin Elde Edilmesi

Tez çalışmasında kullanılan verinin tamamı züccaciye, elektrikli ev aletleri, dekoratif ürünler ve ev tekstili alanında satış faaliyeti gösteren bir firmaya aittir. Bu verinin yaklaşık %95'i için Dax sorguları yazılmaktadır. Dax ortamında oluşturulan bu sorgular Python'da çalıştırılarak analizde kullanılacak veriler elde edilmektedir. Ürünlerin stokta bulunma, aktif satışta olmasına ilişkin verinin temini ise e-ticaret ekibinin oluşturduğu Excel dosyalarından Python kodlarıyla çekilmektedir.

Şekil 4.2.'de gösterildiği gibi Dax sorgusunun python programlama içine yazılmasıyla ilgili verinin elde edilmesi sağlanmıştır. Üzerinde analizi yapılacak veriler 2021 Mart ve Ekim ayları arasında gerçekleşmiş alışverişleri kapsamaktadır. Burada sadece etkin olan müşteriler seçilmiş ve bazı ürün grupları filtrelenerek çıkarılmıştır.

```
import time
start = time.time()
query = """EVALUATE
SUMMARIZECOLUMNS(
    ContactDim[ContactId],
    ChannelDim[ChannelName],
    ProductDim[ProductGroup],
    ProductDim[ProductSubGroup],
    ProductDim[InventId],
    ProductDim[InventName],
    ProductDim[BrandName],
    DataAreaDim[DataAreaId],
    StoreDim[StoreName],
    DateDim[Day],
    Sales[ReturnsSalesId],
    KEEPFILTERS( TREATAS( {"Etkin"}, ContactDim[StateCode] )),
    KEEPFILTERS( FILTER( ALL( DateDim[Day] ), DateDim[Day] >= DATE(2021,3,1) && DateDim[Day] <= DATE(2021,10,31))),
    KEEPFILTERS( FILTER( ALL( ProductDim[ProductGroup] ), NOT( ProductDim[ProductGroup] IN {"POSET","SSS %18","POP","MSSS %18","%
))),
    "SalesNetAmountWoVat", [SalesNetAmountWoVat],
    "NetSales", [NetSales]
)"""

from pandas import DataFrame

with Pyadomd(conn_str) as conn:
    with conn.cursor().execute(query) as cur:
        df_firma_original = DataFrame(cur.fetchone(), columns=[i.name for i in cur.description])

stop = time.time()
print(stop-start)
```

Şekil 4.2. Kullanılacak verilere ait Dax Sorgusu ve Python kodlarıyla çekilmesi

Tablo 4.1.'de analizde kullanılacak verilerin metrik tanımlamalarına yer verilmiştir. Analizde kullanılması amacıyla ürüne ait ürün grubu, ana kategori bilgileri, müşterinin beğenilerinde kullanılması amacıyla müşteri ID'si gibi veriler kullanılmıştır.

Tablo 4.1. Veri Tanımlamaları

| Veri Tipi                   | Veri Açıklaması                          |
|-----------------------------|------------------------------------------|
| ContactDim[ContactId]       | Müşterilere ait unique ID'ler            |
| ChannelDim[ChannelName]     | Satış kanalı bilgisi (Perakende, online) |
| ProductDim[ProductGroup]    | Ürün grubu bilgisi                       |
| ProductDim[ProductSubGroup] | Ürün alt kategori bilgisi                |
| ProductDim[InventId]        | Ürün kodları(unique)                     |
| ProductDim[InventName]      | Ürün adları                              |
| ProductDim[BrandName]       | Ürünlerin ait olduğu marka isimleri      |
| DataAreaDim[DataAreaId]     | Şirket isimleri                          |
| StoreDim[StoreName]         | Web site ismi, mağaza ismi               |
| DateDim[Day]                | Gün bilgisi                              |
| Sales[ReturnSalesId]        | Satış fatura numaraları                  |
| [SalesNetAmountWoVat]       | KDV'siz net ciro                         |
| [NetSales]                  | Net satış adedi                          |

## 4.2. Yöntem

### 4.2.1. Kullanılan araç-gereçler

Çalışmada tüm veri ön işleme adımları ve analizleri Python programlam dili ile gerçekleştirilmiştir. Kullanılan python kütüphaneleri aşağıdaki gibidir:

- Pandas,
- Numpy,
- Sklearn,
- Matplot.pyplot,
- Datetime

Geliştirilen ürün öneri sisteminin mevcut durumla karşılaştırmasının yapılması ve öneri sisteminin performansının gözlemlenebilmesi için Google Tag Manager kodlarıyla beslenen Google Analytics Tool'u kullanıldı.

#### 4.2.2. Veri ön işleme (veri manipülasyonu)

Oluşturulması hedeflenen ürün öneri sistemi gerçek veriler üzerinden oluşturulmuştur. Kullanılacak algoritmaların doğru ve etkin bir şekilde çalışabilmesi için ve önerilecek ürünlerin doğrudan müşteriye gösterimi yapılacağı için detaylı ve kapsamlı bir veri ön işleme çalışması yapılmıştır.

E-ticaret ekibinden alınan veri üzerinde satışa açık (aktif) ve stokta bulunma durumlarına göre incelenmiştir ve filtreleme işlemi yapılarak analiz çalışmasında dahil edilmiştir. Stoku olmayan veya stokta bulunmasına rağmen e-ticaret satışına kapalı ürünlerin olması ihtimaline karşı bu ürünler öneri olarak müşterilere gösterimini engellemek için bu işlem yapılmıştır.

Outlet ürünler, fiyat segmentasyon çalışmasında yanıltıcı olmaması adına analiz kapsamından çıkarılmıştır.

Satış cirosu 0,01 TL'den büyük olan faturalar alınarak iade faturaları kapsam dışı bırakılmıştır.

Toptan alımların çalıştırılacak algoritmayı etkilememesi için satış adedi 20'den fazla olan faturalar dahil edilmemiştir.

##### 4.2.2.1. Ürün gruplarının düzenlenmesi

Ürün grupları için algoritmanın doğru çalışması amacıyla bazı düzenlemeler yapılmıştır. Öneri sisteminin çalışmasından önce Python üzerinden ürünler ve ait oldukları ürün grupları incelenmiştir. Bazı ürünlerin boş ve hatalı geldiği tespit edilmiştir. Bu ürünlerin bağlı olduğu ürün grupları filtrelenerek analiz kapsamından çıkarılmıştır. Çıkarılan ürün gruplarından bazıları aşağıdaki gibidir:

- "Poşet", "SSS %18", "Pop", "MSSS %18", "Yedek Parça-KEA", "SarfMalzemesi", "Yedek Parça-Sofra", " Yedek Parça-Dekoratif", " Yedek Parça-

Mutfak", " Yedek Parça-Tekstil", "Throw", "Tanımsız", "Diğer", "SSS %8", "Mağaza Dekorasyon Malzemesi", "MSSS %8", "Teşhir-Bardak Seti", " Teşhir-Yemek TK", " Teşhir-ÇKB TK", "Test", "GIFTCARD", "9010111", "Hediye Sepeti", "1113110"

Öneri sistemi ürün grubu bazlı çalışmaktadır ve 3 ürün önerisi sunması hedeflenmektedir. Bu yüzden 3’den az ürüne sahip ürün grupları çıkarılmıştır.

Tavsiyelerin daha doğru çalışması için ihtiyaç görülen durumlarda yeni ürün grupları oluşturulmuş ve ilgili ürünler bu gruplara atanmıştır:

- Ürünler, servis seti ürün grubuna aitse ve ürün adı içerisinde “SET” geçmiyorsa bu ürünler sofra servis gereçleri ürün grubuna atanmıştır.
- Ürünler, servis gereçleri ürün grubuna aitse ve ürün adı içerisinde “SET” geçiyorsa Mutfak Aksesuarları Servis Seti ürün grubuna atanmıştır.
- Kahve yanı bardaklar için ürün grubu bulunmadığı ve bu tip ürünler “6’lı Bardak” kategorisinde değerlendirildiği için “Kahve Yanı” ürün grubu oluşturularak ilgili ürünler buraya atanmıştır.

#### 4.3. Segment ve Fonksiyonların Oluşturulması

Tekil Değer Ayırışımı algoritmasından gelen korelasyon değerleri ile beraber dikkate alınacak renk, marka ve fiyat özellikleri için fonksiyonlar ve segmentler oluşturularak ürün öneri tablosunun çok yönlü ve doğru bir şekilde tamamlanması gerçekleştirilmiştir.

##### 4.3.1. Renk fonksiyonu tanımlaması

Ürün isimlerinde geçen renklere dair bilgiler incelenerek fonksiyon tanımlanmıştır. Buradaki amaç bir renge ait kısaltma, İngilizce karakter veya yanlış yazımdan kaynaklı oluşan farklılıkları gidererek renk bilgisinin doğru bir şekilde algoritmada kullanılmasını sağlamaktır. Bir örnek üzerinden incelenecek olursa; “Su Yeşili” rengi

ürün isimlerinde “S.YESİL”, “SU YESİLİ”, “AQUA GREEN” gibi farklı olarak yazılabilmektedir. Burada ürün isimlerinde geçen farklılıkları gidererek “SU YEŞİL” olarak tanımlamasını yapılmıştır. Renk bilgisi yazılmayan ürünlerde ise “RENKSİZ” tanımlaması yapıldı. Şekil 4.3.’de renk bilgisine ait tanımlamaların yazıldığı fonksiyon gösterilmiştir.

```
def find_color(product_name):
    if "MAVİ" in product_name or "Mavi" in product_name or "MAVI" in product_name or "BLUE" in product_name:
        color="MAVİ"
    elif "ROSEGOLD" in product_name or "ROSE GOLD" in product_name:
        color="ROSEGOLD"
    elif "BLACKGOLD" in product_name or "BLACK KGOLD" in product_name or "BLACK&GOLD" in product_name:
        color="BLACKGOLD"
    elif "ALTIN" in product_name or "GOLD" in product_name:
        color="ALTIN"
    elif "GÜMÜS" in product_name or "PLATIN" in product_name or "SILVER" in product_name or "PLATINIUM" in product_name:
        color="GÜMÜS"
    elif "PINK" in product_name or "PEMBE" in product_name or "Pembe" in product_name or "PİNK" in product_name:
        color="PEMBE"
    elif "KIRMIZI" in product_name or "RED" in product_name or "Kırmızı" in product_name:
        color="KIRMIZI"
    elif "SİYAH" in product_name or "BLACK" in product_name:
        color="SİYAH"
    elif "BEYAZ" in product_name or "WHITE" in product_name:
        color="BEYAZ"
    elif "KREM" in product_name or "CREAM" in product_name:
        color="KREM"
    elif "MURDUM" in product_name or "MURDUM" in product_name:
        color="MURDUM"
    elif "LACİVERT" in product_name or "LACİVERT" in product_name:
        color="LACİVERT"
    elif "SU YEŞİL" in product_name or "S.YESİL" in product_name or "SU YESİLİ" in product_name or "AQUA GREEN" in product_name:
        color="SU YEŞİL"
    elif "YEŞİL" in product_name or "YESİL" in product_name or "GREEN" in product_name:
        color="YEŞİL"
    elif "ANTRASİT" in product_name or "ANTRASİT" in product_name:
        color="ANTRASİT"
    elif "BEJ" in product_name or "BEIGE" in product_name:
        color="BEJ"
    elif "SARI" in product_name or "YELLOW" in product_name:
        color="SARI"
    elif "GRİ" in product_name or "GREY" in product_name or "GRI" in product_name:
        color="GRİ"
    elif "WOOD" in product_name:
        color="WOOD"
    else:
        color="RENKSİZ"
    return color
```

Şekil 4.3. Renk bilgisi fonksiyon tanımlama

#### 4.3.2. K-Ortalamlar ile fiyat segmentasyonun oluşturulması

Ürünlerin ortalama satış fiyatları belirlendikten sonra K-Ortalamlar algoritması kullanılarak “Good”, “Better” ve “Best” segmentleri oluşturulmuştur. Bu segmentlerin oluşturulmasına yönelik Python kodlamasının bir kısmı Şekil 4.4.’de gösterildiği gibidir. Bu işlem yapılarak aynı segmentten ürün önerilerinin kullanıcılara sunulması amaçlanmıştır. Ayrıca firmanın değer gördüğü bazı özel ürün grupları için doğrudan segment tanımlamaları yapılarak fiyat segmentinden dolayı öneri sisteminin düşük fiyatlı ürün önerilerini sunmasının önüne geçilmiştir.

```

from sklearn import cluster

def kmeans_segment(segment_key):
    if segment_key==good_segment:
        segment="Good"
    elif segment_key==best_segment:
        segment="Best"
    else :
        segment="Better"
    return segment

# 3'den az çeşitli olan grupların ürünlerin kaldırılması

aaa=df_karaca.groupby("ProductDim[ProductGroup]")["ProductDim[InventName]"].nunique()
aaa=aaa.reset_index()
fazla_çesit_olan_gruplar=aaa[aaa["ProductDim[InventName]"]>=3]["ProductDim[ProductGroup]"].values
aaa=aaa[aaa["ProductDim[InventName]"]<3]["ProductDim[ProductGroup]"]
az_çesitli_gruplar=aaa.values
df_karaca=df_karaca[~df_karaca["ProductDim[ProductGroup]"].isin(aaa)].reset_index(drop=True)

group_product_names["segment"]=" "

for i in fazla_çesit_olan_gruplar:
    prices=group_product_names[group_product_names["ProductDim[ProductGroup]"]==i]["Price"].reset_index(drop=True)
    k_means = cluster.KMeans(n_clusters=3)
    k_means.fit(np.array([prices]).reshape(-1, 1)) #K-means training
    y_pred = k_means.predict(np.array([prices]).reshape(-1, 1))
    pred = pd.DataFrame(y_pred)
    pred.columns = ['segment_key']
    # we merge this dataframe with df
    prediction = pd.concat([prices,pred], axis = 1)
    good_segment=prediction[prediction["Price"]==prediction["Price"].min()]["segment_key"].values[0]
    best_segment=prediction[prediction["Price"]==prediction["Price"].max()]["segment_key"].values[0]
    prediction["segment"]=prediction["segment_key"].apply(kmeans_segment)
    prediction=prediction.drop(columns=("segment_key"))
    group_product_names.loc[group_product_names["ProductDim[ProductGroup]"]==i,"segment"]=prediction["segment"].values

group_product_names.loc[group_product_names["segment"]=="", "segment"]="Better"

```

Şekil 4.4. K-Ortalamalar algoritması ile fiyat segmentasyonunun yapılması

### 4.3.3. Ürünlere ait marka fonksiyonunun tanımlanması

Ürün isimlerinde geçen marka adlarının farklı olarak geldiği gözlemlenmiştir. Aynı marka ürün önerilerinin yapılabilmesi için yeni bir sütun oluşturularak ürünlere ait marka tanımlamaları yapılmıştır. Bir örnek üzerinden incelenecek olursa; ürün isimleri içerisinde geçen “KAID”, “Kitchenaid”, “KitchenAid” olarak farklı gelmesinin önüne geçilerek bu ürünlerin tümü için “KITCHENAID” marka tanımlaması yapılmıştır.

## 4.4. Text Similarity Algoritması

Bu algoritmanın kullanılmasındaki amaç ürün isimleri içerisinde geçen model ailelerinin tespit ederek birbirlerine olan uzaklıklarını tespit etmek ve benzer ürün önerilerini müşterilere sunmaktır.

Şekil 4.5.’de gösterildiği gibi öncelikli olarak ürün isimleri incelenerek içerisinde geçen kelimelerden liste oluşturulmaktadır. Listenin oluşturulmasındaki temel amaç

bu listede bulunan kelimeler ürün adı içerisinde bulunuyorsa hariç tutularak ceza puanlarının yüksek çıkmasını önlemek ve yanlış ürün önerilerinin önüne geçilmesini sağlamaktır. Burada yer alan kelimeler birçok ürün içerisinde bulunduğu için birbirleri olan ilişkide uzaklığı azaltmakta ve ceza puanını düşürerek algoritmayı yanıltmaktadır. Bunun sonucunda da aynı ürün ailesinde bulunmamasına rağmen ürün önerileri getirmesine neden olacaktır.

#### # Liste Oluşturulması

```
liste=np.array(["YMT", "PRC", "KARACA", "CKB", "KAHVALLI", "SETİ", "PARÇA", "SET", "KUTULU", "CM", "X", "KRC", "SERVİS",
"KAYIK", "KASE", "ÇUKUR", "KİŞİLİK", "KŞLK", "ÇAYDANLIK", "CEZVE", "TAKIM", "TAKIMI", "BAHARAT", "SALATA",
"ÇAY", "BARDAGI", "BARDAK", "PASTA", "GONDOL", "TABAK", "TABAGI", "BAHARAT", "DEĞİRMENİ", "SERVİS", "BIÇAK",
"TECERE", "ML", "CEREZLİK", "SUNUM", "OVAL", "14CM", "19CM", "20CM", "25CM", "29,5", "24X12", "BİBERLİK", "SAKLAMA",
"KABI", "Lİ", "-", "MUG", "BÜYÜK", "KAYIK", "BOY", "BÜYÜK", "ORTA", "FİNCANI", "+", "RECELLİK", "15,5", "30,5", "YEMEK",
"TAVA", "SAHAN", "33*25", "LÜ", "DEĞİRMENİ", "ÇAYDANLIK", "23CM", "BOL", "6'LI", "LH192-3", "11255-2", "12789-1",
"11255-1", "13X7", "15X7", "26.6", "15.7", "13.6", "K.F.", "11254-3", "11255-3", "7,75", "9,75", "11,75", "3LÜ", "PRC",
"MEŞRUBAT", "SU", "SPATULA", "BİBLO", "MUMLUK", "NUM", "6'LI", "FİNCAN", "MUG", "KASESİ", "LH193-2", "34X17",
"LH193-2", "13X7", "15X7", "11254-5", "15,5X8", "KÜÇÜK", "PİKNİK", "SEPETİ", "SEPET", "TEPSİ", "KABI/CORBALIK", "7'LI",
"PREMIUM", "NB", "KŞ+LK", "RND", "KAHVE", "KABI", "SAKLAMA", "KARE", "ÇAY", "C098BA1201-102588", "SALATA", "KUPA",
"12X12X10", "12X12X12", "12X12X16", "FENER", "VAZO", "CERCEVE", "CERCEVE", "ANAHTARLIK", "OBJE", "20X3,5CM", "35X19,5CM",
"26X45", "41,5X21CM", "DEKORATİF", "2'LI", "KESME", "TAHTA", "TAHTASI", "4KİŞİLİK", "12PRÇ", "L", "M", "S", "BYK",
"HAVLU", "AİLE", "AİLE", "BANYO", "6'", "12'", "12078-1", "16,5X14", "12078-2", "12078-3", "11'", "11254-6", "18X9", "35,5",
"PLT", "V2", "SETİ", "KATLI", "PP'LI", "PP'LI", "TATLI", "FİRİN", "K.F.", "7,75", "9,75", "11,75", "İKRAMLIK",
"GRILL", "NOK", "KREP", "ANTİBAKTERİYEL", "SÜTLÜK", "KARNİYARİK", "BAKRAÇ", "16cm", "20cm", "YASTIK", "PIKE", "PIKE",
"K.YARIK", "NEVRESİM", "NEV", "NEVRESİM", "BAŞUCU", "29X14", "13X7", "10849-10", "CKR", "19X9", "2Lİ", "7,5",
"17CM", "3,5", "TOST", "MAKİNESİ", "TEA", "YUMURTA", "HAŞLAMA", "WAFFLE", "DAR", "UZUN", "ANTB", "TAVA+SAHAN", "BIÇAGI",
"KAŞIĞI", "SU", "TASI", "GUVEÇ", "1,6", "6,7", "LT", "LT", "SARMA", "PANKEK", "TOHBİK", "PİLAV", "ISITICISI",
"V2", "KCK", "BYK", "11254-1", "39X19,5", "3' LÜ", "KEK", "ORTA", "SÜRAHİŞİ", "EHLİ KEYF", "5,5", "2,9", "CİFT",
"KİSİLİK", "PAELLA", "TAVASI", "TEK", "16CM", "ŞAMDAN", "35CM", "DEK", "23X28", "25X", "40X19CM", "9X9X15", "11X11X20",
"KABI-L", "KABI-M", "KELEPCELİ", "26CM", "KALIBI", "BLENDER", "KAVANOZ", "-XL", "L", "XL", "M", "XXL", "S", "SİRKELİK", "8,5",
"6,9", "4,7", "10", "OVAL", "YUVARLAK", "MUTFAK", "ROBOT", "ROBOTU", "KARE", "DİKDÖRTGEN", "F", "MAKARNA", "İBRİK",
"57-6030-0050-V00", "57-9917-0390-T00", "68-B328-0370", "6KİŞİLİK", "-L", "KAHVALLILIK", "ÇAYD.", "T.K.M.", "MK.",
"İSİR", "PATLATMA", "&NEV.", "BATT", "TK", "10X10X14", "MAMA", "FRENCH", "PRESS", "13X18CM", "FOTOĞRAF", "CERÇEVESİ",
"57-9917-0390-T00", "8", "35*27", "35*27", "13X18CM", "1.5", "TENC.", "16X21", "5' Lİ", "NEV.", "T.K.M.", "BLNDR", "DOĞRAYICI",
"VE", "SORBE", "GÖZLEME", "Lİ", "+SAHAN", "TEPSİLİ", "TERMOS", "SMALL", "MANTI", "6LI", "24,75CM", "HB8005CC",
"68-4048-0350", "350ML", "30X25", "7CM", "İZGARA", "350CC", "23,5CM", "BM027-8-XP-10.5", "30X45CM", "19X26",
"DOĞRAYICI-1250ML", "DAİLY-MIX5501", "SKHBV53E08", "SKHBV53EWH", "EL", "MİKSERİ", "1933H", "1904H", "7010H", "SKS84026",
"KİSİYE", "OZEL", "1920H", "1,19", "SKFC0516", "K400", "STAND", "MİKSER", "STANT", "SKSM175P",
"TAİNABİLİR SARJLI", "BLENDFIT&GO", "PERSONAL", "SMOOTHIE", "SHAKER", "TOSTMK", "2101H", "DOĞRAYICI", "CHOPPER",
"38CM", "29CM", "2' Lİ", "16X33CM", "8*8*10.8", "14*14*4", "LARGE", "TABAGI", "6,5X10", "KC-915", "8,5X9,5CM",
"11,5X12,5CM", "23X12CM", "A.SERVİSİ", "2LI", "9X10CM", "5,5X29", "24X20", "2*(7,5*7,5", "cm)DG106", "DG58", "DG62-8",
"DG62", "12x17", "cm", "10x15", "C.F.", "LT116", "32,5CM", "SUPLA", "3LÜ", "9,5X6CM", "SALAD&PASTA", "M-648",
"1.9L", "4LÜ", "MK", "LI", "16,5X16CM", "19X16", "150X220CM", "MASA", "ÖRTÜSÜ", "11,5X36,5CM", "17.8X17.8X6.4",
"19X38", "34CM", "27,5X6,5", "27,5", "KC-860", "SEBZE", "SOYACAĞI", "TİRBÜŞON", "KC-871", "CEZVE-L", "KCK", "16X16X16",
"KC-857", "21X10", "ÇATAL", "KAŞIK", "3MM", "V3", "24PRÇ", "ROBOCHOP", "MİDİMİX", "MİDİMİX", "MOMS",
"2833H", "2804H", "1.7", "1.7", "2.1", "2,1", "MK-37", "QUICK", "1500W", "4,8", "SKSM175PS", "TAİNABİLİR", "SARJLI",
"SHKR", "SMOOTHİE", "TOASTBUSTER", "1330H", "MAK.", "1331H", "KITCHEN", "NARENCİYE", "SİKACAY", "AP.", "SKFP0719EER",
"17X17X12", "13X13X24", "12X12X19", "12X12X26", "29X13", "24,5X16,5CM", "19X17CM", "37X33X13", "14X47", "M-649",
"6.3*18.5cm", "SE-286", "G-230", "T-269", "29,3x7,5", "3MM",
"SKFP0919", "SKFP0919", "SKFP0719EBM", "EF802", "PROMIX", "3003H",
"18PRÇ", "PCS", "10X18CM", "19,5X17CM", "19,5X14CM", "11X28,5", "25X19CM", "29X17X17", "20,5X18,5X4,2", "17X27CM",
"12*12*14.1", "7X7X13", "33X7X42", "13,5X13,5X23", "20X14", "30X20", "14X9X6CM", "70X25", "40CM", "20X19", "24X35CM",
"25X26", "8X38CM", "10X", "12X30", "50CM", "47X18CM", "36X21CM", "30CM", "32CM", "50X31CM", "TK1225", "39*28*19CM",
"1503H", "ELEKTRİKLI", "CEZVESİ", "6KŞLK", "6KŞLK", "2.", "8PRÇ", "25,4", "14", "37,5", "24*32", "ODA", "KOKUSU"])
```

Şekil 4.5. Text Similarity uygulaması için liste oluşturulması

Algoritmanın çalışması sıralamaya bakılmaksızın kelimelerin benzerliğine göre ilerlemektedir. Karşılıklı 2 metne bakılarak skor puanları verilmektedir. Örneğin; “Ahmet okula gitti” ve “Okula Ahmet gitti” cümleleri kullanılırsa ceza puanı 0 gelir. “Ahmet okulda gitti” ve “Okula Ahmet gitti” cümlelerini kullanırsak ceza puanı 0,84 gelmektedir. Listedeki bulunan kelimeler ürün adı içerisinde bulunuyorsa bu kelimeleri alınmamaktadır. En düşük skor puanına sahip ürün seçilerek ilerlenmektedir. Skor puanı ne kadar yüksekse o kadar cezalandırılmış kabul edilmektedir. Ceza puanı filtresi 0,86 olarak belirlenerek ilerlenmiştir.

Çok az sayıda üründe, oluşturulan listedeki kelimeler hariç tutulduğunda ürün alanında az sayıda da olsa boş gelen değerler tespit edilmiştir. Programın hata vermesini engellemek amacıyla boş gelen değerler çıkarılmıştır. Ayrıca sayısal değer içeren ürünler de kapsam dışı bırakılmıştır. Şekil 4.6.'da Metin Benzerliği algoritmasına ait yazılan kodların ekran görüntüsü paylaşılmıştır. Ürün öneri sisteminde tablonun öncelikli olarak doldurulması burada gerçekleşmiştir.

```
for i in product_recommendation_table["Product_Name"]:
    #ürün grup name alınması
    group_name=group_product_names[group_product_names["ProductDim[InventName]"==i]["ProductDim[ProductGroup]"].values[0]
    #ürün segment değeri
    #segment_level=group_product_names[group_product_names["ProductDim[InventName]"==i]["segment"].values[0]
    #segment_filtered_list=segment_filtreleme(Corr_List, segment_Level)
    boyut=updated_reco_table[updated_reco_table.index== group_name].dropna(axis=1).shape[1]

    abc=' '.join([w for w in i.split(" ")[0:9] if w not in liste and not w.isdigit()])
    print(abc)

    if (updated_reco_table[updated_reco_table.index== group_name].shape[0])!=0:

        for k in np.arange(boyut):
            reco_group=updated_reco_table[updated_reco_table.index== group_name].dropna(axis=1).iloc[0,k]
            first_product=[]
            second_product=[]
            distance_score=[]

            for j in df_karaca[df_karaca["ProductDim[ProductGroup]"==reco_group]["ProductDim[InventName]"].unique():
                #print(k)

                if (' '.join([w for w in i.split(" ")[0:9] if w not in liste and not w.isdigit()])=="") or \
                (' '.join([w for w in j.split(" ")[0:9] if w not in liste and not w.isdigit()])==""):
                    i
                    #print("+++++++",i,"-----",j)
                else:
                    distance=documentSimilarity(' '.join([w for w in i.split(" ")[0:9] if w not in liste and not w.isdigit()]), '
                    first_product.append(i)
                    second_product.append(j)
                    distance_score.append(distance)

            similarity_distance=pd.DataFrame({"first_product":first_product, "second_product":second_product,
            "distance_score":distance_score})

            if similarity_distance.sort_values("distance_score").shape[0]!=0:

                #print(similarity_distance[similarity_distance["first_product"]==i].sort_values("distance_score").values[0].resha
                first,second, distance=similarity_distance.sort_values("distance_score").values[0]

                if distance<0.86:
                    #print(first,"---",second,"---",distance)
                    index_value=product_recommendation_table[product_recommendation_table["Product_Name"]==i].index.values[0]
                    product_recommendation_table.iloc[index_value,k+1]=second+"..."
            else:
                print(group_name)
```

Şekil 4.6. Metin Benzerliği algoritmasının uygulanması

Örneğin; “KARACA HELEN 24 KİŞİLİK 64 PARÇA YEMEK TAKIMI” için aranılan özellik “HELEN” modelidir ve bu model ailesine dahil ürünler çıkarılarak ilişki kurulmak istenmektedir. Bu şekilde ilerlenerek tüm ürünler model aileleri bazında değerlendirilmiştir ve tabloda ilgili alanlar doldurularak ürün öneri sisteminin ilk adımı tamamlanmıştır.

#### 4.5. TDA (Tekil Değer Ayırışımı) Algoritmasının Uygulanması

Metin Benzerliği algoritmasının çalışmasından sonra ürün öneri tablosundaki çoğu alan boş olarak görülmektedir. Bu alanların doldurulması için öncelikli olarak tüm ürünlerin TDA algoritmasına girmesi sağlanmaktadır.

TDA (Tekil değer ayrıştırması) algoritması müşteri ID'lerini ve ürün adlarını temel olarak ilerlemektedir. Kullanıcı tabanlı olmayıp ürün içerik tabanlı ilerlemektedir. Müşteri ID'lerinin kullanılmasındaki amaç birlikte alınan ürünleri yakalamaktır ama hangi ürünün kim tarafından alındığı tez çalışmanın kapsamında bulunmamaktadır.

Mağazadan müşteri kaydı olmadan yapılan alışverişlerde veya benzeri durumlarda müşteriye bağlı satış bulunamadığı için “None” gelen değerler bulunmaktadır. Algoritmanın çalışmasını olumsuz yönde etkilememesi için fatura numaraları bulunan ama müşteri ID'si bulunmayan ilgili satırlar filtrelenmektedir.

Tek bir ürün alan müşteriler ürün önerisine fayda sağlamayacağı ve işlemsel olarak yük getireceği için çıkartılmaktadır. Ayrıca çoğunluğu personellere ait olduğu tespit edilen, müşteri ID üzerinde 80 adetten fazla ürün bulunan satışlar ve bir müşteriye ait 3'den az alışveriş olduğu durumlar da çıkartılmaktadır. 2 ürün satın almış veriler daha az kullanışlı olması ve bellek kullanımını artırması nedeniyle kapsam dışında bırakılmıştır.

Şekil 4.7.'de TDA algoritmasının uygulanmasına yönelik yazılan Python kodlarının bazı kısımları ekran görüntüsü olarak paylaşılmıştır.

```

#tümü 1'Den oluşan yeni sütun ekle
item_user["rating"]=1

item_user.rename(columns={"ContactDim[ContactId]":"user_id", "ProductDim[InventName]":"item_id"}, inplace=True)

#item_user = item_user.rename(columns={"ContactDim[ContactId]":"user_id", "ProductDim[InventName]":"item_id"})

#tek 1 defa alışveriş yapan customer
user_list=pd.DataFrame(item_user["user_id"].value_counts())
user_list[user_list["user_id"]!=1]

# tek bulunan customer ID'lerin çıkarılması
item_user=item_user[item_user['user_id'].duplicated(keep=False)].reset_index(drop=True)
item_user

alışveriş_sayıları=item_user.groupby("user_id").agg("count").reset_index()

user_id_liste=alışveriş_sayıları[(alışveriş_sayıları["item_id"]<80) & (alışveriş_sayıları["item_id"]>2)][["user_id"].values

user_id_liste

item_user=item_user[item_user["user_id"].isin(user_id_liste)].reset_index(drop=True)

item_user

# user_item pivot table oluşturulması
rating_crosstab = item_user.pivot_table(values='rating', index='user_id', columns='item_id', fill_value=0)
#rating_crosstab.head()

#sadece kontrol boyutlar için
rating_crosstab.shape

#transpozunu aldık
X = rating_crosstab.T

SVD = TruncatedSVD(n_components=40, n_iter=7, random_state=5)
resultant_matrix = SVD.fit_transform(X)
resultant_matrix.shape

resultant_matrix[0]

### correlation matrix
corr_mat = np.corrcoef(resultant_matrix)
corr_mat.shape

```

Şekil 4.7. TDA algoritmasının uygulanması

Alışveriş yaptığı ürünleri, müşterilerin beğendiği kabulü ile ilerlenmektedir ve tüm müşteri ID'lerine aldığı her bir ürün için matriste 1 değeri atanmaktadır.

Müşterilerin aldığı ürünlerin bulunduğu sütuna “1”, diğer sütunlara “0” değeri verilmektedir. Örneğin; Bir müşteri 5 ürün almış ise ilgili satır bazında ilişkili olduğu 5 sütuna “1” değeri verilirken diğer tüm sütunlara “0” değeri atanmaktadır. Bu şekilde ilerlenerek yaklaşık 300 bin satır (müşteri ID) ve 2800 sütunluk(ürün) bir matris oluşturulmaktadır.

Matrisin transpozu alınarak 40 bileşenli ve 7 iterasyonlu bir işlem uygulanmaktadır.

Ürünler için de ürün sayısı kadar satır ve sütun oluşturularak karşılıklı geldiği değere 1 ve diğer alanlara 0-1 arası değerler getirilmektedir. Burada da yaklaşık [2800,2800] boyutunda bir matris oluşturulmaktadır. Ürün için matris çarpımı yaparak korelasyon



“Tava” ürün grubu için yine aynı ürün grubundan öneriler yapılabilmesi sağlanmaktadır. Burada aynı ölçülerde ürün önerilmemesi için farklı boyutlarda gelmesi kriteri de eklenmiştir. Bu ve benzeri birçok ürün grubu için de özel tanımlamalar yapılmıştır.

```
if (group_name=="TAVA") and (reco_group=="TAVA"):
    first_product=[]
    second_product=[]
    distance_score=[]

    for j in group_product_names[group_product_names["ProductDim[ProductGroup]"]==reco_group]["ProductDim[InventName]":
        a="".join([str(x) for x in i if x.isdigit()])
        b="".join([str(x) for x in j if x.isdigit()])

        if (a=="") or (b==""):
            distance=documentSimilarity(i,j)
            if distance>0.6:
                first_product.append(i)
                second_product.append(j)
                distance_score.append(distance)

        else:
            number_check_distance=documentSimilarity("".join([str(x) for x in i if x.isdigit()]),"".join([str(x) for
            if (number_check_distance>0) and (a not in b) and (b not in a):
                distance=documentSimilarity(i,j)
                first_product.append(i)
                second_product.append(j)
                distance_score.append(distance)

    similarity_distance=pd.DataFrame({"first_product":first_product, "second_product":second_product,
                                     "distance_score":distance_score})
    product=similarity_distance[similarity_distance["distance_score"]>0].sort_values("distance_score").reset_index(dr

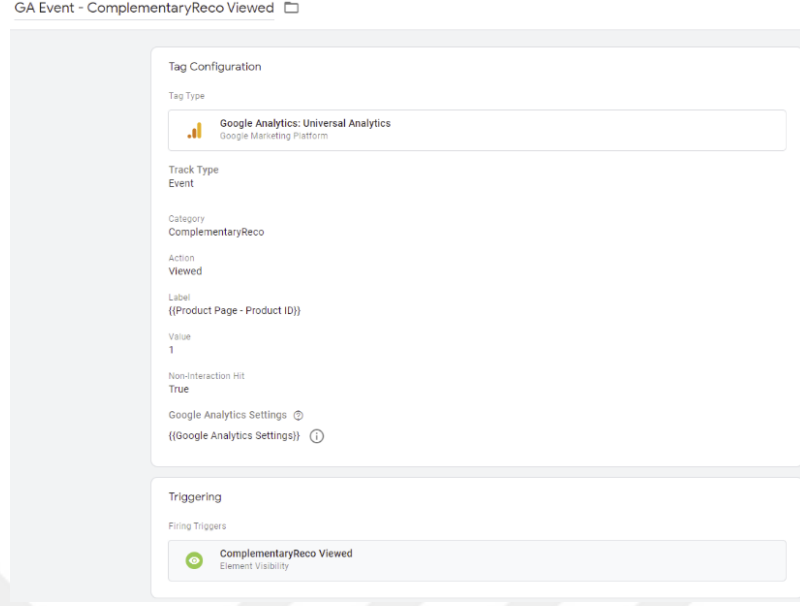
    if k==0:
        product=similarity_distance[similarity_distance["distance_score"]>0].sort_values("distance_score").reset_inde
    elif k==1:
        product=similarity_distance[similarity_distance["distance_score"]>0].sort_values("distance_score").reset_inde

    product_recommendation_table.iloc[index_value,k+1]=product
```

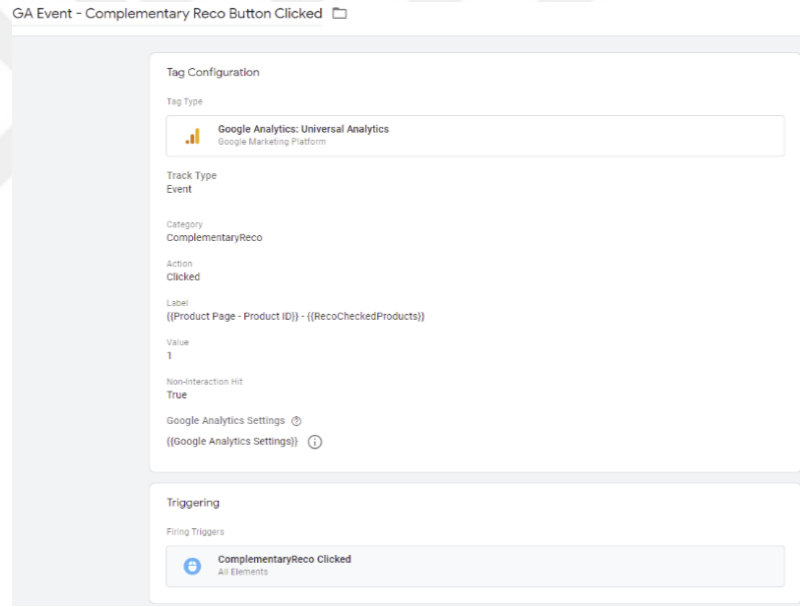
Şekil 4.9. Tava Ürün Grubuna Yönelik Düzenleme Örneği

#### 4.7. GTM ile Google Analytics’e İlgili Alan Bilgilerinin Aktarımı

Google Analytics üzerinden kullanıcıların geliştirilen ürün öneri sisteminin etkisini mevcut sisteme kıyasla değişimini ölçümlemek amacıyla sitede kullanıcıların tamamlayıcı ürünler alanının görüntülenme ve tıklanma sayılarının karşılaştırılması yapılmak istenmektedir. Ürün sayfasında yer alan bu bilgiler Google Tag Manager üzerinden “Event” tanımlamaları yapılarak Google Analytics’e bağlanmaktadır. GTM ile köprü kurularak Google Analytics’e sürekli bir veri akışı sağlanmaktadır. Sitede görüntülenme sayılarını almak için Şekil 4.10.’da, tıklanma sayılarını almak için Şekil 4.11.’de görüldüğü üzere tag’ler oluşturulmuştur. Sayfada ilgili alan %51 ve üzeri oranında görüntülendiği zaman GTM üzerinden event tetiklenecek şekilde Google Analytics’e yazılmaktadır.



Şekil 4.10. Tamamlayıcı ürün önerileri alanı görüntülenme Tag konfigürasyonu



Şekil 4.11. Tamamlayıcı ürün önerileri alanı tıklanma Tag konfigürasyonu

#### 4.8. Çıktıların Web Sitesi ve Mobil Uygulamaya Aktarılması

Ürün öneri sistemine ait tablonun tamamlanmasının ardından her bir ürün için 3 tamamlayıcı ürün önerisi getirilmiştir. Bu ürünler site içerisine aktarılmadan önce ürün

isimlerine karşılık gelen ürün ID'lerine Tablo 4.2.'de (ilk 5 satır) görüldüğü üzere çevrilmiştir.

Tablo 4.2. Ürün Önerilerinin ID olarak düzenlenmesi

| Product ID | Tamamlayıcı 1 | Tamamlayıcı 2 | Tamamlayıcı 3 |
|------------|---------------|---------------|---------------|
| 6207371    | 6208536       | 5900904       | 6223884       |
| 3922486    | 5833330       | 5900904       | 6223884       |
| 6159097    | 23271         | 6165057       | 6223884       |
| 5931869    | 417120        | 6165060       | 6223884       |
| 5895349    | 417120        | 6165060       | 6223884       |

Sonraki adımda Şekil 4.12'de gösterildiği gibi liste haline getirilmektedir. Elde edilen listenin csv formatında çıktısı alınarak web sitesi ve mobil uygulamada ilgili alanı besleyecek şekilde eklenmektedir.

```
liste=[]
a=0
for i in son_hali["Product_Name"]:
    if son_hali.loc[son_hali["Product_Name"]==i, "Tamamlayıcı_1"].reset_index(drop=True).shape[0]!=0:
        ürün1=son_hali.loc[son_hali["Product_Name"]==i, "Tamamlayıcı_1"].reset_index(drop=True)[0]
        liste.append(str(i)+","+str(ürün1))
    if son_hali.loc[son_hali["Product_Name"]==i, "Tamamlayıcı_2"].reset_index(drop=True).shape[0]!=0:
        ürün2=son_hali.loc[son_hali["Product_Name"]==i, "Tamamlayıcı_2"].reset_index(drop=True)[0]
        liste.append(str(i)+","+str(ürün2))
    if son_hali.loc[son_hali["Product_Name"]==i, "Tamamlayıcı_3"].reset_index(drop=True).shape[0]!=0:
        ürün3=son_hali.loc[son_hali["Product_Name"]==i, "Tamamlayıcı_3"].reset_index(drop=True)[0]
        liste.append(str(i)+","+str(ürün3))

liste=[]
a=0
for i in son_hali["Product_Name"]:
    ürün1=son_hali.loc[son_hali["Product_Name"]==i, "Tamamlayıcı_1"].reset_index(drop=True)[0]
    ürün2=son_hali.loc[son_hali["Product_Name"]==i, "Tamamlayıcı_2"].reset_index(drop=True)[0]
    ürün3=son_hali.loc[son_hali["Product_Name"]==i, "Tamamlayıcı_3"].reset_index(drop=True)[0]
    if type(ürün1)!=float:
        liste.append(str(i)+","+str(ürün1))
    if type(ürün2)!=float:
        liste.append(str(i)+","+str(ürün2))
    if type(ürün3)!=float:
        liste.append(str(i)+","+str(ürün3))

ürün1=son_hali.loc[son_hali["Product_Name"]==i, "Tamamlayıcı_1"].reset_index(drop=True)[0]
ürün2=son_hali.loc[son_hali["Product_Name"]==i, "Tamamlayıcı_2"].reset_index(drop=True)[0]
ürün3=son_hali.loc[son_hali["Product_Name"]==i, "Tamamlayıcı_3"].reset_index(drop=True)[0]
liste.append(str(i)+","+str(ürün1))
liste.append(str(i)+","+str(ürün2))
liste.append(str(i)+","+str(ürün3))
```

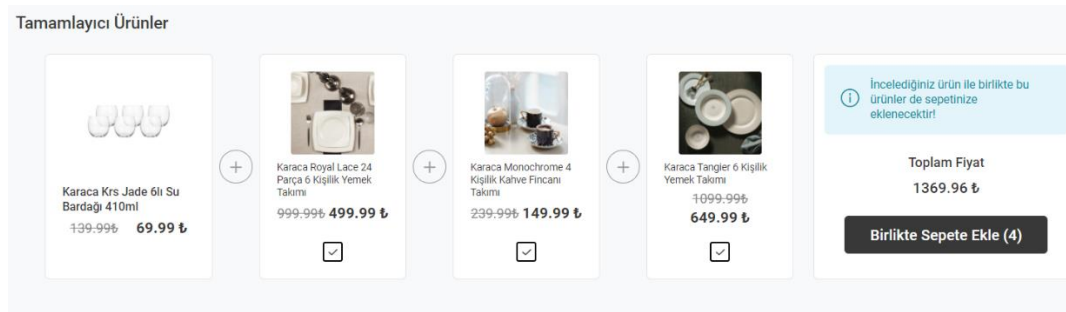
Şekil 4.12. Ürün ID'lerinin liste formatına çevrilmesi

## BÖLÜM 5. ARAŞTIRMA BULGULARI

Ürün öneri sisteminin geliştirilme ihtiyacı ilk olarak mevcut durumda bulunan ürün önerilerinin eksik ve ilgili olmayan ürünlerin gösteriminin tespiti ile ortaya çıkmıştır. Bu kısımda, geliştirilen ürün öneri sisteminin kullanıcı ekranlarının mevcut duruma göre kıyaslanması, tıklanma ve görüntülenme sayılarındaki değişim ve tamamlayıcı olarak satışı yapılan tekli ve set halindeki ürünlerin yüzdesel farklılıkları analiz edilerek sistemin performansı değerlendirilecektir.

### 5.1. Kullanıcı Ekranlarının Karşılaştırılması

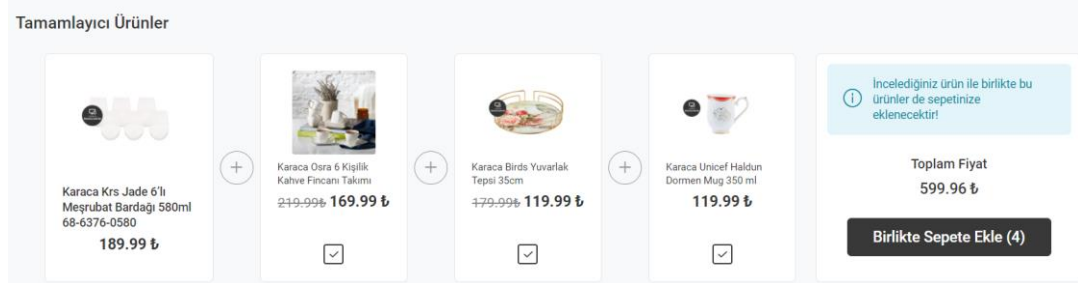
Mevcut durumda önerilen ürünlerin, ürün öneri sisteminde kullanıcılara tavsiye olarak gösteriminde eksik ve mantıksız ürün önerileri gösterdiği tespit edilmiştir. Şekil 5.1.'de gösterildiği gibi düşük fiyat segmentine sahip su bardağı seçmiş bir müşteriye mevcut sistem, yüksek fiyatlı yemek takımı önerebilmekte ve bunun sonucunda da istenen satışların gerçekleşmemesi ve müşteri deneyiminin olumsuz yönde etkilenmesi gerçekleşmektedir. Ayrıca mevcut sistemin bazı ürünlerde de eksik ürün önerileri yaptığı tespit edilmiştir.



Şekil 5.1. Mevcut sistemin ürün önerileri

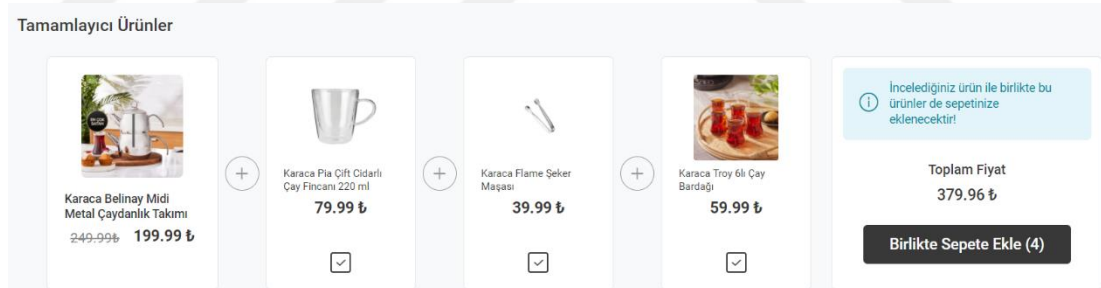
Yeni geliştirilen öneri sistemi mobil uygulama ve e-ticaret sitesinde uygulamaya alındıktan sonra tamamlayıcı ürünleri birbirine benzer, aynı fiyat segmentinde,

istenilen ürün gruplarında ve sepete eklenen bir ürüne 3 ürün önerisini eksiksiz olarak getirdiği gözlemlenmiştir. Şekil 5.2.'de gösterildiği gibi kullanıcıya ilgili ürünlerin fiyat segmentasyonu dikkate alınarak önerilmiş olduğu gözlemlenmektedir.



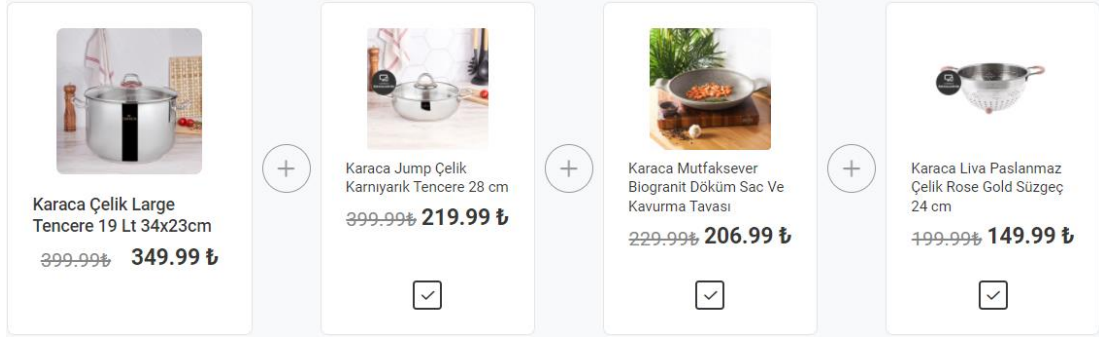
Şekil 5.2. Geliştirilen sistemin ürün önerileri (fiyat segmentasyonu odaklı)

Ürün önerileri yapılırken fiyat, renk, marka vb. parametreler dikkate alınmıştır. Bu kriterlere ek olarak ürünlerin birbirleri ile ilişkili olduğu durumlar ürün grupları bazında değerlendirilerek gösterimi sağlanmıştır. Şekil 5.3.'de gösterildiği gibi çaydanlık ürününe yönelik bu ürün ile birlikte satın alınma durumu yüksek olan çay fincanı, şeker maşası ve çay bardağı gibi ürün önerileri sunulması sağlanmıştır.



Şekil 5.3. Geliştirilen sistemin ürün önerileri (ürün ilişkileri odaklı)

Özellikle züccaciye sektöründe önemli bir kriter olarak bulunan ürün boyutları dikkate alınarak belli bir ölçüde tava, tencere vb. ürün seçen müşterilere farklı ölçülerde ürün gösterimleri yapılarak mevcut duruma göre çok daha iyi sonuçlar verdiği gözlemlenmiştir. Ürün boyutları dikkate alınarak geliştirilen sisteminin ürün önerileri Şekil 5.4.'de verilmiştir.



Şekil 5.4. Geliştirilen sistemin ürün önerileri (ürün boyutları odaklı)

## 5.2. Öneri Sistemi Performansının Google Analytics ile Kıyaslanması

Google Analytics sonuçları incelendiğinde sistemin uygulanmaya başlandığı tarih olan 3 Ocak 2022'den 2 Şubat 2022 tarihine kadar olan görüntülenme ve tıklanma sayıları bir önceki aya göre gün bazında kıyaslanmıştır. Performans karşılaştırılması yapılacak olan bu tarihlerin, firmanın özel olarak tanımladığı satışların yüksek olarak gerçekleştiği bazı özel dönemler (Black Friday, Anneler Günü vb.) haricinde olmasına da dikkat edilmiştir. Bu sayede elde edilen sonuçların öneri sisteminin performansına yönelik yanıltıcı etki göstermesinin önüne geçilmiştir.

Şekil 5.5.'de görüleceği üzere 1 önceki ay ile kıyaslandığında ürün öneri sistemi alanının görüntülenme sayılarında %7,17 artış gözlemlenirken, tıklanma sayıları incelendiğinde %28,93 olarak artış gerçekleşmiştir. Bu sonuçlar neticesinde ürünlerin tıklama sayılarının görüntülenme sayılarına oranla daha yüksek çıkması, ilgili alanda yapılan ürün önerilerinin kullanıcılar için daha uygun olduğunun, geliştirilen sistemin ürün önerilerinin müşterilerin dikkatini çekerek ürün sayfalarına daha yüksek sayıda eriştiğinin göstergesi olarak kabul edilmektedir ve yapılan çalışmanın olumlu yönde katkıda sağladığının kanıtı niteliğindedir.

| <input type="checkbox"/> | Etkinlik İşlemi ?         | Toplam Etkinlik Sayısı ? ↓           |
|--------------------------|---------------------------|--------------------------------------|
|                          |                           | <b>%6,75</b> ↑<br>812.387 ve 761.042 |
| <input type="checkbox"/> | 1. Viewed                 |                                      |
|                          | 03.Oca.2022 - 02.Şub.2022 | <b>802.232</b> (%98,75)              |
|                          | 03.Ara.2021 - 02.Oca.2022 | <b>748.561</b> (%98,36)              |
|                          | <b>Değişiklik %</b>       | <b>%7,17</b>                         |
| <input type="checkbox"/> | 2. Clicked                |                                      |
|                          | 03.Oca.2022 - 02.Şub.2022 | <b>16.139</b> (%1,97)                |
|                          | 03.Ara.2021 - 02.Oca.2022 | <b>12.518</b> (%1,64)                |
|                          | <b>Değişiklik %</b>       | <b>%28,93</b>                        |

Şekil 5.5. Google Analytics Görüntülenme ve Tıklanma Sayıları Karşılaştırması

Öneri sisteminin uygulanmaya başlandığı 3 Ocak 2022 tarihinden itibaren sitede ilgili alanda “Clicked” sayılarında günlük bazda bir önceki aya göre değişimleri kıyaslandığında Şekil 5.6’da görüleceği üzere mevcut sisteme göre her gün için daha iyi performans gösterdiği gözlemlenmiştir. Bir önceki sene ile kıyaslama, ilgili alana ait GTM kodları ile tanımlamaları o dönem olmadığı için yapılamamaktadır. Gelecek dönemlerde elde edilecek veriler neticesinde bir önceki yıla göre veya özel dönemler için (anneler günü, black friday vb.) karşılaştırma yapılabilecektir.



Şekil 5.6. “Clicked” sayılarının bir önceki aya göre günlük değişimi

### 5.3. Öneri Sisteminin Satışa Olan Etkisinin Yüzdesel Değişimi

Çalışmada geliştirilen sistemde bir ürüne ait 3 farklı ürün önerisi sunulmaktadır. Burada yeni sistemin performansı karşılaştırılırken, belirlenen ürüne yönelik elde edilen satış miktarları üzerinden tamamlayıcı ürünlerin beraber satılan ürün sayısı baz alınarak mevcut sisteme göre yüzdesel farkları hesaplanmıştır. Buradaki amaç yeni ürün öneri sistemi oluşturulduktan sonra tavsiye edilen tamamlayıcı ürünlerin performansını tespit etmektir.

Tüm set halinde satılan ürünlerde yeni sistemin mevcut sisteme göre artış oranı %7,62 olarak ölçülürken, tek olarak satılan ürünlerde artış oranı %11,2 olarak hesaplanmıştır. Tablo 5.1’de set halinde satılan ürünlere ait örnek veri grubu, Tablo 5.2’de ise tekli olarak satılan ürünlere ait örnek veri grubu paylaşılmıştır. Her iki tabloda için de ürünlerin seçimi uzman görüşleri neticesinde belirlenen ürünlere ait olmakla birlikte yeni ve eski öneri sisteminde müşterilerin seçimlerinde ne oranda tercih edildiğinin ve mevcut sisteme kıyasla yüzde artışlarının değerleri gösterilmiştir.

Tablo 5.1. Set halinde satılan ürünlere ait örnek veri grubu

| Ürün Adı                                  | Tamamlayıcı Ürünlerin Toplam Satış Oranları |                    | Tamamlayıcı Ürün Satışları Artış Oranı |
|-------------------------------------------|---------------------------------------------|--------------------|----------------------------------------|
|                                           | Yeni Öneri Sistemi                          | Eski Öneri Sistemi |                                        |
|                                           |                                             |                    |                                        |
| Pure 6 kşlk ymt rnd                       | 10,85%                                      | 2,62%              | 8,23%                                  |
| Blendfit 4 in 1 set red gold              | 9,18%                                       | 3,71%              | 5,47%                                  |
| Stea 24 prç ymt                           | 10,79%                                      | 1,66%              | 9,13%                                  |
| Point tree 24 prç ymt                     | 6,58%                                       | 1,34%              | 5,24%                                  |
| Biogranit blackgold 7 prç tencere seti    | 12,71%                                      | 5,16%              | 7,55%                                  |
| Overcut 6 prc bıçak seti                  | 14,87%                                      | 2,81%              | 12,06%                                 |
| Woodland herbal 6 parça servis seti       | 8,22%                                       | 1,31%              | 6,91%                                  |
| Bio diamond master 3 prc tencere seti     | 12,09%                                      | 3,42%              | 8,67%                                  |
| Karaca pop art 12 prç çay seti            | 14,62%                                      | 4,19%              | 10,43%                                 |
| Kh rubi siyah çift kişilik ytk örtüsü set | 13,09%                                      | 3,47%              | 9,62%                                  |

Tablo 5.2. Tekli olarak satılan ürünlere ait örnek veri grubu

| Ürün Adı                                | Tamamlayıcı Ürünlerin Toplam Satış Oranları |                    | Tamamlayıcı Ürün Satışları Artış Oranı |
|-----------------------------------------|---------------------------------------------|--------------------|----------------------------------------|
|                                         | Yeni Öneri Sistemi                          | Eski Öneri Sistemi |                                        |
|                                         |                                             |                    |                                        |
| Zeugma 25cm servis tabağı               | 15,48%                                      | 4,89%              | 10,59%                                 |
| Mesopotamia bakır pilav tenceresi 18 cm | 18,55%                                      | 3,71%              | 14,84%                                 |
| Elvis 8 şef bıçağı                      | 16,86%                                      | 3,37%              | 13,49%                                 |
| Frida kahlo kase                        | 9,75%                                       | 2,49%              | 7,26%                                  |
| Lulya dekoratif tabak                   | 17,85%                                      | 8,98%              | 8,87%                                  |
| Mare pasta tabağı                       | 16,90%                                      | 3,57%              | 13,33%                                 |
| Topic tribal 280ml saklama kabı         | 9,63%                                       | 1,49%              | 8,14%                                  |
| Berry dalgalı kek kalıbı 28 cm          | 14,89%                                      | 4,17%              | 10,72%                                 |
| Black gold küçük sunum tabağı           | 15,32%                                      | 2,53%              | 12,79%                                 |
| Karaca spin kesme tahtası-s             | 17,55%                                      | 2,64%              | 14,91%                                 |

Örnek veri gruplarında ve toplam veri setinde tek olarak satılan ürünler incelendiğinde geliştirilen ürün öneri sisteminin mevcut duruma kıyasla daha iyi sonuçlar verdiği gözlemlenmiştir. Burada tekli olarak satılan ürünlerin fiyat ortalamalarının daha düşük olması set halinde satılan ürünlere göre yüksek performans göstermesinin nedenlerinden biri olarak kabul edilmektedir.

## BÖLÜM 6. TARTIŞMA VE SONUÇ

Öneri sistemleri son zamanlarda e-ticaret alanının hızla gelişmesiyle birlikte öne çıkan bir uygulama olarak görülmektedir. Uygulandığı firmanın dinamiklerine göre hazırlanması son derece önem arz etmektedir. Yapılan çalışmada ağırlıklı olarak züccaciye ve ev tekstili alanında satış yapan bir perakende firmasına yönelik ürün öneri sistemi geliştirilmiştir. Geliştirilen ürün öneri sisteminin mevcut duruma kıyasla web sitesi ve mobil uygulamada ilgili alanın görüntülenme ve tıklanma sayıları Google Analytics'den elde edilen veriler ile karşılaştırılmıştır. Yeni sistemin mevcut duruma kıyasla daha iyi performans gösterdiği tespit edilmiştir. Ayrıca ürünlere önerilen tamamlayıcı 3 tavsiye ürünün beraber satış adetleri üzerinden python aracılığıyla yüzdesel farkları hesaplanmıştır. Burada da tekli satılan ürünlerde %11,2 ve set halinde satılan ürünlerde %7,62 artış sağlanarak olumlu yönde katkı sağladığı gözlemlenmiştir.

Oluşturulan ürün öneri sisteminin etkilerinin daha iyi ölçümlenebilmesi için uzun süreli olarak incelenmesi gerekmektedir. Günümüzde etkisini fazlasıyla hissettiğimiz pandemi dönemi ve döviz kuruna bağlı olarak yaşanan fiyat değişiklikleri gibi durumlar satın alma davranışlarını etkilediği için farklı dönemlere ait karşılaştırmalar, oluşturulan öneri sisteminin performansının ölçülmesini ve geliştirmeye açık olan yönlerinin daha net bir şekilde ortaya çıkmasını sağlayacaktır.

Ayrıca gerçek ortamda uygulanabilir durumda olması sebebiyle sürekli gözlem yapma imkanı bulunmaktadır. Yapılacak olan değerlendirmeler sonucu ürün önerilerinin anlam bakımından hatalı bulunması veya uzman görüşlerinin olumsuz olması durumunda verilerde düzenleme, yeni ürün grupları oluşturma ve gerekli filtreleme işlemlerinin yapılmasına esneklik sağlayabilmektedir.

Gelecek çalışmalarda RFM analizi uygulanarak müşteri segmentlerinin oluşturulması ve ürün temelli önerilere destek olacak şekilde müşteri davranışlarının dahil edildiği hibrit bir ürün öneri sistemi geliştirilmesi amaçlanmaktadır. Hibrit sisteme uygun farklı algoritmalar kullanılarak mevcut çalışma ile performans karşılaştırılması yapılabilecektir.



## KAYNAKLAR

- [1] Adomavicius, G., Tuzhilin, A. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. IEEE Transactions on Knowledge and Data Engineering, vol. 17, no. 6. doi: 10.1109/TKDE.2005.99, 2005.
- [2] Sarwar, B., Karypis, G., Konstan, J., Riedl, J. 2001. Item-based collaborative filtering recommendation algorithms. Proceedings of the tenth international conference on World Wide Web - WWW '01, New York., 285-295.
- [3] Gomez-Urbe, C. A., Gomez-Urbe, Hunt, N. The netflix recommender system: Algorithms, business value, and innovation. ACM Trans. Manag. Inf. Syst., vol. 6, no. 4, doi: 10.1145/2843948, 2015.
- [4] Smith, B., Linden, G. Two decades of recommender systems at Amazon.com. IEEE Internet Comput., vol. 21, no. 3, doi: 10.1109/MIC.2017.72, 2017.
- [5] Qin, S., Menezes, R., Silaghi, M. A recommender system for youtube based on its network of reviewers. 2010, doi: 10.1109/SocialCom.2010.53.
- [6] M. Schedl, H. Zamani, C. W. Chen, Y. Deldjoo, and M. Elahi, "Current challenges and visions in music recommender systems research," Int. J. Multimed. Inf. Retr., vol. 7, no. 2, 2018, doi: 10.1007/s13735-018-0154-2.
- [7] Bozkurt, M.B., Acı, Ç.İ. 2021. Öneri algoritmalarının film önerme problemi üzerinde karşılaştırılması: Movielens örneği. Bilgisayar Bilimleri ve Teknolojileri Dergisi, 2(2); 36-42
- [8] Yalçın. E, "İşbirlikçi Filtreleme Algoritmalarının Çok-Beğenilen Ürünlere Yönelik Yanlılığı", BŞEÜ Fen Bilimleri Dergisi 8(1), 279-291, 2021.
- [9] Dundar, A., Gorgulu Kakisim, A. Kıyafet öneri sistemi için giyim metaverilerine dayalı temsil öğrenimi. Avrupa Bilim ve Teknoloji Dergisi Özel Sayı 29, S. 105-110. 10, 2021.
- [10] Choi, Y. K., Kim, S. K., An auxiliary recommendation system for repetitively purchasing items in E-commerce. 2014. doi: 10.1109/BIGCOMP.2014.6741415.

- [11] Alharthi, D. Inkpen, and S. Szpakowicz, "A survey of book recommender systems," J. Intell. Inf. Syst., vol. 51, no. 1, 2018, doi: 10.1007/s10844-017-0489-9.
- [12] Uçar, M. 2021. E-ticaret siteleri için bulanık mantık ve makine öğrenmesi tabanlı bir öneri sistemi. Sakarya Üniversitesi, Fen Bilimleri Enstitüsü, Bilgisayar ve Bilişim Mühendisliği, Yüksek Lisans Tezi.
- [13] Yıldız, E. 2018. Müşteri segmentasyonu odaklı bütünleşik bir ürün öneri sistemi: moda perakendesi sektöründe bir uygulama. Yıldız Teknik Üniversitesi, Fen Bilimleri Enstitüsü, Endüstri Mühendisliği, Yüksek Lisans Tezi.
- [14] Erdogdu, E. 2020. Banka verileriyle ürün öneri sistemi. Bahçeşehir Üniversitesi, Fen Bilimleri Enstitüsü, Bilgi Teknolojileri, Yüksek Lisans Tezi.
- [15] Abdullah, W. 2019. E-ticaret için ürün tavsiye sistemi geliştirilmesi. İstanbul Ticaret Üniversitesi, Fen Bilimleri Enstitüsü, Bilgisayar Mühendisliği Bölümü, Yüksek Lisans Tezi.
- [16] Guler, S. 2019. Öneri sistemleri ve e-ticarete öneri sistemlerinin kullanımı. Sakarya Üniversitesi, Fen Bilimleri Enstitüsü, Bilgisayar ve Bilişim Mühendisliği, Yüksek Lisans Tezi.
- [17] Batmaz, B. 2018. Yükseköğretimde öneri sistemlerine dayalı ders seçme modeli. Eskişehir Osmangazi Üniversitesi, Fen Bilimleri Enstitüsü, İstatistik Bilgi Sistemleri, Doktora Tezi.
- [18] Keleş, Ü., Duru, N. 2021. Metin Benzerliği algoritmaları ile veri tekilleştirme: oteller veri tabanında bir uygulama. Türkiye Bilişim Vakfı Bilgisayar Bilimleri ve Mühendisliği Dergisi, Cilt:5, Sayı:1.
- [19] Jin, X., Zhang, S., Liu, J., Guan. 2017. H. Research on similarity detection of massive text based on semantic fingerprint, Proceedings of Science.
- [20] Vijaymeena, M.K., Kavitha, K. 2016. A survey on similarity measures in Text Mining. Machine Learning and Applications: An International Journal (MLAIJ), cilt 3, no. 1, pp. 19-28.
- [21] Recchia, G., Louwerse, M. 2013. A comparison of string similarity measures for toponym matching, ResearchGate.

- [22] Öztemiz, F., Karcı, A. akademik yazarların yayınları arasındaki ilişkinin sosyal ağ benzerlik yöntemleri ile tespit edilmesi. Uludağ Üniversitesi Mühendislik Fakültesi Dergisi, cilt 25, no. 1, pp. 591- 608, 2020.
- [23] Ülkü, B. K. 2019. Panoramik radyografi görüntülerinde maksiller ve mandibular yapıların sınırlarının belirlenmesi için yarı gözetimli bir metot. TOBB Ekonomi ve Teknoloji Üniversitesi Fen Bilimleri Enstitüsü Yüksek Lisans Tezi.
- [24] Hashemi, S. R., Salehi, S. S. M., Erdogmus, D. tversky as a loss function for highly unbalanced image segmentation using 3d fully convolutional deep networks, 2018.
- [25] Ionescu, R. T., Popescu, M. Knowledge Transfer between Computer Vision and Text Mining, Switzerland: Springer, 2016.
- [26] [https://en.wikipedia.org/wiki/Levenshtein\\_distance](https://en.wikipedia.org/wiki/Levenshtein_distance)., Erişim Tarihi: 16.09.2021.
- [27] Aung, K. M. M., Htwe, A. N. 2019. Comparison of Levenshtein Distance Algorithm and Needleman Wunsch Distance Algorithm for string matching. National Journal of Parallel and Soft Computing, cilt 1, no. 1, pp. 209-213.
- [28] Christen, P. A Comparison of Personal Name Matching: Techniques and Practical Issues,» 2006.
- [29] Khudeer, H., Erbay, H. 2021. Hibrit Karga-Genetik Algoritmasını kullanarak 3 boyutlu kutu paketleme problemi çözme. Veri Bilimi Dergisi, cilt 4, no. 1, pp. 8-22.
- [30] Christen, P. a comparison of personal name matching: techniques and practical issues, 2006.
- [31] Ilyankou, I. Comparison of Jaro-Winkler and Ratcliff/Obershelp algorithms in spell check, IB Extended Essay Computer Science, 2014.
- [32] Ionescu, R. T., Popescu, M. Knowledge Transfer between Computer Vision and Text Mining”, Switzerland: Springer, 2016.
- [33] Li, Z. 2006. Matris ayrışımı. İstanbul Üniversitesi, Sosyal Bilimler Enstitüsü. Ekonometri Anabilim Dalı, Yüksek Lisans Tezi.

- [34] Abidin, T.F., Yusuf, B., Umran, M. Singular Value Decomposition for Dimensionality Reduction in Unsupervised Text Learning Problems. 2nd International Conference on Education Technology and Computer (ICETC), 4: 422–426 (2010).
- [35] Han, J., Pei, J., Kamber, M. “Data mining: concepts and techniques”, 3rd Edition, Morgan Kaufmann Publishers, Vol.7 No.6, 2011.



## ÖZGEÇMİŞ

Adı Soyadı : Bilal Erdemir

### ÖĞRENİM DURUMU

| Derece        | Eğitim Birimi                                                          | Mezuniyet Yılı |
|---------------|------------------------------------------------------------------------|----------------|
| Yüksek Lisans | Sakarya Üniversitesi / Fen Bilimleri Enstitüsü / Endüstri Mühendisliği | Devam ediyor   |
| Lisans        | Sakarya Üniversitesi / Mühendislik Fakültesi / Endüstri Mühendisliği   | 2018           |
| Lise          | Malatya Koleji Anadolu Lisesi                                          | 2013           |

### İŞ DENEYİMİ

| Yıl        | Yer                   | Görev         |
|------------|-----------------------|---------------|
| 2021-Halen | Karaca Züccaciye A.Ş. | Veri Analitik |
| 2020-2021  | Karaca Züccaciye A.Ş. | Ürün Planlama |

### YABANCI DİL

İngilizce

### ESERLER

1.Girişimcilik Seviyesini Etkileyen Kriter Ağırlıklarının SWARA Yöntemi ile Belirlenmesi (Bildiri)

### HOBİLER

Geleneksel Okçuluk, Satranç, Ebru Sanatı