Market Basket Analysis

Hayrettin Kaan Özsoy
Bilgisayar Mühendisliği Bölümü
TOBB Ekonomi ve Teknoloji Üniversitesi
Ankara, Türkiye
hozsoy@etu.edu.tr

Abstract **- Bu çalışmada, perakende sektöründe müşteri davranışlarını anlamak amacıyla gerçekleştirilen Market Sepeti Analizi (Market Basket Analysis) ele alınmıştır. Amacımız, bir e-ticaret platformundaki işlem verilerini analiz ederek sıkça birlikte satın alınan ürünler arasındaki ilişkileri keşfetmektir. Bu doğrultuda, Apriori ve FP-Growth algoritmaları kullanılarak, sık ürün kümeleri ve ilişki kuralları çıkarılmıştır. FP-Growth algoritması, büyük veri setlerinde daha verimli olmasıyla öne çıkmaktadır. Veri analizi sürecinde, müşteri segmentasyonu yapılarak farklı alışveriş davranışlarına sahip müşteri grupları belirlenmiştir. Bu segmentasyon, hedeflenmiş pazarlama stratejileri geliştirilmesine ve kaynakların daha etkili kullanılmasına olanak tanımaktadır.**

Bu rapor, problem tanımı, veri analizi, tanımlayıcı analiz, öngörücü analiz ve ayrıştırıcı analiz başlıkları altında detaylı bir şekilde sunulmuştur.

Dizin Terimleri – Data Mining, Market Basket Analysis, Association Rule Mining, Consumer Segmentation

I. GİRİŞ

Günümüzde satış sektöründe rekabetin artması ve müşteri beklentilerinin değişmesiyle birlikte, veri madenciliği ve analitik yöntemlerin kullanımı giderek önem kazanmaktadır. Özellikle müşterilerin alışveriş alışkanlıklarını anlamak ve bu doğrultuda stratejik kararlar almak, işletmeler için kritik bir rol oynamaktadır. Bu bağlamda, *Market Basket Analysis (Market Sepeti Analizi)* olarak bilinen yöntem, müşterilerin bir arada satın aldığı ürünlerin belirlenmesi ve bu ürünler arasındaki ilişkilerin incelenmesi için yaygın olarak kullanılan bir tekniktir.

Market Sepeti Analizi, müşterilerin sepetlerindeki ürünler arasında sıkça birlikte satın alınan ürün çiftlerini veya gruplarını tespit etmeyi amaçlar. Bu analiz satıcılara; çapraz satış stratejileri geliştirme, ürün yerleşimlerini optimize etme ve kişiselleştirilmiş pazarlama kampanyaları oluşturma konularında yardımcı olabilmektedir.

Projenin temel amacı, bir e-ticaret veri seti üzerinde, müşterilerin birlikte satın aldığı ürünleri belirleyerek satış ve pazarlama stratejileri geliştirmektir. Bu doğrultuda, yaygın olarak kullanılan Apriori ve FP-Growth algoritmaları uygulanmıştır. Apriori algoritması, belirli bir destek değeri eşiği altında sıkça bir arada görülen ürün kombinasyonlarını bulmaya yarayan bir tekniktir. FP-Growth algoritması ise büyük veri setlerinde daha verimli çalışarak bellekte daha az yer kaplayan bir veri yapısı kullanarak benzer sonuçlar üretir.

Proje boyunca kullanılan metrikler arasında destek (support), güven (confidence) ve lift değerleri yer almaktadır. Bu metrikler, bulunan kuralların geçerliliğini ve kuralların satıcılar için ne kadar anlamlı olabileceğini değerlendirmek amacıyla kullanılmaktadır.

Bu çalışma ile işletmelerin müşteri segmentasyonlarını daha iyi yapabilmesi, kişiselleştirilmiş pazarlama kampanyaları düzenleyebilmesi ve nihayetinde gelirlerini artırabilmesi hedeflenmiştir.

II. VERİ ANALİZİ

Bu aşamada, veri setinde bulunan eksik ve yinelenen kayıtlar temizlenmiştir, veri türleri doğru biçimlere dönüştürülmüştür ve veri seti üzerinde temel istatistiksel analizler gerçekleştirilmiştir.

A. Veri Seti

Ürünlerin hangi sıklıkla birlikte satın alındığını ve müşteri alışkanlıklarını analiz etmek için kullanılan veri seti, yaklaşık 500 bin satırdan oluşup "BillNo, Itemname, Quantity, Date, Price, CustomerID, Country" özniteliklerini içermektedir.

B. Eksik ve Yinelenen Verilerin İşlenmesi

Veri seti incelendiğinde, bazı satırlarda eksik değerler bulunduğu tespit edilmiştir. Bazı satırlarda "CustomerID" değerleri null değer içermekteydi. Bu eksik veriler, analiz sürecini olumsuz etkilememesi için uygun yöntemlerle doldurulmuştur. Ayrıca, veri setindeki yinelenen kayıtlar da tespit edilip, veri setinden çıkarılmıştır.

C. Veri Türlerinin Dönüştürülmesi

Veri türlerinin doğru bir şekilde dönüştürülmesi, analiz sonuçlarının doğruluğu açısından büyük önem taşımaktadır. Başlangıçta tüm öznitelik değerleri "object" tipindedir. Bu aşamada; tarih, fiyat ve miktar gibi sayısal veriler uygun formatlara dönüştürülmüştür.

```
BillNo              int32
Itemname           object
Quantity            int32
Date       datetime64[ns]
Price             float64
CustomerID          Int64
Country            object
```

Ayrıca, veri setinde yer alan *Quantity* ve *Price* özniteliklerinin negatif ve sıfır olan değerleri filtrelenerek analiz sürecinde hatalı sonuçlarla karşılaşma riskinin önüne geçilmiştir.

D. Korelasyon Matrisi

Bu çalışmada, *Price* ve *Quantity* gibi çeşitli öznitelikler arasındaki ilişkileri incelemek amacıyla başlangıçta korelasyon matrisi kullanılmıştır. Ancak, korelasyonların, birlikte sıkça satın alınan ürünler arasındaki ilişkileri anlamada anlamlı bir farkındalık sağlamadığı görülmüştür. Öznitelikler arası ilişkiler neredeyse sıfıra yakın değerler vermiştir.

[Şekil 1: Korelasyon Matrisi]

E. Undersampling

Veri setinde, alışverişlerin hangi ülkelerde yapıldığını gösteren *Country* özniteliği bulunmaktadır. Ülke bazında verilerin dağılımı incelendiğinde, Birleşik Krallık'ın diğer ülkelere kıyasla çok fazla veriye sahip olduğu görülmüştür. Bu durum, veri setinde dengesizliğe neden olmuş ve analiz sonuçlarını etkileyebilecek bir faktör olduğu tespit edilmiştir.

[Şekil 2. Country öznitelik değerlerinin dağılımı]

Bu dengeyi sağlamak amacıyla, Birleşik Krallık verilerinden rastgele seçilen bir alt kümenin boyutu, diğer ülkelerin toplam veri sayısına eşit olacak şekilde ayarlanarak *undersampling* işlemi uygulanmıştır.

[Şekil 3. Undersampling sonrası Country öznitelik değerlerinin dağılımı]

Ancak, bu işlem *tahminsel (predictive)* ve *ayrıştırıcı (discriminative)* analiz aşamalarında kullanılmıştır. *Tanımlayıcı (descriptive)* analiz aşamasında, önemli olan *associative* kuralların gözden kaçırılmaması için orijinal veri seti kullanılmıştır. Bu sayede, veri setindeki kritik ilişkilerin korunması sağlanmış ve analizin doğruluğu artırılmıştır.

F. Veriyi İkilileştirme (Binarization)

Bu adımda verilerin, her bir ürün satın alımını "var" ya da "yok" olarak ifade eden ikili (binary) bir formata dönüştürülmesi için *One-hot Encoding* yöntemi kullanılmıştır.

Veri analizi aşamasında elde edilen bulgular, projede kullanılacak olan Apriori, FP-Growth ve ECLAT algoritmaları ile gerçekleştirilecek birliktelik kuralı çıkarma gibi analizler için bir temel oluşturmuştur. Bu analizler, müşterilerin alışveriş davranışlarını daha derinlemesine incelemek ve pazarlama stratejilerini optimize etmek için kullanılacaktır.

III. DESCRIPTIVE (TANIMLAYICI) ANALİZ

Bu bölümde, Market Basket Analysis sürecinde kullanılan Apriori ve FP-Growth algoritmalarıyla gerçekleştirilen tanımlayıcı (descriptive) analiz adımları açıklanacak ve elde edilen sonuçlar değerlendirilecektir.

A. Apriori

Apriori algoritması, market basket analysis işlemlerinde en yaygın kullanılan yöntemlerden biridir. Bu algoritma, sık rastlanan ürün gruplarını bulmak için iteratif bir yaklaşım kullanır. İlk olarak, her bir ürünün tek başına satılma sıklığı (support) hesaplanır ve belirli bir destek (support) değeri eşik olarak kabul edilir (minimum support). Sonraki aşamalarda, bu eşik değerin üzerindeki ürün kombinasyonları genişletilerek sık rastlanan ürün grupları elde edilir.

B. FP-Growth

Frequent Pattern Growth algoritması, Apriori algoritmasının etkin bir alternatifidir. Bu algoritma, sıkça rastlanan öğe kümelerini bulmak için veriyi sık rastlanan desen ağacı (*Frequent Pattern Tree - FP-Tree*) adı verilen bir yapı içerisinde organize eder. Bu yapı sayesinde, veri tekrarlamalarından kaçınarak çok daha hızlı bir şekilde sık rastlanan ürün kümeleri keşfedilir.

C. Ölçüm Metrikleri

1. *Support (Destek):* Öğe setinin veri setinde ne sıklıkla görüldüğünün bir göstergesidir.

$$\text{Support}(A) = \frac{\text{A setini içeren işlem sayısı}}{\text{Toplam işlem sayısı}}$$

2. *Confidence (Güven):* Bu metrik, kuralın doğru olma olasılığını temsil eder. Hem X hem de Y içeren işlem sayısının, X içeren işlem sayısına bölünmesiyle hesaplanır.

$$\text{Confidence}(A \rightarrow B) = \frac{\text{Support}(A \cup B)}{\text{Support}(A)}$$

3. Lift: X ve Y'nin bağımsızlığına kıyasla, ikisi arasındaki ilişkinin derecesini temsil eder. İki öğe arasındaki bağımsızlık derecesini gösterir.

$$\text{Lift}(A \rightarrow B) = \frac{\text{Confidence}(A \rightarrow B)}{\text{Support}(B)}$$

Lift değerinin 1'den büyük olması, X ve Y öğelerinin birlikte meydana gelme olasılığının, bu öğelerin bağımsız olarak meydana gelme olasılığından daha yüksek olduğunu gösterir.

4. Antecedent Support: Kurala göre, önce gelen öğe ya da öğe kümesinin veri setinde görülme sıklığını gösterir.
5. Consequent Support: Kurala göre, sonra gelen öğe ya da öğe kümesinin veri setinde görülme sıklığını gösterir.
6. Leverage: İki öğe arasındaki bağımsız olmayan ilişkileri tanımlamak için kullanılır. Leverage değeri ne kadar yüksekse, kuralların veri setinde o kadar etkili olduğu söylenebilir.

7. Conviction: Antecedent'in gerçekleştiği durumlarda, consequent'in gerçekleşmemesi olasılığına dayanarak bir kuralın ne kadar güçlü olduğunu ifade eder. Conviction değeri 1'den büyükse, bu durum antecedent'in consequent'in yokluğunu engellediğini gösterir. Yani, conviction ne kadar yüksekse, kuralın gücü de o kadar fazladır.
8. Zhang's Metric: Özellikle düşük support veya confidence değerlerine sahip kuralları değerlendirmek için kullanılır, daha dengeli bir değerlendirme sunar.

Apriori ve FP-Growth algoritmaları sonucunda elde edilen ürün ilişkileri (association rules), yukarıda belirtilen metriklere göre değerlendirilir ve en anlamlı olanları seçilir.

D. *Bellek Kullanımı Sorunu ve Çözümü*

Projenin bu aşamasında, Apriori algoritması uygulanırken büyük veri setleri üzerinde çalışmanın getirdiği bellek kullanımı sorunlarıyla karşılaşıldı. Bu tür durumlarda, verinin boyutu ve minimum destek (min_support) değeri, bellek kullanımını önemli ölçüde etkilemektedir. Apriori algoritmasının tüm öğe kombinasyonlarını hesaplamaya çalışması, bu tür büyük veri setlerinde bellek yetersizliği sorunlarına yol açabilmektedir.

1. Minimum Destek Değerini Ayarlamak:
   İlk olarak, min_support değeri 0.01 olarak belirlendi. Bu durum, daha fazla öğe kümesinin analiz edilmesine yol açarak bellek kullanımını artırdı. Üzerinde çalıştığımız büyük veri setinde support değerleri düşük oluğu için, min_support değerinin 0.5 gibi yüksek bir değere ayarlanması, analiz edilen öğe kümesi sayısını azaltarak birçok anlamlı ilişkinin gözden kaçmasına sebep olmuştur.

2. low_memory Parametresi:
   Bellek kullanımı sorununu çözmek için *low_memory=True* parametresi kullanıldı. Bu parametre, veri kümesinin parçalara ayrılarak işlenmesini sağlamaktadır. Böylece bellek tüketimi optimize edilerek büyük veri setleri üzerinde algoritmanın uygulanabilirliği artırılmaktadır.

E. *Kullanılacak Algoritmanın Seçimi ve Birliktelik Kurallarının Çıkarımı*

Apriori ve FP-Growth algoritmaları kullanılarak sık rastlanan öğe kümeleri elde edildikten sonra, bu kümeler arasındaki ilişki kuralları *mlxtend* kütüphanesinin *association_rules* fonksiyonu ile çıkartıldı. Bu adımda, elde edilen sık rastlanan öğe kümeleri kullanılarak belirli bir güven (confidence) veya lift eşiğine göre anlamlı ilişki kuralları çıkarılmıştır.

Apriori algoritmasının çalışma süresi yaklaşık iki dakika sürerken FP-Growth algoritmasının çalışma süresi yaklaşık 30 saniye sürmüştür. Burada, FP-Growth algoritmasının, Apriori algoritmasına göre genellikle büyük veri kümeleri üzerinde daha hızlı çalıştığı ve bellek kullanımı açısından daha verimli olduğu tespit edilmiştir. FP-Growth algoritması ile elde edilen kurallar, Apriori algoritması ile çıkarılan kurallarla karşılaştırıldığında %89 oranında bir benzerlik göstermiştir. Bu benzerlik, hangi algoritmanın kullanıldığının önemli bir kayba yol açmayacağını, daha verimli bir şekilde çalışan FP-Growth algoritmasının tercih edilmesi gerektiğini göstermiştir.

F. *Zorluklar ve Performans Sorunları*

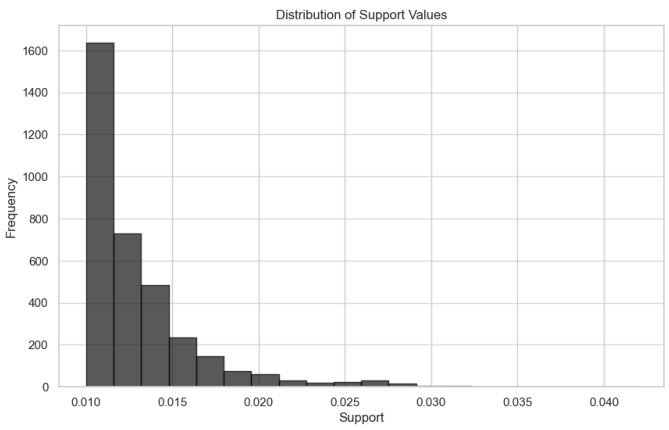
Apriori, FP-Growth ve ECLAT algoritmalarını kullanarak market sepeti analizi gerçekleştirilmesi planlanmıştır. Apriori ve FP-Growth algoritmaları, veri seti üzerinde başarılı bir şekilde çalıştırılmış ve makul sürelerde sonuçlar elde edilmiştir. Ancak, ECLAT algoritmasının uygulanması sırasında ciddi performans sorunları yaşanmıştır.

pyECLAT kütüphanesi kullanılarak gerçekleştirilen ECLAT algoritmasının çalışması yaklaşık dört saat sürmesine ragmen sonuçlar elde edilememiştir ve işlem durdurulmak zorunda kalınmıştır. Ayrıca, multiprocessing (çoklu işlem) uygulandığında da aynı performans sorunları devam etmiştir.

Bu durum, ECLAT algoritmasının bu tür büyük veri setleri için uygun bir çözüm olmadığını göstermiştir. Alternatif olarak, daha verimli çalışan Apriori ve FP-Growth algoritmalarının bu tür analizlerde tercih edilmesi gerektiği gözlemlenmiştir.

G. *Sonuçların Yorumlanması ve Grafik Analizi*

Elde edilen support değerlerinin dağılımı, verinin büyük kısmının düşük support değerlerine sahip olduğunu göstermektedir. Çok az sayıda kural, daha yüksek support değerine sahiptir. Bu kurallar, veri setinde daha sık gözlemlenen kurallardır.



Düşük support (destek) değerlerinde confidence (güven) değerleri geniş bir aralıkta değişmektedir. Bu durum, nadir görülen kombinasyonların bazen çok güvenilir, bazen ise güvenilmez olabileceğini gösterir. Yüksek support değerlerinde ise genellikle daha yüksek confidence değerleri gözlemlenir, bu da bu kombinasyonların daha güvenilir olduğunu gösterir.

[Şekil 4. Support – Confidence Dağılım Grafiği]

Düşük support değerlerinde lift değeri geniş bir dağılıma sahipken, yüksek support değerlerinde lift daha düşük seviyelerde kalmaktadır. Bu durum, sık görülen kombinasyonların genellikle daha az anlamlı (daha düşük lift) olduğunu gösterir.

[Şekil 5. Support – Lift Dağılım Grafiği]

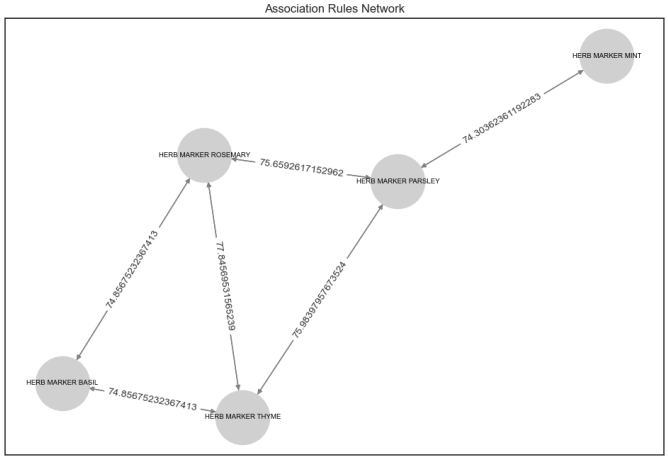
Confidence ve lift arasında pozitif bir ilişki gözlemlenir. Yüksek confidence değerine sahip kurallar genellikle daha yüksek lift değerleriyle ilişkilidir, bu da daha güvenilir kuralların aynı zamanda daha anlamlı olabileceğini gösterir.

[Şekil 6. Confidence – Lift Dağılım Grafiği]

Antecedent (öncül) ve consequent (sonuç) support değerleri arasında belirgin bir doğrusal ilişki gözlemlenmemektedir. Bu durum, öğelerin birlikte görülme olasılıklarının bağımsız olduğunu gösterebilir.

[Şekil 7. Antecedent Support – Consequent Support Dağılım Grafiği]

"Association Rules Network" grafı ile sık kullanılan öğeler arasındaki ilişkiler görselleştirilmiştir. Grafın düğümleri ürünleri, kenarları ise ürünlerin arasındaki lift değerlerini temsil etmektedir.



Market Basket Analysis sürecinde elde edilen kuralların genel özelliklerine bakıldığında aşağıdaki çıkarımlar elde edilmiştir:

$$d(p,q) = \sqrt{\sum_{i=1}^{n}(q_i - p_i)^2}$$

p ve q, n boyutlu uzayda iki nokta.

p_i ve q_i, bu noktaların i-inci bileşenleri.

$d(p,q)$, p ve q noktaları arasındaki Euclidean mesafesidir.

1. *Support değerlerinin standart sapması*nın düşük olması (0.0036), kuralların destek değerlerinin büyük ölçüde birbirine yakın olduğunu göstermektedir. Bu durum, kuralların genel olarak ürünlerin birlikte satın alınma olasılıklarının benzer olduğunu ifade eder.
2. *Confidence değerleri* arasında *yüksek bir varyasyon* (std=0.19) gözlemlenmektedir. Bu durum, bazı kuralların çok güvenilir olduğunu (yüksek confidence), bazılarının ise daha az güvenilir olduğunu (düşük confidence) gösterir.
3. Lift değerlerinin ortalaması 10.68 olarak belirlenmiştir. Bu, kuralların, ürünlerin birlikte satın alınma olasılığını yaklaşık 10 kat artırdığını göstermektedir. Yani, belirli ürün kombinasyonlarının birlikte satın alınma olasılığı, bu ürünlerin bağımsız olarak satın alınma olasılığından çok daha yüksektir.

IV. ÖNGÖRÜCÜ (PREDICTIVE) ANALİZ

Bu bölümde, Market Basket Analysis projesinde *K-Nearest Neighbors (KNN)* algoritmasını kullanarak bir müşteri için ürün önerisi yapılması işlemi ele alınmaktadır.

A. KNN Algoritması

KNN algoritması, bir veri noktasının yakınındaki diğer veri noktalarını (komşularını) değerlendirerek sınıflandırma veya regresyon kararları almaktadır. KNN'de 'k' sayısı, algoritmanın değerlendireceği en yakın komşu sayısını ifade etmektedir. Bu komşular, *Euclidean mesafesi* gibi bir mesafe metriği kullanılarak belirlenir.

B. Ürün Tavsiye Sistemi

Burada KNN algoritması, belirli bir müşteri için önerilecek ürünleri tespit etmek amacıyla kullanılmaktadır.

1. Her müşteri için bir vektör oluşturularak satin alınan ürünler ikili (binary) hale getirildi (Ürün satin alınmışsa 1, yoksa 0 olacak şekilde değerler atanır). Bu sayede, müşteriler arasında benzerlik ölçümleri yapılabilir hale geldi.
2. Bir müşterinin hangi ürünleri alabileceğini tahmin edebilmek için, bu müşteriye en yakın olan 'k' sayıda müşterinin satın aldığı ürünler dikkate alınmaktadır.

Her bir müşterinin en yakın komşuları belirlenerek her birinin alışveriş verileri toplanıp ürün sıklıklarına göre sıralanır.

Bu işlemlere ek olarak; sorgu yapılan müşteriye yeni ürünler önerilmesini sağlamak amacıyla, öneri listesinde daha önceden aldığı ürünler bulunuyorsa bunlar listeden çıkarılmaktadır.

[Şekil 8. Predictive Analysis – Ürün Tavsiye Sistemi Kod Parçası]

KNN algoritması sayesinde, herhangi bir model eğitimine ihtiyaç duyulmadan veri noktalarının komşuluk ilişkilerine dayalı olarak hızlı ve etkili öneriler sunulabilmiştir. Ayrıca, belirli bir müşteriye en yakın diğer müşterilerin tespiti yapılarak başarılı bir şekilde ürün önerileri sağlanmıştır.

V. AYRIŞTIRICI (DISCRIMINATIVE) ANALİZ

Bu bölümde, Market Basket Analysis projemizde müşteri segmentasyonu yapılmıştır. *Segmentasyon*, müşteri gruplarını ayırt etmek ve belirli müşteri gruplarına yönelik stratejiler geliştirmek için kullanılan bir yöntemdir. Bu adımda segmentasyon işlemi, ülkeler bazında yapılmıştır. Özellikle farklı ülkeler arasındaki satın alma davranışlarını incelemek, pazarlama stratejileri geliştirmek ve müşteri profillemesi yapmak için önemlidir.

Farklı ülkelerdeki müşterilerin satın alma desenlerini karşılaştırmak için *chi-squared (ki-kare)* testi kullanılmıştır.

$$\chi^2 = \sum \frac{(O_i - E_i)^2}{E_i}$$

O_i, gözlenen frekansları temsil eder (gerçek veriler).

E_i, beklenen frekansları temsil eder (hipoteze göre beklenen değerler)

Bu adımda, Birleşik Krallık – Fransa ve Birleşik Krallık – Almanya arasındaki satın alma desenleri incelenip aralarında yüksek ki-kare değerleri ve p-değerinin sıfır olduğu gözlemlenmiştir.

Yüksek ki-kare değeri, iki ülke arasındaki satın alma desenlerinin büyük ölçüde farklı olduğunu göstermektedir. Düşük p-değeri ise (< 0.05), bu farkların istatistiksel olarak anlamlı olduğunu göstermektedir.

VI. SONUÇ

Bu çalışmada, market basket analizi üzerine odaklanılarak çeşitli veri madenciliği teknikleri ve algoritmaları uygulanmıştır. Proje kapsamında, bir e-ticaret sitesine ait veri seti üzerinde sık rastlanan ürün birlikteliklerini belirlemek, müşterilere yönelik öneri sistemleri geliştirmek ve müşteri segmentasyonunu sağlamak için farklı yaklaşımlar denenmiştir.

Apriori ve FP-Growth algoritmaları kullanılarak, sık ürün kümeleri ve bu kümelerden çıkarılabilecek anlamlı kurallar belirlenmiştir. Apriori algoritması ile bellek kullanımı optimize edilerek büyük veri setlerinde bile etkili analizler gerçekleştirilmiştir. FP-Growth algoritmasının ise büyük veri setleri üzerinde daha hızlı ve verimli çalıştığı gözlemlenmiştir. Ayrıca, bu iki algoritmanın çıkardığı kuralların %89 oranında benzerlik gösterdiği ve herhangi birinin kullanılmasının büyük bir kayba yol açmayacağı tespit edilmiştir.

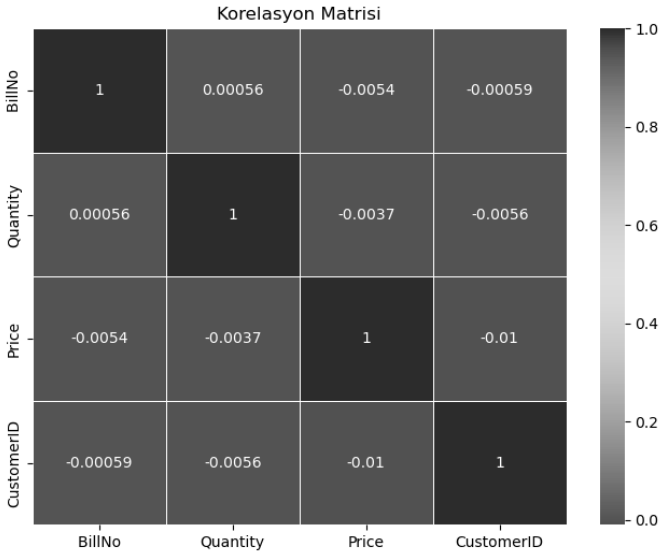
Müşteri segmentasyonu ve öneri sistemleri üzerine yapılan çalışmalarda, K-Nearest Neighbors algoritması kullanılarak belirli müşterilere kişiselleştirilmiş ürün önerileri sunulmuştur. Segmentasyon işlemi ile müşterilerin, alışveriş alışkanlıklarına göre gruplandırılması sağlanarak işletmelerin pazarlama stratejilerini daha etkili bir şekilde yönlendirmesine olanak tanınmıştır.

Ancak, proje sürecinde bazı kısıtlamalarla karşılaşılmıştır. pyECLAT kütüphanesi ile gerçekleştirilmek istenen ECLAT algoritması, büyük veri setlerinde istenilen performansı gösterememiştir ve işlem süresi beklenenden uzun sürmüştür. Bu sebeple, ECLAT algoritması, istendiği şekilde uygulanamamıştır. Ayrıca, Country değeri 'Birleşik Krallık' olan verilerin aşırı derecede baskın olmasının, analiz sonuçlarını etkileyecek bir faktör olduğu tespit edilip undersampling yöntemi uygulanmıştır. Ancak, bu işlem yalnızca predictive ve discriminative analizlerde kullanılmıştır, descriptive analizde kullanılması durumunda önemli ilişkilerin gözden kaçabileceği düşünülerek tercih edilmemiştir.

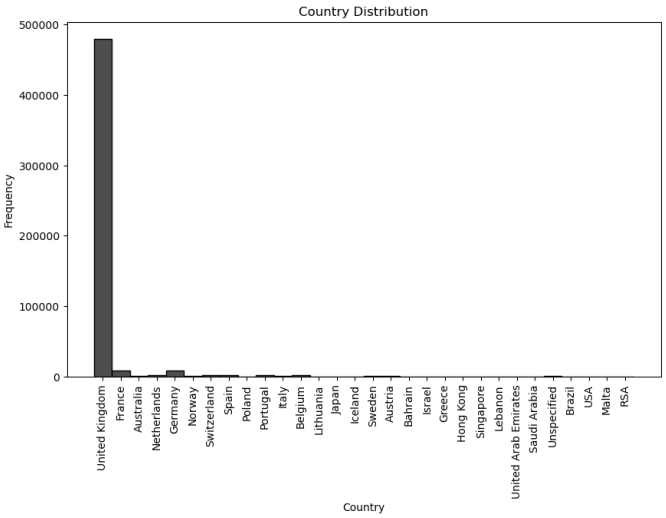
Gelecek çalışmalar için, farklı algoritmaların kullanılması ve veri setlerinin dengelenmesi adına alternatif yöntemler araştırılabilir. Ayrıca, büyük veri setleriyle çalışırken bellek ve zaman optimizasyonu üzerine daha fazla odaklanılması gerektiği görülmüştür. Bu doğrultuda, paralel işleme teknikleri ve bulut tabanlı çözümler gibi yaklaşımlar incelenebilir.

Bu çalışma, market basket analizi konusunda önemli sonuçlar elde edilmesini sağlamaktadır. Ayrıca, işletmelere, müşteri davranışlarını daha iyi anlama ve kişiselleştirilmiş pazarlama stratejileri geliştirme konusunda katkı sağlamaktadır.

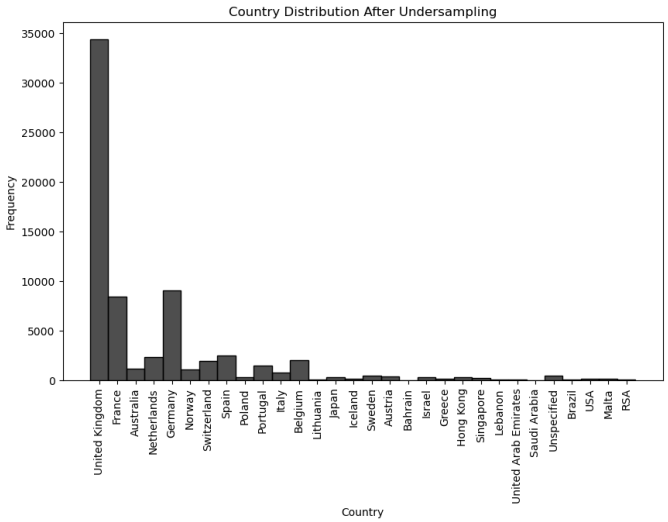
EKLER



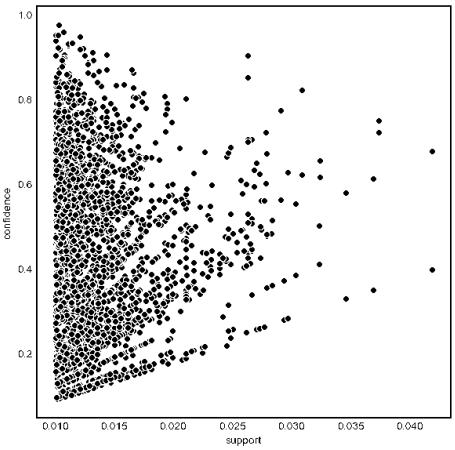
Şekil 1. Korelasyon matrisi



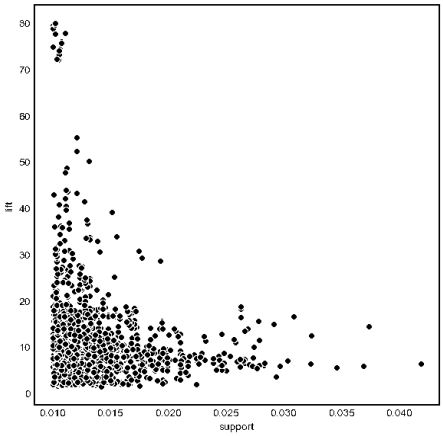
Şekil 2. Country öznitelik değerlerinin dağılımı



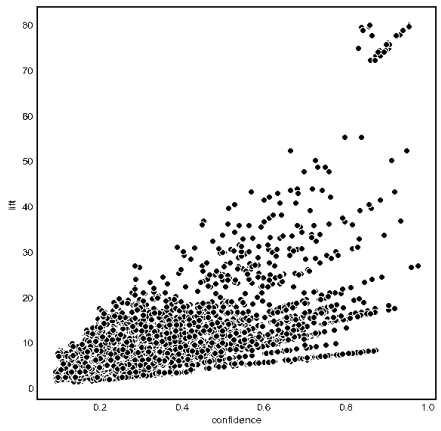
Şekil 3. Undersampling sonrası Country öznitelik değerlerinin dağılımı



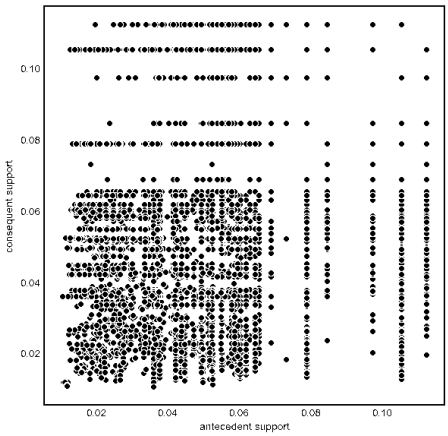
Şekil 4. Support – Confidence Dağılım Grafiği



Şekil 5. Support – Lift Dağılım Grafiği



Şekil 6. Confidence – Lift Dağılım Grafiği



Şekil 7. Antecedent Support – Consequent Support Dağılım Grafiği

```python
def get_recommendations(customer_id, n_recommendations=5):
    if customer_id in basket_customer.index:
        distances, indices = model.kneighbors(basket_customer.loc[customer_id].values.reshape(1, -1), n_neighbors=n_recommendations+1)
        similar_customers = basket_customer.iloc[indices.flatten()[1:]].index
        similar_items = basket_customer.loc[similar_customers].sum().sort_values(ascending=False)

        # Musterinin zaten satin almis oldugu urunleri cikar
        customer_items = set(basket_customer.loc[customer_id].loc[basket_customer.loc[customer_id] > 0].index)
        recommended_items = [item for item in similar_items.index if item not in customer_items]

        return recommended_items[:n_recommendations]
    else:
        return f"Customer ID {customer_id} not found in the dataset."

# customer_id'ye ait veriler kullanilarak, en yakin komsu musterileri, yani diger musteriler belirlenir.
# benzer musterilerin alisveris verileri toplanarak urunler sikliklarina gore siralanir.
# sorgu yaptigimiz musterinin zaten aldigi urunler varsa onlar cikarilir.
```

Şekil 8. Predictive Analysis – Ürün Tavsiye Sistemi Kod Parçası

REFERANSLAR

[1] Analytics Vidhya, Market Basket Analysis: A Comprehensive Guide for Businesses, https://www.analyticsvidhya.com/blog/2021/10/a-comprehensive-guide-on-market-basket-analysis/

[2] Medium, Market Basket Analysis: Techniques, Applications, and Benefits for Retailers, https://medium.com/@data-overload/market-basket-analysis-techniques-applications-and-benefits-for-retailers-d66eed1f917e

[3] Data Headhunters, How to conduct market basket analysis in Python: A Comprehensive Guide, https://dataheadhunters.com/academy/how-to-conduct-market-basket-analysis-in-python-a-comprehensive-guide/

[4] Github, Market Basket Analysis in Python, https://goldinlocks.github.io/Market-Basket-Analysis-in-Python/

[5] Github, mlxtend library, https://rasbt.github.io/mlxtend/api_subpackages/mlxtend.frequent_patterns/

[6] Medium, Association Rules — The ECLAT algorithm https://medium.com/@gabrielreversi/association-rules-the-eclat-algorithm-96d47f32f992