Veri Madenciliği: İlişki Kuralları ve Apriori Algoritması

Veri madenciliği teknikleri arasında önemli bir yere sahip olan ilişki kuralları madenciliği ve Apriori algoritması, büyük veri kümelerindeki gizli ilişkileri keşfetmemize olanak tanımaktadır.

Dr. Öğr. Üyesi Alper Talha KARADENİZ

İlişki Kuralları Madenciliği Nedir?

İlişki kuralları madenciliği, büyük veri kümeleri içerisindeki öğeler arasındaki sık tekrarlayan ilişkileri ortaya koymak amacıyla kullanılan güçlü bir veri madenciliği tekniğidir. Bu yöntem, özellikle perakende sektöründe müşteri alışveriş verileri üzerinden anlamlı ilişkiler çıkarmak için yaygın biçimde uygulanmaktadır.

Perakende sektöründe yapılan bu tür analizlere "Market Basket Analysis" (Pazar Sepeti Analizi) adı verilir. Bu analiz sayesinde "hangi ürünler genellikle birlikte satın alınıyor?" sorusuna yanıt bulunabilir.

Veri Keşfi

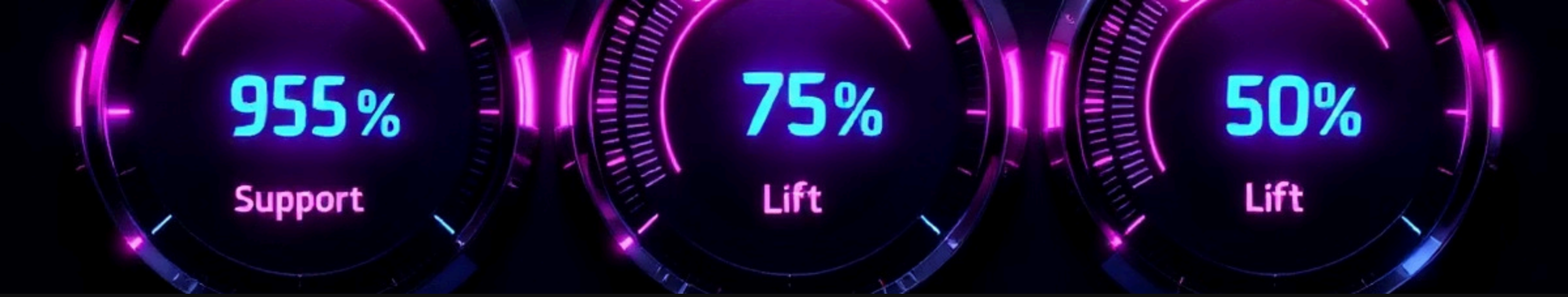
Büyük veri kümelerindeki gizli ilişkileri ortaya çıkarır

Pazar Sepeti

Müşterilerin alışveriş davranışlarını analiz eder

İş Stratejisi

Satış ve pazarlama stratejilerini optimize eder



Temel Kavramlar

İlişki kuralları madenciliğinde kullanılan üç temel ölçüt vardır. Bunlar, kuralların anlamlılığını ve kullanışlılığını değerlendirmek için kullanılır.

Destek (Support)

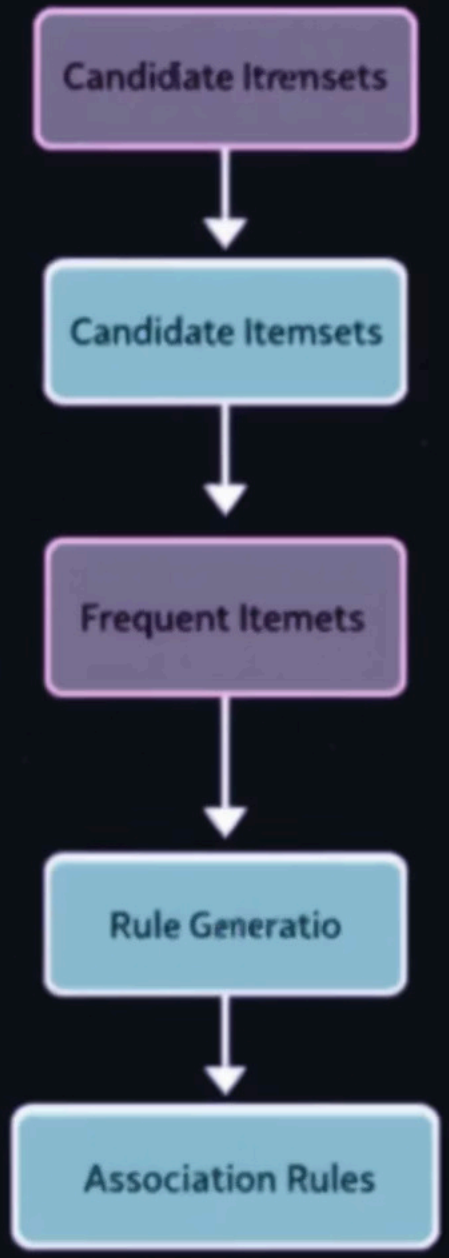
Bir öğe setinin veri kümesinde kaç kez geçtiğini gösterir. Yüksek destek değeri, ilgili öğelerin veri setinde sık görüldüğünü ifade eder.

Güven (Confidence)

X → Y kuralının doğruluk oranını gösterir. X ürünü alındığında Y ürününün de alınma olasılığını ifade eder.

Artış (Lift)

X ve Y'nin birlikte görülme olasılığının, birbirinden bağımsız olma durumuna göre ne kadar arttığını gösterir. 1'den büyük değerler pozitif ilişkiyi gösterir.



Apriori Algoritması Nedir?

Apriori algoritması, ilişki kuralları madenciliğinde en yaygın kullanılan algoritmalardan biridir. Bu algoritma, minimum destek değerinin altındaki öğe kümelerini eler ve kalanlarla ilişki kuralları üretir.

Algoritmanın temel özelliği, bir öğe kümesi sık değilse, bu kümeyi içeren daha büyük kümelerin de sık olamayacağı varsayımıdır. Bu prensibe "Apriori ilkesi" adı verilir ve algoritmanın hesaplama verimliliğini önemli ölçüde artırır.

Minimum Destek Belirleme

Analizde kullanılacak minimum destek eşiği belirlenir

Sık Öğe Kümeleri Bulma

Minimum destek değerini karşılayan öğe kümeleri tespit edilir

İlişki Kuralları Oluşturma

Sık öğe kümeleri kullanılarak güven değeri yüksek kurallar çıkarılır

Örnek Veri Seti

Apriori algoritmasını anlamak için basit bir alışveriş veri seti kullanacağız. Bu veri seti, beş farklı alışveriş işleminde satın alınan ürünleri göstermektedir.

Veri setinde süt, ekmek ve tereyağı ürünlerinin farklı kombinasyonlarda satın alındığı görülmektedir. Bu basit örnek üzerinden ilişki kuralları çıkararak algoritmanın çalışma prensibini inceleyeceğiz.

Transaction ID	Items
1	Milk, Bread, Butter
2	Bread, Butter
3	Milk, Bread
4	Milk, Butter
5	Bread, Butter

Python ile Uygulama

Apriori algoritmasını Python programlama dili kullanarak uygulayacağız. Bu uygulama Google Colab ortamında çalıştırılabilir ve mlxtend kütüphanesini kullanır.

Öncelikle gerekli kütüphaneleri yükleyip, örnek veri setimizi hazırlayacağız. Daha sonra apriori algoritmasını kullanarak sık öğe kümelerini bulacak ve ilişki kurallarını oluşturacağız.

Kütüphanelerin Yüklenmesi

mlxtend, pandas gibi gerekli kütüphanelerin kurulumu

Veri Setinin Hazırlanması

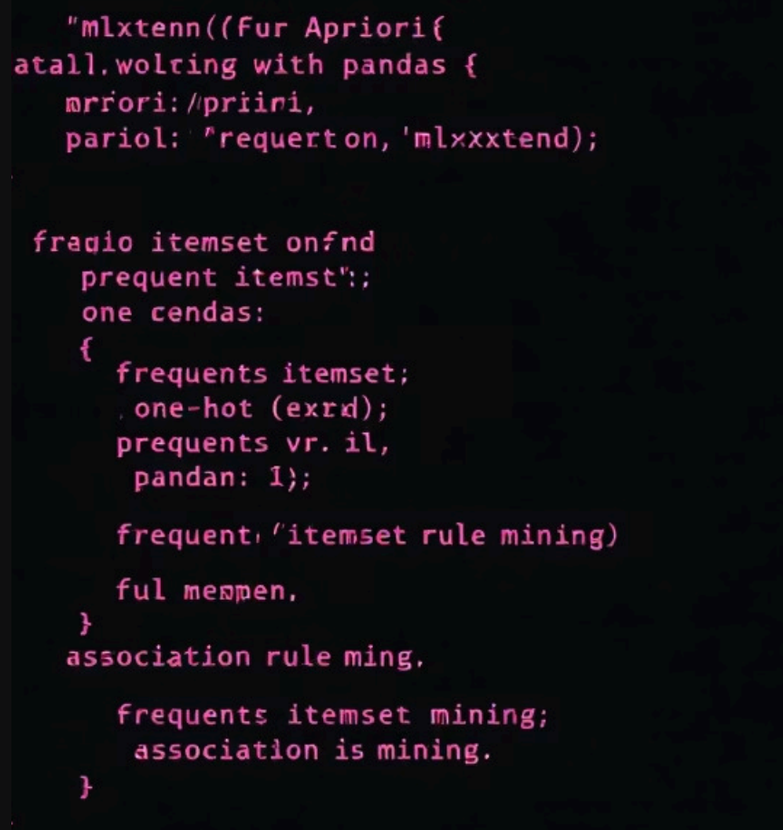
TransactionEncoder ile verilerin uygun formata dönüştürülmesi

Sık Öğe Kümelerinin Bulunması

Apriori algoritması ile minimum destek değerini karşılayan kümelerin tespiti

İlişki Kurallarının Oluşturulması

Association rules fonksiyonu ile kuralların çıkarılması



Gerekli Kütüphaneler

!pip install mlxtend --quiet

import pandas as pd from mlxtend.preprocessing import TransactionEncoder from mlxtend.frequent_patterns import apriori, association_rules

Veri Setinin Hazırlanması

dataset = [['Milk', 'Bread', 'Butter'], ['Bread', 'Butter'], ['Milk', 'Bread'], ['Milk', 'Butter'], ['Bread', 'Butter']]

te = TransactionEncoder() te_array = te.fit(dataset).transform(dataset) df = pd.DataFrame(te_array, columns=te.columns_)

df.head()

Sık Öğe Kümelerinin Bulunması

frequent_itemsets = apriori(df, min_support=0.6, use_colnames=True) print(frequent_itemsets)

İlişki Kurallarının Oluşturulması

rules = association_rules(frequent_itemsets, metric="confidence", min_threshold=0.7) rules[['antecedents', 'consequents', 'support', 'confidence', 'lift']]

Görselleştirme

import matplotlib.pyplot as plt import seaborn as sns

plt.figure(figsize=(8,6)) sns.scatterplot(data=rules, x='support', y='confidence', size='lift', hue='lift', palette='viridis', sizes=(40, 200)) plt.title('Association Rule Scatter Plot') plt.xlabel('Support') plt.ylabel('Confidence') plt.grid(True) plt.show()

Sonuçların Yorumlanması ve Görselleştirme

Algoritma sonucunda elde edilen ilişki kurallarını yorumlayarak, veri setindeki anlamlı ilişkileri ortaya çıkarabiliriz. Örneğin 'Bread' → 'Butter' kuralı için elde edilen metrikler şu şekilde yorumlanabilir:

0.6

Destek (Support)

Bu ikili 5 işlemin 3'ünde birlikte alınmıştır

0.75

Güven (Confidence)

Ekmek alanların %75'i aynı zamanda tereyağı da almıştır

1.25

Artış (Lift)

Bu kural, tereyağının alınma olasılığını %25 artırmaktadır

Sonuçları görselleştirmek için scatter plot kullanılabilir. Bu görselleştirme, kuralların destek, güven ve artış değerlerini bir arada görmemizi sağlar.

Akademik Değerlendirme ve Sonuç

Apriori algoritması, düşük hesaplama maliyeti ve kolay yorumlanabilirliği nedeniyle özellikle pazar sepeti analizinde tercih edilmektedir. Ancak, yüksek boyutlu veri kümelerinde performans problemi yaratabilir.

Alternatif olarak FP-Growth algoritması, sık öğe kümelerini daha hızlı bulma avantajına sahiptir. Bu sunum, küçük ölçekli bir örnekle temel kavramların öğrenilmesini hedeflemektedir.

Avantajlar

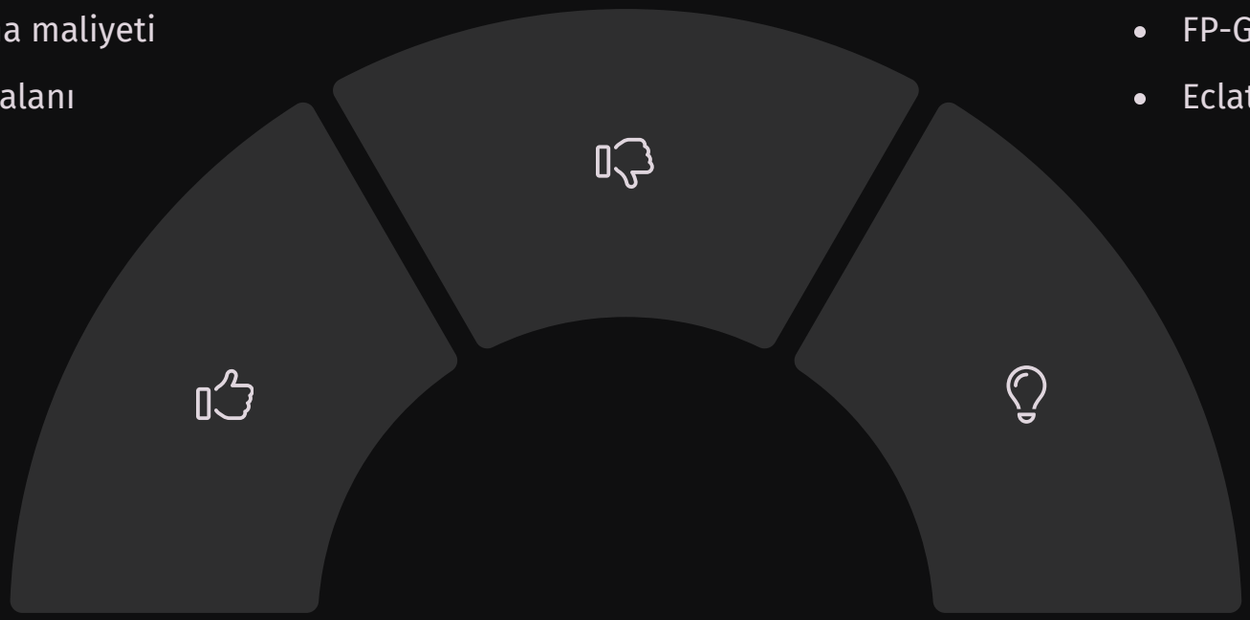
- Kolay yorumlanabilirlik
- Düşük hesaplama maliyeti
- Yaygın kullanım alanı

Dezavantajlar

- Yüksek boyutlu verilerde performans sorunu
- Çok sayıda aday küme oluşturma

Alternatifler

- FP-Growth algoritması
- Eclat algoritması



Veri Madenciliğinde Desen Çıkarma

Veri madenciliği, büyük veri kümelerinden anlamlı desenlerin keşfedilmesini sağlar. Desen çıkarma, verilerde sık tekrarlanan desenlerin bulunmasıdır. Bu, pazarlama, sağlık ve finans gibi alanlarda karar destek sistemlerine yön verir.

Desen çıkarma yöntemleri; sıklıkla kullanılan ilişki kuralları, sıralı desenler ve kümeleme analizini içerir. Her biri veride farklı içgörüler sunar.

İlişki Kuralları

Öğeler arasındaki eşzamanlı ilişkileri keşfeder.

Sıralı Desenler

Zaman veya sıra bağımlı örüntüleri tanımlar.

Kümeleme Analizi

Veri noktalarını benzerliklerine göre gruplar.

Desen Türleri ve Nerelerde Kullanılır?

- **Sık Öğeler (Frequent Itemsets):** Verilerde sıkça birlikte görülen öğe grupları.
- **Ardışıl Desenler (Sequential Patterns):** Zaman veya sıra bağımlı tekrar eden desenler.
- **Zaman Serisi Desenleri:** Zaman içinde değişen verilerdeki eğilimler ve döngüler.
- Kullanım Alanları: Market sepeti analizi, tıklama verisi, sağlık ve finansal dolandırıcılık.
- İpucu: "Sepetinde süt varsa, ekmek alma olasılığı yüksektir."