Öneri Sistemi Dökümantasyonu

1. Veri Setinin Genel Mimarisi ve Kullanımı

Veri Seti Yapısı

Veri seti, kullanıcıların ve ürünlerin özelliklerini kapsayan bir çok boyutlu veri yapısından oluşmaktadır. Bu yapıda, iki ana bölüm bulunmaktadır:

- **Kullanıcı Özellikleri:** Yaş, cinsiyet, konum, beden, daha önceki alışveriş sayısı ve alışveriş sıklığı gibi veriler.
- **Ürün Özellikleri:** Satın alınan ürün adı, kategori, renk, sezon ve fiyat bilgisi gibi veriler.

Bu veri yapısı, `Customer ID` kolonu aracılığıyla kullanıcı ve ürün bilgilerini birbirine bağlamaktadır.

Kullanım Amacı

Veri seti, kullanıcı bazlı (önceki benzer kullanıcı davranışları) ve ürün bazlı (ürün özellik benzerliğine dayalı) öneri sistemlerinin geliştirilmesi için kullanılmaktadır.

---

2. Öneri Modeli İçin İzlenen Yol

Genel Yaklaşım

Sistem iki temel öneri modu üzerine inşa edilmiştir:

1. **Kullanıcı Bazlı Model:** Kullanıcılar arası benzerlikleri kullanarak hedef kullanıcıya öneriler sunar.
2. **Ürün Bazlı Model:** Ürünler arası benzerlikleri kullanarak hedef ürünle benzer özelliklere sahip ürünleri listeler.

Veri İşleme Aşamaları

Veri Ayrıştırma

`data_preprocessing.py` dosyasındaki `split_dataset` fonksiyonu, veri setini kullanıcı (örneğin, yaş, cinsiyet) ve ürün (örneğin, kategori, fiyat) özelliklerine göre ikiye ayırır:

```
user_df, item_df = split_dataset(data_path)
```

iki parçaya ayrılan dataset'te iki tarafı birbirine bağlı tutan sütun `Customer ID` özelliği oluyor.

---

3. Yöntem Aşamaları ve Uygulama

Kullanıcı Bazlı Öneri

```
        recommender = UserBasedRecommender(user_df, item_df)
        recommendations, target_info = recommender.get_recommendations(
            args.user_id,
            args.num_recommendations
        )
```

`main`'den bu şekilde çağrılan constructor aşağıdaki fonksiyonları çalıştırıp özellikleri ayarlar:

```
    def __init__(self, user_df, item_df):
        self.user_df = user_df.copy()
        self.item_df = item_df.copy()
        self.encoded_users = encode_features(user_df)
        self.similarity_matrix = cosine_similarity(self.encoded_users)
```

Veri Kodlama

Veri, kategorik değerlerin sayısal formatta kullanılabilmesi için **One-Hot Encoding** ve özel kodlama yöntemleriyle işlenmiştir:

- `Size` ve `Frequency of Purchases` gibi alanlar için özel kodlama.
- Sayısal veriler için **MinMaxScaler** kullanılarak normalleştirme.

Her özelliğin kendi ihtiyacına göre belli bir yöntem uygulanmıştır. Bu işlem, **`data_preprocessing.py`** dosyasındaki `encode_features` fonksiyonunda gerçekleştirilmektedir.

MinMaxScaler:

- Age
- Previous Purchases

One-Hot Encoding:

- Gender
- Location
- Color
- Category
- Season

Mapping (manuel olarak yapılanlar):

- Size:
  - `{'S': 0, 'M': 1, 'L': 2, 'XL': 3}`

- Frequency of purchases
    - `{'Rarely': 0, 'Occasionally': 1, 'Monthly': 2, 'Weekly': 3, 'Often': 4}`

Kullanıcılar arası benzerlik, **kosinüs benzerlik metriği** ile hesaplanır. Şu adımlar izlenir:

1. Kullanıcı özellikleri `encode_features` ile kodlanır.
2. Kodlanan veriler arasında `cosine_similarity` kullanılarak benzerlik matrisi oluşturulur.
3. Hedef kullanıcıya en benzer diğer kullanıcılar bulunarak, bu kullanıcıların önceki alışverişleri hedef kullanıcıya önerilir.

Koddan örnek:

```
similar_users_idx = np.argsort(user_similarities)[::-1][1:n_recommendations+1]
similar_users = self.user_df.iloc[similar_users_idx]
```

Ürün Bazlı Öneri

Ürün benzerlikleri de benzer yöntemlerle hesaplanır. Hedef kullanıcının önceki alışverişlerine dayalı olarak benzer ürünler önerilir.

Koddan örnek:

```
item_similarities = self.similarity_matrix[user_item_idx]
similar_items_idx = np.argsort(item_similarities)[::-1][1:n_recommendations+1]
```

---

3. Kümeleme Bazlı Öneri (KMeans)

Sistem üçüncü bir yaklaşım olarak KMeans kümeleme algoritmasını kullanmaktadır. Bu yaklaşım hem kullanıcıları hem de ürünleri kümelerine ayırarak çok boyutlu bir öneri sistemi oluşturur.

Kümeleme Mimarisi

1. Veri Hazırlama ve Kümeleme

Kümeleme işlemi iki ayrı boyutta gerçekleştirilir:

Kullanıcı Kümeleri İçin Kullanılan Özellikler:

```
user_features = [
    'Age',                    # Sayısal
    'Gender',                 # Kategorik
    'Previous Purchases',     # Sayısal
    'Frequency of Purchases', # Özel kodlama
    'Size',                   # Özel kodlama
    'Subscription Status'     # Kategorik
]
```

Ürün Kümeleri İçin Kullanılan Özellikler:

```
item_features = [
    'Item Purchased',  # Kategorik
    'Category',        # Kategorik
    'Season'          # Kategorik
]
```

2. Optimal Küme Sayısı Belirleme

Sistem, "Elbow Method" kullanarak optimal küme sayısını otomatik olarak belirler:

```
def find_optimal_k(self, data, k_range=range(1, 11)):
    inertias = []
    for k in k_range:
        kmeans = KMeans(n_clusters=k, random_state=42)
        kmeans.fit(data)
        inertias.append(kmeans.inertia_)

    kneedle = KneeLocator(
        x=list(k_range),
        y=inertias,
        S=1.0,
        curve='convex',
        direction='decreasing'
    )
    return kneedle.knee if kneedle.knee else 3
```

Benzerlik Skorlama Sistemi

Öneriler oluşturulurken, çeşitli faktörler göz önünde bulundurularak bir benzerlik skoru hesaplanır:

1. **Ürün Kümesi Benzerliği (0.31):**
    - Aynı ürün kümesindeki ürünlere en yüksek ağırlık verilir

    ```
    recommendations.loc[recommendations['Cluster'] == item_cluster, 'Similarity']
    += 0.35
    ```

2. **Kategori Benzerliği (0.25):**
    - Aynı kategorideki ürünler ikinci öncelikli

    ```
    recommendations.loc[recommendations['Category'] == user_item['Category'],
    'Similarity'] += 0.25
    ```

3. **Sezon Benzerliği (0.15):**
    - Mevsimsel uyum

    ```
    recommendations.loc[recommendations['Season'] == user_item['Season'],
    'Similarity'] += 0.15
    ```

4. **Kullanıcı Kümesi Benzerliği (0.19):**

- Benzer kullanıcı profillerinin tercihleri

```
recommendations.loc[recommendations['User_Cluster'] == user_cluster,
'Similarity'] += 0.15
```

5. **Fiyat Aralığı Benzerliği (0.05):**
   - ±%20 fiyat aralığındaki ürünler

```
price_range = 0.2 * user_item['Purchase Amount (USD)']
recommendations.loc[price_mask, 'Similarity'] += 0.05
```

6. **Renk Benzerliği (0.05):**
   - Aynı renkteki ürünler

```
recommendations.loc[recommendations['Color'] == user_item['Color'],
'Similarity'] += 0.05
```

Filtreleme ve Öneri Mekanizması

1. **Minimum Benzerlik Eşiği:**
   - Sistem yalnızca 0.95 ve üzeri benzerlik skoruna sahip ürünleri önerir

```
MIN_SIMILARITY_THRESHOLD = 0.95
recommendations = recommendations[recommendations['Similarity'] >=
MIN_SIMILARITY_THRESHOLD]
```

2. **Sıralama:**
   - Öneriler benzerlik skoruna göre azalan sırada listelenir

```
recommendations = recommendations.sort_values('Similarity', ascending=False)
```

Küme İçgörüleri

Sistem her küme için detaylı analizler sunar:

Kullanıcı Kümeleri İçin:

- Küme büyüklüğü
- Ortalama yaş
- Baskın cinsiyet
- Ortalama alışveriş sayısı
- Abonelik oranı
- En yaygın beden

Ürün Kümeleri İçin:

- Küme büyüklüğü
- Baskın kategori
- Baskın sezon
- Ortalama fiyat

- En yaygın renk

Avantajları

1. Çok boyutlu analiz imkanı
2. Hem kullanıcı hem ürün benzerliklerini dikkate alma
3. Otomatik küme sayısı optimizasyonu
4. Detaylı benzerlik skorlama sistemi
5. Esnek filtreleme mekanizması

Modelin Çağrılması ve Kullanımı

```
if args.mode == 'cluster':
    recommender = ClusteringRecommender(user_df, item_df)
    insights = recommender.get_cluster_insights()
    recommendations, target_item = recommender.get_cluster_recommendations(
        args.user_id,
        args.num_recommendations
    )
```

Bazı Çıktılar:

```
~$ make help
Kullanılabilir komutlar:
  make setup                              - Gerekli bağımlılıkları yükler
  make user USER_ID=123                   - Kullanıcı bazlı öneriler oluşturur
  make item USER_ID=123                   - Ürün bazlı öneriler oluşturur
  make cluster USER_ID=123                - Küme bazlı öneriler oluşturur
  make clean                              - Geçici dosyaları temizler
  make evaluate N_TEST_USERS=100          - Öneri sistemlerini değerlendirir

Örnekler:
  make user USER_ID=123
  make user USER_ID=123 NUM_RECOMMENDATIONS=10
  make item USER_ID=123
  make cluster USER_ID=123
```

```
~$ make user USER_ID=123
-> Kullanıcı bazlı öneriler oluşturuluyor...
python3 src/main.py \
    --mode user \
    --user_id 123 \
    --num_recommendations 3
/usr/lib/python3/dist-packages/scipy/__init__.py:146: UserWarning: A NumPy
version >=1.17.3 and <1.25.0 is required for this version of SciPy (detected
version 1.25.0
  warnings.warn(f"A NumPy version >={np_minversion} and <{np_maxversion}"
```

```
/home/ahmet/.local/lib/python3.10/site-
packages/matplotlib/projections/__init__.py:63: UserWarning: Unable to import
Axes3D. This may be due to multiple versions of Matplotlib being installed (e.g.
as a system package and as a pip package). As a result, the 3D projection is not
available.
  warnings.warn("Unable to import Axes3D. This may be due to multiple versions of
"

Kullanıcı bazlı öneriler hazırlanıyor...

==================================================
HEDEF BİLGİLERİ:

Kullanıcı Bilgileri:
  - Yaş: 40
  - Cinsiyet: Male
  - Konum: Washington
  - Beden: L
  - Önceki Alışverişler: 44
  - Alışveriş Sıklığı: Fortnightly
==================================================

==================================================
KULLANICI BAZLI ÖNERİLER:

==================== Öneri 1 ====================
Ürün Bilgileri:
  - İsim: T-shirt
  - Kategori: Clothing
  - Renk: Purple
  - Sezon: Fall
  - Fiyat: $31.00
  - Benzerlik Skoru: 0.9979

Kullanıcı Bilgileri:
  - Yaş: 36
  - Cinsiyet: Male
  - Konum: Washington
  - Beden: L
  - Önceki Alışverişler: 35
  - Alışveriş Sıklığı: Every 3 Months

==================== Öneri 2 ====================
Ürün Bilgileri:
  - İsim: Dress
  - Kategori: Clothing
  - Renk: Purple
  - Sezon: Summer
  - Fiyat: $67.00
  - Benzerlik Skoru: 0.9932

Kullanıcı Bilgileri:
  - Yaş: 43
  - Cinsiyet: Male
  - Konum: Washington
  - Beden: L
```

```
  - Önceki Alışverişler: 27
  - Alışveriş Sıklığı: Every 3 Months

==================== Öneri 3 ====================
Ürün Bilgileri:
  - İsim: Jewelry
  - Kategori: Accessories
  - Renk: Beige
  - Sezon: Winter
  - Fiyat: $33.00
  - Benzerlik Skoru: 0.9767

Kullanıcı Bilgileri:
  - Yaş: 48
  - Cinsiyet: Male
  - Konum: Washington
  - Beden: XL
  - Önceki Alışverişler: 36
  - Alışveriş Sıklığı: Annually
==================================================
```

```
~$ make item USER_ID=123
-> Ürün bazlı öneriler oluşturuluyor...
python3 src/main.py \
    --mode item \
    --user_id 123 \
    --num_recommendations 3
/usr/lib/python3/dist-packages/scipy/__init__.py:146: UserWarning: A NumPy
version >=1.17.3 and <1.25.0 is required for this version of SciPy (detected
version 1.25.0
  warnings.warn(f"A NumPy version >={np_minversion} and <{np_maxversion}"
/home/ahmet/.local/lib/python3.10/site-
packages/matplotlib/projections/__init__.py:63: UserWarning: Unable to import
Axes3D. This may be due to multiple versions of Matplotlib being installed (e.g.
as a system package and as a pip package). As a result, the 3D projection is not
available.
  warnings.warn("Unable to import Axes3D. This may be due to multiple versions of
"

Ürün bazlı öneriler hazırlanıyor...

==================================================
HEDEF BİLGİLERİ:

==================================================
Ürün Bilgileri:
  - İsim: Backpack
  - Kategori: Accessories
  - Renk: Brown
  - Sezon: Spring
  - Fiyat: $0.00
==================================================

==================================================
ÜRÜN BAZLI ÖNERİLER:
```

```
==================== Öneri 1 ====================
Ürün Bilgileri:
  - İsim: Gloves
  - Kategori: Accessories
  - Renk: Brown
  - Sezon: Spring
  - Fiyat: $45.00
  - Benzerlik Skoru: 0.7500

==================== Öneri 2 ====================
Ürün Bilgileri:
  - İsim: Backpack
  - Kategori: Accessories
  - Renk: Orange
  - Sezon: Spring
  - Fiyat: $37.00
  - Benzerlik Skoru: 0.7500

==================== Öneri 3 ====================
Ürün Bilgileri:
  - İsim: Backpack
  - Kategori: Accessories
  - Renk: Orange
  - Sezon: Spring
  - Fiyat: $63.00
  - Benzerlik Skoru: 0.7500
==================================================
```

```
~$ make cluster USER_ID=123
-> Küme bazlı öneriler oluşturuluyor...
python3 src/main.py \
    --mode cluster \
    --user_id 123 \
    --num_recommendations 3
/usr/lib/python3/dist-packages/scipy/__init__.py:146: UserWarning: A NumPy
version >=1.17.3 and <1.25.0 is required for this version of SciPy (detected
version 1.25.0
  warnings.warn(f"A NumPy version >={np_minversion} and <{np_maxversion}"
/home/ahmet/.local/lib/python3.10/site-
packages/matplotlib/projections/__init__.py:63: UserWarning: Unable to import
Axes3D. This may be due to multiple versions of Matplotlib being installed (e.g.
as a system package and as a pip package). As a result, the 3D projection is not
available.
  warnings.warn("Unable to import Axes3D. This may be due to multiple versions of
"

Küme bazlı öneriler hazırlanıyor...

Optimal küme sayıları belirlendi:
- Kullanıcı kümeleri: 2
- Ürün kümeleri: 4

KÜME ANALİZİ SONUÇLARI:
==================================================
```

```
Kullanıcı Kümeleri:

Küme 0:
  - Küme Büyüklüğü: 2808 kullanıcı
  - Ortalama Yaş: 43.9
  - Baskın Cinsiyet: Male
  - Ortalama Alışveriş: 25.3

Küme 1:
  - Küme Büyüklüğü: 1092 kullanıcı
  - Ortalama Yaş: 44.5
  - Baskın Cinsiyet: Male
  - Ortalama Alışveriş: 25.5

Ürün Kümeleri:

Küme 0:
  - Küme Büyüklüğü: 842 ürün
  - Baskın Kategori: Footwear
  - Baskın Sezon: Summer
  - Ortalama Fiyat: $59.61

Küme 1:
  - Küme Büyüklüğü: 535 ürün
  - Baskın Kategori: Clothing
  - Baskın Sezon: Spring
  - Ortalama Fiyat: $60.03

Küme 2:
  - Küme Büyüklüğü: 1283 ürün
  - Baskın Kategori: Clothing
  - Baskın Sezon: Winter
  - Ortalama Fiyat: $59.68

Küme 3:
  - Küme Büyüklüğü: 1240 ürün
  - Baskın Kategori: Accessories
  - Baskın Sezon: Fall
  - Ortalama Fiyat: $59.84
==================================================

Kullanıcı ID: 123 için öneriler hazırlanıyor...
Kullanıcı kümesi: 0
Kullanıcının ürün kümesi: 3
Kullanıcının mevcut ürünü: Backpack
Toplam değerlendirilecek ürün sayısı: 3899
Aynı ürün kümesinde olan ürün sayısı: 1239
Aynı kategoride olan ürün sayısı: 1239
Aynı sezonda olan ürün sayısı: 998
Aynı kullanıcı kümesindeki kullanıcılardan ürün sayısı: 2807
Benzer fiyat aralığında olan ürün sayısı: 827
Aynı renkte olan ürün sayısı: 140

Benzerlik skorları dağılımı:
count    3899.000000
```

```
mean        0.349449
std         0.294627
min         0.000000
25%         0.150000
50%         0.200000
75%         0.750000
max         1.000000
Name: Similarity, dtype: float64

0.3 üzeri benzerlik skoruna sahip ürün sayısı: 1732

Toplam önerilen ürün sayısı: 3

==================================================
HEDEF BİLGİLERİ:

==================================================
Ürün Bilgileri:
  - İsim: Backpack
  - Kategori: Accessories
  - Renk: Brown
  - Sezon: Spring
  - Fiyat: $40.00
==================================================

==================================================
KÜME BAZLI ÖNERİLER:

==================== Öneri 1 ====================
Ürün Bilgileri:
  - İsim: Jewelry
  - Kategori: Accessories
  - Renk: Brown
  - Sezon: Spring
  - Fiyat: $32.00
  - Benzerlik Skoru: 1.0000
  - Kullanıcı Kümesi: 0
  - Ürün Kümesi: 3

Kullanıcı Bilgileri:
  - Yaş: 35
  - Cinsiyet: Female
  - Konum: Kentucky
  - Beden: S
  - Önceki Alışverişler: 41
  - Alışveriş Sıklığı: Annually

==================== Öneri 2 ====================
Ürün Bilgileri:
  - İsim: Jewelry
  - Kategori: Accessories
  - Renk: Brown
  - Sezon: Spring
  - Fiyat: $39.00
  - Benzerlik Skoru: 1.0000
  - Kullanıcı Kümesi: 0
```

```
  - Ürün Kümesi: 3

Kullanıcı Bilgileri:
  - Yaş: 64
  - Cinsiyet: Male
  - Konum: Louisiana
  - Beden: L
  - Önceki Alışverişler: 40
  - Alışveriş Sıklığı: Quarterly

==================== Öneri 3 ====================
Ürün Bilgileri:
  - İsim: Hat
  - Kategori: Accessories
  - Renk: Magenta
  - Sezon: Spring
  - Fiyat: $38.00
  - Benzerlik Skoru: 0.9500
  - Kullanıcı Kümesi: 0
  - Ürün Kümesi: 3

Kullanıcı Bilgileri:
  - Yaş: 57
  - Cinsiyet: Male
  - Konum: Nebraska
  - Beden: L
  - Önceki Alışverişler: 33
  - Alışveriş Sıklığı: Annually
================================================+
```

---

4. Model Performans Değerlendirmesi

Değerlendirme Yaklaşımı

Sistemdeki üç farklı öneri modelinin (Kullanıcı Bazlı, Ürün Bazlı ve Kümeleme Bazlı) performansını değerlendirmek için kapsamlı bir değerlendirme sistemi geliştirilmiştir. Bu sistem, önerilerin kalitesini ve tutarlılığını ölçmek için çeşitli metrikler kullanmaktadır.

Değerlendirme Mimarisi

1. Benzerlik Matrisi Oluşturma

```python
self.similarity_features = [
    'Item Purchased',
    'Category',
    'Color',
    'Season',
    'Purchase Amount (USD)'
]
```

Bu özellikler kullanılarak ürünler arasında bir benzerlik matrisi oluşturulur. Bu matris, önerilerin kalitesini değerlendirmek için temel bir metrik olarak kullanılır.

2. Başarı Skoru Hesaplama

Her öneri için başarı skoru şu adımlarla hesaplanır:

1. **Özellik Eşleştirme:**

```
potential_matches = self.item_df[
    ((self.item_df['Item Purchased'] == rec['Item Purchased']) |
     (self.item_df['Category'] == rec['Category'])) &
    (self.item_df['Customer ID'] != user_id)
]
```

2. **Bonus Skor Sistemi:**

```
bonus = 0.0
if potential_matches.loc[idx, 'Season'] == user_item['Season']:
    bonus += 0.1
if potential_matches.loc[idx, 'Color'] == user_item['Color']:
    bonus += 0.05

# Fiyat benzerliği için bonus
price_diff = abs(potential_matches.loc[idx, 'Purchase Amount (USD)'] -
                 user_item['Purchase Amount (USD)'])
if price_diff <= 20:
    bonus += 0.05
```

3. **Final Skor Hesaplama:**

- Temel benzerlik skoru ve bonus skorlar birleştirilerek final skor oluşturulur
- Her öneri için bu skorların ortalaması alınır

Test Süreci

1. **Test Kullanıcıları Seçimi:**

```
test_users = np.random.choice(user_df['Customer ID'].unique(),
                              size=min(n_test_users, len(user_df)),
                              replace=False)
```

2. **Öneri Sayısı Aralıkları:**

- 10'dan 100'e kadar 10'ar artışla farklı öneri sayıları test edilir
- Her öneri sayısı için tüm modeller değerlendirilir

3. **Model Başına Test:**

```
for n_recommendations in recommendation_ranges:
    # Her model için test
    for user_id in test_users:
        # User-based öneriler
        recommendations, _ = user_recommender.get_recommendations(
```

```
            user_id, n_recommendations * 2)

        # Item-based öneriler
        recommendations, _ = item_recommender.get_recommendations(
            user_id, n_recommendations * 2)

        # Cluster-based öneriler
        recommendations, _ = cluster_recommender.get_cluster_recommendations(
            user_id, n_recommendations * 2)
```

Normalizasyon ve Karşılaştırma

1. **Skor Normalizasyonu:**

```
scaler = MinMaxScaler()
all_scores = []
for scores in results.values():
    all_scores.extend(scores)

all_scores = np.array(all_scores).reshape(-1, 1)
normalized_scores = scaler.fit_transform(all_scores).flatten()
```

2. **Adil Karşılaştırma Mekanizması:**

- Her model için tam olarak aynı sayıda öneri değerlendirilir
- Yetersiz öneri durumunda skor 0 olarak atanır
- Tüm skorlar aynı ölçekte normalize edilir

Değerlendirme Sonuçları

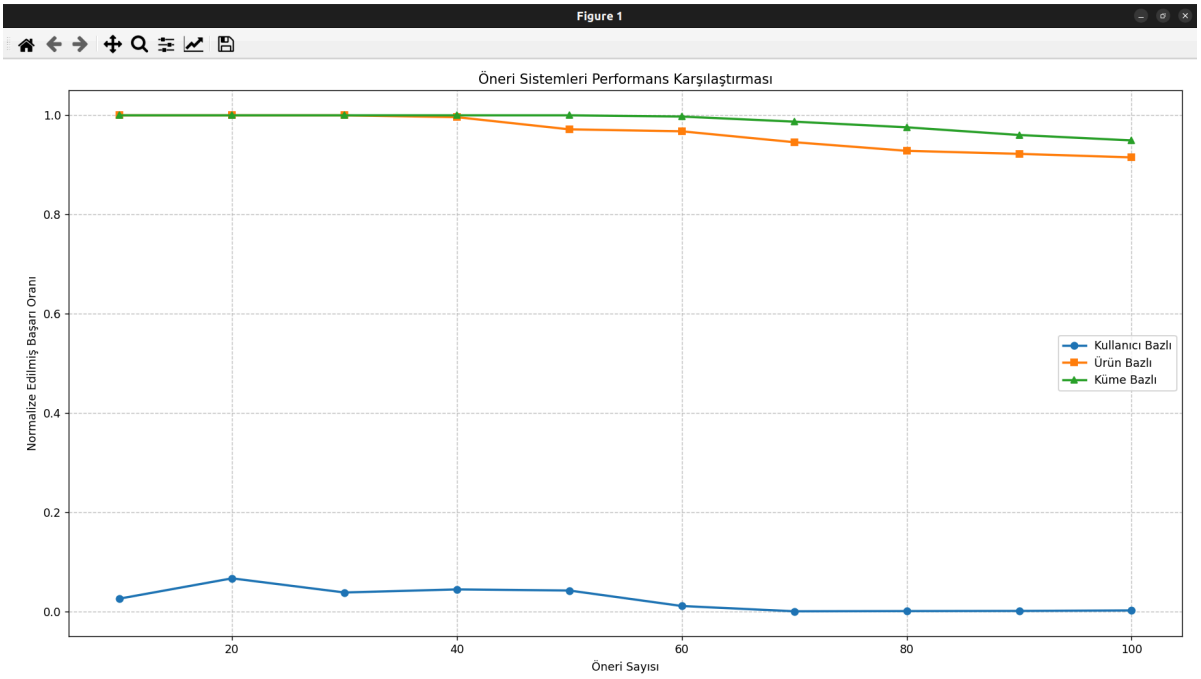
Performans değerlendirmesi sonucunda elde edilen grafikte:

1. **Küme Bazlı Model (Yeşil Çizgi):**
    - En yüksek ve en tutarlı performansı göstermekte
    - Farklı öneri sayılarında ~1.0 civarında stabil başarı oranı
    - Çok boyutlu analiz yaklaşımının etkisi görülmekte
2. **Ürün Bazlı Model (Turuncu Çizgi):**
    - İkinci en iyi performansı sergiliyor
    - Öneri sayısı arttıkça hafif performans düşüşü
    - Ürün özellikleri arasındaki benzerliğin etkili olduğunu gösteriyor
3. **Kullanıcı Bazlı Model (Mavi Çizgi):**
    - En düşük performansı gösteriyor
    - Kullanıcı özelliklerinin tek başına yeterli olmadığını işaret ediyor

Sonuç ve Çıkarımlar

1. **Model Performansı:**
    - Kümeleme yaklaşımı en başarılı sonuçları veriyor
    - Çok boyutlu analiz, tek boyutlu analizden daha etkili
    - Kullanıcı bazlı yaklaşım bu veri seti için yetersiz kalıyor
2. **Öneri Sayısı Etkisi:**
    - Küme bazlı model öneri sayısından minimum etkileniyor
    - Ürün bazlı model artan öneri sayısıyla kademeli düşüş gösteriyor
    - Kullanıcı bazlı model düşük performansını koruyor
3. **Geliştirme Önerileri:**
    - Kullanıcı bazlı modelin özellik seçimi geliştirilebilir
    - Hibrit yaklaşımlar denenebilir
    - Farklı kümeleme algoritmaları test edilebilir

Performans Grafiği:



5. Genel Mimari

Dosya Yapısı

1. `main.py`: Sistem için ana çalışma dosyası. Kullanıcı ve ürün bazlı modları çalıştırmak için gerekli argümanları alır.
2. `data_preprocessing.py`: Veri temizleme, ayırma ve kodlama işlemleri.
3. `recommender_models.py`: Kullanıcı ve ürün bazlı öneri algoritmalarının tanımlandığı dosya.