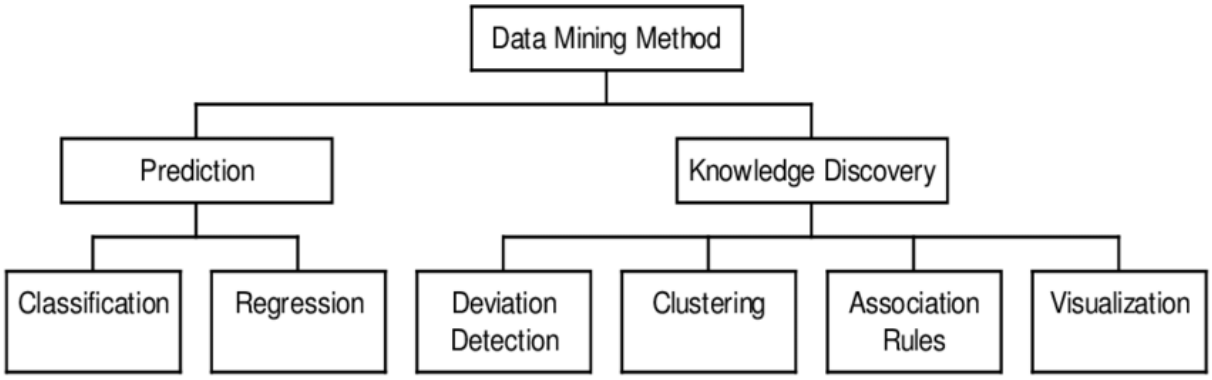
Veri Madenciliğine Giriş Dersi Dönem Projesi Raporu

Öğrenci Adı: Esranur Sevilmiş

Öğrenci No: 20360859026

VERİ MADENCİLİĞİ(DATA MINING) NEDİR?

Veri madenciliği, geniş bir veri yelpazesinden bilgi üretmeyi ve sonuçlarını kullanıcılara kapsamlı bir şekilde sunmayı amaçlayan bir süreç olarak tanımlanabilir. Veri madenciliğinin birçok yaklaşımı vardır ve aşağıda gösterilmiştir.



4 temel metodu kısaca özetleyecek olursak:

Classification(Sınıflandırma):Bir veri setindeki her öğeyi önceden belirlenmiş bir grup veya sınıf kümesine yerleştirmek için kullanılır. Bu işlem, nesneleri özelliklerine göre gruplandırmayı ifade eder.

Regresyon analizi: Bir bağımlı değişken ile bir veya daha fazla bağımsız değişken arasındaki ilişkiyi açıklamak için kullanılır.(Bağımsız değişkenler, zaten bilinen özniteliklerdir.)

Association Rules(Birliktelik Kuralı): Veri setinde ilişkileri keşfetmek için kullanılır. Bir ilişki kuralının örnek ifadesi şöyledir: "Bir müşteri bir düzine yumurta satın alırsa, %80 ihtimalle süt de satın alır."

Clustering(Kümeleme):Devasa veri tabanlarında anlamlı örüntüleri tanımlamak için kullanılan, bilinen bir veri madenciliği yöntemidir.

K-MEANS KÜMELEME

K-Means, *unsupervised learning(gözetimsiz)* bir öğrenme algoritmasıdır. Bu kümeleme yönteminde, *supervised learning(gözetimli)* öğrenmenin aksine, önceden etiketlenmiş veri yoktur. Birbirine benzer ve diğer kümelerin elemanlarından farklı olan nesneleri kümelere ayırma işlemini gerçekleştirir. Aynı kümeye ait örüntüler, ayrı kümelere ait örüntülerden daha çok birbirine benzer. Burada 'K' ortaya çıkan yapıdaki küme sayısını temsil eder.

K-means kümeleme algoritması 2 aşamalı bir işlem kullanır:

İlk işlem, k değerinin sabit olduğu k merkezi rastgele seçmektir. İkinci işlem ise her veri nesnesini en yakın merkeze seçmektir. K-means algoritmaları genellikle veri nesneleri ile küme merkezleri arasındaki mesafeleri hesaplamak için Öklid uzaklığı kullanır.

Veri madenciliği uygulamalarında K-Means algoritmasının en önemli avantajı, büyük veri kümelerini kümelemede gösterdiği verimliliğidir.

VERİ SETİ HAKKINDA

Bu projede, bir İngiliz şirketine ait online perakende e-ticaret web sitesinin satışları hakkında olan UCI Machine Learning sitesinden elde edinilen "Online Retail" veri seti kümelendirilecektir. Kullanılan işlem verileri çoğunlukla İngiltere'de olmak üzere 2010 ve 2011 yıllarına aittir. Verilerin öznitelikleri ile ilgili UCI Machine Learning tablosundan alınan bilgi tablosu aşağıdaki gibidir:

Variables Table

Variable Name	Role	Type	Description
InvoiceNo	ID	Categorical	a 6-digit integral number uniquely assigned to each transaction. If this code starts with letter 'c', it indicates a cancellation
StockCode	ID	Categorical	a 5-digit integral number uniquely assigned to each distinct product
Description	Feature	Categorical	product name
Quantity	Feature	Integer	the quantities of each product (item) per transaction
InvoiceDate	Feature	Date	the day and time when each transaction was generated
UnitPrice	Feature	Continuous	product price per unit
CustomerID	Feature	Categorical	a 5-digit integral number uniquely assigned to each customer
Country	Feature	Categorical	the name of the country where each customer resides

METODOLOJİ

Bu projede, Jupyter Lab üzerinden Python'ın gerekli kütüphaneleri ile çalışarak kümeleme işlemi aşamalı ve pratik olarak uygulanmıştır.

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import datetime as dt
from sklearn.preprocessing import MinMaxScaler
from sklearn.cluster import KMeans
from sklearn.metrics import silhouette_score
```

Veri Çıkarma(Data Extraction)

İlk önce veriler Jupyter Lab üzerinde csv formatında okundu ve ilk hali Figure 3'teki gibiydi.

```
ds = pd.read_csv('Data/Online Retail.csv')
```

```
ds.head(n=len(ds))
```

	InvoiceNo	StockCode	Description	Quantity	InvoiceDate	UnitPrice	CustomerID	Country	Amount	Diff
0	536365	85123A	WHITE HANGING HEART T-LIGHT HOLDER	6	2010-12-01 08:26:00	2.55	17850.0	United Kingdom	15.30	373 days 04:24:00
1	536365	71053	WHITE METAL LANTERN	6	2010-12-01 08:26:00	3.39	17850.0	United Kingdom	20.34	373 days 04:24:00
2	536365	84406B	CREAM CUPID HEARTS COAT HANGER	8	2010-12-01 08:26:00	2.75	17850.0	United Kingdom	22.00	373 days 04:24:00
3	536365	84029G	KNITTED UNION FLAG HOT WATER BOTTLE	6	2010-12-01 08:26:00	3.39	17850.0	United Kingdom	20.34	373 days 04:24:00
4	536365	84029E	RED WOOLLY HOTTIE WHITE HEART.	6	2010-12-01 08:26:00	3.39	17850.0	United Kingdom	20.34	373 days 04:24:00
...	...	...	...	...	...	...	...	...	...	...
541904	581587	22613	PACK OF 20 SPACEBOY NAPKINS	12	2011-12-09 12:50:00	0.85	12680.0	France	10.20	0 days 00:00:00
541905	581587	22899	CHILDREN'S APRON DOLLY GIRL	6	2011-12-09 12:50:00	2.10	12680.0	France	12.60	0 days 00:00:00
541906	581587	23254	CHILDRENS CUTLERY DOLLY GIRL	4	2011-12-09 12:50:00	4.15	12680.0	France	16.60	0 days 00:00:00
541907	581587	23255	CHILDRENS CUTLERY CIRCUS PARADE	4	2011-12-09 12:50:00	4.15	12680.0	France	16.60	0 days 00:00:00
541908	581587	22138	BAKING SET 9 PIECE RETROSPOT	3	2011-12-09 12:50:00	4.95	12680.0	France	14.85	0 days 00:00:00

406829 rows × 10 columns

Veri Hazırlama(Data Processing)

Veri kümeleme sürecini engelleyecek olan veri setlerinin temizlenmesi ve işlenmesi gerekir. Veri ön işlemenin bir parçası olarak, verilerden anlamlı sonuçlar elde edebilmek için eksik değerleri(missing values) ve boş değer(null) içeren satırları temizleriz. Bu temizleme işlemi, veri setini "clean data" haline getirir.
Verilerin temizlenmesinden sonra, veri setindeki satır sayısı 541.909 satırdan 406829 satıra düşmüştür.

```
#data cleaning
print(ds.info())
print(ds.shape)
print(ds.isnull().sum())
ds = ds.dropna()
print(ds.info())
print(ds.shape)
```

RFM Modeli

RFM Modeli, bir organizasyonun en iyi müşterilerini tanımlamak için kullanılan bir müşteri analizi aracıdır. RFM Modelinin 3 temel faktörü vardır:

- **Yenilik (Recency):** Müşterinin en son ne zaman alışveriş yaptığı.
- **Sıklık (Frequency):** Müşterinin ne sıklıkla alışveriş yaptığı.
- **Tutar (Monetary):** Müşterinin alışveriş başına ortalama ne kadar harcama yaptığı.

Model, müşterileri bu üç kategorinin her birinde sıralar. Bu sıralama, pazarlama stratejilerinin en değerli müşterilere odaklanmasına yardımcı olur.

Öncelikle müşteri başına toplam miktar 'rfm_ds_m' dizisinde saklanacak şekilde hesaplanır ve ön işlenmiş veriler tabloda gösterilir:

```
#data preprocessing
ds['CustomerID'] = ds['CustomerID'].astype(str)
ds['Amount'] = ds['Quantity']*ds['UnitPrice']
rfm_ds_m = ds.groupby('CustomerID')['Amount'].sum()
rfm_ds_m.reset_index()
rfm_ds_m.columns = ['CustomerID', 'Amount']
print(rfm_ds_m)
```

```
CustomerID
12346.0       0.00
12347.0    4310.00
12348.0    1797.24
12349.0    1757.55
12350.0     334.40
            ...
18280.0     180.60
18281.0      80.82
18282.0     176.60
18283.0    2094.88
18287.0    1837.28
Name: Amount, Length: 4372, dtype: float64
```

Ardından müşteri başına fatura sayılarına göre ne kadar sıklıkla sipariş verdiği de 'rfm_ds_f' olarak hesaplanır ve tabloda gösterilir:

```
rfm_ds_f = ds.groupby('CustomerID')['InvoiceNo'].count()
rfm_ds_f = rfm_ds_f.reset_index()
rfm_ds_f.columns = ['CustomerID', 'Frequency']
print(rfm_ds_f)
```

```
     CustomerID  Frequency
0       12346.0          2
1       12347.0        182
2       12348.0         31
3       12349.0         73
4       12350.0         17
...         ...        ...
4367    18280.0         10
4368    18281.0          7
4369    18282.0         13
4370    18283.0        756
4371    18287.0         70

[4372 rows x 2 columns]
```

Ve son olarak faturalar tarih formatına çevrilerek en son satın alma zamanları hesaplanır. 'rfm_ds_p' dizisinden müşteri başına tutar belirlenir ve bu veriler de tabloda gösterilir.

```
ds['InvoiceDate'] = pd.to_datetime(ds['InvoiceDate'],format='mixed')
max_date = max(ds['InvoiceDate'])
ds['Diff'] = max_date - ds['InvoiceDate']
rfm_ds_p = ds.groupby('CustomerID')['Diff'].min()
rfm_ds_p = rfm_ds_p.reset_index()
rfm_ds_p.columns = ['CustomerID','Diff']
rfm_ds_p['Diff'] = rfm_ds_p['Diff'].dt.days
print(rfm_ds_p)
```

```
     CustomerID  Diff
0       12346.0   325
1       12347.0     1
2       12348.0    74
3       12349.0    18
4       12350.0   309
...         ...   ...
4367    18280.0   277
4368    18281.0   180
4369    18282.0     7
4370    18283.0     3
4371    18287.0    42

[4372 rows x 2 columns]
```

Son olarak müşterilerin işlem veri kümesinin tablo görüntüsü de aşağıda verilmiştir:

```
rfm_ds_final = pd.merge(rfm_ds_m,rfm_ds_f,on='CustomerID',how='inner')
rfm_ds_final = pd.merge(rfm_ds_final,rfm_ds_p,on='CustomerID',how='inner')
rfm_ds_final.columns = ['CustomerID', 'Amount', 'Frequency', 'Recency']
print(rfm_ds_final.head())
```

```
   CustomerID   Amount  Frequency  Recency
0     12346.0     0.00          2      325
1     12347.0  4310.00        182        1
2     12348.0  1797.24         31       74
3     12349.0  1757.55         73       18
4     12350.0   334.40         17      309
```

Outliers

Wikipedia'nın tanımına göre, diğer gözlemlerden kayda değer derecede uzak olan gözleme aykırı veya uç değer denir. Aykırı gözlemler, veri setinin geri kalanından farklı davranır ve bu nedenle dikkat çeker.
Birçok istatistiksel test ve makine öğrenmesi algoritması aykırı değerlere karşı hassastır. Bu nedenle, aykırı gözlemlerin tespit edilip, gözden geçirilmesi ve duruma göre müdahale edilmesi gerekmektedir.

Aşağıda verilen kod bloğu, 'rfm_ds_final' adlı veri kümesinde bulunan aykırı değerleri belirleyebilmek için kullanılan istatistiksel bir yöntem olan IQR (Interquartile Range - Çeyreklik Arası Menzil) yöntemini kullanır.

```
Q1 = rfm_ds_final.Amount.quantile(0.05)
Q3 = rfm_ds_final.Amount.quantile(0.95)
IQR = Q3 - Q1
rfm_ds_final = rfm_ds_final[(rfm_ds_final.Amount >= Q1 - 1.5*IQR) & (rfm_ds_final.Amount <= Q3 + 1.5*IQR)]

Q1 = rfm_ds_final.Recency.quantile(0.05)
Q3 = rfm_ds_final.Recency.quantile(0.95)
IQR = Q3 - Q1
rfm_ds_final = rfm_ds_final[(rfm_ds_final.Recency >= Q1 - 1.5*IQR) & (rfm_ds_final.Recency <= Q3 + 1.5*IQR)]

Q1 = rfm_ds_final.Frequency.quantile(0.05)
Q3 = rfm_ds_final.Frequency.quantile(0.95)
IQR = Q3 - Q1
rfm_ds_final = rfm_ds_final[(rfm_ds_final.Frequency >= Q1 - 1.5*IQR) & (rfm_ds_final.Frequency <= Q3 + 1.5*IQR)]
```

```
print(rfm_ds_final.shape)
```

```
(4293, 4)
```

Elde edilen son veri seti ise oldukça işlemden geçip hem müşteri sayısı hem de özellik sayısı ilk haline göre oldukça sadeleşmiştir.

Feature Scaling

Veri kümesinde farklı özellikler farklı ölçeklerde olabilir veya kullanılan algoritmalar hassas olabilir. Normalizasyon ve Standardizasyon ölçeklendirme yapmakta kullanılan iki temel tekniktir.

Bu çalışmada, Min-Max Scaler(Normalizasyon) tekniği kullanılmıştır.

```
#scaling
X = rfm_ds_final[['Amount', 'Frequency', 'Recency']]
scaler = MinMaxScaler()
rfm_ds_scaled = scaler.fit_transform(X)
```

```
rfm_ds_scaled = pd.DataFrame(rfm_ds_scaled)
rfm_ds_scaled.columns = ['Amount', 'Frequency', 'Recency']
rfm_ds_scaled.head()
```

	Amount	Frequency	Recency
0	0.238663	0.001395	0.871314
1	0.478571	0.252441	0.002681
2	0.338703	0.041841	0.198391
3	0.336494	0.100418	0.048257
4	0.257277	0.022315	0.828418

En son ölçeklenen verilerin ‘rfm_ds_scaled’ veri setinin ilk beş satırı tabloda gösterilir.

Clustering

Veriler artık kümelenebilir bir duruma geldiğinden ‘rfm_ds_scaled’ veri kümesindeki müşterileri K-Means algoritmasıyla 3 kümeye ayrılır.
K-means modeli eğitilir; model her bir veri noktasını hangi kümeye atacağını öğrenir. Ve her bir küme etiketi bir numpy dizisine atanır. Ve en son satırda ise bu dizideki etiketler yazdırılır:

```
#model creation
kmeans = KMeans(n_clusters=3, max_iter=50)
kmeans.fit(rfm_ds_scaled)
lbs = kmeans.labels_
print(kmeans.labels_)
```

```
[1 2 0 ... 1 0 0]
```

Elbow Metodu

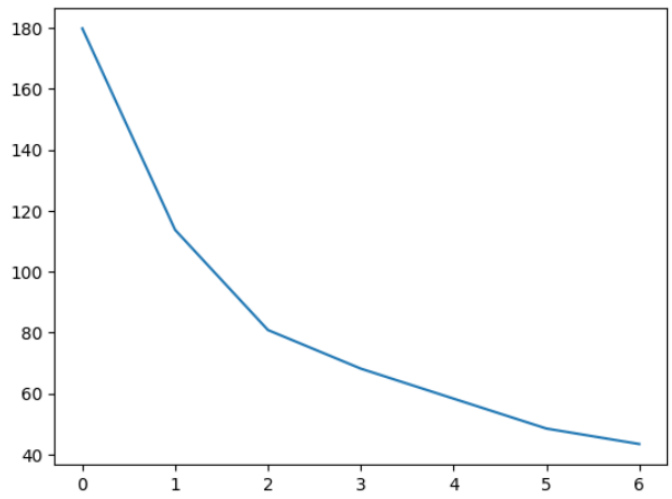
Elbow metodu, belirlediğimiz aralıktaki her 'k' değeri için verilerin küme merkezlerine uzaklıklarının karesinin toplamını grafiksel olarak gözlemlememizi sağlar. Grafik üzerinde toplamlar arasındaki farkın azalmaya başladığı dirsek noktası en uygun 'k' değeri seçilir.

Dirsek yöntemine göre, grafiğin kırılma noktalarından en uygun sayının 3 veya 4 olduğu söylenebilir.

```
#wsslistesi - Within-Cluster Sum of Squares
wss = []
range_n_clusters = [2, 3, 4, 5, 6, 7, 8]
for num_clusters in range_n_clusters:
    kmeans = KMeans(n_clusters=num_clusters, max_iter=50)
    kmeans.fit(rfm_ds_scaled)
    wss.append(kmeans.inertia_)

plt.plot(wss)
```

```
[<matplotlib.lines.Line2D at 0x1dae55f2810>]
```



Silhoutte Score Nedir?

Silhouette skor değerini hesaplamak için kullanılan yöntemde; herhangi bir k değeri için, veri setinde yer alan her veri için Silhouette değeri hesaplanır. Bu değerlerin ortalaması da k değeri için Silhouette skor değerini verir. Aşağıda bu amacı sağlayan döngü yer almaktadır:

```
#silhouette score
range_n_clusters = [2, 3, 4, 5, 6, 7, 8]
for num_clusters in range_n_clusters:
    kmeans = KMeans(n_clusters=num_clusters, max_iter=50)
    kmeans.fit(rfm_ds_scaled)
    cluster_labels = kmeans.labels_
    silhouette_avg = silhouette_score(rfm_ds_scaled, cluster_labels)
    print("For n_clusters={0}, the silhouette score is {1}".format(num_clusters, silhouette_avg))
```

```
For n_clusters=2, the silhouette score is 0.5889064222823147
For n_clusters=3, the silhouette score is 0.5465877794368479
For n_clusters=4, the silhouette score is 0.5041317149907137
For n_clusters=5, the silhouette score is 0.387474377529309
For n_clusters=6, the silhouette score is 0.4040369658704452
For n_clusters=7, the silhouette score is 0.4025844830615727
For n_clusters=8, the silhouette score is 0.389767244090359
```

Uygun küme sayısı aynı zamanda en yüksek siluet değerine sahip olan küme sayısıdır. Dolayısıyla k değeri 3 olarak belirlenir.

Son olarak ‘rfm_ds_final’ veri kümesi müşteri başına değerleri tabloda gösterilir:

```
rfm_ds_final['Cluster_Id'] = lbs
rfm_ds_final.head(n=len(rfm_ds_final))
```

	CustomerID	Amount	Frequency	Recency	Cluster_Id
0	12346.0	0.00	2	325	0
1	12347.0	4310.00	182	1	1
2	12348.0	1797.24	31	74	2
3	12349.0	1757.55	73	18	2
4	12350.0	334.40	17	309	0
...	...	...	...	...	...
4366	18278.0	173.90	9	73	2
4367	18280.0	180.60	10	277	0
4368	18281.0	80.82	7	180	0
4369	18282.0	176.60	13	7	2
4371	18287.0	1837.28	70	42	2

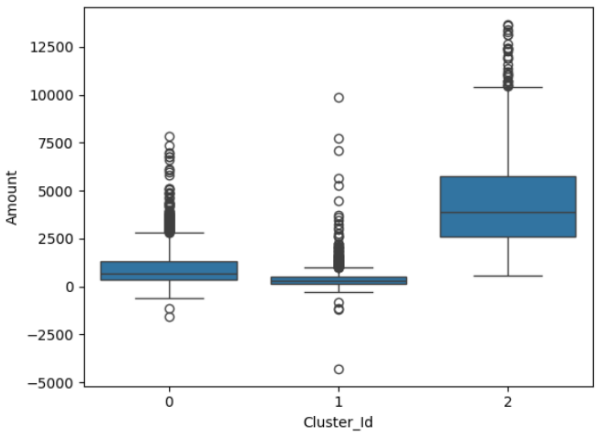
4293 rows × 5 columns

Veri görselleştirme(Data Visualization)

Aşağıda müşterilerin özelliklerine göre kümelenmiş son hali box-plot gösterimi ile modellenmiştir:

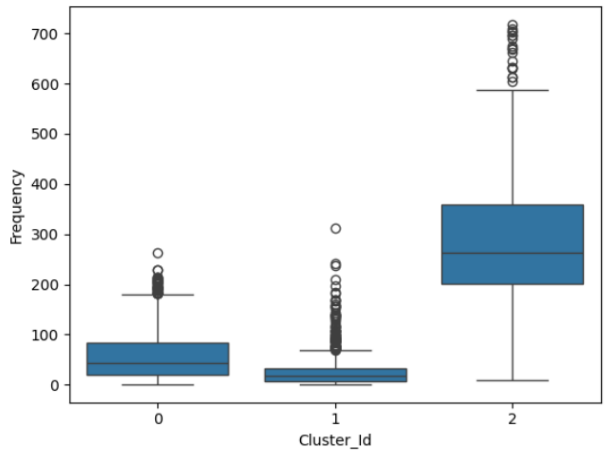
```
sns.boxplot(x='Cluster_Id', y='Amount', data=rfm_ds_final)
```

```
<Axes: xlabel='Cluster_Id', ylabel='Amount'>
```



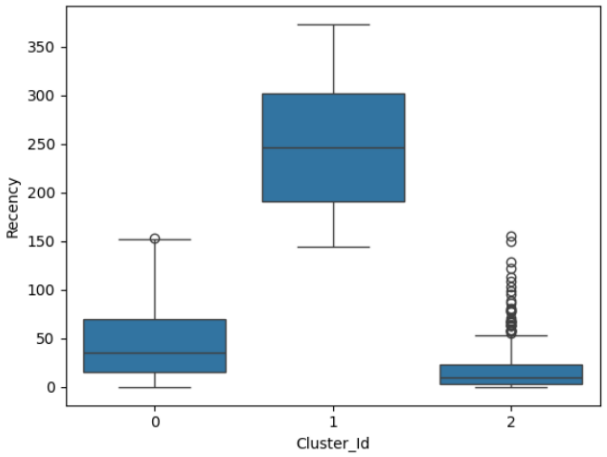
```
sns.boxplot(x='Cluster_Id', y='Frequency', data=rfm_ds_final)
```

```
<Axes: xlabel='Cluster_Id', ylabel='Frequency'>
```




```
sns.boxplot(x='Cluster_Id', y='Recency', data=rfm_ds_final)
```

```
<Axes: xlabel='Cluster_Id', ylabel='Recency'>
```



Sonuçlar

Online perakende mağazasının transaction(işlem) verileri kullanılarak RFM modelini ve K-Means algoritmasını temel alan müşteri segmentasyonu çalışmasına göre, müşterileri özellikleri bazında 3 kümeye ayırdık. Bu 3 kümeyi 0,1,2 olarak numaralandırdık. Cluster 2 en yüksek geliri, Cluster 1 ise en düşük geliri oluşturur. Sadık müşteri olarak Cluster 2'nin sık sipariş verdiğini de anlayabiliriz. Cluster 0'ı ise çok gelir getirmeyen ve düşük sadakate sahip küme olarak düşünebiliriz. Elde ettiğimiz sonuca göre, silhouette değeri 0,546 bulunmuştur, bu da verilen veri seti için iyi kabul edilir.

Müşteri segmentasyonu, şirketlere pazarlama stratejileri geliştirmesine ve sadık müşterilerine promosyon aracı olarak kullanılmasına yardımcı olabilir. Bu ve benzeri çalışmalar daha farklı araçlarla da desteklenebilir.

Akademik Makale Karşılaştırması

UCI Machine Learning sitesinden elde ettiğimiz Online Retail veri seti ile çalışılmış ve RFM modeli çıkartılarak K-means algoritması uygulanmış bir akademik çalışmanın sonuçları ile bu çalışmanın sonuçları karşılaştırılacaktır.

Makale Adı: “Customer Segmentation using RFM Model and K-Means Clustering”,Rahul Shirole, Laxmiputra Salokhe, Saraswati Jadhav, Department of Computer Engineering, Vishwakarma Institute of Technology Pune, Maharashtra, India, International Journal of Scientific Research in Science and Technology Volume 8 (June 2021)

Makale Linki: https://ijsrst.com/home/issue/view/article.php?id=IJSRST2183118

Bu makalenin analizine göre aynı veri setinde RFM modeli uygulanmış olup 3 farklı k sayısında kümeleme elde edilmiş. Ama en uygunun 4 küme olduğu belirtilmiş.

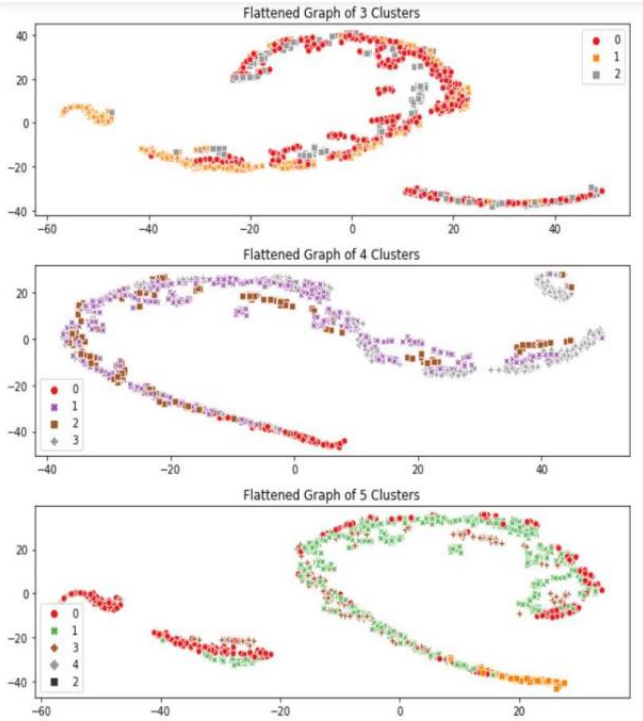


Figure 6 Clustering

Sonuçlarına göre; Bu 4 küme temel olarak A Sınıfı, B Sınıfı, C Sınıfı ve D Sınıfı'dır. A Sınıfı en yüksek geliri, D Sınıfı ise en düşük geliri oluşturmuştur. Silhouette değeri 0,442 bulunduğu belirtiliyor. Yukarıdaki gösterime göre düşük gelir getiren ve sık sipariş veren sadık müşteriler olduğu kadar tam tersi müşteriler de bulunuyor. Sonuçları bu çalışma ile birebir direkt karşılaştırmak pek mümkün değil model eğitimi ve görselleştirme farklar barındırıyor fakat aynı veri setinde benzer istatistik değerlere ulaşılmış, veriyi ön işlemeden geçirme süreçleri, normalizasyon sonucunda verilen histogram ile bu çalışmadaki değerler örtüşüyor.

Kaynaklar

- https://archive.ics.uci.edu/dataset/352/online+retail
- https://ravenfo.com/2021/02/11/aykiri-deger-analizi/
- https://medium.com/@anilguven1055/k-means-k%C3%BCmeleme-y%C3%B6ntemi-ed1c537046c6
- https://medium.com/@kizilaslandeniz/feature-scaling-veri-i%CC%87%C5%9Fleme-tekni%C4%9Fi-3b847e3e71cb
- https://www.aasmr.org/jsms/Vol13/No.6/Vol.13%20No.6.06.pdf
- https://gemini.google.com/app

Sunum Video Linki: https://youtu.be/ikYx8ST5ATE