1) Kütüphenelerimizi tanımlayarak başlıyoruz.

```
!pip install apyori

import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import matplotlib
matplotlib.use('Qt5Agg')
import warnings

warnings.simplefilter(action="ignore")
```

2) Keşifsel Veri Analizi bölümüyle devam ediyor. Veri setini tanıyoruz.

```
## Keşifsel Veri Analizi
df = pd.read_csv("datasets/Groceries_dataset.csv")
df.head()
df.info() # 1 nümerik, 2 kategorik değişkenimiz var.
#Bunlardan "Member_number" nümerik olan değişkenimiz.

df.isnull().sum() #Null değerlerimiz var mı diye kontrol ediyoruz.
```

Çıktılarımız :

```
In [4]: df.head()
Out[4]:
   Member_number        Date   itemDescription
0           1808  21-07-2015    tropical fruit
1           2552  05-01-2015        whole milk
2           2300  19-09-2015         pip fruit
3           1187  12-12-2015  other vegetables
4           3037  01-02-2015        whole milk
In [5]: df.info() # 1 nümerik, 2 kategorik değişkenimiz var.
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 38765 entries, 0 to 38764
Data columns (total 3 columns):
 #   Column           Non-Null Count  Dtype
---  ------           --------------  -----
 0   Member_number    38765 non-null  int64
 1   Date             38765 non-null  object
 2   itemDescription  38765 non-null  object
dtypes: int64(1), object(2)
memory usage: 908.7+ KB
In [6]: df.isnull().sum() #Null değerlerimiz var mı diye kontrol ediyoruz.
Out[6]:
Member_number      0
Date               0
itemDescription    0
dtype: int64
```

+ Date sütünümuzun object tipinde olduğunu görüyoruz, tarih formatına almalıyız.

+ Hiç boş (null) değişkenimizin olmadığını gözlemliyoruz.

3) Veri Ön İşleme ile değişikliklerimizi yapıyoruz.

```
## Veri Ön-İşleme
df['Date'] = pd.to_datetime(df['Date']) # Date sütunumuz kategorikti, Tarih türünde olmalı. Bu yüzden type 'ını değişmeliyiz.
df.info()
```

Çıktımız :

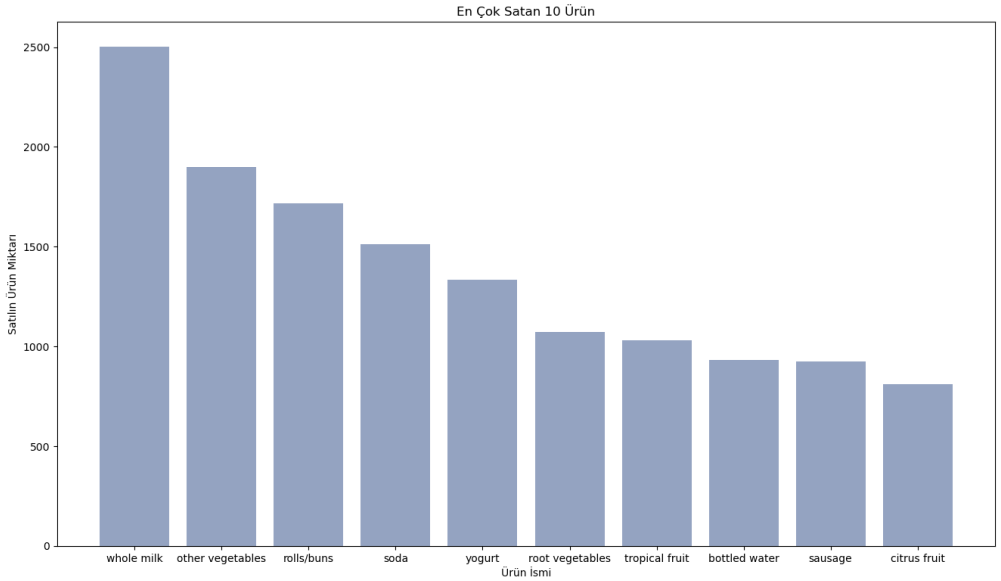
```
In [8]: df.info()
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 38765 entries, 0 to 38764
Data columns (total 3 columns):
 #   Column           Non-Null Count  Dtype
---  ------           --------------  -----
 0   Member_number    38765 non-null  int64
 1   Date             38765 non-null  datetime64[ns]
 2   itemDescription  38765 non-null  object
dtypes: datetime64[ns](1), int64(1), object(1)
memory usage: 908.7+ KB
```

4) Veri setimizi incelemeye devam ediyoruz.

```
## En Çok Satan 10 Ürünün İncelenmesi
EnCokSatanlar = df.groupby(by = "itemDescription").size().reset_index(name='Frequency').sort_values(by = 'Frequency',ascending=False).head(10)
#En çok satan 10 ürünümüzü grupladık.

### GÖRSELLEŞTİRME ###
bars = EnCokSatanlar["itemDescription"]
height = EnCokSatanlar["Frequency"]
x_pos = np.arange(len(bars))
plt.figure(figsize=(16,9))
plt.bar(x_pos, height, color=(0.3, 0.4, 0.6, 0.6)) #Kategorik değişkenleri en iyi bar plot ile görselleştirebiliriz.
plt.title("En Çok Satan 10 Ürün") #Başlık ekledik.
plt.xlabel("Ürün İsmi") #x eksenini isimlendirdik.
plt.ylabel("Satılın Ürün Miktarı") #y eksenini isimlendirdik.
plt.xticks(x_pos, bars) #x ekseninin adlarını oluşturduk.
plt.show() #Görseli getirdik.
```

Çıktımız :



+ En çok satan ürünün açık ara farkla sütler (tüm süt türlerini gruplamıştık) olduğunu gözlemliyoruz.

+ En az satışı da narenciyelerde yapıldığını gözlemliyoruz.

5) Aylık-Yıllık satışlara bakarak veri setini incelemeye devam ediyoruz.

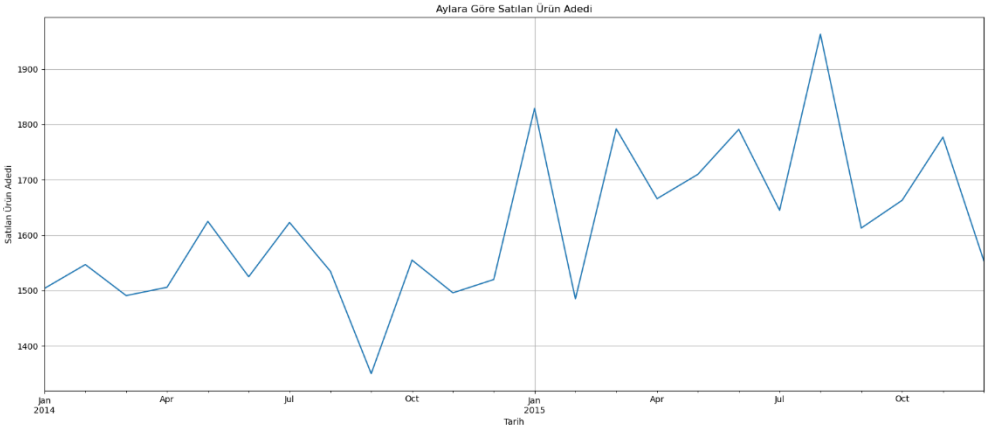
```
## Aylık-Yıllık Satışlar
df_date=df.set_index(['Date']) #Tarihi indeks olarak ayarlıyoruz. Daha sonra grafikte kullanacağız.
df_date

df_date.resample("M")['itemDescription'].count().plot(figsize = (20,8), grid = True, title = "Aylara Göre Satılan Ürün Adedi").set(xlabel = "Tarih", ylabel = "Satılan Ürün Adedi")
```

Çıktımız :

```
Out[22]:
            Member_number       itemDescription
Date
2015-07-21           1808         tropical fruit
2015-05-01           2552             whole milk
2015-09-19           2300              pip fruit
2015-12-12           1187       other vegetables
2015-01-02           3037             whole milk
                      ...                    ...
2014-08-10           4471          sliced cheese
2014-02-23           2022                  candy
2014-04-16           1097              cake bar
2014-03-12           1510  fruit/vegetable juice
2014-12-26           1521              cat food

[38765 rows x 2 columns]
```



+ En çok satışın 2015 Ağustos ayında yapıldığını ve hemen ardından Eylül ayında büyük bir düşüş yaşadığını gözlemliyoruz.

+ En az satışın da 2014 Eylülde yapıldığını gözlemliyoruz. Böylece rahatlıkla Eylül aylarında “işler kesat” diyebiliriz.

6) Şimdi birliktelik analizi için veriyi hazırlıyoruz.

```
## Şimdi Birliktelik Analizi İçin Veriyi Hazırlayalım
Birliktelik_modeli = df[["Member_number", "itemDescription"]].sort_values(by = "Member_number", ascending = False)
#Modelleme için gerekli değişkenleri seçeceğiz.
Birliktelik_modeli['itemDescription'] = Birliktelik_modeli['itemDescription'].str.strip()
#Varsa boşlukların kaldırıyoruz.
Birliktelik_modeli

transactions = [a[1]['itemDescription'].tolist() for a in list(Birliktelik_modeli.groupby(['Member_number']))]
#Her müşteri için tüm ürünleri bir listeye atıyoruz.
```

Çıktımız :

```
In [26]: Birliktelik_modeli
Out[26]:
       Member_number        itemDescription
3578            5000                   soda
34885           5000    semi-finished bread
11728           5000  fruit/vegetable juice
9340            5000          bottled beer
19727           5000        root vegetables
              ...                    ...
13331           1000             whole milk
17778           1000     pickled vegetables
6388            1000               sausage
20992           1000    semi-finished bread
8395            1000             whole milk

[38765 rows x 2 columns]
```

7) Modelimizi kuruyoruz.

```
## Modelin Kurulması
from apyori import apriori
rules = apriori(transactions = transactions, min_support = 0.002, min_confidence = 0.05, min_lift = 3, min_length = 2, max_length = 2)
#Modeli kurduk.

Sonuc = list(rules) #Sonuçları daha iyi okuyabilmek için listeliyoruz.
Sonuc #Hala pek okunabilir değiller.
```

Çıktımız :

```
In [31]: Sonuc #Hala pek okunabilir değiller.
Out[31]:
[RelationRecord(items=frozenset({'UHT-milk', 'kitchen towels'}), support=0.002308876346844536, ordered_statistics=[OrderedStatistic(items_base=frozenset({'kitchen towels'}), items_add=frozenset({'UHT-milk
 RelationRecord(items=frozenset({'beef', 'potato products'}), support=0.002565418163160595, ordered_statistics=[OrderedStatistic(items_base=frozenset({'potato products'}), items_add=frozenset({'beef'}), c
 RelationRecord(items=frozenset({'canned fruit', 'coffee'}), support=0.002308876346844536, ordered_statistics=[OrderedStatistic(items_base=frozenset({'canned fruit'}), items_add=frozenset({'coffee'}), con
 RelationRecord(items=frozenset({'domestic eggs', 'meat spreads'}), support=0.003591585428424833 4, ordered_statistics=[OrderedStatistic(items_base=frozenset({'meat spreads'}), items_add=frozenset({'domest
 RelationRecord(items=frozenset({'flour', 'mayonnaise'}), support=0.002308876346844536, ordered_statistics=[OrderedStatistic(items_base=frozenset({'flour'}), items_add=frozenset({'mayonnaise'}), confidenc
 RelationRecord(items=frozenset({'napkins', 'rice'}), support=0.003078501795792714 3, ordered_statistics=[OrderedStatistic(items_base=frozenset({'rice'}), items_add=frozenset({'napkins'}), confidence=0.244
 RelationRecord(items=frozenset({'sparkling wine', 'waffles'}), support=0.002565418163160595, ordered_statistics=[OrderedStatistic(items_base=frozenset({'sparkling wine'}), items_add=frozenset({'waffles'}
```

+ Çıktımız pek okunaklı olmadı, bu yüzden bir işlem daha yapacağız.

```
def inspect(results):
    lhs         = [tuple(result[2][0][0])[0] for result in results]
    rhs         = [tuple(result[2][0][1])[0] for result in results]
    supports    = [result[1] for result in results]
    confidences = [result[2][0][2] for result in results]
    lifts       = [result[2][0][3] for result in results]
    return list(zip(lhs, rhs, supports, confidences, lifts))

SonuclarDataFrame = pd.DataFrame(inspect(Sonuc), columns = ['Left Hand Side', 'Right Hand Side', 'Support', 'Confidence', 'Lift'])

SonuclarDataFrame.nlargest(n=10, columns="Lift") #Mümkün olan en iyi senaryoyu gösterir.
```

Sonuçlarımız :

```
In [37]: SonuclarDataFrame.nlargest(n=10, columns="Lift") #Mümkün olan en iyi senaryoyu gösterir.
Out[37]:
    Left Hand Side Right Hand Side   Support  Confidence      Lift
0   kitchen towels        UHT-milk  0.002309    0.300000  3.821569
1  potato products            beef  0.002565    0.454545  3.802185
2     canned fruit          coffee  0.002309    0.428571  3.728954
4            flour      mayonnaise  0.002309    0.063380  3.338599
6   sparkling wine         waffles  0.002565    0.217391  3.150154
5             rice         napkins  0.003079    0.244898  3.011395
3     meat spreads   domestic eggs  0.003592    0.400000  3.004239
```

+ Artık yorumlamak çok daha kolay.

Sonuçlarımızın Yorumlanması :

Örnek olarak 0. İndeksi alarak yorumlarımı yapacağım. Yaptığım yorumlar diğer tüm satırlar için de uygulanabilir.

+ “kitchen towels” ve “UHT-milk” ürünlerinin birlikte görülme olasılıklarının (Support) %0.02 olduğunu,

+ “kitchen towels” alan kişilerin (Confidence) %30 olasılıkla “UHT-milk” satın aldığını,

+ “kitchen towels” ürününün yer aldığı bir alışveriş sepetinde “UHT-milk” in satışlarının (lift) 3,82 kat arttığını söyleyebiliriz.

Ali Kerem Şimşek

Veri Madenciliği Dersi Ödev Raporu.