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

```
data = pd.read_excel("ecom customer_data.xlsx")
data.head()
```

	Cust_ID	Gender	Orders	Jordan	Gatorade	Samsung	Asus	Udis	Mondelez International	Wrangler	...	LG	Dior	Scabal	Tommy Hilfiger	Hollister	Forever 21	Colavita	Microsoft	Jiffy mix	Kra
0	1	M	7	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	
1	2	F	0	0	1	0	0	0	0	0	...	0	1	0	0	0	0	0	0	0	
2	3	M	7	0	1	0	0	0	0	0	...	0	0	0	0	0	0	0	1	0	
3	4	F	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	
4	5	NaN	10	0	0	0	0	0	0	0	...	0	0	2	0	0	0	0	0	1	

5 rows × 38 columns

```
df=data.copy()
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 30000 entries, 0 to 29999
Data columns (total 38 columns):
 #   Column                  Non-Null Count  Dtype
---  ------                  --------------  -----
 0   Cust_ID                 30000 non-null  int64
 1   Gender                  27276 non-null  object
 2   Orders                  30000 non-null  int64
 3   Jordan                  30000 non-null  int64
 4   Gatorade                30000 non-null  int64
 5   Samsung                 30000 non-null  int64
 6   Asus                    30000 non-null  int64
 7   Udis                    30000 non-null  int64
 8   Mondelez International  30000 non-null  int64
 9   Wrangler                30000 non-null  int64
 10  Vans                    30000 non-null  int64
 11  Fila                    30000 non-null  int64
 12  Brooks                  30000 non-null  int64
 13  H&M                     30000 non-null  int64
 14  Dairy Queen             30000 non-null  int64
 15  Fendi                   30000 non-null  int64
```

```
 16  Hewlett Packard         30000 non-null  int64
 17  Pladis                  30000 non-null  int64
 18  Asics                   30000 non-null  int64
 19  Siemens                 30000 non-null  int64
 20  J.M. Smucker            30000 non-null  int64
 21  Pop Chips               30000 non-null  int64
 22  Juniper                 30000 non-null  int64
 23  Huawei                  30000 non-null  int64
 24  Compaq                  30000 non-null  int64
 25  IBM                     30000 non-null  int64
 26  Burberry                30000 non-null  int64
 27  Mi                      30000 non-null  int64
 28  LG                      30000 non-null  int64
 29  Dior                    30000 non-null  int64
 30  Scabal                  30000 non-null  int64
 31  Tommy Hilfiger          30000 non-null  int64
 32  Hollister               30000 non-null  int64
 33  Forever 21              30000 non-null  int64
 34  Colavita                30000 non-null  int64
 35  Microsoft               30000 non-null  int64
 36  Jiffy mix               30000 non-null  int64
 37  Kraft                   30000 non-null  int64
dtypes: int64(37), object(1)
memory usage: 8.7+ MB
```

```
df['Gender'] = df['Gender'].fillna(df['Gender'].mode()[0])
```

```
df.isna().sum().sum()
```

```
0
```

DATA VISUALIZATION

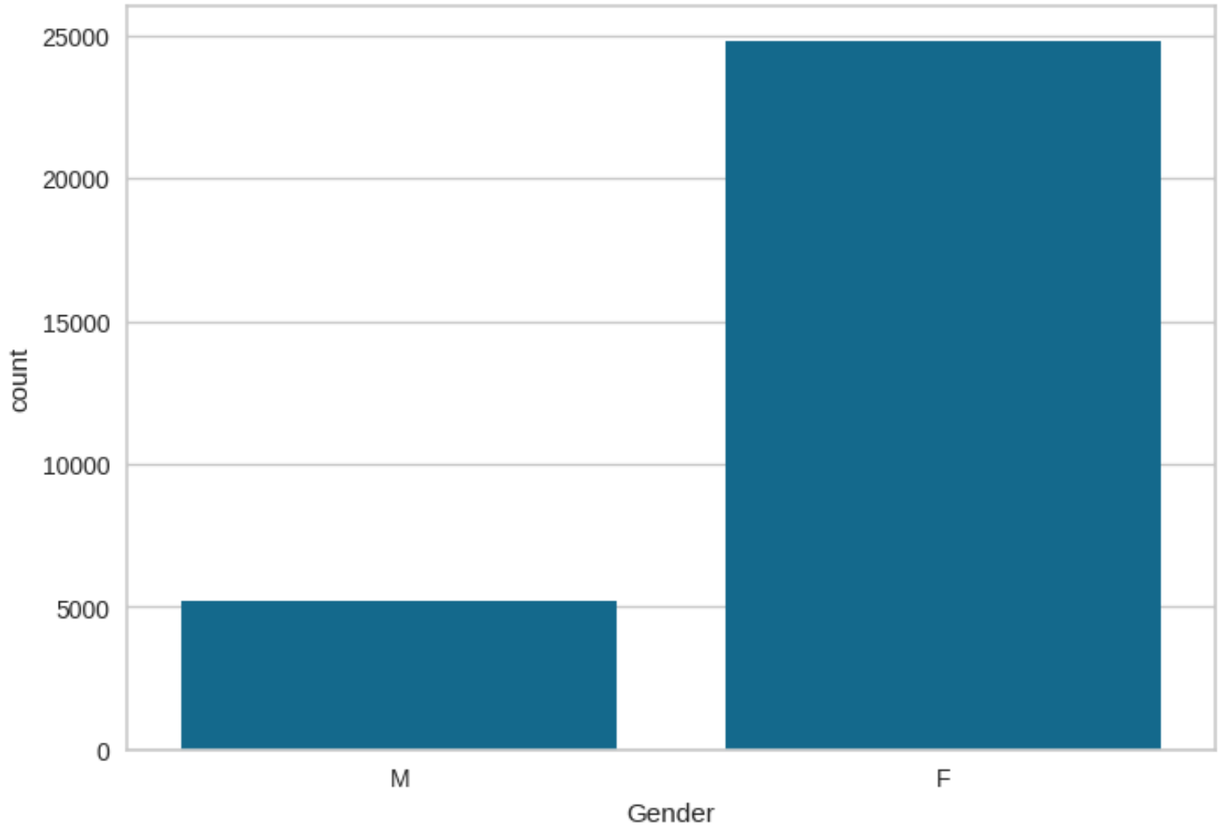
```
df.Gender.value_counts()
```

```
Gender
F    24778
M     5222
Name: count, dtype: int64
```

```
sns.countplot(data=df,x='Gender')
plt.show
```

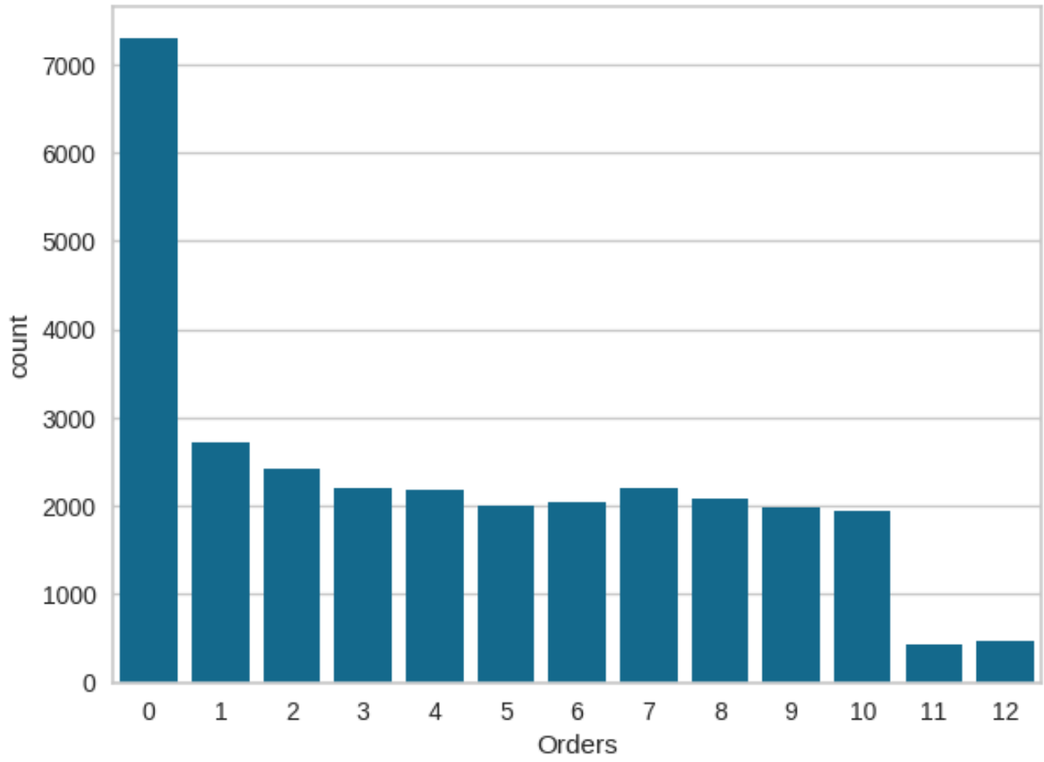
```
matplotlib.pyplot.show
def show(*args, **kwargs)
```

```
Display all open figures.

Parameters
----------
block : bool, optional
    Whether to wait for all figures to be closed before returning.
```



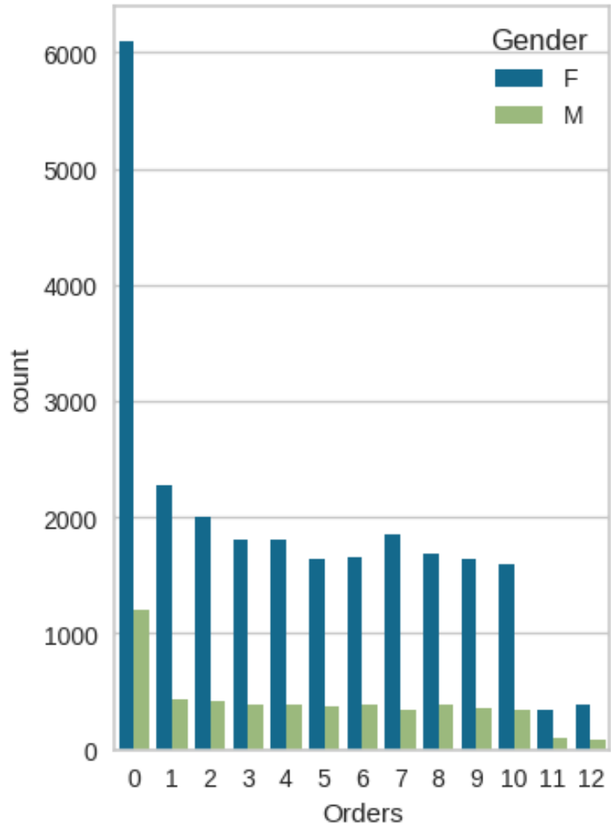

```
#Overall order count
plt.figure(figsize=(15,5))
plt.subplot(1,2,1)
sns.countplot(data=df,x='Orders')
```

```
<Axes: xlabel='Orders', ylabel='count'>
```

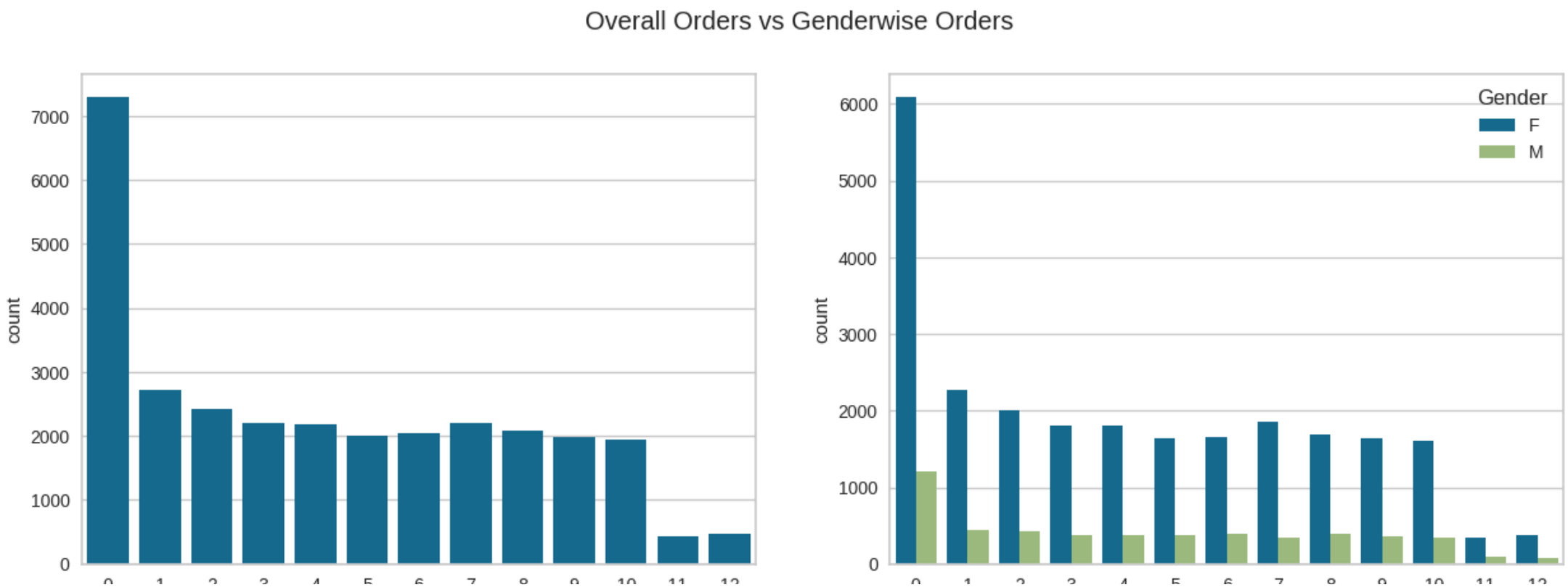



```
#Order count by each gender
plt.subplot(1,2,2)
sns.countplot(data=df,x='Orders',hue='Gender')
plt.suptitle("Overall Orders vs Genderwise Orders")
plt.show()
```

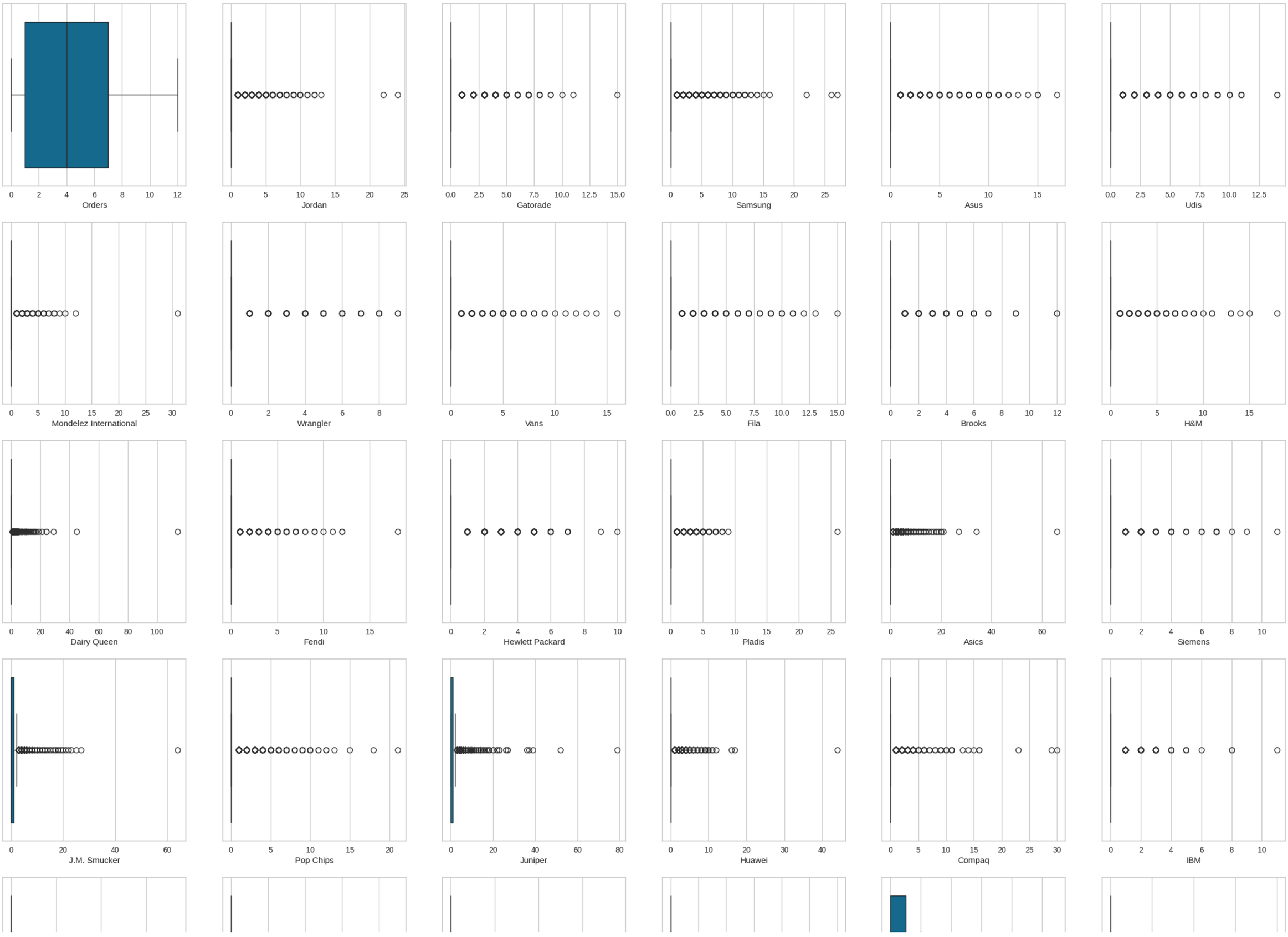
Overall Orders vs Genderwise Orders

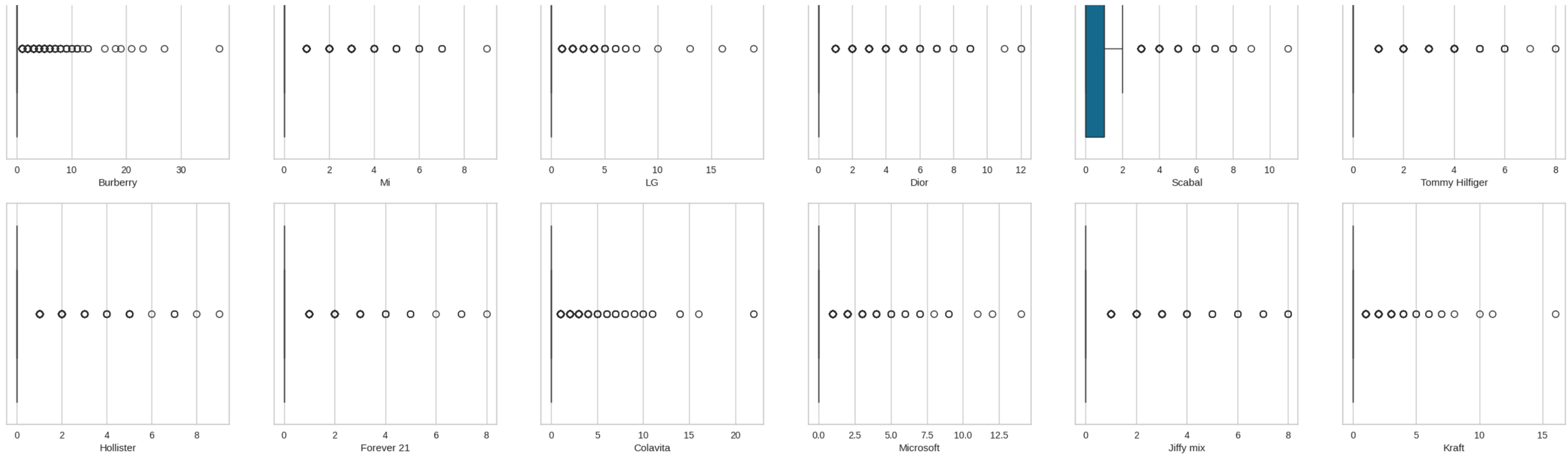



```
#Overall order count
plt.figure(figsize=(15,5))
plt.subplot(1,2,1)
sns.countplot(data=df,x='Orders')

#Order count by each gender
plt.subplot(1,2,2)
sns.countplot(data=df,x='Orders',hue='Gender')
plt.suptitle("Overall Orders vs Genderwise Orders")
plt.show()
```

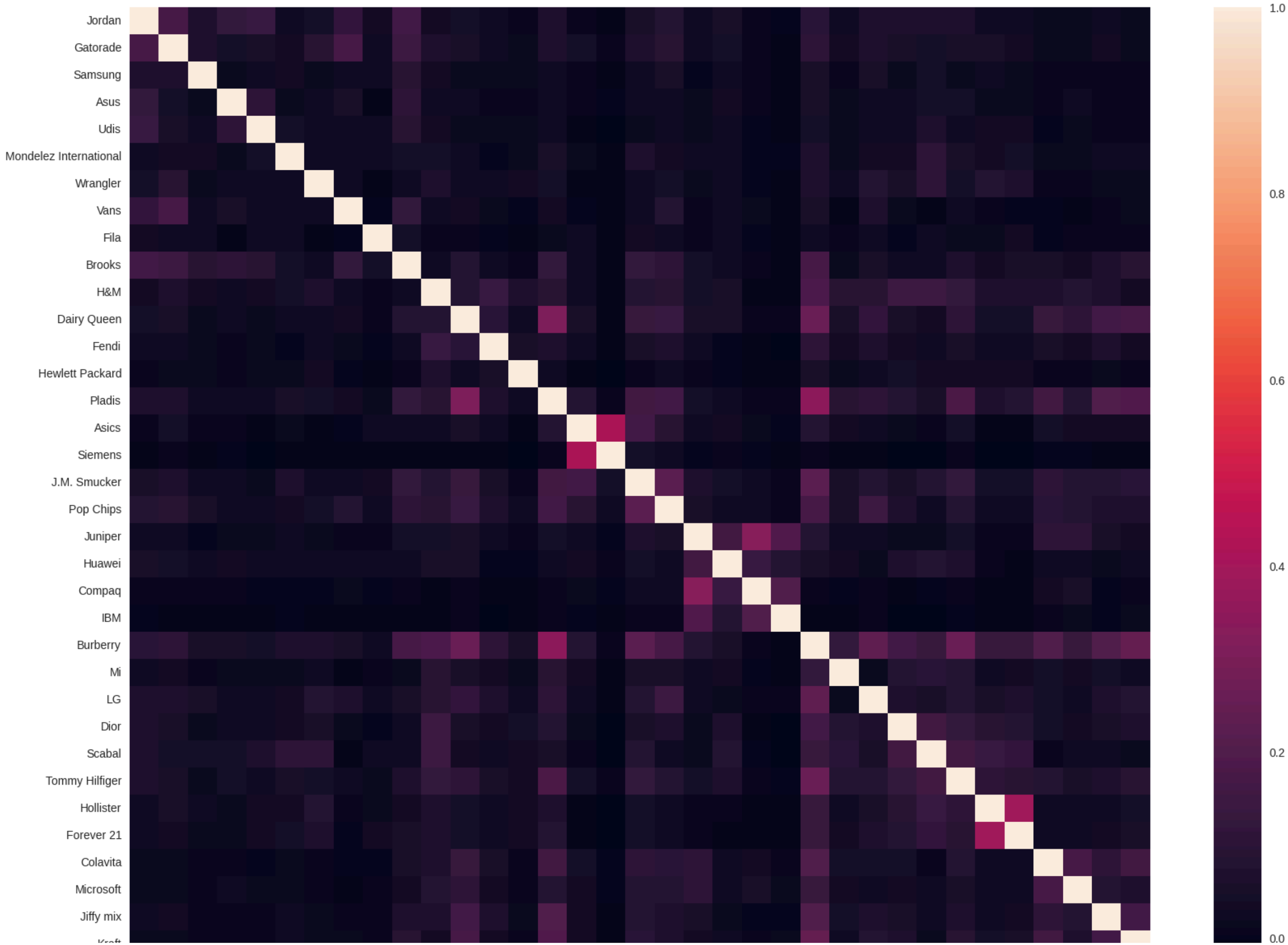


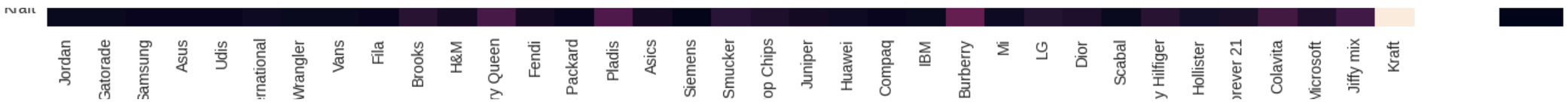

```
#Bocplot for orders and searches of each brands

cols=list(df.columns[2:])
def dist_list(lst):
  plt.figure(figsize=(30,30))
  for i,col in enumerate(lst,1):
    plt.subplot(6,6,i)
    sns.boxplot(data=df,x=df[col])
dist_list(cols)
```


Orders
Jordan
Gatorade
Samsung
Asus
Udis
Mondelez International
Wrangler
Vans
Fila
Brooks
H&M
Dairy Queen
Fendi
Hewlett Packard
Pladis
Asics
Siemens
J.M. Smucker
Pop Chips
Juniper
Huawei
Compaq
IBM

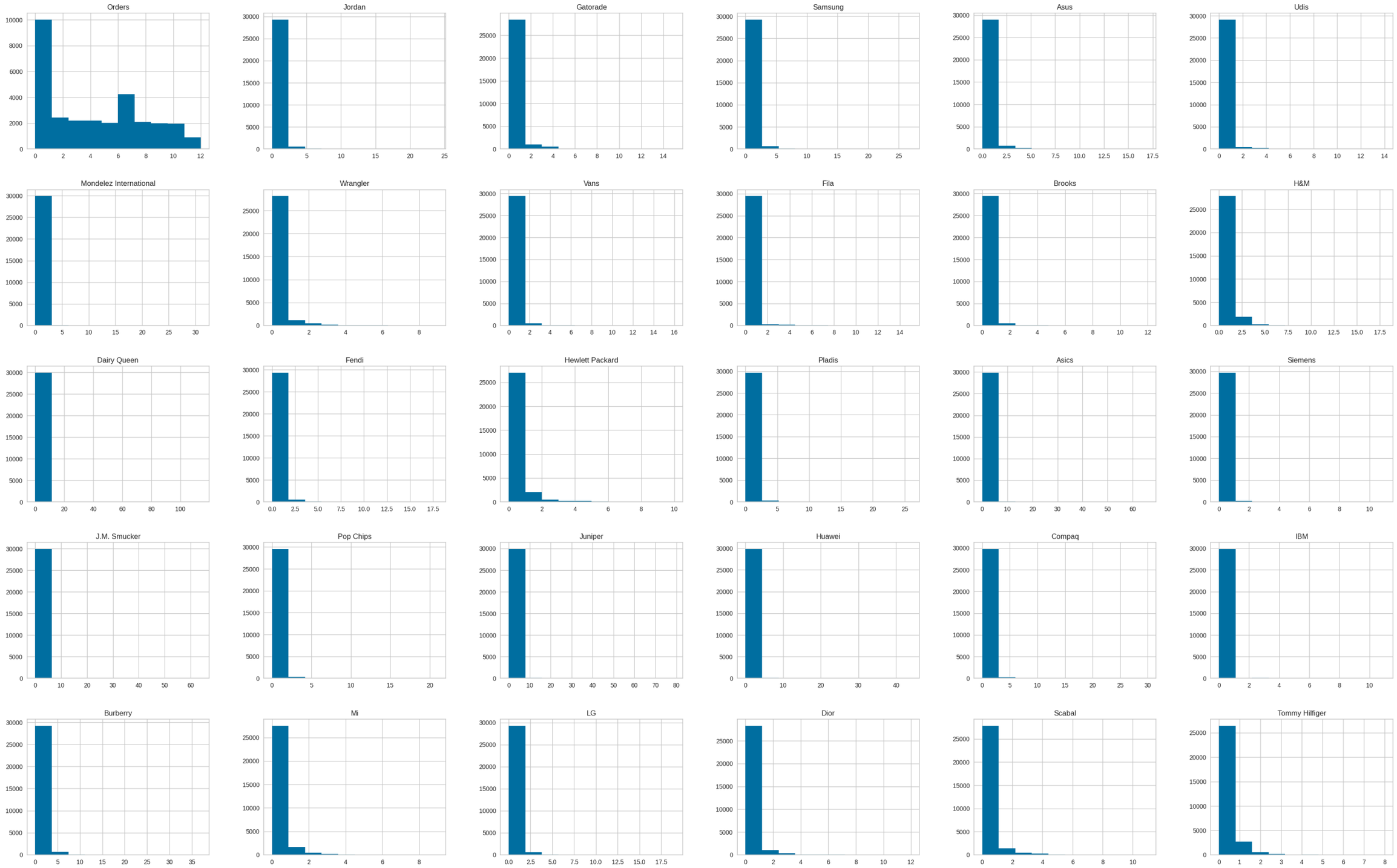



```
#Correlation Graph(Heatmap)
plt.figure(figsize=(20,15))
sns.heatmap(df.iloc[:,3:].corr())
plt.show()
```


Jordan
Gatorade
Samsung
Asus
Udis
Mondelez International
Wrangler
Vans
Fila
Brooks
H&M
Dairy Queen
Fendi
Hewlett Packard
Pladis
Asics
Siemens
J.M. Smucker
Pop Chips
Juniper
Huawei
Compaq
IBM
Burberry
Mi
LG
Dior
Scabal
Tommy Hilfiger
Hollister
Forever 21
Colavita
Microsoft
Jiffy mix
1.0
0.8
0.6
0.4
0.2
0.0




```
df.iloc[:,2:].hist(figsize=(40,30))
plt.show()
```



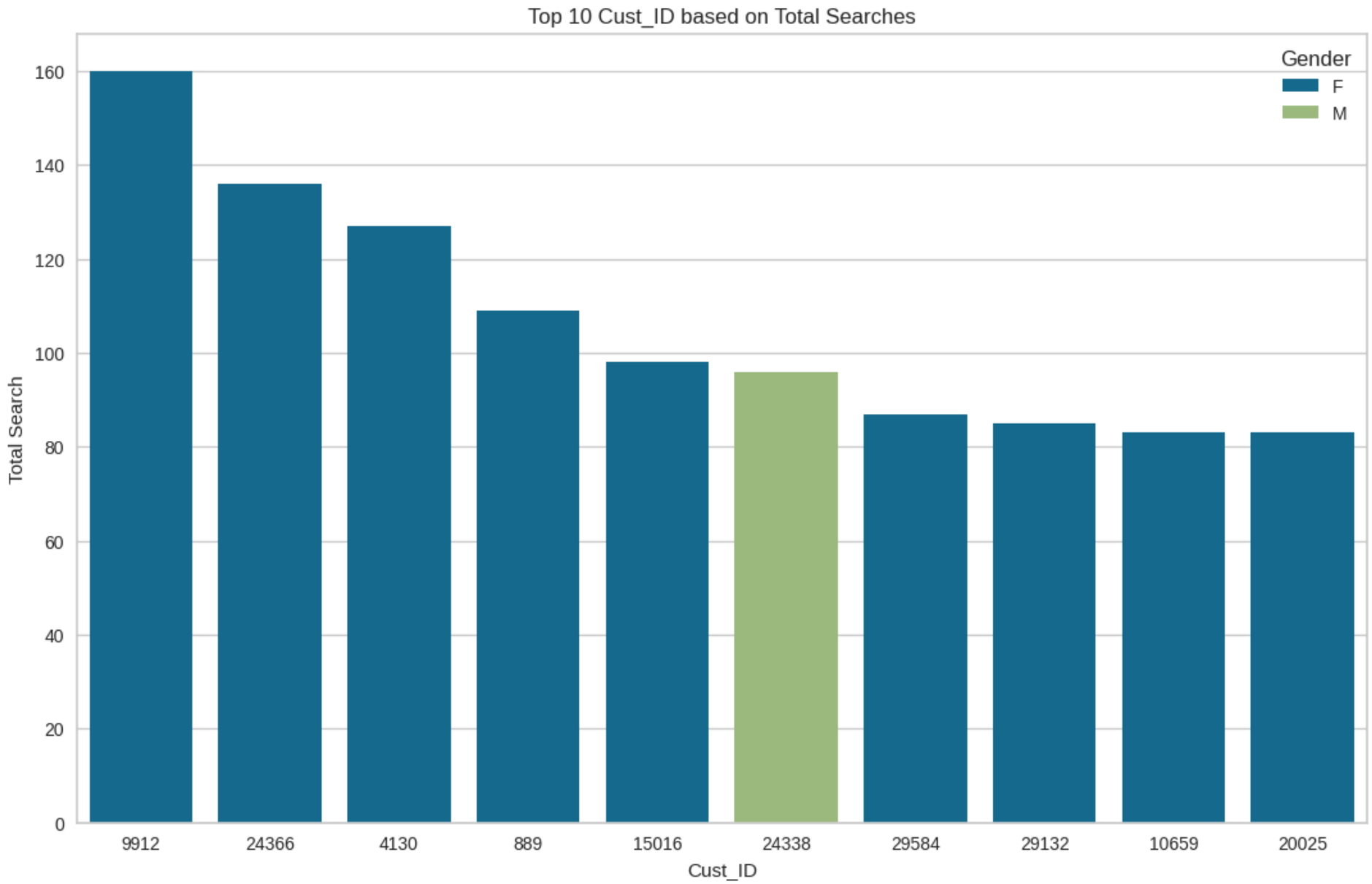

```
new_df=df.copy()
new_df['Total Search']=new_df.iloc[:,3:].sum(axis=1)
```

```
new_df.sort_values('Total Search', ascending=False)
```

	Cust_ID	Gender	Orders	Jordan	Gatorade	Samsung	Asus	Udis	Mondelez International	Wrangler	...	Dior	Scabal	Tommy Hilfiger	Hollister	Forever 21	Colavita	Microsoft	Jiffy mix	Kra
9911	9912	F	2	0	11	0	0	0	0	6	...	2	8	4	9	1	4	1	3	
24365	24366	F	2	3	3	2	2	0	1	2	...	2	2	6	4	1	4	3	3	
4129	4130	F	1	0	0	0	0	0	0	0	...	0	0	0	0	0	0	1	0	
888	889	F	0	2	1	3	2	0	1	3	...	3	1	5	3	2	5	1	3	
15015	15016	F	10	2	2	2	0	0	0	0	...	0	1	0	0	1	7	4	2	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
15000	15001	F	4	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	1	
7247	7248	F	3	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	
7255	7256	M	8	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	
7259	7260	F	6	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	
7263	7264	M	8	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	

30000 rows × 39 columns

```
plt.figure(figsize=(13,8))
plt_data = new_df.sort_values('Total Search',ascending=False)[['Cust_ID','Gender','Total Search']].head(10)
sns.barplot(data=plt_data,
            x='Cust_ID',
            y='Total Search',
            hue='Gender', order=plt_data.sort_values('Total Search',ascending = False).Cust_ID)
plt.title("Top 10 Cust_ID based on Total Searches")
plt.show()
```



SCALING

```
x = df.iloc[:,2: ].values
x
```

```
array([[7, 0, 0, ..., 0, 0, 0],
       [0, 0, 1, ..., 0, 0, 0],
       [7, 0, 1, ..., 1, 0, 0],
       ...,
       [0, 0, 1, ..., 0, 0, 0],
```

```
       [0, 0, 0, ..., 0, 0, 0],
       [3, 2, 0, ..., 0, 0, 0]])
```

```
scale = MinMaxScaler()
features = scale.fit_transform(x)
features
```

```
array([[0.58333333, 0.        , 0.        , ..., 0.        , 0.        ,
        0.        ],
       [0.        , 0.        , 0.06666667, ..., 0.        , 0.        ,
        0.        ],
       [0.58333333, 0.        , 0.06666667, ..., 0.07142857, 0.        ,
        0.        ],
       ...,
       [0.        , 0.        , 0.06666667, ..., 0.        , 0.        ,
        0.        ],
       [0.        , 0.        , 0.        , ..., 0.        , 0.        ,
        0.        ],
       [0.25      , 0.08333333, 0.        , ..., 0.        , 0.        ,
        0.        ]])
```

Elbow Method to get optimal k value

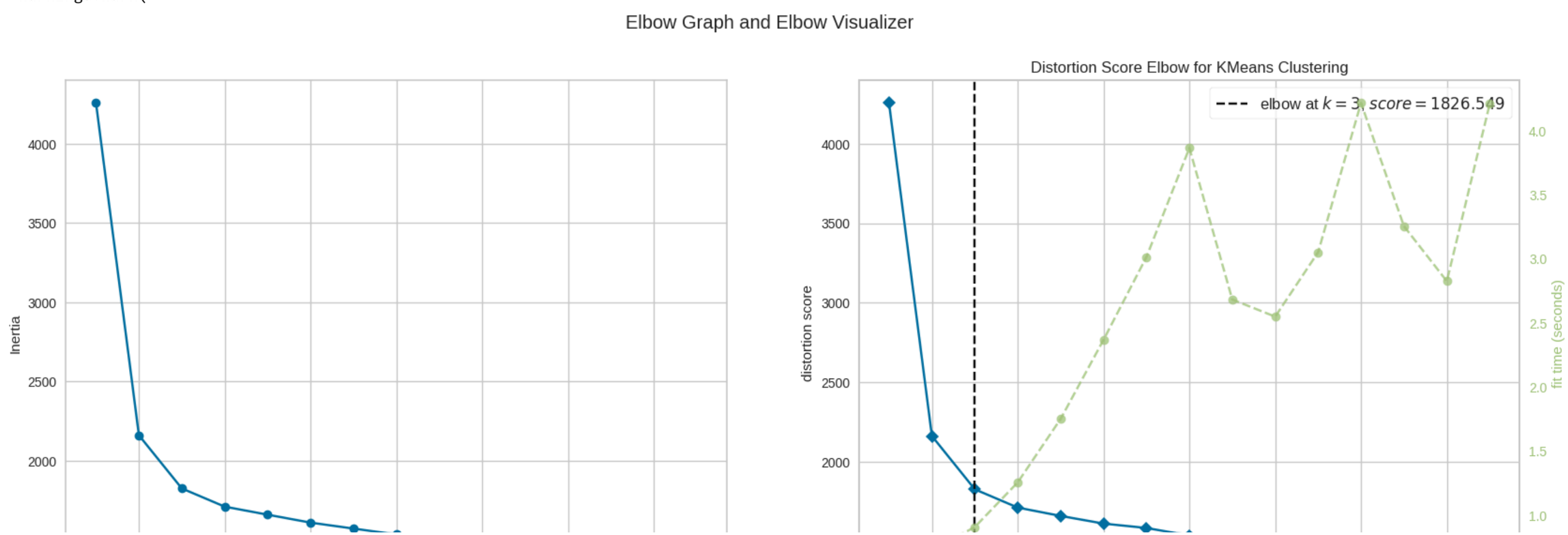
```
inertia = []
for i in range(1,16):
  k_means = KMeans(n_clusters=i)
  k_means = k_means.fit(features)
  inertia.append(k_means.inertia_)
```

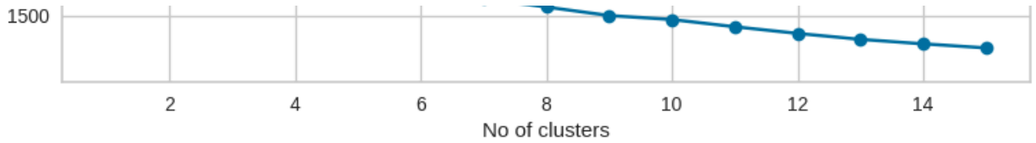
```
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning: The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of `
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning: The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of `
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning: The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of `
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning: The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of `
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning: The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of `
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning: The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of `
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning: The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of `
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning: The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of `
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning: The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of `
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning: The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of `
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning: The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of `
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning: The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of `
```

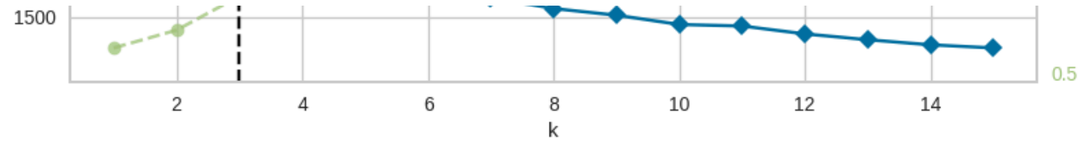
```
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning: The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of `
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning: The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of `
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning: The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of `
  warnings.warn(
```

```
# Elbow Graph
plt.figure(figsize=(20,7))
plt.subplot(1,2,1)
plt.plot(range(1,16), inertia, 'bo-')
plt.xlabel('No of clusters'), plt.ylabel('Inertia')

# Kelbow visualizer
plt.subplot(1,2,2)
kmeans = KMeans()
visualize = KElbowVisualizer(kmeans,k=(1,16))
visualize.fit(features)
plt.suptitle("Elbow Graph and Elbow Visualizer")
visualize.poof()
plt.show()
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning: The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning: The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning: The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning: The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning: The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning: The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning: The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning: The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning: The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning: The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning: The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning: The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning: The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning: The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning: The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of
  warnings.warn(
```





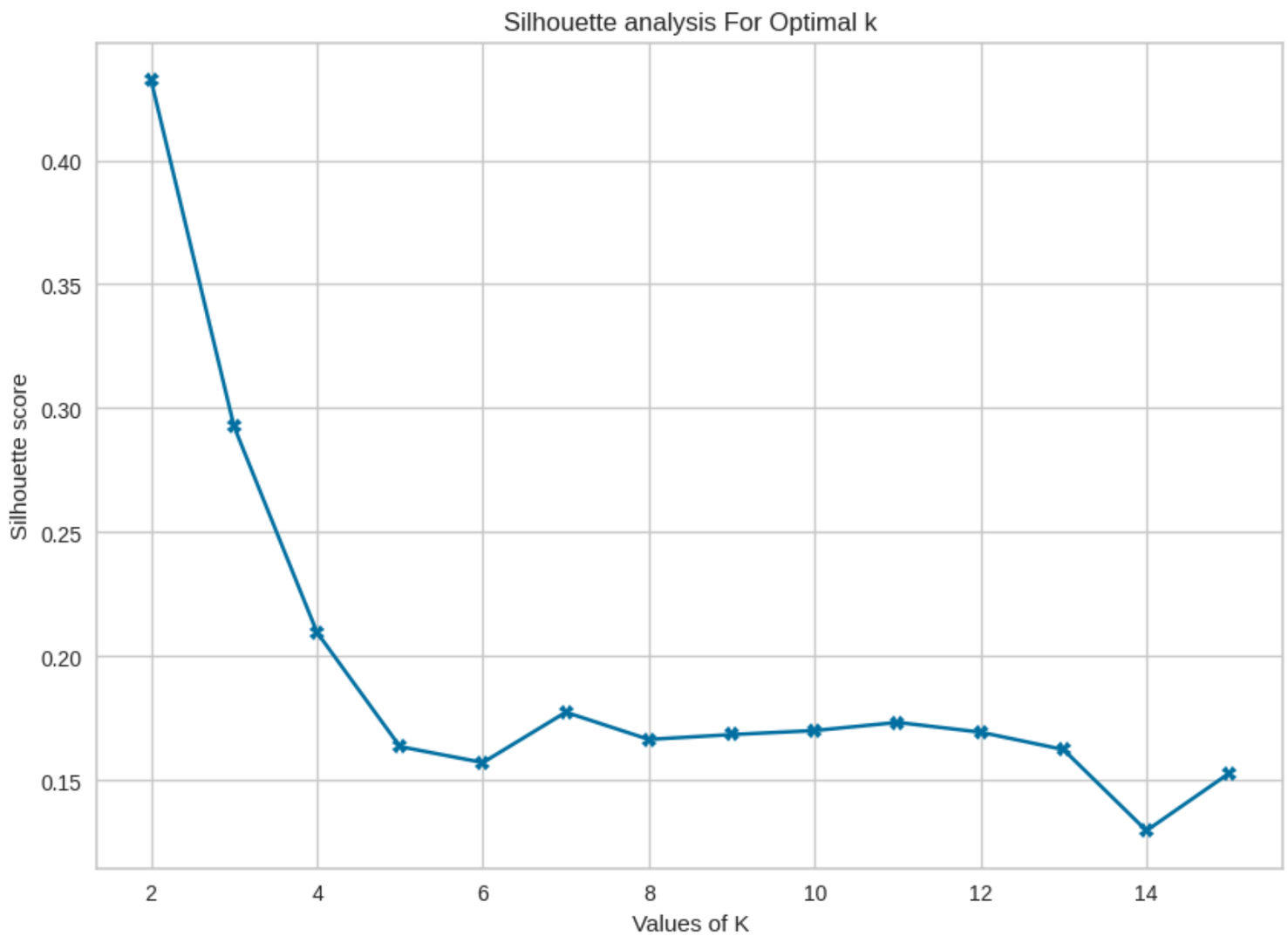


Silhouette Score for each k value

```
silhouette_avg = []
for i in range(2,16):
  # initialise kmeans
  kmeans = KMeans(n_clusters= i)
  cluster_labels = kmeans.fit_predict(features)
  # silhouette score
  silhouette_avg.append(silhouette_score(features,cluster_labels))
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning: The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of `
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning: The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of `
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning: The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of `
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning: The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of `
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning: The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of `
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning: The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of `
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning: The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of `
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning: The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of `
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning: The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of `
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning: The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of `
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning: The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of `
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning: The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of `
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning: The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of `
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning: The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of `
  warnings.warn(
```

```
plt.figure(figsize=(10,7))
plt.plot(range(2,16),silhouette_avg,'bX-')
plt.xlabel('Values of K')
plt.ylabel('Silhouette score')
plt.title('Silhouette analysis For Optimal k')
plt.show()
```



K-means Model

Taking k value as 3 as per Elbow Method

```
model = KMeans(n_clusters=3)
model = model.fit(features)
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning: The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of `
  warnings.warn(
```

```
y_km = model.predict(features)
centers = model.cluster_centers_
```

```
df['Cluster'] = pd.DataFrame(y_km)
df.to_csv("Cluster_data",index=False)
```

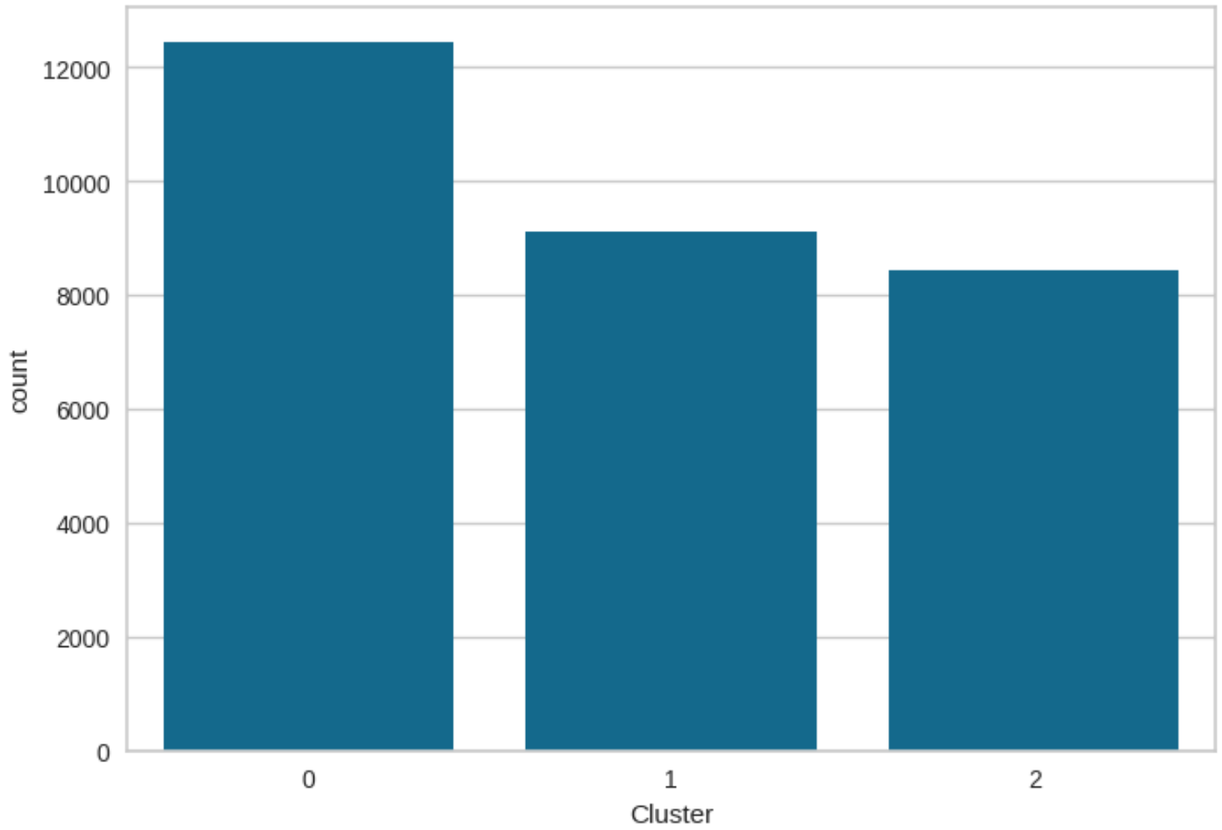
```
df['Cluster'].value_counts()
```

```
Cluster
0    12432
1     9128
2     8440
Name: count, dtype: int64
```

```
sns.countplot(data=df,x='Cluster')
plt.show
```

```
matplotlib.pyplot.show
def show(*args, **kwargs)
```

```
Display all open figures.

Parameters
----------
block : bool, optional
    Whether to wait for all figures to be closed before returning.
```



Analysing Clusters

```
c_df = pd.read_csv('Cluster_data')
c_df.head()
```

	Cust_ID	Gender	Orders	Jordan	Gatorade	Samsung	Asus	Udis	Mondelez International	Wrangler	...	Dior	Scabal	Tommy Hilfiger	Hollister	Forever 21	Colavita	Microsoft	Jiffy mix	Kraft
0	1	M	7	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0
1	2	F	0	0	1	0	0	0	0	0	...	1	0	0	0	0	0	0	0	0

```
c_df['Total Search']=c_df.iloc[:,3:38].sum(axis=1)
```

3	4	F	0	0	0	0	0	0	0	0		0	0	0	0	0	0	0	0	0

Analyzing Cluster 0

```
cl_0=c_df.groupby(['Cluster', 'Gender'], as_index=False).sum().query('Cluster == 0')
cl_0
```

	Cluster	Gender	Cust_ID	Orders	Jordan	Gatorade	Samsung	Asus	Udis	Mondelez International	...	Dior	Scabal	Tommy Hilfiger	Hollister	Forever 21	Colavita	Microsoft	Jiffy mix	Kraft
0	0	F	154699357	6269	2470	1947	2086	1813	1632	1480	...	3041	4194	1807	847	654	1960	1153	870	776
1	0	M	28245384	1291	601	777	435	12	75	162	...	283	175	172	83	55	386	186	160	142

2 rows × 40 columns

```
plt.figure(figsize=(15,6))
plt.subplot(1,2,1)
sns.countplot(data=c_df.query('Cluster == 0'),x= 'Gender')
plt.title('Customers count')

plt.subplot(1,2,2)
sns.barplot(data=cl_0,x='Gender',y='Total Search')
plt.title('Total Searches by Gender')
plt.suptitle('No. of Customers and their Total Searches in "Cluster 0"')
plt.show()
```