Ansh Awasthi IT A

2100290139002

Clustering wholesale customers data using K-Means

In [8]:
```
import numpy as np
import pandas as pd

import matplotlib.pyplot as plt
import seaborn as sns

from sklearn.cluster import KMeans
from sklearn.preprocessing import StandardScaler
```

In [3]:
```
data = pd.read_csv(r'C:\Users\anshlife1\Desktop\Wholesale.csv')
data.head()
```

Out[3]:

	Channel	Region	Fresh	Milk	Grocery	Frozen	Detergents_Paper	Delicassen
0	2	3	12669	9656	7561	214	2674	1338
1	2	3	7057	9810	9568	1762	3293	1776
2	2	3	6353	8808	7684	2405	3516	7844
3	1	3	13265	1196	4221	6404	507	1788
4	2	3	22615	5410	7198	3915	1777	5185

Exploratory Data Analysis (EDA)

Data Preprocessing

In [ ]:
```
scaler = StandardScaler()
scaled_df = scaler.fit_transform(data)

pd.DataFrame(scaled_df).describe()
```

In [ ]:
```
model = KMeans(n_clusters=3,
               init='k-means++',
               n_init=10,
               max_iter=300,
               tol=0.0001,
               precompute_distances='auto',
               verbose=0,
               random_state=42,
               copy_x=True,
               n_jobs=None,
               algorithm='auto')

model.fit(scaled_df)
model.inertia_
```

Find optimum value of K

In [ ]:
```
clusters = range(1, 20)
sse=[]
for cluster in clusters:
    model = KMeans(n_clusters=cluster,
                init='k-means++',
                n_init=10,
                max_iter=300,
                tol=0.0001,
                precompute_distances='auto',
                verbose=0,
                random_state=42,
                copy_x=True,
                n_jobs=None,
                algorithm='auto')

    model.fit(scaled_df)
    sse.append(model.inertia_)

sse_df = pd.DataFrame(np.column_stack((clusters, sse)), columns=['cluster', 'SSE'])
fig, ax = plt.subplots(figsize=(13, 5))
ax.plot(sse_df['cluster'], sse_df['SSE'], marker='o')
ax.set_xlabel('Number of clusters')
ax.set_ylabel('Inertia or SSE')
```

Build the final model

We will choose K=5 and fit the model.

In [ ]:
```
model = KMeans(n_clusters=5,
               init='k-means++',
               n_init=10,
               max_iter=300,
               tol=0.0001,
               precompute_distances='auto',
               verbose=0,
               random_state=42,
               copy_x=True,
               n_jobs=-1,
               algorithm='auto')

model.fit(scaled_df)
```

In [ ]:
```
print('SSE: ', model.inertia_)
print('\nCentroids: \n', model.cluster_centers_)

pred = model.predict(scaled_df)
data['cluster'] = pred
print('\nCount in each cluster: \n', data['cluster'].value_counts())
```

In [ ]:
```

```