# LAB # 06

## UNSUPERVISED LEARNING (K-MEANS CLUSTERINGALGORITHM) AND UNSUPERVISED LEARNING (APRIORI ALGORITHM)

## Lab Tasks:-

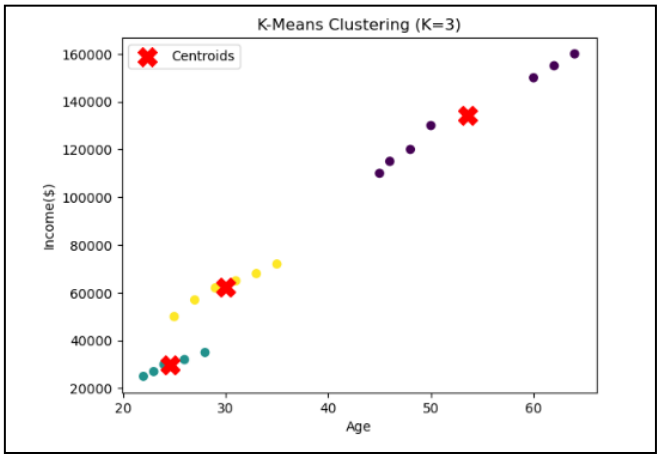
1. A dataset (income.csv) has been provided. Implement K-Means Clustering Algorithm on this dataset using K (number of clusters = 3). Also find out new centroid values based on the mean values of the coordinates of all the data instances from the corresponding cluster.

**Code:-**

```
import pandas as pd
from sklearn.cluster import KMeans
import matplotlib.pyplot as plt

df = pd.read_csv("Income.csv")

print("Columns:", df.columns.tolist())

kmeans = KMeans(n_clusters=3, random_state=42)
df["Cluster"] = kmeans.fit_predict(df[["Age", "Income($)"]])

centroids = kmeans.cluster_centers_
print("\nCentroids (Age, Income($)):")
print(centroids)

print("\nDataset with clusters:")
print(df)

plt.scatter(df["Age"], df["Income($)"], c=df["Cluster"], cmap='viridis')
plt.scatter(centroids[:, 0], centroids[:, 1],
            s=200, c='red', marker='X', label='Centroids')

plt.xlabel("Age")
plt.ylabel("Income($)")
plt.title("K-Means Clustering (K=3)")
plt.legend()
plt.show()
```

**Output:-**

```
Columns: ['Age', 'Income($)']

Centroids (Age, Income($)):
[[5.35714286e+01 1.34285714e+05]
 [2.46000000e+01 2.98000000e+04]
 [3.00000000e+01 6.23333333e+04]]

Dataset with clusters:
    Age  Income($)  Cluster
0    25      50000        2
1    27      57000        2
2    29      62000        2
3    31      65000        2
4    33      68000        2
5    35      72000        2
6    45     110000        0
7    46     115000        0
8    48     120000        0
9    50     130000        0
10   22      25000        1
11   23      27000        1
12   24      30000        1
13   26      32000        1
14   28      35000        1
15   60     150000        0
16   62     155000        0
17   64     160000        0
```



**2.** The following sample dataset contains 8 objects with their X, Y and Z coordinates. Your task is to cluster these objects into two clusters using K-Means Clustering Algorithm (hereyou define the value of K (of K-Means) in essence to be 2).

| Objects | X | Y | Z |
|---|---|---|---|
| OB-1 | 1 | 4 | 1 |
| OB-2 | 1 | 2 | 2 |
| OB-3 | 1 | 4 | 2 |
| OB-4 | 2 | 1 | 2 |
| OB-5 | 1 | 1 | 1 |
| OB-6 | 2 | 4 | 2 |
| OB-7 | 1 | 1 | 2 |
| OB-8 | 2 | 1 | 1 |

**Code:-**

```
import pandas as pd
from sklearn.cluster import KMeans

data = {
    "Object": ["OB-1","OB-2","OB-3","OB-4","OB-5","OB-6","OB-7","OB-8"],
    "X": [1,1,1,2,1,2,1,2],
    "Y": [4,2,4,1,1,4,1,1],
    "Z": [1,2,2,2,1,2,2,1]
}

df = pd.DataFrame(data)

kmeans = KMeans(n_clusters=2, random_state=42)
df["Cluster"] = kmeans.fit_predict(df[["X","Y","Z"]])

centroids = kmeans.cluster_centers_

print("Cluster Assignments:")
print(df)

print("\nCentroids:")
print(centroids)
```

**Output:-**

```
Cluster Assignments:
  Object  X  Y  Z  Cluster
0   OB-1  1  4  1        0
1   OB-2  1  2  2        1
2   OB-3  1  4  2        0
3   OB-4  2  1  2        1
4   OB-5  1  1  1        1
5   OB-6  2  4  2        0
6   OB-7  1  1  2        1
7   OB-8  2  1  1        1

Centroids:
[[1.33333333 4.         1.66666667]
 [1.4        1.2        1.6       ]]
```

**3.** Run the given code of Apriori Algorithm and show the output.

**Code:-**

```
import pandas as pd
from mlxtend.frequent_patterns import apriori
from mlxtend.frequent_patterns import association_rules
df = pd.read_excel('/content/sample_data/Book1.xlsx')
df.head()
df['Description'] = df['Description'].str.strip()
df.dropna(axis=0, subset=['InvoiceNo'], inplace=True)
df['InvoiceNo'] = df['InvoiceNo'].astype('str')
df = df[~df['InvoiceNo'].str.contains('C')]
df
basket = (df[df['Country'] =="United Kingdom"]
.groupby(['InvoiceNo', 'Description'])['Quantity']
.sum().unstack().reset_index().fillna(0)
.set_index('InvoiceNo'))
basket
def encode_units(x):
  if x <= 0:
    return 0
  if x >= 1:
    return 1
basket_sets = basket.applymap(encode_units)
basket_sets.drop('ASSORTED COLOUR BIRD ORNAMENT', inplace=True, axis=1)
basket_sets
frequent_itemsets = apriori(basket_sets, min_support=0.07, use_colnames=True)
rules = association_rules(frequent_itemsets, metric="lift", min_threshold=1)
rules.head()
rules[ (rules['lift'] >= 3) &
(rules['confidence'] >= 0.8) ]
```

**Output:-**

| | antecedents | consequents | antecedent support | consequent support | support | confidence | lift |
|---|---|---|---|---|---|---|---|
| 0 | (CREAM CUPID HEARTS COAT HANGER) | (GLASS STAR FROSTED T-LIGHT HOLDER) | 0.333333 | 0.333333 | 0.333333 | 1.0 | 3.0 |
| 1 | (GLASS STAR FROSTED T-LIGHT HOLDER) | (CREAM CUPID HEARTS COAT HANGER) | 0.333333 | 0.333333 | 0.333333 | 1.0 | 3.0 |
| 2 | (KNITTED UNION FLAG HOT WATER BOTTLE) | (CREAM CUPID HEARTS COAT HANGER) | 0.333333 | 0.333333 | 0.333333 | 1.0 | 3.0 |
| 3 | (CREAM CUPID HEARTS COAT HANGER) | (KNITTED UNION FLAG HOT WATER BOTTLE) | 0.333333 | 0.333333 | 0.333333 | 1.0 | 3.0 |
| 4 | (CREAM CUPID HEARTS COAT HANGER) | (RED WOOLLY HOTTIE WHITE HEART) | 0.333333 | 0.333333 | 0.333333 | 1.0 | 3.0 |
| ... | ... | ... | ... | ... | ... | ... | ... |
| 1979 | (KNITTED UNION FLAG HOT WATER BOTTLE) | (GLASS STAR FROSTED T-LIGHT HOLDER, RED WOOLLY... | 0.333333 | 0.333333 | 0.333333 | 1.0 | 3.0 |
| 1980 | (WHITE METAL LANTERN) | (GLASS STAR FROSTED T-LIGHT HOLDER, RED WOOLLY... | 0.333333 | 0.333333 | 0.333333 | 1.0 | 3.0 |
| 1981 | (SET 7 BABUSHKA NESTING BOXES) | (GLASS STAR FROSTED T-LIGHT HOLDER, RED WOOLLY... | 0.333333 | 0.333333 | 0.333333 | 1.0 | 3.0 |
| 1982 | (CREAM CUPID HEARTS COAT HANGER) | (GLASS STAR FROSTED T-LIGHT HOLDER, RED WOOLLY... | 0.333333 | 0.333333 | 0.333333 | 1.0 | 3.0 |
| 1983 | (WHITE HANGING HEART T-LIGHT HOLDER) | (GLASS STAR FROSTED T-LIGHT HOLDER, RED WOOLLY... | 0.333333 | 0.333333 | 0.333333 | 1.0 | 3.0 |

**4.** In given code there is a support value of at least 7%, Generate frequent item sets that haveSupport value of at least 5%.

## Code:-

```
import pandas as pd
from mlxtend.frequent_patterns import apriori
from mlxtend.frequent_patterns import association_rules
df = pd.read_excel('/content/sample_data/Book1.xlsx')
df.head()
df['Description'] = df['Description'].str.strip()
df.dropna(axis=0, subset=['InvoiceNo'], inplace=True)
df['InvoiceNo'] = df['InvoiceNo'].astype('str')
df = df[~df['InvoiceNo'].str.contains('C')]
df
basket = (df[df['Country'] =="United Kingdom"]
.groupby(['InvoiceNo', 'Description'])['Quantity']
.sum().unstack().reset_index().fillna(0)
.set_index('InvoiceNo'))
basket
def encode_units(x):
  if x <= 0:
    return 0
  if x >= 1:
    return 1
basket_sets = basket.applymap(encode_units)
basket_sets.drop('ASSORTED COLOUR BIRD ORNAMENT', inplace=True, axis=1)
basket_sets
frequent_itemsets = apriori(basket_sets, min_support=0.05, use_colnames=True)
rules = association_rules(frequent_itemsets, metric="lift", min_threshold=1)
rules.head()
rules[ (rules['lift'] >= 2) &
(rules['confidence'] >= 0.6) ]
```

## Output:-

| | antecedents | consequents | antecedent support | consequent support | support | confidence | lift |
|---|---|---|---|---|---|---|---|
| 0 | (CREAM CUPID HEARTS COAT HANGER) | (GLASS STAR FROSTED T-LIGHT HOLDER) | 0.333333 | 0.333333 | 0.333333 | 1.0 | 3.0 |
| 1 | (GLASS STAR FROSTED T-LIGHT HOLDER) | (CREAM CUPID HEARTS COAT HANGER) | 0.333333 | 0.333333 | 0.333333 | 1.0 | 3.0 |
| 2 | (KNITTED UNION FLAG HOT WATER BOTTLE) | (CREAM CUPID HEARTS COAT HANGER) | 0.333333 | 0.333333 | 0.333333 | 1.0 | 3.0 |
| 3 | (CREAM CUPID HEARTS COAT HANGER) | (KNITTED UNION FLAG HOT WATER BOTTLE) | 0.333333 | 0.333333 | 0.333333 | 1.0 | 3.0 |
| 4 | (CREAM CUPID HEARTS COAT HANGER) | (RED WOOLLY HOTTIE WHITE HEART) | 0.333333 | 0.333333 | 0.333333 | 1.0 | 3.0 |
| ... | ... | ... | ... | ... | ... | ... | ... |
| 1979 | (KNITTED UNION FLAG HOT WATER BOTTLE) | (GLASS STAR FROSTED T-LIGHT HOLDER, RED WOOLLY... | 0.333333 | 0.333333 | 0.333333 | 1.0 | 3.0 |
| 1980 | (WHITE METAL LANTERN) | (GLASS STAR FROSTED T-LIGHT HOLDER, RED WOOLLY... | 0.333333 | 0.333333 | 0.333333 | 1.0 | 3.0 |
| 1981 | (SET 7 BABUSHKA NESTING BOXES) | (GLASS STAR FROSTED T-LIGHT HOLDER, RED WOOLLY... | 0.333333 | 0.333333 | 0.333333 | 1.0 | 3.0 |
| 1982 | (CREAM CUPID HEARTS COAT HANGER) | (GLASS STAR FROSTED T-LIGHT HOLDER, RED WOOLLY... | 0.333333 | 0.333333 | 0.333333 | 1.0 | 3.0 |
| 1983 | (WHITE HANGING HEART T-LIGHT HOLDER) | (GLASS STAR FROSTED T-LIGHT HOLDER, RED WOOLLY... | 0.333333 | 0.333333 | 0.333333 | 1.0 | 3.0 |