

In [103]:

```
# Import Libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as pyplot

# Import dataset
churn_clean = pd.read_csv("C:/Users/hkeim/OneDrive/Documents/School/D212/churn_clean.csv")
churn_clean.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10000 entries, 0 to 9999
Data columns (total 50 columns):
#   Column                                Non-Null Count  Dtype
---  -
0   CaseOrder                            10000 non-null  int64
1   Customer_id                          10000 non-null  object
2   Interaction                          10000 non-null  object
3   UID                                  10000 non-null  object
4   City                                 10000 non-null  object
5   State                                10000 non-null  object
6   County                              10000 non-null  object
7   Zip                                  10000 non-null  int64
8   Lat                                  10000 non-null  float64
9   Lng                                  10000 non-null  float64
10  Population                           10000 non-null  int64
11  Area                                 10000 non-null  object
12  TimeZone                            10000 non-null  object
13  Job                                  10000 non-null  object
14  Children                            10000 non-null  int64
15  Age                                  10000 non-null  int64
16  Income                              10000 non-null  float64
17  Marital                             10000 non-null  object
18  Gender                              10000 non-null  object
19  Churn                               10000 non-null  object
20  Outage_sec_perweek                  10000 non-null  float64
21  Email                               10000 non-null  int64
22  Contacts                            10000 non-null  int64
23  Yearly_equip_failure                10000 non-null  int64
24  Techie                              10000 non-null  object
25  Contract                            10000 non-null  object
26  Port_modem                          10000 non-null  object
27  Tablet                              10000 non-null  object
28  InternetService                    10000 non-null  object
29  Phone                               10000 non-null  object
30  Multiple                            10000 non-null  object
31  OnlineSecurity                     10000 non-null  object
32  OnlineBackup                       10000 non-null  object
33  DeviceProtection                   10000 non-null  object
34  TechSupport                        10000 non-null  object
35  StreamingTV                        10000 non-null  object
36  StreamingMovies                    10000 non-null  object
37  PaperlessBilling                   10000 non-null  object
38  PaymentMethod                      10000 non-null  object
39  Tenure                             10000 non-null  float64
40  MonthlyCharge                      10000 non-null  float64
41  Bandwidth_GB_Year                  10000 non-null  float64
42  Item1                              10000 non-null  int64
43  Item2                              10000 non-null  int64
44  Item3                              10000 non-null  int64
45  Item4                              10000 non-null  int64
46  Item5                              10000 non-null  int64
47  Item6                              10000 non-null  int64
48  Item7                              10000 non-null  int64
49  Item8                              10000 non-null  int64
```

dtypes: float64(7), int64(16), object(27)  
memory usage: 3.8+ MB

```
In [104]: # Select dimensions
dimensions = churn_clean.filter(items = ['Churn', 'Tenure', 'MonthlyCharge'])

# Convert Churn to category codes
dimensions['Churn'] = dimensions['Churn'].astype('category')
dimensions['Churn'] = dimensions['Churn'].cat.codes
```

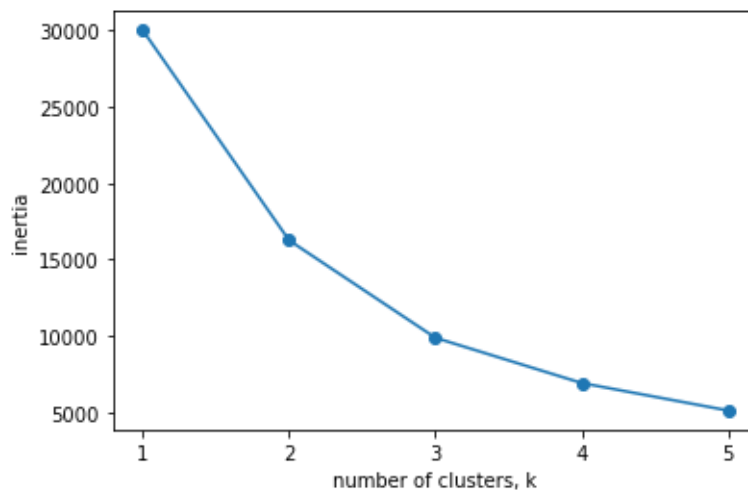
```
In [105]: # Import StandardScaler
from sklearn.preprocessing import StandardScaler

# Scale data
scaler = StandardScaler()
scaler.fit(dimensions)
dim_scaled=scaler.transform(dimensions)
```

```
In [106]: # Import KMeans
from sklearn.cluster import KMeans

# Create elbow graph to determine number of clusters
ks = range(1, 6)
inertias = []

for k in ks:
    model=KMeans(n_clusters=k)
    model.fit(dim_scaled)
    inertias.append(model.inertia_)
plt.plot(ks, inertias, '-o')
plt.xlabel('number of clusters, k')
plt.ylabel('inertia')
plt.xticks(ks)
plt.show()
```



```
In [107]: # Run KMeans with 4 clusters
kmeans = KMeans(n_clusters=4)
kmeans.fit(dim_scaled)

# Assign the columns of dim_scaled: xs and ys
xs = dim_scaled[:,1]
```

```

ys = dim_scaled[:,2]

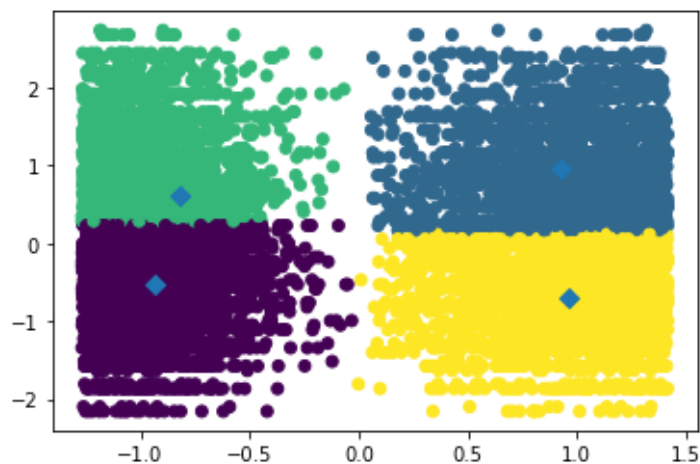
# Make a scatter plot of xs and ys, using labels to define the colors
plt.scatter(xs, ys, c=labels)

# Assign the cluster centers: centroids
centroids = kmeans.cluster_centers_

# Assign the columns of centroids: centroids_x, centroids_y
centroids_x = centroids[:,1]
centroids_y = centroids[:,2]

# Make a scatter plot of centroids_x and centroids_y
plt.scatter(centroids_x, centroids_y, marker='D', s=50)
plt.show()

```



```

In [108]: # Create crosstab
df = pd.DataFrame({'labels': labels, 'Churn': Churn})
ct = pd.crosstab(df['labels'], df['Churn'])
print(ct)

```

Churn	No	Yes
labels		
0	2214	965
1	1760	263
2	417	1402
3	2959	20

```

In [110]: # Export dim_scaled for report purposes
dimensions.to_csv("C:/Users/hkeim/OneDrive/Documents/School/D212/Keim D212 Task One Cleaned D

```

```

In [ ]:

```