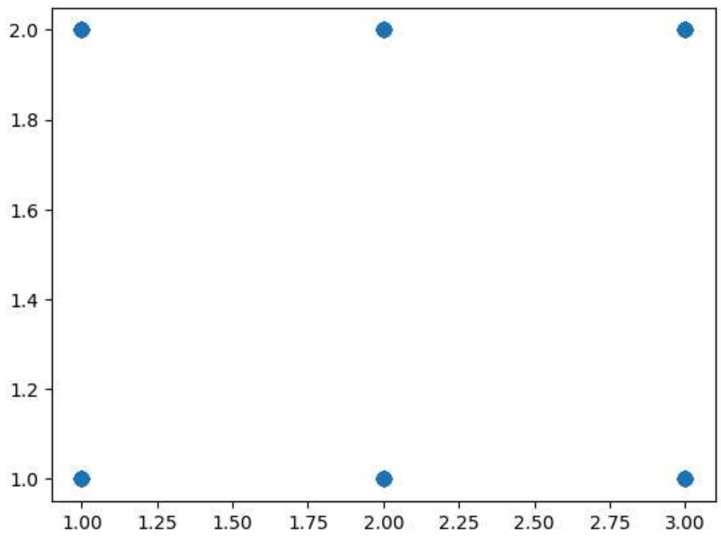
K-Means Clustering

```
# Import necessary libraries
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.cluster import KMeans
from sklearn.datasets import make_blobs
```

```
df = pd.read_csv("/content/titanic_cleaned.csv")
```

```
X = df.drop('Survived',axis=1)
y = df['Survived']
```

```
plt.scatter(X.iloc[:, 0], X.iloc[:, 1], s=50)
plt.show()
```



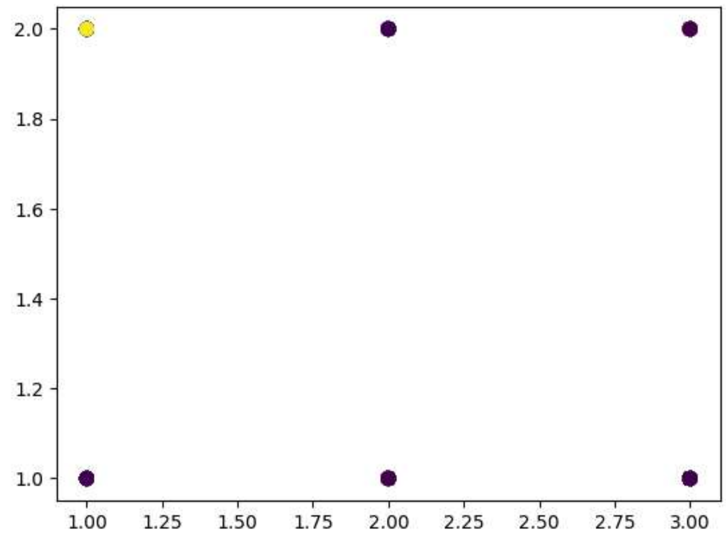
```
# Apply K-Means clustering
kmeans = KMeans(n_clusters=3)
kmeans.fit(X)
```

```
▾ KMeans ⓘ ?
KMeans(n_clusters=3)
```

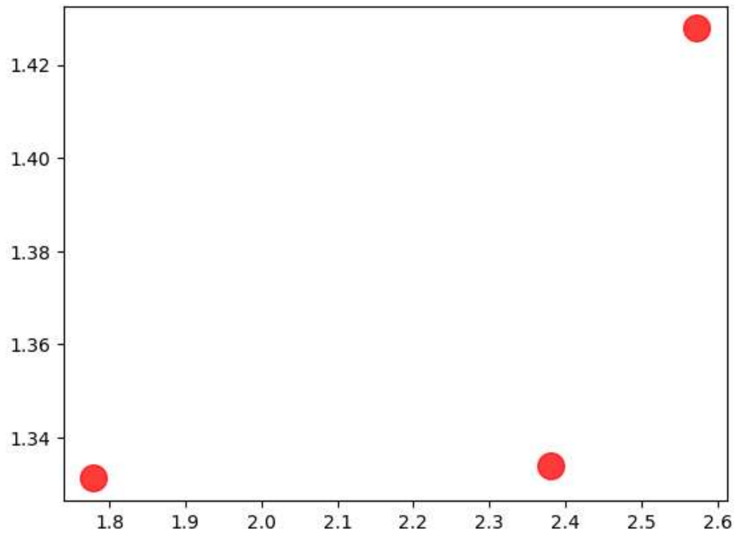
```
# Get the cluster centers and labels
centroids = kmeans.cluster_centers_
labels = kmeans.labels_
```

```
# Visualize the clusters
plt.scatter(X.iloc[:, 0], X.iloc[:, 1], c=labels, s=50, cmap='viridis')
```

```
<matplotlib.collections.PathCollection at 0x7cafaeb2e8f0>
```

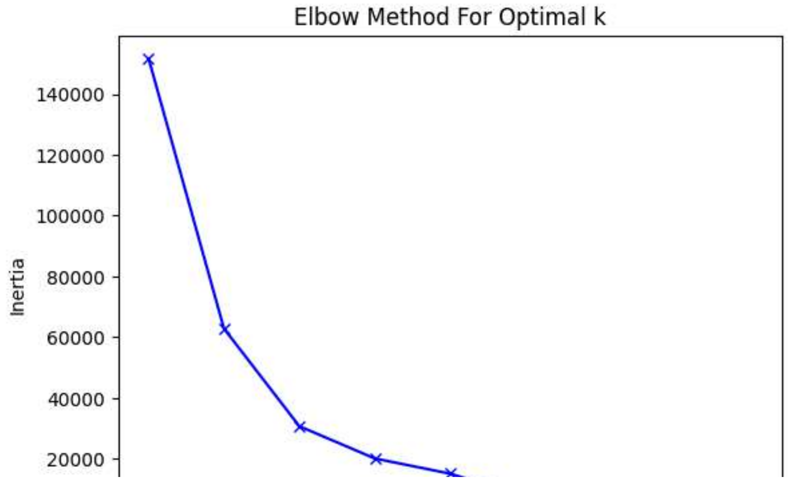


```
plt.scatter(centroids[:, 0], centroids[:, 1], c='red', s=200, alpha=0.75)
plt.show()
```



```
# Elbow method to determine the optimal number of clusters
inertia = []
K = range(1, 10)
for k in K:
    kmeans = KMeans(n_clusters=k)
    kmeans.fit(X)
    inertia.append(kmeans.inertia_)


# Plot the elbow curve
plt.plot(K, inertia, 'bx-')
plt.xlabel('Number of clusters')
plt.ylabel('Inertia')
plt.title('Elbow Method For Optimal k')
plt.show()
```


Elbow Method For Optimal k
Inertia
140000
120000
100000
80000
60000
40000
20000