Name: Raghav Sharma

Roll No: 25201313

In [1]:
```
import numpy as np
import matplotlib.pyplot as plt
from sklearn.datasets import make_blobs, load_iris
from sklearn.cluster import KMeans
from sklearn.metrics import silhouette_score
```

In [2]:
```
X, y_true = make_blobs(n_samples=500, centers=4, cluster_std=0.6, random_state=4
```

In [3]:
```
Ks = range(2,9)
inertias = []
sil_scores = []
```

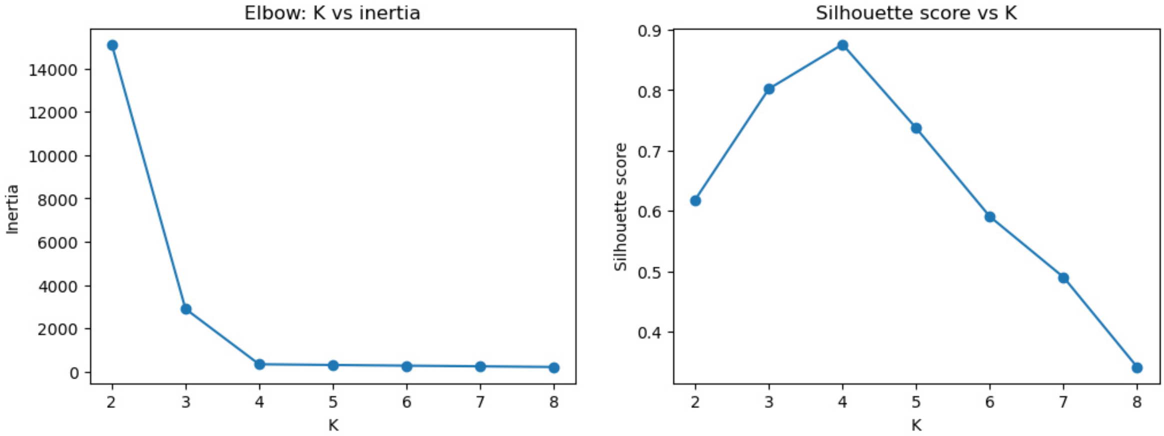
In [4]:
```
for k in Ks:
    km = KMeans(n_clusters=k, n_init=10, random_state=42)
    labels = km.fit_predict(X)
    inertias.append(km.inertia_)
    sil_scores.append(silhouette_score(X, labels))
```

```
D:\Users\RaghavSharma\anaconda3\Lib\site-packages\sklearn\cluster\_kmeans.py:141
9: UserWarning: KMeans is known to have a memory leak on Windows with MKL, when t
here are less chunks than available threads. You can avoid it by setting the envi
ronment variable OMP_NUM_THREADS=2.
  warnings.warn(
D:\Users\RaghavSharma\anaconda3\Lib\site-packages\sklearn\cluster\_kmeans.py:141
9: UserWarning: KMeans is known to have a memory leak on Windows with MKL, when t
here are less chunks than available threads. You can avoid it by setting the envi
ronment variable OMP_NUM_THREADS=2.
  warnings.warn(
D:\Users\RaghavSharma\anaconda3\Lib\site-packages\sklearn\cluster\_kmeans.py:141
9: UserWarning: KMeans is known to have a memory leak on Windows with MKL, when t
here are less chunks than available threads. You can avoid it by setting the envi
ronment variable OMP_NUM_THREADS=2.
  warnings.warn(
D:\Users\RaghavSharma\anaconda3\Lib\site-packages\sklearn\cluster\_kmeans.py:141
9: UserWarning: KMeans is known to have a memory leak on Windows with MKL, when t
here are less chunks than available threads. You can avoid it by setting the envi
ronment variable OMP_NUM_THREADS=2.
  warnings.warn(
D:\Users\RaghavSharma\anaconda3\Lib\site-packages\sklearn\cluster\_kmeans.py:141
9: UserWarning: KMeans is known to have a memory leak on Windows with MKL, when t
here are less chunks than available threads. You can avoid it by setting the envi
ronment variable OMP_NUM_THREADS=2.
  warnings.warn(
D:\Users\RaghavSharma\anaconda3\Lib\site-packages\sklearn\cluster\_kmeans.py:141
9: UserWarning: KMeans is known to have a memory leak on Windows with MKL, when t
here are less chunks than available threads. You can avoid it by setting the envi
ronment variable OMP_NUM_THREADS=2.
  warnings.warn(
D:\Users\RaghavSharma\anaconda3\Lib\site-packages\sklearn\cluster\_kmeans.py:141
9: UserWarning: KMeans is known to have a memory leak on Windows with MKL, when t
here are less chunks than available threads. You can avoid it by setting the envi
ronment variable OMP_NUM_THREADS=2.
  warnings.warn(
```

In [6]:

```
fig, ax = plt.subplots(1,2, figsize=(12,4))
ax[0].plot(Ks, inertias, marker='o')
ax[0].set_title('Elbow: K vs inertia')
ax[0].set_xlabel('K'); ax[0].set_ylabel('Inertia')

ax[1].plot(Ks, sil_scores, marker='o')
ax[1].set_title('Silhouette score vs K')
ax[1].set_xlabel('K'); ax[1].set_ylabel('Silhouette score')
plt.show()
```

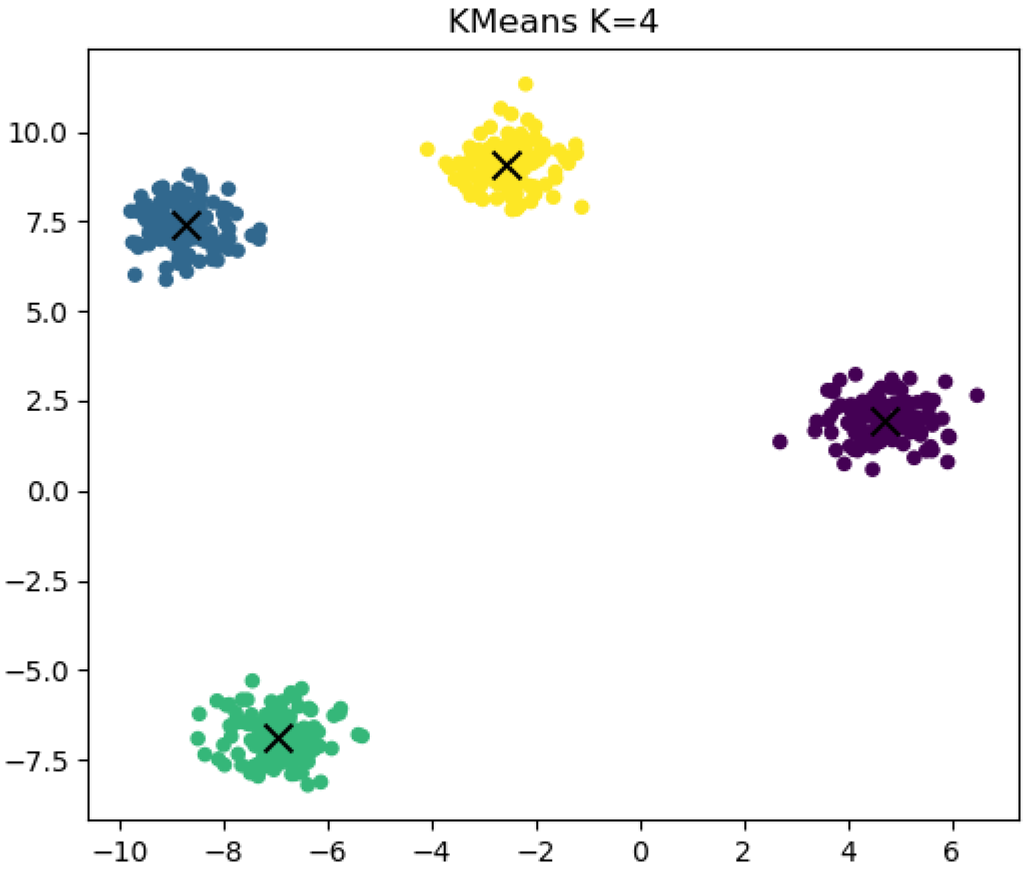


In [8]:

```
best_k = 4
km = KMeans(n_clusters=best_k, n_init=20, init='k-means++', random_state=42)
labels = km.fit_predict(X)

plt.figure(figsize=(6,5))
plt.scatter(X[:,0], X[:,1], c=labels, s=20)
plt.scatter(km.cluster_centers_[:,0], km.cluster_centers_[:,1], c='black', s=100
plt.title(f'KMeans K={best_k}')
plt.show()
```

```
D:\Users\RaghavSharma\anaconda3\Lib\site-packages\sklearn\cluster\_kmeans.py:141
9: UserWarning: KMeans is known to have a memory leak on Windows with MKL, when t
here are less chunks than available threads. You can avoid it by setting the envi
ronment variable OMP_NUM_THREADS=2.
  warnings.warn(
```



In [ ]: