K-Mean Clustering

In [ ]:

```
from sklearn.cluster import KMeans
```

In [2]:

```
import pandas as pd
```

In [3]:

```
import numpy as np
```

In [4]:

```
import matplotlib.pyplot as plt
```

In [5]:

```
np.random.seed(2)
X =np.random.standard_normal((100,2))
X[:50,0] = X[:50,0]+3
X[:50,1] = X[:50,1]-4
km1 = KMeans(n_clusters=2, n_init=20)
km1.fit(X)
np.random.seed(4)
km2 =  KMeans(n_clusters=3, n_init=20)
km2.fit(X)
```

```
C:\ProgramData\anaconda3\lib\site-packages\sklearn\cluster\_kmeans.py:138
2: UserWarning: KMeans is known to have a memory leak on Windows with MKL,
when there are less chunks than available threads. You can avoid it by set
ting the environment variable OMP_NUM_THREADS=1.
  warnings.warn(
C:\ProgramData\anaconda3\lib\site-packages\sklearn\cluster\_kmeans.py:138
2: UserWarning: KMeans is known to have a memory leak on Windows with MKL,
when there are less chunks than available threads. You can avoid it by set
ting the environment variable OMP_NUM_THREADS=1.
  warnings.warn(
```

Out[5]:

```
KMeans(n_clusters=3, n_init=20)
```

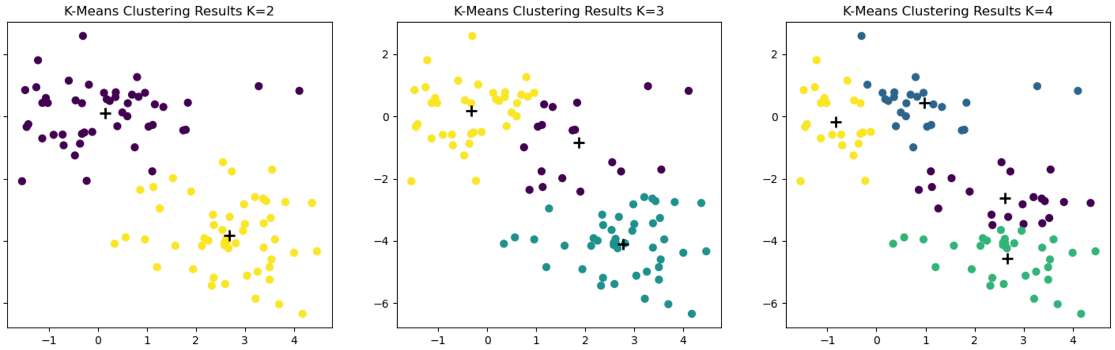
In a Jupyter environment, please rerun this cell to show the HTML representation or trust the notebook.
On GitHub, the HTML representation is unable to render, please try loading this page with nbviewer.org.

In [6]:

```
np.random.seed(6)
km3 = KMeans(n_clusters=4, n_init=20)
km3.fit(X)
fig, (ax1, ax2, ax3) = plt.subplots(1,3, figsize=(18,5))
ax1.scatter(X[:,0], X[:,1], s=40, c=km1.labels_)
ax1.set_title('K-Means Clustering Results K=2')
ax1.scatter(km1.cluster_centers_[:,0], km1.cluster_centers_[:,1], marke
ax2.scatter(X[:,0], X[:,1], s=40, c=km2.labels_)
ax2.set_title('K-Means Clustering Results K=3')
ax2.scatter(km2.cluster_centers_[:,0], km2.cluster_centers_[:,1], marke
ax3.scatter(X[:,0], X[:,1], s=40, c=km3.labels_)
ax3.set_title('K-Means Clustering Results K=4')
ax3.scatter(km3.cluster_centers_[:,0], km3.cluster_centers_[:,1], marke

```

```
C:\ProgramData\anaconda3\lib\site-packages\sklearn\cluster\_kmeans.py:138
2: UserWarning: KMeans is known to have a memory leak on Windows with MKL,
when there are less chunks than available threads. You can avoid it by set
ting the environment variable OMP_NUM_THREADS=1.
  warnings.warn(
```



In [ ]:

```

```