In [4]:

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

In [5]:

```
df=pd.read_csv("f:/dataset/fruits_cluster.csv")
X=df.iloc[:,:].values
```

In [6]:

```
model=KMeans(n_clusters=3,max_iter=50,n_init=5)
model.fit(X)
print(model.labels_)
```

```
C:\Users\Ducat\anaconda3\Lib\site-packages\sklearn\cluster\_kmeans.py:1436: UserWarning:
KMeans is known to have a memory leak on Windows with MKL, when there are less chunks th
an available threads. You can avoid it by setting the environment variable OMP_NUM_THREA
DS=1.
  warnings.warn(
[0 1 2 2 1 0 1 0 1 1 0 1 0 0 2 2 0 0 2 2 1 1]
```

In [7]:

```
model=KMeans(n_clusters=2,max_iter=50,n_init=5)
model.fit(X)
print(model.labels_)
```

```
C:\Users\Ducat\anaconda3\Lib\site-packages\sklearn\cluster\_kmeans.py:1436: UserWarning:
KMeans is known to have a memory leak on Windows with MKL, when there are less chunks th
an available threads. You can avoid it by setting the environment variable OMP_NUM_THREA
DS=1.
  warnings.warn(
[0 1 1 1 0 0 1 0 1 0 0 0 0 0 1 1 0 0 1 1 0 0]
```

In [ ]:

```
#how to select number of clusters in KMeans
>elbow technique can be used in select num of clusters
>it is a graphical technique where we plot line chart b/w num_of_clustrs & wsse scores
```

In [9]:

```
import warnings
warnings.filterwarnings('ignore')
```

In [17]:

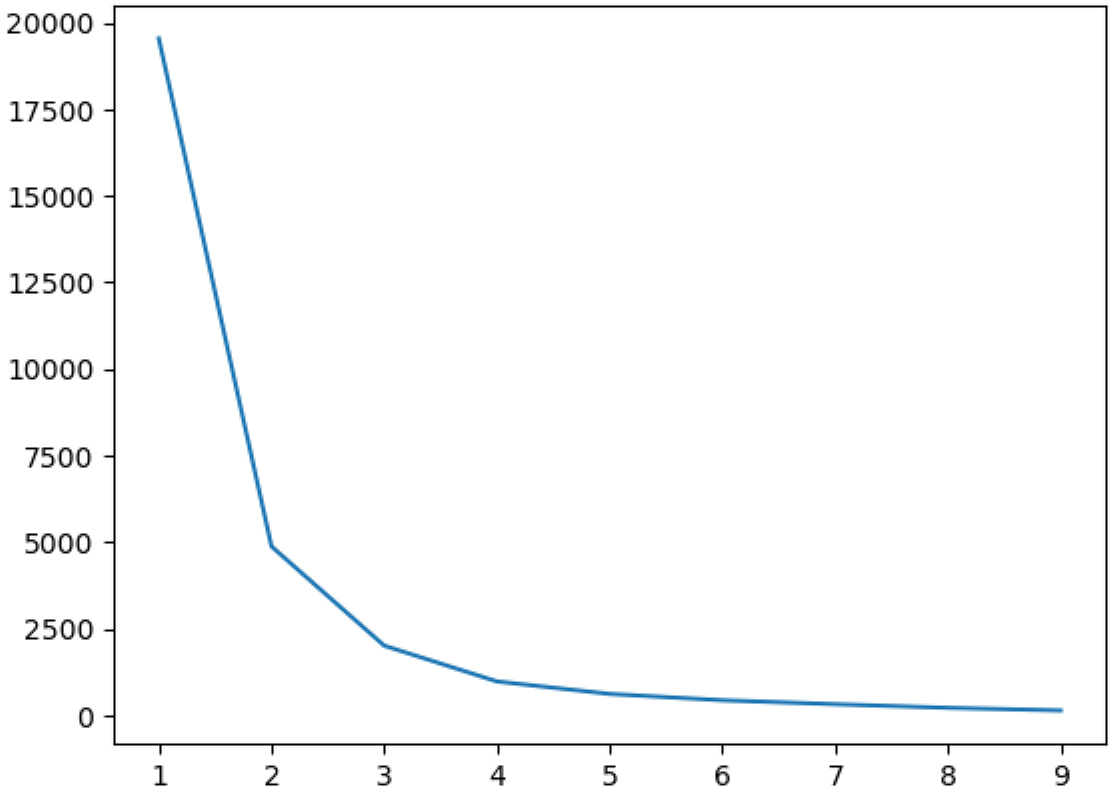
```
from sklearn.metrics import silhouette_score,silhouette_samples
```

In [10]:

```
wsse=[]
for i in range(1,10):
    model=KMeans(n_clusters=i,max_iter=50,n_init=5)
    model.fit(X)
    wsse.append(model.inertia_)
```

In [12]:

```
import matplotlib.pyplot as plt
plt.plot(range(1,10),wsse)
plt.show()
```



```
In [14]: model=KMeans(n_clusters=2,max_iter=50,n_init=5)
         model.fit(X)
         print(silhouette_score(X,model.labels_))

         0.6047687920287002
```

```
In [15]: model=KMeans(n_clusters=3,max_iter=50,n_init=5)
         model.fit(X)
         print(silhouette_score(X,model.labels_))

         0.5562767664418119
```

```
In [16]: model=KMeans(n_clusters=4,max_iter=50,n_init=5)
         model.fit(X)
         print(silhouette_score(X,model.labels_))

         0.5952186571525124
```

```
In [18]: model=KMeans(n_clusters=2,max_iter=50,n_init=5)
         model.fit(X)
         print(silhouette_samples(X,model.labels_))

         [0.756896   0.08910924 0.7106429  0.73673795 0.37057472 0.68911747
          0.48795214 0.74580381 0.09269296 0.32304302 0.68826806 0.54228985
          0.73426441 0.72721378 0.74553973 0.67113401 0.75763073 0.6883263
          0.66879425 0.74543725 0.66687718 0.66656766]
```

```
In [20]: model.labels_

Out[20]: array([0, 1, 1, 1, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 1, 1, 0, 0, 1, 1, 0, 0])
```

```
In [ ]:
```