In [1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn import datasets
```

In [2]:
```
iris = datasets.load_iris()
data_iris =pd.DataFrame(iris.data,columns=iris.feature_names)
data_iris.head()
```

Out[2]:

	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)
0	5.1	3.5	1.4	0.2
1	4.9	3.0	1.4	0.2
2	4.7	3.2	1.3	0.2
3	4.6	3.1	1.5	0.2
4	5.0	3.6	1.4	0.2

In [3]:
```
data_iris.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 150 entries, 0 to 149
Data columns (total 4 columns):
sepal length (cm)    150 non-null float64
sepal width (cm)     150 non-null float64
petal length (cm)    150 non-null float64
petal width (cm)     150 non-null float64
dtypes: float64(4)
memory usage: 4.8 KB
```

In [4]:
```
data_iris.describe()
```

Out[4]:

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)
count	150.000000	150.000000	150.000000	150.000000
mean	5.843333	3.057333	3.758000	1.199333
std	0.828066	0.435866	1.765298	0.762238
min	4.300000	2.000000	1.000000	0.100000
25%	5.100000	2.800000	1.600000	0.300000
50%	5.800000	3.000000	4.350000	1.300000
75%	6.400000	3.300000	5.100000	1.800000
max	7.900000	4.400000	6.900000	2.500000

In [5]:
```
sum(data_iris.duplicated())
```

Out[5]: 1

In [6]:
```
data_iris.drop_duplicates()
```

Out[6]:

	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)
0	5.1	3.5	1.4	0.2
1	4.9	3.0	1.4	0.2
2	4.7	3.2	1.3	0.2
3	4.6	3.1	1.5	0.2
4	5.0	3.6	1.4	0.2
5	5.4	3.9	1.7	0.4
6	4.6	3.4	1.4	0.3
7	5.0	3.4	1.5	0.2
8	4.4	2.9	1.4	0.2
9	4.9	3.1	1.5	0.1

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)
10	5.4	3.7	1.5	0.2
11	4.8	3.4	1.6	0.2
12	4.8	3.0	1.4	0.1
13	4.3	3.0	1.1	0.1
14	5.8	4.0	1.2	0.2
15	5.7	4.4	1.5	0.4
16	5.4	3.9	1.3	0.4
17	5.1	3.5	1.4	0.3
18	5.7	3.8	1.7	0.3
19	5.1	3.8	1.5	0.3
20	5.4	3.4	1.7	0.2
21	5.1	3.7	1.5	0.4
22	4.6	3.6	1.0	0.2
23	5.1	3.3	1.7	0.5
24	4.8	3.4	1.9	0.2
25	5.0	3.0	1.6	0.2
26	5.0	3.4	1.6	0.4
27	5.2	3.5	1.5	0.2
28	5.2	3.4	1.4	0.2
29	4.7	3.2	1.6	0.2
...	...	...	...	...
119	6.0	2.2	5.0	1.5
120	6.9	3.2	5.7	2.3
121	5.6	2.8	4.9	2.0
122	7.7	2.8	6.7	2.0

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)
123	6.3	2.7	4.9	1.8
124	6.7	3.3	5.7	2.1
125	7.2	3.2	6.0	1.8
126	6.2	2.8	4.8	1.8
127	6.1	3.0	4.9	1.8
128	6.4	2.8	5.6	2.1
129	7.2	3.0	5.8	1.6
130	7.4	2.8	6.1	1.9
131	7.9	3.8	6.4	2.0
132	6.4	2.8	5.6	2.2
133	6.3	2.8	5.1	1.5
134	6.1	2.6	5.6	1.4
135	7.7	3.0	6.1	2.3
136	6.3	3.4	5.6	2.4
137	6.4	3.1	5.5	1.8
138	6.0	3.0	4.8	1.8
139	6.9	3.1	5.4	2.1
140	6.7	3.1	5.6	2.4
141	6.9	3.1	5.1	2.3
143	6.8	3.2	5.9	2.3
144	6.7	3.3	5.7	2.5
145	6.7	3.0	5.2	2.3
146	6.3	2.5	5.0	1.9
147	6.5	3.0	5.2	2.0
148	6.2	3.4	5.4	2.3

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)
149	5.9	3.0	5.1	1.8

149 rows × 4 columns

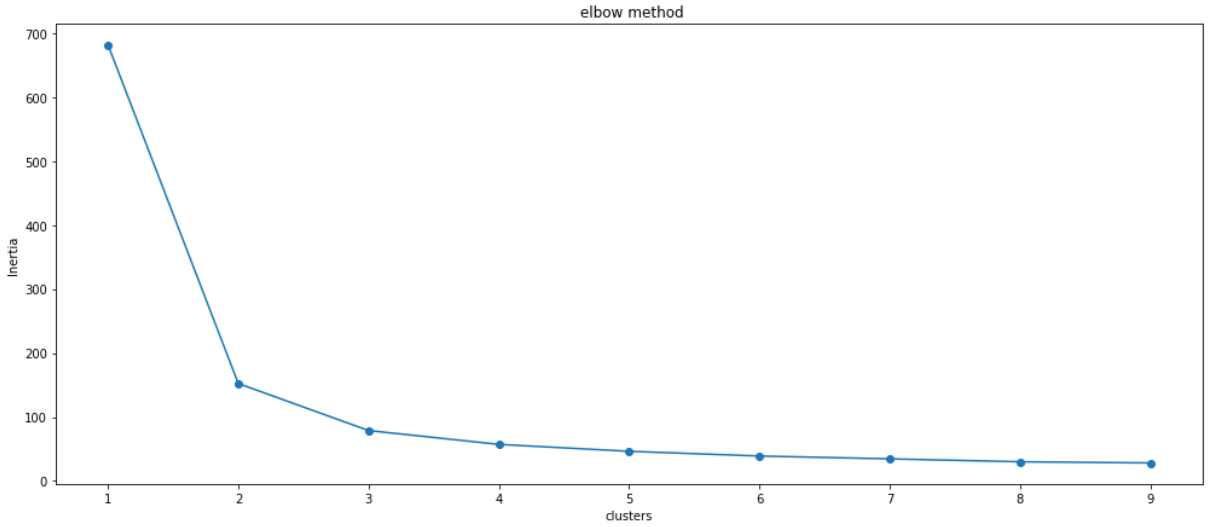
In [8]:
```
from sklearn.cluster import KMeans

SSE=[]
for i in range(1,10):
    kmeans= KMeans(n_clusters=i)
    kmeans.fit(data_iris)
    SSE.append(kmeans.inertia_)
```

In [9]:
```
DFrame =pd.DataFrame({'Cluster':range(1,10),'SSE':SSE})
```

In [10]:
```
plt.figure(figsize=(17,7))
plt.plot(DFrame['Cluster'],DFrame['SSE'],marker='o')
plt.xlabel("clusters")
plt.ylabel("Inertia")
plt.title(' elbow method')
```

Out[10]:
```
Text(0.5, 1.0, ' elbow method')
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD



In [11]:
```
x = data_iris.iloc[:, [0, 1, 2, 3]].values
kmeans= KMeans(n_clusters=3)
kmeans.fit(x)
pred =kmeans.predict(x)
pred
```

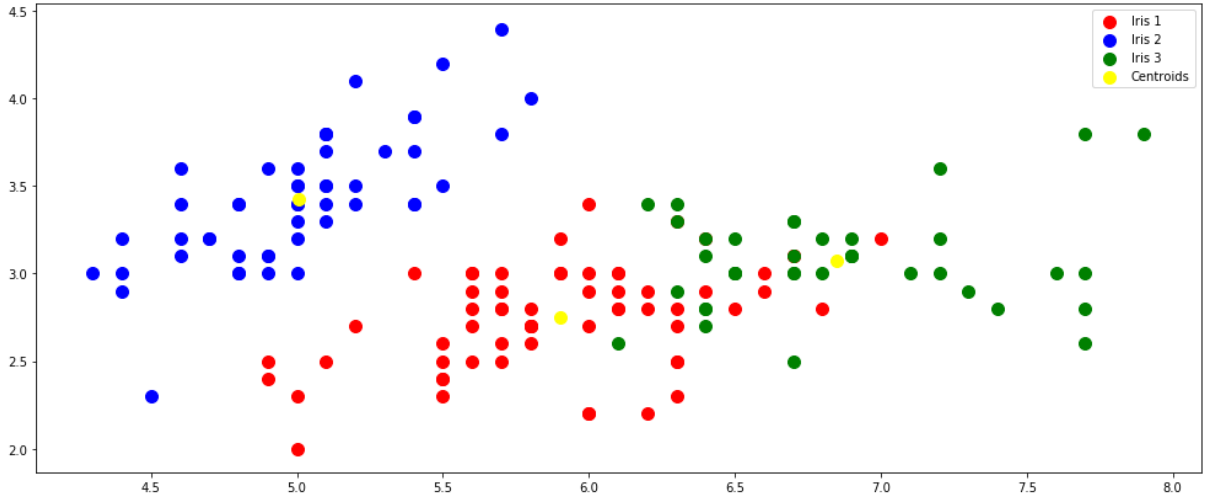
Out[11]:
```
array([1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1,
       1, 1, 1, 1, 1, 1, 0, 0, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 2, 0, 0, 0, 0, 0, 0, 0, 0,
0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 2, 0, 2, 2, 2, 2, 0, 2,
2,
       2, 2, 2, 0, 0, 2, 2, 2, 2, 0, 2, 0, 2, 0, 2, 2, 0, 0, 2, 2,
2,
       2, 0, 2, 2, 2, 2, 0, 2, 2, 2, 0, 2, 2, 2, 0, 2, 2, 0])
```

In [12]:
```
plt.figure(figsize=(17,7))
plt.scatter(x[pred == 0, 0], x[pred == 0, 1],
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

```
            s = 100, c = 'red', label = 'Iris 1')
plt.scatter(x[pred == 1, 0], x[pred == 1, 1],
            s = 100, c = 'blue', label = 'Iris 2')
plt.scatter(x[pred == 2, 0], x[pred == 2, 1],
            s = 100, c = 'green', label = 'Iris 3')


plt.scatter(kmeans.cluster_centers_[:, 0], kmeans.cluster_centers_[:,1
],
            s = 100, c = 'yellow', label = 'Centroids')

plt.legend()
```

Out[12]: <matplotlib.legend.Legend at 0x200d1215198>



In [ ]:

In [ ]:

In [ ]:

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD