prediction using unsupervised ML

In [7]:

```python
import pandas as pd
import numpy as np
from sklearn.cluster import KMeans
from sklearn.preprocessing import MinMaxScaler
import matplotlib.pyplot as plt
#matplotlib inline
```

In [9]:

```python
iris_df = pd.read_csv('Iris.csv')
```

In [11]:

```python
iris_df.head()
```

Out[11]:

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
0	1	5.1	3.5	1.4	0.2	Iris-setosa
1	2	4.9	3.0	1.4	0.2	Iris-setosa
2	3	4.7	3.2	1.3	0.2	Iris-setosa
3	4	4.6	3.1	1.5	0.2	Iris-setosa
4	5	5.0	3.6	1.4	0.2	Iris-setosa

In [13]:

```python
iris_df
```

Out[13]:

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
0	1	5.1	3.5	1.4	0.2	Iris-setosa
1	2	4.9	3.0	1.4	0.2	Iris-setosa
2	3	4.7	3.2	1.3	0.2	Iris-setosa

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
3	4	4.6	3.1	1.5	0.2	Iris-setosa
4	5	5.0	3.6	1.4	0.2	Iris-setosa
...	...	...	...	...	...	...
145	146	6.7	3.0	5.2	2.3	Iris-virginica
146	147	6.3	2.5	5.0	1.9	Iris-virginica
147	148	6.5	3.0	5.2	2.0	Iris-virginica
148	149	6.2	3.4	5.4	2.3	Iris-virginica
149	150	5.9	3.0	5.1	1.8	Iris-virginica

150 rows × 6 columns

Dropping the unecessery columns

```
In [15]: iris_df.drop(['Id', 'Species'], axis='columns',inplace=True)
```

```
In [16]: iris_df
```

```
Out[16]:
```

	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm
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
...	...	...	...	...
145	6.7	3.0	5.2	2.3
146	6.3	2.5	5.0	1.9

	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm
147	6.5	3.0	5.2	2.0
148	6.2	3.4	5.4	2.3
149	5.9	3.0	5.1	1.8

150 rows × 4 columns

KMeans

In [19]:

```
x=iris_df.iloc[:,[0,1,2,3]].values
sse = []
k_rng = range(1,10)
for k in k_rng:
    km = KMeans(n_clusters=k)
    km.fit(x)
    sse.append(km.inertia_)
```

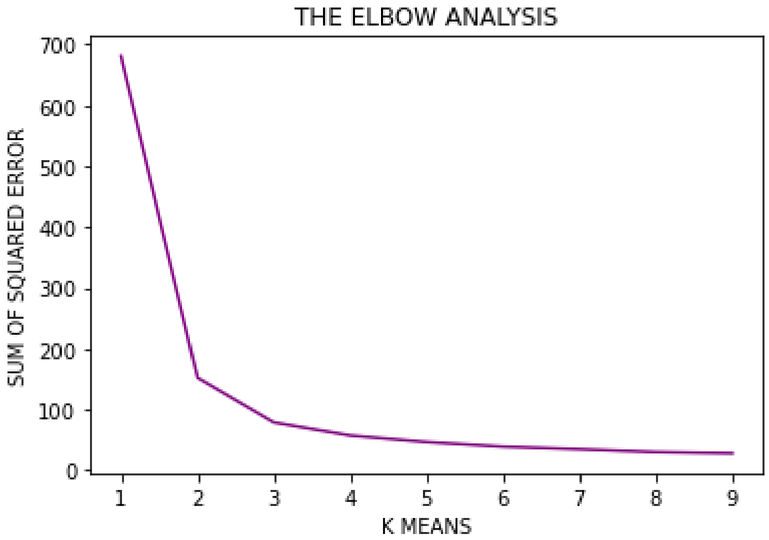
plotting an Elbow Graph to find the correct number of cluster

In [20]:

```
plt.xlabel('K MEANS')
plt.ylabel('SUM OF SQUARED ERROR')
plt.title('THE ELBOW ANALYSIS')
plt.plot(k_rng,sse,color='purple')
```

Out[20]:

```
[<matplotlib.lines.Line2D at 0x7f061cdee9b0>]
```



this show that 3 iis optimum number of cluster to form in iris dataset

In [23]:
```
km = KMeans(n_clusters=3)
y_predict = km.fit_predict(x)
y_predict
```

Out[23]:
```
array([0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0,
       0, 0, 0, 0, 0, 0, 2, 2, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
2,
       2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2,
2,
       2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 2, 1, 1, 1, 1, 2, 1, 1,
1,
       1, 1, 1, 2, 2, 1, 1, 1, 1, 2, 1, 2, 1, 2, 1, 1, 2, 2, 1, 1, 1,
1,
       1, 2, 1, 1, 1, 1, 2, 1, 1, 1, 2, 1, 1, 1, 2, 1, 1, 2], dtype=int
32)
```

Adding a cluster to show which cluster does the particular feature belong to

In [24]:

```
iris_df['cluster']=y_predict
iris_df
```

Out[24]:

	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	cluster
0	5.1	3.5	1.4	0.2	0
1	4.9	3.0	1.4	0.2	0
2	4.7	3.2	1.3	0.2	0
3	4.6	3.1	1.5	0.2	0
4	5.0	3.6	1.4	0.2	0
...	...	...	...	...	...
145	6.7	3.0	5.2	2.3	1
146	6.3	2.5	5.0	1.9	2
147	6.5	3.0	5.2	2.0	1
148	6.2	3.4	5.4	2.3	1
149	5.9	3.0	5.1	1.8	2

150 rows × 5 columns

In [25]:

```
iris_df.cluster.unique()
```

Out[25]:

```
array([0, 2, 1], dtype=int32)
```

In [26]:

```
iris_df1 = iris_df[iris_df.cluster==0]
```

```
iris_df2 = iris_df[iris_df.cluster==1]
iris_df3 = iris_df[iris_df.cluster==2]
```

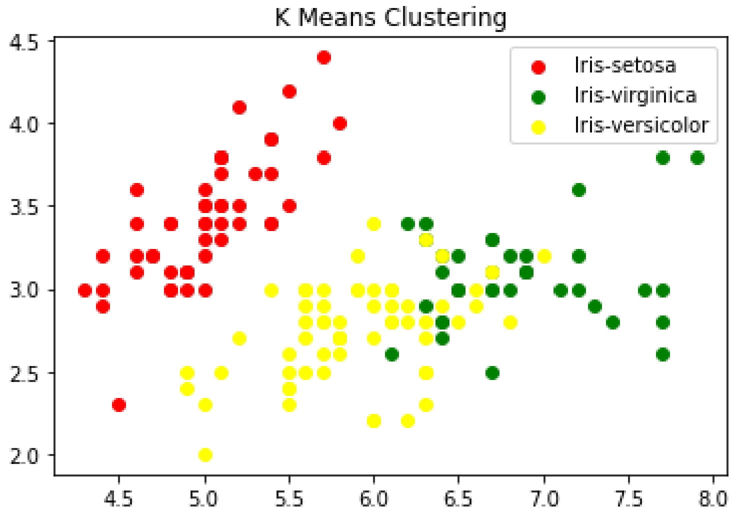
Plotting a Scatter plot showing the cluster

In [28]:

```
plt.title('K Means Clustering')
plt.scatter(x[y_predict==0,0],x[y_predict==0,1],c='red',label='Iris-setosa')
plt.scatter(x[y_predict==1,0],x[y_predict==1,1],c='green',label='Iris-virginica')
plt.scatter(x[y_predict==2,0],x[y_predict==2,1],c='yellow',label='Iris-versicolor')
plt.legend(loc='best')
```

Out[28]:

```
<matplotlib.legend.Legend at 0x7f061ca0d908>
```



Finish