In [28]:	import pandas as pd import numpy as np from sklearn.cluster import KMeans from sklearn.preprocessing import MinMaxScaler from matplotlib import pyplot as plt %matplotlib inline
In [5]: In [6]: Out[6]:	df=pd.read_csv("C:/Users/ankit/Downloads/Iris.csv") df.head(10) 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
	5 6 5.4 3.9 1.7 0.4 Iris-setosa 6 7 4.6 3.4 1.4 0.3 Iris-setosa 7 8 5.0 3.4 1.5 0.2 Iris-setosa 8 9 4.4 2.9 1.4 0.2 Iris-setosa 9 10 4.9 3.1 1.5 0.1 Iris-setosa
In [7]: Out[7]: In [8]:	<pre>df.shape (150, 6) df.info()</pre>
	<pre><class 'pandas.core.frame.dataframe'=""> RangeIndex: 150 entries, 0 to 149 Data columns (total 6 columns): # Column Non-Null Count Dtype</class></pre>
In [9]:	2 SepaiwidthCm 150 non-null float64 3 PetalLengthCm 150 non-null float64 4 PetalWidthCm 150 non-null float64 5 Species 150 non-null object dtypes: float64(4), int64(1), object(1) memory usage: 7.2+ KB df.describe()
Out[9]:	count 150.000000 150.000000 150.000000 150.000000 150.000000 150.000000 mean 75.500000 5.843333 3.054000 3.758667 1.198667 std 43.445368 0.828066 0.433594 1.764420 0.763161
	min 1.000000 4.300000 2.000000 0.100000 25% 38.250000 5.10000 1.60000 0.300000 50% 75.500000 5.80000 3.00000 1.300000 75% 112.750000 6.40000 3.30000 5.10000 1.800000 max 150.000000 7.900000 4.400000 6.900000 2.500000
In [12]:	Data Tansformation data=df.iloc[:,1:5] data.head()
Out[12]:	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 Scaling the dataset
In [24]: Out[24]:	mms=MinMaxScaler() mms.fit(data) da=mms.transform(data) da array([[0.22222222, 0.625 , 0.06779661, 0.04166667],
	[0.16666667, 0.41666667, 0.06779661, 0.04166667], [0.11111111, 0.5 , 0.05084746, 0.04166667], [0.08333333, 0.45833333, 0.08474576, 0.04166667], [0.19444444, 0.66666667, 0.06779661, 0.04166667], [0.30555556, 0.79166667, 0.11864407, 0.125], [0.08333333, 0.58333333, 0.06779661, 0.08333333], [0.19444444, 0.58333333, 0.08474576, 0.04166667], [0.02777778, 0.375 , 0.06779661, 0.04166667],
	[0.16666667, 0.45833333, 0.08474576, 0.], [0.30555556, 0.70833333, 0.08474576, 0.04166667], [0.13888889, 0.58333333, 0.10169492, 0.04166667], [0.13888889, 0.41666667, 0.06779661, 0.], [0. , 0.41666667, 0.01694915, 0.], [0.41666667, 0.83333333, 0.03389831, 0.04166667], [0.38888889, 1. , 0.08474576, 0.125], [0.30555556, 0.79166667, 0.05084746, 0.125],
	[0.2222222, 0.625
	[0.19444444, 0.41666667, 0.10169492, 0.04166667], [0.19444444, 0.58333333, 0.10169492, 0.04166667], [0.25 , 0.625 , 0.08474576, 0.04166667], [0.25 , 0.58333333, 0.06779661, 0.04166667], [0.11111111, 0.5 , 0.10169492, 0.04166667], [0.13888889, 0.45833333, 0.10169492, 0.04166667], [0.30555556, 0.58333333, 0.08474576, 0.125], [0.25 , 0.875 , 0.08474576, 0.],
	[0.33333333, 0.91666667, 0.06779661, 0.04166667], [0.16666667, 0.45833333, 0.08474576, 0.], [0.19444444, 0.5 , 0.03389831, 0.04166667], [0.333333333, 0.625 , 0.05084746, 0.04166667], [0.16666667, 0.45833333, 0.08474576, 0.], [0.16666667, 0.45833333, 0.08474576, 0.], [0.02777778, 0.41666667, 0.05084746, 0.04166667], [0.22222222, 0.58333333, 0.08474576, 0.04166667], [0.19444444, 0.625 , 0.05084746, 0.08333333],
	[0.05555556, 0.125
	[0.19444444, 0.54166667, 0.06779661, 0.04166667], [0.75 , 0.5 , 0.62711864, 0.54166667], [0.58333333, 0.5 , 0.59322034, 0.58333333], [0.72222222, 0.45833333, 0.66101695, 0.58333333], [0.33333333, 0.125 , 0.50847458, 0.5], [0.61111111, 0.33333333, 0.61016949, 0.58333333], [0.38888889, 0.33333333, 0.59322034, 0.5], [0.55555556, 0.54166667, 0.62711864, 0.625],
	[0.16666667, 0.16666667, 0.38983051, 0.375], [0.63888889, 0.375 , 0.61016949, 0.5], [0.25 , 0.29166667, 0.49152542, 0.54166667], [0.19444444, 0. , 0.42372881, 0.375], [0.44444444, 0.41666667, 0.54237288, 0.58333333], [0.47222222, 0.08333333, 0.50847458, 0.375], [0.5 , 0.375 , 0.62711864, 0.54166667], [0.36111111, 0.375 , 0.44067797, 0.5],
	[0.66666667, 0.45833333, 0.57627119, 0.54166667], [0.36111111, 0.41666667, 0.59322034, 0.58333333], [0.41666667, 0.29166667, 0.52542373, 0.375], [0.52777778, 0.08333333, 0.59322034, 0.58333333], [0.36111111, 0.20833333, 0.49152542, 0.41666667], [0.444444444, 0.5 , 0.6440678 , 0.70833333], [0.5 , 0.33333333, 0.50847458, 0.5], [0.555555556, 0.20833333, 0.66101695, 0.58333333],
	[0.5
	[0.33333333, 0.16666667, 0.45762712, 0.375], [0.41666667, 0.29166667, 0.49152542, 0.45833333], [0.47222222, 0.29166667, 0.69491525, 0.625], [0.30555556, 0.41666667, 0.59322034, 0.58333333], [0.47222222, 0.58333333, 0.59322034, 0.625], [0.66666667, 0.45833333, 0.62711864, 0.58333333], [0.55555556, 0.125 , 0.57627119, 0.5], [0.36111111, 0.41666667, 0.52542373, 0.5],
	[0.33333333, 0.20833333, 0.50847458, 0.5], [0.33333333, 0.25 , 0.57627119, 0.45833333], [0.5 , 0.41666667, 0.61016949, 0.54166667], [0.41666667, 0.25 , 0.50847458, 0.45833333], [0.19444444, 0.125 , 0.38983051, 0.375], [0.36111111, 0.29166667, 0.54237288, 0.5], [0.38888889, 0.41666667, 0.54237288, 0.45833333], [0.38888889, 0.375 , 0.54237288, 0.5],
	[0.52777778, 0.375
	[0.91666667, 0.41666667, 0.94915254, 0.83333333], [0.16666667, 0.20833333, 0.59322034, 0.66666667], [0.83333333, 0.375 , 0.89830508, 0.70833333], [0.66666667, 0.20833333, 0.81355932, 0.70833333], [0.80555556, 0.66666667, 0.86440678, 1.], [0.6111111, 0.5 , 0.69491525, 0.79166667], [0.58333333, 0.29166667, 0.72881356, 0.75], [0.69444444, 0.41666667, 0.76271186, 0.8333333],
	[0.38888889, 0.20833333, 0.6779661 , 0.79166667], [0.41666667, 0.33333333, 0.69491525, 0.95833333], [0.58333333, 0.5 , 0.72881356, 0.91666667], [0.61111111, 0.41666667, 0.76271186, 0.70833333], [0.94444444, 0.75 , 0.96610169, 0.875], [0.94444444, 0.25 , 1. , 0.91666667], [0.47222222, 0.08333333, 0.6779661 , 0.58333333], [0.72222222, 0.5 , 0.79661017, 0.91666667],
	[0.3611111, 0.33333333, 0.66101695, 0.79166667], [0.94444444, 0.33333333, 0.96610169, 0.79166667], [0.555555556, 0.29166667, 0.66101695, 0.70833333], [0.66666667, 0.54166667, 0.79661017, 0.83333333], [0.80555556, 0.5 , 0.84745763, 0.70833333], [0.52777778, 0.33333333, 0.6440678 , 0.70833333], [0.5 , 0.41666667, 0.66101695, 0.70833333], [0.5 , 0.41666667, 0.66101695, 0.70833333], [0.5 , 0.33333333, 0.77966102, 0.8333333],
	[0.80555556, 0.41666667, 0.81355932, 0.625], [0.86111111, 0.33333333, 0.86440678, 0.75], [1. , 0.75 , 0.91525424, 0.79166667], [0.58333333, 0.33333333, 0.77966102, 0.875], [0.55555556, 0.33333333, 0.69491525, 0.58333333], [0.5 , 0.25 , 0.77966102, 0.54166667], [0.94444444, 0.41666667, 0.86440678, 0.91666667], [0.955555556, 0.58333333, 0.77966102, 0.95833333],
	[0.58333333, 0.45833333, 0.76271186, 0.70833333], [0.47222222, 0.41666667, 0.6440678 , 0.70833333], [0.72222222, 0.45833333, 0.74576271, 0.8333333], [0.66666667, 0.45833333, 0.77966102, 0.95833333], [0.72222222, 0.45833333, 0.77966102, 0.95833333], [0.72222222, 0.45833333, 0.69491525, 0.91666667], [0.41666667, 0.29166667, 0.69491525, 0.75], [0.69444444, 0.5 , 0.83050847, 0.91666667], [0.60666667, 0.54166667, 0.79661017, 1.],
In [30]:	[0.66666667, 0.41666667, 0.71186441, 0.91666667], [0.55555556, 0.20833333, 0.6779661, 0.75], [0.61111111, 0.41666667, 0.71186441, 0.79166667], [0.52777778, 0.58333333, 0.74576271, 0.91666667], [0.44444444, 0.41666667, 0.69491525, 0.70833333]]) data.columns
Out[30]: In [31]: Out[31]:	<pre>Index(['SepalLengthCm', 'SepalWidthCm', 'PetalLengthCm', 'PetalWidthCm'], dtype='object') #convert to dataframe da=pd.DataFrame(da,columns=['SepalLengthCm', 'SepalWidthCm', 'PetalLengthCm']) da SepalLengthCm SepalWidthCm PetalLengthCm PetalWidthCm</pre>
	0 0.222222 0.625000 0.041667 1 0.166667 0.416667 0.041667 2 0.11111 0.500000 0.050847 0.041667 3 0.083333 0.458333 0.084746 0.041667
	4 0.194444 0.666667 0.067797 0.041667 145 0.666667 0.416667 0.711864 0.916667 146 0.555556 0.208333 0.677966 0.750000 147 0.611111 0.416667 0.711864 0.791667
	148 0.527778 0.583333 0.745763 0.916667 149 0.444444 0.416667 0.694915 0.708333 150 rows × 4 columns 4 columns
In [44]:	Elbow method to minimize wss(within cluster sum of square) ss=[] K=range(1,8) for k in K: km=KMeans(k)
In [47]: Out[47]:	<pre>km=km.fit(da) ss.append(km.inertia_) ss [41.13817202297779, 12.143688281579719, 0.00011400400070007</pre>
In [56]:	6.9981140048267605, 5.5328310030819, 4.577447617879765, 3.915800611971109, 3.4742909290175743] plt.plot(K,ss,color='blue',marker='*') plt.xlabel('k')
	plt.ylabel('sum of squared distance') plt.title('Elbow Method for optimal k') plt.show() Elbow Method for optimal k
	35 - 10 - 35 - 25 - 10 -
	from the plot ,we can conclude that cluster of number i.e $k=3$ will be optimum number of cluster for this datset
In [64]: Out[64]:	<pre>km3=KMeans(3) km3.fit(da) KMeans(n_clusters=3)</pre> km3.fit_predict(da)
In [65]:	array([1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1
<pre>In [66]: In [67]: Out[67]:</pre>	2, 0, 0, 2, 2, 2, 0, 2, 2, 2, 0, 2, 2, 2, 0, 2, 2, 0]) df['cluster']=km3.fit_predict(da) df ld SepalLengthCm SepalWidthCm PetalLengthCm PetalWidthCm Species cluster
	0 1 5.1 3.5 1.4 0.2 Iris-setosa 0 1 2 4.9 3.0 1.4 0.2 Iris-setosa 0 2 3 4.7 3.2 1.3 0.2 Iris-setosa 0 3 4 4.6 3.1 1.5 0.2 Iris-setosa 0 4 5 5.0 3.6 1.4 0.2 Iris-setosa 0
	148 149 6.2 3.4 5.4 2.3 Iris-virginica 1 149 150 5.9 3.0 5.1 1.8 Iris-virginica 2 150 rows × 7 columns
In [71]:	<pre>Visualization f, (ax1, ax2)=plt.subplots(1,2,figsize=(15,5)) f.suptitle('SepalLengthCm Vs SepalWidthCm') ax1.scatter(data['SepalLengthCm'], data['SepalWidthCm']) ax1.set_title('Before clustering', size=15) ax1.set(xlabel='SepalLengthCm', ylabel='SepalWidthCm')</pre>
Out[71]:	<pre>ax2.scatter(data['SepalLengthCm'], data['SepalWidthCm'], c=df['cluster'], cmap='rainbow') ax2.set_title('After clustering', size=15) ax2.set(xlabel='SepalLengthCm', ylabel='SepalWidthCm') [Text(0.5, 0, 'SepalLengthCm'), Text(0, 0.5, 'SepalWidthCm')]</pre>
	Before clustering 4.5 4.0 After clustering 4.5 4.0
	W 3.5 - W 3.5 - W 3.5 - W 3.5 - W 3.6 - W 3.5 - W 3.6
	2.5 - 2.0 - 2.0 - 2.0 -
In [73]:	<pre>f,(ax3,ax4)=plt.subplots(1,2,figsize=(15,5)) f.suptitle('PetalLengthCm Vs SepalWidthCm') ax3.scatter(data['PetalLengthCm'],data['PetalWidthCm']) ax3.set_title('Before clustering',size=15) ax3.set(xlabel='PetalLengthCm',ylabel='PetalWidthCm') ax4.scatter(data['PetalLengthCm'],data['PetalWidthCm'],c=df['cluster'],cmap='rainbow') ax4.scatter(data['PetalLengthCm'],data['PetalWidthCm'],c=df['cluster'],cmap='rainbow')</pre>
Out[73]:	ax4.set_title('After clustering', size=15) ax4.set(xlabel='PetalLengthCm', ylabel='PetalWidthCm') [Text(0.5, 0, 'PetalLengthCm'), Text(0, 0.5, 'PetalWidthCm')] PetalLengthCm Vs SepalWidthCm Before clustering After clustering
	2.5 - 2.0 -
	Definition of the property of
In []:	0.0 1 2 3 4 5 6 7 0.0 1 2 3 4 5 6 7 PetalLengthCm PetalLengthCm

Name:Ankita chaubey