In [1]:
```
import pandas as pd
import numpy as np
```

In [2]:
```
df=pd.read_csv("I:\AIML\income.csv")
```

In [3]:
```
df
```

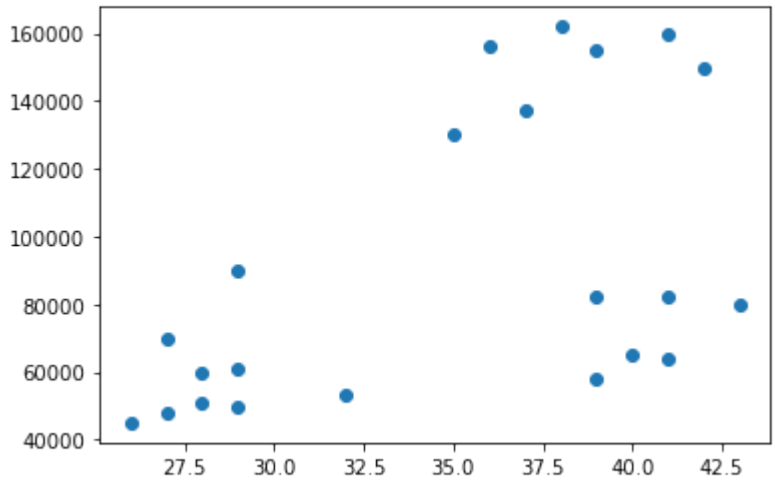
Out[3]:

| | name | age | imcome | Unnamed: 3 |
|---|---|---|---|---|
| **0** | Hritik | 27 | 70000 | NaN |
| **1** | Arpit | 29 | 90000 | NaN |
| **2** | Manav | 29 | 61000 | NaN |
| **3** | Kirti | 28 | 60000 | NaN |
| **4** | Siddhi | 42 | 150000 | NaN |
| **5** | Riya | 39 | 155000 | NaN |
| **6** | Ankita | 41 | 160000 | NaN |
| **7** | Vikash | 38 | 162000 | NaN |
| **8** | Priyank | 36 | 156000 | NaN |
| **9** | Kranti | 35 | 130000 | NaN |
| **10** | Rohit | 37 | 137000 | NaN |
| **11** | Aakash | 26 | 45000 | NaN |
| **12** | Durgesh | 27 | 48000 | NaN |
| **13** | Varun | 28 | 51000 | NaN |
| **14** | Vicky | 29 | 49500 | NaN |
| **15** | Priyank | 32 | 53000 | NaN |
| **16** | Kranti | 40 | 65000 | NaN |
| **17** | Rohit | 41 | 64000 | NaN |
| **18** | Aakash | 43 | 80000 | NaN |
| **19** | Durgesh | 39 | 82000 | NaN |
| **20** | Varun | 41 | 82000 | NaN |
| **21** | Vicky | 39 | 58000 | NaN |

In [4]:
```
from matplotlib import pyplot as pl
```

In [5]:
```
pl.scatter(df['age'],df['imcome'])
```

Out[5]:
```
<matplotlib.collections.PathCollection at 0x1d418ad4b20>
```




# Apply Kmean Clustering

In [6]:
```
from sklearn.cluster import KMeans
```

In [7]:
```
kmean=KMeans(n_clusters=3)
```

In [8]:
```
kmean
```

Out[8]:
```
KMeans(n_clusters=3)
```

In [9]:
```
y_predict=kmean.fit_predict(df[['age','imcome']])
```

In [10]:
```
y_predict
```

Out[10]:
```
array([0, 0, 2, 2, 1, 1, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 2, 2, 0, 0, 0, 2])
```

# Revise the dataframe

In [11]:
```
df['cluster']=y_predict
```

In [12]:
```
df
```

Out[12]:

| | name | age | imcome | Unnamed: 3 | cluster |
|---|---|---|---|---|---|
| **0** | Hritik | 27 | 70000 | NaN | 0 |
| **1** | Arpit | 29 | 90000 | NaN | 0 |
| **2** | Manav | 29 | 61000 | NaN | 2 |
| **3** | Kirti | 28 | 60000 | NaN | 2 |
| **4** | Siddhi | 42 | 150000 | NaN | 1 |
| **5** | Riya | 39 | 155000 | NaN | 1 |
| **6** | Ankita | 41 | 160000 | NaN | 1 |
| **7** | Vikash | 38 | 162000 | NaN | 1 |
| **8** | Priyank | 36 | 156000 | NaN | 1 |
| **9** | Kranti | 35 | 130000 | NaN | 1 |
| **10** | Rohit | 37 | 137000 | NaN | 1 |
| **11** | Aakash | 26 | 45000 | NaN | 2 |
| **12** | Durgesh | 27 | 48000 | NaN | 2 |
| **13** | Varun | 28 | 51000 | NaN | 2 |
| **14** | Vicky | 29 | 49500 | NaN | 2 |
| **15** | Priyank | 32 | 53000 | NaN | 2 |
| **16** | Kranti | 40 | 65000 | NaN | 2 |
| **17** | Rohit | 41 | 64000 | NaN | 2 |
| **18** | Aakash | 43 | 80000 | NaN | 0 |
| **19** | Durgesh | 39 | 82000 | NaN | 0 |
| **20** | Varun | 41 | 82000 | NaN | 0 |
| **21** | Vicky | 39 | 58000 | NaN | 2 |

In [13]:
```
kmean.predict([[34,50000]])
```

```
C:\Users\Admin\anaconda3\lib\site-packages\sklearn\base.py:450: UserWarnin
g: X does not have valid feature names, but KMeans was fitted with feature
names
  warnings.warn(
```

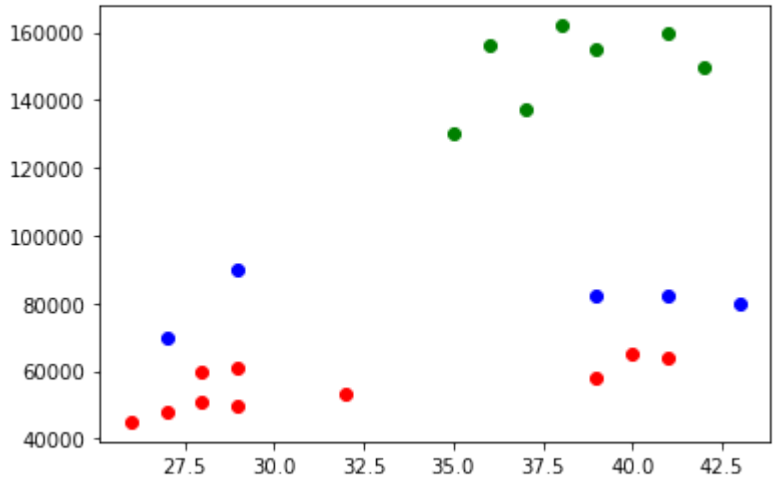
Out[13]:
```
array([2])
```

# # divide the dataframe according to the clusters

In [14]:
```
df1=df[df.cluster==0]
df2=df[df.cluster==1]
df3=df[df.cluster==2]
```

In [15]:
```
pl.scatter(df1.age,df1.imcome,color='blue')
pl.scatter(df2.age,df2.imcome,color='green')
pl.scatter(df3.age,df3.imcome,color='red')
```

Out[15]:
```
<matplotlib.collections.PathCollection at 0x1d41b7d5f10>
```



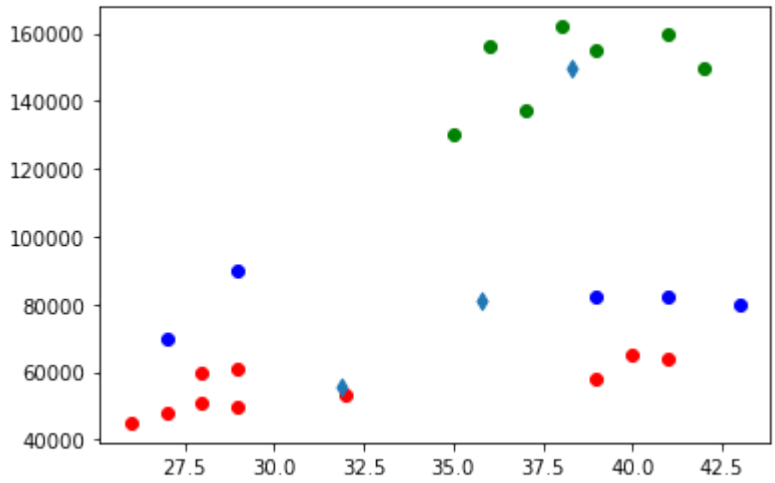

# display mean value

In [16]:
```
kmean.cluster_centers_
```

Out[16]:
```
array([[3.58000000e+01, 8.08000000e+04],
       [3.82857143e+01, 1.50000000e+05],
       [3.19000000e+01, 5.54500000e+04]])
```

In [17]:
```
pl.scatter(df1.age,df1.imcome,color='blue')
pl.scatter(df2.age,df2.imcome,color='green')
pl.scatter(df3.age,df3.imcome,color='red')
pl.scatter(kmean.cluster_centers_[:,0],
           kmean.cluster_centers_[:,1],marker='d')
```

Out[17]:
```
<matplotlib.collections.PathCollection at 0x1d41b8598b0>
```




## scale the attribute values from 0 to 1 by applying MinMaxScaler

In [18]:
```
from sklearn.preprocessing import MinMaxScaler
```

In [19]:
```
scaler=MinMaxScaler()
```

In [20]:
```
scaler.fit(df[['age']])
```

Out[20]:
```
MinMaxScaler()
```

In [21]:
```
df['Age']=scaler.transform(df[['age']])
```

In [22]:
```
df
```

Out[22]:

| | name | age | imcome | Unnamed: 3 | cluster | Age |
|---|---|---|---|---|---|---|
| **0** | Hritik | 27 | 70000 | NaN | 0 | 0.058824 |
| **1** | Arpit | 29 | 90000 | NaN | 0 | 0.176471 |
| **2** | Manav | 29 | 61000 | NaN | 2 | 0.176471 |
| **3** | Kirti | 28 | 60000 | NaN | 2 | 0.117647 |
| **4** | Siddhi | 42 | 150000 | NaN | 1 | 0.941176 |
| **5** | Riya | 39 | 155000 | NaN | 1 | 0.764706 |
| **6** | Ankita | 41 | 160000 | NaN | 1 | 0.882353 |
| **7** | Vikash | 38 | 162000 | NaN | 1 | 0.705882 |
| **8** | Priyank | 36 | 156000 | NaN | 1 | 0.588235 |
| **9** | Kranti | 35 | 130000 | NaN | 1 | 0.529412 |
| **10** | Rohit | 37 | 137000 | NaN | 1 | 0.647059 |
| **11** | Aakash | 26 | 45000 | NaN | 2 | 0.000000 |
| **12** | Durgesh | 27 | 48000 | NaN | 2 | 0.058824 |
| **13** | Varun | 28 | 51000 | NaN | 2 | 0.117647 |
| **14** | Vicky | 29 | 49500 | NaN | 2 | 0.176471 |
| **15** | Priyank | 32 | 53000 | NaN | 2 | 0.352941 |
| **16** | Kranti | 40 | 65000 | NaN | 2 | 0.823529 |
| **17** | Rohit | 41 | 64000 | NaN | 2 | 0.882353 |
| **18** | Aakash | 43 | 80000 | NaN | 0 | 1.000000 |
| **19** | Durgesh | 39 | 82000 | NaN | 0 | 0.764706 |
| **20** | Varun | 41 | 82000 | NaN | 0 | 0.882353 |
| **21** | Vicky | 39 | 58000 | NaN | 2 | 0.764706 |

In [23]:
```
scaler.fit(df[['imcome']])
```

Out[23]:
```
MinMaxScaler()
```

In [24]:
```
df['Income']=scaler.transform(df[['imcome']])
df
```

Out[24]:

| | name | age | imcome | Unnamed: 3 | cluster | Age | Income |
|---|---|---|---|---|---|---|---|
| **0** | Hritik | 27 | 70000 | NaN | 0 | 0.058824 | 0.213675 |
| **1** | Arpit | 29 | 90000 | NaN | 0 | 0.176471 | 0.384615 |
| **2** | Manav | 29 | 61000 | NaN | 2 | 0.176471 | 0.136752 |
| **3** | Kirti | 28 | 60000 | NaN | 2 | 0.117647 | 0.128205 |
| **4** | Siddhi | 42 | 150000 | NaN | 1 | 0.941176 | 0.897436 |
| **5** | Riya | 39 | 155000 | NaN | 1 | 0.764706 | 0.940171 |
| **6** | Ankita | 41 | 160000 | NaN | 1 | 0.882353 | 0.982906 |
| **7** | Vikash | 38 | 162000 | NaN | 1 | 0.705882 | 1.000000 |
| **8** | Priyank | 36 | 156000 | NaN | 1 | 0.588235 | 0.948718 |
| **9** | Kranti | 35 | 130000 | NaN | 1 | 0.529412 | 0.726496 |
| **10** | Rohit | 37 | 137000 | NaN | 1 | 0.647059 | 0.786325 |
| **11** | Aakash | 26 | 45000 | NaN | 2 | 0.000000 | 0.000000 |
| **12** | Durgesh | 27 | 48000 | NaN | 2 | 0.058824 | 0.025641 |
| **13** | Varun | 28 | 51000 | NaN | 2 | 0.117647 | 0.051282 |
| **14** | Vicky | 29 | 49500 | NaN | 2 | 0.176471 | 0.038462 |
| **15** | Priyank | 32 | 53000 | NaN | 2 | 0.352941 | 0.068376 |
| **16** | Kranti | 40 | 65000 | NaN | 2 | 0.823529 | 0.170940 |
| **17** | Rohit | 41 | 64000 | NaN | 2 | 0.882353 | 0.162393 |
| **18** | Aakash | 43 | 80000 | NaN | 0 | 1.000000 | 0.299145 |
| **19** | Durgesh | 39 | 82000 | NaN | 0 | 0.764706 | 0.316239 |
| **20** | Varun | 41 | 82000 | NaN | 0 | 0.882353 | 0.316239 |
| **21** | Vicky | 39 | 58000 | NaN | 2 | 0.764706 | 0.111111 |

In [25]:
```
km=KMeans(n_clusters=3)
y_predicted = km.fit_predict(df[['Age','Income']])
y_predicted
```

Out[25]:
```
array([1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 2])
```

In [26]:
```
df['cluster']=y_predicted
df
```

Out[26]:

| | name | age | imcome | Unnamed: 3 | cluster | Age | Income |
|---|---|---|---|---|---|---|---|
| 0 | Hritik | 27 | 70000 | NaN | 1 | 0.058824 | 0.213675 |
| 1 | Arpit | 29 | 90000 | NaN | 1 | 0.176471 | 0.384615 |
| 2 | Manav | 29 | 61000 | NaN | 1 | 0.176471 | 0.136752 |
| 3 | Kirti | 28 | 60000 | NaN | 1 | 0.117647 | 0.128205 |
| 4 | Siddhi | 42 | 150000 | NaN | 0 | 0.941176 | 0.897436 |
| 5 | Riya | 39 | 155000 | NaN | 0 | 0.764706 | 0.940171 |
| 6 | Ankita | 41 | 160000 | NaN | 0 | 0.882353 | 0.982906 |
| 7 | Vikash | 38 | 162000 | NaN | 0 | 0.705882 | 1.000000 |
| 8 | Priyank | 36 | 156000 | NaN | 0 | 0.588235 | 0.948718 |
| 9 | Kranti | 35 | 130000 | NaN | 0 | 0.529412 | 0.726496 |
| 10 | Rohit | 37 | 137000 | NaN | 0 | 0.647059 | 0.786325 |
| 11 | Aakash | 26 | 45000 | NaN | 1 | 0.000000 | 0.000000 |
| 12 | Durgesh | 27 | 48000 | NaN | 1 | 0.058824 | 0.025641 |
| 13 | Varun | 28 | 51000 | NaN | 1 | 0.117647 | 0.051282 |
| 14 | Vicky | 29 | 49500 | NaN | 1 | 0.176471 | 0.038462 |
| 15 | Priyank | 32 | 53000 | NaN | 1 | 0.352941 | 0.068376 |
| 16 | Kranti | 40 | 65000 | NaN | 2 | 0.823529 | 0.170940 |
| 17 | Rohit | 41 | 64000 | NaN | 2 | 0.882353 | 0.162393 |
| 18 | Aakash | 43 | 80000 | NaN | 2 | 1.000000 | 0.299145 |
| 19 | Durgesh | 39 | 82000 | NaN | 2 | 0.764706 | 0.316239 |
| 20 | Varun | 41 | 82000 | NaN | 2 | 0.882353 | 0.316239 |
| 21 | Vicky | 39 | 58000 | NaN | 2 | 0.764706 | 0.111111 |

# Assignment
## Implement Kmean clustering on the following dataset

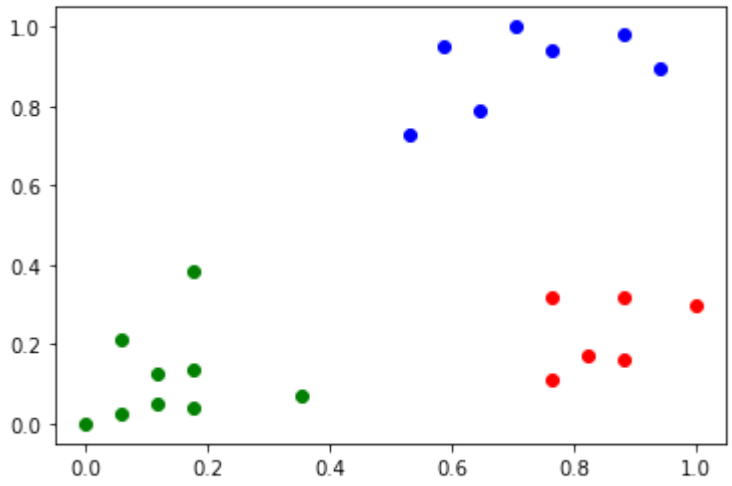
```
Object      Attribute1(X):Weight Index  Attribute 2(y):PH
Medicine A  1   1
Medicine B  2   1
Medicine C  4   3
Medicine D  5   4
```

In [27]:
```
km.cluster_centers_
```

Out[27]:
```
array([[0.72268908, 0.8974359 ],
       [0.1372549 , 0.11633428],
       [0.85294118, 0.22934473]])
```

In [34]:
```
df1=df[df.cluster==0]
df2=df[df.cluster==1]
df3=df[df.cluster==2]
pl.scatter(df1.Age,df1.Income,color='blue')
pl.scatter(df2.Age,df2.Income,color='green')
pl.scatter(df3.Age,df3.Income,color='red')
```

Out[34]:
```
<matplotlib.collections.PathCollection at 0x1d41babb160>
```




# # Elbow Method

In [29]:
```
k_range=range(1,10)
sse=[]
for k in k_range:
    km=KMeans(n_clusters=k)
    km.fit(df[['Age','Income']])
    sse.append(km.inertia_)
```

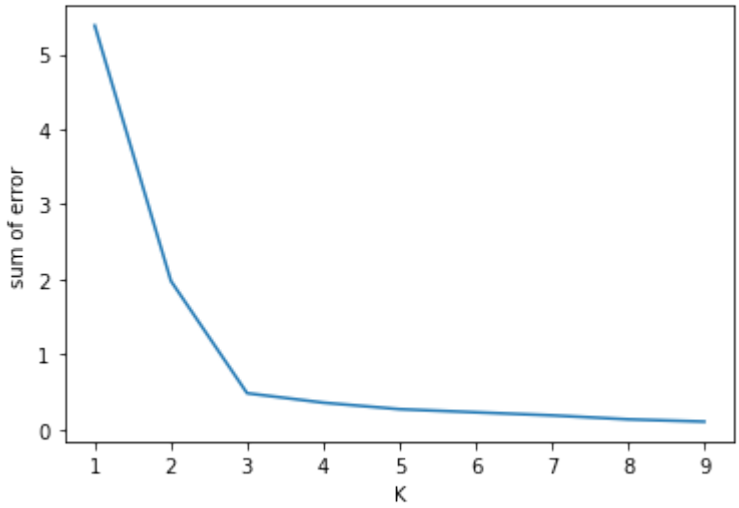
```
C:\Users\Admin\anaconda3\lib\site-packages\sklearn\cluster\_kmeans.py:1036:
UserWarning: KMeans is known to have a memory leak on Windows with MKL, whe
n there are less chunks than available threads. You can avoid it by setting
the environment variable OMP_NUM_THREADS=1.
  warnings.warn(
```

In [30]:
```
sse
```

Out[30]:
```
[5.383034948191102,
 1.9781765163703693,
 0.48132424071658514,
 0.35535060030323207,
 0.2684251670957969,
 0.2266223077689822,
 0.18593161552692655,
 0.13305958273870028,
 0.10240408154856742]
```

In [31]:
```
pl.xlabel('K')
pl.ylabel("sum of error")
pl.plot(k_range,sse)
```

Out[31]:
```
[<matplotlib.lines.Line2D at 0x1d41b979940>]
```




In [ ]: