In [42]:
```
import seaborn as sns
import pandas as pd
import numpy as np
from sklearn.cluster import KMeans
from sklearn.datasets import make_blobs
import matplotlib.pyplot as plt
from sklearn.metrics import confusion_matrix
```

In [2]:
```
help(make_blobs)
```

```
Help on function make_blobs in module sklearn.datasets._samples_generator:

make_blobs(n_samples=100, n_features=2, *, centers=None, cluster_std=1.0, center_b
ox=(-10.0, 10.0), shuffle=True, random_state=None, return_centers=False)
    Generate isotropic Gaussian blobs for clustering.

    Read more in the :ref:`User Guide <sample_generators>`.

    Parameters
    ----------
    n_samples : int or array-like, default=100
        If int, it is the total number of points equally divided among
        clusters.
        If array-like, each element of the sequence indicates
        the number of samples per cluster.

        .. versionchanged:: v0.20
            one can now pass an array-like to the ``n_samples`` parameter

    n_features : int, default=2
        The number of features for each sample.

    centers : int or ndarray of shape (n_centers, n_features), default=None
        The number of centers to generate, or the fixed center locations.
        If n_samples is an int and centers is None, 3 centers are generated.
        If n_samples is array-like, centers must be
        either None or an array of length equal to the length of n_samples.

    cluster_std : float or array-like of float, default=1.0
        The standard deviation of the clusters.

    center_box : tuple of float (min, max), default=(-10.0, 10.0)
        The bounding box for each cluster center when centers are
        generated at random.

    shuffle : bool, default=True
        Shuffle the samples.

    random_state : int, RandomState instance or None, default=None
        Determines random number generation for dataset creation. Pass an int
        for reproducible output across multiple function calls.
        See :term:`Glossary <random_state>`.

    return_centers : bool, default=False
        If True, then return the centers of each cluster.

        .. versionadded:: 0.23

    Returns
    -------
    X : ndarray of shape (n_samples, n_features)
        The generated samples.

    y : ndarray of shape (n_samples,)
        The integer labels for cluster membership of each sample.

    centers : ndarray of shape (n_centers, n_features)
        The centers of each cluster. Only returned if
        ``return_centers=True``.

    See Also
    --------
    make_classification : A more intricate variant.
```

```
Examples
--------
>>> from sklearn.datasets import make_blobs
>>> X, y = make_blobs(n_samples=10, centers=3, n_features=2,
...                   random_state=0)
>>> print(X.shape)
(10, 2)
>>> y
array([0, 0, 1, 0, 2, 2, 2, 1, 1, 0])
>>> X, y = make_blobs(n_samples=[3, 3, 4], centers=None, n_features=2,
...                   random_state=0)
>>> print(X.shape)
(10, 2)
>>> y
array([0, 1, 2, 0, 2, 2, 2, 1, 1, 0])
```

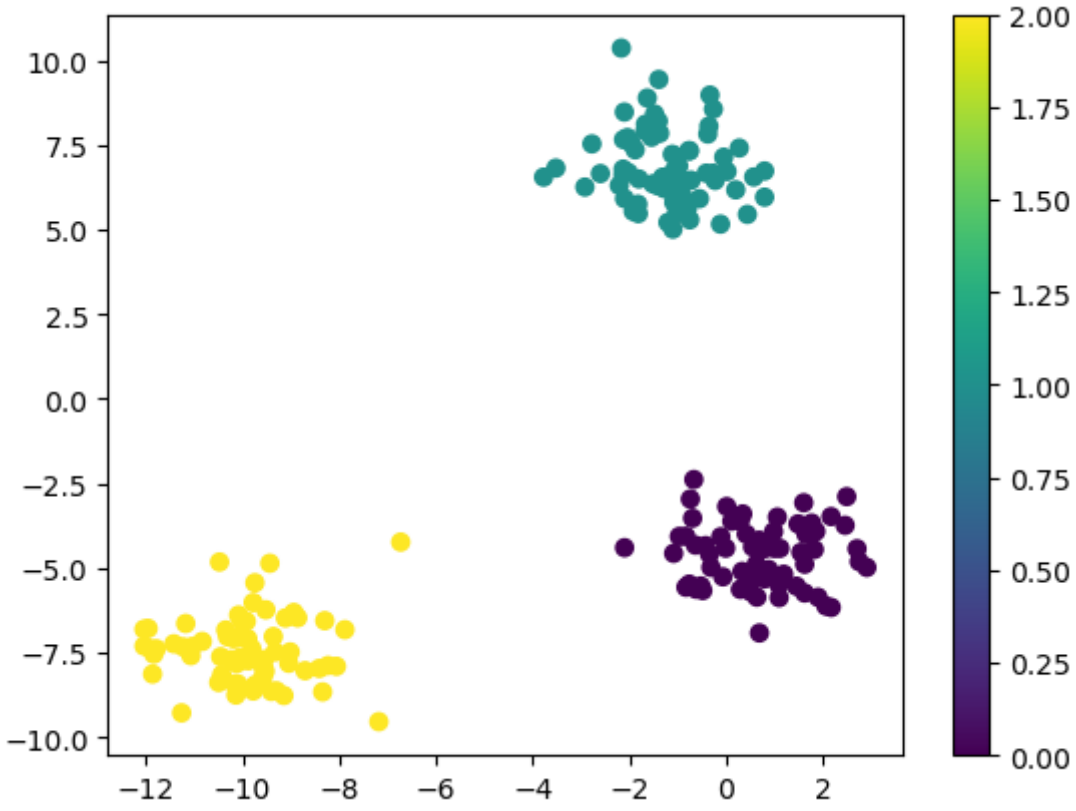
In [3]:
```
data= make_blobs(n_samples=200,centers=3,n_features=2,random_state=100)
```

In [4]:
```
data[1]
```

Out[4]:
```
array([2, 2, 2, 2, 1, 1, 0, 2, 0, 1, 0, 0, 1, 1, 0, 1, 0, 0, 2, 0, 1, 0,
       0, 2, 2, 1, 1, 0, 2, 0, 1, 2, 2, 1, 2, 0, 0, 1, 2, 2, 2, 0, 2, 0,
       2, 2, 2, 1, 0, 1, 0, 0, 0, 1, 1, 1, 0, 2, 1, 1, 0, 1, 0, 0, 0, 1,
       2, 2, 2, 0, 2, 1, 2, 2, 0, 2, 2, 2, 0, 2, 1, 0, 1, 1, 0, 0, 0, 2,
       1, 0, 0, 2, 2, 0, 1, 1, 2, 2, 0, 0, 1, 1, 1, 0, 0, 2, 1, 1, 0, 2,
       0, 2, 0, 1, 1, 1, 1, 2, 0, 2, 1, 0, 1, 0, 1, 1, 1, 0, 0, 0, 2, 0,
       1, 1, 0, 0, 1, 0, 0, 2, 1, 1, 2, 2, 1, 2, 1, 2, 1, 0, 2, 1, 0, 1,
       1, 2, 1, 2, 2, 2, 1, 2, 2, 1, 0, 0, 1, 2, 1, 2, 1, 0, 1, 0, 2, 1,
       1, 2, 2, 1, 0, 2, 0, 2, 2, 2, 0, 2, 2, 0, 1, 0, 0, 0, 1, 2, 0, 1,
       1, 2])
```

In [5]:
```
x=data[0][:,0]
y=data[0][:,1]
```

In [6]:
```
plt.scatter(x,y,c=data[1])
plt.colorbar()
plt.show()
```



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

In [8]:
```
km.fit(data[0])
```

```
C:\Users\Sony Vaio\anaconda3\lib\site-packages\sklearn\cluster\_kmeans.py:870: FutureWarning: The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of `n_init` explicitly to suppress the warning
  warnings.warn(
C:\Users\Sony Vaio\anaconda3\lib\site-packages\sklearn\cluster\_kmeans.py:1382: UserWarning: KMeans is known to have a memory leak on Windows with MKL, when there are less chunks than available threads. You can avoid it by setting the environment variable OMP_NUM_THREADS=1.
  warnings.warn(
```

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

In [9]:
```
km.cluster_centers_
```

Out[9]:
```
array([[-9.92850459, -7.41424071],
       [-1.21275037,  6.86392139],
       [ 0.68032106, -4.62125961]])
```

In [10]:
```
km.labels_
```

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

In [11]:
```
confusion_matrix(km.labels_,data[1])
```

Out[11]:
```
array([[ 0,  0, 66],
       [ 0, 67,  0],
       [67,  0,  0]], dtype=int64)
```

In [12]:
```
df=pd.read_csv('Downloads/College_data')
```

In [13]:
```
df
```

Out[13]:

	Unnamed: 0	Private	Apps	Accept	Enroll	Top10perc	Top25perc	F.Undergrad	P.Undergra
0	Abilene Christian University	Yes	1660	1232	721	23	52	2885	53
1	Adelphi University	Yes	2186	1924	512	16	29	2683	122
2	Adrian College	Yes	1428	1097	336	22	50	1036	9
3	Agnes Scott College	Yes	417	349	137	60	89	510	6
4	Alaska Pacific University	Yes	193	146	55	16	44	249	86
...	...	...	...	...	...	...	...	...	
772	Worcester State College	No	2197	1515	543	4	26	3089	202
773	Xavier University	Yes	1959	1805	695	24	47	2849	110
774	Xavier University of Louisiana	Yes	2097	1915	695	34	61	2793	16
775	Yale University	Yes	10705	2453	1317	95	99	5217	8
776	York College of Pennsylvania	Yes	2989	1855	691	28	63	2988	172

777 rows × 19 columns

In [14]:
```
df2=df.drop('Unnamed: 0',axis=1)
```

In [15]:
```
df2
```

Out[15]:

	Private	Apps	Accept	Enroll	Top10perc	Top25perc	F.Undergrad	P.Undergrad	Outstate
0	Yes	1660	1232	721	23	52	2885	537	7440
1	Yes	2186	1924	512	16	29	2683	1227	12280
2	Yes	1428	1097	336	22	50	1036	99	11250
3	Yes	417	349	137	60	89	510	63	12960
4	Yes	193	146	55	16	44	249	869	7560
...	...	...	...	...	...	...	...	...	...
772	No	2197	1515	543	4	26	3089	2029	6797
773	Yes	1959	1805	695	24	47	2849	1107	11520
774	Yes	2097	1915	695	34	61	2793	166	6900
775	Yes	10705	2453	1317	95	99	5217	83	19840
776	Yes	2989	1855	691	28	63	2988	1726	4990

777 rows × 18 columns

In [16]:
```
km=KMeans(2)
```

In [17]:
```
km.fit(df2.drop('Private', axis=1))
```

```
C:\Users\Sony Vaio\anaconda3\lib\site-packages\sklearn\cluster\_kmeans.py:870: Fut
ureWarning: The default value of `n_init` will change from 10 to 'auto' in 1.4. Se
t the value of `n_init` explicitly to suppress the warning
  warnings.warn(
```

Out[17]:
```
KMeans(n_clusters=2)
```

In [18]:
```
km.cluster_centers_
```

Out[18]:
```
array([[1.81323468e+03, 1.28716592e+03, 4.91044843e+02, 2.53094170e+01,
        5.34708520e+01, 2.18854858e+03, 5.95458894e+02, 1.03957085e+04,
        4.31136472e+03, 5.41982063e+02, 1.28033632e+03, 7.04424514e+01,
        7.78251121e+01, 1.40997010e+01, 2.31748879e+01, 8.93204634e+03,
        6.51195815e+01],
       [1.03631389e+04, 6.55089815e+03, 2.56972222e+03, 4.14907407e+01,
        7.02037037e+01, 1.30619352e+04, 2.46486111e+03, 1.07191759e+04,
        4.64347222e+03, 5.95212963e+02, 1.71420370e+03, 8.63981481e+01,
        9.13333333e+01, 1.40277778e+01, 2.00740741e+01, 1.41705000e+04,
        6.75925926e+01]])
```

In [19]:
```
km.labels_
```

Out[19]:
```
array([0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1,
       0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0,
       0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 1,
       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, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0,
       1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1,
       0, 1, 1, 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, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 1, 0, 1, 0, 0, 0, 0, 1, 0,
       0, 0, 1, 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, 1, 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, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 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, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0,
       1, 1, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0,
       0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1,
       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, 1, 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, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 0, 1, 1, 1, 0, 0, 1, 0,
       0, 0, 0, 1, 1, 0, 1, 1, 1, 0, 1, 0, 1, 0, 0, 0, 0, 1, 1, 0, 1, 1,
       0, 0, 1, 0, 1, 0, 0, 0, 0, 1, 0, 1, 0, 1, 1, 0, 0, 0, 0, 1, 0, 0,
       1, 1, 1, 1, 1, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 1, 1, 1, 0, 0, 0, 0,
       0, 1, 0, 1, 0, 0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 0, 0, 1, 1, 0, 0,
       0, 0, 0, 0, 1, 0, 1, 1, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0,
       0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 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, 1, 0])
```

In [20]:
```
x=df2['Private']
for i in range(len(x)):
    if x[i]=='Yes':
        x[i]=1
    elif x[i]=='No':
        x[i]=0
```

```
C:\Users\Sony Vaio\AppData\Local\Temp\ipykernel_12352\3068038229.py:4: SettingWith
CopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stabl
e/user_guide/indexing.html#returning-a-view-versus-a-copy
  x[i]=1
C:\Users\Sony Vaio\AppData\Local\Temp\ipykernel_12352\3068038229.py:6: SettingWith
CopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stabl
e/user_guide/indexing.html#returning-a-view-versus-a-copy
  x[i]=0
```

In [21]:
```
df2
```

Out[21]:

	Private	Apps	Accept	Enroll	Top10perc	Top25perc	F.Undergrad	P.Undergrad	Outstate
0	1	1660	1232	721	23	52	2885	537	7440
1	1	2186	1924	512	16	29	2683	1227	12280
2	1	1428	1097	336	22	50	1036	99	11250
3	1	417	349	137	60	89	510	63	12960
4	1	193	146	55	16	44	249	869	7560
...	...	...	...	...	...	...	...	...	...
772	0	2197	1515	543	4	26	3089	2029	6797
773	1	1959	1805	695	24	47	2849	1107	11520
774	1	2097	1915	695	34	61	2793	166	6900
775	1	10705	2453	1317	95	99	5217	83	19840
776	1	2989	1855	691	28	63	2988	1726	4990

777 rows × 18 columns

In [22]:
```
len(df2['Private'])
print(df2['Private'])
print(km.labels_)
```

```
0      1
1      1
2      1
3      1
4      1
      ..
772    0
773    1
774    1
775    1
776    1
Name: Private, Length: 777, dtype: object
[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 1 0 0 0 1 0 0 0 0 0 0 0 0
 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 0 0 0 0 0 0 0 1 1 0 0 0
 0 0 0 0 0 1 1 0 0 0 0 0 0 0 1 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 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 1 0 0 0
 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1
 1 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 1 0 0 0 0 0 0 1 0
 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 1 0 0 1 0 1 0 0 0 0 1 0 0 0 1 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 1 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 1 0 0 0 0 0 0 0 0 0 1 1 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
 1 0 0 0 0 1 0 0 0 0 0 1 1 1 0 0 0 1 0 0 0 0 0 0 0 1 0 0 0 1 0 0 0 0 0 0 0
 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 1 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 1 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 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 1 0 0 0 0 0 0 0 1 1 1 0 1 1 1 0 0 1 0 0 0 0 1 1 0 1 1 1 0 1 0 1
 0 0 0 0 1 1 0 1 1 0 0 1 0 1 0 0 0 0 1 0 1 0 1 1 0 0 0 0 1 0 0 1 1 1 1 1 0
 0 1 0 1 0 0 0 0 0 1 1 1 0 0 0 0 0 1 0 1 0 0 0 0 0 1 1 1 1 0 0 0 0 0 1 1 0
 0 0 0 0 0 1 0 1 1 0 1 0 0 0 0 0 0 1 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0
 0 0 0 1 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 1 0]
```

In [23]:
```
len(km.labels_)
```

Out[23]:
```
777
```

In [44]:
```
y=(df2["Private"].astype(int))
```

In [45]:
```
confusion_matrix(km.labels_,y)
```

Out[45]:
```
array([[138, 531],
       [ 74,  34]], dtype=int64)
```

In [2]:
```
((138+134)*100)/(531+74+138+34)
```

Out[2]:
```
35.006435006435005
```

In [ ]:
```

```