K-Means Clustering

Import Packages

In [33]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

%matplotlib inline
```

Investigate Dataset

Visualize Data

In [34]:
```
from sklearn.datasets import make_blobs
```

In [35]:
```
raw_data = make_blobs(
    n_samples = 200,
    n_features = 2,
    centers = 4,
    cluster_std = 1.8
)
```

In [36]:
```
display(raw_data)
```

```
(array([[ 8.24584393e+00, -4.23939536e-01],
        [-4.95804810e+00,  7.47637297e-01],
        [ 6.61695489e+00, -1.36447342e+00],
        [-4.21607021e+00, -1.31254862e+00],
        [ 4.79325496e-01, -5.24863568e-01],
        [ 7.33346724e+00,  2.23008649e+00],
        [ 8.90195934e+00,  4.68724873e+00],
        [ 6.63097360e+00, -2.19596527e-01],
        [-5.38979972e+00,  1.40182902e-03],
        [ 1.05371533e+01,  5.96549871e+00],
        [ 6.47348551e+00,  2.82490697e-02],
        [ 8.23750895e+00,  5.37858170e+00],
        [-5.15149934e-01, -2.02083124e+00],
        [-2.49896002e+00, -1.30305864e+00],
        [-3.59987222e+00, -2.72428576e+00],
        [ 5.58899359e+00,  9.30710627e+00],
        [-5.67155996e+00, -2.44748753e+00],
        [-2.72278175e+00, -2.88158070e+00],
        [-1.94835212e+00, -1.44642403e+00],
        [-7.72555454e+00,  3.01313424e+00],
        [ 8.85763447e+00,  1.71068394e+00],
        [-1.70303712e+00, -1.70299053e+00],
        [-5.71401616e+00, -3.44001393e+00],
        [ 6.27992969e+00, -1.04597927e+00],
        [ 6.36316948e+00, -3.49993637e-01],
        [ 6.08914360e+00, -2.78068514e+00],
        [ 9.92448917e+00,  9.49924767e+00],
        [ 8.98942472e-01, -7.32522365e-01],
        [-7.38192413e+00, -1.13553527e+00],
        [ 1.06048540e+01,  7.38926534e+00],
        [-1.01964717e+00, -1.05409645e+00],
        [-5.94130403e+00,  1.10921299e-01],
        [-6.64817602e+00, -8.56500701e-01],
        [ 7.09864768e+00, -6.12930977e-01],
        [ 6.41012120e+00, -2.40482112e+00],
        [ 7.66205003e+00, -1.49994337e+00],
        [ 9.64768142e+00, -1.37430927e+00],
        [ 9.24640210e+00,  2.96995909e+00],
        [ 7.22690991e+00,  5.41961736e+00],
        [ 7.27190588e+00, -1.77940759e+00],
        [ 8.40315221e+00,  8.30781179e+00],
        [-1.05608172e+00,  2.30796211e-01],
        [-9.23915455e-01, -6.72605096e-01],
        [-9.94151351e-01, -1.50917537e+00],
        [ 7.50697330e+00,  5.92991517e+00],
        [-5.07873799e+00, -1.20293471e+00],
        [ 1.03675834e+01,  4.69569897e+00],
        [ 9.98667077e+00,  4.90655387e+00],
        [-1.33962888e+00,  2.07656554e+00],
        [-4.62154428e-01, -1.54549212e+00],
        [-3.66135850e+00, -2.43467825e+00],
        [-2.94098233e+00, -1.84697472e+00],
        [-9.78221395e-01, -2.80749565e+00],
        [ 5.23243194e+00,  2.26466310e+00],
        [-3.33743508e+00,  9.07309546e-02],
        [-1.63733646e-01,  2.07259343e+00],
```

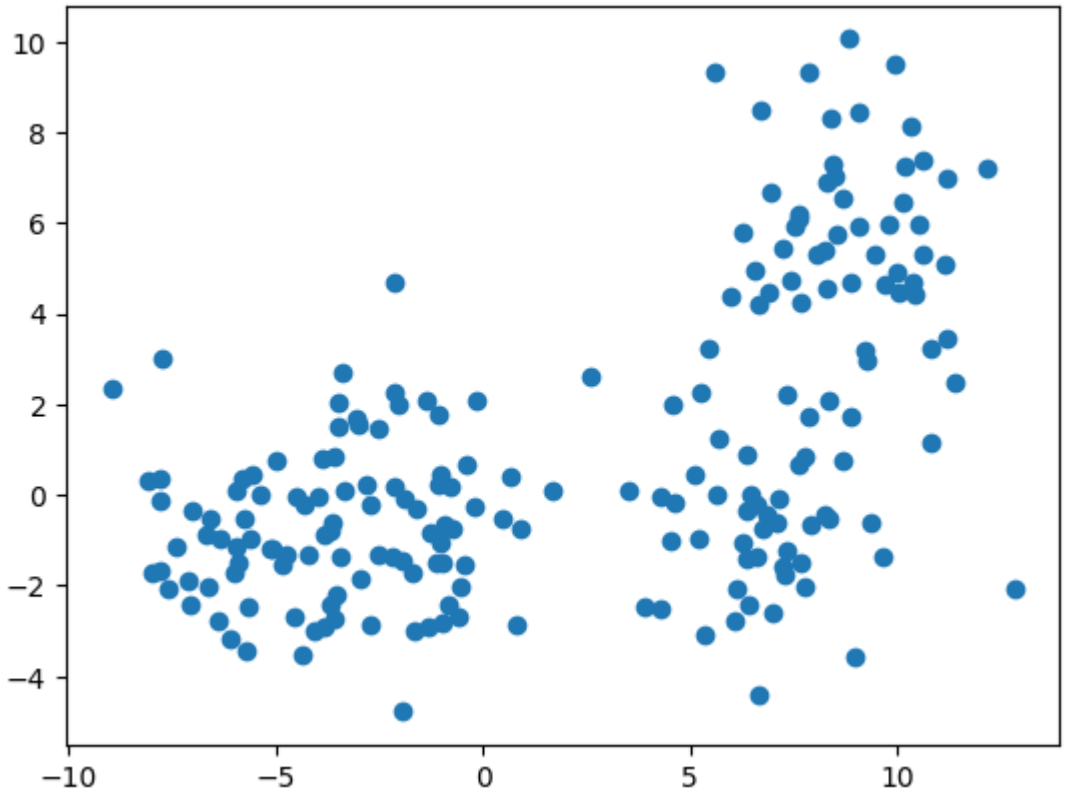
```
[-3.64993898e+00, -6.03351324e-01],
[ 1.28158931e+01, -2.08067150e+00],
[ 6.87370067e+00, -4.39821035e-01],
[ 4.29752970e+00, -2.52311740e+00],
[ 7.85644517e+00,  1.71853646e+00],
[ 5.45647754e+00,  3.24802559e+00],
[-5.93487378e+00, -1.12936429e+00],
[ 5.64749299e+00, -1.89581249e-04],
[-3.48186489e+00,  2.04915312e+00],
[ 8.99093158e+00, -3.57956601e+00],
[ 1.07989704e+01,  3.23792181e+00],
[-1.64892457e+00, -2.97757233e+00],
[ 7.12881107e+00, -9.95827079e-02],
[ 1.11932591e+01,  3.45284499e+00],
[ 1.07937879e+01,  1.14014411e+00],
[ 8.32558094e+00,  2.08482362e+00],
[-3.89134648e+00,  7.84356617e-01],
[ 6.27824698e+00,  5.77650163e+00],
[ 7.41957183e+00,  4.71149664e+00],
[ 6.50535163e-01,  3.84940140e-01],
[-3.80474731e+00, -8.85309933e-01],
[-1.98109637e-01, -2.65598183e-01],
[-7.80287609e+00, -1.07444160e-01],
[ 3.54176444e+00,  1.08768382e-01],
[ 1.12076391e+01,  6.97212814e+00],
[ 6.95807779e+00,  6.69594322e+00],
[ 5.68437926e+00,  1.26083427e+00],
[-1.26842897e+00, -8.10159726e-01],
[ 6.65103580e+00,  4.18106071e+00],
[-1.90376203e+00, -6.49318930e-02],
[-3.56229169e+00,  8.61691833e-01],
[ 9.05440845e+00,  8.45892439e+00],
[ 6.38953319e+00,  8.79497784e-01],
[-3.84479248e+00, -2.92631925e+00],
[ 1.01534583e+01,  6.43825079e+00],
[-6.06560374e-01, -2.67678487e+00],
[-2.11226325e+00,  2.26061665e+00],
[-1.10186034e+00, -1.51130031e+00],
[-2.17111889e+00, -1.35332677e+00],
[ 8.30392483e+00,  4.57013528e+00],
[-4.08027871e+00, -2.99457564e+00],
[ 1.04267135e+01,  4.41813650e+00],
[-4.47829447e+00, -3.94589152e-02],
[-6.56335013e+00, -5.03397656e-01],
[-1.03271514e+00,  4.47023259e-01],
[-7.79555825e+00,  3.58099185e-01],
[ 6.67155522e+00, -4.40203577e+00],
[ 9.35638475e+00, -6.20557963e-01],
[ 8.69122880e+00,  7.61307304e-01],
[-7.08907473e+00, -1.87981833e+00],
[ 3.90583230e+00, -2.45750435e+00],
[-5.81561163e+00,  3.57364492e-01],
[ 6.57983407e+00,  4.94331668e+00],
[-2.04059816e+00,  1.98740382e+00],
[-4.82773733e+00, -1.54845757e+00],
[ 1.11703278e+01,  5.09583092e+00],
```

```
[ 6.91310599e+00,  4.45636900e+00],
[-2.53579364e+00,  1.44697930e+00],
[-2.13056827e+00,  4.70790164e+00],
[ 8.83288494e+00,  1.00548966e+01],
[-7.06469675e+00, -2.42090306e+00],
[ 8.04828170e+00,  5.29143935e+00],
[-8.34314342e-01, -2.43232352e+00],
[ 1.03147668e+01,  8.11765383e+00],
[ 8.45562902e+00,  7.29400461e+00],
[-6.33887198e+00, -9.57002258e-01],
[-4.35476442e+00, -3.51089442e+00],
[ 9.46216895e+00,  5.32545772e+00],
[-2.99189591e+00,  1.55175906e+00],
[ 7.61542808e+00,  6.17144503e+00],
[-8.95799437e+00,  2.35690512e+00],
[-3.68364325e+00, -7.83471023e-01],
[ 5.34851932e+00, -3.08050970e+00],
[ 5.21259336e+00, -9.50652604e-01],
[-2.80507986e+00,  2.13477204e-01],
[-3.06069206e+00,  1.68425402e+00],
[ 8.66107252e+00,  6.54547375e+00],
[ 2.60485885e+00,  2.59535215e+00],
[-3.47773564e+00,  1.51470151e+00],
[-3.38326368e+00,  2.70355938e+00],
[-7.66722930e-01,  1.94769630e-01],
[-1.92250277e+00, -4.78248230e+00],
[-1.58702048e+00, -2.88638753e-01],
[-7.11847973e-01, -7.64955832e-01],
[-1.09232540e+00,  1.75827181e+00],
[ 9.70449362e+00,  4.65614016e+00],
[-2.70987832e+00, -1.96532956e-01],
[ 8.29988346e+00,  6.91436126e+00],
[ 6.73576287e+00, -7.45960846e-01],
[-7.01201106e+00, -3.60379451e-01],
[ 7.77891645e+00,  8.30018061e-01],
[-6.07997841e+00, -3.16894871e+00],
[ 1.06300144e+01,  5.28739327e+00],
[-6.35736068e+00, -2.77725448e+00],
[ 6.97643260e+00, -2.58301413e+00],
[ 7.85194441e+00,  9.30942588e+00],
[ 5.10129924e+00,  4.64540875e-01],
[ 7.66583720e+00,  4.26224391e+00],
[-1.32584138e+00, -2.90220193e+00],
[ 9.78015590e+00,  5.98586728e+00],
[ 7.60347620e+00,  6.12156212e+00],
[-5.88484464e+00, -1.49719457e+00],
[ 7.74456184e+00, -2.02888427e+00],
[-3.73427372e-01,  6.91724721e-01],
[ 7.92693384e+00, -6.60571283e-01],
[-3.54934379e+00, -2.20589758e+00],
[ 7.24773397e+00, -1.58547745e+00],
[ 1.01654869e+01,  7.25113419e+00],
[ 4.56828062e+00,  2.00782439e+00],
[ 1.00509040e+01,  4.46822285e+00],
[ 7.63983673e+00,  6.64552575e-01],
[-4.71608184e+00, -1.31366148e+00],
```

```
        [-7.77315216e+00, -1.67100505e+00],
        [ 8.49700924e+00,  7.02747110e+00],
        [ 8.34390986e+00, -5.38153767e-01],
        [-8.06168425e+00,  3.01427479e-01],
        [-5.11224564e+00, -1.17786687e+00],
        [ 5.99570748e+00,  4.39935964e+00],
        [-4.28617837e+00, -2.06634092e-01],
        [-3.94294541e+00, -2.04462527e-02],
        [-2.15365506e+00,  1.84316148e-01],
        [ 4.54783998e+00, -1.00617794e+00],
        [ 9.06534802e+00,  5.90817852e+00],
        [ 1.21739114e+01,  7.20124632e+00],
        [ 9.19224606e+00,  3.18843303e+00],
        [-5.77318718e+00, -5.41463205e-01],
        [-3.42441506e+00, -1.34650645e+00],
        [-7.60986631e+00, -2.05693864e+00],
        [ 6.34411178e+00, -1.41425700e+00],
        [-7.95477025e+00, -1.72179275e+00],
        [ 4.28106998e+00, -5.23850829e-02],
        [ 8.52076555e+00,  5.75212027e+00],
        [-6.64380983e+00, -2.02574426e+00],
        [-5.62894676e+00, -9.53420035e-01],
        [ 1.69572743e+00,  1.03944827e-01],
        [-4.54261297e+00, -2.69756161e+00],
        [ 6.68636896e+00,  8.50803169e+00],
        [ 6.14712944e+00, -2.08556435e+00],
        [ 4.62813120e+00, -1.56715815e-01],
        [ 8.12680280e-01, -2.88375619e+00],
        [ 7.34076186e+00, -1.22823915e+00],
        [-5.97537368e+00, -1.72629531e+00],
        [ 1.13976704e+01,  2.46478259e+00],
        [-5.56551088e+00,  4.37748241e-01]]),
 array([0, 1, 0, 1, 1, 0, 2, 0, 3, 2, 0, 2, 1, 1, 3, 2, 3, 1, 1, 3, 0, 1,
        3, 0, 0, 0, 2, 1, 3, 2, 3, 3, 3, 0, 0, 0, 0, 2, 2, 0, 2, 1, 1, 1,
        2, 3, 2, 2, 1, 1, 1, 1, 1, 0, 1, 1, 3, 0, 0, 0, 0, 0, 3, 0, 1, 0,
        2, 1, 0, 2, 0, 0, 3, 2, 2, 1, 3, 1, 3, 0, 2, 2, 0, 1, 2, 1, 3, 2,
        0, 3, 2, 1, 1, 1, 1, 2, 3, 2, 3, 3, 1, 3, 0, 0, 0, 3, 0, 3, 2, 1,
        3, 2, 2, 1, 1, 2, 3, 2, 3, 2, 2, 3, 3, 2, 3, 2, 3, 3, 0, 0, 1, 3,
        2, 1, 1, 1, 1, 1, 1, 1, 3, 2, 1, 2, 0, 3, 0, 3, 2, 3, 0, 2, 0, 2,
        1, 2, 2, 3, 0, 1, 0, 3, 0, 2, 0, 2, 0, 3, 3, 2, 0, 3, 3, 2, 3, 1,
        1, 0, 2, 2, 2, 1, 3, 3, 0, 3, 0, 2, 3, 3, 1, 1, 2, 0, 0, 1, 0, 3,
        2, 3]))
```

In [37]:
```
# Verify our Dataset has 4 cluster centers.
# Plot all rows in the first column of Dataset against all rows in second column of
plt.scatter(raw_data[0][:, 0], raw_data[0][:, 1])
```

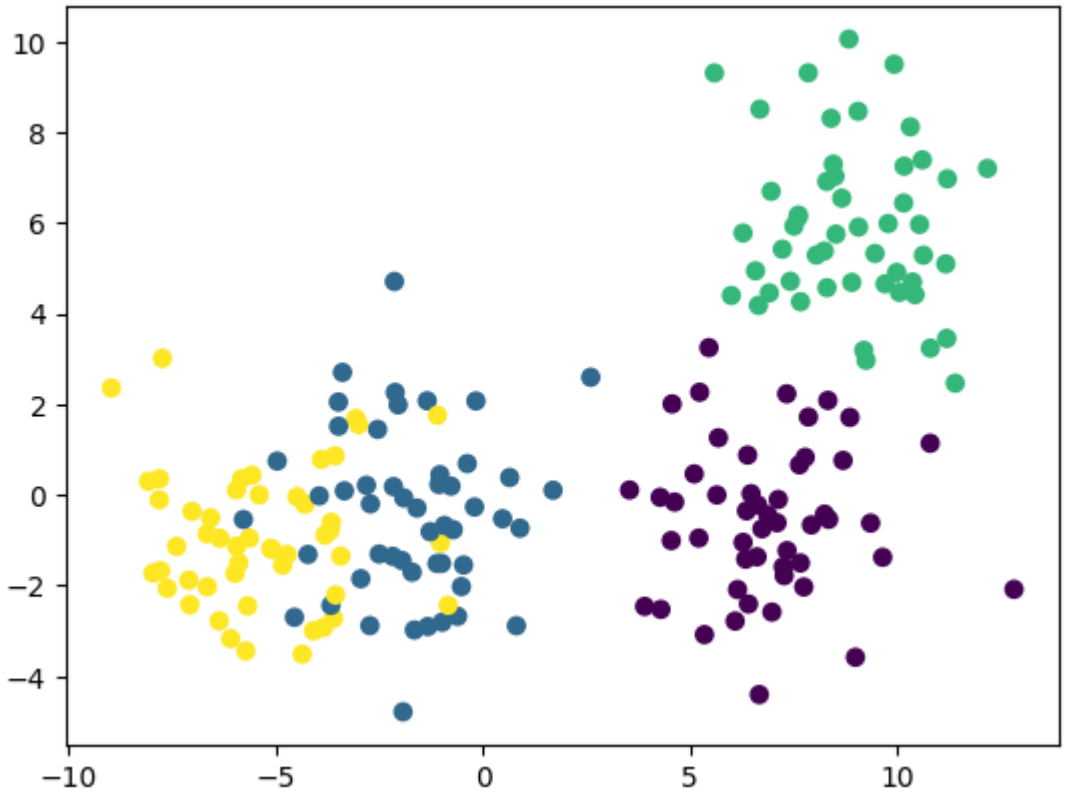
Out[37]:
```
<matplotlib.collections.PathCollection at 0x1c68595b4f0>
```



Modify the scatterplot to color code each data cluster.

In [39]:
```
# Need to reference the second element of object raw_data.
plt.scatter(
    raw_data[0][:, 0],
    raw_data[0][:, 1],
    c = raw_data[1]
)
```

Out[39]:
```
<matplotlib.collections.PathCollection at 0x1c6859d8cd0>
```



K-Means Model

In [40]:
```
from sklearn.cluster import KMeans
```

In [41]:
```
model = KMeans(n_clusters = 4).fit(raw_data[0])
```

Make Predictions

Predict which cluster each data point belongs to.

Access `labels_` attribute of the model to generate a `NumPy` array with predictions for each data point.

In [42]:
```
display(model.labels_)
```

```
array([3, 1, 3, 1, 2, 3, 0, 3, 1, 0, 3, 0, 2, 2, 1, 0, 1, 2, 2, 1, 3, 2,
       1, 3, 3, 3, 0, 2, 1, 0, 2, 1, 1, 3, 3, 3, 3, 0, 0, 3, 0, 2, 2, 2,
       0, 1, 0, 0, 2, 2, 1, 2, 2, 3, 2, 2, 1, 3, 3, 3, 3, 3, 1, 3, 2, 3,
       0, 2, 3, 0, 3, 3, 1, 0, 0, 2, 1, 2, 1, 3, 0, 0, 3, 2, 0, 2, 2, 0,
       3, 1, 0, 2, 2, 2, 2, 0, 1, 0, 1, 1, 2, 1, 3, 3, 3, 1, 3, 1, 0, 2,
       1, 0, 0, 2, 2, 0, 1, 0, 2, 0, 0, 1, 1, 0, 2, 0, 1, 1, 3, 3, 2, 2,
       0, 2, 2, 2, 2, 2, 2, 2, 2, 0, 2, 0, 3, 1, 3, 1, 0, 1, 3, 0, 3, 0,
       2, 0, 0, 1, 3, 2, 3, 1, 3, 0, 3, 0, 3, 1, 1, 0, 3, 1, 1, 0, 1, 1,
       2, 3, 0, 0, 0, 1, 1, 1, 3, 1, 3, 0, 1, 1, 2, 1, 0, 3, 3, 2, 3, 1,
       0, 1])
```

Access `cluster_centers_` attribute from the object model to generate a two-dimensional

`NumPy` array containing the coordinates of each cluster's center.

In [43]:
```
display(model.cluster_centers_)
```

```
array([[ 8.88625359,  5.90743421],
       [-5.63718384, -1.09687535],
       [-1.40906452, -0.24292299],
       [ 6.85109188, -0.53546267]])
```

Visualize the Clusters

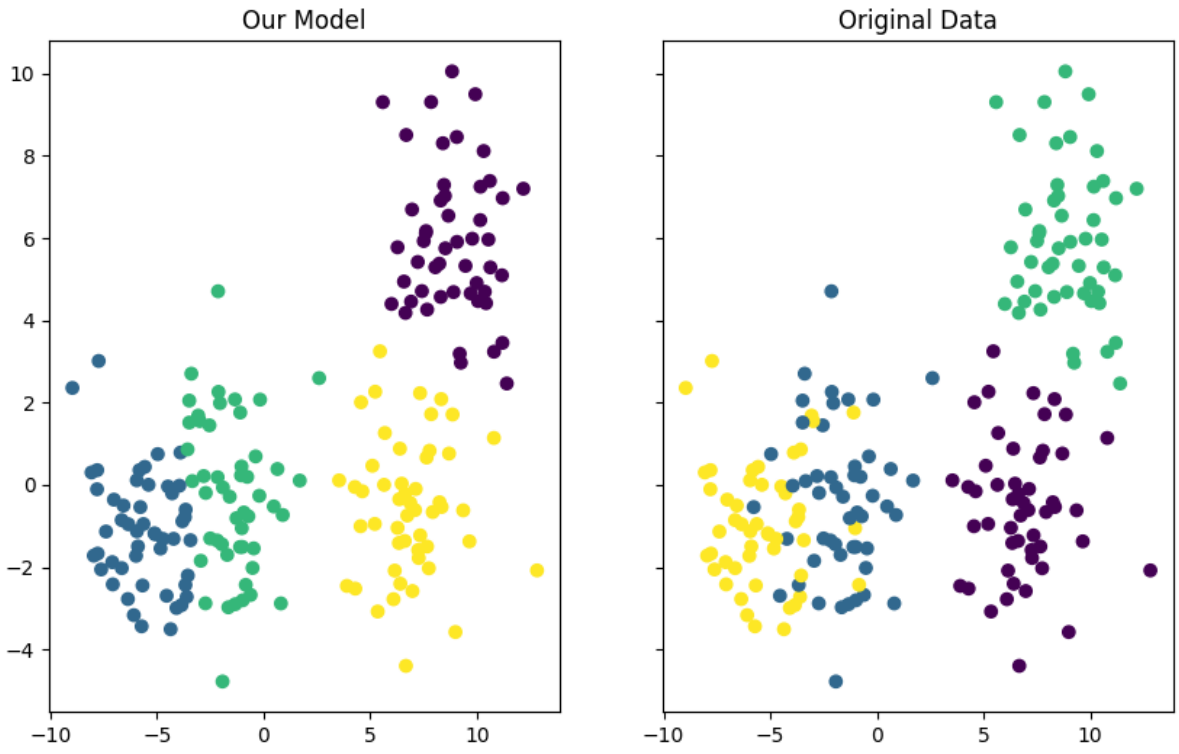
Visualize the accuracy of the **K-Means Clustering** Model by generating two different plots side-by-side.

In [44]:
```
# Note that the coloring between the two plots may be different.
f, (ax1, ax2) = plt.subplots(1, 2, sharey = True, figsize = (10, 6))

ax1.set_title('Our Model')
ax1.scatter(raw_data[0][:, 0], raw_data[0][:, 1], c = model.labels_)

ax2.set_title('Original Data')
ax2.scatter(raw_data[0][:, 0], raw_data[0][:, 1], c = raw_data[1])
```

Out[44]:
```
<matplotlib.collections.PathCollection at 0x1c685e7fee0>
```



The plot on the left shows the clusters according to our Machine Learning Model. We will notice that the model wasn't perfect. The data points are occasionally misclassified, usually along a cluster's edge.

Note that when measuring the predictive accuracy of a **K-Means Clustering** Model,

practitioners often don't know the clusters in advance. This is due to the fact that the **K-Means Clustering** Machine Learning Algorithm is used to find patterns that aren't obvious in a Dataset (i.e. Unsupervised Learning).