

In [1]:

```
#unsupervised learning
import numpy as np
import matplotlib.pyplot as plt
```

In [2]:

```
data_train = np.loadtxt('/home/bhavy/Dropbox/7th-semester/courses/ML/Assignments/A
```

In [3]:

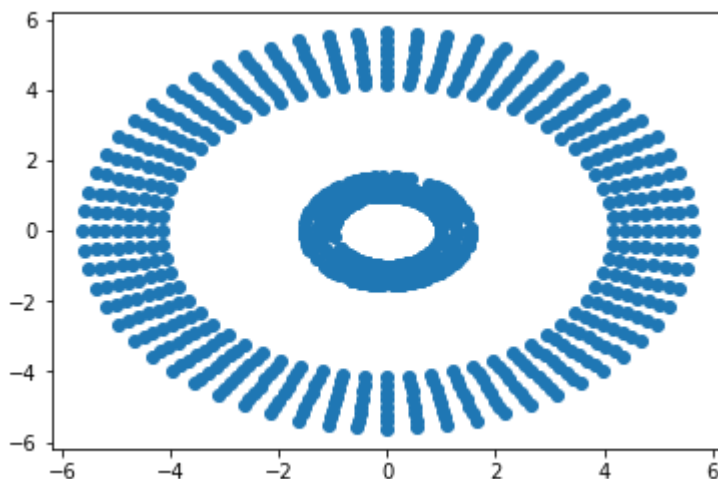
```
N = len(data_train)
```

In [4]:

```
x1, x2 = data_train[:, 0] , data_train[:, 1]
```

In [5]:

```
#data not linearly separable so cannot apply k means directly, so first transform t
plt.plot(x1, x2, 'o')
plt.savefig('original-plot-kmeans.png')
```



In [6]:

```
x3 = (np.sqrt(x1**2 + x2**2) <= 3).astype(int)
x3[x3 == 0] = x3[x3 == 0] - 10
x3[x3 == 1] = x3[x3 == 1] + 10
```

In [7]:

```
X = np.transpose(np.array([x1, x2, x3]))
#initial cluster means
mu0, mu1 = X[0], X[1]
#label
y = np.zeros(N) - 1
```

In [8]:

```
p = np.diag(np.dot(X-mu0, np.transpose(X-mu0)))
q = np.diag(np.dot(X-mu1, np.transpose(X-mu1)))
y = (p>q).astype(int)
mu0 = np.sum( X[y == 0] , axis = 0)/len(X[y==0])
mu1 = np.sum( X[y == 1] , axis = 0)/len(X[y==1])
print(len(X[y==0]))
```

500

In [9]:

```
plt.plot(X[y==0][:, 0], X[y==0][:, 1], 'go')
plt.plot(X[y==1][:, 0], X[y==1][:, 1], 'ro')
plt.savefig('k-means-hand-crafted.png')
```

