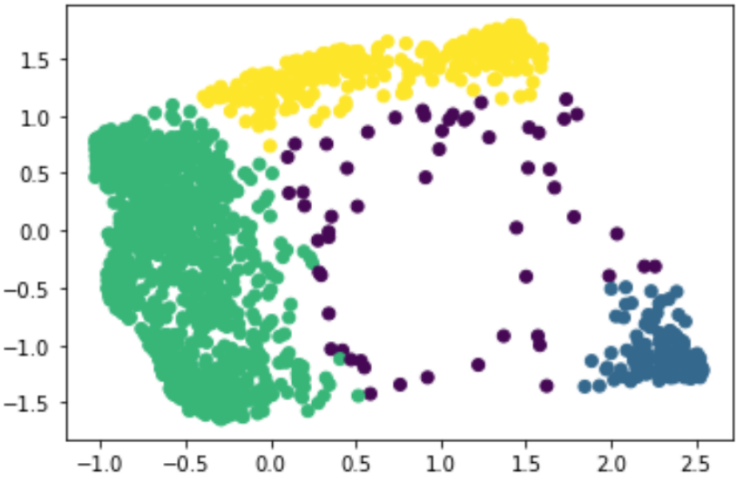
INFS4203 ASSIGNMENT 3

Student Number: 45360714
By Michael Delmastro

1. The DBSCAN algorithm is a suitable algorithm for clustering for this dataset. This algorithm is a method that uses the density of points to create clusters. An advantage to DBSCAN is that the number of clusters does not need to be specified. It assumes that there are two dense regions seperated by one sparse one and consequently is great at modelling spherical collections of points. By using this algorithm on the dataset, three clear clusters of different sizes are visible with similar densities and noise points are also shown in a different shade:

   This algorithm is more approriate than the kmeans algorithm for this dataset since the k-means algoirthm aims to create clusters that are the same size and does not account for how the data is scattered. Subsequently, the k-means is not concerned with the density of different parts of the dataset and is not great for modelling non-globular structures.



2. The step by step process below demonstrates how the DBSCAN algorithm works where when $print(y\_pred)$ is called, -1 denotes a noisy point, 0, 1 and 2 specify clusters.

   With the DBSCAN function call, eps is the maximum distance between two samples for one to be considered in the neighbourhood of the other. The Eps value (Eps = 0.2) was chosen after viewing and determining what is the maximum distance between two samples should be in the dataset. This is used so that all points within the esp circle of a core point can be used to form clusters. Consequently, points were considered in the same cluster if there distance is less than or equal to 0.2. This seems quite a reliable choice by viewing the graph with the noise points and three obvious defined clusters.

The $min\_samples = 10$ variable is the number of samples in a neighbourhood for a point to be considered as core point including the core point. Perhaps, this minimum number could have been adjusted to be a greater number due to the number of data points provided. However, this allows for more possible clusters to be identified and in a dataset of this size we are not sure about how many clusters could be present so this is a reasonable number for minimum samples to find core points. Therefore, these parameters were chosen to help form clusters with the DBSCAN algorithm.

```
import numpy as np
import matplotlib.pyplot as plt
from sklearn.datasets import make_circles
from sklearn.preprocessing import StandardScaler
from sklearn.cluster import DBSCAN
from pandas import DataFrame

1. Import the necessary packages we use above
2. Load the data from csv using pd.read_csv(.)
3. Normalise the data using X = data[[x, y]].values
4. Predict using DBSCAN with Eps = 0.2 and minPts = 10 using:
y_pred = DBSCAN(eps=0.2, min_samples=10).fit_predict(X)
print(y_pred)

5. Plot the results using:
plt.scatter(X[:,0], X[:,1], c=y_pred)
print(Number of clusters: {}.format(len(set(y_pred[np.where(y_pred !=
    -1)]))))
```