In [13]:

```
import numpy as np
import pandas as pd

from scipy.cluster.hierarchy import dendrogram, linkage
from matplotlib import pyplot as plt
from sklearn.preprocessing import StandardScaler, normalize

from scipy.cluster.hierarchy import single, cophenet
from scipy.spatial.distance import pdist, squareform

from sklearn.cluster import AgglomerativeClustering
from sklearn.metrics import silhouette_score
```

In [7]:

```
df = pd.read_csv('GPS-Dataset1.csv', header=None)
df = df.iloc[:-1,:2]
df = normalize(df)
df
```

Out[7]:

```
array([[0.99014976, 0.14001231],
       [0.99019221, 0.13971177],
       [0.99018302, 0.13977692],
       ...,
       [0.98692908, 0.16115518],
       [0.99449702, 0.10476487],
       [0.98673464, 0.16234144]])
```

In [12]:

```
c1,coph=cophenet(linkage(df, 'single'),pdist(df))
c2,coph=cophenet(linkage(df, 'complete'),pdist(df))
c3,coph=cophenet(linkage(df, 'average'),pdist(df))
c4,coph=cophenet(linkage(df, 'weighted'),pdist(df))
c5,coph=cophenet(linkage(df, 'centroid'),pdist(df))
linkage=pd.DataFrame({"Linkage":['Single','Complete','Average','Weighted','Centroid'],
                      "Cophenet Coeff":[c1,c2,c3,c4,c5]})
linkage
```

Out[12]:

	Linkage	Cophenet Coeff
0	Single	0.802681
1	Complete	0.840010
2	Average	0.860798
3	Weighted	0.848959
4	Centroid	0.860798

In [15]:

```
print('Average linkage is the most optimal')
```

```
Average linkage is the most optimal
```

In [9]:

```
plt.figure(figsize=(5, 5))
dendrogram(linkage(df, 'average'))
plt.title('Hierarchical Clustering Dendrogram (Average)', fontsize=20)
plt.show()
```

In [17]:

```
db_index=[]
for i in range(2,8):
    db_index.append(silhouette_score(df, AgglomerativeClustering(n_clusters=i, linkage='
average', affinity='euclidean').fit(df).labels_))
db_index = pd.DataFrame({'K Values':['2','3','4','5','6','7'],
                         'Silhouette index':db_index})
print(db_index)
```

```
K Values  Silhouette index
0        2          0.733312
1        3          0.830027
2        4          0.890782
3        5          0.901664
4        6          0.852721
5        7          0.812360
```

In [18]:

```
print('Optimal number of clusters is 3')
```

```
Optimal number of clusters is 3
```

In [24]:

```
scaler = StandardScaler()
X_scaled = scaler.fit_transform(df)
X_normalized = normalize(X_scaled)
X_normalized = pd.DataFrame(X_normalized)
ac2 = AgglomerativeClustering(n_clusters=2,linkage='average',affinity='euclidean')
plt.figure(figsize =(10, 10))
plt.scatter(X_normalized[0], X_normalized[1],
           c = ac2.fit_predict(X_normalized), cmap ='rainbow')
```

Out[24]:

```
<matplotlib.collections.PathCollection at 0x2218c5578d0>
```

In [ ]: