In [1]:

```
from IPython.display import Image
Image(filename='logo.PNG', height=340, width=900)
```

Out[1]:

In [2]:

```
# Importing Libraries

import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
```

In [3]:

```
# Import Dataset
df = pd.read_csv('creditcard.csv')
```

In [4]:

```
df.head()
```

Out[4]:

	Cust ID	Gender	Age	Monthly Income in 1000s	CreditScore (1-100)
0	1	Male	19	15	39
1	2	Male	21	15	81

	Cust ID	Gender	Age	Monthly Income in 1000s	CreditScore (1-100)
2	3	Female	20	16	6
3	4	Female	23	16	77
4	5	Female	31	17	40

In [5]:
```
df.tail()
```

Out[5]:

	Cust ID	Gender	Age	Monthly Income in 1000s	CreditScore (1-100)
195	196	Female	35	120	79
196	197	Female	45	126	28
197	198	Male	32	126	74
198	199	Male	32	137	18
199	200	Male	30	137	83

In [6]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 200 entries, 0 to 199
Data columns (total 5 columns):
Cust ID                    200 non-null int64
Gender                     200 non-null object
Age                        200 non-null int64
Monthly Income in 1000s    200 non-null int64
CreditScore (1-100)        200 non-null int64
dtypes: int64(4), object(1)
memory usage: 7.9+ KB
```

In [7]:
```
df.describe()
```

Out[7]:

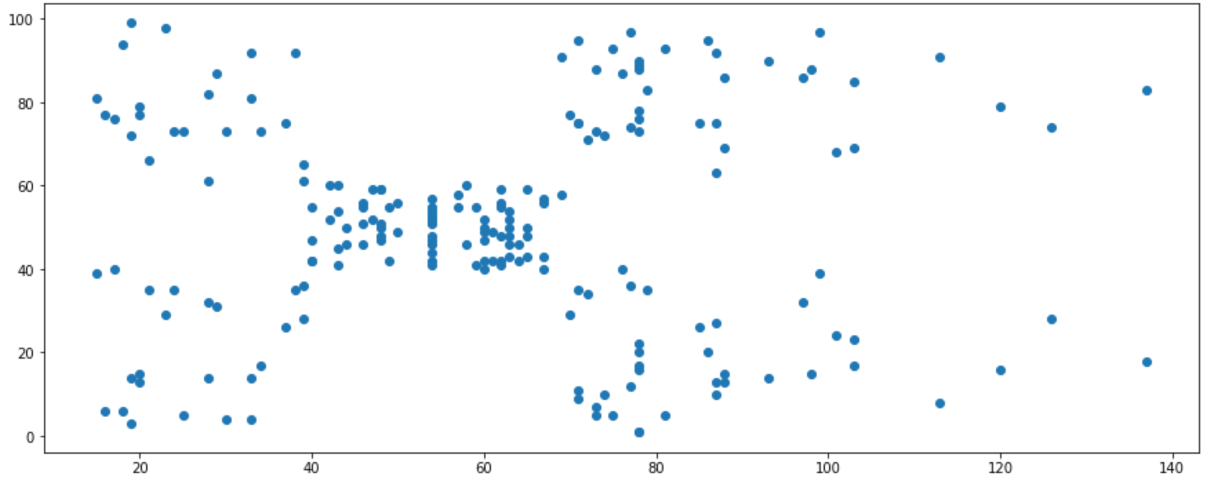
	Cust ID	Age	Monthly Income in 1000s	CreditScore (1-100)
count	200.000000	200.000000	200.000000	200.000000

	Cust ID	Age	Monthly Income in 1000s	CreditScore (1-100)
mean	100.500000	38.850000	60.560000	50.200000
std	57.879185	13.969007	26.264721	25.823522
min	1.000000	18.000000	15.000000	1.000000
25%	50.750000	28.750000	41.500000	34.750000
50%	100.500000	36.000000	61.500000	50.000000
75%	150.250000	49.000000	78.000000	73.000000
max	200.000000	70.000000	137.000000	99.000000

```
In [8]: X = df.iloc[:,[3,4]].values
```

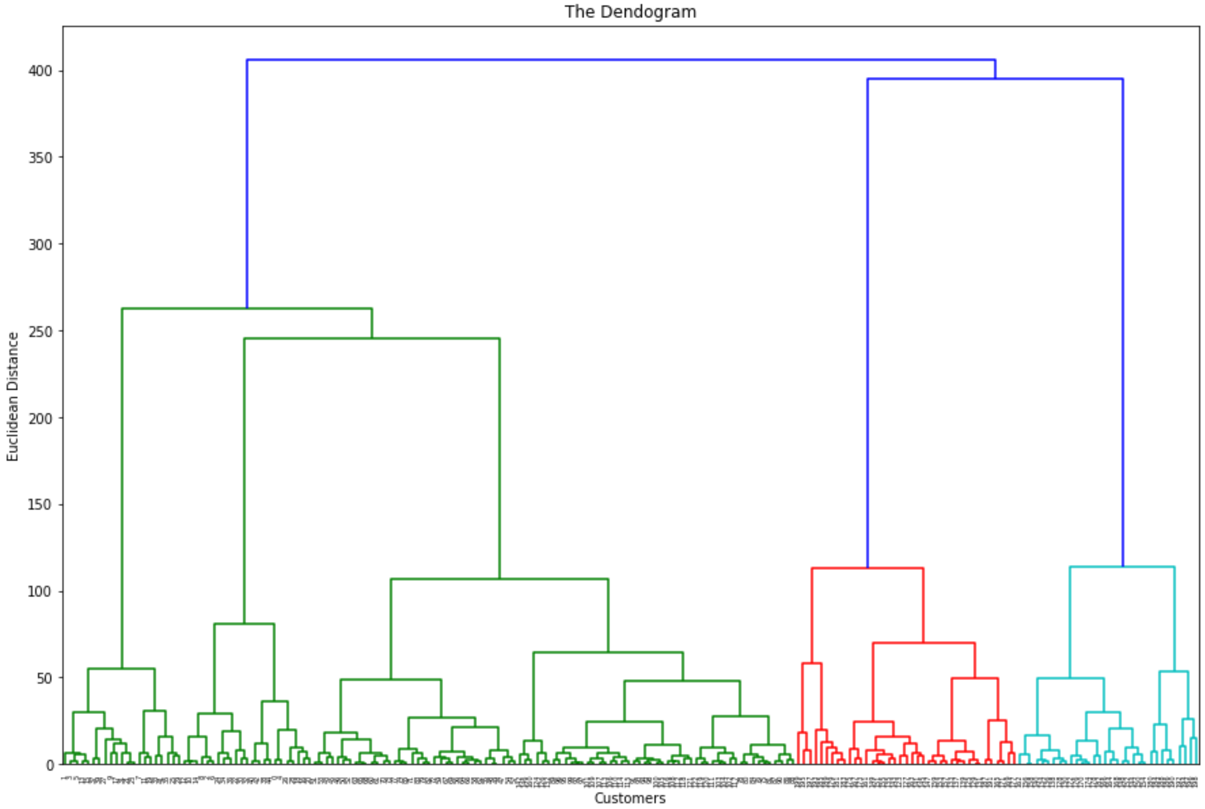
```
In [9]: # Initial View of the Data
        plt.figure(figsize=(15,6))
        plt.scatter(X[:,0], X[:,1])
```

```
Out[9]: <matplotlib.collections.PathCollection at 0x1e49a6e7e88>
```



```
In [10]: # Finding the optimum clusters using the DENDOGRAMS
         import scipy.cluster.hierarchy as sch
```

```
plt.figure(figsize = (15,10))
dendogram = sch.dendrogram(sch.linkage(X, method='ward'))
plt.title('The Dendogram')
plt.xlabel('Customers')
plt.ylabel('Euclidean Distance')
plt.show()
```



In [11]:

```
# Fitting the model
from sklearn.cluster import AgglomerativeClustering
hcluster = AgglomerativeClustering(n_clusters = 5, affinity = 'euclidea
n', linkage = 'ward')
y_clustering = hcluster.fit_predict(X)
```

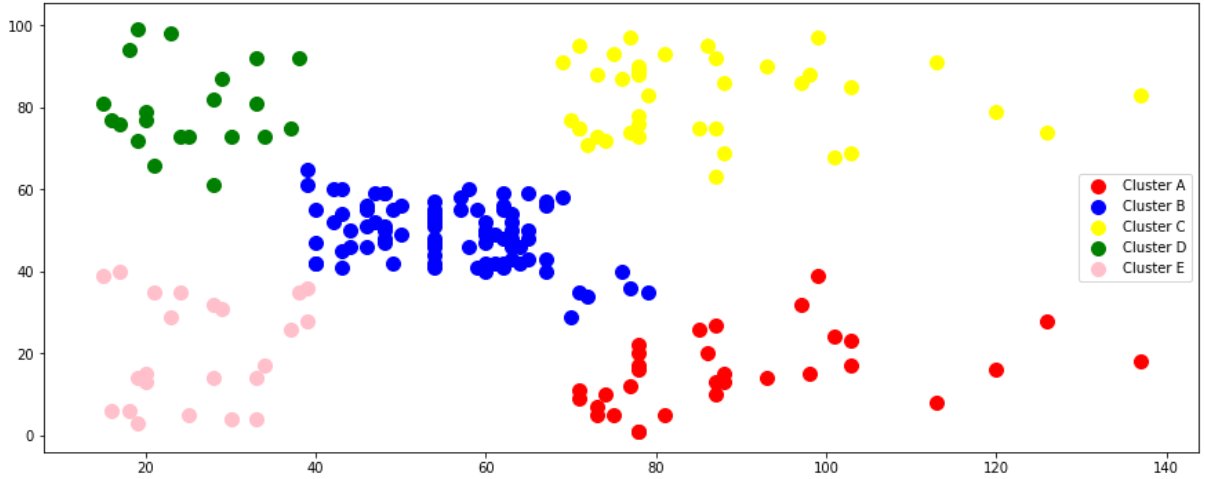
In [12]:

```
y_clustering
```

```
Out[12]: array([4, 3, 4, 3, 4, 3, 4, 3, 4, 3, 4, 3, 4, 3, 4, 3, 4, 3, 4, 3, 4,
3,
       4, 3, 4, 3, 4, 3, 4, 3, 4, 3, 4, 3, 4, 3, 4, 3, 4, 3, 4, 3, 4,
1,
       4, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 1, 2, 1, 2, 0, 2, 0,
2,
       1, 2, 0, 2, 0, 2, 0, 2, 0, 2, 1, 2, 0, 2, 1, 2, 0, 2, 0, 2, 0,
2,
       0, 2, 0, 2, 0, 2, 1, 2, 0, 2, 0, 2, 0, 2, 0, 2, 0, 2, 0, 2, 0,
2,
       0, 2, 0, 2, 0, 2, 0, 2, 0, 2, 0, 2, 0, 2, 0, 2, 0, 2, 0, 2, 0,
2,
       0, 2], dtype=int64)
```

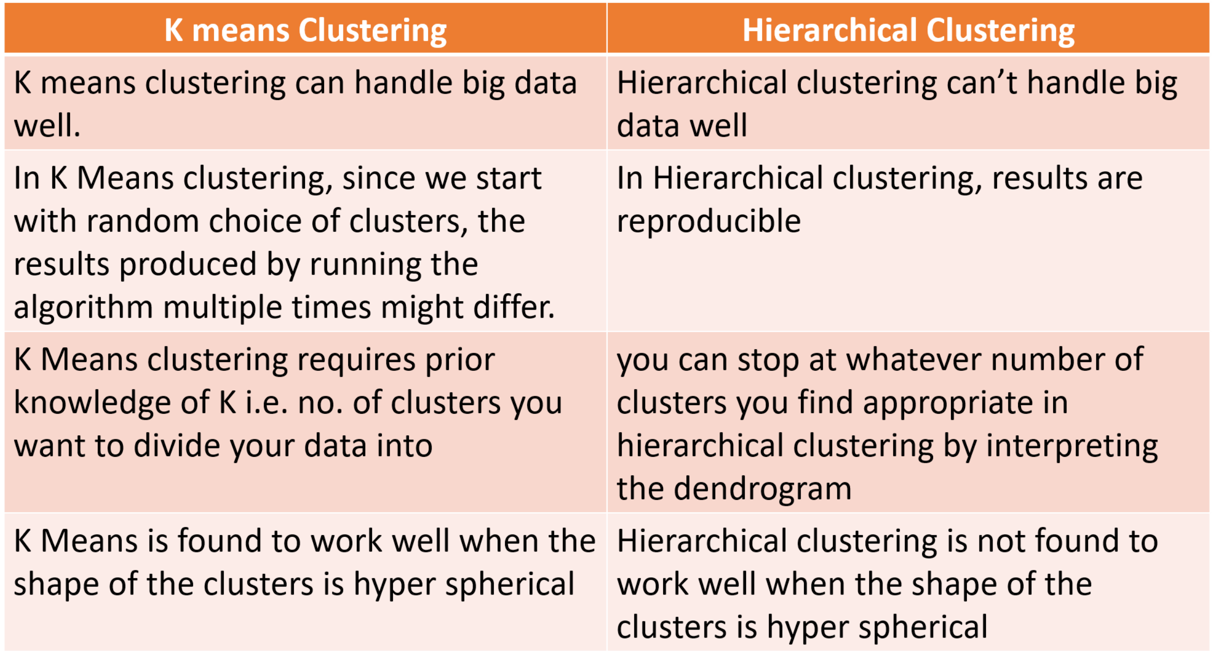
```
In [13]: # Visualizing Results
plt.figure(figsize=(15,6))
plt.scatter(X[y_clustering==0, 0], X[y_clustering==0, 1], s=100, c='re
d', label = 'Cluster A')
plt.scatter(X[y_clustering==1, 0], X[y_clustering==1, 1], s=100, c='blu
e', label = 'Cluster B')
plt.scatter(X[y_clustering==2, 0], X[y_clustering==2, 1], s=100, c='yel
low', label = 'Cluster C')
plt.scatter(X[y_clustering==3, 0], X[y_clustering==3, 1], s=100, c='gre
en', label = 'Cluster D')
plt.scatter(X[y_clustering==4, 0], X[y_clustering==4, 1], s=100, c='pin
k', label = 'Cluster E')
plt.legend()
```

```
Out[13]: <matplotlib.legend.Legend at 0x1e49d5c37c8>
```



In [14]:
```
from IPython.display import Image
Image(filename='Difference.PNG', height=340, width=900)
```

Out[14]:



K means Clustering	Hierarchical Clustering
K means clustering can handle big data well.	Hierarchical clustering can’t handle big data well
In K Means clustering, since we start with random choice of clusters, the results produced by running the algorithm multiple times might differ.	In Hierarchical clustering, results are reproducible
K Means clustering requires prior knowledge of K i.e. no. of clusters you want to divide your data into	you can stop at whatever number of clusters you find appropriate in hierarchical clustering by interpreting the dendrogram
K Means is found to work well when the shape of the clusters is hyper spherical	Hierarchical clustering is not found to work well when the shape of the clusters is hyper spherical

In [ ]: