Digital Assignment 5

Course: Data Structure & Algorithms Lab

Course Code: PMDS605P

Name: Soumyadeep Ganguly

Reg No: 24MDT0082

In [21]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
```

In [3]:
```
df = pd.read_csv("Credit_Card_Applications.csv")
df.head()
```

Out[3]:

	CustomerID	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11	A12	A13	A14	Class
0	15776156	1	22.08	11.46	2	4	4	1.585	0	0	0	1	2	100	1213	0
1	15739548	0	22.67	7.00	2	8	4	0.165	0	0	0	0	2	160	1	0
2	15662854	0	29.58	1.75	1	4	4	1.250	0	0	0	1	2	280	1	0
3	15687688	0	21.67	11.50	1	5	3	0.000	1	1	11	1	2	0	1	1
4	15715750	1	20.17	8.17	2	6	4	1.960	1	1	14	0	2	60	159	1

In [7]:
```
df.columns
```

Out[7]:
```
Index(['CustomerID', 'A1', 'A2', 'A3', 'A4', 'A5', 'A6', 'A7', 'A8', 'A9',
       'A10', 'A11', 'A12', 'A13', 'A14', 'Class'],
      dtype='object')
```

In [8]:
```
df.columns.value_counts().sum()
```

Out[8]:
```
np.int64(16)
```

In [9]:
```
df.isnull().sum()
```

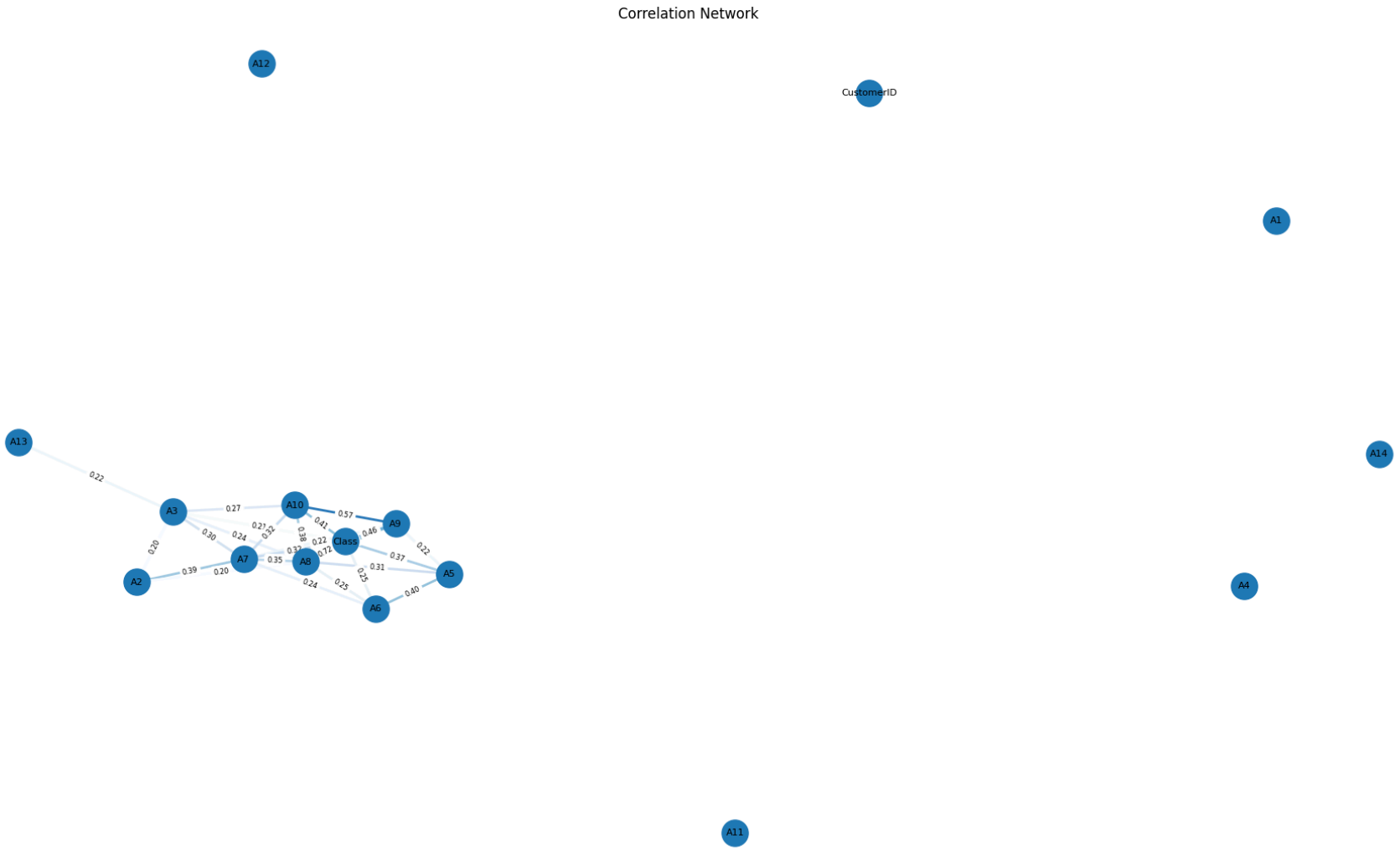
Out[9]:
```
CustomerID    0
A1            0
A2            0
A3            0
A4            0
A5            0
A6            0
A7            0
A8            0
A9            0
A10           0
A11           0
A12           0
A13           0
A14           0
Class         0
dtype: int64
```

In [13]:
```
import networkx as nx
G = nx.Graph()

corr_matrix = df.corr()
for col in corr_matrix.columns:
    G.add_node(col)

for i, col1 in enumerate(corr_matrix.columns):
    for j, col2 in enumerate(corr_matrix.columns):
        if i < j:
            weight = abs(corr_matrix.iloc[i, j])
```

```
            if weight > 0.2:
                G.add_edge(col1, col2, weight=weight)
```

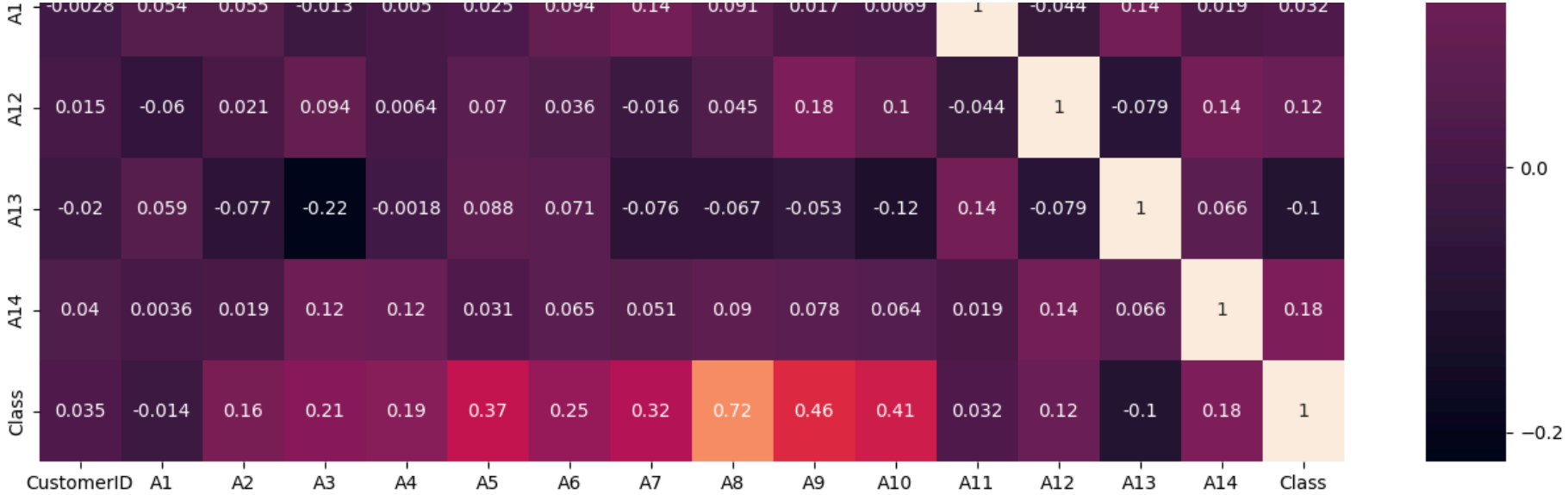
Plotting Correlation Network:

In [19]:
```
plt.figure(figsize=(18, 10))
pos = nx.spring_layout(G)
edges = G.edges(data=True)
edge_weights = [d['weight'] for (u, v, d) in edges]

nx.draw(G, pos, with_labels=True, edge_color=edge_weights, edge_cmap=plt.cm.Blues, width=2, font_size=8, node_size=500)
nx.draw_networkx_edge_labels(G, pos, edge_labels={(u, v): f'{d["weight"]:.2f}' for u, v, d in edges}, font_size=6)
plt.title("Correlation Network")
plt.show()
```



In [25]:
```
import seaborn as sns
plt.figure(figsize=(16, 16))
sns.heatmap(corr_matrix, annot=True)
plt.title("Correlation Plot")
plt.show()
```


Correlation Plot
CustomerID 1 -0.018 0.0015 0.048 0.072 -0.053 -0.057 0.055 0.041 0.02 -0.0069 -0.0028 0.015 -0.02 0.04 0.035
A1 -0.018 1 0.037 -0.028 -0.064 -0.0053 0.053 0.097 0.00092 -0.063 -0.0084 0.054 -0.06 0.059 0.0036 -0.014
A2 0.0015 0.037 1 0.2 0.09 -0.094 0.029 0.39 0.2 0.084 0.19 0.055 0.021 -0.077 0.019 0.16
A3 0.048 -0.028 0.2 1 0.092 0.028 0.095 0.3 0.24 0.17 0.27 -0.013 0.094 -0.22 0.12 0.21
A4 0.072 -0.064 0.09 0.092 1 0.089 0.047 0.081 0.13 0.16 0.11 0.005 0.0064 -0.0018 0.12 0.19
A5 -0.053 -0.0053 -0.094 0.028 0.089 1 0.4 0.14 0.31 0.22 0.15 0.025 0.07 0.088 0.031 0.37
A6 -0.057 0.053 0.029 0.095 0.047 0.4 1 0.24 0.25 0.08 0.099 0.094 0.036 0.071 0.065 0.25
A7 0.055 0.097 0.39 0.3 0.081 0.14 0.24 1 0.35 0.22 0.32 0.14 -0.016 -0.076 0.051 0.32
A8 0.041 0.00092 0.2 0.24 0.13 0.31 0.25 0.35 1 0.43 0.38 0.091 0.045 -0.067 0.09 0.72
A9 0.02 -0.063 0.084 0.17 0.16 0.22 0.08 0.22 0.43 1 0.57 0.017 0.18 -0.053 0.078 0.46
A10 -0.0069 -0.0084 0.19 0.27 0.11 0.15 0.099 0.32 0.38 0.57 1 0.0069 0.1 -0.12 0.064 0.41
1.0
0.8
0.6
0.4
0.2



Adjacency matrix

In [20]:
```
adj_matrix = nx.to_numpy_array(G)

eigenvalues = np.linalg.eigvals(adj_matrix)

print("Adjacency Matrix:")
print(adj_matrix)
print("\nEigenvalues:")
print(eigenvalues)
```

```
Adjacency Matrix:
[[0.         0.         0.         0.         0.         0.
  0.         0.         0.         0.         0.         0.
  0.         0.         0.         0.        ]
 [0.         0.         0.         0.         0.         0.
  0.         0.         0.         0.         0.         0.
  0.         0.         0.         0.        ]
 [0.         0.         0.         0.20131524 0.         0.
  0.         0.39278787 0.20434226 0.         0.         0.
  0.         0.         0.         0.        ]
 [0.         0.         0.20131524 0.         0.         0.
  0.         0.29890156 0.24431672 0.         0.27120674 0.
  0.         0.22234629 0.         0.20629374]
 [0.         0.         0.         0.         0.         0.
  0.         0.         0.         0.         0.         0.
  0.         0.         0.         0.        ]
 [0.         0.         0.         0.         0.         0.
  0.40228376 0.         0.31183258 0.2229572  0.         0.
  0.         0.         0.         0.37371161]
 [0.         0.         0.         0.         0.         0.40228376
  0.         0.24039951 0.24619305 0.         0.         0.
  0.         0.         0.         0.24656749]
 [0.         0.         0.39278787 0.29890156 0.         0.
  0.24039951 0.         0.34568876 0.22298177 0.32232967 0.
  0.         0.         0.         0.32247536]
 [0.         0.         0.20434226 0.24431672 0.         0.31183258
  0.24619305 0.34568876 0.         0.43203236 0.37953196 0.
  0.         0.         0.         0.72040682]
 [0.         0.         0.         0.         0.         0.2229572
  0.         0.22298177 0.43203236 0.         0.57149811 0.
  0.         0.         0.         0.45830133]
 [0.         0.         0.         0.27120674 0.         0.
  0.         0.32232967 0.37953196 0.57149811 0.         0.
  0.         0.         0.         0.40641001]
 [0.         0.         0.         0.         0.         0.
  0.         0.         0.         0.         0.         0.
  0.         0.         0.         0.        ]
 [0.         0.         0.         0.         0.         0.
  0.         0.         0.         0.         0.         0.
  0.         0.         0.         0.        ]
 [0.         0.         0.         0.22234629 0.         0.
```

```
  0.         0.         0.         0.         0.         0.
  0.         0.         0.         0.        ]
 [0.         0.         0.         0.         0.         0.
  0.         0.         0.         0.         0.         0.
  0.         0.         0.         0.        ]
 [0.         0.         0.         0.20629374 0.         0.37371161
  0.24656749 0.32247536 0.72040682 0.45830133 0.40641001 0.
  0.         0.         0.         0.        ]]

Eigenvalues:
[ 2.03327667  0.5004629   0.32934294  0.09980836 -0.76644    -0.69433576
 -0.5844269  -0.41833539 -0.22821825 -0.27113456  0.          0.
  0.          0.          0.          0.        ]
```

Analysis of Eigenvalues:

- The largest eigenvalue represents the most significant connection in the dataset.
- The spread of eigenvalues indicates how interconnected the attributes are.
- Higher eigenvalues suggest stronger correlations among certain attributes.
- The number of zero eigenvalues indicates disconnected components in the dataset.