Report

1) Data Collection and Formatting:

```
with open('data.txt') as f:
    lines=f.readlines()

A = []
B = []
C = []

for line in lines:
    Temp = line.split()
    A.append(Temp[0])
    B.append(Temp[1])
    C.append(Temp[2])
```

```
print(len(A))
print(len(B))
print(len(C))

C = list(map(float, C))

maximum = max(C)
minimum = min(C)
weights = []

for c in C:
    normalized_weight = (c - minimum) / (maximum - minimum)
    weights.append(normalized_weight)
```

2) . Community Detection:

```
communities = list(nx.community.greedy_modularity_communities(G))
print(len(communities))
```

```
4
```

greedy_modularity_communities measure based on the density of edges. So strong connections node → one community

3) Identifying Key Businesses:

I use degree_centrality → to know the influence of a person and who done more transactions

Picking top 10 person who influence more

```
Key Businesses in each Community:
Community 1: [('388', 0.16304347826086957), ('65', 0.16304347826086957), ('301', 0.15217391304347827), ('398', 0.14130434782608695), ('128', 0.14130434782608695), ('162', 0.1358695652173913), ('342', 0.1358695652173913), ('187', 0.1358695652173913), ('437', 0.1358695652173913), ('180', 0.13586956521739133)]
Community 2: [('313', 0.1875), ('102', 0.17613636363636365), ('300', 0.17045454545454547), ('243', 0.1590909090909091), ('448', 0.14772727272727273), ('391', 0.14772727272727273), ('389', 0.14772727272727273), ('126', 0.14204545454545456), ('154', 0.14204545454545456), ('258', 0.14204545454545456)]
Community 3: [('446', 0.19469026548672566), ('350', 0.18584070796460178), ('44', 0.17699115044247787), ('273', 0.16814159292035397), ('123', 0.15929203539823009), ('224', 0.1592920353982301), ('28', 0.1592920353982301), ('262', 0.1592920353982301), ('34', 0.1504424778761062), ('196', 0.1504424778761062)]
Community 4: [('440', 0.43478260869565216), ('83', 0.34782608695652173), ('110', 0.2608695652173913), ('78', 0.2608695652173913), ('360', 0.2608695652173913), ('175', 0.2608695652173913), ('395', 0.21739130434782608), ('282', 0.21739130434782608), ('230', 0.21739130434782608), ('494', 0.21739130434782608)]
```

4) Anomaly Detection:
Threshold value

```
if normalized_weight > 0.81:
```

```
Total anomaly transactions: 328
Total transactions: 10000
Percentage of anomaly transactions: 3.28%
```