



Σημασιολογικός και Κοινωνικός Ιστός 2022-23

Απαλλακτική Εργασία

Νίκος Λιθαρής Π2019083

Contents

Ερώτημα Α)	3
Κώδικας	3
Ερώτημα Β)	4
Αποτελέσματα	4
Υποερώτημα 1	4
Υποερώτημα 2	4
Κώδικας	4
Ερώτημα Γ)	6
Αποτελέσματα	6
Υποερώτημα 1	6
Υποερώτημα 2	6
Υποερώτημα 3	6
Υποερώτημα 4	7
Κώδικας	7

Ερώτημα Α)

Στο ερώτημα Α) έπρεπε να δείξουμε τη δομή του αντικειμένου που επιστρέφει η κλήση της συνάρτησης `verifyCredentials`. Η συνάρτηση επιστρέφει ένα `<class 'twitter.models.User'>` το οποίο είναι ένα αντικείμενο τύπου `json`.

Κώδικας

```
1  import twitter
2  import json
3
4  keys = json.load(open('data/json/keys.json'))
5
6  consumer_key = keys['api_key']
7  consumer_secret = keys['api_secret']
8  access_token_key = keys['access_token']
9  access_token_secret = keys['access_secret']
10 api = twitter.Api(consumer_key=consumer_key, consumer_secret=consumer_secret,
11                  access_token_key=access_token_key, access_token_secret=access_token_secret)
12
13 print(type(api.VerifyCredentials()))
14 json.dump(api.VerifyCredentials().__dict__, open(
15     'data/json/credentials.json', 'w'), indent=4)
```

Ερώτημα Β)

Στο ερώτημα Β) έπρεπε να δημιουργήσουμε ένα **Ego-network** βασισμένο στο αρχείο **ego.csv** που μας δόθηκε, και στη συνέχεια να υπολογίσουμε τα εξής:

1. Πόσοι κόμβοι και ακμές περιέχονται στο δίκτυο;
2. Βασικές μετρικές του γράφου:
 - Μέσος βαθμός
 - Χαρακτηριστικό μήκος μονοπατιού
 - Μέσος συντελεστής συσταδοποίησης.

Αποτελέσματα

Υποερώτημα 1

Κόμβοι: **580**

Ακμές: **587**

Υποερώτημα 2

Μέσος βαθμός: **2.024137931034483**

Χαρακτηριστικό μήκος μονοπατιού: **2.854553034363647**

Μέσος συντελεστής συσταδοποίησης: **0.0018448861455234646**

Κώδικας

```
1 import pandas as pd
2 import networkx as nx
3 import matplotlib.pyplot as plt
4
5 # Read the ego.csv file
6 df = pd.read_csv('data/ego.csv')
7
8 # Create graph
9 G = nx.Graph()
10
11 # Add nodes and edges to the graph
12 for _, row in df.iterrows():
13     source = row['source']
14     target = row['target']
15     G.add_edge(source, target)
16
17 # Calculate the number of nodes and edges
18 num_nodes = G.number_of_nodes()
19 num_edges = G.number_of_edges()
20
21 # Calculate graph metrics
```

```

22 degree = dict(G.degree())
23 mean_degree = sum(degree.values()) / num_nodes
24 characteristic_path_length = nx.average_shortest_path_length(G)
25 mean_clustering_coefficient = nx.average_clustering(G)
26
27 # Print the results
28 print("Number of nodes:", num_nodes)
29 print("Number of edges:", num_edges)
30 print("Mean degree:", mean_degree)
31 print("Characteristic path length:", characteristic_path_length)
32 print("Mean clustering coefficient:", mean_clustering_coefficient)
33
34 # Draw the network
35 nx.draw(G, node_size=10, node_color='red', edge_color='blue', with_labels=True)
36 plt.show()

```

Ερώτημα Γ)

Στο ερώτημα Γ) μας δόθηκε το αρχείο **users_friends.csv** με τα id κάποιων χρηστών και των φίλων τους. Για αυτούς του χρήστες έπρεπε να δημιουργηθεί ένα δίκτυο και να υπολογιστούν τα εξής:

1. Ο αριθμός των κόμβων και των ακμών.
2. Είναι όλοι οι χρήστες συνδεδεμένοι μεταξύ τους;
3. Βασικές μετρικές του γράφου:
 - Μέσος βαθμός
 - Χαρακτηριστικό μήκος μονοπατιού
 - Μέσος συντελεστής συσταδοποίησης
4. Ταξινόμηση των βαθμών και απεικόνιση τους σε διάγραμμα.

Αποτελέσματα

Υποερώτημα 1

Κόμβοι: **3237**

Ακμές: **3438**

Υποερώτημα 2

Όλοι οι χρήστες είναι συνδεδεμένοι μεταξύ τους

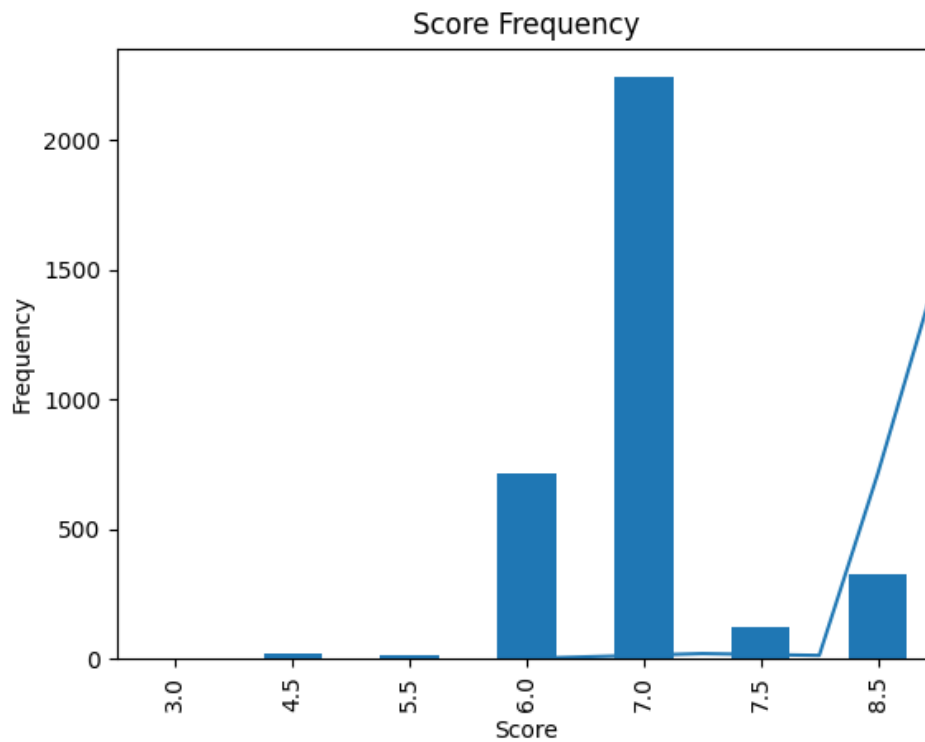
Υποερώτημα 3

Μέσος βαθμός: **2.1241890639481**

Χαρακτηριστικό μήκος μονοπατιού: **3.5304275006272117**

Μέσος συντελεστής συσταδοποίησης: **0.0**

Υποερώτημα 4



Από το διάγραμμα παρατηρούμε ότι οι περισσότεροι έχουν score ίσο με 7

Κώδικας

```
1 import pandas as pd
2 import networkx as nx
3 import matplotlib.pyplot as plt
4
5 # Read the users_friends.csv file
6 df = pd.read_csv('data/users_friends.csv')
7
8 # Create an empty graph
9 G = nx.Graph()
10
11 # Add edges to the graph
12 for _, row in df.iterrows():
13     source = row['id_a']
14     target = row['id_b']
15     G.add_edge(source, target)
16
17 # Calculate the number of nodes and edges
```

```

18 num_nodes = G.number_of_nodes()
19 num_edges = G.number_of_edges()
20
21 # Check if all users are connected
22 is_connected = nx.is_connected(G)
23
24 # Calculate graph metrics
25 mean_degree = sum(dict(G.degree()).values()) / num_nodes
26 characteristic_path_length = nx.average_shortest_path_length(G)
27 mean_clustering_coefficient = nx.average_clustering(G)
28
29 # Print the results
30 print("Number of nodes:", num_nodes)
31 print("Number of edges:", num_edges)
32 print("Are all users connected?", is_connected)
33 print("Mean degree:", mean_degree)
34 print("Characteristic path length:", characteristic_path_length)
35 print("Mean clustering coefficient:", mean_clustering_coefficient)
36
37 sorted_score = df['score'].value_counts().sort_index(ascending=True)
38
39 # Plot the score
40 sorted_score.plot(kind='bar')
41 plt.plot(sorted_score)
42 plt.xlabel('Score')
43 plt.ylabel('Frequency')
44 plt.title('Score Frequency')
45 plt.show()

```