A PROJECT REPORT ON

DBLP DATA ANALYSIS USING GRAPH CHARACTERISTICS: PROJECT 2

BY

ABHISHEK SHRINIVAS JOSHI

ANUSHKA CHANDRAKANT PAWAR

UNDER THE GUIDANCE OF PROF. SHARMA CHAKRAVARTHY

CONTENTS

INTRODUCTION

In this project, we learned and implemented Graphical Analysis. We learned to implement Data Analysis on real-world Datasets. Python is one of the most suitable languages for data analysis today. The analysis is performed using the programming language Python as it has a variety of libraries available. We used libraries like pandas, Networkx, matplotlib, and NumPy to analyze data, draw the required graphs, derive conclusions on the given dataset, and get meaningful results. Working on the given DBLP data set we analyzed and derived some useful information about the Authors, Papers, Publications, collaborations, etc.

The goal of this project is to perform an analysis based on modeling the given data as Graphs and to understand map analysis requirements in the real world. The Graphical analysis helps to understand the data better, it also helps to understand the patterns and correlations in the parameters in the data we have. Here, we have performed

DATASET

We have a DBLP Data Set. Which Consists of Authors, Papers, Publications, Conferences, etc. We analyzed the connections and correlations between the parameter in the given data set.

Size of Data Set: 4GB

WORK DISTRIBUTION

- Data pre-processing is done by both Abhishek Shrinivas Joshi and Anushka Chandrakant Pawar.
- The Known_Authors Graph and Paper_citation graph are done by Abhishek Shrinivas Joshi and The Author_conference graph is done by Anushka Chandrakant Pawar.
- Conclusion and Report are Made by both equally.

METHODOLOGY(Overall Status)

- As the data set given for analysis is very large, we used chunk size and loaded the chunks in one single Data frame to get one single data frame consisting of all the data we have.
- After fetching the data, we had to build graph1 (i.e Known_author graph). As we are supposed to eliminate the duplicate Authors, we created a list of authors of one single paper and created a complete graph, and then composed that across common nodes to a new graph with consists of all the previous graphs.
- For 2nd Graph, We created a Dictionary where the key is Paper ID and then the values are the References. Then we created a For loop to connect each reference to the Id of the Paper.
- For the 3rd Graph, we used a similar logic as Graph 2 (paper_citation graph) we created a Dictionary where the key is Author and the values are Conferences(venues). Then we created a For loop where we connected the Edges from the author to the conference to obtain a Graph.
- For Analysis 0 , we approached it by learning about Networkx, so that we can get graph Characteristics by implementing noworkx methods.
- For Analysis 1, We performed manual calculations for all the graph characteristics we have to get. We compared the manual and actual values which we got in analysis 0.
- For analysis 3a and 3b we took the authors and papers having the highest degree of centrality so that they are the top 10 authors and papers in the given dataset.
- We have created known/Authors1.csv where we stored our sample data.

ANALYSIS AND RESULT

Analysis 0:

- Graph 1 (Known_Author Graph)
  n = G.number_of_nodes()
  m = G.number_of_edges()
  density_g1 = nx.density(G)
  num_connected_comp = len(list(nx.connected_components(G)))
  degree_values = list(dict(G.degree()).values())
  min_degree = min(degree_values)
  max_degree = max(degree_values)
  avg_degree = sum(degree_values)/n
  # diameter = infinity

- Graph 2 (Paper_citation Graph)
  n = paper_citation.number_of_nodes()
  m = paper_citation.number_of_edges()
  density_g1 = nx.density(paper_citation)
  degree_values = list(dict(paper_citation.degree()).values())
  min_degree = min(degree_values)
  max_degree = max(degree_values)
  avg_degree = sum(degree_values)/n
  # diameter = infinity

- Graph 3 (Author_venue Graph)
  n = av.number_of_nodes()
  m = av.number_of_edges()
  density_g1 = nx.density(av)
  num_connected_comp = len(list(nx.connected_components(av)))
  degree_values = list(dict(av.degree()).values())
  min_degree = min(degree_values)
  max_degree = max(degree_values)
  avg_degree = sum(degree_values)/n
  # diameter = infinity

Analysis 1:

Actual Values Table

Graph	Number of Nodes	Number of Edges	Density	Number of Connected Components	Minimum Degree	Maximum Degree	Average Degree	Standard Deviation Degree	Diameter
G1	17	26	0.19	5	1	5	3.059	1.6	infinity
G2	41	36	0.022	NA	1	14	1.76	2.667	infinity
G3	22	17	0.074	5	1	6	1.5454	1.2331	infinity

Manual Calculations Table

Graph	Number of Nodes	Number of Edges	Density	Number of Connected Components	Minimum Degree	Maximum Degree	Average Degree	Standard Deviation Degree	Diameter
G1	17	26	0.19	5	1	5	3.0859	1.59	infinity
G2	41	36	0.0233	NA	1	14	1.7561	2.659	infinity
G3	22	17	0.0736	5	1	6	1.5454	1.2241	infinity

Comments:

- For Calculating Density for undirected Graph, Density= 2 *m/n(n-1), where m= number of edges and n= number of nodes .
- For directed Graph, Density= m/n(n-1)
- We do not have connected components in Graph 2. Hence there are no no. of connected components present in Graph 2.
- There is no shortest path present in the most distanced nodes, therefore the diameter for all the graphs is infinity.

Analysis 2:

Maximal group of Authors who are mutually connected,

[['Uriel Martinez-Hernandez',

'Hector Barron-Gonzalez',

'Mathew H. Evans',

'Giorgio Metta',

'Nathan F. Lepora',

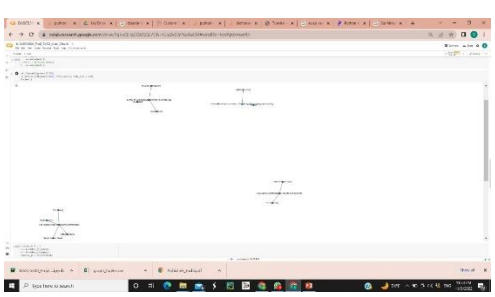
'Tony J. Prescott'],

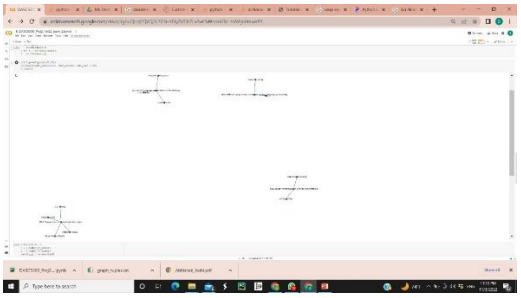
['Parikshit Yadav', 'Rajesh Kumar', 'Sanjib Kumar Panda', 'C.S. Chang'],

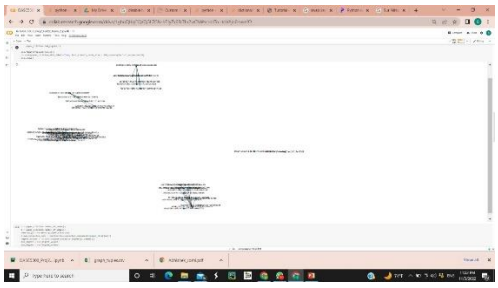
['Yi-Ching Ting', 'Tian-Sheuan Chang'],

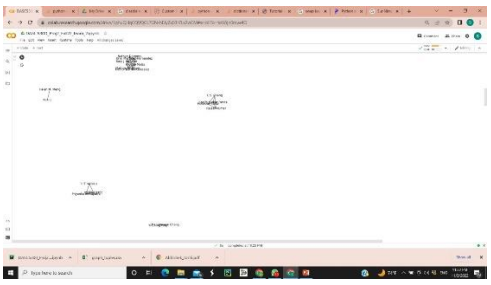
['Kun Li', 'Helen M. Meng'],

['H.T. Neisius', 'Priyanka Mohapatra', 'Izudin Dzafic']]









Analysis 3:

Analysis 3a:

10 papers that are cited most from two years (2012-2013) from the paper citation graph

Paper

f07b9c56-f13f-4e5f-a31e-b3b7c7a44509

735d071e-4dba-413b-9604-1e948e54772e

a5b89add-eda2-4a97-aa73-47a739c24f37

ce765697-3426-4c23-8806-4b246f204695

fdcd0f7d-0f21-43cf-aa93-073319144e2d

fdf8f78d-3a8f-4e3c-95c6-9b04a1e77f10

2e904d90-00e9-410d-9021-7549ade79991

3925ee21-0475-41ee-8ce2-b638da2e9b19

7750219a-459e-447e-86b2-6a66f2e412e3

7fa2dcc1-6510-4216-b45d-531407d3ac4b

Analysis 3b:

10 authors who have published most papers in your data set irrespective of the conferences.

Author

Yi-Ching Ting

Tian-Sheuan Chang

Izudin Dzafic

H.T. Neisius

Priyanka Mohapatra

Nathan F. Lepora

Uriel Martinez-Hernandez

Hector Barron-Gonzalez

Mathew H. Evans

Giorgio Metta

Tony J. Prescott

Parikshit Yadav

Rajesh Kumar

Sanjib Kumar Panda

C.S. Chang

Kun Li

Helen M. Meng

PROJECT TIMELINE:

12th October 2022- Started with reading and understanding the project and learning networkx for the project.

19th October 2022- Started with Data fetching and pre-processing

24th October 2022- Started creating the first Graph

28th October 2022- By the 28th, we completed building all the 3 graphs.

29th- 31st October 2022- We completed all 3 analyses.

31st October– 2nd November- We worked on corrections in our code.

3rd November- Worked on Project Report.

DIFFICULTIES FACED IN THE PROJECT:

- We had to eliminate duplicate authors and Graph 1. And this is where we faced difficulty. We were not able to eliminate the duplicate authors for many days of our project work.
  We solved the issue by creating a list each time of the Authors of one paper and created a complete graph and then we composed that graph into a new graph which consisted of the previous graph across common nodes.
- Next issue was similar to the first, where we had to eliminate the duplicate paper citations. For a long time, we were not able to eliminate it. But we applied the same logic as in the previous to solve this problem.
- In the initial stage of the project, we were not able to fetch a large amount of dataset, and even by using chunk size, were not able to fetch all the data as a single file. Hence to solve this issue, we used Chunk size and loaded the data in one single data frame, and created a data frame having all the data.

REFERENCES:

1. https://networkx.org/documentation/networkx-1.9/tutorial/tutorial.html
2. https://colab.research.google.com/drive/1ghuQHqCQVQGL7GN-hEJyZx03r7Lx2wCM?usp=sharing
3. https://www.analyticsvidhya.com/blog/2018/09/introduction-graph-theory-applications-python/
4. Project 2 Helper Slides