Social Network & Graph: A Quantitative Analysis

Abstract :- This report provides a thorough analysis of the mean, median, maximum, and minimum values of the datasets, along with graph implementation using MST (Minimum Spanning Tree) and relevant algorithms. The key algorithms employed include Prim's, Kruskal's, Dijkstra's, and Bellman-Ford algorithms. By understanding these algorithms and exploring the central tendency measures, we can gain deeper insights into comparing the three datasets and obtain valuable outcomes.

Introduction :-

A minimum spanning tree (MST) is a tree that connects all the vertices of a connected graph with the minimum possible total weight. MSTs are commonly utilized in design problems to address complex network-related challenges. To obtain the desired values, we employ appropriate algorithms that effectively compute the MST and assist in analyzing the data. These algorithms play a crucial role in finding optimal solutions and enabling efficient decision-making processes.

Datasets :-

i)Twitch :- These datasets used for node classification and transfer learning are Twitch user-user networks of gamers who stream in a certain language. Nodes are the users themselves and the links are mutual friendships between them. Vertex features are extracted based on the games played and liked, location and streaming habits. Datasets share the same set of node features, this makes transfer learning across networks possible. These social networks were collected in May 2018. The supervised task related to these networks is binary node classification - one has to predict whether a streamer uses explicit language.

ii)LastFm :- A social network of LastFM users which was collected from the public API in March 2020. Nodes are LastFM users from Asian countries and edges are mutual follower relationships between them. The vertex features are extracted based on the artists liked by the users. The task related to the graph is multinomial node classification - one has to predict the location of users. This target feature was derived from the country field for each user.

iii)Resistance :- The dynamic face-to-face interaction networks represent the interactions that happen during discussions between a group of participants playing the Resistance game. This dataset contains networks extracted from 62 games. Each game is played by 5-8 participants and lasts between 45--60 minutes. We extract dynamically evolving networks from the free-form discussions using the ICAF algorithm. The extracted networks are used to characterize and detect group deceptive behavior using the DeceptionRank algorithm.

Result :-

Central tendencies are determined by applying a specific algorithm. To start, we write the code for this algorithm. Additionally, we download a dataset from an available website to work with. After acquiring the dataset, we import a library that provides pre-defined functions for our analysis. By using a for loop, we process the text file of the dataset and execute the necessary coding to compute the mean, median, maximum, and minimum values.

For graph analysis, two algorithms, Prim's and Kruskal's, are utilized to find the Minimum Spanning Tree (MST). Prim's algorithm connects an arbitrary starting vertex to all other vertices through edges with the lowest weights, ensuring there are no cycles, thus yielding the MST. This approach can also be adapted to analyze graphs, although graphs can be more complex due to their larger number of nodes and edges.

Moreover, Dijkstra's algorithm is implemented to find the shortest path in a weighted graph. Given a weighted graph G and a source vertex S, the algorithm identifies the shortest path from S to any other vertex that needs to be visited. Similarly, the Bellman Ford algorithm serves the same purpose and allows the determination of the shortest path from a start vertex to all other vertices. One key advantage of the Bellman Ford algorithm is its ability to handle graphs with negative weights.

Conclusion :-

After obtaining the values of the mean, median, maximum, and minimum, we can draw insightful inferences from the analysis. By comparing the mean and median, we observe that the distribution is asymmetric in datasets, indicating a potential skewness. On the other hand, in graphs, the distribution appears to be symmetric. This discrepancy highlights the different characteristics of datasets and graphs in terms of their distribution patterns.
Furthermore, by comparing the maximum and minimum values, we deduce that datasets tend to have a wider spread of values, indicating greater variability. In contrast, graphs exhibit a more concentrated range of values, suggesting a higher level of consistency or coherence in the data.
Through these comparisons and observations, we can identify trends and patterns across different datasets. These trends and patterns provide valuable insights that can contribute to our understanding and decision-making processes. By recognizing and analyzing these variations, we can gain a deeper understanding of the data and extract meaningful information that can be beneficial in various domains.

References :-

S. Kumar, C.Bai, V.S. Subrahmanian, J. Leskovec. Deception Detection in Group Video Conversations using Dynamic Interaction Networks. 15th International AAAI Conference on Web and Social Media (ICWSM), 2021.

B. Rozemberczki and R. Sarkar. Characteristic Functions on Graphs: Birds of a Feather, from Statistical Descriptors to Parametric Models. 2020.

C.Bai, S. Kumar, J. Leskovec, M. Metzger, J.F. Nunamaker, V.S. Subrahmanian. Predicting Visual Focus of Attention in Multi-person Discussion Videos. International Joint Conference on Artificial Intelligence (IJCAI), 2019.

	Nodes	**Edges**
Twitch	4,385	37,304
LastFm	7,624	27,806
Resistance	451	31,26,993

***Note: all the datasets are undirected, and weight is assumed to be 1 for all the nodes.**

Graphs :-

SNAP DATASETS RESULTS :-

	Prims	**Kruskal**	**Bellman Ford**	**Dijkastra**
Mean	7202.333	7202.333	6	6
Median	4383	4383	5	5
Minimum	61	61	2	2
Maximum	7623	7623	11	11

MID SEM DATASETS RESULTS :-

	Prims	**Kruskal**	**Bellman Ford**	**Dijkastra**
Mean	11.333	11.333	6.33	6.33
Median	12	12	6	6
Minimum	10	10	4	4
Maximum	12	12	10	10

Inputs and Outputs of Codes (Lab-11)

Linked-List (1) :- Data structure for Students

Input	Output
Create a student details :-	
Enter name :- Ravi Enter roll number :- 51 Enter age :- 19 Enter branch :- DSAI	---------------------------------- Name | Roll | Age | Branch ---------------------------------- Ravi | 51 | 19 | DSAI
Insert new student details :-	
Enter name :- Rudra Enter roll number :- 101 Enter age :- 18 Enter branch :- CSE	---------------------------------- Name | Roll | Age | Branch ---------------------------------- Ravi | 51 | 19 | DSAI Rudra | 101 | 18 | CSE
Delete student details :-	
Enter the roll no. of the student to delete :- 101	---------------------------------- Name | Roll | Age | Branch ---------------------------------- Ravi | 51 | 19 | DSAI

Linked-List (2a) :- Concatenate two given list

Output	Input
Create first Linked-List :-	
Enter the number of elements in the first linked list :- Enter the element 1 :- Enter the element 2 :- Enter the element 3 :-	3 34 67 90
Create second Linked-List :-	
Enter the number of elements in the second linked list :- Enter the element 1 :- Enter the element 2 :-	2 13 57
Concatenated List :-	
List 1: 34 -> 67 -> 90 -> NULL List 2: 13 -> 57 -> NULL Concatenated List: 34 -> 67 -> 90 -> 13 -> 57 -> NULL	

Linked-List (2b) :- Linked-list in sorted order

Output	Input
Unsorted List :-	
Enter the number of elements :- Enter the 1 element :- Enter the 2 element :- Enter the 3 element :- Enter the 4 element :-	4 45 12 7 31
Sorted List :-	
Sorted Linked List :- 7 -> 12 -> 31 -> 45 -> NULL	

Linked-List (2b) :- Queue (First-In-First-Out)

Output	Input
Enqueue :-	
Enter the number of elements to enqueue :- Enter the element 1 :- Enter the element 2 :- Enter the element 3 :- Enter the element 4 :-	4 54 24 87 12
Dequeued elements :-	
Queue: 12 -> 87 -> 24 -> 54 -> NULL Dequeued elements :- 54, 24	