|  |
| --- |
| **St. Joseph’s University**  **Bangalore, Karnataka**  PROJECT REPORT  ON  Advance Statistical Methods  Submitted by:  Ashwin S - 222BDA22  Yaswanth J – 222BDA59  Rachel Roshni – 222BDA42  Submitted to:  JAYATI BHADRA  Assistant Professor  Department of Advanced Computing  St. Joseph’s University  **Graph Machine Learning Implementation**  **(ML and Deep Learning, NetworkX, Graph Metrics and Benchmarks)**  **PROBLEM STATEMENT:**  Implementation of Graph Machine learning  **INTRODUCTION:**  NetworkX is a Python package designed for the creation, manipulation, and analysis of complex networks or graphs. It is widely used in various fields such as social network analysis, biology, computer science, and many more. NetworkX offers a comprehensive set of tools for handling different types of graphs including directed and undirected graphs, weighted and unweighted graphs, and graphs with self-loops and parallel edges.  Machine Learning on graphs has been gaining more and more attention in recent years due to the growing interest in analyzing complex data structures that can be represented as graphs. NetworkX provides a useful platform for applying various machine learning algorithms on graphs, such as clustering, classification, and link prediction.  With NetworkX, users can easily preprocess and transform their graph data to feed it into different machine learning models. Additionally, NetworkX offers a rich set of algorithms for feature extraction and graph visualization, which can be used to gain insights into the structure and properties of graphs. Overall, NetworkX provides a flexible and powerful toolset for graph analysis and machine learning on graphs.  **LITERATURE SURVEY:**  "NetworkX: A Python language software package for the creation, manipulation, and study of the structure, dynamics, and functions of complex networks." by Aric A. Hagberg, Daniel A. Schult and Pieter J. Swart. In 2008, this paper introduced the NetworkX package and its functionality for the analysis of complex networks.  "Shortest path algorithms: An overview" by Andrew V. Goldberg and Chris Harrelson. This paper provides an overview of various shortest path algorithms used in graph analysis. It covers algorithms such as Dijkstra's algorithm, Bellman-Ford algorithm, and Floyd-Warshall algorithm and discusses their relative strengths and weaknesses.  "A Survey of Centrality Measures in Social Networks" by Linton C. Freeman. This paper provides an overview of centrality measures used in social network analysis. It covers various metrics, such as degree centrality, betweenness centrality, and eigenvector centrality, and discusses their strengths and limitations.  "Measuring and Predicting the Importance of Nodes in a Network" by Lada A. Adamic and Natalie Glance. This paper explores various metrics for measuring node importance in a network, such as PageRank and HITS. It also discusses techniques for predicting node importance based on network structure and node attributes.  **AIM OF THE WORK:**  The aim of the literature survey is to provide an overview of the current techniques and algorithms in the field of graph analysis and machine learning on graphs using NetworkX. The survey explores functionalities such as graph visualization, shortest path algorithms, and efficiency metrics and their real-world applications.  **METHODS AND MATERIALS:**  The methods and materials for a graph ML metrics and shortest path project typically involve the following steps:   1. Data collection and preprocessing: Collecting and preparing the graph data for analysis, which may include cleaning, filtering, and transforming the data into a suitable format for NetworkX. 2. Graph creation and visualization: Using NetworkX to create a graph object and visualize the network structure to gain insights into its properties. 3. Feature extraction: Extracting relevant features from the graph, such as degree centrality, betweenness centrality, and eigenvector centrality, which can be used as input for machine learning models. 4. Machine learning model selection and training: Choosing a suitable machine learning algorithm for the task at hand, such as clustering or classification, and training the model using the extracted features. 5. Model evaluation: Evaluating the performance of the machine learning model on a validation dataset using appropriate metrics, such as accuracy or F1 score. 6. Shortest path algorithm implementation: Implementing a suitable shortest path algorithm, such as Dijkstra's algorithm or A\* search, to calculate the shortest path between two nodes in the graph. 7. Efficiency metric calculation: Calculating relevant efficiency metrics, such as the diameter, average path length, and clustering coefficient, to gain insights into the efficiency of the network. 8. Results analysis and visualization: Analyzing the results of the machine learning model and the efficiency metrics, and visualizing the results to gain a deeper understanding of the graph structure and properties.   The materials required for the project typically include a programming language, such as Python, and relevant packages, such as NetworkX, NumPy, and scikit-learn, as well as a dataset or a source of graph data. Additionally, appropriate hardware resources may be required to handle large and complex graph datasets.  **EXPLORATORY DATA ANALYSIS**  Exploratory data analysis (EDA) is a crucial step in any graph ML metrics and shortest path project. The purpose of EDA is to gain a better understanding of the graph data, identify important features, and establish relationships that can guide the selection of appropriate machine learning models and shortest path algorithms. Below are the key steps involved in EDA for this type of project:  Graph visualization: Visualization of the graph can provide a high-level view of its structure and expose any patterns or relationships between nodes. NetworkX offers various graph visualization tools that can be used to create visualizations like node-link diagrams and matrix plots.  Node and edge analysis: Analyzing nodes and edges in the graph can help to identify important features and relationships. Measures like degree centrality, betweenness centrality, and eigenvector centrality can be calculated with NetworkX and used to identify significant nodes or clusters in the graph.  Shortest path analysis: Analyzing the shortest path between nodes in the graph can help to identify crucial paths and connectivity patterns. NetworkX provides various shortest path algorithms like Dijkstra's algorithm and A\* search that can be used to calculate the shortest path between nodes.  Efficiency metrics: Efficiency metrics like the diameter, average path length, and clustering coefficient can provide insights into the efficiency and connectivity of the graph. NetworkX provides functions to calculate these metrics and visualize them using graphs and charts.  Correlation analysis: Examining correlations between graph features can help to identify strong predictors of significant graph properties like the shortest path length. Correlation analysis can be conducted using tools like Pandas' correlation function or heatmap visualizations.  Overall, EDA can provide valuable insights into the structure and properties of the graph data, which can guide the selection of appropriate machine learning models and shortest path algorithms for the project  **RESULTS:**  Identifying important nodes or clusters, optimizing shortest path algorithms, identifying bottlenecks or inefficiencies, and predicting network behavior. These results can provide valuable insights into network structure and behavior, which can inform decision-making in real-world networks.  The logistic Regression model with accuracy of 0.90(90%) and the deep learning GCN model with accuracy of 0.79 (79%) done for cora graph structured data.  **DISCUSSION/FUTURE WORK:**  There are several potential directions for future research that could build on the findings of a graph metrics and machine learning project. First, future work could test different machine learning algorithms to determine whether they produce more accurate or efficient results than the algorithm used in the original study. Second, larger or more complex datasets could be used to test the generalizability of the findings. Third, different optimization techniques could be explored and compared to determine their effectiveness. Fourth, future research could incorporate real-world constraints and limitations to better model real-world network structures. Finally, future work could evaluate the applicability of the findings to real-world problems and assess their impact. By pursuing these and other avenues of research, researchers can continue to develop and refine methods for analyzing and optimizing network structures using machine learning techniques.  **Git-hub Repository Link:**  [**https://github.com/ashwin200026/FDS\_EXAM**](https://github.com/ashwin200026/FDS_EXAM)  **REFERENCE:**  <https://huggingface.co/blog/intro-graphml>  <https://www.javatpoint.com/python-networkx> Exploring network structure, dynamics, and function using networkx<https://www.osti.gov/biblio/960616>Generative Graph Models with NetworkX: <https://towardsdatascience.com/generative-graph-models-with-networkx-727b154ceda4>Exploring Network Structure, Dynamics, and Function Using NetworkX: <https://www.researchgate.net/publication/236407765_Exploring_Network_Structure_Dynamics_and_Function_Using_NetworkX> |