**Lab 06**

**IT21312694 | Sandeeptha P.K.T**

Task 01

* **Stable Graph Density:** The graph density remains around 0.5 across different node counts, reflecting the set probability of edge creation (0.5) in the random graph model.
* **Smoother Degree Distributions:** As the number of nodes increases, the degree distributions become smoother and more symmetric, resembling a Gaussian distribution.
* **Increased Degree Variability:** Larger graphs show a wider range of node degrees, indicating more variability in connectivity among nodes.
* **Fluctuating Density with Node Increase:** Initially, graph density slightly decreases as nodes increase, due to the faster growth of possible connections compared to actual edges formed.

Task 02

Q1)

Supervised Learning:

* Uses labeled data to train models, meaning each input has a corresponding output label.
* The model learns to map inputs to the correct outputs based on these labels.
* Commonly used for classification and regression tasks.

Self-Supervised Learning:

* Uses unlabeled data and generates labels from the data itself, often through a pretext task.
* The model learns useful features without explicit human-provided labels.
* Commonly used in representation learning, such as training neural networks for feature extraction.

Semi-Supervised Learning:

* Combines a small amount of labeled data with a large amount of unlabeled data during training.
* The model leverages the labeled data to guide learning from the unlabeled data, improving performance with limited labeled examples.
* Often used when labeling data is expensive or time-consuming.

Q2)

Transductive Learning:

* Learns directly from the specific training and test data it encounters.
* The model does not generalize to unseen data beyond the current test set; it only aims to perform well on the given test data.
* Often used in scenarios like semi-supervised learning where test data is partially available during training.

Inductive Learning:

* Learns a general model from training data that can be applied to unseen test data.
* The model aims to generalize well to new, unseen data, predicting outcomes based on the learned patterns.
* Commonly used in standard supervised learning tasks where the focus is on generalization to future data.

Task 03

Q1)

Performance Improvements:

* 3 to 4 Layers: Minor improvement in accuracy due to enhanced feature extraction capabilities.
* 4 to 5 Layers: Further improvement as the model captures deeper graph structures.
* 5 to 6 Layers: Optimal performance observed; accuracy typically peaks around this point as the model balances complexity and overfitting.

Performance Degradation:

* 6 to 7 Layers: Slight decline in accuracy may begin due to overfitting or over-smoothing, where node representations become too similar.
* 7 to 8 Layers: Significant degradation observed; adding too many layers causes information loss, diminishing the model’s effectiveness in distinguishing between nodes.

**Overall Observations:**

* A sweet spot generally exists around 5-6 layers, where the model achieves the best trade-off between depth and performance.
* Beyond this, accuracy tends to decline unless mitigated by techniques such as skip connections or tuning of hyperparameters.

Q2)

Message Passing GNN:

* General framework for neural networks on graphs.
* Nodes exchange information through their neighbors in multiple layers.
* Key concept: aggregation of neighbor information.

Graph Convolution Network (GCN):

* Uses convolution operations on graphs similar to CNNs on images.
* Aggregates feature information from neighboring nodes.
* Often suffers from over-smoothing with too many layers.

Graph Attention Network (GAT):

* Adds attention mechanisms to GCNs.
* Allows the model to weigh the importance of neighboring nodes dynamically.
* Improves performance by focusing on more relevant neighbors.

GraphSAGE:

* Uses sampling of neighbors and aggregation to handle large-scale graphs.
* Can generalize to unseen nodes due to inductive learning capabilities.
* Efficient for large and evolving graphs.