CoreRec - connecting to the unseen

Core Recommendation Engine by Vishesh Yadav

- Current Progress: Implemented Engine Base
- Running Progress:
    - Functinal Overriding,
    - Analyzing multithread principles to enhance effeciency (3D Graphs),
    - Writing Custom Edge Labels (VishGraphs)

---

15 Visions for CoreRec :

1. **Graph-based Generative Adversarial Networks (GANs)**: Implement a graph-based GAN that generates new nodes or edges in the graph, allowing for the creation of new relationships and patterns.
2. **Graph Attention-based Recommendation**: Use graph attention mechanisms to focus on specific nodes or relationships when making recommendations, allowing for more accurate and personalized recommendations.
3. **Node Embedding-based Recommendation**: Use node embeddings, such as GraphSAGE or Graph Attention Networks, to learn dense representations of nodes and improve the recommendation accuracy.
4. **Graph-based Transfer Learning**: Implement a graph-based transfer learning approach that allows the model to learn from one graph and apply it to another, enabling the model to generalize to new graphs.
5. **Graph-based Meta-Learning**: Implement a graph-based meta-learning approach that allows the model to learn how to learn from new graphs, enabling the model to adapt to new graphs and tasks.
6. **Graph-based Explainable AI (XAI)**: Implement a graph-based XAI approach that provides explanations for the recommended nodes, such as the reasons why a particular node was chosen or the relevance of the node to the target node.
7. **Graph-based Adversarial Training**: Implement a graph-based adversarial training approach that trains the model to be robust against adversarial attacks, such as node or edge perturbations.
8. **Graph-based Transfer Learning with Adversarial Training**: Implement a graph-based transfer learning approach with adversarial training, allowing the model to learn from one graph and apply it to another while being robust against adversarial attacks.
9. **Graph-based Meta-Learning with Adversarial Training**: Implement a graph-based meta-learning approach with adversarial training, allowing the model to learn how to learn from new graphs and be robust against adversarial attacks.
10. **Graph-based Explainable AI with Adversarial Training**: Implement a graph-based XAI approach with adversarial training, providing explanations for the recommended nodes while being robust against adversarial attacks.
11. **Graph-based Multi-Task Learning**: Implement a graph-based multi-task learning approach that allows the model to learn multiple tasks simultaneously, such as node classification and edge prediction.

12. **Graph-based Transfer Learning with Multi-Task Learning**: Implement a graph-based transfer learning approach with multi-task learning, allowing the model to learn from one graph and apply it to another while learning multiple tasks.
13. **Graph-based Meta-Learning with Multi-Task Learning**: Implement a graph-based meta-learning approach with multi-task learning, allowing the model to learn how to learn from new graphs and learn multiple tasks.
14. **Graph-based Explainable AI with Multi-Task Learning**: Implement a graph-based XAI approach with multi-task learning, providing explanations for the recommended nodes while learning multiple tasks.
15. **Graph-based Graph Neural Networks (GNNs)**: Implement a graph-based GNN that learns node representations and edge weights, allowing for more accurate and personalized recommendations.