Course Outline: Social Network Analysis I (SNA-I)

Instructor: [Dr. Mudassir Shabbir]

Fall Semester, [2025]

Course Overview

- **Level:** 300-level undergraduate
- **Credits:** 3
- **Prerequisite:** Algorithms course with grade B- or higher
- **Eligibility:** Open to undergraduate students. Graduate students may attend informally but cannot take the course for credit.
- **Schedule:** Lectures will be scheduled within the 11:00 AM – 2:00 PM window.
- **Format:** Hands-on lectures, labs, homeworks (front-loaded), group activities, and final project

Learning Objectives

By the end of the course, students will:

- Understand and apply core concepts in supervised and unsupervised machine learning
- Gain proficiency in Python for machine learning and data analysis
- Model, analyze, and visualize real-world networks using graph theory
- Apply machine learning techniques to network problems such as link prediction and node classification
- Collaborate on applied projects and communicate results effectively

Weekly Breakdown

Unit I: Machine Learning Foundations (Weeks 1–6)

Week 1: Python Fundamentals & Data Handling

- Python review, Jupyter/Colab setup
- Using `pandas` for data cleaning and summaries
- **Lab:** Clean and summarize a real-world dataset
- **Homework 1:** Explore and clean a dataset (UCI/Kaggle); compute stats and visualize

Week 2: Supervised Learning – Classification

- Train/test splits, decision trees, k-NN
- Accuracy, precision, recall, F1 score
- **Lab:** Build classifiers on Titanic or similar
- **Homework 2:** Compare classifiers on a binary classification dataset, include visualizations

Week 3: Regression & Feature Engineering

- Linear regression, polynomial regression
- Feature scaling, encoding, binning
- **Lab:** Predict housing prices with engineered features
- **Homework 3:** Train a regressor with at least 3 engineered features; evaluate performance

Week 4: Model Evaluation & Pipelines

- Cross-validation, bias-variance tradeoff
- `sklearn` pipelines, reusable components
- **Lab:** Build and test classification pipelines
- **Homework 4:** Design a full pipeline with evaluation and visual reporting

Week 5: Clustering & Unsupervised Learning

- k-means, hierarchical clustering, silhouette score
- **Lab:** Cluster student or consumer datasets
- **Group Activity:** Present clustering results and interpretations

Week 6: Dimensionality Reduction

- PCA, t-SNE, embeddings intro
- **Lab:** Visualize projections in 2D
- **Optional Homework 5:** Apply dimensionality reduction and cluster/visualize results

Unit II: Network Analysis and Graph ML (Weeks 7–14)

Week 7: Introduction to Graphs

- Graph terminology, adjacency matrix/list, edge lists
- **Lab:** Load and visualize networks in `networkx`

Week 8: Centrality Measures

- Degree, betweenness, closeness, eigenvector
- **Lab:** Analyze real-world graphs for key nodes

Week 9: Communities & Subgraphs

- Louvain, label propagation, modularity
- **Lab:** Community detection in ego networks

Week 10: Network Diffusion

- SI/SIR models, cascade simulations
- **Lab:** Simulate influence spread in social graphs

Week 11: Link Prediction & Node Classification

- Heuristics (Jaccard, Adamic-Adar), ML classifiers
- **Lab:** Predict edges using logistic regression

Week 12: Dynamic Networks

- Temporal graphs, snapshots, growth modeling
- **Lab:** Analyze change over time in collaboration/email networks

Weeks 13–14: Final Projects

- Student presentations and demos
- **Deliverables:** Code notebook, slides, 2-page summary

Assessment Breakdown

Component	**Weight**
Homework Assignments (Weeks 1–5)	35%
Labs and Participation	15%
Group Activities (Weeks 5, 9)	10%
Final Project	30%
Peer Review/Reflection	10%

Tools and Platforms

- Python: `pandas`, `matplotlib`, `seaborn`, `scikit-learn`, `networkx`
- Platforms: Google Colab / Jupyter Notebooks
- Optional: Gephi, PyTorch Geometric, HuggingFace Datasets

Example Project Ideas

- Modeling diffusion of memes in Twitter networks
- Link prediction in a GitHub or DBLP collaboration graph
- Detecting clusters in Discord server graphs
- Visualizing temporal evolution of a citation network
- Combining PCA and community detection for student grouping