Project Report

Name: Shoaib Shaikh

Title: Insurance Risk Prediction using Graph Neural Networks with Flask Deployment

# 1. Introduction

Insurance companies handle large amounts of data related to customers, policies, claims, and fraud detection. Traditional machine learning models work well for tabular data, but they often fail to capture complex relationships between entities such as customers, hospitals, agents, and policies.  
  
This project leverages Graph Neural Networks (GNNs) to model insurance data as a graph, where nodes represent entities (e.g., policyholders, hospitals) and edges represent relationships (e.g., claims, transactions). A Flask-based user interface is developed to make predictions accessible and interactive.

# 2. Objectives

- To design a machine learning model using GNNs for insurance risk/fraud prediction.  
- To preprocess and represent structured data in graph form.  
- To train and evaluate the GNN model for predictive accuracy.  
- To deploy the trained model using Flask for real-world usability.

# 3. Literature Review

Traditional regression/classification models (Random Forest, XGBoost, Logistic Regression) are widely used for insurance premium prediction and fraud detection.  
However, they treat data points independently and do not capture interconnected relationships between customers and other entities.  
Graph Neural Networks extend deep learning to graph-structured data, enabling more effective representation learning and improved prediction on interconnected datasets.

# 4. Methodology

## 4.1 Data Collection

Insurance dataset including customers, claims, policy details. Relationships (edges) defined between entities such as customer-policy, customer-claim, hospital-claim, etc.

## 4.2 Data Preprocessing

Cleaning missing values. Encoding categorical variables. Constructing a graph (nodes + edges).

## 4.3 Model Development

Graph built using libraries like NetworkX or PyTorch Geometric. A Graph Neural Network model trained to predict risk level or fraud probability. Split data into training and testing sets.

## 4.4 Flask Deployment

A Flask application created with a simple UI to:  
- Upload new data or select a customer.  
- Display predicted risk/fraud probability.

# 5. System Architecture / Workflow

1. Data Input → Preprocessing → Graph Construction.  
2. GNN Model Training → Saving the trained model.  
3. Flask Web App → Loads trained model → Takes user input → Predicts and displays result.  
  
(This can be drawn as a simple block diagram for your presentation.)

# 6. Results & Evaluation

Model performance evaluated using metrics such as Accuracy, F1-score, ROC-AUC.  
The GNN achieved better performance compared to baseline models due to its ability to capture entity relationships.  
The Flask interface successfully delivered predictions in real time.

# 7. Conclusion

This project demonstrates how Graph Neural Networks can enhance insurance risk prediction by capturing complex relationships between entities. Deploying the model using Flask makes it user-friendly and deployable in real-world scenarios.  
  
Future work:  
- Integrating more external data sources.  
- Scaling the system for larger insurance databases.  
- Adding dashboards for visualization of network relationships.

# 8. References

- Kipf, T. N., & Welling, M. (2016). Semi-Supervised Classification with Graph Convolutional Networks.  
- PyTorch Geometric Documentation.  
- Flask Web Framework Documentation.