# Identifying Influential Users using Classical Models and Graph Neural Networks

Harshavardana Reddy Kolan Shikha Kumari

February 2025

**Abstract**

Stack Overflow, one of the most popular QA platforms for software development, hosts a vast community in which certain users stand out for their expertise and contributions. Identifying these "influential users" has important implications for community management, user engagement, and content curation. This paper presents a comparative study on classifying influential users by leveraging both classical machine learning algorithms (such as XGBoost) and Graph Neural Networks (GNNs). We collect user, question, and answer data through the Stack Exchange API, preprocess it to derive relevant features, and then build predictive models to determine which users are influential. Our experiments reveal the relative strengths and weaknesses of classical machine learning models compared to GNNs, offering insights into how different approaches handle community-based data. We also discuss practical dataset construction, feature engineering, and model-tuning considerations.

The results demonstrate that while traditional methods can achieve solid performance with careful feature design, GNNs show promise in capturing network relationships inherent in the platform's user interactions.

## 1 Introduction

### 1.1 Background

Stack Overflow serves as a primary hub for developers to ask questions and share expertise, amassing millions of posts and user interactions daily. In such a knowledge-sharing environment, some individuals naturally rise to prominence due to their high-quality answers, reputation, and community engagement. Identifying these "influential users" can offer valuable insights for platform moderators, community managers, and researchers aiming to enhance user experiences, reward meaningful contributions, and design incentive mechanisms.

Despite the abundance of raw data available through the Stack Exchange API, effectively classifying influential users remains challenging due to data imbalance, varying user behavior patterns, and the dynamic nature of online communities. Classical machine learning algorithms, such as logistic regression,

random forests, and XGBoost, often provide robust baselines when sufficient structured features (e.g., question score, answer score, accepted answers) are engineered. However, these methods typically overlook the graph structure underlying user interactions. Recent advances in Graph Neural Networks (GNNs) allow us to exploit the relational nature of online platforms by modeling user connectivity and influence propagation more explicitly.

# 2 Graph Neural Networks

## 2.1 Introduction to GNNs

Graph Neural Networks (GNNs) have gained popularity for modeling data with an inherent graph structure. Unlike traditional machine learning models that assume independent and identically distributed data points, GNNs consider relationships between nodes, allowing information to propagate through connected entities.

Unlike conventional machine learning models that rely solely on tabular features, GNNs exploit the topology of the data by iteratively aggregating information from neighboring nodes. This iterative message-passing mechanism enables GNNs to learn meaningful representations that capture both local and global graph structures.

For example, in social networks, GNNs can model influence propagation by aggregating information from a user's immediate neighbors, enabling predictions based on both direct and indirect connections. Similarly, in recommendation systems, GNNs can leverage collaborative filtering by analyzing user-item interactions as a bipartite graph.

Several key variants of GNNs exist, including Graph Convolutional Networks (GCNs), Graph Attention Networks (GATs), and GraphSAGE, each employing different strategies for node aggregation and message passing. These models have been successfully applied to a wide range of domains, such as fraud detection, molecular chemistry, and natural language processing, showcasing their versatility and effectiveness in structured data environments.

## 2.2 Types of Graphs

Graphs can be categorized into several types based on their properties:

- **Homogeneous Graphs:** All nodes and edges belong to the same type. For example, a simple user-interaction network.
- **Heterogeneous Graphs:** Nodes and edges belong to different types, capturing richer relationships. This is the case for Stack Overflow, where users, questions, and answers form different types of nodes and interactions.

- **Directed vs. Undirected Graphs:** In directed graphs, edges have a direction (e.g., a user answering a question), whereas undirected graphs lack this distinction.
- **Dynamic vs. Static Graphs:** A dynamic graph changes over time, while a static graph remains unchanged once constructed.

## 2.3 Heterogeneous Graph Representation for Stack Overflow

The GNN model used in our study is based on a heterogeneous graph representation, capturing the intricate relationships between different entities in the Stack Overflow platform. This heterogeneous graph consists of multiple node types—users, questions, and answers—along with directed edges representing interactions such as asks," answers," and "accepted answers."

Three primary node types:

- **Users:** Representing contributors who ask and answer questions.
- **Questions:** Representing queries posted by users.
- **Answers:** Representing responses to questions.

Edges in the graph represent interactions between these entities:

- Asks" (User → Question)
- Answers" (User → Answer)
- Has" (Question → Answer)
- Accepted Answer" (Question → Answer)
- Reverse edges" to capture bidirectional influence.
- Self-loops" to enhance message passing in GNN models.

To enhance message passing, we incorporate reverse edges, allowing bidirectional flow of information between users and content, as well as self-loops on user nodes to reinforce their intrinsic properties. Additionally, we utilize a two-layer GraphSAGE model with a sum-based aggregator, leveraging neighborhood information to enhance feature propagation. The heterogeneous GNN model is trained using a NeighborLoader, which efficiently samples neighbors for training and inference.

By structuring the Stack Overflow interactions as a heterogeneous graph, our GNN is able to leverage both direct and indirect user contributions, capturing influence propagation and contextual relationships more effectively than classical models.

# 3 Experimental Results and Discussion

## 3.1 Performance Metrics

To evaluate the performance of the proposed XGBoost-based classification model for predicting influential users on Stack Overflow, we used several standard metrics. Table 1 (Table 1) presents the performance metrics for the XGBoost classifier, while Table 2 (Table 2) reports the evaluation metrics for the GNN-based model, allowing a comparative analysis between the two approaches.

Table 1: Performance Metrics of the XGBoost Classifier

| **Metric** | **Value** |
|---|---|
| F1-score | 0.3331 |
| Accuracy | 0.8145 |
| Precision | 0.2600 |
| Recall | 0.4633 |

The initial performance of our GNN-based model is also evaluated using the same metrics. Table 2 provides the test results:

Table 2: Performance Metrics of the GNN Classifier

| **Metric** | **Value** |
|---|---|
| F1-score | 0.4732 |
| Accuracy | 0.8982 |
| Precision | 0.4491 |
| Recall | 0.5000 |