CSCI 4050U – Final Project

Steam Game Recommendation System
GROUP ID: 48
Beatriz Provido, Daniel Bryon, Gordon Law

Github Repository

Problem

- Problem: on a large platform like Steam, users face a huge catalogue of games; finding games they'll like is difficult.

- Goal: Build a **personalized recommender** to suggest games based on similar users from their reviews on certain games

- Why use a GNN?: In our application in creating a recommendation system, a graph neural network (GNN) is a well suited approach as it allows us to model the relationships and patterns between users and games. GNNs are especially effective for:
  - Collaborative filtering
  - Link prediction
  - Handling heterogeneous data (users, games, tags)

Dataset

- Games Dataset
  - Consist of ~71700 unique entries within the original dataset
  - Metadata includes:
    - Title
    - Price
    - Game tags
    - Overall reviews
    - etc
- User Review Dataset
  - Consist of +100 million unique entries within the original dataset
  - Metadata includes:
    - Steam id
    - Play time
    - Game review score (binary value)
    - Review usefulness from other users
    - etc

Neural Network Design

The design for our neural network uses a graph neural network (GNN) as its well suited for capturing the patterns that appear in heterogeneous datasets

Graph representation

- Nodes: user, games
  - User node features: review score, usefulness of review, etc
  - Game node features: tags, features, developers, etc
- Edges:
  - Users – reviewed --> Game

GNN Design

- 2 SAGEConv layers and a a torch dropout layer

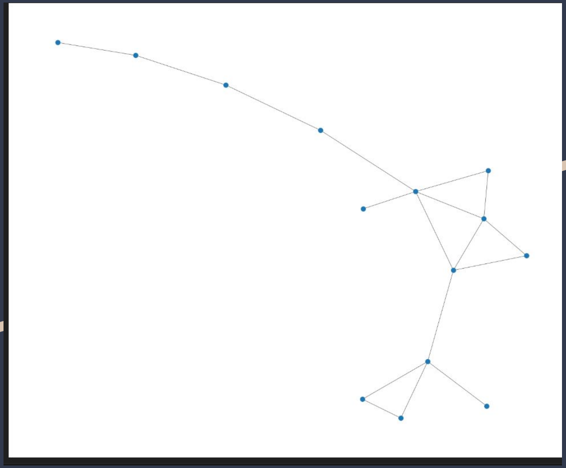
Training & Deployment

Deployment

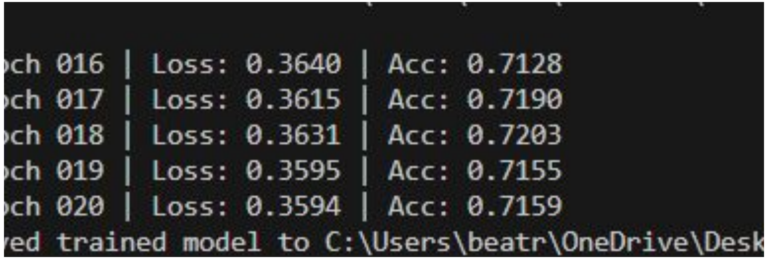
Training

- Trained a GraphSAGE GNN on the constructed user–game interaction graph.
- Used Binary Cross-Entropy loss to predict whether a user would interact positively with a game.
- Achieved accuracy improvement from ~50% to ~71.6% over 20 epochs.
  - The model peaked at about 72% accuracy around epoch 18, after which additional epochs gave no real improvement and introduced a bit of noise, which is typical mild overfitting.
- Loss consistently decreased, showing successful learning of graph structure.
- Saved the trained model and graph as gnn_model.pth and graph_data.pt for deployment.

Results



- The GNN achieved stable training, with accuracy improving from ~50% to ~71% across epochs.
- Loss decreased steadily, indicating successful learning of user–game interaction patterns.
- The model learned meaningful node embeddings that grouped related games based on review behavior.
- The deployed recommendation inference produced sensible, genre-consistent suggestions for sample



Strengths & Limitations

- **Graph-based approach** captures complex relationships between users and games that traditional recommenders (e.g., matrix factorization) cannot.
- **GraphSAGE model** learns node embeddings inductively, allowing generalization to unseen nodes.
- **Multiple feature types** (game metadata + review behavior) improve recommendation quality.
- **Efficient training** with stable convergence and competitive accuracy (~71%).
- **Lightweight deployment**—the model can be loaded and queried quickly for recommendations.

- **Sparse and fragmented graph**, with many small connected components, reduces the amount of shared structure the GNN can learn.
- **Cold-start problem**: new users or new games lack edges, making recommendations difficult.
- **Limited review metadata** (e.g., binary review score) restricts model richness; richer embeddings (text, sentiment) would perform better.
- **Accuracy ceiling** due to dataset noise and user subjectivity in reviews.
- **Evaluation constraint**: no held-out test set of real user interactions to measure recommendation relevance directly.

Conclusion

- We successfully designed and implemented a **Graph Neural Network–based recommendation system** for Steam games.

- Constructed a **user–game interaction graph** from review data and metadata, enabling graph-based learning.

- Trained a **GraphSAGE model** that achieved stable convergence, reaching **~71% accuracy** in predicting user–game interactions.

- Learned meaningful **node embeddings** that captured relationships between similar games based on review patterns.

- Implemented a full **inference pipeline**, allowing the model to load, compute embeddings, and generate Top-K game recommendations.

- Overall, the project demonstrates the value of **graph-structured learning** in improving personalized recommendation systems.

References

- https://github.com/Revadike/InternalSteamWebAPI
- https://docs.pytorch.org/docs/stable/index.html
- https://angular.dev/
- https://flask.palletsprojects.com/en/stable/
- https://uvadlc-notebooks.readthedocs.io/en/latest/index.html
- https://www.dgl.ai/dgl_docs/en/2.3.x/index.html
- https://distill.pub/2021/gnn-intro/
- https://www.kaggle.com/datasets/kieranpoc/steam-reviews?select=weighted_score_above_08.csv
- https://www.kaggle.com/datasets/nikatomashvili/steam-games-dataset