Building an Image-Based Recommendation System for Ecommerce Using Reinforcement Learning and Graph Neural Networks

Kartik Mehra - PN1096139

Research Proposal

October 2024

Abstract

This research proposes a novel image recommendation system that leverages GraphSage, a powerful graph neural network architecture, and deep learning for image feature extraction. The system aims to address the growing demand for personalized recommendations in e-commerce and other online platforms. Current recommendation systems often struggle to capture the rich information contained within product images. By combining GraphSage and deep learning, the proposed system aims to achieve a more comprehensive understanding of user preferences and product characteristics, ultimately leading to more relevant and personalized image recommendations.

Table of Contents

1. Background

The rise of e-commerce and online platforms has revolutionized the way consumers shop. Recommender systems have become a crucial tool for these platforms, helping users discover relevant products and navigate vast product selections. However, traditional recommendation systems often rely on textual data or user-reported preferences, which can be limited in capturing the full context of user needs.

1.1 History of Recommendation Engines

Historically, recommender systems have been based mostly on collaborative filtering techniques (Schafer et al., 2007). Such methods leverage user-item interaction data to infer usage patterns and thus make predictions. The reason for the general adoption of collaborative filtering approaches, such as matrix factorization (Zhang, Y., 2022) and k-nearest neighbors (Sharma, A. et al., 2023), across e-commerce, entertainment, and social media industries is based on their ability to return personalized recommendations that users can relate to and increase activity. However, as volume and complexity increase, the limitations of traditional methods began to slowly come to the forefront. Although these methods captured some such user-item associations, they often lack the various data modalities among them textual descriptions, kind social interactions, and visual contents by which a good deal of information about individual tastes or characteristics of items could be found.

There have been five important phases of evolution in recommender systems.

- All of this started with the use of collaborative filtering; however, that utilized interactions between users and items to predict preferences.
- This was followed by content-based filtering, a process that recommends items similar in features to what was liked by a user in the past.
- The Netflix Prize provided a turning point for machine learning-based methods, which were much more accurate in recommendations.
- Social recommendation systems and hybrid approaches were derived once social media had arisen.
- Deep learning is currently in the process of a revolution, capture of complex patterns, and fueling the development of context-aware and session-based recommendations.

1.2 Emerging Solutions: Deep Learning and Graph Neural Networks

Recent advancements in deep learning, particularly Convolutional Neural Networks (CNNs) (Albawi, S., Mohammed, A, T. and Al-Zawi, S., 2017), offer powerful tools for extracting meaningful features from images. These features can then be incorporated into recommendation algorithms to improve their effectiveness.

Furthermore, Graph Neural Networks (GNNs) like GraphSage have emerged as promising tools for modeling relationships within complex data structures like user-item interaction graphs. These networks can capture the intricate connections between users and the items they interact with, providing valuable insights into user preferences.

1.3 The Challenge of Image Data in Recommendations

Product images hold a wealth of information about a product's features, style, and overall appeal. These visual cues can significantly influence user purchase decisions. However, most existing recommendation systems lack the capacity to effectively leverage image data. This presents a significant challenge:

- Limited Understanding of User Preferences: Current systems might struggle to capture user preferences based solely on text descriptions or purchase history, potentially overlooking visually appealing products that align with user tastes.
- Inability to Handle Diverse Product Categories: Image data becomes particularly important for product categories where visual presentation is crucial, such as fashion, furniture, or home décor.

1.4 Need for the Proposed Research

By combining the strengths of deep learning for image feature extraction and GraphSage for modeling user-item interactions, this research proposes a novel approach to image recommendation systems. This approach has the potential to overcome the limitations of existing systems and achieve the following:

- Enhanced Recommendation Accuracy: Leveraging image data can lead to more accurate and personalized recommendations, ultimately leading to a better user experience and increased sales for online businesses.
- Improved Understanding of User Preferences: The proposed system aims to capture a more holistic understanding of user preferences by considering both visual appeal and user interaction data.
- Increased Applicability Across Product Categories: The ability to effectively utilize image data can broaden the applicability of the recommendation system to product categories where visual representation plays a significant role.

This research addresses a critical gap in current recommendation system capabilities and has the potential to significantly improve the effectiveness and user experience of online shopping platforms.

2. Problem Statement

Recommender systems have now become fundamental for online platforms that are putting their all into satisfying the user by guiding them on what to buy and exposing them to as many new things as possible. From the beginning of the internet, the issue of linking users to a large volume of accessible information has continuously existed. This issue prompted the development of recommendation systems, which are digital instruments intended to guide users towards content matching their likings.

Early recommender systems used collaborative and content based filtering approaches, which involved creating user profiles and matching users with similar preferences using historical interactions.

2.1 Collaborative Filtering Methodology:

Collaborative filtering is a recommendation system technique that analyzes user interaction data (ratings, purchases) to identify users with similar tastes (Schafer et al., 2007). These "similar users" then become the key - the system recommends items enjoyed by these like-minded users that the target user has yet to interact with. This approach is effective for uncovering hidden gems or niche products popular among similar users, but it can struggle with new users (limited data for comparison) and large datasets (computational demands). It mainly tries to predict missing ratings or recommend items based on utility matrix using algorithms like **Singular Value Decomposition (SVD)** and **Alternating Leas Squares (ALS)**. they decompose the user-item interaction matrix into lower-dimensional representations (latent factors) for both users and items. These factors capture hidden preferences and item characteristics (Bethapudi, 2024).

users	item 1	item 2	item 3	item 4
user 1	0	3	0	1
user 2	0	0	3	4
user 3	1	1	2	2
user 4	1	0	3	4
user 5	4	0	0	3

Table 1 Matrix for user Item interaction based on ratings

2.2 Content-Based Filtering Methodology:

Content-based recommender systems are another subset of recommender systems that unlike collaborative filtering, focuses on latent features of the items, rather than just the interaction data (Pazzani & Billsus, 2007). Content-based filtering uses features such as genre, keywords, metadata, and other descriptive elements, to generate profiles for both users and items. This allows the recommendation engine to make recommendations matching user preferences based on the features and characteristics of the similar items.

movie id	duration	genre	actor	year
movie 1	60	Action	actor 1	2023
movie 2	120	Comedy	actor 1	2022
movie 3	87	Romance	actor 2	2024
movie 4	55	Sci-fi	actor 4	2021
movie 5	90	Comedy	actor 1	2021

Table 2 Matrix for item and latent features

In terms of image recommendation systems, it was mainly based on the feature extraction of image components and similarity calculations which is similar to text-based recommendation systems. Instead of words, engine just uses images and analyses them in terms of visual representation before evaluating how similar they are using their pixels (Alamdari et al., 2022), meaning additional steps need to be added to the traditional recommendation pipeline to incorporate complex and large image feature vectors.

In their work, Shankar et al. (2017) introduced a comprehensive approach for constructing an e-commerce recommendation engine based on image features. They utilized a deep Convolutional Neural Network (CNN) architecture to learn embeddings that capture visual similarity across different semantic levels. By evaluating their method against the state-of-the-art model using the Exact Street2Shop dataset, they demonstrated the effectiveness of their image recommendation system for online businesses.

In their study, He and McAuley (2016) explored the application of image vectors and features derived from product images to enhance personalized ranking tasks using implicit feedback datasets. They developed the Visual Bayesian Personalized Ranking (VBPR) algorithm, which utilizes a Convolutional Neural Network (CNN) model to extract features from product images and integrates these features into a matrix factorization (MF) framework. Their findings demonstrated that this method significantly outperformed other ranking techniques across various large-scale, real-world datasets.

Chu and Tsai (2017) developed a recommendation system for restaurants that utilized both restaurant and food images. The model incorporated image features and textual features as input data, which were extracted using Convolutional Neural Network (CNN) and deep learning models. These features were then used in matrix factorization, Bayesian Personalized Ranking Matrix Factorization (BPRMF), and factorization machine (FM) methods to evaluate the relevance of recommendations. Their results indicated a slight improvement in performance.

Gharaei, Dadkhah, and Daryoush (2021) investigated the use of Convolutional Neural Network (CNN) models to create a recommendation engine for clothing and apparel. They developed two independent classification networks, both using required inputs and clothing images. The first network classified the input into target audience categories (men, women, children), while the second network categorized the input into specific clothing types. To generate the final recommendation vectors, they combined the values from the last (12th) layer of both networks and recommended products based on pairwise cosine similarity.

However, the above-mentioned traditional recommendation methods lack in identifying underlying connections and pattern because:

- It often relies on detailed descriptions or attributes associated with items. For many products, these descriptions might be limited or not very informative. Sparse data can lead to inaccurate recommendations or a lack of recommendations altogether.
- Without sufficient data on user preferences or item characteristics, the system struggles to generate accurate recommendations for new items being added to the dataset.

To solve these drawbacks and identify more complex pattern after adding image vectors, more advanced techniques like Graph Neural Netwoks like Feepwalk and Graphsage come into play, offering improved accuracy and personalization in recommendation systems.

Wu et al. (2022) researched on providing a comprehensive review of recent research on Graph Neural Networks (GNNs) in recommender systems, highlighting their advantages and categorizing existing models. They analysed the challenges of applying GNNs to different types of data in recommender systems and discuss how existing works address these challenges. Based on the survey, GNN-based models outperform traditional methods and achieve state-of-the-art results on public benchmark datasets, demonstrating their superiority in learning on graph-structured data.

2.3 Deepwalk:

Introduced in 2014, **Deepwalk** is a type of GNN (graph neural network) which uses graph structure to represent the user and item interaction and represent them into an Embedding space. It uses a concept of Random walk for embedding generation and infer main local components in the graph. Then it uses skipgram to train parameters for embeddings (Perozzi et al., 2014).

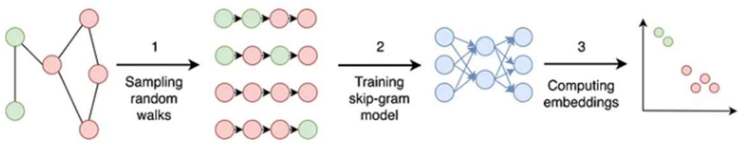

Figure 1 DeepWalk representation

Algorithm 1 DEEPWALK(G, w, d, γ, t)

Input: graph $G(V, E)$
window size w
embedding size d
walks per vertex γ
walk length t
Output: matrix of vertex representations $\Phi \in \mathbb{R}^{|V| \times d}$

```
Initialization: Sample Φ from U^{|V|×d}
Build a binary Tree T from V
for i = 0 to γ do
  O = Shuffle(V)
  for each v_i ∈ O do
    W_{v_i} = RandomWalk(G, v_i, t)
    SkipGram(Φ, W_{v_i}, w)
  end for
end for
```

Figure 2 Deepwalk Algorithm

2.4 GraphSage:

Introduced in 2017, GraphSage leverages the power of graph structures to model item relationships explicitly. Items are represented as nodes in a graph and features as node features, and edges connect them based on predefined similarities (e.g., genre similarity for movies, product category for items). During the training process, GraphSage employs a message passing mechanism where nodes aggregate information not only from their own features but also from the features of their connected neighbours in the graph. This allows GraphSage to learn richer item representations that consider both inherent item features and the context of related items.

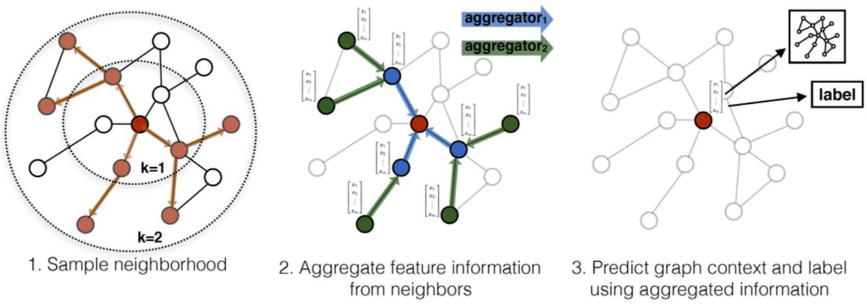

Figure 3 Visual Representaion of GraphSAGE

Algorithm 1: GraphSAGE embedding generation (i.e., forward propagation) algorithm

Input : Graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$; input features $\{\mathbf{x}_v, \forall v \in \mathcal{V}\}$; depth K; weight matrices $\mathbf{W}^k, \forall k \in \{1, ..., K\}$; non-linearity σ; differentiable aggregator functions $\text{AGGREGATE}_k, \forall k \in \{1, ..., K\}$; neighborhood function $\mathcal{N} : v \rightarrow 2^{\mathcal{V}}$

Output : Vector representations $\mathbf{z}_v$ for all $v \in \mathcal{V}$

$\mathbf{h}_v^0 \leftarrow \mathbf{x}_v, \forall v \in \mathcal{V}$;
for $k = 1...K$ **do**
  for $v \in \mathcal{V}$ **do**
    $\mathbf{h}_{\mathcal{N}(v)}^k \leftarrow \text{AGGREGATE}_k(\{\mathbf{h}_u^{k-1}, \forall u \in \mathcal{N}(v)\})$;
    $\mathbf{h}_v^k \leftarrow \sigma\left(\mathbf{W}^k \cdot \text{CONCAT}(\mathbf{h}_v^{k-1}, \mathbf{h}_{\mathcal{N}(v)}^k)\right)$
  end
  $\mathbf{h}_v^k \leftarrow \mathbf{h}_v^k / \|\mathbf{h}_v^k\|_2, \forall v \in \mathcal{V}$
end
$\mathbf{z}_v \leftarrow \mathbf{h}_v^K, \forall v \in \mathcal{V}$

Figure 4 GraphSAGE Algorithm

Hence, this research we focus on overcoming the limitations of the traditional approaches for image recommendation and increase the relevancy of recommendations, by experimenting GNN network like Graphsage, that can handle vast amount of dataset and also identify and recommend right information.

3. Aim and Objectives

This research aims to develop a novel item-to-item recommendation engine that leverages deep learning for feature extraction and GraphSage to capture relationships between items. We will achieve this through the following objectives:

- Utilize Deep Learning Models (like Convolutional Neural Networks for images and Recurrent Neural Networks for text) trained on large datasets, to capture the high-level and discriminative features that can represent each item's characteristics.
- Represent Item-Item relationship using graph theory, which will contain nodes as each item and edges based on item similarity.
- Utilize deep learning models like GraphSage, to understand Item – Item relationship based on their interaction and feature similarity.
- Evaluate the performance of recommendation engine using robust evaluation metrics like, precision-recall, AUC-ROC score and Hit rate@k.

4. Significance of the Study

Through this research we will try to leverage the power of deep learning-based approaches to identify and understand complex patterns within the data leading to better and relevant recommendations. Here's how this study contributes:

4.1 Enhancing Recommendation Accuracy

By leveraging the power of deep learning for feature extraction and GraphSage for capturing item relationships, this study aims to develop a more accurate and personalized recommendation experience for users. This can lead to increased user engagement and satisfaction with the recommendation system.

4.2 Addressing Limitations of Traditional Methods

Traditional content-based filtering approaches often struggle to capture the nuances and complexities of item representations, like cold start problem and sparse data problem. This study explores a novel approach that combines deep learning and graph theory, offering the potential to overcome these limitations and deliver more relevant recommendations.

4.3 Improved Scalability

The proposed recommendation engine leverages GraphSage, which is known for its scalability to handle large datasets. This is crucial for real-world applications were dealing with vast amounts of item data is commonplace.

4.4 Fostering Innovation

The successful implementation of this study can pave the way for further research and development in item-to-item recommendation systems. By demonstrating the effectiveness of deep learning and GraphSage, this study can inspire the exploration of new techniques and applications in this domain.

To summarize above points this study emphasizes:

1. Improved user experience through more accurate recommendations.
2. Addressing limitations of existing methods.
3. Scalability for real-world applications.
4. Potential for further innovation in recommendation systems

1. Scope of the Study

This research project aims to develop a novel item-to-item recommendation system using GraphSage and deep learning within a well-defined scope to ensure feasibility and successful completion. Here are the key considerations:

- This research will focus on developing the recommendation system for e-commerce products domain. This will allow us to focus on finding specific datasets, and extracting features related to the characteristics of the specific domain.
- The feasibility of the research hinges on the availability of a suitable dataset relevant to the chosen item domain. We will explore publicly available datasets of manageable size that allows for efficient training and evaluation of the model.
- While GraphSage offers advantages, we will consider the computational resources available and opt for a model complexity that balances performance with training time. This might involve exploring hyperparameter tuning or efficient implementation techniques to optimize the model for the chosen dataset and computational resources.
- We will use a set of relevant evaluation metrics (e.g., Hit Rate@k , precision, recall) to assess the performance of the proposed recommendation system. However, the specific metrics chosen might be tailored based on the chosen item domain and the specific goals of the recommendations (e.g., emphasizing diversity or purchase conversion).
- Developing and refining deep learning models can be time-consuming. We will establish a realistic timeline for the research, considering data exploration, model development, optimization, and evaluation phases.

There are some other milestones that can be included in further studies, but are out of scope in this research:

- This research will utilize pre-trained deep learning models for feature extraction and will not explore retraining custom models for the same use case.
- This research focuses on item-to-item recommendations and will not explore incorporating explicit user-item interactions into the model.
- Building and deploying a fully-fledged real-time recommendation system is beyond the scope of this research. However, the developed model can serve as a foundation for future exploration in this direction.
- This research will focus on utilizing features extracted from the chosen item data (e.g., product descriptions, product images). Exploring additional data sources (e.g., user reviews, social media data) might require further investigation in future studies.

- While we will evaluate the performance of GraphSage, comparing it to a wider range of recommendation algorithms might require additional resources and time.

By achieving these objectives, this research will try to contribute to the advancement of item-to-item recommendation systems, leading to more accurate and relevant suggestions for users.

6. Research Methodology

This section outlines the methodological approach for developing the item-to-item recommendation engine using GraphSage and deep learning.

GraphSage is a graph convolutional neural network (GCN) architecture that excels in recommendation tasks by leveraging the power of graph structures. In this items are represented as nodes in a graph and features as node features, and edges connect them based on predefined similarities (e.g., genre similarity for movies, product category for items). During the training process, GraphSage employs a message passing mechanism where nodes aggregate information not only from their own features but also from the features of their connected neighbours in the graph. This allows GraphSage to learn richer item representations that consider both inherent item features and the context of related items.

It works in following steps:

- Constructing the Knowledge Graph:

  The foundation of GraphSage lies in the construction of a knowledge graph specifically tailored to the recommendation task. This graph represents items as nodes, and edges are established between them based on a pre-defined similarity measure. In the context of image recommendations, similarity metrics could be derived from deep learning models (e.g., CNNs) that extract visual features from the images. These features capture the inherent characteristics of each item, allowing for establishing connections between visually similar items.

- Message Passing and Aggregation:

  GraphSage operates through an iterative process of message passing between connected nodes within the graph. These messages encapsulate information packets containing a node's own features and potentially the features of its neighbors. As messages propagate across the graph, an aggregation function plays a vital role. This function judiciously combines the information received from the node itself and its neighboring nodes, resulting in a more comprehensive representation for the node.

- Learning from the Neighbourhood:

As a result, each node also has the capability to internalize not only its native features but also the features of its neighbouring nodes via this repeated feedback exchange. This widen a range of the world how there are all connected and support another one to some degree. Such as an item that is Footwear sneakers in nature not only will be able to base its learning on just its visual features but too can also attain insights from similar footwears like athletic shoes and socks ending up enriching the overall representation.

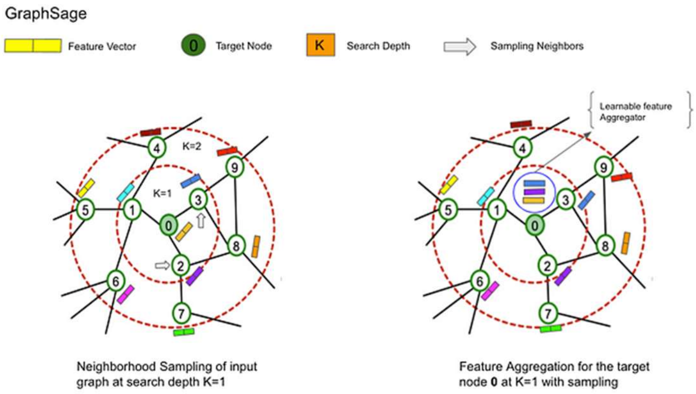

Figure 5 Neighbourhood sampling and Aggregation

- Tailored for Recommendation Tasks:

Unlike to the usual convolution neural networks used in recognizing images tasks, GraphSage functions directly on graph structure data whereas. Therefore, the process helps in efficient knowledge gathering of all the nodes of the graph with the purpose of creating for the entities representations that have not only their inherent features but also that valuable context as it is given by their relationships.

By integrating GraphSAGE into the recommendation engine provides:

- Enhanced Recommendation Accuracy By leveraging item relationships,
- Give better results for items with limited interaction history or data scarcity (new items) by using graph structure that allows the model to benefit from information gleaned from similar items, leading to improved recommendation performance.
- Scalability by efficiently handling large datasets, making it a suitable choice for real-world recommendation systems that often manage vast amounts of data.

Below mentioned is the methodology to build and evaluate recommendation engine using GraphSage learning:

6.1 Data Acquisition and Preprocessing:

- **Data Source:** Identify and acquire a suitable dataset relevant to your item domain (e.g., product information, movie details, user reviews). For this research [Amazon Product Dataset] will be used.


```
{
  "asin": "0000031852",
  "title": "Girls Ballet Tutu Zebra Hot Pink",
  "feature": ["Botiquecutie Trademark exclusive Brand",
              "Hot Pink Layered Zebra Print Tutu",
              "Fits girls up to a size 4T",
              "Hand wash / Line Dry",
              "Includes a Botiquecutie TM Exclusive hair flower
bow"],
  "description": "This tutu is great for dress up play for your
little ballerina. Botiquecute Trade Mark exclusive brand. Hot Pink
Zebra print tutu.",
  "price": 3.17,
  "imageURL": "http://ecx.images-
amazon.com/images/I/51fAmVkTbyL._SY300_.jpg",
  "imageURLHighRes": "http://ecx.images-
amazon.com/images/I/51fAmVkTbyL.jpg",
  "also_buy": ["B00JHONN1S", "B002BZX8Z6", "B00D2K1M3O",
"0000031909", "B00613WDTQ", "B00D0WDS9A", "B00D0GCI8S", "0000031895",
"B003AVKOP2", "B003AVEU6G", "B003IEDM9Q", "B002R0FA24", "B00D23MC6W",
"B00D2K0PA0", "B00538F5OK", "B00CEV86I6", "B002R0FABA", "B00D10CLVW",
"B003AVNY6I", "B002GZGI4E", "B001T9NUFS", "B002R0F7FE", "B00E1YRI4C",
"B008UBQZKU", "B00D103F8U", "B007R2RM8W"],
  "also_viewed": ["B002BZX8Z6", "B00JHONN1S", "B008F0SU0Y",
"B00D23MC6W", "B00AFDOPDA", "B00E1YRI4C", "B002GZGI4E", "B003AVKOP2",
"B00D9C1WBM", "B00CEV8366", "B00CEUX0D8", "B0079ME3KU", "B00CEUWY8K",
"B004FOEEHC", "0000031895", "B00BC4GY9Y", "B003XRKA7A", "B00K18LKX2",
"B00EM7KAG6", "B00AMQ17JA", "B00D9C32NI", "B002C3Y6WG", "B00JLL4L5Y",
"B003AVNY6I", "B008UBQZKU", "B00D0WDS9A", "B00613WDTQ", "B00538F5OK",
"B005C4Y4F6", "B004LHZ1NY", "B00CPHX76U", "B00CEUWUZC", "B00IJVASUE",
"B00GOR07RE", "B00J2GTM0W", "B00JHNSNSM", "B003IEDM9Q", "B00CYBU84G",
"B008VV8NSQ", "B00CYBULSO", "B00I2UHSZA", "B005F50FXC", "B007LCQI3S",
"B00DP68AVW", "B009RXWNSI", "B003AVEU6G", "B00HSOJB9M", "B00EHAGZNA",
"B0046W9T8C", "B00E79VW6Q", "B00D10CLVW", "B00B0AVO54", "B00E95LC8Q",
"B00GOR92SO", "B007ZN5Y56", "B00AL2569W", "B00B608000", "B008F0SMUC",
"B00BFXLZ8M"],
  "salesRank": {"Toys & Games": 211836},
  "brand": "Coxlures",
  "categories": [["Sports & Outdoors", "Other Sports", "Dance"]]
}
```


Figure 6 Data Snapshot

feature	description
asin	Unique item identifier
title	Item name
feature	List of item features
description	Item textual description
price	Item price in US dollars, at time of data extraction
imageURL	Item image URL
imageURLHighRes	High resolution item image URL
related	related products (also bought, also viewed, bought together, buy after viewing)
salesRank	sales rank information
brand	brand name

categories	list of categories the item belongs to
tech1	the first technical detail table of the product
tech2	the second technical detail table of the product
similar	similar product table

Table 3 Data Dictionary

- **Data Cleaning:** Address missing values, outliers, and inconsistencies within the data. Techniques like data imputation or removal might be necessary. Converting json dataset and image will be linked to the original table.
- **Feature Engineering:** Depending on the item domain
- **For Images:** Extract visual features using the pre-trained deep learning model (e.g., CNN) trained on a large image dataset. Convolutional Neural Networks (CNNs) are a powerful deep learning architecture designed to learn directly from raw data. Their strength lies in pattern recognition, especially within images. By applying convolutional filters, CNNs can effectively identify and classify objects, categories, and even analyze sequential data like audio or time series.
- **For Textual Data:** Preprocess text data through techniques like tokenization, stemming/lemmatization, and stop word removal. Subsequently, use word embeddings or pre-trained models (e.g., BERT) to extract features that capture the semantic meaning of the text. BERT (Bidirectional Encoder Representations from Transformers) is a transformer based deep learning model, uses a bi-directional approach which focuses on both left and right context of words in a sentence simultaneously.

6.2 Constructing the Item-Similarity Graph:

- Nodes: Represent each item in the dataset as a node in the graph.
- Edges: Connect nodes based on the similarity of their features obtained from the deep learning models.
- Define a threshold or metric (e.g., cosine similarity) to determine the strength of the connection between items. This reflects how closely related the items are in terms of their characteristics.

6.3 GraphSage:

- **Node Features:** Assign the deep learning extracted features from pre-processing as the initial features for each item node in the graph.
- **Message Passing:** Implement GraphSage layers to iteratively aggregate information from neighbouring nodes. This allows each item to incorporate not only its own features but also the features of similar items, effectively learning a richer representation that captures broader relationships within the item domain.
- **Hyperparameter Tuning:** Optimize hyperparameters like the number of GraphSage layers, message passing function, and learning rate through techniques like grid search or Bayesian optimization.

6.4 Recommendation Generation:

- Utilize the learned item representations from GraphSage.
- For a given target item, recommend a set of similar items based on their proximity in the embedding space. This can be achieved using techniques like nearest neighbour search or ranking-based approaches.

6.5 Evaluation Metrics:

- **Hit Rate@k (HR@k):** This metric measures the proportion of times the recommended items appear within the top-k retrieved items for a given target item. It assesses the model's ability to recommend relevant items at the top of the recommendation list.

$$HR = \frac{|U_{hit}^{L}|}{|U_{all}|}$$

Where $|U_{hit}^{L}|$ is the number of items for which correct answer is included in top L recommendations, $|U_{all}|$ is the total number of items in the test dataset.

- **Precision and Recall:** These metrics can also be employed to evaluate the model's performance. Precision measures the ratio of truly relevant items among the retrieved recommendations, while Recall indicates the proportion of relevant items retrieved from the dataset.

$$precision = \frac{|\{relevantdocuments\} \wedge \{retrieveddocuments\}|}{|\{retrieveddocuments\}|}$$

$$recall = \frac{|\{relevantdocuments\} \wedge \{retrieveddocuments\}|}{|\{relevantdocuments\}|}$$

6.6 Workflow:

- Preprocess data and extract and visual and textual features using deep learning models.
- Construct the item-item graph based on items interactions.
- Implement GraphSage to learn enriched item representations.
- Develop a recommendation generation strategy using the learned representations.
- Evaluate the model's performance using metrics like HR@k, precision, and recall.

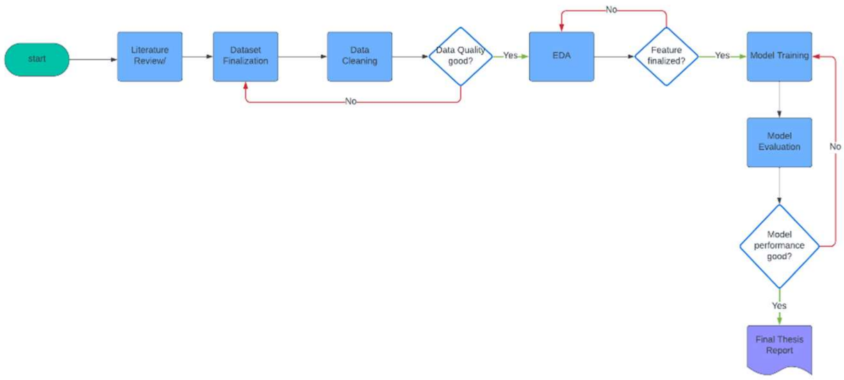

Figure 7 Workflow diagram

7. Requirements Resources

Developing a robust item-to-item recommendation system using GraphSage and deep learning required various types of resources:

7.1 Minimum Hardware Requirements:

- Processor: A recent generation Intel Core i7 or AMD Ryzen 7 processor with at least 8 cores and 16 threads.

- Graphics Processing Unit (GPU): A mid-range NVIDIA GeForce RTX 3060 or AMD Radeon RX 6600 series GPU with at least 6GB of dedicated GDDR6 memory is recommended. The GPU is crucial for accelerating deep learning training.
- System Memory (RAM): At least 32GB of DDR4 RAM will provide sufficient memory for handling data processing and model training.
- Storage: A combination of a solid-state drive (SSD) for the operating system and software installation, and a larger capacity hard disk drive (HDD) for storing the dataset is recommended. The SSD will improve overall system responsiveness, while the HDD offers the necessary space for large datasets.
- For very large datasets or highly complex models, leveraging cloud-based computing resources (e.g., Google Colab, Amazon SageMaker) might be a viable option. These services offer access to powerful GPUs and scalable storage solutions on a pay-as-you-go basis.

7.2 Software:

- Deep learning libraries like TensorFlow, PyTorch, or Keras will be required for using pre-trained models for feature extraction.
- Graph libraries like PyG (PyTorch Geometric) or DGL (Deep Graph Library) will be used to build the item-similarity graph and implementing GraphSage.
- Additional libraries like NumPy, Pandas, and scikit-learn will be required for data manipulation, analysis, and evaluation.

8. Research Plan

For this research purpose, the tasks have been divided in 3 milestone:

1. Research Proposal
    - Literature Search
    - Literature Review
    - Research Proposal Submission
2. Interim Report
    - Additional Literature Review
    - EDA and Feature Extraction

- Feature Selection
- Interim Repot Submission

3. Final Thesis

- Model Development
- Model Evaluation
- Hyperparameter Tuning
- Model Evaluation
- Complete Report

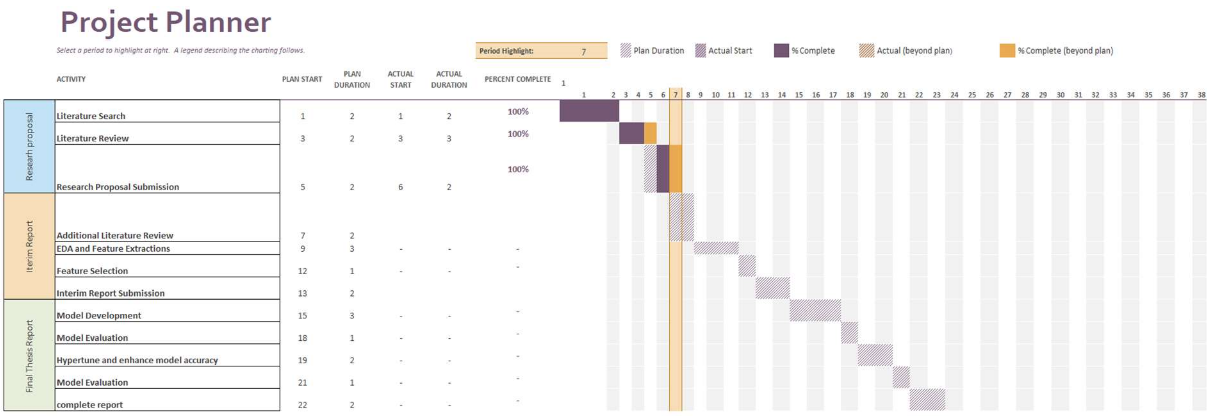

Figure 8 - Project Planner

During the course of this research there are certain risks involved, that also needs to be considered. Following are the risks involved and contingency plan for the same:

8.1 Data Quality and Availability:

There might be insufficient or low-quality user-item interaction data or image data, hindering model training and performance.

<u>Contingency Plans</u>

- Explore alternative data sources, such as publicly available datasets or collaborating with companies that can provide relevant data.
- If data quantity is limited, consider data augmentation techniques to artificially increase the dataset size while preserving its characteristics.
- Implement data cleaning techniques to address inconsistencies or errors within the existing data.

8.2 Computational Resources:

Training deep learning models can be computationally expensive and resource-intensive, exceeding available hardware or budget.

Contingency Plans

- Use cloud computing platforms that offer scalable resources to handle the computational demands.
- Simplify deep learning architecture or using pre-trained models to decrease training time and resource consumption.

8.3 Model Performance:

The recommendation model might not achieve the desired level of accuracy or fail to capture complex user preferences.

Contingency Plans

- Experiment and try with different GraphSage network architectures or hyperparameter tuning to improve model performance.
- Explore different deep learning architectures for image feature extraction to increase model performance
- Additional features can be included (e.g., user demographics, purchase history) to increase model performance

9. References

- Schafer, J.B. *et al.* (2007) 'Collaborative Filtering recommender systems,' in *Springer eBooks*, pp. 291–324. https://doi.org/10.1007/978-3-540-72079-9_9.
- Pazzani, M.J. and Billsus, D. (no date) "Content-Based Recommendation Systems," in *The Adaptive Web*. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 325–341. Available at: https://doi.org/10.1007/978-3-540-72079-9_10.
- Zhang, Y. (2022) 'An Introduction to Matrix factorization and Factorization Machines in Recommendation System, and Beyond,' arXiv (Cornell University) [Preprint]. https://doi.org/10.48550/arxiv.2203.11026.

- Bethapudi, D.P. (2024) “Matrix Factorization Based Recommendation System using Hybrid Optimization Technique,” *SSRN Electronic Journal* [Preprint]. Available at: https://doi.org/10.2139/ssrn.4793182.
- Sharma, A. et al. (2023) 'A Review On Collaborative Filtering Using Knn Algorithm,' A Review on Collaborative Filtering Using Knn Algorithm [Preprint]. https://doi.org/10.1109/otcon56053.2023.10113985.
- Malekpour Alamdari, P. *et al.* (2022) “Image-based Product Recommendation Method for E-commerce Applications Using Convolutional Neural Networks,” *Acta Informatica Pragensia*, 11(1), pp. 15–35. Available at: https://doi.org/10.18267/j.aip.167.
- Albawi, S., Mohammed, T.A. and Al-Zawi, S. (2017) 'Understanding of a convolutional neural network,' *Understanding of a Convolutional Neural Network* [Preprint]. https://doi.org/10.1109/icengtechnol.2017.8308186.
- Oliveira, J. *et al.* (2023b) 'Footwear segmentation and recommendation supported by deep learning: an exploratory proposal,' *Procedia Computer Science*, 219, pp. 724–735. https://doi.org/10.1016/j.procs.2023.01.345.
- Won, H.-M., Heo, Y.S. and Kwak, N. (2023) 'Image recommendation system based on environmental and human face information,' *Sensors*, 23(11), p. 5304. https://doi.org/10.3390/s23115304.
- *Image feature extraction and analysis algorithm based on multi-level neural network* (2021). https://ieeexplore.ieee.org/document/9418309.
- Alamdari, P.M. *et al.* (2022) 'Image-based product recommendation method for e-commerce applications using convolutional neural networks,' *Acta Informatica Pragensia*, 11(1), pp. 15–35. https://doi.org/10.18267/j.aip.167.
- Bilski, J. *et al.* (2023) 'Fast computational approach to the Levenberg-Marquardt algorithm for training feedforward neural networks,' *Journal of Artificial Intelligence and Soft Computing Research*, 13(2), pp. 45–61. https://doi.org/10.2478/jaiscr-2023-0006.
- Afoudi, Y., Lazaar, M. and Hmaidi, S. (2023) 'An enhanced recommender system based on heterogeneous graph link prediction,' *Engineering Applications of Artificial Intelligence*, 124, p. 106553. https://doi.org/10.1016/j.engappai.2023.106553.
- Angadi, S. Keerthi Gorripati, V. Rachapudi, Y. Krishna Kuppili and P. Dileep, (2021) ‘*Image-based Content Recommendation System with CNN’* (2021) Fifth International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC), Palladam, India, 2021, pp. 1260-1264. https://ieeexplore.ieee.org/document/9641026.

- Wang, Y. (2022) 'Friend Recommendation using GraphSAGE - Stanford CS224W GraphML Tutorials - Medium,' *Medium*, 12 February. https://medium.com/stanford-cs224w/friend-recommendation-using-graphsage-ffcda2aaf8d6.
- Perozzi, B., Al-Rfou, R. and Skiena, S. (2014) 'DeepWalk,' *DeepWalk: Online Learning of Social Representations* [Preprint]. https://doi.org/10.1145/2623330.2623732.
- Hamilton, W.L., Ying, R. and Leskovec, J. (2017b) *Inductive representation learning on large graphs*. https://arxiv.org/abs/1706.02216v4.
- Shankar, D. et al. (2017) 'Deep Learning based Large Scale Visual Recommendation and Search for E-Commerce,' arXiv (Cornell University) [Preprint]. https://doi.org/10.48550/arxiv.1703.02344.
- He, R. and McAuley, J. (2016) “Ups and Downs,” in Proceedings of the 25th International Conference on World Wide Web. Republic and Canton of Geneva, Switzerland: International World Wide Web Conferences Steering Committee, pp. 507–517. Available at: https://doi.org/10.1145/2872427.2883037.
- Chu, W.-T. and Tsai, Y.-L. (2017) “A hybrid recommendation system considering visual information for predicting favorite restaurants,” World Wide Web, 20(6), pp. 1313–1331. Available at: https://doi.org/10.1007/s11280-017-0437-1.
- Yarahmadi Gharaei, N., Dadkhah, C. and Daryoush, L. (2021) “Content-based Clothing Recommender System using Deep Neural Network,” in 2021 26th International Computer Conference, Computer Society of Iran (CSICC). IEEE, pp. 1–6. Available at: https://doi.org/10.1109/CSICC52343.2021.9420544.
- Wu, S. *et al.* (2022) 'Graph Neural networks in Recommender Systems: A survey,' *ACM Computing Surveys*, 55(5), pp. 1–37. https://doi.org/10.1145/3535101.