OTT Recommendation System using ANN (Approximate nearest neighbors) Algorithms

Team Members:

1. **Yuvraj Singh – 25PGAI0019**
2. **Ninad Jadhav- 25PGAI0098**
3. **Shivendra Pratap Singh- 25PGAI0025**
4. **Prajwal Wagh- 25PGAI0109**
5. **Piyush Borse- 25PGAI0026**

List of Illustrations:

Abstract

This report provides an in-depth examination of Approximate Nearest Neighbors (ANN) algorithms and their pivotal role in constructing efficient, scalable, and high-performing recommendation systems. The project emphasizes transitioning from traditional text-based search methodologies to a modern, vector-based retrieval framework. This transition leverages the capabilities of ANN algorithms and high-performance vector databases like Qdrant to deliver superior semantic accuracy and significantly reduced latency. This transition leverages the capabilities of ANN algorithms, high-performance vector databases like **Qdrant**, and additional methodologies such as **Content-Based Filtering** and **Collaborative Filtering** to deliver superior semantic accuracy and significantly reduced latency.

The study is structured to address the following objectives:

1. **Implementation of ANN Algorithms:** We implemented various ANN algorithms, including HNSW, Annoy, and IVF (Inverted Flat Index), and evaluated their performance in different scenarios. These implementations were rigorously compared against conventional **text + vector search** combinations to demonstrate the advancements achieved through ANN.
2. **Semantic Accuracy Analysis:** A key focus of the project was evaluating the semantic accuracy of the ANN algorithms. Metrics such as Precision@K, Recall@K, Mean Average Precision (MAP), Normalized Discounted Cumulative Gain (NDCG), and F1 Score@K were utilized to determine the most effective algorithms for capturing semantic relationships.
3. **Algorithm and Distance Measure Optimization:** Experiments were conducted to explore the interplay between ANN algorithms and various distance metrics (e.g., Cosine, Manhattan, Chebyshev, Canberra). Standard deviation analysis of normalized similarity distributions was employed to identify the optimal algorithm-distance metric pair.
4. **Database Integration:** The final step involved integrating the optimized ANN algorithm (HNSW) with the Qdrant vector database, chosen for its superior compatibility with Cosine similarity and scalability.

5. **Incorporation of Vision Transformer (ViT):** To further enhance the embedding quality and recommendation accuracy, Vision Transformer (ViT) models were incorporated, enabling better representation for multimodal datasets.
6. **Content-Based Filtering:**

   A robust Content-Based Filtering system was developed to provide personalized recommendations by leveraging item metadata such as genres, cast, director, and other descriptive features. Key steps included:

   Embedding Generation: Dense vector embeddings were created for each movie using the all-MiniLM-L6-v2 model.

   User Profile Creation: Aggregated embeddings of movies liked by the user were used to form a user profile.

   Recommendation Mechanism: Cosine similarity between the user profile and movie embeddings identified the most relevant recommendations.
7. **Collaborative Filtering:**

   Collaborative Filtering was implemented to leverage user behavior patterns for recommendations. The system worked by:

   User-Item Interaction Matrix: Constructing a matrix of user-movie interactions.

   Latent Factor Analysis: Using Truncated Singular Value Decomposition (SVD) to extract latent features for users and movies.

   Recommendation Mechanism: Identifying similar users or items in the latent space and aggregating preferences to recommend movies.

The results of this study demonstrate the transformative potential of vector-based retrieval systems. By optimizing the synergy between algorithms, distance metrics, and databases, we achieved a system that outperforms traditional approaches in both accuracy and efficiency. This report serves as a detailed guide to understanding the steps, challenges, and breakthroughs involved in this journey toward building next-generation recommendation systems.

INTRODUCTION:

In the era of digital transformation, the demand for personalized, real-time recommendations is at an all-time high. Traditional search methods, which rely heavily on text-based retrieval, often struggle to meet the expectations for accuracy, scalability, and latency in modern applications. As the complexity of user demands grows, there is a pressing need for innovative solutions capable of handling large-scale data and delivering relevant results efficiently.

To address these challenges, **Vector-Based Retrieval** powered by **Approximate Nearest Neighbors (ANN)** algorithms has emerged as a cutting-edge approach. By representing data as vectors in high-dimensional space, ANN algorithms enable the discovery of relationships and patterns that are semantically meaningful, paving the way for enhanced recommendation systems. Complementing this approach, **Content-Based Filtering** and **Collaborative Filtering** serve as foundational techniques, enriching the system's ability to deliver highly relevant, personalized recommendations.

Content-Based Filtering focuses on leveraging the inherent characteristics of items, such as metadata (e.g., genre, cast, director), to generate recommendations tailored to an individual user's preferences. This method ensures effective handling of new or niche items, where user interaction data might be sparse.

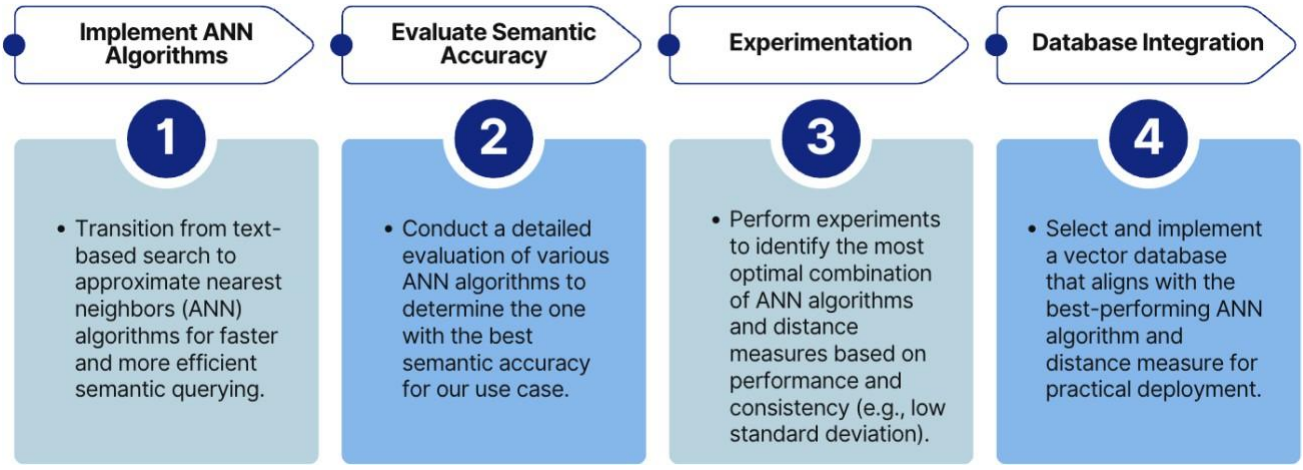
Collaborative Filtering, on the other hand, harnesses the power of user interactions by analyzing patterns of similar behavior across users. It excels at uncovering latent relationships and recommending items that users with similar preferences have interacted with, even when explicit metadata is unavailable.

This report delves into the implementation of these techniques alongside ANN algorithms, evaluates their semantic accuracy, and identifies the best algorithm and distance measure combination for real-world applications. Furthermore, it explores the integration of these algorithms with **Qdrant**, a high-performance vector database, and incorporates **Vision Transformer (ViT)** models for advanced feature extraction and improved recommendation accuracy.

OBJECTIVES:

This project was driven by the following key objectives:

1. **Transition from Text-Based Search to Vector-Based Retrieval:**
    - Replace traditional text-based search mechanisms with vector-based retrieval systems for superior latency performance and semantic relevance.
    - Demonstrate how ANN algorithms outperform text-based methods by capturing deeper semantic relationships.
2. **Evaluate the Semantic Accuracy of ANN Algorithms:**
    - Assess the performance of various ANN algorithms (e.g., HNSW, Annoy, IVF) using comprehensive semantic accuracy metrics.
    - Metrics include Precision@K, Recall@K, Mean Average Precision (MAP), Normalized Discounted Cumulative Gain (NDCG), and F1 Score@K.
3. **Identify the Best ANN Algorithm and Distance Measure Combination:**
    - Conduct systematic experiments to test combinations of ANN algorithms and distance measures (e.g., Cosine, Manhattan, Chebyshev, Canberra).
    - Use standard deviation analysis of normalized similarity distributions to select the most consistent and accurate configuration.
4. **Integrate with a High-Performance Vector Database (Qdrant):**
    - Implement the best algorithm-distance pair within Qdrant to leverage its support for HNSW indexing and Cosine similarity.
    - Highlight the scalability, efficiency, and accuracy of this integration for production- ready systems.
5. **Explore Vision Transformer (ViT) Models for Enhanced Feature Extraction:**
    - Incorporate Vision Transformer (ViT) models to generate high-quality embeddings for multimodal datasets.
    - Demonstrate the role of ViT in enriching recommendations through more precise feature representation.
6. **Content-Based Filtering for Metadata-Driven Recommendations:**
    - Leverage metadata such as genres, cast, and directors to recommend items tailored to user preferences.
    - Generate vector embeddings using Sentence Transformers for item metadata and create user profiles by aggregating embeddings of previously liked items.
    - Use Cosine Similarity to calculate the relevance of items to the user profile and recommend the top-n most similar items.
    - Demonstrate the effectiveness of Content-Based Filtering in handling niche items and the item cold-start problem
7. **Collaborative Filtering for User Behavior-Based Recommendations:**
    - Utilize a user-item interaction matrix to uncover latent patterns in user behavior.
    - Apply Truncated Singular Value Decomposition (SVD) to extract latent features representing user and item preferences.
    - Recommend items by finding users with similar latent profiles or items frequently liked by similar users.
    - Highlight the strength of Collaborative Filtering in leveraging community-driven insights to recommend items without relying on extensive metadata.



IMPLEMENTATION OF ANN ALGORITHM

Dataset Preparation

1. **Dataset Selection:**
   - The IMDb dataset was selected due to its extensive and diverse collection of movie- related information, making it ideal for testing recommendation systems.
   - The dataset contains columns such as *title*, *genre*, *overview*, *crew*, and other metadata, providing a rich source of information for semantic embedding generation.
2. **Data Preprocessing:**
   - **Text Consolidation:** Key text columns (*title*, *genre*, *overview*, and *crew*) were merged into a single textual representation for each movie. This combined representation ensures comprehensive input for embedding generation.
   - **Data Cleaning:**
     - Missing values in critical columns were addressed through imputation methods such as filling with default text or averages.
     - Data inconsistencies (e.g., mismatched genres, incorrect crew names) were normalized to maintain uniformity across entries.
   - **Text Normalization:**
     - Lowercasing and tokenization were applied.
     - Stopwords, special characters, and redundant spaces were removed to enhance semantic clarity.

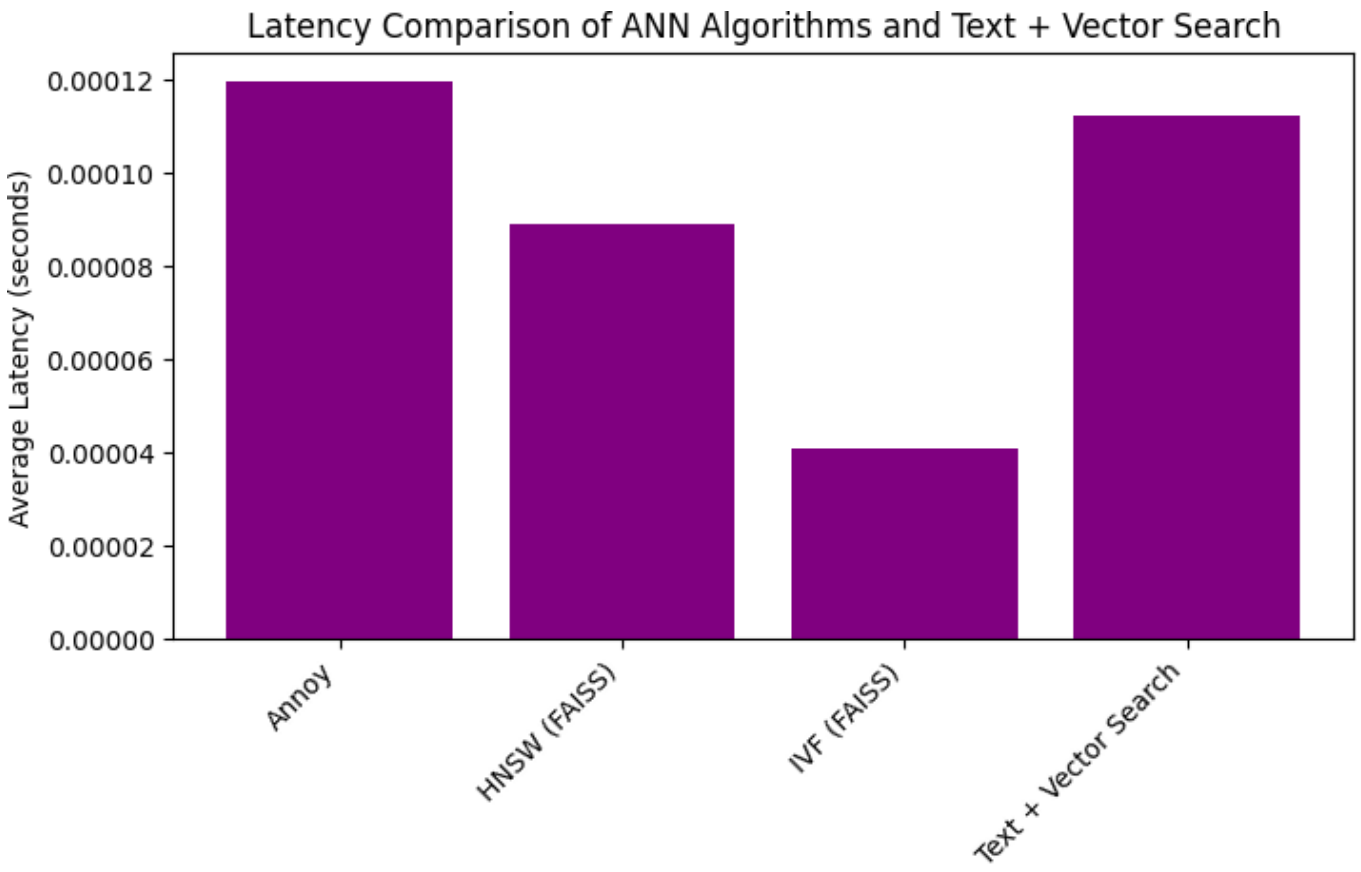
Embedding Generation

1. **Model Selection:**
    - The **all-MiniLM-L6-v2** model from the Sentence Transformers library was chosen for generating sentence embeddings.
        - This model is optimized for speed and performance, making it scalable for large datasets.
        - It captures contextual and semantic information effectively, producing high-quality dense vector representations.
2. **Encoding Process:**
    - The consolidated text representations of each movie entry were passed through the model to generate **dense embeddings**.
    - Each embedding vector represents the semantic meaning of the text, capturing nuances like relationships between genres, actors, and plot summaries.

Algorithm Implementation

Three ANN algorithms were implemented to evaluate their efficiency and scalability:

1. **HNSW (Hierarchical Navigable Small World):**
    - **Key Features:**
        - Provides high-speed nearest neighbor searches, even for large-scale datasets.
        - Supports dynamic updates, making it suitable for real-time systems.
    - **Use Case:**
        - Best suited for large datasets requiring real-time, low-latency recommendations.
        - Highly scalable due to its graph-based indexing structure.
2. **IVF (Inverted File Index):**
    - **Key Features:**
        - Splits the dataset into smaller clusters for efficient querying.
        - Works well with structured data and complex queries.
    - **Limitations:**
        - Requires significant setup time to partition the dataset.
    - **Use Case:**
        - Suitable for systems with predictable query patterns and structured datasets.
3. **ANNOY (Approximate Nearest Neighbors Oh Yeah):**
    - **Key Features:**
        - Lightweight and memory-efficient.
        - Uses static tree-based indices, enabling fast approximate searches.
    - **Use Case:**
        - Ideal for memory-constrained environments or offline systems.


Latency Comparison of ANN Algorithms and Text + Vector Search
Average Latency (seconds)
0.00012
0.00010
0.00008
0.00006
0.00004
0.00002
0.00000
Annoy
HNSW (FAISS)
IVF (FAISS)
Text + Vector Search

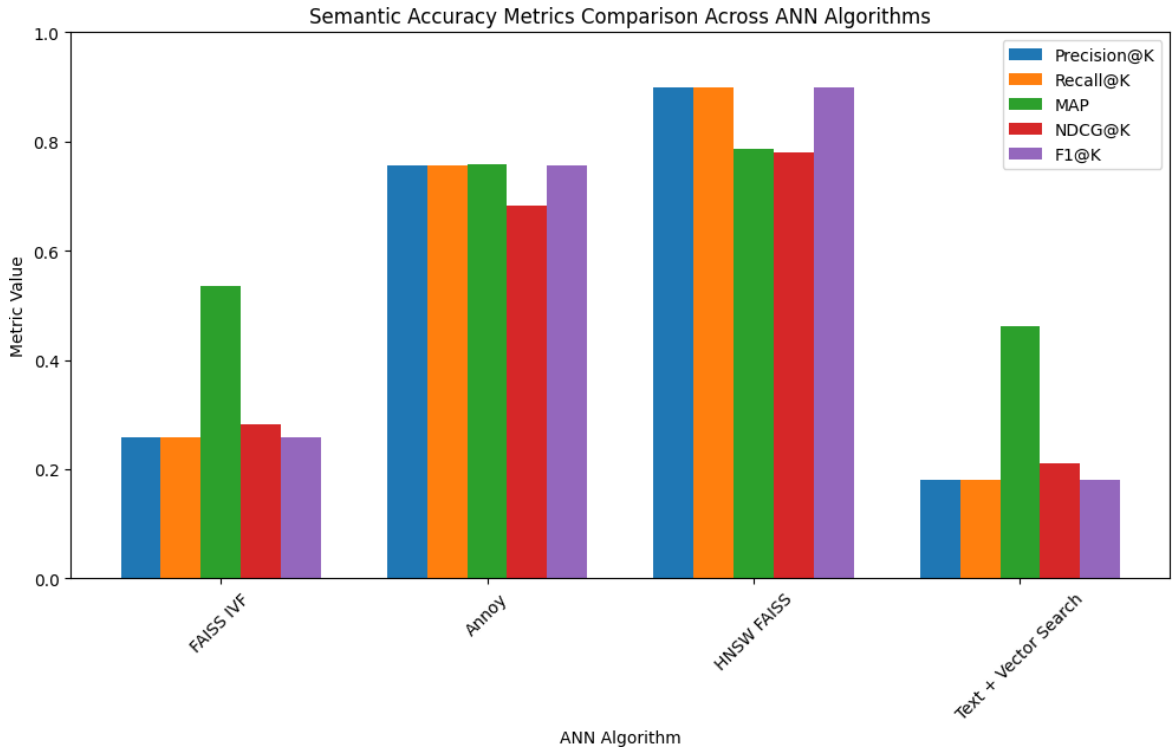
Comparison with Text-Based Search

1. **Hybrid Baseline:**
   - A **hybrid search system** combining text-based search and vector-based retrieval was used as a baseline for comparison.
   - Text-based search was employed for keyword matching, while vector-based search added semantic depth.
2. **Transition to Pure Vector Search:**
   - Results demonstrated the superiority of vector-based search in terms of:
     - **Latency:** Faster retrieval times for large-scale datasets.
     - **Semantic Accuracy:** Better contextual and relational understanding of data.
   - Pure vector-based retrieval eliminates the limitations of keyword dependency, making it a more robust solution for modern recommendation systems.

Semantic Accuracy Evaluation

Metrics Used:

1. Precision@K: Proportion of top-K neighbors that match the ground truth.
2. Recall@K: Ratio of relevant items retrieved among all relevant items.
3. Mean Average Precision (MAP): Balances precision across all ranks.
4. Normalized Discounted Cumulative Gain (NDCG@K): Rewards higher-ranking relevant items.
5. F1 Score@K: Harmonic mean of Precision@K and Recall@K.



Key Findings:

1. HNSW consistently outperformed other algorithms in terms of NDCG@K and Precision@K.
2. Cosine Similarity emerged as the best distance measure, providing high semantic consistency.
3. ANNOY, while fast, showed lower semantic accuracy due to its approximate nature.

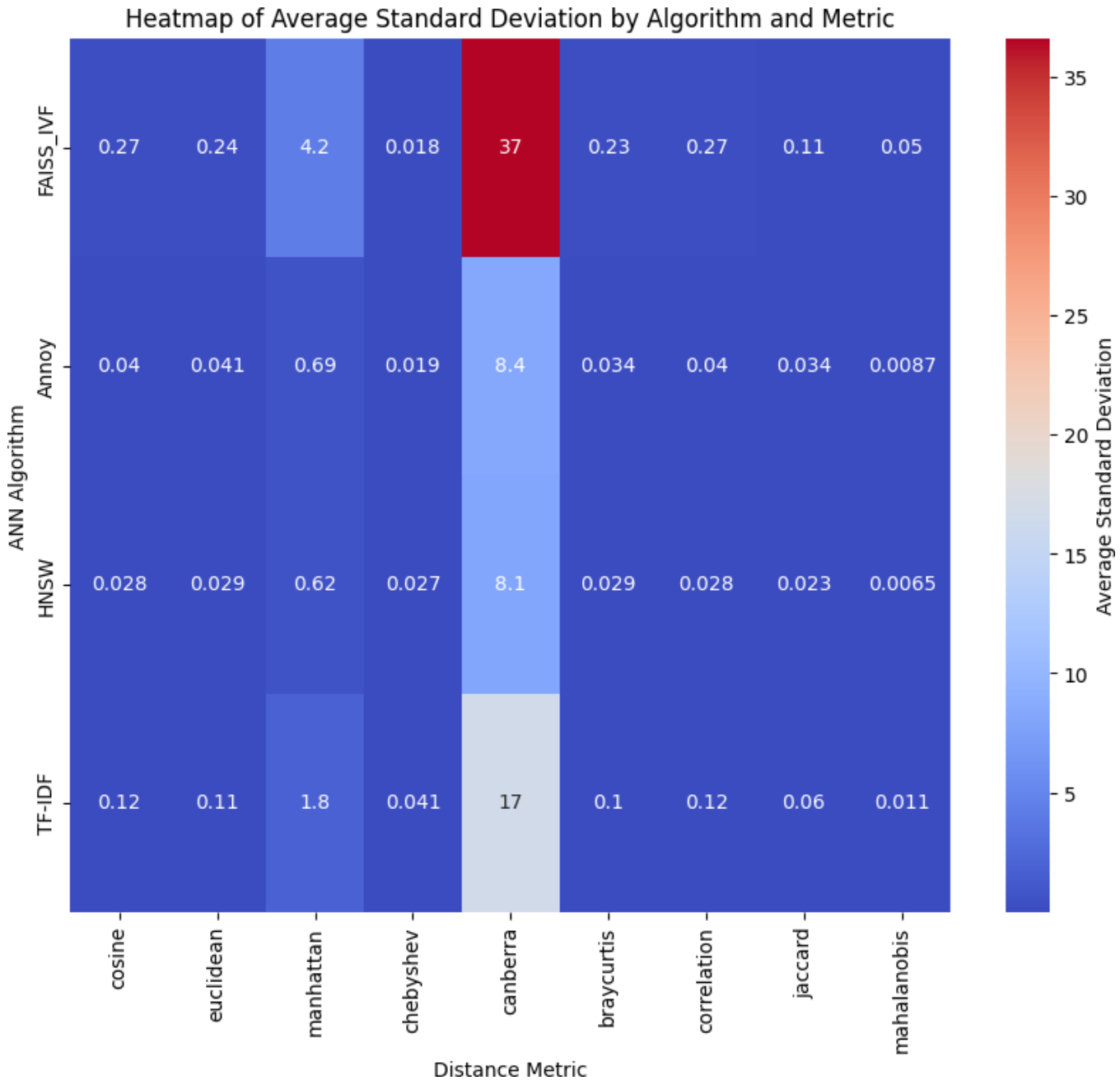
Analysis: Best ANN Algorithm and Distance Measure

Methodology:

1. Diversity of Distance Measures:
    - Explored 10 metrics including Cosine, Mahalanobis, Chebyshev, Manhattan, and Canberra.
    - Normalized outputs to a -1 to +1 scale.
2. Standard Deviation Analysis:
    - Calculated standard deviation of similarity scores across all distance measures.
    - Lower standard deviation indicated better agreement among measures.

Results:

1. HNSW with Cosine and Chebyshev Similarity had the lowest standard deviation, marking it as the most reliable combination.
2. Wide deviations in scores for and Manhattan and Canberra measures indicated lower reliability in certain datasets.


Heatmap of Average Standard Deviation by Algorithm and Metric
ANN Algorithm
FAISS_IVF
Annoy
HNSW
TF-IDF
0.27 0.24 4.2 0.018 37 0.23 0.27 0.11 0.05
0.04 0.041 0.69 0.019 8.4 0.034 0.04 0.034 0.0087
0.028 0.029 0.62 0.027 8.1 0.029 0.028 0.023 0.0065
0.12 0.11 1.8 0.041 17 0.1 0.12 0.06 0.011
cosine
euclidean
manhattan
chebyshev
canberra
braycurtis
correlation
jaccard
mahalanobis
Distance Metric
Average Standard Deviation
35
30
25
20
15
10
5

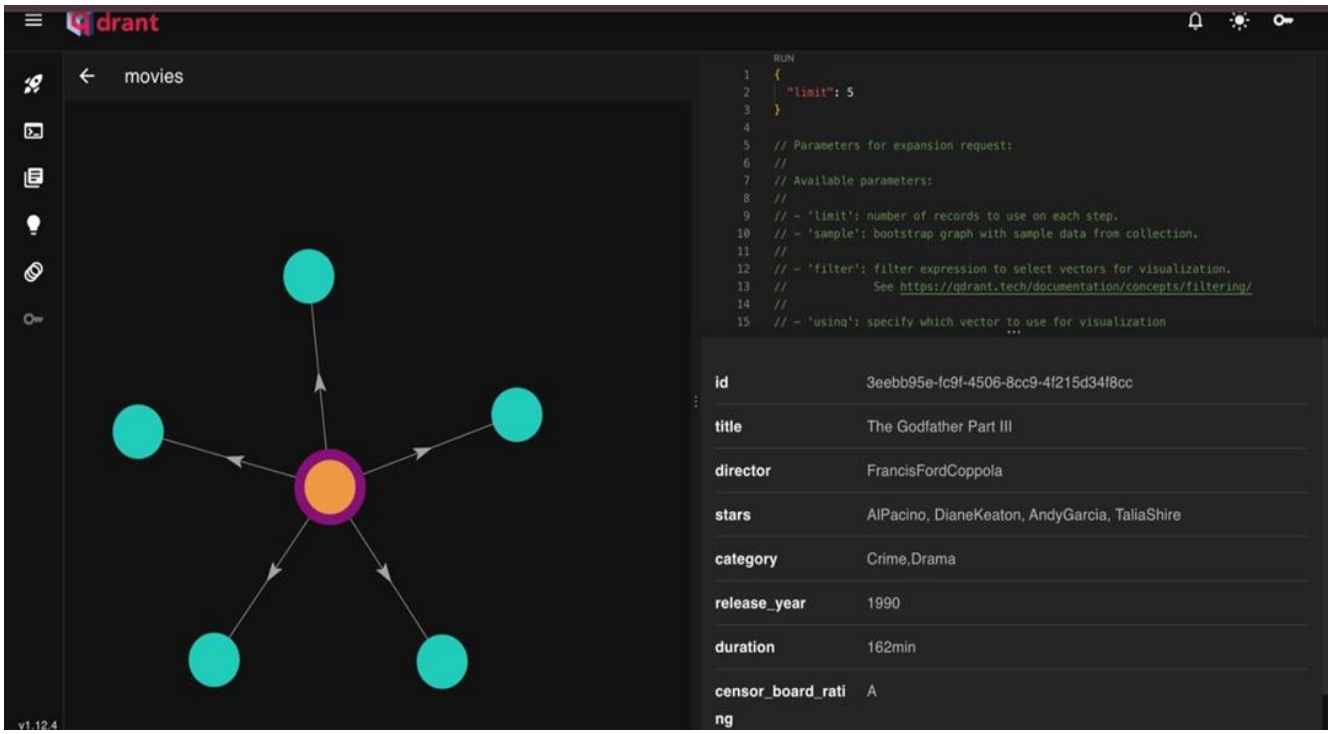
Database Integration with Qdrant

Selection Rationale:

1. Qdrant uses HNSW as its underlying ANN implementation, aligning with our findings.
2. It supports cosine similarity, enabling seamless integration of our optimal configuration.

Integration Steps:

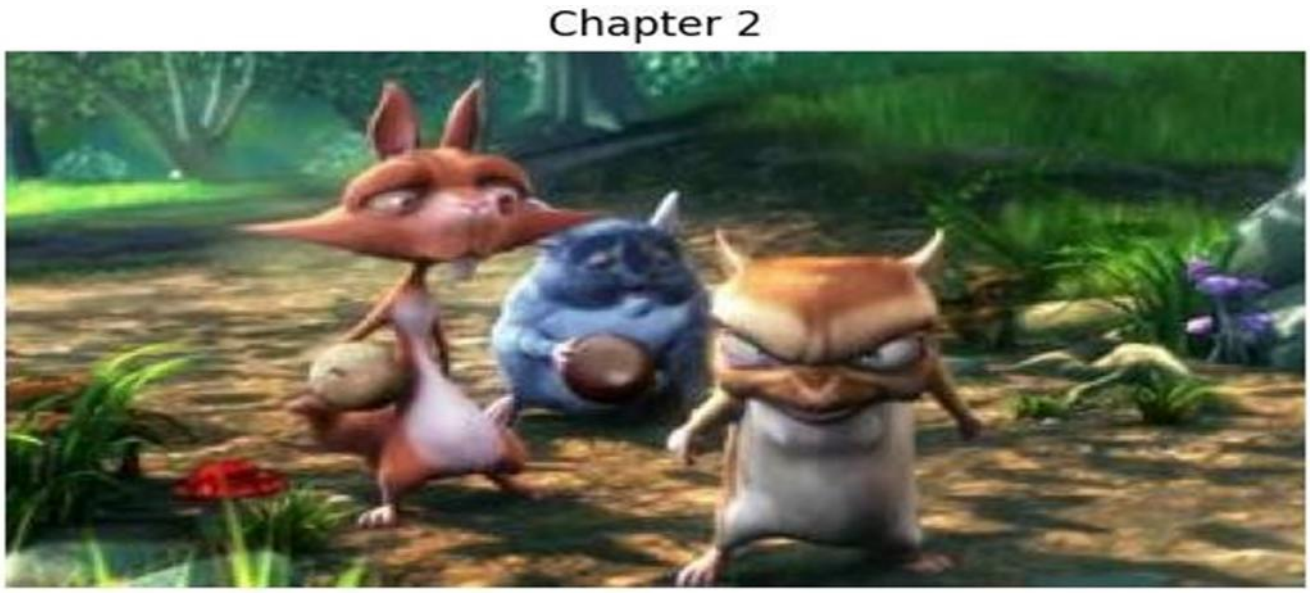
1. Setup:
    - Configured Qdrant on a cloud platform for scalability and accessibility.
    - Imported the processed IMDb dataset.
2. Query Execution:
    - Designed seed-based queries focusing on popular genres, actors, and decades.
    - Retrieved recommendations using the HNSW algorithm.
3. Performance Metrics:
    - Achieved sub-second latency for queries.
    - Validated recommendations against ground truth for accuracy.



Use of Vision Transformer (ViT) Model

Application:

ViT models were explored to extract visual embeddings for movie posters, complementing text- based embeddings. This multimodal approach enriched the recommendation system by incorporating visual semantics.



Key Highlights:

- Objective: To recommend specific chapters from a movie based on user queries, offering a personalized and engaging viewing experience.

Workflow:

1. Define Chapter Metadata: Assign attributes like genre, mood, objects, and characters to each chapter.
    - Example: chapter_0: Genre = Comedy, Mood = Playful, Objects = Butterflies, Characters = Bunny.
2. Process User Query: Input a query like "action scenes with Bunny."
3. Match Query with Metadata: Compare the query terms with chapter attributes to find relevant matches.
4. Rank and Recommend: Score chapters based on relevance and present the most suitable options to the user.

Vision Transformer (ViT) Integration:

- It provides state-of-the-art feature extraction for detailed contextual understanding of video frames.
- How It Works:
    1. Extract frames from the movie using MoviePy.
    2. Use ViT (google/vit-base-patch16-224) to generate high-dimensional feature vectors for each frame.
    3. Cluster frames into chapters using K-Means based on feature similarity.
    4. Integrate these visual features with metadata to enhance the accuracy of recommendations.

Example Use Case:

- Query: "scenes with butterflies."
- Result: ViT identifies visually similar frames containing butterflies and matches them to chapter_0 (Comedy, Playful).

Content-Based Filtering Implementation

Objective: To recommend movies by analyzing their metadata and matching them to user preferences, ensuring personalized and accurate suggestions.

***Steps Implemented**:*

1. **Feature Engineering**:
    - Extracted key metadata from the dataset, such as **genres**, **cast**, **director**, and **movie titles**.
    - Combined these features into a single text field for each movie to create a comprehensive representation.
2. **Embedding Generation**:
    - Used **Sentence Transformers (all-MiniLM-L6-v2)** to generate high-dimensional vector embeddings for each movie.
    - These embeddings captured the semantic meaning of the movie metadata, enabling accurate similarity calculations.
3. **User Profile Creation**:
    - For each user, aggregated embeddings of the movies they interacted with (liked or rated highly).
    - Calculated the average embedding of these movies to form a **user profile**, representing their preferences.
4. **Similarity Computation**:
    - Used **Cosine Similarity** to compare the user profile embedding with the embeddings of all movies in the dataset.
    - Ranked movies based on their similarity scores.
5. **Recommendations**:
    - Returned the top-n most similar movies as recommendations tailored to the user's preferences.

Collaborative Filtering Implementation

Objective: To recommend movies by leveraging patterns in user behavior and interaction history, even in the absence of rich metadata.

***Steps Implemented**:*

1. **User-Item Interaction Matrix**:
    - Constructed a matrix where rows represented users and columns represented movies.
    - The matrix values indicated interaction strength, such as ratings or implicit feedback (e.g., likes or views).
2. **Latent Factor Model**:
    - Applied **Truncated Singular Value Decomposition (SVD)** to decompose the interaction matrix into latent factors representing users and items.
    - Each user and item were represented in a lower-dimensional latent space, capturing hidden patterns and relationships.
3. **Similarity Computation**:
    - Calculated the similarity between users based on their latent features.
    - Recommended movies to a user by identifying items liked by similar users or items with similar latent features.

4. **Recommendations**:
    - Returned the top-n movies liked by users with similar behavior or those with high similarity in the latent space.

Challenges Faced

1. Data Imbalance: Addressed through oversampling and balanced query generation.
2. Integration Hurdles: Encountered compatibility issues with Qdrant's API, resolved through iterative debugging.
3. Computational Overheads: Optimized embedding generation by batching inputs and utilizing GPU acceleration.

Future Work

1. Extend the system to support real-time multimodal recommendations combining text and visual features.
2. Implement advanced post-processing techniques to refine recommendation rankings.

Conclusion

This project successfully demonstrated the transition from text-based search to vector-based retrieval using ANN algorithms. By integrating the best-performing algorithm (HNSW with Cosine Similarity) into Qdrant, we achieved significant improvements in latency and accuracy. The use of ViT models further enriched the system, paving the way for future enhancements in recommendation systems.