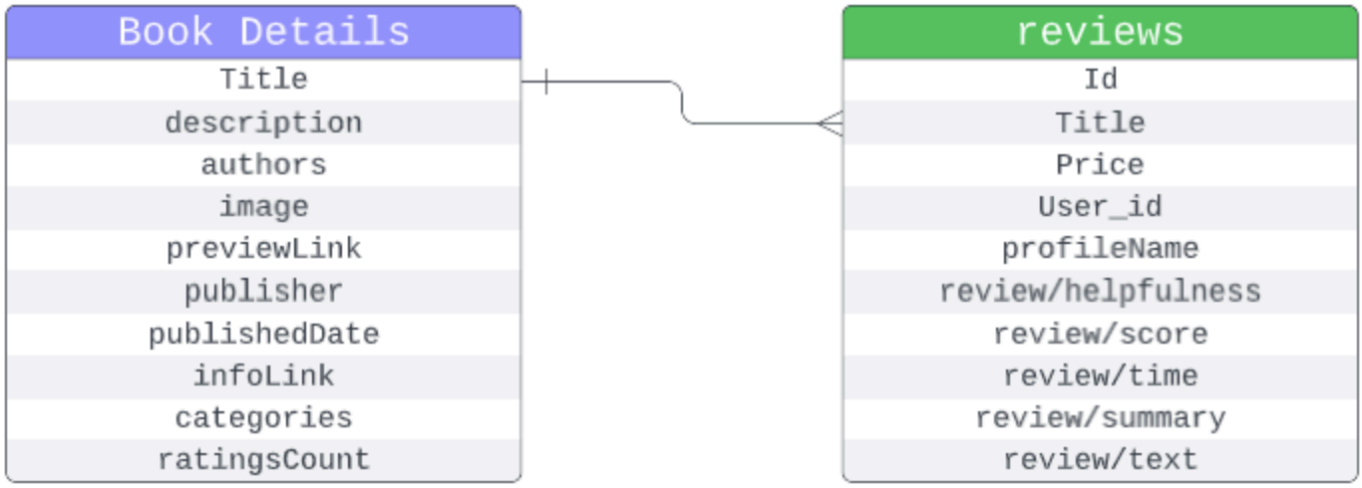
📖 My Slides 💻 My Demo

AI Book Recommendation

Team 1: Caitlin Bailey, Nina Ebensperger & Liang Gao

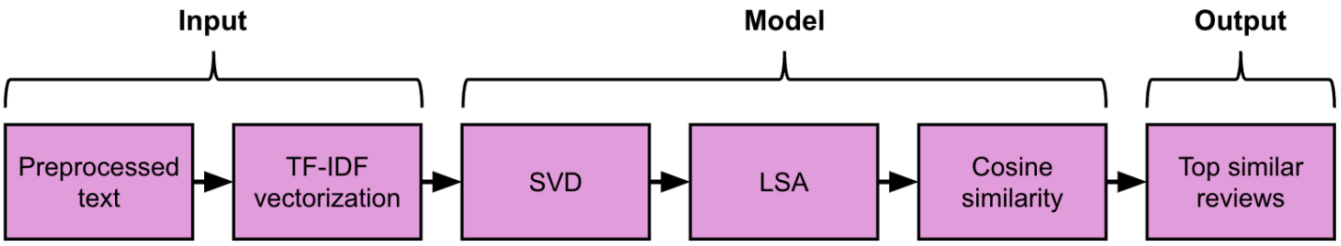
Dataset



SVD Model

SVD Model

Model Architecture



Text preprocessing: Cleaning and standardizing the text data to remove noise and irrelevant information.

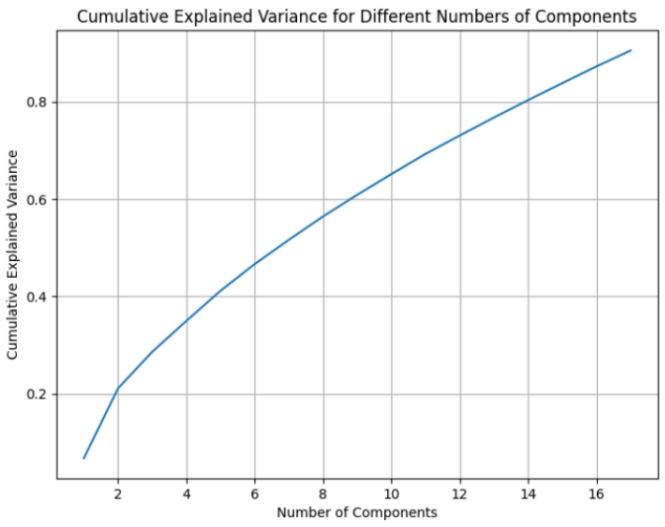
TF-IDF vectorization: Converting the text data into numerical vectors while emphasizing the importance of rare words in distinguishing documents.

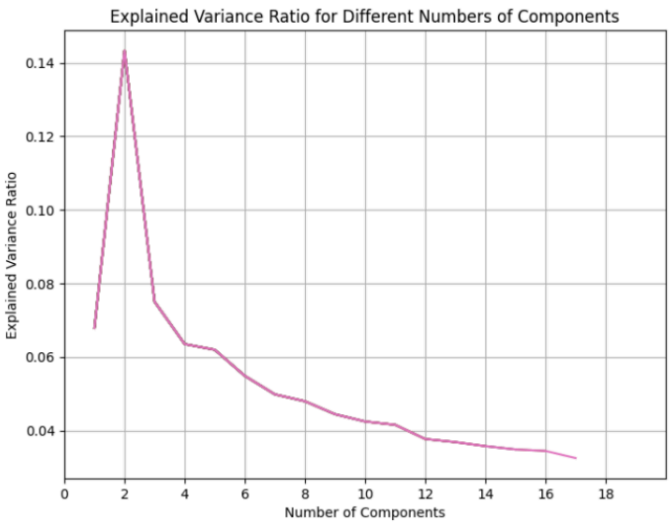
SVD (Singular Value Decomposition): Reducing the dimensionality of the TF-IDF matrix to capture latent semantic relationships.

LSA (Latent Semantic Analysis): Applying SVD to extract underlying topics or concepts from the document-term matrix.

Cosine similarity: Calculating the similarity between documents based on their vector representations.

Experiment





Result

Results of the qualitative analysis conducted through manual inspection of the model's book recommendations.

	SVD/LSA (2 components)	SVD/LSA (17 components)	SVD/LSA (20 components)
Mystery	0/10	8/10	8/10
Romance	1/10	9/10	7/10
Science Fiction	0/10	7/10	4/10
Fantasy	0/10	7/10	9/10
Thriller	1/10	8/10	6/10
Historical Fiction	1/10	7/10	8/10
Young Adult	2/10	6/10	6/10
Nonfiction	10/10	10/10	10/10

KNN Model

- **First**, we used a pre-trained **BertForSequenceClassification** model to predict the **category** based on the input review.
- **Second**, we filtered the dataframe to include only those data where the 'categories' column matches the predicted category and use **KNN** to recommend.)

Dataset preprocessing

- The observations of raw data is 3 million, about **3GB**. **Lack of GPU capacity**.
- Dropped rows with **null value**.
- Dropped categories with a count of **less than 5,000 and greater than 20,000**.
- After filtering, the dataset for later modeling has **58,199 observations**.

BertForSequenceClassification

- Tokenizatin: Apply the Pretrained BertTokenizer.
- LabelEncoder: Encoded the book categories.

- Dataset: Train 80%, Test 10% and Validation 10%.

K-Nearst-Neighbors KNN

- **Filtered** the dataframe to include only data match that predicted category.
- KNN calculates the distance between a **user's new review text** and the **existing reviews** in our dataset.
- We can effectively recommend books from our dataset that align with the user's preferences using these indices.

Post processing

- Dropped rows that have the **same review summary**.
- Dropped duplicate book that have the **same title**.

	Title	authors
4463	Pride & Prejudice (Classic Library)	Ibi Zoboi
1345	Pride & Prejudice (New Windmill)	Ibi Zoboi
7602	Pride & Prejudice (Penguin Classics)	Ibi Zoboi

- **Rank books based on review score** and recommend the top 6.

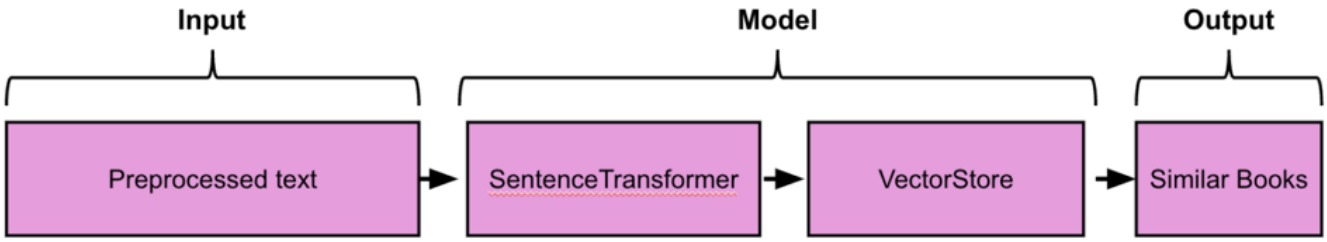
Results

- Bert classifier Test results: F1-micro at 0.84, F1-Macro at 0.83, Cohen Kappa score at 0.80.
- Manul assessment result:

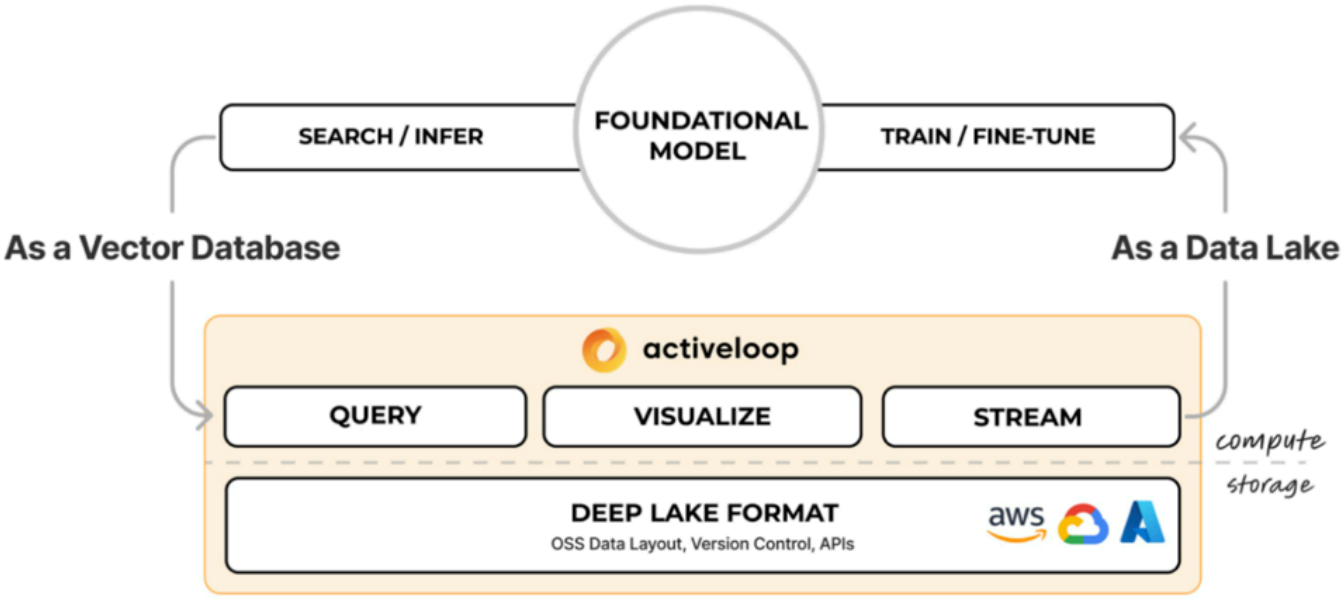
Religion	3/6
Business & Economics	5/6
Young Adult Fiction	5/6
Social Science	4/6
Philosophy	6/6
Science	1/6

DeepLake Sentence Transformer

Model Architecture



DeepLake Model Architecture



Vector Store

A specialized storage solution for handling vector embeddings.

Optimizes retrieval operations using embeddings for similarity searches.

Integration with Sentence Transformers

- Utilizes pre-trained Sentence Transformer models to generate embeddings.
- Converts text data into vector embeddings for efficient similarity matching.

Application in Book Recommendation

- Stores book descriptions as embeddings.

- Facilitates finding books with similar themes using cosine similarity searches.

Advantages of VectorStore

- Fast retrieval times optimized for high-dimensional data.
- Scalable and adaptable to various types of data including text and images.