# IMT 589 Building And Applying LLM
# Project Report

**Team Members:**

Haocheng Bao, Jiashu Chen, Sophie Deng, Hongfan Lu, Bella Wei, Dexter Xu

## I. Problem & Motivation

We are embarking on this project to harness LLM's advanced text analysis capabilities for a novel approach to movie recommendations. Traditionally, recommendation algorithms have relied on structured data, which limits their flexibility and depth. However, unstructured data, such as movie reviews, introductions, remains underutilized despite its rich potential. By leveraging LLM's sophisticated understanding of unstructured text and its extensive knowledge base, we aim to develop a more nuanced recommendation system. This innovative approach seeks to enhance user experience by providing more personalized and insightful movie suggestions.

## II. References to related projects

**A. Semantic Search:**

Similar to some other semantic search projects, we use vector database ChromaDB to store vector embeddings. There are two main differentiations. For one, our project focuses on the movie topic. We use movie synopsis as input text data. For another, we developed a simple web app to enable user interaction.

References projects:
https://medium.com/ai-science/build-semantic-search-applications-using-open-source-vector-database-chromadb-a15e9e7f14cehttps://www.kaggle.com/code/warcoder/chromadb-semantic-search

**B. Genre Classification:**

To do the genre classification using the BERT model, we referenced the paper '*Movie Genre Prediction based on the Bidirectional Encoder Representations from Transformer*' This project is similar to our project in using the BERT model to predict movie genre. The difference is that they used the movie's cover images and movie titles as predictors. To process the image they use a CNN model, ResNet-50-2 from torchvision. Their experiment had an accuracy of 34.67%, a recall of 74.5%, and a F1 score of 0.4755. Although there are still areas of improvement, this project paper gives us many insights into using the BERT model to do classification tasks. To improve the result, we tried another approach and method. We decided to use movie descriptions as a predictor to predict movie genres. It turns out that we have a better prediction accuracy of around 40%, but accuracy alone is not enough to evaluate a multi-label classification. We will talk about more details of the evaluation measures in the following sections.

References projects:
https://www.researchgate.net/publication/379004750_Movie_genre_prediction_based_on_the_bidirectional_encoder_representations_from_transformer

# III. Appropriateness of Evaluation Measures

Quantitative evaluations are calculated in the movie genre classification task. The model's performance was evaluated through a combination of overall accuracy, confusion matrix, and per-genre accuracy. Overall accuracy offers a quick snapshot of the model's performance. However, it does not capture the nuances of individual genre predictions. Therefore, it is essential to complement it with more detailed metrics.

The confusion matrix provides a detailed breakdown of true positive, true negative, false positive, and false negative predictions for each genre, helping identify specific genres that the model struggles to distinguish, enabling us to diagnose classification performance across different genres and highlight specific areas of misclassification. Additionally, per-genre accuracy calculates the accuracy for each genre individually, which is useful for identifying genres that are either consistently well-predicted or frequently misclassified. Combining these metrics ensures a comprehensive understanding of the model's strengths and weaknesses.

# IV. Quality/ Effectiveness of Outcomes or Analysis

**A. Semantic Search**

The first component of our recommendation system is based on semantic search. The objective is for users to input a description of the type of movies they are interested in (e.g., "romantic funny" or "anti-dystopian sci-fi movie"), and our database will return results that match the semantic meaning of the user's query. We have selected the open-source ChromaDB as our database infrastructure, which utilizes the lightweight "all-MiniLM-v6" model. Initially, we load the IMDb dataset into ChromaDB and convert the descriptions and overviews of the movies into embeddings. Subsequently, we transform the user's search query into an embedding. Finally, we generate results based on the cosine similarities and ranking index of the embeddings. Below is a qualitative assessment of the semantic search quality using the ChatGPT generated search query.

**Demo 1:**

Query:

"I'm looking for a heartwarming, coming-of-age drama that explores the complexities of adolescence. The movie should have a strong emotional core and focus on the growth and development of its characters. Ideally, it should be set in a small town, feature beautiful cinematography, and have a compelling storyline about friendship and self-discovery. Bonus points if it includes a subtle romantic subplot and a memorable soundtrack." (Query is ChatGPT generated)

Results:

a. {'Genre': 'Crime, Drama', 'Overview': 'A young boy becomes friends with a gang of skinheads. Friends soon become like family, and relationships will be pushed to the very limit.', 'Title': **'This Is England'**},
b. {'Genre': 'Comedy, Drama, Romance', 'Overview': "An offbeat romantic comedy about a woman who doesn't believe true love exists, and the young man who falls for her.", 'Title': **'(500) Days of Summer'** },
c. {'Genre': 'Drama', 'Overview': 'A rebellious young man with a troubled past comes to a new town, finding friends and enemies.', 'Title': **'Rebel Without a Cause**'},
d. {'Genre': 'Action, Crime, Drama', 'Overview': 'An impressionable teenage girl from a dead-end town and her older greaser boyfriend embark on a killing spree in the South Dakota badlands.', 'Title': **'Badlands**'}

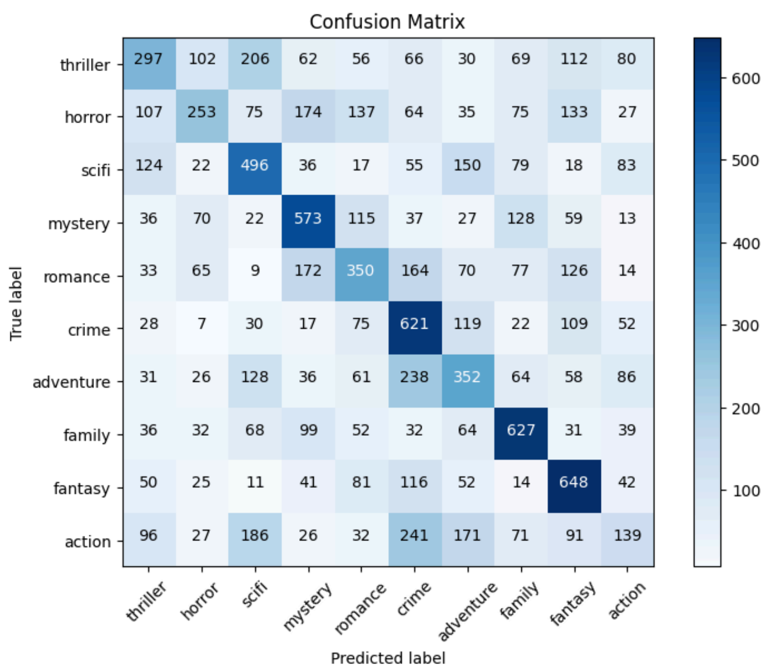
As one can observe from the search results returned above, the semantic element of "Adolescence", "Emotional Core", "Subtle romantic subplot", "Friendship & Self-discovery" and "Small-town setting" were all more or less reflected in the returned search results.

Specifically,

1. **This Is England** matches themes of coming-of-age, adolescence, and emotional core but diverges with its darker tone and setting.
2. **(500) Days of Summer** matches with emotional core and romantic subplot but is more of a romantic comedy rather than a coming-of-age drama.
3. **Rebel Without a Cause** matches well with coming-of-age, adolescence, emotional core, small town setting, and friendship/self-discovery themes.
4. **Badlands** has elements of adolescence and a small town setting but diverges significantly with its dark and violent themes.

Overall, the user's semantic intentions have obviously been registered by the LLM model and matched accordingly based on the available movie embeddings in the vector database. So from a semantic recommendation perspective, this piece of our work has achieved what we primarily envisioned. Some movies, like **Badlands** and **This is England,** have slightly darker themes rather than heart-warming ones. But It is important to remember that the limitation in the available numbers of movie vector embeddings might contribute at least in part to this, as we only have 1000 IMDb movie data loaded to our ChromaDB database infrastructure.

**B. Genre Classification**




(Fig: Confusion Matrix)

The model gives a relatively high accuracy in predicting Fantasy (60.00%) and Family (58.06%) genres, indicating that these genres have distinctive textual features that the model can reliably identify. The Crime genre also shows strong performance with an accuracy of 57.50%, suggesting that crime-related narratives are well captured by the BERT embeddings.

On the other hand, genres such as Action (12.87%), Horror (23.43%), and Thriller (27.50%) exhibit lower accuracy. This indicates that these genres are more challenging for the model to distinguish, possibly due to overlapping characteristics with each other. For example, Action movies are often confused with Thriller and Adventure due to shared elements of tension and suspense. Similarly, the Horror genre shows considerable misclassification with Thriller and Mystery, reflecting common thematic elements like fear that blur the lines between these genres.

These suggest that the effectiveness of text data as a tool to predict genres can be enhanced by incorporating additional contextual information such as metadata about the movies. Moreover, understanding misclassification patterns gives insights on how to improve models and make more nuanced recommendations. For example, grouping similar genres might better cater to user preferences.

# V. Summary

Our project utilizes the capabilities of Large Language Models (LLMs) to revolutionize the traditional movie recommendation systems. While conventional systems predominantly utilize structured data such as year, cast, director and rating, our approach focuses on exploiting the rich potential of unstructured textual data such as movie summary. In our project, we did the movie recommendation in two methods - Similarity search based on movie summary and Movie Genre Prediction.

However, although the preliminary results suggest that text-based analysis using LLMs holds considerable promise for enhancing movie recommendation systems, there are areas for improvement. Future work could include integrating additional data types, such as metadata, user behavioral data and other unstructured data such as movie reviews, to enrich the model's understanding and recommendations. By categorizing and labeling movie reviews by our model, we believe we can make our recommendations more accurate. Furthermore, exploring strategies to address misclassification, such as genre grouping or the development of more specialized models for challenging genres, could enhance accuracy and user satisfaction.

Overall, our project demonstrates a success in the application of LLMs to movie recommendation systems, offering insights into both the capabilities and limitations of current technologies.

# VI. Code link

https://github.com/Jiashu2000/llm_project
https://github.com/tomatofriedegg/semantic_search_llm
https://colab.research.google.com/drive/1YaYlLjwbJ3qYnAcGtrKC3sW0O4z4efy6