**Movie Genre Detection Using Trailers**

### **1. Genre Labelling**

I have defined a multi-label genre classification task using a genre codebook derived from the MovieLens dataset. Based on frequency and relevance in modern cinema, I selected the following seven genres:

* Action
* Comedy
* Drama
* Romance
* Horror
* Thriller
* Sci-Fi

And many other genres, each movie trailer can belong to atleast one genre. This genre classification will happen as a result of my supervised learning model I will employ in the first step of the process.

### **2. Dataset Preparation**

I utilise the Movie Lens Dataset, which maps movie IDs to YouTube trailer IDs. The process involves:

1. From the MovieLens dataset, select 25,000 movie IDs.
2. Use yt-dlp to download YouTube trailers. Ideally, at least 500 trailers are needed for meaningful training and analysis. However, for the scope of this experimental design, I will demonstrate the process using fewer trailers due to computing power constraints.
3. Extraction to separate audio and video tracks and preprocessing of the audio and video before extraction.
4. Use an off-the-shelf automatic speech recognition (ASR) model, like Google's Speech Recognition API, to transcribe audio into text.
5. Cleaning transcripts, removing symbols and applying tokenisation and lemmatisation to clean the transcripts.
6. Collect and store the following for each trailer:  
   1. Movie name 2. Genre  
   3. Cleaned transcript
7. Store the data in a CSV file with fields: movie\_name, genre, and transcript.

(The metadata associated with a trailer will not be used anywhere except in the evaluation phase for checking performance)

### **3. N-gram Analysis and Feature Engineering**

1. **Single Genre Analysis:** Analyse n-gram (n10) frequency distributions for a single genre to find common words.
2. **Multi-Genre Comparison:** Compare across genres to find distinct words per genre. This helps in filtering out overlapping terms and decreasing the amount of common words.
3. **CSV Storage:** Store all distinct words for each genre along with movie\_name, genre and cleaned transcript in the CSV.

### **4. Modelling Approach**

**Supervised Learning Phase:**

* Train a text classification model (like logistic regression) using the distinct words per genre as input features.
* Input - Cleaned transcript features (word tokens)
* Output - Multi-label genre predictions

**Unsupervised Learning Phase:**

* After the supervised step, extend the pipeline by performing unsupervised learning techniques (like clustering) on the larger dataset.
* Use transcripts from all MovieLens trailers (movie IDs > 25,000).

### **5. Data Partitioning**

Split the dataset as follows:

* 70% Training
* 30% Testing

Ensure balanced genre distribution across splits.

### **6. Performance Metrics**

1. **Model Training:**
   * Train model using cleaned, lemmatised transcripts and distinct word features.
   * Apply supervised training with early stopping and hyperparameter tuning.
2. **Model Evaluation:**

Evaluate on the test set with actual vs predicted genre labels. Metrics:

1. Accuracy (Top-k prediction accuracy)
2. Micro and Macro F1-score
3. Mean Average Precision (mAP)
4. **Tools:**

Use scikit-learn and other libraries for modelling and metric evaluation.

### **7. References**

1. Radford et al., "Robust Speech Recognition via Large‑Scale Weak Supervision," Whisper ASR, 2022.
2. Harper & Konstan, "The MovieLens Datasets: History and Context," ACM TiiS, 2015.
3. Sulun et al., "Movie Trailer Genre Classification Using Multimodal Pretrained Features," *Expert Systems with Applications*, 2024.
4. Wenfu Liu et al., "Research on Multi-label Text Classification Method Based on tALBERT-CNN," *Int. J. Comput. Intell. Syst.*, 2021.