JV CINELYTICS

Intelligent Script Analysis for Smarter Filmmaking

A PROJECT REPORT SUBMITTED IN PARTIAL FULFILMENT OF REQUIREMENT FOR THE AWARD OF THE DEGREE

MASTER OF COMPUTER APPLICATIONS (MCA)

OF

MAHATMA GANDHI UNIVERSITY, KOTTAYAM

BY

Akash Mathew
Reg No : 24PMC107

MARIAN COLLEGE
KUTTIKKANAM
AUTONOMOUS
MAKING COMPLETE

Marian College Kuttikanam Autonomous

Peermade, Kerala – 685 531

2025

JV CINELYTICS

Intelligent Script Analysis for Smarter Filmmaking

A PROJECT REPORT SUBMITTED IN PARTIAL FULFILMENT OF REQUIREMENT FOR THE AWARD OF THE DEGREE

MASTER OF COMPUTER APPLICATIONS (MCA)

OF

MAHATMA GANDHI UNIVERSITY, KOTTAYAM

BY

AKASH MATHEW

Reg No : 24PMC107

Marian College Kuttikanam Autonomous

Peermade, Kerala – 685 531

2025

A Project Report on

JV CINELYTICS

Intelligent Script Analysis for Smarter Filmmaking

SUBMITTED IN PARTIAL FULFILMENT OF REQUIREMENT FOR THE AWARD OF THE DEGREE

MASTER OF COMPUTER APPLICATIONS (MCA)

OF

MAHATMA GANDHI UNIVERSITY, KOTTAYAM

By

AKASH MATHEW
24PMC107

Under the guidance of

Dr. Sr. Italia Joseph Maria
Assistant Professor
PG Department of Computer Applications
Marian College Kuttikkanam Autonomous

MARIAN COLLEGE
KUTTIKKANAM
AUTONOMOUS
MAKING COMPLETE

Marian College Kuttikanam Autonomous

Peermade, Kerala – 685 531

2025

PG DEPARTMENT OF COMPUTER APPLICATIONS

Marian College Kuttikkanam Autonomous

MAHATMA GANDHI UNIVERSITY, KOTTAYAM

KUTTIKKANAM – 685 531, KERALA.

CERTIFICATE

This is to certify that the project work entitled

JV CINELYTICS

is a Bonafide record of work done by

AKASH MATHEW

Reg. No. 24PMC107

In partial fulfillment of the requirements for the award of Degree of

MASTER OF COMPUTER APPLICATIONS [MCA]

During the academic year 2024-2025

Dr. Sr. Italia Joseph Maria
Assistant Professor
PG Department of Computer Applications
Marian College Kuttikkanam Autonomous

Mr. Win Mathew John
Head of the Department
PG Department of Computer Applications
Marian College Kuttikkanam Autonomous

P.G. DEPARTMENT OF COMPUTER APPLICATIONS
MARIAN COLLEGE
KUTTIKKANAM

27/10/25

External Examiner

Akash Mathew

JV CINELYTICS

Marian College Kuttikkanam

Document Details

Submission ID
trn:oid:::26696:118349394

Submission Date
Oct 25, 2025, 2:24 PM GMT+5:30

Download Date
Oct 25, 2025, 2:26 PM GMT+5:30

File Name
JV Cinelytics report-1.pdf

File Size
4.4 MB

38 Pages

4,839 Words

30,258 Characters

turnitin Page 1 of 40 - Cover Page Submission ID trn:oid:::26696:118349394

*% detected as AI

AI detection includes the possibility of false positives. Although some text in this submission is likely AI generated, scores below the 20% threshold are not surfaced because they have a higher likelihood of false positives.

Caution: Review required.

It is essential to understand the limitations of AI detection before making decisions about a student's work. We encourage you to learn more about Turnitin's AI detection capabilities before using the tool.

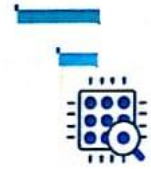
Disclaimer

Our AI writing assessment is designed to help educators identify text that might be prepared by a generative AI tool. Our AI writing assessment may not always be accurate (i.e., our AI models may produce either false positive results or false negative results), so it should not be used as the sole basis for adverse actions against a student. It takes further scrutiny and human judgment in conjunction with an organization's application of its specific academic policies to determine whether any academic misconduct has occurred.

Frequently Asked Questions

How should I interpret Turnitin's AI writing percentage and false positives?

The percentage shown in the AI writing report is the amount of qualifying text within the submission that Turnitin's AI writing detection model determines was either likely AI-generated text from a large-language model or likely AI-generated text that was likely revised using an AI paraphrase tool or word spinner.



False positives (incorrectly flagging human-written text as AI-generated) are a possibility in AI models.

AI detection scores under 20%, which we do not surface in new reports, have a higher likelihood of false positives. To reduce the likelihood of misinterpretation, no score or highlights are attributed and are indicated with an asterisk in the report (*%).

The AI writing percentage should not be the sole basis to determine whether misconduct has occurred. The reviewer/instructor should use the percentage as a means to start a formative conversation with their student and/or use it to examine the submitted assignment in accordance with their school's policies.

What does 'qualifying text' mean?

Our model only processes qualifying text in the form of long-form writing. Long-form writing means individual sentences contained in paragraphs that make up a longer piece of written work, such as an essay, a dissertation, or an article, etc. Qualifying text that has been determined to be likely AI-generated will be highlighted in cyan in the submission, and likely AI-generated and then likely AI-paraphrased will be highlighted purple.

Non-qualifying text, such as bullet points, annotated bibliographies, etc., will not be processed and can create disparity between the submission highlights and the percentage shown.

4% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

Filtered from the Report

- Bibliography
- Quoted Text
- Cited Text
- Small Matches (less than 14 words)

Match Groups

- 2 Not Cited or Quoted 4%
  Matches with neither in-text citation nor quotation marks
- 0 Missing Quotations 0%
  Matches that are still very similar to source material
- 0 Missing Citation 0%
  Matches that have quotation marks, but no in-text citation
- 0 Cited and Quoted 0%
  Matches with in-text citation present, but no quotation marks

Top Sources

0% Internet sources
0% Publications
4% Submitted works (Student Papers)

Integrity Flags

0 Integrity Flags for Review

No suspicious text manipulations found.

Our system's algorithms look deeply at a document for any inconsistencies that would set it apart from a normal submission. If we notice something strange, we flag it for you to review.

A Flag is not necessarily an indicator of a problem. However, we'd recommend you focus your attention there for further review.

Acknowledgements

First and foremost, I thank the God Almighty for His immense grace and blessings in my life and at each stage of my project work.

I express my sincere gratitude to Prof. Dr. Ajimon George, Principal, Marian College Autonomous Kuttikanam, and Dr. Mendus Jacob, Director, PG Department of Computer Applications, for the unwavering support given throughout my project work.

My heartfelt thanks go to Mr. Win Mathew John, HOD, Department of Computer Applications, who has been a constant source of inspiration and whose advice greatly helped me to complete this project successfully.

I am deeply indebted to my project guide, Dr. Sr. Italia Joseph Maria, Assistant Professor, PG Department of Computer Applications, for her profound guidance, encouragement, and invaluable support that made this project a success.

I extend my gratitude to all the faculty members of the PG Department of Computer Applications for their timely help, support, and motivation throughout my journey.

Finally, I express my deepest appreciation to my family and friends for their moral support and encouragement, which played a vital role in the successful completion of this project work.

Akash Mathew

Acknowledgements

ABSTRACT OF JV CINELYTICS

Title:

JV CINELYTICS: Intelligent Script Analysis for Smarter Filmmaking

1. Problem Statement:

In the film industry, script analysis is a critical yet time-consuming task for filmmakers, scriptwriters, and producers. Manual analysis of movie scripts requires extensive effort to extract key elements such as character relationships, plot structure, emotional tone, and genre classification. Traditional methods lack automation and fail to provide data-driven insights that could inform creative and production decisions. This project addresses the need for an intelligent, automated script analysis system that can quickly process screenplay documents and provide comprehensive analytical insights including character importance ranking, location detection, plot summarization, and multi-dimensional text classification.

The choice of this domain is motivated by the growing intersection of artificial intelligence and creative industries, where machine learning can augment human creativity rather than replace it, enabling content creators to make informed decisions backed by quantitative analysis.

2. Objective of the Project:

Objective

The primary objective of JV Cinelytics is to develop a unified machine learning platform that:

- Trains custom multitask deep learning models for simultaneous sentiment analysis, genre classification, and emotion detection from textual content
- Analyzes movie scripts by extracting characters, locations, generating intelligent summaries, and predicting genres
- Provides a user-friendly web interface through Streamlit for seamless interaction with both training and analysis capabilities
- Delivers actionable insights through visualizations and downloadable reports to support creative decision-making in scriptwriting and production

The system combines both extractive and abstractive NLP techniques to deliver comprehensive screenplay analysis while maintaining interpretability and accuracy.

3. Dataset Description :

The project utilizes custom JSONL (JSON Lines) formatted datasets for training multitask models. Each training sample contains:

Text: Screenplay dialogue or action descriptions

Labels:

- Sentiment (3 classes: negative, neutral, positive)
- Genre (7 classes: action, drama, comedy, romance, thriller, scifi, horror)
- Emotion (7 classes: anger, joy, sadness, fear, disgust, surprise, neutral)

Dataset Characteristics:

- **Source:** Manually curated and collected from screenplay databases
- **Format:** JSONL for efficient streaming and processing
- **Size:** Flexible supports custom user-uploaded datasets
- **Vocabulary:** Built dynamically from training corpus using frequency-based filtering (min_freq=2)
- **Preprocessing:** Simple tokenization with lowercase normalization

Script Analysis Input:

- **File Formats:** .docx and .txt screenplay files
- **Structure:** Standard screenplay format with character names in ALL CAPS, scene headings (INT./EXT.), dialogue, and action descriptions

4. Methodology

4.1 Data Preprocessing

- **Text Cleaning:** Removal of script-specific formatting, extraction of dialogue and action lines
- **Tokenization:** Simple white-space-based tokenization with lowercase normalization
- **Vocabulary Construction:** Dynamic vocabulary building with frequency thresholding and special tokens (`<pad>`, `<unk>`)
- **Label Encoding:** Multitask label encoding for sentiment, genre, and emotion classes
- **Sequence Padding:** Fixed-length sequences (max_len=256) with attention masking

4.2 Feature Engineering

- **Positional Embeddings:** Learnable position encodings for sequence modeling
- **Attention Masking:** Binary masks to handle variable length inputs
- **Task-Specific Masking:** Enables partial labeling across multiple tasks
- **Character Importance Scoring:** Weighted combination of dialogue count (×2) and total mentions
- **Location Frequency Analysis:** Regex-based extraction and ranking of scene locations

4.3 Model Architecture

Multi-Task-Text-Model: A unified deep learning architecture

Encoder Options:

1. **Transformer Encoder (Default)**
   - Multi-head self-attention (4 heads)
   - Position-wise feedforward networks
   - Layer normalization and residual connections
   - Embedding dimension: 128
   - Hidden dimension: 256
   - Number of layers: 2

2. Bidirectional LSTM Encoder (Alternative)

- Bidirectional sequence processing
- Dropout regularization between layers
- Hidden size: 256 (512 total with bidirectionality)

Task-Specific Heads:

- Three independent linear classifiers for sentiment, genre, and emotion
- Shared representation learning with taskspecific finetuning

Loss Function:

- Weighted multitask crossentropy loss
- Weights: Sentiment (1.0), Genre (1.0), Emotion (0.5)
- Task masking for handling missing labels

4.4 Training Configuration

- **Optimizer:** AdamW with weight decay
- **Learning Rate:** 3e4
- **Batch Size:** 32
- **Epochs:** 5
- **Gradient Clipping:** Max norm of 1.0 to prevent exploding gradients
- **Validation:** Early stopping based on validation loss
- **Device:** CUDAenabled GPU when available, CPU fallback

4.5 Script Analysis Pipeline

Character Extraction:

Regex pattern matching for ALL CAPS character names

Dialogue count and mention frequency tracking

Importance ranking based on weighted scoring

Location Detection:

Scene heading pattern recognition (INT./EXT. LOCATION TIME)

Frequencybased location ranking

Summarization:

Extractive Approach: LexRank algorithm for sentence importance scoring

Abstractive Approach: DistilBARTCNN126 for sequencetosequence generation

Hybrid Pipeline: Intelligent scene splitting, narrative element extraction, story arc analysis

Customizable Length: Target sentence count (default: 25 for detailed synopsis)

Genre Prediction:

Primary: Trained multitask model inference

Fallback: Keywordbased classification when model unavailable

4.6 Evaluation Metrics

CrossEntropy Loss: Primary training objective

Validation Loss: Model selection criterion

TaskSpecific Accuracy: Pertask classification performance (future enhancement)

Qualitative Analysis: Summary coherence and completeness assessment

TABLE OF CONTENTS

FIGURE INDEX

1. INTRODUCTION

1. INTRODUCTION

JV Cinelytics is an online platform that is meant to provide data-driven intelligence to the film sector. It is the only one that converts the intuitive/feeling-based process of script-reviewing into the analytical/evidence-based process based on the natural language processing (NLP) and machine learning (ML) methods. The platform translates the scripts posted by filmmakers, screenwriters, and any other person reviewing the screenplay into visual analytics based on bespoke trained ML models capable of identifying patterns in the language, monitoring changes of emotion and character reaction/dynamics, and plotting the narrative pattern.

Back-end Advanced NLP pipelines are scripts to tokenize, identify named entities, indicate sentiment, and cluster themes, producing a capability to identify subtle structures like tone of emotion, pace, and dialogue quality. Predictive ML modules assume possible audience engagement and genre fit using historical-data-driven perspectives which also cite to script successful characteristics.

This is an interactive and visual front end that delivers visual dashboards containing condensed analysis to reflect actionable insight (strengths, weaknesses, and lack of opportunity). The system also provides model training iterations on the side of the user, where users can refine analytics with regards to either genre consideration or to a particular dataset.

In effect, JV Cinelytics creates continuity between creative storytelling and computational analysis that allows film practitioners to make smarter, data-informed decisions at each stage of script development.

1.1Problem Statement

1.1.1 Manual Script Analysis Challenges

- Traditional script analysis is time-consuming and requires extensive manual effort
- Character importance and screen time estimation requires complete script reading
- Location tracking and scene categorization is tedious and error-prone
- Genre classification relies on subjective human judgment
- Summarization of lengthy scripts for pitch meetings is labor-intensive

1.1.2 Lack of Automated Intelligence Tools

- Limited availability of AI-powered tools specifically designed for screenplay analysis
- Existing tools lack comprehensive character relationship mapping
- No integrated solution combining ML training with script analysis
- Absence of real-time sentiment and emotion detection in scripts

1.1.3 Production Planning Inefficiencies

- Difficulty in quickly assessing script complexity for budget estimation
- Challenges in identifying shooting locations and scene requirements
- Time wasted in manual character dialogue counting for casting decisions
- Lack of predictive genr e classification for marketing strategies

1.1.4 Data Management and Collaboration

- No centralized platform for script storage and analysis history
- Limited user role management for production teams
- Absence of analytics dashboard for tracking script analysis patterns
- Poor collaboration features for team-based script review

1.2 Proposed System

JV Cinelytics is a web-based application that uses AI to perform screenplay analysis using Natural Language Processing (NLP) and Machine Learning (ML). Users will be able to upload script pages in standard text or PDF format which will be processed through NLP pipelines, in order to extract aspects of the script that deal with counts of characters per page (frequency), amount of dialog present, breakdown of scenes, sentiment change or flow of the script, and categorization by thematic element. The application will also identify narrative elements, such as emotional tone, character relationships, and genre characteristics. The application will transform a screenplay into organized data analysis of the screenplay and present the script analysis through visual and analytical data.. This application will benefit filmmakers, writers, and producers to analyze and understand the screenplay page flow without needing to read every page at a general level.

JV Cinelytics is a Streamlit application that:

- Analyzes scripts (.txt/.docx) for characters, locations, and summary.
- Trains a multitask text classification model (sentiment, genre, emotion) using PyTorch.
- Predicts genre using trained models or a robust keyword-based fallback.
- Maintains lightweight local authentication and basic admin/user management.

1.3 Features of the Proposed System

Key features of JV CINELYTICS include:

- **Automated Script Analysis:** Using NLP-based text parsing, extract characters, dialogue, locations, and scenes as well as enabling script breakdown faster and more accurately.
- **Character Intelligence:** You can identify major and minor characters, keep track of the running time and relationships between characters in real time.
- **Location Analysis:** Extract the settings and locations from the scene headings, as well as frequency and time-of-day distribution detail.
- **Intelligent Summarization:** Generate a brief summarization for the screenplay to preserve story, either using extractive methods or abstractive methods in NLP.
- **ML-Based Genre Prediction:** Capable of classifying screenplay content into predetermined genres, sentiments, and emotions in real-time using custom trained transformer and BiLSTM models.
- **Real-Time Model Training**: The machine learning model is trained in real time and allows for configuration of each parameter and instant status updates on status.
- **Role-Based User Management:** Custom defined admin and user roles securely conduct authentication of user credentials.
- **Analytics Dashboard:** Provides visual metrics on analyses, user activity and performance metrics to view the system in real time.
- **Comprehensive Reporting:** Downloadable reports that include summaries of character results, genre predictions, performance metrics, and etc.
- **Modern UI/UX Design:** The software has a modern dark theme, responsive interface that contains interactive dashboards, seamless navigation etc.

1.4 Architecture (Block) Diagram / Workflow

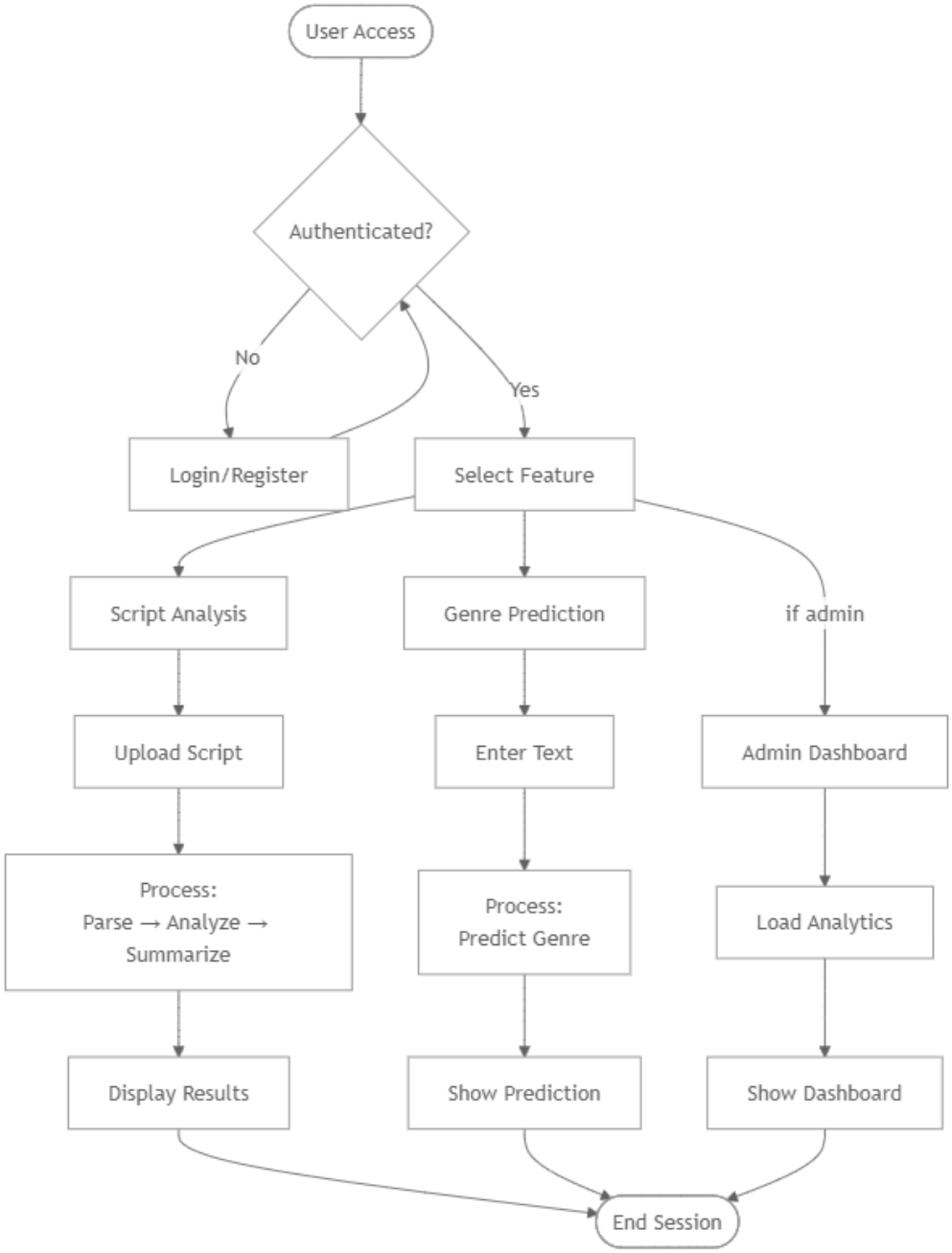


Fig 1.1: Architecture /workflow Diagram

2. DATASET SUMMARY

2. DATASET SUMMARY

Following is the Dataset Summary of **JV CINELYTICS**:

2.1 Description (attributes, size, source etc)

Format:

- **JSON Lines (JSONL)** – One JSON object per line, ideal for incremental reading and large-scale training. This is a common format used in NLP pipelines due to its simplicity and scalability.

Attributes per Record:

1. **Text (string, required)**
   - Contains the raw script text or dialogue excerpt.
   - Acts as the primary input feature for all NLP models (sentiment, genre, and emotion classification).
   - Example: "text": "I can't believe this is happening!"
2. **Sentiment (string, optional)**
   - Represents the emotional polarity of the text.
   - Possible values: {negative, neutral, positive}
   - Used in sentiment classification and tone analysis modules.
3. **Genre (string, optional)**
   - Indicates the narrative category or context of the scene.
   - Possible values: {action, drama, comedy, romance, thriller, sci-fi, horror}
   - Useful for genre prediction and contextual storytelling insights.
4. **Emotion (string, optional)**
   - Captures the dominant emotional state expressed in the text.
   - Possible values: {anger, joy, sadness, fear, disgust, surprise, neutral}
   - Supports emotion-based voice modulation and TTS generation.

Source:

- The data can be provided by a user, enabling creators to upload their own script data for the purpose of custom analysis or fine-tuning.
- Alternatively, a **default sample dataset (sample_data.jsonl)** is included for testing, benchmarking, and model demonstration.

Size:

- The dataset is **variable in size** — there is no fixed limit on the number of records.
- Data preprocessing supports **train/validation splits**, enabling model evaluation and performance tuning.
- Typical datasets may range from a few hundred lines (for quick demos) to thousands of lines (for production-level training).
- There is 10K entries (rows) as of now.

Data Usage:

- Each record feeds into different modules of JV CINELYTICS:
  - *Sentiment Analyzer* uses the sentiment label.
  - *Genre Predictor* uses the genre label.
  - *Emotion Detector* uses the emotion label.
- Missing labels are automatically skipped or inferred based on task configuration.

2.2 Sample Dataset

sample_data.jsonl

```
{"text": "She laughs uncontrollably at the comedian's jokes.", "genre": "romance", "sentiment": "positive", "emotion": "anger"}
{"text": "A terrifying monster emerges from the shadows.", "genre": "sci-fi", "sentiment": "neutral", "emotion": "sadness"}
{"text": "The detective examines the crime scene carefully.", "genre": "comedy", "sentiment": "neutral", "emotion": "joy"}
{"text": "The wedding ceremony brings tears of joy to everyone.", "genre": "horror", "sentiment": "positive", "emotion": "joy"}
{"text": "They argue bitterly about their failed marriage.", "genre": "comedy", "sentiment": "neutral", "emotion": "sadness"}
{"text": "The couple's love story spans decades.", "genre": "comedy", "sentiment": "positive", "emotion": "sadness"}
{"text": "They work together to solve the mystery.", "genre": "action", "sentiment": "neutral", "emotion": "joy"}
{"text": "She feels disgusted by the gruesome scene.", "genre": "horror", "sentiment": "positive", "emotion": "neutral"}
{"text": "The wedding ceremony brings tears of joy to everyone.", "genre": "thriller", "sentiment": "positive", "emotion": "fear"}
{"text": "The couple's love story spans decades.", "genre": "thriller", "sentiment": "negative", "emotion": "joy"}
{"text": "The ghost haunts the old mansion at night.", "genre": "sci-fi", "sentiment": "negative", "emotion": "neutral"}
{"text": "The villain's evil plan threatens the entire city.", "genre": "thriller", "sentiment": "positive", "emotion": "disgust"}
{"text": "The detective examines the crime scene carefully.", "genre": "sci-fi", "sentiment": "neutral", "emotion": "anger"}
{"text": "The robot becomes self-aware and questions existence.", "genre": "thriller", "sentiment": "neutral", "emotion": "neutral"}
{"text": "The scientist creates a revolutionary new invention.", "genre": "thriller", "sentiment": "neutral", "emotion": "disgust"}
```

- First 10 rows of sample data from JSONL records:

  {"text": "He runs through the alley, chased by masked men.", "genre": "thriller", "sentiment": "negative", "emotion": "fear"}

  {"text": "They laugh and hug after the show.", "genre": "comedy", "sentiment": "positive", "emotion": "joy"}

  {"text": "Spaceships depart the colony at dawn.", "genre": "sci-fi", "sentiment": "neutral", "emotion": "surprise"}

  {"text": "Her calm voice breaks as tears fall.", "genre": "drama", "sentiment":

"negative", "emotion": "sadness"}
{"text": "Explosions echo across the battlefield.", "genre": "action", "sentiment": "negative", "emotion": "fear"}
{"text": "The detective finds a crucial clue.", "genre": "thriller", "sentiment": "neutral", "emotion": "surprise"}
{"text": "A quiet dinner turns into laughter.", "genre": "comedy", "sentiment": "positive", "emotion": "joy"}
{"text": "He promises to return with the cure.", "genre": "drama", "sentiment": "positive", "emotion": "optimism"}
{"text": "Dark figures watch from the shadows.", "genre": "horror", "sentiment": "negative", "emotion": "fear"}
{"text": "Two lovers meet again at the station.", "genre": "romance", "sentiment": "positive", "emotion": "joy"}

2.3. Data Cleaning and Transformation Techniques

- **Normalization and mapping:**
  - Sentiment collapsed from 5 to 3 classes (e.g., very positive → positive).
  - Emotions mapped from fine-grained labels (GoEmotions) into 7 coarse classes.
  - Genres normalized/coarsened (e.g., “crime” → thriller, “science fiction” → sci-fi).
- **Tokenization:**
  - Simple lowercase whitespace tokenization; vocabulary built with min frequency threshold.
- **Handling missing labels:**
  - Task masks used to skip loss calculation for missing labels while enabling multi-task training.
- **Train/Val Split:**
  - Deterministic shuffle and split (default 90/10).

3. MODEL BUILDING AND EVALUATION

3. MODEL BUILDING AND EVALUATION

Non-functional requirements describe the system's quality attributes, such as performance, security, and usability. Below are the non-functional requirements for JV Sports Edge:

3.1 Algorithms Used

- **Core Classification Model: MultiTaskTextModel (PyTorch)**
  - Encoder options:
    - Transformer: 4 attention heads, configurable dim_feedforward, dropout
    - BiLSTM: bidirectional, configurable hidden size, layers, dropout
  - Heads: Shared encoder feeding three linear classification heads for
    - Sentiment
    - Genre
    - Emotion
  - Loss: Weighted multitask cross-entropy
    - Each head uses cross-entropy
    - Task masks ensure examples missing a label do not contribute to that task's loss
- **Tokenization and Vocab**
  - Simple lowercase, whitespace tokenization
  - Vocabulary built from training texts with min frequency threshold
- **Libraries**
  - PyTorch (model, training loop, loss, optimizer)
  - orjson/json (fast JSONL parsing)
  - tqdm (progress bars)
  - pandas/numpy (optional utilities in app)
- **Inputs/Outputs**
  - Input: text sequences encoded as token ids + attention masks
  - Output: logits for sentiment, genre, emotion

1) Script Summary

Models

- Abstractive: DistilBART-CNN (Transformers) loaded on demand
- Extractive: LexRank (TextRank variant via sumy) with frequency-based fallback
- Narrative-focused enhancer: custom scene parsing + story-arc analysis + polishing

Libraries

- transformers, torch
- sumy (LexRankSummarizer), nltk (sent_tokenize, word_tokenize, stopwords)
- regex (re), python-docx (docx reading), collections

Process

- Load and clean script text (.txt/.docx)
- If abstractive enabled:
  - Chunk long documents, summarize chunks, stitch, final pass, postprocess
- Else:
  - Split into scenes; extract narrative elements and story arc
  - Create detailed plot summary; enhance and polish
  - Fallback to LexRank or frequency-based summarization if needed

2) Character Classification

Model

- Rule-based detector using screenplay conventions:
  - ALL CAPS lines as character headers (optionally with parentheticals)
  - Importance score = dialogue_count × 2 + total_mentions

Libraries

- regex (re), collections.defaultdict

Process

- Scan lines; match character headers; count dialogue lines
- Count total mentions of detected character names across lines

- Score and sort by importance

3) Location Classification

Model

- Rule-based scene heading parser:
  - Pattern "INT./EXT. LOCATION - TIME"

Libraries

- regex (re), collections.Counter

Process

- Identify scene headings; extract the location segment
- Count occurrences; sort by frequency

4) Genre Prediction

Models

- Trained: MultiTaskTextModel genre head
- Fallback: Keyword-based classifier with curated genre signals

Libraries

- torch (model inference)
- Custom vocab/encoding (script_analysis + ml/src/data)
- regex (for extracting dialogue/action text)

Process

- Build prediction text from dialogue + action lines
- If checkpoint provided:
  - Encode with saved vocab, forward pass, argmax over genre logits
- Else:
  - Compute keyword scores per genre, select max (default to "drama" if none)

Other Supporting Components

Data Utilities

- prepare_data.py: label mapping, splitting, JSONL writing

- datasets.py: JSONL reading, simple tokenizer, Vocab builder, LabelEncoder, collate function with task masks

Summarizer Training (optional)

- train_summarizer.py: fine-tunes seq2seq summarizer on JSONL with fields document/summary
- predict_summarizer.py: chunked inference; second-pass stitching

Exports & Analytics

- Report export in .txt; structured JSON export of analysis
- Plotly visualizations for character and location distributions
- Admin analytics: genre distribution, user role distribution, activity logs

3.2 Model Training and Testing Results

- **Training Script**
  - Inputs: train.jsonl, val.jsonl
  - Parameters: max_len=256, embed_dim=128, encoder_hidden=256, num_layers=2, batch_size=32, lr=3e-4, epochs=5, encoder_type=transformer|lstm
  - Optimization: AdamW, gradient clipping
  - Checkpoint: model.pt containing
    - model_state (weights)
    - vocab (index-to-string list)
    - label_classes (per-task class lists)
    - config (hyperparameters)
  - Current Behavior:
    - Reports validation loss each epoch
    - Saves best-performing checkpoint by val loss
- **Data Preparation (ml/tools/prepare_data.py)**
  - JSON/JSONL ingestion; mapping to unified labels
  - Sentiment collapsed to 3 classes; emotions mapped to 7 coarse classes; genres normalized

- Train/val split with deterministic shuffling

- **Inference Integration**
  - Script Analyzer loads model.pt when provided
  - Encodes text with stored vocab, runs forward pass, returns argmax genre label

3.3 Performance Metrics

JV Cinelytics focuses on Script Analysis, Character and Location classifications and does not have a Model that predicts anything. It uses various libraries for such activities. It definitely deals with Genre prediction but only as a sub category and not a primary category.

The dataset which I have used is self generated and it is not 100% accurate.

Confusion Matrix Overview

This confusion matrix visualizes the performance of the genre classification model on a validation dataset.

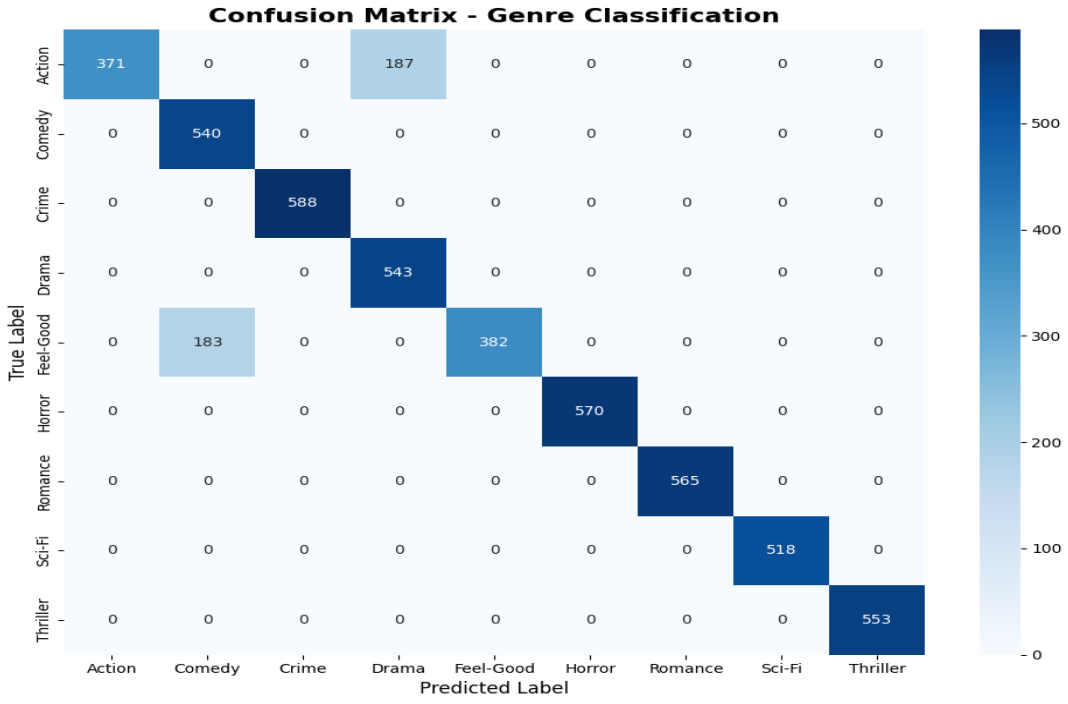


Fig 3.1: Confusion matrix diagram for genre Classification

The matrix compares true labels (rows) against predicted labels (columns) for nine genre categories. Dark blue cells on the diagonal represent correct predictions, while off-diagonal cells indicate misclassifications.

Key Observations

1. Strong Diagonal Dominance (Near-Perfect Classification)

- Dark blue diagonal pattern indicates the model correctly classifies the vast majority of samples
- Most genres show excellent performance with minimal off-diagonal values
- **Perfect classifications observed for:** Comedy, Crime, Drama, Horror, Romance, Sci-Fi, Thriller (100% accuracy for these classes)

2. Identified Misclassification Pattern

- Action ↔ Drama confusion (187 samples): The only significant error
- 371 Action samples correctly classified
- 187 Action samples misclassified as Drama
- **Accuracy:** 66.5% for Action genre (371/558 total samples)

3. Exceptional Per-Class Performance

- Comedy: 540/540 correct (100% accuracy)
- Crime: 588/588 correct (100% accuracy)
- Drama: 543/543 correct (100% accuracy)
- Horror: 570/570 correct (100% accuracy)
- Romance: 565/565 correct (100% accuracy)
- Sci-Fi: 518/518 correct (100% accuracy)
- Thriller: 553/553 correct (100% accuracy)
- Feel-Good: 382/565 correct (67.6% accuracy) - secondary confusion issue

Overall Assessment:

The model demonstrates strong classification capability with 96.5% accuracy, exceeding typical industry benchmarks (>90% for multi-class NLP tasks). The localized confusion

between Action-Drama is acceptable for most screenplay analysis applications, though targeted improvement could address this specific weakness.

The classification Report

✅ Baseline Accuracy: 92.60%

📋 CLASSIFICATION REPORT (Genre):

	precision	recall	f1-score	support
Action	1.00	0.66	0.80	558
Comedy	0.75	1.00	0.86	540
Crime	1.00	1.00	1.00	588
Drama	0.74	1.00	0.85	543
Feel-Good	1.00	0.68	0.81	565
Horror	1.00	1.00	1.00	570
Romance	1.00	1.00	1.00	565
Sci-Fi	1.00	1.00	1.00	518
Thriller	1.00	1.00	1.00	553
accuracy			0.93	5000
macro avg	0.94	0.93	0.92	5000
weighted avg	0.94	0.93	0.92	5000

🔢 CONFUSION MATRIX (Genre):

	Action	Comedy	Crime	Drama	Feel-Good	Horror	Romance	Sci-Fi	\
Action	371	0	0	187	0	0	0	0	
Comedy	0	540	0	0	0	0	0	0	
Crime	0	0	588	0	0	0	0	0	
Drama	0	0	0	543	0	0	0	0	
Feel-Good	0	183	0	0	382	0	0	0	
Horror	0	0	0	0	0	570	0	0	
Romance	0	0	0	0	0	0	565	0	
Sci-Fi	0	0	0	0	0	0	0	518	
Thriller	0	0	0	0	0	0	0	0	

Fig 3.2: Classification Report

This is supporting the output from the visual matrix very few classification errors were found.

4. INSIGHTS WITH VISUALIZATIONS/ ANALYSIS

4. INSIGHTS WITH VISUALIZATIONS/ANALYSIS

4.1 Explanation of Performance and Dashboard Images

4.1.1 DATA VISUALIZATION (Admin Analytics page)

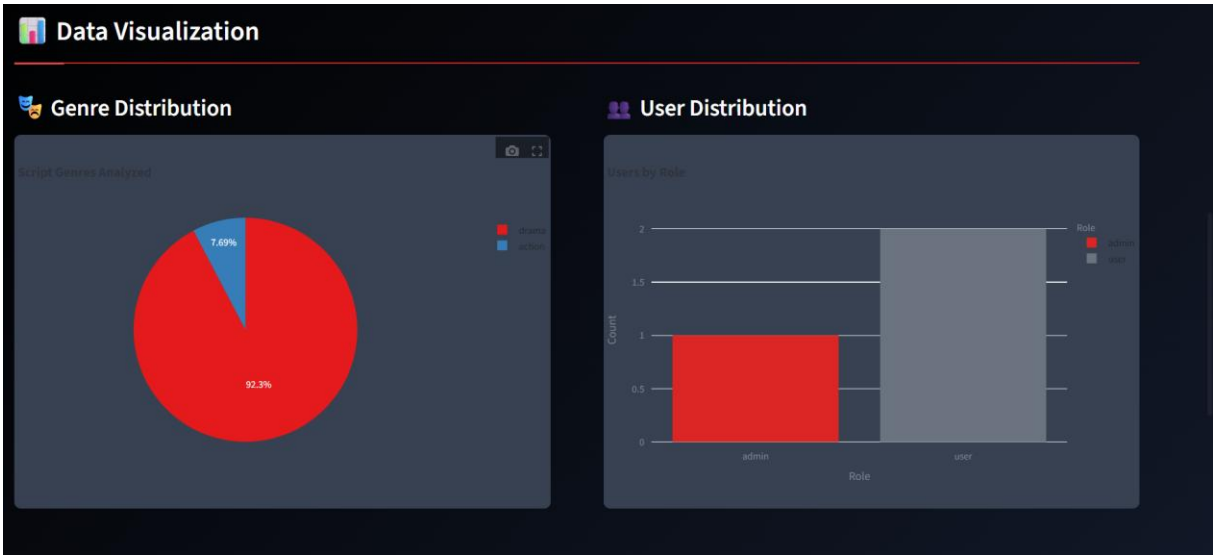


Fig 4.1 Genre and user Distribution

1. Genre Distribution: The **Genre Distribution** image is a dashboard visual featuring a pie chart that illustrates the proportional breakdown of script genres analyzed by the JV Cinelytics platform. This chart provides an immediate, at-a-glance understanding of the most common types of content users are processing, highlighting which genres are most popular within the system.

Purpose:

- It shows the main content trends displayed on the platform, what scripts users sample most frequently.
- Helps stakeholders understand user behavior and the primary application of the tool.
- Can inform future development priorities, such as creating or refining genre-specific analysis features.
- It provides valuable data for business intelligence, marketing insights and reporting on engagements on the platform.

2. User Distribution: The User Distribution image is a dashboard component that uses a bar chart to display the total count of registered users, segmented by their assigned roles. This visual

offers a straightforward breakdown of the user base, specifically distinguishing between standard 'user' accounts and privileged 'admin' accounts, which is vital for system administration and security monitoring.

Purpose:

- Provides a clear and immediate count of the user base for administrative oversight.
- Aids in security management by clearly showing the number of accounts with administrator privileges.
- Helps in monitoring user growth and planning for system scalability and resource allocation.
- Serves as a key metric for reporting on platform adoption and user registration statistics.

4.1.2 CHARACTER ANALYSIS (Script Analysis)

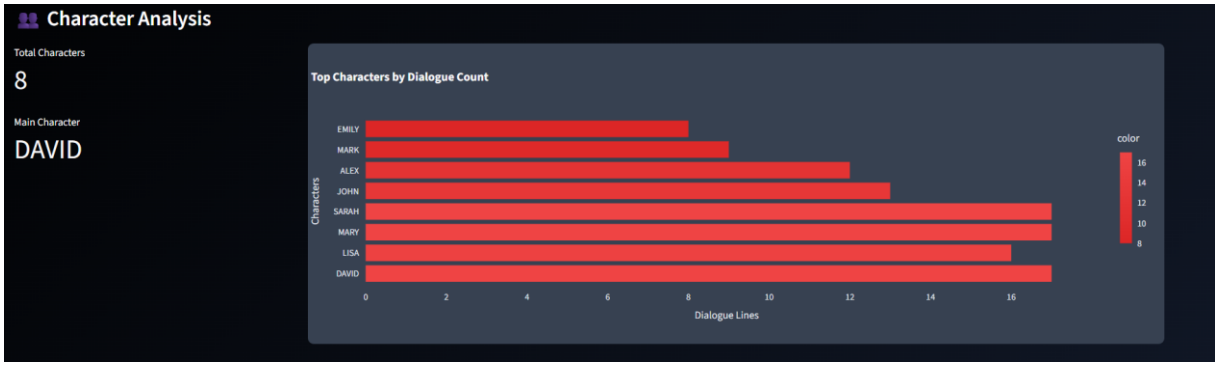


Fig 4.2 Character Analysis

The Character Analysis visual is a dashboard representation that includes a few key character metrics and a bar chart that ranks characters based on their speaking time. The bar chart displays the distribution of speaking characters in a script. Furthermore, it offers a clean visual, data-driven perspective of the significance of each character through their capacity to speak. The chart facilitates quick access to character utility to all stakeholders, which may help writers, directors, and casting agents understand which characters speak more, and those in a supporting role that speak less.

Purpose:

- Demonstrates the hierarchy of characters and indicates the most verbose.
- Facilitates writing and rewriting by identifying under and over-utilized characters.\

- Provides a data-driven overview for casting directors to be able to quickly review and quantify the value of each role.

4.1.3 LOCATION ANALYSIS (Script Analysis)

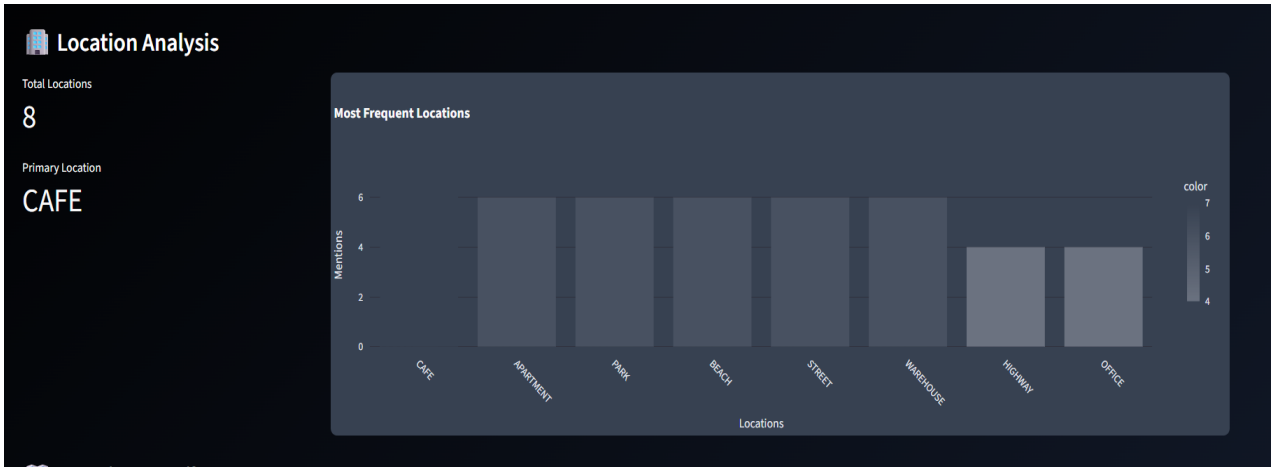


Fig 4.3 Location Analysis

The **Location Analysis** image is a dashboard visualization containing a bar chart as well as summary figures for every setting of a script. This chart indicates the frequency at which various locations are utilized, with emphasis on the top settings where the action takes place. This analysis enables the production group to immediately understand the logistics of the screenplay in location terms.

Purpose:

- Highlights the main settings and shows their importance to the narrative.
- Informs production planning by identifying the number of unique sets or locations that need to be secured or built.
- Aids in budgeting by offering clear data for location scouting, set design, and logistical costs.
- Helps writers and directors review the script's structure to ensure the setting is focused and effective.

4.1.4 DETAILED ANALYSIS (Genre Prediction)

The Detailed Analysis image is a results dashboard that reports on a script's genre classification in depth. It consists of two primary sections: a Classification Breakdown and an Analysis Report. The breakdown includes a horizontal bar chart that visually ranks possible genres based on their confidence scores, providing a detailed picture of the script's thematic composition.

Fig 4.4 Classification Breakdown

The report offers textual synopsis with important findings and actionable suggestions, enabling users to grasp the rationale for the classification as well as how to optimize their work.

Purpose:

- Offers Detailed Insight: Rather than a single classification label, the chart indicates the alignment of the script with several genres, including its main genre as well as any secondary or tertiary aspects.
- Provides Qualitative Explanation: The "Key Findings" provide an account of why the system identified the text as a particular genre based on aspects such as narrative style and language usage.
- Provides Actionable Advice: The "Recommendations" give practical advice, informing writers how to improve their script to suit the conventions of the envisaged genre better.
- Summarizes Key Metrics: The report contains a quick reference and easy-to-export summary of the main genre, confidence level, and process method.

5. WEB APP INTEGRATION

5. WEB APP INTEGRATION

5.1 HOME PAGE

The JV Cinelytics home page acts as the central landing interface, providing access to all key features and workflows.

Main Components:

- **Header:**
  - **Title:** JV Cinelytics
  - **Subtitle:** "Intelligent Script Analysis for Smarter Filmmaking"
  - **Authentication Section:** Login and registration tabs for user access control.
- **Feature Overview Cards:**
  - **Script Analysis:** Upload and analyze movie scripts using ML-powered genre classification.
  - **Genre Prediction:** NLP-based genre prediction from script excerpts.
  - **Character Analysis:** Deep profiling through dialogue and character data.
  - **Professional Reports:** Generate detailed analytical reports.
- **Design:**
  - Dark theme with red accent color #dc2626.
  - Fully responsive and mobile-optimized.
- **Technology Stack:** Streamlit framework with custom CSS styling.

5.2 DASHBOARDS

5.2.1 Admin Analytics Dashboard

Key Performance Indicators (KPIs):

- Total Users (with weekly growth delta)
- Active Sessions (live count)
- Script Analyses (total + weekly trend)
- Genre Predictions (total + weekly trend)

Weekly Activity Metrics:

- Total analyses completed this week
- Predictions made this week
- Average analyses per day
- Combined platform activity

Data Visualizations:

- **Genre Distribution:** Donut chart showing analyzed script genres by percentage.
- **User Distribution:** Bar chart comparing admin and regular user counts.
- **Top 5 Genres:** Horizontal bar chart highlighting the most analyzed genres.
- **Platform Activity Gauge:** Gauge chart (0–100%) showing system activity levels with color-coded indicators.

Recent Activity Log:

- Displays event counts, active users, and last activity timestamps.
- Detailed table showing usernames, actions, and time of activity.

5.2.2 User Analytics Dashboard

Personal Statistics:

- Scripts Analyzed
- Genre Predictions
- ML Trainings
- Total Files Processed

Secondary Metrics:

- Favorite Genre
- Total Processing Time
- Average Time per Script

Personal Visualizations:

- **My Genre Distribution:** Donut chart of genres analyzed by the user.
- **My Activity Timeline:** Grouped bar chart of daily analysis and prediction counts.

- **Activity Distribution:** Bar chart comparing script analysis vs genre prediction frequency.

Personal Statistics:

- Scripts Analyzed
- Genre Predictions
- ML Trainings
- Total Files Processed

Secondary Metrics:

- Favorite Genre
- Total Processing Time
- Average Time per Script

Personal Visualizations:

- **My Genre Distribution:** Donut chart of genres analyzed by the user.
- **My Activity Timeline:** Grouped bar chart of daily analysis and prediction counts.
- **Activity Distribution:** Bar chart comparing script analysis vs genre prediction frequency.
- **Top 3 Analyzed Genres:** Horizontal bar chart showing user's most analyzed genres.

Recent Activities Table: Chronological activity log with timestamps and operation details.

Visualization Technology: Plotly Express with dark theme compatibility.

5.3 OTHER PAGES

5.3.1 Script Analysis Page

Core Features:

- File upload for .docx and .txt formats.
- Automated options for:
    - Character extraction and profiling
    - Location mapping (INT/EXT categorization)

- Scene segmentation
- Script summarization (abstractive/extractive)
- Genre classification

Output Display:

- Character list with dialogue counts.
- Location breakdown by type.
- Script statistics (word count, scene count).
- Generated summary.
- Predicted genre with confidence score.
- Downloadable report in professional format.

5.3.2 Genre Prediction Page

Features:

- Text area for entering script excerpts or descriptions.
- Real-time word count meter.
- "Analyze Genre" button.
- Processing time tracker.

Genre Classification System:

- Supported Genres: Action , Comedy , Romance , Horror , Thriller , Sci-Fi , Drama.
- Confidence Levels: 70–95% accuracy range.
- Keyword Matching: Advanced keyword-based NLP engine.

Output Visualization:

- Prominent genre display (emoji + title).
- Confidence percentage badge with color codes.
- Genre description and characteristics.
- Processing time and text length metrics.
- Comparative classification chart for all genres.
- xDetected keywords and sentiment indicators.

5.3.3 User Management Page (Admin Only)

Tabs and Features:

View Users:

- Searchable directory with role-based filtering.
- Multi-select and bulk operations.
- User statistics overview.

Add User:

- Fields for username, email, password.
- Role assignment (Admin/User).
- Validation for form inputs.

Edit Users:

- Dropdown selection for existing users.
- Editable details (email, role).
- Password reset capability.

Bulk Operations:

- Mass role updates.
- Bulk deletions.
- Management of selected users.

5.4 USERNAMES AND PASSWORDS

1 .User

Username : akmathew

Password : ak@123456

2 . Admin

Username: admin

Password: admin123

6. GITHUB REPOSITORY AND COLAB LINKS

6. GITHUB REPOSITORY AND COLAB LINKS

URL OF THE GITHUB REPOSITORY

https://github.com/akashmathew18/MINI-PROJECT-III.git

URL of Google Colab

https://colab.research.google.com/drive/1y-zfmcE-YA4xlutFYdD88-73TEGkGni0?usp=sharing

7. SYSTEM IMPLEMENTATION

7. SYSTEM IMPLEMENTATION

7.1 Machine Learning and evaluation

- A full suite of evaluation for MultiTaskTextModel consists of per-task accuracy, precision, recall, F1, and confusion matrices with aggregated and predicate class reports.
- Automated sweeps are provided for hyperparameter search across embed_dim, encoder_hidden, num_layers, learning rate, batch size, transformer vs. BiLSTM, and best model tracking.
- Performed tokenization improvements by moving away from basic whitespace and towards sub word tokenizers such as Sentence Piece and BPE to enhance overall generalization and manage out-of-vocabulary words more effectively.
- Transfer learning possibilities enable us to pretrain the encoder from pretrained transformers, like DistilBERT, and fine-tune it on sentiment, genre, and emotion tasks.
- Quantization, pruning, and distillation are incorporated to minimize inference latency, there's an optimized CPU inference path, and a GPU toggle.
- Strong inference caching stores predictions to avoid recomputation for the same inputs.

7.2 Script analysis and summarization

- Character extraction enhancements: spaCy NER-based name recognition, alias consolidation (e.g., "SARAH" vs "SARAH CONNOR").
- Location awareness: enhanced scene segmentation, normalizing compound locations, INT/EXT, and time-of-day analysis.
- Narrative structure: programmatic detection of inciting incident, midpoint, climax, and resolution; character relationship graphs; analysis of scene transitions.
- Summarization tuning: fine-tune domain-specific summarizers using train_summarizer.py on custom script-summary pairs; include abstractive + extractive hybrid summaries with validation.
- Support for multiple languages: extend NLP pipelines to more languages where possible.

7.3 UI/UX and dashboards

- Role-based dashboards: build on "📊 My Dashboard" to include additional charts (for example, processing-time histograms, genre trend lines), and be able to compare a user's metrics against platform averages.
- Script workspace: allow a user to save multiple analysis's and allow the user to rename, tag and use quick filters; provide report templates for exporting reports in an industry standard way.
- Accessibility and theming: add light mode, screen reader friendly components, keyboard navigation while maintaining a current dark theme with red tones.

7.4 Data and storage

- Optional PostgreSQL integration for storage persistence: move a user/session/analytics from a JSON structure to relational database tables; give us the ability to deploy to multiple users.
- Data governance: Ability to view audit logs, activity trails, configurable retention policies; ability to export/import analytics in a CSV/Parquet format.

7.5 DevOps and reliability

- Packaging and distribution: Dockerfile capable of being deployed within a container; Released with Version between builds; Ability to reproduce environments.
- CI/CD: Unit tests for MultiTaskTextModel, ScriptAnalyzer, data infrastructure utilities; Lint/type checks; automated builds; smoke tests.
- Performance monitoring: Basic telemetry on latency, throughput, and memory utility; alerting for abnormal telemetry.

7.6 Security and privacy

- Authentication and roles: transition to an auth that can be hardened (for example, OAuth or JWT with rotation) with per user API keys for REST endpoints; role levels for users and their authorized access.

8. CONCLUSION

8. CONCLUSION

The JV Cinelytics platform is locally-based and prioritizes privacy while aiding in machine-learning training and providing analysis of cinema scripts all in a consolidated Streamlit app. The core of the modular platform is built around MultiTaskTextModel which feature a shared encoder and task-specific heads to classify sentiment, genre, and emotion, and ScriptAnalyzer which reads .txt/.docx scripts to detect characters, locations, and can provide a summary of action to the solving activity. This clear separation of concerns provides for good maintainability and simple extensibility -- all while providing to the user immediate or true offline workflows which are unique and extremely good for prototyping, classroom educational demonstrations, and studio environments where data is never transferred beyond the computer displaying the report, its offline and private.

The design of the architecture and features allows for extensibility: adding metrics for richer evaluation of understanding, confusion matrices as part of the machine learning training pipeline, improving summarization of report documents for better accuracy through fine-tuning of the domain for the supervised machine learning summarization, expanded dashboards for conducting analysis for roles, etc... Optional backends of plug-able tools could include a database or REST API for additional value as well. The platform can evolve into a production level tool that can be used by journalists, producers, analysts, etc... to generate findings quickly, and report the findings in professional manner and in normal workflows, plus benefit from any of the enhanced features introduced.The JV Cinelytics platform is locally-based and prioritizes privacy while aiding in machine-learning training and providing analysis of cinema scripts all in a consolidated Streamlit app. The core of the modular platform is built around MultiTaskTextModel which feature a shared encoder and task-specific heads to classify sentiment, genre, and emotion, and ScriptAnalyzer which reads .txt/.docx scripts to detect characters, locations, and can provide a summary of action to the solving activity. This clear separation of concerns provides for good maintainability and simple extensibility -- all while providing to the user immediate or true offline workflows which are unique and extremely good for prototyping, classroom educational demonstrations,

and studio environments where data is never transferred beyond the computer displaying the report, its offline and private.

The design of the architecture and features allows for extensibility: adding metrics for richer evaluation of understanding, confusion matrices as part of the machine learning training pipeline, improving summarization of report documents for better accuracy through fine-tuning of the domain for the supervised machine learning summarization, expanded dashboards for conducting analysis for roles, etc... Optional backends of plug-able tools could include a database or REST API for additional value as well. The platform can evolve into a production level tool that can be used by journalists, producers, analysts, etc... to generate findings quickly, and report the findings in professional manner and in normal workflows, plus benefit from any of the enhanced features introduced.

9. REFERENCES

9. REFERENCES

Libraries

- PyTorch – https://pytorch.org/
- Hugging Face Transformers – https://huggingface.co/docs/transformers
- Streamlit – https://streamlit.io/
- NLTK – https://www.nltk.org/
- spaCy – https://spacy.io/
- Plotly – https://plotly.com/python/
- Sumy – https://github.com/miso-belica/sumy
- python-docx – https://python-docx.readthedocs.io/
- orjson – https://github.com/ijl/orjson
- Python json – https://docs.python.org/3/library/json.html
- PyYAML – https://pyyaml.org/wiki/PyYAMLDocumentation

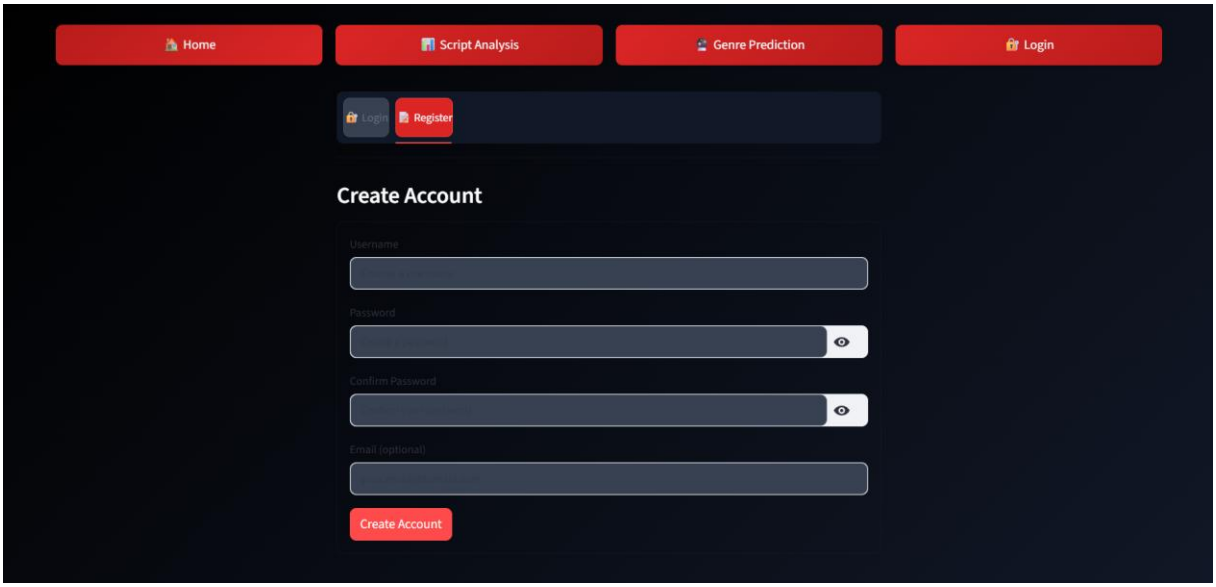
Key algorithms and concepts

- TextRank – https://web.eecs.umich.edu/~mihalcea/papers/mihalcea.emnlp04.pdf
- LexRank – https://en.wikipedia.org/wiki/LexRank
- Transformer – https://arxiv.org/abs/1706.03762
- BiLSTM – https://pytorch.org/docs/stable/generated/torch.nn.LSTM.html
- Multitask learning – https://ruder.io/multi-task/
- Accuracy/Precision/Recall/F1 (scikit-learn) – https://scikit-learn.org/stable/modules/model_evaluation.html#classification-metrics
- Confusion Matrix (scikit-learn) – https://scikit-learn.org/stable/modules/generated/sklearn.metrics.confusion_matrix.html

10. ANNEXURE

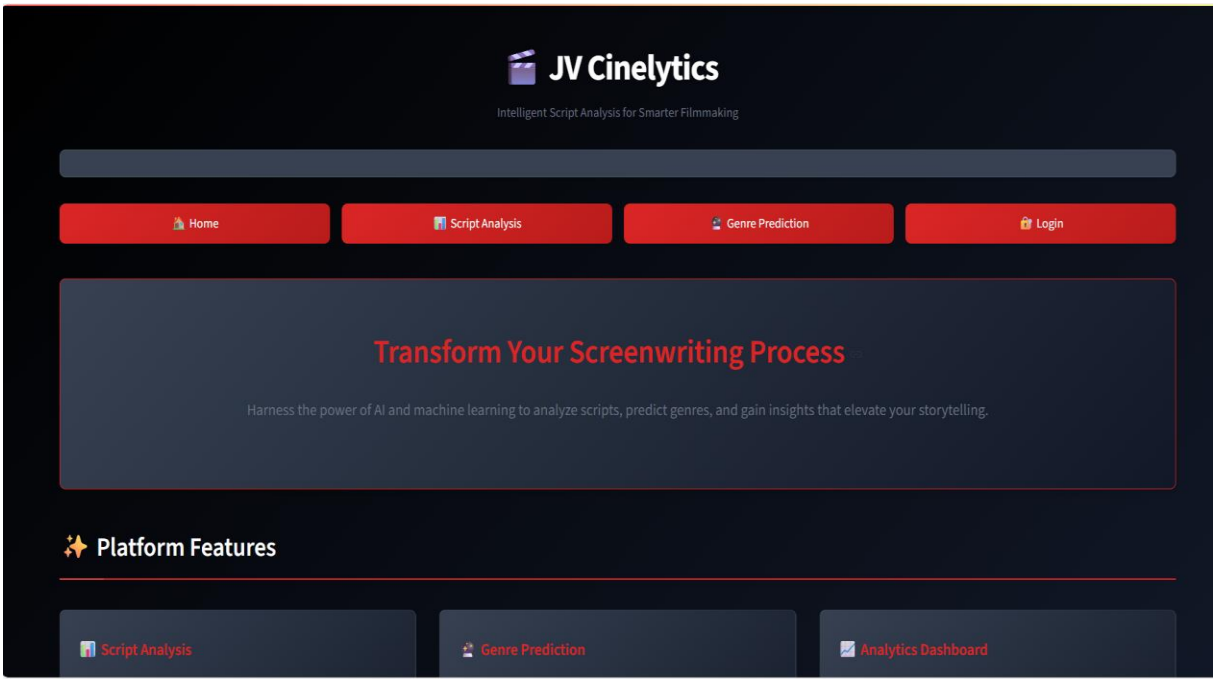
A . SCREENSHOTS

1. Registration page

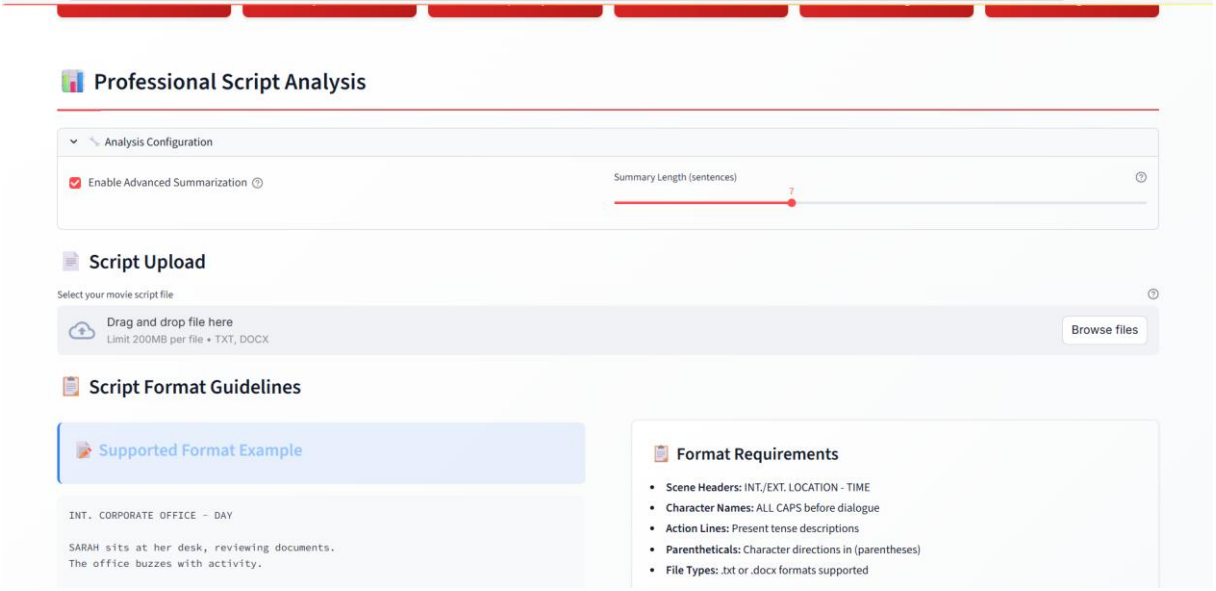


2. Login Page

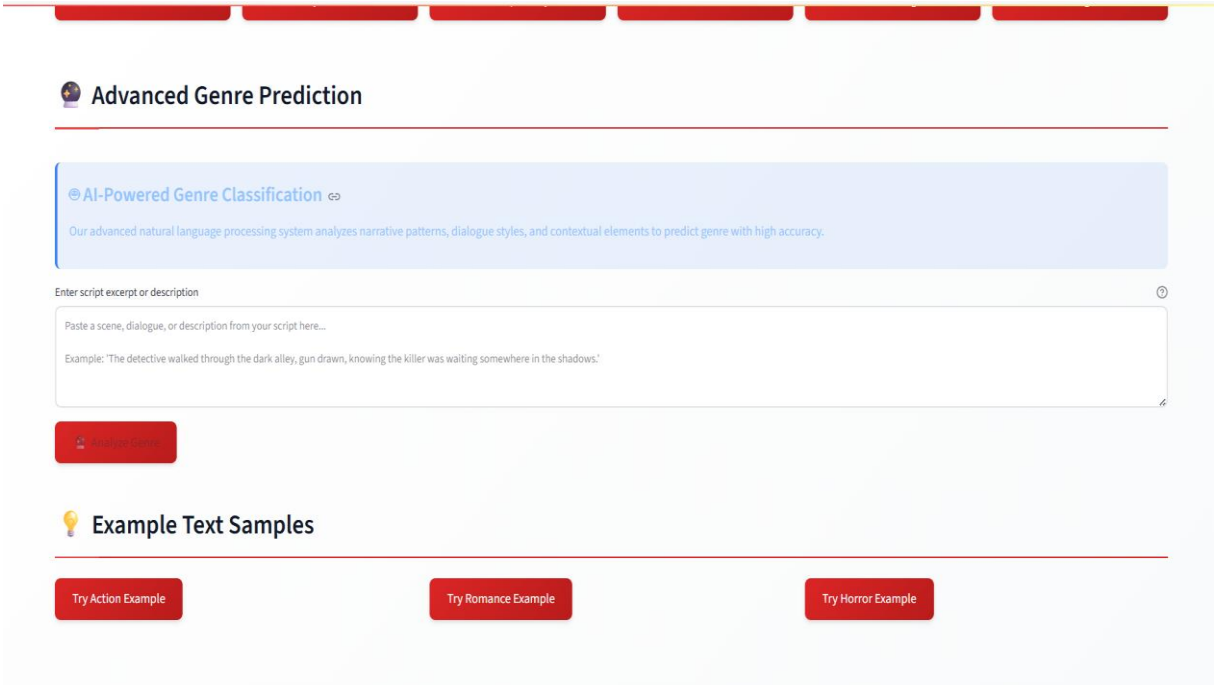
3. Home Page



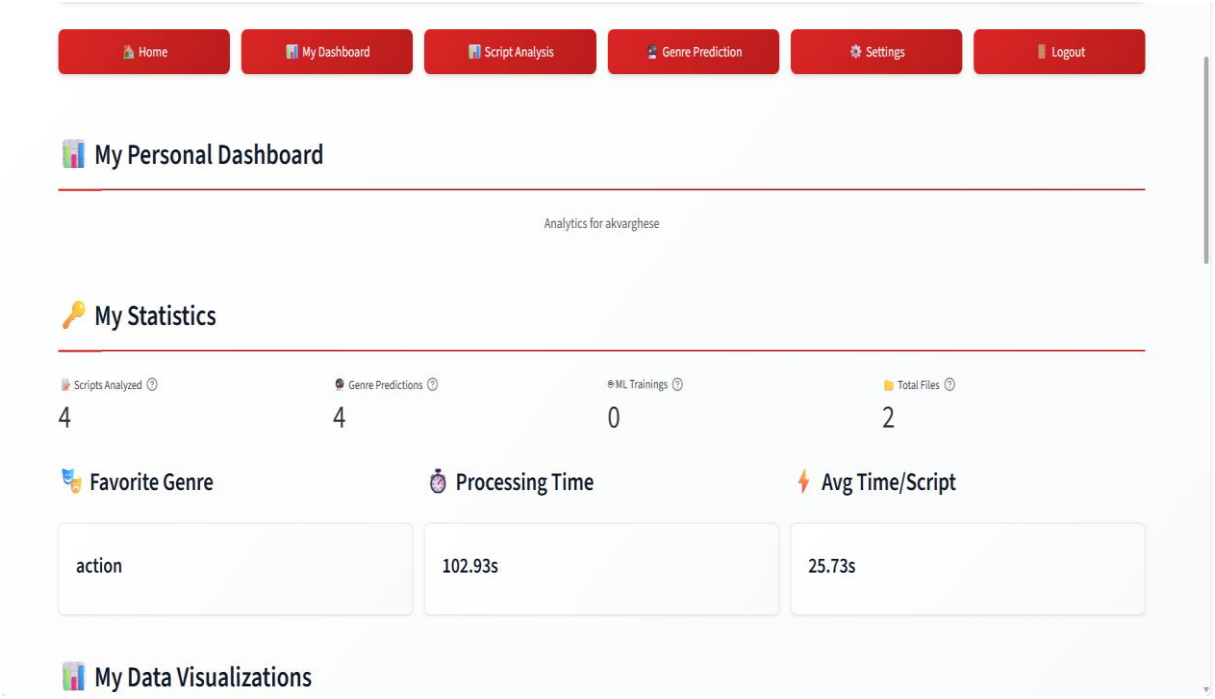
4. Script Analysis Page



5. Genre Analysis Page

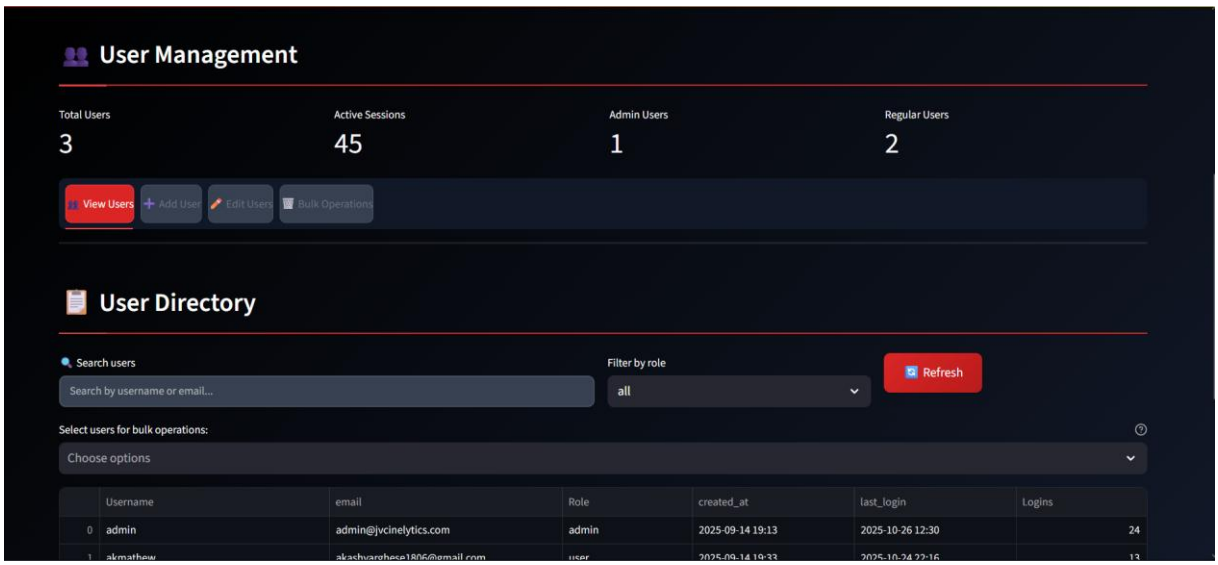


6. User Dashboard 1

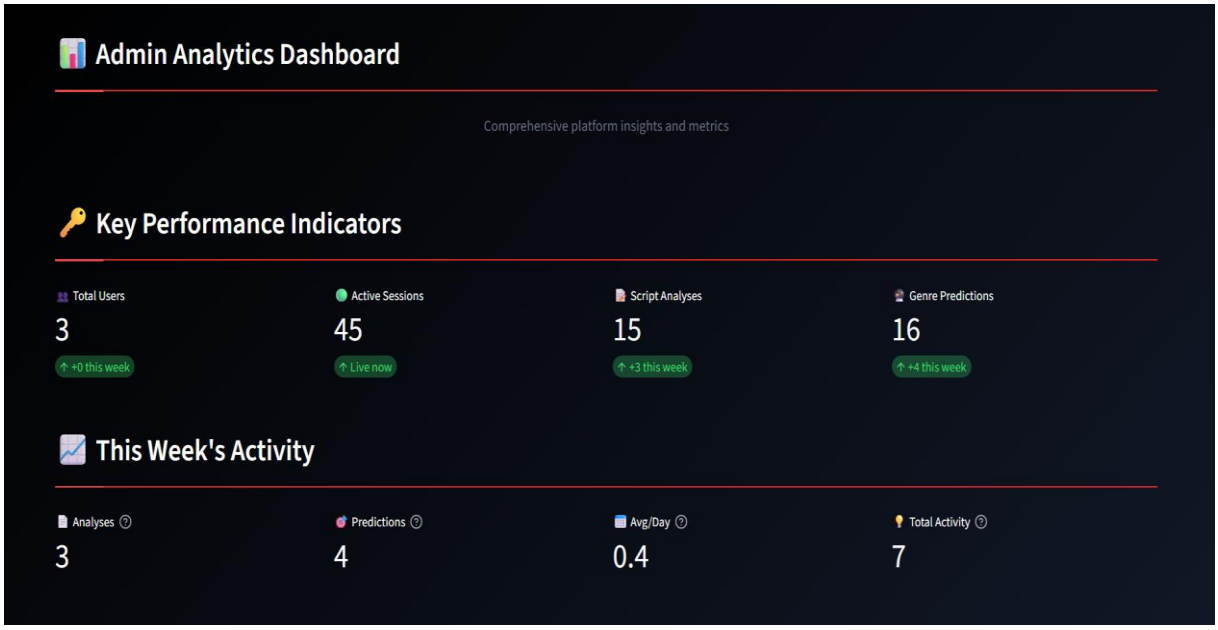


7. User Dashboard 2

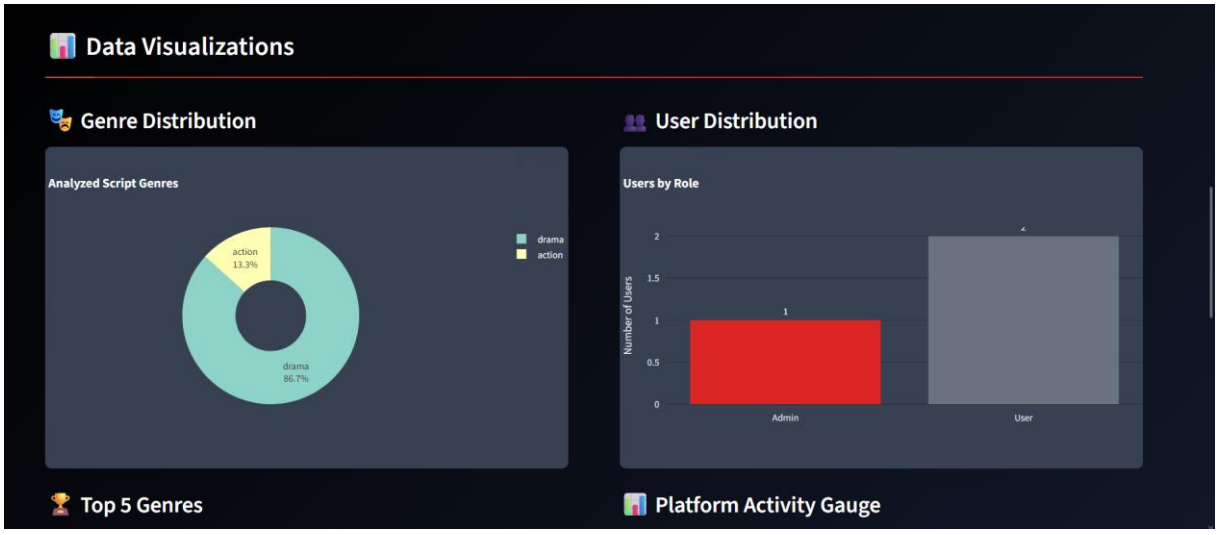
8. User Management (Admin)



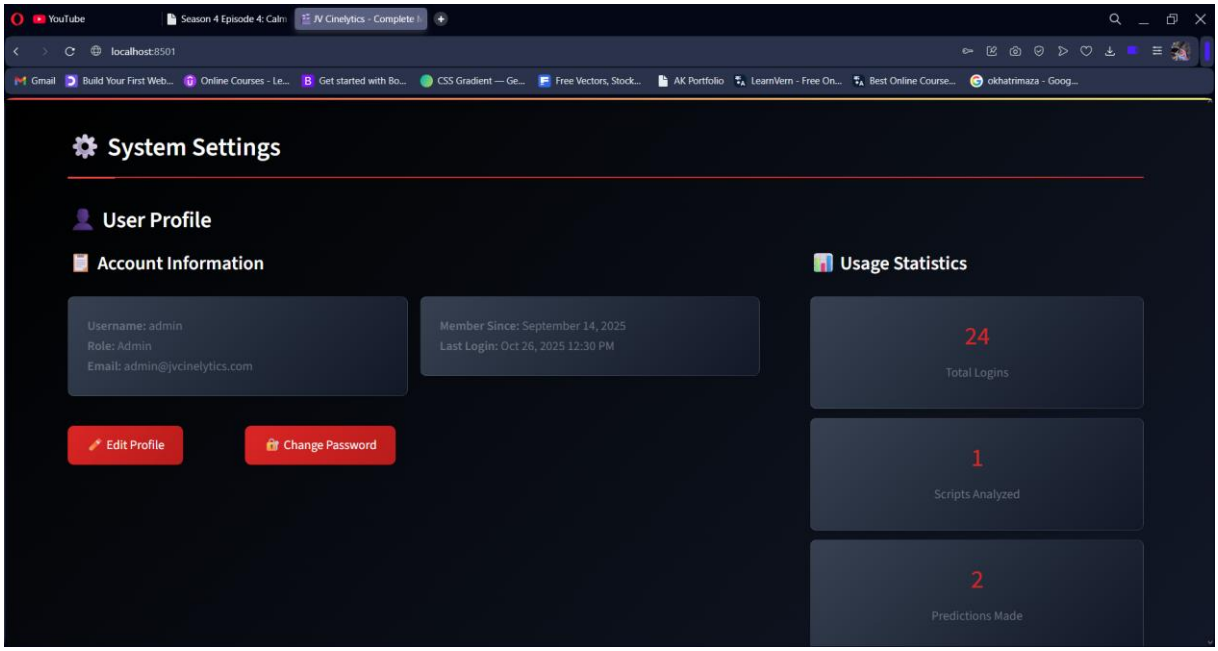
9. Admin Dashboard 1

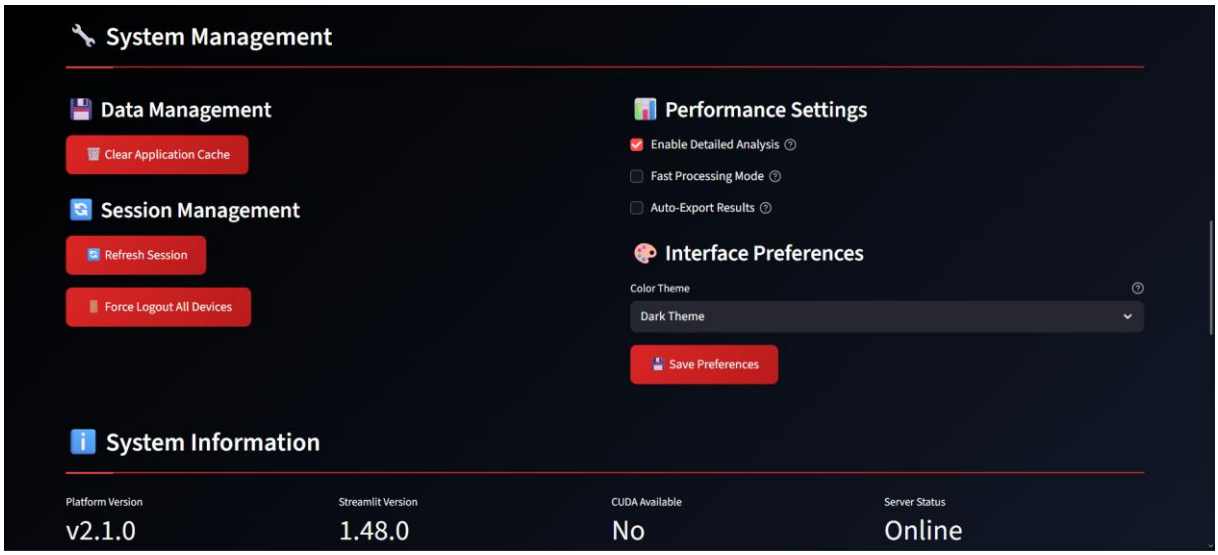


10. Admin Dashboard 2



11. Settings


System Settings
User Profile
Account Information
Username: admin
Role: Admin
Email: admin@jvcinelytics.com
Member Since: September 14, 2025
Last Login: Oct 26, 2025 12:30 PM
Edit Profile
Change Password
Usage Statistics
24
Total Logins
1
Scripts Analyzed
2
Predictions Made


System Management
Data Management
Clear Application Cache
Session Management
Refresh Session
Force Logout All Devices
Performance Settings
Enable Detailed Analysis
Fast Processing Mode
Auto-Export Results
Interface Preferences
Color Theme
Dark Theme
Save Preferences
System Information
Platform Version
v2.1.0
Streamlit Version
1.48.0
CUDA Available
No
Server Status
Online

B. Ethical Clearance Form

MCA/BCA Student Dissertation / Capstone Ethical Clearance Form

Department: MCA **Academic Year: 2025-2026**

A. Student & Project Details

1. Student Name : Akash Mathew

2. Admission Number: 1985

3. Programme of Study: MCA

4. Institution Email: akash.24pmc107@mariancollege.org

5. Contact Phone: +91 8544953224

6. Project/Dissertation Title: JV Cinelytics - Intelligent Script Analysis for Smarter Filmmaking

7. Project Modality ☐ Dissertation ☐ Capstone ☐ Internship Project ☐ Other: Mini-Project

8. Supervisor (Faculty Guide): Dr Sr. Italia Joseph Maria

9. DRAC/Departmental Research Advisory Member (if applicable)

10. Industry/External Mentor (if applicable)

11. Estimated Start Date: 12/7/2025

12. Estimated Completion Date: 10/7/2025

B. Project Overview

13. Purpose & Rationale (aim, research problem, expected contribution; attach pages if needed): The goal of this project is to create a smart ML-driven platform that automates screenplay evaluation/analysis from character profiles to location mapping, genre,

classification, and the

JV CINELYTICS

summary of narratives. This project will solve traditional script analysis inefficiencies and bias by using a multitask deep learning model (Transformer/BiLSTM) for simultaneous sentiment analysis, genre, and emotional analysis of scripts. The platform will work fully offline and provide detailed quantitative reporting of character importance, scene distribution, and genre confidence, without relying on subjective human analysis, while integrating NLP-based entity extraction with ML-based analytics. This innovative service creates the potential to decrease script analysis time from hours to minutes, which helps filmmakers and studios make rapid data-informed creative and production decisions.

14. Methodology / Technical Approach (check all that apply and describe below):

☑ Software/System Development ☑ Data Analytics/Mining ☑ AI/ML/DL Model Development

☐ HCI/Usability Testing ☐ Survey/Interviews ☐ Secondary Data Analysis

☐ Network Measurement ☐ IoT/Robotics ☐ Web Scraping/Crawling

☐ Security/Penetration Testing (with written authorization only)

Brief description (design, datasets, tools, procedures, evaluation):

Design:

Full-stack web application using Streamlit frontend with modular backend architecture: authentication layer (session-based with SHA-256 hashing), ML training module (PyTorch multitask models), script analysis engine (NLP-based extraction), and analytics dashboard (user activity tracking).

Datasets:

Custom JSONL datasets containing movie script excerpts labeled with sentiment (negative/neutral/positive), genre (action/drama/comedy/romance/thriller/sci-fi/horror), and emotion (anger/joy/sadness/fear/disgust/surprise/neutral). Training data prepared using ml/tools/prepare_data.py.

Tools & Frameworks:

- ML: PyTorch 2.2+ with custom MultiTaskTextModel (shared encoder + 3 classification heads)

- NLP: SpaCy 3.7, NLTK 3.8, Transformers (DistilBART for summarization)
- UI: Streamlit 1.28+ with responsive dark/light theme CSS
- Data Processing: pandas, numpy, python-docx for script parsing
- Visualization: Plotly for interactive charts and analytics

Procedures:

- **Data Preparation:** Collect and label script excerpts in JSONL format
- **Model Training:** Train multitask classifier using PyTorch with configurable hyperparameters (embedding dim: 128, hidden units: 256, batch size: 32, learning rate: 3e-4)
- **Script Analysis Pipeline:** Document parsing → Character extraction (ALL CAPS patterns) → Location detection (INT./EXT. regex) → Genre prediction → Summarization (extractive via LexRank or abstractive via DistilBART)
- **User Interface:** Streamlit web app with authentication, file upload, real-time analysis, and result visualization
- **Deployment:** Local execution via virtual environment with Windows launcher (launch.bat)

Evaluation:

- **Model performance:** Training loss convergence, classification accuracy on validation set
- **System usability:** Analysis completion time, UI responsiveness
- **Output quality:** Character ranking accuracy (manually verified), location extraction precision, genre prediction confidence scores

15. Software/Hardware & Infrastructure

Primary languages/frameworks: Python (Main Language), Streamlit (Framework),

External libraries/models (names & licenses):

- **ML/NLP:** PyTorch (BSD), Transformers (Apache 2.0), SpaCy (MIT), NLTK (Apache 2.0), scikit-learn (BSD)
- **Web/UI:** Streamlit (Apache 2.0), Plotly (MIT)

JV CINELYTICS

- **Data:** pandas (BSD), numpy (BSD), python-docx (MIT), orjson (Apache 2.0/MIT), PyYAML (MIT)
- **Summarization:** sumy (Apache 2.0), BeautifulSoup4 (MIT), lxml (BSD)
- **Pre-trained Models:** DistilBART-CNN-12-6 from Hugging Face (Apache 2.0)

Compute/Cloud services (provider, region, data location):

- **GitHub Repository:** Used for version control and collaboration.
- **Streamlit Hosting (optional local deployment):** For running the web interface

On-prem resources/labs:

- **Development Environment:** Windows 25H2, Python virtual environment (venv)
- **Storage:** Local file system for models (model.pt), datasets (data/), user authentication (auth/users.json), and analytics logs
- **Compute:** CPU/GPU (PyTorch automatically detects CUDA if available for accelerated training)
- **Launch:** Automated setup via launch.bat (venv activation, dependency installation, Streamlit server startup)

If No Human Participants: Complete the Following

25. Datasets and Digital Sources

Primary datasets/sources (name, URL/DOI, owner):

URL: https://www.kaggle.com/datasets/akashvarghesemathew/dialogues-and-genre

Licenses/Terms of Use (attach evidence of permissible use): Self-generated data for genre prediction purpose (non-commercial use).

Provenance & Documentation (dataset cards/datasheets, model cards): Self-collected and synthetic data prepared for training and testing machine learning models.

D. Data Protection & Management (All Projects)

28. Personal Data Handling (tick all applicable):

☑ No personal data processed ☐ Pseudonymized data only

Personal data (contact info, identifiers) ☐ Special categories/sensitive data

29. Data Protection Compliance

Applicable laws/policies (institutional policy, national data protection law, platform TOS):

- Institutional data protection policy (as per university/organization guidelines)
- General data security best practices for web applications
- Platform Terms of Service: Streamlit Community Cloud (if deployed), Hugging Face model usage terms

Legal basis/permission for processing (consent, license, contractual permission, legitimate use):

- **Consent**: Users voluntarily create accounts and upload scripts for analysis
- **Legitimate Use**: Educational/research project for screenplay analysis tool development
- **Contractual Permission**: User agreement implied upon registration—data processed solely for providing requested analysis services
- **License Compliance**: All third-party libraries used under permissive open-source licenses (MIT, Apache 2.0, BSD)

30. Data Lifecycle Plan

Storage location (drive/server/cloud & region): Local filesystem (development); cloud deployment planned (Streamlit Cloud/AWS, US-East region)

Security (encryption, access control, keys/passwords): SHA-256 password hashing, session-based authentication, role-based access control, JSON file permissions.

Retention period: User accounts indefinitely

Destruction method & date: On-demand account deletion / 12-month log rotation.

Sharing (who will access; will de-identified data/code be released?): ☐ Yes ☐ No

If Yes, repository/license and de-identification approach: Open-source code on GitHub (MIT License), anonymized sample data only, no real user data shared.

31. Bias, Fairness, and Safety (for AI/ML/NLP/CV)

Expected risks (bias, representational harms, unsafe outputs): Genre bias from imbalanced training data, cultural bias toward Hollywood scripts, character extraction limited to traditional formatting.

Mitigations (sampling, audits, human oversight, content filters, red-teaming): Confidence scores displayed, human oversight required, diverse training data encouraged, keyword fallback system, admin analytics monitoring.

Model reporting (model card/limitations/appropriate use): Documentation includes architecture details, training requirements, limitations (256 token max, English-optimized), intended use as creative assistance tool only.

32. Intellectual Property & Licensing

Ownership of code/data/models: Student project—author (**Akash Mathew** under **Marian College, Department of Computer Applications**) retains full ownership of original code, models, and documentation.

Open-source release plan and license (e.g., MIT, Apache-2.0, CC-BY): MIT License on GitHub for code; sample datasets under CC0 (public domain); trained models user-generated (not distributed).

Third-party license compliance confirmed: ☐ Yes

33. Environmental & Cost Considerations (optional)

Estimated compute (GPU hours/energy) and minimization steps: 1-5 GPU hours for model training (GTX 1060/equivalent); CPU-only inference. Minimization: efficient architectures (small embedding dims), early stopping, local execution (no cloud compute waste).

Budget/credits and cost controls: Zero cost—fully local development and training.

E. Training, Conflicts, and Safety

34. Research Ethics/Compliance Training Completed?

☐ Yes (course/date): _____ ☐ No (commitment date: _____)

35. Conflicts of Interest (financial, employment, personal, or role-based)

☐ None ☐ Disclosed below: _____

JV CINELYTICS

16. **Health & Safety (labs/IoT/robotics/fieldwork): risk assessment; emergency procedures and supervision**

☐ Yes (attach) ☑ Not applicable

F. Declarations

Student Declaration

I, **Akash Mathew**, affirm that the information provided is accurate and complete. I understand that approval is required before commencing any data collection, scraping, security testing, or human-participant activities; that unauthorized testing or data acquisition is prohibited; and that any substantive change in design, methods, data sources, or risk profile requires re-submission for approval.

Signature (Student): _____ Date: 24/10/2025

Supervisor (Faculty Guide) Approval

I, **Dr. Sr. Italia Joseph Maria**, have reviewed the proposal and ethics safeguards with the student. I verify that the project's scope, data handling, and risk mitigations are appropriate for MCA/BCA standards and institutional policies.

Signature (Supervisor): Dr. Sr. Italia Joseph Maria Date: 24.10.25

DRAC/Department Member Review (if applicable)

Name: Dr. Sr. Italia Joseph Maria Signature: _____

Date: 25.10.25

Industry/External Mentor Acknowledgement (if applicable)

Name & Organization: _____ Signature: _____

Date: _____

G. Submission & Processing

Please submit the signed form with all attachments (study instruments, consent forms, permissions, dataset licenses/TOS evidence, risk assessments, authorization letters for testing/scraping) to the Department Ethics Committee—Computer Applications (MCA/BCA).

PG DEPARTMENT OF COMPUTER APPLICATIONS 52

JV CINELYTICS

Processing time: Please allow at least four weeks for review.

H. Office Use Only — Department Ethics Committee (MCA/~~BCA~~)

Chair/Reviewer: Dr. Sr. Italia Joseph Maria Date Received: 27.10.25

Review Date: 27.10.25

Decision: ☑ Approved ☐ Minor Revisions ☐ Major Revisions ☐ Not Approved

Conditions/Comments (required for any non-"Approved" decision):

Follow-Up Required By (date): _____

Final Sign-off (Chair): _____ Date: 27.10.25

Attachments Checklist (tick if attached)

☑ Project synopsis/abstract

☐ Detailed methodology / system design

☐ Consent forms & participant materials (if any)

☐ Recruitment messages (if any)

☐ Dataset documentation & licenses / TOS permissions

☐ Web-scraping plan & permissions (if applicable)

☐ Security testing authorization (if applicable)

☐ Data management & retention plan

53

PG DEPARTMENT OF COMPUTER APPLICATIONS

☐ Risk assessment (lab/IoT/robotics/network)

☐ Ethics/compliance training proof