**Movie Genre Prediction**

**Abdullah Nawaz i21-1393**

**Adnan Nasir 21L-7596**

**M Bilal 21L-5499**

**Shais ur Rehman 21L-7734**

**Movie Genre Prediction**

**Repository link:**

https://github.com/Bilal-42/Movie-Genre-Predictor/blob/main/preprocessing%20(1).ipynb

**Description**:

The "Movie Genre from its Poster" project aims to predict the genre of a movie by analyzing its poster. By leveraging machine learning algorithms, the project explores whether color characteristics, local texture-based features, and structural cues of movie posters contain valuable information for genre prediction. The dataset consists of movie posters obtained from IMDB, along with relevant information such as IMDB ID, IMDB score, and genre labels. Through this analysis, the project seeks to determine the extent to which visual cues in movie posters can accurately predict movie genres, thus offering insights into the relationship between poster design and movie content

Dataset: <https://www.kaggle.com/datasets/neha1703/movie-genre-from-its-poster>  
Features: 6 (imdbId, Imdb Link, Title, IMDB Score, Genre, Poster)  
Instances: 40107

Overview:

* Cleaned the unwanted features.
* Created a random sample of 1000 movies to work with.
* Downloaded the images from the Poster link features.
* Split the genres.
* One-hot encoded the genres in df1.
* Label encoded the genres in df2.
* Reduced the genres to 12 and 8 classes instead of 26.
* Attempted various ML models on the two dataframes.
* Did random oversampling using class imbalance handling (it gives the most accuracy so far)

Model 1: K-NN (K Nearest Neighbours) (12 classes)

|  |  |
| --- | --- |
| **DATAFRAME 1 (ONE HOT ENCODING)** | **DATAFRAME 2 (LABEL ENCODING)** |
| Accuracy: 0.1279 | Accuracy: 0.3353 |

Model 2: CNN with 3 Convolutional Layers (12 classes)

|  |  |
| --- | --- |
| **DATAFRAME 1 (ONE HOT ENCODING)** | **DATAFRAME 2 (LABEL ENCODING)** |
| Accuracy:  Train: 0.1865  Test: 0.0000 | Accuracy:  Train: 1.0000  Test: 0.2600 |

Model 3: CNN with Two Dropout Layers (0.3, 0.7) (12 classes)

|  |  |
| --- | --- |
| **DATAFRAME 1 (ONE HOT ENCODING)** | **DATAFRAME 2 (LABEL ENCODING)** |
| Accuracy:  Train: 0.1773  Test: 0.2600 | Accuracy:  Train: 0.8214  Test: 0.1900 |

Model 4: K-NN (8 classes)

|  |
| --- |
| **DATAFRAME 1 (ONE HOT ENCODING)** |
| Accuracy: 0.4476 |

Model 5: 3 layer CNN (8 classes)

|  |
| --- |
| **DATAFRAME 1 (ONE HOT ENCODING)** |
| Accuracy:  Train: 0.3400  Test: 0.0600 |

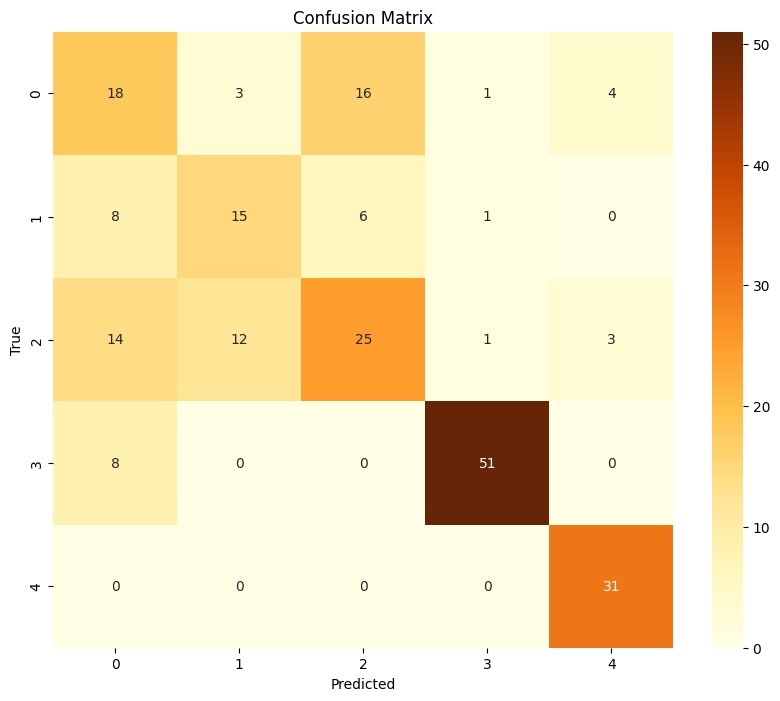
Model 6: VGG-16 (8 classes)

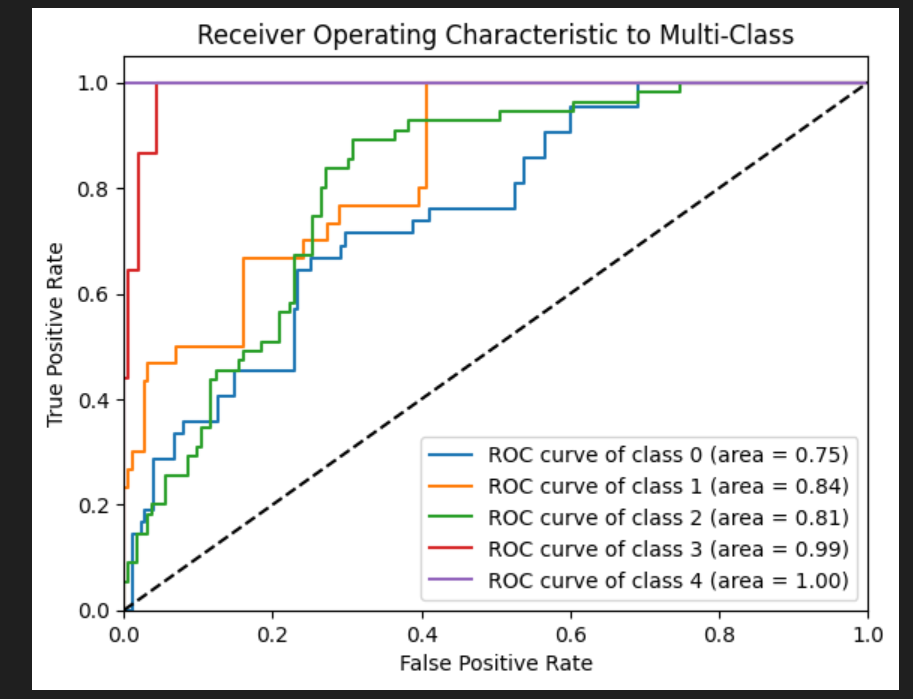
|  |
| --- |
| **DATAFRAME 1 (ONE HOT ENCODING)** |
| Accuracy:  Train: 0.6673  Test: 0.4500 |

### OPTIONAL if this doesn’t run, remove  
A close up of a sign

Description automatically generated

|  |
| --- |
| **DATAFRAME 2 (LABEL ENCODING) (5 CLASSES)** |
| Accuracy:  Train: 0.672  Test: 0.65 |
|  |





**VGG 16**

The VGG16 model is a convolutional neural network architecture renowned for its simplicity and effectiveness in image classification tasks. It features a deep structure with 16 layers, primarily comprising 3x3 convolutional filters and max-pooling layers. Its straightforward design, consisting of stacked convolutional layers followed by fully connected layers, enables powerful feature extraction and hierarchical representation learning. Despite its simplicity compared to later models like ResNet and Inception, VGG16 remains a benchmark in the field of deep learning, serving as a foundational model for various computer vision applications.