IMDb Classification Final Project Report

Introduction & Problem Statement:

Family Friendly Productions, a recent 2-year startup movie production company, saw a loss of profits in movie revenue in the past 2023 financial cycle. In strategizing for the next 2024 financial period, the CEO, CFO, and Marketing Director see potential causes to be lingering feelings from the COVID-19 as well as not capitalizing quick enough for marketing to their avid movie goers and spreading the word on their upcoming movies. As a hired Data Scientist, one pitched strategy is to design a deep learning neural network image classification model to predict the movie genre based on the IMDb movie poster data. This is to give the Marketing team more focused insights to tune their respective marketing methods(Facebook and Instagram Advertising) to target their specific audience members so that Family Friendly Productions will be positive by the end of 2024 financial year.

Data Wrangling:

In order to begin steps of model development, a training and test dataset had to be created regarding RGB values for movie poster images and their corresponding genre labels from the IMDb movie database. An initial dataset from IMDb of movie data from 1990 to 2023 for movies within just the G and PG rating were retrieved. The G and PG rating were chosen since Family Friendly Productions wanted the model to focus on movies that were generally going to be the rating the production company would serve. The movie data was retrieved through a paid IMDb API and loaded through json module and python get requests. Then the entire dataset of array of movies from 1990 to 2023 were concatenated into a dataframe. The dataset of movie data included several features such as id, image url link, respective genre label, IMDb votes, etc.

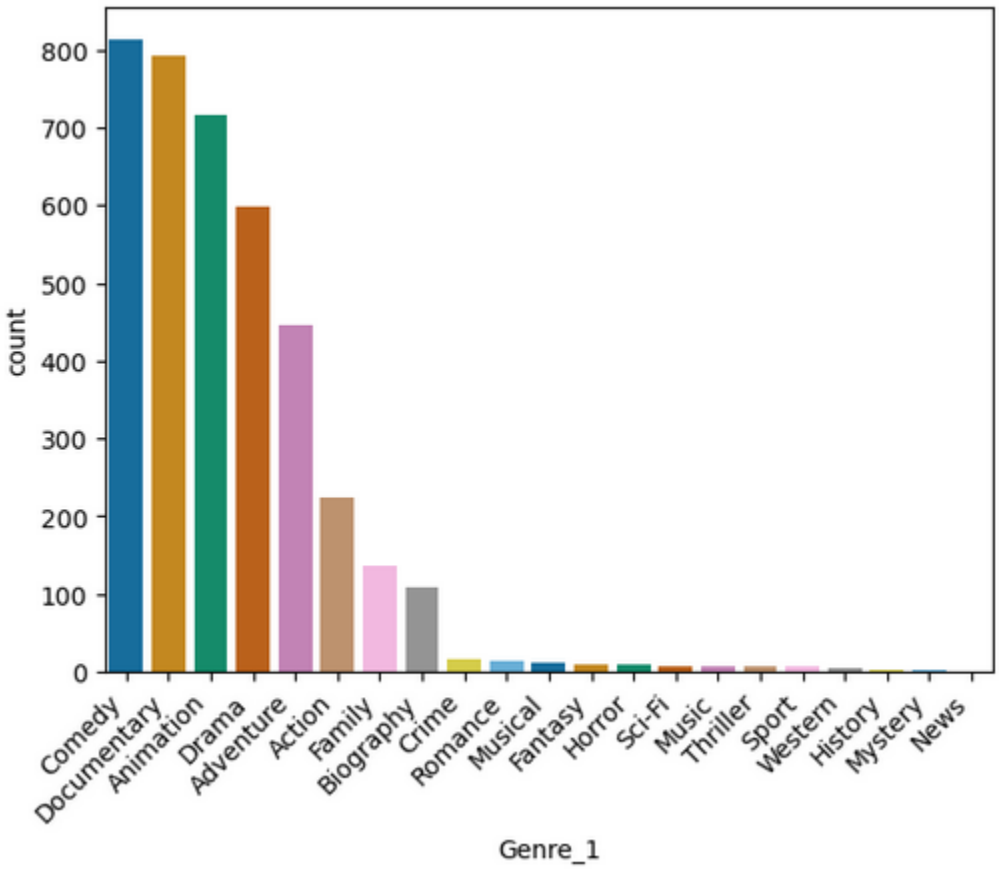
	id	image	title	description	runtimeStr	genres	genreList	contentRating	imDbRating	imDbRatingVotes	metacriticRating	plot
0	tt0099785	https://m.media-amazon.com/images/M/MV5BMzFkM2...	Home Alone	1990	103 mins	Comedy, Family	[{'key': 'Comedy', 'value': 'Comedy'}, {'key':...	PG	7.7	645257	63	An eight-year-old troublemaker, mistakenly lef...
1	tt0099810	https://m.media-amazon.com/images/M/MV5BZDdkOD...	The Hunt for Red October	1990	135 mins	Action, Adventure, Thriller	[{'key': 'Action', 'value': 'Action'}, {'key':...	PG	7.5	212075	58	In November 1984, the Soviet Union's best subm...
2	tt0100758	https://m.media-amazon.com/images/M/MV5BNzg3NT...	Teenage Mutant Ninja Turtles	1990	93 mins	Action, Adventure, Comedy	[{'key': 'Action', 'value': 'Action'}, {'key':...	PG	6.8	103577	51	Four teenage mutant ninja turtles emerge from ...
3	tt0099088	https://m.media-amazon.com/images/M/MV5BYjhlMG...	Back to the Future Part III	1990	118 mins	Adventure, Comedy, Sci-Fi	[{'key': 'Adventure', 'value': 'Adventure'}, {...	PG	7.4	475051	55	Stranded in 1955, Marty McFly learns about the...
4	tt0099422	https://m.media-amazon.com/images/M/MV5BMzA5MD...	Dick Tracy	1990	105 mins	Action, Comedy, Crime	[{'key': 'Action', 'value': 'Action'}, {'key':...	PG	6.2	65117	68	The comic strip detective finds his life vastl...

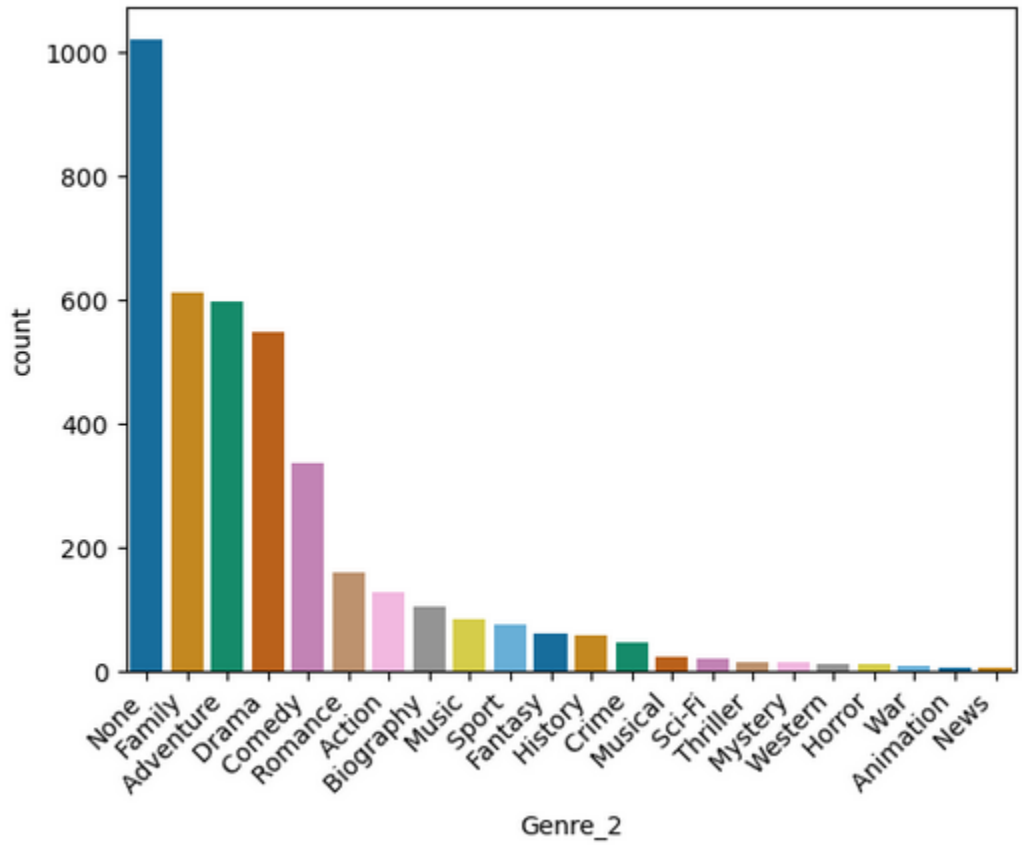
*Snapshot of IMDb dataset showing several of the rows

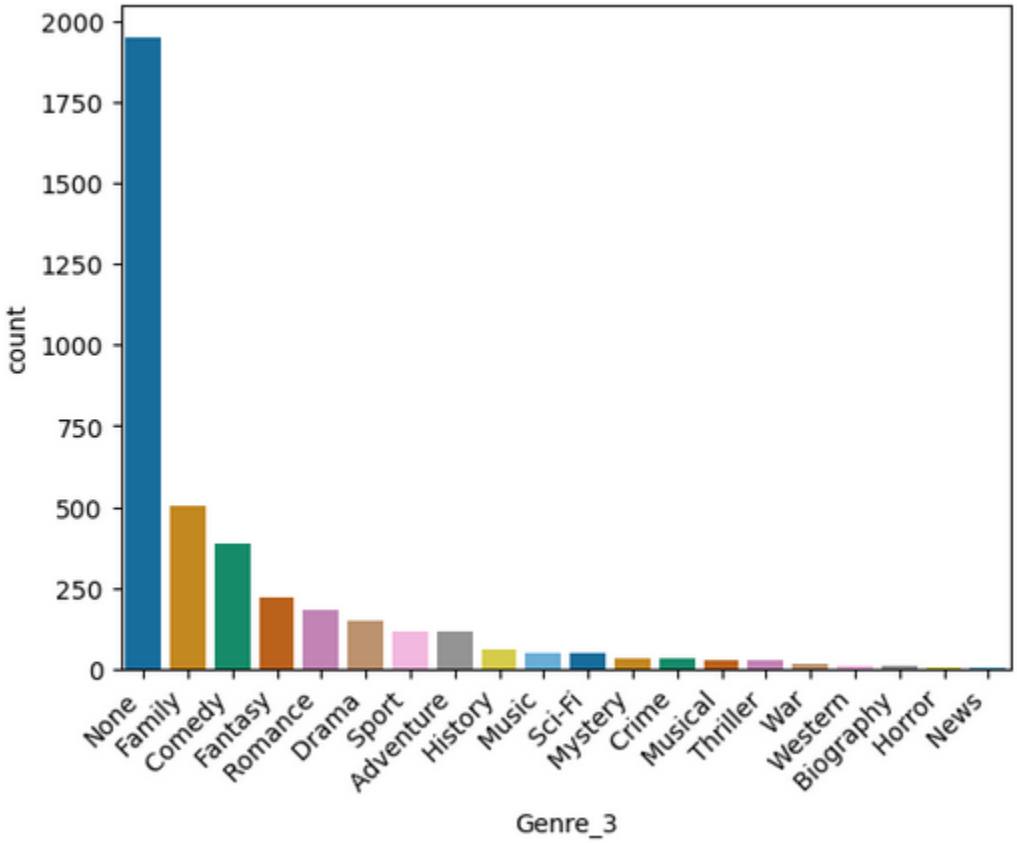
After organizing the dataset as above, the dataset was assessed for any null values within the either the “genre” or “image” column as the image classifier model needs values within both these columns to effectively be trained to make predictions. After assessing the dataset, the image column had 86 image url’s marked as null and the genre column had 45 values of missing genres for films. These null and missing values were filtered from the dataset and after the cleaning the dataset, the various movie poster images from the url links were retrieved saved on a local folder through python get requests. Finally, the different genres within the genre column of the dataset were separated as different features for a further genre class analysis in EDA. Since all movies within the IMDb dataset can have a max of three genre labels, movies with one or two genre labels were assigned with a ‘None’ category.

Exploratory Data Analysis:

The respective genre label dataset was then further assessed to determine overall class balance of the dataset for the different movie genres.




count
1000
800
600
400
200
0
None
Family
Adventure
Drama
Comedy
Romance
Action
Biography
Music
Sport
Fantasy
History
Crime
Musical
Sci-Fi
Thriller
Mystery
Western
Horror
War
Animation
News
Genre_2


count
2000
1750
1500
1250
1000
750
500
250
0
None
Family
Comedy
Fantasy
Romance
Drama
Sport
Adventure
History
Music
Sci-Fi
Mystery
Crime
Musical
Thriller
War
Western
Biography
Horror
News
Genre_3

Looking at the above plots, there is a clear class imbalance within the dataset based on the types of genres for movies being represented. The top 5 movie classes for the first genre label are Comedy, Documentaries, Animation, Drama, and Adventure. The top second class of genre labels are None, Family, Adventure, Drama, and Comedy. The top 5 third class labels are None, Family, Comedy, Fantasy, and Romance. Also, it seems that if a label within Genre_2 has a 'None' within its value, Genre_3 also has a 'None' in its value which can be seen as collinearity between the values in the dataset. It seems then that most movies have either 1 or 2 genre class labels associated with them.

Preprocessing:

To begin preprocessing for the image classification model, the different images retrieved from the url's from the initial IMDb dataset were reshaped and converted into a concatenated 200x200 pixel resolution image data array with RGB values. The genre labels for each of the images were further transformed so that instead of separating each of the genres as different labels, the respective genres themselves became features that were then one-hot encoded. This change was made to the label dataframe because the image classification model does not need to predict the exact order for the different genres of movies but just what the specific genre(s) the movie is predicted to be. Then after matching each of the label genres with the respective RGB value arrays, a training data and test set with scikit learn's train_test_split with a train test size of 80/20 was created. The training and test data were then saved as corresponding numpy arrays for RGB values and dataframes for the respective genre feature labels.

Training and Modeling:

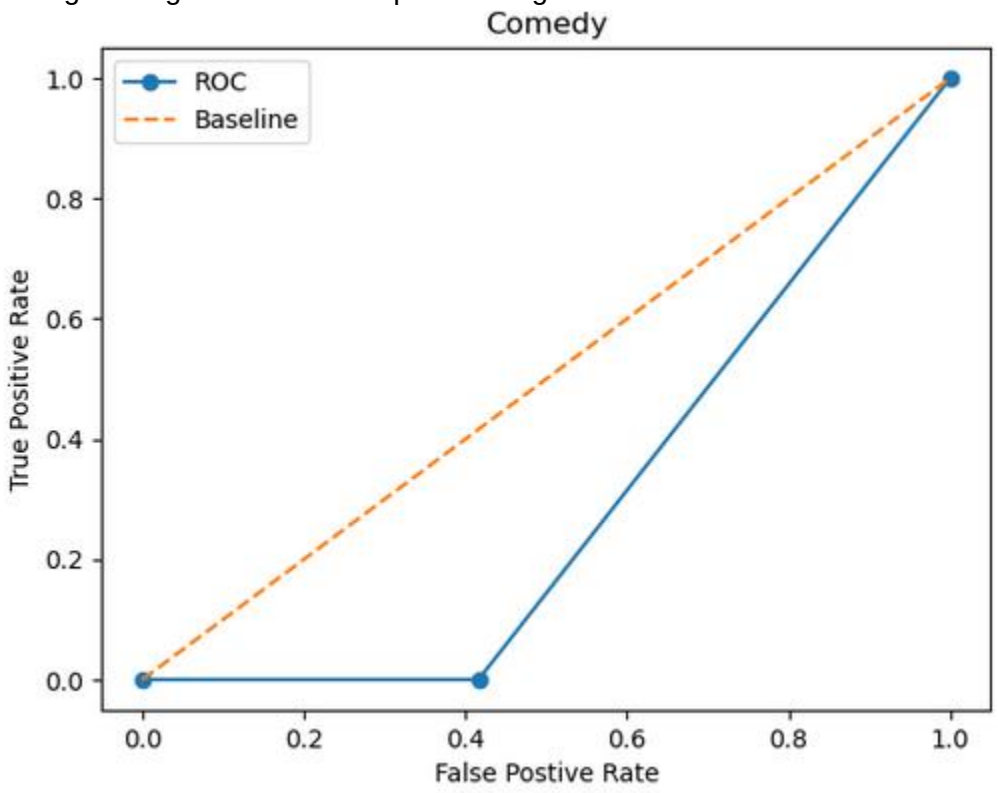
The deep learning neural network image classification model was developed with Keras with a tensorflow backend. The model includes several layers in order to take the specified input of the 200 x 200 pixel resolution movie poster image data array and provide a genre classification based on the image data. The beginning layers for the model are convolution layers with a 3x3 sliding window or kernal that searches the area of the input image to detect features of the image. Subsequent conolvtion layers were added as with each layer the model is able to detect more and more complex features of the image. The number of filters within each convolution layer were progressively decreased to prevent the model from overfitting. Then after each convolution layer a MaxPooling layer was added as this layer then takes the features that were detected in the convolution layers and selects or retains only the most important features from the search (chatGPT). After these MaxPooling layers, then a dropout layer is added to assist in preventing overfitting for the model as this layer effectively 'drops out' or deactivates certain neurons during model training. Finally dense layers were added at the end of the model so that the model can further learn the features of the image after being flattened (turned from three dimensional to one dimension) after the convolution layers, ending finally at a classification to classify the image into 21 different outputs or genres to be predicted. The mentioned structure is then compiled for the model to be developed where the compiler provides the optimizer or what function the model has to update weights in training, loss function which provides the difference between the predicted and true output, and finally metrics to assess the model during training. The metrics chosen to asses model performance was the model's f1_score because f1_score

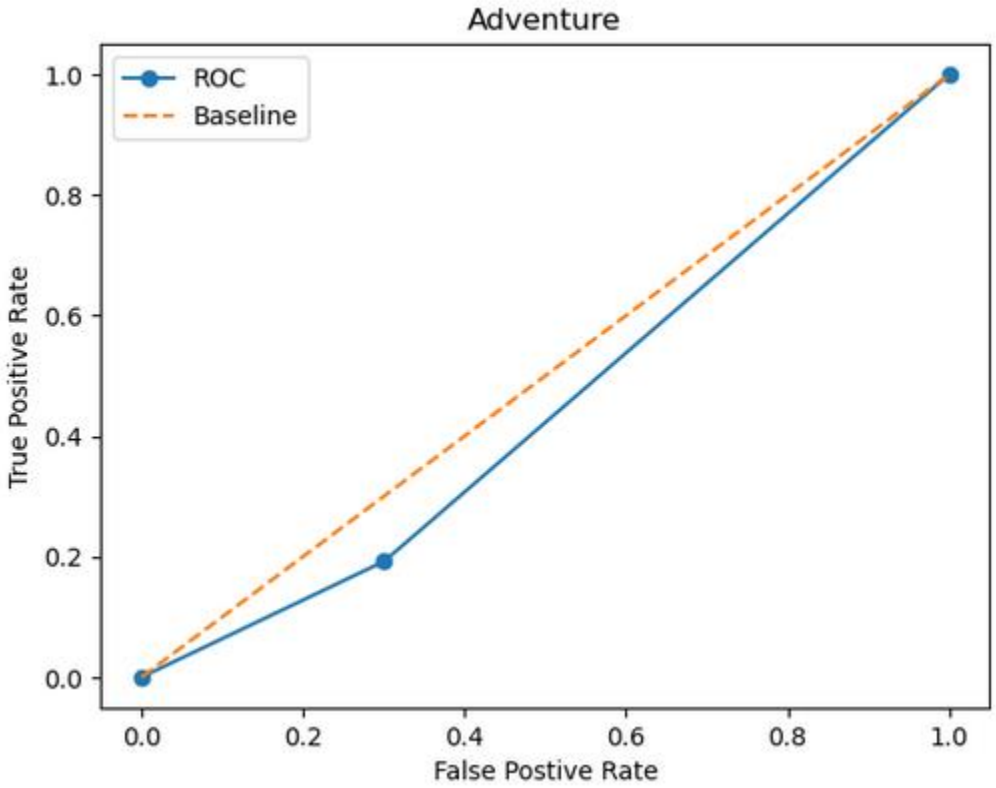
provided a mean or general sense of the model's performance when in regards to both its precision and recall for predictions of genres. A GridSearchCV was also implemented for hyperparameter tuning of the model to provide a generalizable model that is a high performer. 3-fold cross-validation was chosen due to limited resources and time versus a 5-fold and hyperparameters chosen to GridSearch was the model's compiler and model activatiion functions for each of the different layers. After training the model with the given train set, the final model metrics for the model with the highest f1_score had an optimizer of SGD or stochastic gradient descent and relu optimizer with an f1_score of 0.25. The best performing model was also evaluated based on its ROC AUC values for each of the different genres represented.

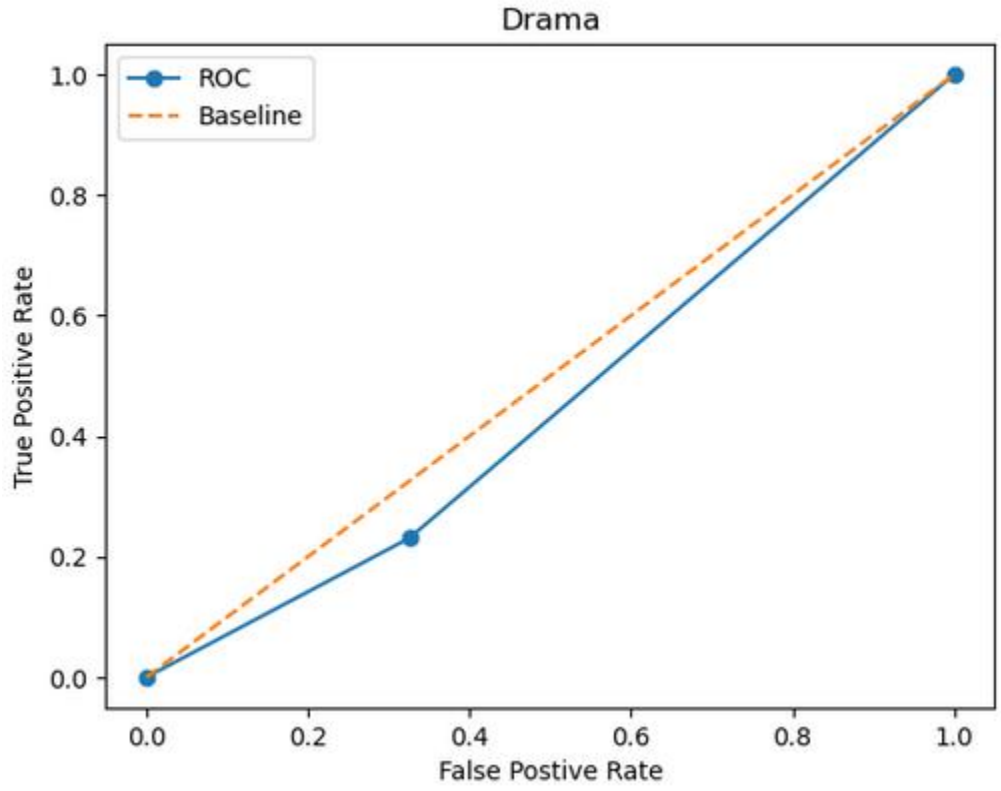
Layer (type)	Output Shape	Param #
conv2d_21 (Conv2D)	(None, 198, 198, 50)	1400
max_pooling2d_21 (MaxPooling2D)	(None, 99, 99, 50)	0
dropout_35 (Dropout)	(None, 99, 99, 50)	0
conv2d_22 (Conv2D)	(None, 97, 97, 20)	9020
max_pooling2d_22 (MaxPooling2D)	(None, 48, 48, 20)	0
dropout_36 (Dropout)	(None, 48, 48, 20)	0
conv2d_23 (Conv2D)	(None, 46, 46, 10)	1810
max_pooling2d_23 (MaxPooling2D)	(None, 23, 23, 10)	0
dropout_37 (Dropout)	(None, 23, 23, 10)	0
flatten_7 (Flatten)	(None, 5290)	0
dense_21 (Dense)	(None, 64)	338624
dropout_38 (Dropout)	(None, 64)	0
dense_22 (Dense)	(None, 32)	2080
dropout_39 (Dropout)	(None, 32)	0
dense_23 (Dense)	(None, 21)	693

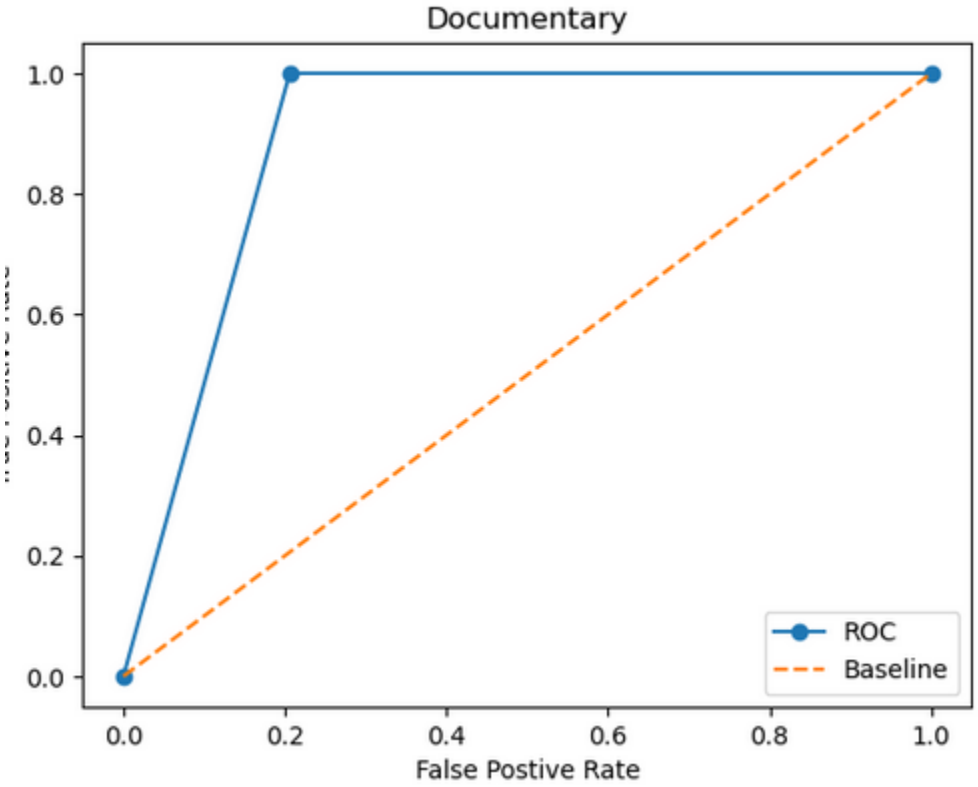
*Model.summary() done to reveal image classifier layer breakdown

Below is a sample of a few of the model's ROC AUC scores compared with a baseline of 50/50 random guessing for few of the represented genres.






Drama
ROC
Baseline
True Positive Rate
False Postive Rate
0.0
0.2
0.4
0.6
0.8
1.0


Documentary
ROC
Baseline
False Postive Rate
0.0
0.2
0.4
0.6
0.8
1.0

Results & Conclusion:

Based on the results, few genres according to the ROC AUC values seemed to underperform from the baseline but one genre seemed to have a high performance, Documentary. The model's overall f1_score of 0.25 indicates that there will need to be improvements needed before deploying this model within production.

Few reasons for the cause of the model's underperformance are potentially due to the imbalance of the initial dataset(higher number of one or few genres over others), the number of layers within the image classifier might not be the most optimal for generalizability of the model, 3-fold cross validation might not be the most optimal in terms of providing an overall generalizable model, also the number of hyperparameters checked within the GridSearchCV might need to be expanded to include parameters such as learning rate or checking more optimization parameters as well as activation functions. Therefore, in order to improve the model performance, factors such as increasing the movie genre sample size and applying a 5-fold cross validation instead of a 3-fold as well as exposing GridSearchCV to expanded set of hyperparameters might lead to a better model performance.

Future Work:

Future work for the model will involve increasing the samples of different genres of movies in the train set, adjusting the layers within the image classifier to be more optimal for RGB value input to predict a genre output, and increasing the number of hyperparameters to include different learning rates, optimization functions, and activation functions for the different layers.

Acknowledgements:

The author would personally like to thank Springboard Mentor Eric Callahan for his guidance and support throughout this project development.