

# Deep Learning Classification – Model Overview

Thomas Poozhikala

Springboard Data Science Career Track

<https://github.com/tpoozhikala>

---

Slide template provided by Sacramento State University  
Acknowledgement to [Eric Callahan](#) for guidance and support

# Introduction of Problem & Objective

---

- Problem
  - 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. This decline is potentially caused by lingering feelings from the COVID-19 as well as not capitalizing or marketing quick enough to their more avid movie goers.
- Objective: Potential strategy to implement in the 2024 financial year is to design a deep learning neural network image classification model to predict future Family Friendly Production movie genres based on the movie posters before the movies hit market.

# Current Model Status & Improvement

---

Keras Image Classification →

Model:

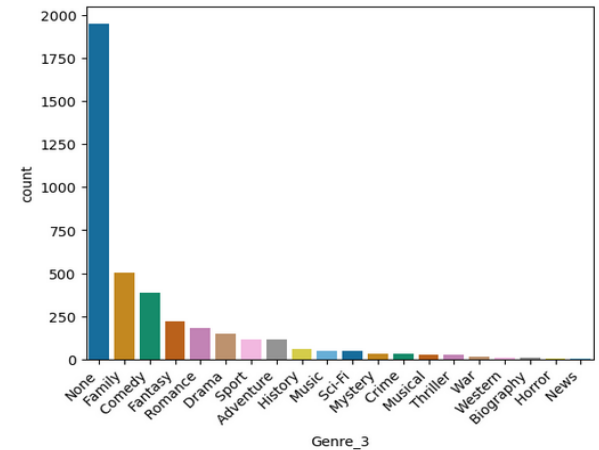
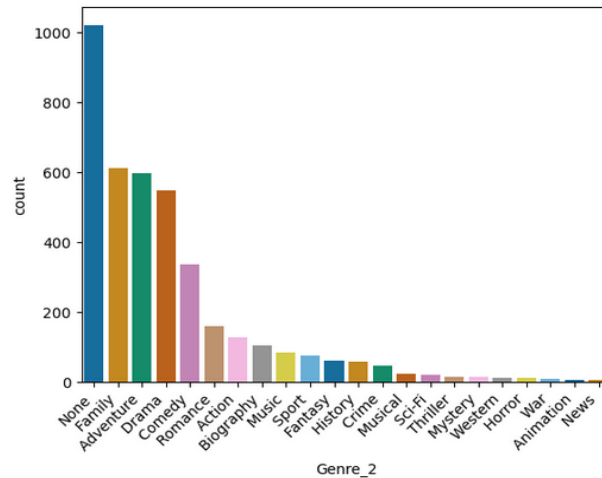
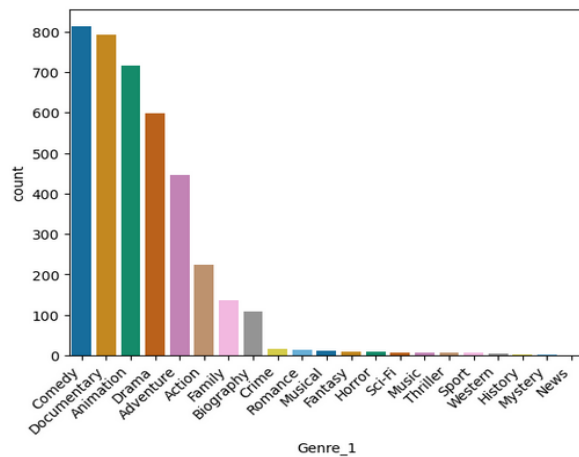
- F1 Score: 0.253
- ROC AUC Comedy: 0.498
- ROC AUC Adventure: 0.492
- ROC AUC Drama: 0.493
- ROC AUC Documentary: 0.503

Steps to Improve:

- Enlarge sample size
- Increase computational resources
  - 5-fold cross validation
  - Optimal model layering
  - Expose more hyperparameters in GridSearch

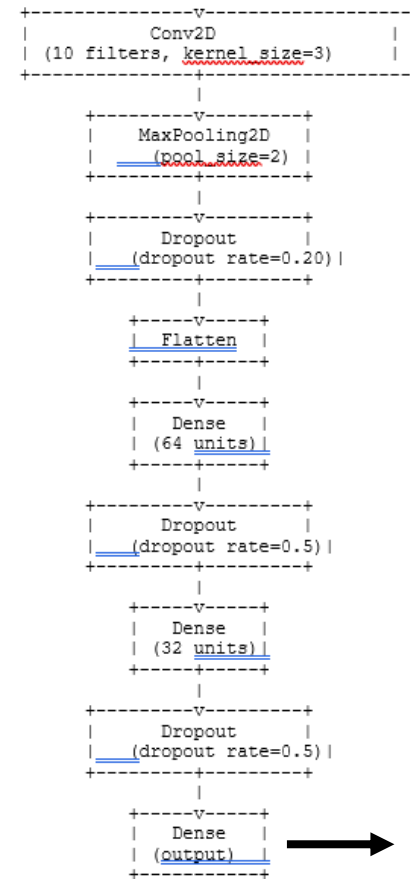
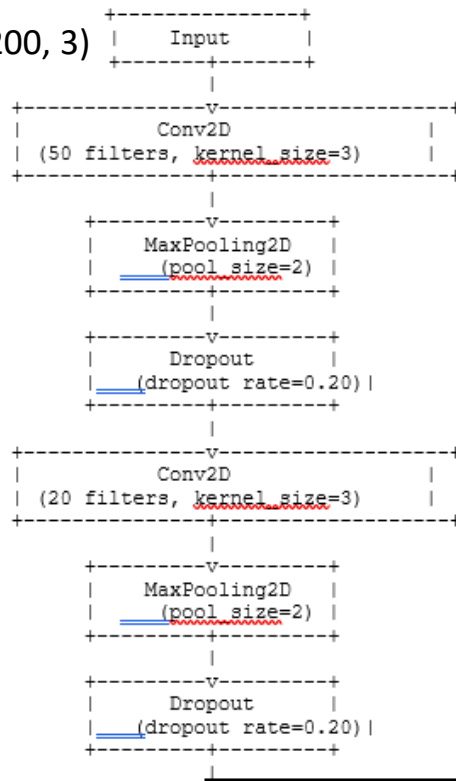
# Assessing Genre Class

- G and PG Movies from 1990 – 2023 retrieved through paid IMDb API
- 21 Unique Genre Classes Retrieved aside from 'None'



# Image Classifier Model Build

RGB Input Array: (200,200, 3)



→ Classification of Genre Class(es)

\*Model layer overview provided by chatGPT

# Image Classifier Model Results

- Image Classifier model trained with an 80/20 split and 3-fold Cross Validated for compiler and activation functions
- Best parameters SGD compiler and 'relu' activation function
- Model performance assessed based on f1\_score and best model performer ROC AUC values for different genres assessed.

```
param_grid = {  
    'activation': ['relu', 'tanh', 'linear'],  
    'optimizer': ['SGD', 'Adagrad', 'Adam'],  
    'learning_rate': [0.001],  
    'epochs': [10],  
    'batch_size': [10]  
}  
*Current model GridSearchCV Param Grid
```

Model Name	F1_Score	ROC_AUC_ Comedy	ROC_AUC_ Adventure	ROC_AUC_Drama	ROC_AUC_ Documentary
Keras Image Classification Model	0.25258895	0.4978308	0.49173069	0.49308443	0.503067484

\*Table Displaying Model Results f1\_score and respective ROC AUC for few genres

# Model Results Key Takeaways

---

- F1\_score of 0.25 means there are improvements to be made prior to deploying the model to production
- Reasons for model underperformance:
  - Imbalance of genres within initial dataset
  - Number of layers or filters within the model might not be most optimal for generalizability of the model
  - 3-fold cross validation not best method to generalize the model
  - More hyperparameters will need to be tested within GridSearchCV (learning rate, activation functions, model compilers, etc.)

# Future Improvements

---

- **Increase** sample size of movie genres
  - Include PG-13 movies and movies dating from 1980s to 2024
  - Augment current movie poster images
- **Increase** computational resources
  - Perform 5-fold cross validation instead of 3-fold
  - Optimize the convolutional and dense layers for appropriate filter numbers and kernel window
  - Expose current model's GridSearchCV to more hyperparameters (learning rate, activation functions, and model compilers)



# Summary

---

- Current Keras Image Classification model requires improvements before deploying to production.
- Suggested improvements include:
  - Enlarge movie genre sample size
    - Include PG-13 movies and movies from 1980s to present
    - Augment current movie poster images
  - Increase computational resources
    - 5-fold cross validation instead of 3-fold
    - Optimize different model layers
    - Expose current model to increased GridSearch parameters (learning rate, model compiler, activation functions, etc.)

# Thank You! Questions?

---

[https://github.com/tpoozhikala/IMDB\\_Classification](https://github.com/tpoozhikala/IMDB_Classification)