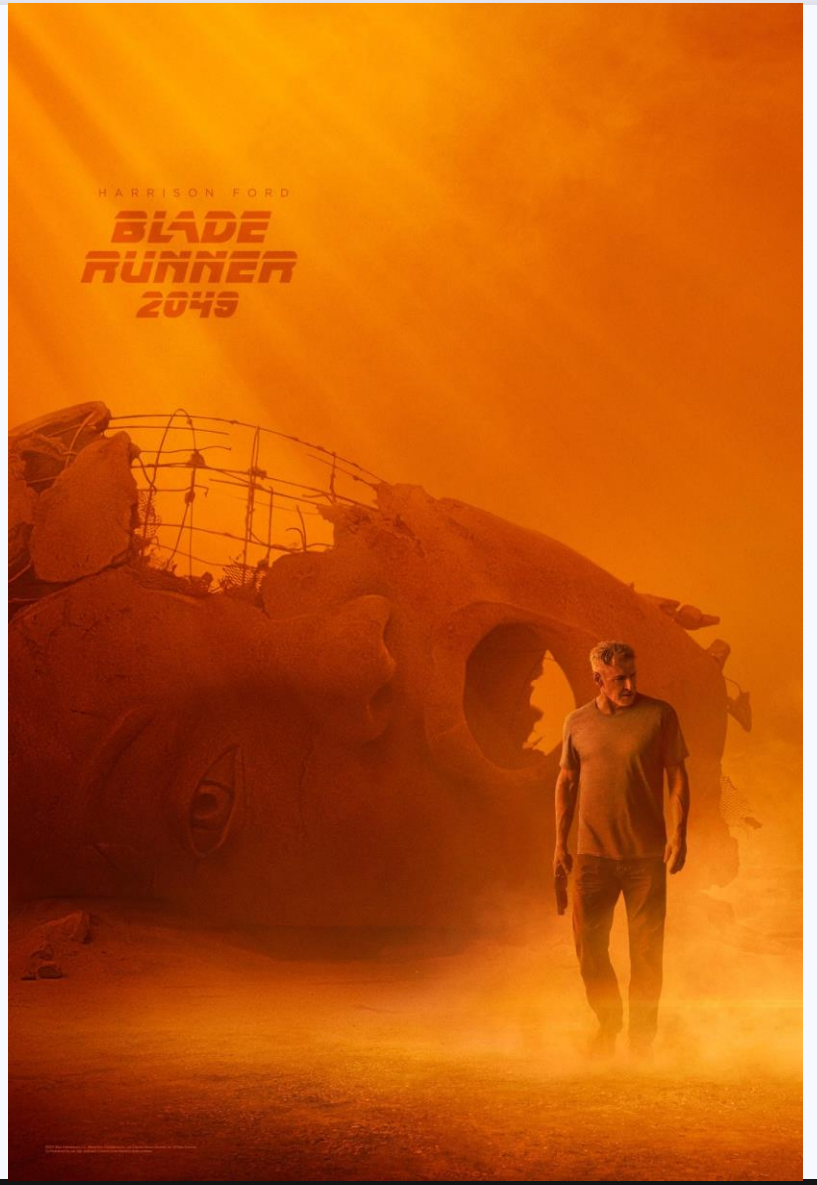
MOVIE DATABASE INSIGHTS

SOUMYA SARKAR



INTRODUCTION

As part of the challenge, we were tasked with deriving insights about the dataset and try to predict genre and year of release for any director.

Accomplishments so far:

1) Created ML model to predict top 6 genre using a simple sequential model with a best test accuracy of **81.129%**.

2) Created a separate ML model to predict year of release with best test MSE loss of **33981**.

ABOUT THE DATASET

- Initial Number of points = 5043
- 1274 rows have NaN values in them, while 3769 are clean
- Gross income of movie is the field with the most NaN values, indicating it as a tricky parameter to keep track of
- There were 1668 directors in total
- There were 22 genres considered
- There were 33 languages of movies in total from 45 countries.
- Several columns such as actor_1_facebook_likes were used for the prediction process as they can provide important information. For example, the facebook likes give an indication of what kind of actors the director has made movies with, and hence if that means the director is successful and popular, then there is a good chance he/she would continue making movies.

DATA PREPROCESSING

- We dropped two columns specifically – ‘plot_keywords’ and ‘movie_imdb_link’ as these would not have contributed to the overall result
- Rows with NaN values were dropped entirely
- ‘genre’ column values were first separated into different genres (22 genres), and then combined to create a multi-hot encoded vector

DATASET SPLIT

- A custom split function had to be made
- Train and test dataset were split in a fixed ratio for each director. For example, if James Cameron had 6 movies, then 4 would be put in the training set, and 2 in the test dataset
- The ratio was set at 80:20 arbitrarily

TRAINING PROCESS

- Two models were created for this – one dedicated to predicting genre, and one dedicated to predicting release year. Both consisted of Linear layers followed by batch normalization and ReLU.
- Genre model used KL Divergence loss (it gave good results), whereas Year model used MSE loss.
- Adam optimizer used.
- Predicting genre was particularly tricky as we could not use a regular softmax function, so we had to create a custom one which would not choose the top class to create a one-hot encoding. We instead use pytorch.topk() to choose top 6 genres and create a multi-hot encoded vector on that basis.
- Hyperparameters used are mentioned in the README file.
- Entire code is written in Python and Pytorch.
- Model is saved in a dedicated folder.
- An older model combined both predictions but was not performing really well (evaluation.py).

ABOUT THE CODE

- Log files ensure easy tracking and readability of code.
- I have tried to add comments wherever possible to ensure code is easy to understand
- I have tried to create modular code that can be easily scalable.
- ‘run_evaluation.py’ can be used to predict the genres and year for any director if given the proper input (that part has been left to the user) using the pre-trained models.

ACCURACY AND LOSS

- **Genre Accuracy**:
  1. Training – 81.222
  2. Testing - 81.129
- **Year Loss**:
  1. Training – 6655.680
  2. Testing – 33981.299
- We can see that while the genre model performs well with not much overfitting, year model had some overfitting issues. This can be fixed with some regularization.
- These are the best results we could obtain after testing with some hyperparameters.
- We use torchmetrics' inbuilt functionalities to calculate these values.
- Log files of both runs have been uploaded on GitHub in results folder.

CONCLUSION

- Achieved a best accuracy of 81.129% accuracy for genre and MSE loss of 33981 for year of release.
- Code has been written in a modular fashion with a view towards scalability.
- More experimentation can be conducted using other types of models such as decision trees.