Predicting Movie Ratings Using

Rotten Tomatoes Data

Avery Anjard

Abstract

In this report, the movie rating prediction project is introduced and the motivation behind it is explained. Then, it is explained how the model for the prediction was created using the Keras API and a dataset containing over 17,000 entries from the Rotten Tomatoes movie review website. Finally, conclusions about what trends the project highlights are drawn and how accurate a movie rating prediction model can be is discussed.

Introduction

This project aims to predict ratings of movies based on several factors including content rating, genres, director, actors, runtime, and original release date. This topic was chosen because what makes a movie “good”, or at least good to movie critics, is a topic that has been discussed for decades. There can be connections drawn between highly rated movies and who directed them, who acted in them, what their content rating is, etc. This application’s objective is to allow the user to see these connections more clearly by letting them play around with inputting different movie data and seeing what rating their hypothetical movie would receive.

Methodology

The model used for this application was created using Keras. Keras is a machine learning API that is built on top of the TensorFlow platform. The dataset used to train the model is the “Rotten Tomatoes movies and critic reviews dataset” created by Stefano Leone [1]. The dataset was first read into pandas DataFrame. Several of the columns were dropped from the DataFrame because they were either irrelevant, such as “rotten\_tomatoes\_link” and “movie\_info” or would make the prediction of ratings too easy, such as “tomatometer\_status”. It was then split into 2 smaller sets: one for training the actual model, and one for evaluating it at the end. The actual critic ratings were then separated from both sets to be used for validation during training and evaluation.

To begin training the model, the inputs would have to be preprocessed first [2]. This is done by first taking all the columns, turning them into Keras Input objects, and putting them in a dictionary. The numeric inputs were then put into their own dictionary, concatenated together, and put through a normalization layer. The result of this is then put into a list where all the other results of preprocessing will go.

For the string inputs, they are first converted into integer indices using the StringLookup method. These integers are then put through a CategoryEncoding layer to convert them into float32 data so that they can be more easily used in the model. Each column is then appended to the preprocessed data list. Once this is all done, all the preprocessed inputs are concatenated together to create the preprocessing model.

Before the training data can be used in the model, it must first be converted into a dictionary of tensors, since Keras models have trouble knowing if pandas DataFrames should be converted into one tensor or a dictionary of them. Finally, the model is built using the preprocessing model, several Dense layers with progressively smaller amounts of neurons, and a few BatchNormalization and Dropout layers in between them. All the Dense layers except for the outer layer use ReLU as their activation formulas, and a few of them use L1L2 regularization. When the model is compiled, it uses mean absolute error as the loss and adam as the optimizer. The model went through 50 epochs of training. It was then saved so that it could be used in the actual prediction application.

Conclusions

During the training of the model, the mean absolute loss was able to get down to around 12-13. However, during the evaluation, it was around 18-19. When predicting movies with known ratings, it was able to get very close at times (e.g., guessing 70 for a movie whose rating is 73), and very far other times (e.g., guessing 63 for movie rated 17). The model is far from perfect; however, it can reveal some interesting patterns. For instance, changing a movie’s content rating from G to R tends to decrease its predicted rating. Another interesting thing to note is the longer a movie is, the higher it tends to be rated. Finally, comedy and horror movies tend to be rated lower than drama movies.

Although the movie rating predicter is not completely accurate, it does highlight some interesting trends in how movies are rated by critics that many people might not notice. It would most likely not be possible to create a 100 percent accurate movie predicter, anyway. Although there are trends that can be found among movie ratings, there are also plenty of outliers. Movies made by unknown directors and actors can turn out to be unexpected classics. On the opposite side, films that are supposed to be box-office hits made by big-name studios can end up completely bombing in the ratings. So, although it might not be possible to predict how a movie will be rated completely accurately, it is possible to analyze the trends and come up with some interesting conclusions.

References

[1] S. Leone, 2021, “Rotten Tomatoes movies and critic reviews dataset,” Kaggle. [Online]. Available: <https://www.kaggle.com/datasets/stefanoleone992/rotten-tomatoes-movies-and-critic-reviews-dataset/code>

[2] “Load CSV data,” 24-Mar-2023. [Online]. TensorFlow. Available: <https://www.tensorflow.org/tutorials/load_data/csv>