Movie Rating Prediction Using Convolutional Neural Networks

Christopher Pyles

March 20, 2020

1 Introduction

2 Data

The data used in this project is provided by The Movie Database's API (TMDb). The data queried covers movies released in 2018 in English, spanning many genres:

- Action ($n = 245$)
- Adventure ($n = 134$)
- Animation ($n = 117$)
- Comedy ($n = 569$)
- Crime ($n = 155$)
- Documentary ($n = 464$)
- Drama ($n = 764$)
- Family ($n = 148$)
- Fantasy ($n = 144$)
- History ($n = 67$)
- Horror ($n = 446$)

Column	Description
`id`	**primary key**, a unique ID number for each movie
`adult`	whether or not the movie is R+ rated
`genre_ids`	list of genre ID numbers corresponding to `genres` table
`original_language`	the language that the movie was originally released in
`original_title`	the title of the movie
`overview`	movie synopsis
`release_date`	the date of first release
`vote_average`	average of rating votes on a 10-point scale
`vote_count`	the number of votes
`poster_path`	URL path to movie poster

Table 1: Column descriptions for TMDb API data (the `movies` table).

- Music ($n = 126$)
- Mystery ($n = 144$)
- Romance ($n = 222$)
- Science Fiction ($n = 188$)
- TV Movie ($n = 197$)
- Thriller ($n = 511$)
- War ($n = 29$)
- Western ($n = 31$)

The data, after being queried, were joined into a single table and written to a CSV file. The columns of interest are described in Table 1. After dropping rows with missing values in the columns of interest, the data contained $n = 2700$ rows.

2.1 Data Cleaning

Cleaning the data for this project, after dropping missing values, involved rounding the `vote_average` column, cleaning the synopsis strings, joining the `genres` and `movies` tables, and converting the movie posters into $140 \times 92 \times 3$ arrays of RGB values.

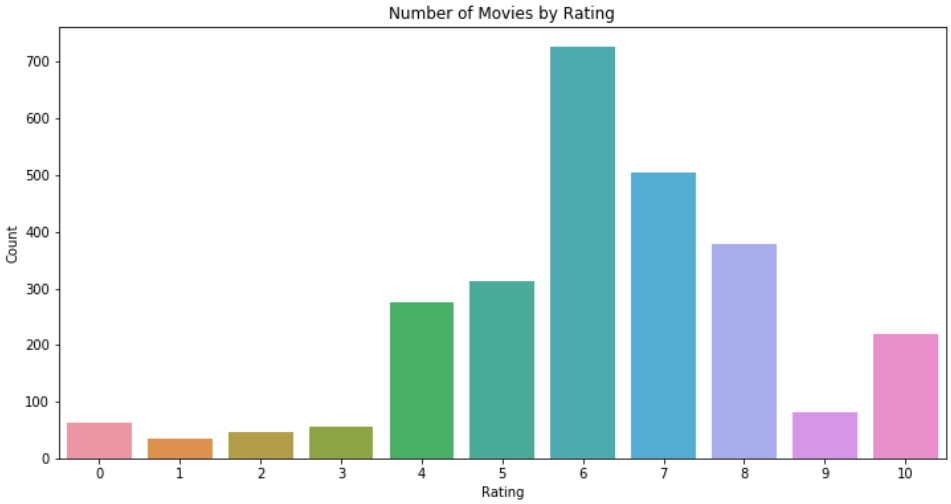

Figure 1: Number of movies for each value of ratings, rounded from `vote_average`.

The `vote_average` column ($\mu = 6.628$, $\sigma = 1.939$) describes the variable of interest, representing the average rating across votes for a single movie. In this analysis, this column will be transformed from a continuous variable into an ordinal variable with possible values 1 to 10 by rounding the vote average to a whole number. After rounding, the distribution of ratings is provided in Figure 1.

To clean the movie synopses, all letters were transformed to lowercas, and any characters not matching the regex `[A-Za-z0-9 ]` were replaced with spaces.

The `genre_ids` column is a list of genre IDs encoded as a string, so to start any empty lists, `"[]"`, were replaced with `NaN`. Then, the string were split into Python lists of IDs, and were joined with the `genres` table, so that `movies` had a column `genres` where each value is a list of genres for that movie. Figure 2 shows the breakdown of movies by genre.

Lastly, each movie poster was downloaded from TMDb and stored as a 140 $\times$ 92 $\times$ 3 array in NumPy, stored as a .mat file. The structure of this file is analagous to a dictionary, where each key is a movie ID (as a string) and each value is the 3-D array of RGB values describing the poster.

2.2 Exploratory Data Analysis

Before modeling, a cursory analysis of the data yielded the interesting relationships:

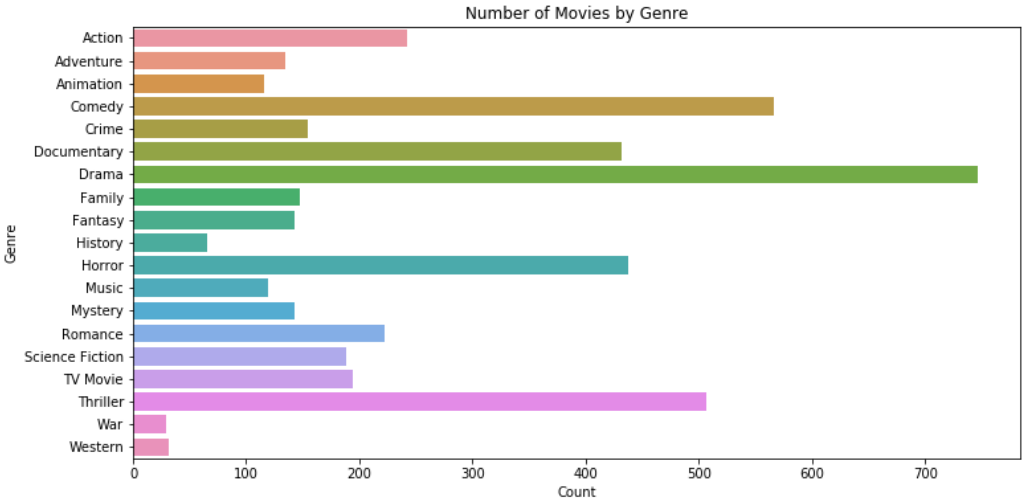

Figure 2: Number of movies in each genre.

- Figure 3 shows that vote count tends to increase logarithmically with rating until 8, after which it drops significantly. This demonstrates that having more votes tends to "bring down the curve."
- There are significantly more non-adult movies than adult movies, as demonstrated in Figure 4. It is interesting to note that adult movies have a double-peak distribution with far less spread than do the non-adult movies.
- Figure 5 shows the distribution of ratings for each genre.
- No discernible relationship is seen between the length of the overview and the movie's rating.
- The distributions of ratings by release date day of week appear to be different in skew (Figure 6), indicating that this may be an important feature. No discernible relationship is seen between rating and day of month or month of release.

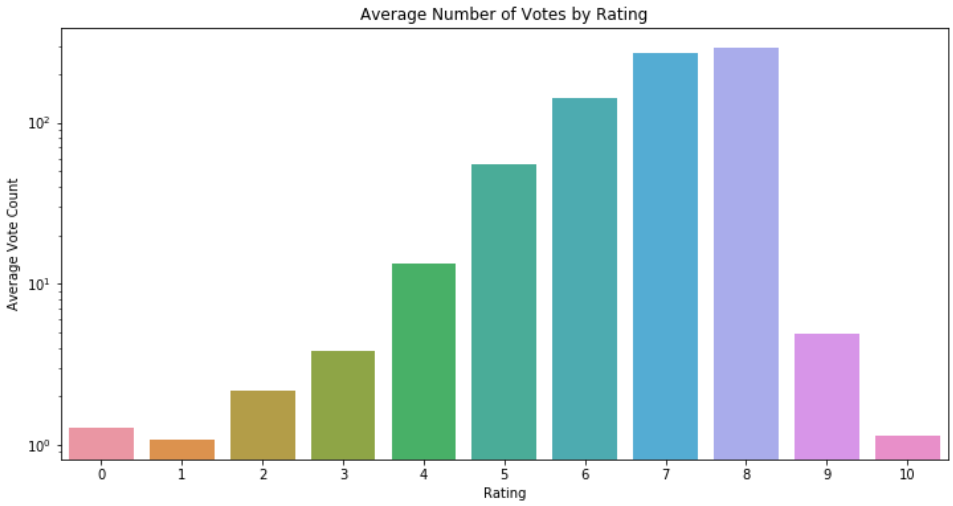

Figure 3: Average vote count by rating.

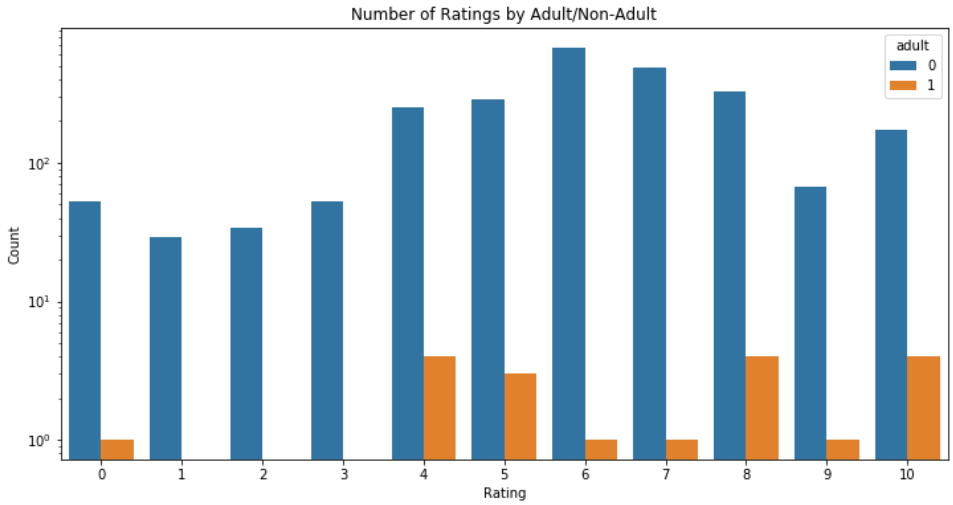

Figure 4: Number of ratings by adult/non-adult.

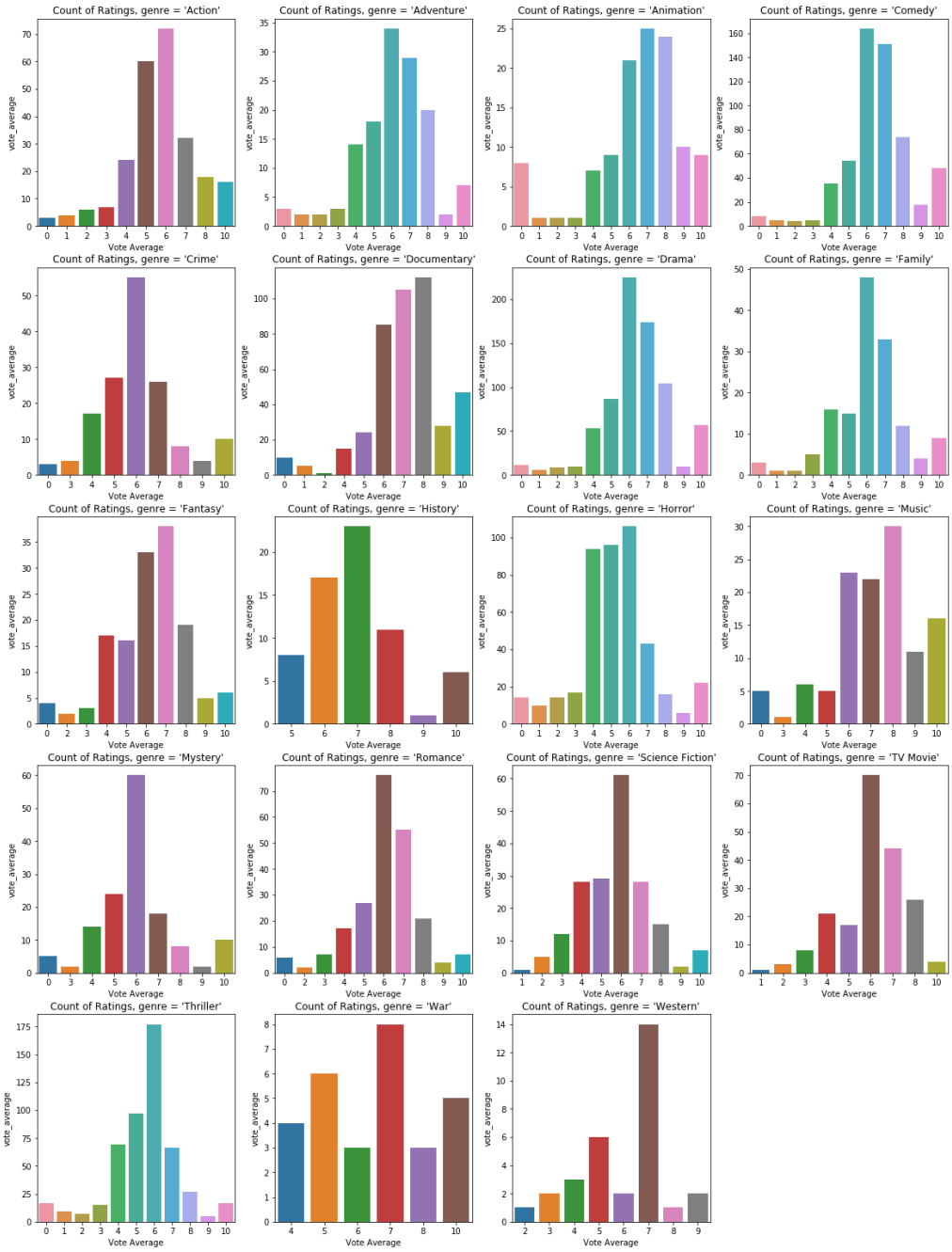

Figure 5: Distribution of ratings by genre.

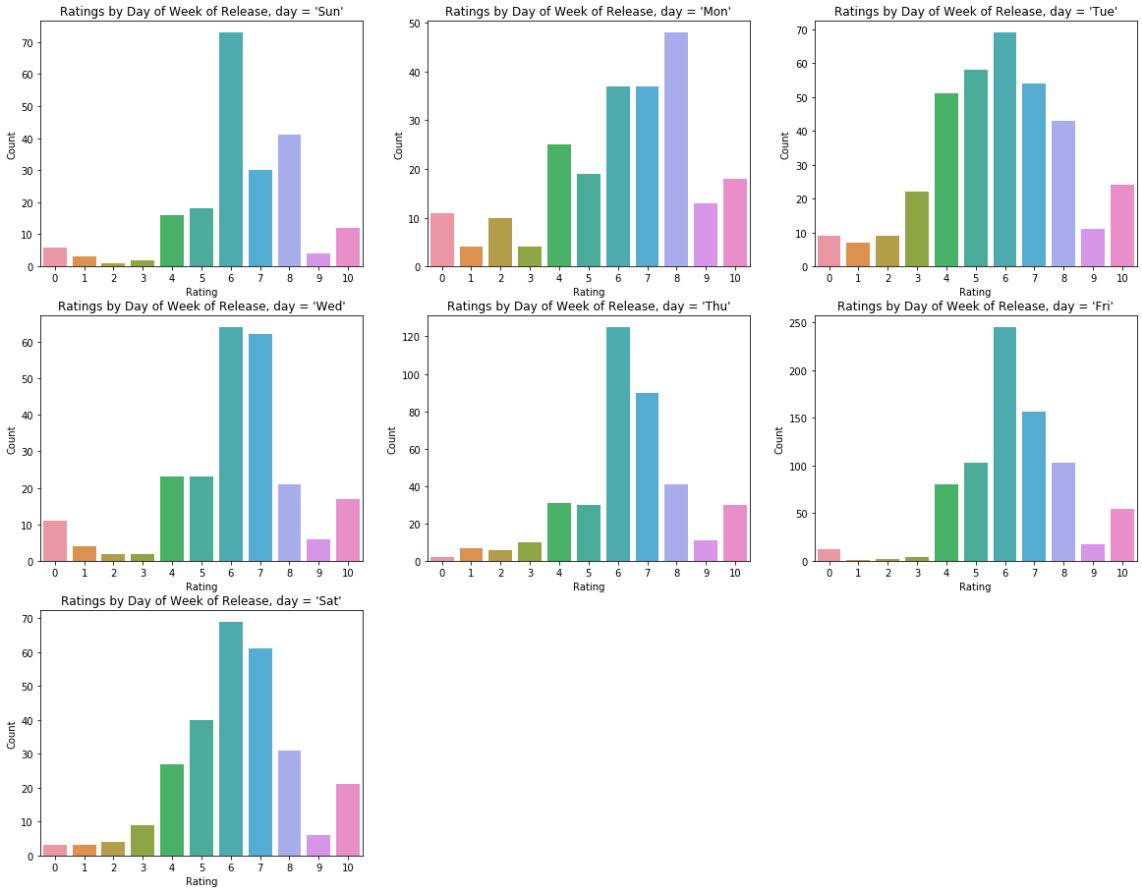

Figure 6: Distribution of ratings by day of week of release.

3 Analysis

3.1 Words in Synopses

As the first part of this analysis, the statistical significance of the presence of different words in synopses is studied using hypothesis testing. The null hypothesis for this test is that the presence of a word does not affect the average rating, and the alternative is that the presence of the word results in an increase in rating. The test statistic used here is the aboslute difference between the mean rating of movies where the word is present in the synopsis and the movies where it is not. Under the null hypothesis, the truth value of a word being present in a synopsis is shuffled for each observation, so that same number of "present"s is obtained. Then, the test statistic is computed and the p-value is calculated as the proportion of the test statistic values that are greater than or equal to the observed value for that word.

3.2 Classification without Posters

The next step in the analysis is to attempt classification using the features in `movies` alone (i.e. without the posters).

4 Results

4.1 Words in Synopses