Dylan Davis

For my project I wanted to see if I could predict the average rating of a TV show based off information about that TV show. This could be something interesting to TV show writers as they can take into account things that may improve the overall rating of their show. I was specifically interested in animated TV shows, so my first step was to find more information about TV show ratings. I ended up finding the IMDB data set, which consisted of multiple tables providing all sorts of information about TV shows and movies. I didn't want to use just one dataset, so I came up with the idea of adding TV show script word count alongside my current data. I found a website that contained TV show scripts for each episode of every show I was interested in.

The IMDB TV show ratings information was in the format of tsv files. After downloading the files into a folder, I was able to import the files into R using the read.csv function and adding the "sep" parameter so the function knew the file was separated by tabs. To import the TV show episode scripts from the website I needed to use the "rvest" package, which makes it easy to scrape data from html web pages. With the help of a post from "r-bloggers.com" I was able to import the TV show scripts for each episode within each TV show.

The next step required cleaning the massive amount of data that I had obtained. I first needed to combine the three tables of IMDB data in order to have one complete table with all the necessary information such as: number of votes, runtime, start year, average rating, etc. I was able to do this using the merge function, which allowed me to merge on a column that contained a unique Id for each TV show episode that all three tables contained. The next thing I did was subset this data frame because I only needed the data for the eight TV shows that I was interested in looking at. I was able to search the name of TV episode for a particular show, which contained a feature called "parentTconst" and this was a parent code for that particular show. By subsetting

this data frame by all eight parent codes I was able to obtain a subset of the data frame with just the information that I wanted. Lastly, I cleaned the column titled “originalTitle” and this was because I would need this column later in order to merge the IMDB data with the word count data. My next step was to clean the script data so that for each TV show I had a data frame containing the word count for each episode. Again, with the help of a post from “r-bloggers.com” I was able to construct this data frame.

Next I bound the word count data from each TV show together by row using “rbind.fill”, but there was a problem because I had over 19000 columns. In order to reduce this vast number of columns set a threshold for the average required word count per column and I eliminated any columns that did not meet this threshold. I then used a function to scrape all the episode names for each TV show from the website that contained the TV show scripts and then I combined the episode names into one big list. Finally, I was able to add a column containing the episode names to the word count data frame. From here I was able to merge the IMDB data frame and the word count data frame based on the name of the episode. I then removed unnecessary values such as episode title, title type, etc. These features would have no predictive value in the model, so it made sense to eliminate them. I also made sure to convert factor values to numeric so that they were in the correct format. After all cleaning and merging was complete, I was left with a data frame containing 37 columns.

Now that I had my feature matrix, I was able to partition my data into a training and a testing set. The variable I was trying to predict called “averageRating” was continuous, so I used linear regression. After running my model and testing some values it seemed that it was underpredicting the rating of the TV show and in some cases, this was quite significant. It was interesting to see that certain words did in fact have a statistically significant positive effect on

the rating of the TV show such as the words come, get, just, okay, think, and yeah. Another interesting outcome was that the season number had statistically significant negative effect on the rating of the TV show, so did TV show episode run time. However, after looking at the Residuals vs. Fitted plot it is clear that there are some outliers and the relationship isn't entirely linear. The Scale-Location plot showed a non-uniform variance in the lower range. The mean absolute error was 0.09855234, which shows that there was some inaccuracy in the predictive values of the model, which was verified when looking at the actual values in the data set.

This model was by no means the best model and it definitely needs some tweaking. If I had more time to play around with the features of this model, I think it could be fine-tuned to bring more insightful predictions. I was not able to accurately predict TV show rating, but I did enjoy this project overall as it gave me great real-life experience.