Manan Maniyar
30129713

CMPT 353 Project Report

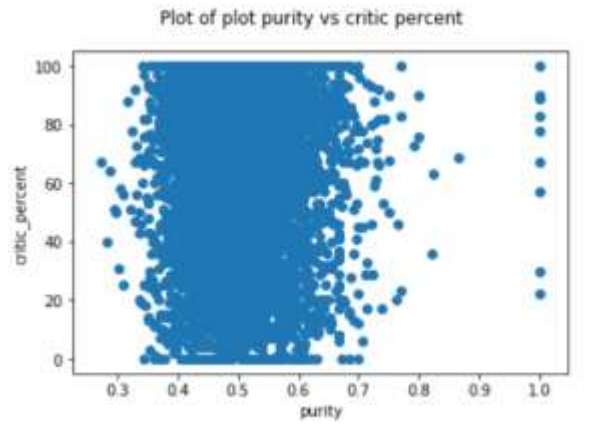
Problem Addressed:

My goal was to prove that plot summary given by IMDB had no relation with Rotten Tomatoes critic ratings.

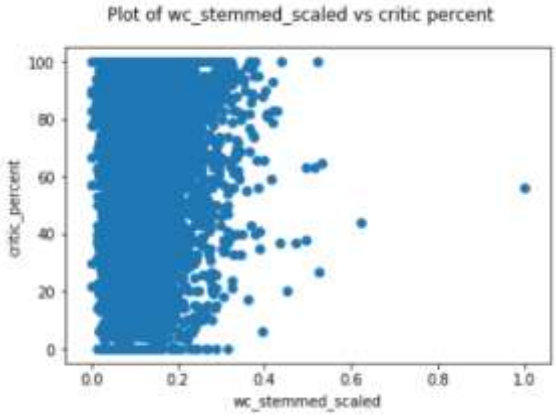
Getting/Cleaning Data:

I had two data sources, one by IMDB and one by Rotten Tomatoes. The IMDB dataset had good textual information which encouraged to explore and experiment NLP, and the Rotten Tomatoes dataset had numerical information like critic rating, audience ratings etc. So, I merged the two datasets and removed the unnecessary columns which I didn't needed. The obvious intuition was to apply regression methods to plot summary and critic percent column and state that these aren't correlated, but the problem was to convert the textual data (plot summary) into some form of numerical data. I started with tokenizing the summary (separate into words) and then removing the punctuations (',', '.', '!'. '%') and stop words ('is', 'a', 'the', 'and'). After that I had only nouns and verbs in the plot summary column and I also added some separate columns like purity (Weighted Words/Total word count). Also, I removed the columns. I also removed all the rows where the values were Nan to remove all the redundant data. After my data cleaning, I lost around 30% of my data points but at least it was not redundant. Also I had to do some rearrangements in the dataset's columns in order to get according to my needs.

Manan Maniyar
30129713



Below is the graph of plot_summary purity (proper nouns and verbs) vs **critic_percent** (Target Variable)



Data Analysis:

After cleaning the data, I had to apply TFIDF vectorize to my plot summary data to reflect how important a word is to a document in a collection or corpus. Tfidf is just a simple ranking algorithm which works on term frequency. After applying to the TFIDF I was left with a N*N matrix which represented the plot summary in numerical form. N is the no. of distinct words in the plot summary column (after cleaning, as having stop words like 'a', 'is' make no sense). I tested my training

datasets on Lasso Regression (as Linear Regression wasn't working well on multiple columns.) by throwing 4 different values of Alpha.

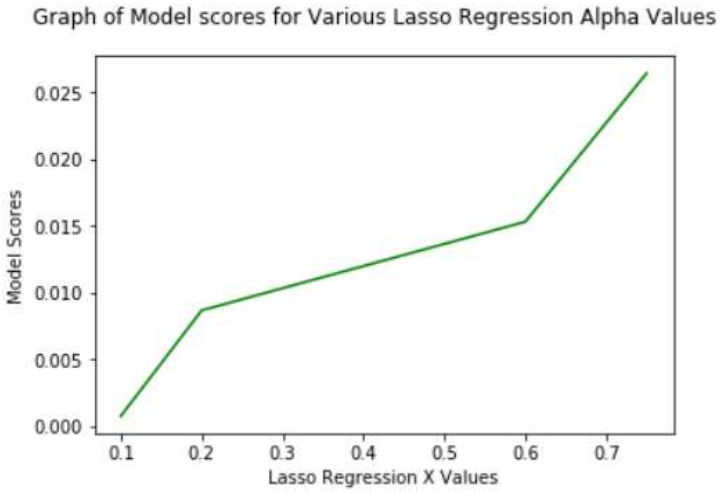
Results:

LassoRegression(Alpha=0.1) = 0.0813

LassoRegression(Alpha=0.2) = 0.0093

LassoRegression(Alpha=0.6) = 0.0147

LassoRegression(Alpha=0.75) = 0.0267



The graph shows how the Lasso Regression performed over different range of alpha values.

Limitations:

Initially I wanted to make a recommendation engine which predicts the top 3 related genres to an input of a plot_summary. But in the dataset for a plot_summary there was a list of genres and for throwing the training sets into classification Algorithms I had to have just one label(genre) for each plot_summary but wasn't able to find some sensible method to concatenate 3 genres per plot into one.