July 15, 2021

Francisco Barba

Final Report:

TV-Movie Recommendation System

I am pleased to provide you with this report of my project in conjunction to my analysis that utilizes IMDb data from 10,000 movie titles and approximately 6,500 tv-series. Hence, the recommendations and/or findings herein are purely objective and derived from the data collected.

# **1. Executive Summary**

My analysis indicates that in the movie and television sector, many of the content provided are described in roughly 100 and 17 keywords, respectively. The items shown below are discussed in greater detail within the body of the report.

* The entertainment data collected were cleaned, organized and merged together in order to accurately conduct the analysis.
* We had to use the top 10,000 movies with the highest amounts of votes for our project to conserve memory and efficiency.
* Applied Matrix Similarity model to the pre-processed data to search for movies, tv-series, or search for the content using the keyword search, based on the input of the user.
* Applied the Cosine Similarity and Jaccard Similarity model to the preprocessed data that took in both movie and tv-series title as an input.

# **2. Project Understanding**

I understand that the purpose of this project is to optimize an entertainment recommendation system in order to retain customers. This is especially true in the dawn of streaming, in which massive influx of content is provided and costumers are giving the opportunity to watch them in giant bulks – or binge. Therefore, with wide and broad selections of content, comes an even greater obligation to personalize these selections to individuals.

Further, the cost benefits of improving the system for the customers to use is substantial.

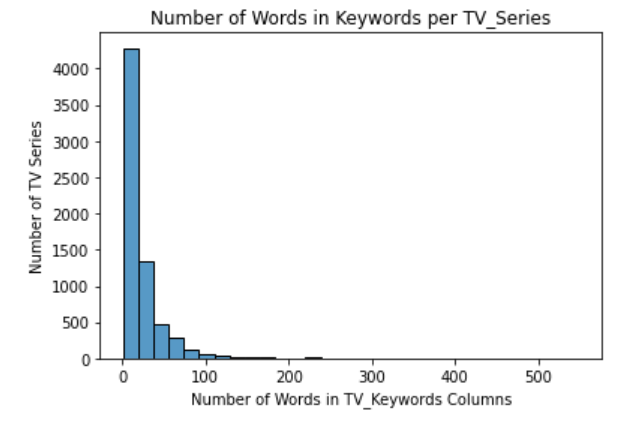
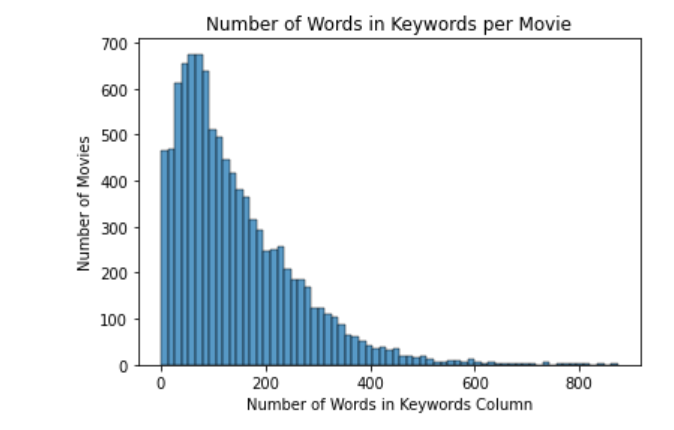
*“A “5% increase in customer retention produces more than a 25% increase in profit. Why? Return customers tend to buy more from the company overtime. As they do, your operating costs to serve them decline. What’s more, return customers refer others to your company.” (1)*

The influence that loyal customers have in attracting new customers is intangible. Once there are loyal customers, their influence will ripple, and in turn, attract new ones by word of mouth. This is why it is important to always improve and innovate existing products with the target audience being the loyal costumer. It proves that improving can have profound external affects.

# **3. Findings, Trends and Insight**

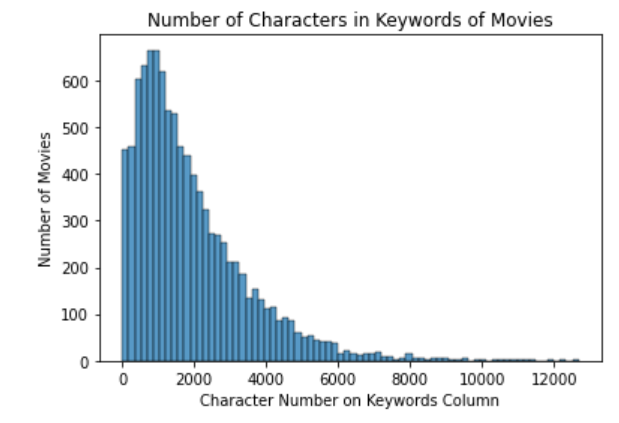
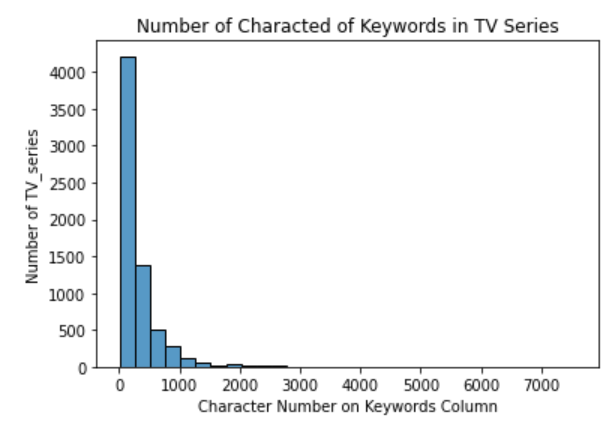
During the exploratory phase of the analysis, I’ve managed to retrieve keywords each of the movie and tv titles from the IMDb API. The amount of keywords in majority of the movies in the sample is 100, and 17 for tv-series, as shown below in **Figure 1**.

**Figure 1: Keyword Amount Comparison between Movies and TV-Series**



As shown above, both the graphs depict that the distribution is right-tailed. This tells us that both movies and tv shows have relatively small amounts of keywords to describe them, presumably except for the most successful titles. Further, this pattern is also replicated by the number of characters within each title’s keyword section – almost intuitively so-, as depicted in the following page as **Figure 2.**

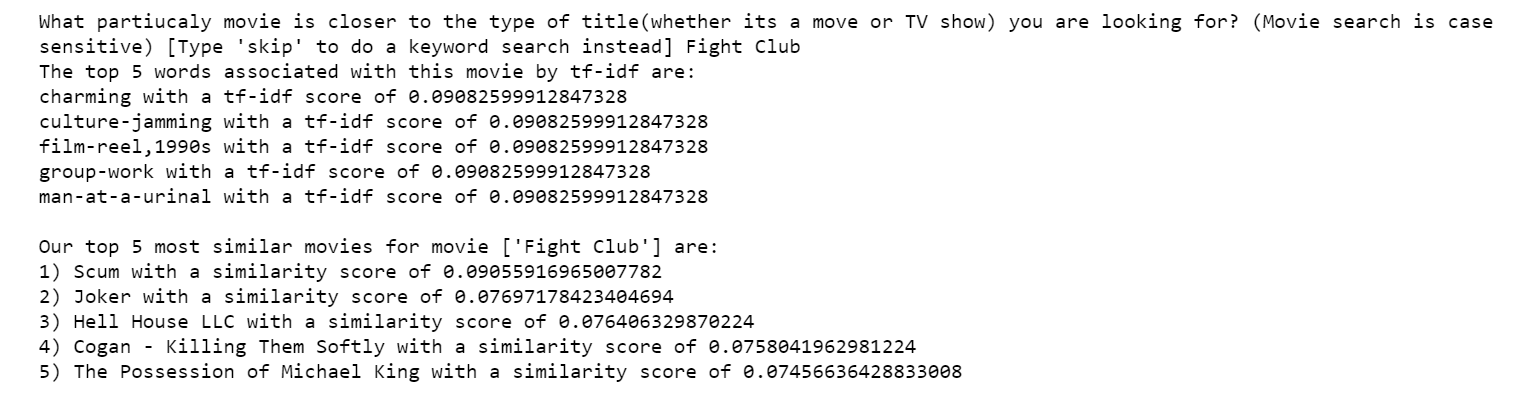
**Figure 2: Character Count Comparison of Keywords between Movies and TV-Series**



# 4. Model Selection and Metrics

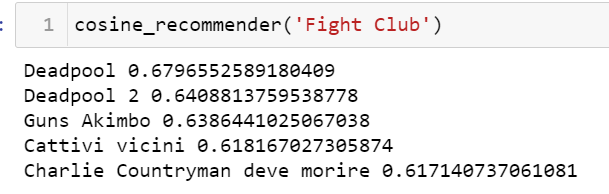
To prepare the data for the machine learning model, I needed to not only reformat the keywords that we’ve retrieved but also change its data type. To do so, I’ve had to convert the data to a list, and tokenized each of the keywords, so they can be its own individual elements that will be used through prime to model for the recommendation system.

To make the model more robust I’ve created 3 functions that will take the user’s inputs, decide whether it’s a movie, tv show, or keywords. Further, I reformatted each keyword to a dictionary with their own ID values. This dictionary will then be transformed into a bag-of-words with values of “1” for every time the keyword has appeared. This prepared it enough to the use term frequency, inverse document frequency method where, based on how many times the word appears in it local document and global document, will give each word a certain weight that will influence the outcome of the recommendation system.

* **Matrix Similarity:** Essentially, it creates an index and tells how similar the query document is to each document in the index. This will provide a vector of numbers as large as the initial set of documents. These similarities are found based on their cosine distances of the vectors of the indexes. Utilizing this model, I was able to create a function that takes in the user’s movie/tv title and/or keywords and return the top 5 most similar corresponding content with their tf-idf and similarity values - shown below in **Figure 3** with the movie “Fight Club”.

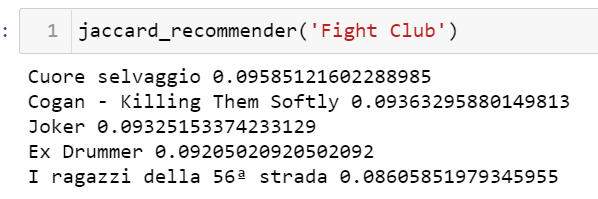
**Figure 3: Matrix Similarity Output for “Fight Club”**

* **Cosine Similarity:** The cosine similarity bases its results on the normalized dot product of the input with all the other documents in the sample. Utilizing CountVectorizer to convert the keywords of the to a matrix of token counts. The similarity portion is found based on the cosine angles of the matrix created of the document. Below in **Figure 4** is an example of the output for “Fight Club”.



**Figure 4: Cosine Similarity Output for “Fight Club”**

* **Jaccard Similarity:** Out of the other models chosen, Jaccard is the simplest and most straightforward. It utilizes ratio of the intersection of the set input keyword and all the keywords available with the length of the set of input keyword and all the keywords. The main idea behind is in its rigid mathematical approach in which the higher amount of keywords are in both the input document and the others, the higher the ration – hence higher similarity. In **Figure 5** below I depict the results of this similarity model using the movie “Fight Club”.



**Figure 5: Jaccard Similarity Output for “Fight Club”**

The metric used to evaluate these models was intuition and research on the output content based on the inputs. With that, I opted for Matrix Similarity to be the most accurate model of the three. Although, the Cosine Similarity model and the Matrix Similarity model were similar – in the way they utilize cosine distances – the reason why they are different is because the Matrix Similarity model takes advantage of the tf-idf values and the Cosine Similarity model uses Count Vectorizer. Hence, the difference in recommendation becomes very apparent when comparing both the models. Further, since Matrix Similarity in general show the most accuracy, I revamped the function to take in user inputs in real time and filter between movie/tv titles and keywords.

# 5. Limitations and Constraints

The analysis does have it’s drawbacks when it comes to the size of the content. There are more than 10,000 movies and 6,500 tv titles but since that takes up too much memory on my laptop, I opted in shortening the movies. Additionally, the content provided is not up to date to the time of this writing, so it will not have the most recent releases in the dataframe.

# 6. Conclusion

The recommendation system, as any other, is not a final product. It is meant to be continuously worked on for improvements and optimization. This is what drives competition and keeps customers retention high. It should be re-emphasized that, at the time of this writing, the data collected for these models do not include the most recent content.

Furthermore, the model of choice for this project is Matrix Similarity because of its intuitive correlation between the inputs and its recommendation. The main goal for creating this models was to find a way to improve content recommendation systems in order to be more up to date as well as to improve user experience. With streaming services being as massive as they are now, and with the high influx of content constantly coming out, the customers need better recommendation systems to provide them a simplified amount of content without overwhelming them. Further, some of the next steps for this project will be creating a movie bot and deploying the model in order for users to actually be able to utilize it.