Luis Ruiz

Recommendation System

Final Report:

Recommendation System

**Problem Statement**

**How can machine learning be used to create a movie recommendation system that will return a film based on previous viewed films? The film will be selected using a scale that measures compatibility using a ‘match’ scale. Using this scale will decrease the average time it takes to select a film.**

There are hundreds of movies released on an annual basis, some great and some not so great. Being able to narrow down on what movies to watch based on personal preference is a strategy many companies are using in today's age. For example, on Amazon they’ll recommend similar items to buy based on what you recently bought or what was the last item you viewed. Imagine a personalized movie recommendation system that returns what movie to watch based on previous movies. My goal is to create a recommendation system that makes selecting a film much easier by returning a match factor that ranges from 0-100. 0 being you will not like it and 100 meaning you will love it. Our system aims to return at most 10 movies, so the user does not suffer from choice overload. Now I will only focus on films and maybe once I create an accurate enough system I will try to venture into the anime or tv shows. Some constraints we might face are dealing with movies that fall under several genres. Trying to combine collaborative filtering and popularity-based selection. To verify if our model is accurate, I will feed it movies I like and don’t like based on the recommendation I will identify if it is accurate. I will also ask other people to verify my findings. The dataset I will use contains 3 different csv lists that contain the movies, ratings, and keywords.

**Data Wrangling & Exploratory Data Analysis**

My initial dataset contained about one million films with 15 columns, but dataset was eliminated due to the lack of computing power I had available. I ended up selecting two csv files, one named movies\_collab.csv which contained the movies titles along with other relevant features like genres, keywords, overview. This csv filled contained 4809 rows and 23 columns. The other csv file, collab\_rating.csv, contained a movieId which was used as a unique identifier for each movie that was later used to merge both of our datasets. Other important columns it contained were ratings, userId, number of ratings just to name a few. The movie dataset contained a lot of unnecessary features so using both lasso regression and my domain knowledge to eliminate some of the features. This left us with 9 columns. There were four columns that were converted from a string literal to a string. The columns were genres, crew, keywords, and cast. The crew column contained everyone involved in the production of the movie, so the backend of filming production like directors, writers etc. The director was used for our model since my I only cared for the director. The cast column contained the front-end of the film crew so actors and extras. I extracted the first 10 actors to cut down the complexity. The genre column was a dictionary of string literals which made it hard to work with but after converting the column to a string I was able to extract the columnsChart, histogram

Description automatically generated.

Visualizing the genre column using a bar-plot, it was clear that there wasn’t an even number of movies per genre. This could be because movies can be classified with multiple genres as well as the popularity of certain genres vs others. I did not perform oversampling or under sampling because our dataset is a good representation of films.

**Feature Engineering & Modeling**