Recommender Systems

A recommender system, on its most basic level, assigns weights to different criteria allowing for a result to be displayed. This result could be predicted or could simply be a list of results matching a genre, for example. Using a recommender system, a company can predict results for a user, increasing user engagement. Netflix uses a recommender system to predict what movies a user will enjoy, using data Netflix has gathered on the user. I have learned four different recommender systems so far: basic, knowledge-based, plot description-based and metadata-based recommender systems.

The basic recommender system is more of a sorted list, more than a system that can recommend results based on different users. This system uses the pandas library for Python to create a dataframe from the read in CSV file containing movie data from IMDB. Taking the vote\_count and vote\_average, a weighted score is calculated using IMDB’s own formula. This program then creates the IMDB Top 250 Movies list.

When building the knowledge-based system, more criteria must be kept in mind. There is also no more hardcoding of inputs for the system. These will instead be handled by the user. For this system, the starting and ending years of movie production, the low and high extremes for the movie’s duration and genre and taken into consideration. These values are then used with IMDB’s formula to recommend movies based on the criteria the user entered. The system then outputs the top movies in descending vote average order.

The first system that allows some magic to happen in the background is the plot description-based recommender. As the name suggests, this recommender system compares the plot of each movie to recommend similar movies. As with the rest, a CSV file is read in to use as data. Instead of using the data in its raw form like the other recommender systems, this system uses TfidfVectorizer. This allows the words in the CSV file to be converted into a term frequency-inverse document frequency matrix. This is used to text mine. In the case of this example, each word that appears more than once and is not a stop word, gets a higher weight. The theory being if The Lion King has the word “lion” a lot in its plot and another movie as has “lion” a lot in its plot, the movies are similar.

This method can also be used with a neural network. I have a neural network using this technique and training data to determine whether a post is obscene, toxic, hate speech and some other criterion. Combining this recommender system with a neural network using Keras or Tensorflow can make for a very powerful system for recommendations based on a single value to measure.

This last recommender system is essentially a more advanced plot description-based recommender system. This model uses the genre, director, the movie's three major stars and sub genres or keywords to give a recommendation. Using this information, the system is able to give a much better prediction than the last recommender system. There are some downsides, however. This system took a minute to run and used 23 GB of RAM at it’s peak. There could be a memory leak or something going on in my program, but this is not scalable in its current state. If done correctly, this system can generate a word cloud that can be used to scrub data. Doing so would allow for greater and more accurate recommendations for companies once data is supplied and available. As mentioned above, this system combined with a neural network could be a very powerful unit used in large scale systems.