Coding Nomads: Capstone project proposal

Recommender systems:
A personalized movie experience

Student: Alberto P. Sosa
Date:April 2022

Domain Background

Description: Recommender systems are the norm of today's era; probably every major online service (store, streaming, service, etc.) uses a powerful recommender algorithm behind it to enhance user experience. What are recommender systems? How do they operate?
 A common definition of recommender systems states: *A recommender system calculates and provides relevant content to the user based on knowledge of the user,content, and interactions between the user and the item* (Falk,2019).
In general terms , recommender systems are a set of algorithms that filter information with the goal of suggesting relevant items to a given user.

Problem statement

As companies that offer services grow and competition increases, they need a way to maintain a loyal base of consumers. There are different methods to ensure that, such as lower prices, better quality, and **retention.**
Imagine there is a streaming website for movies, however in this site only movies that the user explicitly looks for will pop up and once the movie finishes people leave since there is not anything more to see. That does not sound right, correct?
Now imagine that exact website, but now at the homepage there is a list of the *most popular movies,* and once you have finished a movie there is a list of *similar movies* waiting for you. Suppose you watch Spider-man and did not know there is a second part? No problem, the system will show that second part.
With this system, after using the site for a while, you have a list of recommendations *just for you.*
Then, to clearly state the problem, given a set of users and items , predict what other items from the list the user will like.

Datasets and Inputs

The dataset used to simulate a database with movies and user information is the famous MovieLens (F. Maxwell Harper and Joseph A. Konstan, 2015) which contains information about 62K movies and 25M users (for the purposes of this project a reduced version will be used which contains about 10K movies and 100K users). The data is separated in two tables, one containing movie information such as name and the genres (although with some preprocessing of the data more features can be obtained). The other table contains information about the users such as the movies they have watched and the ratings they have assigned.
This information is enough to build a recommender system., From the features of the movie dataset we can find similar movies: if a user likes a movie, we can predict that they will like other films that are similar. With the information about the users it is possible to find similarities amongst them: we can assume that similar users will enjoy similar movies. . In addition, a combination of the two tables can give information about the user-item relationship.

Solution statement

The solution consists of two components: the filtering algorithm, and the machine learning model.

- Filtering algorithm: This will be the initial step to select from the whole dataset the movies that the user **may** like, however the list obtained in this step is not definitive: we will use an ML model to approve this filtered list. Before explaining the ML model, we explain the filtering algorithms:
    1. Content-based filtering: The purpose of this algorithm is to search for similar items, based on the features selected from the movie dataset.
       For example, a movie with the genres action, fantasy and comedy is similar to another movie with genres fantasy, comedy and mystery (from the point of view of a computer). This can be accomplished by calculating the distance between the vectors that represent the movie features .
       We will compare the following distance metrics : Cosine similarity, Pearson coefficient, euclidean distance.

       In addition, we plan to experiment with an NLP approach to search for similar plots between movies using their synopses (even if all the other features are different).

    2. Collaborative filtering: Instead of looking at similarities between movies, this approach looks for similarities between users. Similar to the previous approach, similar users are found by computing the distances between the representative features vectors.. Once similar users are identified, the algorithm finds the differences between the movies that each user has seen and then recommends the unseen films.

- Machine learning model: Once we have a filtered list, it is possible to try and predict the rating a user will give to each movie t, and then recommend those predicted to have high ratings. This is a regression problem,since our model will be predicting ratings and making recommendations out of those ratings.

Benchmark Model

The baseline model will be a recommender system based on popularity. This system does not offer personalized experience, however usually people like what is currently popular, and even this is a good approach when users are new to a website and they have not watched or rated many movies (in other words, we have little or no information about them).

Evaluation Metrics

Since the model will predict a value, RMSE and MSE will be the metrics used to measure the performance of the model, in addition there are some metrics proposed in this site(https://towardsdatascience.com/evaluation-metrics-for-recommender-systems-df56c6611093) to evaluate recommender systems. Given the metrics in the site, Coverage and personalization seem good metrics to see if our algorithm performs better than the popularity model .

- Coverage: This metric measures the percentage of items that are being recommended out of the total items. In other words, how many items are being considered at the time of making recommendations.
- Personalization: This measures the differences between the recommendations for each user. Since we want a personalized system for each user, the recommendations must be different and this metric is a way to measure how different they are.

Project Design

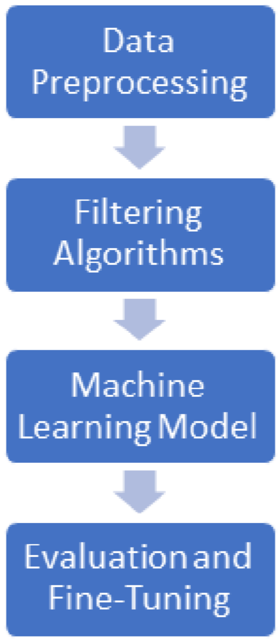

Table 1.0 Project flow Process

In summary, table 1.0 shows the process flow of the project.First, from the datasets used it is necessary to do some preprocessing (such as one-hot encoding, tokenization for NLP etc). Second, use this data as input to the filtering algorithms that as a result will produce a set of items the user may like.Third, feed this set of items to a machine learning model to predict what rating the user will give to them and based on those ratings recommend the highest rated items.Finally, evaluate the performance of the model with the proposed metrics and tune the model according to them.