MOVIE RECOMMENDATION

Project proposal Using MovieLens dataset

Table of content:

1. Introduction:

Recommendation systems are very important applications of Machine learning that aim to improve the quality of the experience provided to the internet user. There are many available movies everywhere that users can watch. Recommendation systems are an important part of suggesting items especially in streaming services. Netflix, for example, uses these systems to help users find the movies that they may like. Similarly, this paper is suggesting different approaches for The Movielens streaming service. Given a list of users, movies and ratings with their features, we want to build a recommendation system that takes the most informative features from the user's information and movies history to recommend for him/her other movies that he/she might enjoy.

1. Literature Review:

There are two well-known recommendations systems approaches which are the Collaborative Filtering and the Content Based Filtering. The Collaborative Filtering Approach is basically recommending new movies for the user based on how similar this user is to the rest of the users. The Content Based Filtering is basically focusing on the content of the movie being watched. So, if the user watched many Comedy movies, similar movies genres should be recommended. Many researchers contribute to this recommendation systems field by using different Machine and Deep learning algorithms to achieve higher accuracies. Lund and Ng (2018) use deep learning approaches in the Collaborative Filtering approaches and compared it to approaches like K-NN, and they got a lower RMSE. Vilakone & Park (2018) also used new approaches other than the K-NN and Cosine Similarity, k-clique methods and they called it the improved k-clique method. Similar projects and proposals suggested by Zhang, Yao et.al (2018) in the University of New South Wales. Most researchers are interested to beat the baseline and trivial approaches using the Deep learning and improved models approaches. In this paper, we are interested in the Collaborative Filtering Approach as well. We will use different approaches other than the ones suggested by the authors. Moreover, we will shift the problem into a classification problem, whether the user will dislike or like the movie.

2. Dataset Description:

Data Source: http://www.grouplens.org/node/73

This is the MovieLens Dataset. It has three tables: Users, Movies and Ratings. The Users data provides us with some information about the current MovieLens Users. The Movies data provides us with information about the current Movies in the Movielens platform. The ratings data is the users' ratings for the movies they watched.

A. Users:

The users data consists of some information about the current users the MovieLens have.

Variables Description:

- User ID : Unique ID for the user
- Gender : 'F' for female , 'M' for male.
- Age : 6 different values representing 6 different age ranges.
    - 18 represent age in the range "18-24"
    - 25 represent age in the range "25-34"
    - 35 represent age in the range "35-44"
    - 45 represent age in the range "45-49"
    - 50 represent age in the range "50-55"
    - 56 represent age in the range "56+"
- Occupation: 20 Different fields cover all occupations in the Dataset
    - 0: "other" or not specified
    - 1: "academic/educator"
    - 2: "artist"
    - 3: "clerical/admin"
    - 4: "college/grad student"
    - 5: "customer service"
    - 6: "doctor/health care"
    - 7: "executive/managerial"
    - 8: "farmer"
    - 9: "homemaker"

    - 10: "K-12 student"
    - 11: "lawyer"
    - 12: "programmer"
    - 13: "retired"
    - 14: "sales/marketing"
    - 15: "scientist"
    - 16: "self-employed"
    - 17: "technician/engineer"
    - 18: "tradesman/craftsman"
    - 19: "unemployed"
    - 20: "writer"
- Zip code

B. Ratings:

This dataset provides information about the users' ratings for the movies they watched.

- UserID: Unique
- MovieID:Unique
- Rating : on scale from 0 to 5
- Timestamp

C. Movies:

This dataset provides information about the movies' features.

- Movie ID: Unique
- Movie Name: has the format Movie Name (Year)
- Year of Production
- Genres: Comma separated values representing the different genres of the movie (The dataset has 20 different genres)
    - Action
    - Adventure
    - Animation
    - Children's
    - Comedy
    - Crime

- Documentary
- Drama
- Fantasy
- Film-Noir
- Horror
- Musical
- Mystery
- Romance
- Sci-Fi
- Thriller
- War
- Western

3. Features Selection and Exploratory Data Analysis (EDA):

The features that we are going to use are:

From User Data: [age, occupation, gender, Zip Code]

From Ratings Data: Rating

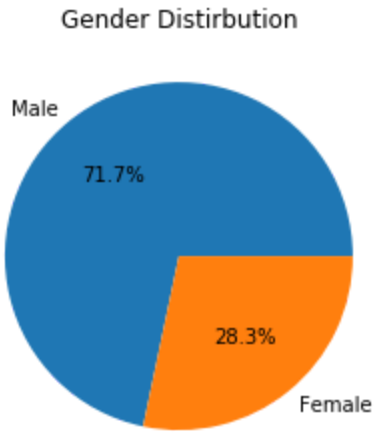
From Movies Data: [Genres, Year of Production]

In this section, we analyzed each variable separately, which helped us to understand our dataset more. We started by visualizing our data and notice the distribution of each variable.

A. Users Data

1.1. Gender

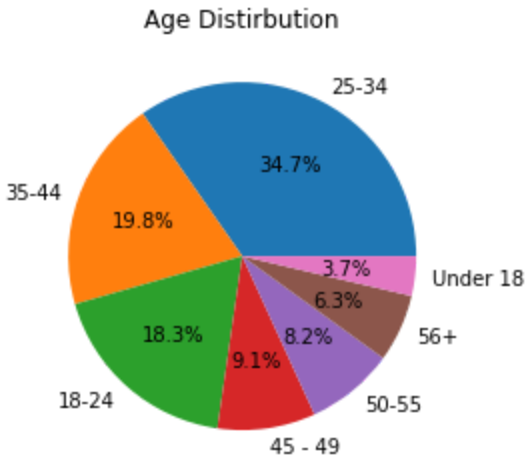
The below figure shows that the users data has more males (71.7%) than females (28.3%). We have unbalanced data. It might affect our Recommendation system. Various oversampling techniques will be performed in the next phases.



1.2. Age

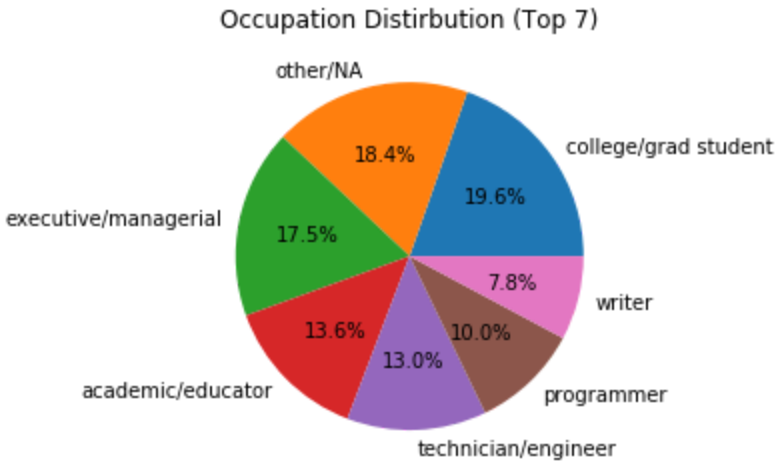
The below figure shows that the 25-34 years old category has the highest number of users (34.7%) among the other age categories. It was unexpected to see the

"Under 18" category has the lowest percentage of users. However, since MovieLens is an online platform. It might be the case that users below 18 years old do not really want to mention their true ages to avoid parental control issues or the fear of being prevented from creating their own accounts or being labeled as underage. As for the other categories, it was expected to see low percentages of users in the 45-49, 50-55, 56+ categories since they might not be interested in watching movies as the 18-24, 25-34 and 35-44 users do.

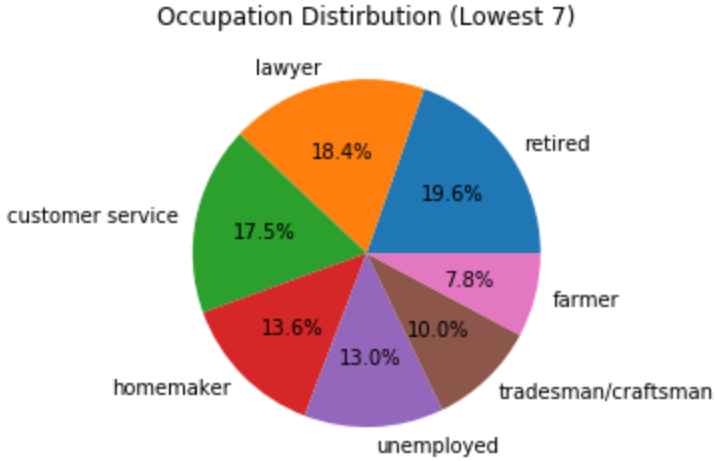


1.3. Occupation

The below figure shows the top 7 Users Occupation out of 20. Given the age distribution with the 25-34 is the highest percentage, it is reasonable to see that the college/grad student has the highest percentage (19.6%). Moreover, the college/ grad represents a significant part of the population. Other categories have lower percentage simply because they are not present in the population with the same percentage as students are. However, it is interesting to see 7.8% writers among the top 7 Occupations, writers are not present in the actual population as graduate students are. Once again we have an unbalanced data problem. Moreover, many users did not mention their Occupation (other/NA), this proves the point that it might be the case that the below 18 years users did not want to record their true ages.

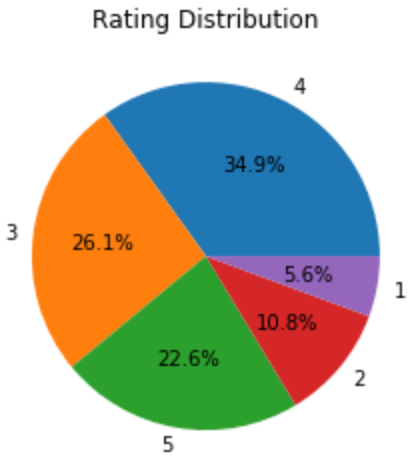


The below chart shows the lowest 7 Users Occupation. It was expected to see "unemployed" as one of the lowest 7 Occupation status (13%). People usually feel ashamed mentioning this type of information. It's also not surprising to see such categories.



B. Rating Data

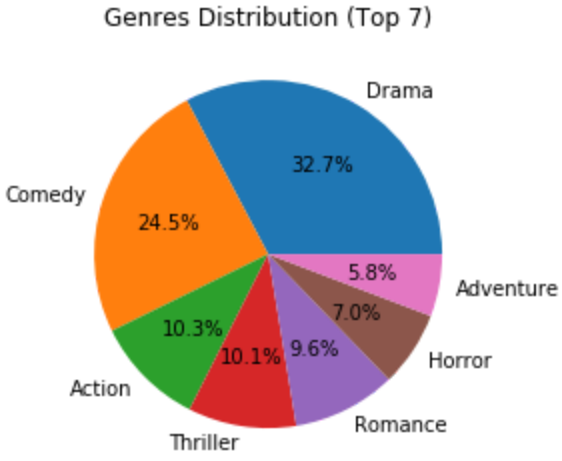
From the figure below, most users give the movies a rating of "4" (34.9%), and It is not common that the users give a rating of "1" (5.6%). The ratings data is negatively skewed.



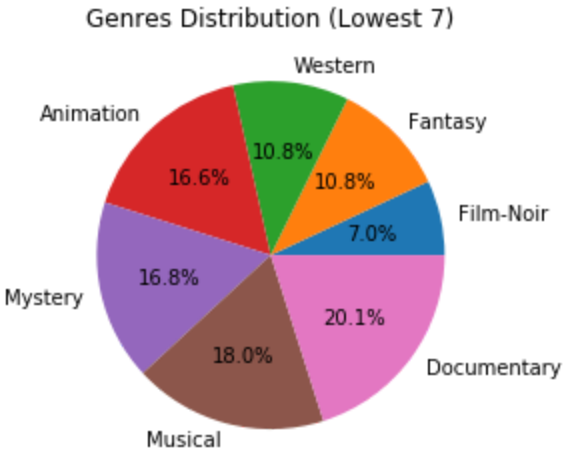
C. Movies Data

1.4. Genres Distribution

The figure below shows the Top 7 genres out of 18 Genres. As we can see, most of the movies have Drama (32.7%) and Comedy (24.5%) scenes. We can see the low percentage of Adventure (5.8%) and Horror (7.0%) because it is not common to have an Horror movie that can be labeled as “Comedy” or “Drama”.



It’s expected to have these categories as the lowest 7 since they are not as popular as “Drama” and “Comedy”.



D. Association measures

In this section, we will compute the Chi-Square Association measure, since all our features are categorical variables. We will declare that two features are independent if the evaluated p-value is less than 5%. (Reject the Null Hypothesis)

Action: 1, Adventure: 2, Animation: 3, Children's: 4, Comedy: 5, Crime: 6, Documentary: 7, Drama: 8, Fantasy: 9, Film - Noir: 10, Horror: 11, Musical: 12, Mystery: 13, Romantic: 14, Sci- Fi: 15, Thriller: 16, War: 17, Wester: 18

	1	2	3	4	5	6	7	8	9
1									
2	0								
3	0.007	0.02							
4	0.0002	0	0						
5	0	0	0.13	0.03					
6	0	0.11	0.02	0	0				

7	0	0	0.10	0	0	0.01			
8	0	0	0	0	0	0.73	0		
9	0.038	0	0.006	0.14	0.69	0.24	0.23	0	
10	0.019	0.32	0.77	0	0	0.002	0.4	0	0.76
11	0.001	0.94	0.007	0	0	0.02	0.5	0	0.02
12	0.001	0.1	0	0.08	0.28	0.003	0.1	0	0.72
13	0.72	0.2		0.07	0	0	0	0.02	0.3
14	0	0	0.01	0	0	0	0	0.36	0.78
15	0	0	0.99	0.4	0	0.02	0	0	0
16	0	0.4	0.009	0	0	0	0	0	0.01
17	0	0.7	0.47	0.02	0	0.006	0.12	0.004	0.5
18	0.8	0.8	0.3	0.34	0.35	0.08	0.23	0	0.5

	10	11	12	13	14	15	16	17
10								
11	0.2							
12	0.48	0.01						
13	0	0.32	0.13					
14	0.07	0	0.28	0.91				
15	0.711	0	0.04	0.69	0			
16	0	0.01	0	0	0	0		
17	0.37	0	0.72	0.075	0.57	0.9	0.01	
18	0.75	0.02	0.3	0.31	0.075	0.53	0	0

This table shows each feature and their highly associated features.

Note: If feature "Action" is associated with "Mystery". It will not be stated that "Mystery" is associated with Action in the table.

Feature	**Associated with**
Action	Mystery, Western
Adventure	Crime, Film-Noir, Musical, Mystery, Romance, Thriller, Western
Animation	Comedy, Documentary, Film-Noir, Mystery, Sci-Fi, War, Western
Children's	Film-Noir, Mystery, Sci-Fi, Western
Comedy	Fantasy, Musical, Western
Crime	Drama, Fantasy, Western
Documentary	Fantasy, Film-Noir, Musical, Mystery, War, Western
Drama	Romance
Fantasy	Film-Noir, Musical, Mystery, Romance, War, Western
Film-Noir	Horror, Musical, Sci-Fi, War, Western
Horror	Mystery
Musical	Mystery, Romance, War, Western
Mystery	Romance, Sci-Fi
Romance	War
Sci-Fi	War, Western
Thriller	-
War	Western

Initial Results

The Movie recommendation problem could be dealt with as a classification problem where the model classifies the users' preferences on a specific movie whether the user will like it or not or it can also predict the rating of the user to the specific movie in this section we will showcase different models and their performance on the Movielens dataset

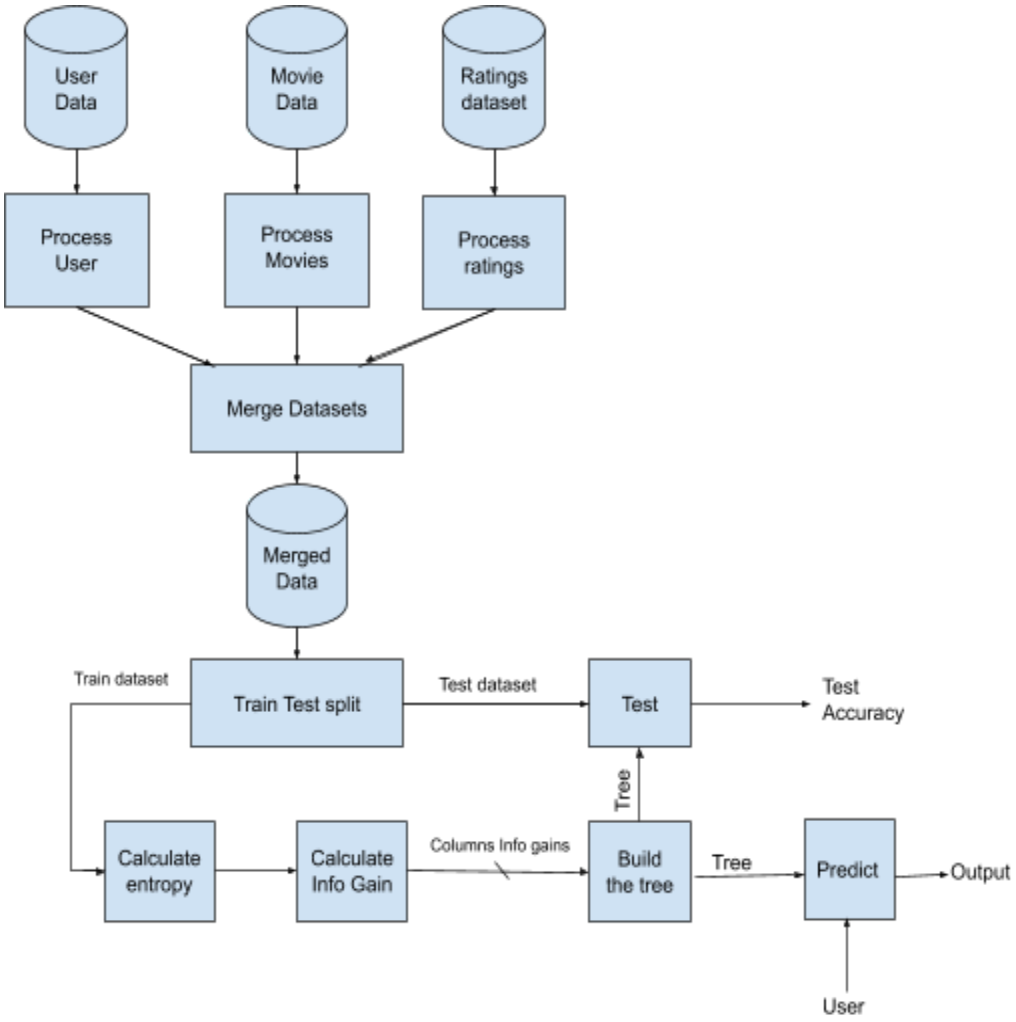
Model	**Accuracy**	**Pros**	**Cons**
Knn-Collaborative Filtering	35%	-It takes into consideration other users behaviour and provides a recommendation based on that	- Very computationally expensive - Low Accuracy - Cold-Start: It doesn't work with cold-start user or items, since the dot product will be all 0s. It can't recommend anything. - Sparsity: Similarly, it doesn't work with sparse data, since the intersection between 2 users is 0, the dot product is also 0 - Scalability: we need to calculate the user similarity or item similarity matrix. This is a large matrix that doesn't scale with a large number of users, movies
Logistic regression	40%	-Not computationally expensive -Simple	-Logistic regression requires s the observations to be independent of each other -assumes linearity of independent variables and log odds which is very hard to guarantee in case of the movie lens
Neural Network	60%	-High accuracy -Ability to capture complex functions	- Complex - since the number of features is small there's a great tendency for overfitting the data
Decision tree	60%	Simple and computationally inexpensive (for predicting) once the tree is built	-Doesn't perform well in unexpected events
Random forest	64%	High accuracy for the training data	The random forest algorithm tends to overfit and will have low performance for an external data

Other suggested models:

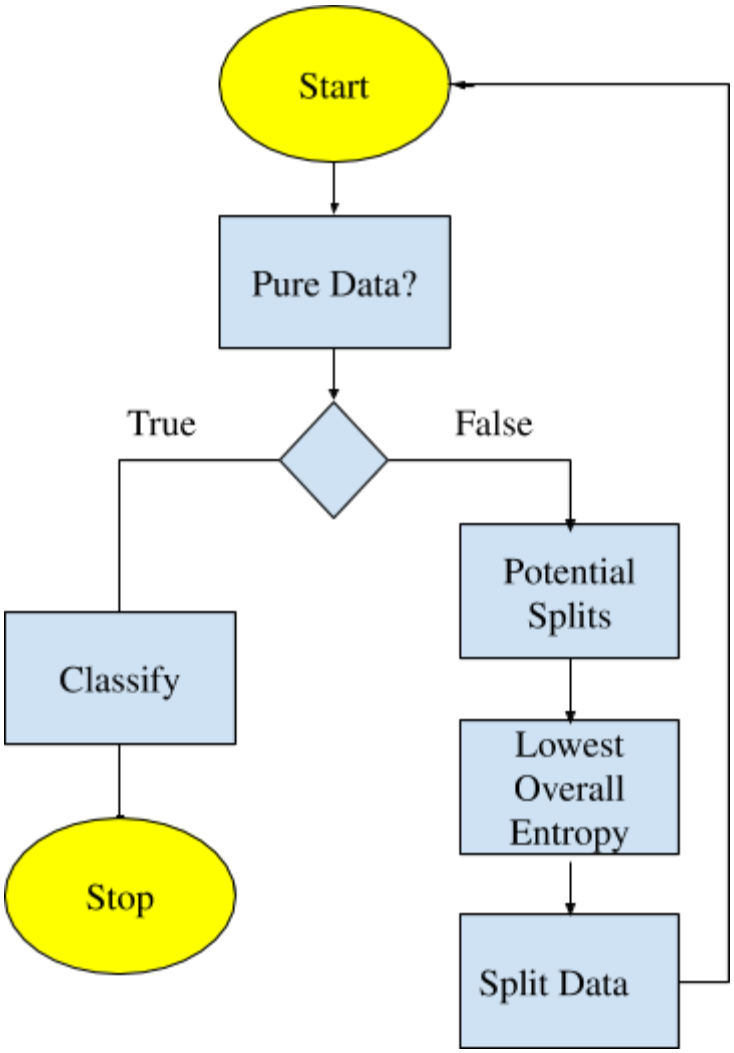
Model	Description	Accuracy (RMSE)	Pros	Cons
Mean approach	get the average rating of each movie across all users. In the testing data, if the movie exists in the training data, we get the average of its users' ratings. Otherwise, we return 3.0 (normal average rating).	0.98	Simple approach	- Affected by outliers.
Weighted Mean Approach	Compute an nxn cosine similarity matrix (n is the number of users in the training matrix). Then we use this matrix to get the weighted average of the users' ratings which will be compared to the existing movies in the testing data.	0.97	- Simple Approach -Not assuming that users are equally similar to each other.	- Not robust
Gender - Demographics	For each Movie ID we get the average rating per gender as we believe it's a significant feature from the users' info.	0.98	Using users' features not only their movie ratings.	-Not robust

Singular Value Decomposition (SVD)	It's a linear algebra approach. What it does is that it decomposes a matrix into constituent arrays of feature vectors corresponding to each row and each column. It will take the training matrix and the parameter `k` which is the number of features into which each user and movie will be resolved into.	0.92	-It is a dimensionality reduction technique, and it avoids features dependency. -Robust approach.	-It may lead to some amount of data loss.

Structure of the model:



Decision tree implementation:



Design Choices and tradeoffs:

1- Input Choices and tradeoffs:

We started with a feature set that has more than 60 features including:

1) The user data [Age, gender , occupation , state(zip code)] (one hot encoded)
2) The user interests scores which had 19 interest scores representing the user interests in all the different 19 movie genres
3) The movie data which had a 19 one hot encoded features representing the movie genre(s)
4) The movie production year

After research the the following improvements were made to Improve the previously mentioned data:

1- Removing columns with very low information gain. (less than a 0.0001)

Most of the movie insterest scores had very low Information gain because the majority of the values were zeros because most of the users have only one rating and therefore it is very hard to determine their interests.
The following is a table with the features and their corresponding information gains

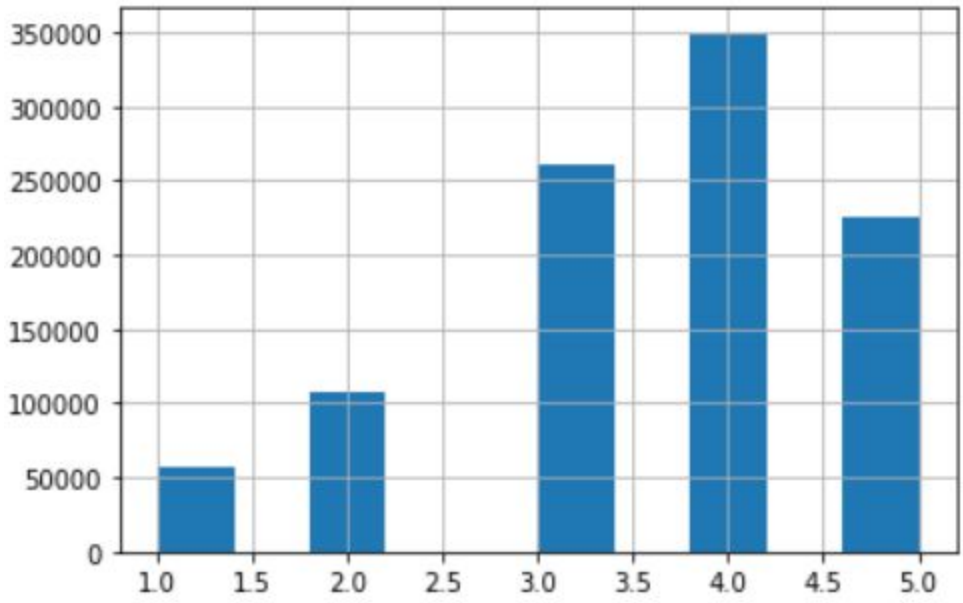
Feature	Information Gain
Age	0.00138
Gender	0..000234
Year of production	0.01313
Genre_1	0.0137
Genre_2	0.006
Occupation	0.001498
State	0.00221

2- Trying label encoding instead of one hot encoding for the most important features (such as the top two movie genres representing the movie)
This solves the sparsity problem of the one hot encoding method however it has another potential problem because it assumes an order or a rank but this problem is not a major problem because the decision tree algorithm doesn't depend on the numerical values of those labels
3- Converting numerical values such as age and movie production year to bins where every bin represents a range of values
Instead of having more than 60 possible values for the age we only have 4 which means less tree branches and better generalizability :
Child: representing the age from (1:12) years old
Teen: representing the age from (13:19) years old
Adult: representing the age from (20:45) years old
Old: representing the age from (45:69) years old
Instead of having 80 values for the movies production year we divided it into 10 bins representing the decade:
1920's: including years between [1910-1920]
1930's: including years between [1921-1930]]
And so on

Output Choices and tradeoffs:

For the output there are different design choices: the first design choice is to predict the movie rating and return a value of [1, 2, 3,4 or 5]

The problem with this design choice is the distribution of the dataset shown below.



This distribution shows that the percentage of rating= 1 is around 5%
And the percentage of instances where rating = 2 is around 10%
Whereas the percentage of instances where rating = 4 is around 35%
And the percentage of instances where rating = 3 is around 27%
And the percentage of instances where rating = 5 is around 22%
This unbalanced ratings problem could be solved by selecting only 5% randomly from each rating however this will decrease the number of ratings significantly and the effectiveness of the model
The other option was to split the ratings into two values (Likes and dislike) where like is represented by a 1 and dislike is represented by a zero.
The choice of the threshold is really tricky because the ratings are discrete values so there are no ratings with value 2.5 and therefore we have to split the ratings at either the value 2 or 3.
If we chose the likes to be [1,2] this will re-create the unbalanced data problem since the dislikes will be only 15% of the data. However, if we chose the dislikes to be at [1, 2, 3] an the likes to be at [4,5] this will create a perfectly balanced data but the accuracy and recall drop significantly as shown in the table

Likes range	Dislikes = [1,2]	Dislikes = [1,2, 3]
TP	15637	7208
FP	3062	5321
TN	1081	3161
FN	219	4309

Accuracy	83.5%	51%
Recall	83.6%	57%

So the output of the model is equal to zero (dislike) if the rating is equivalent to 1 or 2 and the output is one (Like) otherwise.

After implementing the previous adjustments we ended up with 7 features that represent most of the important 60 features and we ended with a training time equal to 1 hour and 20 minutes to build the ID3 decision tree based on 666564 (~ 600K) training instances.

Phase 5:

User Input

The user is expected to feed the application with the following inputs:

1- Gender
2- Age
3- Movie production year
4- Occupation (Selected)
5- ZipCode
6- First Movie Genre (Selected)
7-Second Movie Genre if exists (Selected)

** A better design choice is to let the users enter the movie title and retrieve the movie genres from the table however this choice given this dataset is not applicable because the movie titles are full of noise such as repeated characters, incorrect spelling and different non alpha characters that are very hard to determine if they belong to the movie name for example Toy Story 3 has the number '3' which distinguishes it from other toy story movie versions and other movies are written in this format (JOK3R) where the 3 represents the letter 'E' and cleaning the titles requires human analysis which is hard to implement given the time constraint and the size of the data. So the prediction is whether the user will like the movies based on the movie genres and the user data.

However this design choice allows the user to get the prediction for movies that don't even exist in the movie dataset.

User input snippet:

(1)

```
Please enter your Gender F for female and M for male :  F
Please Enter your age :  1
Please enter the production year :  1990
Choose the movie genre(s) :
Choose   0   for  Action
Choose   1   for  Adventure
Choose   2   for  Animation
Choose   3   for  Children's
Choose   4   for  Comedy
Choose   5   for  Crime
Choose   6   for  Documentary
Choose   7   for  Drama
Choose   8   for  Fantasy
Choose   9   for  Film-Noir
Choose   10   for  Horror
Choose   11   for  Musical
Choose   12   for  Mystery
Choose   13   for  Romance
Choose   14   for  Sci-Fi
Choose   15   for  Thriller
Choose   16   for  War
Choose   17   for  Western
First Genre  :  1
Second Genre  :  3
Please Enter your ZipCode :   48067
```

(2)

```
Please select your occupation


0:  other or not specified
1:  academic/educator
2:  artist
4:  clerical/admin
4:  college/grad student
5:  customer service
6:  doctor/health care
7:  executive/managerial
8:  farmer
9:  homemaker
10:  K-12 student
11:  lawyer
12:  programmer
13:  retired
14:  sales/marketing
15:  scientist
16:  self-employed
17:  technician/engineer
18:  tradesman/craftsman
19:  unemployed
20:  writer
 19
```

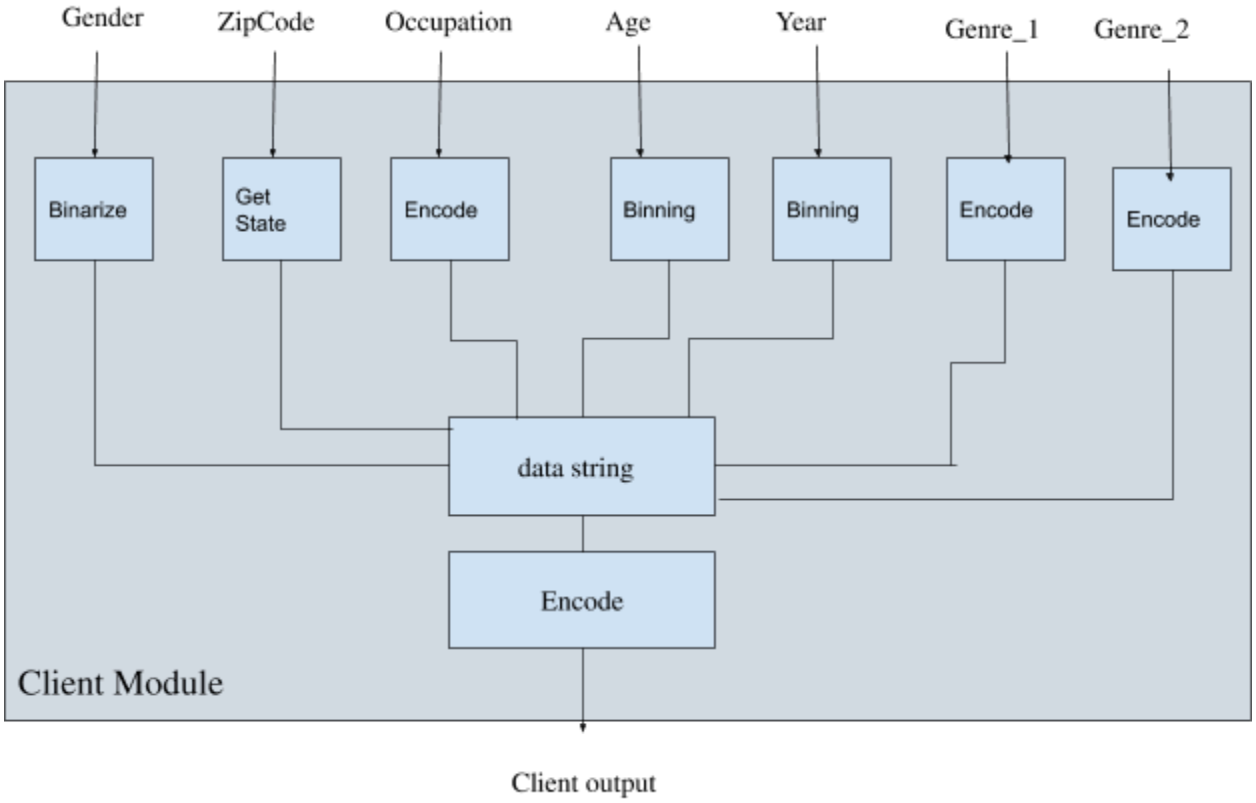
The output:
The output either suggests that the user watches the given movie or suggest that the user tries another one based on the value of the prediction (0 or 1)

```
Go Ahead you will like this movie!
```

Client :

Client Design:

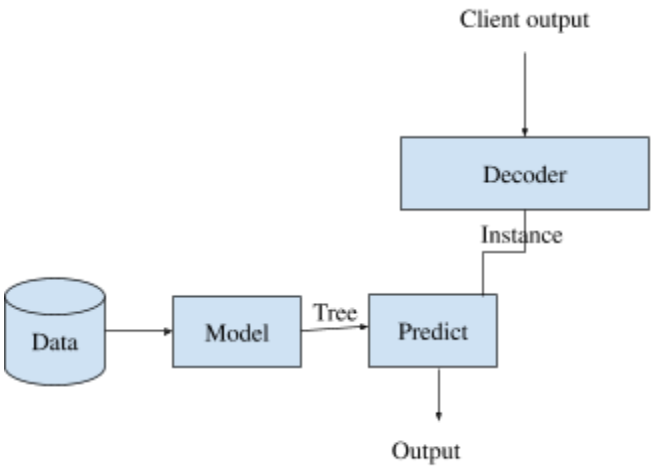
The client takes the input and processes it and provides in to the server in the form of an encoded string and the server takes the string , decodes it and feed it to the model so and provides a prediction for the user to either watch the movie or not.



1- Binarize: A function responsible for taking a categorical variable and returning 0 or 1 based on a threshold fed to the function (0 if value < threshold) and (1 if value > threshold)
2- Get state: function that uses the Uszipcode package to return the state.
3- Encode: Takes a value and returns a code to that value based on label encoding
4- Binning: takes a value and returns the bin that this value belongs to based on bin ranges and the value

Server:

Server Design:



The Model is described above and the output is either to watch or not to watch the movie.

Code:

Client Server Application:

https://github.com/mjaafar97/Client-Server-implementation-of-ID3

Model Design Code:

https://github.com/mjaafar97/MoviesRecommendationSystem

Demo:

MovieRecommenderDemo

Sources: