SVKM's Narsee Monjee Mukesh Patel School of Technology Management & Engineering, NMIMS

DOCUMENTATION

FOR

Analysing The Relation Between Users And Genres

A Mini Project for Predictive Modelling

By,

Team: E003, E015, E019, E020, E024, E028

College: Mukesh Patel School of Technology Management & Engineering, NMIMS

Goal / Problem Statement

Movies are classified by the genre, which is dictated primarily by how the movie is presented to the audience, and what might be the plot and nature of the film. Throughout the world, and the huge population, people have different choices, tastes and preferences are governed by a variety of factors. Some of these factors are trivial in terms of influence, while some of them have a bigger influence. In the dataset offered to us by GroupLens - MovieLens100K, we take a variety of data points that capture elements like Movie Ratings, the demographic (gender, location, zipcodes, state of residence) of the users of the website who offer those ratings as per timestamps, genres and a few more data points.

So, in this data exploration, we seek to accomplish a few objectives like -

1. Observing various rating patterns with respect to location, user demographics, age groups and other factors.
2. Constructing expressive visualizations that offer visible trends that influence genre selection, ratings, movie choices and many more.
3. Creating a prediction model that helps us identify the favourite genre of a user demographic that takes into account the age groups, occupation, location and gender.

Understanding the Training Data at hand

The dataset provided to us is the MovieLens100K. MovieLens datasets were collected by the GroupLens Research Project at the University of Minnesota. This data set consists of:

- 100,000 ratings (1-5) from 943 users on 1682 movies.
- Each user has rated at least 20 movies.
- Simple demographic info for the users (age, gender, occupation, zip).

The data was collected through the MovieLens web site (movielens.umn.edu) during the seven-month period from September 19th, 1997 through April 22nd, 1998.

Here are brief descriptions of the data.

`ml-data.tar.gz` -- Compressed tar file. To rebuild the `u.data` files do `gunzip ml-data.tar.gz, tar xvf ml-data.tar, mku.sh`.

`u.data` -- The full `u.data` set, 100000 ratings by 943 users on 1682 items. Each user has rated at least 20 movies. Users and items are numbered consecutively from 1. The data is randomly ordered. This is a tab-separated list of - `user id | item id | rating | timestamp.` The timestamps are UNIX seconds since 1/1/1970 UTC

`u.info` -- The number of users, items, and ratings in the `u.data` set.

`u.item` -- Information about the items (movies); this is a tab-separated list of `movie id | movie title | release date | video release date | IMDb URL | unknown | Action | Adventure | Animation | Children's | Comedy | Crime | Documentary | Drama | Fantasy | Film-Noir | Horror | Musical | Mystery | Romance | Sci-Fi | Thriller | War | Western |`. The last 19 fields

are the genres, a 1 indicates the movie is of that genre, a 0 indicates it is not; movies can be in several genres at once. The movie ids are the ones used in the u.data data set.

u.genre -- A list of the genres.

u.user -- Demographic information about the users; this is a tab-separated list of user id | age | gender | occupation | zip code. The user ids are the ones used in the u.data data set.

u.occupation -- A list of the occupations.

u1.base -- The data sets u1.base and u1.test through u5.base and u5.test

u1.test are 80%/20% splits of the u.data into training and test data.

u2.base Each of u1, …, u5 have disjoint test sets; this if for

u2.test 5 fold cross-validation (where you repeat your experiment

u3.base with each training and test set and average the results).

u3.test These data sets can be generated from u.data by mku.sh.

u4.base

u4.test

u5.base

u5.test

ua.base -- The data sets ua.base, ua.test, ub.base, and ub.test

ua.test split the u.data into a training set and a test set with ub.base exactly 10 ratings per user in the test set. The sets ub.test, ua.test and ub.test are disjoint. These data sets can be generated from u.data by mku.sh.

allbut.pl -- The script that generates training and test sets where
all but n of a users ratings are in the training data.
mku.sh -- A shell script to generate all the u.data sets from u.data.

Data Cleaning

We have taken an additional data frame `zip_data` for gathering information about location details of the users as well.

zip	type	decommissioned	primary_city	acceptable_cities	unacceptable_cities	state	county

Zipcodes consist of integer values which range from three-digit values to five digit values. For making the data easier to operate upon, the `zip` column was converted to string type and 'zeroes' were appended to make a constant digit count of five.

A new table mapping has been created by merging the user data `(users_data)` and the modified zip code data by using zip as the key to merging the two tables. After dropping irrelevant data column from the table, we obtain the following table

UserID	Gender	Age	Occupation	zip	state	county	timezone

On the `ratings_data` table, the mean rating is calculated for users based on the number of reviews given per user. Along with this, the mean rating of movies is calculated per movie and this data column is merged with the `movie_data` table which gives us the following output:

	MovieID	Title	Genres	Rating
3701	3948	Meet the Parents (2000)	Comedy	3.635731

Here, the last column represents the mean rating for the movie.

One problem which we faced with the movies data is that one movie had multiple genres, which was represented in the following format:

4	Comedy
5	Action|Crime|Thriller
6	Comedy|Romance

Therefore, there was a need to divide up all the genre columns, and hence each genre was made into a separate column, and binary values indicated whether the particular movie consisted of the genre or not.

	MovieID	Title	Genres	Rating	Action	Adventure	Animation	Children's	Comedy	Crime	...	Fantasy	Film-Noir	Horror
0	1	Toy Story (1995)	Animation|Children's|Comedy	4.146846	0	0	1	1	1	0	...	0	0	0

By doing this step, it makes it easier for us to analyse movies, ratings even if they have multiple genres.

Next, the average rating is calculated according to state, according to occupation and gender:

state	Rating
AE	3.658979
AK	3.985730

Occupation	Rating
0	3.537544
1	3.576642

Gender	Rating
F	3.620366
M	3.568879

Now, average movie rating is merged along with the new table which consists of user data and movie data. This gives us the final table:

UserID	Gender	Age	Occupation	zip	state	county	timezone	MovieID	Rating_x	Timestamp	Title	Genres	Rating_y

Finally, the genre columns are again separated into different columns and given binary indicators as explained above.

	UserID	Gender	Age	Occupation	zip	state	county	timezone	MovieID	Rating_x	Timestamp	Title	Genres	Rating_y	Action	Adventure	Animation	Children's	Comedy	Crime	Documentary	D
0	1	F	1	10	48067	MI	Oakland County	America/Detroit	1193	5	978300760	One Flew Over the Cuckoo's Nest (1975)	Drama	4.390725	0	0	0	0	0	0	0	
1	1	F	1	10	48067	MI	Oakland County	America/Detroit	661	3	978302109	James and the Giant Peach (1996)	Animation|Children's|Musical	3.464762	0	0	1	1	0	0	0	
2	1	F	1	10	48067	MI	Oakland County	America/Detroit	914	3	978301968	My Fair Lady (1964)	Musical|Romance	4.154088	0	0	0	0	0	0	0	
3	1	F	1	10	48067	MI	Oakland County	America/Detroit	3408	4	978300275	Erin Brockovich (2000)	Drama	3.863878	0	0	0	0	0	0	0	

Ready Data Summary

	UserID	MaxGenre	Gender	Age	Occupation	state
0	1	Drama	F	1	10	MI
1	2	Drama	M	56	16	LA
2	3	Comedy	M	25	15	MN
3	4	Action	M	45	7	MA
4	5	Drama	M	25	20	MN

Once we have all of the above data, we want to use that to create a final table that would have the features that we want to use for predicting the target output or the target variable, which in our case are: Gender, Age, Occupation and state. Another column that we would need here is the target variable itself, which is necessary for the supervised learning algorithms that we will be using.

We generate this table by merging data from our 3 tables, namely User Data, Movie Data and Ratings Data, after all the data cleaning and preprocessing is done. To generate the target variable column, i.e. the MaxGenre column, we take the genre where each user has watched more movies as compared to other genres. We finally convert the genres, states and gender column to numerical form.

We use this table shown below for all of our prediction models.

	UserID	MaxGenre	Gender	Age	Occupation	state
0	1	7	0	1	10	25
1	2	7	1	56	16	21
2	3	4	1	25	15	26

Model Information

For prediction, we made use of three models that allowed us to compare within accuracies and prediction results. These models were -

1) Decision Tree [DT]
2) Logistic Regression [LR]
3) Support Vector Machine [SVM]

Let's detail these models and their principle below.

Decision Trees [DT]

A decision tree is a flowchart-like tree structure where an internal node represents a feature (or attribute), the branch represents a decision rule, and each leaf node represents the outcome. The topmost node in a decision tree is known as the root node. It learns to partition on the basis of the attribute value. It partitions the tree in a recursive manner called recursive partitioning. This flowchart-like structure helps you in decision making.

The basic idea behind any decision tree algorithm is as follows:

1. Select the best attribute using Attribute Selection Measures(ASM) to split the records.
2. Make that attribute a decision node and breaks the dataset into smaller subsets.
3. Starts tree building by repeating this process recursively for each child until one of the condition will match:
    - All the tuples belong to the same attribute value.
    - There are no more remaining attributes.
    - There are no more instances.

Logistic Regression [LR]

Logistic regression is a classification algorithm. It is a generalized linear model that we can use to predict categorical outcome variables. For example, logistic regression can be used to classify

whether someone will be denied or approved for a loan based on his existing features (such as salary, debts, savings etc.). In logistic regression, we're essentially trying to find the weights that maximize the likelihood of producing our given data and use them to categorize the response variable.

Decision boundary helps to differentiate probabilities into positive class and negative class. It can be linear or nonlinear, of any shape. We train the model to match the decision boundary to separate one class from another. In Logistic regression, it is assumed the Decision boundary is linear and the model will be trained to find the best linear decision boundary to separate between classes.

Support Vector Machines [SVM]

Support Vector Machine (SVM) is a supervised machine learning algorithm which can be used for both classification or regression challenges. However, it is mostly used in classification problems. In the SVM algorithm, we plot each data item as a point in n-dimensional space (where n is the number of features you have) with the value of each feature being the value of a particular coordinate. Support Vectors are simply the coordinates of individual observation. The SVM classifier is a frontier which best segregates the two classes (hyper-plane/ line).

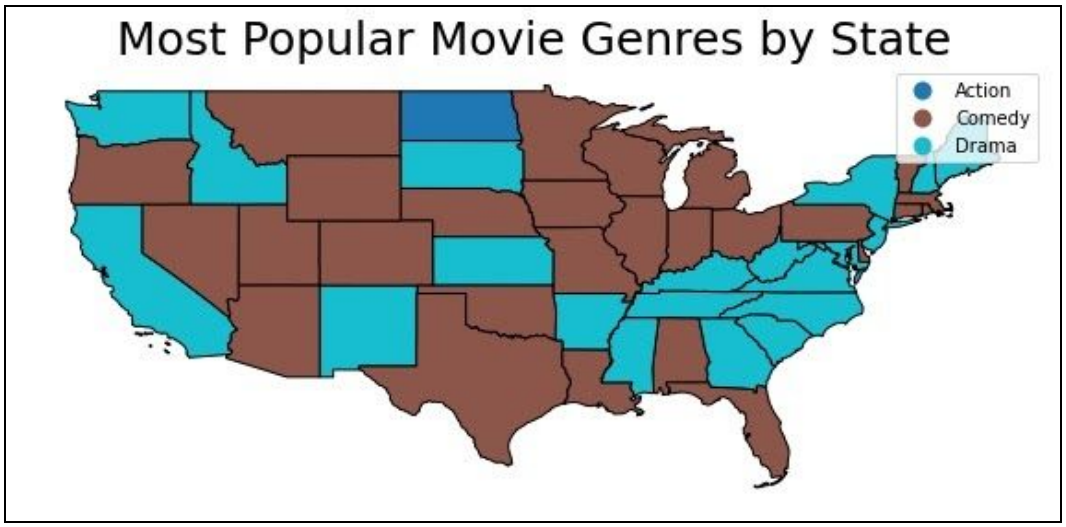
In the SVM classifier, it is easy to have a linear hyper-plane between these two classes. But, another burning question which arises is, should we need to add this feature manually to have a hyper-plane. No, the SVM algorithm has a technique called the kernel trick. The SVM kernel is a function that takes low dimensional input space and transforms it into a higher dimensional space i.e. it converts not separable problem to separable problem. It is mostly useful in non-linear separation problems. Simply put, it does some extremely complex data transformations, then finds out the process to separate the data based on the labels or outputs you've defined.

Input & Output

This section will briefly show what all visualizations, graphs, trends we achieved and how the prediction models help us predict the favourite genre for a user demographic.

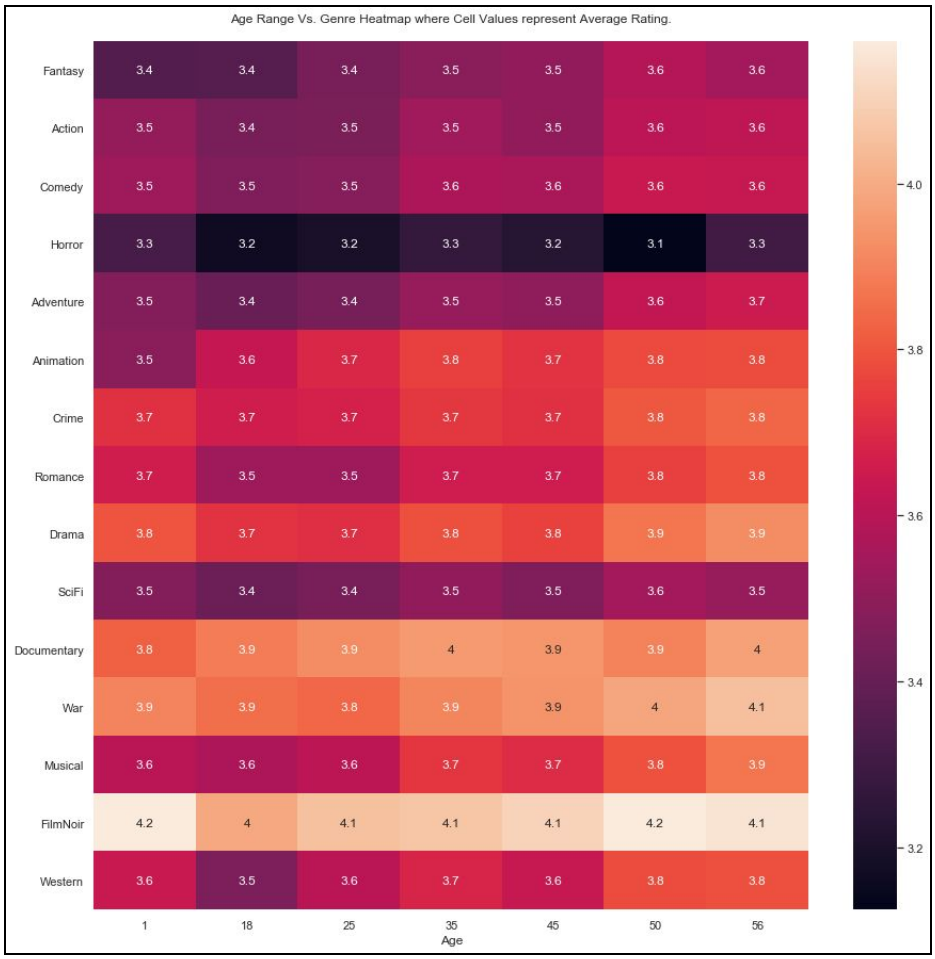
DATA VISUALIZATIONS

1) POPULAR MOVIE GENRE BY STATE



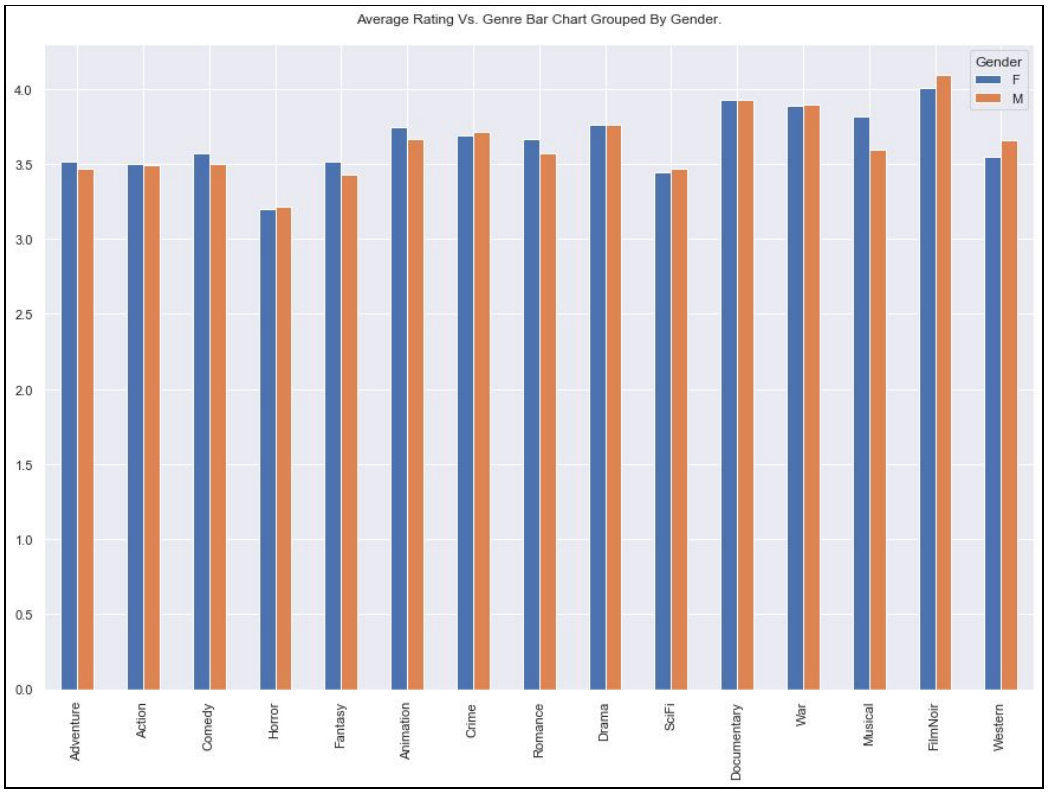
In this visualization, we use the map of the United States and then aggregate the user data together and find the most popular genres in each state.

2) AGE GROUP VS. GENRE



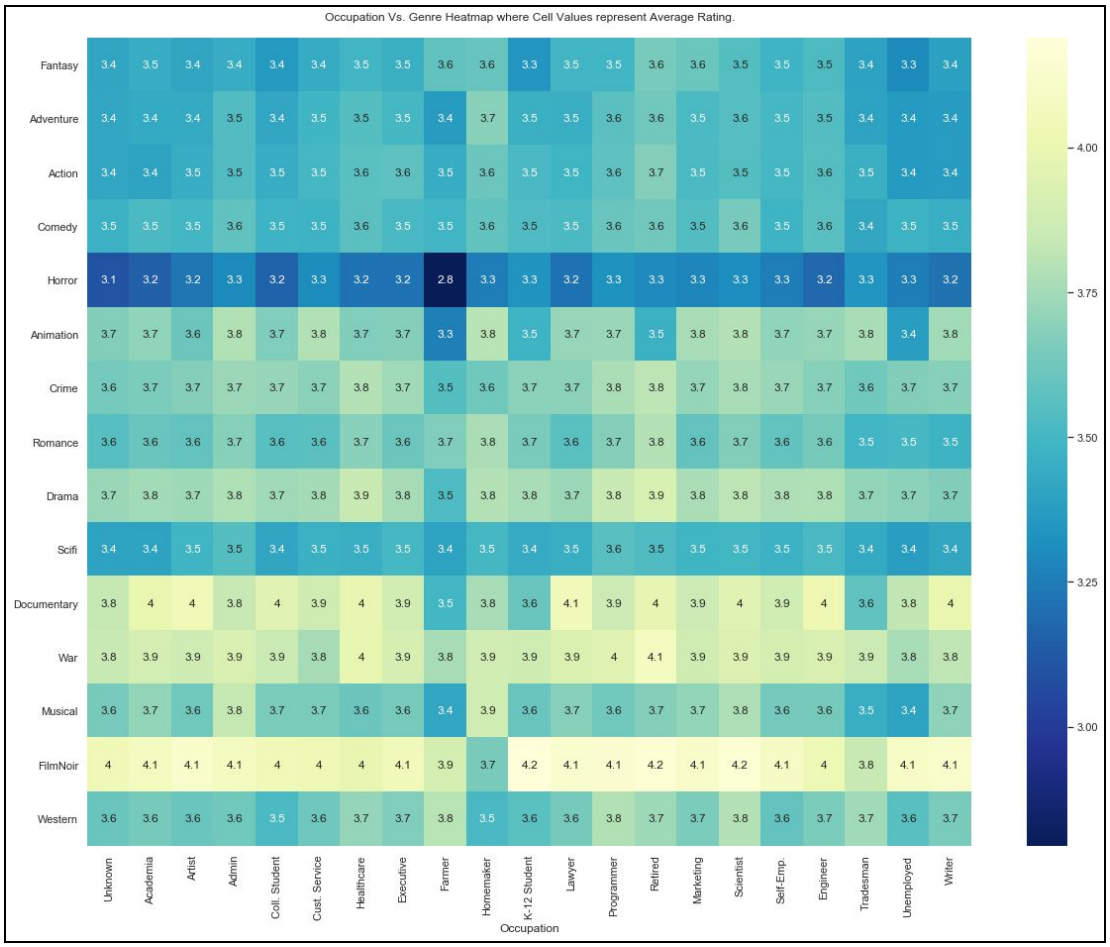
In this visualization, we use a Heat Map to map how strongly various age groups prefer the various genres.

3) GENDER VS. GENRE



In this visualization, we use a Bar Graph to see how the two genders rate movies from various genres.

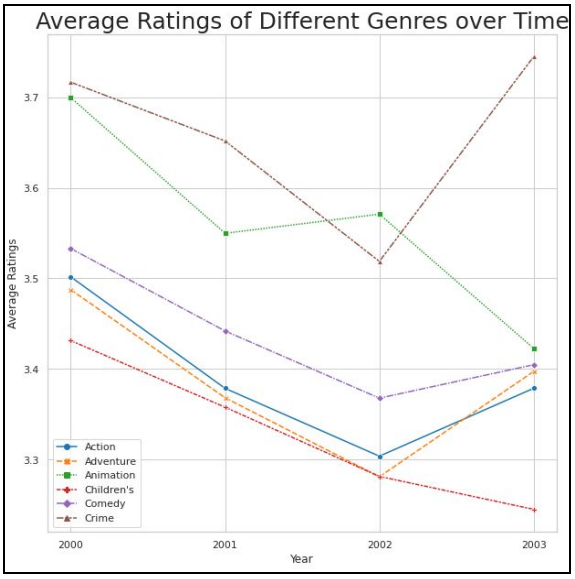
4) OCCUPATION VS. GENRE

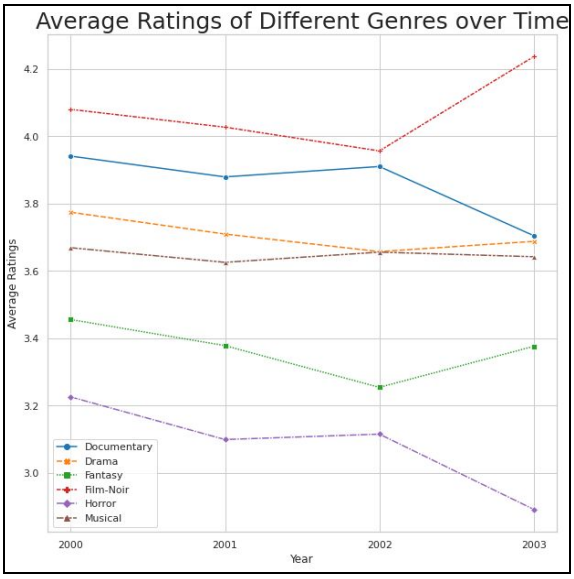


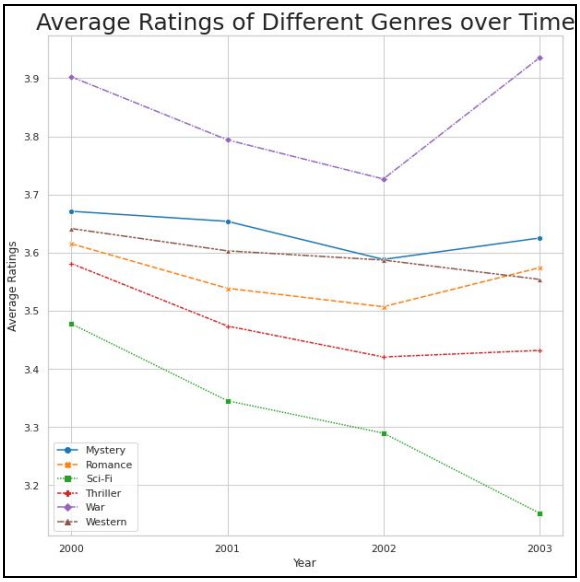
In this visualization, we again use a Heat Map to demonstrate how a person from each occupation rates movies in different genres.

5) AVERAGE GENRE RATINGS OVER YEARS

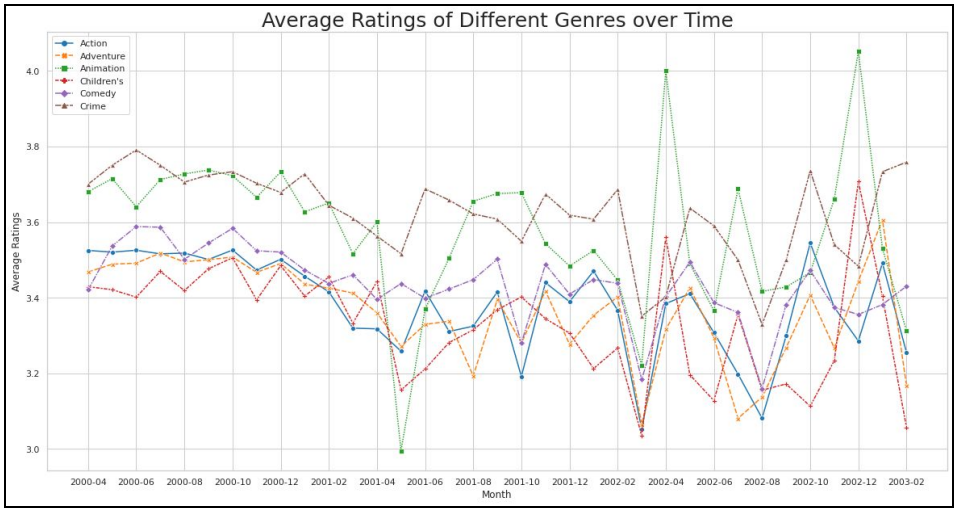
There are 18 genres in total, and we used the Line Graph to demonstrate how the ratings of various genres varied over the time period of 2000-2003. This is split as a group of 6 and thus has 3 graphs, indicating the average rating vs time.

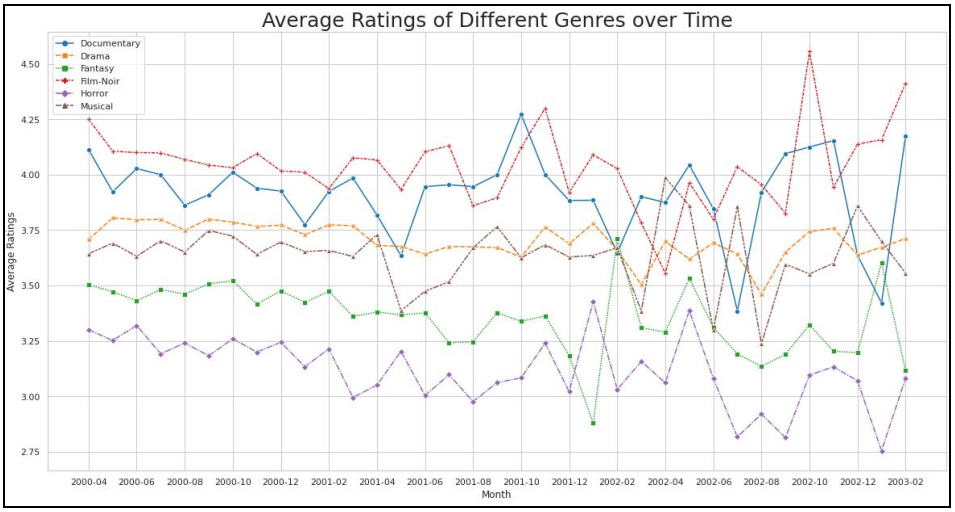


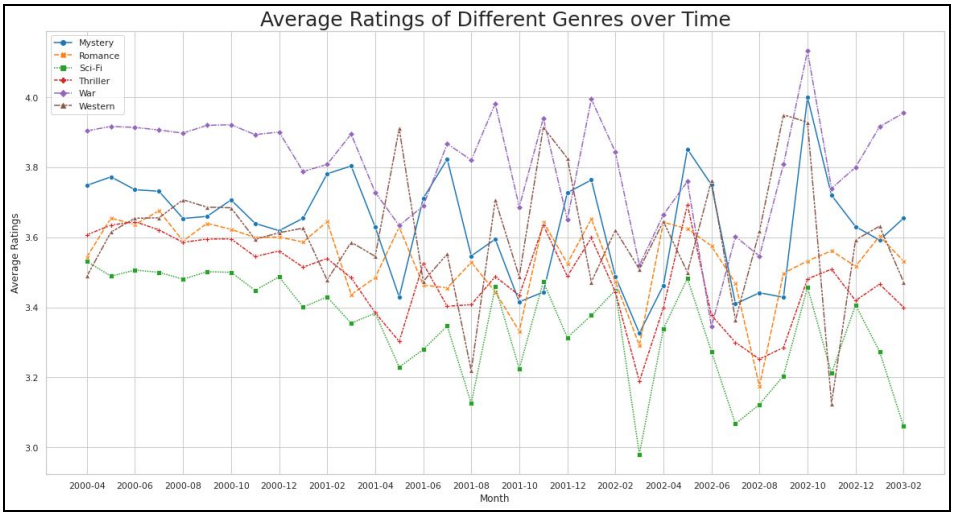




More detailed work was done in the same category, where all genres' average movie ratings were plotted between the same years, but with the intervals of 2 months.






Average Ratings of Different Genres over Time
Mystery
Romance
Sci-Fi
Thriller
War
Western
Average Ratings
4.0
3.8
3.6
3.4
3.2
3.0
2000-04
2000-06
2000-08
2000-10
2000-12
2001-02
2001-04
2001-06
2001-08
2001-10
2001-12
2002-02
2002-04
2002-06
2002-08
2002-10
2002-12
2003-02
Month

DATA PREDICTIONS

We used various prediction algorithms that helped us predict the favourite genre of a user with information

1) USING DECISION TREE

```
Current max depth:  1


Accuracy per fold:  [0.640728476821192, 0.6556291390728477,
0.7102649006622517,  0.6837748344370861,  0.6721854304635762,
0.6903973509933775,  0.6390728476821192,  0.7086092715231788,
0.7152317880794702, 0.6837748344370861]


Average accuracy:  0.6799668874172186

========================================



Current max depth:  2


Accuracy per fold:  [0.8774834437086093, 0.8576158940397351,
0.8857615894039735,  0.8890728476821192,  0.8874172185430463,
0.9188741721854304,  0.8956953642384106,  0.8923841059602649,
0.9105960264900662, 0.8990066225165563]
```

```
Average accuracy:  0.8913907284768212

========================================



Current max depth:  3


Accuracy per fold:  [0.9254966887417219, 0.9105960264900662,
0.9139072847682119,  0.9139072847682119,  0.9321192052980133,
0.9519867549668874,  0.9205298013245033,  0.9387417218543046,
0.9403973509933775, 0.9288079470198676]


Average accuracy:  0.9276490066225167

========================================



Current max depth:  4


Accuracy per fold:  [0.9701986754966887, 0.9784768211920529,
0.9751655629139073,  0.9801324503311258,  0.9834437086092715,
0.9933774834437086,  0.9801324503311258,  0.9884105960264901,
0.9735099337748344, 0.9867549668874173]
```

```
Average accuracy:  0.9809602649006622

========================================



Current max depth:  5

Accuracy per fold:  [0.9884105960264901, 0.9900662251655629,
0.9884105960264901, 0.9933774834437086, 0.9884105960264901,
0.9950331125827815, 0.9917218543046358, 0.9983443708609272,
0.9867549668874173, 0.9917218543046358]

Average accuracy:  0.991225165562914

========================================



Current max depth:  6

Accuracy fold:  [1.0, 0.9950331125827815, 1.0,
0.9983443708609272, 0.9966887417218543, 1.0, 0.9966887417218543,
0.9983443708609272, 0.9966887417218543, 0.9983443708609272]
```

```
Average accuracy:  0.9980132450331126

========================================

Max Depth  Average Accuracy
         1          0.679967
         2          0.891391
         3          0.927649
         4          0.980960
         5          0.991225
         6          0.998013
```

Thus, using a Decision Tree, we observed 99.8013% Accuracy, on a tree 6 levels deep.

2) USING LOGISTIC REGRESSION

```
Accuracy of the model on train data: 0.8654761904761905

Accuracy of the model on eval data: 0.8724206349206349
```

Using Logistic Regression and training our model for 10,000 iterations, we didn't receive a satisfactory accuracy, giving us only 86.54% on training data, and 87.24% of evaluation data.

3) USING SUPPORT VECTOR MACHINE

```
Accuracy of the model on train data: 0.9992063492063492

Accuracy of the model on eval data: 0.8807539682539682
```

Using Support Vector Machine (or SVC - Support Vector Classifier - as it is referred to in the package used) we get 99.92 % accuracy for train data, but an unsatisfactory 88.07% on the evaluation data.

Results & Conclusion

Thus, our research objective and goal is accomplished, where we have carefully and meticulously cleaned and preprocessed the dataset provided to us. Using this data, we go on to generate various tables which were used for a variety of prediction models and visualizations.

Through our exploration, we have found out that the Decision Tree [DT] algorithm yields a better accuracy (99.80%) than the rest of the two algorithms viz. Support Vector Machine [SVM] (88.07%) & Logistic Regression [LR] (87.24%). This helps us effectively predict the favourite genre for a particular user demographic which is dictated by the factors like - Age Group, Sex, Location (State), Occupation.

In our visualization, we were able to visualize trends clearly, using many factors. These expressive visualizations were chosen as per applicability and give us all the information necessary. We visualized occupation trends, age group trends, gender trends and location trends against genres. After those, we also saw the average rating over the time-series from 2000-2003 to see how genres were preferred throughout the user base, through time.

This collection of visualizations, predictions and trends has given us a multi-modal and versatile knowledge base that helps us one realise the might of data as small as 100K data points. This fragment of data and the obtained predictions, still stay true on the surface because the general sentiment and the preferences don’t change drastically over time, thus allowing us to comprehend how tendencies change only infinitesimally, thus helping us plan or create things better.

Future Scope

Using these data cleaning measures, visualizations, and prediction models, it's clear how once can derive a multitude of relationships, patterns and trends through just a small dataset. In today's world, there are voluminous data points that are generated by even the most primitive & simple incidences. This data can be cleaned, aggregated, correlated, and worked on in several ways, using a variety of algorithms that help us derive knowledge in a much easier way, and accurately. This knowledge can be used by every user/actor/person that is involved in the system to thus improve their knowledge and use the data extracted to improve the productivity in each phase of the industry or system. This project can serve as a great start to the students or concerned users who wish to see and use this set of collated processes for their projects, research or application in other domains.