Business Intelligence and Data Mining Report

1. Introduction

This project was created in order to satisfy the necessity of companies on the cinematographic sector. Especially those that distribute movies either in digital or physical format. In that sense, our project pretends to analyze the data collected from movies that have been rated in a digital environment. This analysis contemplates not only movie characteristics and evaluations, but information about the users who rated those movies as demographical data.

2. Business Question

In order to establish what the goal of the project is when analyzing the data, we have established some business questions that will help to lead the project into the right direction. For that reason, we have developed the following business questions:

- Can we recommend a movie based on a movie and user characteristics?
- How accurate will be a possible movie recommendation?
- Is it possible to segment the users so we can better recommend movies and conduct the business efforts?
- How accurate will be the segmentation based in users and movies characteristics?
- Is it possible to increment the accuracy of the possible movie recommendations?
- What would be necessary to do next in order to better improve the business performance?

These questions are aimed to increase the performance of the current business processes and to make the industries involved in this sector more profitable. In that context, we expect to analyze the data having these questions in mind. Therefore, the answer to these questions will be reflected on the development of the models exposed in this report, and in the insights revealed by the results of these models.

Nevertheless, in order to solve this business questions, we have gone through a process that range from exploring and analyze the data to analyze the model results and its implications on the business. Thus, the first steps required not only to analyzed, but to clean and process the data gathered for this project.

In the following sections we will describe the data set used for the mentioned proposes, as well as the variables and tables that were contain in such data. Additionally, we will show an exploratory analysis that can make us better understand the gathered data. The following steps involved the construction of the models and the analysis of the results.

3. Description of the Data

To achieve this endeavor, we have selected a dataset found in the webpage “grouplens.org”, which is called MovieLens. GroupLens is an organization that collected information about movies and users during a long period of time (GroupLens, 2009). We can find in this webpage different data sets, having most of them the same variables.

The GroupLens Research Project is a group from the university of Minnesota. The group that developed this dataset comes from the Department of Computer Science and Engineering, and they are involved in research projects related to information filtering, collaborative filtering, and recommender systems. This project was led by professors John Riedl and Joseph Konstan, who stated exploring automated collaborative filtering in 1992. Moreover, they are very well known for their research based on the trial of an automated collaborative filtering system for Usenet news (Konstan, 2015).

The dataset we have used from this page is the “MovieLens 1M Dataset”, which is a data set released in 2009 and has around 1 million ratings and tags that were written by approximately 6,000 users in over 4,000 movies. Users were selected at random for inclusion. All users selected had rated at least 20 movies. The data are contained in four files, movies.dat, users.dat, ratings.dat and tags.dat. Also included are scripts for generating subsets of the data to support five-fold cross-validation of rating predictions. The tables contained in this data set are described as follow:

3.1. Users

User information is in the file "users.dat" and is divide by UserId, Gender, Age, Occupation, and Zip-Code. We can see the specifications of each variable as follows:

- Gender: is denoted by a "M" for male and "F" for female
- Age is chosen from the following ranges:
  - 1: "Under 18"
  - 18: "18-24"
  - 25: "25-34"
  - 35: "35-44"
  - 45: "45-49"
  - 50: "50-55"
  - 56: "56+"
- Occupation is chosen from the following choices:
  - 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: This is digit that ranges from 3 digits to 5 digits and that sometimes contains multiple zip codes.

3.2. Movies

Movie information is in the file "movies.dat" and is divide by MovieID, Title, and Genres. We can see the specifications of each variable as follows:

- MovieID is a number starting from 1 and identifies the movie.
- Title is the name of the movie and they are identical to titles provided by the IMDB. These titles contain the year of release in parenthesis.
- Genres are separated by a pipe and are shown as follow:
  - Action
  - Adventure
  - Animation
  - Children's
  - Comedy
  - Crime
  - Documentary
  - Drama
  - Fantasy
  - Film-Noir
  - Horror

- Musical
- Mystery
- Romance
- Sci-Fi
- Thriller
- War
- Western

3.3. Ratings

All ratings are contained in the file "ratings.dat" and are divided by UserID, MovieID, Ratings, Timestamp. We can see the specifications of each variable as follows:

- UserIDs range between 1 and 6040
- MovieIDs range between 1 and 3952
- Ratings are made on a 5-star scale (whole-star ratings only)
- Timestamp is represented in seconds since the epoch as returned by time(2)
- Each user has at least 20 ratings

Additionally, after obtaining the data set, we had to clean and process the data to a format that can be used by SAS to produce models. For that reason, we execute the following tasks on the substracted dataset.

- Matched the userID and movieID between the data sets. So we can create a joined table to be upload to the SAS software.
- Convert the file to a CSV format.
- Split title and year into separated columns.
- Convert and split Genres into binary variables, 1 column for each genre.
- Convert ratings into a binary variable called “like” that is 1 if rating >= 3 and 0 otherwise.
- Calculate and create an additional column from the average rating and popularity (number of ratings) for each movie.
- Calculate the average rating by genre for each user and create an additional column.
- Convert the zip code into an state level.
- Additionally, in order to work with SAS we reduce the sample to 100,000 records. This reduction was also due to the processing time took for converting the zip codes into states.

In order to achieve the above results, we had to used software outside of the SAS scope. For this preprocessing task we used Python and the libraries contained on it. For transforming the Zip Code, we used a library called “uszipcode 0.1.3” from the Python website and created by a third-party organization (Hu, 2016).

4. Exploratory plots and tables

Following is the new data set:

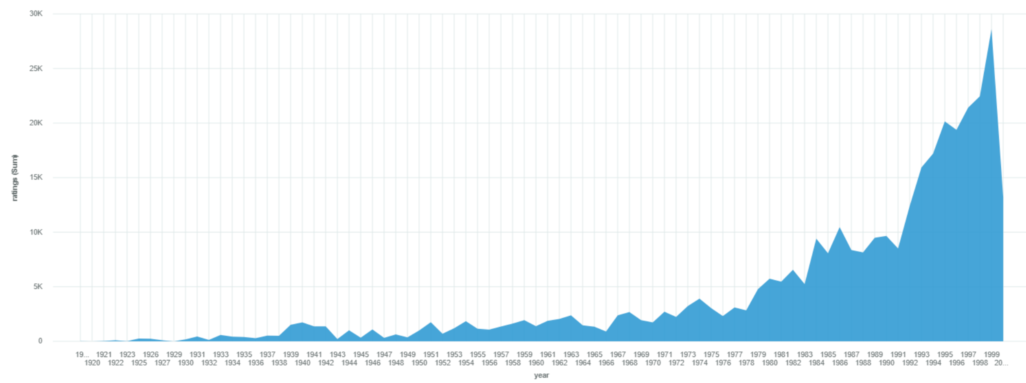
Obs #	Variable ...	Label	Type	Percent ...	Minimum	Maximum	Mean	Number o...	Mode Per...	Mode
1	gender		CLASS	0	.	.	.	2	74.915	M
2	state		CLASS	0	.	.	.	52	18.325	CA
3	title		CLASS	0	.	.	.	128+	2.466091	STAR WAR...
4	Action		VAR	0	0	1	0.2542	.	.	
5	Action_avg		VAR	6.67	1	5	3.454379	.	.	
6	Adventure		VAR	0	0	1	0.1326	.	.	
7	Adventure_...		VAR	13.29	1	5	3.435288	.	.	
8	Animation		VAR	0	0	1	0.04445	.	.	
9	Animation_...		VAR	40.135	1	5	3.669043	.	.	
10	Children_avg		VAR	29.12	1	5	3.411326	.	.	
11	Childrens		VAR	0	0	1	0.0714	.	.	
12	Comedy		VAR	0	0	1	0.35655	.	.	
13	Comedy_avg		VAR	3.41	1	5	3.535198	.	.	
14	Crime		VAR	0	0	1	0.0796	.	.	
15	Crime_avg		VAR	21.04	1	5	3.654859	.	.	
16	Documentary		VAR	0	0	1	0.0071	.	.	
17	Documenta...		VAR	79.865	1	5	3.829753	.	.	
18	Drama		VAR	0	0	1	0.34975	.	.	
19	Drama_avg		VAR	2.775	1	5	3.75678	.	.	
20	Fantasy		VAR	0	0	1	0.0346	.	.	
21	Fantasy_avg		VAR	41.15	1	5	3.370063	.	.	
22	FilmNoir		VAR	0	0	1	0.01865	.	.	
23	FilmNoir_avg		VAR	61.59	1	5	4.008539	.	.	
24	Horror		VAR	0	0	1	0.08135	.	.	
25	Horror_avg		VAR	27.115	1	5	3.211824	.	.	
26	Musical		VAR	0	0	1	0.04235	.	.	
27	Musical_avg		VAR	39.765	1	5	3.626759	.	.	
28	Mystery		VAR	0	0	1	0.04175	.	.	
29	Mystery_avg		VAR	36.55	1	5	3.637303	.	.	
30	Romance		VAR	0	0	1	0.15305	.	.	
31	Romance_...		VAR	10.425	1	5	3.590855	.	.	
32	SciFi		VAR	0	0	1	0.1567	.	.	
33	SciFi_avg		VAR	13.025	1	5	3.42015	.	.	
34	Thriller		VAR	0	0	1	0.19155	.	.	
35	Thriller_avg		VAR	8.07	1	5	3.54788	.	.	
36	War		VAR	0	0	1	0.0674	.	.	
37	War_avg		VAR	22.11	1	5	3.839411	.	.	
38	Western		VAR	0	0	1	0.0217	.	.	
39	Western_avg		VAR	55.215	1	5	3.573249	.	.	
40	age		VAR	0	1	56	29.7506	.	.	
41	like		VAR	0	0	1	0.57865	.	.	
42	movieID		VAR	0	1	3952	1876.247	.	.	
43	movie_avg...		VAR	0	1	5	3.584743	.	.	
44	movie_pop...		VAR	0	1	291	78.55585	.	.	
45	occupation		VAR	0	0	20	8.078	.	.	
46	ratings		VAR	0	1	5	3.59305	.	.	
47	userID		VAR	0	2	6040	3041.84	.	.	
48	year		VAR	0	1920	2000	1986.624	.	.	

We can find the following insights:

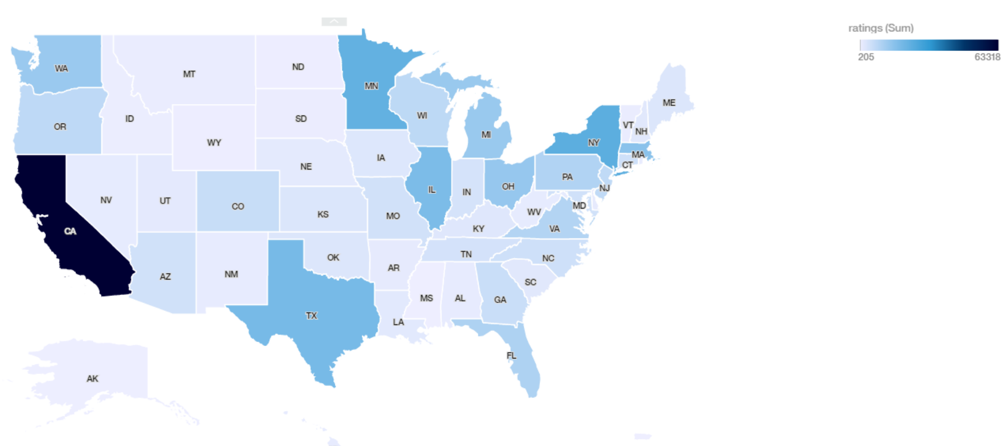
- From the Percent Missing column, we can see that some values of genres are missing. Thus it's necessary for us to impute this variable in preparation for regression.
- From Mode column, we can see that most of the users are male, most of them are from California, and the Star War is the most popular movies.

For the variable correlations we have found the following insights:

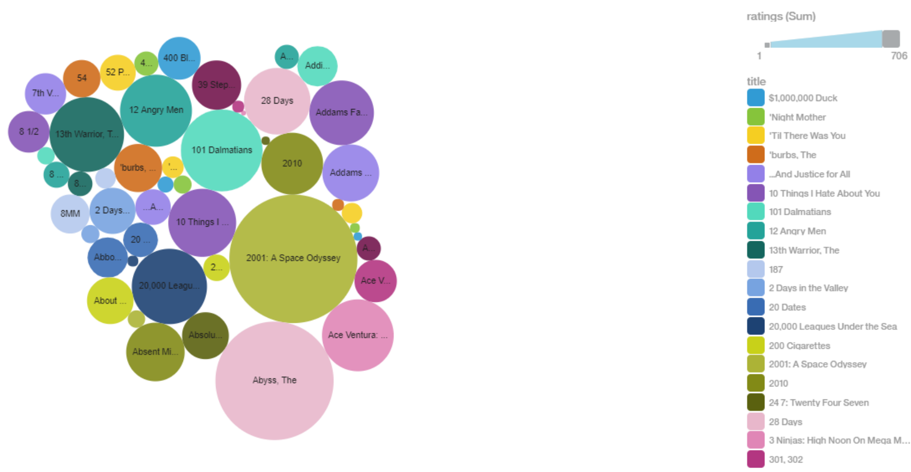
- Ratings over years can show us how the ratings have been increasing year by year. It shows most of the movies rated were released between 1992 and 2000. This can be explained by the growth of the digital movie industry. In companies as Netflix or Hulu, ratings are provided every day with more and more people are using these webpages every year instead of using physical devices. This provides an opportunity to collect even more data from the users. A plot can be visualized bellow.



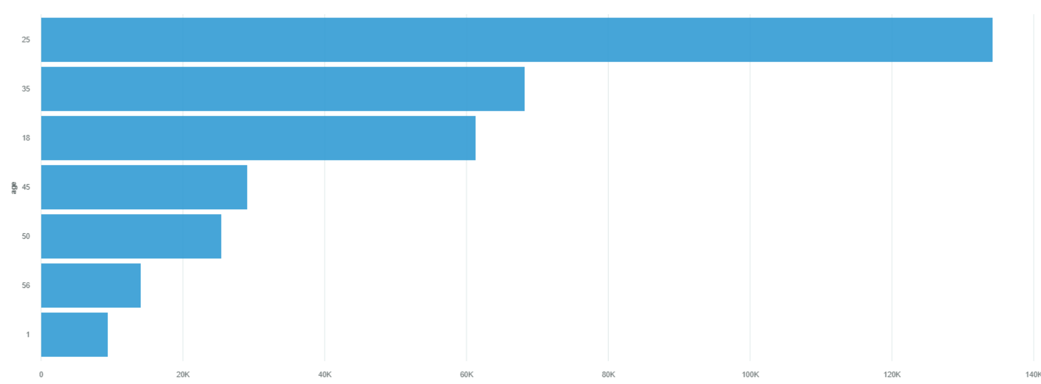
- Ratings over state can give us insights about how many people are rating movies in each state. This can help us know how many people is watching movies, or, at least, how many peoples is rating movies by state. You can visualize the graph below.



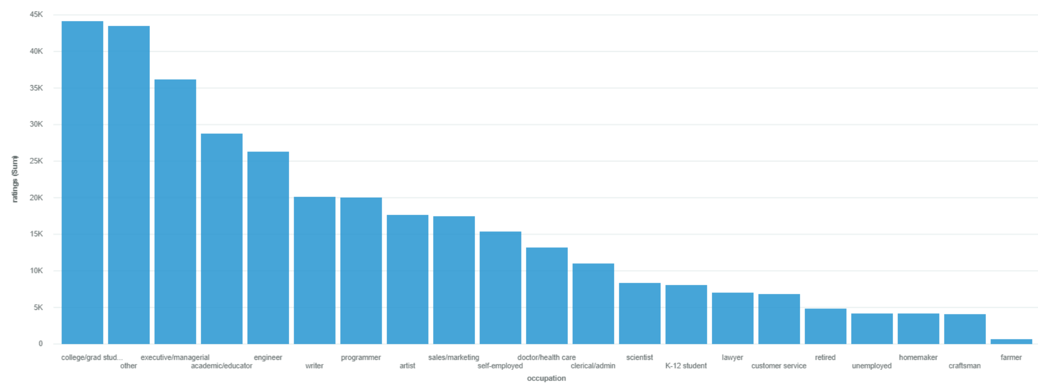
- Rating over Title can give us insights about what is the most watched movie, or at least which movies is being rated more. This can help the company to ask the question: why are some movies rated more than others? This can also help on recommending popular movies. You can visualize the graph as follows:



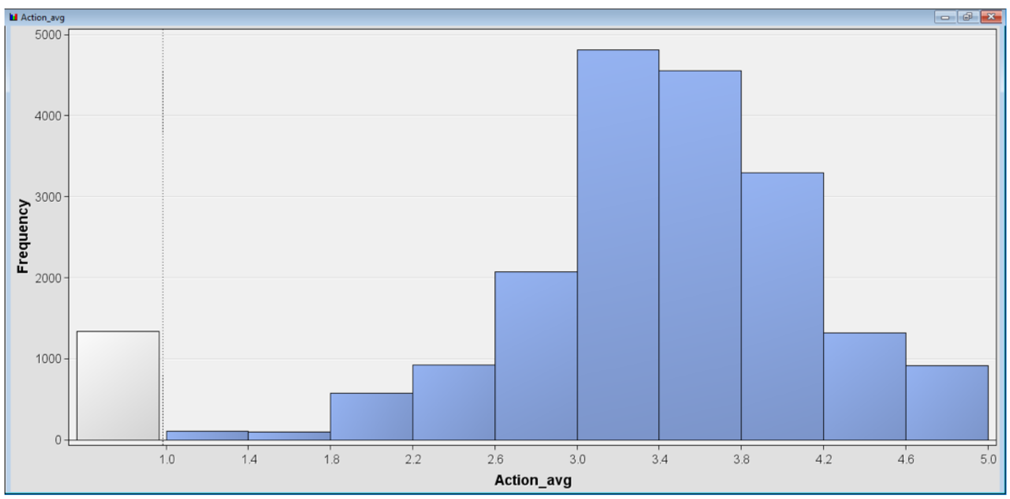
- Ratings over Age can tell us how age is related to the rating gave to a particular movie. We can see that the survey was mainly taken by users aged 25 to 34. This can help us better understand the consumers behavior respect of their age. Therefore, we can ask questions as: What makes people of certain age rate movies more than other ages respect of their proportion of movies being watched? What is the age of the people that is willing to watch more movies or rate more movies? Additionally, this can help us to recommend popular movies to certain ages. You can visualize the graph below.



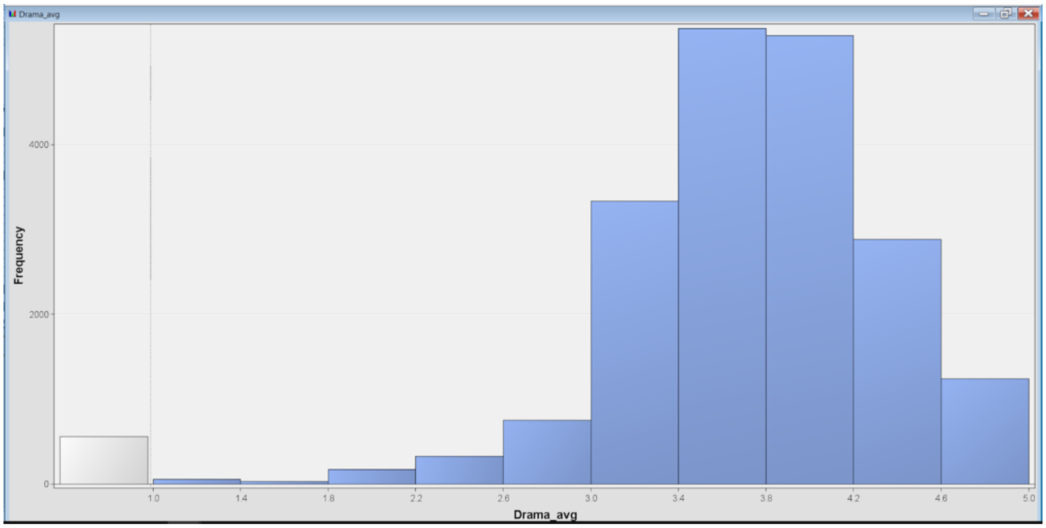
- Rating over occupation can give us some insights of what kind of people, respect to their occupation, are watching or rating more movies. The questions that this kind of correlation generates is similar to what was mentioned in the points made before. It shows that most of the users were college/grad students. You can visualize the graph as follows:



- Comparing average rating of different genres can give us some insights of what genres receive high rating from users. We choose two average ratings here and compare them. The first graph is the distribution of average rating of action movies. It shows that most of the users rated action movies between 3 and 3.8.



- Below is the distribution for average rating of drama movies. It shows that most of the users rated action movies between 3.4 and 4.2. The average rating is higher than that or Drama Movies.



5. **BIDM METHODOLOGY**

5.1. **Unsupervised learning**

1. **Association Rules**

We used association rules to discover associations between movies users like.

We first filtered out movies they don't like. For each user, if the rating of a movie is less than 3 stars, it means he doesn't like the movie, and it should be removed from the data set. We also deleted genre columns, hoping SAS would run faster. After that, we changed userID to ID role, changed title to target role, and left others default.

We first ran two-item association rules. As shown below, lift values are all greater than 1, it means these rules are significant, and each two occurrences are dependent on one another. To interpret confidence, support, and lift, let's take the first rule as an example. A user who likes Patton is 7.74 as likely to like Amadeus than a user chosen at random. The probability that a user likes both Patton and Amadeus is 0.27. The probability that a user likes Amadeus given that he likes Patton is 21.43. You can visualize the graph as follow:

Association Report

Relations	Expected Confidence (%)	Confidence (%)	Support (%)	Lift	Transaction Count	Rule	Left Hand of Rule
2	2.77	21.43	0.27	7.74	15.00	Patton ==> Amadeus	Patton
2	2.25	15.74	0.30	6.98	17.00	Glory ==> Full Metal Jacket	Glory
2	1.92	13.39	0.30	6.98	17.00	Full Metal Jacket ==> Glory	Full Metal Jacket
2	2.15	13.27	0.23	6.18	13.00	Splash ==> Lion King, The	Splash
2	1.74	10.74	0.23	6.18	13.00	Lion King, The ==> Splash	Lion King, The
2	2.54	15.29	0.23	6.02	13.00	Trading Places ==> Ferris Bueller's Day Off	Trading Places
2	2.06	10.65	0.32	5.17	18.00	Die Hard ==> Batman	Die Hard
2	3.00	15.52	0.32	5.17	18.00	Batman ==> Die Hard	Batman
2	2.75	14.13	0.23	5.14	13.00	Witness ==> When Harry Met Sally...	Witness
2	3.37	17.28	0.25	5.12	14.00	Brazil ==> Terminator, The	Brazil
2	2.89	14.77	0.23	5.11	13.00	Last of the Mohicans, The ==> Hunt for Red October, The	Last of the Mohicans, The
2	2.80	14.29	0.25	5.09	14.00	Mad Max 2 (a.k.a. The Road Warrior) ==> Blade Runner	Mad Max 2 (a.k.a. The Roa
2	2.08	10.24	0.23	4.93	13.00	Full Metal Jacket ==> Blues Brothers, The	Full Metal Jacket
2	2.25	11.11	0.23	4.93	13.00	Blues Brothers, The ==> Full Metal Jacket	Blues Brothers, The
2	2.40	11.76	0.25	4.91	14.00	Shining, The ==> Dr. Strangelove or: How I Learned to Stop Worrying and Love the Bomb	Shining, The
2	2.11	10.37	0.25	4.91	14.00	Dr. Strangelove or: How I Learned to Stop Worrying and Love the Bomb ==> Shining, The	Dr. Strangelove or: How I
2	3.43	16.67	0.25	4.86	14.00	Armageddon ==> Fugitive, The	Armageddon
2	3.00	14.29	0.23	4.76	13.00	Crocodile Dundee ==> Die Hard	Crocodile Dundee
2	3.21	15.15	0.27	4.72	15.00	Top Gun ==> E.T. the Extra-Terrestrial	Top Gun
2	3.69	17.33	0.23	4.69	13.00	Dead Man Walking ==> Shawshank Redemption, The	Dead Man Walking
2	2.80	13.00	0.23	4.63	13.00	Star Trek: First Contact ==> Blade Runner	Star Trek: First Contact
2	3.14	14.29	0.28	4.55	16.00	Graduate, The ==> Casablanca	Graduate, The
2	2.43	11.02	0.25	4.53	14.00	Clueless ==> Big	Clueless
2	2.25	10.22	0.25	4.53	14.00	Big ==> Clueless	Big
2	4.79	21.67	0.23	4.52	13.00	Excalibur ==> Star Wars: Episode VI - Return of the Jedi	Excalibur

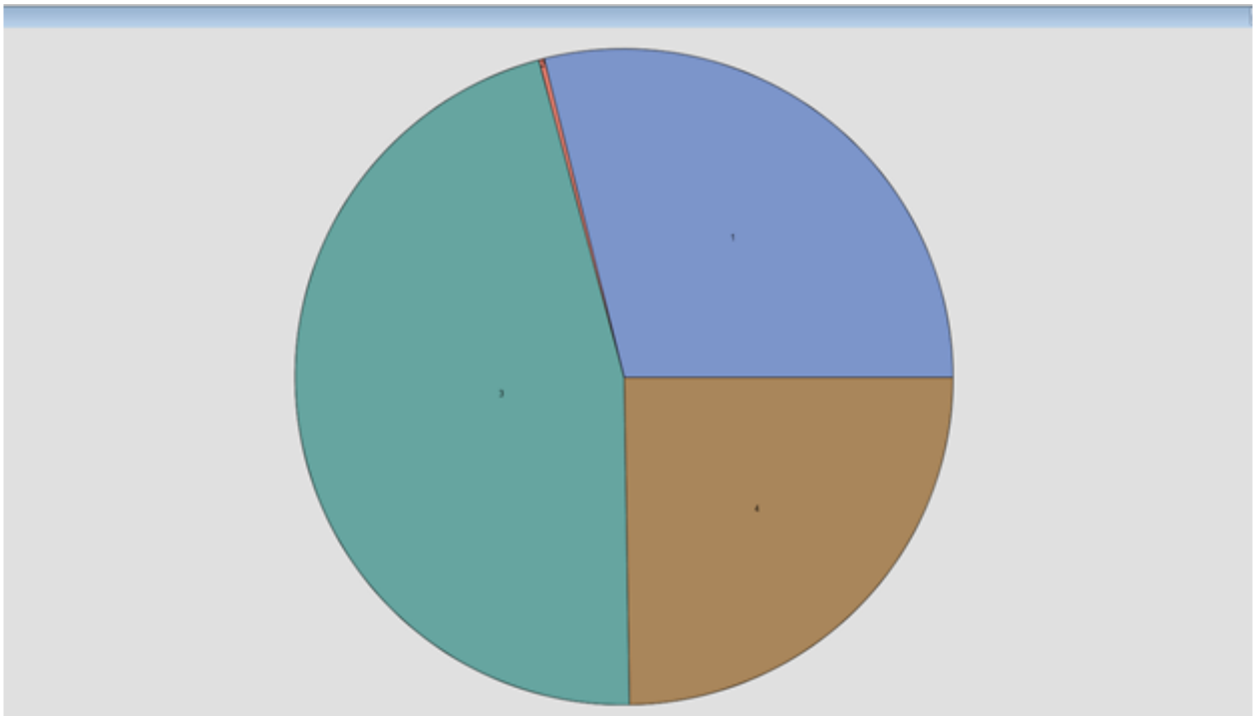
We also ran three-item association rules, but ended up getting the same outcome.

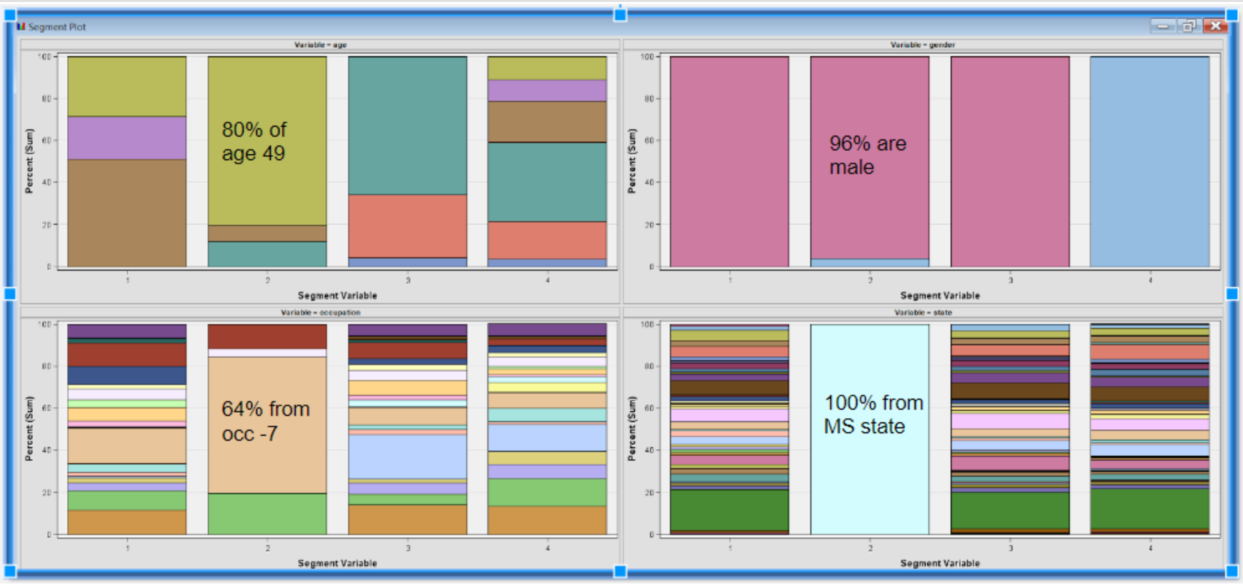
2. Clustering

We have decided to cluster users in order to understand their preferences and makes suggestions based on those preferences. Users were clustered based on their demographics like age, gender, occupation and state variables. The model segregated the users into 4 clusters. The users in a cluster are similar to one another and would have similar movie preferences. We can than find out the most watched movie in the cluster and suggest the movie to a user who has not yet watched that movie.

Mean Statistics

Clustering Criterion	Maximum Relative Change in Cluster Seeds	Improvement in Clustering Criterion	Segment Id	Frequency of Cluster	Root-Mean -Square Standard Deviation	Maximum Distance from Cluster Seed	Nearest Cluster	Distance to Nearest Cluster	age	gender=F	gender=M
0.179662	0	0	1	27630	0.179146	7.095445	3	1.712167	41.92646	4.58E-14	1
0.179662	0	0	2	51	0.122357	2.655644	1	6.024779	46.47059	0.039216	0.960784
0.179662	0	0	3	43981	0.165854	5.169271	1	1.712167	21.90032	1.87E-13	1
0.179662	0	0	4	23653	0.203489	5.503461	3	2.418935	29.92601	1	-1.3E-13





From the above clusters we can interpret that 80% of the people in the cluster 2 are of age 49, 96% of them are male,64% of them are from occupation 7 and all of them are from Mississippi. Similarly all other clusters can be interpreted from the results on SAS.

5.2. Supervised learning

We wanted to predict if a user likes a movie or not based on the rating he might give a movie. Anything above 3.5 is a like and below 3.5 is not like. We created additional calculated columns to create a likes column which is binary. 1 for like and 0 for not like and then we used supervised learning models to predict the probability of a person to like a movie or not like a movie based various characteristics of the movie like genre, year of release etc. and person like gender, age, state etc.

Based on the above criteria we have made the like column as a target variable and others as inputs like the table shown below :

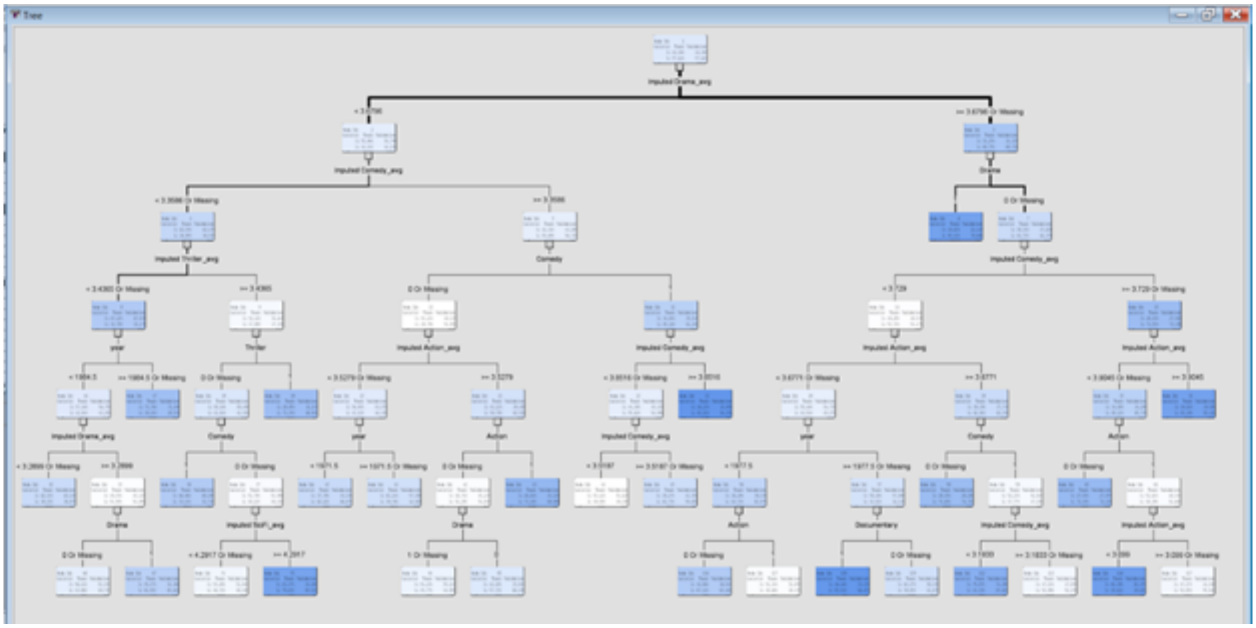
(none) not Equal to

Columns: Label Mining Basic

Name	Role	Level	Report	Order	Drop	Lower Limit	Upper Limit
Action	Input	Binary	No		No	.	.
Action_avg	Input	Interval	No		No	.	.
Adventure	Input	Binary	No		No	.	.
Adventure_avg	Input	Interval	No		No	.	.
age	Input	Nominal	No		No	.	.
Animation	Input	Binary	No		No	.	.
Animation_avg	Input	Interval	No		No	.	.
Childrens	Input	Binary	No		No	.	.
Children_avg	Input	Interval	No		No	.	.
Comedy	Input	Binary	No		No	.	.
Comedy_avg	Input	Interval	No		No	.	.
Crime	Input	Binary	No		No	.	.
Crime_avg	Input	Interval	No		No	.	.
Documentary	Input	Binary	No		No	.	.
Documentary_	Input	Interval	No		No	.	.
Drama	Input	Binary	No		No	.	.
Drama_avg	Input	Interval	No		No	.	.
Fantasy	Input	Binary	No		No	.	.
Fantasy_avg	Input	Interval	No		No	.	.
FilmNoir	Input	Binary	No		No	.	.
FilmNoir_avg	Input	Interval	No		No	.	.
gender	Input	Binary	No		No	.	.
Horror	Input	Binary	No		No	.	.
Horror_avg	Input	Interval	No		No	.	.
like	Target	Binary	No		No	.	.
movieID	ID	Nominal	No		No	.	.
movie_avg_rat	Rejected	Interval	No		No	.	.
movie_popular	Rejected	Interval	No		No	.	.
Musical	Input	Binary	No		No	.	.
Musical_avg	Input	Interval	No		No	.	.
Mystery	Input	Binary	No		No	.	.
Mystery_avg	Input	Interval	No		No	.	.
occupation	Input	Nominal	No		No	.	.
ratings	Rejected	Interval	No		No	.	.
Romance	Input	Binary	No		No	.	.
Romance_avg	Input	Interval	No		No	.	.
SciFi	Input	Binary	No		No	.	.
SciFi_avg	Input	Interval	No		No	.	.
state	Input	Nominal	No		No	.	.
Thriller	Input	Binary	No		No	.	.
Thriller_avg	Input	Interval	No		No	.	.
title	Rejected	Nominal	No		No	.	.
userID	ID	Nominal	No		No	.	.
War	Input	Binary	No		No	.	.
War_avg	Input	Interval	No		No	.	.
Western	Input	Binary	No		No	.	.
Western_avg	Input	Interval	No		No	.	.
year	Input	Interval	No		No	.	.

3. Decision Tree:

We has chosen the decision tree model to predict if a user will like or not like the movie. We have calculated the action_avg , drama_avg etc variables which are the average rating given by users to movies in each genre. We have imputed the avg variables with ‘0’ wherever the values are missing. This means the user has not yet rated a movie from that genre. We have partitioned the data set into 70% training and 30% validation sets.



As we can see from the above decision tree the first significant split is made on imputed drama_avg . Drama has been a significant genre in the given data set and depending on the users rating for drama movies we can segregate the movies. If a user doesn't like drama then the next significant node is comedy , if a user likes drama then the next significant node is if the movie is a drama movie or not. If the movie is a drama genre then we can safely recommend that movie. If it's not a drama movie then we can see the users avg rating for comedy and if the movie is a comedy movie then we can suggest it, if not we can check the users action_avg and suggest a movie based on action genre.

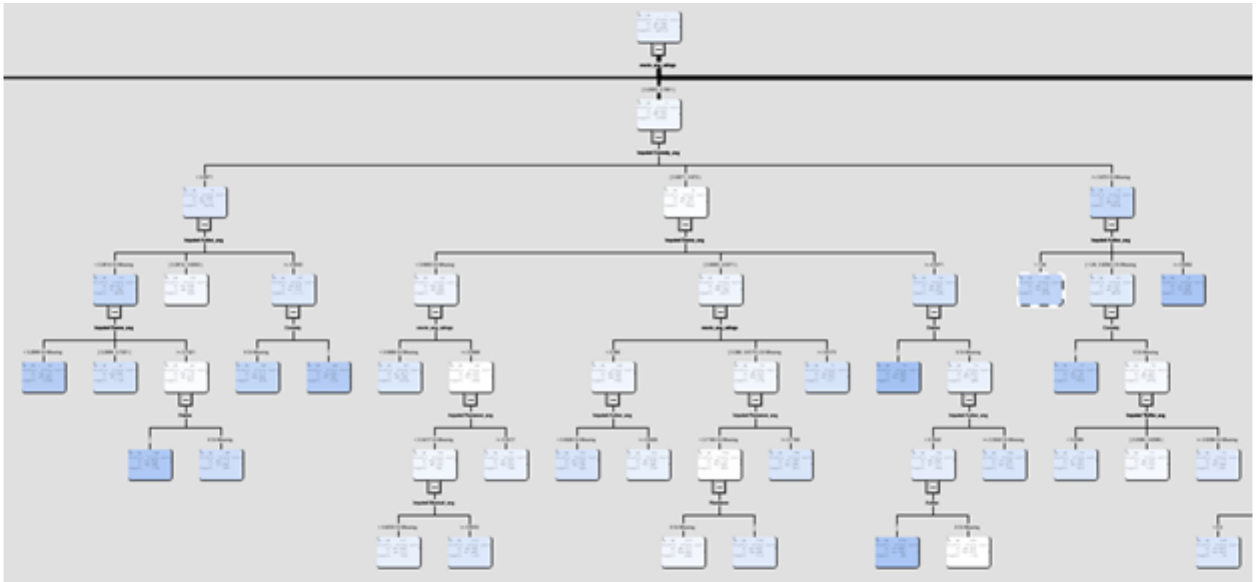
We have achieved a misclassification rate of 0.26 that means the model was ~75% accurate in recommending the movie the user likes.

Target	Target Label	Fit Statistics	Statistics Label	Train	Validation
like		_NOBS_	Sum of Frequencies	66719	28596
like		_MISC_	Misclassification Rate	0.265247	0.266261
like		_MAX_	Maximum Absolute Error	0.863733	0.863733
like		_SSE_	Sum of Squared Errors	24732.01	10653.22
like		_ASE_	Average Squared Error	0.185345	0.186271
like		_RASE_	Root Average Squared Error	0.430517	0.431591
like		_DIV_	Divisor for ASE	133438	57192
like		_DFT_	Total Degrees of Freedom	66719	.

Variable Importance

Variable Name	Label	Number of Splitting Rules	Importance	Validation Importance	Ratio of Validation to Training Importance
movie_avg_ratings		5	1.0000	1.0000	1.0000
IMP_Drama_avg	Imputed Drama_avg	4	0.4712	0.4384	0.9305
IMP_Action_avg	Imputed Action_avg	2	0.3707	0.3546	0.9564
IMP_Comedy_avg	Imputed Comedy_avg	4	0.2687	0.2607	0.9702
Action		3	0.1687	0.1695	1.0048
Drama		3	0.1548	0.1641	1.0600
IMP_Thriller_avg	Imputed Thriller_avg	1	0.1108	0.0947	0.8542
Thriller		1	0.0673	0.0680	1.0104

We have also tried to split the tree into 3 nodes:



Target	Target Label	Fit Statistics	Statistics Label	Train	Validation
like		_NOBS_	Sum of Frequencies	66719	28596
like		_MISC_	Misclassification Rate	0.254905	0.262309
like		_MAX_	Maximum Absolute Error	0.959319	0.959319
like		_SSE_	Sum of Squared Errors	23438.56	10237.1
like		_ASE_	Average Squared Error	0.175651	0.178995
like		_RASE_	Root Average Squared Error	0.419108	0.423078
like		_DIV_	Divisor for ASE	133438	57192
like		_DFT_	Total Degrees of Freedom	66719	.

Variable Importance

Variable Name	Label	Number of Splitting Rules	Importance	Validation Importance	Ratio of Validation to Training Importance
movie_avg_ratings		8	1.0000	1.0000	1.0000
IMP_Comedy_avg	Imputed Comedy_avg	9	0.3998	0.3828	0.9576
IMP_Drama_avg	Imputed Drama_avg	8	0.3966	0.3556	0.8967
IMP_Action_avg	Imputed Action_avg	6	0.3732	0.3421	0.9168
Comedy		5	0.1946	0.1461	0.7507
Drama		6	0.1898	0.1912	1.0078
Action		5	0.1134	0.1120	0.9877
IMP_Thriller_avg	Imputed Thriller_avg	3	0.1021	0.0831	0.8142
IMP_Horror_avg	Imputed Horror_avg	2	0.0668	0.0507	0.7582
Horror		2	0.0563	0.0554	0.9832
IMP_Romance_avg	Imputed Romance_avg	2	0.0562	0.0343	0.6096
IMP_Fantasy_avg	Imputed Fantasy_avg	1	0.0482	0.0000	0.0000
IMP_Musical_avg	Imputed Musical_avg	1	0.0380	0.0228	0.6007
Romance		1	0.0357	0.0000	0.0000
Crime		1	0.0316	0.0096	0.3041

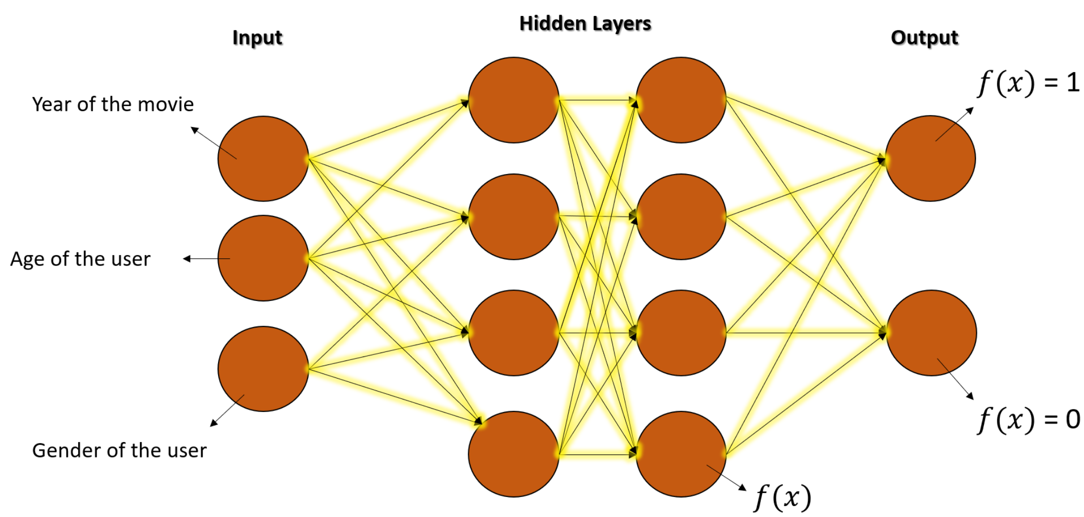
We have found that when we split into 3 nodes the order of significant variables has changed. Initially in the 2 nodes decision tree it was drama, action and comedy but in the 3 node decision tree its comedy, drama and action. The misclassification has also gone down from .2662 to .2623

4. Neural Networks

While Neural Networks were not cover in this class, we decide to work on them due to its high accuracy in this type of classification problems. For this reason, we will briefly explain what a Neural Network is and then show the results found in this model.

A Neural Network is a model based on the real-life neurons that are found in our brains. These neurons as in the brain interconnect to each other in order to perform certain task. In this model we can find an input layer that works as the entrance for the dataset in our case. Then we can find intermediate layer that perform certain functions on the data in order to better classify elements based on their characteristics. For this intermediate layer, each neuron has a weight that brings a value to each of the variables. After the execution of that function a sigmoid function is executed on the output layers to find if the result of each neuron is either a 0 or a 1. In our case our result can be interpreted as a 10 or 01 for either like or dislike. That means one of the two final neurons will activate to classify the input.

We interpret that our model is based on backpropagation, which is a type of Neural Network that calculates the mentioned function with respect of the neuron weights and that relearn based on the outputs of continuous layers. You can find an image of a simple neural network bellow. This image can help us better interpret how a neural network works.



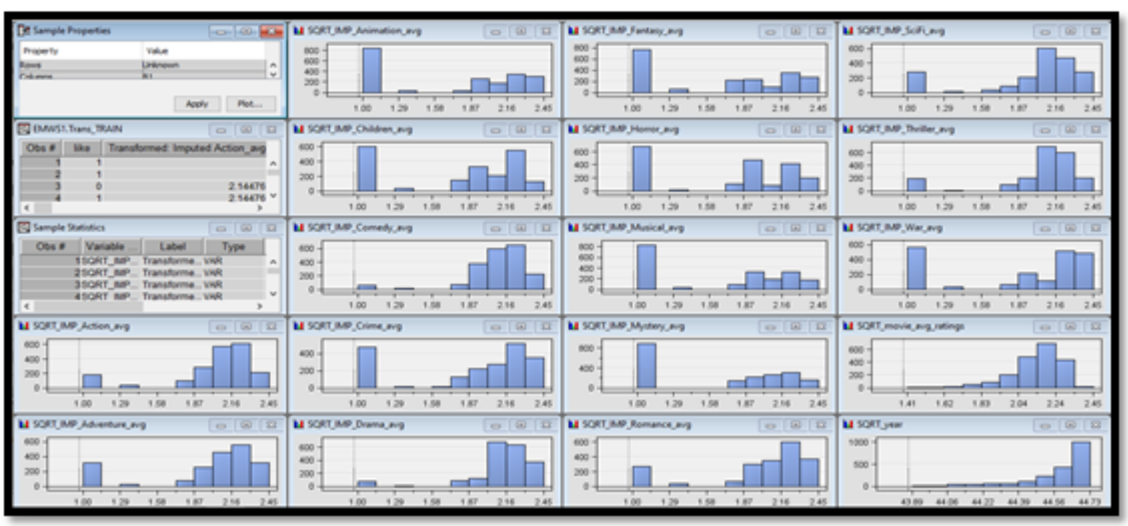
The results when implemented the neural network on our dataset were the best among the other models. The indicator used for measuring the error in this model as well as in the other implemented models was the misclassification error. You can find the results as follow:

Target	Target Label	Fit Statistics	Statistics Label	Train	Validation
like		_DFT_	Total Degrees of Freedom	66719	.
like		_DFE_	Degrees of Freedom for Error	66373	.
like		_DFM_	Model Degrees of Freedom	346	.
like		_NW_	Number of Estimated Weights	346	.
like		_AIC_	Akaike's Information Criterion	69360.75	.
like		_SBC_	Schwarz's Bayesian Criterion	72512.2	.
like		_ASE_	Average Squared Error	0.171648	0.17378
like		_MAX_	Maximum Absolute Error	0.97795	0.991585
like		_DIV_	Divisor for ASE	133438	57192
like		_NOBS_	Sum of Frequencies	66719	28596
like		_RASE_	Root Average Squared Error	0.414304	0.416869
like		_SSE_	Sum of Squared Errors	22904.3	9938.821
like		_SUMW_	Sum of Case Weights Times Freq	133438	57192
like		_FPE_	Final Prediction Error	0.173437	.
like		_MSE_	Mean Squared Error	0.172542	0.17378
like		_RFPE_	Root Final Prediction Error	0.416458	.
like		_RMSE_	Root Mean Squared Error	0.415382	0.416869
like		_AVERR_	Average Error Function	0.514612	0.520947
like		_ERR_	Error Function	68668.75	29793.98
like		_MISC_	Misclassification Rate	0.257633	0.257763
like		_WRONG_	Number of Wrong Classifications	17189	7371

5.2 Logistic Regression

The next supervised model that we developed was logistic regression. When calculating each user's average rating on each genre, missing values were generated if a user had never rated a certain genre. The first step in building this logistic regression model was imputing these missing values with 0, assuming that if a user has never watched a certain genre, they most likely don't like that genre. Next, we transformed the variables year, movie_avg_rating, and user's average rating for each genre with a square root transformation because they were heavily skewed. That resulted in the distributions in the image below for those variables. The transformation did not work as well as we expected. We would have liked more of the variables to have a distribution closer to normal. If we could continue working on the project to try to get better results, we may have attempted different transformations for these variables in an attempt to make the distribution more normal.

Figure 1: Transformed Variables



Next, we partitioned the data with 70% going to the training set, and 30% going to the validation set. Saving 30% for the validation step helps avoid overfitting by testing the model on data that was not used to train the model. We then attempted to predict the "like" variable (a binary variable that is 1 if the user liked the movie, 0 otherwise) using all variables as inputs. We used stepwise model selection to determine which combination of input variables gave the best results. Figure 2 shows the results for logistic regression. The model had an average squared error of 0.190355 and a misclassification rate of 0.290041 on the validation set.

Figure 2: Logistic Regression Results

Target	Target Label	Fit Statistics	Statistics Label	Train	Validation
like		_AIC_	Akaike's Information Criterion	75323.58	.
like		_ASE_	Average Squared Error	0.190972	0.190355
like		_AVERR_	Average Error Function	0.563045	0.562196
like		_DFE_	Degrees of Freedom for Error	66623	.
like		_DFM_	Model Degrees of Freedom	96	.
like		_DFT_	Total Degrees of Freedom	66719	.
like		_DIV_	Divisor for ASE	133438	57192
like		_ERR_	Error Function	75131.58	32153.11
like		_FPE_	Final Prediction Error	0.191522	.
like		_MAX_	Maximum Absolute Error	0.994695	0.997014
like		_MSE_	Mean Square Error	0.191247	0.190355
like		_NOBS_	Sum of Frequencies	66719	28596
like		_NW_	Number of Estimate Weights	96	.
like		_RASE_	Root Average Sum of Squares	0.437003	0.436296
like		_RFPE_	Root Final Prediction Error	0.437632	.
like		_RMSE_	Root Mean Squared Error	0.437318	0.436296
like		_SBC_	Schwarz's Bayesian Criterion	76197.97	.
like		_SSE_	Sum of Squared Errors	25482.89	10886.76
like		_SUMW_	Sum of Case Weights Times Freq	133438	57192
like		_MISC_	Misclassification Rate	0.292121	0.290041

The model that was chosen included the following variables: Adventure, Documentary, FilmNoir, Horror, Action_avg, Children_avg, Comedy_avg, Crime_avg, Drama_avg, Musical_avg, Mystery_avg, SciFi_avg, Thriller_avg, War_avg, movie_avg_ratings, SciFi, War, age, gender, occupation and state. From Figure 3, it can be seen that the following variables are most important in the model: Action_avg, Comedy_avg, Crime_avg, Drama_avg, Mystery_avg, Thriller_avg, War_avg, movie_avg_ratings, age, occupation and state.

Figure 3: Logistic Regression Variable Importance

Type 3 Analysis of Effects

Effect	DF	Wald Chi-Square	Pr > ChiSq
Adventure	1	7.9425	0.0048
Documentary	1	5.1595	0.0231
FilmNoir	1	13.0173	0.0003
Horror	1	9.1387	0.0025
SQRT_IMP_Action_avg	1	167.2818	<.0001
SQRT_IMP_Children_avg	1	13.1345	0.0003
SQRT_IMP_Comedy_avg	1	550.4792	<.0001
SQRT_IMP_Crime_avg	1	24.4962	<.0001
SQRT_IMP_Drama_avg	1	400.7670	<.0001
SQRT_IMP_Musical_avg	1	13.9406	0.0002
SQRT_IMP_Mystery_avg	1	58.7357	<.0001
SQRT_IMP_SciFi_avg	1	14.6825	0.0001
SQRT_IMP_Thriller_avg	1	145.9993	<.0001
SQRT_IMP_War_avg	1	16.1247	<.0001
SQRT_movie_avg_ratings	1	8636.7561	<.0001
SciFi	1	4.4525	0.0348
War	1	5.2598	0.0218
age	6	29.7085	<.0001
gender	1	9.7987	0.0017
occupation	20	80.2392	<.0001
state	51	123.5415	<.0001

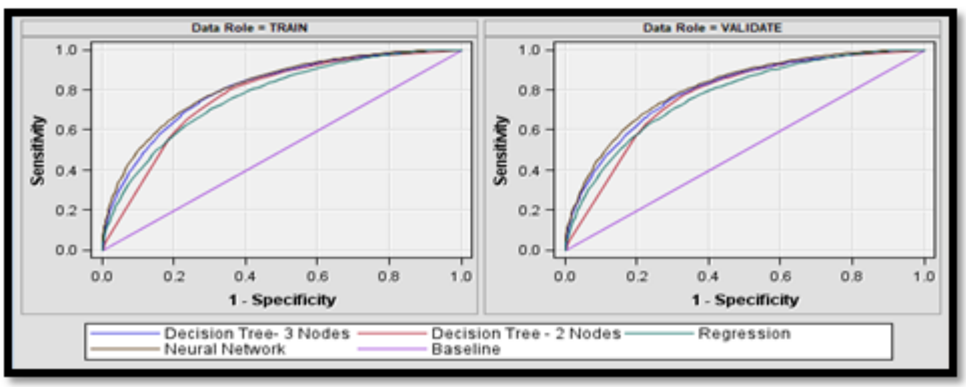
5.3. Supervised Model Comparison

After building all of our models, we passed the results from each to the Model Comparison node in SAS, which produced Figures 4 and 5. The Neural Network performed the best with a misclassification rate of 0.257763 on the validation set, followed by the three-node decision tree with a misclassification rate of 0.262309 on the validation set, the two-node decision tree with a misclassification rate of 0.266261 on the validation set, and then logistic regression with a misclassification rate of 0.290041 on the validation set. This is also evident in the ROC curves in Figure 5. The best performing model will always have an ROC curve closest to the top left corner, and the Neural Network is closer to the top left compared to the other models in this case.

Figure 4: Supervised Model Comparison

Selected Model	Predecessor Node	Model Node	Model Description	Target Variable	Target Label	Selection Criterion: Valid: Misclassification Rate
Y	Neural	Neural	Neural Net...	like		0.257763
	Tree2	Tree2	Decision Tr...	like		0.262309
	Tree	Tree	Decision Tr...	like		0.266261
	Reg	Reg	Regression	like		0.290041

Figure 5: ROC Curves for Each Supervised Model



6. Conclusions and Further Steps

Through this project, we have learned a lot about predicting whether someone will like a movie based on their preferences and demographics. The main conclusions are listed below.

1. Association rules can help us predict if a user will like a movie based on other movies they've already rated highly.

2. Clustering users based on demographics can help us recommend movies based on what other users in the same cluster have rated highly.

3. Neural Networks have the lowest misclassification rate of all the supervised models we tested, allowing us to predict whether a user will like a movie with an accuracy of 74.2237%.

4. The most important variables in predicting whether a user will like a movie are average rating of a movie (over all users), the user's average rating on different genres (drama, action, comedy, etc.), the genre of the movie, the user's age, the user's occupation, and the state the user is from.

If we had more time, we think we could get better results by improving the data set and adding more variables. For example, text tags were provided with the data set. Text mining or sentiment analysis with these tags could help us determine which words are associated with higher ratings, as well as give us more information about why a user liked or did not like a movie. Also, the ID for each movie on IMDB.com was provided, which could be used to scrape more information about each movie. Lastly, additional models could be tested that were not included in this analysis, such as Support Vector Machine and ensemble methods.

References

1. GroupLens. (2009, January 1). *www.grouplens.org*. Retrieved from grouplens: https://grouplens.org/datasets/movielens/
2. Hu, S. (2016, August 08). *https://github.com/MacHu-GWU/uszipcode-project*. Retrieved from pypi.python.org: https://github.com/MacHu-GWU/uszipcode-project/tarball/2016-08-08
3. Konstan, H. F. (2015). The MovieLens Datasets: History and Conext. *ACM Transactions on Interactive Intelligent Systems (TiiS) 5, 4, Article 19*, 1-19.