HW 1 – Stat 578 MovieLens Data Prediction

NetID: shriyak2 Date: 20 Feb 18

The Data:

MovieLens was formed in 1997 by GroupLens Research University of Minnesota and recommends movies for its users to watch, based on their film preferences using collaborative filtering of members' movie ratings and movie reviews. The full data set contains over 26 million ratings and 750 thousand tag applications applied to 45 thousand movies by 270 thousand users with the aim was to gather research data on personalized recommendations [1].
The 100K MovieLens data consists of 100,000 anonymous ratings on a five-star scale from 1,000 users on 1,700 movies. There are four user-related covariates, including gender, age, occupation and zip code. There are 24 item-related covariates. Ratings are integers on a 5-star scale. Each user and each movie are identified by a unique id. The data set includes information about the age (7 age groups), gender, occupation (21 types) and zip code for each user, as well as the title and genre (18 types) for each movie. Each user has at least 20 ratings.

Goal:

The goal of this homework is to threefold:

1. To perform some predictions on the ratings and report the CV error
2. To identify and depict the pattern of missing ratings
3. To improve predictions incorporating the pattern of the missing ratings

Challenges:

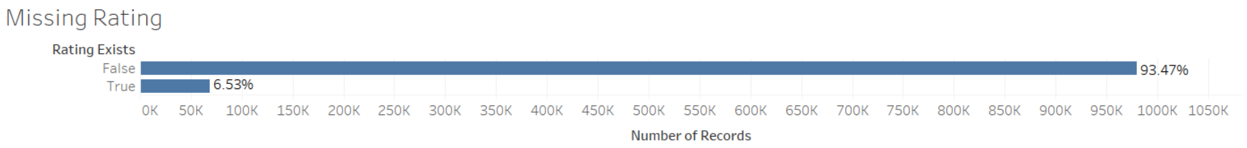
With the vast number of movies available today on Amazon, Netflix or other streaming sites, there are a lot of challenges associated with providing the user with the best recommendations. To aid in these recommendations, we aim to predict the user's scores for a movie and correspondingly provide him a better personalized recommendation. Some challenges for building such a system could be:

1. If a movie has very few ratings, we do not have enough data to predict its recommendations.
2. User ratings are heterogeneous. Some users provide extreme ratings while others provide very similar ratings and these both provide separate challenges.
3. The presence of confounding variables (Oscar Awards, friend's recommendations, availability in a streaming service etc.) are not recording but could influence the user's ratings.

Preliminary Analysis:

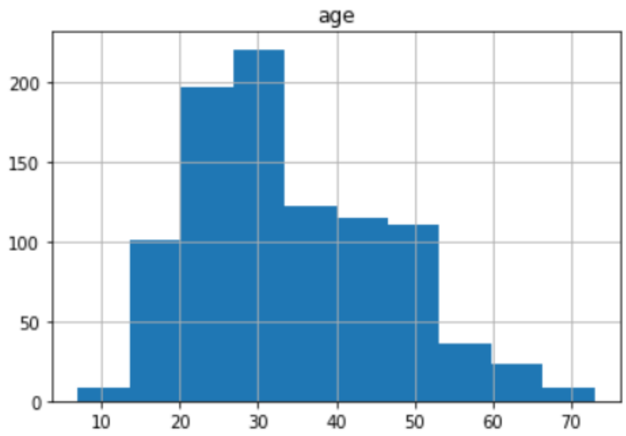
The data is largely cleaned but we perform some manipulations to explode the dataset and attach the features associated with users and movies. We look at the average ratings of movies with respect to certain movie features and then certain users to identify some useful predictive features for the models. There are several transformations we can perform including creating buckets for some of the continuous variables, converting zip codes to latitude and longitude, adding movie features like cast and award information however since the goal of this homework is more about incorporating hidden patterns, we first create an additional column *rating_exists*.

This column *rating_exists* is a Binary indicator which we will use to identify and plot missing patterns. We will show that some data imputation on this column improves the performance for even the most simple predictive model. Roughly 93.47% of ratings are missing and we look at several different cuts of the data that shows the missing rates later corresponding to Part B.



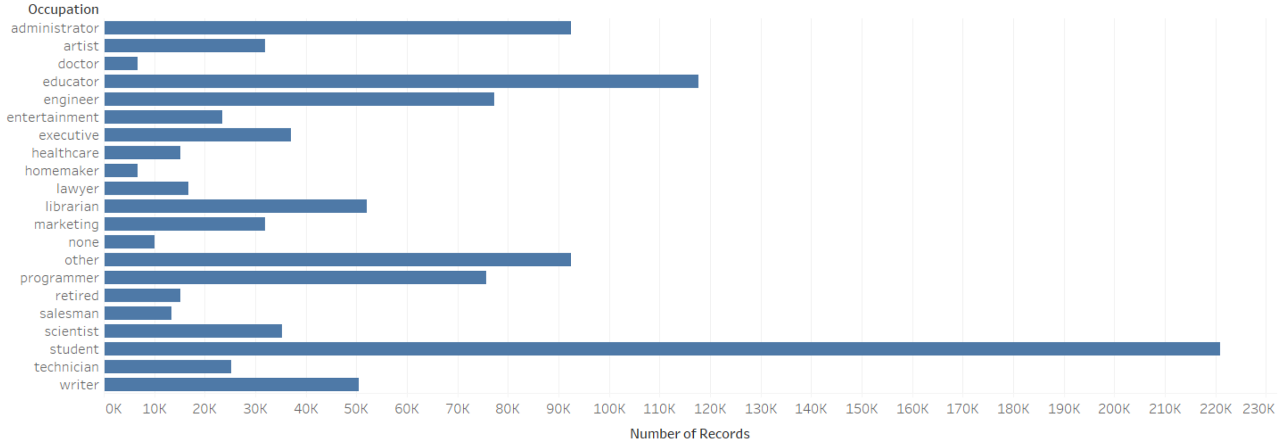
Missing Ratings: 93% of the data is missing

Let us also look at some basic plots of the users and the information we have about them. We see that there is a skew of members in the younger age groups, specifically from 20-35 and the others taper off.



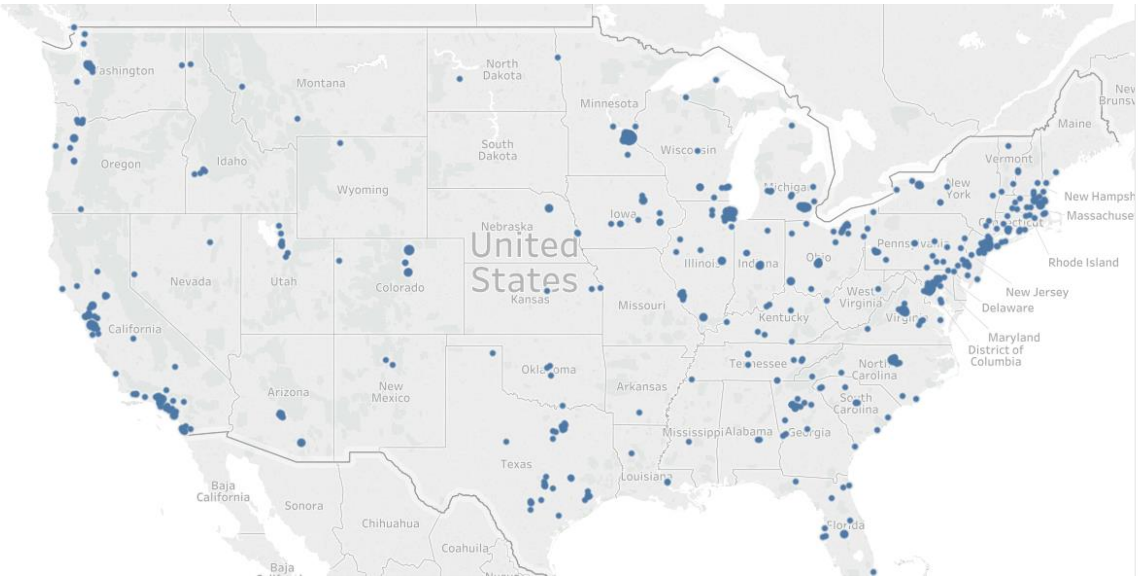
Age: Distribution of users

For occupations, we see a heavy skew towards those from Education, Administration, programmers, engineer and writers. These are all technological savvy fields and make senses that they show up in an internet ratings database. The majority class however is that of students. This matches the age group that we discussed earlier, and students love watching films, have the time and energy to do so and are very engaged with internet recommendation systems.



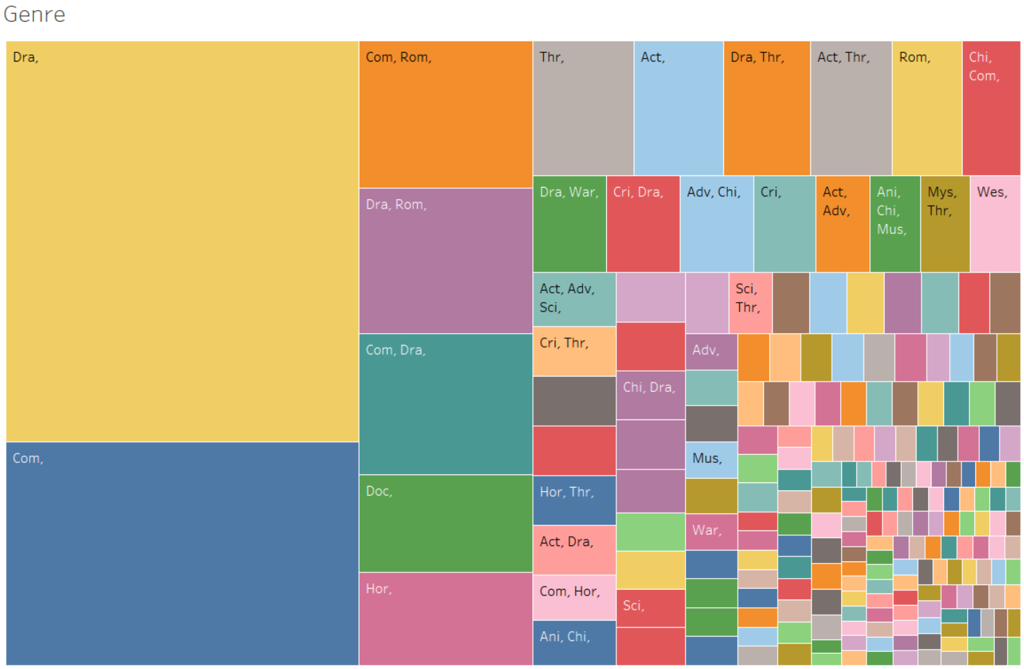
Occuptation: Distribution of users

Looking at the distribution of users across the nation from the zipcode, we see a fairly even distribution from across the country matching the normal population density.



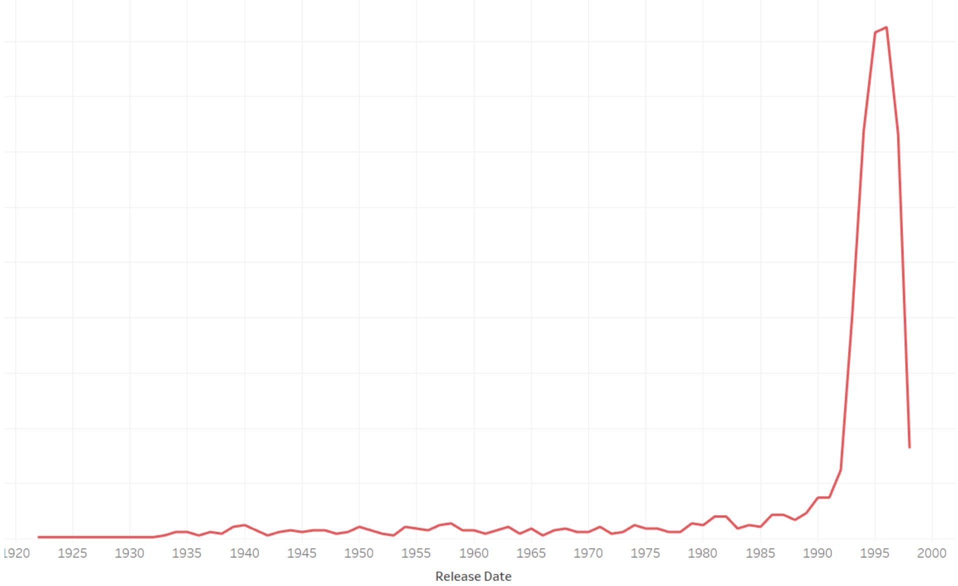
Zipcodes: Distribution of users across the nation

Let us look at similar results for some of the fields for the categories corresponding to the movie items. Over 15 genres are listed and they have been exploded to form a One-Hot-Encoding to enable easy analysis of the data. If we look at the movies by combining all the genres, we can see the most popular genres. Drama is by far the biggest single genre with Comedy, Documentaries, Horror and Thriller being other popular genres in the database. There are multiple popular mixed genres, Romantic-Comedy, Romantic-Drama and Comedy-Drama are the big ones there.



Genres: Distribution across combinations

A distribution of movies by Release Date shows us an expected patter, since the database was formed around 2005, movies from nearly a decade back are the mostly present. There is a fair number of movies from older times and we would guess that these are not very well seen but we will show that it is not the case actually.



Release Date: Distribution of movies

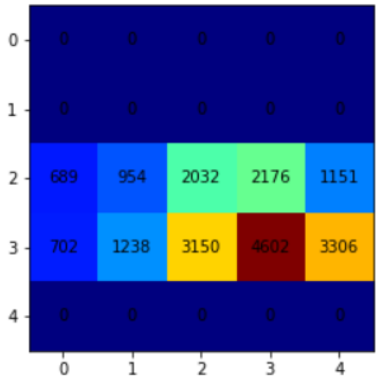
Part A:

Predict the preference score for each user using a 5-fold Cross-Validation and compute the RMSE.

We will employ two methods for the predictions: Linear Regression and Decision Trees. Performing the predictions using the IDs alone, the original *ratings* dataset (u.data) is a futile attempt as it simply contains a set of identification records and no predictive features. We will treat the task as both a classification problem and a regression problem, report the Confusion Matrix, Root Mean Square Error and a couple of other useful metrics to track model performance.

Attempt 0: Fit a Linear Regression model on the raw Ratings data (with no predictive features)

This is an attempt to simply throw the data at the model and get some baseline performance using Python's sklearn library. It will also motivate the poor performance of Linear Regression and lead us to question some of the assumptions the model makes.



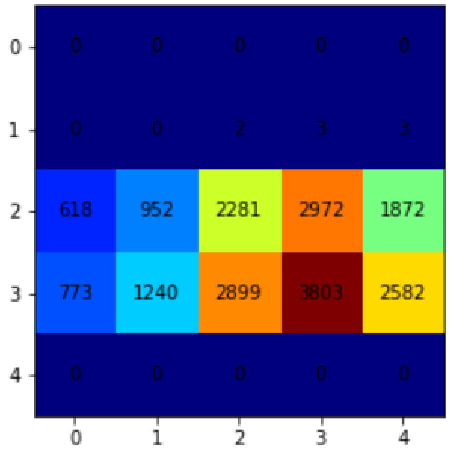
	0	1	2	3	4
0	0	0	0	0	0
1	0	0	0	0	0
2	689	954	2032	2176	1151
3	702	1238	3150	4602	3306
4	0	0	0	0	0

We fit the model for the User_ID, Movie_ID and Timestamp, here is the equation for the linear regression model:

Score = 3.87 – 4.1e-5*User_ID – 6.28e-4*Movie_ID – 6.06e-11*Timestamp

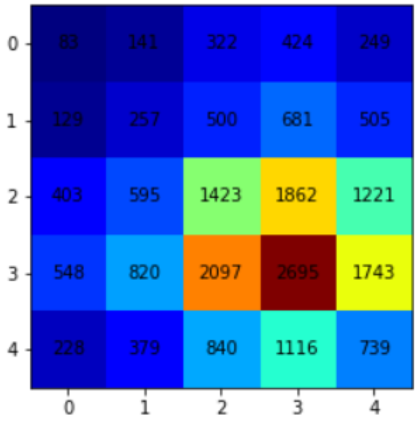
Apart from being a meaningless model in terms of features, it also has poor model performance as seen by the confusion matrix above. It also predicts values in the range 3-4 and that is not very useful. Since the majority of the ratings are around 3 or 4, this model still has a low rmse for a 5-fold CV:
[1.23417588 1.19474463 1.1633054   1.15975578 1.17085502]
Final RMSE: 1.08830498356

Attempt 1: Ridge Regression after adding predictive features
We add all the predictive features but only consider the numeric features (dropping elements like imdb url, movie title and other string factors). We expect that adding features will strongly improve the performance of the model but unfortunately, that is not the case. The confusion matrix for the model shows that the Ridge does not perform much better than the model showing that the relationship between these features are not linear and we must explore other models.



We definitively see better results, but the predictions are still in the 3-4 range. The coefficients for the features using the tuning parameter *alpha=100* are shown in the Jupyter Notebook. Results for RMSE:
[1.23411223 1.19452638 1.16332716 1.15967624 1.17112594]
Final RMSE: 1.08829895308

Attempt 2: Decision Tree Classifier using the predictively features
This problem can be tackled using a DT Classifier and that has several advantages for us. We obtain a interpretable model which can help us understand the most important features, it makes no underlying assumptions on the distribution of the data and using CV, we can avoid overfitting on any given single dataset. The performance of the DT Classifier is far better than that of the two Regression models.



Confusion Matrix for Ratings: Decision Tree Classifier

We have overcome the issue where the model could not represent the data and have adequately represented different classes. While the model still has confusion distinguishing between scores of 3 and 4, it still has a pretty decent predictive accuracy. The important features for the Classifier are:

0.2044	0.1940	0.1702	0.1391	0.0128	0.0106	0.0102	0.0095	0.0091
timestamp	user_id	age	movie_id	Drama	Romance	Thriller	Comedy	War

Feature Importance: Decision Tree

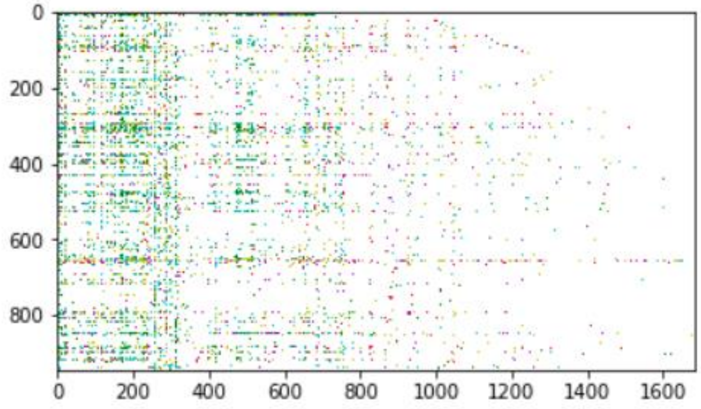
However, we still have not been able to reduce the error too much and for that, we will first analyze the patterns in the missing data. RMSE results for the five folds:

```
[ 6.30783461  2.09655     1.97735     2.01645     2.85614281]
Final RMSE: 1.69514003094
```

Note, we used a 5-fold CV with predictive factors in addition to the IDs because we believed that the IDs alone did not contain the predictive power, hence we do not report the CV of the 5 datasets in the data but perform our own 5-fold CV in Python. The model on the CV generated in the original dataset is present in the code.

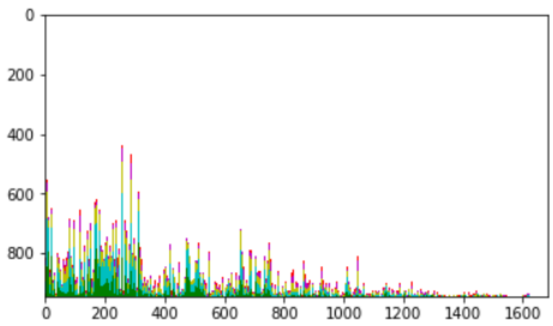
Part B: Does the missing data occur at random? Answer, NO!

Now we will show through our analysis and visualizations that the missing data does not occur at random but that there is a pattern to the missing ratings. We can use this pattern to perform some imputation and get better predictive results. The first task here is get the Movie-User matrix and represent the scores as a HeatMap.

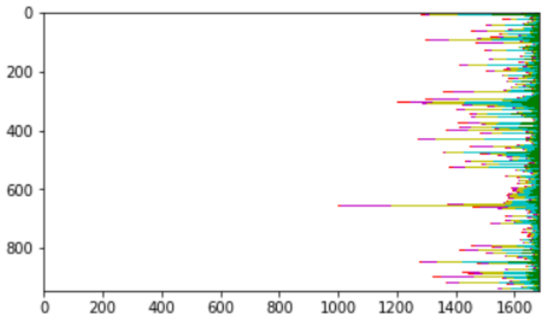


Ratings: Users and Movies

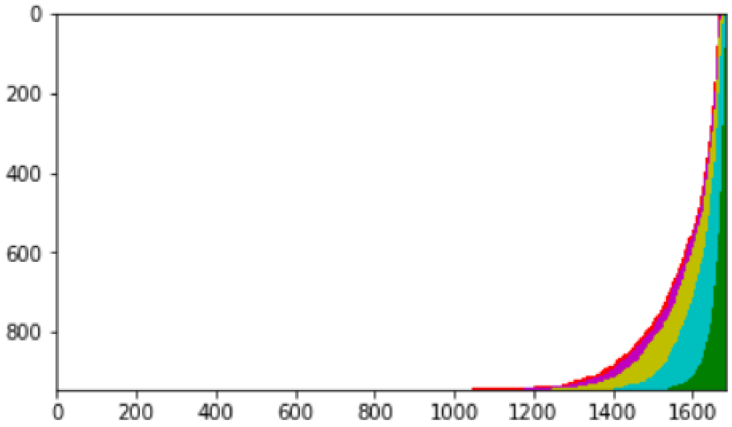
This shows the distributions of scores for movies and users and there is a definite pattern of some movies being rated higher than others, some users rating movies higher than others and some movies, users both having a lot of ratings while some others are always empty. Let us now take a look at the ratings sorted by movies and then the ratings sorted by users ie. The movies with the highest average ratings and the users with the highest average ratings.



Ratings: Sorted by Users

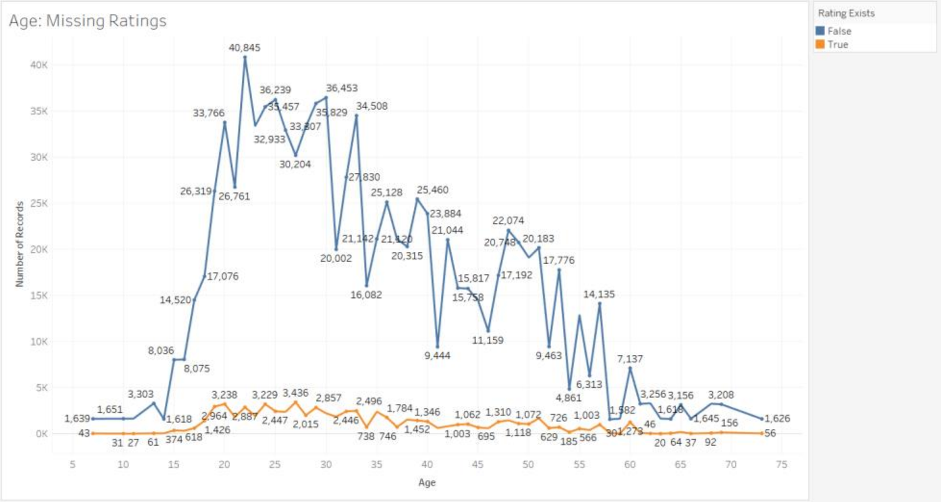


Ratings: Sorted by movies



Ratings: Sorted by users and movies

This clearly shows that the ratings are not missing at random, that there are patterns and clusters to the available ratings so let us take a look at the distribution of the ratings for some of the other factors for each the movies and ratings.

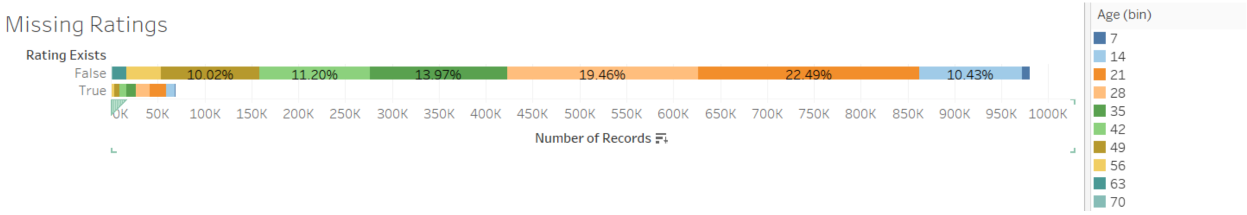


Age: Missing values pattern

This is the first clear indicator of a massive population of missing values within a specific subcategory. While the number of members in the age group 18-28 is higher than the other populations, we see that the vast number of missing ratings corresponding to that age group represent a couple of basic facts:

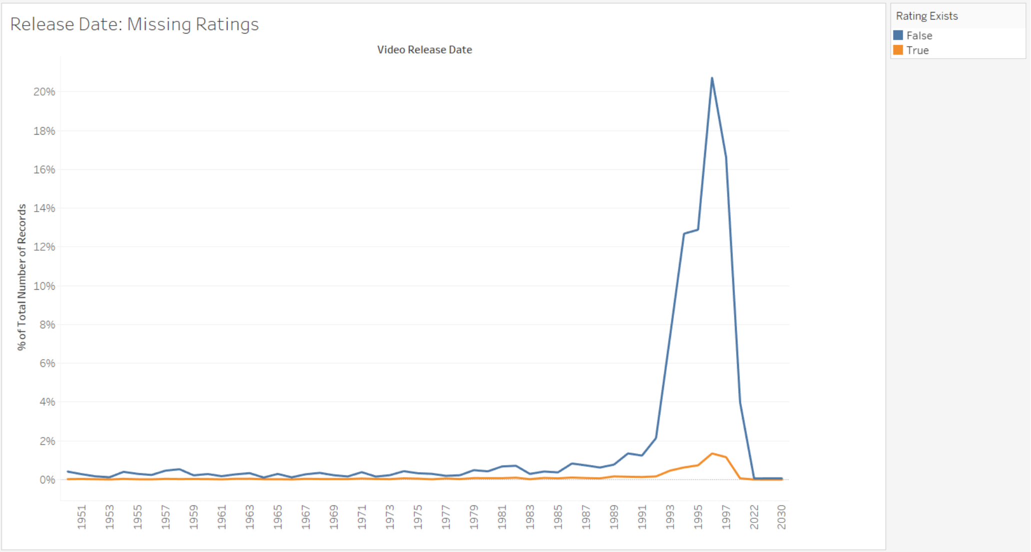
1. There is a maximum number of movies that a user can watch and hence rate
2. Younger users may not watch movies released before they were born

Interestingly, the missing elements reveals a bimodal distribution with another peak around the 46-52 age group and this could be an interesting group to impute the data for as well. Looking at the different bins that contribute to the missing values reveals some more insight.



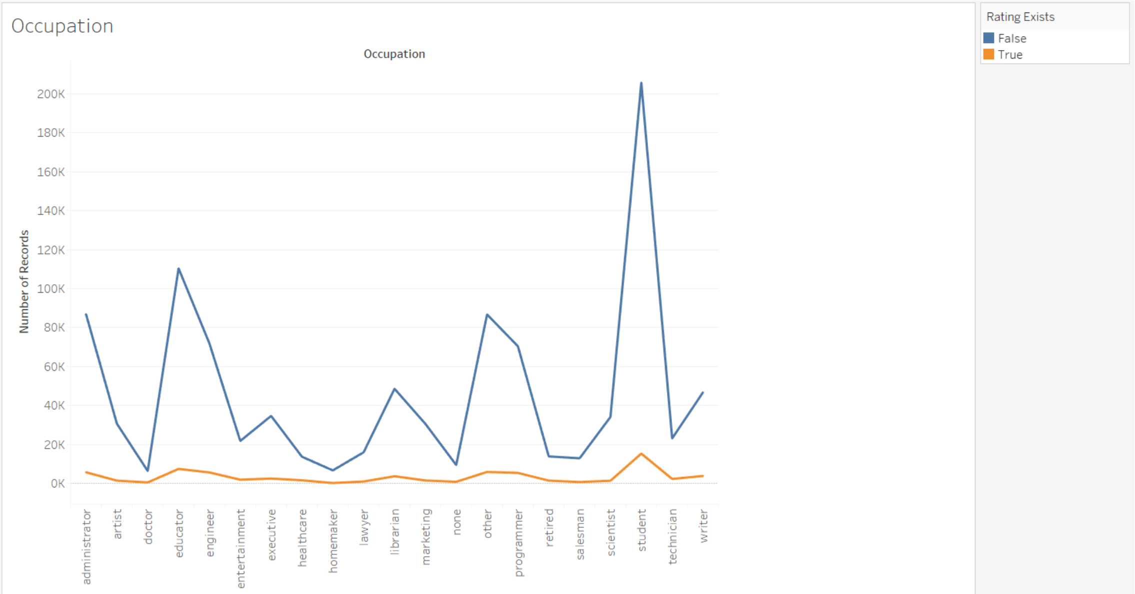
Missing Ratings: Contribution to missing values by Age Bin

Let us now look at the ratings for the Release Date. This sharp peak in the 1990s corresponds to the large number of films collected from this date range that most users may not have watched. Users often match only popular movies and it is a bad idea to have a dataset that collects data so heavily from one period of time as seen in Figure 6. Some imputation of the data along this axis might be beneficial for analysis as well.



Release Date: Missing values pattern

We finally also show a pattern in the missing data for Occupation. While I explored the pattern for several of the other features, not all of them had a significant pattern to analyze.

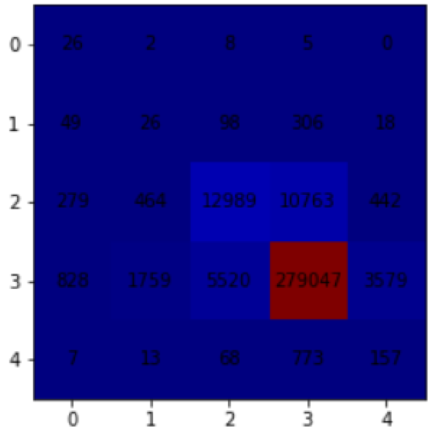


Occupation: Missing values pattern

This has conclusively shown us that there is a missing pattern along age, occupation and release year. What we will then do is perform imputation of values across the mean for this triplet of values and also attempt to impute the values by other smarter techniques.

Part 3: Enhance predictive performance using the missing patterns

We obtain the mean ratings across each set of values for Occupation, Release Date and Age, impute the values in the dataset with the mean values and then perform the predictions. While this is one imputation we can perform, we could also have used the mode, median or a custom imputation performed by a Decision Tree on those set of values. Here are the results of the Decision Tree after the imputation has been performed.



Confusion Matrix: Decision Tree with Data Imputation

Clearly, the results are far better now compared to the original results. The feature importance shows us the important elements with the highest predictive power for this dataset.

0.4549	0.3363	0.1712	0.0053	0.0038	0.0034	0.0032	0.0025	0.0024	0.0021
age	user_id	movie_id	Romance	Adventure	Thriller	Crime	Action	War	Musical

Feature Importance: Decision Tree with data imputation

Conclusion:

For a dataset where over 90% of the data is missing and there is a need to perform recommendations, we cannot simply ignore the missing values due to the cold start problems mentioned. The solution to that is to impute the missing values with sensible predictions for the users, then use that as the training set to perform new predictions using all the features available in our data. This will have high correlation with the initial imputation performed so we must be careful to avoid a method that centers around the mean like Regression as it will not be able to perform personalized suggestions.

We employ Decision Trees as they do not make any assumptions on the structure of the underlying data and provide an easy model that we are able to interpret. We were able to impute the missing values using the patterns we detected for Occupation, Release Date and Age. This incorporates both movie and user features and hence provides us with improved predictive power. The imputation across the Age category was especially powerful as demonstrated by the increased predictive power of Age relative to the earlier model

References:

[1] https://en.wikipedia.org/wiki/MovieLens