Big Data in Cloud Platforms

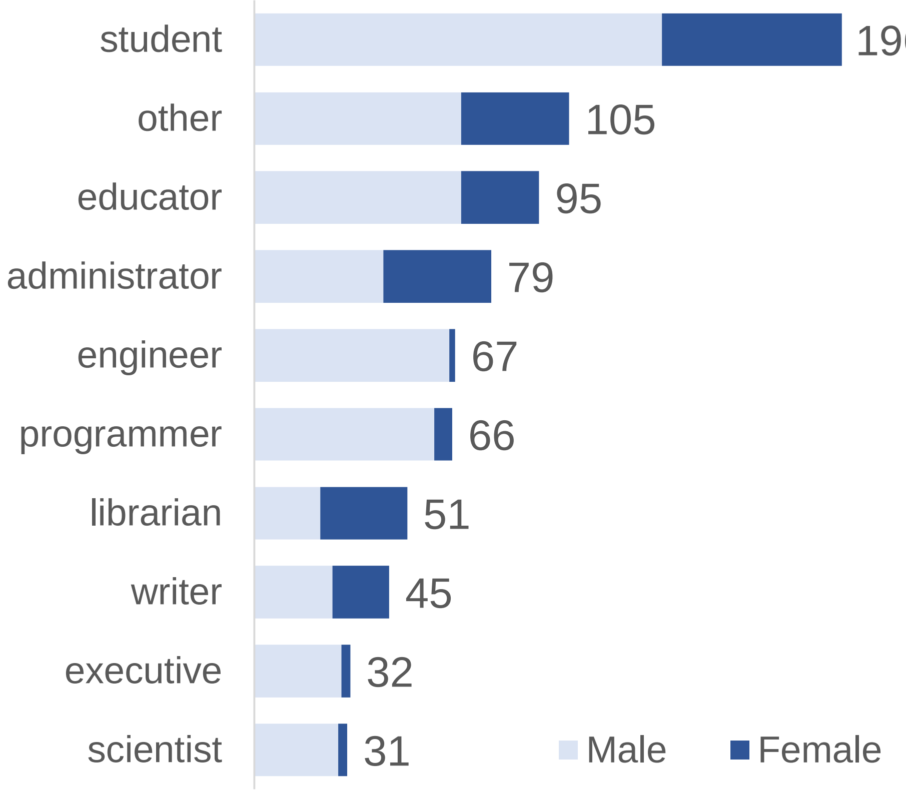
Case analysis: 100k MovieLens Dataset

André Águas	M20170973@novaims.unl.pt
António Correia	M20170975@novaims.unl.pt
João Januário	M20170985@novaims.unl.pt
Valter Bento	M20170999@novaims.unl.pt

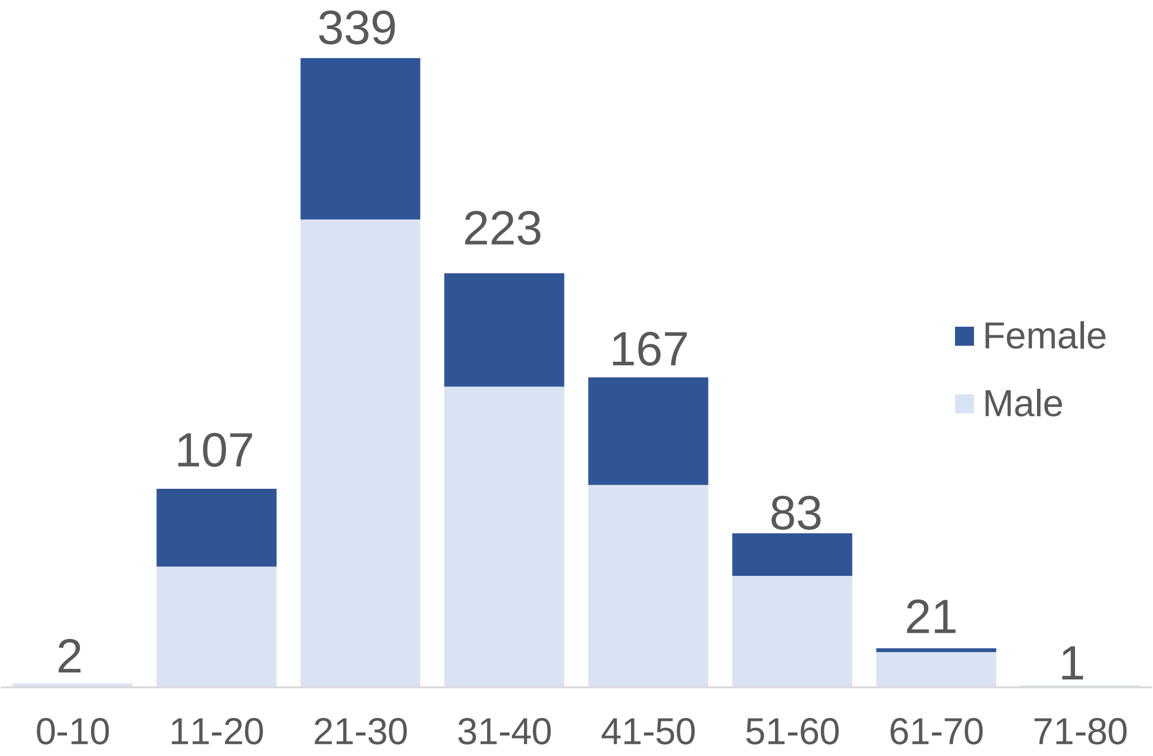
WHO is rating movies in our data?

Total 943 users
71% male

Number of users by occupation
(top 10 occupations, 81% of users)

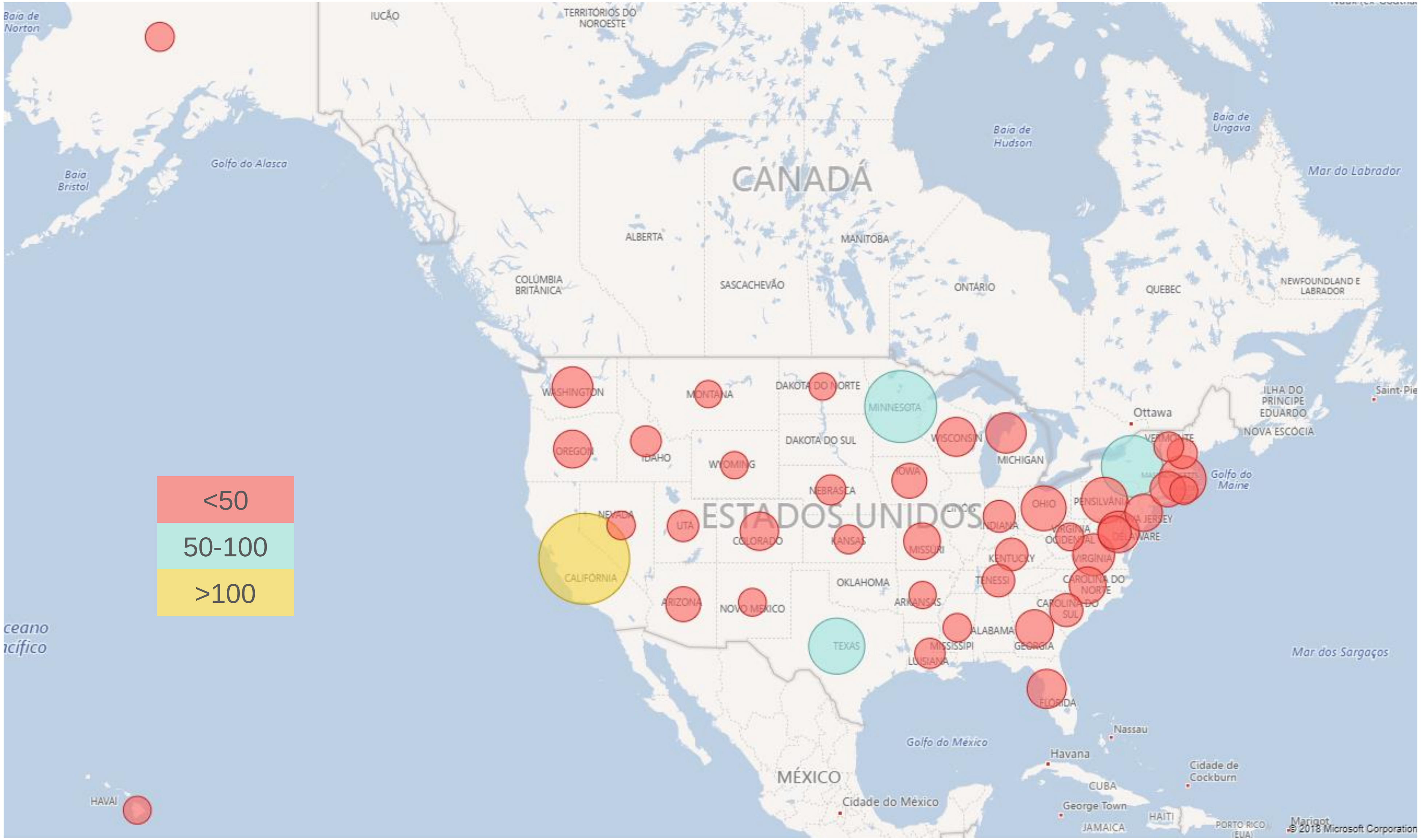


Number of users by age interval



WHO is rating movies in our data?

Total 943 users

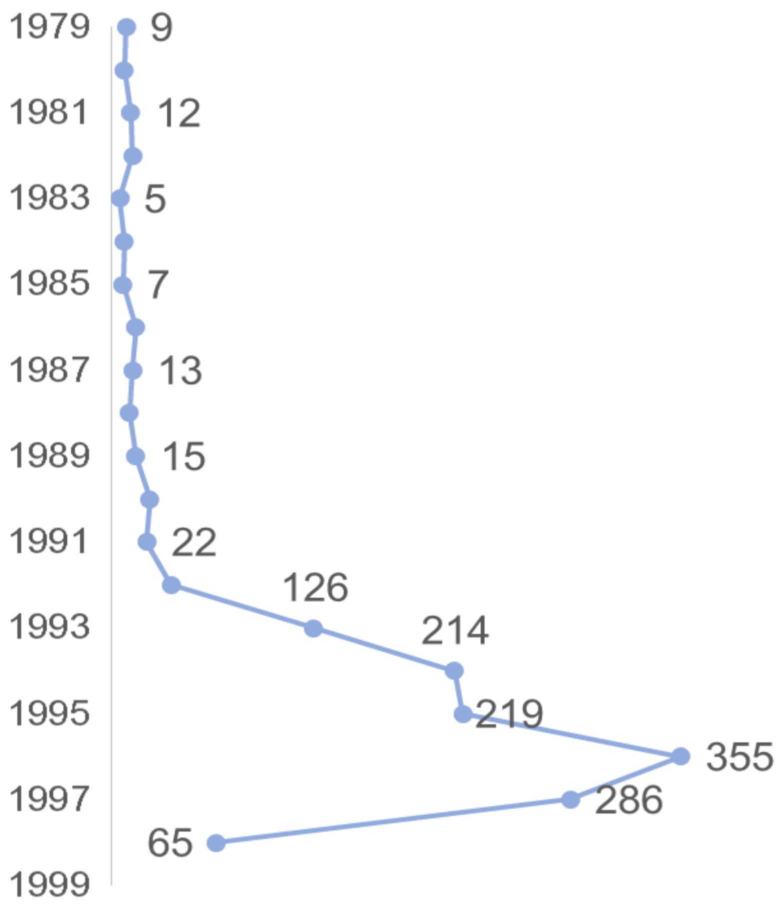


WHAT movies are being rated?

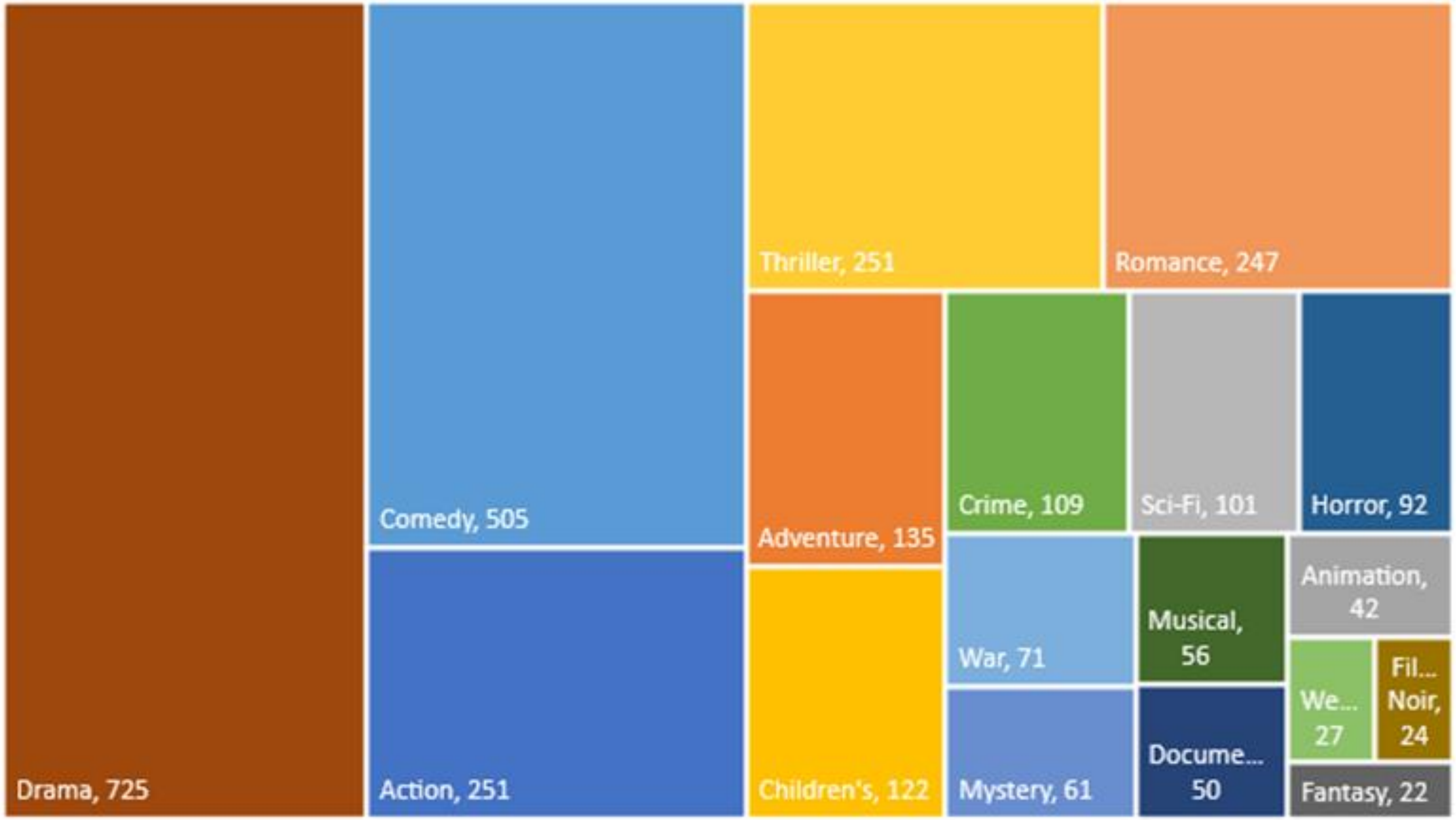
Total 1681 movies

Movies by year of release

(latest 20 years, 87% of movies)

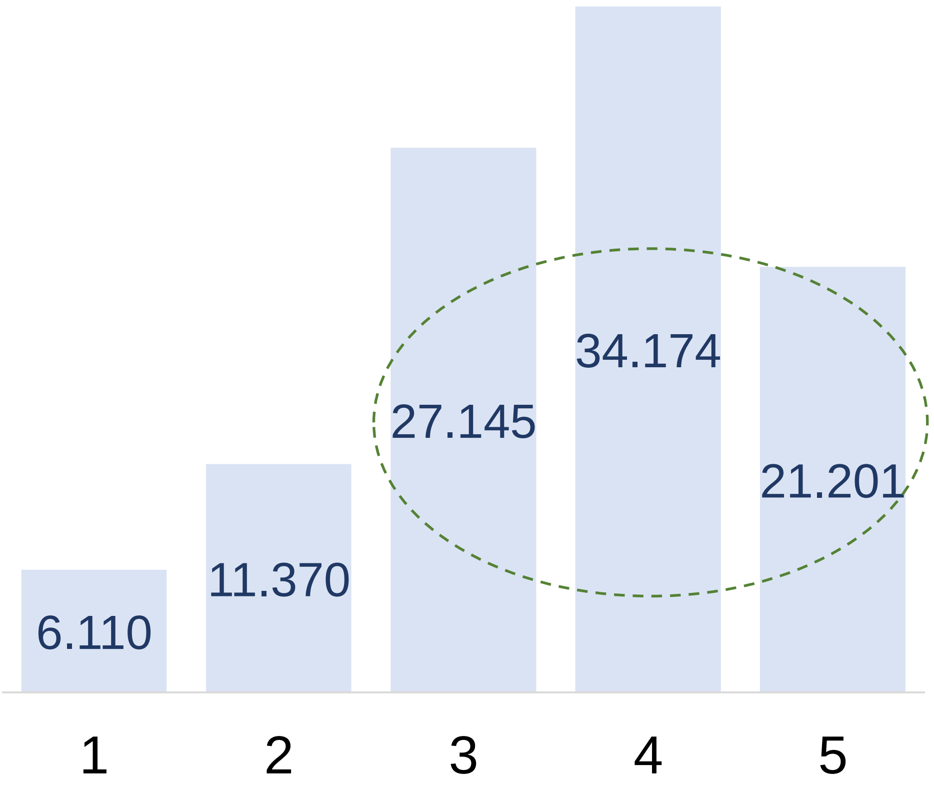


Movies by genre



HOW do users rate movies?

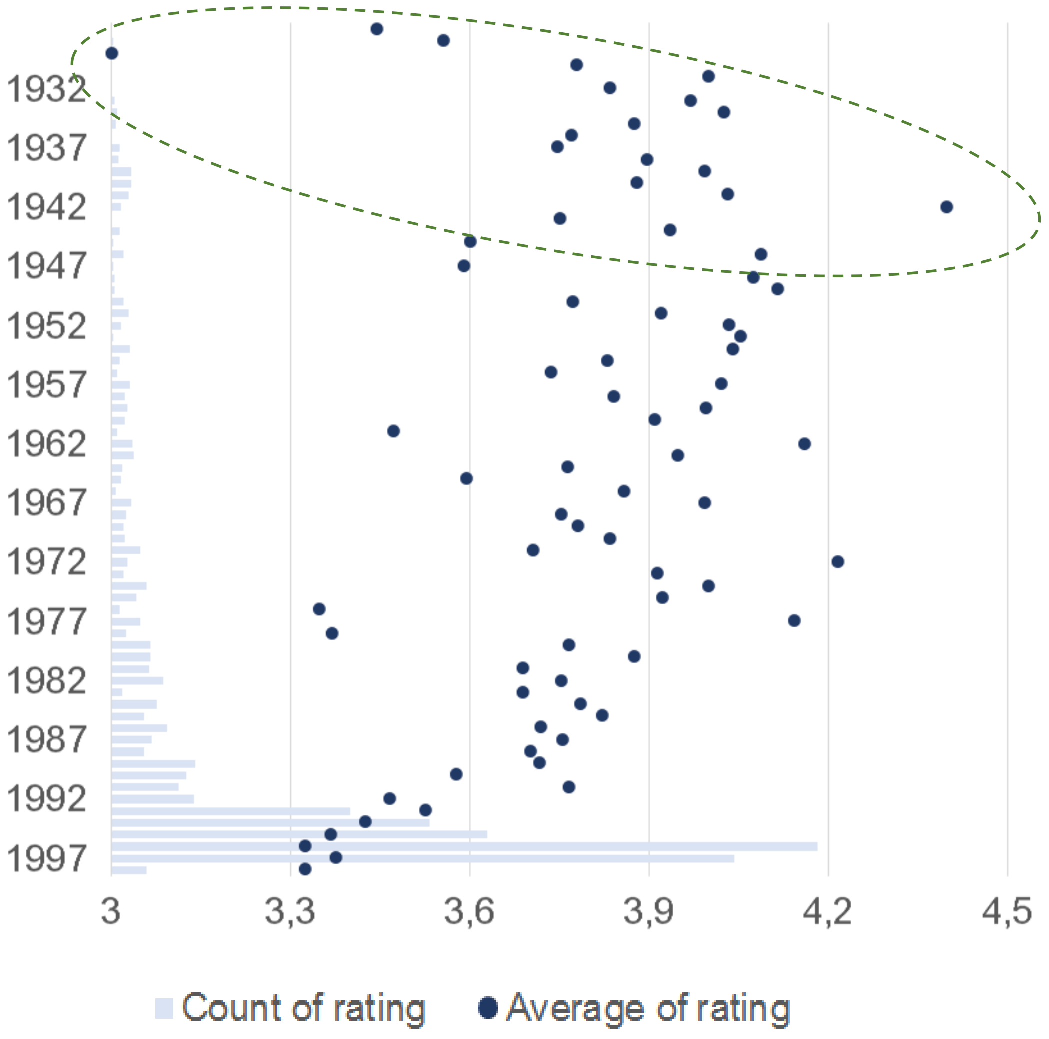
Number of ratings per category



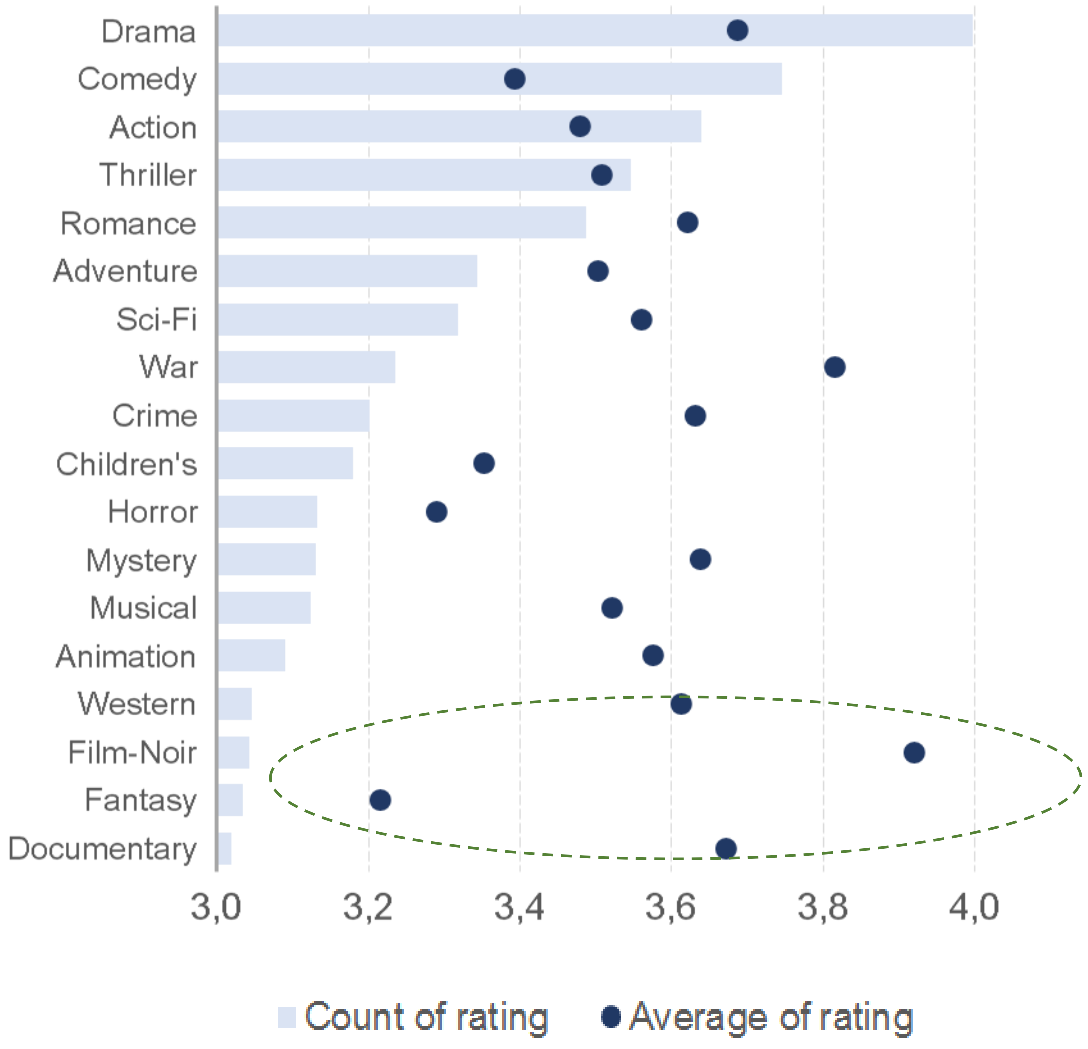
- Average rating = 3,5
- Equal average across male/female users
- 83% of ratings are between 3 and 5
- Very few 2s and 1s

Do movies age well? Do people like a bit of drama?

Rating by movie year of release

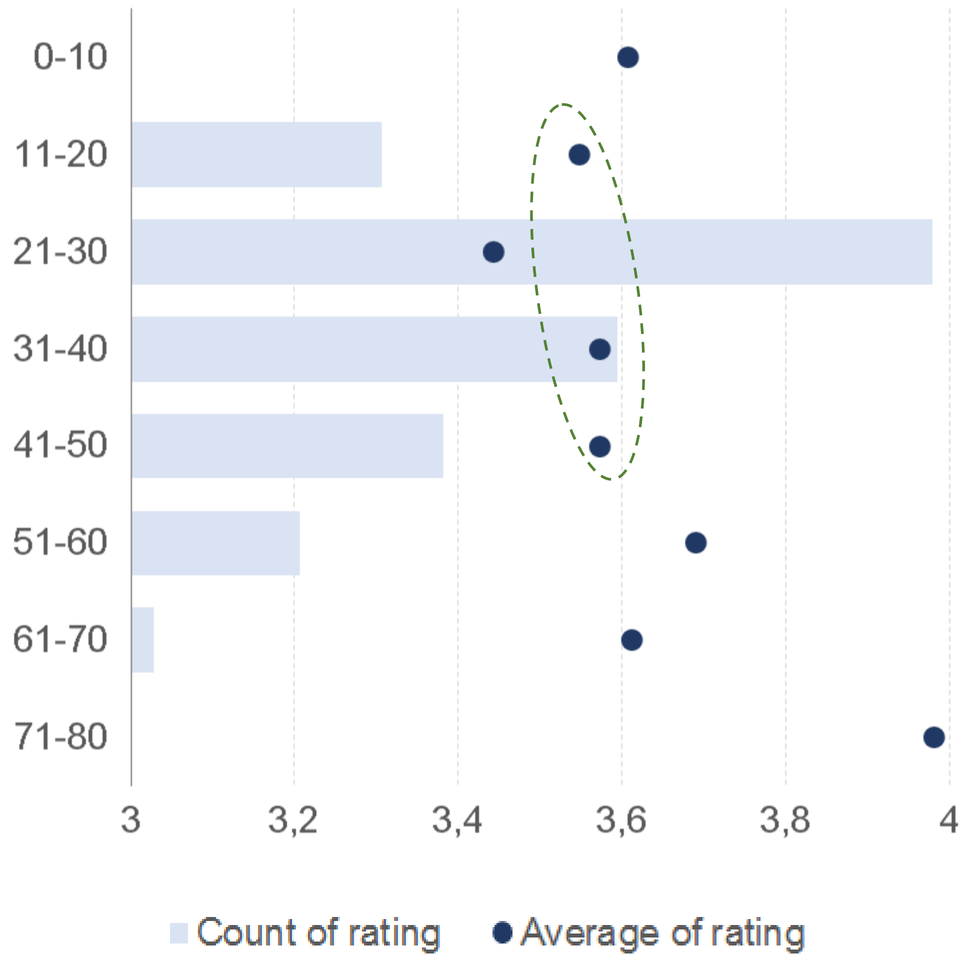


Rating by movie genre

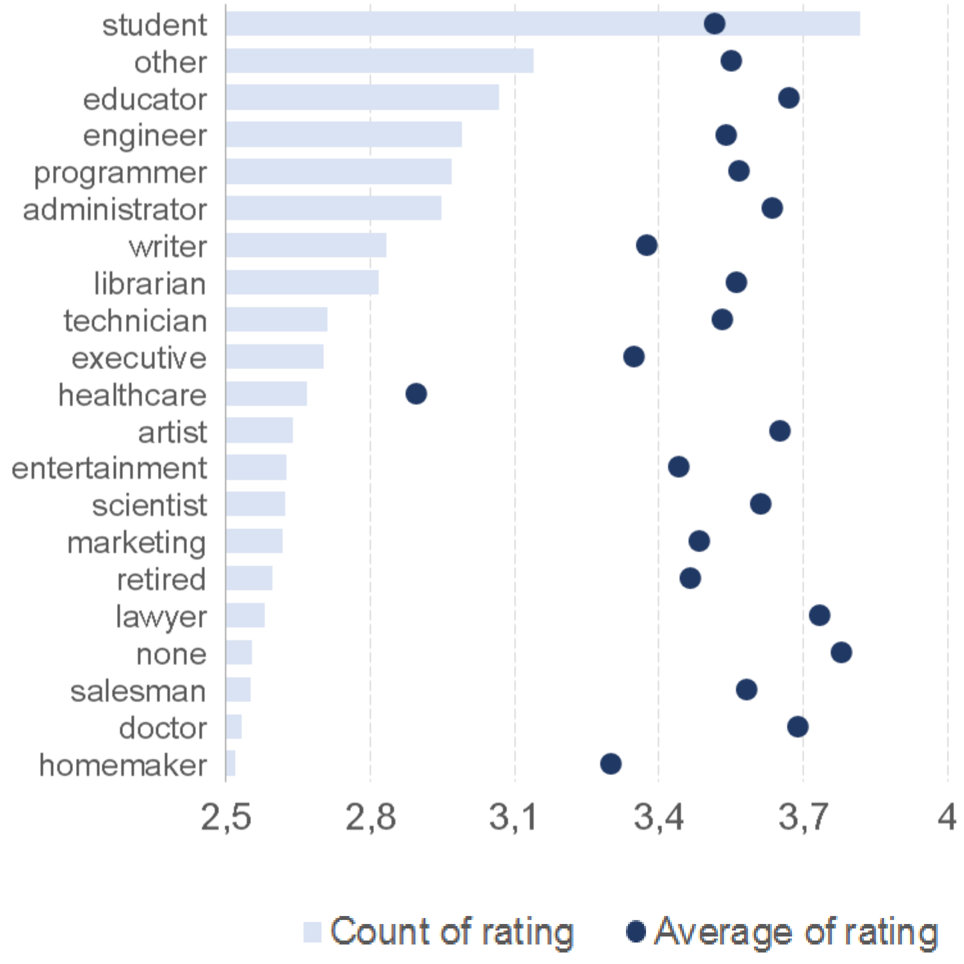


Are older users and lawyers being more generous?

Rating by user age

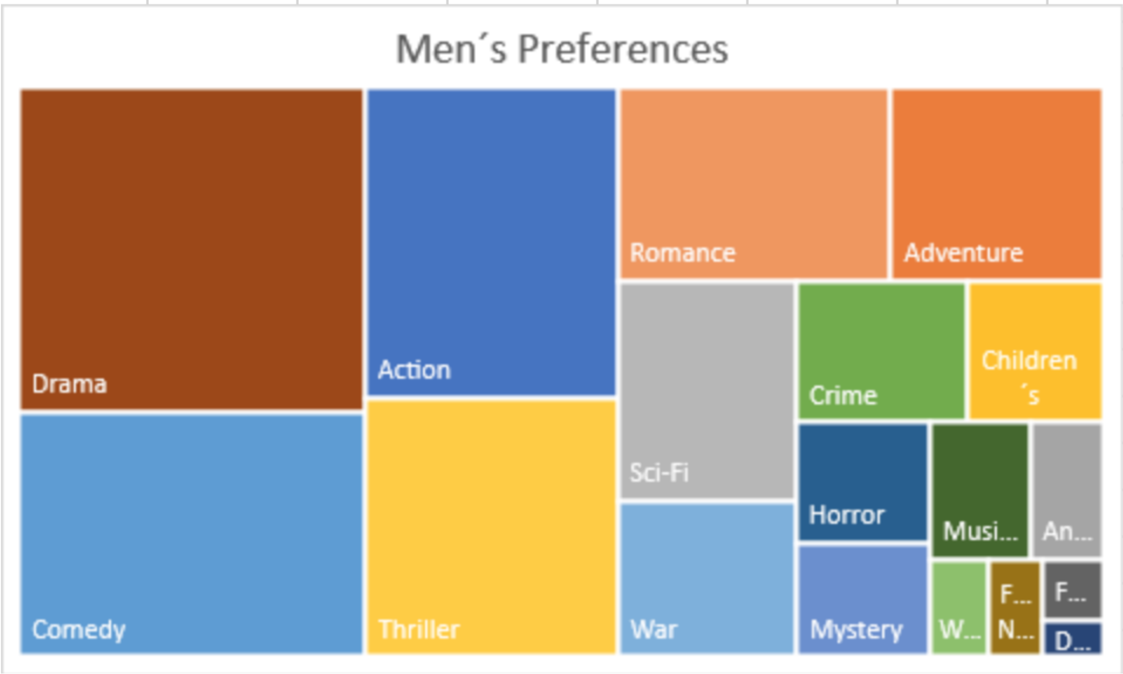


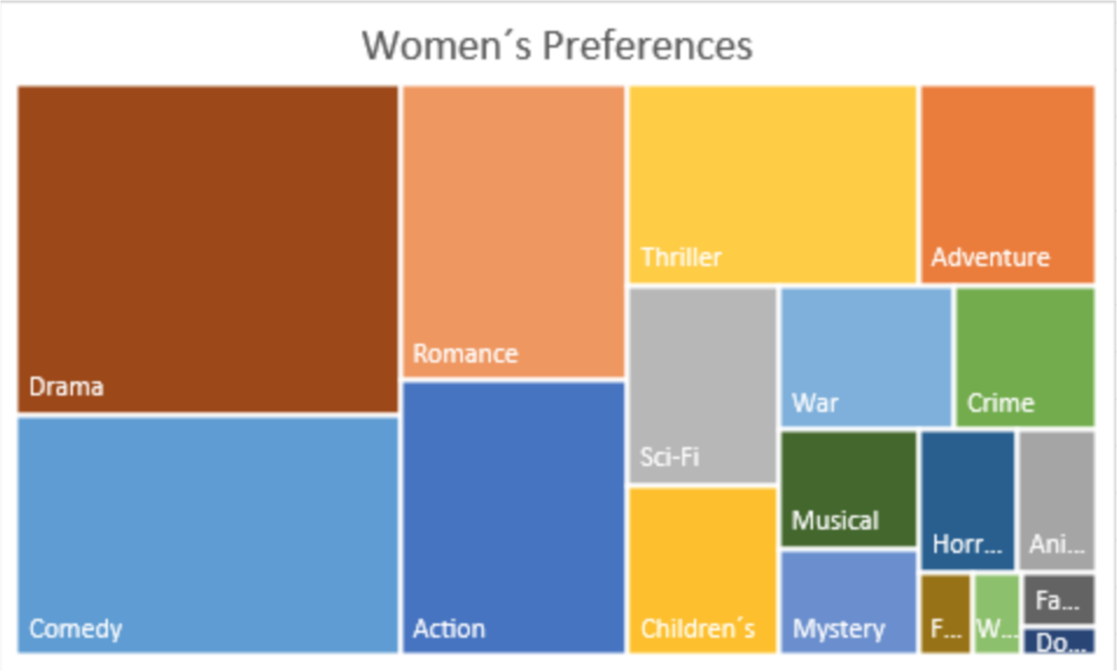
Rating by user occupation



What are the gender preferences on movie types?

We have counted the number of reviews by movie by genre in order to find men's and women´s preferences





Can we predict a rating for a pair user-movie?

- Random forest model
- The features with most predictive value were:
  - Year of movie release
  - Manhattan and Baltimore zip code dummies
  - Drama, war and romance dummies
  - User age
- The model is only slightly better than random (average accuracy of 29%)

Confusion matrix for model predictions

Actual \ Predicted	1	2	3	4	5
1	**45%**	16%	10%	12%	17%
2	23%	**23%**	16%	15%	24%
3	16%	17%	**16%**	19%	31%
4	13%	11%	12%	**23%**	41%
5	11%	7%	7%	18%	**56%**

Average accuracy = 29%

Limitations and the way forward for the analysis

Limitations

- Performance with 100.000 record database turned out to be low using the VM that we have been working with in class
- Exporting data from queries to visualization and modelling tools was done in a manual way
- Rating model is a first step, but could be improved

Way forward

- Given the goal of the project, aim at obtaining deeper insight with smaller dataset
- We decided to export data from queries manually, but we could connect data automatically to obtain better scaling
- Improve our rating model and even attempt to produce a recommendation system based on users' previous ratings

Annexes

Hive Tables Structure

- Table ‘u_user’

Attribute	Data Type
userId	int
userAge	tinyint
userGender	string
userOccupation	string
userZIPCODE	string

- Table ‘u_genre’

Attribute	Data Type
genre	string
genreId	tinyint

- Table ‘u_movie’

Attribute	Data Type
movieId	int
movieTitle	string
movieDate	string
ignore	string
movieURL	string
genre_1	tinyint
genre_2	tinyint
genre_3	tinyint
genre_4	tinyint
genre_5	tinyint
genre_6	tinyint
genre_7	tinyint
genre_8	tinyint
genre_9	tinyint
genre_10	tinyint
genre_11	tinyint
genre_12	tinyint

Exploratory Analysis Querys: **USERSc**

Occupation per Gender

```
SELECT userOccupation, count(CASE WHEN userGender='M' THEN 1 END) AS male_cnt, count(CASE WHEN
userGender='F' THEN 1 END) AS female_cnt
FROM u_user
GROUP BY userOccupation
```

Age per Gender

```
SELECT FLOOR(userAge/5.00)*5 AS bucket_floor, count(*) AS COUNT
FROM u_user WHERE userGender ="M"
GROUP BY 1
ORDER BY 1;
```

Users per ZIPCODE

```
SELECT userZIPCODE,count(*)
FROM u_user a
INNER JOIN ratings b on a.userId=b.userId
GROUP BY userZIPCODE
ORDER BY userZIPCODE
```

Exploratory Analysis Querys: **MOVIES**

Unpivot Genre

```
CREATE VIEW IF NOT EXISTS movie_genre AS
SELECT movieId, genre FROM
(SELECT movieId, MAP("unknown",genre_1,  "Action",genre_2,  "Adventure",genre_3,  "Animation",genre_4,
"Children's",genre_5,  "Comedy",genre_6,  "Crime",genre_7,  "Documentary",genre_8,  "Drama",genre_9,
"Fantasy",genre_10,  "Film-Noir",genre_11,  "Horror",genre_12,  "Musical",genre_13,  "Mystery",genre_14,
"Romance",genre_15,  "Sci-Fi",genre_16,  "Thriller",genre_17,  "War",genre_18,  "Western",genre_19) as map1
FROM u_movie) as t1
LATERAL VIEW EXPLODE(map1) xyz as genre, m_val  WHERE m_val=1
```

Gender Count

```
SELECT DISTINCT genre, count(genre) FROM movie_genre GROUP BY genre
```

Exploratory Analysis Querys: **Preferences**

Movie Preference by Gender

```
SELECT C.userGENDER, SUM( CAST(A.ACTION AS INT) ), SUM( CAST(A.Adventure AS INT) ), SUM(
CAST(A.Animation AS INT) ), SUM( CAST(A.CHILDREN AS INT) ), SUM( CAST(A.Comedy AS INT) ), SUM(
CAST(A.Crime AS INT) ), SUM( CAST(A.Documentary AS INT) ), SUM( CAST(A.Drama AS INT) ), SUM(
CAST(A.Fantasy AS INT) ), SUM( CAST(A.Film-Noir AS INT) ), SUM( CAST(A.Horror AS INT) ), SUM( CAST(A.Musical
AS INT) ), SUM( CAST(A.Mystery AS INT) ), SUM( CAST(A.Romance AS INT) ), SUM( CAST(A.Sci-Fi AS INT) ), SUM(
CAST(A.Thriller AS INT) ), SUM( CAST(A.War AS INT) ), SUM( CAST(A.Western AS INT))
FROM U_ITEM AS A
JOIN U_DATA AS B ON B.ITEMID=A.MOVIEID
JOIN U_USER AS C ON C.USERID=B.USERID
GROUP BY C.userGENDER
ORDER BY C.userGENDER ASC
```

Average Rating by Gender

```
SELECT B.userGENDER, AVG(A.RATING)
FROM U_DATA AS A
JOIN U_USER AS B ON B.USERID=A.USERID
GROUP BY B.userGENDER
ORDER BY AVG(A.RATING) DESC
```

Exploratory Analysis Querys: **Ratings by Occupation**

Average and Count Rating by Occupation

```
Select B.userOccupation, Avg(A.Rating), COUNT(A.Rating)
from ratings as A
JOIN u_user as B
on A.UserID=B.UserID
Group By B.userOccupation
```

Exploratory Analysis Querys: **Ratings by Age**

Average and Count Rating by Age

```
Select B.userAge, Avg(A.Rating), COUNT(A.Rating)
from ratings as A
JOIN u_user as B
on A.UserID=B.UserID
Group By B.userAge
```

Exploratory Analysis Querys: **Ratings count**

Ratings count

```
Select A.Rating, COUNT(A.Rating)
from ratings as A
Group By A.Rating
Order by A.Rating
```

Exploratory Analysis Querys: **Rating by movie year of release**

Average and Count Rating by MovieDate

```
Select right(B.movieDate,4), Avg(A.Rating), COUNT(A.Rating)
from ratings as A
JOIN u_movie as B
on A.movieID=B.movieID
Group By right(B.movieDate,4)
```

Exploratory Analysis Querys: **Rating by movie genre**

Average and Count Rating by Movie Genre

```
Select B.genre, Avg(A.Rating), COUNT(A.Rating)
from ratings as A
JOIN movie_genre as B
on A.movieID=B.movieID
Group By  B.genre
```

Random forest model description

- Random forest model with 100 decision trees and maximum tree depth of 8
- We use Synthetic Minority Over-sampling (SMOTE) to address the problem of imbalance of ratings categories, given that 83% of ratings fall between 3 and 5
- Model accuracy is somewhat low, partly due to not having many features to extract predictive power from

Confusion matrix for model predictions

Actual \ Predicted	1	2	3	4	5
1	**45%**	16%	10%	12%	17%
2	23%	**23%**	16%	15%	24%
3	16%	17%	**16%**	19%	31%
4	13%	11%	12%	**23%**	41%
5	11%	7%	7%	18%	**56%**

Average accuracy = 29%