Modelling a MovieLens Dataset

Aditya Sarma Desai

Contents

Introduction

Chosen Data Set is MovieLens Data set

This dataset (ml-latest-small) describes 5-star rating and free-text tagging activity from [MovieLens](http://movielens.org), a movie recommendation service. It contains 100004 ratings and 1296 tag applications across 9125 movies. These data were created by 671 users between January 09, 1995 and October 16, 2016. This dataset was generated on October 17, 2016.

Users were selected at random for inclusion. All selected users had rated at least 20 movies. No demographic information is included. Each user is represented by an id, and no other information is provided.

The data are contained in the files `links.csv`, `movies.csv`, `ratings.csv` and `tags.csv`.

Below are the Download URL for the CSV files (dataset)

https://www.kaggle.com/tushar987/movielens/
https://www.kaggle.com/tushar987/movielens/downloads/links.csv/1
https://www.kaggle.com/tushar987/movielens/downloads/movies.csv/1
https://www.kaggle.com/tushar987/movielens/downloads/ratings.csv/1
https://www.kaggle.com/tushar987/movielens/downloads/tags.csv/1

View the data in neo4j

```
// see the top 5 rows in the file
LOAD CSV WITH HEADERS FROM   "file:///MovieLens/movies.csv" AS line
return line.movieId,line.title,line.genres
Limit 5
```

Screenshot of the output

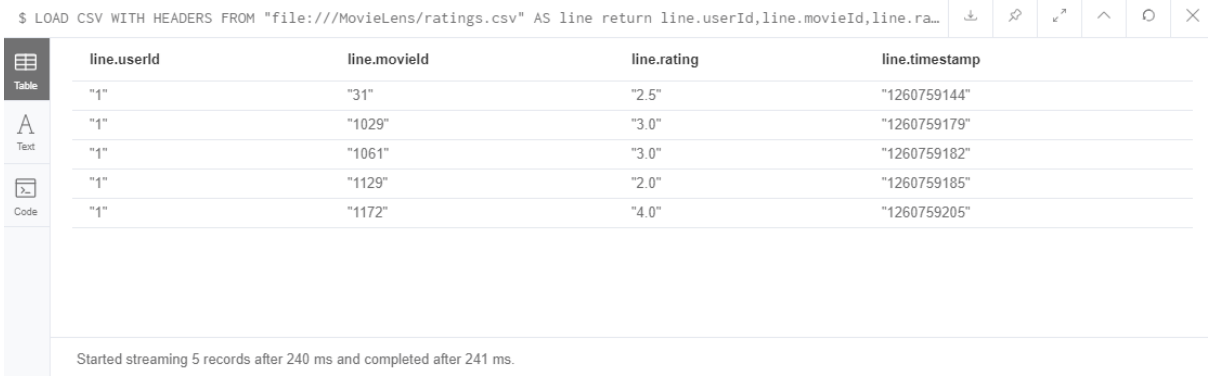
$ LOAD CSV WITH HEADERS FROM "file:///MovieLens/movies.csv" AS line return line.movieId,line.title,line.genr…

line.movieId	line.title	line.genres
"1"	"Toy Story (1995)"	"Adventure|Animation|Children|Comedy|Fantasy"
"2"	"Jumanji (1995)"	"Adventure|Children|Fantasy"
"3"	"Grumpier Old Men (1995)"	"Comedy|Romance"
"4"	"Waiting to Exhale (1995)"	"Comedy|Drama|Romance"
"5"	"Father of the Bride Part II (1995)"	"Comedy"

Started streaming 5 records after 48 ms and completed after 53 ms.

```
// see the top 5 rows in the file
LOAD CSV WITH HEADERS FROM   "file:///MovieLens/ratings.csv" AS line
return line.userId,line.movieId,line.rating,line.timestamp
Limit 5
```

Screenshot of the output



$ LOAD CSV WITH HEADERS FROM "file:///MovieLens/ratings.csv" AS line return line.userId,line.movieId,line.ra…

line.userId	line.movieId	line.rating	line.timestamp
"1"	"31"	"2.5"	"1260759144"
"1"	"1029"	"3.0"	"1260759179"
"1"	"1061"	"3.0"	"1260759182"
"1"	"1129"	"2.0"	"1260759185"
"1"	"1172"	"4.0"	"1260759205"

Started streaming 5 records after 240 ms and completed after 241 ms.

```
// see the top 5 rows in the file
LOAD CSV WITH HEADERS FROM   "file:///MovieLens/tags.csv" AS line
return line.userId,line.movieId,line.tag,line.timestamp
Limit 5
```

Screenshot of the response

line.userId	line.movieId	line.tag	line.timestamp
"15"	"339"	"sandra 'boring' bullock"	"1138537770"
"15"	"1955"	"dentist"	"1193435061"
"15"	"7478"	"Cambodia"	"1170560997"
"15"	"32892"	"Russian"	"1170626366"
"15"	"34162"	"forgettable"	"1141391765"

To load the data into the nodes

1. Load movies.csv into Neo4j

```
LOAD CSV WITH HEADERS FROM   "file:///MovieLens/movies.csv" AS line
CREATE (l:movies {
                movieId : line.movieId,
                title :  line.title,
                genres :  line.genres
                })
```

Screenshot of the response

2. Create a constraint on movie.id

```
CREATE CONSTRAINT ON (n:movies) ASSERT n.movieId IS UNIQUE
```

Screenshot of the response

3. Load ratings.csv into Neo4j

```
LOAD CSV WITH HEADERS FROM   "file:///MovieLens/ratings.csv" AS line
CREATE (l:ratings {
                movieId : line.movieId,
                userId :  line.userId,
                rating :  line.rating
                })
```

Screenshot of the response

4. Load tags.csv

```
LOAD CSV WITH HEADERS FROM   "file:///MovieLens/tags.csv" AS line
CREATE (l:tags {
                    movieId : line.movieId,
                    userId :  line.userId,
                    tag :  split(line.tag,"," )
                    })
```

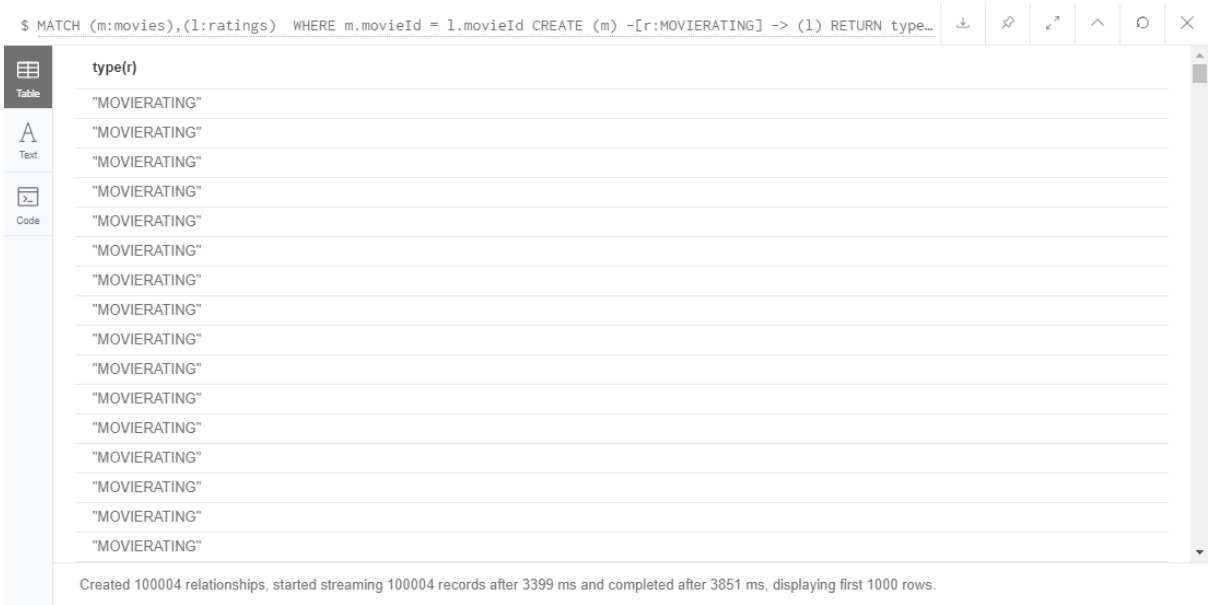
Screenshot of the response

Create Relationships Among the Nodes created

1. Relationship between movies and ratings

```
MATCH (m:movies),(l:ratings)
WHERE m.movieId = l.movieId
CREATE (m) -[r:MOVIERATING] -> (l)
RETURN type(r)
```

Screenshot of the response



2. Relationship between Movies and Tags

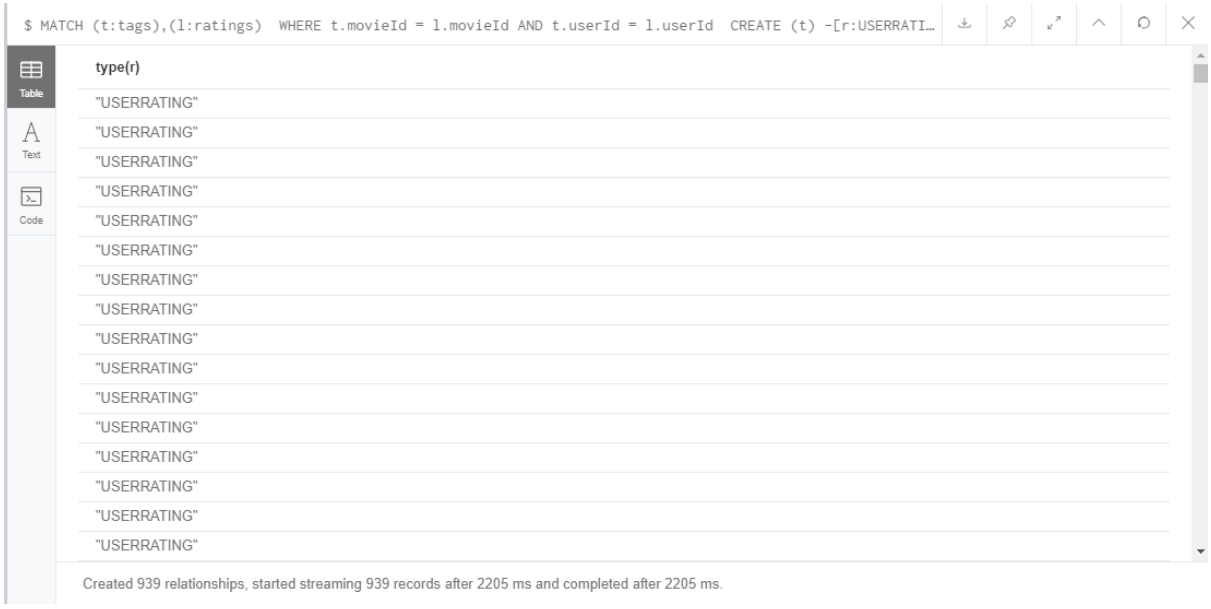
```
MATCH (m:movies),(l:tags)
WHERE m.movieId = l.movieId
CREATE (m) -[r:TAGGIVENBYUSER] -> (l)
RETURN type(r)
```

Screenshot of the response

3. Relationship between tags and rating

```
MATCH (t:tags),(l:ratings)
WHERE t.movieId = l.movieId AND t.userId = l.userId
CREATE (t) -[r:USERRATING] -> (l)
RETURN type(r)
```

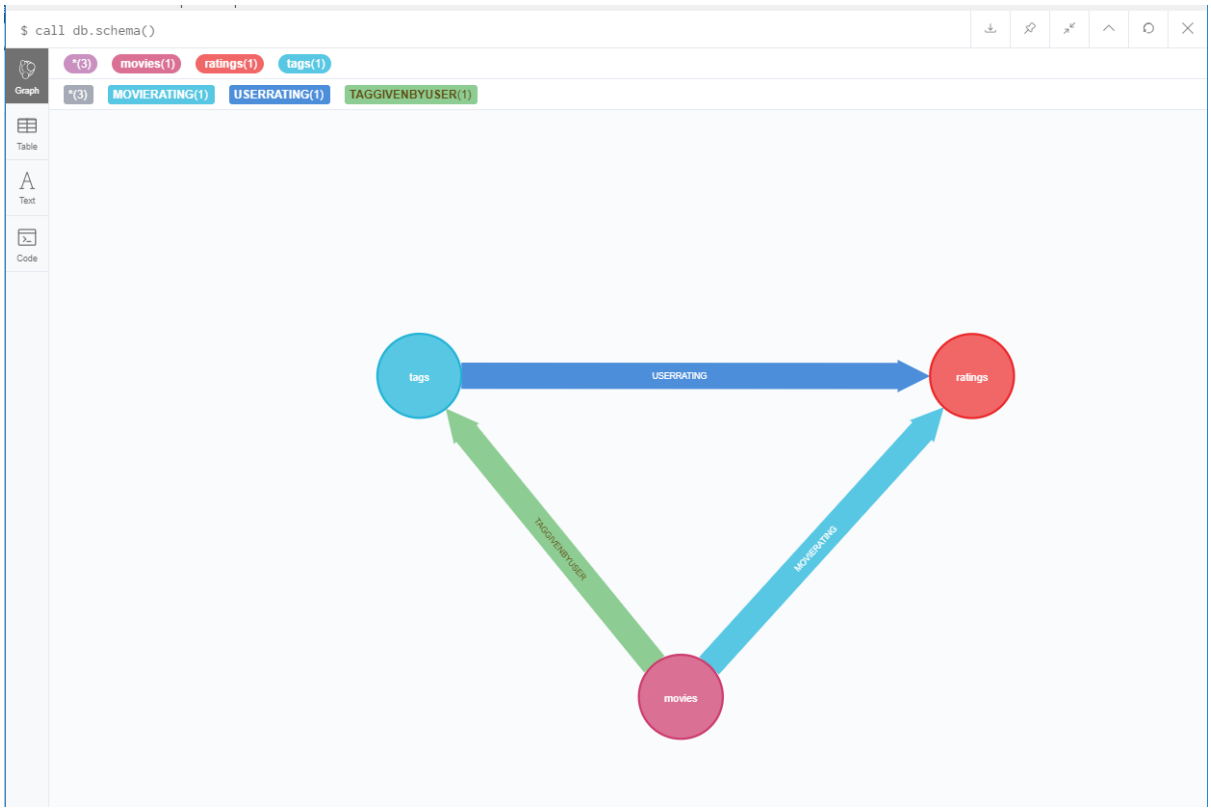
Screenshot of the response

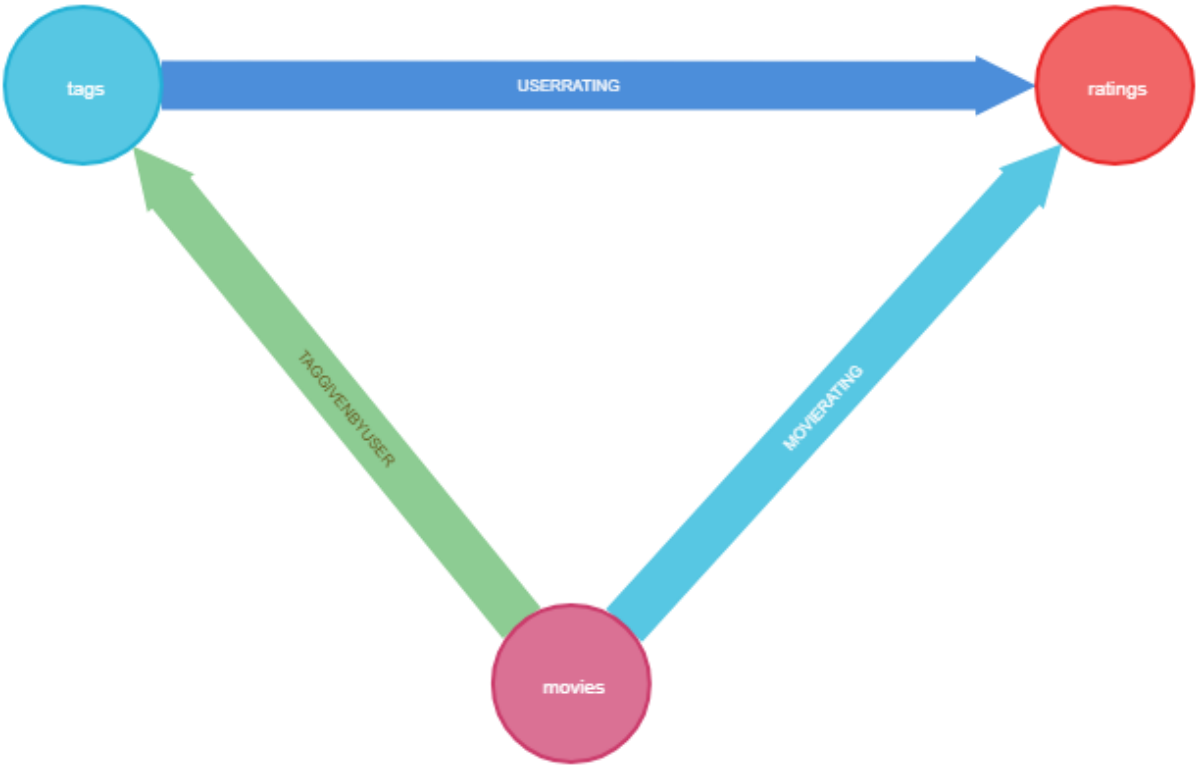


Viewing the Neo4j DB Schema

```
call db.schema()
```

Screenshot of the response



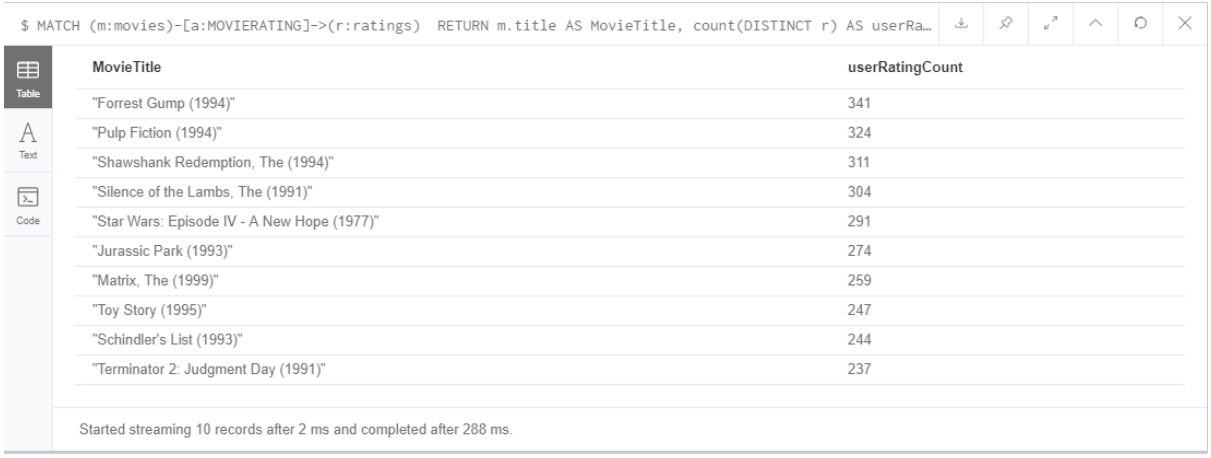

tags
USERRATING
ratings
TAGGIVENBYUSER
MOVIERATING
movies

Querying the Neo4j Database

1. Movie Title vs UserRatingCount

```
// Movie Title vs UserRatingCount
MATCH (m:movies)-[a:MOVIERATING]->(r:ratings)
RETURN m.title AS MovieTitle, count(DISTINCT r) AS userRatingCount
ORDER BY userRatingCount DESC LIMIT 10
```

Screenshot of the response



MovieTitle	userRatingCount
"Forrest Gump (1994)"	341
"Pulp Fiction (1994)"	324
"Shawshank Redemption, The (1994)"	311
"Silence of the Lambs, The (1991)"	304
"Star Wars: Episode IV - A New Hope (1977)"	291
"Jurassic Park (1993)"	274
"Matrix, The (1999)"	259
"Toy Story (1995)"	247
"Schindler's List (1993)"	244
"Terminator 2: Judgment Day (1991)"	237

Forrest Gump (1994) is the movie for which users have rated

2. Highest number of rating given to a movie and its average rating given by user

```
MATCH (m:movies)-[a:MOVIERATING]->(r:ratings)
RETURN m.title AS MovieTitle,count(DISTINCT r) AS countOfRating ,
avg(toFloat(r.rating)) AS userAvgRating
ORDER BY countOfRating  DESC LIMIT 10
```

Screenshot of the response

```
$ MATCH (m:movies)-[a:MOVIERATING]->(r:ratings)  RETURN m.title AS MovieTitle,count(DISTINCT r) AS countOf…
```

MovieTitle	countOfRating	userAvgRating
"Forrest Gump (1994)"	341	4.05425219941349
"Pulp Fiction (1994)"	324	4.25617283950617
"Shawshank Redemption, The (1994)"	311	4.487138263665601
"Silence of the Lambs, The (1991)"	304	4.1381578947368425
"Star Wars: Episode IV - A New Hope (1977)"	291	4.221649484536084
"Jurassic Park (1993)"	274	3.7062043795620427
"Matrix, The (1999)"	259	4.183397683397685
"Toy Story (1995)"	247	3.872469635627531
"Schindler's List (1993)"	244	4.303278688524592
"Terminator 2: Judgment Day (1991)"	237	4.00632911392405

Started streaming 10 records after 706 ms and completed after 706 ms.

3. Genres which has highest Average User Rating

```
MATCH (m:movies)-[a:MOVIERATING]->(r:ratings)
UNWIND split(m.genres,"|" ) AS x
RETURN DISTINCT(x) AS Genres, avg(toFloat(r.rating)) AS AverageUserRating
ORDER BY AverageUserRating  DESC
```

Screenshot of the response

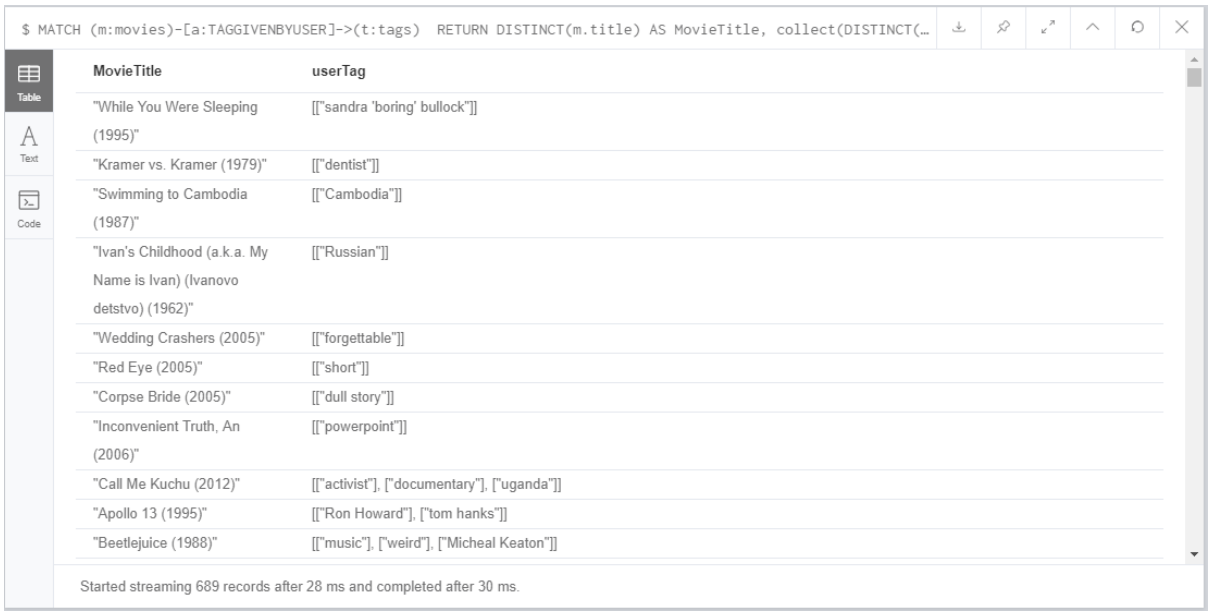
```
$ MATCH (m:movies)-[a:MOVIERATING]->(r:ratings) UNWIND split(m.genres,"|" ) AS x  RETURN DISTINCT(x) AS Genres, avg(toFloat(…
```

Genres	AverageUserRating
"Film-Noir"	3.9557017543859625
"War"	3.8172139303482537
"Documentary"	3.8132992327365733
"(no genres listed)"	3.7777777777777777
"Drama"	3.6817795852699287
"Crime"	3.6796385097749913
"Mystery"	3.6795409836065534
"Animation"	3.636061588330632
"Musical"	3.598792884371029
"IMAX"	3.571134347275033
"Western"	3.5664225941422587
"Romance"	3.5561646669424998
"Adventure"	3.5203933324249106
"Thriller"	3.518502377179081
"Fantasy"	3.5180632448156057
"Children"	3.4661866359447018
"Sci-Fi"	3.460429547673281
"Comedy"	3.4460369221058875
"Action"	3.4456128030751114
"Horror"	3.315243004418262

4. Movie Title and the tag given by the user to that movie

```
MATCH (m:movies)-[a:TAGGIVENBYUSER]->(t:tags)
RETURN DISTINCT(m.title) AS MovieTitle, collect(DISTINCT(t.tag)) AS userTag
```

Screenshot of the response



MovieTitle	userTag
"While You Were Sleeping (1995)"	[["sandra 'boring' bullock"]]
"Kramer vs. Kramer (1979)"	[["dentist"]]
"Swimming to Cambodia (1987)"	[["Cambodia"]]
"Ivan's Childhood (a.k.a. My Name is Ivan) (Ivanovo detstvo) (1962)"	[["Russian"]]
"Wedding Crashers (2005)"	[["forgettable"]]
"Red Eye (2005)"	[["short"]]
"Corpse Bride (2005)"	[["dull story"]]
"Inconvenient Truth, An (2006)"	[["powerpoint"]]
"Call Me Kuchu (2012)"	[["activist"], ["documentary"], ["uganda"]]
"Apollo 13 (1995)"	[["Ron Howard"], ["tom hanks"]]
"Beetlejuice (1988)"	[["music"], ["weird"], ["Micheal Keaton"]]

5. Tags and Average movie rating given by users to a movie

```
MATCH (t:tags) <-  [a:TAGGIVENBYUSER] - (m:movies) - [mr:MOVIERATING] -> (
r:ratings)
RETURN DISTINCT(m.title) AS MovieTitle,  collect(DISTINCT(t.tag)) AS userTag ,
avg(toFloat(r.rating)) AS userAvgRating
ORDER BY userAvgRating DESC LIMIT 10
```

Screenshot of the response

```
$ MATCH (t:tags) <- [a:TAGGIVENBYUSER] - (m:movies) - [mr:MOVIERATING] -> ( r:ratings) RETURN DISTINCT(m.t…
```

MovieTitle	userTag	userAvgRating
"Brooklyn (2015)"	[["tivo"], ["toplist15"]]	5.0
"Holy Motors (2012)"	[["dvd"], ["toplist12"]]	5.0
"Andrei Rublev (Andrey Rublyov) (1969)"	[["sightsound"]]	5.0
"Get Carter (1971)"	[["holes70s"]]	5.0
"Borgman (2013)"	[["Alex van Warmerdam"], ["magical realism"], ["weird"]]	5.0
"Au Hasard Balthazar (1966)"	[["sightsound"]]	5.0
"Journey, The (El viaje) (1992)"	[["beautiful scenery"], ["coming of age"], ["dark humor"], ["Fernando E. Solanas"], ["imaginative"], ["quirky"], ["whimsical"]]	5.0
"Sympathy for Mr. Vengeance (Boksuneun naui geot) (2002)"	[["stylized"], ["violent"]]	5.0
"The Beatles: Eight Days a Week - The Touring Years (2016)"	[["toplist16"]]	5.0
"Jesus of Montreal (Jésus de Montréal) (1989)"	[["jesus"]]	4.75

Started streaming 10 records after 212 ms and completed after 212 ms.

6. Movie Titles sorts with respect to the no of ratings given to it by the users and the average rating

```
MATCH (t:tags) <-  [a:TAGGIVENBYUSER] - (m:movies) - [mr:MOVIERATING] -> (
r:ratings)
RETURN DISTINCT(m.title) AS MovieTitle,  collect(DISTINCT (t.tag))  AS userTag ,
        count(DISTINCT r) AS countOfRating , avg(toFloat(r.rating)) AS userAvgRating
ORDER BY countOfRating DESC  LIMIT 10
```

Screenshot of the response

```
$ MATCH (t:tags) <- [a:TAGGIVENBYUSER] - (m:movies) - [mr:MOVIERATING] -> ( r:ratings) RETURN DISTINCT(m.t…
```

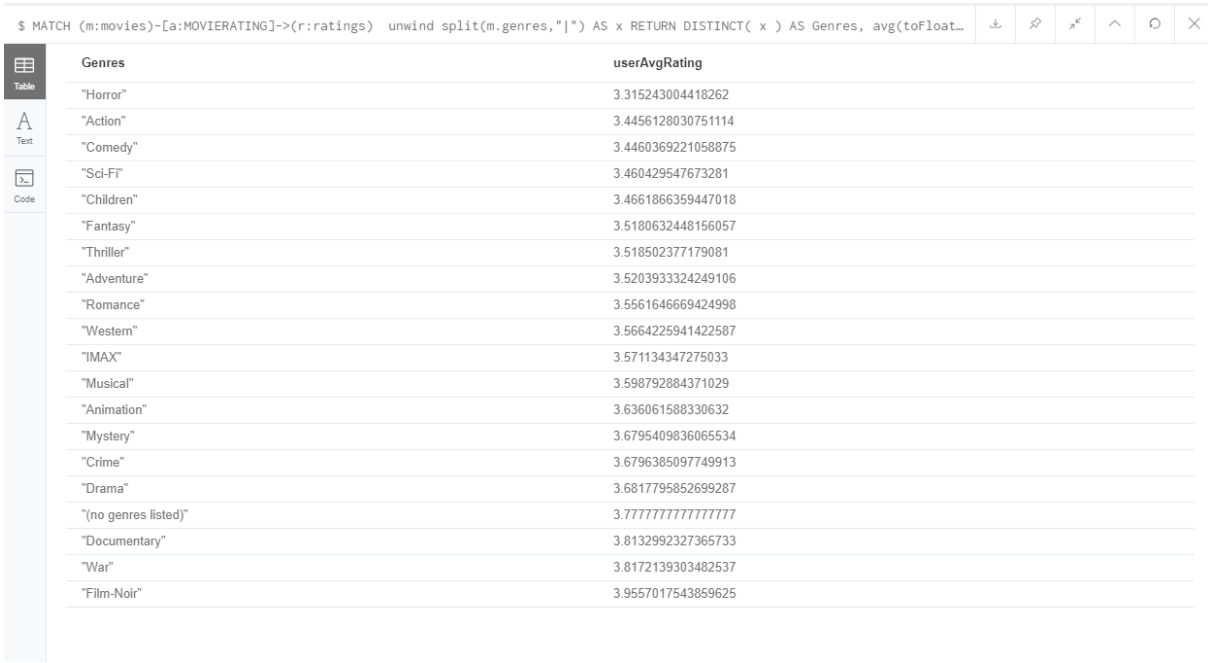
MovieTitle	userTag	countOfRating	userAvgRating
"Pulp Fiction (1994)"	[["intense"], ["r:violence"], ["tarantino"], ["dark comedy"], ["Quentin Tarantino"]]	324	4.25617283950617
"Shawshank Redemption, The (1994)"	[["friendship"], ["Morgan Freeman"], ["narrated"], ["prison"], ["prison escape"], ["revenge"], ["Tim Robbins"], ["wrongful imprisonment"], ["Phenomenal!"]]	311	4.487138263665602
"Silence of the Lambs, The (1991)"	[["Katso Sanna!"]]	304	4.1381578947368425
"Star Wars: Episode IV - A New Hope (1977)"	[["cult classic"], ["Science Fiction"], ["nerdy"], ["critically acclaimed"], ["awesome"], ["awesome soundtrack"], ["jedi"], ["space adventure"], ["coming of age"], ["space epic", " science fiction", " hero's journey"], ["classic"], ["sci-fi"], ["supernatural powers"], ["George Lucas"], ["starwars"], ["space"], ["classic sci-fi"], ["series"], ["imaginary world", " characters", " story", " philosophical"], ["script"], ["action"], ["Syfy"]]	291	4.221649484536096
"Matrix, The (1999)"	[["sci-fi"], ["virtual reality"], ["philosophy"]]	259	4.183397683397684
"Toy Story (1995)"	[["Pixar"]]	247	3.872469635627531

Started streaming 10 records after 240 ms and completed after 240 ms.

7. Average user rating per genres

```
MATCH (m:movies)-[a:MOVIERATING]->(r:ratings)
unwind split(m.genres,"|") AS x
RETURN DISTINCT( x ) AS Genres, avg(toFloat(r.rating)) AS userAvgRating
ORDER BY userAvgRating
```

Screenshot of the response



Genres	userAvgRating
"Horror"	3.315243004418262
"Action"	3.4456128030751114
"Comedy"	3.4460369221058875
"Sci-Fi"	3.460429547673281
"Children"	3.4661866359447018
"Fantasy"	3.5180632448156057
"Thriller"	3.518502377179081
"Adventure"	3.5203933324249106
"Romance"	3.5561646669424998
"Western"	3.5664225941422587
"IMAX"	3.571134347275033
"Musical"	3.598792884371029
"Animation"	3.636061588330632
"Mystery"	3.6795409836065534
"Crime"	3.6796385097749913
"Drama"	3.6817795852699287
"(no genres listed)"	3.7777777777777777
"Documentary"	3.8132992327365733
"War"	3.8172139303482537
"Film-Noir"	3.9557017543859625