

# Algo-Alchemists

## Recommender Systems

- Adithya (220587)
- Yash Giri (221218)

# Aspects of our project



## Target

Recommending a movie from the given dataset.  
Predicting the rating of unwatched movies.

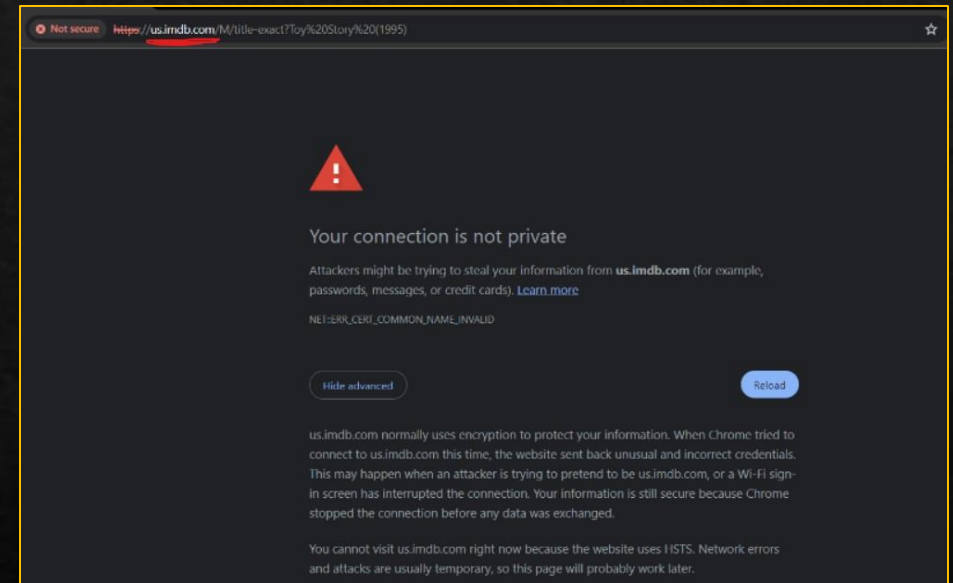


## Methods

Data Preprocessing  
Content Based Recommendation  
Collaborative Filtering

# Some Problems we faced...

- URLs' provided were not functioning.
- Sparsity of the Data.
- Movies were 20th Century classics.  
( Didn't watch many of those ☹️☹️)





# Data Pre-processing

## Steps:

1. Cleaning and merging the dataset.
2. **Creating Movie-Genre Matrix**

## OUTPUT :

[illegible]

# Recommender System



```
graph TD; A([Recommender System]) --> B[Content Based];
```

## Content Based

- Genre based.
- Based on average Rating.
- Recommending the 'Genre'

# Recommender System

```
graph TD; RS([Recommender System]) --> CB[Content Based]; RS --> CF[Collaborative filtering]; CB --> CB_Bul1[➤ Genre based.]; CB --> CB_Bul2[➤ Based on average Rating.]; CB --> CB_Bul3[➤ Recommending the 'Genre']; CF --> MB[Memory based]; CF --> MD[Model based]; MB --> MB_Bul1[➤ User based]; MB --> MB_Bul2[➤ Item based]; MD --> MD_Bul1[➤ K-NN];
```

Content Based

- Genre based.
- Based on average Rating.
- Recommending the 'Genre'

Collaborative filtering

Memory based

- User based
- Item based

Model based

- K-NN

# Recommender System

## Content Based

- Genre based.
- Based on average Rating.
- Recommending the 'Genre'.

## Collaborative filtering

### Memory based

- User based
- Item based

### Model based

- K-NN

## Our ideas

- Exploratory Data Analysis.
- Top Genre by profession.



# Target movie : The Lion King

## Binary Feature matrix

Applies straight forward comparison between movies and genres

```
98      Snow White and the Seven Dwarfs (1937)
102      All Dogs Go to Heaven 2 (1996)
94      Aladdin (1992)
101      Aristocats, The (1970)
90      Nightmare Before Christmas, The (1993)
141      Bedknobs and Broomsticks (1971)
131      Wizard of Oz, The (1939)
417      Cinderella (1950)
419      Alice in Wonderland (1951)
431      Fantasia (1940)
Name: movie title, dtype: object
```

## Bag Of Words

Creates a list of words from genres to apply similarity.

```
movie_title
Dumbo (1941)      1.0
Cinderella (1950) 1.0
Lion King, The (1994) 1.0
James and the Giant Peach (1996) 1.0
Snow White and the Seven Dwarfs (1937) 1.0
Three Caballeros, The (1945) 1.0
Cats Don't Dance (1997) 1.0
All Dogs Go to Heaven 2 (1996) 1.0
Beauty and the Beast (1991) 1.0
Hunchback of Notre Dame, The (1996) 1.0
Name: Lion King, The (1994), dtype: float64
```

## Tf - Idf technique

Assigns weights to important and frequently occurred terms

```
movie_title
James and the Giant Peach (1996)      1.0
Cats Don't Dance (1997)                1.0
Lion King, The (1994)                  1.0
Hunchback of Notre Dame, The (1996)    1.0
Beauty and the Beast (1991)            1.0
Pete's Dragon (1977)                  1.0
Dumbo (1941)                          1.0
Anastasia (1997)                      1.0
Alice in Wonderland (1951)             1.0
Fantasia (1940)                       1.0
Name: Lion King, The (1994), dtype: float64
```





**MOVIES**  
**The Little Mermaid**  
1989

In Disney's enchanting animated design, a mermaid "falls in love" with a human prince.



**MOVIES**  
**Tarzan**  
1999

This movie is about the life of Tarzan, a young man who grows up in the jungle.



**MOVIES**  
**Up**  
2009

An old man, Carl Fredricksen, decides to explore the world and escape his lonely life.



**MOVIES**  
**Snow White and the Seven Dwarfs**  
1937

The beautiful and benevolent Snow White is the only woman who can save the kingdom from the evil witch.



**MOVIES**  
**The Wizard of Oz**  
1939

When a tornado hits through Kansas, Dorothy Gale and her dog, Toto, are whisked away to the land of Oz.



**MOVIES**  
**The Prince of Egypt**  
1998

This is the extraordinary tale of two brothers, Moses and Pharaoh, who are destined to lead the people of Israel.



**MOVIES**  
**The Lion King II: Simba's Pride**  
1998

Simba and Nala have a daughter, Kiara, who is destined to lead the people of the Pride Lands.



**MOVIES**  
**The Hunchback of Notre Dame**  
1996

In 15th-century Paris, Clopin, the master of the gypsies, leads the people of the city to the cathedral of Notre Dame.



**MOVIES**  
**Toy Story**  
1995

A little boy named Andy loves to play with his toys, especially his old-fashioned "Woody".



**MOVIES**  
**Aladdin**  
1992

Aladdin is a poor, street-smart boy who is destined to marry the beautiful Princess Jasmine.



**MOVIES**  
**Beauty and the Beast**  
1991

When a young girl is kidnapped by a monster, she must learn to love him to save the world.



**MOVIES**  
**Mulan**  
1998

This is the story of the Chinese heroine who saves her father and her country.



**MOVIES**  
**Bambi**  
1942

It's spring, and all the animals of the forest are celebrating the birth of Bambi.



**MOVIES**  
**Tangled**  
2010

After escaping the evil queen, Rapunzel is kidnapped by the prince and must learn to love him.



**MOVIES**  
**Hercules**  
1997

Hercules, son of the Greek God, Zeus, is a young man who is destined to become a hero.



**MOVIES**  
**Pocahontas**  
1995

This is the story of the Native American woman who saves the English explorer.



**MOVIES**  
**Finding Nemo**  
2003

A clown fish named Nemo goes to the Great Barrier Reef and is kidnapped by a shark.



**MOVIES**  
**Monsters, Inc.**  
2001

A city of monsters with no humans, Monsters, Inc. is a company that collects screams.



**MOVIES**  
**Toy Story 3**  
2010

Woody, Buzz, and the other toys are about to be donated to a museum.



**MOVIES**  
**How to Train Your Dragon**  
2010

Long ago, the Vikings and the dragons of Berk, the young Viking, Hiccup, learn to live together.



**MOVIES**  
**Ratatouille**  
2007

A rat named Remy dreams of becoming a great chef, but his family and the kitchen staff don't want him.



**MOVIES**  
**The Jungle Book**  
1967

Mowgli, son of the British man, is a young boy who is destined to become a hero.



**MOVIES**  
**WALL-E**  
2008

In a distant, but not too far, future, WALL-E is a small robot who is destined to become a hero.



**MOVIES**  
**Brother Bear**  
2003

When a young boy is kidnapped by a bear, he must learn to love him.



**MOVIES**  
**The Emperor's New Groove**  
2006

A little boy named Kuzco is kidnapped by a witch and must learn to love her.



**MOVIES**  
**Oliver & Company**  
1988

Supporting Charles Dickens' "Oliver Twist", a homeless boy named Oliver is kidnapped by a gang of thieves.



**MOVIES**  
**Dumbo**  
1941

This story tells of a young elephant who is born with large ears and must learn to fly.



**MOVIES**  
**The Many Adventures of...**  
1977

Winnie the Pooh, a bear who lives in the Hundred Acre Wood, is a very happy bear.



**MOVIES**  
**The Fox and the Hound**  
1981

When a fox and a hound are born, they are destined to become friends.



**MOVIES**  
**The Incredibles**  
2004

Mr. Incredible, a superhero, is a man who is destined to become a hero.



**MOVIES**  
**Robin Hood**  
1953

An imaginative Disney version of the Robin Hood legend, this movie is a very happy story.



**MOVIES**  
**Alice in Wonderland**  
1951

Alice is a young girl who is kidnapped by a witch and must learn to love her.



**MOVIES**  
**Aristocats**  
1970

A group of cats, including a young girl, are kidnapped by a witch and must learn to love her.



**MOVIES**  
**The Tigger Movie**  
2000

Tigger is a young tiger who is kidnapped by a witch and must learn to love her.



**MOVIES**  
**The Great Mouse Detective**  
1986

When a mouse is kidnapped by a witch, he must learn to love her.



**MOVIES**  
**The Prince of the Amazon**  
1999

A young boy is kidnapped by a witch and must learn to love her.



**MOVIES**  
**Sleeping Beauty**  
1959

A young girl is kidnapped by a witch and must learn to love her.



**MOVIES**  
**The Princess and the Frog**  
2009

A young girl is kidnapped by a witch and must learn to love her.



**MOVIES**  
**Toy Story 2**  
1999

A little boy named Andy loves to play with his toys, especially his old-fashioned "Woody".



**MOVIES**  
**Inside Out**  
2015

A young girl is kidnapped by a witch and must learn to love her.



- Simply recommending based on high average rating of the movie given by users.
- Suggesting Genres to the users based on past activity.

```
suggest(85)
```

✓ 0.0s

Python

```
['Comedy',  
'Romance',  
'Drama War',  
'Child',  
'Child Comedy',  
'Comedy Musical Romance',  
'Crime Drama Mystery',  
'Drama',  
'Drama Musical',  
'Drama Romance War']
```

## ➤ User Based Collaborative filtering

- ❑ Created a normalized matrix to apply *Pearson Correlation*.
- ❑ Dropped movies that are already watched by user.
- ❑ Recommended movie to the user based on similarity score as *weight* of the user.
- ❑ Predicting the rating by the user.

	movie	movie_score	new_rating
155	875	1.952381	4.826667
176	955	1.952381	4.826667
115	511	1.773333	4.647619
19	133	1.773333	4.647619
129	607	1.773333	4.647619
18	132	1.773333	4.647619
117	513	1.773333	4.647619
9	56	1.773333	4.647619
25	187	1.773333	4.647619
112	475	1.476190	4.350476

# ➤ Item based Collaborative filtering

- ❑ Sorting the dataset by movies that are NOT watched by the user.
- ❑ Getting a *similarity score* between target movie and watched movies.
- ❑ Predicts the rating by calculating *weighted mean* of ratings of similar watched movies

the predicted rating for movie id 20 by user 90 is 0.54067

90		similarity_scores
movie_id		
1192	1.615385	1.000000
836	1.230769	1.000000
821	-0.045455	1.000000
19	-0.956522	1.000000
889	-0.384615	0.878310
1097	0.500000	0.852803
903	0.888889	0.774597
18	0.200000	0.765532
1137	-1.965517	0.755929
632	1.103448	0.740593



## ➤ Model Based filtering using KNN

- ❑ Takes input movie name/id and gets its ratings vector to use as input.
- ❑ Defined a *movie\_engine* function which sorts the nearest neighbours by their distances from input movie,

```
#getting recommendations  
no_recommen = 10  
movie_engine('Lion King, The (1994)', matrix,no_recommen)  
✓ 0.0s
```

	Title	Distance
9	Maverick (1994)	0.633655
8	Young Guns (1988)	0.648028
7	Billy Madison (1995)	0.661017
6	Promesse, La (1996)	0.661710
5	Jack (1996)	0.662719
4	Adventures of Priscilla, Queen of the Desert, ...	0.669608
3	Career Girls (1997)	0.672660
2	Nikita (La Femme Nikita) (1990)	0.672883
1	Mary Reilly (1996)	0.673278

# OUR IDEAS

- Did some exploratory data Analysis on the given data set from *movielens*.
  - **Getting Genres by profession.**
  - **Average ratings by profession.**
  - **Number of ratings by age group.**
  - **Number of ratings by profession.**
  - **Number of ratings by Gender.**

# ❖ Average ratings given by the user to a Genre of a movie.

genres	Animation	Animation Child	Animation Child Comedy	Animation Child Comedy Fantasy	Animation Child Comedy Musical	Animation Child Comedy Romance	Animation Child Fantasy	Animation Child Musical	Animation Child Musical Romance	...	Romance Thriller	Romance War	Sci- Fi	Sci-Fi Thriller	Sci-Fi Thriller War	Sci- Fi War	Thriller	Thriller War	
user																			
1	2.625	5.0	2.0	5.0	0.0	4.0	0.0	0.0	2.333333	0.0	...	4.0	4.0	4.100	3.142857	5.0	3.0	3.538462	0.0
2	0.000	0.0	0.0	4.0	0.0	0.0	0.0	0.0	0.000000	0.0	...	0.0	0.0	0.000	3.000000	0.0	0.0	3.666667	0.0
3	0.000	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.000000	0.0	...	0.0	0.0	0.000	2.500000	0.0	3.0	2.750000	0.0
4	3.000	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.000000	0.0	...	0.0	0.0	0.000	2.500000	0.0	4.0	5.000000	0.0
5	3.500	0.0	2.5	4.0	0.0	4.0	0.0	0.0	3.250000	0.0	...	0.0	1.0	3.250	1.000000	3.0	4.0	3.000000	0.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
939	2.000	0.0	0.0	0.0	0.0	4.0	0.0	0.0	0.000000	0.0	...	0.0	0.0	5.000	3.333333	0.0	5.0	4.400000	0.0
940	4.000	0.0	0.0	0.0	0.0	5.0	0.0	0.0	4.000000	0.0	...	3.0	0.0	4.000	2.000000	4.0	2.5	4.500000	3.0
941	4.000	0.0	0.0	5.0	0.0	4.0	0.0	0.0	0.000000	0.0	...	0.0	0.0	3.500	3.500000	0.0	0.0	4.333333	0.0
942	4.500	0.0	4.0	0.0	0.0	5.0	0.0	0.0	5.000000	0.0	...	5.0	5.0	0.000	0.000000	0.0	0.0	4.000000	4.0
943	4.000	0.0	3.0	0.0	0.0	0.0	0.0	0.0	0.000000	0.0	...	3.5	0.0	1.875	3.200000	0.0	3.0	4.000000	0.0
943 rows × 143 columns																			

943 rows x 143 columns

```
fin = pd.merge(user_top_5_list,used,on='user')# used is the data frame containing user and profession as columns
✓ 0.0s
```

```
top_genres_dict = {}
for index, row in fin.iterrows():

    profession = row['profession']
    genres = row['genre_5']

    genre_counts = pd.Series(genres).value_counts()

    top_genres = genre_counts.index[:3].tolist()
    top_genres_dict[profession] = top_genres
result_df = pd.DataFrame(list(top_genres_dict.items()), columns=['profession', 'top_5_genres'])
```

- ❖ From Genre recommendation data, we got the *top\_5* genre liked by the user.
- ❖ Mapped the users with their profession and got the overall most liked genres by that profession.

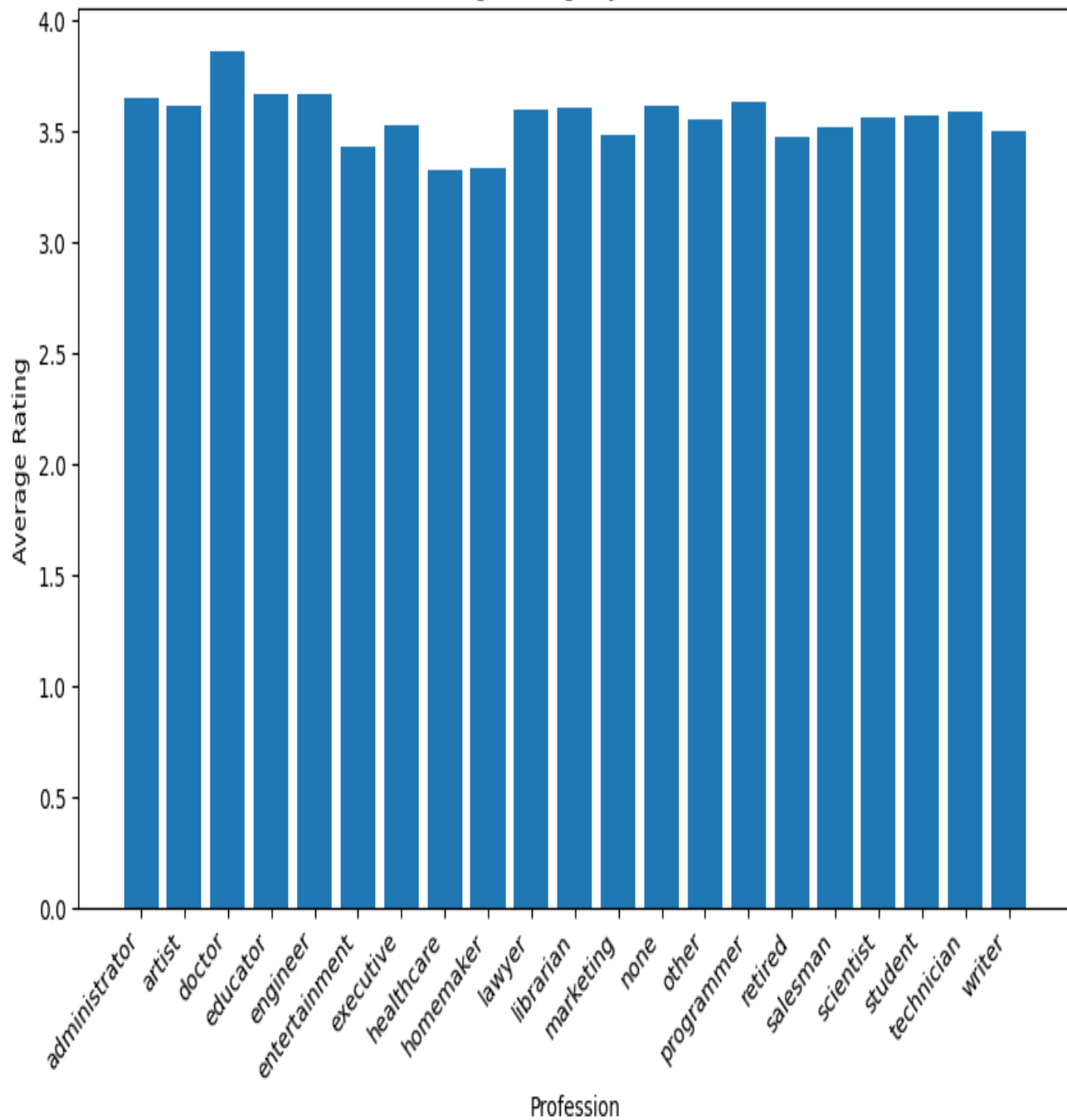
user		genre_5
0	1	[Horror Sci-Fi Thriller, Drama Romance Sci-Fi ...
1	2	[Crime Film-Noir Mystery Thriller, Comedy Dram...
2	3	[Mystery Romance Thriller, Documentary, Crime ...
3	4	[Crime Drama Thriller, Mystery, Crime Drama My...
4	5	[Animation Comedy, Animation Comedy Thriller, ...
...	...	...
938	939	[Drama War, Sci-Fi War, Sci-Fi, Comedy Romance...
939	940	[Animation Child Comedy Musical, Comedy Crime,...
940	941	[Animation Comedy Thriller, Animation Child Co...
941	942	[Drama War, Drama Romance Sci-Fi War, Drama Th...
942	943	[Crime Drama Thriller, Comedy Horror Sci-Fi, C...



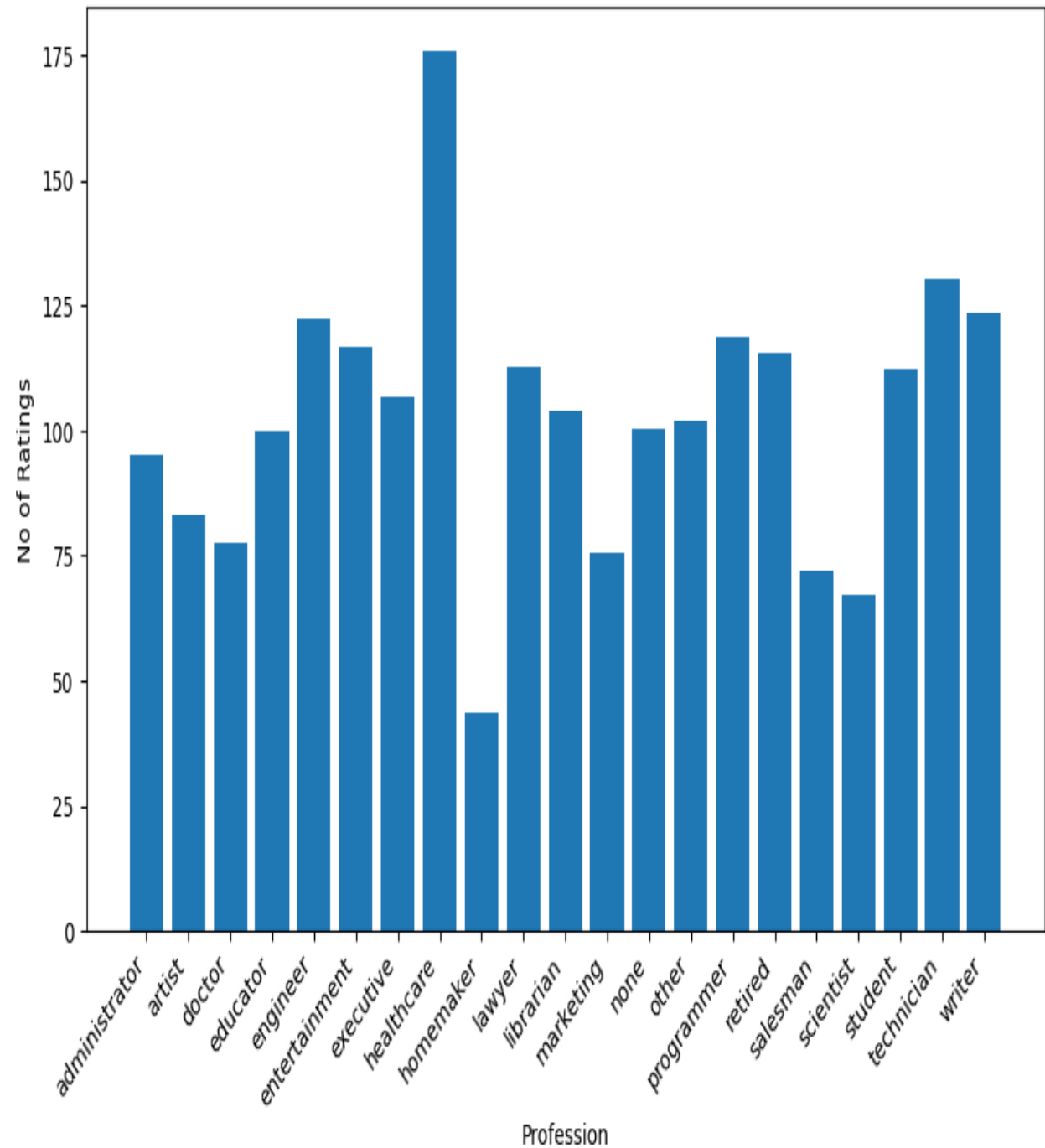
```
for index, row in result_df.iterrows():  
    print(f"{row['profession']}: {row['top_5_genres']}")
```

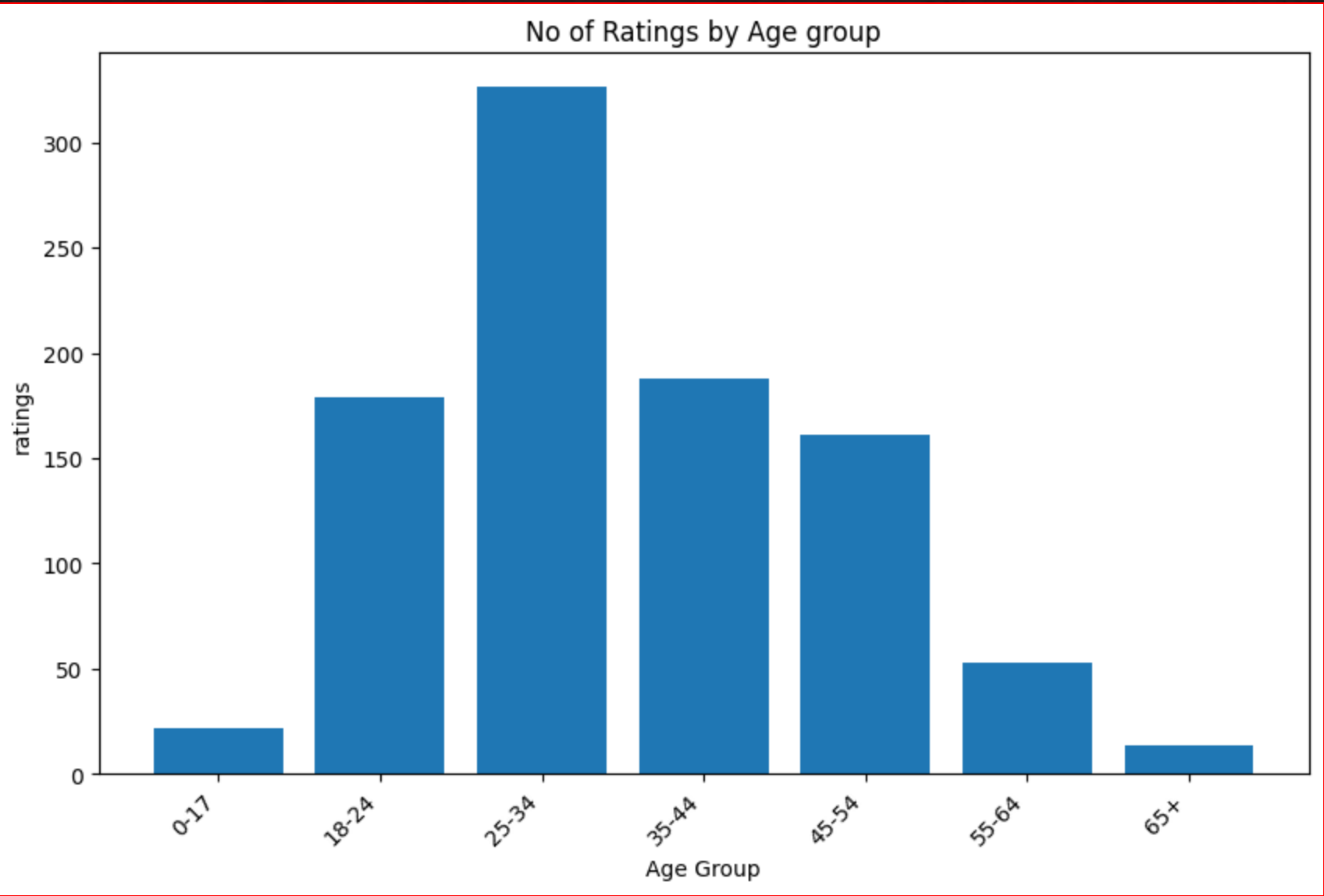
```
technician: ['Drama Horror', 'Romance Sci-Fi War', 'Animation Child Comedy Musical', 'Sci-Fi War', 'Crime Drama Thriller']  
other: ['Drama Mystery Romance', 'Drama Musical', 'Sci-Fi', 'Drama Romance War', 'Comedy Drama']  
writer: ['Drama Mystery', 'Comedy Musical', 'Child Comedy Musical', 'Sci-Fi War', 'Comedy Western']  
executive: ['Romance', 'Musical Romance', 'Comedy Western', 'Comedy Crime', 'Crime Thriller']  
administrator: ['Animation Child Comedy Musical', 'Comedy Crime', 'Child Comedy Drama', 'Drama Mystery Romance Thriller', 'Crime Drama']  
student: ['Crime Drama Thriller', 'Comedy Horror Sci-Fi', 'Comedy Musical', 'Comedy Romance War', 'Crime Drama Romance Thriller']  
lawyer: ['Child Drama Musical', 'Film-Noir Mystery Thriller', 'Film-Noir Sci-Fi', 'Comedy Crime Horror', 'Documentary']  
educator: ['Animation Comedy Thriller', 'Romance Sci-Fi War', 'Drama Mystery', 'Crime Drama', 'Documentary Drama']  
scientist: ['Drama Thriller', 'Sci-Fi War', 'Drama', 'Drama Romance', 'Thriller']  
entertainment: ['Drama Sci-Fi', 'Crime Film-Noir Mystery Thriller', 'Drama Romance War', 'Comedy', 'Drama']  
programmer: ['Drama', 'Sci-Fi War', 'Crime Thriller', 'Animation Child Comedy Musical', 'Horror Thriller']  
librarian: ['Drama War', 'Drama Romance Sci-Fi War', 'Drama Thriller War', 'Drama Western', 'Film-Noir Mystery']  
homemaker: ['Crime', 'Comedy Musical Romance', 'Crime Drama Mystery', 'Drama Thriller', 'Drama Sci-Fi Thriller']  
artist: ['Romance', 'Comedy Drama', 'Drama Romance', 'Crime Drama', 'Crime Film-Noir Mystery Thriller']  
engineer: ['Comedy Horror', 'Film-Noir Sci-Fi', 'Comedy Romance War', 'Drama Mystery', 'Drama Romance Sci-Fi War']  
marketing: ['Child Comedy Musical', 'War', 'Horror Romance Thriller', 'Drama Mystery', 'Musical']  
none: ['Musical', 'Crime Drama Romance Thriller', 'Crime Drama Thriller', 'Mystery Thriller', 'Drama Musical']  
healthcare: ['', 'Comedy Drama', 'Horror Sci-Fi Thriller', 'Drama War', 'Mystery']  
retired: ['Western', 'Crime Drama Thriller', 'Comedy Thriller', 'Drama Mystery', 'Drama Romance War']  
salesman: ['Crime Drama Thriller', 'Horror Mystery Thriller', 'Horror Romance Thriller', 'Horror Thriller', 'Mystery']  
doctor: ['Drama Romance War', 'Drama Thriller', 'Comedy', 'Drama Romance', 'Thriller']
```

Average Ratings by Profession



No of Ratings by Profession





	gender	average_rating
0	F	3.587179
1	M	3.588604

	gender	ratings
0	F	273
1	M	670

# Pizza powered Prowess :)

- First and foremost, we are glad the data was of **movies!** Didn't lose interest at any point of project timeline.
- Having watched less movies from the dataset, it was challenging to compare the correctness of output with respect to the input, unlike Topic-based modelling.
- High fives to **Kritnandan's team** (MK) for collaborating with us and helping us out.
- Lastly, We wonder how true **Tarnoff's Axiom** is:

Monkey with a Typewriter + Infinite time = Works of Shakespeare

Humans with a computer + Infinite time = PowerPoint Slides



**THANK YOU !**