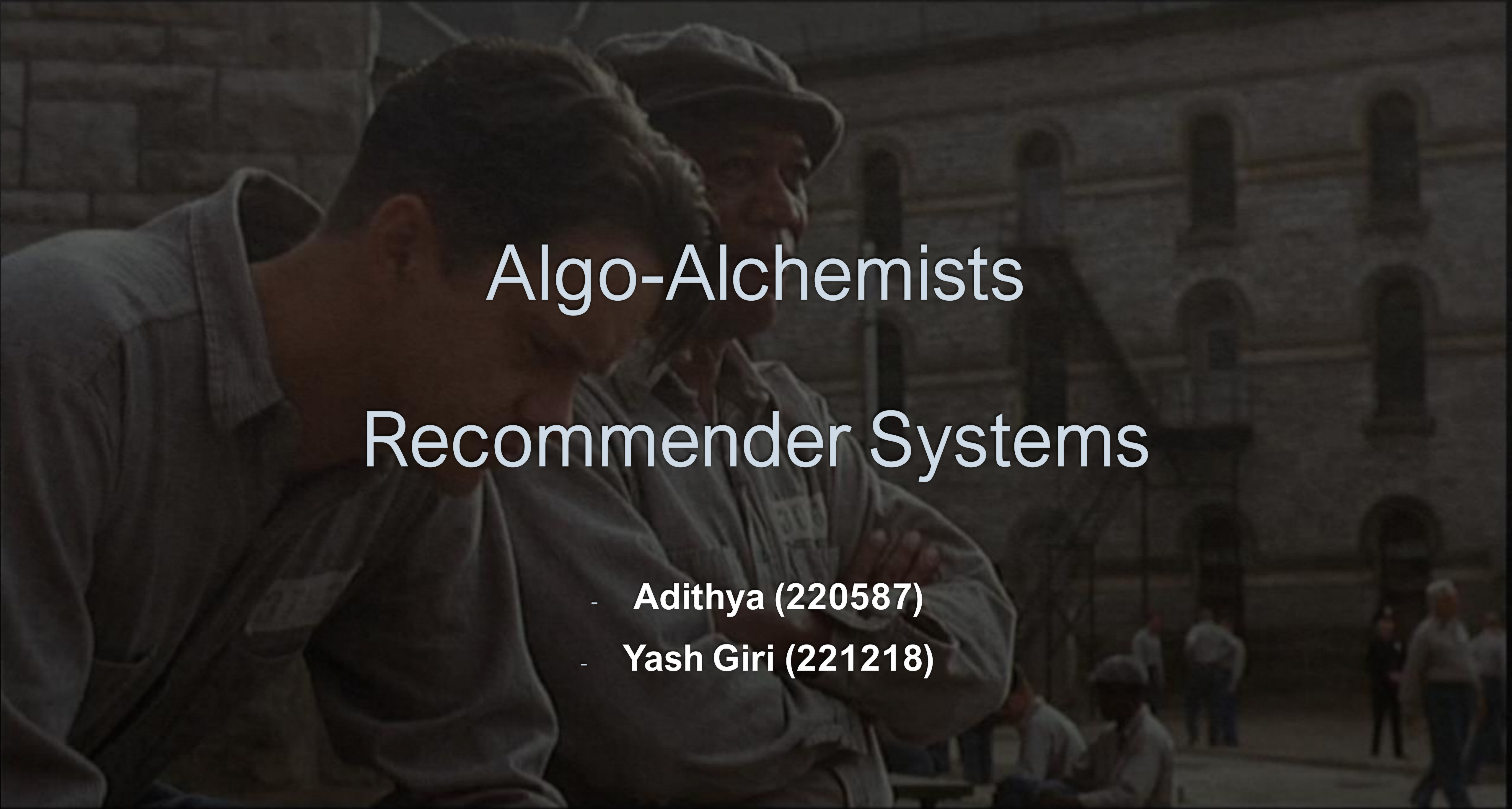


Algo-Alchemists

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

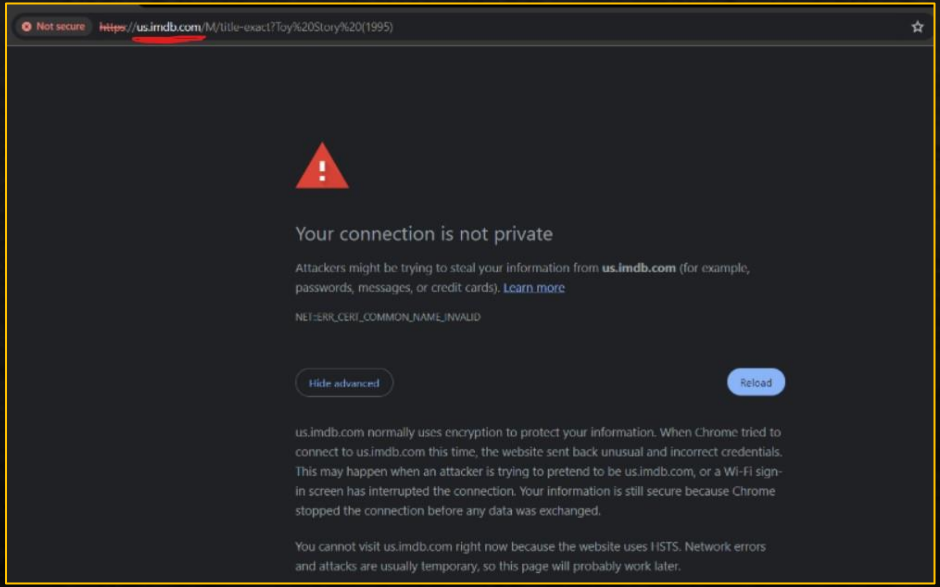
Data Preprocessing

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

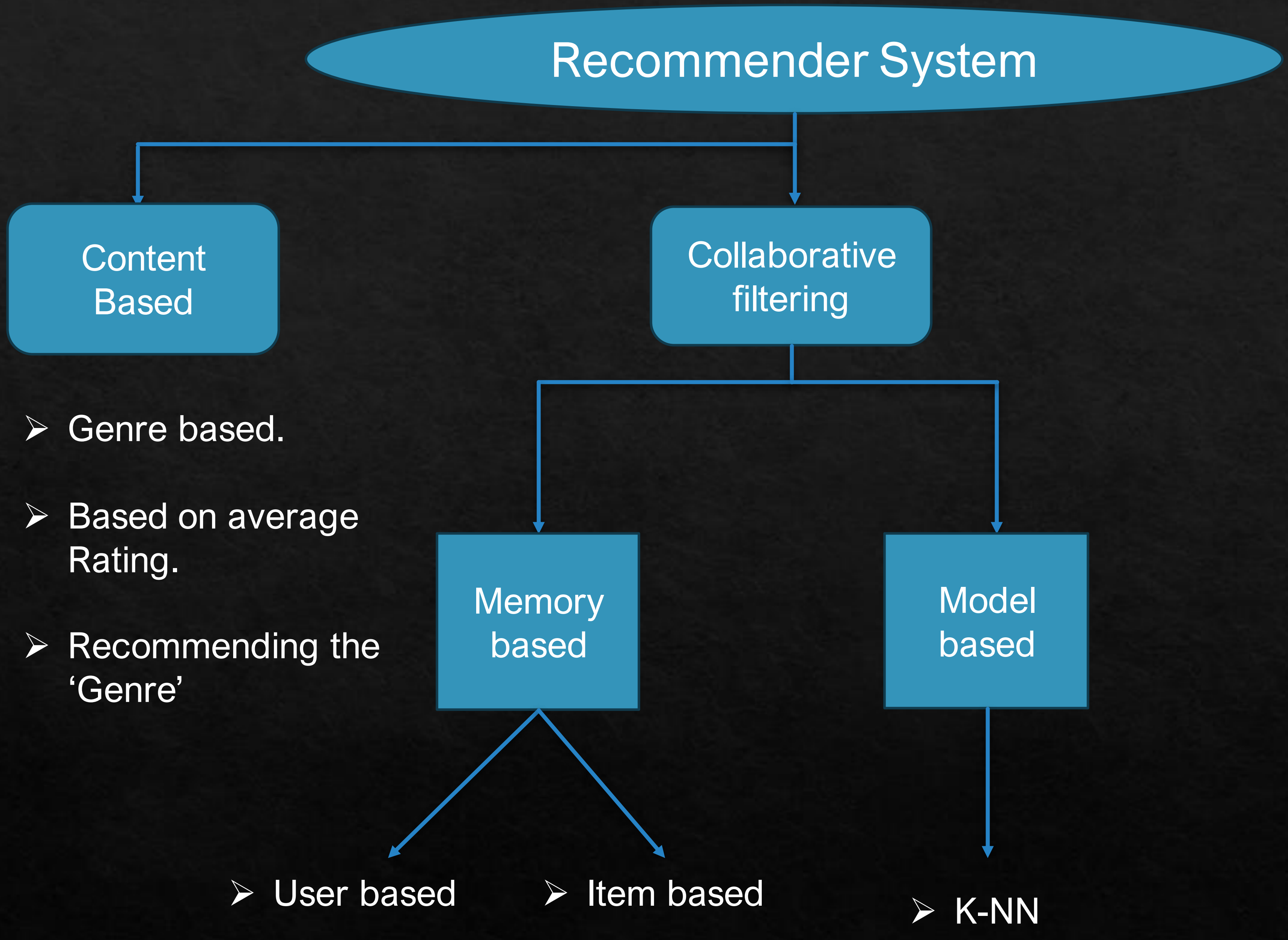
- **Steps:**

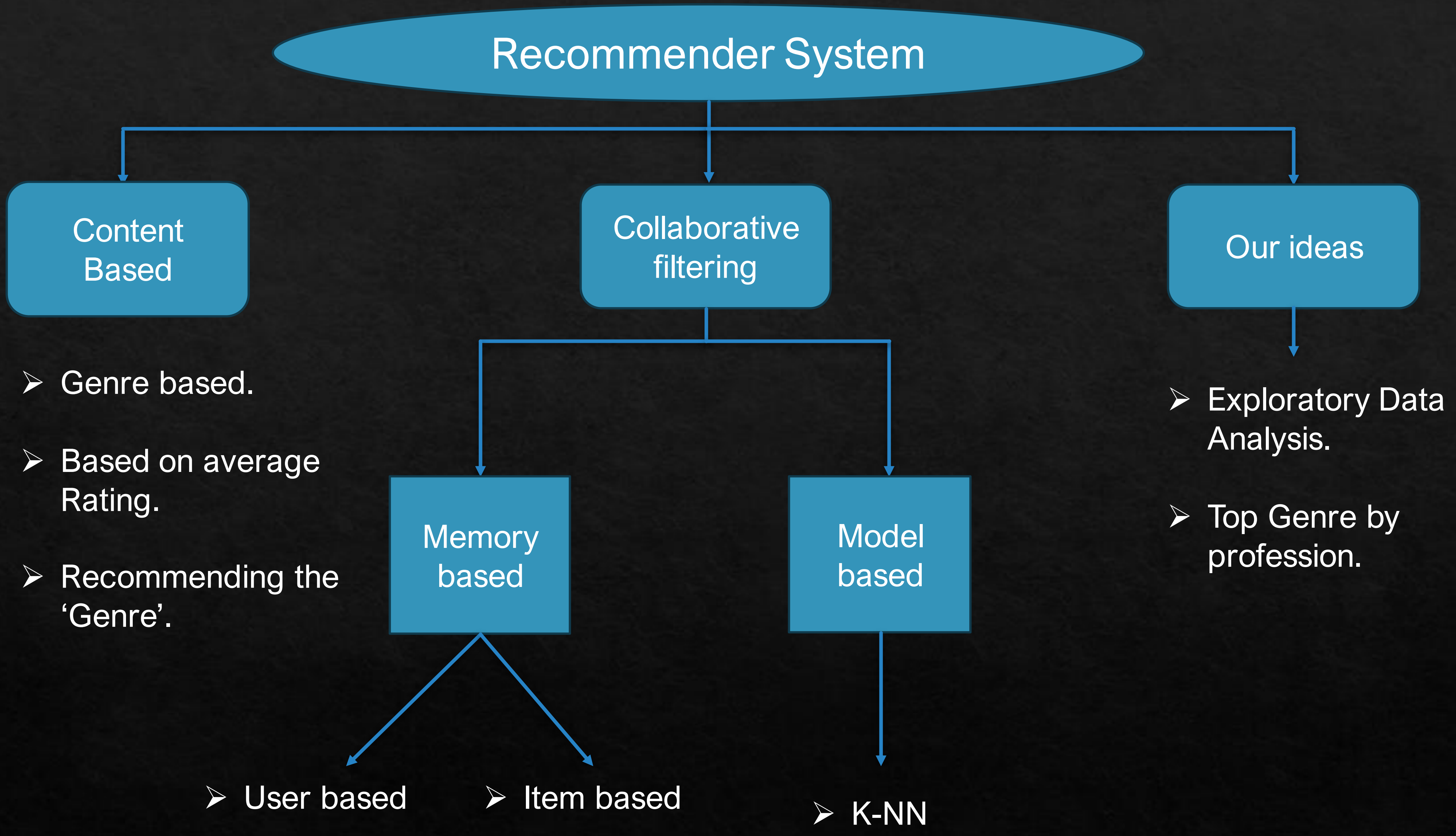
1. Cleaning and merging the dataset.
2. **Creating Movie-Genre Matrix**

OUTPUT :

movie_id	movie_title	rel_date	URL	unknown	Action	Adventure	Animation	Child	Comedy	...	Fantasy	Film-Noir	Horror	Musical	Mystery	Romance	Sci-Fi	Thriller	War	We
1	Toy Story (1995)	01-Jan-1995	http://us.imdb.com/M/title-exact?Toy%20Story%2...	0	0	0	1	1	1	...	0	0	0	0	0	0	0	0	0	
2	GoldenEye (1995)	01-Jan-1995	http://us.imdb.com/M/title-exact?GoldenEye%20(...	0	1	1	0	0	0	...	0	0	0	0	0	0	0	1	0	
3	Four Rooms (1995)	01-Jan-1995	http://us.imdb.com/M/title-exact?Four%20Rooms%...	0	0	0	0	0	0	...	0	0	0	0	0	0	0	1	0	
4	Get Shorty (1995)	01-Jan-1995	http://us.imdb.com/M/title-exact?Get%20Shorty%...	0	1	0	0	0	1	...	0	0	0	0	0	0	0	0	0	
5	Copycat (1995)	01-Jan-1995	http://us.imdb.com/M/title-exact?Copycat%20(1995)	0	0	0	0	0	0	...	0	0	0	0	0	0	0	1	0	

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


Recommender System
Content Based
Collaborative filtering
Genre based.
Based on average Rating.
Recommending the 'Genre'
Memory based
Model based
User based
Item based
K-NN


Recommender System
Content Based
Collaborative filtering
Our ideas
Genre based.
Based on average Rating.
Recommending the 'Genre'.
Memory based
Model based
User based
Item based
K-NN
Exploratory Data Analysis.
Top Genre by profession.

Target movie : The Lion King

Binary Feature matrix

Applies straight forward comparison between movies and genres

Bag Of Words

Creates a list of words from genres to apply similarity.

Tf - Idf technique

Assigns weights to important and frequently occurred terms

```
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
```

```
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
```

```
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

MOVIES
Tarzan

MOVIES
Up

MOVIES
Snow White and the Seven Dwarfs

MOVIES
The Wizard of Oz

MOVIES
The Prince of Egypt

MOVIES
The Lion King II: Simba's Pride

MOVIES
The Hunchback of Notre Dame

MOVIES
Toy Story

MOVIES
Aladdin

MOVIES
Beauty and the Beast

MOVIES
Mulan

MOVIES
Bambi

MOVIES
Tangled

MOVIES
Hercules

MOVIES
Pocahontas

MOVIES
Finding Nemo

MOVIES
Monsters, Inc.

MOVIES
Toy Story 3

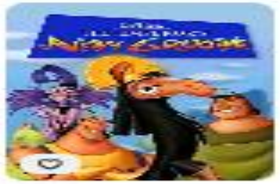
MOVIES
How to Train Your Dragon

MOVIES
Ratatouille

MOVIES
The Jungle Book

MOVIES
WALL-E

MOVIES
Brother Bear



MOVIES
The Emperor's New Groove

MOVIES
Oliver & Company

MOVIES
Dumbo

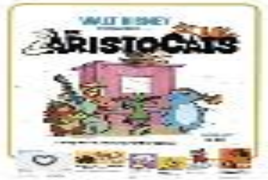
MOVIES
The Many Adventures of...

MOVIES
The Fox and the Hound

MOVIES
The Incredibles

MOVIES
Robin Hood

MOVIES
Alice in Wonderland



- 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

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

✓ 0.0s

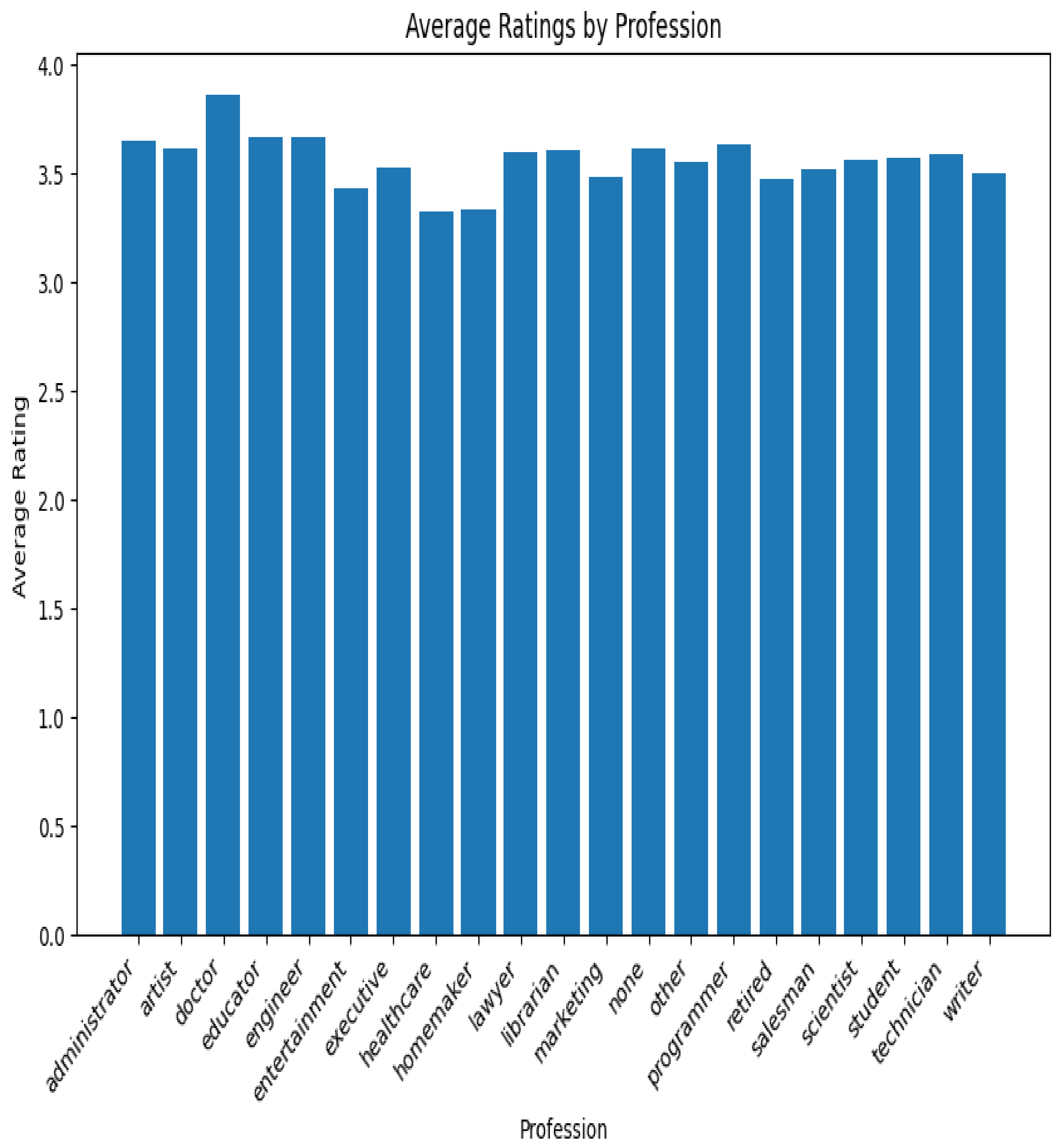
```python
top_genres_dict = {}
for index, row in fin.iterrows():

    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.

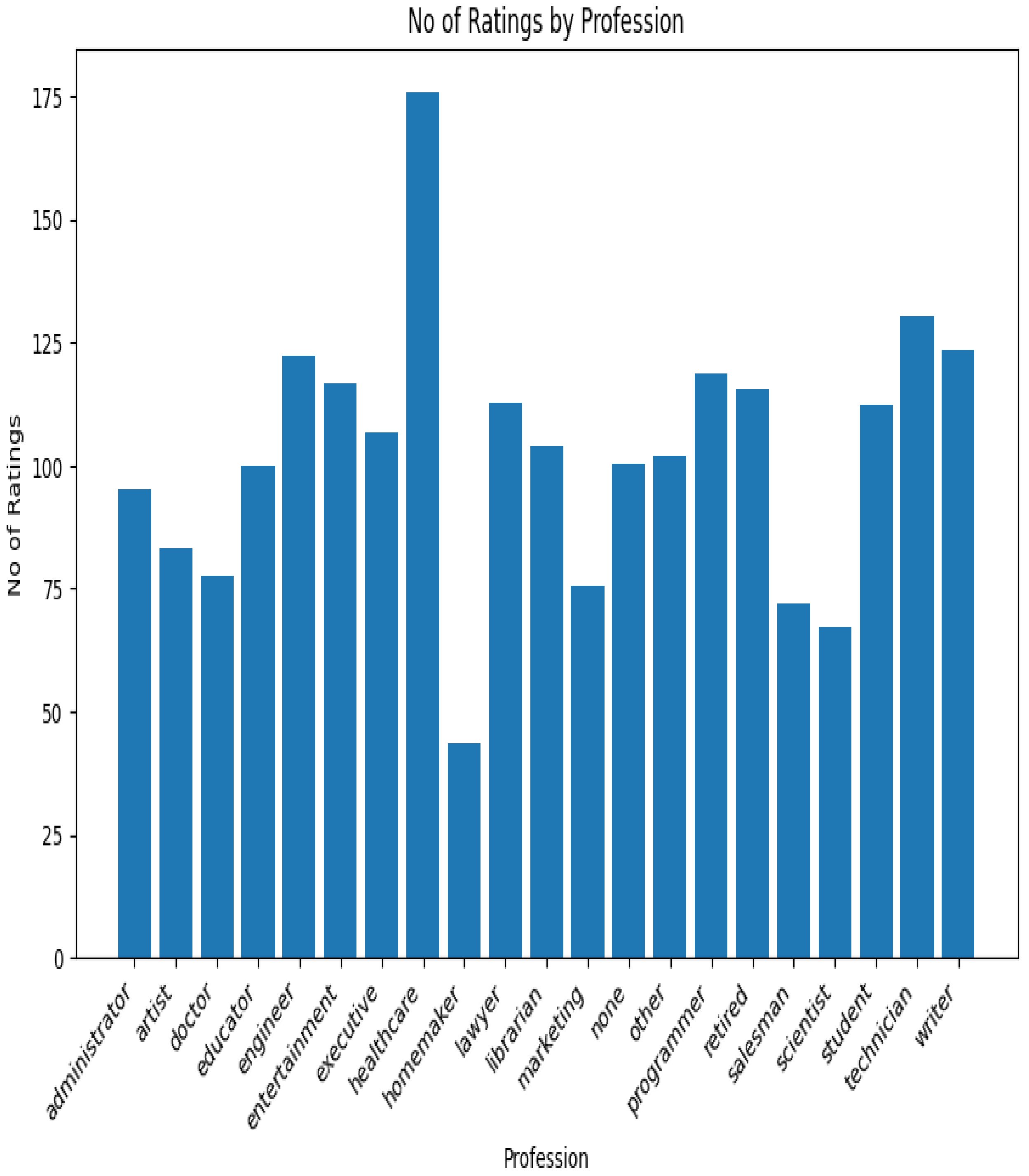
	user	genre_5
0	1	[Horror Sci-Fi Thriller, Drama Romance Sci-Fi ...
1	2	[Crime Film-Noir Mystery Thriller, Comedy Dram...
2	3	[Mystery Romance Thriller, Documentary, Crime ...
3	4	[Crime Drama Thriller, Mystery, Crime Drama My...
4	5	[Animation Comedy, Animation Comedy Thriller, ...
...	...	...
938	939	[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...

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

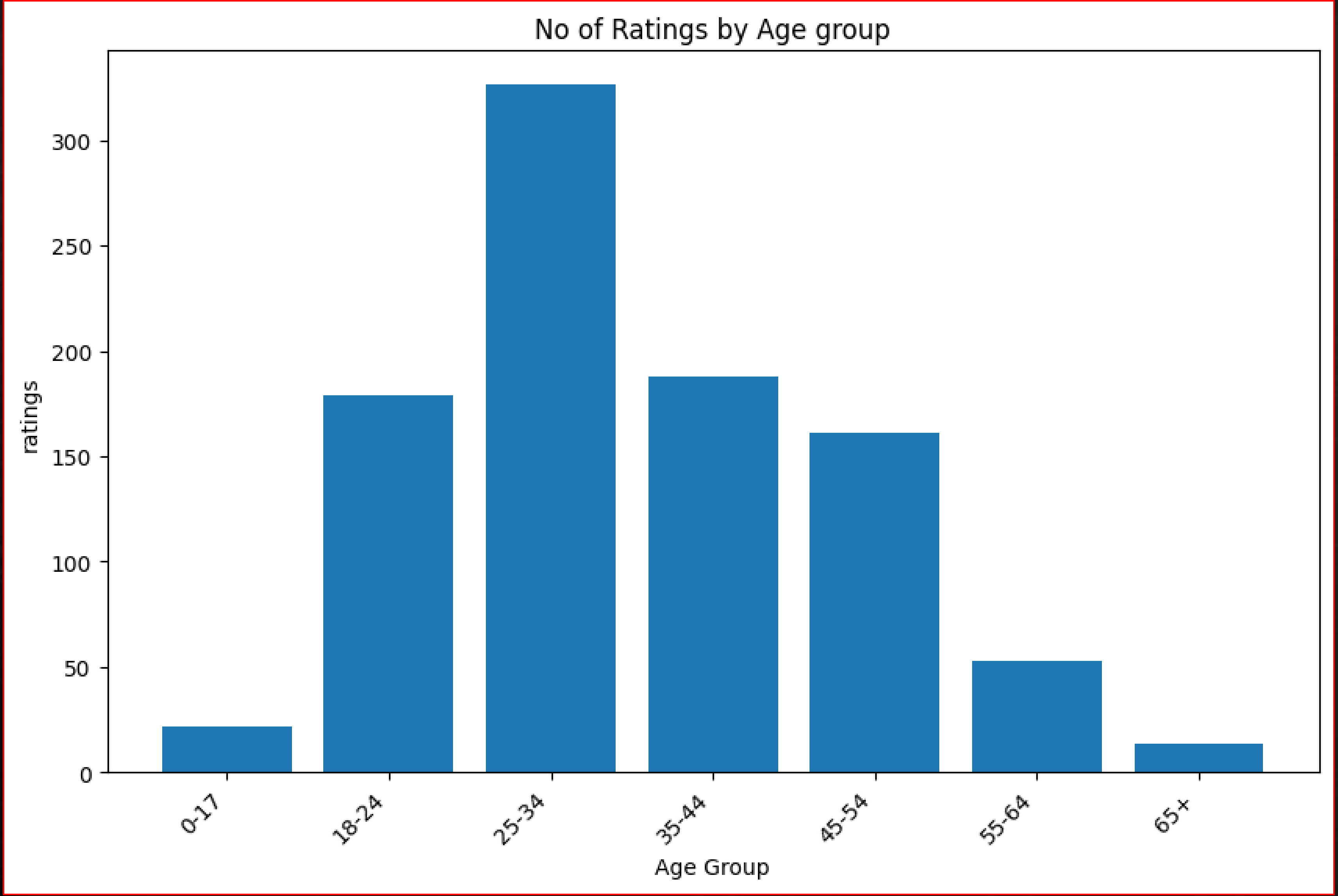
```
technician: ['Drama Horror', 'Romance Sci-Fi War', 'Animation Child Comedy Musical', 'Sci-Fi War', 'Crime Drama Thriller']
other: ['Drama Mystery Romance', 'Drama Musical', 'Sci-Fi', 'Drama Romance War', 'Comedy Drama']
writer: ['Drama Mystery', 'Comedy Musical', 'Child Comedy Musical', 'Sci-Fi War', '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
Average Rating
4.0
3.5
3.0
2.5
2.0
1.5
1.0
0.5
0.0
administrator
artist
doctor
educator
engineer
entertainment
executive
healthcare
homemaker
lawyer
librarian
marketing
none
other
programmer
retired
salesman
scientist
student
technician
writer
Profession



No of Ratings by Profession
No of Ratings
175
150
125
100
75
50
25
0
administrator
artist
doctor
educator
engineer
entertainment
executive
healthcare
homemaker
lawyer
librarian
marketing
none
other
programmer
retired
salesman
scientist
student
technician
writer
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 !