Building a Film Recommender Engine

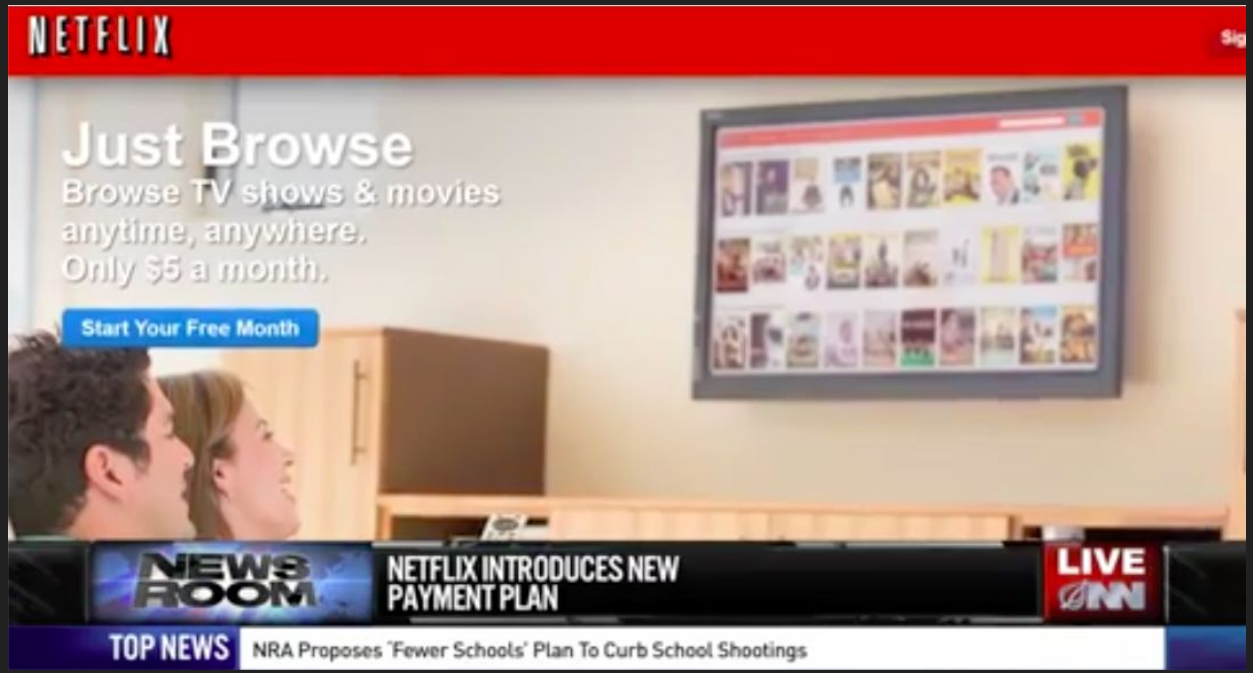
Owen Swetenburg

Data Science Process

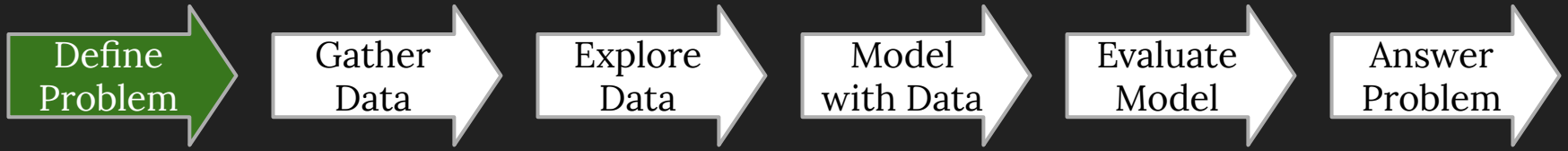
1. **Define problem**
2. **Gather data**
3. **Explore data**
4. **Model with data**
5. **Evaluate model**
6. **Answer problem**

Define Problem → Gather Data → Explore Data → Model with Data → Evaluate Model → Answer Problem

“Netflix Introduces New Browse Endlessly Plan”



https://www.theonion.com/netflix-introduces-new-browse-endlessly-plan-1819595604



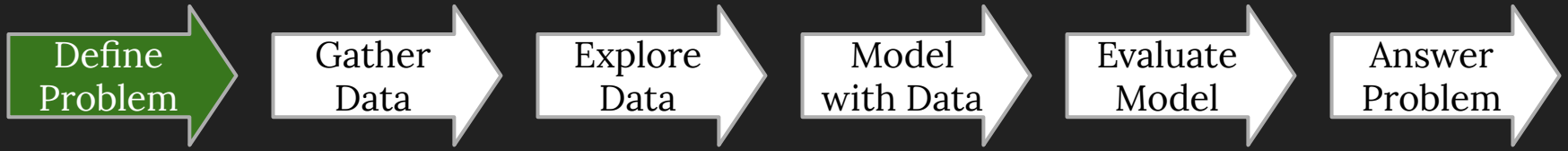
Types of choosers:

Maximizers

- Which option is the optimal choice?

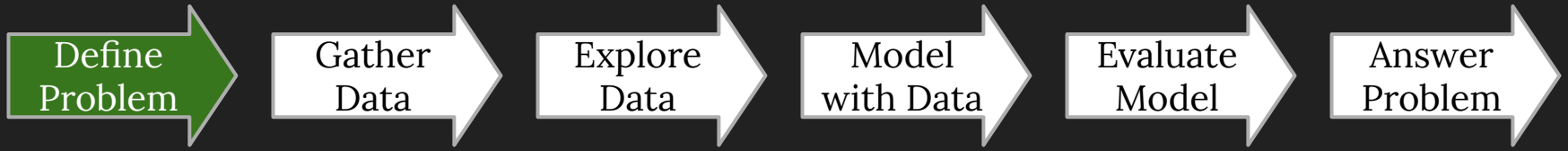
Satisficers

- Which option satisfies the criteria?



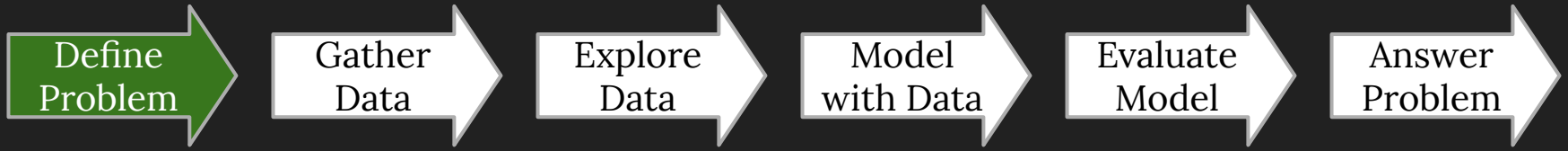
Problem Statement:

Using data science, how can we help people pick their next movie?



Answer:

Recommendation Engines!



Collaborative Recommendation Engines:

Item-based collaborative filtering

- Based on similar items
- “Because you bought this…”

User-based collaborative filtering

- Based on similar users
- “People who bought this item also bought this…”

Define Problem → **Gather Data** → Explore Data → Model with Data → Evaluate Model → Answer Problem

Source of Data:

- GroupLens Research
- MovieLens.org
  - **Full set**
    - 25 Million ratings
    - 62,000 films
    - 162,000 users
  - **Subset**
    - 100,000 ratings
    - 9,000 films
    - 600 users

movielens

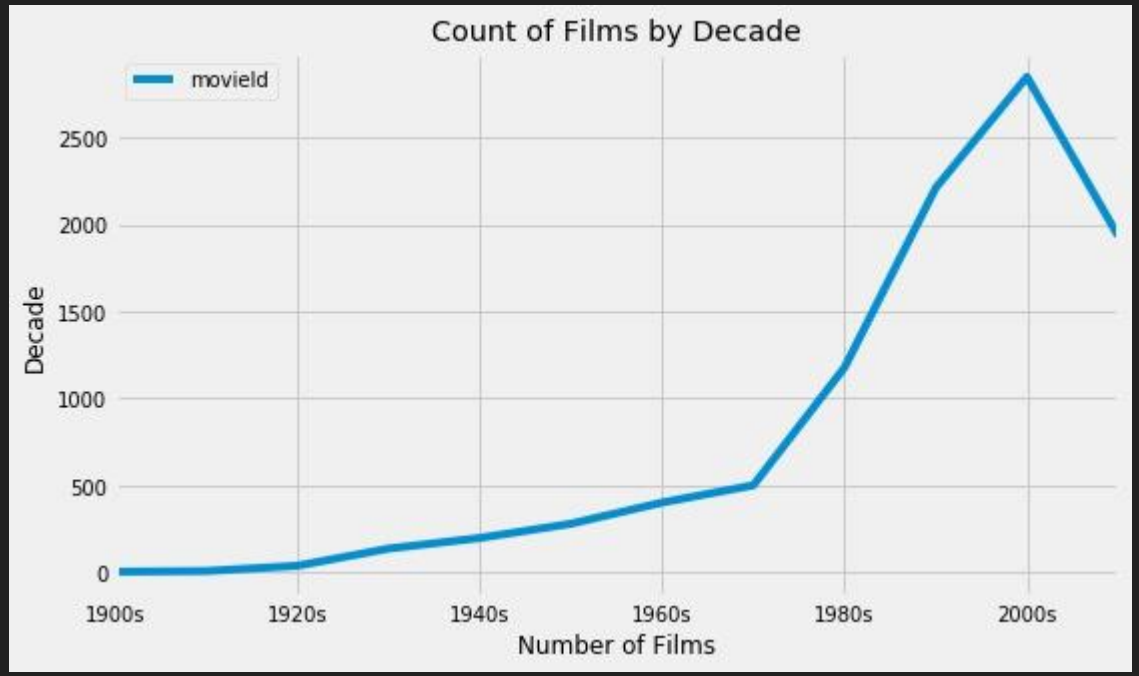
MovieLens is a web site that helps people find movies to watch. It has hundreds of thousands of registered users. We conduct online field experiments in MovieLens in the areas of automated content recommendation, recommendation interfaces, tagging-based recommenders and interfaces, member-maintained databases, and intelligent user interface design.

Exploratory Data Analysis

- Release year range: 1902 - 2018
- 19 genres:

  Action, Adventure, Animation, Children's, Comedy, Crime, Documentary, Drama, Fantasy, Film-Noir, Horror, Musical, Mystery, Romance, Sci-Fi, Thriller, War, Western, (no genres listed)

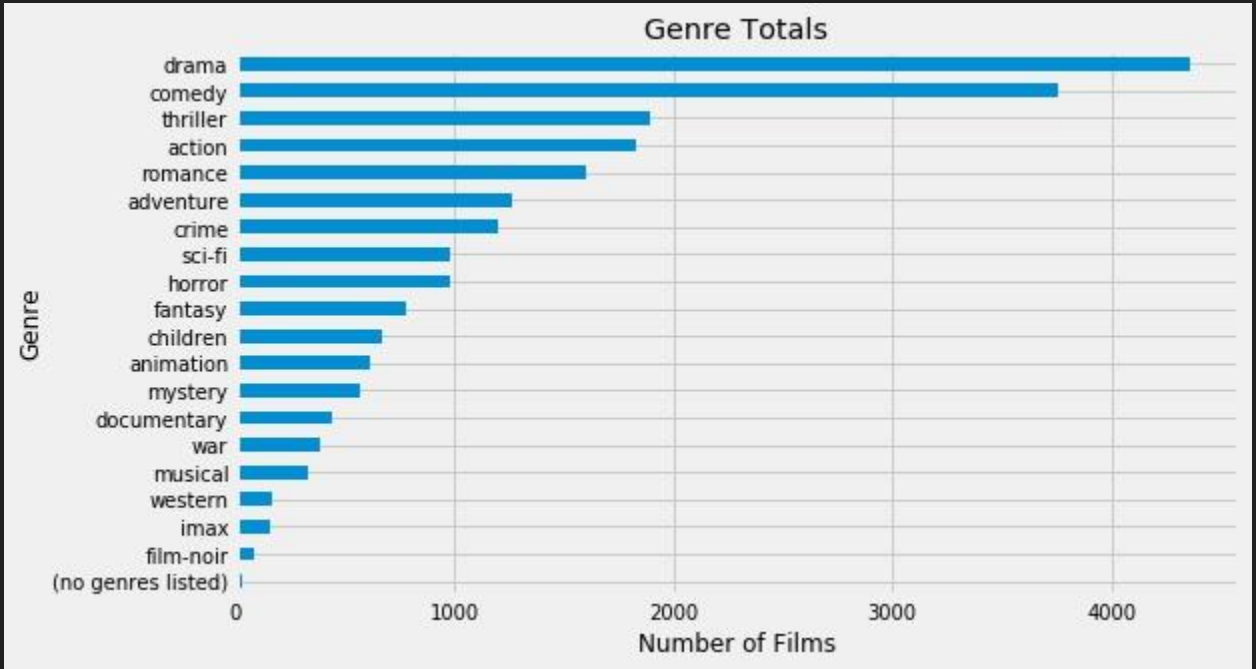
Define Problem → Gather Data → **Explore Data** → Model with Data → Evaluate Model → Answer Problem



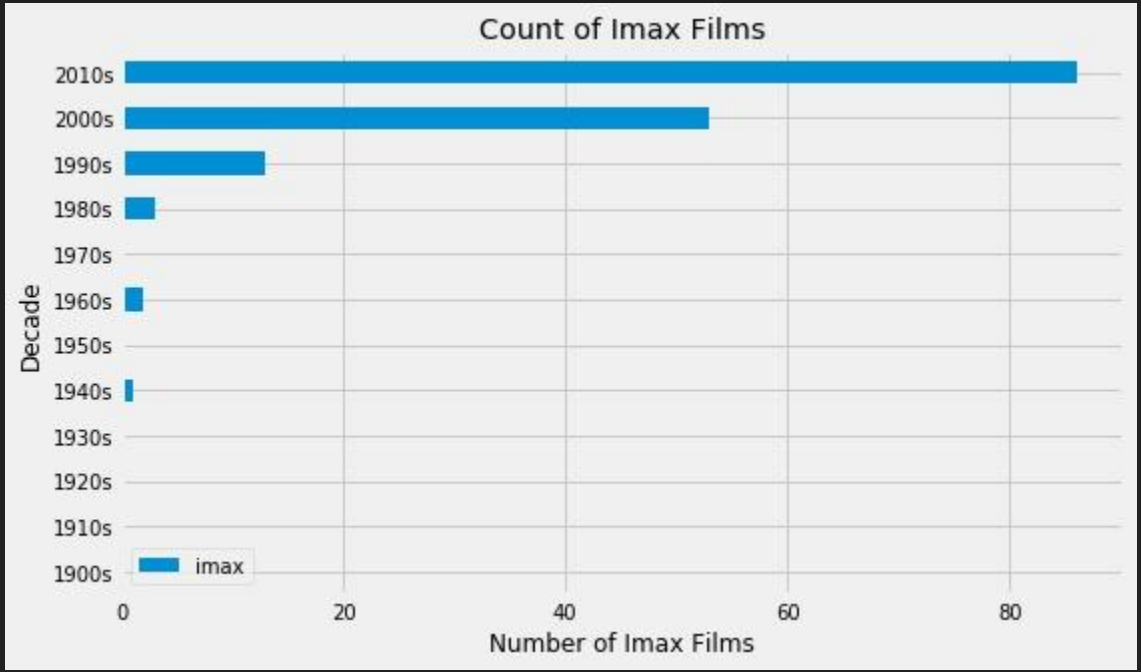

Define Problem
Gather Data
Explore Data
Model with Data
Evaluate Model
Answer Problem

Genres

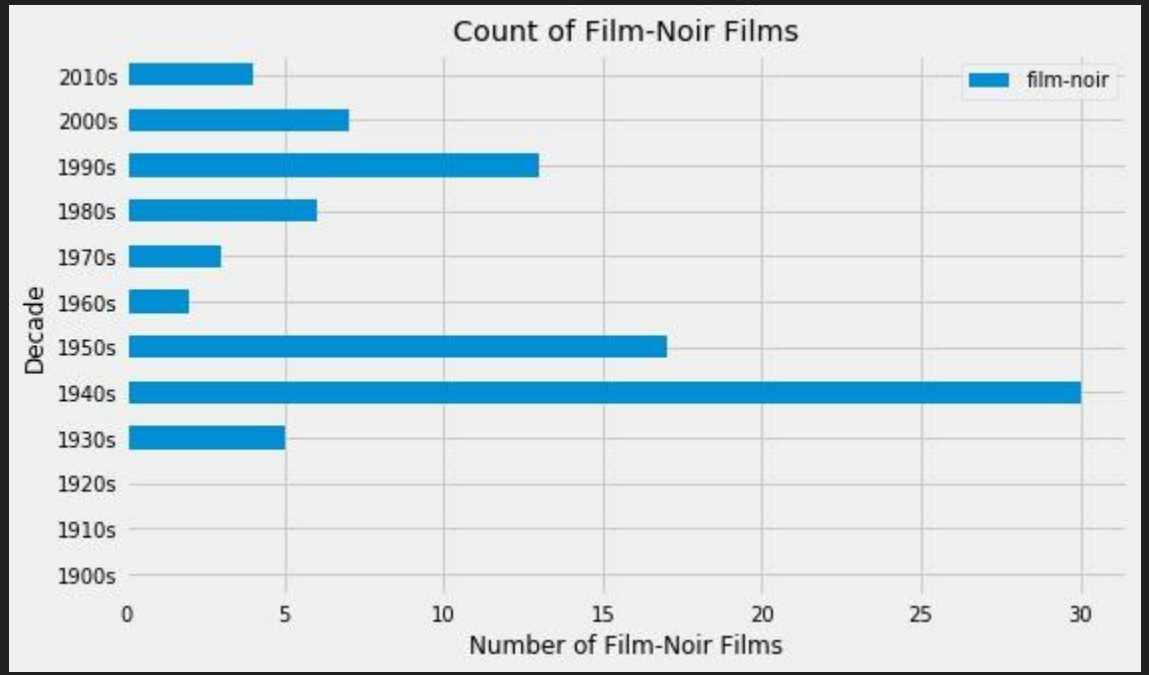
Define Problem → Gather Data → **Explore Data** → Model with Data → Evaluate Model → Answer Problem



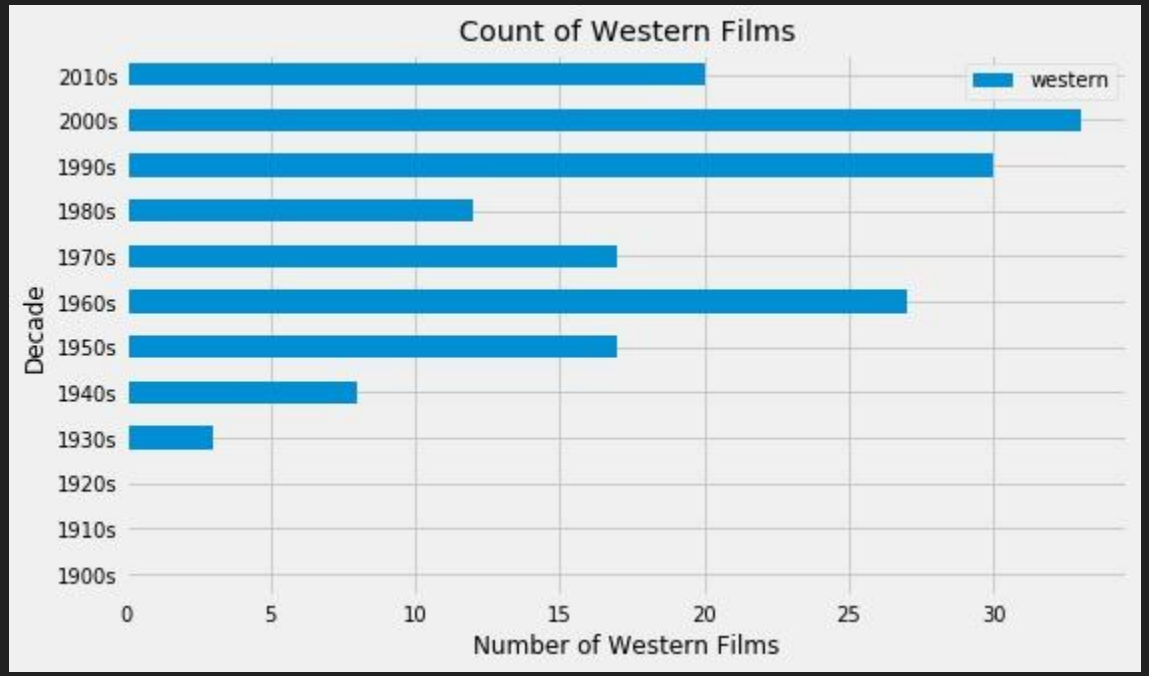
Define Problem → Gather Data → **Explore Data** → Model with Data → Evaluate Model → Answer Problem



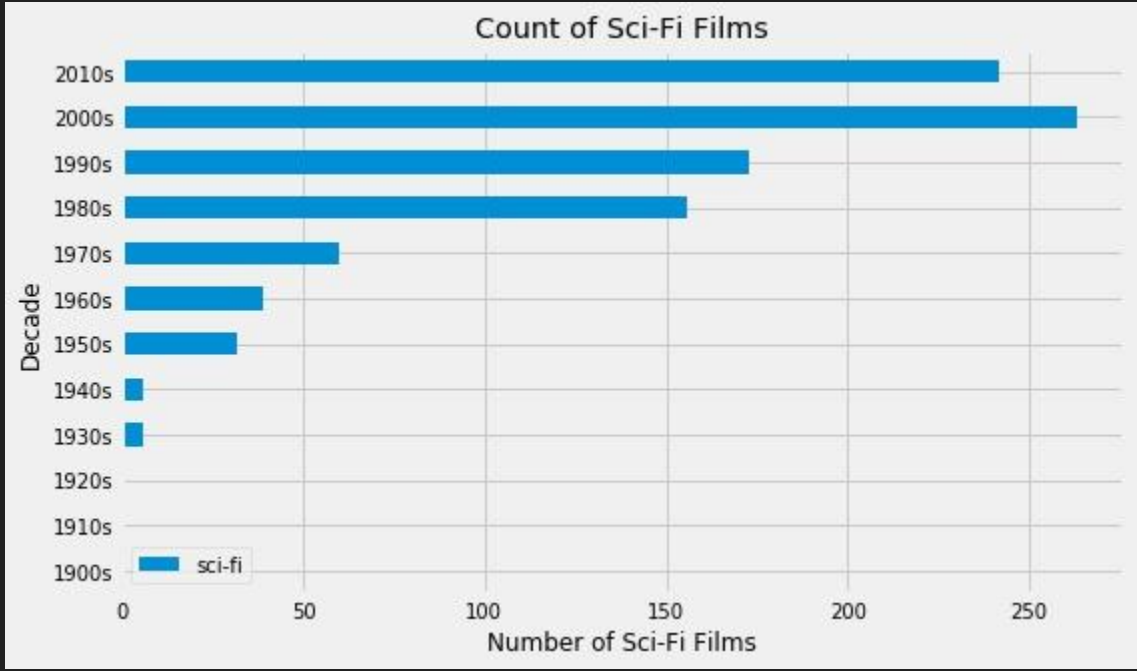
Define Problem → Gather Data → **Explore Data** → Model with Data → Evaluate Model → Answer Problem



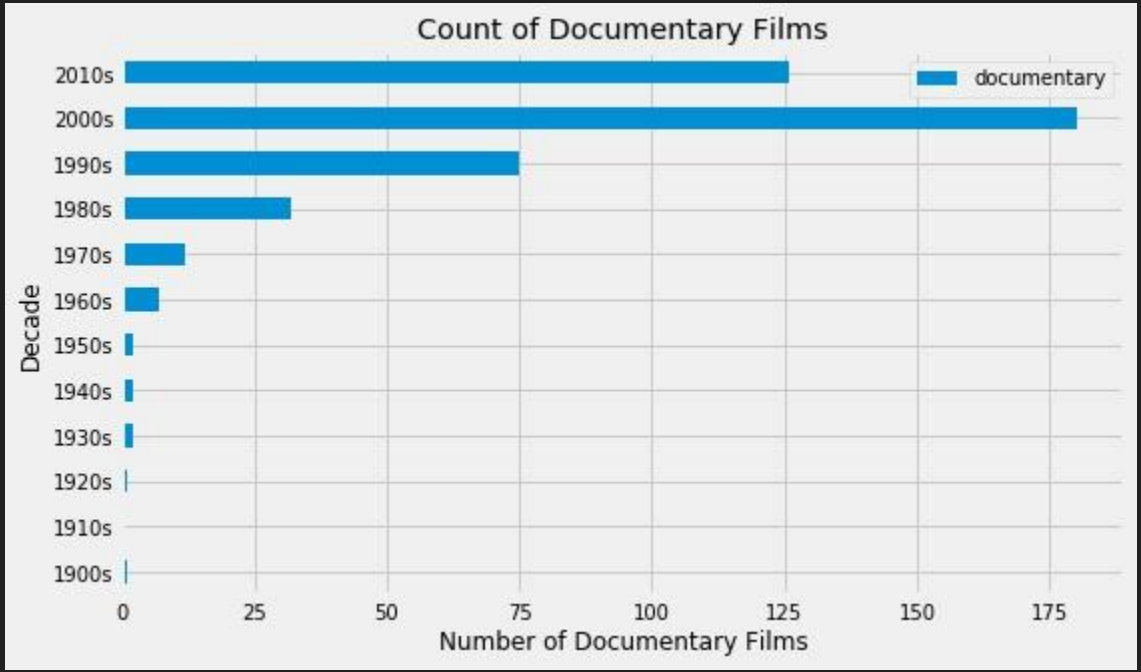
Define Problem → Gather Data → **Explore Data** → Model with Data → Evaluate Model → Answer Problem



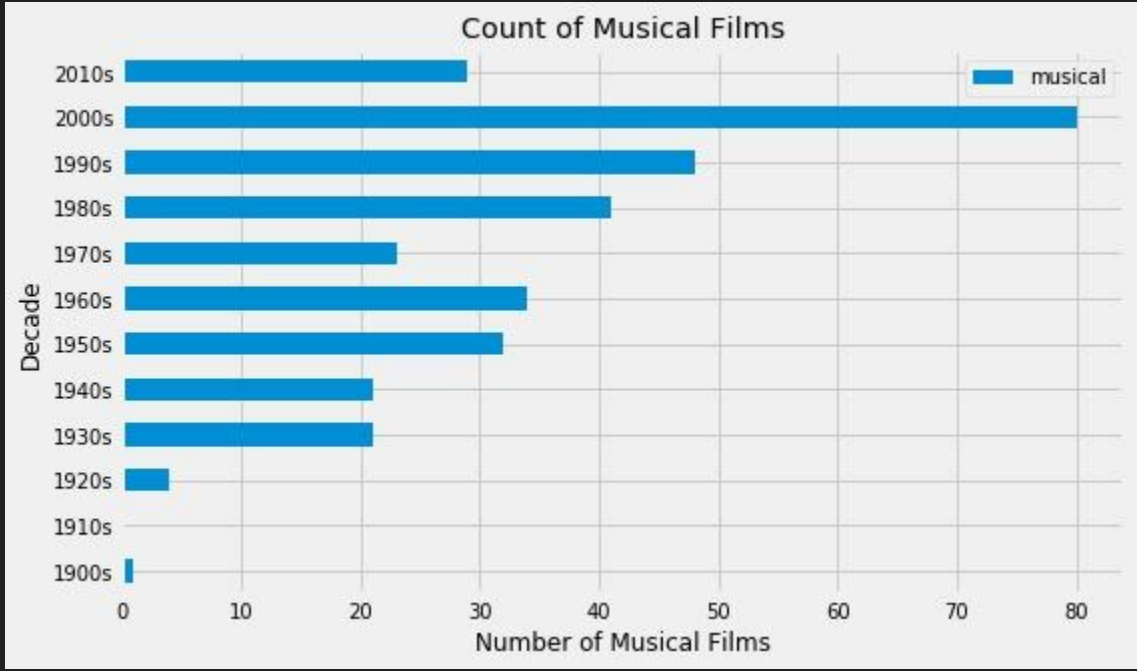
Define Problem → Gather Data → **Explore Data** → Model with Data → Evaluate Model → Answer Problem



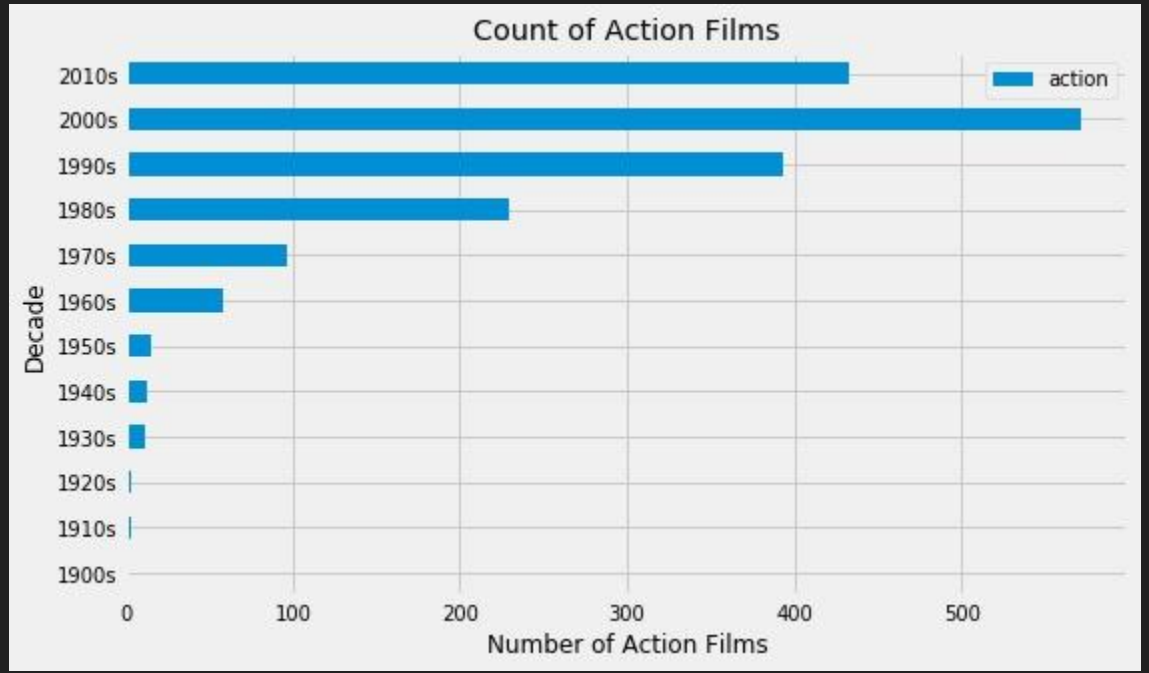
Define Problem → Gather Data → **Explore Data** → Model with Data → Evaluate Model → Answer Problem



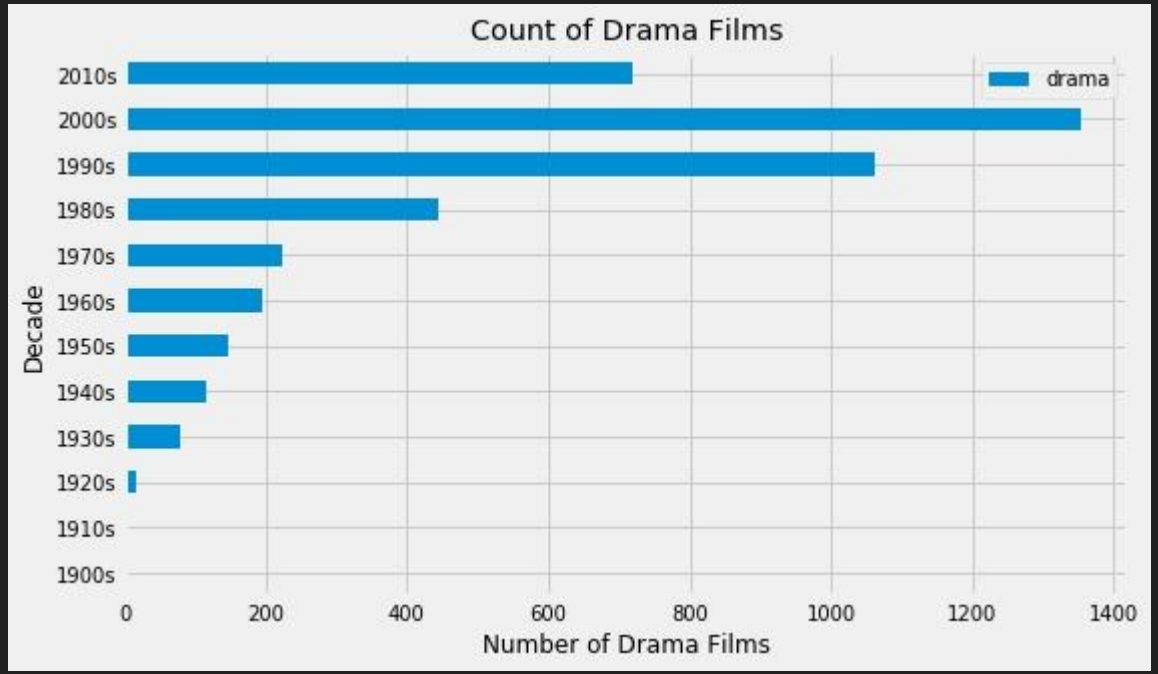
Define Problem → Gather Data → **Explore Data** → Model with Data → Evaluate Model → Answer Problem



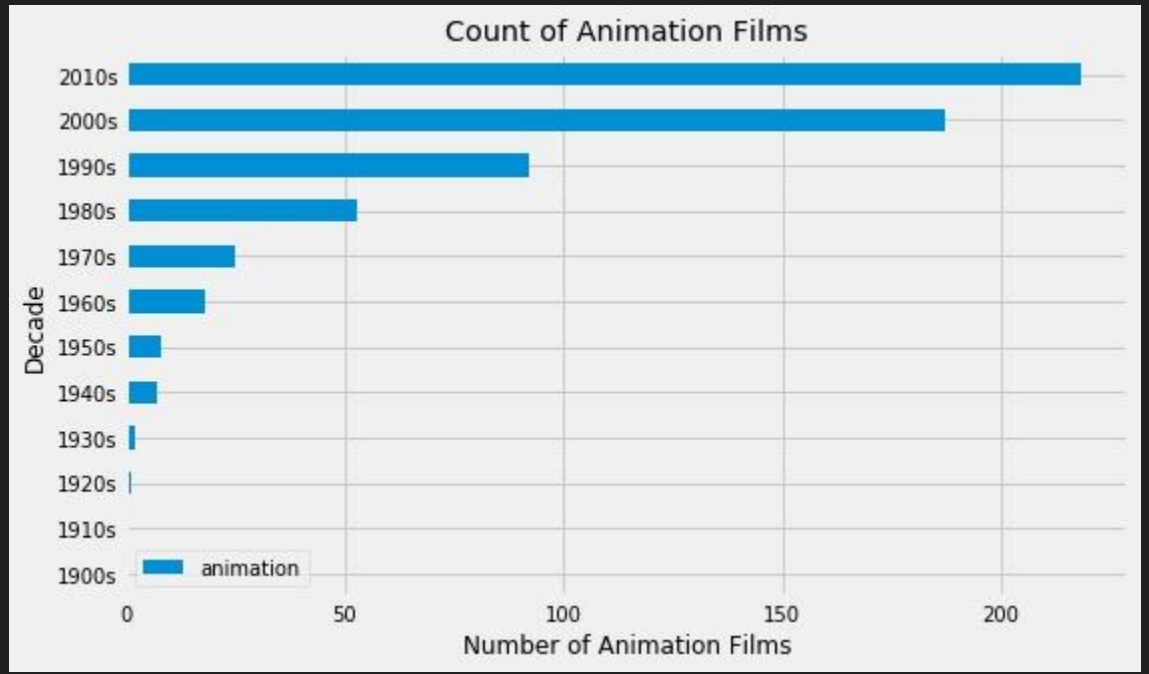
Define Problem → Gather Data → **Explore Data** → Model with Data → Evaluate Model → Answer Problem



Define Problem → Gather Data → **Explore Data** → Model with Data → Evaluate Model → Answer Problem



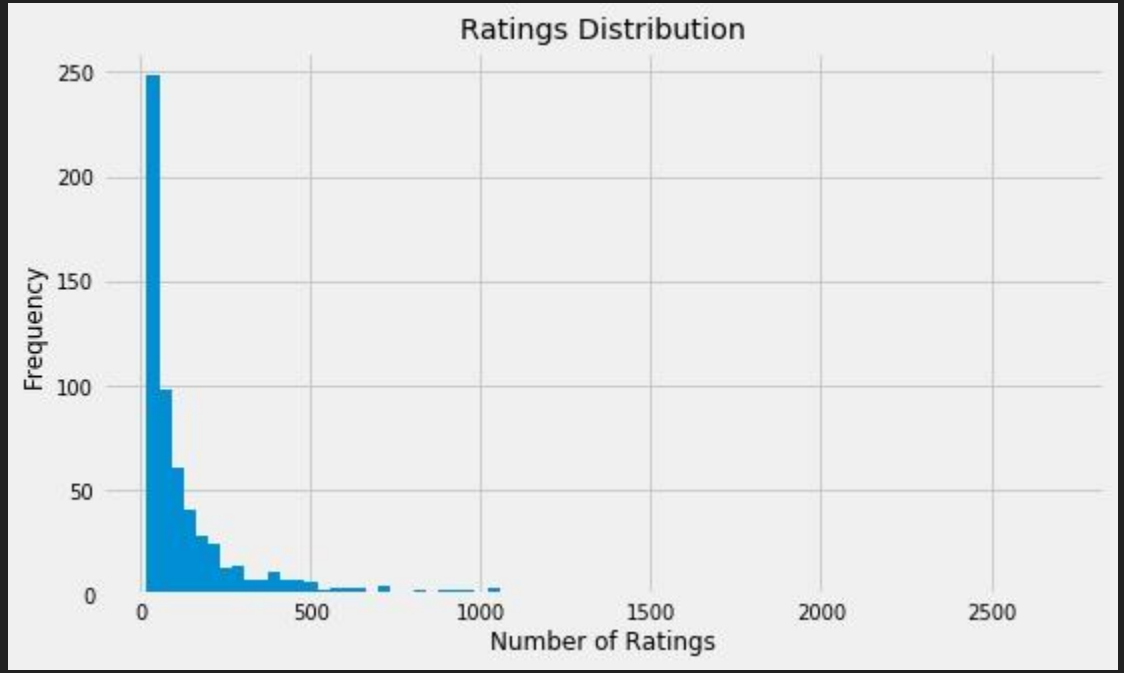
Define Problem → Gather Data → **Explore Data** → Model with Data → Evaluate Model → Answer Problem



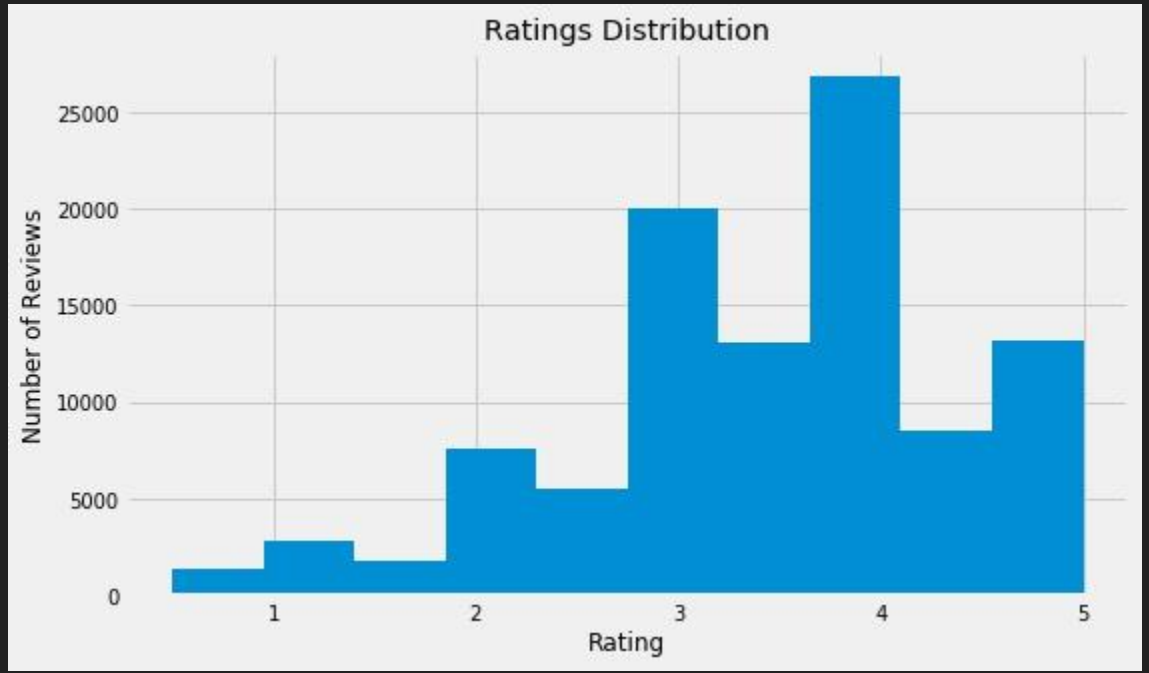

Define Problem
Gather Data
Explore Data
Model with Data
Evaluate Model
Answer Problem

Ratings

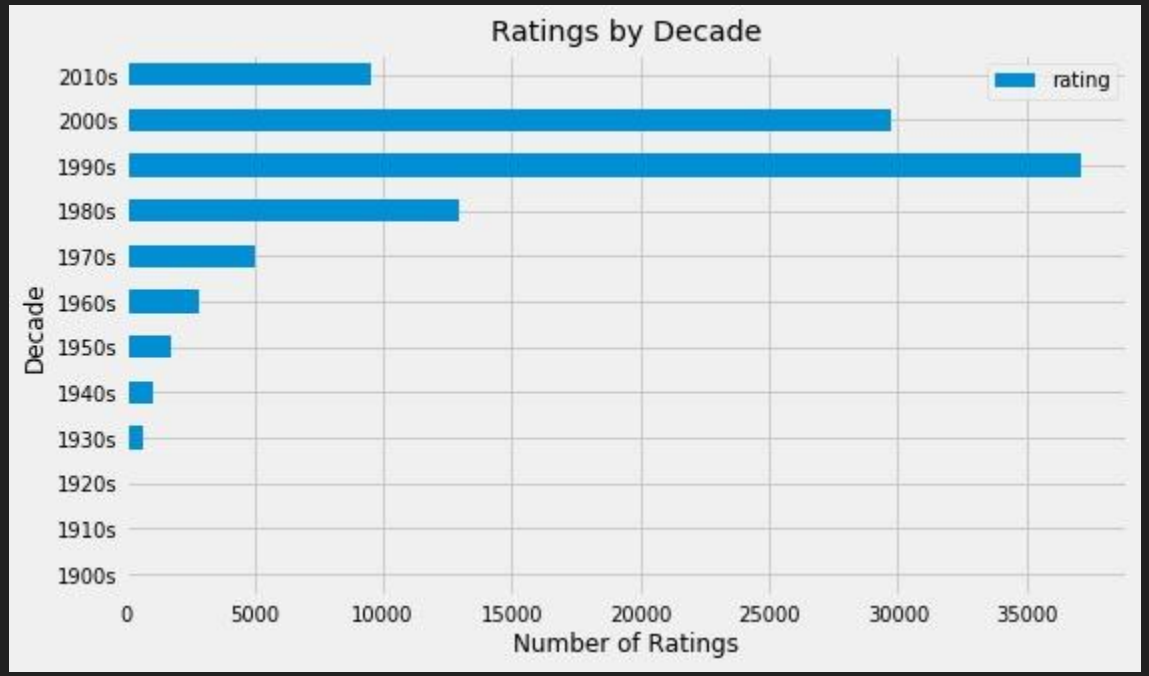
Define Problem → Gather Data → **Explore Data** → Model with Data → Evaluate Model → Answer Problem



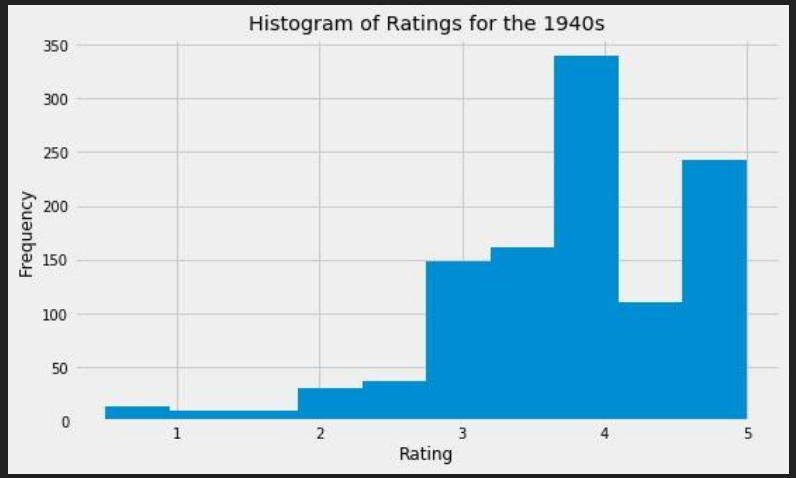
Define Problem → Gather Data → **Explore Data** → Model with Data → Evaluate Model → Answer Problem



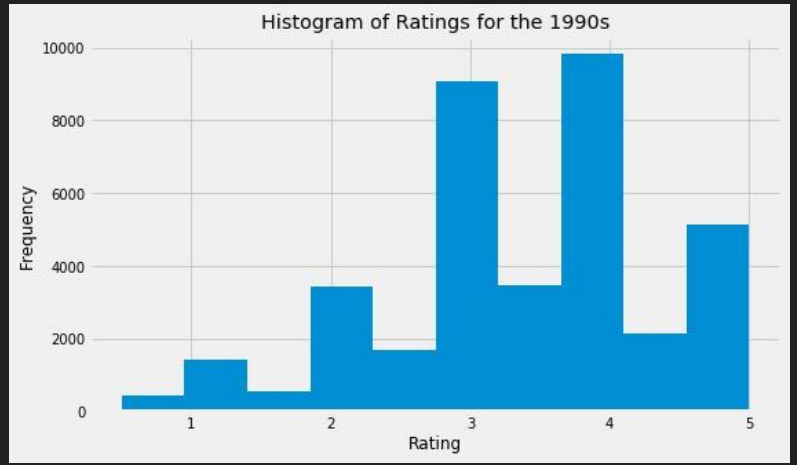
Define Problem → Gather Data → **Explore Data** → Model with Data → Evaluate Model → Answer Problem



Define Problem → Gather Data → **Explore Data** → Model with Data → Evaluate Model → Answer Problem



decade	rating
1940s	3.87
1950s	3.85
1960s	3.81
1970s	3.78
1920s	3.74
1930s	3.73
1980s	3.52
2010s	3.49
2000s	3.47
1990s	3.43
1900s	3.31
1910s	3.31



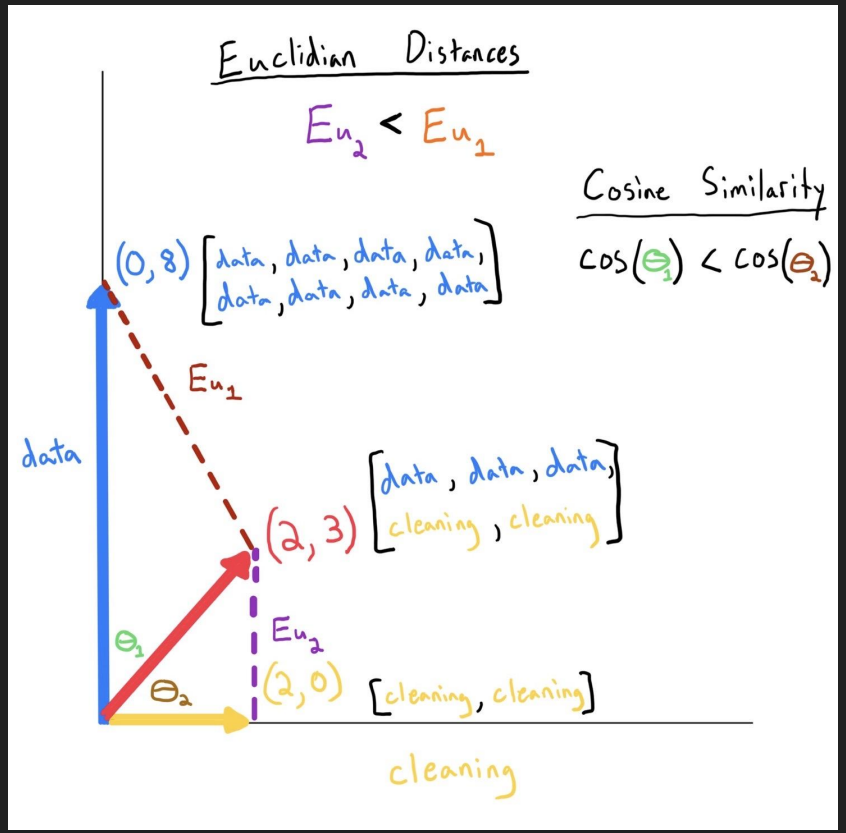

Define Problem
Gather Data
Explore Data
Model with Data
Evaluate Model
Answer Problem

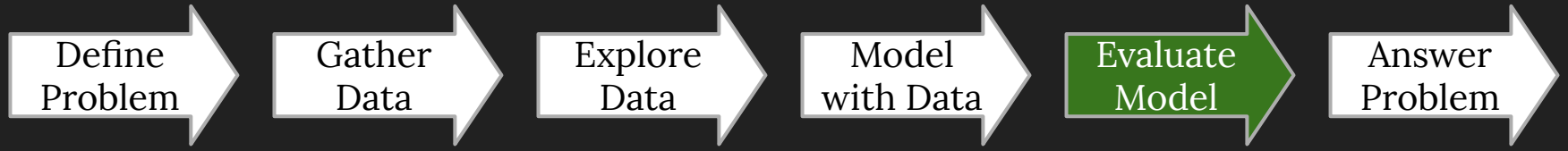
Modeling

Item-based collaborative engine

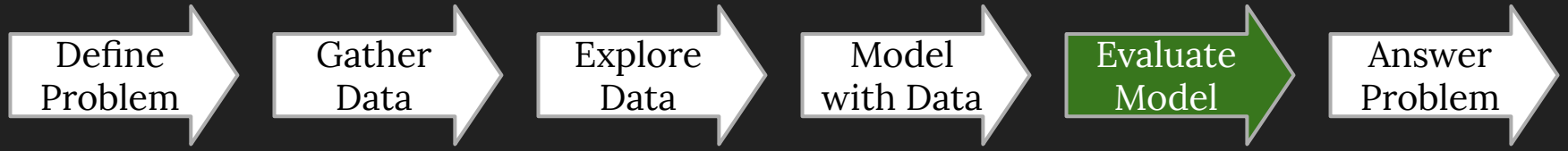
- Pivot table
  - Sparse matrix to
    - handles missing rating values
- Cosine similarity
  - Comparing vectors
  - Range is [0 : 1]
  - 1 is most similar

Define Problem → Gather Data → Explore Data → **Model with Data** → Evaluate Model → Answer Problem





Model Evaluation



High performance on films with many ratings

```
Alice in Wonderland (1951)
Genre: Adventure|Animation|Children|Fantasy|Musical
Average rating: 3.375
Number of ratings: 40

10 closest films:
title
Peter Pan (1953)                  0.335316
Bambi (1942)                      0.383785
Robin Hood (1973)                 0.410830
Sword in the Stone, The (1963)    0.433815
Cinderella (1950)                 0.443176
Sleeping Beauty (1959)            0.452931
Pinocchio (1940)                  0.454511
Dumbo (1941)                      0.474701
Little Mermaid, The (1989)        0.488092
Jungle Book, The (1967)           0.493763
Name: Alice in Wonderland (1951), dtype: float64

**********************************************************************************************************
```

Define Problem → Gather Data → Explore Data → Model with Data → **Evaluate Model** → Answer Problem

High performance on films with many ratings

```
Alice in Wonderland (2010)
Genre: Adventure|Fantasy|IMAX
Average rating: 2.875
Number of ratings: 28

10 closest films:
title
Charlie and the Chocolate Factory (2005)                0.485144
Madagascar (2005)                                       0.523539
Hobbit: An Unexpected Journey, The (2012)               0.525638
National Treasure: Book of Secrets (2007)               0.526238
Up in the Air (2009)                                    0.531201
Pirates of the Caribbean: At World's End (2007)         0.534200
Sweeney Todd: The Demon Barber of Fleet Street (2007)   0.537643
Kick-Ass (2010)                                         0.539236
Life of Pi (2012)                                       0.543657
Corpse Bride (2005)                                     0.546206
Name: Alice in Wonderland (2010), dtype: float64

**********
```

Define Problem → Gather Data → Explore Data → Model with Data → **Evaluate Model** → Answer Problem

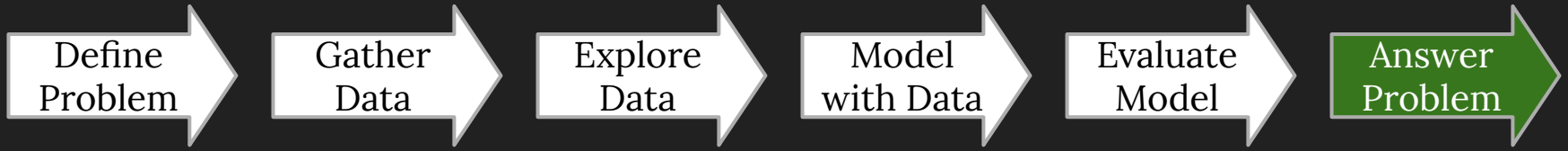
Low performance on films with fewer ratings

```
Alice in Wonderland (1933)
Genre: Adventure|Children|Fantasy
Average rating: 4.0
Number of ratings: 1

10 closest films:
title
Abominable Snowman, The (Abominable Snowman of the Himalayas, The) (1957)    0.0
Aelita: The Queen of Mars (Aelita) (1924)                                    0.0
Agony and the Ecstasy, The (1965)                                            0.0
7 Faces of Dr. Lao (1964)                                                    0.0
Alice in Wonderland (1933)                                                   0.0
Alien from L.A. (1988)                                                       0.0
20 Million Miles to Earth (1957)                                             0.0
Allegro non troppo (1977)                                                    0.0
10th Victim, The (La decima vittima) (1965)                                  0.0
American Friend, The (Amerikanische Freund, Der) (1977)                      0.0
Name: Alice in Wonderland (1933), dtype: float64

**********************************************************************************************************
```

Define Problem → Gather Data → Explore Data → Model with Data → **Evaluate Model** → Answer Problem

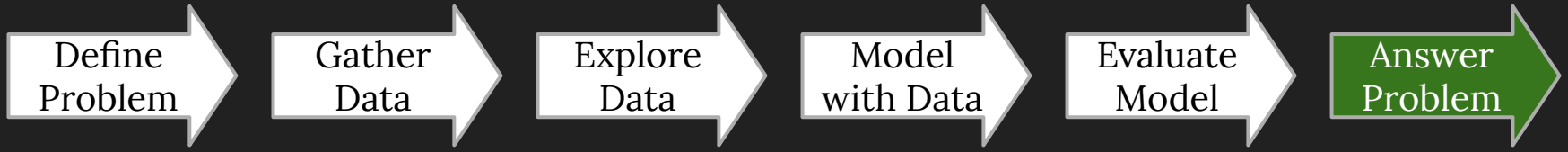
“Yeah? Well you know that’s just like uh your opinion, man.”

```
Big Lebowski, The (1998)
Genre: Comedy|Crime
Average rating: 3.9245283018867925
Number of ratings: 106

10 closest films:
title
Reservoir Dogs (1992)                    0.396056
Clockwork Orange, A (1971)               0.421605
Snatch (2000)                            0.432700
Truman Show, The (1998)                  0.439860
Fear and Loathing in Las Vegas (1998)    0.443134
Being John Malkovich (1999)              0.446035
Full Metal Jacket (1987)                 0.451033
Kill Bill: Vol. 2 (2004)                 0.454125
Office Space (1999)                      0.455962
Fight Club (1999)                        0.460324
Name: Big Lebowski, The (1998), dtype: float64

**********
```



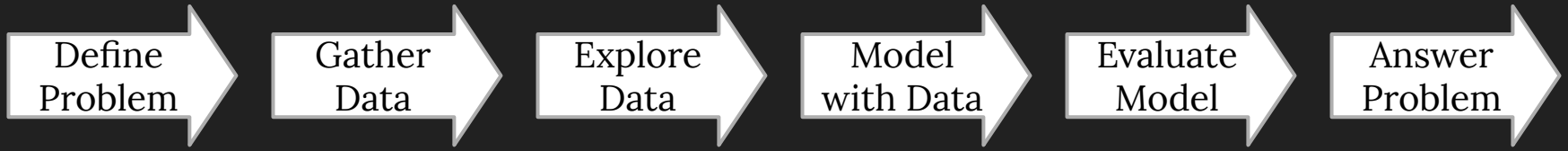
Problem Statement:

Using data science, how can we help people pick their next movie?



Problem Answer:

- Build an item-based recommender system
  - Pro:
    - With adequate data, this engine can provide remarkably keen suggestions
  - Con:
    - Requires lots of participation



Future Work:

- Use full-size dataset
- Incorporate IMDB data
  - Other films by the same director
- Examine user tags using sentiment analysis
- Explore timestamps for ratings and tags
- Combining user-based and item-based collaborators

Questions?