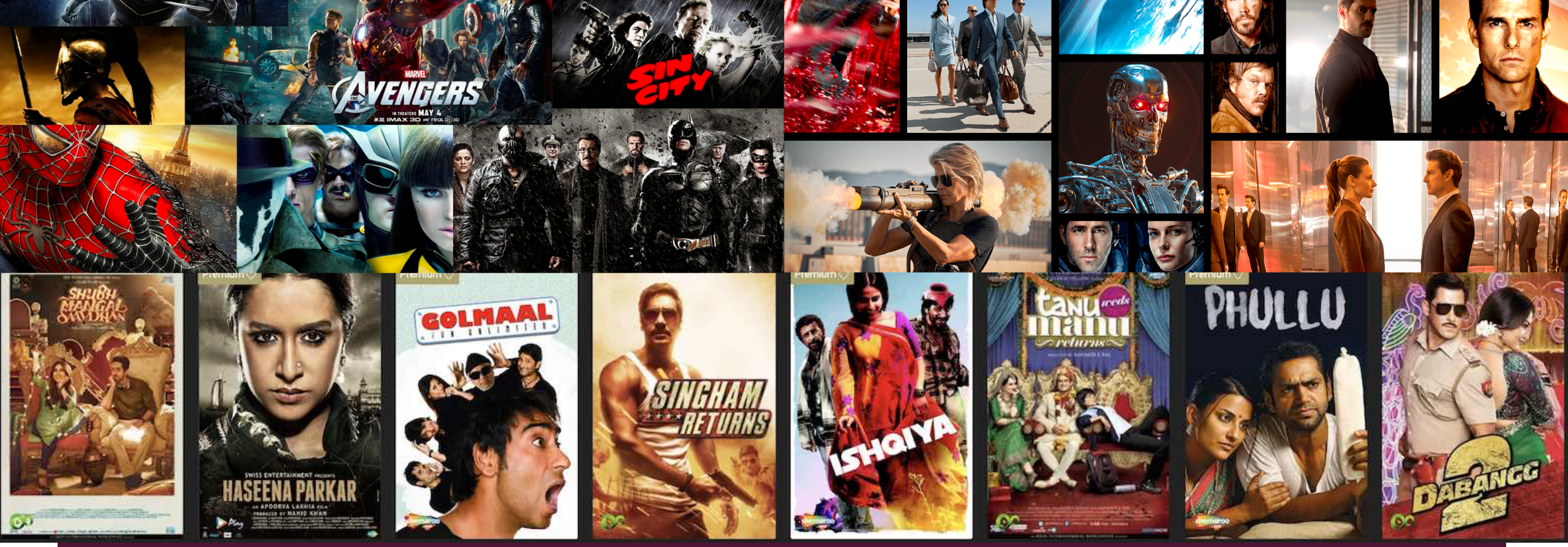


MOVIE RECOMMENDATIONS

SMES

AMRITA DUTTA, METIKA SIKKA, SANCHYA SAHAY, AND MALIN ORTENBLAD

TABLE OF CONTEXT

- Movie Market Overview
- Word Clouds
- Recommendation Algorithms:
  - Shortest Path (by movie title)
  - Cosine similarity (by movie title)
  - TF-IDF (by keywords)
  - Cosine similarity (by keywords and genre)
- Conclusion

People spend on average nearly an hour(51 min to be exact) per day deciding what to watch

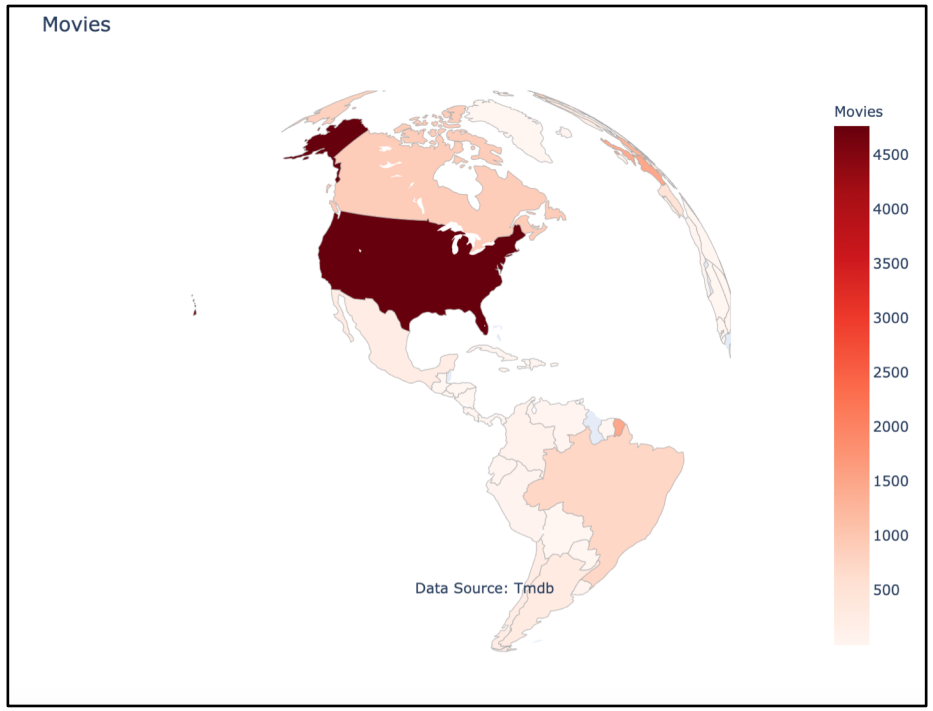
51 min per day, 7 days a week → **357 minutes per week** → **18564 minutes per year** → **309.4 hours per year**

- **Issues:**
  - More movie options than ever before
  - A big time-loss trying to choose between these options
  - Sometimes you have a movie in mind and sometimes you have a "feeling"
- **Solution:**
  - We decided to build models with two functionalities:
    - People searching for recommendations based on a movie title that they had watched and liked
    - People searching for recommendation based on their mood, thoughts and specific keywords
- **The Data:**
  - Used APIs to extract the data from the TMDB
  - All movies released worldwide in the past 2 years (2018 & 2019)

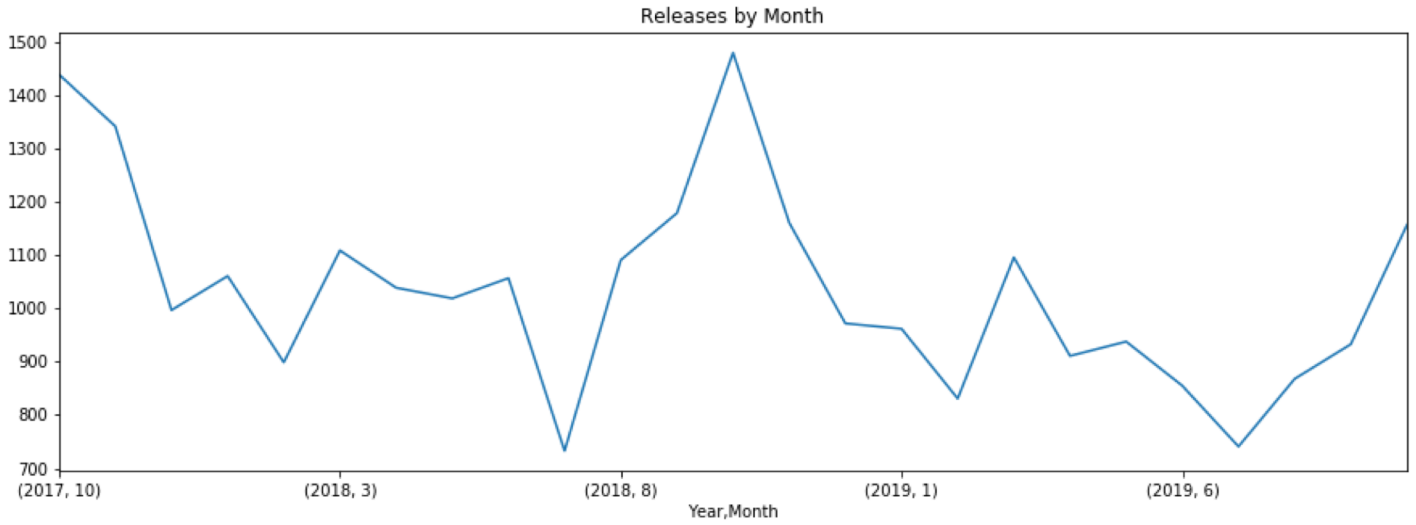
SOURCE: HTTPS://LOVIN.IE/ENTERTAINMENT/TV-MOVIES/HERES-HOW-LONG-WE-SPEND-LOOKING-FOR-TV-SHOWS-A-DAY

About 17,000 new movies are released each year, US is the clear market leader

Movie Releases by Country (Oct 18- Oct 19)



Movie Releases by Month (Oct 18- Oct 19)



Link: file:///Users/sanchyasahay/Documents/Fall%202019/Data%20Analytics/DA%20projec/Movies%20Reco/Code/temp-plot.html

Word clouds provide a good overview of common words by Genre

Genre: Animation

Genre: Romance

Genre: Crime

Genre: Horror

The first model calculates the shortest path between movies in a network with the edge weight representing two movies' similarity scores

1. Calculate similarity scores between all movies in our data

The first model calculates the shortest path between movies in a network with the edge weight representing two movies' similarity scores

1. Calculate similarity scores between all movies in our data

Genre

$$Genre\ Ratio = \frac{Genre\ overlap}{Unique\ genres\ between\ movies}$$

Director

$$Director\ Overlap$$

Production Country

$$Production\ Country\ Overlap$$

The first model calculates the shortest path between movies in a network with the edge weight representing two movies' similarity scores

1. Calculate similarity scores between all movies in our data

Genre

$$Genre\ Ratio = \frac{Genre\ overlap}{Unique\ genres\ between\ movies}$$

Director

$$Director\ Overlap$$

Production Country

$$Production\ Country\ Overlap$$

2. Add weightage to each score component

The first model calculates the shortest path between movies in a network with the edge weight representing two movies' similarity scores

1. Calculate similarity scores between all movies in our data

Genre

$$Genre\ Ratio = \frac{Genre\ overlap}{Unique\ genres\ between\ movies}$$

Director

$$Director\ Overlap$$

Production Country

$$Production\ Country\ Overlap$$

2. Add weightage to each score component

Similarity Score = 1 - (Genre Ration x 0.8) + (Director Overlap x 0.1) + (Production Country Overlap x 0.1)

The first model calculates the shortest path between movies in a network with the edge weight representing two movies' similarity scores

1. Calculate similarity scores between all movies in our data

Genre

$$Genre\ Ratio = \frac{Genre\ overlap}{Unique\ genres\ between\ movies}$$

Director

$$Director\ Overlap$$

Production Country

$$Production\ Country\ Overlap$$

2. Add weightage to each score component

Similarity Score = 1 - (Genre Ration x 0.8) + (Director Overlap x 0.1) + (Production Country Overlap x 0.1)

3. Adding edges to network and calculate shortest path from input movie

The first model calculates the shortest path between movies in a network with the edge weight representing two movies' similarity scores

1. Calculate similarity scores between all movies in our data

Genre

$$Genre\ Ratio = \frac{Genre\ overlap}{Unique\ genres\ between\ movies}$$

Director

Director Overlap

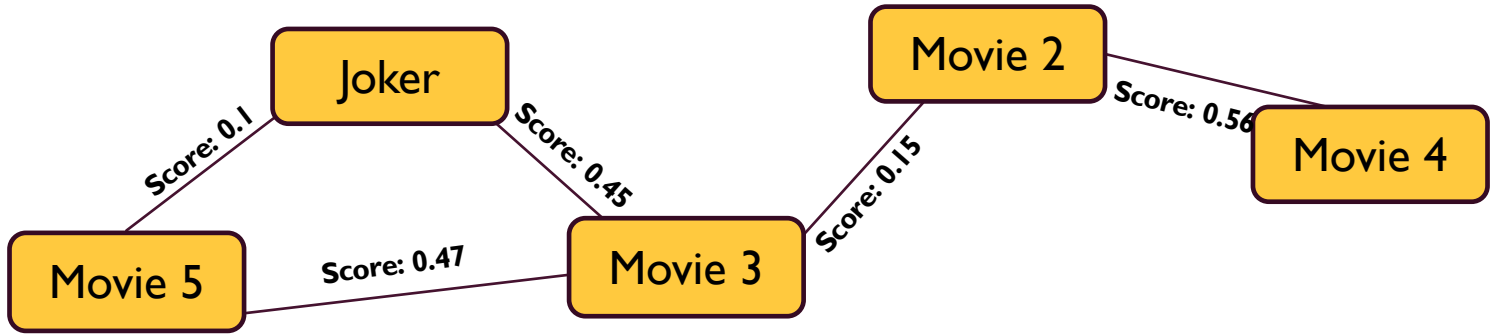
Production Country

Production Country Overlap

2. Add weightage to each score component

Similarity Score = 1 - (Genre Ration x 0.8) + (Director Overlap x 0.1) + (Production Country Overlap x 0.1)

3. Adding edges to network and calculate shortest path from input movie



The first model calculates the shortest path between movies in a network with the edge weight representing two movies' similarity scores

1. Calculate similarity scores between all movies in our data

Genre

$$Genre\ Ratio = \frac{Genre\ overlap}{Unique\ genres\ between\ movies}$$

Director

$$Director\ Overlap$$

Production Country

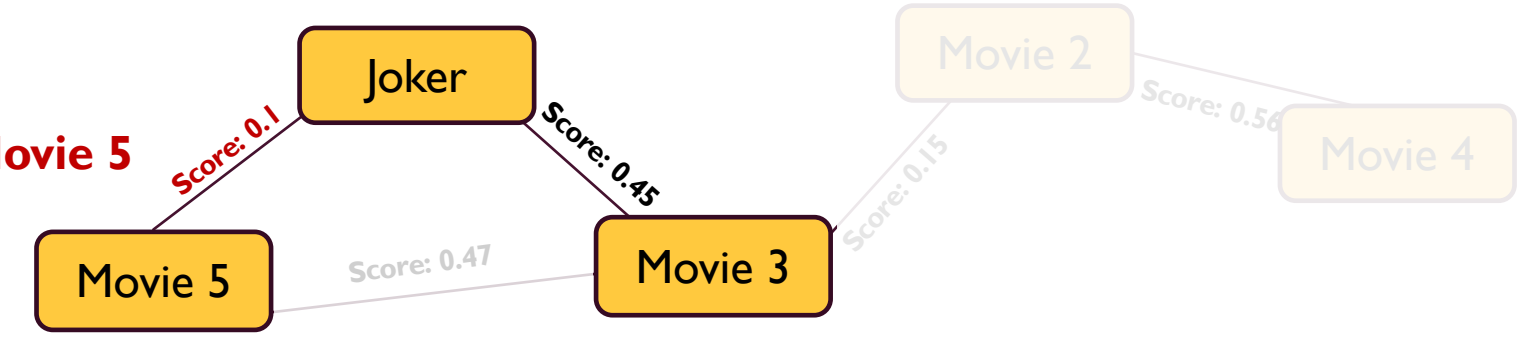
$$Production\ Country\ Overlap$$

2. Add weightage to each score component

Similarity Score = 1 - (Genre Ration x 0.8) + (Director Overlap x 0.1) + (Production Country Overlap x 0.1)

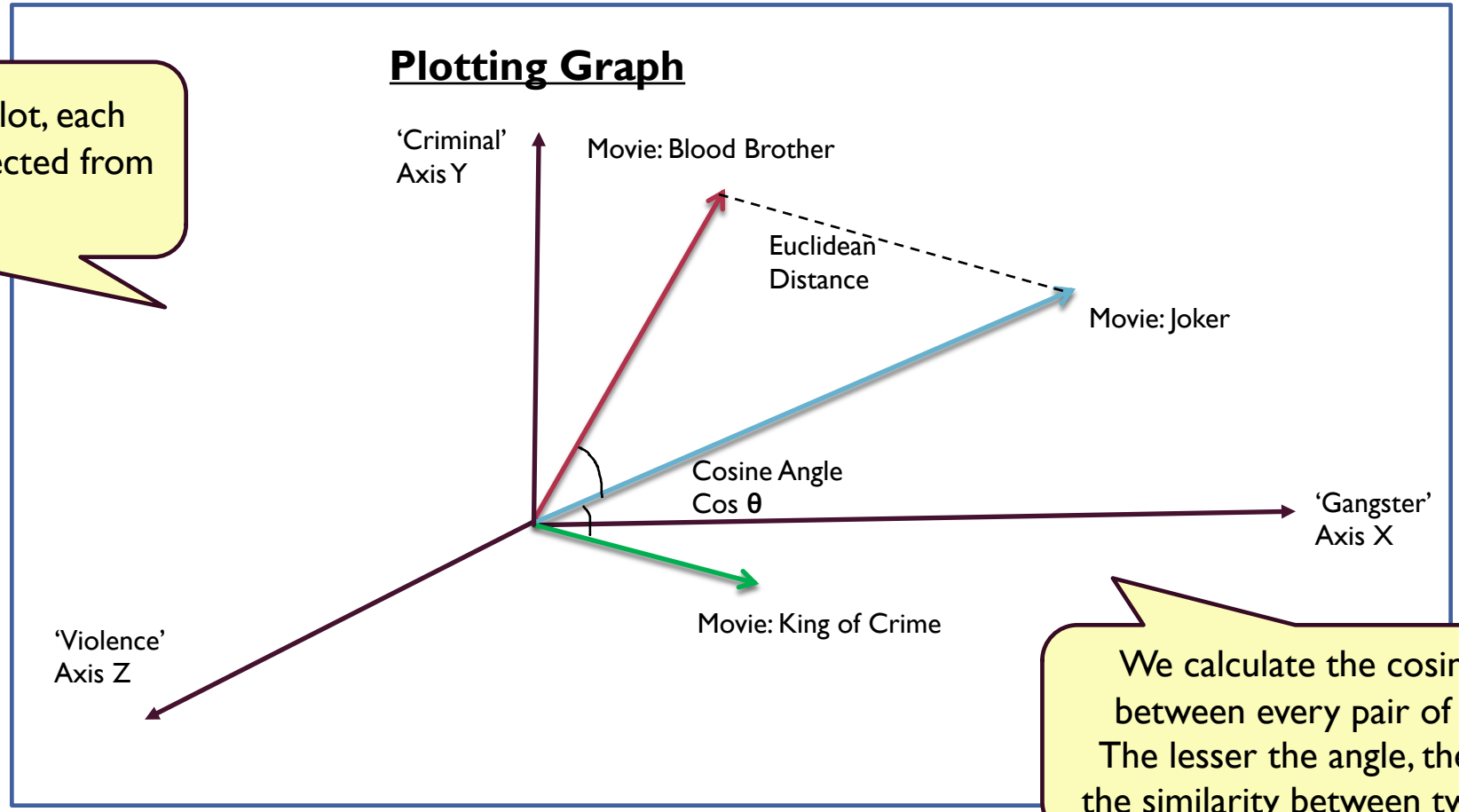
3. Adding edges to network and calculate shortest path from input movie

Shortest path: 0.1
Recommended Movie: Movie 5



The second model is based on cosine similarity and matches movies based on genre, director, and movie overview

In the multidimensional plot, each axis represents a word selected from the movie data



We calculate the cosine angle between every pair of movies. The lesser the angle, the greater the similarity between two movies

Source : machineleaningplus.com

The third model, recommends movies based on similarity of plot by taking keywords as input

We tried two model methods:

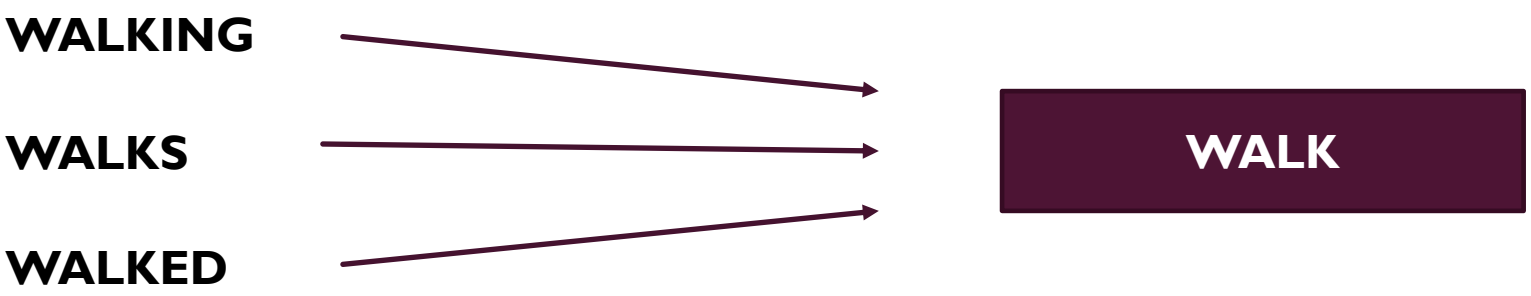
1. TF-IDF
2. Cosine Similarity

The third model, recommends movies based on similarity of plot by taking keywords as input

We tried two model methods:

1. TF-IDF
2. Cosine Similarity

Data preparation: Removing stop words, lemmatizing words (Converting words with 'inflections' to their root verbs)

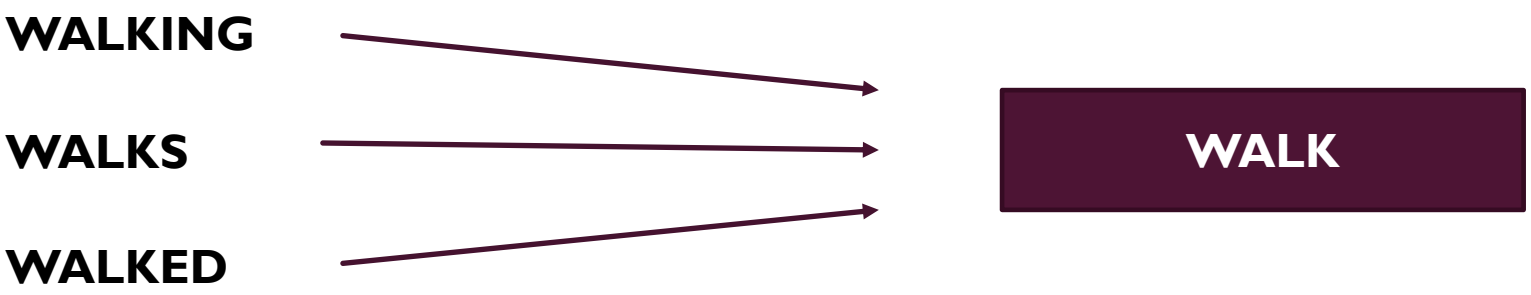


The third model, recommends movies based on similarity of plot by taking keywords as input

We tried two model methods:

1. TF-IDF
2. Cosine Similarity

Data preparation: Removing stop words, lemmatizing words (Converting words with 'inflections' to their root verbs)



Testing: It's the Holiday Season! We feel like watching Christmas movies…

So we feed the models 'Snow Christmas Holiday' 🎄 👯

Comparing the results…which is better?

TF-IDF

Movies	Score
Santa World	0.505354
Dark Horizon	0.363008
Snow Dinosaur	0.358279
Prince of Peoria A Christmas Moose Miracle	0.307477
A Christmas Movie Christmas	0.305620

Cosine Similarity

Movies	Score
Christmas Made to Order	0.375823
The Christmas Pact	0.370991
A Christmas Movie Christmas	0.368932
Jingle Around the Clock	0.344691
Christmas in Evergreen: Letters to Santa	0.339683

Comparing the results…which is better?

TF-IDF

Movies	Score
Santa World	0.505354
Dark Horizon	0.363008
Snow Dinosaur	0.358279
Prince of Peoria A Christmas Moose Miracle	0.307477
A Christmas Movie Christmas	0.305620

The recommendations seem good enough but wait…the TF-IDF model recommends 'Dark Horizon' for our search words.

Let's look at it's plot…

Cosine Similarity

Movies	Score
Christmas Made to Order	0.375823
The Christmas Pact	0.370991
A Christmas Movie Christmas	0.368932
Jingle Around the Clock	0.344691
Christmas in Evergreen: Letters to Santa	0.339683

Comparing the results…which is better?

TF-IDF: Gives importance to '**rare**' words

Movies	Score
Santa World	0.505354
Dark Horizon	0.363008
Snow Dinosaur	0.358279
Prince of Peoria A Christmas Moose Miracle	0.307477
A Christmas Movie Christmas	0.305620

The recommendations seem good enough but wait…the TF-IDF model recommends 'Dark Horizon' for our search words.

Let's look at it's plot…

"*Between here and there, we do not belong anywhere. The world collapses into* ***snow*** *and flames*"

Cosine Similarity

Movies	Score
Christmas Made to Order	0.375823
The Christmas Pact	0.370991
A Christmas Movie Christmas	0.368932
Jingle Around the Clock	0.344691
Christmas in Evergreen: Letters to Santa	0.339683

Subjectivity and data capture were the main challenges we faced…

Joker Recommendations

Once Upon a Time… in Hol… 7.5 ★

Spider-Man: Far from Home 7.6 ★

It Chapter Two 6.8 ★

We compared our model results to the TMDB website recommendations and picked the models that most aligned with their results and our preferences

Thank You