# data_challenge

February 17, 2025

## 1 Movie Recommender System Challenge

In this notebook, I, Joshua Zingale, create a movie recommender system with a subset of a kaggle dataset, which I have also included in this repository.

### 1.1 Data Loading and Cleaning

[1]:
```
import pandas as pd
pd.options.mode.copy_on_write = True

# Load only smaller subset of the movies per the instructions
df = pd.read_csv("data/movie_dataset.csv").sample(500, random_state = 115)

# View a small random sample to get a feel for the data
df.sample(3, random_state = 935)
```

[1]:
```
      index    budget                genres homepage      id  \
1879   1879  25000000                Comedy      NaN   57431
2672   2672  14000000                Comedy      NaN     864
651     651  65000000   Comedy Drama Family      NaN  196867

                                               keywords original_language  \
1879                        babysitter duringcreditsstinger                en
2672  winter trainer olympic games jamaica training …                en
651                             musical orphan foster child                en

     original_title                                           overview  \
1879     The Sitter  Noah, is not your typical entertain-the-kids-n…
2672  Cool Runnings  When a Jamaican sprinter is disqualified from …
651           Annie  Ever since her parents left her as a baby, lit…

      popularity  … runtime  \
1879   19.428994  …    81.0
2672   22.409117  …    98.0
651    33.439187  …   119.0

                                       spoken_languages    status  \
1879          [{"iso_639_1": "en", "name": "English"}]  Released
```

```
2672  [{"iso_639_1": "en", "name": "English"}, {"iso…  Released
651            [{"iso_639_1": "en", "name": "English"}]  Released

                                          tagline          title  \
1879                       Worst. Babysitter. Ever.     The Sitter
2672  One dream. Four Jamaicans. Twenty below zero.  Cool Runnings
651                           It's a Hard Knock Life          Annie

     vote_average vote_count  \
1879          5.4        325
2672          6.8        491
651           6.0        466

                                                   cast  \
1879  Sam Rockwell Jonah Hill Max Records Ari Grayno…
2672  Leon Robinson Doug E. Doug Rawle D. Lewis Mali…
651   Quvenzhan\u00e9 Wallis Jamie Foxx Rose Byrne C…

                                                   crew           director
1879  [{'name': 'Michael De Luca', 'gender': 2, 'dep…  David Gordon Green
2672  [{'name': 'Hans Zimmer', 'gender': 2, 'departm…      Jon Turteltaub
651   [{'name': 'Will Smith', 'gender': 2, 'departme…          Will Gluck

[3 rows x 24 columns]
```

```
[2]: df.columns
```

```
[2]: Index(['index', 'budget', 'genres', 'homepage', 'id', 'keywords',
            'original_language', 'original_title', 'overview', 'popularity',
            'production_companies', 'production_countries', 'release_date',
            'revenue', 'runtime', 'spoken_languages', 'status', 'tagline', 'title',
            'vote_average', 'vote_count', 'cast', 'crew', 'director'],
           dtype='object')
```

### 1.1.1 Choosing Columns

Looking at the column names and some examples, the most immediately relevant columns for this challenge seem to be “genres”, “keywords”, “original_title”, and “overview”, though other columns could be used to improve relevancy. Also, I am going to filter the data only to include those movies originally in English.

```
[3]: # Load the most relevant columns for all movies in the English language,␣
      ↪dropping any rows with NaN values
     df = df[df["original_language"] == "en"][["index", "genres", "keywords",␣
      ↪"original_title", "overview"]].dropna()

     print(len(df))
```

```
438
```

```
[4]: df.sample(3, random_state=115)
```

```
[4]:       index                  genres  \
     2980   2980  Action Horror Thriller
     3812   3812                   Drama
     4178   4178          Drama Thriller

                                        keywords  \
     2980                dystopia sequel legalized murder
     3812          corruption sex adultery television profit
     4178  baby wife husband relationship christian faith…

                   original_title  \
     2980  The Purge: Election Year
     3812                   Network
     4178             Higher Ground

                                                   overview
     2980  Two years after choosing not to kill the man w…
     3812  A TV network cynically exploits a deranged ex-…
     4178  A chronicle of one woman's lifelong struggle w…
```

### 1.1.2 Combining Columns

Fortunately, most of the data are English movies so I retained enough data. To finish cleaning the data, I now will concatenate genres, the keywords, and the film overview into a single column, which will later be turned into a vector.

```
[5]: # Build a new DataFrame with a composite "desription" column
     df["description"] = df["original_title"] + " " + df["genres"] + " " +␣
      ↪df["keywords"] + " " + df["overview"]
```

```
[6]: df.sample(3, random_state = 935)
```

```
[6]:       index                          genres  \
     867     867             Crime Drama Thriller
     2402   2402  Horror Drama Mystery Thriller
     2455   2455             Comedy Romance Drama

                                          keywords  \
     867   italy christianity new york assassination ital…
     2402  nanny haunted house channel islands parallel w…
     2455  new york wife husband relationship restaurant …

                  original_title  \
     867   The Godfather: Part III
```

```
2402               The Others
2455   When Harry Met Sally…

                                                 overview  \
867   In the midst of trying to legitimize his busin…
2402  Grace is a religious woman who lives in an old…
2455  During their travels from Chicago to New York,…

                                              description
867   The Godfather: Part III Crime Drama Thriller i…
2402  The Others Horror Drama Mystery Thriller nanny…
2455  When Harry Met Sally… Comedy Romance Drama n…
```

## 1.2 Vectorized Movie Storage

I created a vectorized database wherein each movie has an associated vector. A movie's vector is a TF-IDF vector of the text in the composite "description" field created above.

To generate a TF-IDF vector for a piece of text, I implemented a tokenizer. The tokenizer removes punctuation, sets everything to lowercase, and then attempts to get the stem of each word, e.g. by removing verb endings or plural markers. Once the text is in a normilized ("stemified") form, each word is a token; and a vector is asigned to the text based on the frequency of words present in the text and based on the inverse document frequency of each token.

I downloaded a list of English stop words from here, which I used to remove stop words from the descriptions during the tokenization stage. I added "movie" and "movies" to the list of stop words because queries of the form "I like moves that…" were resulting in irrelevant movies simply for a match to "movie".

When vectorizing a query, any word following I is removed to prevent "love" in constructions like "I love horrific war films" from biasing the results toward romantic comedies.

[7]:

```python
from nltk.stem import PorterStemmer
import numpy as np
ps = PorterStemmer()
```

[8]:

```python
# Load in the stopwords list
with open("stopwords.txt") as f:
    stopwords = f.read().split()
stopwords = set(stopwords)
```

[9]:

```python
class Tokenizer():
    def __init__(self, documents):
        """
        Initializes a tokenizer for a set of documents.
        The vocabulary for the Tokenizer is determined by the documents used to␣
 ↪initialize it.
        """
        ## Get the vocabulary
        self.vocabulary = set()
        for text in documents:
            self.vocabulary = self.vocabulary.union(set(self._stemify(text)))

        self.vocabulary_size = len(self.vocabulary)


        ## Get the stem to token id mappings
        self.stem_to_id = dict()
        self.id_to_stem = dict()

        for i, word in enumerate(self.vocabulary):
            self.stem_to_id[word] = i
            self.id_to_stem[i] = word

        ## Get the inverse document frequencies for each token
        self.idf = np.zeros(self.vocabulary_size)
        for text in documents:
            self.idf += self.frequency_vectorize(text).clip(0, 1)

        self.idf = np.log(len(documents)/self.idf)

    def tokenize(self, text: str) -> list:
        """Tokenizes input text"""
        return [self.stem_to_id[stem] for stem in self._stemify(text) if stem␣
↪in self.vocabulary]

    def vectorize(self, text: str, smoothing = 0.0) -> np.ndarray:
        """Returns a tf-idf vector for the input text, where each index␣
↪contains the tf-idf of a token in the input text.
        smoothing is the amount of smoothing added to the vector."""
        vec = np.zeros(self.vocabulary_size) + smoothing
        for token_id in self.tokenize(text):
            vec[token_id] += 1
        return vec * self.idf

    def frequency_vectorize(self, text: str) -> np.ndarray:
        """Returns a frequency vector for the input text, where each index␣
↪contains the number of appearances of a token in the input text."""
        vec = np.zeros(self.vocabulary_size)
        for token_id in self.tokenize(text):
            vec[token_id] += 1
        return vec

    def _remove_punctuation(self, text: str) -> str:
        """ Removes common punctuation from a string """
        for mark in ["!", "(", ")", ";", ":", "\"", ",", ".", "?"]:
            text = text.replace(mark, "")
        return text

    def _stemify(self, text: str) -> list:
        """ Converts text into a list of stems without punctuation"""
        text = self._remove_punctuation(text)
        words = text.lower().split()
        words = [ps.stem(word) for word in words if word not in stopwords]
        return words
```

[10]:

```
class VectorDB():
    """
    Stores documents in a vector database, wherein lookups use a vector␣
 ↪similarity metric, i.e. cosin similarity.
    """
    def __init__(self, data, embedded_row, embedding_function):
        """
        Initializes a vector database for a set of data.

        :param data: pandas DataFrame around which this database wraps
        :embedding_function: function that takes a document to an embedding␣
 ↪thereof
        """

        ## Build the vector database
        embedding_size = embedding_function(data[embedded_row].iloc[0]).size


        self.db = np.ndarray((len(data), embedding_size))
        self.data = data

        for i, document in enumerate(data[embedded_row]):
            self.db[i] = embedding_function(document)

        # normalize each db row
        self.db /= np.linalg.norm(self.db, axis = 1, keepdims = True)

    def search(self, x, k = 1, return_similarities = False):
        """Returns the top k closest matches for input vector x"""
        # normalize x
        x = x / np.linalg.norm(x)

        # Get top k
        scores = self.db @ x
        top_idc = np.argpartition(scores, -k)[-k:]
        # Sort top k
        top_idc = sorted(top_idc, key = lambda i: -scores[i])
        if return_similarities:
            return self.data.iloc[top_idc], scores[top_idc]
        return self.data.iloc[top_idc]
```

[11]:

```
# Get a Tokenizer for the data
tokenizer = Tokenizer(df.loc[:, "description"])

print(f"The vocabulary has {tokenizer.vocabulary_size} words")
```

```
The vocabulary has 5124 words
```

[12]:

```
# Create the vectorized database
db = VectorDB(df, embedded_row = "description", embedding_function = tokenizer.
 ↪vectorize)
```

[13]:

```
def vectorize_query(text):
    """Vectorizes a search query"""

    words = text.lower().split()
    new_words = [words[0]]
    # Remove any word the follows "I"
    for prev_word, word in zip(words[:-1], words[1:]):
        if prev_word != "i":
            new_words.append(word)

    text = " ".join(new_words)

    return tokenizer.vectorize(text)
```

## 1.3 Testing

[14]:

```
def search(text, k = 5):
    rows, scores = db.search(vectorize_query(text), k = k, return_similarities␣
 ↪= True)

    print("Results the following query:", text)
    for row, score in zip(rows.iloc, scores):
        title = row["original_title"]
        index = row["index"]
        overview = row["overview"]
        keywords = row["keywords"]
        genres = row["genres"]
        print(f"Title: {title} ({index})\nCosine Similarity {score}\nKeywords &␣
 ↪Genres: {keywords}, {genres}\nOverview: {overview}\n")
```

[15]:
```
search("I love thrilling action movies set in space, with a comedic twist.")
```

```
Results the following query: I love thrilling action movies set in space, with a
comedic twist.
Title: Zathura: A Space Adventure (661)
Cosine Similarity 0.13740214238710163
Keywords & Genres: adventure house alien giant robot outer space, Family Fantasy
Science Fiction Adventure
Overview: After their father is called into work, two young boys, Walter and
Danny, are left in the care of their teenage sister, Lisa, and told they must
stay inside. Walter and Danny, who anticipate a boring day, are shocked when
they begin playing Zathura, a space-themed board game, which they realize has
mystical powers when their house is shot into space. With the help of an
astronaut, the boys attempt to return home.

Title: Hard Rain (603)
Cosine Similarity 0.1253841897040928
Keywords & Genres: sheriff rain evacuation armored car crook, Thriller
Overview: Get swept up in the action as an armored car driver (Christian Slater)
tries to elude a gang of thieves (led by Morgan Freeman) while a flood ravages
the countryside. Hard Rain is "a wild, thrilling, chilling action ride" filled
with close calls, uncertain loyalties and heart-stopping heroics.

Title: Capricorn One (3668)
Cosine Similarity 0.11277092550372445
Keywords & Genres: helicopter nasa texas spacecraft beguilement, Drama Action
Thriller Science Fiction
Overview: In order to protect the reputation of the American space program, a
team of scientists stages a phony Mars landing. Willingly participating in the
deception are a trio of well-meaning astronauts, who become liabilities when
their space capsule is reported lost on re-entry. Now, with the help of a
crusading reporter,they must battle a sinister conspiracy that will stop at
nothing to keep the truth

Title: Up (66)
Cosine Similarity 0.10420413628128734
Keywords & Genres: age difference central and south america balloon animation
floating in the air, Animation Comedy Family Adventure
Overview: Carl Fredricksen spent his entire life dreaming of exploring the globe
and experiencing life to its fullest. But at age 78, life seems to have passed
him by, until a twist of fate (and a persistent 8-year old Wilderness Explorer
named Russell) gives him a new lease on life.

Title: Galaxina (3534)
Cosine Similarity 0.0858089039722489
Keywords & Genres: android harley davidson cryogenics space travel love, Comedy
Science Fiction
```

```
Overview: Galaxina is a lifelike, voluptuous android who is assigned to oversee
the operations of an intergalactic Space Police cruiser captained by incompetent
Cornelius Butt. When a mission requires the ship's crew to be placed in
suspended animation for decades, Galaxina finds herself alone for many years,
developing emotions and falling in love with the ship's pilot, Thor.
```

[16]:

```
search("I like action movies set in space")
```

```
Results the following query: I like action movies set in space
Title: Zathura: A Space Adventure (661)
Cosine Similarity 0.2652030078167088
Keywords & Genres: adventure house alien giant robot outer space, Family Fantasy
Science Fiction Adventure
Overview: After their father is called into work, two young boys, Walter and
Danny, are left in the care of their teenage sister, Lisa, and told they must
stay inside. Walter and Danny, who anticipate a boring day, are shocked when
they begin playing Zathura, a space-themed board game, which they realize has
mystical powers when their house is shot into space. With the help of an
astronaut, the boys attempt to return home.

Title: Capricorn One (3668)
Cosine Similarity 0.2176617345136039
Keywords & Genres: helicopter nasa texas spacecraft beguilement, Drama Action
Thriller Science Fiction
Overview: In order to protect the reputation of the American space program, a
team of scientists stages a phony Mars landing. Willingly participating in the
deception are a trio of well-meaning astronauts, who become liabilities when
their space capsule is reported lost on re-entry. Now, with the help of a
crusading reporter,they must battle a sinister conspiracy that will stop at
nothing to keep the truth

Title: Galaxina (3534)
Cosine Similarity 0.1656217220164086
Keywords & Genres: android harley davidson cryogenics space travel love, Comedy
Science Fiction
Overview: Galaxina is a lifelike, voluptuous android who is assigned to oversee
the operations of an intergalactic Space Police cruiser captained by incompetent
Cornelius Butt. When a mission requires the ship's crew to be placed in
suspended animation for decades, Galaxina finds herself alone for many years,
developing emotions and falling in love with the ship's pilot, Thor.

Title: Mission to Mars (373)
Cosine Similarity 0.13767166216069004
Keywords & Genres: mars spacecraft space travel alien long take, Science Fiction
Overview: When contact is lost with the crew of the first Mars expedition, a
rescue mission is launched to discover their fate.
```

```
Title: Monsters vs Aliens (67)
Cosine Similarity 0.11372259311824234
Keywords & Genres: alien giant robot duringcreditsstinger, Animation Family
Adventure Science Fiction
Overview: When Susan Murphy is unwittingly clobbered by a meteor full of outer
space gunk on her wedding day, she mysteriously grows to 49-feet-11-inches. The
military jumps into action and captures Susan, secreting her away to a covert
government compound. She is renamed Ginormica and placed in confinement with a
ragtag group of Monsters…
```

[17]:

```python
search("I like movies that are informative and teach me something")
```

```
Results the following query: I like movies that are informative and teach me
something
Title: Black Mass (877)
Cosine Similarity 0.131107487118041
Keywords & Genres: boston based on true story organized crime, Crime Drama
Overview: The true story of Whitey Bulger, the brother of a state senator and
the most infamous violent criminal in the history of South Boston, who became an
FBI informant to take down a Mafia family invading his turf.

Title: Crazy, Stupid, Love. (925)
Cosine Similarity 0.10819608049792716
Keywords & Genres: soulmates midlife crisis marriage crisis womanizer law
school, Comedy Drama Romance
Overview: Cal Weaver is living the American dream. He has a good job, a
beautiful house, great children and a beautiful wife, named Emily. Cal's
seemingly perfect life unravels, however, when he learns that Emily has been
unfaithful and wants a divorce. Over 40 and suddenly single, Cal is adrift in
the fickle world of dating. Enter, Jacob Palmer, a self-styled player who takes
Cal under his wing and teaches him how to be a hit with the ladies.

Title: Honey (2337)
Cosine Similarity 0.09731245536804409
Keywords & Genres: new york dancing hip-hop dream dance, Romance Music Family
Overview: Honey Daniels (Jessica Alba) dreams of making a name for herself as a
hip-hop choreographer. When she's not busy hitting downtown clubs with her
friends, she teaches dance classes at a nearby community center in Harlem, N.Y.,
as a way to keep kids off the streets. Honey thinks she's hit the jackpot when
she meets a hotshot director (David Moscow) who casts her in one of his music
videos. But, when he starts demanding sexual favors from her, Honey makes a
decision that will change her life.

Title: The Firm (1116)
Cosine Similarity 0.08316471519063105
Keywords & Genres: fbi law tennessee lawyer law firm, Drama Mystery Thriller
Overview: Mitch McDeere is a young man with a promising future in Law. About to
```

```
sit his Bar exam, he is approached by 'The Firm' and made an offer he doesn't
refuse. Seduced by the money and gifts showered on him, he is totally oblivious
to the more sinister side of his company. Then, two Associates are murdered. The
FBI contact him, asking him for information and suddenly his life is ruined. He
has a choice - work with the FBI, or stay with the Firm. Either way he will lose
his life as he knows it. Mitch figures the only way out is to follow his own
plan…

Title: On the Waterfront (4432)
Cosine Similarity 0.08191917021459776
Keywords & Genres: murder suspense union dock longshoreman, Crime Drama
Overview: Terry Malloy dreams about being a prize fighter, while tending his
pigeons and running errands at the docks for Johnny Friendly, the corrupt boss
of the dockers union. Terry witnesses a murder by two of Johnny's thugs, and
later meets the dead man's sister and feels responsible for his death. She
introduces him to Father Barry, who tries to force him to provide information
for the courts that will smash the dock racketeers.
```

```
[18]: search("I like calm documentaries about nature.")
```

```
Results the following query: I like calm documentaries about nature.
Title: Give Me Shelter (4660)
Cosine Similarity 0.1681818266225291
Keywords & Genres: helping animals, Documentary
Overview: Give Me Shelter is a documentary to raise awareness for important
animal issues around the world. This film uncovers the most prevalent issues in
the animal world through the eyes of individuals dedicating their lives to them
daily.

Title: The Horse Whisperer (713)
Cosine Similarity 0.16392059011182014
Keywords & Genres: love triangle new york montana attachment to nature
confidence, Drama Romance
Overview: Based on the novel by the same name from Nicholas Evans, the talented
Robert Redford presents this meditative family drama set in the country side.
Redford not only directs but also stars in the roll of a cowboy with a magical
talent for healing.

Title: Wordplay (4520)
Cosine Similarity 0.15831009727168782
Keywords & Genres: competition documentary contest crossword puzzle, Documentary
Overview: From the masters who create the mind-bending diversions to the tense
competition at the American Crossword Puzzle Tournament, Patrick Creadon's
documentary reveals a fascinating look at a decidedly addictive pastime. Creadon
captures New York Times editor Will Shortz at work, talks to celebrity solvers
-- including Bill Clinton and Ken Burns -- and presents an intimate look at the
national tournament and its competitors.
```

```
Title: Roger & Me (4713)
Cosine Similarity 0.1469336147466958
Keywords & Genres: capitalism economics unemployment corporate greed,
Documentary History
Overview: A documentary about the closure of General Motors' plant at Flint,
Michigan, which resulted in the loss of 30,000 jobs. Details the attempts of
filmmaker Michael Moore to get an interview with GM CEO Roger Smith.

Title: The Relic (759)
Cosine Similarity 0.12402640549203663
Keywords & Genres: chicago based on novel monster museum pile of dead bodies,
Horror Mystery Thriller
Overview: A researcher at Chicago's Natural History Museum returns from South
America with some crates containing his findings. When the crates arrive at the
museum without the owner there appears to be very little inside. However, police
discover gruesome murders on the cargo ship that brought the crates to the US
and then another murder in the museum itself.
```

[19]:

```
search("I like calm documentaries about war.")
```

```
Results the following query: I like calm documentaries about war.
Title: Give Me Shelter (4660)
Cosine Similarity 0.24095242118775184
Keywords & Genres: helping animals, Documentary
Overview: Give Me Shelter is a documentary to raise awareness for important
animal issues around the world. This film uncovers the most prevalent issues in
the animal world through the eyes of individuals dedicating their lives to them
daily.

Title: Wordplay (4520)
Cosine Similarity 0.22680929326386495
Keywords & Genres: competition documentary contest crossword puzzle, Documentary
Overview: From the masters who create the mind-bending diversions to the tense
competition at the American Crossword Puzzle Tournament, Patrick Creadon's
documentary reveals a fascinating look at a decidedly addictive pastime. Creadon
captures New York Times editor Will Shortz at work, talks to celebrity solvers
-- including Bill Clinton and Ken Burns -- and presents an intimate look at the
national tournament and its competitors.

Title: Roger & Me (4713)
Cosine Similarity 0.21051032051486893
Keywords & Genres: capitalism economics unemployment corporate greed,
Documentary History
Overview: A documentary about the closure of General Motors' plant at Flint,
Michigan, which resulted in the loss of 30,000 jobs. Details the attempts of
filmmaker Michael Moore to get an interview with GM CEO Roger Smith.
```

```
Title: We Were Soldiers (579)
Cosine Similarity 0.1861381455027855
Keywords & Genres: vietnam veteran missile vietnam war army major, Action
History War
Overview: The story of the first major battle of the American phase of the
Vietnam War and the soldiers on both sides that fought it.

Title: Censored Voices (4597)
Cosine Similarity 0.16626624324089007
Keywords & Genres: woman director, History Documentary
Overview: The 1967 'Six-Day' war ended with Israel's decisive victory;
conquering Jerusalem, Gaza, Sinai and the West Bank. It is a war portrayed, to
this day, as a righteous undertaking - a radiant emblem of Jewish pride. One
week after the war, a group of young kibbutzniks, led by renowned author Amos
Oz, recorded intimate conversations with soldiers returning from the
battlefield. The recording revealed an honest look at the moment Israel turned
from David to Goliath. The Israeli army censored the recordings, allowing the
kibbutzniks to publish only a fragment of the conversations. 'Censored Voices'
reveals the original recordings for the first time.
```

```
[20]: search("I love comedies for the family")
```

```
Results the following query: I love comedies for the family
Title: August: Osage County (1885)
Cosine Similarity 0.184359662874574
Keywords & Genres: suicide drug addiction funeral dysfunctional family based on
play, Comedy Drama
Overview: A look at the lives of the strong-willed women of the Weston family,
whose paths have diverged until a family crisis brings them back to the Midwest
house they grew up in, and to the dysfunctional woman who raised them.

Title: Post Grad (2619)
Cosine Similarity 0.13244780221747893
Keywords & Genres: career family unemployment woman director graduation speech,
Comedy
Overview: Ryden Malby has a master plan. Graduate college, get a great job, hang
out with her best friend and find the perfect guy. But her plan spins
hilariously out of control when she’s forced to move back home with her
eccentric family.

Title: The Royal Tenenbaums (1710)
Cosine Similarity 0.1313249277804484
Keywords & Genres: forgiveness child prodigy terminal illness dysfunctional
family cigarette smoking, Comedy Drama
Overview: An estranged family of former child prodigies reunites when their
father announces he has a terminal illness.
```

```
Title: Neighbors (2312)
Cosine Similarity 0.09990164592593007
Keywords & Genres: alcohol baby party family fraternity, Comedy
Overview: A couple with a newborn baby face unexpected difficulties after they
are forced to live next to a fraternity house.

Title: Meet Dave (778)
Cosine Similarity 0.08668132818749304
Keywords & Genres: new york captain starships new love earth, Comedy Science
Fiction Adventure Family
Overview: A crew of miniature aliens operate a spaceship that has a human form.
While trying to save their planet, the aliens encounter a new problem, as their
ship becomes smitten with an Earth woman.
```

[21]:

```
search("I love comedies")
```

```
Results the following query: I love comedies
Title: Dry Spell (4781)
Cosine Similarity 0.13474827472071382
Keywords & Genres: dating divorce sex scene sex comedy anti romantic comedy,
Comedy Romance
Overview: Sasha tries to get her soon-to-be ex husband Kyle laid so she can move
on with her sex life guilt-free.

Title: Dumb and Dumber To (1171)
Cosine Similarity 0.10993042811701818
Keywords & Genres: friendship sequel road movie buddy comedy, Comedy
Overview: 20 years after the dimwits set out on their first adventure, they head
out in search of one of their long lost children in the hope of gaining a new
kidney.

Title: The Salon (4241)
Cosine Similarity 0.09109911344825801
Keywords & Genres: independent film, Comedy
Overview: A Beauty shop owner finds romance as she struggles to save her
business.

Title: Bandits (534)
Cosine Similarity 0.08048798673807012
Keywords & Genres: prison, Action Comedy Crime Romance
Overview: Two bank robbers fall in love with the girl they've kidnapped.

Title: Money Talks (1854)
Cosine Similarity 0.0784026982777355
Keywords & Genres: prison diamant liberation of prisoners transport of prisoners
interview, Action Adventure Comedy
```

```
Overview: Money Talks is a 1997 American comedy film directed by Brett Ratner.
Sought by police and criminals, a small-time huckster makes a deal with a TV
newsman for protection.
```

```
[22]: search("I like horror films that take place in the wild")
```

```
Results the following query: I like horror films that take place in the wild
Title: Black Snake Moan (2608)
Cosine Similarity 0.2214723689312577
Keywords & Genres: southern usa blues military service independent film, Drama
Overview: A God-fearing bluesman takes to a wild young woman who, as a victim of
childhood sexual abuse, is looking everywhere for love, but never quite finding
it.

Title: The Theory of Everything (2547)
Cosine Similarity 0.12317056138960344
Keywords & Genres: wife husband relationship biography physicist based on memoir
stephen hawking, Drama Romance
Overview: The Theory of Everything is the extraordinary story of one of the
world’s greatest living minds, the renowned astrophysicist Stephen Hawking, who
falls deeply in love with fellow Cambridge student Jane Wilde.

Title: Where the Wild Things Are (293)
Cosine Similarity 0.11762624358179284
Keywords & Genres: children's book igloo wolf costume swallowed whole hit with a
rock, Family Fantasy
Overview: Max imagines running away from his mom and sailing to a far-off land
where large talking beasts -- Ira, Carol, Douglas, the Bull, Judith and
Alexander -- crown him as their king, play rumpus, build forts and discover
secret hideaways.

Title: Urban Legends: Final Cut (2576)
Cosine Similarity 0.11512644818373664
Keywords & Genres: film making high school sequel serial killer slasher, Horror
Overview: The making of a horror movie takes on a terrifying reality for
students at the most prestigious film school in the country in 'Urban Legends:
Final Cut', the suspenseful follow up to the smash hit 'Urban Legend'. At Alpine
University, someone is determined to win the best film award at any cost - even
if it means eliminating the competition. No one is safe and everyone is a
suspect. 'Urban Legends: Final Cut' is an edge-of-your-seat thriller that will
keep you guessing until the shocking climax.

Title: A Guy Thing (2189)
Cosine Similarity 0.1074538473861435
Keywords & Genres: infidelity bachelor blackmail fantasy truth, Comedy Romance
Overview: Paul Morse is a good guy. When his friends throw him a wild bachelor
party, he just wants to keep his conscience clean -- which is why he's shocked
```

when he wakes up in bed with a beautiful girl named Becky and can't remember the night before. Desperate to keep his fiancée, Karen, from finding out what may or may not be the truth, he tells her a teensy lie. Soon his lies are spiraling out of control and his life is a series of comical misunderstandings.