modelling_topic_modelling

August 11, 2025

0.0.1 Topic Modelling - V1

Load reviews data

```
[41]: from pymongo import MongoClient

def get_client(uri):
    client = MongoClient(uri)
    return client
```

```
[ ]: load_dotenv(".env")  # defaults to loading a `.env` file in the current␣
 ↪directory

uri = os.getenv("MONGO_DB_URI")

client = get_client(uri)
```

```
[43]: database = client.get_database("anime_recommendation")
reviews = database.get_collection("reviews")
cursor = reviews.find()
```

```
[45]: data_lst = []
for item in cursor:
    for rev in item["reviews"]:
        data = {"anime_id": item["anime_id"], "review": rev["review"]}
        data_lst.append(data)

df = pd.DataFrame(data_lst)
```

```
[46]: df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 201853 entries, 0 to 201852
Data columns (total 2 columns):
 #   Column    Non-Null Count   Dtype
---  ------    --------------   -----
 0   anime_id  201853 non-null  int64
 1   review    201853 non-null  object
dtypes: int64(1), object(1)
memory usage: 3.1+ MB
```

[73]:
```
df.isna().sum()
```

[73]:
```
anime_id            0
review              0
processed_review    0
dtype: int64
```

Remove stopwords

[ ]:
```
import os

import pandas as pd
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.decomposition import TruncatedSVD
from sklearn.metrics.pairwise import cosine_similarity
import nltk
from nltk.tokenize import word_tokenize
from nltk.corpus import stopwords
from nltk.stem import WordNetLemmatizer
import re

import joblib
```

[215]:
```
nltk.download('stopwords')
nltk.download('wordnet')
base_stop_words = set(stopwords.words('english'))
```

```
[nltk_data] Downloading package stopwords to C:\Users\Asus-
[nltk_data]     Home\AppData\Roaming\nltk_data…
[nltk_data]   Package stopwords is already up-to-date!
[nltk_data] Downloading package wordnet to C:\Users\Asus-
[nltk_data]     Home\AppData\Roaming\nltk_data…
[nltk_data]   Package wordnet is already up-to-date!
```

Pre-processing

[216]:
```
lemmatizer = WordNetLemmatizer()
```

[223]:
```
def preprocess(text):
    text = re.sub(r'[^a-zA-Z]', ' ', text).lower()
    tokens = text.split()
    tokens = [lemmatizer.lemmatize(word) for word in tokens if word not in␣
  ↪base_stop_words]
    return ' '.join(tokens)
```

[224]:
```
df['processed_review'] = df['review'].apply(preprocess)
```

[ ]:
```
from textwrap import wrap
```

Let's see how the result of the preprocess step on the reviews

[225]:
```
df.head()
```

[225]:
```
   anime_id                                             review  \
0         5  Cowboy Bebop: Knockin' on Heaven's Door is a g…
1         5  I'm never that comfortable with films of serie…
2         5  Cowboy Bebop: Knockin' on Heaven's Door was re…
3         5  It was a good follow up (in-between, actually)…
4         5  Warning : minor spoilers ahead It was a month …

                                    processed_review
0  cowboy bebop knockin heaven door great additio…
1  never comfortable film series mostly often fil…
2  cowboy bebop knockin heaven door released japa…
3  good follow actually series suggest youre plan…
4  warning minor spoiler ahead month ago finished…
```

[226]:
```
#wrap text to make it easier to read and see the effects of preprocessing
df["review"] = df["review"].apply(lambda x: "\n".join(wrap(x, width=180)))
df["processed_review"] = df["processed_review"].apply(lambda x: "\n".
 ↪join(wrap(x, width=180)))
```

[227]:
```
print(df.loc[0, "review"])
```

```
Cowboy Bebop: Knockin' on Heaven's Door is a great addition to the Cowboy Bebop
series, but no more. It is by no means a sequel, and after watching it, I found
that it's best
watched in the middle of the series, and not neccesarily at the end. If it's got
a specific place or not, that I don't know, but that's not very important at any
rate, and if
you've watched the entire series, it shouldn't be hard to mentally place it
inside the series anyway. This time, a terrorist possesses a weapon capable of
killing countless people,
and there's a bounty of 300 millionwoolongs on him; the largest bounty ever
given. Of course, this means that our heroes will chase him. And so starts the
process of gathering
information, meeting and getting to know people related to the bounty in some
way, and eventually, squaring off against him in a final fight. Oh, and throw in
a save-the-world
thing this time, and there you have the movie. Nothing really new, a formula
that's been used several times. There's also details here and there left
unexplained, and things  may
just happen for no reason at the rare occasion. Its 120 minutes might be a
little too long to some, but it never came off as boring at any point to me;
they certainly did a good
job of fleshing out those 120 minutes.   Though, that may be credited more to
the characters than the plot itself, as the movie threw some really interesting
characters at us. The
orignal cast is, well, pretty much the same as they always are, the same
```

```
characters which you (probably) got to love while watching the original series.
As for the movie
characters, we have for example Vincent, the main bad guy. He's quite the
interesting fellow, though the more I think about it, the more I can't help but
feel that I've experienced
his type somewhat before - he's got a mysterious past; a forgotten love
included, he's going to kill loads of people for no good reason, and he blathers
out sentences about
religion and whatnot. Nevertheless, he comes off as an interesting character,
mostly because of him being similar to Spike - both in physical prowess and
their considering
themselves 'dead' men due to past events. Then we have Electra, Vincent's past
love once forgotten. She remembers him though, and well, she wants him to
remember her as well. We
can see where that's heading…  The animation quality is superb; its detail and
overall quality is unmistakably a work done by people who knows what they are
doing. Be it
backgrounds or landscapes, they're all top-notch. Lighting effects are good, and
more than I'd exect from something out of 2001, and the overall quality of
special effects are
great; much, much better than the original series. The character designs are the
same old, with some improvements, and they work very well with this anime and
movie. The character
motions and their fluidity are great, and the few action scenes in the movie are
done so well that I could probably learn some nice figthing moves merely from
studying them. The
coloring is the only thing that's a bit behind, but considering its age it's not
a problem. And moreso, the dulled coloring actually melds perfectly with the
style of the movie,
and helps on the movie's atmosphere.  The soundtrack is what you should expect
from the original series; awesome. Yoko Kanno does her work as she did in the
series; with an amazing
soundtrack that fits perfectly with the atmosphere of the movie and its
individual scenes, and the opening and ending themes are wonderful to listen to.
The only downside is that
there is a lot of silent scenes, where no background music is present at all.
Cowboy Bebop: Knockin' On Heaven's Door is a movie that delivers the goods, but
stops at that. It's
not marvelous, but it's great, and a must-see movie for any Cowboy Bebop fan.
```

[228]:
```python
print(df.loc[0, "processed_review"])
```

```
cowboy bebop knockin heaven door great addition cowboy bebop series mean sequel
watching found best watched middle series neccesarily end got specific place
know important rate
watched entire series hard mentally place inside series anyway time terrorist
possesses weapon capable killing countless people bounty millionwoolongs largest
bounty ever given
```

```
course mean hero chase start process gathering information meeting getting know
people related bounty way eventually squaring final fight oh throw save world
thing time movie
nothing really new formula used several time also detail left unexplained thing
may happen reason rare occasion minute might little long never came boring point
certainly good job
fleshing minute though may credited character plot movie threw really
interesting character u orignal cast well pretty much always character probably
got love watching original
series movie character example vincent main bad guy quite interesting fellow
though think help feel experienced type somewhat got mysterious past forgotten
love included going kill
load people good reason blather sentence religion whatnot nevertheless come
interesting character mostly similar spike physical prowess considering dead men
due past event electra
vincent past love forgotten remembers though well want remember well see heading
animation quality superb detail overall quality unmistakably work done people
know background
landscape top notch lighting effect good exect something overall quality special
effect great much much better original series character design old improvement
work well anime
movie character motion fluidity great action scene movie done well could
probably learn nice figthing move merely studying coloring thing bit behind
considering age problem moreso
dulled coloring actually meld perfectly style movie help movie atmosphere
soundtrack expect original series awesome yoko kanno work series amazing
soundtrack fit perfectly
atmosphere movie individual scene opening ending theme wonderful listen downside
lot silent scene background music present cowboy bebop knockin heaven door movie
delivers good stop
marvelous great must see movie cowboy bebop fan
```

Training model

[229]:
```python
# TF-IDF
tfidf_vectorizer = TfidfVectorizer(max_df=0.85, min_df=2, stop_words='english')
# Fit and transform the processed descriptions
tfidf_matrix = tfidf_vectorizer.fit_transform(df['processed_review'])
```

[230]:
```python
# LSA reduces the dimensionality of the TF-IDF matrix while preserving semantic␣
↪relationships.
# n_components determines the number of topics/latent dimensions.
num_topics = 100
lsa_model = TruncatedSVD(n_components=num_topics, random_state=42)
lsa_topic_matrix = lsa_model.fit_transform(tfidf_matrix)
```

Saving model

```
[ ]: model_bundle = {
         'tfidf_matrix': tfidf_matrix,
         'df': df,
         'lsa_model': lsa_model,
         'tfidf_vectorizer': tfidf_vectorizer
     }
```

```
[ ]: joblib.dump(model_bundle, "E:\\applied data science␣
       ↪capstone\\topics\\topic_model_v1.joblib")
```

```
[ ]: model_bundle = joblib.load("E:\\applied data science␣
       ↪capstone\\topics\\topic_model_v1.joblib")
     df = model_bundle["df"]
     lsa_model = model_bundle["lsa_model"]
     tfidf_matrix = model_bundle["tfidf_matrix"]
     tfidf_vectorizer = model_bundle["tfidf_vectorizer"]
```

Assessing Model The recommendations to be made are textual and based on the reviews. The assessment of the model will therefore include: 1. Checking if the topics make sense 2. Manually inspecting some of the recommendations given based on the prompt. Domain specific knowledge about the anime titles will be used here. 3. Comparing the cosine similarity scores - if the graph is flat, the model sees everything as similar not good. If the graph clusters most values to the lower scores and tapers towards the higher scores this is good as it recognizes the reviews which are most similar to the text given. If the graph has a lot of values closer to the higher values this indicates overfitting and results in bad recommendations

Topic Space Interpretability

```
[231]: # Display the top words for each topic
       terms = tfidf_vectorizer.get_feature_names_out()
       print("Top words for each LSA Topic:")
       for i, comp in enumerate(lsa_model.components_):
           terms_in_comp = [(terms[j], comp[j]) for j in comp.argsort()[-10:][::-1]]
           print(f"Topic {i+1}: {', '.join([t[0] for t in terms_in_comp])}")
       print("\n" + "="*50 + "\n")
```

```
Top words for each LSA Topic:
Topic 1: anime, character, story, like, really, episode, good, series, season,
time
Topic 2: movie, film, series, scene, animation, ghibli, original, shinkai, hour,
minute
Topic 3: season, series, arc, second, new, previous, episode, better, fight,
titan
Topic 4: anime, season, movie, watch, manga, really, watched, watching, good,
amazing
Topic 5: series, story, character, art, great, anime, manga, amazing, overall,
episode
Topic 6: really, good, character, pretty, art, sound, story, overall, enjoyment,
bad
Topic 7: series, episode, manga, naruto, movie, arc, like, bad, read, filler
Topic 8: series, girl, love, romance, comedy, school, life, cute, really, slice
Topic 9: film, really, manga, like, episode, think, read, love, series, ending
Topic 10: film, series, harem, manga, good, mc, girl, que, pretty, isekai
Topic 11: game, series, really, film, sao, kirito, online, player, watch, anime
Topic 12: que, la, en, el, episode, se, lo, los, una, como
Topic 13: manga, game, que, really, read, love, la, life, adaptation, story
Topic 14: episode, game, manga, romance, film, character, plot, comedy, girl,
main
Topic 15: naruto, arc, girl, filler, love, fight, comedy, episode, game, hunter
Topic 16: story, mc, love, art, girl, harem, series, isekai, film, sound
Topic 17: amp, quot, girl, episode, sound, art, rsquo, music, animation, voice
Topic 18: isekai, mc, episode, comedy, life, world, great, slice, watch, fun
Topic 19: really, amp, arc, quot, girl, rsquo, mc, harem, world, anime
Topic 20: amp, quot, naruto, rsquo, romance, like, character, game, mc, boruto
Topic 21: arc, good, romance, comedy, sao, story, amp, kirito, pretty, bad
Topic 22: love, amazing, great, character, mc, romance, comedy, amp, scene,
fight
Topic 23: romance, good, love, naruto, story, action, comedy, ending, fight,
plot
Topic 24: feel, like, mc, romance, animation, scene, music, felt, naruto, sport
Topic 25: like, episode, story, world, fight, demon, feel, love, action, main
Topic 26: good, mc, life, slice, school, sport, bad, character, ending, episode
Topic 27: sao, kirito, art, naruto, asuna, life, online, sword, sound, really
Topic 28: story, really, comedy, scene, mc, fight, clannad, sport, ecchi, make
Topic 29: love, isekai, good, sport, violet, animation, bad, world, pretty, dont
Topic 30: watch, fight, watching, sport, life, slice, pretty, scene, romance,
time
Topic 31: gate, stein, time, fate, life, girl, clannad, okabe, fight, slice
Topic 32: plot, girl, violet, make, watch, good, feel, cute, scene, want
Topic 33: ending, isekai, fate, clannad, game, zero, felt, end, scene, death
Topic 34: great, sport, harem, isekai, sao, school, team, dont, review, kirito
Topic 35: romance, plot, watch, dont, like, death, really, geass, arc, note
Topic 36: romance, great, think, dont, fate, people, lot, mc, im, pretty
Topic 37: gundam, violet, fate, war, geass, code, good, zero, death, mecha
Topic 38: plot, love, great, gundam, life, story, original, slice, bad, fan
Topic 39: harem, ecchi, violet, great, life, plot, people, game, titan, review
Topic 40: fate, love, zero, night, novel, stay, think, sport, art, shirou
Topic 41: dont, violet, im, didnt, sport, thats, gate, doesnt, stein, ending
Topic 42: pretty, violet, death, school, light, note, great, demon, novel, story
Topic 43: violet, plot, romance, great, girl, evergarden, art, sport, cute,
pretty
Topic 44: pretty, amazing, sport, titan, clannad, lot, think, story, best,
isekai
Topic 45: pretty, harem, ending, make, watch, demon, feel, great, life, romance
Topic 46: violet, watch, school, ending, evergarden, pretty, high, gate, song,
stein
```

```
Topic 47: titan, pretty, feel, fate, attack, amazing, gundam, time, eren, art
Topic 48: school, demon, feel, gundam, slayer, think, geass, make, high, code
Topic 49: plot, demon, amazing, clannad, fate, sport, world, slayer, original,
watched
Topic 50: gundam, demon, titan, slayer, isekai, ending, goblin, sport, cute, arc
Topic 51: music, clannad, pretty, review, plot, world, action, sport, geass,
scene
Topic 52: clannad, dragon, titan, ball, feel, original, review, novel, watch,
pretty
Topic 53: gundam, music, thing, mystery, amazing, make, harem, demon, death,
light
Topic 54: geass, code, lelouch, make, isekai, lot, titan, violet, time, voice
Topic 55: titan, hero, music, attack, think, quite, eren, fun, fate, song
Topic 56: world, music, animation, geass, dragon, fan, ecchi, titan, code, demon
Topic 57: clannad, world, bad, interesting, watching, art, boring, think, quite,
titan
Topic 58: mystery, think, dragon, time, clannad, watching, ball, geass,
interesting, pretty
Topic 59: review, clannad, quite, mystery, bit, love, world, time, fun,
interesting
Topic 60: fairy, tail, time, lot, people, ending, feel, fan, bad, ghoul
Topic 61: time, hunter, think, evangelion, art, ecchi, angel, watch, like, ghoul
Topic 62: fairy, tail, hunter, scene, gundam, think, cute, make, animation,
pretty
Topic 63: fan, service, fun, hero, clannad, watching, good, mystery, half, cute
Topic 64: amazing, scene, people, cute, relationship, animation, felt, hero,
gundam, child
Topic 65: think, little, bit, world, bad, fan, hero, action, service, quot
Topic 66: original, hero, time, new, evangelion, novel, light, romance, pretty,
fight
Topic 67: watch, original, bit, evangelion, fairy, half, tail, end, quite,
shinji
Topic 68: half, voice, second, think, thing, fun, cute, like, dub, acting
Topic 69: interesting, quite, fan, make, review, ghoul, amazing, service, hero,
kill
Topic 70: quot, time, amazing, animation, kill, la, novel, thing, quite,
watching
Topic 71: ghoul, harem, animation, think, tokyo, lot, original, watching,
action, amazing
Topic 72: quot, review, interesting, half, cute, lot, funny, second, watched,
best
Topic 73: half, second, kill, la, rsquo, watching, amazing, style, main, fun
Topic 74: animation, gintama, bit, make, fun, death, little, know, kill, isekai
Topic 75: bad, vampire, review, kill, way, loved, different, felt, original, fun
Topic 76: action, people, amazing, cute, time, bebop, development, school,
boring, cowboy
Topic 77: gintama, half, better, watching, way, vampire, novel, style, fan,
ending
```

```
Topic 78: interesting, novel, lot, vampire, quot, madoka, end, fan, know,
magical
Topic 79: fight, main, end, ghoul, people, cute, little, bit, feel, voice
Topic 80: quite, guy, novel, say, ghoul, moment, la, piece, main, funny
Topic 81: main, thing, animation, make, felt, death, lot, style, review, ecchi
Topic 82: idol, fun, interesting, end, best, new, ecchi, scene, bad, song
Topic 83: action, angel, main, moment, say, idol, nice, beat, guy, mob
Topic 84: animation, hero, half, second, bad, unique, loved, ecchi, different,
know
Topic 85: make, idol, better, mob, watched, loved, power, cute, psycho, sense
Topic 86: ecchi, say, moment, interesting, people, angel, bit, little, vampire,
beat
Topic 87: interesting, say, novel, gintama, year, la, evangelion, action, felt,
bebop
Topic 88: mob, original, way, world, psycho, action, guy, music, fan, best
Topic 89: better, vampire, bit, hero, angel, monster, best, human, evangelion,
fairy
Topic 90: angel, beat, loved, subaru, way, protagonist, make, novel, little, bit
Topic 91: felt, quite, boring, vampire, new, guy, people, battle, original,
pokemon
Topic 92: watched, moment, goblin, thing, main, fan, got, action, vampire, mob
Topic 93: definitely, gintama, fun, thing, music, end, guy, hunter, girl,
interesting
Topic 94: better, goblin, boring, mob, way, ecchi, want, gintama, know, idol
Topic 95: say, want, loved, idol, original, recommend, goblin, end, quite,
favorite
Topic 96: piece, work, development, sound, watched, thing, girl, liked, fight,
little
Topic 97: goblin, new, felt, say, slayer, make, bad, perfect, mob, unique
Topic 98: enjoy, unique, end, watched, development, style, enjoyed, idol,
vampire, moment
Topic 99: kind, way, piece, nice, little, say, scene, main, better, different
Topic 100: felt, want, nice, style, kind, going, liked, know, piece, watched

==================================================
```

Relevance Consistency Check and Similarity Distribution Plot

```
[232]: def recommend_from_query_text(query_text, df_items, tfidf_vec, tfidf_matrix,␣
        ↪lsa_model, num_recommendations=5):
           processed_query = preprocess(query_text)
           query_tfidf = tfidf_vec.transform([processed_query])
           query_lsa = lsa_model.transform(query_tfidf)

           similarities = cosine_similarity(query_lsa, lsa_model.
        ↪transform(tfidf_matrix)).flatten()
```

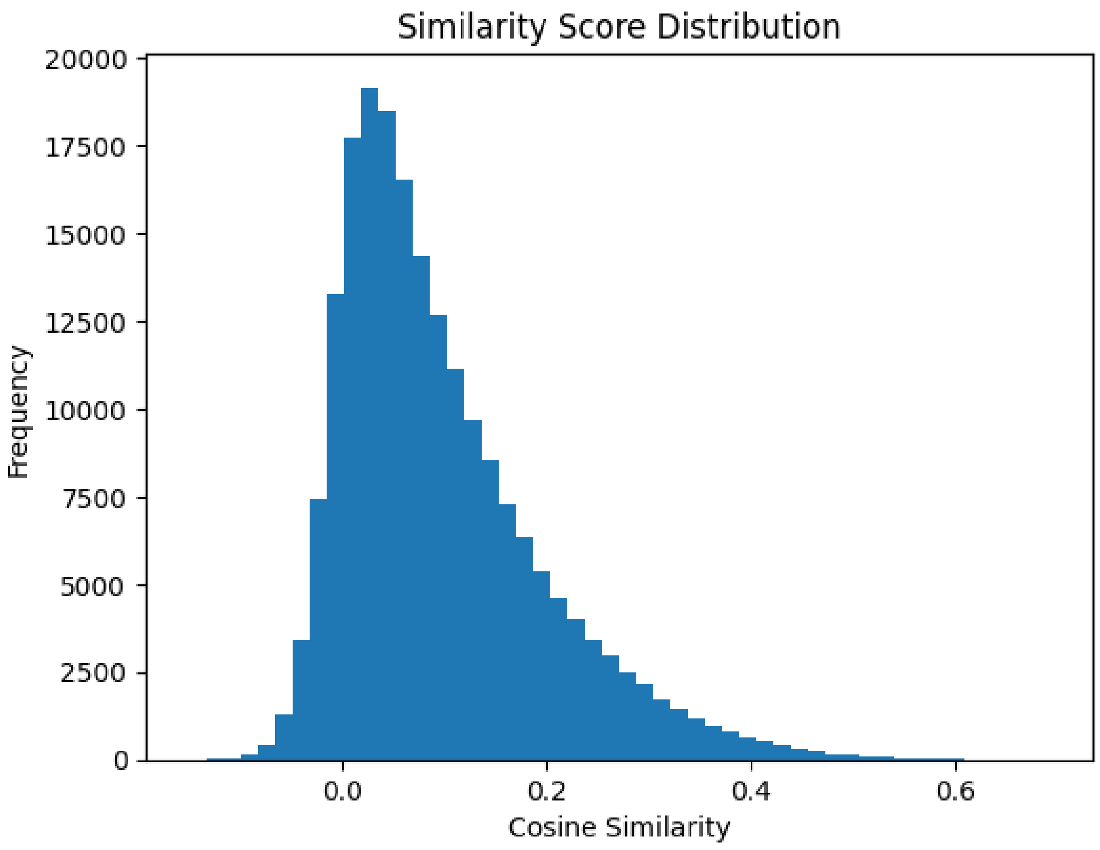
```
    plt.hist(similarities, bins=50)
    plt.title("Similarity Score Distribution")
    plt.xlabel("Cosine Similarity")
    plt.ylabel("Frequency")
    plt.show()

    top_indices = similarities.argsort()[-num_recommendations:][::-1]

    recommended_items = df_items.iloc[top_indices]
    recommended_items['similarity_score'] = similarities[top_indices]

    return recommended_items[['anime_id', 'review', "processed_review",␣
↪'similarity_score']]
```

[233]:
```
# load anime df
anime_df = pd.read_csv("E:\\applied data science␣
↪capstone\\data\\combined\\anime-for-db-27-Jun\\anime_list_27_Jun.csv")
```

[234]:
```
# sample queries
query_1 = "action packed series about super powers"
query_2 = "romantic comedy about a high school couple"
query_3 = "a young boy pursuing his dream of playing sports"
query_4 = "people get trapped in a game they were playing"
query_5 = "science made to be fun and easy to learn"
query_6 = "cute girls doing cute stuff"
query_7 = "fast cars pulling off crazy stunts"
query_8 = "a whole lot of guns and shooting"
query_9 = "germophobe navigating life"
query_10 = "pop idols entertaining their fans"
```

1. action packed series about super powers

[235]:
```
recommendations_new_query = recommend_from_query_text(query_1, df,␣
↪tfidf_vectorizer, tfidf_matrix, lsa_model, num_recommendations=10)
```



```
C:\Users\Asus-Home\AppData\Local\Temp\ipykernel_20432\122375758.py:17:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  recommended_items['similarity_score'] = similarities[top_indices]
```

[236]:
```
recommendations_new_query.head()
```

[236]:
```
        anime_id                                             review  \
13661        287  A very old fashion fighting anime series very …
24743        967  Ken is basically Bruce Lee on steroids living …
114111     29758  Want an anime about 'Murica? You got it! This …
28601       1519  Please note this review covers both TV seasons…
43870       3655  This one's a little tricky for me to think abo…

                                         processed_review  similarity_score
13661   old fashion fighting anime series heavy pure a…          0.692255
```

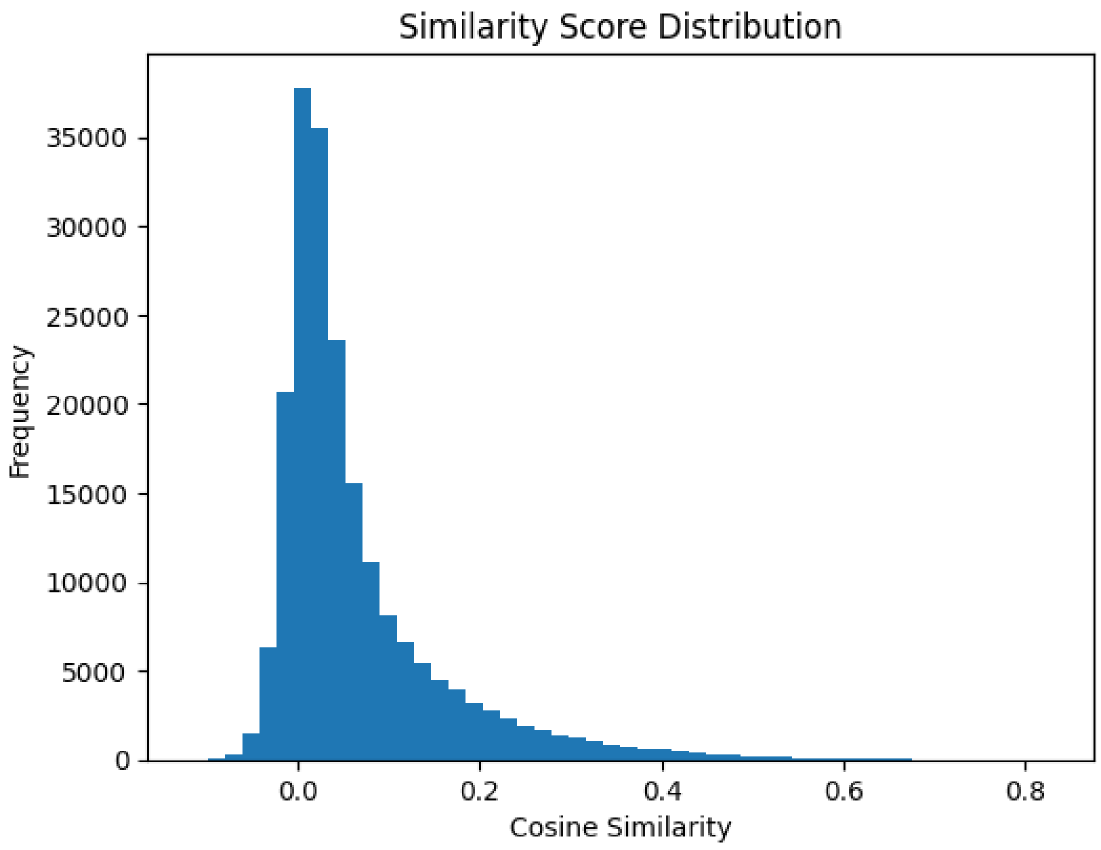
```
24743   ken basically bruce lee steroid living mad max…          0.691463
114111  want anime murica got show american going japa…          0.689831
28601   please note review cover tv season black lagoo…          0.670802
43870   one little tricky think surface nabari ou yet …          0.664046
```

[237]:
```
recommendation_ids = set(recommendations_new_query["anime_id"].unique())
anime_df.loc[anime_df["anime_id"].isin(recommendation_ids), ["anime_id",␣
↪"title", "synopsis"]]
```

[237]:
```
      anime_id                              title  \
212        287                   Baki the Grappler
550        889                        Black Lagoon
607        967                Fist of the North Star
841       1519  Black Lagoon: The Second Barrage
1641      3407                         Blassreiter
1704      3655                        Nabari no Ou
2828     10294  Towanoquon: The Ephemeral Petals
3608     20787                        Black Bullet
4017     29758                        Taboo Tattoo
4991     37521                        Vinland Saga

                                             synopsis
212   Ever since he was born, Baki Hanma has always …
550   hin Thailand is Roanapur, a depraved, crime-ri…
607   In the year 19XX, after being betrayed and lef…
841   okurou "Rock" Okajima has joined the Lagoon Co…
1641  odern Germany is plagued by an outbreak of "Am…
1704  Silent, apathetic, yet mischievous, 14-year-ol…
2828  This story follows a boy named Quon and his fr…
3608  In the year 2021, a parasitic virus known as "…
4017  Seigi, a martial arts trained middle schooler,…
4991  Young Thorfinn grew up listening to the storie…
```

2. romantic comedy about a high school couple

[238]:
```
recommendations_new_query = recommend_from_query_text(query_2, df,␣
↪tfidf_vectorizer, tfidf_matrix, lsa_model, num_recommendations=10)
```



```
C:\Users\Asus-Home\AppData\Local\Temp\ipykernel_20432\122375758.py:17:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  recommended_items['similarity_score'] = similarities[top_indices]
```

[239]:
```
recommendations_new_query.head()
```

[239]:
```
        anime_id                                             review  \
5932          66  Well despite that Azumanga Daioh is about a st…
76355      11843  With my own high school experience in mind, an…
157963     37999  Kaguya-sama: Love is War is a romantic comedy …
115888     30240  A prison gay review of Prison School Prison Sc…
2101          24  If you're looking for a documentary to sate yo…

                                          processed_review  similarity_score
5932    well despite azumanga daioh story friend attem…          0.827615
```

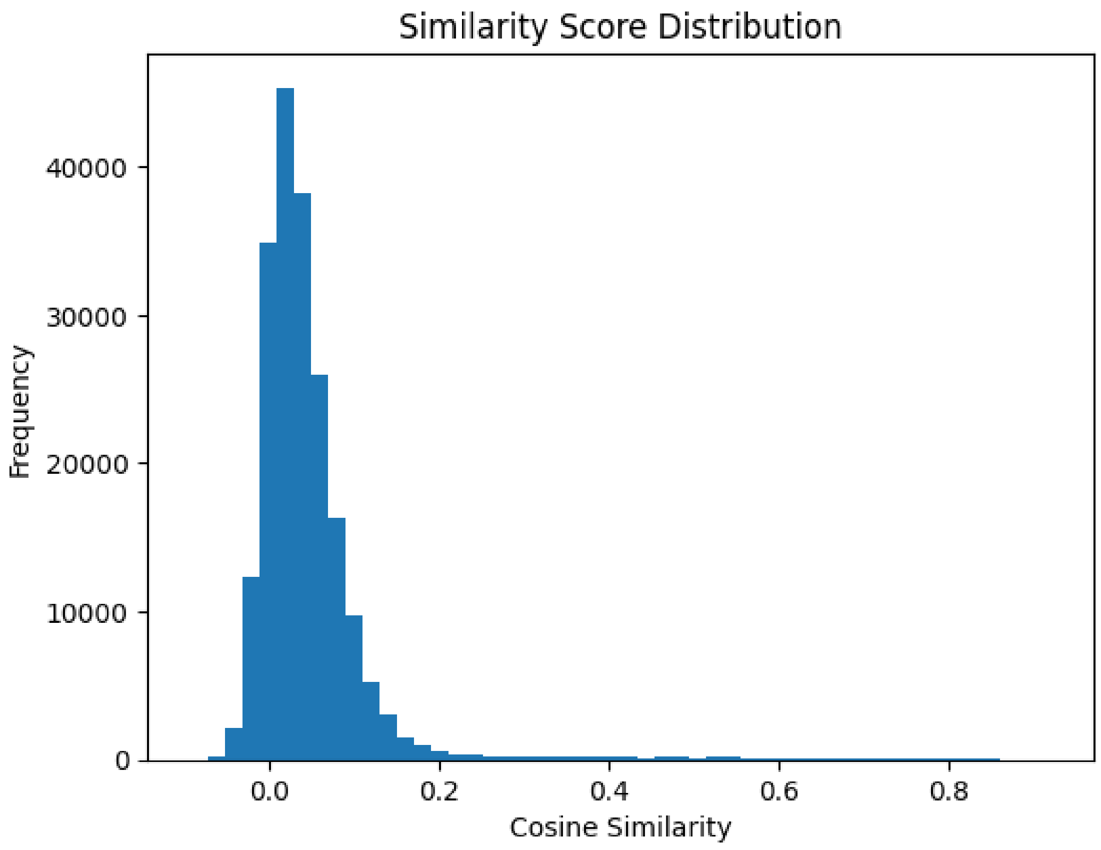
```
76355   high school experience mind watching high scho…         0.825274
157963  kaguya sama love war romantic comedy focused t…         0.824761
115888  prison gay review prison school prison school …         0.818548
2101    looking documentary sate curiosity obscure phe…         0.813565
```

[240]:
```
recommendation_ids = set(recommendations_new_query["anime_id"].unique())
anime_df.loc[anime_df["anime_id"].isin(recommendation_ids), ["anime_id",␣
↪"title", "synopsis"]]
```

[240]:
```
      anime_id                            title  \
14          24                     School Rumble
44          66    Azumanga Daioh: The Animation
1079      2034                   Lovely Complex
2990     11843  Daily Lives of High School Boys
3169     14813    My Teen Romantic Comedy SNAFU
4079     30240                    Prison School
4564     34281               High School DxD Hero
5071     37999           Kaguya-sama: Love is War

                                               synopsis
14    Just the words "I love you," and everything ch…
44    Chiyo Mihama begins her high school career as …
1079  ove is unusual for Koizumi Risa and Ootani Ats…
2990  oaming the halls of the all-boys Sanada North …
3169  Hachiman Hikigaya is an apathetic high school …
4079  ocated on the outskirts of Tokyo, Hachimitsu P…
4564             The fourth season of High School DxD .
5071  he renowned Shuchiin Academy, Miyuki Shirogane…
```

3. a young boy pursuing his dream of playing sports

[241]:
```
recommendations_new_query = recommend_from_query_text(query_3, df,␣
↪tfidf_vectorizer, tfidf_matrix, lsa_model, num_recommendations=10)
```



```
C:\Users\Asus-Home\AppData\Local\Temp\ipykernel_20432\122375758.py:17:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  recommended_items['similarity_score'] = similarities[top_indices]
```

[242]:
```
recommendations_new_query.head()
```

[242]:
```
        anime_id                                             review  \
111390     28391  I liked to call this one the "Reverse Gender S…
186005     44274  First review just had to. First off Japan beat…
76278      11771  If you wanted your Haikyuu!! fix after finishi…
201112     49052  Many sports anime come and go over the years, …
182743     42395  Kabaddi is a contact team sport which involves…

                                         processed_review  similarity_score
111390  liked call one reverse gender sport anime seri…          0.920439
```

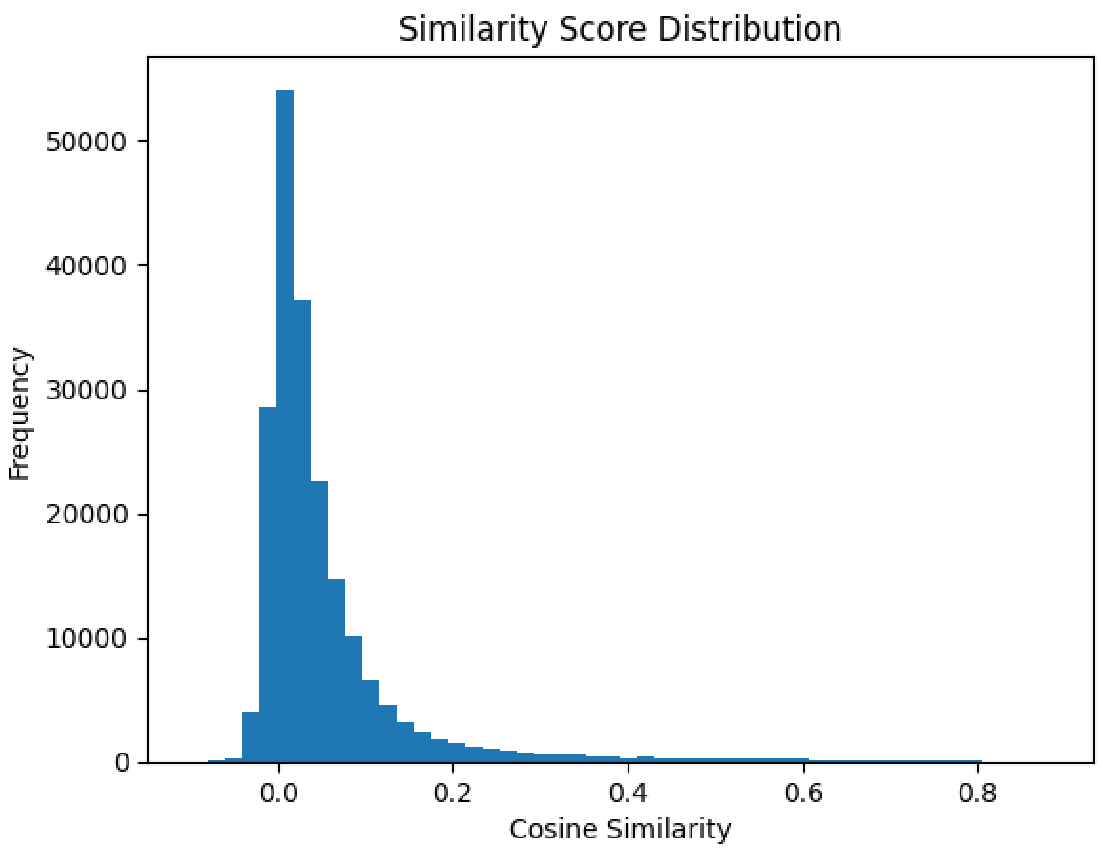
```
186005  first review first japan beating team canada r…         0.913579
76278   wanted haikyuu fix finishing care sport take p…         0.911041
201112  many sport anime come go year barely staying m…         0.909249
182743  kabaddi contact team sport involves material e…         0.906465
```

[243]:
```
recommendation_ids = set(recommendations_new_query["anime_id"].unique())
anime_df.loc[anime_df["anime_id"].isin(recommendation_ids), ["anime_id",␣
↪"title", "synopsis"]]
```

[243]:
```
      anime_id                               title  \
526        857                            Air Gear
2982     11771                 Kuroko's Basketball
3956     28391  Aokana: Four Rhythm Across the Blue
5565     42395                     Burning Kabaddi
5585     42774               Farewell, My Dear Cramer
5658     44274              PuraOra! PRIDE OF ORANGE
5814     49052                              Aoashi

                                               synopsis
526   Trecks, also known as AT, are motorized and fu…
2982  Teikou Junior High School's basketball team is…
3956  h the invention of anti-gravitational shoes kn…
5565  First-year high schooler Tatsuya Yoigoshi, the…
5585  h no soccer accomplishments to speak of during…
5658  The story takes place in Nikko city, Tochigi P…
5814  In a quiet rural town, the spotlight of a loca…
```

4. people get trapped in a game they were playing

[244]:
```
recommendations_new_query = recommend_from_query_text(query_4, df,␣
↪tfidf_vectorizer, tfidf_matrix, lsa_model, num_recommendations=10)
```



```
C:\Users\Asus-Home\AppData\Local\Temp\ipykernel_20432\122375758.py:17:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  recommended_items['similarity_score'] = similarities[top_indices]
```

[245]:
```
recommendations_new_query.head()
```

[245]:
```
        anime_id                                             review  \
189908     48441  I really liked this show, however I'm not sure…
189914     48441  As a reference, I have played the original two…
194455     50205  Arknights : Prelude to dawn is an anime adapta…
140187     34933  Listen, I've watched this show twice, and beli…
182292     42307  The World End With You is a game that original…

                                         processed_review  similarity_score
189908  really liked show however sure much one would …          0.883596
```

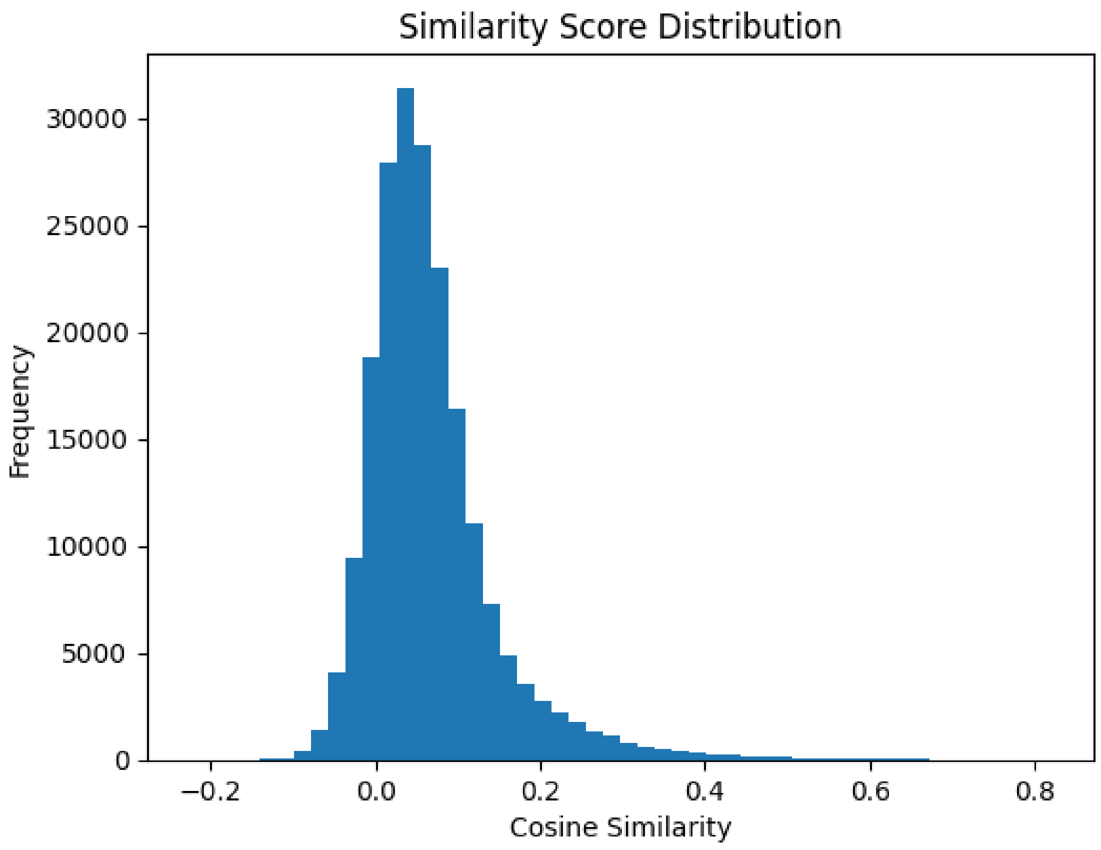
```
189914  reference played original two game series show…         0.883448
194455  arknights prelude dawn anime adaptation game n…         0.880112
140187  listen watched show twice believe appearance d…         0.872387
182292  world end game originally came nd game masterp…         0.871310
```

[246]:
```
recommendation_ids = set(recommendations_new_query["anime_id"].unique())
anime_df.loc[anime_df["anime_id"].isin(recommendation_ids), ["anime_id",␣
↪"title", "synopsis"]]
```

[246]:
```
      anime_id                                              title  \
447        658                                              Akagi
4392     33023                          Touken Ranbu - Hanamaru
4652     34933                                          Kakegurui
5184     38790  BOFURI: I Don't Want to Get Hurt, so I'll Max …
5556     42307               The World Ends with You The Animation
5660     44276  Full Dive: The Ultimate Next-Gen Full Dive RPG…
5759     48441  The Legend of Heroes: Trails of Cold Steel - N…
5888     50205                Arknights Animation: Prelude to Dawn

                                               synopsis
447   hile mahjong is a game that is often played wi…
4392  In the year 2205, a special sage known as Sani…
4652  Unlike many schools, attending Hyakkaou Privat…
5184  fter an enthusiastic invitation from her frien…
5556  Neku Sakuraba, a 15-year-old boy with a hobby …
5660  In an unexpected turn of events, dull high sch…
5759  Eiyuu Densetsu: Sen no Kiseki centers around R…
5888  The discovery of a black crystalline mineral k…
```

5. science made to be fun and easy to learn

[247]:
```
recommendations_new_query = recommend_from_query_text(query_5, df,␣
↪tfidf_vectorizer, tfidf_matrix, lsa_model, num_recommendations=10)
```



```
C:\Users\Asus-Home\AppData\Local\Temp\ipykernel_20432\122375758.py:17:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  recommended_items['similarity_score'] = similarities[top_indices]
```

[248]:
```
recommendations_new_query.head()
```

[248]:
```
        anime_id                                             review  \
123793     31953  This anime is just a delight to watch. Sure it…
11058        223  Speed review: Pretty much everything is good. …
183664     42923  This show is honestly just fun, at the current…
118856     31157  Classicaloids is. A lot to take in. It's not g…
183752     42923  Its been a while since Ive watched anime. Ok, …

                                         processed_review  similarity_score
123793  anime delight watch sure remarkable revolution…          0.819687
```

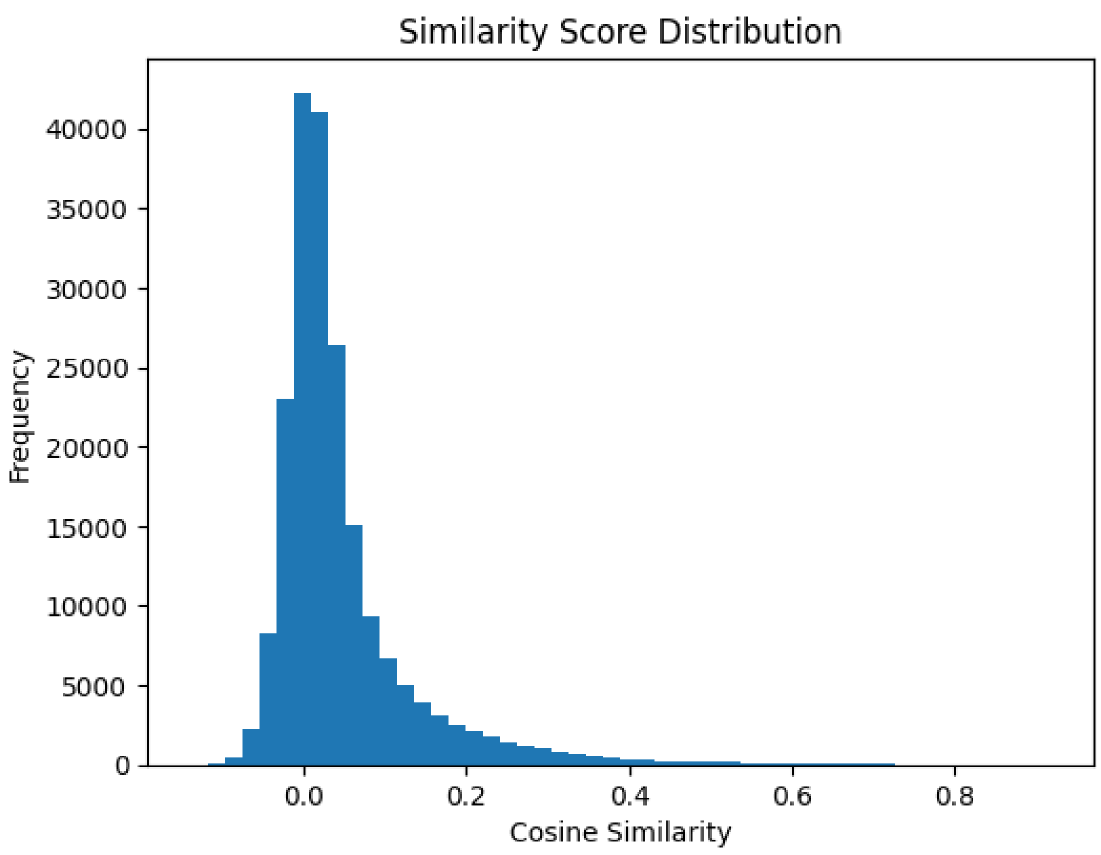
```
11058   speed review pretty much everything good enjoy…          0.811543
183664  show honestly fun current episode count give m…          0.800381
118856  classicaloids lot take good fun extremely fun …          0.778501
183752  since ive watched anime ok thats entirely true…          0.775002
```

[249]:
```python
recommendation_ids = set(recommendations_new_query["anime_id"].unique())
anime_df.loc[anime_df["anime_id"].isin(recommendation_ids), ["anime_id",␣
↪"title", "synopsis"]]
```

[249]:
```
      anime_id                                        title  \
162        223                                   Dragon Ball
2245      6347                Baka & Test - Summon the Beasts
2715      9756                      Puella Magi Madoka Magica
3416     18095                                        No-Rin
3769     23289                     Monthly Girls' Nozaki-kun
4160     31157                                   ClassicaLoid
4267     31953                                     New Game!
5535     42072  She Professed Herself Pupil of the Wise Man
5599     42923                              SK8 the Infinity

                                               synopsis
162   Gokuu Son is a young boy who lives in the wood…
2245  Fumizuki Academy isn't a typical Japanese high…
2715  adoka Kaname and Sayaka Miki are regular middl…
3416  Idol-obsessed Kousaku Hata is left devastated …
3769  Chiyo Sakura is a cheerful high school girl wh…
4160  Ever since her father used up the last of the …
4267  Since childhood, Aoba Suzukaze has loved the F…
5535  Sakimori Kagami plays a VRMMORPG called Arch E…
5599  High school student Reki Kyan is passionate ab…
```

6. cute girls doing cute stuff

[250]:
```python
recommendations_new_query = recommend_from_query_text(query_6, df,␣
↪tfidf_vectorizer, tfidf_matrix, lsa_model, num_recommendations=10)
```



```
C:\Users\Asus-Home\AppData\Local\Temp\ipykernel_20432\122375758.py:17:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  recommended_items['similarity_score'] = similarities[top_indices]
```

[251]:
```
recommendations_new_query.head()
```

[251]:
```
        anime_id                                             review  \
165380     39324  Ok the 1st reason that makes me rate this anim…
157726     37993  A lot of this reviewers are really quick to sa…
70349      10495  Best CGDCT (cute girls doing cute things). I w…
52501       5680  K-On! is my comfort anime and it never fails t…
142940     35756  If you just want to see cute girls doing cute …

                                         processed_review  similarity_score
165380  ok st reason make rate anime low latina cute l…          0.917949
```

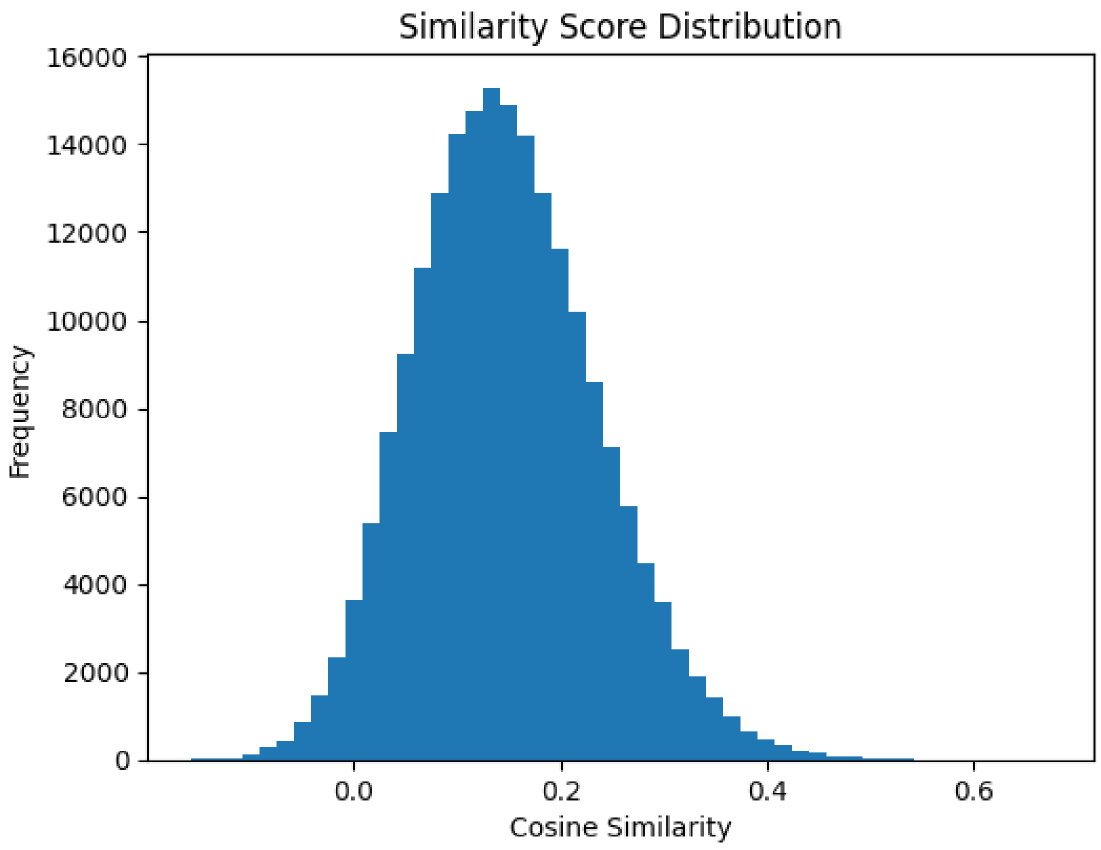
```
157726  lot reviewer really quick say cosplay cute stu…          0.911360
70349   best cgdct cute girl cute thing speak general …          0.907592
52501   k comfort anime never fails make smile definit…          0.902175
142940  want see cute girl cute stuff mainly drawing m…          0.885544
```

[252]:
```
recommendation_ids = set(recommendations_new_query["anime_id"].unique())
anime_df.loc[anime_df["anime_id"].isin(recommendation_ids), ["anime_id",␣
↪"title", "synopsis"]]
```

[252]:
```
      anime_id                                              title  \
2136      5680                                              K-ON!
2855     10495                            YuruYuri: Happy Go Lily
4617     34618                                            BLEND-S
4635     34798                                     Laid-Back Camp
4694     35241                                     Konohana Kitan
4745     35756                                        Comic Girls
5069     37993                  WataTen! An Angel Flew Down to Me
5238     39324  If It's for My Daughter, I'd Even Defeat a Dem…

                                               synopsis
2136  fresh high school year always means much to co…
2855  fter a year in grade school without her childh…
4617  shing to be independent, 16-year-old Maika Sak…
4635  hile the perfect getaway for most girls her ag…
4694  In a bustling village of spirits, Yuzu, a chee…
4745  Kaoruko "Chaos" Moeta is a young manga artist …
5069  College student Miyako Hoshino is quite shy ar…
5238  Eighteen-year-old Dale Reki is a skilled, kind…
```

7. fast cars pulling off crazy stunts

[253]:
```
recommendations_new_query = recommend_from_query_text(query_7, df,␣
↪tfidf_vectorizer, tfidf_matrix, lsa_model, num_recommendations=10)
```



```
C:\Users\Asus-Home\AppData\Local\Temp\ipykernel_20432\122375758.py:17:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  recommended_items['similarity_score'] = similarities[top_indices]
```

[254]:
```
recommendations_new_query.head()
```

[254]:
```
       anime_id                                             review  \
24831       974  Dead Leaves… How can I even describe thi…
93592     18679  I usually never write reviews, but this anime …
24816       974  I thought I had seen it all, then I saw Dead L…
23849       889  What with the whole crippling recession thing …
24006       889  Guns are blazing and bodies start to drop like…

                                        processed_review  similarity_score
24831  dead leaf even describe anime animated product…          0.674803
```

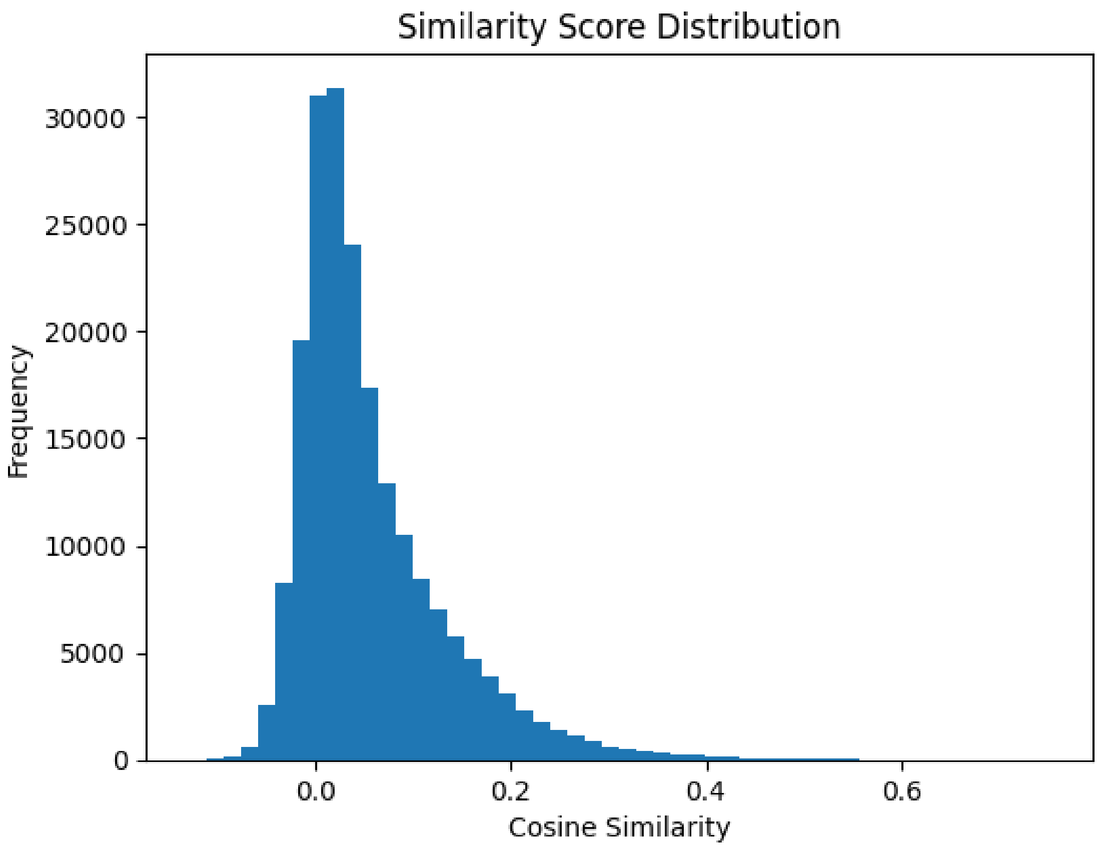
```
93592  usually never write review anime many thing ri…          0.614657
24816  thought seen saw dead leaf preface mind hiroyu…          0.612024
23849  whole crippling recession thing massive black …          0.607125
24006  gun blazing body start drop like fly left righ…          0.606095
```

[255]:
```
recommendation_ids = set(recommendations_new_query["anime_id"].unique())
anime_df.loc[anime_df["anime_id"].isin(recommendation_ids), ["anime_id",␣
 ↪"title", "synopsis"]]
```

[255]:
```
      anime_id                                  title  \
550        889                           Black Lagoon
613        974                            Dead Leaves
2288      6675                                Redline
3471     18679                           Kill la Kill
3613     20899  JoJo's Bizarre Adventure: Stardust Crusaders
3856     25183                                Gangsta.

                                             synopsis
550   hin Thailand is Roanapur, a depraved, crime-ri…
613   Pandy and Retro awaken naked on Earth with no …
2288  Every five years, an exhilarating race called …
3471  fter the murder of her father, Ryuuko Matoi ha…
3613  Years after an ancient evil was salvaged from …
3856  Nicholas Brown and Worick Arcangelo, known in …
```

8. a whole lot of guns and shooting

[256]:
```
recommendations_new_query = recommend_from_query_text(query_8, df,␣
 ↪tfidf_vectorizer, tfidf_matrix, lsa_model, num_recommendations=10)
```



```
C:\Users\Asus-Home\AppData\Local\Temp\ipykernel_20432\122375758.py:17:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  recommended_items['similarity_score'] = similarities[top_indices]
```

[257]:
```
recommendations_new_query.head()
```

[257]:
```
        anime_id                                             review  \
80586      12815  I watched the sub. It took a maybe 4 or so epi…
54219       6444  this anime made me cry a lot but i gain a lot …
187568     46118  This anime got a lot of hype when it was annou…
69353      10165  Nichijou is a sol with a lot of over-the-top h…
48809       4975  A rambling mess of show that confuses subtlety…

                                          processed_review  similarity_score
80586   watched sub took maybe episode really get flow…          0.751484
```

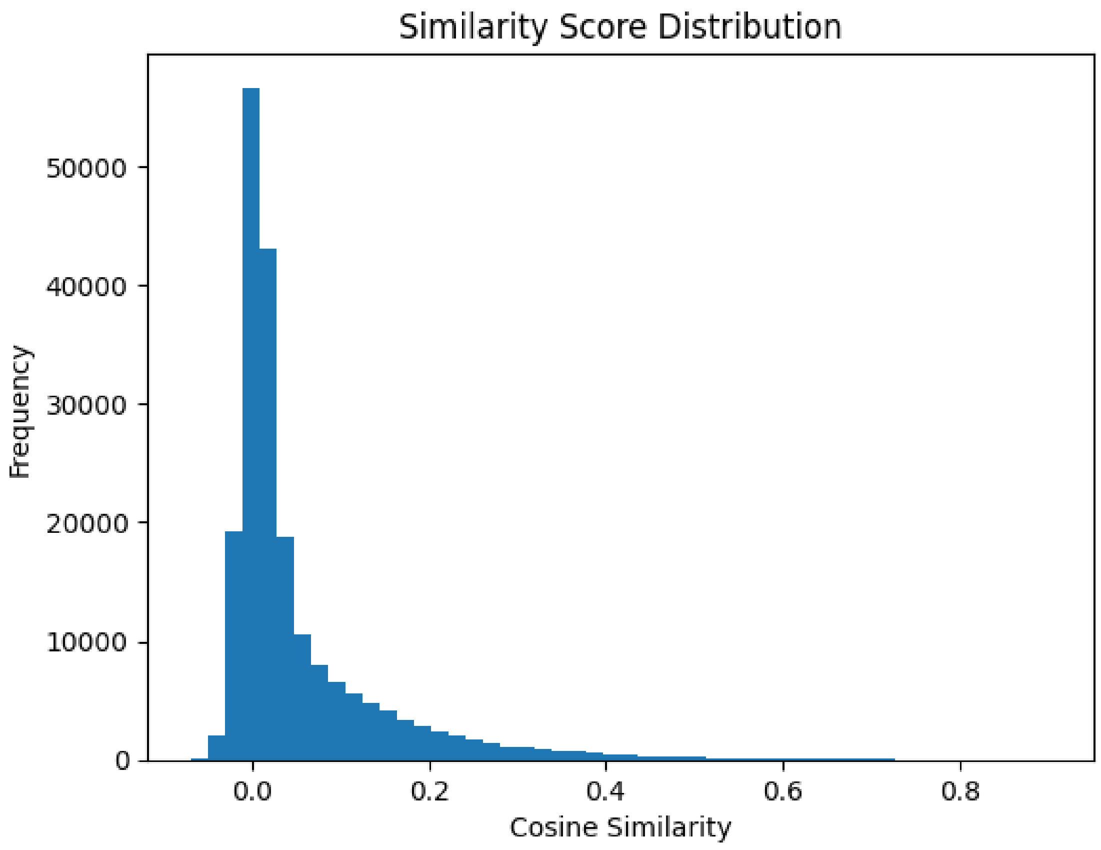
```
54219   anime made cry lot gain lot happiness learned …          0.733521
187568  anime got lot hype announced disappointed lot …          0.720121
69353   nichijou sol lot top humor lot insane stuff go…          0.700051
48809   rambling mess show confuses subtlety complexit…          0.689718
```

[258]:
```
recommendation_ids = set(recommendations_new_query["anime_id"].unique())
anime_df.loc[anime_df["anime_id"].isin(recommendation_ids), ["anime_id",␣
↪"title", "synopsis"]]
```

[258]:
```
      anime_id                         title  \
173        237                  Eureka Seven
875       1571                    Ghost Hunt
1995      4975                     ChäoS;HEAd
2254      6444     Tegami Bachi: Letter Bee
2809     10165  Nichijou - My Ordinary Life
2876     10620              The Future Diary
3061     12815               Polar Bear Cafe
5039     37828                Seven Days War
5516     41694   Cells at Work! CODE BLACK!
5707     46118  WAVE!! -Let's go surfing!!-

                                               synopsis
173   In the backwater town of Bellforest lives a 14…
875   hile at school, Taniyama Mai and her friends l…
1995  Throughout Shibuya, a series of murders dubbed…
2254  h his mother taken away from him and having lo…
2809  Nichijou primarily focuses on the daily antics…
2876  onely high school student, Yukiteru Amano, spe…
3061  Situated near the local zoo and owned by the c…
5039  amoru Suzuhara is an introverted high school s…
5516  Due to poor lifestyle choices, a certain human…
5707  asaki Hinaoka, who grew up near the coast of O…
```

9. germophobe navigating life

[259]:
```
recommendations_new_query = recommend_from_query_text(query_9, df,␣
↪tfidf_vectorizer, tfidf_matrix, lsa_model, num_recommendations=10)
```



```
C:\Users\Asus-Home\AppData\Local\Temp\ipykernel_20432\122375758.py:17:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  recommended_items['similarity_score'] = similarities[top_indices]
```

```
[260]: recommendations_new_query.head()
```

```
[260]:        anime_id                                             review  \
198978     52973  The Caf Terrace and Its Goddesses is a delight…
82705      13759  Sakurasou is one anime that shows that not eve…
88511      16417  I sat down expecting some generic moeblob slic…
194531     48675  If you don't like Slice of Life anime, don't w…
186714     44511  "CHAINSAW MAN" Genre- Action, Thriller, Fantas…

                                         processed_review  similarity_score
198978  caf terrace goddess delightful slice life anim…          0.903163
```

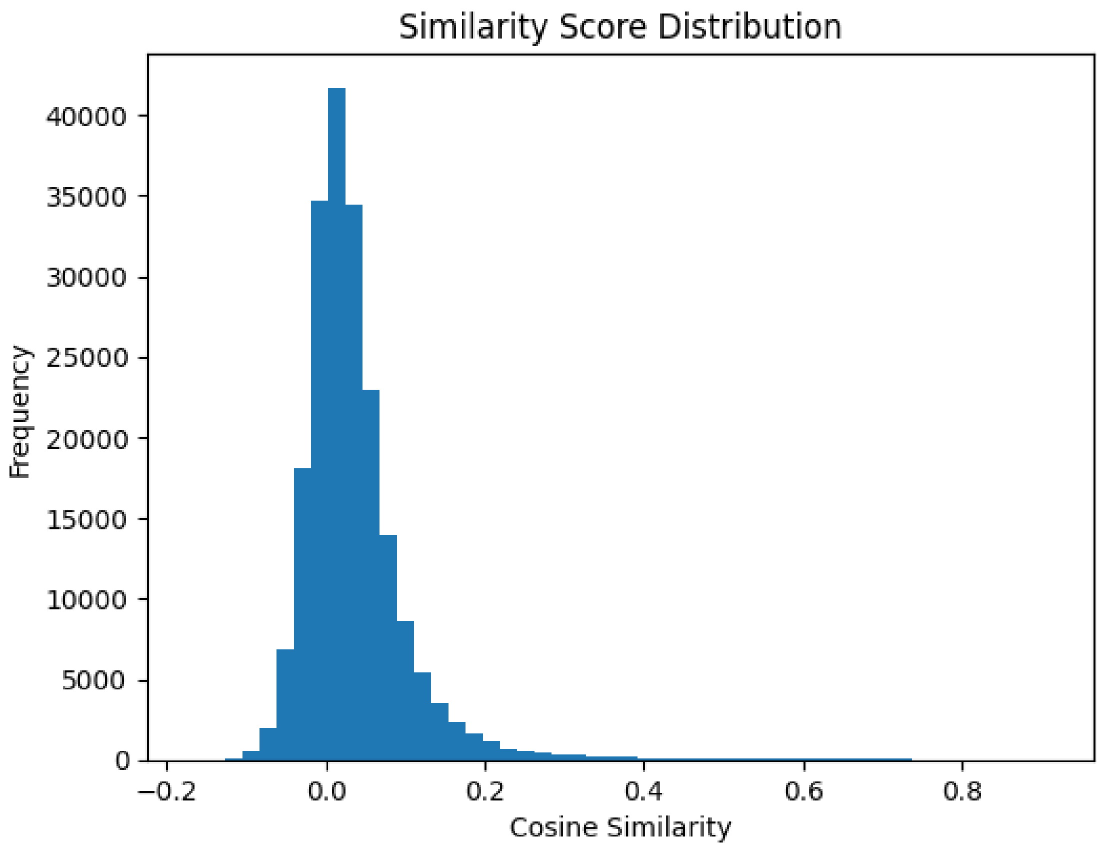
```
82705   sakurasou one anime show everything life want …         0.901897
88511   sat expecting generic moeblob slice life kind …         0.895509
194531  like slice life anime watch since one thoes bo…         0.875090
186714  chainsaw man genre action thriller fantasy sho…         0.871350
```

[261]:
```
recommendation_ids = set(recommendations_new_query["anime_id"].unique())
anime_df.loc[anime_df["anime_id"].isin(recommendation_ids), ["anime_id",␣
 ↪"title", "synopsis"]]
```

[261]:
```
      anime_id                                 title  \
2935     11179  Listen to Me, Girls. I Am Your Father!
3122     13759              The Pet Girl of Sakurasou
3261     16417                          Tamako Market
3290     16664         The Tale of the Princess Kaguya
4895     36990         Non Non Biyori Movie: Vacation
5171     38680               Fruits Basket 1st Season
5665     44511                           Chainsaw Man
5785     48675                     A Couple of Cuckoos
5996     52034                          [Oshi No Ko]
6038     52973      The Café Terrace and Its Goddesses

                                               synopsis
2935  Yuuta Segawa has just started his freshman yea…
3122  hen abandoned kittens and his good conscience …
3261  Inside the Usagiyama Shopping District lies an…
3290  Deep in the countryside, a man named Okina wor…
4895  Summer vacation is drawing to an end. When Sug…
5171  Tooru Honda has always been fascinated by the …
5665  Denji has a simple dream-to live a happy and p…
5785  Nagi Umino and Erika Amano, a studious high sc…
5996  In the entertainment world, celebrities often …
6038  After his grandmother Sachiko passes away, Hay…
```

10. pop idols entertaining their fans

[262]:
```
recommendations_new_query = recommend_from_query_text(query_10, df,␣
 ↪tfidf_vectorizer, tfidf_matrix, lsa_model, num_recommendations=10)
```



```
C:\Users\Asus-Home\AppData\Local\Temp\ipykernel_20432\122375758.py:17:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  recommended_items['similarity_score'] = similarities[top_indices]
```

[263]:
```
recommendations_new_query.head()
```

[263]:
```
        anime_id                                             review  \
156319     37890  At the start of the season I saw a new Idol an…
198483     49692  Heroines Run the Show is an interesting look a…
156844     37976  Zombieland Saga is both a great starting point…
156313     37890  It's unique, that's why it's awesome. Most of …
156307     37890  If My Favorite Pop Idol Made It to the Budokan…

                                         processed_review  similarity_score
156319  start season saw new idol anime airing another…          0.911362
```

```
198483  heroine run show interesting look idol culture…         0.911040
156844  zombieland saga great starting point anime fan…         0.889127
156313  unique awesome time idol anime would idol star…         0.887985
156307  favorite pop idol made budokan would die engli…         0.880445
```

```
[264]: recommendation_ids = set(recommendations_new_query["anime_id"].unique())
       anime_df.loc[anime_df["anime_id"].isin(recommendation_ids), ["anime_id",␣
        ↪"title", "synopsis"]]
```

```
[264]:       anime_id                                              title  \
       5043     37890  If My Favorite Pop Idol Made It to the Budokan…
       5059     37976                                    Zombie Land Saga
       5283     39609                            Dropout Idol Fruit Tart
       5852     49692  Heroines Run the Show: The Unpopular Girl and …

                                                        synopsis
       5043  girl is obsessed with her favorite idol, a min…
       5059  Sakura Minamoto dreams of becoming an idol. Un…
       5283  Fourth dormitory of the Rat Production (common…
       5852  Freshly graduating middle school, energetic 15…
```

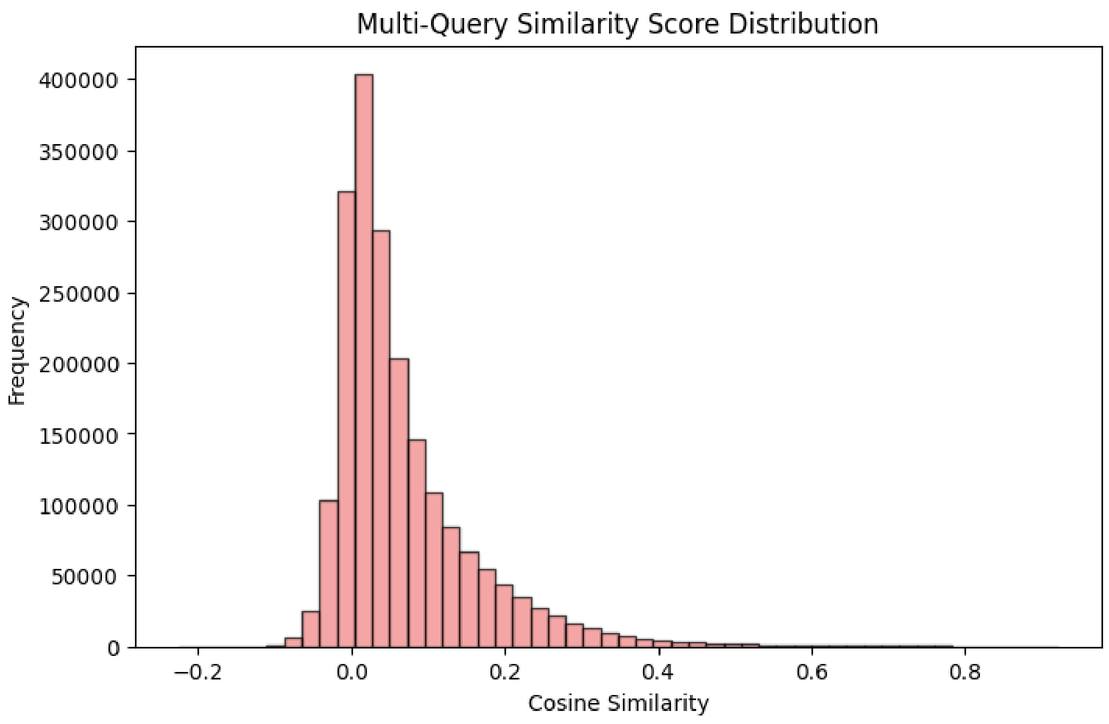
Similarity Distribution

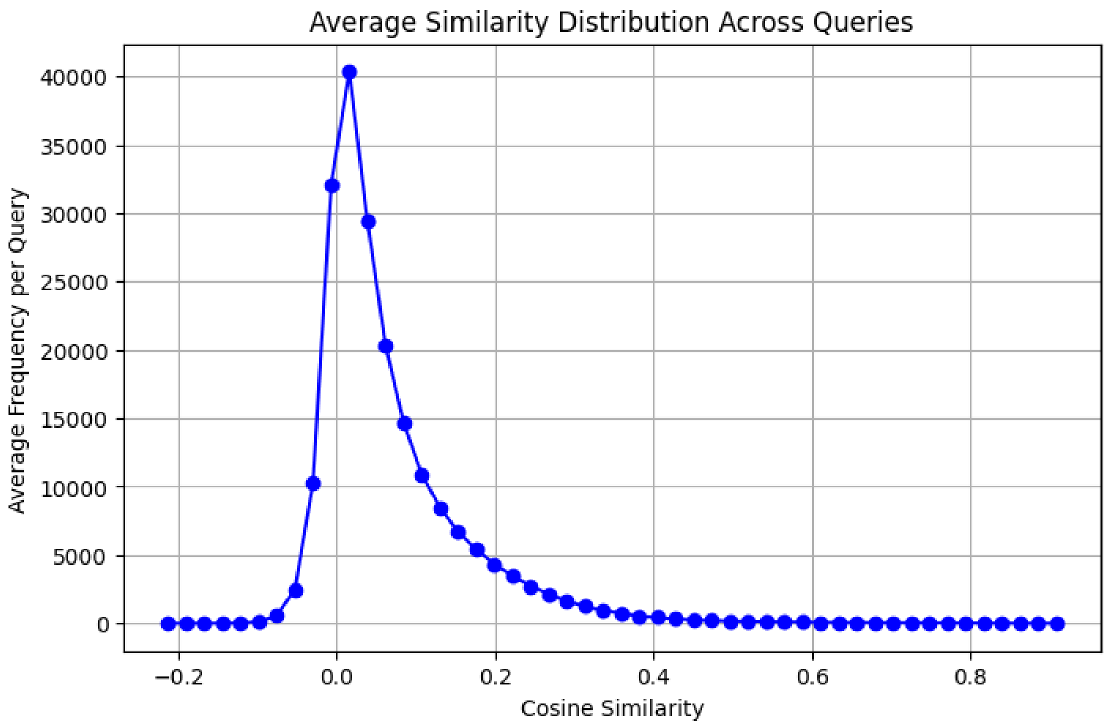
```
[265]: def plot_multiquery_similarity_distribution(query_texts, tfidf_vec,␣
        ↪tfidf_matrix, lsa_model, bins=50):
           all_similarities = []

           for query in query_texts:
               # Preprocess and transform query
               processed_query = preprocess(query)
               query_tfidf = tfidf_vec.transform([processed_query])
               query_lsa = lsa_model.transform(query_tfidf)

               # Compute similarities with all items
               sims = cosine_similarity(query_lsa, lsa_model.transform(tfidf_matrix)).
        ↪flatten()
               all_similarities.append(sims)

           # Convert to numpy array for averaging
           all_similarities = np.array(all_similarities)

           # Flatten for combined histogram
           flat_sims = all_similarities.flatten()

           # Plot histogram
           plt.figure(figsize=(8, 5))
           plt.hist(flat_sims, bins=bins, color='lightcoral', edgecolor='black',␣
        ↪alpha=0.7)
```

```
    plt.title("Multi-Query Similarity Score Distribution")
    plt.xlabel("Cosine Similarity")
    plt.ylabel("Frequency")
    plt.show()

    # Optionally plot average similarity curve across bins
    bin_counts, bin_edges = np.histogram(flat_sims, bins=bins)
    bin_centers = (bin_edges[:-1] + bin_edges[1:]) / 2
    avg_counts = bin_counts / len(query_texts)

    plt.figure(figsize=(8, 5))
    plt.plot(bin_centers, avg_counts, marker='o', color='blue')
    plt.title("Average Similarity Distribution Across Queries")
    plt.xlabel("Cosine Similarity")
    plt.ylabel("Average Frequency per Query")
    plt.grid(True)
    plt.show()
```

The model performs well over a diverse set of prompts as evidenced by the cluster of similarities around the lower scores and a tail towards the higher similarity scores.
The curve shows that the average similarities peak around the lower scores.

[266]:
```
sample_queries = [query_1, query_2, query_3, query_4, query_5, query_6,␣
 ↪query_7, query_8, query_9, query_10]
plot_multiquery_similarity_distribution(sample_queries, tfidf_vectorizer,␣
 ↪tfidf_matrix, lsa_model)
```





0.0.2 Topic Modelling - V2

Remove stopwords

[75]:
```
texts = df['review'].tolist()
all_text = " ".join(texts).lower()

# remove punctuation and numbers
clean_text = re.sub(r"[^a-z\s]", "", all_text)
```

[76]:
```
import os

import pandas as pd
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.decomposition import TruncatedSVD
from sklearn.metrics.pairwise import cosine_similarity
import nltk
from nltk.tokenize import word_tokenize
from nltk.corpus import stopwords
from nltk.stem import WordNetLemmatizer
import re
import joblib
```

```
[ ]: nltk.download('punkt')
nltk.download('punkt_tab')
nltk.download('stopwords')
nltk.download('wordnet')

#These stop words are very generic and the list will need to be expanded to␣
↪include those that are common in reviews and anime which would add no value
base_stop_words = set(stopwords.words('english'))
```

```
[nltk_data] Downloading package punkt to C:\Users\Asus-
[nltk_data]     Home\AppData\Roaming\nltk_data…
[nltk_data]   Package punkt is already up-to-date!
[nltk_data] Downloading package punkt_tab to C:\Users\Asus-
[nltk_data]     Home\AppData\Roaming\nltk_data…
[nltk_data]   Unzipping tokenizers\punkt_tab.zip.
[nltk_data] Downloading package stopwords to C:\Users\Asus-
[nltk_data]     Home\AppData\Roaming\nltk_data…
[nltk_data]   Package stopwords is already up-to-date!
[nltk_data] Downloading package wordnet to C:\Users\Asus-
[nltk_data]     Home\AppData\Roaming\nltk_data…
[nltk_data]   Package wordnet is already up-to-date!
```

```
[80]: # tokenize
tokens = word_tokenize(clean_text)

# remove base stopwords
base_stopwords = set(stopwords.words('english'))
filtered_tokens = [word for word in tokens if word not in base_stopwords]
```

```
[81]: from collections import Counter

word_freq = Counter(filtered_tokens)
```

```
[84]: common_words = word_freq.most_common(50)
for word, count in common_words:
    print(f"{word}: {count}")
```

```
anime: 627349
characters: 477957
like: 425269
story: 409315
show: 400881
one: 355578
really: 317232
character: 314188
good: 272917
```

```
series: 259298
even: 223767
well: 217350
much: 210487
first: 205506
time: 201708
also: 193819
would: 185863
get: 167674
dont: 167551
watch: 164955
art: 160044
episode: 151561
season: 149087
way: 148409
main: 145186
plot: 141059
episodes: 135744
people: 133553
great: 133196
see: 129564
think: 125426
make: 125384
love: 123269
say: 122238
still: 121659
animation: 120306
something: 115659
lot: 115077
overall: 110429
pretty: 110204
many: 109146
watching: 108708
end: 107497
feel: 107130
im: 105222
world: 104374
could: 102387
know: 101675
life: 99464
doesnt: 99002
```

[87]:
```
#add most frequent common words to stop words list
custom_anime_stop_words = set(word for word, count in common_words)
final_stop_words = base_stop_words.union(custom_anime_stop_words)
```

```
[90]: # used to store data to disk
def write_to_txt(filename, data):
    with open(filename, "w") as f:
        for word in sorted(data):
            f.write(word + "\n")
    print(f"file saved to {filename}")

def read_from_txt(filename):
    with open(filename, "r") as f:
        target = set(line.strip() for line in f if line.strip())
    print(f"contents read from {filename}")
    return target
```

```
[89]: write_to_txt("E:\\applied data science capstone\\topics\\stopwords\\v1.txt",␣
 ↪final_stop_words)
```

```
file saved to E:\applied data science capstone\topics\stopwords\v1.txt
```

```
[91]: final_stop_words = read_from_txt("E:\\applied data science␣
 ↪capstone\\topics\\stopwords\\v1.txt")
```

```
contents read from E:\applied data science capstone\topics\stopwords\v1.txt
```

Pre-processing

```
[217]: lemmatizer = WordNetLemmatizer()
```

```
[92]: def preprocess(text):
    text = re.sub(r'[^a-zA-Z]', ' ', text).lower()
    text = re.sub(r'(&quot;)', '', text)
    tokens = text.split()
    tokens = [lemmatizer.lemmatize(word) for word in tokens if word not in␣
 ↪final_stop_words]
    return ' '.join(tokens)
```

```
[93]: df['processed_review'] = df['review'].apply(preprocess)
```

```
[94]: from textwrap import wrap
```

Let's see how the result of the preprocess step on the reviews

```
[95]: df.head()
```

```
[95]:    anime_id                                             review  \
0          5  Cowboy Bebop: Knockin' on Heaven's Door is a g…
1          5  I'm never that comfortable with films of serie…
2          5  Cowboy Bebop: Knockin' on Heaven's Door was re…
3          5  It was a good follow up (in-between, actually)…
4          5  Warning : minor spoilers ahead It was a month …
```

```
                                    processed_review
0  cowboy bebop knockin heaven door addition cowb…
1  never comfortable film mostly often film consi…
2  cowboy bebop knockin heaven door released japa…
3  follow actually suggest youre planning cowboy …
4  warning minor spoiler ahead month ago finished…
```

[96]:
```
#wrap text to make it easier to read and see the effects of preprocessing
df["review"] = df["review"].apply(lambda x: "\n".join(wrap(x, width=180)))
df["processed_review"] = df["processed_review"].apply(lambda x: "\n".
 ↪join(wrap(x, width=180)))
```

[97]:
```
print(df.loc[0, "review"])
```

```
Cowboy Bebop: Knockin' on Heaven's Door is a great addition to the Cowboy Bebop
series, but no more. It is by no means a sequel, and after watching it, I found
that it's best
watched in the middle of the series, and not neccesarily at the end. If it's got
a specific place or not, that I don't know, but that's not very important at any
rate, and if
you've watched the entire series, it shouldn't be hard to mentally place it
inside the series anyway. This time, a terrorist possesses a weapon capable of
killing countless people,
and there's a bounty of 300 millionwoolongs on him; the largest bounty ever
given. Of course, this means that our heroes will chase him. And so starts the
process of gathering
information, meeting and getting to know people related to the bounty in some
way, and eventually, squaring off against him in a final fight. Oh, and throw in
a save-the-world
thing this time, and there you have the movie. Nothing really new, a formula
that's been used several times. There's also details here and there left
unexplained, and things  may
just happen for no reason at the rare occasion. Its 120 minutes might be a
little too long to some, but it never came off as boring at any point to me;
they certainly did a good
job of fleshing out those 120 minutes.   Though, that may be credited more to
the characters than the plot itself, as the movie threw some really interesting
characters at us. The
orignal cast is, well, pretty much the same as they always are, the same
characters which you (probably) got to love while watching the original series.
As for the movie
characters, we have for example Vincent, the main bad guy. He's quite the
interesting fellow, though the more I think about it, the more I can't help but
feel that I've experienced
his type somewhat before - he's got a mysterious past; a forgotten love
included, he's going to kill loads of people for no good reason, and he blathers
out sentences about
religion and whatnot. Nevertheless, he comes off as an interesting character,
mostly because of him being similar to Spike - both in physical prowess and
their considering
themselves 'dead' men due to past events. Then we have Electra, Vincent's past
love once forgotten. She remembers him though, and well, she wants him to
remember her as well. We
can see where that's heading…  The animation quality is superb; its detail and
overall quality is unmistakably a work done by people who knows what they are
doing. Be it
backgrounds or landscapes, they're all top-notch. Lighting effects are good, and
more than I'd exect from something out of 2001, and the overall quality of
special effects are
great; much, much better than the original series. The character designs are the
same old, with some improvements, and they work very well with this anime and
movie. The character
motions and their fluidity are great, and the few action scenes in the movie are
done so well that I could probably learn some nice figthing moves merely from
studying them. The
coloring is the only thing that's a bit behind, but considering its age it's not
a problem. And moreso, the dulled coloring actually melds perfectly with the
style of the movie,
and helps on the movie's atmosphere.  The soundtrack is what you should expect
from the original series; awesome. Yoko Kanno does her work as she did in the
series; with an amazing
soundtrack that fits perfectly with the atmosphere of the movie and its
individual scenes, and the opening and ending themes are wonderful to listen to.
The only downside is that
there is a lot of silent scenes, where no background music is present at all.
Cowboy Bebop: Knockin' On Heaven's Door is a movie that delivers the goods, but
stops at that. It's
not marvelous, but it's great, and a must-see movie for any Cowboy Bebop fan.
```

[98]:
```python
print(df.loc[0, "processed_review"])
```

```
cowboy bebop knockin heaven door addition cowboy bebop mean sequel found best
watched middle neccesarily got specific place important rate watched entire hard
mentally place inside
anyway terrorist possesses weapon capable killing countless bounty
millionwoolongs largest bounty ever given course mean hero chase start process
gathering information meeting
getting related bounty eventually squaring final fight oh throw save thing movie
nothing new formula used several time detail left unexplained thing may happen
reason rare occasion
minute might little long never came boring point certainly job fleshing minute
though may credited movie threw interesting u orignal cast always probably got
original movie example
vincent bad guy quite interesting fellow though help experienced type somewhat
got mysterious past forgotten included going kill load reason blather sentence
religion whatnot
```

```
nevertheless come interesting mostly similar spike physical prowess considering
dead men due past event electra vincent past forgotten remembers though want
remember heading
quality superb detail quality unmistakably work done know background landscape
top notch lighting effect exect quality special effect better original design
old improvement work
movie motion fluidity action scene movie done probably learn nice figthing move
merely studying coloring thing bit behind considering age problem moreso dulled
coloring actually
meld perfectly style movie help movie atmosphere soundtrack expect original
awesome yoko kanno work amazing soundtrack fit perfectly atmosphere movie
individual scene opening
ending theme wonderful listen downside silent scene background music present
cowboy bebop knockin heaven door movie delivers good stop marvelous must movie
cowboy bebop fan
```

Training model

[100]:
```python
# TF-IDF
tfidf_vectorizer = TfidfVectorizer(max_df=0.85, min_df=2, stop_words='english')
# Fit and transform the processed descriptions
tfidf_matrix = tfidf_vectorizer.fit_transform(df['processed_review'])
```

[101]:
```python
# LSA reduces the dimensionality of the TF-IDF matrix while preserving semantic␣
 ↪relationships.
# n_components determines the number of topics/latent dimensions.
num_topics = 100
lsa_model = TruncatedSVD(n_components=num_topics, random_state=42)
lsa_topic_matrix = lsa_model.fit_transform(tfidf_matrix)
```

Saving model

[103]:
```python
model_bundle = {
    'tfidf_matrix': tfidf_matrix,
    'df': df,
    'lsa_model': lsa_model,
    'tfidf_vectorizer': tfidf_vectorizer
}
```

[104]:
```python
joblib.dump(model_bundle, "E:\\applied data science␣
 ↪capstone\\topics\\topic_model_v2.joblib")
```

[104]:
```
['E:\\applied data science capstone\\topics\\topic_model_v2.joblib']
```

[ ]:
```python
model_bundle = joblib.load("E:\\applied data science␣
 ↪capstone\\topics\\topic_model_v2.joblib")
df = model_bundle["df"]
lsa_model = model_bundle["lsa_model"]
tfidf_matrix = model_bundle["tfidf_matrix"]
```

```
tfidf_vectorizer = model_bundle["tfidf_vectorizer"]
```

Assessing Model The recommendations to be made are textual and based on the reviews. The assessment of the model will therefore include: 1. Checking if the topics make sense 2. Manually inspecting some of the recommendations given based on the prompt. Domain specific knowledge about the anime titles will be used here. 3. Comparing the cosine similarity scores - if the graph is flat, the model sees everything as similar not good. If the graph clusters most values to the lower scores and tapers towards the higher scores this is good as it recognizes the reviews which are most similar to the text given. If the graph has a lot of values closer to the higher values this indicates overfitting and results in bad recommendations

Topic Space Interpretability

[102]:
```
# Display the top words for each topic
terms = tfidf_vectorizer.get_feature_names_out()
print("Top words for each LSA Topic:")
for i, comp in enumerate(lsa_model.components_):
    terms_in_comp = [(terms[j], comp[j]) for j in comp.argsort()[-10:][::-1]]
    print(f"Topic {i+1}: {', '.join([t[0] for t in terms_in_comp])}")
print("\n" + "="*50 + "\n")
```

```
Top words for each LSA Topic:
Topic 1: movie, thing, girl, scene, sound, bad, music, interesting, want, review
Topic 2: movie, film, ghibli, shinkai, hour, tv, minute, beautiful, original,
scene
Topic 3: film, arc, manga, action, scene, fight, naruto, theme, viewer, work
Topic 4: film, girl, school, romance, cute, comedy, harem, slice, relationship,
friend
Topic 5: game, mc, isekai, arc, fight, sao, kirito, power, hero, bad
Topic 6: manga, film, mc, girl, read, harem, isekai, bad, guy, adaptation
Topic 7: game, film, manga, sao, kirito, player, romance, video, online, play
Topic 8: que, la, en, el, se, lo, los, una, game, como
Topic 9: manga, girl, school, read, movie, adaptation, work, viewer,
relationship, cast
Topic 10: arc, naruto, comedy, girl, filler, film, funny, romance, school, game
Topic 11: comedy, isekai, mc, romance, action, scene, harem, fun, genre,
enjoyable
Topic 12: arc, romance, mc, sao, relationship, kirito, felt, clannad, second,
feeling
Topic 13: naruto, isekai, mc, game, boruto, amazing, filler, sport, sasuke, best
Topic 14: amp, mc, quot, rsquo, romance, arc, isekai, harem, beautiful, naruto
Topic 15: girl, amazing, arc, mc, music, sound, beautiful, best, song, harem
Topic 16: naruto, girl, interesting, quite, harem, felt, ending, bit, filler,
development
Topic 17: romance, fight, action, scene, sao, kirito, harem, fate, battle, titan
Topic 18: mc, sport, fight, romance, team, school, demon, felt, scene, haikyuu
Topic 19: sao, kirito, music, scene, violet, bad, asuna, sound, online, sword
Topic 20: isekai, cute, sao, hero, fun, fantasy, kirito, demon, interesting, bit
```

Topic 21: sport, harem, fan, review, team, school, ecchi, high, gundam, fate
Topic 22: fate, violet, girl, game, gate, stein, zero, fight, scene, felt
Topic 23: gate, stein, okabe, romance, travel, cute, sao, mc, interesting, kirito
Topic 24: harem, ecchi, violet, review, game, arc, hunter, mystery, evergarden, scene
Topic 25: romance, bad, isekai, sound, voice, arc, music, hunter, school, look
Topic 26: gundam, mc, original, war, mecha, romance, fan, cute, watched, geass
Topic 27: fate, zero, death, light, novel, review, note, night, cute, stay
Topic 28: sport, girl, isekai, amazing, bad, ending, best, romance, titan, death
Topic 29: titan, attack, eren, school, isekai, mystery, scene, aot, high, season
Topic 30: violet, school, evergarden, death, note, high, geass, gundam, boring, interesting
Topic 31: scene, clannad, music, fight, school, action, isekai, dragon, geass, thing
Topic 32: review, ending, gundam, comedy, school, violet, action, gintama, sound, different
Topic 33: review, demon, slayer, cute, death, geass, goblin, note, code, girl
Topic 34: demon, amazing, slayer, clannad, interesting, gundam, girl, best, seen, geass
Topic 35: clannad, titan, harem, thing, tomoya, gundam, nagisa, attack, sound, interesting
Topic 36: clannad, sport, violet, mystery, original, ending, action, dragon, horror, novel
Topic 37: gundam, demon, slayer, music, thing, arc, goblin, scene, sport, ecchi
Topic 38: scene, isekai, voice, felt, watched, sport, action, girl, dub, gintama
Topic 39: gundam, music, death, comedy, note, half, light, felt, review, second
Topic 40: music, interesting, watched, titan, hunter, idol, anime, hero, boring, geass
Topic 41: hunter, ending, fun, gundam, cute, titan, watched, definitely, bad, felt
Topic 42: hero, fun, action, bit, clannad, half, little, review, man, romance
Topic 43: original, thing, hunter, new, different, light, scene, novel, fight, hero
Topic 44: fairy, tail, thing, amazing, magic, fan, school, service, titan, bit
Topic 45: ghoul, half, second, fan, human, tokyo, service, vampire, ecchi, amazing
Topic 46: fairy, second, scene, tail, half, year, want, review, best, watched
Topic 47: hunter, bit, little, action, dragon, bebop, start, best, amazing, cowboy
Topic 48: interesting, amazing, review, quite, novel, arc, fun, gintama, romance, titan
Topic 49: hunter, amazing, mystery, new, fan, novel, old, original, comedy, bad
Topic 50: hunter, thing, voice, half, review, fairy, cute, tail, gon, mystery
Topic 51: quite, best, bit, actually, seen, fan, hero, ghoul, scene, hunter
Topic 52: original, fun, angel, new, evangelion, moment, sound, shinji, mystery, beat
Topic 53: ghoul, fun, tokyo, felt, fight, kaneki, human, harem, rsquo, best

Topic 54: cute, quot, sound, year, got, boring, felt, gintama, evangelion, style
Topic 55: fan, interesting, gintama, service, voice, best, bebop, felt, cowboy, idol
Topic 56: quot, novel, ghoul, light, hero, little, tokyo, idol, going, voice
Topic 57: angel, action, cute, beat, hero, best, amazing, ghoul, review, moment
Topic 58: quot, action, idol, liked, best, fun, quite, relationship, mob, guy
Topic 59: quite, harem, cute, amazing, new, slice, ghoul, kill, original, idol
Topic 60: quite, gintama, hunter, action, novel, sound, start, rsquo, want, piece
Topic 61: vampire, better, original, kill, action, idol, cute, la, boring, anime
Topic 62: interesting, ecchi, quite, thing, moment, quot, felt, review, watched, bad
Topic 63: felt, half, bad, evangelion, fight, mystery, la, quot, voice, second
Topic 64: vampire, idol, definitely, que, death, anime, felt, clannad, bad, different
Topic 65: idol, watched, fight, angel, bad, song, beat, thing, group, mystery
Topic 66: vampire, gintama, thing, beautiful, cute, fan, bad, quite, bleach, review
Topic 67: gintama, fun, hero, scene, watched, amazing, original, felt, music, friend
Topic 68: enjoy, quite, want, felt, mob, comedy, recommend, thing, original, psycho
Topic 69: vampire, piece, felt, watched, novel, fun, quite, filler, bleach, anime
Topic 70: better, gintama, bebop, cowboy, moment, funny, cute, hunter, mystery, goblin
Topic 71: ecchi, gintama, original, half, felt, bleach, cute, mystery, brotherhood, different
Topic 72: mob, little, watched, bit, psycho, boring, song, subaru, felt, idol
Topic 73: nice, madoka, magical, subaru, watched, enjoy, power, fun, relationship, actually
Topic 74: watched, nice, vampire, isekai, original, bebop, work, cowboy, sound, interesting
Topic 75: angel, new, guy, beat, development, gintama, ecchi, music, goblin, anime
Topic 76: angel, beat, better, thing, drama, hero, style, funny, kind, song
Topic 77: anime, fun, development, beautiful, better, loved, actually, quite, comedy, got
Topic 78: anime, moment, nice, drama, want, mob, year, bebop, little, old
Topic 79: subaru, loved, goblin, want, fight, nice, slice, fantasy, zero, bebop
Topic 80: definitely, bebop, bit, cowboy, relationship, boring, protagonist, perfect, horror, battle
Topic 81: want, better, protagonist, cute, new, make, sound, piece, female, friend
Topic 82: going, start, mob, best, cute, bad, seen, funny, felt, madoka
Topic 83: ecchi, want, style, development, quite, nice, unique, loved, demon, work
Topic 84: piece, friend, unique, ecchi, mob, slice, scene, bebop, point, gundam

```
Topic 85: enjoy, cute, new, bit, didnt, harem, interesting, scene, theme, want
Topic 86: kind, development, idol, want, actually, style, little, piece, unique,
loved
Topic 87: definitely, goblin, protagonist, little, song, felt, comedy, moment,
looking, slayer
Topic 88: bit, anime, want, moment, going, liked, loved, mob, monster, drama
Topic 89: loved, point, enjoyable, goblin, friend, didnt, thats, start, real,
feel
Topic 90: look, loved, nice, enjoy, better, favorite, probably, comedy, shounen,
piece
Topic 91: nice, want, actually, haruhi, got, liked, power, best, make, theme
Topic 92: work, power, start, different, brotherhood, piece, loved, kinda,
comedy, definitely
Topic 93: guy, better, friend, development, understand, loved, style, moment,
bleach, half
Topic 94: thought, recommend, drama, piece, new, theme, song, little, enjoyed,
going
Topic 95: protagonist, going, kinda, voice, bleach, bit, thought, new, best,
enjoyed
Topic 96: guy, point, song, haruhi, style, interesting, protagonist, actually,
action, theme
Topic 97: point, guy, seen, thought, beautiful, want, unique, different, cute,
comedy
Topic 98: got, drama, nice, season, point, pokemon, start, recommend, real,
horror
Topic 99: kind, liked, time, original, digimon, kid, real, enjoyable, far, op
Topic 100: going, make, quite, definitely, little, understand, time, kinda,
friend, jojo

==================================================
```

Relevance Consistency Check and Similarity Distribution Plot

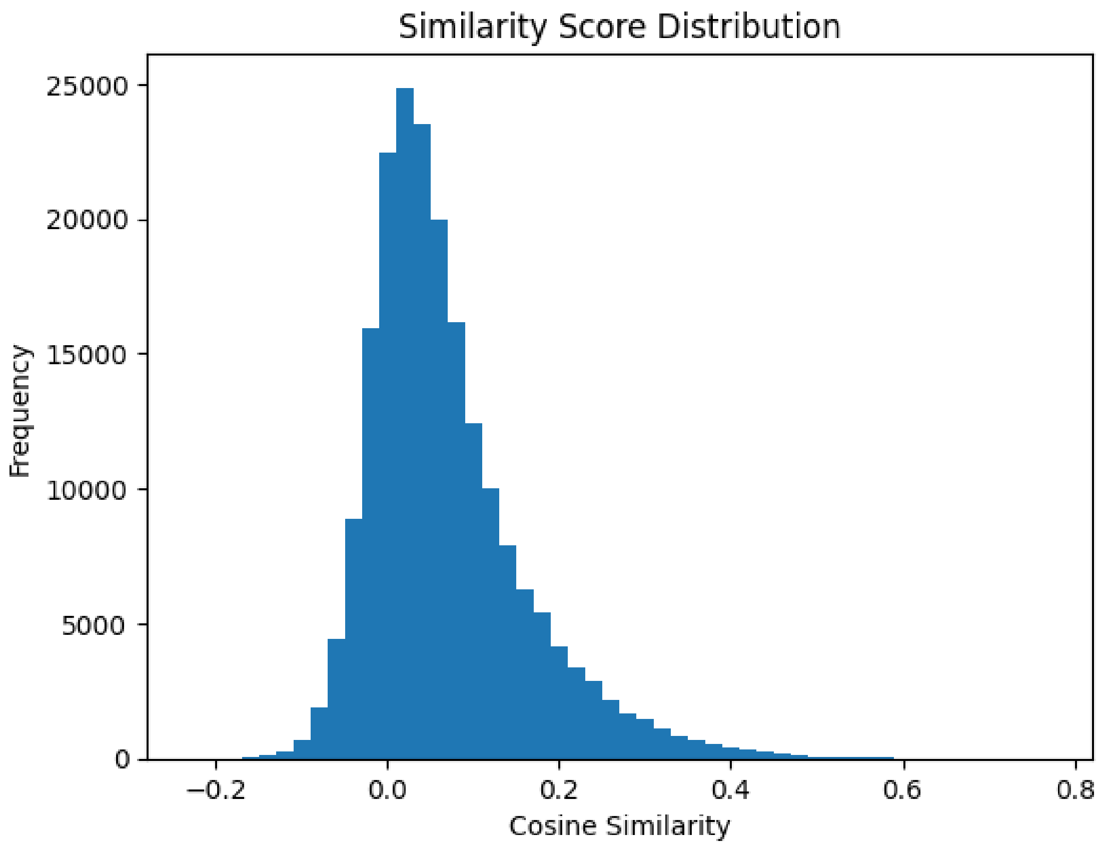
[170]:
```python
def recommend_from_query_text(query_text, df_items, tfidf_vec, tfidf_matrix,␣
↪lsa_model, num_recommendations=5):
    processed_query = preprocess(query_text)
    query_tfidf = tfidf_vec.transform([processed_query])
    query_lsa = lsa_model.transform(query_tfidf)

    similarities = cosine_similarity(query_lsa, lsa_model.
↪transform(tfidf_matrix)).flatten()

    plt.hist(similarities, bins=50)
    plt.title("Similarity Score Distribution")
    plt.xlabel("Cosine Similarity")
    plt.ylabel("Frequency")
    plt.show()
    top_indices = similarities.argsort()[-num_recommendations:][::-1]

    recommended_items = df_items.iloc[top_indices]
    recommended_items['similarity_score'] = similarities[top_indices]

    return recommended_items[['anime_id', 'review', "processed_review",␣
↪'similarity_score']]
```

[145]:
```
# load anime df
anime_df = pd.read_csv("E:\\applied data science␣
↪capstone\\data\\combined\\anime-for-db-27-Jun\\anime_list_27_Jun.csv")
```

[197]:
```
# sample queries
query_1 = "action packed series about super powers"
query_2 = "romantic comedy about a high school couple"
query_3 = "a young boy pursuing his dream of playing sports"
query_4 = "people get trapped in a game they were playing"
query_5 = "science made to be fun and easy to learn"
query_6 = "cute girls doing cute stuff"
query_7 = "fast cars pulling off crazy stunts"
query_8 = "a whole lot of guns and shooting"
query_9 = "germophobe navigating life"
query_10 = "pop idols entertaining their fans"
```

1. action packed series about super powers

[179]:
```
recommendations_new_query = recommend_from_query_text(query_1, df,␣
↪tfidf_vectorizer, tfidf_matrix, lsa_model, num_recommendations=10)
```



```
C:\Users\Asus-Home\AppData\Local\Temp\ipykernel_20432\122375758.py:17:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  recommended_items['similarity_score'] = similarities[top_indices]
```

[180]:
```
recommendations_new_query.head()
```

[180]:
```
        anime_id                                             review  \
410            6  Without spoiling anything, I'll just say that …
175042     40748  i've watched this anime twice and i'll still r…
88567      16512  I am in love with this anime at the moment! Th…
46429       4334  Ever thought of watching anime + marvel at the…
108021     25777  Every time I watch an episode of this series I…

                                         processed_review  similarity_score
410     without spoiling anything probably surprise go…          0.771056
```

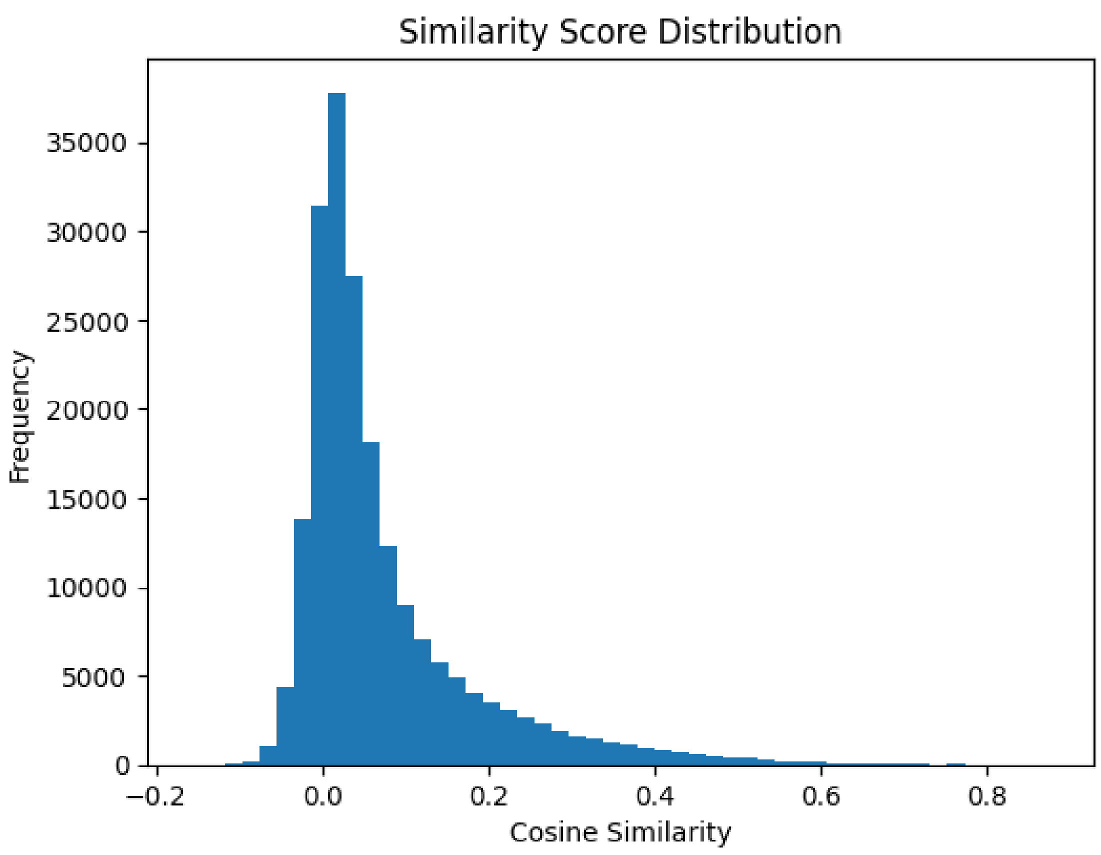
```
175042  watched twice every full action entertaining c…          0.762091
88567   moment use supernatural power easy fall line b…          0.742754
46429   ever thought marvel collaboration stan lee stu…          0.720654
108021  every cant look away captivated loose basicall…          0.698470
```

[181]:
```
recommendation_ids = set(recommendations_new_query["anime_id"].unique())
anime_df.loc[anime_df["anime_id"].isin(recommendation_ids), ["anime_id",␣
↪"title", "synopsis"]]
```

[181]:
```
      anime_id                          title  \
2            6                         Trigun
46          68                      Black Cat
599        957          The Story of Saiunkoku
1029      1914              Tales of Saiunkoku
1693      3615             Neo Angelique Abyss
1883      4334                        Heroman
3270     16512   Devil Survivor 2 The Animation
3876     25777          Attack on Titan Season 2
4602     34542             Inuyashiki: Last Hero
5419     40748                   Jujutsu Kaisen

                                               synopsis
2     Vash the Stampede is the man with a $$60,000,0…
46    Completing every job with ruthless accuracy, T…
599   Shuurei Kou, the daughter of a noble yet impov…
1029  Shuurei Kou and her friend Eigetsu To, a boy p…
1693  hile the young Angelique lives out her days pe…
1883  In California's Center City, shy but kindheart…
3270  The countdown to extinction begins on Sunday w…
3876  For centuries, humanity has been hunted by gia…
4602  Ichirou Inuyashiki is a 58-year-old family man…
5419  Idly indulging in baseless paranormal activiti…
```

2. romantic comedy about a high school couple

[182]:
```
recommendations_new_query = recommend_from_query_text(query_2, df,␣
↪tfidf_vectorizer, tfidf_matrix, lsa_model, num_recommendations=10)
```



```
C:\Users\Asus-Home\AppData\Local\Temp\ipykernel_20432\122375758.py:17:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  recommended_items['similarity_score'] = similarities[top_indices]
```

[183]:
```
recommendations_new_query.head()
```

[183]:
```
        anime_id                                             review  \
5932          66  Well despite that Azumanga Daioh is about a st…
2101          24  If you're looking for a documentary to sate yo…
115888     30240  A prison gay review of Prison School Prison Sc…
76408      11843  Unless you were incredibly popular during your…
85806      14813  Yahari Ore no Seishun Love Comedy wa Machigatt…

                                         processed_review  similarity_score
5932    despite azumanga daioh friend attempt daily sc…          0.877907
```

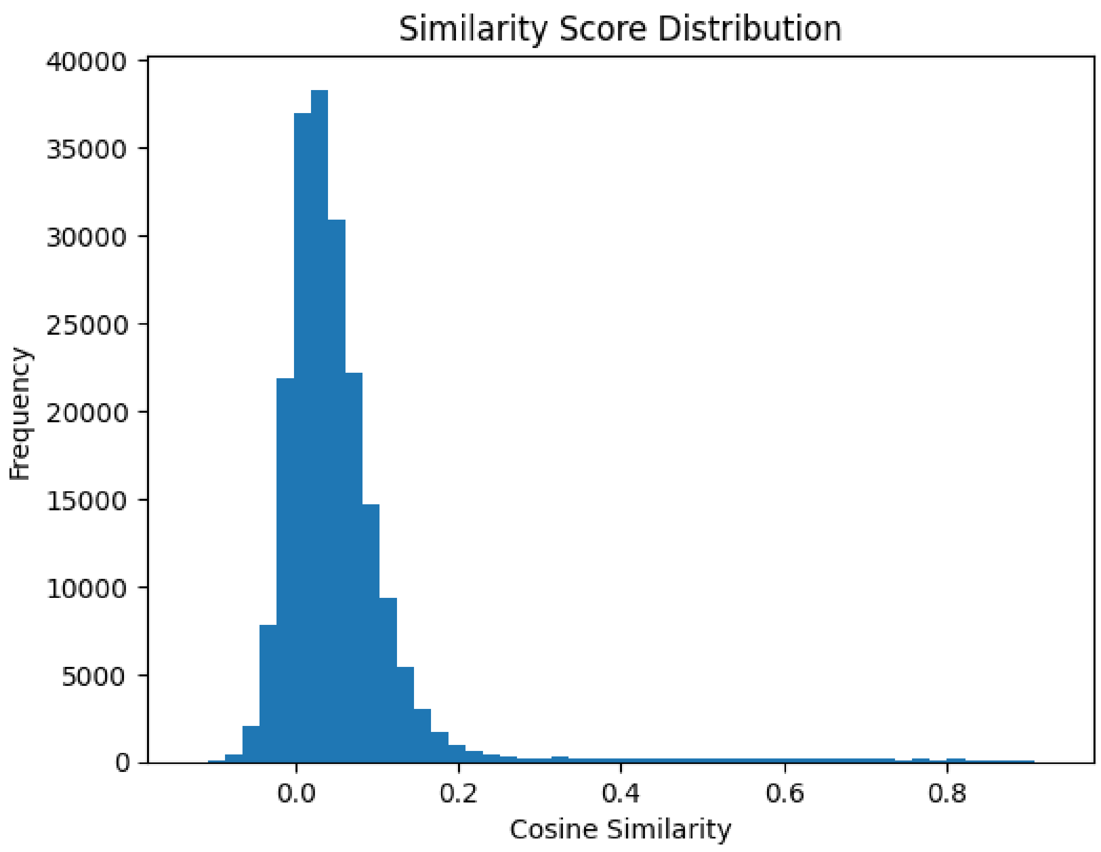
```
2101    looking documentary sate curiosity obscure phe…          0.860351
115888  prison gay review prison school prison school …          0.853149
76408   unless incredibly popular school rule due webs…          0.846322
85806   yahari ore seishun comedy wa machigatteiru yah…          0.837718
```

[184]:
```
recommendation_ids = set(recommendations_new_query["anime_id"].unique())
anime_df.loc[anime_df["anime_id"].isin(recommendation_ids), ["anime_id",␣
↪"title", "synopsis"]]
```

[184]:
```
      anime_id                            title  \
14          24                     School Rumble
44          66    Azumanga Daioh: The Animation
522        853      Ouran High School Host Club
2267      6547                     Angel Beats!
2990     11843  Daily Lives of High School Boys
3169     14813    My Teen Romantic Comedy SNAFU
4079     30240                    Prison School
4564     34281              High School DxD Hero

                                              synopsis
14    Just the words "I love you," and everything ch…
44    Chiyo Mihama begins her high school career as …
522   Haruhi Fujioka is a bright scholarship candida…
2267  Otonashi awakens only to learn he is dead. A r…
2990  oaming the halls of the all-boys Sanada North …
3169  Hachiman Hikigaya is an apathetic high school …
4079  ocated on the outskirts of Tokyo, Hachimitsu P…
4564            The fourth season of High School DxD .
```

3. a young boy pursuing his dream of playing sports

[185]:
```
recommendations_new_query = recommend_from_query_text(query_3, df,␣
↪tfidf_vectorizer, tfidf_matrix, lsa_model, num_recommendations=10)
```



```
C:\Users\Asus-Home\AppData\Local\Temp\ipykernel_20432\122375758.py:17:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  recommended_items['similarity_score'] = similarities[top_indices]
```

[186]:
```
recommendations_new_query.head()
```

[186]:
```
        anime_id                                             review  \
182743     42395  Kabaddi is a contact team sport which involves…
75163      11697  Another Sports anime added to my fave sports a…
156506     37965  I never really enjoyed any sports anime ever b…
186005     44274  First review just had to. First off Japan beat…
125629     32494  I think there's a lot of merit to judging an a…

                                          processed_review  similarity_score
182743  kabaddi contact team sport involves material e…          0.927157
```

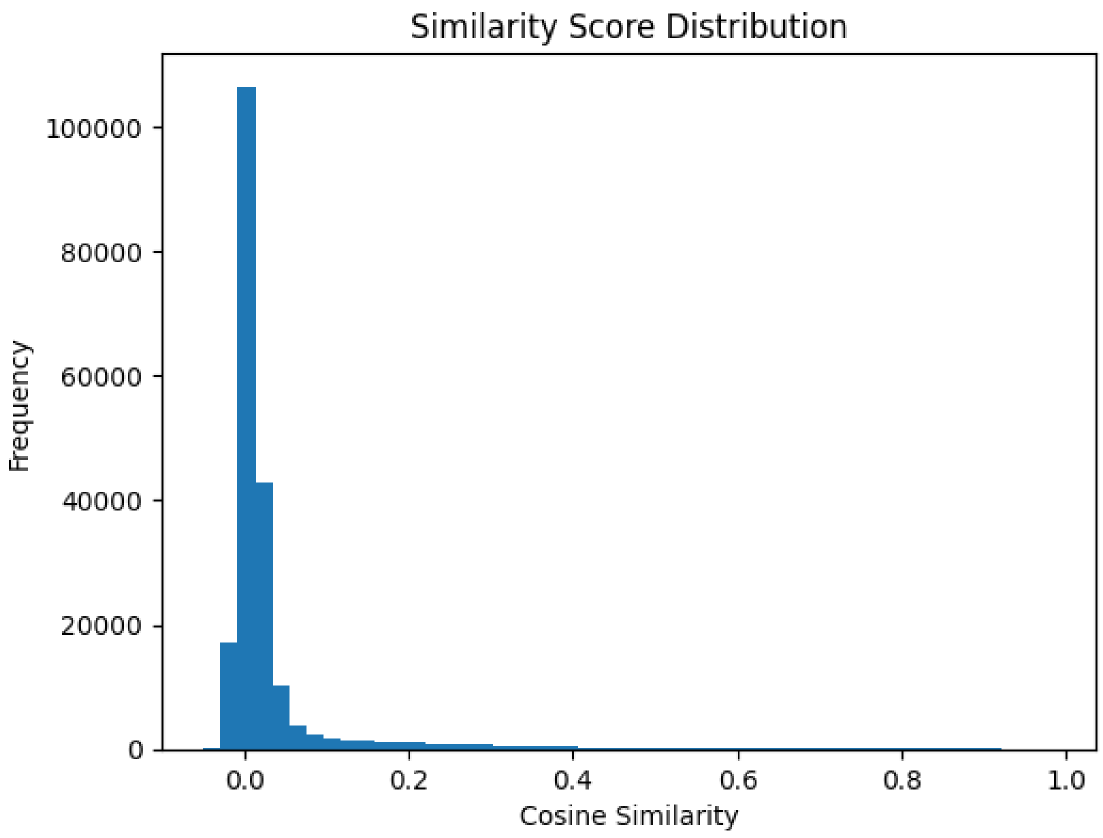
```
75163   another sport added fave sport list seems cool…          0.925690
156506  never enjoyed sport ever reason never liked sp…          0.925670
186005  review japan beating team canada set tone hock…          0.921727
125629  merit judging tell show day football american …          0.920485
```

[187]:
```python
recommendation_ids = set(recommendations_new_query["anime_id"].unique())
anime_df.loc[anime_df["anime_id"].isin(recommendation_ids), ["anime_id",␣
↪"title", "synopsis"]]
```

[187]:
```
      anime_id                              title  \
2969     11697                The Knight in the Area
3956     28391  Aokana: Four Rhythm Across the Blue
4312     32494                                  Days
5053     37965                      Run with the Wind
5565     42395                       Burning Kabaddi
5585     42774                 Farewell, My Dear Cramer
5658     44274                 PuraOra! PRIDE OF ORANGE
5814     49052                                Aoashi

                                              synopsis
2969  Kakeru and Suguru are brothers who both have a…
3956  h the invention of anti-gravitational shoes kn…
4312  The series is about two boys named Tsukushi an…
5053  Former ace runner of Sendai Josei High School,…
5565  First-year high schooler Tatsuya Yoigoshi, the…
5585  h no soccer accomplishments to speak of during…
5658  The story takes place in Nikko city, Tochigi P…
5814  In a quiet rural town, the spotlight of a loca…
```

4. people get trapped in a game they were playing

[188]:
```python
recommendations_new_query = recommend_from_query_text(query_4, df,␣
↪tfidf_vectorizer, tfidf_matrix, lsa_model, num_recommendations=10)
```



```
C:\Users\Asus-Home\AppData\Local\Temp\ipykernel_20432\122375758.py:17:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  recommended_items['similarity_score'] = similarities[top_indices]
```

[189]:
```
recommendations_new_query.head()
```

[189]:
```
        anime_id                                             review  \
182300     42307  Should you watch this instead of playing the g…
98202      21511  I can't belive this… They ruined it… It's …
77109      11757  The story is great so far. Although following …
189901     48441  If you've played and liked the Cold Steel seri…
194455     50205  Arknights : Prelude to dawn is an anime adapta…

                                         processed_review  similarity_score
182300  instead playing game hell game masterpiece spe…          0.984227
98202   belive ruined kadokawa compane created though …          0.978889
```

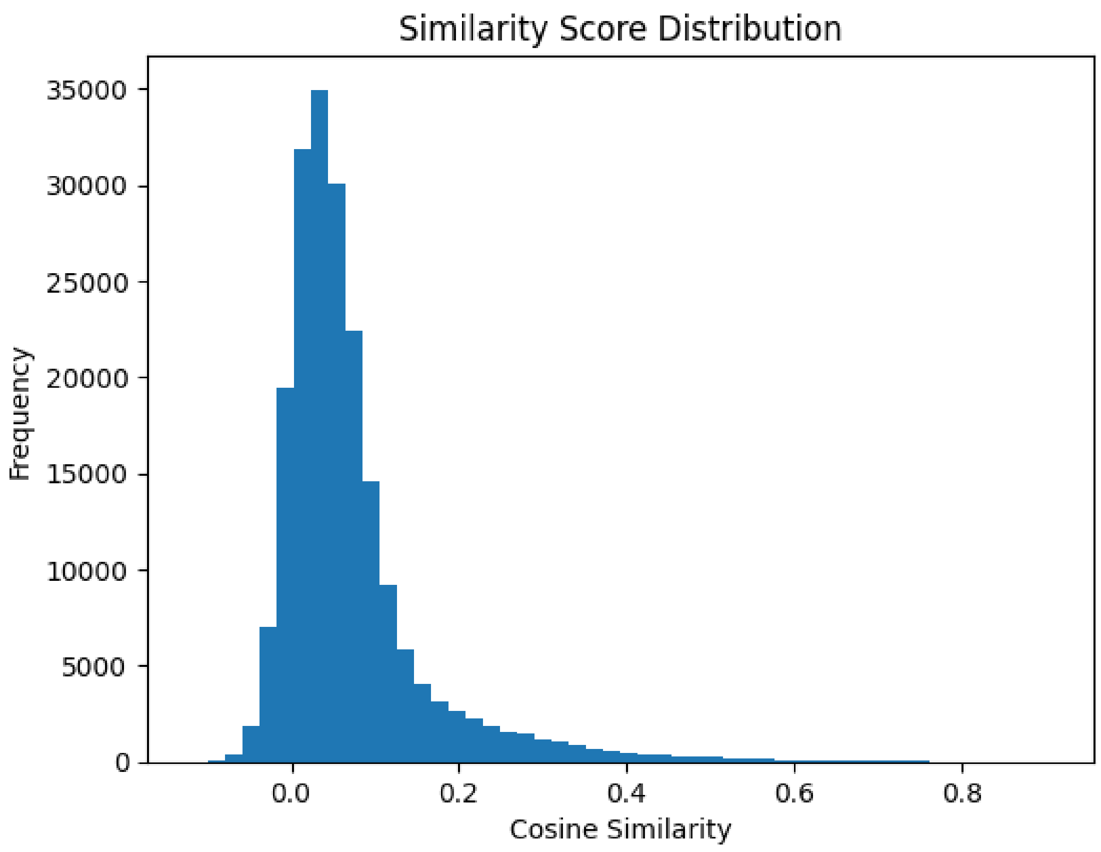
```
77109   far although following rule game contain const…         0.973301
189901  played liked cold steel game appreciate game o…         0.972333
194455  arknights prelude dawn adaptation game name ar…         0.969872
```

[190]:
```
recommendation_ids = set(recommendations_new_query["anime_id"].unique())
anime_df.loc[anime_df["anime_id"].isin(recommendation_ids), ["anime_id",␣
↪"title", "synopsis"]]
```

[190]:
```
      anime_id                                                title  \
2869     10588                               Persona 4 the Animation
2978     11757                                     Sword Art Online
3286     16592                              Danganronpa: The Animation
3535     19815                                      No Game, No Life
3661     21511                              KanColle: Kantai Collection
3947     28223                                          Death Parade
5370     40403                                Sakura Wars the Animation
5556     42307                     The World Ends with You The Animation
5759     48441  The Legend of Heroes: Trails of Cold Steel - N…
5888     50205                       Arknights Animation: Prelude to Dawn

                                               synopsis
2869  Yuu Narukami moves to Inaba, a seemingly quiet…
2978  In the year 2022, virtual reality has progress…
3286  Hope's Peak Academy is an elite high school th…
3535  No Game No Life is a surreal comedy that follo…
3661  h the seas under constant threat from the host…
3947  fter death, there is no heaven or hell, only a…
5370  In 1930, two years after the events of So Long…
5556  Neku Sakuraba, a 15-year-old boy with a hobby …
5759  Eiyuu Densetsu: Sen no Kiseki centers around R…
5888  The discovery of a black crystalline mineral k…
```

5. science made to be fun and easy to learn

[191]:
```
recommendations_new_query = recommend_from_query_text(query_5, df,␣
↪tfidf_vectorizer, tfidf_matrix, lsa_model, num_recommendations=10)
```



```
C:\Users\Asus-Home\AppData\Local\Temp\ipykernel_20432\122375758.py:17:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  recommended_items['similarity_score'] = similarities[top_indices]
```

[192]:
```
recommendations_new_query.head()
```

[192]:
```
        anime_id                                             review  \
123793     31953  This anime is just a delight to watch. Sure it…
112884     28825  This story is quite simple, this is a kid who …
176102     40852  Dr. Stone: Stone Wars is such a fun anime to w…
132892     33489  This is a simple and fun show, one of those wi…
196000     49385  Kaijin Kaihatsu-bu no Kuroitsu-san (2022) This…

                                         processed_review  similarity_score
123793  delight sure remarkable revolutionize fun enjo…          0.906178
```

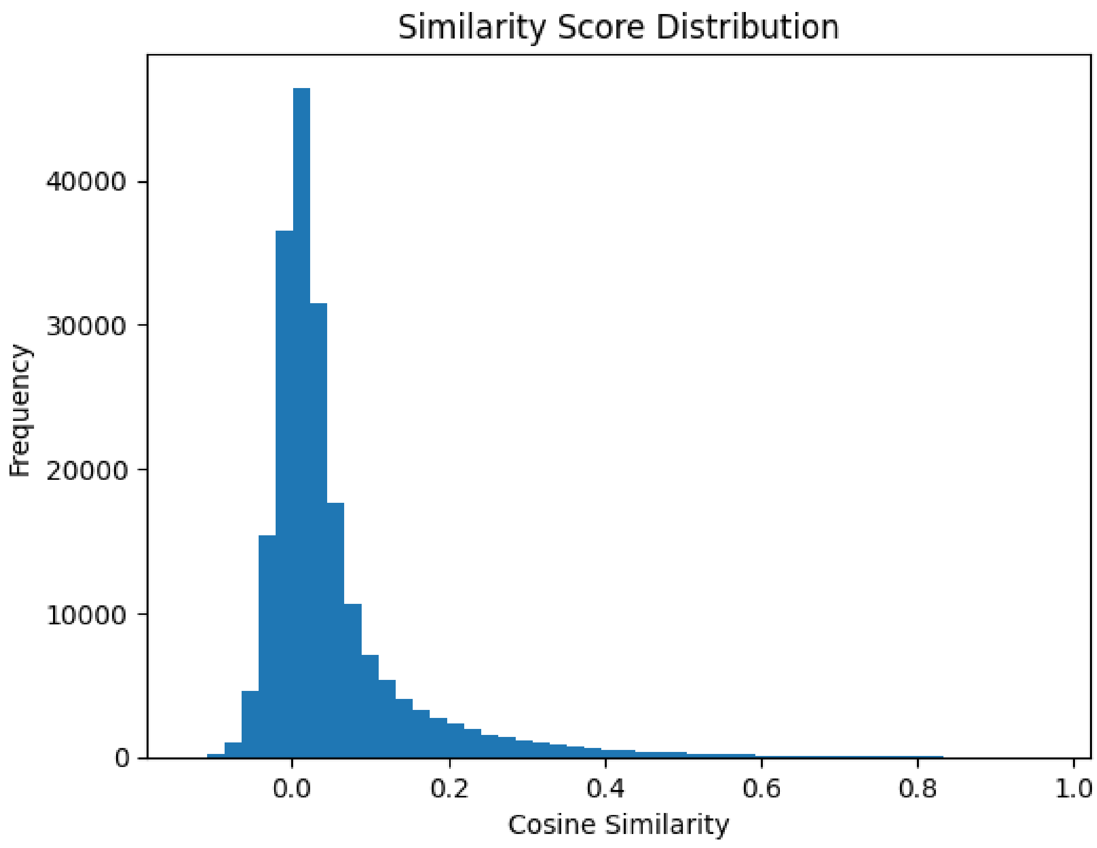
```
112884  quite simple kid decides live double personali…        0.900556
176102  dr stone stone war fun continuation coming hyp…        0.892922
132892  simple fun intention fun ride explain everythi…        0.886631
196000  kaijin kaihatsu bu kuroitsu san fun expected e…        0.872348
```

[193]:
```
recommendation_ids = set(recommendations_new_query["anime_id"].unique())
anime_df.loc[anime_df["anime_id"].isin(recommendation_ids), ["anime_id",␣
↪"title", "synopsis"]]
```

[193]:
```
      anime_id                                              title  \
162        223                                        Dragon Ball
2245      6347                     Baka & Test - Summon the Beasts
3702     22147                                Amagi Brilliant Park
3941     28121  Is It Wrong to Try to Pick Up Girls in a Dungeon?
3982     28825                                  Himouto! Umaru-chan
4267     31953                                           New Game!
4468     33489                              Little Witch Academia
5434     40852      Dr. Stone 2nd Season, Dr. Stone Second Season
5599     42923                                    SK8 the Infinity
5833     49385  Miss Kuroitsu from the Monster Development Dep…

                                               synopsis
162   Gokuu Son is a young boy who lives in the wood…
2245  Fumizuki Academy isn't a typical Japanese high…
3702  Seiya Kanie, a smart and extremely narcissisti…
3941  fe in the bustling city of Orario is never dul…
3982  People are not always who they appear to be, a…
4267  Since childhood, Aoba Suzukaze has loved the F…
4468  "A believing heart is your magic!"-these were …
5434  Senkuu has made it his goal to bring back two …
5599  High school student Reki Kyan is passionate ab…
5833  On the outside, Agastia is a firm specializing…
```

6. cute girls doing cute stuff

[198]:
```
recommendations_new_query = recommend_from_query_text(query_6, df,␣
↪tfidf_vectorizer, tfidf_matrix, lsa_model, num_recommendations=10)
```



```
C:\Users\Asus-Home\AppData\Local\Temp\ipykernel_20432\122375758.py:17:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  recommended_items['similarity_score'] = similarities[top_indices]
```

[199]:
```
recommendations_new_query.head()
```

[199]:
```
        anime_id                                             review  \
52501       5680  K-On! is my comfort anime and it never fails t…
157726     37993  A lot of this reviewers are really quick to sa…
91447      17549  A sweet and cute anime that just relaxing to w…
157750     37993  This show has yet again relit my flame for cut…
141486     35241  "Cute fox girls doing cute things." Konohana K…

                                         processed_review  similarity_score
52501   k comfort never fails smile definition definit…          0.966288
```

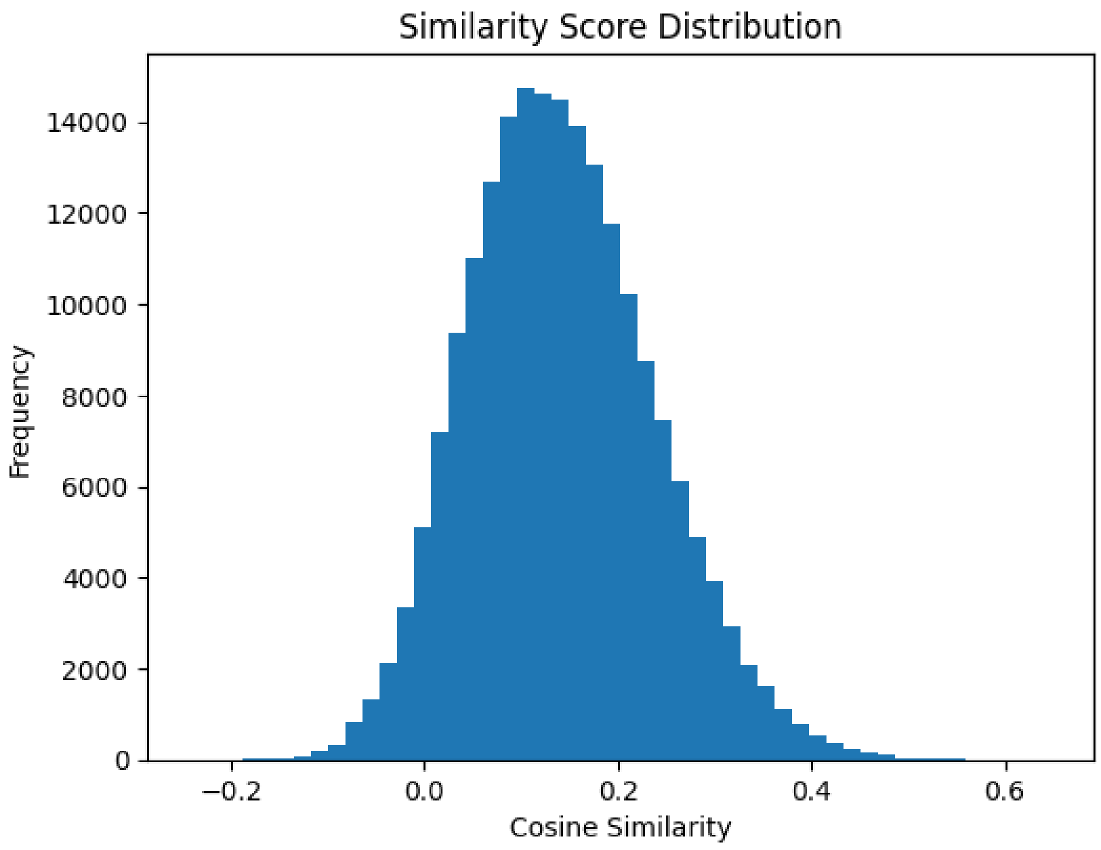
```
157726  reviewer quick cosplay cute stuff loving shut …         0.957362
91447   sweet cute relaxing want type girl likable sup…         0.954955
157750  yet relit flame cute show cute girl absolutely…         0.930994
141486  cute fox girl cute thing konohana kitan litera…         0.926689
```

[200]:
```
recommendation_ids = set(recommendations_new_query["anime_id"].unique())
anime_df.loc[anime_df["anime_id"].isin(recommendation_ids), ["anime_id",␣
↪"title", "synopsis"]]
```

[200]:
```
      anime_id                                              title  \
2136      5680                                             K-ON!
3138     14283                                 Vividred Operation
3372     17549                                     Non Non Biyori
4371     32901                                   Eromanga Sensei
4694     35241                                     Konohana Kitan
5016     37704                                      Haifuri Movie
5069     37993                   WataTen! An Angel Flew Down to Me
5238     39324  If It's for My Daughter, I'd Even Defeat a Dem…
5394     40571                Wandering Witch: The Journey of Elaina

                                               synopsis
2136  fresh high school year always means much to co…
3138  Friendship is the key to protecting the world…
3372  sahigaoka might look like typical, boring coun…
4371  One year ago, Sagiri Izumi became step-sibling…
4694  In a bustling village of spirits, Yuzu, a chee…
5016  Yokosuka Girls' Marine High School is bustling…
5069  College student Miyako Hoshino is quite shy ar…
5238  Eighteen-year-old Dale Reki is a skilled, kind…
5394  Since childhood, Elaina has always been fascin…
```

7. fast cars pulling off crazy stunts

[201]:
```
recommendations_new_query = recommend_from_query_text(query_7, df,␣
↪tfidf_vectorizer, tfidf_matrix, lsa_model, num_recommendations=10)
```



```
C:\Users\Asus-Home\AppData\Local\Temp\ipykernel_20432\122375758.py:17:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  recommended_items['similarity_score'] = similarities[top_indices]
```

[202]:
```
recommendations_new_query.head()
```

[202]:
```
        anime_id                                             review  \
55053       6675  FAST CARS! FAST CARS! FAST CARS! FAST CARS! FA…
84312      14513  The story drags with barely any fighting scene…
24831        974  Dead Leaves… How can I even describe thi…
93592      18679  I usually never write reviews, but this anime …
191246     49387  I'm only at episode 5 and I'm frustrated. I lo…

                                         processed_review  similarity_score
55053   fast car fast car fast car fast car fast car e…          0.647523
```

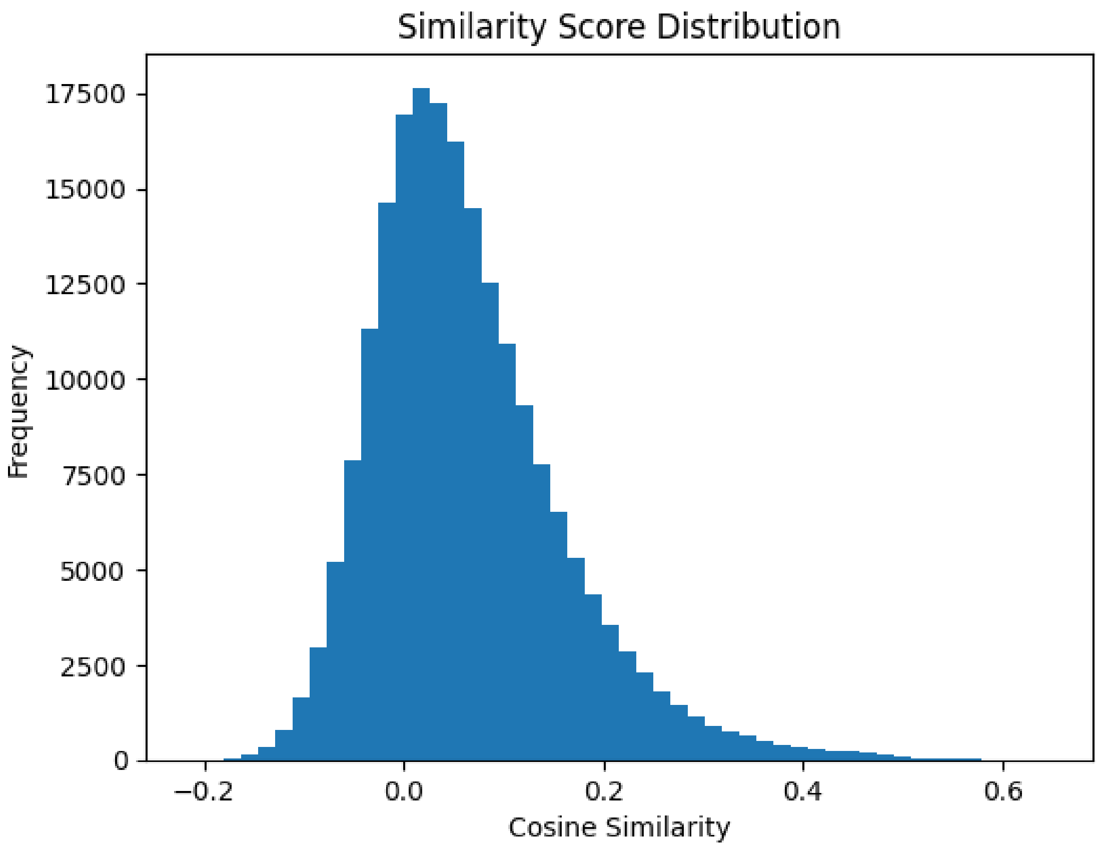
```
84312   drag barely fighting scene pacing painfully sl…          0.610955
24831   dead leaf describe animated production g direc…          0.605487
93592   usually never write review thing right kind pa…          0.604848
191246  frustrated loved vinland saga probably seen sa…          0.604649
```

[203]:
```
recommendation_ids = set(recommendations_new_query["anime_id"].unique())
anime_df.loc[anime_df["anime_id"].isin(recommendation_ids), ["anime_id",␣
↪"title", "synopsis"]]
```

[203]:
```
      anime_id                        title  \
613        974                  Dead Leaves
757       1292                 Afro Samurai
2288      6675                      Redline
2354      7088            Demon King Daimao
3151     14513  Magi: The Labyrinth of Magic
3471     18679                  Kill la Kill
5732     47917               Bocchi the Rock!
5834     49387          Vinland Saga Season 2
5848     49596                    Blue Lock

                                               synopsis
613   Pandy and Retro awaken naked on Earth with no …
757   hen he was a young boy, Afro witnessed his fat…
2288  Every five years, an exhilarating race called …
2354  Dreaming of changing the world for good, Akuto…
3151  Dispersed around the world, there are several …
3471  fter the murder of her father, Ryuuko Matoi ha…
5732  Hitori Gotou is a high school girl who's start…
5834  After his father's death and the destruction o…
5848  Yoichi Isagi was mere moments away from scorin…
```

8. a whole lot of guns and shooting

[204]:
```
recommendations_new_query = recommend_from_query_text(query_8, df,␣
↪tfidf_vectorizer, tfidf_matrix, lsa_model, num_recommendations=10)
```



```
C:\Users\Asus-Home\AppData\Local\Temp\ipykernel_20432\122375758.py:17:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  recommended_items['similarity_score'] = similarities[top_indices]
```

[205]:
```
recommendations_new_query.head()
```

[205]:
```
        anime_id                                            review  \
188390     46604  A decent gun action anime. It's no Lycoris Rec…
77850      11757  The main reason Sword Art Online gets a 10/10 …
16857        411  Quentin Tarantino is one strange guy. His cult…
54786       6594  This is a show about swords, the title even ro…
23877        889  Welcome back Studio Madhouse. The person chose…

                                         processed_review  similarity_score
188390  decent gun action lycoris recoil sao sao alter…          0.647498
```

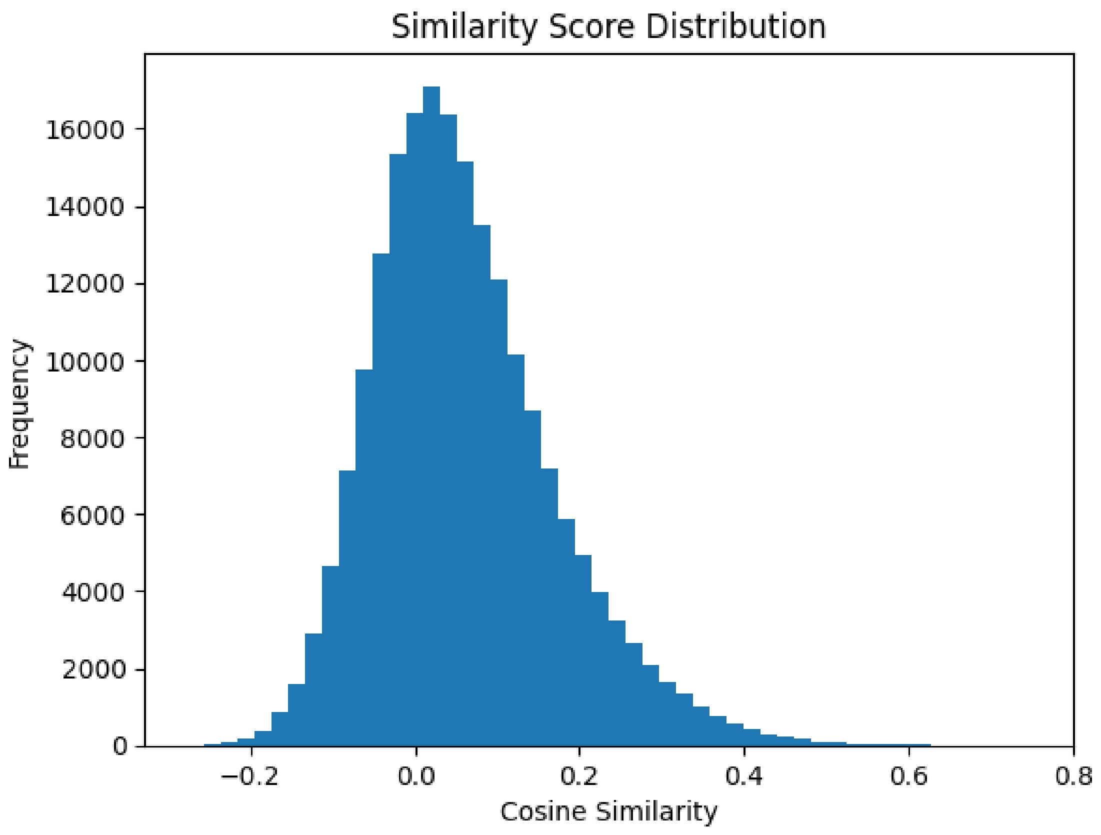
```
77850   reason sword online get found kirito almost th…         0.628612
16857   quentin tarantino strange guy cult film homage…         0.628402
54786   sword title roughly translates sword sword swo…         0.626034
23877   welcome back studio madhouse person chosen dir…         0.614767
```

[206]:
```
recommendation_ids = set(recommendations_new_query["anime_id"].unique())
anime_df.loc[anime_df["anime_id"].isin(recommendation_ids), ["anime_id",␣
↪"title", "synopsis"]]
```

[206]:
```
      anime_id                                              title  \
293        411                                        Gun x Sword
550        889                                       Black Lagoon
757       1292                                       Afro Samurai
2272      6594                                       Katanagatari
2978     11757                                   Sword Art Online
5282     39597  Sword Art Online: Alicization - War of Underworld
5715     46604                                   Girls' Frontline

                                               synopsis
293   Van, a lanky and apathetic swordsman, is on a …
550   hin Thailand is Roanapur, a depraved, crime-ri…
757   hen he was a young boy, Afro witnessed his fat…
2272  In an Edo-era Japan lush with a variety of swo…
2978  In the year 2022, virtual reality has progress…
5282  Despite the defeat of Quinella-the pontifex of…
5715  After World War III decimated the world's popu…
```

9. germophobe navigating life

[207]:
```
recommendations_new_query = recommend_from_query_text(query_9, df,␣
↪tfidf_vectorizer, tfidf_matrix, lsa_model, num_recommendations=10)
```



```
C:\Users\Asus-Home\AppData\Local\Temp\ipykernel_20432\122375758.py:17:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  recommended_items['similarity_score'] = similarities[top_indices]
```

[208]:
```
recommendations_new_query.head()
```

[208]:
```
        anime_id                                             review  \
193935     50425  "More than a Married Couple, but Not Lovers" i…
177370     40938  "Higehiro: After Being Rejected, I Shaved and …
184166     42897  Title: Horimiya - A Heartfelt Dive into Youthf…
83454      14289  Suki tte Ii na yo (Say "I Love You") - A Since…
193917     50425  More than a Married Couple, but Not Lovers - A…

                                          processed_review  similarity_score
193935  married couple lover captivating drama delf co…          0.749078
177370  higehiro rejected shaved took high school runa…          0.736836
```

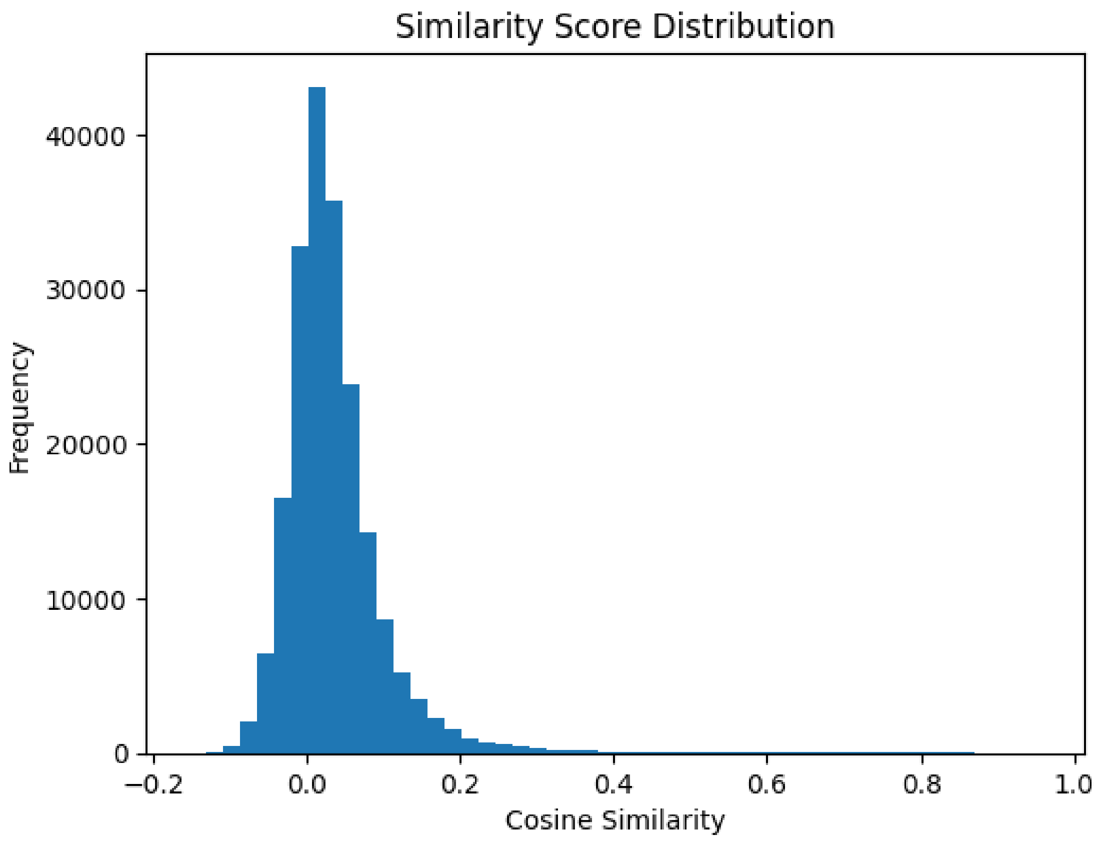
```
184166  title horimiya heartfelt dive youthful relatio…          0.731445
83454   suki tte ii na yo sincere relatable journey gr…          0.729850
193917  married couple lover bittersweet exploration r…          0.726310
```

```
[209]: recommendation_ids = set(recommendations_new_query["anime_id"].unique())
       anime_df.loc[anime_df["anime_id"].isin(recommendation_ids), ["anime_id",␣
        ↪"title", "synopsis"]]
```

```
[209]:      anime_id                                              title  \
       3139    14289                                    Say "I Love You."
       3244    16067                                    A Lull in the Sea
       4485    33654                                    Hitorijime My Hero
       4807    36220     Our love has always been 10 centimeters apart.
       5450    40938  Higehiro: After Being Rejected, I Shaved and T…
       5595    42897                                             Horimiya
       5708    46352                                          Blue Period
       5908    50425          More than a married couple, but not lovers.
       6017    52578                               The Dangers in My Heart

                                                        synopsis
       3139  Friends will only let you down-that is the sad…
       3244  ong ago, all humans lived beneath the sea. How…
       4485  asahiro Setagawa is a hopeless teenager who is…
       4807  ou Aida and Haruki Serizawa might seem like po…
       5450  Office worker Yoshida has been crushing on his…
       5595  On the surface, the thought of Kyouko Hori and…
       5708  Second-year high school student Yatora Yaguchi…
       5908  Third-year high school student Jirou Yakuin is…
       6017  Kyoutarou Ichikawa may look like a shy and res…
```

10. pop idols entertaining their fans

```
[210]: recommendations_new_query = recommend_from_query_text(query_10, df,␣
        ↪tfidf_vectorizer, tfidf_matrix, lsa_model, num_recommendations=10)
```



```
C:\Users\Asus-Home\AppData\Local\Temp\ipykernel_20432\122375758.py:17:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  recommended_items['similarity_score'] = similarities[top_indices]
```

[211]:
```
recommendations_new_query.head()
```

[211]:
```
        anime_id                                             review  \
156319     37890  At the start of the season I saw a new Idol an…
156313     37890  It's unique, that's why it's awesome. Most of …
198483     49692  Heroines Run the Show is an interesting look a…
191474     52034  To follow non-idol fans… I don't like idol a…
156327     37890  Don't let the poster fool you: this is not an …

                                         processed_review  similarity_score
156319  start saw new idol airing another maybe downlo…          0.957675
```

```
156313  unique awesome idol idol star group girl dream…        0.949029
198483  heroine run interesting look idol culture lens…        0.933319
191474  follow non idol fan idol honest despise idol c…        0.917976
156327  let poster fool idol comedy idol fan culture a…        0.912640
```

[212]:
```
recommendation_ids = set(recommendations_new_query["anime_id"].unique())
anime_df.loc[anime_df["anime_id"].isin(recommendation_ids), ["anime_id",␣
 ↪"title", "synopsis"]]
```

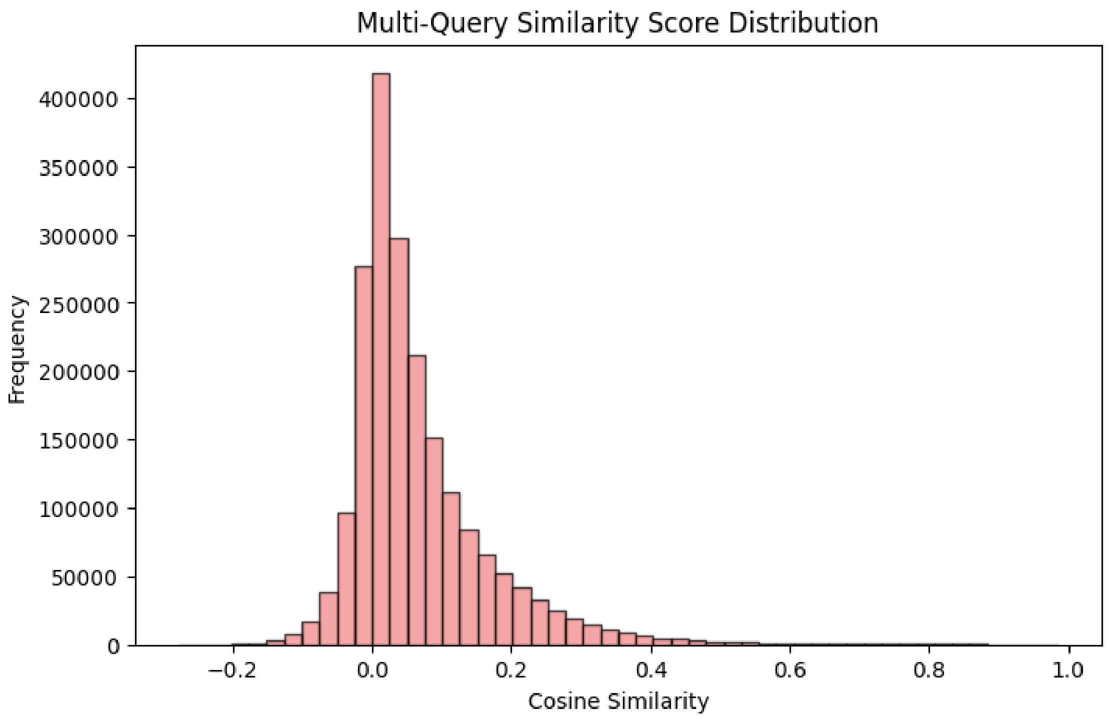
[212]:
```
      anime_id                                              title  \
3013     12149                                            AKB0048
5043     37890  If My Favorite Pop Idol Made It to the Budokan…
5059     37976                                    Zombie Land Saga
5283     39609                              Dropout Idol Fruit Tart
5852     49692  Heroines Run the Show: The Unpopular Girl and …
5996     52034                                         [Oshi No Ko]

                                               synopsis
3013  fter an interplanetary war at the beginning of…
5043  girl is obsessed with her favorite idol, a min…
5059  Sakura Minamoto dreams of becoming an idol. Un…
5283  Fourth dormitory of the Rat Production (common…
5852  Freshly graduating middle school, energetic 15…
5996  In the entertainment world, celebrities often …
```

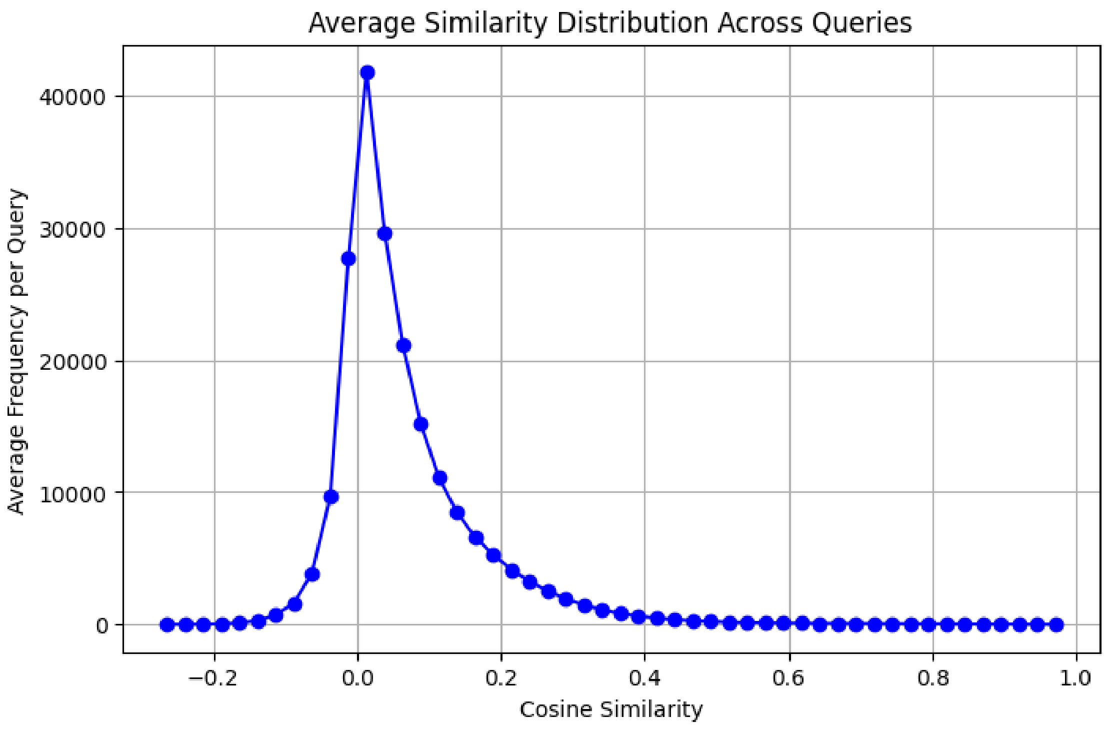
Similarity Distribution

[213]:
```
def plot_multiquery_similarity_distribution(query_texts, tfidf_vec,␣
 ↪tfidf_matrix, lsa_model, bins=50):
    all_similarities = []

    for query in query_texts:
        # Preprocess and transform query
        processed_query = preprocess(query)
        query_tfidf = tfidf_vec.transform([processed_query])
        query_lsa = lsa_model.transform(query_tfidf)

        # Compute similarities with all items
        sims = cosine_similarity(query_lsa, lsa_model.transform(tfidf_matrix)).
 ↪flatten()
        all_similarities.append(sims)

    # Convert to numpy array for averaging
    all_similarities = np.array(all_similarities)

    # Flatten for combined histogram
    flat_sims = all_similarities.flatten()
    # Plot histogram
    plt.figure(figsize=(8, 5))
    plt.hist(flat_sims, bins=bins, color='lightcoral', edgecolor='black',␣
  ↪alpha=0.7)
    plt.title("Multi-Query Similarity Score Distribution")
    plt.xlabel("Cosine Similarity")
    plt.ylabel("Frequency")
    plt.show()

    # Optionally plot average similarity curve across bins
    bin_counts, bin_edges = np.histogram(flat_sims, bins=bins)
    bin_centers = (bin_edges[:-1] + bin_edges[1:]) / 2
    avg_counts = bin_counts / len(query_texts)

    plt.figure(figsize=(8, 5))
    plt.plot(bin_centers, avg_counts, marker='o', color='blue')
    plt.title("Average Similarity Distribution Across Queries")
    plt.xlabel("Cosine Similarity")
    plt.ylabel("Average Frequency per Query")
    plt.grid(True)
    plt.show()
```

The model performs well over a diverse set of prompts as evidenced by the cluster of similarities around the lower scores and a tail towards the higher similarity scores.
The curve shows that the average similarities peak around the lower scores.

[214]:
```
sample_queries = [query_1, query_2, query_3, query_4, query_5, query_6,␣
  ↪query_7, query_8, query_9, query_10]
plot_multiquery_similarity_distribution(sample_queries, tfidf_vectorizer,␣
  ↪tfidf_matrix, lsa_model)
```


Multi-Query Similarity Score Distribution
Frequency
Cosine Similarity
0
50000
100000
150000
200000
250000
300000
350000
400000
−0.2
0.0
0.2
0.4
0.6
0.8
1.0


Average Similarity Distribution Across Queries
Average Frequency per Query
Cosine Similarity
0
10000
20000
30000
40000
−0.2
0.0
0.2
0.4
0.6
0.8
1.0