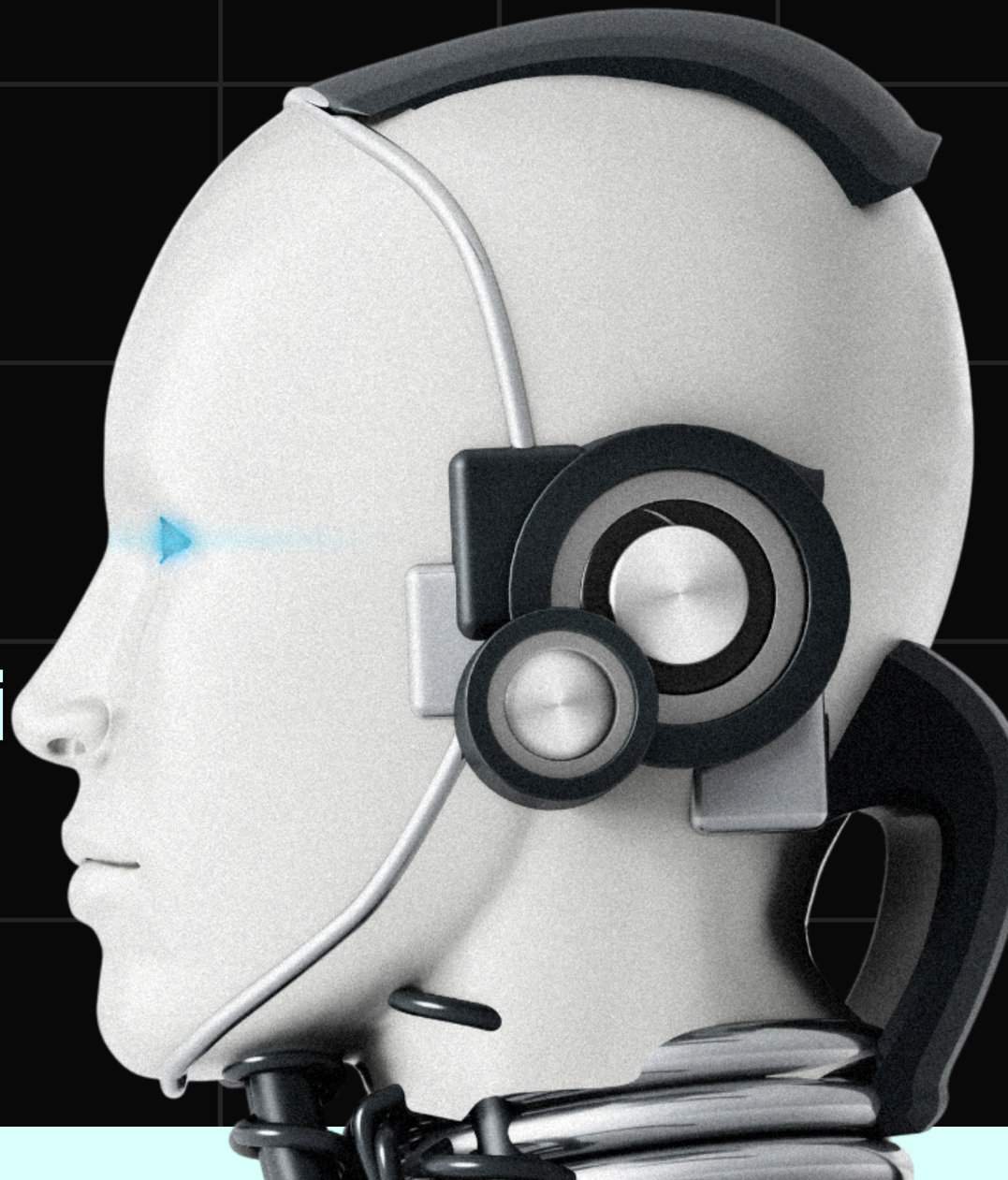
Sistem za preporuku filma

Projekat iz predmeta
računarska inteligencija

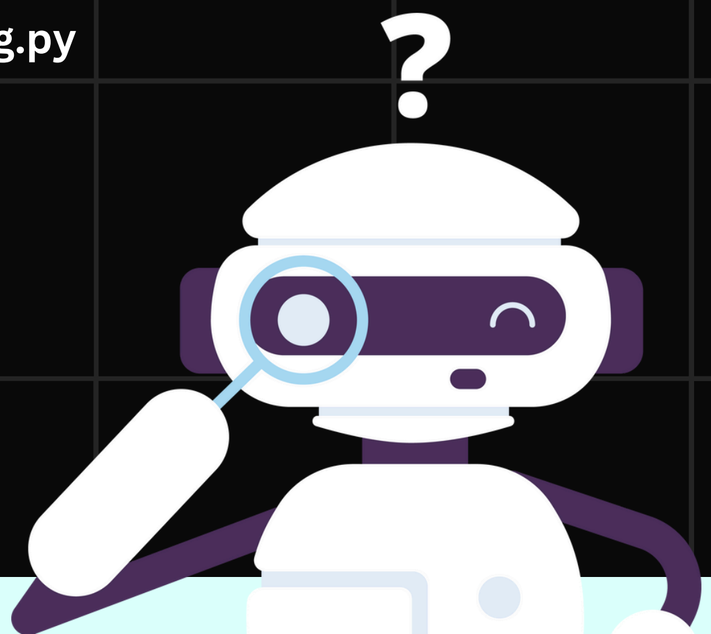
Ideja našeg projekta je bila da napravimo sistem za preporuku filma koji predlaže slične filmove pre svega na osnovu njihovog naslova, žanra, opisa, ključnih reči, takođe se pored ovih parametara gleda i popularnost, prosečna ocena i prihod koji je film ostvario.

Koriste se metode kao što su KNN (K-najbliži susedi) i kosinusna sličnost kako bismo korisnicima preporučili filmove slične onima koje već vole.



Podaci i pretprocesiranje

datoteka preprocessing.py



Podaci

Koristili smo podatke koji se nalaze u datoteci tmdb_5000_movies.csv koja sadrži približno 5000 filmova zajedno sa svojim atributima od kojih smo pomenuli već nama bitne.

Pretprocesiranje

Pretprocesiranjem podataka koje smo dobili učitavanjem našeg data seta ćemo izdvojiti i transformisati bitne atribute koje ćemo koristiti u ostatku programa.

datoteka preprocessing.py

```python
def parse_list_column(df, column_name):
    parsed_list = []
    for i in range(len(df)):
        items = ast.literal_eval(df[column_name].iloc[i])
        parsed_list.append(' '.join([item['name'] for item in items]))
    return parsed_list

# Processing genres and keywords
movies_df['genres'] = parse_list_column(movies_df, 'genres')
movies_df['keywords'] = parse_list_column(movies_df, 'keywords')
```

parse_list_column - funkcija prima DataFrame i naziv kolone, parsira sadržaj kolone iz stringa u listu rečnika, i vraća listu reči koje predstavljaju žanrove ili ključne reči.

datoteka preprocessing.py

```
# Load/train word vectors
model = Word2Vec(sentences=common_texts, vector_size=100, window=5, min_count=1, workers=4)
model.wv.save("word2vec.wordvectors")

# Loading word vectors
word_vectors = KeyedVectors.load("word2vec.wordvectors", mmap='r')

# Function to get sentence embedding with a weight for overview
def get_sentence_embedding(title, overview, model, overview_weight=2):
    title_words = title.split()
    overview_words = overview.split()

    title_embeddings = [model[word] for word in title_words if word in model]
    overview_embeddings = [model[word] for word in overview_words if word in model]

    if title_embeddings or overview_embeddings:
        combined_embeddings = title_embeddings + overview_embeddings * overview_weight
        return np.mean(combined_embeddings, axis=0)
    else:
        return np.zeros(model.vector_size)

# Processing overview
movies_df['overview'] = movies_df['overview'].fillna('')
movies_df['embedding'] = movies_df.apply(lambda row: get_sentence_embedding(row['title'],
                                                    row['overview'], word_vectors), axis=1)

# Converting embeddings to a numpy array
embedding_matrix = np.vstack(movies_df['embedding'].values)
```

- Koristimo Word2Vec model za kreiranje vektorskih reprezentacija (embeddinga) za naslove i opise filmova odnosno za kreiranje vektorskih reprezentacija reči.
- get_sentence_embedding funkcija kreira embedding za svaku rečenicu tako što kombinuje embeddinge naslova i opisa, s tim što naglašavamo da je opis značajniji od naslova.

datoteka preprocessing.py

```python
# Normalizing numerical features
numerical_features = ['popularity', 'vote_average', 'vote_count', 'revenue']
scaler = MinMaxScaler()
movies_df[numerical_features] = scaler.fit_transform(movies_df[numerical_features])

# Combining genres, keywords, and overview
movies_df['combined_features'] = movies_df['genres'] + ' ' + movies_df['keywords'] + ' ' + movies_df['overview']

# Vectorizing combined features
count_vectorizer = CountVectorizer(stop_words='english')
count_matrix = count_vectorizer.fit_transform(movies_df['combined_features'])

# Combining all features into a single feature matrix
features_matrix = np.hstack((count_matrix.toarray(), embedding_matrix, movies_df[numerical_features].values))
```

Normalizujemo numeričke karakteristike i kombinujemo različite karakteristike u jednu matricu. Koristimo MinMaxScaler za normalizaciju numeričkih karakteristika. Kombinujemo tekstualne karakteristike (genres, keywords, overview) i kreiramo vektorsku reprezentaciju pomoću CountVectorizer. Spajamo sve karakteristike u jednu veliku matricu.

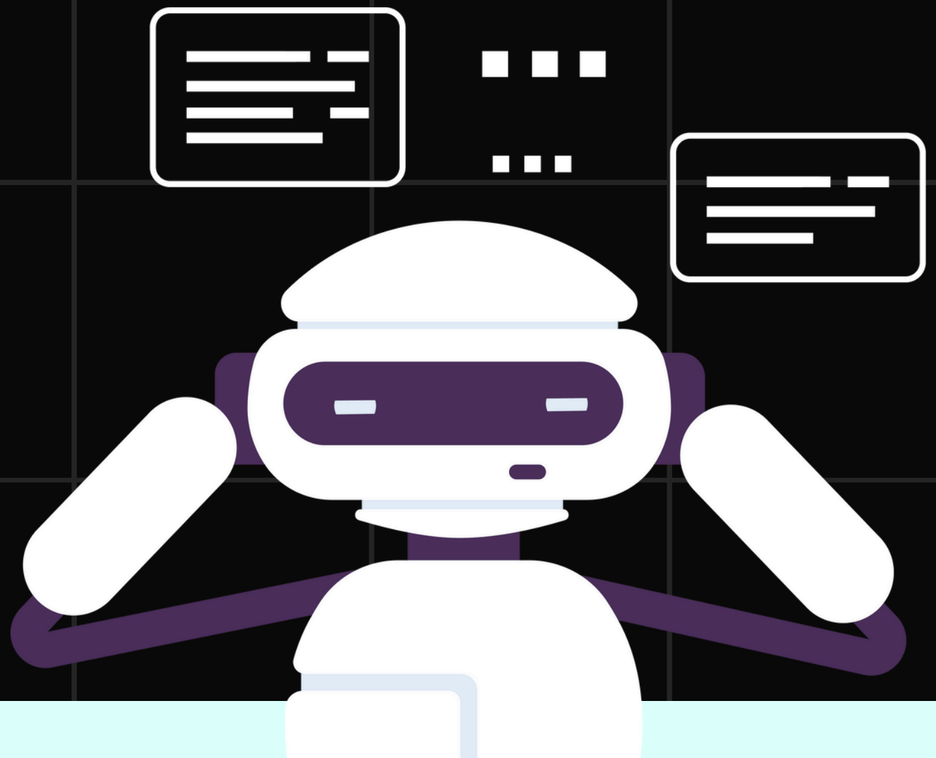
datoteka preprocessing.py

```
# Splitting the data
train_data, test_data, train_indices, test_indices = train_test_split(features_matrix,
                                                                      movies_df.index,
                                                                      test_size=0.2,
                                                                      random_state=42)
#print(f'Train data shape: {train_data.shape}, Test data shape: {test_data.shape}')  # Debug statement
```

Kada završimo sa preprocesiranjem podataka, onda te podatke, koje smo dobili u velikoj matrici koja sadrži standardizovane atribute koji će nam biti važni za primenu odgovarajućih algoritama, delimo na trening i test deo.

Glavni deo programa

main.py



Korisnik unosi naziv filma za koji želi da dobije slične preporuke. Program pronalazi film u dataset-u i omogućava korisniku da odabere metod preporuke (KNN ili kosinusna sličnost). Prikazuje detalje filma koji je korisnik odabrao i njemu slične filmove zajedno sa opisima .

Metode preporuke

KNN

Jednostavan, ali efikasan algoritam za klasifikaciju i regresiju. KNN meri udaljenost između tačaka (u ovom slučaju filmova) pomoću različitih metričkih funkcija, kao što su Euklidska ili Manhetnova udaljenost. Kada korisnik unese film, algoritam pronalazi K filmova u datasetu koji su najbliži unesenom filmu na osnovu njihove udaljenosti. Algoritam vraća tih K najbližih filmova kao preporuke.

Cosine similarity

Kosinusna sličnost je mera sličnosti između dva vektora koja se određuje pomoću kosinusa ugla između njih. Prvo, svaki film se predstavlja kao vektor karakteristika. To mogu biti numeričke karakteristike, tekstualne reprezentacije, ili kombinacija obe. Kosinusna sličnost daje vrednost između -1 i 1, gde 1 znači da su vektori potpuno identični, 0 da su potpuno nepovezani, a -1 da su potpuno suprotni. Algoritam pronalazi filmove sa najvećom kosinusnom sličnosti u odnosu na uneti film i vraća ih kao preporuke.

datoteka knn.py

```python
import numpy as np
from sklearn.neighbors import NearestNeighbors


# Initialize a Nearest Neighbors model with 10 neighbors
knn_model = NearestNeighbors(n_neighbors=10, algorithm='auto')


# Function to fit the Nearest Neighbors model with training data
def fit_knn_model(train_data):
    knn_model.fit(train_data)  # Train the KNN model using the training data
```

Inicijalizacija i treniranje KNN modela. Koristimo NearestNeighbors iz sklearn biblioteke za inicijalizaciju i treniranje KNN modela.

datoteka knn.py

```python
def find_similar_movies(features_matrix, input_index, k=10):
    # Find the k-nearest neighbors for the given movie (excluding itself)
    distances, indices = knn_model.kneighbors([features_matrix[input_index]])
    similar_indices = indices[0][1:]  # Exclude the first item (itself) from the list of neighbors
    return similar_indices  # Return indices of similar movies
```

Ova metoda se koristi za pronalaženje k sličnih filmova na osnovu ulaznog filma.

datoteka knn.py

```python
# Function to evaluate the model using Precision@k with a subset of the test data
def precision_at_k(features_matrix, test_indices, train_indices, k=10, subset_size=100):
    precisions = []  # List to store precision values for each subset

    # Randomly select a subset of test indices for evaluation
    subset_indices = np.random.choice(test_indices, size=min(subset_size, len(test_indices)), replace=False)
    for index in subset_indices:
        # All training indices are treated as potential true neighbors
        true_neighbors = set(train_indices)

        # Get recommended neighbors using KNN
        recommended_neighbors = set(find_similar_movies(features_matrix, index, k))

        # Calculate the number of true positives (correct recommendations)
        true_positives = len(true_neighbors & recommended_neighbors)

        # Calculate precision as the ratio of true positives to k
        precision = true_positives / k
        precisions.append(precision)  # Append precision value to the list

    # Return the mean precision for the subset
    return np.mean(precisions)


# Function to compute and print Precision@10
def compute_precision(features_matrix, test_indices, train_indices, k=10):
    precision_k = precision_at_k(features_matrix, test_indices, train_indices, k)  # Calculate Precision@10
    print(f'Precision@10: {precision_k:.4f}')  # Print the result
```

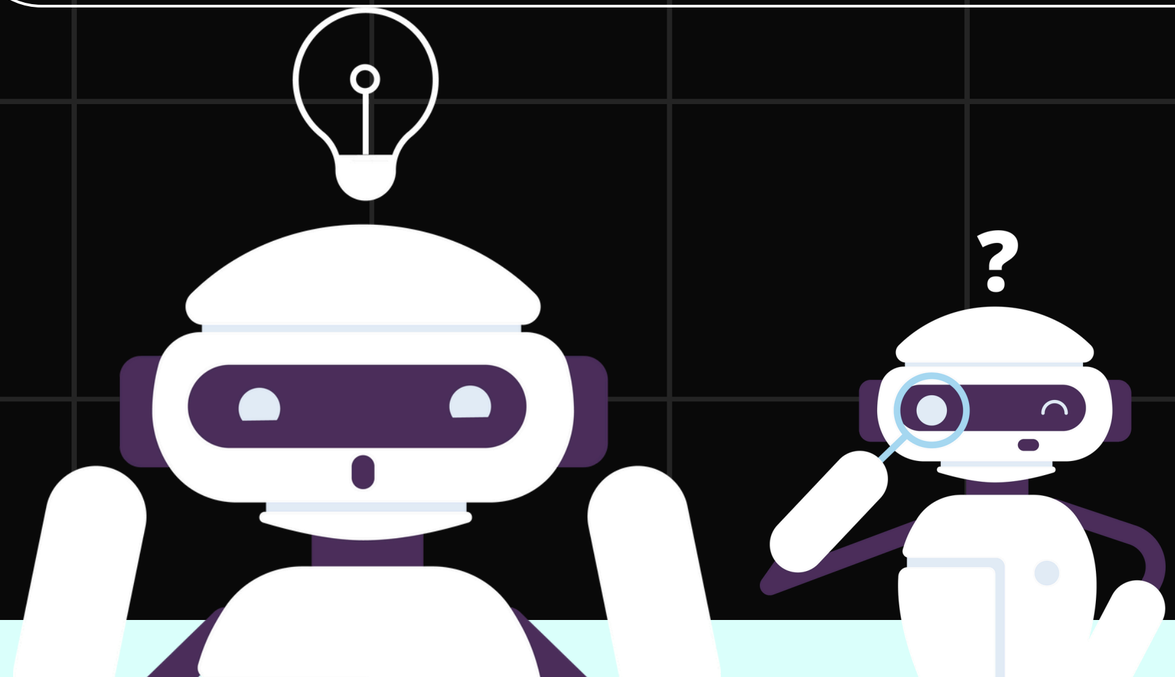
Evaluacija modela koristeći Precision@k funkciju koja meri preciznost preporuka među prvih k preporučenih filmova. Na nasumičan način biramo podskup iz testnog skupa za evaluaciju, dok se svi indeksi iz trening skupa smatraju potencijalnim komšijama. Zatim se dobiju prave komšije pokretanjem funkcije find similar movies i porede se rezultati. Izračunate preciznosti kao odnos pravih prema k se smeštaju u listu, i na kraju se vraća srednja vrednost kada se testira ceo testni podskup.

datoteka cosine_sim.py

```python
from sklearn.metrics.pairwise import cosine_similarity
import numpy as np

# Function to find similar movies using cosine similarity
def find_similar_movies(input_index, features_matrix, k=10):
    # Calculate cosine similarity between the input movie and all other movies
    cosine_sim = cosine_similarity([features_matrix[input_index]], features_matrix)

    # Sort movies by similarity and take the indices of the top k most similar movies
    # Exclude the first element as it is the movie itself
    similar_indices = cosine_sim[0].argsort()[-k-1:-1][::-1]
    return similar_indices  # Return the indices of similar movies
```

Jedina razlika između ova dva algoritma je zapravo ova funkcija. Ona se zasniva na računanju kosinusne sličnosti između unetog filma i ostalih filmova. Sortiramo vrednosti kosinusne sličnosti i uzimamo sve njemu slične filmove koje vraćamo kao rezultat.

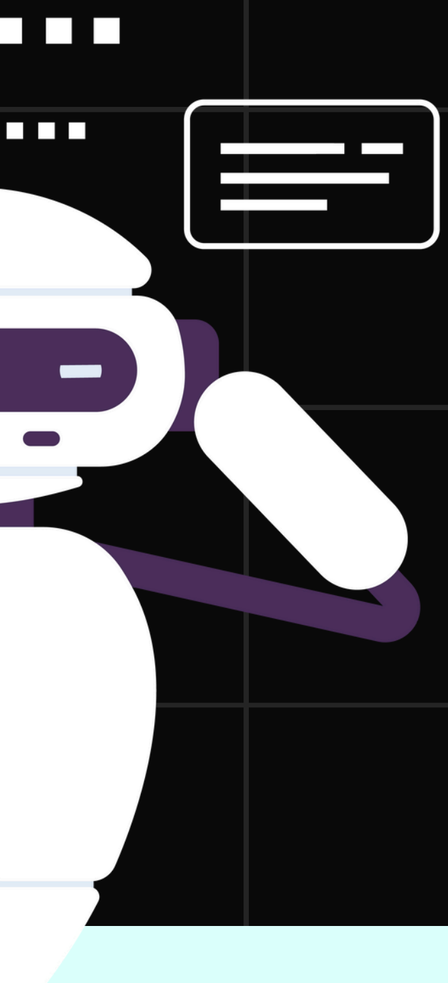
U projektu za preporuku filmova, koristili smo dve metode: KNN (najbliži susedi) i kosinusnu sličnost. KNN funkcioniše tako što pronalazi najbližih K suseda za određeni film na osnovu različitih atributa, dok kosinusna sličnost meri kut između vektora atributa dva filma kako bi se odredila njihova sličnost. Iako obe metode imaju svoje prednosti, kosinusna sličnost se pokazala kao efikasnija u našem slučaju. Kosinusna sličnost bolje hvata semantičku sličnost između filmova jer koristi vektorske reprezentacije, omogućavajući preciznije prepoznavanje sličnih filmova jer se naše upoređivanje prvenstveno zasniva na opisu filmova. Rezultati pokazuju da kosinusna sličnost daje preciznije preporuke, čineći je pogodnijim izborom za ovaj sistem preporuke filmova.



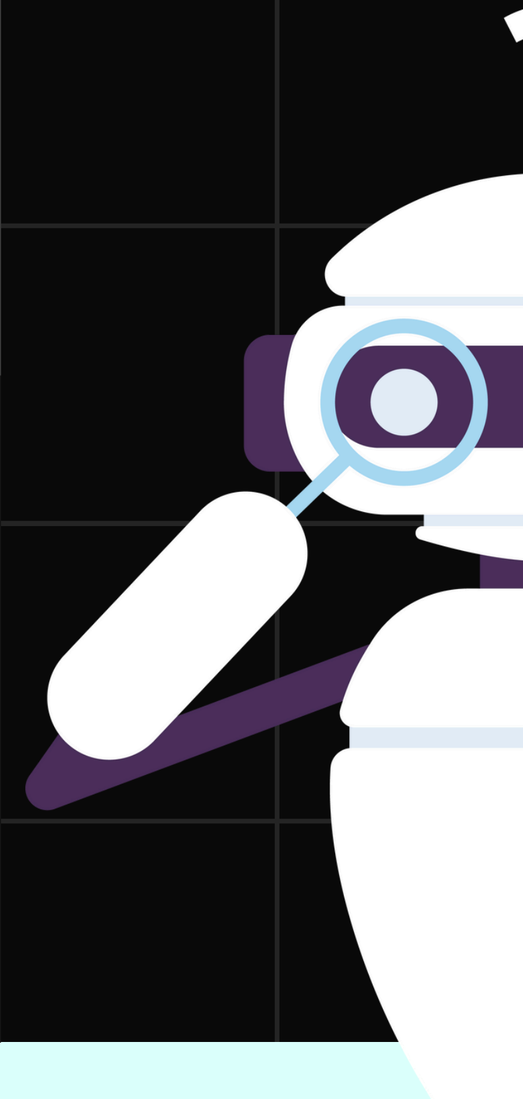
Upoređivanje rezultata

Prikaz rada našeg projekta

-korišćenjem knn algoritma

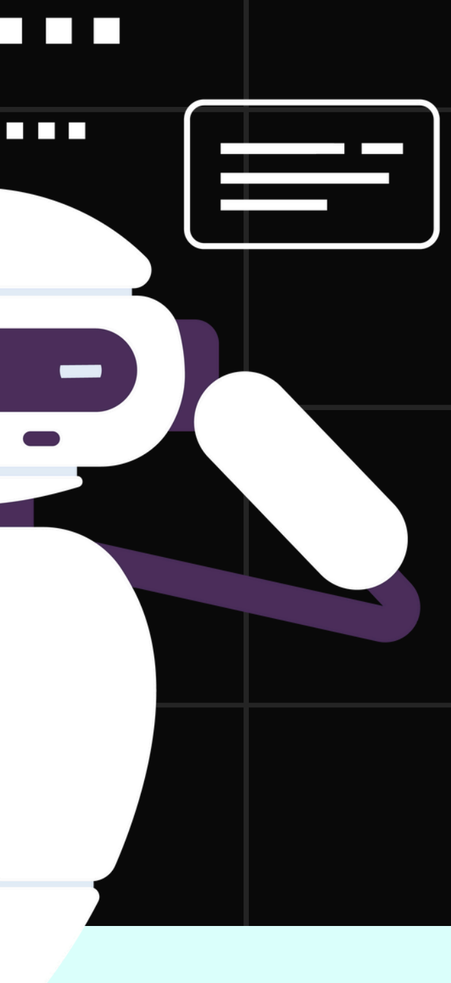


```
Enter a movie name you would like
to find similar movies to: Shutter Island
------------------------------------------------------------------------------------------------------------------------
-------------------------
Choose a model (Knn/cosine): knn
Precision@10: 0.6960

------------------------------------------------------------------------------------------------------------------------
-------------------------
Movie details for 'Shutter Island':

Title: Shutter Island
Overview: World War II soldier-turned-U.S. Marshal Teddy Daniels investigates the disappearance of a patient from a hospital f
or the criminally insane, but his efforts are compromised by his troubling visions and also by a mysterious doctor.

Movies similar to 'Shutter Island':
========================================================================================================================
========================
Title: Crocodile Dundee II
Overview: Australian outback expert protects his New York love from gangsters who've followed her down under.

------------------------------------------------------------------------------------------------------------------------
-------------------------
Title: Star Trek Into Darkness
Overview: When the crew of the Enterprise is called back home, they find an unstoppable force of terror from within their own
organization has detonated the fleet and everything it stands for, leaving our world in a state of crisis.  With a personal sc
ore to settle, Captain Kirk leads a manhunt to a war-zone world to capture a one man weapon of mass destruction. As our heroes
 are propelled into an epic chess game of life and death, love will be challenged, friendships will be torn apart, and sacrifi
ces must be made for the only family Kirk has left: his crew.

------------------------------------------------------------------------------------------------------------------------
-------------------------
Title: Southpaw
Overview: Billy "The Great" Hope, the reigning junior middleweight boxing champion, has an impressive career, a loving wife an
d daughter, and a lavish lifestyle. However, when tragedy strikes, Billy hits rock bottom, losing his family, his house and hi
s manager. He soon finds an unlikely savior in Tick Willis, a former fighter who trains the city's toughest amateur boxers. Wi
th his future on the line, Hope fights to reclaim the trust of those he loves the most.

------------------------------------------------------------------------------------------------------------------------
-------------------------
Title: 2012
```



Prikaz rada našeg projekta

-korišćenjem cosine similarity algoritma



```
Enter a movie name you would like
to find similar movies to: Shutter Island
------------------------------------------------------------------------------------------------------------------------
-------------------------
Choose a model (Knn/cosine): cosine
Precision@10: 0.8130

------------------------------------------------------------------------------------------------------------------------
_________________________
Movie details for 'Shutter Island':

Title: Shutter Island
Overview: World War II soldier-turned-U.S. Marshal Teddy Daniels investigates the disappearance of a patient from a hospital f
or the criminally insane, but his efforts are compromised by his troubling visions and also by a mysterious doctor.

Movies similar to 'Shutter Island':
========================================================================================================================
=========================
Title: Black Book
Overview: In the Nazi-occupied Netherlands during World War II, a Jewish singer infiltrates the regional Gestapo headquarters
for the Dutch resistance.

------------------------------------------------------------------------------------------------------------------------
-------------------------
Title: Miracle at St. Anna
Overview: Miracle at St. Anna chronicles the story of four American soldiers who are members of the all-black 92nd "Buffalo So
ldier" Division stationed in Tuscany, Italy during World War II.

------------------------------------------------------------------------------------------------------------------------
-------------------------
Title: The Thin Red Line
Overview: Based on the graphic novel by James Jones, The Thin Red Line tells the story of a group of men, an Army Rifle compan
y called C-for-Charlie, who change, suffer, and ultimately make essential discoveries about themselves during the fierce World
 War II battle of Guadalcanal. It follows their journey, from the surprise of an unopposed landing, through the bloody and exh
austing battles that follow, to the ultimate departure of those who survived. A powerful frontline cast - including Sean Penn,
 Nick Nolte, Woody Harrelson and George Clooney - explodes into action in this hauntingly realistic view of military and moral
 chaos in the Pacific during World War II.

------------------------------------------------------------------------------------------------------------------------
-------------------------
Title: Catch-22
```

Hvala na pažnji