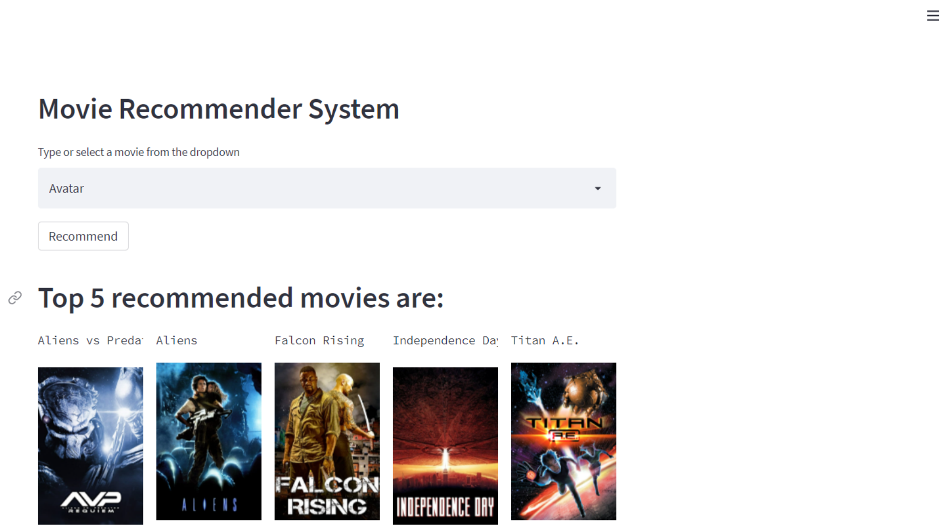
UX Report - Movie Recommendation System

About the web application:

I have made a web application that can recommend movies to the user based on the movie he selects.I have used the python streamlit library for making the web app.I made few of my friends test the application and took the review from them.



Movie Recommender System

Type or select a movie from the dropdown

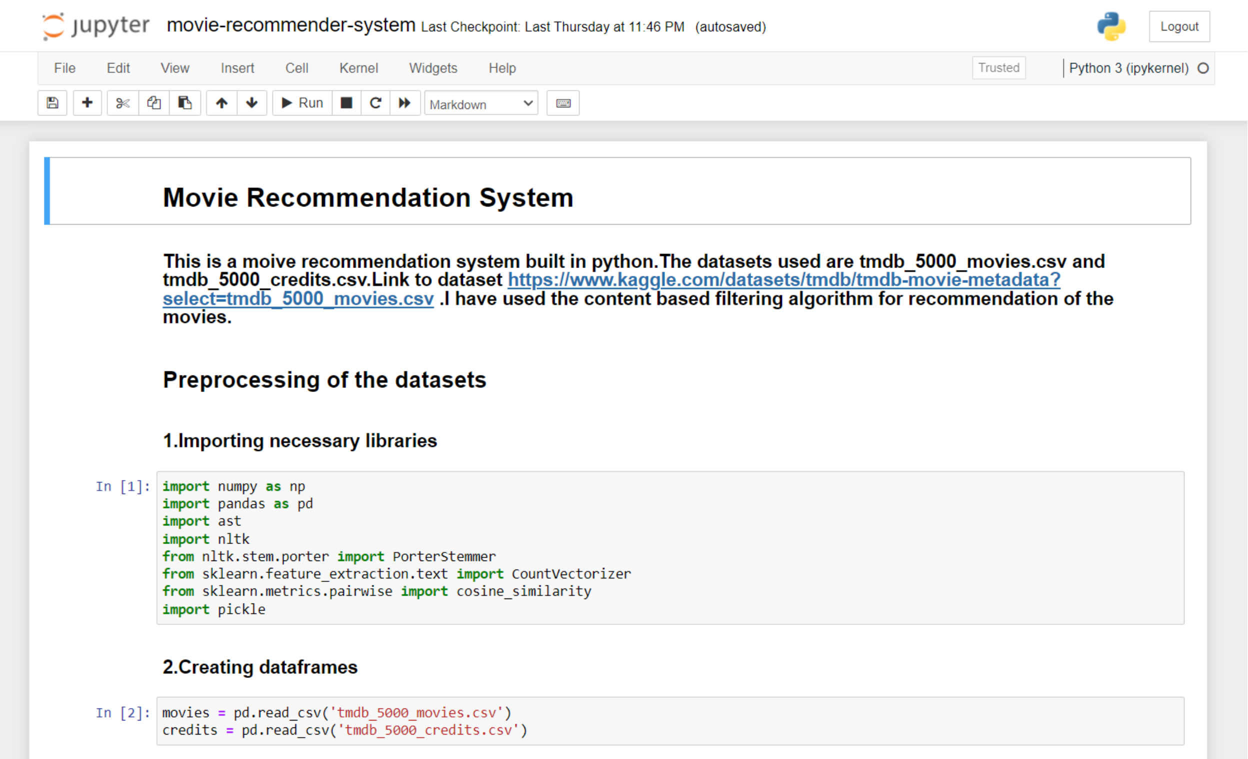
Avatar

Top 5 recommended movies are:

Aliens vs Preda Aliens Falcon Rising Independence Day Titan A.E.

Methodology:

The web application uses content based filtering to recommend movies to the user.I have used 2 datasets from TMDB website.I combined both the datasets and extracted useful features that will be used as tags for filtering. The filtering is then done by finding similarity between these tags of different movies.For the frontend part, streamlit library is used.A simple,clean and basic frontend is made using it.The TMDB api is used for displaying the posters of the recommended movies.


jupyter movie-recommender-system Last Checkpoint: Last Thursday at 11:46 PM (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3 (ipykernel)
Run
Markdown

Movie Recommendation System

This is a moive recommendation system built in python.The datasets used are tmdb_5000_movies.csv and tmdb_5000_credits.csv.Link to dataset https://www.kaggle.com/datasets/tmdb/tmdb-movie-metadata?select=tmdb_5000_movies.csv .I have used the content based filtering algorithm for recommendation of the movies.

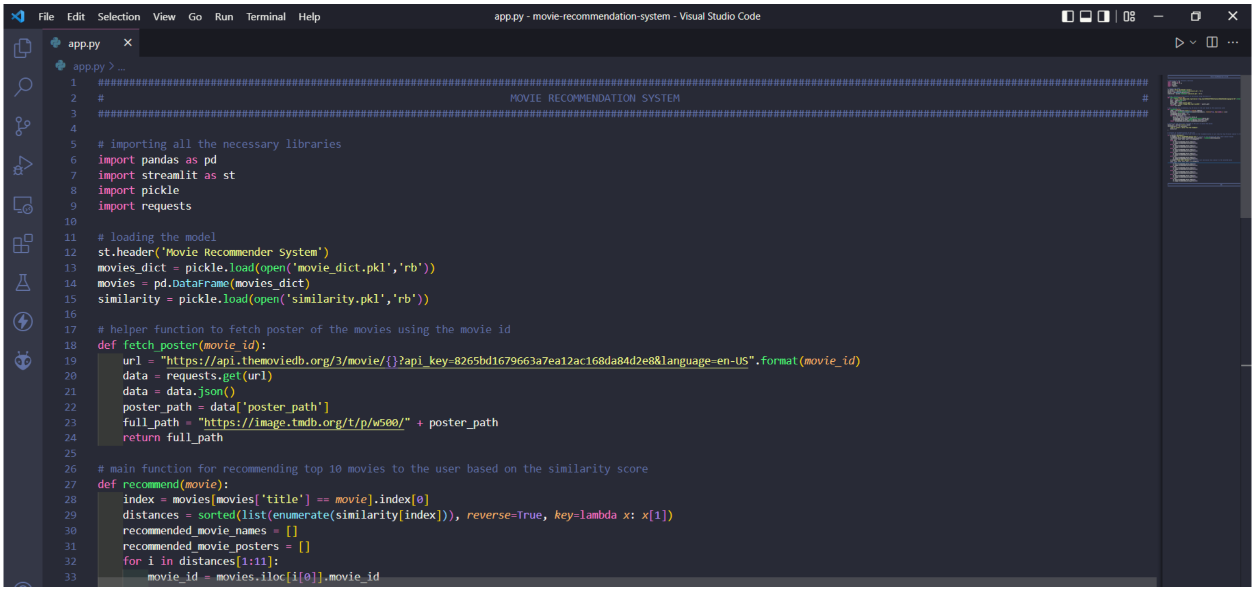
Preprocessing of the datasets

1.Importing necessary libraries

In [1]:
```
import numpy as np
import pandas as pd
import ast
import nltk
from nltk.stem.porter import PorterStemmer
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.metrics.pairwise import cosine_similarity
import pickle
```

2.Creating dataframes

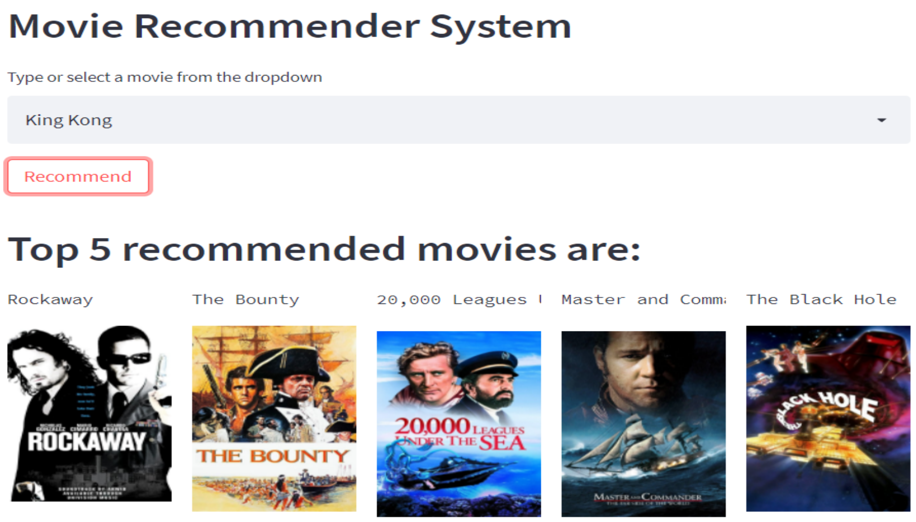
In [2]:
```
movies = pd.read_csv('tmdb_5000_movies.csv')
credits = pd.read_csv('tmdb_5000_credits.csv')
```


File Edit Selection View Go Run Terminal Help
app.py - movie-recommendation-system - Visual Studio Code
app.py
app.py > ...

```
##########################################################################################################################################
#                                                  MOVIE RECOMMENDATION SYSTEM                                                           #
##########################################################################################################################################

# importing all the necessary libraries
import pandas as pd
import streamlit as st
import pickle
import requests

# loading the model
st.header('Movie Recommender System')
movies_dict = pickle.load(open('movie_dict.pkl','rb'))
movies = pd.DataFrame(movies_dict)
similarity = pickle.load(open('similarity.pkl','rb'))

# helper function to fetch poster of the movies using the movie id
def fetch_poster(movie_id):
    url = "https://api.themoviedb.org/3/movie/{}?api_key=8265bd1679663a7ea12ac168da84d2e8&language=en-US".format(movie_id)
    data = requests.get(url)
    data = data.json()
    poster_path = data['poster_path']
    full_path = "https://image.tmdb.org/t/p/w500/" + poster_path
    return full_path

# main function for recommending top 10 movies to the user based on the similarity score
def recommend(movie):
    index = movies[movies['title'] == movie].index[0]
    distances = sorted(list(enumerate(similarity[index])), reverse=True, key=lambda x: x[1])
    recommended_movie_names = []
    recommended_movie_posters = []
    for i in distances[1:11]:
        movie_id = movies.iloc[i[0]].movie_id
```

User experience:

Over 90% of the time similar movies are recommended.The app fails for some movies like King Kong,Harry Potter and the Philosopher's Stone,etc. Overall the UI of the web application is clean and simple.More changes can be made in future.



Movie Recommender System

Type or select a movie from the dropdown

King Kong

Top 5 recommended movies are:

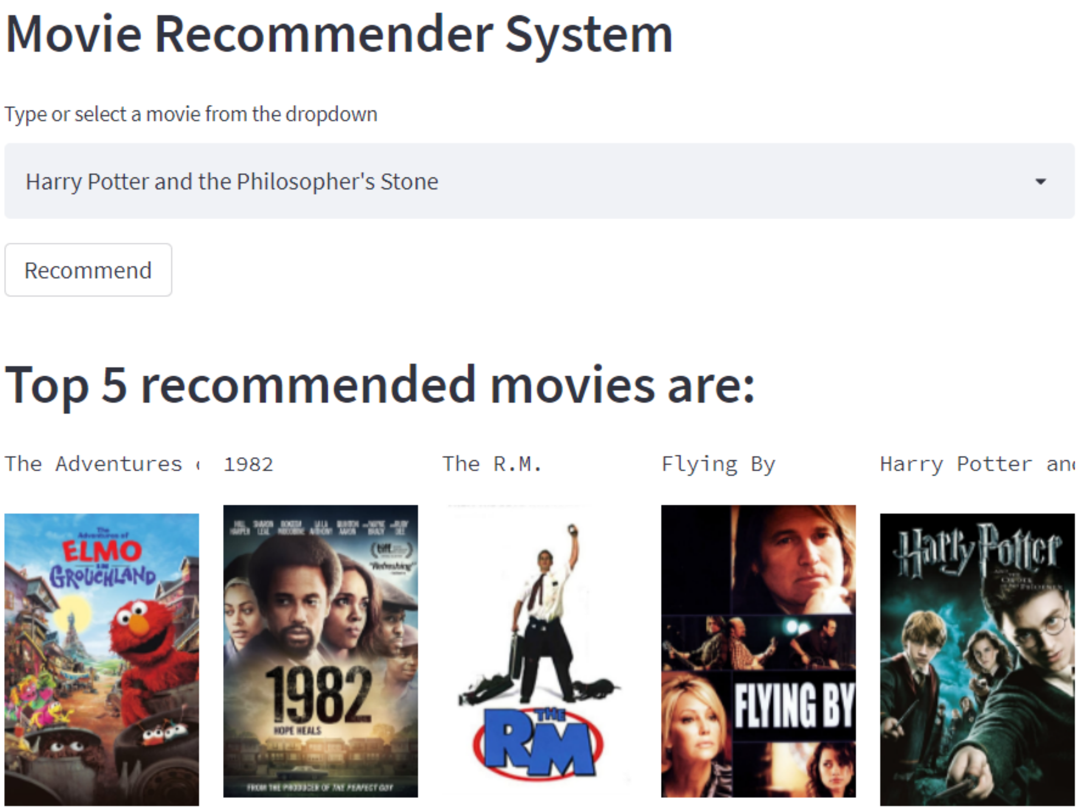
Rockaway

The Bounty

20,000 Leagues

Master and Comm

The Black Hole



Movie Recommender System

Type or select a movie from the dropdown

Harry Potter and the Philosopher's Stone

Top 5 recommended movies are:

The Adventures

1982

The R.M.

Flying By

Harry Potter an

Key Insights:
Most important insight is that the recommendation of the movies is mostly dependent on the genre of the movie.Other factors like cast,director,overview,etc don't contribute much to the recommendation of a movie.

Expandability:
In future updates, we can provide users the option to select on what basis they want to filter movies.This can make the recommendation more accurate and based on the user's decision.

Future Scope:
The frontend can be made more interactive and user friendly.Selected movies by the user can be stored and recommendations can be made using that data.More features can be added like provide user information about the movie.