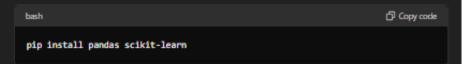
Project Requirements for Execution

- 1. Python:
 - Make sure Python 3.6+ is installed. You can download it from the official Python website.

2. Libraries:

· Install the required Python libraries:



3. Dataset:

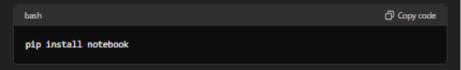
Ensure you have the 'dataset.csv' file. This file should contain the movie data with
columns such as 'id', 'title', 'genre', 'overview', 'popularity', 'release_date',
'vote_average', and 'vote_count'.

4. IDE or Text Editor:

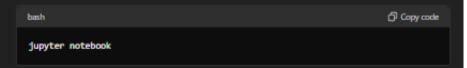
 Use an IDE like PyCharm, VSCode, or a text editor like Sublime Text to edit and run the Python script.

5. Execution Environment:

- · A terminal or command prompt to run the script.
- Alternatively, you can use Jupyter Notebook for running the code in a more interactive manner. To install Jupyter Notebook, use:



Then start Jupyter Notebook with:



6. Pickle for Serialization:

Ensure "pickle" is available (it is part of the Python standard library, so no additional
installation is needed).

Example Script Execution

Below is an example of how to run the code in a script:

```
(7) Copy code
python
Import pandas as pd
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.metrics.pairwise import cosine_similarity
Import pickle
# Load dataset
movies = pd.read_csv('dataset.csv')
# Handling missing values
movies['overview'] = movies['overview'].fillna('')
movies['genre'] = movies['genre'].fillna('')
# Feature selection
movies = movies[['Id', 'title', 'overview', 'genre']]
# Combine 'overview' and 'genre' into a new 'tags' feature
movies[ tags ] = movies[ overview ] + " + movies[ genre']
# Dropping the 'overview' and 'genre' columns
new_data = movies.drop(columns=['overview', 'genre'])
# Text vectorization
cv = CountVectorizer(max_features=18888, stop_words='english')
vector = cv.fit_transform(new_data['tags'].values.astype('U')).toarray()
# Calculate cosine similarity
similarity = cosine_similarity(vector)
# Function to recommend movies
def recommend(movie_title):
   index = new_data[new_data['title'] == movie_title].index[0]
   distances = sorted(list(enumerate(similarity[index])), reverse=True, key=lambda x: x[i]
   for i in distances[1:6]: # Skip the first one as it will be the movie itself
        print(new_data.iloc[i[0]].title)
# Example usage
recommend("Iron Man")
# Serialize the data
pickle.dump(new_data, open('movies_list.pkl', 'wb'))
pickle.dump(similarity, open('similarity.pkl', 'wb'))
# To load the data back
loaded_data = pickle.load(open('movies_list.pkl', 'rb'))
loaded_similarity = pickle.load(open('similarity.pkl', 'rb'))
# Check loaded data
print(loaded_data.head())
```