collaborative filtering

# example code

|  |
| --- |
| import pandas as pd  import numpy as np  import matplotlib.pyplot as plt  import seaborn as sns  from ast import literal\_eval  from sklearn.feature\_extraction.text import TfidfVectorizer  from sklearn.feature\_extraction.text import CountVectorizer  from sklearn.metrics.pairwise import cosine\_similarity  data = pd.read\_csv('ratings\_small.csv')  data.head()  #user-item table  data = data.pivot\_table('rating', index = 'userId', columns = 'movieId')  data.head()  data.shape  #영화 타이틀 merge  ratings = pd.read\_csv('ratings\_small.csv')  movies = pd.read\_csv('tmdb\_5000\_movies.csv')  movies.rename(columns = {'id': 'movieId'}, inplace = True)  ratings\_movies = pd.merge(ratings, movies, on = 'movieId')  ratings\_movies.head(1)  ratings\_movies.shape  #row를 item 기반으로 변경  #cosine similarity를 구할 때 row 기반으로 유사도를 측정하기 때문  data = ratings\_movies.pivot\_table('rating', index = 'userId', columns = 'title').fillna(0)  data.shape  data = data.transpose()  data.head(2)  data.shape  movie\_sim = cosine\_similarity(data, data)  print(movie\_sim.shape)  movie\_sim\_df = pd.DataFrame(data = movie\_sim, index = data.index, columns = data.index)  movie\_sim\_df.head(3)  movie\_sim\_df["X-Men Origins: Wolverine"].sort\_values(ascending=False)[1:10]  movie\_sim\_df["Harry Potter and the Half-Blood Prince"].sort\_values(ascending=False)[1:10]  movie\_sim\_df["Harry Potter and the Half-Blood Prince"].sort\_values(ascending=False)[:10]  movie\_sim\_df["King Kong"].sort\_values(ascending=False)[1:10] |

# testing result

|  |
| --- |
| title  Romeo Must Die 0.649625  The Wedding Planner 0.631669  Dogtown and Z-Boys 0.501189  An Unfinished Life 0.485643  Conquest of the Planet of the Apes 0.474626  Reign Over Me 0.458155  The Terminal 0.445337  Young Frankenstein 0.423840  Whale Rider 0.394136  Name: X-Men Origins: Wolverine, dtype: float64  title  Liar Liar 1.000000  Family Plot 1.000000  Once 1.000000  Synecdoche, New York 1.000000  Rendition 1.000000  Harry Potter and the Half-Blood Prince 1.000000  The Astronaut Farmer 0.970143  Schindler's List 0.724286  The Last King of Scotland 0.707107  Name: Harry Potter and the Half-Blood Prince, dtype: float64  title  The Blue Lagoon 1.000000  Liar Liar 1.000000  Family Plot 1.000000 Once 1.000000  Synecdoche, New York 1.000000  Rendition 1.000000  Harry Potter and the Half-Blood Prince 1.000000  The Astronaut Farmer 0.970143  Schindler's List 0.724286  The Last King of Scotland 0.707107  Name: Harry Potter and the Half-Blood Prince, dtype: float64  title  Fantasia 0.648886  2046 0.648886  Rendition 0.486664  Family Plot 0.486664  Liar Liar 0.486664  Harry Potter and the Half-Blood Prince 0.486664  Synecdoche, New York 0.486664  The Blue Lagoon 0.486664  Once 0.486664 Name: King Kong, dtype: float64 |

# example code

|  |
| --- |
| import pandas as pd  import numpy as np  import pandas as pd  import numpy as np  import matplotlib.pyplot as plt  import seaborn as sns  from ast import literal\_eval  from sklearn.feature\_extraction.text import TfidfVectorizer  from sklearn.feature\_extraction.text import CountVectorizer  from sklearn.metrics.pairwise import cosine\_similarity  rating\_data = pd.read\_csv('ratings.csv')  movie\_data = pd.read\_csv('movies.csv')  rating\_data.head(2)  movie\_data.head(2)  #사용자-영화에 따른 평점 데이터를 pivot-table로 만들기  rating\_data.drop('timestamp', axis = 1, inplace=True)  rating\_data.head(2)  user\_movie\_rating = pd.merge(rating\_data, movie\_data, on = 'movieId')  user\_movie\_rating.head(2)  movie\_user\_rating = user\_movie\_rating.pivot\_table('rating', index = 'title', columns='userId')  user\_movie\_rating = user\_movie\_rating.pivot\_table('rating', index = 'userId', columns='title')  user\_movie\_rating.head(5)  movie\_user\_rating.head()  movie\_user\_rating.fillna(0, inplace = True)  movie\_user\_rating.head(3)  #cosine\_similarity  item\_based\_collabor = cosine\_similarity(movie\_user\_rating)  item\_based\_collabor  print(movie\_user\_rating.shape)  print(item\_based\_collabor.shape)  item\_based\_collabor = pd.DataFrame(data = item\_based\_collabor, index = movie\_user\_rating.index, columns = movie\_user\_rating.index)  item\_based\_collabor.head()  def get\_item\_based\_collabor(title):      return item\_based\_collabor[title].sort\_values(ascending=False)[:6]  get\_item\_based\_collabor('Godfather, The (1972)') |

# testing result

|  |
| --- |
| title  Godfather, The (1972) 1.000000  Godfather: Part II, The (1974) 0.773685  Goodfellas (1990) 0.620349  One Flew Over the Cuckoo's Nest (1975) 0.568244  American Beauty (1999) 0.557997  Star Wars: Episode IV - A New Hope (1977) 0.546750  Name: Godfather, The (1972), dtype: float64 |