|  |
| --- |
| {  Movie\_recommender |
|  |  | "cells": [ |
|  |  | { |
|  |  | "cell\_type": "code", |
|  |  | "execution\_count": 18, |
|  |  | "metadata": {}, |
|  |  | "outputs": [], |
|  |  | "source": [ |
|  |  | "import numpy as np\n", |
|  |  | "import pandas as pd\n", |
|  |  | "import sklearn" |
|  |  | ] |
|  |  | }, |
|  |  | { |
|  |  | "cell\_type": "code", |
|  |  | "execution\_count": 39, |
|  |  | "metadata": {}, |
|  |  | "outputs": [], |
|  |  | "source": [ |
|  |  | "df1= pd.read\_csv('credits.csv')\n", |
|  |  | "df2= pd.read\_csv('movies.csv')" |
|  |  | ] |
|  |  | }, |
|  |  | { |
|  |  | "cell\_type": "code", |
|  |  | "execution\_count": 40, |
|  |  | "metadata": {}, |
|  |  | "outputs": [ |
|  |  | { |
|  |  | "data": { |
|  |  | "text/html": [ |
|  |  | "<div>\n", |
|  |  | "<style scoped>\n", |
|  |  | " .dataframe tbody tr th:only-of-type {\n", |
|  |  | " vertical-align: middle;\n", |
|  |  | " }\n", |
|  |  | "\n", |
|  |  | " .dataframe tbody tr th {\n", |
|  |  | " vertical-align: top;\n", |
|  |  | " }\n", |
|  |  | "\n", |
|  |  | " .dataframe thead th {\n", |
|  |  | " text-align: right;\n", |
|  |  | " }\n", |
|  |  | "</style>\n", |
|  |  | "<table border=\"1\" class=\"dataframe\">\n", |
|  |  | " <thead>\n", |
|  |  | " <tr style=\"text-align: right;\">\n", |
|  |  | " <th></th>\n", |
|  |  | " <th>movie\_id</th>\n", |
|  |  | " <th>title</th>\n", |
|  |  | " <th>cast</th>\n", |
|  |  | " <th>crew</th>\n", |
|  |  | " </tr>\n", |
|  |  | " </thead>\n", |
|  |  | " <tbody>\n", |
|  |  | " <tr>\n", |
|  |  | " <td>0</td>\n", |
|  |  | " <td>19995</td>\n", |
|  |  | " <td>Avatar</td>\n", |
|  |  | " <td>[{\"cast\_id\": 242, \"character\": \"Jake Sully\", \"...</td>\n", |
|  |  | " <td>[{\"credit\_id\": \"52fe48009251416c750aca23\", \"de...</td>\n", |
|  |  | " </tr>\n", |
|  |  | " <tr>\n", |
|  |  | " <td>1</td>\n", |
|  |  | " <td>285</td>\n", |
|  |  | " <td>Pirates of the Caribbean: At World's End</td>\n", |
|  |  | " <td>[{\"cast\_id\": 4, \"character\": \"Captain Jack Spa...</td>\n", |
|  |  | " <td>[{\"credit\_id\": \"52fe4232c3a36847f800b579\", \"de...</td>\n", |
|  |  | " </tr>\n", |
|  |  | " <tr>\n", |
|  |  | " <td>2</td>\n", |
|  |  | " <td>206647</td>\n", |
|  |  | " <td>Spectre</td>\n", |
|  |  | " <td>[{\"cast\_id\": 1, \"character\": \"James Bond\", \"cr...</td>\n", |
|  |  | " <td>[{\"credit\_id\": \"54805967c3a36829b5002c41\", \"de...</td>\n", |
|  |  | " </tr>\n", |
|  |  | " <tr>\n", |
|  |  | " <td>3</td>\n", |
|  |  | " <td>49026</td>\n", |
|  |  | " <td>The Dark Knight Rises</td>\n", |
|  |  | " <td>[{\"cast\_id\": 2, \"character\": \"Bruce Wayne / Ba...</td>\n", |
|  |  | " <td>[{\"credit\_id\": \"52fe4781c3a36847f81398c3\", \"de...</td>\n", |
|  |  | " </tr>\n", |
|  |  | " <tr>\n", |
|  |  | " <td>4</td>\n", |
|  |  | " <td>49529</td>\n", |
|  |  | " <td>John Carter</td>\n", |
|  |  | " <td>[{\"cast\_id\": 5, \"character\": \"John Carter\", \"c...</td>\n", |
|  |  | " <td>[{\"credit\_id\": \"52fe479ac3a36847f813eaa3\", \"de...</td>\n", |
|  |  | " </tr>\n", |
|  |  | " </tbody>\n", |
|  |  | "</table>\n", |
|  |  | "</div>" |
|  |  | ], |
|  |  | "text/plain": [ |
|  |  | " movie\_id title \\\n", |
|  |  | "0 19995 Avatar \n", |
|  |  | "1 285 Pirates of the Caribbean: At World's End \n", |
|  |  | "2 206647 Spectre \n", |
|  |  | "3 49026 The Dark Knight Rises \n", |
|  |  | "4 49529 John Carter \n", |
|  |  | "\n", |
|  |  | " cast \\\n", |
|  |  | "0 [{\"cast\_id\": 242, \"character\": \"Jake Sully\", \"... \n", |
|  |  | "1 [{\"cast\_id\": 4, \"character\": \"Captain Jack Spa... \n", |
|  |  | "2 [{\"cast\_id\": 1, \"character\": \"James Bond\", \"cr... \n", |
|  |  | "3 [{\"cast\_id\": 2, \"character\": \"Bruce Wayne / Ba... \n", |
|  |  | "4 [{\"cast\_id\": 5, \"character\": \"John Carter\", \"c... \n", |
|  |  | "\n", |
|  |  | " crew \n", |
|  |  | "0 [{\"credit\_id\": \"52fe48009251416c750aca23\", \"de... \n", |
|  |  | "1 [{\"credit\_id\": \"52fe4232c3a36847f800b579\", \"de... \n", |
|  |  | "2 [{\"credit\_id\": \"54805967c3a36829b5002c41\", \"de... \n", |
|  |  | "3 [{\"credit\_id\": \"52fe4781c3a36847f81398c3\", \"de... \n", |
|  |  | "4 [{\"credit\_id\": \"52fe479ac3a36847f813eaa3\", \"de... " |
|  |  | ] |
|  |  | }, |
|  |  | "execution\_count": 40, |
|  |  | "metadata": {}, |
|  |  | "output\_type": "execute\_result" |
|  |  | } |
|  |  | ], |
|  |  | "source": [ |
|  |  | "df1.head()" |
|  |  | ] |
|  |  | }, |
|  |  | { |
|  |  | "cell\_type": "code", |
|  |  | "execution\_count": 41, |
|  |  | "metadata": {}, |
|  |  | "outputs": [ |
|  |  | { |
|  |  | "data": { |
|  |  | "text/html": [ |
|  |  | "<div>\n", |
|  |  | "<style scoped>\n", |
|  |  | " .dataframe tbody tr th:only-of-type {\n", |
|  |  | " vertical-align: middle;\n", |
|  |  | " }\n", |
|  |  | "\n", |
|  |  | " .dataframe tbody tr th {\n", |
|  |  | " vertical-align: top;\n", |
|  |  | " }\n", |
|  |  | "\n", |
|  |  | " .dataframe thead th {\n", |
|  |  | " text-align: right;\n", |
|  |  | " }\n", |
|  |  | "</style>\n", |
|  |  | "<table border=\"1\" class=\"dataframe\">\n", |
|  |  | " <thead>\n", |
|  |  | " <tr style=\"text-align: right;\">\n", |
|  |  | " <th></th>\n", |
|  |  | " <th>budget</th>\n", |
|  |  | " <th>genres</th>\n", |
|  |  | " <th>homepage</th>\n", |
|  |  | " <th>id</th>\n", |
|  |  | " <th>keywords</th>\n", |
|  |  | " <th>original\_language</th>\n", |
|  |  | " <th>original\_title</th>\n", |
|  |  | " <th>overview</th>\n", |
|  |  | " <th>popularity</th>\n", |
|  |  | " <th>production\_companies</th>\n", |
|  |  | " <th>production\_countries</th>\n", |
|  |  | " <th>release\_date</th>\n", |
|  |  | " <th>revenue</th>\n", |
|  |  | " <th>runtime</th>\n", |
|  |  | " <th>spoken\_languages</th>\n", |
|  |  | " <th>status</th>\n", |
|  |  | " <th>tagline</th>\n", |
|  |  | " <th>title</th>\n", |
|  |  | " <th>vote\_average</th>\n", |
|  |  | " <th>vote\_count</th>\n", |
|  |  | " </tr>\n", |
|  |  | " </thead>\n", |
|  |  | " <tbody>\n", |
|  |  | " <tr>\n", |
|  |  | " <td>0</td>\n", |
|  |  | " <td>237000000</td>\n", |
|  |  | " <td>[{\"id\": 28, \"name\": \"Action\"}, {\"id\": 12, \"nam...</td>\n", |
|  |  | " <td>http://www.avatarmovie.com/</td>\n", |
|  |  | " <td>19995</td>\n", |
|  |  | " <td>[{\"id\": 1463, \"name\": \"culture clash\"}, {\"id\":...</td>\n", |
|  |  | " <td>en</td>\n", |
|  |  | " <td>Avatar</td>\n", |
|  |  | " <td>In the 22nd century, a paraplegic Marine is di...</td>\n", |
|  |  | " <td>150.437577</td>\n", |
|  |  | " <td>[{\"name\": \"Ingenious Film Partners\", \"id\": 289...</td>\n", |
|  |  | " <td>[{\"iso\_3166\_1\": \"US\", \"name\": \"United States o...</td>\n", |
|  |  | " <td>2009-12-10</td>\n", |
|  |  | " <td>2787965087</td>\n", |
|  |  | " <td>162.0</td>\n", |
|  |  | " <td>[{\"iso\_639\_1\": \"en\", \"name\": \"English\"}, {\"iso...</td>\n", |
|  |  | " <td>Released</td>\n", |
|  |  | " <td>Enter the World of Pandora.</td>\n", |
|  |  | " <td>Avatar</td>\n", |
|  |  | " <td>7.2</td>\n", |
|  |  | " <td>11800</td>\n", |
|  |  | " </tr>\n", |
|  |  | " <tr>\n", |
|  |  | " <td>1</td>\n", |
|  |  | " <td>300000000</td>\n", |
|  |  | " <td>[{\"id\": 12, \"name\": \"Adventure\"}, {\"id\": 14, \"...</td>\n", |
|  |  | " <td>http://disney.go.com/disneypictures/pirates/</td>\n", |
|  |  | " <td>285</td>\n", |
|  |  | " <td>[{\"id\": 270, \"name\": \"ocean\"}, {\"id\": 726, \"na...</td>\n", |
|  |  | " <td>en</td>\n", |
|  |  | " <td>Pirates of the Caribbean: At World's End</td>\n", |
|  |  | " <td>Captain Barbossa, long believed to be dead, ha...</td>\n", |
|  |  | " <td>139.082615</td>\n", |
|  |  | " <td>[{\"name\": \"Walt Disney Pictures\", \"id\": 2}, {\"...</td>\n", |
|  |  | " <td>[{\"iso\_3166\_1\": \"US\", \"name\": \"United States o...</td>\n", |
|  |  | " <td>2007-05-19</td>\n", |
|  |  | " <td>961000000</td>\n", |
|  |  | " <td>169.0</td>\n", |
|  |  | " <td>[{\"iso\_639\_1\": \"en\", \"name\": \"English\"}]</td>\n", |
|  |  | " <td>Released</td>\n", |
|  |  | " <td>At the end of the world, the adventure begins.</td>\n", |
|  |  | " <td>Pirates of the Caribbean: At World's End</td>\n", |
|  |  | " <td>6.9</td>\n", |
|  |  | " <td>4500</td>\n", |
|  |  | " </tr>\n", |
|  |  | " <tr>\n", |
|  |  | " <td>2</td>\n", |
|  |  | " <td>245000000</td>\n", |
|  |  | " <td>[{\"id\": 28, \"name\": \"Action\"}, {\"id\": 12, \"nam...</td>\n", |
|  |  | " <td>http://www.sonypictures.com/movies/spectre/</td>\n", |
|  |  | " <td>206647</td>\n", |
|  |  | " <td>[{\"id\": 470, \"name\": \"spy\"}, {\"id\": 818, \"name...</td>\n", |
|  |  | " <td>en</td>\n", |
|  |  | " <td>Spectre</td>\n", |
|  |  | " <td>A cryptic message from Bond’s past sends him o...</td>\n", |
|  |  | " <td>107.376788</td>\n", |
|  |  | " <td>[{\"name\": \"Columbia Pictures\", \"id\": 5}, {\"nam...</td>\n", |
|  |  | " <td>[{\"iso\_3166\_1\": \"GB\", \"name\": \"United Kingdom\"...</td>\n", |
|  |  | " <td>2015-10-26</td>\n", |
|  |  | " <td>880674609</td>\n", |
|  |  | " <td>148.0</td>\n", |
|  |  | " <td>[{\"iso\_639\_1\": \"fr\", \"name\": \"Fran\\u00e7ais\"},...</td>\n", |
|  |  | " <td>Released</td>\n", |
|  |  | " <td>A Plan No One Escapes</td>\n", |
|  |  | " <td>Spectre</td>\n", |
|  |  | " <td>6.3</td>\n", |
|  |  | " <td>4466</td>\n", |
|  |  | " </tr>\n", |
|  |  | " <tr>\n", |
|  |  | " <td>3</td>\n", |
|  |  | " <td>250000000</td>\n", |
|  |  | " <td>[{\"id\": 28, \"name\": \"Action\"}, {\"id\": 80, \"nam...</td>\n", |
|  |  | " <td>http://www.thedarkknightrises.com/</td>\n", |
|  |  | " <td>49026</td>\n", |
|  |  | " <td>[{\"id\": 849, \"name\": \"dc comics\"}, {\"id\": 853,...</td>\n", |
|  |  | " <td>en</td>\n", |
|  |  | " <td>The Dark Knight Rises</td>\n", |
|  |  | " <td>Following the death of District Attorney Harve...</td>\n", |
|  |  | " <td>112.312950</td>\n", |
|  |  | " <td>[{\"name\": \"Legendary Pictures\", \"id\": 923}, {\"...</td>\n", |
|  |  | " <td>[{\"iso\_3166\_1\": \"US\", \"name\": \"United States o...</td>\n", |
|  |  | " <td>2012-07-16</td>\n", |
|  |  | " <td>1084939099</td>\n", |
|  |  | " <td>165.0</td>\n", |
|  |  | " <td>[{\"iso\_639\_1\": \"en\", \"name\": \"English\"}]</td>\n", |
|  |  | " <td>Released</td>\n", |
|  |  | " <td>The Legend Ends</td>\n", |
|  |  | " <td>The Dark Knight Rises</td>\n", |
|  |  | " <td>7.6</td>\n", |
|  |  | " <td>9106</td>\n", |
|  |  | " </tr>\n", |
|  |  | " <tr>\n", |
|  |  | " <td>4</td>\n", |
|  |  | " <td>260000000</td>\n", |
|  |  | " <td>[{\"id\": 28, \"name\": \"Action\"}, {\"id\": 12, \"nam...</td>\n", |
|  |  | " <td>http://movies.disney.com/john-carter</td>\n", |
|  |  | " <td>49529</td>\n", |
|  |  | " <td>[{\"id\": 818, \"name\": \"based on novel\"}, {\"id\":...</td>\n", |
|  |  | " <td>en</td>\n", |
|  |  | " <td>John Carter</td>\n", |
|  |  | " <td>John Carter is a war-weary, former military ca...</td>\n", |
|  |  | " <td>43.926995</td>\n", |
|  |  | " <td>[{\"name\": \"Walt Disney Pictures\", \"id\": 2}]</td>\n", |
|  |  | " <td>[{\"iso\_3166\_1\": \"US\", \"name\": \"United States o...</td>\n", |
|  |  | " <td>2012-03-07</td>\n", |
|  |  | " <td>284139100</td>\n", |
|  |  | " <td>132.0</td>\n", |
|  |  | " <td>[{\"iso\_639\_1\": \"en\", \"name\": \"English\"}]</td>\n", |
|  |  | " <td>Released</td>\n", |
|  |  | " <td>Lost in our world, found in another.</td>\n", |
|  |  | " <td>John Carter</td>\n", |
|  |  | " <td>6.1</td>\n", |
|  |  | " <td>2124</td>\n", |
|  |  | " </tr>\n", |
|  |  | " </tbody>\n", |
|  |  | "</table>\n", |
|  |  | "</div>" |
|  |  | ], |
|  |  | "text/plain": [ |
|  |  | " budget genres \\\n", |
|  |  | "0 237000000 [{\"id\": 28, \"name\": \"Action\"}, {\"id\": 12, \"nam... \n", |
|  |  | "1 300000000 [{\"id\": 12, \"name\": \"Adventure\"}, {\"id\": 14, \"... \n", |
|  |  | "2 245000000 [{\"id\": 28, \"name\": \"Action\"}, {\"id\": 12, \"nam... \n", |
|  |  | "3 250000000 [{\"id\": 28, \"name\": \"Action\"}, {\"id\": 80, \"nam... \n", |
|  |  | "4 260000000 [{\"id\": 28, \"name\": \"Action\"}, {\"id\": 12, \"nam... \n", |
|  |  | "\n", |
|  |  | " homepage id \\\n", |
|  |  | "0 http://www.avatarmovie.com/ 19995 \n", |
|  |  | "1 http://disney.go.com/disneypictures/pirates/ 285 \n", |
|  |  | "2 http://www.sonypictures.com/movies/spectre/ 206647 \n", |
|  |  | "3 http://www.thedarkknightrises.com/ 49026 \n", |
|  |  | "4 http://movies.disney.com/john-carter 49529 \n", |
|  |  | "\n", |
|  |  | " keywords original\_language \\\n", |
|  |  | "0 [{\"id\": 1463, \"name\": \"culture clash\"}, {\"id\":... en \n", |
|  |  | "1 [{\"id\": 270, \"name\": \"ocean\"}, {\"id\": 726, \"na... en \n", |
|  |  | "2 [{\"id\": 470, \"name\": \"spy\"}, {\"id\": 818, \"name... en \n", |
|  |  | "3 [{\"id\": 849, \"name\": \"dc comics\"}, {\"id\": 853,... en \n", |
|  |  | "4 [{\"id\": 818, \"name\": \"based on novel\"}, {\"id\":... en \n", |
|  |  | "\n", |
|  |  | " original\_title \\\n", |
|  |  | "0 Avatar \n", |
|  |  | "1 Pirates of the Caribbean: At World's End \n", |
|  |  | "2 Spectre \n", |
|  |  | "3 The Dark Knight Rises \n", |
|  |  | "4 John Carter \n", |
|  |  | "\n", |
|  |  | " overview popularity \\\n", |
|  |  | "0 In the 22nd century, a paraplegic Marine is di... 150.437577 \n", |
|  |  | "1 Captain Barbossa, long believed to be dead, ha... 139.082615 \n", |
|  |  | "2 A cryptic message from Bond’s past sends him o... 107.376788 \n", |
|  |  | "3 Following the death of District Attorney Harve... 112.312950 \n", |
|  |  | "4 John Carter is a war-weary, former military ca... 43.926995 \n", |
|  |  | "\n", |
|  |  | " production\_companies \\\n", |
|  |  | "0 [{\"name\": \"Ingenious Film Partners\", \"id\": 289... \n", |
|  |  | "1 [{\"name\": \"Walt Disney Pictures\", \"id\": 2}, {\"... \n", |
|  |  | "2 [{\"name\": \"Columbia Pictures\", \"id\": 5}, {\"nam... \n", |
|  |  | "3 [{\"name\": \"Legendary Pictures\", \"id\": 923}, {\"... \n", |
|  |  | "4 [{\"name\": \"Walt Disney Pictures\", \"id\": 2}] \n", |
|  |  | "\n", |
|  |  | " production\_countries release\_date revenue \\\n", |
|  |  | "0 [{\"iso\_3166\_1\": \"US\", \"name\": \"United States o... 2009-12-10 2787965087 \n", |
|  |  | "1 [{\"iso\_3166\_1\": \"US\", \"name\": \"United States o... 2007-05-19 961000000 \n", |
|  |  | "2 [{\"iso\_3166\_1\": \"GB\", \"name\": \"United Kingdom\"... 2015-10-26 880674609 \n", |
|  |  | "3 [{\"iso\_3166\_1\": \"US\", \"name\": \"United States o... 2012-07-16 1084939099 \n", |
|  |  | "4 [{\"iso\_3166\_1\": \"US\", \"name\": \"United States o... 2012-03-07 284139100 \n", |
|  |  | "\n", |
|  |  | " runtime spoken\_languages status \\\n", |
|  |  | "0 162.0 [{\"iso\_639\_1\": \"en\", \"name\": \"English\"}, {\"iso... Released \n", |
|  |  | "1 169.0 [{\"iso\_639\_1\": \"en\", \"name\": \"English\"}] Released \n", |
|  |  | "2 148.0 [{\"iso\_639\_1\": \"fr\", \"name\": \"Fran\\u00e7ais\"},... Released \n", |
|  |  | "3 165.0 [{\"iso\_639\_1\": \"en\", \"name\": \"English\"}] Released \n", |
|  |  | "4 132.0 [{\"iso\_639\_1\": \"en\", \"name\": \"English\"}] Released \n", |
|  |  | "\n", |
|  |  | " tagline \\\n", |
|  |  | "0 Enter the World of Pandora. \n", |
|  |  | "1 At the end of the world, the adventure begins. \n", |
|  |  | "2 A Plan No One Escapes \n", |
|  |  | "3 The Legend Ends \n", |
|  |  | "4 Lost in our world, found in another. \n", |
|  |  | "\n", |
|  |  | " title vote\_average vote\_count \n", |
|  |  | "0 Avatar 7.2 11800 \n", |
|  |  | "1 Pirates of the Caribbean: At World's End 6.9 4500 \n", |
|  |  | "2 Spectre 6.3 4466 \n", |
|  |  | "3 The Dark Knight Rises 7.6 9106 \n", |
|  |  | "4 John Carter 6.1 2124 " |
|  |  | ] |
|  |  | }, |
|  |  | "execution\_count": 41, |
|  |  | "metadata": {}, |
|  |  | "output\_type": "execute\_result" |
|  |  | } |
|  |  | ], |
|  |  | "source": [ |
|  |  | "df2.head()" |
|  |  | ] |
|  |  | }, |
|  |  | { |
|  |  | "cell\_type": "code", |
|  |  | "execution\_count": 42, |
|  |  | "metadata": {}, |
|  |  | "outputs": [], |
|  |  | "source": [ |
|  |  | "# joining two datasets\n", |
|  |  | "df1.columns=['id','tittle','cast','crew']\n", |
|  |  | "df2=df2.merge(df1,on='id')" |
|  |  | ] |
|  |  | }, |
|  |  | { |
|  |  | "cell\_type": "code", |
|  |  | "execution\_count": 43, |
|  |  | "metadata": {}, |
|  |  | "outputs": [ |
|  |  | { |
|  |  | "data": { |
|  |  | "text/html": [ |
|  |  | "<div>\n", |
|  |  | "<style scoped>\n", |
|  |  | " .dataframe tbody tr th:only-of-type {\n", |
|  |  | " vertical-align: middle;\n", |
|  |  | " }\n", |
|  |  | "\n", |
|  |  | " .dataframe tbody tr th {\n", |
|  |  | " vertical-align: top;\n", |
|  |  | " }\n", |
|  |  | "\n", |
|  |  | " .dataframe thead th {\n", |
|  |  | " text-align: right;\n", |
|  |  | " }\n", |
|  |  | "</style>\n", |
|  |  | "<table border=\"1\" class=\"dataframe\">\n", |
|  |  | " <thead>\n", |
|  |  | " <tr style=\"text-align: right;\">\n", |
|  |  | " <th></th>\n", |
|  |  | " <th>budget</th>\n", |
|  |  | " <th>genres</th>\n", |
|  |  | " <th>homepage</th>\n", |
|  |  | " <th>id</th>\n", |
|  |  | " <th>keywords</th>\n", |
|  |  | " <th>original\_language</th>\n", |
|  |  | " <th>original\_title</th>\n", |
|  |  | " <th>overview</th>\n", |
|  |  | " <th>popularity</th>\n", |
|  |  | " <th>production\_companies</th>\n", |
|  |  | " <th>...</th>\n", |
|  |  | " <th>runtime</th>\n", |
|  |  | " <th>spoken\_languages</th>\n", |
|  |  | " <th>status</th>\n", |
|  |  | " <th>tagline</th>\n", |
|  |  | " <th>title</th>\n", |
|  |  | " <th>vote\_average</th>\n", |
|  |  | " <th>vote\_count</th>\n", |
|  |  | " <th>tittle</th>\n", |
|  |  | " <th>cast</th>\n", |
|  |  | " <th>crew</th>\n", |
|  |  | " </tr>\n", |
|  |  | " </thead>\n", |
|  |  | " <tbody>\n", |
|  |  | " <tr>\n", |
|  |  | " <td>0</td>\n", |
|  |  | " <td>237000000</td>\n", |
|  |  | " <td>[{\"id\": 28, \"name\": \"Action\"}, {\"id\": 12, \"nam...</td>\n", |
|  |  | " <td>http://www.avatarmovie.com/</td>\n", |
|  |  | " <td>19995</td>\n", |
|  |  | " <td>[{\"id\": 1463, \"name\": \"culture clash\"}, {\"id\":...</td>\n", |
|  |  | " <td>en</td>\n", |
|  |  | " <td>Avatar</td>\n", |
|  |  | " <td>In the 22nd century, a paraplegic Marine is di...</td>\n", |
|  |  | " <td>150.437577</td>\n", |
|  |  | " <td>[{\"name\": \"Ingenious Film Partners\", \"id\": 289...</td>\n", |
|  |  | " <td>...</td>\n", |
|  |  | " <td>162.0</td>\n", |
|  |  | " <td>[{\"iso\_639\_1\": \"en\", \"name\": \"English\"}, {\"iso...</td>\n", |
|  |  | " <td>Released</td>\n", |
|  |  | " <td>Enter the World of Pandora.</td>\n", |
|  |  | " <td>Avatar</td>\n", |
|  |  | " <td>7.2</td>\n", |
|  |  | " <td>11800</td>\n", |
|  |  | " <td>Avatar</td>\n", |
|  |  | " <td>[{\"cast\_id\": 242, \"character\": \"Jake Sully\", \"...</td>\n", |
|  |  | " <td>[{\"credit\_id\": \"52fe48009251416c750aca23\", \"de...</td>\n", |
|  |  | " </tr>\n", |
|  |  | " <tr>\n", |
|  |  | " <td>1</td>\n", |
|  |  | " <td>300000000</td>\n", |
|  |  | " <td>[{\"id\": 12, \"name\": \"Adventure\"}, {\"id\": 14, \"...</td>\n", |
|  |  | " <td>http://disney.go.com/disneypictures/pirates/</td>\n", |
|  |  | " <td>285</td>\n", |
|  |  | " <td>[{\"id\": 270, \"name\": \"ocean\"}, {\"id\": 726, \"na...</td>\n", |
|  |  | " <td>en</td>\n", |
|  |  | " <td>Pirates of the Caribbean: At World's End</td>\n", |
|  |  | " <td>Captain Barbossa, long believed to be dead, ha...</td>\n", |
|  |  | " <td>139.082615</td>\n", |
|  |  | " <td>[{\"name\": \"Walt Disney Pictures\", \"id\": 2}, {\"...</td>\n", |
|  |  | " <td>...</td>\n", |
|  |  | " <td>169.0</td>\n", |
|  |  | " <td>[{\"iso\_639\_1\": \"en\", \"name\": \"English\"}]</td>\n", |
|  |  | " <td>Released</td>\n", |
|  |  | " <td>At the end of the world, the adventure begins.</td>\n", |
|  |  | " <td>Pirates of the Caribbean: At World's End</td>\n", |
|  |  | " <td>6.9</td>\n", |
|  |  | " <td>4500</td>\n", |
|  |  | " <td>Pirates of the Caribbean: At World's End</td>\n", |
|  |  | " <td>[{\"cast\_id\": 4, \"character\": \"Captain Jack Spa...</td>\n", |
|  |  | " <td>[{\"credit\_id\": \"52fe4232c3a36847f800b579\", \"de...</td>\n", |
|  |  | " </tr>\n", |
|  |  | " <tr>\n", |
|  |  | " <td>2</td>\n", |
|  |  | " <td>245000000</td>\n", |
|  |  | " <td>[{\"id\": 28, \"name\": \"Action\"}, {\"id\": 12, \"nam...</td>\n", |
|  |  | " <td>http://www.sonypictures.com/movies/spectre/</td>\n", |
|  |  | " <td>206647</td>\n", |
|  |  | " <td>[{\"id\": 470, \"name\": \"spy\"}, {\"id\": 818, \"name...</td>\n", |
|  |  | " <td>en</td>\n", |
|  |  | " <td>Spectre</td>\n", |
|  |  | " <td>A cryptic message from Bond’s past sends him o...</td>\n", |
|  |  | " <td>107.376788</td>\n", |
|  |  | " <td>[{\"name\": \"Columbia Pictures\", \"id\": 5}, {\"nam...</td>\n", |
|  |  | " <td>...</td>\n", |
|  |  | " <td>148.0</td>\n", |
|  |  | " <td>[{\"iso\_639\_1\": \"fr\", \"name\": \"Fran\\u00e7ais\"},...</td>\n", |
|  |  | " <td>Released</td>\n", |
|  |  | " <td>A Plan No One Escapes</td>\n", |
|  |  | " <td>Spectre</td>\n", |
|  |  | " <td>6.3</td>\n", |
|  |  | " <td>4466</td>\n", |
|  |  | " <td>Spectre</td>\n", |
|  |  | " <td>[{\"cast\_id\": 1, \"character\": \"James Bond\", \"cr...</td>\n", |
|  |  | " <td>[{\"credit\_id\": \"54805967c3a36829b5002c41\", \"de...</td>\n", |
|  |  | " </tr>\n", |
|  |  | " <tr>\n", |
|  |  | " <td>3</td>\n", |
|  |  | " <td>250000000</td>\n", |
|  |  | " <td>[{\"id\": 28, \"name\": \"Action\"}, {\"id\": 80, \"nam...</td>\n", |
|  |  | " <td>http://www.thedarkknightrises.com/</td>\n", |
|  |  | " <td>49026</td>\n", |
|  |  | " <td>[{\"id\": 849, \"name\": \"dc comics\"}, {\"id\": 853,...</td>\n", |
|  |  | " <td>en</td>\n", |
|  |  | " <td>The Dark Knight Rises</td>\n", |
|  |  | " <td>Following the death of District Attorney Harve...</td>\n", |
|  |  | " <td>112.312950</td>\n", |
|  |  | " <td>[{\"name\": \"Legendary Pictures\", \"id\": 923}, {\"...</td>\n", |
|  |  | " <td>...</td>\n", |
|  |  | " <td>165.0</td>\n", |
|  |  | " <td>[{\"iso\_639\_1\": \"en\", \"name\": \"English\"}]</td>\n", |
|  |  | " <td>Released</td>\n", |
|  |  | " <td>The Legend Ends</td>\n", |
|  |  | " <td>The Dark Knight Rises</td>\n", |
|  |  | " <td>7.6</td>\n", |
|  |  | " <td>9106</td>\n", |
|  |  | " <td>The Dark Knight Rises</td>\n", |
|  |  | " <td>[{\"cast\_id\": 2, \"character\": \"Bruce Wayne / Ba...</td>\n", |
|  |  | " <td>[{\"credit\_id\": \"52fe4781c3a36847f81398c3\", \"de...</td>\n", |
|  |  | " </tr>\n", |
|  |  | " <tr>\n", |
|  |  | " <td>4</td>\n", |
|  |  | " <td>260000000</td>\n", |
|  |  | " <td>[{\"id\": 28, \"name\": \"Action\"}, {\"id\": 12, \"nam...</td>\n", |
|  |  | " <td>http://movies.disney.com/john-carter</td>\n", |
|  |  | " <td>49529</td>\n", |
|  |  | " <td>[{\"id\": 818, \"name\": \"based on novel\"}, {\"id\":...</td>\n", |
|  |  | " <td>en</td>\n", |
|  |  | " <td>John Carter</td>\n", |
|  |  | " <td>John Carter is a war-weary, former military ca...</td>\n", |
|  |  | " <td>43.926995</td>\n", |
|  |  | " <td>[{\"name\": \"Walt Disney Pictures\", \"id\": 2}]</td>\n", |
|  |  | " <td>...</td>\n", |
|  |  | " <td>132.0</td>\n", |
|  |  | " <td>[{\"iso\_639\_1\": \"en\", \"name\": \"English\"}]</td>\n", |
|  |  | " <td>Released</td>\n", |
|  |  | " <td>Lost in our world, found in another.</td>\n", |
|  |  | " <td>John Carter</td>\n", |
|  |  | " <td>6.1</td>\n", |
|  |  | " <td>2124</td>\n", |
|  |  | " <td>John Carter</td>\n", |
|  |  | " <td>[{\"cast\_id\": 5, \"character\": \"John Carter\", \"c...</td>\n", |
|  |  | " <td>[{\"credit\_id\": \"52fe479ac3a36847f813eaa3\", \"de...</td>\n", |
|  |  | " </tr>\n", |
|  |  | " </tbody>\n", |
|  |  | "</table>\n", |
|  |  | "<p>5 rows × 23 columns</p>\n", |
|  |  | "</div>" |
|  |  | ], |
|  |  | "text/plain": [ |
|  |  | " budget genres \\\n", |
|  |  | "0 237000000 [{\"id\": 28, \"name\": \"Action\"}, {\"id\": 12, \"nam... \n", |
|  |  | "1 300000000 [{\"id\": 12, \"name\": \"Adventure\"}, {\"id\": 14, \"... \n", |
|  |  | "2 245000000 [{\"id\": 28, \"name\": \"Action\"}, {\"id\": 12, \"nam... \n", |
|  |  | "3 250000000 [{\"id\": 28, \"name\": \"Action\"}, {\"id\": 80, \"nam... \n", |
|  |  | "4 260000000 [{\"id\": 28, \"name\": \"Action\"}, {\"id\": 12, \"nam... \n", |
|  |  | "\n", |
|  |  | " homepage id \\\n", |
|  |  | "0 http://www.avatarmovie.com/ 19995 \n", |
|  |  | "1 http://disney.go.com/disneypictures/pirates/ 285 \n", |
|  |  | "2 http://www.sonypictures.com/movies/spectre/ 206647 \n", |
|  |  | "3 http://www.thedarkknightrises.com/ 49026 \n", |
|  |  | "4 http://movies.disney.com/john-carter 49529 \n", |
|  |  | "\n", |
|  |  | " keywords original\_language \\\n", |
|  |  | "0 [{\"id\": 1463, \"name\": \"culture clash\"}, {\"id\":... en \n", |
|  |  | "1 [{\"id\": 270, \"name\": \"ocean\"}, {\"id\": 726, \"na... en \n", |
|  |  | "2 [{\"id\": 470, \"name\": \"spy\"}, {\"id\": 818, \"name... en \n", |
|  |  | "3 [{\"id\": 849, \"name\": \"dc comics\"}, {\"id\": 853,... en \n", |
|  |  | "4 [{\"id\": 818, \"name\": \"based on novel\"}, {\"id\":... en \n", |
|  |  | "\n", |
|  |  | " original\_title \\\n", |
|  |  | "0 Avatar \n", |
|  |  | "1 Pirates of the Caribbean: At World's End \n", |
|  |  | "2 Spectre \n", |
|  |  | "3 The Dark Knight Rises \n", |
|  |  | "4 John Carter \n", |
|  |  | "\n", |
|  |  | " overview popularity \\\n", |
|  |  | "0 In the 22nd century, a paraplegic Marine is di... 150.437577 \n", |
|  |  | "1 Captain Barbossa, long believed to be dead, ha... 139.082615 \n", |
|  |  | "2 A cryptic message from Bond’s past sends him o... 107.376788 \n", |
|  |  | "3 Following the death of District Attorney Harve... 112.312950 \n", |
|  |  | "4 John Carter is a war-weary, former military ca... 43.926995 \n", |
|  |  | "\n", |
|  |  | " production\_companies ... runtime \\\n", |
|  |  | "0 [{\"name\": \"Ingenious Film Partners\", \"id\": 289... ... 162.0 \n", |
|  |  | "1 [{\"name\": \"Walt Disney Pictures\", \"id\": 2}, {\"... ... 169.0 \n", |
|  |  | "2 [{\"name\": \"Columbia Pictures\", \"id\": 5}, {\"nam... ... 148.0 \n", |
|  |  | "3 [{\"name\": \"Legendary Pictures\", \"id\": 923}, {\"... ... 165.0 \n", |
|  |  | "4 [{\"name\": \"Walt Disney Pictures\", \"id\": 2}] ... 132.0 \n", |
|  |  | "\n", |
|  |  | " spoken\_languages status \\\n", |
|  |  | "0 [{\"iso\_639\_1\": \"en\", \"name\": \"English\"}, {\"iso... Released \n", |
|  |  | "1 [{\"iso\_639\_1\": \"en\", \"name\": \"English\"}] Released \n", |
|  |  | "2 [{\"iso\_639\_1\": \"fr\", \"name\": \"Fran\\u00e7ais\"},... Released \n", |
|  |  | "3 [{\"iso\_639\_1\": \"en\", \"name\": \"English\"}] Released \n", |
|  |  | "4 [{\"iso\_639\_1\": \"en\", \"name\": \"English\"}] Released \n", |
|  |  | "\n", |
|  |  | " tagline \\\n", |
|  |  | "0 Enter the World of Pandora. \n", |
|  |  | "1 At the end of the world, the adventure begins. \n", |
|  |  | "2 A Plan No One Escapes \n", |
|  |  | "3 The Legend Ends \n", |
|  |  | "4 Lost in our world, found in another. \n", |
|  |  | "\n", |
|  |  | " title vote\_average vote\_count \\\n", |
|  |  | "0 Avatar 7.2 11800 \n", |
|  |  | "1 Pirates of the Caribbean: At World's End 6.9 4500 \n", |
|  |  | "2 Spectre 6.3 4466 \n", |
|  |  | "3 The Dark Knight Rises 7.6 9106 \n", |
|  |  | "4 John Carter 6.1 2124 \n", |
|  |  | "\n", |
|  |  | " tittle \\\n", |
|  |  | "0 Avatar \n", |
|  |  | "1 Pirates of the Caribbean: At World's End \n", |
|  |  | "2 Spectre \n", |
|  |  | "3 The Dark Knight Rises \n", |
|  |  | "4 John Carter \n", |
|  |  | "\n", |
|  |  | " cast \\\n", |
|  |  | "0 [{\"cast\_id\": 242, \"character\": \"Jake Sully\", \"... \n", |
|  |  | "1 [{\"cast\_id\": 4, \"character\": \"Captain Jack Spa... \n", |
|  |  | "2 [{\"cast\_id\": 1, \"character\": \"James Bond\", \"cr... \n", |
|  |  | "3 [{\"cast\_id\": 2, \"character\": \"Bruce Wayne / Ba... \n", |
|  |  | "4 [{\"cast\_id\": 5, \"character\": \"John Carter\", \"c... \n", |
|  |  | "\n", |
|  |  | " crew \n", |
|  |  | "0 [{\"credit\_id\": \"52fe48009251416c750aca23\", \"de... \n", |
|  |  | "1 [{\"credit\_id\": \"52fe4232c3a36847f800b579\", \"de... \n", |
|  |  | "2 [{\"credit\_id\": \"54805967c3a36829b5002c41\", \"de... \n", |
|  |  | "3 [{\"credit\_id\": \"52fe4781c3a36847f81398c3\", \"de... \n", |
|  |  | "4 [{\"credit\_id\": \"52fe479ac3a36847f813eaa3\", \"de... \n", |
|  |  | "\n", |
|  |  | "[5 rows x 23 columns]" |
|  |  | ] |
|  |  | }, |
|  |  | "execution\_count": 43, |
|  |  | "metadata": {}, |
|  |  | "output\_type": "execute\_result" |
|  |  | } |
|  |  | ], |
|  |  | "source": [ |
|  |  | "df2.head(5)" |
|  |  | ] |
|  |  | }, |
|  |  | { |
|  |  | "cell\_type": "markdown", |
|  |  | "metadata": {}, |
|  |  | "source": [ |
|  |  | "# Demographic filtering" |
|  |  | ] |
|  |  | }, |
|  |  | { |
|  |  | "cell\_type": "code", |
|  |  | "execution\_count": 44, |
|  |  | "metadata": {}, |
|  |  | "outputs": [ |
|  |  | { |
|  |  | "data": { |
|  |  | "text/plain": [ |
|  |  | "6.092171559442011" |
|  |  | ] |
|  |  | }, |
|  |  | "execution\_count": 44, |
|  |  | "metadata": {}, |
|  |  | "output\_type": "execute\_result" |
|  |  | } |
|  |  | ], |
|  |  | "source": [ |
|  |  | "C= df2['vote\_average'].mean()\n", |
|  |  | "C" |
|  |  | ] |
|  |  | }, |
|  |  | { |
|  |  | "cell\_type": "code", |
|  |  | "execution\_count": 45, |
|  |  | "metadata": {}, |
|  |  | "outputs": [ |
|  |  | { |
|  |  | "data": { |
|  |  | "text/plain": [ |
|  |  | "1838.4000000000015" |
|  |  | ] |
|  |  | }, |
|  |  | "execution\_count": 45, |
|  |  | "metadata": {}, |
|  |  | "output\_type": "execute\_result" |
|  |  | } |
|  |  | ], |
|  |  | "source": [ |
|  |  | "m= df2['vote\_count'].quantile(0.9)\n", |
|  |  | "m" |
|  |  | ] |
|  |  | }, |
|  |  | { |
|  |  | "cell\_type": "code", |
|  |  | "execution\_count": 46, |
|  |  | "metadata": {}, |
|  |  | "outputs": [ |
|  |  | { |
|  |  | "data": { |
|  |  | "text/plain": [ |
|  |  | "(481, 23)" |
|  |  | ] |
|  |  | }, |
|  |  | "execution\_count": 46, |
|  |  | "metadata": {}, |
|  |  | "output\_type": "execute\_result" |
|  |  | } |
|  |  | ], |
|  |  | "source": [ |
|  |  | "# now we can filter out movies that qualify for the chart\n", |
|  |  | "q\_movies=df2.copy().loc[df2['vote\_count']>=m]\n", |
|  |  | "q\_movies.shape" |
|  |  | ] |
|  |  | }, |
|  |  | { |
|  |  | "cell\_type": "code", |
|  |  | "execution\_count": 47, |
|  |  | "metadata": {}, |
|  |  | "outputs": [], |
|  |  | "source": [ |
|  |  | "def weighted\_movie(x,m=m,C=C):\n", |
|  |  | " v=x['vote\_count']\n", |
|  |  | " R=x['vote\_average']\n", |
|  |  | " #calculation based on IMDB formula\n", |
|  |  | " \n", |
|  |  | " return (v/(v+m)\*R) + (m/(m+v)\*C)" |
|  |  | ] |
|  |  | }, |
|  |  | { |
|  |  | "cell\_type": "code", |
|  |  | "execution\_count": 48, |
|  |  | "metadata": {}, |
|  |  | "outputs": [], |
|  |  | "source": [ |
|  |  | "#define a new feature 'score' and calculate its value with weighted\_rating()\n", |
|  |  | "\n", |
|  |  | "q\_movies['score']= q\_movies.apply(weighted\_movie,axis=1)" |
|  |  | ] |
|  |  | }, |
|  |  | { |
|  |  | "cell\_type": "code", |
|  |  | "execution\_count": 49, |
|  |  | "metadata": {}, |
|  |  | "outputs": [ |
|  |  | { |
|  |  | "data": { |
|  |  | "text/html": [ |
|  |  | "<div>\n", |
|  |  | "<style scoped>\n", |
|  |  | " .dataframe tbody tr th:only-of-type {\n", |
|  |  | " vertical-align: middle;\n", |
|  |  | " }\n", |
|  |  | "\n", |
|  |  | " .dataframe tbody tr th {\n", |
|  |  | " vertical-align: top;\n", |
|  |  | " }\n", |
|  |  | "\n", |
|  |  | " .dataframe thead th {\n", |
|  |  | " text-align: right;\n", |
|  |  | " }\n", |
|  |  | "</style>\n", |
|  |  | "<table border=\"1\" class=\"dataframe\">\n", |
|  |  | " <thead>\n", |
|  |  | " <tr style=\"text-align: right;\">\n", |
|  |  | " <th></th>\n", |
|  |  | " <th>title</th>\n", |
|  |  | " <th>vote\_count</th>\n", |
|  |  | " <th>vote\_average</th>\n", |
|  |  | " <th>score</th>\n", |
|  |  | " </tr>\n", |
|  |  | " </thead>\n", |
|  |  | " <tbody>\n", |
|  |  | " <tr>\n", |
|  |  | " <td>1881</td>\n", |
|  |  | " <td>The Shawshank Redemption</td>\n", |
|  |  | " <td>8205</td>\n", |
|  |  | " <td>8.5</td>\n", |
|  |  | " <td>8.059258</td>\n", |
|  |  | " </tr>\n", |
|  |  | " <tr>\n", |
|  |  | " <td>662</td>\n", |
|  |  | " <td>Fight Club</td>\n", |
|  |  | " <td>9413</td>\n", |
|  |  | " <td>8.3</td>\n", |
|  |  | " <td>7.939256</td>\n", |
|  |  | " </tr>\n", |
|  |  | " <tr>\n", |
|  |  | " <td>65</td>\n", |
|  |  | " <td>The Dark Knight</td>\n", |
|  |  | " <td>12002</td>\n", |
|  |  | " <td>8.2</td>\n", |
|  |  | " <td>7.920020</td>\n", |
|  |  | " </tr>\n", |
|  |  | " <tr>\n", |
|  |  | " <td>3232</td>\n", |
|  |  | " <td>Pulp Fiction</td>\n", |
|  |  | " <td>8428</td>\n", |
|  |  | " <td>8.3</td>\n", |
|  |  | " <td>7.904645</td>\n", |
|  |  | " </tr>\n", |
|  |  | " <tr>\n", |
|  |  | " <td>96</td>\n", |
|  |  | " <td>Inception</td>\n", |
|  |  | " <td>13752</td>\n", |
|  |  | " <td>8.1</td>\n", |
|  |  | " <td>7.863239</td>\n", |
|  |  | " </tr>\n", |
|  |  | " <tr>\n", |
|  |  | " <td>3337</td>\n", |
|  |  | " <td>The Godfather</td>\n", |
|  |  | " <td>5893</td>\n", |
|  |  | " <td>8.4</td>\n", |
|  |  | " <td>7.851236</td>\n", |
|  |  | " </tr>\n", |
|  |  | " <tr>\n", |
|  |  | " <td>95</td>\n", |
|  |  | " <td>Interstellar</td>\n", |
|  |  | " <td>10867</td>\n", |
|  |  | " <td>8.1</td>\n", |
|  |  | " <td>7.809479</td>\n", |
|  |  | " </tr>\n", |
|  |  | " <tr>\n", |
|  |  | " <td>809</td>\n", |
|  |  | " <td>Forrest Gump</td>\n", |
|  |  | " <td>7927</td>\n", |
|  |  | " <td>8.2</td>\n", |
|  |  | " <td>7.803188</td>\n", |
|  |  | " </tr>\n", |
|  |  | " <tr>\n", |
|  |  | " <td>329</td>\n", |
|  |  | " <td>The Lord of the Rings: The Return of the King</td>\n", |
|  |  | " <td>8064</td>\n", |
|  |  | " <td>8.1</td>\n", |
|  |  | " <td>7.727243</td>\n", |
|  |  | " </tr>\n", |
|  |  | " <tr>\n", |
|  |  | " <td>1990</td>\n", |
|  |  | " <td>The Empire Strikes Back</td>\n", |
|  |  | " <td>5879</td>\n", |
|  |  | " <td>8.2</td>\n", |
|  |  | " <td>7.697884</td>\n", |
|  |  | " </tr>\n", |
|  |  | " </tbody>\n", |
|  |  | "</table>\n", |
|  |  | "</div>" |
|  |  | ], |
|  |  | "text/plain": [ |
|  |  | " title vote\_count vote\_average \\\n", |
|  |  | "1881 The Shawshank Redemption 8205 8.5 \n", |
|  |  | "662 Fight Club 9413 8.3 \n", |
|  |  | "65 The Dark Knight 12002 8.2 \n", |
|  |  | "3232 Pulp Fiction 8428 8.3 \n", |
|  |  | "96 Inception 13752 8.1 \n", |
|  |  | "3337 The Godfather 5893 8.4 \n", |
|  |  | "95 Interstellar 10867 8.1 \n", |
|  |  | "809 Forrest Gump 7927 8.2 \n", |
|  |  | "329 The Lord of the Rings: The Return of the King 8064 8.1 \n", |
|  |  | "1990 The Empire Strikes Back 5879 8.2 \n", |
|  |  | "\n", |
|  |  | " score \n", |
|  |  | "1881 8.059258 \n", |
|  |  | "662 7.939256 \n", |
|  |  | "65 7.920020 \n", |
|  |  | "3232 7.904645 \n", |
|  |  | "96 7.863239 \n", |
|  |  | "3337 7.851236 \n", |
|  |  | "95 7.809479 \n", |
|  |  | "809 7.803188 \n", |
|  |  | "329 7.727243 \n", |
|  |  | "1990 7.697884 " |
|  |  | ] |
|  |  | }, |
|  |  | "execution\_count": 49, |
|  |  | "metadata": {}, |
|  |  | "output\_type": "execute\_result" |
|  |  | } |
|  |  | ], |
|  |  | "source": [ |
|  |  | "q\_movies = q\_movies.sort\_values('score', ascending=False)\n", |
|  |  | "\n", |
|  |  | "#Print the top 15 movies\n", |
|  |  | "q\_movies[['title', 'vote\_count', 'vote\_average', 'score']].head(10)" |
|  |  | ] |
|  |  | }, |
|  |  | { |
|  |  | "cell\_type": "code", |
|  |  | "execution\_count": 51, |
|  |  | "metadata": {}, |
|  |  | "outputs": [], |
|  |  | "source": [ |
|  |  | "# We have made our first(though very basic) recommender.\n", |
|  |  | "#Under the Trending Now tab of these systems we find movies that are very popular \n", |
|  |  | "#and they can just be obtained by sorting the dataset by the popularity column." |
|  |  | ] |
|  |  | }, |
|  |  | { |
|  |  | "cell\_type": "code", |
|  |  | "execution\_count": 53, |
|  |  | "metadata": {}, |
|  |  | "outputs": [ |
|  |  | { |
|  |  | "data": { |
|  |  | "image/png": "\n", |
|  |  | "text/plain": [ |
|  |  | "<Figure size 864x288 with 1 Axes>" |
|  |  | ] |
|  |  | }, |
|  |  | "metadata": { |
|  |  | "needs\_background": "light" |
|  |  | }, |
|  |  | "output\_type": "display\_data" |
|  |  | } |
|  |  | ], |
|  |  | "source": [ |
|  |  | "pop= df2.sort\_values('popularity',ascending=False)\n", |
|  |  | "import matplotlib.pyplot as plt\n", |
|  |  | "plt.figure(figsize=(12,4))\n", |
|  |  | "\n", |
|  |  | "plt.barh(pop['title'].head(6),pop['popularity'].head(6), align='center',\n", |
|  |  | " color='skyblue')\n", |
|  |  | "plt.gca().invert\_yaxis()\n", |
|  |  | "plt.xlabel(\"Popularity\")\n", |
|  |  | "plt.title(\"Popular Movies\")\n", |
|  |  | "plt.show()" |
|  |  | ] |
|  |  | }, |
|  |  | { |
|  |  | "cell\_type": "markdown", |
|  |  | "metadata": {}, |
|  |  | "source": [ |
|  |  | "# Content Based Filtering" |
|  |  | ] |
|  |  | }, |
|  |  | { |
|  |  | "cell\_type": "code", |
|  |  | "execution\_count": 55, |
|  |  | "metadata": {}, |
|  |  | "outputs": [], |
|  |  | "source": [ |
|  |  | "# In this recommender system the content of the movie (overview, cast, crew, keyword, tagline etc) is used to find its similarity with other movies.\n", |
|  |  | "#Then the movies that are most likely to be similar are recommended.\n", |
|  |  | "\n" |
|  |  | ] |
|  |  | }, |
|  |  | { |
|  |  | "cell\_type": "code", |
|  |  | "execution\_count": 56, |
|  |  | "metadata": {}, |
|  |  | "outputs": [ |
|  |  | { |
|  |  | "data": { |
|  |  | "text/plain": [ |
|  |  | "0 In the 22nd century, a paraplegic Marine is di...\n", |
|  |  | "1 Captain Barbossa, long believed to be dead, ha...\n", |
|  |  | "2 A cryptic message from Bond’s past sends him o...\n", |
|  |  | "3 Following the death of District Attorney Harve...\n", |
|  |  | "4 John Carter is a war-weary, former military ca...\n", |
|  |  | "Name: overview, dtype: object" |
|  |  | ] |
|  |  | }, |
|  |  | "execution\_count": 56, |
|  |  | "metadata": {}, |
|  |  | "output\_type": "execute\_result" |
|  |  | } |
|  |  | ], |
|  |  | "source": [ |
|  |  | "df2['overview'].head(5)" |
|  |  | ] |
|  |  | }, |
|  |  | { |
|  |  | "cell\_type": "code", |
|  |  | "execution\_count": 57, |
|  |  | "metadata": {}, |
|  |  | "outputs": [ |
|  |  | { |
|  |  | "data": { |
|  |  | "text/plain": [ |
|  |  | "(4803, 20978)" |
|  |  | ] |
|  |  | }, |
|  |  | "execution\_count": 57, |
|  |  | "metadata": {}, |
|  |  | "output\_type": "execute\_result" |
|  |  | } |
|  |  | ], |
|  |  | "source": [ |
|  |  | "from sklearn.feature\_extraction.text import TfidfVectorizer\n", |
|  |  | "\n", |
|  |  | "tfidf= TfidfVectorizer(stop\_words='english')\n", |
|  |  | "\n", |
|  |  | "# Replace Nan with an empty string\n", |
|  |  | "df2['overview']=df2['overview'].fillna('')\n", |
|  |  | "\n", |
|  |  | "# construct the required TF-IDF matrix by fitting and transforming the data\n", |
|  |  | "tfidf\_matrix= tfidf.fit\_transform(df2['overview'])\n", |
|  |  | "\n", |
|  |  | "tfidf\_matrix.shape" |
|  |  | ] |
|  |  | }, |
|  |  | { |
|  |  | "cell\_type": "code", |
|  |  | "execution\_count": 58, |
|  |  | "metadata": {}, |
|  |  | "outputs": [], |
|  |  | "source": [ |
|  |  | "from sklearn.metrics.pairwise import linear\_kernel\n", |
|  |  | "\n", |
|  |  | "#compute the cosine similarity matrix\n", |
|  |  | "cosine\_sim= linear\_kernel(tfidf\_matrix,tfidf\_matrix)" |
|  |  | ] |
|  |  | }, |
|  |  | { |
|  |  | "cell\_type": "markdown", |
|  |  | "metadata": {}, |
|  |  | "source": [ |
|  |  | "# We are going to define a function that takes in a movie title as an input and outputs a list of the 10 most similar movies. Firstly, for this, we need a reverse mapping of movie titles and DataFrame indices. In other words, we need a mechanism to identify the index of a movie in our metadata DataFrame, given its title.\n", |
|  |  | "\n" |
|  |  | ] |
|  |  | }, |
|  |  | { |
|  |  | "cell\_type": "code", |
|  |  | "execution\_count": 59, |
|  |  | "metadata": {}, |
|  |  | "outputs": [], |
|  |  | "source": [ |
|  |  | "# construct a reverse map of indices and movie titles\n", |
|  |  | "indices= pd.Series(df2.index, index=df2['title']).drop\_duplicates()" |
|  |  | ] |
|  |  | }, |
|  |  | { |
|  |  | "cell\_type": "code", |
|  |  | "execution\_count": 68, |
|  |  | "metadata": {}, |
|  |  | "outputs": [], |
|  |  | "source": [ |
|  |  | "# Function that takes in movie title & outputs similar movies\n", |
|  |  | "def get\_recommendation(title,cosine\_sim=cosine\_sim):\n", |
|  |  | " # get the index of the movie that matches the title\n", |
|  |  | " idx= indices[title]\n", |
|  |  | " \n", |
|  |  | " # get the pairwise similarity scores of all the movies with that movie\n", |
|  |  | " sim\_scores=list(enumerate(cosine\_sim[idx]))\n", |
|  |  | " \n", |
|  |  | " #sort the movies based on similarity scores\n", |
|  |  | " sim\_scores= sorted(sim\_scores,key=lambda x :x[1], reverse=True)\n", |
|  |  | " \n", |
|  |  | " # get the scors of the 10 most similar movies\n", |
|  |  | " sim\_scores=sim\_scores[1:11]\n", |
|  |  | " \n", |
|  |  | " #get the movie indices\n", |
|  |  | " movie\_indices=[i[0] for i in sim\_scores]\n", |
|  |  | " \n", |
|  |  | " # return the top 10 most similar movies\n", |
|  |  | " return df2['title'].iloc[movie\_indices]" |
|  |  | ] |
|  |  | }, |
|  |  | { |
|  |  | "cell\_type": "code", |
|  |  | "execution\_count": 69, |
|  |  | "metadata": {}, |
|  |  | "outputs": [ |
|  |  | { |
|  |  | "data": { |
|  |  | "text/plain": [ |
|  |  | "65 The Dark Knight\n", |
|  |  | "299 Batman Forever\n", |
|  |  | "428 Batman Returns\n", |
|  |  | "1359 Batman\n", |
|  |  | "3854 Batman: The Dark Knight Returns, Part 2\n", |
|  |  | "119 Batman Begins\n", |
|  |  | "2507 Slow Burn\n", |
|  |  | "9 Batman v Superman: Dawn of Justice\n", |
|  |  | "1181 JFK\n", |
|  |  | "210 Batman & Robin\n", |
|  |  | "Name: title, dtype: object" |
|  |  | ] |
|  |  | }, |
|  |  | "execution\_count": 69, |
|  |  | "metadata": {}, |
|  |  | "output\_type": "execute\_result" |
|  |  | } |
|  |  | ], |
|  |  | "source": [ |
|  |  | "get\_recommendation('The Dark Knight Rises')" |
|  |  | ] |
|  |  | }, |
|  |  | { |
|  |  | "cell\_type": "code", |
|  |  | "execution\_count": 70, |
|  |  | "metadata": {}, |
|  |  | "outputs": [ |
|  |  | { |
|  |  | "data": { |
|  |  | "text/plain": [ |
|  |  | "7 Avengers: Age of Ultron\n", |
|  |  | "3144 Plastic\n", |
|  |  | "1715 Timecop\n", |
|  |  | "4124 This Thing of Ours\n", |
|  |  | "3311 Thank You for Smoking\n", |
|  |  | "3033 The Corruptor\n", |
|  |  | "588 Wall Street: Money Never Sleeps\n", |
|  |  | "2136 Team America: World Police\n", |
|  |  | "1468 The Fountain\n", |
|  |  | "1286 Snowpiercer\n", |
|  |  | "Name: title, dtype: object" |
|  |  | ] |
|  |  | }, |
|  |  | "execution\_count": 70, |
|  |  | "metadata": {}, |
|  |  | "output\_type": "execute\_result" |
|  |  | } |
|  |  | ], |
|  |  | "source": [ |
|  |  | "get\_recommendation('The Avengers')" |
|  |  | ] |
|  |  | }, |
|  |  | { |
|  |  | "cell\_type": "markdown", |
|  |  | "metadata": {}, |
|  |  | "source": [ |
|  |  | "# Credits, genere and keywords Based Recommender" |
|  |  | ] |
|  |  | }, |
|  |  | { |
|  |  | "cell\_type": "markdown", |
|  |  | "metadata": {}, |
|  |  | "source": [ |
|  |  | "It goes without saying that the quality of our recommender would be increased with the usage of better metadata. That is exactly what we are going to do in this section. We are going to build a recommender based on the following metadata: the 3 top actors, the director, related genres and the movie plot keywords.\n", |
|  |  | "\n", |
|  |  | "From the cast, crew and keywords features, we need to extract the three most important actors, the director and the keywords associated with that movie. Right now, our data is present in the form of \"stringified\" lists , we need to convert it into a safe and usable structure\n", |
|  |  | "\n" |
|  |  | ] |
|  |  | }, |
|  |  | { |
|  |  | "cell\_type": "code", |
|  |  | "execution\_count": 72, |
|  |  | "metadata": {}, |
|  |  | "outputs": [], |
|  |  | "source": [ |
|  |  | "# parse the stringified features into their corresponding python objects\n", |
|  |  | "from ast import literal\_eval\n", |
|  |  | "features=['cast','crew','keywords','genres']\n", |
|  |  | "for feature in features:\n", |
|  |  | " df2[feature]= df2[feature].apply(literal\_eval)" |
|  |  | ] |
|  |  | }, |
|  |  | { |
|  |  | "cell\_type": "code", |
|  |  | "execution\_count": 73, |
|  |  | "metadata": {}, |
|  |  | "outputs": [], |
|  |  | "source": [ |
|  |  | "# get the director's name from crew feature. If director is not listed, return NAn\n", |
|  |  | "\n", |
|  |  | "def get\_director(x):\n", |
|  |  | " for i in x:\n", |
|  |  | " if i['job']=='Director':\n", |
|  |  | " return i['name']\n", |
|  |  | " return np.nan" |
|  |  | ] |
|  |  | }, |
|  |  | { |
|  |  | "cell\_type": "code", |
|  |  | "execution\_count": 74, |
|  |  | "metadata": {}, |
|  |  | "outputs": [], |
|  |  | "source": [ |
|  |  | "# return the list top 3 elements or entire list ; whichever is more.\n", |
|  |  | "def get\_list(x):\n", |
|  |  | " if isinstance(x,list):\n", |
|  |  | " names=[i['name'] for i in x]\n", |
|  |  | " # check if more than 3 elements exists, If yes, return only first three. If no , return entire list\n", |
|  |  | " if len(names)>3:\n", |
|  |  | " names=names[:3]\n", |
|  |  | " return names\n", |
|  |  | " #return empty list in case of missing/malformed data\n", |
|  |  | " return []" |
|  |  | ] |
|  |  | }, |
|  |  | { |
|  |  | "cell\_type": "code", |
|  |  | "execution\_count": 76, |
|  |  | "metadata": {}, |
|  |  | "outputs": [], |
|  |  | "source": [ |
|  |  | "# define new director , cast , genere and keywords features that are in suitable form\n", |
|  |  | "\n", |
|  |  | "df2['director']=df2['crew'].apply(get\_director)\n", |
|  |  | "features=['cast','keywords','genres']\n", |
|  |  | "for feature in features:\n", |
|  |  | " df2[feature]=df2[feature].apply(get\_list)" |
|  |  | ] |
|  |  | }, |
|  |  | { |
|  |  | "cell\_type": "code", |
|  |  | "execution\_count": 77, |
|  |  | "metadata": {}, |
|  |  | "outputs": [ |
|  |  | { |
|  |  | "data": { |
|  |  | "text/html": [ |
|  |  | "<div>\n", |
|  |  | "<style scoped>\n", |
|  |  | " .dataframe tbody tr th:only-of-type {\n", |
|  |  | " vertical-align: middle;\n", |
|  |  | " }\n", |
|  |  | "\n", |
|  |  | " .dataframe tbody tr th {\n", |
|  |  | " vertical-align: top;\n", |
|  |  | " }\n", |
|  |  | "\n", |
|  |  | " .dataframe thead th {\n", |
|  |  | " text-align: right;\n", |
|  |  | " }\n", |
|  |  | "</style>\n", |
|  |  | "<table border=\"1\" class=\"dataframe\">\n", |
|  |  | " <thead>\n", |
|  |  | " <tr style=\"text-align: right;\">\n", |
|  |  | " <th></th>\n", |
|  |  | " <th>title</th>\n", |
|  |  | " <th>cast</th>\n", |
|  |  | " <th>director</th>\n", |
|  |  | " <th>keywords</th>\n", |
|  |  | " <th>genres</th>\n", |
|  |  | " </tr>\n", |
|  |  | " </thead>\n", |
|  |  | " <tbody>\n", |
|  |  | " <tr>\n", |
|  |  | " <td>0</td>\n", |
|  |  | " <td>Avatar</td>\n", |
|  |  | " <td>[Sam Worthington, Zoe Saldana, Sigourney Weaver]</td>\n", |
|  |  | " <td>NaN</td>\n", |
|  |  | " <td>[culture clash, future, space war]</td>\n", |
|  |  | " <td>[Action, Adventure, Fantasy]</td>\n", |
|  |  | " </tr>\n", |
|  |  | " <tr>\n", |
|  |  | " <td>1</td>\n", |
|  |  | " <td>Pirates of the Caribbean: At World's End</td>\n", |
|  |  | " <td>[Johnny Depp, Orlando Bloom, Keira Knightley]</td>\n", |
|  |  | " <td>NaN</td>\n", |
|  |  | " <td>[ocean, drug abuse, exotic island]</td>\n", |
|  |  | " <td>[]</td>\n", |
|  |  | " </tr>\n", |
|  |  | " <tr>\n", |
|  |  | " <td>2</td>\n", |
|  |  | " <td>Spectre</td>\n", |
|  |  | " <td>[Daniel Craig, Christoph Waltz, Léa Seydoux]</td>\n", |
|  |  | " <td>NaN</td>\n", |
|  |  | " <td>[spy, based on novel, secret agent]</td>\n", |
|  |  | " <td>[]</td>\n", |
|  |  | " </tr>\n", |
|  |  | " </tbody>\n", |
|  |  | "</table>\n", |
|  |  | "</div>" |
|  |  | ], |
|  |  | "text/plain": [ |
|  |  | " title \\\n", |
|  |  | "0 Avatar \n", |
|  |  | "1 Pirates of the Caribbean: At World's End \n", |
|  |  | "2 Spectre \n", |
|  |  | "\n", |
|  |  | " cast director \\\n", |
|  |  | "0 [Sam Worthington, Zoe Saldana, Sigourney Weaver] NaN \n", |
|  |  | "1 [Johnny Depp, Orlando Bloom, Keira Knightley] NaN \n", |
|  |  | "2 [Daniel Craig, Christoph Waltz, Léa Seydoux] NaN \n", |
|  |  | "\n", |
|  |  | " keywords genres \n", |
|  |  | "0 [culture clash, future, space war] [Action, Adventure, Fantasy] \n", |
|  |  | "1 [ocean, drug abuse, exotic island] [] \n", |
|  |  | "2 [spy, based on novel, secret agent] [] " |
|  |  | ] |
|  |  | }, |
|  |  | "execution\_count": 77, |
|  |  | "metadata": {}, |
|  |  | "output\_type": "execute\_result" |
|  |  | } |
|  |  | ], |
|  |  | "source": [ |
|  |  | "# print new features of the first three films\n", |
|  |  | "\n", |
|  |  | "df2[['title','cast','director','keywords','genres']].head(3)" |
|  |  | ] |
|  |  | }, |
|  |  | { |
|  |  | "cell\_type": "markdown", |
|  |  | "metadata": {}, |
|  |  | "source": [ |
|  |  | "The next step would be to convert the names and keyword instances into lowercase and strip all the spaces between them. This is done so that our vectorizer doesn't count the Johnny of \"Johnny Depp\" and \"Johnny Galecki\" as the same." |
|  |  | ] |
|  |  | }, |
|  |  | { |
|  |  | "cell\_type": "code", |
|  |  | "execution\_count": 78, |
|  |  | "metadata": {}, |
|  |  | "outputs": [], |
|  |  | "source": [ |
|  |  | "# function to conver all strings to lower case and strip names of spaces\n", |
|  |  | "\n", |
|  |  | "def clean\_data(x):\n", |
|  |  | " if isinstance(x,list):\n", |
|  |  | " return[str.lower(i.replace(\" \",\"\")) for i in x]\n", |
|  |  | " else:\n", |
|  |  | " #check if director exists. If not, return empty string\n", |
|  |  | " if isinstance(x,str):\n", |
|  |  | " return str.lower(x.replace(\" \",\"\"))\n", |
|  |  | " else:\n", |
|  |  | " return ''" |
|  |  | ] |
|  |  | }, |
|  |  | { |
|  |  | "cell\_type": "code", |
|  |  | "execution\_count": 79, |
|  |  | "metadata": {}, |
|  |  | "outputs": [], |
|  |  | "source": [ |
|  |  | "# apply clean\_data function to your features.\n", |
|  |  | "features=['cast','keywords','director','genres']\n", |
|  |  | "\n", |
|  |  | "for feature in features:\n", |
|  |  | " df2[feature]=df2[feature].apply(clean\_data)" |
|  |  | ] |
|  |  | }, |
|  |  | { |
|  |  | "cell\_type": "markdown", |
|  |  | "metadata": {}, |
|  |  | "source": [ |
|  |  | "We are now in a position to create our \"metadata soup\", which is a string that contains all the metadata that we want to feed to our vectorizer (namely actors, director and keywords)." |
|  |  | ] |
|  |  | }, |
|  |  | { |
|  |  | "cell\_type": "code", |
|  |  | "execution\_count": 80, |
|  |  | "metadata": {}, |
|  |  | "outputs": [], |
|  |  | "source": [ |
|  |  | "def create\_soup(x):\n", |
|  |  | " return ' '.join(x['keywords']) + ' ' + ' '.join(x['cast']) + ' ' + x['director'] + ' ' + ' '.join(x['genres'])\n", |
|  |  | "df2['soup'] = df2.apply(create\_soup, axis=1)\n" |
|  |  | ] |
|  |  | }, |
|  |  | { |
|  |  | "cell\_type": "markdown", |
|  |  | "metadata": {}, |
|  |  | "source": [ |
|  |  | "The next steps are the same as what we did with our plot description based recommender. One important difference is that we use the CountVectorizer() instead of TF-IDF. This is because we do not want to down-weight the presence of an actor/director if he or she has acted or directed in relatively more movies. It doesn't make much intuitive sense." |
|  |  | ] |
|  |  | }, |
|  |  | { |
|  |  | "cell\_type": "code", |
|  |  | "execution\_count": 85, |
|  |  | "metadata": {}, |
|  |  | "outputs": [], |
|  |  | "source": [ |
|  |  | "from sklearn.feature\_extraction.text import CountVectorizer\n", |
|  |  | "\n", |
|  |  | "count= CountVectorizer(stop\_words='english')\n", |
|  |  | "count\_matrix= count.fit\_transform(df2['soup'])" |
|  |  | ] |
|  |  | }, |
|  |  | { |
|  |  | "cell\_type": "code", |
|  |  | "execution\_count": 86, |
|  |  | "metadata": {}, |
|  |  | "outputs": [], |
|  |  | "source": [ |
|  |  | "# compute the cosine similarity matrix based on the count\_matrix\n", |
|  |  | "from sklearn.metrics.pairwise import cosine\_similarity\n", |
|  |  | "cosine\_sim2=cosine\_similarity(count\_matrix,count\_matrix)" |
|  |  | ] |
|  |  | }, |
|  |  | { |
|  |  | "cell\_type": "code", |
|  |  | "execution\_count": 87, |
|  |  | "metadata": {}, |
|  |  | "outputs": [], |
|  |  | "source": [ |
|  |  | "# reset index of our main datafrsme and construct reverse mapping as before\n", |
|  |  | "df2= df2.reset\_index()\n", |
|  |  | "indices= pd.Series(df2.index, index=df2['title'])" |
|  |  | ] |
|  |  | }, |
|  |  | { |
|  |  | "cell\_type": "markdown", |
|  |  | "metadata": {}, |
|  |  | "source": [ |
|  |  | "We can now reuse our get\_recommendations() function by passing in the new cosine\_sim2 matrix as your second argument." |
|  |  | ] |
|  |  | }, |
|  |  | { |
|  |  | "cell\_type": "code", |
|  |  | "execution\_count": 88, |
|  |  | "metadata": {}, |
|  |  | "outputs": [ |
|  |  | { |
|  |  | "data": { |
|  |  | "text/plain": [ |
|  |  | "65 The Dark Knight\n", |
|  |  | "3073 Romeo Is Bleeding\n", |
|  |  | "119 Batman Begins\n", |
|  |  | "747 Gangster Squad\n", |
|  |  | "1470 Stolen\n", |
|  |  | "1503 Takers\n", |
|  |  | "1986 Faster\n", |
|  |  | "2154 Street Kings\n", |
|  |  | "2277 The Baader Meinhof Complex\n", |
|  |  | "3332 Harry Brown\n", |
|  |  | "Name: title, dtype: object" |
|  |  | ] |
|  |  | }, |
|  |  | "execution\_count": 88, |
|  |  | "metadata": {}, |
|  |  | "output\_type": "execute\_result" |
|  |  | } |
|  |  | ], |
|  |  | "source": [ |
|  |  | "get\_recommendation('The Dark Knight Rises',cosine\_sim2)" |
|  |  | ] |
|  |  | }, |
|  |  | { |
|  |  | "cell\_type": "code", |
|  |  | "execution\_count": 89, |
|  |  | "metadata": {}, |
|  |  | "outputs": [ |
|  |  | { |
|  |  | "data": { |
|  |  | "text/plain": [ |
|  |  | "4124 This Thing of Ours\n", |
|  |  | "867 The Godfather: Part III\n", |
|  |  | "347 Cloudy with a Chance of Meatballs\n", |
|  |  | "496 Cloudy with a Chance of Meatballs 2\n", |
|  |  | "499 Jack and Jill\n", |
|  |  | "2401 City of Ghosts\n", |
|  |  | "2649 The Son of No One\n", |
|  |  | "4147 Small Apartments\n", |
|  |  | "3069 Danny Collins\n", |
|  |  | "20 The Amazing Spider-Man\n", |
|  |  | "Name: title, dtype: object" |
|  |  | ] |
|  |  | }, |
|  |  | "execution\_count": 89, |
|  |  | "metadata": {}, |
|  |  | "output\_type": "execute\_result" |
|  |  | } |
|  |  | ], |
|  |  | "source": [ |
|  |  | "get\_recommendation('The Godfather',cosine\_sim2)" |
|  |  | ] |
|  |  | }, |
|  |  | { |
|  |  | "cell\_type": "code", |
|  |  | "execution\_count": 98, |
|  |  | "metadata": {}, |
|  |  | "outputs": [], |
|  |  | "source": [ |
|  |  | "from surprise import Reader , Dataset , SVD \n", |
|  |  | "from surprise.model\_selection import cross\_validate" |
|  |  | ] |
|  |  | }, |
|  |  | { |
|  |  | "cell\_type": "code", |
|  |  | "execution\_count": 94, |
|  |  | "metadata": {}, |
|  |  | "outputs": [ |
|  |  | { |
|  |  | "data": { |
|  |  | "text/html": [ |
|  |  | "<div>\n", |
|  |  | "<style scoped>\n", |
|  |  | " .dataframe tbody tr th:only-of-type {\n", |
|  |  | " vertical-align: middle;\n", |
|  |  | " }\n", |
|  |  | "\n", |
|  |  | " .dataframe tbody tr th {\n", |
|  |  | " vertical-align: top;\n", |
|  |  | " }\n", |
|  |  | "\n", |
|  |  | " .dataframe thead th {\n", |
|  |  | " text-align: right;\n", |
|  |  | " }\n", |
|  |  | "</style>\n", |
|  |  | "<table border=\"1\" class=\"dataframe\">\n", |
|  |  | " <thead>\n", |
|  |  | " <tr style=\"text-align: right;\">\n", |
|  |  | " <th></th>\n", |
|  |  | " <th>userId</th>\n", |
|  |  | " <th>movieId</th>\n", |
|  |  | " <th>rating</th>\n", |
|  |  | " <th>timestamp</th>\n", |
|  |  | " </tr>\n", |
|  |  | " </thead>\n", |
|  |  | " <tbody>\n", |
|  |  | " <tr>\n", |
|  |  | " <td>0</td>\n", |
|  |  | " <td>1</td>\n", |
|  |  | " <td>31</td>\n", |
|  |  | " <td>2.5</td>\n", |
|  |  | " <td>1260759144</td>\n", |
|  |  | " </tr>\n", |
|  |  | " <tr>\n", |
|  |  | " <td>1</td>\n", |
|  |  | " <td>1</td>\n", |
|  |  | " <td>1029</td>\n", |
|  |  | " <td>3.0</td>\n", |
|  |  | " <td>1260759179</td>\n", |
|  |  | " </tr>\n", |
|  |  | " <tr>\n", |
|  |  | " <td>2</td>\n", |
|  |  | " <td>1</td>\n", |
|  |  | " <td>1061</td>\n", |
|  |  | " <td>3.0</td>\n", |
|  |  | " <td>1260759182</td>\n", |
|  |  | " </tr>\n", |
|  |  | " <tr>\n", |
|  |  | " <td>3</td>\n", |
|  |  | " <td>1</td>\n", |
|  |  | " <td>1129</td>\n", |
|  |  | " <td>2.0</td>\n", |
|  |  | " <td>1260759185</td>\n", |
|  |  | " </tr>\n", |
|  |  | " <tr>\n", |
|  |  | " <td>4</td>\n", |
|  |  | " <td>1</td>\n", |
|  |  | " <td>1172</td>\n", |
|  |  | " <td>4.0</td>\n", |
|  |  | " <td>1260759205</td>\n", |
|  |  | " </tr>\n", |
|  |  | " </tbody>\n", |
|  |  | "</table>\n", |
|  |  | "</div>" |
|  |  | ], |
|  |  | "text/plain": [ |
|  |  | " userId movieId rating timestamp\n", |
|  |  | "0 1 31 2.5 1260759144\n", |
|  |  | "1 1 1029 3.0 1260759179\n", |
|  |  | "2 1 1061 3.0 1260759182\n", |
|  |  | "3 1 1129 2.0 1260759185\n", |
|  |  | "4 1 1172 4.0 1260759205" |
|  |  | ] |
|  |  | }, |
|  |  | "execution\_count": 94, |
|  |  | "metadata": {}, |
|  |  | "output\_type": "execute\_result" |
|  |  | } |
|  |  | ], |
|  |  | "source": [ |
|  |  | "reader= Reader()\n", |
|  |  | "ratings= pd.read\_csv('ratings.csv')\n", |
|  |  | "ratings.head()" |
|  |  | ] |
|  |  | }, |
|  |  | { |
|  |  | "cell\_type": "code", |
|  |  | "execution\_count": 99, |
|  |  | "metadata": {}, |
|  |  | "outputs": [ |
|  |  | { |
|  |  | "ename": "AttributeError", |
|  |  | "evalue": "'DatasetAutoFolds' object has no attribute 'split'", |
|  |  | "output\_type": "error", |
|  |  | "traceback": [ |
|  |  | "\u001b[1;31m---------------------------------------------------------------------------\u001b[0m", |
|  |  | "\u001b[1;31mAttributeError\u001b[0m Traceback (most recent call last)", |
|  |  | "\u001b[1;32m<ipython-input-99-7ac32b723610>\u001b[0m in \u001b[0;36m<module>\u001b[1;34m\u001b[0m\n\u001b[0;32m 1\u001b[0m \u001b[0mdata\u001b[0m \u001b[1;33m=\u001b[0m \u001b[0mDataset\u001b[0m\u001b[1;33m.\u001b[0m\u001b[0mload\_from\_df\u001b[0m\u001b[1;33m(\u001b[0m\u001b[0mratings\u001b[0m\u001b[1;33m[\u001b[0m\u001b[1;33m[\u001b[0m\u001b[1;34m'userId'\u001b[0m\u001b[1;33m,\u001b[0m \u001b[1;34m'movieId'\u001b[0m\u001b[1;33m,\u001b[0m \u001b[1;34m'rating'\u001b[0m\u001b[1;33m]\u001b[0m\u001b[1;33m]\u001b[0m\u001b[1;33m,\u001b[0m \u001b[0mreader\u001b[0m\u001b[1;33m)\u001b[0m\u001b[1;33m\u001b[0m\u001b[1;33m\u001b[0m\u001b[0m\n\u001b[1;32m----> 2\u001b[1;33m \u001b[0mdata\u001b[0m\u001b[1;33m.\u001b[0m\u001b[0msplit\u001b[0m\u001b[1;33m(\u001b[0m\u001b[0mn\_folds\u001b[0m\u001b[1;33m=\u001b[0m\u001b[1;36m5\u001b[0m\u001b[1;33m)\u001b[0m\u001b[1;33m\u001b[0m\u001b[1;33m\u001b[0m\u001b[0m\n\u001b[0m", |
|  |  | "\u001b[1;31mAttributeError\u001b[0m: 'DatasetAutoFolds' object has no attribute 'split'" |
|  |  | ] |
|  |  | } |
|  |  | ], |
|  |  | "source": [ |
|  |  | "data = Dataset.load\_from\_df(ratings[['userId', 'movieId', 'rating']], reader)\n", |
|  |  | "data.split(n\_folds=5)" |
|  |  | ] |
|  |  | }, |
|  |  | { |
|  |  | "cell\_type": "code", |
|  |  | "execution\_count": 101, |
|  |  | "metadata": {}, |
|  |  | "outputs": [ |
|  |  | { |
|  |  | "data": { |
|  |  | "text/plain": [ |
|  |  | "<surprise.prediction\_algorithms.matrix\_factorization.SVD at 0x1f5619aee88>" |
|  |  | ] |
|  |  | }, |
|  |  | "execution\_count": 101, |
|  |  | "metadata": {}, |
|  |  | "output\_type": "execute\_result" |
|  |  | } |
|  |  | ], |
|  |  | "source": [ |
|  |  | "svd=SVD()\n", |
|  |  | "trainset = data.build\_full\_trainset()\n", |
|  |  | "svd.fit(trainset)" |
|  |  | ] |
|  |  | }, |
|  |  | { |
|  |  | "cell\_type": "code", |
|  |  | "execution\_count": 102, |
|  |  | "metadata": {}, |
|  |  | "outputs": [ |
|  |  | { |
|  |  | "data": { |
|  |  | "text/html": [ |
|  |  | "<div>\n", |
|  |  | "<style scoped>\n", |
|  |  | " .dataframe tbody tr th:only-of-type {\n", |
|  |  | " vertical-align: middle;\n", |
|  |  | " }\n", |
|  |  | "\n", |
|  |  | " .dataframe tbody tr th {\n", |
|  |  | " vertical-align: top;\n", |
|  |  | " }\n", |
|  |  | "\n", |
|  |  | " .dataframe thead th {\n", |
|  |  | " text-align: right;\n", |
|  |  | " }\n", |
|  |  | "</style>\n", |
|  |  | "<table border=\"1\" class=\"dataframe\">\n", |
|  |  | " <thead>\n", |
|  |  | " <tr style=\"text-align: right;\">\n", |
|  |  | " <th></th>\n", |
|  |  | " <th>userId</th>\n", |
|  |  | " <th>movieId</th>\n", |
|  |  | " <th>rating</th>\n", |
|  |  | " <th>timestamp</th>\n", |
|  |  | " </tr>\n", |
|  |  | " </thead>\n", |
|  |  | " <tbody>\n", |
|  |  | " <tr>\n", |
|  |  | " <td>0</td>\n", |
|  |  | " <td>1</td>\n", |
|  |  | " <td>31</td>\n", |
|  |  | " <td>2.5</td>\n", |
|  |  | " <td>1260759144</td>\n", |
|  |  | " </tr>\n", |
|  |  | " <tr>\n", |
|  |  | " <td>1</td>\n", |
|  |  | " <td>1</td>\n", |
|  |  | " <td>1029</td>\n", |
|  |  | " <td>3.0</td>\n", |
|  |  | " <td>1260759179</td>\n", |
|  |  | " </tr>\n", |
|  |  | " <tr>\n", |
|  |  | " <td>2</td>\n", |
|  |  | " <td>1</td>\n", |
|  |  | " <td>1061</td>\n", |
|  |  | " <td>3.0</td>\n", |
|  |  | " <td>1260759182</td>\n", |
|  |  | " </tr>\n", |
|  |  | " <tr>\n", |
|  |  | " <td>3</td>\n", |
|  |  | " <td>1</td>\n", |
|  |  | " <td>1129</td>\n", |
|  |  | " <td>2.0</td>\n", |
|  |  | " <td>1260759185</td>\n", |
|  |  | " </tr>\n", |
|  |  | " <tr>\n", |
|  |  | " <td>4</td>\n", |
|  |  | " <td>1</td>\n", |
|  |  | " <td>1172</td>\n", |
|  |  | " <td>4.0</td>\n", |
|  |  | " <td>1260759205</td>\n", |
|  |  | " </tr>\n", |
|  |  | " <tr>\n", |
|  |  | " <td>5</td>\n", |
|  |  | " <td>1</td>\n", |
|  |  | " <td>1263</td>\n", |
|  |  | " <td>2.0</td>\n", |
|  |  | " <td>1260759151</td>\n", |
|  |  | " </tr>\n", |
|  |  | " <tr>\n", |
|  |  | " <td>6</td>\n", |
|  |  | " <td>1</td>\n", |
|  |  | " <td>1287</td>\n", |
|  |  | " <td>2.0</td>\n", |
|  |  | " <td>1260759187</td>\n", |
|  |  | " </tr>\n", |
|  |  | " <tr>\n", |
|  |  | " <td>7</td>\n", |
|  |  | " <td>1</td>\n", |
|  |  | " <td>1293</td>\n", |
|  |  | " <td>2.0</td>\n", |
|  |  | " <td>1260759148</td>\n", |
|  |  | " </tr>\n", |
|  |  | " <tr>\n", |
|  |  | " <td>8</td>\n", |
|  |  | " <td>1</td>\n", |
|  |  | " <td>1339</td>\n", |
|  |  | " <td>3.5</td>\n", |
|  |  | " <td>1260759125</td>\n", |
|  |  | " </tr>\n", |
|  |  | " <tr>\n", |
|  |  | " <td>9</td>\n", |
|  |  | " <td>1</td>\n", |
|  |  | " <td>1343</td>\n", |
|  |  | " <td>2.0</td>\n", |
|  |  | " <td>1260759131</td>\n", |
|  |  | " </tr>\n", |
|  |  | " <tr>\n", |
|  |  | " <td>10</td>\n", |
|  |  | " <td>1</td>\n", |
|  |  | " <td>1371</td>\n", |
|  |  | " <td>2.5</td>\n", |
|  |  | " <td>1260759135</td>\n", |
|  |  | " </tr>\n", |
|  |  | " <tr>\n", |
|  |  | " <td>11</td>\n", |
|  |  | " <td>1</td>\n", |
|  |  | " <td>1405</td>\n", |
|  |  | " <td>1.0</td>\n", |
|  |  | " <td>1260759203</td>\n", |
|  |  | " </tr>\n", |
|  |  | " <tr>\n", |
|  |  | " <td>12</td>\n", |
|  |  | " <td>1</td>\n", |
|  |  | " <td>1953</td>\n", |
|  |  | " <td>4.0</td>\n", |
|  |  | " <td>1260759191</td>\n", |
|  |  | " </tr>\n", |
|  |  | " <tr>\n", |
|  |  | " <td>13</td>\n", |
|  |  | " <td>1</td>\n", |
|  |  | " <td>2105</td>\n", |
|  |  | " <td>4.0</td>\n", |
|  |  | " <td>1260759139</td>\n", |
|  |  | " </tr>\n", |
|  |  | " <tr>\n", |
|  |  | " <td>14</td>\n", |
|  |  | " <td>1</td>\n", |
|  |  | " <td>2150</td>\n", |
|  |  | " <td>3.0</td>\n", |
|  |  | " <td>1260759194</td>\n", |
|  |  | " </tr>\n", |
|  |  | " <tr>\n", |
|  |  | " <td>15</td>\n", |
|  |  | " <td>1</td>\n", |
|  |  | " <td>2193</td>\n", |
|  |  | " <td>2.0</td>\n", |
|  |  | " <td>1260759198</td>\n", |
|  |  | " </tr>\n", |
|  |  | " <tr>\n", |
|  |  | " <td>16</td>\n", |
|  |  | " <td>1</td>\n", |
|  |  | " <td>2294</td>\n", |
|  |  | " <td>2.0</td>\n", |
|  |  | " <td>1260759108</td>\n", |
|  |  | " </tr>\n", |
|  |  | " <tr>\n", |
|  |  | " <td>17</td>\n", |
|  |  | " <td>1</td>\n", |
|  |  | " <td>2455</td>\n", |
|  |  | " <td>2.5</td>\n", |
|  |  | " <td>1260759113</td>\n", |
|  |  | " </tr>\n", |
|  |  | " <tr>\n", |
|  |  | " <td>18</td>\n", |
|  |  | " <td>1</td>\n", |
|  |  | " <td>2968</td>\n", |
|  |  | " <td>1.0</td>\n", |
|  |  | " <td>1260759200</td>\n", |
|  |  | " </tr>\n", |
|  |  | " <tr>\n", |
|  |  | " <td>19</td>\n", |
|  |  | " <td>1</td>\n", |
|  |  | " <td>3671</td>\n", |
|  |  | " <td>3.0</td>\n", |
|  |  | " <td>1260759117</td>\n", |
|  |  | " </tr>\n", |
|  |  | " </tbody>\n", |
|  |  | "</table>\n", |
|  |  | "</div>" |
|  |  | ], |
|  |  | "text/plain": [ |
|  |  | " userId movieId rating timestamp\n", |
|  |  | "0 1 31 2.5 1260759144\n", |
|  |  | "1 1 1029 3.0 1260759179\n", |
|  |  | "2 1 1061 3.0 1260759182\n", |
|  |  | "3 1 1129 2.0 1260759185\n", |
|  |  | "4 1 1172 4.0 1260759205\n", |
|  |  | "5 1 1263 2.0 1260759151\n", |
|  |  | "6 1 1287 2.0 1260759187\n", |
|  |  | "7 1 1293 2.0 1260759148\n", |
|  |  | "8 1 1339 3.5 1260759125\n", |
|  |  | "9 1 1343 2.0 1260759131\n", |
|  |  | "10 1 1371 2.5 1260759135\n", |
|  |  | "11 1 1405 1.0 1260759203\n", |
|  |  | "12 1 1953 4.0 1260759191\n", |
|  |  | "13 1 2105 4.0 1260759139\n", |
|  |  | "14 1 2150 3.0 1260759194\n", |
|  |  | "15 1 2193 2.0 1260759198\n", |
|  |  | "16 1 2294 2.0 1260759108\n", |
|  |  | "17 1 2455 2.5 1260759113\n", |
|  |  | "18 1 2968 1.0 1260759200\n", |
|  |  | "19 1 3671 3.0 1260759117" |
|  |  | ] |
|  |  | }, |
|  |  | "execution\_count": 102, |
|  |  | "metadata": {}, |
|  |  | "output\_type": "execute\_result" |
|  |  | } |
|  |  | ], |
|  |  | "source": [ |
|  |  | "ratings[ratings['userId']==1]" |
|  |  | ] |
|  |  | }, |
|  |  | { |
|  |  | "cell\_type": "code", |
|  |  | "execution\_count": 103, |
|  |  | "metadata": {}, |
|  |  | "outputs": [ |
|  |  | { |
|  |  | "data": { |
|  |  | "text/plain": [ |
|  |  | "Prediction(uid=1, iid=302, r\_ui=3, est=2.70523000658422, details={'was\_impossible': False})" |
|  |  | ] |
|  |  | }, |
|  |  | "execution\_count": 103, |
|  |  | "metadata": {}, |
|  |  | "output\_type": "execute\_result" |
|  |  | } |
|  |  | ], |
|  |  | "source": [ |
|  |  | "svd.predict(1, 302, 3)" |
|  |  | ] |
|  |  | }, |
|  |  | { |
|  |  | "cell\_type": "markdown", |
|  |  | "metadata": {}, |
|  |  | "source": [ |
|  |  | "For movie with ID 302, we get an estimated prediction of 2.618. One startling feature of this recommender system is that it doesn't care what the movie is (or what it contains). It works purely on the basis of an assigned movie ID and tries to predict ratings based on how the other users have predicted the movie." |
|  |  | ] |
|  |  | }, |
|  |  | { |
|  |  | "cell\_type": "code", |
|  |  | "execution\_count": null, |
|  |  | "metadata": {}, |
|  |  | "outputs": [], |
|  |  | "source": [] |
|  |  | } |
|  |  | ], |
|  |  | "metadata": { |
|  |  | "kernelspec": { |
|  |  | "display\_name": "Python 3", |
|  |  | "language": "python", |
|  |  | "name": "python3" |
|  |  | }, |
|  |  | "language\_info": { |
|  |  | "codemirror\_mode": { |
|  |  | "name": "ipython", |
|  |  | "version": 3 |
|  |  | }, |
|  |  | "file\_extension": ".py", |
|  |  | "mimetype": "text/x-python", |
|  |  | "name": "python", |
|  |  | "nbconvert\_exporter": "python", |
|  |  | "pygments\_lexer": "ipython3", |
|  |  | "version": "3.7.4" |
|  |  | } |
|  |  | }, |
|  |  | "nbformat": 4, |
|  |  | "nbformat\_minor": 2 |
|  |  | } |