David Lee

Dax Martineau

12.1.2022

Final Project

1. Project Title
   1. Movie Recommendation
2. Project Description and Scope
   1. Movie Recommendations will be provided based on user input specifying what genres they do and don’t prefer
3. What Type of Agent? (Problem Solving, Logic, Learning, Other)
   1. Problem Solving
4. Description of the Task Environment (Performance Measure, Environment, Percepts, Actuators)
   1. Prompt user what genres they like
   2. Add movie title to list
   3. Add movieId to Id list
   4. Prompt user for genre they don’t like
   5. Add movie title to not-like list
   6. Add movieId to not-like Id list
   7. Compare list and not-like list and remove overlapping titles
   8. Compare Id list and don’t like Id list and remove overlapping Ids
   9. Take ratings for movieIds in list and get average rating
   10. Return movies and their ratings that match criteria
5. Description of any datasets used in the project.
   1. <https://files.grouplens.org/datasets/movielens/ml-latest.zip>
   2. A user generated movie rating database
   3. Movies.csv and ratings.csv contains movieId, title, rating, and genres
6. Screenshots

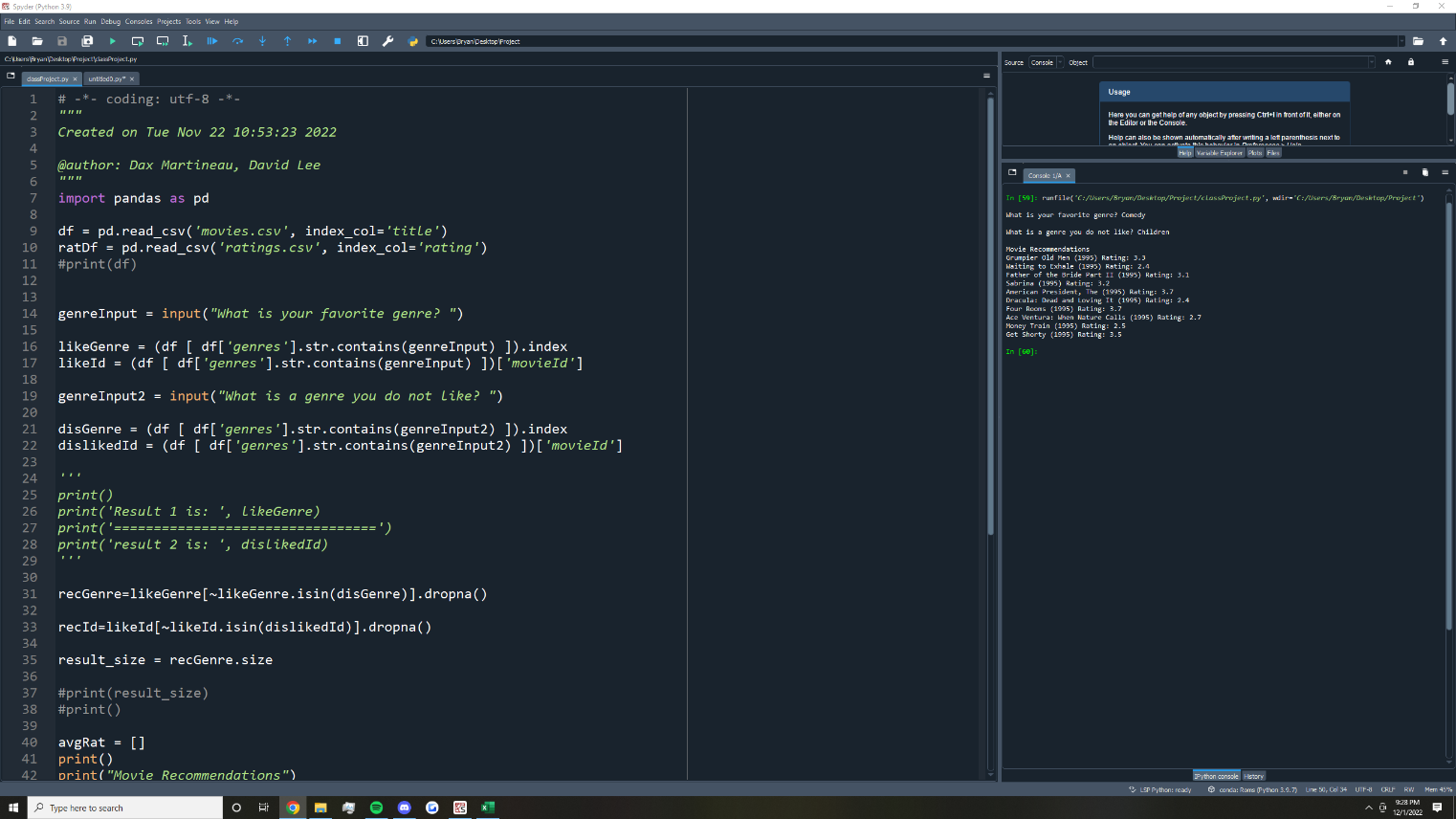
Text

Description automatically generated

Text

Description automatically generated

A screenshot of a computer

Description automatically generated with medium confidence

1. Plaintext code

# -\*- coding: utf-8 -\*-

"""

Created on Tue Nov 22 10:53:23 2022

@author: Dax Martineau, David Lee

"""

import pandas as pd

df = pd.read\_csv('movies.csv', index\_col='title') #import movieID title and genre dataset

ratDf = pd.read\_csv('ratings.csv', index\_col='rating') #import MovieId rating dataset

genreInput = input("What is your favorite genre? ") #promt user for genre they like listed in movies.csv

likeGenre = (df [ df['genres'].str.contains(genreInput) ]).index #query for genre and store movie title

likeId = (df [ df['genres'].str.contains(genreInput) ])['movieId'] #query for genre and store movieID

genreInput2 = input("What is a genre you do not like? ") #query for genre for genre they dont like listed in movies.csv

disGenre = (df [ df['genres'].str.contains(genreInput2) ]).index #query for genre and store movie title

dislikedId = (df [ df['genres'].str.contains(genreInput2) ])['movieId'] #query for genre and store movieID

recGenre=likeGenre[~likeGenre.isin(disGenre)].dropna() #removes unliked genres from liked movietitle list

recId=likeId[~likeId.isin(dislikedId)].dropna() #removes unliked genres from liked movieID list

result\_size = recGenre.size #store number of recomended titles

avgRat = [] #used to store that ratings per title

print() #formatting

print("Movie Recommendations") #formatting

for i in range(10): #get ratings for first ten results

likeRat = (ratDf [ ratDf['movieId'] == recId[i] ]).index #print movie title

if likeRat.size > 0: #make sure movie has ratings

ratList = 0.0 #prevents ratings from different movies from spilling over

for i in range(likeRat.size): #get average rating per movie

ratList += likeRat[i] #add rating from same movie

ratList = ratList / float(likeRat.size) #calcualte average rating per movie

avgRat.append(round(ratList, 1)) #adding rating to array that corralates to it's movie

for i in range(10):

print(recGenre[i], 'Rating:', avgRat[i]) #print the movie title and average rating of 10 movies