REC 2020 Programming Competition Report: Team Rabbits

The Problem:
With the recent COVID-19 outbreak and lockdowns occurring globally, many people now find themselves consuming more video entertainment than usual. This creates an unusually high demand for different streaming companies like our employer Netflux. With a large number of movies available on Netflux's catalog, users are having difficulty finding movies they would like to watch. This poses a problem for Netflux as we want the users viewing experience to be as easy as possible to keep them subscribed longer.

The Solution:
Decreasing the amount of time that it takes for a user to find the movies they would like to watch is the key to keeping customers happy. We can decrease this time by implementing a recommendation system to recommend movies that the user may like based on their interests.

High-Level Overview of the Solution:
- Parse Movie Data and put into usable data structures
- Ask the user for their preferences regarding movies such as actors, genres, keywords, directors, runtime
- Different biases are assigned to the different inputs so that the different preferences are weighed differently.
- Each movie is searched with each of the user's preferences and if a match is found in a particular movie then the bias associated with that preference is added to the "weight" of the movie
- The weight for each movie describes how compatible it is with the user's preferences
- When all of the weights are assigned to each movie, the top five weights in all the movies is output to the user

The Processes
Design Process
- Defining the requirements of the project
- Brainstorming possible algorithms to use in the recommendation system
- Thinking about our classes and methods we would need to use in such an algorithm/system
- Deciding on a specific algorithm to use and class structure for that algorithm
- Implementing the object-oriented design

Management Process
- Deciding on who will focus on which aspect of the project
- Delegating tasks fairly
- Periodically checking which members of the team need assistance with their part
- As a group, deciding which features are the most important to implement given time constraints

Development Process
- The data had to first be parsed correctly
- Decided which data structures would be most useful for this specific task
- Once the data was in the proper data structures we made the Movie Class which would hold all the data for a particular movie
- Once we were able to implement the Movie Class and test that it was working based on our requirements, then we moved on to the recommendation system.
- We decided in the design phase to have a Recommender Class which would contain our algorithm for recommending the movies.

- We then implemented our recommendation algorithm within this class and then tested it with several inputs.
- Once the code was working and outputting proper recommendations, we then went back to refactor and document the code base.

Our design, management, and development processes were all successful and resulted in a working prototype.

Libraries Used:
We used two built-in libraries for this project

1. csv (Comma Separated Values)
2. ast (Abstract Syntax Trees)

We used the csv built-in module in order to process the movie dataset provided to us. We realized after completing the processing that it would be easier and possibly more efficient to use the pandas library to parse the data. We used the ast built-in module to convert string representations of data structures into actual data structures that we can access and manipulate. Again we could have used the pandas library to do the exact same thing which would have resulted in less code being written and increased efficiency.

How the Software Works:
Our software project uses two classes and a main file to drive those classes. The two classes are Movie and Recommender. The Movie class is simply used as a data structure to hold movie data as well as data about whether or not it should be recommended. The Recommender class is used to store information about the users preferences and implement the recommendation algorithm. The core part of our software lies in the Weight method the Recommender class where it iterates through all the movie objects and assigns weights to each of the movies based on the users preferences.

Installation Method and Steps to use:
In order to run our program, the user will need to have python version 3.8.4 installed on their system.
Steps to execute:

1. Unzip the project folder
2. From the terminal, in the current directory, run the command:
   $ python3 main.py
3. The user will be prompted to enter their movie preferences.
   Example Input:
   Action Adventure Crime
   Christian Bale, Tom Hardy
   Christopher Nolan
   Christopher Nolan
   English
   long
   terrorist, dc comics
4. The top 5 five recommended movies will be output to a csv file in the same directory.

Citations:
Real Python. (2020, November 21). Reading and Writing CSV Files in Python. Retrieved November 28, 2020, from https://realpython.com/python-csv/
Ast - Abstract Syntax Trees¶. (n.d.). Retrieved November 28, 2020, from https://docs.python.org/3/library/ast.html