Chayton Morris

Final Project Paper

11/13/23

CS 470

Final Project Paper

**Overview:**

For my final project, I built a workout recommendation system using item-based collaborative filtering. The program uses a validation system that requires users to enter a userID and a password to enter the program. This validation system aids in storing user information in a user database that also stores workouts that have been previously recommended to the user. The program then uses item-based collaborative filtering to recommend workouts to users based on workouts that other users have completed, when both users have completed similar workouts. For new users who haven’t completed any workouts, the system will simply recommend workouts that are stored within the database. The database contains thousands of workouts split into sections by muscle group.

**Methodology:**

The first thing I did to begin my project was gather data. I searched many websites for datasets that contained a large number of workouts that were split into groups. I eventually found a dataset on Kaggle that contained thousands of workouts split into sections by muscle groups. I next researched machine learning techniques in order to create a recommendation system to recommend workouts to users. I quickly realized it didn’t make much sense to build my project that way as the most I could do was randomly choose workouts from specified groups. I then researched other ways to implement my idea while still using some type of machine learning. This is when I came up with the idea to build a user validation system. I tested many ways to implement this and eventually decided to store all of the user data within a csv file. This allowed me to store workouts to specific users that could later be used in a learning model. After, I began researching ways to use this user database to build a system that could recommend workouts to users based on workouts that similar users had completed. I then found a few videos explaining the use of item based collaborative filtering. After watching these videos I decided to implement this model into my system. I then had to research more about sklearn and eventually researched and learned about using a similarity matrix. This led me to the scipy library. After researching more of this library, I was able to implement it into my program. To test this model, I used my previous recommendation system to fill my user database with workouts for each user. I then implemented the collaborative item based filtering technique to find similar users and recommend workouts based on these findings.

**What I did and didn’t do:**

One of the first things I decided to do with my program is load the workout database into a dictionary. After the database was loaded and I performed tests to make sure everything was being stored correctly, I needed someway to prepare the dataset. To do this I used a dataframe and referred to the sklearn documentation on label encoding. I also used GeeksforGeeks to help with my knowledge of label encoding. From these sources I learned how to use label encoder classes and mapped the muscle groups and fitness levels by zipping them as dictionaries. Next, I had to train the model, to do this I used a random forest classifier. We briefly learned about this model in class and I used that knowledge to implement it into my program. My next task was creating user profiles. Creating new profiles, validating a users credentials, and loading a users credentials were all simple tasks that I was able to complete fairly easily. However, when saving user credentials I ran into issues with the writer completely overwriting every other row in the csv instead of simply entering the values to the respective row. For this problem I consulted many sources to eventually create the function that is in my program now. I then needed to build the model for item based collaborative filtering. I consulted the scikitlearn documentation to find a function that would suffice. I began with using a dok\_matrix but eventually changed to using the cosine\_similarity matrix to calculate the similarity matrix. This calculation was based on a workout matrix that was created using numpy. I also was unaware of how to join strings in python. I eventually found sources and used .join multiple times throughout this program. I also had to remove whitespace on many strings throughout the program, and consulted sources to learn the use of .strip() throughout my program. I also had to learn how to sort the indices of a numpy array to ensure the similarity matrix was sorted correctly based on the users completed workouts. Also, I had to learn how to change the dimensions of the label encoder for the numpy array to hold as many columns as possible. I consulted the numpy documentation for this and used .reshape(). I also consulted a few GitHub repos of other people who have created similar programs to generate ideas.

**Final results:**

In this project I managed to implement a machine learning model to create a recommendation system to users based on the completion of their previous workouts. I was also able to create a user validation system in which users could create accounts that will in turn store the workouts that they have completed. There were many things in this project that I wanted to accomplish but unfortunately did not. The biggest obstacle I faced when completing this project was time management without a doubt. I struggled with managing how much time to set aside to work on my project and highly underestimated the time needed. I learned that my knowledge of AI learning models was not sufficient enough to finish this program, and the help of outside sources was necessary to build the project I wanted. I learned more about sklearn libraries and item-based collaborative filtering methods that I was unaware of before. There are many things that I would like to work on and hopefully implement into this project. From the beginning I wanted to implement this program into an app and deploy it. The main drawback to this idea was the fact that I didn’t complete many of the other things that I wanted to with this project. One of the things I wanted to complete with this program is hiding user input when they are typing in their password. IDLE doesn’t allow input to be hidden so that could not be implemented without using some type of user interface. I also wanted the program to be able to let existing users chose the muscle group for their recommended workouts. This proved to be more difficult than I imagined and I resorted to only allowing newly created users choose the muscle group they wanted to be recommended. If this project continued I would definitely allow users to choose their muscle group along with choosing multiple muscle groups. I would also create some type of frontend interface for this project to make it appear more clean. Also, I would like to make the user experience more interactive by allowing users to see the workouts completed by other users while also seeing their own completed workouts.

**Resources:**

[Label Encoding in Python - GeeksforGeeks](https://www.geeksforgeeks.org/ml-label-encoding-of-datasets-in-python/)

[sklearn.preprocessing.LabelEncoder — scikit-learn 1.3.2 documentation](https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.LabelEncoder.html)

[Save results to csv file with Python - Stack Overflow](https://stackoverflow.com/questions/3345336/save-results-to-csv-file-with-python)

[How to Write to CSV Files in Python (pythontutorial.net)](https://www.pythontutorial.net/python-basics/python-write-csv-file/)

[sklearn.metrics.pairwise.cosine\_similarity — scikit-learn 1.3.2 documentation](https://scikit-learn.org/stable/modules/generated/sklearn.metrics.pairwise.cosine_similarity.html)

[1.11. Ensembles: Gradient boosting, random forests, bagging, voting, stacking — scikit-learn 1.3.2 documentation](https://scikit-learn.org/stable/modules/ensemble.html)

[sklearn.model\_selection.train\_test\_split — scikit-learn 1.3.2 documentation](https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.train_test_split.html)

[Python Dictionaries (w3schools.com)](https://www.w3schools.com/python/python_dictionaries.asp)

[Python Dictionary: How To Create And Use, With Examples •](https://python.land/python-data-types/dictionaries)

[Python String join() Method - GeeksforGeeks](https://www.geeksforgeeks.org/python-string-join-method/)

[Python String strip() Method (w3schools.com)](https://www.w3schools.com/python/ref_string_strip.asp)

[How to get the indices of the sorted array using NumPy in Python? - GeeksforGeeks](https://www.geeksforgeeks.org/how-to-get-the-indices-of-the-sorted-array-using-numpy-in-python/)

[numpy.argsort() in Python - GeeksforGeeks](https://www.geeksforgeeks.org/numpy-argsort-in-python/)

[GitHub - zinedkaloc/ai-workout-planner: Workout Planner&Fitness Coach - Using OpenAI Chat Completion API](https://github.com/zinedkaloc/ai-workout-planner)

[numpy.reshape — NumPy v1.26 Manual](https://numpy.org/doc/stable/reference/generated/numpy.reshape.html)