Mark Myers

C964 Task 2 Part D

Movie Recommendation Post-Implementation Report

The Organization’s Vision

Our organization was struggling to find a good way to recommend movies to our users. We were simply going off of the top-rated scores for movies and nothing else. This new application provides a way for the user to get a quality recommendation based on searching for a movie title, genre, actor, or director. This gives our user a more positive experience when using our service and helps to generate an expanding customer base.

Machine learning has been applied to the program to generate movie recommendations based on the title that is searched for. This will give the user movies similar to what they searched for and enjoy rather than getting a generic movie recommendation based on score.

Datasets

The raw data contains columns for movie title, rating, genre, score, director, writer, lead actor, and movie length. There are 7,633 movies stored in this dataset. The dataset was organized to be in alphabetical order by movie title (the name column in the dataset). Unneeded columns such as the country that produced the movie and the release date were removed. The dataset can be viewed in the root folder of the project marked as ‘movies.csv’. A screenshot of the beginning of the dataset can be viewed below.

Table

Description automatically generated

Data Product Code

The program was developed in PyCharm. A simple interface for the user to interact with is displayed in the console output screen. The user is able to select various options in order to view movies or data about the movies. The dataset was loaded into a pandas data frame and this data frame was used to give the results of the queries.

There are three descriptive methods: to view the average movie score per genre, the number of movies per rating, and the average score per rating. Data from the pandas data frame was loaded into either a dictionary, which were then broken down into lists by keys and values, or immediately separated into lists.

These visualizations were created using the matplotlib library. The method for developing these visualizations was referenced at <https://matplotlib.org/>.

The non-descriptive method is used to search for a movie title and find recommendations that are similar. The columns from the dataset are concatenated and then the words are turned into vectors using CountVectorizer from the sklearn library. The vectors are loaded into an array via cosine similarity, also from sklearn. All of this is done upon initial loading of the program. Once the method is called for the user to get a recommendation by searching for a title, the similarities are sorted by the index of the search movie and the results are displayed with the closest match listed on the top and descending from there.

The CountVectorizer and the cosine similarity functions utilize machine learning from the sklearn library. They are appropriate for this application since they take key words, such as the genre or the actors name, and factor how similar another movie is to the searched movie. This method was used since it is similar to the KNN algorithm and due to its ease of implementation. Multiple tests were done using different movies and analyzing the results to see if the recommendation made sense.

The machine learning processes were created using the sklearn library. The method for developing the implementation was referenced at <https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.CountVectorizer.html> and <https://scikit-learn.org/stable/modules/generated/sklearn.metrics.pairwise.cosine_similarity.html>.

Objective Verification

The objective of the project was to give users quality movie recommendations based on movies similar to a specific movie. The objective was met as movie recommendations are provided to the user based on a movie title for which they searched. In addition, there are recommendations based on a genre, director, or lead actor and presented according to score.

Effective Visualization and Reporting

The visualizations provide a picture of what kind of movies people typically enjoy. Interestingly, the average score per genre is relatively close to each other. And while movies rated ‘R’ have the most occurrences in the dataset, the average score for the rating is third from the bottom.

These visualizations were chosen for the ease of creation, as well as it is a good way to show common trends and offer information showing that movie in other genres or ratings might be enjoyed more. The three visualizations can be viewed below.

Chart, pie chart

Description automatically generated

Chart, bar chart

Description automatically generated

Chart, scatter chart

Description automatically generated

Accuracy Analysis

The non-descriptive method is accurate for its recommendations. It takes the title, rating, genre, score, director, writer, and lead actor and finds the movies based on the most common features from these categories. In the future, improvements could be made by having a dataset that has descriptions of the movies to help narrow down more similar movies rather than just being based on basic data.

Seen in the screenshot below is the recommendations based on the movie ‘Men in Black’. As one would expect, ‘Men in Black II’ and ‘Men in Black 3’ are on the top of the list of movies returned.

Text

Description automatically generated

Application Testing

As the program was coded, there was continual and extensive testing. Data loaded into the list to construct the visuals were printed to the console to ensure that the proper data was loaded and in the order desired. Originally, there was not input check for the interface and incorrect inputs would result in program errors and termination. In addition, for the CountVectorizer, the number of max features had to be reduced several times due to lack of memory space.

Application Files

The program was coded in PyCharm 3.7. Libraries needed to run the program are: sklearn, pandas, matplotlib, re, and collections.

The submission includes a zip file containing the .py file and the .csv file. The Post-Implementation Report, Letter of Transmittal, Project Recommendation, and the Project Proposal are also included in the zip file.

User Guide

1. Open PyCharm and load the program main.py
2. In the terminal, install the necessary libraries:

* pip install regex
* pip install sklearn
* pip install matplotlib
* pip install pandas

1. Press the run button and wait for the program to load.
2. Type 1 in the console after ‘Please make a selection:’ to enter the search function.
3. Searching by title will result results containing the words enter, i.e.: ‘good’ in the title search will return all movies that have the word good in the title. Similarly, searching director and entering ‘Steven’ or entering ‘Tom’ in the lead actor search will display the appropriate results.
4. Return to the main menu by selecting the appropriate number.
5. Type 2 in the console to enter the view function.
6. Options 1 and 2 will display the highest and lowest rated movies respectively and display the results in the console. Options 3, 4, and 5 display the visuals of the program. Close the box that opens in order to proceed.
7. Return to the main menu by selecting the appropriate number.
8. Type 3 in the console to enter the recommendation function.
9. Option 1 is the machine learning function to get a recommendation based on the title. Type in ‘Goodfellas’ to see the recommendations. Try some other popular movies as well. Options 2, 3, and 4 display recommendations of genre, lead actor, and director respectively. Results are displayed in the console.
10. Select exit to close the program.

Summation of Learning Experience

I have previously taken a couple of classes focused on Python. The basic syntax was easy for me to apply. I had not, however, ever used pandas, so that took me some time to understand the difference in the syntax as well as learning the tricks to do what I wanted to do. I had to do a lot of reading and watch several tutorial videos to understand things. I had previously used sklearn, but it was very limited. I watched several tutorial videos to understand how to implement the functions effectively. I also had not previously used matplotlib. It took some thinking and trials to see what visuals to use and display and how to get the data that I needed to work accordingly.

Working on this project has been a learning experience. I had not previously used a lot of libraries, and it was interesting to be able to learn different aspects of programming and how to do different things. This project has helped show me that there is a multitude of libraries and ways to go about things. It has made me want to learn more libraries and further expand my knowledge base to improve my programming skills.