**Final Project: Movie Night**

**Team Members**

Connie Nguyen

Lap Wu

Joshua Nezianya

Katherine Cheng

1. **Summary of group project, accomplishments and learning experience**

Throughout the semester this group project helped us improve our abilities as programmers, but more importantly this experience improved our resolve as people. We all learned what it was like working with others. We likened it to a work environment where you may work separately but there comes a time when we all come together.

Our group project focused on looking at movie databases and extracting specific data that answered some questions we each posed at the beginning of the project. In order to extract and display the information we found using these databases, we implemented many of the tools python provides us, this included APIs, pandas, word clouds, dictionaries and many other things. Although we all ran into roadblocks and issues either as a group, or as individuals, we pushed through and were able to complete our final project. We accomplished what we needed to and we all learned a lot through this shared experience.

1. **Summary of individual accomplishments and learning experience**

Connie: I learned a lot through this project. This is one of the most hands-on courses I have ever taken, it required me to be thoroughly engaged in every activity in each week and for me to do more research outside of classwork. This process not only applied to classwork but also to this project. This was a big project that required a good understanding of python and how each of its tools can be used. In the beginning I ran into a lot of errors trying to work with all these new tools I had never worked with before like APIs, pandas and the matplot library but with more practice and research I was able to push through.

In the beginning, I relied a lot on my group members, being the youngest member and still being in highschool I have a lot to learn about programming and how to work with others, so I am very grateful for the members that helped me along the way. As the group project continued, I found myself more confident in my abilities to program. Something I think I would do differently if I could was possibly add more questions and more complexity to my implementation. I also think that I could have made my program much more simpler in the beginning portion in which I extract data from the API. In the future I want to implement a portion of code that displays the upcoming movies and possibly scapes the web for images/posters for those upcoming movies. My three questions were the top rated movies, the movies with the longest runtimes and the movie genres with the highest/lowest ratings.

**API:** The first implementation I added to my portion of the program was using an API to extract information about moving ratings. In the beginning I struggled a lot with API usage and constantly asked questions, but with the help of my group members I was able to accurately extract the information I needed. In the beginning I was not correctly calling on the API, so I kept getting a bad status code, I was able to fix this and I implemented a section of code that checks the status code before performing any operations. From there I used other python tools to extract the information.

**JSON:** After implementing the API, I used JSON, I loaded the information extracted from the API into JSON format, this made it easier for me to access and extract the information I needed to. I was able to simply get the data from the section labeled, ‘results’ and then later from that section get the ‘original\_title’ and ‘vote\_average’. From here I used other python tools to display the information I needed.

**List/Dictionary 1:** I then used a for loop to extract the first twenty from the results which are the top twenty rated movies. For each of these results, I extracted the title and the rating and stored them into two different lists. I also created a dictionary that had one key called ‘Ratings’ and stored the list of ratings as its values. Using these lists and dictionaries I was able to display the information using matplot and pandas.

**Pandas 1:** I used the dictionary and list from before to create a dataframe containing the ratings of each movie along with its movie name. I then used a simple method display(), to display the dataframe in a tabular format.

**Matplot 1:** Using the lists from before I was able to display the top twenty rated movies in a plot graph, with the movie names labeled on the x-axis and the ratings on the y-axis. Although the plot does display as I wanted, due to some issues with how matplot renders languages, the plot displays incorrectly because some of the movies were not in English. I spent a lot of time researching, trying to find the issue but could not find one which would work for any language, instead I would have to know what specific language to check for and translate.

**Pandas 2:** The second portion of my program read data from a csv file and I used pandas to read from that file and created a dataframe. With this I was able to create smaller subsets of data, smaller dataframes for each of the specific sets of information I needed. This included getting movies with the longest runtimes, genres with the highest ratings and the genres with the lowest ratings. Not only with pandas could I create smaller subsets of data I could also sort them in the way that I wanted, by ratings and I could gather the top 10 (head) or bottom 10 (tail). I really enjoyed using pandas throughout this project as it was very useful.

**Matplot 2:** The second use of matplot was to display a bar graph with the information about the highest movie runtimes. With this I display the movie titles on the x-axis and the time in minutes on the y-axis, this helps to compare how long the movies were individually.

**Word cloud 1:** My first use in word cloud was to display the genres of highest rating movies. So the genres that appeared more in the top 250 movies display in larger text while those genres that did not perform as well displayed in smaller text.

**Word cloud 2:** Similar to the first use of word clouds, the genres that performed the worst appeared in larger text while those that performed better (appeared less in the bottom 250 movies) appeared in smaller text.

Lap:

Working on this project helped me learn how to use APIs and process the data from an API. For my part of the project, I answered the questions of what genres are the most prevalent in the top rated or most popular movies. I used the movie database API to retrieve the data and JSON and Pandas data frame to process the data.

**API:** I retrieved three sets of data from this API: 160 best rated movies, 160 most popular movies, and the genre id keys. I retrieved the 160 movies by going through 8 pages of movie data, one page at a time. Initially, I wanted to retrieve a massive amount of movies at once or only retrieve data of a certain genre, but I didn’t find a way to do these things with this API.

**JSON:** All the data sets retrieved from the API are loaded into this data format. I used this format because it made extracting certain information much easier.

**Pretty Printer:** This display is really helpful in displaying all the JSON data in an organized manner. By displaying the JSON data with Pretty Printer, the data becomes much easier to read. This display helped me search for the necessary information through displaying data in an organized fashion and I used this method for displaying JSON data due to its convenience.

**List and dictionary:** I usedthe dictionary and the list class to store the information in an organized manner. I stored the genre information, the movie’s genre ids and genre names, in a list. I used the dictionary class to keep track of each movie and their info list, with the movie’s title as the key. Using this format, I was able to get each movie’s genre information using their title, a format that I can use and remember fairly easily. I also used a dictionary to keep track of how many counts there are in each genre.

**Pandas:** I used the panda data frame to organize the genre data into a table, which is useful for creating charts. This object class is also helpful in sorting the value and limiting how many genres that appear in the chart.

**Matplot :** I created two plots using the genre data frames. I created a pie chart for the top rated movie genres and a bar graph for the most popular movie genres. Both of these graphs show which genres are the most prevalent. I used a bar chart for the popular movie because the genre counts are closer to each other making it difficult to tell which is greater in a pie chart.

Joshua:

This was a challenging project. It incorporated all of the concepts that we learned this semester. For my contribution to this project, I decided to focus on a few questions and functionalities. I wanted the user to be able to look up their favorite movie, see the ratings for that movie from the most popular review sites. I also wanted to answer these questions.

1. What are the stats for your favorite movie?
2. Is your favorite movie one of the top 200 highest grossing films of all time?
3. What’s the full cast for your movie?
4. What are the highest grossing films domestically/international?

**API**: To return movie lookup, I decided to use the IMDB API. It’s pretty comprehensive and it has several endpoints. I used the search movie, highest grossing films, ratings, and full cast endpoints. The difficulty here was data retrieval, but also data manipulation. Cross referencing data points from multiple endpoints can be difficult. The data structures that you use to store the data becomes increasingly important. Also, you have to be cognizant to not call your endpoint too many times

**List/Dictionaries:** I decided to store the majority of my data in lists of dictionaries. I knew that I would have to loop through lists multiple times for comparisons. For instance you’ll see in my code that I get a list of cast members from a movie and I cross reference the unique movie id to determine whether the film the user selects is in the top 200 highest grossing films. I created a dictionary of lists to generate a csv file.

**Pandas 1 & 2:** A graph can be generated using a dictionary, but it’s much easier to do so with a csv file. I thought the data that I had from the API was sufficient so I didn’t want to search the web for another data file to parse. I decided to use some of the lists I used in prior functions to create a dictionary. I converted the dictionary to a csv using pandas. I also used pandas to create dataframes for the critic reviews and also for the highest grossing films.

**JSON:** Data is passed around the web using JSON. When you do an API call to the ImDB API, it returns a list of dictionaries. To work with APIs for this project, I used the request library, but I also used the json library to load the endpoint data after converting the call to text. As I mentioned earlier, I either stored the data in a list or dictionary.

**Matplotlib 1 & 2:** To show the data from the csv that I created, I decided to use Matplotlib. I created two graphs. One is a bar graph and it shows the years with the highest ranked movies. The second graph is a line graph and it shows the years with the highest total gross.