The implementation of the web scrapper involved utilizing the BeautifulSoup library, which was used for parsing a requested webpage in HTML format. Movieweb.com was chosen to be scrapped due to the arrangement of movies on this site. Particularly, Movieweb displayed movies in a list arrangement with the relevant data needed for this project, such as title, plot, and genre, when viewed by year.

The web scrapper that was implemented will iterate through every page of results for each year that is specified. On the first iteration for a particular year, the pagination links will be parsed to get the last page of results for that year. This was required for the web scrapper because Movieweb will not return a 404 HTML Error for a URL that is specified with a page that does not exist; Movieweb will simply return the last page in that year’s results.

For example, there are only 19 result pages for the year 2019. If a GET request is sent to the Movieweb server with the year 2019 and page 58, Movieweb will return page 19.

Initially, the web scrapper was implemented in a way that wrote all the results for a particular year to a .csv file after the last page of the results page for that year was parsed. However, since Movieweb is structured in the manner described above, the pagination link for the last page would have to be stored for comparison as a way to exit the iteration and write to .csv. Otherwise, the last page would be appended forever with many duplicate entries.

The web scrapper would send a GET request to the Movieweb server with a properly formatted URL based on the page iteration that was going to be parsed. The BeautifulSoup html.parser was then used to extract the a list of movies that were listed on that HTML page. On a ‘full’ page, there would be a list of twenty movies. This list would then be iterated through to extract the title, plot, and genre for each movie. If the genre for each movie matched one of the genres that our project was classifying for and if the genre was present and if the plot for the movie contained more than 127 characters, the movie would be stored in a pandas DataFrame along with all the other relevant information. However, if the genre did not match or if the genre was not present or if the plot was too short for analysis, the movie would be discarded. For the purposes of this project, the years 1910 to 2020 were scrapped for supplementing the training set. Any duplicate entries were removed in a separate script that aggregated the scrapper data with the previous training data from Kaggle. Any remaining entries that were not used for the training set were used for the test set.