A. About the project

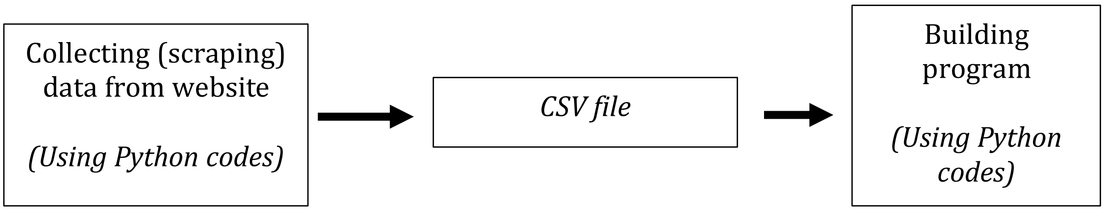
1. General content

This project is created for people, especially movie executive producers, who would like to gain an idea on the potential gross of a genre (a type of a movie) given its specific budget.

The principal method used to predict in this program is linear model based on the budget and the gross of each genre. That information was collected from data in the past (data before 2019).

2. Development process and problems

The general process for the project was drawn as below.



Because of time-comsuming process of scraping data from website, the whole process was divided into 3 mains as above for speeding up building the final program.

The processes for building code of collecting data and building program are attached in appendix.

2.1. Collecting data

The data was collected from website named “The Numbers”. In particular,

- All the elements which are “Release date or years”, “Production Budget”, “Worldwide Gross” and “Domestic Gross” were extracted from https://www.the-numbers.com/movie/budgets/all
- The “Genre” was collected from the hyperlink attached under the name of the movie from the link above.

The main porblems and solution were indicated in the table below.

Problems	**Solutions**
Errors which were difficult to discover	- Narrowing down the size of observations (movies) and choosing the page which seemed to have a lot of problems (such as missing data or different structures) to discover. - Looking for the same errors on internet to find the solutions.

Some data were hard to identify the location to extract because of missing typical features such as id or name of the class in inspection (e.g.: extracting data from website table without any id or class in inspection)	- Looking for other features (might be less typical but can be solved by loop or 'if' functions). For example: Looking for the class with id/name that contains that table and use 'for' loop or 'if' functions to select.
Time-consuming for scraping	- Trying to solve the codes many weeks before the due date and organise the to-do-list tasks for every day. - Separating the full pages into 6 bins for easily discovering the errors and saving time, particularly: ▪ Bin 1: 0 – 1000 movies ▪ Bin 2: 1000 – 2000 movies ▪ Bin 3: 2000 – 3000 movies ▪ Bin 4: 3000 – 4000 movies ▪ Bin 5: 4000 – 5000 movies ▪ Bin 6: 5000 – 5855 movies After this, combining all into one csv file

2.2. Data file

After extracting, the data was written into a csv file. Due to a lot of missing amount in "Domestic Gross", the scripter did not process this data further. As to "Genre" and "Year", the movies without that specific information were treated as "Unknown" for easy classification and was not included in the final program. These solutions for missing data would be efficient because of the huge number of movies.

2.3. Building the program

Basically, the scripter gathered all the types of genres which were collected before 2019 from the website and put into a box (combobox). After that, the linear model was applied and built for each genre on its gross and budget by splitting data into training testing dataset. Next, the writer applied the graphs into each genre and built the results (predicted grosses) and comparisons through designing structures in GUI.

The problems and issues at this stage were mentioned in the below table.

Problems	Solutions
Most of the examples or lessons about Python code online is for Windows, but the scripter uses OSX system.	- Taking time to understand the codes and finding the similarity, then applying to OSX.
Finding lesson sources to understand new knowledge	- Finding the best source through videos on Youtube, Google or similar examples on stackoverflow website.

3. Application and further development for the project

3.1. Application

As mentioned above, apart from helping movie executive producers to predict the gross of a genre based on the given budget, this project also helps them to compare with other genres' grosses or types of gross-budget linear relationships to have the most suitable decisions.

3.2. Future development

At first, for better prediction, the algorithm could be changed into more flexible equations such as polynomial regression with more variables to predict than one budget (such as names of directors, actresses, actors, types of ratings, and public scores) to gain ideas about suitable actors/actress, public scores and so forth.

Also, instead of predicting the gross, the movie executive producers could predict the profit which includes all expenses. This result could give the movie executive producers more precise decisions.

Besides, the program could be updated by adding more functions/ features such as the year (or period of time) that the movie executive producers would gain the “break-even” point (when the revenue equals the expenses to start to gain the profit) with that investment.

B. Instructions for using

(The instructions below were based on OSX system, it might not perform properly or exactly the same other system (such as on Windows or Linux), but the function should be the same.)

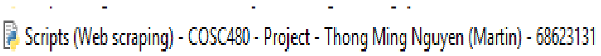
After extracting zip file, a folder including one python file for web-scraping, one data csv file, and one program python file in one folder under the name "COSC480 - Project - Thong Minh Nguyen (Martin) – 68623131".

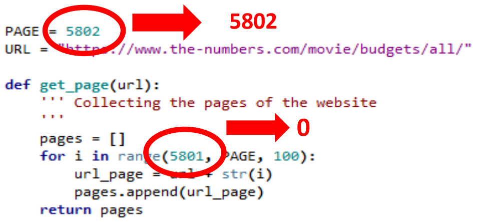
1. Instruction for web-scraping

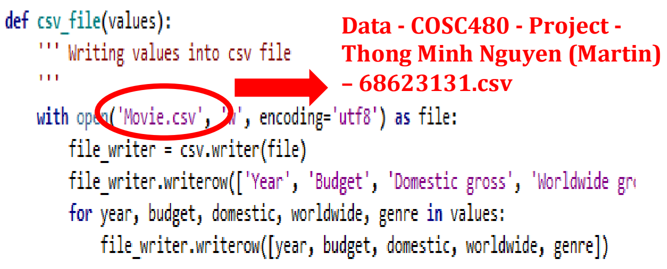
- The codes can be run on Python (version 3.6) and Wing IDE 101 6.1
- The packages which would be used were listed as table below for installing.

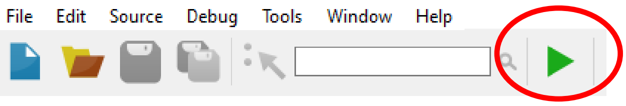
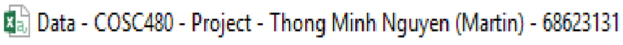
Packages	***Windows*** (in cmd)	***OSX*** (in terminal)
Requests	pip install requests	pip or pip3 install requests
beautifulSoup	pip install beautifulSoup	pip or pip3 install beautifulSoup or beautifulSoup4

- The instructions to scrape the movie data from website was described in the below table.

Instruction	**Pictures**
Open the file name "Scripts (Web scraping) - COSC480 - Project - Thong Ming Nguyen (Martin) – 68623131.py" on Wing IDE 101	 Scripts (Web scraping) - COSC480 - Project - Thong Ming Nguyen (Martin) - 68623131

Step 1:
Selecting the pages that need to be extracted (for all pages: from 0 → 5082) or it was set as in the picture below (5801 and 5082 were a trial for the last page).
	
	PAGE = 5802
	URL = "https://www.the-numbers.com/movie/budgets/all/"
	
	def get_page(url):
	    ''' Collecting the pages of the website
	    '''
	    pages = []
	    for i in range(5801, PAGE, 100):
	        url_page = url + str(i)
	        pages.append(url_page)
	    return pages
	5802
	0

Step 2:
Typing the file name for extracting (this name should be consistent with the name for calling in later program file, "Movie.csv" in the pic was a trial, and it should be change into official name "Data - COSC480 - Project - Thong Minh Nguyen (Martin) – 68623131.csv")
	
	def csv_file(values):
	    ''' Writing values into csv file
	    '''
	    with open('Movie.csv', 'w', encoding='utf8') as file:
	        file_writer = csv.writer(file)
	        file_writer.writerow(['Year', 'Budget', 'Domestic gross', 'Worldwide gr
	        for year, budget, domestic, worldwide, genre in values:
	            file_writer.writerow([year, budget, domestic, worldwide, genre])
	Data - COSC480 - Project - Thong Minh Nguyen (Martin) – 68623131.csv

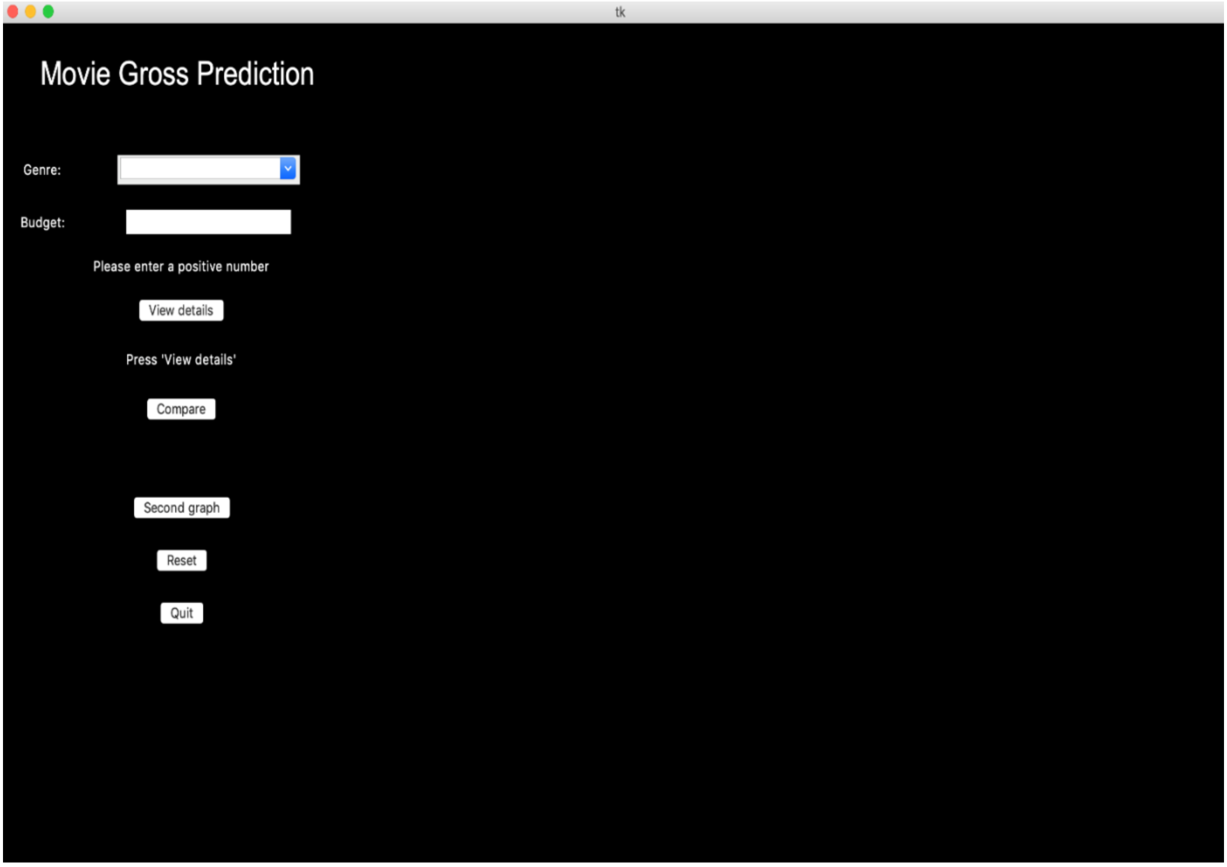
Step 3: Click on green play button in the item bar menu (under main bar menu)	
After **5 hours**, a file (as the picture) will appear under csv form in the same folder.	 Data - COSC480 - Project - Thong Minh Nguyen (Martin) - 68623131

2. Instruction for "Movie Gross Prediction" program

- The codes can be run on Python (version 3.6) and Wing IDE 101 6.1
- The packages which would be used were listed as table below for installing.

Packages	***Windows*** (in cmd)	***OSX*** (in terminal)
pandas	pip install pandas	pip or pip3 install pandas
numpy	pip install numpy	pip or pip3 install numpy
seaborn	pip install seaborn	pip or pip3 install seaborn
sklearn	pip install sklearn	pip or pip3 install sklearn
matplotlib	pip install matplotlib	pip or pip3 install matplotlib

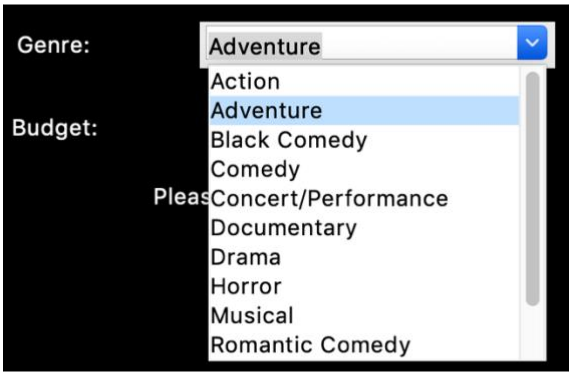
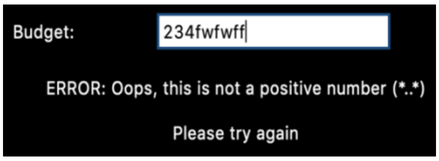
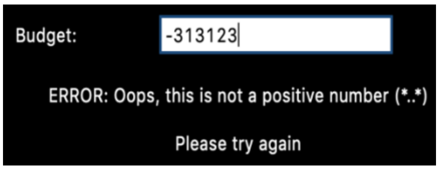
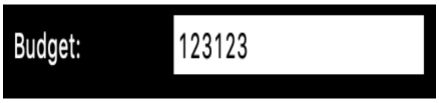
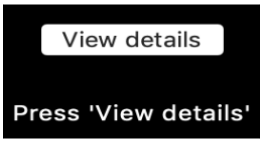
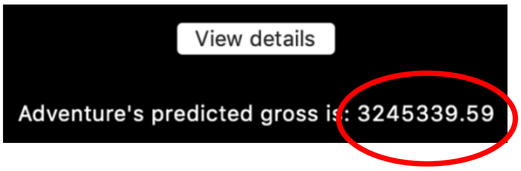
- After opening python file named "Scripts (Web scraping) - COSC480 - Project - Thong Minh Nguyen (Martin) – 68623131.csv", and clicking on ▶ button, the program as below would appear after 1 minutes.



Please enter a positive number

Press 'View details'

- From that screen, the instruction for using it would be described in the table below.

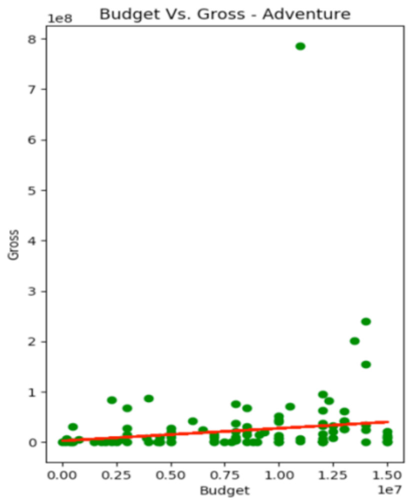
Instruction	Pictures
Step 1: Choosing the genre from the selection box. (There are 12 selections following alphabetical order. The example is "Adventure")	 Genre: Adventure Budget: Pleas Action Adventure Black Comedy Comedy Concert/Performance Documentary Drama Horror Musical Romantic Comedy
Step 2: Typing your budget. The number should be in a correct form of positive number (or the message "ERROR: Oops, this is not a positive number (*..*) Please try again" will appear.)	✗  Budget: 234fwfwff ERROR: Oops, this is not a positive number (*.*) Please try again ✗  Budget: -313123 ERROR: Oops, this is not a positive number (*.*) Please try again ✓  Budget: 123123
Step 3.1: Clicking "View details" button	 View details Press 'View details'
Step 3.2: After that, the predicted gross appears (based on the linear model of that genre – equation: y = ax + b)	 View details Adventure's predicted gross is: 3245339.59

Step 3.3:

The graph would also be displayed on the right hand simultaneously.

The graph shows the linear relationship between the gross and budget of 'Adventure' genre (built from the past data). In this case, the relationship is positive and the predicted gross is also postive, thus a movie executive producer might want to invest in this Adventure.

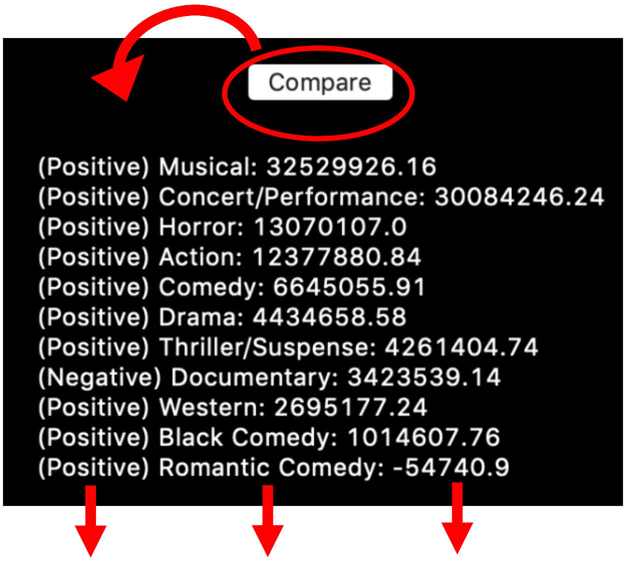
However, he/she might also want to compare with other genres' grosses.



Step 4:

Clicking "Compare" button:

- The list of other genres, genres' grosses, and genres' linear model type (positive or negative) appear (based on the same budget as the selected genre's).
- They are displayed as gross descending order.



```
(Positive) Musical: 32529926.16
(Positive) Concert/Performance: 30084246.24
(Positive) Horror: 13070107.0
(Positive) Action: 12377880.84
(Positive) Comedy: 6645055.91
(Positive) Drama: 4434658.58
(Positive) Thriller/Suspense: 4261404.74
(Negative) Documentary: 3423539.14
(Positive) Western: 2695177.24
(Positive) Black Comedy: 1014607.76
(Positive) Romantic Comedy: -54740.9
```

Relationship (budget & gross)

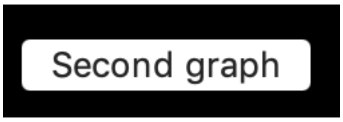
Genre

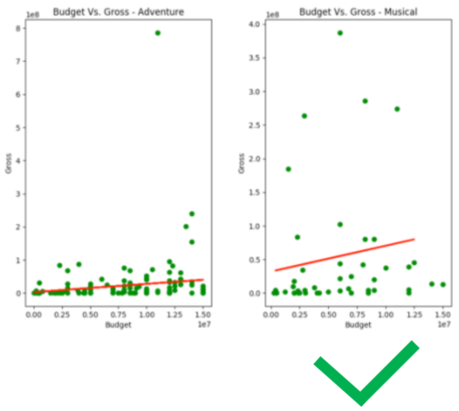
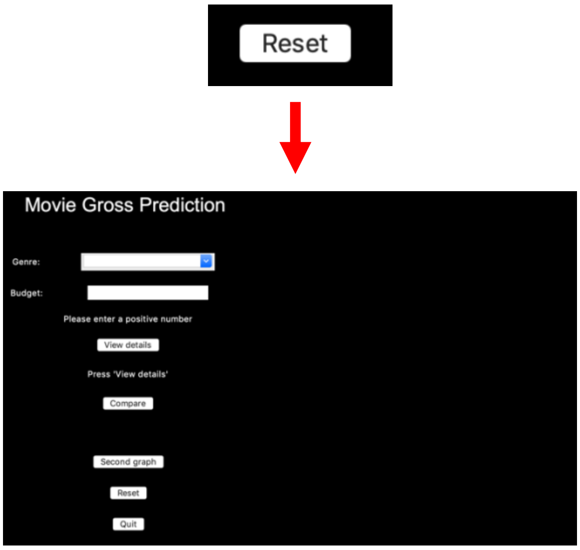
Predicted gross

Step 5.1:

The movie executive producer might also want to see the graph of the genre which has the highest gross in the list (in this case, 'Musical' genre has the highest gross) to compare.

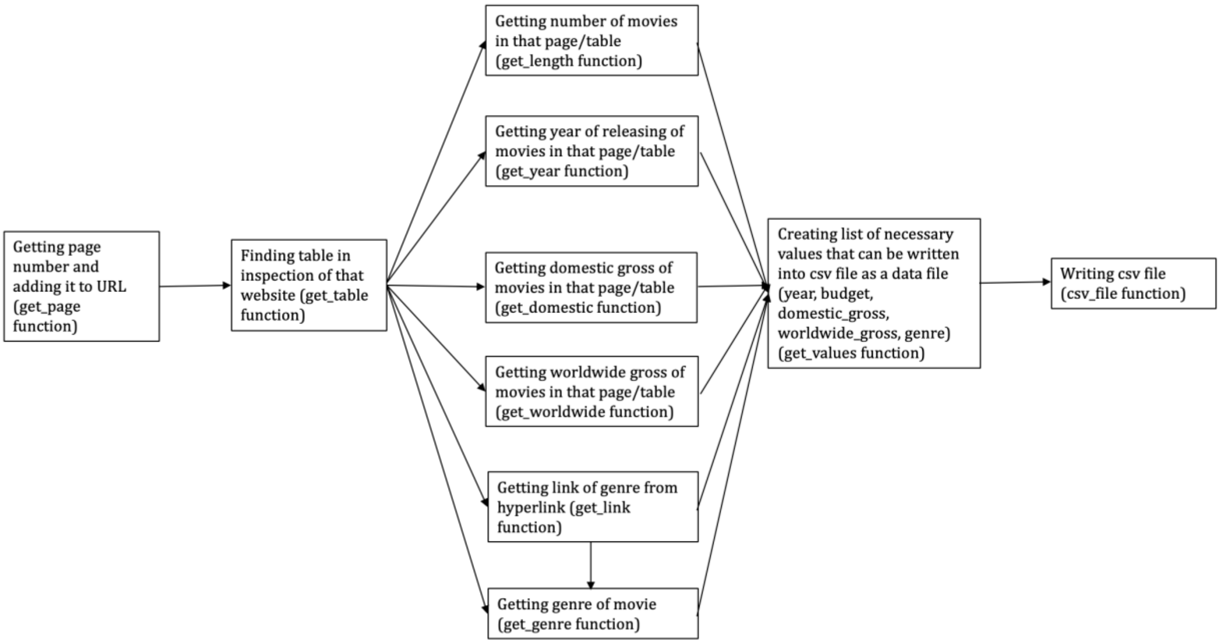
→ Clicking "Compare" button



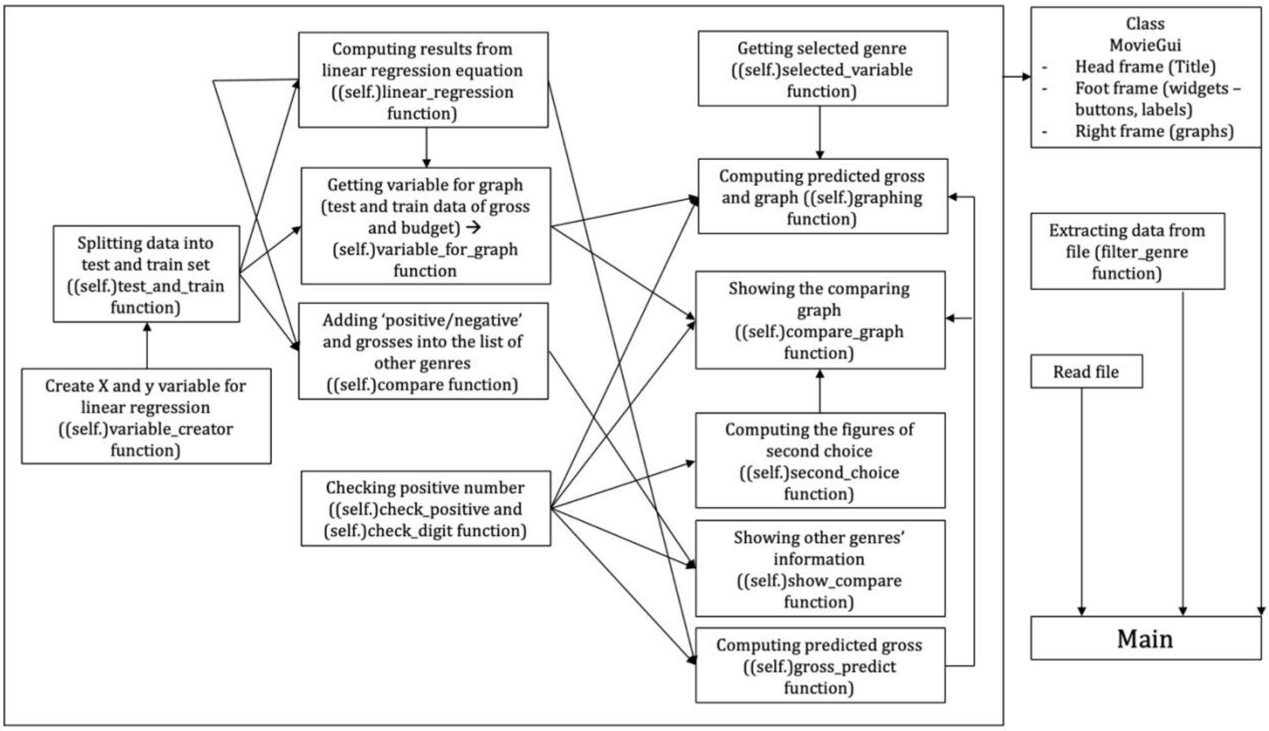
Step 5.2: After clicking, another graph (of the 'Musical' genre) would be shown next to the first graph to compare. In this case, it might be a better choice to invest in 'Musical' genre than 'Adventure' genre due to higher gross with the same budget (this is based on the higher slope of the linear model).	
Step 6: If the movie executive producer wants to start the process with other budget or other genre, he/she can click on "Reset" button to start the new one.	 Movie Gross Prediction Genre: Budget: Please enter a positive number Press 'View details'
Step 7: If the movie executive producer would like to cancel out the program, he/she can click on "Quit" button.	

C. Appendix

Web-scraping process


Getting page number and adding it to URL (get_page function)
Finding table in inspection of that website (get_table function)
Getting number of movies in that page/table (get_length function)
Getting year of releasing of movies in that page/table (get_year function)
Getting domestic gross of movies in that page/table (get_domestic function)
Getting worldwide gross of movies in that page/table (get_worldwide function)
Getting link of genre from hyperlink (get_link function)
Getting genre of movie (get_genre function)
Creating list of necessary values that can be written into csv file as a data file (year, budget, domestic_gross, worldwide_gross, genre) (get_values function)
Writing csv file (csv_file function)

Program process


Computing results from linear regression equation ((self.)linear_regression function)
Getting variable for graph (test and train data of gross and budget) → (self.)variable_for_graph function
Adding 'positive/negative' and grosses into the list of other genres ((self.)compare function)
Splitting data into test and train set ((self.)test_and_train function)
Create X and y variable for linear regression ((self.)variable_creator function)
Checking positive number ((self.)check_positive and (self.)check_digit function)
Getting selected genre ((self.)selected_variable function)
Computing predicted gross and graph ((self.)graphing function)
Showing the comparing graph ((self.)compare_graph function)
Computing the figures of second choice ((self.)second_choice function)
Showing other genres' information ((self.)show_compare function)
Computing predicted gross ((self.)gross_predict function)
Class
MovieGui
- Head frame (Title)
- Foot frame (widgets – buttons, labels)
- Right frame (graphs)
Extracting data from file (filter_genre function)
Read file
Main