ITBW51 - Text and Social Analysis

Proposal

Submitted By:	Parikshit Joshi (210895H) Jia Jun (210897S) Skye Ong (210903A)
Module Group:	ITBW51 - 01
Team Name	Reverse Oreo
Module Tutor:	Ms Jane Zhang
Submission Date:	15/1/2023

Description of business area:

The streaming services have been blooming and genre has been an important part in movies, which we will be investigating in. A movie's genre determines its structure and type of audience.Streaming services categorise their content according to genre, such as action, comedy, drama, and others. It makes it easy for users to find and browse content they are interested in. Genre classification is also used for targeted advertising and content recommendations. Netflix, Amazon Prime Video, and Disney Plus dominate the streaming services industry.

Background:

The number of streaming services that are interested in determining movie genres automatically is steadily rising.

Although genre is crucial to the description, exploration, and discovery of films, it is rarely subjected to extensive quantitative research.

Movies are categorised by genre, in other words. It is simpler for the spectator to find movies that he or she would enjoy and desire to watch when movies are categorised. The streaming firms can identify the genre that is now popular among their audience and can add more content in that genre, increasing viewer watch rates.

Scenario:

We are an upcoming media streaming company,ZenStream, that is interested in improving their recommendation system for their users. For content streaming companies to succeed in this business area, they must understand the media and entertainment industry, as well as their target audience preferences and viewing habits.Despite having a large dataset of user reviews, these reviews do not include the genre of the content. Users' reviews will be used to classify the genre of each piece of content. By analysing this information, they will be able to improve their recommendation algorithm and suggest content that users will be more likely to enjoy based on their previously expressed preferences. The classification project involves training a machine learning model on a dataset of labelled reviews, with the label representing the genre of the content, and then using that model to predict the genre of new, unseen reviews.

Objective:

The business objective for a classification project that classifies genre by looking at user reviews is to enhance the media streaming company's recommendation system. The company will be able to match content to user preferences better by accurately categorising movies and TV shows based on user reviews. As a result, users will be more inclined to find content they enjoy, increasing engagement and retention. Increasing subscribers and consumption of content will ultimately drive revenue growth for the company. Furthermore, the classification project will help the media streaming company to understand their user's preferences, which will help them to create targeted marketing campaigns and to improve their content curation.

Main & sub tasks for the team. Delegation of tasks/responsibility.

We will be using reviews given by customers/critics from movie reviews websites to predict the genre of the movies. We will be categorising mainly on 4 genres, action, sci-fi, romance and comedy.

Each genre, we will scrape about 350-400 rows.

Name	Website	Genre
Parikshit	https://www.imdb.com/	Action
Jia Jun	https://www.rottentomatoes.com/	Romance & Sci Fi
Skye	https://www.metacritic.com/	Comedy

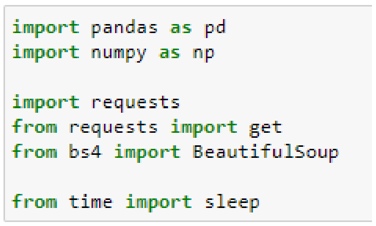
Columns

1) Title
2) Year
3) Genre
4) Ratings
5) Metascores
6) Reviews

Description of how we will collect our own dataset.

IMDB - Parikshit

1) Import the libraries:



```
import pandas as pd
import numpy as np

import requests
from requests import get
from bs4 import BeautifulSoup

from time import sleep
```

2) Setup BeautifulSoup and load the webpage

3) Extract Data:

   Extract the name and the hyperlink to an individual movie, which can be found with the class name ‘lister-item’, then within this class, we can find the name and link of this movie.Put it in a for loop to run it for the whole page.

4) Scrape the reviews:

   1)To scrape the reviews, get the unique id of each movie and add it to a list

   2)By looping through the list, add the unique id to the base url and add(/reviews) to get the url of the reviews page and add it to another list

   For eg.

   Base_url:
   https://www.imdb.com/search/title/?title_type=feature&genres=comedy&start=

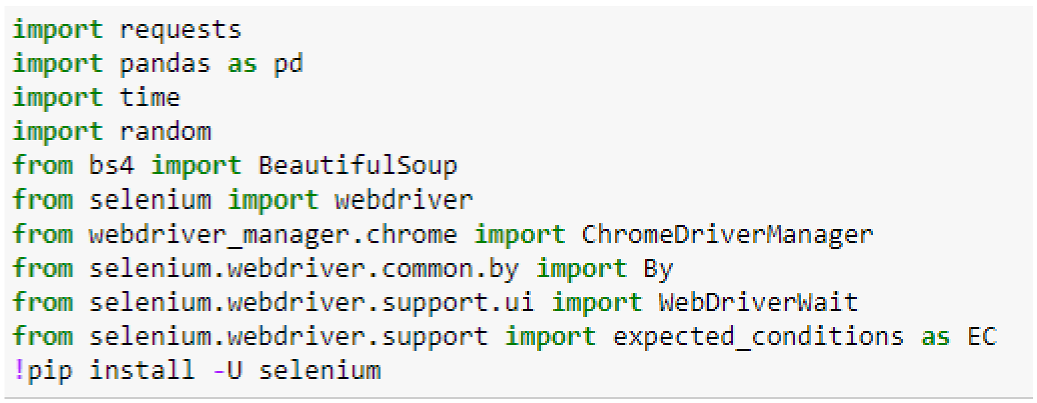
   Reviews page: base_url + unique_id + /reviews

   3) Loop through the list of urls and scrape the review only and continue to do so for the other movies.

5) Store the movie information:
After extracting the movie details, create multiple empty list and store the details in the list and put it in one dataframe.

Rotten Tomatoes - Jia Jun

1) Imported necessary libraries such as Selenium, webdriver, BeautifulSoup and pandas.



```
import requests
import pandas as pd
import time
import random
from bs4 import BeautifulSoup
from selenium import webdriver
from webdriver_manager.chrome import ChromeDriverManager
from selenium.webdriver.common.by import By
from selenium.webdriver.support.ui import WebDriverWait
from selenium.webdriver.support import expected_conditions as EC
!pip install -U selenium
```

2) Configure and launch Chrome web driver to the specific genres link.
Eg. https://www.rottentomatoes.com/browse/movies_at_home/genres:romance~sort:popular

3) Scraped movie links from Rotten Tomatoes website, then concat the scraped movie links to the end of the main website link to get the main movie link. We can also get to the movie's review link by adding /reviews to the movie's link.
Eg. Scraped movie links: /m/ticket_to_paradise_2022
Main website link: https://www.rottentomatoes.com
Main movie link: https://www.rottentomatoes.com/m/ticket_to_paradise_2022
Movie reviews link: https://www.rottentomatoes.com/m/ticket_to_paradise_2022/reviews

4) Loop through the list of movie links, navigate to each movie page and extract the year, genre and movie name.

5) Loop through the list of review links, navigate to each review page, extract the reviews and store them in a list.

6) Created a dictionary containing the extracted data (reviews, movie name, year, and genre)

7) Convert the dictionary to a dataframe and added to an existing dataframe

8) Repeat the process for each movie link

9) Close the browser

10) Save the final dataframe to a csv file

Metacritic - Skye

Import packages.

```
import pandas as pd
import numpy as np

import requests
from requests import get
from bs4 import BeautifulSoup

from time import sleep
from random import randint
```

Parse the page using BS

website:
https://www.metacritic.com/search/movie/results?genres[comedy]

Create a dictionary to hold the info that is being scraped.

Process:

Main movie link: https://www.metacritic.com/movie/(movie name)
Scrape Year, Genre,Ratings,Metascores and append it in the dictionary.
To scrape the year, we find the (“span” class = release_year lighter) and get the text
To scrape the genre we find the (class = genres) and get the text
To scrape the ratings we find the (class = metascore_w user larger movie mixed) and get the text
To scrape the metadata we find the (class = metascore_w larger movie positive) and get the text

Reviews page: https://www.metacritic.com/movie/movie_name/user-reviews

We use find the title of movies within the page by using the code below.

```
soup.find_all("div",class_="product_page_title oswald")
```

We use find the year of movies within the page by using the code below.

```
soup.find_all("div",class_="product_info lighter oswald upper pad_btm1").text
```

We use find all to get all reviews within the page by using the code below.

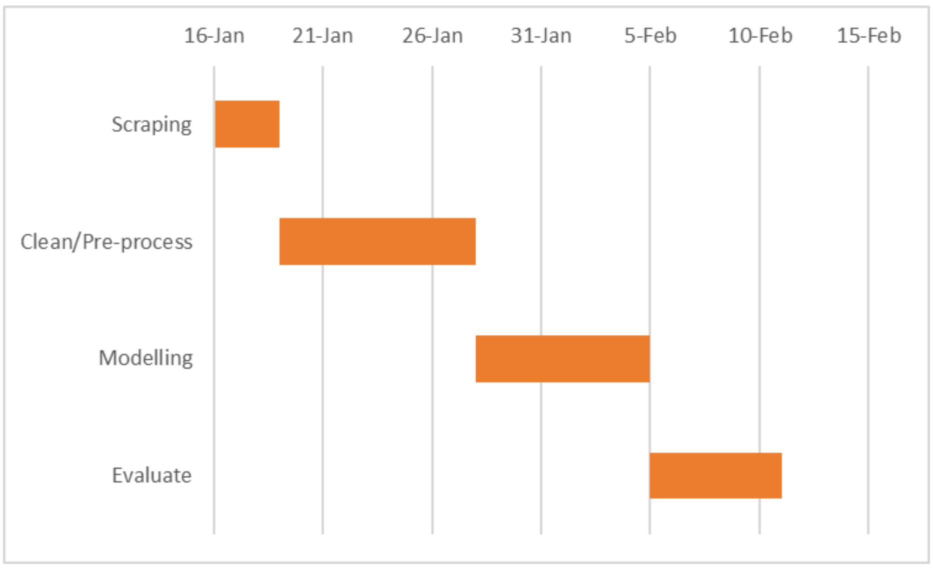
```
l("span",class_="blurb blurb_expanded")
```

We loop through all the reviews and scrape the reviews from the review page.

Once data has been collected, create a dictionary containing the data that has been collected and append data into it.

Lastly, Convert the dictionary into a dataframe.

Timeline


16-Jan
21-Jan
26-Jan
31-Jan
5-Feb
10-Feb
15-Feb
Scraping
Clean/Pre-process
Modelling
Evaluate