A Mini Project Report

On

“Web-Scraping IMDb Website”

Submitted in partial fulfilment requirements for the award of the Degree

BACHELOR OF ENGINEERING
IN

INFORMATION SCIENCE AND ENGINEERING

Submitted By

MR.HRITHIK M NAYAK (4NM20IS056)

MR.JEEVITH D R (4NM20IS057)

MR PRAMITH BHANDARY D (4NM20IS098)

Under the Guidance of

Dr. SUMATHI PAWAR

ASSOCIATE PROFESSOR
Department of Information Science and Engineering

Department of Information Science and Engineering
NMAM Institute of Technology, Nitte 2022– 2023

NITTE EDUCATION TRUST | NMAM INSTITUTE OF TECHNOLOGY

CERTIFICATE

This is to certify that **HRITHIK M NAYAK 4NM20IS056, JEEVITH D R 4NM20IS057** and **PRAMITH BHANDARY D 4NM20IS098**, bonafide students of **NMAM Institute of Technology, Nitte** has submitted the seminar report for the mini-project entitled **"Web Scraping IMDb Website"** in partial fulfillment of the requirements for the award of Bachelor of Engineering in Information Science and Engineering during the year 2022-23. It is verified that all corrections / suggestions indicated for internal assessment have been incorporated in the report deposited in the departmental library. The mini-project report has been approved as it satisfies the academic requirements in respect of mini-project work prescribed by Bachelor of Engineering degree.

Signature of the Guide	**Signature of the HOD**
Dr. Sumathi Pawar	Dr.Karthik Pai B

DECLARATION

I hereby declare that the entire work embodied in this Seminar report titled "**WEBSCRAPING IMDB WEBSITE**" has been carried out by us at NMAM Institute of Technology, Nitte under the supervision and Guidance of **Dr. Sumathi Pawar** for Bachelor of Engineering in Information Science and Engineering. This report has not been submitted to this or any other University for the award of any other degree.

Hrithik M Nayak	4NM20IS056
Jeevith D R	4NM20IS057
Pramith Bhandary D	4NM20IS098

Department of ISE
NMAMIT, Nitte

ABSTRACT

This project involves extracting movie data from the IMDb website using web scraping techniques. The goal of the project is to gather information on movies, such as their titles, release dates, genres, ratings, and cast members.

The extracted data will be cleaned and transformed into a usable format, and which later on can be used for analysis to identify trends and insights about popular movies and the film industry.The project will utilize various web scraping tools and technologies to efficiently and accurately extract data from IMDb's website.

The resulting dataset can be used for a range of purposes, including movie recommendations, market research, and academic studies.
Overall, this project aims to showcase the power and potential of web scraping for data analysis and insights in the entertainment industry.

LIST OF CONTENTS

Front Page	**i**
Certificate	**ii**
Declaration	**iii**
Abstract	**iv**
Chapters	**v**

CHAPTERS		**PageNo.**
1. INTRODUCTION		
1.1	Introduction to Project	**1**
1.2	Problem Statement	**1**
1.3	Scope	**2**
1.4	Objectives	**2**
1.5	Expected Outcomes	**2**
2. LITERATURE SURVEY		**3**
3. REQUIREMENT SPECIFICATION		
3.1	Project Requirements	**4**
3.2	Hardware Requirements	**4**
3.3	Software Requirements	**4**
4. DESIGN		
4.1	Block Diagram	**5-6**
4.2	Algorithm/ Flowchart	**6-7**
5. IMPLEMENTATION		
5.1	Module Implementations and API's used	**8-9**
6. RESULT		**10-11**
7. CONCLUSION AND FUTURE ENHANCEMENT		**12-13**
REFERENCES		**14**

CHAPTER 1

INTRODUCTION

1.1 Introduction to Project

Web scraping is a powerful technique for extracting data from websites, and it has numerous applications in fields such as data science, market research, and business intelligence. In this project, we will use web scraping to extract data from IMDb, the popular online database of movies, TV shows, and other entertainment content.

The goal of this project is to collect a comprehensive dataset of movies and TV shows from IMDb, along with their associated metadata such as ratings, genres, cast and crew, and plot summaries. We will use Python and the Beautiful Soup library to extract the data, and then store it in a structured format such as excel and also into a MYSQL server database.

1.2 Problem Statement

The Aim of this project is to utilize web scraping techniques to extract and transform movie data from the IMDb website into a structured and usable format thus creating a comprehens-ive and reliable dataset for analysis and insights.

Accessing and analysing data on movies available on IMDb's website can be challenging due to the lack of a public API or structured data source.

While IMDb provides extensive information on movies, it is not easily accessible in a struc-tured format that can be used for analysis or research. Web scraping provides a solution to this problem by allowing us to extract data from web pages and transform it into a usable format.

1.3 Scope

- Monitor ratings and reviews: IMDb can be scraped to monitor ratings and reviews of movies and TV shows. This can be useful for tracking the popularity of a particular movie or show, or for identifying trends in user preferences
- Analyze user behavior: IMDb can be scraped to analyze user behavior, such as the types of movies or TV shows that users are searching for or viewing, or the keywords that users are using to search for movies or TV shows.
- Build recommendation engines: You can scrape IMDb to build recommendation engines that suggest movies or TV shows based on user preferences, ratings, or other factors.

1.4 Objectives

The main objectives of this project are as follows:

- To collect accurate data on movies available on the IMDb website to create a comprehensive and usable dataset that can be used for various purposes, such as research, analysis,etc to be stored in an excel sheet and mysql database.
- To automate the data collection process.
- To create charts/visualisation displaying the distribution of movie ratings, the most popular genres, the top-rated actors, or the most profitable movies

1.5 Expected outcome

The expected outcomes of this project are as follows:

- A comprehensive and reliable dataset of movie information extracted from the IMDb website, including movie titles, release dates, ranks, ratings,etc stored in an excel sheet and a sql database
- An automated data collection process.
- Charts/visualisation displaying the distribution of movie ratings, the most popular genres, the top-rated actors, or the most profitable movie.

CHAPTER 2

Literature Survey

[1]"Web scraping for data acquisition and analysis in film studies" by A. Mirrlees and P. Altena (2017): This paper discusses the use of web scraping to collect data from IMDB for film studies research. The authors describe their methodology for collecting data on movies and actors, and the challenges they faced in the process.

[2]"Web Scraping and Data Analysis of IMDb Top 250 Movies" by J. Q. Chuah and T. N. Wong (2018): This paper describes a project to collect data from IMDb's top 250 movies and analyse the data to identify trends and patterns. The authors use Python and web scraping tools to collect data on movie ratings, genres, and other
attributes.

[3]"Web Scraping as a Tool for Social Science Research" by E. A. Roberts et al. (2020): This paper discusses the use of web scraping for social science research, including the collection of data from IMDB. The authors describe their methodology for collecting data on movie ratings and other attributes, and the ethical considerations involved in web scraping.

[4]"Web Scraping Using Python for Data Science:A Case Study with IMDb" by S. A. Ahmed (2021): This article provides a step-by-step guide to web scraping IMDb using Python. The author describes the tools and techniques involved in the process, and provides sample code for collecting data on movie ratings, reviews, and other attributes.

[5]"IMDb movie reviews: A dataset for sentiment analysis" by Maas et al. This paper introduces a dataset of 50,000 movie reviews from IMDB, along with sentiment polarity labels, which can be used for training and evaluating sentiment analysis models.

[6]"How to Scrape Data from IMDB using Python?" by Bhalla. This blog post provides a step-by-step tutorial for scraping movie data from IMDB using Python and the BeautifulSoup library.

CHAPTER 3
REQUIREMENT SPECIFICATION

3.1 Project Requirements

• Scrape data from IMDb's Top 250 Movies webpage.

• Extract key information from each movie, such as title, year of release, rating,etc

• Store the extracted data in a suitable format, such as excel.

• Comply with ethical and legal guidelines related to web scraping, including not infringing upon any copyright or other intellectual property rights.

• Ensure that the web scraping process is efficient and does not overload the server or cause any performance issues.

3.2 Hardware Requirements

• A computer with a stable internet connection.

• 4 GB RAM.

• Dual (or more) core processor with at least 2GHz clock speed.

3.3 Software Requirements

• Integrated development environment (IDE) - Jupyter Notebook

• Web scraping tool -Python's Beautiful Soup

• Request modules.

CHAPTER 4

DESIGN

4.1 Block Diagram

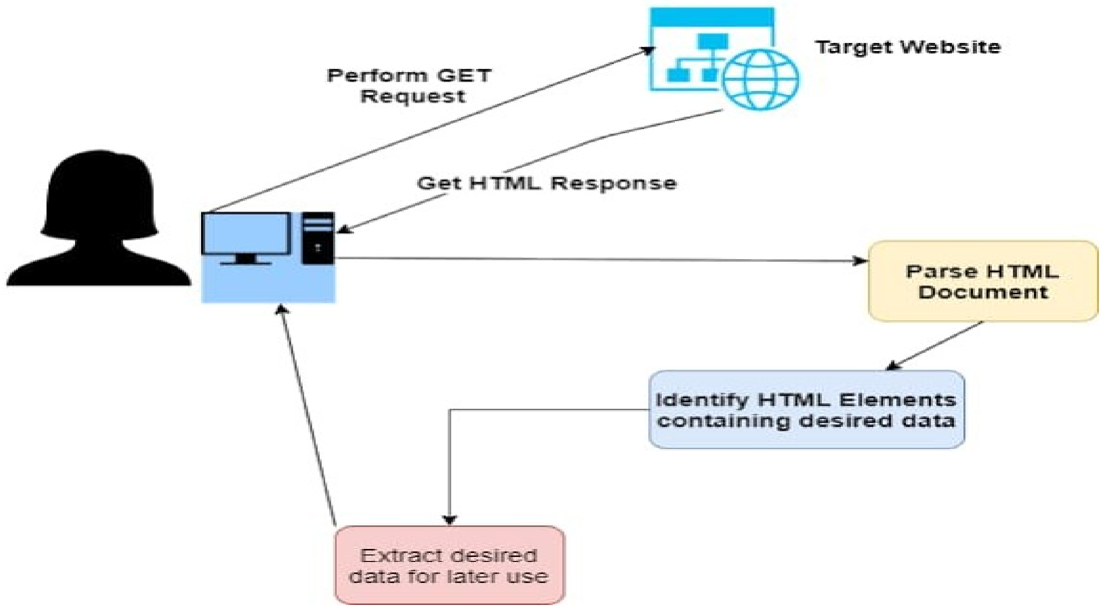

Target Website
Perform GET Request
Get HTML Response
Parse HTML Document
Identify HTML Elements containing desired data
Extract desired data for later use

Figure 4.1.1: Block diagram of the system

Web Request: The first component of web scraping is to send a request to the IMDb website using requests library. This request will retrieve the HTML code of the website, which can be used to extract the desired information.

HTML Parser: Once the HTML code of the website has been retrieved, it needs to be parsed to extract relevant information. Beautiful Soup is a Python library that provides a convenient way to parse HTML code and extract information from it.

Data Extraction: With the help of Beautiful Soup, desired data (movie title, genre, ratings, etc) can be extracted from the HTML code by using its various methods like find, find_all, select, etc. These method locate the specific HTML tags that contain the data to be extracted.

Data Storage : After extracting the desired data, ata storage options like excel and a database is used to store the extracted data.

Data Analysis: Once the data has been extracted, and stored, analysis can be performed on it using libraries like pandas and matplotlib.

4.2 Algorithm/ Flow Chart

Step 1.**Identify the web pages to scrape:**In this step, we need to determine which pages on the IMDB website you want to scrape.

Step 2.**Analyze the structure of the web pages**: Once you've identified the web pages to scrape, you need to analyze their structure to determine where the data you want to extract is located. This involves looking at the HTML code of the web pages and identifying the tags, attributes, and classes that contain the data you need.

Step 3.**Import necessary tools/modules:** installed and initialize the necessary libraries and tools.such as Requests and BeautifulSoup.

Step 4.**Send a request to the website**: Before you can scrape data from a website, you need to send a request to the website to retrieve the web pages you want to scrape. This can be done using the Requests library in Python.

Step 5.**Parse the HTML content**: Once you've received the HTML content of the web pages, you need to parse it to extract the data you need. This can be done using the BeautifulSoup library in Python.

Step 6.**Extract the data**: Using the information you gathered in step 2, you can extract the data you need from the parsed HTML content. This might involve finding specific tags, attributes, or classes that contain the data, or using regular expressions to extract patterns of text.

Step 7.**Store the data**: Once you've extracted the data, you'll need to store it in a format that you can use later. This might involve writing the data to a CSV or Excel file, storing it in a database, or using it directly in your Python code.

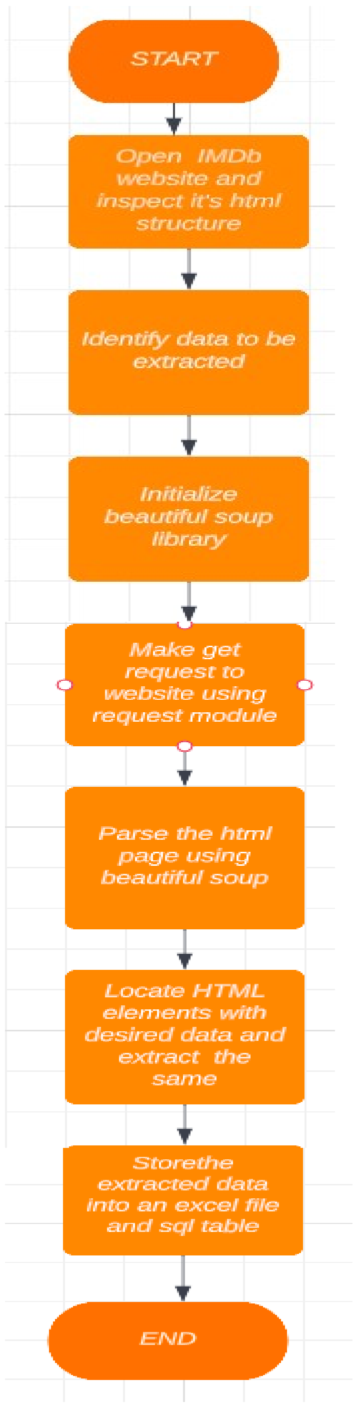

START
Open IMDb website and inspect it's html structure
Identify data to be extracted
Initialize beautiful soup library
Make get request to website using request module
Parse the html page using beautiful soup
Locate HTML elements with desired data and extract the same
Storethe extracted data into an excel file and sql table
END

Figure 4.2.1: Flowchart showing the web scraping process steps

CHAPTER 5

IMPLEMENTATION

5.1 Module implementations and API's used

requests: A Python library used for making HTTP requests to web pages or web APIs. In this code, it is used to send a GET request to the URL of the IMDB top rated movies page to retrieve the HTML content.

bs4 (BeautifulSoup): A Python library used for web scraping. It allows parsing of HTML and XML documents to extract information from web pages. In this code, it is used to extract the relevant information (movie title, rating, rank, and year of release) from the HTML content of the IMDB top rated movies page.

openpyxl: A Python library used for working with Excel files. In this code, it is used to create an Excel workbook and worksheet to store the scraped movie data.

sqlite3: A Python library used for working with SQLite databases. In this code, it is used to create a SQLite database file named movie_data1.db to store the scraped movie data.

tkinter: A Python library used for creating GUI applications. In this code, it is used to create a simple GUI window with a button to trigger the data scraping function and a label to display the status of the data scraping operation.

requests.get() : Sends a GET request to the specified URL and returns the server's response.

Input: URL

Output: Response Object

BeautifulSoup(): Creates a BeautifulSoup-object from the HTML content of the response obtained from the website.

Input:HTML content, default parser

Output:BeautifulSoup object

find() : Searches the HTML content for the specified tag and returns the first match

Input:tag name

Output: first matching tag

append(): Adds a row of data to the Excel worksheet.

Input:list of values to add to the Excel worksheet

Output:adds a row of data to the worksheet

c.execute(): Executes a SQL statement on the database cursor.

Input:SQL statement and a tuple of parameters

Output:Executes the statement on the database cursor.

Commit(): Saves the changes made to the database.

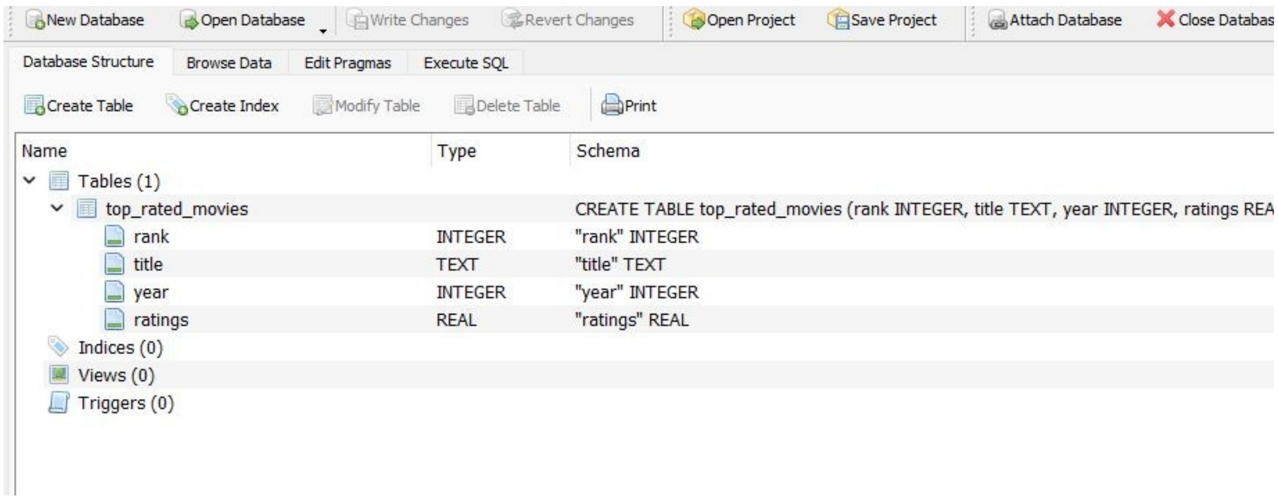
Input: No input

Output: saves the changes made to the database

CHAPTER 6

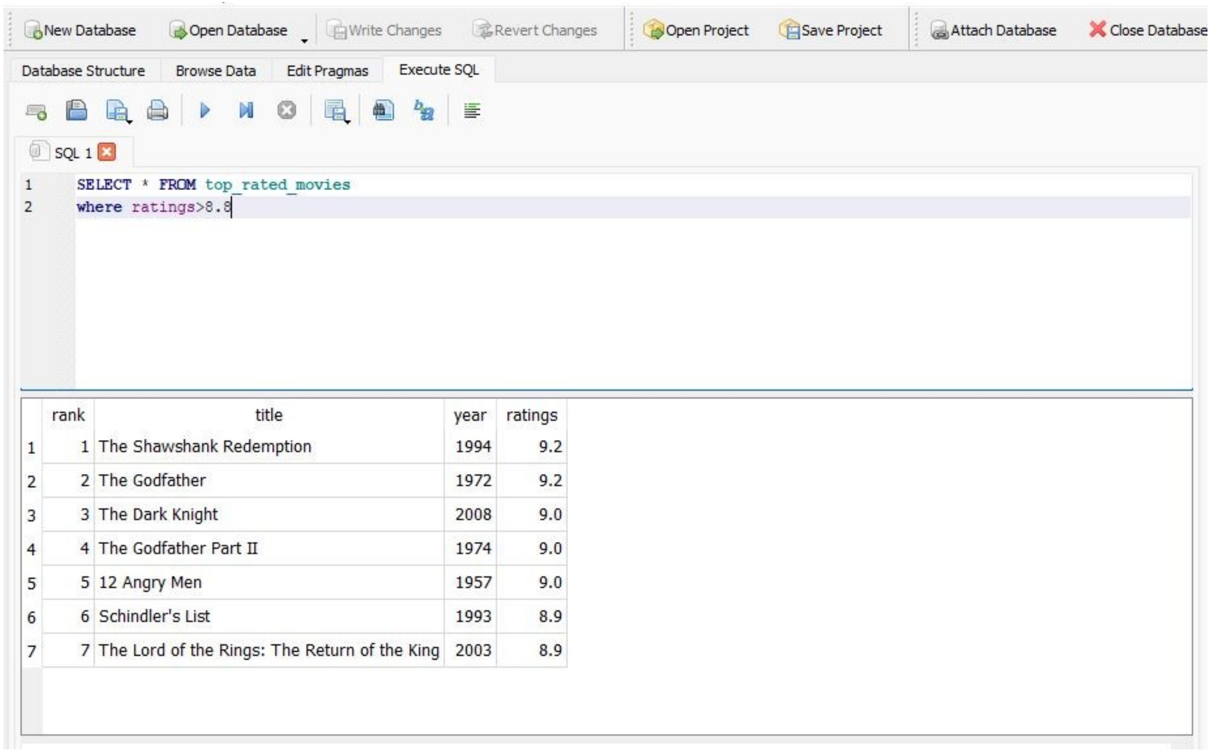
Result

6.1 SCREENSHOTS



Name	Type	Schema
Tables (1)		
top_rated_movies		CREATE TABLE top_rated_movies (rank INTEGER, title TEXT, year INTEGER, ratings REA
rank	INTEGER	"rank" INTEGER
title	TEXT	"title" TEXT
year	INTEGER	"year" INTEGER
ratings	REAL	"ratings" REAL
Indices (0)		
Views (0)		
Triggers (0)		

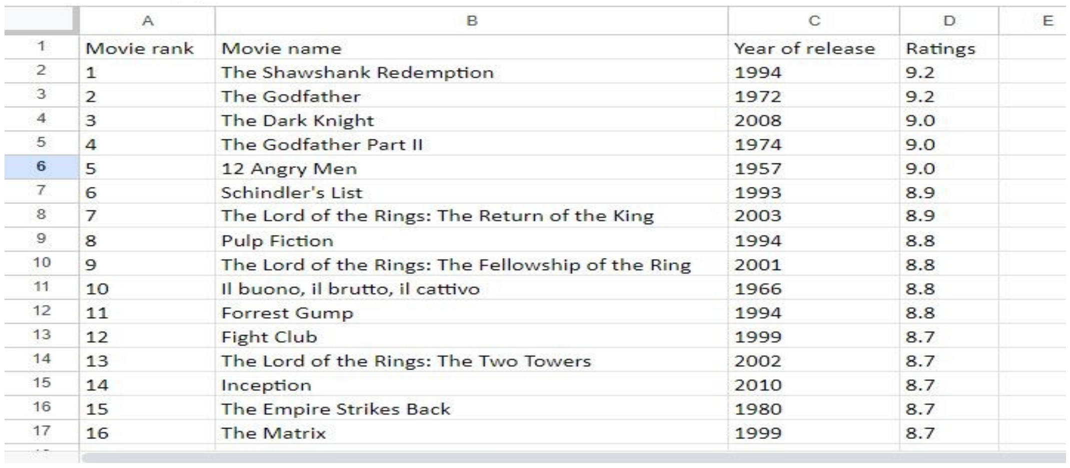
Figure 6.1.1 Extracted data stored on MYSQL server



```sql
SELECT * FROM top_rated_movies
where ratings>8.8
```

	rank	title	year	ratings
1	1	The Shawshank Redemption	1994	9.2
2	2	The Godfather	1972	9.2
3	3	The Dark Knight	2008	9.0
4	4	The Godfather Part II	1974	9.0
5	5	12 Angry Men	1957	9.0
6	6	Schindler's List	1993	8.9
7	7	The Lord of the Rings: The Return of the King	2003	8.9

Figure 6.1.2: Accessing the stored data via querying



	A	B	C	D	E
1	Movie rank	Movie name	Year of release	Ratings	
2	1	The Shawshank Redemption	1994	9.2	
3	2	The Godfather	1972	9.2	
4	3	The Dark Knight	2008	9.0	
5	4	The Godfather Part II	1974	9.0	
6	5	12 Angry Men	1957	9.0	
7	6	Schindler's List	1993	8.9	
8	7	The Lord of the Rings: The Return of the King	2003	8.9	
9	8	Pulp Fiction	1994	8.8	
10	9	The Lord of the Rings: The Fellowship of the Ring	2001	8.8	
11	10	Il buono, il brutto, il cattivo	1966	8.8	
12	11	Forrest Gump	1994	8.8	
13	12	Fight Club	1999	8.7	
14	13	The Lord of the Rings: The Two Towers	2002	8.7	
15	14	Inception	2010	8.7	
16	15	The Empire Strikes Back	1980	8.7	
17	16	The Matrix	1999	8.7	

Figure 6.1.3: Extracted data stored in excel format

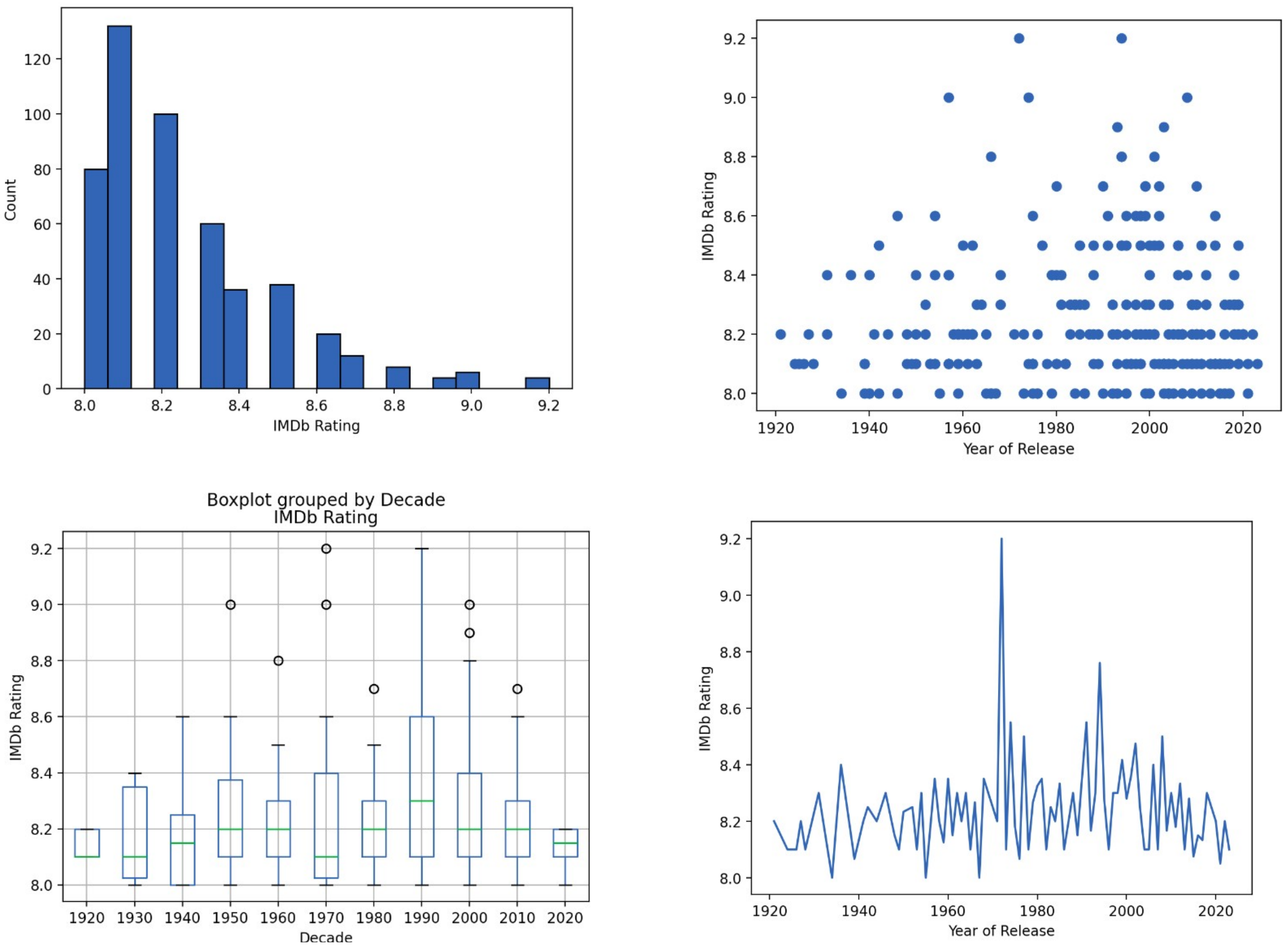


Figure 6.1.4: Visualisations

CHAPTER 7

CONCLUSION & FUTURE ENHANCEMENT

Web scraping allows for the automation of data extraction, which can save time and effort. Web scraping tools can be customized to extract specific types of data or information from IMDb. This makes it possible to extract only the data that is relevant to a particular project. Web scraping also allows for the extraction of real time data from IMDb (up-to-date data). There are some limitations too. Web scraping may infringe on the copyrights of the website owners. Before scraping data from any website, it's essential to check the terms and conditions of the website to avoid any legal issues. The data extracted from IMDb may not always be accurate or consistent. The data may contain errors, inconsistencies, or missing information. Also, IMDb may not allow access to all the data on the website through its APIs. Web scraping can put a significant strain on the website servers, leading to performance issues. Overall, the project demonstrates the process of web scraping and it's applications in the domain of movie and TV show data.

There are several potential enhancements that could be made to the web scraping project. Here are a few ideas:

1.Implementing error handling: Add code to handle errors that may occur during the scraping process, such as website downtime or changes in website structure.
2.Adding parallel processing: Implement parallel processing to scrape multiple pages at once, which can significantly increase the speed of the scraping process.
3.Using machine learning: Incorporate machine learning algorithms to help identify patterns in the data being scraped, such as sentiment analysis or topic modeling.
4.Integrating with APIs: Combine web scraping with API calls to obtain additional data or cross-reference data from multiple sources.
5.Creating a user interface: Develop a user interface that allows users to input search queries or select websites to scrape, and presents the data in a readable format.

REFERENCES

Websites Referred

- https://careerfoundry.com/en/blog/data-analytics/web-scraping-guide/
- https://www.crummy.com/software/BeautifulSoup/bs4/doc/
- https://docs.python-requests.org/en/latest/
- https://www.imdb.com/
- https://www.dataquest.io/blog/web-scraping-tutorial-python/
- https://www.scrapingbee.com/blog/web-scraping-with-python/