VIETNAM NATIONAL UNIVERSITY OF HO CHI MINH CITY
INTERNATIONAL UNIVERSITY

Data Science & Data Visualization

NETFLIX REPORT

Course by: Dr. Tran Thanh Tung
Team Member:

NAME	STUDENT ID
Phan Danh Đức	ITDSIU21012
Lê Nguyễn Thành Long	ITDSIU21097
Nguyễn Nguyên Hiệu	ITDSIU21087
Nguyễn Ngọc Sang	ITDSIU21114
Phạm Huỳnh Thanh Quân	ITDSIU21110

School of Computer Science and Engineering
International University, VNU-HCM
2024

Table of contents

I. INTRODUCTION

1. Abstract

❖ This project visualizes the evolution of Netflix's content over the years. It is divided into three scenes, each focusing on a unique aspect of the content that Netflix has produced over time.

❖ Scene Overview

- **Content Growth: Movies and TV Shows added over the years**

  This scene visualizes the growth in the quantity of content that Netflix has produced over time. It presents a timeline on the x-axis and the count of Movies and TV Shows on the y-axis. The user can explore the progression of Netflix's content production over the years and observe how their focus has shifted between Movies and TV shows.

- **Netflix's Top Content Producing Countries: A Closer Look**

  This scene shines a spotlight on the countries that produce the most content for Netflix. It offers a global distribution of Netflix's content production, with a particular emphasis on the United States and India. It showcases the significant growth in content production starting from 2008, reflecting Netflix's ambitious global expansion strategy.

- **Content Distribution by Age Rating: Shifting the Focus**

  This scene explores how Netflix's content distribution has evolved over time-based on age ratings. The user can discover the shifting focus in their content strategy and the significant increase in content production for mature audiences.

2. Purpose

❖ Provide a brief summary of the entire project, including its scope, objectives, and the main themes explored in each scene.

❖ The overview outlines the key aspects of Netflix's content evolution that will be visualized and analyzed throughout the project, giving the audience a clear understanding of the project's focus and significance.

3. Goal

- Process data to identify necessary attributes for visualization.
- Propose various charts and layouts for designing dashboards to assist users in gaining insights.
- Manage and adjust charts using D3.js libraries for better alignment with potential future alterations.
- Convey information through charts and effectively present critical insights from the dashboard.

4. Technique & Tools used

- IDE for web programming: Visual Studio Code - HTML/CSS/D3.js.
- Draft the UI of the dashboard: Figma
- Cleansing data: Python Jupyter Notebook by Google Colab.
- Code version management: Git, VSCode
- Member contacting measures: Microsoft Teams

II. TASK TIMELINE

1. Contribution

NAME	STUDENT ID	CONTRIBUTION
Phan Danh Đức	ITDSIU21012	20%
Lê Nguyễn Thành Long	ITDSIU21097	20%
Nguyễn Nguyên Hiệu	ITDSIU21087	20%
Nguyễn Ngọc Sang	ITDSIU21114	20%
Phạm Huỳnh Thanh Quân	ITDSIU21110	20%

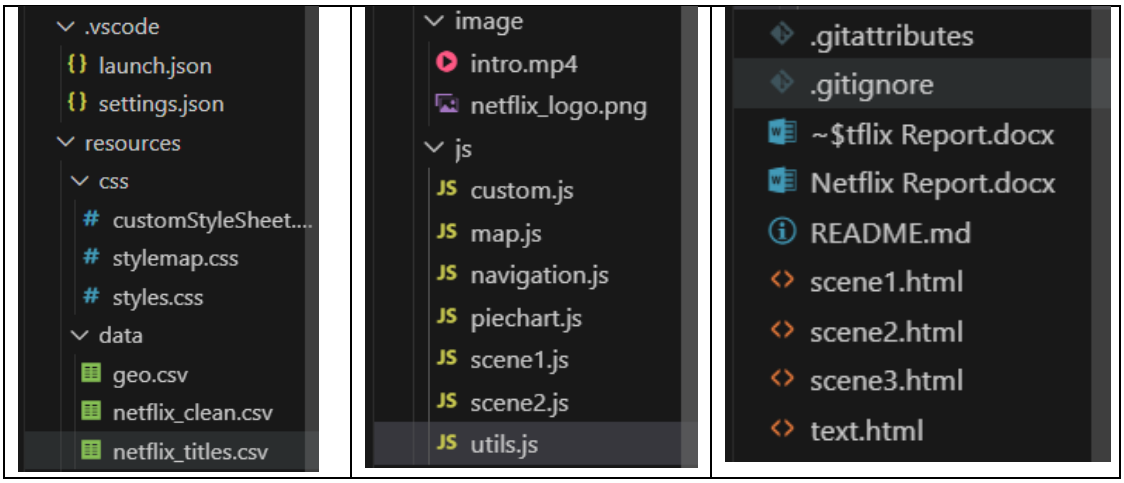
2. Timeline & Division

PHASE	ACTION	MEMBER	WEEK
PLANNING	Set up meeting	Đức	1
	Build the planning	Long	
	Individual topic research	All	
	Topic confirmation	All	
	Determine the goals and purpose of the analysis	All	
	Research for information	All	
DATA PULLING	Dataset collecting	Hiệu Quân	2
	References and documentation research	All	
	Determine the proper analysis technique for the project	All	
PREPROCESSING DATA	Remove some unnecessary attributes	Sang	3
	Examine values that affected the accuracy and precision of the findings	Quân Sang	
	Fill in missing data	Đức Long	
	Clean the data	Hiệu Quân	
CONCEPTUAL DESIGN	Design UI/UX	Hiệu	5-8
	Choose a theme and charts to visualize	Sang Quân	
	Sketch the diagrams and user flow	All	
	Sketch activities diagram	All	
IMPLEMENTING DESIGN	Structure the web layout	Đức Long	8-13
	Coding the web functions and charts	All	

	Implement storyline and add touch-up	Sang	
	Finalize the web coding	All	
PRESENTATION	Final report	All	14
	Presentation slides	All	

III. METHODDOLOGY

1. Project structure



- scene1.html, scene2.html, scene3.html, and text.html are the pages shown to users
- custom.js, map.js, navigation.js, piechart.js, scene1.js, scene2.js, and utils.js are the functions that draw charts
- geo.csv, netflix_clean.csv, and netflix_titles.csv are dataset used in the project
- customStyleSheet.css, stylemap.css, and styles.css create style for this project

2. Cleaning data

- The data we collect is still impure data, so we need to clean and apply aggregation on this dataset

- A part of the dataset file:
  (geo.csv)

```
identifier,latitude,longitude,country
AD,42.546245,1.601554,Andorra
AE,23.424076,53.847818,"United Arab Emirates"
AF,33.93911,67.709953,Afghanistan
AG,17.060816,-61.796428,"Antigua and Barbuda"
AI,18.220554,-63.068615,Anguilla
AL,41.153332,20.168331,Albania
AM,40.069099,45.038189,Armenia
AN,12.226079,-69.060087,"Netherlands Antilles"
AO,-11.202692,17.873887,Angola
AQ,-75.250973,-0.071389,Antarctica
AR,-38.416097,-63.616672,Argentina
AS,-14.270972,-170.132217,"American Samoa"
AT,47.516231,14.550072,Austria
AU,-25.274398,133.775136,Australia
AW,12.52111,-69.968338,Aruba
AZ,40.143105,47.576927,Azerbaijan
BA,43.915886,17.679076,"Bosnia and Herzegovina"
BB,13.193887,-59.543198,Barbados
BD,23.684994,90.356331,Bangladesh
```

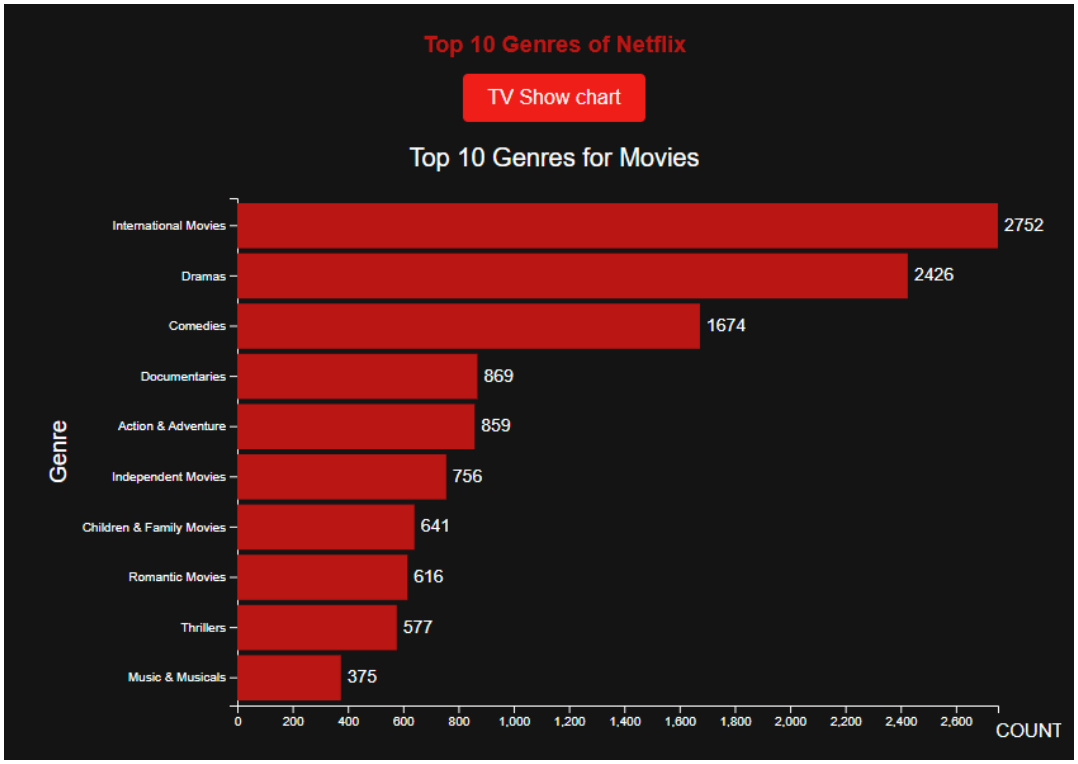
- The same with two files: (netflix_clean.csv), (netflix_titles.csv)

- After we have some basic information about the data, we start doing the cleaning data.

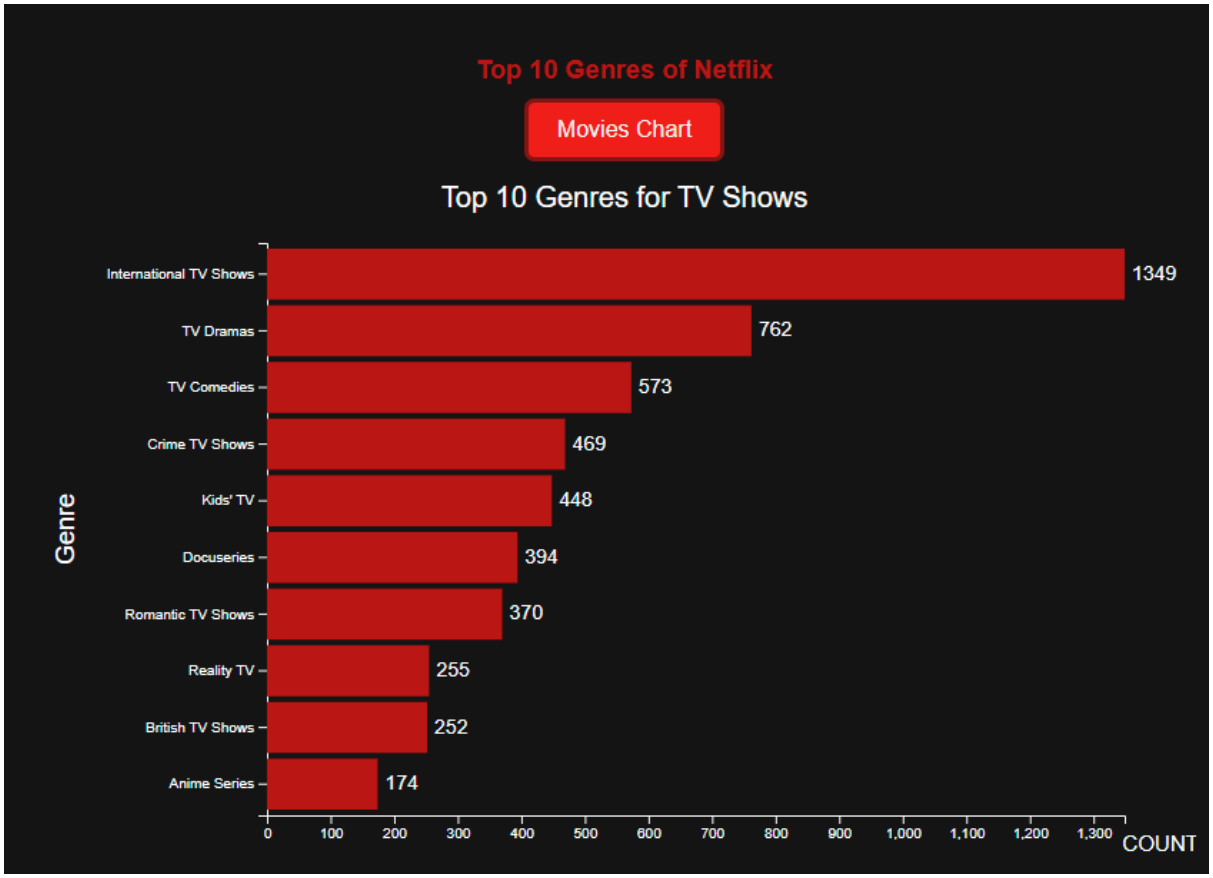
```
function processData(data, type) {
    const filteredData = data.filter(d => d.type === type);

    const categoryCount = {};
    filteredData.forEach(d => {
        d.listed_in.split(', ').forEach(category => {
            if (categoryCount[category]) {
                categoryCount[category]++;
            } else {
                categoryCount[category] = 1;
            }
        });
    });

    const sortedCategories = Object.keys(categoryCount).sort((a, b) => categoryCount[b] - categoryCount[a]);
    const topCategories = sortedCategories.slice(0, 10);

    return topCategories.map(category => ({ category, value: categoryCount[category] }));
}

d3.csv('resources/data/netflix_titles.csv').then(data => {
    drawScene2(data);
});
```

- The data we got will look like this:

- Movies Chart



- TV Show Chart



3. Chart usage

In this project, we use Pie Chart, Area Chart, Horizontal Bar Charts and Bubble Charts:

Chart name	Purpose	Channel	Mark	Description
Pie Chart	To compare the number of TV Shows and Movies viewers	Red and light red color, area of TV Shows and Movies	The slices of the pie, each representing a different category of data	Compare using requirements between TV Shows and Movies of users from Netflix
Area Chart	To show trends over time, compare magnitudes, and visualize cumulative totals	Position, length/height, and red/light red color	The line representing data points and the filled area between the line and the x-axis	The number of TV Shows and Movies added to a platform over a timeline from 2008 to 2021
Horizontal Bar Charts	View ranking of the minimum degree requirement	Vertical spatial position Horizontal lengths, red color, Rectangle shape	Lines	The horizontal bar chart illustrates how popular each film type in Netflix
Bubble Charts	To visualize and compare relationships between the number of users in each countries	Position (x and y coordinates), size, and red color	The bubbles, map	Displays a world map with various red circles of different size of TV Shows and Movies in each countries

4. Implementation with D3.js

Reviewing d3.js documentation and discussion:

Pie Chart:

```
// Chọn phần tử SVG và thiết lập thuộc tính chiều rộng và chiều cao
const svgPie = d3.select("#piechart").html("")
  .append("svg")
  .attr("width", widthPie+500)
  .attr("height", heightPie + 300)
  .append("g")
  .attr("transform", "translate(" + 250 + "," + (350) + ")");

  svgPie.append("text")
  .attr("x", 0)
  .attr("y", -300) // Điều chỉnh vị trí của tiêu đề
  .attr("text-anchor", "middle")
  .style("font-size", "25px")
  .style("font-weight", "bold")
  .attr("fill", "white")
  .text("Total Movies & TV Shows on Netflix");


// Dữ liệu Movies và TV shows
d3.csv("resources/data/netflix_titles.csv")
  .then(function (data) {
    const totalMovies = data.filter(d => d.type === "Movie").length;
    const totalTVShows = data.filter(d => d.type === "TV Show").length;
    const total = totalMovies + totalTVShows;

    const moviesPercentage = (totalMovies / total) * 100;
    const tvShowsPercentage = (totalTVShows / total) * 100;

    const pieData = [
      { category: "Movies", percent: moviesPercentage },
      { category: "TV Shows", percent: tvShowsPercentage }
    ];

    draw_pie(pieData);
  })
  .catch(function (error) {
    console.log(error);
  });
```

Area Chart:

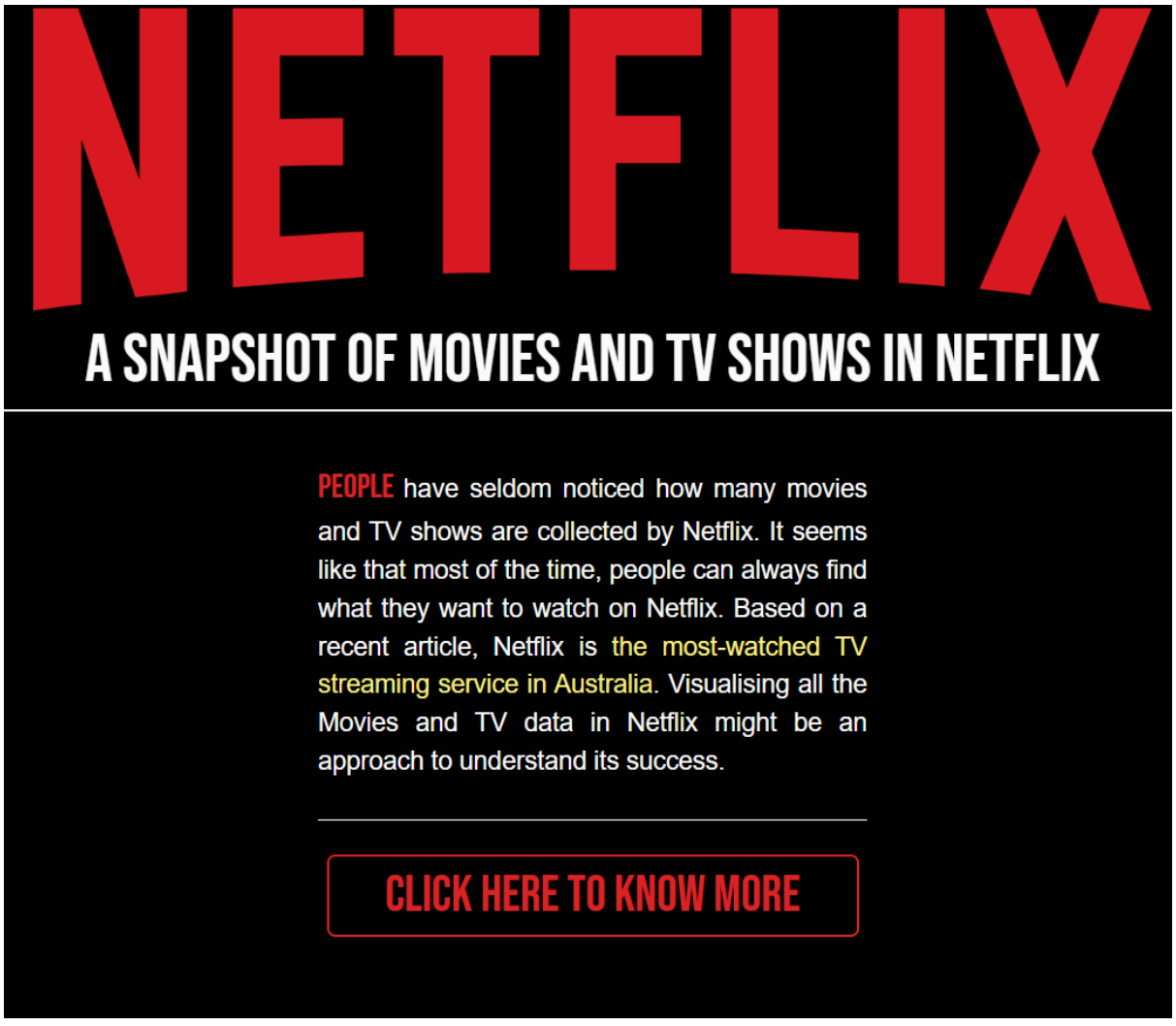
```
function drawScene1(data) {
    const svg = d3.select("#chart").html("");
    const width = 1500, height = 700, margin = {top: 50, right: 0, bottom: 50, left: 125};

    const parseTime = d3.timeParse("%Y-%m-%d");
    const x = d3.scaleTime().range([0, width - margin.left - margin.right]);
    const y = d3.scaleLinear().range([height - margin.top - margin.bottom, 0]);

    const area = d3.area()
        .x(d => x(d.date))
        .y0(y(0))
        .y1(y(0));

    const g = svg.append("g")
        .attr("transform", `translate(${margin.left}, ${margin.top})`);

    data.forEach(d => {
        d.date = parseTime(d.date_added.split(" ")[0]);
        d.value = +d.show_id;
    });

    const sumData = d3.rollups(data, v => v.length, d => d.type, d => d.date.getFullYear());

    const newData = [];
    sumData.forEach(d => {
        d[1].forEach(e => {
            newData.push({date: new Date(e[0], 0, 1), type: d[0], value: e[1]});
        });
    });

    newData.sort((a, b) => a.date - b.date);

    x.domain(d3.extent(newData, d => d.date));
    y.domain([0, d3.max(newData, d => d.value)]);

    g.selectAll(".area")
        .data(d3.groups(newData, d => d.type))
        .join("path")
        .attr("class", "area")
        .attr("fill", d => d[0] === "Movie" ? "#b20710" : "#f75151") // màu hình
        .attr("stroke", "black")
        .attr("stroke-width", 3)
        .attr("opacity", 0.8)
        .attr("d", d => area(d[1]))
        .transition()
        .duration(2000)
        .attr("d", d => {
```

Horizontal Bar Charts:

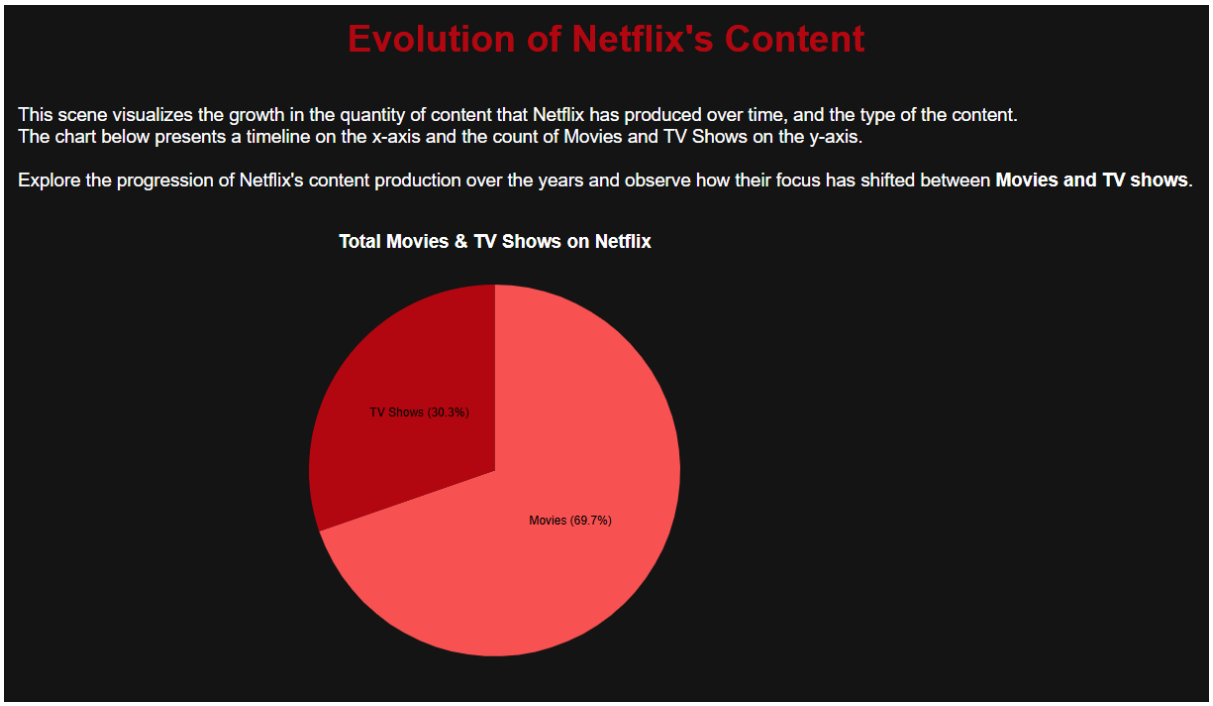
```
function drawChart(container, data, title) {
    const svg = d3.select(container).append("svg")
        .attr("width", 800)
        .attr("height", 500);

    const width = 800, height = 500;
    const margin = { top: 50, right: 50, bottom: 50, left: 150 };  // Increase left margin to 150

    const categories = data.map(d => d.category);
    const maxValue = d3.max(data, d => d.value);

    const yScale = d3.scaleBand()
        .domain(categories)
        .range([margin.top, height - margin.bottom])
        .padding(0.1);

    const xScale = d3.scaleLinear()
        .domain([0, maxValue])
        .range([margin.left, width - margin.right]);

    const xAxis = d3.axisBottom(xScale);
    const yAxis = d3.axisLeft(yScale);

    svg.append("g")
        .attr("class", "x-axis")
        .attr("transform", `translate(0, ${height - margin.bottom})`)
        .call(xAxis);

    svg.append("g")
        .attr("class", "y-axis")
        .attr("transform", `translate(${margin.left}, 0)`)
        .call(yAxis);

    const tooltip = d3.select("body").append("div")
        .attr("class", "tooltip")
        .style("position", "absolute")
        .style("background-color", "white")
        .style("padding", "5px")
        .style("border", "1px solid black")
        .style("border-radius", "5px")
        .style("visibility", "hidden")
        .text("");
```

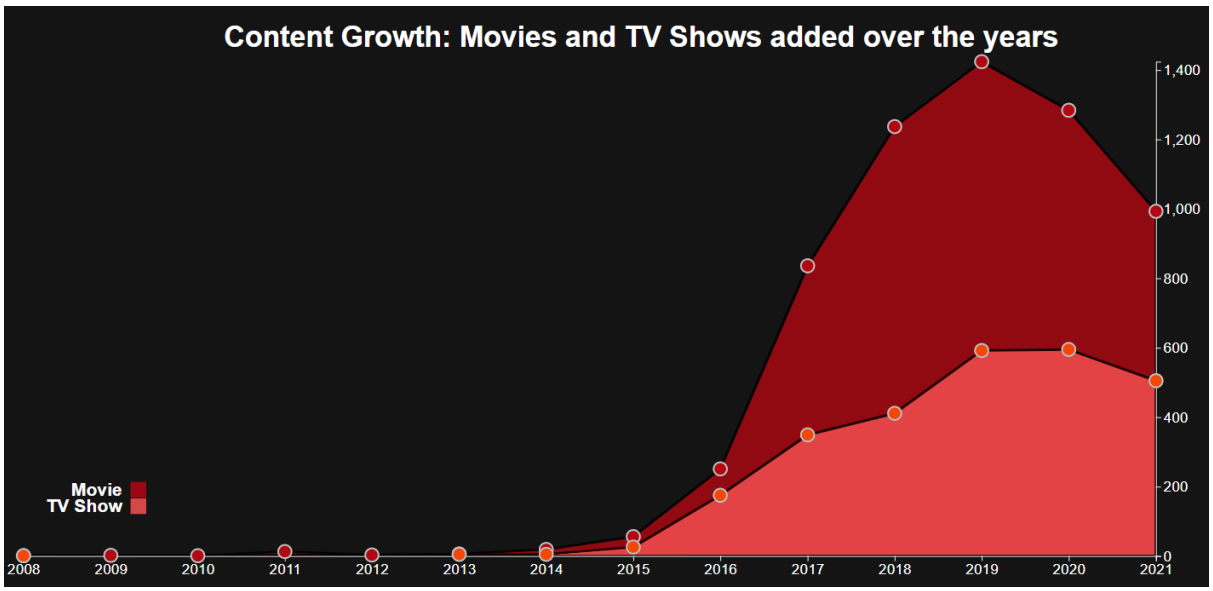
IV. RESULT

At the first page, we keep it simple with a background image, an introduction, and press "**CLICK HERE TO KNOW MORE**" to go to the next page.

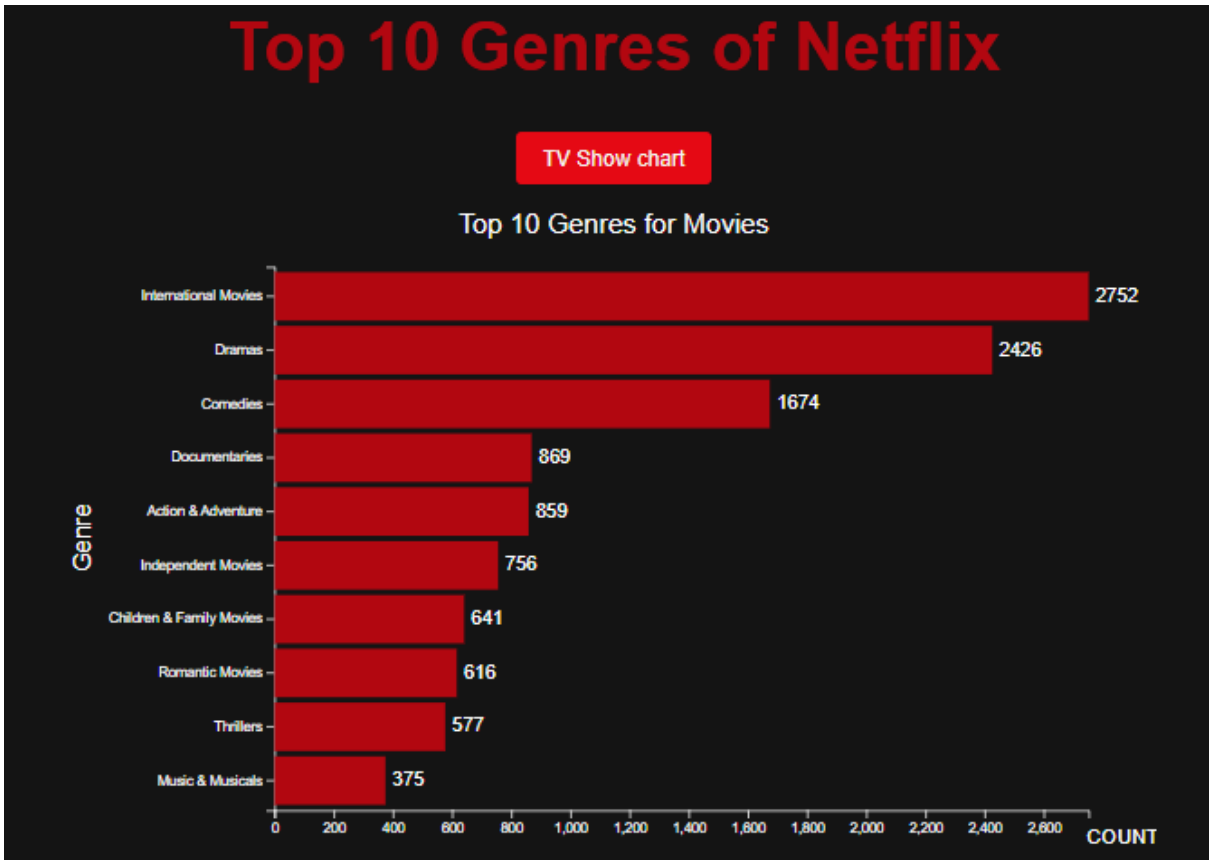


When user presses this, leading to the next page is Pie Chart and Area Chart for information of TV Shows and Movies.

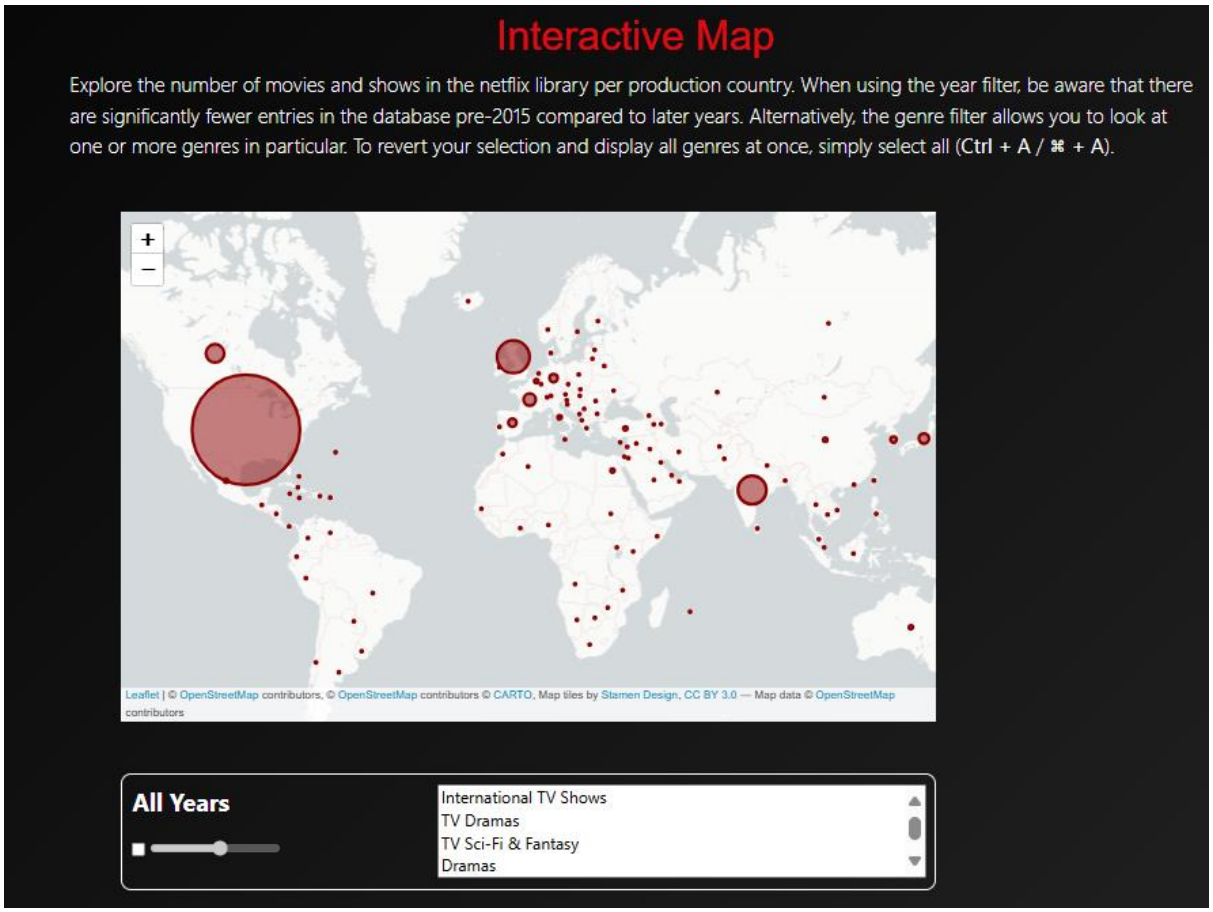




On the second page, the users can refer to the number of viewers for several different types of films on Netflix with Horizontal Bar Charts.



When choosing "**Scene 3**" button, this is the map presentation viewers watch TV Shows and Movies in each countries.



V. Conclusion:

1. Achieved goals

The main goal of this project is to provide 4 charts, comprehensive the number of viewers of TV Shows and Movies on Netflix that provide inside from raw data. We present a degree that demonstrates the practical application of ideas learned in theory classes using charts and interactive elements, allowing users to view complex data in an understandable and natural way. We will deliver a discriplined project managemennt approach and coodebase to make thiis happen, ensuring effective development and maintainability of the dashboard. To improve the usability and brevity of dashboards, we will also evaluate how well the dashboard can handle new features.

In addition, by meeting these objectives, we hope to provide users with a powerful and revolutionary dashboard that allows them to explore the depths of difficult-to-access insights concealed inside the raw data. Users will be able to

dynamically change and examine the data on the dashboard thanks to its interactive nature, allowing them to draw insightful inferences and extract valuable knowledge. With these insights at their disposal, users can better develop sound strategies, make educated decisions, and promote success in their fields.

2. Future work:

Future work has several intriguing opportunities to improve the dashboard's usability and functionality even more. Enhancing the user interface by including concise and detailed text descriptions alongside the visualizations. This feature would enable a deeper data analysis and offer users detailed insights. Drill-down tools, interactive filters, and sorting choices would enable users to Explore the data in a more relevant and personalized manner. This could entail investigating various chart styles, improving visualization strategies, and ensuring the selected charts appropriately represent the data and convey insights. The dashboard would also be able to manage more user queries and deliver the most recent data possible if real-time data updates were included. Data pipelines and other procedures would need to be implemented to fetch and handle real-time data seamlessly. By focusing on these potential future development areas, the dashboard can become a more potent and dynamic tool, giving users an even richer experience and a more insightful understanding of the data.

3. Concluding:

We want to thank the lecturer for helping and supporting us. Their guidance during lab sessions has been very helpful. We're grateful for the clear and well-documented resources that have helped us learn and grow. We also want to congratulate the team members for working well together and achieving great results through effective teamwork. Our collaborative effort has resulted in creating a dashboard that's easy to use and helps people explore data in a better way. We're excited to keep moving forward and learn more.

VI. References

1. Kaggle: Your Machine Learning and Data Science Community
2. Resources/css/stylemap.css
3. https://cdnjs.cloudflare.com/ajax/libs/twitter-bootstrap/4.3.1/css/bootstrap.min.css
4. https://stackpath.bootstrapcdn.com/font-awesome/4.7.0/css/font-awesome.min.css
5. https://unpkg.com/leaflet@1.4.0/dist/leaflet.css