

**ĐẠI HỌC BÁCH KHOA HÀ NỘI**  
**TRƯỜNG CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG**

\*\*\*\*\*



**BÁO CÁO**

**Nhập môn khoa học dữ liệu**

**Nhóm 6 - IT4930 - 147708**

**ĐỀ TÀI**

**Phân tích và dự đoán doanh thu của Video Game dựa trên dữ liệu  
thu thập được từ trang VGchartz**

**Sinh viên thực hiện**

<b>Họ tên</b>	<b>MSSV</b>	
1. Lê Minh Vũ	20205050	
2. Hoàng Văn Kiên	20205089	
3. Trần Minh Quân	20205015	
4. Lê Trường Giang	20205077	
5. Nguyễn Chí Thành	20205127	
6. Hòa Đức Việt	20205046	Nhóm Trưởng

*Hà Nội, tháng 5 năm 2024*

**Link Github:** [Github.com/ohayotmq/DataScience\\_GameRevenuePrediction](https://github.com/ohayotmq/DataScience_GameRevenuePrediction)

### **Phân chia công việc các thành viên trong nhóm**

<b>Thành viên</b>	<b>MSSV</b>	<b>Đóng góp</b>	<b>Đánh giá</b>
Lê Trường Giang	20205077	Xử lý dữ liệu, Frontend	100%
Trần Minh Quân	20205015	Xử lý dữ liệu, Frontend	100%
Hoàng Văn Kiên	20205089	Backend, các Model cơ bản, ghép API	100%
Lê Minh Vũ	20205050	Tìm hiểu ứng dụng, đánh giá hiệu suất các mô hình	100%
Nguyễn Chí Thành	20205127	Tìm hiểu ứng dụng các mô hình, Backend	100%
Hòa Đức Việt	20205046	Huấn luyện và tối ưu hoá các mô hình	100%

## MỤC LỤC

<b>CHƯƠNG 1: GIỚI THIỆU ĐỀ TÀI.....</b>	<b>5</b>
<b>CHƯƠNG 2: THU THẬP DỮ LIỆU.....</b>	<b>7</b>
2.1 Chọn nguồn thu thập dữ liệu.....	7
2.2 Quá trình thu thập dữ liệu.....	7
<b>CHƯƠNG 3: TIỀN XỬ LÝ DỮ LIỆU.....</b>	<b>11</b>
3.1 Tìm hiểu dữ liệu.....	11
3.2 Tiền xử lý dữ liệu.....	11
1. Giảm chiều dữ liệu.....	11
2. Điền trường thông tin bị thiếu.....	11
3. Điều chỉnh bộ dữ liệu & Chuẩn hóa dữ liệu.....	13
<b>CHƯƠNG 4: PHÂN TÍCH DỮ LIỆU.....</b>	<b>15</b>
4.1 Trực quan hóa dữ liệu.....	15
4.2 Tổng kết.....	19
<b>CHƯƠNG 5: MÔ HÌNH VÀ PHƯƠNG PHÁP HUẤN LUYỆN.....</b>	<b>20</b>
5.1 Phương pháp huấn luyện.....	20
5.2 Các mô hình lựa chọn.....	21
5.2.1 Hồi quy tuyến tính đa biến (Multiple Linear Regression).....	21
5.2.2 Hồi quy đa thức (Polynomial Regression).....	21
5.2.3 Hồi quy K-Nearest Neighbors (KNN Regression).....	21
5.2.4 Hồi quy cây quyết định (Decision Tree Regression).....	22
5.2.5 Hồi quy rừng ngẫu nhiên (Random Forest Regression).....	22
5.2.6 Hồi quy Vector hỗ trợ tuyến tính (Linear SVR).....	22
5.2.7 Hồi quy Vector hỗ trợ phi tuyến (Non-linear SVR).....	23
5.2.8 Hồi quy XGBoost (XGBoost Regression).....	23
5.2.9 Hồi quy OLS:.....	23
*Tổng kết:.....	24
5.3 Tối ưu hóa mô hình.....	24
<b>CHƯƠNG 6: Triển khai thành sản phẩm.....</b>	<b>30</b>
6.1 Sơ đồ use case.....	30
6.2 Flowchart.....	30
6.3 Các màn hình chức năng.....	31
6.3.1 Màn hình trang chủ.....	31
6.3.2 Màn hình dự đoán game theo doanh số.....	31

6.3.3 Màn hình dự đoán game theo đặc trưng game.....	32
6.3.4 Màn hình dự đoán khu vực/hệ máy phát hành tốt nhất.....	32
<b>CHƯƠNG 7: Kết luận.....</b>	<b>33</b>
7.1 Kết quả thực nghiệm.....	33
7.2 Các khó khăn và hướng phát triển trong tương lai.....	39
7.2.1 Khó Khăn.....	39
7.2.1 Hướng Phát Triển:.....	40
7.3 Kết luận.....	40

# CHƯƠNG 1: GIỚI THIỆU ĐỀ TÀI

## 1.1 Giới thiệu

Trong cuộc sống hiện đại, video game đã trở thành một phần không thể thiếu, không chỉ là một hình thức giải trí mà còn là một ngành công nghiệp khổng lồ với doanh thu hàng tỷ đô la mỗi năm. Ngành công nghiệp game phát triển mạnh mẽ, mang lại lợi nhuận lớn và tạo ra hàng triệu công việc trên toàn cầu.

## 1.2 Vấn đề đặt ra

Tuy nhiên, việc thiết kế và phát triển một trò chơi điện tử không hề đơn giản. Các nhà phát triển phải đầu tư rất nhiều thời gian và tiền bạc vào việc tạo ra sản phẩm. Khi hoàn thiện, việc phát hành game cũng đòi hỏi chi phí quảng bá đáng kể tại các khu vực phát hành. Nếu trò chơi được phát hành không đạt doanh thu cao như mong đợi, doanh nghiệp sẽ phải đối mặt với những tổn thất lớn.

Để giảm thiểu rủi ro này, các doanh nghiệp trong ngành công nghiệp game luôn mong muốn tìm ra cách dự đoán chính xác nhất sự phát triển và phát hành của một trò chơi để đạt kết quả tốt nhất.

## 1.3 Giải pháp đã có

Hiện nay, một trong những phương pháp phổ biến để dự đoán doanh thu của game là thông qua hoạt động khảo sát thị trường. Các nhà phân tích kinh doanh (Business Analysts - BA) thực hiện các khảo sát, bao gồm việc phỏng vấn người dùng, phân tích xu hướng thị trường, và nhiều hoạt động khác. Tuy nhiên, giải pháp này tiêu tốn nhiều công sức và nguồn lực, đồng thời kết quả có thể không hoàn toàn chính xác.

## 1.4 Mục tiêu của bài tập lớn này

Nhằm khắc phục những hạn chế trên, bài tập lớn này sẽ tập trung vào việc áp dụng các thuật toán học máy để dự đoán doanh thu của video game dựa trên dữ liệu thu thập từ trang VGchartz. Bằng cách sử dụng dữ liệu lớn và các mô hình học máy, chúng em hy vọng sẽ giúp doanh nghiệp có thể lựa chọn phương hướng thiết kế, phát triển và phát hành game hiệu quả hơn, giảm thiểu rủi ro và tối đa hóa doanh thu.

Thông qua việc phân tích dữ liệu lịch sử và các yếu tố ảnh hưởng đến doanh thu, chúng em sẽ xây dựng các mô hình dự đoán đáng tin cậy, giúp các nhà phát triển game đưa ra quyết định sáng suốt hơn trong quá trình tạo ra và đưa game ra thị trường. Hơn nữa, chúng em cũng sẽ sử dụng các mô hình đó để phát triển

sản phẩm web app cung cấp giao diện và chức năng dễ dàng tiếp cận và sử dụng cho mọi người dùng.

Các mô hình sẽ được mô tả chi tiết trong chương 5

Các chức năng của sản phẩm web app sẽ được mô tả chi tiết trong chương 6

## CHƯƠNG 2: THU THẬP DỮ LIỆU

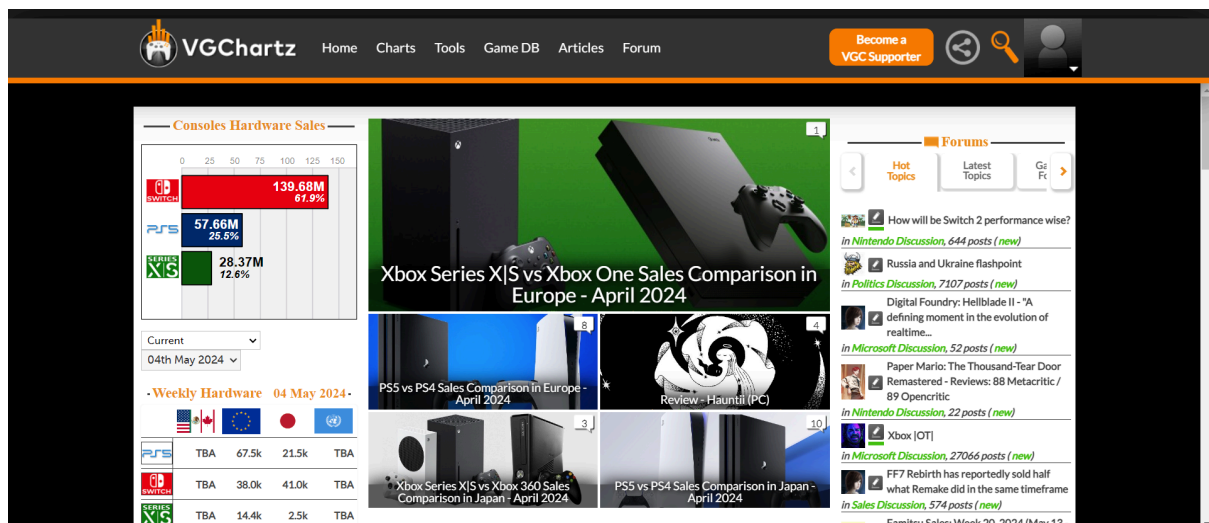
### 2.1 Chọn nguồn thu thập dữ liệu

Nguồn dữ liệu được nhóm crawl về từ trang web Vgchartz.com

Trang web này chuyên cung cấp dữ liệu về các tựa game hiện tại, bao gồm nhà phát triển, hệ máy, ngày phát hành, doanh thu và đánh giá của người dùng với từng tựa game

Lý do nhóm lựa chọn trang web này vì trang Web có đầy đủ thông tin doanh thu của các tựa game, cập nhật đều đặn dữ liệu game mỗi tuần, không cần trả phí hay bị giới hạn truy cập như những trang web khác. Giao diện trang web trực quan, dễ nhìn. Và quan trọng hơn cả là thuận tiện cho việc thu thập và xử lý dữ liệu

Hình ảnh về trang web:



### 2.2 Quá trình thu thập dữ liệu

Sau khi xem chi tiết về cấu trúc của trang web, nhóm quyết định sử dụng thư viện scrapy của ngôn ngữ Python để thực hiện crawl dữ liệu

Nhóm sử dụng BeautifulSoup, một framework mạnh mẽ trong công việc thu thập và trích xuất dữ liệu từ các trang web

# BeautifulSoup

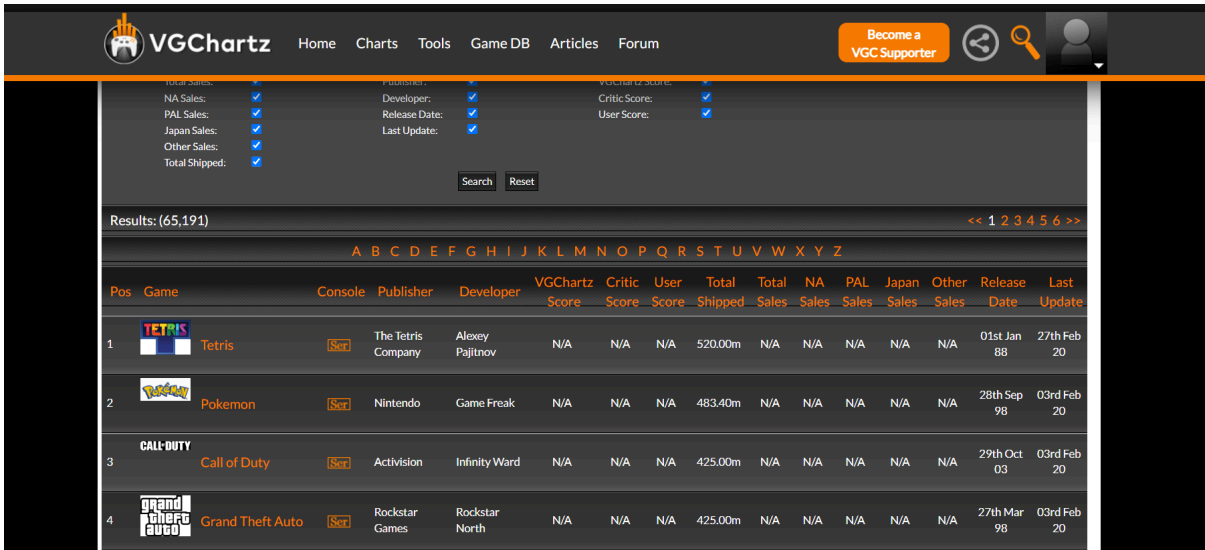
Một số tính năng:

- Beautiful Soup là một thư viện Python nổi bật với những đặc điểm sau:
- Dễ sử dụng: BeautifulSoup cung cấp một cú pháp đơn giản và dễ hiểu để trích xuất dữ liệu từ các tài liệu HTML và XML. Điều này làm cho việc học và sử dụng thư viện này trở nên dễ dàng ngay cả với những người mới bắt đầu.
- Xử lý linh hoạt: BeautifulSoup có thể xử lý các tài liệu HTML không chuẩn hoặc bị lỗi cú pháp, giúp người dùng trích xuất dữ liệu mà không gặp khó khăn với các tài liệu có cấu trúc không đồng nhất.
- Tích hợp tốt với các parser: BeautifulSoup có thể làm việc với nhiều trình phân tích cú pháp (parser) khác nhau như lxml và html.parser, giúp tăng cường hiệu suất và khả năng xử lý các tài liệu phức tạp.
- Tìm kiếm và điều hướng hiệu quả: Thư viện này cung cấp các phương thức mạnh mẽ để tìm kiếm và điều hướng qua cây DOM của tài liệu HTML, chẳng hạn như tìm kiếm theo thẻ, class, id, hoặc các thuộc tính khác.
- Chuyển đổi linh hoạt giữa các định dạng: BeautifulSoup hỗ trợ việc chuyển đổi dễ dàng giữa các định dạng khác nhau, giúp người dùng xuất dữ liệu theo cách mình muốn, từ HTML đến XML hoặc plain text.

Trên đây là một vài tính năng của công cụ Crawl dữ liệu, tiếp theo nhóm sẽ trình bày chi tiết về quá trình thu thập dữ liệu của mình



Trước tiên để có thể trích xuất dữ liệu thời tiết ta cần biết url nào của trang web có data mà mình cần lấy. Đó chính là trang [Vgchartz GameDB](#)



Pos	Game	Console	Publisher	Developer	VGChartz Score	Critic Score	User Score	Total Shipped	Total Sales	NA Sales	PAL Sales	Japan Sales	Other Sales	Release Date	Last Update
1	Tetris	[Scr]	The Tetris Company	Alexey Pajitnov	N/A	N/A	N/A	520.00m	N/A	N/A	N/A	N/A	N/A	01st Jan 88	27th Feb 20
2	Pokemon	[Scr]	Nintendo	Game Freak	N/A	N/A	N/A	483.40m	N/A	N/A	N/A	N/A	N/A	28th Sep 98	03rd Feb 20
3	Call of Duty	[Scr]	Activision	Infinity Ward	N/A	N/A	N/A	425.00m	N/A	N/A	N/A	N/A	N/A	29th Oct 03	03rd Feb 20
4	Grand Theft Auto	[Scr]	Rockstar Games	Rockstar North	N/A	N/A	N/A	425.00m	N/A	N/A	N/A	N/A	N/A	27th Mar 98	03rd Feb 20

Dữ liệu của trang đã được chia sẵn dưới dạng table nên cần tìm thẻ html<table> của trang để chọn nguồn dữ liệu.

Lược bỏ những cột header không cần thiết

Tiến hành lấy dữ liệu từng cột theo index. Ví dụ như Tên nằm ở cột 4 được trích xuất theo thẻ<tablerow>

```
[11]: #We make a List of games starting from row 4 onward
game_rows = table.find_all('tr')[3:]
game_rows

[11]: [<tr style="background-image:url(../imgs/chartBar_large.gif); height:70px">
<td>1</td>
<td>
<div id="photo3">
<a href="/games/game.php?id=226182&region=All">
<div style="height:60px; width:60px; overflow:hidden;"> 
</div>
</a>
</div>
</td> <td style="font-size:12pt;"> <a href="https://www.vgchartz.com/game/226182/wii-fit/?region=All">Wii Fit </a> </td>
<td align="center">

</td> <td width="100">Nintendo </td> <td width="100">Nintendo </td> <td align="center">43.80m</td> <td align="center">N/A</td> <td align="center">N/A</td> <td align="center">N/A</td> <td align="center">N/A</td> <td align="center">21st May 08 </td></tr>
<tr style="background-image:url(../imgs/chartBar_alt_large.gif); height:70px">
<td>2</td>
<td>
</td>
</tr>]
```

Dữ liệu trả về sẽ có dạng html và được sort lại để hiển thị được dưới dạng mảng tương ứng.

Tương tự với các trường dữ liệu còn lại

Với những trường dữ liệu không lấy trực tiếp từ trang thống kê ( Ví dụ như thể loại game), nhóm sẽ lặp lại quá trình trích xuất dữ liệu trên một số trang được lọc theo một thể loại cụ thể. Bằng cách đó, chúng tôi có thể lấy được thể loại của từng trò chơi mà không cần phải trích xuất nó từ trang riêng của từng trò chơi, nhằm tiết kiệm thời gian đáng kể.

Quá trình trích xuất tóm gọn trong các bước như sau:

1. Tìm table chứa thông tin các tựa game
2. Sử dụng css selector để lấy toàn bộ row của bảng này
3. Sử dụng BeautifulSoup để trích xuất các thông tin theo từng cột trong bảng theo từng yêu cầu
4. Lưu lại dữ liệu trích xuất được vào file

Kết quả trích xuất:

	Video Game Title	Genre	Console	Publishers(s)	Developer(s)	Units Sold	Total Sales	NA Sales	PAL Sales	JP Sales	Other Sales	Release Date
0	The Walking Dead: Telltale Games Series	Adventure	Series	Telltale Games	Telltale Games	50.00m	N/A	N/A	N/A	N/A	N/A	24th Apr 12
1	Professor Layton	Adventure	Series	Nintendo	Level-5	18.00m	N/A	N/A	N/A	N/A	N/A	10th Feb 08
2	Myst	Adventure	Series	Broderbund	Cyan, Inc.	12.50m	N/A	N/A	N/A	N/A	N/A	24th Sep 93
3	Ace Attorney	Adventure	Series	Capcom	Capcom	11.00m	N/A	N/A	N/A	N/A	N/A	12th Oct 05
4	Broken Sword	Adventure	Series	Virgin Interactive	Revolution Software	10.00m	N/A	N/A	N/A	N/A	N/A	30th Sep 96
...	...	...	...	...	...	...	...	...	...	...	...	...
64963	XBlaze Lost: Memories	Visual+Novel	PC	Aksys Games	Arc System Works	N/A	N/A	N/A	N/A	N/A	N/A	11th Aug 16
64964	Yoru, Tomosu	Visual+Novel	NS	Nippon Ichi Software	Nippon Ichi Software	N/A	N/A	N/A	N/A	N/A	N/A	30th Jul 20
64965	Yoru, Tomosu	Visual+Novel	PS4	Nippon Ichi Software	Nippon Ichi Software	N/A	N/A	N/A	N/A	N/A	N/A	30th Jul 20
64966	Yunohana SpRING! ~Mellow Times~	Visual+Novel	NS	Idea Factory	Otomate	N/A	N/A	N/A	N/A	N/A	N/A	28th Feb 19
64967	Yurukill: The Calumniation	Visual+Novel	PS4	Unknown	G.rev Ltd.	N/A	N/A	N/A	N/A	N/A	N/A	N/A

## CHƯƠNG 3: TIỀN XỬ LÝ DỮ LIỆU

### 3.1 Tìm hiểu dữ liệu

Dữ liệu gồm 1 file, 12 cột chứa dữ liệu của các tựa game

Ý nghĩa mỗi cột:

- Video Game Title: Tên tựa game
- Genre: Thể loại
- Console: Hệ điều hành
- Publisher(s): Nhà phát hành
- Developer(s): Nhà phát triển
- Unit Sold: Số lượng bản game đã bán
- Total Sales: Doanh thu tổng
- NA Sales: Doanh thu Bắc Mỹ
- PAL Sales: Doanh thu Châu Âu
- JP Sales: Doanh thu Nhật Bản
- Others Sales: Doanh thu khu vực khác
- Release Date: Thời điểm phát hành

### 3.2 Tiền xử lý dữ liệu

#### 1. Giảm chiều dữ liệu

Trước tiên hành phân tích dữ liệu, cần giảm số chiều dữ liệu của các cột category xuống.

Bỏ cột Developer(s), bởi vì 1 tựa game sẽ bao gồm quá nhiều các nhà phát triển, khiến việc dự báo trở nên khó khăn

Bỏ cột Unit Sold vì tỷ lệ giá trị null quá cao (90%), sẽ giảm hiệu quả cho mô hình

```
df = df.drop(['Developer(s)'], axis = 1)
df
```

#### 2. Điền trường thông tin bị thiếu

Khi kiểm tra dữ liệu không phát hiện dữ liệu trống, nhưng một số có giá trị bằng 0 hoặc NaN.

Với cột Title, Genre, Console, Publisher, sẽ xóa những game thiếu thông tin vì sẽ ảnh hưởng đến dự đoán của mô hình.

Với cột Total Result, những game có giá trị là 0 hoặc NaN sẽ bị loại bỏ

```
df = df[df['Total Sales'] != 0]
df
```

	Video Game Title	Genre	Console	Publishers(s)	Total Sales	NA Sales	PAL Sales	JP Sales	Other Sales	Release Date
15	Tomb Raider II	Adventure	PS	Eidos Interactive	5.24m	2.30m	2.46m	0.20m	0.28m	31st Oct 97
31	LEGO Indiana Jones: The Original Adventures	Adventure	X360	LucasArts	3.76m	2.40m	1.01m	NaN	0.36m	03rd Jun 08
33	Tomb Raider III: Adventures of Lara Croft	Adventure	PS	Eidos Interactive	3.54m	1.66m	1.58m	0.12m	0.18m	21st Nov 98
34	LEGO Batman: The Videogame	Adventure	X360	Warner Bros. Interactive	3.44m	2.07m	1.04m	NaN	0.34m	23rd Sep 08
36	L.A. Noire	Adventure	PS3	Rockstar Games	3.21m	1.29m	1.31m	0.12m	0.49m	15th Nov 11
...	...	...	...	...	...	...	...	...	...	...
64649	Nora, Princess, and Stray Cat	Visual+Novel	NS	Harukaze	0.00m	NaN	NaN	0.00m	NaN	25th Oct 18
64650	Memories Off: Innocent File	Visual+Novel	NS	5pb	0.00m	NaN	NaN	0.00m	NaN	25th Oct 18
64651	Enkan no Memoria: Kakeru Tomoshi	Visual+Novel	PSV	Dramatic Create	0.00m	NaN	NaN	0.00m	NaN	29th Mar 18

```
df = df[df['Total Sales'] != '0.00m']
df
```

	Video Game Title	Genre	Console	Publishers(s)	Total Sales	NA Sales	PAL Sales	JP Sales	Other Sales	Release Date
15	Tomb Raider II	Adventure	PS	Eidos Interactive	5.24m	2.30m	2.46m	0.20m	0.28m	31st Oct 97
31	LEGO Indiana Jones: The Original Adventures	Adventure	X360	LucasArts	3.76m	2.40m	1.01m	NaN	0.36m	03rd Jun 08
33	Tomb Raider III: Adventures of Lara Croft	Adventure	PS	Eidos Interactive	3.54m	1.66m	1.58m	0.12m	0.18m	21st Nov 98
34	LEGO Batman: The Videogame	Adventure	X360	Warner Bros. Interactive	3.44m	2.07m	1.04m	NaN	0.34m	23rd Sep 08
36	L.A. Noire	Adventure	PS3	Rockstar Games	3.21m	1.29m	1.31m	0.12m	0.49m	15th Nov 11
...	...	...	...	...	...	...	...	...	...	...
64628	Amatsutsumi	Visual+Novel	PSV	Prototype	0.01m	NaN	NaN	0.01m	NaN	17th May 18
64629	Dance with Devils: My Carol	Visual+Novel	PSV	Rejet	0.01m	NaN	NaN	0.01m	NaN	22nd Mar 18

Cột Release Date cũng sẽ xóa các trường hợp không đủ định dạng ngày/tháng hoặc NaN

Với cột NA Sales, EU Sales hoặc JP Sales, do có những tựa game chỉ phát hành ở một số khu vực nhất định, hoặc không có thông tin về các khu vực khác, nên chúng tôi giữ các trường hợp giá trị = 0 và chuyển các cột NaN thành 0 nếu cả 3 doanh thu 3 khu vực đều không NaN.

...	...	...	...	...	...	...	...	...	...	...
64628	Amatsutsumi	Visual+Novel	PSV	Prototype	0.01m	NaN	NaN	0.01m	NaN	17th May 18
64629	Dance with Devils: My Carol	Visual+Novel	PSV	Rejet	0.01m	NaN	NaN	0.01m	NaN	22nd Mar 18
64630	Memories Off: Innocent File	Visual+Novel	PSV	MAGES	0.01m	NaN	NaN	0.01m	NaN	29th Mar 18
64631	World End Syndrome	Visual+Novel	NS	Arc System Works	0.01m	NaN	NaN	0.01m	NaN	02nd May 19

Tổng kết:

- Ở các cột TotalSales hay Date, giá trị trống/ bằng 0 sẽ bị xóa
- Ở các cột Title, Genre, Console, Publisher, thiếu giá trị/trống sẽ bị xóa
- Ở các cột Doanh thu khu vực, giá trị trống/ bằng 0 sẽ được quy định là 0

### 3. Điều chỉnh bộ dữ liệu & Chuẩn hóa dữ liệu

Nhiều trường dữ liệu cần chuẩn hóa kiểu để có thể tính toán trong tương lai.

Các cột doanh số được điều chỉnh về dạng float:

```
df_task4['Total Sales'] = df_task4['Total Sales'].str.replace('m','').astype(float)
df_task4['NA Sales'] = df_task4['NA Sales'].str.replace('m','').astype(float)
df_task4['PAL Sales'] = df_task4['PAL Sales'].str.replace('m','').astype(float)
df_task4['JP Sales'] = df_task4['JP Sales'].str.replace('m','').astype(float)
df_task4['Other Sales'] = df_task4['Other Sales'].str.replace('m','').astype(float)
df_task4
```

	Video Game Title	Genre	Console	Publishers(s)	Total Sales	NA Sales	PAL Sales	JP Sales	Other Sales	Release Date
0	Tomb Raider II	Adventure	PS	Eidos Interactive	5.24	2.30	2.46	0.20	0.28	31st Oct 97
1	LEGO Indiana Jones: The Original Adventures	Adventure	X360	LucasArts	3.76	2.40	1.01	0.00	0.36	03rd Jun 08
2	Tomb Raider III: Adventures of Lara Croft	Adventure	PS	Eidos Interactive	3.54	1.66	1.58	0.12	0.18	21st Nov 98
3	LEGO Batman: The Videogame	Adventure	X360	Warner Bros. Interactive	3.44	2.07	1.04	0.00	0.34	23rd Sep 08
4	L.A. Noire	Adventure	PS3	Rockstar Games	3.21	1.29	1.31	0.12	0.49	15th Nov 11
...	...	...	...	...	...	...	...	...	...	...
17512	Amatsutsumi	Visual+Novel	PSV	Prototype	0.01	0.00	0.00	0.01	0.00	17th May 18
17513	Dance with Devils: My Carol	Visual+Novel	PSV	Rejet	0.01	0.00	0.00	0.01	0.00	22nd Mar 18
17514	Memories Off: Innocent File	Visual+Novel	PSV	MAGES	0.01	0.00	0.00	0.01	0.00	29th Mar 18

Cột thời gian được chuyển về dạng date

```
[255]: #Before converting to datetime, we stripped 'rd', 'st', 'nd', etc from the day of the week
df_task4['Release Date'] = df_task4['Release Date'].str[:2] + df_task4['Release Date'].str[4:]
df_task4['Release Date'] = pd.to_datetime(df_task4['Release Date'], format = '%d %b %y')
df_task4
```

3	LEGO Batman: The Videogame	Adventure	X360	Warner Bros. Interactive	3.44	2.07	1.04	0.00	0.34	2008-09-23
4	L.A. Noire	Adventure	PS3	Rockstar Games	3.21	1.29	1.31	0.12	0.49	2011-11-15
...	...	...	...	...	...	...	...	...	...	...
17512	Amatsutsumi	Visual+Novel	PSV	Prototype	0.01	0.00	0.00	0.01	0.00	2018-05-17
17513	Dance with Devils: My Carol	Visual+Novel	PSV	Rejet	0.01	0.00	0.00	0.01	0.00	2018-03-22
17514	Memories Off: Innocent File	Visual+Novel	PSV	MAGES	0.01	0.00	0.00	0.01	0.00	2018-03-29
17515	World End Syndrome	Visual+Novel	NS	Arc System Works	0.01	0.00	0.00	0.01	0.00	2019-05-02
17516	Sweet Pool	Visual+Novel	PSV	Dramatic Create	0.01	0.00	0.00	0.01	0.00	2018-05-31

Chia data thành cột Ngày, tháng, năm, sau đó xóa cột ngày. Lý do là vì ngày không ảnh hưởng đến tính toán (Không tính những ngày lễ đặc biệt), trong khi tháng, năm sẽ quyết định được thời điểm trong năm phát hành ảnh hưởng đến doanh thu như thế nào

```
[255]: #Before converting to datetime, we stripped 'rd', 'st', 'nd', etc from the day of the week
df_task4['Release Date'] = df_task4['Release Date'].str[:2] + df_task4['Release Date'].str[4:]
df_task4['Release Date'] = pd.to_datetime(df_task4['Release Date'], format = '%d %b %y')
df_task4
```

3	LEGO Batman: The Videogame	Adventure	X360	Warner Bros. Interactive	3.44	2.07	1.04	0.00	0.34	2008-09-23
4	L.A. Noire	Adventure	PS3	Rockstar Games	3.21	1.29	1.31	0.12	0.49	2011-11-15
...	...	...	...	...	...	...	...	...	...	...
17512	Amatsutsumi	Visual+Novel	PSV	Prototype	0.01	0.00	0.00	0.01	0.00	2018-05-17
17513	Dance with Devils: My Carol	Visual+Novel	PSV	Rejet	0.01	0.00	0.00	0.01	0.00	2018-03-22
17514	Memories Off: Innocent File	Visual+Novel	PSV	MAGES	0.01	0.00	0.00	0.01	0.00	2018-03-29
17515	World End Syndrome	Visual+Novel	NS	Arc System Works	0.01	0.00	0.00	0.01	0.00	2019-05-02
17516	Sweet Pool	Visual+Novel	PSV	Dramatic Create	0.01	0.00	0.00	0.01	0.00	2018-05-31

17517 rows x 10 columns

```
57]: #First months
list_months = []
for i in range(len(df_task5)):
    list_months.append(df_task5['Release Date'][i].month)
list_months
```

```
df_task5['Release Month'] = list_months
df_task5['Release Year'] = list_years
df_task5
```

```
df_task5 = df_task5.drop(['Release Date'], axis = 1)
df_task5
```

	Video Game Title	Genre	Console	Publishers(s)	Total Sales	NA Sales	PAL Sales	JP Sales	Other Sales	Release Month	Release Year
0	Tomb Raider II	Adventure	PS	Eidos Interactive	5.24	2.30	2.46	0.20	0.28	10	1997
1	LEGO Indiana Jones: The Original Adventures	Adventure	X360	LucasArts	3.76	2.40	1.01	0.00	0.36	6	2008
2	Tomb Raider III: Adventures of Lara Croft	Adventure	PS	Eidos Interactive	3.54	1.66	1.58	0.12	0.18	11	1998
3	LEGO Batman: The Videogame	Adventure	X360	Warner Bros. Interactive	3.44	2.07	1.04	0.00	0.34	9	2008
4	L.A. Noire	Adventure	PS3	Rockstar Games	3.21	1.29	1.31	0.12	0.49	11	2011
...	...	...	...	...	...	...	...	...	...	...	...
17512	Amatsutsumi	Visual+Novel	PSV	Prototype	0.01	0.00	0.00	0.01	0.00	5	2018
17513	Dance with Devils: My Carol	Visual+Novel	PSV	Rejet	0.01	0.00	0.00	0.01	0.00	3	2018
17514	Memories Off: Innocent File	Visual+Novel	PSV	MAGES	0.01	0.00	0.00	0.01	0.00	3	2018
17515	World End Syndrome	Visual+Novel	NS	Arc System Works	0.01	0.00	0.00	0.01	0.00	5	2019

Cuối cùng điều chỉnh bộ dữ liệu bằng cách sửa lại tên cột, điều chỉnh cột (Những cột như Genre có 2 thể loại trở lên) lại để hoàn tất tiền xử lý:

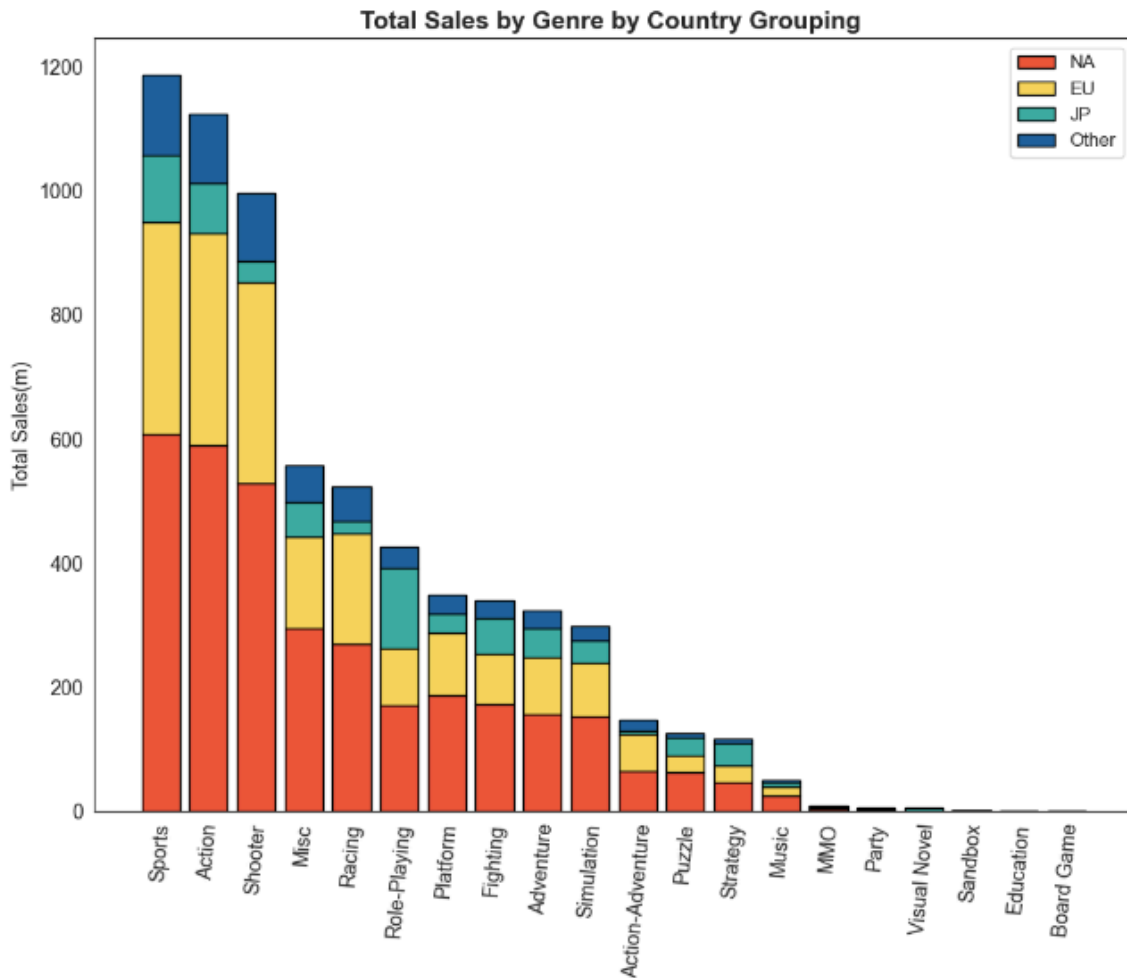
```
3]: df_task5.columns = ['Title', 'Genre', 'Console', 'Publisher', 'Total Sales (m)', 'NA Sales (m)', 'EU Sales (m)', 'JP Sales (m)', 'Other Sales (m)', 'Release Month', 'Release Year']
df_task5
```

```
df_task5['Genre'] = df_task5['Genre'].str.replace('+', ' ')
df_task5
```

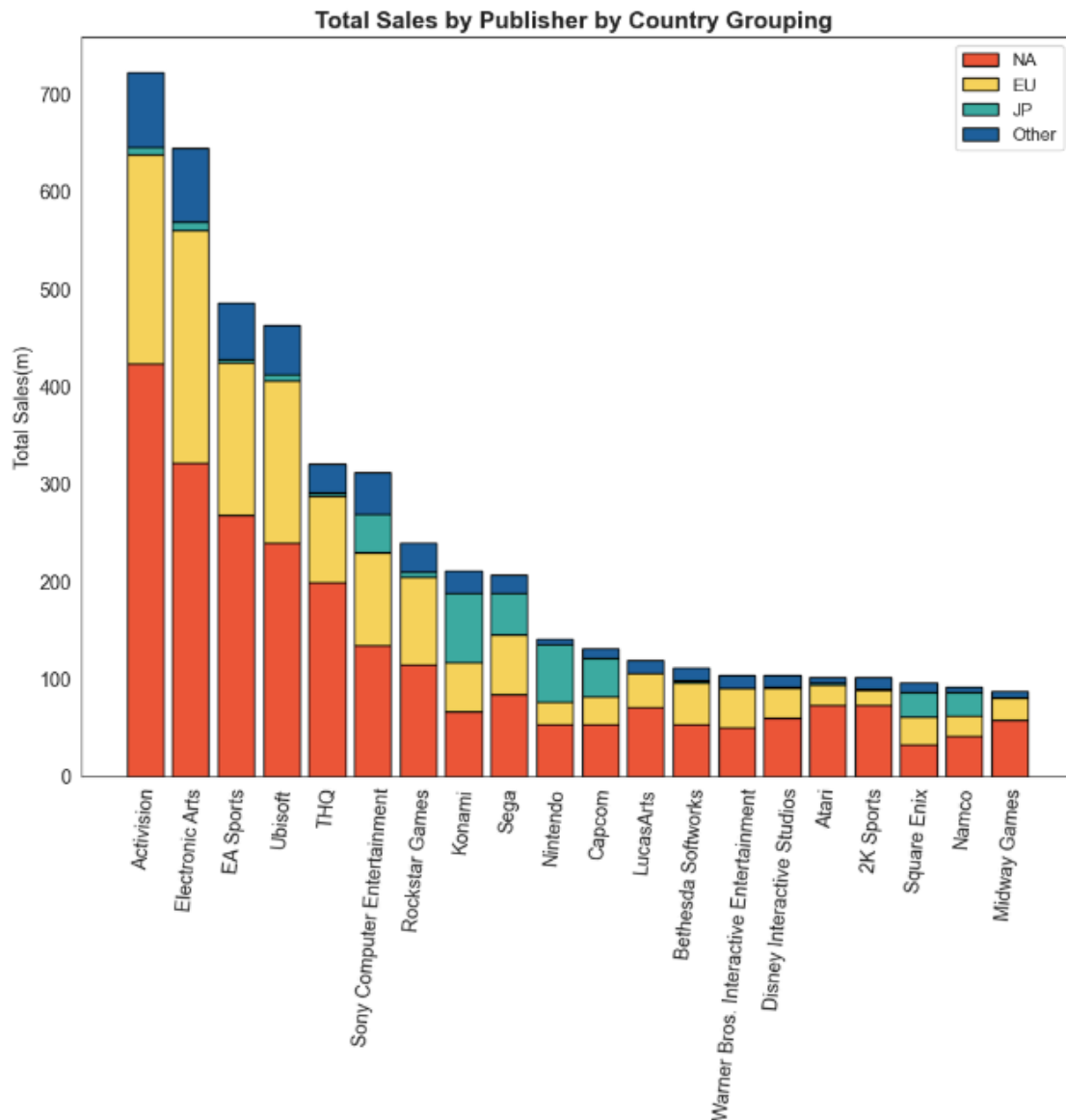
	Title	Genre	Console	Publisher	Total Sales (m)	NA Sales (m)	EU Sales (m)	JP Sales (m)	Other Sales (m)	Release Month	Release Year
0	Tomb Raider II	Adventure	PS	Eidos Interactive	5.24	2.30	2.46	0.20	0.28	10	1997
1	LEGO Indiana Jones: The Original Adventures	Adventure	X360	LucasArts	3.76	2.40	1.01	0.00	0.36	6	2008
2	Tomb Raider III: Adventures of Lara Croft	Adventure	PS	Eidos Interactive	3.54	1.66	1.58	0.12	0.18	11	1998
3	LEGO Batman: The Videogame	Adventure	X360	Warner Bros. Interactive	3.44	2.07	1.04	0.00	0.34	9	2008
4	L.A. Noire	Adventure	PS3	Rockstar Games	3.21	1.29	1.31	0.12	0.49	11	2011

## CHƯƠNG 4: PHÂN TÍCH DỮ LIỆU

### 4.1 Trực quan hóa dữ liệu

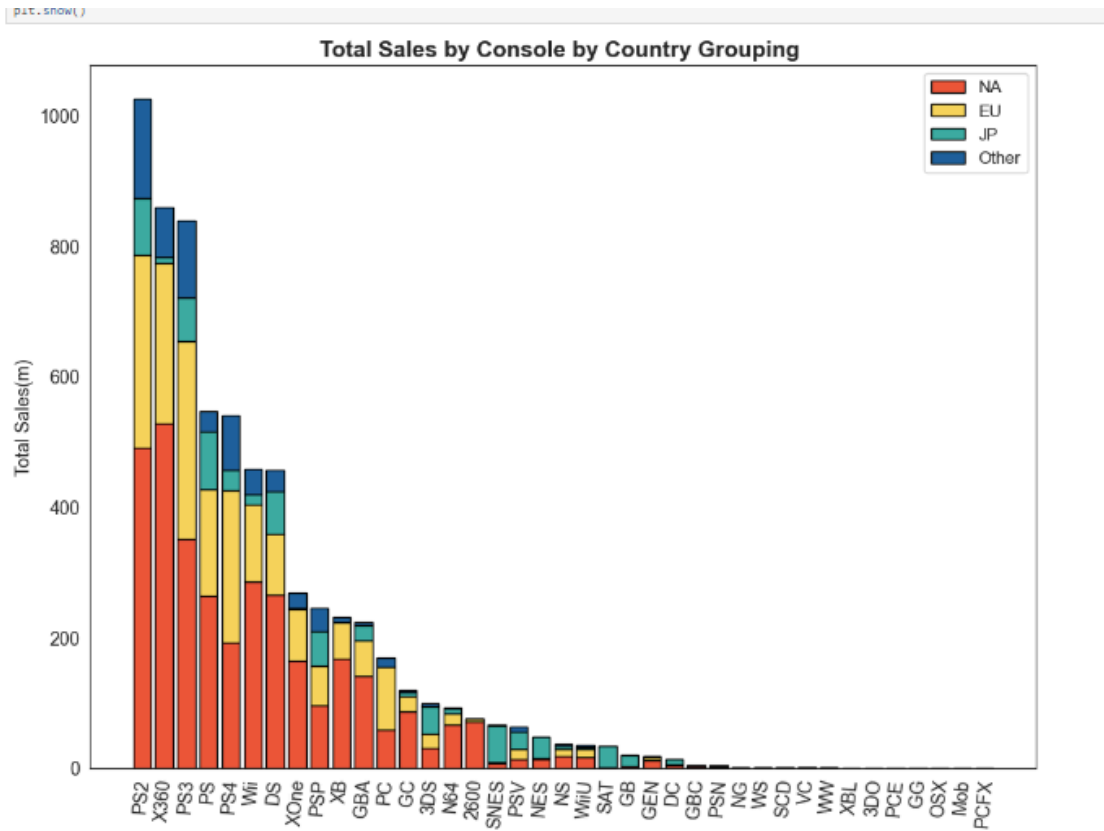


Biểu đồ thể hiện Doanh thu game các khu vực theo thể loại. Các thể loại liên quan vận động mạnh như thể thao, bắn súng, hành động được ưa chuộng.

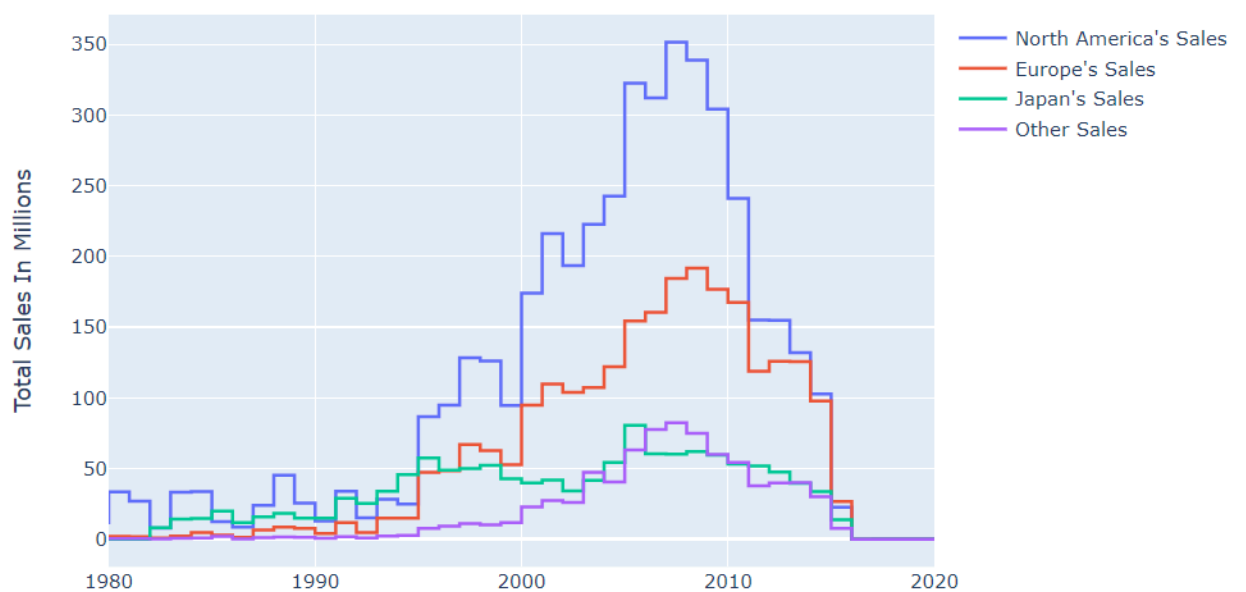


Biểu đồ thể hiện Doanh thu game các khu vực theo nhà phát triển. Sự phân bố đồng đều chứng tỏ một môi trường phát triển năng động, nơi ai cũng có thể kiếm lời từ việc phân phối và làm Game.

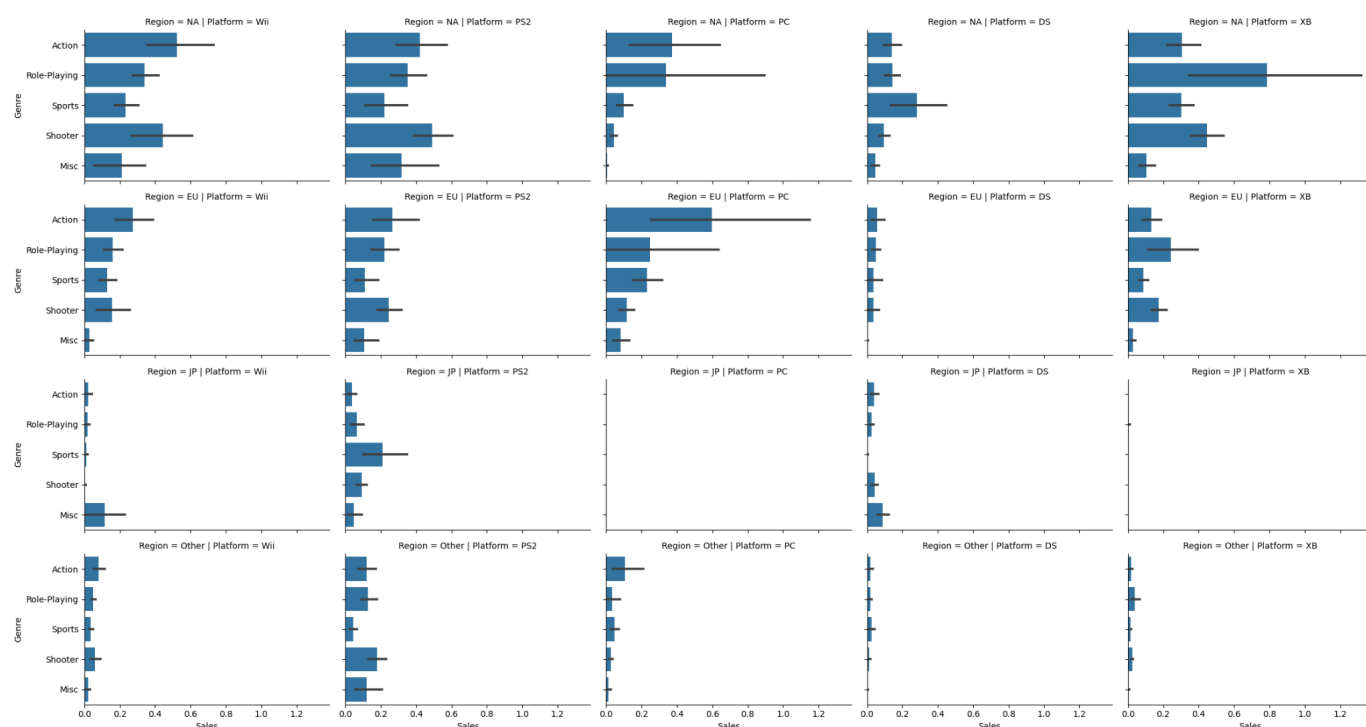




Biểu đồ thể hiện Doanh thu game các khu vực theo hệ máy. Có thể thấy PS và XBOX có doanh thu vượt xa các loại máy khác trên thị trường



Biểu đồ thể hiện Doanh thu game các khu vực theo năm. Bắc Mỹ, Châu Âu và Nhật Bản vẫn là 3 khu vực lớn và là nguồn tiêu thụ game chủ yếu của thế giới trong nửa thế kỷ trở lại đây.



Dựa trên các biểu đồ trên, nhóm xây dựng Biểu đồ mô tả mối quan hệ giữa Thể loại và Doanh số bán hàng, xem xét các quốc gia và nền tảng khác nhau. 5 thể loại và 5 hệ máy phổ biến nhất được lựa chọn với doanh thu từng khu vực.

Biểu đồ cho thấy rằng mỗi khu vực lại có đặc điểm riêng với mỗi thể loại và hệ máy. Ví dụ Bắc Mỹ chuộng hệ máy Wii và PS2, với thể loại là bắn súng và hành động, hay Châu Âu sử dụng hệ máy PC với thể loại hành động.

## **4.2 Tổng kết**

- Doanh thu 3 khu vực lớn ảnh hưởng nhiều đến doanh thu tổng của các tựa game
- Các yếu tố như thể loại game, nhà phát triển hay hệ máy sẽ ảnh hưởng đến các khu vực khác nhau
- Doanh thu game cũng bị ảnh hưởng theo thời điểm phát hành

# CHƯƠNG 5: MÔ HÌNH VÀ PHƯƠNG PHÁP HUẤN LUYỆN

## 5.1 Phương pháp huấn luyện

Phương pháp huấn luyện mô hình bao gồm các bước chuẩn bị dữ liệu, lựa chọn mô hình, huấn luyện mô hình và đánh giá hiệu suất. Các bước cụ thể như sau:

\*Chuẩn bị dữ liệu:

- Khám phá dữ liệu: Hiểu cấu trúc dữ liệu, kiểm tra các giá trị thiếu, thống kê mô tả.
- Tiền xử lý: Xử lý các giá trị thiếu, mã hóa các biến phân loại, chuẩn hóa dữ liệu.

\*Chia dữ liệu:

- Train-test split: Chia tập dữ liệu thành hai phần, 80% dữ liệu để huấn luyện (training) và 20% dữ liệu để kiểm tra (testing). Điều này giúp đánh giá mô hình trên dữ liệu chưa từng thấy, đảm bảo tính tổng quát của mô hình.

```
## Splitting the dataset into independent and dependent variables
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test = train_test_split(x,y,test_size=0.2,random_state=0)
```

- Cross-validation: Áp dụng phương pháp k-fold cross-validation (thường sử dụng k=5 hoặc k=10) để đảm bảo mô hình không overfit và có thể tổng quát hóa tốt trên dữ liệu mới.

```
[18]: ## Training the multiple linear regression on the training set
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import cross_val_score
from sklearn.metrics import accuracy_score
regressor_Multilinear = LinearRegression(fit_intercept=False)
cv_score_lr = cross_val_score(regressor_Multilinear, x, y, cv = 10)
print(cv_score_lr)

mean_accuracy_lr = sum(cv_score_lr)/len(cv_score_lr)
mean_accuracy_lr = mean_accuracy_lr*100
print (mean_accuracy_lr)

[0.99795039 0.98956941 0.99489413 0.98268214 0.99088551 0.99333948
 0.99830092 0.99297101 0.97841403 0.99524016]
99.1424719222715
```

\*Huấn luyện và điều chỉnh mô hình

- Huấn luyện mô hình: Sử dụng dữ liệu huấn luyện để huấn luyện mô hình.
- Điều chỉnh tham số (Hyperparameter Tuning): Sử dụng grid search hoặc random search để tìm các tham số tối ưu cho mô hình.

\*Đánh giá mô hình

- R2 Score: Đo lường mức độ phù hợp của mô hình với dữ liệu thực tế. R2 score càng cao, mô hình càng chính xác.
- Adj. R-squared (hệ số xác định điều chỉnh): Đây là phiên bản điều chỉnh của R-squared, tính toán dựa trên số lượng biến độc lập trong mô hình.
- Đánh giá trên dữ liệu kiểm tra: Từ kết quả dự đoán trên tập kiểm tra để đánh giá hiệu suất mô hình.

## 5.2 Các mô hình lựa chọn

### 5.2.1 Hồi quy tuyến tính đa biến (*Multiple Linear Regression*)

Giới thiệu: Mô hình hồi quy tuyến tính đa biến sử dụng các biến độc lập để dự đoán biến phụ thuộc bằng cách tìm mối quan hệ tuyến tính giữa chúng.

Input: NA\_Sales, EU\_Sales, JP\_Sales, Other\_Sales

Output: Dự đoán Global\_Sales

Tham số:

Mặc định: fit\_intercept=True, copy\_X=True, n\_jobs=None, positive=False

Hiệu chỉnh: Không có

Lý do chọn: Đây là mô hình cơ bản và thường được sử dụng làm baseline để so sánh với các mô hình phức tạp hơn

### 5.2.2 Hồi quy đa thức (*Polynomial Regression*)

Giới thiệu: Mô hình hồi quy đa thức mở rộng hồi quy tuyến tính bằng cách thêm các biến độc lập dưới dạng đa thức để nắm bắt mối quan hệ phi tuyến.

Input: Các đặc trưng sau khi biến đổi đa thức từ các đặc trưng ban đầu.

Output: Dự đoán Global\_Sales

Tham số:

Mặc định: degree=2

Hiệu chỉnh: Không có

Lý do chọn: Mô hình này giúp nắm bắt các mối quan hệ phi tuyến giữa các đặc trưng và biến mục tiêu.

### 5.2.3 Hồi quy K-Nearest Neighbors (*KNN Regression*)

Giới thiệu: Mô hình KNN sử dụng khoảng cách để dự đoán giá trị của biến phụ thuộc dựa trên các điểm dữ liệu gần nhất.

Input: Các đặc trưng đã được chuẩn hóa.

Output: Dự đoán Global\_Sales

Tham số: Mặc định: n\_neighbors=5

Hiệu chỉnh: n\_neighbors (số lượng hàng xóm)

Lý do chọn: KNN là một mô hình phi tham số hữu ích khi dữ liệu không có cấu trúc tuyến tính rõ ràng.

#### **5.2.4 Hồi quy cây quyết định (Decision Tree Regression)**

Giới thiệu: Mô hình cây quyết định sử dụng cấu trúc cây để dự đoán giá trị của biến phụ thuộc dựa trên các điều kiện phân chia.

Input: Các đặc trưng đã được chuẩn hóa.

Output: Dự đoán Global\_Sales

Tham số:

Mặc định: min\_samples\_split=2

Hiệu chỉnh: min\_samples\_leaf (số mẫu tối thiểu ở mỗi lá)

Lý do chọn: Cây quyết định giúp nắm bắt các tương tác phi tuyến và có thể điều chỉnh để tránh overfitting.

#### **5.2.5 Hồi quy rừng ngẫu nhiên (Random Forest Regression)**

Giới thiệu: Mô hình rừng ngẫu nhiên sử dụng nhiều cây quyết định để cải thiện độ chính xác và độ ổn định của dự đoán.

Input: Các đặc trưng đã được chuẩn hóa.

Output: Dự đoán Global\_Sales

Tham số:

Mặc định: n\_estimators=100

Hiệu chỉnh: n\_estimators=300

Lý do chọn: Kết hợp nhiều cây quyết định để cải thiện độ chính xác và giảm overfitting.

#### **5.2.6 Hồi quy Vector hỗ trợ tuyến tính (Linear SVR)**

Giới thiệu: Mô hình SVR tuyến tính sử dụng hàm mất mát epsilon-insensitive để tìm siêu phẳng tối ưu dự đoán giá trị của biến phụ thuộc.

Input: Các đặc trưng đã được chuẩn hóa.

Output: Dự đoán Global\_Sales

Tham số:

Mặc định: kernel='linear'

Hiệu chỉnh: C (tham số điều chỉnh độ phạt)

Lý do chọn: Mô hình mạnh mẽ cho dữ liệu tuyến tính và dễ dàng giải thích.

### **5.2.7 Hồi quy Vector hỗ trợ phi tuyến (Non-linear SVR)**

Giới thiệu: Mô hình SVR phi tuyến sử dụng các kernel như RBF để nắm bắt các mối quan hệ phi tuyến trong dữ liệu.

Input: Các đặc trưng đã được chuẩn hóa.

Output: Dự đoán Global\_Sales

Tham số:

Mặc định: kernel='rbf'

Hiệu chỉnh: C, gamma (tham số của kernel)

Lý do chọn: Sử dụng các kernel để nắm bắt các mối quan hệ phi tuyến trong dữ liệu.

### **5.2.8 Hồi quy XGBoost (XGBoost Regression)**

Giới thiệu: Mô hình XGBoost là một kỹ thuật boosting mạnh mẽ, thường cho kết quả tốt với dữ liệu phức tạp và có khả năng chống overfitting cao.

Input: Các đặc trưng đã được chuẩn hóa.

Output: Dự đoán Global\_Sales

Tham số:

Mặc định: n\_estimators=100

Hiệu chỉnh: learning\_rate, max\_depth

Lý do chọn: Mô hình mạnh mẽ, thường cho kết quả tốt với dữ liệu phức tạp và có khả năng chống overfitting cao

### **5.2.9 Hồi quy OLS:**

Giới thiệu: Mô hình hồi quy OLS (Ordinary Least Squares) là một kỹ thuật hồi quy tuyến tính cổ điển. Đây là phương pháp phổ biến để ước lượng các tham số

trong mô hình hồi quy tuyến tính, giúp tối thiểu hóa tổng bình phương các sai số giữa giá trị thực tế và giá trị dự đoán. OLS thường được sử dụng như một điểm khởi đầu cơ bản để so sánh với các mô hình phức tạp hơn.

Input: Các đặc trưng đã được chuẩn hóa.

Output: Dự đoán Global\_Sales.

Tham số:

Mặc định:

Hiệu chỉnh: Không có

Lý do chọn: Mô hình hồi quy OLS đơn giản, dễ hiểu và dễ triển khai. Mô hình cho phép kiểm tra mối quan hệ tuyến tính giữa các đặc trưng và biến mục tiêu, từ đó có thể cung cấp cái nhìn tổng quan ban đầu về dữ liệu trước khi áp dụng các mô hình phức tạp hơn.

### ***\*Tổng kết:***

Việc sử dụng nhiều mô hình khác nhau giúp chúng ta so sánh và lựa chọn mô hình tốt nhất cho bài toán dự đoán doanh số toàn cầu của trò chơi. Mỗi mô hình có ưu và nhược điểm riêng, và quá trình điều chỉnh tham số là cần thiết để tối ưu hóa hiệu suất của từng mô hình.

## **5.3 Tối ưu hóa mô hình**

Sử dụng hệ số Mean Absolute Error (MAE): MAE tính trung bình của giá trị tuyệt đối của sai số giữa giá trị dự đoán và giá trị thực tế.

Công thức: 
$$MAE = \frac{1}{n} \sum_{i=1}^n |y_{true,i} - y_{pred,i}|$$

MAE đo lường độ lớn trung bình của sai số mà không xem xét hướng của sai số. Dùng để đánh giá hiệu suất của mô hình hồi quy.

Khi giá trị của cả Training Loss và Validation Loss đều thấp là dấu hiệu của một mô hình có khả năng dự đoán tốt trên dữ liệu huấn luyện cũng như dữ liệu mới. Từ đó vẽ biểu đồ để so sánh các tham số của mô hình từ đó chọn được mô hình tối ưu.

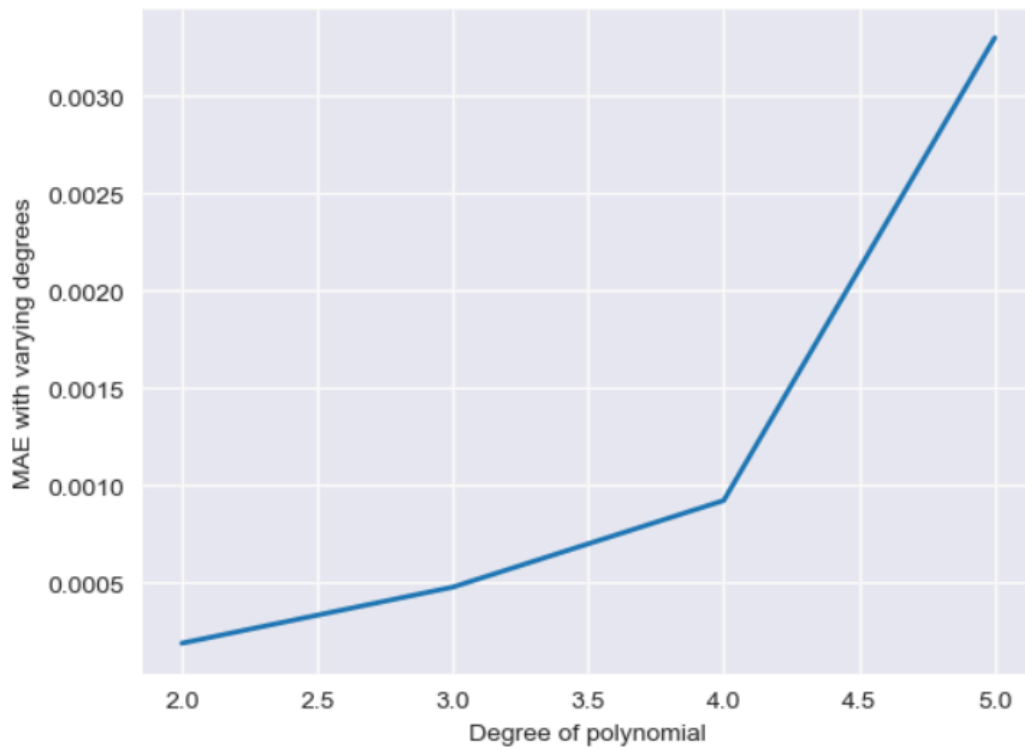
### ***\*Multiple Linear Regression***

Validation Loss: 0.018328747947102826

Training Loss: 0.018241806543339678



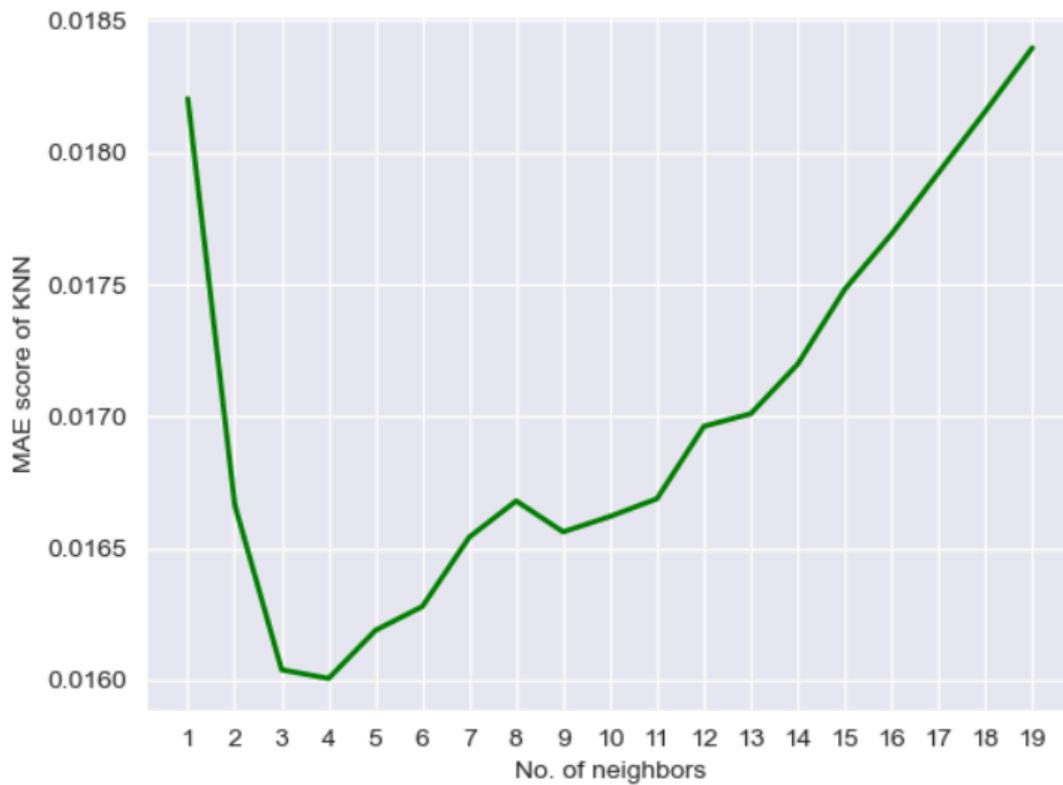
\* Hồi quy đa thức (Polynomial Regression)



Biểu đồ độ chênh lệch giữa Validation Loss và Training Loss

Kết luận Degree=2 là tốt nhất

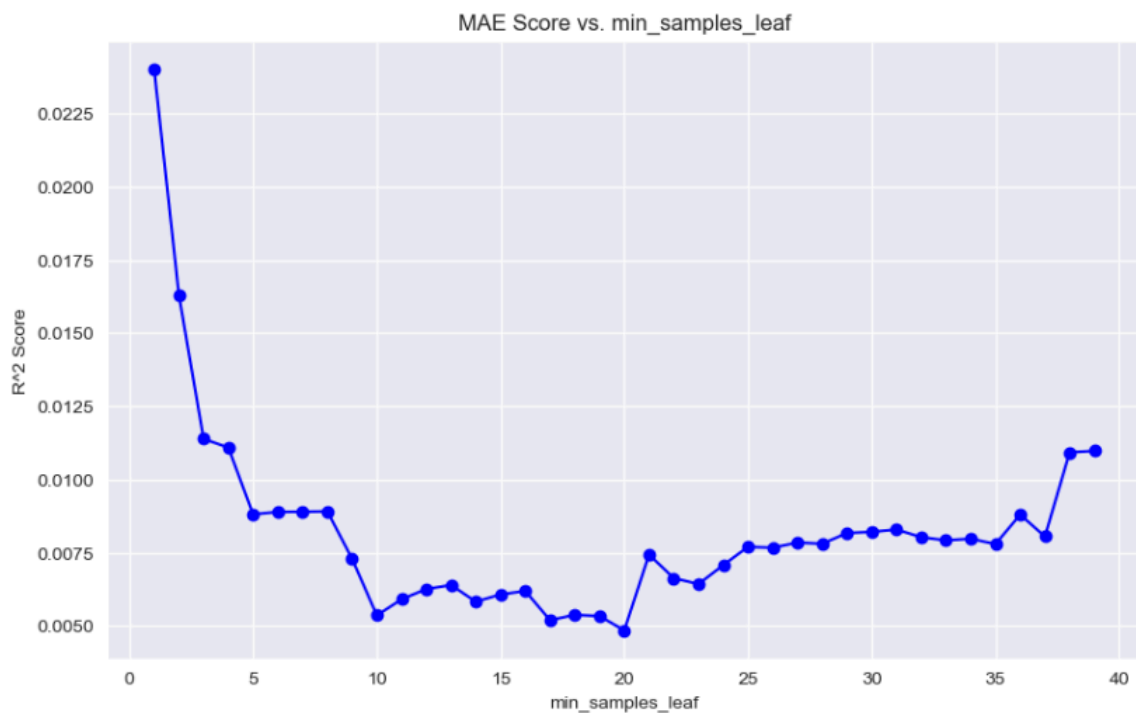
\*Hồi quy K-Nearest Neighbors (KNN Regression)



Biểu đồ độ chênh lệch giữa Validation Loss và Training Loss

Kết luận:  $n\_neighbors=4$

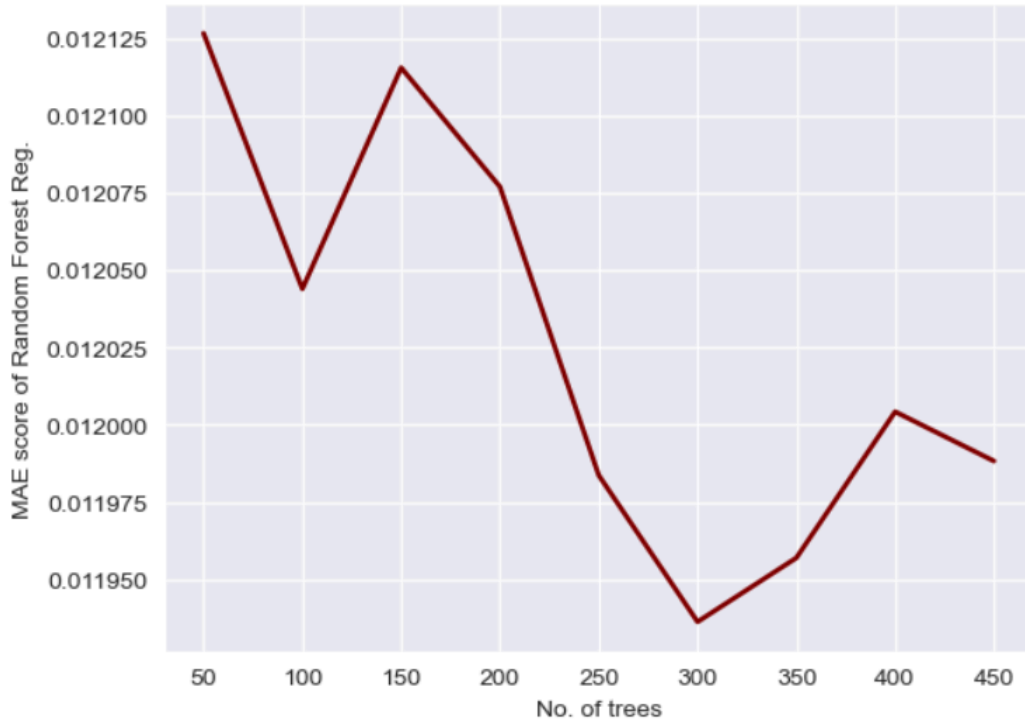
\*Hồi quy cây quyết định (Decision Tree Regression)



Biểu đồ độ chênh lệch giữa Validation Loss và Training Loss

min\_samples\_leaf=20

\*Random Forest Regression

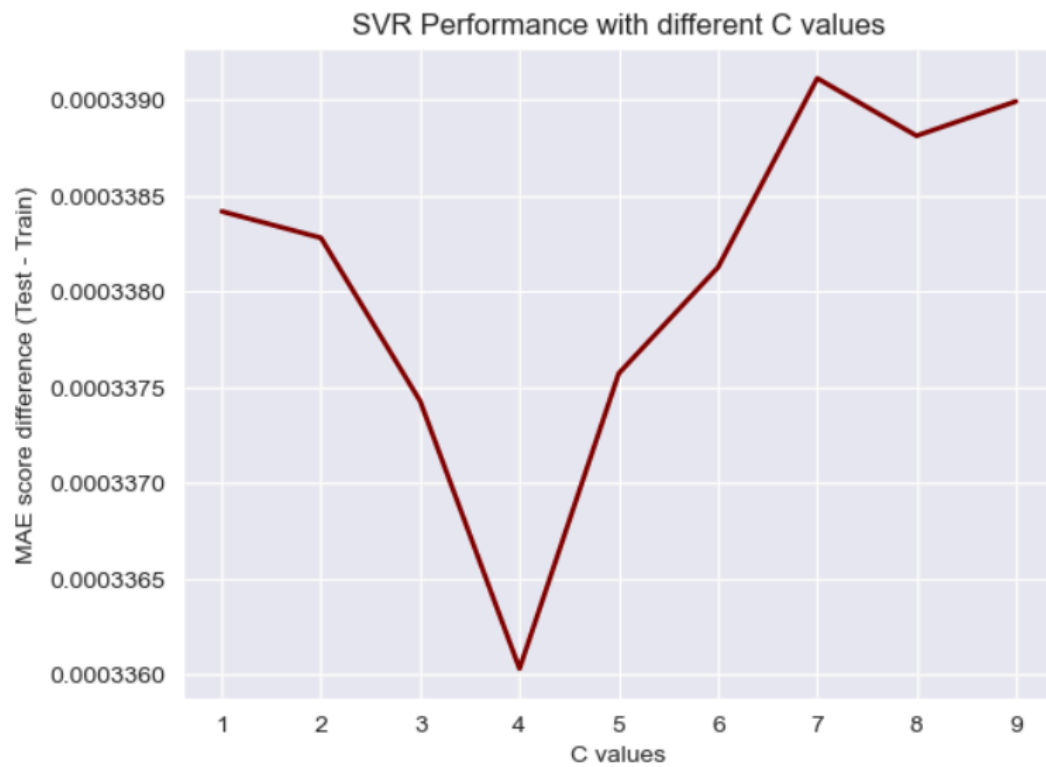


Biểu đồ độ chênh lệch giữa Validation Loss và Training Loss

Kết luận : n\_estimators=300

\*Hồi quy Vector hỗ trợ tuyến tính (Linear SVR)

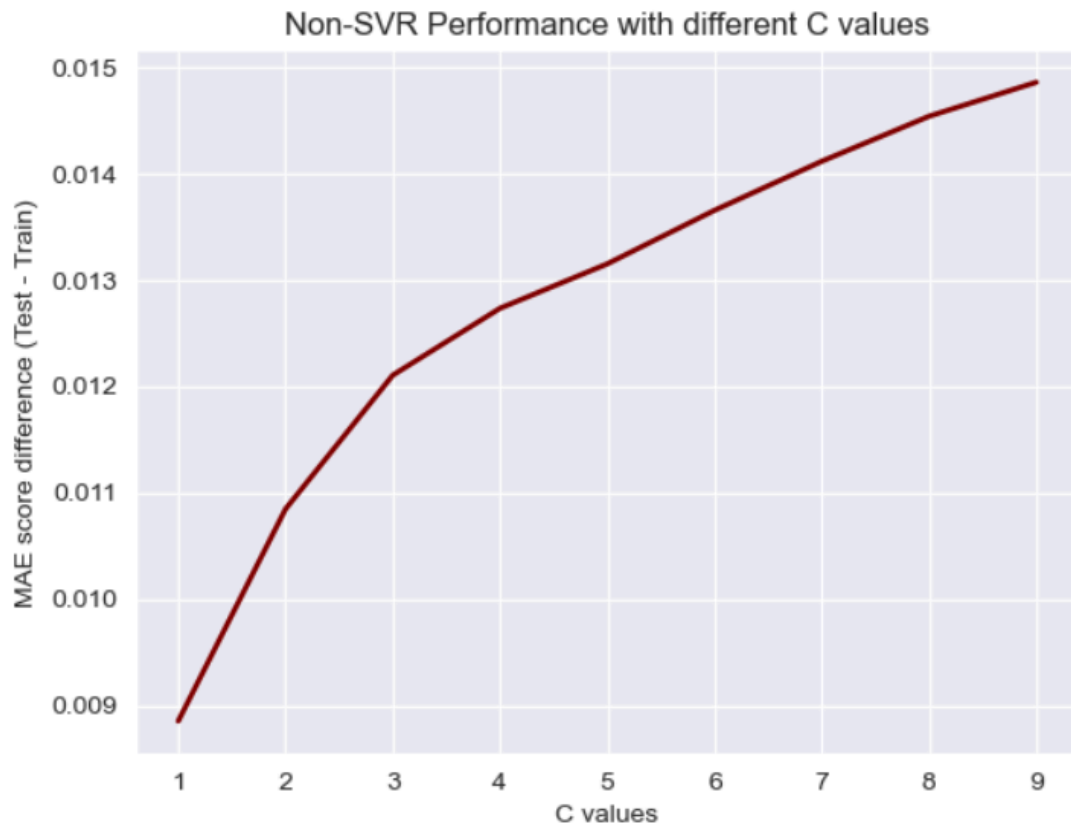
\*Hồi quy Vector hỗ trợ phi tuyến (Non-linear SVR)



Biểu đồ độ chênh lệch giữa Validation Loss và Training Loss

Kết luận : C=4, Gamma=auto

\*Hồi quy XGBoost (XGBoost Regression)

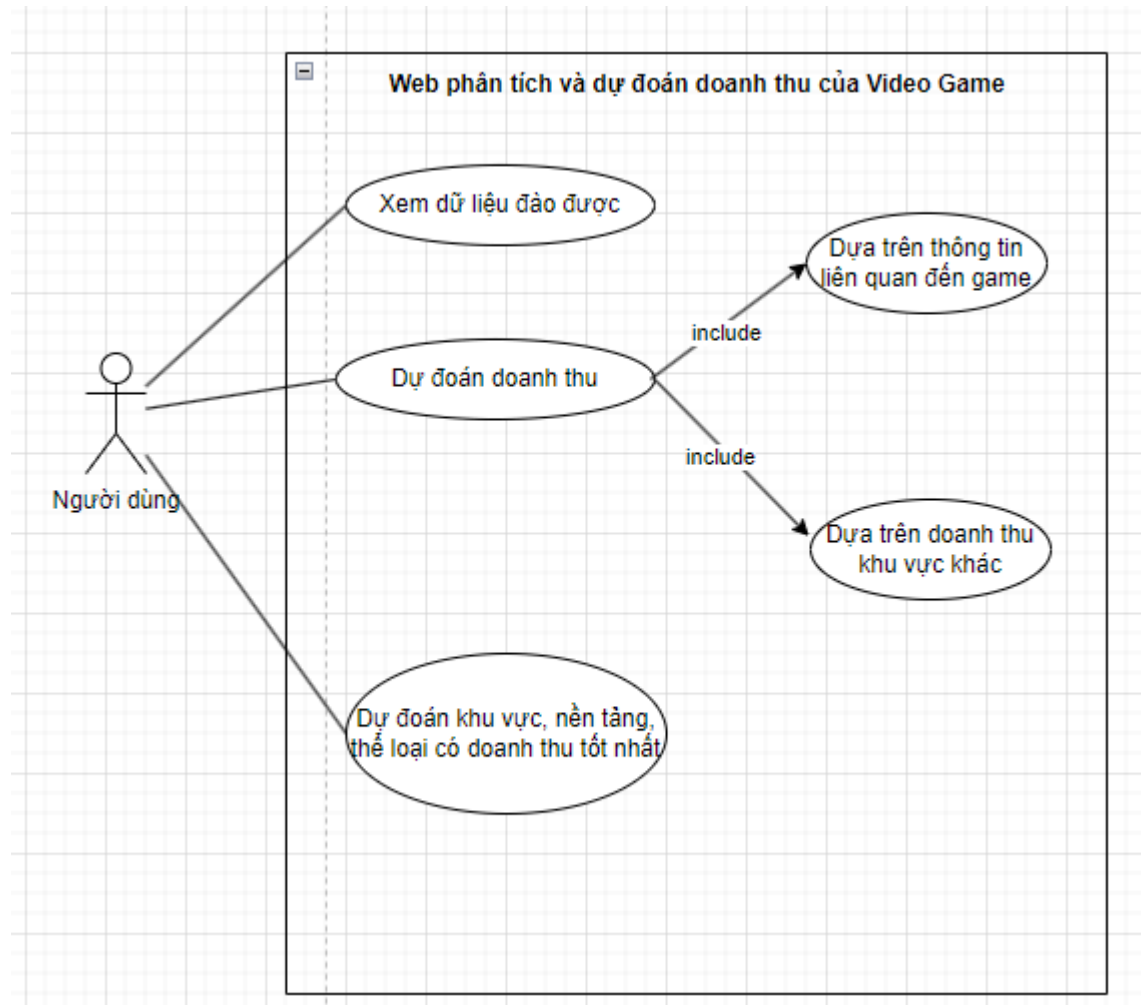


Biểu đồ độ chênh lệch giữa Validation Loss và Training Loss

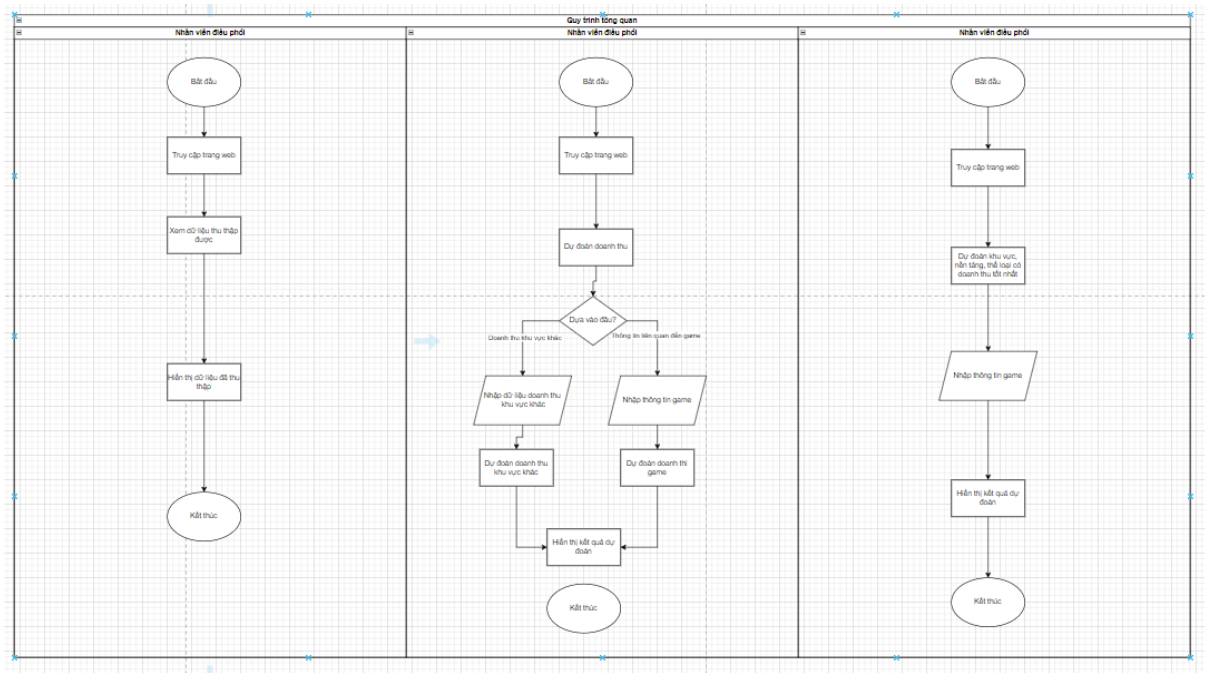
Kết luận :  $C=1$ ,  $\text{Gamma}=\text{auto}$

## CHƯƠNG 6: Triển khai thành sản phẩm

### 6.1 Sơ đồ use case



### 6.2 Flowchart



## 6.3 Các màn hình chức năng

### 6.3.1 Màn hình trang chủ

Game	Console	Publisher	Release Date	NA Revenue	JP Revenue	EU Revenue	Revenue Prediction
Tomb Raider II	PS	Eidos Interactive	1997-10	2.3	0.2	2.46	5.24
LEGO Indiana Jones: The Original Adventures	X360	LucasArts	2008-6	2.4	0.0	1.01	3.76
Tomb Raider III: Adventures of Lara Croft	PS	Eidos Interactive	1998-11	1.66	0.12	1.58	3.54
LEGO Batman: The Videogame	X360	Warner Bros. Interactive	2008-9	2.07	0.0	1.04	3.44
L.A. Noire	PS3	Rockstar Games	2011-11	1.29	0.12	1.31	3.21
Club Penguin: Elite Penguin Force	DS	Disney Interactive Studios	2008-11	1.87	0.0	0.97	3.14
LEGO Batman: The Videogame	Wii	Warner Bros. Interactive	2008-9	1.8	0.0	0.98	3.08
LEGO Batman: The Videogame	DS	Warner Bros. Interactive	2008-9	1.75	0.0	1.02	3.06
L.A. Noire	X360	Rockstar Games	2011-11	1.55	0.02	0.92	2.73
Harry Potter and the Chamber of Secrets	PS2	Electronic Arts	2002-11	0.9	0.04	1.22	2.61

### 6.3.2 Màn hình dự đoán game theo doanh số

Logo

Home

Game Prediction

Analysis

Game Comparing

Dự đoán doanh số

Dự đoán theo đặc trưng game

Gợi ý khu vực phát hành

\* Game:

\* NA Revenue:

\* JP Revenue:

\* EU Revenue:

\* Other Sales:

\* Release Month:

\* Release Year:

Submit

Reset

Game	NA Revenue	JP Revenue	EU Revenue	Other Sales	Release Month	Release Year	Multi Linear	Polynomial Linear	KNN	Decision Tree	Random Forest	SVR Linear	SVR Non Linear	XGB
abcgame	1.2	4.5	0.9	1	2	2024	7.089374608321353	6.821719418291871	3.2249999999999996	2.2776000000000005	2.5659333333333335	1.7790456143541287	1.8836768075681983	1.6548741724014282
edugame	1.2	3.6	4.5	2.1	6	2023	10.56594329504578	12.47052148193696	7.1325	6.795517241379314	8.156366666666664	3.466033775451638	3.813440795230988	3.1026105308532715

6.3.3 Màn hình dự đoán game theo đặc trưng game

Logo

Home

Game Prediction

Analysis

Game Comparing

Dự đoán doanh số

Dự đoán theo đặc trưng game

Gợi ý khu vực phát hành

\* Console: Select a console

\* Genre: Select a genre

\* Publisher:

\* NA Sales:

\* EU Sales:

\* JP Sales:

Submit

Reset

Console	Genre	Publisher	NA Sales	EU Sales	JP Sales	Multi Linear	Polynomial Linear	KNN	Decision Tree	Random Forest	SVR Linear	SVR Non Linear	XGB
PS	Action	vuleminh	2.3	4.5	3.1	11.211526313209463	12.412282963393626	5.7025000000000001	6.795517241379315	8.174033333333336	4.2194016425931915	3.798630434901991	4.1235270500183105
DS	Board Game	mikigame	6.1	2.4	1.3	10.77976880341977	10.771919534287311	3.2925000000000004	12.575454545454546	10.635266666666667	4.797460150022175	3.8266449367843984	4.3418192863464355

6.3.4 Màn hình dự đoán khu vực/hệ máy phát hành tốt nhất



Logo
Home
Game Prediction
Analysis
Game Comparing

Dự đoán theo doanh số
Dự đoán theo đặc trưng game
Gợi ý khu vực phát hành

\* Genre: Select a genre
Region: Select a region
Console: Select a console
Submit
Reset

Genre	Region	Console	Region Max	Console Max	Revenue Max
Adventure	JP		JP	NES	0.3468976856839384
Action		X360	NA	X360	0.34220430223617326
Action-Adventure		PS	NA	PS	0.311979810111306
Action-Adventure	PAL		PAL	NES	0.49550514077561714

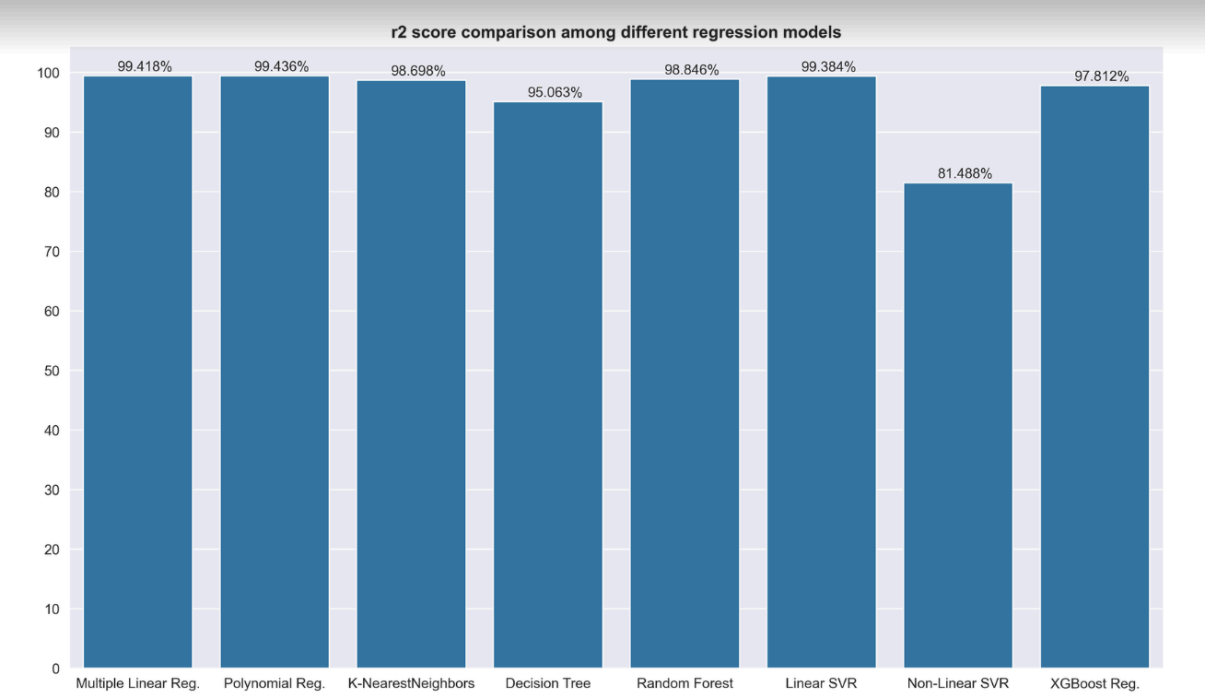
1

## CHƯƠNG 7: Kết luận

### 7.1 Kết quả thực nghiệm

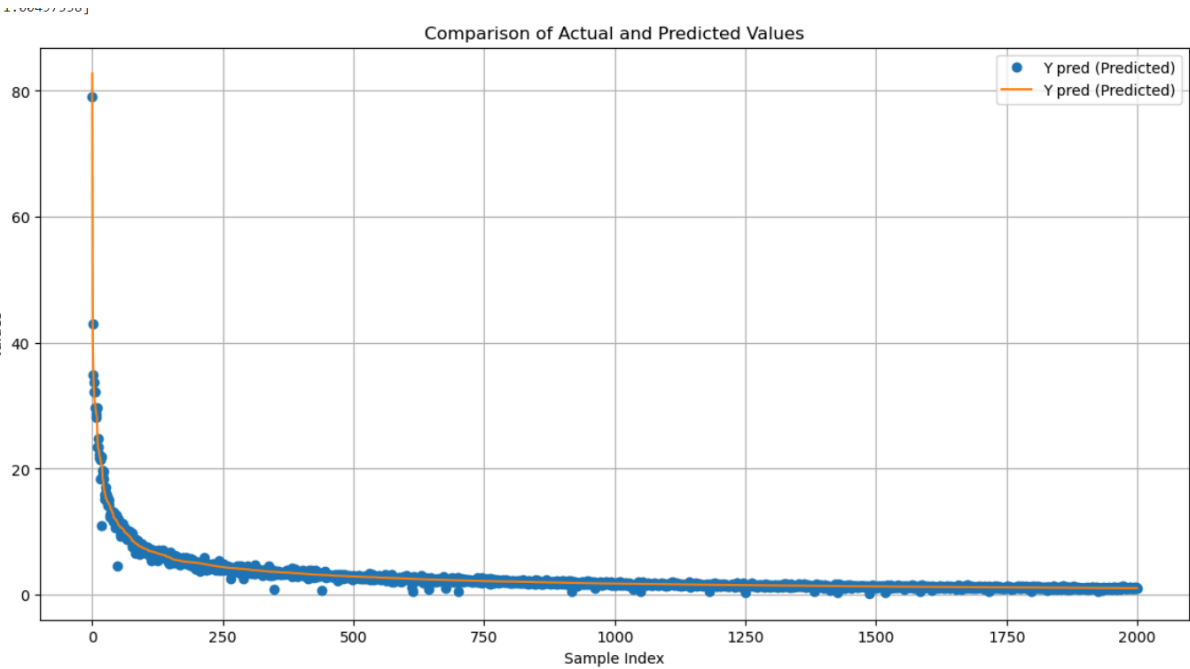
Option 1 : Dự đoán dựa trên doanh thu 3 khu vực lớn

Đánh giá bằng R2



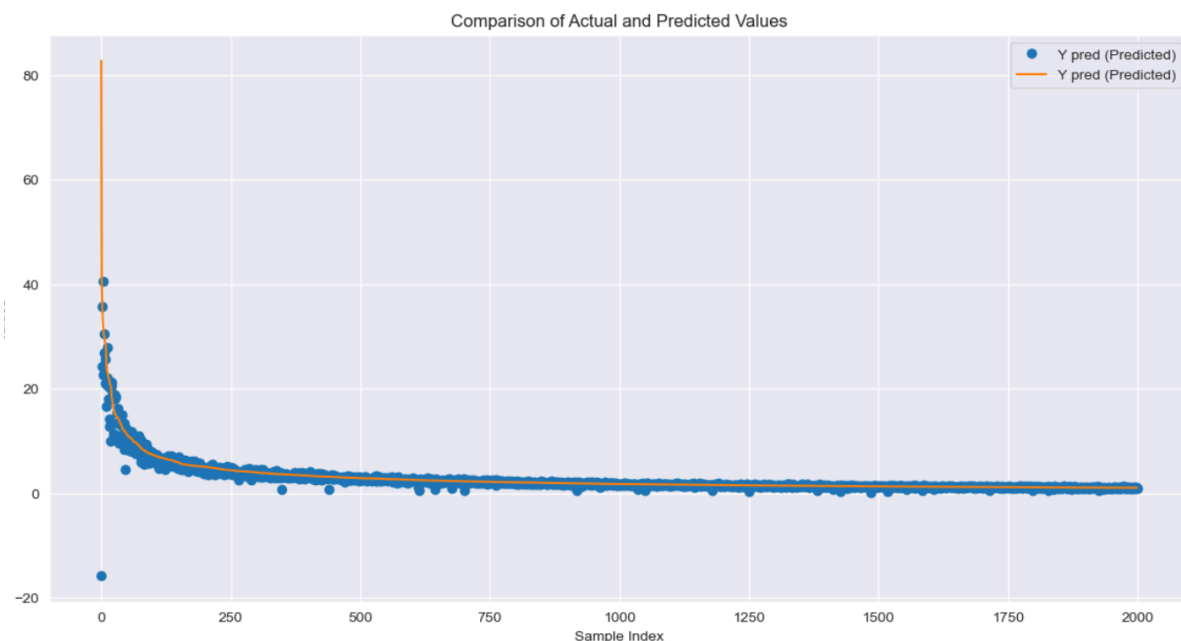
Kết quả thực tế

Multi Linear Regression



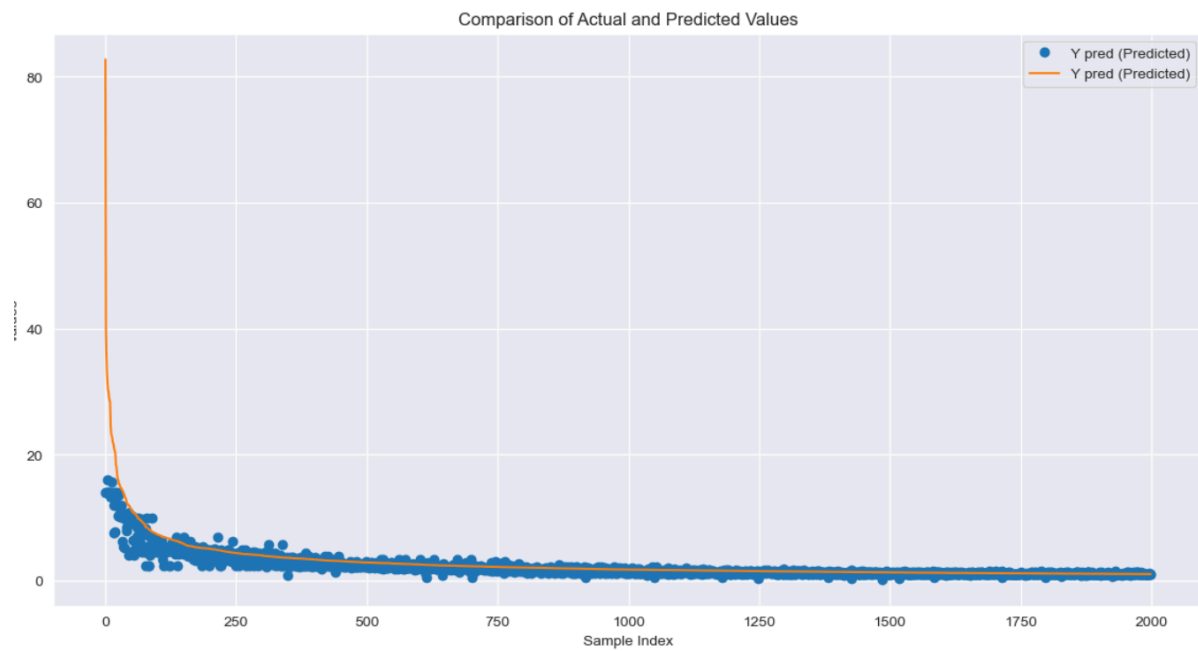
## Polynomial Linear Regression

```
## Training the polynomial regression on the training model  
poly_reg = PolynomialFeatures(degree=2)  
x_poly = poly_reg.fit_transform(x_train)  
poly_regressor = LinearRegression()  
poly_regressor.fit(x_poly,y_train)  
y_pred = poly_regressor.predict(poly_reg.fit_transform(x_test))  
r2_poly = r2_score(y_test,y_pred)  
print(r2_poly)
```

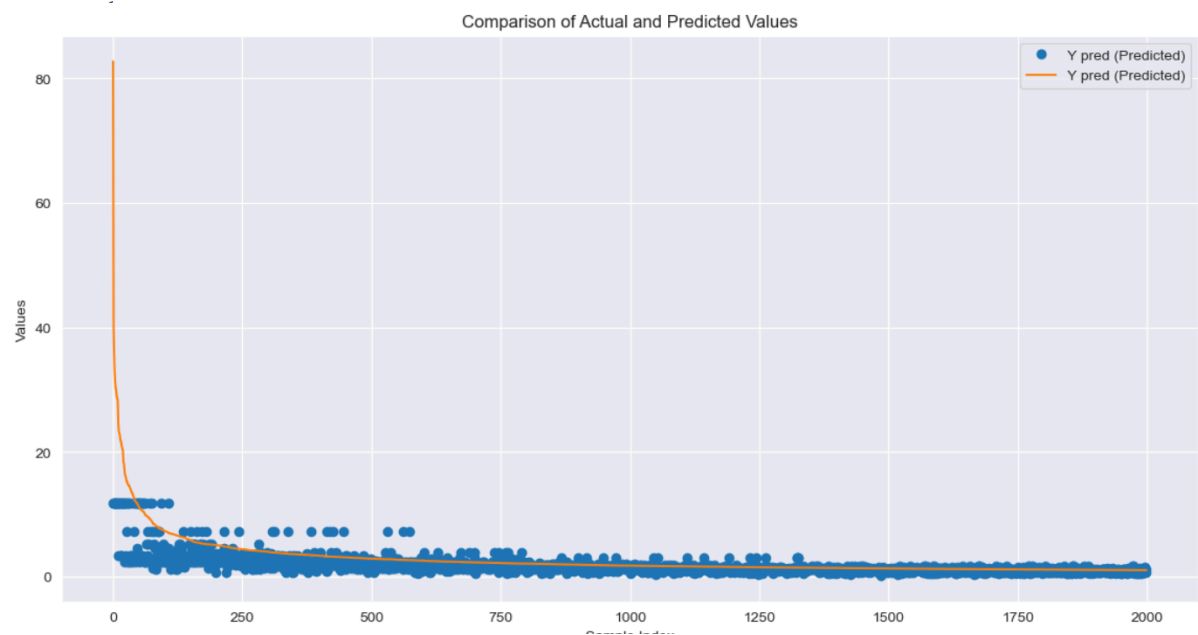


## KNN Regression

```
# Training the KNN model on the training set  
regressor_knn = KNeighborsRegressor(n_neighbors=4)  
regressor_knn.fit(x_train,y_train)  
y_pred = regressor_knn.predict(x_test)  
r2_knn = r2_score(y_test,y_pred)  
print(r2_knn)
```

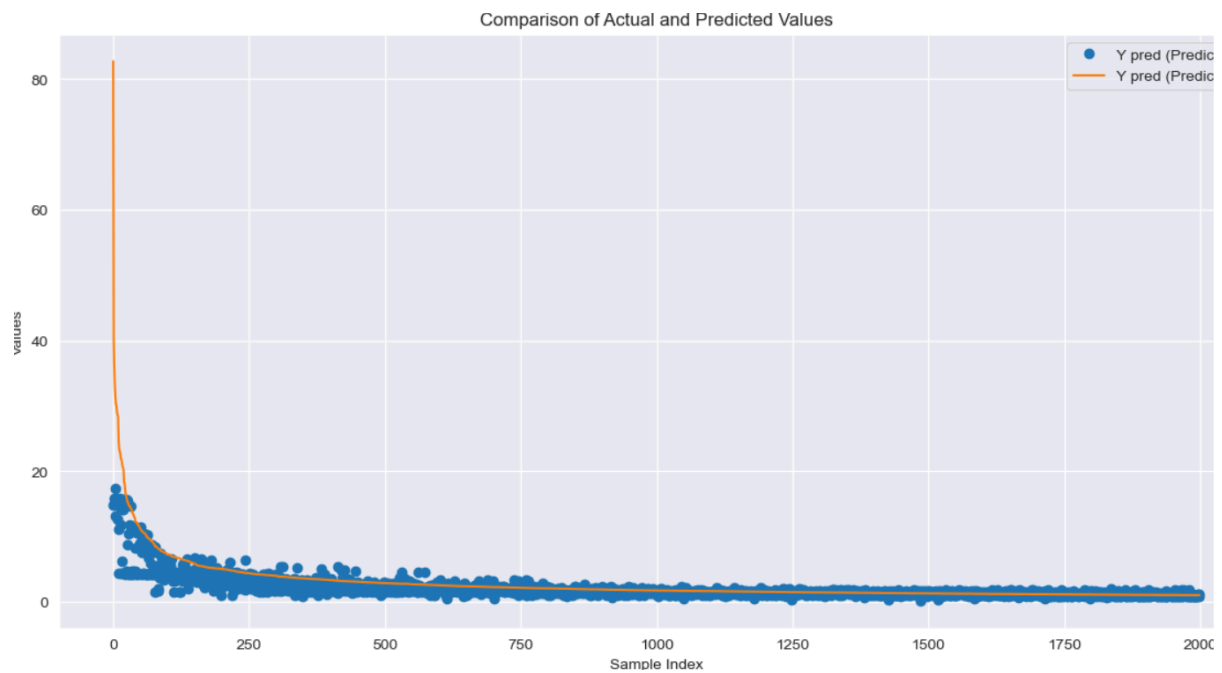


## Decision Tree Regression



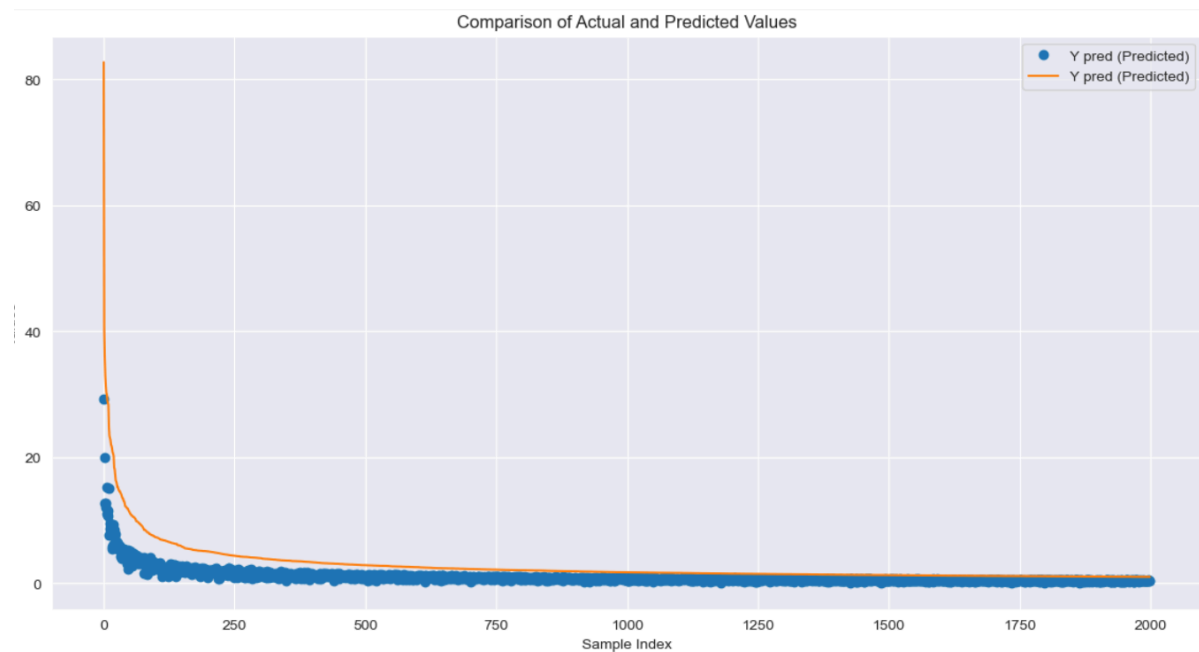
## Random Forest Regression

```
[21]: # Training the Random Forest regression on the training model
regressor_Forest = RandomForestRegressor(n_estimators=300, random_state=0)
regressor_Forest.fit(x_train, y_train)
y_pred = regressor_Forest.predict(x_test)
r2_forest = r2_score(y_test, y_pred)
print(r2_forest)
```



## SVR Linear

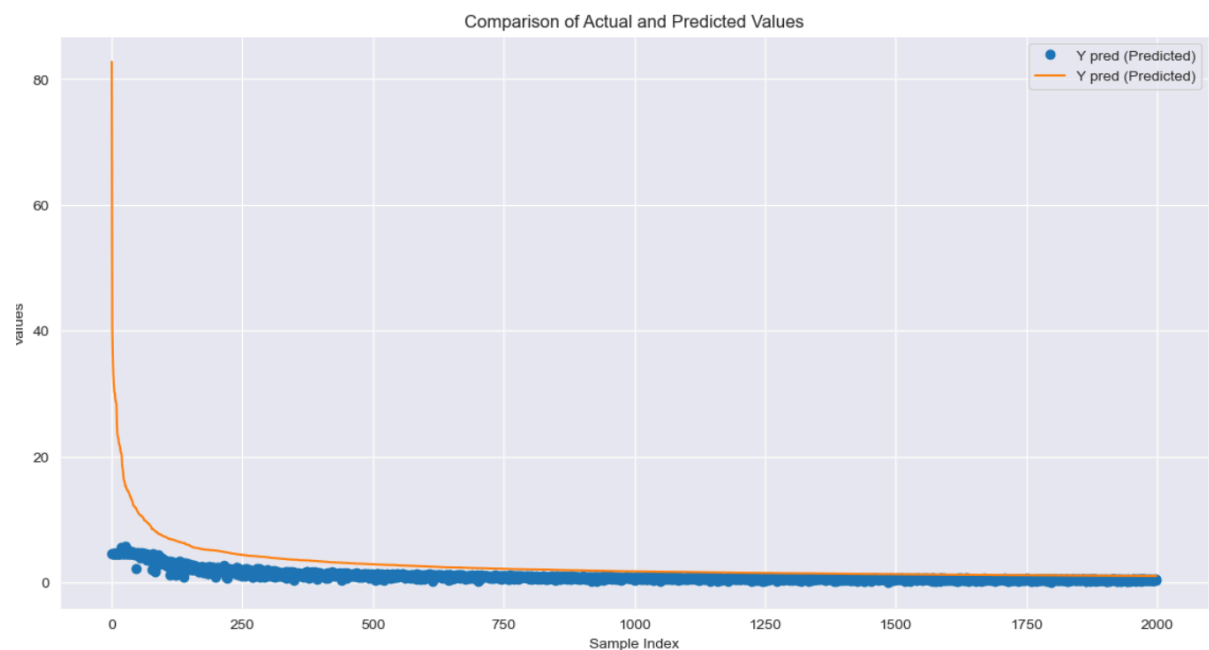
```
## Training the Linear SVR model on the training set
from sklearn.svm import SVR
regressor_SVR = SVR(kernel='linear', degree=4, gamma='auto')
regressor_SVR.fit(x_train, y_train)
```



## SVR Non Linear

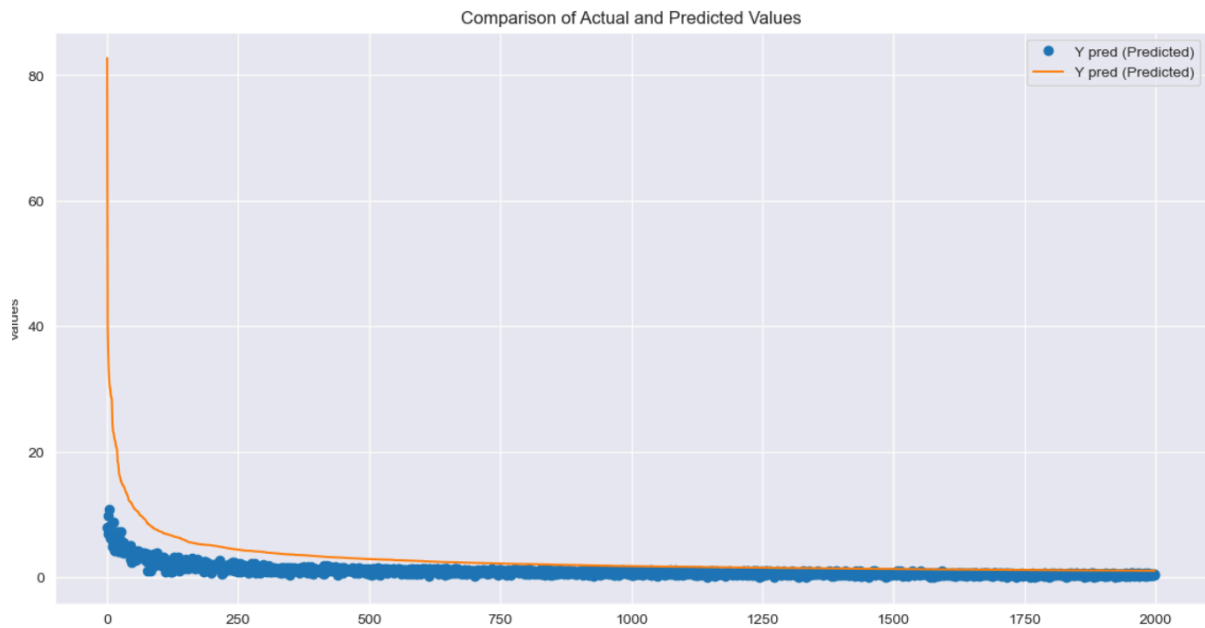
```
1]: ## Training the Non-linear SVR model on the training set  
from sklearn.svm import SVR  
regressor_NonLinearSVR = SVR(kernel='rbf', degree=1, gamma='auto')  
regressor_NonLinearSVR.fit(x_train, y_train)
```

```
1]: SVR  
SVR(degree=1, gamma='auto')
```

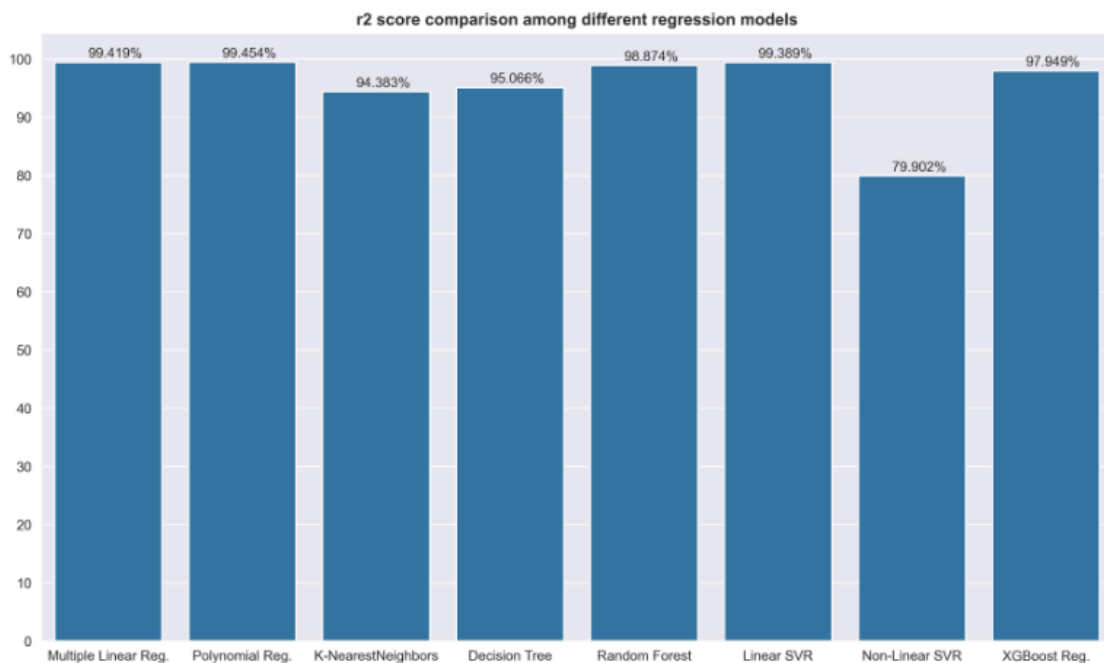


## XGB Regression

```
from xgboost import XGBRegressor  
regressor_xgb = XGBRegressor()  
regressor_xgb.fit(x_train, y_train)
```



Option 2 :Dự đoán dựa trên doanh thu đặc trưng của game



## 7.2 Các khó khăn và hướng phát triển trong tương lai

### 7.2.1 Khó Khăn

- Dữ liệu không đầy đủ hoặc không cập nhật liên tục: Dữ liệu về doanh thu game có thể không hoàn chỉnh hoặc không cập nhật liên tục, đặc biệt là khi thu thập từ nhiều nguồn khác nhau. Điều này có thể dẫn đến hiệu suất dự đoán kém nếu model được huấn luyện trên dữ liệu không đủ.
- Mô hình không đủ phức tạp: Một mô hình hồi quy đơn giản có thể không đủ mạnh để hiểu và dự đoán các yếu tố phức tạp ảnh hưởng đến doanh thu game như xu hướng thị trường, sở thích của người chơi, hoặc chiến lược tiếp thị.
- Nhiều trong dữ liệu: Dữ liệu về doanh thu game có thể bị ảnh hưởng bởi nhiều yếu tố ngoại vi như sự kiện đặc biệt, thông tin tiếp thị, hay thậm chí là các sự kiện bất ngờ như dịch bệnh.
- Quản lý biến số: Doanh thu game có thể bị ảnh hưởng bởi nhiều yếu tố như giá cả, đánh giá từ người chơi, hoặc thậm chí là thông tin trên các trang web xã hội. Quản lý và xử lý đồng thời nhiều biến số này có thể phức tạp

### **7.2.1 Hướng Phát Triển:**

- Thu thập và tiền xử lý dữ liệu cẩn thận: Đảm bảo rằng dữ liệu được thu thập từ nhiều nguồn có chất lượng và đầy đủ. Tiền xử lý dữ liệu là bước quan trọng để loại bỏ nhiễu và chuẩn hóa dữ liệu để mô hình hoạt động tốt hơn.
- Sử dụng mô hình phức tạp hơn: Thay vì sử dụng một mô hình hồi quy đơn giản, có thể cần phát triển mô hình phức tạp hơn như mạng nơ-ron hồi quy (RNN) hoặc mạng nơ-ron hồi quy dài hạn ngắn (LSTM) để hiểu các mẫu phức tạp trong dữ liệu.
- Sử dụng kỹ thuật feature engineering: Xây dựng các đặc trưng (features) phản ánh các yếu tố quan trọng ảnh hưởng đến doanh thu game như thời tiết, sự kiện đặc biệt, hoặc sở thích của người chơi.
- Kiểm tra và đánh giá thường xuyên: Kiểm tra và đánh giá hiệu suất của mô hình thường xuyên trên tập dữ liệu mới và kiểm tra xem nó có đáp ứng được yêu cầu dự đoán không.
- Kết hợp nhiều mô hình: Có thể kết hợp nhiều mô hình dự đoán để tận dụng sức mạnh của mỗi mô hình và cải thiện hiệu suất dự đoán tổng thể.

## **7.3 Kết luận**

Về cơ bản nhóm đã hoàn thành được mục tiêu đề ra là tạo được một hệ thống dự đoán doanh thu game bằng những mô hình đơn giản bằng những áp dụng các kiến thức lý thuyết từ môn học Data Science, bao gồm các phương pháp hồi quy, các phép đo hiệu suất mô hình, và các kỹ thuật xử lý dữ liệu. Mô hình về cơ bản đã giải quyết được bài toán ở mức độ cơ bản, giúp xác định được dự đoán của các tựa game dựa trên yếu tố chủ quan lẫn khách quan. Các mô hình đa dạng cũng giúp việc dự đoán có chiều sâu và thể hiện sự đa dạng các đầu vào của mô hình. Tuy nhiên, bên cạnh vẫn còn khá nhiều thách thức để có thể tạo ra được một hệ thống có độ chính xác cao, lồng ghép nhiều hơn các yếu tố đặc trưng của tựa game và cập nhật một cách liên tục. Nhóm sẽ



tiếp tục nỗ lực nghiên cứu và cải thiện hơn để tạo ra một sản phẩm đem lại giá trị cao cho phát triển game nói riêng và nền công nghiệp Game và giải trí nói chung.