

**SENG 462**  
**Machine Learning in Game**  
**Programming**

**Project Report**

Eyüp Öztürk

201928026

Machine Learning with Game Sales

**Instructor**

Prof.Dr. Mehmet Reşit Tolun

## **1. Executive Summary**

For my project, I used a dataset that lists video games that have sold over 100,000 copies. The size of this dataset is small compared to the others, due to the fact that there are not many video games with these many sales.

The dataset includes the following sections:

- Rank - Ranking all sales
- Name – Games name
- Platform – Platform of the Games (PC, Mobile, PS4)
- Year – Release Year of the games
- Genre- Type of the game
- Publisher – Publisher of the game
- NA\_Sales – in North America
- EU\_Sales – in Europe
- JP\_Sales – in Japan
- Other\_Sales -in the Rest of the world
- Global\_Sales – Worldwide

The best-selling game in this dataset is Wii Sports, which is in the sport genre. This game, which was put up for sale in 2006, has sold approximately 82.74 million kopecks worldwide.

In addition, the company that sells the most games in this data set is Nintendo.

## 2. Benchmarking of Other Solutions

I have a lot of examples to compare the makers of this dataset because the example I chose was so common. That's why I examined the projects of 3 different people who were worse than me, like me and better than me, according to my own transactions.

1)

I think the first project did it with less processing than I did because he didn't use any visual data in his project, and I don't think he explained the processes very well.

Although it does a lot of specific operations in the notebook, it seems that they are separate from each other because it does not customize them.

he generally used the operations with the data, not the data itself, for example he used methods such as `s.sort_index()`, `s.sort_values(ascending=False)`

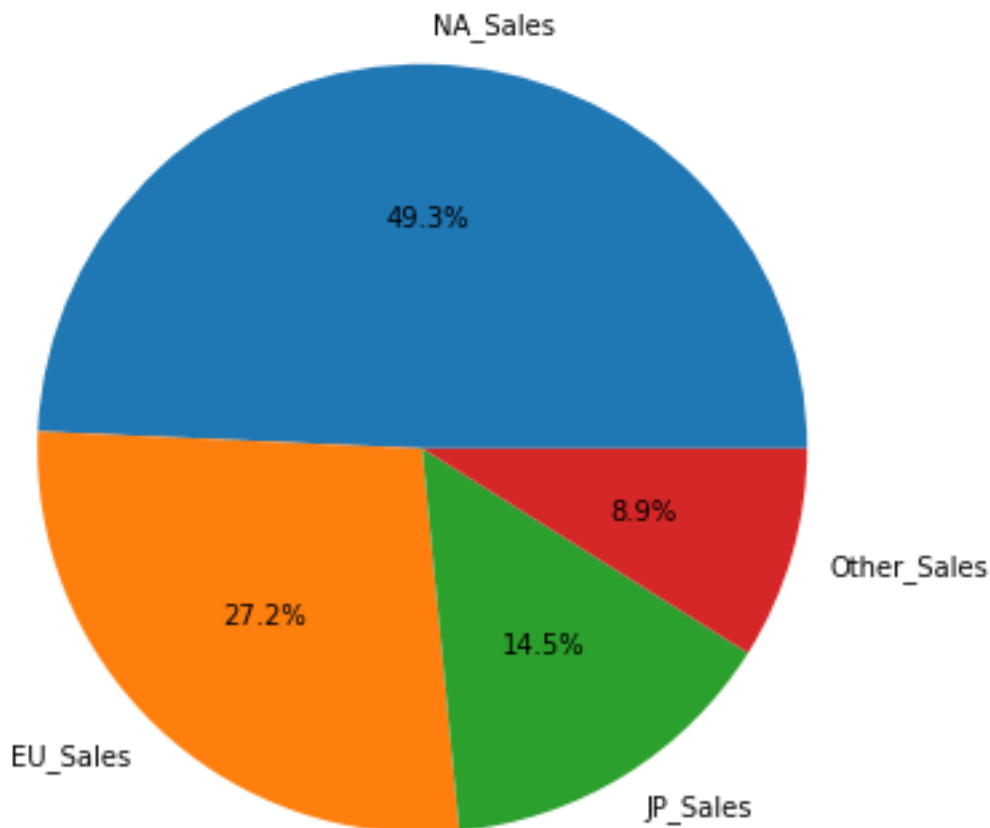
url:

<https://www.kaggle.com/code/sweetyvinayakmore/series-part-1>

2)

The second project has almost the same qualities as the project I have done, enriching its operations with visuals. He demonstrated well the data he used and the relationships between these data. Since it processes the values of the data set in general, not specifically, the operations are in a way

that almost everyone can easily read. for example, he created a pie chart and thus an overall chart of sales



url: <https://www.kaggle.com/code/gjquimbo/video-games-sales/notebook>

**3)**

The last project is a very well-prepared project, in my opinion, the analyzes he made, the data operations he used and the visualizations were all very well designed. Since it uses both special operations and general operations, it helps us to fully understand the operations performed in the data set, and it has been an easy-to-understand project as it graphs each specific operation.

url: <https://www.kaggle.com/code/aslamahmed07/sales-analysis-video-games-sales-dsml-project>

### **3. Data Description and Initial Processing**

The data analyzes are not very diverse, as the dataset I chose does not focus on a fundamental piece. Comparisons to be made will be at the basic level and in a visualized form.

- There are 11493 unique values in the dataset
- The platforms with the highest number of these values are DS and PS2 with 13%.
- In the dataset, the years with the highest number of game releases between 2008 and 2009.
- The genre of best-selling games is action and sports.
- The top selling publisher is EA (8%), followed by Activision with 6%.
- The sales volumes in the sales regions are as follows:

North America: 41.5M

Europe: 29M

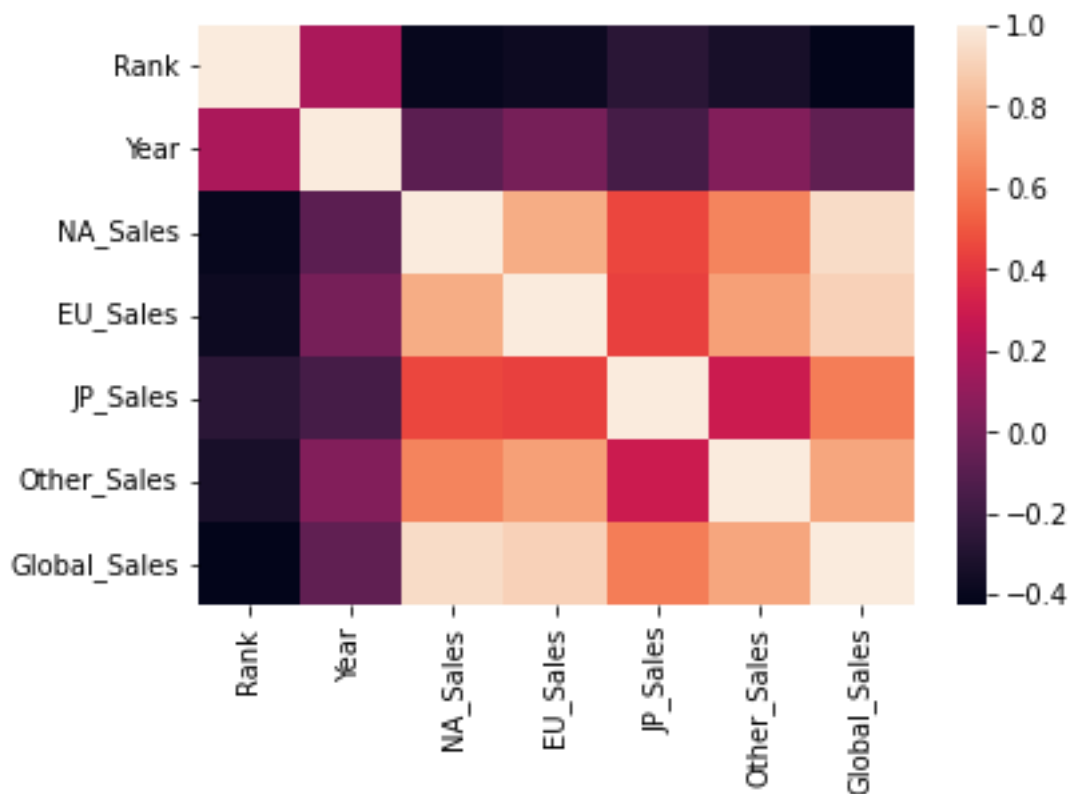
Japan: 10.2M

Other: 10.6M

The best way to understand the relationships between the values in the data set is to find the correlation coefficients.

One of the best visualizations for this is to make a heatmap using the `heatmap()` function.

HeatMap:

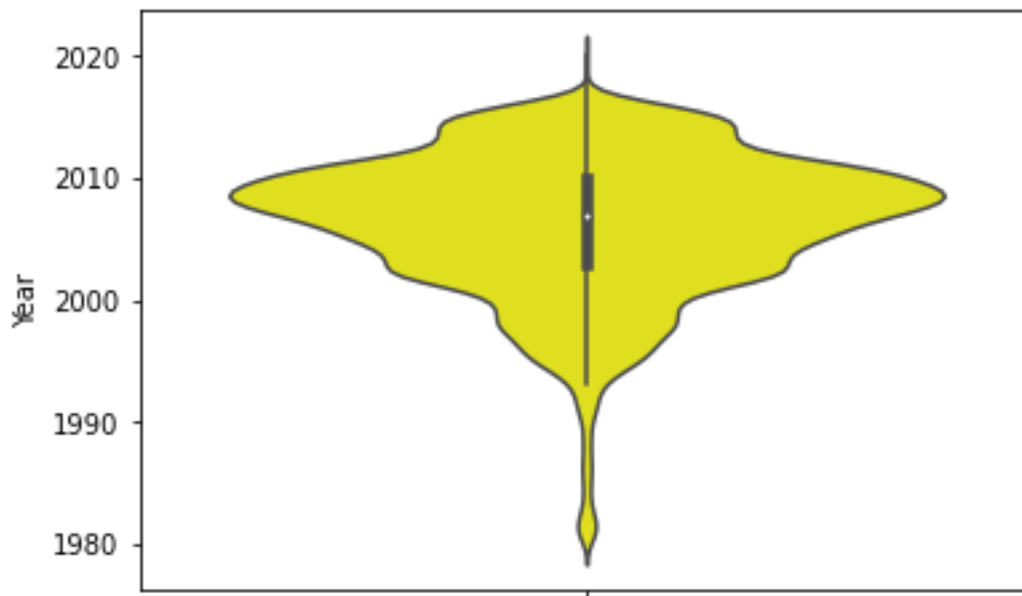


In the heatmap, dark colors represent the distant relationship to each other, while light colors show the excess of relationship to each other. For example, the relationship between sales in North America and Global sales seems to be the highest.

Another graphic that makes it easier to read the data in the data sets is the violin graph. The maximum point of the curve in the graph is the arithmetic mean, and this curve must be

symmetrical. We can create this graph with the `violinplot()` method.

Violin graph:

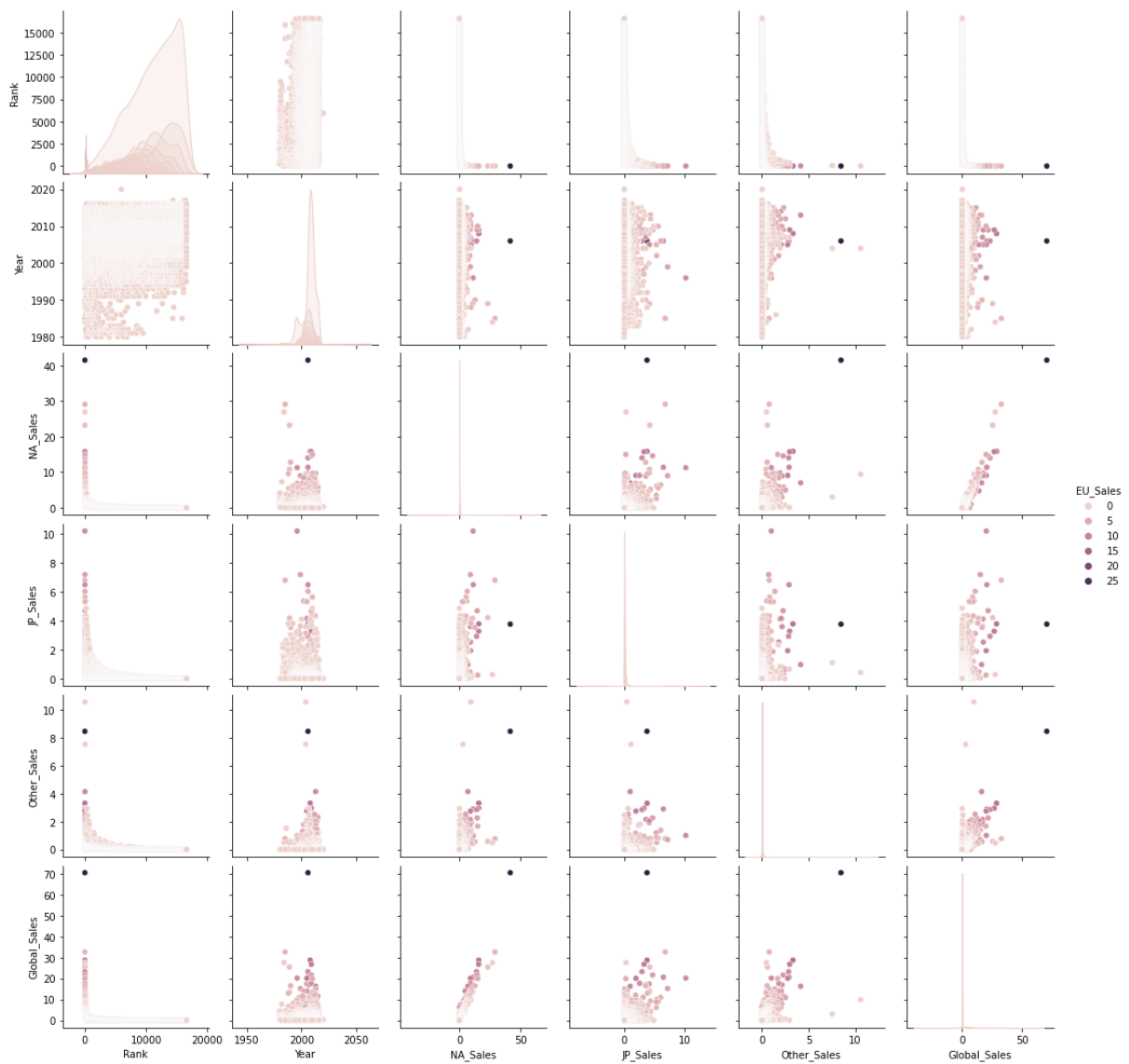


The chart above shows the changes in sales of video games over the years. This chart shows that sales from 2008 or 2009 are the most

Another graph is the pairplot graph. Actually, this chart shows a summary of all the data. It is created by charting the relationships of all the data in the data set with each other one by one. It is created with the `pairplot()` method.

The most correlated data should occupy a less dense area.

## Pair Plot Graph:





## **4. Modeling**

For modelling, I examined the sales of video games in Europe, North America, and Japan in the dataset. I will show you the amount of game sales in these regions, which publisher company is better in which region, which game genre is very popular, and in which years they had the best sales.

Wii Sports, the most sold game, won the first place in North America and Europe, while the game Pokemon red / Pokemon Blue, made by the same publisher company Nintendo in Japan, became the first with 10.22M sales.

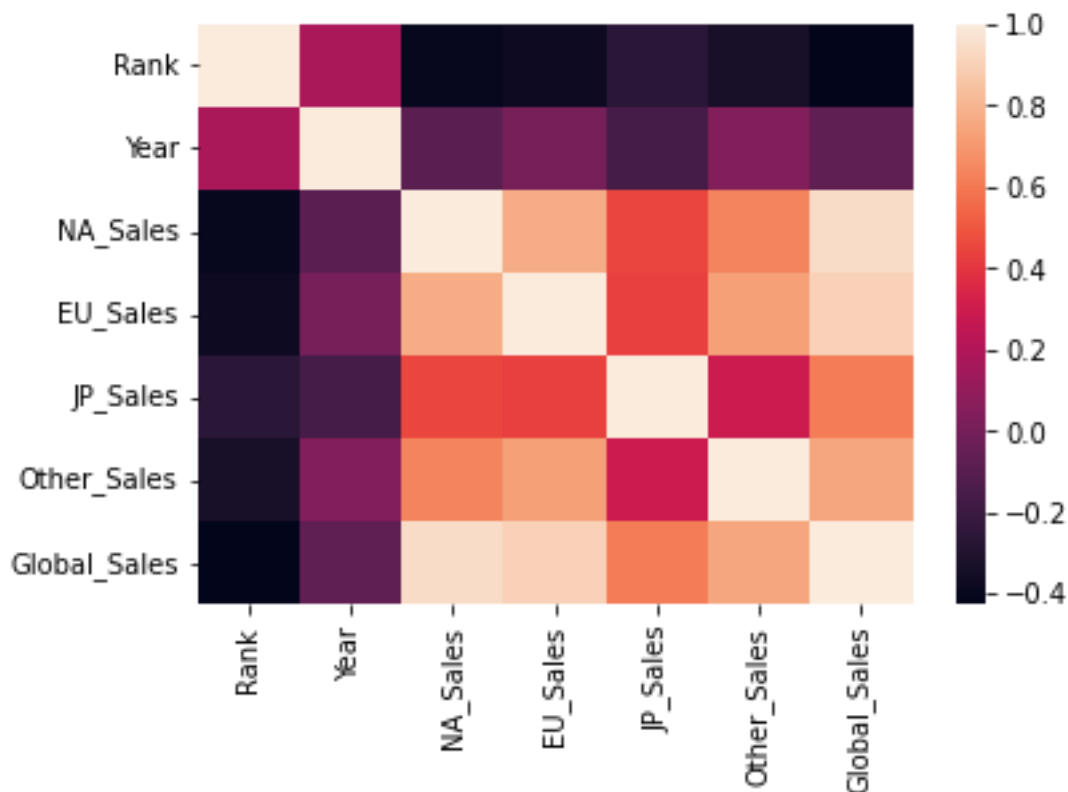
Standard deviations in game sales were different from each other.

North America: 0.8

Europe: 0.5

Japan: 0.3

These data show that North America is more stable



As I said before, it would be most logical to use a heat map to better understand the relationships of these 3 data with each other.

The North America - Europe relationship corresponds to a value of approximately 0.8

The North America - Japan relationship corresponds to a value of approximately 0.5

The Europe-Japan relationship corresponds to a value of approximately 0.3

These data show that Europe and North America have very similar sales transactions, while Japan has very different sales transactions.

## **5. Appendix**

Github url:

<https://github.com/softflied/MachineLearning.git>