Video Game Sales

Kholood Alhejori

T5 Data Science BootCamp

Project Target

- Which Region Will affect the global sale?

- Tools
- Data processing :Pandas , Numpy.
- Modelling: Scikit-learn
- Visualizations Matplotlib and Seaborn
- Algorithm
- Cleaning the data using EDA process
- For modelling Linear Regression

Dataset Description

- This project is one of the T5 Data Science BootCamp requirements. Data provided by Kaggle has been used in this project. The dataset is provided in .csv format.This dataset contains a list of video games with greater sales. It contains 16,598 records, each record has 11 features. The most relevant feature to this project is Total worldwide sales.This feature is extracted from other features such as Sales in North America,Sales in Europe,Sales in Japan, and Sales in the rest of the world.

Data Definition

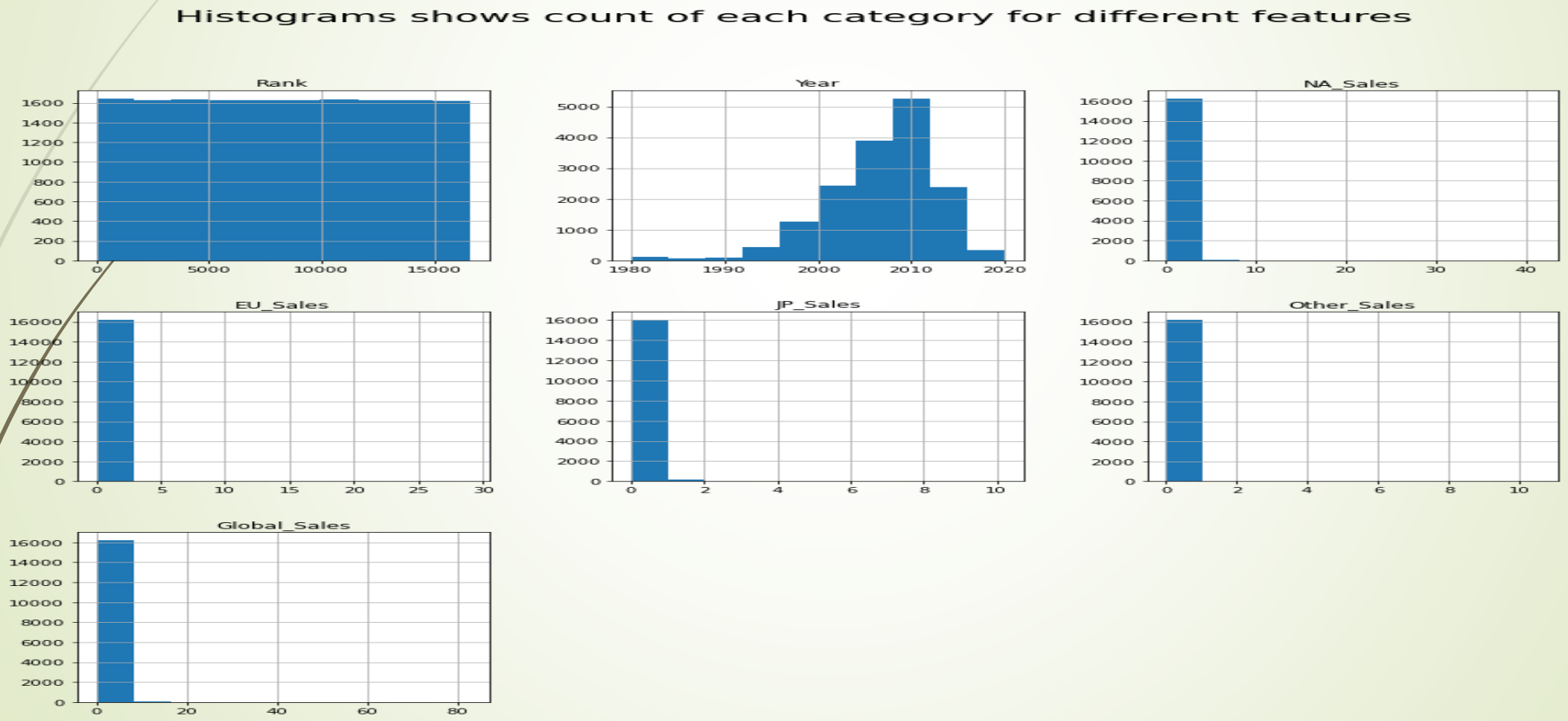
- Fields include
- • Rank - Ranking of overall sales
- • Name - The games name
- • Platform - Platform of the games release (i.e. PC,PS4, etc.)
- • Year - Year of the game's release
- • Genre - Genre of the game
- • Publisher - Publisher of the game
- • NA_Sales - Sales in North America (in millions)
- • EU_Sales - Sales in Europe (in millions)
- • JP_Sales - Sales in Japan (in millions)
- • Other_Sales - Sales in the rest of the world (in millions)
- • Global_Sales - Total worldwide sales.

Data Definition

```
dataset.info()#Gives a basic overview of all the columns present in our dataset.
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 16598 entries, 0 to 16597
Data columns (total 11 columns):
 #   Column        Non-Null Count  Dtype  
---  ------        --------------  -----  
 0   Rank          16598 non-null  int64  
 1   Name          16598 non-null  object 
 2   Platform      16598 non-null  object 
 3   Year          16327 non-null  float64
 4   Genre         16598 non-null  object 
 5   Publisher     16540 non-null  object 
 6   NA_Sales      16598 non-null  float64
 7   EU_Sales      16598 non-null  float64
 8   JP_Sales      16598 non-null  float64
 9   Other_Sales   16598 non-null  float64
 10  Global_Sales  16598 non-null  float64
dtypes: float64(6), int64(1), object(4)
memory usage: 1.4+ MB
```

Data Definition



Data Cleaning

```
dataset.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 16598 entries, 0 to 16597
Data columns (total 11 columns):
 #    Column          Non-Null Count   Dtype
---   ------          --------------   -----
 0    Rank            16598 non-null   int64
 1    Name            16598 non-null   object
 2    Platform        16598 non-null   object
 3    Year            16327 non-null   float64
 4    Genre           16598 non-null   object
 5    Publisher       16540 non-null   object
 6    NA_Sales        16598 non-null   float64
 7    EU_Sales        16598 non-null   float64
 8    JP_Sales        16598 non-null   float64
 9    Other_Sales     16598 non-null   float64
 10   Global_Sales    16598 non-null   float64
dtypes: float64(6), int64(1), object(4)
memory usage: 1.4+ MB
```

check our data is clean or not

Data Cleaning

```
dataset.isnull().values.any()# Checking for null values in the dataset
```

```
9]: True
```

```
dataset.isna().sum()# Checking which columns contain null values
```

```
1]: Rank            0
    Name            0
    Platform        0
    Year          271
    Genre           0
    Publisher      58
    NA_Sales        0
    EU_Sales        0
    JP_Sales        0
    Other_Sales     0
    Global_Sales    0
    dtype: int64
```

Above results show that out of 11 variables, 2 variables Year and Publisher have missing values. There are aroud 271 records in Year column and around 58 records in Publisher column whoch are NA Value records.

Data Cleaning

Let's check the percentage of the data are missing column

```
dataset.isna().sum()/(len(dataset))*100
```

```
2]: Rank            0.000000
    Name            0.000000
    Platform        0.000000
    Year            1.632727
    Genre           0.000000
    Publisher       0.349440
    NA_Sales        0.000000
    EU_Sales        0.000000
    JP_Sales        0.000000
    Other_Sales     0.000000
    Global_Sales    0.000000
    dtype: float64
```

Since the % of the data missing is very less, I can remove those rows from the dataset

```
dataset = dataset.dropna(axis=0, subset=['Year','Publisher'])# Removing the missing value rows in the dataset
```

```
dataset.isnull().values.any()# checking the missing value it Removing or not
```

```
4]: False
```

Data Cleaning

```
: 
# Selecting duplicate rows except first occurrence based on all columns
duplicate = dataset[dataset.duplicated()]

print("Duplicate Rows :")

# Print the resultant Dataframe
duplicate
```

```
Duplicate Rows :
```

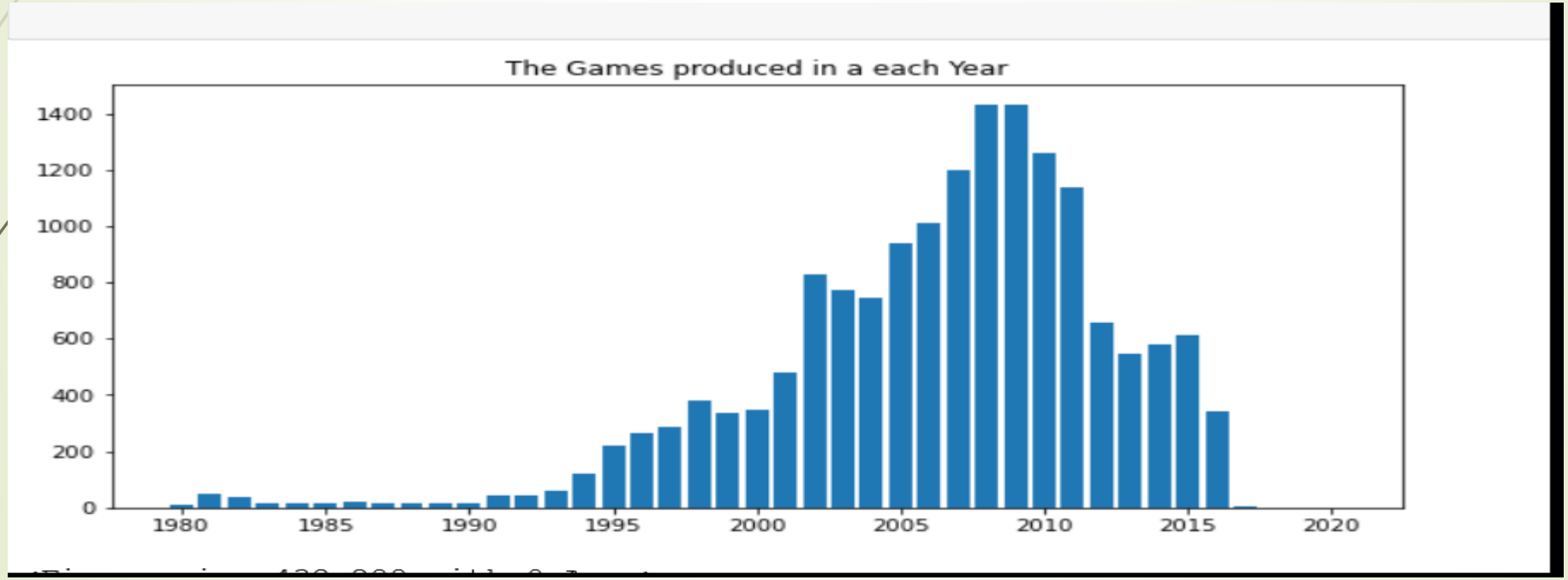
[17]:

Rank	Name	Platform	Year	Genre	Publisher	NA_Sales	EU_Sales	JP_Sales	Other_Sales	Global_Sales

As output show no Duplicate Rows

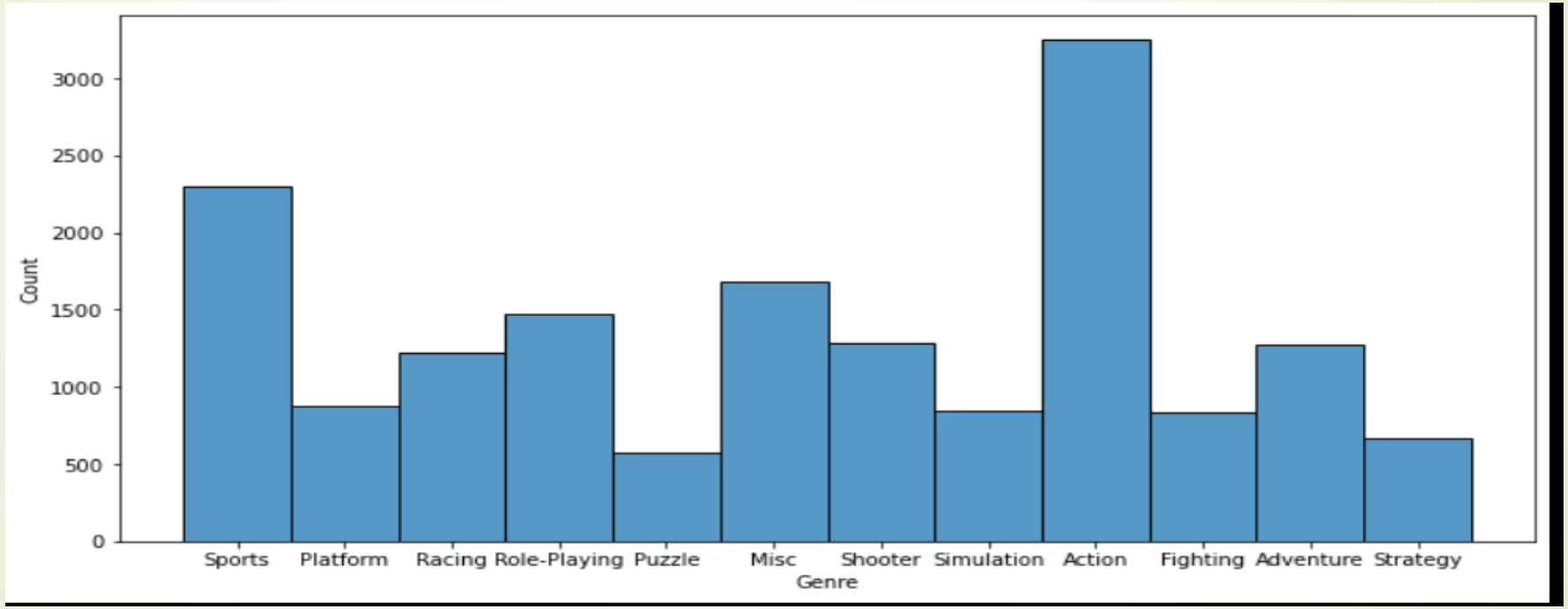
Questions for Analysis

- The video games produced in an each year.



Questions for Analysis

- Which genre sold the most globally?



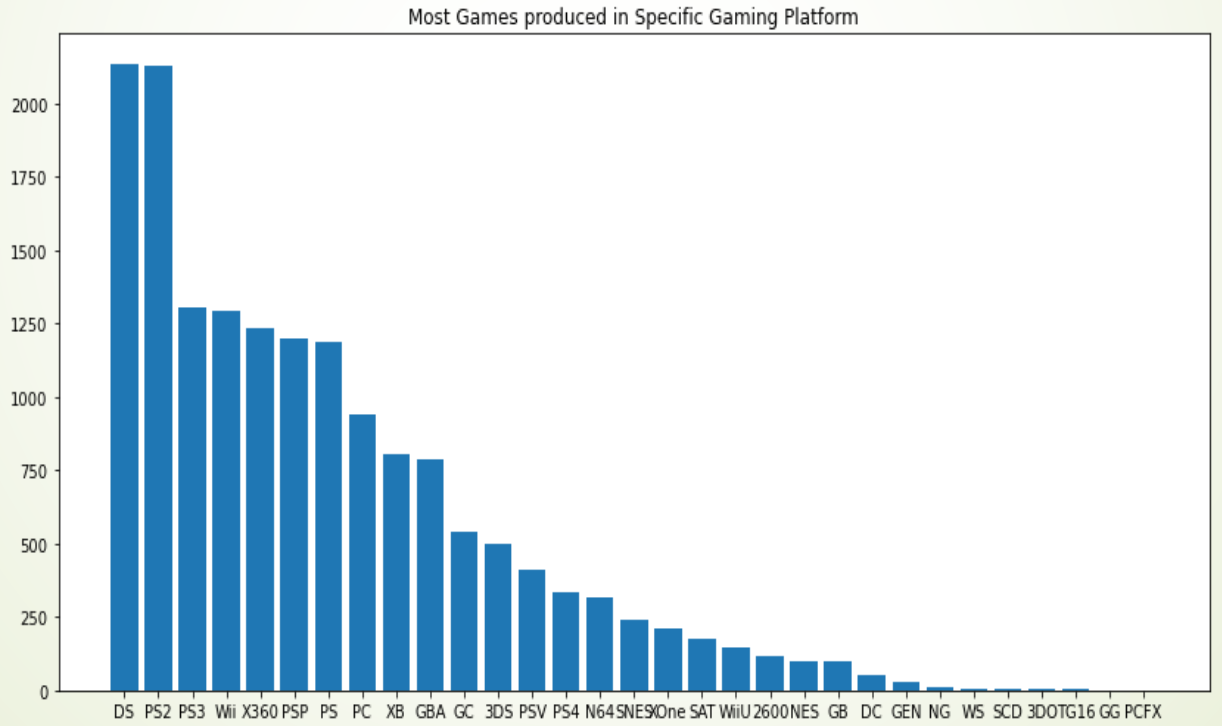
Questions for Analysis

- The platform,game and the publisher which has the top sales in global sales

	Rank	Name	Platform	Year	Genre	Publisher	NA_Sales	EU_Sales	JP_Sales	Other_Sales	Global_Sales
0	1	Wii Sports	Wii	2006.0	Sports	Nintendo	41.49	29.02	3.77	8.46	82.74

Questions for Analysis

- What are most games produced in a specific Gaming Platform?

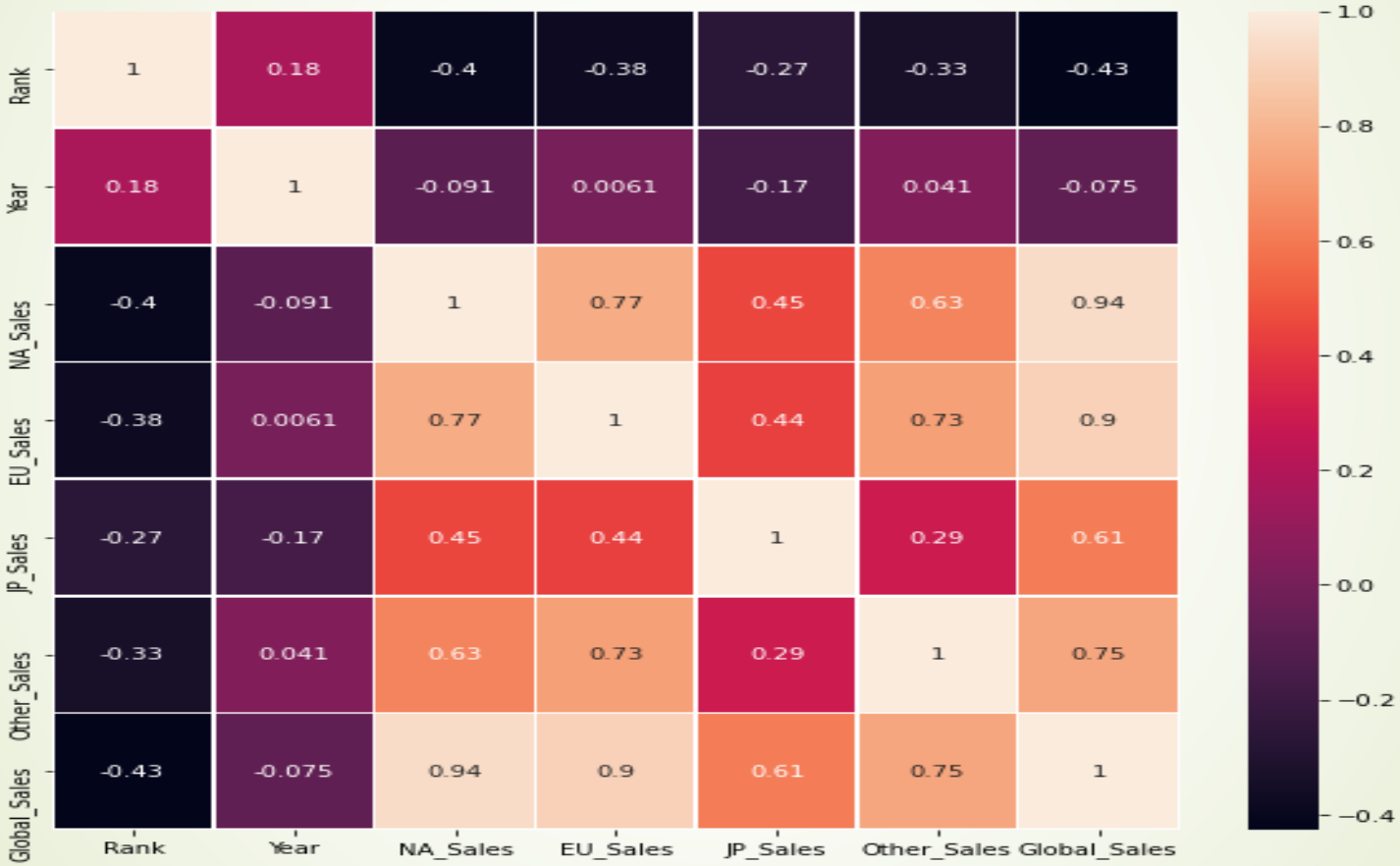


Picking the model

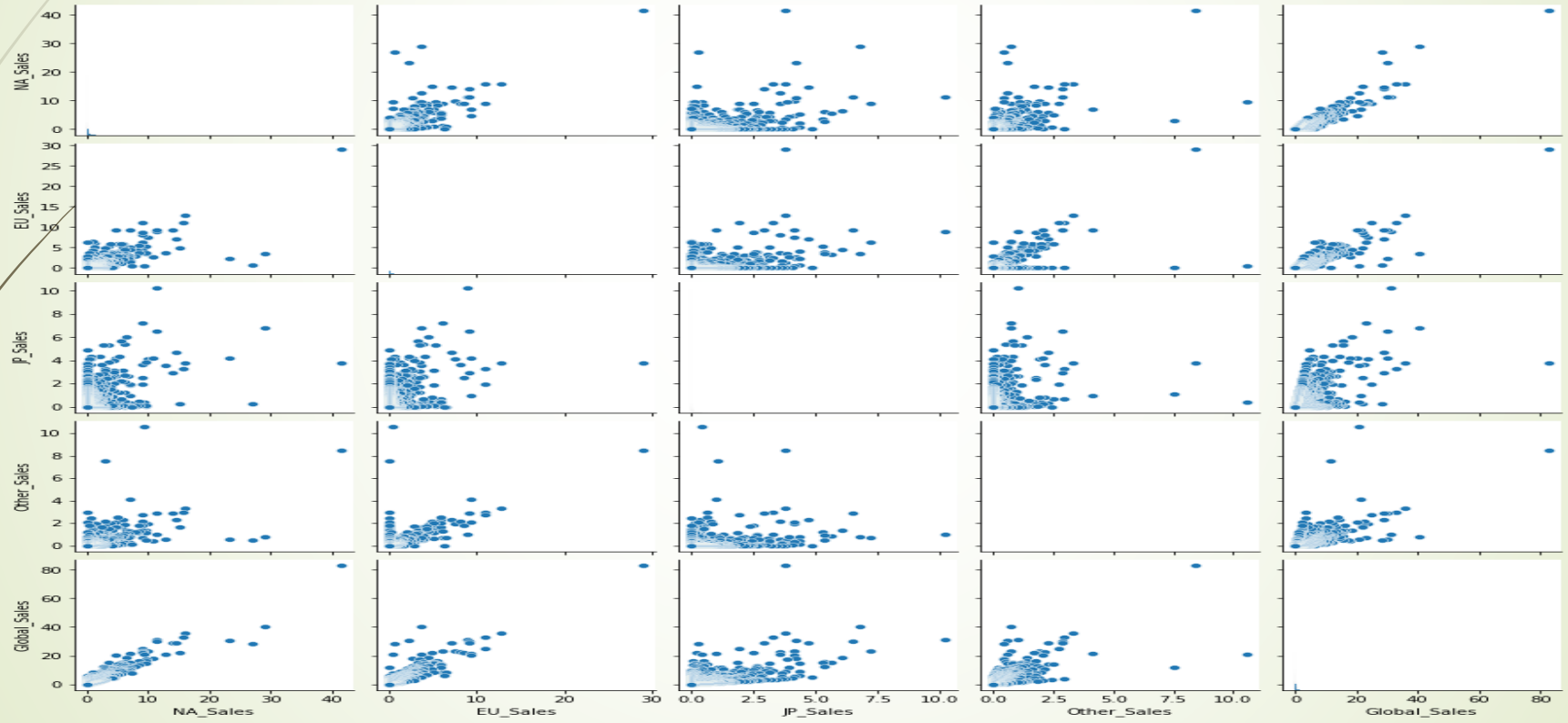
- Determining the relevancy of features using correlations,heatmap and pairplot

	Rank	Year	NA_Sales	EU_Sales	JP_Sales	Other_Sales	Global_Sales
Rank	1.000000	0.178027	-0.400315	-0.379137	-0.269323	-0.332735	-0.426975
Year	0.178027	1.000000	-0.091285	0.006108	-0.169387	0.041128	-0.074647
NA_Sales	-0.400315	-0.091285	1.000000	0.768923	0.451283	0.634518	0.941269
EU_Sales	-0.379137	0.006108	0.768923	1.000000	0.436379	0.726256	0.903264
JP_Sales	-0.269323	-0.169387	0.451283	0.436379	1.000000	0.290559	0.612774
Other_Sales	-0.332735	0.041128	0.634518	0.726256	0.290559	1.000000	0.747964
Global_Sales	-0.426975	-0.074647	0.941269	0.903264	0.612774	0.747964	1.000000

Picking the model



Picking the model



modelling Linear Regression

```
#adjusted_r_squaredr for training set
adjusted_r_squared = 1 - (1-Sales_R.score(X_train,y_train))
adjusted_r_squared
```

```
: 0.9999894120081414
```

```
#r_squared for testing set
r2_score(y_test, predictions)
```

```
: 0.9999863067614225
```

```
#adjusted_r_squaredr for testing set
adjusted_r_squared_1 = 1 - (1-(r2_score(y_test, predictions
adjusted_r_squared_1
```

```
: 0.9999863033982299
```

Thank you