Technical Assessment Report

– PYGASAL –
VIDEO -GAME SALES PREDICTION USING PYTHON

Promotion: August 2021
Mentor: Gaspard GRIMM
Participants: Ammar HASAN
Cristóbal GALLARDO

Report date: 14 October 2021

Contents

1. Introduction

The gaming industry is definitely one of the booming industries in the modern era and one of the industries most affected by technological progress. With technologies such as augmented reality (AR) / virtual reality (VR) available in consumer products such as game consoles and even smartphones, the gaming market is showing great potential.
In Data mining, we as data scientists should use our analytical skills to predict video game sales based on certain factors (input features). There are 10 distinct factors that can affect video game 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
1	2	Super Mario Bros.	NES	1985.0	Platform	Nintendo	29.08	3.58	6.81	0.77	40.24
2	3	Mario Kart Wii	Wii	2008.0	Racing	Nintendo	15.85	12.88	3.79	3.31	35.82
3	4	Wii Sports Resort	Wii	2009.0	Sports	Nintendo	15.75	11.01	3.28	2.96	33.00
4	5	Pokemon Red/Pokemon Blue	GB	1996.0	Role-Playing	Nintendo	11.27	8.89	10.22	1.00	31.37

Figure 1 - Database extract

During our research, we have found three papers that can be considered similar to the problem treated in this report, even though they refer to the film industry. See section 8 for further details.

2. Objective

Our goal as data scientists is to build a machine learning model that can accurately predict sales in millions of units for a given game. If we have the time, we may divide the predictions in world regions.

Additional databases can be extracted from Metacritic or jeuxvideo.com. BeautifulSoup library in Python will be used for web scraping purposes to extract the data out of Metacritic or jeuxvideo HTML and XML files. It creates a parse tree from page source code that can be used to extract data in a hierarchical and more readable manner.

3. Data

3.1 Data Structure

The database for this project will be limited to the database https://www.kaggle.com/gregorut/videogamesales

- containing a list of video games with sales greater than 100 000 copies. The database was created 26-Oct-2016 and it contains 16598 entries (see Figure 1) considering the following list of variables :
- 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 of copies)
- EU_Sales - Sales in Europe (in millions of copies)
- JP_Sales - Sales in Japan (in millions of copies)
- Other_Sales - Sales in the rest of the world (in millions of copies)
- Global_Sales - Total worldwide sales.(in millions of copies).

An analisis of the dataset (see annex 10.1) showing that the majority the games are released between years 2000 to 2016 (see Figure 2). The dataset is incomplete for the year after 2017.

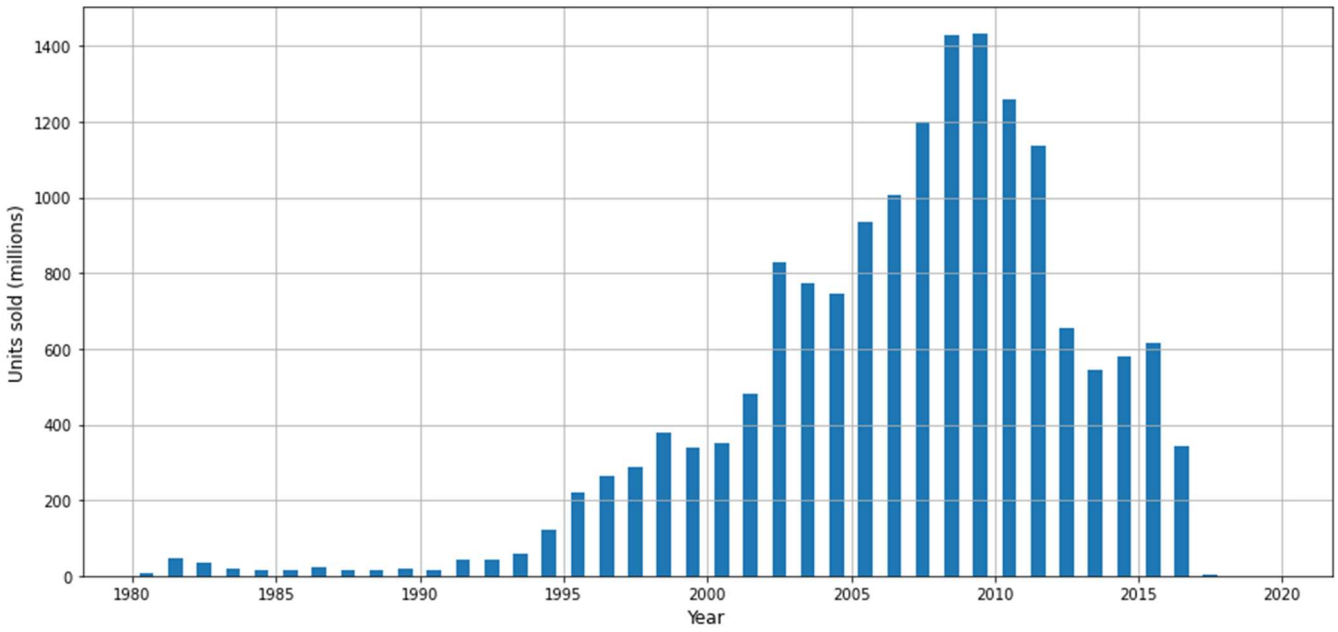


Figure 2 - Number of titles released per year

3.2 Data Completion

Unfortunately, the data we currently have only contains the total sales of the game, the studio, the country and the publisher which not enough to build a prediction model, so we need to gather extra factors/features that reflect the importance of each game from the view point of cutomers and users. Such information can be find on sites such as Metacritic or jeuxvideo.com.
Metacritic.com is a site that aggregates professional reviewer scores from video games (various platforms). Metacritic calculates an average score called Metascore, based on the various professional reviewers by converting the reviewers' local score into a score of 0 to 100 (e.g. a local score of 8 out of 10 renders a Metascore of 80). These scores are weighted (based on the quality and overall stature of the source) and finalized into a professional Metascore.
Regular non-professional users are also allowed to score the media on a scale of 0 to 10. The unweighted average of this score is presented by Metacritic as the Userscore. Non-professional users can also post their own reviews along with their score.
The user score is divided into three tiers: Positive, Neutral and Negative, where Positive is ratings 8 to 10, Neutral is ratings 5 to 7, and Negative is ratings 0 to 4. The rating tiers are color coded in green for Positive, yellow for Neutral and red for Negative. We will proceed scraping these ratings from the Metacritic website (https://www.metacritic.com).

3.2.1 Web Scraping

A notebook preparing for scraping the metacritic's webpage has been preprared using the library bs4 from BeatifulSoup and urllib.request (see annex 0).
When grouping by year A first look at the webpage, we have noticed critics are made for games released from 1996 to 2021. Typically, the webpage lists their database with an url as follows:
https://www.metacritic.com/browse/games/score/metascore/year/all/filtered?year_selected=2020&distribution=&sort=desc&view=detailed&page=5

Scraping is a long process. To avoid overloading the scrapped page with our requests, we have carried it out by 5-year period creating thus databases df0 to df4. Using a "for" loop, each url has been called upon by the algorithm for scraping. We have also used the sleep function from the time library to limit the impact of Metacritic's server.
To select the data within the webpage, we have used SelectorGadget for Chrome. This powerful tool has allowed us to select for each title the platform, year (of release), Meta score and User score for the 18817 titles in Metacritic's database.
We have combined our scraping data with other data scraped by other people found on Kaggle. The final database has a volume of 50 000 rows with after dropping the duplicated lines and the NaNs amount to the final 40 000 lines.

3.2.2 Databases Merging

The databases vgsales and Meta_vg will be merged so that all data is available in a single set for later operations. Given that the same title may be available for different platforms, we will use the triple criteria of matching columns “Name”, “Platform” and “Year” to select the rows to join. To maximize the number of matches, we previously proceed to normalize the fields “Name” and “Platform” as NFKD (to remove tildes, dieresis, …) and enforce uppercasing. See annex 0 for further details.

Our resulting database is named “VG_Meta_Score.csv” and has 5044 rows and 13 columns.

	Rank	Name	Platform	Year	Genre	Publisher	NA_Sales	EU_Sales	JP_Sales	Other_Sales	Global_Sales	Meta_Score	User_Score
0	1	WII SPORTS	WII	2006.0	Sports	Nintendo	41.49	29.02	3.77	8.46	82.74	76	8.1
1	3	MARIO KART WII	WII	2008.0	Racing	Nintendo	15.85	12.88	3.79	3.31	35.82	82	8.4
2	4	WII SPORTS RESORT	WII	2009.0	Sports	Nintendo	15.75	11.01	3.28	2.96	33.00	80	8.2
3	7	NEW SUPER MARIO BROS.	DS	2006.0	Platform	Nintendo	11.38	9.23	6.50	2.90	30.01	89	8.5
4	9	NEW SUPER MARIO BROS. WII	WII	2009.0	Platform	Nintendo	14.59	7.06	4.70	2.26	28.62	87	8.3

Figure 3 – Extract from VG_Meta_Score (merged database)

3.3 Data cleaning and methodology.

A look at the dataset confirms that there are non NaN values.

A correlation analysis between the variables (see Figure 4) shows that the NA_Sales, EU_Sales, JP_Sales, and Other_Sales are the highest correlated with target variable Global_Sales, and even between each other. These higher correlations between sales variables help us to understand the global nature of the video game industry, so success on one continent usually means success on another as well. This finding support selecting the Global_Sales variable to be used as a dependent variable. There is also very weak correlation between Year and the other numeric variables.

Figure 4 - Correlation matrix from appendix 0

3.3.1 Methodology

- For a first approach, numeric variables NA_Sales, EU_Sales, JP_Sales, and Other_Sales will be disregarded. We will keep only “Global Sales” as it is our target variable.
- We will proceed later with a second approach where NA sales will be taken into account. This will cover the case where a game is released in one region ahead of the release in another region. We expect this will increase the score of the predictions.

- Finally, a further study will be performed where alternative modelling options will be presented that may improve these scores.

Rank variable is also dropped as meaningless in the context of this database.
We also proceed dropping the variable "Publisher". This is a categorical variable with 204 unique values. We will apply a get_dummies method to the dataframe and keeping this categorical variable will generate a high number of columns that will be generate unnecessary processing time for this iteration.

3.3.2 Numerical Variables.

A quick look at the units sold in the database confirms the removal of all data before year 1996. We can also see we do not have data available from year 2018 onwards.

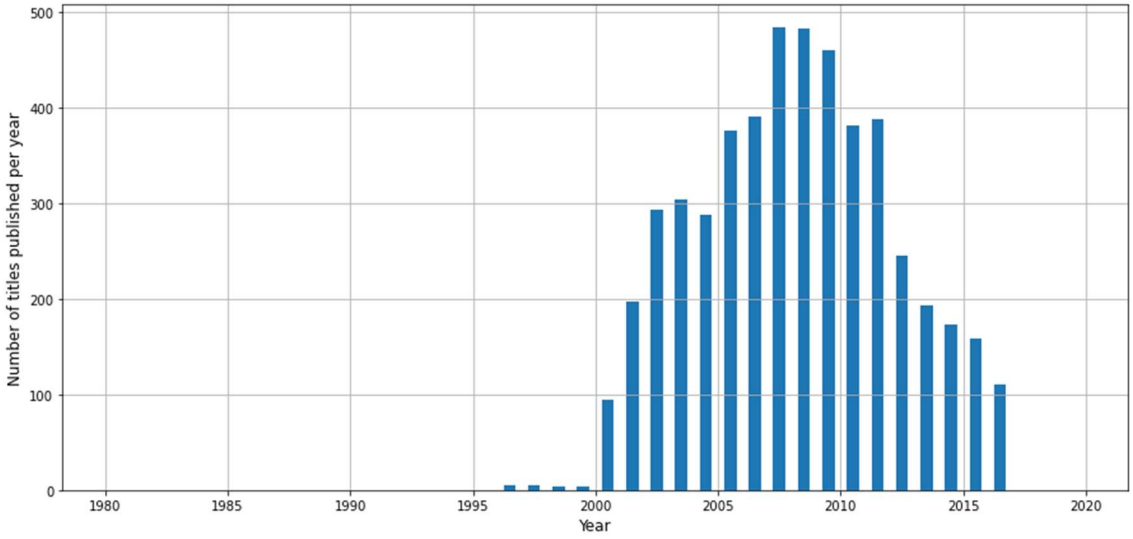


Figure 5 – Number of games released per year in final database

It is worth to note that figure above shows the units sold of a title per year of game release. The fact that there is a change of tendance between year 2008 and 2009, and that data is available up to year 2017, indicates the sales cycle of a game is around 8-9 years (to be confirmed by further study).

3.3.3 Categorical Variables.

The rest of the variables (Platform and Genre) are categorical. They count 16 and 12 unique values respectively. We will plot the count of the number of titles against each categorical variable to better understand their relations. Please refer to appendix 0 for further details.

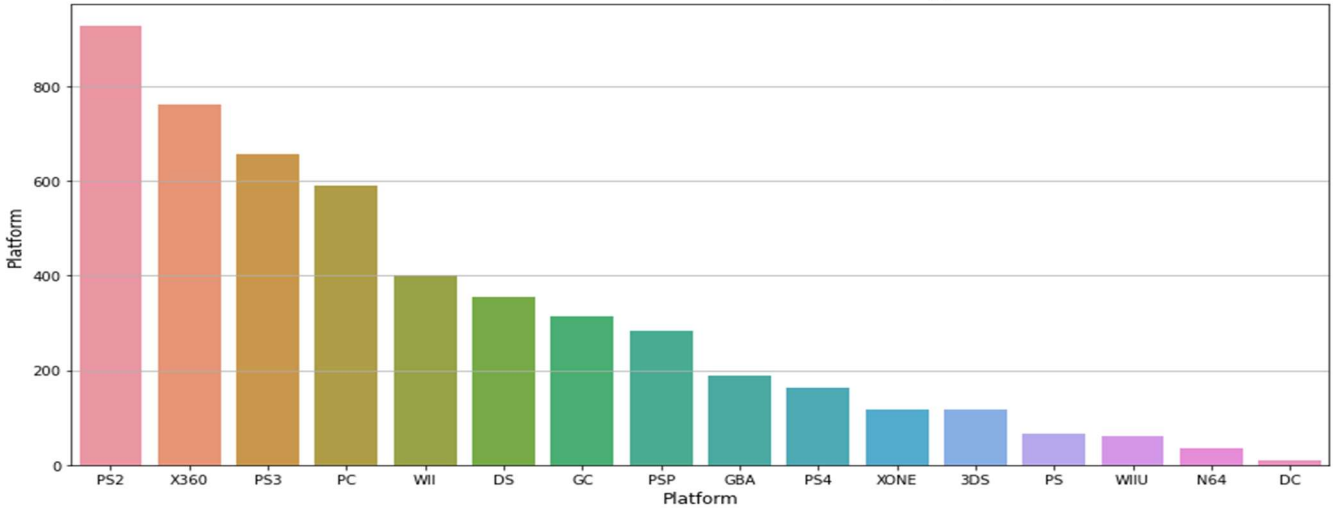


Figure 6 - Number of games released grouped by platform

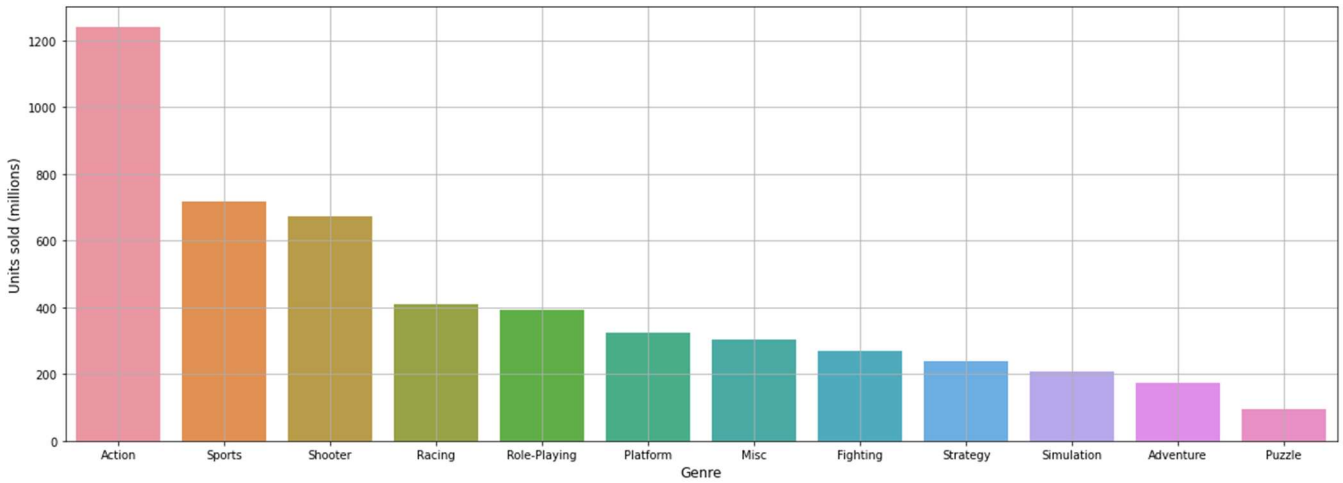


Figure 7 - Number of games released grouped by genre

4. Initial Approach

4.1 Description

Our target is to predict global sales of a video game which is a continuous variable. This limits our model choice to Regression model. To enlarge the number of models available we will also try to transform the Regression problem into a classification problem (Logistic Regression).

4.1.1 Data Preprocessing:

As we have noted above, no NaNs are present and all data is ready. We have therefore proceeded with the following steps (see 0 for further details):

- Apply the *pd.get_dummies* function to the categorical features (Platform and Genre).
- Divide the database between the target variable and the feature variables.
- As values range at very different scales for all variables, apply Standard Scaler to normalize them:
- Proceed with a train-test split of 20%.
- Proceed applying a logarithmic transformation to the target label: This improves results.

4.1.2 Linear Regression Model:

The above set has been trained using a liner regression model with the following results:
score train : 0.27
score test : 0.28
mse test: 0.37

4.2 Model Selection and Optimisation

4.2.1 Gradiant Boosting Regressor with GridSearchCV

We have performed a Gradient Boosting Regressor algorithm to the data together with a grid search to find the parameters that yield the best results. We have tried the following grid:

- Maximum number of features: 6, 8, 10, 12.
- Maximum depth: 5, 7 and 9.

The best parameters are : {'max_depth': 5, 'max_features': 8}, achieving a test score of 0.57.

4.2.2 Optimisation of the Linear Regression using Lasso.

We have used a Lasso model using different alphas ([10, 1, 0.1, 0.01, 0.001, 0.0005]) to find the best value for alpha. We obtain the best alpha is 0.0005, and that there are 2 variables (out of 31) to be eliminated.
The scores are:
score test: 0.30
mse test: 0.50, showing very little improvement from those obtained in 4.1.2.

4.2.3 Logistic Regression Model:

The set of data has been divided in four bins (quantiles): Silver, Gold, Platinum and Diamond according to the global sales. With a maximum number of iterations of 1000, the score with this approach goes up to 0.484.

4.2.4 SVM Model:

Similarly the data has been trained on a SVM model yielding a slightly lower score of 0.440.

5. Second Approach

5.1 Description

A similar approach to that following in section 4 is followed, but this time we will consider in our models the regional sales. The notebook including the model described in this section can be found in annex 0.

5.1.1 Data Preprocessing:

- NA sales data is this time kept as feature. For the missing values, we will use the SimpleImputer function from sklearn. We will replace numerical data with the mean value for columns ['Critic_Score', 'Critic_Count', 'User_Score', 'User_Count'].
- We will also use the categorical_imputer function from feature_engine.imputation library to replace NAN values in the ['Genre','Rating'] columns by the most frequent value.
- Proceed with a train-test split of 20%.

5.1.2 XGBRegressor model

We implement the regressor using XGBRegressor (where XGB stands for extreme gradient boosting). XGBoost is an ensemble machine learning algorithm based on decision trees similar to the RandomForest algorithm. However, unlike RandomForest that makes use of fully grown trees, XGBoost combines trees that are not too deep. Also, the number of trees combined in XGBoost is more in comparison to RandomForest. Ensemble algorithms effectively combine weak learners to produce a strong learner. XGBoost has additional features focused on performance and speed when compared to gradient boosting.
We have set the parameters for the number of estimators to 200 and the learning rate at 0.08.
We obtain a test score of 0.89, a r2 score of 0.893 and a RMSE of 0.503.

5.1.3 Gradient Boosting Regressor with GridSearch

We have performed a Gradient Boosting Regressor algorithm to the data together with a grid search to find the parameters that yield the best results. We have tried the following grid:

- Number of estimators: 200 and 500
- Maximum number of features: 6, 8, 10, 12.
- Maximum depth: 5, 7 and 9.

The best parameters are : {'max_depth': 5, 'max_features': 12, 'n_estimators': 500}.
We obtain a best cross-validation score of 0.72 and a test score of 0.55.

5.1.4 Linear Regression model:

Similarly to the previous section, the above set has been trained using a liner regression model with the following results:
score train : 0.89
score test : 0.90
mse test: 0.49

5.1.5 Logistic Regression model:

The set of data has been divided in four bins: Silver, Gold, Platinum and Diamond according to the global sales. The test score with this approach is 0.74.

6. Further work to improve the modelling.

6.1 Influence of outliers

Review of the data shows the presence of outliers - see boxplot in figure below:

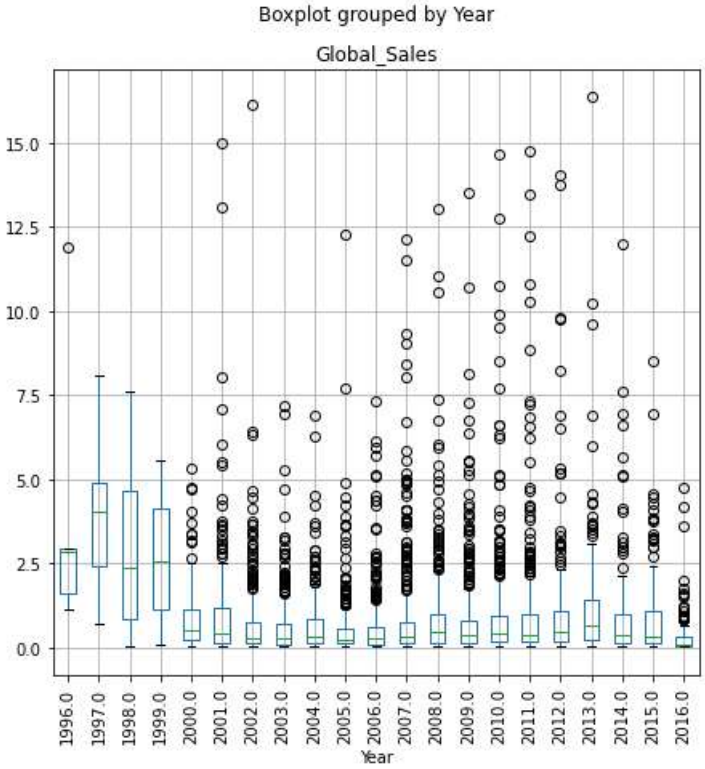


Figure 8 - Boxplot of data grouped by year

The analyses described in previous sections have been repeated by gradually removing the outlyers down to global sales of 8.0 million copies and above. The obtained scored are included in the notebook in section 0 showing that the removal of outliers data does not impact significantly the results.

6.2 Multi-layer back propagation neural network model approach.

We also propose a multi-layer back-propagation neural network model based on (Rhee & Zulkernine, 2016). This work proposes a neural network for the prediction of whether or not a film will become a hit (binary classification). Our case is a 4-class classification meaning that we would expect slightly worse scores.

6.2.1 Neural network model.

We propose a network which features 235 variables across 3 layers:
- 1 input layer of 64 nodes.
- 1 hidden layer of 16 nodes.
- 1 input layer of 4 nodes, one for each category.

Activation function is set to relu except of the output layer which is set to softmax given the class nature of the output. The data has been normalised using a MinMaxScaler which presents better results than the StandardScaler.

6.2.2 Results

Results show that the trained data shows a score of 0.67- However after comparing with the test set, a score of 0.51 is achieved, denoting some degree of overfitting as anticipated. The confusion matrix is shown in Table 1

Classes	0	1	2	3
0	154	70	33	13
1	50	104	51	25
2	32	71	82	79
3	16	20	36	173

Table 1 - Confusion matrix

The figures above show that the algorithm is better to predict low and high sale classes than intermediate classes.

7. Description of tasks carried out

7.1 Repartition of tasks by team member and along the project.

Iteration	Task nr.	Task performed in chronological order	Main contributor
I		Initial data exploration	A. Hasan
		Web scraping – modelling	A. Hasan / C. Gallardo
		Web scraping – reporting	C. Gallardo
II		Github setup and maintenance	C. Gallardo
		First approach - modelling	A. Hasan
		First approach – reporting	C. Gallardo
		First approach – report review	A. Hasan
III		Second approach - modelling	A. Hasan
		Second approach - reporting	C. Gallardo
		Second approach – report review	A. Hasan
		Further study – Influence of outllers	C. Gallardo
		Further study – Neural network model	C. Gallardo
IV		Final reporting	A. Hasan / C. Gallardo
V		Presentation / StreamLit	A. Hasan / C. Gallardo

7.2 Main encountered difficulties

The global sales of a title depend on many subjective parameters and not only on some features related to the games themselves. It also depends on other factors like the audience they are intended for, the time of the year they are released and the political, economic and social characteristics of the market they are released into.

To take into account the above, we have used not only the historical data but also score reviews from specialized websites. But the amount of data is available upto 2016. We would need to complete the data which is not readily available for free.

8. Bibliography

Galvão, M., & Henriques, R. (2018). Forecasting Movie Box Office Profitability. Journal of Information Systems Engineering & Management. *10.20897/jisem/2658*, (p. 3).

Quader, N., Gani, M., & Chaki, D. (2018). Performance evaluation of seven machine learning classification techniques for movie box office success prediction. *10.1109/EICT.2017.8275242*, (pp. 1-6).

Rhee, T., & Zulkernine, F. (2016). Predicting Movie Box Office Profitability: A Neural Network Approach. *10.1109/ICMLA.2016.0117*, (pp. 665-670).

9. Conclusions and further study.

9.1 Conclusions

A summary with the test results for each model we have built is shown in

Situation	Model	Test Score
Initial (regional North American sales unknow)	Linear Regression model	0.28
	Gradiant Boosting Regressor with GridSearchCV	**0.57**
	Linear Regression model with Lasso	0.28
	Logistic Regression (4 bins)	0.48
	SVM model	0.44
Second (regional North American sales known)	XGBRegressor	0.80
	Gradient Boosting Regressor with GridSearchCV	0.55
	Linear Regression model	**0.90**
	Logistic Regression (4 bins)	0.74
Additional (regional North American sales unknow)	Multi-layer back propagation neural network model	0.51

Table 2 - Summary of results

In the case the regional sales for NA is known, then the Linear Regression model works best. In the other hand, when the NA regional sales is unknown, we obtain better results using a Gradiant Boosting Regressor, followed very closely by the multi-layer back propagation neural network model.

9.2 Further study

9.2.1 Twitter counts

It has been found that the number of tweets generated the week preceding the release of a movie is critical to the box office revenue (Rhee & Zulkernine, 2016). This same principle can be applied to video games. We present here two free tools to scrape tweets:

- snscrape: it is a scraper for social networking services like Tweeter, Facebook, Reddit. It scrapes user profiles, hashtags, or searches and returns the discovered items. In 0 we have tried an example with videogame “Red Dead Redemption 2”, tweets 48 hours before release. We collect 6132 tweets.
- twint: is an advanced Twitter scraping tool written in Python that allows for scraping Tweets from Twitter profiles without using Twitter's API. We have tried the same search as above and obtain 60 tweets. A second search yields 100 tweets. We reach the conclusion that this tool is not reliable.

9.2.2 Other options for further study

- Reduce overfitting by dropping features in the neural network.
- Use the name’s title as a feature (e.g. games with the name of Mario or FIFA traditionally sell well).

10. Annexes

10.1 VG_Project.ipynb


A1_VG_Project.ipynb


vgsales.csv

In [ ]:
```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
df= pd.read_csv('vgsales.csv')
df.head()
```

Out[ ]:

	Rank	Name	Platform	Year	Genre	Publisher	NA_Sales	EU_Sales	JP_Sales	Other
0	1	Wii Sports	Wii	2006.0	Sports	Nintendo	41.49	29.02	3.77	
1	2	Super Mario Bros.	NES	1985.0	Platform	Nintendo	29.08	3.58	6.81	
2	3	Mario Kart Wii	Wii	2008.0	Racing	Nintendo	15.85	12.88	3.79	
3	4	Wii Sports Resort	Wii	2009.0	Sports	Nintendo	15.75	11.01	3.28	
4	5	Pokemon Red/Pokemon Blue	GB	1996.0	Role-Playing	Nintendo	11.27	8.89	10.22	

In [ ]:
```
df.info() #show data's 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
```

In [ ]:
```
df.describe()
```

Out[ ]:

	Rank	Year	NA_Sales	EU_Sales	JP_Sales	Other_Sales	Glob
count	16598.000000	16327.000000	16598.000000	16598.000000	16598.000000	16598.000000	1659
mean	8300.605254	2006.406443	0.264667	0.146652	0.077782	0.048063	(
std	4791.853933	5.828981	0.816683	0.505351	0.309291	0.188588	
min	1.000000	1980.000000	0.000000	0.000000	0.000000	0.000000	(
25%	4151.250000	2003.000000	0.000000	0.000000	0.000000	0.000000	(
50%	8300.500000	2007.000000	0.080000	0.020000	0.000000	0.010000	(
75%	12449.750000	2010.000000	0.240000	0.110000	0.040000	0.040000	(

Rank	Year	NA_Sales	EU_Sales	JP_Sales	Other_Sales	Glob

Data cleaning

In [ ]:
```
print(df.isnull().sum()) #check the null values
```

```
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
```

From the code above, seen that there are two variables have many null values. Year and Publisher varaiables have 271 and 58 null values, respectively.

While our task is to predict the Global_Sales, so We ignore the null values.

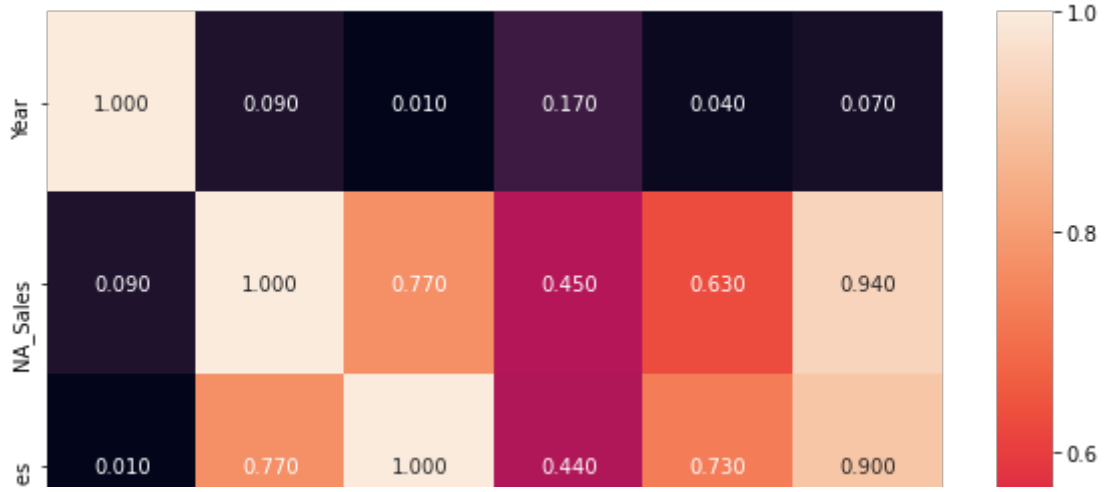
Then we can move to the next step of removing unnecessary columns. From dataset, we can observe that Rank column has no meaning in our dataset, so we remvoe it from the dataset

In [ ]:
```
df.dropna(inplace=True)
df.drop('Rank', axis=1, inplace=True)
#df.drop(columns =['Rank','NA_Sales', 'EU_Sales', 'JP_Sales', 'Other_Sales'
```

Variables Correlation

In [ ]:
```
plt.figure(figsize=(10,10))
sns.heatmap(df.corr().abs().round(2),annot=True,fmt= '.3f')
```

Out[ ]: <AxesSubplot:>



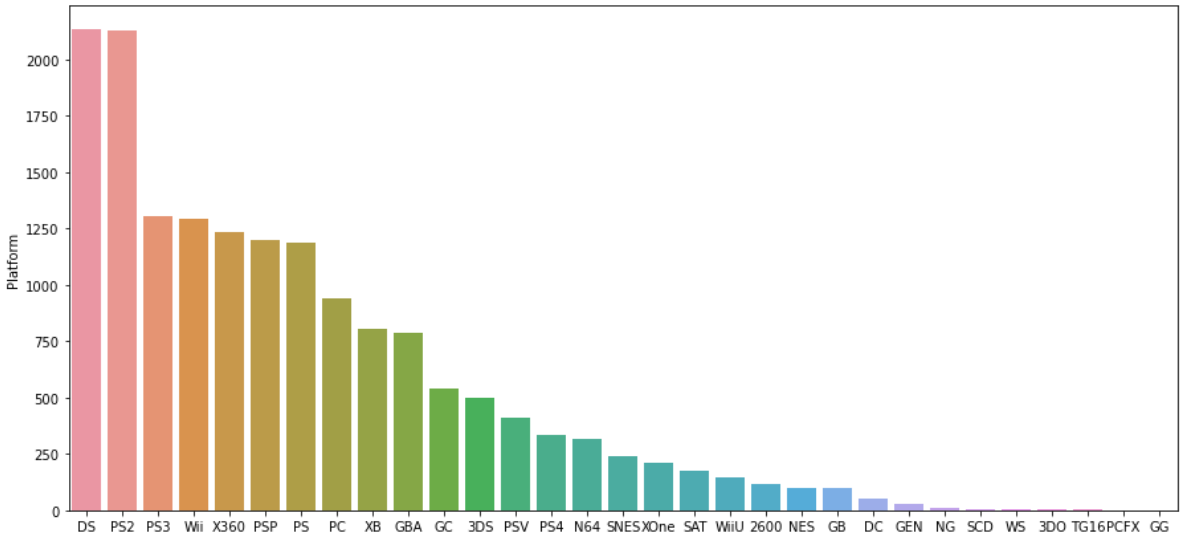
The correlaion matrix above shows the correlation between numerical variables. As we can see the NA_Sales, EU_Sales, JP_Sales, and Other_Sales are the highest correlated with target variable Global_Sales, and even between each other, this expected because the value of Global_Sales is the summtion of the these variables. There is very weak correclation between Year and other numeric variables.

Now we will start investigate the relation between different variables

In [ ]:
```
plt.figure(figsize=(15,7))
sns.barplot(df['Platform'].value_counts().index, df['Platform'].value_count
```

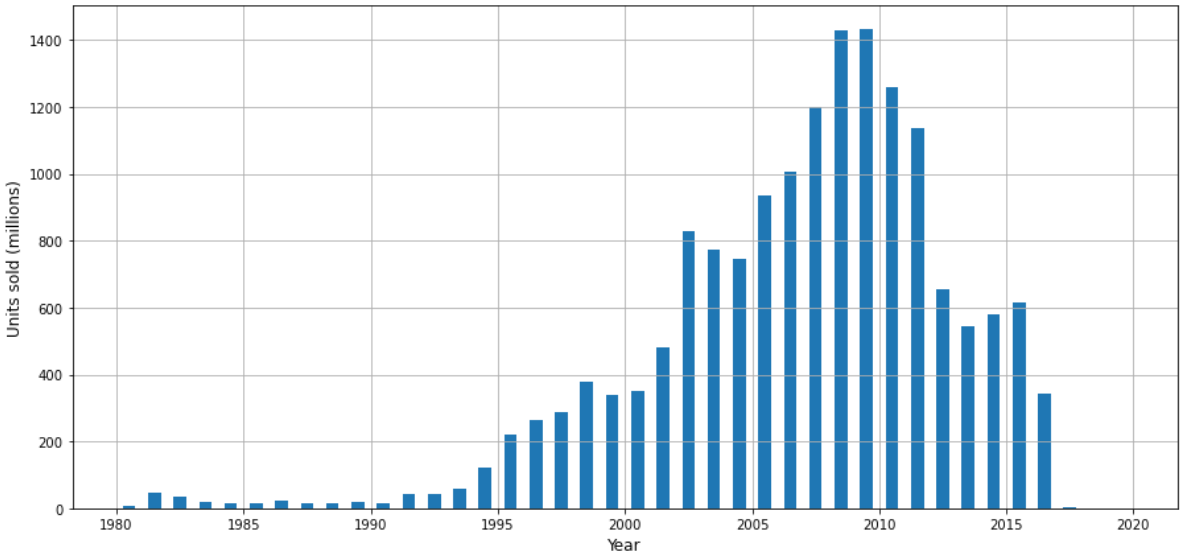
```
C:\Users\cgals\anaconda3\lib\site-packages\seaborn\_decorators.py:36: Futur
eWarning: Pass the following variables as keyword args: x, y. From version
0.12, the only valid positional argument will be `data`, and passing other
arguments without an explicit keyword will result in an error or misinterpr
etation.
  warnings.warn(
```

Out[ ]: <AxesSubplot:ylabel='Platform'>



As we can see third of the platforms have the majority of the games.

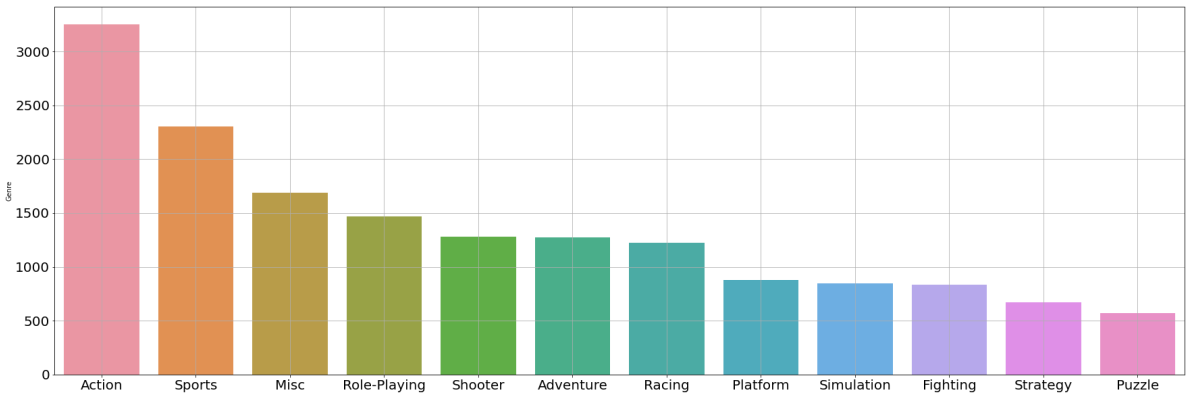
In [ ]:
```
plt.figure(figsize=(15,7))
plt.xlabel('Year',fontsize=12)
plt.ylabel('Units sold (millions)', fontsize=12)
plt.hist(df['Year'],bins=[a for a in range(1980,2021)],rwidth=0.5)
plt.grid()
plt.show()
```



As we can see the majority the games released between 2000 to 2017. It seems the dataset not enough for the year after 2017

In [ ]:
```
plt.figure(figsize=(30,10))
sns.barplot(df['Genre'].value_counts().index,df['Genre'].value_counts())

plt.xticks(fontsize=20)
plt.yticks(fontsize=20)
plt.grid()
plt.show()
```

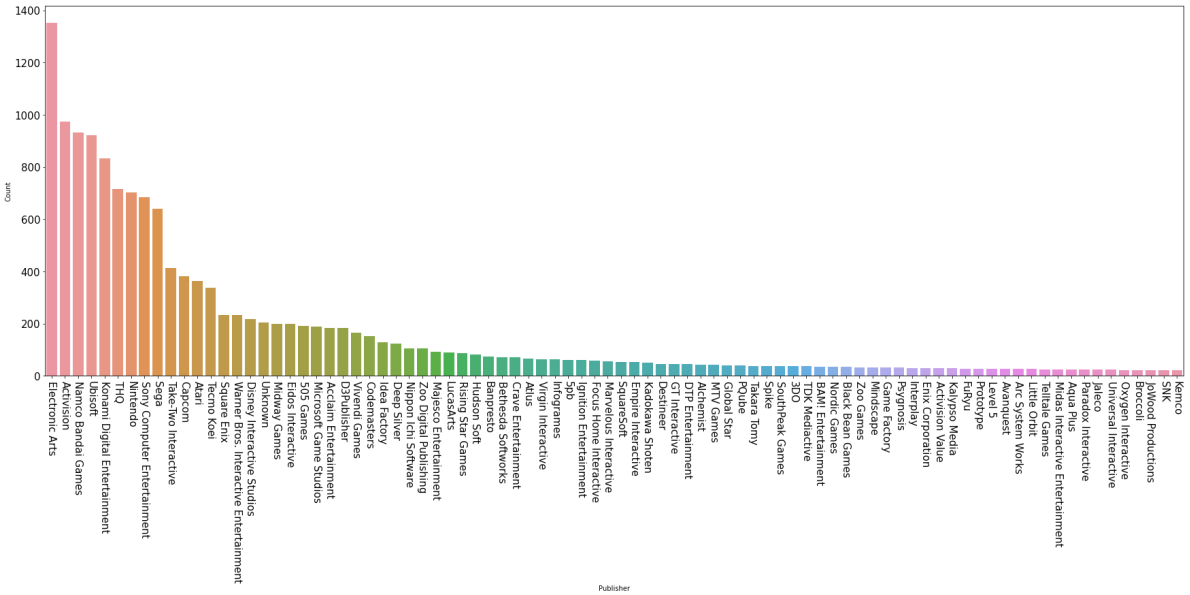
```
C:\Users\cgals\anaconda3\lib\site-packages\seaborn\_decorators.py:36: Futur
eWarning: Pass the following variables as keyword args: x, y. From version
0.12, the only valid positional argument will be `data`, and passing other
arguments without an explicit keyword will result in an error or misinterpr
etation.
  warnings.warn(
```



The action and sport games have the highest no of sold games amongest the other games genre

Game Publishers vs their games count

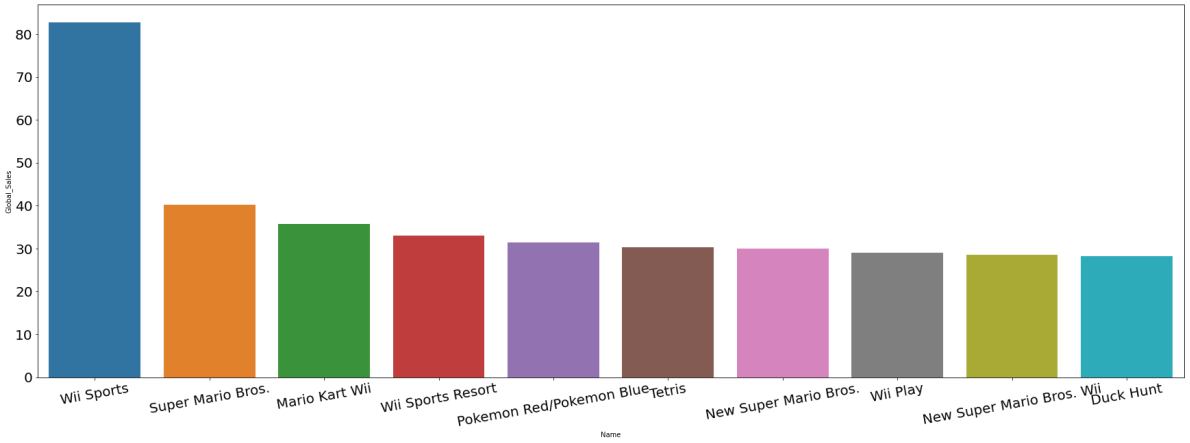
In [ ]:
```
x=pd.DataFrame(data=df.Publisher.value_counts(), columns=['Publisher'])
x.reset_index(inplace=True)
x.rename(columns={'Publisher': 'Count','index':'Publisher'}, inplace=True)
plt.figure(figsize=(30,10))
x=x[x.Count>20]
sns.barplot(x=x.Publisher, y=x["Count"], label='Publisher')
plt.xticks(rotation=-90, fontsize=15)
plt.yticks(fontsize=15)

plt.show()
```



the above graph show those game Publishers with more than 20 games count. Most of the games are belong to few publishers

10 Highest Global_Sales game

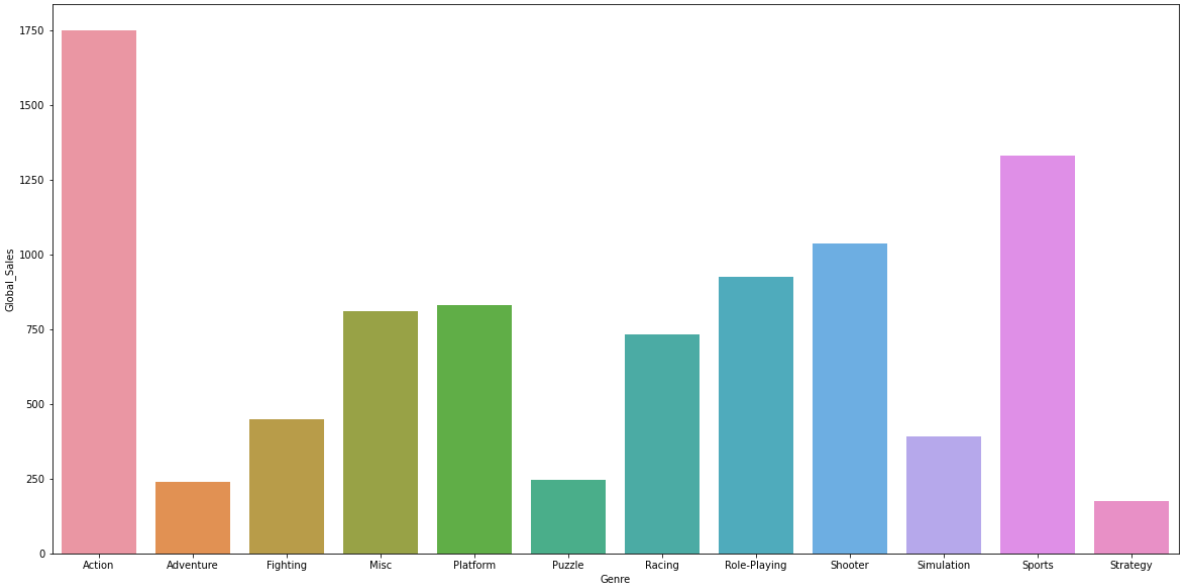
In [ ]:
```
df_gsale=df[['Name','Global_Sales']].sort_values(by='Global_Sales',ascendir
plt.figure(figsize=(30,10))
sns.barplot(data=df_gsale, x='Name', y='Global_Sales')
plt.xticks(rotation= 10, fontsize=20)
plt.yticks(fontsize=20)
plt.show()
```



Game Global Sales per Genre

In [ ]:
```
Genre_gsale = pd.DataFrame(df.groupby('Genre')['Global_Sales'].sum())
Genre_gsale.reset_index(inplace=True)
plt.figure(figsize=(20,10))
sns.barplot(data=Genre_gsale, x='Genre', y='Global_Sales')
```

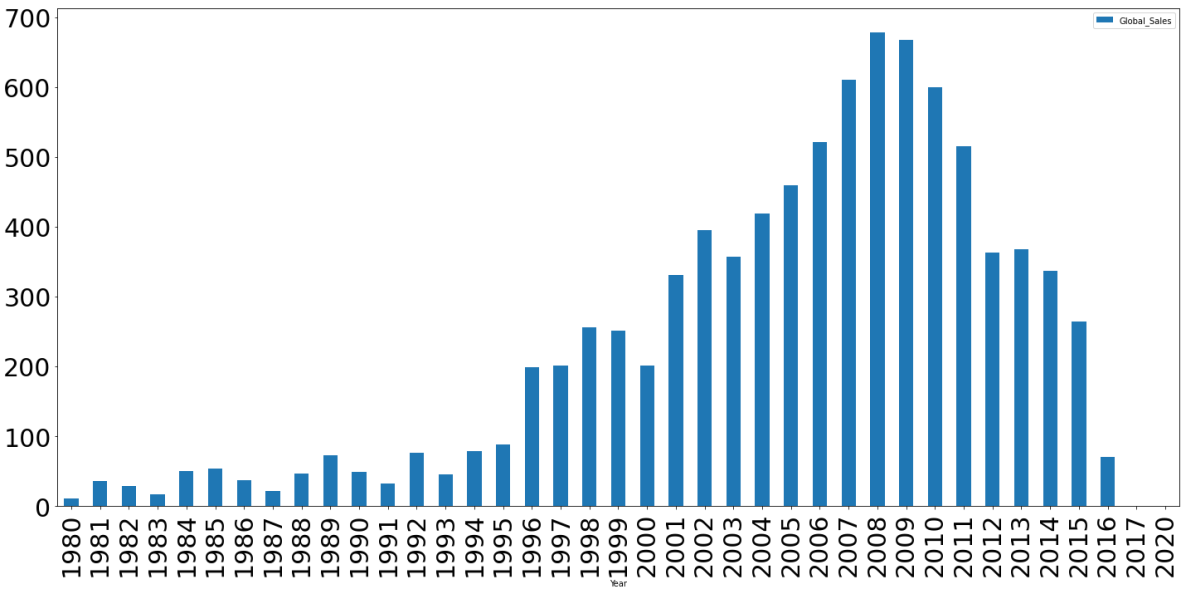
Out[ ]: <AxesSubplot:xlabel='Genre', ylabel='Global_Sales'>



The most game genre sold are Action and sports games

Year vs Global sales plot

In [ ]:
```
sales=df[['Year', 'Global_Sales']].groupby('Year').sum()
sales.index=sales.index.astype(int)
sales.plot.bar(figsize=(20,10),fontsize=30)
plt.tight_layout()
plt.show()
```



10.2 Web scraping.ipynb

A2_Web scraping.ipynb

MetaDF.csv

In [ ]:
```
from bs4 import BeautifulSoup
import random
import time
import pandas as pd
from urllib.request import Request, urlopen
```

In [ ]:
```
# Years 1996-2000
games_name=[]
games_platform=[]
rel_date=[]
meta_score=[]
user_score = []
for j,k in zip((1996,1997,1998,1999,2000),(1,1,1,1,4)):
    for i in range(0,k):
        url = "https://www.metacritic.com/browse/games/score/metascore/year
        print(url)
        req = Request(url, headers={'User-Agent': 'Firefox/92.0'})
        webpage = urlopen(req).read()
        soup = BeautifulSoup(webpage, 'html.parser')
        for element in soup.select(".title h3"):
            games_name.append(element.text)
        for element in soup.select(".platform .data"):
            games_platform.append(element.text.strip())
        for element in soup.select(".platform+ span"):
            rel_date.append(element.text)
        for element in soup.select(".clamp-metascore .positive"):
            meta_score.append(element.text)
        for element in soup.select(".clamp-metascore .mixed"):
            meta_score.append(element.text)
        for element in soup.select(".clamp-metascore .negative"):
            meta_score.append(element.text)
        for element in soup.select(".user"):
            user_score.append(element.text)
        time.sleep(30*random.uniform(0.3,1))

lustrum1 = pd.DataFrame(list(zip(games_name,games_platform, rel_date, meta_
                           columns=['Name', 'Platform', 'Release Date', 'Met
print("Data size:",lustrum1.shape)
lustrum1.head()
```

```
https://www.metacritic.com/browse/games/score/metascore/year/all/filtered?y
ear_selected=1996&distribution=&sort=desc&view=detailed&page=0
https://www.metacritic.com/browse/games/score/metascore/year/all/filtered?y
ear_selected=1997&distribution=&sort=desc&view=detailed&page=0
---------------------------------------------------------------------------
HTTPError                                 Traceback (most recent call last)
<ipython-input-11-31f667872f09> in <module>
     10         print(url)
     11         req = Request(url, headers={'User-Agent': 'Firefox/92.0'})
---> 12         webpage = urlopen(req).read()
     13         soup = BeautifulSoup(webpage, 'html.parser')
     14         for element in soup.select(".title h3"):

~\anaconda3\lib\urllib\request.py in urlopen(url, data, timeout, cafile, ca
path, cadefault, context)
    220     else:
    221         opener = _opener
--> 222     return opener.open(url, data, timeout)
    223
    224 def install_opener(opener):
```

```
~\anaconda3\lib\urllib\request.py in open(self, fullurl, data, timeout)
    529         for processor in self.process_response.get(protocol, []):
    530             meth = getattr(processor, meth_name)
--> 531             response = meth(req, response)
    532
    533         return response

~\anaconda3\lib\urllib\request.py in http_response(self, request, response)
    638         # request was successfully received, understood, and accept
ed.
    639         if not (200 <= code < 300):
--> 640             response = self.parent.error(
    641                 'http', request, response, code, msg, hdrs)
    642

~\anaconda3\lib\urllib\request.py in error(self, proto, *args)
    567         if http_err:
    568             args = (dict, 'default', 'http_error_default') + orig_a
rgs
--> 569             return self._call_chain(*args)
    570
    571 # XXX probably also want an abstract factory that knows when it mak
es

~\anaconda3\lib\urllib\request.py in _call_chain(self, chain, kind, meth_na
me, *args)
    500         for handler in handlers:
    501             func = getattr(handler, meth_name)
--> 502             result = func(*args)
    503             if result is not None:
    504                 return result

~\anaconda3\lib\urllib\request.py in http_error_default(self, req, fp, cod
e, msg, hdrs)
    647 class HTTPDefaultErrorHandler(BaseHandler):
    648     def http_error_default(self, req, fp, code, msg, hdrs):
--> 649         raise HTTPError(req.full_url, code, msg, hdrs, fp)
    650
    651 class HTTPRedirectHandler(BaseHandler):
```

In [ ]:
```
len(games_name)
```

Out[ ]: 20

In [ ]:
```
for j,k in zip((1997,1998,1999,2000),(1,1,1,4)):
    for i in range(0,k):
        url = "https://www.metacritic.com/browse/games/score/metascore/year
        print(url)
        req = Request(url, headers={'User-Agent': 'Firefox/92.0'})
        webpage = urlopen(req).read()
        soup = BeautifulSoup(webpage, 'html.parser')
        for element in soup.select(".title h3"):
            games_name.append(element.text)
        for element in soup.select(".platform .data"):
            games_platform.append(element.text.strip())
        for element in soup.select(".platform+ span"):
            rel_date.append(element.text)
        for element in soup.select(".clamp-metascore .positive"):
            meta_score.append(element.text)
        for element in soup.select(".clamp-metascore .mixed"):
            meta_score.append(element.text)
        for element in soup.select(".clamp-metascore .negative"):
            meta_score.append(element.text)
        for element in soup.select(".user"):
            user_score.append(element.text)
        time.sleep(30*random.uniform(0.3,1))

lustrum1 = pd.DataFrame(list(zip(games_name,games_platform, rel_date, meta_
                                 columns=['Name', 'Platform', 'Release Date', 'Met
print("Data size:",lustrum1.shape)
lustrum1.head()
```

```
https://www.metacritic.com/browse/games/score/metascore/year/all/filtered?y
ear_selected=1997&distribution=&sort=desc&view=detailed&page=0
https://www.metacritic.com/browse/games/score/metascore/year/all/filtered?y
ear_selected=1998&distribution=&sort=desc&view=detailed&page=0
https://www.metacritic.com/browse/games/score/metascore/year/all/filtered?y
ear_selected=1999&distribution=&sort=desc&view=detailed&page=0
https://www.metacritic.com/browse/games/score/metascore/year/all/filtered?y
ear_selected=2000&distribution=&sort=desc&view=detailed&page=0
https://www.metacritic.com/browse/games/score/metascore/year/all/filtered?y
ear_selected=2000&distribution=&sort=desc&view=detailed&page=1
https://www.metacritic.com/browse/games/score/metascore/year/all/filtered?y
ear_selected=2000&distribution=&sort=desc&view=detailed&page=2
https://www.metacritic.com/browse/games/score/metascore/year/all/filtered?y
ear_selected=2000&distribution=&sort=desc&view=detailed&page=3
Data size: (502, 5)
```

Out[ ]:

	Name	Platform	Release Date	Meta_Score	User_Score
0	Sid Meier's Civilization II	PC	February 29, 1996	94	8.8
1	Quake	PC	June 22, 1996	94	8.8
2	Diablo	PC	December 31, 1996	94	8.7
3	Super Mario 64	Nintendo 64	September 26, 1996	94	9.1
4	Wipeout XL	PlayStation	September 30, 1996	93	8.6

In [ ]:
```
lustrum1.to_csv('lustrum1.csv', index=False)
```

In [ ]:

```
# Years 2001-2005
games_name=[]
games_platform=[]
rel_date=[]
meta_score=[]
user_score = []
for j,k in zip((2001,2002,2003,2004,2005),(6,8,9,8,9)):
    for i in range(0,k):
        url = "https://www.metacritic.com/browse/games/score/metascore/year
        print(url)
        req = Request(url, headers={'User-Agent': 'Firefox/92.0'})
        webpage = urlopen(req).read()
        soup = BeautifulSoup(webpage, 'html.parser')
        for element in soup.select(".title h3"):
            games_name.append(element.text)
        for element in soup.select(".platform .data"):
            games_platform.append(element.text.strip())
        for element in soup.select(".platform+ span"):
            rel_date.append(element.text)
        for element in soup.select(".clamp-metascore .positive"):
            meta_score.append(element.text)
        for element in soup.select(".clamp-metascore .mixed"):
            meta_score.append(element.text)
        for element in soup.select(".clamp-metascore .negative"):
            meta_score.append(element.text)
        for element in soup.select(".user"):
            user_score.append(element.text)
        time.sleep(30*random.uniform(0.3,1))


lustrum2 = pd.DataFrame(list(zip(games_name,games_platform, rel_date, meta_
                          columns=['Name', 'Platform', 'Release Date', 'Met

lustrum2.info
```

```
https://www.metacritic.com/browse/games/score/metascore/year/all/filtered?y
ear_selected=2001&distribution=&sort=desc&view=detailed&page=0
https://www.metacritic.com/browse/games/score/metascore/year/all/filtered?y
ear_selected=2001&distribution=&sort=desc&view=detailed&page=1
https://www.metacritic.com/browse/games/score/metascore/year/all/filtered?y
ear_selected=2001&distribution=&sort=desc&view=detailed&page=2
https://www.metacritic.com/browse/games/score/metascore/year/all/filtered?y
ear_selected=2001&distribution=&sort=desc&view=detailed&page=3
https://www.metacritic.com/browse/games/score/metascore/year/all/filtered?y
ear_selected=2001&distribution=&sort=desc&view=detailed&page=4
https://www.metacritic.com/browse/games/score/metascore/year/all/filtered?y
ear_selected=2001&distribution=&sort=desc&view=detailed&page=5
https://www.metacritic.com/browse/games/score/metascore/year/all/filtered?y
ear_selected=2002&distribution=&sort=desc&view=detailed&page=0
https://www.metacritic.com/browse/games/score/metascore/year/all/filtered?y
ear_selected=2002&distribution=&sort=desc&view=detailed&page=1
https://www.metacritic.com/browse/games/score/metascore/year/all/filtered?y
ear_selected=2002&distribution=&sort=desc&view=detailed&page=2
https://www.metacritic.com/browse/games/score/metascore/year/all/filtered?y
ear_selected=2002&distribution=&sort=desc&view=detailed&page=3
https://www.metacritic.com/browse/games/score/metascore/year/all/filtered?y
ear_selected=2002&distribution=&sort=desc&view=detailed&page=4
https://www.metacritic.com/browse/games/score/metascore/year/all/filtered?y
ear_selected=2002&distribution=&sort=desc&view=detailed&page=5
https://www.metacritic.com/browse/games/score/metascore/year/all/filtered?y
ear_selected=2002&distribution=&sort=desc&view=detailed&page=6
https://www.metacritic.com/browse/games/score/metascore/year/all/filtered?y
ear_selected=2002&distribution=&sort=desc&view=detailed&page=7
https://www.metacritic.com/browse/games/score/metascore/year/all/filtered?y
```

```
ear_selected=2003&distribution=&sort=desc&view=detailed&page=0
https://www.metacritic.com/browse/games/score/metascore/year/all/filtered?y
ear_selected=2003&distribution=&sort=desc&view=detailed&page=1
https://www.metacritic.com/browse/games/score/metascore/year/all/filtered?y
ear_selected=2003&distribution=&sort=desc&view=detailed&page=2
https://www.metacritic.com/browse/games/score/metascore/year/all/filtered?y
ear_selected=2003&distribution=&sort=desc&view=detailed&page=3
https://www.metacritic.com/browse/games/score/metascore/year/all/filtered?y
ear_selected=2003&distribution=&sort=desc&view=detailed&page=4
https://www.metacritic.com/browse/games/score/metascore/year/all/filtered?y
ear_selected=2003&distribution=&sort=desc&view=detailed&page=5
https://www.metacritic.com/browse/games/score/metascore/year/all/filtered?y
ear_selected=2003&distribution=&sort=desc&view=detailed&page=6
https://www.metacritic.com/browse/games/score/metascore/year/all/filtered?y
ear_selected=2003&distribution=&sort=desc&view=detailed&page=7
https://www.metacritic.com/browse/games/score/metascore/year/all/filtered?y
ear_selected=2003&distribution=&sort=desc&view=detailed&page=8
https://www.metacritic.com/browse/games/score/metascore/year/all/filtered?y
ear_selected=2004&distribution=&sort=desc&view=detailed&page=0
https://www.metacritic.com/browse/games/score/metascore/year/all/filtered?y
ear_selected=2004&distribution=&sort=desc&view=detailed&page=1
https://www.metacritic.com/browse/games/score/metascore/year/all/filtered?y
ear_selected=2004&distribution=&sort=desc&view=detailed&page=2
https://www.metacritic.com/browse/games/score/metascore/year/all/filtered?y
ear_selected=2004&distribution=&sort=desc&view=detailed&page=3
https://www.metacritic.com/browse/games/score/metascore/year/all/filtered?y
ear_selected=2004&distribution=&sort=desc&view=detailed&page=4
https://www.metacritic.com/browse/games/score/metascore/year/all/filtered?y
ear_selected=2004&distribution=&sort=desc&view=detailed&page=5
https://www.metacritic.com/browse/games/score/metascore/year/all/filtered?y
ear_selected=2004&distribution=&sort=desc&view=detailed&page=6
https://www.metacritic.com/browse/games/score/metascore/year/all/filtered?y
ear_selected=2004&distribution=&sort=desc&view=detailed&page=7
https://www.metacritic.com/browse/games/score/metascore/year/all/filtered?y
ear_selected=2005&distribution=&sort=desc&view=detailed&page=0
https://www.metacritic.com/browse/games/score/metascore/year/all/filtered?y
ear_selected=2005&distribution=&sort=desc&view=detailed&page=1
https://www.metacritic.com/browse/games/score/metascore/year/all/filtered?y
ear_selected=2005&distribution=&sort=desc&view=detailed&page=2
https://www.metacritic.com/browse/games/score/metascore/year/all/filtered?y
ear_selected=2005&distribution=&sort=desc&view=detailed&page=3
https://www.metacritic.com/browse/games/score/metascore/year/all/filtered?y
ear_selected=2005&distribution=&sort=desc&view=detailed&page=4
https://www.metacritic.com/browse/games/score/metascore/year/all/filtered?y
ear_selected=2005&distribution=&sort=desc&view=detailed&page=5
https://www.metacritic.com/browse/games/score/metascore/year/all/filtered?y
ear_selected=2005&distribution=&sort=desc&view=detailed&page=6
https://www.metacritic.com/browse/games/score/metascore/year/all/filtered?y
ear_selected=2005&distribution=&sort=desc&view=detailed&page=7
https://www.metacritic.com/browse/games/score/metascore/year/all/filtered?y
```

Out[ ]:
```
<bound method DataFrame.info of
Name       Platform  \
0                      Tony Hawk's Pro Skater 3  PlayStation 2
1                          Grand Theft Auto III  PlayStation 2
2                          Halo: Combat Evolved          Xbox
3           Metal Gear Solid 2: Sons of Liberty  PlayStation 2
4                        Gran Turismo 3: A-Spec  PlayStation 2
...                                         ...           ...
3722                         Chicago Enforcer          Xbox
3723                           Dragon Booster            DS
3724  Land of the Dead: Road to Fiddler's Green          Xbox
3725                       Crime Life: Gang Wars          Xbox
3726         Charlie and the Chocolate Factory            PC

          Release Date Meta_Score User_Score
0      October 28, 2001         97        7.5
1      October 22, 2001         97        8.4
2     November 14, 2001         97        8.7
3     November 12, 2001         96        8.8
```

```
4          July 9, 2001         95        8.4
...                 ...        ...        ...
3722  February 23, 2005         33        2.6
3723   December 7, 2005         33        7.0
3724   October 26, 2005         32        6.5
3725  November 22, 2005         30        5.1
3726      July 12, 2005         26        2.5

[3727 rows x 5 columns]>
```

In [ ]:
```
lustrum2.to_csv('lustrum2.csv', index=False)
```

In [ ]:
```
# Years 2006-2010
games_name=[]
games_platform=[]
rel_date=[]
meta_score=[]
user_score = []
for j,k in zip((2006,2007,2008,2009,2010),(9,10,10,10,9)):
    for i in range(0,k):
        url = "https://www.metacritic.com/browse/games/score/metascore/year
        print(url)
        req = Request(url, headers={'User-Agent': 'Firefox/92.0'})
        webpage = urlopen(req).read()
        soup = BeautifulSoup(webpage, 'html.parser')
        for element in soup.select(".title h3"):
            games_name.append(element.text)
        for element in soup.select(".platform .data"):
            games_platform.append(element.text.strip())
        for element in soup.select(".platform+ span"):
            rel_date.append(element.text)
        for element in soup.select(".clamp-metascore .positive"):
            meta_score.append(element.text)
        for element in soup.select(".clamp-metascore .mixed"):
            meta_score.append(element.text)
        for element in soup.select(".clamp-metascore .negative"):
            meta_score.append(element.text)
        for element in soup.select(".user"):
            user_score.append(element.text)
        time.sleep(30*random.uniform(0.3,1))


lustrum3 = pd.DataFrame(list(zip(games_name,games_platform, rel_date, meta_
                         columns=['Name', 'Platform', 'Release Date', 'Met

print("Data size:",lustrum3.shape)
lustrum3.to_csv('lustrum3.csv', index=False)
```

```
https://www.metacritic.com/browse/games/score/metascore/year/all/filtered?y
ear_selected=2006&distribution=&sort=desc&view=detailed&page=0
https://www.metacritic.com/browse/games/score/metascore/year/all/filtered?y
ear_selected=2006&distribution=&sort=desc&view=detailed&page=1
https://www.metacritic.com/browse/games/score/metascore/year/all/filtered?y
ear_selected=2006&distribution=&sort=desc&view=detailed&page=2
https://www.metacritic.com/browse/games/score/metascore/year/all/filtered?y
ear_selected=2006&distribution=&sort=desc&view=detailed&page=3
https://www.metacritic.com/browse/games/score/metascore/year/all/filtered?y
ear_selected=2006&distribution=&sort=desc&view=detailed&page=4
https://www.metacritic.com/browse/games/score/metascore/year/all/filtered?y
ear_selected=2006&distribution=&sort=desc&view=detailed&page=5
https://www.metacritic.com/browse/games/score/metascore/year/all/filtered?y
ear_selected=2006&distribution=&sort=desc&view=detailed&page=6
https://www.metacritic.com/browse/games/score/metascore/year/all/filtered?y
ear_selected=2006&distribution=&sort=desc&view=detailed&page=7
```

```
https://www.metacritic.com/browse/games/score/metascore/year/all/filtered?y
ear_selected=2006&distribution=&sort=desc&view=detailed&page=8
https://www.metacritic.com/browse/games/score/metascore/year/all/filtered?y
ear_selected=2007&distribution=&sort=desc&view=detailed&page=0
https://www.metacritic.com/browse/games/score/metascore/year/all/filtered?y
ear_selected=2007&distribution=&sort=desc&view=detailed&page=1
https://www.metacritic.com/browse/games/score/metascore/year/all/filtered?y
ear_selected=2007&distribution=&sort=desc&view=detailed&page=2
https://www.metacritic.com/browse/games/score/metascore/year/all/filtered?y
ear_selected=2007&distribution=&sort=desc&view=detailed&page=3
https://www.metacritic.com/browse/games/score/metascore/year/all/filtered?y
ear_selected=2007&distribution=&sort=desc&view=detailed&page=4
https://www.metacritic.com/browse/games/score/metascore/year/all/filtered?y
ear_selected=2007&distribution=&sort=desc&view=detailed&page=5
https://www.metacritic.com/browse/games/score/metascore/year/all/filtered?y
ear_selected=2007&distribution=&sort=desc&view=detailed&page=6
https://www.metacritic.com/browse/games/score/metascore/year/all/filtered?y
ear_selected=2007&distribution=&sort=desc&view=detailed&page=7
https://www.metacritic.com/browse/games/score/metascore/year/all/filtered?y
ear_selected=2007&distribution=&sort=desc&view=detailed&page=8
https://www.metacritic.com/browse/games/score/metascore/year/all/filtered?y
ear_selected=2007&distribution=&sort=desc&view=detailed&page=9
https://www.metacritic.com/browse/games/score/metascore/year/all/filtered?y
ear_selected=2008&distribution=&sort=desc&view=detailed&page=0
https://www.metacritic.com/browse/games/score/metascore/year/all/filtered?y
ear_selected=2008&distribution=&sort=desc&view=detailed&page=1
https://www.metacritic.com/browse/games/score/metascore/year/all/filtered?y
ear_selected=2008&distribution=&sort=desc&view=detailed&page=2
https://www.metacritic.com/browse/games/score/metascore/year/all/filtered?y
ear_selected=2008&distribution=&sort=desc&view=detailed&page=3
https://www.metacritic.com/browse/games/score/metascore/year/all/filtered?y
ear_selected=2008&distribution=&sort=desc&view=detailed&page=4
https://www.metacritic.com/browse/games/score/metascore/year/all/filtered?y
ear_selected=2008&distribution=&sort=desc&view=detailed&page=5
https://www.metacritic.com/browse/games/score/metascore/year/all/filtered?y
ear_selected=2008&distribution=&sort=desc&view=detailed&page=6
https://www.metacritic.com/browse/games/score/metascore/year/all/filtered?y
ear_selected=2008&distribution=&sort=desc&view=detailed&page=7
https://www.metacritic.com/browse/games/score/metascore/year/all/filtered?y
ear_selected=2008&distribution=&sort=desc&view=detailed&page=8
https://www.metacritic.com/browse/games/score/metascore/year/all/filtered?y
ear_selected=2008&distribution=&sort=desc&view=detailed&page=9
https://www.metacritic.com/browse/games/score/metascore/year/all/filtered?y
ear_selected=2009&distribution=&sort=desc&view=detailed&page=0
https://www.metacritic.com/browse/games/score/metascore/year/all/filtered?y
ear_selected=2009&distribution=&sort=desc&view=detailed&page=1
https://www.metacritic.com/browse/games/score/metascore/year/all/filtered?y
ear_selected=2009&distribution=&sort=desc&view=detailed&page=2
https://www.metacritic.com/browse/games/score/metascore/year/all/filtered?y
ear_selected=2009&distribution=&sort=desc&view=detailed&page=3
https://www.metacritic.com/browse/games/score/metascore/year/all/filtered?y
ear_selected=2009&distribution=&sort=desc&view=detailed&page=4
https://www.metacritic.com/browse/games/score/metascore/year/all/filtered?y
ear_selected=2009&distribution=&sort=desc&view=detailed&page=5
https://www.metacritic.com/browse/games/score/metascore/year/all/filtered?y
ear_selected=2009&distribution=&sort=desc&view=detailed&page=6
https://www.metacritic.com/browse/games/score/metascore/year/all/filtered?y
ear_selected=2009&distribution=&sort=desc&view=detailed&page=7
https://www.metacritic.com/browse/games/score/metascore/year/all/filtered?y
ear_selected=2009&distribution=&sort=desc&view=detailed&page=8
https://www.metacritic.com/browse/games/score/metascore/year/all/filtered?y
ear selected=2009&distribution=&sort=desc&view=detailed&page=9
---------------------------------------------------------------------------
HTTPError                                 Traceback (most recent call last)
<ipython-input-21-228a887f1cde> in <module>
     10         print(url)
     11         req = Request(url, headers={'User-Agent': 'Firefox/92.0'})
---> 12         webpage = urlopen(req).read()
     13         soup = BeautifulSoup(webpage, 'html.parser')
```

```
     14         for element in soup.select(".title h3"):

~\anaconda3\lib\urllib\request.py in urlopen(url, data, timeout, cafile, ca
path, cadefault, context)
    220     else:
    221         opener = _opener
--> 222     return opener.open(url, data, timeout)
    223
    224 def install_opener(opener):

~\anaconda3\lib\urllib\request.py in open(self, fullurl, data, timeout)
    529         for processor in self.process_response.get(protocol, []):
    530             meth = getattr(processor, meth_name)
--> 531             response = meth(req, response)
    532
    533         return response

~\anaconda3\lib\urllib\request.py in http_response(self, request, response)
    638         # request was successfully received, understood, and accept
ed.
    639         if not (200 <= code < 300):
--> 640             response = self.parent.error(
    641                 'http', request, response, code, msg, hdrs)
    642

~\anaconda3\lib\urllib\request.py in error(self, proto, *args)
    567         if http_err:
    568             args = (dict, 'default', 'http_error_default') + orig_a
rgs
--> 569             return self._call_chain(*args)
    570
    571 # XXX probably also want an abstract factory that knows when it mak
es

~\anaconda3\lib\urllib\request.py in _call_chain(self, chain, kind, meth_na
me, *args)
    500         for handler in handlers:
    501             func = getattr(handler, meth_name)
--> 502             result = func(*args)
    503             if result is not None:
    504                 return result

~\anaconda3\lib\urllib\request.py in http_error_default(self, req, fp, cod
e, msg, hdrs)
    647 class HTTPDefaultErrorHandler(BaseHandler):
    648     def http_error_default(self, req, fp, code, msg, hdrs):
--> 649         raise HTTPError(req.full_url, code, msg, hdrs, fp)
    650
    651 class HTTPRedirectHandler(BaseHandler):
```

In [ ]:
```
lustrum3a = pd.DataFrame(list(zip(games_name,games_platform, rel_date, meta
                          columns=['Name', 'Platform', 'Release Date', 'Met
lustrum3a.shape
```

Out[ ]:
```
(3667, 5)
```

In [ ]:
```
lustrum3a.to_csv('lustrum3a.csv', index=False)
```

In [ ]:

```
# Years 2011-2015
games_name=[]
games_platform=[]
rel_date=[]
meta_score=[]
user_score = []
for j,k in zip((2011,2012,2013,2014,2015),(9,8,8,8,9)):
    for i in range(0,k):
        url = "https://www.metacritic.com/browse/games/score/metascore/year
        print(url)
        req = Request(url, headers={'User-Agent': 'Firefox/92.0'})
        webpage = urlopen(req).read()
        soup = BeautifulSoup(webpage, 'html.parser')
        for element in soup.select(".title h3"):
            games_name.append(element.text)
        for element in soup.select(".platform .data"):
            games_platform.append(element.text.strip())
        for element in soup.select(".platform+ span"):
            rel_date.append(element.text)
        for element in soup.select(".clamp-metascore .positive"):
            meta_score.append(element.text)
        for element in soup.select(".clamp-metascore .mixed"):
            meta_score.append(element.text)
        for element in soup.select(".clamp-metascore .negative"):
            meta_score.append(element.text)
        for element in soup.select(".user"):
            user_score.append(element.text)
        time.sleep(30*random.uniform(0.3,1))


lustrum4 = pd.DataFrame(list(zip(games_name,games_platform, rel_date, meta_
                          columns=['Name', 'Platform', 'Release Date', 'Met

print("Data size:",lustrum4.shape)
lustrum4.to_csv('lustrum4.csv', index=False)
```

```
https://www.metacritic.com/browse/games/score/metascore/year/all/filtered?y
ear_selected=2011&distribution=&sort=desc&view=detailed&page=0
https://www.metacritic.com/browse/games/score/metascore/year/all/filtered?y
ear_selected=2011&distribution=&sort=desc&view=detailed&page=1
https://www.metacritic.com/browse/games/score/metascore/year/all/filtered?y
ear_selected=2011&distribution=&sort=desc&view=detailed&page=2
https://www.metacritic.com/browse/games/score/metascore/year/all/filtered?y
ear_selected=2011&distribution=&sort=desc&view=detailed&page=3
https://www.metacritic.com/browse/games/score/metascore/year/all/filtered?y
ear_selected=2011&distribution=&sort=desc&view=detailed&page=4
https://www.metacritic.com/browse/games/score/metascore/year/all/filtered?y
ear_selected=2011&distribution=&sort=desc&view=detailed&page=5
https://www.metacritic.com/browse/games/score/metascore/year/all/filtered?y
ear_selected=2011&distribution=&sort=desc&view=detailed&page=6
https://www.metacritic.com/browse/games/score/metascore/year/all/filtered?y
ear_selected=2011&distribution=&sort=desc&view=detailed&page=7
https://www.metacritic.com/browse/games/score/metascore/year/all/filtered?y
ear_selected=2011&distribution=&sort=desc&view=detailed&page=8
https://www.metacritic.com/browse/games/score/metascore/year/all/filtered?y
ear_selected=2012&distribution=&sort=desc&view=detailed&page=0
https://www.metacritic.com/browse/games/score/metascore/year/all/filtered?y
ear_selected=2012&distribution=&sort=desc&view=detailed&page=1
https://www.metacritic.com/browse/games/score/metascore/year/all/filtered?y
ear_selected=2012&distribution=&sort=desc&view=detailed&page=2
https://www.metacritic.com/browse/games/score/metascore/year/all/filtered?y
ear_selected=2012&distribution=&sort=desc&view=detailed&page=3
https://www.metacritic.com/browse/games/score/metascore/year/all/filtered?y
ear_selected=2012&distribution=&sort=desc&view=detailed&page=4
```

```
https://www.metacritic.com/browse/games/score/metascore/year/all/filtered?y
ear_selected=2012&distribution=&sort=desc&view=detailed&page=5
https://www.metacritic.com/browse/games/score/metascore/year/all/filtered?y
ear_selected=2012&distribution=&sort=desc&view=detailed&page=6
https://www.metacritic.com/browse/games/score/metascore/year/all/filtered?y
ear_selected=2012&distribution=&sort=desc&view=detailed&page=7
https://www.metacritic.com/browse/games/score/metascore/year/all/filtered?y
ear_selected=2013&distribution=&sort=desc&view=detailed&page=0
https://www.metacritic.com/browse/games/score/metascore/year/all/filtered?y
ear_selected=2013&distribution=&sort=desc&view=detailed&page=1
https://www.metacritic.com/browse/games/score/metascore/year/all/filtered?y
ear_selected=2013&distribution=&sort=desc&view=detailed&page=2
https://www.metacritic.com/browse/games/score/metascore/year/all/filtered?y
ear_selected=2013&distribution=&sort=desc&view=detailed&page=3
https://www.metacritic.com/browse/games/score/metascore/year/all/filtered?y
ear_selected=2013&distribution=&sort=desc&view=detailed&page=4
https://www.metacritic.com/browse/games/score/metascore/year/all/filtered?y
ear_selected=2013&distribution=&sort=desc&view=detailed&page=5
https://www.metacritic.com/browse/games/score/metascore/year/all/filtered?y
ear_selected=2013&distribution=&sort=desc&view=detailed&page=6
https://www.metacritic.com/browse/games/score/metascore/year/all/filtered?y
ear_selected=2013&distribution=&sort=desc&view=detailed&page=7
https://www.metacritic.com/browse/games/score/metascore/year/all/filtered?y
ear_selected=2014&distribution=&sort=desc&view=detailed&page=0
https://www.metacritic.com/browse/games/score/metascore/year/all/filtered?y
ear_selected=2014&distribution=&sort=desc&view=detailed&page=1
https://www.metacritic.com/browse/games/score/metascore/year/all/filtered?y
ear_selected=2014&distribution=&sort=desc&view=detailed&page=2
https://www.metacritic.com/browse/games/score/metascore/year/all/filtered?y
ear_selected=2014&distribution=&sort=desc&view=detailed&page=3
https://www.metacritic.com/browse/games/score/metascore/year/all/filtered?y
ear_selected=2014&distribution=&sort=desc&view=detailed&page=4
https://www.metacritic.com/browse/games/score/metascore/year/all/filtered?y
ear_selected=2014&distribution=&sort=desc&view=detailed&page=5
https://www.metacritic.com/browse/games/score/metascore/year/all/filtered?y
ear_selected=2014&distribution=&sort=desc&view=detailed&page=6
https://www.metacritic.com/browse/games/score/metascore/year/all/filtered?y
ear_selected=2014&distribution=&sort=desc&view=detailed&page=7
https://www.metacritic.com/browse/games/score/metascore/year/all/filtered?y
ear_selected=2015&distribution=&sort=desc&view=detailed&page=0
https://www.metacritic.com/browse/games/score/metascore/year/all/filtered?y
ear_selected=2015&distribution=&sort=desc&view=detailed&page=1
https://www.metacritic.com/browse/games/score/metascore/year/all/filtered?y
ear_selected=2015&distribution=&sort=desc&view=detailed&page=2
https://www.metacritic.com/browse/games/score/metascore/year/all/filtered?y
ear_selected=2015&distribution=&sort=desc&view=detailed&page=3
https://www.metacritic.com/browse/games/score/metascore/year/all/filtered?y
ear_selected=2015&distribution=&sort=desc&view=detailed&page=4
https://www.metacritic.com/browse/games/score/metascore/year/all/filtered?y
ear_selected=2015&distribution=&sort=desc&view=detailed&page=5
https://www.metacritic.com/browse/games/score/metascore/year/all/filtered?y
ear_selected=2015&distribution=&sort=desc&view=detailed&page=6
https://www.metacritic.com/browse/games/score/metascore/year/all/filtered?y
ear_selected=2015&distribution=&sort=desc&view=detailed&page=7
https://www.metacritic.com/browse/games/score/metascore/year/all/filtered?y
ear_selected=2015&distribution=&sort=desc&view=detailed&page=8
```

In [ ]:

```
# Years 2016-2020
games_name=[]
games_platform=[]
rel_date=[]
meta_score=[]
user_score = []
for j,k in zip((2016,2017,2018,2019,2020),(10,11,12,11,11)):
    for i in range(0,k):
        url = "https://www.metacritic.com/browse/games/score/metascore/year
        print(url)
        req = Request(url, headers={'User-Agent': 'Firefox/92.0'})
        webpage = urlopen(req).read()
        soup = BeautifulSoup(webpage, 'html.parser')
        for element in soup.select(".title h3"):
            games_name.append(element.text)
        for element in soup.select(".platform .data"):
            games_platform.append(element.text.strip())
        for element in soup.select(".platform+ span"):
            rel_date.append(element.text)
        for element in soup.select(".clamp-metascore .positive"):
            meta_score.append(element.text)
        for element in soup.select(".clamp-metascore .mixed"):
            meta_score.append(element.text)
        for element in soup.select(".clamp-metascore .negative"):
            meta_score.append(element.text)
        for element in soup.select(".user"):
            user_score.append(element.text)
        time.sleep(30*random.uniform(0.3,1))


lustrum5 = pd.DataFrame(list(zip(games_name,games_platform, rel_date, meta_
                          columns=['Name', 'Platform', 'Release Date', 'Met

print("Data size:",lustrum5.shape)
lustrum5.to_csv('lustrum5.csv', index=False)
```

```
https://www.metacritic.com/browse/games/score/metascore/year/all/filtered?y
ear_selected=2016&distribution=&sort=desc&view=detailed&page=0
https://www.metacritic.com/browse/games/score/metascore/year/all/filtered?y
ear_selected=2016&distribution=&sort=desc&view=detailed&page=1
https://www.metacritic.com/browse/games/score/metascore/year/all/filtered?y
ear_selected=2016&distribution=&sort=desc&view=detailed&page=2
https://www.metacritic.com/browse/games/score/metascore/year/all/filtered?y
ear_selected=2016&distribution=&sort=desc&view=detailed&page=3
https://www.metacritic.com/browse/games/score/metascore/year/all/filtered?y
ear_selected=2016&distribution=&sort=desc&view=detailed&page=4
https://www.metacritic.com/browse/games/score/metascore/year/all/filtered?y
ear_selected=2016&distribution=&sort=desc&view=detailed&page=5
https://www.metacritic.com/browse/games/score/metascore/year/all/filtered?y
ear_selected=2016&distribution=&sort=desc&view=detailed&page=6
https://www.metacritic.com/browse/games/score/metascore/year/all/filtered?y
ear_selected=2016&distribution=&sort=desc&view=detailed&page=7
https://www.metacritic.com/browse/games/score/metascore/year/all/filtered?y
ear_selected=2016&distribution=&sort=desc&view=detailed&page=8
https://www.metacritic.com/browse/games/score/metascore/year/all/filtered?y
ear_selected=2016&distribution=&sort=desc&view=detailed&page=9
https://www.metacritic.com/browse/games/score/metascore/year/all/filtered?y
ear_selected=2017&distribution=&sort=desc&view=detailed&page=0
https://www.metacritic.com/browse/games/score/metascore/year/all/filtered?y
ear_selected=2017&distribution=&sort=desc&view=detailed&page=1
https://www.metacritic.com/browse/games/score/metascore/year/all/filtered?y
ear_selected=2017&distribution=&sort=desc&view=detailed&page=2
https://www.metacritic.com/browse/games/score/metascore/year/all/filtered?y
ear_selected=2017&distribution=&sort=desc&view=detailed&page=3
```

```
https://www.metacritic.com/browse/games/score/metascore/year/all/filtered?y
ear_selected=2017&distribution=&sort=desc&view=detailed&page=4
https://www.metacritic.com/browse/games/score/metascore/year/all/filtered?y
ear_selected=2017&distribution=&sort=desc&view=detailed&page=5
https://www.metacritic.com/browse/games/score/metascore/year/all/filtered?y
ear_selected=2017&distribution=&sort=desc&view=detailed&page=6
https://www.metacritic.com/browse/games/score/metascore/year/all/filtered?y
ear_selected=2017&distribution=&sort=desc&view=detailed&page=7
https://www.metacritic.com/browse/games/score/metascore/year/all/filtered?y
ear_selected=2017&distribution=&sort=desc&view=detailed&page=8
https://www.metacritic.com/browse/games/score/metascore/year/all/filtered?y
ear_selected=2017&distribution=&sort=desc&view=detailed&page=9
https://www.metacritic.com/browse/games/score/metascore/year/all/filtered?y
ear_selected=2017&distribution=&sort=desc&view=detailed&page=10
https://www.metacritic.com/browse/games/score/metascore/year/all/filtered?y
ear_selected=2018&distribution=&sort=desc&view=detailed&page=0
https://www.metacritic.com/browse/games/score/metascore/year/all/filtered?y
ear_selected=2018&distribution=&sort=desc&view=detailed&page=1
https://www.metacritic.com/browse/games/score/metascore/year/all/filtered?y
ear_selected=2018&distribution=&sort=desc&view=detailed&page=2
https://www.metacritic.com/browse/games/score/metascore/year/all/filtered?y
ear_selected=2018&distribution=&sort=desc&view=detailed&page=3
https://www.metacritic.com/browse/games/score/metascore/year/all/filtered?y
ear_selected=2018&distribution=&sort=desc&view=detailed&page=4
https://www.metacritic.com/browse/games/score/metascore/year/all/filtered?y
ear_selected=2018&distribution=&sort=desc&view=detailed&page=5
https://www.metacritic.com/browse/games/score/metascore/year/all/filtered?y
ear_selected=2018&distribution=&sort=desc&view=detailed&page=6
https://www.metacritic.com/browse/games/score/metascore/year/all/filtered?y
ear_selected=2018&distribution=&sort=desc&view=detailed&page=7
https://www.metacritic.com/browse/games/score/metascore/year/all/filtered?y
ear_selected=2018&distribution=&sort=desc&view=detailed&page=8
https://www.metacritic.com/browse/games/score/metascore/year/all/filtered?y
ear_selected=2018&distribution=&sort=desc&view=detailed&page=9
https://www.metacritic.com/browse/games/score/metascore/year/all/filtered?y
ear_selected=2018&distribution=&sort=desc&view=detailed&page=10
https://www.metacritic.com/browse/games/score/metascore/year/all/filtered?y
ear_selected=2018&distribution=&sort=desc&view=detailed&page=11
https://www.metacritic.com/browse/games/score/metascore/year/all/filtered?y
ear_selected=2019&distribution=&sort=desc&view=detailed&page=0
https://www.metacritic.com/browse/games/score/metascore/year/all/filtered?y
ear_selected=2019&distribution=&sort=desc&view=detailed&page=1
https://www.metacritic.com/browse/games/score/metascore/year/all/filtered?y
ear_selected=2019&distribution=&sort=desc&view=detailed&page=2
https://www.metacritic.com/browse/games/score/metascore/year/all/filtered?y
ear_selected=2019&distribution=&sort=desc&view=detailed&page=3
https://www.metacritic.com/browse/games/score/metascore/year/all/filtered?y
ear_selected=2019&distribution=&sort=desc&view=detailed&page=4
https://www.metacritic.com/browse/games/score/metascore/year/all/filtered?y
ear_selected=2019&distribution=&sort=desc&view=detailed&page=5
https://www.metacritic.com/browse/games/score/metascore/year/all/filtered?y
ear_selected=2019&distribution=&sort=desc&view=detailed&page=6
https://www.metacritic.com/browse/games/score/metascore/year/all/filtered?y
ear_selected=2019&distribution=&sort=desc&view=detailed&page=7
https://www.metacritic.com/browse/games/score/metascore/year/all/filtered?y
ear_selected=2019&distribution=&sort=desc&view=detailed&page=8
https://www.metacritic.com/browse/games/score/metascore/year/all/filtered?y
ear_selected=2019&distribution=&sort=desc&view=detailed&page=9
https://www.metacritic.com/browse/games/score/metascore/year/all/filtered?y
ear_selected=2019&distribution=&sort=desc&view=detailed&page=10
https://www.metacritic.com/browse/games/score/metascore/year/all/filtered?y
ear_selected=2020&distribution=&sort=desc&view=detailed&page=0
https://www.metacritic.com/browse/games/score/metascore/year/all/filtered?y
ear_selected=2020&distribution=&sort=desc&view=detailed&page=1
https://www.metacritic.com/browse/games/score/metascore/year/all/filtered?y
ear_selected=2020&distribution=&sort=desc&view=detailed&page=2
https://www.metacritic.com/browse/games/score/metascore/year/all/filtered?y
ear_selected=2020&distribution=&sort=desc&view=detailed&page=3
https://www.metacritic.com/browse/games/score/metascore/year/all/filtered?y
ear_selected=2020&distribution=&sort=desc&view=detailed&page=4
```

```
https://www.metacritic.com/browse/games/score/metascore/year/all/filtered?y
ear_selected=2020&distribution=&sort=desc&view=detailed&page=5
https://www.metacritic.com/browse/games/score/metascore/year/all/filtered?y
ear_selected=2020&distribution=&sort=desc&view=detailed&page=6
https://www.metacritic.com/browse/games/score/metascore/year/all/filtered?y
ear_selected=2020&distribution=&sort=desc&view=detailed&page=7
https://www.metacritic.com/browse/games/score/metascore/year/all/filtered?y
ear_selected=2020&distribution=&sort=desc&view=detailed&page=8
https://www.metacritic.com/browse/games/score/metascore/year/all/filtered?y
ear_selected=2020&distribution=&sort=desc&view=detailed&page=9
https://www.metacritic.com/browse/games/score/metascore/year/all/filtered?y
ear_selected=2020&distribution=&sort=desc&view=detailed&page=10
```

In [ ]:

```
# Complete lustrum3
games_name=[]
games_platform=[]
rel_date=[]
meta_score=[]
user_score = []
#Year 2009 page 9
url = "https://www.metacritic.com/browse/games/score/metascore/year/all/fil
print(url)
req = Request(url, headers={'User-Agent': 'Chrome/92.0'})
webpage = urlopen(req).read()
soup = BeautifulSoup(webpage, 'html.parser')
for element in soup.select(".title h3"):
    games_name.append(element.text)
for element in soup.select(".platform .data"):
    games_platform.append(element.text.strip())
for element in soup.select(".platform+ span"):
    rel_date.append(element.text)
for element in soup.select(".clamp-metascore .positive"):
    meta_score.append(element.text)
for element in soup.select(".clamp-metascore .mixed"):
    meta_score.append(element.text)
for element in soup.select(".clamp-metascore .negative"):
    meta_score.append(element.text)
for element in soup.select(".user"):
    user_score.append(element.text)
time.sleep(30*random.uniform(0.3,1))
# Year 2010
for i in range(0,9):
    url = "https://www.metacritic.com/browse/games/score/metascore/year/all
    print(url)
    req = Request(url, headers={'User-Agent': 'Chrome/92.0'})
    webpage = urlopen(req).read()
    soup = BeautifulSoup(webpage, 'html.parser')
    for element in soup.select(".title h3"):
        games_name.append(element.text)
    for element in soup.select(".platform .data"):
        games_platform.append(element.text.strip())
    for element in soup.select(".platform+ span"):
        rel_date.append(element.text)
    for element in soup.select(".clamp-metascore .positive"):
        meta_score.append(element.text)
    for element in soup.select(".clamp-metascore .mixed"):
        meta_score.append(element.text)
    for element in soup.select(".clamp-metascore .negative"):
        meta_score.append(element.text)
    for element in soup.select(".user"):
        user_score.append(element.text)

    time.sleep(30*random.uniform(0.3,1))

lustrum3b = pd.DataFrame(list(zip(games_name,games_platform, rel_date, meta
                          columns=['Name', 'Platform', 'Release Date', 'Met
print("Data size:",lustrum3b.shape)
lustrum3b.to_csv('lustrum3b.csv', index=False)
```

```
https://www.metacritic.com/browse/games/score/metascore/year/all/filtered?y
ear_selected=2009&distribution=&sort=desc&view=detailed&page=9
https://www.metacritic.com/browse/games/score/metascore/year/all/filtered?y
ear_selected=2010&distribution=&sort=desc&view=detailed&page=0
https://www.metacritic.com/browse/games/score/metascore/year/all/filtered?y
ear_selected=2010&distribution=&sort=desc&view=detailed&page=1
```

```
https://www.metacritic.com/browse/games/score/metascore/year/all/filtered?y
ear_selected=2010&distribution=&sort=desc&view=detailed&page=2
https://www.metacritic.com/browse/games/score/metascore/year/all/filtered?y
ear_selected=2010&distribution=&sort=desc&view=detailed&page=3
https://www.metacritic.com/browse/games/score/metascore/year/all/filtered?y
ear_selected=2010&distribution=&sort=desc&view=detailed&page=4
https://www.metacritic.com/browse/games/score/metascore/year/all/filtered?y
ear_selected=2010&distribution=&sort=desc&view=detailed&page=5
https://www.metacritic.com/browse/games/score/metascore/year/all/filtered?y
ear_selected=2010&distribution=&sort=desc&view=detailed&page=6
https://www.metacritic.com/browse/games/score/metascore/year/all/filtered?y
ear_selected=2010&distribution=&sort=desc&view=detailed&page=7
https://www.metacritic.com/browse/games/score/metascore/year/all/filtered?y
ear_selected=2010&distribution=&sort=desc&view=detailed&page=8
Data size: (909, 5)
```

In [ ]:

```
# Year 2021
games_name=[]
games_platform=[]
rel_date=[]
meta_score=[]
user_score = []
for i in range(0,7):
    url = "https://www.metacritic.com/browse/games/score/metascore/year/all
    req = Request(url, headers={'User-Agent': 'Chrome/92.0'})
    webpage = urlopen(req).read()
    soup = BeautifulSoup(webpage, 'html.parser')
    for element in soup.select(".title h3"):
        games_name.append(element.text)
    for element in soup.select(".platform .data"):
        games_platform.append(element.text.strip())
    for element in soup.select(".platform+ span"):
        rel_date.append(element.text)
    for element in soup.select(".clamp-metascore .positive"):
        meta_score.append(element.text)
    for element in soup.select(".clamp-metascore .mixed"):
        meta_score.append(element.text)
    for element in soup.select(".clamp-metascore .negative"):
        meta_score.append(element.text)
    for element in soup.select(".user"):
        user_score.append(element.text)
    print(url)
    time.sleep(30*random.uniform(0.3,1))

lustrum6 = pd.DataFrame(list(zip(games_name,games_platform, rel_date, meta_
                           columns=['Name', 'Platform', 'Release Date', 'Met
print("Data size:",lustrum6.shape)
lustrum6.to_csv('lustrum6.csv', index=False)
```

```
https://www.metacritic.com/browse/games/score/metascore/year/all/filtered?y
ear_selected=2021&distribution=&sort=desc&view=detailed&page=0
https://www.metacritic.com/browse/games/score/metascore/year/all/filtered?y
ear_selected=2021&distribution=&sort=desc&view=detailed&page=1
https://www.metacritic.com/browse/games/score/metascore/year/all/filtered?y
ear_selected=2021&distribution=&sort=desc&view=detailed&page=2
https://www.metacritic.com/browse/games/score/metascore/year/all/filtered?y
ear_selected=2021&distribution=&sort=desc&view=detailed&page=3
https://www.metacritic.com/browse/games/score/metascore/year/all/filtered?y
ear_selected=2021&distribution=&sort=desc&view=detailed&page=4
https://www.metacritic.com/browse/games/score/metascore/year/all/filtered?y
ear_selected=2021&distribution=&sort=desc&view=detailed&page=5
https://www.metacritic.com/browse/games/score/metascore/year/all/filtered?y
ear_selected=2021&distribution=&sort=desc&view=detailed&page=6
Data size: (660, 5)
```

In [ ]:
```
ListLustrums = [lustrum1, lustrum2, lustrum3a,lustrum3b,lustrum4,lustrum5,l
MetaDF = pd.concat(ListLustrums)
```

In [ ]:
```
MetaDF.to_csv('MetaDF.csv', index=False)
```

In [ ]:
```
MetaDF.head()
```

Out[ ]:

	Name	Platform	Release Date	Meta_Score	User_Score
0	Sid Meier's Civilization II	PC	February 29, 1996	94	8.8
1	Quake	PC	June 22, 1996	94	8.8
2	Diablo	PC	December 31, 1996	94	8.7
3	Super Mario 64	Nintendo 64	September 26, 1996	94	9.1
4	Wipeout XL	PlayStation	September 30, 1996	93	8.6

10.3 Merging of vgsales and Meta_vg databases – Project_VG_1.ipynb


A3_Project_VG_1.ipyn
b


VG_Meta_Score.csv

Than this exercise, I want to review if the outliers have any influence in the results. Also, we will perform build a cnn model for the data.

In [ ]:
```
import pandas as pd
import numpy as np
from sklearn.preprocessing import PolynomialFeatures
from sklearn.linear_model import LinearRegression, LogisticRegression
import matplotlib.pyplot as plt
import seaborn as sns

from sklearn.preprocessing import MinMaxScaler, StandardScaler
from sklearn.model_selection import cross_validate, cross_val_predict, cros
from sklearn.metrics import mean_squared_error
from sklearn.linear_model import Ridge, LassoCV
```

Concatenation the five metacritic datasets

In [ ]:
```
# upload all different files of meta data
df0 =pd.read_csv('Meta_vg_0.csv')
df1 =pd.read_csv('Meta_vg_1.csv')
df2 =pd.read_csv('Meta_vg_2.csv')
df3 =pd.read_csv('Meta_vg_3.csv')
df4 =d.read_csv('Meta_vg_4.csv')
# chage release date to be only year of release
df0['Year'] = pd.DatetimeIndex(df0['Year']).year
df1['Year'] = pd.DatetimeIndex(df1['Year']).year
df2['Year'] = pd.DatetimeIndex(df2['Year']).year
df3['Year'] = pd.DatetimeIndex(df3['Year']).year
df4['Year'] = pd.DatetimeIndex(df4['Year']).year

# concatenate all different files of metacritics and remove duplicate rows
df = pd.concat([df0,df1,df2, df3, df4]).drop_duplicates(subset = ['Name', '

# change non numeric values in Meta_Scor and User_Score columns to NaN and

df['User_Score'] = pd.to_numeric(df['User_Score'], errors='coerce')
df['Meta_Score'] = pd.to_numeric(df['Meta_Score'], errors='coerce')

# drop rows with Nan values
df.dropna( inplace= True)

# save final file to csv file
df.to_csv('Meta_vg.csv',sep=',', encoding='utf-8', index=False)
(df.Name=='WII SPORTS').sum()
```

Merge the original Vgame_sales with metacritic dataset

In [ ]:
```
# upload the meta critic data

meta =pd.read_csv('Meta_vg.csv')
# upload the vedio game sales data

vgsl = pd.read_csv('vgsales.csv')
# Standarize the values the Name and Platform columns to Upper letters

meta["Name"] = meta["Name"].str.normalize('NFKD').str.encode('ascii', error
vgsl["Name"] = vgsl["Name"].str.normalize('NFKD').str.encode('ascii', error

meta["Platform"] = meta["Platform"].str.normalize('NFKD').str.encode('ascii
vgsl["Platform"] = vgsl["Platform"].str.normalize('NFKD').str.encode('ascii

# merge all rows in both dataset with similar Name, Platform, and Year

vg_meta = vgsl.merge(meta, how='inner', left_on=['Name', 'Platform','Year']

# drop rows with NaN value
vg_meta.dropna(inplace=True)
vg_meta.to_csv('VG_Meta_Score.csv',sep=',', encoding='utf-8', index=False)
```

In [ ]:
```
df = pd.read_csv('VG_Meta_Score.csv')
print(df.shape)
df.head()
```

```
(5044, 13)
```

Out[ ]:

	Rank	Name	Platform	Year	Genre	Publisher	NA_Sales	EU_Sales	JP_Sales	Other_Sales
0	1	WII SPORTS	WII	2006.0	Sports	Nintendo	41.49	29.02	3.77	8.46
1	3	MARIO KART WII	WII	2008.0	Racing	Nintendo	15.85	12.88	3.79	3.31
2	4	WII SPORTS RESORT	WII	2009.0	Sports	Nintendo	15.75	11.01	3.28	2.96
3	7	NEW SUPER MARIO BROS.	DS	2006.0	Platform	Nintendo	11.38	9.23	6.50	2.90
4	9	NEW SUPER MARIO BROS. WII	WII	2009.0	Platform	Nintendo	14.59	7.06	4.70	2.26

Data cleaning

In [ ]:
```
print(df.describe().T)
print(df.isnull().sum())
df.head()
```

```
              count         mean          std      min       25%       50%
\
```

```
Rank          5044.0  6211.623910  4487.369718     1.00  2347.75  5343.50
Year          5044.0  2007.647105     4.000591  1996.00  2005.00  2008.00
NA_Sales      5044.0     0.448953     1.074748     0.00     0.07     0.17
EU_Sales      5044.0     0.272873     0.746828     0.00     0.02     0.08
JP_Sales      5044.0     0.053872     0.251223     0.00     0.00     0.00
Other_Sales   5044.0     0.095535     0.282226     0.00     0.01     0.03
Global_Sales  5044.0     0.871455     2.134602     0.01     0.13     0.35
Meta_Score    5044.0    70.925852    13.576600    17.00    63.00    73.00
User_Score    5044.0    11.268993    15.479091     0.50     6.60     7.60

                  75%       max
Rank          9518.75  16580.00
Year          2011.00   2016.00
NA_Sales         0.45     41.49
EU_Sales         0.25     29.02
JP_Sales         0.01      6.50
Other_Sales      0.09     10.57
Global_Sales     0.89     82.74
Meta_Score      81.00     99.00
User_Score       8.30     92.00
Rank             0
Name             0
Platform         0
Year             0
Genre            0
Publisher        0
NA_Sales         0
EU_Sales         0
JP_Sales         0
Other_Sales      0
Global_Sales     0
Meta_Score       0
User_Score       0
dtype: int64
```

Out[ ]:

	Rank	Name	Platform	Year	Genre	Publisher	NA_Sales	EU_Sales	JP_Sales	Other_Sales
0	1	WII SPORTS	WII	2006.0	Sports	Nintendo	41.49	29.02	3.77	8.46
1	3	MARIO KART WII	WII	2008.0	Racing	Nintendo	15.85	12.88	3.79	3.31
2	4	WII SPORTS RESORT	WII	2009.0	Sports	Nintendo	15.75	11.01	3.28	2.96
3	7	NEW SUPER MARIO BROS.	DS	2006.0	Platform	Nintendo	11.38	9.23	6.50	2.90
4	9	NEW SUPER MARIO BROS. WII	WII	2009.0	Platform	Nintendo	14.59	7.06	4.70	2.26

In [ ]:
```
df.describe()
```

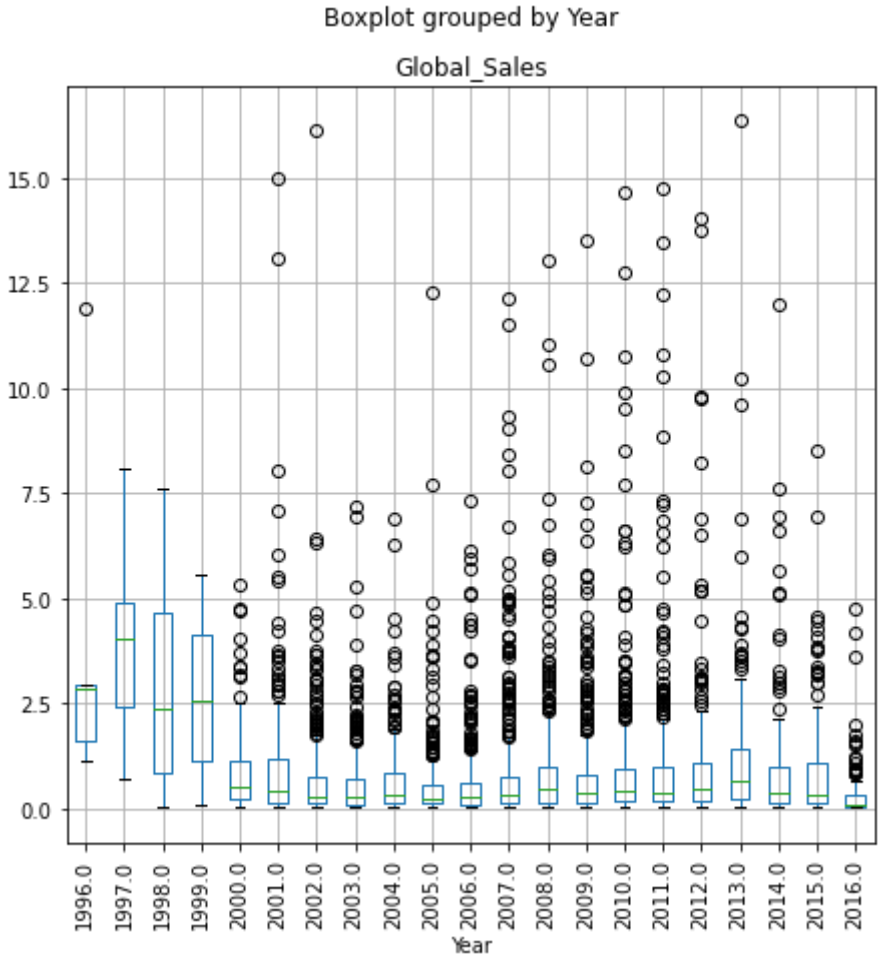
Out[ ]:

	Rank	Year	NA_Sales	EU_Sales	JP_Sales	Other_Sales	Global_Sale
count	5044.000000	5044.000000	5044.000000	5044.000000	5044.000000	5044.000000	5044.00000
mean	6211.623910	2007.647105	0.448953	0.272873	0.053872	0.095535	0.87145

	Rank	Year	NA_Sales	EU_Sales	JP_Sales	Other_Sales	Global_Sale
std	4487.369718	4.000591	1.074748	0.746828	0.251223	0.282226	2.13460
min	1.000000	1996.000000	0.000000	0.000000	0.000000	0.000000	0.01000
25%	2347.750000	2005.000000	0.070000	0.020000	0.000000	0.010000	0.13000
50%	5343.500000	2008.000000	0.170000	0.080000	0.000000	0.030000	0.35000
75%	9518.750000	2011.000000	0.450000	0.250000	0.010000	0.090000	0.89000

Add boxplots

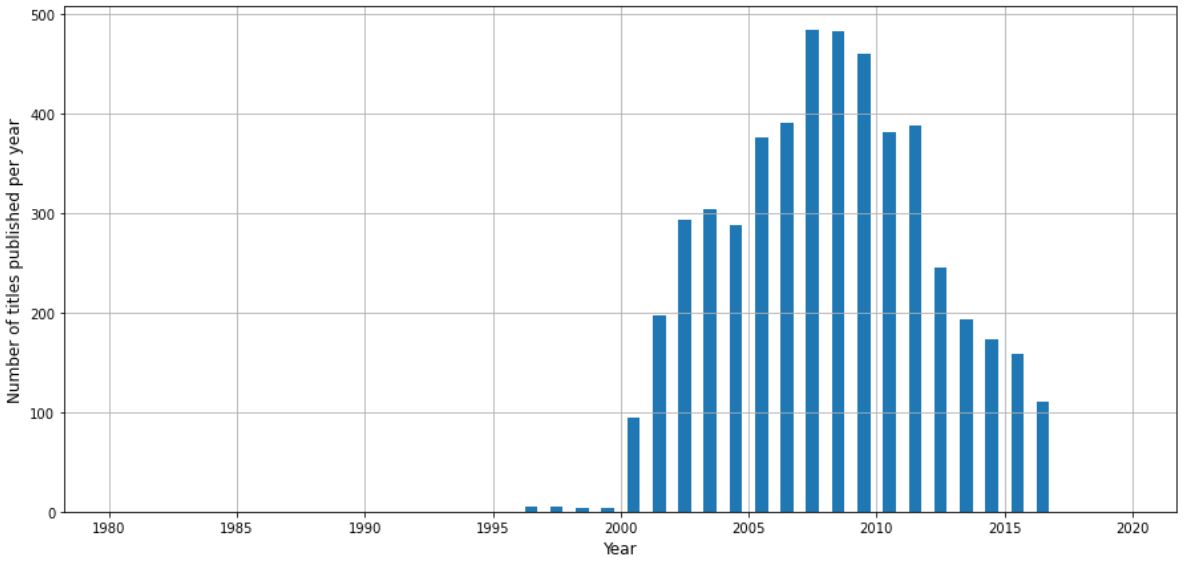
In [ ]:
```
df.drop(df[df['Global_Sales'] > 20].index, inplace = True)
df.boxplot(column= 'Global_Sales', by='Year', figsize= (7,7))
plt.xticks(rotation=90);
```



In [ ]:
```
df.Global_Sales[df['Year']==2009.0].sum()
```

Out[ ]: 342.66

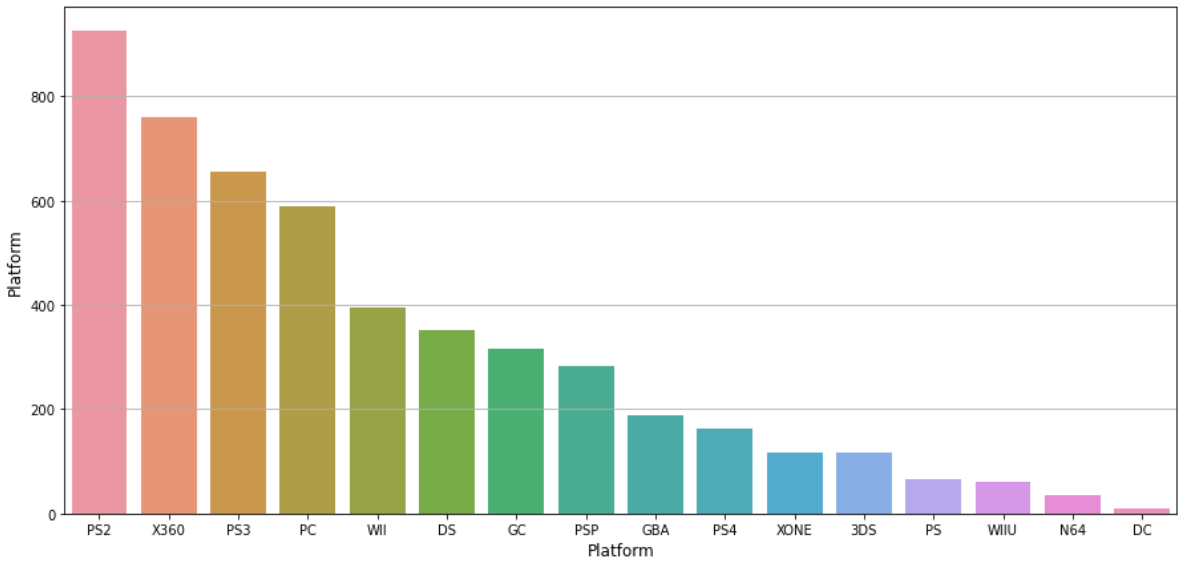
In [ ]:
```
plt.figure(figsize=(15,7))
plt.xlabel('Year',fontsize=12)
plt.ylabel('Number of titles published per year', fontsize=12)
plt.hist(df['Year'],bins=[a for a in range(1980,2021)],rwidth=0.5)
plt.grid()
plt.show()
```



In [ ]:
```
plt.figure(figsize=(15,7))
plt.xlabel('Platform',fontsize=12)
plt.ylabel('Number of titles published per platform', fontsize=12)
plt.grid()
sns.barplot(df['Platform'].value_counts().index, df['Platform'].value_count
```

```
C:\Users\cgals\anaconda3\lib\site-packages\seaborn\_decorators.py:36: Futur
eWarning: Pass the following variables as keyword args: x, y. From version
0.12, the only valid positional argument will be `data`, and passing other
arguments without an explicit keyword will result in an error or misinterpr
etation.
  warnings.warn(
```

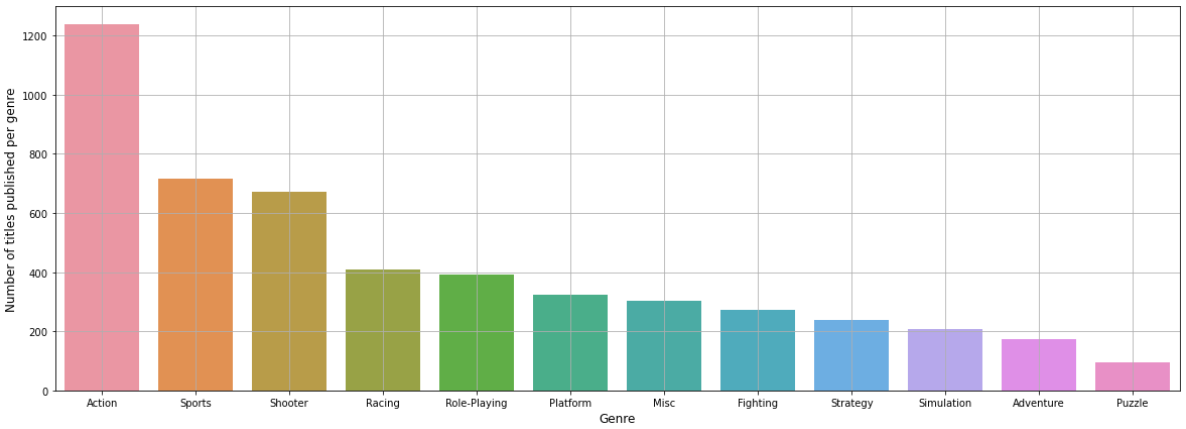
Out[ ]: <AxesSubplot:xlabel='Platform', ylabel='Platform'>



In [ ]:
```
plt.figure(figsize=(20,7))
sns.barplot(df['Genre'].value_counts().index,df['Genre'].value_counts())
plt.xlabel('Genre',fontsize=12)
plt.ylabel('Number of titles published per genre', fontsize=12)
plt.xticks(fontsize=10)
plt.yticks(fontsize=10)
plt.grid()
plt.show()
```

```
C:\Users\cgals\anaconda3\lib\site-packages\seaborn\_decorators.py:36: Futur
eWarning: Pass the following variables as keyword args: x, y. From version
```

```
0.12, the only valid positional argument will be `data`, and passing other
arguments without an explicit keyword will result in an error or misinterpr
etation.
  warnings.warn(
```

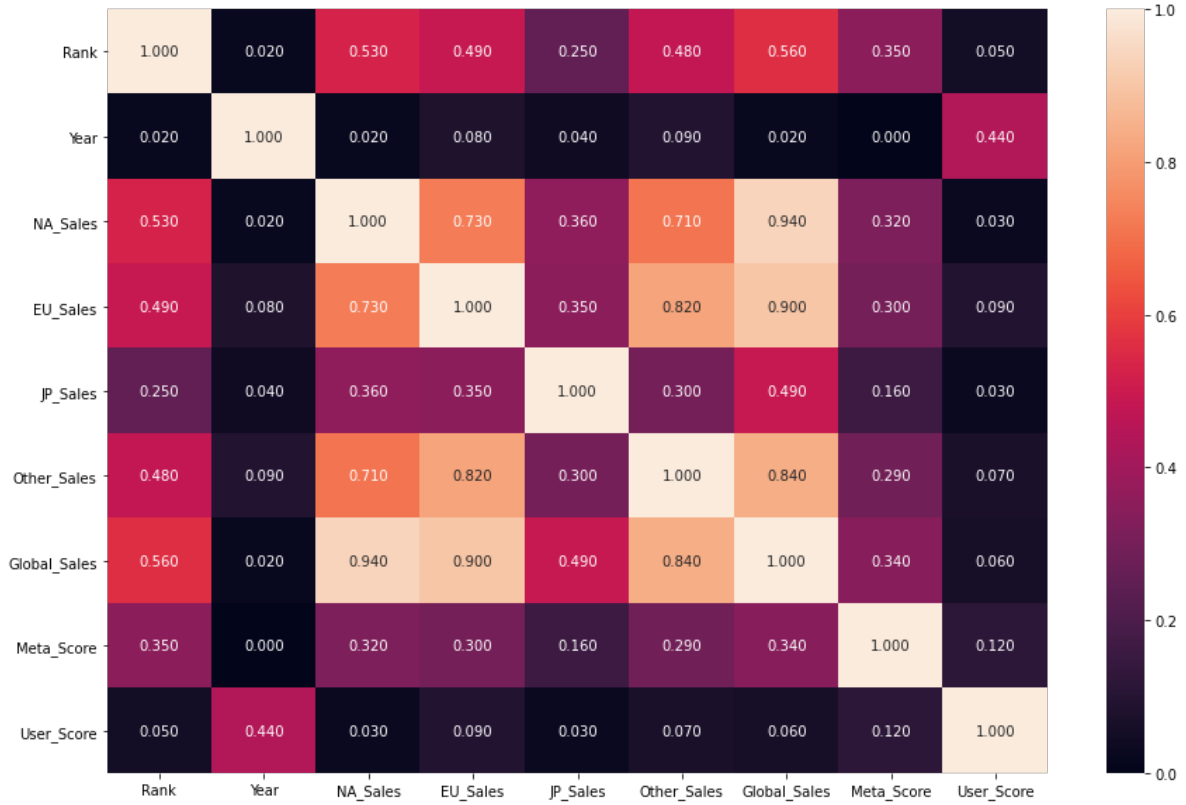


Variables Correlation

In [ ]:

```
import matplotlib.pyplot as plt
import seaborn as sns

plt.figure(figsize=(15,10))
sns.heatmap(df.corr().abs().round(2),annot=True,fmt= '.3f')
```

Out[ ]: <AxesSubplot:>



Based on correlation heat map only the sales numbers are highly correlated with each other. Higher correlations between sales variables help us to understand the global nature of the video game industry, so success on one continent usually means success on another as well. This finding support selecting the Global_Sales variable to be used as a dependent variable.

In [ ]:
```
cat_var = df.select_dtypes(exclude = np.number).columns
df[cat_var].nunique()
```

Out[ ]:
```
Name          3336
Platform        16
Genre           12
Publisher      204
dtype: int64
```

Now we have to deal with categorical values in our dataset. We need to change the following independent variables to dummy variables:

Platform, Genre, Publisher, and Name

However, As we can see Name and Publisher variables have so many unique values which will cause a massive influx of features. So so we will drop these two variable and create dummies only for Platform and Genre categorical variables

In [ ]:
```
print(df.shape) #pre-dummies shape
df1 = pd.get_dummies(data=df, columns=['Platform', 'Genre'])
print(df1.shape) #post-dummies shape
df1.head() #Check to verify that dummies are ok
```

```
(5034, 13)
(5034, 39)
```

Out[ ]:

	Rank	Name	Year	Publisher	NA_Sales	EU_Sales	JP_Sales	Other_Sales	Global_Sale
10	24	GRAND THEFT AUTO V	2013.0	Take-Two Interactive	9.63	5.31	0.06	1.38	16.3
11	25	GRAND THEFT AUTO: VICE CITY	2002.0	Take-Two Interactive	8.41	5.49	0.47	1.78	16.1
12	29	GRAN TURISMO 3: A-SPEC	2001.0	Sony Computer Entertainment	6.85	5.09	1.87	1.16	14.9
13	30	CALL OF DUTY: MODERN WARFARE 3	2011.0	Activision	9.03	4.28	0.13	1.32	14.7
14	32	CALL OF DUTY: BLACK OPS	2010.0	Activision	9.67	3.73	0.11	1.13	14.6

5 rows × 39 columns

Let's now drop the variables from our model. These include:

Rank Name Publisher NA_Sales EU_Sales JP_Sales Other_Sales

In [ ]:
```
df1 = df1.drop(['Rank', 'Name', 'Publisher', 'NA_Sales', 'EU_Sales', 'JP_Sa
```

- Center and reduce the numeric variables of the data frame, using an instance of the class preprocessing.StandardScaler

In [ ]:
```
num_var = df1.select_dtypes(include=np.number).columns

scaler = StandardScaler()
df1[num_var] = pd.DataFrame(scaler.fit_transform(df1[num_var]), index = df1
```

Let's define data and target and then split into training set (X_train, y_train) and a test set (X_test, y_test), with 20% of the original data for the test

In [ ]:
```
target= df1.Global_Sales
data = df1.drop('Global_Sales', axis=1)
X_train, X_test, y_train, y_test = train_test_split(data, target, test_size
Y_train = np.log1p(y_train)
Y_test = np.log1p(y_test)
```

In [ ]:
```
from sklearn.model_selection import GridSearchCV
from sklearn.ensemble import GradientBoostingRegressor

#Parameter grid for Gradient Boosting Regressor
param_grid_gbr = [
    {'max_features': [6, 8, 10, 12], 'max_depth': [5, 7, 9]}]

Y_train = np.log1p(y_train)
Y_test = np.log1p(y_test)

grid_search_gbr = GridSearchCV(GradientBoostingRegressor(n_estimators=200),
grid_search_gbr.fit(X_train, Y_train)
print("Best parameters: {}".format(grid_search_gbr.best_params_))
gbr_best_cross_val_score = (np.sqrt(-grid_search_gbr.best_score_))
print("Best cross-validation score: {:.2f}".format(np.expm1(gbr_best_cross_
gbr_score = np.sqrt(-grid_search_gbr.score(X_test, Y_test))
print("Test set score: {:.2f}".format(np.expm1(gbr_score)))
```

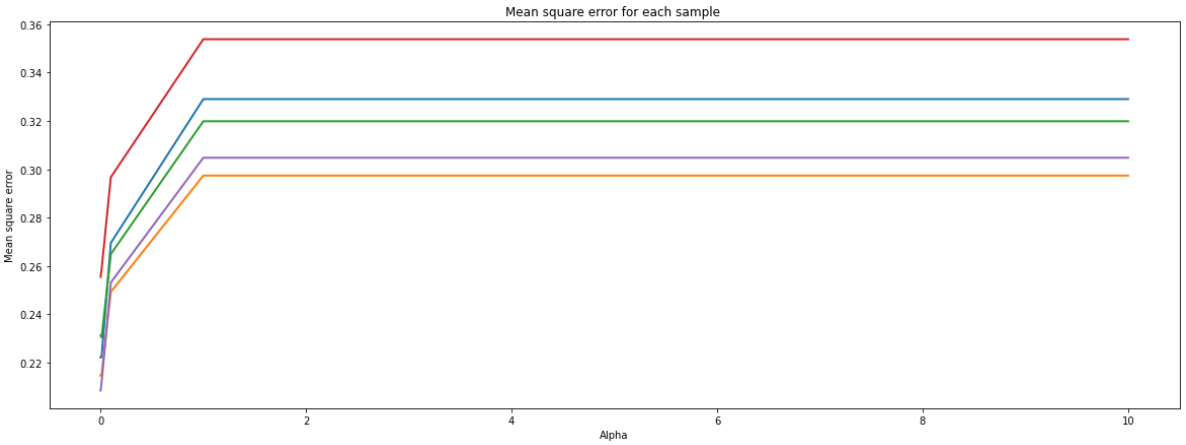
```
Best parameters: {'max_depth': 5, 'max_features': 8}
Best cross-validation score: 0.55
Test set score: 0.57
```

In [ ]:
```
l_reg = LinearRegression()
l_reg.fit(X_train, Y_train)
pred_test = l_reg.predict(X_test)

print("score train :", l_reg.score(X_train, Y_train))
print("score test :", l_reg.score(X_test, Y_test))
print("mse test:", np.sqrt(mean_squared_error(pred_test, Y_test)))
```

```
score train : 0.30265479510738014
score test : 0.307159211381398
mse test: 0.495807070334269
```

In [ ]:

```
alpha = [10, 1, 0.1, 0.01, 0.001, 0.0005]
model_lasso = LassoCV(alphas=alpha, cv=5).fit(X_train, Y_train)

alphas = model_lasso.alphas_
plt.figure(figsize=(20,7))
plt.plot(alphas, model_lasso.mse_path_, '-', lw=2)

plt.xlabel('Alpha')
plt.ylabel('Mean square error')
plt.title('Mean square error for each sample ')
plt.show()
print('the best value of alpha is',model_lasso.alpha_)
```




```
the best value of alpha is 0.0005
```

In [ ]:

```
pred_test = model_lasso.predict(X_test)

print("score test:", model_lasso.score(X_test, Y_test))
print("mse test:", np.sqrt(mean_squared_error(pred_test, Y_test)))
print('The number of variables kept =',(model_lasso.coef_!=0).sum())
print('The number of variables eliminated =',(model_lasso.coef_==0).sum())
```

```
score test: 0.30712852965059156
mse test: 0.49581804836190774
The number of variables kept = 29
The number of variables eliminated = 2
```

Change from Regression to Classification model

In [ ]:

```
target.describe()
bin_labels_4 = [ 'Silver', 'Gold', 'Platinum', 'Diamond']

# bin_labels_5 = ['Bronze', 'Silver', 'Gold', 'Platinum', 'Diamond']
# bin_labels_10 = ['0', '1','2','3','4','5','6','7','8','9', '10','11','12

label = pd.qcut(target, q=4, retbins = True, labels=bin_labels_4)
X_train1, X_test1, y_train1, y_test1 = train_test_split(data, label[0], tes
```

In [ ]:
```
# Classification using LogisticRegession
lg = LogisticRegression(max_iter = 1000, n_jobs = -1)
lg.fit(X_train1, y_train1)
lg_sc=[]
lg_sc.append(lg.score(X_test1, y_test1))
lg_sc
```

Out[ ]: [0.48361469712015887]

In [ ]:
```
# Classification using SVM
from sklearn.svm import SVC
svm = SVC()
svm.fit(X_train1, y_train1)
svm_sc=[]
svm_sc.append(svm.score(X_test1, y_test1))
svm_sc
```

Out[ ]: [0.44091360476663355]

Impact of outliers

In [ ]:
```
df2=df[df['Year']>1999]
df2 = pd.get_dummies(data=df2, columns=['Platform', 'Genre'])
df2 = df2.drop(['Rank', 'Name', 'Publisher', 'NA_Sales', 'EU_Sales', 'JP_Sa

# normalisation
num_var = df2.select_dtypes(include=np.number).columns
scaler = StandardScaler()
df2[num_var] = pd.DataFrame(scaler.fit_transform(df2[num_var]), index = df2

label = pd.qcut(target, q=4, retbins = True, labels=bin_labels_4)
target= df2.Global_Sales
data = df2.drop('Global_Sales', axis=1)
label = pd.qcut(target, q=4, retbins = True, labels=bin_labels_4)
# Train the model
X_train1, X_test1, y_train1, y_test1 = train_test_split(data, label[0], tes

# L R
lg.fit(X_train1, y_train1)
lg_sc.append(lg.score(X_test1, y_test1))
print(lg_sc)


# SVM
svm.fit(X_train1, y_train1)
svm_sc.append(svm.score(X_test1, y_test1))
print(svm_sc)
```

```
[0.48361469712015887, 0.45717131474103584]
[0.44091360476663355, 0.4312749003984064]
```

In [ ]:
```
for i in range (19,7,-1):

    df3=df[df['Global_Sales'] <= i]
    df3 = pd.get_dummies(data=df3, columns=['Platform', 'Genre'])
    df3 = df3.drop(['Rank', 'Name', 'Publisher', 'NA_Sales', 'EU_Sales', 'J

    # normalisation
    num_var = df3.select_dtypes(include=np.number).columns
    scaler = StandardScaler()
    df3[num_var] = pd.DataFrame(scaler.fit_transform(df3[num_var]), index =

    label = pd.qcut(target, q=4, retbins = True, labels=bin_labels_4)
    target= df3.Global_Sales
    data = df3.drop('Global_Sales', axis=1)
    label = pd.qcut(target, q=4, retbins = True, labels=bin_labels_4)
    # Train the model
    X_train1, X_test1, y_train1, y_test1 = train_test_split(data, label[0],

    # L R
    lg.fit(X_train1, y_train1)
    lg_sc.append(lg.score(X_test1, y_test1))



    # SVM
    svm.fit(X_train1, y_train1)
    svm_sc.append(svm.score(X_test1, y_test1))


print(lg_sc)
print(svm_sc)
```

```
[0.48361469712015887, 0.45717131474103584, 0.48361469712015887, 0.483614697
12015887, 0.48361469712015887, 0.45183714001986097, 0.45183714001986097, 0.
47713717693836977, 0.4646766169154229, 0.4701195219123506, 0.45363908275174
475, 0.45209580838323354, 0.4485514485514486, 0.45345345345345345]
[0.44091360476663355, 0.4312749003984064, 0.44091360476663355, 0.4409136047
6663355, 0.44091360476663355, 0.4438927507447865, 0.4438927507447865, 0.457
2564612326044, 0.46766169154228854, 0.4442231075697211, 0.4596211365902293,
0.4620758483033932, 0.46053946053946054, 0.4444444444444444]
```

In [ ]:
```
label[0]
```

Out[ ]:
```
51      Diamond
52      Diamond
53      Diamond
54      Diamond
55      Diamond
         ...
5039     Silver
5040     Silver
5041     Silver
5042     Silver
5043     Silver
Name: Global_Sales, Length: 4993, dtype: category
Categories (4, object): ['Silver' < 'Gold' < 'Platinum' < 'Diamond']
```

In [ ]:

```
GSCV_score=[]
for i in range (19,7,-1):
    df4=df[df['Global_Sales'] <= i]
    df4 = pd.get_dummies(data=df4, columns=['Platform', 'Genre'])
    df4 = df4.drop(['Rank', 'Name', 'Publisher', 'NA_Sales', 'EU_Sales', 'J
    target= df4.Global_Sales
    data = df4.drop('Global_Sales', axis=1)
    X_train, X_test, y_train, y_test = train_test_split(data, target, test_
    Y_train = np.log1p(y_train)
    Y_test = np.log1p(y_test)

    #Parameter grid for Gradient Boosting Regressor
    param_grid_gbr = [
        {'max_features': [6, 8, 10, 12], 'max_depth': [5, 7, 9]}]

    grid_search_gbr = GridSearchCV(GradientBoostingRegressor(n_estimators=2
    grid_search_gbr.fit(X_train, Y_train)
    print("Best parameters: {}".format(grid_search_gbr.best_params_))
    gbr_best_cross_val_score = (np.sqrt(-grid_search_gbr.best_score_))
    print("Best cross-validation score: {:.2f}".format(np.expm1(gbr_best_cr
    gbr_score = np.sqrt(-grid_search_gbr.score(X_test, Y_test))
    print(i)
    print("Test set score: {:.4f}".format(np.expm1(gbr_score)))
    GSCV_score.append(np.expm1(gbr_score))
```

```
Best parameters: {'max_depth': 5, 'max_features': 8}
Best cross-validation score: 0.42
19
Test set score: 0.4436
Best parameters: {'max_depth': 5, 'max_features': 6}
Best cross-validation score: 0.42
18
Test set score: 0.4398
Best parameters: {'max_depth': 5, 'max_features': 8}
Best cross-validation score: 0.42
17
Test set score: 0.4425
Best parameters: {'max_depth': 5, 'max_features': 6}
Best cross-validation score: 0.42
16
Test set score: 0.4356
Best parameters: {'max_depth': 5, 'max_features': 10}
Best cross-validation score: 0.42
15
Test set score: 0.4368
Best parameters: {'max_depth': 5, 'max_features': 8}
Best cross-validation score: 0.42
14
Test set score: 0.4337
Best parameters: {'max_depth': 5, 'max_features': 6}
Best cross-validation score: 0.42
13
Test set score: 0.4227
Best parameters: {'max_depth': 5, 'max_features': 12}
Best cross-validation score: 0.42
12
Test set score: 0.4150
Best parameters: {'max_depth': 5, 'max_features': 6}
Best cross-validation score: 0.41
11
Test set score: 0.4302
Best parameters: {'max_depth': 5, 'max_features': 6}
Best cross-validation score: 0.41
10
Test set score: 0.4446
```

```
Best parameters: {'max_depth': 5, 'max_features': 12}
Best cross-validation score: 0.41
9
Test set score: 0.4020
Best parameters: {'max_depth': 5, 'max_features': 6}
Best cross-validation score: 0.40
8
```

In [ ]:
```
GSCV_score
```

Out[ ]:
```
[0.4436413102978027,
 0.4397501326376228,
 0.44248562492466775,
 0.4356033055402448,
 0.4368353855232752,
 0.4336871855635897,
 0.42265780846137974,
 0.41495085112927665,
 0.4301705628743069,
 0.4445677885968156,
 0.40203635249554476,
 0.4111242729535207]
```

Result shows that the gradual removal of outliers has no meaningful impact on results

10.4 PYGSL.ipynb

A4_PYGSL.ipynb

Video_Games_Sales_as_at_22_Dec_2016.csv

In [ ]:

In [ ]:
```
# Importing the required libraries
import pandas as pd
import numpy as np

# Importing the dataset
df = pd.read_csv('Video_Games_Sales_as_at_22_Dec_2016.csv')

# Dropping certain less important features
df.drop(columns = ['Year_of_Release', 'Publisher', 'Platform', 'EU_Sales',

# To view the columns with missing values
print('Feature name || Total missing values')
print(df.isna().sum())
```

```
Feature name || Total missing values
Name                2
Genre               2
NA_Sales            0
Global_Sales        0
Critic_Score     8582
Critic_Count     8582
User_Score       9129
User_Count       9129
Rating           6769
dtype: int64
```

In [ ]:
```
# drop two rows with missing Name values
df.dropna(subset =['Name'], axis=0, inplace=True)
df.head()
```

Out[ ]:

	Name	Genre	NA_Sales	Global_Sales	Critic_Score	Critic_Count	User_Score	User_Cou
0	Wii Sports	Sports	41.36	82.53	76.0	51.0	8.0	32
1	Super Mario Bros.	Platform	29.08	40.24	NaN	NaN	NaN	N
2	Mario Kart Wii	Racing	15.68	35.52	82.0	73.0	8.3	70
3	Wii Sports Resort	Sports	15.61	32.77	80.0	73.0	8.0	19
4	Pokemon Red/Pokemon Blue	Role-Playing	11.27	31.37	NaN	NaN	NaN	N

Replacing the missing data with substituted values. Here, we'll use the SimpleImputer class from the scikit-learn library to impute the columns with missing values and to impute the columns with values of type string, we'll use CategoricalImputer from feature_engine and replace the missing values with 'NA' i.e. Not Available.

In [ ]:
```
from sklearn.impute import SimpleImputer
imputer = SimpleImputer(strategy='mean')
df.loc[:, [ 'Critic_Score', 'Critic_Count', 'User_Score', 'User_Count']] =

from feature_engine.imputation import CategoricalImputer

categorical_imputer = CategoricalImputer( imputation_method = 'frequent')
df.loc[:, ['Genre','Rating' ]] = categorical_imputer.fit_transform(df.loc[:
```

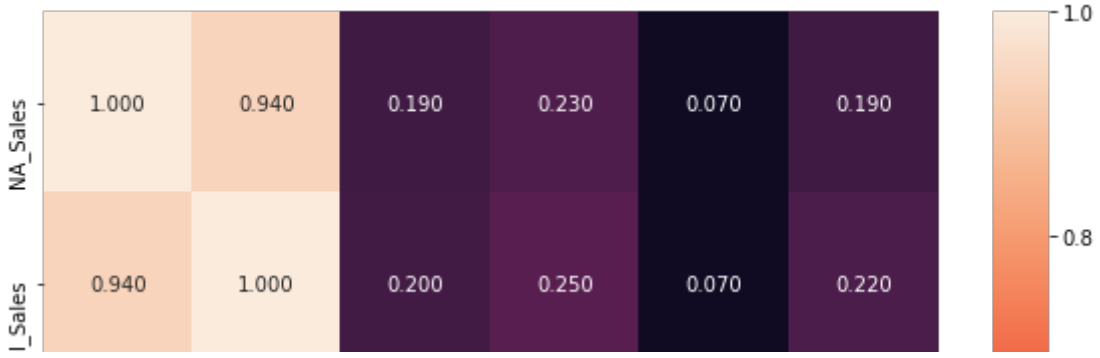
In [ ]:
```
# To view the columns with missing values
print('Feature name || Total missing values')
print(df.isna().sum())
```

```
Feature name || Total missing values
Name             0
Genre            0
NA_Sales         0
Global_Sales     0
Critic_Score     0
Critic_Count     0
User_Score       0
User_Count       0
Rating           0
dtype: int64
```

Variables Correlation

In [ ]:
```
import seaborn as sns
import matplotlib.pyplot as plt

plt.figure(figsize=(10,10))
sns.heatmap(df.corr().abs().round(2),annot=True,fmt= '.3f')
```

Out[ ]: <AxesSubplot:>



Splitting the dataset into Train & Test sets

In [ ]:

```
X = df.drop('Global_Sales', axis=1) # copy dataset without the target colum
y = df['Global_Sales'] # copy the target variable Global_Sales to y

# Splitting the dataset into Train and Test sets
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.3,

# Saving name of the games in training and test set
games_in_training_set = X_train.loc[:,'Name']
games_in_test_set = X_test.loc[:,'Name']

# Dropping the column that contains the name of the games
X_train = X_train.drop('Name', axis=1)
X_test = X_test.drop('Name', axis=1)
```

Here, we initialize 'X' and 'y' where 'X' is the set of independent variables and 'y' the target variable i.e. the Global_Sales. We save the name of the games in a separate array named 'games_in_training_set' and 'games_in_test_set' as these names will not be of much help when predicting the global sales.

OneHotEncoding We encode the categorical columns of 'X' using ColumnTransformer and OneHotEncoder from the scikit-learn library. This will assign one separate column to each category present in a categorical column of 'X'.

In [ ]:

```
from sklearn.compose import ColumnTransformer
from sklearn.preprocessing import OneHotEncoder

ct = ColumnTransformer(transformers = [('encoder', OneHotEncoder(), ['Genre
X_train = ct.fit_transform(X_train)
X_test = ct.transform(X_test)
```

Building the models We'll implement our model i.e. the regressor using XGBRegressor (where XGB stands for extreme gradient boosting). XGBoost is an ensemble machine learning algorithm based on decision trees similar to the RandomForest algorithm. However, unlike RandomForest that makes use of fully grown trees, XGBoost combines trees that are not too deep. Also, the number of trees combined in XGBoost is more in comparison to RandomForest. Ensemble algorithms effectively combine weak learners to produce a strong learner. XGBoost has additional features focused on performance and speed when compared to gradient boosting.

In [ ]:
```
from xgboost import XGBRegressor
model = XGBRegressor(n_estimators = 200, learning_rate= 0.08)
model.fit(X_train, y_train)
```

Out[ ]:
```
XGBRegressor(base_score=0.5, booster='gbtree', colsample_bylevel=1,
             colsample_bynode=1, colsample_bytree=1, gamma=0, gpu_id=-1,
             importance_type='gain', interaction_constraints='',
             learning_rate=0.08, max_delta_step=0, max_depth=6,
             min_child_weight=1, missing=nan, monotone_constraints='()',
             n_estimators=200, n_jobs=8, num_parallel_tree=1, random_state=
0,
             reg_alpha=0, reg_lambda=1, scale_pos_weight=1, subsample=1,
             tree_method='exact', validate_parameters=1, verbosity=None)
```

Making predictions on the Test set Global Sales i.e. the target variable 'y' for the games in the test set is predicted using the model.predict() method.

In [ ]:
```
# Predicting test set results
y_pred = model.predict(X_test)

# Visualising actual and predicted sales
predictions = np.concatenate([games_in_test_set.values.reshape(-1, 1), y_pr
predictions = pd.DataFrame(predictions, columns = ['Name', 'Predicted_Globa
```

In [ ]:
```
model.score(X_test, y_test)
```

Out[ ]:
```
0.7993053137995038
```

In [ ]:
```
predictions
```

Out[ ]:

	Name	Predicted_Global_Sales	Actual_Global_Sales
0	Mario Party 3	1.243476	1.91
1	Bullet Girls	0.108133	0.06
2	Injustice: Gods Among Us	0.205462	0.13
3	Rhythm Heaven	1.541074	3.11
4	Broken Sword II: The Smoking Mirror	0.070998	0.06
...	...	...	...
5011	Bakusou Kyoudai Let's & Go!! Eternal Wings	0.138477	0.05
5012	Darkened Skye	0.021511	0.01
5013	Burnout Paradise: The Ultimate Box	0.138477	0.07
5014	Soul Edge	1.151762	1.59
5015	Whiteout	0.020969	0.03

5016 rows × 3 columns

Evaluating model performance We'll use r2_score and root mean squared error (RMSE) to evaluate the model performance where closer the r2_score is to 1 & lower the magnitude of RMSE, the better the model is.

In [ ]:
```
from sklearn.metrics import r2_score, mean_squared_error
import math
r2_score = r2_score(y_test, y_pred)
rmse = math.sqrt(mean_squared_error(y_test, y_pred))

print(f"r2 score of the model : {r2_score:.3f}")
print(f"Root Mean Squared Error of the model : {rmse:.3f}")
```

```
r2 score of the model : 0.799
Root Mean Squared Error of the model : 0.689
```

GradientBoostingRegressor model

In [ ]:
```
from sklearn.model_selection import GridSearchCV
from sklearn.ensemble import GradientBoostingRegressor

#Parameter grid for Gradient Boosting Regressor
param_grid_gbr = [
    {'n_estimators': [200, 500],'max_features': [6, 8, 10, 12], 'max_depth'

grid_search_gbr = GridSearchCV(GradientBoostingRegressor(n_estimators=200),
grid_search_gbr.fit(X_train, y_train)
print("Best parameters: {}".format(grid_search_gbr.best_params_))
gbr_best_cross_val_score = (np.sqrt(-grid_search_gbr.best_score_))
print("Best cross-validation score: {:.2f}".format(gbr_best_cross_val_score
gbr_score = np.sqrt(-grid_search_gbr.score(X_test, y_test))
print("Test set score: {:.2f}".format(gbr_score))
```

```
Best parameters: {'max_depth': 5, 'max_features': 12, 'n_estimators': 200}
Best cross-validation score: 0.72
Test set score: 0.55
```

In [ ]:
```
gbr_best_cross_val_score = (np.sqrt(-grid_search_gbr.best_score_))
print("Best cross-validation score: {:.2f}".format(gbr_best_cross_val_score
gbr_score = np.sqrt(-grid_search_gbr.score(X_test, y_test))
print("Test set score: {:.2f}".format(gbr_score))
```

```
Best cross-validation score: 0.72
Test set score: 0.55
```

LinearRegression Model

In [ ]:
```
from sklearn.linear_model import LinearRegression

l_reg = LinearRegression()
l_reg.fit(X_train, y_train)
pred_test = l_reg.predict(X_test)

print("score train :", l_reg.score(X_train, y_train))
print("score test :", l_reg.score(X_test, y_test))
print("mse test:", np.sqrt(mean_squared_error(pred_test, y_test)))
```

```
score train : 0.8853231790170023
score test : 0.8975412961770481
mse test: 0.49222718713635094
```

Change from Regression to Classification model

In [ ]:
```
print(y.describe())
bin_labels_3 = [ 'Silver', 'Gold', 'Platinum']

bin_labels_4 = [ 'Silver', 'Gold', 'Platinum', 'Diamond']

bin_labels_5 = ['Bronze', 'Silver', 'Gold', 'Platinum', 'Diamond']
bin_labels_10 = ['0', '1','2','3','4','5','6','7','8','9']#, '10','11','12

label = pd.qcut(y, q=3, retbins = True, labels=bin_labels_3)
X_train1, X_test1, y_train1, y_test1 = train_test_split(X, label[0], test_s

# Saving name of the games in training and test set
games_in_training_set = X_train1.loc[:,'Name']
games_in_test_set = X_test1.loc[:,'Name']

# Dropping the column that contains the name of the games
X_train1 = X_train1.drop('Name', axis=1)
X_test1 = X_test1.drop('Name', axis=1)

from sklearn.compose import ColumnTransformer
from sklearn.preprocessing import OneHotEncoder

ct = ColumnTransformer(transformers = [('encoder', OneHotEncoder(), ['Genre
X_train1 = ct.fit_transform(X_train1)
X_test1 = ct.transform(X_test1)
```

```
count    16717.000000
mean         0.533462
std          1.547956
min          0.010000
25%          0.060000
50%          0.170000
75%          0.470000
max         82.530000
Name: Global_Sales, dtype: float64
```

In [ ]:
```
# Classification using LogisticRegession
from sklearn.linear_model import LogisticRegression

lg = LogisticRegression(max_iter = 1000, n_jobs = -1)
lg.fit(X_train1, y_train1)
lg.score(X_test1, y_test1)
```

Out[ ]:
```
0.7366427432216905
```

10.5 PYGALSAL_CNN.ipynb


A6_PYGALSAL_CNN.ipynb


df_clean.csv

Try a CNN model to check if better results can be achieved

In [ ]:
```
import pandas as pd
import numpy as np
```

In [ ]:
```
df = pd.read_csv('df_clean.csv')
print(df.shape)
df.head()
```

```
(5044, 8)
```

Out[ ]:

	Name	Platform	Year	Genre	Publisher	Global_Sales	Meta_Score	User_Score
0	WII SPORTS	WII	2006.0	Sports	Nintendo	82.74	76	8.1
1	MARIO KART WII	WII	2008.0	Racing	Nintendo	35.82	82	8.4
2	WII SPORTS RESORT	WII	2009.0	Sports	Nintendo	33.00	80	8.2
3	NEW SUPER MARIO BROS.	DS	2006.0	Platform	Nintendo	30.01	89	8.5
4	NEW SUPER MARIO BROS. WII	WII	2009.0	Platform	Nintendo	28.62	87	8.3

In [ ]:
```
from keras.models import Sequential
from keras.layers import Dense
```

Create the bins according to business requirements

In [ ]:
```
df1=df
cut_labels_4 = [0,1,2,3] #['silver', 'gold', 'platinum', 'diamond'],
# cut_bins = [0, 1, 2.5, 5, 100]
df1['target'] = pd.qcut(df1['Global_Sales'], q=4, labels=cut_labels_4)
print(df1.head())
df1['target'].value_counts()
```

```
                       Name Platform    Year     Genre Publisher  \
0                WII SPORTS      WII  2006.0    Sports  Nintendo
1             MARIO KART WII      WII  2008.0    Racing  Nintendo
2          WII SPORTS RESORT      WII  2009.0    Sports  Nintendo
3      NEW SUPER MARIO BROS.       DS  2006.0  Platform  Nintendo
4  NEW SUPER MARIO BROS. WII      WII  2009.0  Platform  Nintendo

   Global_Sales  Meta_Score  User_Score target
0         82.74          76         8.1      3
1         35.82          82         8.4      3
2         33.00          80         8.2      3
3         30.01          89         8.5      3
4         28.62          87         8.3      3
```

Out[ ]:
```
0    1304
1    1251
3    1246
2    1243
```

```
Name: target, dtype: int64
```

In [ ]:
```
df1 = pd.get_dummies(data=df, columns=['Platform', 'Genre', 'Publisher'])
print(df1.shape) #post-dummies shape
df1.head() #Check to verify that dummies are ok
```

```
(5044, 238)
```

Out[ ]:

	Name	Year	Global_Sales	Meta_Score	User_Score	target	Platform_3DS	Platform_DC	Pla
0	WII SPORTS	2006.0	82.74	76	8.1	3	0	0	
1	MARIO KART WII	2008.0	35.82	82	8.4	3	0	0	
2	WII SPORTS RESORT	2009.0	33.00	80	8.2	3	0	0	
3	NEW SUPER MARIO BROS.	2006.0	30.01	89	8.5	3	0	0	
4	NEW SUPER MARIO BROS. WII	2009.0	28.62	87	8.3	3	0	0	

5 rows × 238 columns

In [ ]:
```
df1 = df1.drop(['Name','Global_Sales'], axis=1)
```

In [ ]:
```
df1.head()
```

Out[ ]:

	Year	Meta_Score	User_Score	target	Platform_3DS	Platform_DC	Platform_DS	Platform_GB
0	2006.0	76	8.1	3	0	0	0	
1	2008.0	82	8.4	3	0	0	0	
2	2009.0	80	8.2	3	0	0	0	
3	2006.0	89	8.5	3	0	0	1	
4	2009.0	87	8.3	3	0	0	0	

5 rows × 236 columns

In [ ]:
```
target=df1['target']
feats=df1.drop(['target'],axis=1)
```

In [ ]:
```
from sklearn.model_selection import train_test_split
X_train,X_test,y_train,y_test = train_test_split(feats,target,test_size=0.2
from sklearn import metrics # Pour évaluer les modèles
```

In [ ]:
```
from sklearn.preprocessing import MinMaxScaler
scaler = MinMaxScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)
```

In [ ]:
```
# define the keras model
model = Sequential()
model.add(Dense(64, input_dim=235, activation='relu'))
model.add(Dense(16, activation='relu'))
model.add(Dense(4, activation='softmax'))
# compile the keras model
model.compile(loss='sparse_categorical_crossentropy', optimizer='adam', met
# fit the keras model on the dataset
model.fit(X_train, y_train, epochs=20, batch_size=8, validation_split=0.2)

# make class predictions with the model
predictions = model.predict(X_test)
# Evaluation du modèle
score = model.evaluate(X_test, y_test)
print ("score is: ",score)
# confusion matrix
predict_class = np.argmax(predictions, axis=1)
cnf_matrix = metrics.confusion_matrix(y_test, predict_class)
print(cnf_matrix)
```

```
Epoch 1/20
404/404 [==============================] - 1s 2ms/step - loss: 1.3387 - acc
uracy: 0.3386 - val_loss: 1.2915 - val_accuracy: 0.3556
Epoch 2/20
404/404 [==============================] - 1s 1ms/step - loss: 1.1997 - acc
uracy: 0.4473 - val_loss: 1.1908 - val_accuracy: 0.4659
Epoch 3/20
404/404 [==============================] - 1s 1ms/step - loss: 1.1249 - acc
uracy: 0.4867 - val_loss: 1.1807 - val_accuracy: 0.4672
Epoch 4/20
404/404 [==============================] - 1s 1ms/step - loss: 1.0723 - acc
uracy: 0.5198 - val_loss: 1.1673 - val_accuracy: 0.4845
Epoch 5/20
404/404 [==============================] - 1s 1ms/step - loss: 1.0326 - acc
uracy: 0.5369 - val_loss: 1.1764 - val_accuracy: 0.4845
Epoch 6/20
404/404 [==============================] - 1s 1ms/step - loss: 1.0017 - acc
uracy: 0.5567 - val_loss: 1.1730 - val_accuracy: 0.4796
Epoch 7/20
404/404 [==============================] - 1s 1ms/step - loss: 0.9735 - acc
uracy: 0.5623 - val_loss: 1.1874 - val_accuracy: 0.4870
Epoch 8/20
404/404 [==============================] - 1s 1ms/step - loss: 0.9485 - acc
uracy: 0.5781 - val_loss: 1.1954 - val_accuracy: 0.4696
Epoch 9/20
404/404 [==============================] - 1s 2ms/step - loss: 0.9307 - acc
uracy: 0.5901 - val_loss: 1.1885 - val_accuracy: 0.4808
Epoch 10/20
404/404 [==============================] - 1s 1ms/step - loss: 0.9099 - acc
uracy: 0.6001 - val_loss: 1.2167 - val_accuracy: 0.4796
Epoch 11/20
404/404 [==============================] - 1s 1ms/step - loss: 0.8904 - acc
uracy: 0.6165 - val_loss: 1.2055 - val_accuracy: 0.4870
```

```
Epoch 12/20
404/404 [==============================] - 1s 1ms/step - loss: 0.8763 - acc
uracy: 0.6146 - val_loss: 1.2777 - val_accuracy: 0.4684
Epoch 13/20
404/404 [==============================] - 1s 1ms/step - loss: 0.8571 - acc
uracy: 0.6298 - val_loss: 1.2519 - val_accuracy: 0.4585
Epoch 14/20
404/404 [==============================] - 1s 1ms/step - loss: 0.8439 - acc
uracy: 0.6341 - val_loss: 1.2530 - val_accuracy: 0.4895
Epoch 15/20
404/404 [==============================] - 1s 1ms/step - loss: 0.8350 - acc
uracy: 0.6369 - val_loss: 1.2762 - val_accuracy: 0.4734
Epoch 16/20
404/404 [==============================] - 1s 1ms/step - loss: 0.8187 - acc
uracy: 0.6499 - val_loss: 1.2868 - val_accuracy: 0.4746
Epoch 17/20
404/404 [==============================] - 1s 1ms/step - loss: 0.8141 - acc
uracy: 0.6549 - val_loss: 1.2938 - val_accuracy: 0.4808
Epoch 18/20
404/404 [==============================] - 1s 1ms/step - loss: 0.7971 - acc
uracy: 0.6524 - val_loss: 1.3190 - val_accuracy: 0.4758
Epoch 19/20
404/404 [==============================] - 1s 1ms/step - loss: 0.7874 - acc
uracy: 0.6583 - val_loss: 1.3322 - val_accuracy: 0.4721
Epoch 20/20
404/404 [==============================] - 1s 1ms/step - loss: 0.7762 - acc
uracy: 0.6657 - val_loss: 1.3195 - val_accuracy: 0.4734
32/32 [==============================] - 0s 1ms/step - loss: 1.2332 - accur
acy: 0.5084
score is:  [1.2331687211990356, 0.5084241628646851]
[[154  70  33  13]
 [ 50 104  51  25]
 [ 32  71  82  79]
 [ 16  20  36 173]]
```

10.6 A6_Twitter.ipynb

A6_Twitter.ipynb

Twitter count

In [ ]:
```
import os
import pandas as pd
from datetime import date
```

In [ ]:
```
import snscrape
import snscrape.modules.twitter as sntwitter
import sys
import pandas as pd
# Creating list to append tweet data to
tweets_list2 = []
# Using TwitterSearchScraper to scrape data and append tweets to list
for i,tweet in enumerate(sntwitter.TwitterSearchScraper('Red Dead Redemptic
    # if i>50:
    #     break
    tweets_list2.append([tweet.date, tweet.id, tweet.content, tweet.usernam
    sys.stdout.write('\r'+str(i))
    sys.stdout.flush()
# Creating a dataframe from the tweets list above
tweets_df2 = pd.DataFrame(tweets_list2, columns=['Datetime', 'Tweet Id', 'T
tweets_df2
```

```
6131
```

Out[ ]:

	Datetime	Tweet Id	Text	Username
0	2018-10-24 23:59:41+00:00	1055247487244603392	@Gameplanet_Mex buen día, busco el Red Dead Re...	rfcf
1	2018-10-24 23:59:36+00:00	1055247465794977795	#RedDeadRedemption2 #RedDeadRedemptionII #RDR2...	Archangel491
2	2018-10-24 23:59:32+00:00	1055247452918419456	Catch me drinking one of these while I play Re...	FalsehoodeTV
3	2018-10-24 23:59:11+00:00	1055247363227312128	Going to try to get all my streaming out today...	darksilence669
4	2018-10-24 23:59:11+00:00	1055247361746841600	Vamos pra mais um dia de live com meu consagra...	NILTON_NTN_
...	...	...	...	...
6127	2018-10-24 00:00:56+00:00	1054885415575060481	Fãs fazem "trailer" incrível de RED DEAD REDEM...	maitohorizon
6128	2018-10-24 00:00:47+00:00	1054885378493157376	I liked a @YouTube video https://t.co/YHRf7KhC...	markpierc
6129	2018-10-24 00:00:46+00:00	1054885371794735104	Hope yall ready for @RockstarGames Red dead ...	Homie_da_Clown
6130	2018-10-24 00:00:06+00:00	1054885205176139776	I won 2 achievements in Red Dead Redemption fo...	Wertho_
6131	2018-10-24 00:00:00+00:00	1054885179859365888	so excited for red dead redemption 2	darrenclark

6132 rows × 4 columns

In [ ]:
```
import twint
import nest_asyncio
nest_asyncio.apply()
```

In [ ]:
```
c = twint.Config()
c.Search = "\"Red Dead Redemption 2\""
c.Since = "2018-10-24"
c.Until = "2018-10-25"
# c.Store_csv = True
# c.Output = "Test.csv"
c.Count = True
twint.run.Search(c)
```

```
1055247487244603392 2018-10-25 01:59:41 +0200 <rfcf> @Gameplanet_Mex buen d
ía, busco el Red Dead Redemption 2 para ps4, el 26 de octubre estará dispon
ible para su compra en tienda? Me queda cerca Plaza Dorada en la ciudad de
Puebla.
1055247465794977795 2018-10-25 01:59:36 +0200 <Archangel491> #RedDeadRedemp
tion2 #RedDeadRedemptionII #RDR2 #Ps4 #XboxOne Red Dead Redemption 2 Ps4/Xb
ox One : Prix de lancement :  https://t.co/AbFuNNQAAV
1055247452918419456 2018-10-25 01:59:32 +0200 <FalsehoodeTV> Catch me drink
ing one of these while I play Red Dead Redemption 2 in a couple of days.  h
ttps://t.co/0U3mpPNaim
1055247363227312128 2018-10-25 01:59:11 +0200 <darksilence669> Going to try
to get all my streaming out today to focus on Red Dead Redemption 2 inspite
of being sick 😅 ill be all over Twitch and Mixer starting up here in a few
😉
1055247361746841600 2018-10-25 01:59:11 +0200 <NILTON_NTN_> Vamos pra mais
um dia de live com meu consagrado @ocarluz , se a tarde teve stardew valey
comigo, por que não adentra a noite/madruga com ele !!  https://t.co/QcW6MY
CWYm  lembrando que vai ter sorteio do red dead redemption 2 na sexta  #str
eaming  #twitch  https://t.co/5dKjRpntWm
1055247328557387776 2018-10-25 01:59:03 +0200 <Shura_stereo> Quiero avisarl
es los privilegios del primer mundo, mañana podré jugar Red Dead Redemption
2 a partir de las 9pm y en físico 😎😎😎😎. Antes que los digital Lovers
Thank you @EBGamesCanada  https://t.co/JRyEWAmwL9
1055247230863634432 2018-10-25 01:58:40 +0200 <DJRobolution> Report: Just 2
More Days And You Can Forget All Of This, Vanish Into 'Red Dead Redemption
2'  https://t.co/2JMe8rBqo1
1055247199129559040 2018-10-25 01:58:32 +0200 <RiseIsLate> I kinda think th
ey made Red Dead Redemption 2 a few years back, and they just remastered it
for the release in a couple days.
1055247183426002944 2018-10-25 01:58:28 +0200 <Kable_10> Do I get Red Dead
Redemption 2 on Friday? (looks awesome but I have never played a Red Dead R
edemption game)
1055247181425319936 2018-10-25 01:58:28 +0200 <M4NyA20> I liked a @YouTube
video  https://t.co/21E88FhNB6 10 Reasons RED DEAD REDEMPTION 2 Is Better T
han GTA 5! (RDR2 vs GTA 5 Comparison)
1055246973295583236 2018-10-25 01:57:38 +0200 <yalnjr> "Red Dead Redemption
2 will let you use your tablet as a map, no pausing required"  https://t.c
o/T2CbYoRV6p  #news #technology #TechTongue #gadgets #Techno
1055246887379521536 2018-10-25 01:57:18 +0200 <garethashe> I liked a @YouTu
be video  https://t.co/jEfM5QO7CK The Problems With Red Dead Redemption 2 Y
ouTubers (RDR2)
1055246852298366978 2018-10-25 01:57:09 +0200 <ceden98> I wish Red Dead Red
emption 2 was coming out for PC, I'm more of a pc gamer and I'm a bit disap
pointed.
1055246840197726208 2018-10-25 01:57:06 +0200 <PjPlayhouse> @AskPlayStation
Hey, I bought Red Dead Redemption 2  Collectors Edition on US PS store  an
d I have 2 accounts on my ps. one EU and one US. in EU its 23 hours to  rel
ease and 1 day and 4 hours on US. Can I use my EU account to play it  earli
er since that is my main account ?
1055246792743440389 2018-10-25 01:56:55 +0200 <Distribucionyma> Red Dead Re
demption 2 tendrá una aplicación complementaria que actuará como mapa inter
activo  https://t.co/379qfhTNSe
```

1055246785168445440 2018-10-25 01:56:53 +0200 <Don_the_dragon2> @TheBubblepup Yeah I absolutely despise everything Bethesda has done with fallout 76. I love the company but with red dead redemption 2 coming out and a ton of games I either want to buy or have but haven't played I'm not going to get it for a long time.
1055246641249353733 2018-10-25 01:56:19 +0200 <congotoday> Red Dead Redemption 2 will let you use your tablet as a map, no pausing required - TechCrunch https://t.co/3uvebDmsux
1055246633460531200 2018-10-25 01:56:17 +0200 <Maliss_Turcon> Suckers will get playing red dead redemption 2 this Friday. I'm on that new shit. https://t.co/CA9QmUldkp
1055246569816145920 2018-10-25 01:56:02 +0200 <urres_> che como que este viernes sale el Red Dead Redemption 2 y yo tengo que subir una banda de materias y rockstar la puta que te pario
1055246552200069121 2018-10-25 01:55:58 +0200 <LWWLloikleloup1> I liked a @YouTube video https://t.co/h7xOhR8aEG RED DEAD REDEMPTION 2 : 10 CHOSES À SAVOIR POUR BIEN DÉBUTER
1055246491118329856 2018-10-25 01:55:43 +0200 <bcolbymartin> Red Dead Redemption 2: Exploring the Open World Gameplay Livestream - IG... https://t.co/O7sDFlCuAR via @YouTube
1055246344074551301 2018-10-25 01:55:08 +0200 <doctormamb0> Started a ff8 100%...and red dead redemption 2 comes out tomorrow...good one me...good one
1055246328639430657 2018-10-25 01:55:04 +0200 <GaymerPriincess> @eguafelipe Força miga eu também tô louca pra comprar o Red dead redemption 2 e não sei se vou ter grana, o PT acabou com as nossas vidas! 😂
1055246309379080192 2018-10-25 01:55:00 +0200 <ScottMimic> Thanks to @TBob53 for coming on the show to explain why a "white out" game is a bad idea, Devin White's half game suspension, how he spent the bye week before Bama when he played, the Eli Apple trade, Pelican power, Queen songs & Red Dead Redemption 2. https://t.co/tbSCRRe4iC
1055246297568030720 2018-10-25 01:54:57 +0200 <SamuelTolbert> If you're looking forward to Red Dead Redemption 2 like I am, I highly recommend listening to this wonderful song by @miracleofsound to help set the mood properly. It's appropriately melancholic and captures the character of John Marston extremely well. https://t.co/rjlPMdNSWf
1055246194002350082 2018-10-25 01:54:32 +0200 <ItsShayton> A ver, voy a mirar en la cama, en el móvil vídeos de Red dead redemption 2. ¿Si me lo pillo queréis verlo en el canal?
1055246190667853824 2018-10-25 01:54:32 +0200 <Scruffy_Man> I have 3 games coming this week. Soul Calibur 6, My Hero One's Justice, and Red Dead Redemption 2. Don't let anyone tell you otherwise, being an adult is awesome because you get to make great financial decisions like me! https://t.co/4JSPBZRr1O
1055246171042668546 2018-10-25 01:54:27 +0200 <dregsofpluto> Quem for fazer stream de Red Dead Redemption 2 tem a obrigação de botar Jack Matador
1055246083461468160 2018-10-25 01:54:06 +0200 <MatgomesCf> @AretaGarcia Seno mês que vem, vc não me der o Red Dead Redemption 2... VOCÊ VAI VER
1055245964724854790 2018-10-25 01:53:38 +0200 <smilum77> Red Dead Redemption 2(2018) https://t.co/cpQURJxdJB
1055245940452417536 2018-10-25 01:53:32 +0200 <Roderick_15> Picking up Red Dead Redemption 2 tomorrow 😈
1055245755626278915 2018-10-25 01:52:48 +0200 <10as3> Gostei de um vídeo @YouTube https://t.co/6hrwb063oa detalhes INÉDITOS em Red Dead Redemption 2
1055245738748329990 2018-10-25 01:52:44 +0200 <RegionPS> Habrá parche de lanzamiento en Red Dead Redemption 2 - https://t.co/xHBZQ90LfP - @RockstarGames @PlayStation @PlayStationEs https://t.co/DiJzy0t8T6
1055245657697595393 2018-10-25 01:52:24 +0200 <jonaskenazi> Red dead redemption 2 map leaked. https://t.co/9xQCQMERGx
1055245638080884741 2018-10-25 01:52:20 +0200 <dirtandroses> My room after I try to make my PS4 run all 100 GB of Red Dead Redemption 2 https://t.co/3Q98cMVywN
1055245626223538178 2018-10-25 01:52:17 +0200 <StraFeyGC> Red dead redemption 2 tomorrow night game of the year is finally here
1055245603322523649 2018-10-25 01:52:12 +0200 <DjGoonieClark> Two more days, fellas. Red Dead Redemption 2. https://t.co/enq7J7z9PG
1055245333893017600 2018-10-25 01:51:07 +0200 <bcolbymartin> Skyrim Special Edition Mods - Red Dead Redemption 2 in Skyrim https://t.co/ayy9UxRBWx via @YouTube
1055245099628675073 2018-10-25 01:50:11 +0200 <PaladinoRPG> Confira tudo qu

e poderá ser feito pelo app de Red Dead Redemption 2 https://t.co/KG7U4ldIxi via @th3deejay
1055245091693047810 2018-10-25 01:50:10 +0200 <thetater112> I can't believe I finally get to play Red Dead Redemption 2 tomorrow. I never thought this day would come
1055245072285937664 2018-10-25 01:50:05 +0200 <tecnologiayvida> Red Dead Redemption 2 tendrá una aplicación complementaria que actuará como mapa interactivo https://t.co/1mjKE4e45D
1055245003361054720 2018-10-25 01:49:48 +0200 <Bowde94> Quand tu te rends compte qu'il reste 2 jours avant Red Dead Redemption 2
1055244999032479746 2018-10-25 01:49:47 +0200 <1933Eagles> Red Dead Redemption 2 - 10+ NEW IMAGES! Gunslingers, Cheat Codes, Gamepl... https://t.co/f1rAWJXu9r via @YouTube
1055244950898495488 2018-10-25 01:49:36 +0200 <deals_fruvbw> Check out what I found on eBay >> #8364 #deal #game Red Dead Redemption 2 Playstation 4 PS4 - Pre-... https://t.co/CmiPbrPIgQ
1055244811429543936 2018-10-25 01:49:03 +0200 <ThatNegroSwag> Bro I'm about to have absolutely no life when red dead redemption 2 comes out.
1055244762767212544 2018-10-25 01:48:51 +0200 <weirdone707> This new fortnite update is freakin awesome, ill never win a match but im okay with that haha red dead redemption 2 is out on friday...i need more time in a day or longer weekends 🤯
1055244742869561344 2018-10-25 01:48:46 +0200 <rafitis33> Gostei de um vídeo @YouTube https://t.co/XYJ4i6kFSP 1ª GAMEPLAY DO RED DEAD REDEMPTION 2! (SEM SPOILERS)
1055244635507961857 2018-10-25 01:48:21 +0200 <KingBean904> Red Dead Redemption Full Story - Before You Play Red Dead Redemption 2 https://t.co/8VYII2q2Kw via @YouTube You should check this out before you play. It's a recap of Red Dead Redemption 1 @WORKBIRDY
1055244633771511808 2018-10-25 01:48:20 +0200 <jlu_2002> TC Gaming news: Red Dead Redemption 2 will let you use your tablet as a map, no pausing required https://t.co/AUQotxEGIM
1055244531480911873 2018-10-25 01:47:56 +0200 <Joelasfuck> @Tesco Can you tell me if Red Dead Redemption 2 is gonna have a midnight release tomorrow night, specifically at the Newton Aycliffe store?
1055244395438661633 2018-10-25 01:47:24 +0200 <Re5urge> This Friday at 1am EST, tune in to the world premiere of Red Dead Redemption 2 exclusively on https://t.co/7GWl8AiZzo follow and tune in!!!! https://t.co/4535zj8Bfb
1055244380020322304 2018-10-25 01:47:20 +0200 <ARealCorndog> @StevenVsMixer Red dead redemption 2 all the way!!! https://t.co/brnqw7rQJP
1055244087996178432 2018-10-25 01:46:10 +0200 <ShinkuAura> Here is "my" fan-made trailer for Red Dead Redemption 2. #RDR2 https://t.co/4MOAXFtdRl I hope you enjoy, was fun editing and doing this. Can't wait to play on the 26th!
1055244073143951361 2018-10-25 01:46:07 +0200 <tha_rami> I think there's currently two types of developers: those who are excited to play Red Dead Redemption 2, and those that are excited about Red Dead Redemption 2 to knock the wind out of the industry and giving them time to do everything else they've been wanting to catch up on.
1055244023743557633 2018-10-25 01:45:55 +0200 <KnusperblubTV> Wir sehen uns dann am Freitag zu Red Dead Redemption 2 wieder bei mir auf Twitch! Danke für den schönen Stream!
1055243914721062912 2018-10-25 01:45:29 +0200 <1rfan_namina> Saya suka video @YouTube https://t.co/HY1e4kPJ7b Red Dead Redemption 2 - GAMEPLAY
1055243889983008768 2018-10-25 01:45:23 +0200 <MooseNCuse> I think the perfect day would be her chilling with me while I play Red Dead Redemption 2. She could get us snacks and Fresca. #Mooseiversary #RedDeadRedemption2 #releasedatefriday
1055243822408429568 2018-10-25 01:45:07 +0200 <hotcelebnews360> #Entertiment #Access #dead Red Dead Redemption 2 release time: PS4 and Xbox early access from Rockstar? https://t.co/EH5sHIKzHs https://t.co/tSt4znD0YH
1055243811251605504 2018-10-25 01:45:04 +0200 <Sickle_Claw> @NECA_TOYS Are you guys planning to do John Marston for Red Dead Redemption 2?
1055243723133468672 2018-10-25 01:44:43 +0200 <SamuelLeahy11> Red dead redemption 2 tomorow ... can't wait
[!] No more data! Scraping will stop now.
found 0 deleted tweets in this search.
[!] Finished! Successfully collected 60 Tweets

In [ ]:

```
c = twint.Config()
c.Search = "\"Red Dead Redemption 2\""
c.Since = "2018-10-24"
c.Until = "2018-10-25"
# c.Store_csv = True
# c.Output = "Test.csv"
c.Count = True
twint.run.Search(c)
```

```
1055247487244603392 2018-10-25 01:59:41 +0200 <rfcf> @Gameplanet_Mex buen d
ía, busco el Red Dead Redemption 2 para ps4, el 26 de octubre estará dispon
ible para su compra en tienda? Me queda cerca Plaza Dorada en la ciudad de
Puebla.
1055247465794977795 2018-10-25 01:59:36 +0200 <Archangel491> #RedDeadRedemp
tion2 #RedDeadRedemptionII #RDR2 #Ps4 #XboxOne Red Dead Redemption 2 Ps4/Xb
ox One : Prix de lancement :  https://t.co/AbFuNNQAAV
1055247452918419456 2018-10-25 01:59:32 +0200 <FalsehoodeTV> Catch me drink
ing one of these while I play Red Dead Redemption 2 in a couple of days.  h
ttps://t.co/0U3mpPNaim
1055247363227312128 2018-10-25 01:59:11 +0200 <darksilence669> Going to try
to get all my streaming out today to focus on Red Dead Redemption 2 inspite
of being sick 😇 ill be all over Twitch and Mixer starting up here in a few
😋
1055247361746841600 2018-10-25 01:59:11 +0200 <NILTON_NTN_> Vamos pra mais
um dia de live com meu consagrado @ocarluz , se a tarde teve stardew valey
comigo, por que não adentra a noite/madruga com ele !!  https://t.co/QcW6MY
CWYm  lembrando que vai ter sorteio do red dead redemption 2 na sexta  #str
eaming  #twitch  https://t.co/5dKjRpntWm
1055247328557387776 2018-10-25 01:59:03 +0200 <Shura_stereo> Quiero avisarl
es los privilegios del primer mundo, mañana podré jugar Red Dead Redemption
2 a partir de las 9pm y en físico 😎😎😎😎. Antes que los digital Lovers
Thank you @EBGamesCanada  https://t.co/JRyEWAmwL9
1055247230863634432 2018-10-25 01:58:40 +0200 <DJRobolution> Report: Just 2
More Days And You Can Forget All Of This, Vanish Into 'Red Dead Redemption
2'  https://t.co/2JMe8rBqo1
1055247199129559040 2018-10-25 01:58:32 +0200 <RiseIsLate> I kinda think th
ey made Red Dead Redemption 2 a few years back, and they just remastered it
for the release in a couple days.
1055247183426002944 2018-10-25 01:58:28 +0200 <Kable_10> Do I get Red Dead
Redemption 2 on Friday? (looks awesome but I have never played a Red Dead R
edemption game)
1055247181425319936 2018-10-25 01:58:28 +0200 <M4NyA20> I liked a @YouTube
video  https://t.co/21E88FhNB6 10 Reasons RED DEAD REDEMPTION 2 Is Better T
han GTA 5! (RDR2 vs GTA 5 Comparison)
1055246973295583236 2018-10-25 01:57:38 +0200 <yalnjr> "Red Dead Redemption
2 will let you use your tablet as a map, no pausing required"  https://t.c
o/T2CbYoRV6p  #news #technology #TechTongue #gadgets #Techno
1055246887379521536 2018-10-25 01:57:18 +0200 <garethashe> I liked a @YouTu
be video  https://t.co/jEfM5QO7CK The Problems With Red Dead Redemption 2 Y
ouTubers (RDR2)
1055246852298366978 2018-10-25 01:57:09 +0200 <ceden98> I wish Red Dead Red
emption 2 was coming out for PC, I'm more of a pc gamer and I'm a bit disap
pointed.
1055246840197726208 2018-10-25 01:57:06 +0200 <PjPlayhouse> @AskPlayStation
Hey, I bought Red Dead Redemption 2  Collectors Edition on US PS store  an
d I have 2 accounts on my ps. one EU and one US. in EU its 23 hours to  rel
ease and 1 day and 4 hours on US. Can I use my EU account to play it  earli
er since that is my main account ?
1055246792743440389 2018-10-25 01:56:55 +0200 <Distribucionyma> Red Dead Re
demption 2 tendrá una aplicación complementaria que actuará como mapa inter
activo  https://t.co/379qfhTNSe
1055246785168445440 2018-10-25 01:56:53 +0200 <Don_the_dragon2> @TheBubblep
up Yeah I absolutely despise everything Bethesda has done with fallout 76.
I love the company but with red dead redemption 2 coming out and a ton of g
ames I either want to buy or have but haven't played I'm not going to get i
t for a long time.
```

1055246641249353733 2018-10-25 01:56:19 +0200 <congotoday> Red Dead Redemption 2 will let you use your tablet as a map, no pausing required - TechCrunch https://t.co/3uvebDmsux
1055246633460531200 2018-10-25 01:56:17 +0200 <Maliss_Turcon> Suckers will get playing red dead redemption 2 this Friday. I'm on that new shit. https://t.co/CA9QmUldkp
1055246569816145920 2018-10-25 01:56:02 +0200 <urres_> che como que este viernes sale el Red Dead Redemption 2 y yo tengo que subir una banda de materias y rockstar la puta que te pario
1055246552200069121 2018-10-25 01:55:58 +0200 <LWWLloikleloup1> I liked a @YouTube video https://t.co/h7xOhR8aEG RED DEAD REDEMPTION 2 : 10 CHOSES À SAVOIR POUR BIEN DÉBUTER
1055246491118329856 2018-10-25 01:55:43 +0200 <bcolbymartin> Red Dead Redemption 2: Exploring the Open World Gameplay Livestream - IG... https://t.co/O7sDFlCuAR via @YouTube
1055246344074551301 2018-10-25 01:55:08 +0200 <doctormamb0> Started a ff8 100%...and red dead redemption 2 comes out tomorrow...good one me...good one
1055246328639430657 2018-10-25 01:55:04 +0200 <GaymerPriincess> @eguafelipe Força miga eu também tô louca pra comprar o Red dead redemption 2 e não sei se vou ter grana, o PT acabou com as nossas vidas! 😂
1055246309379080192 2018-10-25 01:55:00 +0200 <ScottMimic> Thanks to @TBob53 for coming on the show to explain why a "white out" game is a bad idea, Devin White's half game suspension, how he spent the bye week before Bama when he played, the Eli Apple trade, Pelican power, Queen songs & Red Dead Redemption 2. https://t.co/tbSCRRe4iC
1055246297568030720 2018-10-25 01:54:57 +0200 <SamuelTolbert> If you're looking forward to Red Dead Redemption 2 like I am, I highly recommend listening to this wonderful song by @miracleofsound to help set the mood properly. It's appropriately melancholic and captures the character of John Marston extremely well. https://t.co/rjlPMdNSWf
1055246194002350082 2018-10-25 01:54:32 +0200 <ItsShayton> A ver, voy a mirar en la cama, en el móvil vídeos de Red dead redemption 2. ¿Si me lo pillo queréis verlo en el canal?
1055246190667853824 2018-10-25 01:54:32 +0200 <Scruffy_Man> I have 3 games coming this week. Soul Calibur 6, My Hero One's Justice, and Red Dead Redemption 2. Don't let anyone tell you otherwise, being an adult is awesome because you get to make great financial decisions like me! https://t.co/4JSPBZRr1O
1055246171042668546 2018-10-25 01:54:27 +0200 <dregsofpluto> Quem for fazer stream de Red Dead Redemption 2 tem a obrigação de botar Jack Matador
1055246083461468160 2018-10-25 01:54:06 +0200 <MatgomesCf> @AretaGarcia Se no mês que vem, vc não me der o Red Dead Redemption 2... VOCÊ VAI VER
1055245964724854790 2018-10-25 01:53:38 +0200 <smilum77> Red Dead Redemption 2(2018) https://t.co/cpQURJxdJB
1055245940452417536 2018-10-25 01:53:32 +0200 <Roderick_15> Picking up Red Dead Redemption 2 tomorrow 😈
1055245755626278915 2018-10-25 01:52:48 +0200 <10as3> Gostei de um vídeo @YouTube https://t.co/6hrwb063oa detalhes INÉDITOS em Red Dead Redemption 2
1055245738748329990 2018-10-25 01:52:44 +0200 <RegionPS> Habrá parche de lanzamiento en Red Dead Redemption 2 - https://t.co/xHBZQ90LfP - @RockstarGames @PlayStation @PlayStationEs https://t.co/DiJzy0t8T6
1055245657697595393 2018-10-25 01:52:24 +0200 <jonaskenazi> Red dead redemption 2 map leaked. https://t.co/9xQCQMERGx
1055245638080884741 2018-10-25 01:52:20 +0200 <dirtandroses> My room after I try to make my PS4 run all 100 GB of Red Dead Redemption 2 https://t.co/3Q98cMVywN
1055245626223538178 2018-10-25 01:52:17 +0200 <StraFeyGC> Red dead redemption 2 tomorrow night game of the year is finally here
1055245603322523649 2018-10-25 01:52:12 +0200 <DjGoonieClark> Two more days, fellas. Red Dead Redemption 2. https://t.co/enq7J7z9PG
1055245333893017600 2018-10-25 01:51:07 +0200 <bcolbymartin> Skyrim Special Edition Mods - Red Dead Redemption 2 in Skyrim https://t.co/ayy9UxRBWx via @YouTube
1055245099628675073 2018-10-25 01:50:11 +0200 <PaladinoRPG> Confira tudo que poderá ser feito pelo app de Red Dead Redemption 2 https://t.co/KG7U4ldIxi via @th3deejay
1055245091693047810 2018-10-25 01:50:10 +0200 <thetater112> I can’t believe I finally get to play Red Dead Redemption 2 tomorrow. I never thought this day would come

1055245072285937664 2018-10-25 01:50:05 +0200 <tecnologiayvida> Red Dead Redemption 2 tendrá una aplicación complementaria que actuará como mapa interactivo https://t.co/1mjKE4e45D
1055245003361054720 2018-10-25 01:49:48 +0200 <Bowde94> Quand tu te rends compte qu'il reste 2 jours avant Red Dead Redemption 2
1055244999032479746 2018-10-25 01:49:47 +0200 <1933Eagles> Red Dead Redemption 2 - 10+ NEW IMAGES! Gunslingers, Cheat Codes, Gamepl... https://t.co/f1rAWJXu9r via @YouTube
1055244950898495488 2018-10-25 01:49:36 +0200 <deals_fruvbw> Check out what I found on eBay >> #8364 #deal #game Red Dead Redemption 2 Playstation 4 PS4 - Pre-... https://t.co/CmiPbrPIgQ
1055244811429543936 2018-10-25 01:49:03 +0200 <ThatNegroSwag> Bro I'm about to have absolutely no life when red dead redemption 2 comes out.
1055244762767212544 2018-10-25 01:48:51 +0200 <weirdone707> This new fortnite update is freakin awesome, ill never win a match but im okay with that haha red dead redemption 2 is out on friday...i need more time in a day or longer weekends 😨
1055244742869561344 2018-10-25 01:48:46 +0200 <rafitis33> Gostei de um vídeo @YouTube https://t.co/XYJ4i6kFSP 1ª GAMEPLAY DO RED DEAD REDEMPTION 2! (SEM SPOILERS)
1055244635507961857 2018-10-25 01:48:21 +0200 <KingBean904> Red Dead Redemption Full Story - Before You Play Red Dead Redemption 2 https://t.co/8VYII2q2Kw via @YouTube You should check this out before you play. It's a recap of Red Dead Redemption 1 @WORKBIRDY
1055244633771511808 2018-10-25 01:48:20 +0200 <jlu_2002> TC Gaming news: Red Dead Redemption 2 will let you use your tablet as a map, no pausing required https://t.co/AUQotxEGIM
1055244531480911873 2018-10-25 01:47:56 +0200 <Joelasfuck> @Tesco Can you tell me if Red Dead Redemption 2 is gonna have a midnight release tomorrow night, specifically at the Newton Aycliffe store?
1055244395438661633 2018-10-25 01:47:24 +0200 <Re5urge> This Friday at 1am EST, tune in to the world premiere of Red Dead Redemption 2 exclusively on https://t.co/7GWl8AiZzo follow and tune in!!!! https://t.co/4535zj8Bfb
1055244380020322304 2018-10-25 01:47:20 +0200 <ARealCorndog> @StevenVsMixer Red dead redemption 2 all the way!!! https://t.co/brnqw7rQJP
1055244087996178432 2018-10-25 01:46:10 +0200 <ShinkuAura> Here is "my" fan-made trailer for Red Dead Redemption 2. #RDR2 https://t.co/4MOAXFtdRl I hope you enjoy, was fun editing and doing this. Can't wait to play on the 26th!
1055244073143951361 2018-10-25 01:46:07 +0200 <tha_rami> I think there's currently two types of developers: those who are excited to play Red Dead Redemption 2, and those that are excited about Red Dead Redemption 2 to knock the wind out of the industry and giving them time to do everything else they've been wanting to catch up on.
1055244023743557633 2018-10-25 01:45:55 +0200 <KnusperblubTV> Wir sehen uns dann am Freitag zu Red Dead Redemption 2 wieder bei mir auf Twitch! Danke für den schönen Stream!
1055243914721062912 2018-10-25 01:45:29 +0200 <1rfan_namina> Saya suka video @YouTube https://t.co/HY1e4kPJ7b Red Dead Redemption 2 - GAMEPLAY
1055243889983008768 2018-10-25 01:45:23 +0200 <MooseNCuse> I think the perfect day would be her chilling with me while I play Red Dead Redemption 2. She could get us snacks and Fresca. #Mooseiversary #RedDeadRedemption2 #releasedatefriday
1055243822408429568 2018-10-25 01:45:07 +0200 <hotcelebnews360> #Entertiment #Access #dead Red Dead Redemption 2 release time: PS4 and Xbox early access from Rockstar? https://t.co/EH5sHIKzHs https://t.co/tSt4znD0YH
1055243811251605504 2018-10-25 01:45:04 +0200 <Sickle_Claw> @NECA_TOYS Are you guys planning to do John Marston for Red Dead Redemption 2?
1055243723133468672 2018-10-25 01:44:43 +0200 <SamuelLeahy11> Red dead redemption 2 tomorow ... can't wait
1055243642233864193 2018-10-25 01:44:24 +0200 <J3R3MY87> I liked a @YouTube video https://t.co/7cECQnoRIA Red Dead Redemption 2 - 10+ NEW IMAGES! Gunslingers, Cheat Codes, Gameplay Info &
1055243488101650433 2018-10-25 01:43:47 +0200 <mrkx1221> Me when I see that the download for Red Dead Redemption 2 is almost 100 gb https://t.co/CRQUTTnGlJ
1055243349261729793 2018-10-25 01:43:14 +0200 <atribunasantos> Santos terá evento de lançamento de 'Red Dead Redemption 2' nesta quinta-feira https://t.co/jbAPpMUOs6

1055243309269090305 2018-10-25 01:43:05 +0200 <matt_wade22> Me when Red Dea
d Redemption 2 comes out this weekend. https://t.co/8p9TcNMXpC
1055243279254450176 2018-10-25 01:42:57 +0200 <CMQNetwork> Two days till Re
d Dead Redemption 2. Who’s excited for it?
1055243144881692672 2018-10-25 01:42:25 +0200 <N_Nickros_C> J'aime une vidé
o @YouTube : "RED DEAD REDEMPTION 2 THE SNOW GAMEPLAY" à l'adresse http
s://t.co/jc0peDggvz.
1055242964719538176 2018-10-25 01:41:42 +0200 <RavenLothina> 2 dias para sa
ir Red Dead Redemption 2, e 4 dias para as eleições. Outubro está sendo o m
ês mais louco dos últimos 4 anos.
1055242930791858177 2018-10-25 01:41:34 +0200 <DrNovik_XBL> I liked a @YouT
ube video https://t.co/xrKrffUvEd Unboxing Red Dead Redemption 2 EARLY for
Nintendo
1055242920213864448 2018-10-25 01:41:32 +0200 <OhSuty> yo lo que quiero es
red dead redemption 2
1055242837783130112 2018-10-25 01:41:12 +0200 <DarkCigars> Red Dead Redempt
ion 2: verrà offerto del cibo ai giocatori in fila presso i negozi GAME Ho
detto tutto #RDR2 https://t.co/Xz33Rd1lgo
1055242726214696960 2018-10-25 01:40:46 +0200 <JordanNunes123> Red Dead Red
emption 2 pre ordered and the download has already started. Only thing left
is to not die between now and Friday
1055242688809889792 2018-10-25 01:40:37 +0200 <mwfarr> Me: I should really
save money if I'm trying to buy a van, new guitar, Red Dead Redemption 2, e
tc. My stomach: I'm hungry My brain: Order some Chinese food! Me: Good i
dea! *orders 36 dollars worth of Chinese food https://t.co/Y6MH2izHkh
1055242651493130240 2018-10-25 01:40:28 +0200 <eyezaeyuhh> Red dead redempt
ion 2 Tomorrow https://t.co/KPUUkobsA2
1055242436623122434 2018-10-25 01:39:37 +0200 <BIGGIEsnacks> Check out [!ra
ffle !freeshit !autobot] It's a DEER.. BRAH!! 1 DAYS UNTIL RED DEAD REDEMPT
ION 2 !!!!!! https://t.co/v1vZYyWOP6
1055242362774056960 2018-10-25 01:39:19 +0200 <MParras_ps3> Ahora mismo Yak
uza Kiwami 2 es el juego que mas me apetece jugar de todos los existentes,
y lo digo a falta de 23 horas para que salga Red Dead Redemption 2, el cual
compré esta mañana.
1055242232662499328 2018-10-25 01:38:48 +0200 <LeonardHoover62> Would it be
wrong to quit my job, drop out of school, and break up with my girlfriend j
ust for 3 days? so I can enjoy red dead redemption 2 with no real life int
erruptions.
1055242169953501184 2018-10-25 01:38:33 +0200 <Ricardozam> While you play R
ed Dead Redemption 2, make time to read this book https://t.co/llu8ZT76D7
1055242121274425345 2018-10-25 01:38:21 +0200 <InvaderBrain88> Drunken Red
Dead Redemption 2 on Friday.
1055242031293964293 2018-10-25 01:38:00 +0200 <CaraotaDigital> Cantante ven
ezolano Arca será parte del videojuego Red Dead Redemption 2 / Vía: @Caraot
aShowVE https://t.co/dmuQRgmZ9z
1055241965380493312 2018-10-25 01:37:44 +0200 <marciojmsilva> Red Dead Rede
mption 2: Trailers, release date, gameplay, plot details, bonus content and
more https://t.co/io7druHJIp
1055241958204100609 2018-10-25 01:37:42 +0200 <RNCRADIOLIVE> Missed the deb
ut episode of #RAOPGaming? Tune in as @ampaveli & @LightSkinNasty talk
WWE 2k19, Red Dead Redemption 2, the Playstation classic + more. Soundclou
d: https://t.co/e4elbAETzK iTunes: https://t.co/3Rbb002tA0 Spotify: http
s://t.co/KQe3qYlrYo
1055241918781677568 2018-10-25 01:37:33 +0200 <TheSeshEmpire> Red Dead Rede
mption 2 is launching with a companion app for iOS and Android that offers
"real-time" information from your game like health and ammo,so you can play
with no HUD in the game. Also check the map, place waypoints, read journals
and so on. Sweet https://t.co/35QfRcORFw
1055241898783322112 2018-10-25 01:37:28 +0200 <marciojmsilva> Red Dead Rede
mption 2 app lets you explore game's world in Android, iOS https://t.co/qu
roJuiTnE
1055241700845805568 2018-10-25 01:36:41 +0200 <NicSimard1> @savardf Nous vo
us écrivons à propos de votre commande pour l'article suivant : Red Dead Re
demption 2 - Xbox One Standard Edition. Nous avons récemment appris qu’il n
e nous sera pas possible de respecter la promesse de livraison que nous vou
s avons faite, et nous tenions à nous en excuse
1055241689554714626 2018-10-25 01:36:38 +0200 <Roxzothina> 3. Spyro Reignit
ed Trylogy, KH3, Red Dead Redemption 2, Let's Go Eevee, & Call of Cathu
lhu.
1055241632721879040 2018-10-25 01:36:25 +0200 <AltMagOnline> @linton85 I wi

ll probably play the new TR next (at last!) then eventually replay TR2 (my fav of the classic TRs). I don’t stream unfortunately (I’m playing via back wards compatability on my PS3) but I’m considering writing a retrospective! However, Red Dead Redemption 2 is nearly here...
1055241507698028545 2018-10-25 01:35:55 +0200 <mimitchi> im a sad fucking c owboy that im not going to be able to buy red dead redemption 2 and am gonn a just have to watch youtube videos or something
1055241371404193792 2018-10-25 01:35:23 +0200 <andre_andre_007> Red Dead Re demption 2 will let you use your tablet as a map, no pausing required – Tec hCrunch https://t.co/eL0B6nbYRK
1055241343348420609 2018-10-25 01:35:16 +0200 <tristanshane> @Chris_Smoove you plan on playing red dead redemption 2?
1055241238922887168 2018-10-25 01:34:51 +0200 <JezCorden> Red Dead Redempti on 2 is so close.
1055241007669878784 2018-10-25 01:33:56 +0200 <ComicBookAlex> Me when watch ing Marvel's Spider-Man trailer: I can't wait to swing around New York Cit y, fight crime and take down the bad guys. Me when watching Red Dead Redem ption 2 trailer: I can't wait to rob banks, get in gun fights and be an out law in the wild west.
1055240895451316224 2018-10-25 01:33:29 +0200 <cesar_moretto> já to assim e sperando lançar red dead redemption 2 #RDR2 https://t.co/6peLDeABon
1055240871493451777 2018-10-25 01:33:23 +0200 <Gamer_Canada> The Countdown To Red Dead Redemption 2, Canadian Gamer https://t.co/x6VXMtrQqm via @YouT ube
1055240842812801024 2018-10-25 01:33:17 +0200 <peterfss1> passar o aniversá rio jogando red dead redemption 2
1055240795270180864 2018-10-25 01:33:05 +0200 <gamedaim> Rockstar Games Pam erkan Konten Awal Untuk Red Dead Redemption 2! https://t.co/T6YirHBiP7 ht tps://t.co/wQoKpYZwg6
1055240728660586497 2018-10-25 01:32:49 +0200 <AsianFolder_> 🔍Red Dead Red emption 2 changed the name of a city because of Mafia 3 Red Dead Redemptio n 2, the new video game from the parents of Grand Theft Auto V, is just aro und the corner. #game #games #RedDeadRdemption2 #RedDead 🔎 https://t.co/b BQ1fUPqg4
1055240606149013504 2018-10-25 01:32:20 +0200 <GoodFace00> To sleep like a normal person for once or stay up for the midnight release of red dead rede mption 2 🤔
1055240590378631169 2018-10-25 01:32:16 +0200 <doozerblake> Looks like @Nat Geo is in on that Red Dead Redemption 2 hype. https://t.co/WXz2TbLvew
1055240556262109185 2018-10-25 01:32:08 +0200 <AdamsFlaFan> Report: Just 2 More Days And You Can Forget All Of This, Vanish Into ‘Red Dead Redemption 2’ https://t.co/xtSaVhOr9h
1055240554903146497 2018-10-25 01:32:08 +0200 <FelipeKent2> No tengo ningún jueguito y en lo que llega red dead redemption 2 Estoy mal :'(https://t.c o/5p4z75Hbzs
[!] No more data! Scraping will stop now.