Video Game Data Sales Analysis With Python

A FINAL YEAR PROJECT REPORT SUBMITTED FOR PARTIAL FULFILLMENT OF BACHELOR OF TECHNOLOGY COMPUTER & SCIENCE ENGINEERING

Dr. Shakuntala Misra National Rehabilitation University.
DECEMBER 2021

SUBMITTED TO-
Ms.Abha Sweta
Assistant Professor
Department of Computer Science & Engineering
Dr. Shakuntala Misra National Rehabilitation University

SUBMITTED BY-
Pulkit Singh
Roll.No.-188330041
B.Tech
Computer Science & Engineering
4th Year(7th Semester)

Department of Computer Science & Engineering
Dr. Shakuntala Misra National Rehabilitation University.

Declaration

I Pulkit Singh hereby declare that the work which is being presented in the B. Tech. The project entitled "**Video Games Data Sales Analysis with Python** " in partial fulfillment of the requirements for the award of the degree of Bachelor of Technology (Computer Science and Engineering) submitted in the Department of Computer Science and Engineering, Institute of Engineering & Technology is an authentic record of our work carried out during a period from 2020 to 2021 under the supervision of Ms. Abha Sweta, Department of Computer Science and Engineering, Institute of Engineering & Technology, Dr. Shakuntala Misra National Rehabilitation University, Lucknow. The work presented in the present report is original and has not been submitted elsewhere by us for the award of any other degree/diploma at this or any other Institution/ University.

Pulkit Singh 188330041

Date: 28th December 2021

Declaration

We hereby affirm that no part of my/ our work can be considered plagiarism and take full responsibility if such complaints occur. Plagiarism is attributed to any one or combination of the following instances:

(1) Uncredited verbatim copying of individual sentences, paragraphs, illustrations such as diagrams, graphs, etc. from any source (published or not), inclusive of the internet.
(2) Uncredited improper paraphrasing of paragraphs or pages.
(3) Credited verbatim copying of a major section of a published paper or chapter without clarifying the person who wrote it or worked on it.
(4) Long phrases or sentences that had to be used verbatim from literature must be specified using quotation marks. We completely understand that the supervisor may not be in a position to possibly check such incidences of plagiarism in this project report.

Pulkit Singh 188330041

Date: 28th December 2021

Certificate

This is to certify that the Project report entitled "**Video Games Data Sales Analysis with Python**" is a bonafide work of Pulkit Singh B.Tech. (Computer Science and Engineering) of the Institute of Engineering & Technology, Dr. Shakuntala Misra National Rehabilitation University, Lucknow, Uttar Pradesh in the academic session 2020-21, in partial fulfillment of the requirements for the award of the degree in B. Tech. (Computer Science and Engineering). The work is comprehensive, complete, and suitable for final evaluation.

Date:28th December 2021

Signature
Ms. Abha Sweta
Assistant Professor
Department of Computer Science
and Engineering
Institute of Engineering & Technology
Dr. Shakuntala Misra
National Rehabilitation University
Lucknow

Acknowledgments

I wish to record my deep sense of gratitude and profound thanks to my Project report supervisor Ms.Abha Sweta, Department of Computer Science and Engineering, Institute of Engineering & Technology, Dr. Shakuntala Misra National Rehabilitation University for her keen interest, inspiring guidance, constant encouragement with my work during all stages, to bring this work into fruition. I would like to extend my sincere thanks to Prof. (Dr.) Ashok Kumar Mishra, Incharge, Institute of Engineering and Technology, Dr. Shakuntala Misra National Rehabilitation University, and other faculty members for providing excellent platform and resources to carry out my project work. I would like to thank the panel members for their valuable suggestions and support during the presentation of my work. I thank my friends and family members who have encouraged me and provided valuable support throughout my Project work.

Pulkit Singh
188330041

Abstract

As each and every sector of the market is growing, data is building up day by day, we need to keep a record of the data which can be helpful for analytics and evaluation. Now we don't have data in gigabyte or terabyte but in zettabyte and petabyte and this data can not be handled with day-to-day software such as Excel or Matlab. Therefore in this report, we will be dealing with large data sets with the high-level programming language 'Python'.

The main objective of this project is to **aggregate** and **analyze** the data collected from the different data sources available on the internet. This project mainly focuses on the usage of the python programming language in the field of Sales Analysis. This language has not only its application in the field of just analyzing the data but also for the prediction of the upcoming scenarios in the Video Games Sales.

The purpose of using this specific language is due to its versatility, vast libraries (Pandas, Numpy, Matplotlib, etc.), speed limitations, and ease of understanding. We will be analyzing large Video Games data sets in this project which can not be easily analyzed in other tools as compared to python. Python does not have its limitation to only data analytics but also in many other fields such as Data Science, Machine learning, and many more.

Contents

Section 1

Data Science Introduction

1.1 Data Science

Data science is the field of data analytics and data visualization in which raw data or unstructured data is cleaned and made ready for analysis purposes. Data scientists use this data to get the required information for future purposes. ”Data science uses many processes and methods on the big data, the data may be structured or unstructured”. Data frames available on the internet are the raw data we get. It may be either in an unstructured or semi-structured format. This data is further filtered, cleaned and then the number of the required task is performed for the analysis with the use of the high programming language. This data is further analyzed and then presented for our better understanding and evaluation.

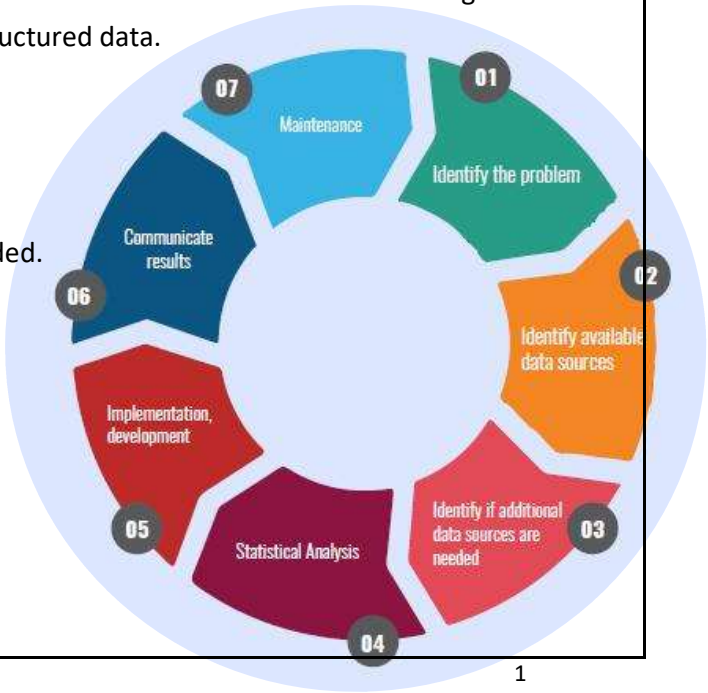
One must be clear that data science is not about making complicated models or making awe-some visualization nor it is about writing code but about using the data to create an impact for your company, for this impact we need tools like complicated data models and data visualization.

1.2 Stages of Data Science

There are many tools used to handle the big data available to us.” Data scientists use programming tools such as Python, R, SAS, Java, Perl, and C/C++ to extract knowledge from prepared data”.

Data scientists use many algorithms and mathematical models on the data. Following are the stages and their cycle performed on the unstructured data.

- Identifying the problem
- Identify available data sources
- Identify available data sources
- Identify if additional data sources are needed.
- Statistical analysis
- Implementation, development
- Communicate results
- Maintenance



Data science finds its application in many fields. With the assistance of data science, it is easy to get the search query on search engines in plenty of time. The role of the data scientist is to have a deep understanding of the data as well as a good command of the programming language, he should also know how to work with the raw data extracted from the data source. Many programming languages are used to analyze and evaluate the data such as Python, Java, MATLAB, Scala, Julia, R., SQL, and TensorFlow. Among which python is the most user-friendly and vastly used programming language in the field of data science.
This life cycle is applied in every field, in this project we will be considering all these seven stages of data science to analyze the data. The process will be starting from data collection, data preparation, data modeling, and finally data evaluation. For instance, As we have huge

1.3 Concept Data Science in Video Games Sales Analysis

This project aims to predict the top-selling video game sales in North America between 1983 and 2016. The dataset is collected from an internet platform known as Kaggle.com. The dataset was generated by vgchartz.com. The exploitation of the dataset, the VSCode IDE tool, and Python-programming language are used for data cleaning, analysis, and representation. The machine learning algorithm used in this project is linear regression. Based on the Video Games Sales knowledge, it would be fascinating to know what area unit the required factors that make a game further successfully sold-out than others in North America. So, we would like to research what quite a video games that area unit further successfully sold-out in North America. We tend to jointly would like to point the results of this Analysis in Associate in nursing intuitive methodology by visualizing outcome victimization ggplot2 in python. In this project, we tend to require NA_Sales (North America sales) as the response variable and specialize in operative predictions by analyzing the rest of the variables inside the video games sales data. The results can facilitate film companies to know the key to generating an advertisement success game amount of data we can create an energy model for a particular country by collecting its previous energy data, we can also predict the future requirement of it with the same data.

Section 2

INTRODUCTION TO PROJECT

In this modern world, video games are contended by all age teams of an individual. These video games are contended on a video screen (on television, computer). There are many video games based on platform and genre like WII, NES, GB, etc., and puzzle, racing, shooting, etc., respectively. Video games are released by a publisher like Activision, Nintendo, etc., across different platforms. Video games have become a major platform of entertainment especially for kids and are used for business purposes as well. The video games such as NFS, Call of Duty, Battlefield, and many more have produced exceptional income for about two to three decades and still earning.

Predictive modeling has helped loads of people and organizations. This data science has several techniques, simulations, and machine learning. The main purpose of this study is to seek out the issue that ends up in high video selling games Sales in North America. The dataset has 16,598 data, and 11 attributes. This dataset is created based on the data from www.vgchartz.com/gamedb/. Then VSCode is used to predict which factor leads to top video game sales in North America.

2.1 METHODOLOGY

- **Machine Learning Algorithm** The ML algorithm is a logic that grasps one step ahead when exposed to more information/data. When ML is exposed to training data it produces a model. To build a Model, the Machine Learning Algorithm is used here Linear Regression (Supervised Learning). It predicts the output values based on the input data fed. This algorithm builds a model based on the training data produced and predicts the new data.
- **Dataset** The VSCode is used to import the dataset. Dataset can be in excel or CSV format. The dataset is reviewed and normalized. Normalization is changing the value of numeric columns of the dataset to common values and fit into a specific range. The attributes of the dataset are:

1 Rank — Ranking of overall sales

2-Name — The games name

3-Platform — Platform of the game's release (i.e. PC, PS4, etc.)

4-Year — Year of the game's release

5-Genre — Genre of the game

6-Publisher — Publisher of the game

7-NA_Sales — Sales in North America (in millions)

8-EU_Sales — Sales in Europe (in millions)

9-JP_Sales — Sales in Japan (in millions)

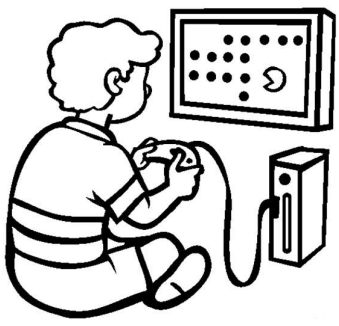
10-Other_Sales — Sales in the rest of the world (in millions)

11-Global_Sales — Total worldwide sales

2.2 EXPERIMENT: This project/study uses a special video game sale dataset sold in different countries. This dataset is created by VGChartz.com.The VSCode is used to run an experiment. The VSCode is free, open-source and an Integrated Development Environment (IDE) for Python-language, used for statistical computing, programming, GUI, and Graphics. VSCode is most famous for Graphical capabilities, but in recent times it gained importance for analyzing data.

A . Data cleaning At this stage, by using VSCode we import datasets and remove redundant, missing, duplicate, and unnecessary data for further processing. This stage is the most time-consuming in Data Science because it prevents wrongful prediction and gets rid of the inconsistencies of data.

B . Data Exploration and Analysis In this stage, we detect patterns, trends, and behavior in the data or dataset. This process makes further analysis easier because it excludes irrelevant data points and searches for no results data. It uses visualization which makes it easy to analyze. From our analysis, we concluded Platform attribute has mainly affected the Video Game Sales in North America



Section 3
Python Programming language basics

3.1Why only Python?

”Python is an interpreted, object-oriented, high-level programming language with dynamic semantics”. This language consists of main data structures which make it very easy for the data scientists to analyze the data very effectively. It does not only help in forecasting and analysis it also helps in connecting the two different languages. Two best features of this programming language are that it does not have any compilation step as compared to the other programming language in which compilation is done before the program is being executed and the other one is the reuse of the code, it consists of modules and packages due to which we can use the previously written code anywhere in between the program whenever is required.

There are multiple languages, for example, R., Java, SQL, Julia, Scala, MATLAB available in the market which can be used to analyze and evaluate the data, but due to some outstanding features python is the most famous language used in the field of data science.

Python is mostly used and easy among all other programming languages is due to the following reasons.

- Easy to Learn and Use.
- Mature and Supportive Python Community.
- Support from Renowned Corporate Sponsors.
- Hundreds of Python Libraries and Frameworks.
- Versatility, Efficiency, Reliability, and Speed.
- Big data, Machine Learning and Cloud Computing.
- First-choice Language.

3.2Data structures in Python

Data structures are the way of storing the data so that we can easily perform different operations on the data whenever required. When the data has been collected from the data source the data is available in different forms. So later it is easy for the data scientists to perform a different operation on the data once it is sorted into different data structures.
Data structures are mainly classified into two categories and then further their subcategories are shown below.

1. Primitive Data Structures.
They are also called basic data structures. This type of data structure contains simple values of the data.

Integers - All the whole numbers from negative infinity to positive infinity come under integer data types. for example 4,9,-2,-6.

Float - The decimal figure numbers or rational numbers come under float data types. for example 3.1,2.2,8.96

Strings - The collection of alphabets or characters are called strings. We enclose the string either in single or double quotes in python. for example 'hello' and "bread".

Boolean- These are the built-in data types that take two values that are 'True' and 'False'. Truly represents the 1 and False represents 0 in python.

2. Non-Primitive Data Structures
These are the derived type or reference variables data structures. They are called derived data structures because they are derived from the basic data structures such as integer and float. Python has mainly five types of data structures.
Following are the non-primitive data structures.

List - ”A list is a value that contains multiple values in an ordered sequence”. Values in the list refer to the list itself, that is the value that can be stored in a variable or passed to a function. List are changeable and values in the list are enclosed inside a square bracket, we can perform multiple operations such as indexing, slicing, adding, and multiplying.

```
In [48]: cols = ['Name','Platform','Year','Publisher']
         games.loc[games.Name=='FIFA 14', cols].sort_values(['Year'])
```

Out[48]:

	Name	Platform	Year	Publisher
112	FIFA 14	PS3	2013.0	Electronic Arts
256	FIFA 14	X360	2013.0	Electronic Arts
493	FIFA 14	PS4	2013.0	Electronic Arts
1698	FIFA 14	XOne	2013.0	Electronic Arts
3988	FIFA 14	PSV	2013.0	Electronic Arts
4503	FIFA 14	PC	2013.0	Electronic Arts
4538	FIFA 14	Wii	2013.0	Electronic Arts
6661	FIFA 14	3DS	2013.0	Electronic Arts
7034	FIFA 14	PSP	2013.0	Electronic Arts

Figure 2.2: list example

Tuple - A tuple is a list of non changeable objects. The differences between tuples and lists are that the tuples cannot be changed, tuples use parentheses, whereas the list uses square brackets.

TUPEL_AND _SETS > tuple_function.py > ...

```
1  mytuple = 12,12351,65,15,11,81,5,41,1,2,33,2,5,54,52,18,41,52,541,651,541,62,3521
2
3  print(mytuple.count(1),'occurance of 1')
4  print(mytuple.count(65),'index has 65')
5
6  for i in mytuple:
7      print(i,end = ' ')
8  print('\nreversed')
9
10 for i in  mytuple[::-1]:
11     print(i,end=' ')
```

Figure 2.3: tuple example

Dictionary- These are nothing but a type of data structure that consists of key-value pairs enclosed in the curly brackets. It is same as the any dictionary we use in day-to-day life in which we find the meaning of the particular words. So if I compare a normal dictionary to this python dictionary data structure then the word in a dictionary will be our key and its meaning will be the value of the dictionary. In the figure name, occupation and Student are the keys, and student, data analyst, and name, maths, etc, are the values assigned to the keys.

CTIONARY > Creating_dictionary.py > ...

```
1   student = ['ramu,17,15,12,14,False,True,9621852114']
2
3   stddict={
4       'name':'bille',
5       'maths': 17,
6       'english':15,
7       'hindi':12,
8       'age':14,
9       'studies':False,
10      'delinquent':True,
11      'phone':9621852114
12  }
13  print(stddict)
14
15  myinfo = {
16      'city':'lucknow',
17      'state':'uttarpradesh',
18      'ph.no':68415616516,
19      'ph.no':845145184545,    #if we wirte to same comtent to first one will remove and second will be added
```

Figure 2.4: dictionary example

Sets - Sets are used for calculating mathematical operations such as union, intersection, and symmetric difference.

adding_sets.py ×

TUPEL_AND _SETS > adding_sets.py > ...

```
1   a = {1,2,3}
2
3   a.add(129)
4   a.add('Charbagh')
5   a.add((1,2,3))
6   print(a)
7
8   # error -> u cannot add a list , dict or set in set
9   #a.add([1,2,3])
10  #print(a)
11
12  x = [1,2,3,2,1,1,521,3,1,3,1]
13  a.update(x)
14  print(a)
```

Below is the data structure tree which explains the category and sub-category of each data type.

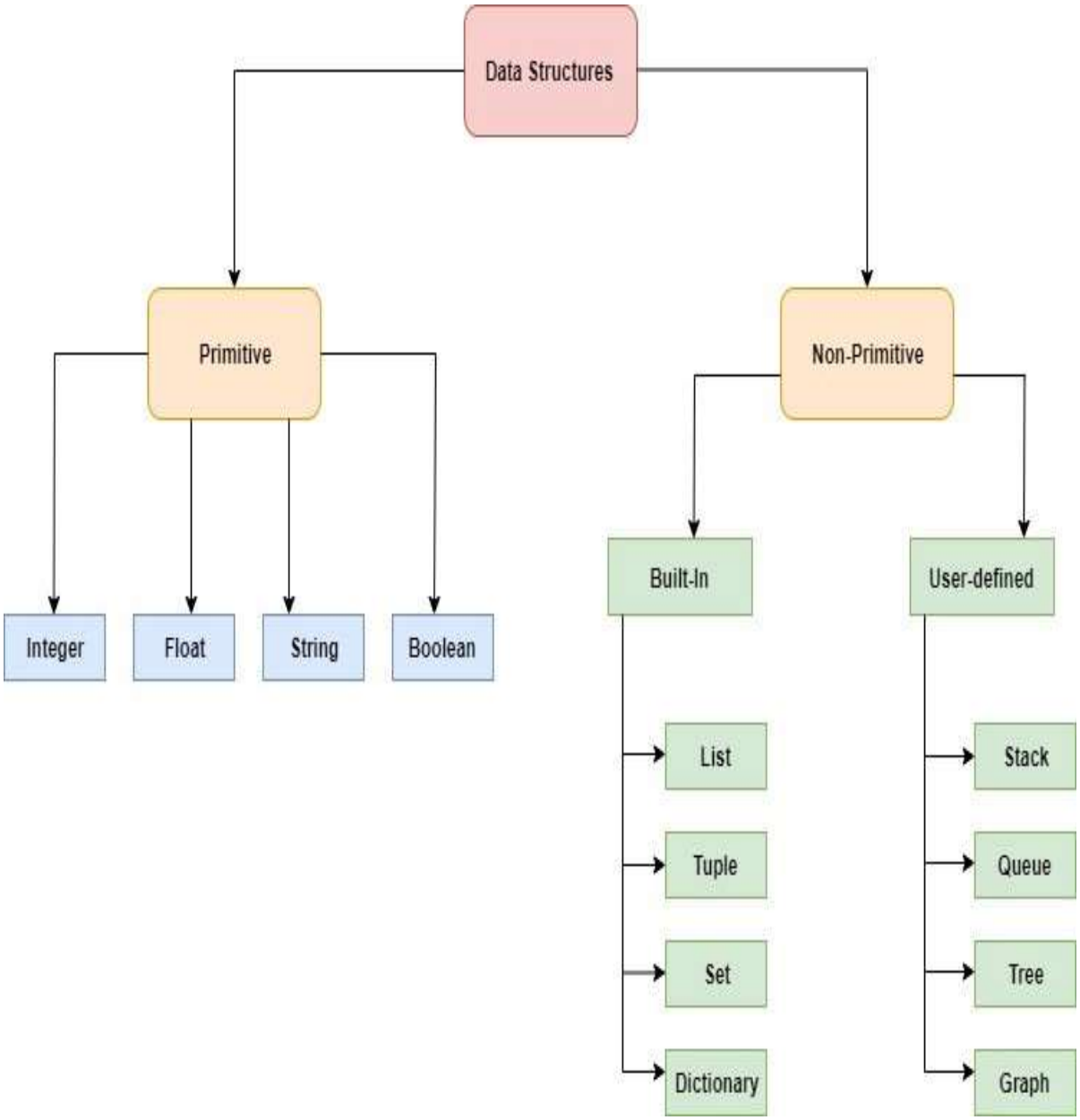


Figure 2.5: A data structure tree at a glance

3.3 Condition statements

3.3.1 If else statements

”The most common type of statement is the if statement. if statement consist of a block which is called as clause”, it is the block after if statement, it executed the statement if the condition is true. The statement is omitted if the condition is False. then the statement in the else part is printed

If statement consist of following -

If keyword itself

Condition which may be True or False

Colon

If clause or a block of code

Below is the figure shows how If and else statements are used with description inside it.

```
In [4]: x = 32


        if x<40:
            print('x is less than 40')

        else:
            print('x is greater than 40')

        #In the above program we have if and else statements both
        #first line we have assigned value 32 to a variable x
        #second line contain keyword with condition x<40
        #If the condtion is true then the statement is printed
        #else the other condtion is executed in else statement


        x is less than 40
```

Figure 2.6: if else statement

3.3.2 elif statements

In this statement only one statement is executed, There are many cases in which there is only one possibility to execute. ”The elif statement is an else if statement that always follows an if or another elif statement”[8]. The elif statement provides another condition that is checked only if any of the previous conditions were False. In code, an elif statement always consists of the following:. The only difference between if else and elif statement is that in elif statement we have the condition where as in else statement we do not have any condition.

elIf statement consist of following -

elIf keyword itself

Condition which may be True or False

Colon

elIf clause or a block of code

Below is the figure shows how elIf statement is used with description inside

Figure 2.7: elif example

3.4 Module, Package and Functions

Module

Modules are Python files which has extension as .py. The name of the module will be the name of the file. A Python module can have a set of functions, classes or variables defined and implemented.

Module has some python codes, this codes can define the classes, functions and variables.The reason behind using the module is that it organizes your python code by group-ing the python code so that it is easier to use.

Package

A package consist of the collection of modules in which python codes are written with name init.py. It means that each python code inside of the python path, which contains a file named init.py, will be treated as a package by Python. Packages are used for organiz-ing the module by using dotted names.

for example -

We have a package named simple package which consist of two modules a and b. We will import the module from package in following way.

from simple package import a, b

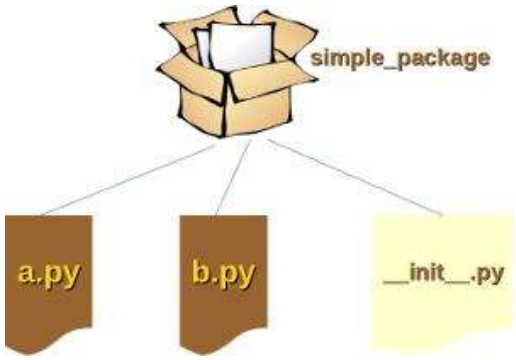


Figure 2.10: packages example

3.5 Functions

A function is a python code which can be reused at any anytime in the whole python code. Function performs specific task whenever it is called during the program.With the help of function the program is divided in to multiple codes.

Built in functions - The functions which are already in the python programming and have specific action to perform are called as built in functions. This function are immutable. Some examples of this functions are -

chr() - used to get string

print() - used to print an object in terminal min() -

used to get minimum value in terminal

User defined functions - This functions are user to defined functions and it starts with the key word 'def' as shown in the example below. We have defined the function names groupingdata & dark_theam and its task to be performed when called. Below is the example of it.

```
23  def groupingdata(data):
24      x = data.groupby("Year")
25
26      return x
27
28  def dark_theme():
29      st.markdown("<style>h1{color : white;}</style>" , unsafe_allow_html = True)
30      st.markdown("<style>h2{color : white;}</style>" , unsafe_allow_html = True)
31      st.markdown("<style>h3{color : white;}</style>" , unsafe_allow_html = True)
32      st.markdown("<style>h4{color : white;}</style>" , unsafe_allow_html = True)
33      st.markdown("<style>h5{color : white;}</style>" , unsafe_allow_html = True)
34      st.markdown("<style>h6{color : white;}</style>" , unsafe_allow_html = True)
35
36
37  #function calling
38
39  data , data_year = load_data()
```

Figure 2.11: function example

Section 4

Libraries in Python

Python library is vast. There are built-in functions in the library that are written in C language. This library provides access to system functionality such as file input-output and that is not accessible to Python programmers. These modules and libraries provide the solution to the many problems in programming.

Following are some Python libraries.-

Pandas

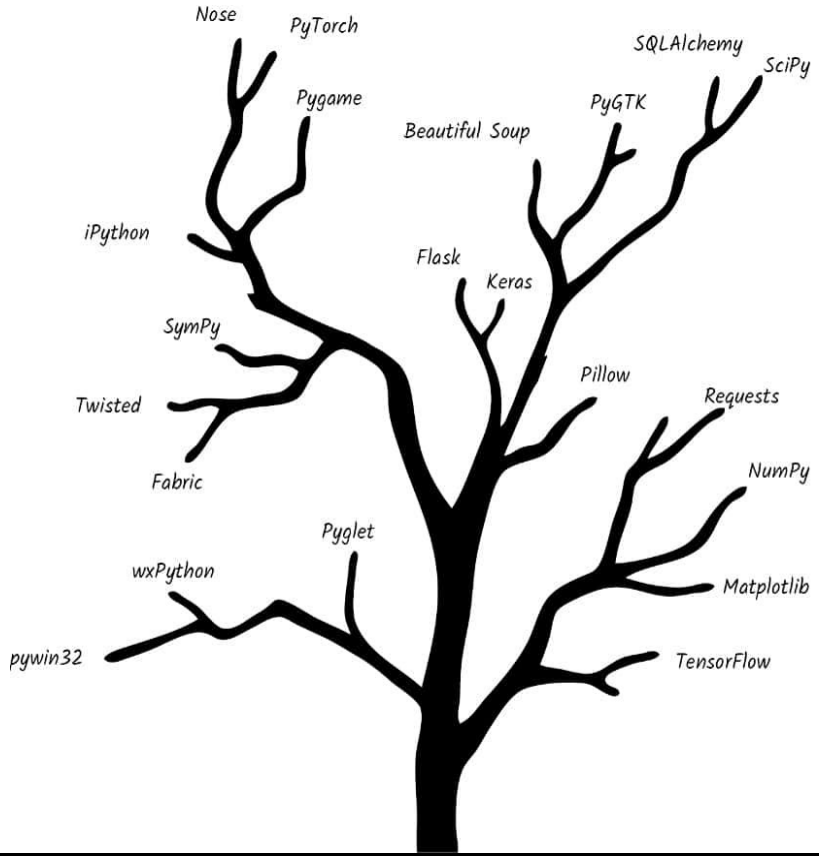
Numpy

Matplotlib.pyplot

Seaborn

Streamlit

Plotly.express



4.1 Pandas

Pandas is also a library or a data analysis tool in python which is written in the python programming language. It is mostly used for data analysis and data manipulation. It is also used for data structures and time series.

We can see the application of python in many fields such as - Economics, Recommendation Systems - Spotify, Netflix and Amazon, Stock Prediction, Neuroscience, Statistics, Advertising, Analytics, Natural Language Processing. Data can be analyzed in pandas in two ways -

Data frames - In this data is two-dimensional and consist of multiple series. Data is always represented in the rectangular table.

Series - In this data is one-dimensional and consists of a single list with an index.

```
top_sale_reg = games[['NA_Sales', 'EU_Sales', 'JP_Sales', 'Other_Sales']]
# pd.DataFrame(top_sale_reg.sum(), columns=['a', 'b'])
top_sale_reg = top_sale_reg.sum().reset_index()
top_sale_reg = top_sale_reg.rename(columns={"index": "region", 0: "sale"})
top_sale_reg
```

	region	sale
0	NA_Sales	4392.95
1	EU_Sales	2434.13
2	JP_Sales	1291.02
3	Other_Sales	797.75

Figure 4.1: data frame in pandas

4.2 NumPy

”NumPy is a library for the Python programming language, adding support for large, multi-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays”. The previous similar programming of NumPy is Numeric, and this language was created by Jim Hugunin with contributions from several other developers. In 2005, Travis Oliphant created NumPy by incorporating features of the computing Numarray into Numeric, with extensive modifications. It is an open-source library and free of cost.

```
122
123  st.subheader("Highest Grossing Publisher/Genre/Game in a Particular Year : ")
124
125  user_input_year = st.text_input("Enter a year between 1970 and 2016" ,2000)
126  user_input_field = st.selectbox("Choose an Option" , ["Publisher" , "Genre" , "Name"])
127  try:
128      x1 = np.int32(user_input_year)
129      if np.int32(user_input_year) not in list(data_year):
130          st.write("Not data found")
131      else:
132          specific_df= grouped_data.get_group(np.int32(user_input_year))[["Name" ,"Genre" , "Publisher", "Global_Sales"]]
133          if len(list(specific_df["Name"])) <= 15:
134              st.write(px.pie(specific_df, values = "Global_Sales" , names = user_input_field , width = 900))
135          else:
136
137              st.write(px.pie(specific_df[specific_df["Global_Sales"] > 2], values = "Global_Sales" , names = user_input_fiel
138
139          if st.checkbox("Show data" , False):
140              st.write(specific_df)
141  except ValueError:
142      st.write("Please enter a valid value")
```

Figure 4.2: NumPy basic example

4.3 Matplotlib.pyplot

matplotlib. pyplot is a collection of functions that make matplotlib work like MATLAB. Each pyplot function makes some change to a figure: e.g., creates a figure, creates a plotting area in a figure, plots some lines in a plotting area, decorates the plot with labels, etc.

In matplotlib. pyplot various states are preserved across function calls so that it keeps track of things like the current figure and plotting area, and the plotting functions are directed to the current axes.

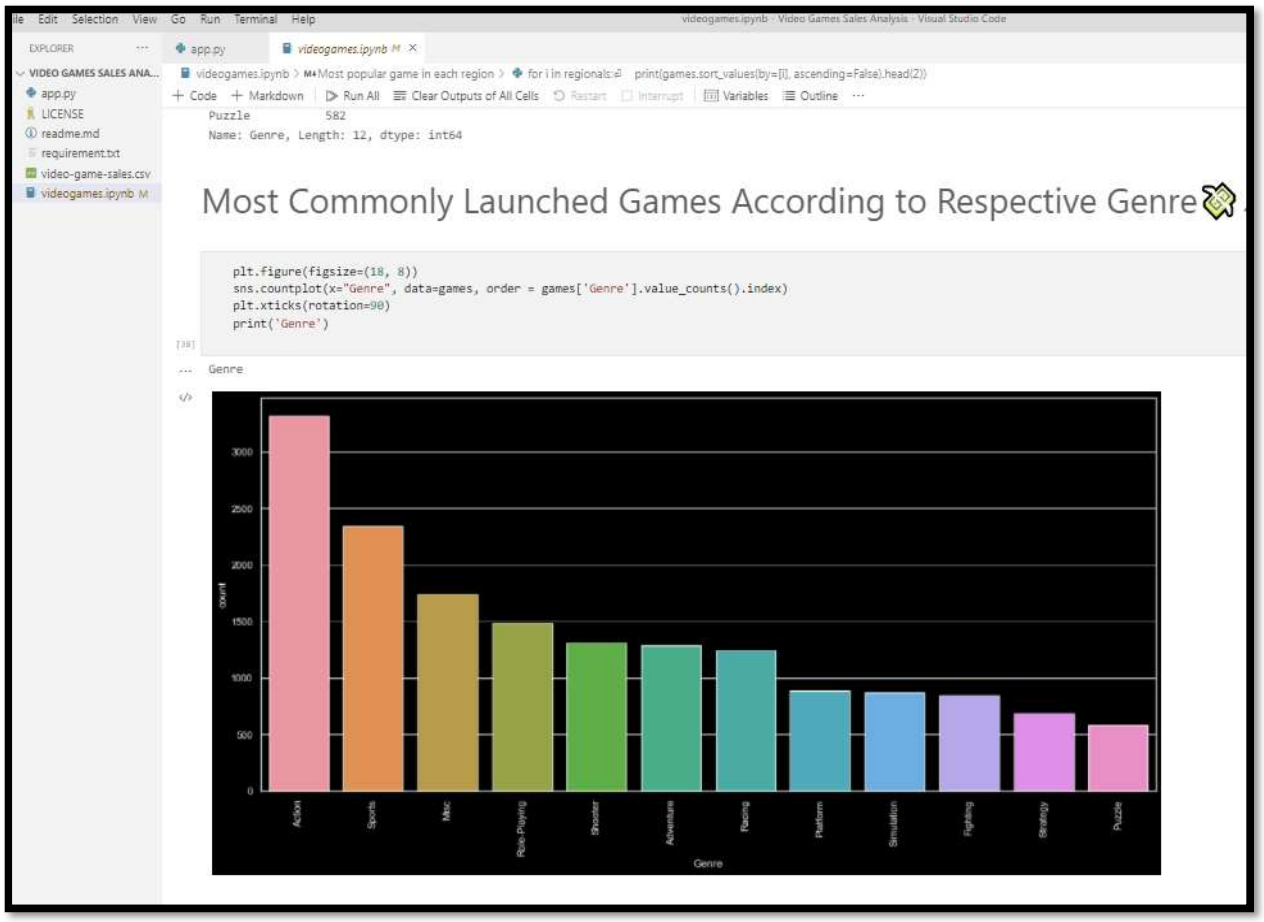


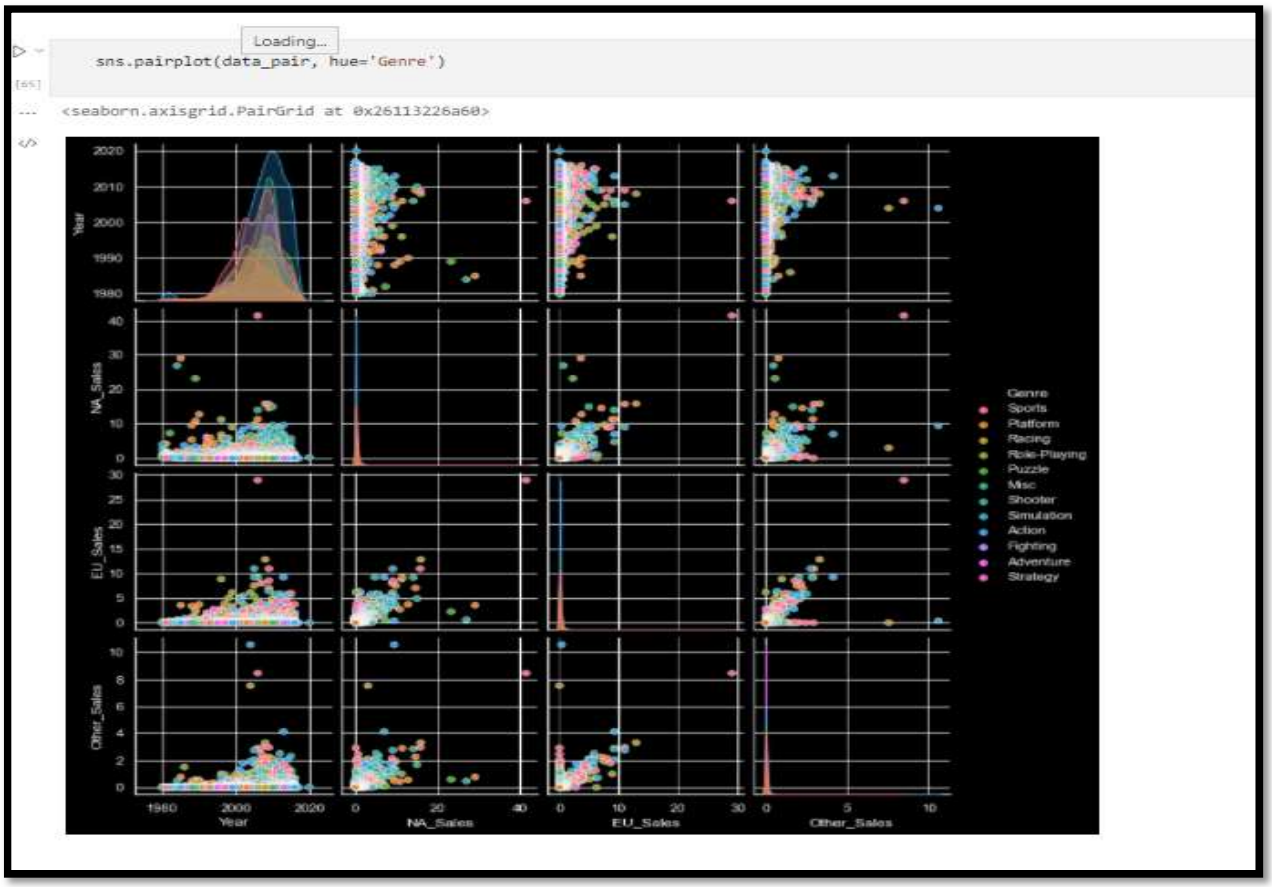
Figure 4.3: Matplotlib.pyplot basic example

4.4 Seaborn

Seaborn is an open-source Python library built on top of matplotlib. It is used for data visualization and exploratory data analysis. Seaborn works easily with data frames and the Pandas library. The graphs created can also be customized easily. Below are a few benefits of Data Visualization.

Graphs can help us find data trends that are useful in any machine learning or forecasting project.

- Graphs make it easier to explain your data to non-technical people.
- Visually attractive graphs can make presentations and reports much more appealing to the reader.



4.5 Streamlit

Streamlit is an open-source python framework for building web apps for Machine Learning and Data Science. We can instantly develop web apps and deploy them easily using Streamlit. Streamlit allows you to write an app the same way you write a python code. Streamlit makes it seamless to work on the interactive loop of coding and viewing results in the web app

```python
import pandas as pd
import numpy as np
import plotly.express as px
import streamlit as st

##############################################################################################
# This function loads the data and does some very basic data cleaning.
@st.cache(persist = True)
def load_data():
    data = pd.read_csv(('video-game-sales.csv'))
    data.dropna(inplace = True)
    data["Year"] = data["Year"].astype("int")
    data.set_index("Rank")
    temp = data["Year"]
    temp = temp[temp <= 2020]
    return data , temp



# Here data is grouped by the Year attribute.
@st.cache(persist = True , allow_output_mutation=True)
def groupingdata(data):
    x = data.groupby("Year")

    return x

def dark_theme():
    st.markdown("<style>h1{color : white;}</style>" , unsafe_allow_html = True)
    st.markdown("<style>h2{color : white;}</style>" , unsafe_allow_html = True)
    st.markdown("<style>h3{color : white;}</style>" , unsafe_allow_html = True)
    st.markdown("<style>h4{color : white;}</style>" , unsafe_allow_html = True)
    st.markdown("<style>h5{color : white;}</style>" , unsafe_allow_html = True)
    st.markdown("<style>h6{color : white;}</style>" , unsafe_allow_html = True)

```

Figure 4.5 Matplotlib basic example

The output of Fig 4.5

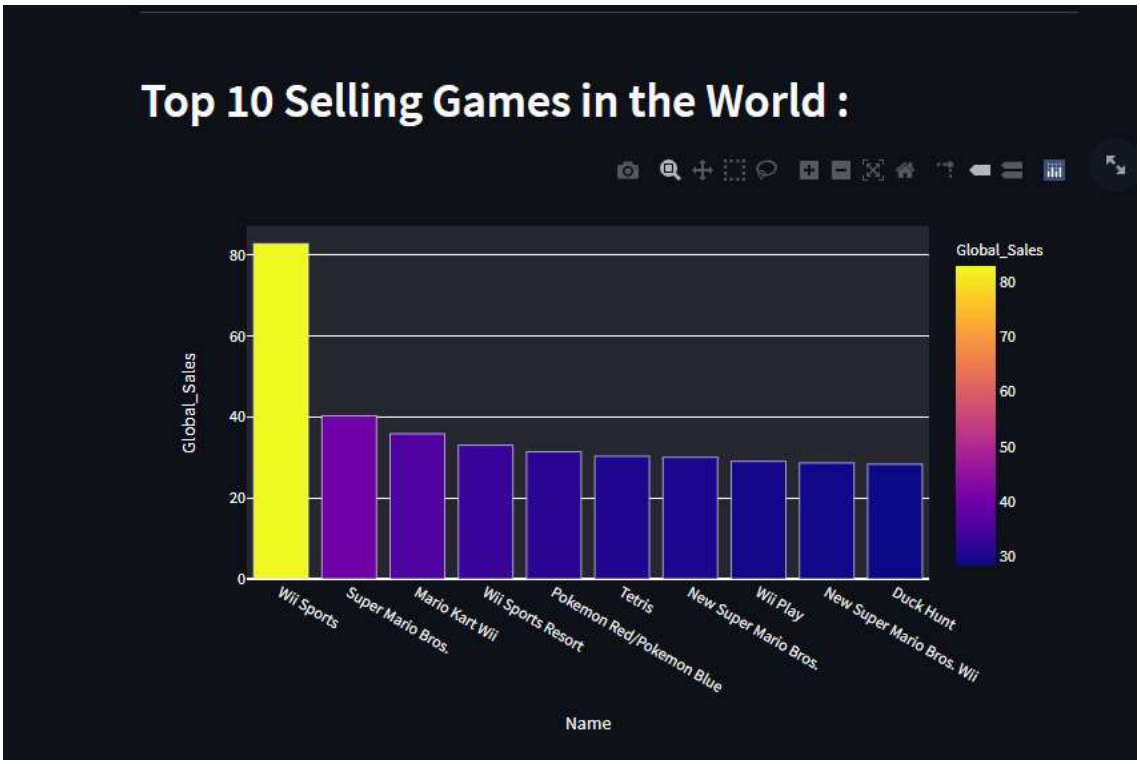
4.6 Plotly.express

The plotly.express module (usually imported as px) contains functions that can create entire figures at once, and is referred to as Plotly Express or PX. Plotly Express is a built-in part of the plotly library and is the recommended starting point for creating the most common figures. Every Plotly Express function uses graph objects internally and returns a plotly.graph_objects.Figure instance. Throughout the plotly documentation, you will find the Plotly Expressway of building figures at the top of any applicable page, followed by a section on how to use graph objects to build similar figures. Any figure created in a single function call with Plotly Express could be created using graph objects alone, but with between 5 and 100 times more code.

```
56  ###############################################################################
57
58
59  st.header("Top 10 Selling Games in the World : ")
60  other_sales = data[["Name","Global_Sales"]].head(10)
61
62  st.write(px.bar( other_sales,x = "Name" , y = "Global_Sales" , hover_data = ["Name" , "Global_Sales"] , color = "Global_Sal
63  if st.checkbox("Show Raw Data",False):
64      st.write(other_sales)
65  st.markdown("---")
66  ###############################################################################
```

Figure 4.7: Matplotlib basic example

The output of Fig 4.7



Section 5

5.1 Analysis On Streamlit

Lines Of Codes

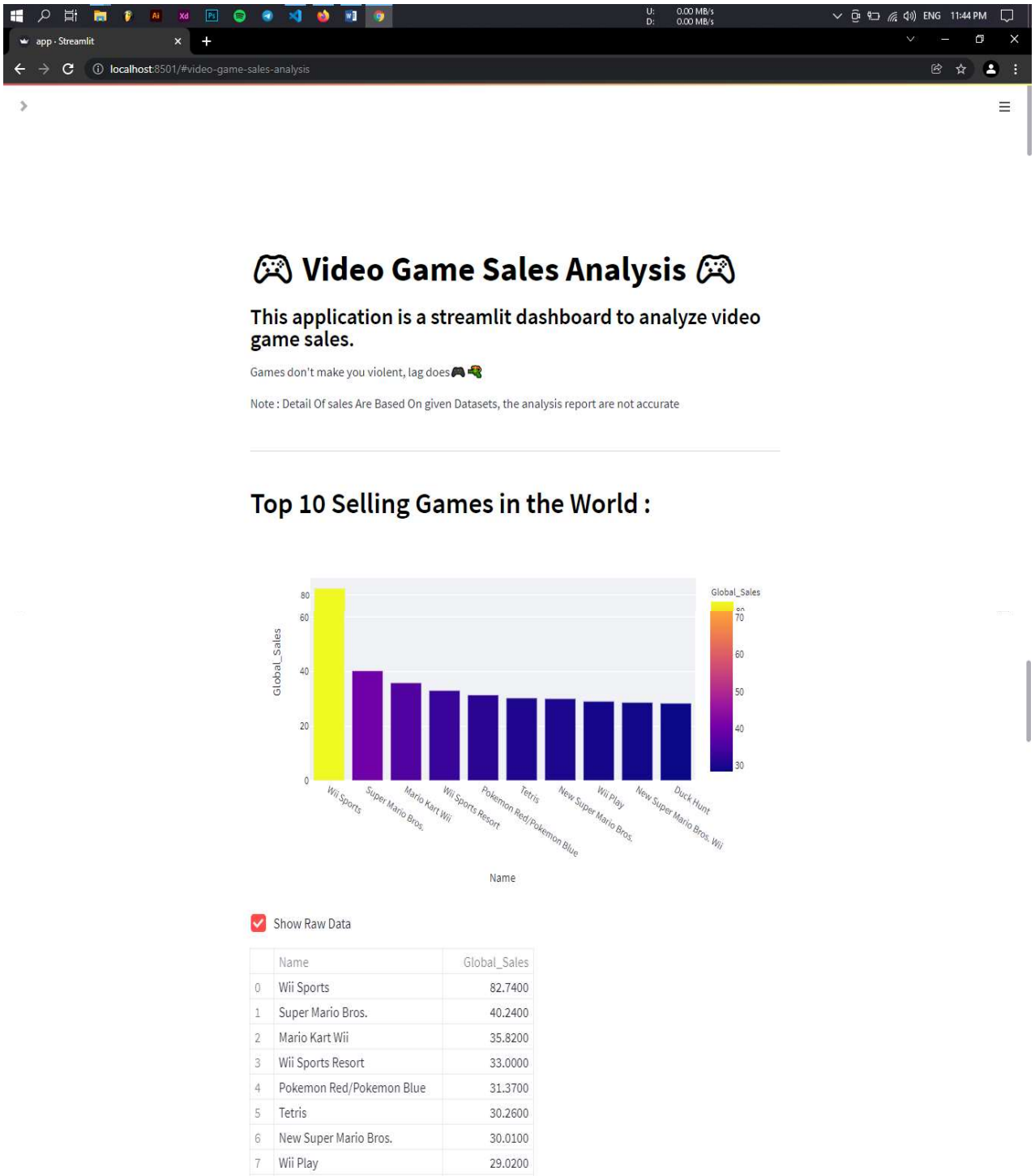
```
import pandas as pd
import numpy as np
import plotly.express as px
import streamlit as st

#############################################################################
# This function loads the data and does some very basic data cleaning.
@st.cache(persist = True)
def load_data():
    data = pd.read_csv(('video-game-sales.csv'))
    data.dropna(inplace = True)
    data["Year"] = data["Year"].astype("int")
    data.set_index("Rank")
    temp = data["Year"]
    temp = temp[temp <= 2020]
    return data , temp




# Here data is grouped by the Year attribute.
@st.cache(persist = True , allow_output_mutation=True)
def groupingdata(data):
    x = data.groupby("Year")

    return x

def dark_theme():
    st.markdown("<style>h1{color : black;}</style>" , unsafe_allow_html =
True)
    st.markdown("<style>h2{color : black;}</style>" , unsafe_allow_html =
True)
    st.markdown("<style>h3{color : black;}</style>" , unsafe_allow_html =
True)
    st.markdown("<style>h4{color : black;}</style>" , unsafe_allow_html =
True)
    st.markdown("<style>h5{color : black;}</style>" , unsafe_allow_html =
True)
    st.markdown("<style>h6{color : black;}</style>" , unsafe_allow_html =
True)


#function calling
```

```
data , data_year = load_data()

grouped_data = groupingdata(data)

theme = st.sidebar.selectbox("Choose a Theme" ,["Light" , "Dark"])
if theme == "Dark":
    dark_theme()

################################################################################
# The Title and Subheader for the web app.

st.title(" 🎮 Video Game Sales Analysis 🎮 ")
st.subheader("This application is a streamlit dashboard to analyze video game
sales.")
st.markdown("Games don't make you violent, lag does🎮🔫")
st.markdown("Note : Detail Of sales Are Based On given Datasets, the analysis
report are not accurate")
st.markdown("---")

################################################################################
st.header("Top 10 Selling Games in the World : ")
other_sales = data[["Name","Global_Sales"]].head(10)

st.write(px.bar( other_sales,x = "Name" , y = "Global_Sales" , hover_data =
["Name" , "Global_Sales"] , color = "Global_Sales" ))
if st.checkbox("Show Raw Data",False):
    st.write(other_sales)
st.markdown("---")
################################################################################
st.header("Top Selling Games  in different parts of the world : ")
check = st.selectbox("Select a option" , ["North
America","Europe","Japan","Rest of the World"])

if check == "North America":

    st.subheader(" Highest Grossing Games in North America.")
    other_sales = data[["Name","Year","NA_Sales"]].sort_values(by =
"NA_Sales",ascending = False).head(10)
    st.write(px.bar(other_sales, x = "Name" , y = "NA_Sales" , hover_data =
["Name" , "NA_Sales"] , color = "NA_Sales"))

if check == "Europe":

    st.subheader("Highest Grossing Games in Europe.")
    other_sales = data[["Name","Year","EU_Sales"]].sort_values(by =
"EU_Sales",ascending = False).head(10)
    st.write(px.bar(other_sales, x = "Name" , y = "EU_Sales" , hover_data =
["Name" , "EU_Sales"] , color = "EU_Sales"))
```

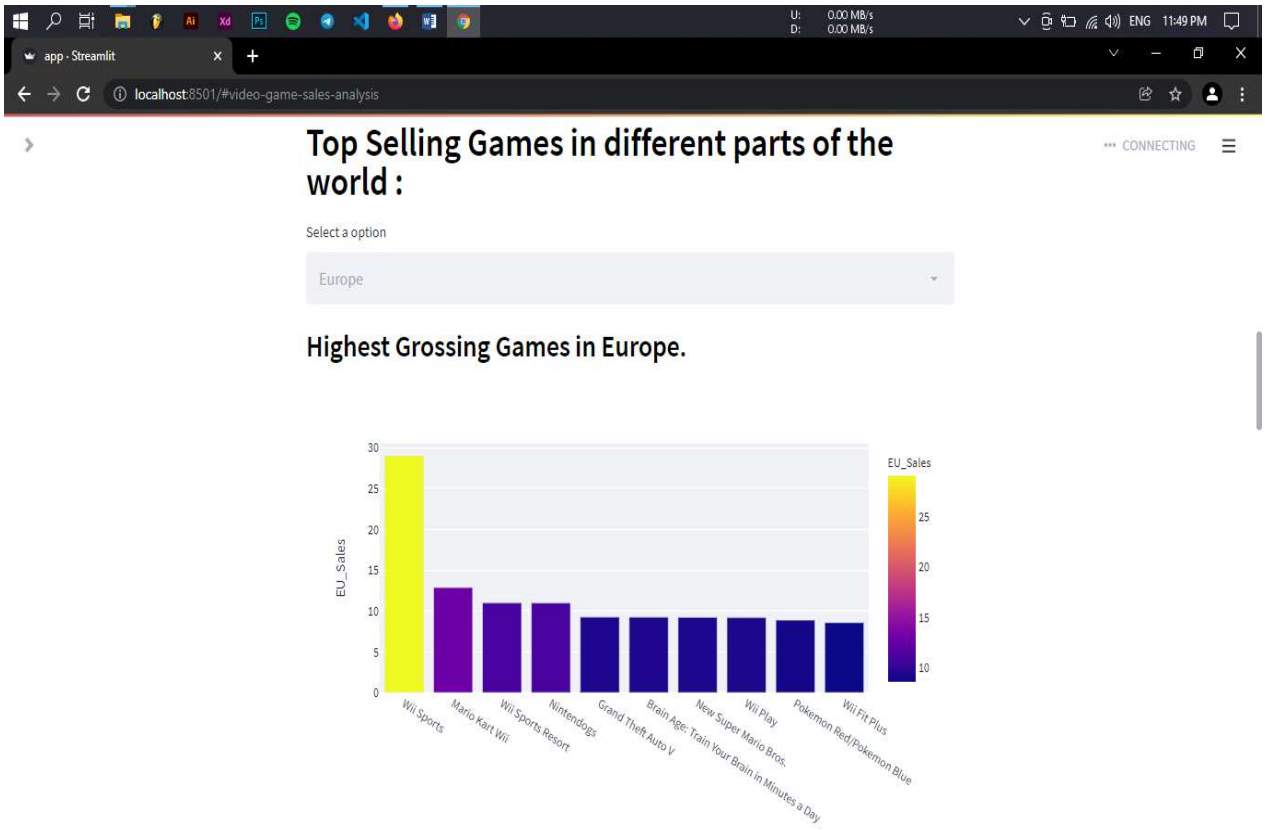
```
if check == "Japan":

    st.subheader("Highest Grossing Games in Japan.")
    other_sales = data[["Name","Year","JP_Sales"]].sort_values(by =
"JP_Sales",ascending = False).head(10)
    st.write(px.bar(other_sales, x = "Name" , y = "JP_Sales" , hover_data =
["Name" , "JP_Sales"] , color = "JP_Sales"))


elif check == "Rest of the World":
    st.subheader("Highest Grossing Games in Rest of the world.")
    other_sales = data[["Name","Year","Other_Sales"]].sort_values(by =
"Other_Sales",ascending = False).head(10)
    st.write(px.bar(other_sales, x = "Name" , y = "Other_Sales" , hover_data =
["Name" , "Other_Sales"] , color = "Other_Sales"))

if st.checkbox("Show Raw data",False):
    st.write(other_sales)

st.markdown("---")
##############################################################################
st.subheader("Highest Grossing Genres : ")
other_sales =
data[["Genre","Global_Sales"]].groupby("Genre").agg("sum").sort_values(by =
"Global_Sales",ascending = False)
other_sales["Genre"] = other_sales.index
st.write(px.bar(other_sales, x = "Genre" , y = "Global_Sales" , hover_data =
["Genre" , "Global_Sales"] , color = "Global_Sales"))
if st.checkbox("show Raw Data",False):
    st.write(other_sales)

st.markdown("---")

##############################################################################
st.subheader("Highest Grossing Publishers : ")
other_sales =
data[["Publisher","Global_Sales"]].groupby("Publisher").agg("sum").sort_values
(by = "Global_Sales",ascending = False).head(20)
other_sales["Publisher"] = other_sales.index
st.write(px.bar(other_sales, x = "Publisher" , y = "Global_Sales" , hover_data
= ["Publisher" , "Global_Sales"] , color = "Global_Sales"))
if st.checkbox("Show raw Data",False):
    st.write(other_sales)

st.markdown("---")

##############################################################################
st.subheader("Highest Grossing Publisher/Genre/Game in a Particular Year : ")

user_input_year = st.text_input("Enter a year between 1970 and 2016" ,2000)
```

```
user_input_field = st.selectbox("Choose an Option" , ["Publisher" , "Genre" ,
"Name"])
try:
    x1 = np.int32(user_input_year)
    if np.int32(user_input_year) not in list(data_year):
        st.write("Not data found")
    else:
        specific_df= grouped_data.get_group(np.int32(user_input_year))[["Name"
,"Genre" , "Publisher", "Global_Sales"]]
        if len(list(specific_df["Name"])) <= 15:
            st.write(px.pie(specific_df, values = "Global_Sales" , names =
user_input_field , width = 900))
        else:


            st.write(px.pie(specific_df[specific_df["Global_Sales"] > 2],
values = "Global_Sales" , names = user_input_field , width = 900))

        if st.checkbox("Show data" , False):
            st.write(specific_df)
except ValueError:
    st.write("Please enter a valid value")



st.markdown("---")
```

5.2 Output

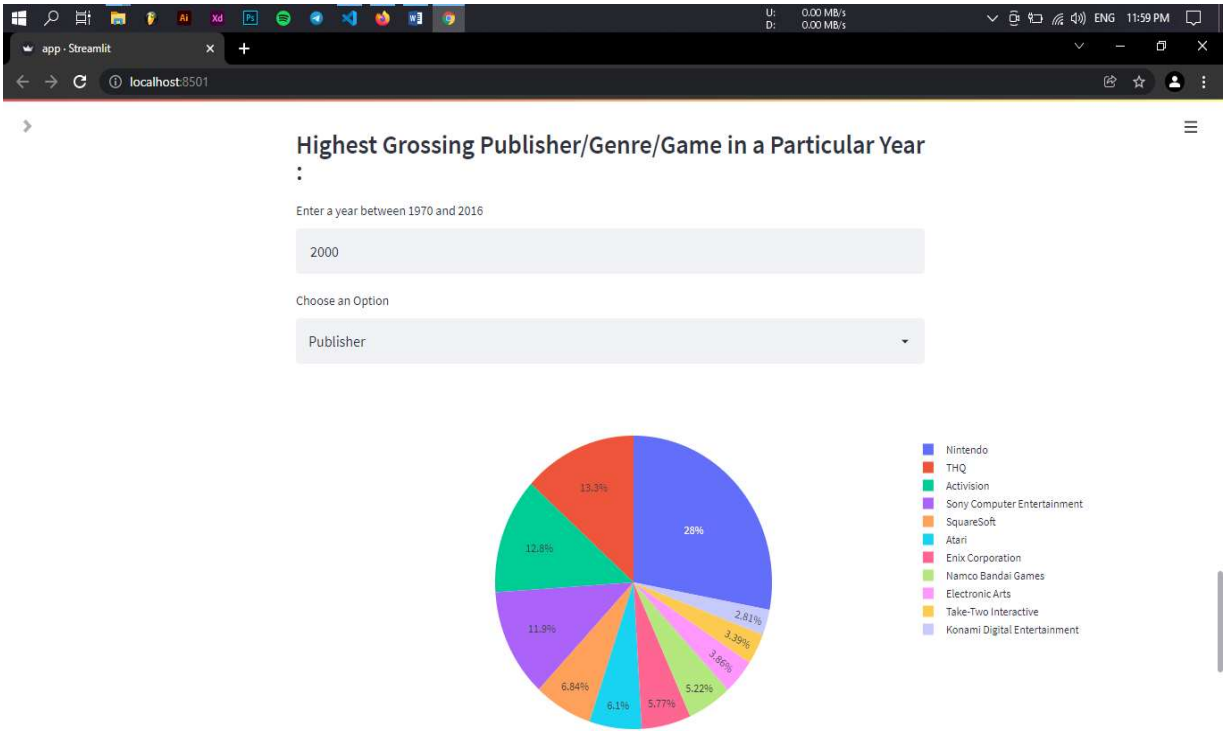


	Name	Global_Sales
0	Wii Sports	82.7400
1	Super Mario Bros.	40.2400
2	Mario Kart Wii	35.8200
3	Wii Sports Resort	33.0000
4	Pokemon Red/Pokemon Blue	31.3700
5	Tetris	30.2600
6	New Super Mario Bros.	30.0100
7	Wii Play	29.0200


app · Streamlit
localhost:8501/#video-game-sales-analysis
Top Selling Games in different parts of the world :
CONNECTING
Select a option
Europe
Highest Grossing Games in Europe.
EU_Sales
30
25
20
15
10
5
0
Wii Sports
Mario Kart Wii
Wii Sports Resort
Nintendogs
Grand Theft Auto V
Brain Age: Train Your Brain in Minutes a Day
New Super Mario Bros.
Wii Play
Pokemon Red/Pokemon Blue
Wii Fit Plus


app · Streamlit
localhost:8501/#video-game-sales-analysis
Highest Grossing Genres :
CONNECTING
Global_Sales
1500
1000
500
0
Action
Sports
Shooter
Role-Playing
Platform
Misc
Racing
Fighting
Simulation
Puzzle
Adventure
Strategy
Genre
show Raw Data


app · Streamlit
localhost:8501/#video-game-sales-analysis
CONNECTING
Highest Grossing Publishers :
Global_Sales
1500
1000
500
0
Nintendo
Electronic Arts
Activision
Sony Computer Entertainment
Ubisoft
Take-Two Interactive
THQ
Konami Digital Entertainment
Sega
Namco Bandai Games
Microsoft Game Studios
Capcom
Atari
Square Enix
Warner Bros. Interactive Entertainment
Disney Interactive Studios
Eidos Interactive
LucasArts
Bethesda Softworks
Midway Games
Publisher
Show raw Data


app · Streamlit
localhost:8501
Highest Grossing Publisher/Genre/Game in a Particular Year :
Enter a year between 1970 and 2016
2000
Choose an Option
Publisher
28%
13.3%
12.8%
11.9%
6.84%
6.1%
5.77%
5.22%
3.86%
3.39%
2.81%
Nintendo
THQ
Activision
Sony Computer Entertainment
SquareSoft
Atari
Enix Corporation
Namco Bandai Games
Electronic Arts
Take-Two Interactive
Konami Digital Entertainment

References

- https://www.vgchartz.com/gamedb/
- https://blog.jovian.ai/trends-on-video-game-sales-using-exploratory-data-analysis-and-case-study-c4fb4e17fa67
- https://datascience.fm/video-game-sales-analysis
- Automate the Boring Stuff with Python: Practical Programming for Total Beginners Book by Al Sweiga
- Data Analytics using Python by Bharti Motwani
- https://www.kaggle.com/snanilim/video-games-sales-analysis-and-visualization

*****