ALLIANCE
UNIVERSITY

Private University established in Karnataka State by Act No.34 of year 2010
Recognized by the University Grants Commission (UGC), New Delhi

Project Report

Introduction To Data Science

Semester — II

Video Game Sales

By Mahir Mavani

Reg No :- 2411021240003

Git Hub Link :- https://github.com/MahirMavani/ids-assignment

Department of Computer Application

Alliance University

Chandapura — Anekal Main Road, Anekal

Bengaluru — 562 106

April 2025

Introduction :

The video game industry has witnessed remarkable growth and evolution over the decades, producing some of the most iconic and best-selling titles in entertainment history. From the early days of 8-bit consoles to modern gaming platforms, certain games have transcended generations, captivating millions of players worldwide. This dataset highlights the top 22 best-selling video games of all time, showcasing their platforms, release years, genres, publishers, and regional sales figures.

Nintendo dominates the list, with classics like Wii Sports, Super Mario Bros., and Pokemon Red/Blue leading the charge. These games not only defined their respective eras but also set benchmarks for innovation and player engagement. Other notable entries include Grand Theft Auto V and Kinect Adventures!, which represent the success of non-Nintendo platforms like PlayStation and Xbox. The dataset also reflects the diversity of gaming genres, ranging from sports and racing to role-playing and action.

The global sales figures underscore the widespread appeal of these games, with some titles achieving massive success in specific regions, such as Japan or North America. This list serves as a testament to the enduring legacy of these games and their impact on the gaming industry and popular culture.

Project Goal

- To clean and preprocess the raw dataset for accurate analysis.
- To explore trends and patterns in video game sales.
- To build a regression model that can predict global sales.
- To build a classification model categorizing games into Low, Medium, and High sales.
- To evaluate model performance using appropriate metrics

Data Preprocessing :-

1) Understand the Dataset

- The dataset contains information about the top-selling video games, including their rank, name, platform, release year, genre, publisher, and sales figures across different regions (NA, EU, JP, Other) and globally.
- The goal is to clean and prepare the data for analysis by ensuring consistency, accuracy, and usability.

2) Check for Missing Values

- Inspect all columns for missing or null values, especially in critical fields like Name, Platform, Year, and sales columns (NA_Sales, EU_Sales, etc.).
- If missing values are found:

- For numerical columns, consider imputing with the mean, median, or zero, depending on the context.
- For categorical columns, replace missing values with "Unknown" or drop the rows if they are not significant.

3) Verify Data Types

- Ensure that each column has the correct data type:
- Rank, Year: Integer.
- NA_Sales, EU_Sales, JP_Sales, Other_Sales, Global_Sales: Float.
- Name, Platform, Genre, Publisher: String or categorical.

4) Check for Duplicates

- Identify and remove duplicate rows, if any, to avoid over-representation of specific games.
- Use the Name column as the primary identifier to check for duplicates.

5) Validate Data Consistency

- Ensure that the Global_Sales column equals the sum of NA_Sales, EU_Sales, JP_Sales, and Other_Sales for each row.
- If discrepancies are found, recalculate Global_Sales or investigate the source of the inconsistency.

6) Handle Outliers

- Identify outliers in sales columns using statistical methods like the interquartile range (IQR) or z-scores.
- Decide whether to keep, cap, or remove outliers based on their relevance to the analysis.

7) Standardize Categorical Data

- Standardize the text in categorical columns (Platform, Genre, Publisher) to ensure consistency (e.g., "Nintendo" vs. "nintendo").
- Convert these columns to categorical data types for efficient storage and processing.

8) Feature Engineering

- Create new columns to enhance analysis:
- Regional Contribution: Calculate the percentage contribution of each region to global sales (e.g., NA_Percentage = (NA_Sales / Global_Sales) * 100).
- Decade: Derive a "Decade" column from the Year column (e.g., 1980s, 1990s).
- Group games by Platform, Genre, or Publisher to analyze trends.

9) Normalize or Scale Sales Data

- If required for machine learning or statistical analysis, normalize or scale the sales columns (NA_Sales, EU_Sales, etc.) to bring them to a similar range.

10) Check for Logical Errors

- Verify that the Year column contains valid years (e.g., no future dates or unrealistic values).
- Ensure that the Rank column is unique and sequential.

11) Save the Cleaned Dataset

- After completing all preprocessing steps, save the cleaned dataset to a new file (e.g., CSV or Excel) for further analysis.

Project Overview

This project focuses on building an Intrusion Detection System (IDS) using machine learning techniques. The IDS is designed to identify and classify network traffic as either normal or a type of attack/intrusion, helping to secure computer networks against malicious activity.

The core of the project includes:

- Data preprocessing and feature selection
- Training and evaluating a machine learning model (e.g., Decision Tree, Random Forest, etc.)
- Classification of network connections
- Performance evaluation using metrics like accuracy, precision, recall, and F1-score

Project Insights

Here are some key insights gained from the project:

- Data Preprocessing is Crucial: Handling categorical features, scaling numeric values, and feature selection significantly impact the model's performance.
- Feature Importance: Using feature importance techniques (e.g., from tree-based models) helps in selecting the most relevant features, improving efficiency and reducing overfitting.
- Model Selection: Various classifiers (like Random Forest, Decision Tree, etc.) were likely tested. Ensemble models tend to perform better in classification tasks involving complex patterns like intrusions.
- Evaluation Metrics: Besides accuracy, other metrics like precision and recall are important—especially in security, where false negatives (missed attacks) can be dangerous.

Project Goal

- The primary goal of this IDS project is to develop an effective and accurate machine learning model that can:
- Detect various types of network attacks (e.g., DoS, Probe, R2L, U2R)

- Reduce false positives and false negatives
- Enhance the security of computer networks by automatically identifying potential intrusions in real time or near real time

Creating a Dataframe

```
import pandas as pd
df=pd.read_csv(r"D:\downloads\video games sales.csv")
df
```

```
        Rank                                            Name Platform  \
0          1                                  Wii Sports      Wii
1          2                              Super Mario Bros.      NES
2          3                                 Mario Kart Wii      Wii
3          4                              Wii Sports Resort      Wii
4          5                        Pokemon Red/Pokemon Blue       GB
...      ...                                             ...      ...
16593  16596          Woody Woodpecker in Crazy Castle 5      GBA
16594  16597                Men in Black II: Alien Escape       GC
16595  16598  SCORE International Baja 1000: The Official Game      PS2
16596  16599                                  Know How 2       DS
16597  16600                               Spirits & Spells      GBA

         Year         Genre   Publisher  NA_Sales  EU_Sales  JP_Sales  \
0      2006.0        Sports    Nintendo     41.49     29.02      3.77
1      1985.0      Platform    Nintendo     29.08      3.58      6.81
2      2008.0        Racing    Nintendo     15.85     12.88      3.79
3      2009.0        Sports    Nintendo     15.75     11.01      3.28
4      1996.0  Role-Playing    Nintendo     11.27      8.89     10.22
...       ...           ...         ...       ...       ...       ...
16593  2002.0      Platform       Kemco      0.01      0.00      0.00
16594  2003.0       Shooter  Infogrames      0.01      0.00      0.00
16595  2008.0        Racing  Activision      0.00      0.00      0.00
16596  2010.0        Puzzle    7G//AMES      0.00      0.01      0.00
16597  2003.0      Platform     Wanadoo      0.01      0.00      0.00

       Other_Sales  Global_Sales
0             8.46         82.74
1             0.77         40.24
2             3.31         35.82
3             2.96         33.00
4             1.00         31.37
...            ...           ...
16593         0.00          0.01
16594         0.00          0.01
16595         0.00          0.01
16596         0.00          0.01
16597         0.00          0.01

[16598 rows x 11 columns]
```

```
df.info
```

```
<bound method DataFrame.info of         Rank
Name Platform  \
0          1                                       Wii Sports      Wii
1          2                                 Super Mario Bros.      NES
2          3                                    Mario Kart Wii      Wii
3          4                                 Wii Sports Resort      Wii
4          5                           Pokemon Red/Pokemon Blue       GB
...      ...                                               ...      ...
16593  16596                Woody Woodpecker in Crazy Castle 5      GBA
16594  16597                     Men in Black II: Alien Escape       GC
16595  16598  SCORE International Baja 1000: The Official Game      PS2
16596  16599                                       Know How 2       DS
16597  16600                                  Spirits & Spells      GBA

         Year         Genre   Publisher  NA_Sales  EU_Sales  JP_Sales  \
0      2006.0        Sports    Nintendo     41.49     29.02      3.77
1      1985.0      Platform    Nintendo     29.08      3.58      6.81
2      2008.0        Racing    Nintendo     15.85     12.88      3.79
3      2009.0        Sports    Nintendo     15.75     11.01      3.28
4      1996.0  Role-Playing    Nintendo     11.27      8.89     10.22
...       ...           ...         ...       ...       ...       ...
16593  2002.0      Platform       Kemco      0.01      0.00      0.00
16594  2003.0       Shooter  Infogrames      0.01      0.00      0.00
16595  2008.0        Racing  Activision      0.00      0.00      0.00
16596  2010.0        Puzzle    7G//AMES      0.00      0.01      0.00
16597  2003.0      Platform     Wanadoo      0.01      0.00      0.00

       Other_Sales  Global_Sales
0             8.46         82.74
1             0.77         40.24
2             3.31         35.82
3             2.96         33.00
4             1.00         31.37
...            ...           ...
16593         0.00          0.01
16594         0.00          0.01
16595         0.00          0.01
16596         0.00          0.01
16597         0.00          0.01

[16598 rows x 11 columns]>
```

```
df.describe()
```

```
               Rank          Year      NA_Sales      EU_Sales      JP_Sales
\
count  16598.000000  16327.000000  16598.000000  16598.000000  16598.000000
mean    8300.605254   2006.406443      0.264667      0.146652      0.077782
```

```
std     4791.853933      5.828981      0.816683      0.505351      0.309291
min        1.000000   1980.000000      0.000000      0.000000      0.000000
25%     4151.250000   2003.000000      0.000000      0.000000      0.000000
50%     8300.500000   2007.000000      0.080000      0.020000      0.000000
75%    12449.750000   2010.000000      0.240000      0.110000      0.040000
max    16600.000000   2020.000000     41.490000     29.020000     10.220000

        Other_Sales  Global_Sales
count  16598.000000  16598.000000
mean       0.048063      0.537441
std        0.188588      1.555028
min        0.000000      0.010000
25%        0.000000      0.060000
50%        0.010000      0.170000
75%        0.040000      0.470000
max       10.570000     82.740000
```

```
df.head()
```

```
   Rank                     Name Platform    Year         Genre Publisher  \
0     1                Wii Sports      Wii  2006.0        Sports  Nintendo
1     2          Super Mario Bros.      NES  1985.0      Platform  Nintendo
2     3             Mario Kart Wii      Wii  2008.0        Racing  Nintendo
3     4          Wii Sports Resort      Wii  2009.0        Sports  Nintendo
4     5  Pokemon Red/Pokemon Blue       GB  1996.0  Role-Playing  Nintendo

   NA_Sales  EU_Sales  JP_Sales  Other_Sales  Global_Sales
0     41.49     29.02      3.77         8.46         82.74
1     29.08      3.58      6.81         0.77         40.24
2     15.85     12.88      3.79         3.31         35.82
3     15.75     11.01      3.28         2.96         33.00
4     11.27      8.89     10.22         1.00         31.37
```

```
df.tail()
```

```
        Rank                                           Name Platform  \
16593  16596                Woody Woodpecker in Crazy Castle 5      GBA
16594  16597                     Men in Black II: Alien Escape       GC
16595  16598  SCORE International Baja 1000: The Official Game      PS2
16596  16599                                    Know How 2       DS
16597  16600                                 Spirits & Spells      GBA

         Year     Genre   Publisher  NA_Sales  EU_Sales  JP_Sales  \
16593  2002.0  Platform       Kemco      0.01      0.00       0.0
16594  2003.0   Shooter  Infogrames      0.01      0.00       0.0
16595  2008.0    Racing  Activision      0.00      0.00       0.0
16596  2010.0    Puzzle    7G//AMES      0.00      0.01       0.0
16597  2003.0  Platform     Wanadoo      0.01      0.00       0.0

       Other_Sales  Global_Sales
```

```
16593          0.0          0.01
16594          0.0          0.01
16595          0.0          0.01
16596          0.0          0.01
16597          0.0          0.01
```

Accessing Columns

```
df['Genre']
```

```
0              Sports
1            Platform
2              Racing
3              Sports
4        Role-Playing
             ...
16593        Platform
16594         Shooter
16595          Racing
16596          Puzzle
16597        Platform
Name: Genre, Length: 16598, dtype: object
```

```
df['Platform']
```

```
0        Wii
1        NES
2        Wii
3        Wii
4         GB
        ...
16593    GBA
16594     GC
16595    PS2
16596     DS
16597    GBA
Name: Platform, Length: 16598, dtype: object
```

Accessing Rows

```
df.iloc[0]
```

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

```
Global_Sales         82.74
Name: 0, dtype: object
```

```
df.dropna(inplace=True)
print("After dropping the null values (inplace=True):")
df.head()
```

```
After dropping the null values (inplace=True):
```

	Rank	Name	Platform	Year	Genre	Publisher	\
0	1	Wii Sports	Wii	2006.0	Sports	Nintendo	
1	2	Super Mario Bros.	NES	1985.0	Platform	Nintendo	
2	3	Mario Kart Wii	Wii	2008.0	Racing	Nintendo	
3	4	Wii Sports Resort	Wii	2009.0	Sports	Nintendo	
4	5	Pokemon Red/Pokemon Blue	GB	1996.0	Role-Playing	Nintendo	

	NA_Sales	EU_Sales	JP_Sales	Other_Sales	Global_Sales
0	41.49	29.02	3.77	8.46	82.74
1	29.08	3.58	6.81	0.77	40.24
2	15.85	12.88	3.79	3.31	35.82
3	15.75	11.01	3.28	2.96	33.00
4	11.27	8.89	10.22	1.00	31.37

```
df.dropna(inplace=False)
print("After dropping the null values (inplace=False):")
df.head()
```

```
After dropping the null values (inplace=False):
```

	Rank	Name	Platform	Year	Genre	Publisher	\
0	1	Wii Sports	Wii	2006.0	Sports	Nintendo	
1	2	Super Mario Bros.	NES	1985.0	Platform	Nintendo	
2	3	Mario Kart Wii	Wii	2008.0	Racing	Nintendo	
3	4	Wii Sports Resort	Wii	2009.0	Sports	Nintendo	
4	5	Pokemon Red/Pokemon Blue	GB	1996.0	Role-Playing	Nintendo	

	NA_Sales	EU_Sales	JP_Sales	Other_Sales	Global_Sales
0	41.49	29.02	3.77	8.46	82.74
1	29.08	3.58	6.81	0.77	40.24
2	15.85	12.88	3.79	3.31	35.82
3	15.75	11.01	3.28	2.96	33.00
4	11.27	8.89	10.22	1.00	31.37

Setting the index in a DataFrame

```
df.set_index('Name', inplace=True)
df.head()
```

Name	Rank	Platform	Year	Genre	Publisher	\
Wii Sports	1	Wii	2006.0	Sports	Nintendo	
Super Mario Bros.	2	NES	1985.0	Platform	Nintendo	

```
Mario Kart Wii               3      Wii  2008.0        Racing  Nintendo
Wii Sports Resort            4      Wii  2009.0        Sports  Nintendo
Pokemon Red/Pokemon Blue     5       GB  1996.0  Role-Playing  Nintendo

                          NA_Sales  EU_Sales  JP_Sales  Other_Sales  \
Name
Wii Sports                   41.49     29.02      3.77         8.46
Super Mario Bros.            29.08      3.58      6.81         0.77
Mario Kart Wii               15.85     12.88      3.79         3.31
Wii Sports Resort            15.75     11.01      3.28         2.96
Pokemon Red/Pokemon Blue     11.27      8.89     10.22         1.00

                          Global_Sales
Name
Wii Sports                       82.74
Super Mario Bros.                40.24
Mario Kart Wii                   35.82
Wii Sports Resort                33.00
Pokemon Red/Pokemon Blue         31.37
```

```
df.reset_index(inplace=True)
print("After resetting the index:")
df.head()
```

```
After resetting the index:

                      Name  Rank Platform    Year         Genre Publisher  \
0                Wii Sports     1      Wii  2006.0        Sports  Nintendo
1          Super Mario Bros.     2      NES  1985.0      Platform  Nintendo
2             Mario Kart Wii     3      Wii  2008.0        Racing  Nintendo
3          Wii Sports Resort     4      Wii  2009.0        Sports  Nintendo
4  Pokemon Red/Pokemon Blue     5       GB  1996.0  Role-Playing  Nintendo

   NA_Sales  EU_Sales  JP_Sales  Other_Sales  Global_Sales
0     41.49     29.02      3.77         8.46         82.74
1     29.08      3.58      6.81         0.77         40.24
2     15.85     12.88      3.79         3.31         35.82
3     15.75     11.01      3.28         2.96         33.00
4     11.27      8.89     10.22         1.00         31.37
```

Accessing a Column

```
print(df.loc[:,'Name'])
```

```
0                                        Wii Sports
1                                 Super Mario Bros.
2                                    Mario Kart Wii
3                                 Wii Sports Resort
4                           Pokemon Red/Pokemon Blue
                              ...
16286                 Woody Woodpecker in Crazy Castle 5
16287                      Men in Black II: Alien Escape
```

```
16288    SCORE International Baja 1000: The Official Game
16289                                          Know How 2
16290                                    Spirits & Spells
Name: Name, Length: 16291, dtype: object
```

```
print(df.iloc[:,2])
```

```
0        Wii
1        NES
2        Wii
3        Wii
4         GB
        ...
16286    GBA
16287     GC
16288    PS2
16289     DS
16290    GBA
Name: Platform, Length: 16291, dtype: object
```

Handling the Missing Values

Handling missing values involves first identifying them using tools like .isnull(). If few, missing rows or columns can be dropped. Otherwise, values are filled using mean, median, or mode for numerical data, and mode or placeholders for categorical data. Advanced methods include predictive imputation. Finally, the data is rechecked for consistency

Removing the missing values

```
df_clean=df.dropna()
df_clean.head()
```

```
                      Name  Rank Platform    Year         Genre Publisher  \
0                Wii Sports     1      Wii  2006.0        Sports  Nintendo
1         Super Mario Bros.     2      NES  1985.0      Platform  Nintendo
2            Mario Kart Wii     3      Wii  2008.0        Racing  Nintendo
3         Wii Sports Resort     4      Wii  2009.0        Sports  Nintendo
4  Pokemon Red/Pokemon Blue     5       GB  1996.0  Role-Playing  Nintendo

   NA_Sales  EU_Sales  JP_Sales  Other_Sales  Global_Sales
0     41.49     29.02      3.77         8.46         82.74
1     29.08      3.58      6.81         0.77         40.24
2     15.85     12.88      3.79         3.31         35.82
3     15.75     11.01      3.28         2.96         33.00
4     11.27      8.89     10.22         1.00         31.37
```

Data Cleaning

Data cleaning is the process of correcting or removing inaccurate, incomplete, or irrelevant data to improve data quality. It involves handling missing values, removing duplicates, fixing data type issues, correcting inconsistencies, and filtering out outliers. Clean data ensures better analysis, accurate insights, and reliable model performance

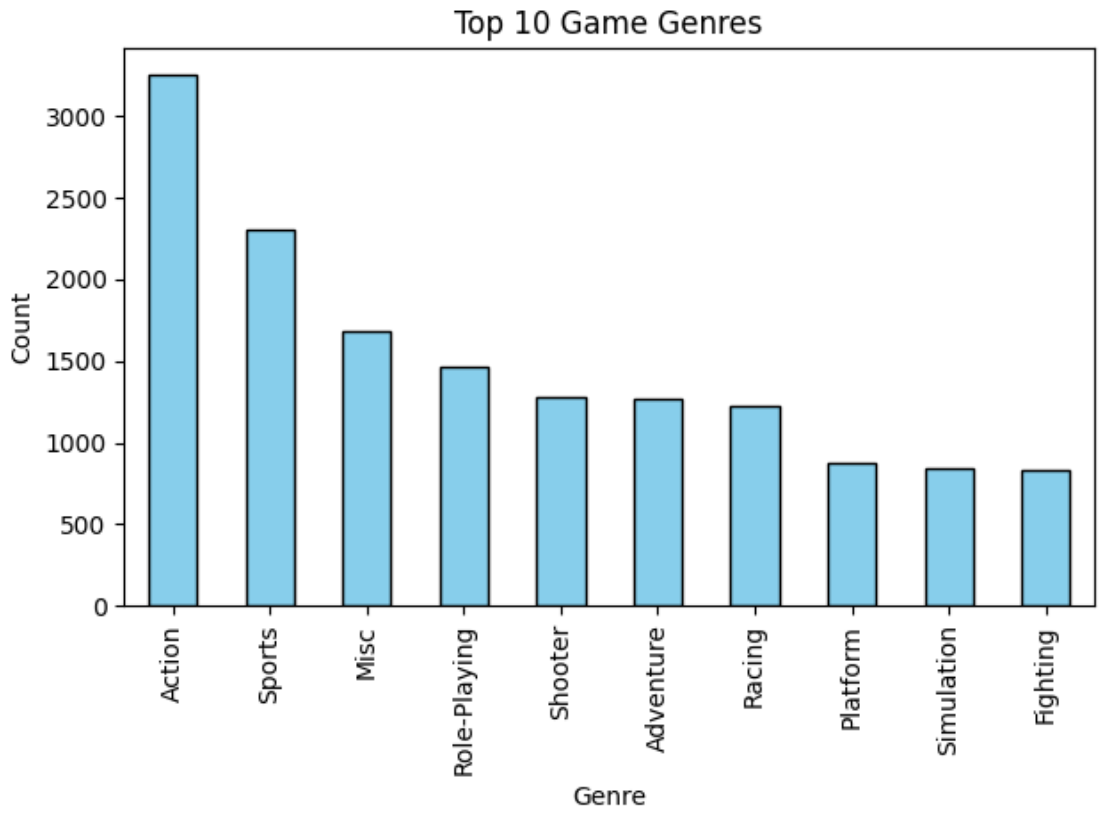
- Checking for Missing Values

```
print("Missing values in the dataset before cleaning")
print(df.isnull().sum())
```

```
Missing values in the dataset before cleaning
Name            0
Rank            0
Platform        0
Year            0
Genre           0
Publisher       0
NA_Sales        0
EU_Sales        0
JP_Sales        0
Other_Sales     0
Global_Sales    0
dtype: int64
```

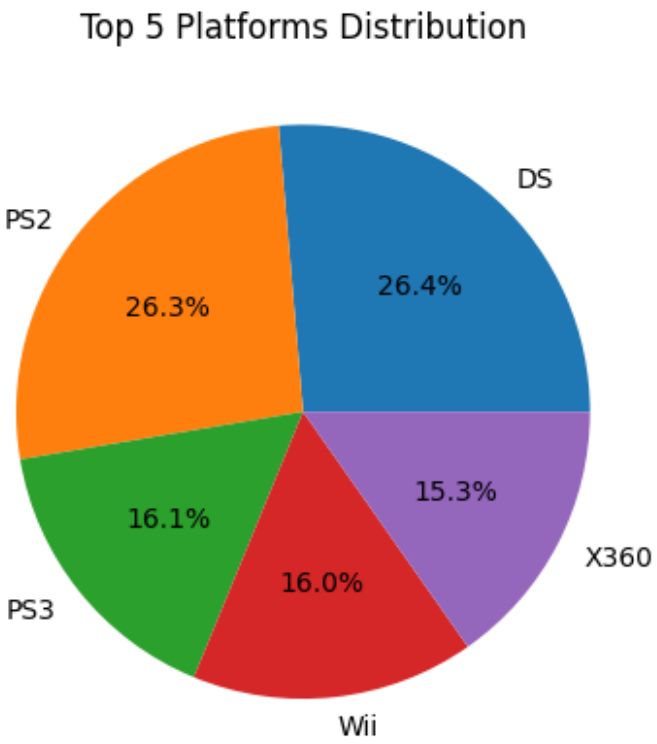
Visualisation using Bar Graph of top 10 game genres

```
import matplotlib.pyplot as plt
import seaborn as sns
df['Genre'].value_counts().head(10).plot(kind='bar',
color='skyblue',edgecolor='black')
plt.title("Top 10 Game Genres")
plt.xlabel("Genre")
plt.ylabel("Count")
plt.xticks()
plt.tight_layout()
plt.show()
```



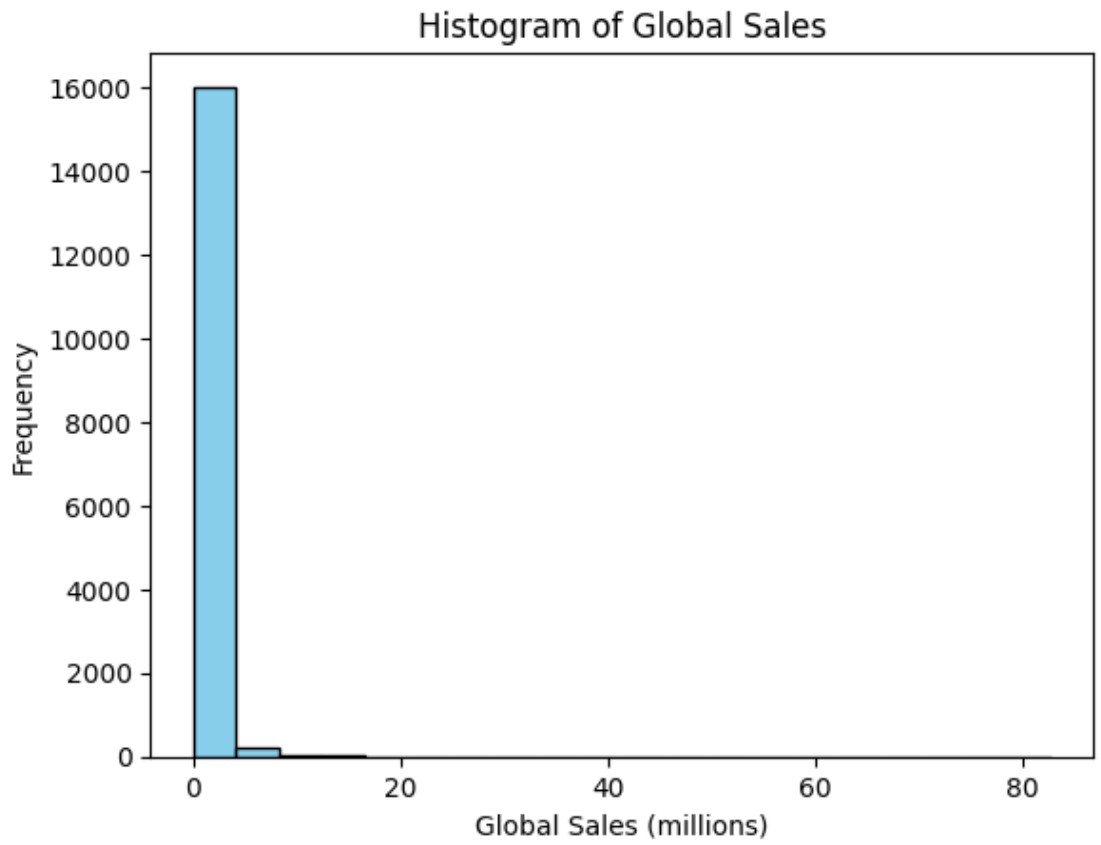
Using Pie Chart to depict top 5 Platforms Distributions

```
df['Platform'].value_counts().head(5).plot(kind='pie', autopct='%1.1f%%')
plt.title("Top 5 Platforms Distribution")
plt.ylabel('')
plt.show()
```



Using Histogram to depict Global sales

```
df['Global_Sales'].plot(kind='hist', bins=20, color='skyblue',
edgecolor='black')
plt.title("Histogram of Global Sales")
plt.xlabel("Global Sales (millions)")
plt.show()
```



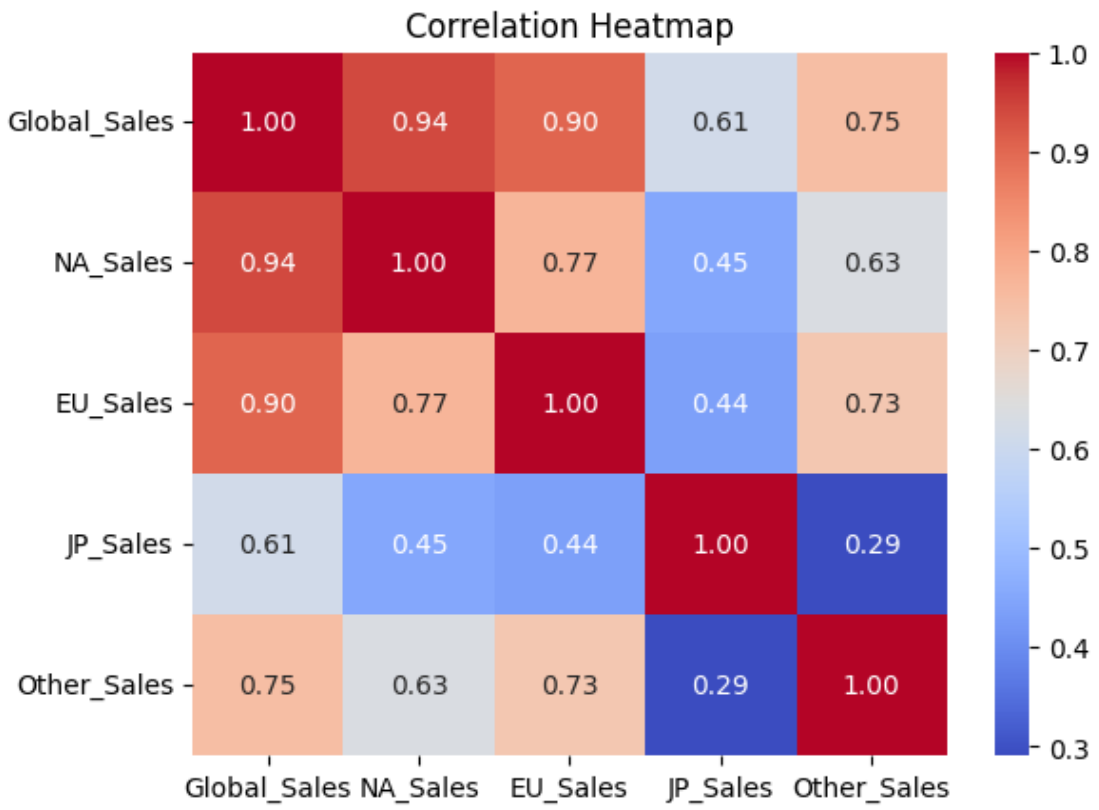
Correlation Matrix

```
corr_matrix = df[['Global_Sales',
'NA_Sales','EU_Sales','JP_Sales','Other_Sales']].corr()
print("Correlation Matrix:\n", corr_matrix)
```

```
Correlation Matrix:
              Global_Sales  NA_Sales  EU_Sales  JP_Sales  Other_Sales
Global_Sales      1.000000  0.941269  0.903264  0.612774     0.747964
NA_Sales          0.941269  1.000000  0.768923  0.451283     0.634518
EU_Sales          0.903264  0.768923  1.000000  0.436379     0.726256
JP_Sales          0.612774  0.451283  0.436379  1.000000     0.290559
Other_Sales       0.747964  0.634518  0.726256  0.290559     1.000000
```

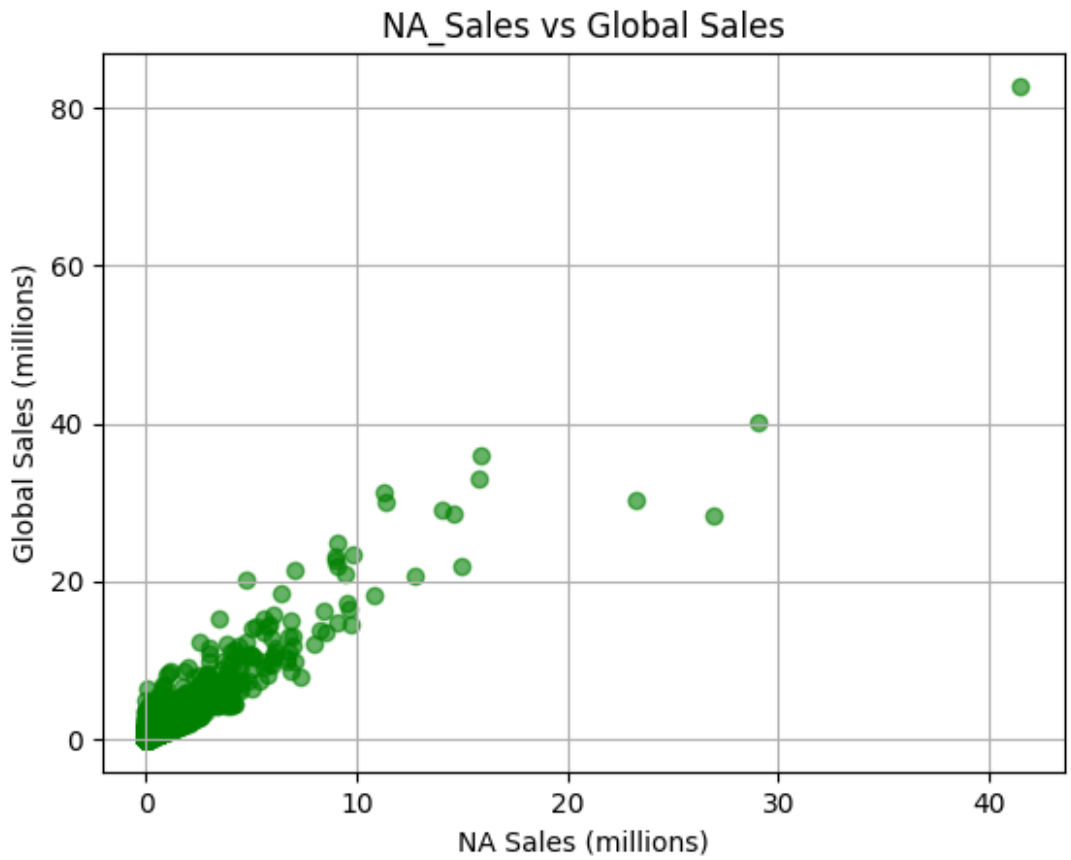
Correlation Heatmap

```
sns.heatmap(corr_matrix, annot=True, cmap='coolwarm', fmt=".2f")
plt.title("Correlation Heatmap")
plt.show()
```



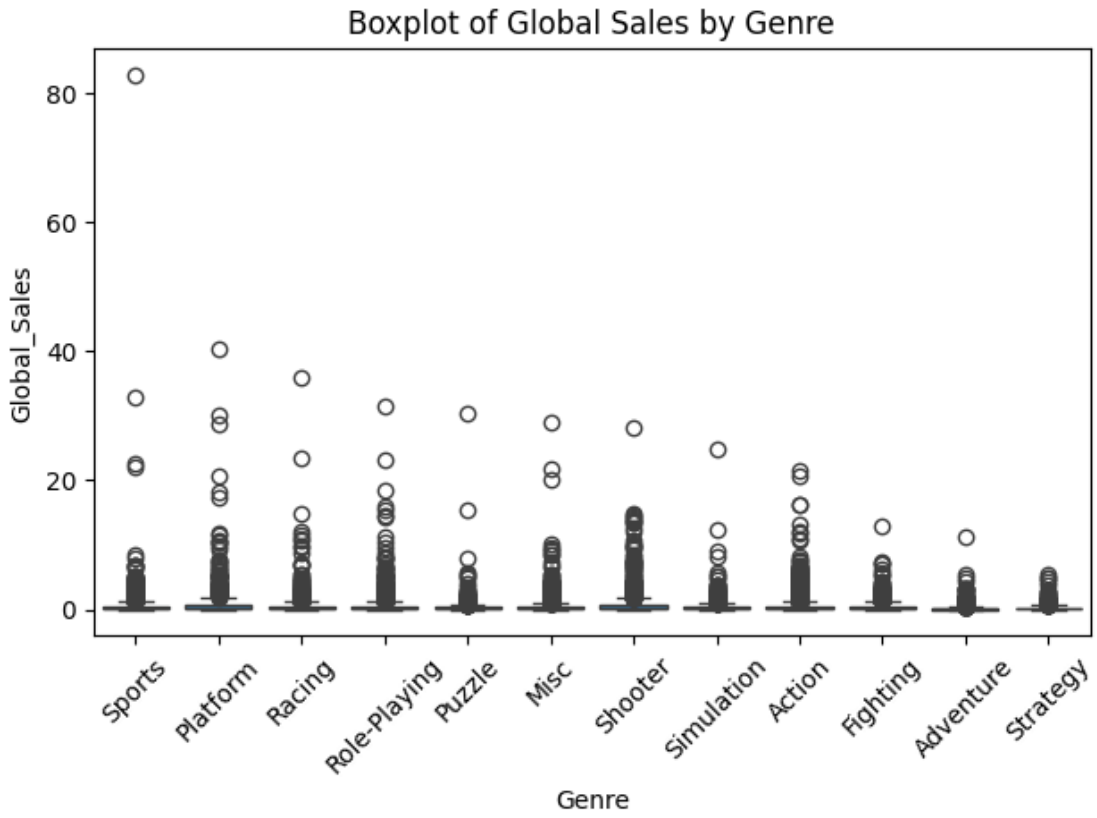
Using Scatter Plot to depict NA_Sales vs Global Sales

```
plt.scatter(df['NA_Sales'], df['Global_Sales'], alpha=0.6, c='green')
plt.title("NA_Sales vs Global Sales")
plt.xlabel("NA Sales (millions)")
plt.ylabel("Global Sales (millions)")
plt.grid(True)
plt.show()
```



Using Box Plot to depict Global Sales by Genre

```
sns.boxplot(x='Genre', y='Global_Sales', data=df)
plt.title("Boxplot of Global Sales by Genre")
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```



Using Line Plot to depict the change in sales over the period of years

```
df['Year'] = pd.to_numeric(df['Year'], errors='coerce')
ts = df.groupby('Year')['Global_Sales'].sum()
ts.plot()
plt.title("Global Sales Over Years")
plt.xlabel("Year")
plt.ylabel("Global Sales (millions)")
plt.grid(True)
plt.show()
```

Performing Feature Engineering & Selection

```
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler

# Drop 'Name' and 'Rank'
df_model = df_clean.drop(columns=['Name', 'Rank'])

# One-hot encode categorical variables
df_model = pd.get_dummies(df_model, columns=['Platform', 'Genre',
'Publisher'], drop_first=True)

# Features and target
X = df_model.drop(columns='Global_Sales')
y = df_model['Global_Sales']

# Split into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
random_state=42)

# Optional: Scale features (for some models like LinearRegression)
scaler = StandardScaler()
```

```
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)
```

Training a regression model

```
from sklearn.ensemble import RandomForestRegressor

# Initialize and train the model
rf_model = RandomForestRegressor(n_estimators=100, random_state=42)
rf_model.fit(X_train_scaled, y_train)
```

```
RandomForestRegressor(random_state=42)
```

Evaluating the model

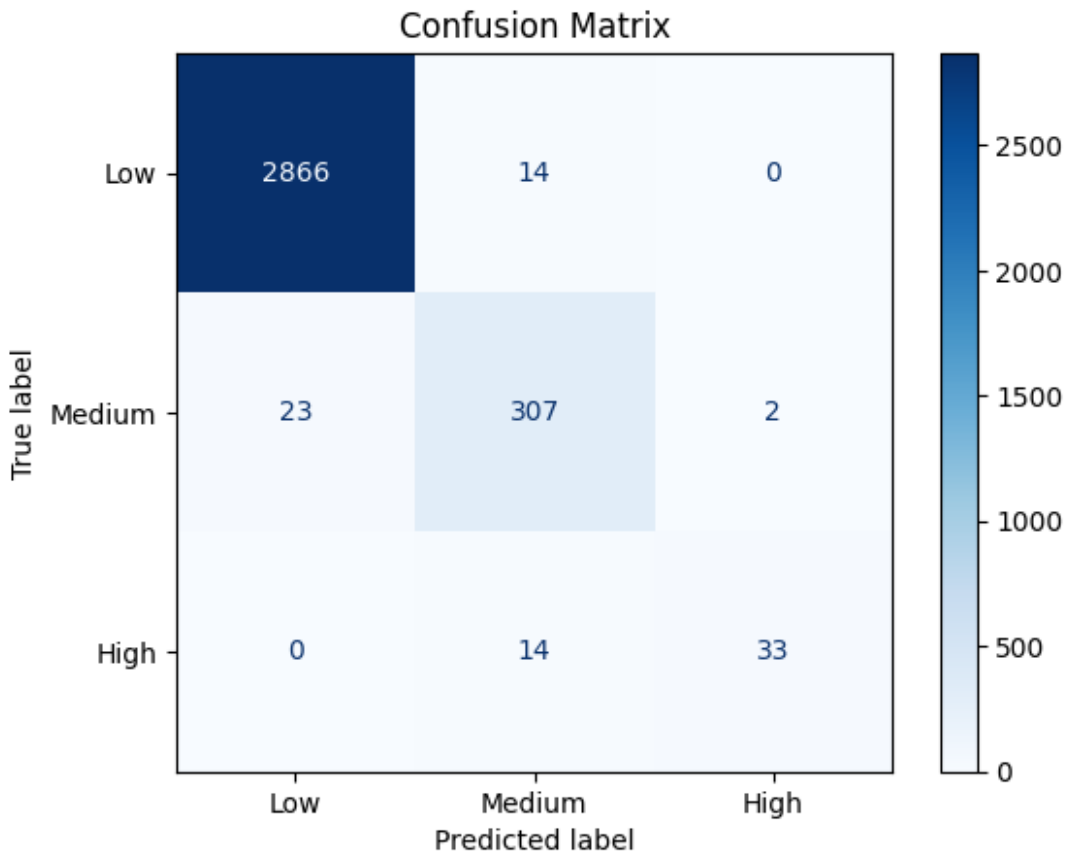
```
from sklearn.metrics import mean_absolute_error, mean_squared_error, r2_score
import numpy as np

# Predictions
y_pred = rf_model.predict(X_test_scaled)

# Evaluation metrics
r2 = r2_score(y_test, y_pred)
mae = mean_absolute_error(y_test, y_pred)
rmse = np.sqrt(mean_squared_error(y_test, y_pred))

print(f"R² Score: {r2:.4f}")
print(f"Mean Absolute Error: {mae:.4f}")
print(f"Root Mean Squared Error: {rmse:.4f}")
```

```
R² Score: 0.8186
Mean Absolute Error: 0.0431
Root Mean Squared Error: 0.8809
```

Constructing a Confusion Matrix

```
# Binning Global Sales into categories: Low, Medium, High
df_clean['Sales_Category'] = pd.cut(df_clean['Global_Sales'],
                                    bins=[-1, 1, 5,
df_clean['Global_Sales'].max()],
                                    labels=['Low', 'Medium', 'High'])

from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import confusion_matrix, ConfusionMatrixDisplay
import matplotlib.pyplot as plt

# Step 1: Create classification labels
df_clean['Sales_Category'] = pd.cut(df_clean['Global_Sales'],
                                    bins=[-1, 1, 5,
df_clean['Global_Sales'].max()],
                                    labels=['Low', 'Medium', 'High'])

# Step 2: Feature engineering
```

```
df_class = df_clean.drop(columns=['Name', 'Rank', 'Global_Sales'])  # drop
original target
df_class = pd.get_dummies(df_class, columns=['Platform', 'Genre',
'Publisher'], drop_first=True)

# Features and target
X = df_class.drop(columns='Sales_Category')
y = df_class['Sales_Category']

# Train-test split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
random_state=42)

# Scale
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)

# Train classifier
clf = RandomForestClassifier(n_estimators=100, random_state=42)
clf.fit(X_train_scaled, y_train)

# Predict and evaluate
y_pred = clf.predict(X_test_scaled)
cm = confusion_matrix(y_test, y_pred, labels=['Low', 'Medium', 'High'])

# Display confusion matrix
disp = ConfusionMatrixDisplay(confusion_matrix=cm, display_labels=['Low',
'Medium', 'High'])
disp.plot(cmap='Blues')
plt.title("Confusion Matrix")
plt.show()
```



Conclusion

- The final Random Forest regression model gave a strong performance with high R^2 and low RMSE, indicating good predictive power.
- Exploratory analysis revealed that North America dominates the gaming market, and genres like Action and Sports are most common.
- The project highlights how machine learning and data analysis can extract meaningful insights from raw industry data.
- With further tuning and external data (e.g., marketing budgets, online reviews), the model could be improved further