

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | | | |
|  | | Video Games Sales | | | | |  | |
|  |  | | | | | | |  |
|  | | | |  |  | | | |
|  | | | | Talha Tajammal |  | | | |
|  | | | | June 30, 2024—AI LAB Project—Hafiz Abdul Rehman |  | | | |
|  | | |  | | |  | | |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | | |  |  | | |  |
|  | INTRODUCTION | | | | | | |  |
|  |  | | |  |  | | |  |
|  |  | |  | | |  | |  |
|  |  |  | The objective of this project is to analyze the video game sales data and build machine learning models to predict the success of video games based on regional sales data. The dataset used in this project is vgsales.csv, which contains information about video game sales across different regions and platforms. | | |  |  |  |
|  | | | | |
|  |  |  |  |
|  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Decorative |  |  |  |  |

# Explanation

#### Dataset Overview:

The dataset consists of the following columns:

* Name: Name of the video game
* Platform: Platform of the video game release (e.g., Wii, NES, GB)
* Year: Year of release
* Genre: Genre of the video game (e.g., Sports, Platform, Racing)
* Publisher: Publisher of the video game
* NA\_Sales: Sales in North America (in millions)
* EU\_Sales: Sales in Europe (in millions)
* JP\_Sales: Sales in Japan (in millions)
* Other\_Sales: Sales in other regions (in millions)
* Global\_Sales: Total global sales (in millions)

#### Data Cleaning:

We dropped rows that contain null values to ensure the quality and integrity of our analysis.

#### Basic Information:

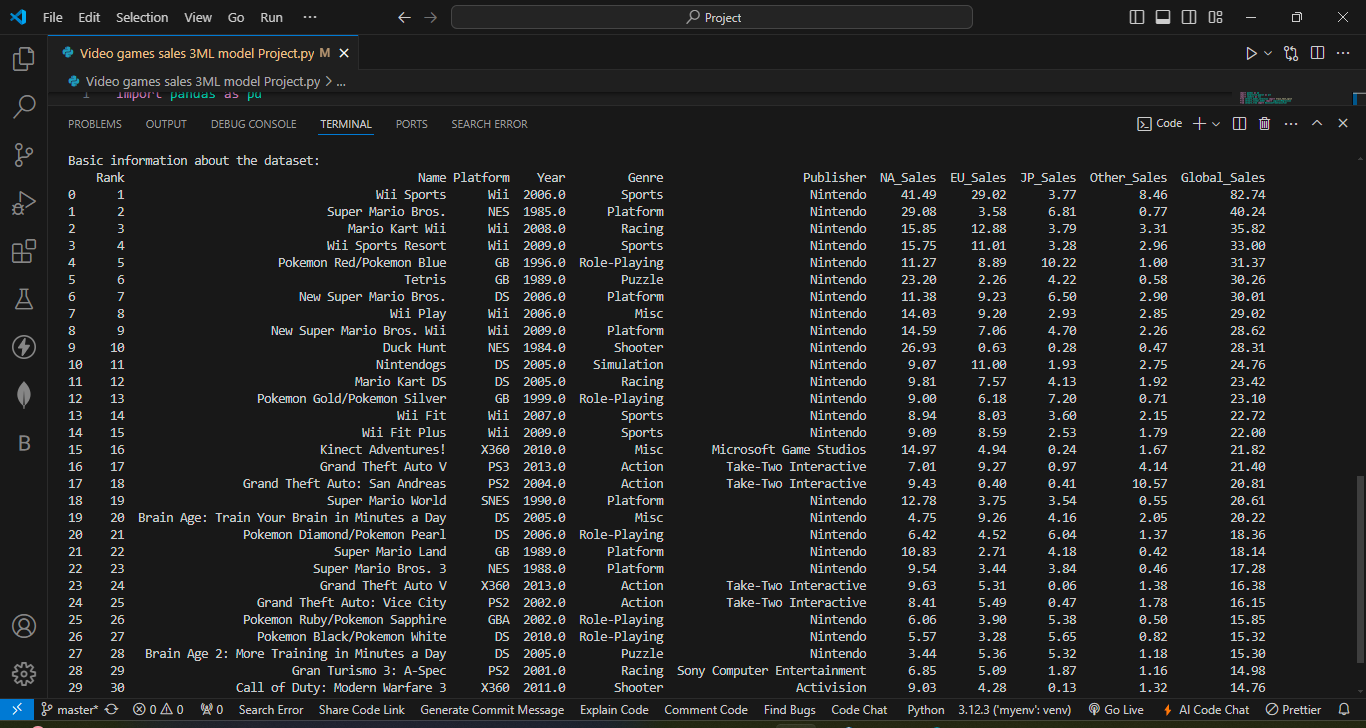
print("Basic information about the dataset:")

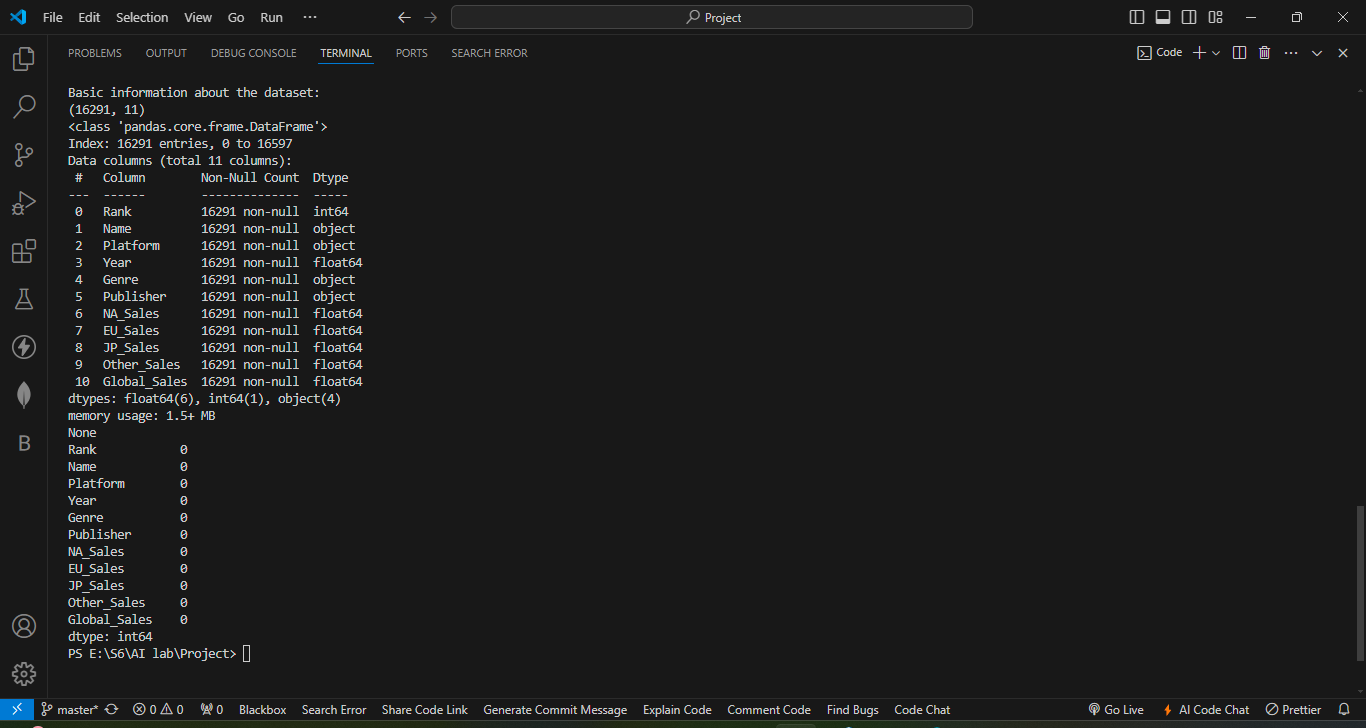
print(df.head(30))

print(df.shape)

print(df.info())

print(df.isnull().sum())





### Visualizations:

#### Number of Games Launched on Each Platform:

plt.figure(figsize=(10, 6))

sns.countplot(x='Platform', data=df)

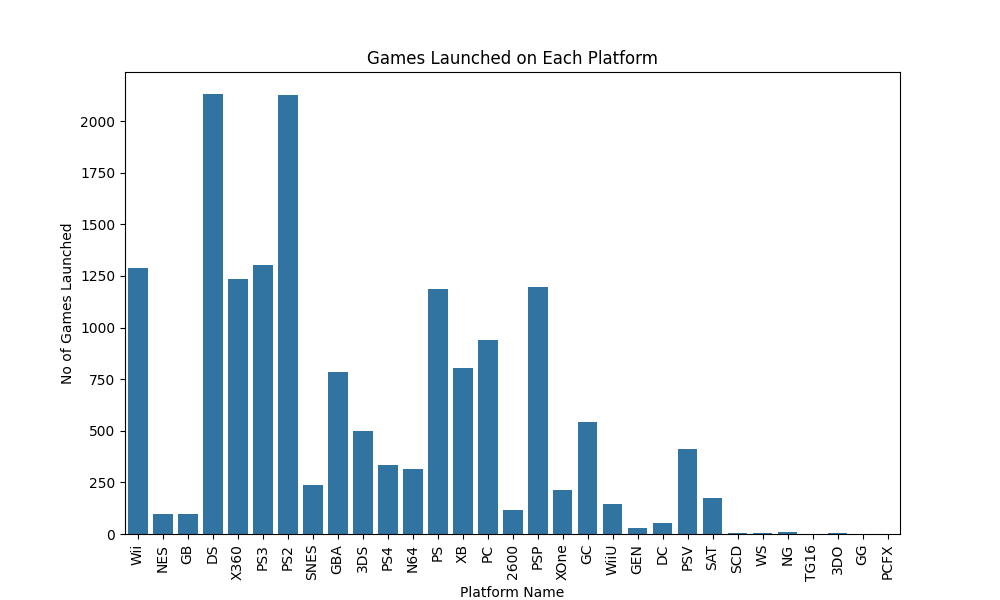
plt.xticks(rotation=90)

plt.ylabel('No of Games Launched')

plt.title('Games Launched on Each Platform')

plt.xlabel('Platform Name')

plt.show()



#### Number of Games Launched Per Year:

plt.figure(figsize=(10, 6))

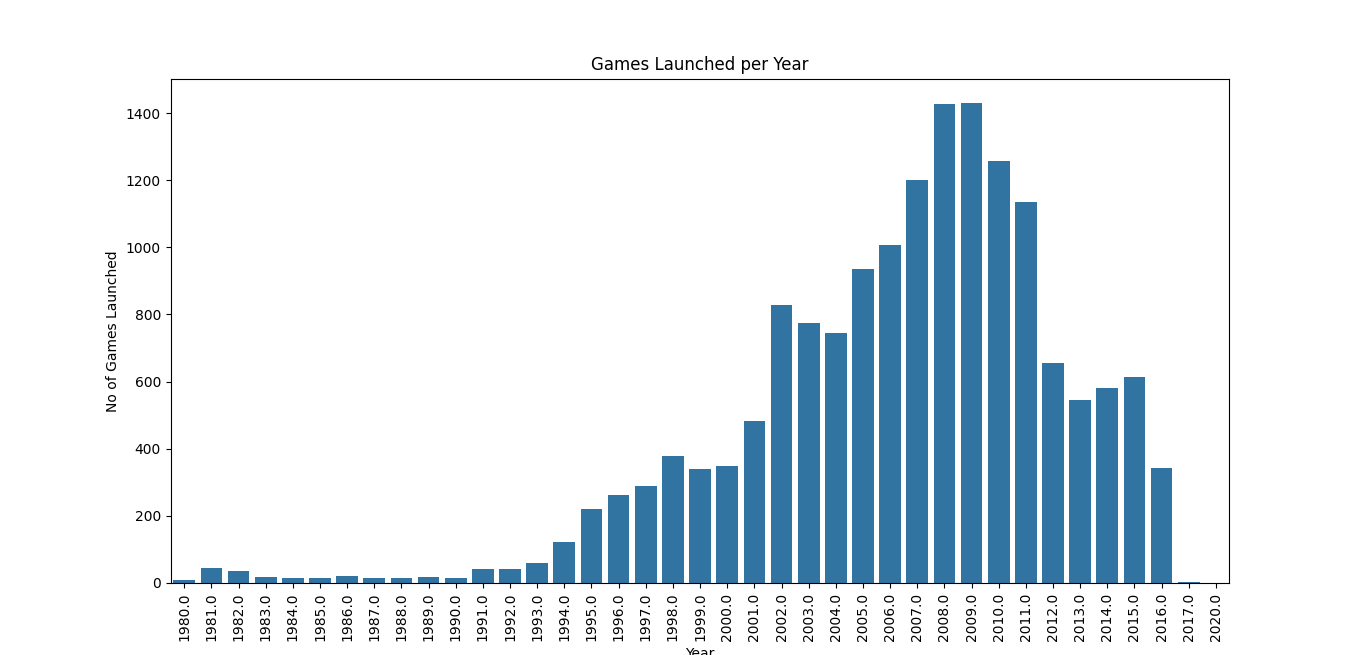
sns.countplot(x='Year', data=df)

plt.xticks(rotation=90)

plt.ylabel('No of Games Launched')

plt.xlabel('Year')

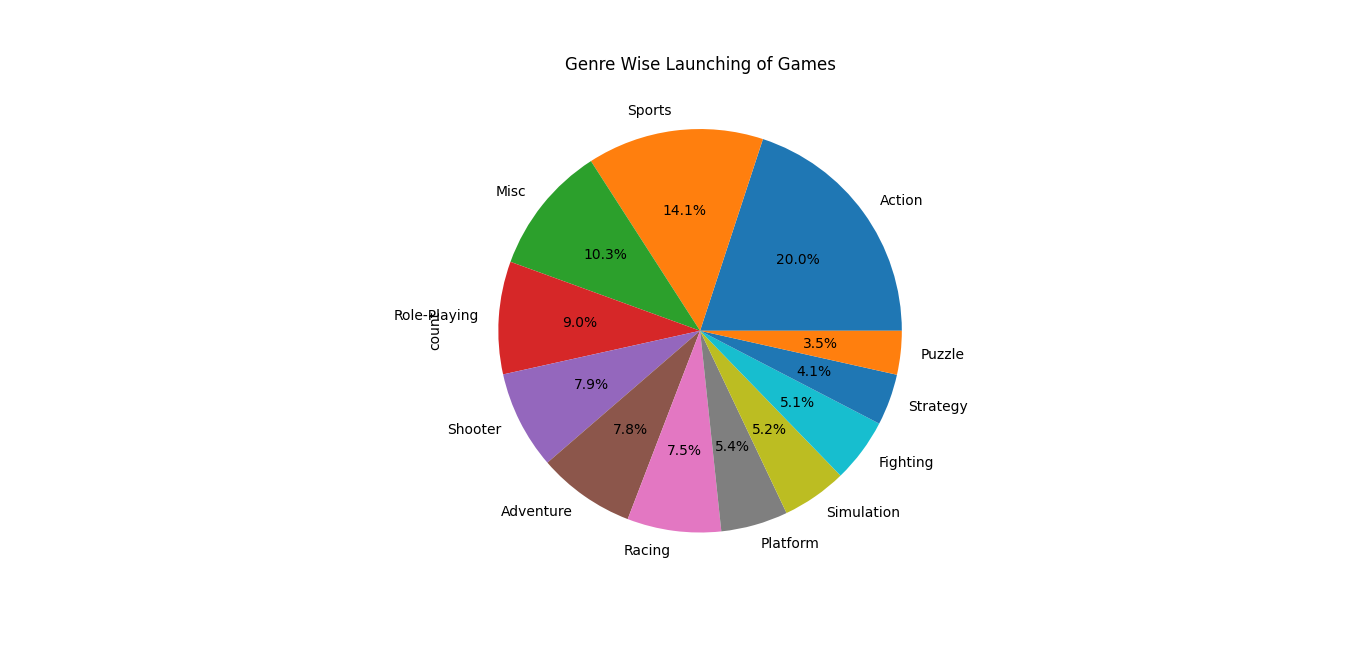
plt.title('Games Launched per Year')

plt.show()

#### Genre-wise Launching of Games as a Pie Chart:

df['Genre'].value\_counts().plot(kind='pie', autopct='%1.1f%%', figsize=(8, 8))

plt.title("Genre Wise Launching of Games")

plt.show()

#### Top 10 Publishers by Market Share:

publishers = (df['Publisher'].value\_counts() / len(df)) \* 100

print("Top 10 publishers by market share:")

print(publishers.head(10))

publishers.head(10).plot(kind='bar', figsize=(10, 6))

plt.title("Top 10 Publishers by Market Share")

plt.ylabel('Percentage of Market Share')

plt.show()

#### Sales Totals in Different Regions:

print("\nSales totals in different regions:")

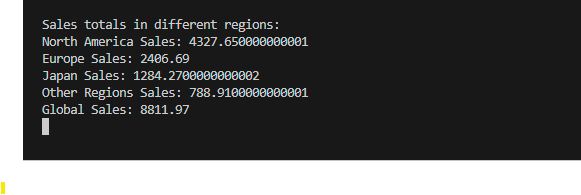
print("North America Sales:", df['NA\_Sales'].sum())

print("Europe Sales:", df['EU\_Sales'].sum())

print("Japan Sales:", df['JP\_Sales'].sum())

print("Other Regions Sales:", df['Other\_Sales'].sum())

print("Global Sales:", df['Global\_Sales'].sum())



#### Genre-wise Sales in Different Regions:

def plot\_genre\_sales(region, title, color):

    genre\_sales = df.groupby('Genre')[region].sum().sort\_values(ascending=False)

    plt.figure(figsize=(12, 6))

    genre\_sales.plot(kind='bar', color=color)

    plt.title(title)

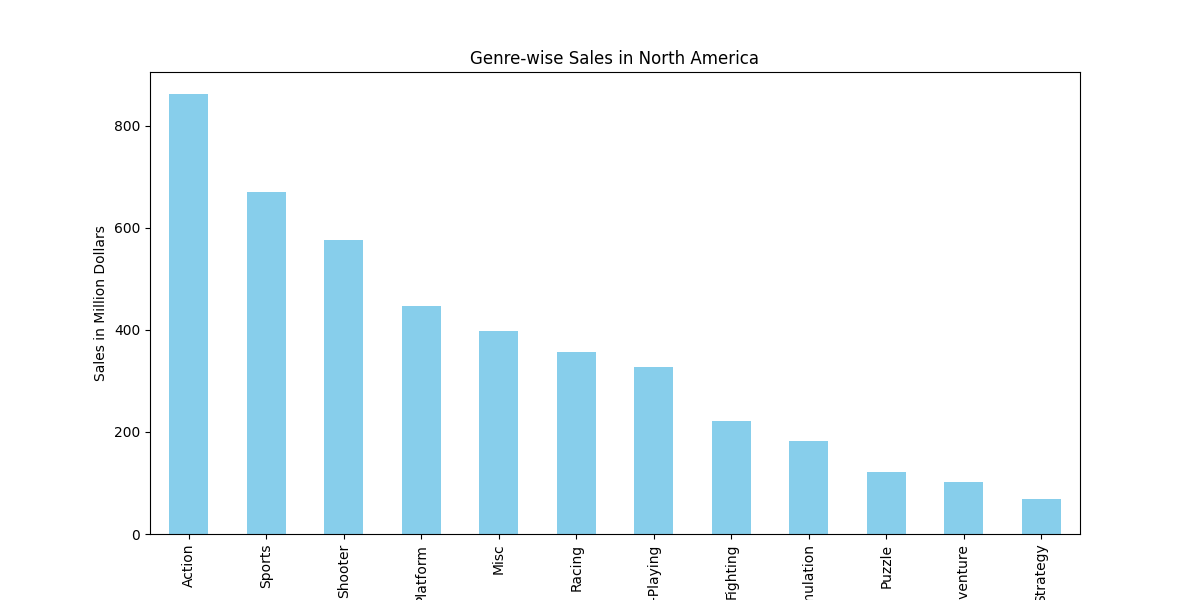
    plt.ylabel("Sales in Million Dollars")

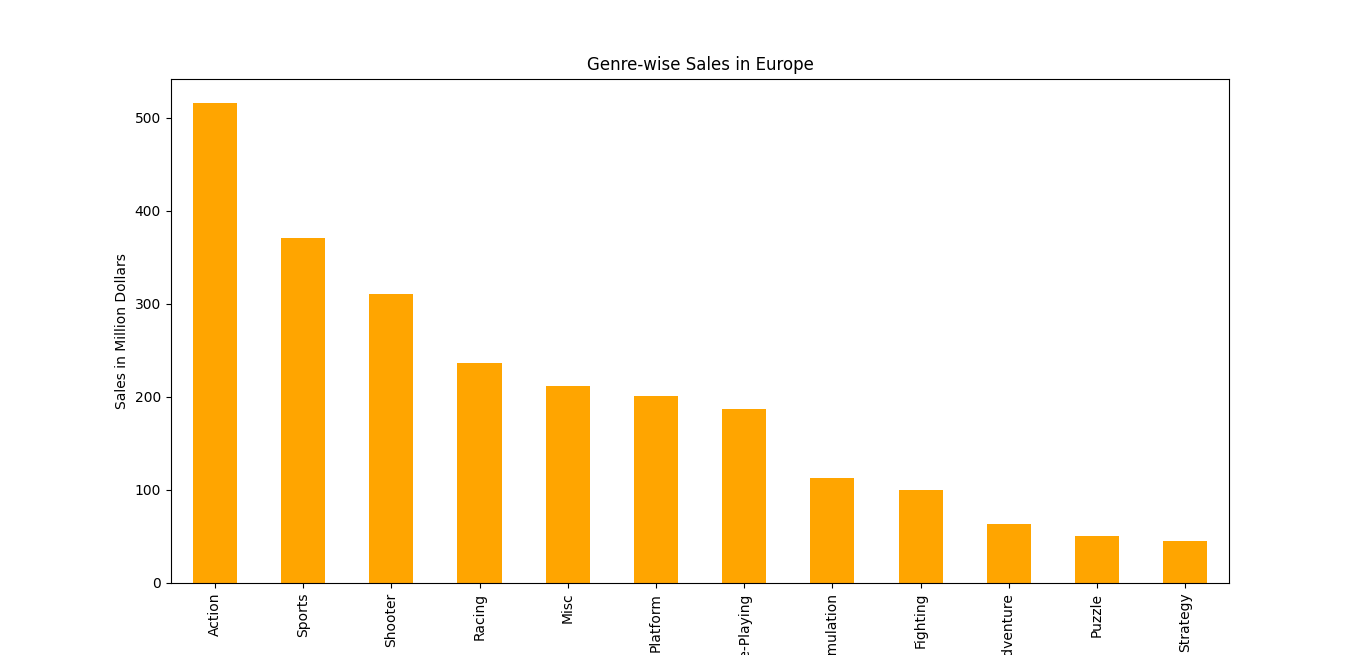
    plt.xlabel("Genre")

    plt.show()  
  
plot\_genre\_sales('NA\_Sales', "Genre-wise Sales in North America", 'skyblue')

plot\_genre\_sales('EU\_Sales', "Genre-wise Sales in Europe", 'orange')

plot\_genre\_sales('JP\_Sales', "Genre-wise Sales in Japan", 'skyblue')







#### Year-wise Sales in Different Regions (2001 - 2016):

def plot\_yearwise\_sales(region, title, color):

    yearwise\_sales = df.groupby('Year')[region].sum().loc[2001:2016]

    plt.figure(figsize=(12, 6))

    yearwise\_sales.plot(kind='bar', color=color)

    for i, v in enumerate(yearwise\_sales):

        plt.text(i, v + 0.5, f'{v:.2f}', ha='center')

    plt.title(title)

    plt.xlabel('Year')

    plt.ylabel('Total Sales (in millions)')

    plt.xticks(rotation=45)

    plt.grid(True, axis='y')

    plt.tight\_layout()

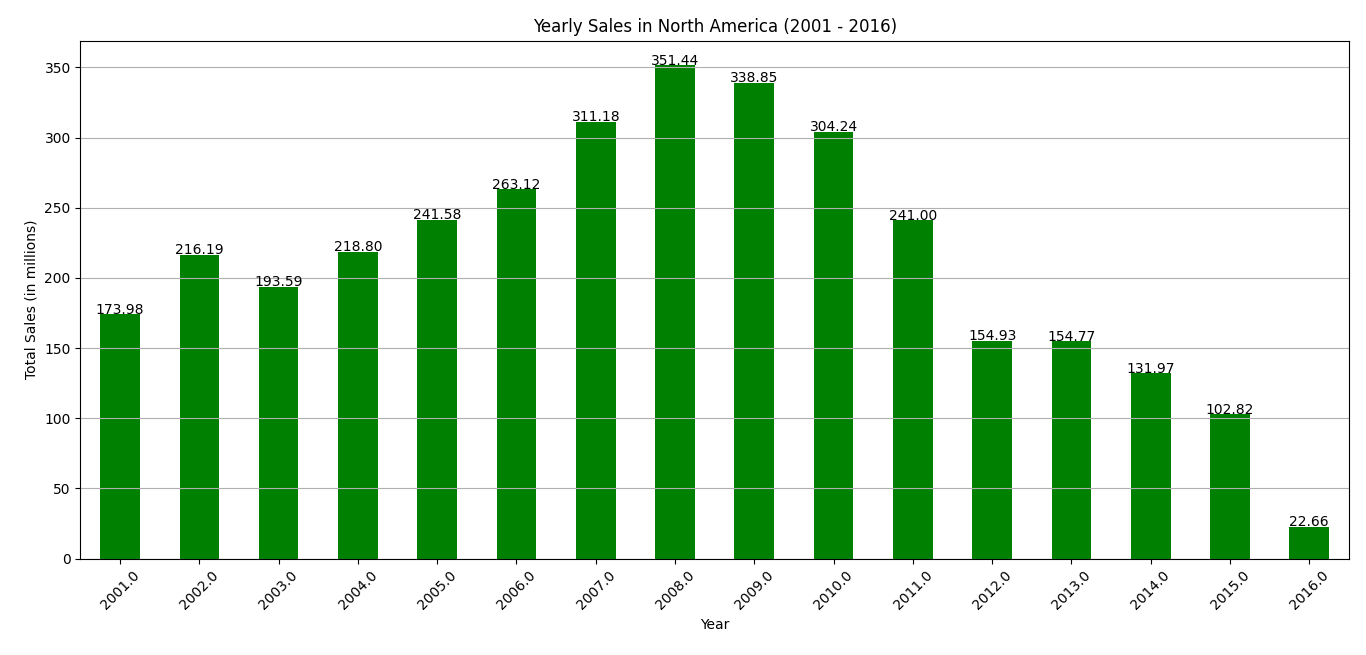
    plt.show()

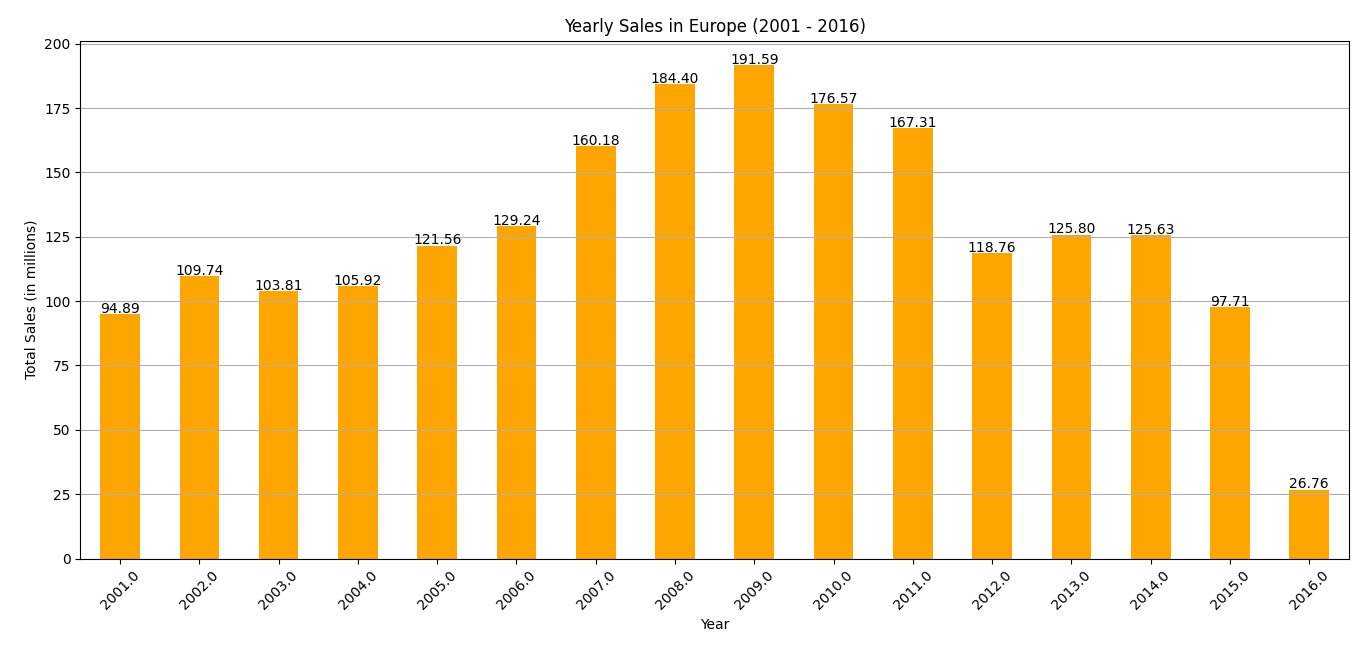
plot\_yearwise\_sales('NA\_Sales', 'Yearly Sales in North America (2001 - 2016)', 'green')

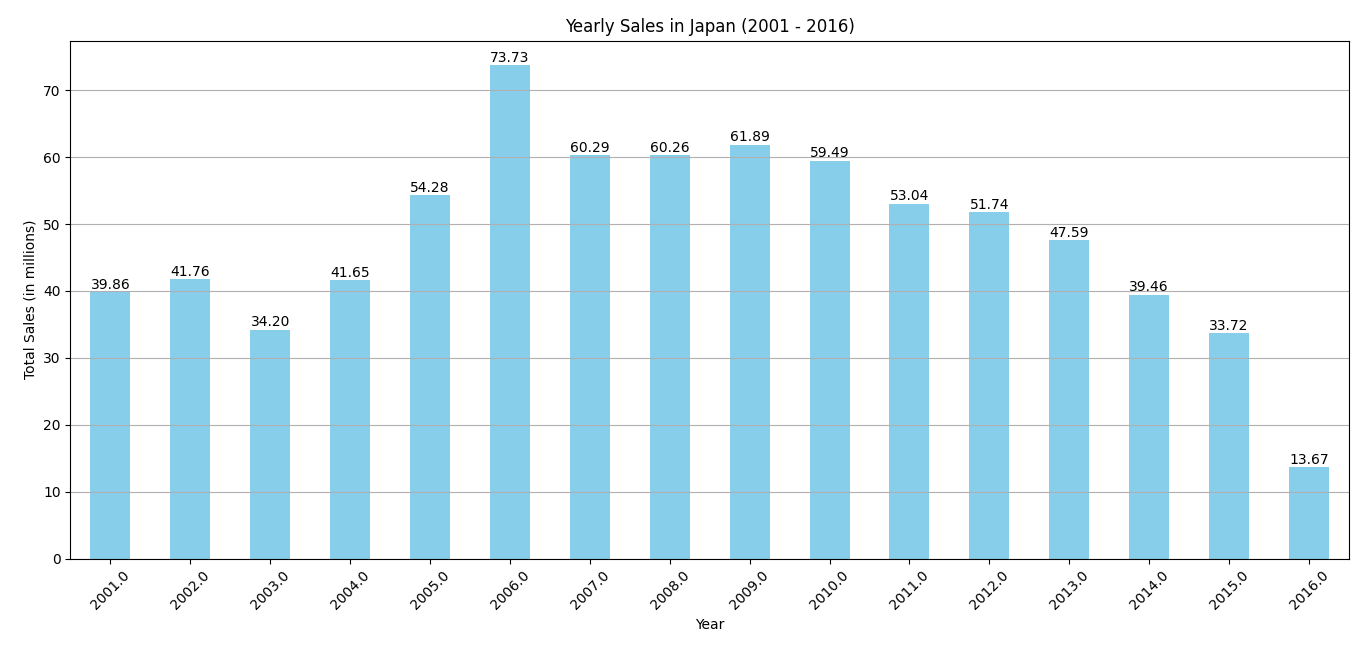
plot\_yearwise\_sales('EU\_Sales', 'Yearly Sales in Europe (2001 - 2016)', 'orange')

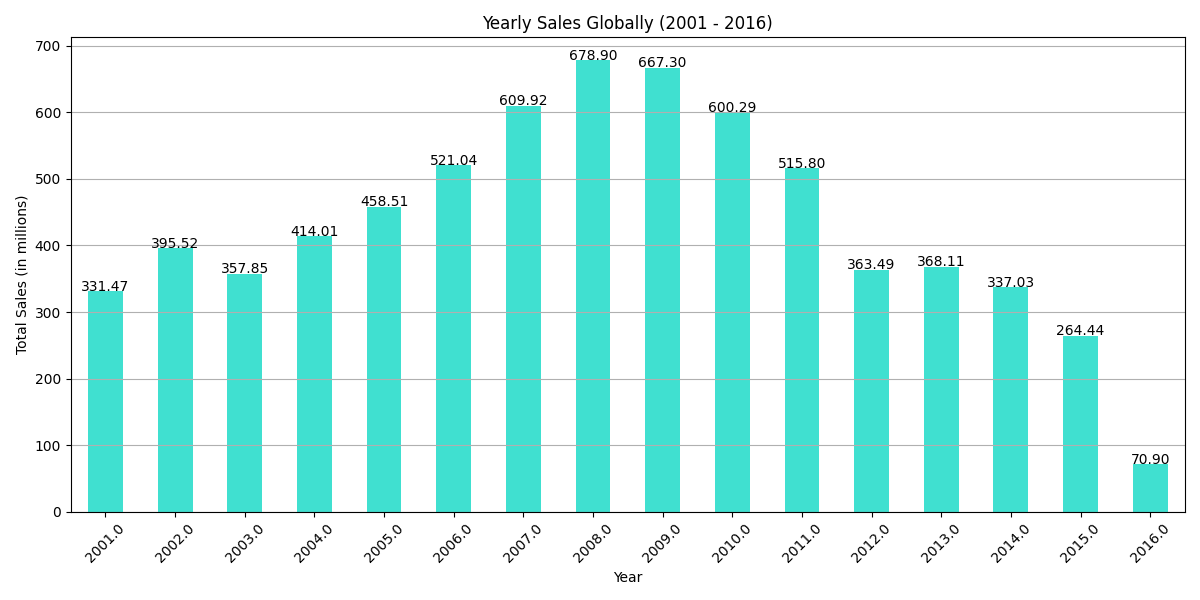
plot\_yearwise\_sales('JP\_Sales', 'Yearly Sales in Japan (2001 - 2016)', 'skyblue')

plot\_yearwise\_sales('Global\_Sales', 'Yearly Sales Globally (2001 - 2016)', 'turquoise')









#### Top 10 Games by Sales in Different Regions:

def plot\_top\_10\_sales(region, title, color):

    top\_10\_sales = df.groupby('Name')[region].sum().sort\_values(ascending=False).head(10)

    plt.figure(figsize=(12, 6))

    top\_10\_sales.plot(kind='barh', color=color)

    plt.title(title)

    plt.xlabel('Total Sales (in millions)')

    plt.ylabel('Games')

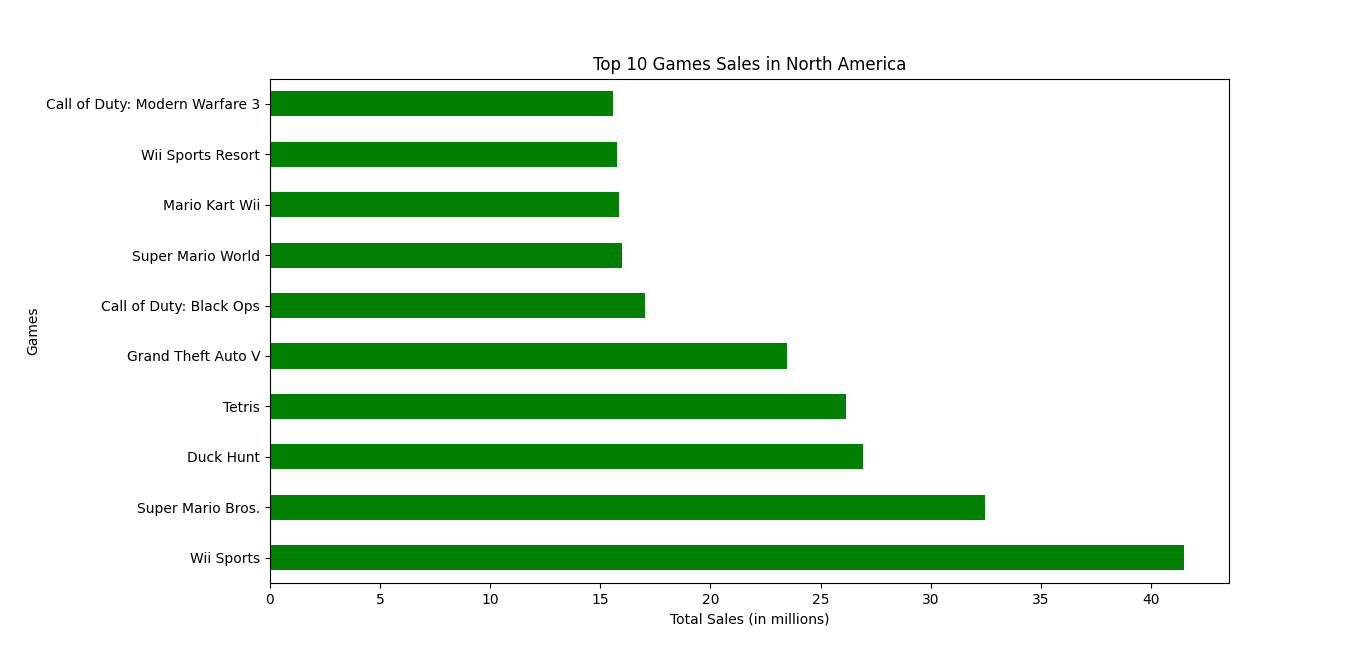
    plt.show()

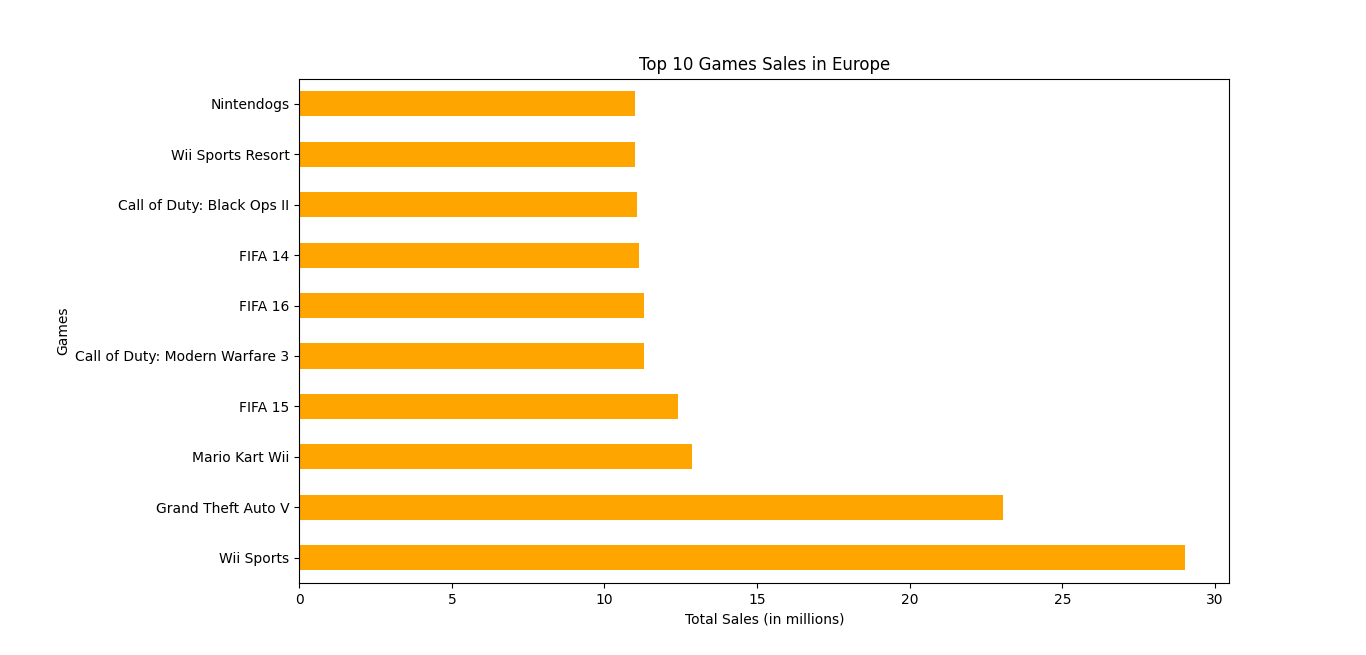
plot\_top\_10\_sales('NA\_Sales', 'Top 10 Games Sales in North America', 'green')

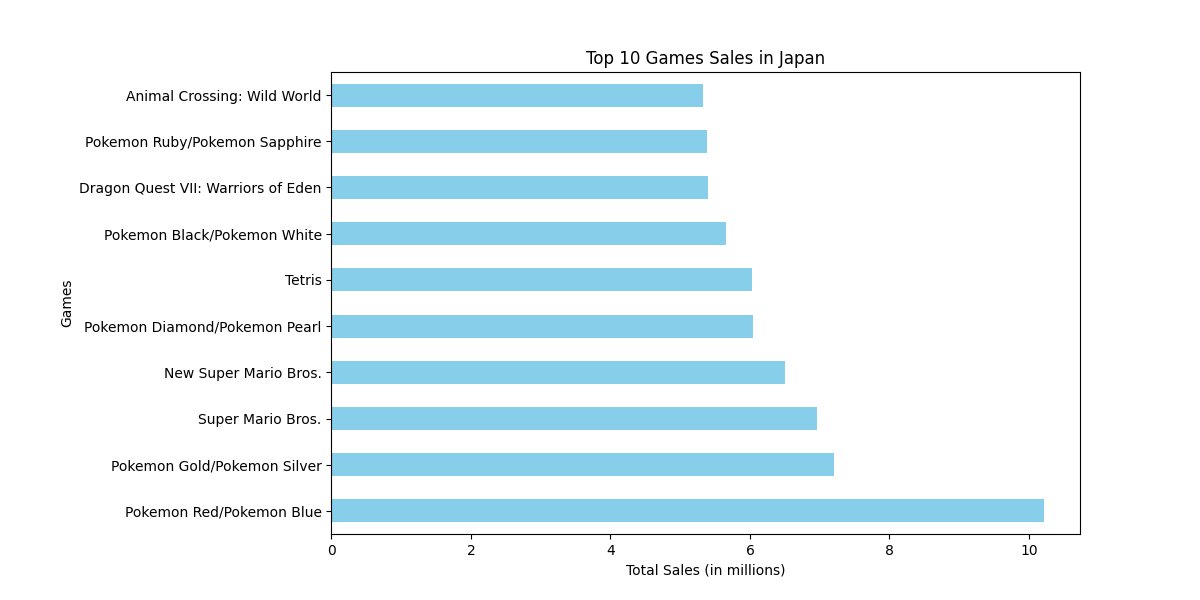
plot\_top\_10\_sales('EU\_Sales', 'Top 10 Games Sales in Europe', 'orange')

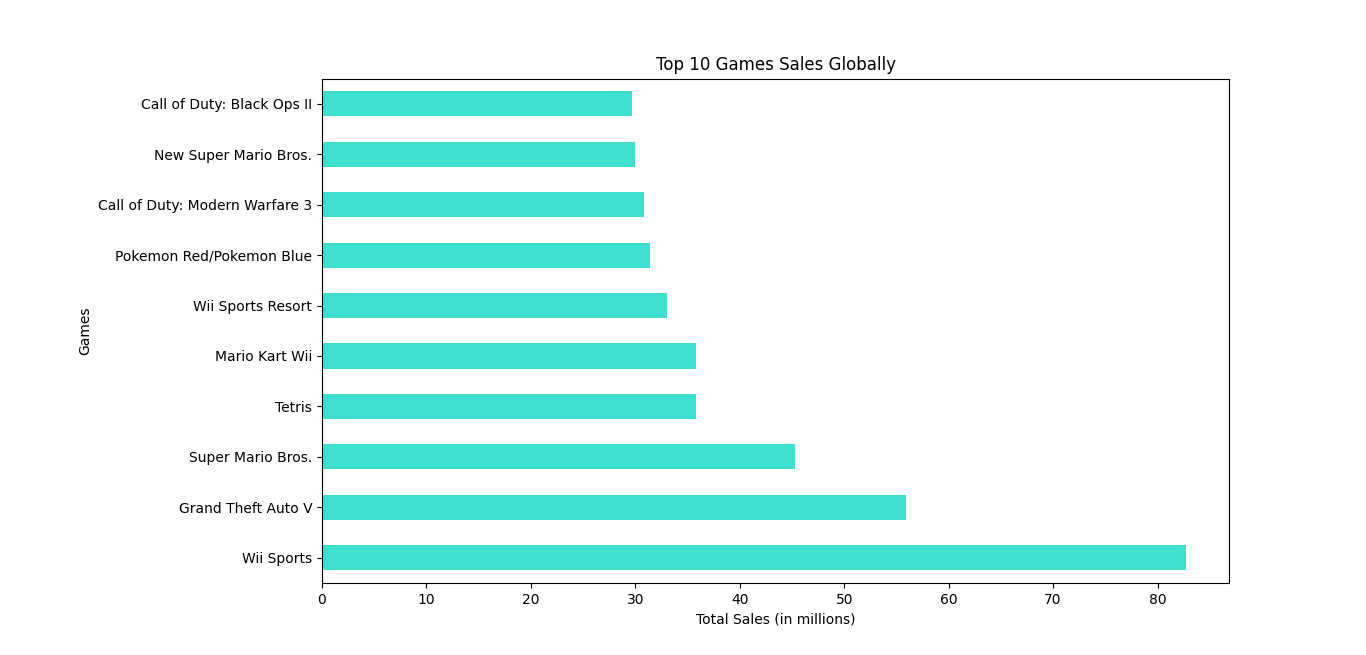
plot\_top\_10\_sales('JP\_Sales', 'Top 10 Games Sales in Japan', 'skyblue')

plot\_top\_10\_sales('Global\_Sales', 'Top 10 Games Sales Globally', 'turquoise')









### Machine Learning Models

Define Features and Target Variable:

X = df[['NA\_Sales', 'EU\_Sales', 'JP\_Sales', 'Other\_Sales']] # Features variables

y = df['Global\_Sales'] > 1

# Train-Test Split:

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

#### Logistic Regression Model:

logreg = LogisticRegression(max\_iter=1000)

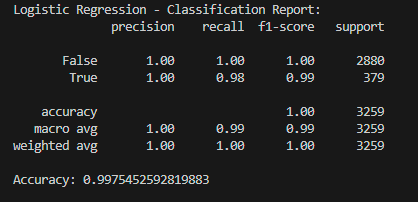
logreg.fit(X\_train, y\_train)

y\_pred\_logreg = logreg.predict(X\_test)

print("Logistic Regression - Classification Report:")

print(classification\_report(y\_test, y\_pred\_logreg))

print("Accuracy:", accuracy\_score(y\_test, y\_pred\_logreg))



#### Decision Tree Classifier Model:

dt\_classifier = DecisionTreeClassifier(random\_state=42)

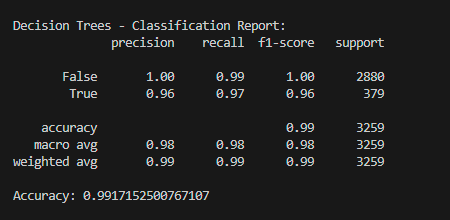
dt\_classifier.fit(X\_train, y\_train)

y\_pred\_dt = dt\_classifier.predict(X\_test)

print("Decision Trees - Classification Report:")

print(classification\_report(y\_test, y\_pred\_dt))

print("Accuracy:", accuracy\_score(y\_test, y\_pred\_dt))



#### Random Forest Classifier Model:

rf\_classifier = RandomForestClassifier(n\_estimators=100, random\_state=42)

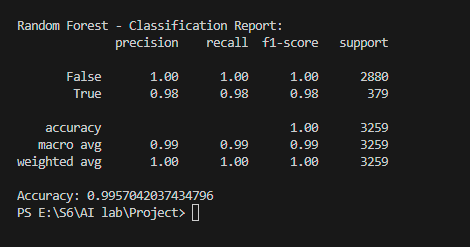
rf\_classifier.fit(X\_train, y\_train)

y\_pred\_rf = rf\_classifier.predict(X\_test)

print("Random Forest - Classification Report:")

print(classification\_report(y\_test, y\_pred\_rf))

print("Accuracy:", accuracy\_score(y\_test, y\_pred\_rf))



#### Model Evaluation:

* **Logistic Regression:**
  + Precision, recall, and F1-score are high, indicating good model performance.
  + Accuracy: 0.9975
* **Decision Tree:**
  + Performed well but slightly less accurate than logistic regression.
  + Accuracy: 0.9917
* **Random Forest:**
  + Performed very well with high precision and recall.
  + Accuracy: 0.9957

#### Conclusion:

The analysis and visualizations gave us a better understanding of the video game market, highlighting trends in game releases and sales across different regions and genres. The machine learning models we built using logistic regression, decision trees, and random forest classifiers showed high accuracy in predicting the success of video games based on regional sales data.

Both logistic regression and random forest classifiers performed exceptionally well, with accuracies above 99%, making them suitable models for this prediction task.