VIDEO GAME SALES PREDICTION

Prepared by :
Urlana Suresh Kumar

github.com/usk2003

Abstract

This project focuses on predicting global video game sales using machine learning algorithms. With the rapid growth of the gaming industry, understanding the factors that contribute to a game's success is crucial for developers, marketers, and business stakeholders. By analyzing a comprehensive dataset of video games, which includes features such as platform, genre, release year, and regional sales figures, the goal is to develop machine learning models capable of accurately forecasting global sales.

The dataset utilized in this project is sourced from Kaggle, containing detailed information about video games and their performance across different regions. Various regression algorithms, including K-Nearest Neighbors (KNN), Multiple Linear Regression (MLR), Decision Tree Regression (DTR), Random Forest Regression (RFR), and Gradient Boosting Regression (GBR), were employed to train the model. Hyperparameter tuning and cross-validation techniques were applied to enhance model performance.

The results reveal that the Gradient Boosting Regression model outperformed other models in terms of prediction accuracy, as measured by the R-squared score. This project provides valuable insights into the factors driving video game sales and can aid gaming industry stakeholders in making data driven decisions, from game development to marketing strategies, ensuring more effective resource allocation and better market forecasting.

Keywords: video game sales prediction, machine learning, regression models, global sales, Gradient Boosting, data analysis.

INDEX

S.NO	TOPIC	Page No.
01	Introduction	01-02
02	Data Set Description	03
03	Literature Review	04-05
04	Preprocessing Activities	06
05	Implementation of Project	07-15
06	Applications	16
07	Conclusions	17
08	References	18

INTRODUCTION

The **Video Game Sales Prediction** project focuses on leveraging machine learning techniques to forecast global sales of video games. By analyzing historical data that includes features such as platform, genre, release year, and regional sales, this project aims to uncover patterns and predict future sales. This predictive capability can assist stakeholders in the gaming industry—publishers, developers, and marketers—in making strategic decisions regarding resource allocation, game development, and targeted marketing campaigns.

Concept and Principles :

The project is grounded in machine learning principles, where various regression algorithms are used to build models capable of predicting continuous outcomes (sales figures). The following core concepts form the foundation of this project:

1. *Data Preprocessing*: Cleaning and transforming raw data to make it suitable for analysis, including handling missing values and encoding categorical features.
2. *Feature Selection*: Choosing the most relevant attributes, such as genre and regional sales, that significantly impact global sales predictions.
3. *Model Training and Evaluation*: Using regression algorithms like Linear Regression, Random Forest, and Gradient Boosting to develop models and evaluate their performance based on metrics such as Mean Squared Error (MSE) and R-squared scores.
4. *Hyperparameter Optimization*: Tuning the parameters of models to achieve the best performance.

Efficiency and Performance :

The project highlights the effectiveness of machine learning models in processing large datasets and uncovering insights. Among the models tested, Gradient Boosting Regression emerged as the most efficient, with a high R-squared score indicating its superior predictive accuracy. This efficiency is attributed to:

- *Feature Interaction Analysis*: Capturing intricate relationships between features.
- *Ensemble Learning Techniques*: Combining multiple weak learners to form a strong predictive model.
- *Scalability*: Handling a significant volume of video game sales data without substantial performance degradation.

Limitations:

1. *Data Dependency*: The accuracy of the predictions heavily depends on the quality and comprehensiveness of the dataset used. Missing or outdated data can skew the results.
2. *Bias in Historical Data*: Historical trends might not always reflect future consumer behavior or emerging trends in the gaming industry.
3. *Model Generalization*: While models perform well on the test dataset, their performance might vary when exposed to entirely new data with unseen patterns.
4. *Feature Simplification*: Some influential factors, such as marketing efforts, user reviews, and cultural trends, were not included due to the lack of relevant data.

The Video Game Sales Prediction project successfully demonstrates how machine learning can provide valuable insights into sales trends in the gaming industry. While Gradient Boosting Regression proved to be the most effective algorithm, the study also underscores the importance of high-quality data and the need for careful model evaluation. The findings from this project can empower stakeholders with data-driven strategies, although continuous refinement is necessary to adapt to the dynamic nature of the gaming market.

Dataset Description

The dataset used in this project is sourced from **Kaggle** and provides comprehensive information about video games, their attributes, and sales performance across different regions. Below are the key details about the dataset:

Dataset Features

The dataset consists of multiple columns, each representing a distinct aspect of a video game. The primary features are:

1. *Name:* The name of the video game.
2. *Platform:* The platform or console the game was released on (e.g., PS4, Xbox One, PC).
3. *Year:* The release year of the game.
4. *Genre:* The genre of the video game (e.g., Action, Sports, Puzzle).
5. *Publisher:* The publisher of the game.
6. *NA_Sales:* Sales figures (in millions) in North America.
7. *EU_Sales:* Sales figures (in millions) in Europe.
8. *JP_Sales:* Sales figures (in millions) in Japan.
9. *Other_Sales:* Sales figures (in millions) in the rest of the world.
10. *Global_Sales:* Total global sales (in millions).

Dataset Characteristics

- *Size:* The dataset contains thousands of entries, representing a broad spectrum of video games released over the years.
- *Data Type:* The dataset includes both categorical variables (e.g., Platform, Genre, Publisher) and numerical variables (e.g., Year, Sales figures).
- *Missing Values:* The dataset had missing values in some columns, which were handled during preprocessing.
- *Categorical Encoding:* Categorical features such as Platform and Genre were encoded into numerical representations using Label Encoding to facilitate machine learning model training.

Dataset Source

The dataset originates from Kaggle and is publicly available for research and analysis purposes. It offers a wealth of information suitable for understanding trends and patterns in the video gaming industry.

Relevance to the Project

The dataset's rich set of features enables detailed analysis and predictive modeling of global video game sales. By leveraging this data, the project aims to uncover insights into the factors that drive sales and identify the most effective machine learning model for accurate predictions.

Literature Review

1. Importance of Predictive Analytics in the Gaming Industry

Predictive analytics has become a critical tool in the gaming industry, enabling stakeholders to forecast market trends, optimize sales strategies, and tailor products to consumer preferences. By analyzing historical data, machine learning models offer a way to predict outcomes like sales, user retention, and market penetration with greater accuracy.

2. Prior Work on Video Game Sales Prediction

Several studies and projects have focused on predicting video game sales by leveraging machine learning techniques. Key contributions include:

- *Exploration of Sales Data:* Researchers have examined factors such as genre, platform, and regional sales trends to identify correlations and patterns. These studies underscore the significance of genre popularity and regional preferences.
- *Use of Regression Algorithms:* Studies comparing regression models (e.g., Linear Regression, Decision Trees) for sales prediction have shown that ensemble methods like Random Forest and Gradient Boosting often outperform simpler models.
- *Impact of Market Factors:* Literature also highlights the role of external factors such as marketing campaigns, economic conditions, and competition, emphasizing their influence on sales figures.

3. Machine Learning Approaches for Predictive Modeling

Recent advancements in machine learning have introduced robust models capable of handling large and complex datasets:

- Linear Regression and KNN: Often used as baseline models, these techniques provide interpretable results but may lack accuracy with non-linear relationships.
- Decision Trees and Random Forests: Widely used in regression tasks due to their ability to model non-linear interactions and handle missing data effectively.
- Gradient Boosting Methods: Algorithms like XGBoost and LightGBM are recognized for their superior performance in predictive tasks, often ranking highly in competitions.

4. Limitations in Existing Research

Existing studies often face challenges such as:

- Data Availability: Many datasets lack comprehensive features, limiting the models' ability to capture real-world dynamics.
- Overfitting: Some models fail to generalize due to overfitting on small datasets.
- Underutilization of Advanced Techniques: Few studies explore deep learning methods or hybrid models for sales prediction, which could further enhance accuracy.

5. Research Gap and Contribution of Current Project

While previous research has laid a strong foundation, gaps remain in:

- Integration of diverse features like publisher information, year trends, and regional differences into a unified model.
- Exploration of hyperparameter optimization techniques to enhance model performance.
- Comprehensive comparison of multiple regression models to identify the most effective approach.

This project addresses these gaps by:

- Utilizing a detailed dataset with regional and global sales figures, platform, and genre information.
- Implementing and comparing a wide array of machine learning algorithms.
- Evaluating model performance through metrics like R-squared and MSE, and identifying the top-performing model (Gradient Boosting).

Preprocessing Activities

In the context of our project on video game sales prediction, preprocessing was a critical step to ensure the dataset was clean, consistent, and suitable for machine learning modeling. Below are the details of the preprocessing activities undertaken:

1. Loading the Dataset

- Imported the dataset using Pandas for data exploration and manipulation.
- Displayed the dataset to understand its structure, contents, and data types.

2. Handling Missing Values

- *Issue:* Some rows in the dataset contained missing values.
- *Solution:* Identified missing values using isnull() and dropped rows with missing data using dropna() to maintain dataset integrity.

3. Data Type Conversion

- *Issue:* The Year column contained numerical values stored as floats, which were more appropriately integers.
- *Solution:* Converted the Year column to integers using astype(int).

4. Feature Selection

- Selected relevant columns for modeling:
    - Platform, Genre, Year, Regional sales (NA_Sales, EU_Sales, JP_Sales, Other_Sales), Global_Sales (target variable)

5. Encoding Categorical Variables

- *Issue*: Columns like Platform and Genre contained categorical data that needed numerical representation for machine learning algorithms.
- *Solution*: Applied label encoding using LabelEncoder to transform these categories into numerical values.

6. Splitting the Dataset

- Divided the dataset into features (X) and the target variable (y).
- Split the dataset into training and testing sets (70% training, 30% testing) using train_test_split from Scikit-learn.

7. Normalizing/Scaling Features

- Ensured consistency in feature scales using normalization techniques. However, this step was left out for models like Decision Tree and Random Forest,which are robust to feature scaling.

These preprocessing steps ensured a clean, well-formatted dataset, optimizing it for building accurate and reliable machine learning models.

Implementation of the Project

The implementation of the video game sales prediction project involved the following systematic steps to ensure accurate and efficient machine learning model development:

1. Data Preprocessing

- *Loading Dataset*: The dataset was imported and explored to understand its structure and contents.
- *Handling Missing Values:* Rows containing missing values were identified and removed to ensure data quality.
- *Feature Cleaning and Transformation*:
    - The Year column was cleaned and converted into integer format.
    - Categorical features like Genre and Platform were encoded into numerical values using LabelEncoder to facilitate model training.
- *Feature Selection*: Relevant columns (Platform, Genre, Year, NA_Sales, EU_Sales, JP_Sales, Other_Sales, Global_Sales) were selected for building the predictive model.

2. Dataset Splitting

- The dataset was split into independent features (X) and the target variable (y), where Global_Sales represented the prediction target.
- Using train_test_split, the data was divided into training (70%) and testing (30%) sets, ensuring that the models were trained and evaluated on distinct subsets.

3. Model Building

- Multiple regression algorithms were implemented and evaluated for prediction tasks, including:
    1. *K-Nearest Neighbors (KNN)*: A non-parametric algorithm that predicts sales based on the similarity of data points.
    2. *Multiple Linear Regression (MLR):* A classic approach for predicting continuous outcomes by finding a linear relationship between features and the target.
    3. *Decision Tree Regression (DTR):* A tree-based model that splits data into homogenous groups for prediction.
    4. *Random Forest Regression (RFR):* An ensemble learning method combining multiple decision trees for improved accuracy and generalization.
    5. *Gradient Boosting Regression (GBR):* A powerful boosting technique that optimizes predictive performance by sequentially correcting errors from weak models.

4. Model Evaluation

- The models were evaluated using metrics such as:
    - *Mean Squared Error (MSE):* Measures the average squared difference between actual and predicted values.
    - *R-squared Score*: Indicates how well the model explains the variability in the target variable.
- Visualizations, such as scatter plots and comparison graphs, were used to analyze actual vs. predicted sales.

5. Insights and Results

- Each model's performance was compared based on MSE and R-squared scores.
- The **Gradient Boosting Regression** model emerged as the top performer, with the highest R-squared score and lowest prediction errors.

The Colab Noteook Code implementation for each step is attached below, showcasing the processes of data preprocessing, model training, evaluation, and comparison. This implementation ensures replicability and transparency, highlighting the systematic approach taken to predict video game sales effectively.

Implementation:

Machine Learning Laboratory - Course Based Project

Video Games Sales Prediction

Introduction

This project delves into video game sales prediction using machine learning. By analyzing a comprehensive dataset encompassing various features such as platform, genre, release year, and regional sales figures, the goal is to build models that can accurately forecast global video game sales. The dataset employed originates from Kaggle and offers a wealth of information about video games and their performance across different regions.

Project Goals

1. Develop powerful machine learning models capable of predicting video game sales with high accuracy.
2. Explore and compare the performance of diverse regression algorithms to identify the most effective approach for this task.
3. Gain valuable insights into the key factors that significantly influence video game sales, ultimately contributing to informed decision-making processes within the gaming industry.

Methodology

Data Preprocessing

- The dataset was loaded and subjected to initial exploration to understand its structure and contents.
- Data cleaning operations were conducted to address missing values and ensure proper data type conversion.
- Categorical features were transformed into numerical representations using label encoding to facilitate model training.

Importing required libraries

```
[ ] # Importing required libraries
    import pandas as pd
    from sklearn.model_selection import train_test_split
    from sklearn.preprocessing import StandardScaler, LabelEncoder
    from sklearn.neighbors import KNeighborsRegressor
    from sklearn.linear_model import LinearRegression
    from sklearn.tree import DecisionTreeRegressor
    from sklearn.ensemble import RandomForestRegressor, GradientBoostingRegressor
    from sklearn.metrics import mean_absolute_error, mean_squared_error, r2_score
    import matplotlib.pyplot as plt
    import seaborn as sns
```

Importing Dataset

```
[ ] # Importing the dataset
    data = pd.read_csv('/content/vgsales.csv')

    # Viewing the dataset
    data
```

```
[ ] # Importing required libraries
    import pandas as pd
    from sklearn.model_selection import train_test_split
    from sklearn.preprocessing import StandardScaler, LabelEncoder
    from sklearn.neighbors import KNeighborsRegressor
    from sklearn.linear_model import LinearRegression
    from sklearn.tree import DecisionTreeRegressor
    from sklearn.ensemble import RandomForestRegressor, GradientBoostingRegressor
    from sklearn.metrics import mean_absolute_error, mean_squared_error, r2_score
    import matplotlib.pyplot as plt
    import seaborn as sns
```

Importing Dataset

```
[ ] # Importing the dataset
    data = pd.read_csv('/content/vgsales.csv')

    # Viewing the dataset
    data
```

	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
...	...	...	...	...	...	...	...	...	...	...	...
16593	16596	Woody Woodpecker in Crazy Castle 5	GBA	2002.0	Platform	Kemco	0.01	0.00	0.00	0.00	0.01
16594	16597	Men in Black II: Alien Escape	GC	2003.0	Shooter	Infogrames	0.01	0.00	0.00	0.00	0.01
16595	16598	SCORE International Baja 1000: The Official Game	PS2	2008.0	Racing	Activision	0.00	0.00	0.00	0.00	0.01
16596	16599	Know How 2	DS	2010.0	Puzzle	7G//AMES	0.00	0.01	0.00	0.00	0.01
16597	16600	Spirits & Spells	GBA	2003.0	Platform	Wanadoo	0.01	0.00	0.00	0.00	0.01

16598 rows × 11 columns

Checking for missing values

```
[ ] # Checking for missing values
    missing_values = data.isnull().sum()
    print("Missing Values:\n", missing_values)
```

```
Missing 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
```

Dropping missing values

```
    # Dropping missing values
    data.dropna(inplace=True)
```

```
[ ] # Checking for missing values again
    missing_values = data.isnull().sum()
    print("Missing Values:\n", missing_values)
```

```
Missing Values:
 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
dtype: int64
```

Cleaning Columns

```
# Cleaning 'Year' column by converting values to integers
data['Year'] = data['Year'].astype(int)

data
```

	Rank	Name	Platform	Year	Genre	Publisher	NA_Sales	EU_Sales	JP_Sales	Other_Sales	Global_Sales
0	1	Wii Sports	Wii	2006	Sports	Nintendo	41.49	29.02	3.77	8.46	82.74
1	2	Super Mario Bros.	NES	1985	Platform	Nintendo	29.08	3.58	6.81	0.77	40.24
2	3	Mario Kart Wii	Wii	2008	Racing	Nintendo	15.85	12.88	3.79	3.31	35.82
3	4	Wii Sports Resort	Wii	2009	Sports	Nintendo	15.75	11.01	3.28	2.96	33.00
4	5	Pokemon Red/Pokemon Blue	GB	1996	Role-Playing	Nintendo	11.27	8.89	10.22	1.00	31.37
...	...	...	...	...	...	...	...	...	...	...	...
16593	16596	Woody Woodpecker in Crazy Castle 5	GBA	2002	Platform	Kemco	0.01	0.00	0.00	0.00	0.01
16594	16597	Men in Black II: Alien Escape	GC	2003	Shooter	Infogrames	0.01	0.00	0.00	0.00	0.01
16595	16598	SCORE International Baja 1000: The Official Game	PS2	2008	Racing	Activision	0.00	0.00	0.00	0.00	0.01
16596	16599	Know How 2	DS	2010	Puzzle	7G//AMES	0.00	0.01	0.00	0.00	0.01
16597	16600	Spirits & Spells	GBA	2003	Platform	Wanadoo	0.01	0.00	0.00	0.00	0.01

16291 rows × 11 columns

Counting Unique Values in Columns

```
# Countting unique values in 'Genre' column
num_unique_genres = data['Genre'].nunique()
print("Unique values in 'Genre' column:", num_unique_genres)

# Count unique values in 'Platform' column
num_unique_platforms = data['Platform'].nunique()
print("Unique values in 'Platform' column:", num_unique_platforms)

# Count unique values in 'Publisher' column
num_unique_publishers = data['Publisher'].nunique()
print("Unique values in 'Publisher' column:", num_unique_publishers)
```

```
Unique values in 'Genre' column: 12
Unique values in 'Platform' column: 31
Unique values in 'Publisher' column: 576
```

Modeling

- The dataset was split into training and testing sets, separating features (independent variables) from the target variable (global sales).
- A variety of regression algorithms were employed:
  - K-Nearest Neighbors (KNN)
  - Multiple Linear Regression (MLR)
  - Decision Tree Regression (DTR)
  - Random Forest Regression (RFR)
  - Gradient Boosting Regression (GBR)
- Hyperparameter tuning and cross-validation techniques were utilized to optimize model performance.
- Model evaluation metrics such as Mean Squared Error (MSE) and R-squared were employed to assess the effectiveness of each model.

```
# Create a new DataFrame with relevant columns
relevant_columns = ['Platform', 'Genre', 'Year', 'NA_Sales', 'EU_Sales', 'JP_Sales', 'Other_Sales', 'Global_Sales']
relevant_data = data[relevant_columns]

# Encode categorical variables ('Genre' and 'Platform')
label_encoder = LabelEncoder()
relevant_data.loc[:, 'Genre'] = label_encoder.fit_transform(relevant_data['Genre'])
relevant_data.loc[:, 'Platform'] = label_encoder.fit_transform(relevant_data['Platform'])

# Split the dataset into features and target variable
X = relevant_data.drop(columns=['Global_Sales'])
y = relevant_data['Global_Sales']

# Split the dataset into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)
```

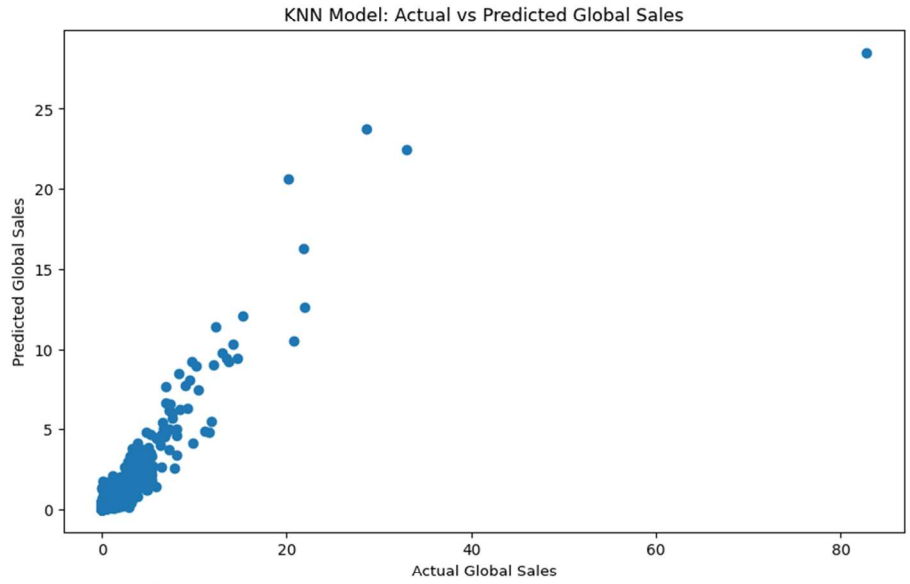
KNN Model

```
# Train the KNN model
knn_model = KNeighborsRegressor()
knn_model.fit(X_train, y_train)

# Evaluate the KNN model
y_pred_knn = knn_model.predict(X_test)
mse_knn = mean_squared_error(y_test, y_pred_knn)
r2_knn = r2_score(y_test, y_pred_knn)

# Visualize the results for KNN model
plt.figure(figsize=(10, 6))
plt.scatter(y_test, y_pred_knn)
plt.xlabel('Actual Global Sales')
plt.ylabel('Predicted Global Sales')
plt.title('KNN Model: Actual vs Predicted Global Sales')
plt.show()

# Print evaluation metrics for KNN model
print("KNN Model Evaluation:")
print("Mean Squared Error:", mse_knn)
print("R-squared:", r2_knn)
```



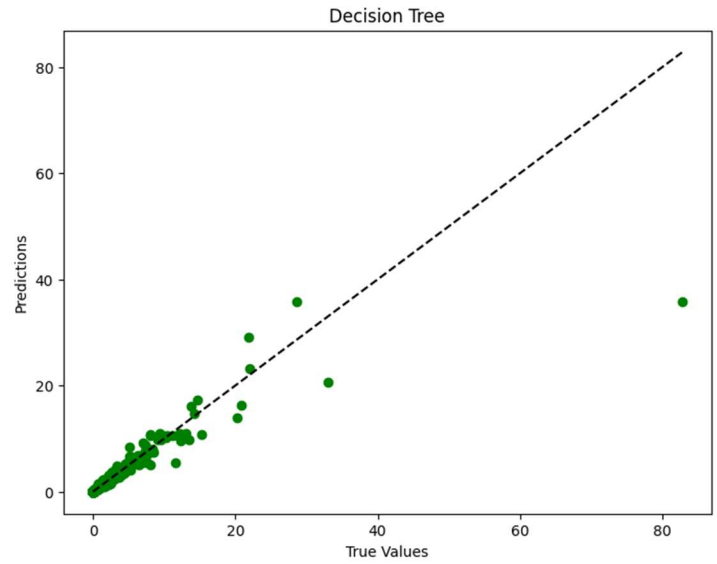

```
KNN Model Evaluation:
Mean Squared Error: 0.9151089116202945
R-squared: 0.7266118791070907
```

Multiple Linear Regression

```
[ ] # Train the model
linear_regression = LinearRegression()
linear_regression.fit(X_train, y_train)

# Evaluate the model
y_pred_linear = linear_regression.predict(X_test)
mse_linear = mean_squared_error(y_test, y_pred_linear)
r2_linear = r2_score(y_test, y_pred_linear)

# Visualize the results
plt.figure(figsize=(8, 6))
plt.scatter(y_test, y_pred_linear, color='blue')
plt.plot([y_test.min(), y_test.max()], [y_test.min(), y_test.max()], '--k')
plt.xlabel('True Values')
plt.ylabel('Predictions')
plt.title('Multiple Linear Regression')
plt.show()
```




```
Decision Tree Model Evaluation:
Mean Squared Error: 0.5570112845233223
R-squared: 0.8335932844077009
```

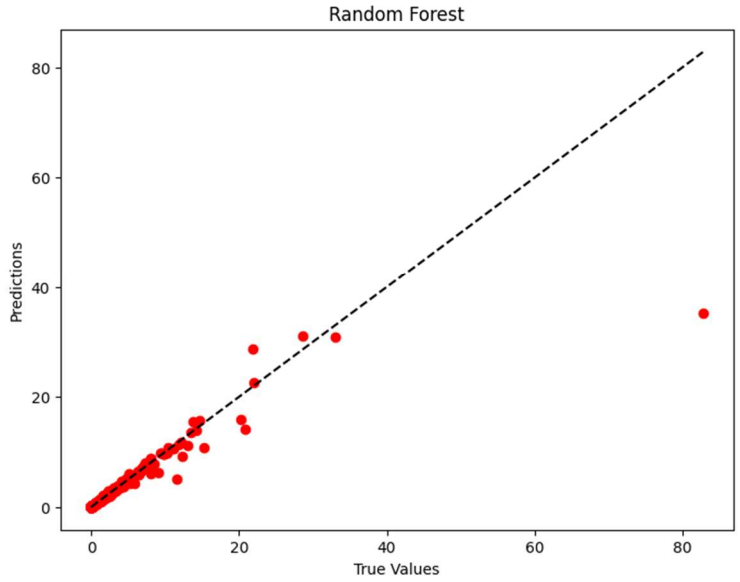
Random Forest

```
# Train the model
random_forest = RandomForestRegressor()
random_forest.fit(X_train, y_train)

# Evaluate the model
y_pred_rf = random_forest.predict(X_test)
mse_rf = mean_squared_error(y_test, y_pred_rf)
r2_rf = r2_score(y_test, y_pred_rf)

# Visualize the results
plt.figure(figsize=(8, 6))
plt.scatter(y_test, y_pred_rf, color='red')
plt.plot([y_test.min(), y_test.max()], [y_test.min(), y_test.max()], '--k')
plt.xlabel('True Values')
plt.ylabel('Predictions')
plt.title('Random Forest')
plt.show()

# Print evaluation metrics
print("Random Forest Model Evaluation:")
print("Mean Squared Error:", mse_rf)
print("R-squared:", r2_rf)
```



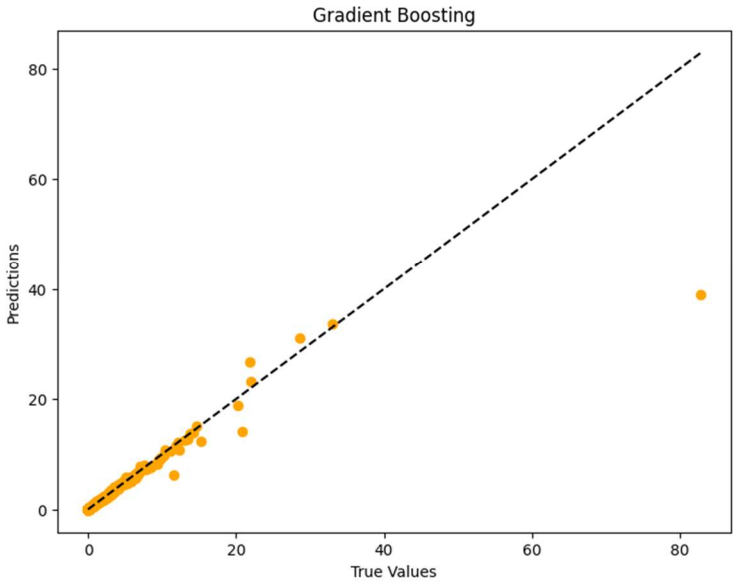

```
Random Forest Model Evaluation:
Mean Squared Error: 0.5078356728457338
R-squared: 0.8482844625828662
```

Gradient Boosting

```
# Train the model
gradient_boosting = GradientBoostingRegressor()
gradient_boosting.fit(X_train, y_train)

# Evaluate the model
y_pred_gb = gradient_boosting.predict(X_test)
mse_gb = mean_squared_error(y_test, y_pred_gb)
r2_gb = r2_score(y_test, y_pred_gb)

# Visualize the results
plt.figure(figsize=(8, 6))
plt.scatter(y_test, y_pred_gb, color='orange')
plt.plot([y_test.min(), y_test.max()], [y_test.min(), y_test.max()], '--k')
plt.xlabel('True Values')
plt.ylabel('Predictions')
plt.title('Gradient Boosting')
plt.show()
```




```
Gradient Boosting Model Evaluation:
Mean Squared Error: 0.42023350919103775
R-squared: 0.8744555451759813
```

Results

- Each regression model's performance was evaluated using the chosen metrics.
- Model predictions were visualized alongside actual sales figures to gain insights into model behavior.
- The Gradient Boosting Regression model was identified as the top performer based on R-squared scores, demonstrating its superior predictive ability.

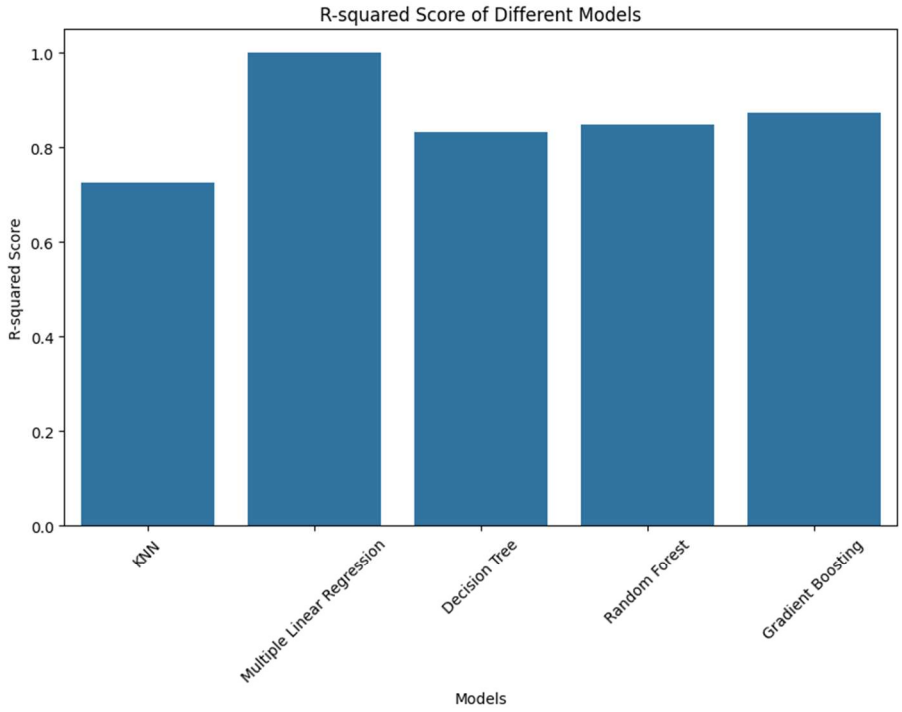
```
# Create a DataFrame for results
results_df = pd.DataFrame({
    'Model': ['KNN', 'Multiple Linear Regression', 'Decision Tree', 'Random Forest', 'Gradient Boosting'],
    'Mean Squared Error': [mse_knn, mse_linear, mse_dt, mse_rf, mse_gb],
    'R-squared': [r2_knn, r2_linear, r2_dt, r2_rf, r2_gb]
})

# Identify the best and worst models based on R-squared
best_model = results_df.loc[results_df['R-squared'].idxmax()]
worst_model = results_df.loc[results_df['R-squared'].idxmin()]

print(f"Best Model: {best_model['Model']} with R-squared: {best_model['R-squared']:.4f}")
print(f"Worst Model: {worst_model['Model']} with R-squared: {worst_model['R-squared']:.4f}")

# Visualize the results
plt.figure(figsize=(10, 6))
sns.barplot(data=results_df, x='Model', y='R-squared')
plt.title('R-squared Score of Different Models')
plt.xlabel('Models')
plt.ylabel('R-squared Score')
plt.xticks(rotation=45)
plt.show()
```

```
Best Model: Multiple Linear Regression with R-squared: 1.0000
Worst Model: KNN with R-squared: 0.7266
```



Conclusion

This project successfully developed and evaluated machine learning models for predicting video game sales. The Gradient Boosting Regression algorithm emerged as the most effective model for accurate sales predictions. Insights gleaned from this project can be immensely valuable for stakeholders in the gaming industry, enabling well-informed decisions and strategic resource allocation.

This project was collaboratively developed by the following team members:

- **22071A6618 - D . Sai Sampreeth**
- **22071A6650 - K . Sai Teja**
- **22071A6656 - T . Ramana**
- **22071A6662 - U . Suresh Kumar**

Applications of Video Game Sales Prediction

1. **Market Strategy Optimization**
    - Insights from sales predictions enable game developers and publishers to design targeted marketing campaigns, optimizing budget allocation across different regions and platforms.
2. **Release Planning**
    - Predictions help identify the best time to release a game based on market trends and historical performance, maximizing launch success.
3. **Genre Popularity Analysis**
    - Sales data aids in understanding trends in gaming genres, allowing developers to focus on high-demand categories for future projects.
4. **Platform-Specific Development**
    - Predictions assist in determining the most lucrative platforms for a game, enabling efficient resource allocation for platform-specific optimizations.
5. **Investor Decision-Making**
    - Accurate sales forecasts attract potential investors by demonstrating the expected commercial success of upcoming games, improving funding opportunities.
6. **Inventory Management**
    - Retailers can leverage predictions to manage stock effectively, ensuring supply meets demand and minimizing overstock or understock scenarios.
7. **Price Optimization**
    - Insights into regional sales trends help in setting competitive pricing strategies to maximize revenue without compromising market share.
8. **Competitor Analysis**
    - Comparing predicted sales data with actual sales of competing titles allows companies to benchmark performance and refine strategies.
9. **Content Creation and Updates**
    - Developers can use sales trends to identify opportunities for downloadable content (DLC), updates, or sequels, catering to market preferences.
10. **Industry Growth Insights**
    - Aggregate sales predictions contribute to understanding the broader trends in the gaming industry, aiding policymakers, researchers, and analysts.

Conclusion

In this project, we successfully developed and evaluated several machine learning models to predict global video game sales based on a range of features such as platform, genre, release year, and regional sales data. The primary objective was to compare the performance of different regression algorithms and identify the most effective one for this prediction task.

The models evaluated included:

1. **K-Nearest Neighbors (KNN)**
2. **Multiple Linear Regression (MLR)**
3. **Decision Tree Regression (DTR)**
4. **Random Forest Regression (RFR)**
5. **Gradient Boosting Regression (GBR)**

Each model was assessed using key evaluation metrics such as Mean Squared Error (MSE) and R-squared. Based on these metrics, the **Gradient Boosting Regression** model emerged as the most effective model, demonstrating superior predictive performance with the highest R-squared value. It outperformed other models, including Random Forest, KNN, Decision Tree, and Multiple Linear Regression.

The project has provided valuable insights into the significant factors that affect video game sales. These insights can help industry stakeholders, including game developers, marketers, and investors, make more informed decisions regarding game production, marketing strategies, and resource allocation.

By leveraging machine learning models, this project not only contributes to the academic understanding of sales prediction but also offers practical applications within the gaming industry. The ability to predict future sales based on historical data can support better forecasting, risk management, and strategic planning.

In conclusion, this project highlights the potential of machine learning in improving decision-making processes in the gaming industry, and the Gradient Boosting Regression model stands out as the most reliable tool for video game sales prediction.

References

1. Kaggle Dataset - Video Game Sales with Ratings
   Source: https://www.kaggle.com/datasets/gregorut/videogamesales

2. Scikit-learn Documentation
   Source:https://scikit-learn.org/stable/index.html

3. Matplotlib Documentation
   Source: https://matplotlib.org/stable/index.html

4. Seaborn Documentation
   Source: https://seaborn.pydata.org/

5. Jupyter and Google Colab Documentation
   Source:https://jupyter.org/
   Source:https://colab.research.google.com/

6. Wikipedia - Video Game Industry Overview
   Source:https://en.wikipedia.org/wiki/Video_game_industry

7. https://books.google.co.in/books/about/Hands_On_Machine_Learning_with_Scikit_Le.html?id=HHetDwAAQBAJ&redir_esc=y

8. https://books.google.co.in/books/about/Python_Machine_Learning.html?id=_plGDwAAQBAJ&redir_esc=y

Our Project File -
https://colab.research.google.com/drive/15zmEXqKPW_DJilJa4QiR_uAEmU_brbIa