

# Video Game Genre Prediction Program

John McGinnes

## Task 2 parts A, B, C and D

Part A: Letter of Transmittal .....	2
Letter of Transmittal Requirements .....	<b>Error! Bookmark not defined.</b>
Letter Template .....	<b>Error! Bookmark not defined.</b>
Part B: Project Proposal Plan .....	4
Project Summary.....	4
Data Summary.....	4
Implementation .....	5
Timeline.....	5
Evaluation Plan.....	6
Resources and Costs .....	6
Part C: Application .....	8
Part D: Post-implementation Report .....	9
Solution Summary.....	9
Data Summary.....	9
Machine Learning.....	11
Validation .....	12
Visualizations .....	13
User Guide .....	17
Reference Page .....	26

# Part A:

03/22/2016

Reginald Stephenson

Bethesda Softworks

1370 Piccard Dr.

Rockville, MD 20850

Dear Mr. Stephenson,

I am pleased to present the proposal for the development of a predictive model designed to forecast the top video game genres for 2018, based on historical sales data up to 2016. The project utilizes machine-learning techniques to analyze consumer trends in the gaming industry, providing Bethesda Softworks with valuable insights to guide marketing, product development, and resource allocation.

The video game industry is highly dynamic, with shifts in consumer preferences affecting game sales year over year. Accurately predicting which genres will dominate the market in the near future is crucial to staying competitive. Without these insights, your teams may struggle to prioritize game development and marketing strategies, leading to inefficient resource allocation and missed opportunities.

The solution I propose is a data-driven application that uses historical data from 1980 to 2016 to predict the top video game genres for 2018. Using the Gradient Boosting Regressor machine learning model, the application forecasts global sales for each genre in 2018, based on regional sales trends and genre popularity. This predictive model will allow Bethesda Softworks to strategically target the most promising genres, optimizing game releases and marketing campaigns for maximum market impact.

The project will be developed using Python in the Jupyter Notebook development environment. Libraries such as pandas, sklearn, and seaborn will be used for data analysis and visualization, and the model will include interactive visualizations to explore sales data and predictions for individual genres.

The expected project timeline is approximately four weeks, including model training, visualization development, and user interface design. The project development will follow the agile methodology to ensure


timely delivery, and all data will be sourced from Kaggle.com, an open-source platform. As the data is freely available and does not include any personally identifiable information, there are no ethical or legal concerns regarding its use. The estimated cost of the project is \$4500.00, covering hardware, software tools, and labor.

With a strong background in data science, machine learning, and Python programming, I am confident in my ability to lead this project to success. I have successfully developed similar predictive models in the past, and my experience with data visualization ensures that this solution will be both technically sound and accessible to non-technical stakeholders.

Thank you for considering this proposal. I look forward to your feedback and am available for any further questions.

Sincerely,

---



---

John McGinnes, Lead Developer

## Part B:

### Project Summary

This proposal addresses the need for a predictive model that forecasts the top video game genres for 2018, based on historical sales data up to 2016. By leveraging machine learning techniques, specifically the Gradient Boosting Regressor algorithm, we aim to help Bethesda Softworks make more informed decisions regarding game development, marketing strategies, and resource allocation.

The main deliverables will include:

- A fully trained predictive model capable of forecasting global sales for each genre.
- Interactive data visualizations that allow users to explore historical sales trends and predictions.
- A user guide that explains how to use the application, interpret predictions, and generate reports.
- Source code documentation for reproducibility and future maintenance.

As a leading video game company in the industry, Bethesda Softworks faces the challenge of a competitive market where preferences shift rapidly. To remain ahead of competitors, the company would greatly benefit from predictive insight into which game genres will be the most popular in the near future. Our product will fulfill the need for actionable data, enabling Bethesda Softworks to prioritize development and marketing efforts more effectively.

Currently, there is no comprehensive tool that combines historical sales data with predictive modeling to project which genres will dominate the market. Bethesda's existing tools might provide sales trends, but they lack the predictive power and accuracy that this model offers. By integrating sales data from multiple regions and genre classifications, the product will bridge this gap and provide more granular forecasts.

### Data Summary

The raw data used for this project comes from Kaggle's publicly available dataset Video Game Sales. Published by Pederson, U.T. (2021). (Link to the raw dataset:

<https://www.kaggle.com/datasets/ulrikthygpedersen/video-games-sales>) This dataset includes global sales data by region as well as genre and publisher information. The data will be imported to the program from a csv file and used to train a machine learning model. Any data anomalies such as missing values or outliers will be handled using imputation methods and standardization techniques.

Since this dataset is anonymized and freely available, there are no significant ethical or legal concerns associated with its use.

The data will be processed and managed through preprocessing, where missing values will be imputed, and categorical variables will be encoded. Standardization techniques will be applied to numerical data to ensure the model trains efficiently.

## Implementation

The project will follow an agile methodology, allowing for iterative development and constant feedback throughout the process.

We will begin by preprocessing the data, including handling missing values, encoding categorical variables, and normalizing numerical values. Next, we will train the Gradient Boosting Regressor model on the data to predict global sales for each genre in 2018. During model development, cross-validation will be used to assess model performance and tune hyperparameters for optimal results.

## Timeline

Milestone or deliverable	Duration (hours or days)	Projected start date	Anticipated end date
Data Collection and Preprocessing	1 Week	03/30/2016	04/06/2016
Model Training and Hyperparameter tuning	2 Weeks	04/07/2016	04/20/2016
Visualization Development and Integration	1 Week	04/21/2016	04/27/2016
Final Testing, Documentation, and	1 Week	04/28/2016	05/04/2016

Deployment			
------------	--	--	--

## Evaluation Plan

To verify that the model meets the requirements at each stage, development verification will involve testing the model's accuracy and performance on the dataset. To ensure the model performs within specifications, the program will include the output of R-squared, Mean Absolute Error, and Cross-Validated Mean Absolute Error scores. These scores will be observed at each iteration and recorded to ensure improvement.

Final validation will compare the R-squared, Mean Absolute Error, and Cross-Validated Mean Absolute Error scores in addition to the final program runtime to ensure the program meets the specifications below:

- R-squared score > 80%
- Mean Absolute Error < 0.05
- Cross-validated Mean Absolute Error < 0.05
- Overall Runtime < 10 seconds

## Resources and Costs

Resource	Description	Cost
Kaggle Dataset	Access to Video Game Sales dataset	\$0.00
Development Tools	Python libraries (Pandas, Scikit-learn, etc.)	\$0.00
Computational Power	Local machine computations will require an Intel i7 CPU and 16GB RAM Lenovo Legion 7i laptop or comparable recommended.	\$1500.00
Data Scientist	Responsible for model development, training, and optimization. Labor cost for Data Scientist (40 hours at \$50/hour)	\$2000.00

Data Analyst	Responsible for data cleaning, preprocessing, and initial data exploration. Labor cost for Data Analyst (20 hours at \$30/hour)	\$600.00
Project Manager	Responsible for coordinating the project and communicating progress to stakeholders. Labor cost for Project Manager (10 hours at \$40/hour)	\$400.00
	<b>Total</b>	\$4500.00

## Part C: Application

Along with this document, the following files have been submitted for the program:

- C964\_Task2\_Video\_Game\_Genre\_Predictor.ipynb
- Video\_Games\_Sales\_as\_of\_2016.csv



## Part D: Post-implementation Report

### Solution Summary

The problem addressed in this project proposal was the need for Bethesda Softworks to predict the top video game genres for 2018 using historical sales data up to 2016. The objective was to provide predictive insights that would assist the company in making informed decisions about game development, marketing strategies, and resource allocation.

The solution was a predictive model built using machine learning techniques, specifically the Gradient Boosting Regressor algorithm, which was trained on historical sales data. The model forecasted global sales for each genre, helping Bethesda Softworks prioritize which genres to focus on in 2018. The solution also includes interactive data visualizations that allow the user to explore sales trends and predictions for 2018. These visualizations provide a user-friendly interface for stakeholders to understand the trends and make decisions based on the forecast.

The main interface of the application is shown below, currently displaying the predicted sales for the 'Shooter' genre in 2018:



The screenshot shows a web application interface. At the top, there is a label 'Select Genre:' followed by a dropdown menu. The dropdown menu is open, showing 'Shooter' as the selected option. Below the dropdown menu, there is a text display that reads 'Predicted sales for Shooter in 2018: 0.8166 million units'.

The project has successfully met its objectives by delivering a predictive model, accompanying visualizations, and a user guide. It provides Bethesda Softworks with actionable insights that can drive future game development and marketing decisions.

### Data Summary

The raw data used in this project was sourced from Kaggle's "Video Game Sales" dataset (Pederson, U.T., 2021). The dataset includes global video game sales data by region along with game genre and publisher information. The data was collected from public sources, aggregated, and made available on Kaggle for analysis. Because the data was intended to show the sales history from 1980 to 2016, any outliers outside of that timeframe were removed, as well as any records with null data in the year column.

The data was imported into the application from a CSV file, which was then processed through several stages:

1. Preprocessing: Missing values in sales data were imputed using the mean of the respective region. Categorical variables, such as genre and publisher, were encoded using Label Encoding. Numerical data was standardized to ensure efficient model training.
2. Model Development: After preprocessing, the data was split into training and testing sets. The training set was used to train the Gradient Boosting Regressor, and the model was evaluated using cross-validation techniques to ensure functionality.
3. Maintenance: The model is designed to allow future updates by re-training it with newer data if available. Additionally, future maintenance will involve regularly testing and fine-tuning the model to ensure its continued accuracy.

The dataset was appropriate for this project because it spans a wide range of years (1980-2016) and includes relevant features such as genre and regional sales, which are key to understanding market trends and making predictions. There were no significant ethical or legal concerns with using this publicly available dataset.

Here is an example of the code used for the data processing steps:

```
#####
#
#       Data Processing
#
#####

# Preprocessing and handling missing sales data
df['na_sales'] = df['na_sales'].fillna(df['na_sales'].mean())
df['eu_sales'] = df['eu_sales'].fillna(df['eu_sales'].mean())
df['jp_sales'] = df['jp_sales'].fillna(df['jp_sales'].mean())
df['other_sales'] = df['other_sales'].fillna(df['other_sales'].mean())

# Ensure 'year' is an integer
df['year'] = df['year'].astype(int)

# Apply Label Encoding to categorical columns ('genre' and 'publisher')
label_encoder_genre = LabelEncoder()
df['genre_encoded'] = label_encoder_genre.fit_transform(df['genre'])

label_encoder_publisher = LabelEncoder()
df['publisher_encoded'] = label_encoder_publisher.fit_transform(df['publisher'].astype(str))

# Filter data for years <= 2016 for training
train_df = df[df['year'] <= 2016]

# Prepare the features and target variable for model training
X_train = train_df[['genre_encoded', 'na_sales', 'eu_sales', 'jp_sales', 'other_sales']]
y_train = train_df['global_sales']
```

Here is a sample of the dataset CSV file used for input:

	A	B	C	D	E	F	G	H	I	J	K
1	rank	name	platform	year	genre	publisher	na_sales	eu_sales	jp_sales	other_sales	global_sales
2	222	FIFA 17	PS4	2016	Sports	Electronic	0.28	3.75	0.06	0.69	4.77
3	272	Uncharted	PS4	2016	Shooter	Sony Com	1.3	2.07	0.18	0.65	4.2
4	352	Tom Clanc	PS4	2016	Shooter	Ubisoft	1.28	1.61	0.15	0.57	3.61
5	772	Far Cry: Pr	PS4	2016	Action	Ubisoft	0.59	1.16	0.06	0.33	2.13
6	847	Tom Clanc	XOne	2016	Shooter	Ubisoft	1.2	0.62	0	0.18	2.01
7	1028	Overwatch	PS4	2016	Shooter	Activision	0.64	0.68	0.14	0.26	1.73
8	1158	No Man's	PS4	2016	Action	Hello Gam	0.58	0.74	0.02	0.26	1.6
9	1191	Dark Soul	PS4	2016	Role-Playi	Namco Ba	0.58	0.44	0.33	0.21	1.56
10	1226	FIFA 17	XOne	2016	Sports	Electronic	0.17	1.26	0	0.1	1.53
11	1391	Doom (20	PS4	2016	Shooter	Bethesda	0.49	0.66	0.02	0.22	1.39
12	1570	Yokai Wat	3DS	2016	Action	Level 5	0	0	1.27	0	1.27
13	1630	Madden N	PS4	2016	Sports	Electronic	0.92	0.08	0	0.23	1.23
14	1703	NBA 2K17	PS4	2016	Sports	Take-Two	0.83	0.14	0	0.22	1.19
15	1729	Ratchet &	PS4	2016	Platform	Sony Com	0.32	0.64	0.04	0.18	1.17
16	1960	Naruto Sh	PS4	2016	Fighting	Namco Ba	0.39	0.41	0.1	0.16	1.06
17	2214	The Legen	WiiU	2016	Action	Nintendo	0.48	0.3	0.08	0.08	0.94
18	2244	Pokken Tc	WiiU	2016	Fighting	Namco Ba	0.47	0.22	0.16	0.07	0.93
19	2344	EA Sports	PS4	2016	Sports	Electronic	0.28	0.47	0	0.14	0.89
20	2436	Far Cry: Pr	XOne	2016	Action	Ubisoft	0.46	0.32	0	0.07	0.85
21	2446	Overwatch	XOne	2016	Shooter	Activision	0.52	0.25	0	0.08	0.85
22	2450	Kirby: Plai	3DS	2016	Action	Nintendo	0.26	0.1	0.44	0.04	0.85
23	2459	MLB 16: Th	PS4	2016	Action	Sony Com	0.68	0	0	0.16	0.84
24	2507	Madden N	XOne	2016	Sports	Electronic	0.72	0.02	0	0.09	0.82
25	2513	Street Figh	PS4	2016	Fighting	Capcom	0.35	0.26	0.08	0.13	0.82

## Machine Learning

### Machine Learning Method: Gradient Boosting Regressor

The Gradient Boosting Regressor is an ensemble machine-learning algorithm that builds a series of decision trees to improve accuracy. Each model is trained to correct the errors made by the previous model, leading to a final model that performs well on unseen data.

The model was trained on historical video game sales data, which includes features such as genre and regional sales. After preprocessing the data, the model was trained using the training set, and cross-validation was applied to ensure the model did not overfit the training data.

Hyperparameters were tuned to optimize performance, and the model was validated using metrics like R-squared and Mean Absolute Error.

The Gradient Boosting Regressor was chosen because it is well-suited for regression tasks, handles both linear and non-linear relationships in data, and performs well even with relatively small datasets. It is known for producing accurate predictions, which is crucial for forecasting sales trends in an industry with rapidly changing preferences.

Here is an example of the machine-learning model training:

```
#####
#
#   Machine Learning Model Training
#
#####

# Initialize the GradientBoostingRegressor model
model = GradientBoostingRegressor()

# Perform Cross-Validation (5-fold) and evaluate the MAE
cross_val_scores = cross_val_score(model, X_train, y_train, cv=5, scoring='neg_mean_absolute_error')
cross_val_mae = -cross_val_scores.mean()
print(f'Cross-validated Mean Absolute Error: {cross_val_mae:.4f}')

# Fit the model on the full training data
model.fit(X_train, y_train)

# Calculate the average sales per genre across all years
genre_sales_avg = df.groupby('genre').agg({
    'na_sales': 'mean',
    'eu_sales': 'mean',
    'jp_sales': 'mean',
    'other_sales': 'mean'
}).reset_index()

# Calculate global sales by summing sales across all regions for each genre
genre_sales_avg['global_sales_avg'] = genre_sales_avg[['na_sales', 'eu_sales', 'jp_sales', 'other_sales']].sum(axis=1)
```

Here is an example of the prediction process:

```
#####
#
#   Machine Learning Model Predictions
#
#####

# Create a prediction dataset for all genres in 2018
predict_2018_data = {
    'genre_encoded': label_encoder_genre.transform(genre_sales_avg['genre']),
    'na_sales': genre_sales_avg['na_sales'],
    'eu_sales': genre_sales_avg['eu_sales'],
    'jp_sales': genre_sales_avg['jp_sales'],
    'other_sales': genre_sales_avg['other_sales']
}

predict_df = pd.DataFrame(predict_2018_data)

# Make predictions for global sales in 2018
X_predict = predict_df[['genre_encoded', 'na_sales', 'eu_sales', 'jp_sales', 'other_sales']]
y_pred_2018 = model.predict(X_predict)

# Store predicted sales for all genres for 2018
predict_df['predicted_sales_2018'] = y_pred_2018

# Convert encoded genre labels back to the original strings
predict_df['genre'] = label_encoder_genre.inverse_transform(predict_df['genre_encoded'])

# Sort the genres by the predicted sales in 2018 and assign a rank to each genre
top_genres_2018 = predict_df[['genre', 'predicted_sales_2018']].sort_values(by='predicted_sales_2018', ascending=False).reset_index(drop=True)
top_genres_2018['Rank'] = top_genres_2018.index + 1
```

## Validation

The model performance was validated using the following methods:

- R-squared: This score measures how well the model explains the variance in the target variable (global sales). A high R-squared score indicates that the model is able to capture the underlying trend in the data.
  - Goal: R-squared value greater than 0.80
  - Result: An R-squared value of 0.99 was achieved.
- Mean Absolute Error (MAE): The MAE measures the average magnitude of errors in the model's predictions. It provides an interpretable measure of how far off the predictions were, in terms of actual sales units.
  - Goal: MAE value less than 0.5.
  - Result: The final MAE was 0.0229 million units, which is a very small error, further confirming the accuracy of the model.
- Cross-Validated Mean Absolute Error: The model was evaluated using 5-fold cross-validation to ensure that the results were not dependent on a specific training-test split.
  - Goal: Cross-Validated MAE less than 0.5
  - Result: The cross-validated MAE was 0.0353 million units, showing that the model's performance was consistent across different data subsets.
- Overall Runtime: The program is designed to provide the output data quickly and efficiently, without requiring significant time to provide predictions.
  - Goal: Overall program runtime less than 10 seconds.
  - Result: On average hardware, the program consistently runs within 6 seconds.

The actual output of the accuracy metrics is shown below:

```
--- Model Evaluation Metrics ---
R-squared: 0.9990
Mean Absolute Error (MAE): 0.0229
Cross-validated MAE: 0.0348
```

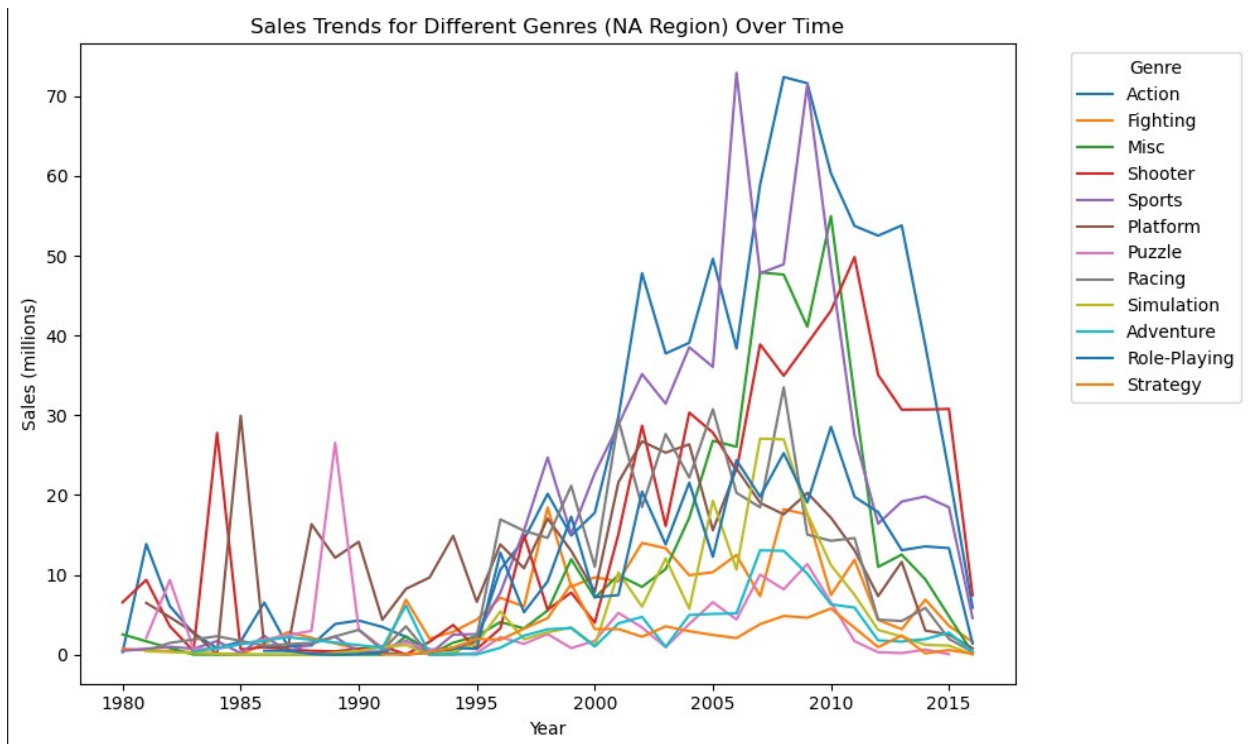
## Visualizations

The project includes the following visualizations produced by the program:

Sales Trends for Different Genres (NA Region):

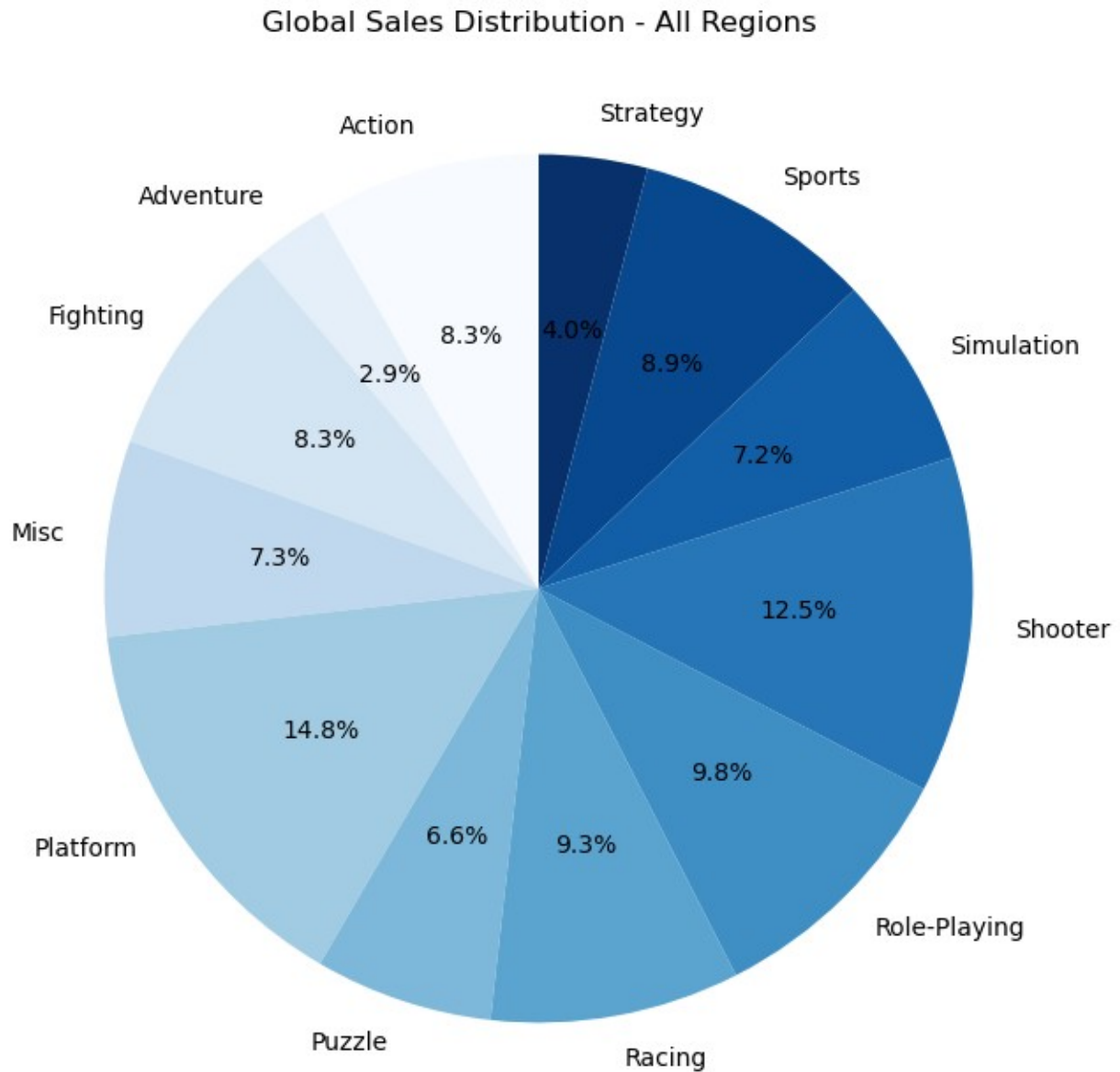
A line plot showing sales trends over time for different genres in the North American market





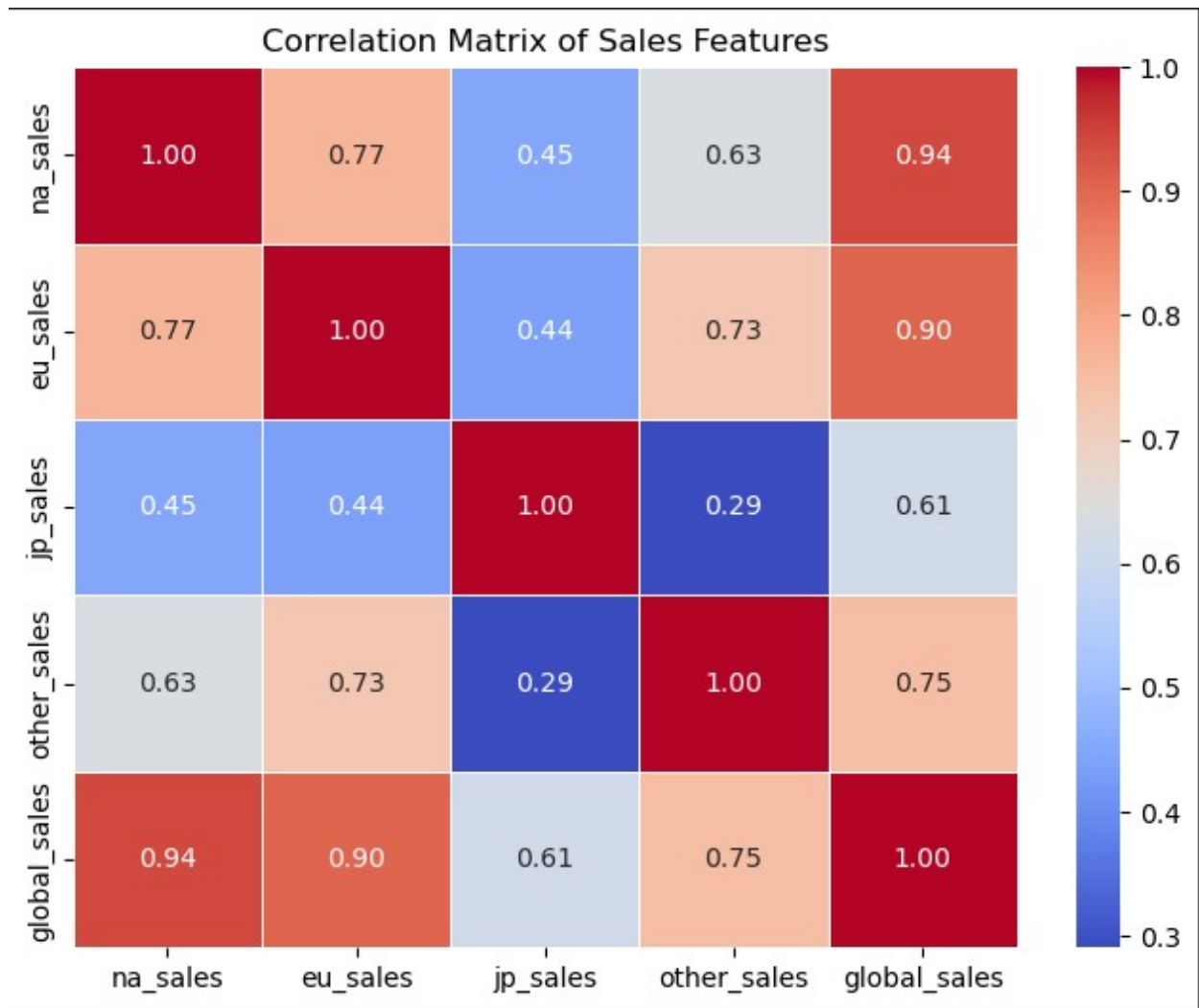
Global Sales Distribution - All Regions:

A pie chart showing the distribution of sales across all regions.



Correlation Matrix of Sales Features;

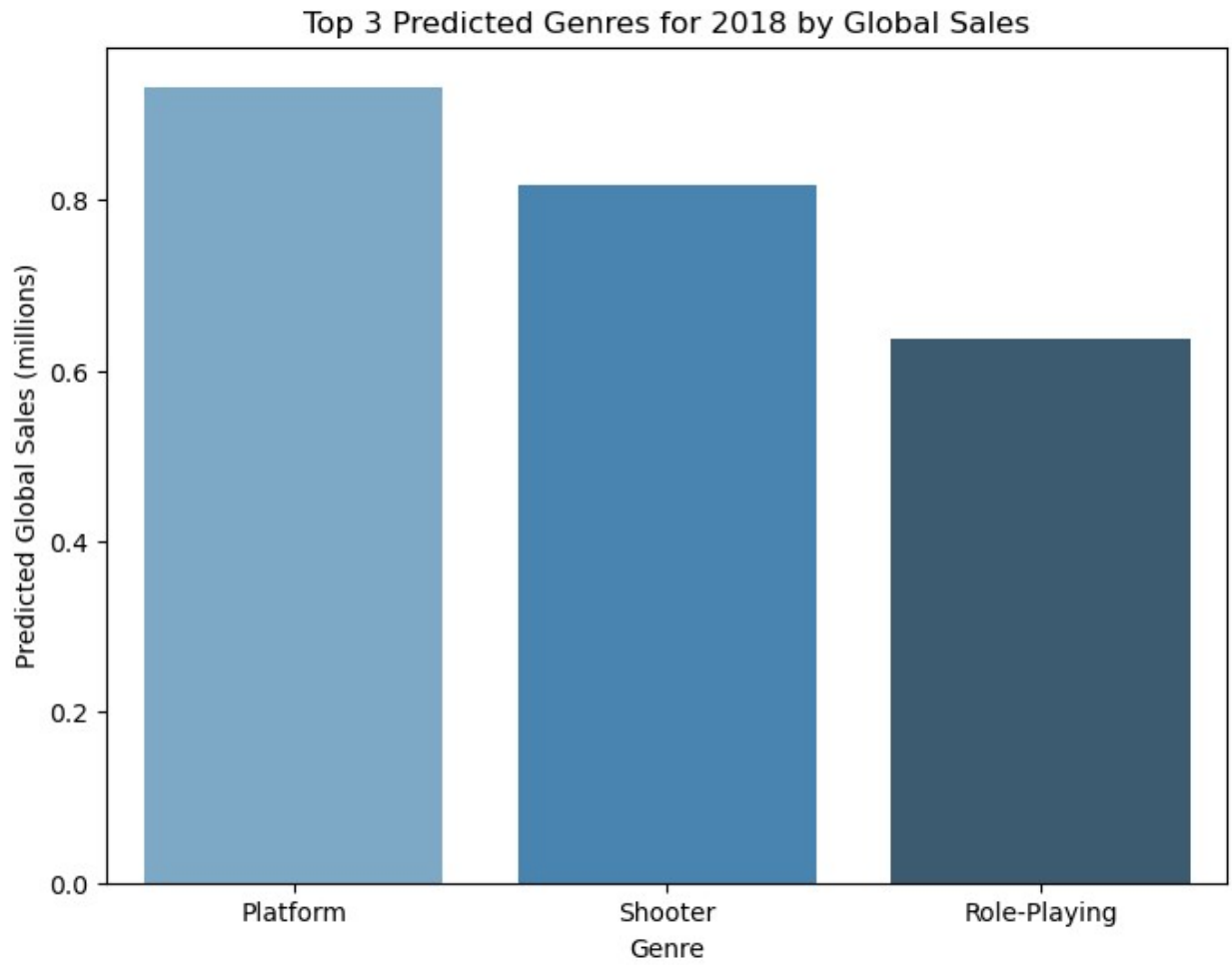
A heatmap illustrating the relations between various sales features.



Top 3 Predicted Genres for 2018 by Global Sales:

A bar plot that helps stakeholders visualize the most lucrative genres to prioritize.



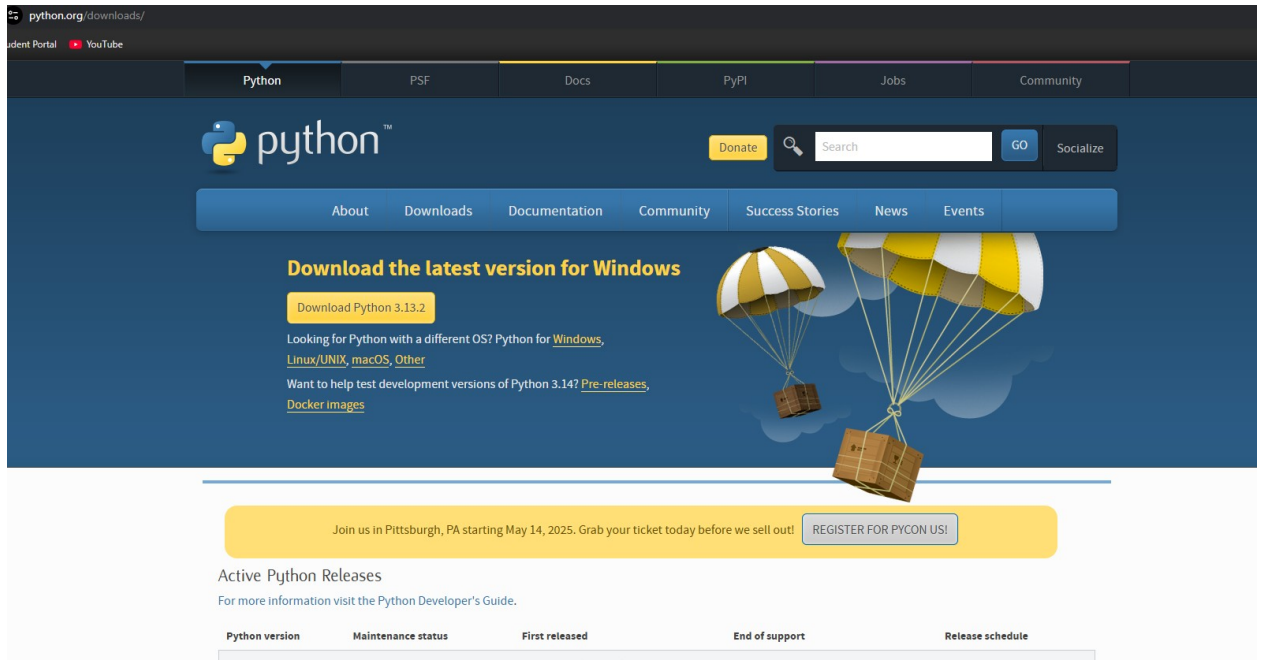


---

## User Guide

### Installation Instructions:

1. Install Python 3.12.3 on your computer using the following link:  
<https://www.python.org/downloads/> Be sure to check the box "Add Python to Path" during installation.



You can verify the Python install by opening the Windows Command Line (Windows Key + R) and typing the following:

Python --version

This should return the version number for Python.

```
Microsoft Windows [Version 10.0.26100.3476]
(c) Microsoft Corporation. All rights reserved.

C:\Users\mcgin>python --version
Python 3.11.9

C:\Users\mcgin>
```

2. Install Jupyter Notebook using the following command in the Windows Command Line:

pip install notebook

```
C:\Users\mcgin>pip install notebook
```

You can verify the installation with the following command:

Jupyter --version

```
.8.4 prompt_toolkit-3.0.50 psutil-7.0.0 pure-eval-
C:\Users\mcgin>jupyter --version
Selected Jupyter core packages...
IPython          : 9.0.2
ipykernel        : 6.29.5
ipywidgets       : not installed
jupyter_client   : 8.6.3
jupyter_core     : 5.7.2
jupyter_server   : 2.15.0
jupyterlab       : 4.3.6
nbclient         : 0.10.2
nbconvert        : 7.16.6
nbformat         : 5.10.4
notebook         : 7.3.3
qtconsole        : not installed
traitlets        : 5.14.3

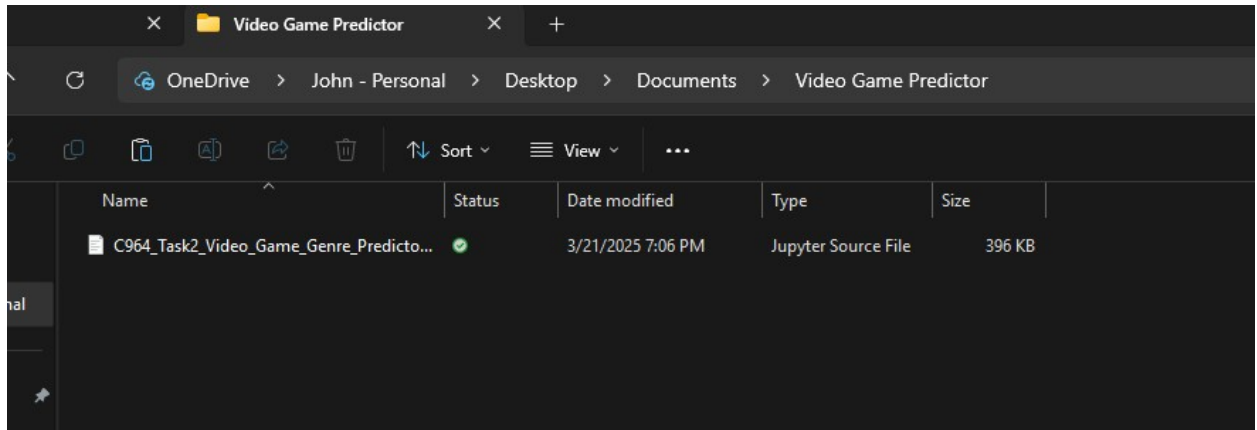
C:\Users\mcgin>
```

3. Install the necessary libraries by running the following command in your terminal or command prompt:

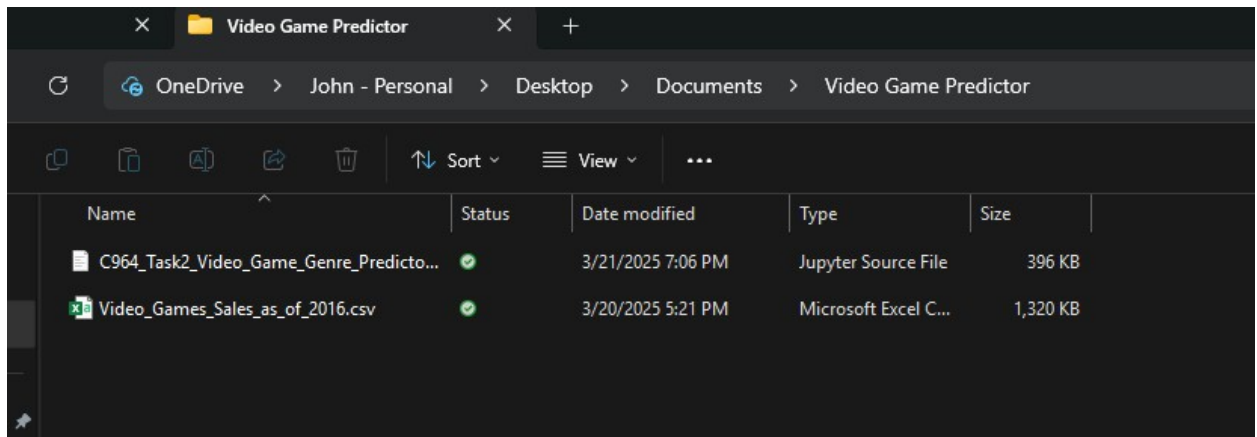
pip install pandas scikit-learn ipywidgets matplotlib seaborn

```
C:\Users\mcgin>pip install pandas scikit-learn ipywidgets matplotlib seaborn
Collecting pandas
```

4. Download the C964\_Task2\_Video\_Game\_Genre\_Predictor.ipynb and save it to a directory on your computer.



5. Download the included dataset “Video\_Games\_Sales\_as\_of\_2016.csv” and ensure that it is in the same directory as the .ipynb file.



6. In the command prompt, navigate to the folder using the command ‘cd’ followed by the path to the folder containing the .ipynb file.

```
C:\Users\mcgin>cd C:\Users\mcgin\OneDrive\Desktop\Documents\Video Game Predictor
```

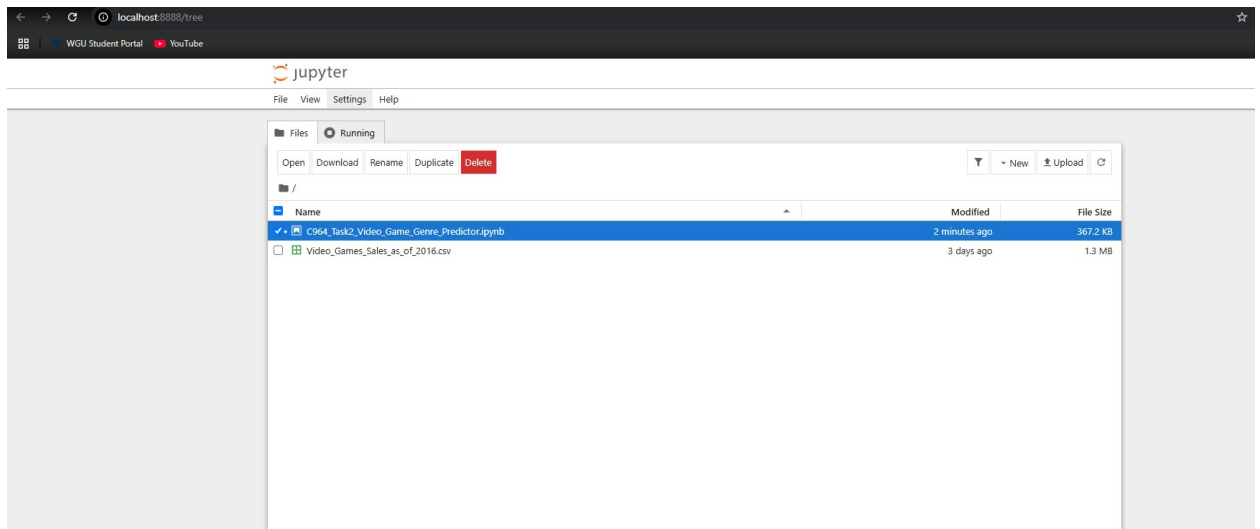
7. Open Jupyter Notebooks using the following command from the command prompt inside your directory:

jupyter notebook

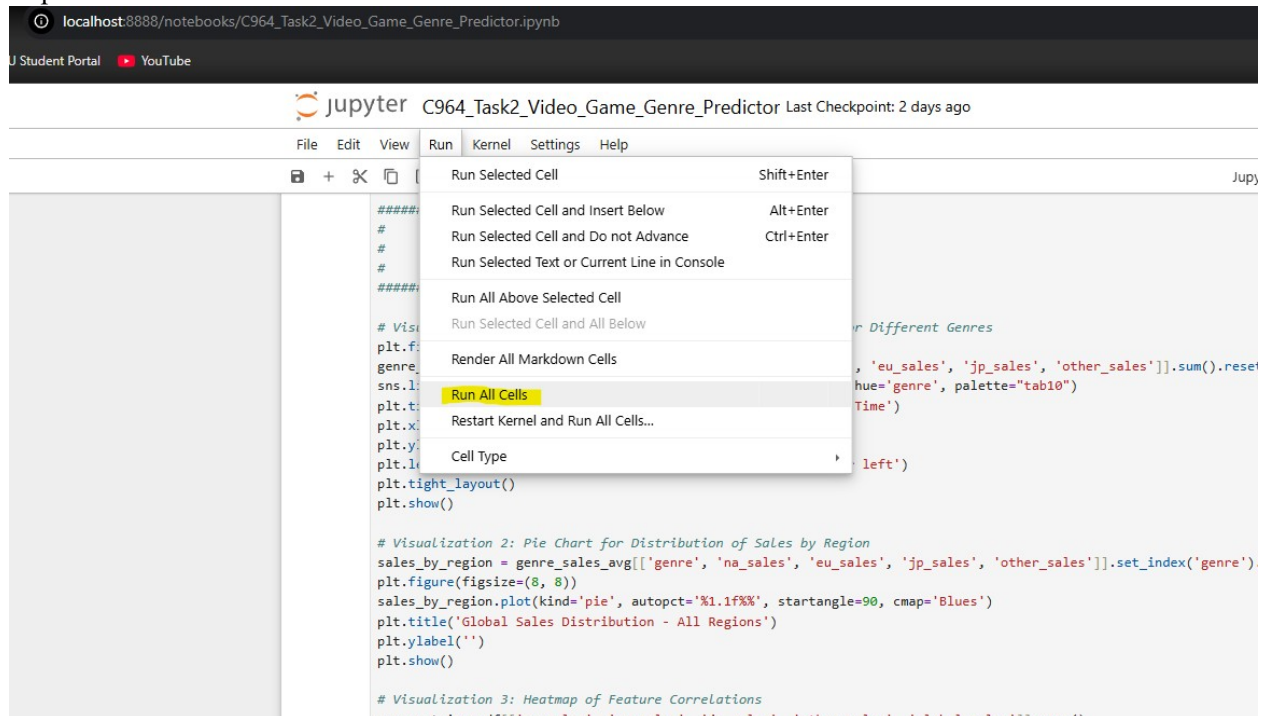
```
C:\Users\mcgin\OneDrive\Desktop\Documents\Video Game Predictor>jupyter notebook
57 2025-03-22 00:00:15.884 [Info] Extension package jupyterlab not installed
```

This will open Jupyter Notebook in your default web browser.

8. Open the C964\_Task2\_Video\_Game\_Genre\_Predictor.ipynb file.

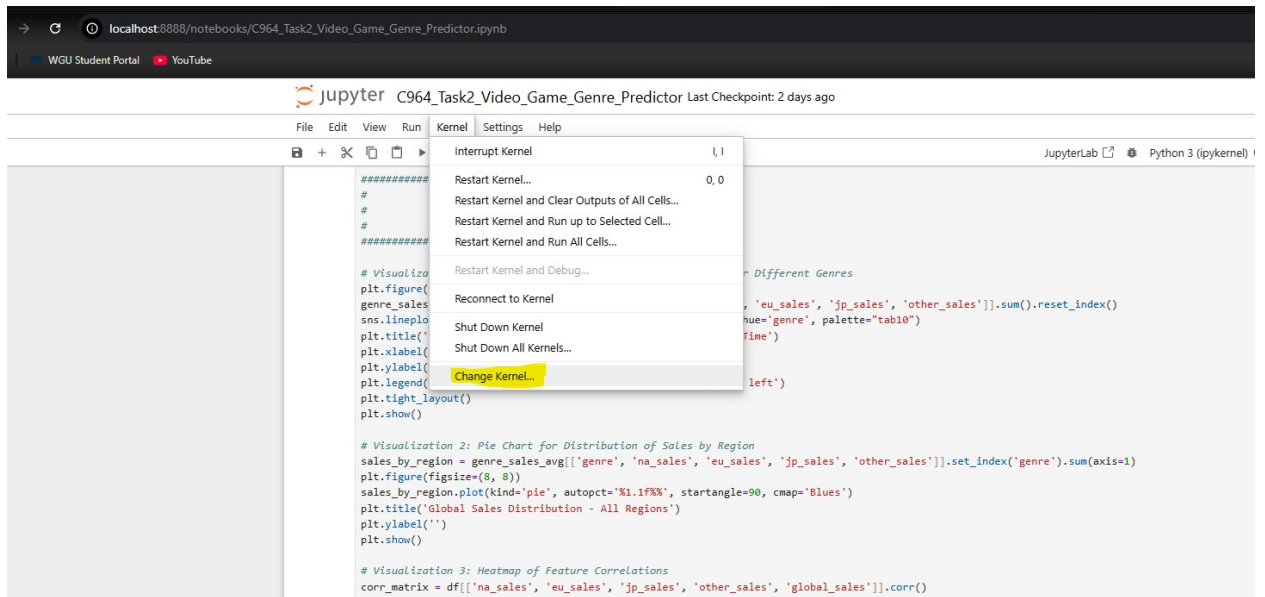


- Click the 'Run' menu at the top of the code and select 'Run All Cells' to begin the program or press Shift+Enter.

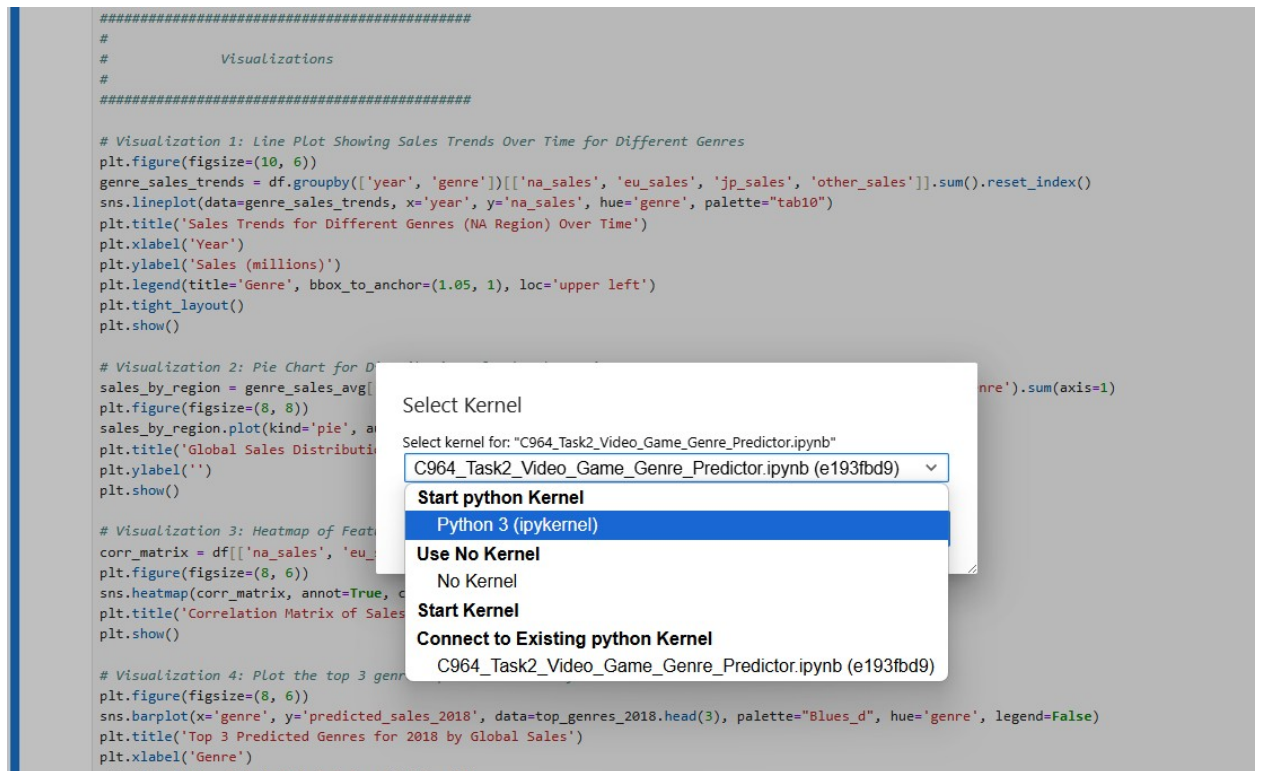


If for any reason Jupyter Notebook is not running with the Python Kernel this could cause an error.

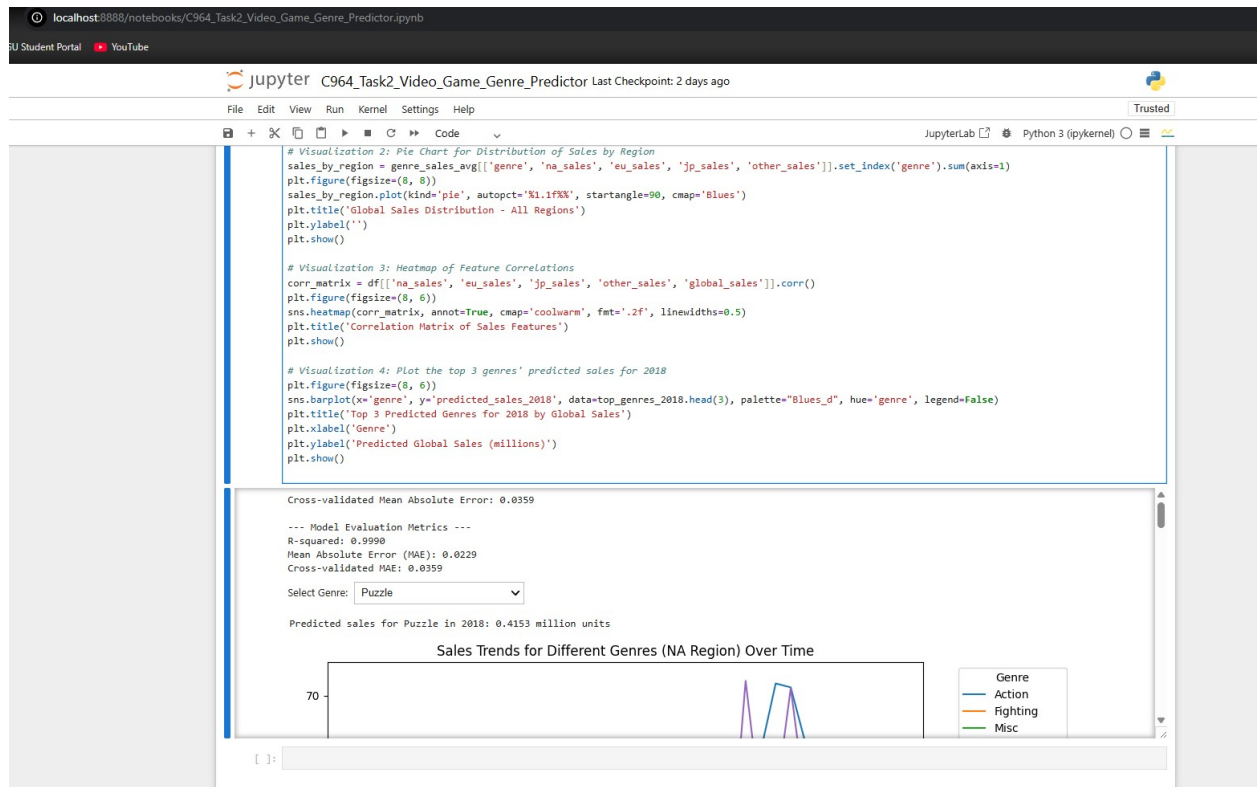
Select the Kernel Menu at the top and click 'Change Kernel':



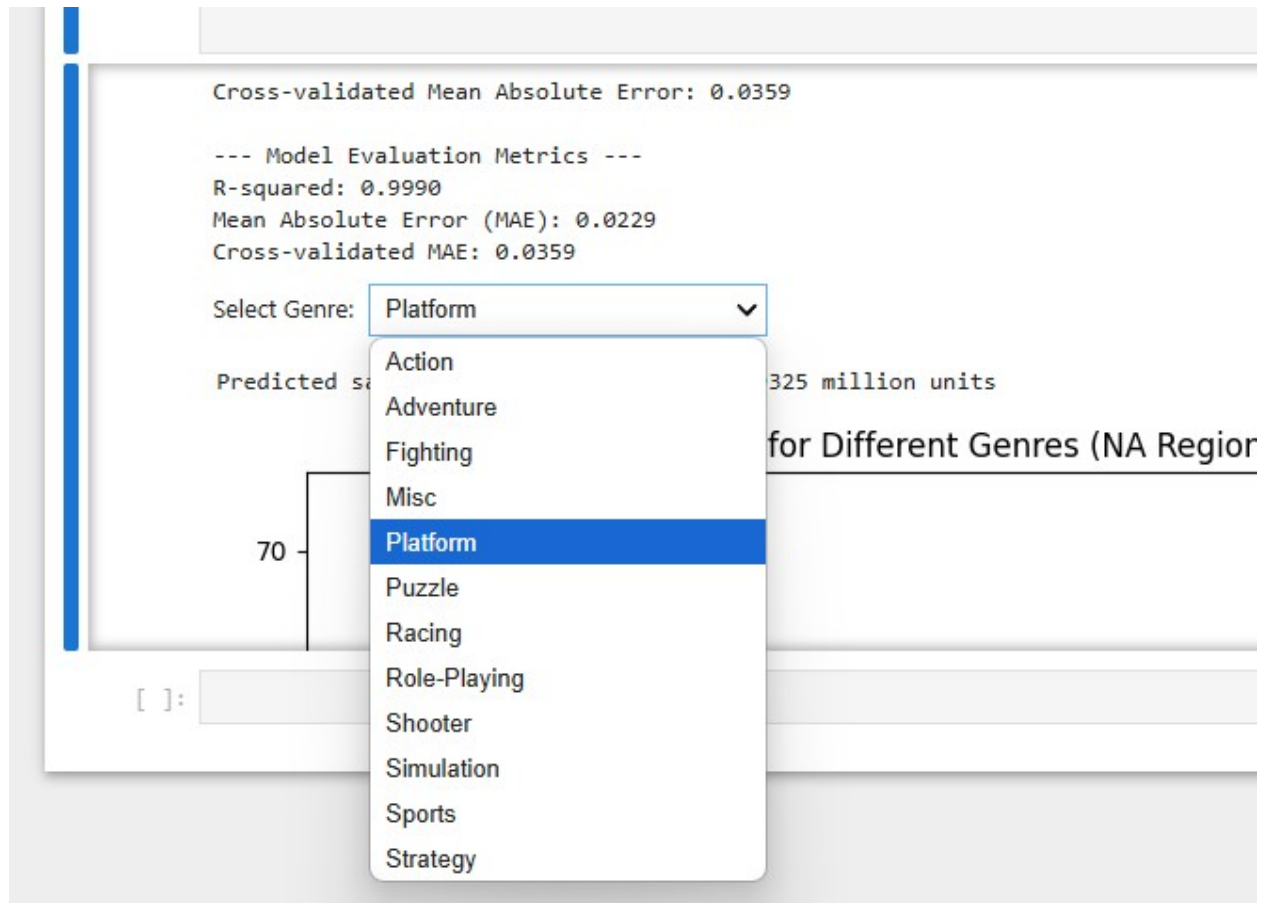
Select the Python 3 Kernel from the dropdown menu and click ‘Select’.



10. Scroll to the bottom of the code to see the output.

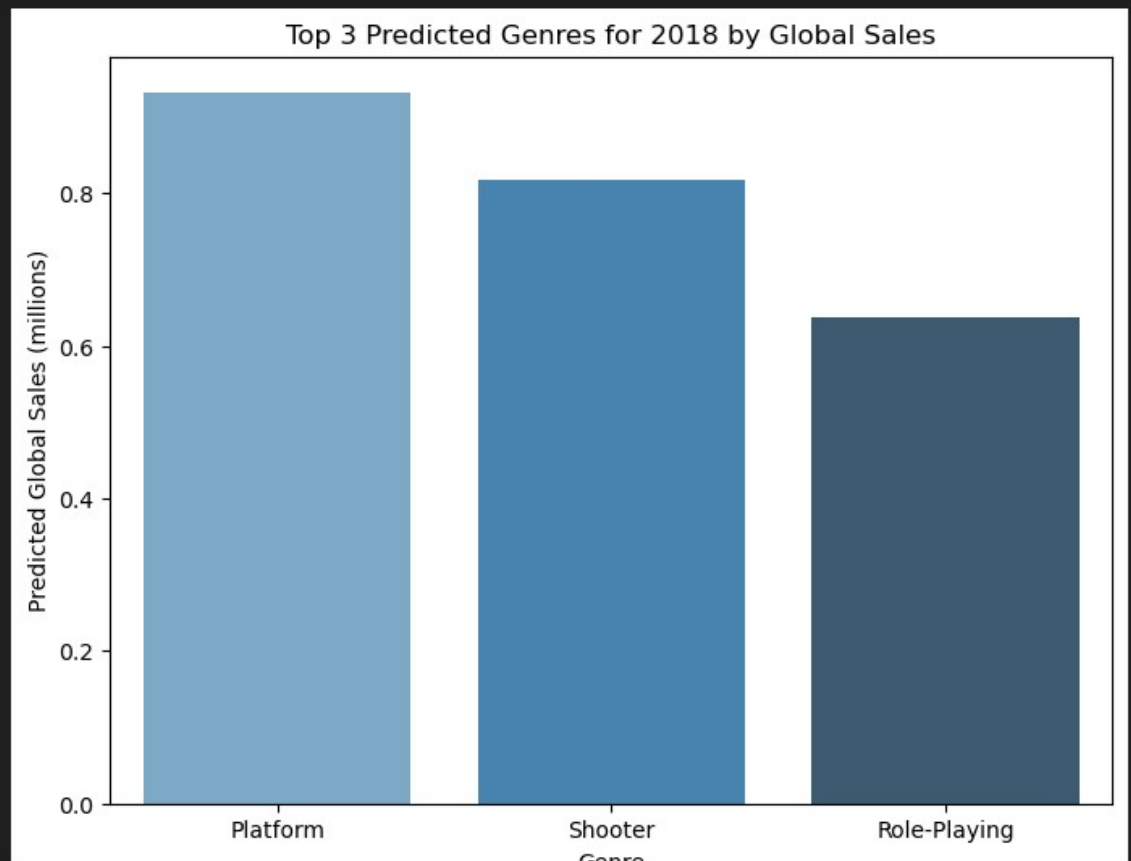
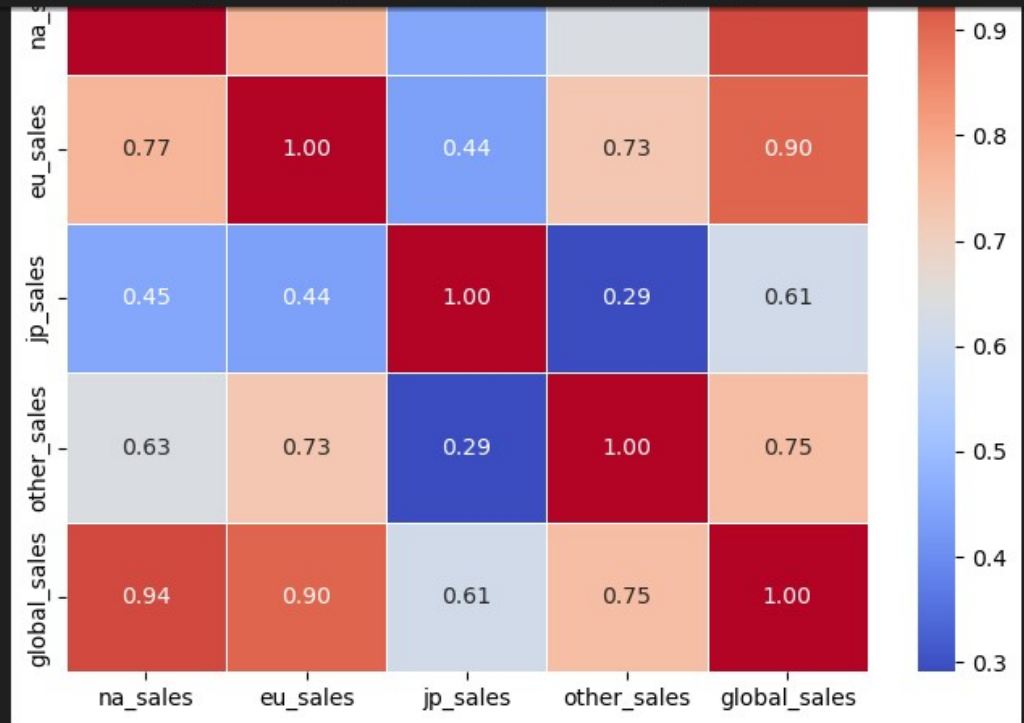


11. After running the script, you can select a genre from the dropdown menu to see the predicted sales for 2018.



12. The program will also display the visualizations of the data for review as part of the output.





# Reference Page

Pederson, U.T. (2021). *Video Game Sales*. Kaggle

<https://www.kaggle.com/datasets/ulrikthgepedersen/video-games-sales>