PREDICTING VIDEO GAME SALES

IVAN CHERNIAVSKYI
MUNEEB SAMAD
ROBERT MOLENDA
RAHEEM SULTAN

PROBLEM STATEMENT

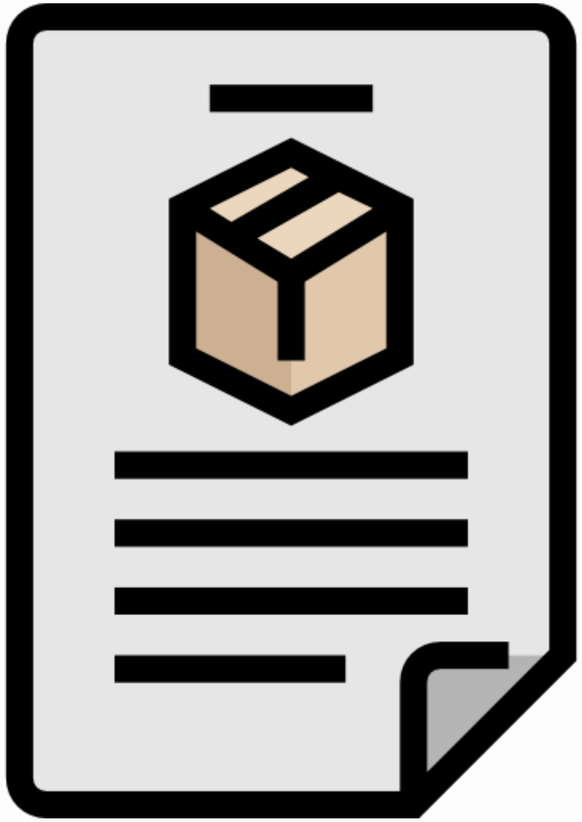
For our project we decided to take a look at the video game industry, focusing on the task of predicting future sales. Accurate sales forecasting is essential for developers, publishers and marketers to make informed decisions about game development and marketing strategies.

Our goal is to develop a machine-learning model that predicts future video game sales based on historical data. We'll analyze key factors such as genre, platform, region, and historical sales to create a predictive model that provides valuable insights.

DATA DESCRIPTION

Our data source was obtained through Kaggle. It contains data concerning video games such as name, platform, year of release, genre, and sales figures to allow us to solve for future forecasting given it is historical data. The dataset ranges between 1980 and 2020. It contained 11 columns of categorical and numerical information concerning video games.

For future feature engineering we added two additional tables. The first table contains world GDP indicator for years 1960 - 2023. The second table contains the average % of the population ageing 15-64 by year. The age 15-64 is considered main target audience for games' consumption. Both datasets were acquired from World Bank website.



SOLUTION OUTLINE

Our source provided us with a CSV file through Kaggle and we ensured the data was clean. We then furthered our approach using Python's Pandas library and extracting from the dataset. The model that we created allowed us to drop and entertain relevant variables that we needed for our project. Then we were able to perform data analysis using libraries like Matplotlib and Plotly for visualizations.

kaggle → pandas → SQL → TensorFlow, scikit learn, matplotlib, plotly, seaborn

DATA PREPARATION

Loading VGSales Data

Loading CSVs into DataFrame

Cleaning Data

Checking for nulls to handle for missing values, filling missing values

Data Transformation

Converting variable types
Grouping and calculating averages

Storing Data

Storing Video Game Sales data to SQLite Database

Loading GDP Data

Transpose data format
Calculate annual GDP sum
Save to SQLite database

Loading Population Data

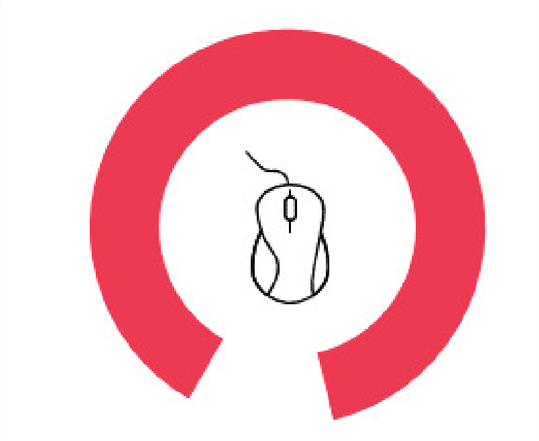
Drop unneeded columns
Transpose data format and calculate mean population
Save to SQLite database

Statistical Analysis

Year

The mean year of our data set is 2006.40, with a standard deviation of 5.82

Global Sales



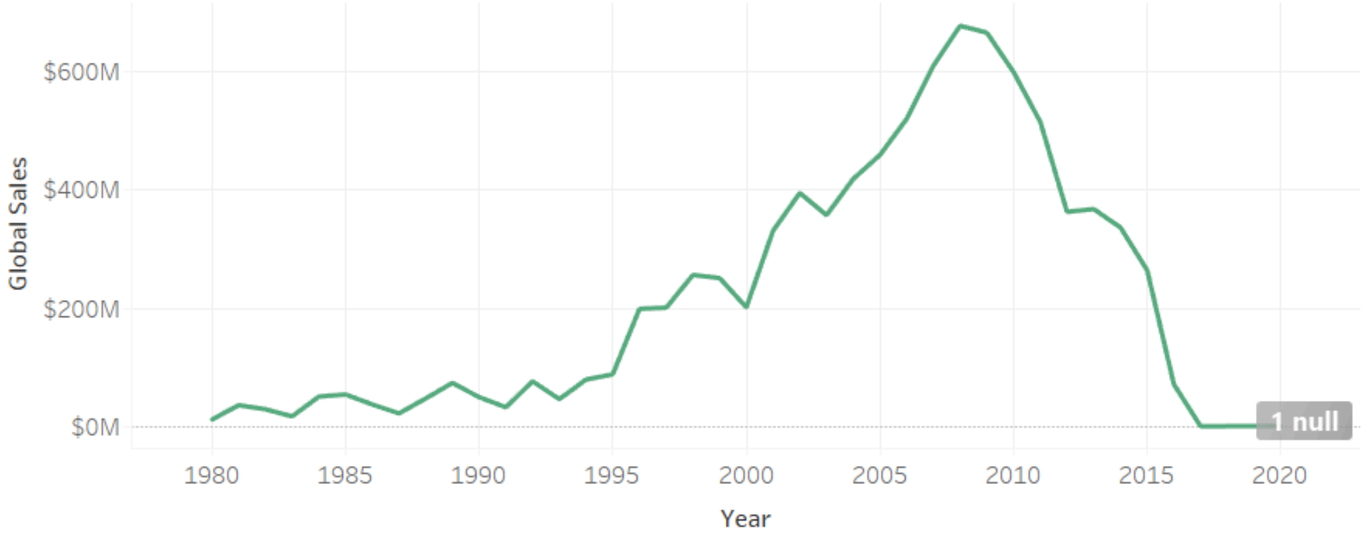
The mean figure for global sales is $540,232

Outliers

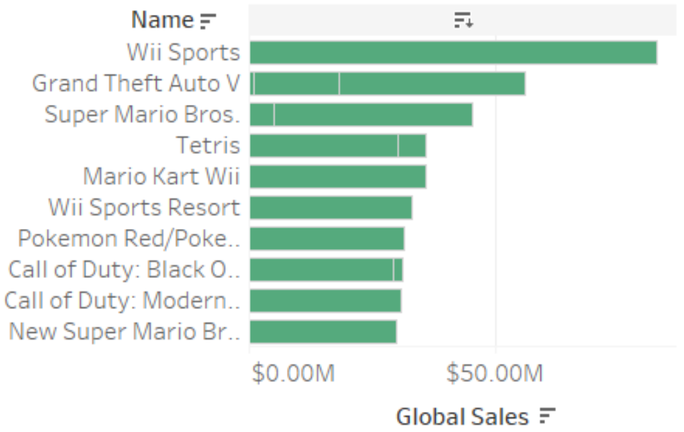
Wii Sports was a major outlier in the dataset, selling $82.74M.

Video Games Sales Dashboard

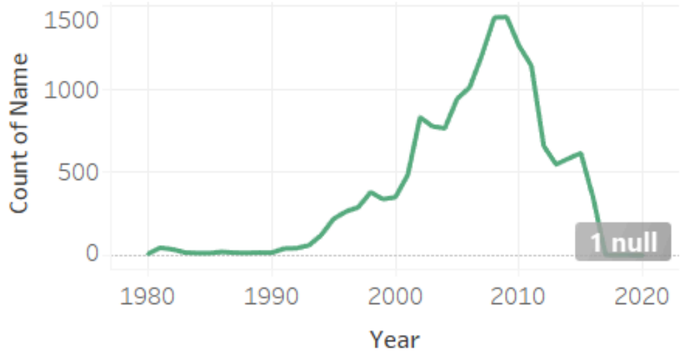
Global Sales Per Year



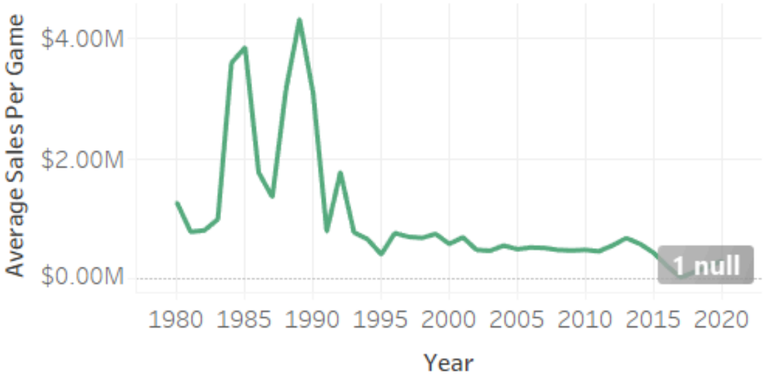
Top 10 Best Selling Games



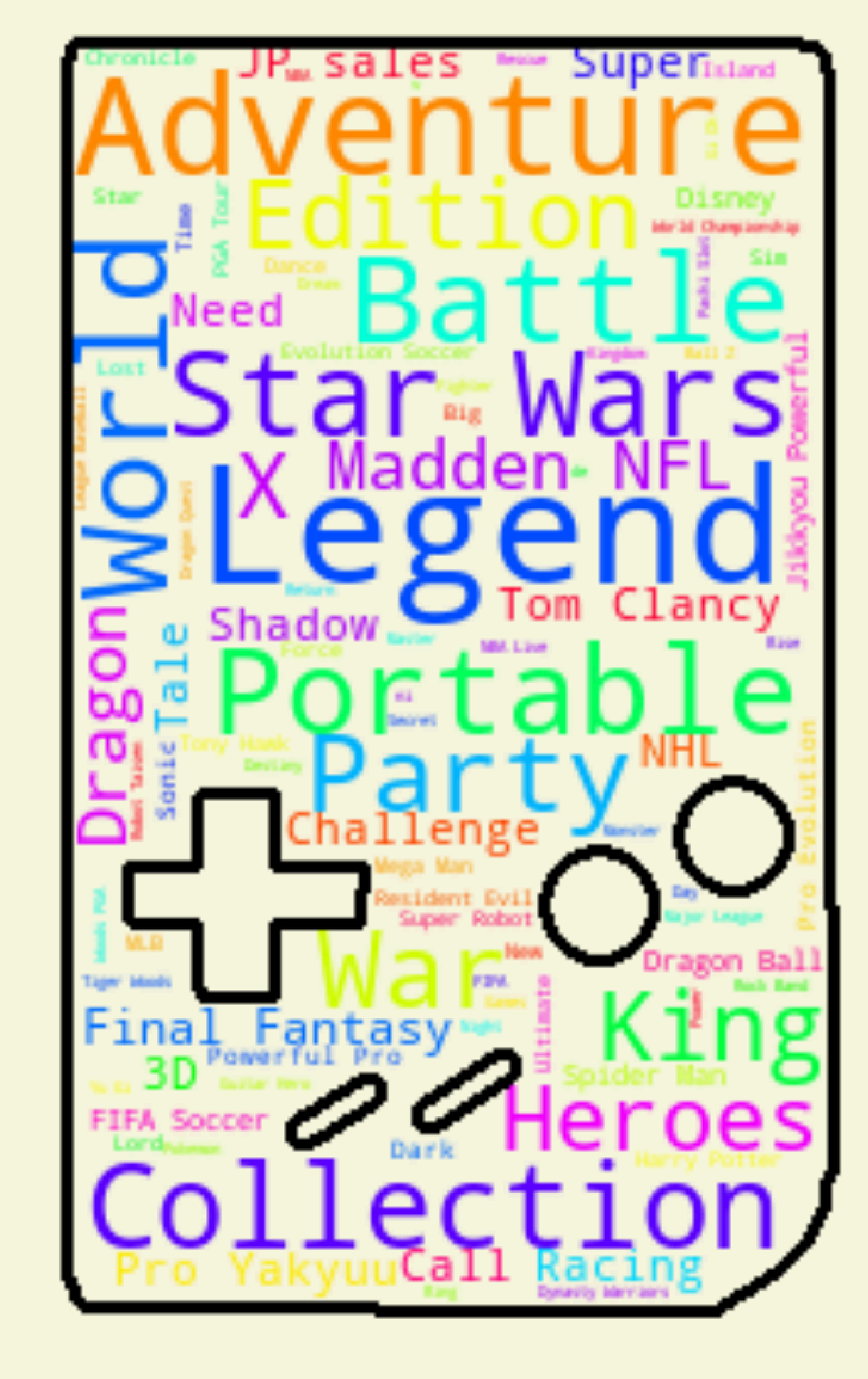
Releases Per Year



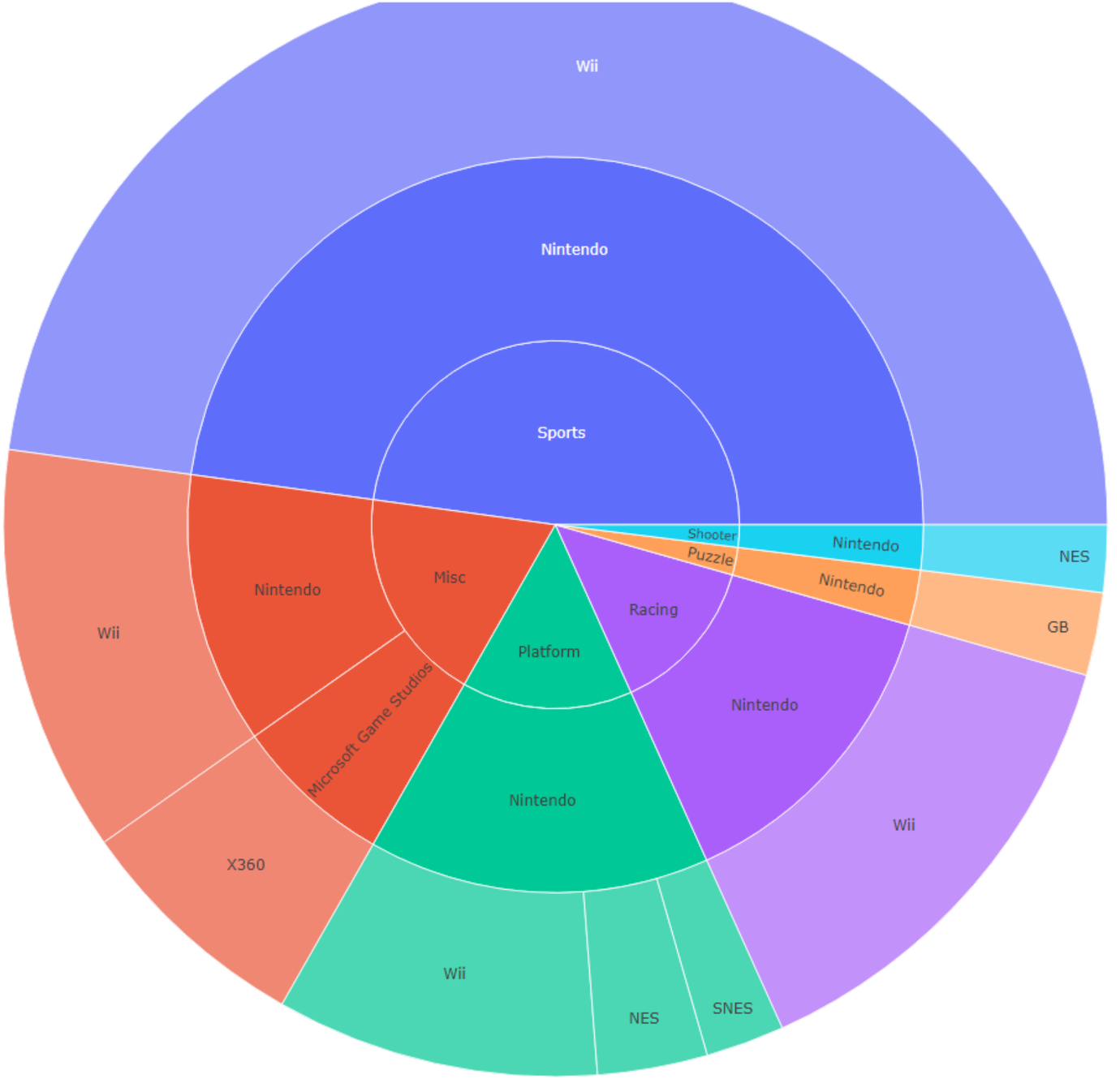
Average Sales Per Release

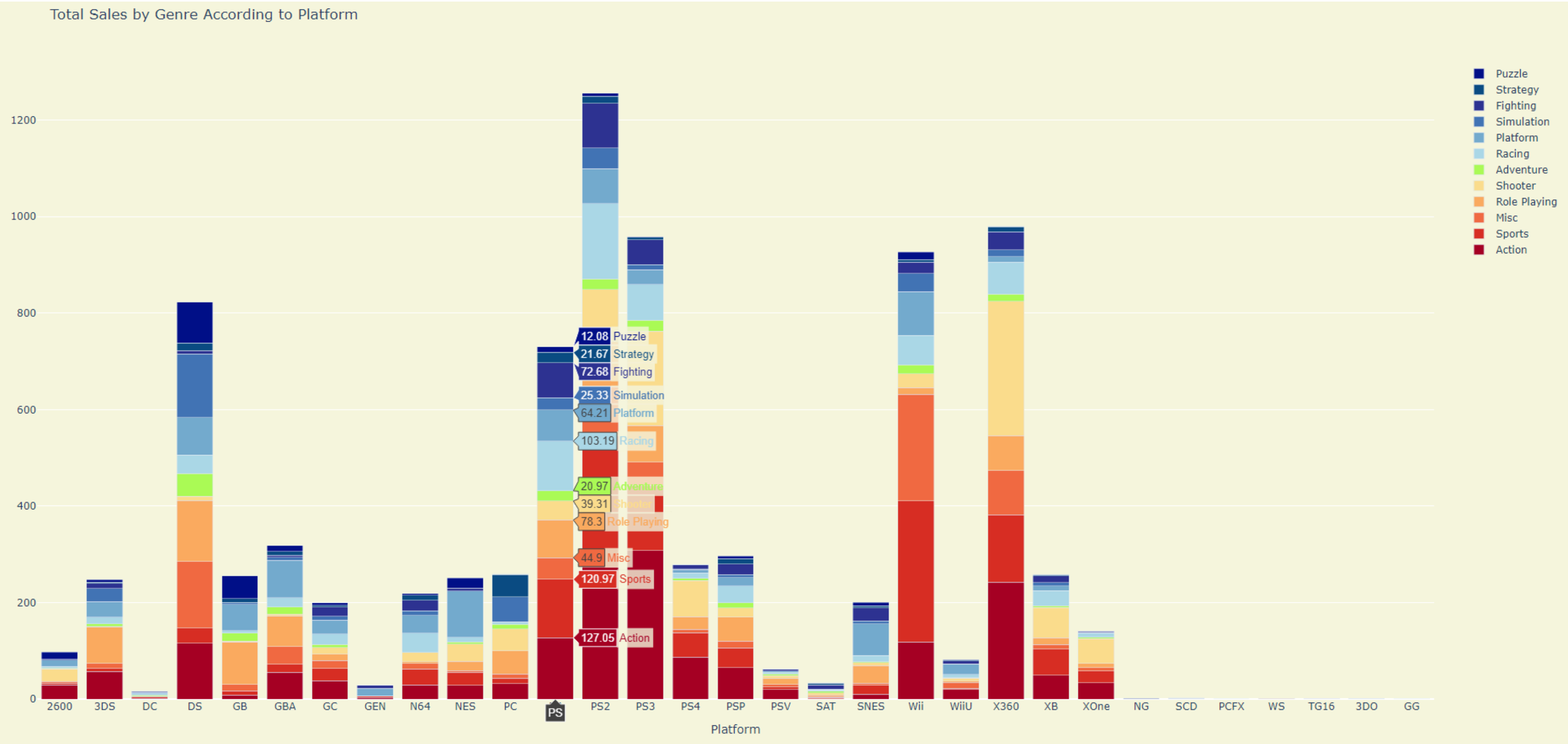


Top 10 Platforms by Sales

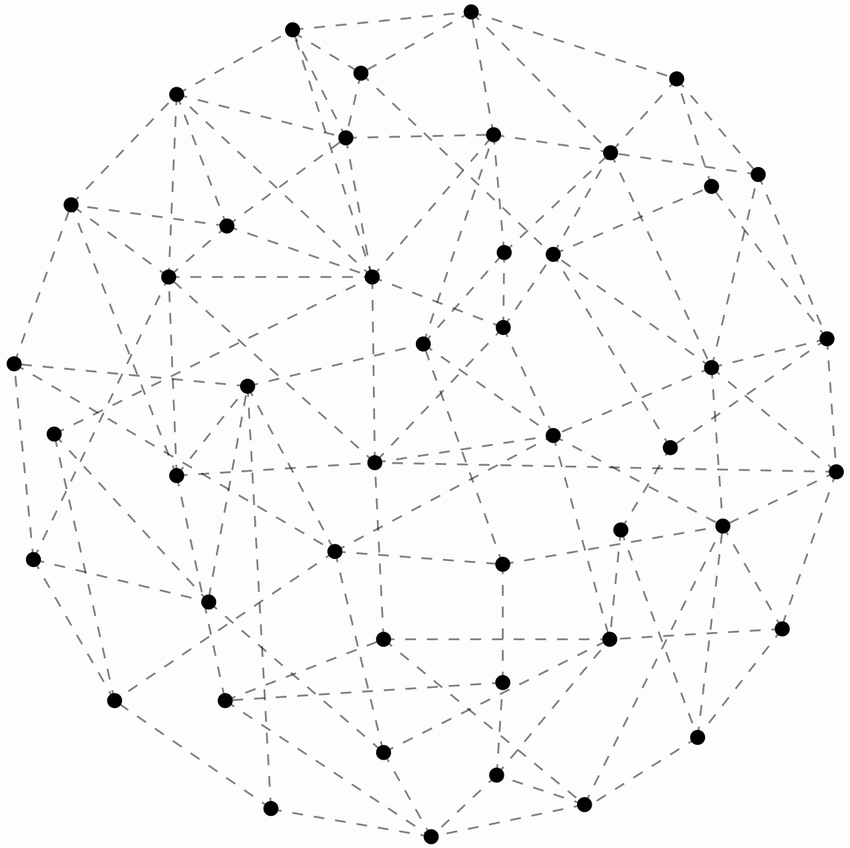

Chronicle
JP sales
Super
Island
Adventure
Star
Edition
Disney
World Championship
Time
PGA Tour
Dance
Sim
World
Need
Battle
Evolution Soccer
Lost
Star Wars
Big
Powerful
X
Madden NFL
Legend
Jikkyou
Tom Clancy
Shadow
Dragon
Tale
Portable
Sonic
Tony Hawk
Party
NHL
Challenge
Mega Man
Resident Evil
Super Robot
MLB
War
Dragon Ball
Tiger Woods
Final Fantasy
Ultimate
King
3D
Powerful Pro
Spider Man
FIFA Soccer
Heroes
Lord
Dark
Harry Potter
Collection
Pro Yakyuu
Call
Racing
Dynasty Warriors

Top Selling Platforms




Total Sales by Genre According to Platform
Puzzle
Strategy
Fighting
Simulation
Platform
Racing
Adventure
Shooter
Role Playing
Misc
Sports
Action
1200
1000
800
600
400
200
0
12.08 Puzzle
21.67 Strategy
72.68 Fighting
25.33 Simulation
64.21 Platform
103.19 Racing
20.97 Adventure
39.31 Shooter
78.3 Role Playing
44.9 Misc
120.97 Sports
127.05 Action
2600
3DS
DC
DS
GB
GBA
GC
GEN
N64
NES
PC
PS
PS2
PS3
PS4
PSP
PSV
SAT
SNES
Wii
WiiU
X360
XB
XOne
NG
SCD
PCFX
WS
TG16
3DO
GG
Platform

MODELLING EXPERIMENT DESIGN



Objective:

To predict global video game sales using various factors such as regional sales, GDP, and population data.

Datasets:

Video game sales data, population statistics, and GDP data, stored in an SQL database.

Approach:

Integrate data from multiple sources, perform feature engineering, and test various regression models to identify the most effective predictor of global sales.

Metrics:

- **Mean Squared Error (MSE)**: Measures the average squared difference between actual and predicted values, indicating the model's prediction accuracy.
- **R-squared (R^2)**: Represents the proportion of variance in the dependent variable that is predictable from the independent variables, showing the model's explanatory power.

FEATURES AND FEATURE ENGINEERING

Feature Engineering

Yearly Averages: Helps to account for seasonal variations and provides a more stable predictor of future sales by focusing on long-term trends.

Shifting Averages: Assumes that past regional sales trends are indicative of future sales, making it a straightforward approach to incorporate historical performance.

```
games_avg_genre = games.groupby(by = ["Genre", "Year"], as_index=False)
["Global_Sales"].mean()
games_avg_genre["Year"]=games_avg_genre["Year"] + 1
```

Log Transformation: Applied to Global sales, population percentage, and GDP. Log transformation helps to manage extreme values and skewed distributions by compressing the range of values.

```
games["Global_Sales"] = np.log(games["Global_Sales"])
```

Label Encoding: Label encoding provides a way to handle categorical features by converting them into a format suitable for the models.

```
le = LabelEncoder()
games.Platform = le.fit_transform(games.Platform)
```

Standard Scaling: Standard scaling ensures that features are on the same scale.

```
scaler = StandardScaler()
```

```
# Creating a dataframe with the freaquency of used words for games
from collections import Counter
word_counts = Counter(filtered_words)
word_counts_df = pd.DataFrame.from_dict(word_counts, orient='index', columns=['Frequency'])
word_counts_df = word_counts_df.sort_values(by='Frequency', ascending=False)
top_5_percent_threshold = np.percentile(word_counts_df['Frequency'], 95)
top_5_percent_words = word_counts_df[word_counts_df['Frequency'] >= top_5_percent_threshold]
words_list =list(top_5_percent_words.index)
```

```
def world_counter(game_name):
    # Split the game name into a list of words
    words = game_name.split()
    # Initialize a counter to keep track of how many words match the words in words_list
    counter = 0
    # Loop through each word in the list of words
    for w in words:
        # Check if the word is present in the predefined list called words_list
        if w in words_list:
            # If the word is found in words_list, increment the counter by 1
            counter +=1
    return counter
```

```
# Add a "Word_Counter" column by applying the world_counter function to
games["Word_Counter"] = games.apply(lambda x: world_counter(x['Name']),

# Creating  avarege sales by Word_Counter column
word_sales_mean = games.groupby('Word_Counter')['Global_Sales'].mean()
games['Word_Counter_Sales'] = games['Word_Counter'].map(word_sales_mean
```

MODELS INSIGHTS RESULTS

Linear Regression

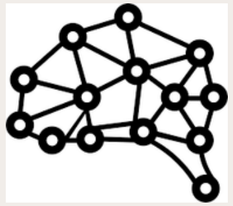
LinearRegression()

Mean Squared Error (MSE): 1.54

R-squared (R^2): 0.27

R^2 Score: The model yielded a low R^2 score, indicating that the model explains very little of the variance in the global sales data.

MSE: The Mean Squared Error was relatively high, pointing to significant errors in the model's predictions.



Decision Tree

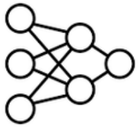
DecisionTreeRegressor()

Mean Squared Error (MSE): 2.33

R-squared (R^2): -0.09

R^2 Score: Although slightly better than linear regression, the R^2 score remained unsatisfactory, implying poor predictive accuracy.

MSE: The decision tree also resulted in a high MSE, reflecting large prediction errors.



Neural Network

Tuned using Hyperband with varying layers and activation functions.

Mean Squared Error (MSE): 5.18

R^2 Score: The TensorFlow model also failed to achieve a satisfactory R^2 score, suggesting that even complex models struggle to capture the underlying patterns.

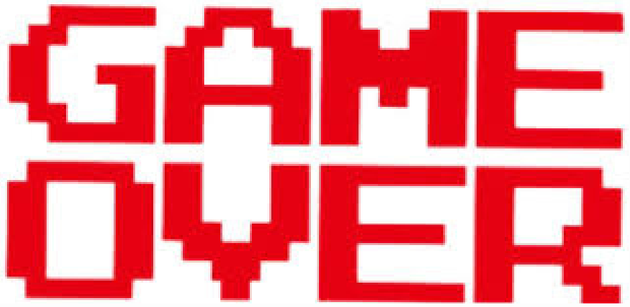
MSE: The high MSE confirmed that the model's predictions were still off by a significant margin.

INSIGHTS REPORTING

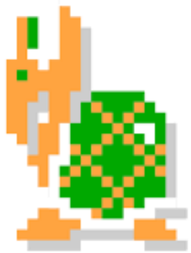
Results

R^2 Scores: The R^2 scores were low across all models, indicating that the models failed to adequately explain the variance in global sales. This suggests potential inadequacies in the features selected or indicates that the relationships between the features and sales are more complex than the models could capture.

MSE Values: High Mean Squared Error (MSE) values were observed in all models, confirming the poor predictive performance. This indicates large discrepancies between the predicted and actual sales figures.



Why Unsuccessful?



Data Quality: The performance issues may stem from poor data quality or insufficient feature engineering. Inconsistent data, improper encoding, or irrelevant features can all negatively impact model accuracy.

Model Complexity: The simplicity of linear regression and decision trees may prevent them from capturing the complex relationships in the data. Consideration of polynomial features, interaction terms, or more sophisticated models like ensemble methods might be necessary.

Neural Network Tuning: Despite using Hyperband for tuning, neural networks require careful architecture design. Additional tuning, better architecture selection, or more data might be needed to enhance the model's performance.

CONSIDERATIONS FOR FUTURE & KEY LEARNINGS

Problem Transformation

Instead of predicting the exact numerical value of global video game sales, the problem could be transformed into a categorical classification problem. For example, you could categorize sales into bins such as "Low", "Medium", and "High" based on predefined thresholds.

More relevant features

Collecting data like marketing expenses and consumer trust ratings can improve model accuracy by capturing more factors influencing game sales. To prevent multicollinearity, use feature selection techniques to ensure each feature provides unique information.

New data collection

Issues with recent data - our data set used figures that were scraped from VGChartz. As of 2018, VGChartz no longer releases estimates sales data for video game sales due the shift to digital purchases of games vs. retail, which were more producing estimates

Time for Models' Experimentation

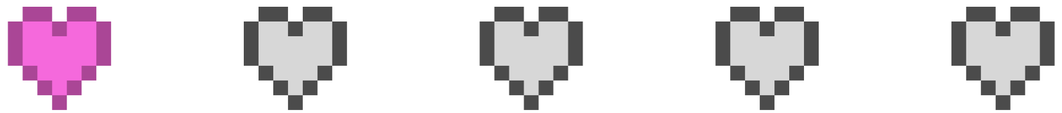
Due to time constraints, we couldn't explore other models, but options like Random Forest, Gradient Boosting Machines (GBM), and Support Vector Machines (SVM) could enhance accuracy and capture complex patterns. Additionally, K-Nearest Neighbors (KNN) might provide useful insights. Exploring these models could improve predictive performance and offer a deeper understanding of the data.

Forge your own path, not mirror others.

We discovered another modeling project using the same dataset that reported a 98% accuracy. However, this high accuracy was misleading because the model included a rank column that directly corresponded to the raw numbers, leading to artificially inflated performance. We set this accuracy as our target and invested significant time in feature engineering to improve our model, only to realize that the reported accuracy was not genuinely reflective of model quality.

MENU

THANKS FOR PLAYING!



GAME OVER