STEAM

DATA ANALYSIS

Prepared by
PRAFFUL SINGH

CONTENT

1. INTRODUCTION
2. DATA PREPARATION
3. DATA CLEANING
4. DATA VISUALIZATION
5. DATA INSIGHTS

Prafful Singh

DATA PREPARATION

- Importing Libraries and Loading the Dataset
- Converting Columns to Appropriate Data Types
- Data Overview and Missing Values
- Missing Values Analysis

Prafful Singh

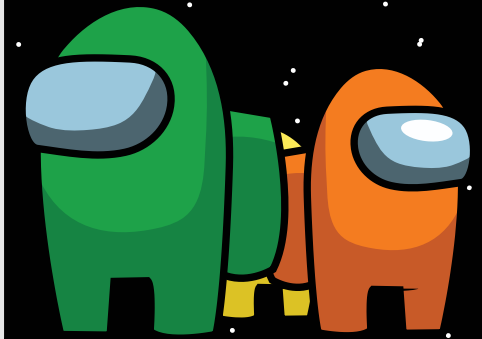
DATA CLEANING

- Handling Missing Values
- Handling Missing Values in 'About the Game' Column
- Handling Missing Values in 'Supported Languages,' 'Full Audio Languages,' and 'Reviews' Columns
- Creating OS Columns and Data Type Conversion
- Initial Data Exploration
- Cleaning Developers and Publishers Columns
- Handling Missing Data in 'Categories' Column
- Cleaning Tags and Screenshots Columns
- Removing Redundant Columns

Prafful Singh

DATA VISUALIZATION

- Most Downloaded Games
- Most Expensive Games
- Most Downloaded Games and Their Prices
- Prices Distribution
- Average Downloads for Free and Paid Games
- Most Supported Languages
- Average Downloads for Games with Support vs. Games without Support
- Average Downloads for Games by Operating System
- Top Rated Games by Positive and Negative Ratings
- Most Downloaded Games with Their Positive and Negative Ratings
- Relationship Between Number of Downloads and Ratings of Games
- Relationship Between Price and Ratings
- Top Developers
- Top Publishers
- Distribution of Games by Categories
- Top Genres
- Most Common Year in Which Games Were Produced
- Correlation Analysis of Game Metrics
- Standard Deviation of Average and Median Playtime for the Last Two Decades

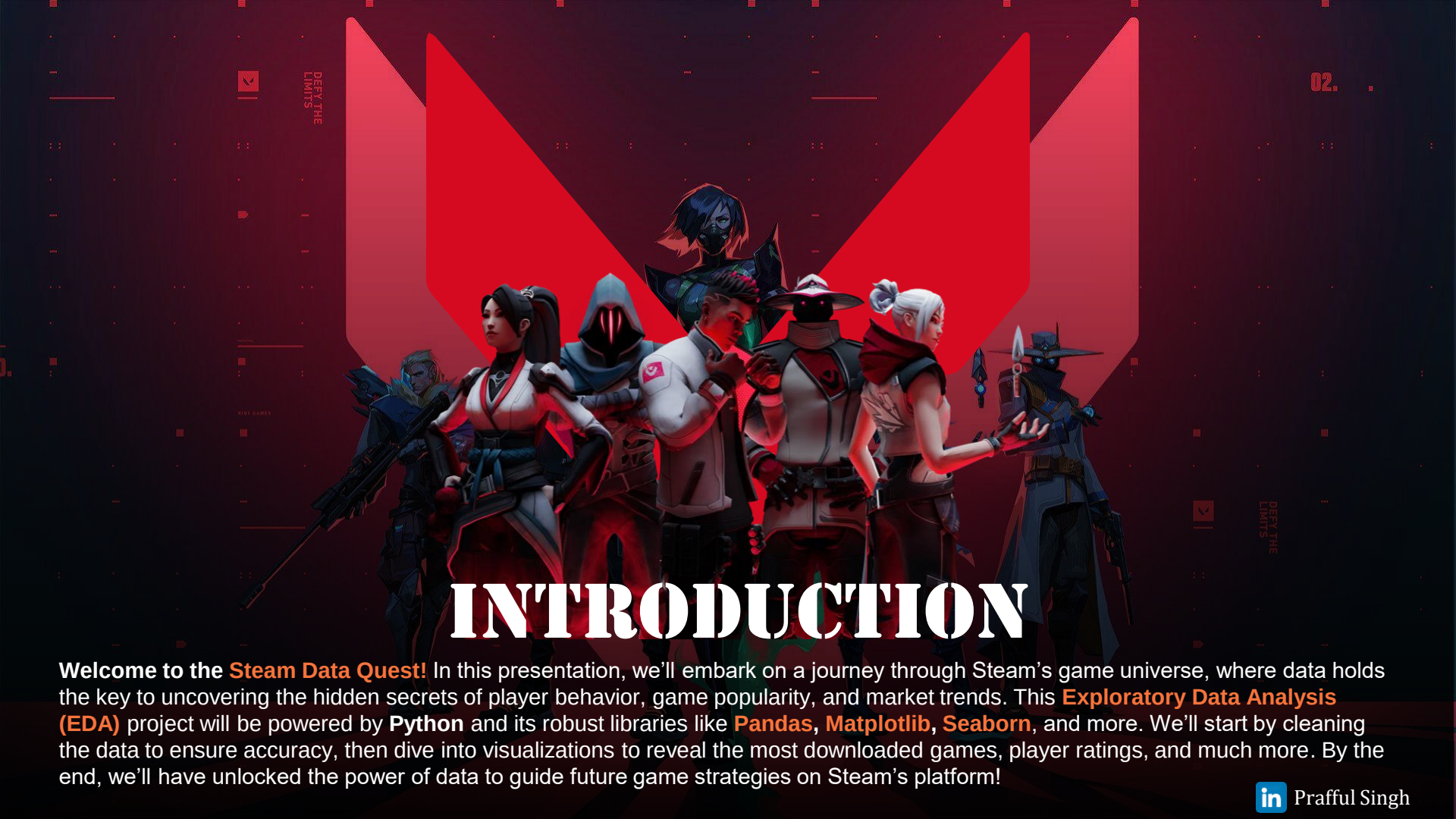


Prafful Singh

DATA INSIGHTS

- Business impact
- Recommendations
- Key Findings
- Conclusion

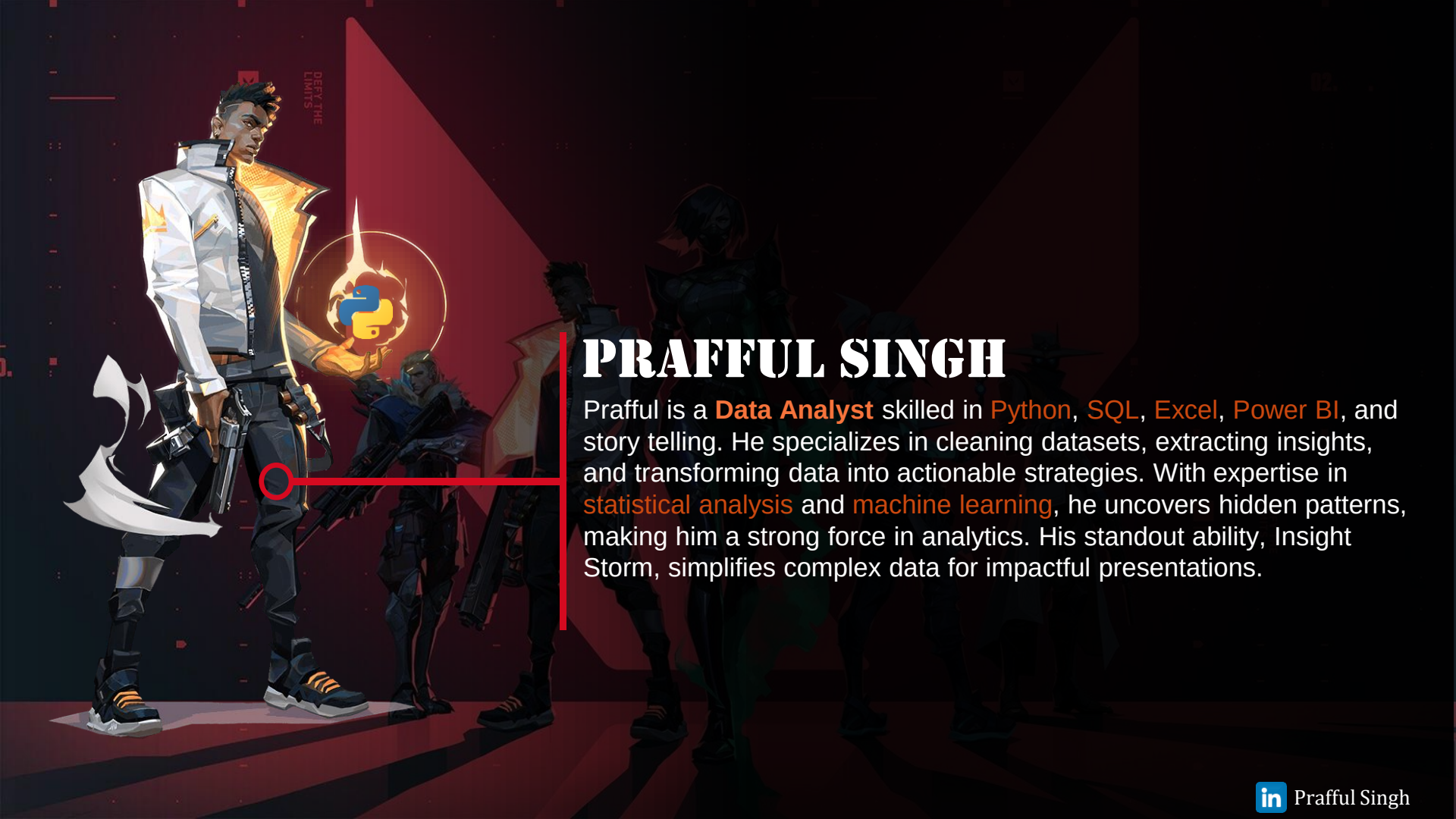
Prafful Singh



INTRODUCTION

Welcome to the Steam Data Quest! In this presentation, we'll embark on a journey through Steam's game universe, where data holds the key to uncovering the hidden secrets of player behavior, game popularity, and market trends. This **Exploratory Data Analysis (EDA)** project will be powered by **Python** and its robust libraries like **Pandas**, **Matplotlib**, **Seaborn**, and more. We'll start by cleaning the data to ensure accuracy, then dive into visualizations to reveal the most downloaded games, player ratings, and much more. By the end, we'll have unlocked the power of data to guide future game strategies on Steam's platform!

Prafful Singh



PRAFFUL SINGH

Prafful is a **Data Analyst** skilled in Python, SQL, Excel, Power BI, and story telling. He specializes in cleaning datasets, extracting insights, and transforming data into actionable strategies. With expertise in statistical analysis and machine learning, he uncovers hidden patterns, making him a strong force in analytics. His standout ability, Insight Storm, simplifies complex data for impactful presentations.

Prafful Singh

RICK GRIMES

Rick is the **Marketing Director for Steam**, skilled in promoting games and boosting player engagement. With a keen sense of market trends and player behavior, he creates campaigns that maximize game exposure. His expertise in community building and data-driven strategies ensures long-term success, and his signature move, Hype Surge, turns game releases into must-see events.

Prafful Singh

Importing Libraries and Loading the Dataset

Libraries Imported:

- **pandas:** A powerful data manipulation library, used to handle tabular data in DataFrames.
- **numpy:** Essential for numerical operations, such as array manipulation.
- **matplotlib.pyplot:** A popular library for creating static visualizations, such as plots and charts.
- **seaborn:** Built on top of Matplotlib, it provides a high-level interface for drawing attractive statistical graphics.
- **plotly.express:** Useful for creating interactive visualizations and plots.

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import plotly.express as px
```

Dataset:

- We have loaded the **Steam Games 2024** dataset, which contains a wide range of information about games available on Steam. The dataset is loaded using pandas.read_csv and is stored in a DataFrame (df).

```
df =
pd.read_csv("C:/Users/Prafful/Steam
Games 2024.csv")
```

Prafful Singh


Here’s a dataset we’ve collected from Steam games. Can you clean it up and extract some valuable insights for me to improve our marketing strategies?
Absolutely! Let’s start by cleaning the dataset to make sure it’s ready for analysis.
Also can you provide a quick overview of the dataset? Do we have any missing or incomplete data?
I’ll give you a summary of the data, highlighting any gaps or missing values that need to be cleaned up.
Prafful Singh

Converting Columns to Appropriate Data Types

In this step, I ensured that each column in the dataset had the correct data type to allow for accurate analysis and manipulation.

Purpose:

Data type conversion ensures that columns have the correct format for future operations like statistical analysis, visualization, and modeling. It also helps prevent errors during data processing.

Data type Conversion

```
df['AppID'] = pd.to_numeric(df['AppID'], errors='coerce')
df['Release date'] = pd.to_datetime(df['Release date'], errors='coerce')
df['Estimated owners'] = df['Estimated owners'].astype(str)
df['Peak CCU'] = pd.to_numeric(df['Peak CCU'], errors='coerce')
df['Required age'] = pd.to_numeric(df['Required age'], errors='coerce')
df['Price'] = pd.to_numeric(df['Price'], errors='coerce')
df['Discount'] = pd.to_numeric(df['Discount'], errors='coerce')
df['DLC count'] = pd.to_numeric(df['DLC count'], errors='coerce')
df['Windows'] = df['Windows'].astype(bool)
df['Mac'] = df['Mac'].astype(bool)
df['Linux'] = df['Linux'].astype(bool)
df['Metacritic score'] = pd.to_numeric(df['Metacritic score'], errors='coerce')
df['User score'] = pd.to_numeric(df['User score'], errors='coerce')
df['Positive'] = pd.to_numeric(df['Positive'], errors='coerce')
df['Negative'] = pd.to_numeric(df['Negative'], errors='coerce')
df['Achievements'] = pd.to_numeric(df['Achievements'], errors='coerce')
df['Recommendations'] = pd.to_numeric(df['Recommendations'], errors='coerce')
df['Average playtime forever'] = pd.to_numeric(df['Average playtime forever'],
errors='coerce')
df['Average playtime two weeks'] = pd.to_numeric(df['Average playtime two weeks'],
errors='coerce')
df['Median playtime forever'] = pd.to_numeric(df['Median playtime forever'],
errors='coerce')
df['Median playtime two weeks'] = pd.to_numeric(df['Median playtime two weeks'],
errors='coerce')
```

Prafful Singh

Data Overview and Missing Values

(df.info()): This command provides a detailed summary of the dataset, including the number of entries, the data types of each column, and how many non-null values are present.

df.info()

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 97428 entries, 0 to 97427
Data columns (total 40 columns):
 #   Column                      Non-Null Count  Dtype
---  ------                      --------------  -----
 0   AppID                       97410 non-null  float64
 1   Name                        97412 non-null  object
 2   Release date                97279 non-null  datetime64[ns]
 3   Estimated owners            97428 non-null  object
 4   Peak CCU                    97410 non-null  float64
 5   Required age                97410 non-null  float64
 6   Price                       97410 non-null  float64
 7   Discount                    97410 non-null  float64
 8   DLC count                   97410 non-null  float64
 9   About the game              92537 non-null  object
 10  Supported languages         97410 non-null  object
 11  Full audio languages        97410 non-null  object
 12  Reviews                     10132 non-null  object
 13  Header image                97409 non-null  object
 14  Website                     42745 non-null  object
 15  Support url                 45905 non-null  object
 16  Support email               81374 non-null  object
 17  Windows                     97428 non-null  bool
 18  Mac                         97428 non-null  bool
 19  Linux                       97428 non-null  bool
 20  Metacritic score            97403 non-null  float64
 21  Metacritic url              3961 non-null   object
 22  User score                  97404 non-null  float64
 23  Positive                    97404 non-null  float64
 24  Negative                    97403 non-null  float64
 25  Score rank                  52 non-null     object
 26  Achievements                97402 non-null  float64
 27  Recommendations             97402 non-null  float64
 28  Notes                       15477 non-null  object
 29  Average playtime forever    97402 non-null  float64
 30  Average playtime two weeks  97402 non-null  float64
 31  Median playtime forever     97402 non-null  float64
 32  Median playtime two weeks   97402 non-null  float64
 33  Developers                  92531 non-null  object
 34  Publishers                  92237 non-null  object
 35  Categories                  91495 non-null  object
 36  Genres                      92567 non-null  object
 37  Tags                        67653 non-null  object
 38  Screenshots                 94513 non-null  object
 39  Movies                      89520 non-null  object
dtypes: bool(3), datetime64[ns](1), float64(16),
object(20)
memory usage: 27.8+ MB
```

Prafful Singh

Missing Values Analysis

```
df.isna().sum()
```

This step calculates the number of missing (NaN) values in each column, giving a clear view of where data is incomplete. Identifying missing data is essential for deciding how to handle it, whether through imputation, removal, or other data cleaning methods.

df.isna().sum()

AppID	18
Name	16
Release date	149
Estimated owners	0
Peak CCU	18
Required age	18
Price	18
Discount	18
DLC count	18
About the game	4891
Supported languages	18
Full audio languages	18
Reviews	87296
Header image	19
Website	54683
Support url	51523
Support email	16054
Windows	0
Mac	0
Linux	0
Metacritic score	25
Metacritic url	93467
User score	24
Positive	24
Negative	25
Score rank	97376
Achievements	26
Recommendations	26
Notes	81951
Average playtime forever	26
Average playtime two weeks	26
Median playtime forever	26
Median playtime two weeks	26
Developers	4897
Publishers	5191
Categories	5933
Genres	4861
Tags	29775
Screenshots	2915
Movies	7908

dtype: int64

Prafful Singh

Data cleaning

Data cleaning is a crucial step in ensuring the dataset is accurate, consistent, and free from errors. This process involves handling missing values, correcting data types, and addressing inconsistencies. By cleaning the data, we enhance the quality and reliability of the analysis, leading to more accurate insights.

Why Data Cleaning Matters:

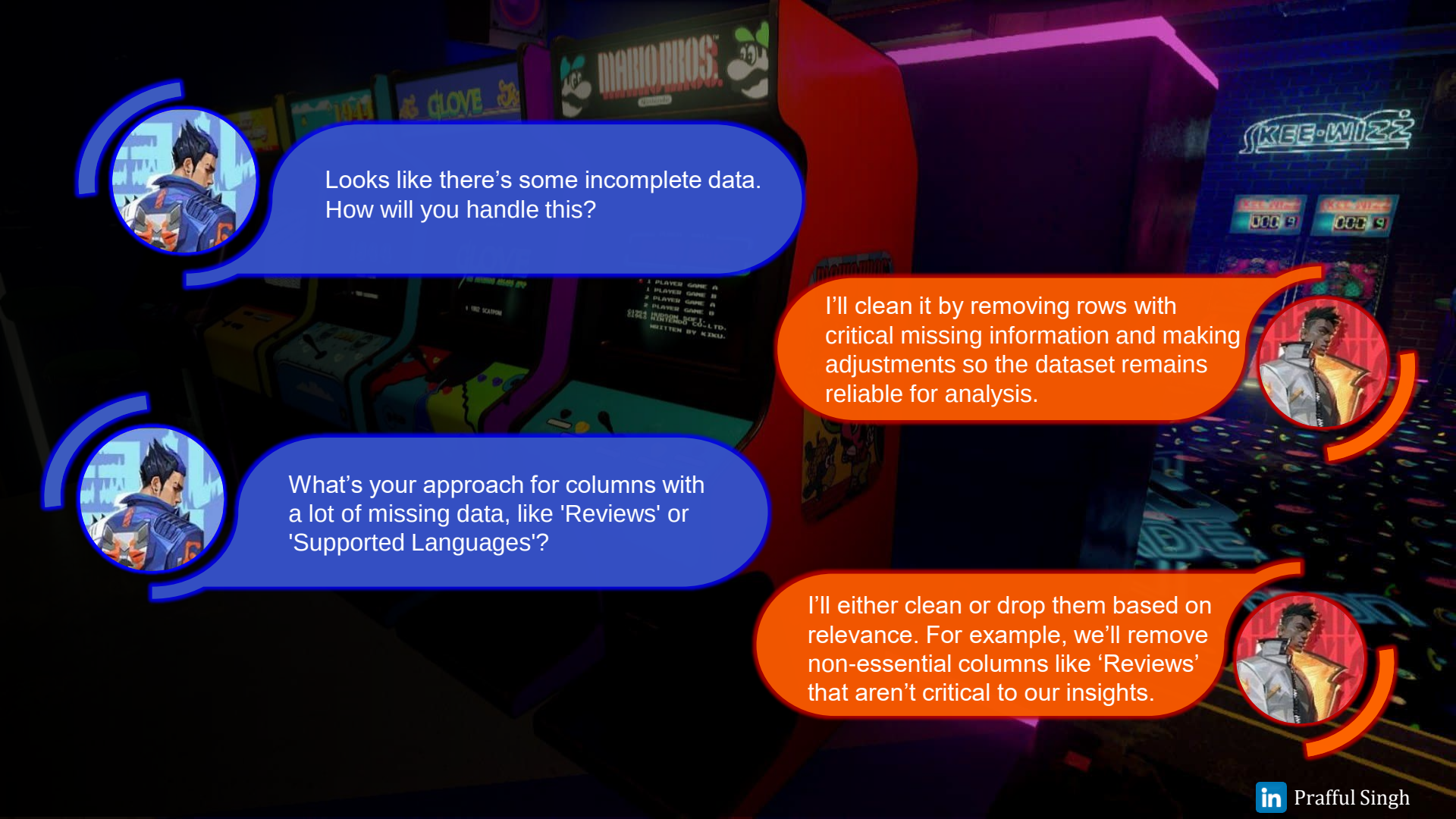
- Ensures data consistency and accuracy.
- Helps prevent skewed analysis due to erroneous or incomplete data.
- Improves model performance and overall data quality.

As a best practice, I created a copy of the original dataset to maintain data integrity during cleaning.

creating a copy

```
df_copy = df.copy()
```

Prafful Singh


Looks like there’s some incomplete data. How will you handle this?
I’ll clean it by removing rows with critical missing information and making adjustments so the dataset remains reliable for analysis.
What’s your approach for columns with a lot of missing data, like 'Reviews' or 'Supported Languages'?
I’ll either clean or drop them based on relevance. For example, we’ll remove non-essential columns like ‘Reviews’ that aren’t critical to our insights.
Prafful Singh

Handling Missing Values

Dealing with missing values is a critical aspect of data cleaning. In this step, I focused on removing rows with missing data in key columns, ensuring the dataset remains reliable for analysis.

AppID Column:
I identified 18 rows with missing values in the 'AppID' column and removed them, as this column serves as a unique identifier, making it crucial for every row.

Name Column:
6 rows with missing values in the 'Name' column were dropped, as the game names are essential for identification and analysis.

Release Date Column:
131 rows had missing 'Release Date' values. Since this accounts for only a small fraction (0.14%) of the dataset, I dropped these rows. Filling them with estimates wouldn't provide meaningful insights, and removing them keeps the dataset clean and accurate.

Dealing with null values in AppID column

```
df = df.dropna(subset=['AppID'])
```

Dealing with null values in Name column

```
df = df.drop(df.loc[df['Name'].isna()].index)
```

Dealing with null values in Release date column

```
df.dropna(subset=['Release date'],
inplace=True)
```

Prafful Singh

Handling Missing Values in 'About the Game' Column

analyzing 'About the game' column

```
df.loc[df['About the game'].isna(),"Name"]
```

output

```
105                    æ°ª é£Žè·ä¹‹æ˜ Playtest
180                       Burial Stone Playtest
214                   Emperial Knights Playtest
220                      Slotracers VR Playtest
291        Pirates of the Asteroid Belt Playtest
                           ...
97381                    Feed The Gods Playtest
97403                The Invading Dark Playtest
97405     ä¸§å°¸å¡¿ç‰Œï¼šçº¿Ÿå˜å¯¹å†³ Playtest
97410                  Karate Survivor Playtest
97422                          Carbon Playtest
Name: Name, Length: 4875, dtype: object
```

I identified patterns in the data and applied specific descriptions to games based on their type.

Here's the approach:

Playtest Games: Most missing descriptions were for Playtest games. I replaced these with:
"This is a playtest game and is the process by which a game designer tests a new game for bugs and design flaws before releasing it to market."

Playtest Games

```
df.loc[df["Name"].str.contains('Playtest'),'About the game'] =
"this is a playtest game is the process by which a game designer
tests a new game for bugs and design flaws before releasing it to
market."
```

Prafful Singh

Other Game Types: For remaining missing values (237 games left), I replaced them based on game types found in the **'Name'** column:

- **Beta**: "This game is beta and still under testing"
- **Alpha**: "This game is alpha and still under testing"
- **Test**: "This game is under testing"
- **SDK**: "Software Development Kit of the game"
- **Demo**: "This game is demo and still under testing"
- **Server**: "This is a server for the game"
- **Editor**: "This is an editor for the game"

Other Game Types

```
for index, row in df.iterrows():
    if(pd.isnull(row['About the game'])):
        if 'Beta' in row['Name']:
            df.at[index, 'About the game'] = 'this game is beta and still under testing'
        elif "Alpha" in row['Name']:
            df.at[index, 'About the game'] = 'this game is Alpha and still under testing'
        elif "beta" in row['Name']:
            df.at[index, 'About the game'] = 'this game is beta and still under testing'
        elif "BETA" in row['Name']:
            df.at[index, 'About the game'] = 'this game is beta and still under testing'
        elif "Test" in row['Name']:
            df.at[index, 'About the game'] = 'this game is and still under testing'
        elif "playtest" in row['Name']:
            df.at[index, 'About the game'] = 'this game is still under testing'
        elif "SDK" in row['Name']:
            df.at[index, 'About the game'] = 'Software Development Kit of the game'
        elif "Demo" in row['Name']:
            df.at[index, 'About the game'] = 'this game is Demo and still under testing'
        elif "Server" in row['Name']:
            df.at[index, 'About the game'] = 'this is a Server for the game'
        elif "Editor" in row['Name']:
            df.at[index, 'About the game'] = 'this is an Editor for the game'
        else:
            df.at[index, 'About the game'] = 'this game does not have a description'
```

Prafful Singh

Handling Missing Values in 'Supported languages', 'Full audio languages' Columns and 'Reviews' Column

Upon analysis, I found that:

- There were **8 rows** with missing data in either the **'Supported languages'** or **'Full audio languages'** columns.
- Further inspection revealed that these rows contained **many null values** across other columns as well.
- Since these rows had multiple missing values and these two columns aren't critical to the core analysis, I decided that **dropping these rows** would be the most efficient and cleanest solution.

Dropping rows

```
df = df.dropna(subset=['Supported languages', 'Full audio languages'])
```

Dropping Reviews column

```
df = df.drop('Reviews', axis=1)
```

The **'Reviews'** column had a significant number of missing values (87,165). Since:

- The proportion of missing data was too large to impute accurately.
- The **'Reviews'** column is not central to the objectives of this analysis.
- I chose to **drop the entire 'Reviews' column** from the dataset.

Prafful Singh

Creating OS Columns and Data Type Conversion

To streamline the dataset, I merged the **Windows**, **Mac**, and **Linux** columns into a single column, **'OS'**, representing the supported operating systems for each game. This provides a cleaner and more efficient structure for analysis.

creating new column OS

```
for index, row in df.iterrows():
    if row["Windows"] == True:
        df.at[index, 'OS'] = "Windows"
        if row["Mac"] == True:
            df.at[index, 'OS'] = "Windows, Mac"
            if row["Linux"] == True:
                df.at[index, 'OS'] = "Windows, Mac, Linux"
        elif row["Linux"] == True:
            df.at[index, 'OS'] = "Windows, Linux"
    elif row["Windows"] == False and row["Mac"] == True:
        df.at[index, 'OS'] = "Mac"
        if row["Linux"] == True:
            df.at[index, 'OS'] = "Windows, Linux"
    elif row["Windows"] == False and row["Mac"] == False
     and row["Linux"] == True:
        df.at[index, 'OS'] = "Linux"
```

Data Type Conversion

```
df['Positive'] = df['Positive'].astype(int)
df['Negative'] = df['Negative'].astype(int)
df['Achievements'] = df['Achievements'].astype(int)
df['Metacritic score'] = df['Metacritic score'].astype(int)
df['User score'] = df['User score'].astype(int)
df['Recommendations'] = df['Recommendations'].astype(int)
df['Average playtime forever'] = df['Average playtime
forever'].astype(int)
df['Average playtime two weeks'] = df['Average playtime two
weeks'].astype(int)
df['Median playtime forever'] = df['Median playtime
forever'].astype(int)
df['Median playtime two weeks'] = df['Median playtime two
weeks'].astype(int)
```

Additionally, I converted several columns to the **integer data type** after handling null values. This ensures that future calculations and visualizations work seamlessly.

Prafful Singh

Initial Data Exploration

In this step, I explored various game performance metrics and their distribution across the dataset. I focused on:

- **Metacritic Score:** Grouped games by their Metacritic scores and counted the number of games corresponding to each score.
- **User Score:** Similar to Metacritic scores, grouped games by user scores and analyzed the count.
- **Positive and Negative Reviews:** Grouped games based on the number of positive and negative reviews to gain insights into user feedback.
- **In-game Achievements:** Analyzed the number of in-game achievements for each game.
- **Playtime:** Investigated the distribution of both "Average playtime forever" and "Average playtime in two weeks" to see how much time players are spending on games.

EDA

```
df["Metacritic score"].unique()
df_mscore = df.groupby("Metacritic score").agg({"Name":"count"})
df["Metacritic url"].unique()
df["User score"].unique()
df_uscore = df.groupby("User score").agg({"Name":"count"})
df_score =
df.groupby(["Name","Positive","Negative"]).agg({"Name":"count"})
df["Positive"].unique()
df["Negative"].unique()
df["Achievements"].unique()
df_ach = df.groupby("Achievements").agg({"Name":"count"})
df["Recommendations"].unique()
df_avgf = df.groupby("Average playtime
forever").agg({"Name":"count"})
df_avgw = df.groupby("Average playtime two
weeks").agg({"Name":"count"})
```

This exploration provides an overview of how various metrics like scores, reviews, and playtime are distributed across the dataset.

Prafful Singh

Cleaning Developers and Publishers Columns

Initial Exploration: I first checked the uniqueness of developers and publishers and found some missing values in both columns.
There were:
- **350 games** where the developer was mentioned, but the publisher was not.
- **58 games** where the publisher was mentioned, but the developer was not.

Handling Missing Values:
- For the **350 games**, I assigned the same value from the 'Developers' column to the 'Publishers' column.
- For the **58 games**, I assigned the same value from the 'Publishers' column to the 'Developers' column.

Finding games with a developer but no publisher

```
count = 0
for index, row in df.iterrows():
    if(pd.isnull(row['Publishers'])):
        if(pd.isnull(row['Developers'])):
            continue
        else:
            print(row['Developers'])
            count += 1
print(count)
```

Finding games with a publisher but no developer

```
count = 0
for index, row in df.iterrows():
    if(pd.isnull(row['Developers'])):
        if(pd.isnull(row['Publishers'])):
            continue
        else:
            print(row['Publishers'])
            count += 1
print(count)
```

Assigning the publisher to be the same as the develop

```
for index, row in df.iterrows():
    if(pd.isnull(row['Publishers'])):
        if(pd.isnull(row['Developers'])):
            continue
        else:
            df.at[index, 'Publishers'] =
            df.at[index, 'Developers']
```

Assigning the developer to be the same as the publisher

```
for index, row in df.iterrows():
    if(pd.isnull(row['Developers'])):
        if(pd.isnull(row['Publishers'])):
            continue
        else:
            df.at[index, 'Developers'] =
            df.at[index, 'Publishers']
```

© 2022 MARVEL

Prafful Singh

Special Cases: I identified specific games (e.g., Beta, Demo, SDK) where both developers and publishers were not mentioned. I replaced missing values in these cases with context-appropriate descriptions such as:

- *"This game is Playtest and still under development."*
- *"This game is Alpha and still under testing; the developer is not mentioned."*
- This ensured that games in early stages of development were properly documented.

Similarly, for the 'Publishers' column, missing values were handled by applying the same logic used for 'Developers', with appropriate descriptions based on the game type.

This process helped in filling missing values systematically, retaining data consistency for both developers and publishers.

Special Cases

```
for index, row in df.iterrows():
    if(pd.isnull(row['Developers'])):
        if 'Playtest' in row['Name']:
            df.at[index, 'Developers'] = 'this game is Playtest and still under development'
        elif "Alpha" in row['Name']:
            df.at[index, 'Developers'] = 'this game is Alpha and still under testing the
developer is not mentioned'
        elif "beta" in row['Name']:
            df.at[index, 'Developers'] = 'this game is beta and still under testing the
developer is not mentioned'
        elif "BETA" in row['Name']:
            df.at[index, 'Developers'] = 'this game is beta and still under testing the
developer is not mentioned'
        elif "Test" in row['Name']:
            df.at[index, 'Developers'] = 'this game is and still under testing the developer
is not mentioned'
        elif "playtest" in row['Name']:
            df.at[index, 'Developers'] = 'this game is still under testing the developer is
not mentioned'
        elif "SDK" in row['Name']:
            df.at[index, 'Developers'] = 'Software Development Kit of the game the developer
is not mentioned'
        elif "Demo" in row['Name']:
            df.at[index, 'Developers'] = 'this game is Demo and still under testing the
developer is not mentioned'
        elif "Server" in row['Name']:
            df.at[index, 'Developers'] = 'this is a Server for the game the developer is not
mentioned'
        elif "Editor" in row['Name']:
            df.at[index, 'Developers'] = 'this is an Editor for the game the developer is not
mentioned'
        elif "Beta" in row['Name']:
            df.at[index, 'Developers'] = 'this game is beta and still under testing the
developer is not mentioned'
        else:
            df.at[index, 'Developers'] = 'No developer mentioned for this game'
```

Prafful Singh

Handling Missing Data in 'Categories' Column

- Explored the **Categories** column and identified recurring keywords such as *Playtest, Alpha, Beta, SDK,* and others.
- Based on the type of game, missing values were replaced with appropriate descriptions like:
- *'Beta game not playable'*, *'Alpha game not playable'*, *'Playtest game not playable'*, and similar labels for other game types.
- After applying this approach, very few games were left with missing categories.
- Any remaining games without category information were assigned the label *'no Category added'*.

A similar approach is applied to the **'Genres'**, **'Movies'** and **'Support email'** column

Giving Appropriate Description

```
for index, row in df.iterrows():
    if(pd.isnull(row['Categories'])):
        df.at[index, 'Categories'] = 'no Category added'
```

Giving Appropriate Description

```
for index, row in df.iterrows():
    if(pd.isnull(row['Categories'])):
        if 'Playtest' in row['Name']:
            df.at[index, 'Categories'] = 'Playtest game not playable'
        elif "Alpha" in row['Name']:
            df.at[index, 'Categories'] = 'Alpha game not playable'
        elif "beta" in row['Name']:
            df.at[index, 'Categories'] = 'Beta game not playable'
        elif "BETA" in row['Name']:
            df.at[index, 'Categories'] = 'Beta game not playable'
        elif "Test" in row['Name']:
            df.at[index, 'Categories'] = 'test game not playable'
        elif "playtest" in row['Name']:
            df.at[index, 'Categories'] = 'Playtest game not playable'
        elif "SDK" in row['Name']:
            df.at[index, 'Categories'] = 'Software Development Kit of
            the game not playable'
        elif "Demo" in row['Name']:
            df.at[index, 'Categories'] = 'Demo game not playable'
        elif "Server" in row['Name']:
            df.at[index, 'Categories'] = 'Server of a game not playable'
        elif "Editor" in row['Name']:
            df.at[index, 'Categories'] = 'Editor of a game not playable'
        elif "Beta" in row['Name']:
            df.at[index, 'Categories'] = 'Beta game not playable'
        else:
            continue
```

Prafful Singh

Cleaning Tags and Screenshots Columns

Upon analysis, I noticed that:

- The Tags for each game often mirrored the Genres. To enhance data consistency, I replaced any missing Tags with the corresponding Genres. This ensures that even if tags are not explicitly provided, the dataset retains relevant classification for each game.

- For the Screenshots column, I found that Playtest games frequently lacked screenshots. To address this, I implemented a specific replacement strategy:
  - For Playtest games, missing screenshots were labeled as "Playtest game no screenshot available."
  - For all other games without screenshots, I used the label "no screenshot available."

This approach not only fills in gaps but also maintains clarity in the dataset, allowing for more accurate and meaningful analyses in subsequent steps.

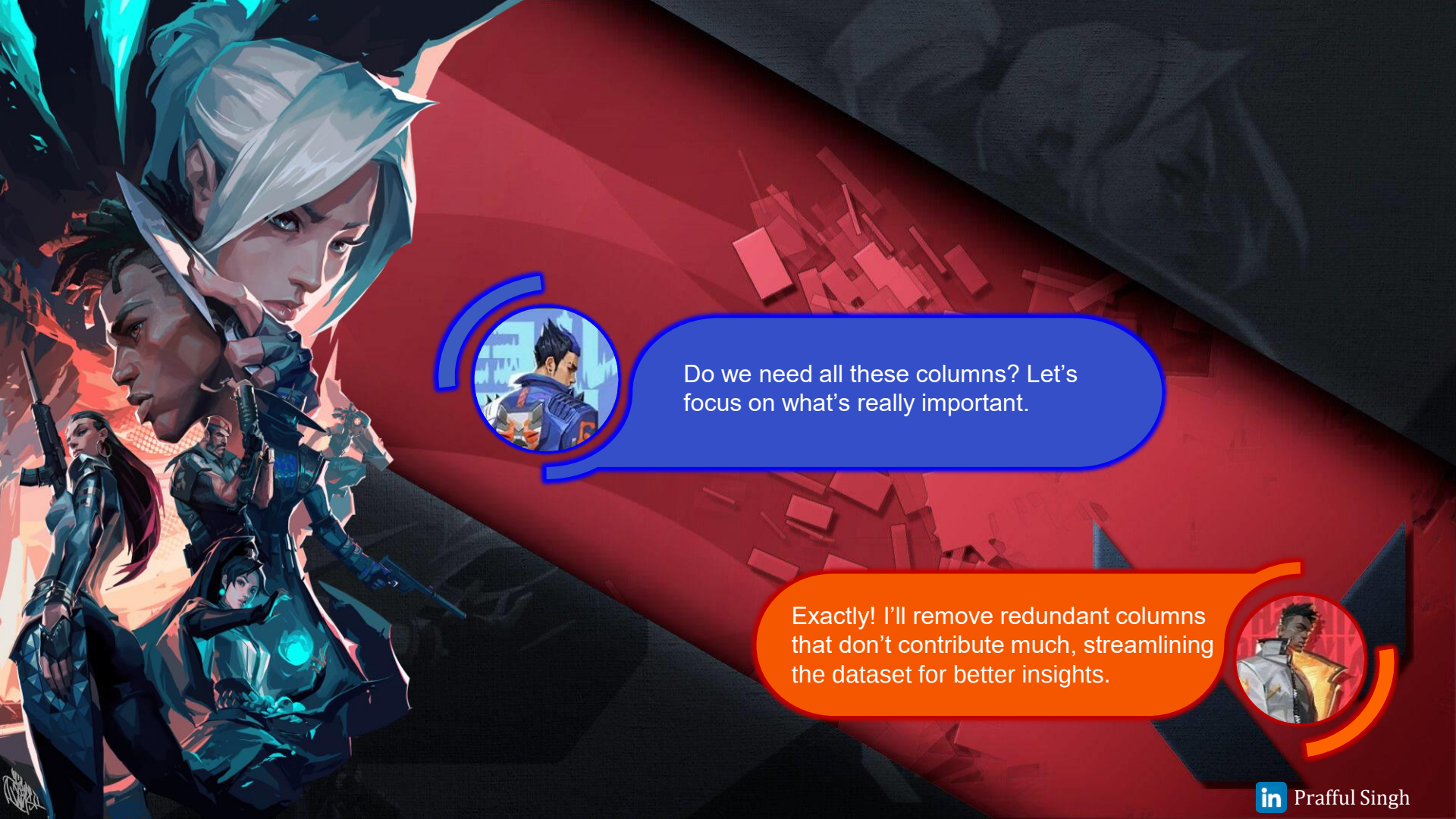
Replacing missing Tags with Genres

```
for index, row in df.iterrows():
    if(pd.isnull(row['Tags'])):
        df.at[index, 'Tags'] = df.at[index,
'Genres']
```

Handling missing Screenshots

```
for index, row in df.iterrows():
    if(pd.isnull(row['Screenshots'])):
        if 'Playtest' in row['Name']:
            df.at[index, 'Screenshots'] =
     'Playtest game no screenshot available'
        else:
            df.at[index, 'Screenshots'] =
            'no screenshot available'
```

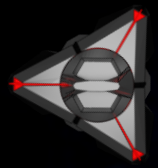
Prafful Singh


Do we need all these columns? Let’s focus on what’s really important.
Exactly! I’ll remove redundant columns that don’t contribute much, streamlining the dataset for better insights.
Prafful Singh

Removing Redundant Columns

To enhance the quality of my dataset, I conducted a thorough review and identified specific columns that had over 50% missing values, which were:

- Notes
- Score rank
- Metacritic url
- Website
- Support url



Removing Redundant Columns

```
df = df.drop('Notes', axis=1)
df = df.drop('Score rank', axis=1)
df = df.drop('Metacritic url', axis=1)
df = df.drop('Support url', axis=1)
df = df.drop('Website', axis=1)
```

Recognizing that these columns did not contribute valuable insights for my analysis, I decisively removed them to maintain a clean and focused dataset.

After this cleanup, I verified the dataset's integrity, confirming that all remaining columns are free of null values

The updated summary indicates that there are **0 null values** in all remaining columns, ensuring a robust dataset ready for analysis.

df.isnull().sum()

AppID	0
Name	0
Release date	0
Estimated owners	0
Peak CCU	0
Required age	0
Price	0
Discount	0
DLC count	0
About the game	0
Supported languages	0
Full audio languages	0
Header image	0
Support email	0
Windows	0
Mac	0
Linux	0
Metacritic score	0
User score	0
Positive	0
Negative	0
Achievements	0
Recommendations	0
Average playtime forever	0
Average playtime two weeks	0
Median playtime forever	0
Median playtime two weeks	0
Developers	0
Publishers	0
Categories	0
Genres	0
Tags	0
Screenshots	0
Movies	0
OS	0

dtype: int64

Prafful Singh

Great job with the data cleaning! Now, I’m excited to see what insights we can gather to improve our game marketing strategies.

Let’s dive into the visualizations and I’ll show you key trends that will help drive your campaigns.

Prafful Singh

Data Visualization

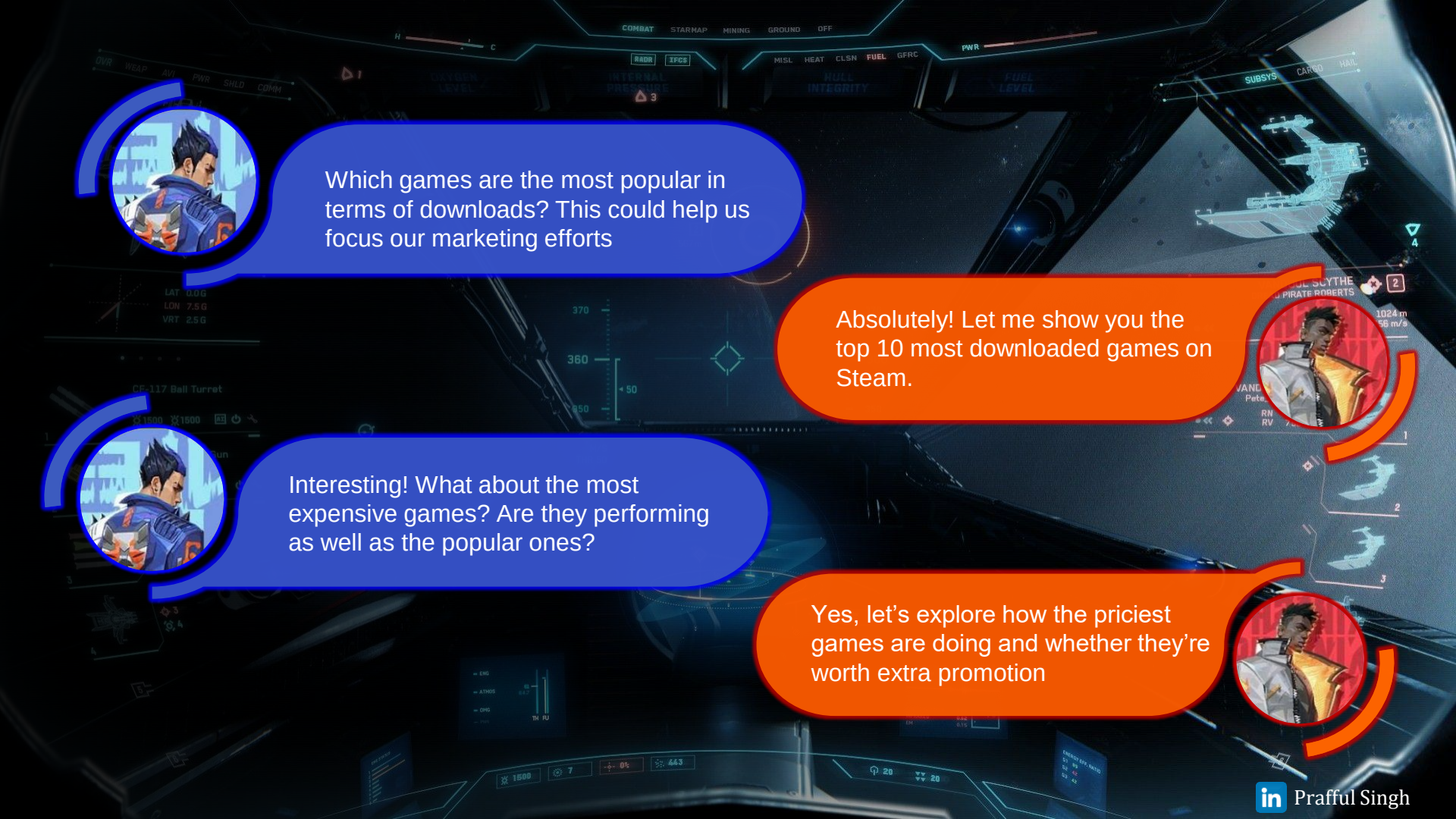
Why Data Visualization Matters:

- **Reveals hidden patterns and trends**: Helps uncover insights that may not be visible in raw data.
- **Simplifies complex datasets**: Breaks down large datasets into intuitive, understandable graphics.
- **Facilitates decision-making**: Presents data in an accessible way for both technical and non-technical audiences.
- **Enhances communication**: Visuals effectively communicate insights, making it easier to explain trends and drive conclusions.
- **Supports storytelling**: Transforms data into a narrative, helping to tell a compelling story through graphics.

I decided to use a visually consistent dark theme for all my plots to make them more striking and professional. To achieve this, I used Plotly's dark mode feature by adding the following code:

```
from plotly import express as px
import plotly.io as pio
pio.templates.default = "plotly_dark"
```

Prafful Singh


Which games are the most popular in terms of downloads? This could help us focus our marketing efforts
Absolutely! Let me show you the top 10 most downloaded games on Steam.
Interesting! What about the most expensive games? Are they performing as well as the popular ones?
Yes, let’s explore how the priciest games are doing and whether they’re worth extra promotion
Prafful Singh

Most Downloaded Games

I analyzed the most downloaded games based on the **Estimated Owners** column, which provided ownership ranges. To make the data more usable, I created a function, owners_clean(), to calculate the median value of each range. This involved splitting the range, converting it to integers, and computing the median.

Next, I applied this function to each row, creating a new column, **owners clean**. Using the groupby method, I aggregated and summed the median owners for each game, then sorted the data to highlight the **top 10 most downloaded games**.

calculating median

```
def owners_clean(x):
    x = x.strip()
    x= x.split("-")
    x = [num.strip(' ') for num in x]
    num1 = int(x[0])
    num2 = int(x[1])
    med = (num2 - num1)/2
    return med
```

Most Downloaded games

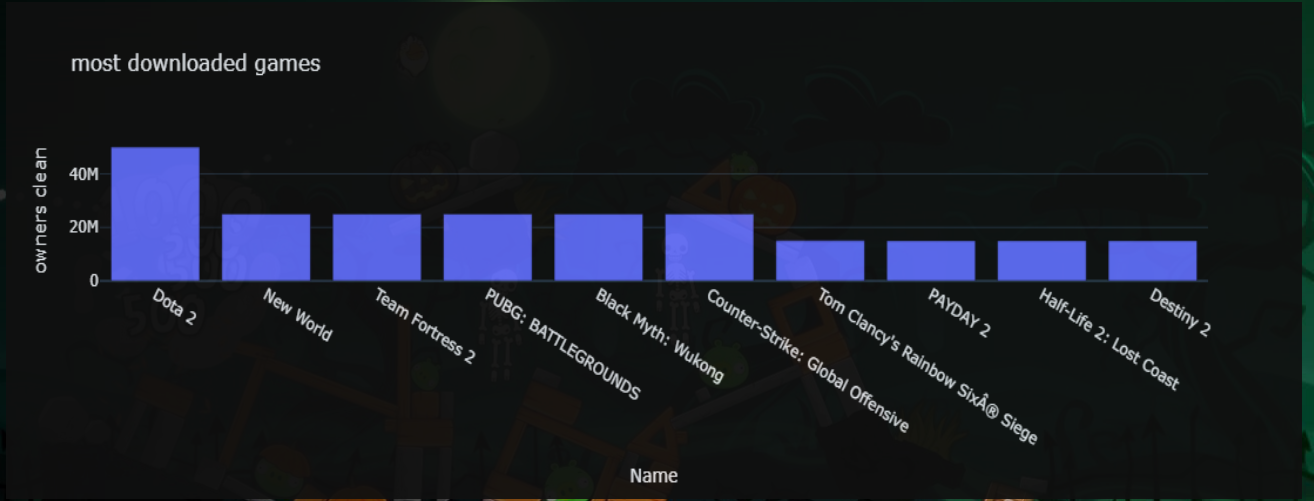
```
for index, row in df.iterrows():
    df.at[index, 'owners clean'] = owners_clean(df.at[index,
'Estimated owners'])
most_downloaded_games = df.groupby("Name").agg({"owners
clean":"sum"}).reset_index().sort_values("owners
clean",ascending=False).head(10)
```

Visualization: To visualize this, I created a **bar chart** using Plotly, displaying the top 10 most downloaded games. The games are plotted on the x-axis, while their respective owner counts are on the y-axis.

most_downloaded_games

```
most_downloaded_games_graph = px.bar(most_downloaded_games,
y='owners clean', x='Name', title='most downloaded games')
```

	Name	owners clean
22261	Dota 2	50000000.0
53166	New World	25000000.0
77141	Team Fortress 2	25000000.0
56491	PUBG: BATTLEGROUNDS	25000000.0
9082	Black Myth: Wukong	25000000.0
16387	Counter-Strike: Global Offensive	25000000.0
83131	Tom Clancy's Rainbow SixÂ® Siege	15040000.0
56208	PAYDAY 2	15000000.0
34497	Half-Life 2: Lost Coast	15000000.0
20761	Destiny 2	15000000.0



Key Results:

- **Dota 2** topped the list with an estimated 50 million owners.
- Other popular games included **New World**, **Team Fortress 2**, **PUBG: BATTLEGROUNDS** and **Black Myth: Wukong** each with 25 million owners.

Prafful Singh

Most expensive games

In this analysis, I identified the **most expensive games** by aggregating the **Price** column. Using the groupby method, I summed the prices for each game and sorted the results in descending order. This highlighted the top 10 most expensive games, providing insight into their high price points. A bar chart was created to visually represent these prices for easy comparison.

	Name	Price
80009	The Leverage Game	999.98
80010	The Leverage Game Business Edition	999.98
5479	Ascent Free-Roaming VR Experience	999.00
84532	True Love	500.00
2038	Aartform Curvy 3D 3.0	299.90
23879	EA SPORTS FCâ„¢ 24	279.96
37581	Houdini Indie	269.99
86226	VEGAS 19 Edit - Steam Edition	249.00
71812	Space Warrior	199.99
62557	ReRise	199.99

Key Insights:

- The **Leverage Game** and **Ascent Free-Roaming VR Experience** stand out with price points nearing $1,000, indicating a niche market for premium gaming experiences.
- The chart provides a clear visual comparison, making it evident how some games command significantly higher prices, likely due to their unique content or production quality.

Prafful Singh


Now that we know the most downloaded and most expensive games, how do price and downloads correlate?
Let’s take a look at how price affects downloads and see if we can spot any trends for free vs. paid games
Also are free games getting more downloads than paid ones? This is critical for our campaign strategies
Yes, let’s explore the data to compare how free and paid games perform in terms of downloads.
Prafful Singh

Most Downloaded Games and Their Prices

In this analysis, I explored the **most downloaded games** alongside their **prices**. By aggregating the total number of downloads and sorting the data, I identified the top 10 games by popularity.

	Name	Price	owners clean
22426	Dota 2	0.00	50000000.0
56889	PUBG: BATTLEGROUNDS	0.00	25000000.0
77688	Team Fortress 2	0.00	25000000.0
53540	New World	39.99	25000000.0
16515	Counter-Strike: Global Offensive	0.00	25000000.0
9149	Black Myth: Wukong	59.99	25000000.0
83749	Tom Clancy's Rainbow SixÂ® Siege	19.99	15040000.0
56606	PAYDAY 2	9.99	15000000.0
89095	Warframe	0.00	15000000.0
57650	Path of Exile	0.00	15000000.0

Key Insights:

- **Dota 2** leads with 50 million downloads, while several games like **PUBG: BATTLEGROUNDS** and **Team Fortress 2** also show significant popularity despite having a zero price tag.
- The visualization highlights how price impacts game popularity, showcasing a mix of free-to-play and premium titles.

Prafful Singh

Prices Distribution

In this segment, I examined the **distribution of game prices** using a box plot to identify the range and outliers within the dataset.

Key Insights:

- The box plot reveals that while many games are priced affordably, there are also significant outliers indicating a presence of premium titles.
- Following the price analysis, I categorized the games into **"Free"** and **"Paid"** based on their price points, finding a substantial majority of paid games compared to free ones.

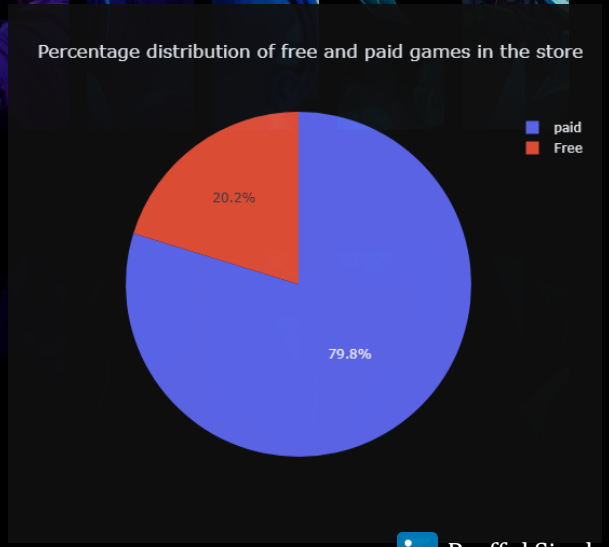
To visualize this distribution, I created a **pie chart** illustrating the percentage distribution of free and paid games in the store.

Free vs Paid Games

```
paid_and_free_count = df.groupby("Free or paid").agg("Name":"count"}).reset_index()
```

paid_and_free_count

	Free or paid	Name
0	Free	19666
1	paid	77599



Prafful Singh

Average downloads for free and paid games

I further analyzed the average downloads for both free and paid games to understand their popularity and market engagement.

Key Insights:

- The average number of owners for free games is significantly **higher** (approximately **43,341**) compared to paid games (around **29,725**), indicating that free games attract a larger player base.
- This suggests that while free games may drive more downloads, paid games might struggle to match that engagement, highlighting different market dynamics between free-to-play and paid models.

paid_and_free_downloads

```
paid_and_free_downloads =
df.groupby("Free or paid").agg({"owners
clean":"mean"}).reset_index()
```

paid_and_free_downloads

	Free or paid	owners clean
0	Free	43341.299705
1	paid	29724.545419

Prafful Singh

Prafful Singh

Most Supported languages

In this analysis, I examined the most supported languages in the dataset, focusing on the "Supported languages" column. Here's how I processed the data:

- **Data Cleaning**: I created a dictionary to count the occurrences of each language, stripping extra spaces and removing unnecessary characters from the language strings. This ensured accurate language representation.
- **Aggregation**: After processing, I converted the language count dictionary into a DataFrame and sorted it to identify the top 10 most supported languages.

Data Cleaning

```
language_count = {}
for language in df['Supported languages'].to_list():
    language = language.strip()
    language_sub = language.split(',')
    for key in language_sub:
        key = key.strip()
        key = key.replace("'", "")
        key = key.replace("[", "")
        key = key.replace("]", "")
        key = key.replace("&amp;lt;strong&amp;gt;&amp;lt;/strong&amp;gt;", "")
        key = key.replace("b/b", "")
        key = key.replace("/b", "")
        key = key.replace(" \\r\\n\\r\\nb/b ", "")
        key = key.replace("/b", "")
        key = key.replace("\\r\\nb/b", "")
        key = key.replace("\\r\\n", "")
        key = key.replace("#", "")
        key = key.replace("\r\\n\\r\\n", "")
        key = key.replace("Russian\\r\\nEnglish\\r\\nSpanish -
                          Spain\\r\\nFrench\\r\\nJapanese\\r\\nCzech", "")
        key = key.replace("\r\\n", "")
        key = key.replace(" &amp;lt;br /&amp;gt;&amp;lt;br /&amp;gt; ", "")
        key = key.replace("RussianEnglishSpanish - SpainFrenchJapaneseCzech", "")
        language_count[key] = language_count.get(key, 0) + 1
```

language_count

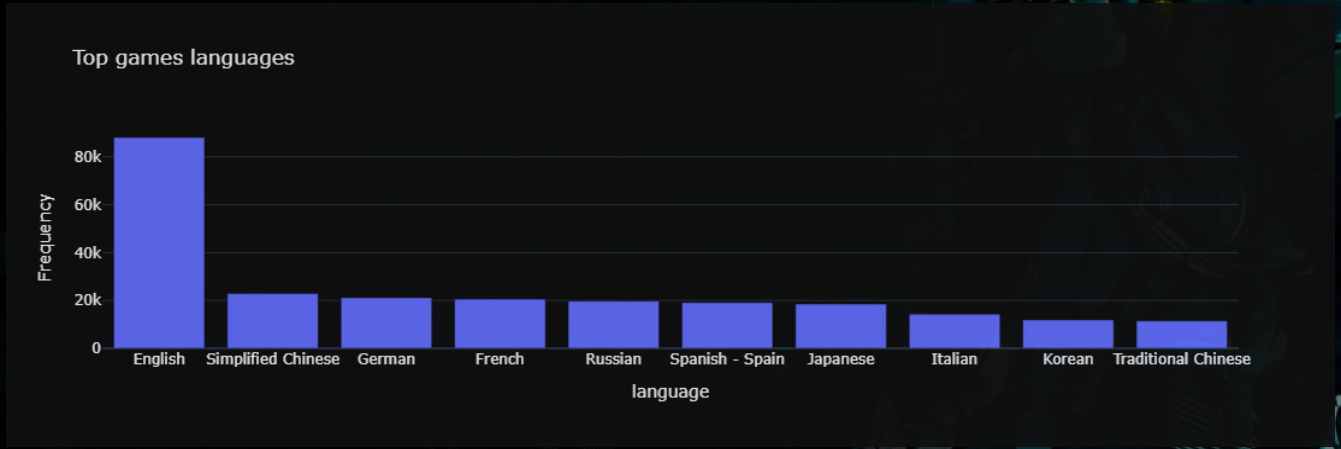
```
language_count = pd.DataFrame.from_dict(language_count,
orient='index').reset_index()
language_count.columns = ['language', 'Frequency']
language_count = language_count.sort_values('Frequency',
ascending = False).head(10)
```

Prafful Singh

This bar chart illustrates the distribution of the most supported languages in the dataset, highlighting the linguistic diversity in gaming. Understanding this can help developers and marketers target their audience more effectively.

```
language_count_graph.show()

language_count_graph = px.bar(language_count,
y='Frequency', x='language', title='Top games
languages')
```



	language	Frequency
0	English	88121
8	Simplified Chinese	22891
3	German	21171
1	French	20590
7	Russian	19753
4	Spanish - Spain	19127
5	Japanese	18507
2	Italian	14319
10	Korean	11913
9	Traditional Chinese	11450

Key Insights:

• **English Dominance**: The analysis revealed that English is the most commonly supported language, with a frequency of 88,121 occurrences, followed by Simplified Chinese and German.

• **Widespread Localization**: Besides English, several other languages, including French, Spanish, and Japanese, have significant support, indicating the global reach and localization efforts of game developers.

Prafful Singh

Average downloads for games with support vs. games without support

In this analysis, I examined the relationship between game support availability and average downloads. By categorizing games based on whether they have a support email, I calculated the average number of owners for both supported and unsupported games.

Key Insights:

- **Significant Difference**: Games without support have a much higher average of approximately 114,776 downloads compared to just 21,761 for games with support. This suggests that users may gravitate towards unsupported games, possibly due to perceived quality or popularity.
- **Visual Representation**: A bar chart effectively illustrates this disparity, making it easy to compare the average downloads for games with and without support at a glance.

Games with support vs. games without support

```
for index, row in df.iterrows():
    if  df.at[index, 'Support email'] == "no Support email available":
        df.at[index, 'Game have support'] = False
    else:
        df.at[index, 'Game have support'] = True
downloads_for_supported_games = df.groupby("Game have support").agg({
    "owners clean":"mean"
}).reset_index()
```

downloads_for_supported_games

	Game have support	owners clean
0	False	114776.012850
1	True	21761.407871

downloads_for_supported_games_graph.show()

```
downloads_for_supported_games_graph =
px.bar(downloads_for_supported_games, y='owners
clean', x='Game have support',title='Average downloads
for games with support vs. games without support')
```

Prafful Singh

Average Downloads for Games by Operating System

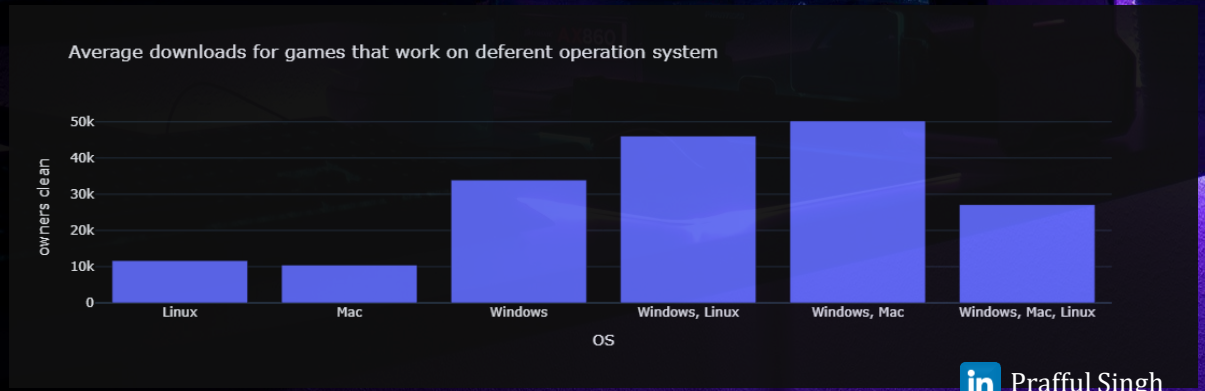
In this analysis, I explored how the availability of games across different operating systems (OS) impacts their average downloads. By grouping the data based on OS compatibility, I calculated the mean number of owners for each category.

Key Insights:

- **Windows Dominance:** Games compatible with Windows have the highest average downloads, at approximately 33,911, highlighting its strong user base compared to other operating systems.
- **Multi-OS Advantage:** Games that support multiple systems, particularly those that run on both Windows and Mac, see significantly higher average downloads (about 50,236), indicating that broader compatibility may enhance a game's appeal.
- **Visual Comparison:** A bar chart provides a clear visual representation of average downloads across different OS, facilitating easy comparisons of their popularity.

average_downloads_for_different_os

	OS	owners clean
0	Linux	11666.666667
1	Mac	10454.545455
2	Windows	33911.260254
3	Windows, Linux	46085.832472
4	Windows, Mac	50236.263736
5	Windows, Mac, Linux	27136.244564



Prafful Singh


What do the ratings tell us about the best and worst games? This could help us highlight or adjust marketing efforts.
I'll break down the games with the highest positive and negative ratings to help guide your strategy
Are the most downloaded games also highly rated? Can we spot any trends?
I'll show you how downloads and ratings are related, which will help us understand user engagement.
Is there a clear relationship between how many downloads a game has and how well it's rated?
Let's explore the correlation between downloads and ratings to see what we can uncover.
CAPCOM
Prafful Singh

Top Rated Games by Positive and Negative Ratings

In this analysis,

- I aggregated the data by game name to sum the positive ratings, allowing me to pinpoint the games with the highest player approval. The same method was applied to determine the games with the most negative ratings.
- The results were sorted in descending order, enabling me to easily identify the top-rated games in both categories.

Top Positive Rated Games

	Name	Positive
16387	Counter-Strike: Global Offensive	5764420
83131	Tom Clancy's Rainbow SixÂ® Siege	2179536
22261	Dota 2	1477153
33122	Grand Theft Auto V	1171197
56491	PUBG: BATTLEGROUNDS	1154655
77500	Terraria	964983
77141	Team Fortress 2	823693
31507	Garry's Mod	822326
65567	Rust	703687
9082	Black Myth: Wukong	663109

Key Insights:

Top Positive Ratings: **Counter-Strike: Global Offensive** leads with over 5.76 million positive ratings, followed by *Tom Clancy's Rainbow Six® Siege* and *Dota 2*, indicating their strong popularity and user satisfaction.

Prafful Singh
CREATED BY PRAFFULMAZZ

Top Negative Rated Games

```
top_negative_rated_games =
df.groupby("Name").agg({"Negative":"sum"}).reset_index()
     .sort_values("Negative",ascending = False).head(10)
```

Top Negative Rated Games Chart

```
top_negative_rated_games_graph =
px.bar(top_negative_rated_games, y='Negative',
x='Name', title='Top Negative Rated Games')
```

Top Negative Rated Games

	Name	Negative
56491	PUBG: BATTLEGROUNDS	895978
16387	Counter-Strike: Global Offensive	766677
83131	Tom Clancy's Rainbow SixÂ® Siege	395207
22261	Dota 2	300437
33122	Grand Theft Auto V	210154
17970	Cyberpunk 2077	129925
19581	Dead by Daylight	112924
65567	Rust	108223
7907	Battlefieldâ„¢ 2042	106038
96042	Tale of Immortal	103661

Key Insights:

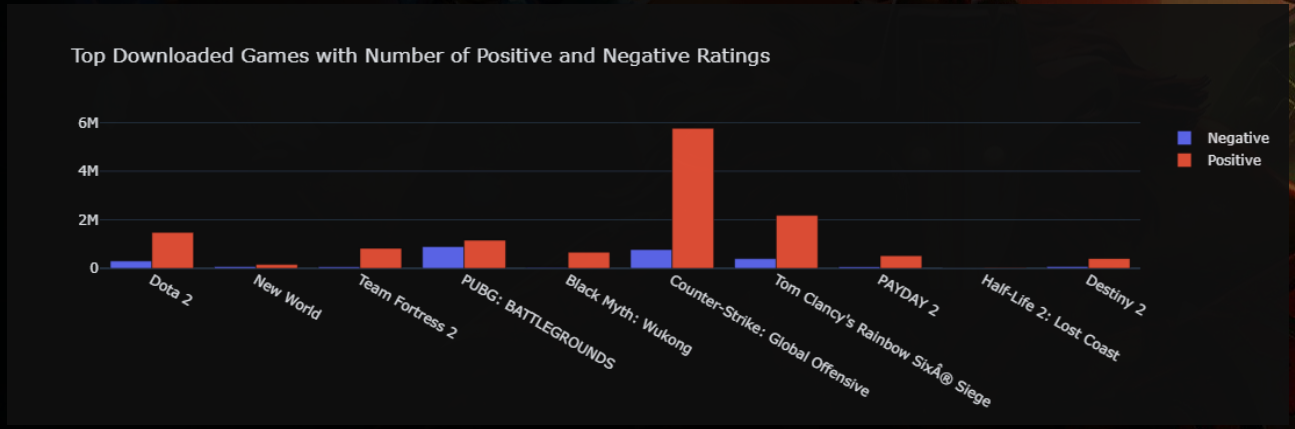
Top Negative Ratings: Conversely, *PUBG: BATTLEGROUNDS* has the highest number of negative ratings, reaching nearly 896,000, suggesting significant player dissatisfaction. Notably, *Counter-Strike: Global Offensive* also appears on this list, highlighting the complexity of player experiences.

Prafful Singh

Most Downloaded Games with Their Positive and Negative Ratings

In this section, I aggregated the data by game name to sum total downloads, positive, and negative ratings, offering a clear view of each game's popularity and user feedback. Then, I sorted the results by downloads to highlight the top 10 most downloaded games along with their ratings.

	Name	owners clean	Negative	Positive
22261	Dota 2	50000000.0	300437	1477153
53166	New World	25000000.0	73900	154914
77141	Team Fortress 2	25000000.0	56683	823693
56491	PUBG: BATTLEGROUNDS	25000000.0	895978	1154655
9082	Black Myth: Wukong	25000000.0	28700	663109
16387	Counter-Strike: Global Offensive	25000000.0	766677	5764420
83131	Tom Clancy's Rainbow SixÂ® Siege	15040000.0	395207	2179536
56208	PAYDAY 2	15000000.0	62574	520826
34497	Half-Life 2: Lost Coast	15000000.0	1261	9306
20761	Destiny 2	15000000.0	77006	403109



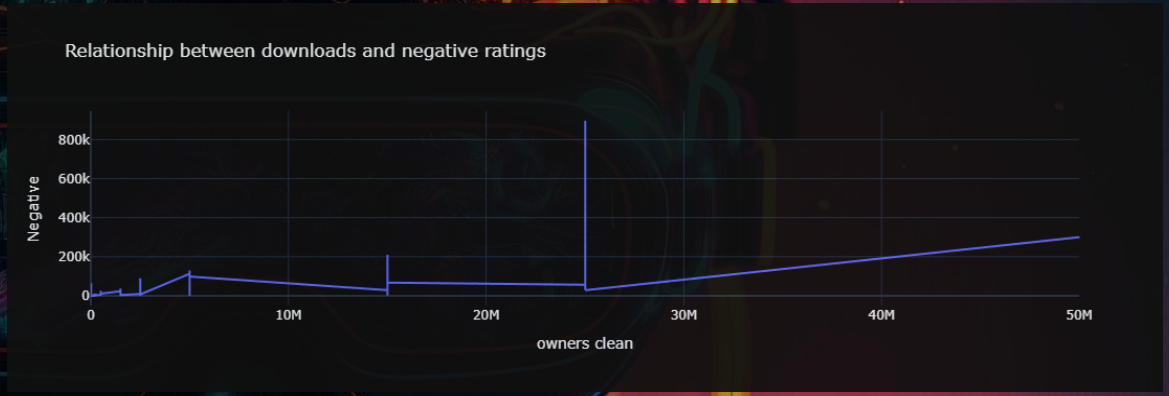
Key Insights:

- *Dota 2* tops the list with 50 million downloads, accompanied by a significant positive rating of 1,477,153, showcasing its popularity among players.
- Games like *PUBG: BATTLEGROUNDS* and *Counter-Strike: Global Offensive* illustrate the dual nature of player experiences, having high download numbers but also substantial negative ratings. This suggests that popularity does not always correlate with user satisfaction.

Prafful Singh

Relationship between the number of downloads and the ratings of games

In this section of the analysis, I explored the relationship between the number of downloads and game ratings. I visualized the correlation between downloads and both positive and negative ratings using line charts. Sorting the data by downloads allowed me to observe trends in how user feedback relates to game popularity.



Key Insight:
The line graphs reveal that games with 20 to 30 million users show the highest peaks in both positive and negative ratings. This suggests that games with a larger user base tend to attract the most feedback, reflecting their widespread popularity and active player engagement.

Prafful Singh

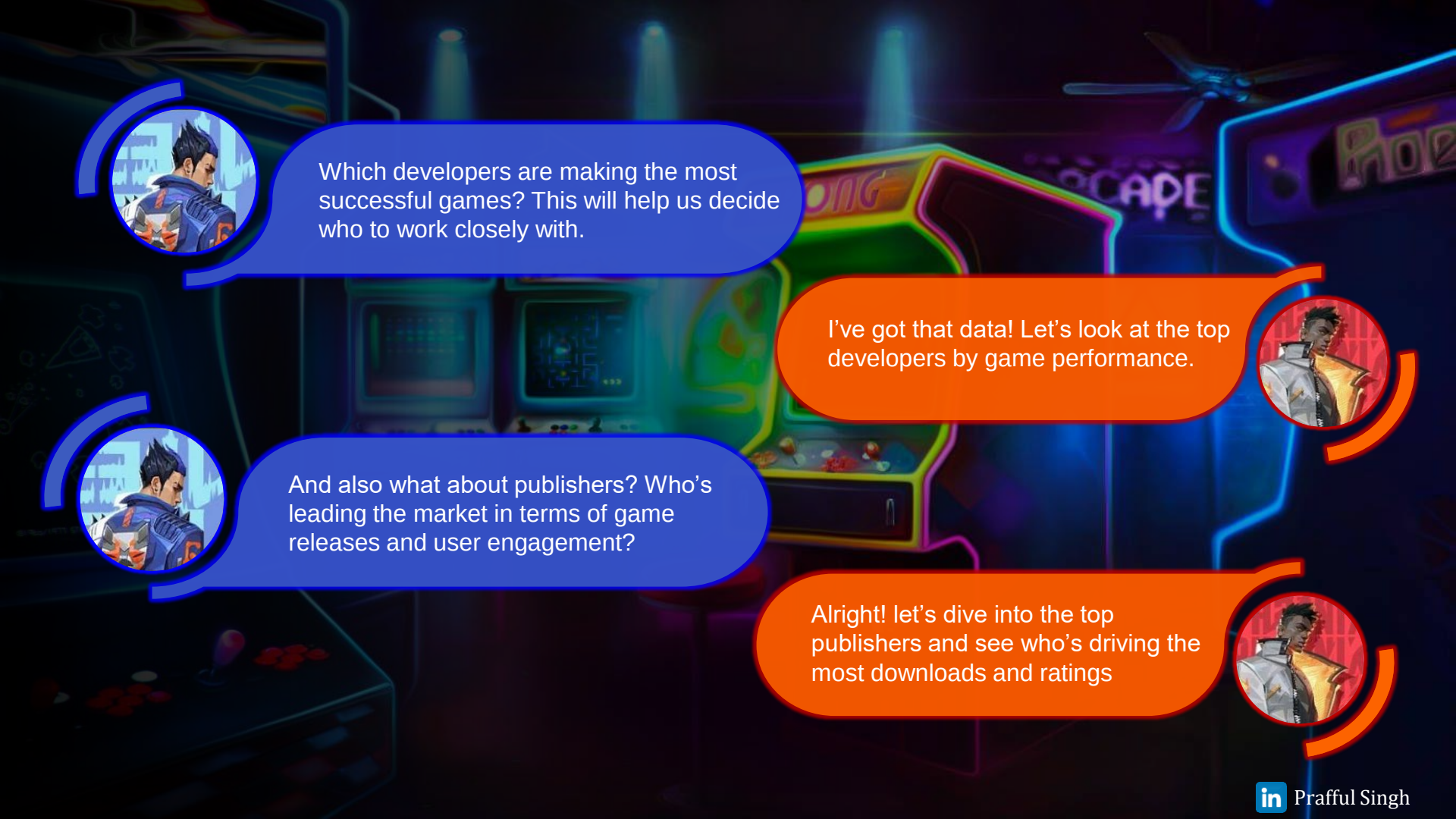
Relationship Between Price and Ratings

I sorted the data by price in descending order and used box plots to visualize the distribution of positive and negative ratings across various price ranges. Each point on the graphs represents a game, and hovering over the points shows the specific game name. This allows us to analyze the variability and spread of ratings within different pricing categories.

Key Insight:

The majority of games are concentrated in the price range of 0 to 200, indicating that most user feedback—both positive and negative—comes from games within this pricing bracket. This suggests that games in this range may attract more users, leading to higher interaction in terms of ratings and reviews.

Prafful Singh



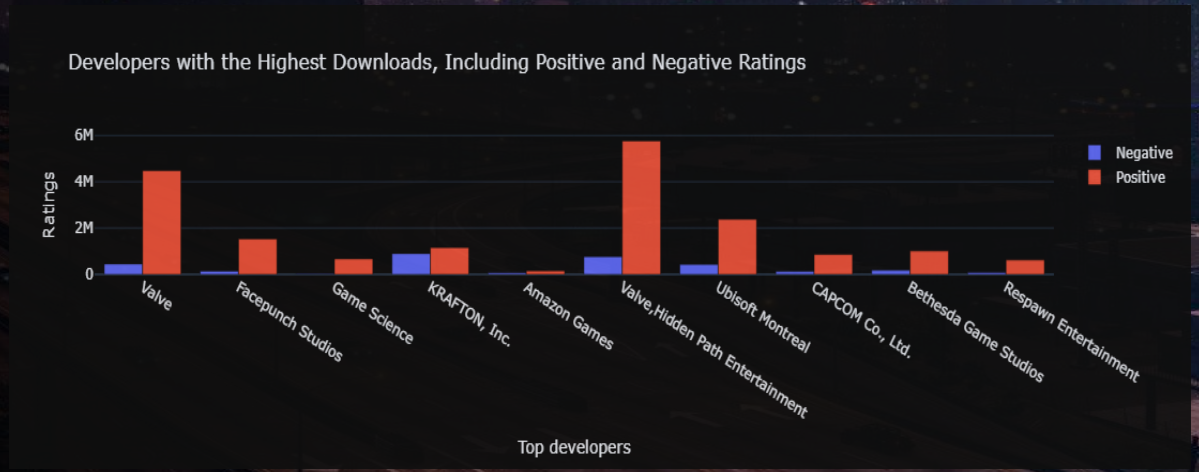
Prafful Singh

Top Developers

In this section of the analysis,

- I aggregated the data by developers to sum the total downloads, positive ratings, and negative ratings. This analysis provided a holistic view of each developer's overall performance, focusing on popularity (downloads) and user feedback (positive and negative ratings).
- I identified the top 10 developers based on total downloads, positive ratings, and negative ratings. The results were presented using bar charts to highlight the key players in the gaming industry.
- I created grouped bar charts to show the relationship between positive and negative ratings for the top developers, and separate bar charts to display developers with the highest positive and negative ratings.

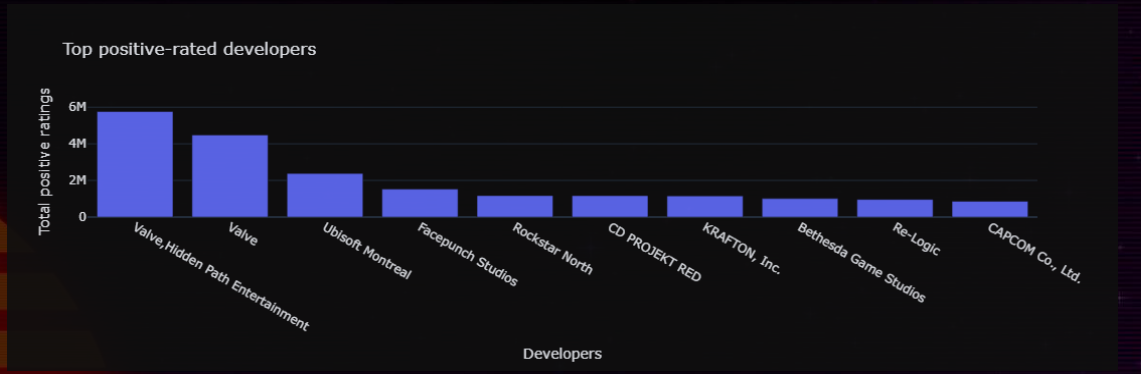
	Developers	owners clean	Positive	Negative
48507	Valve	174370000.0	4485026	450698
15180	Facepunch Studios	30050000.0	1527225	137407
17447	Game Science	25500000.0	668350	30105
24017	KRAFTON, Inc.	25275000.0	1157817	897352
2491	Amazon Games	25000000.0	154914	73900
48509	Valve,Hidden Path Entertainment	25000000.0	5764420	766677
47778	Ubisoft Montreal	22395000.0	2386620	432780
7306	CAPCOM Co., Ltd.	19210000.0	861749	125303
5352	Bethesda Game Studios	17250000.0	1011497	174578
38055	Respawn Entertainment	17075000.0	620908	84736



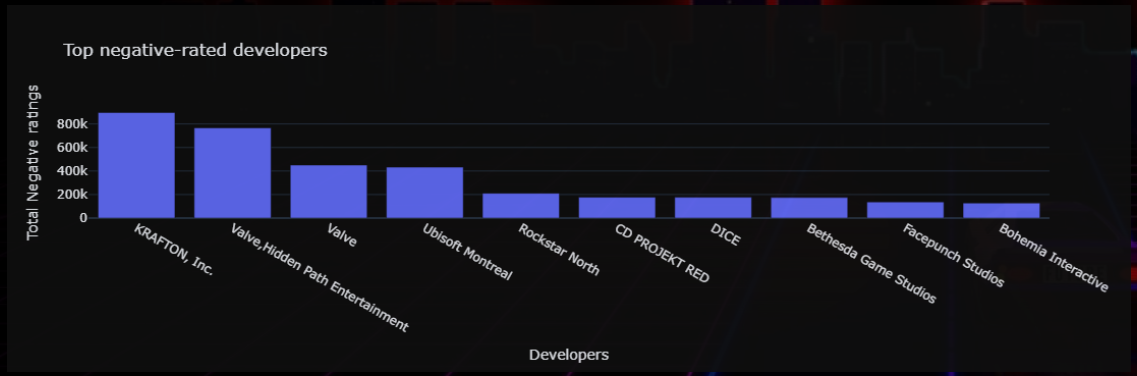
Prafful Singh

Key Insights:

- **Valve Dominates**: Valve, as well as Valve in collaboration with Hidden Path Entertainment, leads the market with the highest number of downloads and positive ratings, suggesting that their games are not only popular but also well-received by users.
- **KRAFTON's Polarizing Impact**: Despite high download numbers, KRAFTON, Inc. also has the highest number of negative ratings, indicating that while their games attract many users, they also face significant criticism.



- **Top Developers Have High Engagement**: The top developers in terms of downloads, such as Valve, Ubisoft Montreal, and Facepunch Studios, consistently have high levels of both positive and negative ratings, showing strong user engagement and a mix of reactions to their games.



- **Mixed Feedback for Rockstar and CD Projekt Red**: Developers like Rockstar North and CD PROJEKT RED feature prominently in both the positive and negative ratings, suggesting that while they create highly anticipated games, they can be polarizing for the player base.

Prafful Singh

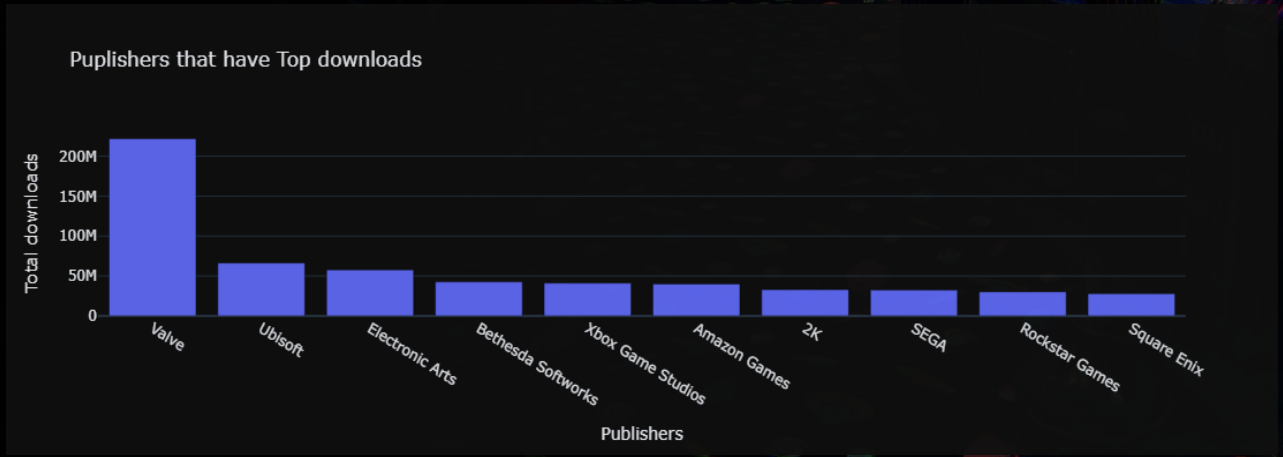
Top Publishers

In this section of the analysis,
- I summed up the total downloads, positive ratings, and negative ratings for each publisher, providing insights into the overall popularity and user sentiment.
- Sorted publishers based on the total number of downloads, highlighting the top 10 publishers by their download counts.
- Created bar charts to show both positive and negative ratings alongside download figures for each publisher.

Top Publishers

```
top_publishers_total_downloads =
df.groupby("Publishers").agg({"owners
clean":"sum"}).reset_index().sort_values("owners
clean",
ascending = False).head(10)
```

	Publishers	owners clean
42561	Valve	222020000.0
41896	Ubisoft	66395000.0
12050	Electronic Arts	57750000.0
4678	Bethesda Softworks	42730000.0
44559	Xbox Game Studios	41120000.0
2182	Amazon Games	40000000.0
259	2K	32845000.0
34491	SEGA	32475000.0
33955	Rockstar Games	30365000.0
37511	Square Enix	28015000.0

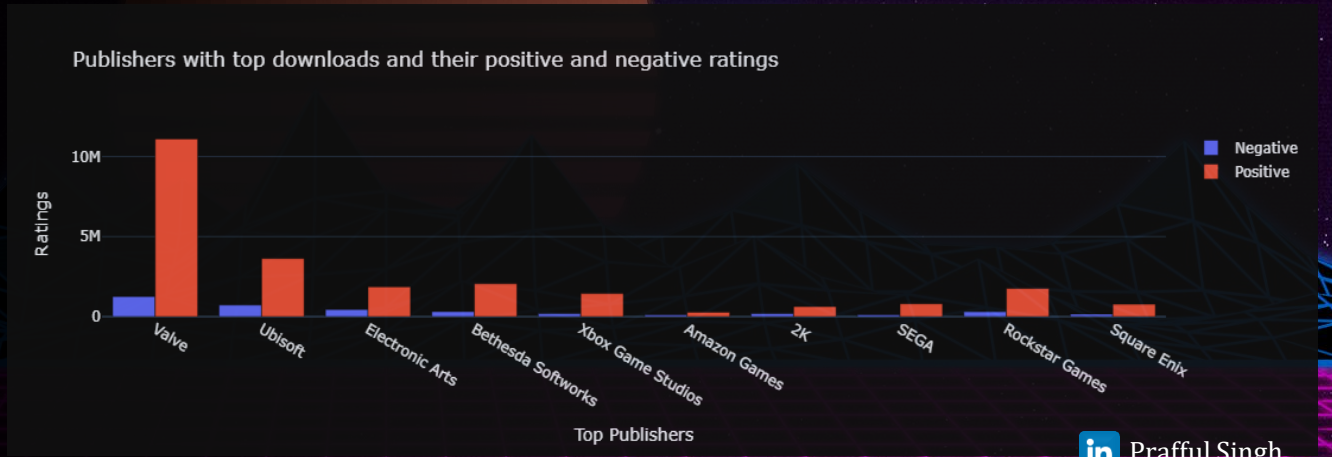


Prafful Singh

Key Insights:

- **Valve's Dominance**: Valve stands out with over 222 million downloads, far surpassing all other publishers. It also has the highest number of positive ratings (11 million) but also the highest negative ratings (1.2 million), indicating a wide user base with diverse feedback
- **Ubisoft and Electronic Arts**: Ubisoft ranks second with nearly 66 million downloads, and Electronic Arts follows with 57.7 million downloads. Both have significant positive ratings, with 3.6 million and 1.8 million, respectively, but also face substantial negative feedback, which is a point to consider in customer satisfaction.
- **Consistent Trends Across Major Publishers**: Bethesda Softworks, Xbox Game Studios, and Rockstar Games also show large download counts (ranging from 30M to 42M) with high positive feedback. However, the trend of high downloads correlating with noticeable negative ratings continues.
- **Smaller Publishers**: Publishers like 2K and SEGA, despite having lower download numbers compared to the top tier, maintain fairly balanced positive-to-negative rating ratios, indicating a more favorable user sentiment.

	Publishers	owners clean	Positive	Negative
42561	Valve	222020000.0	11097327	1248173
41896	Ubisoft	66395000.0	3630187	727119
12050	Electronic Arts	57750000.0	1863575	444783
4678	Bethesda Softworks	42730000.0	2057753	315314
44559	Xbox Game Studios	41120000.0	1445136	189528
2182	Amazon Games	40000000.0	279222	117966
259	2K	32845000.0	628702	195795
34491	SEGA	32475000.0	809011	120151
33955	Rockstar Games	30365000.0	1764218	304783
37511	Square Enix	28015000.0	774824	161570



Prafful Singh

What types of games are dominating the platform? We need to know this for targeted promotions.

I'll show you the distribution of game categories to help guide your marketing efforts.

And which game genres are the most popular? This is key for planning our next campaigns

Let's take a look at the top genres and what's trending on Steam.

NEW RETRO ARCADE TECH DEMO

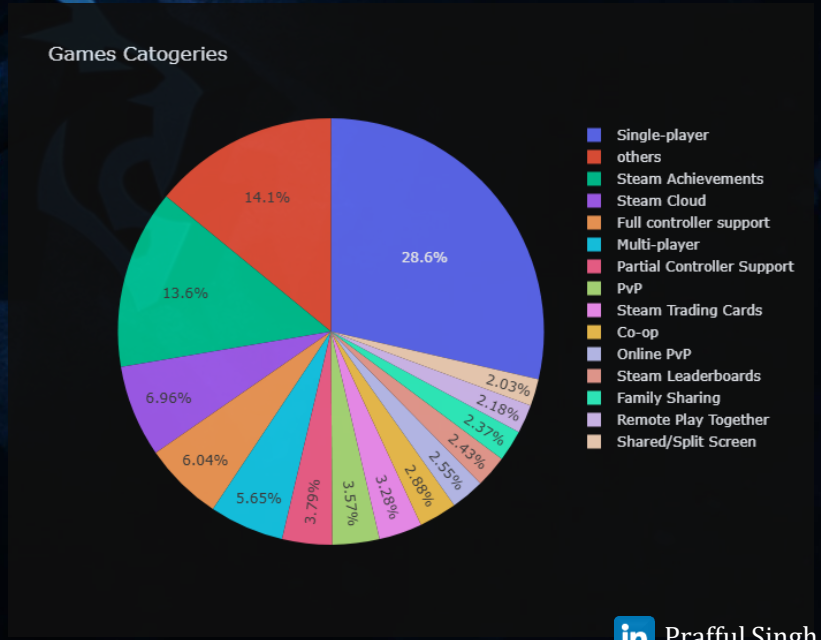
Prafful Singh

Distribution of Games Categories

In this section, I analyzed game categories by extracting and cleaning data to identify the types of games available. I created a frequency count for each category, grouped less common ones into an "others" category, and visualized the distribution with a pie chart to highlight the proportions of each category.

Key Insights:

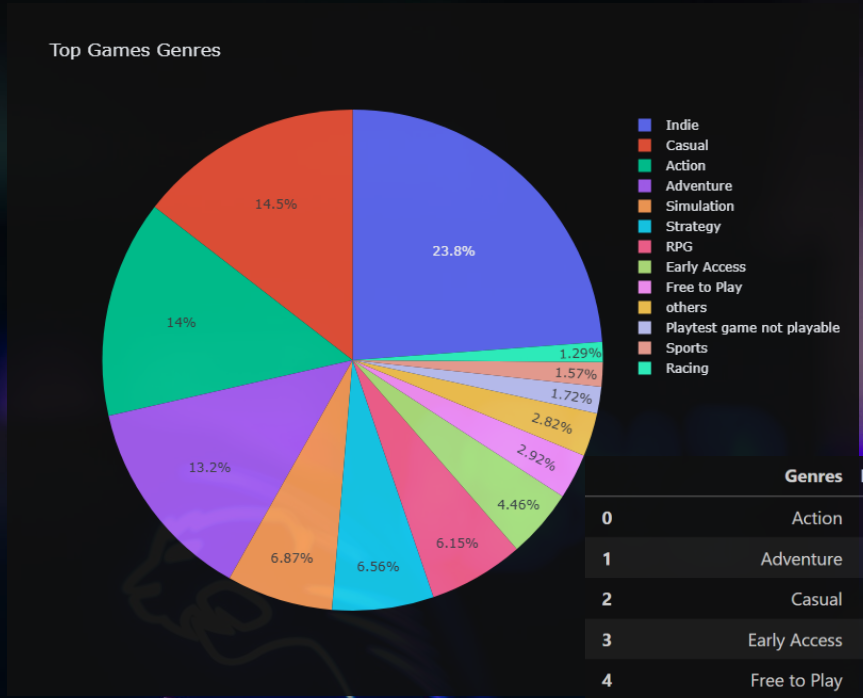
- **Dominance of Single-player Games**: 86,771 titles highlight a strong preference for solo gaming experiences.
- **Significant Interest in Achievements**: Categories like "Steam Achievements" (41,189 titles) and "Steam Cloud" (21,107 titles) show gamers value features that enhance gameplay.
- **Multi-player and Co-op Modes**: Multi-player games have 17,150 titles, with co-op options indicating a solid interest in social gaming dynamics.
- **Other Categories**: The "others" category totals 42,712 titles, suggesting collectively notable diversity in the gaming landscape.
- **Balanced Game Offering**: A healthy variety of game types caters to diverse player preferences, enhancing the platform's overall appeal.



Prafful Singh

Top Genres

In this section, I analyzed game genres by extracting and cleaning the data to identify unique genres. I created a frequency count for each genre, grouped less common ones into an "others" category, and visualized the distribution with a pie chart to highlight the proportions of each genre.



	Genres	Frequency
0	Action	37946
1	Adventure	35802
2	Casual	39263
3	Early Access	12058
4	Free to Play	7891
5	Indie	64418
6	Playtest game not playable	4641
7	RPG	16613
8	Racing	3473
9	Simulation	18567
10	Sports	4254
11	Strategy	17725
12	others	7609

Key Insights:

- **Indie Dominance**: The "Indie" genre leads with 64,418 titles, indicating a strong interest in independent game development.
- **Casual and Action Genres**: "Casual" and "Action" genres follow, with 39,263 and 37,946 titles, respectively, reflecting a significant market for engaging and accessible gameplay.
- **Diverse Genre Representation**: Other genres like "Adventure," "Simulation," and "RPG" also have notable frequencies, suggesting a well-rounded offering for various player preferences.
- **Collective Impact of Lesser Genres**: The "others" category aggregates 7,609 titles, indicating that while individually less common, these genres contribute to the diversity of the gaming landscape.

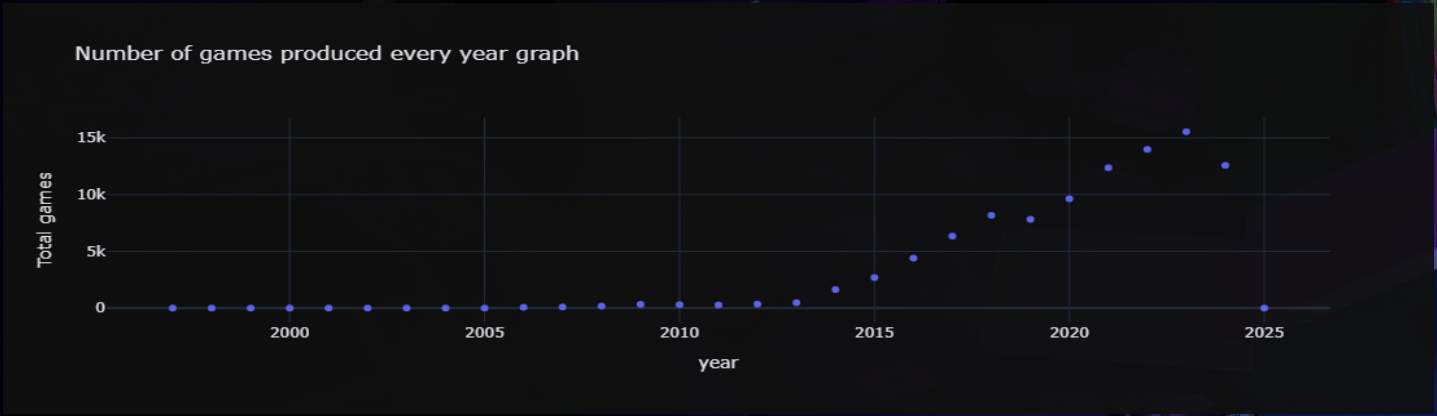
Prafful Singh

Most common year in which games were produced

In this section, I analyzed the production of games over the years by extracting the release year from the dataset. I created a count of games produced for each year, sorted the data to identify trends, and visualized the results with a scatter plot to showcase the number of games released each year.

Key Insights:

- **Peak Year**: 2023 had the highest releases at 15,536 titles, showcasing a strong market.
- **Rising Trend**: Steady growth in game production from 2020 to 2023 indicates increased interest.
- **Historical Shift**: Significant drop in production before 2010 highlights rapid market growth in recent years



common years

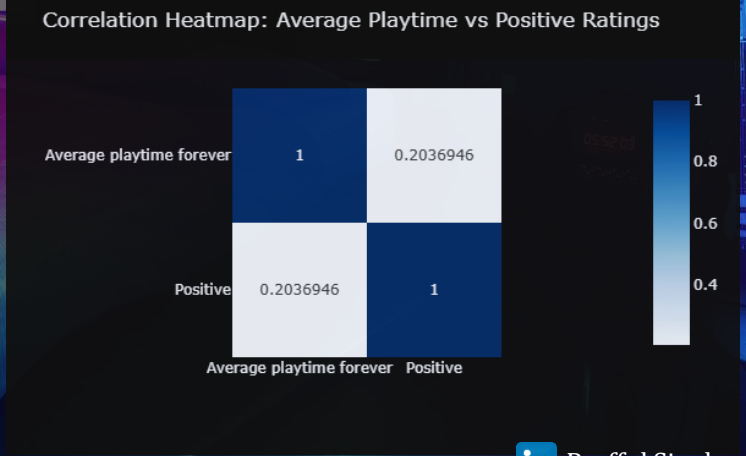
	year	Name
26	2023	15536
25	2022	13979
27	2024	12577
24	2021	12374
23	2020	9628
21	2018	8166
22	2019	7808
20	2017	6324
19	2016	4406
18	2015	2683
17	2014	1627
16	2013	490
15	2012	356
12	2009	339
13	2010	300
14	2011	286
11	2008	176
10	2007	109
9	2006	69
8	2005	7
7	2004	7
4	2001	4
6	2003	3
2	1999	3
0	1997	2
3	2000	2
28	2025	2
1	1998	1
5	2002	1

Prafful Singh


Can you analyze how different game metrics like playtime and ratings are related? This could help us predict engagement.
Sure! I've conducted a correlation analysis. Let's see how metrics like playtime and ratings interact.
Prafful Singh

Correlation Analysis of Game Metrics

In this section, I conducted a correlation analysis to examine the relationships between various game metrics. Specifically, I analyzed the correlation between positive ratings and recommendations, as well as between average playtime and positive ratings. To effectively visualize these correlations, I created heatmaps of these relationships.



Key Insights:

- **Strong Relationship**: A high correlation (0.90) exists between positive ratings and recommendations, indicating that games with better ratings tend to receive more recommendations.
- **Moderate Connection**: The correlation between average playtime and positive ratings is weaker (0.20), suggesting that playtime does not significantly influence positive ratings.
- **Interpretation of Trends**: The strong link between ratings and recommendations highlights the importance of user satisfaction in influencing game popularity.

Prafful Singh

Standard Deviation of Average and Median Playtime for the Last Two Decades

In this section, I calculated the standard deviation of average and median playtime for games from 2004 to 2024. The results were visualized in a heatmap, highlighting the variability in playtime over the years.

Std of Average and Median Playtime for the Last Two Decades

```
playtime_std = df[df['year'].between(2004, 2024)].groupby('year').agg({
    'Average playtime forever': 'std',
    'Median playtime forever': 'std'
}).reset_index()
fig = px.imshow(playtime_std.set_index('year').T, color_continuous_scale='Blues', text_auto=True)
fig.show()
```

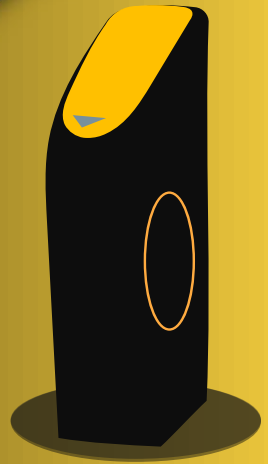
Key Insights:

- **Variability in Playtime**: Higher standard deviation indicates greater disparity in playtime among games.
- **Trends Over Time**: The heatmap reveals fluctuations in player engagement, showing how gameplay experiences have evolved.
- **Player Engagement**: Insights gained can guide developers in creating experiences that align with player preferences.

Prafful Singh

Business Impact:

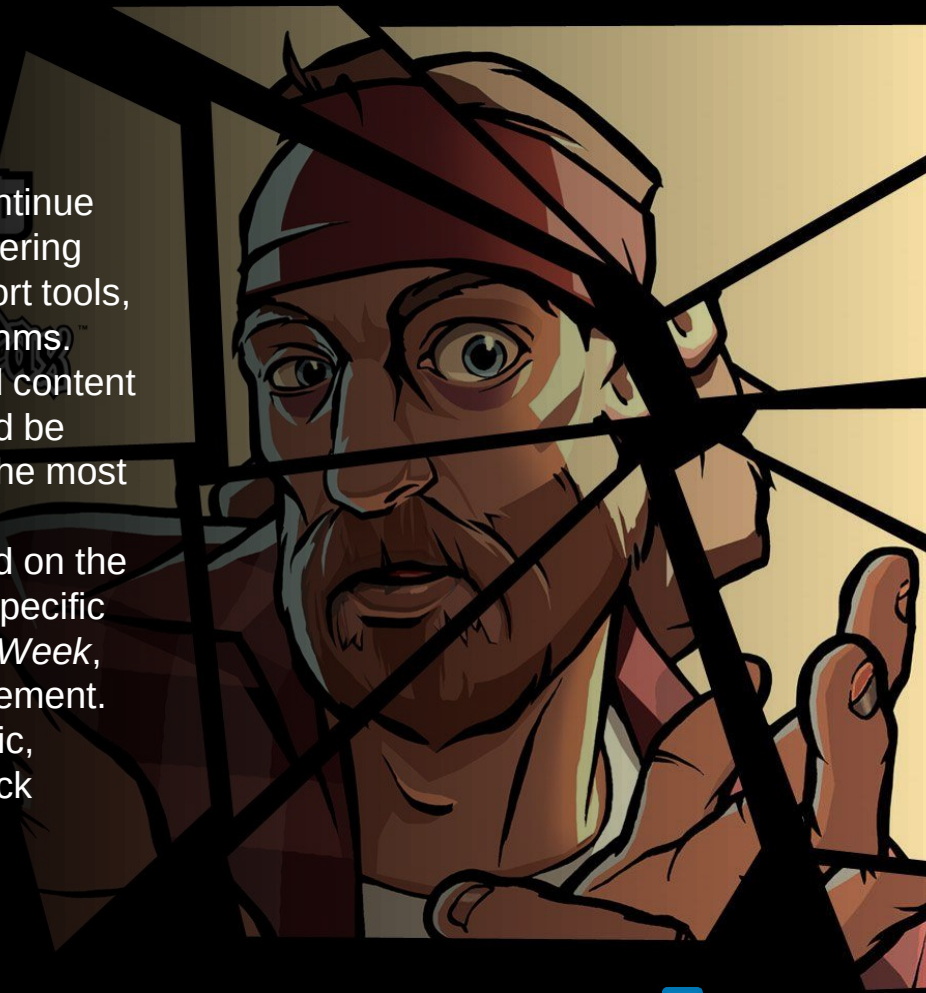
- **Diverse Developer and Publisher Landscape**: The analysis showed that a large number of games are released by a small set of major publishers, while the majority are from independent developers, indicating Steam's role as a vital platform for both indie and big-budget games.
- **Genre Dominance**: Our findings on *Indie*, *Action*, and *Casual* games being the most popular genres reveal Steam's audience preference, while smaller genres also contribute meaningfully to the platform's content diversity.
- **Growth in Game Releases**: There is a clear upward trend in the number of games produced annually, especially after 2020, which showcases a growing market.
- **User Engagement and Recommendations**: Strong correlations between positive reviews and recommendations reflect a virtuous cycle, where user satisfaction drives visibility and potential sales.



Prafful Singh

Recommendations:

- **Support Independent Developers**: Steam should continue and expand support initiatives for indie developers, offering more resources like better marketing, developer support tools, and visibility through Steam's recommendation algorithms.
- **Focus on Community Engagement**: User-generated content such as reviews, guides, and recommendations should be leveraged further by introducing incentives to reward the most helpful contributors.
- **Leverage Genre Insights for Curated Events**: Based on the dominant genres, Steam could organize more genre-specific events like *Indie Festivals* or *Action-Adventure Game Week*, aligning content with user preferences to boost engagement.
- **Diversify Content**: While popular genres bring in traffic, exploring niche genres with curated events could unlock untapped segments, especially those with smaller but dedicated fan bases.



Key Findings

- **Developer-Publisher Relationships**: We identified that several publishers release large volumes of games, while many smaller developers release a single title. This suggests that large publishers may be outsourcing game development or collaborating with multiple developers.
- **Positive Correlation Between User Engagement and Game Success**: A strong correlation was found between positive ratings, playtime, and recommendations, which indicates that user engagement directly boosts a game's visibility and success.
- **Significant Variability in Playtime**: Games released between 2004 and 2024 show high variability in both average and median playtime, with standard deviations indicating that some games succeed in keeping players engaged much more than others.
- **Growing Popularity of Multiplayer and Online Games**: Games with higher playtime also tend to be multiplayer-focused, indicating a market shift towards games that offer longer engagement through online or social interactions.
- **Consistency in Recent Years**: The rise in the number of games produced, especially during and after the pandemic years (2020-2023), indicates sustained growth in game development. This suggests a promising future for both indie and AAA game developers on Steam.

Prafful Singh

This has been incredibly insightful. Thanks for walking me through the data. These insights will really help with our marketing strategies.

Thanks for the opportunity! These insights can help us drive player engagement and optimize game promotions on Steam.

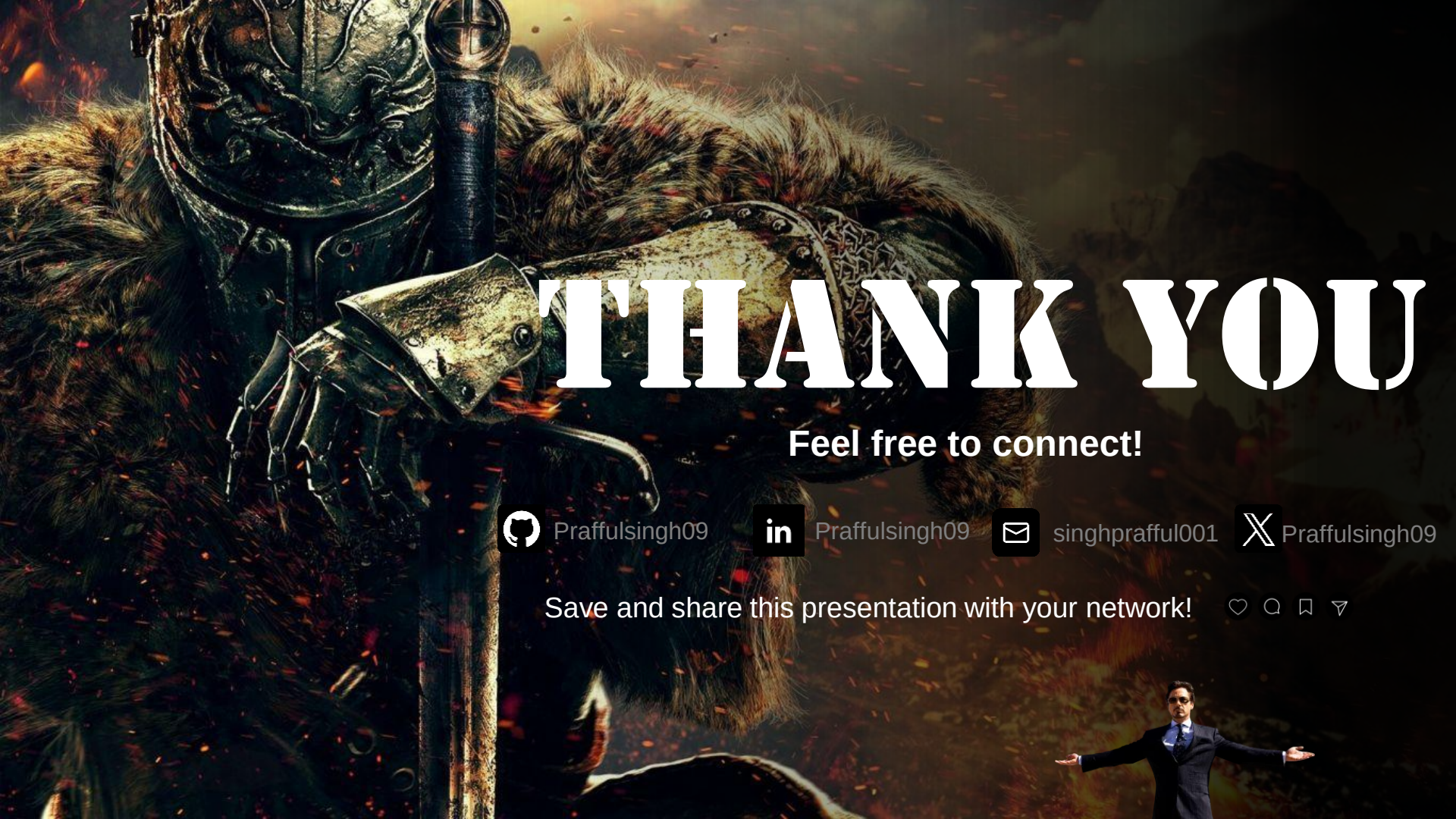
Prafful Singh

Conclusion

This project provided a comprehensive exploration of Steam's game data, offering key insights into game genres, production trends, developer-publisher dynamics, and user engagement. We found that Steam is not only a thriving hub for independent and AAA games but also that positive player feedback and engagement directly contribute to game success. The growth in game production post-2020 further underlines the platform's ability to attract new content, making it a key player in the gaming industry. Looking ahead, Steam can build on its strengths by fostering deeper community interactions, investing in indie game development, and further diversifying its content. The findings from this project can serve as a basis for enhancing both player and developer experiences on the platform.

And yes, I know you've already figured it out—this was just an imaginary story!
But through this fun journey, I hope you enjoyed the **little twist I added**.
This was a creative way to present the data, blending insights with a **playful narrative**.
Thanks for staying with me, and I hope you liked it!
For any questions or further discussion, feel free to reach out!

Prafful Singh


THANK YOU
Feel free to connect!
Praffulsingh09
Praffulsingh09
singhprafful001
Praffulsingh09
Save and share this presentation with your network!