Vineetha Maddikunta

Fall 2024, Big Data Applications

Steam Games Dataset Analysis Using AWS Cloud Infrastructure

1 Introduction

The Steam Games Dataset Analysis project aims to explore the gaming industry's dynamics through Steam's extensive game library. This analysis will investigate key aspects of the gaming market including pricing strategies, user engagement patterns, and platform distribution trends. Through AWS's cloud infrastructure, including S3 for data storage, EC2 for computing, SageMaker for machine learning, and QuickSight for visualization, the project will process and analyze this substantial dataset.

By leveraging automated data pipelines and advanced analytics tools, this project seeks to uncover meaningful patterns in gaming market dynamics, providing valuable insights for industry stakeholders, developers, and researchers alike.

2 Data

The Steam Games Dataset is a comprehensive collection of data related to video games available on the Steam platform. It provides detailed metadata and features about various games, which can be utilized to gain insights into the gaming industry.

It contains information on thousands of games available on the Steam platform.

Large and varied enough to derive meaningful insights about gaming trends, user preferences, and market dynamics.

Below is an overview of the dataset:

Source: The dataset is hosted on Kaggle - https://www.kaggle.com/datasets/mexwell/steamgames/data

- Game Metadata:

Name: The title of the game.

Release Date: Information about when the game was released on the platform.

Developer/Publisher: The studios or entities responsible for creating and distributing the game.

- Categorical Information:

Genres: Categories of the games, such as Action, Strategy, Adventure, etc.

Tags: User-generated or predefined labels describing the game.

- Platform Information:

Supported operating systems like Windows, macOS, and Linux.

- Pricing Information:

The cost of the game, including discounts or free-to-play status.

- User Feedback:

Data related to user reviews, ratings, or sentiment scores.

- User stats:

Data related to user engagement - average playtime, recommendations.

Dataset information:

Number of records : 71716 records

Size : 217.43 MB

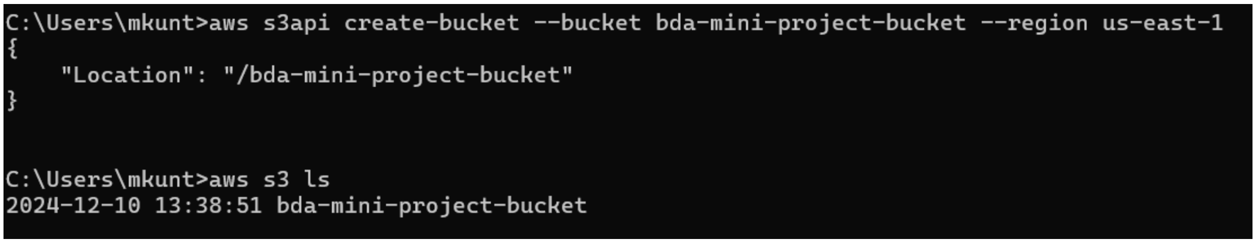
File type : csv

3 Methodology

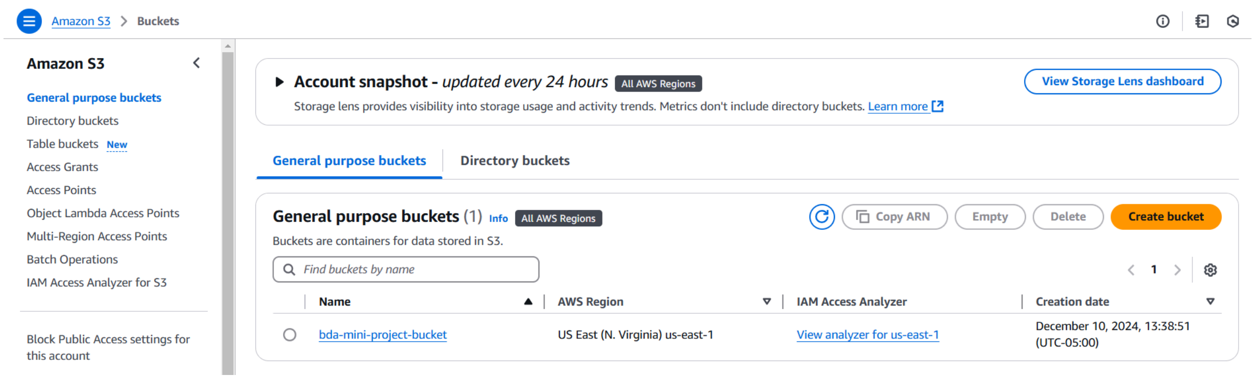
3.1 Environment Setup

- Creating S3 Bucket :

Create an S3 Bucket under the S3 section of AWS Console Management or run this command in the command prompt in the local machine to create the required bucket.



```
C:\Users\mkunt>aws s3api create-bucket --bucket bda-mini-project-bucket --region us-east-1
{
    "Location": "/bda-mini-project-bucket"
}


C:\Users\mkunt>aws s3 ls
2024-12-10 13:38:51 bda-mini-project-bucket
```



Account snapshot - *updated every 24 hours* All AWS Regions

Storage lens provides visibility into storage usage and activity trends. Metrics don't include directory buckets. Learn more

General purpose buckets (1) Info All AWS Regions

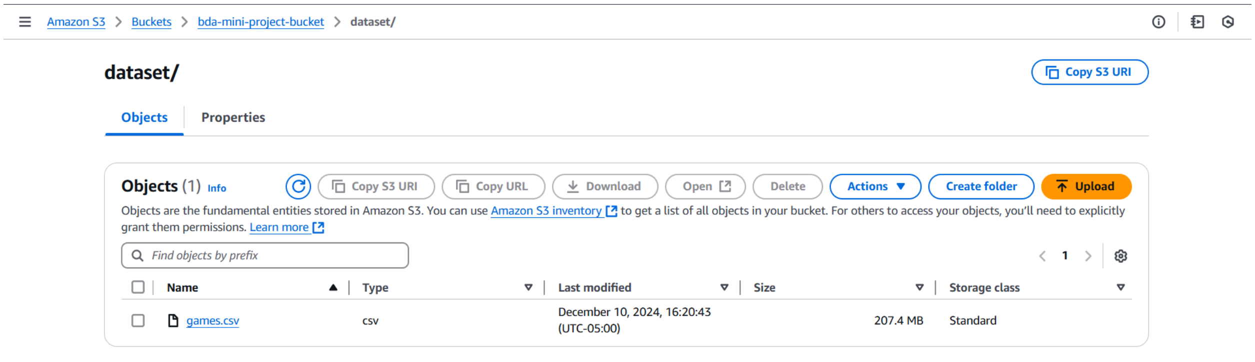
Buckets are containers for data stored in S3.

	Name	AWS Region	IAM Access Analyzer	Creation date
○	bda-mini-project-bucket	US East (N. Virginia) us-east-1	View analyzer for us-east-1	December 10, 2024, 13:38:51 (UTC-05:00)

The command uses default settings (same as the ones we see in the website).

- Uploading files to S3 Bucket :

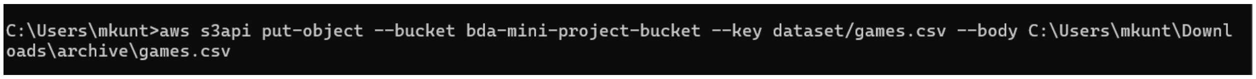
Navigate to the S3 bucket we just created and upload the dataset. Initially, a folder named ‘dataset’ is created and then the dataset is uploaded to this folder.



Objects are the fundamental entities stored in Amazon S3. You can use Amazon S3 inventory to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. Learn more

□	Name	Type	Last modified	Size	Storage class
□	games.csv	csv	December 10, 2024, 16:20:43 (UTC-05:00)	207.4 MB	Standard

Or using AWS CLI,

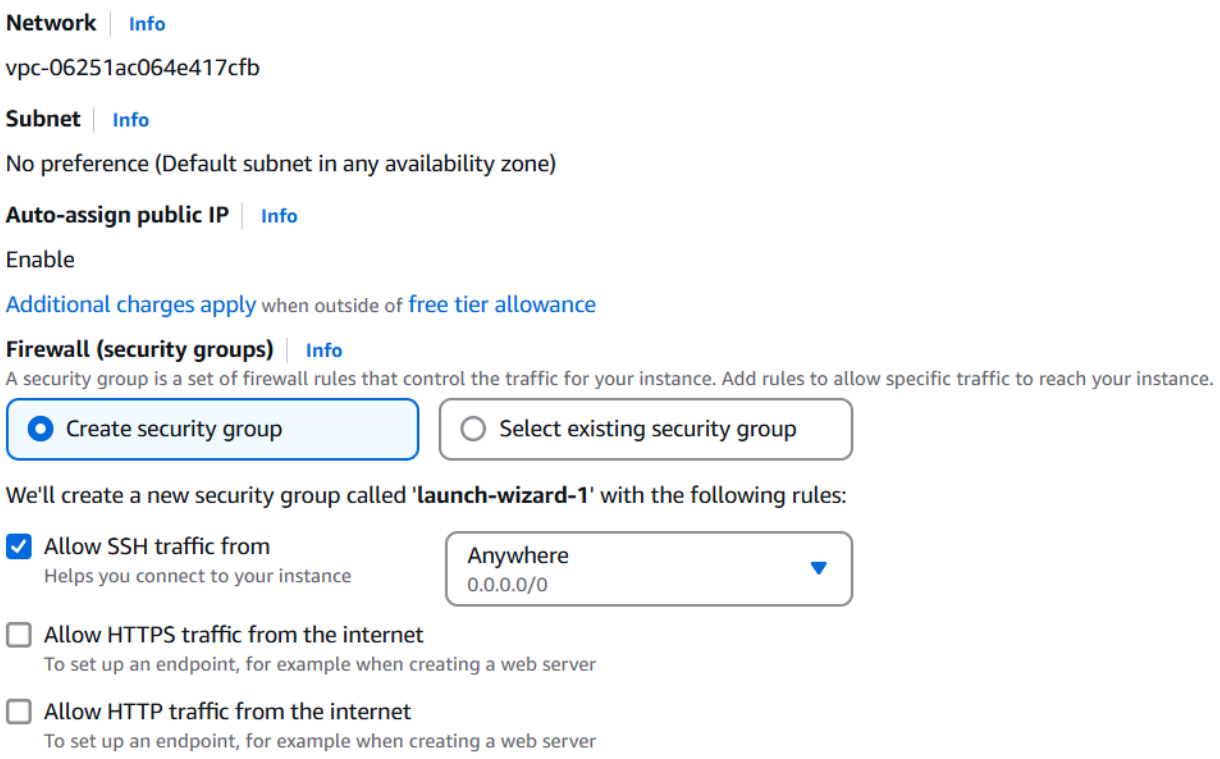


```
C:\Users\mkunt>aws s3api put-object --bucket bda-mini-project-bucket --key dataset/games.csv --body C:\Users\mkunt\Downl
oads\archive\games.csv
```

- Launching AWS EC2 Instance

Navigate to the EC2 section in AWS Console Management, and then Launch Instance. Details of the Instance:

- AMI from catalog : Amazon Linux 2 AMI (HVM) - Kernel 5.10, SSD Volume Type
- Instance type : t2.micro
- Key-pair (login) : New key pair has been created with RSA encoding and the .pem file is downloaded for the same.
- Network settings:



Network | Info

vpc-06251ac064e417cfb

Subnet | Info

No preference (Default subnet in any availability zone)

Auto-assign public IP | Info

Enable

Additional charges apply when outside of free tier allowance

Firewall (security groups) | Info

A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

● Create security group ○ Select existing security group

We'll create a new security group called '**launch-wizard-1**' with the following rules:

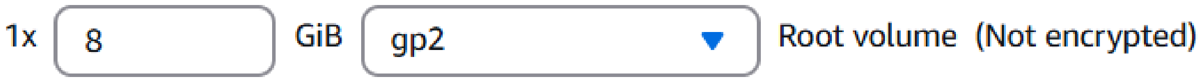
☑ Allow SSH traffic from — Anywhere 0.0.0.0/0
Helps you connect to your instance

☐ Allow HTTPS traffic from the internet
To set up an endpoint, for example when creating a web server

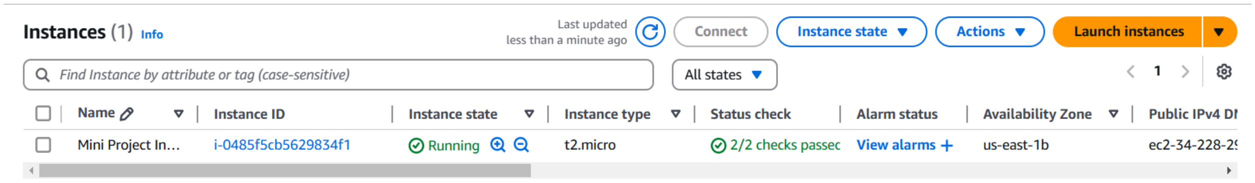
☐ Allow HTTP traffic from the internet
To set up an endpoint, for example when creating a web server

VPC for the network has been used (default) and we have a security group specified with an inbound rule allowing SSH (port 22) from anywhere.

- Storage : Using General purpose SSD gp2



1x 8 GiB gp2 Root volume (Not encrypted)



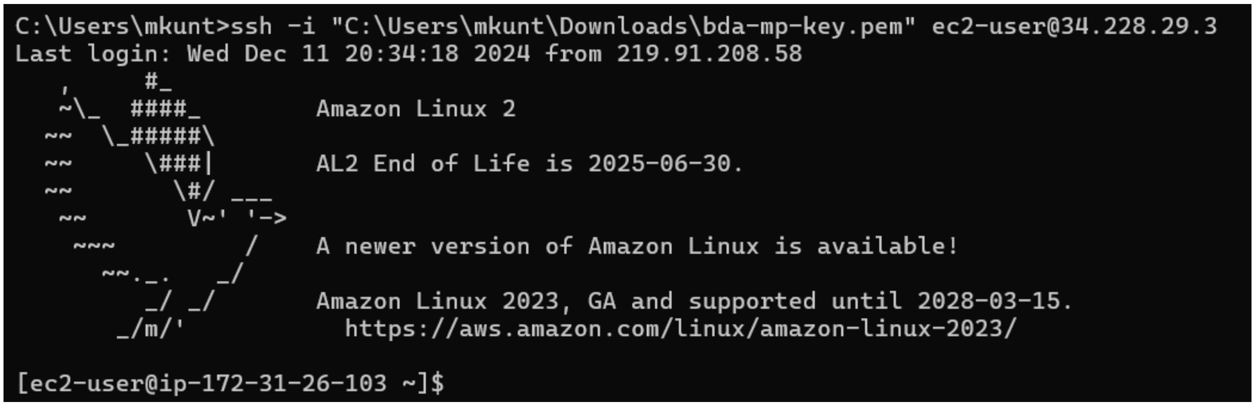
Instances (1) Info

Last updated less than a minute ago

☐	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DI
☐	Mini Project In…	i-0485f5cb5629834f1	✓ Running	t2.micro	✓ 2/2 checks passec	View alarms +	us-east-1b	ec2-34-228-2

Connecting to the Instance from local using this command:

ssh -i "location to.pem file" ec2-user@<Public Ipv4 address>



```
C:\Users\mkunt>ssh -i "C:\Users\mkunt\Downloads\bda-mp-key.pem" ec2-user@34.228.29.3
Last login: Wed Dec 11 20:34:18 2024 from 219.91.208.58
   ,     #_
   ~\_  ####_        Amazon Linux 2
  ~~  \_#####\
  ~~     \###|       AL2 End of Life is 2025-06-30.
  ~~       \#/ ___
   ~~       V~' '->
    ~~~         /    A newer version of Amazon Linux is available!
      ~~._.   _/
         _/ _/       Amazon Linux 2023, GA and supported until 2028-03-15.
       _/m/'           https://aws.amazon.com/linux/amazon-linux-2023/

[ec2-user@ip-172-31-26-103 ~]$
```

- Installing PySpark :

Commands :

sudo yum update -y

sudo yum install -y java-11-amazon-corretto

java -version

wget

https://dlcdn.apache.org/spark/spark-3.5.3/spark-3.5.3-bin-hadoop3.tgz

tar -xvzf spark-3.5.3-bin-hadoop3.tgz

sudo mv spark-3.5.3-bin-hadoop3 /opt/spark

export JAVA_HOME=/usr/lib/jvm/java-11-amazon-corretto

PATH=$JAVA_HOME/bin:$PATH

export SPARK_HOME=/opt/spark export

PATH=$SPARK_HOME/bin:$PATH export

HADOOP_HOME=/usr/lib/hadoop

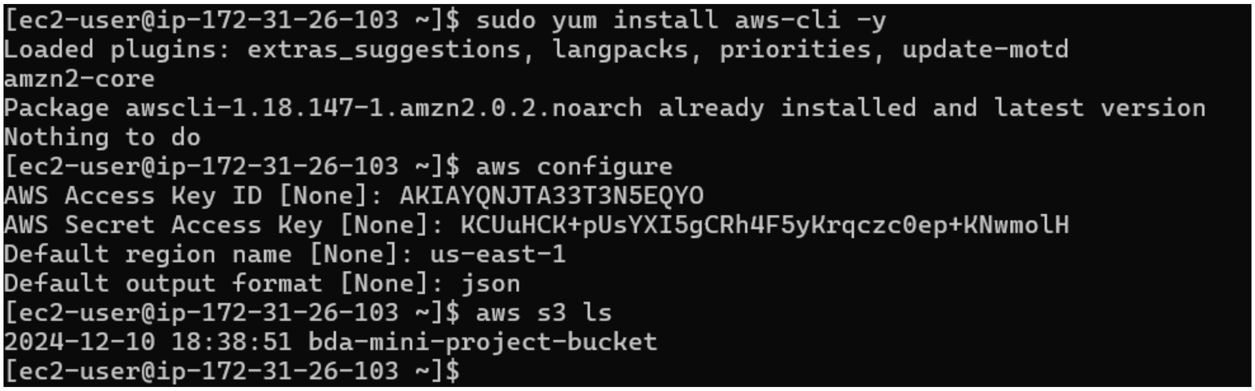
source ~/.bash_profile

sudo yum install python3-pip -y

pip3 install pyspark

These commands will install PySpark accessible through commands : ‘spark-shell’ or ‘pyspark’.

- Configure AWS CLI to interact with S3 buckets:



```
[ec2-user@ip-172-31-26-103 ~]$ sudo yum install aws-cli -y
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
amzn2-core
Package awscli-1.18.147-1.amzn2.0.2.noarch already installed and latest version
Nothing to do
[ec2-user@ip-172-31-26-103 ~]$ aws configure
AWS Access Key ID [None]: AKIAYQNJTA33T3N5EQYO
AWS Secret Access Key [None]: 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
Default region name [None]: us-east-1
Default output format [None]: json
[ec2-user@ip-172-31-26-103 ~]$ aws s3 ls
2024-12-10 18:38:51 bda-mini-project-bucket
[ec2-user@ip-172-31-26-103 ~]$
```

Following these commands:

aws configure

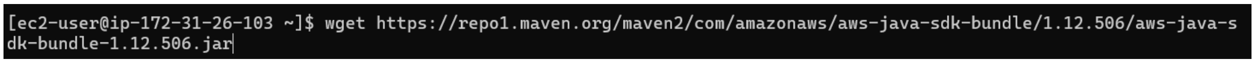
aws s3 ls

This will show the S3 bucket created in the above steps.

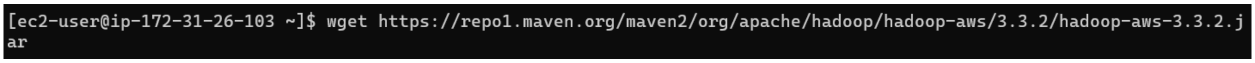
3.2 Data Pipeline

Task 1 : Data Ingestion from S3

We need to download a few jars before we try to run the scripts. So after running this command: *tar -xvzf spark-3.5.3-bin-hadoop3.tgz,* navigate to the jars folder and run these commands to download the required jars to handle aws connections:

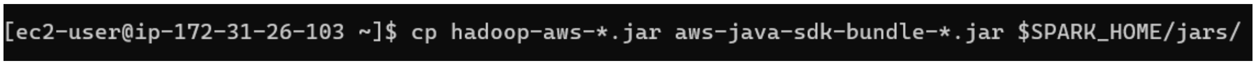


```
[ec2-user@ip-172-31-26-103 ~]$ wget https://repo1.maven.org/maven2/com/amazonaws/aws-java-sdk-bundle/1.12.506/aws-java-s
dk-bundle-1.12.506.jar
```



```
[ec2-user@ip-172-31-26-103 ~]$ wget https://repo1.maven.org/maven2/org/apache/hadoop/hadoop-aws/3.3.2/hadoop-aws-3.3.2.j
ar
```

Now add the jars location to respective Path variable as well :

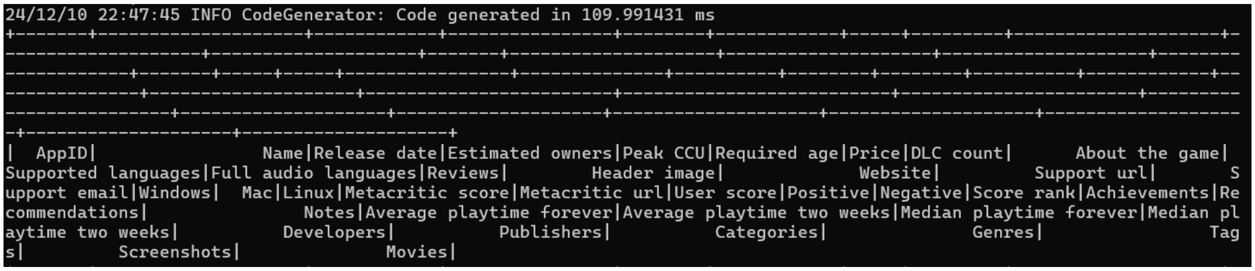


```
[ec2-user@ip-172-31-26-103 ~]$ cp hadoop-aws-*.jar aws-java-sdk-bundle-*.jar $SPARK_HOME/jars/
```

Now a python script for loading the file from S3 bucket is written using vim editor and the file is run using this command:

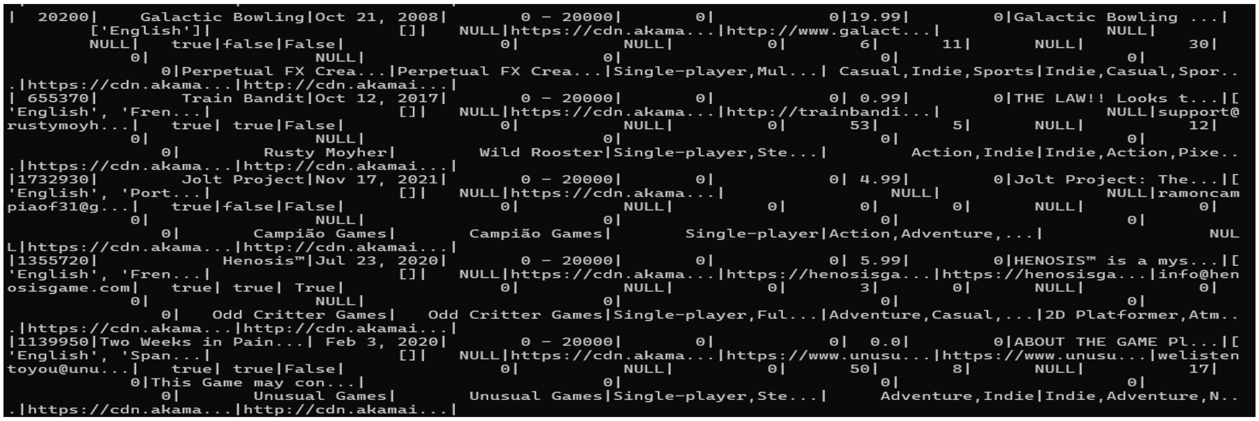
spark-submit S3_load_data.py

Schema of dataset:



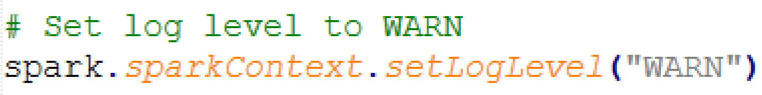
```
24/12/10 22:47:45 INFO CodeGenerator: Code generated in 109.991431 ms
+-------+-------------------+------------+----------------+--------+------------+-----+---------+-------------------+-
-------------------+-------------------+-------+-------------------+-------------------+-------------------+--------
-----------+-------+-----+-----+----------------+-------------+----------+--------+--------+----------+------------+--
------------+-------------------+-------------------------+-------------------------+-------------------+--------
---------------+-------------------+-------------------+-------------------+-------------------+-----------------
-+-------------------+-------------------+
|  AppID|               Name|Release date|Estimated owners|Peak CCU|Required age|Price|DLC count|      About the game|
Supported languages|Full audio languages|Reviews|        Header image|             Website|          Support url|       S
upport email|Windows|  Mac|Linux|Metacritic score|Metacritic url|User score|Positive|Negative|Score rank|Achievements|Re
commendations|              Notes|Average playtime forever|Average playtime two weeks|Median playtime forever|Median pl
aytime two weeks|          Developers|          Publishers|          Categories|              Genres|               Tag
s|         Screenshots|              Movies|
```

Top 5 records of dataset:



```
|  20200|    Galactic Bowling|Oct 21, 2008|        0 - 20000|        0|             0|19.99|          0|Galactic Bowling ...|
        ['English']|                           []|   NULL|https://cdn.akama...|http://www.galact...|                   NULL|
        NULL|   true|false|False|                 0|           NULL|         0|        6|       11|       NULL|          30|
             0|                  NULL|                  0|                    0|                   0|
                0|Perpetual FX Crea...|Perpetual FX Crea...|Single-player,Mul...| Casual,Indie,Sports|Indie,Casual,Spor..
.|https://cdn.akama...|http://cdn.akamai...|
| 655370|        Train Bandit|Oct 12, 2017|        0 - 20000|        0|             0| 0.99|          0|THE LAW!! Looks t...|[
'English', 'Fren...|                           []|   NULL|https://cdn.akama...|http://trainbandi...|                   NULL|support@
rustymoyh...|   true| true|False|                 0|           NULL|         0|       53|        5|       NULL|          12|
             0|                  NULL|                  0|                    0|                   0|
                0|        Rusty Moyher|        Wild Rooster|Single-player,Ste...|        Action,Indie|Indie,Action,Pixe..
.|https://cdn.akama...|http://cdn.akamai...|
|1732930|        Jolt Project|Nov 17, 2021|        0 - 20000|        0|             0| 4.99|          0|Jolt Project: The...|[
'English', 'Port...|                           []|   NULL|https://cdn.akama...|                NULL|                   NULL|ramoncam
piaof31@g...|   true|false|False|                 0|           NULL|         0|        0|        0|       NULL|           0|
             0|                  NULL|                  0|                    0|                   0|
                0|       Campião Games|       Campião Games|       Single-player|Action,Adventure,...|                   NUL
L|https://cdn.akama...|http://cdn.akamai...|
|1355720|            Henosis™|Jul 23, 2020|        0 - 20000|        0|             0| 5.99|          0|HENOSIS™ is a mys...|[
'English', 'Fren...|                           []|   NULL|https://cdn.akama...|https://henosisga...|https://henosisga...|info@hen
osisgame.com|   true| true| True|                 0|           NULL|         0|        3|        0|       NULL|           0|
             0|                  NULL|                  0|                    0|                   0|
                0|   Odd Critter Games|   Odd Critter Games|Single-player,Ful...|Adventure,Casual,...|2D Platformer,Atm..
.|https://cdn.akama...|http://cdn.akamai...|
|1139950|Two Weeks in Pain...| Feb 3, 2020|        0 - 20000|        0|             0|  0.0|          0|ABOUT THE GAME Pl...|[
'English', 'Span...|                           []|   NULL|https://cdn.akama...|https://www.unusu...|https://www.unusu...|welisten
toyou@unu...|   true| true|False|                 0|           NULL|         0|       50|        8|       NULL|          17|
             0|This Game may con...|                  0|                    0|                   0|
                0|       Unusual Games|       Unusual Games|Single-player,Ste...|     Adventure,Indie|Indie,Adventure,N..
.|https://cdn.akama...|http://cdn.akamai...|
```

Note : This code was used to block other logs except for INFO and WARN.



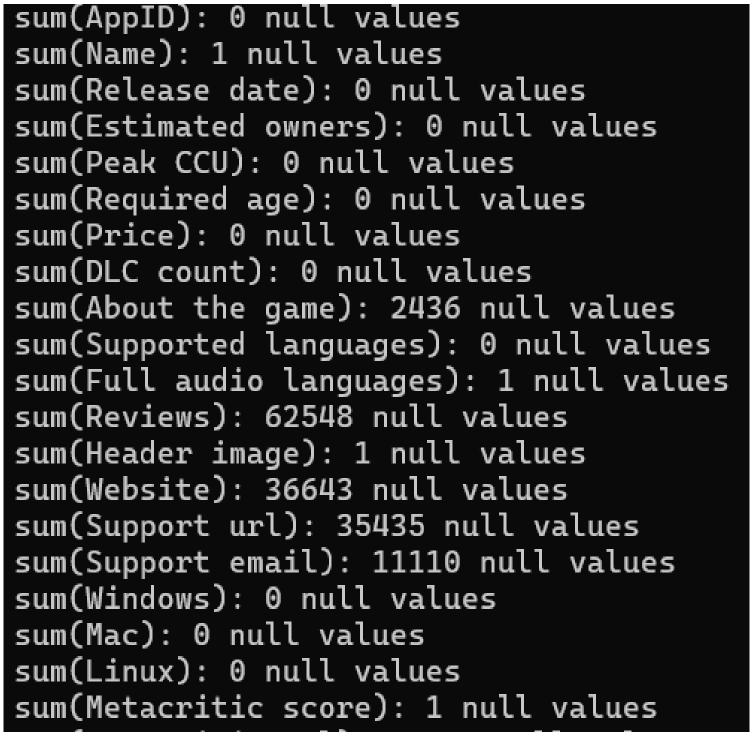
```
# Set log level to WARN
spark.sparkContext.setLogLevel("WARN")
```

Task 2 : Data Processing with Pyspark

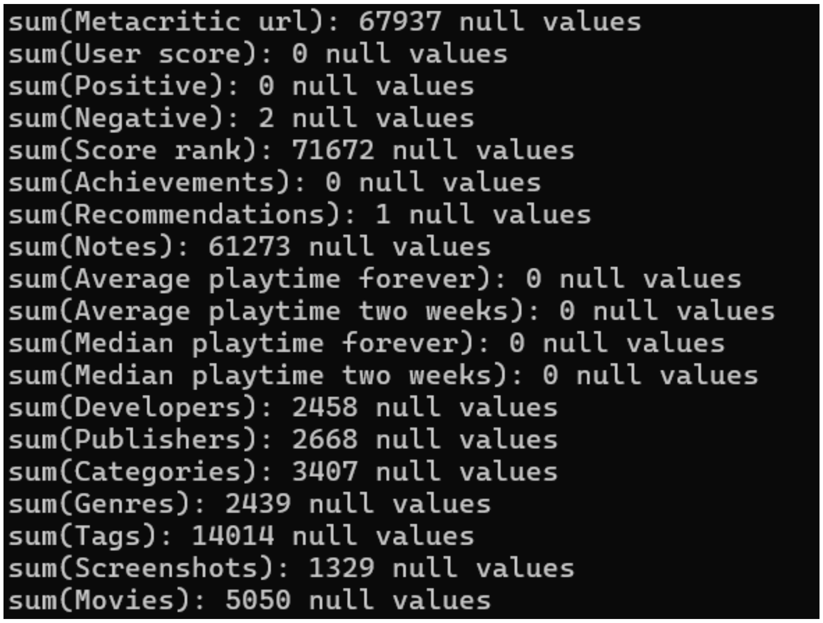
Going step by step :

- Null values handling :

  Null value counts for each column have been checked and these are the results:



```
sum(AppID): 0 null values
sum(Name): 1 null values
sum(Release date): 0 null values
sum(Estimated owners): 0 null values
sum(Peak CCU): 0 null values
sum(Required age): 0 null values
sum(Price): 0 null values
sum(DLC count): 0 null values
sum(About the game): 2436 null values
sum(Supported languages): 0 null values
sum(Full audio languages): 1 null values
sum(Reviews): 62548 null values
sum(Header image): 1 null values
sum(Website): 36643 null values
sum(Support url): 35435 null values
sum(Support email): 11110 null values
sum(Windows): 0 null values
sum(Mac): 0 null values
sum(Linux): 0 null values
sum(Metacritic score): 1 null values
```



```
sum(Metacritic url): 67937 null values
sum(User score): 0 null values
sum(Positive): 0 null values
sum(Negative): 2 null values
sum(Score rank): 71672 null values
sum(Achievements): 0 null values
sum(Recommendations): 1 null values
sum(Notes): 61273 null values
sum(Average playtime forever): 0 null values
sum(Average playtime two weeks): 0 null values
sum(Median playtime forever): 0 null values
sum(Median playtime two weeks): 0 null values
sum(Developers): 2458 null values
sum(Publishers): 2668 null values
sum(Categories): 3407 null values
sum(Genres): 2439 null values
sum(Tags): 14014 null values
sum(Screenshots): 1329 null values
sum(Movies): 5050 null values
```

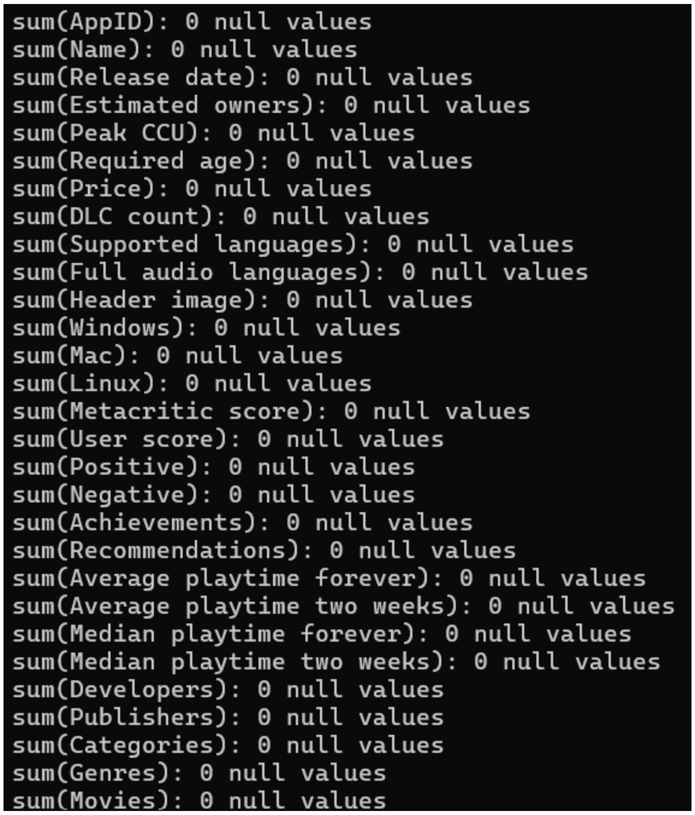
Null handling :

Few columns like Name, Full audio languages, Metacritic score, Recommendations, Negative have very few null values so we just drop those records.

Few other columns like - Support Url, Support email, Metacritic Url, Score rank, Tags, Notes have more null values (considering the dataset size of 71K values) and columns which won’t be useful for our analysis like - About the game, Screenshots are dropped.

There are some columns like - Developers, Publishers, Categories, Genres and Movies which are useful to answer our research question but have null values and almost all of them are Categorical columns. So to counter these nulls, imputing with a placeholder - ‘Unknown’ has been done.

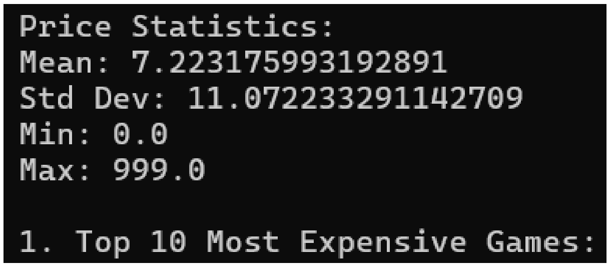
Result :



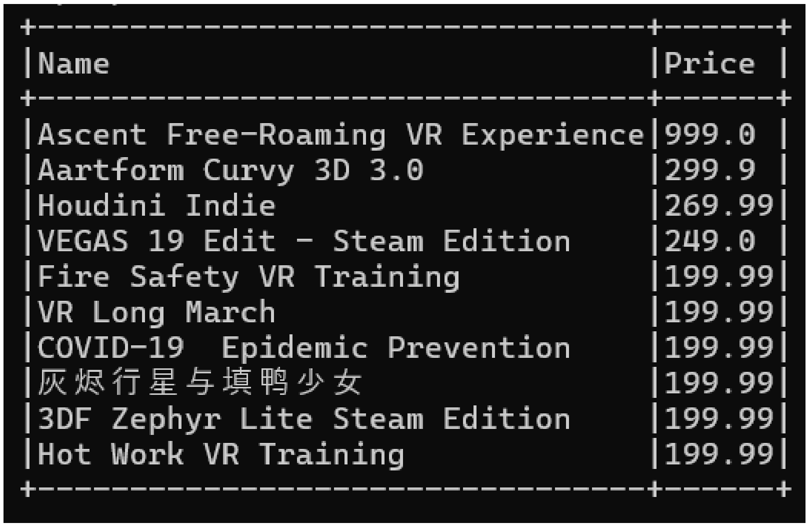
```
sum(AppID): 0 null values
sum(Name): 0 null values
sum(Release date): 0 null values
sum(Estimated owners): 0 null values
sum(Peak CCU): 0 null values
sum(Required age): 0 null values
sum(Price): 0 null values
sum(DLC count): 0 null values
sum(Supported languages): 0 null values
sum(Full audio languages): 0 null values
sum(Header image): 0 null values
sum(Windows): 0 null values
sum(Mac): 0 null values
sum(Linux): 0 null values
sum(Metacritic score): 0 null values
sum(User score): 0 null values
sum(Positive): 0 null values
sum(Negative): 0 null values
sum(Achievements): 0 null values
sum(Recommendations): 0 null values
sum(Average playtime forever): 0 null values
sum(Average playtime two weeks): 0 null values
sum(Median playtime forever): 0 null values
sum(Median playtime two weeks): 0 null values
sum(Developers): 0 null values
sum(Publishers): 0 null values
sum(Categories): 0 null values
sum(Genres): 0 null values
sum(Movies): 0 null values
```

- Outliers check for Target variable - Price :

Identifying the minimum and maximum prices first:



```
Price Statistics:
Mean: 7.223175993192891
Std Dev: 11.072233291142709
Min: 0.0
Max: 999.0

1. Top 10 Most Expensive Games:
```



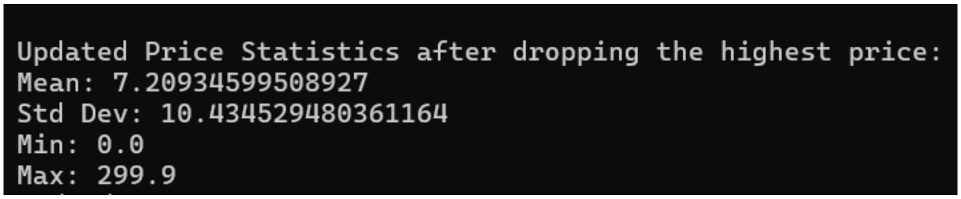
```
+--------------------------------+------+
|Name                            |Price |
+--------------------------------+------+
|Ascent Free-Roaming VR Experience|999.0 |
|Aartform Curvy 3D 3.0           |299.9 |
|Houdini Indie                   |269.99|
|VEGAS 19 Edit - Steam Edition   |249.0 |
|Fire Safety VR Training         |199.99|
|VR Long March                   |199.99|
|COVID-19  Epidemic Prevention   |199.99|
|灰烬行星与填鸭少女               |199.99|
|3DF Zephyr Lite Steam Edition   |199.99|
|Hot Work VR Training            |199.99|
+--------------------------------+------+
```

The game ‘`Ascent Free-Roaming VR Experience`’ is priced at $999, significantly higher than the next top price of $300. While this is an outlier, its classification as a Virtual Reality (VR) game justifies the high price due to the costly development and niche market of VR experiences.

The $999 price is an extreme outlier compared to the next highest price of $300, which is for non-VR software. Retaining it could distort statistical metrics like mean and standard deviation.

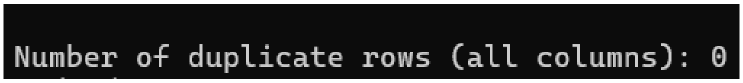
With no other VR games priced similarly, this single data point does not represent a broader trend in the dataset.

Removing the outlier ensures that the dataset reflects typical pricing trends, making insights more generalizable.

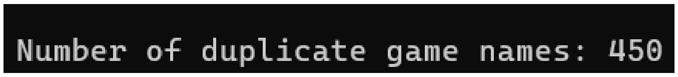


```
Updated Price Statistics after dropping the highest price:
Mean: 7.20934599508927
Std Dev: 10.434529480361164
Min: 0.0
Max: 299.9
```

- <u>Duplicates</u> :



```
Number of duplicate rows (all columns): 0
```



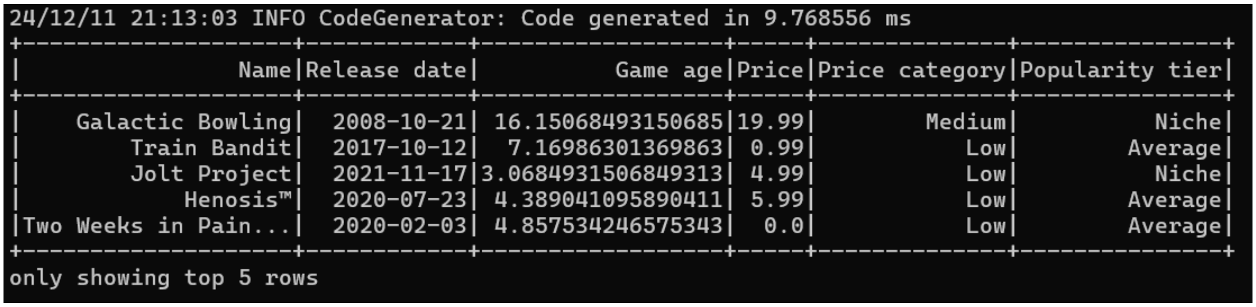
```
Number of duplicate game names: 450
```

There are no duplicate records in the dataset but there are duplicate names of games.

We will not be removing them because these records are not through error but games often have multiple versions or editions, such as special editions, remasters, or different language versions. Each of these could be listed separately in the dataset, leading to duplicate game names.

- <u>Transformations</u> :

Three new columns were added which would support our research question.



```
24/12/11 21:13:03 INFO CodeGenerator: Code generated in 9.768556 ms
+--------------------+------------+------------------+-----+--------------+---------------+
|                Name|Release date|          Game age|Price|Price category|Popularity tier|
+--------------------+------------+------------------+-----+--------------+---------------+
|    Galactic Bowling|  2008-10-21| 16.15068493150685|19.99|        Medium|          Niche|
|        Train Bandit|  2017-10-12|  7.16986301369863| 0.99|           Low|        Average|
|        Jolt Project|  2021-11-17|3.0684931506849313| 4.99|           Low|          Niche|
|            Henosis™|  2020-07-23| 4.389041095890411| 5.99|           Low|        Average|
|Two Weeks in Pain...|  2020-02-03| 4.857534246575343|  0.0|           Low|        Average|
+--------------------+------------+------------------+-----+--------------+---------------+
only showing top 5 rows
```

1. Game age - calculated using the Release date column which gives game age considering the current date.

2. Price category - Based on the price range (< 10$ Low, >=10$ and <=50$ Medium and >50$ High). This categorization helps analyze market segmentation, pricing strategies, and distribution patterns across different game tiers.

3. Popularity tier - considering the columns Positive, Negative and Achievements.

Elite Games: Exceptional titles with overwhelmingly positive reception (P/N ratio ≥ 10) and rich content (100+ achievements)

Popular Games: Well-received games (P/N ratio ≥ 5) with substantial content (50+ achievements)

Rising Games: Promising titles (P/N ratio ≥ 2) with moderate engagement (20+ achievements)

Average Games: Games with more positive than negative reviews (P/N ratio ≥ 1)

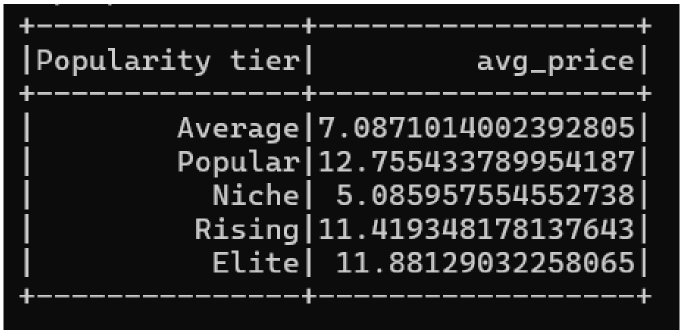
Niche Games: Games with mixed or negative reception or limited engagement

This classification combines user sentiment (Positive/Negative review ratio) with game depth (Achievement count) to create a comprehensive popularity metric.

4. Additionally, two columns - Genres and Categories have been transformed from string to list of strings(split by comma).

- <u>Aggregations</u> :

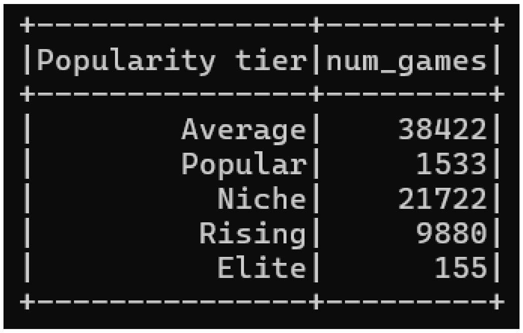
  1. Average Price by Popularity Tier



```
+---------------+-----------------+
|Popularity tier|        avg_price|
+---------------+-----------------+
|        Average|7.087101400239280 5|
|        Popular|12.755433789954187|
|          Niche| 5.085957554552738|
|         Rising|11.419348178137643|
|          Elite| 11.88129032258065|
+---------------+-----------------+
```

Games with a balanced reception are priced moderately at 7.09. Well-received games with many achievements command a reasonable price of around 12.76.

2. Distribution of Games Across Popularity Tiers

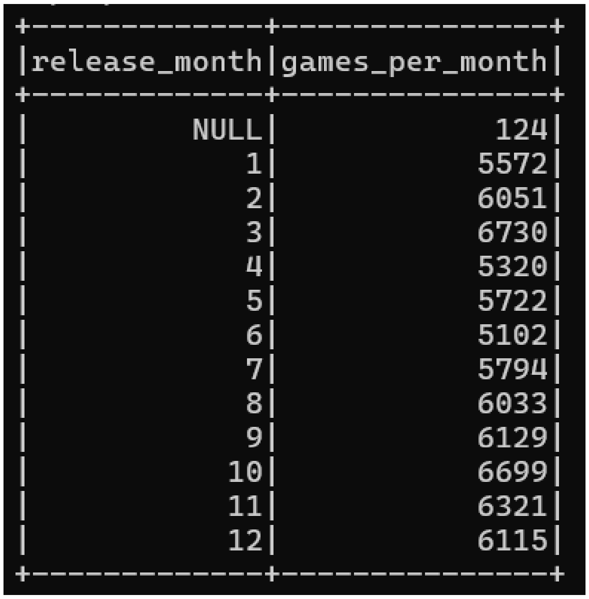


```
+---------------+---------+
|Popularity tier|num_games|
+---------------+---------+
|        Average|    38422|
|        Popular|     1533|
|          Niche|    21722|
|         Rising|     9880|
|          Elite|      155|
+---------------+---------+
```

Average: The majority of games fall into this tier with 38,422 games, indicating a broad base of games with balanced reception.

Niche: A significant number of games, 21,722, are in the niche category, suggesting a large market for specialized or less popular games.

3. Number of Games Released Per Month

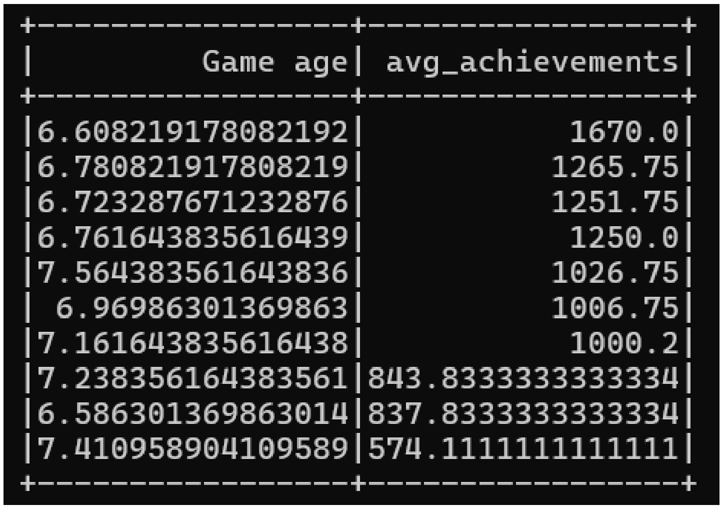


```
+-------------+---------------+
|release_month|games_per_month|
+-------------+---------------+
|         NULL|            124|
|            1|           5572|
|            2|           6051|
|            3|           6730|
|            4|           5320|
|            5|           5722|
|            6|           5102|
|            7|           5794|
|            8|           6033|
|            9|           6129|
|           10|           6699|
|           11|           6321|
|           12|           6115|
+-------------+---------------+
```

NULL: There are 124 games with an unknown release month, possibly due to missing data.

Monthly Distribution: The number of game releases varies, with March having the highest at 6,730 and June the lowest at 5,102.

4. Average Achievements by Game Age

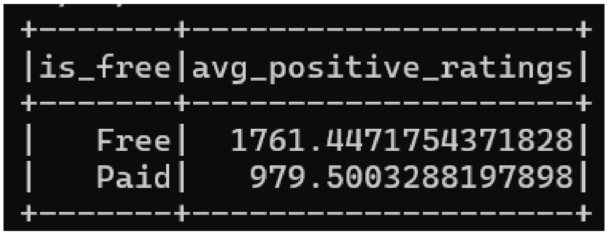


```
+-----------------+-----------------+
|         Game age| avg_achievements|
+-----------------+-----------------+
|6.608219178082192|           1670.0|
|6.780821917808219|          1265.75|
|6.723287671232876|          1251.75|
|6.761643835616439|           1250.0|
|7.564383561643836|          1026.75|
| 6.96986301369863|          1006.75|
|7.161643835616438|           1000.2|
|7.238356164383561|843.8333333333334|
|6.586301369863014|837.8333333333334|
|7.410958904109589|574.1111111111111|
+-----------------+-----------------+
```

Game Age: The top 10 games by average achievements range from 6.61 to 7.56 years old, indicating that older games tend to have more achievements.

Achievements: The highest average is 1,670 for a game aged 6.61 years, suggesting that games with more time on the market accumulate more achievements.

5. Impact of Free-to-Play Models

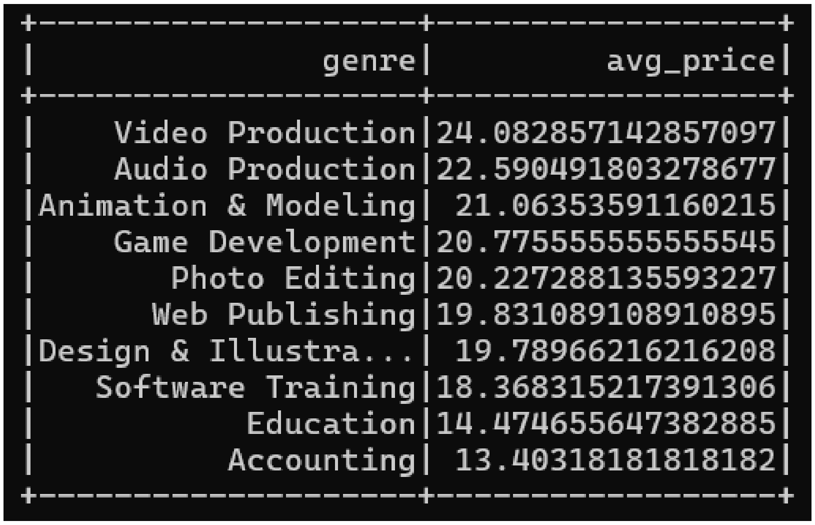


```
+-------+--------------------+
|is_free|avg_positive_ratings|
+-------+--------------------+
|   Free|  1761.4471754371828|
|   Paid|   979.5003288197898|
+-------+--------------------+
```

Free: Free-to-play games have an average of 1,761 positive ratings, indicating higher engagement or popularity among players.

Paid: Paid games, on the other hand, have an average of 980 positive ratings, suggesting that while they might have a dedicated audience, free games attract more players.

6. Average prices per Genre

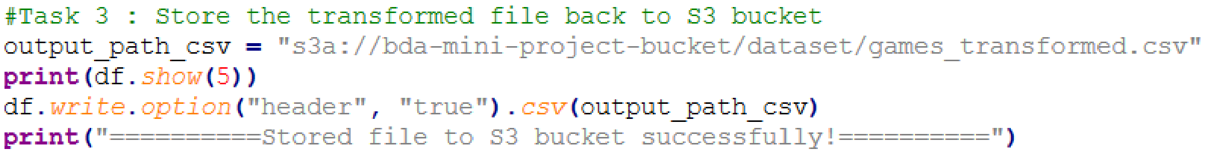


```
+--------------------+------------------+
|               genre|         avg_price|
+--------------------+------------------+
|    Video Production|24.082857142857097|
|    Audio Production|22.590491803278677|
|Animation & Modeling| 21.06353591160215|
|    Game Development|20.775555555555545|
|       Photo Editing|20.227288135593227|
|      Web Publishing|19.831089108910895|
|Design & Illustra...| 19.78966216216208|
|   Software Training|18.368315217391306|
|           Education|14.474655647382885|
|          Accounting| 13.40318181818182|
+--------------------+------------------+
```

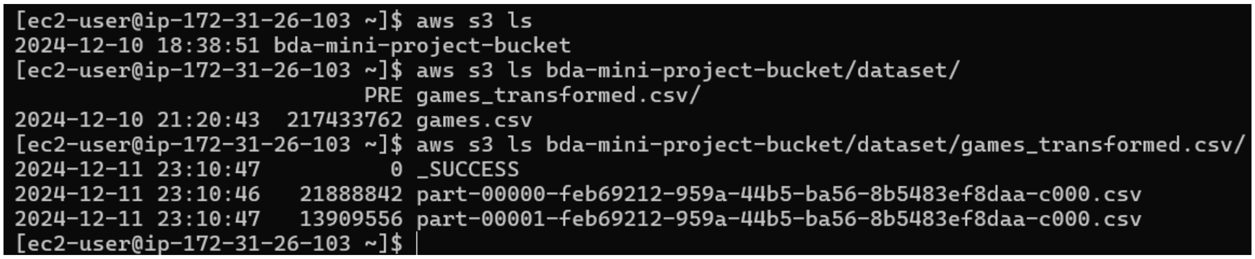
Video Production and Audio Production: Games in these genres have the highest average price, likely due to specialized software or tools included. These games might also include professional tools or features, justifying the higher cost.

Task 3 : Store processed data back to S3 :

By running this script after the transformations, we can store the dataframe back to S3 as a csv file.



```
#Task 3 : Store the transformed file back to S3 bucket
output_path_csv = "s3a://bda-mini-project-bucket/dataset/games_transformed.csv"
print(df.show(5))
df.write.option("header", "true").csv(output_path_csv)
print("==========Stored file to S3 bucket successfully!==========")
```

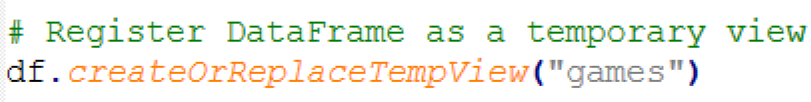


```
[ec2-user@ip-172-31-26-103 ~]$ aws s3 ls
2024-12-10 18:38:51 bda-mini-project-bucket
[ec2-user@ip-172-31-26-103 ~]$ aws s3 ls bda-mini-project-bucket/dataset/
                           PRE games_transformed.csv/
2024-12-10 21:20:43  217433762 games.csv
[ec2-user@ip-172-31-26-103 ~]$ aws s3 ls bda-mini-project-bucket/dataset/games_transformed.csv/
2024-12-11 23:10:47          0 _SUCCESS
2024-12-11 23:10:46   21888842 part-00000-feb69212-959a-44b5-ba56-8b5483ef8daa-c000.csv
2024-12-11 23:10:47   13909556 part-00001-feb69212-959a-44b5-ba56-8b5483ef8daa-c000.csv
[ec2-user@ip-172-31-26-103 ~]$ 
```

But as we can see, the output is not directly a csv file but a folder which has the data split into two part files. This is because Spark is designed for distributed computing, where data is processed in parallel across multiple nodes or cores, and each partition's data is written to a separate file.

Task 4 : Data Analysis using Spark SQL

Analysing the newly transformed dataset by utilizing some SQL queries through PySpark. Before we run queries on the dataframe, we need to create a view of the dataframe :

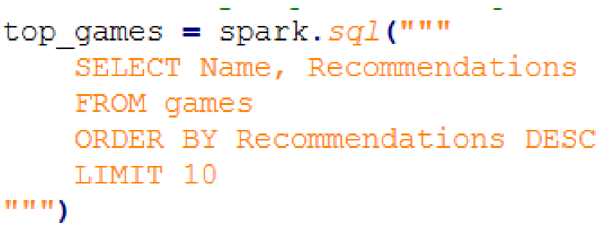


```
# Register DataFrame as a temporary view
df.createOrReplaceTempView("games")
```

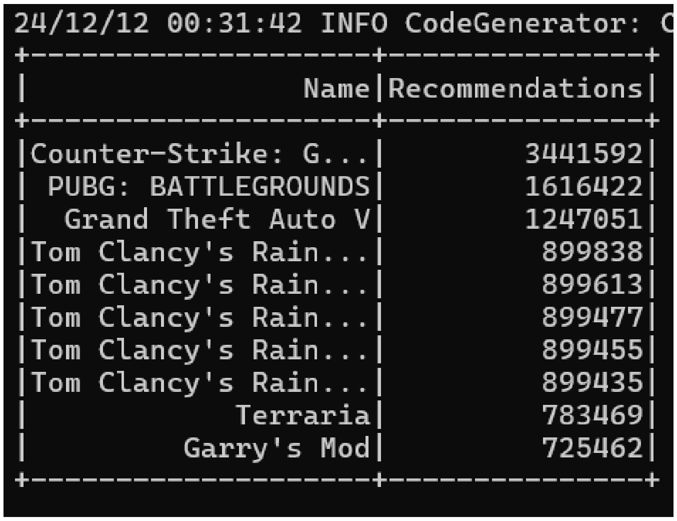
Now to the queries:

- Identifying the top performing games by User Recommendation

  Query:



```
top_games = spark.sql("""
    SELECT Name, Recommendations
    FROM games
    ORDER BY Recommendations DESC
    LIMIT 10
""")
```

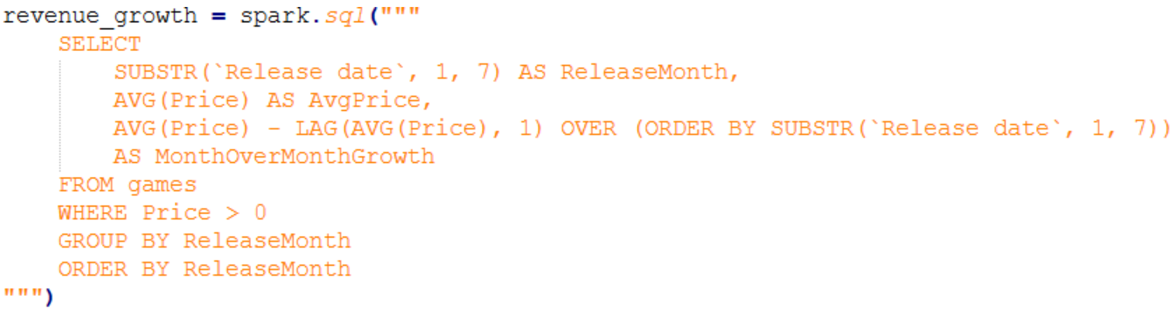


```
24/12/12 00:31:42 INFO CodeGenerator: C
+--------------------+---------------+
|                Name|Recommendations|
+--------------------+---------------+
|Counter-Strike: G...|        3441592|
| PUBG: BATTLEGROUNDS|        1616422|
|  Grand Theft Auto V|        1247051|
|Tom Clancy's Rain...|         899838|
|Tom Clancy's Rain...|         899613|
|Tom Clancy's Rain...|         899477|
|Tom Clancy's Rain...|         899455|
|Tom Clancy's Rain...|         899435|
|            Terraria|         783469|
|         Garry's Mod|         725462|
+--------------------+---------------+
```

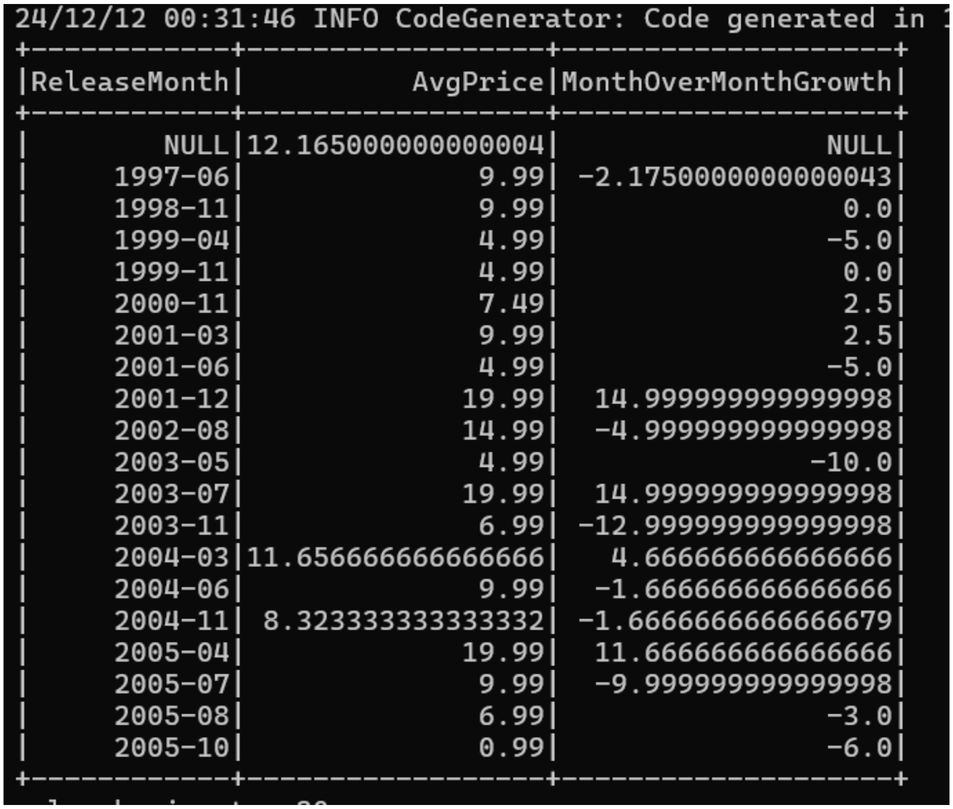
  The Top Performing Games by user recommendations are led by Counter-Strike: Global Offensive with 3,441,592 recommendations, followed by PUBG: BATTLEGROUNDS and Grand Theft Auto V with 1,616,422 and 1,247,051 recommendations respectively.

- Analysing the Month over Month Revenue growth

  Query:



```
revenue_growth = spark.sql("""
    SELECT
        SUBSTR(`Release date`, 1, 7) AS ReleaseMonth,
        AVG(Price) AS AvgPrice,
        AVG(Price) - LAG(AVG(Price), 1) OVER (ORDER BY SUBSTR(`Release date`, 1, 7))
        AS MonthOverMonthGrowth
    FROM games
    WHERE Price > 0
    GROUP BY ReleaseMonth
    ORDER BY ReleaseMonth
""")
```

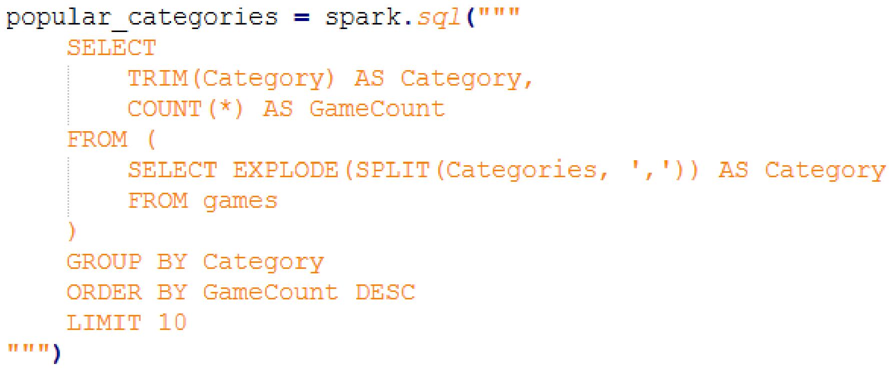


```
24/12/12 00:31:46 INFO CodeGenerator: Code generated in
+------------+------------------+--------------------+
|ReleaseMonth|          AvgPrice|MonthOverMonthGrowth|
+------------+------------------+--------------------+
|        NULL|12.165000000000004|                NULL|
|     1997-06|              9.99| -2.1750000000000043|
|     1998-11|              9.99|                 0.0|
|     1999-04|              4.99|                -5.0|
|     1999-11|              4.99|                 0.0|
|     2000-11|              7.49|                 2.5|
|     2001-03|              9.99|                 2.5|
|     2001-06|              4.99|                -5.0|
|     2001-12|             19.99|  14.999999999999998|
|     2002-08|             14.99|  -4.999999999999998|
|     2003-05|              4.99|               -10.0|
|     2003-07|             19.99|  14.999999999999998|
|     2003-11|              6.99| -12.999999999999998|
|     2004-03|11.656666666666666|   4.666666666666666|
|     2004-06|              9.99|  -1.666666666666666|
|     2004-11| 8.323333333333332| -1.6666666666666679|
|     2005-04|             19.99|  11.666666666666666|
|     2005-07|              9.99|  -9.999999999999998|
|     2005-08|              6.99|                -3.0|
|     2005-10|              0.99|                -6.0|
+------------+------------------+--------------------+
```

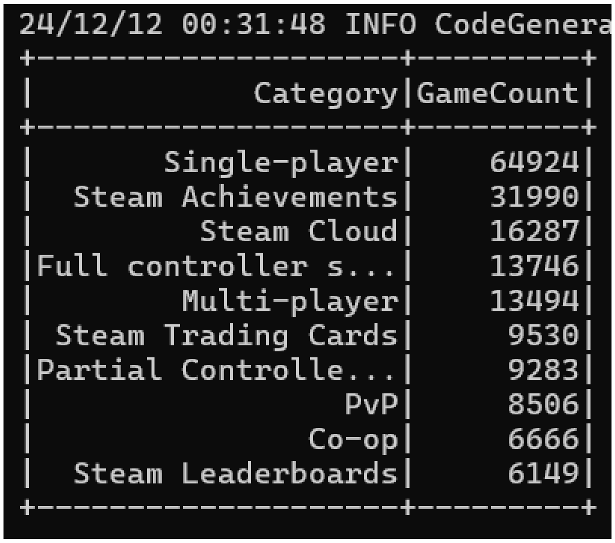
(showing top 20) The Month-Over-Month Revenue Growth analysis shows significant fluctuations in average game prices over time, with notable increases in December 2001 and November 2003, and decreases in May 2003 and October 2005.

- Most popular game Categories

  Query:



```
popular_categories = spark.sql("""
    SELECT
        TRIM(Category) AS Category,
        COUNT(*) AS GameCount
    FROM (
        SELECT EXPLODE(SPLIT(Categories, ',')) AS Category
        FROM games
    )
    GROUP BY Category
    ORDER BY GameCount DESC
    LIMIT 10
""")
```

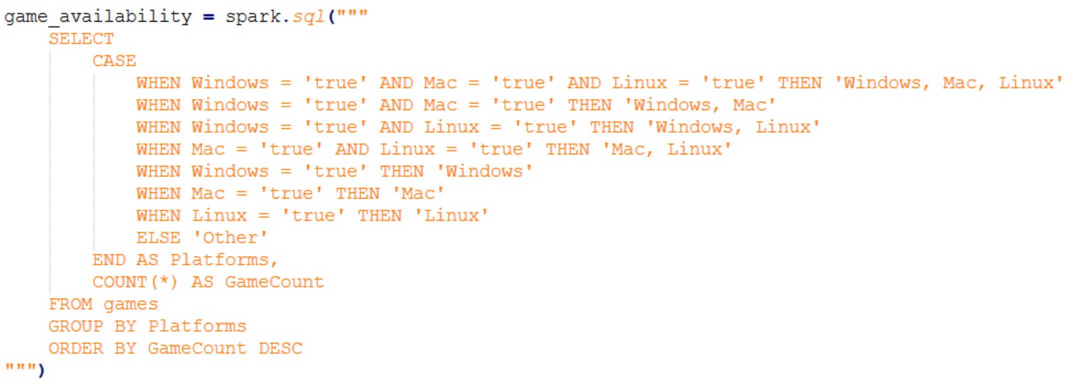


```
24/12/12 00:31:48 INFO CodeGenera
+--------------------+---------+
|            Category|GameCount|
+--------------------+---------+
|       Single-player|    64924|
|  Steam Achievements|    31990|
|         Steam Cloud|    16287|
|Full controller s...|    13746|
|        Multi-player|    13494|
| Steam Trading Cards|     9530|
|Partial Controlle...|     9283|
|                 PvP|     8506|
|               Co-op|     6666|
|  Steam Leaderboards|     6149|
+--------------------+---------+
```

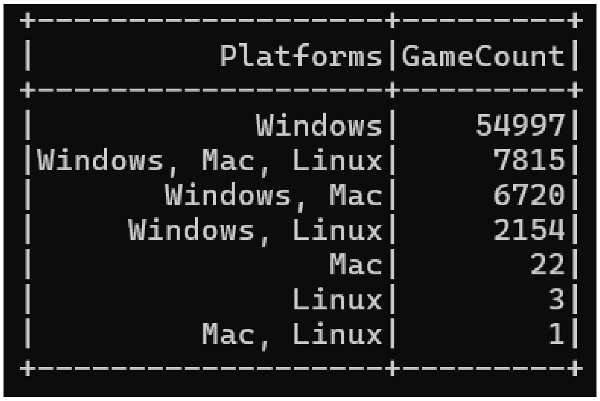
The Most Popular Game Categories are dominated by Single-player games with 64,924 titles, followed by Steam Achievements and Steam Cloud with 31,990 and 16,287 games respectively.

- Analysing the game availability across platforms

  Query:



```
game_availability = spark.sql("""
    SELECT
        CASE
            WHEN Windows = 'true' AND Mac = 'true' AND Linux = 'true' THEN 'Windows, Mac, Linux'
            WHEN Windows = 'true' AND Mac = 'true' THEN 'Windows, Mac'
            WHEN Windows = 'true' AND Linux = 'true' THEN 'Windows, Linux'
            WHEN Mac = 'true' AND Linux = 'true' THEN 'Mac, Linux'
            WHEN Windows = 'true' THEN 'Windows'
            WHEN Mac = 'true' THEN 'Mac'
            WHEN Linux = 'true' THEN 'Linux'
            ELSE 'Other'
        END AS Platforms,
        COUNT(*) AS GameCount
    FROM games
    GROUP BY Platforms
    ORDER BY GameCount DESC
""")
```

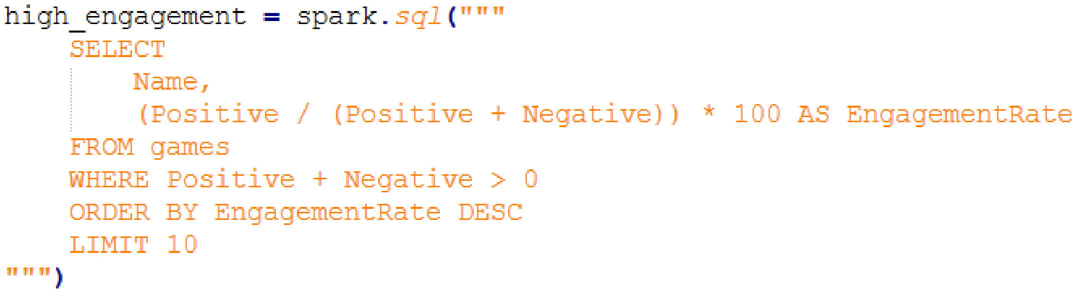


```
+-------------------+---------+
|          Platforms|GameCount|
+-------------------+---------+
|            Windows|    54997|
|Windows, Mac, Linux|     7815|
|       Windows, Mac|     6720|
|     Windows, Linux|     2154|
|                Mac|       22|
|              Linux|        3|
|         Mac, Linux|        1|
+-------------------+---------+
```

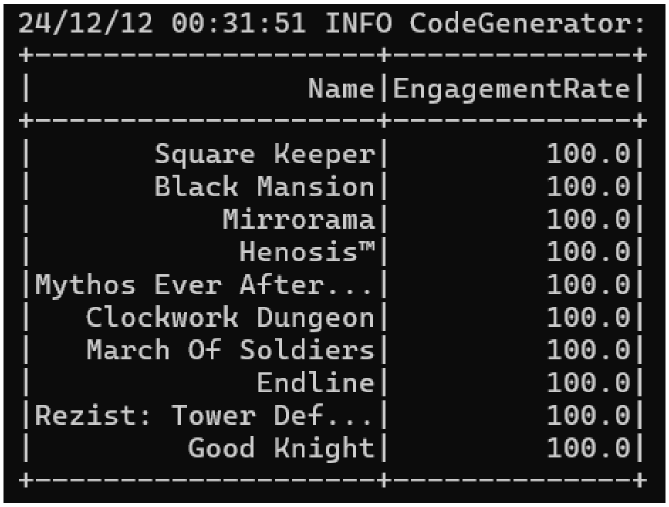
The Game Availability Across Platforms shows that the majority of games are available on Windows alone with 54,997 titles, while cross-platform availability is less common, with Windows, Mac, Linux having 7,815 games.

- Identify games with higher user engagement

  Query:



```
high_engagement = spark.sql("""
    SELECT
        Name,
        (Positive / (Positive + Negative)) * 100 AS EngagementRate
    FROM games
    WHERE Positive + Negative > 0
    ORDER BY EngagementRate DESC
    LIMIT 10
""")
```

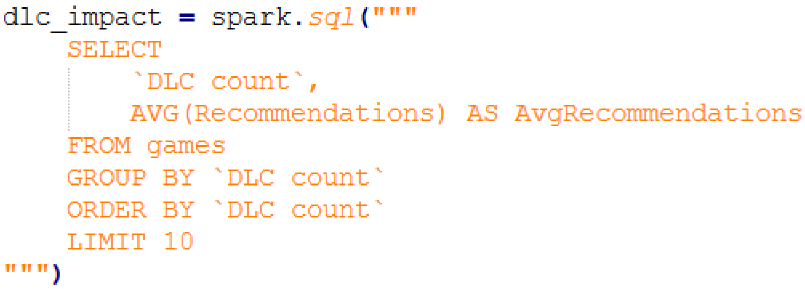


```
24/12/12 00:31:51 INFO CodeGenerator:
+--------------------+--------------+
|                Name|EngagementRate|
+--------------------+--------------+
|       Square Keeper|         100.0|
|       Black Mansion|         100.0|
|           Mirrorama|         100.0|
|            Henosis™|         100.0|
|Mythos Ever After...|         100.0|
|    Clockwork Dungeon|         100.0|
|    March Of Soldiers|         100.0|
|             Endline|         100.0|
|Rezist: Tower Def...|         100.0|
|         Good Knight|         100.0|
+--------------------+--------------+
```

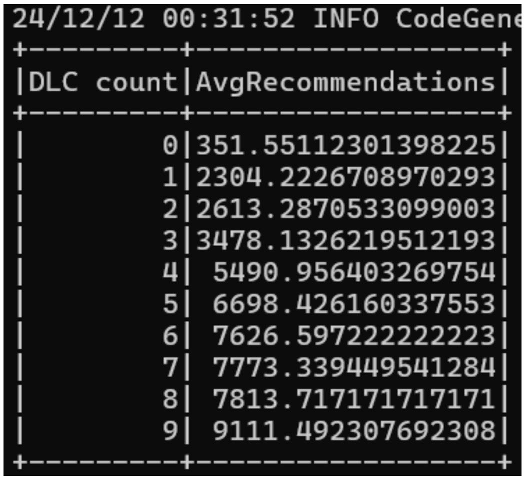
The result where the top games have an EngagementRate of 100% suggests that for these games, there were no negative reviews (Negative = 0), or the data might be skewed in some way.

- Impact of DLC count on Recommendations

  Query:



```
dlc_impact = spark.sql("""
    SELECT
        `DLC count`,
        AVG(Recommendations) AS AvgRecommendations
    FROM games
    GROUP BY `DLC count`
    ORDER BY `DLC count`
    LIMIT 10
""")
```



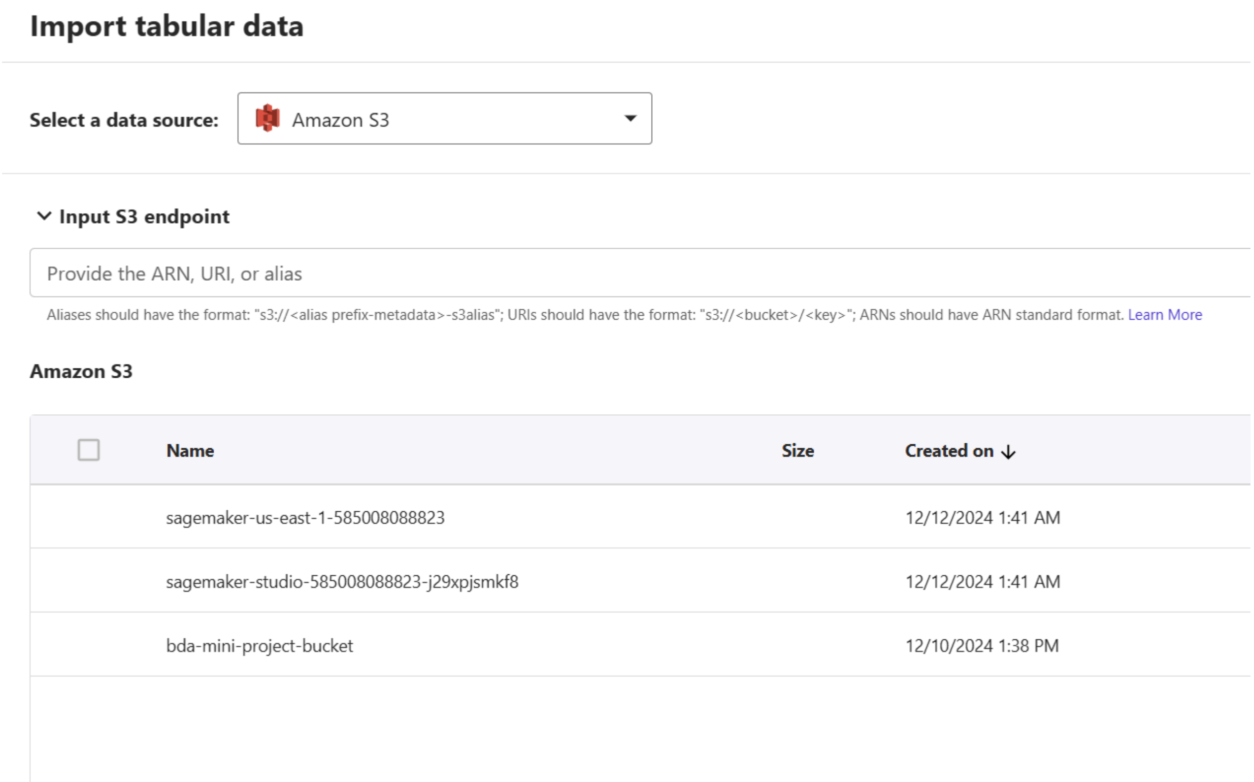
```
24/12/12 00:31:52 INFO CodeGene
+---------+-----------------+
|DLC count|AvgRecommendations|
+---------+-----------------+
|        0|351.55112301398225|
|        1|2304.2226708970293|
|        2|2613.2870533099003|
|        3|3478.1326219512193|
|        4| 5490.956403269754|
|        5| 6698.426160337553|
|        6| 7626.597222222223|
|        7| 7773.339449541284|
|        8| 7813.717171717171|
|        9| 9111.492307692308|
+---------+-----------------+
```

The Impact of DLC(Downloadable Content) Count on Recommendations shows a positive correlation, with average recommendations increasing significantly as the number of DLCs rises, peaking at 9 DLCs with over 9,000 average recommendations.

Task 5 : Machine Learning with AWS SageMaker Autopilot

- SageMaker Canvas - Amazon SageMaker AI offers to generate accurate machine learning predictions — no code required. We need to first setup for a single user and then we can access Canvas, where we can import the data from S3 bucket, transform it and store it to Canvas, build models using this dataset, make predictions and finally deploy the model.
- Canvas : Data wrangler - to load and combine data and then import to canvas.

  Select the data source as Amazon S3 as shown below and we can find the buckets.



Import tabular data

Select a data source: Amazon S3

Input S3 endpoint

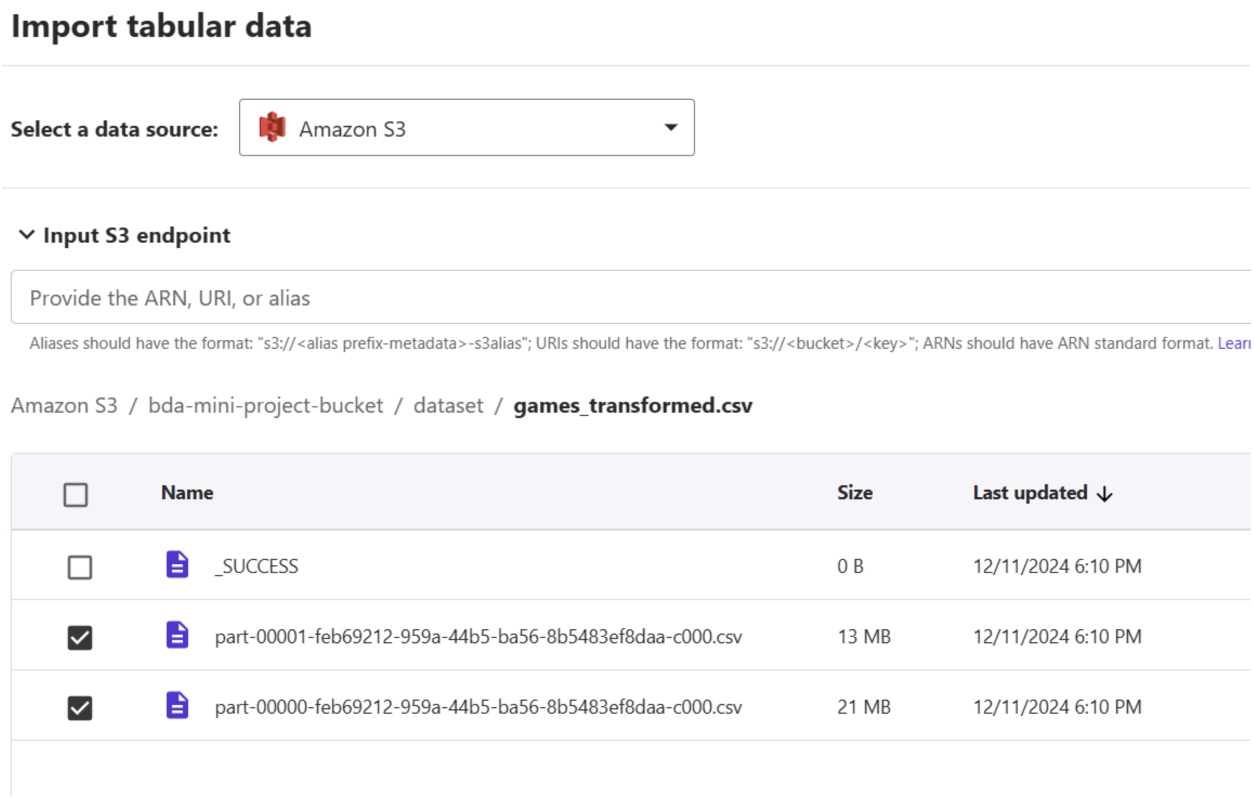
Provide the ARN, URI, or alias

Aliases should have the format: "s3://<alias prefix-metadata>-s3alias"; URIs should have the format: "s3://<bucket>/<key>"; ARNs should have ARN standard format. Learn More

Amazon S3

□	Name	Size	Created on ↓
	sagemaker-us-east-1-585008088823		12/12/2024 1:41 AM
	sagemaker-studio-585008088823-j29xpjsmkf8		12/12/2024 1:41 AM
	bda-mini-project-bucket		12/10/2024 1:38 PM

Navigated to the location where my part files are and selected both of them at once, which would automatically be combined into a single file by Data Wrangler.



Import tabular data

Select a data source: Amazon S3

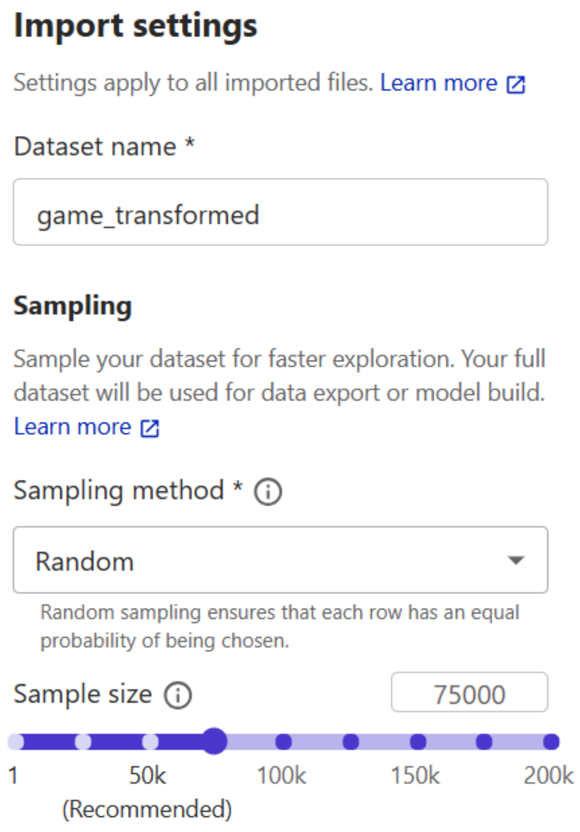
Input S3 endpoint

Provide the ARN, URI, or alias

Aliases should have the format: "s3://<alias prefix-metadata>-s3alias"; URIs should have the format: "s3://<bucket>/<key>"; ARNs should have ARN standard format. Learn

Amazon S3 / bda-mini-project-bucket / dataset / **games_transformed.csv**

□	Name	Size	Last updated ↓
□	_SUCCESS	0 B	12/11/2024 6:10 PM
☑	part-00001-feb69212-959a-44b5-ba56-8b5483ef8daa-c000.csv	13 MB	12/11/2024 6:10 PM
☑	part-00000-feb69212-959a-44b5-ba56-8b5483ef8daa-c000.csv	21 MB	12/11/2024 6:10 PM



Import settings

Settings apply to all imported files. Learn more

Dataset name *

game_transformed

Sampling

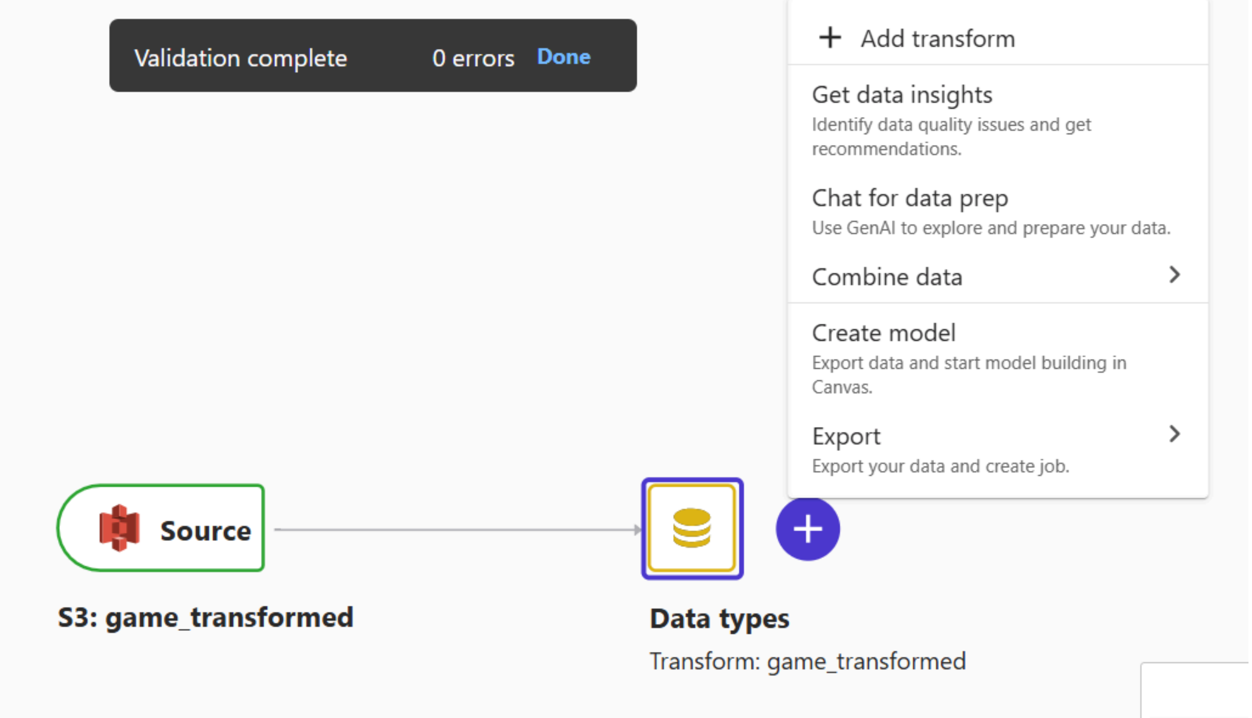
Sample your dataset for faster exploration. Your full dataset will be used for data export or model build. Learn more

Sampling method * ⓘ

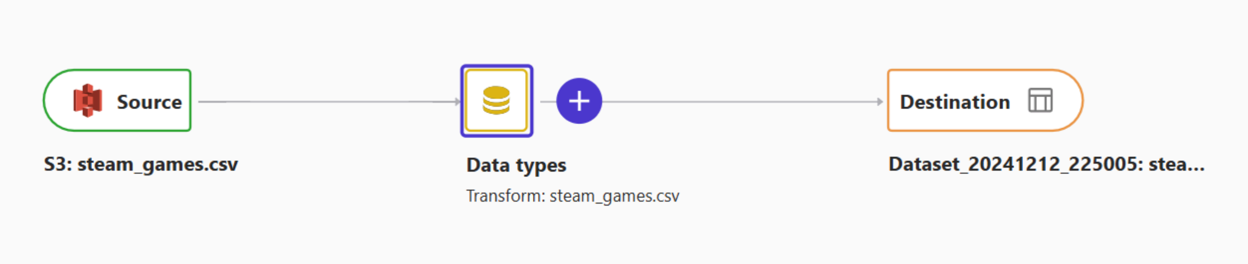
Random

Random sampling ensures that each row has an equal probability of being chosen.

Sample size ⓘ 75000


Validation complete
0 errors
Done
Add transform
Get data insights
Identify data quality issues and get recommendations.
Chat for data prep
Use GenAI to explore and prepare your data.
Combine data
Create model
Export data and start model building in Canvas.
Export
Export your data and create job.
Source
S3: game_transformed
Data types
Transform: game_transformed

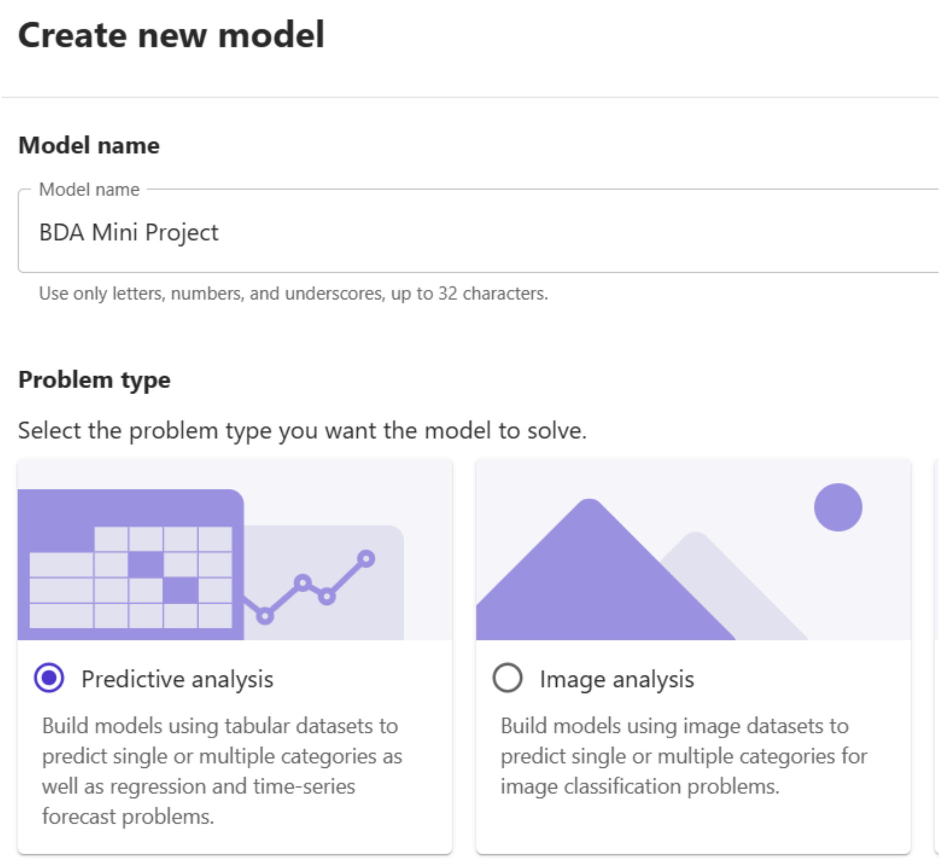
Once the import is done from S3 bucket, this the view of the data flow. Now the next step is to export this data. When we click on the plus icon as shown above, we can see the Export options - Export to Canvas Datasets.



Now we can access the dataset in Canvas.

- Building a model

  Once the dataset is ready and available in Canvas, the next step is to create a model. This model will employ predictive analysis.



Create new model

Model name

Model name: BDA Mini Project

Use only letters, numbers, and underscores, up to 32 characters.

Problem type

Select the problem type you want the model to solve.

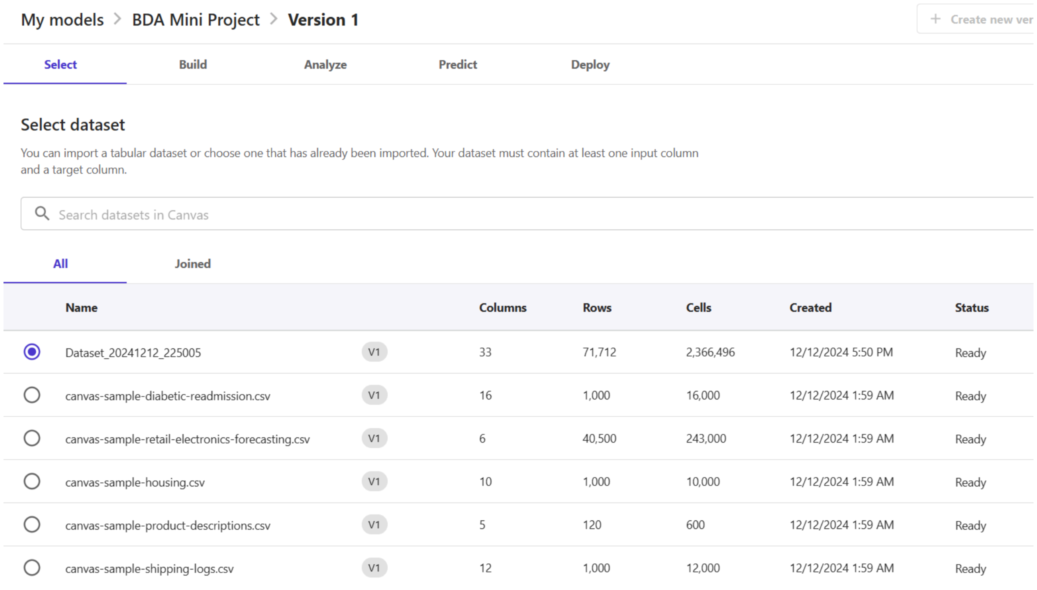
● Predictive analysis

Build models using tabular datasets to predict single or multiple categories as well as regression and time-series forecast problems.

○ Image analysis

Build models using image datasets to predict single or multiple categories for image classification problems.

We can select the dataset we just exported into Canvas as shown below:



Select dataset

You can import a tabular dataset or choose one that has already been imported. Your dataset must contain at least one input column and a target column.

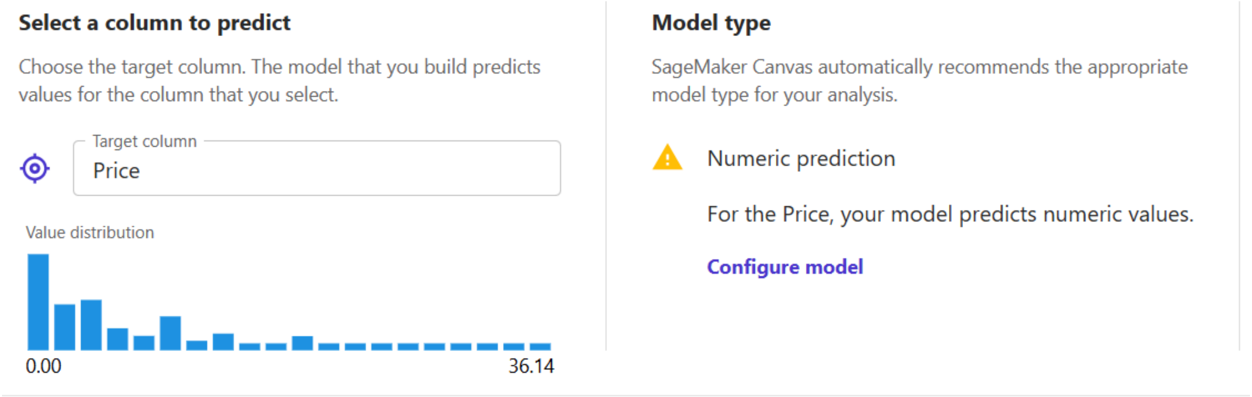
Search datasets in Canvas

	Name		Columns	Rows	Cells	Created	Status
●	Dataset_20241212_225005	V1	33	71,712	2,366,496	12/12/2024 5:50 PM	Ready
○	canvas-sample-diabetic-readmission.csv	V1	16	1,000	16,000	12/12/2024 1:59 AM	Ready
○	canvas-sample-retail-electronics-forecasting.csv	V1	6	40,500	243,000	12/12/2024 1:59 AM	Ready
○	canvas-sample-housing.csv	V1	10	1,000	10,000	12/12/2024 1:59 AM	Ready
○	canvas-sample-product-descriptions.csv	V1	5	120	600	12/12/2024 1:59 AM	Ready
○	canvas-sample-shipping-logs.csv	V1	12	1,000	12,000	12/12/2024 1:59 AM	Ready

The next steps would be to select the Target column - Based on the information available from the dataset about steam games, the goal here is to predict the Prices of these games. So Price is selected as the target variable.

Following this, we can select the Model type. The model type suited for this analysis is Numeric type.

We can also go with Time series prediction as this was also another option recommended by Autopilot but our analysis only focuses on predicting the Price of the game using the details of the games and not solely focusing on the time or day the game was released. This is a Data-driven decision where the primary interest is in non-temporal features.



Select a column to predict

Choose the target column. The model that you build predicts values for the column that you select.

Target column: Price

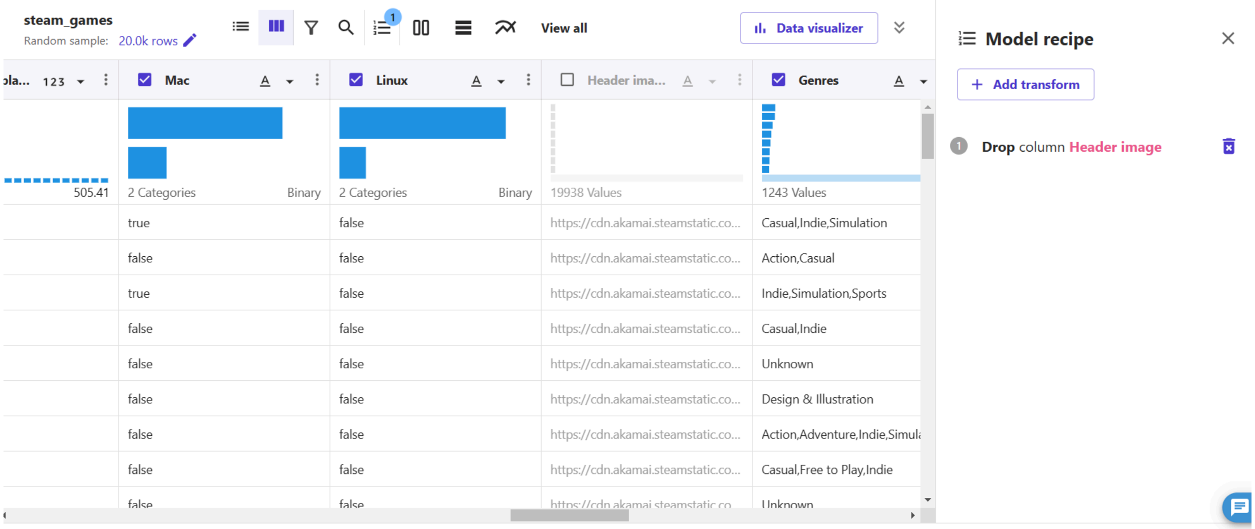
Model type

SageMaker Canvas automatically recommends the appropriate model type for your analysis.

⚠ Numeric prediction

For the Price, your model predicts numeric values.

Additionally, I want to drop the Header image column as it is not useful for predicting the Price of a game.



steam_games

Random sample: 20.0k rows

pla... 123	☑ Mac	☑ Linux	□ Header ima...	☑ Genres
505.41	2 Categories Binary	2 Categories Binary	19938 Values	1243 Values
	true	false	https://cdn.akamai.steamstatic.co...	Casual,Indie,Simulation
	false	false	https://cdn.akamai.steamstatic.co...	Action,Casual
	true	false	https://cdn.akamai.steamstatic.co...	Indie,Simulation,Sports
	false	false	https://cdn.akamai.steamstatic.co...	Casual,Indie
	false	false	https://cdn.akamai.steamstatic.co...	Unknown
	false	false	https://cdn.akamai.steamstatic.co...	Design & Illustration
	false	false	https://cdn.akamai.steamstatic.co...	Action,Adventure,Indie,Simul
	false	false	https://cdn.akamai.steamstatic.co...	Casual,Free to Play,Indie
	false	false	https://cdn.akamai.steamstatic.co	Unknown

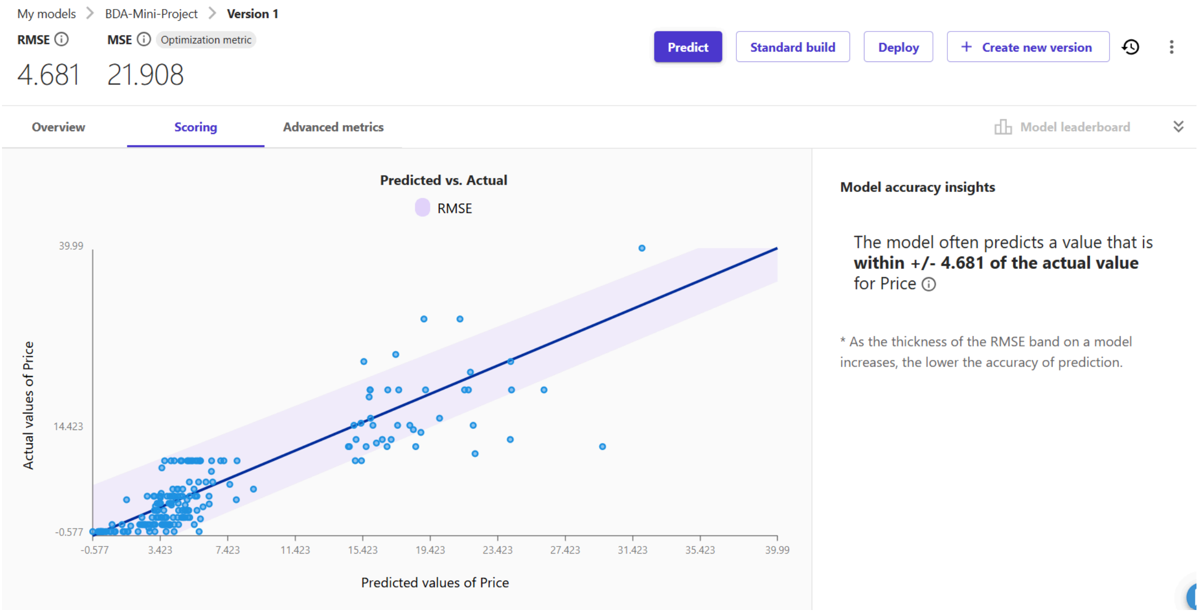
Model recipe

1 Drop column Header image

Header image was unchecked from the dataset view and Autopilot automatically added the Model recipe as shown.

And now we perform a Quick Build on this dataset. Initially some pre-processing would be done and then the model creation would begin.

The results are as follows.



My models > BDA-Mini-Project > Version 1

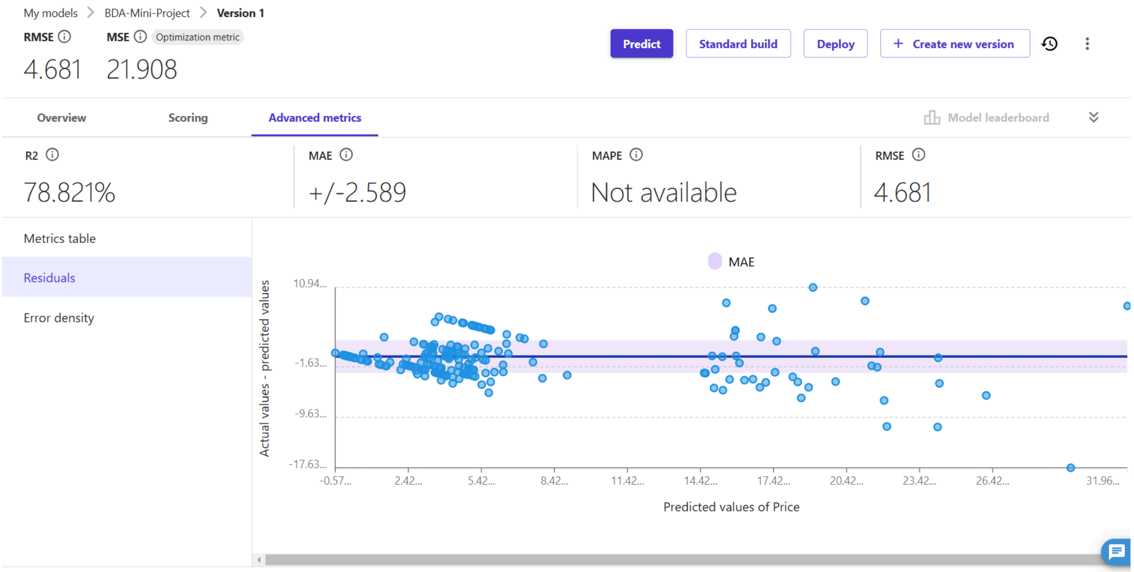
RMSE 4.681

MSE (Optimization metric) 21.908

Model accuracy insights

The model often predicts a value that is **within +/- 4.681 of the actual value** for Price

* As the thickness of the RMSE band on a model increases, the lower the accuracy of prediction.



My models > BDA-Mini-Project > **Version 1**

RMSE: 4.681

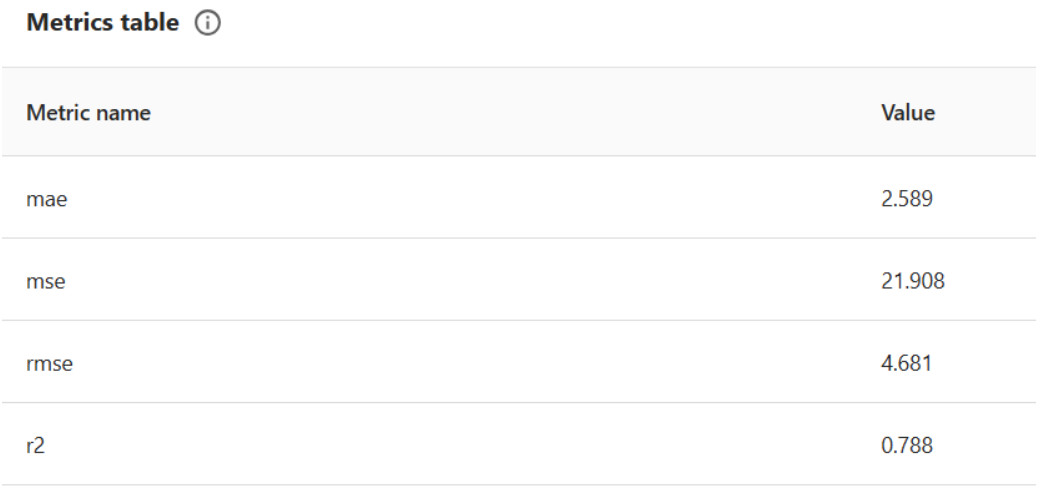
MSE (Optimization metric): 21.908

R2: 78.821%

MAE: +/-2.589

MAPE: Not available

RMSE: 4.681



Metrics table

Metric name	Value
mae	2.589
mse	21.908
rmse	4.681
r2	0.788

Metrics Explanation:

- Mean Absolute Error (MAE) indicates that, on average, the predictions are off by about 2.589 units from the actual price.
- Mean Squared Error (MSE) gives a higher penalty to larger errors. This value suggests there might be some significant

outliers or that some predictions are quite far off from the true values.

- Root Mean Squared Error (RMSE) is in the same unit as the target variable (Price). Here, it's slightly higher than MAE, indicating that there's some variance in the error magnitudes.
- R-squared (R^2) is a measure of how well the model fits the data. An R^2 of 0.788 means that 78.8% of the variance in the Price can be explained by the model, which is quite good, indicating a strong fit.

3.3 Visualization using AWS QuickSight

Initially we need to create a QuickSight Account. Once we are in, we need to load the dataset from S3 bucket. But before we connect to our bucket, we need to have access enabled from QuickSight to S3 (can be done while registering an account or through Manage QuickSight)

For this, navigate to Datasets and click on New Dataset.

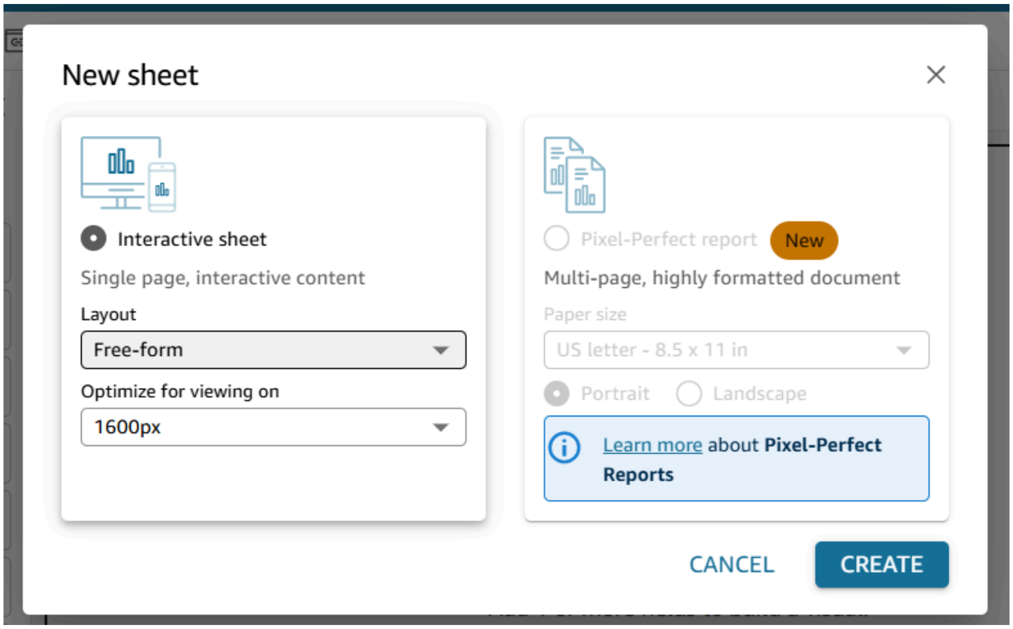
Choose S3 option, and entered the following details:

Name : steam_games

Connecting to S3 bucket through manifest file - a JSON file used to connect to S3 Bucket.

```
{
  "fileLocations": [
    {
      "URIPrefixes": [
"https://s3-us-east-1.amazonaws.com/bda-mini-project-bucket/dataset/games_transformed.csv/"
      ]
    }
  ],
  "globalUploadSettings": {
    "textqualifier": "\""
  }
}
```

Now a new sheet is created using this dataset we loaded (it automatically considers both part files)



New sheet

● Interactive sheet

Single page, interactive content

Layout: Free-form

Optimize for viewing on: 1600px

○ Pixel-Perfect report New

Multi-page, highly formatted document

Paper size: US letter - 8.5 x 11 in

● Portrait ○ Landscape

Learn more about **Pixel-Perfect Reports**

Visualizations based on the transformed data have been created:

- Sum of Achievements by Price and Peak CCU

There's a concentration of games with lower prices and fewer achievements, suggesting many indie or casual games in the dataset.

A few outliers exist with high achievement counts and moderate prices, possibly indicating feature-rich games or long-running titles with multiple updates.

- Sum of Game Age and Achievements by Popularity Tier

More popular games tend to have a higher sum of achievements, which could indicate that popular games often provide more content or replayability.

The "Elite" tier shows the highest sum of game age, suggesting that long-standing games often reach top popularity levels.

- Sum of Price by Developers and Popularity Tier

Certain developers consistently produce games in higher popularity tiers, possibly indicating successful studios or franchises.

There's significant price variation within each popularity tier, showing that price alone doesn't determine a game's popularity.

- Sum of DLC Count by Negative and Positive Reviews

Games with more positive reviews tend to have a higher DLC count, possibly indicating successful games that warrant additional content.

There's a cluster of games with low negative reviews and varying amounts of DLC, suggesting that DLC doesn't necessarily correlate with negative reception.

- Sum of Price by Windows/Linux/Mac and Price Category

Windows has the highest total price sum across all categories, reflecting its dominance in the gaming market.

Linux and Mac show similar patterns but with lower overall price sums, indicating fewer games or lower-priced options for these platforms.

The "High" price category has the largest price sum for all platforms, suggesting that premium-priced games are available across operating systems.

- Sum of Average/Median Playtime Forever and PN Ratio by Release Date

There's a general upward trend in playtime metrics over time, possibly indicating that newer games are designed for longer engagement.

The PN (Positive to Negative) ratio fluctuates but shows an overall increase, suggesting improving game quality or user satisfaction over time.

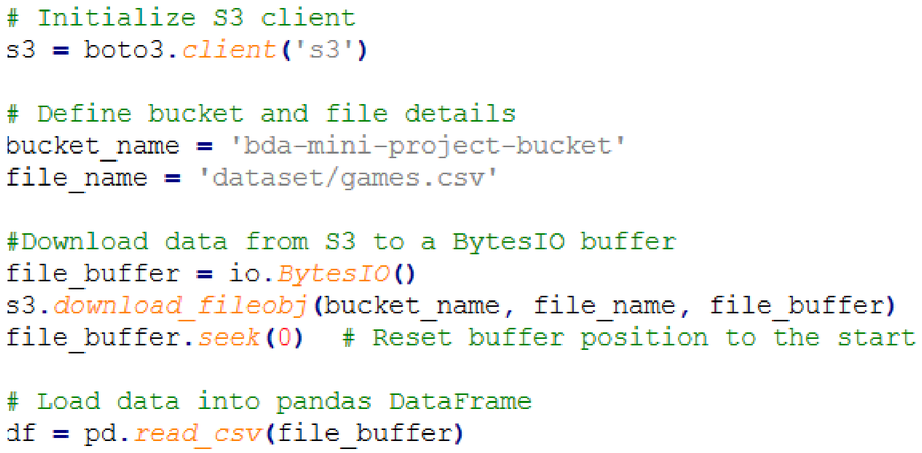
3.4 Automation :

Automation Script :

A python script for automating the flow : Data retrieval from S3 bucket, processing and storing it back S3 bucket.

This script will make use of ‘boto3’ library to connect with the s3 client.

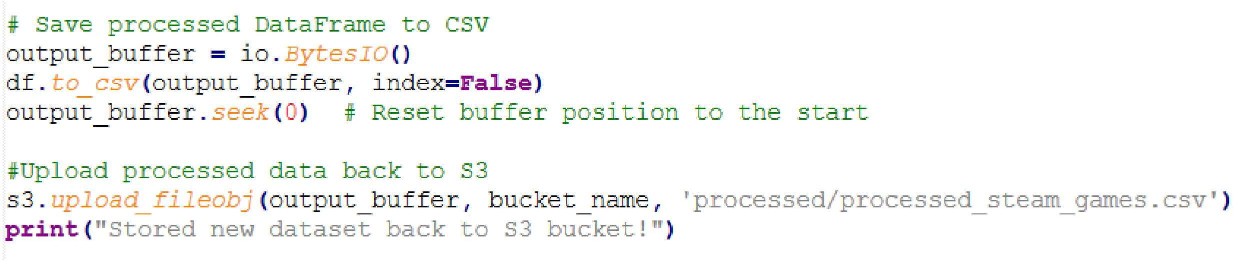
First stage is to get the data from s3 :



```
# Initialize S3 client
s3 = boto3.client('s3')

# Define bucket and file details
bucket_name = 'bda-mini-project-bucket'
file_name = 'dataset/games.csv'

#Download data from S3 to a BytesIO buffer
file_buffer = io.BytesIO()
s3.download_fileobj(bucket_name, file_name, file_buffer)
file_buffer.seek(0)  # Reset buffer position to the start

# Load data into pandas DataFrame
df = pd.read_csv(file_buffer)
```

Second stage is to apply all the transformations (Null handling, Outlier handling and Adding new columns). These are the same transformations applied before.

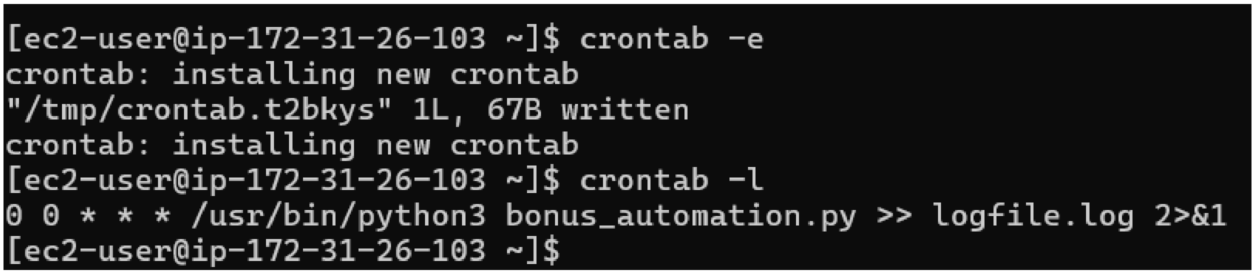
Third stage is to store this processed file back to S3. Unlike the previous method where the files are directly stored as part files and they have been used in the following steps, here the part files are combined into a single file. Before uploading the final file to S3, the files are first stored in local.



```
# Save processed DataFrame to CSV
output_buffer = io.BytesIO()
df.to_csv(output_buffer, index=False)
output_buffer.seek(0)  # Reset buffer position to the start

#Upload processed data back to S3
s3.upload_fileobj(output_buffer, bucket_name, 'processed/processed_steam_games.csv')
print("Stored new dataset back to S3 bucket!")
```

Scheduling Tools

Using cron to Schedule the process.

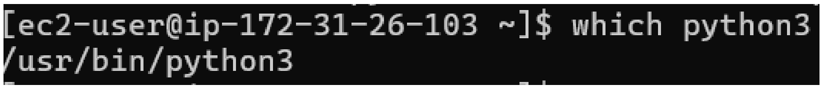


```
[ec2-user@ip-172-31-26-103 ~]$ crontab -e
crontab: installing new crontab
"/tmp/crontab.t2bkys" 1L, 67B written
crontab: installing new crontab
[ec2-user@ip-172-31-26-103 ~]$ crontab -l
0 0 * * * /usr/bin/python3 bonus_automation.py >> logfile.log 2>&1
[ec2-user@ip-172-31-26-103 ~]$
```

0 0 * * * — Cron Time Schedule

This specifies when the script should run. Each field represents a unit of time - Minute Hour Day Month DayofWeek

/usr/bin/python3 — Python Interpreter. This path can be found by running the following command:



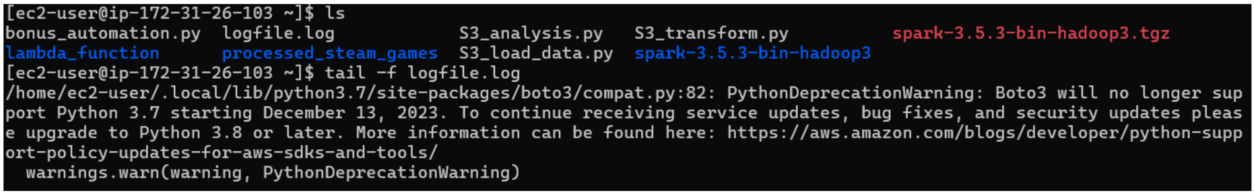
```
[ec2-user@ip-172-31-26-103 ~]$ which python3
/usr/bin/python3
```

bonus_automation.py — Script to execute.

>> logfile.log — Append Output to Log File.

2>&1 — Redirect Standard Error to Standard Output.

Once the command executes at the expected time give, we can see the logfile created and its contents:

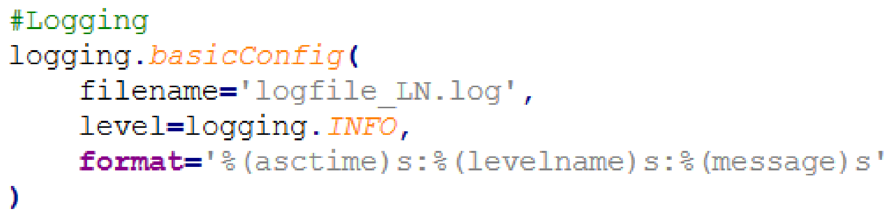


```
[ec2-user@ip-172-31-26-103 ~]$ ls
bonus_automation.py  logfile.log           S3_analysis.py   S3_transform.py         spark-3.5.3-bin-hadoop3.tgz
lambda_function      processed_steam_games  S3_load_data.py  spark-3.5.3-bin-hadoop3
[ec2-user@ip-172-31-26-103 ~]$ tail -f logfile.log
/home/ec2-user/.local/lib/python3.7/site-packages/boto3/compat.py:82: PythonDeprecationWarning: Boto3 will no longer sup
port Python 3.7 starting December 13, 2023. To continue receiving service updates, bug fixes, and security updates pleas
e upgrade to Python 3.8 or later. More information can be found here: https://aws.amazon.com/blogs/developer/python-supp
ort-policy-updates-for-aws-sdks-and-tools/
  warnings.warn(warning, PythonDeprecationWarning)
```

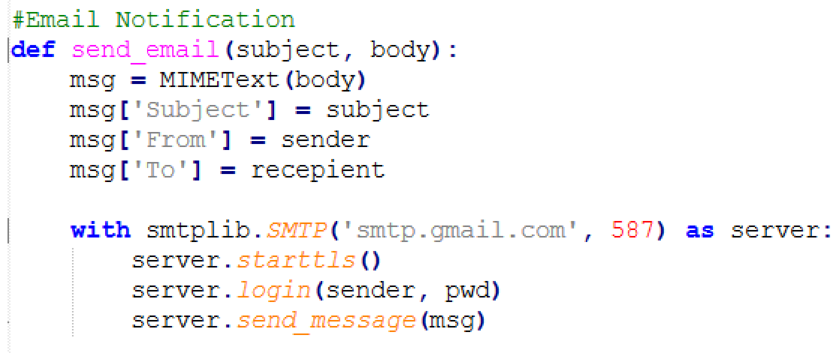
Logging and Notifications

To add logging and notifications, we need to modify the existing python script.

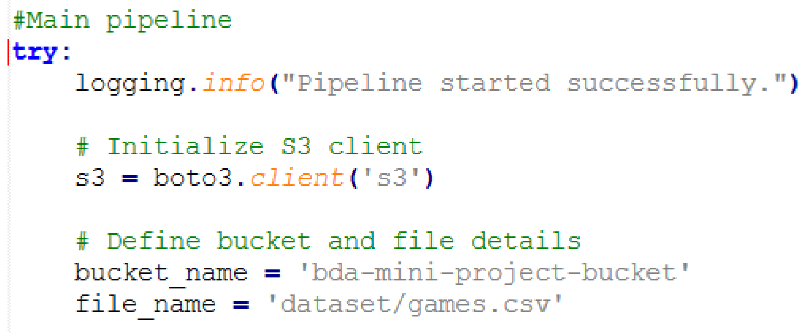
The entire script will now be a part of the try-catch block. Logging and Notification codes are added :



```
#Logging
logging.basicConfig(
    filename='logfile_LN.log',
    level=logging.INFO,
    format='%(asctime)s:%(levelname)s:%(message)s'
)
```

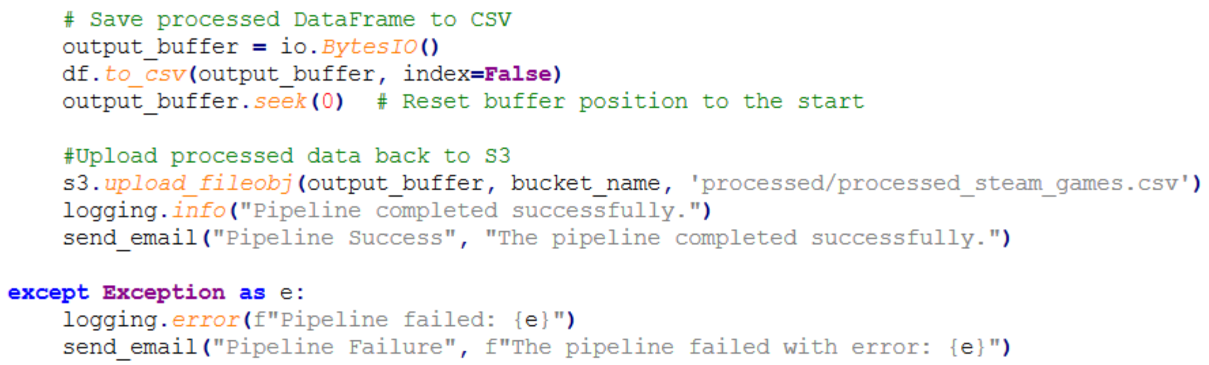


```
#Email Notification
def send_email(subject, body):
    msg = MIMEText(body)
    msg['Subject'] = subject
    msg['From'] = sender
    msg['To'] = recepient

    with smtplib.SMTP('smtp.gmail.com', 587) as server:
        server.starttls()
        server.login(sender, pwd)
        server.send_message(msg)
```



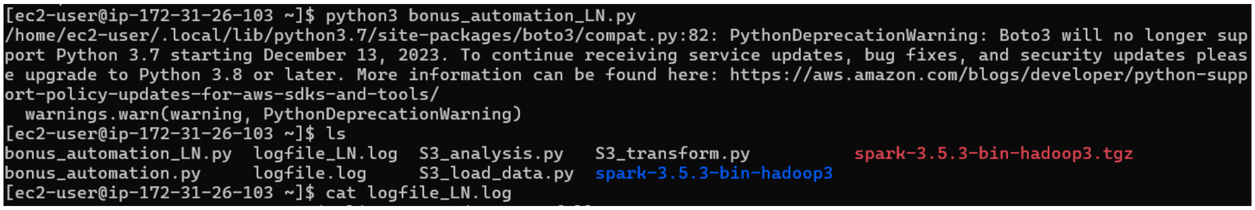
```
#Main pipeline
try:
    logging.info("Pipeline started successfully.")

    # Initialize S3 client
    s3 = boto3.client('s3')

    # Define bucket and file details
    bucket_name = 'bda-mini-project-bucket'
    file_name = 'dataset/games.csv'
```

...

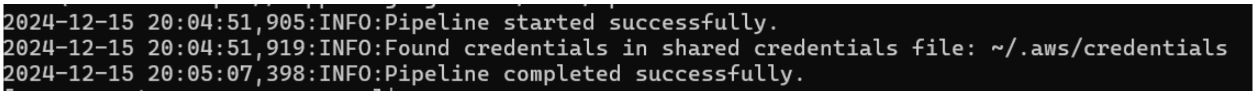


```
    # Save processed DataFrame to CSV
    output_buffer = io.BytesIO()
    df.to_csv(output_buffer, index=False)
    output_buffer.seek(0)  # Reset buffer position to the start

    #Upload processed data back to S3
    s3.upload_fileobj(output_buffer, bucket_name, 'processed/processed_steam_games.csv')
    logging.info("Pipeline completed successfully.")
    send_email("Pipeline Success", "The pipeline completed successfully.")

except Exception as e:
    logging.error(f"Pipeline failed: {e}")
    send_email("Pipeline Failure", f"The pipeline failed with error: {e}")
```

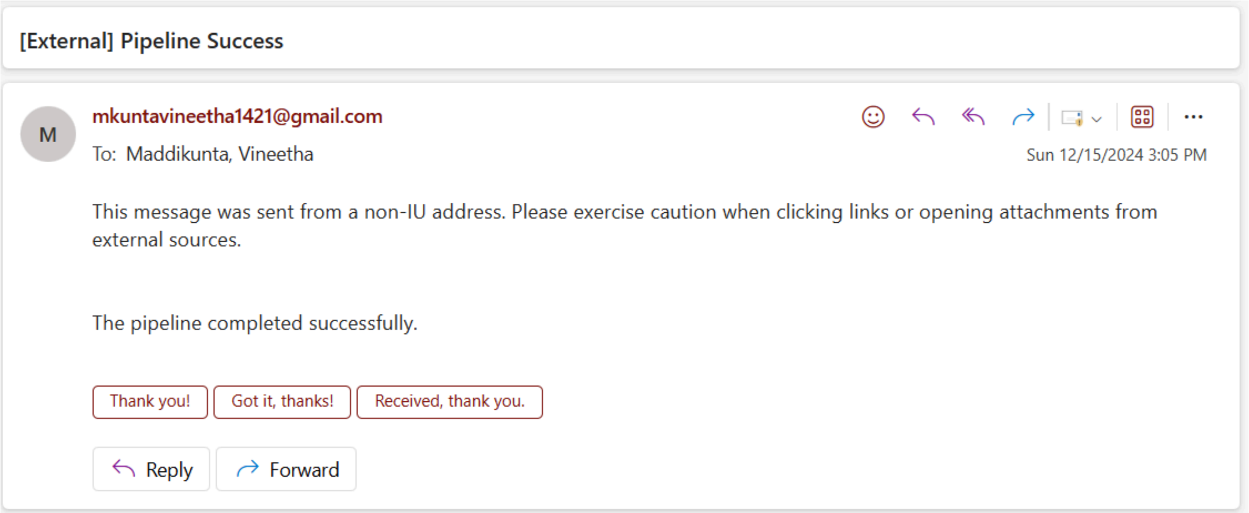
Once this script is executed as per the schedule, new log files will be created (for the first time) and the recipient will get the mail.



```
[ec2-user@ip-172-31-26-103 ~]$ python3 bonus_automation_LN.py
/home/ec2-user/.local/lib/python3.7/site-packages/boto3/compat.py:82: PythonDeprecationWarning: Boto3 will no longer sup
port Python 3.7 starting December 13, 2023. To continue receiving service updates, bug fixes, and security updates pleas
e upgrade to Python 3.8 or later. More information can be found here: https://aws.amazon.com/blogs/developer/python-supp
ort-policy-updates-for-aws-sdks-and-tools/
  warnings.warn(warning, PythonDeprecationWarning)
[ec2-user@ip-172-31-26-103 ~]$ ls
bonus_automation_LN.py  logfile_LN.log  S3_analysis.py   S3_transform.py          spark-3.5.3-bin-hadoop3.tgz
bonus_automation.py     logfile.log     S3_load_data.py  spark-3.5.3-bin-hadoop3
[ec2-user@ip-172-31-26-103 ~]$ cat logfile_LN.log
```



```
2024-12-15 20:04:51,905:INFO:Pipeline started successfully.
2024-12-15 20:04:51,919:INFO:Found credentials in shared credentials file: ~/.aws/credentials
2024-12-15 20:05:07,398:INFO:Pipeline completed successfully.
```



[External] Pipeline Success

mkuntavineetha1421@gmail.com

To: Maddikunta, Vineetha

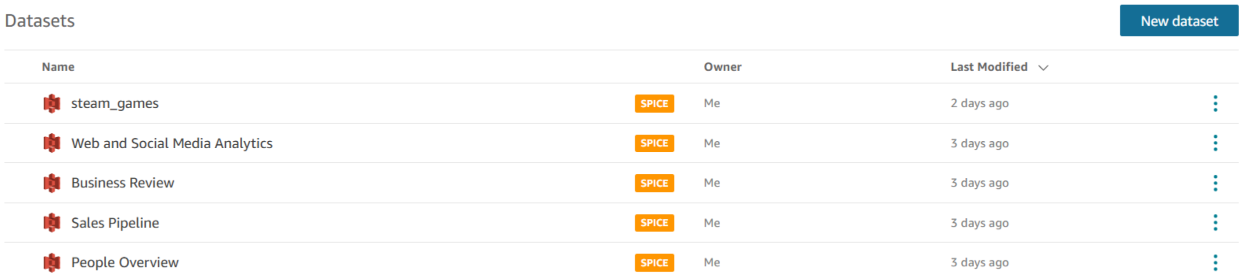
Sun 12/15/2024 3:05 PM

This message was sent from a non-IU address. Please exercise caution when clicking links or opening attachments from external sources.

The pipeline completed successfully.

Automate Dashboard updates

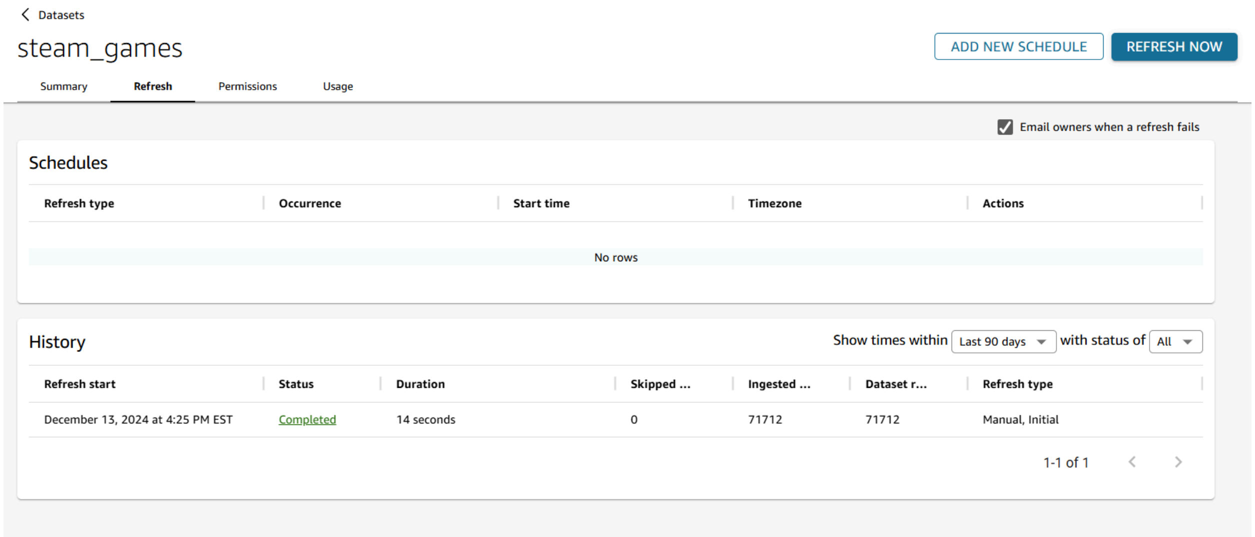
To automate the Visualization dashboards, open QuickSight, load the dataset from the new folder location in S3 bucket.



Datasets

Name		Owner	Last Modified
steam_games	SPICE	Me	2 days ago
Web and Social Media Analytics	SPICE	Me	3 days ago
Business Review	SPICE	Me	3 days ago
Sales Pipeline	SPICE	Me	3 days ago
People Overview	SPICE	Me	3 days ago

When we click on the dataset, we can check the ‘Refresh’ tab - ‘Add new schedule’.



steam_games

☑ Email owners when a refresh fails

Schedules

Refresh type	Occurrence	Start time	Timezone	Actions
No rows				

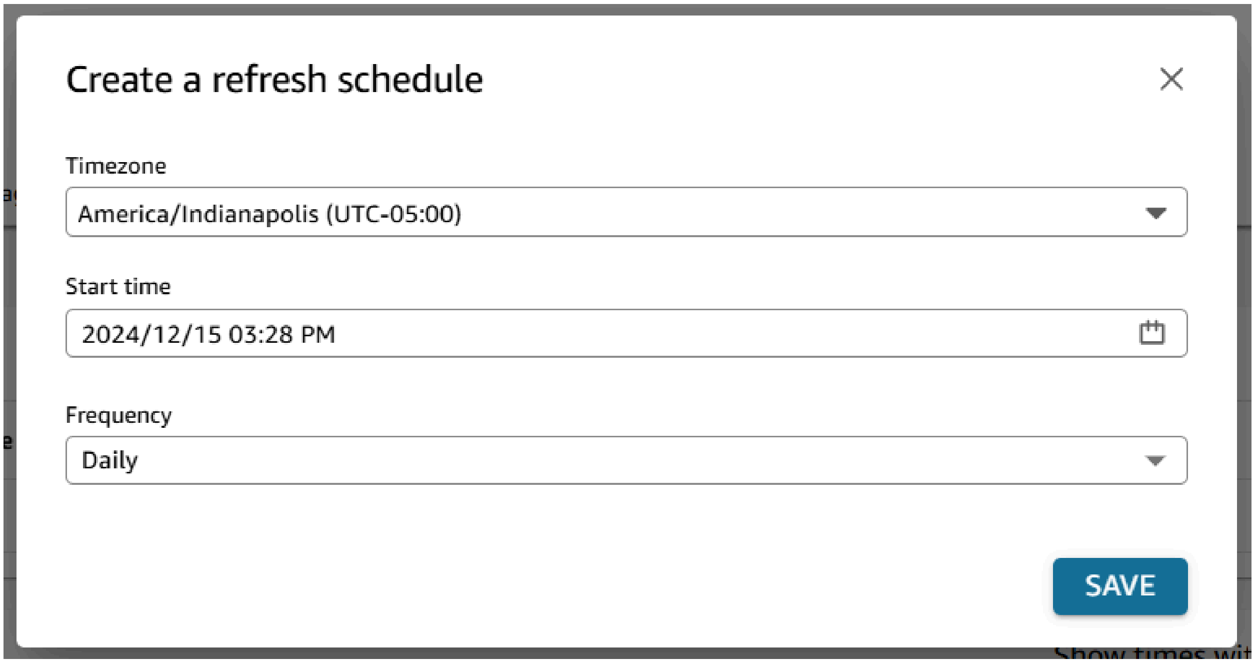
History

Show times within Last 90 days with status of All

Refresh start	Status	Duration	Skipped ...	Ingested ...	Dataset r...	Refresh type
December 13, 2024 at 4:25 PM EST	Completed	14 seconds	0	71712	71712	Manual, Initial

1-1 of 1

Since the dataset will be automatically updated everyday at midnight, the frequency is set to Daily:



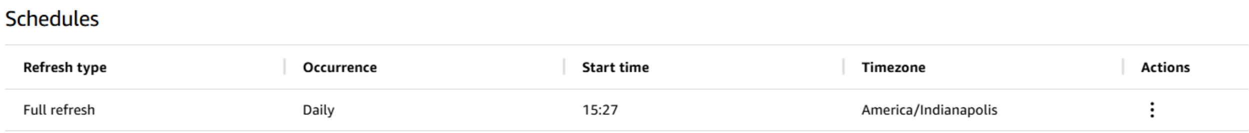
Create a refresh schedule

Timezone: America/Indianapolis (UTC-05:00)

Start time: 2024/12/15 03:28 PM

Frequency: Daily

A new schedule will be created and this will update the dataset and in turn the dashboard.



Schedules

Refresh type	Occurrence	Start time	Timezone	Actions
Full refresh	Daily	15:27	America/Indianapolis	⋮

4 Results

Key insights identified through this analysis:

- Market Analysis

Counter-Strike: Global Offensive leads user recommendations with 3,441,592 recommendations, followed by PUBG: BATTLEGROUNDS (1,616,422) and Grand Theft Auto V (1,247,051).

Single-player games dominate the platform with 64,924 titles, while Steam Achievements and Steam Cloud features appear in 31,990 and 16,287 games respectively.

The high engagement rates of free-to-play games, evidenced by higher positive ratings (1,761 average) compared to paid games (980 average), suggests a shifting market dynamic toward free-to-play models.

- Platform Distribution

Windows remains the dominant platform with 54,997 exclusive titles

Cross-platform availability (Windows, Mac, Linux) is limited to 7,815 games.

The platform shows a clear dominance of Windows-based games, suggesting a continued focus on PC gaming.

- Price Analysis

The average game price is $7.21

Price categories show distinct patterns:

Low (<$10): Most common

Medium ($10-$50): Moderate representation

High (>$50): Limited number of titles

The pricing structure indicates a market that favors affordable games, with most titles falling in the lower price range.

- Machine Learning Performance

The price prediction model achieved:

R² score: 0.788 (78.8% accuracy)

Mean Absolute Error: 2.589

Root Mean Squared Error: Slightly higher than MAE

Game prices can be effectively predicted using the available features, though there's room for improvement.

5 Conclusion

In conclusion, the implementation of the Steam Games Dataset Analysis project on AWS cloud services has showcased the power and efficiency of big data tools and technologies in the realm of gaming analytics. By leveraging a comprehensive data pipeline that includes Amazon S3 for storage, EC2 for computational needs, and PySpark for distributed processing, we effectively managed and analyzed a vast dataset.

This project tackled data quality challenges through automated preprocessing, managing null values and outliers, and introduced new analytical dimensions by adding columns like game age, price category, and popularity tier. The use of Spark SQL for querying provided deep insights into game performance, revenue trends, and platform distribution. Moreover, the integration of machine learning through Amazon SageMaker Autopilot not only simplified model development but also achieved a commendable 78.8% accuracy in price prediction.

The visualization of these insights was made accessible and interactive with Amazon QuickSight, enhancing decision-making capabilities. Automation of the entire data processing pipeline using cron jobs, alongside robust logging and notification systems, underscores the project's scalability and efficiency, proving the value of cloud infrastructure in transforming large-scale gaming data into actionable business insights.

References

1. Steam Games Dataset. (2023). Kaggle Datasets. https://www.kaggle.com/datasets/fronkongames/steam-games-dataset
2. Rose, M. (2024, July). What Games Are Selling on Steam: The Q2 Report. How To Market A Game Blog.
3. Hu, W., Wang, Y., & Xia, R. (2024). Machine Learning-Based Steam Platform Game's Popularity Analysis. International Conference on Data Science.
4. AWS Documentation. (2024). Amazon Web Services Cloud Computing Services.
5. Video tutorials :
   AWS Basics for Beginners - freeCodeCamp.org (2023).
   Getting Started with Amazon SageMaker, AWS Online Tech Talks (2023).
   Data Analysis with PySpark, Tech With Tim (2023).
   Automating File Migration to S3 with AWS Sync and Cron Jobs - Sai's Artifacts(2024).