Video Game Sales on Steam Prediction

Team ID: CS_8

ID	Name
2022170385	محمد متولي عبدالحميد عوض محمد
2022170375	محمد عادل علي حسن
2022170389	محمد منير تاج الدين منصور
2022170373	محمد طارق الحسين محمد منصور العراقي
2022170456	مينا باسم نادي

Merging DataFrames

1. Converted the common column (appid) in all dataframes to a single datatype (Int64).

2. Merged info_base_games (99k rows) & gamalytic_steam_games (93k rows, target) using inner merge.

3. Merged demos (15k rows only) & dlcs (5k rows only) to the resultant df using left merge.

Final dataframe shape (69428, 23):

	Name	Data_Type	Top_10_Unique_Values	Nunique_Values	Nulls	Percent_of_Nulls	Duplicates
0	appid	Int64	[3297920, 3175890, 2317310, 2316930, 3332170, ...	67909	0	0.000000	0
1	name_x	object	[Echoes, Delirium, Alone, Zombie Hunter, Lost,...	67531	0	0.000000	0
2	metacritic	object	[80, 76, 73, 68, 75, 81, 79.0, 81.0, 80.0, 77]	169	66495	95.775480	0
3	steam_achievements	bool	[True, False]	2	0	0.000000	0
4	steam_trading_cards	bool	[False, True]	2	0	0.000000	0
5	workshop_support	bool	[False, True]	2	0	0.000000	0
6	genres	object	[Casual, Indie, Action, Indie, Action, Adventu...	2016	104	0.149795	0
7	achievements_total	object	[10, 12, 6, 20, 15, 10.0, 8, 5, 11, 16]	691	32133	46.282480	0
8	release_date	object	[Coming soon, Q1 2025, Oct 31, 2024, Dec 5, 20...	4541	2	0.002881	0
9	supported_platforms	object	[['windows'], ['windows', 'mac', 'linux'], ['w...	7	0	0.000000	0
10	steamId	int64	[3297920, 3175890, 2317310, 2316930, 3332170, ...	67909	0	0.000000	0
11	price	float64	[0.0, 4.99, 0.99, 9.99, 1.99, 2.99, 3.99, 14.9...	295	0	0.000000	0
12	copiesSold	int64	[1, 15, 30, 36, 45, 18, 75, 72, 60, 90]	17774	0	0.000000	0
13	publisherClass	object	[Hobbyist, Indie, AA, AAA]	4	0	0.000000	0
14	reviewScore	int64	[100, 0, 50, 67, 75, 80, 83, 88, 86, 89]	99	0	0.000000	0
15	aiContent	float64	[]	0	69428	100.000000	0
16	Unnamed: 0	float64	[3153.0, 1580.0, 2285.0, 1096.0, 3048.0, 14638...	7352	61893	89.147030	0
17	full_game_appid	object	[3172700, 2317010, 3170800, 3172880, 3171450, ...	7352	61893	89.147030	0
18	demo_appid	object	[3173190, 2318980, 3288690, 3210760, 3216360, ...	7352	61893	89.147030	0
19	name_y	object	[Bonds Demo, InfectionWarfare Demo, Encounter ...	7348	61894	89.148470	0
20	base_appid	object	[3321700, 2319940, 2315830, 3175700, 3182330, ...	3806	65597	94.482053	0
21	dlc_appid	object	[3324850, 3324760, 2315910, 3175810, 3333910, ...	3806	65597	94.482053	0
22	name	object	[Christmas Fables: The Wishing Store DLC, 버튜버 ...	3805	65597	94.482053	0

Preprocessing Features

- **Dropped Features:**
  - With high NULL%.
  - Unrelated to the target (game IDs).

	appid	steamId	copiesSold
appid	1.000000	1.000000	-0.053396
steamId	1.000000	1.000000	-0.053396
copiesSold	-0.053396	-0.053396	1.000000

- Filled features with low NULL% using forward/backward fill.

- **Dropped:**
  - Duplicate rows.
  - Rows with NULLs (only 2 rows with NULL release_date).

- Converted **release_date** to age by years (>=2026 to 0, 2025 to 1, 2024 to 2, etc.).

- Tried handling outliers but the **models' accuracies were worse**.

Final dataframe shape (69426, 11):

	Name	Data_Type	Top_10_Unique_Values	Nunique_Values	Nulls	Percent_of_Nulls	Duplicates
0	name_x	object	[Delirium, Echoes, Arena, Dodge, Zombie Hunter...	67529	0	0.0	2
1	steam_achievements	bool	[True, False]	2	0	0.0	2
2	steam_trading_cards	bool	[False, True]	2	0	0.0	2
3	workshop_support	bool	[False, True]	2	0	0.0	2
4	genres	object	[Casual, Indie, Action, Indie, Action, Adventu...	2016	0	0.0	2
5	supported_platforms	object	[['windows'], ['windows', 'mac', 'linux'], ['w...	7	0	0.0	2
6	price	float64	[0.0, 4.99, 0.99, 9.99, 1.99, 2.99, 3.99, 14.9...	295	0	0.0	2
7	copiesSold	int64	[1, 15, 30, 36, 45, 18, 75, 72, 60, 90]	17772	0	0.0	2
8	publisherClass	object	[Hobbyist, Indie, AA, AAA]	4	0	0.0	2
9	reviewScore	int64	[100, 0, 50, 67, 75, 80, 83, 88, 86, 89]	99	0	0.0	2
10	age_years	int32	[2, 5, 3, 8, 4, 7, 6, 9, 1, 12]	30	0	0.0	2

Feature Engineering

1. GameRating

- Combined multiple features that (are assumed to) correlate with a game's sales.
- **Components:**
  - **extras_mean**
    - The mean of game-related extras: **Achievements**, **Trading Cards**, and **Workshop Support**.
    - +1 to avoid multiplication by zero.
    - **Intuition**: More extras → more engagement (typically) → more sales (direct relationship).
  - **reviewScore**
    - +1 to avoid zero values.
    - **Intuition**: Better reviews → attract more players → more sales (direct relationship).
  - **publisher_encode**
    - Ordinal encoding of publisher type (AAA >>> AA >> Indie > Hobbyist).
    - **Intuition**: Well-known publishers → greater marketing power → higher sales (direct relationship).
  - **age_years**
    - Release date converted into the game's age in years (2026 and beyond = 0, 2025 = 1, 2024 = 2, etc.).
    - +1 to avoid multiplication by zero.
    - **Intuition**: Older games → more time to accumulate sales (inverse relationship).

2. GameRatingWithGenres

- Included the **genres** column in the **GameRating** feature.
- Slightly worse correlation from **0.209** → **0.202**.
- **Steps:**
  1. Get total **copiesSold** for each unique genre across the dataframe.
  2. Replace every row in **genres** with the **mean** of **copiesSold** of its genres.
  3. Divide by **10 million** to make the values smaller.
  4. Multiply **GameRating** by the new **genres** value to create the new feature.

3. RatingOverPrice

- Divided **GameRatingWithGenres** feature by **price** feature.
- +1 to avoid division by zero.
- Improves correlation from **0.202** → **0.389**.
- **Intuition**: Lower price (generally) means more sales (inverse relation).

4. GameRatingWithPlatforms

- Included the **supported_platforms** column in the **RatingOverPrice** feature.
- Improves correlation from **0.389** → **0.584**.
- **Steps:**
  1. Set each platform to a specific value (trial and errored the choices).
  2. Replace each value in **supported_platforms** with the sum of its platforms.
  3. Multiply **RatingOverPrice** by the new column.

5. NameAsCopiesSold

- Encoded **name_x** column like **genres**.
- **Steps:**
  1. Preprocessed the names using NLP techniques (tokenization, stopwords removal, lemmatization).
  2. Get total **copiesSold** for each unique token.
  3. Replace every row with mean of its tokens.
  4. Divide by **10,000** to make the values smaller.

6. GameRatingWithNames

- Multiplied **GameRatingWithPlatforms** feature by the **NameAsCopiesSold** feature.
- Improves correlation from **0.584** → **0.799**.

Correlation between engineered features & target:

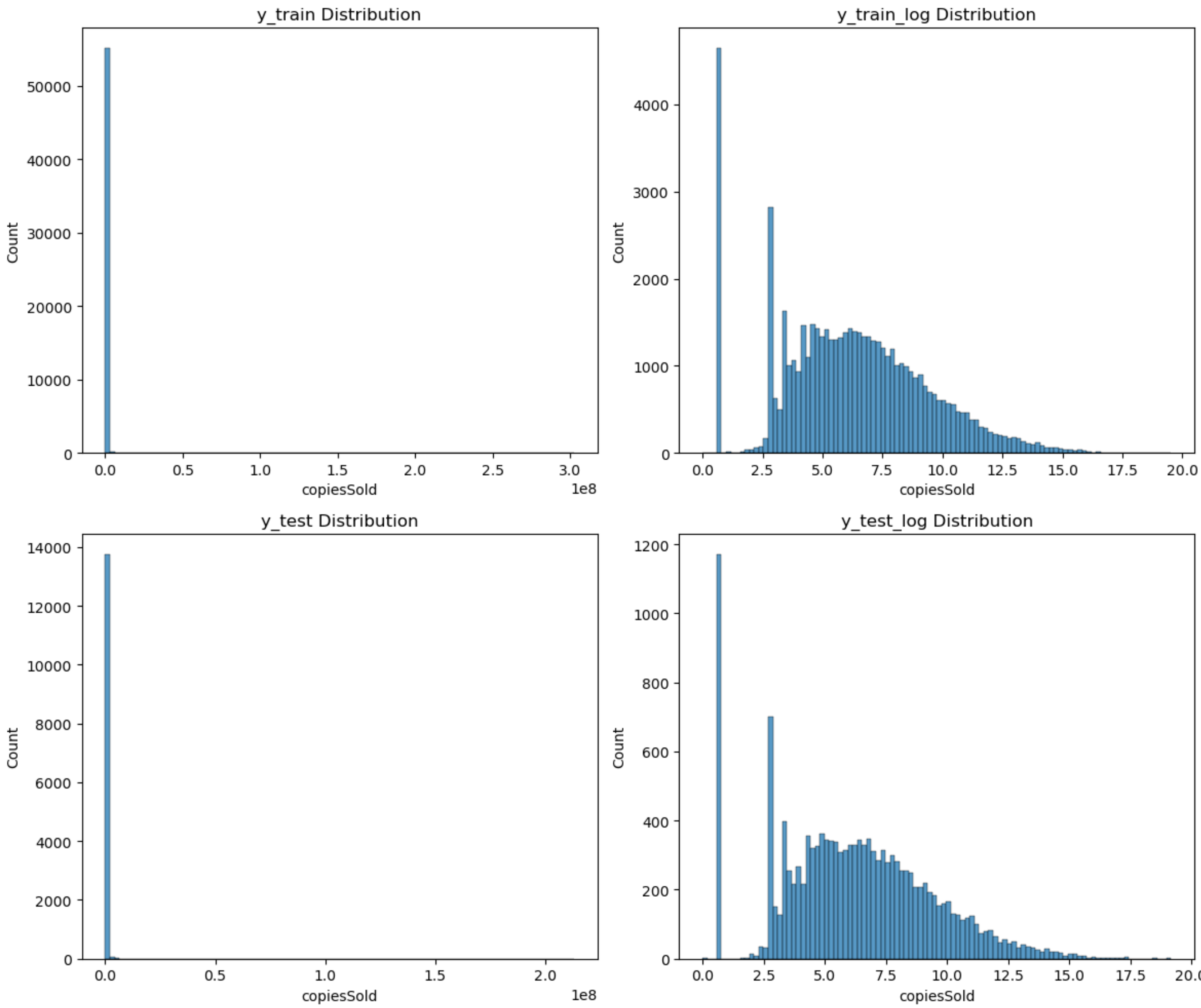
	1-GameRating	2-GameRatingWithGenres	3-RatingOverPrice	4-GameRatingWithPlatforms	5-NameAsCopiesSold	6-GameRatingWithNames	copiesSold
1-GameRating	1.000000	0.909428	0.476529	0.391259	0.131325	0.138996	0.209514
2-GameRatingWithGenres	0.909428	1.000000	0.501508	0.422715	0.120347	0.154042	0.202118
3-RatingOverPrice	0.476529	0.501508	1.000000	0.799745	0.093774	0.402159	0.389837
4-GameRatingWithPlatforms	0.391259	0.422715	0.799745	1.000000	0.116051	0.702144	0.584625
5-NameAsCopiesSold	0.131325	0.120347	0.093774	0.116051	1.000000	0.145555	0.159047
6-GameRatingWithNames	0.138996	0.154042	0.402159	0.702144	0.145555	1.000000	0.799956
copiesSold	0.209514	0.202118	0.389837	0.584625	0.159047	0.799956	1.000000

Before Scaling/Encoding

Dropped duplicate rows (134 rows).

Split data into **80% training** & **20% testing** sets with **random_state=42**.

Made another **Y** to train/evaluate models with (**original Y logp1 transformed** because original Y is **skewed**):



Scaling

- Scaled **reviewScore** using **MinMaxScaler** (because it's within a **fixed range** → [0,100]).
- Scaled the remaining **continuous features** using **RobustScaler** (**less sensitive** than **StandardScaler** to **outliers**).

Encoding

- Encoded **genres** & **supported_platforms** using **MultiLabelBinarizer**.
- Encoded the remaining **discrete features** using **OneHotEncoder**.

Final x_train/x_test shape:

	price	reviewScore	age_years	1-GameRating	2-GameRatingWithGenres	3-RatingOverPrice	4-GameRatingWithPlatforms	5-NameAsCopiesSold	6-GameRatingWithNames	Accounting	Action	Adventure	Animation & Modeling	Audio Production	Casual	Design & Illustration
54352	0.000000	0.00	6	-0.029322	-0.034247	-0.135608	-0.137734	-0.226431	-0.068923	0	0	0	0	0	1	0
152	-0.444444	0.96	4	1.250330	1.537357	6.820913	21.240703	-0.256058	0.277350	0	0	1	0	0	0	0
45187	0.555556	0.57	7	1.194906	1.606990	1.179427	0.872284	-0.186457	0.177979	0	1	1	0	0	0	0
38784	-0.444444	1.00	4	-0.003035	0.006822	0.047160	0.002642	-0.260993	-0.068965	0	1	1	0	0	0	0
57856	2.777778	0.80	5	1.252969	1.826629	0.392841	0.268143	0.879986	1.450858	0	1	0	0	0	0	0

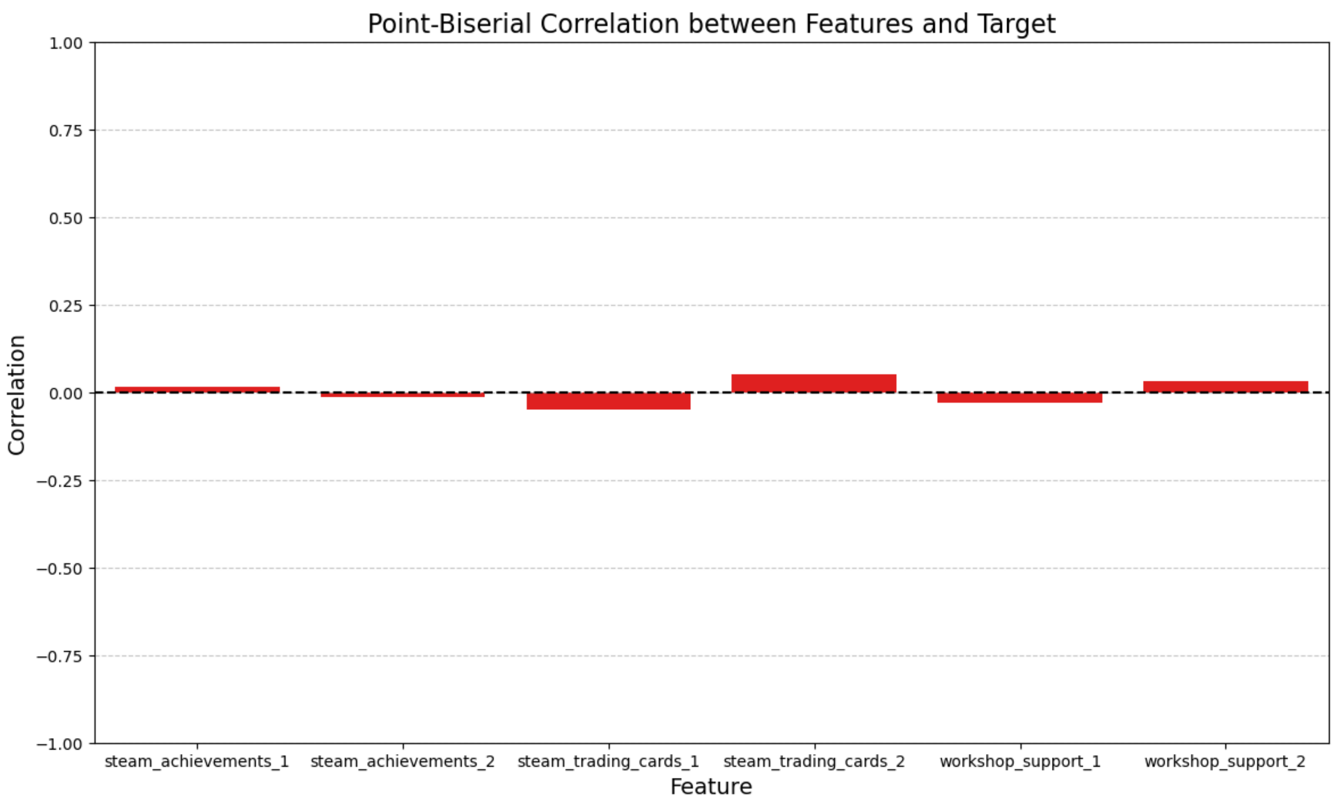
Early Access	Education	Free To Play	Game Development	Gore	Indie	Massively Multiplayer	Nudity	RPG	Racing	Sexual Content	Simulation	Software Training	Sports	Strategy	Utilities	Video Production	Violent	Web Publishing	linux	mac	windows	steam_achievements_1	steam_achievements_2
0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0
0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	1	0
0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0
0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0

steam_trading_cards_1	steam_trading_cards_2	workshop_support_1	workshop_support_2	publisherClass_1	publisherClass_2	publisherClass_3	publisherClass_4
1	0	1	0	1	0	0	0
0	1	0	1	0	1	0	0
1	0	1	0	0	1	0	0
1	0	1	0	1	0	0	0
1	0	1	0	0	1	0	0

Feature Selection

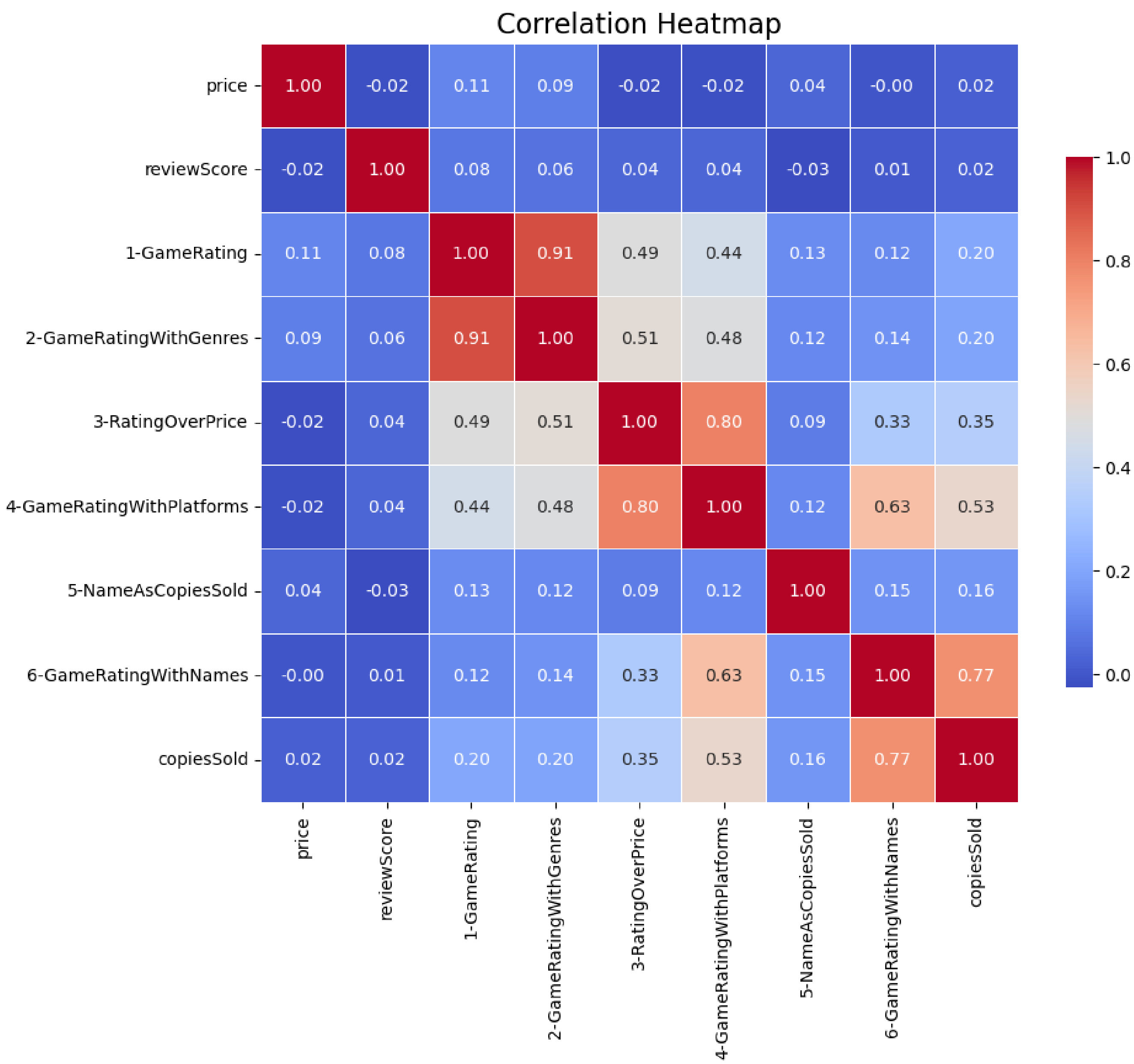
Binary Features:

- using **Point-Biserial Correlation**.
- low correlation features made no difference in model evaluation.
- **removed no features**.



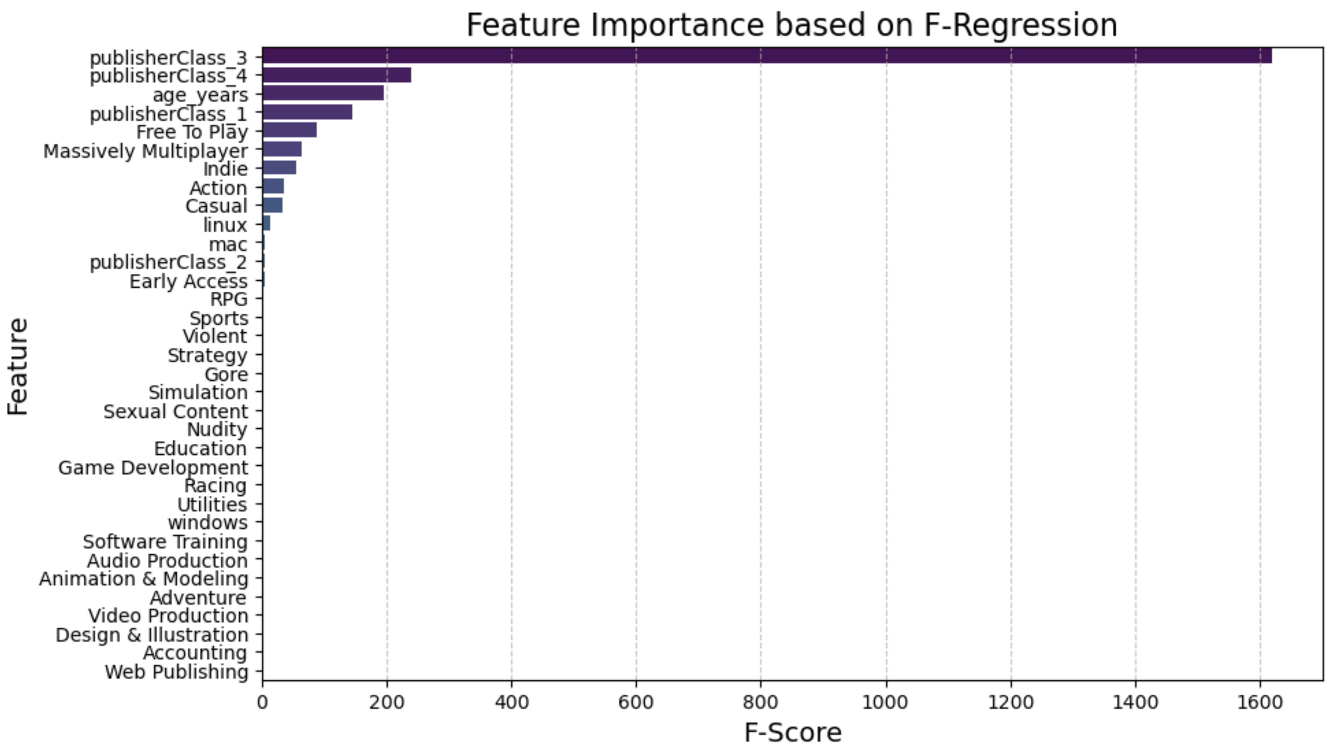
Continuous Features:

- using **normal correlation**.
- features with low correlation affected evaluation negatively.
- **removed** features with **correlation < 0.08**.



Categorical Features:

- using **ANOVA**.
- features with low F-Score affected evaluation negatively.
- **removed** features with **F-Score < 10**.



Final x_train/x_test features

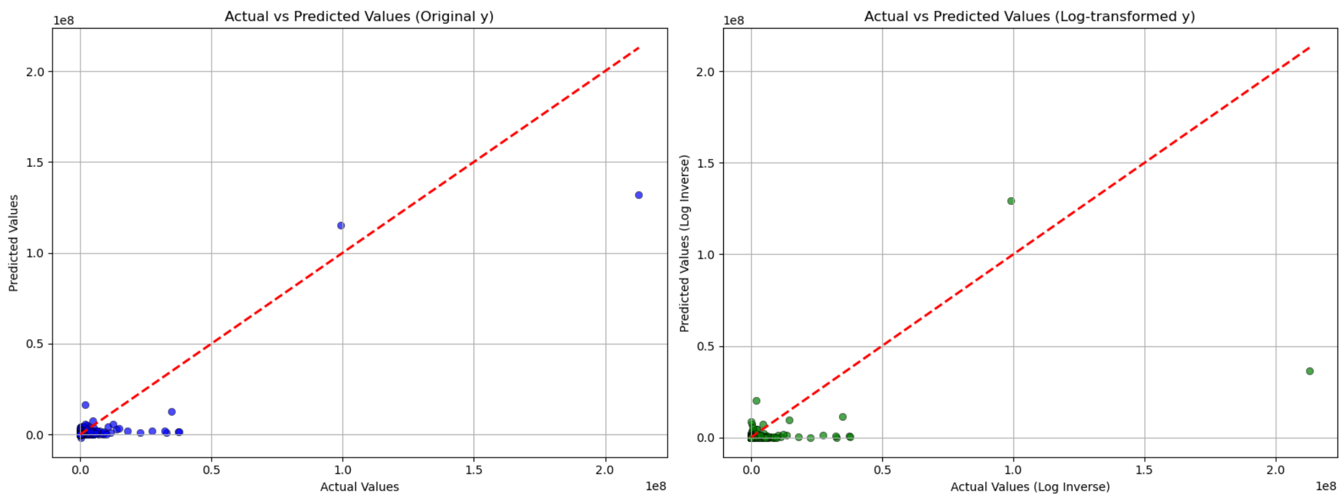
age_years	1-GameRating	2-GameRatingWithGenres	3-RatingOverPrice	4-GameRatingWithPlatforms	5-NameAsCopiesSold	6-GameRatingWithNames	Action	Casual	Free To Play	Indie	Massively Multiplayer	linux	steam_achievements_1	steam_achievements_2

steam_trading_cards_1	steam_trading_cards_2	workshop_support_1	workshop_support_2	publisherClass_1	publisherClass_3	publisherClass_4

Model Training

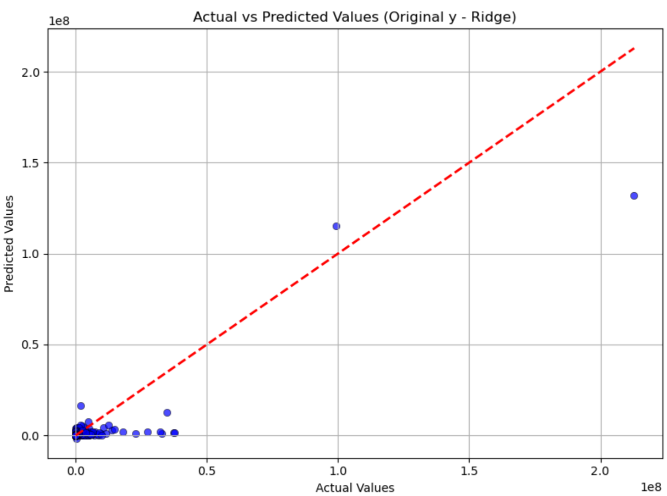
Linear Regression:

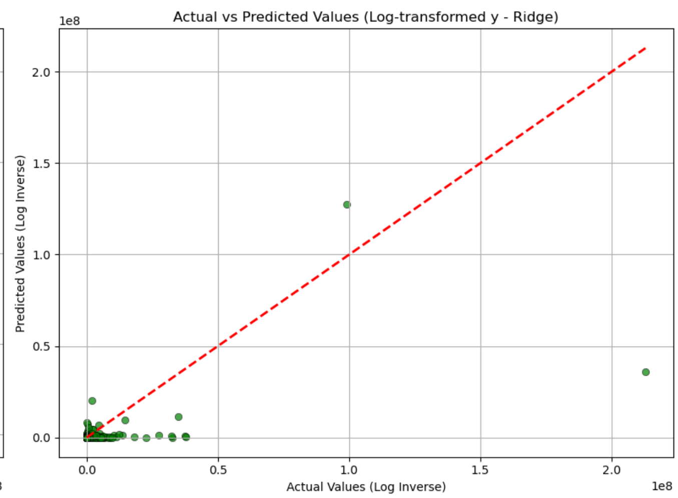
```
Original y:
Mean Squared Error: 1144033631944.808
Root Mean Squared Error (RMSE): 1069595.08
Mean Absolute Error: 159391.99383292862
R^2 Score: 0.7578110377933094

Log1p-transformed y:
Mean Squared Error: 3012322236277.1294
Root Mean Squared Error (RMSE): 1735604.29
Mean Absolute Error: 87333.58971093452
R^2 Score: 0.36229917035228265
```



Ridge Regression:

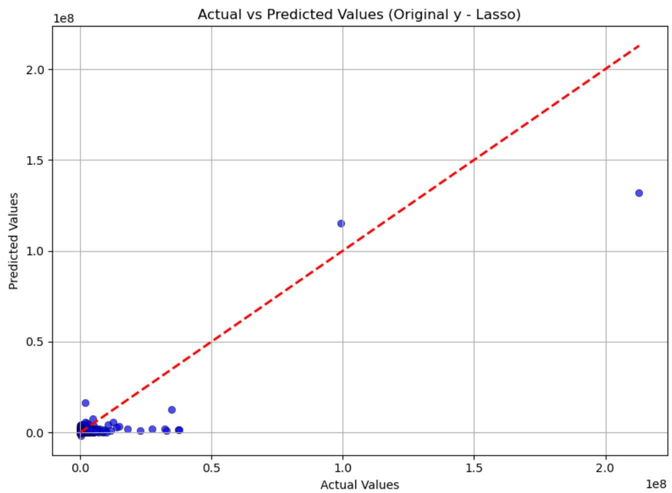
```
Original y:
Mean Squared Error: 1144012526017.4348
Root Mean Squared Error (RMSE): 1069585.21
Mean Absolute Error: 159370.352039883
R^2 Score: 0.7578155058635692

Log1p-transformed y:
Mean Squared Error: 3013204283373.6704
Root Mean Squared Error (RMSE): 1735858.37
Mean Absolute Error: 87196.37602225087
R^2 Score: 0.36211244326230585
```

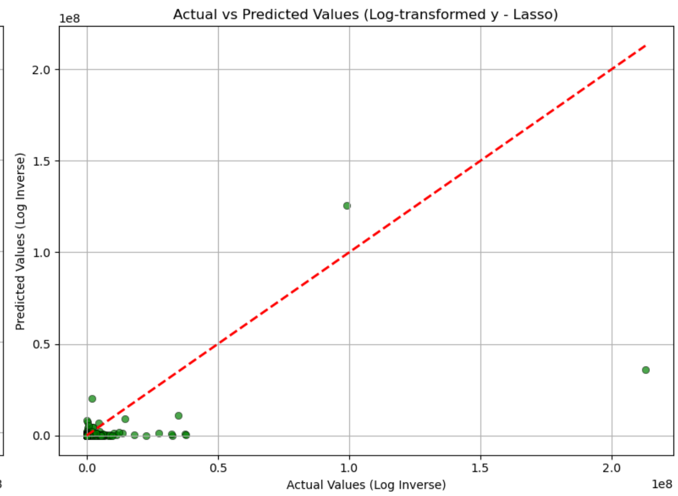




Lasso Regression:

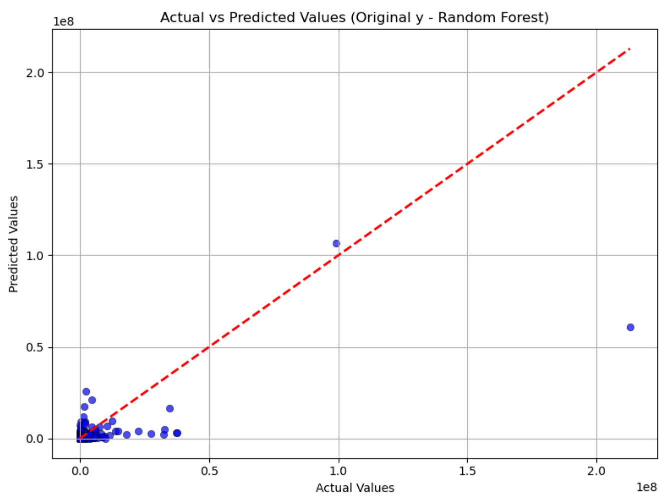
```
Original y:
Mean Squared Error: 1144033631850.5657
Root Mean Squared Error (RMSE): 1069595.08
Mean Absolute Error: 159391.99330945662
R^2 Score: 0.7578110378132603

Log1p-transformed y:
Mean Squared Error: 3010483254821.072
Root Mean Squared Error (RMSE): 1735074.42
Mean Absolute Error: 87064.49282684566
R^2 Score: 0.36268847797220183
```

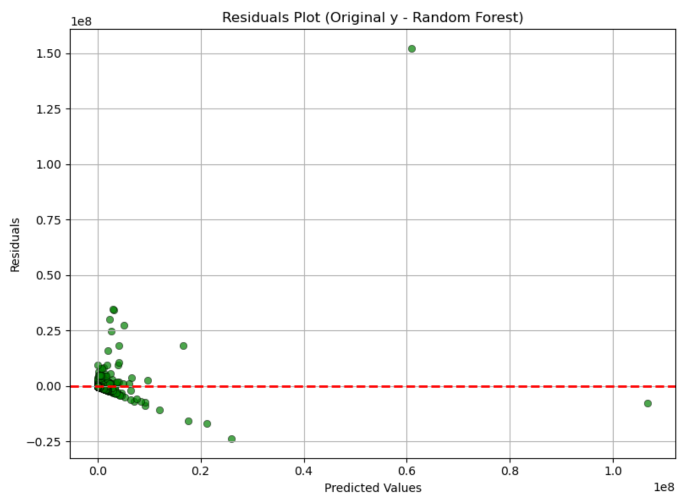


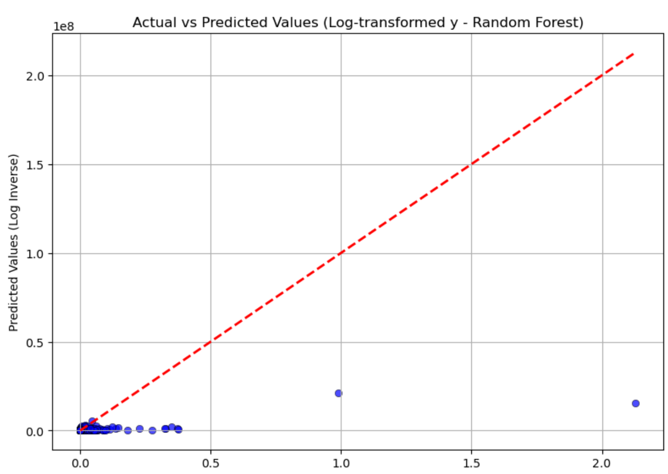


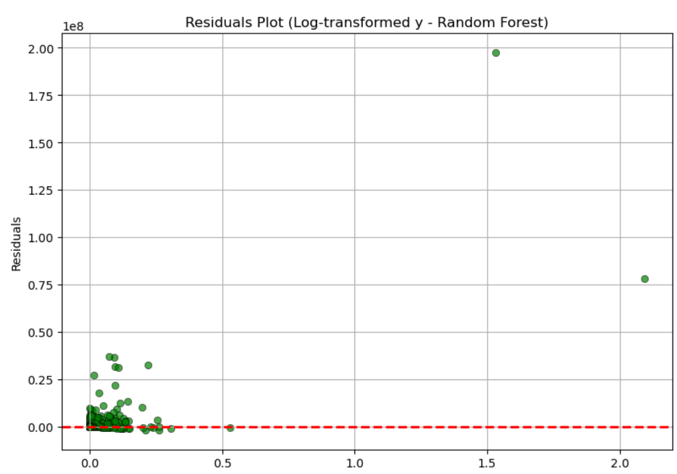
RandomForest (w/ GridSearch):

```
Original y:
Mean Squared Error: 2317632060855.369
Root Mean Squared Error (RMSE): 1522377.11
Mean Absolute Error: 105695.0123425091
R^2 Score: 0.5093632845029905

Log1p-transformed y:
Mean Squared Error: 3941507055143.627
Root Mean Squared Error (RMSE): 1985322.91
Mean Absolute Error: 85973.86614892342
R^2 Score: 0.16559314642453027
```









SVM (with PCA):

Original y:

Mean Squared Error: 4732765618111.92

Root Mean Squared Error (RMSE): 2175492.04

Mean Absolute Error: 96101.23775416201

R^2 Score: -0.0019142457110299382

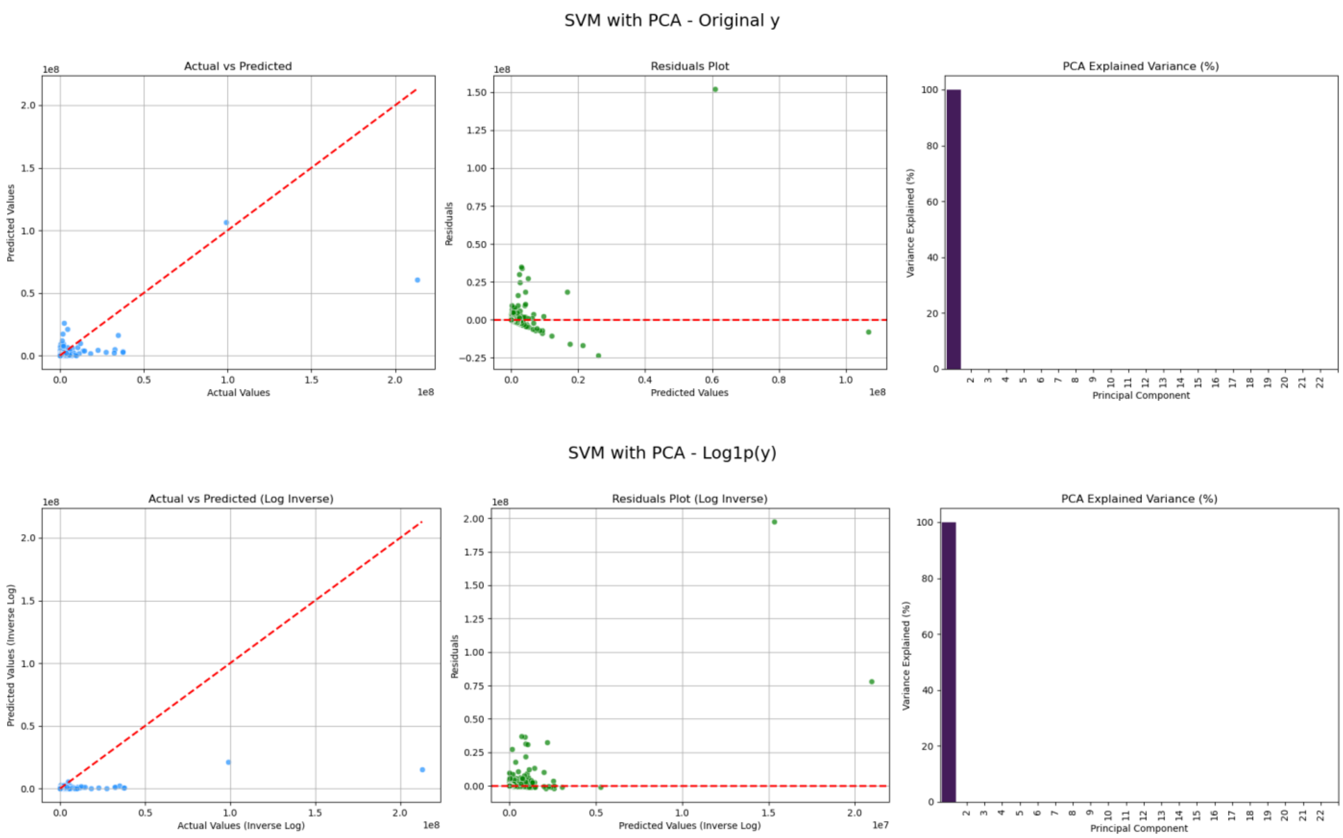
Log1p-transformed y:

Mean Squared Error: 1.4015269905505807e+39

Root Mean Squared Error (RMSE): 37436973576273240064.00

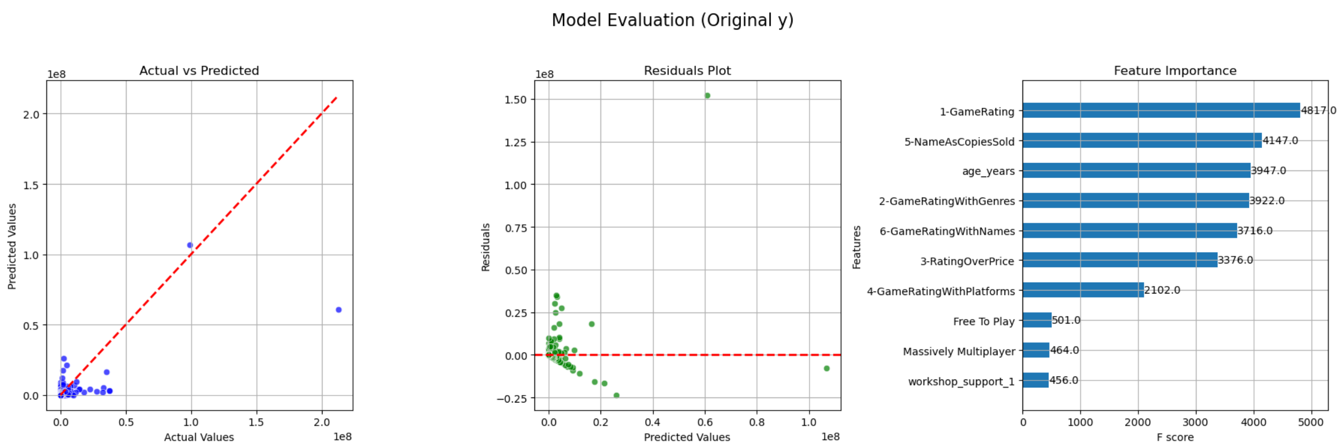
Mean Absolute Error: 3.180056189282022e+17

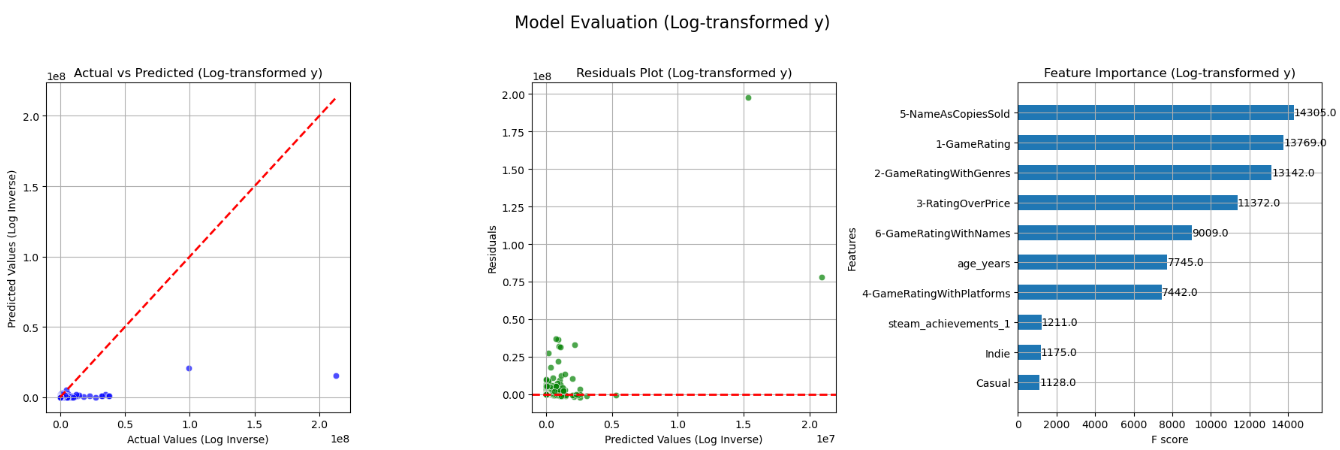
R^2 Score: -2.9669964052463e+26



XGBoost (w/ GridSearch):

```
Original y:
Mean Squared Error: 4143141955252.739
Root Mean Squared Error (RMSE): 2035470.94
Mean Absolute Error: 112115.04667977823
R^2 Score: 0.12290755935917452

Log1p-transformed y:
Mean Squared Error: 1680790372626.9573
Root Mean Squared Error (RMSE): 1296453.00
Mean Absolute Error: 76201.79668910433
R^2 Score: 0.6441810234708573
```

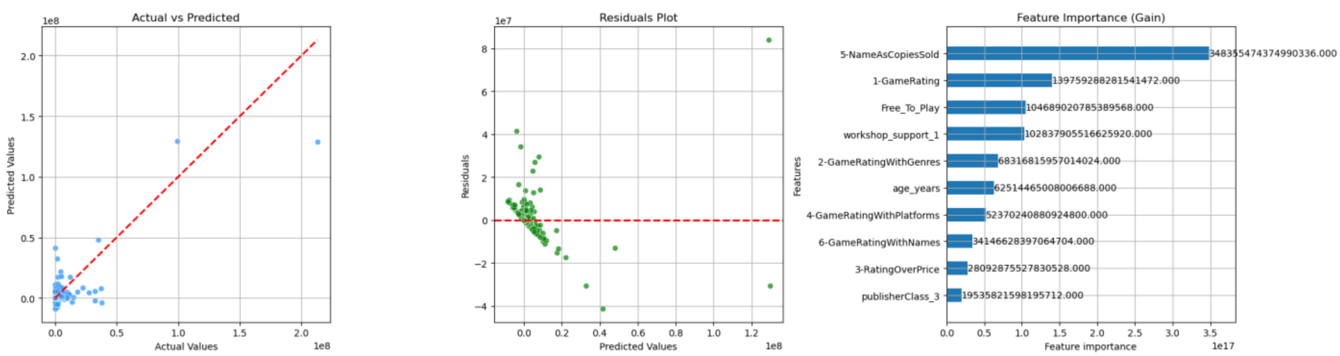




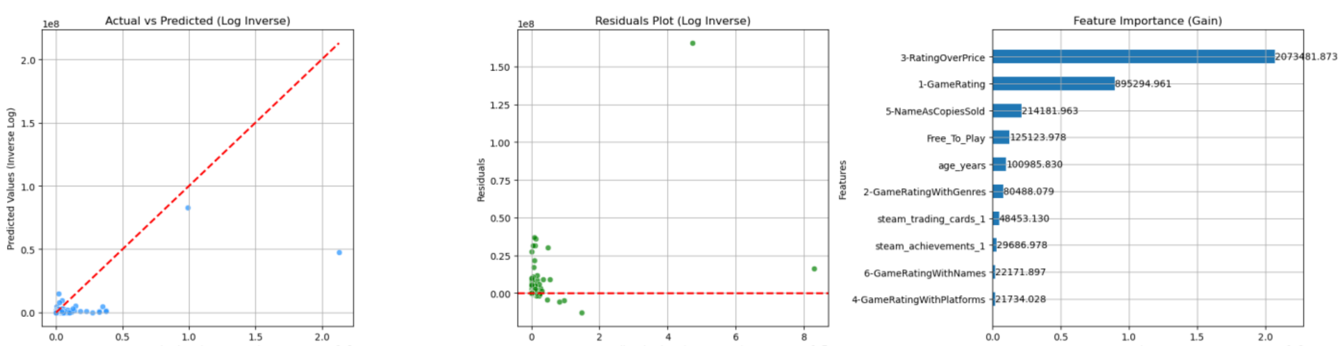
<u>LGB (w/ GridSearch):</u>

```
Original y:
Mean Squared Error: 1505539063262.4695
Root Mean Squared Error (RMSE): 1227004.10
Mean Absolute Error: 125751.11448444106
R^2 Score: 0.6812812725852091

Log1p-transformed y:
Mean Squared Error: 2670382417747.758
Root Mean Squared Error (RMSE): 1634130.48
Mean Absolute Error: 79718.47688664119
R^2 Score: 0.4346869459161813
```

Model 1L: LGBM on Original y

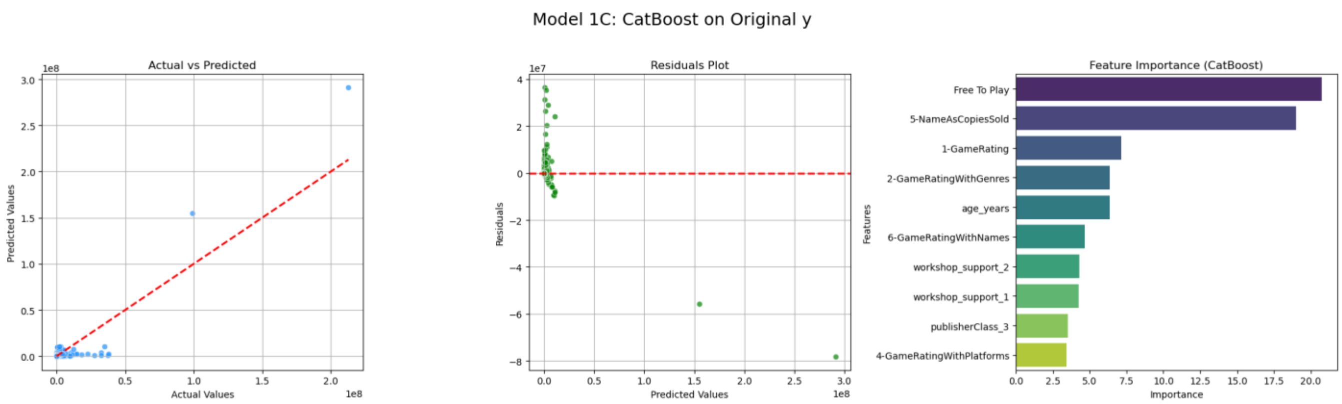


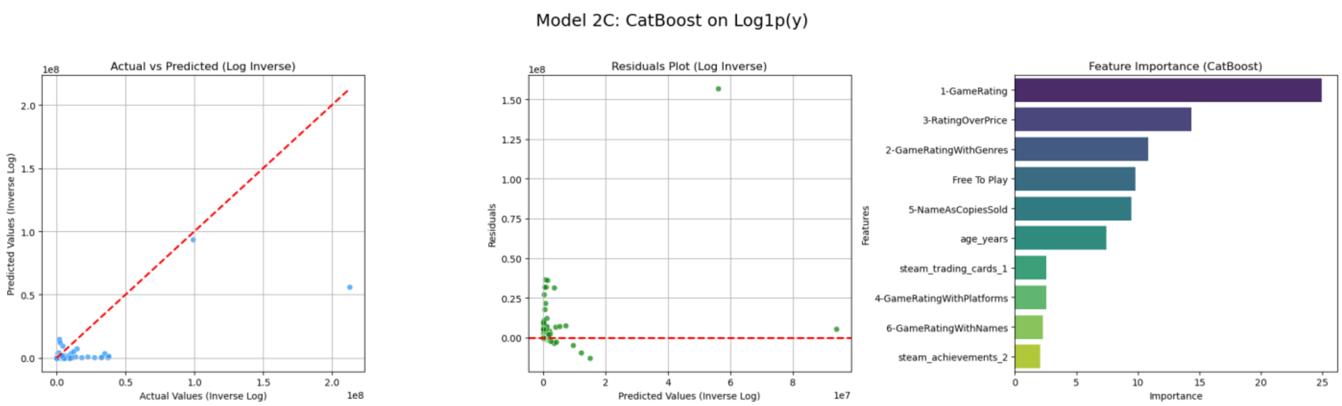
Model 2L: LGBM on Log1p(y)



CatBoost (w/ GridSearch):

```
Original y:
Mean Squared Error: 1298143124204.8699
Root Mean Squared Error (RMSE): 1139360.84
Mean Absolute Error: 103725.97993923663
R^2 Score: 0.7251864567019164

Log1p-transformed y:
Mean Squared Error: 2460193345830.2173
Root Mean Squared Error (RMSE): 1568500.35
Mean Absolute Error: 78967.46622992316
R^2 Score: 0.4791834290382373
```





Concluding Remarks

After comparison of models in terms of (MSE, MAE, R^2): **CatBoost** (with original target) is the best model for deployment.

Intuition:

- Features such as (name, genres, release date, price, platforms, publisher and review score) are expected to have the highest effect on our prediction.
- Features such as (achievements, trading cards and workshop support) are expected to have no significant effect.

Actual:

- The first intuition was correct.
- However, our second intuition was wrong, those features did have an effect in feature engineering better features for our prediction.