# Video Game Average User Score Classification Project

Francisco Trejo
7/20/2022

# TABLE OF CONTENTS

## 1. PROBLEM STATEMENT

Why does this matter?

## 2. DATA

How did we get this data?

## 3. EDA AND PREPROCESSING

Explore the data and get it ready for modeling.

## 4. MODELING

Fine tune models and extract coefficients.

## 5. CONCLUSIONS AND RECOMMENDATIONS

What did we learn and how can we improve?

# 01

## Problem Statement

# PROBLEM BACKGROUND

Video games take years of development time and cost anywhere from a few million to well over a hundred million dollars to produce. There are plenty of examples of video games that are poorly received and never establish a lasting player base. There are a few potential metrics that may determine how a game performs. One metric that can determine how a game performs is video game reviews, particularly user scores for a game.

# PROBLEM STATEMENT

Using video game reviews from Metacritic we want to create a model that will help us identify the attributes of a video game that will help it get a high average user score.

# 02

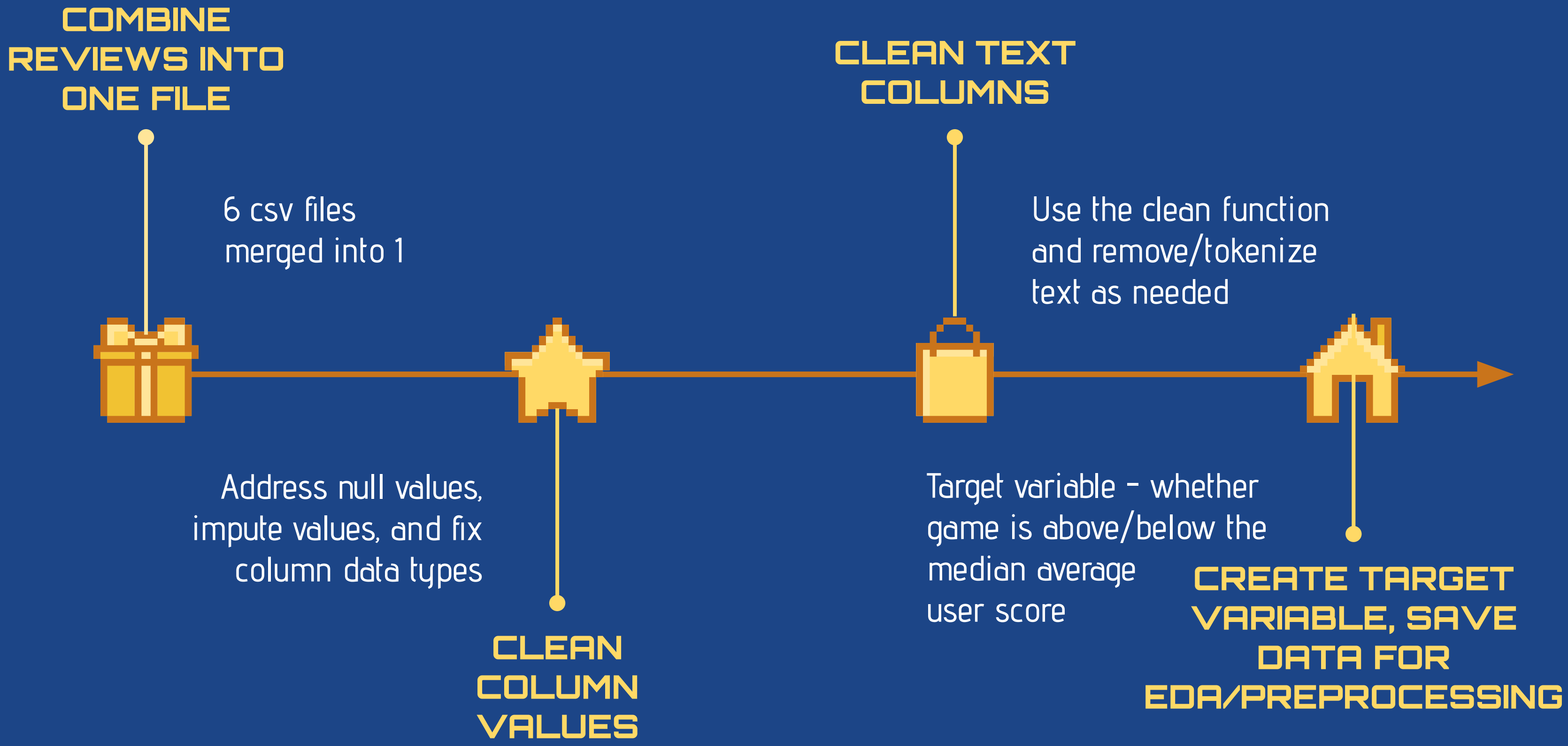
# Data

metacritic

## Data Collection

- Scraped user reviews from Metacritic for 6 different consoles
- Limited to 1000 reviews max per game
- Over 100,000 user reviews for over 400 games

# DATA CLEANING



**COMBINE REVIEWS INTO ONE FILE**

6 csv files merged into 1

**CLEAN COLUMN VALUES**

Address null values, impute values, and fix column data types

**CLEAN TEXT COLUMNS**

Use the clean function and remove/tokenize text as needed

**CREATE TARGET VARIABLE, SAVE DATA FOR EDA/PREPROCESSING**

Target variable - whether game is above/below the median average user score

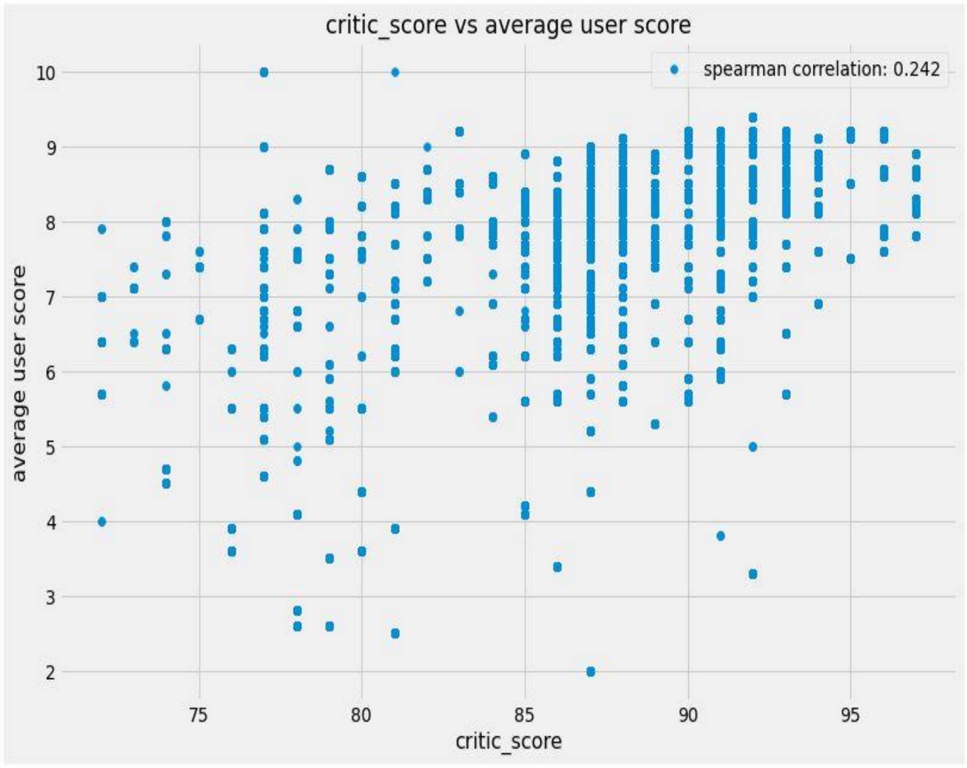
# 03

# EDA & PREPROCESSING

# EDA

## Continuous Variables

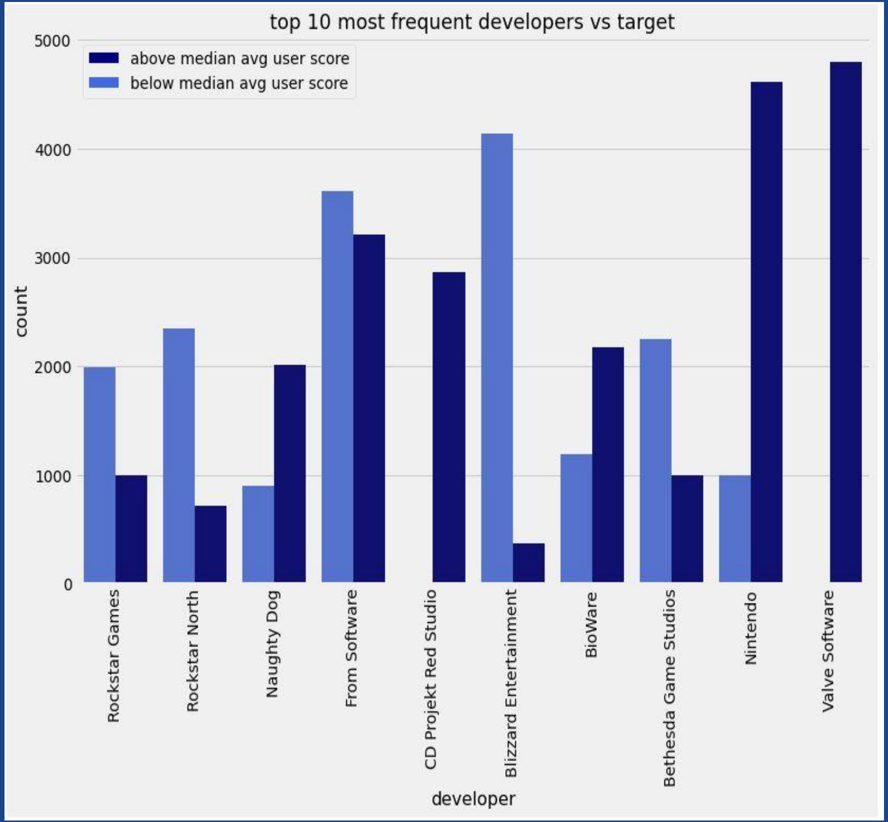
- Looked at descriptive statistics and boxplot to identify potential outliers
- Looked at histograms to see distribution of data
- Looked at scatter plots to see any potential relationships with target



# EDA

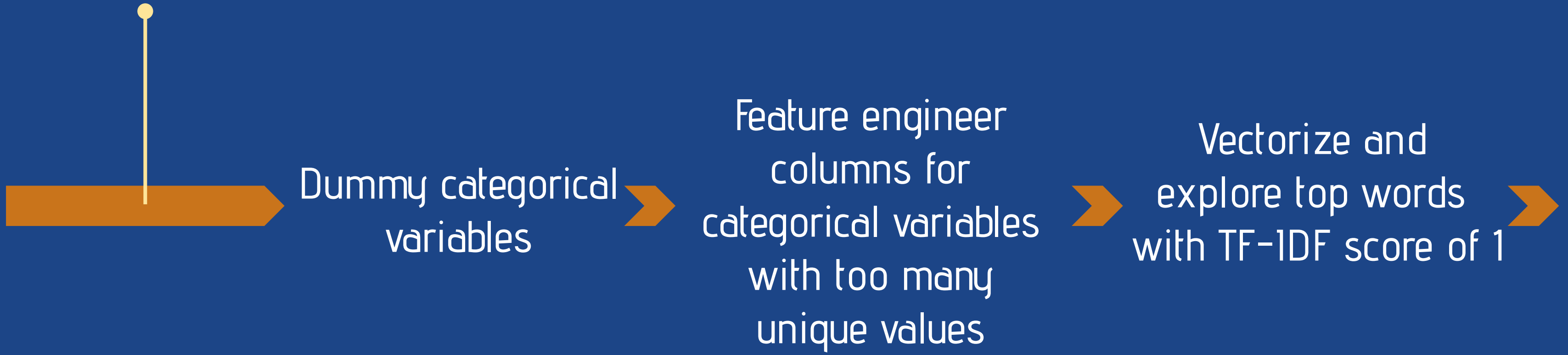
## Categorical Variables

- Looked at bar plots with value counts
- For columns with too many unique values looked at top 10
- Used grouped bar charts to see counts above/below target



# PREPROCESSING

## STEPS TAKEN



Dummy categorical variables

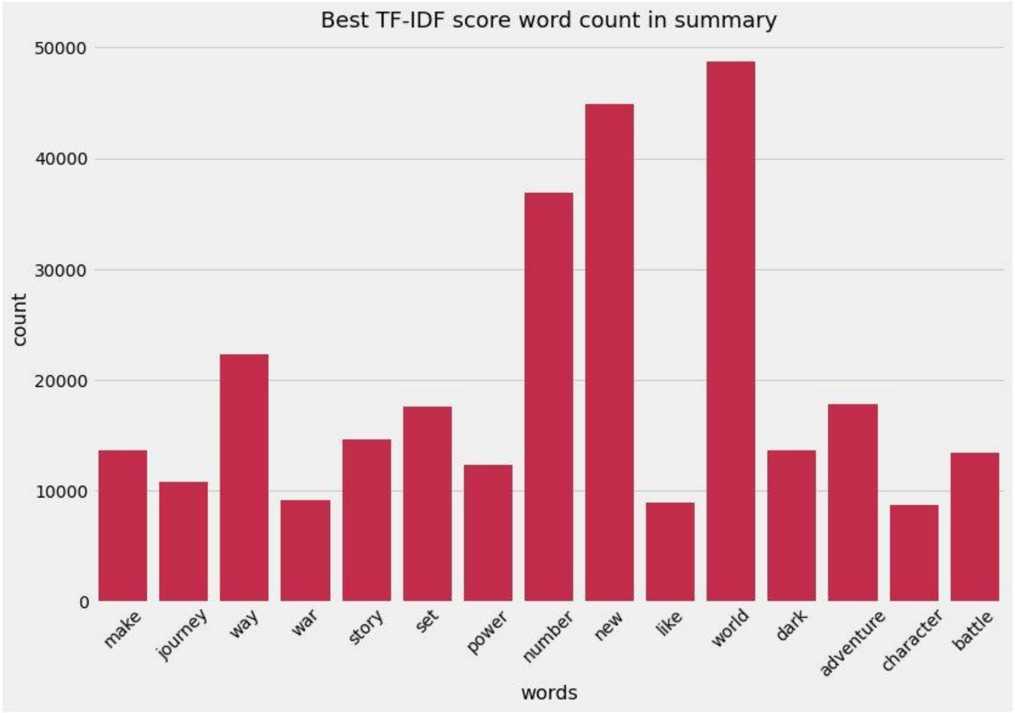
Feature engineer columns for categorical variables with too many unique values

Vectorize and explore top words with TF-1DF score of 1

# PREPROCESSING

## Vectorized Text

- Plotted words with TF-IDF score of 1
- Top 3 words showed up over 35,000 times
- Most words show up 10,000-20,000 times

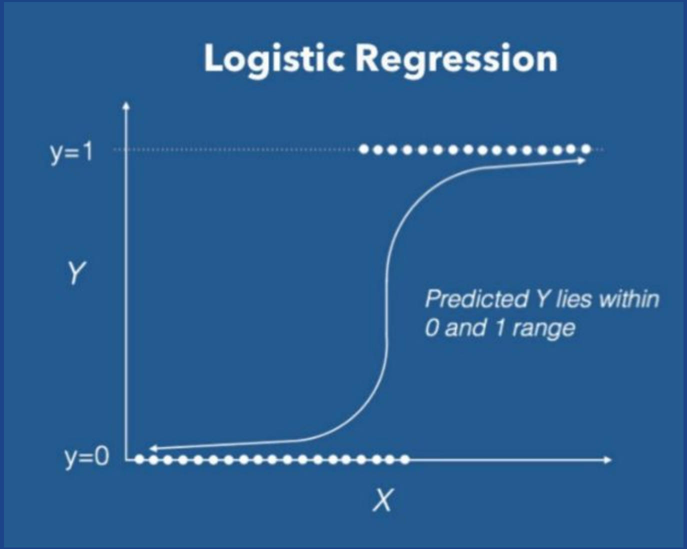


# 04

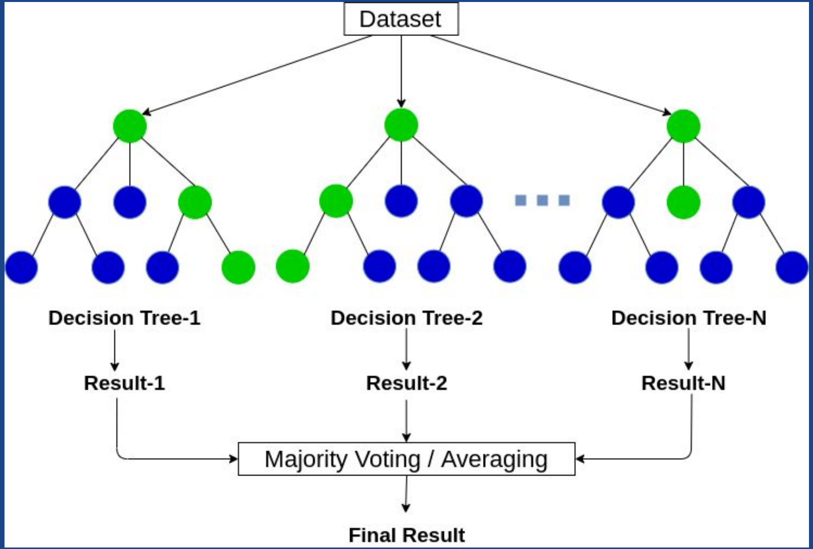
# MODELING

# MODEL TYPES

## Logistic Regression



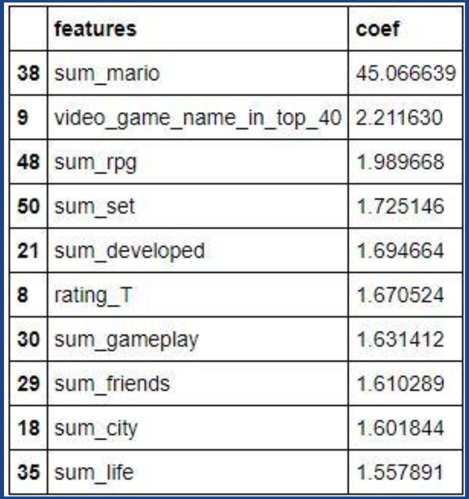
## Random Forests



Grid search was used for both models to find the best hyperparameters

# MODEL ATTRIBUTES

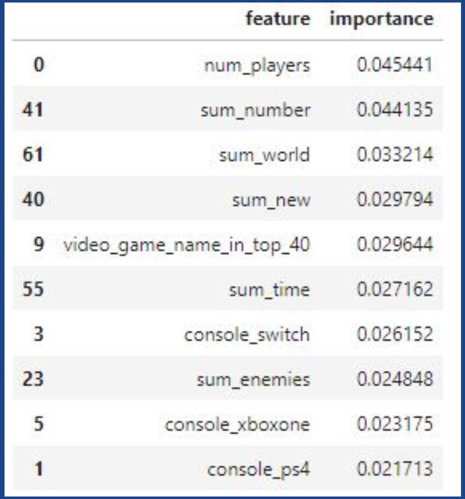
## Logistic Regression



| | features | coef |
|---|---|---|
| 38 | sum_mario | 45.066639 |
| 9 | video_game_name_in_top_40 | 2.211630 |
| 48 | sum_rpg | 1.989668 |
| 50 | sum_set | 1.725146 |
| 21 | sum_developed | 1.694664 |
| 8 | rating_T | 1.670524 |
| 30 | sum_gameplay | 1.631412 |
| 29 | sum_friends | 1.610289 |
| 18 | sum_city | 1.601844 |
| 35 | sum_life | 1.557891 |

- Top 10 coefficients
- Interpretable results (coefficients exponentiated)
- Best coefficient increases likelihood of above the median average user score by 45 times

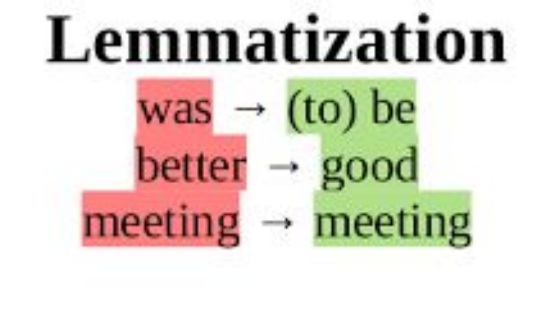
## Random Forests



| | feature | importance |
|---|---|---|
| 0 | num_players | 0.045441 |
| 41 | sum_number | 0.044135 |
| 61 | sum_world | 0.033214 |
| 40 | sum_new | 0.029794 |
| 9 | video_game_name_in_top_40 | 0.029644 |
| 55 | sum_time | 0.027162 |
| 3 | console_switch | 0.026152 |
| 23 | sum_enemies | 0.024848 |
| 5 | console_xboxone | 0.023175 |
| 1 | console_ps4 | 0.021713 |

- Top 10 features
- Larger importance val = more important but not very descriptive
- Better score but we want more interpretable coefficients

# LEMMATIZATION AND RESULTS



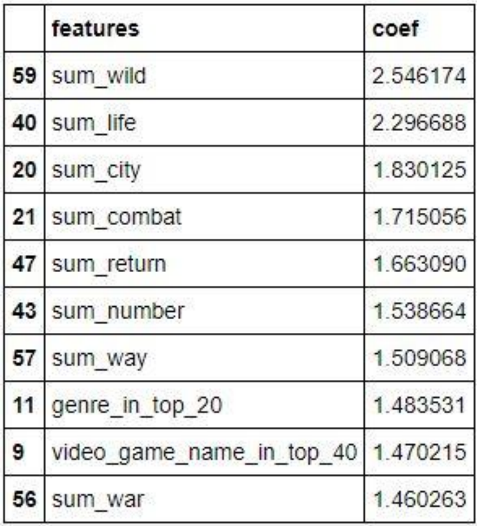
Lemmatization

was → (to) be

better → good

meeting → meeting

Aiming to remove inflectional endings only and to return the base or dictionary form of a word.



|  | features | coef |
|---|---|---|
| 59 | sum_wild | 2.546174 |
| 40 | sum_life | 2.296688 |
| 20 | sum_city | 1.830125 |
| 21 | sum_combat | 1.715056 |
| 47 | sum_return | 1.663090 |
| 43 | sum_number | 1.538664 |
| 57 | sum_way | 1.509068 |
| 11 | genre_in_top_20 | 1.483531 |
| 9 | video_game_name_in_top_40 | 1.470215 |
| 56 | sum_war | 1.460263 |

- Coefficients changed
- No one coefficient dominating
- Improved accuracy score by 0.48%

# LOGISTIC REGRESSION LEMMATIZED TEXT

87.11%

Accuracy Score

Baseline: 50%

# LOGISTIC REGRESSION LEMMATIZED TEXT

**0.8719 ± 0.0112**

Confidence Interval

# 05

# CONCLUSIONS AND RECOMMENDATIONS

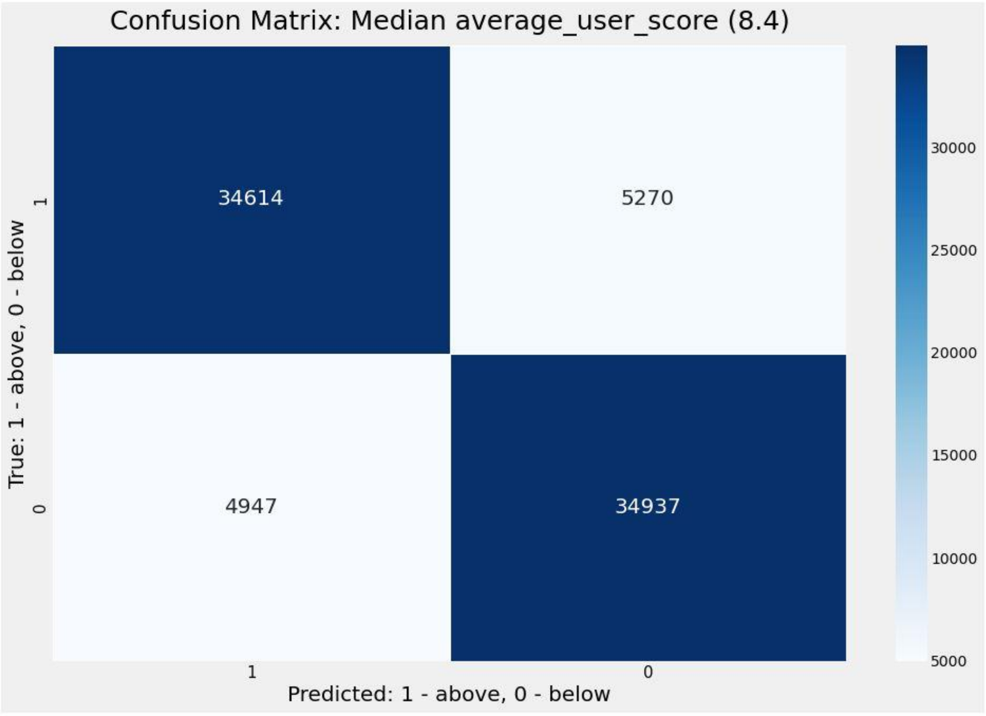
# CONCLUSIONS

## Top 5 coefficients

- Sum_wild
- Sum_life
- Sum_city
- Sum_combat
- Sum_return
- All increase likelihood of a game being above the median average user score by 2.54 - 1.66 times

## Confusion Matrix

- Low number of false negatives and false positives
- Accuracy score of 87.11%



# RECOMMENDATIONS

## How can we do better?

- Collect more data!
- Collect video game info by year!
- Streamline process to return predictions based on certain parameters, potentially host online as well!

# THANKS!

CREDITS: This presentation template was created by Slidesgo,
including icons by Flaticon, and infographics & images by Freepik.

**Please keep this slide for attribution.**