Yanis Benneouala
DAFT NOV2021

VIDEO GAMES

And ratings

February 11th, 2022

TABLE OF CONTENTS

01.

USE CASE

02.

EDA

03.

DATA CLEANING

04.

MODELS & EVALUATION

05.

CONCLUSIONS

01.

USE CASE

In 2021, the video game market size in the United States surpassed 85.86 billion U.S. dollars. Worldwide, it generated total revenues of 180.3 billion U.S. dollars

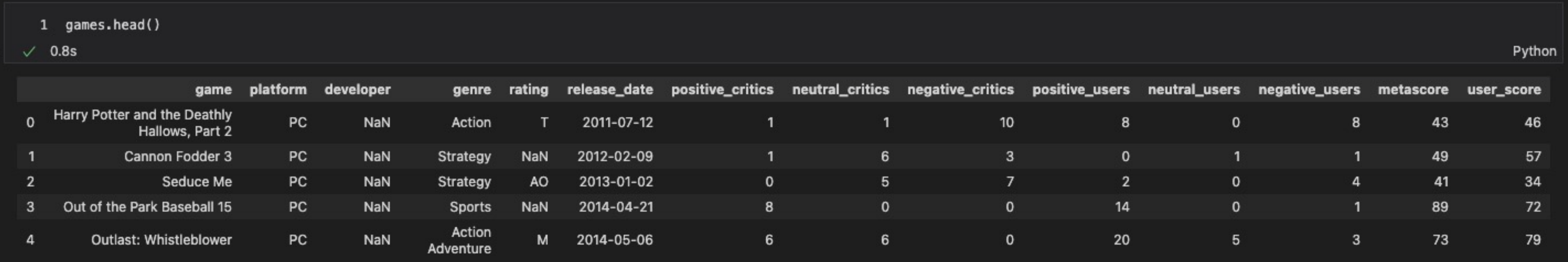
CAN RATINGS BE ACCURATELY PREDICTED ?
KNOWING THAT : RATING = SUCCESS

02.

EDA

Dataset

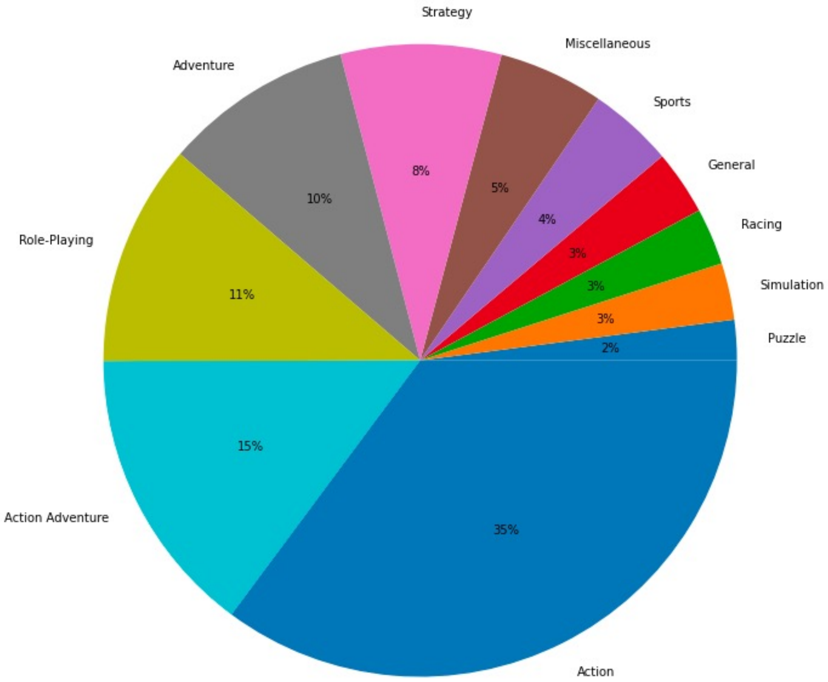
Dataset overview



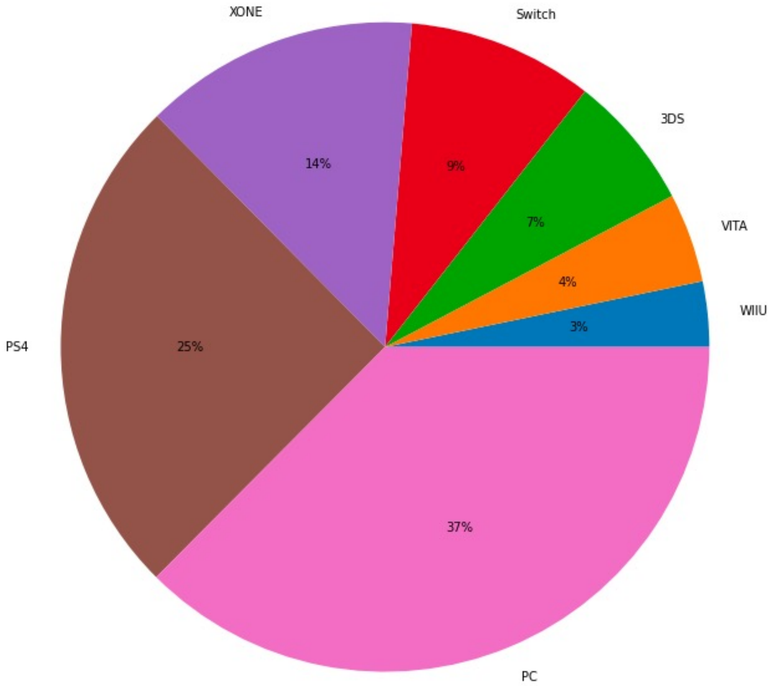
```
games.head()
```

	game	platform	developer	genre	rating	release_date	positive_critics	neutral_critics	negative_critics	positive_users	neutral_users	negative_users	metascore	user_score
0	Harry Potter and the Deathly Hallows, Part 2	PC	NaN	Action	T	2011-07-12	1	1	10	8	0	8	43	46
1	Cannon Fodder 3	PC	NaN	Strategy	NaN	2012-02-09	1	6	3	0	1	1	49	57
2	Seduce Me	PC	NaN	Strategy	AO	2013-01-02	0	5	7	2	0	4	41	34
3	Out of the Park Baseball 15	PC	NaN	Sports	NaN	2014-04-21	8	0	0	14	0	1	89	72
4	Outlast: Whistleblower	PC	NaN	Action Adventure	M	2014-05-06	6	6	0	20	5	3	73	79

MOST REPRESENTED GENRES

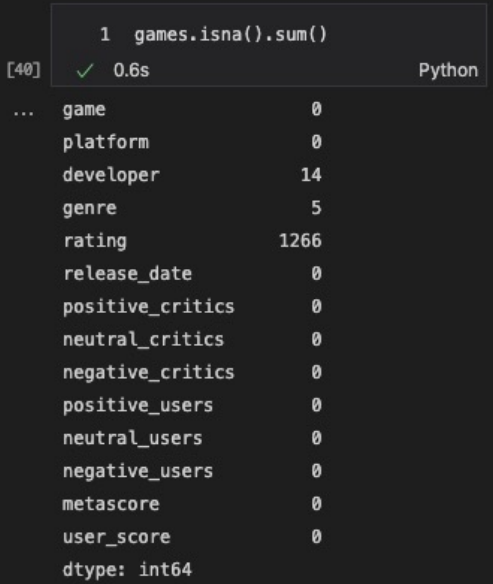


MOST REPRESENTED PLATFORMS



03.
DATA CLEANING

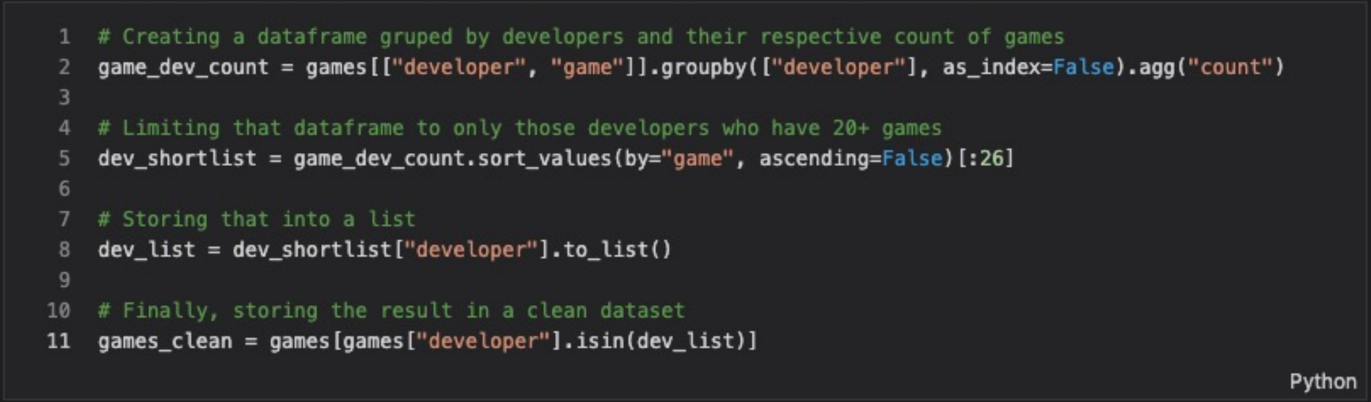
Missing values :



```python
games.isna().sum()
```

```
game                   0
platform               0
developer             14
genre                  5
rating              1266
release_date           0
positive_critics       0
neutral_critics        0
negative_critics       0
positive_users         0
neutral_users          0
negative_users         0
metascore              0
user_score             0
dtype: int64
```

Filtering to work only on developers with 20+ games



```python
# Creating a dataframe gruped by developers and their respective count of games
game_dev_count = games[["developer", "game"]].groupby(["developer"], as_index=False).agg("count")

# Limiting that dataframe to only those developers who have 20+ games
dev_shortlist = game_dev_count.sort_values(by="game", ascending=False)[:26]

# Storing that into a list
dev_list = dev_shortlist["developer"].to_list()

# Finally, storing the result in a clean dataset
games_clean = games[games["developer"].isin(dev_list)]
```

04.

MODELS & EVALUATION

TWO APPROACHES :

Time-series Analysis

Using Auto-ARIMA

Classification

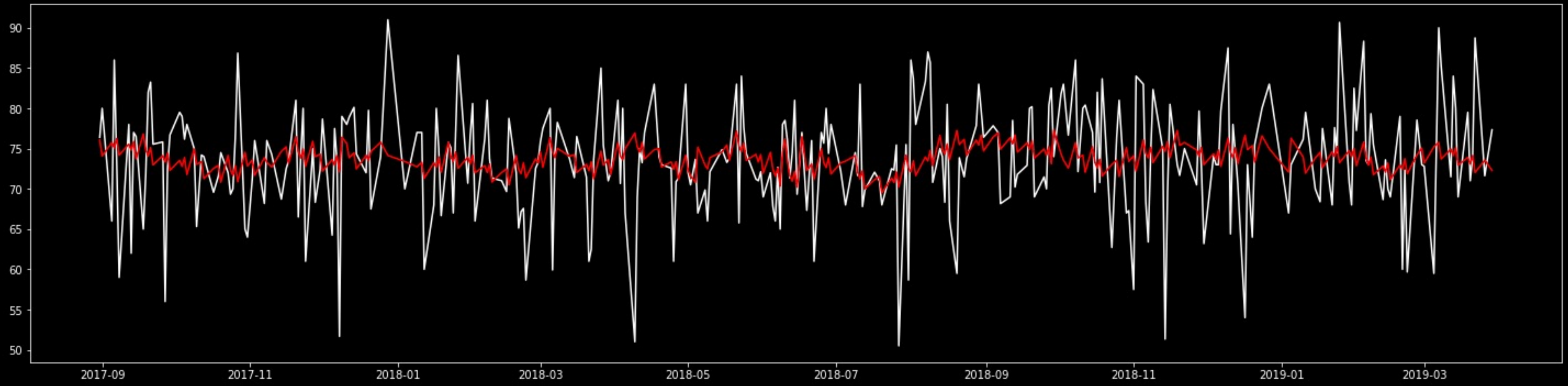
Using TPOT and ExrtraTreeClassifier

Time-series Analysis

>>> We try determine whether the rating of a game is dependant on its release date

Time-series Analysis

ARIMA(5,1,0)



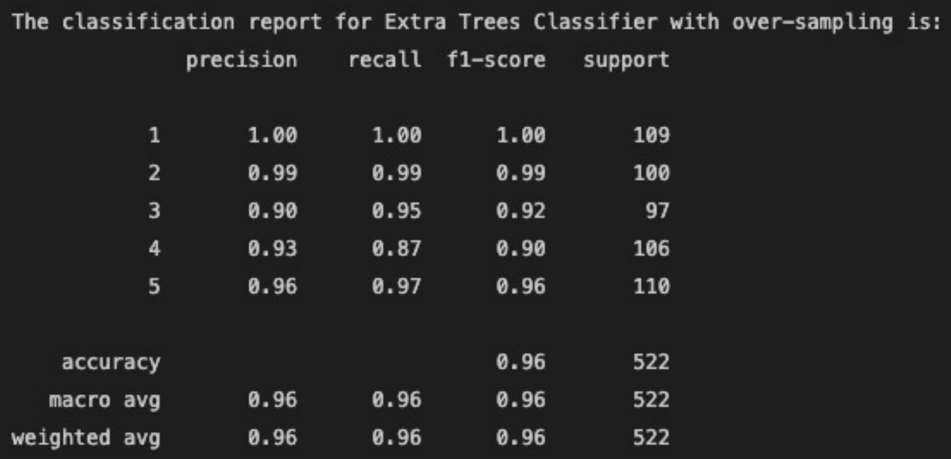
Classification

>>> Here, we are trying to find the best model in order to classify the games, basically by good or bad

Classification

Optimal classifier, found with TPOT :

ExtraTreesClassifier(CombineDFs(bootstrap=False, criterion=gini, max_features=0.9000000000000001, min_samples_leaf=5, min_samples_split=3, n_estimators=100)



```
The classification report for Extra Trees Classifier with over-sampling is:
              precision    recall  f1-score   support

           1       1.00      1.00      1.00       109
           2       0.99      0.99      0.99       100
           3       0.90      0.95      0.92        97
           4       0.93      0.87      0.90       106
           5       0.96      0.97      0.96       110

    accuracy                           0.96       522
   macro avg       0.96      0.96      0.96       522
weighted avg       0.96      0.96      0.96       522
```

05.

CONCLUSIONS

THANKS!

Do you have any questions?