

A Deep Dive Into Video Game Ratings



<https://www.kaggle.com/datasets/muhammadadiltalay/imdb-video-games>

Jacob Russell, Croix Westbrook, Blake Nelson

Ask A Question

Project Goal

- Question
 - Do the genre and certificate of a game have a significant impact on its overall rating (1-10 score) in IMDB?
- Possible patterns / predictions
 - Certain genres will score higher on average than others
 - Games that have less restrictive certificates may reach more consumers and perform better



Dataset Information

- The dataset we used is a collection of data from video games rated on the IMDB website ([Dataset](#)).
- The data includes:
 - Name - string
 - Year - int
 - Certificate - string
 - Rating - int
 - Votes - int
 - Genre - boolean
- Size
 - 20804 data entries

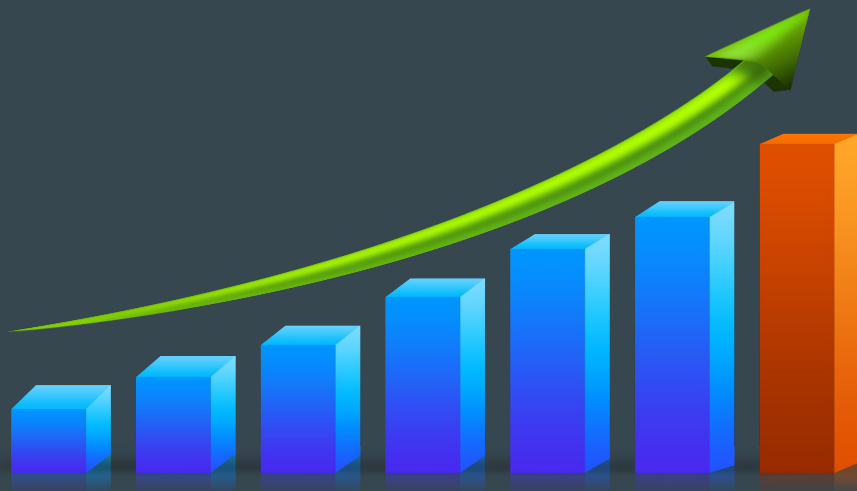
The IMDb logo is displayed in a bold, black, sans-serif font. The letters 'I', 'M', 'D', and 'b' are connected, while the 'I' and 'M' are slightly separated from the 'D'. The logo is set against a bright yellow rectangular background with rounded corners.

Prior to Cleaning

Unnamed: 0		name	url	year	certificate	rating	votes	plot	Action	Adventure
0	0	Spider-Man	https://www.imdb.com/title/tt5807780/?ref_=adv...	2018.0	T	9.2	20,759	When a new villain threatens New York City, Pe...	True	True
1	1	Red Dead Redemption II	https://www.imdb.com/title/tt6161168/?ref_=adv...	2018.0	M	9.7	35,703	Amidst the decline of the Wild West at the tur...	True	True
2	2	Grand Theft Auto V	https://www.imdb.com/title/tt2103188/?ref_=adv...	2013.0	M	9.5	59,986	Three very different criminals team up for a s...	True	False
3	3	God of War	https://www.imdb.com/title/tt5838588/?ref_=adv...	2018.0	M	9.6	26,118	After wiping out the gods of Mount Olympus, Kr...	True	True

Benefits

- Based on the results, we hope to be able to determine what makes a game more likely to have a higher rating and if there is a way to maximize a games potential before development.



Get and Prepare the Data

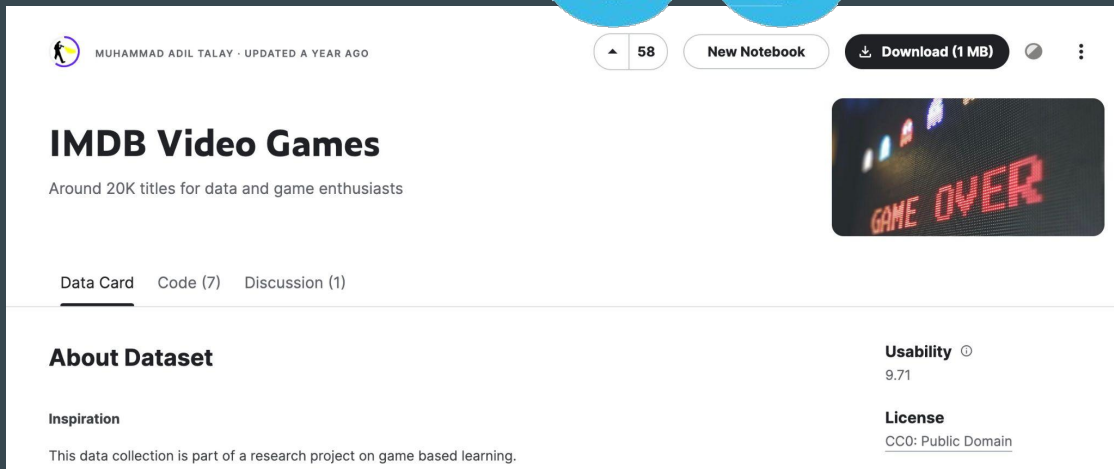
Where did our data come from?

We are using a public domain IMDB dataset from Kaggle.

We did not have a reason to scrape our data from the web.

High usability score of 9.71/10 on Kaggle's scoring system.

kaggle



The screenshot shows the Kaggle dataset page for 'IMDB Video Games'. At the top, it says 'MUHAMMAD ADIL TALAY · UPDATED A YEAR AGO'. There are buttons for 'New Notebook' and 'Download (1 MB)'. The dataset title 'IMDB Video Games' is prominently displayed, followed by the description 'Around 20K titles for data and game enthusiasts'. Below this, there are tabs for 'Data Card', 'Code (7)', and 'Discussion (1)'. The 'Data Card' is selected. The 'About Dataset' section includes an 'Inspiration' link and a description: 'This data collection is part of a research project on game based learning.' On the right side, the 'Usability' score is 9.71, and the 'License' is 'CC0: Public Domain'. A small image of a 'GAME OVER' sign is also visible.

MUHAMMAD ADIL TALAY · UPDATED A YEAR AGO

58 New Notebook Download (1 MB)

IMDB Video Games

Around 20K titles for data and game enthusiasts

Data Card Code (7) Discussion (1)

About Dataset

Inspiration

This data collection is part of a research project on game based learning.

Usability 9.71

License CC0: Public Domain

How did we clean the data?

- Removed duplicates
- Dropped NaN values
- Remove unnecessary columns
 - 'Unnamed: 0'
 - 'name'
 - 'url'
 - 'plot'
 - 'year'
- Cleaned votes to remove commas between numbers

```
df = df.drop_duplicates()
df = df.drop(columns=['Unnamed: 0', 'name', 'year', 'url', 'plot'])

df = df.dropna()

df['votes'] = df['votes'].str.replace(',', '').astype(int)
df = df.loc[df['votes'] >= 100, :]
df
```

Did we need to use multiple data sets?

No we did not. We put trust into IMDB to be a credible source for data.

This dataset has information from around 20k titles.

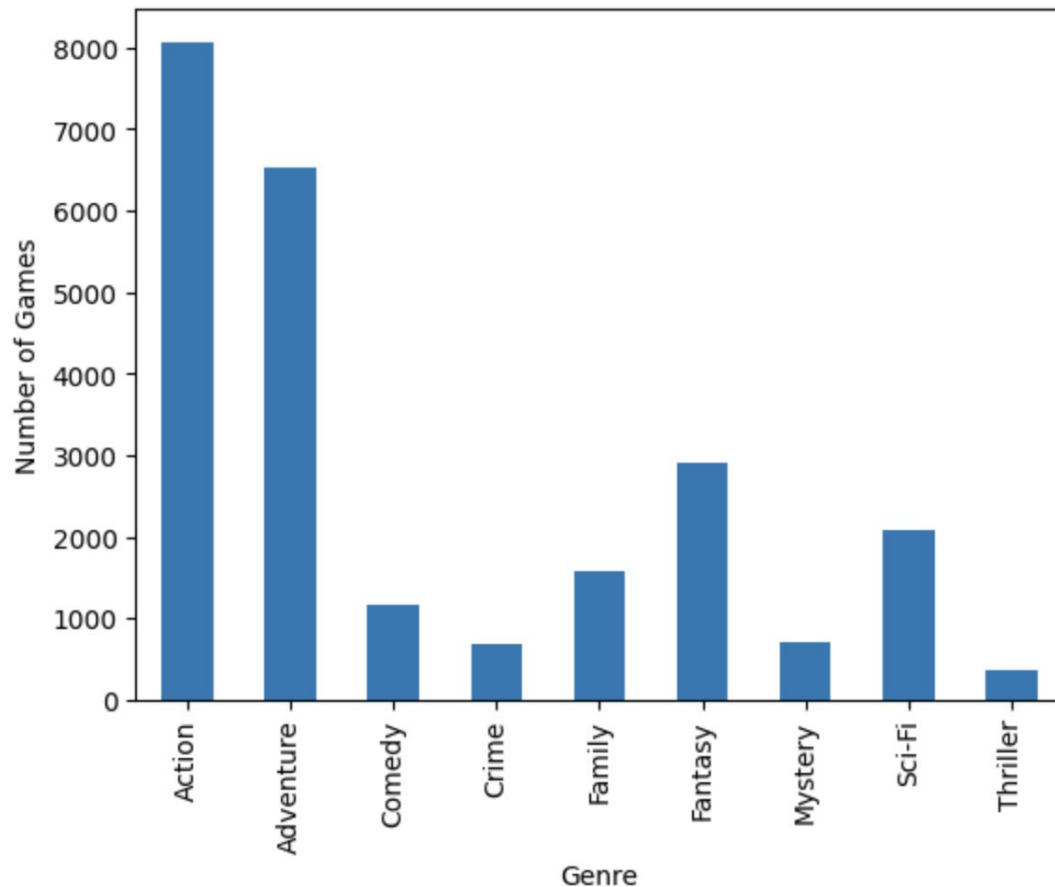
After cleaning the data, we still have above 3k titles to train our models on.

	certificate	rating	votes	Action
0	T	9.2	20759	True
1	M	9.7	35703	True
2	M	9.5	59986	True
3	M	9.6	26118	True
4	T	9.5	28722	True
...
3310	M	7.8	5029	True
3311	M	8.1	1357	False
3312	M	8.1	103	False
3313	M	4.9	197	False
3314	M	7.9	333	True
3315 rows x 12 columns				

Explore The Data

Genre Totals

<Axes: xlabel='Genre', ylabel='Number of Games'>

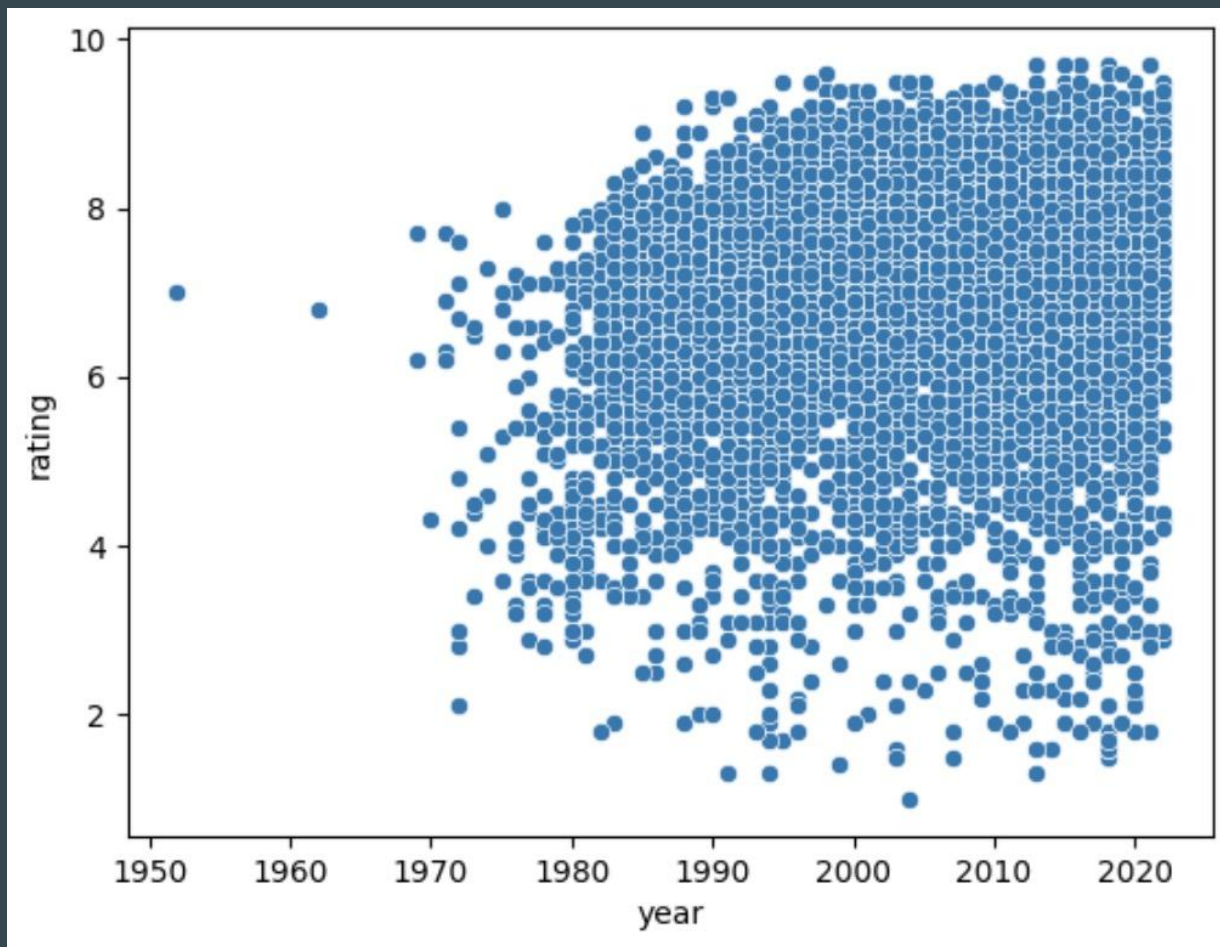


Descriptive Analysis

- Total Game Counts
 - Action and Adventure have most by far
- Small correlations overall

	rating	Action	Adventure	Comedy	Crime	Family	Fantasy	Mystery	Sci-Fi	Thriller	certificate_E	certificate_E10+	certificate_M	certificate_T
rating	1.000000	-0.014190	0.113191	-0.022854	-0.001054	-0.112482	0.097069	0.066282	-0.033889	-0.046338	-0.055288	-0.065581	0.111439	-0.025518
Action	-0.014190	1.000000	0.058034	-0.118929	0.040347	-0.162226	-0.032060	-0.258077	0.022116	-0.086676	-0.108294	-0.027045	0.058518	0.046891
Adventure	0.113191	0.058034	1.000000	0.045899	-0.218322	0.010764	0.188603	-0.148973	-0.201422	-0.232944	0.088446	0.079249	-0.178371	0.057338
Comedy	-0.022854	-0.118929	0.045899	1.000000	-0.049281	-0.036297	-0.186418	-0.081064	-0.114536	-0.088057	0.015284	0.157455	-0.085817	-0.026380
Crime	-0.001054	0.040347	-0.218322	-0.049281	1.000000	-0.132590	-0.189918	0.032126	-0.116399	0.054113	-0.130212	-0.098060	0.225064	-0.058316
Family	-0.112482	-0.162226	0.010764	-0.036297	-0.132590	1.000000	-0.108094	-0.079100	-0.138716	-0.091978	0.431218	0.164573	-0.287686	-0.167754
Fantasy	0.097069	-0.032060	0.188603	-0.186418	-0.189918	-0.108094	1.000000	-0.138200	-0.176661	-0.149763	-0.010692	-0.048828	-0.094425	0.135605
Mystery	0.066282	-0.258077	-0.148973	-0.081064	0.032126	-0.079100	-0.138200	1.000000	-0.052751	0.059723	-0.074115	-0.062398	0.143863	-0.044936
Sci-Fi	-0.033889	0.022116	-0.201422	-0.114536	-0.116399	-0.138716	-0.176661	-0.052751	1.000000	0.038777	-0.107851	-0.068786	0.087127	0.044265
Thriller	-0.046338	-0.086676	-0.232944	-0.088057	0.054113	-0.091978	-0.149763	0.059723	0.038777	1.000000	-0.113859	-0.080477	0.183374	-0.040771
certificate_E	-0.055288	-0.108294	0.088446	0.015284	-0.130212	0.431218	-0.010692	-0.074115	-0.107851	-0.113859	1.000000	-0.162684	-0.363434	-0.348297
certificate_E10+	-0.065581	-0.027045	0.079249	0.157455	-0.098060	0.164573	-0.048828	-0.062398	-0.068786	-0.080477	-0.162684	1.000000	-0.256881	-0.246181
certificate_M	0.111439	0.058518	-0.178371	-0.085817	0.225064	-0.287686	-0.094425	0.143863	0.087127	0.183374	-0.363434	-0.256881	1.000000	-0.549967
certificate_T	-0.025518	0.046891	0.057338	-0.026380	-0.058316	-0.167754	0.135605	-0.044936	0.044265	-0.040771	-0.348297	-0.246181	-0.549967	1.000000

Why we went back and Dropped Year



Hypothesis

- Certificate and Genre will be highly correlated with the overall rating of a video game in IMDBs data and will allow us to predict the rating of games in the future accurately

Problems with the Dataset

- Game Number Discrepancy
- Compare various vote counts?
- Low correlation levels
- Games have multiple genres
 - Red Dead Redemption II is action, adventure, and crime game.
 - Elden Ring is action, adventure, and fantasy.



Improvements to the Dataset

- Consistent data types - no object types
- Then, numeric data for analyzing
- Drop < 100 ratings

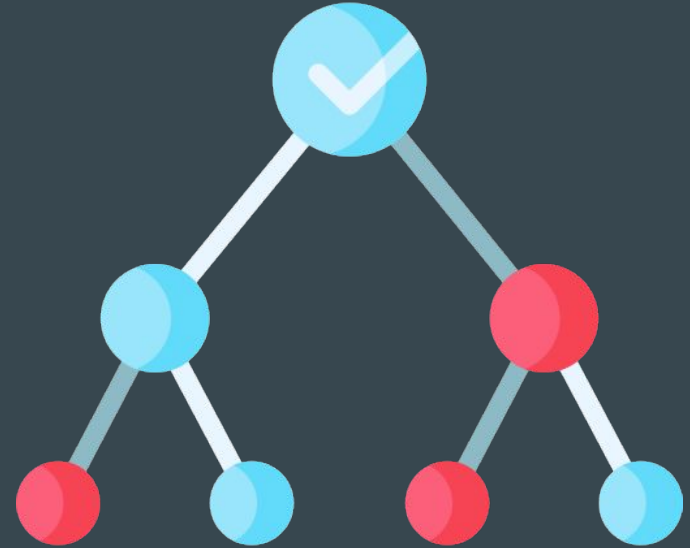
```
df['votes'] = df['votes'].str.replace(',', '', '').astype(int)
df = df.loc[df['votes'] >= 100, :]
df.reset_index(inplace=True, drop=True)
```


Model the Data

What are we trying to predict?

We are using genre and certificate to determine how well a game will perform in rating points overall on IMDB.

To do this, we are analyzing our data that has had its dimensions reduced using PCA. The models that we are using are KNN, Logistic Regression, Decision Tree, and SVC.



Why did we not scale our data?

One Hot Encoding

No need to scale 1s and 0s

	rating	Action	Adventure	Comedy	Crime	Family	Fantasy	Mystery	Sci-Fi	Thriller	certificate_E	certificate_E10+	certificate_M	certificate_T
0	9.2	1	1	0	0	0	1	0	0	0	0	0	0	1
1	9.7	1	1	0	1	0	0	0	0	0	0	0	1	0
2	9.5	1	0	0	1	0	0	0	0	0	0	0	1	0
3	9.6	1	1	0	0	0	0	0	0	0	0	0	1	0
4	9.5	1	1	0	0	0	0	0	0	0	0	0	0	1
...
20716	7.8	1	0	0	0	0	0	1	1	0	0	0	1	0
20717	8.1	0	0	0	0	0	0	0	0	1	0	0	1	0
20718	8.1	0	0	0	0	0	0	0	0	1	0	0	1	0
20725	4.9	0	0	0	0	0	0	0	0	0	0	0	1	0
20735	7.9	1	0	0	1	0	0	0	0	1	0	0	1	0

Limiting the Scope

Additionally, it was important to use a Principal Component Analysis, because with so many factors being taken into account, it is important to limit our scope and focus on the factors/variables that contribute the most to finding a relationship to rating.

```
from sklearn.decomposition import PCA

pca = PCA()
X = pca.fit(scaledDf).transform(scaledDf)

pd.DataFrame(X)
```

	0	1	2	3	4	5	6	7	8
0	-1.509611	-0.928773	0.908494	0.059360	-0.119914	0.321452	0.243582	-0.008650	0.294382
1	0.937081	-0.633896	0.523910	2.311790	-1.068248	-1.575811	-0.491423	1.622971	0.287132
2	2.051610	-0.590780	0.445974	1.891949	-1.505414	-1.663430	0.888581	0.769301	-0.047636
3	-0.457116	-0.405560	-0.178091	0.307331	0.051030	-0.172431	-0.888593	-0.114050	-0.684546
4	-0.457116	-0.405560	-0.178091	0.307331	0.051030	-0.172431	-0.888593	-0.114050	-0.684546
...
5649	0.792460	1.239810	0.259822	-0.564819	0.150703	0.241450	1.133436	0.467286	-1.761212
5650	3.125423	0.904947	1.085430	-0.406262	-1.122455	4.556601	-0.496387	0.552778	-0.831761
5651	3.125423	0.904947	1.085430	-0.406262	-1.122455	4.556601	-0.496387	0.552778	-0.831761
5652	3.125423	0.904947	1.085430	-0.406262	-1.122455	4.556601	-0.496387	0.552778	-0.831761
5653	4.384573	-0.925644	1.271582	2.050505	-2.778572	2.651721	-0.741243	0.854792	0.881814

5654 rows x 9 columns

Why did we decide to use the models that we did?

We decided to use KNN, Logistic Regression, and Decision Tree because they are all very similar in their ease of use and set up. By using all 3 simultaneously, we are able to compare the accuracy scores and determine which model best represents and predicts our data.

```
knn2 = KNeighborsClassifier(n_neighbors= 2)
knn5 = KNeighborsClassifier(n_neighbors= 5)
knn8 = KNeighborsClassifier(n_neighbors= 8)
tree = DecisionTreeClassifier()
tree3 = DecisionTreeClassifier(max_depth= 3)
tree5 = DecisionTreeClassifier(max_depth= 5)
tree8 = DecisionTreeClassifier(max_depth= 8)
log_reg = LogisticRegression()
svc = SVC()
```

What is our training split?

We used a stratified KFold to split our data.

In a typical training split, a set percentage of the data is used to test. Normally 20% testing, 80% training.


However, because we have so much variation between the total numbers of games in each data, we break it down into stratified KFold. This helps us keep the distribution of genres more even within our training and testing splits.

How did we show the accuracy of our models?

The accuracy of our models is calculated by using stratified KFold.

For each split we fit every model to the training data and used the accuracy_score function from sklearn to calculate the accuracy using the actual values and the predicted values of y.

```
KNeighborsClassifier(n_neighbors=2)
0.2774584929757343
KNeighborsClassifier()
0.3537675606641124
KNeighborsClassifier(n_neighbors=8)
0.38409961685823757
DecisionTreeClassifier()
0.43263090676883775
DecisionTreeClassifier(max_depth=3)
0.4492337164750958
DecisionTreeClassifier(max_depth=5)
0.44667943805874843
DecisionTreeClassifier(max_depth=8)
0.43582375478927204
LogisticRegression()
0.45721583652618136
SVC()
0.45721583652618136
```



Feature Engineering and Data Augmentation

We used One-Hot-Encoding to break down the four certificate string values into 1s and 0s respectively.

No data augmentation used, as we already have a plethora of data.

certificate
T
M
M
M
T
...
M
M
M
M
M



certificate_E	certificate_E10+	certificate_M	certificate_T
0	0	0	1
0	0	1	0
0	0	1	0
0	0	1	0
0	0	0	1
...
0	0	1	0
0	0	1	0
0	0	1	0
0	0	1	0
0	0	1	0

Model Evaluation and Other Machine Learning Algorithms

Our machine learning algorithms that we used were KNN, LogisticRegression, DecisionTree, and SVC.

It was important to use multiple different algorithms in order to cross check their accuracies and choose the best model for predicting our data and answering testing our hypothesis.

Communicate the Data

Analysis of the Data

The maximum score of the models we used was the Logistic Regression model with a score of 0.4572. This means that the model has around 45% accuracy when predicting the rating of a video game, given the certificate and genre data. Although 45% is a decent accuracy score, this would not be a valid score for real world usage.

Applications of our data

Because the accuracy of our tests is on the lower side of things, our real-world applications are quite limited. We only have the ability to predict game scores with an accuracy of around 45%.

Completely random guesses between 1-10 would be around 10%.

If we revisit our hypothesis: “Certificate and Genre will be highly correlated with the overall rating of a video game in IMDBs data and will allow us to predict the rating of games in the future accurately” we find that this is not exactly accurate.

Link to Our Project

<https://colab.research.google.com/drive/1nUpkvEUmUMNg2-F0l2xk2y2AlVG1btCH>

- To run this, you will need to download the dataset and import it into the Google Colab