Video Game Score/Sales Analysis
ML Hackathon Documentation

Ram S
IMT2017521

Rathin Bhargava
IMT2017522

***Abstract*—There are thousands of video games being released every year on different platforms, different genres and different regions. Gaming is now a world-wide phenomenon and a billion dollar business. However, there are a multitude of games that are released which are poor in quality, which plagiarize from other games and are which are fundamentally flawed. So, it has become an increasingly common practice for game publishers to look at Metacritic scores (Metacritic scores are an amalgamation of scores for games by reputed critics in the industry) to decide/predict a game's success. We wish to analyze the factors that affect such scores as well as perform sales prediction based on such Scores.**

***Index Terms*—Metacritic, Sales Prediction, Gaming**

I. Data Acquisition

We acquired a data set from Kaggle called: Video Game Sales with Ratings. Kaggle link can be found in this ***link***. The drive link for the dataset along with the pickle file for the model can be found in this ***link***.

We have **16719** data points with **16** characteristics available for study:

1) Name - String
2) Platform - String
3) Year of Release - Float
4) Genre - String
5) Publisher - String
6) NA Sales - Float
7) EU Sales - Float
8) JP Sales - Float
9) Other Sales - Float
10) Global Sales - Float
11) Critic Score - Float
12) Critic Count - Float
13) User Score - Object
14) User Count - Float
15) Developer - String
16) Age Rating - String

If we analyze the data type of each feature, we can see that every feature has the expected data type except User Score. When we perform feature wise analysis we shall figure out a method to deal with this problem.

II. Exploratory Data Analysis and Visualization

A. Metacritic Score

Now, if we find the number of null values for the Critic Score feature, we find that **8582** are missing from the data set. This is a roughly half the size of the data set.
Since working with the data points which do not have this feature will be difficult, we decided to drop all the data points that do not have values for this feature. This reduces the size of the data set to **8137** data points, which is a reasonable number to work with.
Note: The large number of missing values is solely because the data set contains a large number of old games that were released before Metacritic started awarding scores formally, resulting in many old games not receiving a Metacritic score.

B. Null values

Let us calculate the number of null values for each feature:

1) Name : 0
2) Platform : 0
3) Year of Release : 154
4) Genre : 0
5) Publisher : 4
6) NA Sales : 0
7) EU Sales : 0
8) JP Sales : 0
9) Other Sales : 0
10) Global Sales : 0
11) Critic Score : 0
12) Critic Count : 0
13) User Score : 38
14) User Count : 1120
15) Developer : 6
16) Rating : 83

We can see that none of the columns seems to have a lot of their values missing. Therefore, we don't need to remove anything else from the data.

C. Strategy for null value replacement

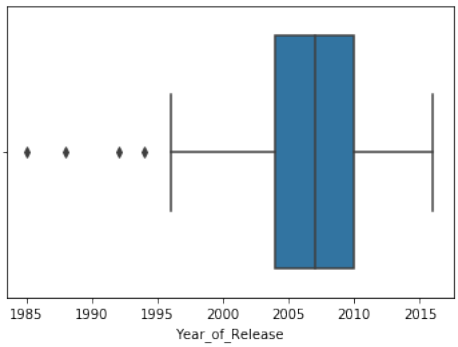
When it comes to replacing null values, we came up with two strategies.

- We can replace the value with the central tendency values of that feature (such as mean, median or mode). This strategy seems to work well for numeric data, especially with less number of null values. However, it cannot perform well for string based data as string based data is categorical, not numeric.
- The second strategy is to create a probability distribution based on the values in that feature and perform

a Inverse Cumulative Distribution function sampling on that distribution to fill null values. This strategy seems to work well for categorical data but introduces a little uncertainty within the data which can prevent production of consistent results.

D. Column wise Analysis

1) Year of Release: The year of release might have some importance when it comes to Critic Score. Let us see the distribution of this feature.



We can see through the box plot that most of the games in this data set have been released from around 2004 to 2010. In real life, the number of games being made have only been increasing from 2004, however, as technology developed, it started to take longer and longer to make video games, thus resulting in a small decline in the number of big budget games being made.

2) Platform: The Platform feature has only 17 values which are 3DS , DC , DS , GBA , GC , PC , PS , PS2 , PS3 , PS4 , PSP , PSV , Wii , WiiU , X360 , XB , XOne. All of which are some of the most popular platforms to play games as of 2016.
Let us print the average Metacritic scores for each platform as well as the number of games released on that platform. (TABLE 1)

Platform	Critic Score	No. Games
3DS	67.101190	168
DC	87.357143	14
DS	63.761506	717
GBA	67.372146	438
GC	69.488839	448
PC	75.928671	715
PS	71.515000	200
PS2	68.727273	1298
PS3	70.382927	820
PS4	72.091270	252
PSP	67.424242	462
PSV	70.791667	120
Wii	62.823932	585
WiiU	70.733333	90
X360	68.616812	916
XB	69.859310	725
XOne	73.325444	169

TABLE I
GROUPING BY PLATFORM

We can see that platforms made by certain companies seem to have a higher score than platforms made by others. Also we can see that majority of the games in this data set were made for Play Station 2.
We also see that the data doesn't differentiate between home consoles and handheld devices. It could be interesting to check the differences in scores for games made for handheld devices and games made for consoles. (This is mainly because the games themselves are fundamentally made in a different manner when it comes to cost, design philosophy etc.) We shall perform some feature engineering to get this aspect into the limelight.

3) Genre: The Genre feature seems to have 11 values which are Action, Adventure, Fighting, Misc, Platform, Puzzle, Racing, Role-Playing, Shooter, Simulation, Sports and Strategy. Let us create a table similar to TABLE 1 for Genre as well. We can see that certain genres seem to get higher Metacritic scores compared to other genres. (This can be due to various factors such as the popularity of the genre etc.)
Also, a majority of the games seem to be Action and Sports, which is also an understandable trend.

Genre	Critic Score	No. Games
Action	66.629101	1890
Adventure	65.331269	323
Fighting	69.217604	409
Misc	66.619503	523
Platform	68.058350	497
Puzzle	67.424107	224
Racing	67.963612	742
Role-Playing	72.652646	737
Shooter	70.181144	944
Simulation	68.619318	352
Sports	71.968174	1194
Strategy	72.086093	302

TABLE II
GROUPING BY GENRE

4) Publisher: There seem to be 304 different publishers in the data set. Now, this is too big a domain to perform any kind of analysis. (As a future exercise one could perform some NLP to categorize/structure the domain to group related publishers together)

5) Developer: There are 1467 different developers in the data set. Similar to Publishers, this domain set is too big to perform any kind of analysis. (As a future exercise, one could cross reference the developers with the genre of games they develop and see how the scores vary as well)

6) Rating: There are only 7 different Rating given to games which are E, E10+, T, M, A0, K-A and RP.
Let us create a table similar to TABLE 1 for Rating We can see that for almost all the data has the ESRB (Entertainment Software Rating Board) rating for the games. Assuming we train a model on this data, we could predict significantly

Rating	Critic Score	No. Games
AO	93.000000	1
E	68.484687	2808
E10+	66.759392	1118
K-A	92.000000	1
M	71.797033	1483
RP	62.000000	3
T	68.828409	2640

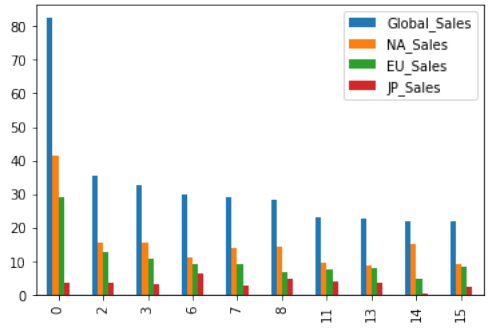
TABLE III
GROUPING BY RATING

incorrect values if a non-ESRB organization rating is used.

7) User Score: User score is supposed to be a quantification of user satisfaction with the game. It ranges from 0-100 and is an amalgamation of all the scores given by all the users in Metacritic.
User Score is supposed to be numeric data but has been provided as string data. This is because of the value 'tbd' which expands to 'To Be Decided'. For these games, the user score hasn't been calculated.

8) User Count and Critic Count: These numbers represent the number of users and critics who provided scores for the game. Generally there is higher participation from both users and critics for games that are successful. However, it is also high for games which are controversial as well, making it a somewhat fickle indicator.

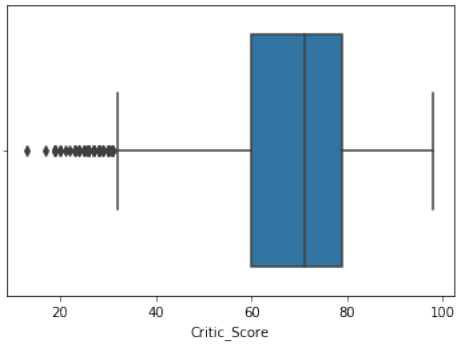
9) Sales features: The sales figures themselves need no explanation. They represent the sales in each region for the game as well as global sales figures.

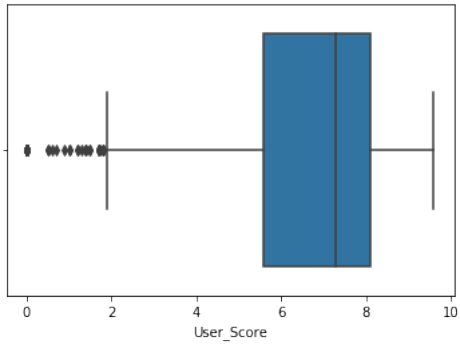


The above graph gives us a comparison of sales between different regions for a random set of games in the data set. We are unable to gleam much information from this graph and will require further analysis for more understanding in the future.

E. Cross Feature Analysis

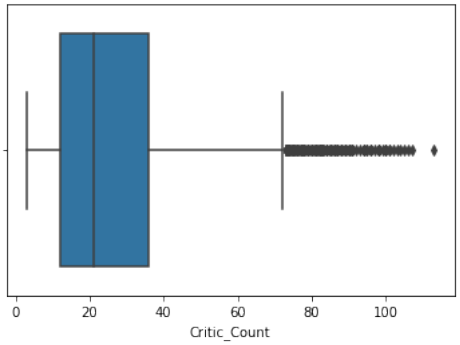
1) Correlation across User and Critic Scores: Let us analyze a box plot for User Score after correction and compare it with Critic Score.

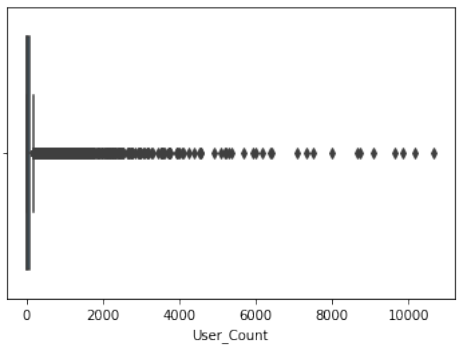




We can see that user scores are a little more generous that critic scores. This can be attuned to human behaviour where only people who buy the game have more inclination to score the game and due to having spent money, will rate the game higher than the norm.

2) Correlation across User and Critic counts: Let us analyze a box plot for User Count after correction and compare it with Critic Count.



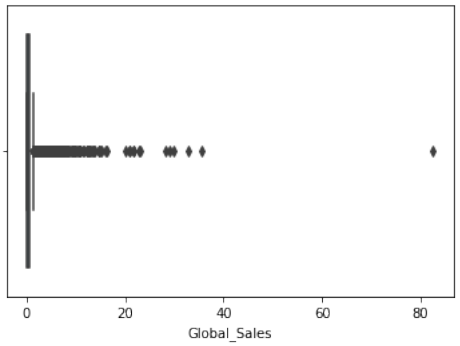


We can see that there are a lot more users rating than

critics rating a game, which makes sense. This however can introduce a lot of noise in the data for user scores.

3) Observations:

1) It would be wrong to assume that a User Score and a Critic Score would correlate in general, however it is a good indicator.
2) The sales in different regions are not directly correlated as different regions might have different preferences for genre, publisher, platform and developer.
3) Let us draw a box plot for Global sales.



We can see that a particular game 'Wii Sports' has an extremely high sales number.This game is a special case as it was given for free to all customers who bought the Wii console (which is one of the most popular consoles of all time). It is an outlier.

F. Feature Engineering and Encoding

1) Year of Release: To help us solve our problem statement, for this feature we will create partitions in this feature which represent big turning points in gaming technology and the industry as well as general gamer expectations.
We will be creating doing a one hot encoding which results in 4 features where each frame represents a range:

- 1985 - 1995
- 1996 - 2005
- 2006 - 2010
- 2010 - beyond

This will help us better understand games released in certain generations.

2) Platform: We shall engineer two different kinds of features for platforms. One is Device Type, the other is Parent Company.
Device Type: We can classify platforms based on whether they are handheld, home console or PC(Personal Computer). This could play a significant role in predicting sales as well as user scores.
Parent Company: We can classify platforms based on whether they are from Sony, Microsoft, Nintendo or Others. This could play a significant role in determining sales in different regions as different companies are popular in different regions. All these features will be one hot encoded on the data. This is because the overhead associated with it is comparatively less.

3) Genre: Genre is a feature that has 12 unique values in its domain. While One Hot Encoding for this feature is very amenable, 12 new features will be added to the data set. It is necessary to check if we can handle such a data set or not. In our case, we didn't find any significant increase in computation time.

4) Publisher: Since we have too many publishers, we shall perform label encoding on this feature.

5) Developer: Since we have too many developers, we shall perform label encoding on this feature.

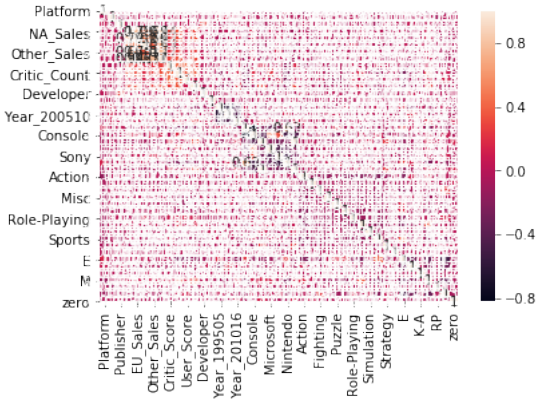
6) Rating: Rating seems to be a feature with only 7 values in the domain. This makes it safe for one hot encoding.

7) User Score: This feature is present as a string in the data set. We shall convert all the scores to numeric while assuming 'tbd' scores to be 0. We shall also replace the missing values with 0.

8) Other Numerical features - Global Sales etc.: We shall simply replace all the null values with the mean of that column.

G. Heat Map

Provided below is a heat map of all the features after feature engineering and encoding.



III. Model Training and Error Evaluation

A. Models

1) Linear Regression: Linear Regression is one of the most extensively used methods in data science. It can be used to fit a predictive model to an observed set of values and explanatory variables. It can also be used to explain the variation of a response variable with some of its attributes. Typically, these models are fitted using the last squares approach. However, these models can tend to over-fit and

don't generalise well as the complexity increases.

2) Ridge Regularisation: Ridge regularisation is linear regression with another factor, a scalar multiple of the sum of the squares of the weights the model is trying to predict. As a result, it penalises models which over-fit as those models tend to have high weights.

3) Lasso Regularisation: Lasso regression is linear regression with another factor, a scalar multiple of the sum of the magnitudes of the weights of the coefficients. It's similar to Ridge regression, but not quite. Lasso regularisation forces some attributes to become zero, even for small values of the scalar multiple . As a result, it is also used as a feature selector. Since we have very less features, it's not very handy to delete any of them. This can be seen with the RMSE values. Lasso tends to get outperformed by both simple Linear Regression and Ridge Regularisation.

Ridge regularisation performs better than simple Linear Regression as it penalises over-fitting models.

B. Model Performance and Evaluation

For model selection, we have compared performance across 3 different models

1) Linear Regression with feature engineering
2) Ridge with feature engineering
3) Lasso with feature engineering

We shall be calculating both RMSE (Root Mean Square Error) as well as the r2 score for each scenario. The results are provided below.

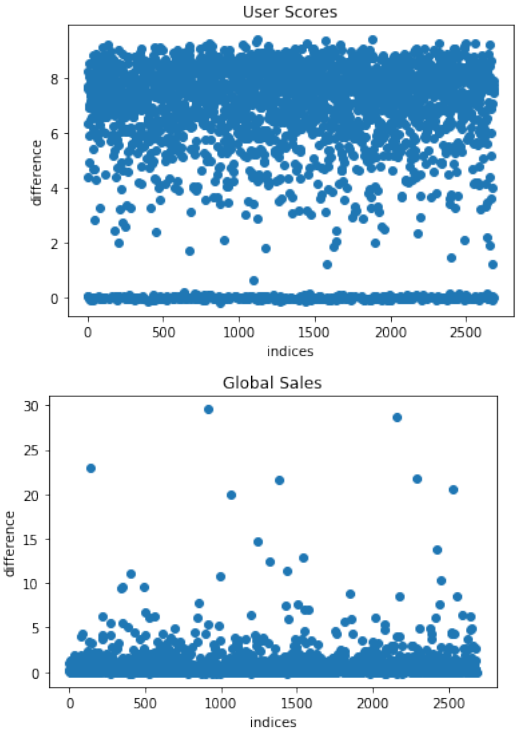
Model	User Score	Global Sales	NA Sales	EU Sales	JP Sales
LR	2.3628	1.6160	0.8125	0.5198	0.2786
Lasso	2.6136	1.7899	0.8933	0.5761	0.2866
Ridge	2.3151	1.5457	0.7321	0.5421	0.2782

TABLE IV
RMSE ERROR

Model	User Score	Global Sales	NA Sales	EU Sales	JP Sales
LR	0.2559	0.1840	0.1714	0.1858	0.0545
Lasso	0.0814	-0.0012	0.0008	-0.0007	-0.0013
Ridge	0.3311	0.1919	0.1879	0.1649	0.0972

TABLE V
R2 ERROR

C. Residual Plots for Ridge Regression



IV. CONCLUSION

We have predicted User Scores, Global Sales, NA Sales, EU Sales and JP Sales using 3 different models (Linear Regression, Lasso, Ridge) with high accuracy.