Report on Game Sales Project

Till Neuber

1/5/2020

Overview

The goal of this project is to build a prediction system for video game sales based on sales data from VGChartz and ratings form Metacritic. The dataset has been created by Rush Kirubi (see https://www.kaggle.com/rush4ratio/video-game-sales-with-ratings). The project and dataset can also be found at: https://github.com/TillNeuber/HarvardX-Data-Science.

The raw dataset consists of 16,719 video games. Besides some general descriptive information (platform, genre, developer, publisher, age rating) the dataset contains sales information for each title. Additionally, aggregated rating information (user & critic score) and the number of users and critics who assessed the game are provided. Unfortunately, the rating information is relatively sparse, after cleaning the dataset (described in more detail in the following section) 7,017 games with sufficient information for the subsequent analysis remain. After that, the dataset is prepared in a way that is suitable for fitting a regression model. The data preparation is outlined in the following section.

Initially, several univariate regressions were performed to identify potentially relevant explanatory variables. I find that particularly the aggregated critics' score, the number of critics who assessed the game, the publisher and the platform have high predictive power. Combining these variables in a multivariate regression further improves the quality of the forecast.

Overall, I was able to improve the RMSE from around 1.20 (naive forecast using the mean sales number) to 1.11 - 1.19 (univariate regressions) and finally to 1.03 (multivariate regression).

Analysis

Data Preparation

Following the import of the dataset, it needs to be cleaned and prepared for the subsequent analysis. This consists of 4 steps:

1. Removing games for which the rating information are incomplete. Note that this step has to be performed with some caution as it has the potential to introduce some bias in the analysis e.g. if there's a correlation between sales and missing data. I leave it for future work to further explore the link between missing data and sales and potentially use other methods to account for missing data (e.g. imputation).

```r
sales <- sales[complete.cases(sales), ]
```

2. The user scores have been encoded as factor, presumambly because some scores have been reported as "tbd". However, step 1 also removed all games with "tbd" user scores and I can transform the data to numerical values in order to be able to later perform a regression.

```r
sales$User_Score <- as.numeric(as.character(sales$User_Score))
```

3. The dataset contains many publishers and developers that are only responsible for very few games. This leads to two problems: (i) the regression needs to determine many coefficients which slows down the fitting (the dataset is not so large that performance is a real concern for this model however) and (ii) the validation set will most likely contain some publishers and developers for which no coefficients have been fitted as they were not present in the training set. This causes the predict function to throw an error. I overcome both problems by assigning all small publishers and developers to a residual category "Other".

```
sales <- sales %>% group_by(Publisher) %>% mutate(n_published = n())
levels(sales$Publisher) <- union(levels(sales$Publisher), "Other")
sales$Publisher[sales$n_published <= 5] = "Other"
sales <- ungroup(sales)

sales <- sales %>% group_by(Developer) %>% mutate(n_developed = n())
levels(sales$Developer) <- union(levels(sales$Developer), "Other")
sales$Developer[sales$n_developed <= 5] = "Other"
sales <- ungroup(sales)
```

4. To make the dataset easier to handle I drop all the columns I will not use in the subsequent analysis. In particular, these are the regional sales columns.

```
sales <- subset(sales, select = -c(NA_Sales, EU_Sales, JP_Sales, Other_Sales, n_published, n_developed))
```

After the data has been prepared, I split the set in a training and a validation subset.

```
#set.seed(1, sample.kind="Rounding")
# if using R 3.5 or earlier, use `set.seed(1)` instead
set.seed(1)

test_index <- createDataPartition(y = sales$Global_Sales, times = 1, p = 0.1, list = FALSE)
train_set <- sales[-test_index,]
test_set <- sales[test_index,]
```

Naive Prediction

To be able to assess the model performance subsequently, I establish a benchmark by making a naive forecast (i.e. use the mean sales as prediction) and calculate the RMSE. Later, I will compare the RMSE of the regression models with this value.

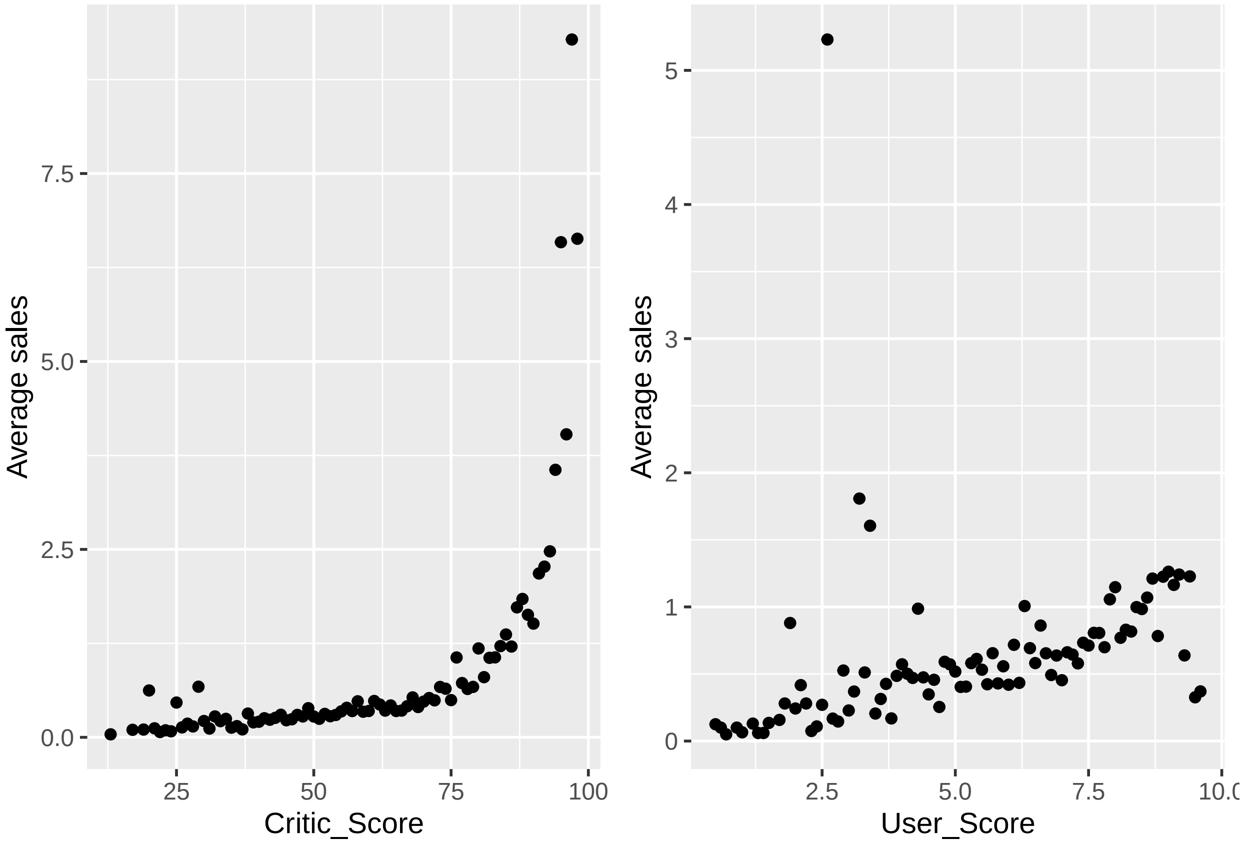
```
# Calculate the mean sales per video game, i.e. a naive prediction
m <- mean(train_set$Global_Sales)

# Calculate the RMSE for the naive prediction using the mean sales as a benchmark
rmse_naive <- RMSE(test_set$Global_Sales, m)
```

The dataset offers several variables that could potentially be used to explain the sales. In order to gather some tentative evidence which variables might be suited best, I begin with a visual exploration of the dataset.

Scores

First, I look at the relationship between ratings (by users & critics) and sales. The following plots visualize this relationship. To get a clearer overview, I plot the mean sales of games for a particular score.



The plots seem to indicate a positive monotonous relationship between scores and sales. In particular, the plot seems to hint at a exponential relationship between critics' score and the sales number and a linear relationship between users' score and sales. I fitted a linear regression model for both variables and additionally an exponential model for the critics' score. However, as it can also be seen by the plot, the predictive power of the users' score is rather low and offers no substantial improvement over the naive prediction. The critics' score on the other hand reduces the RMSE in the linear model. The exponential model performs significantly worse. Refer to the next section for an overview of all model performances.

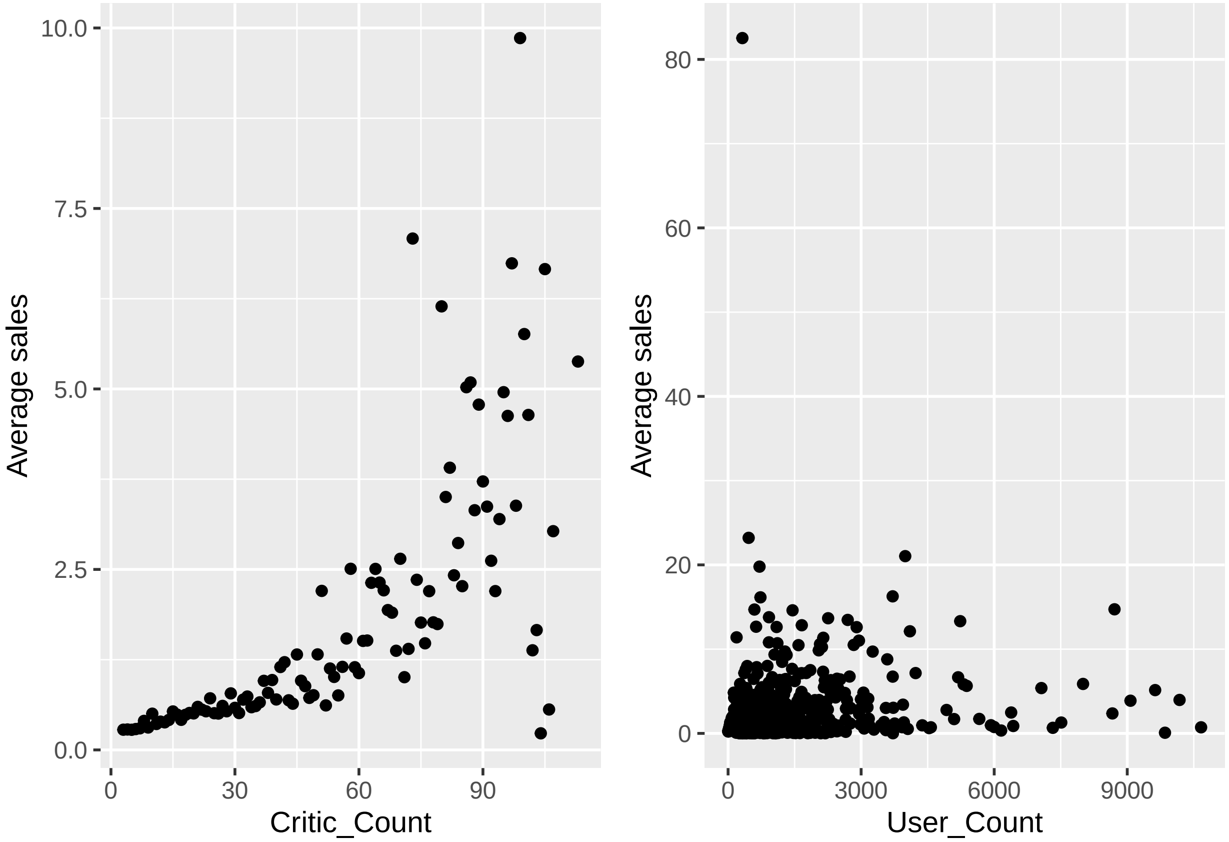
```r
# Fit a linear regression using the critic scores only
fit_critic_score <- lm(Global_Sales ~ Critic_Score, data = train_set)
y_hat_critic_score <- predict(fit_critic_score, test_set)
rmse_critic_score <- RMSE(test_set$Global_Sales, y_hat_critic_score)

# Fit a exponential model using the critic scores only
# (i.e. transform Global_Sales and perform the regression)
fit_critic_score_exp <- lm(log(Global_Sales) ~ Critic_Score, data = train_set)
y_hat_critic_score_exp <- predict(fit_critic_score_exp, test_set)
rmse_critic_score_exp <- RMSE(test_set$Global_Sales, y_hat_critic_score_exp)

# Fit a linear regression using the user scores only
# Note that this step required cleaning the data
# (dropping rows that contain no or "tbd" values and convert the result to numeric values)
fit_user_score <- lm(Global_Sales ~ User_Score, data = train_set)
y_hat_user_score <- predict(fit_user_score, test_set)
rmse_user_score <- RMSE(test_set$Global_Sales, y_hat_user_score)
```

User & Critic Count

The next relationship I analyzed is that between the number of users respectively critics and sales. The following plots illustrate that relationship. Again, I plot the average sales number for a given number of critics / users to gain a better overview.



There seems to be a positive relationship between the number of critics who rated a game and sales. The fitted regression model also shows a clear improvement in RMSE compared to the naive model.

The number of users who rated the game does not seem to predict sales very well. In fact, the regression model based on this variable also does not improve RMSE.

```r
# Fit a linear regression using number of critics
fit_critic_count <- lm(Global_Sales ~ Critic_Count, data = train_set)
y_hat_critic_count <- predict(fit_critic_count, test_set)
rmse_critic_count <- RMSE(test_set$Global_Sales, y_hat_critic_count)

# Fit a linear regression using number of users
fit_user_count <- lm(Global_Sales ~ User_Count, data = train_set)
y_hat_user_count <- predict(fit_user_count, test_set)
rmse_user_count <- RMSE(test_set$Global_Sales, y_hat_user_count)
```

Platform, Developer and Publisher

It seems intuitive that there are differences in sales depending on platform, developer and publisher. The first one determines the number of potential buyers, the latter two are responsible for the quality of the game and its marketing. Two univariate regression with platform respectively publisher as explanatory variables does indeed show a significant improvement in RMSE compared to the benchmark. However, a bit surprisingly the same regression model using the developer as explanatory variable does not show any improvements. It is left for future work to further explore the relationship between developer and sales and potentially find a better suited model to fit that improves the forecast. Note that as described earlier, I introduced another category for publishers and developers - namely "Others" for small publishers / developers - primarily to avoid the issue of having publishers / developers in the validation set that were not present in the training data and for which no coefficients have been fitted. The code below shows the performed regressions:

```r
# Fit a linear regression using the platform only
fit_platform <- lm(Global_Sales ~ Platform, data = train_set)
y_hat_platform <- predict(fit_platform, test_set)
rmse_platform <- RMSE(test_set$Global_Sales, y_hat_platform)

# Fit a linear regression using the publisher only
fit_publisher <- lm(Global_Sales ~ Publisher, data = train_set)
y_hat_publisher <- predict(fit_publisher, test_set)
rmse_publisher <- RMSE(test_set$Global_Sales, y_hat_publisher)

# Fit a linear regression using the developer only
fit_developer <- lm(Global_Sales ~ Developer, data = train_set)
y_hat_developer <- predict(fit_developer, test_set)
rmse_developer <- RMSE(test_set$Global_Sales, y_hat_developer)
```

Final Model

Finally, I combine the variables that have shown to have some explanatory power in one model. These seem to be the number of critics and their score, the publisher and the platform. The model is fitted as follows:

```r
# Fit a multivariate linear regression combining the variables
# that seem to have the most explanatory power:
# critic score, critic count, publisher & platform
fit_mult<- lm(Global_Sales ~ Critic_Score + Critic_Count + Publisher + Platform, data = train_set)
y_hat_mult <- predict(fit_mult, test_set)
rmse_mult <- RMSE(test_set$Global_Sales, y_hat_mult)
```

Results

This section looks at the model performances, i.e. the RMSE of the models outlined in the previous section. The following table shows the RMSE for different modelling approaches:

Model	RMSE
Naive Prediction	1.2029249
Critic score (linear)	1.1474166
Critic score (exponential)	2.2425901
User score	1.2030748
Critic count	1.1377165
User count	1.1994642
Platform	1.1917207
Publisher	1.1057214
Developer	1.2125199
Critic score + critic count + publisher + platform	1.0344809

The variables that seem to have the most explanatory power are Critic Score, Critic Count, Publisher and Platform. Combining these variables together in a multivariate model reduces the RMSE even further. Overall, I was able to significantly reduce the RMSE compared to the benchmark - i.e. the simple forecasting model using the mean - from 1.2029249 to 1.0344809 using the multivariate model.

Conclusion

This report discussed several possible methods to forecast video game sales based on various variables. I outlined several prepatory steps that were required to perform the subsequent analysis on the data. Subsequently various potential explanatory variables were further discussed and corresponding regressions models were fitted. It turned out that particularly the aggregate score of critics, the number of critics who rated a game, the publisher and the platform are useful in making a sales prediction. The final model significantly reduced the RMSE compared to a naive prediction using the mean sales value.

Besides theoretical limitations with regards to the chosen model - more advanced models going beyond linear regression models discussed here might produce even better results - and the availability of data, I do not want to leave unmentioned some limitations concerning the practical applicability of the approach. Although the report clearly shows e.g. that game developers should be thoughtful when deciding for which platform to develop a game or about collaborations with publishers, the use of variables such as the aggregated critics' score and number should be viewed with some caution from a practical point of view. Both variables are only known ex post to game development, so a model using them is not suited e.g. to assist in decision making before or during game development. However, there is still a limited number of use cases for a model such as this, e.g. as part of a sales forecast model for financial analysts.

There are several possible improvements that can be explored by future work. In particular the regional sales could be incorporated to forecast regional sales first instead of directly trying to predict global sales. A natural starting point would be to explore the popularity of certain genres and platforms in certain regions (e.g. considering the populartiy of PSP consoles in Japan) and to subsequently fit forecasting models for each region.