Predicting Global Sales of Video Games Using a Recommendation System

Lizz Judge

6/16/2019

Introduction

The objective of this study is to develop a data-based model to predict global sales of video games using a recommendation system. Project data are from https://www.kaggle.com/rush4ratio/video-game-sales-with-ratings. The data set includes video game name, platform, year of release, genre, publisher, sales figures for different regions, critic score and count, user score and count, developer, and rating. Goodness of the model is measured with a loss function.

Methods

The loss function for the exercise is the root mean square error (RMSE), defined as:

$$\mathrm{RMSE} = \sqrt{\frac{1}{N} \sum_{effect} (\hat{y} - y)^2},$$

where $\hat{y}$ is the predicted rating, y is the actual rating, and $effect$ is the feature used to make the prediction (title, platform, critic score, etc.).

The most naive, or zeroth order, data-based approach to predicting global sales is to use the average of all the records):

$$\hat{y}_n = \mathrm{mean}(Global\ Sales) = \mu.$$

The value of the loss function for the Naive Model is

$$\mathrm{RMSE}_n = 1.9402.$$

Degree of improvement upon this loss function is the metric of success for the current analysis.

The naive approach does not take into account that some titles are more popular than others or that some plaforms might have larger market share. For example, titles with broader mass market appeal will sell more across all platforms, bringing the over all sales figures up for that title. A model that does not take this into account will have artifically low predictions for popular games and artificially high predictions for niche games. The next order corrections to the model include these effects:

$$\hat{y} = \mu + b_{name} + b_{platform} + b_{genre} + b_{publisher} + b_{developer} + b_{rating},$$

where b_{name}, $b_{platform}$, b_{genre}, $b_{publisher}$, $b_{developer}$, and b_{rating} are the correction terms for name, platform, genre, publisher, developer, and rating, respectively. These features were included in the model because they are factors and are therefore appropriate for this type of analsys. User scores are also factors in the data table, but in reality, they are continuous, numeric data. The impact of critic and user scores could be analyzed used a different type of modeling, such as linear regression.

We first ensure that all features are present for each title in the data set by removing incomplete rows. The data set is fairly small, and is significantly reduced after incomplete rows are removed. As variations from the expected value can be large with small data sets, 10-fold cross validation is use with a 90/10 training/test split.

For each resampling split, we ensure that the data used to validate the model contains identical feature levels as the data used to train the model. That is, platforms, etc., that are present in the training data must be present in the test data. From here, calculation of the *b*s was straightforward. Examination of the individual correction terms provided a path forward for which terms to include in a complete models.

A total of seven models are compared:

number	method
	Naive Model
1	Name Effect Model
2	Platform Effect Model
3	Genre Effect Model
4	Publisher Effect Model
5	Developer Effect Model
6	Rating Effect Model
7	Combined Effects Model

Example code to calculate correction from just one effect (the platform effect):

```r
rmses_platforms <- matrix(NA,nrow=1,ncol=10)
for( i in seq(1:10)){
  train_set <- video_games[-test_index[,i],] # 90% of data
  test_set  <- video_games[test_index[,i],]
  test_set <- test_set %>%
    semi_join(train_set, by = "Platform")

  platform_avgs <- train_set %>%
    group_by(Platform) %>%
    summarize(b_p = mean(Global_Sales - mu))

  # predict rating for each entry: add mu and b_p
  predicted_sales_p <- mu + test_set %>%
    left_join(platform_avgs, by='Platform') %>%
    .$b_p

  rmses_platforms[i] <- RMSE(predicted_sales_p, test_set$Global_Sales)
}

model_1_rmse <- mean(rmses_platforms)
```

The other effect corrections were calculated similarly.

Code to calculate corrections for all effects:

```r
rmses <- matrix(NA,nrow=1,ncol=10)
for( i in seq(1:10)){
  train_set <- video_games[-test_index[,i],] # 90% of data
  test_set  <- video_games[test_index[,i],]
  test_set <- test_set %>%
    semi_join(train_set, by = "Name") %>%
    semi_join(train_set, by = "Platform") %>%
    semi_join(train_set, by = "Genre") %>%
    semi_join(train_set, by = "Publisher") %>%
    semi_join(train_set, by = "Developer")%>%
    semi_join(train_set, by = "Rating")

  Name_avgs <- train_set %>%
    group_by(Name) %>%
    summarize(b_n = mean(Global_Sales - mu))

  Platform_avgs <- train_set %>%
    left_join(Name_avgs, by='Name') %>%
    group_by(Platform) %>%
    summarize(b_pl = mean(Global_Sales - mu - b_n))

  Genre_avgs <- train_set %>%
    left_join(Name_avgs, by='Name') %>%
    left_join(Platform_avgs, by='Platform') %>%
    group_by(Genre) %>%
    summarize(b_g = mean(Global_Sales - mu - b_n - b_pl))

  Publisher_avgs <- train_set %>%
    left_join(Name_avgs, by='Name') %>%
    left_join(Platform_avgs, by='Platform') %>%
    left_join(Genre_avgs, by='Genre') %>%
    group_by(Publisher) %>%
    summarize(b_pu = mean(Global_Sales - mu - b_n - b_pl - b_g))

  Developer_avgs <- train_set %>%
    left_join(Name_avgs, by='Name') %>%
    left_join(Platform_avgs, by='Platform') %>%
    left_join(Genre_avgs, by='Genre') %>%
    left_join(Publisher_avgs, by='Publisher') %>%
    group_by(Developer) %>%
    summarize(b_d = mean(Global_Sales - mu - b_n - b_pl - b_g - b_pu))

  Rating_avgs <- train_set %>%
    left_join(Name_avgs, by='Name') %>%
    left_join(Platform_avgs, by='Platform') %>%
    left_join(Genre_avgs, by='Genre') %>%
    left_join(Publisher_avgs, by='Publisher') %>%
    left_join(Developer_avgs, by='Developer') %>%
    group_by(Rating) %>%
    summarize(b_r = mean(Global_Sales - mu - b_n - b_pl - b_g - b_pu - b_d))

  # predict rating for each entry: add mu and bs
  predicted_sales <- test_set %>%
    left_join(Name_avgs, by='Name') %>%
    left_join(Platform_avgs, by='Platform') %>%
    left_join(Genre_avgs, by='Genre') %>%
    left_join(Publisher_avgs, by='Publisher') %>%
      left_join(Developer_avgs, by="Developer") %>%
      left_join(Rating_avgs, by="Rating") %>%
      mutate(pred = mu + b_n + b_pl + b_g + b_pu + b_d + b_r) %>%
      .$pred

    rmses[i] <- RMSE(predicted_sales, test_set$Global_Sales)
  }

  model_7_rmse <- mean(rmses)
```

Results

The RMSEs for the individual effects are:

method	RMSE
Naive Model	1.940
1. Name Effect Model	1.296
2. Platform Effect Model	1.756
3. Genre Effect Model	1.773
4. Publisher Effect Model	1.704
5. Developer Effect Model	1.581
6. Rating Effect Model	1.768

None of these are impressive improvements. Ideally, the RMSE should be below 1, and none of these come close. The Name Effect Model shows a decrease of about 32%, which is substantial, but not sufficient.

Greater improvement is expected to come from combining effects. The result of combining all effects is:

method	RMSE
7. Combined Effects Model	1.203

Combining all effects results in about a 38% decrease in RMSE. Again, this is substantial, but certainly not sufficient to use for business forcasting.

It is possible that a different type of analysis altogether would be appropriate to predict sales. Going back to the other data that are available, one could investigate whether sales could be predicted using critics or users scores. Critics are typically respected members of the community, and high scores from them on a title might drive sales. If so, linear regression would be a viable option for analysis. However, it is clear from Figure 1 that while Global Sales do increase with higher Critic Scores, they do not have the type of bivariate relationship that would justify such an analysis. It is reasonable to assume that the same is true for User Scores.

Conclusion

An attempt was made to predict global sales of video games based on factor data such as game title, platform, genre, publisher, developer, and rating. The best model produced an RMSE of 1.203. The most appropriate conclusion is that global sales cannot be predicted using recommendation systems. See https://github.com/lizzjudge/HarvardX-PH125.9x-Data-Science-Capstone.git

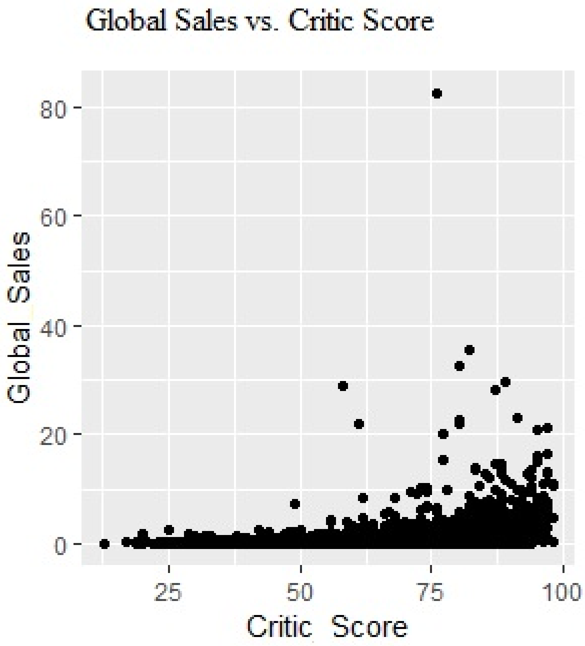

Figure 1: No bivariate relationship is present between global sales and critic scores of video games.