Machine Learning: Videogames Project

Jorge Plaza

30 de abril de 2020

Introduction

This project employs several Machine Learning tecniques to gets results on a dataset based on videogames sales, critic scores and user scores. The main goal of the project its to find relations between different variables in the dataset. This dataset was provided by the user Rush Kirubi on kaggle, so thank to him for give the opportunity to use this data.

The total dataset consist of about 16800 rows that have 16 variables. We have:

- Name: Name of the videogame
- Platform: Console of the game
- Year_Of_Release: Year that the game come out
- Genre: Genre of the game
- Publisher: The company that publish the game
- NA_Sales: Game sales in North America (In millions of units)
- EU_Sales: Game sales in Europe (In millions of units)
- JP_Sales: Game sales in Japan (In millions of units)
- Other_Sales: Game sales in the rest of the world (In millions of units)
- Global_Sales: Total sales of the game (In millions of units)
- Critic_Score: Critic scores based on metacritic page
- Critic_Count: Total of critics scores
- User_Score: User scores based on metacritic
- User_Count: Total of user scores
- Developer: Developer of the game
- Rating: The ESRB rating of the game

Methods and Analysis

Data Cleaning

The dataset its in csv format and have too many NA values because metacritic only covers a subset of the platform. This values come directly by NA format and in "N/A" character. So for cleaning this data we use this command

```
videogames <- read_csv("videogames.csv", na = c("N/A"))
videogames <- na.omit(videogames)
videogames <- as.data.frame(videogames)
```

Data Exploration

So for calculate the main information on the data, we use the function summary.

```
summary(videogames)
```

```
##      Name              Platform         Year_of_Release    Genre
##  Length:6893        Length:6893        Min.   :1985    Length:6893
##  Class :character   Class :character   1st Qu.:2004    Class :character
##  Mode  :character   Mode  :character   Median :2007    Mode  :character
##                                        Mean   :2007
##                                        3rd Qu.:2011
##                                        Max.   :2016
##   Publisher            NA_Sales         EU_Sales          JP_Sales
##  Length:6893        Min.   : 0.000   Min.   : 0.0000   Min.   :0.00000
##  Class :character   1st Qu.: 0.060   1st Qu.: 0.0200   1st Qu.:0.00000
##  Mode  :character   Median : 0.150   Median : 0.0600   Median :0.00000
##                     Mean   : 0.391   Mean   : 0.2345   Mean   :0.06388
##                     3rd Qu.: 0.390   3rd Qu.: 0.2100   3rd Qu.:0.01000
##                     Max.   :41.360   Max.   :28.9600   Max.   :6.50000
##   Other_Sales        Global_Sales      Critic_Score    Critic_Count
##  Min.   : 0.00000   Min.   : 0.0100   Min.   :13.00   Min.   :  3.00
##  1st Qu.: 0.01000   1st Qu.: 0.1100   1st Qu.:62.00   1st Qu.: 14.00
##  Median : 0.02000   Median : 0.2900   Median :72.00   Median : 24.00
##  Mean   : 0.08201   Mean   : 0.7716   Mean   :70.26   Mean   : 28.84
##  3rd Qu.: 0.07000   3rd Qu.: 0.7500   3rd Qu.:80.00   3rd Qu.: 39.00
##  Max.   :10.57000   Max.   :82.5300   Max.   :98.00   Max.   :113.00
##   User_Score          User_Count       Developer            Rating
##  Length:6893        Min.   :    4.0   Length:6893        Length:6893
##  Class :character   1st Qu.:   11.0   Class :character   Class :character
##  Mode  :character   Median :   27.0   Mode  :character   Mode  :character
##                     Mean   :  174.4
##                     3rd Qu.:   89.0
##                     Max.   :10665.0
```

Looking closer in the data, realize that the user score columns it's in character format

```
class(videogames$User_Score)
```

```
## [1] "character"
```

So, we change the columns class to numeric

```
videogames$User_Score <- as.numeric(videogames$User_Score)
```

In other column, especialize in the platform variable. Think that there are too much information that doesn't really mean too much for the analysis we want. So for better approach, we are going to segment this data into the main companies of videogames that are Sony, Nintendo, Microsoft and Sega. So for this, we create the following vectors

```
sony <- c('PS','PS2','PS3','PS4' ,'PSP','PSV')
microsoft<- c('PC','X360','XB','XOne')
nintendo <- c('3DS','DS','GBA','GC','N64','Wii','WiiU')
sega <- c('DC')
```

Then, we create the function to assing the companies by platform and create the new column

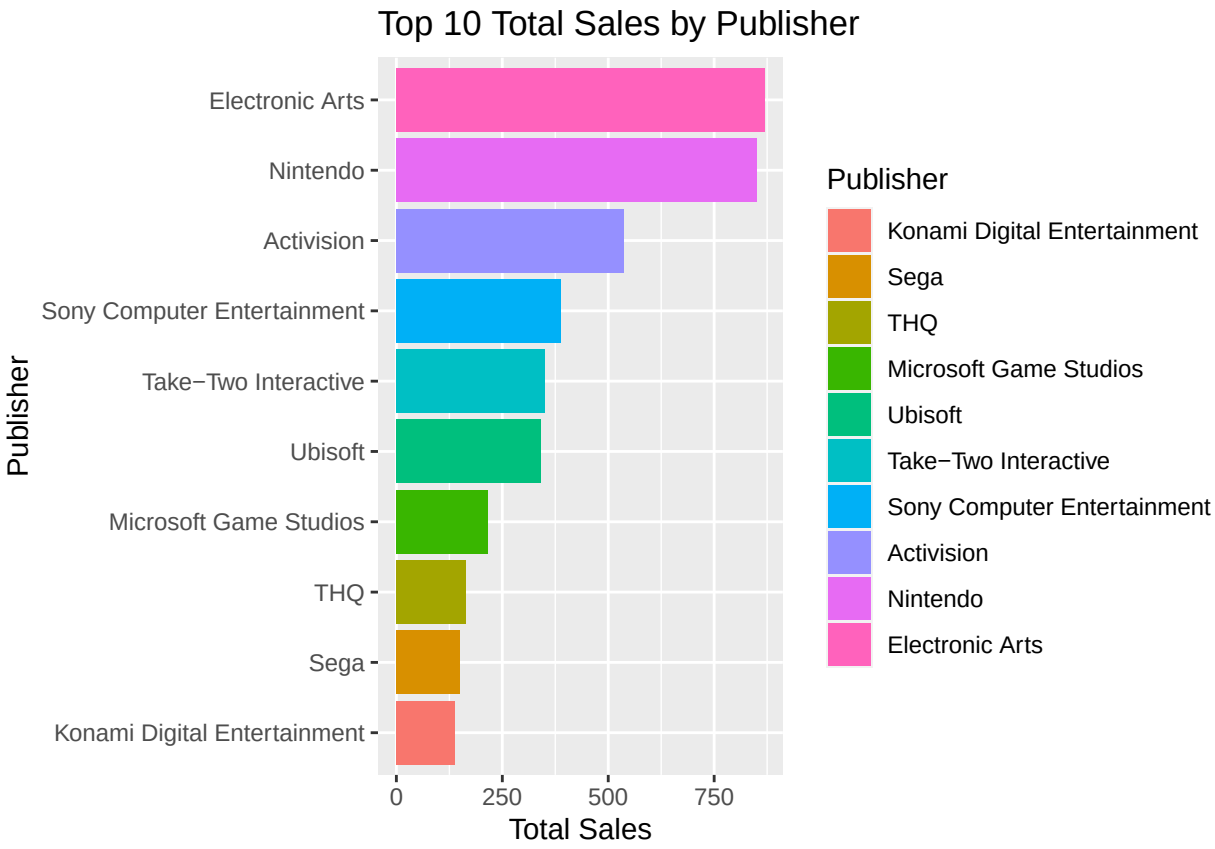
```
changePlatform <-function(x){
  if (x %in% sony == TRUE) {return('Sony')}
  else if(x %in% microsoft == TRUE) {return('Microsoft')}
  else if(x %in% nintendo == TRUE) {return('Nintendo')}
  else if(x %in% sega == TRUE) {return('Sega')}
  else{return('Other')}
}

videogames$company <- sapply(videogames$Platform, changePlatform)
```

Data Visualization

In this section, we're going to see different interesting graph to analyze different aspect of the videogame industry.

The first graph are the top 10 total sales by the most important publisher.

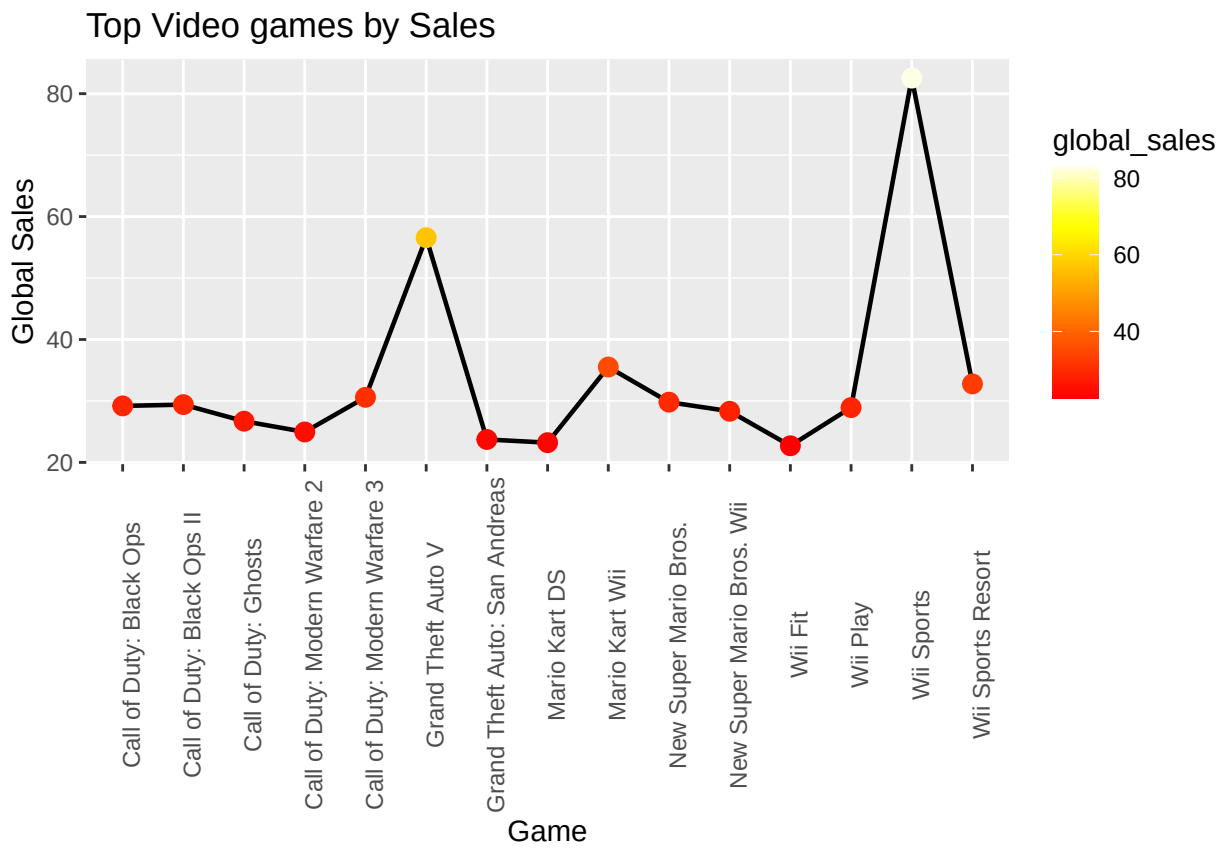
```
top10 %>% ggplot() +
          geom_bar(mapping = aes(x = ttl_sales, y = Publisher, fill = Publisher)
          ,stat = "identity") +
          labs(x = "Total Sales", y = "Publisher") +
          ggtitle("Top 10 Total Sales by Publisher")
```



In this graph, we can see that the most total sales are Electronic Arts, Nintendo and Activision. This first party publisher are known for the high publicity and famous games, like Battlefield series for Electronic Arts, Mario for Nintendo and Call of Duty for Activision.

The following graph, complements the first graph. In this we plot the best 10 games by global sales.

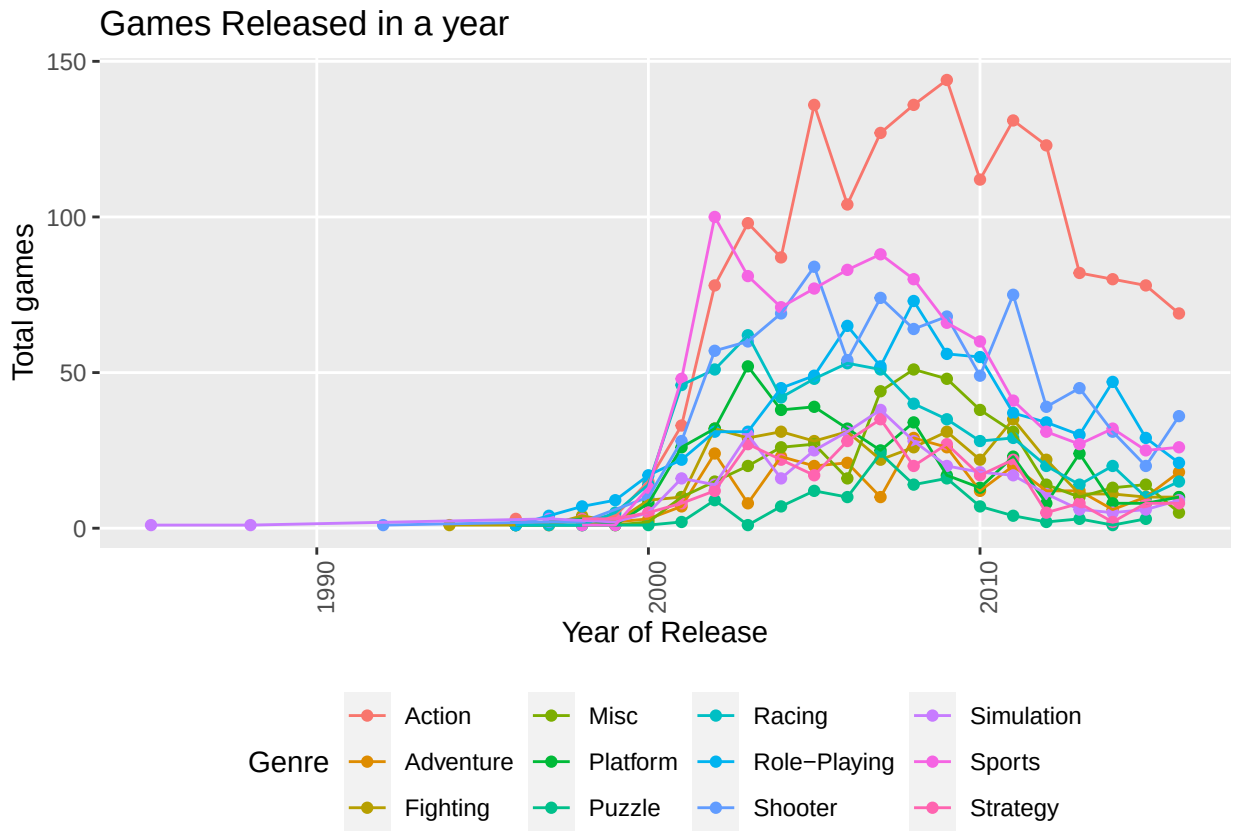
```
videogames %>% group_by(Name) %>%
               summarise(global_sales=sum(Global_Sales)) %>%
               arrange(desc(global_sales)) %>%
               head(15) %>%
               ggplot(aes(x = Name, y = global_sales, group = 1)) +
               geom_line(size=0.8) +
               geom_point(aes(col=global_sales),size=3) +
               scale_color_gradientn(colours = heat.colors(20)) +
               theme(axis.text.x = element_text(angle = 90)) +
               labs(x = "Game", y = "Global Sales", title="Top Video games by Sales")
```



We can see clearly that the most sells games are in mayority Nintendo and Activision games. Where we confirmed that the most selling game in the history of the industry is Wii Sports from Nintendo. We can see two games from Take-Two Interactive, the famous Grand Theft Auto Series. Known as a quality serie.

The next graph are the total games launched by each year. With colors representing the genre of games.

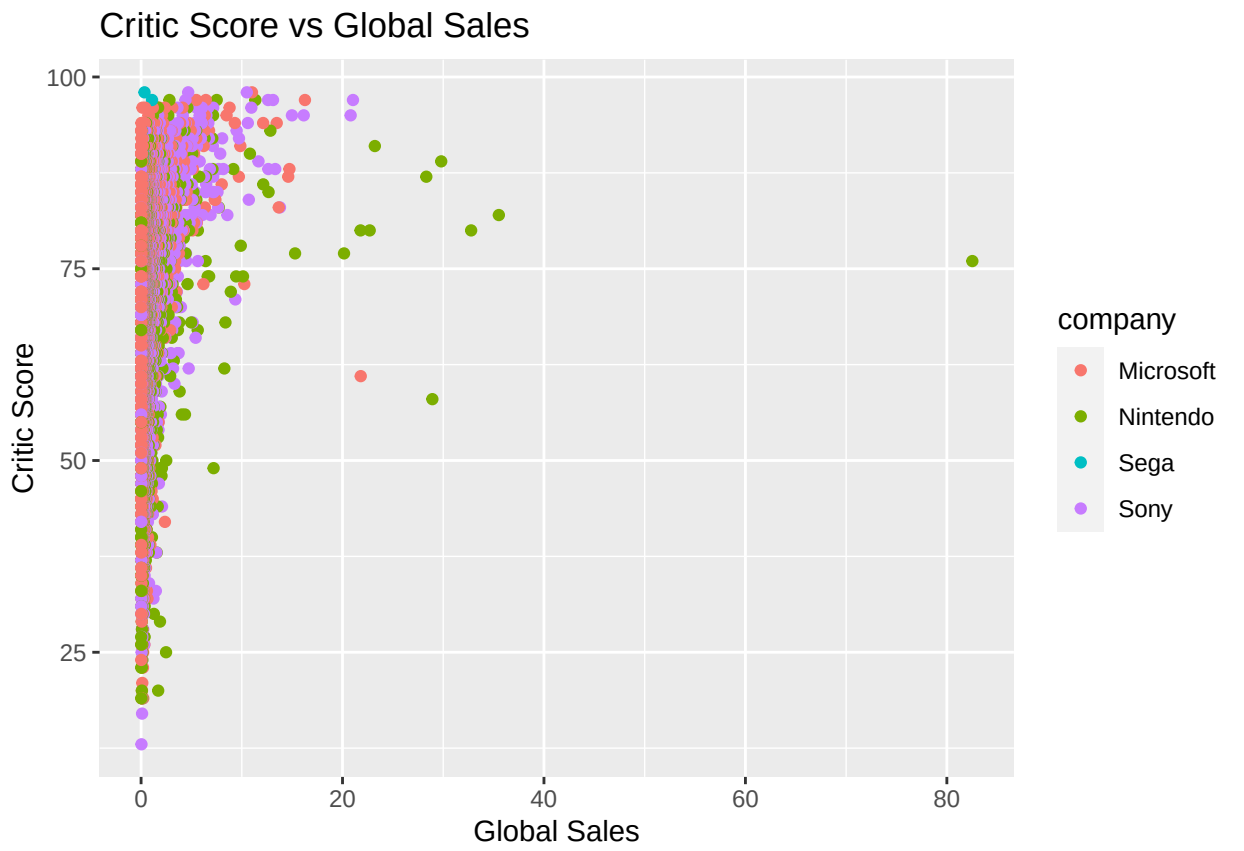
```
videogames %>% group_by(Year_of_Release,Genre)%>%
               summarise(no_of_games = n()) %>%
               ggplot(aes(x = Year_of_Release, y = no_of_games, group = Genre, col = Genre)) +
               geom_point()+
               geom_line() +
               theme(legend.position = "bottom",
                     axis.text.x = element_text(angle = 90),
                     panel.grid.minor = element_blank()) +
               labs(x = "Year of Release", y = "Total games", title = "Games Released in a year")
```



We can see some interesting information. In the start of the industry, by year where a poor quantity of games per year. As the industry grow more games where release. The pick of the graph was in 2009. With a huge number of games released. And because the cost of develop the games after 2009 increases (Better graphics, more expensive) the quantity of games dropped sustancially.

The following graph are the Critic Score (Values 0 - 100) vs the global sales (in million of units) for each game in the dataset. The colors represent the main company in the industry.

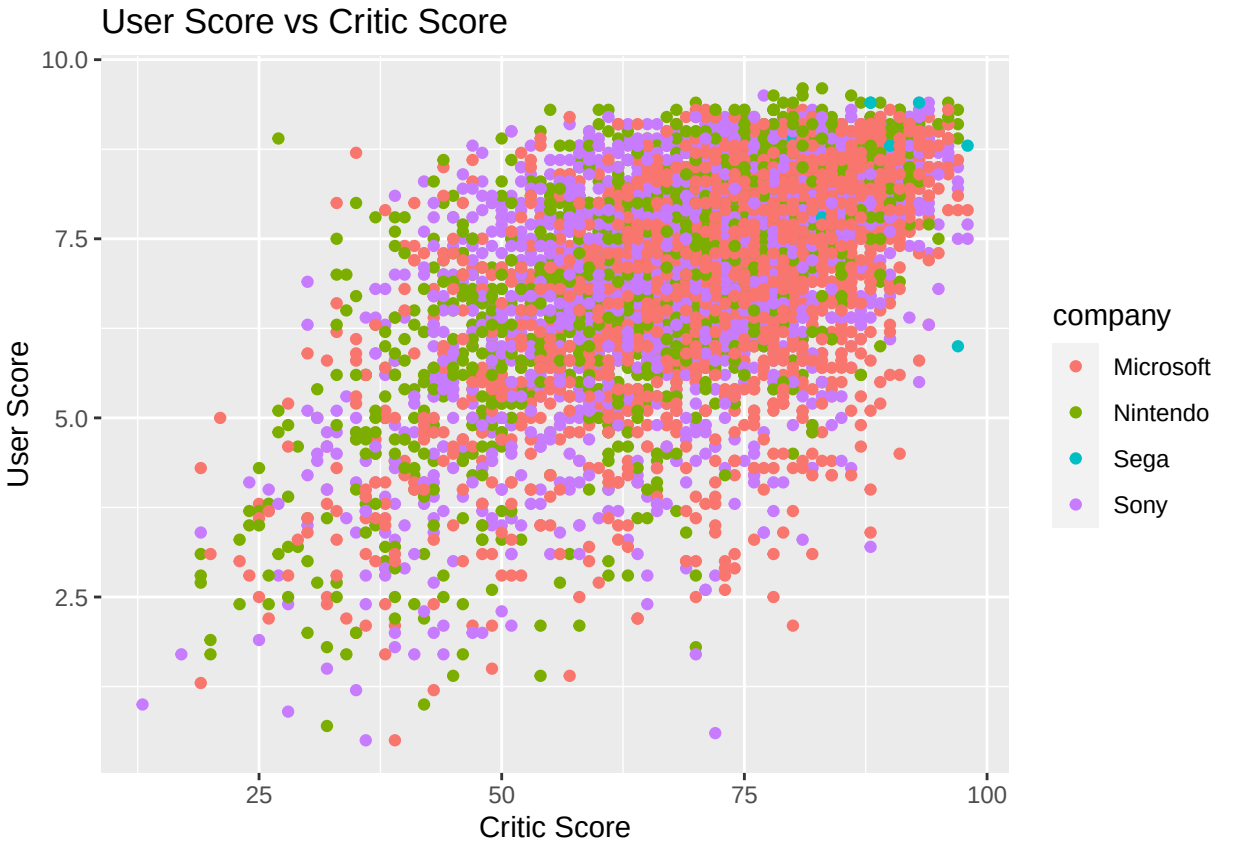
```
videogames %>% ggplot(aes(Global_Sales, Critic_Score, col = company)) +
               geom_point() +
               labs(x = "Global Sales", y = "Critic Score") +
               ggtitle("Critic Score vs Global Sales")
```



we can see that games that are rated with just a few point don't sell to much units but as the score increases, the games sell more units.

The next graph are the user score (values 0 - 10) vs the critic score (0 - 100). The colors represent the main company in the industry.

```
videogames %>% ggplot(aes(Critic_Score, User_Score, col = company)) +
               geom_point() +
               labs(x = "Critic Score", y = "User Score") +
               ggtitle("User Score vs Critic Score")
```



In this graph we can see a relationship between the two variables. In general, as the critic score increases the user score do it as well.

Modelling

We're going to fit four models, Linear Regression, K-Means Clustering, Regression Tree and Random Forest.

The Linear Regression is going to predict the user score of a game based on the critic score. Is important to denote that in the industry, the critics rank the game before its released. Meaning that the user score come after the critic rank. So, this model would be useful in this case.

The K-Means Clustering it's going to generate clusters on the same data, the idea it's to divide the data between very bad games, bad games, regular games, good games and excelent games. The Elbow Method it's going to use to determine the best k to this model.

We're going to use Regression Tree to determine the global sales based on mutiples variables. Really simple, the hierarchical model will show.

Finally, the Random Forest it's going to predict the NA sale on PS3 games. This subset is because the computational power of the method. The other reason is because to year 2016 (Year of the dataset) PS3 cover all his active life and we can see the overall perfomance of the console.

Results

First step before fit the models, we need to split the data into a training set and a test set.

```
test_index <- createDataPartition(videogames$User_Score, times = 1, p = 0.2, list = FALSE)

train_set <- videogames[-test_index,]
test_set <- videogames[test_index,]
```

train set consists on 80% of the data and test set 20%. The splitting proportion was chosen by the number of row that we have. with approximately 6800 rows a split of 90-10 would do that the train set have very few data. And 70-30 looks like a lot of punishment on training set data. So 80-20 looks like the best split.

Linear model

Before we fit the linear model, we're going to analyze the correlation on the two variables

```
cor(train_set$User_Score, train_set$Critic_Score)
```

```
## [1] 0.579607
```

we see a positive correlation and with really strong relation. So we proceed to fit the model

```
fit_lm <- lm(User_Score ~ Critic_Score, data = train_set)
fit_lm
```

```
##
## Call:
## lm(formula = User_Score ~ Critic_Score, data = train_set)
##
## Coefficients:
##  (Intercept)  Critic_Score
##      2.95681       0.06011
```

The equation of the linear model looks like this. Remember that the values of the user score go from 0 to 10 and critic score are in the range of 0 to 100.

$$userScore = 2.98 + criticScore * 0.06 \tag{1}$$

So, for example if the critic score results on 85 (Mario Kart 7). Calculating in the equation the estimated would be 8.08 (Real score 8.2).

So now we test our model on the test data and obtained the final RMSE.

```
y_hat_lm <- predict(fit_lm, test_set)
RMSE(y_hat_lm, test_set$User_Score)
```

```
## [1] 1.164112
```

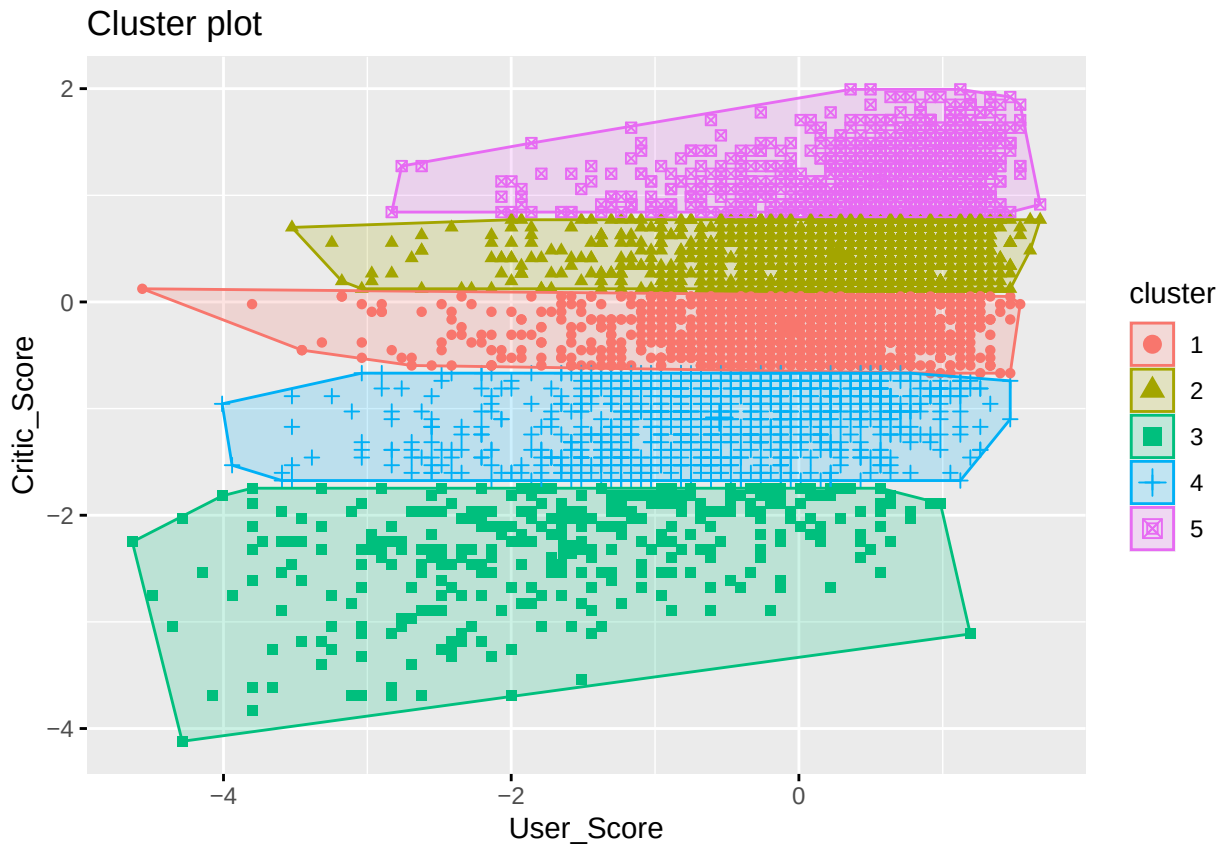
K-Means

The idea behind the k-means is to divide the data in different clusters, in this case we're going to use the user score and the critic score data to determine clusters on the quality of games. The library used to do this is factoextra. So, first we select the variables we want to analyze and then fit the model with k = 5 (Very bad games, bad games, regular games, good games, excellent games).

```
train_kmeans <- train_set %>% select(User_Score, Critic_Score)
fit_kmeans <- kmeans(train_kmeans, centers = 5, nstart = 25)
```

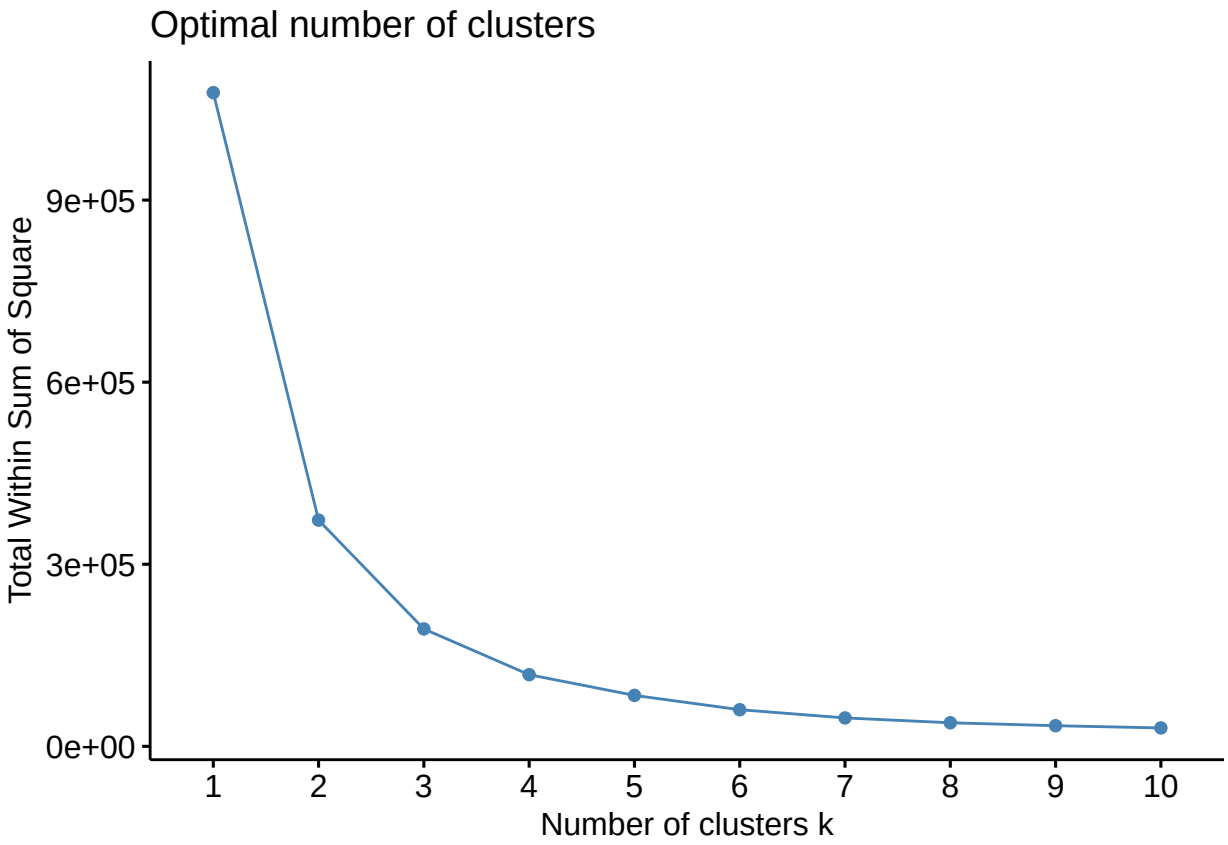
And plot the clusters with the following code.

```
fviz_cluster(fit_kmeans, train_kmeans, labelsize = 0)
```



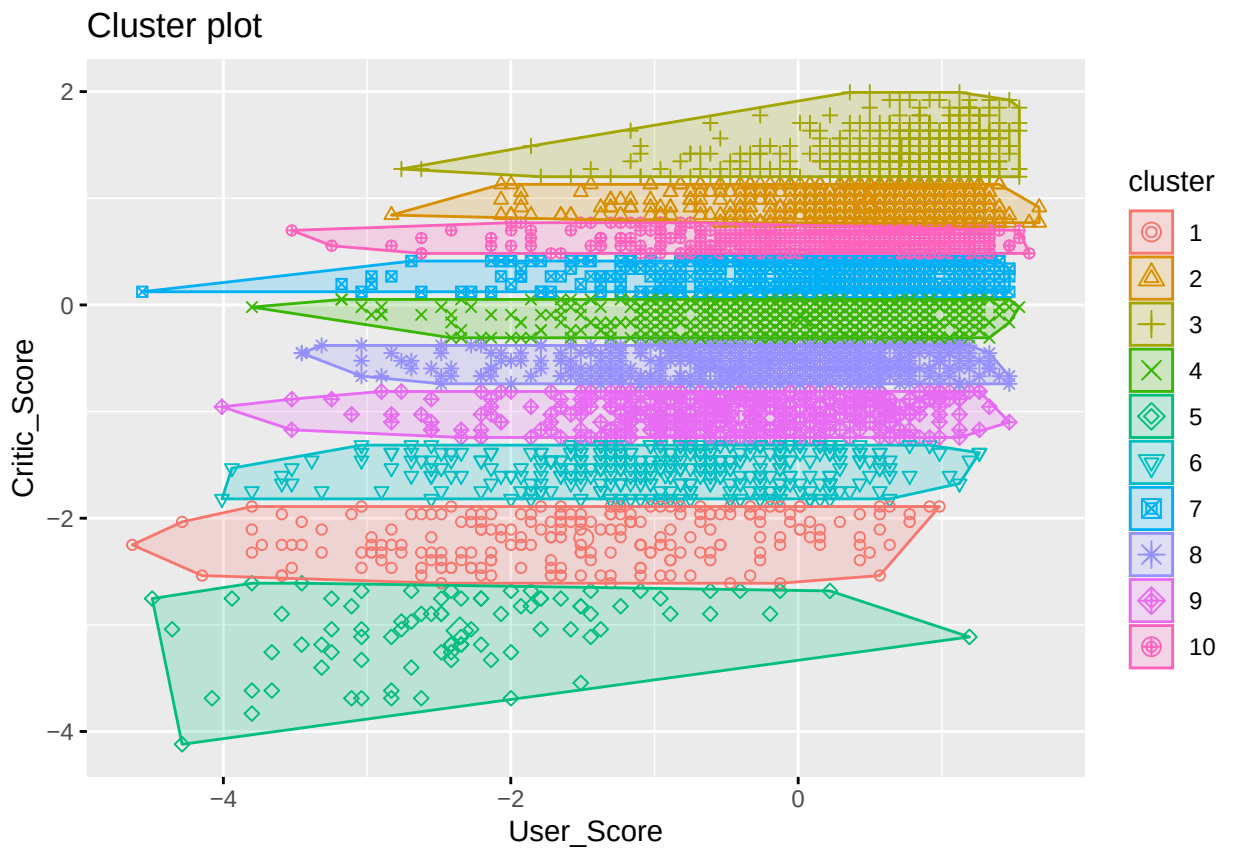
We can see that the cluster divide the data based on the quality of the game but on critic score. The User score have his variability on all the clusters. Now, we are going to test a better number of clusters based on elbow method

```
fviz_nbclust(train_kmeans, kmeans, method = "wss")
```



So basically the k converge to 10 where the error is less. So using k = 10.

```
fit_kmeans_10 <- kmeans(train_kmeans, centers = 10, nstart = 25)
fviz_cluster(fit_kmeans_10, train_kmeans, labelsize = 0)
```



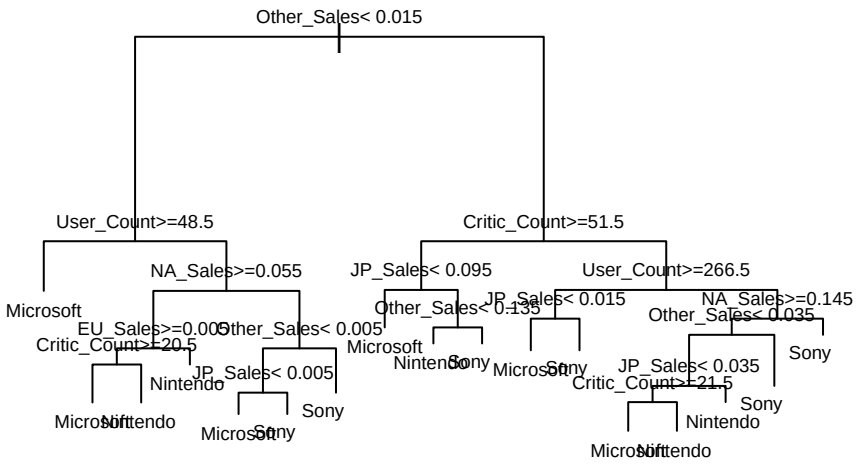
Regression Tree

So, it would be interesting to determine wich company released a particular game based on multiple variables. In this case the Tree could be model with this simple code.

```
fit_reg_tree <- rpart(company ~ User_Score +
                                User_Count +
                                Critic_Count +
                                Global_Sales +
                                Critic_Score +
                                NA_Sales +
                                JP_Sales +
                                Other_Sales +
                                EU_Sales, data = train_set)
```

The regression tree looks like this.

```
plot(fit_reg_tree, margin = 0.1)
text(fit_reg_tree, cex = 0.55)
```



So, we can see that other sales it's on the first place and that the user and critic count are really important in the tree. The critic score and the user score doesn't matter too much. And the sales of the game are predominant.

Random Forest

Like we mention, we're going to analyze the PS3 subset fitting a random forest algorithm.

```
ps3_train_set <- subset(train_set, Platform == "PS3")
ps3_test_set <- subset(test_set, Platform == "PS3")
```

The set have 638 and 137 observation in total. This way the random forest not going to explode our computer.

```
fit_rf <- randomForest(NA_Sales ~ User_Count +
                                  User_Score +
                                  Critic_Count +
                                  Critic_Score +
                                  Genre +
                                  Year_of_Release, data = ps3_train_set)
```

Now predicting the values and obtaining the RMSE value.

```
y_hat_rf <- predict(fit_rf, ps3_test_set)
RMSE(y_hat_rf, ps3_test_set$NA_Sales)
```

```
## [1] 0.4414749
```

Obtaining the variable importance.

```
varImp(fit_rf)
```

```
##                   Overall
## User_Count      116.25374
## User_Score       31.84830
## Critic_Count     35.66233
## Critic_Score     70.37784
## Genre            13.48334
## Year_of_Release  12.57320
```

Conclusion

The linear model fit perfomance really well based on that there is just one explanatory variable in the equation. The RMSE value range over 1 to 1.2 meaning that the score predicted could be in an error of about this values. This perfomance its explained because the two variable have a strong relationship and we see it on the plot.

The kmeans perfomance was good but not perfect. The data it's to close on each other with no clear clusters. The plot it's very clearly in this case. But overall we split the games quality based on critic score.

The regression tree was more for testing porpuse because we don't know the sales of a game before its released, but can be really good to determine how well a main company perfomance based on the popularity on the community and the sales based on world segmentation.

Finally, the random forest algorithm fit really well the perfomance of PS3 life. Really quick because of the fewer data on the dataset. And of course the use of multiples variables help the model to fit well. Importance to denote that the more important variables was the number of critics on the users side and the overall critic score impact a lot on the NA sales of a game.