Project Report

Sandeep Rani(SPR170000)

11/21/2019

Executive Summary:

As someone who was a serious gamer, I took up this interesting project to see how different factors affect the sales of video games. Factors like user and critic ratings, year of release , genre, rating, developers have a huge influence on the game sales. I found out some interesting insights particularly the console war. Playstation developed by Sony was the clearcut winner followed by XBOX and Nintendo. PS4 headed the sales for the 7th generation consoles followed by XboxOne and WiiI. Also after 2008, sales began to decline a bit. This was due to the advent of mobile phones as people began to transition into the mobile world. After performing few exploratory analysis, I created few models to represent the sales the best possible way. Linear regression,Random Forest, Ridge, KNN and Gradient Boosting were the regression models I used for my prediction. I used several techniques to finetune each model and arrive at the best possible accuracy. Finally, the results pointed at Random Forest as the best model for the given dataset as it had the lowest RMSE.

I am analyzing the video games sales from 1980 to 2017

Alongside the fields: Name, Platform, Year_of_Release, Genre, Publisher, NA_Sales, EU_Sales, JP_Sales, Other_Sales, Global_Sales, we have:-

- Critic_score - Aggregate score compiled by Metacritic staff
- Critic_count - The number of critics used in coming up with the Critic_score
- User_score - Score by Metacritic's subscribers
- User_count - Number of users who gave the user_score
- Developer - Party responsible for creating the game
- Rating - The ESRB ratings

Initially I performed EDA for the dataset. Below are the variables in the dataset.

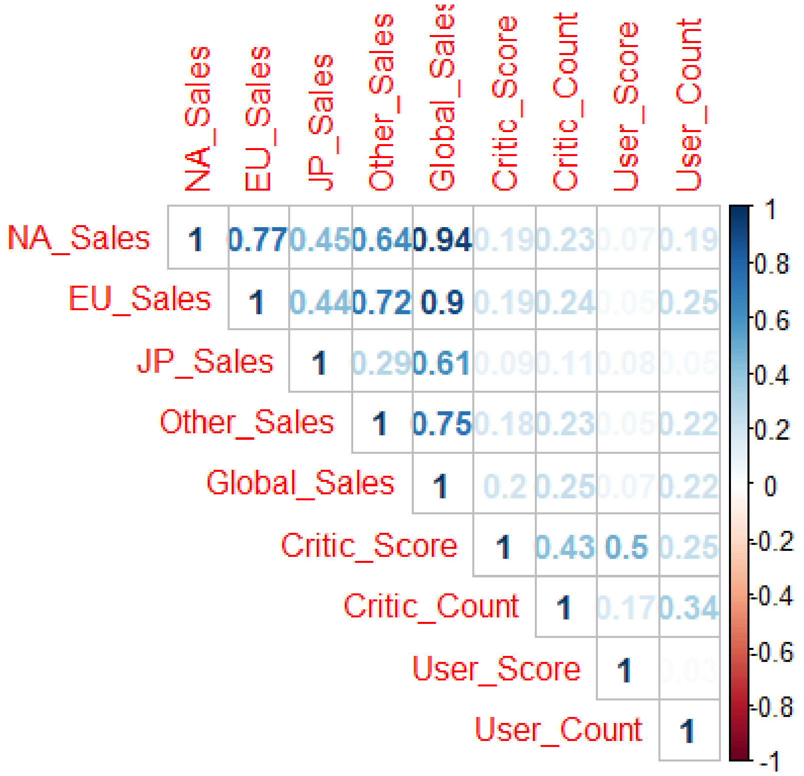
EDA

```
##            Name        Platform Year_of_Release           Genre 
##        "factor"        "factor"        "factor"        "factor" 
##       Publisher        NA_Sales        EU_Sales        JP_Sales 
##        "factor"       "numeric"       "numeric"       "numeric" 
##     Other_Sales    Global_Sales    Critic_Score    Critic_Count 
##       "numeric"       "numeric"       "integer"       "integer" 
##      User_Score      User_Count       Developer          Rating 
##        "factor"       "integer"        "factor"        "factor"
```

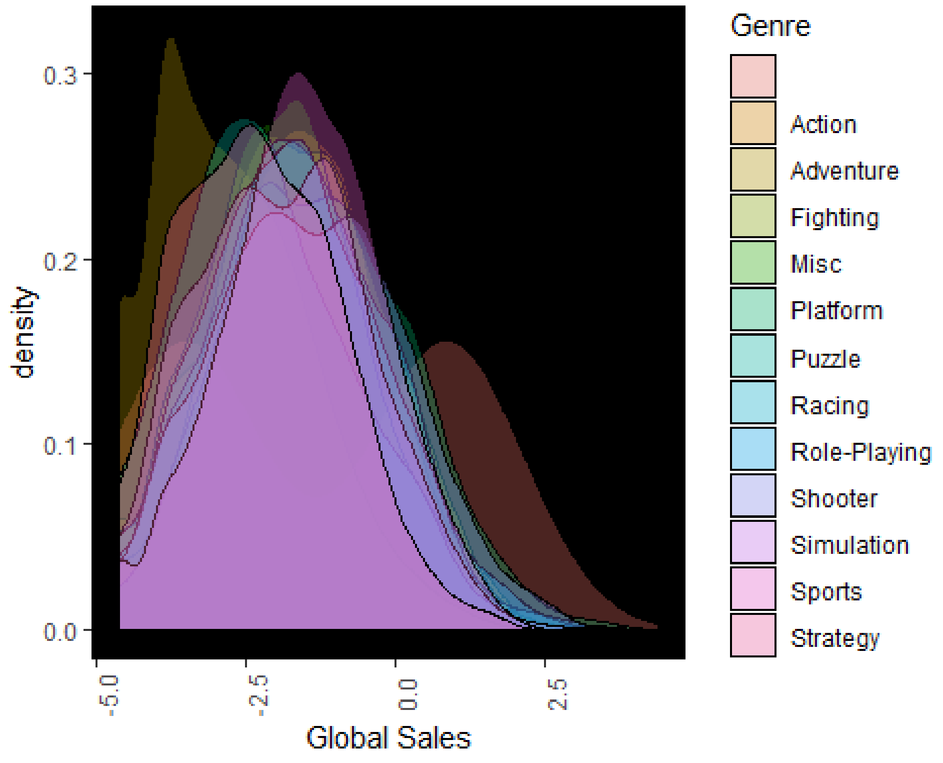
```
## 'data.frame':    16719 obs. of  16 variables:
##  $ Name           : Factor w/ 11563 levels "","'98 Koshien",..: 11059 9406 
5573 11061 7417 9771 6693 11057 6696 2620 ...
##  $ Platform       : Factor w/ 31 levels "2600","3DO","3DS",..: 26 12 26 26 
6 6 5 26 26 12 ...
##  $ Year_of_Release: Factor w/ 40 levels "1980","1981",..: 27 6 29 30 17 10 
27 27 30 5 ...
##  $ Genre          : Factor w/ 13 levels "","Action","Adventure",..: 12 6 8 
12 9 7 6 5 6 10 ...
##  $ Publisher      : Factor w/ 582 levels "10TACLE Studios",..: 371 371 371 
371 371 371 371 371 371 371 ...
##  $ NA_Sales       : num  41.4 29.1 15.7 15.6 11.3 ...
##  $ EU_Sales       : num  28.96 3.58 12.76 10.93 8.89 ...
##  $ JP_Sales       : num  3.77 6.81 3.79 3.28 10.22 ...
##  $ Other_Sales    : num  8.45 0.77 3.29 2.95 1 0.58 2.88 2.84 2.24 0.47 ..
.
##  $ Global_Sales   : num  82.5 40.2 35.5 32.8 31.4 ...
##  $ Critic_Score   : int  76 NA 82 80 NA NA 89 58 87 NA ...
##  $ Critic_Count   : int  51 NA 73 73 NA NA 65 41 80 NA ...
##  $ User_Score     : Factor w/ 97 levels "","0","0.2","0.3",..: 79 1 82 79 
1 1 84 65 83 1 ...
##  $ User_Count     : int  322 NA 709 192 NA NA 431 129 594 NA ...
##  $ Developer      : Factor w/ 1697 levels "","10tacle Studios",..: 1035 1 
1035 1035 1 1 1035 1035 1035 1 ...
##  $ Rating         : Factor w/ 9 levels "","AO","E","E10+",..: 3 1 3 3 1 1 
3 3 3 1 ...

## [1] 0
```

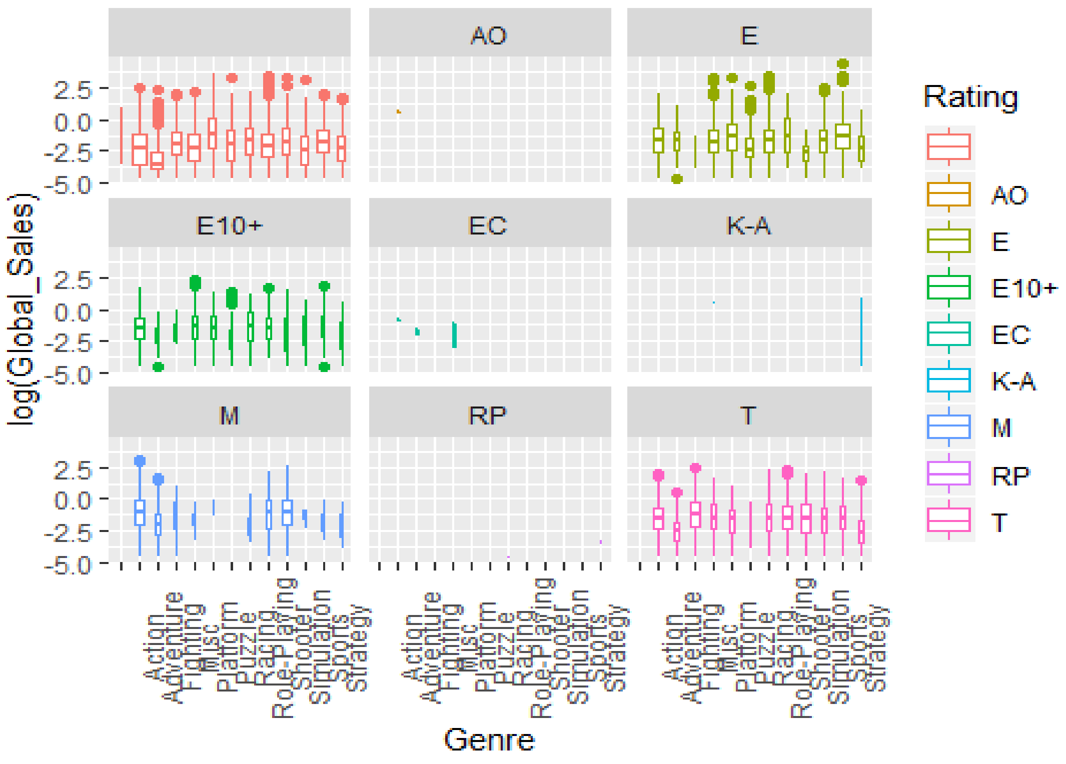
First, I analyze the correlation plot. We can see that the Global sales is more correlated with North America Sales, Europe sales, Japan sales and other sales as expected. Also we can see a correlation between user and critic score, critic score and critic count. User count and user score do not have much of a correlation.



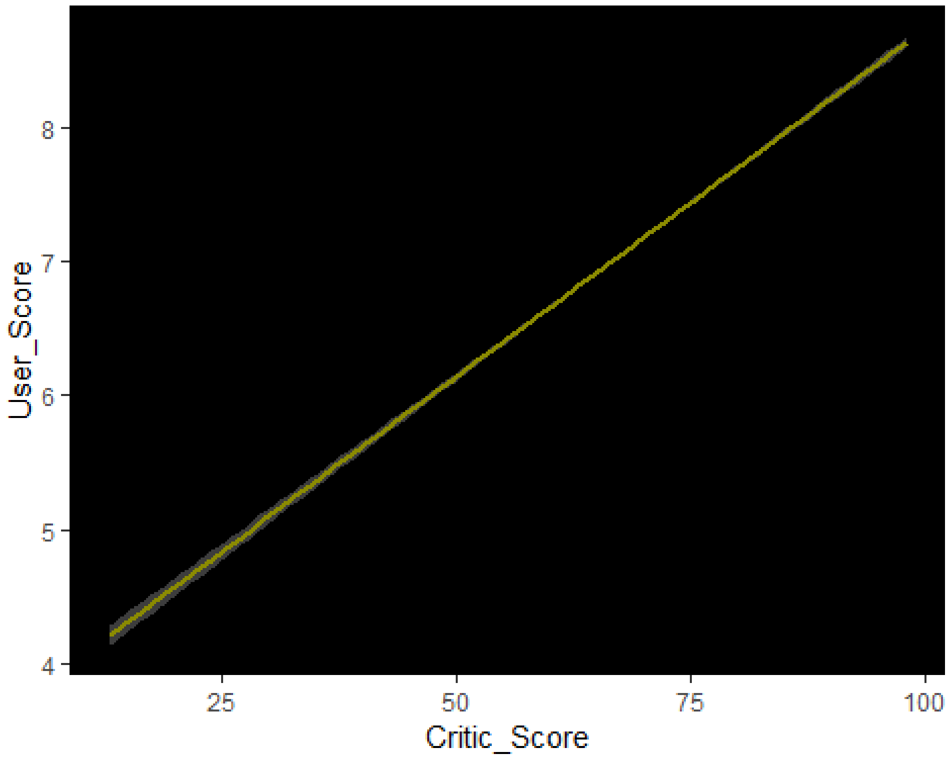
I use log of Sales for showing distribution. All the genres have a wide distribution. Sports being the most popular genre has the widest distribution



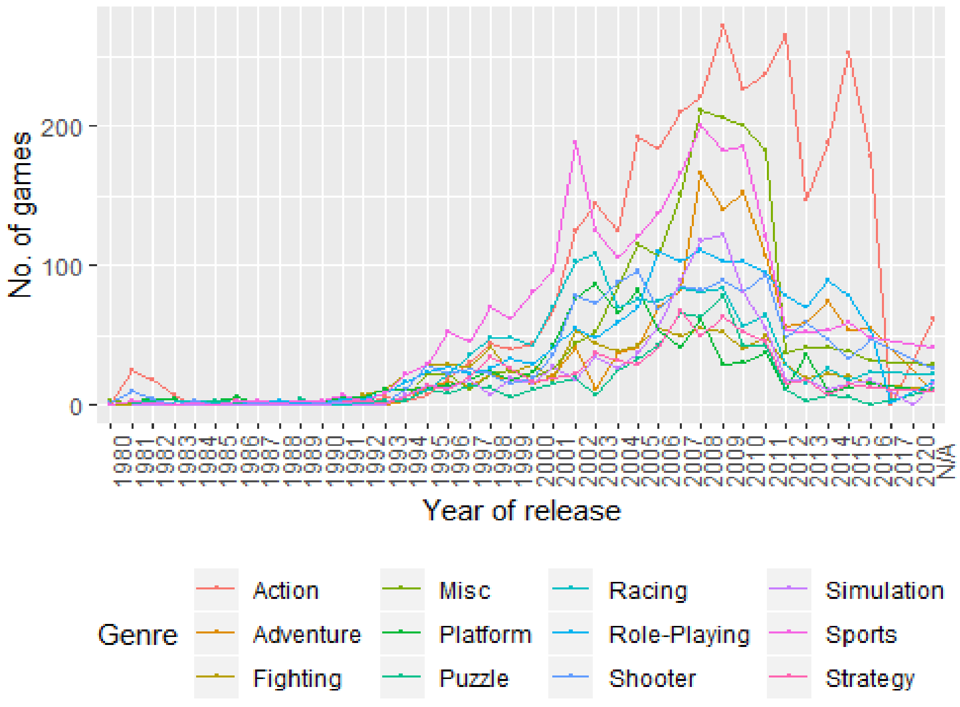
Several Ratings like M, AO, E etc exists for video games, what does it actually mean. Rating refers to age appropriateness for the games. . E - Everyone . E10+- Everyone 10+ . T - Teen . M - Mature . EC- Early Childhood . AO- Adults Only . RP- Rating Pending . K A - Kids to Adults###Rating and Sales AO, EC, K-A, RP do not have much of data . E, E10+,T ,M . In E10+ category, Sports has the highest median and the highest sales. Puzzle, Racing, RPG, Action are all popular for E10+. RPG, Shooter and Action are best selling among Mature genre. For teens, all genres expect adventure, puzzle and strategy seem to do well . We



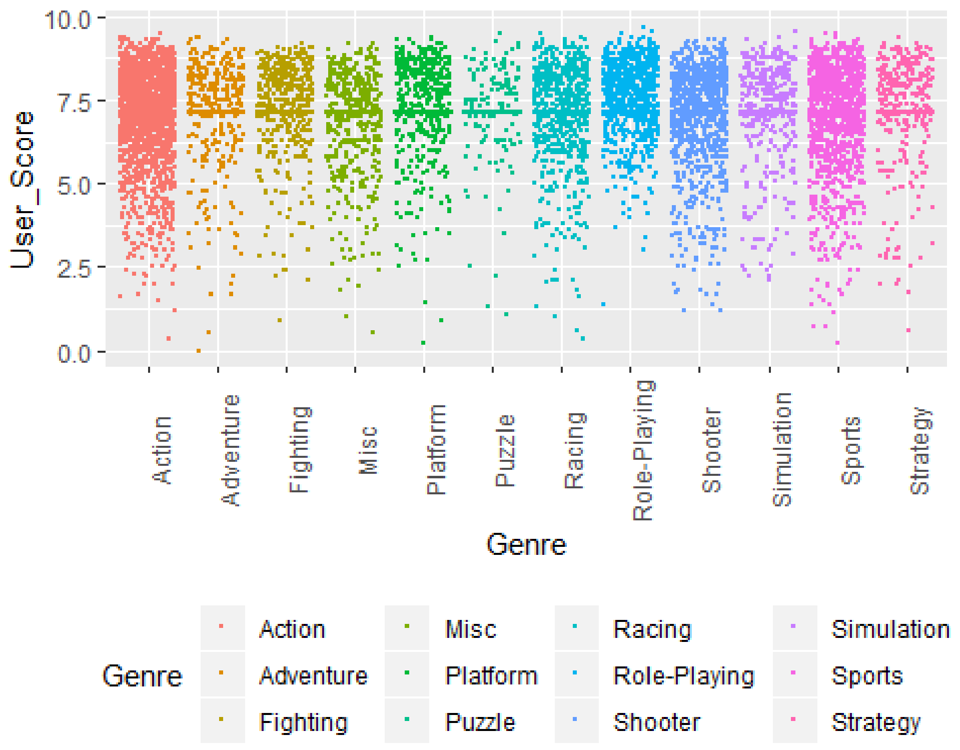
conclude that genres and rating have an impact on global sales.



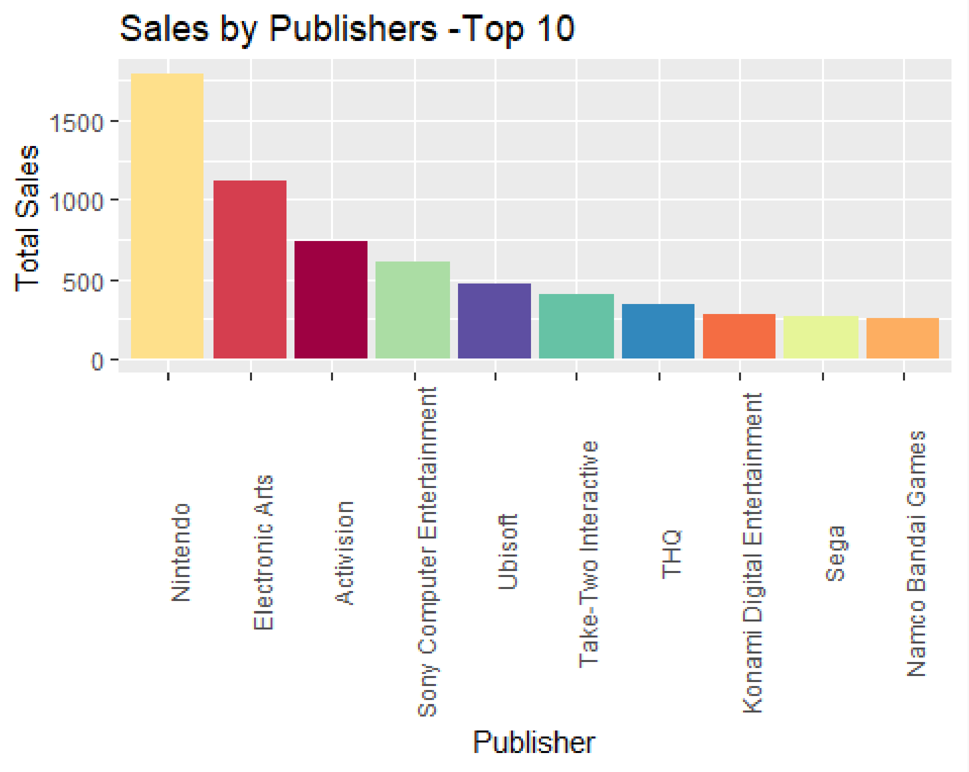
We can see below that between 1980-1995 , the sales was too low because it was still in development stages and not popular. The sales reached started to pick up after 1995 and reached the peak during 2005- 2011. After that it started to drop again.



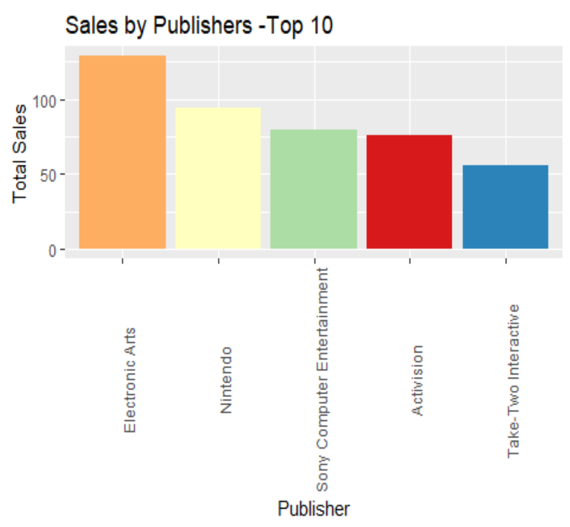
Action games seem to perform the best among all genres followed by Sports.

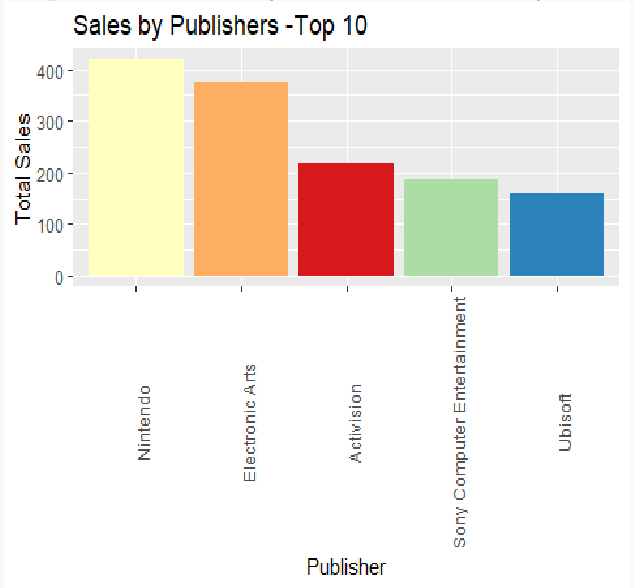


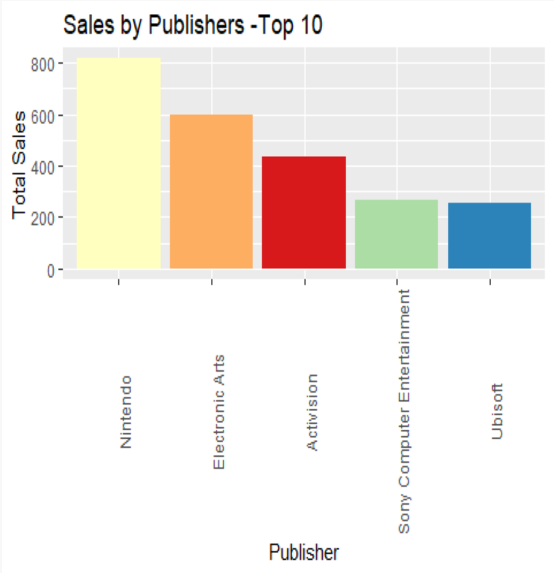
From the below chart, we can see that Nintendo, Electronic Arts, Activision are the top 3 pe rformers globally. People certainly have a preference for games by these publishers.

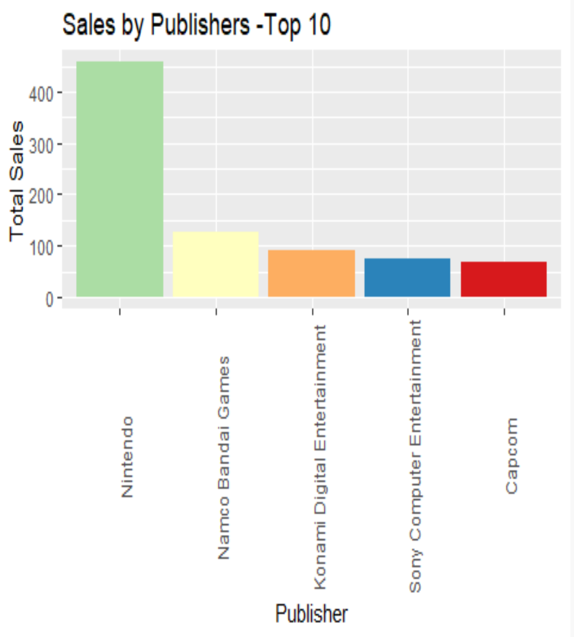


As see below, Nintendo, EA and Activison do very well in North America and Europe. but in Japan. Nintendo is followed by Namco and Konami indication local dominance over global p owers. In other countries EA occupies the first place followed by Nintendo and Sony .

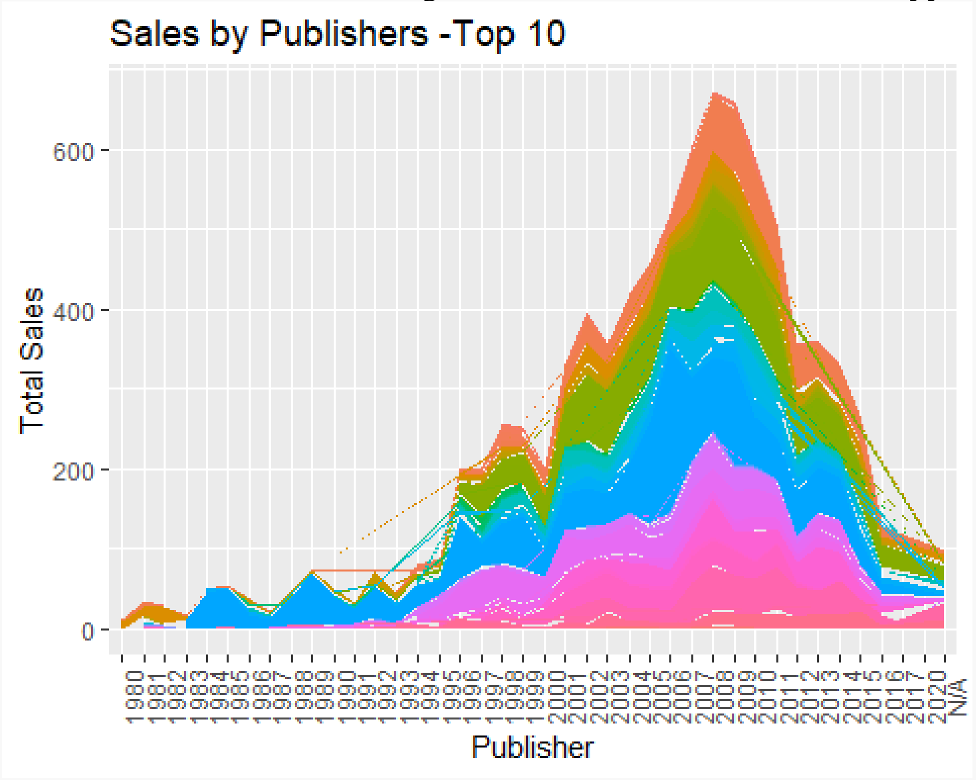




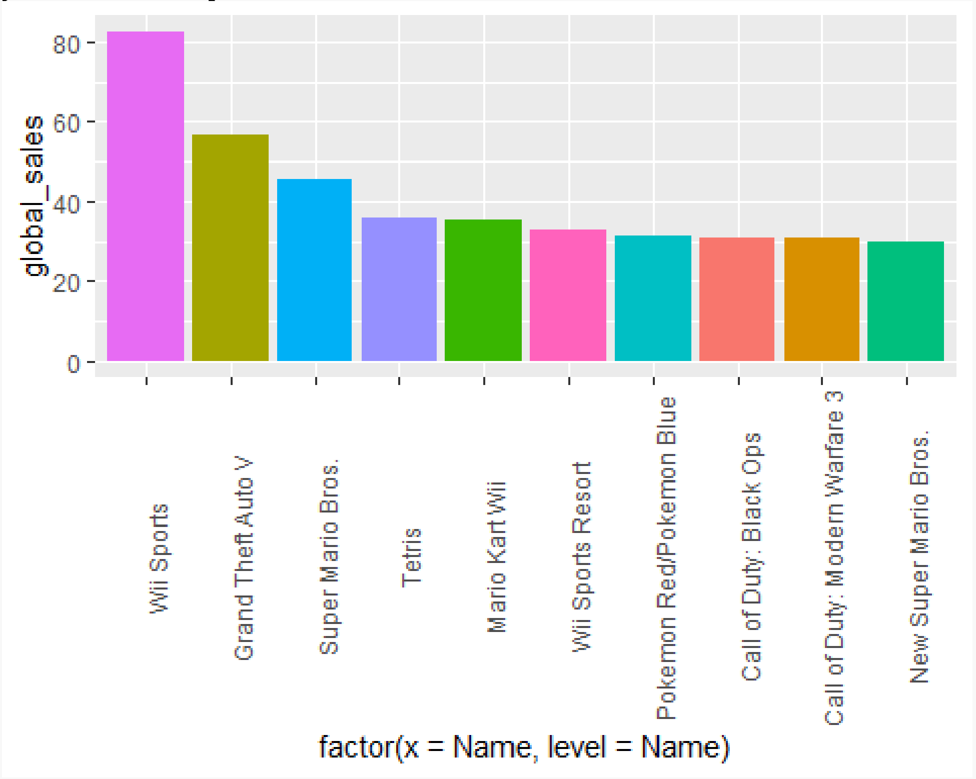




All the publishers have a peak point at 2008 and then started to plummet. Why? Bec ause that is when mobile phones began to become popular and hence people found an alter native to kill time. Some of the games became available in mobile apps.



Next we analyze the best selling video games. Wii Sports is the best seller followed by GTA 5 and Super Mario



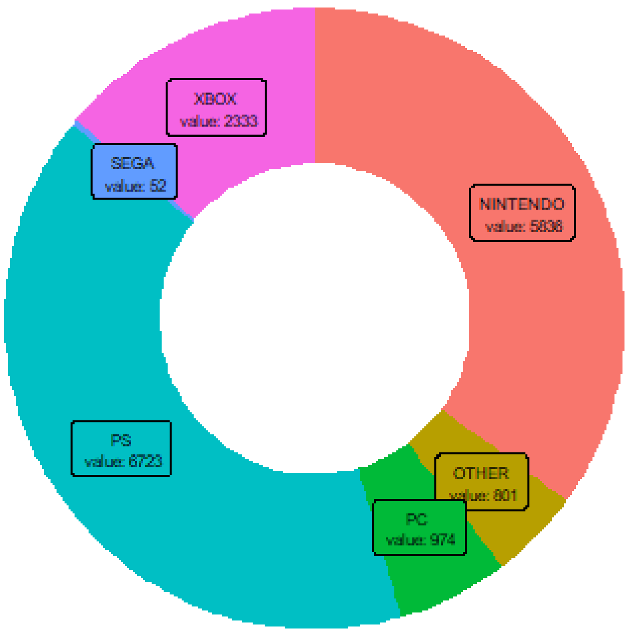
Predictive modelling

I am going to predict Video game Sales and what factors it depends on. For this i cre ated 5 models:

1. Linear regression
2. Random Forest
3. Gradient Boosting
4. Ridge Regression
5. KNN

For all the above models, i have used RMSE as the parameter for comparison. I have used cross validation to fine tune the hyperparameters.

Here I found lot of missing values. The critic score particularly found only in half of the records.I replaced the categorical columns by mode of the column . I created a new column called newPlatform that aggregates as PlayStation,XBOX,Nintendo,Sega and the rest as Others. There are too many different platforms and most of them represent a very small percent of games. I am going to group platforms to reduce the number of features.



Weighted Score and Developer Rating

In order to handle for the missing values in critic and user rating, I created 2 new features: weighted score and developer rating. THe weighted score is calculated as the average of user and critic rating.for Develoepr rating, I find percent of all games created by each developer, then calculate cumulative percent starting with developers with the least number of games. Ultimately, I divide them into 5 groups each containing 20%.Higher the rating, the more number of games it developed

```
##            Name        Platform Year_of_Release           Genre
##        "factor"        "factor"       "numeric"        "factor"
##       Publisher        NA_Sales        EU_Sales        JP_Sales
##        "factor"       "numeric"       "numeric"       "numeric"
##     Other_Sales    Global_Sales    Critic_Score    Critic_Count
##       "numeric"       "numeric"       "numeric"       "numeric"
##      User_Score      User_Count       Developer          Rating
##       "numeric"       "numeric"        "factor"        "factor"
##  weighted_score
##       "numeric"
```

Once all the above feature transforamtion and data cleaning was done, it was time to build models. We split into 70% training data and 30% test data. I then scaled the data to reduce variance.

```
## [1] "build_scales: I will compute scale on  7 numeric columns."
## [1] "build_scales: it took me: 0s to compute scale for 7 numeric columns."

## [1] "fastScale: I will scale 7 numeric columns."
## [1] "fastScale: it took me: 0.02s to scale 7 numeric columns."

## [1] "fastScale: I will scale 7 numeric columns."
## [1] "fastScale: it took me: 0s to scale 7 numeric columns."
```

Linear Regression

I have iniitally run a linear regression model.

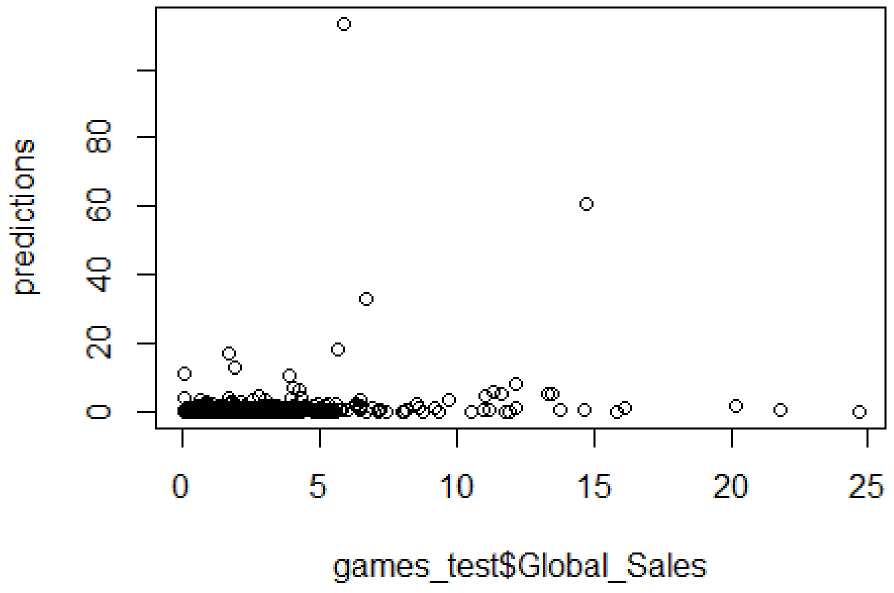
```
## 
## Call:
## lm(formula = logsales ~ newPlatform + Genre + Critic_Score + 
##     Critic_Count + User_Count + User_Score + age + Developer_Rating + 
##     weighted_score + country, data = games_train)
## 
## Residuals:
##     Min      1Q  Median      3Q     Max 
## -4.9880 -0.8167 -0.0304  0.7802  5.3039 
## 
## Coefficients:
##                        Estimate Std. Error t value Pr(>|t|)    
## (Intercept)            -1.60241    0.84637  -1.893   0.0583 .  
## newPlatformOTHER        0.63716    0.06969   9.142  < 2e-16 ***
## newPlatformPC        -1.61994    0.06039 -26.824  < 2e-16 ***
## newPlatformPS         0.01619    0.02732   0.593   0.5534
## newPlatformSEGA       0.27059    0.19981   1.354   0.1757
## newPlatformXBOX      -0.29680    0.03811  -7.787 7.43e-15 ***
## GenreAction           0.34685    0.84568   0.410   0.6817
## GenreAdventure       -0.17313    0.84636  -0.205   0.8379
## GenreFighting         0.37899    0.84654   0.448   0.6544
## GenreMisc             0.45111    0.84600   0.533   0.5939
## GenrePlatform         0.49458    0.84643   0.584   0.5590
## GenrePuzzle           0.00546    0.84730   0.006   0.9949
## GenreRacing           0.22006    0.84638   0.260   0.7949
## GenreRole-Playing     0.49463    0.84609   0.585   0.5588
## GenreShooter          0.24792    0.84622   0.293   0.7695
## GenreSimulation       0.44677    0.84674   0.528   0.5978
## GenreSports           0.50537    0.84581   0.597   0.5502
## GenreStrategy         0.17510    0.84711   0.207   0.8362
## Critic_Score         -0.05058    0.03493  -1.448   0.1476
## Critic_Count          0.47612    0.01970  24.164  < 2e-16 ***
## User_Count            0.21282    0.01282  16.603  < 2e-16 ***
## User_Score            0.27827    0.02759  10.087  < 2e-16 ***
## age                   0.29799    0.01506  19.793  < 2e-16 ***
## Developer_Rating      0.19417    0.01690  11.488  < 2e-16 ***
## weighted_score       -0.27683    0.04304  -6.432 1.31e-10 ***
## countryJapan         -0.83312    0.04565 -18.251  < 2e-16 ***
## countryNorth America -0.29893    0.03963  -7.543 4.95e-14 ***
## countryOther          1.00315    0.18673   5.372 7.93e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.193 on 11488 degrees of freedom
## Multiple R-squared:  0.3356, Adjusted R-squared:  0.334
## F-statistic: 214.9 on 27 and 11488 DF,  p-value: < 2.2e-16
```

newPlatform,Critic Score, Critic Count.Uer Score,User Count,age ,DeveloperRating,weighted score and country are all significant variables in this model. Genre seems highly insignifcant and does not contribute to prediction of the sales. The R-suqare is 33.4%. The train error is as below.

```
## [1] 3.218514
```

I wanted to use cross valdiation to improve the estimates. The model might be overfitting and hence i run a 3 fold cross validation

```
## Linear Regression
##
## 11516 samples
##    11 predictor
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 10364, 10365, 10365, 10365, 10365, 10364, ...
## Resampling results:
##
##   RMSE      Rsquared   MAE
##   1.190355  0.3378525  0.9426
##
## Tuning parameter 'intercept' was held constant at a value of TRUE
```



The resulting test error was 2.09. Lets see if we can reduce the error value in the next models.

```
## [1] 2.095374
```

Random Forest

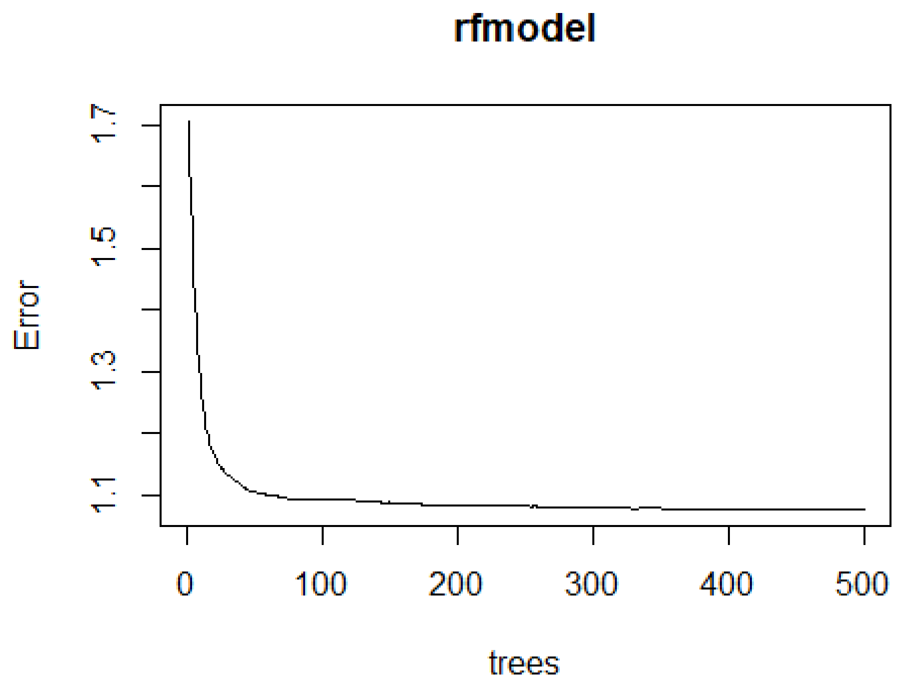
I ran Random forest model which is an ensemble method over decision trees. THey are popular for reducing overfitting and providing good estimates. I initally experiment wiht the number of variables at each split from 1 to 10. Both out of bag error and test errors have been calculated.

We choose 3 as the numebr of predictors to be considered at the split as the curve seems to overfit after 3 . I rebuilt the model and found out the test and train error.

Train error- `1.04619`

Test error- `1.094527`

```
The R-square I achieved was 48.80%.
```

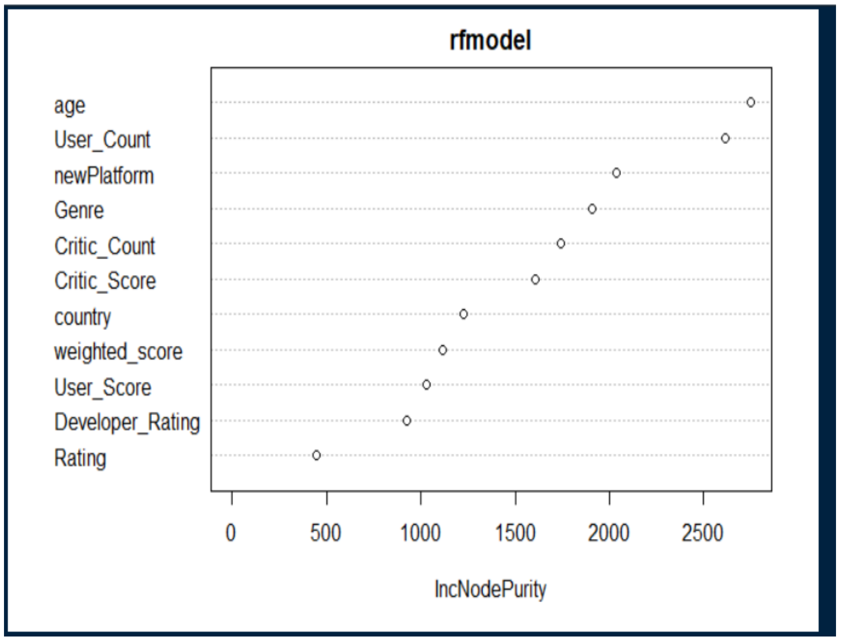


```
## [1] 0.4880195

## [1] 1.04619

## [1] 1.094527
```

The importance fucntion gives the important variables in the order of importance. All of them seem to have a non zero effect on Sales.

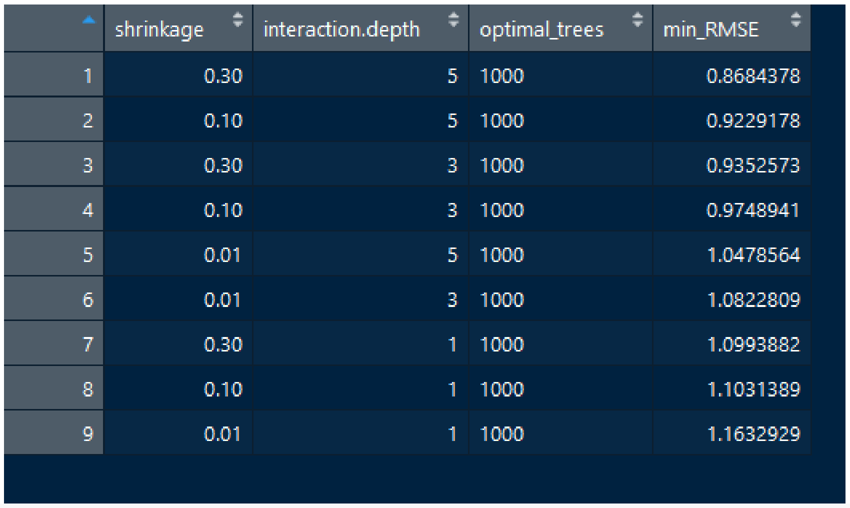


Gradient Boosting

For Gradient Boosting, I experiment with learning rates 0.01,0.10.3 and depth of the tree 1,3,5

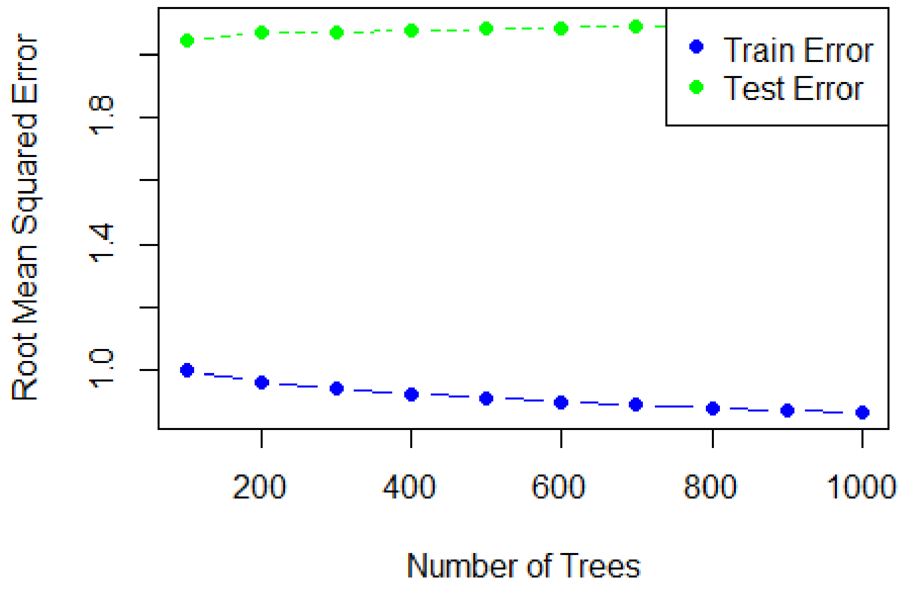
Learning rate=0.3 and Depth=5 gave the lowest errors.

```
hyper_grid<-hyper_grid %>%
  dplyr::arrange(min_RMSE) %>%
  head(10)
```



	shrinkage	interaction.depth	optimal_trees	min_RMSE
1	0.30	5	1000	0.8684378
2	0.10	5	1000	0.9229178
3	0.30	3	1000	0.9352573
4	0.10	3	1000	0.9748941
5	0.01	5	1000	1.0478564
6	0.01	3	1000	1.0822809
7	0.30	1	1000	1.0993882
8	0.10	1	1000	1.1031389
9	0.01	1	1000	1.1632929

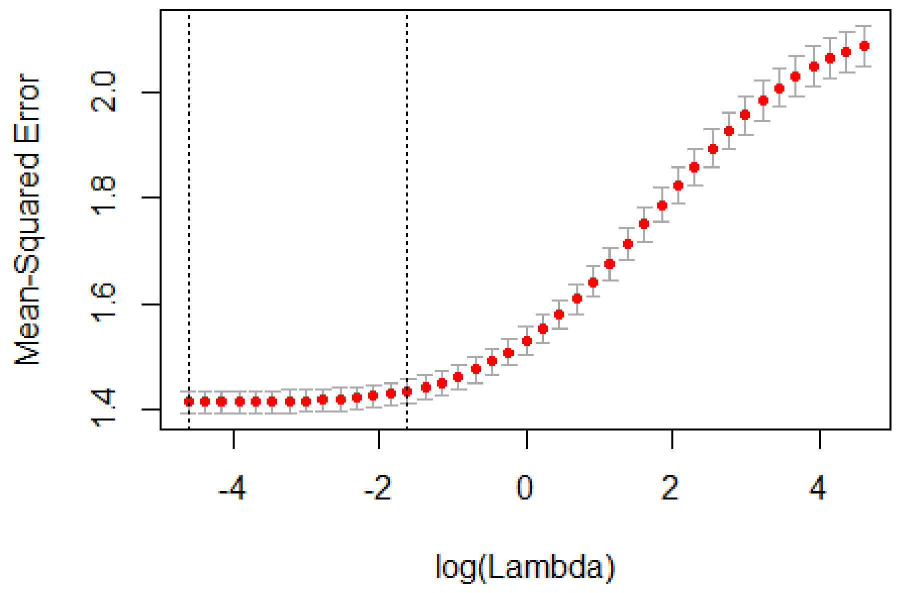
The optimum values are now used to train the model. I find out from the plot that for all number of trees the test error is quite high around 2 while the train error drops below 1. This is a case of good generalization of the train set but not unknown data. This is a clear case of overfitting. Hence I reject gradient boosting model for my prediction.



Ridge Regression

Initially we want to tune the model by experimenting with lambda values.Then I find the best value and rebuild the model and test it with the test data. THe optimum value of lambda was 0.01. I obtained a test error of 2.053557 which was lesser than linear regression but is way higher than random forest.

```
##           Length Class     Mode
## a0          41   -none-    numeric
## beta      1435   dgCMatrix S4
## df          41   -none-    numeric
## dim          2   -none-    numeric
## lambda      41   -none-    numeric
## dev.ratio   41   -none-    numeric
## nulldev      1   -none-    numeric
## npasses      1   -none-    numeric
## jerr         1   -none-    numeric
## offset       1   -none-    logical
## call         5   -none-    call
## nobs         1   -none-    numeric
```



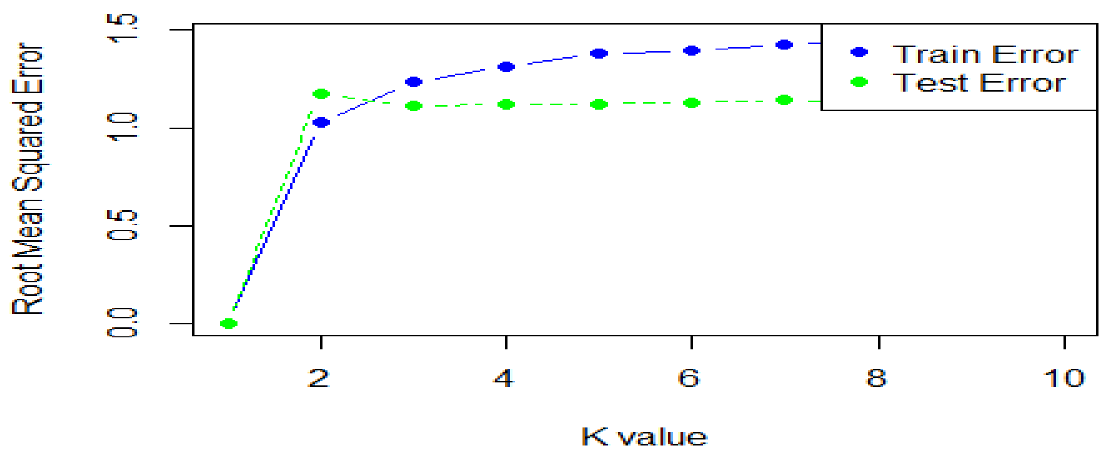
```
##             Length Class     Mode
## a0           1     -none-    numeric
## beta        35     dgCMatrix S4
## df           1     -none-    numeric
## dim          2     -none-    numeric
## lambda       1     -none-    numeric
## dev.ratio    1     -none-    numeric
## nulldev      1     -none-    numeric
## npasses      1     -none-    numeric
## jerr         1     -none-    numeric
## offset       1     -none-    logical
## call         5     -none-    call
## nobs         1     -none-    numeric

## 36 x 1 sparse Matrix of class "dgCMatrix"
##                                 s0
## (Intercept)          -1.4400378487
## (Intercept)           .
## newPlatformOTHER      0.6757908704
## newPlatformPC        -1.5338371821
## newPlatformPS         0.0557132001
## newPlatformSEGA       0.3296293524
## newPlatformXBOX      -0.2429406798
## GenreAction           0.0454988067
## GenreAdventure       -0.5026114007
## GenreFighting         0.1279825526
## GenreMisc             0.1157066164
## GenrePlatform         0.1056629311
## GenrePuzzle          -0.3767444744
## GenreRacing          -0.1721135317
## GenreRole-Playing      0.2056705179
## GenreShooter          -0.0005077739
## GenreSimulation        0.1152270899
## GenreSports            0.0966575355
## GenreStrategy         -0.1513304814
## Critic_Score          -0.0492203640
## Critic_Count           0.4965394491
## User_Count             0.2106570477
## User_Score             0.2900172348
## age                    0.2835926746
## RatingE                0.2305628050
## RatingE10+             0.0628588891
## RatingEC               0.8804702307
## RatingK-A              4.0151986302
## RatingM               -0.0992934900
## RatingRP                .
## RatingT               -0.1889319145
## Developer_Rating       0.1980483268
## weighted_score        -0.2168850443
## countryJapan          -0.8262929073
## countryNorth America -0.2596520649
## countryOther           1.0326800571

## [1] 2.053557
```

KNN

K Nearest Neighbours is another model which requires no training before making predictions.New data can be added seamlessly which will not impact the accuracy of the algorithm. I needed to choose a good value of K for prediction. Hence i plotted the train and test errors for different K values from 2 to 10. WE can see that after k=2, the training error becomes higher than the test error whihc is not ideal. Hence i choose K=2. I rebuilt the model using K=2. I achieved train error of 1.02 and test error of 1.17 which is quite good.



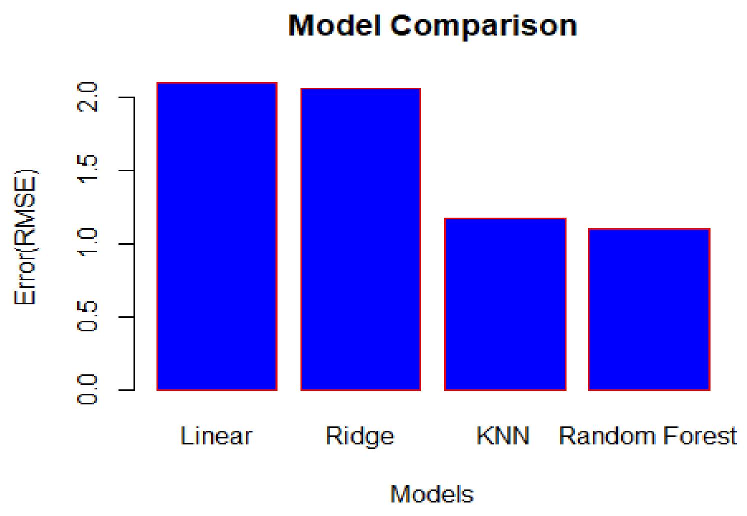
```
## [1] 1.025286

## [1] 1.173264

## [1] "numeric"
```

CONCLUSION:

We can see that Random Forest gives the lowest RMSE at 1.09 followed by KNN. Linear and Ridge regression gave almost similar results with Ridge performing slightly better. With further finetuning of the depth, number of iterators and number of trees, the Random Forest can give even better results. Hence I conclude by saying that for this dataset of video game sales, Random Forest is the best performing model amongst all.


Model Comparison
Error(RMSE)
Linear
Ridge
KNN
Random Forest
Models