

Introduction

Video Game Sales with Ratings dataset. The dataset is obtained from <https://www.kaggle.com/rush4ratio/video-game-sales-with-ratings>. Aim is to predict the global sales of games (Global_Sales). First column (Name) represents the name of games. First column was not used in the models. Ninth column is response/dependent variable, the global sales of games (Global_Sales). Other values are predictors/independent variables.

Analysis

a)

```
1
2 # Libraries
3 library(glmnet)
4 library(mice)
5 library(caret)
6 library(tidyverse)
7 library(rpart)
8 library(randomForest)
9
10 #####
11
12 # import data
13 data <- read.csv("FinalDataset.csv", sep=";")
14
15 #####
16
17 #preprocessing data
18 minMaxNorm <- function(x){
19
20   return ((x-min(x)) / (max(x) - min(x)))
21 }
22
23 data$X=NULL
24
25 data$Global_Sales <- gsub(",", ".", data$Global_Sales)
26 data$Global_Sales <- as.double(data$Global_Sales)
27 data$Platform <- as.factor(data$Platform)
28 data$Genre <- as.factor(data$Genre)
29 data$Rating <- as.factor(data$Rating)
30 data$PublisherR <- as.factor(data$PublisherR)
31
32 data$Critic_Score <- minMaxNorm(data$Critic_Score)
33 data$Critic_Count <- minMaxNorm(data$Critic_Count)
34 data$Global_Sales <- minMaxNorm(data$Global_Sales)
35
36 data <- sapply(data, unclass)
37
38 #data info
39 cor(data)
40 pairs(na.omit(data))
41 md.pattern(data)
```

b)

```
42
43 # train data and test data
44 train <- as.data.frame(data[1:1000,])
45 test <- as.data.frame(data[1001:nrow(data),])
46
47 names(train)
48 names(test)
49
```

c)

```
50
51
52 #Regression Model
53 modelRegression <- lm(Global_sales ~ ., data = train)
54 # coefficients, adjusted R square and F statistic of Regression the model
55 summary(modelRegression)
56
```

d)

```
Coefficients:
      Estimate Std. Error t value Pr(>|t|)
(Intercept)  1.2959120  3.7405064   0.346 0.729075
Platform     0.0017623  0.0013127   1.342 0.179747
Year        -0.0006687  0.0018603  -0.359 0.719334
Genre       -0.0005454  0.0008800  -0.620 0.535517
Rating      -0.0106732  0.0028085  -3.800 0.000153 ***
PublisherR   -0.0034778  0.0022034  -1.578 0.114800
Critic_Score 0.1571656  0.0220638   7.123 2.02e-12 ***
Critic_Count 0.0975811  0.0141178   6.912 8.55e-12 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.0916 on 992 degrees of freedom
Multiple R-squared:  0.1395,    Adjusted R-squared:  0.1334
F-statistic: 22.97 on 7 and 992 DF,  p-value: < 2.2e-16
```

e)

```
59 #Regression Prediction and RMSE
60 predictionsReg <- predict(modelRegression, test)
61 Reg_RMSE <- RMSE(predictionsReg, test$Global_Sales)
```

Result = 0.03693007

f)

```
92 #Determine the lambda parameter using cross-validation
93 lambdas = 10^seq(3,-2,by=-.01)
94 modelRidgeCV <- cv.glmnet(trainRidgeLasso_x,trainRidgeLasso_y, alpha = 0, lambda = lambdas, nfolds = 10)
95 modelRidgeCV$lambda.min
96
97 #Ridge Model
98 modelRidge <- glmnet(trainRidgeLasso_x,trainRidgeLasso_y, alpha = 0, lambda = modelRidgeCV$lambda.min)
99 summary(modelRidge)
100
101 #####
102
103 #Determine the lambda parameter using cross-validation
104 modelLassoCV <- cv.glmnet(trainRidgeLasso_x,trainRidgeLasso_y, alpha = 1, lambda = lambdas, nfolds = 3)
105 modelLassoCV$lambda.min
106
107 #Lasso Model
108 modelLasso <- glmnet(trainRidgeLasso_x,trainRidgeLasso_y, alpha = 1, lambda = modelLassoCV$lambda.min)
109 summary(modelLasso)
110
```

g) The regression model worked more accurately.

h)

```
113 # Ridge and Lasso Prediction and RMSE
114 predictionRidge <- predict(modelRidge, testRidgeLasso_x)
115 predictionLasso <- predict(modelLasso, testRidgeLasso_x)
116
117 Ridge_RMSE <- RMSE(predictionRidge, testRidgeLasso_y)
118 Lasso_RMSE <- RMSE(predictionLasso, testRidgeLasso_y)
119
120 Ridge_RMSE
121 Lasso_RMSE
122
```

Result Ridge = 0.4610888

Result Lasso = 0.4647719

i)

```
124
125 #Regression Tree Model
126 modelRegTree <- rpart(Global_Sales ~ ., data=train)
127
```

j)

```

130 #Regression Tree Prediction and RMSE
131 predictionRegTree <- predict(modelRegTree, test)
132 RT_RMSE <- RMSE(predictionRegTree, test$Global_Sales)
133 RT_RMSE

```

Result = 0.0433108

k)

```

142 # Random Forest Prediction and RMSE
143 predictionRF <- predict(modelRF, test)
144 RF_RMSE <- RMSE(predictionRF, test$Global_Sales)

```

l) Import variables is ntree=500.

m)

```

142 # Random Forest Prediction and RMSE
143 predictionRF <- predict(modelRF, test)
144 RF_RMSE <- RMSE(predictionRF, test$Global_Sales)
145 RF_RMSE

```

Result = 0.03287746

n)

Regression RMSE = 0.03693007

Ridge RMSE = 0.4610888

Lasso RMSE = 0.4647719

Regression Tree RMSE = 0.0433108

Random Forest RMSE = 0.03287746

Best -> Random Forest

Worst -> Lasso

Here's from best to worst:

Random Forest, Regression, Regression Tree, Ridge, Lasso

Conclusion

In general, we achieved good results from the models we obtained. The best models were Random Forest, Regression, Regression Tree.