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 wasnot be 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)

```r

# Libraries
library(glmnet)
library(mice)
library(caret)
library(tidyverse)
library(rpart)
library(randomForest)

#########################################################

# import data
data <- read.csv("C:/Users/talha/Desktop/armut/6/FinalDataset.csv", sep=";")

##########################################################

#preprocessing data
minMaxNorm <- function(x){

  return ((x-min(x)) / (max(x) - min(x)))
}

data$X=NULL

data$Global_Sales <- gsub(",", ".", data$Global_Sales)
data$Global_Sales <- as.double(data$Global_Sales)
data$Platform <- as.factor(data$Platform)
data$Genre <- as.factor(data$Genre)
data$Rating <- as.factor(data$Rating)
data$PublisherR <- as.factor(data$PublisherR)

data$Critic_Score <- minMaxNorm(data$Critic_Score)
data$Critic_Count <- minMaxNorm(data$Critic_Count)
data$Global_Sales <- minMaxNorm(data$Global_Sales)

data <- sapply(data, unclass)

#data info
cor(data)
pairs(na.omit(data))
md.pattern(data)
```

b)

```r
# train data and test data
train <- as.data.frame(data[1:1000,])
test <- as.data.frame(data[1001:nrow(data),])

names(train)
names(test)

```

c)

```r
#Regression Model
modelRegression <- lm(Global_Sales ~ ., data = train)
# coefficients, adjusted R square and F statistic of Regression the model
summary(modelRegression)
```

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)

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

Result = 0.03693007

f)

```
#Determine the lambda parameter using cross-validation
lambdas = 10^seq(3,-2,by=-.01)
modelRidgeCV <- cv.glmnet(trainRidgeLasso_x,trainRidgeLasso_y, alpha = 0, lambda = lambdas, nfolds = 10)
modelRidgeCV$lambda.min

#Ridge Model
modelRidge <- glmnet(trainRidgeLasso_x,trainRidgeLasso_y, alpha = 0, lambda = modelRidgeCV$lambda.min)
summary(modelRidge)

########################################################

#Determine the lambda parameter using cross-validation
modelLassoCV <- cv.glmnet(trainRidgeLasso_x,trainRidgeLasso_y, alpha = 1, lambda = lambdas, nfolds = 3)
modelLassoCV$lambda.min

#Lasso Model
modelLasso <- glmnet(trainRidgeLasso_x,trainRidgeLasso_y, alpha = 1, lambda = modelLassoCV$lambda.min)
summary(modelLasso)
```

g) The regression model worked more accurately.

h)

```
# Ridge and Lasso Prediction and RMSE
predictionRidge <- predict(modelRidge, testRidgeLasso_x)
predictionLasso <- predict(modelLasso, testRidgeLasso_x)

Ridge_RMSE <- RMSE(predictionRidge, testRidgeLasso_y)
Lasso_RMSE <- RMSE(predictionLasso, testRidgeLasso_y)

Ridge_RMSE
Lasso_RMSE
```

Result Ridge = 0.4610888

Result Lasso = 0.4647719

i)

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

j)

```
#Regression Tree Prediction and RMSE
predictionRegTree <- predict(modelRegTree, test)
RT_RMSE <- RMSE(predictionRegTree, test$Global_Sales)
RT_RMSE
```

Result = 0.0433108

k)

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

l) Import variables is ntree=500.

m)

```
# Random Forest Prediction and RMSE
predictionRF <- predict(modelRF, test)
RF_RMSE <- RMSE(predictionRF, test$Global_Sales)
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.