Configuration Manual

MSc Research Project

Data Analytics

Rahul Gupta

Student ID: x18131115

School of Computing

National College of Ireland

Supervisor: Dr. Cristina Muntean

National College of Ireland
Project Submission Sheet – 2019/2020
School of Computing

Student Name:	Rahul Gupta
Student ID:	18131115
Programme:	MSc Data Analytics
Year:	2019/20
Module:	Research Project
Lecturer:	Dr. Cristina Muntean
Submission Due Date:	12/12/2019
Project Title:	Prediction of Major Factors affecting Fans Attendance for the Teams of Major League Baseball
Word Count	1380 **Page Count: 10**

I hereby certify that the information contained in this (my submission) is information pertaining to my own individual work that I conducted for this project. All information other than my own contribution is fully and appropriately referenced and listed in the relevant bibliography section. I assert that I have not referred to any work(s) other than those listed. I also include my TurnItIn report with this submission.

ALL materials used must be referenced in the bibliography section. Students are encouraged to use the Harvard Referencing Standard supplied by the Library. To use other author's written or electronic work is an act of plagiarism and may result in disciplinary action. Students may be required to undergo a viva (oral examination) if there is suspicion about the validity of their submitted work.

Signature:	Rahul Gupta
Date:	December 12, 2019

PLEASE READ THE FOLLOWING INSTRUCTIONS:

1. Please attach a completed copy of this sheet to each project (including multiple copies).
2. **You must ensure that you retain a HARD COPY of ALL projects**, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer. Please do not bind projects or place in covers unless specifically requested.
3. Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

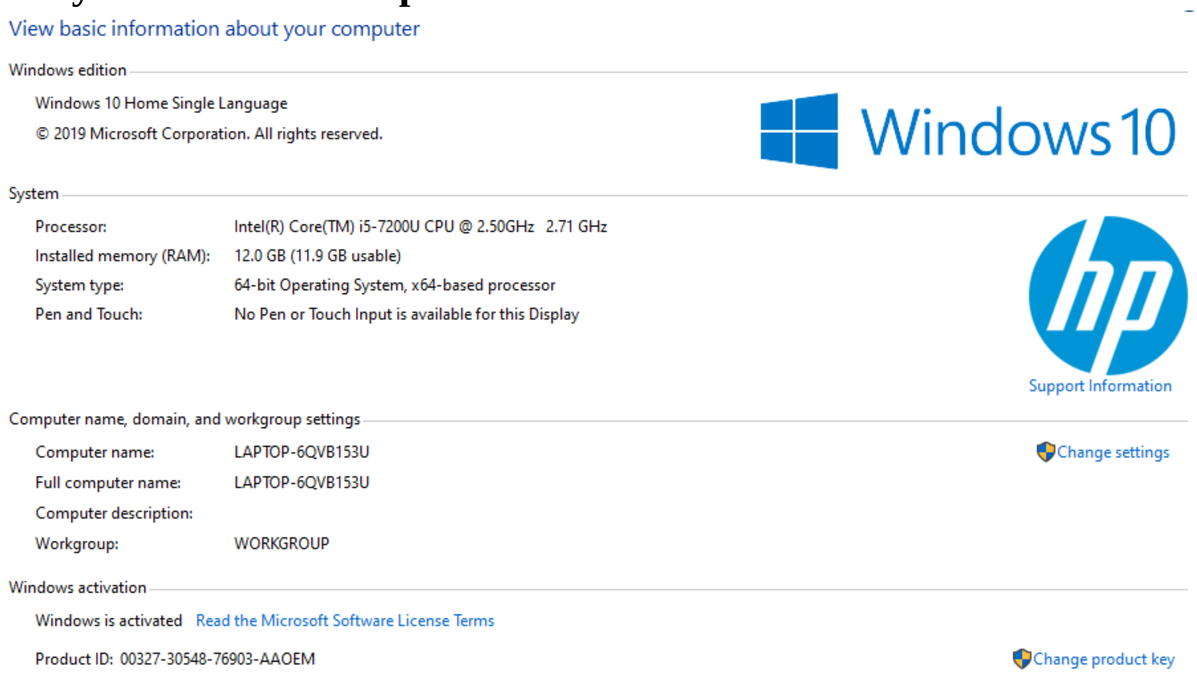
Configuration Manual

Prediction of Major Factors affecting Fans Attendance for the Teams of Major League Baseball

1. Introduction

A Configuration Manual describes the software and hardware requirements for the implementation of research. It also involves screenshots showing step-by-step process for the implementation of research. The title of this research is **“Prediction of Major Factors affecting Fans Attendance for the Teams of Major League Baseball”**. The aim of this research was to determine the various in-game factors that affect the fans attendance figures. For this purpose, 4 machine learning algorithms viz. Multiple Linear Regression (MLR), Random Forest (RF), Artificial Neural Networks (ANN) and Support Vector Regression (SVR) were applied on the extracted data.
This manual details the necessary configuration that were made for the project. It also details the hardware and software configuration. It also details the step-by-step process undertaken to achieve the desired results.

2. System/Hardware Specifications


View basic information about your computer

Windows edition

Windows 10 Home Single Language
© 2019 Microsoft Corporation. All rights reserved.

System

Processor: Intel(R) Core(TM) i5-7200U CPU @ 2.50GHz 2.71 GHz
Installed memory (RAM): 12.0 GB (11.9 GB usable)
System type: 64-bit Operating System, x64-based processor
Pen and Touch: No Pen or Touch Input is available for this Display

Computer name, domain, and workgroup settings

Computer name: LAPTOP-6QVB153U
Full computer name: LAPTOP-6QVB153U
Computer description:
Workgroup: WORKGROUP

Windows activation

Windows is activated Read the Microsoft Software License Terms

Product ID: 00327-30548-76903-AAOEM

Figure 1 System specifications

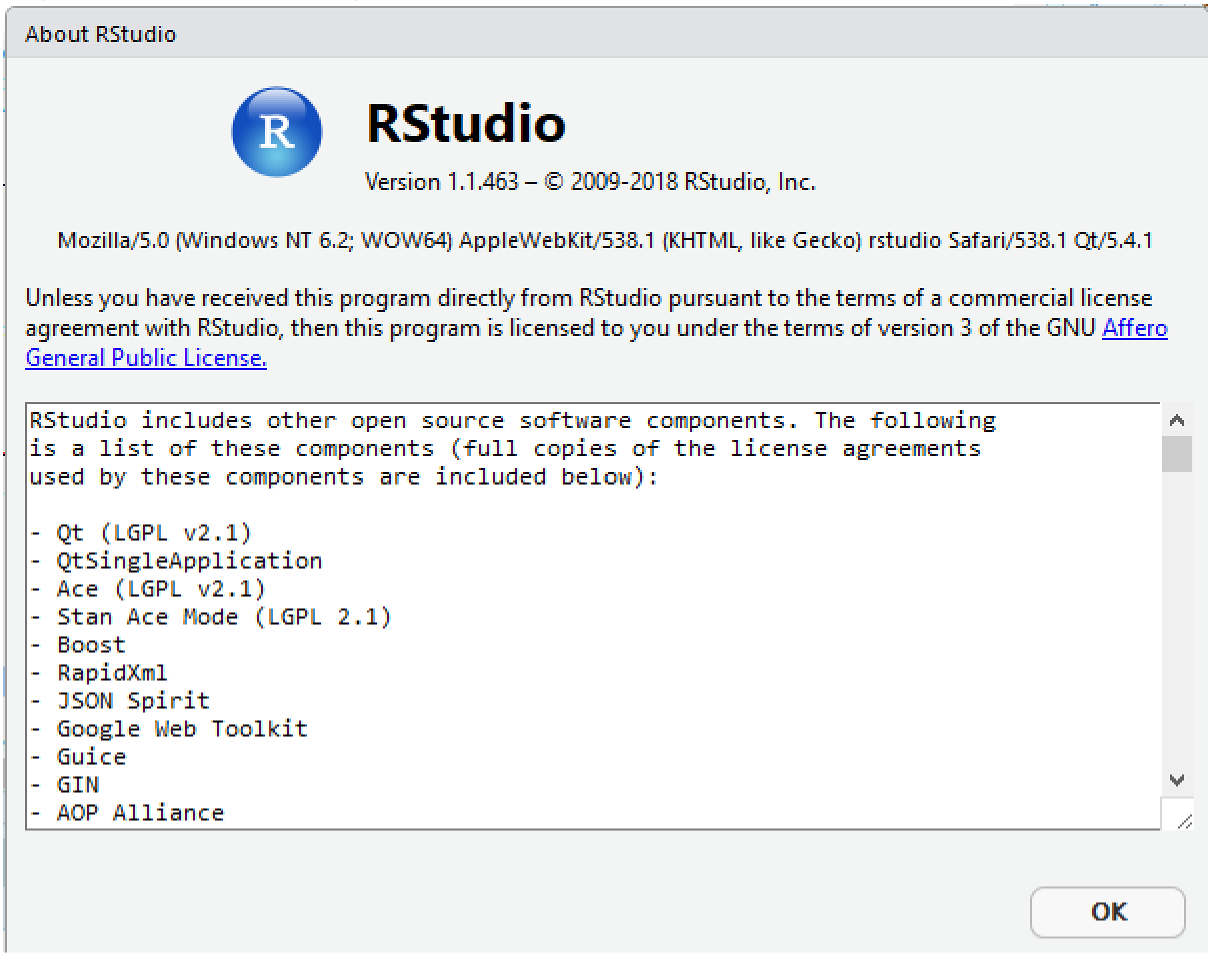
Figure 1 shows the specifications of the hardware that was used to for implementation of this research.

a) Processor: Intel(R) Core(TM) i5-7200 CPU @ 2.50GHz 2.71GHz
b) Operating System: Windows 10
c) RAM: 12 GB
d) System Type: 64-bit Operating System, x64-based processor

3. Software Specifications

a) **R**: R programming language was used for the implementation of the entire project. All the activities like data extraction, data cleaning, data preprocessing, data merging, data transformation, model creation and evaluation of the models was done using R. R can be downloaded from the following link: https://cran.r-project.org/bin/windows/base/

b) **R studio:** R studio was used for the execution of the R code. Version of the R studio is 3.5.2. It can be downloaded from the following link: https://www.rstudio.com/products/rstudio/download/



About RStudio

RStudio

Version 1.1.463 – © 2009-2018 RStudio, Inc.

Mozilla/5.0 (Windows NT 6.2; WOW64) AppleWebKit/538.1 (KHTML, like Gecko) rstudio Safari/538.1 Qt/5.4.1

Unless you have received this program directly from RStudio pursuant to the terms of a commercial license agreement with RStudio, then this program is licensed to you under the terms of version 3 of the GNU Affero General Public License.

```
RStudio includes other open source software components. The following
is a list of these components (full copies of the license agreements
used by these components are included below):

- Qt (LGPL v2.1)
- QtSingleApplication
- Ace (LGPL v2.1)
- Stan Ace Mode (LGPL 2.1)
- Boost
- RapidXml
- JSON Spirit
- Google Web Toolkit
- Guice
- GIN
- AOP Alliance
```

Figure 2 R studio version

c) **Tableau**: Tableau 2018.2 was used for the visualization of the obtained results. All the results were consolidated to form a single file. The average value of RMSE, MAD and MAPE were compared by visualizing them on the same graph.

d) **Microsoft Excel 2016**: Excel 2016 was used for the extraction of the data in the structured format, which was then converted into .csv file.

4. Stepwise Implementation

a) Download the R language and R studio.

b) Installation of R studio 3.5.2

c) Installation of the required packages or libraries.

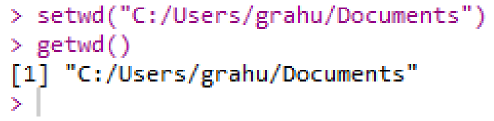
d) Setting up the working directory.

e) Execute the .R file containing R code, to obtain the desired output.

5. Data Preparation

5.1 Set the working directory

Firstly, set the working directory. The working directory was set to “C:\Users\grahu\Documents”. Figure 3 shows how the working directory was set in R studio for the project.

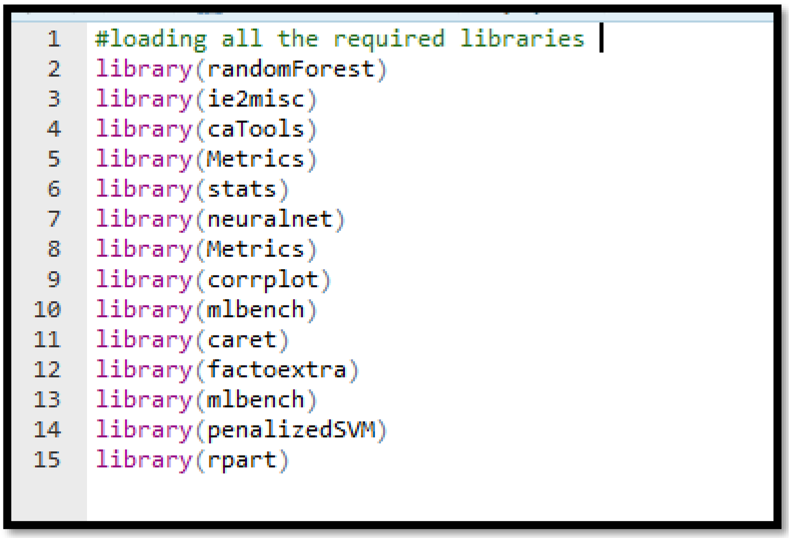


```
> setwd("C:/Users/grahu/Documents")
> getwd()
[1] "C:/Users/grahu/Documents"
> 
```

Figure 3

5.2 Installation of required libraries

All the required packages were installed, and then the library was run before executing any R script in the R studio. Figure 4 shows the list of libraries that were installed for this research project:



```
#loading all the required libraries
library(randomForest)
library(ie2misc)
library(caTools)
library(Metrics)
library(stats)
library(neuralnet)
library(Metrics)
library(corrplot)
library(mlbench)
library(caret)
library(factoextra)
library(mlbench)
library(penalizedSVM)
library(rpart)
```

Figure 4

5.3 Data Extraction

The data was extracted in the structured format. The stats were in .csv file. All the data was loaded in the R studio along with the execution of the code. 3 sets of data were downloaded for all the 30 teams. These sets are:

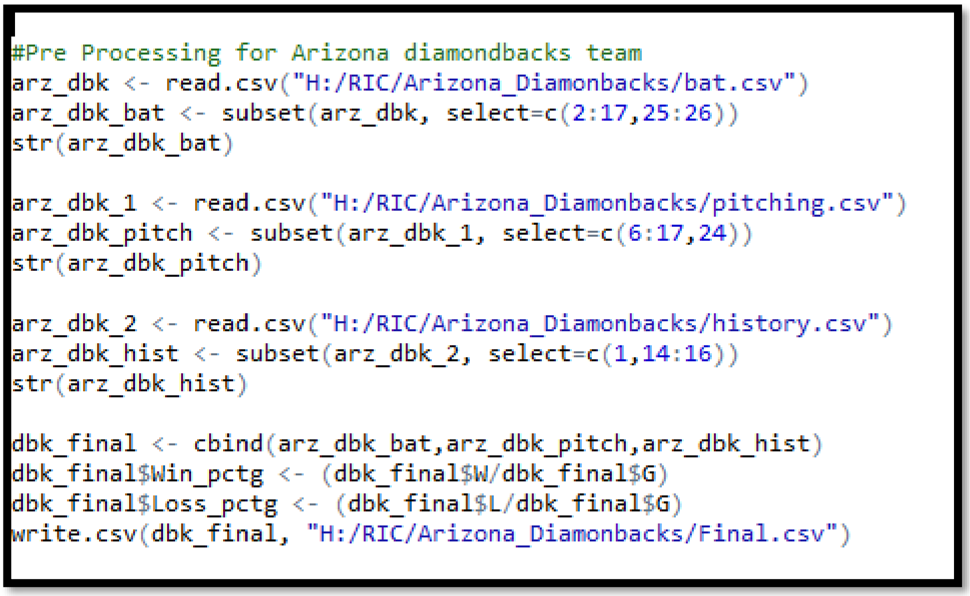
a) Bat.csv contains the statistics related to the batting aspect of the game. The csv contains records of the team since its inception in the league.

b) Pitch.csv contains the statistics related to the pitching aspect of the game. The csv contains records of the team since its inception in the league.

c) History.csv contains the general history along with the attendance figures for the team. The csv contains records of the team since its inception in the league.

5.4 Data Merging

The 3 .csv files were merged to form a single file, so that further preprocessing can be done on the dataset. Figure 5 show the snippet of R code for the same:

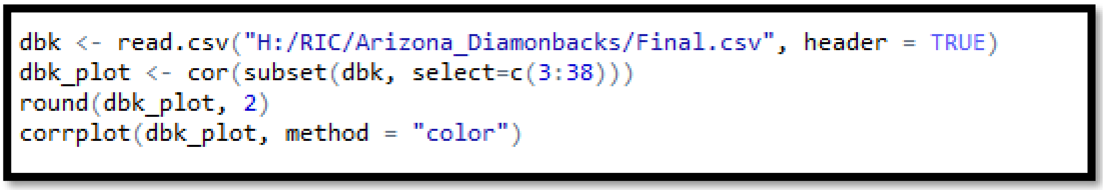


```
#Pre Processing for Arizona diamondbacks team
arz_dbk <- read.csv("H:/RIC/Arizona_Diamonbacks/bat.csv")
arz_dbk_bat <- subset(arz_dbk, select=c(2:17,25:26))
str(arz_dbk_bat)

arz_dbk_1 <- read.csv("H:/RIC/Arizona_Diamonbacks/pitching.csv")
arz_dbk_pitch <- subset(arz_dbk_1, select=c(6:17,24))
str(arz_dbk_pitch)

arz_dbk_2 <- read.csv("H:/RIC/Arizona_Diamonbacks/history.csv")
arz_dbk_hist <- subset(arz_dbk_2, select=c(1,14:16))
str(arz_dbk_hist)

dbk_final <- cbind(arz_dbk_bat,arz_dbk_pitch,arz_dbk_hist)
dbk_final$Win_pctg <- (dbk_final$W/dbk_final$G)
dbk_final$Loss_pctg <- (dbk_final$L/dbk_final$G)
write.csv(dbk_final, "H:/RIC/Arizona_Diamonbacks/Final.csv")
```

Figure 5

5.5 Data Preprocessing

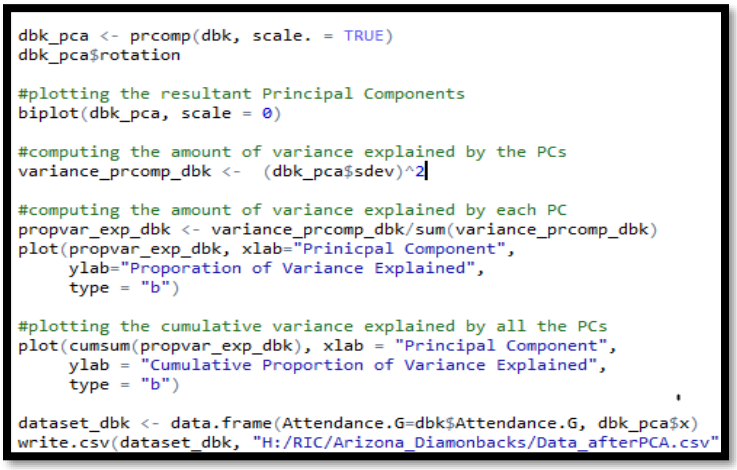
After the merging of data, preprocessing was done on the following data to find out the underlying trends. This was done for all the 30 teams. Correlation plot was used to find out the multicolinearity among the variables. Figure 6 shows the snippet of R code for the same.



```
dbk <- read.csv("H:/RIC/Arizona_Diamonbacks/Final.csv", header = TRUE)
dbk_plot <- cor(subset(dbk, select=c(3:38)))
round(dbk_plot, 2)
corrplot(dbk_plot, method = "color")
```

Figure 6

After drawing the correlation plot, the highly correlated variables were eliminated using principal component analysis. Figure 7 shows the snippet of the code for PCA.

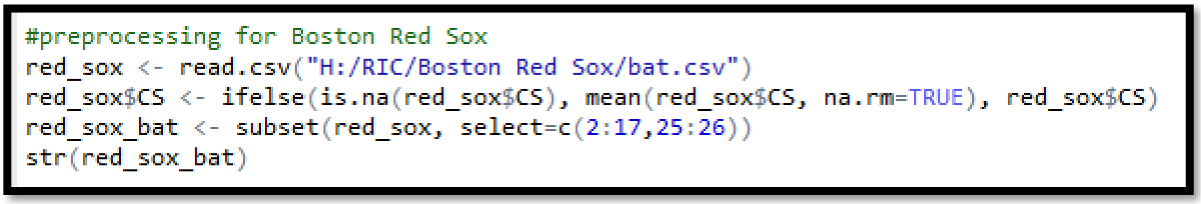


```
dbk_pca <- prcomp(dbk, scale. = TRUE)
dbk_pca$rotation

#plotting the resultant Principal Components
biplot(dbk_pca, scale = 0)

#computing the amount of variance explained by the PCs
variance_prcomp_dbk <-  (dbk_pca$sdev)^2

#computing the amount of variance explained by each PC
propvar_exp_dbk <- variance_prcomp_dbk/sum(variance_prcomp_dbk)
plot(propvar_exp_dbk, xlab="Prinicpal Component",
     ylab="Proporation of Variance Explained",
     type = "b")

#plotting the cumulative variance explained by all the PCs
plot(cumsum(propvar_exp_dbk), xlab = "Principal Component",
     ylab = "Cumulative Proportion of Variance Explained",
     type = "b")

dataset_dbk <- data.frame(Attendance.G=dbk$Attendance.G, dbk_pca$x)
write.csv(dataset_dbk, "H:/RIC/Arizona_Diamonbacks/Data_afterPCA.csv"
```

Figure 7

5.6 Data Transformation

The records were in the desired state. Only 1 transformation was done. The missing values were imputated with the mean values of that particular column. Figure 8 shows the snippet of the code for the same.

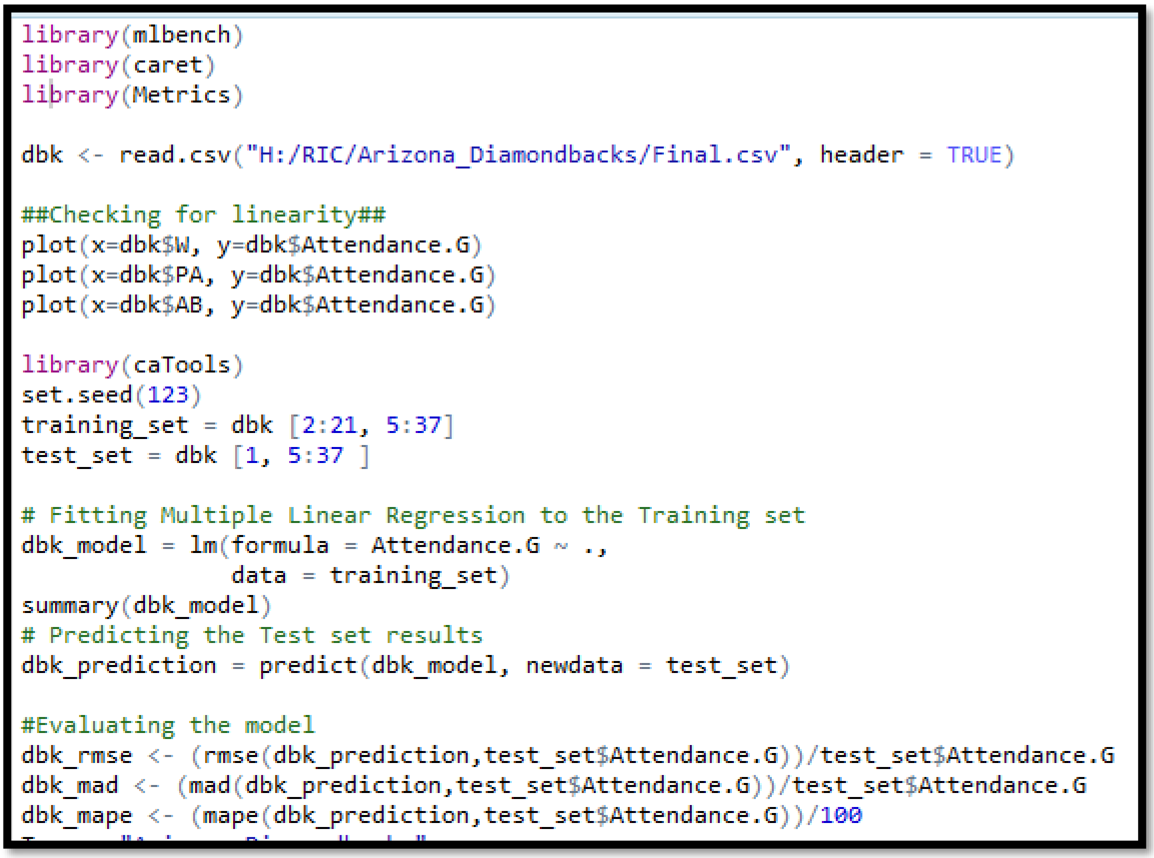


```
#preprocessing for Boston Red Sox
red_sox <- read.csv("H:/RIC/Boston Red Sox/bat.csv")
red_sox$CS <- ifelse(is.na(red_sox$CS), mean(red_sox$CS, na.rm=TRUE), red_sox$CS)
red_sox_bat <- subset(red_sox, select=c(2:17,25:26))
str(red_sox_bat)
```

Figure 8

6. Implementation of models

6.1 Implementation of Multiple Linear Regression model



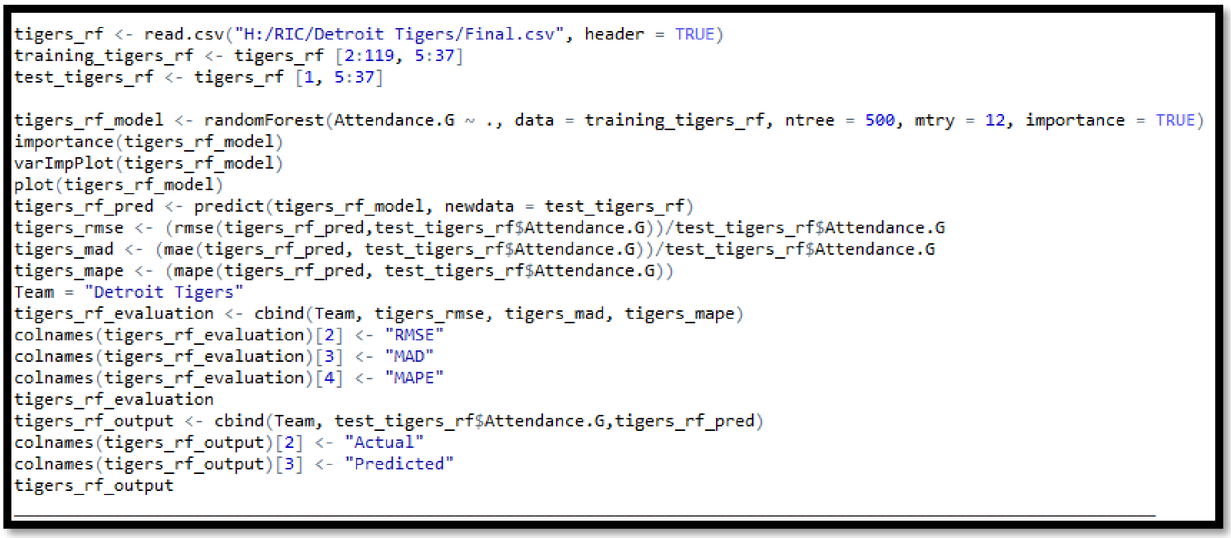
```
library(mlbench)
library(caret)
library(Metrics)

dbk <- read.csv("H:/RIC/Arizona_Diamondbacks/Final.csv", header = TRUE)

##Checking for linearity##
plot(x=dbk$W, y=dbk$Attendance.G)
plot(x=dbk$PA, y=dbk$Attendance.G)
plot(x=dbk$AB, y=dbk$Attendance.G)

library(caTools)
set.seed(123)
training_set = dbk [2:21, 5:37]
test_set = dbk [1, 5:37 ]

# Fitting Multiple Linear Regression to the Training set
dbk_model = lm(formula = Attendance.G ~ .,
               data = training_set)
summary(dbk_model)
# Predicting the Test set results
dbk_prediction = predict(dbk_model, newdata = test_set)

#Evaluating the model
dbk_rmse <- (rmse(dbk_prediction,test_set$Attendance.G))/test_set$Attendance.G
dbk_mad <- (mad(dbk_prediction,test_set$Attendance.G))/test_set$Attendance.G
dbk_mape <- (mape(dbk_prediction,test_set$Attendance.G))/100
```

Figure 9

Figure 9 shows the R code for the execution of Multiple Linear Regression model. This code is for only 1 team. The data was first checked for linearity. For regression purposes, the data must be linear to avoid biased results. The data was divided into training and testing data. The record for 2019 was taken as the test data, whereas rest of the data was used as training data. The results were then evaluated using RMSE, MAD and MAPE. The code for the same is present in the last 3 lines of the code.

6.2 Implementation of Random Forest model

Figure 9 shows the R code for the execution of Random Forest model. This code is for only 1 team. The data was divided into training and testing data. The record for 2019 was taken as the test data, whereas rest of the data was used as training data. Figure 10 shows the R code for the implementation of Random Forest model.

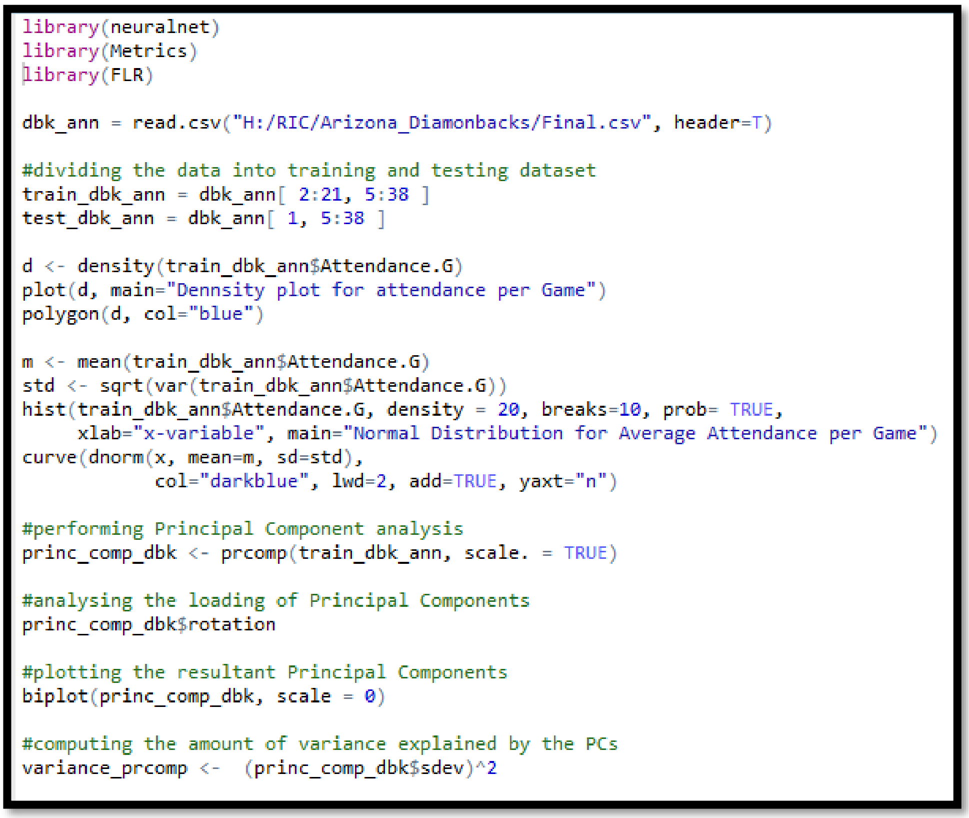


```
tigers_rf <- read.csv("H:/RIC/Detroit Tigers/Final.csv", header = TRUE)
training_tigers_rf <- tigers_rf [2:119, 5:37]
test_tigers_rf <- tigers_rf [1, 5:37]

tigers_rf_model <- randomForest(Attendance.G ~ ., data = training_tigers_rf, ntree = 500, mtry = 12, importance = TRUE)
importance(tigers_rf_model)
varImpPlot(tigers_rf_model)
plot(tigers_rf_model)
tigers_rf_pred <- predict(tigers_rf_model, newdata = test_tigers_rf)
tigers_rmse <- (rmse(tigers_rf_pred,test_tigers_rf$Attendance.G))/test_tigers_rf$Attendance.G
tigers_mad <- (mae(tigers_rf_pred, test_tigers_rf$Attendance.G))/test_tigers_rf$Attendance.G
tigers_mape <- (mape(tigers_rf_pred, test_tigers_rf$Attendance.G))
Team = "Detroit Tigers"
tigers_rf_evaluation <- cbind(Team, tigers_rmse, tigers_mad, tigers_mape)
colnames(tigers_rf_evaluation)[2] <- "RMSE"
colnames(tigers_rf_evaluation)[3] <- "MAD"
colnames(tigers_rf_evaluation)[4] <- "MAPE"
tigers_rf_evaluation
tigers_rf_output <- cbind(Team, test_tigers_rf$Attendance.G,tigers_rf_pred)
colnames(tigers_rf_output)[2] <- "Actual"
colnames(tigers_rf_output)[3] <- "Predicted"
tigers_rf_output
```

Figure 10

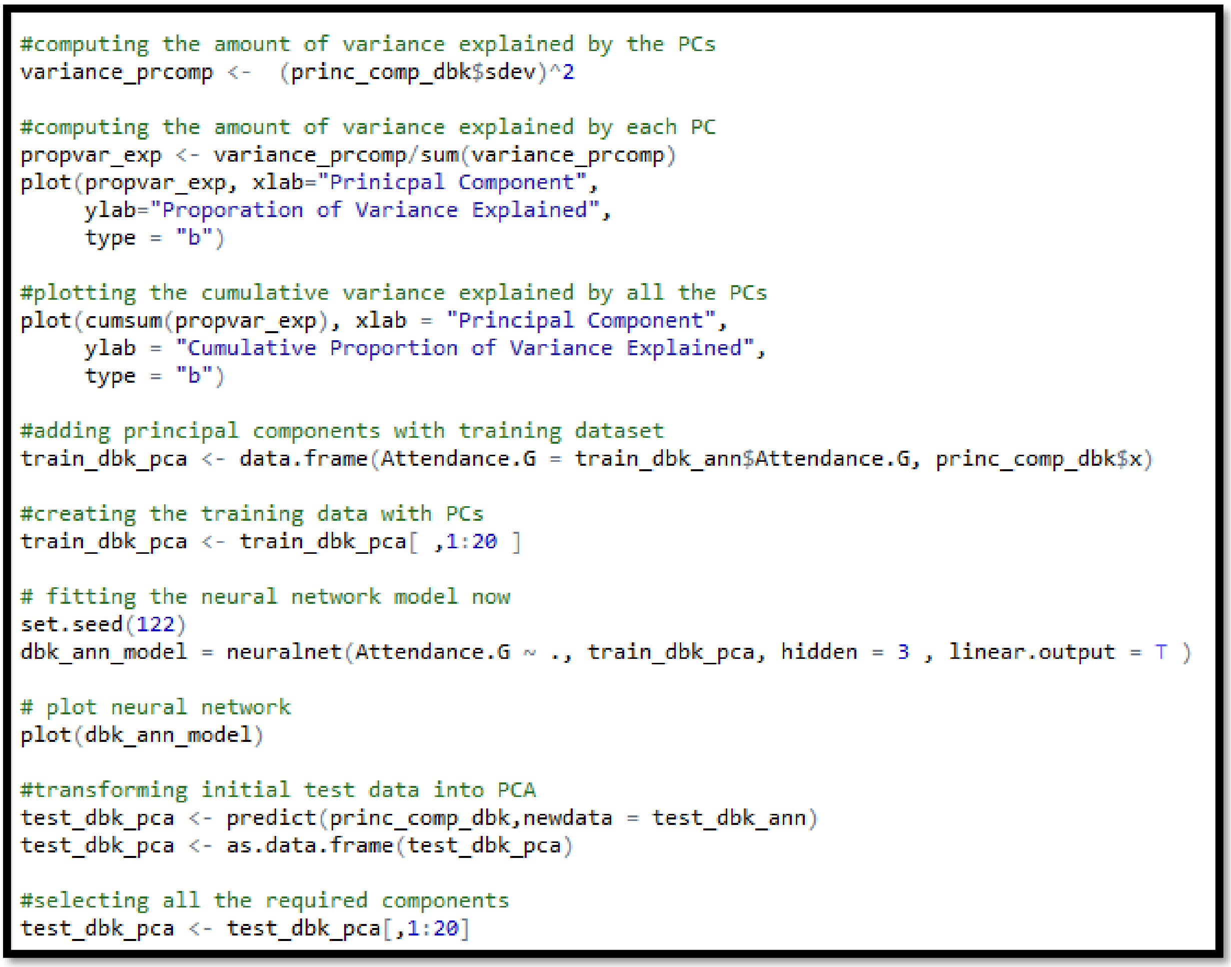
6.3 Implementation of Artificial Neural Network model

Figure 11(a), 11(b) and 11(c) shows the R code for the execution of Artificial Neural Network model. This code is for only 1 team. The data was divided into training and testing data. The record for 2019 was taken as the test data, whereas rest of the data was used as training data.



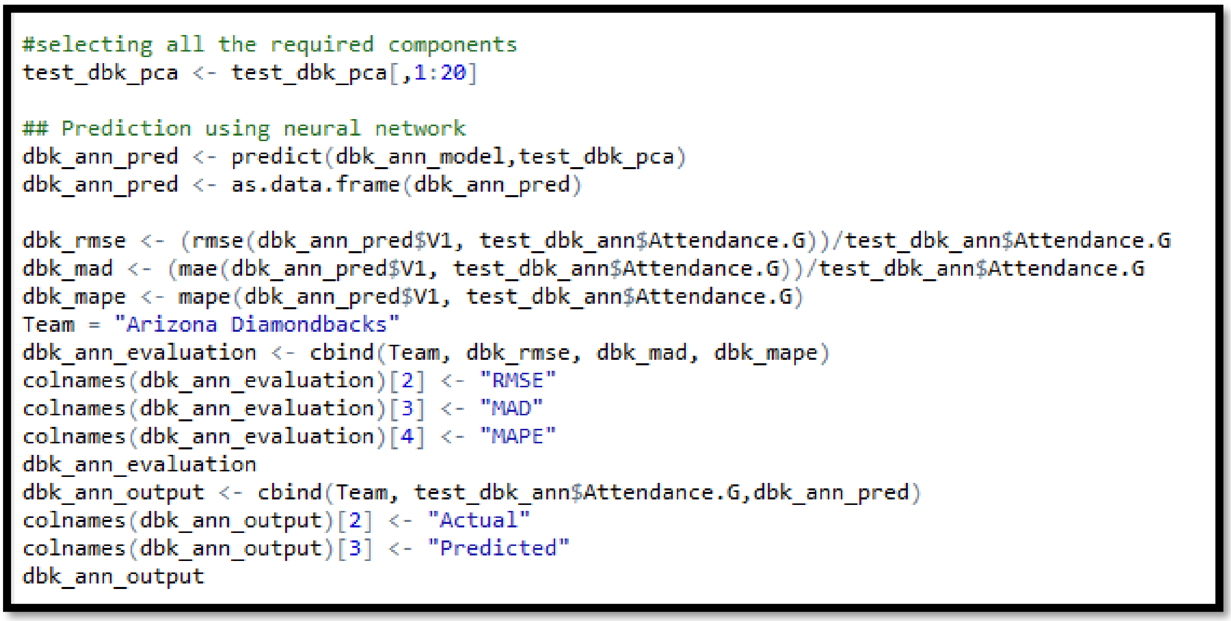
```
library(neuralnet)
library(Metrics)
library(FLR)

dbk_ann = read.csv("H:/RIC/Arizona_Diamonbacks/Final.csv", header=T)

#dividing the data into training and testing dataset
train_dbk_ann = dbk_ann[ 2:21, 5:38 ]
test_dbk_ann = dbk_ann[ 1, 5:38 ]

d <- density(train_dbk_ann$Attendance.G)
plot(d, main="Dennsity plot for attendance per Game")
polygon(d, col="blue")

m <- mean(train_dbk_ann$Attendance.G)
std <- sqrt(var(train_dbk_ann$Attendance.G))
hist(train_dbk_ann$Attendance.G, density = 20, breaks=10, prob= TRUE,
     xlab="x-variable", main="Normal Distribution for Average Attendance per Game")
curve(dnorm(x, mean=m, sd=std),
            col="darkblue", lwd=2, add=TRUE, yaxt="n")

#performing Principal Component analysis
princ_comp_dbk <- prcomp(train_dbk_ann, scale. = TRUE)

#analysing the loading of Principal Components
princ_comp_dbk$rotation

#plotting the resultant Principal Components
biplot(princ_comp_dbk, scale = 0)

#computing the amount of variance explained by the PCs
variance_prcomp <-  (princ_comp_dbk$sdev)^2
```

Figure 11(a)



```
#computing the amount of variance explained by the PCs
variance_prcomp <-  (princ_comp_dbk$sdev)^2

#computing the amount of variance explained by each PC
propvar_exp <- variance_prcomp/sum(variance_prcomp)
plot(propvar_exp, xlab="Prinicpal Component",
     ylab="Proporation of Variance Explained",
     type = "b")

#plotting the cumulative variance explained by all the PCs
plot(cumsum(propvar_exp), xlab = "Principal Component",
     ylab = "Cumulative Proportion of Variance Explained",
     type = "b")

#adding principal components with training dataset
train_dbk_pca <- data.frame(Attendance.G = train_dbk_ann$Attendance.G, princ_comp_dbk$x)

#creating the training data with PCs
train_dbk_pca <- train_dbk_pca[ ,1:20 ]

# fitting the neural network model now
set.seed(122)
dbk_ann_model = neuralnet(Attendance.G ~ ., train_dbk_pca, hidden = 3 , linear.output = T )

# plot neural network
plot(dbk_ann_model)

#transforming initial test data into PCA
test_dbk_pca <- predict(princ_comp_dbk,newdata = test_dbk_ann)
test_dbk_pca <- as.data.frame(test_dbk_pca)

#selecting all the required components
test_dbk_pca <- test_dbk_pca[,1:20]
```

Figure 11(b)

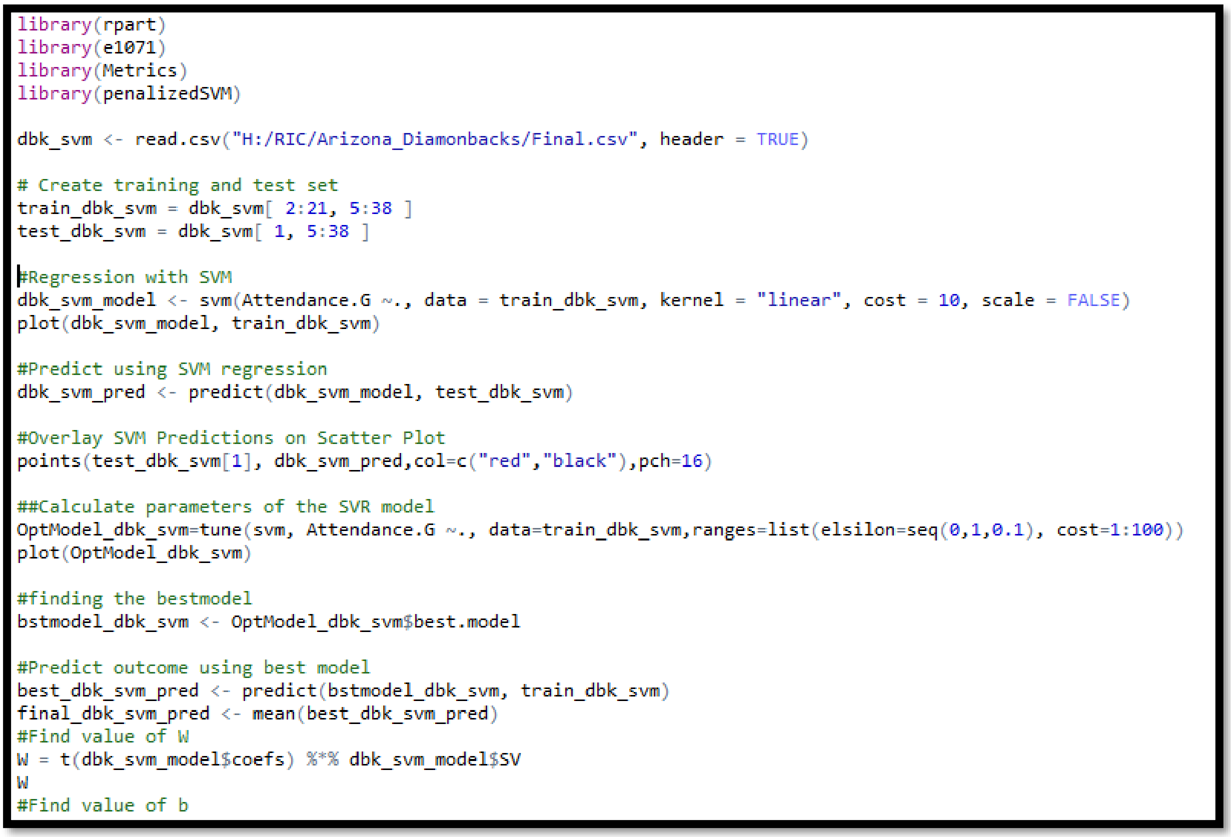


```
#selecting all the required components
test_dbk_pca <- test_dbk_pca[,1:20]

## Prediction using neural network
dbk_ann_pred <- predict(dbk_ann_model,test_dbk_pca)
dbk_ann_pred <- as.data.frame(dbk_ann_pred)

dbk_rmse <- (rmse(dbk_ann_pred$V1, test_dbk_ann$Attendance.G))/test_dbk_ann$Attendance.G
dbk_mad <- (mae(dbk_ann_pred$V1, test_dbk_ann$Attendance.G))/test_dbk_ann$Attendance.G
dbk_mape <- mape(dbk_ann_pred$V1, test_dbk_ann$Attendance.G)
Team = "Arizona Diamondbacks"
dbk_ann_evaluation <- cbind(Team, dbk_rmse, dbk_mad, dbk_mape)
colnames(dbk_ann_evaluation)[2] <- "RMSE"
colnames(dbk_ann_evaluation)[3] <- "MAD"
colnames(dbk_ann_evaluation)[4] <- "MAPE"
dbk_ann_evaluation
dbk_ann_output <- cbind(Team, test_dbk_ann$Attendance.G,dbk_ann_pred)
colnames(dbk_ann_output)[2] <- "Actual"
colnames(dbk_ann_output)[3] <- "Predicted"
dbk_ann_output
```

Figure 11(c)

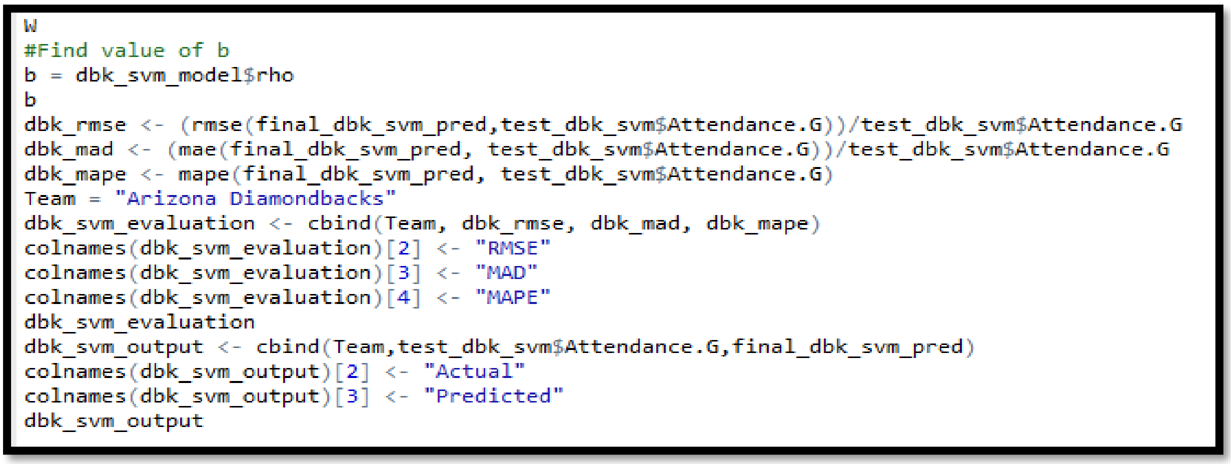
6.4 Implementation of Support Vector Regression model

Figure 12(a) and 12(b) shows the R code for the execution of Support Vector Regression model. This code is for only 1 team. The data was divided into training and testing data. The record for 2019 was taken as the test data, whereas rest of the data was used as training data.



```r
library(rpart)
library(e1071)
library(Metrics)
library(penalizedSVM)

dbk_svm <- read.csv("H:/RIC/Arizona_Diamonbacks/Final.csv", header = TRUE)

# Create training and test set
train_dbk_svm = dbk_svm[ 2:21, 5:38 ]
test_dbk_svm = dbk_svm[ 1, 5:38 ]

#Regression with SVM
dbk_svm_model <- svm(Attendance.G ~., data = train_dbk_svm, kernel = "linear", cost = 10, scale = FALSE)
plot(dbk_svm_model, train_dbk_svm)

#Predict using SVM regression
dbk_svm_pred <- predict(dbk_svm_model, test_dbk_svm)

#Overlay SVM Predictions on Scatter Plot
points(test_dbk_svm[1], dbk_svm_pred,col=c("red","black"),pch=16)

##Calculate parameters of the SVR model
OptModel_dbk_svm=tune(svm, Attendance.G ~., data=train_dbk_svm,ranges=list(elsilon=seq(0,1,0.1), cost=1:100))
plot(OptModel_dbk_svm)

#finding the bestmodel
bstmodel_dbk_svm <- OptModel_dbk_svm$best.model

#Predict outcome using best model
best_dbk_svm_pred <- predict(bstmodel_dbk_svm, train_dbk_svm)
final_dbk_svm_pred <- mean(best_dbk_svm_pred)
#Find value of W
W = t(dbk_svm_model$coefs) %*% dbk_svm_model$SV
W
#Find value of b
```

Figure 12(a)



```r
W
#Find value of b
b = dbk_svm_model$rho
b
dbk_rmse <- (rmse(final_dbk_svm_pred,test_dbk_svm$Attendance.G))/test_dbk_svm$Attendance.G
dbk_mad <- (mae(final_dbk_svm_pred, test_dbk_svm$Attendance.G))/test_dbk_svm$Attendance.G
dbk_mape <- mape(final_dbk_svm_pred, test_dbk_svm$Attendance.G)
Team = "Arizona Diamondbacks"
dbk_svm_evaluation <- cbind(Team, dbk_rmse, dbk_mad, dbk_mape)
colnames(dbk_svm_evaluation)[2] <- "RMSE"
colnames(dbk_svm_evaluation)[3] <- "MAD"
colnames(dbk_svm_evaluation)[4] <- "MAPE"
dbk_svm_evaluation
dbk_svm_output <- cbind(Team,test_dbk_svm$Attendance.G,final_dbk_svm_pred)
colnames(dbk_svm_output)[2] <- "Actual"
colnames(dbk_svm_output)[3] <- "Predicted"
dbk_svm_output
```

Figure 12(b)

References

1) https://www.stackoverflow.com
2) https://www.analyticsvidhya.com
3) https://cran.r-project.org/web/packages/e1071/vignettes/svmdoc.pdf