##Set appropriate working directory to computer

setwd("C:/Users/Vignesh/Downloads/BANA 7046 and 7047-Data Mining")

#Libraries to Load, importing file and seeing general properties

library(readxl)

library(MASS) #data source

library(corrplot) #for correlation

library(lattice) #for correlation

library(knitr) #create tables

library(tidyverse) #use the pipe operator

library(ggplot2) #making graphs

Moneyball <- read\_excel("C:/Users/Vignesh/Downloads/Vignesh Important Documents/Moneyball.xlsx")

View(Moneyball)

dim(Moneyball)

names(Moneyball)

Moneyball$Playoffs <- as.factor(Moneyball$Playoffs)

#Change NA for OOBP and OSLG to median and change NA for Rank Season and Rank Playoffs to 0

Moneyball$OOBP[which(is.na(Moneyball$OOBP))] <- median(Moneyball$OOBP,na.rm = TRUE)

Moneyball$OSLG[which(is.na(Moneyball$`OSLG`))] <- median(Moneyball$OSLG,na.rm = TRUE)

Moneyball$RankSeason[which(is.na(Moneyball$RankSeason))] <- 0

Moneyball$RankPlayoffs[which(is.na(Moneyball$RankPlayoffs))] <- 0

str(Moneyball)

summary(Moneyball)

View(Moneyball)

#Arranging data and creating variable columns

arrange(Moneyball, desc(W))

arrange(Moneyball, desc(RS))

arrange(Moneyball, desc(RA))

arrange(Moneyball, desc(OBP))

arrange(Moneyball, desc(SLG))

arrange(Moneyball, desc(BA))

Moneyball <- Moneyball %>%

mutate(RunDiff = RS - RA,

OBPDiff = OBP - OOBP,

SLGDiff = SLG - OSLG)

dim(Moneyball)

names(Moneyball)

#Rearrange rows of Moneyball to have similar rows next to each other

Moneyball <- Moneyball[c(1:5,16,6:7,14,17,8,15,18,9:13)]

arrange(Moneyball, desc(RunDiff))

arrange(Moneyball, desc(OBPDiff))

arrange(Moneyball, desc(SLGDiff))

#Set seed to split sample in training and testing data, 80-20 split

set.seed(05542052)

index <- sample(nrow(Moneyball), nrow(Moneyball)\*0.80)

Moneyball.train <- Moneyball[index,]

Moneyball.test <- Moneyball[-index,]

#Nature of Each variable

data.frame(Variable = names(Moneyball.train),

Type = sapply(Moneyball.train, class),

row.names = NULL) %>%

kable(format = "markdown")

#Basic Findings of Moneyball training data

dim(Moneyball.train)

names(Moneyball.train)

str(Moneyball.train)

head(Moneyball.train)

summary(Moneyball.train)

#Numeric Values:

Moneyball.numeric <- Moneyball.train[-c(1:2,15)]

#Pairwise Correlation for important numerical variables for all teams

design = Moneyball.numeric

cormat = cor(design)

cormat

image(cormat, col = terrain.colors(8))

#Correlation matrix

corrplot(cormat, type = "upper")

#Histograms of Numeric Variables

par(mfrow = c(4,4))

ggplot(Moneyball.train,aes(x = Year)) + geom\_histogram(binwidth = .25)

ggplot(Moneyball.train,aes(x = RS)) + geom\_histogram(binwidth = .25)

ggplot(Moneyball.train,aes(x = RA)) + geom\_histogram(binwidth = .25)

ggplot(Moneyball.train,aes(x = RunDiff)) + geom\_histogram(binwidth = .25)

ggplot(Moneyball.train,aes(x = W)) + geom\_histogram(binwidth = .25)

ggplot(Moneyball.train,aes(x = OBP)) + geom\_histogram(binwidth = .25)

ggplot(Moneyball.train,aes(x = OOBP)) + geom\_histogram(binwidth = .25)

ggplot(Moneyball.train,aes(x = OBPDiff)) + geom\_histogram(binwidth = .25)

ggplot(Moneyball.train,aes(x = SLG)) + geom\_histogram(binwidth = .25)

ggplot(Moneyball.train,aes(x = OSLG)) + geom\_histogram(binwidth = .25)

ggplot(Moneyball.train,aes(x = SLGDiff)) + geom\_histogram(binwidth = .25)

ggplot(Moneyball.train,aes(x = BA)) + geom\_histogram(binwidth = 0.25)

ggplot(Moneyball.train,aes(x = RankPlayoffs)) + geom\_histogram(binwidth = .25)

ggplot(Moneyball.train,aes(x = RankSeason)) + geom\_histogram(binwidth = .25)

ggplot(Moneyball.train, aes(x = G)) + geom\_histogram(binwidth = .25)

#Combining all histograms

attach(Moneyball.train)

par(mfrow = c(4,4))

hist(Year)

hist(RS)

hist(RA)

hist(RunDiff)

hist(W)

hist(OBP)

hist(OOBP)

hist(OBPDiff)

hist(SLG)

hist(OSLG)

hist(SLGDiff)

hist(BA)

hist(RankSeason)

hist(RankPlayoffs)

hist(G)

#GLM-Logistic

Moneyball.trainstats <- glm(Playoffs~RS+RA+W+BA+OBP+SLG+Year+G+OOBP+OSLG+Team+League, family = binomial(link = "logit"), Moneyball.train)

summary(Moneyball.trainstats)

Moneyball.trainstatsbestmodel <- glm(Playoffs ~ W + Year, family = binomial(link = "logit"), Moneyball.train)

summary(Moneyball.trainstatsbestmodel)

#Multiple Linear Regression

Moneyball.winsfactors <- lm(W~RS+RA+BA+OBP+SLG+OOBP+OSLG+Year+G+Playoffs+Team+League

, data = Moneyball.train)

summary(Moneyball.winsfactors)

Moneyball.winsfactorsclosetobestmodel <- glm(W~RS+RA+OBP+SLG+Playoffs, data = Moneyball.train)

summary(Moneyball.winsfactorsbestmodel)

#Stepwise BIC Logistic

moneyball.glm.step <- step(Moneyball.trainstats, k = log(nrow(Moneyball.train)), direction = c("both"))

#Stepwise BIC Linear

moneyball.lm.wins <- step(Moneyball.winsfactors, k = log(nrow(Moneyball.train)), direction = c("both"))

#Best Models and AUC/ROC Plots Logistic

Moneyballcost <- function(observed, predicted) {

weight1 = 5

weight0 = 1

c1 = (observed == 1) & (predicted == 0) #logical vector - true if actual 1 but predict 0

c0 = (observed == 0) & (predicted == 1) #logical vecotr - true if actual 0 but predict 1

return(mean(weight1 \* c1 + weight0 \* c0))

}

#In Sample Confusion Matrix

prob.glm0.insample <- predict(moneyball.glm.step, Moneyball.train, type = "Playoffs")

predicted.glm0.insample <- prob.glm0.insample > 0.2

predicted.glm0.insample <- as.numeric(predicted.glm0.insample)

table(Moneyball.train$Playoffs, predicted.glm0.insample, dnn = c("Truth", "Predicted"))

mean(ifelse(Moneyball.train$Playoffs != predicted.glm0.insample, 1, 0))

#Out of Sample Confusion Matrix

Moneyball.teststatst <- glm(Playoffs~RS+RA+W+BA+OBP+SLG+Year+G+OOBP+OSLG+Team+League, family = binomial(link = "logit"), Moneyball.test)

summary(Moneyball.teststatst)

moneyball.glm.step <- step(Moneyball.teststatst, k = log(nrow(Moneyball.test)), direction = c("both"))

Moneyball.statstestbestmodel <- glm(Playoffs ~ W+Year+G, family = binomial(link = "logit"), Moneyball.test)

summary(Moneyball.statstestbestmodel)

prob.glm0.outsample <- predict(Moneyball.statstestbestmodel, Moneyball.test, type = "Playoffs")

predicted.glm0.outsample <- prob.glm0.outsample > 0.2

predicted.glm0.outsample <- as.numeric(predicted.glm0.outsample)

table(Moneyball.test$Playoffs, predicted.glm0.outsample, dnn = c("Truth", "Predicted"))

mean(ifelse(Moneyball.test$Playoffs != predicted.glm0.outsample, 1, 0))

#ROC curve

par = (mfrow = c(1,1))

library(verification)

roc.plot(Moneyball.train$Playoffs == "1", prob.glm0.insample)

roc.plot(Moneyball.train$Playoffs == "1", prob.glm0.insample)$roc.vol

roc.plot(Moneyball.test$Playoffs == "1", prob.glm0.outsample)

roc.plot(Moneyball.test$Playoffs == "1", prob.glm0.outsample)$roc.vol

#In and Out of Sample MSE for Wins Multiple Linear Regression

#In sample testing mean squared error on training data

Moneyball.winsfactorsbestmodel <- lm(W~RS+RA+Playoffs, data = Moneyball.train)

summary(Moneyball.winsfactorsbestmodel)

pred.winsbestmodeltrain <- predict(Moneyball.winsfactorsbestmodel,newdata = Moneyball.train)

mse.bestfit <- mean((Moneyball.train$W - pred.winsbestmodeltrain)^2)

mse.bestfit

#Comparison including OBP and SLG

#Out of sample testing mean squared error on testing data

Moneyball.winsfactorstest <- lm(W~RS+RA+BA+OBP+SLG+OOBP+OSLG+Year+G+Playoffs+Team+League

, data = Moneyball.test)

summary(Moneyball.winsfactorstest)

moneyball.lm.winstest <- step(Moneyball.winsfactorstest, k = log(nrow(Moneyball.test)), direction = c("both"))

Moneyball.winsbestmodeltest <- lm(W ~ RS + RA + G + Playoffs, data = Moneyball.test)

summary(Moneyball.winsbestmodeltest)

pred.model1 <- predict(Moneyball.winsbestmodeltest,newdata = Moneyball.test)

mpse.model1 <- mean((Moneyball.test$W - pred.model1)^2)

mpse.model1

#Regression Tree

set.seed(05542052)

par(mfrow = c(1,1))

library(rpart)

#Wins Linear Regression Tree

winsstats <- rpart(W~RS+RA+BA+OBP+SLG+OOBP+OSLG+Year+G+Team+League, data = Moneyball.train)

winsstats

summary(winsstats)

winsstatspredict <- predict(winsstats, Moneyball.test)

plot(winsstats)

text(winsstats)

plotcp(winsstats)

printcp(winsstats)

#Playoffs Classification Tree

playoffstats <- rpart(Playoffs~RS+RA+BA+OBP+SLG+OOBP+OSLG+Year+G+W+League+Team, data = Moneyball.train)

playoffstats

summary(playoffstats)

plot(playoffstats)

text(playoffstats)

plotcp(playoffstats)

printcp(playoffstats)

#Playoffs Logisitc AUC/ROC Curves and Confusion Matrices with MR

prob.glm0.insampleclass <- predict(playoffstats, Moneyball.train)

predicted.glm0.insample <- prob.glm0.insampleclass > 0.2

predicted.glm0.insample <- as.numeric(predicted.glm0.insample)

table(Moneyball.train$Playoffs, predicted.glm0.insample, dnn = c("Truth", "Predicted"))

mean(ifelse(Moneyball.train$Playoffs != predicted.glm0.insample, 1, 0))

playoffstatstest <- rpart(Playoffs~RS+RA+BA+OBP+SLG+OOBP+OSLG+Year+G+W+League+Team, data = Moneyball.test)

playoffstatstest

prob.glm0.outsample <- predict(playoffstatstest, Moneyball.test)

predicted.glm0.outsample <- prob.glm0.outsample > 0.2

predicted.glm0.outsample <- as.numeric(predicted.glm0.outsample)

table(Moneyball.test$Playoffs, predicted.glm0.outsample, dnn = c("Truth", "Predicted"))

mean(ifelse(Moneyball.test$Playoffs != predicted.glm0.outsample, 1, 0))

roc.plot(Moneyball.train$Playoffs == "1", prob.glm0.insampleclass)

roc.plot(Moneyball.train$Playoffs == "1", prob.glm0.insampleclass)$roc.vol

roc.plot(Moneyball.test$Playoffs == "1", prob.glm0.outsample)

roc.plot(Moneyball.test$Playoffs == "1", prob.glm0.outsample)$roc.vol

#Multiple Linear Regression In and Out of Sample MSE

#In Sample Error

Moneyball.train.pred.reg = predict(winsstats, Moneyball.train)

mse <- mean((Moneyball.train.pred.reg - Moneyball.train$W)^2)

mse

#Out of Sample Error

winsstatstest <- rpart(W~RS+RA+BA+OBP+SLG+OOBP+OSLG+Year+G+Team+League, data = Moneyball.test)

winsstatstest

Moneyball.test.pred.reg = predict(winsstatstest, Moneyball.test)

mset <- mean((Moneyball.test.pred.reg - Moneyball.test$W)^2)

mset

#Generalized Additive Models Linear

set.seed(05542052)

library(mgcv)

GAM.Wins <- gam(W~s(RS)+s(RA)+s(BA)+s(OBP)+

s(SLG)+s(OOBP)+s(OSLG)+Year+G+Team+League+Playoffs,

data = Moneyball.train)

summary(GAM.Wins)

GAM.WinsBestModel <- gam(W~(RS)+(RA)+(OBP)+(SLG)+Playoffs+G, data = Moneyball.train)

summary(GAM.WinsBestModel)

#Residual Plot for GAM Moneyball Data

plot(fitted(GAM.WinsBestModel), residuals(GAM.WinsBestModel), xlab = ' fitted',

ylab = 'residuals', main = ' Residuals by Fitted from GAM')

#Generalized Additive Models Logistic

set.seed(05542052)

GAM.Playoffs <- gam(Playoffs~s(RS)+s(RA)+s(BA)+s(OBP)+

s(SLG)+s(OOBP)+s(OSLG)+Year+G+Team+League+s(W),

family = "binomial", data = Moneyball.train)

summary(GAM.Playoffs)

GAM.PlayoffsBestModel <- gam(Playoffs~s(W)+Year, family = "binomial", data = Moneyball.train)

summary(GAM.PlayoffsBestModel)

#Residual Plot for GAM Moneyball Data

plot(fitted(GAM.Playoffs), residuals(GAM.Playoffs), xlab = ' fitted',

ylab = 'residuals', main = ' Residuals by Fitted from GAM')

#Multiple Lienar Regression In and Out of Sample MSE

#Model In-sample MSE and out of sample SSE

#EDF

GAM.Wins.MSE.train <- GAM.WinsBestModel$dev/GAM.WinsBestModel$df.res

GAM.Wins.MSE.train

#In sample error

GAM.Wins.predict.train <- predict(GAM.WinsBestModel, Moneyball.train)

GAM.Wins.predict.train

GAM.Wins.MSE.traine <- mean((GAM.Wins.predict.train - Moneyball.train[,"W"])^2)

GAM.Wins.MSE.traine

#Out of sample summary and Error

GAM.WinsModelt <- gam(W~s(RS)+s(RA)+s(BA)+s(OBP)+

s(SLG)+s(OOBP)+s(OSLG)+Year+G+Team+League+Playoffs,

data = Moneyball.test)

summary(GAM.WinsModelt)

GAM.BestModelt <- gam(W~s(RS)+s(RA)+G+Playoffs, data = Moneyball.test)

summary(GAM.BestModelt)

GAM.Wins.predict.test <- predict(GAM.BestModelt, Moneyball.test)

GAM.Wins.predict.test

GAM.Wins.MSE.test <- mean((GAM.Wins.predict.test - Moneyball.test[,"W"])^2)

GAM.Wins.MSE.test

#Logistic Regression

#Optimal cut off probability search

# define the searc grid from 0.01 to 0.20

searchgrid = seq(0.01, 0.2, 0.01)

# result.gam is a 99x2 matrix, the 1st col stores the cut-off p, the 2nd

# column stores the cost

result.gam = cbind(searchgrid, NA)

# in the cost function, both r and pi are vectors, r=truth, pi=predicted

# probability

cost1 <- function(r, pi) {

weight1 = 1

weight0 = 1

c1 = (r == 1) & (pi < pcut) #logical vector - true if actual 1 but predict 0

c0 = (r == 0) & (pi > pcut) #logical vecotr - true if actual 0 but predict 1

return(mean(weight1 \* c1 + weight0 \* c0))

}

for (i in 1:length(searchgrid)) {

pcut <- result.gam[i, 1]

# assign the cost to the 2nd col

result.gam[i, 2] <- cost1(Moneyball.train$Playoffs, predict(GAM.PlayoffsBestModel, type = "response"))

}

plot(result.gam, ylab = "Cost in Training Set")

#Minimum Cost in sample

index.min <- which.min(result.gam[, 2]) #find the index of minimum value

result.gam[index.min, 2]

#Optimal cut off prob

result.gam[index.min, 1]

#MR in sample

pcut <- result.gam[index.min, 1]

#MR in sample

pcut <- result.gam[index.min, 1]

prob.gam.in <- predict(GAM.PlayoffsBestModel, Moneyball.train, type = "response")

pred.gam.in <- (prob.gam.in >= pcut) \* 1

table(Moneyball.train$Playoffs, pred.gam.in, dnn = c("Observation", "Prediction"))

mean(ifelse(Moneyball.train$Playoffs != pred.gam.in, 1, 0))

#MR out of sample

GAM.Playoffstest <- gam(Playoffs~s(RS)+s(RA)+s(BA)+s(OBP)+

s(SLG)+s(OOBP)+s(OSLG)+Year+G+Team+League+s(W),

family = "binomial", data = Moneyball.test)

summary(GAM.Playoffstest)

GAM.PlayoffsBestModeltest <- gam(Playoffs~Team, family = "binomial", data = Moneyball.train)

summary(GAM.PlayoffsBestModeltest)

pcut <- result.gam[index.min, 1]

prob.gam.out <- predict(GAM.PlayoffsBestModeltest, Moneyball.test, type = "response")

pred.gam.out <- (prob.gam.out >= pcut) \* 1

table(Moneyball.test$Playoffs, pred.gam.out, dnn = c("Observation", "Prediction"))

mean(ifelse(Moneyball.test$Playoffs != pred.gam.out, 1, 0))

roc.plot(Moneyball.train$Playoffs == "1", prob.gam.in)

roc.plot(Moneyball.train$Playoffs == "1", prob.gam.in)$roc.vol

roc.plot(Moneyball.test$Playoffs == "1", prob.gam.out)

roc.plot(Moneyball.test$Playoffs == "1", prob.gam.out)$roc.vol

#Linear Discriminant Analysis

set.seed(05542052)

Moneyballtrain.lda <- lda(Playoffs ~ Team+League+RS+RA+BA+OBP+SLG+OOBP+OSLG+Year+G+W, data = Moneyball.train)

Moneyballtrain.lda

summary(Moneyballtrain.lda)

#In and Out of Sample Analysis

prob.lda.in <- predict(Moneyballtrain.lda, data = Moneyball.train)

pcut.lda <- 0.20

pred.lda.in <- (prob.lda.in$posterior[, 2] >= pcut.lda) \* 1

table(Moneyball.train$Playoffs, pred.lda.in, dnn = c("Obs", "Pred"))

mean(ifelse(Moneyball.train$Playoffs != pred.lda.in, 1, 0))

Moneyballtest.lda <- lda(Playoffs ~ Team+League+RS+RA+BA+OBP+SLG+OOBP+OSLG+Year+G+W, data = Moneyball.test)

Moneyballtest.lda

summary(Moneyballtest.lda)

lda.out <- predict(Moneyballtest.lda, newdata = Moneyball.test)

cut.lda <- 0.20

pred.lda.out <- as.numeric((lda.out$posterior[, 2] >= cut.lda))

table(Moneyball.test$Playoffs, pred.lda.out, dnn = c("Obs", "Pred"))

mean(ifelse(Moneyball.test$Playoffs != pred.lda.out, 1, 0))

library(ROCR)

#In sample

pred <- prediction(prob.lda.in$posterior[, 2], Moneyball.train$Playoffs)

perf <- performance(pred,"tpr","fpr")

plot(perf,colorize = TRUE)

performance(pred,"auc")@y.values

#Out of Sample

predtest <- prediction(lda.out$posterior[, 2], Moneyball.test$Playoffs)

perftest <- performance(predtest,"tpr","fpr")

plot(perftest,colorize = TRUE)

performance(predtest,"auc")@y.values

#Clustering

#Look at numerical variables

set.seed(05542052)

alldata = Moneyball.train[,1:18]

Moneyball.trainnumeric = Moneyball.train[,c(3:14,16:18)]

#Average silhoute coefficient

wss <- (nrow(Moneyball.trainnumeric) - 1)\*sum(apply(Moneyball.trainnumeric,2,var))

for (i in 2:12) wss[i] <- sum(kmeans(Moneyball.trainnumeric,

centers = i)$withinss)

plot(1:12, wss, type = "b", xlab = "Number of Clusters", ylab = "Within groups sum of squares")

fit <- kmeans(Moneyball.trainnumeric, 4) #4 cluster solution

#Display number of clusters in each cluster

table(fit$cluster)

fit$cluster

library(fpc)

plotcluster(Moneyball.trainnumeric, fit$cluster)

aggregate(Moneyball.trainnumeric,by = list(fit$cluster),FUN = mean)

table(fit$cluster, alldata$Playoffs)

d = dist(Moneyball.trainnumeric, method = "euclidean")

result = matrix(nrow = 14, ncol = 3)

for (i in 2:15) {

cluster\_result = kmeans(Moneyball.trainnumeric, i)

clusterstat = cluster.stats(d, cluster\_result$cluster)

result[i - 1,1] = i

result[i - 1,2] = clusterstat$avg.silwidth

result[i - 1,3] = clusterstat$dunn

}

plot(result[,c(1,2)], type = "l", ylab = 'silhouette width', xlab = 'number of clusters')