Code appendix

Julia Lee

```
library(readr)
library(float)
library(devtools)
source_url
('https://gist.githubusercontent.com/fawda123/7471137/raw/466c14
  74d0a505ff044412703516c34f1a4684a5/nnet_plot_update.r')

library(kableExtra)
library(clusterGeneration)
library(tictoc)
library(dplyr)
library(MASS)
library(dplyr)
library(DT)
library(readr)
library(nloptr)
library(e1071)
library(ISLR)
library(GGally)
library(caret)
library(nnet)
library(rpart)
library(MASS)
library(dplyr)
library(ISLR)
library(cluster)
library(flashClust)
library(factoextra)
library(ape)
library(ggdendro)
library(dendextend)
library(ggplot2)
#install.packages("klaR")
library(klaR)
library(gplots)
library(kohonen)
library(circlize)
library(rpart.plot)
library(rattle)
library(tree)
library(class)
library(randomForest)
library(readr)
library(readxl)
```

```
library(ggplot2)
library(tm)
library(e1071)
library(gridExtra)
library(class)
library(ISLR)
library(dplyr)
library(nnet)
```

```
train1 <- read_csv("train.csv")
test1 <- read_csv("test.csv")
```

Data:

```
head(test1)
```

```
## # A tibble: 6 x 19
##   Team  Rushes Passes DidWin FirstDown sacks Interception Fumbles incomplete
##   <chr>  <dbl>  <dbl>  <dbl>     <dbl> <dbl>        <dbl>   <dbl>      <dbl>
## 1 ATL       20     44      0        17     6            1       1         22
## 2 PHI       28     40      1        19     2            1       2         17
## 3 ARI       14     35      0        16     2            1       1         14
## 4 BAL       33     41      1        38     2            0       3         14
## 5 BUF       23     35      0        10     6            2       1         21
## 6 CAR       30     30      1        25     3            0       2         11
## # ... with 10 more variables: TDs <dbl>, twopoint <dbl>, yards <dbl>, N <dbl>,
## #   IsHome <fct>, score <dbl>, favorby <dbl>, roof <chr>, surface <chr>,
## #   RatioPasstoRush <dbl>
```

```
head(train1)
```

```
## # A tibble: 6 x 19
##   Team  Rushes Passes DidWin FirstDown sacks Interception Fumbles incomplete
##   <chr>  <dbl>  <dbl> <fct>      <dbl> <dbl>        <dbl>   <dbl>      <dbl>
## 1 CHI       16     49 0             15     5            1       1         20
## 2 GB        21     32 1             14     6            0       2         14
## 3 ARI       26     56 0             28     5            1       0         25
## 4 ATL       17     50 0             24     5            2       1         13
## 5 BAL       43     29 1             41     1            0       0          5
## 6 BUF       23     40 1             25     1            3       2         13
## # ... with 10 more variables: TDs <dbl>, twopoint <dbl>, yards <dbl>, N <dbl>,
## #   IsHome <fct>, score <dbl>, favorby <dbl>, roof <chr>, surface <chr>,
## #   RatioPasstoRush <dbl>
```

Decision Tree:

```
set.seed(10064)
mod1<-tree(DidWin~.,data=train1, method = "class")
cv_mod1 = cv.tree(mod1)
#plot(cv_mod1$size, cv_mod1$dev, type = 'b')
fittree2<-rpart(DidWin~.,train1,method="class")
plotcp(fittree2)
```

```
fittree2<-rpart(DidWin~.,train1,method="class")
rpart.plot(fittree2,extra=104)
```

```
tree_pred = predict(fittree2, test1, type="class")
table(tree_pred, test1$DidWin) %>%
  kbl(caption = "confusion Table of Classification Tree") %>%
  kable_styling()
```

Random Forest:

```
m1 <- randomForest(
  formula = DidWin~.,
  data    = train1,
   mtry = 20, importance = TRUE)

random_forest_estimate = predict(m1,
                                 newdata = test1)
```

```
plot(m1)
```

```
importance(m1)
```

```
varImpPlot(m1, main = "Random Forest")
```

```
table(random_forest_estimate, test1$DidWin)%>%
kbl(caption = "random_forest_estimate") %>%
  kable_styling()
```

Discriminant analysis

```
ggpairs(train1[,-1])
```

LDA:

```
model_LDA = lda(DidWin~., data = train1)
predictions_LDA = data.frame(predict(model_LDA, test1))
GMeans<-model_LDA$means
GMeans<-t(GMeans)
GMeans%>%
  kbl(caption = "Group Means of Training Data") %>%
  kable_styling()
```

```
model_LDA = lda(DidWin~., data = train1)
predictions_LDA = data.frame(predict(model_LDA, test1))
coeff<-model_LDA$scaling

coeff<-coeff[32:50,]

coeff%>%
  kbl(caption = "coefficients of linear discriminants output") %>%
  kable_styling()
```

```
predictions_LDA = cbind(test, predictions_LDA)

LDA<-predictions_LDA %>%
  count(class, DidWin) %>%
  kbl(caption = "LDA prediction Table") %>%
  kable_styling()
LDA
```

QDA:

```
model_QDA = qda(DidWin~., data = train1)
predictions_QDA = data.frame(predict(model_QDA, test1))

predictions_QDA = cbind(test, predictions_QDA)

Q<-predictions_QDA %>%
  count(class, DidWin) %>%
  kbl(caption = "QDA prediction Table") %>%
  kable_styling()
Q
```

```
set.seed(109)
minval<-1e10
lam<-0.2
for (k in 1:100) {
  init<-runif(12,-0.7,0.7)
fitnn<-nnet(class.ind(DidWin)~.,train1,size=2,decay=lam,entropy=TRUE,maxit=5000,wts=init)
 if (fitnn$value<minval) {
minval<-fitnn$value
fitnn.save<-fitnn
}
}
```

Neural Network

```
plot.nnet(fitnn.save, cex.val = 0.5 )
```

```
k<-test1[,-4]
class.predict<-predict(fitnn.save,k,type="class")
```

```
correct<-(test1$DidWin==class.predict)
n<-length(correct[correct== TRUE])
(1-(n/1530))*100
```

Clustering

```
clust <- read_csv("clust.csv")
C<-clust[,2:13]
c<-data.matrix(C)
rownames(c) <- clust$Team
distance <- get_dist(c)
fviz_dist(distance, gradient = list(low = "#FFFFFF", mid = "#00AFBB", high = "#000066"))
```

Hierarchical Clustering:

```
dist.map <- dist(clust)
hclustavg <- hclust(dist.map,method= "complete")
labels(hclustavg) <- clust$Team


x = c()
y = c(2:10)
N <- 10
for (k in 2:N) {
 p<-cutree(hclustavg,k=k)
 s<-silhouette(p,dist.map)
 S<-mean(s[,3])
 x<-append(x,S)
}
d<-cbind(y,x)
plot(d, type = "line")
```

```
dend<-as.dendrogram(hclustavg)%>%
  color_branches(k=7)
plot(dend)
```

```
set.seed(100)
mycol <- colorRampPalette(c("lightblue","blue","darkblue"))(12)
C<-clust[,2:13]
c<-data.matrix(C)
rownames(c) <- clust$Team

heatmap.2(c, col=mycol, trace="none",scale="column",cexCol =0.65 ,margins=c(7,5))
```

K-means:

```
fviz_nbclust(c, kmeans, method='silhouette')
```

```
set.seed(100)
k<-kmeans(c, centers = 6,nstart = 100)
fviz_cluster(k, data = c)
```

Self-organizing Maps:

```
SOM<-som(c,grid=somgrid(2,3,topo="hexagonal"),rlen=1000)
coords<-SOM$grid$pts
par(bg = rgb(0.9,0.94,0.95), font = 1, pch = 16, pch = 16, cex = 0.75)
plot(SOM,type="mapping",labels = rownames(c), cex.lab = 0.65,
     main = "NFL Team mapping", cex.main = 5.4)
text(coords,labels=seq(1,6), col = "blue")
```