Final Project Code- TED Dataset - December 17, 2018.

Nrithya Saravanabhavan, Liz Boccolini, Xiqiao Tong, Rohith Putcha, Akash Pandey

Question 1: Can we predict the number of views and comments for a new video?

```
ted = read.csv(file.choose(), header=T, sep=",")
attach(ted)
ted_model = lm(views~duration + languages, data=ted)
summary(ted_model)
comments_model = lm(comments~duration + views + languages)
summary(comments_model)
correlation = cor(ted[, c(3,6,53)])
corr2 = cor(ted[, c(1,3,6,53)])
library("corrplot")
corrplot(correlation, method="number")
corrplot(corr2, method="number")
library("scatterplot3d")
scatterplot3d(views, duration, languages, main="Relationship Between Views, Duration, and
Languages", pch=19, highlight.3d = TRUE)
library(data.table)
library(neuralnet)
nn1 = ted[, c(3,6,11,13:26)] ##please note that here the value “ted” was changed to include the
URL of each talk, as well as the number of times each TED rating was applied.  Thus, row
numbers are now different.  The change was done manually in Excel.
max = apply(nn1, 2, max)
min = apply(nn1, 2, min)
scaled = as.data.frame(scale(nn1, center=min, scale=max-min))
index = sample(1:nrow(nn1), round(0.75*nrow(nn1)))
training = scaled[index, ]
testing = scaled[-index, ]
names=names(scaled)
f=as.formula(paste("views ~ ", paste(names[!names %in% "views"], collapse="+")))
nn = neuralnet(f, data=training, hidden=c(3,2), linear.output=T)
plot(nn)
test_nn = compute(nn, testing[c("duration", "languages", "Beautiful", "Confusing", "Courageous",
"Fascinating", "Informative", "Ingenious", "Inspiring", "Jaw.dropping", "Longwinded",
"Obnoxious", "OK", "Persuasive", "Unconvincing")])
y = test_nn$net.result*(max(nn1$views)-min(nn1$views)) + min(nn1$views)
b = (testing$views) * (max(nn1$views)-min(nn1$views)) + min(nn1$views)
MSE = sum((b-y)^2)/nrow(testing)
print(MSE)
nn2 = ted[, c(1,3,6,11,13:26)]
max = apply(nn2, 2, max)
```

```
min = apply(nn2, 2, min)
scaled = as.data.frame(scale(nn2, center=min, scale=max-min))
index = sample(1:nrow(nn2), round(0.75*nrow(nn2)))
training = scaled[index, ]
testing = scaled[-index, ]
names=names(scaled)
f=as.formula(paste("comments ~ ", paste(names[!names %in% "comments"], collapse="+")))
nn = neuralnet(f, data=training, hidden=c(3,2), linear.output=T)
plot(nn)
test_nn = compute(nn, testing[c("duration", "languages", “views”, "Beautiful", "Confusing",
"Courageous", "Fascinating", "Informative", "Ingenious", "Inspiring", "Jaw.dropping",
"Longwinded", "Obnoxious", "OK", "Persuasive", "Unconvincing")])
y = test_nn$net.result*(max(nn2$comments)-min(nn2$comments)) + min(nn2$comments)
b = (testing$comments) * (max(nn2$comments)-min(nn1$comments)) + min(nn1$comments)
MSE = sum((b-y)^2)/nrow(testing)
print(MSE)
```

Question 2: Can we cluster videos based on popularity?

```
# Open the dataset
TEDdata = read.csv(file.choose(),header = T, sep = ",")
# Select the columns (comments, languages and views)
TEDdata2 <- TEDdata[,c(15,1,6,17)]
View (TEDdata2)
# Remove NAs
TEDdata3 <- na.omit(TEDdata2)
# Convert the title to numeric
TEDdata3$title <- as.numeric(as.factor(TEDdata3$title))
View (TEDdata3)
# Attach libraries
library(factoextra)
library(datasets)
library(cluster)
set.seed(123)
# K-mean model with 2 clusters
TED123 <- kmeans(TEDdata3[, 2:4], 2, nstart = 20)
# K-mean model with 3 clusters
TED123 <- kmeans(TEDdata3[, 2:4], 3, nstart = 20)
# K-mean model with 4 clusters
TED123 <- kmeans(TEDdata3[, 2:4], 4, nstart = 20)
# Summary
TED123
#Visualize the clusters
```

```
set.seed(123)
fviz_cluster(TED123, data = TEDdata3, ellipse.type = "convex", palette = "jco", ggtheme =
theme_minimal())
```

Question 3: Based on views comments and ratings, what kind of impact did a video have on viewers?

```
library(anytime)
library(randomForest)
library(gmodels)
library(class)

#import data
ted_main_raw=read.csv(file.choose(), header=TRUE, sep=",")
ratings=read.csv(file.choose(), header=TRUE, sep=",")

#remove na
ted_main=na.omit(ted_main_raw)

#create viewer sentiment metric
attach(ratings)
data=ratings[,2:15]
ratings$Sentiment=as.factor(colnames(data)[apply(data, 1, which.max)])
ratings$metric=ifelse(ratings$Sentiment == 'Beautiful' |
                      ratings$Sentiment == 'Fascinating' |
                      ratings$Sentiment == 'Jaw.dropping' |
                      ratings$Sentiment == 'Ingenious' |
                      ratings$Sentiment == 'Inspiring',
                     4, ifelse(ratings$Sentiment == 'Courageous' |
                               ratings$Sentiment == 'Funny' |
                               ratings$Sentiment == 'Informative' |
                               ratings$Sentiment == 'Persuasive',
                              3, ifelse(ratings$Sentiment == 'OK' |
                                        ratings$Sentiment == 'Longwinded',
                                       2, 1)))
#create new dataset for classification
view_sent=ratings[,-16]
view_sent$comments=ted_main$comments
view_sent$views=ted_main$views
view_sent$duration=ted_main$duration
view_sent$languages=ted_main$languages

#KNN Classification
```

```
ind=sample(2,nrow(view_sent),replace=TRUE, prob=c(0.75, 0.25))
training=view_sent[ind==1,c(2:15,17:20)]
testing=view_sent[ind==2,c(2:15,17:20)]
trainLabels=view_sent[ind==1,16]
testLabels=view_sent[ind==2,16]
model=knn(train=training, test=testing, cl=trainLabels,k=3)
CrossTable(x=model, y=testLabels, prop.chisq = FALSE, prop.r=FALSE,
prop.c=FALSE,prop.t=FALSE )

#Random Forest Classsification
set.seed(1234)
view_sent$metric=as.factor(view_sent$metric)
view_sent=view_sent[,-1]
data=sample(nrow(view_sent), 0.60*nrow(view_sent))
train=view_sent[data,]
test=view_sent[-data,]
model=randomForest(train$metric~., data=train)
model2 = predict(model, newdata=test)
CrossTable(x=model2, y=test$metric,prop.chisq = FALSE, prop.r=FALSE,
prop.c=FALSE,prop.t=FALSE)
plot(model2)
```

Question 4: Can we cluster videos based on tags for a video recommendation system?

```
#load library
library(ggplot2)
library(scales)
library(gmodels)
library(cluster)

options(scipen=5)

#import data
ted_main_raw=read.csv(file.choose(), header=TRUE, sep=",")
ratings=read.csv(file.choose(), header=TRUE, sep=",")

#remove na
ted_main=na.omit(ted_main_raw)

#extract required videos based on tag
tag_tech=subset(ted_main,grepl( "'technology'",ted_main$tags, fixed=TRUE))
tag_cul=subset(ted_main,grepl( "'culture'",ted_main$tags, fixed=TRUE))
```

```
#assign labels
tag_clust=rbind(tag_tech,tag_cul)
tag_clust$dup=duplicated(tag_clust)
repeated=subset(tag_clust,tag_clust$dup==TRUE)
final=tag_clust[!(tag_clust$url %in% repeated$url),]
final$label=ifelse(grepl( '"technology"',final$tags, fixed=TRUE),01, 10)
repeated$label=11
final=rbind(final,repeated)
final$label=as.factor(final$label)

#create curated data with ratings
final=final[,c(17,1,3,6,18,20)]
clust_data=merge(final,ratings, by="url")
clust_data$label=as.factor(clust_data$label)
clust_data_trim=subset(clust_data,views<30000000)
ggplot(clust_data_trim, aes(views,comments,color=label))+geom_point()+labs(title="Tag
Cluster")+xlab("Views")+ylab("Comments")

#Agglomerative
predictor=clust_data_trim[,2:19]
response=clust_data_trim[,20]
model4=agnes(x=predictor, diss=FALSE,stand=TRUE, method="complete")
DendClusters=as.dendrogram(model4)
groups=cutree(model4,k=3)
CrossTable(x=groups, y=clust_data_trim$label,prop.chisq = FALSE, prop.r=FALSE,
prop.c=FALSE,prop.t=FALSE)
Cluster=as.factor(groups)
ggplot(clust_data_trim,
aes(views,comments,color=Cluster))+geom_point()+labs(title="Agglomerative Cluster")
plot(DendClusters, main="Agglomerative Model")
rect.hclust(model4, k=3, border="red")

#Divisive
model5=diana(clust_data_trim[,2:19],metric="manhattan", stand=TRUE)
DendClusters=as.dendrogram(model5)
groups=cutree(model5,k=3)
Cluster=as.factor(groups)
ggplot(clust_data_trim, aes(views,comments,color=Cluster))+geom_point()+labs(title="Divisive
Cluster")
table(groups, clust_data_trim$label)
plot(DendClusters,main="Divisive Model")
rect.hclust(model5, k=3, border="red")
```

Reference

Kabacoff, Robert I. “Scatterplots.” *Quick-R*, 2017, https://www.statmethods.net/graphs/scatterplot.html.

“TED.” TED: Ideas Worth Spreading, https://www.ted.com/.

“TED Talks.” TED: Ideas Worth Spreading, https://www.ted.com/talks.

“Visualize Correlation Matrix Using Correlogram.” STHDA: Statistical Tools for High-Throughput Data Analysis, http://www.sthda.com/english/wiki/visualize-correlation-matrix-using-correlogram.

Cluster Analysis in R: Practical Guide. (2017). http://www.sthda.com/english/articles/25-cluster-analysis-in-r-practical-guide/111-types-of-clustering-methods-overview-and-quick-start-r-code/

ML for DS Fall 2017 Resources kmeans.r. https://content.sakai.rutgers.edu/access/content/group/2b7fe5da-092c-4f14-bc24-012422586a14/Code/kmeans.r Retrieved on Dec 2017.

What does total ss and between ss mean in k-means clustering? (2014). https://stats.stackexchange.com/questions/82776/what-does-total-ss-and-between-ss-mean-in-k-means-clustering/82779

Convert Unix epoch to date object. Retrieved from https://stackoverflow.com/questions/13456241/convert-unix-epoch-to-date-object

How to do a non greedy match in grep. Retrieved from https://stackoverflow.com/questions/3027518/how-to-do-a-non-greedy-match-in-grep

Escaping apostrophes in regex. Retrieved from

https://stackoverflow.com/questions/7400000/escaping-apostrophes-in-regex

Pattern matching and replacement. Retrieved from

https://stat.ethz.ch/R-manual/R-devel/library/base/html/grep.html