DSE6211 Module 07 Lab 07

Joseph Annand

2024-02-27

## Load Libraries

library(dplyr)  
library(caret)  
library(keras)  
library(reticulate)  
library(tensorflow)  
library(MESS)

## Data Pre-processing

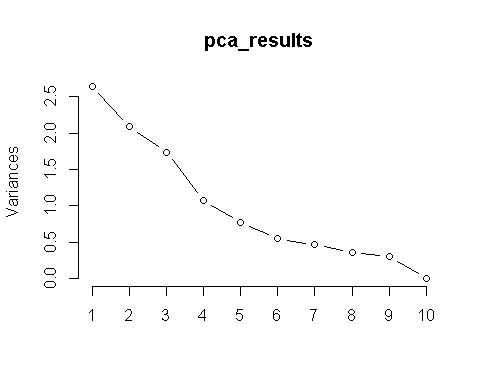
data <- read.csv("lab\_7\_data/lab\_7\_data.csv")  
training\_ind <- createDataPartition(data$lodgepole\_pine,  
 p = 0.75,  
 list = F,  
 times = 1)  
  
training\_set <- data[training\_ind, ]  
test\_set <- data[-training\_ind, ]  
  
top\_20\_soil\_types <- training\_set %>%  
 group\_by(soil\_type) %>%  
 summarise(count = n()) %>%  
 arrange(desc(count)) %>%  
 select(soil\_type) %>%  
 top\_n(20)  
  
training\_set$soil\_type <- ifelse(training\_set$soil\_type %in% top\_20\_soil\_types$soil\_type,  
 training\_set$soil\_type,  
 "other")  
  
training\_set$wilderness\_area <- factor(training\_set$wilderness\_area)  
training\_set$soil\_type <- factor(training\_set$soil\_type)  
  
  
onehot\_encoder <- dummyVars(~ wilderness\_area + soil\_type,  
 training\_set[, c("wilderness\_area", "soil\_type")],  
 levelsOnly = T,  
 fullRank = T)  
  
onehot\_enc\_training <- predict(onehot\_encoder,  
 training\_set[, c("wilderness\_area", "soil\_type")])  
  
training\_set <- cbind(training\_set, onehot\_enc\_training)  
  
  
test\_set$soil\_type <- ifelse(test\_set$soil\_type %in% top\_20\_soil\_types$soil\_type,  
 test\_set$soil\_type,  
 "other")  
  
test\_set$wilderness\_area <- factor(test\_set$wilderness\_area)  
test\_set$soil\_type <- factor(test\_set$soil\_type)  
  
onehot\_enc\_test <- predict(onehot\_encoder, test\_set[, c("wilderness\_area", "soil\_type")])  
test\_set <- cbind(test\_set, onehot\_enc\_test)  
  
  
test\_set[, -c(11:13)] <- scale(test\_set[, -c(11:13)],  
 center = apply(training\_set[, -c(11:13)], 2, mean),  
 scale = apply(training\_set[, -c(11:13)], 2, sd))  
training\_set[, -c(11:13)] <- scale(training\_set[, -c(11:13)])  
  
  
training\_features <- array(data = unlist(training\_set[, -c(11:13)]),  
 dim = c(nrow(training\_set), 33))  
training\_labels <- array(data = unlist(training\_set[, 13]),  
 dim = c(nrow(training\_set)))  
  
test\_features <- array(data = unlist(training\_set[, -c(11:13)]),  
 dim = c(nrow(test\_set), 33))  
test\_labels <- array(data = unlist(training\_set[, 13]),  
 dim = c(nrow(test\_set)))

## Principal Components Analysis

pca\_results <- prcomp(training\_features[, 1:10])  
summary(pca\_results)

## Importance of components:  
## PC1 PC2 PC3 PC4 PC5 PC6 PC7  
## Standard deviation 1.6265 1.4459 1.3199 1.0365 0.87777 0.73981 0.68234  
## Proportion of Variance 0.2646 0.2091 0.1742 0.1074 0.07705 0.05473 0.04656  
## Cumulative Proportion 0.2646 0.4736 0.6478 0.7553 0.83232 0.88705 0.93361  
## PC8 PC9 PC10  
## Standard deviation 0.59868 0.55050 0.04908  
## Proportion of Variance 0.03584 0.03031 0.00024  
## Cumulative Proportion 0.96945 0.99976 1.00000

screeplot(pca\_results, type = "line")



training\_rotated <- as.matrix(training\_features[, 1:10]) %\*% pca\_results$rotation  
training\_features <- cbind(training\_features, training\_rotated[,1:6])  
  
test\_rotated <- as.matrix(test\_features[,1:10]) %\*% pca\_results$rotation  
test\_features <- cbind(test\_features, test\_rotated[,1:6])

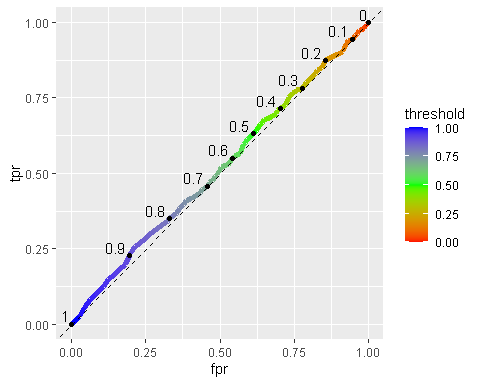
## Neural Network

use\_virtualenv("my\_tf\_workspace")  
  
model <- keras\_model\_sequential(list(  
 layer\_dense(units = 50, activation = "relu"),  
 layer\_dense(units = 25, activation = "relu"),  
 layer\_dense(units = 1, activation = "sigmoid")  
))  
  
compile(model,  
 optimizer = "rmsprop",  
 loss = "binary\_crossentropy",  
 metrics = "accuracy")  
  
history <- fit(model, training\_features[, -1\*c(1:13)], training\_labels,  
 epoch = 40, batch\_size = 512, validation\_split = 0.33)  
  
predictions <- predict(model, test\_features[, -1\*c(1:13)])  
test\_set$p\_prob <- predictions[, 1]

## ROC Curve

roc\_data <- data.frame(threshold=seq(1,0,-0.01), fpr=0, tpr=0)  
for (i in roc\_data$threshold) {  
   
 over\_threshold <- test\_set[test\_set$p\_prob >= i, ]  
   
 fpr <- sum(over\_threshold$lodgepole\_pine==0)/sum(test\_set$lodgepole\_pine==0)  
 roc\_data[roc\_data$threshold==i, "fpr"] <- fpr  
   
 tpr <- sum(over\_threshold$lodgepole\_pine==1)/sum(test\_set$lodgepole\_pine==1)  
 roc\_data[roc\_data$threshold==i, "tpr"] <- tpr  
   
}  
  
# ROC curve plot  
ggplot() +  
 geom\_line(data = roc\_data, aes(x=fpr, y=tpr, color = threshold), size = 2) +  
 scale\_color\_gradientn(colors = rainbow(3)) +  
 geom\_abline(intercept = 0, slope = 1, lty = 2) +  
 geom\_point(data = roc\_data[seq(1, 101, 10), ], aes(x = fpr, y =tpr)) +  
 geom\_text(data = roc\_data[seq(1, 101, 10), ],  
 aes(x = fpr, y = tpr, label = threshold, hjust = 1.2, vjust = -0.2))

## Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.  
## ℹ Please use `linewidth` instead.  
## This warning is displayed once every 8 hours.  
## Call `lifecycle::last\_lifecycle\_warnings()` to see where this warning was  
## generated.



# AUC - area under ROC curve   
  
auc <- auc(x = roc\_data$fpr, y = roc\_data$tpr, type = "spline")  
auc

## [1] 0.5135168

## Exercises

### Exercise 1

The main difference between unsupervised learning and supervised learning is that unsupervised learning uses data where no labels are associated with the observations. Supervised labels uses labeled data and attempts to infer a relationship between the data features and the labels. In unsupervised learning, since there are no labels, a relationship between features and labels is not a concern; instead, the relationship between observations or between features is the focus.

### Exercise 2

Centering is always required for PCA as it ensures the data is centered around the origin, the first PC is not dominated by the means of the variables and captures the true variance of the data, and the first PC passes through the center fo the data cloud. Scaling is almost always required to ensure that all numerical features have similar weighting in the analysis.

### Exercise 3

pca\_results$rotation

## ## PC1 PC2 PC3 PC4 PC5 ## [1,] 0.12436481 0.350838903 -0.35933970 0.042061845 -0.61181150 ## [2,] 0.48445350 -0.163101122 0.08020895 0.086998408 -0.01934947 ## [3,] -0.11242617 -0.526563938 -0.12376189 0.383177908 -0.15215551 ## [4,] 0.10421443 -0.007314845 -0.64999392 -0.222807804 0.16582886 ## [5,] 0.07667941 -0.238066557 -0.60690052 -0.126606077 0.20347536 ## [6,] 0.11848796 0.392522376 -0.15114526 0.521791913 -0.27369245 ## [7,] -0.44960967 0.351784771 -0.03592607 -0.291603920 0.02981930 ## [8,] 0.39507907 0.339247520 0.10892361 -0.359812263 0.14833773 ## [9,] 0.58911834 -0.026206218 0.10410294 -0.006908082 0.07580911 ## [10,] -0.03349689 0.350298914 -0.10708322 0.539770433 0.65552399 ## PC6 PC7 PC8 PC9 PC10 ## [1,] 0.5579826243 -0.06475247 0.18066279 0.089712563 0.0036124845 ## [2,] 0.0692806525 -0.65389834 -0.49679668 0.212444062 -0.0019908809 ## [3,] -0.0159103911 -0.36651761 0.39928983 -0.462369829 -0.1303481236 ## [4,] 0.0001661755 0.11360900 -0.42764412 -0.541907345 0.0010609210 ## [5,] -0.2340238706 -0.06514480 0.32894395 0.586915420 -0.0004049373 ## [6,] -0.6706393399 0.04836036 -0.09807086 -0.012188934 -0.0022861466 ## [7,] -0.1336822946 -0.46682274 -0.01611437 -0.005453788 -0.5922854557 ## [8,] -0.1834442904 -0.30718578 0.47110422 -0.301454761 0.3530740466 ## [9,] 0.0287539261 0.30428795 0.17578563 -0.061494465 -0.7124047672 ## [10,] 0.3559895588 -0.09757876 0.08556102 0.008073466 0.0013932667Exercises

### Exercise 1

The main difference between unsupervised learning and supervised learning is that unsupervised learning uses data where no labels are associated with the observations. Supervised labels uses labeled data and attempts to infer a relationship between the data features and the labels. In unsupervised learning, since there are no labels, a relationship between features and labels is not a concern; instead, the relationship between observations or between features is the focus.

### Exercise 2

Centering is always required for PCA as it ensures the data is centered around the origin, the first PC is not dominated by the means of the variables and captures the true variance of the data, and the first PC passes through the center fo the data cloud. Scaling is almost always required to ensure that all numerical features have similar weighting in the analysis.

### Exercise 3

The ninth feature, which is hillshade\_3pm, has the strongest influence on PC1. The tenth feature, which is horizontal\_distance\_to\_fire\_point, has the weakest influence on PC1.

pca\_results$rotation

## PC1 PC2 PC3 PC4 PC5  
## [1,] 0.12436481 0.350838903 -0.35933970 0.042061845 -0.61181150  
## [2,] 0.48445350 -0.163101122 0.08020895 0.086998408 -0.01934947  
## [3,] -0.11242617 -0.526563938 -0.12376189 0.383177908 -0.15215551  
## [4,] 0.10421443 -0.007314845 -0.64999392 -0.222807804 0.16582886  
## [5,] 0.07667941 -0.238066557 -0.60690052 -0.126606077 0.20347536  
## [6,] 0.11848796 0.392522376 -0.15114526 0.521791913 -0.27369245  
## [7,] -0.44960967 0.351784771 -0.03592607 -0.291603920 0.02981930  
## [8,] 0.39507907 0.339247520 0.10892361 -0.359812263 0.14833773  
## [9,] 0.58911834 -0.026206218 0.10410294 -0.006908082 0.07580911  
## [10,] -0.03349689 0.350298914 -0.10708322 0.539770433 0.65552399  
## PC6 PC7 PC8 PC9 PC10  
## [1,] 0.5579826243 -0.06475247 0.18066279 0.089712563 0.0036124845  
## [2,] 0.0692806525 -0.65389834 -0.49679668 0.212444062 -0.0019908809  
## [3,] -0.0159103911 -0.36651761 0.39928983 -0.462369829 -0.1303481236  
## [4,] 0.0001661755 0.11360900 -0.42764412 -0.541907345 0.0010609210  
## [5,] -0.2340238706 -0.06514480 0.32894395 0.586915420 -0.0004049373  
## [6,] -0.6706393399 0.04836036 -0.09807086 -0.012188934 -0.0022861466  
## [7,] -0.1336822946 -0.46682274 -0.01611437 -0.005453788 -0.5922854557  
## [8,] -0.1834442904 -0.30718578 0.47110422 -0.301454761 0.3530740466  
## [9,] 0.0287539261 0.30428795 0.17578563 -0.061494465 -0.7124047672  
## [10,] 0.3559895588 -0.09757876 0.08556102 0.008073466 0.0013932667