Lab 4

Adeline Casali

2024-02-09

Data Pre-Processing and Building the Model

# Load packages and data  
library(dplyr)

##   
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':  
##   
## filter, lag

## The following objects are masked from 'package:base':  
##   
## intersect, setdiff, setequal, union

library(caret)

## Loading required package: ggplot2

## Loading required package: lattice

data <- read.csv("lab\_4\_data.csv")  
  
# Create testing and training sets  
training\_ind <- createDataPartition(data$lodgepole\_pine,   
 p = 0.75,   
 list = FALSE,   
 times = 1)  
training\_set <- data[training\_ind, ]  
test\_set <- data[-training\_ind, ]  
  
# Assessing, grouping, and factoring categorical variables  
unique(training\_set$wilderness\_area)

## [1] "wilderness\_area\_1" "wilderness\_area\_3" "wilderness\_area\_4"  
## [4] "wilderness\_area\_2"

unique(training\_set$soil\_type)

## [1] "soil\_type\_30" "soil\_type\_12" "soil\_type\_29" "soil\_type\_20" "soil\_type\_23"  
## [6] "soil\_type\_24" "soil\_type\_22" "soil\_type\_32" "soil\_type\_11" "soil\_type\_33"  
## [11] "soil\_type\_13" "soil\_type\_31" "soil\_type\_2" "soil\_type\_5" "soil\_type\_1"   
## [16] "soil\_type\_10" "soil\_type\_3" "soil\_type\_6" "soil\_type\_14" "soil\_type\_17"  
## [21] "soil\_type\_40" "soil\_type\_4" "soil\_type\_38" "soil\_type\_39" "soil\_type\_35"  
## [26] "soil\_type\_25" "soil\_type\_19" "soil\_type\_16" "soil\_type\_18" "soil\_type\_8"   
## [31] "soil\_type\_9" "soil\_type\_28" "soil\_type\_34" "soil\_type\_37" "soil\_type\_21"  
## [36] "soil\_type\_36" "soil\_type\_26" "soil\_type\_27"

top\_20\_soil\_types <- training\_set %>%   
 group\_by(soil\_type) %>%   
 summarize(count = n()) %>%   
 arrange(desc(count)) %>%  
 select(soil\_type) %>%   
 top\_n(20)

## Selecting by soil\_type

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)  
  
class(training\_set$wilderness\_area)

## [1] "factor"

class(training\_set$soil\_type)

## [1] "factor"

levels(training\_set$wilderness\_area)

## [1] "wilderness\_area\_1" "wilderness\_area\_2" "wilderness\_area\_3"  
## [4] "wilderness\_area\_4"

levels(training\_set$soil\_type)

## [1] "other" "soil\_type\_27" "soil\_type\_28" "soil\_type\_29" "soil\_type\_3"   
## [6] "soil\_type\_30" "soil\_type\_31" "soil\_type\_32" "soil\_type\_33" "soil\_type\_34"  
## [11] "soil\_type\_35" "soil\_type\_36" "soil\_type\_37" "soil\_type\_38" "soil\_type\_39"  
## [16] "soil\_type\_4" "soil\_type\_40" "soil\_type\_5" "soil\_type\_6" "soil\_type\_8"   
## [21] "soil\_type\_9"

# One-hot encoding the training set  
onehot\_encoder <- dummyVars(~ wilderness\_area + soil\_type,   
 training\_set[, c("wilderness\_area", "soil\_type")],   
 levelsOnly = TRUE,   
 fullRank = TRUE)  
  
onehot\_enc\_training <- predict(onehot\_encoder,   
 training\_set[, c("wilderness\_area", "soil\_type")])  
training\_set <- cbind(training\_set, onehot\_enc\_training)  
  
# One-hot encoding the test set  
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)  
  
# Scaling test and training sets  
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)])  
  
# Convert data sets to tensors  
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(test\_set[, -c(11:13)]),   
 dim = c(nrow(test\_set), 33))  
test\_labels <- array(data = unlist(test\_set[, 13]),   
 dim = c(nrow(test\_set)))  
  
# Loading keras and tensorflow libraries  
library(reticulate)  
library(tensorflow)

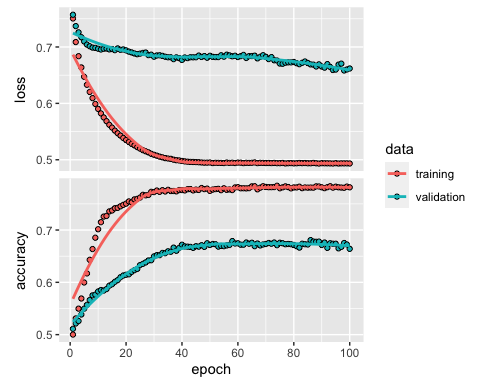
##   
## Attaching package: 'tensorflow'

## The following object is masked from 'package:caret':  
##   
## train

library(keras)  
  
# Building the model  
model <- keras\_model\_sequential(list(  
 layer\_dense(units = 20, activation = "relu"),   
 layer\_dense(units = 10, activation = "relu"),   
 layer\_dense(units = 1, activation = "sigmoid")  
))  
compile(model,   
 optimizer = "rmsprop",   
 loss = "binary\_crossentropy",   
 metrics = "accuracy")  
  
# Training the model  
history <- fit(model, training\_features, training\_labels,   
 epochs = 100, batch\_size = 512, validation\_split = 0.33)

## Epoch 1/100  
## 9/9 - 3s - loss: 0.7506 - accuracy: 0.5001 - val\_loss: 0.7572 - val\_accuracy: 0.5111 - 3s/epoch - 337ms/step  
## Epoch 2/100  
## 9/9 - 0s - loss: 0.7087 - accuracy: 0.5309 - val\_loss: 0.7371 - val\_accuracy: 0.5213 - 176ms/epoch - 20ms/step  
## Epoch 3/100  
## 9/9 - 0s - loss: 0.6837 - accuracy: 0.5497 - val\_loss: 0.7260 - val\_accuracy: 0.5259 - 123ms/epoch - 14ms/step  
## Epoch 4/100  
## 9/9 - 0s - loss: 0.6637 - accuracy: 0.5691 - val\_loss: 0.7173 - val\_accuracy: 0.5385 - 115ms/epoch - 13ms/step  
## Epoch 5/100  
## 9/9 - 0s - loss: 0.6472 - accuracy: 0.5997 - val\_loss: 0.7098 - val\_accuracy: 0.5496 - 114ms/epoch - 13ms/step  
## Epoch 6/100  
## 9/9 - 0s - loss: 0.6330 - accuracy: 0.6170 - val\_loss: 0.7042 - val\_accuracy: 0.5570 - 113ms/epoch - 13ms/step  
## Epoch 7/100  
## 9/9 - 0s - loss: 0.6203 - accuracy: 0.6431 - val\_loss: 0.7013 - val\_accuracy: 0.5663 - 113ms/epoch - 13ms/step  
## Epoch 8/100  
## 9/9 - 0s - loss: 0.6093 - accuracy: 0.6636 - val\_loss: 0.6989 - val\_accuracy: 0.5751 - 114ms/epoch - 13ms/step  
## Epoch 9/100  
## 9/9 - 0s - loss: 0.5990 - accuracy: 0.6853 - val\_loss: 0.6984 - val\_accuracy: 0.5765 - 116ms/epoch - 13ms/step  
## Epoch 10/100  
## 9/9 - 0s - loss: 0.5902 - accuracy: 0.7015 - val\_loss: 0.6969 - val\_accuracy: 0.5825 - 116ms/epoch - 13ms/step  
## Epoch 11/100  
## 9/9 - 0s - loss: 0.5823 - accuracy: 0.7150 - val\_loss: 0.6955 - val\_accuracy: 0.5853 - 114ms/epoch - 13ms/step  
## Epoch 12/100  
## 9/9 - 0s - loss: 0.5751 - accuracy: 0.7257 - val\_loss: 0.6969 - val\_accuracy: 0.5843 - 114ms/epoch - 13ms/step  
## Epoch 13/100  
## 9/9 - 0s - loss: 0.5684 - accuracy: 0.7269 - val\_loss: 0.6972 - val\_accuracy: 0.5871 - 115ms/epoch - 13ms/step  
## Epoch 14/100  
## 9/9 - 0s - loss: 0.5620 - accuracy: 0.7356 - val\_loss: 0.6957 - val\_accuracy: 0.5936 - 114ms/epoch - 13ms/step  
## Epoch 15/100  
## 9/9 - 0s - loss: 0.5570 - accuracy: 0.7374 - val\_loss: 0.6963 - val\_accuracy: 0.5964 - 115ms/epoch - 13ms/step  
## Epoch 16/100  
## 9/9 - 0s - loss: 0.5516 - accuracy: 0.7417 - val\_loss: 0.6944 - val\_accuracy: 0.6010 - 114ms/epoch - 13ms/step  
## Epoch 17/100  
## 9/9 - 0s - loss: 0.5470 - accuracy: 0.7424 - val\_loss: 0.6975 - val\_accuracy: 0.6043 - 114ms/epoch - 13ms/step  
## Epoch 18/100  
## 9/9 - 0s - loss: 0.5429 - accuracy: 0.7454 - val\_loss: 0.6950 - val\_accuracy: 0.6098 - 117ms/epoch - 13ms/step  
## Epoch 19/100  
## 9/9 - 0s - loss: 0.5387 - accuracy: 0.7481 - val\_loss: 0.6956 - val\_accuracy: 0.6131 - 118ms/epoch - 13ms/step  
## Epoch 20/100  
## 9/9 - 0s - loss: 0.5350 - accuracy: 0.7504 - val\_loss: 0.6941 - val\_accuracy: 0.6145 - 114ms/epoch - 13ms/step  
## Epoch 21/100  
## 9/9 - 0s - loss: 0.5316 - accuracy: 0.7547 - val\_loss: 0.6918 - val\_accuracy: 0.6154 - 115ms/epoch - 13ms/step  
## Epoch 22/100  
## 9/9 - 0s - loss: 0.5282 - accuracy: 0.7529 - val\_loss: 0.6903 - val\_accuracy: 0.6200 - 115ms/epoch - 13ms/step  
## Epoch 23/100  
## 9/9 - 0s - loss: 0.5255 - accuracy: 0.7586 - val\_loss: 0.6885 - val\_accuracy: 0.6233 - 117ms/epoch - 13ms/step  
## Epoch 24/100  
## 9/9 - 0s - loss: 0.5225 - accuracy: 0.7595 - val\_loss: 0.6872 - val\_accuracy: 0.6256 - 114ms/epoch - 13ms/step  
## Epoch 25/100  
## 9/9 - 0s - loss: 0.5197 - accuracy: 0.7620 - val\_loss: 0.6882 - val\_accuracy: 0.6330 - 116ms/epoch - 13ms/step  
## Epoch 26/100  
## 9/9 - 0s - loss: 0.5171 - accuracy: 0.7673 - val\_loss: 0.6894 - val\_accuracy: 0.6353 - 115ms/epoch - 13ms/step  
## Epoch 27/100  
## 9/9 - 0s - loss: 0.5151 - accuracy: 0.7664 - val\_loss: 0.6901 - val\_accuracy: 0.6381 - 114ms/epoch - 13ms/step  
## Epoch 28/100  
## 9/9 - 0s - loss: 0.5131 - accuracy: 0.7671 - val\_loss: 0.6856 - val\_accuracy: 0.6418 - 115ms/epoch - 13ms/step  
## Epoch 29/100  
## 9/9 - 0s - loss: 0.5105 - accuracy: 0.7730 - val\_loss: 0.6863 - val\_accuracy: 0.6423 - 120ms/epoch - 13ms/step  
## Epoch 30/100  
## 9/9 - 0s - loss: 0.5088 - accuracy: 0.7726 - val\_loss: 0.6827 - val\_accuracy: 0.6497 - 115ms/epoch - 13ms/step  
## Epoch 31/100  
## 9/9 - 0s - loss: 0.5072 - accuracy: 0.7755 - val\_loss: 0.6850 - val\_accuracy: 0.6520 - 119ms/epoch - 13ms/step  
## Epoch 32/100  
## 9/9 - 0s - loss: 0.5055 - accuracy: 0.7730 - val\_loss: 0.6853 - val\_accuracy: 0.6520 - 118ms/epoch - 13ms/step  
## Epoch 33/100  
## 9/9 - 0s - loss: 0.5043 - accuracy: 0.7762 - val\_loss: 0.6843 - val\_accuracy: 0.6562 - 120ms/epoch - 13ms/step  
## Epoch 34/100  
## 9/9 - 0s - loss: 0.5029 - accuracy: 0.7737 - val\_loss: 0.6827 - val\_accuracy: 0.6557 - 114ms/epoch - 13ms/step  
## Epoch 35/100  
## 9/9 - 0s - loss: 0.5022 - accuracy: 0.7769 - val\_loss: 0.6817 - val\_accuracy: 0.6594 - 116ms/epoch - 13ms/step  
## Epoch 36/100  
## 9/9 - 0s - loss: 0.5016 - accuracy: 0.7746 - val\_loss: 0.6818 - val\_accuracy: 0.6585 - 114ms/epoch - 13ms/step  
## Epoch 37/100  
## 9/9 - 0s - loss: 0.5001 - accuracy: 0.7746 - val\_loss: 0.6785 - val\_accuracy: 0.6664 - 114ms/epoch - 13ms/step  
## Epoch 38/100  
## 9/9 - 0s - loss: 0.4992 - accuracy: 0.7739 - val\_loss: 0.6816 - val\_accuracy: 0.6603 - 117ms/epoch - 13ms/step  
## Epoch 39/100  
## 9/9 - 0s - loss: 0.4988 - accuracy: 0.7764 - val\_loss: 0.6803 - val\_accuracy: 0.6659 - 118ms/epoch - 13ms/step  
## Epoch 40/100  
## 9/9 - 0s - loss: 0.4979 - accuracy: 0.7764 - val\_loss: 0.6770 - val\_accuracy: 0.6719 - 114ms/epoch - 13ms/step  
## Epoch 41/100  
## 9/9 - 0s - loss: 0.4971 - accuracy: 0.7757 - val\_loss: 0.6814 - val\_accuracy: 0.6687 - 119ms/epoch - 13ms/step  
## Epoch 42/100  
## 9/9 - 0s - loss: 0.4966 - accuracy: 0.7780 - val\_loss: 0.6803 - val\_accuracy: 0.6682 - 116ms/epoch - 13ms/step  
## Epoch 43/100  
## 9/9 - 0s - loss: 0.4962 - accuracy: 0.7783 - val\_loss: 0.6818 - val\_accuracy: 0.6677 - 119ms/epoch - 13ms/step  
## Epoch 44/100  
## 9/9 - 0s - loss: 0.4961 - accuracy: 0.7757 - val\_loss: 0.6817 - val\_accuracy: 0.6659 - 118ms/epoch - 13ms/step  
## Epoch 45/100  
## 9/9 - 0s - loss: 0.4959 - accuracy: 0.7773 - val\_loss: 0.6818 - val\_accuracy: 0.6691 - 115ms/epoch - 13ms/step  
## Epoch 46/100  
## 9/9 - 0s - loss: 0.4956 - accuracy: 0.7767 - val\_loss: 0.6848 - val\_accuracy: 0.6682 - 113ms/epoch - 13ms/step  
## Epoch 47/100  
## 9/9 - 0s - loss: 0.4952 - accuracy: 0.7803 - val\_loss: 0.6824 - val\_accuracy: 0.6710 - 115ms/epoch - 13ms/step  
## Epoch 48/100  
## 9/9 - 0s - loss: 0.4948 - accuracy: 0.7783 - val\_loss: 0.6839 - val\_accuracy: 0.6677 - 114ms/epoch - 13ms/step  
## Epoch 49/100  
## 9/9 - 0s - loss: 0.4949 - accuracy: 0.7792 - val\_loss: 0.6809 - val\_accuracy: 0.6747 - 115ms/epoch - 13ms/step  
## Epoch 50/100  
## 9/9 - 0s - loss: 0.4952 - accuracy: 0.7748 - val\_loss: 0.6844 - val\_accuracy: 0.6715 - 116ms/epoch - 13ms/step  
## Epoch 51/100  
## 9/9 - 0s - loss: 0.4942 - accuracy: 0.7773 - val\_loss: 0.6825 - val\_accuracy: 0.6733 - 115ms/epoch - 13ms/step  
## Epoch 52/100  
## 9/9 - 0s - loss: 0.4947 - accuracy: 0.7783 - val\_loss: 0.6821 - val\_accuracy: 0.6728 - 113ms/epoch - 13ms/step  
## Epoch 53/100  
## 9/9 - 0s - loss: 0.4945 - accuracy: 0.7787 - val\_loss: 0.6821 - val\_accuracy: 0.6682 - 118ms/epoch - 13ms/step  
## Epoch 54/100  
## 9/9 - 0s - loss: 0.4944 - accuracy: 0.7762 - val\_loss: 0.6853 - val\_accuracy: 0.6696 - 117ms/epoch - 13ms/step  
## Epoch 55/100  
## 9/9 - 0s - loss: 0.4942 - accuracy: 0.7799 - val\_loss: 0.6828 - val\_accuracy: 0.6691 - 114ms/epoch - 13ms/step  
## Epoch 56/100  
## 9/9 - 0s - loss: 0.4951 - accuracy: 0.7769 - val\_loss: 0.6863 - val\_accuracy: 0.6705 - 117ms/epoch - 13ms/step  
## Epoch 57/100  
## 9/9 - 0s - loss: 0.4945 - accuracy: 0.7776 - val\_loss: 0.6851 - val\_accuracy: 0.6715 - 114ms/epoch - 13ms/step  
## Epoch 58/100  
## 9/9 - 0s - loss: 0.4948 - accuracy: 0.7764 - val\_loss: 0.6835 - val\_accuracy: 0.6789 - 114ms/epoch - 13ms/step  
## Epoch 59/100  
## 9/9 - 0s - loss: 0.4942 - accuracy: 0.7815 - val\_loss: 0.6844 - val\_accuracy: 0.6742 - 114ms/epoch - 13ms/step  
## Epoch 60/100  
## 9/9 - 0s - loss: 0.4950 - accuracy: 0.7819 - val\_loss: 0.6822 - val\_accuracy: 0.6747 - 114ms/epoch - 13ms/step  
## Epoch 61/100  
## 9/9 - 0s - loss: 0.4948 - accuracy: 0.7799 - val\_loss: 0.6838 - val\_accuracy: 0.6719 - 161ms/epoch - 18ms/step  
## Epoch 62/100  
## 9/9 - 0s - loss: 0.4941 - accuracy: 0.7799 - val\_loss: 0.6859 - val\_accuracy: 0.6719 - 121ms/epoch - 13ms/step  
## Epoch 63/100  
## 9/9 - 0s - loss: 0.4946 - accuracy: 0.7796 - val\_loss: 0.6805 - val\_accuracy: 0.6761 - 116ms/epoch - 13ms/step  
## Epoch 64/100  
## 9/9 - 0s - loss: 0.4944 - accuracy: 0.7801 - val\_loss: 0.6838 - val\_accuracy: 0.6738 - 115ms/epoch - 13ms/step  
## Epoch 65/100  
## 9/9 - 0s - loss: 0.4947 - accuracy: 0.7835 - val\_loss: 0.6839 - val\_accuracy: 0.6756 - 138ms/epoch - 15ms/step  
## Epoch 66/100  
## 9/9 - 0s - loss: 0.4953 - accuracy: 0.7837 - val\_loss: 0.6846 - val\_accuracy: 0.6691 - 113ms/epoch - 13ms/step  
## Epoch 67/100  
## 9/9 - 0s - loss: 0.4941 - accuracy: 0.7778 - val\_loss: 0.6798 - val\_accuracy: 0.6742 - 114ms/epoch - 13ms/step  
## Epoch 68/100  
## 9/9 - 0s - loss: 0.4946 - accuracy: 0.7810 - val\_loss: 0.6826 - val\_accuracy: 0.6705 - 113ms/epoch - 13ms/step  
## Epoch 69/100  
## 9/9 - 0s - loss: 0.4944 - accuracy: 0.7837 - val\_loss: 0.6854 - val\_accuracy: 0.6724 - 114ms/epoch - 13ms/step  
## Epoch 70/100  
## 9/9 - 0s - loss: 0.4942 - accuracy: 0.7824 - val\_loss: 0.6825 - val\_accuracy: 0.6733 - 114ms/epoch - 13ms/step  
## Epoch 71/100  
## 9/9 - 0s - loss: 0.4943 - accuracy: 0.7819 - val\_loss: 0.6786 - val\_accuracy: 0.6752 - 113ms/epoch - 13ms/step  
## Epoch 72/100  
## 9/9 - 0s - loss: 0.4941 - accuracy: 0.7819 - val\_loss: 0.6820 - val\_accuracy: 0.6738 - 113ms/epoch - 13ms/step  
## Epoch 73/100  
## 9/9 - 0s - loss: 0.4944 - accuracy: 0.7821 - val\_loss: 0.6764 - val\_accuracy: 0.6761 - 114ms/epoch - 13ms/step  
## Epoch 74/100  
## 9/9 - 0s - loss: 0.4939 - accuracy: 0.7847 - val\_loss: 0.6760 - val\_accuracy: 0.6710 - 112ms/epoch - 12ms/step  
## Epoch 75/100  
## 9/9 - 0s - loss: 0.4940 - accuracy: 0.7799 - val\_loss: 0.6706 - val\_accuracy: 0.6756 - 116ms/epoch - 13ms/step  
## Epoch 76/100  
## 9/9 - 0s - loss: 0.4943 - accuracy: 0.7817 - val\_loss: 0.6702 - val\_accuracy: 0.6747 - 112ms/epoch - 12ms/step  
## Epoch 77/100  
## 9/9 - 0s - loss: 0.4935 - accuracy: 0.7835 - val\_loss: 0.6721 - val\_accuracy: 0.6728 - 113ms/epoch - 13ms/step  
## Epoch 78/100  
## 9/9 - 0s - loss: 0.4939 - accuracy: 0.7821 - val\_loss: 0.6728 - val\_accuracy: 0.6733 - 114ms/epoch - 13ms/step  
## Epoch 79/100  
## 9/9 - 0s - loss: 0.4941 - accuracy: 0.7796 - val\_loss: 0.6727 - val\_accuracy: 0.6728 - 114ms/epoch - 13ms/step  
## Epoch 80/100  
## 9/9 - 0s - loss: 0.4936 - accuracy: 0.7815 - val\_loss: 0.6734 - val\_accuracy: 0.6715 - 113ms/epoch - 13ms/step  
## Epoch 81/100  
## 9/9 - 0s - loss: 0.4939 - accuracy: 0.7815 - val\_loss: 0.6713 - val\_accuracy: 0.6691 - 113ms/epoch - 13ms/step  
## Epoch 82/100  
## 9/9 - 0s - loss: 0.4942 - accuracy: 0.7828 - val\_loss: 0.6695 - val\_accuracy: 0.6664 - 114ms/epoch - 13ms/step  
## Epoch 83/100  
## 9/9 - 0s - loss: 0.4940 - accuracy: 0.7801 - val\_loss: 0.6719 - val\_accuracy: 0.6738 - 113ms/epoch - 13ms/step  
## Epoch 84/100  
## 9/9 - 0s - loss: 0.4942 - accuracy: 0.7815 - val\_loss: 0.6743 - val\_accuracy: 0.6728 - 114ms/epoch - 13ms/step  
## Epoch 85/100  
## 9/9 - 0s - loss: 0.4938 - accuracy: 0.7810 - val\_loss: 0.6716 - val\_accuracy: 0.6724 - 113ms/epoch - 13ms/step  
## Epoch 86/100  
## 9/9 - 0s - loss: 0.4938 - accuracy: 0.7812 - val\_loss: 0.6691 - val\_accuracy: 0.6807 - 114ms/epoch - 13ms/step  
## Epoch 87/100  
## 9/9 - 0s - loss: 0.4943 - accuracy: 0.7810 - val\_loss: 0.6687 - val\_accuracy: 0.6779 - 112ms/epoch - 12ms/step  
## Epoch 88/100  
## 9/9 - 0s - loss: 0.4938 - accuracy: 0.7835 - val\_loss: 0.6654 - val\_accuracy: 0.6775 - 114ms/epoch - 13ms/step  
## Epoch 89/100  
## 9/9 - 0s - loss: 0.4933 - accuracy: 0.7833 - val\_loss: 0.6693 - val\_accuracy: 0.6779 - 113ms/epoch - 13ms/step  
## Epoch 90/100  
## 9/9 - 0s - loss: 0.4939 - accuracy: 0.7817 - val\_loss: 0.6653 - val\_accuracy: 0.6715 - 113ms/epoch - 13ms/step  
## Epoch 91/100  
## 9/9 - 0s - loss: 0.4938 - accuracy: 0.7801 - val\_loss: 0.6633 - val\_accuracy: 0.6761 - 113ms/epoch - 13ms/step  
## Epoch 92/100  
## 9/9 - 0s - loss: 0.4937 - accuracy: 0.7819 - val\_loss: 0.6696 - val\_accuracy: 0.6659 - 113ms/epoch - 13ms/step  
## Epoch 93/100  
## 9/9 - 0s - loss: 0.4936 - accuracy: 0.7837 - val\_loss: 0.6646 - val\_accuracy: 0.6747 - 113ms/epoch - 13ms/step  
## Epoch 94/100  
## 9/9 - 0s - loss: 0.4933 - accuracy: 0.7810 - val\_loss: 0.6592 - val\_accuracy: 0.6738 - 114ms/epoch - 13ms/step  
## Epoch 95/100  
## 9/9 - 0s - loss: 0.4939 - accuracy: 0.7828 - val\_loss: 0.6599 - val\_accuracy: 0.6738 - 114ms/epoch - 13ms/step  
## Epoch 96/100  
## 9/9 - 0s - loss: 0.4933 - accuracy: 0.7817 - val\_loss: 0.6687 - val\_accuracy: 0.6650 - 114ms/epoch - 13ms/step  
## Epoch 97/100  
## 9/9 - 0s - loss: 0.4936 - accuracy: 0.7812 - val\_loss: 0.6705 - val\_accuracy: 0.6677 - 114ms/epoch - 13ms/step  
## Epoch 98/100  
## 9/9 - 0s - loss: 0.4936 - accuracy: 0.7808 - val\_loss: 0.6583 - val\_accuracy: 0.6752 - 113ms/epoch - 13ms/step  
## Epoch 99/100  
## 9/9 - 0s - loss: 0.4936 - accuracy: 0.7831 - val\_loss: 0.6601 - val\_accuracy: 0.6724 - 112ms/epoch - 12ms/step  
## Epoch 100/100  
## 9/9 - 0s - loss: 0.4934 - accuracy: 0.7817 - val\_loss: 0.6619 - val\_accuracy: 0.6640 - 112ms/epoch - 12ms/step

plot(history)



# Using the model to make predictions  
predictions <- predict(model, test\_features)

## 69/69 - 1s - 505ms/epoch - 7ms/step

head(predictions, 10)

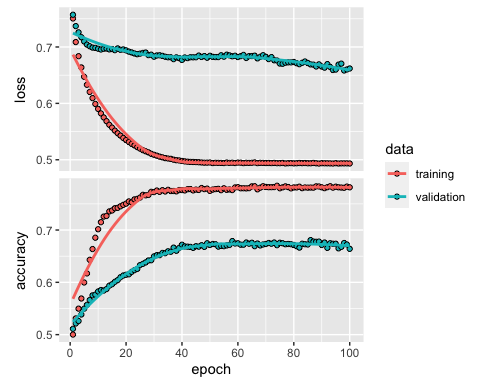
## [,1]  
## [1,] 0.9340833  
## [2,] 0.7535587  
## [3,] 0.8045436  
## [4,] 0.8171836  
## [5,] 0.6078069  
## [6,] 0.7348902  
## [7,] 0.5266478  
## [8,] 0.7145182  
## [9,] 0.6735506  
## [10,] 0.8488410

predicted\_class <- (predictions[, 1] >= 0.5) \* 1  
head(predicted\_class, 10)

## [1] 1 1 1 1 1 1 1 1 1 1

1. Below are the loss and accuracy curves obtained from running the code above.

plot(history)

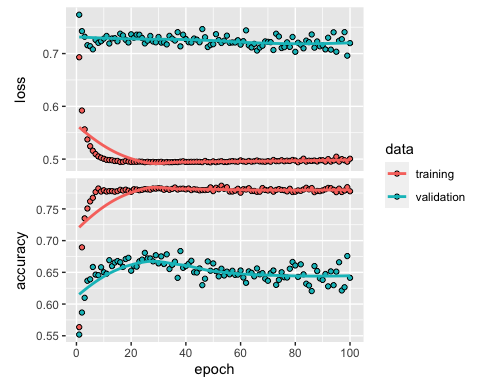


1. Below are the loss and accuracy curves obtained from running the code with 50 units for the first hidden layer and 25 units for the second hidden layer.

model1 <- keras\_model\_sequential(list(  
 layer\_dense(units = 50, activation = "relu"),   
 layer\_dense(units = 25, activation = "relu"),   
 layer\_dense(units = 1, activation = "sigmoid")  
))  
compile(model1,   
 optimizer = "rmsprop",   
 loss = "binary\_crossentropy",   
 metrics = "accuracy")  
history1 <- fit(model1, training\_features, training\_labels,   
 epochs = 100, batch\_size = 512, validation\_split = 0.33)

## Epoch 1/100  
## 9/9 - 1s - loss: 0.6930 - accuracy: 0.5636 - val\_loss: 0.7736 - val\_accuracy: 0.5519 - 1s/epoch - 117ms/step  
## Epoch 2/100  
## 9/9 - 0s - loss: 0.5921 - accuracy: 0.6894 - val\_loss: 0.7425 - val\_accuracy: 0.5867 - 132ms/epoch - 15ms/step  
## Epoch 3/100  
## 9/9 - 0s - loss: 0.5564 - accuracy: 0.7351 - val\_loss: 0.7319 - val\_accuracy: 0.6098 - 120ms/epoch - 13ms/step  
## Epoch 4/100  
## 9/9 - 0s - loss: 0.5373 - accuracy: 0.7506 - val\_loss: 0.7159 - val\_accuracy: 0.6367 - 115ms/epoch - 13ms/step  
## Epoch 5/100  
## 9/9 - 0s - loss: 0.5244 - accuracy: 0.7618 - val\_loss: 0.7139 - val\_accuracy: 0.6390 - 117ms/epoch - 13ms/step  
## Epoch 6/100  
## 9/9 - 0s - loss: 0.5158 - accuracy: 0.7675 - val\_loss: 0.7080 - val\_accuracy: 0.6585 - 112ms/epoch - 12ms/step  
## Epoch 7/100  
## 9/9 - 0s - loss: 0.5095 - accuracy: 0.7767 - val\_loss: 0.7260 - val\_accuracy: 0.6464 - 118ms/epoch - 13ms/step  
## Epoch 8/100  
## 9/9 - 0s - loss: 0.5051 - accuracy: 0.7821 - val\_loss: 0.7203 - val\_accuracy: 0.6455 - 111ms/epoch - 12ms/step  
## Epoch 9/100  
## 9/9 - 0s - loss: 0.5026 - accuracy: 0.7773 - val\_loss: 0.7235 - val\_accuracy: 0.6580 - 113ms/epoch - 13ms/step  
## Epoch 10/100  
## 9/9 - 0s - loss: 0.5006 - accuracy: 0.7794 - val\_loss: 0.7283 - val\_accuracy: 0.6501 - 114ms/epoch - 13ms/step  
## Epoch 11/100  
## 9/9 - 0s - loss: 0.4986 - accuracy: 0.7773 - val\_loss: 0.7334 - val\_accuracy: 0.6478 - 113ms/epoch - 13ms/step  
## Epoch 12/100  
## 9/9 - 0s - loss: 0.4982 - accuracy: 0.7780 - val\_loss: 0.7190 - val\_accuracy: 0.6691 - 115ms/epoch - 13ms/step  
## Epoch 13/100  
## 9/9 - 0s - loss: 0.4978 - accuracy: 0.7780 - val\_loss: 0.7287 - val\_accuracy: 0.6599 - 113ms/epoch - 13ms/step  
## Epoch 14/100  
## 9/9 - 0s - loss: 0.4965 - accuracy: 0.7771 - val\_loss: 0.7292 - val\_accuracy: 0.6654 - 110ms/epoch - 12ms/step  
## Epoch 15/100  
## 9/9 - 0s - loss: 0.4966 - accuracy: 0.7780 - val\_loss: 0.7232 - val\_accuracy: 0.6677 - 112ms/epoch - 12ms/step  
## Epoch 16/100  
## 9/9 - 0s - loss: 0.4946 - accuracy: 0.7801 - val\_loss: 0.7377 - val\_accuracy: 0.6659 - 113ms/epoch - 13ms/step  
## Epoch 17/100  
## 9/9 - 0s - loss: 0.4949 - accuracy: 0.7789 - val\_loss: 0.7344 - val\_accuracy: 0.6585 - 112ms/epoch - 12ms/step  
## Epoch 18/100  
## 9/9 - 0s - loss: 0.4965 - accuracy: 0.7803 - val\_loss: 0.7261 - val\_accuracy: 0.6710 - 111ms/epoch - 12ms/step  
## Epoch 19/100  
## 9/9 - 0s - loss: 0.4954 - accuracy: 0.7787 - val\_loss: 0.7210 - val\_accuracy: 0.6728 - 117ms/epoch - 13ms/step  
## Epoch 20/100  
## 9/9 - 0s - loss: 0.4946 - accuracy: 0.7769 - val\_loss: 0.7365 - val\_accuracy: 0.6525 - 113ms/epoch - 13ms/step  
## Epoch 21/100  
## 9/9 - 0s - loss: 0.4949 - accuracy: 0.7810 - val\_loss: 0.7300 - val\_accuracy: 0.6599 - 113ms/epoch - 13ms/step  
## Epoch 22/100  
## 9/9 - 0s - loss: 0.4949 - accuracy: 0.7805 - val\_loss: 0.7357 - val\_accuracy: 0.6589 - 112ms/epoch - 12ms/step  
## Epoch 23/100  
## 9/9 - 0s - loss: 0.4953 - accuracy: 0.7794 - val\_loss: 0.7357 - val\_accuracy: 0.6673 - 112ms/epoch - 12ms/step  
## Epoch 24/100  
## 9/9 - 0s - loss: 0.4951 - accuracy: 0.7801 - val\_loss: 0.7295 - val\_accuracy: 0.6701 - 111ms/epoch - 12ms/step  
## Epoch 25/100  
## 9/9 - 0s - loss: 0.4942 - accuracy: 0.7812 - val\_loss: 0.7193 - val\_accuracy: 0.6807 - 112ms/epoch - 12ms/step  
## Epoch 26/100  
## 9/9 - 0s - loss: 0.4949 - accuracy: 0.7808 - val\_loss: 0.7243 - val\_accuracy: 0.6719 - 111ms/epoch - 12ms/step  
## Epoch 27/100  
## 9/9 - 0s - loss: 0.4944 - accuracy: 0.7803 - val\_loss: 0.7334 - val\_accuracy: 0.6719 - 112ms/epoch - 12ms/step  
## Epoch 28/100  
## 9/9 - 0s - loss: 0.4948 - accuracy: 0.7821 - val\_loss: 0.7229 - val\_accuracy: 0.6673 - 112ms/epoch - 12ms/step  
## Epoch 29/100  
## 9/9 - 0s - loss: 0.4942 - accuracy: 0.7819 - val\_loss: 0.7213 - val\_accuracy: 0.6770 - 112ms/epoch - 12ms/step  
## Epoch 30/100  
## 9/9 - 0s - loss: 0.4946 - accuracy: 0.7826 - val\_loss: 0.7315 - val\_accuracy: 0.6640 - 113ms/epoch - 13ms/step  
## Epoch 31/100  
## 9/9 - 0s - loss: 0.4949 - accuracy: 0.7821 - val\_loss: 0.7207 - val\_accuracy: 0.6752 - 130ms/epoch - 14ms/step  
## Epoch 32/100  
## 9/9 - 0s - loss: 0.4939 - accuracy: 0.7801 - val\_loss: 0.7211 - val\_accuracy: 0.6631 - 121ms/epoch - 13ms/step  
## Epoch 33/100  
## 9/9 - 0s - loss: 0.4945 - accuracy: 0.7815 - val\_loss: 0.7174 - val\_accuracy: 0.6784 - 115ms/epoch - 13ms/step  
## Epoch 34/100  
## 9/9 - 0s - loss: 0.4941 - accuracy: 0.7801 - val\_loss: 0.7336 - val\_accuracy: 0.6552 - 113ms/epoch - 13ms/step  
## Epoch 35/100  
## 9/9 - 0s - loss: 0.4942 - accuracy: 0.7815 - val\_loss: 0.7284 - val\_accuracy: 0.6617 - 112ms/epoch - 12ms/step  
## Epoch 36/100  
## 9/9 - 0s - loss: 0.4946 - accuracy: 0.7837 - val\_loss: 0.7246 - val\_accuracy: 0.6668 - 120ms/epoch - 13ms/step  
## Epoch 37/100  
## 9/9 - 0s - loss: 0.4944 - accuracy: 0.7805 - val\_loss: 0.7406 - val\_accuracy: 0.6413 - 113ms/epoch - 13ms/step  
## Epoch 38/100  
## 9/9 - 0s - loss: 0.4952 - accuracy: 0.7817 - val\_loss: 0.7137 - val\_accuracy: 0.6835 - 111ms/epoch - 12ms/step  
## Epoch 39/100  
## 9/9 - 0s - loss: 0.4958 - accuracy: 0.7819 - val\_loss: 0.7357 - val\_accuracy: 0.6571 - 111ms/epoch - 12ms/step  
## Epoch 40/100  
## 9/9 - 0s - loss: 0.4946 - accuracy: 0.7808 - val\_loss: 0.7276 - val\_accuracy: 0.6599 - 112ms/epoch - 12ms/step  
## Epoch 41/100  
## 9/9 - 0s - loss: 0.4947 - accuracy: 0.7821 - val\_loss: 0.7216 - val\_accuracy: 0.6636 - 112ms/epoch - 12ms/step  
## Epoch 42/100  
## 9/9 - 0s - loss: 0.4949 - accuracy: 0.7796 - val\_loss: 0.7208 - val\_accuracy: 0.6682 - 112ms/epoch - 12ms/step  
## Epoch 43/100  
## 9/9 - 0s - loss: 0.4948 - accuracy: 0.7810 - val\_loss: 0.7285 - val\_accuracy: 0.6501 - 114ms/epoch - 13ms/step  
## Epoch 44/100  
## 9/9 - 0s - loss: 0.4951 - accuracy: 0.7803 - val\_loss: 0.7263 - val\_accuracy: 0.6497 - 111ms/epoch - 12ms/step  
## Epoch 45/100  
## 9/9 - 0s - loss: 0.4948 - accuracy: 0.7792 - val\_loss: 0.7214 - val\_accuracy: 0.6562 - 110ms/epoch - 12ms/step  
## Epoch 46/100  
## 9/9 - 0s - loss: 0.4943 - accuracy: 0.7796 - val\_loss: 0.7464 - val\_accuracy: 0.6297 - 113ms/epoch - 13ms/step  
## Epoch 47/100  
## 9/9 - 0s - loss: 0.4964 - accuracy: 0.7796 - val\_loss: 0.7310 - val\_accuracy: 0.6399 - 114ms/epoch - 13ms/step  
## Epoch 48/100  
## 9/9 - 0s - loss: 0.4945 - accuracy: 0.7801 - val\_loss: 0.7124 - val\_accuracy: 0.6673 - 112ms/epoch - 12ms/step  
## Epoch 49/100  
## 9/9 - 0s - loss: 0.4938 - accuracy: 0.7840 - val\_loss: 0.7164 - val\_accuracy: 0.6525 - 112ms/epoch - 12ms/step  
## Epoch 50/100  
## 9/9 - 0s - loss: 0.4957 - accuracy: 0.7780 - val\_loss: 0.7177 - val\_accuracy: 0.6640 - 111ms/epoch - 12ms/step  
## Epoch 51/100  
## 9/9 - 0s - loss: 0.4942 - accuracy: 0.7833 - val\_loss: 0.7368 - val\_accuracy: 0.6520 - 111ms/epoch - 12ms/step  
## Epoch 52/100  
## 9/9 - 0s - loss: 0.4952 - accuracy: 0.7810 - val\_loss: 0.7258 - val\_accuracy: 0.6437 - 111ms/epoch - 12ms/step  
## Epoch 53/100  
## 9/9 - 0s - loss: 0.4960 - accuracy: 0.7865 - val\_loss: 0.7201 - val\_accuracy: 0.6557 - 112ms/epoch - 12ms/step  
## Epoch 54/100  
## 9/9 - 0s - loss: 0.4946 - accuracy: 0.7817 - val\_loss: 0.7243 - val\_accuracy: 0.6450 - 111ms/epoch - 12ms/step  
## Epoch 55/100  
## 9/9 - 0s - loss: 0.4945 - accuracy: 0.7842 - val\_loss: 0.7339 - val\_accuracy: 0.6511 - 111ms/epoch - 12ms/step  
## Epoch 56/100  
## 9/9 - 0s - loss: 0.4955 - accuracy: 0.7773 - val\_loss: 0.7204 - val\_accuracy: 0.6469 - 112ms/epoch - 12ms/step  
## Epoch 57/100  
## 9/9 - 0s - loss: 0.4961 - accuracy: 0.7773 - val\_loss: 0.7210 - val\_accuracy: 0.6525 - 111ms/epoch - 12ms/step  
## Epoch 58/100  
## 9/9 - 0s - loss: 0.4955 - accuracy: 0.7787 - val\_loss: 0.7210 - val\_accuracy: 0.6460 - 111ms/epoch - 12ms/step  
## Epoch 59/100  
## 9/9 - 0s - loss: 0.4951 - accuracy: 0.7817 - val\_loss: 0.7253 - val\_accuracy: 0.6464 - 110ms/epoch - 12ms/step  
## Epoch 60/100  
## 9/9 - 0s - loss: 0.4976 - accuracy: 0.7755 - val\_loss: 0.7169 - val\_accuracy: 0.6529 - 112ms/epoch - 12ms/step  
## Epoch 61/100  
## 9/9 - 0s - loss: 0.4957 - accuracy: 0.7810 - val\_loss: 0.7235 - val\_accuracy: 0.6418 - 111ms/epoch - 12ms/step  
## Epoch 62/100  
## 9/9 - 0s - loss: 0.4949 - accuracy: 0.7805 - val\_loss: 0.7440 - val\_accuracy: 0.6335 - 111ms/epoch - 12ms/step  
## Epoch 63/100  
## 9/9 - 0s - loss: 0.4964 - accuracy: 0.7826 - val\_loss: 0.7155 - val\_accuracy: 0.6515 - 112ms/epoch - 12ms/step  
## Epoch 64/100  
## 9/9 - 0s - loss: 0.4951 - accuracy: 0.7783 - val\_loss: 0.7116 - val\_accuracy: 0.6492 - 111ms/epoch - 12ms/step  
## Epoch 65/100  
## 9/9 - 0s - loss: 0.4959 - accuracy: 0.7783 - val\_loss: 0.7184 - val\_accuracy: 0.6437 - 111ms/epoch - 12ms/step  
## Epoch 66/100  
## 9/9 - 0s - loss: 0.4974 - accuracy: 0.7769 - val\_loss: 0.7057 - val\_accuracy: 0.6562 - 113ms/epoch - 13ms/step  
## Epoch 67/100  
## 9/9 - 0s - loss: 0.4957 - accuracy: 0.7771 - val\_loss: 0.7114 - val\_accuracy: 0.6469 - 112ms/epoch - 12ms/step  
## Epoch 68/100  
## 9/9 - 0s - loss: 0.4953 - accuracy: 0.7828 - val\_loss: 0.7148 - val\_accuracy: 0.6487 - 111ms/epoch - 12ms/step  
## Epoch 69/100  
## 9/9 - 0s - loss: 0.4950 - accuracy: 0.7799 - val\_loss: 0.7230 - val\_accuracy: 0.6399 - 113ms/epoch - 13ms/step  
## Epoch 70/100  
## 9/9 - 0s - loss: 0.4962 - accuracy: 0.7789 - val\_loss: 0.7222 - val\_accuracy: 0.6367 - 112ms/epoch - 12ms/step  
## Epoch 71/100  
## 9/9 - 0s - loss: 0.4951 - accuracy: 0.7810 - val\_loss: 0.7216 - val\_accuracy: 0.6487 - 112ms/epoch - 12ms/step  
## Epoch 72/100  
## 9/9 - 0s - loss: 0.4983 - accuracy: 0.7771 - val\_loss: 0.7067 - val\_accuracy: 0.6529 - 112ms/epoch - 12ms/step  
## Epoch 73/100  
## 9/9 - 0s - loss: 0.4961 - accuracy: 0.7799 - val\_loss: 0.7376 - val\_accuracy: 0.6302 - 112ms/epoch - 12ms/step  
## Epoch 74/100  
## 9/9 - 0s - loss: 0.4961 - accuracy: 0.7812 - val\_loss: 0.7324 - val\_accuracy: 0.6381 - 113ms/epoch - 13ms/step  
## Epoch 75/100  
## 9/9 - 0s - loss: 0.4979 - accuracy: 0.7799 - val\_loss: 0.7105 - val\_accuracy: 0.6501 - 115ms/epoch - 13ms/step  
## Epoch 76/100  
## 9/9 - 0s - loss: 0.4960 - accuracy: 0.7794 - val\_loss: 0.7122 - val\_accuracy: 0.6423 - 113ms/epoch - 13ms/step  
## Epoch 77/100  
## 9/9 - 0s - loss: 0.4970 - accuracy: 0.7776 - val\_loss: 0.7213 - val\_accuracy: 0.6418 - 113ms/epoch - 13ms/step  
## Epoch 78/100  
## 9/9 - 0s - loss: 0.4964 - accuracy: 0.7817 - val\_loss: 0.7126 - val\_accuracy: 0.6557 - 111ms/epoch - 12ms/step  
## Epoch 79/100  
## 9/9 - 0s - loss: 0.4983 - accuracy: 0.7767 - val\_loss: 0.7315 - val\_accuracy: 0.6664 - 113ms/epoch - 13ms/step  
## Epoch 80/100  
## 9/9 - 0s - loss: 0.4973 - accuracy: 0.7771 - val\_loss: 0.7033 - val\_accuracy: 0.6608 - 112ms/epoch - 12ms/step  
## Epoch 81/100  
## 9/9 - 0s - loss: 0.4941 - accuracy: 0.7792 - val\_loss: 0.7205 - val\_accuracy: 0.6427 - 111ms/epoch - 12ms/step  
## Epoch 82/100  
## 9/9 - 0s - loss: 0.4967 - accuracy: 0.7785 - val\_loss: 0.7168 - val\_accuracy: 0.6627 - 112ms/epoch - 12ms/step  
## Epoch 83/100  
## 9/9 - 0s - loss: 0.5008 - accuracy: 0.7805 - val\_loss: 0.7099 - val\_accuracy: 0.6469 - 112ms/epoch - 12ms/step  
## Epoch 84/100  
## 9/9 - 0s - loss: 0.4953 - accuracy: 0.7833 - val\_loss: 0.7245 - val\_accuracy: 0.6321 - 112ms/epoch - 12ms/step  
## Epoch 85/100  
## 9/9 - 0s - loss: 0.4970 - accuracy: 0.7808 - val\_loss: 0.7217 - val\_accuracy: 0.6293 - 113ms/epoch - 13ms/step  
## Epoch 86/100  
## 9/9 - 0s - loss: 0.4965 - accuracy: 0.7828 - val\_loss: 0.7362 - val\_accuracy: 0.6205 - 113ms/epoch - 13ms/step  
## Epoch 87/100  
## 9/9 - 0s - loss: 0.4973 - accuracy: 0.7764 - val\_loss: 0.7112 - val\_accuracy: 0.6599 - 112ms/epoch - 12ms/step  
## Epoch 88/100  
## 9/9 - 0s - loss: 0.4955 - accuracy: 0.7783 - val\_loss: 0.7230 - val\_accuracy: 0.6520 - 114ms/epoch - 13ms/step  
## Epoch 89/100  
## 9/9 - 0s - loss: 0.4979 - accuracy: 0.7780 - val\_loss: 0.7138 - val\_accuracy: 0.6404 - 113ms/epoch - 13ms/step  
## Epoch 90/100  
## 9/9 - 0s - loss: 0.4978 - accuracy: 0.7792 - val\_loss: 0.7155 - val\_accuracy: 0.6381 - 112ms/epoch - 12ms/step  
## Epoch 91/100  
## 9/9 - 0s - loss: 0.4952 - accuracy: 0.7810 - val\_loss: 0.7079 - val\_accuracy: 0.6474 - 112ms/epoch - 12ms/step  
## Epoch 92/100  
## 9/9 - 0s - loss: 0.5000 - accuracy: 0.7831 - val\_loss: 0.7300 - val\_accuracy: 0.6279 - 113ms/epoch - 13ms/step  
## Epoch 93/100  
## 9/9 - 0s - loss: 0.4953 - accuracy: 0.7826 - val\_loss: 0.7074 - val\_accuracy: 0.6515 - 112ms/epoch - 12ms/step  
## Epoch 94/100  
## 9/9 - 0s - loss: 0.4974 - accuracy: 0.7801 - val\_loss: 0.7403 - val\_accuracy: 0.6297 - 113ms/epoch - 13ms/step  
## Epoch 95/100  
## 9/9 - 0s - loss: 0.5045 - accuracy: 0.7764 - val\_loss: 0.7034 - val\_accuracy: 0.6511 - 113ms/epoch - 13ms/step  
## Epoch 96/100  
## 9/9 - 0s - loss: 0.4952 - accuracy: 0.7819 - val\_loss: 0.7241 - val\_accuracy: 0.6659 - 120ms/epoch - 13ms/step  
## Epoch 97/100  
## 9/9 - 0s - loss: 0.5000 - accuracy: 0.7767 - val\_loss: 0.7272 - val\_accuracy: 0.6214 - 111ms/epoch - 12ms/step  
## Epoch 98/100  
## 9/9 - 0s - loss: 0.4978 - accuracy: 0.7789 - val\_loss: 0.7406 - val\_accuracy: 0.6265 - 112ms/epoch - 12ms/step  
## Epoch 99/100  
## 9/9 - 0s - loss: 0.4951 - accuracy: 0.7842 - val\_loss: 0.6960 - val\_accuracy: 0.6756 - 112ms/epoch - 12ms/step  
## Epoch 100/100  
## 9/9 - 0s - loss: 0.5007 - accuracy: 0.7778 - val\_loss: 0.7198 - val\_accuracy: 0.6413 - 114ms/epoch - 13ms/step

plot(history1)



1. Based on the curves produced above, the original model with fewer nodes seems to have better performance. Although they both have extremely similar values for both loss and accuracy, the second model displays a high level of variability, particularly regarding the validation loss and accuracy curves. This could be a sign of overfitting or instability. The first model, on the other hand, is very consistent across all curves while maintaining similar overall values.
2. Although they are very similar, the accuracy for the first model (with fewer nodes) is slightly better than that of the second model, at 73.38% for the first model and 72.51% for the second model.

results <- model %>% evaluate(test\_features, test\_labels)

## 69/69 - 0s - loss: 0.5813 - accuracy: 0.7338 - 427ms/epoch - 6ms/step

results

## loss accuracy   
## 0.5813453 0.7338229

results1 <- model1 %>% evaluate(test\_features, test\_labels)

## 69/69 - 0s - loss: 0.6346 - accuracy: 0.7251 - 324ms/epoch - 5ms/step

results1

## loss accuracy   
## 0.6346151 0.7251033