DSE6211 Final Project

Joseph Annand

2024-03-09

## Libraries

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(lubridate)

##   
## Attaching package: 'lubridate'

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

library(caret)

## Warning: package 'caret' was built under R version 4.3.2

## Loading required package: ggplot2

## Loading required package: lattice

library(MESS)

## Warning: package 'MESS' was built under R version 4.3.2

## Data Processing

project\_data <- read.csv("project\_data/project\_data.csv")  
  
rownames(project\_data) <- project\_data$Booking\_ID  
project\_data <- project\_data[, -1]  
  
training\_ind <- createDataPartition(project\_data$booking\_status,  
 p = 0.75,  
 list = F,  
 times = 1)  
  
training\_set <- project\_data[training\_ind, ]  
test\_set <- project\_data[-training\_ind, ]  
  
  
training\_set$booking\_status <- ifelse(training\_set$booking\_status=="canceled",1,0)  
test\_set$booking\_status <- ifelse(test\_set$booking\_status=="canceled",1,0)  
  
  
###### Training Set Categorical Features  
  
top\_8\_previous\_not\_cancelled <- training\_set %>%  
 group\_by(no\_of\_previous\_bookings\_not\_canceled) %>%  
 summarise(count = n()) %>%  
 arrange(desc(count)) %>%  
 select(no\_of\_previous\_bookings\_not\_canceled) %>%  
 slice(1:8)  
  
top\_2\_number\_of\_children <- training\_set %>%  
 group\_by(no\_of\_children) %>%  
 summarise(count = n()) %>%  
 arrange(desc(count)) %>%  
 slice(1:2)  
  
top\_3\_previous\_cancellations <- training\_set %>%  
 group\_by(no\_of\_previous\_cancellations) %>%  
 summarise(count = n()) %>%  
 arrange(desc(count)) %>%  
 slice(1:3)  
  
top\_8\_week\_nights <- training\_set %>%  
 group\_by(no\_of\_week\_nights) %>%  
 summarise(count = n()) %>%  
 arrange(desc(count)) %>%  
 slice(1:8)  
  
top\_6\_weekend\_nights <- training\_set %>%  
 group\_by(no\_of\_weekend\_nights) %>%  
 summarise(count = n()) %>%  
 arrange(desc(count)) %>%  
 slice(1:6)  
  
top\_4\_special\_requests <- training\_set %>%  
 group\_by(no\_of\_special\_requests) %>%  
 summarise(count = n()) %>%  
 arrange(desc(count)) %>%  
 slice(1:4)  
  
  
training\_set$no\_of\_previous\_bookings\_not\_canceled <- ifelse(  
 training\_set$no\_of\_previous\_bookings\_not\_canceled %in% top\_8\_previous\_not\_cancelled$no\_of\_previous\_bookings\_not\_canceled,  
 training\_set$no\_of\_previous\_bookings\_not\_canceled, "8+"  
)  
  
training\_set$no\_of\_children <- ifelse(  
 training\_set$no\_of\_children %in% top\_2\_number\_of\_children$no\_of\_children,  
 training\_set$no\_of\_children, "3+"  
)  
  
training\_set$no\_of\_previous\_cancellations <- ifelse(  
 training\_set$no\_of\_previous\_cancellations %in% top\_3\_previous\_cancellations$no\_of\_previous\_cancellations,  
 training\_set$no\_of\_previous\_cancellations, "3+"  
)  
  
training\_set$no\_of\_week\_nights <- ifelse(  
 training\_set$no\_of\_week\_nights %in% top\_8\_week\_nights$no\_of\_week\_nights,  
 training\_set$no\_of\_week\_nights, "8+"  
)  
  
training\_set$no\_of\_weekend\_nights <- ifelse(  
 training\_set$no\_of\_weekend\_nights %in% top\_6\_weekend\_nights$no\_of\_weekend\_nights,  
 training\_set$no\_of\_weekend\_nights, "6+"  
)  
  
training\_set$type\_of\_meal\_plan <- ifelse(training\_set$type\_of\_meal\_plan %in% c("meal\_plan\_1", "meal\_plan\_2"),  
 training\_set$type\_of\_meal\_plan,  
 "other")  
  
training\_set$no\_of\_special\_requests <- ifelse(training\_set$no\_of\_special\_requests %in% top\_4\_special\_requests$no\_of\_special\_requests,  
 training\_set$no\_of\_special\_requests,  
 "4+")  
  
training\_set$arrival\_date <- parse\_date\_time(training\_set$arrival\_date, "ymd")  
training\_set$booking\_date <- int\_start(interval(training\_set$arrival\_date - ddays(training\_set$lead\_time),   
 training\_set$arrival\_date))  
  
season\_months <- data.frame(winter = c(12,1,2), spring = c(3,4,5),  
 summer = c(6,7,8), fall = c(9,10,11))  
  
  
get\_season <- function(x) {  
 y <- month(x)  
 # print(y)  
 for (j in 1:length(colnames(season\_months))) {  
 # print(j)  
 if (y %in% season\_months[[j]]) {  
 # print(colnames(season\_months)[j])  
 return(colnames(season\_months)[j])  
 }  
 }  
}  
  
training\_set$arrival\_season <- sapply(training\_set$arrival\_date, get\_season)  
training\_set$arrival\_day <- wday(training\_set$arrival\_date)  
training\_set$arrival\_day <- ifelse(training\_set$arrival\_day %in% c(1,6,7),  
 "weekend", "weekday")  
  
  
categorical\_var <- c(1:7,10:13,15,18,19)  
for (i in categorical\_var) {  
 training\_set[[i]] <- factor(training\_set[[i]])  
}  
  
cat\_col <- colnames(training\_set[, c(1:7,10:13,15,18,19)])  
  
onehot\_encoder <- dummyVars(~ no\_of\_adults + no\_of\_children + no\_of\_weekend\_nights  
 + no\_of\_week\_nights + type\_of\_meal\_plan + required\_car\_parking\_space  
 + room\_type\_reserved + market\_segment\_type + repeated\_guest  
 + no\_of\_previous\_cancellations + no\_of\_previous\_bookings\_not\_canceled  
 + no\_of\_special\_requests + arrival\_season + arrival\_day,  
 training\_set[, c("no\_of\_adults","no\_of\_children","no\_of\_weekend\_nights",  
 "no\_of\_week\_nights","type\_of\_meal\_plan",  
 "required\_car\_parking\_space",  
 "room\_type\_reserved","market\_segment\_type",  
 "repeated\_guest","no\_of\_previous\_cancellations",  
 "no\_of\_previous\_bookings\_not\_canceled",  
 "no\_of\_special\_requests","arrival\_season",  
 "arrival\_day")],  
 levelsOnly = F,  
 fullRank = T)  
  
onehot\_enc\_training <- predict(onehot\_encoder, training\_set[, c("no\_of\_adults","no\_of\_children","no\_of\_weekend\_nights",  
 "no\_of\_week\_nights","type\_of\_meal\_plan",  
 "required\_car\_parking\_space",  
 "room\_type\_reserved","market\_segment\_type",  
 "repeated\_guest","no\_of\_previous\_cancellations",  
 "no\_of\_previous\_bookings\_not\_canceled",  
 "no\_of\_special\_requests","arrival\_season",  
 "arrival\_day")])  
  
training\_set <- cbind(training\_set, onehot\_enc\_training)  
  
####### Test Set Categorical Variables  
  
test\_set$no\_of\_previous\_bookings\_not\_canceled <- ifelse(  
 test\_set$no\_of\_previous\_bookings\_not\_canceled %in% top\_8\_previous\_not\_cancelled$no\_of\_previous\_bookings\_not\_canceled,  
 test\_set$no\_of\_previous\_bookings\_not\_canceled, "8+"  
)  
  
test\_set$no\_of\_children <- ifelse(  
 test\_set$no\_of\_children %in% top\_2\_number\_of\_children$no\_of\_children,  
 test\_set$no\_of\_children, "3+"  
)  
  
test\_set$no\_of\_previous\_cancellations <- ifelse(  
 test\_set$no\_of\_previous\_cancellations %in% top\_3\_previous\_cancellations$no\_of\_previous\_cancellations,  
 test\_set$no\_of\_previous\_cancellations, "3+"  
)  
  
test\_set$no\_of\_week\_nights <- ifelse(  
 test\_set$no\_of\_week\_nights %in% top\_8\_week\_nights$no\_of\_week\_nights,  
 test\_set$no\_of\_week\_nights, "8+"  
)  
  
test\_set$no\_of\_weekend\_nights <- ifelse(  
 test\_set$no\_of\_weekend\_nights %in% top\_6\_weekend\_nights$no\_of\_weekend\_nights,  
 test\_set$no\_of\_weekend\_nights, "6+"  
)  
  
test\_set$type\_of\_meal\_plan <- ifelse(test\_set$type\_of\_meal\_plan %in% c("meal\_plan\_1", "meal\_plan\_2"),  
 test\_set$type\_of\_meal\_plan,  
 "other")  
  
test\_set$no\_of\_special\_requests <- ifelse(test\_set$no\_of\_special\_requests %in% top\_4\_special\_requests$no\_of\_special\_requests,  
 test\_set$no\_of\_special\_requests,  
 "4+")  
  
test\_set$arrival\_date <- parse\_date\_time(test\_set$arrival\_date, "ymd")  
test\_set$booking\_date <- int\_start(interval(test\_set$arrival\_date - ddays(test\_set$lead\_time),   
 test\_set$arrival\_date))  
  
  
test\_set$arrival\_season <- sapply(test\_set$arrival\_date, get\_season)  
test\_set$arrival\_day <- wday(test\_set$arrival\_date)  
test\_set$arrival\_day <- ifelse(test\_set$arrival\_day %in% c(1,6,7),  
 "weekend", "weekday")  
  
  
categorical\_var <- c(1:7,10:13,15,18,19)  
for (i in categorical\_var) {  
 test\_set[[i]] <- factor(test\_set[[i]])  
}  
  
cat\_col <- colnames(test\_set[, c(1:7,10:13,15,18,19)])  
  
onehot\_encoder <- dummyVars(~ no\_of\_adults + no\_of\_children + no\_of\_weekend\_nights  
 + no\_of\_week\_nights + type\_of\_meal\_plan + required\_car\_parking\_space  
 + room\_type\_reserved + market\_segment\_type + repeated\_guest  
 + no\_of\_previous\_cancellations + no\_of\_previous\_bookings\_not\_canceled  
 + no\_of\_special\_requests + arrival\_season + arrival\_day,  
 test\_set[, c("no\_of\_adults","no\_of\_children","no\_of\_weekend\_nights",  
 "no\_of\_week\_nights","type\_of\_meal\_plan",  
 "required\_car\_parking\_space",  
 "room\_type\_reserved","market\_segment\_type",  
 "repeated\_guest","no\_of\_previous\_cancellations",  
 "no\_of\_previous\_bookings\_not\_canceled",  
 "no\_of\_special\_requests","arrival\_season",  
 "arrival\_day")],  
 levelsOnly = F,  
 fullRank = T)  
  
onehot\_enc\_test <- predict(onehot\_encoder, test\_set[, c("no\_of\_adults","no\_of\_children","no\_of\_weekend\_nights",  
 "no\_of\_week\_nights","type\_of\_meal\_plan",  
 "required\_car\_parking\_space",  
 "room\_type\_reserved","market\_segment\_type",  
 "repeated\_guest","no\_of\_previous\_cancellations",  
 "no\_of\_previous\_bookings\_not\_canceled",  
 "no\_of\_special\_requests","arrival\_season",  
 "arrival\_day")])  
  
test\_set <- cbind(test\_set, onehot\_enc\_test)  
  
  
##### Numerical Features  
  
test\_set[, c("lead\_time", "avg\_price\_per\_room")] <- scale(test\_set[, c("lead\_time", "avg\_price\_per\_room")],  
 center = apply(training\_set[, c("lead\_time", "avg\_price\_per\_room")], 2, mean),  
 scale = apply(training\_set[, c("lead\_time", "avg\_price\_per\_room")], 2, sd))  
training\_set[, c("lead\_time", "avg\_price\_per\_room")] <- scale(training\_set[, c("lead\_time", "avg\_price\_per\_room")])  
  
  
##### Create tensors  
  
train\_col <- ncol(training\_set)  
test\_col <- ncol(test\_set)  
  
training\_features <- array(data = unlist(training\_set[, c(8,14,20:train\_col)]),  
 dim = c(nrow(training\_set), length(c(8,14,20:train\_col))))  
training\_labels <- array(data = unlist(training\_set[, "booking\_status"]),  
 dim = nrow(training\_set))  
  
test\_features <- array(data = unlist(test\_set[, c(8,14,20:test\_col)]),  
 dim = c(nrow(test\_set), length(c(8,14,20:test\_col))))  
test\_labels <- array(data = unlist(test\_set[, "booking\_status"]),  
 dim = nrow(test\_set))

## Feed-forward Dense Neural Network

### Tensorflow Workspace

library(reticulate)

## Warning: package 'reticulate' was built under R version 4.3.2

library(tensorflow)

## Warning: package 'tensorflow' was built under R version 4.3.2

##   
## Attaching package: 'tensorflow'

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

library(keras)

## Warning: package 'keras' was built under R version 4.3.2

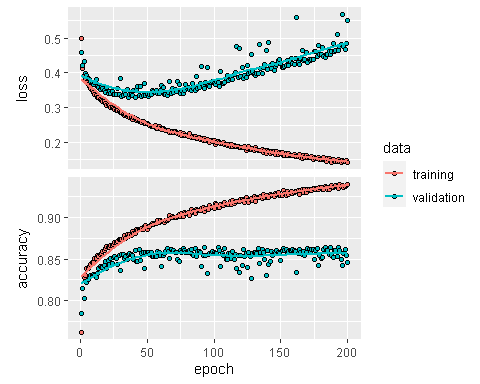
use\_virtualenv("my\_tf\_workspace")

### Overfit Model

model <- keras\_model\_sequential(list(  
 layer\_dense(units = 100, activation = "relu"),  
 layer\_dense(units = 100, activation = "relu"),  
 layer\_dense(units = 50, activation = "relu"),  
 layer\_dense(units = 1, activation = "sigmoid")  
))  
  
compile(model,  
 optimizer = "rmsprop",  
 loss = "binary\_crossentropy",  
 metrics = "accuracy")  
  
history <- fit(model, training\_features, training\_labels,  
 epochs = 200, batch\_size = 512, validation\_split = 0.33)

## Epoch 1/200  
## 36/36 - 1s - loss: 0.4979 - accuracy: 0.7622 - val\_loss: 0.4595 - val\_accuracy: 0.7844 - 880ms/epoch - 24ms/step  
## Epoch 2/200  
## 36/36 - 0s - loss: 0.4122 - accuracy: 0.8143 - val\_loss: 0.4201 - val\_accuracy: 0.8142 - 162ms/epoch - 5ms/step  
## Epoch 3/200  
## 36/36 - 0s - loss: 0.3919 - accuracy: 0.8281 - val\_loss: 0.4334 - val\_accuracy: 0.8022 - 141ms/epoch - 4ms/step  
## Epoch 4/200  
## 36/36 - 0s - loss: 0.3823 - accuracy: 0.8309 - val\_loss: 0.3976 - val\_accuracy: 0.8226 - 141ms/epoch - 4ms/step  
## Epoch 5/200  
## 36/36 - 0s - loss: 0.3722 - accuracy: 0.8390 - val\_loss: 0.3918 - val\_accuracy: 0.8219 - 141ms/epoch - 4ms/step  
## Epoch 6/200  
## 36/36 - 0s - loss: 0.3654 - accuracy: 0.8387 - val\_loss: 0.3850 - val\_accuracy: 0.8252 - 144ms/epoch - 4ms/step  
## Epoch 7/200  
## 36/36 - 0s - loss: 0.3607 - accuracy: 0.8417 - val\_loss: 0.3813 - val\_accuracy: 0.8274 - 131ms/epoch - 4ms/step  
## Epoch 8/200  
## 36/36 - 0s - loss: 0.3561 - accuracy: 0.8432 - val\_loss: 0.3754 - val\_accuracy: 0.8299 - 126ms/epoch - 3ms/step  
## Epoch 9/200  
## 36/36 - 0s - loss: 0.3482 - accuracy: 0.8464 - val\_loss: 0.3705 - val\_accuracy: 0.8303 - 111ms/epoch - 3ms/step  
## Epoch 10/200  
## 36/36 - 0s - loss: 0.3429 - accuracy: 0.8462 - val\_loss: 0.3673 - val\_accuracy: 0.8318 - 116ms/epoch - 3ms/step  
## Epoch 11/200  
## 36/36 - 0s - loss: 0.3391 - accuracy: 0.8506 - val\_loss: 0.3737 - val\_accuracy: 0.8298 - 143ms/epoch - 4ms/step  
## Epoch 12/200  
## 36/36 - 0s - loss: 0.3363 - accuracy: 0.8508 - val\_loss: 0.3802 - val\_accuracy: 0.8284 - 142ms/epoch - 4ms/step  
## Epoch 13/200  
## 36/36 - 0s - loss: 0.3305 - accuracy: 0.8541 - val\_loss: 0.3615 - val\_accuracy: 0.8333 - 143ms/epoch - 4ms/step  
## Epoch 14/200  
## 36/36 - 0s - loss: 0.3298 - accuracy: 0.8536 - val\_loss: 0.3588 - val\_accuracy: 0.8387 - 137ms/epoch - 4ms/step  
## Epoch 15/200  
## 36/36 - 0s - loss: 0.3244 - accuracy: 0.8552 - val\_loss: 0.3518 - val\_accuracy: 0.8415 - 140ms/epoch - 4ms/step  
## Epoch 16/200  
## 36/36 - 0s - loss: 0.3219 - accuracy: 0.8564 - val\_loss: 0.3574 - val\_accuracy: 0.8399 - 142ms/epoch - 4ms/step  
## Epoch 17/200  
## 36/36 - 0s - loss: 0.3165 - accuracy: 0.8593 - val\_loss: 0.3559 - val\_accuracy: 0.8405 - 141ms/epoch - 4ms/step  
## Epoch 18/200  
## 36/36 - 0s - loss: 0.3167 - accuracy: 0.8607 - val\_loss: 0.3485 - val\_accuracy: 0.8428 - 152ms/epoch - 4ms/step  
## Epoch 19/200  
## 36/36 - 0s - loss: 0.3120 - accuracy: 0.8627 - val\_loss: 0.3448 - val\_accuracy: 0.8472 - 134ms/epoch - 4ms/step  
## Epoch 20/200  
## 36/36 - 0s - loss: 0.3066 - accuracy: 0.8650 - val\_loss: 0.3584 - val\_accuracy: 0.8388 - 132ms/epoch - 4ms/step  
## Epoch 21/200  
## 36/36 - 0s - loss: 0.3087 - accuracy: 0.8641 - val\_loss: 0.3413 - val\_accuracy: 0.8483 - 132ms/epoch - 4ms/step  
## Epoch 22/200  
## 36/36 - 0s - loss: 0.3053 - accuracy: 0.8641 - val\_loss: 0.3456 - val\_accuracy: 0.8438 - 123ms/epoch - 3ms/step  
## Epoch 23/200  
## 36/36 - 0s - loss: 0.3017 - accuracy: 0.8666 - val\_loss: 0.3394 - val\_accuracy: 0.8506 - 119ms/epoch - 3ms/step  
## Epoch 24/200  
## 36/36 - 0s - loss: 0.3001 - accuracy: 0.8665 - val\_loss: 0.3458 - val\_accuracy: 0.8454 - 116ms/epoch - 3ms/step  
## Epoch 25/200  
## 36/36 - 0s - loss: 0.2981 - accuracy: 0.8675 - val\_loss: 0.3372 - val\_accuracy: 0.8517 - 127ms/epoch - 4ms/step  
## Epoch 26/200  
## 36/36 - 0s - loss: 0.2951 - accuracy: 0.8702 - val\_loss: 0.3370 - val\_accuracy: 0.8527 - 143ms/epoch - 4ms/step  
## Epoch 27/200  
## 36/36 - 0s - loss: 0.2928 - accuracy: 0.8706 - val\_loss: 0.3368 - val\_accuracy: 0.8511 - 139ms/epoch - 4ms/step  
## Epoch 28/200  
## 36/36 - 0s - loss: 0.2914 - accuracy: 0.8716 - val\_loss: 0.3564 - val\_accuracy: 0.8436 - 140ms/epoch - 4ms/step  
## Epoch 29/200  
## 36/36 - 0s - loss: 0.2890 - accuracy: 0.8734 - val\_loss: 0.3380 - val\_accuracy: 0.8540 - 133ms/epoch - 4ms/step  
## Epoch 30/200  
## 36/36 - 0s - loss: 0.2861 - accuracy: 0.8732 - val\_loss: 0.3833 - val\_accuracy: 0.8369 - 138ms/epoch - 4ms/step  
## Epoch 31/200  
## 36/36 - 0s - loss: 0.2851 - accuracy: 0.8735 - val\_loss: 0.3334 - val\_accuracy: 0.8559 - 139ms/epoch - 4ms/step  
## Epoch 32/200  
## 36/36 - 0s - loss: 0.2818 - accuracy: 0.8753 - val\_loss: 0.3348 - val\_accuracy: 0.8554 - 140ms/epoch - 4ms/step  
## Epoch 33/200  
## 36/36 - 0s - loss: 0.2791 - accuracy: 0.8782 - val\_loss: 0.3440 - val\_accuracy: 0.8504 - 131ms/epoch - 4ms/step  
## Epoch 34/200  
## 36/36 - 0s - loss: 0.2811 - accuracy: 0.8752 - val\_loss: 0.3390 - val\_accuracy: 0.8535 - 140ms/epoch - 4ms/step  
## Epoch 35/200  
## 36/36 - 0s - loss: 0.2739 - accuracy: 0.8796 - val\_loss: 0.3508 - val\_accuracy: 0.8494 - 136ms/epoch - 4ms/step  
## Epoch 36/200  
## 36/36 - 0s - loss: 0.2763 - accuracy: 0.8779 - val\_loss: 0.3305 - val\_accuracy: 0.8559 - 144ms/epoch - 4ms/step  
## Epoch 37/200  
## 36/36 - 0s - loss: 0.2724 - accuracy: 0.8801 - val\_loss: 0.3368 - val\_accuracy: 0.8564 - 121ms/epoch - 3ms/step  
## Epoch 38/200  
## 36/36 - 0s - loss: 0.2733 - accuracy: 0.8781 - val\_loss: 0.3374 - val\_accuracy: 0.8551 - 113ms/epoch - 3ms/step  
## Epoch 39/200  
## 36/36 - 0s - loss: 0.2671 - accuracy: 0.8829 - val\_loss: 0.3355 - val\_accuracy: 0.8544 - 126ms/epoch - 3ms/step  
## Epoch 40/200  
## 36/36 - 0s - loss: 0.2705 - accuracy: 0.8805 - val\_loss: 0.3482 - val\_accuracy: 0.8492 - 139ms/epoch - 4ms/step  
## Epoch 41/200  
## 36/36 - 0s - loss: 0.2695 - accuracy: 0.8813 - val\_loss: 0.3286 - val\_accuracy: 0.8603 - 145ms/epoch - 4ms/step  
## Epoch 42/200  
## 36/36 - 0s - loss: 0.2631 - accuracy: 0.8853 - val\_loss: 0.3529 - val\_accuracy: 0.8473 - 150ms/epoch - 4ms/step  
## Epoch 43/200  
## 36/36 - 0s - loss: 0.2650 - accuracy: 0.8840 - val\_loss: 0.3603 - val\_accuracy: 0.8475 - 141ms/epoch - 4ms/step  
## Epoch 44/200  
## 36/36 - 0s - loss: 0.2600 - accuracy: 0.8854 - val\_loss: 0.3337 - val\_accuracy: 0.8612 - 144ms/epoch - 4ms/step  
## Epoch 45/200  
## 36/36 - 0s - loss: 0.2594 - accuracy: 0.8862 - val\_loss: 0.3596 - val\_accuracy: 0.8489 - 433ms/epoch - 12ms/step  
## Epoch 46/200  
## 36/36 - 0s - loss: 0.2573 - accuracy: 0.8873 - val\_loss: 0.3322 - val\_accuracy: 0.8570 - 135ms/epoch - 4ms/step  
## Epoch 47/200  
## 36/36 - 0s - loss: 0.2602 - accuracy: 0.8863 - val\_loss: 0.3340 - val\_accuracy: 0.8586 - 143ms/epoch - 4ms/step  
## Epoch 48/200  
## 36/36 - 0s - loss: 0.2533 - accuracy: 0.8890 - val\_loss: 0.3688 - val\_accuracy: 0.8387 - 134ms/epoch - 4ms/step  
## Epoch 49/200  
## 36/36 - 0s - loss: 0.2543 - accuracy: 0.8904 - val\_loss: 0.3362 - val\_accuracy: 0.8582 - 144ms/epoch - 4ms/step  
## Epoch 50/200  
## 36/36 - 0s - loss: 0.2561 - accuracy: 0.8861 - val\_loss: 0.3365 - val\_accuracy: 0.8576 - 110ms/epoch - 3ms/step  
## Epoch 51/200  
## 36/36 - 0s - loss: 0.2521 - accuracy: 0.8888 - val\_loss: 0.3332 - val\_accuracy: 0.8582 - 120ms/epoch - 3ms/step  
## Epoch 52/200  
## 36/36 - 0s - loss: 0.2503 - accuracy: 0.8895 - val\_loss: 0.3370 - val\_accuracy: 0.8555 - 119ms/epoch - 3ms/step  
## Epoch 53/200  
## 36/36 - 0s - loss: 0.2474 - accuracy: 0.8915 - val\_loss: 0.3369 - val\_accuracy: 0.8585 - 138ms/epoch - 4ms/step  
## Epoch 54/200  
## 36/36 - 0s - loss: 0.2464 - accuracy: 0.8924 - val\_loss: 0.3485 - val\_accuracy: 0.8532 - 142ms/epoch - 4ms/step  
## Epoch 55/200  
## 36/36 - 0s - loss: 0.2470 - accuracy: 0.8918 - val\_loss: 0.3455 - val\_accuracy: 0.8528 - 143ms/epoch - 4ms/step  
## Epoch 56/200  
## 36/36 - 0s - loss: 0.2477 - accuracy: 0.8930 - val\_loss: 0.3444 - val\_accuracy: 0.8536 - 140ms/epoch - 4ms/step  
## Epoch 57/200  
## 36/36 - 0s - loss: 0.2414 - accuracy: 0.8927 - val\_loss: 0.3490 - val\_accuracy: 0.8530 - 140ms/epoch - 4ms/step  
## Epoch 58/200  
## 36/36 - 0s - loss: 0.2392 - accuracy: 0.8969 - val\_loss: 0.3411 - val\_accuracy: 0.8594 - 140ms/epoch - 4ms/step  
## Epoch 59/200  
## 36/36 - 0s - loss: 0.2451 - accuracy: 0.8937 - val\_loss: 0.3420 - val\_accuracy: 0.8575 - 136ms/epoch - 4ms/step  
## Epoch 60/200  
## 36/36 - 0s - loss: 0.2399 - accuracy: 0.8963 - val\_loss: 0.3508 - val\_accuracy: 0.8548 - 142ms/epoch - 4ms/step  
## Epoch 61/200  
## 36/36 - 0s - loss: 0.2383 - accuracy: 0.8958 - val\_loss: 0.3644 - val\_accuracy: 0.8410 - 150ms/epoch - 4ms/step  
## Epoch 62/200  
## 36/36 - 0s - loss: 0.2402 - accuracy: 0.8948 - val\_loss: 0.3588 - val\_accuracy: 0.8543 - 130ms/epoch - 4ms/step  
## Epoch 63/200  
## 36/36 - 0s - loss: 0.2382 - accuracy: 0.8937 - val\_loss: 0.3376 - val\_accuracy: 0.8566 - 132ms/epoch - 4ms/step  
## Epoch 64/200  
## 36/36 - 0s - loss: 0.2351 - accuracy: 0.8970 - val\_loss: 0.3378 - val\_accuracy: 0.8596 - 119ms/epoch - 3ms/step  
## Epoch 65/200  
## 36/36 - 0s - loss: 0.2353 - accuracy: 0.8963 - val\_loss: 0.3350 - val\_accuracy: 0.8631 - 112ms/epoch - 3ms/step  
## Epoch 66/200  
## 36/36 - 0s - loss: 0.2358 - accuracy: 0.8980 - val\_loss: 0.3425 - val\_accuracy: 0.8596 - 123ms/epoch - 3ms/step  
## Epoch 67/200  
## 36/36 - 0s - loss: 0.2280 - accuracy: 0.9022 - val\_loss: 0.3760 - val\_accuracy: 0.8503 - 133ms/epoch - 4ms/step  
## Epoch 68/200  
## 36/36 - 0s - loss: 0.2301 - accuracy: 0.8998 - val\_loss: 0.3398 - val\_accuracy: 0.8615 - 146ms/epoch - 4ms/step  
## Epoch 69/200  
## 36/36 - 0s - loss: 0.2303 - accuracy: 0.9009 - val\_loss: 0.3539 - val\_accuracy: 0.8589 - 147ms/epoch - 4ms/step  
## Epoch 70/200  
## 36/36 - 0s - loss: 0.2283 - accuracy: 0.9007 - val\_loss: 0.3448 - val\_accuracy: 0.8610 - 149ms/epoch - 4ms/step  
## Epoch 71/200  
## 36/36 - 0s - loss: 0.2316 - accuracy: 0.9002 - val\_loss: 0.3527 - val\_accuracy: 0.8584 - 134ms/epoch - 4ms/step  
## Epoch 72/200  
## 36/36 - 0s - loss: 0.2278 - accuracy: 0.9006 - val\_loss: 0.3457 - val\_accuracy: 0.8584 - 136ms/epoch - 4ms/step  
## Epoch 73/200  
## 36/36 - 0s - loss: 0.2264 - accuracy: 0.9011 - val\_loss: 0.3738 - val\_accuracy: 0.8491 - 148ms/epoch - 4ms/step  
## Epoch 74/200  
## 36/36 - 0s - loss: 0.2243 - accuracy: 0.9039 - val\_loss: 0.3655 - val\_accuracy: 0.8476 - 136ms/epoch - 4ms/step  
## Epoch 75/200  
## 36/36 - 0s - loss: 0.2295 - accuracy: 0.9015 - val\_loss: 0.3419 - val\_accuracy: 0.8610 - 135ms/epoch - 4ms/step  
## Epoch 76/200  
## 36/36 - 0s - loss: 0.2202 - accuracy: 0.9053 - val\_loss: 0.3492 - val\_accuracy: 0.8584 - 140ms/epoch - 4ms/step  
## Epoch 77/200  
## 36/36 - 0s - loss: 0.2267 - accuracy: 0.9018 - val\_loss: 0.3419 - val\_accuracy: 0.8631 - 142ms/epoch - 4ms/step  
## Epoch 78/200  
## 36/36 - 0s - loss: 0.2198 - accuracy: 0.9057 - val\_loss: 0.3507 - val\_accuracy: 0.8605 - 136ms/epoch - 4ms/step  
## Epoch 79/200  
## 36/36 - 0s - loss: 0.2193 - accuracy: 0.9056 - val\_loss: 0.3668 - val\_accuracy: 0.8577 - 114ms/epoch - 3ms/step  
## Epoch 80/200  
## 36/36 - 0s - loss: 0.2205 - accuracy: 0.9052 - val\_loss: 0.3541 - val\_accuracy: 0.8577 - 125ms/epoch - 3ms/step  
## Epoch 81/200  
## 36/36 - 0s - loss: 0.2163 - accuracy: 0.9063 - val\_loss: 0.3571 - val\_accuracy: 0.8586 - 127ms/epoch - 4ms/step  
## Epoch 82/200  
## 36/36 - 0s - loss: 0.2260 - accuracy: 0.9026 - val\_loss: 0.3513 - val\_accuracy: 0.8600 - 143ms/epoch - 4ms/step  
## Epoch 83/200  
## 36/36 - 0s - loss: 0.2134 - accuracy: 0.9098 - val\_loss: 0.3503 - val\_accuracy: 0.8613 - 144ms/epoch - 4ms/step  
## Epoch 84/200  
## 36/36 - 0s - loss: 0.2165 - accuracy: 0.9060 - val\_loss: 0.3686 - val\_accuracy: 0.8553 - 144ms/epoch - 4ms/step  
## Epoch 85/200  
## 36/36 - 0s - loss: 0.2159 - accuracy: 0.9070 - val\_loss: 0.3565 - val\_accuracy: 0.8550 - 143ms/epoch - 4ms/step  
## Epoch 86/200  
## 36/36 - 0s - loss: 0.2179 - accuracy: 0.9048 - val\_loss: 0.3492 - val\_accuracy: 0.8628 - 140ms/epoch - 4ms/step  
## Epoch 87/200  
## 36/36 - 0s - loss: 0.2128 - accuracy: 0.9103 - val\_loss: 0.3897 - val\_accuracy: 0.8533 - 147ms/epoch - 4ms/step  
## Epoch 88/200  
## 36/36 - 0s - loss: 0.2129 - accuracy: 0.9092 - val\_loss: 0.3540 - val\_accuracy: 0.8590 - 132ms/epoch - 4ms/step  
## Epoch 89/200  
## 36/36 - 0s - loss: 0.2139 - accuracy: 0.9083 - val\_loss: 0.3580 - val\_accuracy: 0.8594 - 139ms/epoch - 4ms/step  
## Epoch 90/200  
## 36/36 - 0s - loss: 0.2097 - accuracy: 0.9089 - val\_loss: 0.3694 - val\_accuracy: 0.8579 - 129ms/epoch - 4ms/step  
## Epoch 91/200  
## 36/36 - 0s - loss: 0.2113 - accuracy: 0.9107 - val\_loss: 0.4192 - val\_accuracy: 0.8407 - 135ms/epoch - 4ms/step  
## Epoch 92/200  
## 36/36 - 0s - loss: 0.2054 - accuracy: 0.9119 - val\_loss: 0.3595 - val\_accuracy: 0.8547 - 136ms/epoch - 4ms/step  
## Epoch 93/200  
## 36/36 - 0s - loss: 0.2106 - accuracy: 0.9077 - val\_loss: 0.3538 - val\_accuracy: 0.8635 - 130ms/epoch - 4ms/step  
## Epoch 94/200  
## 36/36 - 0s - loss: 0.2053 - accuracy: 0.9116 - val\_loss: 0.3735 - val\_accuracy: 0.8541 - 109ms/epoch - 3ms/step  
## Epoch 95/200  
## 36/36 - 0s - loss: 0.2080 - accuracy: 0.9099 - val\_loss: 0.4153 - val\_accuracy: 0.8324 - 120ms/epoch - 3ms/step  
## Epoch 96/200  
## 36/36 - 0s - loss: 0.2041 - accuracy: 0.9104 - val\_loss: 0.3631 - val\_accuracy: 0.8554 - 119ms/epoch - 3ms/step  
## Epoch 97/200  
## 36/36 - 0s - loss: 0.2016 - accuracy: 0.9150 - val\_loss: 0.3619 - val\_accuracy: 0.8570 - 148ms/epoch - 4ms/step  
## Epoch 98/200  
## 36/36 - 0s - loss: 0.2059 - accuracy: 0.9125 - val\_loss: 0.3723 - val\_accuracy: 0.8583 - 125ms/epoch - 3ms/step  
## Epoch 99/200  
## 36/36 - 0s - loss: 0.2028 - accuracy: 0.9117 - val\_loss: 0.3748 - val\_accuracy: 0.8592 - 137ms/epoch - 4ms/step  
## Epoch 100/200  
## 36/36 - 0s - loss: 0.2056 - accuracy: 0.9106 - val\_loss: 0.3685 - val\_accuracy: 0.8576 - 133ms/epoch - 4ms/step  
## Epoch 101/200  
## 36/36 - 0s - loss: 0.1955 - accuracy: 0.9176 - val\_loss: 0.3664 - val\_accuracy: 0.8574 - 141ms/epoch - 4ms/step  
## Epoch 102/200  
## 36/36 - 0s - loss: 0.1991 - accuracy: 0.9160 - val\_loss: 0.3755 - val\_accuracy: 0.8570 - 137ms/epoch - 4ms/step  
## Epoch 103/200  
## 36/36 - 0s - loss: 0.2018 - accuracy: 0.9128 - val\_loss: 0.3713 - val\_accuracy: 0.8640 - 136ms/epoch - 4ms/step  
## Epoch 104/200  
## 36/36 - 0s - loss: 0.2005 - accuracy: 0.9127 - val\_loss: 0.3622 - val\_accuracy: 0.8606 - 140ms/epoch - 4ms/step  
## Epoch 105/200  
## 36/36 - 0s - loss: 0.1977 - accuracy: 0.9156 - val\_loss: 0.4042 - val\_accuracy: 0.8411 - 134ms/epoch - 4ms/step  
## Epoch 106/200  
## 36/36 - 0s - loss: 0.1973 - accuracy: 0.9141 - val\_loss: 0.3705 - val\_accuracy: 0.8572 - 131ms/epoch - 4ms/step  
## Epoch 107/200  
## 36/36 - 0s - loss: 0.1991 - accuracy: 0.9149 - val\_loss: 0.3696 - val\_accuracy: 0.8548 - 134ms/epoch - 4ms/step  
## Epoch 108/200  
## 36/36 - 0s - loss: 0.1932 - accuracy: 0.9181 - val\_loss: 0.3723 - val\_accuracy: 0.8630 - 121ms/epoch - 3ms/step  
## Epoch 109/200  
## 36/36 - 0s - loss: 0.1955 - accuracy: 0.9159 - val\_loss: 0.3828 - val\_accuracy: 0.8496 - 108ms/epoch - 3ms/step  
## Epoch 110/200  
## 36/36 - 0s - loss: 0.1909 - accuracy: 0.9156 - val\_loss: 0.3921 - val\_accuracy: 0.8498 - 114ms/epoch - 3ms/step  
## Epoch 111/200  
## 36/36 - 0s - loss: 0.1946 - accuracy: 0.9157 - val\_loss: 0.3792 - val\_accuracy: 0.8564 - 117ms/epoch - 3ms/step  
## Epoch 112/200  
## 36/36 - 0s - loss: 0.1886 - accuracy: 0.9193 - val\_loss: 0.3833 - val\_accuracy: 0.8564 - 147ms/epoch - 4ms/step  
## Epoch 113/200  
## 36/36 - 0s - loss: 0.1958 - accuracy: 0.9160 - val\_loss: 0.3867 - val\_accuracy: 0.8504 - 146ms/epoch - 4ms/step  
## Epoch 114/200  
## 36/36 - 0s - loss: 0.1873 - accuracy: 0.9204 - val\_loss: 0.3799 - val\_accuracy: 0.8596 - 147ms/epoch - 4ms/step  
## Epoch 115/200  
## 36/36 - 0s - loss: 0.1926 - accuracy: 0.9177 - val\_loss: 0.3945 - val\_accuracy: 0.8497 - 143ms/epoch - 4ms/step  
## Epoch 116/200  
## 36/36 - 0s - loss: 0.1876 - accuracy: 0.9198 - val\_loss: 0.4010 - val\_accuracy: 0.8554 - 141ms/epoch - 4ms/step  
## Epoch 117/200  
## 36/36 - 0s - loss: 0.1887 - accuracy: 0.9200 - val\_loss: 0.4746 - val\_accuracy: 0.8384 - 148ms/epoch - 4ms/step  
## Epoch 118/200  
## 36/36 - 0s - loss: 0.1881 - accuracy: 0.9192 - val\_loss: 0.3876 - val\_accuracy: 0.8546 - 150ms/epoch - 4ms/step  
## Epoch 119/200  
## 36/36 - 0s - loss: 0.1870 - accuracy: 0.9192 - val\_loss: 0.4698 - val\_accuracy: 0.8344 - 151ms/epoch - 4ms/step  
## Epoch 120/200  
## 36/36 - 0s - loss: 0.1851 - accuracy: 0.9215 - val\_loss: 0.3898 - val\_accuracy: 0.8550 - 128ms/epoch - 4ms/step  
## Epoch 121/200  
## 36/36 - 0s - loss: 0.1854 - accuracy: 0.9216 - val\_loss: 0.4144 - val\_accuracy: 0.8418 - 138ms/epoch - 4ms/step  
## Epoch 122/200  
## 36/36 - 0s - loss: 0.1867 - accuracy: 0.9205 - val\_loss: 0.3848 - val\_accuracy: 0.8622 - 133ms/epoch - 4ms/step  
## Epoch 123/200  
## 36/36 - 0s - loss: 0.1893 - accuracy: 0.9184 - val\_loss: 0.3915 - val\_accuracy: 0.8528 - 144ms/epoch - 4ms/step  
## Epoch 124/200  
## 36/36 - 0s - loss: 0.1781 - accuracy: 0.9239 - val\_loss: 0.3871 - val\_accuracy: 0.8565 - 116ms/epoch - 3ms/step  
## Epoch 125/200  
## 36/36 - 0s - loss: 0.1831 - accuracy: 0.9211 - val\_loss: 0.4219 - val\_accuracy: 0.8401 - 121ms/epoch - 3ms/step  
## Epoch 126/200  
## 36/36 - 0s - loss: 0.1901 - accuracy: 0.9184 - val\_loss: 0.3993 - val\_accuracy: 0.8579 - 127ms/epoch - 4ms/step  
## Epoch 127/200  
## 36/36 - 0s - loss: 0.1813 - accuracy: 0.9216 - val\_loss: 0.4006 - val\_accuracy: 0.8527 - 150ms/epoch - 4ms/step  
## Epoch 128/200  
## 36/36 - 0s - loss: 0.1805 - accuracy: 0.9230 - val\_loss: 0.4541 - val\_accuracy: 0.8265 - 150ms/epoch - 4ms/step  
## Epoch 129/200  
## 36/36 - 0s - loss: 0.1829 - accuracy: 0.9208 - val\_loss: 0.4034 - val\_accuracy: 0.8565 - 144ms/epoch - 4ms/step  
## Epoch 130/200  
## 36/36 - 0s - loss: 0.1767 - accuracy: 0.9254 - val\_loss: 0.3962 - val\_accuracy: 0.8577 - 138ms/epoch - 4ms/step  
## Epoch 131/200  
## 36/36 - 0s - loss: 0.1845 - accuracy: 0.9195 - val\_loss: 0.3935 - val\_accuracy: 0.8582 - 128ms/epoch - 4ms/step  
## Epoch 132/200  
## 36/36 - 0s - loss: 0.1807 - accuracy: 0.9240 - val\_loss: 0.3969 - val\_accuracy: 0.8571 - 128ms/epoch - 4ms/step  
## Epoch 133/200  
## 36/36 - 0s - loss: 0.1774 - accuracy: 0.9241 - val\_loss: 0.3975 - val\_accuracy: 0.8631 - 143ms/epoch - 4ms/step  
## Epoch 134/200  
## 36/36 - 0s - loss: 0.1766 - accuracy: 0.9246 - val\_loss: 0.4812 - val\_accuracy: 0.8450 - 148ms/epoch - 4ms/step  
## Epoch 135/200  
## 36/36 - 0s - loss: 0.1761 - accuracy: 0.9252 - val\_loss: 0.4119 - val\_accuracy: 0.8580 - 150ms/epoch - 4ms/step  
## Epoch 136/200  
## 36/36 - 0s - loss: 0.1789 - accuracy: 0.9233 - val\_loss: 0.3970 - val\_accuracy: 0.8616 - 150ms/epoch - 4ms/step  
## Epoch 137/200  
## 36/36 - 0s - loss: 0.1727 - accuracy: 0.9274 - val\_loss: 0.4153 - val\_accuracy: 0.8576 - 139ms/epoch - 4ms/step  
## Epoch 138/200  
## 36/36 - 0s - loss: 0.1782 - accuracy: 0.9239 - val\_loss: 0.4235 - val\_accuracy: 0.8496 - 115ms/epoch - 3ms/step  
## Epoch 139/200  
## 36/36 - 0s - loss: 0.1697 - accuracy: 0.9271 - val\_loss: 0.4629 - val\_accuracy: 0.8303 - 113ms/epoch - 3ms/step  
## Epoch 140/200  
## 36/36 - 0s - loss: 0.1772 - accuracy: 0.9242 - val\_loss: 0.4090 - val\_accuracy: 0.8604 - 130ms/epoch - 4ms/step  
## Epoch 141/200  
## 36/36 - 0s - loss: 0.1754 - accuracy: 0.9247 - val\_loss: 0.4860 - val\_accuracy: 0.8435 - 148ms/epoch - 4ms/step  
## Epoch 142/200  
## 36/36 - 0s - loss: 0.1697 - accuracy: 0.9278 - val\_loss: 0.4224 - val\_accuracy: 0.8469 - 150ms/epoch - 4ms/step  
## Epoch 143/200  
## 36/36 - 0s - loss: 0.1734 - accuracy: 0.9265 - val\_loss: 0.4112 - val\_accuracy: 0.8592 - 149ms/epoch - 4ms/step  
## Epoch 144/200  
## 36/36 - 0s - loss: 0.1711 - accuracy: 0.9262 - val\_loss: 0.4282 - val\_accuracy: 0.8543 - 151ms/epoch - 4ms/step  
## Epoch 145/200  
## 36/36 - 0s - loss: 0.1755 - accuracy: 0.9254 - val\_loss: 0.4060 - val\_accuracy: 0.8575 - 149ms/epoch - 4ms/step  
## Epoch 146/200  
## 36/36 - 0s - loss: 0.1671 - accuracy: 0.9294 - val\_loss: 0.4185 - val\_accuracy: 0.8571 - 157ms/epoch - 4ms/step  
## Epoch 147/200  
## 36/36 - 0s - loss: 0.1738 - accuracy: 0.9276 - val\_loss: 0.4242 - val\_accuracy: 0.8606 - 149ms/epoch - 4ms/step  
## Epoch 148/200  
## 36/36 - 0s - loss: 0.1656 - accuracy: 0.9298 - val\_loss: 0.4349 - val\_accuracy: 0.8581 - 155ms/epoch - 4ms/step  
## Epoch 149/200  
## 36/36 - 0s - loss: 0.1708 - accuracy: 0.9272 - val\_loss: 0.4390 - val\_accuracy: 0.8569 - 149ms/epoch - 4ms/step  
## Epoch 150/200  
## 36/36 - 0s - loss: 0.1683 - accuracy: 0.9270 - val\_loss: 0.4265 - val\_accuracy: 0.8483 - 142ms/epoch - 4ms/step  
## Epoch 151/200  
## 36/36 - 0s - loss: 0.1719 - accuracy: 0.9268 - val\_loss: 0.4153 - val\_accuracy: 0.8602 - 125ms/epoch - 3ms/step  
## Epoch 152/200  
## 36/36 - 0s - loss: 0.1635 - accuracy: 0.9326 - val\_loss: 0.4163 - val\_accuracy: 0.8625 - 110ms/epoch - 3ms/step  
## Epoch 153/200  
## 36/36 - 0s - loss: 0.1631 - accuracy: 0.9289 - val\_loss: 0.4175 - val\_accuracy: 0.8606 - 114ms/epoch - 3ms/step  
## Epoch 154/200  
## 36/36 - 0s - loss: 0.1691 - accuracy: 0.9264 - val\_loss: 0.4198 - val\_accuracy: 0.8611 - 120ms/epoch - 3ms/step  
## Epoch 155/200  
## 36/36 - 0s - loss: 0.1619 - accuracy: 0.9318 - val\_loss: 0.4348 - val\_accuracy: 0.8579 - 147ms/epoch - 4ms/step  
## Epoch 156/200  
## 36/36 - 0s - loss: 0.1676 - accuracy: 0.9282 - val\_loss: 0.4210 - val\_accuracy: 0.8598 - 148ms/epoch - 4ms/step  
## Epoch 157/200  
## 36/36 - 0s - loss: 0.1639 - accuracy: 0.9301 - val\_loss: 0.4332 - val\_accuracy: 0.8603 - 148ms/epoch - 4ms/step  
## Epoch 158/200  
## 36/36 - 0s - loss: 0.1628 - accuracy: 0.9312 - val\_loss: 0.4416 - val\_accuracy: 0.8602 - 147ms/epoch - 4ms/step  
## Epoch 159/200  
## 36/36 - 0s - loss: 0.1630 - accuracy: 0.9296 - val\_loss: 0.4268 - val\_accuracy: 0.8622 - 147ms/epoch - 4ms/step  
## Epoch 160/200  
## 36/36 - 0s - loss: 0.1659 - accuracy: 0.9302 - val\_loss: 0.4245 - val\_accuracy: 0.8600 - 144ms/epoch - 4ms/step  
## Epoch 161/200  
## 36/36 - 0s - loss: 0.1602 - accuracy: 0.9316 - val\_loss: 0.4419 - val\_accuracy: 0.8594 - 145ms/epoch - 4ms/step  
## Epoch 162/200  
## 36/36 - 0s - loss: 0.1607 - accuracy: 0.9317 - val\_loss: 0.5593 - val\_accuracy: 0.8343 - 135ms/epoch - 4ms/step  
## Epoch 163/200  
## 36/36 - 0s - loss: 0.1594 - accuracy: 0.9324 - val\_loss: 0.4298 - val\_accuracy: 0.8573 - 136ms/epoch - 4ms/step  
## Epoch 164/200  
## 36/36 - 0s - loss: 0.1667 - accuracy: 0.9283 - val\_loss: 0.4236 - val\_accuracy: 0.8620 - 139ms/epoch - 4ms/step  
## Epoch 165/200  
## 36/36 - 0s - loss: 0.1574 - accuracy: 0.9329 - val\_loss: 0.4422 - val\_accuracy: 0.8580 - 131ms/epoch - 4ms/step  
## Epoch 166/200  
## 36/36 - 0s - loss: 0.1540 - accuracy: 0.9345 - val\_loss: 0.4630 - val\_accuracy: 0.8552 - 110ms/epoch - 3ms/step  
## Epoch 167/200  
## 36/36 - 0s - loss: 0.1628 - accuracy: 0.9306 - val\_loss: 0.4362 - val\_accuracy: 0.8588 - 111ms/epoch - 3ms/step  
## Epoch 168/200  
## 36/36 - 0s - loss: 0.1557 - accuracy: 0.9335 - val\_loss: 0.4392 - val\_accuracy: 0.8548 - 118ms/epoch - 3ms/step  
## Epoch 169/200  
## 36/36 - 0s - loss: 0.1621 - accuracy: 0.9300 - val\_loss: 0.4598 - val\_accuracy: 0.8577 - 139ms/epoch - 4ms/step  
## Epoch 170/200  
## 36/36 - 0s - loss: 0.1574 - accuracy: 0.9337 - val\_loss: 0.4307 - val\_accuracy: 0.8624 - 148ms/epoch - 4ms/step  
## Epoch 171/200  
## 36/36 - 0s - loss: 0.1573 - accuracy: 0.9339 - val\_loss: 0.4393 - val\_accuracy: 0.8614 - 144ms/epoch - 4ms/step  
## Epoch 172/200  
## 36/36 - 0s - loss: 0.1558 - accuracy: 0.9329 - val\_loss: 0.4353 - val\_accuracy: 0.8598 - 141ms/epoch - 4ms/step  
## Epoch 173/200  
## 36/36 - 0s - loss: 0.1586 - accuracy: 0.9322 - val\_loss: 0.4394 - val\_accuracy: 0.8612 - 136ms/epoch - 4ms/step  
## Epoch 174/200  
## 36/36 - 0s - loss: 0.1512 - accuracy: 0.9356 - val\_loss: 0.4392 - val\_accuracy: 0.8620 - 145ms/epoch - 4ms/step  
## Epoch 175/200  
## 36/36 - 0s - loss: 0.1557 - accuracy: 0.9349 - val\_loss: 0.4610 - val\_accuracy: 0.8491 - 145ms/epoch - 4ms/step  
## Epoch 176/200  
## 36/36 - 0s - loss: 0.1548 - accuracy: 0.9340 - val\_loss: 0.4604 - val\_accuracy: 0.8606 - 140ms/epoch - 4ms/step  
## Epoch 177/200  
## 36/36 - 0s - loss: 0.1547 - accuracy: 0.9350 - val\_loss: 0.4480 - val\_accuracy: 0.8631 - 141ms/epoch - 4ms/step  
## Epoch 178/200  
## 36/36 - 0s - loss: 0.1559 - accuracy: 0.9334 - val\_loss: 0.4784 - val\_accuracy: 0.8429 - 143ms/epoch - 4ms/step  
## Epoch 179/200  
## 36/36 - 0s - loss: 0.1517 - accuracy: 0.9364 - val\_loss: 0.4519 - val\_accuracy: 0.8547 - 147ms/epoch - 4ms/step  
## Epoch 180/200  
## 36/36 - 0s - loss: 0.1506 - accuracy: 0.9359 - val\_loss: 0.4529 - val\_accuracy: 0.8511 - 132ms/epoch - 4ms/step  
## Epoch 181/200  
## 36/36 - 0s - loss: 0.1529 - accuracy: 0.9341 - val\_loss: 0.4738 - val\_accuracy: 0.8518 - 112ms/epoch - 3ms/step  
## Epoch 182/200  
## 36/36 - 0s - loss: 0.1574 - accuracy: 0.9335 - val\_loss: 0.4457 - val\_accuracy: 0.8614 - 112ms/epoch - 3ms/step  
## Epoch 183/200  
## 36/36 - 0s - loss: 0.1478 - accuracy: 0.9371 - val\_loss: 0.4450 - val\_accuracy: 0.8643 - 121ms/epoch - 3ms/step  
## Epoch 184/200  
## 36/36 - 0s - loss: 0.1469 - accuracy: 0.9370 - val\_loss: 0.4533 - val\_accuracy: 0.8601 - 139ms/epoch - 4ms/step  
## Epoch 185/200  
## 36/36 - 0s - loss: 0.1531 - accuracy: 0.9344 - val\_loss: 0.4533 - val\_accuracy: 0.8600 - 145ms/epoch - 4ms/step  
## Epoch 186/200  
## 36/36 - 0s - loss: 0.1530 - accuracy: 0.9335 - val\_loss: 0.4625 - val\_accuracy: 0.8559 - 147ms/epoch - 4ms/step  
## Epoch 187/200  
## 36/36 - 0s - loss: 0.1489 - accuracy: 0.9374 - val\_loss: 0.4539 - val\_accuracy: 0.8643 - 147ms/epoch - 4ms/step  
## Epoch 188/200  
## 36/36 - 0s - loss: 0.1440 - accuracy: 0.9394 - val\_loss: 0.4613 - val\_accuracy: 0.8542 - 142ms/epoch - 4ms/step  
## Epoch 189/200  
## 36/36 - 0s - loss: 0.1510 - accuracy: 0.9362 - val\_loss: 0.4831 - val\_accuracy: 0.8520 - 145ms/epoch - 4ms/step  
## Epoch 190/200  
## 36/36 - 0s - loss: 0.1449 - accuracy: 0.9376 - val\_loss: 0.4617 - val\_accuracy: 0.8610 - 143ms/epoch - 4ms/step  
## Epoch 191/200  
## 36/36 - 0s - loss: 0.1460 - accuracy: 0.9393 - val\_loss: 0.4654 - val\_accuracy: 0.8599 - 142ms/epoch - 4ms/step  
## Epoch 192/200  
## 36/36 - 0s - loss: 0.1470 - accuracy: 0.9379 - val\_loss: 0.5089 - val\_accuracy: 0.8528 - 141ms/epoch - 4ms/step  
## Epoch 193/200  
## 36/36 - 0s - loss: 0.1513 - accuracy: 0.9357 - val\_loss: 0.4649 - val\_accuracy: 0.8639 - 139ms/epoch - 4ms/step  
## Epoch 194/200  
## 36/36 - 0s - loss: 0.1446 - accuracy: 0.9399 - val\_loss: 0.4747 - val\_accuracy: 0.8605 - 137ms/epoch - 4ms/step  
## Epoch 195/200  
## 36/36 - 0s - loss: 0.1432 - accuracy: 0.9381 - val\_loss: 0.4823 - val\_accuracy: 0.8492 - 114ms/epoch - 3ms/step  
## Epoch 196/200  
## 36/36 - 0s - loss: 0.1477 - accuracy: 0.9385 - val\_loss: 0.5669 - val\_accuracy: 0.8425 - 111ms/epoch - 3ms/step  
## Epoch 197/200  
## 36/36 - 0s - loss: 0.1471 - accuracy: 0.9366 - val\_loss: 0.4744 - val\_accuracy: 0.8570 - 118ms/epoch - 3ms/step  
## Epoch 198/200  
## 36/36 - 0s - loss: 0.1466 - accuracy: 0.9379 - val\_loss: 0.4672 - val\_accuracy: 0.8619 - 123ms/epoch - 3ms/step  
## Epoch 199/200  
## 36/36 - 0s - loss: 0.1488 - accuracy: 0.9392 - val\_loss: 0.4856 - val\_accuracy: 0.8545 - 142ms/epoch - 4ms/step  
## Epoch 200/200  
## 36/36 - 0s - loss: 0.1432 - accuracy: 0.9401 - val\_loss: 0.5510 - val\_accuracy: 0.8463 - 145ms/epoch - 4ms/step

plot(history)



for (x in 1:4) {  
 if (x %% 2 == 0) {  
 print(max(unlist(history$metrics[x])))  
 } else {  
 print(min(unlist(history$metrics[x])))  
 }  
}

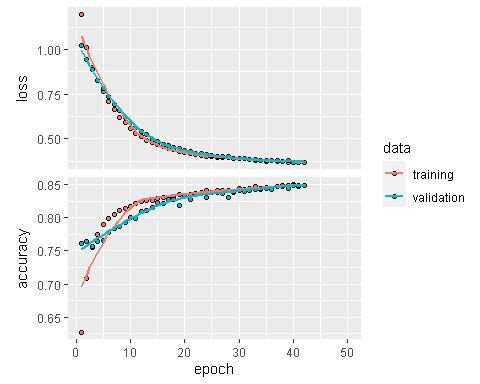
## [1] 0.1432341  
## [1] 0.9400846  
## [1] 0.3286167  
## [1] 0.8643255

### First generalized model

model <- keras\_model\_sequential(list(  
 layer\_dense(units = 100, activation = "relu",  
 kernel\_regularizer = regularizer\_l2(0.002)),  
 layer\_batch\_normalization(),  
 layer\_dropout(rate=0.5),  
 layer\_dense(units = 100, activation = "relu",  
 kernel\_regularizer = regularizer\_l2(0.002)),  
 layer\_batch\_normalization(),  
 layer\_dropout(rate=0.5),  
 layer\_dense(units = 50, activation = "relu",  
 kernel\_regularizer = regularizer\_l2(0.002)),  
 layer\_batch\_normalization(),  
 layer\_dropout(rate=0.5),  
 layer\_dense(units = 1, activation = "sigmoid")  
))  
  
compile(model,  
 optimizer = "rmsprop",  
 loss = "binary\_crossentropy",  
 metrics = "accuracy")  
  
history <- fit(model, training\_features, training\_labels,  
 epochs = 50, batch\_size = 512, validation\_split = 0.33,  
 callbacks = list(callback\_early\_stopping(patience = 2)))

## Epoch 1/50  
## 36/36 - 1s - loss: 1.1954 - accuracy: 0.6273 - val\_loss: 1.0236 - val\_accuracy: 0.7609 - 1s/epoch - 38ms/step  
## Epoch 2/50  
## 36/36 - 0s - loss: 1.0113 - accuracy: 0.7076 - val\_loss: 0.9455 - val\_accuracy: 0.7644 - 205ms/epoch - 6ms/step  
## Epoch 3/50  
## 36/36 - 0s - loss: 0.9004 - accuracy: 0.7553 - val\_loss: 0.8859 - val\_accuracy: 0.7572 - 187ms/epoch - 5ms/step  
## Epoch 4/50  
## 36/36 - 0s - loss: 0.8266 - accuracy: 0.7756 - val\_loss: 0.8252 - val\_accuracy: 0.7645 - 194ms/epoch - 5ms/step  
## Epoch 5/50  
## 36/36 - 0s - loss: 0.7626 - accuracy: 0.7907 - val\_loss: 0.7813 - val\_accuracy: 0.7661 - 196ms/epoch - 5ms/step  
## Epoch 6/50  
## 36/36 - 0s - loss: 0.7118 - accuracy: 0.7988 - val\_loss: 0.7351 - val\_accuracy: 0.7778 - 195ms/epoch - 5ms/step  
## Epoch 7/50  
## 36/36 - 0s - loss: 0.6624 - accuracy: 0.8051 - val\_loss: 0.6929 - val\_accuracy: 0.7837 - 193ms/epoch - 5ms/step  
## Epoch 8/50  
## 36/36 - 0s - loss: 0.6219 - accuracy: 0.8112 - val\_loss: 0.6576 - val\_accuracy: 0.7864 - 188ms/epoch - 5ms/step  
## Epoch 9/50  
## 36/36 - 0s - loss: 0.5918 - accuracy: 0.8135 - val\_loss: 0.6182 - val\_accuracy: 0.7930 - 186ms/epoch - 5ms/step  
## Epoch 10/50  
## 36/36 - 0s - loss: 0.5596 - accuracy: 0.8168 - val\_loss: 0.5907 - val\_accuracy: 0.8003 - 193ms/epoch - 5ms/step  
## Epoch 11/50  
## 36/36 - 0s - loss: 0.5320 - accuracy: 0.8212 - val\_loss: 0.5652 - val\_accuracy: 0.7984 - 204ms/epoch - 6ms/step  
## Epoch 12/50  
## 36/36 - 0s - loss: 0.5133 - accuracy: 0.8250 - val\_loss: 0.5408 - val\_accuracy: 0.8101 - 195ms/epoch - 5ms/step  
## Epoch 13/50  
## 36/36 - 0s - loss: 0.4931 - accuracy: 0.8260 - val\_loss: 0.5218 - val\_accuracy: 0.8114 - 200ms/epoch - 6ms/step  
## Epoch 14/50  
## 36/36 - 0s - loss: 0.4794 - accuracy: 0.8265 - val\_loss: 0.4978 - val\_accuracy: 0.8163 - 190ms/epoch - 5ms/step  
## Epoch 15/50  
## 36/36 - 0s - loss: 0.4683 - accuracy: 0.8285 - val\_loss: 0.4825 - val\_accuracy: 0.8232 - 190ms/epoch - 5ms/step  
## Epoch 16/50  
## 36/36 - 0s - loss: 0.4576 - accuracy: 0.8312 - val\_loss: 0.4697 - val\_accuracy: 0.8214 - 190ms/epoch - 5ms/step  
## Epoch 17/50  
## 36/36 - 0s - loss: 0.4478 - accuracy: 0.8315 - val\_loss: 0.4612 - val\_accuracy: 0.8276 - 187ms/epoch - 5ms/step  
## Epoch 18/50  
## 36/36 - 0s - loss: 0.4401 - accuracy: 0.8330 - val\_loss: 0.4522 - val\_accuracy: 0.8275 - 190ms/epoch - 5ms/step  
## Epoch 19/50  
## 36/36 - 0s - loss: 0.4300 - accuracy: 0.8352 - val\_loss: 0.4459 - val\_accuracy: 0.8179 - 191ms/epoch - 5ms/step  
## Epoch 20/50  
## 36/36 - 0s - loss: 0.4237 - accuracy: 0.8333 - val\_loss: 0.4366 - val\_accuracy: 0.8344 - 186ms/epoch - 5ms/step  
## Epoch 21/50  
## 36/36 - 0s - loss: 0.4175 - accuracy: 0.8360 - val\_loss: 0.4300 - val\_accuracy: 0.8281 - 191ms/epoch - 5ms/step  
## Epoch 22/50  
## 36/36 - 0s - loss: 0.4122 - accuracy: 0.8350 - val\_loss: 0.4178 - val\_accuracy: 0.8366 - 188ms/epoch - 5ms/step  
## Epoch 23/50  
## 36/36 - 0s - loss: 0.4090 - accuracy: 0.8384 - val\_loss: 0.4162 - val\_accuracy: 0.8382 - 196ms/epoch - 5ms/step  
## Epoch 24/50  
## 36/36 - 0s - loss: 0.4029 - accuracy: 0.8412 - val\_loss: 0.4110 - val\_accuracy: 0.8315 - 188ms/epoch - 5ms/step  
## Epoch 25/50  
## 36/36 - 0s - loss: 0.4024 - accuracy: 0.8385 - val\_loss: 0.4076 - val\_accuracy: 0.8388 - 189ms/epoch - 5ms/step  
## Epoch 26/50  
## 36/36 - 0s - loss: 0.3953 - accuracy: 0.8416 - val\_loss: 0.4011 - val\_accuracy: 0.8384 - 190ms/epoch - 5ms/step  
## Epoch 27/50  
## 36/36 - 0s - loss: 0.3959 - accuracy: 0.8418 - val\_loss: 0.4011 - val\_accuracy: 0.8365 - 187ms/epoch - 5ms/step  
## Epoch 28/50  
## 36/36 - 0s - loss: 0.3931 - accuracy: 0.8415 - val\_loss: 0.3987 - val\_accuracy: 0.8315 - 192ms/epoch - 5ms/step  
## Epoch 29/50  
## 36/36 - 0s - loss: 0.3919 - accuracy: 0.8389 - val\_loss: 0.3912 - val\_accuracy: 0.8406 - 192ms/epoch - 5ms/step  
## Epoch 30/50  
## 36/36 - 0s - loss: 0.3890 - accuracy: 0.8418 - val\_loss: 0.3885 - val\_accuracy: 0.8446 - 185ms/epoch - 5ms/step  
## Epoch 31/50  
## 36/36 - 0s - loss: 0.3882 - accuracy: 0.8436 - val\_loss: 0.3886 - val\_accuracy: 0.8397 - 187ms/epoch - 5ms/step  
## Epoch 32/50  
## 36/36 - 0s - loss: 0.3858 - accuracy: 0.8440 - val\_loss: 0.3854 - val\_accuracy: 0.8416 - 203ms/epoch - 6ms/step  
## Epoch 33/50  
## 36/36 - 0s - loss: 0.3788 - accuracy: 0.8468 - val\_loss: 0.3829 - val\_accuracy: 0.8425 - 192ms/epoch - 5ms/step  
## Epoch 34/50  
## 36/36 - 0s - loss: 0.3800 - accuracy: 0.8432 - val\_loss: 0.3809 - val\_accuracy: 0.8450 - 198ms/epoch - 5ms/step  
## Epoch 35/50  
## 36/36 - 0s - loss: 0.3756 - accuracy: 0.8455 - val\_loss: 0.3793 - val\_accuracy: 0.8443 - 202ms/epoch - 6ms/step  
## Epoch 36/50  
## 36/36 - 0s - loss: 0.3794 - accuracy: 0.8446 - val\_loss: 0.3777 - val\_accuracy: 0.8431 - 190ms/epoch - 5ms/step  
## Epoch 37/50  
## 36/36 - 0s - loss: 0.3738 - accuracy: 0.8479 - val\_loss: 0.3770 - val\_accuracy: 0.8468 - 207ms/epoch - 6ms/step  
## Epoch 38/50  
## 36/36 - 0s - loss: 0.3723 - accuracy: 0.8479 - val\_loss: 0.3711 - val\_accuracy: 0.8494 - 238ms/epoch - 7ms/step  
## Epoch 39/50  
## 36/36 - 0s - loss: 0.3703 - accuracy: 0.8488 - val\_loss: 0.3785 - val\_accuracy: 0.8449 - 217ms/epoch - 6ms/step  
## Epoch 40/50  
## 36/36 - 0s - loss: 0.3704 - accuracy: 0.8506 - val\_loss: 0.3686 - val\_accuracy: 0.8491 - 190ms/epoch - 5ms/step  
## Epoch 41/50  
## 36/36 - 0s - loss: 0.3702 - accuracy: 0.8471 - val\_loss: 0.3701 - val\_accuracy: 0.8496 - 193ms/epoch - 5ms/step  
## Epoch 42/50  
## 36/36 - 0s - loss: 0.3685 - accuracy: 0.8497 - val\_loss: 0.3693 - val\_accuracy: 0.8496 - 189ms/epoch - 5ms/step

plot(history)



for (x in 1:4) {  
 if (x %% 2 == 0) {  
 print(max(unlist(history$metrics[x])))  
 } else {  
 print(min(unlist(history$metrics[x])))  
 }  
}

## [1] 0.3684961  
## [1] 0.8505684  
## [1] 0.3686098  
## [1] 0.8496098

### Evaluate generalized model

predictions <- predict(model, test\_features)

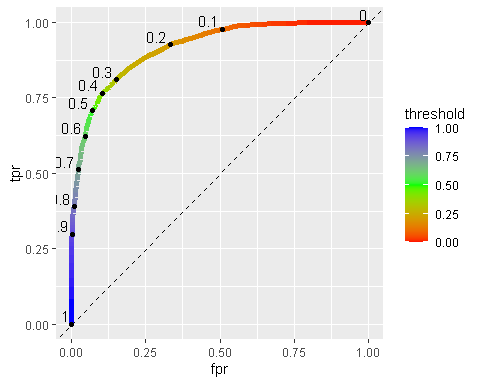
## 284/284 - 0s - 305ms/epoch - 1ms/step

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$booking\_status==0)/sum(test\_set$booking\_status==0)  
 roc\_data[roc\_data$threshold==i, "fpr"] <- fpr  
   
 tpr <- sum(over\_threshold$booking\_status==1)/sum(test\_set$booking\_status==1)  
 roc\_data[roc\_data$threshold==i, "tpr"] <- tpr  
   
}  
  
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

auc <- auc(x = roc\_data$fpr, y = roc\_data$tpr, type = "spline")

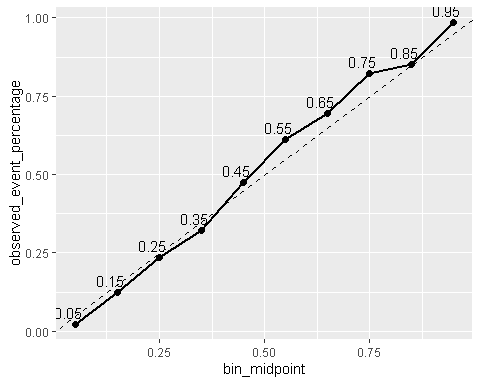
## Warning in regularize.values(x, y, ties, missing(ties)): collapsing to unique  
## 'x' values

auc

## [1] 0.9161274

#### Calibration Curve

calibration\_data <- data.frame(bin\_midpoint=seq(0.05, 0.95, 0.1),  
 observed\_event\_percentage=0)  
for (i in seq(0.05,0.95,0.1)) {  
   
 in\_interval <- test\_set[test\_set$p\_prob >= (i-0.05) & test\_set$p\_prob <= (i+0.05), ]  
 oep <- nrow(in\_interval[in\_interval$booking\_status==1, ])/nrow(in\_interval)  
 calibration\_data[calibration\_data$bin\_midpoint==i, "observed\_event\_percentage"] <- oep  
   
}  
  
ggplot(data = calibration\_data, aes(x = bin\_midpoint, y = observed\_event\_percentage)) +  
 geom\_line(size = 1) +  
 geom\_abline(intercept = 0, slope = 1, lty = 2) +  
 geom\_point(size = 2) +  
 geom\_text(aes(label = bin\_midpoint), hjust = 0.75, vjust = -0.5)

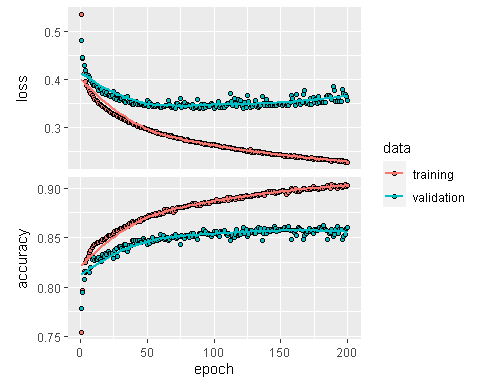


### Model that Fails to Overfit

model\_small <- keras\_model\_sequential(list(  
 layer\_dense(units = 75, activation = "relu"),  
 layer\_dense(units = 37, activation = "relu"),  
 layer\_dense(units = 1, activation = "sigmoid")  
))  
  
compile(model\_small,  
 optimizer = "rmsprop",  
 loss = "binary\_crossentropy",  
 metrics = "accuracy")  
  
history\_small <- fit(model\_small, training\_features, training\_labels,  
 epochs = 200, batch\_size = 512, validation\_split = 0.33)

## Epoch 1/200  
## 36/36 - 1s - loss: 0.5346 - accuracy: 0.7545 - val\_loss: 0.4812 - val\_accuracy: 0.7787 - 597ms/epoch - 17ms/step  
## Epoch 2/200  
## 36/36 - 0s - loss: 0.4429 - accuracy: 0.7961 - val\_loss: 0.4458 - val\_accuracy: 0.7944 - 105ms/epoch - 3ms/step  
## Epoch 3/200  
## 36/36 - 0s - loss: 0.4118 - accuracy: 0.8159 - val\_loss: 0.4291 - val\_accuracy: 0.8071 - 107ms/epoch - 3ms/step  
## Epoch 4/200  
## 36/36 - 0s - loss: 0.3965 - accuracy: 0.8245 - val\_loss: 0.4178 - val\_accuracy: 0.8145 - 102ms/epoch - 3ms/step  
## Epoch 5/200  
## 36/36 - 0s - loss: 0.3872 - accuracy: 0.8279 - val\_loss: 0.4091 - val\_accuracy: 0.8157 - 91ms/epoch - 3ms/step  
## Epoch 6/200  
## 36/36 - 0s - loss: 0.3803 - accuracy: 0.8308 - val\_loss: 0.4072 - val\_accuracy: 0.8161 - 92ms/epoch - 3ms/step  
## Epoch 7/200  
## 36/36 - 0s - loss: 0.3750 - accuracy: 0.8350 - val\_loss: 0.4045 - val\_accuracy: 0.8149 - 90ms/epoch - 2ms/step  
## Epoch 8/200  
## 36/36 - 0s - loss: 0.3706 - accuracy: 0.8367 - val\_loss: 0.3970 - val\_accuracy: 0.8202 - 99ms/epoch - 3ms/step  
## Epoch 9/200  
## 36/36 - 0s - loss: 0.3659 - accuracy: 0.8401 - val\_loss: 0.3910 - val\_accuracy: 0.8279 - 167ms/epoch - 5ms/step  
## Epoch 10/200  
## 36/36 - 0s - loss: 0.3622 - accuracy: 0.8423 - val\_loss: 0.3878 - val\_accuracy: 0.8298 - 120ms/epoch - 3ms/step  
## Epoch 11/200  
## 36/36 - 0s - loss: 0.3590 - accuracy: 0.8432 - val\_loss: 0.3870 - val\_accuracy: 0.8272 - 94ms/epoch - 3ms/step  
## Epoch 12/200  
## 36/36 - 0s - loss: 0.3557 - accuracy: 0.8436 - val\_loss: 0.3842 - val\_accuracy: 0.8275 - 95ms/epoch - 3ms/step  
## Epoch 13/200  
## 36/36 - 0s - loss: 0.3532 - accuracy: 0.8445 - val\_loss: 0.3839 - val\_accuracy: 0.8274 - 92ms/epoch - 3ms/step  
## Epoch 14/200  
## 36/36 - 0s - loss: 0.3497 - accuracy: 0.8447 - val\_loss: 0.3777 - val\_accuracy: 0.8298 - 105ms/epoch - 3ms/step  
## Epoch 15/200  
## 36/36 - 0s - loss: 0.3477 - accuracy: 0.8461 - val\_loss: 0.3805 - val\_accuracy: 0.8281 - 90ms/epoch - 3ms/step  
## Epoch 16/200  
## 36/36 - 0s - loss: 0.3456 - accuracy: 0.8464 - val\_loss: 0.3791 - val\_accuracy: 0.8260 - 90ms/epoch - 3ms/step  
## Epoch 17/200  
## 36/36 - 0s - loss: 0.3425 - accuracy: 0.8483 - val\_loss: 0.3711 - val\_accuracy: 0.8333 - 87ms/epoch - 2ms/step  
## Epoch 18/200  
## 36/36 - 0s - loss: 0.3413 - accuracy: 0.8484 - val\_loss: 0.3738 - val\_accuracy: 0.8304 - 94ms/epoch - 3ms/step  
## Epoch 19/200  
## 36/36 - 0s - loss: 0.3395 - accuracy: 0.8493 - val\_loss: 0.3694 - val\_accuracy: 0.8334 - 111ms/epoch - 3ms/step  
## Epoch 20/200  
## 36/36 - 0s - loss: 0.3369 - accuracy: 0.8502 - val\_loss: 0.3719 - val\_accuracy: 0.8328 - 95ms/epoch - 3ms/step  
## Epoch 21/200  
## 36/36 - 0s - loss: 0.3349 - accuracy: 0.8519 - val\_loss: 0.3655 - val\_accuracy: 0.8359 - 96ms/epoch - 3ms/step  
## Epoch 22/200  
## 36/36 - 0s - loss: 0.3335 - accuracy: 0.8516 - val\_loss: 0.3666 - val\_accuracy: 0.8339 - 92ms/epoch - 3ms/step  
## Epoch 23/200  
## 36/36 - 0s - loss: 0.3314 - accuracy: 0.8524 - val\_loss: 0.3646 - val\_accuracy: 0.8339 - 95ms/epoch - 3ms/step  
## Epoch 24/200  
## 36/36 - 0s - loss: 0.3292 - accuracy: 0.8525 - val\_loss: 0.3667 - val\_accuracy: 0.8337 - 96ms/epoch - 3ms/step  
## Epoch 25/200  
## 36/36 - 0s - loss: 0.3277 - accuracy: 0.8536 - val\_loss: 0.3744 - val\_accuracy: 0.8283 - 98ms/epoch - 3ms/step  
## Epoch 26/200  
## 36/36 - 0s - loss: 0.3261 - accuracy: 0.8562 - val\_loss: 0.3599 - val\_accuracy: 0.8370 - 91ms/epoch - 3ms/step  
## Epoch 27/200  
## 36/36 - 0s - loss: 0.3243 - accuracy: 0.8572 - val\_loss: 0.3580 - val\_accuracy: 0.8394 - 95ms/epoch - 3ms/step  
## Epoch 28/200  
## 36/36 - 0s - loss: 0.3232 - accuracy: 0.8571 - val\_loss: 0.3689 - val\_accuracy: 0.8304 - 94ms/epoch - 3ms/step  
## Epoch 29/200  
## 36/36 - 0s - loss: 0.3202 - accuracy: 0.8581 - val\_loss: 0.3569 - val\_accuracy: 0.8412 - 92ms/epoch - 3ms/step  
## Epoch 30/200  
## 36/36 - 0s - loss: 0.3197 - accuracy: 0.8588 - val\_loss: 0.3611 - val\_accuracy: 0.8407 - 93ms/epoch - 3ms/step  
## Epoch 31/200  
## 36/36 - 0s - loss: 0.3184 - accuracy: 0.8593 - val\_loss: 0.3593 - val\_accuracy: 0.8391 - 96ms/epoch - 3ms/step  
## Epoch 32/200  
## 36/36 - 0s - loss: 0.3158 - accuracy: 0.8603 - val\_loss: 0.3624 - val\_accuracy: 0.8384 - 97ms/epoch - 3ms/step  
## Epoch 33/200  
## 36/36 - 0s - loss: 0.3161 - accuracy: 0.8608 - val\_loss: 0.3540 - val\_accuracy: 0.8430 - 96ms/epoch - 3ms/step  
## Epoch 34/200  
## 36/36 - 0s - loss: 0.3136 - accuracy: 0.8612 - val\_loss: 0.3527 - val\_accuracy: 0.8430 - 92ms/epoch - 3ms/step  
## Epoch 35/200  
## 36/36 - 0s - loss: 0.3123 - accuracy: 0.8624 - val\_loss: 0.3649 - val\_accuracy: 0.8358 - 92ms/epoch - 3ms/step  
## Epoch 36/200  
## 36/36 - 0s - loss: 0.3112 - accuracy: 0.8616 - val\_loss: 0.3548 - val\_accuracy: 0.8389 - 100ms/epoch - 3ms/step  
## Epoch 37/200  
## 36/36 - 0s - loss: 0.3094 - accuracy: 0.8637 - val\_loss: 0.3524 - val\_accuracy: 0.8423 - 96ms/epoch - 3ms/step  
## Epoch 38/200  
## 36/36 - 0s - loss: 0.3094 - accuracy: 0.8645 - val\_loss: 0.3528 - val\_accuracy: 0.8416 - 90ms/epoch - 2ms/step  
## Epoch 39/200  
## 36/36 - 0s - loss: 0.3072 - accuracy: 0.8661 - val\_loss: 0.3484 - val\_accuracy: 0.8470 - 96ms/epoch - 3ms/step  
## Epoch 40/200  
## 36/36 - 0s - loss: 0.3067 - accuracy: 0.8654 - val\_loss: 0.3486 - val\_accuracy: 0.8465 - 94ms/epoch - 3ms/step  
## Epoch 41/200  
## 36/36 - 0s - loss: 0.3061 - accuracy: 0.8651 - val\_loss: 0.3487 - val\_accuracy: 0.8465 - 100ms/epoch - 3ms/step  
## Epoch 42/200  
## 36/36 - 0s - loss: 0.3042 - accuracy: 0.8663 - val\_loss: 0.3535 - val\_accuracy: 0.8436 - 91ms/epoch - 3ms/step  
## Epoch 43/200  
## 36/36 - 0s - loss: 0.3027 - accuracy: 0.8663 - val\_loss: 0.3482 - val\_accuracy: 0.8468 - 92ms/epoch - 3ms/step  
## Epoch 44/200  
## 36/36 - 0s - loss: 0.3020 - accuracy: 0.8684 - val\_loss: 0.3473 - val\_accuracy: 0.8468 - 96ms/epoch - 3ms/step  
## Epoch 45/200  
## 36/36 - 0s - loss: 0.3001 - accuracy: 0.8662 - val\_loss: 0.3495 - val\_accuracy: 0.8452 - 93ms/epoch - 3ms/step  
## Epoch 46/200  
## 36/36 - 0s - loss: 0.2998 - accuracy: 0.8695 - val\_loss: 0.3491 - val\_accuracy: 0.8454 - 100ms/epoch - 3ms/step  
## Epoch 47/200  
## 36/36 - 0s - loss: 0.2992 - accuracy: 0.8685 - val\_loss: 0.3464 - val\_accuracy: 0.8485 - 98ms/epoch - 3ms/step  
## Epoch 48/200  
## 36/36 - 0s - loss: 0.2982 - accuracy: 0.8688 - val\_loss: 0.3447 - val\_accuracy: 0.8466 - 96ms/epoch - 3ms/step  
## Epoch 49/200  
## 36/36 - 0s - loss: 0.2968 - accuracy: 0.8702 - val\_loss: 0.3469 - val\_accuracy: 0.8455 - 95ms/epoch - 3ms/step  
## Epoch 50/200  
## 36/36 - 0s - loss: 0.2947 - accuracy: 0.8705 - val\_loss: 0.3564 - val\_accuracy: 0.8427 - 96ms/epoch - 3ms/step  
## Epoch 51/200  
## 36/36 - 0s - loss: 0.2950 - accuracy: 0.8718 - val\_loss: 0.3443 - val\_accuracy: 0.8466 - 95ms/epoch - 3ms/step  
## Epoch 52/200  
## 36/36 - 0s - loss: 0.2927 - accuracy: 0.8716 - val\_loss: 0.3514 - val\_accuracy: 0.8441 - 96ms/epoch - 3ms/step  
## Epoch 53/200  
## 36/36 - 0s - loss: 0.2937 - accuracy: 0.8722 - val\_loss: 0.3466 - val\_accuracy: 0.8465 - 94ms/epoch - 3ms/step  
## Epoch 54/200  
## 36/36 - 0s - loss: 0.2917 - accuracy: 0.8730 - val\_loss: 0.3445 - val\_accuracy: 0.8499 - 95ms/epoch - 3ms/step  
## Epoch 55/200  
## 36/36 - 0s - loss: 0.2916 - accuracy: 0.8719 - val\_loss: 0.3460 - val\_accuracy: 0.8462 - 99ms/epoch - 3ms/step  
## Epoch 56/200  
## 36/36 - 0s - loss: 0.2893 - accuracy: 0.8738 - val\_loss: 0.3441 - val\_accuracy: 0.8468 - 95ms/epoch - 3ms/step  
## Epoch 57/200  
## 36/36 - 0s - loss: 0.2905 - accuracy: 0.8725 - val\_loss: 0.3446 - val\_accuracy: 0.8483 - 98ms/epoch - 3ms/step  
## Epoch 58/200  
## 36/36 - 0s - loss: 0.2880 - accuracy: 0.8736 - val\_loss: 0.3481 - val\_accuracy: 0.8466 - 93ms/epoch - 3ms/step  
## Epoch 59/200  
## 36/36 - 0s - loss: 0.2867 - accuracy: 0.8752 - val\_loss: 0.3431 - val\_accuracy: 0.8479 - 94ms/epoch - 3ms/step  
## Epoch 60/200  
## 36/36 - 0s - loss: 0.2880 - accuracy: 0.8731 - val\_loss: 0.3415 - val\_accuracy: 0.8528 - 93ms/epoch - 3ms/step  
## Epoch 61/200  
## 36/36 - 0s - loss: 0.2859 - accuracy: 0.8743 - val\_loss: 0.3408 - val\_accuracy: 0.8494 - 94ms/epoch - 3ms/step  
## Epoch 62/200  
## 36/36 - 0s - loss: 0.2857 - accuracy: 0.8745 - val\_loss: 0.3426 - val\_accuracy: 0.8520 - 99ms/epoch - 3ms/step  
## Epoch 63/200  
## 36/36 - 0s - loss: 0.2860 - accuracy: 0.8782 - val\_loss: 0.3436 - val\_accuracy: 0.8484 - 95ms/epoch - 3ms/step  
## Epoch 64/200  
## 36/36 - 0s - loss: 0.2846 - accuracy: 0.8764 - val\_loss: 0.3461 - val\_accuracy: 0.8491 - 98ms/epoch - 3ms/step  
## Epoch 65/200  
## 36/36 - 0s - loss: 0.2835 - accuracy: 0.8780 - val\_loss: 0.3457 - val\_accuracy: 0.8504 - 92ms/epoch - 3ms/step  
## Epoch 66/200  
## 36/36 - 0s - loss: 0.2833 - accuracy: 0.8758 - val\_loss: 0.3558 - val\_accuracy: 0.8470 - 98ms/epoch - 3ms/step  
## Epoch 67/200  
## 36/36 - 0s - loss: 0.2817 - accuracy: 0.8773 - val\_loss: 0.3483 - val\_accuracy: 0.8484 - 106ms/epoch - 3ms/step  
## Epoch 68/200  
## 36/36 - 0s - loss: 0.2822 - accuracy: 0.8763 - val\_loss: 0.3408 - val\_accuracy: 0.8496 - 93ms/epoch - 3ms/step  
## Epoch 69/200  
## 36/36 - 0s - loss: 0.2801 - accuracy: 0.8793 - val\_loss: 0.3434 - val\_accuracy: 0.8513 - 93ms/epoch - 3ms/step  
## Epoch 70/200  
## 36/36 - 0s - loss: 0.2797 - accuracy: 0.8757 - val\_loss: 0.3429 - val\_accuracy: 0.8504 - 94ms/epoch - 3ms/step  
## Epoch 71/200  
## 36/36 - 0s - loss: 0.2796 - accuracy: 0.8773 - val\_loss: 0.3391 - val\_accuracy: 0.8533 - 93ms/epoch - 3ms/step  
## Epoch 72/200  
## 36/36 - 0s - loss: 0.2788 - accuracy: 0.8795 - val\_loss: 0.3492 - val\_accuracy: 0.8489 - 93ms/epoch - 3ms/step  
## Epoch 73/200  
## 36/36 - 0s - loss: 0.2775 - accuracy: 0.8798 - val\_loss: 0.3489 - val\_accuracy: 0.8499 - 99ms/epoch - 3ms/step  
## Epoch 74/200  
## 36/36 - 0s - loss: 0.2766 - accuracy: 0.8806 - val\_loss: 0.3459 - val\_accuracy: 0.8512 - 106ms/epoch - 3ms/step  
## Epoch 75/200  
## 36/36 - 0s - loss: 0.2763 - accuracy: 0.8805 - val\_loss: 0.3605 - val\_accuracy: 0.8426 - 125ms/epoch - 3ms/step  
## Epoch 76/200  
## 36/36 - 0s - loss: 0.2765 - accuracy: 0.8799 - val\_loss: 0.3424 - val\_accuracy: 0.8518 - 109ms/epoch - 3ms/step  
## Epoch 77/200  
## 36/36 - 0s - loss: 0.2754 - accuracy: 0.8799 - val\_loss: 0.3504 - val\_accuracy: 0.8495 - 107ms/epoch - 3ms/step  
## Epoch 78/200  
## 36/36 - 0s - loss: 0.2736 - accuracy: 0.8814 - val\_loss: 0.3510 - val\_accuracy: 0.8486 - 93ms/epoch - 3ms/step  
## Epoch 79/200  
## 36/36 - 0s - loss: 0.2733 - accuracy: 0.8818 - val\_loss: 0.3438 - val\_accuracy: 0.8524 - 95ms/epoch - 3ms/step  
## Epoch 80/200  
## 36/36 - 0s - loss: 0.2743 - accuracy: 0.8813 - val\_loss: 0.3472 - val\_accuracy: 0.8521 - 94ms/epoch - 3ms/step  
## Epoch 81/200  
## 36/36 - 0s - loss: 0.2733 - accuracy: 0.8822 - val\_loss: 0.3452 - val\_accuracy: 0.8513 - 90ms/epoch - 3ms/step  
## Epoch 82/200  
## 36/36 - 0s - loss: 0.2714 - accuracy: 0.8820 - val\_loss: 0.3452 - val\_accuracy: 0.8504 - 95ms/epoch - 3ms/step  
## Epoch 83/200  
## 36/36 - 0s - loss: 0.2698 - accuracy: 0.8832 - val\_loss: 0.3423 - val\_accuracy: 0.8543 - 101ms/epoch - 3ms/step  
## Epoch 84/200  
## 36/36 - 0s - loss: 0.2712 - accuracy: 0.8839 - val\_loss: 0.3466 - val\_accuracy: 0.8513 - 92ms/epoch - 3ms/step  
## Epoch 85/200  
## 36/36 - 0s - loss: 0.2709 - accuracy: 0.8830 - val\_loss: 0.3403 - val\_accuracy: 0.8543 - 97ms/epoch - 3ms/step  
## Epoch 86/200  
## 36/36 - 0s - loss: 0.2688 - accuracy: 0.8845 - val\_loss: 0.3421 - val\_accuracy: 0.8514 - 94ms/epoch - 3ms/step  
## Epoch 87/200  
## 36/36 - 0s - loss: 0.2682 - accuracy: 0.8828 - val\_loss: 0.3439 - val\_accuracy: 0.8515 - 92ms/epoch - 3ms/step  
## Epoch 88/200  
## 36/36 - 0s - loss: 0.2688 - accuracy: 0.8826 - val\_loss: 0.3576 - val\_accuracy: 0.8482 - 96ms/epoch - 3ms/step  
## Epoch 89/200  
## 36/36 - 0s - loss: 0.2678 - accuracy: 0.8834 - val\_loss: 0.3389 - val\_accuracy: 0.8545 - 94ms/epoch - 3ms/step  
## Epoch 90/200  
## 36/36 - 0s - loss: 0.2681 - accuracy: 0.8826 - val\_loss: 0.3423 - val\_accuracy: 0.8541 - 92ms/epoch - 3ms/step  
## Epoch 91/200  
## 36/36 - 0s - loss: 0.2663 - accuracy: 0.8846 - val\_loss: 0.3471 - val\_accuracy: 0.8536 - 103ms/epoch - 3ms/step  
## Epoch 92/200  
## 36/36 - 0s - loss: 0.2674 - accuracy: 0.8842 - val\_loss: 0.3394 - val\_accuracy: 0.8582 - 98ms/epoch - 3ms/step  
## Epoch 93/200  
## 36/36 - 0s - loss: 0.2652 - accuracy: 0.8851 - val\_loss: 0.3479 - val\_accuracy: 0.8540 - 97ms/epoch - 3ms/step  
## Epoch 94/200  
## 36/36 - 0s - loss: 0.2662 - accuracy: 0.8847 - val\_loss: 0.3412 - val\_accuracy: 0.8564 - 94ms/epoch - 3ms/step  
## Epoch 95/200  
## 36/36 - 0s - loss: 0.2634 - accuracy: 0.8848 - val\_loss: 0.3448 - val\_accuracy: 0.8528 - 93ms/epoch - 3ms/step  
## Epoch 96/200  
## 36/36 - 0s - loss: 0.2643 - accuracy: 0.8851 - val\_loss: 0.3428 - val\_accuracy: 0.8536 - 95ms/epoch - 3ms/step  
## Epoch 97/200  
## 36/36 - 0s - loss: 0.2648 - accuracy: 0.8846 - val\_loss: 0.3441 - val\_accuracy: 0.8528 - 92ms/epoch - 3ms/step  
## Epoch 98/200  
## 36/36 - 0s - loss: 0.2622 - accuracy: 0.8867 - val\_loss: 0.3493 - val\_accuracy: 0.8512 - 92ms/epoch - 3ms/step  
## Epoch 99/200  
## 36/36 - 0s - loss: 0.2650 - accuracy: 0.8862 - val\_loss: 0.3469 - val\_accuracy: 0.8531 - 97ms/epoch - 3ms/step  
## Epoch 100/200  
## 36/36 - 0s - loss: 0.2635 - accuracy: 0.8864 - val\_loss: 0.3391 - val\_accuracy: 0.8557 - 93ms/epoch - 3ms/step  
## Epoch 101/200  
## 36/36 - 0s - loss: 0.2620 - accuracy: 0.8866 - val\_loss: 0.3457 - val\_accuracy: 0.8538 - 96ms/epoch - 3ms/step  
## Epoch 102/200  
## 36/36 - 0s - loss: 0.2617 - accuracy: 0.8881 - val\_loss: 0.3438 - val\_accuracy: 0.8557 - 97ms/epoch - 3ms/step  
## Epoch 103/200  
## 36/36 - 0s - loss: 0.2628 - accuracy: 0.8864 - val\_loss: 0.3538 - val\_accuracy: 0.8521 - 96ms/epoch - 3ms/step  
## Epoch 104/200  
## 36/36 - 0s - loss: 0.2601 - accuracy: 0.8866 - val\_loss: 0.3424 - val\_accuracy: 0.8508 - 98ms/epoch - 3ms/step  
## Epoch 105/200  
## 36/36 - 0s - loss: 0.2608 - accuracy: 0.8873 - val\_loss: 0.3447 - val\_accuracy: 0.8560 - 89ms/epoch - 2ms/step  
## Epoch 106/200  
## 36/36 - 0s - loss: 0.2591 - accuracy: 0.8873 - val\_loss: 0.3422 - val\_accuracy: 0.8556 - 95ms/epoch - 3ms/step  
## Epoch 107/200  
## 36/36 - 0s - loss: 0.2586 - accuracy: 0.8884 - val\_loss: 0.3454 - val\_accuracy: 0.8557 - 97ms/epoch - 3ms/step  
## Epoch 108/200  
## 36/36 - 0s - loss: 0.2598 - accuracy: 0.8874 - val\_loss: 0.3411 - val\_accuracy: 0.8574 - 91ms/epoch - 3ms/step  
## Epoch 109/200  
## 36/36 - 0s - loss: 0.2574 - accuracy: 0.8897 - val\_loss: 0.3461 - val\_accuracy: 0.8533 - 90ms/epoch - 3ms/step  
## Epoch 110/200  
## 36/36 - 0s - loss: 0.2578 - accuracy: 0.8902 - val\_loss: 0.3493 - val\_accuracy: 0.8517 - 91ms/epoch - 3ms/step  
## Epoch 111/200  
## 36/36 - 0s - loss: 0.2580 - accuracy: 0.8879 - val\_loss: 0.3492 - val\_accuracy: 0.8543 - 95ms/epoch - 3ms/step  
## Epoch 112/200  
## 36/36 - 0s - loss: 0.2571 - accuracy: 0.8899 - val\_loss: 0.3695 - val\_accuracy: 0.8465 - 93ms/epoch - 3ms/step  
## Epoch 113/200  
## 36/36 - 0s - loss: 0.2558 - accuracy: 0.8884 - val\_loss: 0.3531 - val\_accuracy: 0.8516 - 92ms/epoch - 3ms/step  
## Epoch 114/200  
## 36/36 - 0s - loss: 0.2564 - accuracy: 0.8893 - val\_loss: 0.3457 - val\_accuracy: 0.8524 - 105ms/epoch - 3ms/step  
## Epoch 115/200  
## 36/36 - 0s - loss: 0.2562 - accuracy: 0.8895 - val\_loss: 0.3612 - val\_accuracy: 0.8503 - 103ms/epoch - 3ms/step  
## Epoch 116/200  
## 36/36 - 0s - loss: 0.2549 - accuracy: 0.8885 - val\_loss: 0.3497 - val\_accuracy: 0.8532 - 100ms/epoch - 3ms/step  
## Epoch 117/200  
## 36/36 - 0s - loss: 0.2543 - accuracy: 0.8901 - val\_loss: 0.3426 - val\_accuracy: 0.8585 - 95ms/epoch - 3ms/step  
## Epoch 118/200  
## 36/36 - 0s - loss: 0.2550 - accuracy: 0.8900 - val\_loss: 0.3587 - val\_accuracy: 0.8521 - 93ms/epoch - 3ms/step  
## Epoch 119/200  
## 36/36 - 0s - loss: 0.2541 - accuracy: 0.8892 - val\_loss: 0.3493 - val\_accuracy: 0.8534 - 102ms/epoch - 3ms/step  
## Epoch 120/200  
## 36/36 - 0s - loss: 0.2532 - accuracy: 0.8914 - val\_loss: 0.3570 - val\_accuracy: 0.8531 - 94ms/epoch - 3ms/step  
## Epoch 121/200  
## 36/36 - 0s - loss: 0.2541 - accuracy: 0.8894 - val\_loss: 0.3439 - val\_accuracy: 0.8556 - 95ms/epoch - 3ms/step  
## Epoch 122/200  
## 36/36 - 0s - loss: 0.2520 - accuracy: 0.8913 - val\_loss: 0.3397 - val\_accuracy: 0.8600 - 93ms/epoch - 3ms/step  
## Epoch 123/200  
## 36/36 - 0s - loss: 0.2522 - accuracy: 0.8916 - val\_loss: 0.3523 - val\_accuracy: 0.8545 - 96ms/epoch - 3ms/step  
## Epoch 124/200  
## 36/36 - 0s - loss: 0.2507 - accuracy: 0.8920 - val\_loss: 0.3423 - val\_accuracy: 0.8583 - 93ms/epoch - 3ms/step  
## Epoch 125/200  
## 36/36 - 0s - loss: 0.2504 - accuracy: 0.8918 - val\_loss: 0.3660 - val\_accuracy: 0.8515 - 96ms/epoch - 3ms/step  
## Epoch 126/200  
## 36/36 - 0s - loss: 0.2516 - accuracy: 0.8912 - val\_loss: 0.3537 - val\_accuracy: 0.8532 - 90ms/epoch - 3ms/step  
## Epoch 127/200  
## 36/36 - 0s - loss: 0.2531 - accuracy: 0.8914 - val\_loss: 0.3416 - val\_accuracy: 0.8567 - 92ms/epoch - 3ms/step  
## Epoch 128/200  
## 36/36 - 0s - loss: 0.2487 - accuracy: 0.8918 - val\_loss: 0.3464 - val\_accuracy: 0.8573 - 95ms/epoch - 3ms/step  
## Epoch 129/200  
## 36/36 - 0s - loss: 0.2497 - accuracy: 0.8922 - val\_loss: 0.3443 - val\_accuracy: 0.8553 - 96ms/epoch - 3ms/step  
## Epoch 130/200  
## 36/36 - 0s - loss: 0.2484 - accuracy: 0.8923 - val\_loss: 0.3468 - val\_accuracy: 0.8536 - 91ms/epoch - 3ms/step  
## Epoch 131/200  
## 36/36 - 0s - loss: 0.2505 - accuracy: 0.8935 - val\_loss: 0.3563 - val\_accuracy: 0.8540 - 90ms/epoch - 3ms/step  
## Epoch 132/200  
## 36/36 - 0s - loss: 0.2490 - accuracy: 0.8930 - val\_loss: 0.3439 - val\_accuracy: 0.8576 - 94ms/epoch - 3ms/step  
## Epoch 133/200  
## 36/36 - 0s - loss: 0.2475 - accuracy: 0.8927 - val\_loss: 0.3420 - val\_accuracy: 0.8604 - 95ms/epoch - 3ms/step  
## Epoch 134/200  
## 36/36 - 0s - loss: 0.2492 - accuracy: 0.8936 - val\_loss: 0.3435 - val\_accuracy: 0.8565 - 89ms/epoch - 2ms/step  
## Epoch 135/200  
## 36/36 - 0s - loss: 0.2482 - accuracy: 0.8933 - val\_loss: 0.3460 - val\_accuracy: 0.8586 - 90ms/epoch - 2ms/step  
## Epoch 136/200  
## 36/36 - 0s - loss: 0.2463 - accuracy: 0.8941 - val\_loss: 0.3641 - val\_accuracy: 0.8474 - 96ms/epoch - 3ms/step  
## Epoch 137/200  
## 36/36 - 0s - loss: 0.2471 - accuracy: 0.8941 - val\_loss: 0.3497 - val\_accuracy: 0.8545 - 97ms/epoch - 3ms/step  
## Epoch 138/200  
## 36/36 - 0s - loss: 0.2469 - accuracy: 0.8925 - val\_loss: 0.3456 - val\_accuracy: 0.8595 - 118ms/epoch - 3ms/step  
## Epoch 139/200  
## 36/36 - 0s - loss: 0.2450 - accuracy: 0.8955 - val\_loss: 0.3581 - val\_accuracy: 0.8537 - 110ms/epoch - 3ms/step  
## Epoch 140/200  
## 36/36 - 0s - loss: 0.2459 - accuracy: 0.8955 - val\_loss: 0.3461 - val\_accuracy: 0.8575 - 104ms/epoch - 3ms/step  
## Epoch 141/200  
## 36/36 - 0s - loss: 0.2456 - accuracy: 0.8960 - val\_loss: 0.3496 - val\_accuracy: 0.8584 - 123ms/epoch - 3ms/step  
## Epoch 142/200  
## 36/36 - 0s - loss: 0.2454 - accuracy: 0.8960 - val\_loss: 0.3445 - val\_accuracy: 0.8561 - 109ms/epoch - 3ms/step  
## Epoch 143/200  
## 36/36 - 0s - loss: 0.2443 - accuracy: 0.8955 - val\_loss: 0.3492 - val\_accuracy: 0.8583 - 93ms/epoch - 3ms/step  
## Epoch 144/200  
## 36/36 - 0s - loss: 0.2439 - accuracy: 0.8962 - val\_loss: 0.3451 - val\_accuracy: 0.8573 - 91ms/epoch - 3ms/step  
## Epoch 145/200  
## 36/36 - 0s - loss: 0.2440 - accuracy: 0.8968 - val\_loss: 0.3447 - val\_accuracy: 0.8611 - 96ms/epoch - 3ms/step  
## Epoch 146/200  
## 36/36 - 0s - loss: 0.2440 - accuracy: 0.8949 - val\_loss: 0.3467 - val\_accuracy: 0.8586 - 94ms/epoch - 3ms/step  
## Epoch 147/200  
## 36/36 - 0s - loss: 0.2434 - accuracy: 0.8959 - val\_loss: 0.3447 - val\_accuracy: 0.8560 - 95ms/epoch - 3ms/step  
## Epoch 148/200  
## 36/36 - 0s - loss: 0.2436 - accuracy: 0.8958 - val\_loss: 0.3495 - val\_accuracy: 0.8583 - 97ms/epoch - 3ms/step  
## Epoch 149/200  
## 36/36 - 0s - loss: 0.2432 - accuracy: 0.8962 - val\_loss: 0.3503 - val\_accuracy: 0.8573 - 94ms/epoch - 3ms/step  
## Epoch 150/200  
## 36/36 - 0s - loss: 0.2429 - accuracy: 0.8962 - val\_loss: 0.3473 - val\_accuracy: 0.8580 - 98ms/epoch - 3ms/step  
## Epoch 151/200  
## 36/36 - 0s - loss: 0.2415 - accuracy: 0.8976 - val\_loss: 0.3452 - val\_accuracy: 0.8580 - 95ms/epoch - 3ms/step  
## Epoch 152/200  
## 36/36 - 0s - loss: 0.2407 - accuracy: 0.8976 - val\_loss: 0.3505 - val\_accuracy: 0.8576 - 94ms/epoch - 3ms/step  
## Epoch 153/200  
## 36/36 - 0s - loss: 0.2404 - accuracy: 0.8968 - val\_loss: 0.3475 - val\_accuracy: 0.8599 - 96ms/epoch - 3ms/step  
## Epoch 154/200  
## 36/36 - 0s - loss: 0.2421 - accuracy: 0.8937 - val\_loss: 0.3457 - val\_accuracy: 0.8579 - 94ms/epoch - 3ms/step  
## Epoch 155/200  
## 36/36 - 0s - loss: 0.2401 - accuracy: 0.8967 - val\_loss: 0.3471 - val\_accuracy: 0.8596 - 92ms/epoch - 3ms/step  
## Epoch 156/200  
## 36/36 - 0s - loss: 0.2400 - accuracy: 0.8955 - val\_loss: 0.3455 - val\_accuracy: 0.8609 - 93ms/epoch - 3ms/step  
## Epoch 157/200  
## 36/36 - 0s - loss: 0.2406 - accuracy: 0.8977 - val\_loss: 0.3514 - val\_accuracy: 0.8566 - 99ms/epoch - 3ms/step  
## Epoch 158/200  
## 36/36 - 0s - loss: 0.2399 - accuracy: 0.8996 - val\_loss: 0.3669 - val\_accuracy: 0.8476 - 96ms/epoch - 3ms/step  
## Epoch 159/200  
## 36/36 - 0s - loss: 0.2393 - accuracy: 0.8984 - val\_loss: 0.3601 - val\_accuracy: 0.8571 - 96ms/epoch - 3ms/step  
## Epoch 160/200  
## 36/36 - 0s - loss: 0.2406 - accuracy: 0.8970 - val\_loss: 0.3517 - val\_accuracy: 0.8552 - 98ms/epoch - 3ms/step  
## Epoch 161/200  
## 36/36 - 0s - loss: 0.2384 - accuracy: 0.8992 - val\_loss: 0.3463 - val\_accuracy: 0.8575 - 94ms/epoch - 3ms/step  
## Epoch 162/200  
## 36/36 - 0s - loss: 0.2392 - accuracy: 0.8964 - val\_loss: 0.3679 - val\_accuracy: 0.8525 - 97ms/epoch - 3ms/step  
## Epoch 163/200  
## 36/36 - 0s - loss: 0.2383 - accuracy: 0.8969 - val\_loss: 0.3490 - val\_accuracy: 0.8603 - 94ms/epoch - 3ms/step  
## Epoch 164/200  
## 36/36 - 0s - loss: 0.2388 - accuracy: 0.8968 - val\_loss: 0.3562 - val\_accuracy: 0.8594 - 103ms/epoch - 3ms/step  
## Epoch 165/200  
## 36/36 - 0s - loss: 0.2399 - accuracy: 0.8978 - val\_loss: 0.3466 - val\_accuracy: 0.8581 - 93ms/epoch - 3ms/step  
## Epoch 166/200  
## 36/36 - 0s - loss: 0.2377 - accuracy: 0.8985 - val\_loss: 0.3518 - val\_accuracy: 0.8569 - 91ms/epoch - 3ms/step  
## Epoch 167/200  
## 36/36 - 0s - loss: 0.2368 - accuracy: 0.8998 - val\_loss: 0.3690 - val\_accuracy: 0.8477 - 100ms/epoch - 3ms/step  
## Epoch 168/200  
## 36/36 - 0s - loss: 0.2368 - accuracy: 0.8990 - val\_loss: 0.3768 - val\_accuracy: 0.8521 - 112ms/epoch - 3ms/step  
## Epoch 169/200  
## 36/36 - 0s - loss: 0.2375 - accuracy: 0.8981 - val\_loss: 0.3492 - val\_accuracy: 0.8543 - 107ms/epoch - 3ms/step  
## Epoch 170/200  
## 36/36 - 0s - loss: 0.2361 - accuracy: 0.8986 - val\_loss: 0.3515 - val\_accuracy: 0.8565 - 126ms/epoch - 3ms/step  
## Epoch 171/200  
## 36/36 - 0s - loss: 0.2356 - accuracy: 0.8998 - val\_loss: 0.3476 - val\_accuracy: 0.8589 - 99ms/epoch - 3ms/step  
## Epoch 172/200  
## 36/36 - 0s - loss: 0.2358 - accuracy: 0.8981 - val\_loss: 0.3604 - val\_accuracy: 0.8561 - 106ms/epoch - 3ms/step  
## Epoch 173/200  
## 36/36 - 0s - loss: 0.2364 - accuracy: 0.8993 - val\_loss: 0.3476 - val\_accuracy: 0.8572 - 98ms/epoch - 3ms/step  
## Epoch 174/200  
## 36/36 - 0s - loss: 0.2331 - accuracy: 0.9014 - val\_loss: 0.3614 - val\_accuracy: 0.8537 - 97ms/epoch - 3ms/step  
## Epoch 175/200  
## 36/36 - 0s - loss: 0.2360 - accuracy: 0.8996 - val\_loss: 0.3525 - val\_accuracy: 0.8577 - 90ms/epoch - 2ms/step  
## Epoch 176/200  
## 36/36 - 0s - loss: 0.2346 - accuracy: 0.8992 - val\_loss: 0.3588 - val\_accuracy: 0.8572 - 95ms/epoch - 3ms/step  
## Epoch 177/200  
## 36/36 - 0s - loss: 0.2341 - accuracy: 0.8992 - val\_loss: 0.3501 - val\_accuracy: 0.8569 - 99ms/epoch - 3ms/step  
## Epoch 178/200  
## 36/36 - 0s - loss: 0.2343 - accuracy: 0.8989 - val\_loss: 0.3640 - val\_accuracy: 0.8586 - 100ms/epoch - 3ms/step  
## Epoch 179/200  
## 36/36 - 0s - loss: 0.2328 - accuracy: 0.9000 - val\_loss: 0.3511 - val\_accuracy: 0.8601 - 93ms/epoch - 3ms/step  
## Epoch 180/200  
## 36/36 - 0s - loss: 0.2353 - accuracy: 0.8996 - val\_loss: 0.3554 - val\_accuracy: 0.8589 - 95ms/epoch - 3ms/step  
## Epoch 181/200  
## 36/36 - 0s - loss: 0.2331 - accuracy: 0.9010 - val\_loss: 0.3521 - val\_accuracy: 0.8610 - 94ms/epoch - 3ms/step  
## Epoch 182/200  
## 36/36 - 0s - loss: 0.2330 - accuracy: 0.8999 - val\_loss: 0.3521 - val\_accuracy: 0.8618 - 94ms/epoch - 3ms/step  
## Epoch 183/200  
## 36/36 - 0s - loss: 0.2324 - accuracy: 0.9010 - val\_loss: 0.3526 - val\_accuracy: 0.8583 - 97ms/epoch - 3ms/step  
## Epoch 184/200  
## 36/36 - 0s - loss: 0.2327 - accuracy: 0.9017 - val\_loss: 0.3650 - val\_accuracy: 0.8569 - 94ms/epoch - 3ms/step  
## Epoch 185/200  
## 36/36 - 0s - loss: 0.2322 - accuracy: 0.9020 - val\_loss: 0.3655 - val\_accuracy: 0.8479 - 96ms/epoch - 3ms/step  
## Epoch 186/200  
## 36/36 - 0s - loss: 0.2317 - accuracy: 0.9019 - val\_loss: 0.3662 - val\_accuracy: 0.8541 - 95ms/epoch - 3ms/step  
## Epoch 187/200  
## 36/36 - 0s - loss: 0.2312 - accuracy: 0.8996 - val\_loss: 0.3639 - val\_accuracy: 0.8546 - 94ms/epoch - 3ms/step  
## Epoch 188/200  
## 36/36 - 0s - loss: 0.2296 - accuracy: 0.9033 - val\_loss: 0.3581 - val\_accuracy: 0.8583 - 97ms/epoch - 3ms/step  
## Epoch 189/200  
## 36/36 - 0s - loss: 0.2319 - accuracy: 0.9013 - val\_loss: 0.3853 - val\_accuracy: 0.8515 - 98ms/epoch - 3ms/step  
## Epoch 190/200  
## 36/36 - 0s - loss: 0.2305 - accuracy: 0.9025 - val\_loss: 0.3762 - val\_accuracy: 0.8535 - 92ms/epoch - 3ms/step  
## Epoch 191/200  
## 36/36 - 0s - loss: 0.2318 - accuracy: 0.9012 - val\_loss: 0.3532 - val\_accuracy: 0.8584 - 90ms/epoch - 3ms/step  
## Epoch 192/200  
## 36/36 - 0s - loss: 0.2317 - accuracy: 0.8998 - val\_loss: 0.3531 - val\_accuracy: 0.8612 - 101ms/epoch - 3ms/step  
## Epoch 193/200  
## 36/36 - 0s - loss: 0.2295 - accuracy: 0.9017 - val\_loss: 0.3613 - val\_accuracy: 0.8542 - 94ms/epoch - 3ms/step  
## Epoch 194/200  
## 36/36 - 0s - loss: 0.2313 - accuracy: 0.9024 - val\_loss: 0.3541 - val\_accuracy: 0.8580 - 93ms/epoch - 3ms/step  
## Epoch 195/200  
## 36/36 - 0s - loss: 0.2281 - accuracy: 0.9021 - val\_loss: 0.3532 - val\_accuracy: 0.8580 - 96ms/epoch - 3ms/step  
## Epoch 196/200  
## 36/36 - 0s - loss: 0.2315 - accuracy: 0.9001 - val\_loss: 0.3797 - val\_accuracy: 0.8483 - 94ms/epoch - 3ms/step  
## Epoch 197/200  
## 36/36 - 0s - loss: 0.2291 - accuracy: 0.9020 - val\_loss: 0.3697 - val\_accuracy: 0.8514 - 95ms/epoch - 3ms/step  
## Epoch 198/200  
## 36/36 - 0s - loss: 0.2279 - accuracy: 0.9036 - val\_loss: 0.3684 - val\_accuracy: 0.8527 - 93ms/epoch - 3ms/step  
## Epoch 199/200  
## 36/36 - 0s - loss: 0.2296 - accuracy: 0.9016 - val\_loss: 0.3648 - val\_accuracy: 0.8590 - 93ms/epoch - 3ms/step  
## Epoch 200/200  
## 36/36 - 0s - loss: 0.2272 - accuracy: 0.9027 - val\_loss: 0.3566 - val\_accuracy: 0.8598 - 95ms/epoch - 3ms/step

plot(history\_small)



for (x in 1:4) {  
 if (x %% 2 == 0) {  
 print(max(unlist(history\_small$metrics[x])))  
 } else {  
 print(min(unlist(history\_small$metrics[x])))  
 }  
}

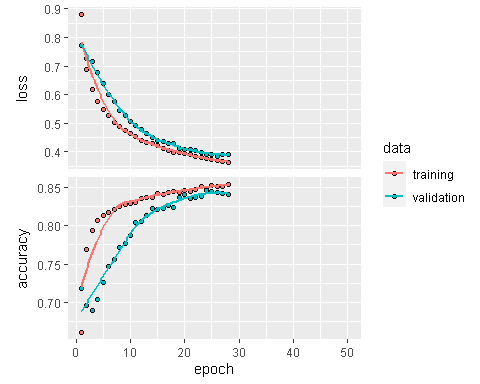
## [1] 0.2272339  
## [1] 0.9036191  
## [1] 0.3389435  
## [1] 0.8617615

### Generalize smaller model

model\_small <- keras\_model\_sequential(list(  
 layer\_dense(units = 75, activation = "relu",  
 kernel\_regularizer = regularizer\_l2(0.002)),  
 layer\_batch\_normalization(),  
 layer\_dropout(rate=0.2),  
 layer\_dense(units = 37, activation = "relu",  
 kernel\_regularizer = regularizer\_l2(0.002)),  
 layer\_batch\_normalization(),  
 layer\_dropout(rate=0.2),  
 layer\_dense(units = 1, activation = "sigmoid")  
))  
  
compile(model\_small,  
 optimizer = "rmsprop",  
 loss = "binary\_crossentropy",  
 metrics = "accuracy")  
  
history\_small <- fit(model\_small, training\_features, training\_labels,  
 epochs = 50, batch\_size = 512, validation\_split = 0.33,  
 callbacks = list(callback\_early\_stopping(patience = 2)))

## Epoch 1/50  
## 36/36 - 1s - loss: 0.8792 - accuracy: 0.6609 - val\_loss: 0.7716 - val\_accuracy: 0.7177 - 1s/epoch - 28ms/step  
## Epoch 2/50  
## 36/36 - 0s - loss: 0.6869 - accuracy: 0.7695 - val\_loss: 0.7276 - val\_accuracy: 0.6960 - 141ms/epoch - 4ms/step  
## Epoch 3/50  
## 36/36 - 0s - loss: 0.6186 - accuracy: 0.7936 - val\_loss: 0.7141 - val\_accuracy: 0.6889 - 137ms/epoch - 4ms/step  
## Epoch 4/50  
## 36/36 - 0s - loss: 0.5772 - accuracy: 0.8066 - val\_loss: 0.6776 - val\_accuracy: 0.7038 - 143ms/epoch - 4ms/step  
## Epoch 5/50  
## 36/36 - 0s - loss: 0.5495 - accuracy: 0.8138 - val\_loss: 0.6404 - val\_accuracy: 0.7259 - 151ms/epoch - 4ms/step  
## Epoch 6/50  
## 36/36 - 0s - loss: 0.5263 - accuracy: 0.8172 - val\_loss: 0.6003 - val\_accuracy: 0.7464 - 142ms/epoch - 4ms/step  
## Epoch 7/50  
## 36/36 - 0s - loss: 0.5032 - accuracy: 0.8218 - val\_loss: 0.5751 - val\_accuracy: 0.7562 - 149ms/epoch - 4ms/step  
## Epoch 8/50  
## 36/36 - 0s - loss: 0.4892 - accuracy: 0.8255 - val\_loss: 0.5456 - val\_accuracy: 0.7717 - 138ms/epoch - 4ms/step  
## Epoch 9/50  
## 36/36 - 0s - loss: 0.4743 - accuracy: 0.8278 - val\_loss: 0.5276 - val\_accuracy: 0.7769 - 145ms/epoch - 4ms/step  
## Epoch 10/50  
## 36/36 - 0s - loss: 0.4638 - accuracy: 0.8294 - val\_loss: 0.5072 - val\_accuracy: 0.7873 - 136ms/epoch - 4ms/step  
## Epoch 11/50  
## 36/36 - 0s - loss: 0.4547 - accuracy: 0.8304 - val\_loss: 0.4916 - val\_accuracy: 0.8040 - 137ms/epoch - 4ms/step  
## Epoch 12/50  
## 36/36 - 0s - loss: 0.4413 - accuracy: 0.8358 - val\_loss: 0.4797 - val\_accuracy: 0.8054 - 145ms/epoch - 4ms/step  
## Epoch 13/50  
## 36/36 - 0s - loss: 0.4348 - accuracy: 0.8369 - val\_loss: 0.4643 - val\_accuracy: 0.8129 - 137ms/epoch - 4ms/step  
## Epoch 14/50  
## 36/36 - 0s - loss: 0.4293 - accuracy: 0.8368 - val\_loss: 0.4498 - val\_accuracy: 0.8230 - 139ms/epoch - 4ms/step  
## Epoch 15/50  
## 36/36 - 0s - loss: 0.4216 - accuracy: 0.8425 - val\_loss: 0.4411 - val\_accuracy: 0.8208 - 137ms/epoch - 4ms/step  
## Epoch 16/50  
## 36/36 - 0s - loss: 0.4135 - accuracy: 0.8413 - val\_loss: 0.4381 - val\_accuracy: 0.8226 - 143ms/epoch - 4ms/step  
## Epoch 17/50  
## 36/36 - 0s - loss: 0.4068 - accuracy: 0.8430 - val\_loss: 0.4313 - val\_accuracy: 0.8266 - 139ms/epoch - 4ms/step  
## Epoch 18/50  
## 36/36 - 0s - loss: 0.3998 - accuracy: 0.8444 - val\_loss: 0.4294 - val\_accuracy: 0.8241 - 135ms/epoch - 4ms/step  
## Epoch 19/50  
## 36/36 - 0s - loss: 0.3984 - accuracy: 0.8442 - val\_loss: 0.4136 - val\_accuracy: 0.8369 - 142ms/epoch - 4ms/step  
## Epoch 20/50  
## 36/36 - 0s - loss: 0.3932 - accuracy: 0.8465 - val\_loss: 0.4088 - val\_accuracy: 0.8407 - 137ms/epoch - 4ms/step  
## Epoch 21/50  
## 36/36 - 0s - loss: 0.3902 - accuracy: 0.8452 - val\_loss: 0.4075 - val\_accuracy: 0.8361 - 139ms/epoch - 4ms/step  
## Epoch 22/50  
## 36/36 - 0s - loss: 0.3839 - accuracy: 0.8475 - val\_loss: 0.4052 - val\_accuracy: 0.8365 - 136ms/epoch - 4ms/step  
## Epoch 23/50  
## 36/36 - 0s - loss: 0.3805 - accuracy: 0.8509 - val\_loss: 0.3999 - val\_accuracy: 0.8384 - 141ms/epoch - 4ms/step  
## Epoch 24/50  
## 36/36 - 0s - loss: 0.3778 - accuracy: 0.8491 - val\_loss: 0.3921 - val\_accuracy: 0.8460 - 141ms/epoch - 4ms/step  
## Epoch 25/50  
## 36/36 - 0s - loss: 0.3735 - accuracy: 0.8522 - val\_loss: 0.3922 - val\_accuracy: 0.8443 - 136ms/epoch - 4ms/step  
## Epoch 26/50  
## 36/36 - 0s - loss: 0.3706 - accuracy: 0.8518 - val\_loss: 0.3846 - val\_accuracy: 0.8439 - 141ms/epoch - 4ms/step  
## Epoch 27/50  
## 36/36 - 0s - loss: 0.3685 - accuracy: 0.8518 - val\_loss: 0.3908 - val\_accuracy: 0.8428 - 138ms/epoch - 4ms/step  
## Epoch 28/50  
## 36/36 - 0s - loss: 0.3639 - accuracy: 0.8541 - val\_loss: 0.3927 - val\_accuracy: 0.8406 - 137ms/epoch - 4ms/step

plot(history\_small)



for (x in 1:4) {  
 if (x %% 2 == 0) {  
 print(max(unlist(history\_small$metrics[x])))  
 } else {  
 print(min(unlist(history\_small$metrics[x])))  
 }  
}

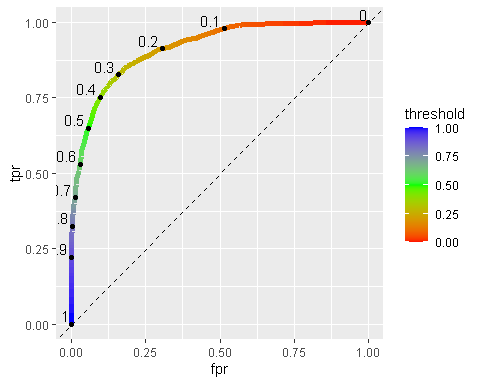
## [1] 0.3638667  
## [1] 0.8541381  
## [1] 0.3845876  
## [1] 0.8460423

### Evaluate smaller model

# Evaluate the smaller capacity model  
  
predictions <- predict(model\_small, test\_features)

## 284/284 - 0s - 239ms/epoch - 843us/step

test\_set$p\_prob2 <- 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\_prob2 >= i, ]  
   
 fpr <- sum(over\_threshold$booking\_status==0)/sum(test\_set$booking\_status==0)  
 roc\_data[roc\_data$threshold==i, "fpr"] <- fpr  
   
 tpr <- sum(over\_threshold$booking\_status==1)/sum(test\_set$booking\_status==1)  
 roc\_data[roc\_data$threshold==i, "tpr"] <- tpr  
   
}  
  
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))



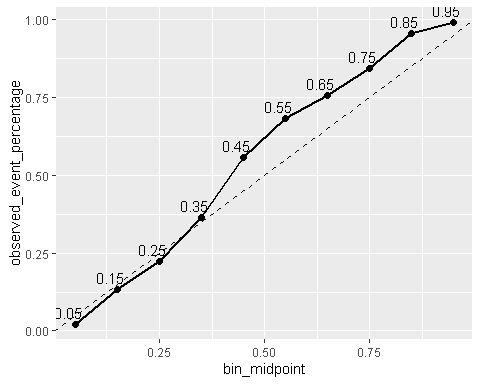
# AUC  
  
auc <- auc(x = roc\_data$fpr, y = roc\_data$tpr, type = "spline")

## Warning in regularize.values(x, y, ties, missing(ties)): collapsing to unique  
## 'x' values

auc

## [1] 0.915843

# Calibration curve  
  
calibration\_data <- data.frame(bin\_midpoint=seq(0.05, 0.95, 0.1),  
 observed\_event\_percentage=0)  
for (i in seq(0.05,0.95,0.1)) {  
   
 in\_interval <- test\_set[test\_set$p\_prob2 >= (i-0.05) & test\_set$p\_prob2 <= (i+0.05), ]  
 oep <- nrow(in\_interval[in\_interval$booking\_status==1, ])/nrow(in\_interval)  
 calibration\_data[calibration\_data$bin\_midpoint==i, "observed\_event\_percentage"] <- oep  
   
}  
  
ggplot(data = calibration\_data, aes(x = bin\_midpoint, y = observed\_event\_percentage)) +  
 geom\_line(size = 1) +  
 geom\_abline(intercept = 0, slope = 1, lty = 2) +  
 geom\_point(size = 2) +  
 geom\_text(aes(label = bin\_midpoint), hjust = 0.75, vjust = -0.5)



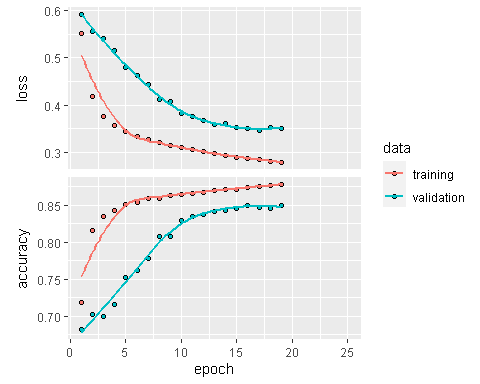
## Appendix

### Add early stop and batch normalization model

model <- keras\_model\_sequential(list(  
 layer\_dense(units = 75, activation = "relu"),  
 layer\_batch\_normalization(),  
 layer\_dense(units = 37, activation = "relu"),  
 layer\_batch\_normalization(),  
 layer\_dense(units = 1, activation = "sigmoid")  
))  
  
compile(model,  
 optimizer = "rmsprop",  
 loss = "binary\_crossentropy",  
 metrics = "accuracy")  
  
history <- fit(model, training\_features, training\_labels,  
 epochs = 25, batch\_size = 512, validation\_split = 0.33,  
 callbacks = list(callback\_early\_stopping(patience = 2)))

## Epoch 1/25  
## 36/36 - 1s - loss: 0.5510 - accuracy: 0.7179 - val\_loss: 0.5912 - val\_accuracy: 0.6824 - 853ms/epoch - 24ms/step  
## Epoch 2/25  
## 36/36 - 0s - loss: 0.4176 - accuracy: 0.8158 - val\_loss: 0.5554 - val\_accuracy: 0.7026 - 115ms/epoch - 3ms/step  
## Epoch 3/25  
## 36/36 - 0s - loss: 0.3766 - accuracy: 0.8351 - val\_loss: 0.5406 - val\_accuracy: 0.6988 - 117ms/epoch - 3ms/step  
## Epoch 4/25  
## 36/36 - 0s - loss: 0.3568 - accuracy: 0.8429 - val\_loss: 0.5165 - val\_accuracy: 0.7153 - 119ms/epoch - 3ms/step  
## Epoch 5/25  
## 36/36 - 0s - loss: 0.3440 - accuracy: 0.8507 - val\_loss: 0.4804 - val\_accuracy: 0.7526 - 115ms/epoch - 3ms/step  
## Epoch 6/25  
## 36/36 - 0s - loss: 0.3341 - accuracy: 0.8535 - val\_loss: 0.4619 - val\_accuracy: 0.7619 - 118ms/epoch - 3ms/step  
## Epoch 7/25  
## 36/36 - 0s - loss: 0.3268 - accuracy: 0.8586 - val\_loss: 0.4429 - val\_accuracy: 0.7776 - 115ms/epoch - 3ms/step  
## Epoch 8/25  
## 36/36 - 0s - loss: 0.3214 - accuracy: 0.8596 - val\_loss: 0.4120 - val\_accuracy: 0.8084 - 118ms/epoch - 3ms/step  
## Epoch 9/25  
## 36/36 - 0s - loss: 0.3152 - accuracy: 0.8630 - val\_loss: 0.4073 - val\_accuracy: 0.8072 - 118ms/epoch - 3ms/step  
## Epoch 10/25  
## 36/36 - 0s - loss: 0.3094 - accuracy: 0.8650 - val\_loss: 0.3817 - val\_accuracy: 0.8299 - 115ms/epoch - 3ms/step  
## Epoch 11/25  
## 36/36 - 0s - loss: 0.3052 - accuracy: 0.8662 - val\_loss: 0.3752 - val\_accuracy: 0.8349 - 117ms/epoch - 3ms/step  
## Epoch 12/25  
## 36/36 - 0s - loss: 0.3015 - accuracy: 0.8667 - val\_loss: 0.3674 - val\_accuracy: 0.8381 - 119ms/epoch - 3ms/step  
## Epoch 13/25  
## 36/36 - 0s - loss: 0.2972 - accuracy: 0.8698 - val\_loss: 0.3598 - val\_accuracy: 0.8410 - 115ms/epoch - 3ms/step  
## Epoch 14/25  
## 36/36 - 0s - loss: 0.2934 - accuracy: 0.8708 - val\_loss: 0.3604 - val\_accuracy: 0.8425 - 117ms/epoch - 3ms/step  
## Epoch 15/25  
## 36/36 - 0s - loss: 0.2896 - accuracy: 0.8714 - val\_loss: 0.3525 - val\_accuracy: 0.8453 - 114ms/epoch - 3ms/step  
## Epoch 16/25  
## 36/36 - 0s - loss: 0.2874 - accuracy: 0.8744 - val\_loss: 0.3511 - val\_accuracy: 0.8495 - 118ms/epoch - 3ms/step  
## Epoch 17/25  
## 36/36 - 0s - loss: 0.2849 - accuracy: 0.8751 - val\_loss: 0.3467 - val\_accuracy: 0.8470 - 117ms/epoch - 3ms/step  
## Epoch 18/25  
## 36/36 - 0s - loss: 0.2810 - accuracy: 0.8761 - val\_loss: 0.3526 - val\_accuracy: 0.8453 - 115ms/epoch - 3ms/step  
## Epoch 19/25  
## 36/36 - 0s - loss: 0.2788 - accuracy: 0.8782 - val\_loss: 0.3495 - val\_accuracy: 0.8503 - 114ms/epoch - 3ms/step

plot(history)



for (x in 1:4) {  
 if (x %% 2 == 0) {  
 print(max(unlist(history$metrics[x])))  
 } else {  
 print(min(unlist(history$metrics[x])))  
 }  
}

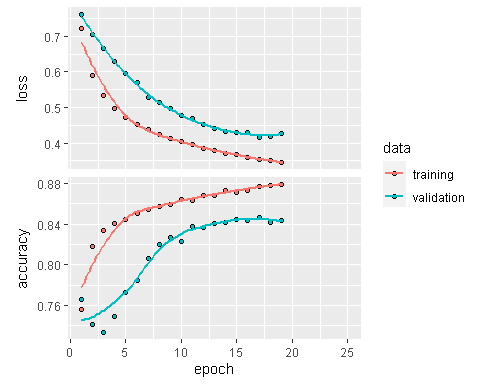
## [1] 0.2788157  
## [1] 0.8781921  
## [1] 0.3466582  
## [1] 0.8502787

### Add L2 regularization

model <- keras\_model\_sequential(list(  
 layer\_dense(units = 75, activation = "relu",  
 kernel\_regularizer = regularizer\_l2(0.002)),  
 layer\_batch\_normalization(),  
 layer\_dense(units = 37, activation = "relu",  
 kernel\_regularizer = regularizer\_l2(0.002)),  
 layer\_batch\_normalization(),  
 layer\_dense(units = 1, activation = "sigmoid")  
))  
  
compile(model,  
 optimizer = "rmsprop",  
 loss = "binary\_crossentropy",  
 metrics = "accuracy")  
  
history <- fit(model, training\_features, training\_labels,  
 epochs = 25, batch\_size = 512, validation\_split = 0.33,  
 callbacks = list(callback\_early\_stopping(patience = 2)))

## Epoch 1/25  
## 36/36 - 1s - loss: 0.7206 - accuracy: 0.7556 - val\_loss: 0.7603 - val\_accuracy: 0.7657 - 874ms/epoch - 24ms/step  
## Epoch 2/25  
## 36/36 - 0s - loss: 0.5908 - accuracy: 0.8176 - val\_loss: 0.7037 - val\_accuracy: 0.7417 - 119ms/epoch - 3ms/step  
## Epoch 3/25  
## 36/36 - 0s - loss: 0.5331 - accuracy: 0.8340 - val\_loss: 0.6651 - val\_accuracy: 0.7338 - 117ms/epoch - 3ms/step  
## Epoch 4/25  
## 36/36 - 0s - loss: 0.4984 - accuracy: 0.8410 - val\_loss: 0.6290 - val\_accuracy: 0.7488 - 116ms/epoch - 3ms/step  
## Epoch 5/25  
## 36/36 - 0s - loss: 0.4728 - accuracy: 0.8448 - val\_loss: 0.5952 - val\_accuracy: 0.7727 - 117ms/epoch - 3ms/step  
## Epoch 6/25  
## 36/36 - 0s - loss: 0.4530 - accuracy: 0.8505 - val\_loss: 0.5696 - val\_accuracy: 0.7849 - 114ms/epoch - 3ms/step  
## Epoch 7/25  
## 36/36 - 0s - loss: 0.4377 - accuracy: 0.8545 - val\_loss: 0.5285 - val\_accuracy: 0.8059 - 121ms/epoch - 3ms/step  
## Epoch 8/25  
## 36/36 - 0s - loss: 0.4252 - accuracy: 0.8572 - val\_loss: 0.5139 - val\_accuracy: 0.8202 - 116ms/epoch - 3ms/step  
## Epoch 9/25  
## 36/36 - 0s - loss: 0.4136 - accuracy: 0.8592 - val\_loss: 0.4975 - val\_accuracy: 0.8270 - 120ms/epoch - 3ms/step  
## Epoch 10/25  
## 36/36 - 0s - loss: 0.4035 - accuracy: 0.8637 - val\_loss: 0.4781 - val\_accuracy: 0.8231 - 123ms/epoch - 3ms/step  
## Epoch 11/25  
## 36/36 - 0s - loss: 0.3953 - accuracy: 0.8637 - val\_loss: 0.4683 - val\_accuracy: 0.8373 - 139ms/epoch - 4ms/step  
## Epoch 12/25  
## 36/36 - 0s - loss: 0.3859 - accuracy: 0.8681 - val\_loss: 0.4527 - val\_accuracy: 0.8365 - 122ms/epoch - 3ms/step  
## Epoch 13/25  
## 36/36 - 0s - loss: 0.3804 - accuracy: 0.8685 - val\_loss: 0.4405 - val\_accuracy: 0.8408 - 119ms/epoch - 3ms/step  
## Epoch 14/25  
## 36/36 - 0s - loss: 0.3719 - accuracy: 0.8731 - val\_loss: 0.4329 - val\_accuracy: 0.8419 - 120ms/epoch - 3ms/step  
## Epoch 15/25  
## 36/36 - 0s - loss: 0.3672 - accuracy: 0.8711 - val\_loss: 0.4288 - val\_accuracy: 0.8444 - 116ms/epoch - 3ms/step  
## Epoch 16/25  
## 36/36 - 0s - loss: 0.3611 - accuracy: 0.8734 - val\_loss: 0.4304 - val\_accuracy: 0.8439 - 127ms/epoch - 4ms/step  
## Epoch 17/25  
## 36/36 - 0s - loss: 0.3545 - accuracy: 0.8769 - val\_loss: 0.4160 - val\_accuracy: 0.8464 - 117ms/epoch - 3ms/step  
## Epoch 18/25  
## 36/36 - 0s - loss: 0.3503 - accuracy: 0.8780 - val\_loss: 0.4198 - val\_accuracy: 0.8415 - 117ms/epoch - 3ms/step  
## Epoch 19/25  
## 36/36 - 0s - loss: 0.3461 - accuracy: 0.8790 - val\_loss: 0.4265 - val\_accuracy: 0.8438 - 118ms/epoch - 3ms/step

plot(history)



for (x in 1:4) {  
 if (x %% 2 == 0) {  
 print(max(unlist(history$metrics[x])))  
 } else {  
 print(min(unlist(history$metrics[x])))  
 }  
}

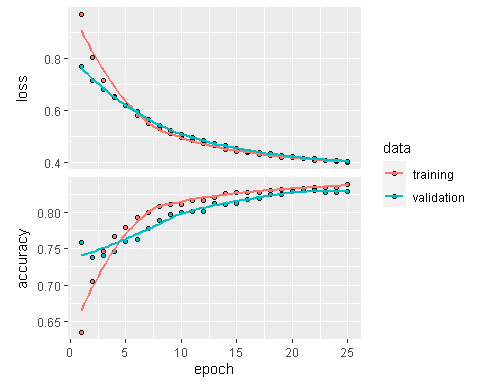
## [1] 0.3461378  
## [1] 0.878961  
## [1] 0.4160261  
## [1] 0.8463768

### Add Dropout

model <- keras\_model\_sequential(list(  
 layer\_dense(units = 75, activation = "relu",  
 kernel\_regularizer = regularizer\_l2(0.002)),  
 layer\_batch\_normalization(),  
 layer\_dropout(rate=0.5),  
 layer\_dense(units = 37, activation = "relu",  
 kernel\_regularizer = regularizer\_l2(0.002)),  
 layer\_batch\_normalization(),  
 layer\_dropout(rate=0.5),  
 layer\_dense(units = 1, activation = "sigmoid")  
))  
  
compile(model,  
 optimizer = "rmsprop",  
 loss = "binary\_crossentropy",  
 metrics = "accuracy")  
  
history <- fit(model, training\_features, training\_labels,  
 epochs = 25, batch\_size = 512, validation\_split = 0.33,  
 callbacks = list(callback\_early\_stopping(patience = 2)))

## Epoch 1/25  
## 36/36 - 1s - loss: 0.9706 - accuracy: 0.6350 - val\_loss: 0.7704 - val\_accuracy: 0.7592 - 1s/epoch - 29ms/step  
## Epoch 2/25  
## 36/36 - 0s - loss: 0.8072 - accuracy: 0.7049 - val\_loss: 0.7181 - val\_accuracy: 0.7379 - 140ms/epoch - 4ms/step  
## Epoch 3/25  
## 36/36 - 0s - loss: 0.7178 - accuracy: 0.7461 - val\_loss: 0.6806 - val\_accuracy: 0.7411 - 147ms/epoch - 4ms/step  
## Epoch 4/25  
## 36/36 - 0s - loss: 0.6564 - accuracy: 0.7668 - val\_loss: 0.6496 - val\_accuracy: 0.7469 - 139ms/epoch - 4ms/step  
## Epoch 5/25  
## 36/36 - 0s - loss: 0.6199 - accuracy: 0.7790 - val\_loss: 0.6205 - val\_accuracy: 0.7604 - 150ms/epoch - 4ms/step  
## Epoch 6/25  
## 36/36 - 0s - loss: 0.5812 - accuracy: 0.7932 - val\_loss: 0.5959 - val\_accuracy: 0.7633 - 136ms/epoch - 4ms/step  
## Epoch 7/25  
## 36/36 - 0s - loss: 0.5516 - accuracy: 0.7997 - val\_loss: 0.5663 - val\_accuracy: 0.7785 - 137ms/epoch - 4ms/step  
## Epoch 8/25  
## 36/36 - 0s - loss: 0.5320 - accuracy: 0.8087 - val\_loss: 0.5430 - val\_accuracy: 0.7890 - 137ms/epoch - 4ms/step  
## Epoch 9/25  
## 36/36 - 0s - loss: 0.5112 - accuracy: 0.8119 - val\_loss: 0.5223 - val\_accuracy: 0.7979 - 142ms/epoch - 4ms/step  
## Epoch 10/25  
## 36/36 - 0s - loss: 0.4978 - accuracy: 0.8113 - val\_loss: 0.5099 - val\_accuracy: 0.8004 - 168ms/epoch - 5ms/step  
## Epoch 11/25  
## 36/36 - 0s - loss: 0.4847 - accuracy: 0.8175 - val\_loss: 0.4957 - val\_accuracy: 0.8018 - 143ms/epoch - 4ms/step  
## Epoch 12/25  
## 36/36 - 0s - loss: 0.4742 - accuracy: 0.8175 - val\_loss: 0.4843 - val\_accuracy: 0.8022 - 154ms/epoch - 4ms/step  
## Epoch 13/25  
## 36/36 - 0s - loss: 0.4663 - accuracy: 0.8215 - val\_loss: 0.4734 - val\_accuracy: 0.8130 - 143ms/epoch - 4ms/step  
## Epoch 14/25  
## 36/36 - 0s - loss: 0.4518 - accuracy: 0.8271 - val\_loss: 0.4646 - val\_accuracy: 0.8119 - 143ms/epoch - 4ms/step  
## Epoch 15/25  
## 36/36 - 0s - loss: 0.4439 - accuracy: 0.8278 - val\_loss: 0.4559 - val\_accuracy: 0.8132 - 139ms/epoch - 4ms/step  
## Epoch 16/25  
## 36/36 - 0s - loss: 0.4384 - accuracy: 0.8284 - val\_loss: 0.4448 - val\_accuracy: 0.8184 - 146ms/epoch - 4ms/step  
## Epoch 17/25  
## 36/36 - 0s - loss: 0.4324 - accuracy: 0.8280 - val\_loss: 0.4402 - val\_accuracy: 0.8195 - 144ms/epoch - 4ms/step  
## Epoch 18/25  
## 36/36 - 0s - loss: 0.4288 - accuracy: 0.8305 - val\_loss: 0.4339 - val\_accuracy: 0.8252 - 150ms/epoch - 4ms/step  
## Epoch 19/25  
## 36/36 - 0s - loss: 0.4201 - accuracy: 0.8315 - val\_loss: 0.4283 - val\_accuracy: 0.8252 - 153ms/epoch - 4ms/step  
## Epoch 20/25  
## 36/36 - 0s - loss: 0.4203 - accuracy: 0.8317 - val\_loss: 0.4223 - val\_accuracy: 0.8292 - 150ms/epoch - 4ms/step  
## Epoch 21/25  
## 36/36 - 0s - loss: 0.4140 - accuracy: 0.8333 - val\_loss: 0.4173 - val\_accuracy: 0.8305 - 140ms/epoch - 4ms/step  
## Epoch 22/25  
## 36/36 - 0s - loss: 0.4100 - accuracy: 0.8351 - val\_loss: 0.4156 - val\_accuracy: 0.8300 - 145ms/epoch - 4ms/step  
## Epoch 23/25  
## 36/36 - 0s - loss: 0.4089 - accuracy: 0.8354 - val\_loss: 0.4091 - val\_accuracy: 0.8282 - 140ms/epoch - 4ms/step  
## Epoch 24/25  
## 36/36 - 0s - loss: 0.4058 - accuracy: 0.8352 - val\_loss: 0.4068 - val\_accuracy: 0.8281 - 144ms/epoch - 4ms/step  
## Epoch 25/25  
## 36/36 - 0s - loss: 0.4010 - accuracy: 0.8390 - val\_loss: 0.4048 - val\_accuracy: 0.8289 - 144ms/epoch - 4ms/step

plot(history)



for (x in 1:4) {  
 if (x %% 2 == 0) {  
 print(max(unlist(history$metrics[x])))  
 } else {  
 print(min(unlist(history$metrics[x])))  
 }  
}

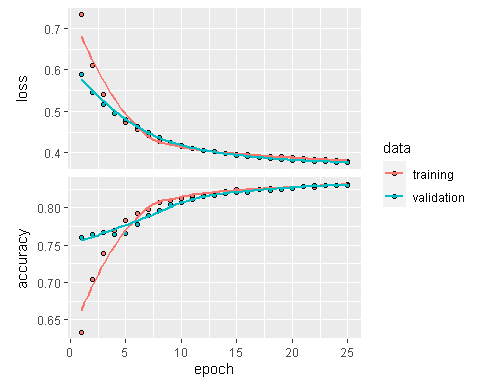
## [1] 0.4010183  
## [1] 0.8390356  
## [1] 0.4047619  
## [1] 0.8305463

# Remove regularization

model <- keras\_model\_sequential(list(  
 layer\_dense(units = 75, activation = "relu"),  
 layer\_batch\_normalization(),  
 layer\_dropout(rate=0.5),  
 layer\_dense(units = 37, activation = "relu"),  
 layer\_batch\_normalization(),  
 layer\_dropout(rate=0.5),  
 layer\_dense(units = 1, activation = "sigmoid")  
))  
  
compile(model,  
 optimizer = "rmsprop",  
 loss = "binary\_crossentropy",  
 metrics = "accuracy")  
  
history <- fit(model, training\_features, training\_labels,  
 epochs = 25, batch\_size = 512, validation\_split = 0.33,  
 callbacks = list(callback\_early\_stopping(patience = 2)))

## Epoch 1/25  
## 36/36 - 1s - loss: 0.7336 - accuracy: 0.6335 - val\_loss: 0.5887 - val\_accuracy: 0.7595 - 928ms/epoch - 26ms/step  
## Epoch 2/25  
## 36/36 - 0s - loss: 0.6099 - accuracy: 0.7038 - val\_loss: 0.5458 - val\_accuracy: 0.7641 - 147ms/epoch - 4ms/step  
## Epoch 3/25  
## 36/36 - 0s - loss: 0.5408 - accuracy: 0.7385 - val\_loss: 0.5154 - val\_accuracy: 0.7673 - 140ms/epoch - 4ms/step  
## Epoch 4/25  
## 36/36 - 0s - loss: 0.4945 - accuracy: 0.7691 - val\_loss: 0.4946 - val\_accuracy: 0.7642 - 140ms/epoch - 4ms/step  
## Epoch 5/25  
## 36/36 - 0s - loss: 0.4736 - accuracy: 0.7833 - val\_loss: 0.4808 - val\_accuracy: 0.7656 - 141ms/epoch - 4ms/step  
## Epoch 6/25  
## 36/36 - 0s - loss: 0.4572 - accuracy: 0.7922 - val\_loss: 0.4624 - val\_accuracy: 0.7771 - 138ms/epoch - 4ms/step  
## Epoch 7/25  
## 36/36 - 0s - loss: 0.4407 - accuracy: 0.7980 - val\_loss: 0.4478 - val\_accuracy: 0.7900 - 144ms/epoch - 4ms/step  
## Epoch 8/25  
## 36/36 - 0s - loss: 0.4276 - accuracy: 0.8075 - val\_loss: 0.4372 - val\_accuracy: 0.7961 - 147ms/epoch - 4ms/step  
## Epoch 9/25  
## 36/36 - 0s - loss: 0.4226 - accuracy: 0.8092 - val\_loss: 0.4249 - val\_accuracy: 0.8041 - 141ms/epoch - 4ms/step  
## Epoch 10/25  
## 36/36 - 0s - loss: 0.4155 - accuracy: 0.8129 - val\_loss: 0.4173 - val\_accuracy: 0.8076 - 143ms/epoch - 4ms/step  
## Epoch 11/25  
## 36/36 - 0s - loss: 0.4091 - accuracy: 0.8153 - val\_loss: 0.4111 - val\_accuracy: 0.8114 - 144ms/epoch - 4ms/step  
## Epoch 12/25  
## 36/36 - 0s - loss: 0.4062 - accuracy: 0.8178 - val\_loss: 0.4065 - val\_accuracy: 0.8157 - 138ms/epoch - 4ms/step  
## Epoch 13/25  
## 36/36 - 0s - loss: 0.4040 - accuracy: 0.8196 - val\_loss: 0.4030 - val\_accuracy: 0.8171 - 141ms/epoch - 4ms/step  
## Epoch 14/25  
## 36/36 - 0s - loss: 0.3991 - accuracy: 0.8225 - val\_loss: 0.3978 - val\_accuracy: 0.8192 - 148ms/epoch - 4ms/step  
## Epoch 15/25  
## 36/36 - 0s - loss: 0.3951 - accuracy: 0.8240 - val\_loss: 0.3936 - val\_accuracy: 0.8203 - 138ms/epoch - 4ms/step  
## Epoch 16/25  
## 36/36 - 0s - loss: 0.3954 - accuracy: 0.8235 - val\_loss: 0.3917 - val\_accuracy: 0.8201 - 141ms/epoch - 4ms/step  
## Epoch 17/25  
## 36/36 - 0s - loss: 0.3917 - accuracy: 0.8249 - val\_loss: 0.3875 - val\_accuracy: 0.8242 - 142ms/epoch - 4ms/step  
## Epoch 18/25  
## 36/36 - 0s - loss: 0.3918 - accuracy: 0.8260 - val\_loss: 0.3868 - val\_accuracy: 0.8237 - 141ms/epoch - 4ms/step  
## Epoch 19/25  
## 36/36 - 0s - loss: 0.3898 - accuracy: 0.8244 - val\_loss: 0.3838 - val\_accuracy: 0.8251 - 142ms/epoch - 4ms/step  
## Epoch 20/25  
## 36/36 - 0s - loss: 0.3879 - accuracy: 0.8261 - val\_loss: 0.3822 - val\_accuracy: 0.8263 - 143ms/epoch - 4ms/step  
## Epoch 21/25  
## 36/36 - 0s - loss: 0.3853 - accuracy: 0.8285 - val\_loss: 0.3802 - val\_accuracy: 0.8291 - 147ms/epoch - 4ms/step  
## Epoch 22/25  
## 36/36 - 0s - loss: 0.3846 - accuracy: 0.8276 - val\_loss: 0.3795 - val\_accuracy: 0.8292 - 161ms/epoch - 4ms/step  
## Epoch 23/25  
## 36/36 - 0s - loss: 0.3842 - accuracy: 0.8303 - val\_loss: 0.3782 - val\_accuracy: 0.8303 - 140ms/epoch - 4ms/step  
## Epoch 24/25  
## 36/36 - 0s - loss: 0.3819 - accuracy: 0.8302 - val\_loss: 0.3771 - val\_accuracy: 0.8293 - 138ms/epoch - 4ms/step  
## Epoch 25/25  
## 36/36 - 0s - loss: 0.3794 - accuracy: 0.8313 - val\_loss: 0.3767 - val\_accuracy: 0.8294 - 142ms/epoch - 4ms/step

plot(history)



for (x in 1:4) {  
 if (x %% 2 == 0) {  
 print(max(unlist(history$metrics[x])))  
 } else {  
 print(min(unlist(history$metrics[x])))  
 }  
}

## [1] 0.3793762  
## [1] 0.8312922  
## [1] 0.3767341  
## [1] 0.8303233