Epoch 1498/1500

96/96 [==============================] - 1s 6ms/step - loss: 0.1438 - categorical\_accuracy: 0.9477 - val\_loss: 1.5665 - val\_categorical\_accuracy: 0.8321

Epoch 1499/1500

96/96 [==============================] - 1s 6ms/step - loss: 0.1435 - categorical\_accuracy: 0.9464 - val\_loss: 1.5837 - val\_categorical\_accuracy: 0.8247

Epoch 1500/1500

96/96 [==============================] - 1s 6ms/step - loss: 0.1436 - categorical\_accuracy: 0.9477 - val\_loss: 1.7059 - val\_categorical\_accuracy: 0.7937

# Preprocessing

```{r Create a recipe for preproc}

set.seed(713)

trainIndex <- initial\_split(data, prop = 0.8, strata = status)

trainingSet <- training(trainIndex)

testSet <- testing(trainIndex)

set.seed(12)

preprocRecipe <-

recipe(status ~., data = data) %>%

step\_dummy(all\_nominal(), one\_hot = TRUE) %>%

step\_range(all\_predictors(), -all\_nominal(), min = 0, max = 1)

```

# In this step the above defined receipt is extracted using the `prep()` function, and then use the `bake()` function to transform a set of data based on that recipe.

```{r Prep and bake the defined recipe}

trainingSet\_processed <- preprocRecipe %>%

prep(trainingSet) %>%

bake(trainingSet)

testSet\_processed <- preprocRecipe %>%

prep(testSet) %>%

bake(testSet)

## Build Model

```{r}

#train\_data <- matrix\_data

train\_data <- data.matrix(matrix\_data)

test\_data <- data.matrix(matrix\_data)

train\_targets <- data.matrix(matrix\_targets)

# Function to build the model

build\_model <- function() {

model <- keras\_model\_sequential() %>%

#layer\_batch\_normalization(axis = -1L, input\_shape = dim(train\_data)[[2]]) %>%

layer\_dense(units = 64, activation = "relu", input\_shape = dim(train\_data)[[2]]) %>%

layer\_dense(units = 128, activation = "relu") %>%

layer\_dense(units = 64, activation = "relu") %>%

layer\_dense(units = 8, activation = "softmax")

model %>% compile(

optimizer = "rmsprop",

loss = "categorical\_crossentropy",

metrics = "categorical\_accuracy"

)

}

```

