Appendix S5

R source code for the definition and training of a model with two convolutional model.

```
####################################
#          DEFINE THE MODEL
####################################


TENSOR_input <- layer_input(shape = DIM_MATRIX, name =
"TENSOR_input")
VEKTOR_input <- layer_input(shape = DIM_VECTOR, name =
"VEKTOR_input")


tensor_flat_out <- TENSOR_input %>%
  layer_conv_2d(filters = 4, kernel_size = c(1, 52), strides=c(1,
52),
                activation = "relu", input_shape = DIM_MATRIX) %>%
  layer_conv_2d(filters = 4, kernel_size = c(1, 6), activation =
"relu") %>%
  layer_flatten()


main_output <- layer_concatenate(c(tensor_flat_out, VEKTOR_input))
%>%
  layer_dropout(rate = 0.5) %>%
  layer_dense(units = 512, activation = "relu") %>%
  layer_dense(units = 256, activation = "relu") %>%
  layer_dense(units = 128, activation = "relu") %>%
  layer_dense(units = 1, activation = "sigmoid", name =
"main_output")


model <- keras_model(
  inputs = c(TENSOR_input, VEKTOR_input),
  outputs = c(main_output)
)
########################
#   COMPILE THE MODEL
########################


model %>% compile(
  loss = "binary_crossentropy",
```

```r
  optimizer = optimizer_adam(lr = 1e-5),
  metrics = c("accuracy",
              "binary_crossentropy",
              tf$keras$metrics$AUC(name="auc")
             )
)
###########################
#    TRAIN THE MODEL
# USING HOLDOUT VALIDATION
###########################

history <- model %>% fit(

x=list(TRAIN_TENSOR_ATC_ICD_MATCH_KONT_NAIVMM_KANDO_1_5_DIAG_S_ATC_S
_ARVECKA_4D, TRAIN_VEKTOR_DEMOGR_MATCH_KONT_NAIVMM_KANDO_1_5),
  y=FACIT_MM_TRAIN,
  epochs = 800,
  batch_size = 1000,
  validation_data =
list(list(VAL_TENSOR_ATC_ICD_MATCH_KONT_NAIVMM_KANDO_1_5_DIAG_S_ATC_
S_ARVECKA_4D_WEIGHED,
VAL_VEKTOR_DEMOGR_MATCH_KONT_NAIVMM_KANDO_1_5_WEIGHED),
FACIT_MM_VAL_WEIGHED),
  class_weight = list("0"=1,"1"=5),
  callbacks = list(
    callback_csv_logger(filename=paste0(logspath, modelnamebest,
".txt"),
                        append=TRUE, separator="\t"),
    callback_model_checkpoint(filepath=paste0(logspath,
modelnamebest,
                        "{epoch:02d}-{val_auc:.4f}", ".h5"),
                        save_best_only=TRUE,
                        monitor="val_auc",
                        mode="max")
  )
)
```