Electron Pion Efficiency

Feedforward Neural Networks

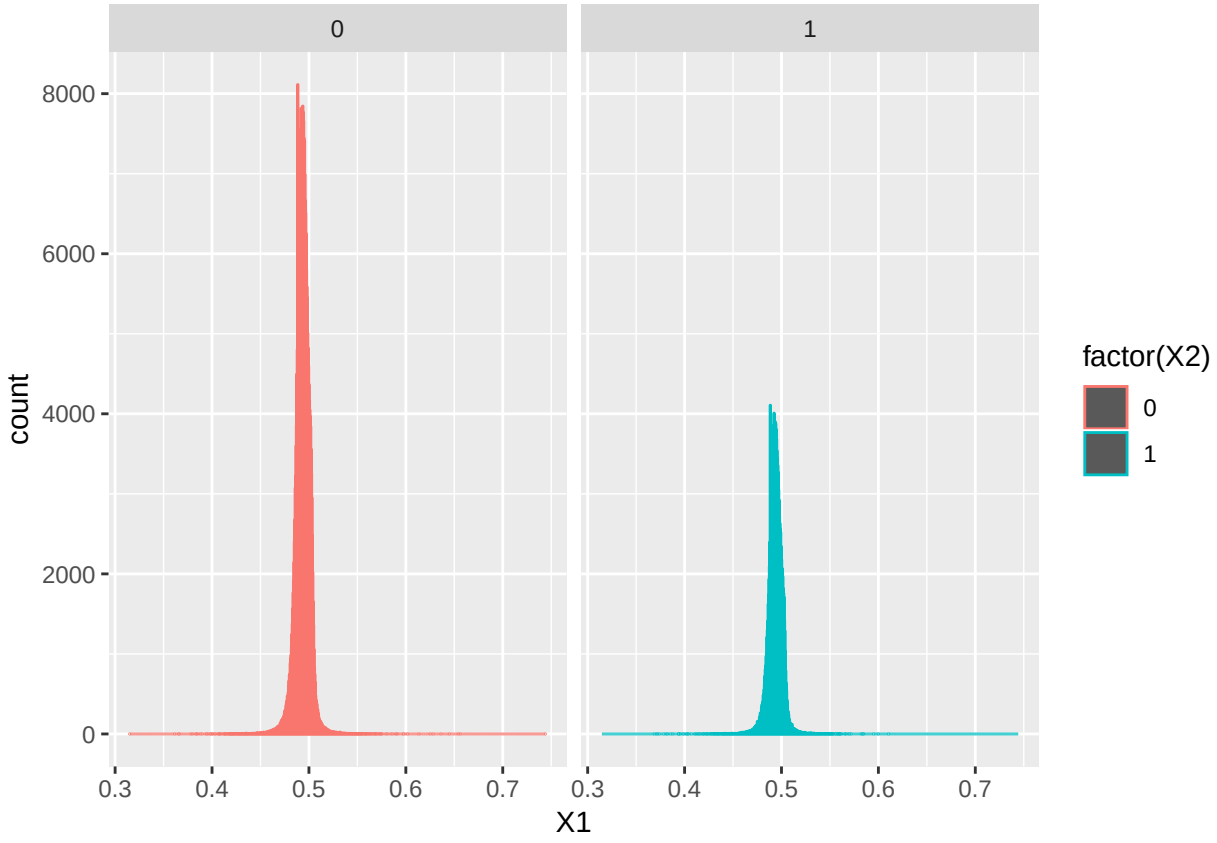
Model 1

```
rm(list=ls())

model_1.preds <- read.csv("C:/USers/gerhard/Documents/msc-hpc/round3/feedforward/round3_model1_results.

model_1.labels <- read.csv("C:/USers/gerhard/Documents/msc-hpc/round3/feedforward/round3_model1_y_test.

model_1 <- data.frame(cbind(model_1.preds[,2],model_1.labels[,2]))

model_1.electrons <- which(model_1[,2]==1)

electrons <- model_1[model_1.electrons,]

pions <- model_1[-as.numeric(model_1.electrons),]

electrons <- data.frame(electrons)

names(electrons) <- c("prediction","label")

pions <- data.frame(pions)

names(pions) <- c("prediction","label")


electron_efficiency <- function(electrons.,par){

  electrons <- electrons.

  electrons$electron_pred <- ifelse(electrons$prediction>=par[1],1,0)

  correct <- ifelse(electrons$electron_pred==electrons$label,1,0)

  error_metric <- sum(correct)/nrow(electrons)

  error_metric <- (error_metric-0.9)^2

  return(error_metric)

}

res <- optim(par=c(0),fn=electron_efficiency,lower = 0,upper = 1,electrons.=electrons,method="Brent")

require(ggplot2)
```

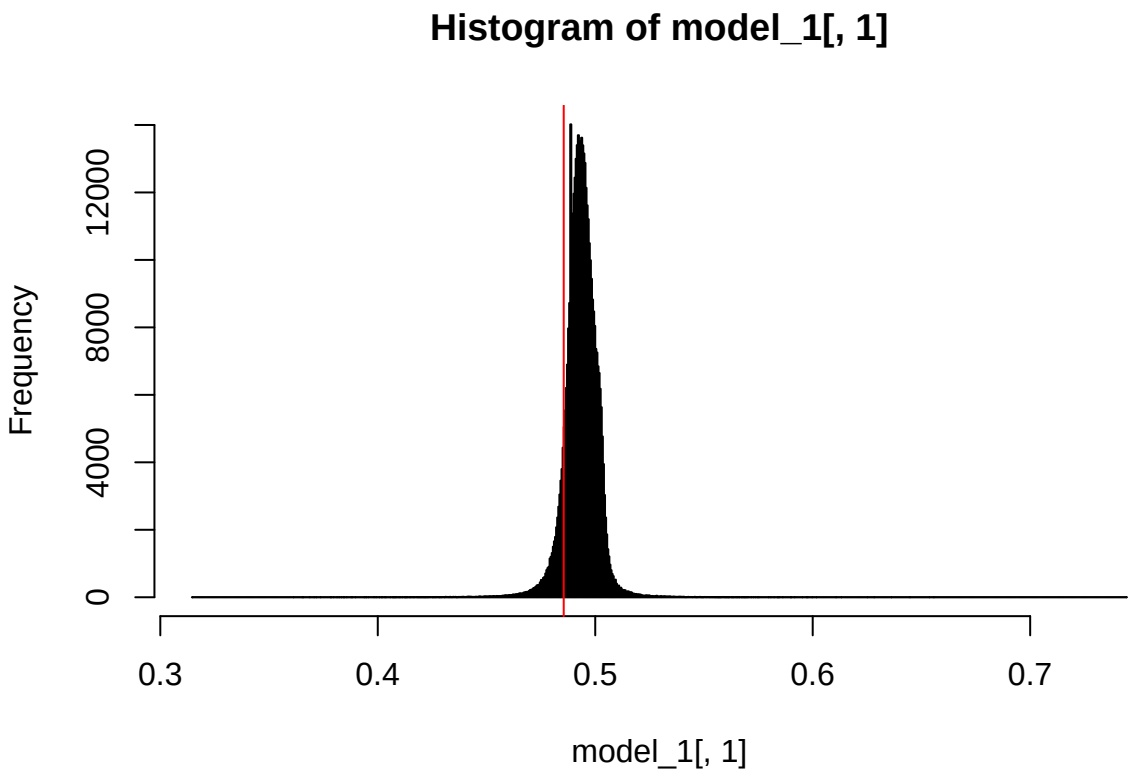
```
## Loading required package: ggplot2
## Registered S3 methods overwritten by 'ggplot2':
```

```
##   method         from
##   [.quosures     rlang
##   c.quosures     rlang
##   print.quosures rlang
```

```
ggplot(model_1,aes(X1,colour=factor(X2)))+geom_histogram(bins = 1000)+facet_wrap(factor(model_1$X2))
```




```
hist(model_1[,1],breaks=1000)
abline(v=res$par,col="red")
```




```
electrons$predicted_label <- ifelse(electrons$prediction>=res$par,1,0)

sum(electrons$predicted_label)/nrow(electrons)
```

```
## [1] 0.8999971
```

```
pions$predicted_label <- ifelse(pions$prediction>=res$par,1,0)

pions$misclassified_as_electron <- ifelse(pions$predicted_label==1,1,0)

sum(pions$misclassified_as_electron)/nrow(pions)
```

```
## [1] 0.8988313
```

```
model_1$final_pred <- ifelse(model_1$X1>=res$par,1,0)

require(caret)
```

```
## Loading required package: caret
## Loading required package: lattice
```

```
caret::confusionMatrix(data=factor(model_1$final_pred),reference = factor(model_1$X2))
```

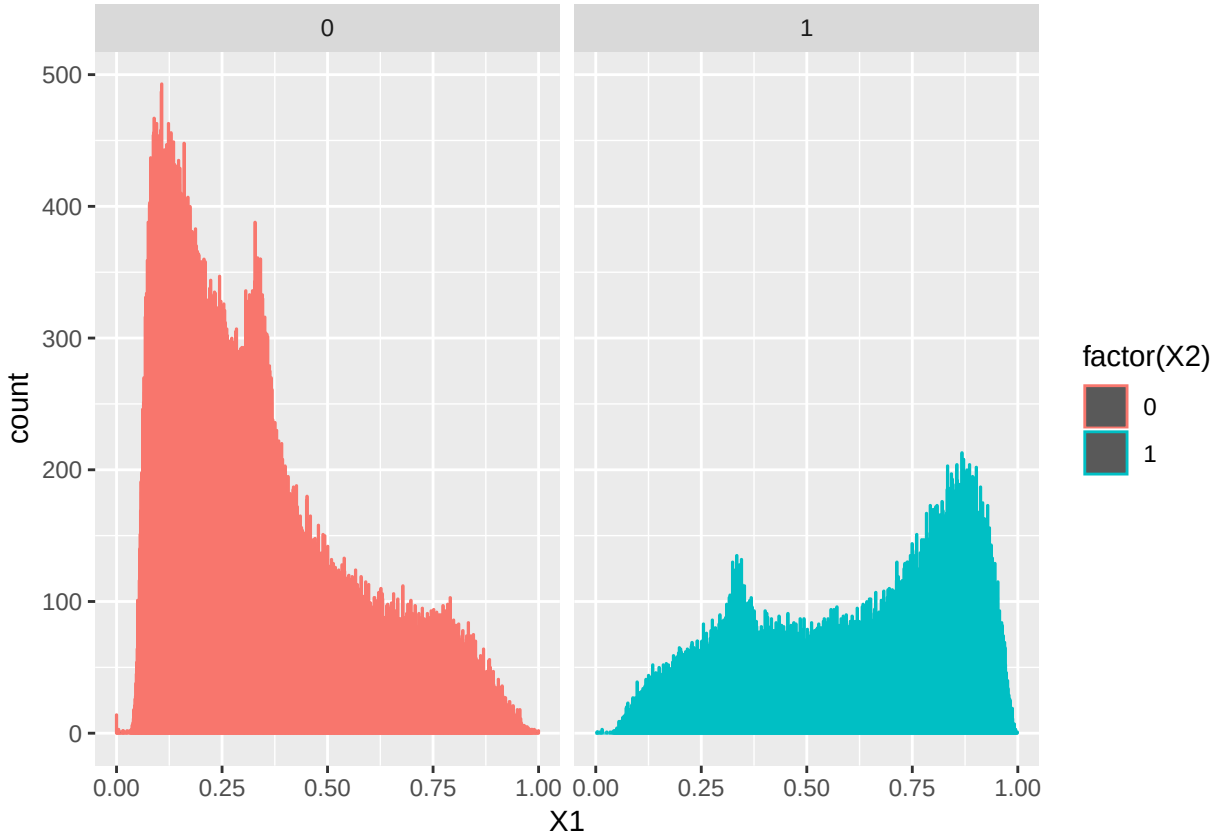
```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction      0      1
##          0  28531  13990
```

```
##          1 253483 125906
##
##                Accuracy : 0.366
##                  95% CI : (0.3646, 0.3675)
##     No Information Rate : 0.6684
##     P-Value [Acc > NIR] : 1
##
##                   Kappa : 8e-04
##
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.10117
##             Specificity : 0.90000
##          Pos Pred Value : 0.67099
##          Neg Pred Value : 0.33187
##              Prevalence : 0.66842
##          Detection Rate : 0.06762
##    Detection Prevalence : 0.10078
##       Balanced Accuracy : 0.50058
##
##        'Positive' Class : 0
##
```

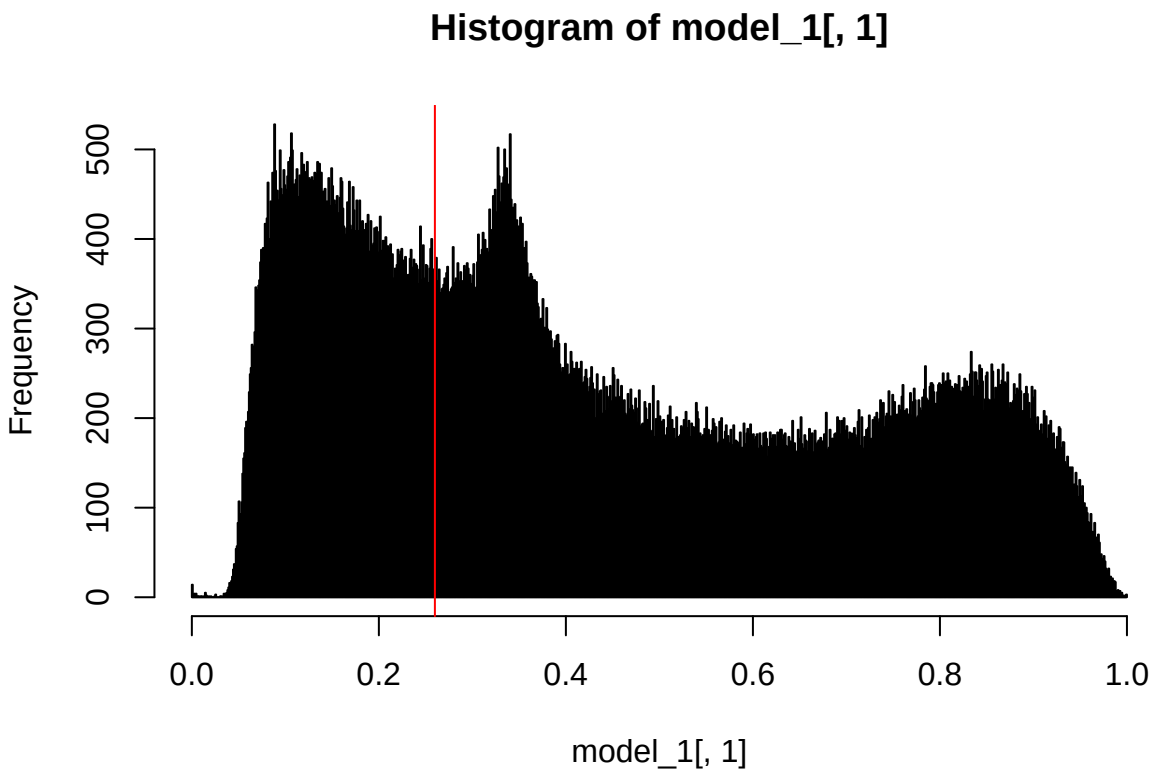
Model 2

```
rm(list=ls())

model_1.preds <- read.csv("C:/USers/gerhard/Documents/msc-hpc/round3/feedforward/local/round3_model1_re

model_1.labels <- read.csv("C:/USers/gerhard/Documents/msc-hpc/round3/feedforward/local/round3_model1_y

model_1 <- data.frame(cbind(model_1.preds[,2],model_1.labels[,2]))

model_1.electrons <- which(model_1[,2]==1)

electrons <- model_1[model_1.electrons,]

pions <- model_1[-as.numeric(model_1.electrons),]

electrons <- data.frame(electrons)

names(electrons) <- c("prediction","label")

pions <- data.frame(pions)

names(pions) <- c("prediction","label")


electron_efficiency <- function(electrons.,par){

  electrons <- electrons.

  electrons$electron_pred <- ifelse(electrons$prediction>=par[1],1,0)
```

```
  correct <- ifelse(electrons$electron_pred==electrons$label,1,0)

  error_metric <- sum(correct)/nrow(electrons)

  error_metric <- (error_metric-0.9)^2

  return(error_metric)
}

res <- optim(par=c(0),fn=electron_efficiency,lower = 0,upper = 1,electrons.=electrons,method="Brent")

require(ggplot2)

ggplot(model_1,aes(X1,colour=factor(X2)))+geom_histogram(bins = 1000)+facet_wrap(factor(model_1$X2))
```




```
hist(model_1[,1],breaks=1000)
abline(v=res$par,col="red")
```




```
electrons$predicted_label <- ifelse(electrons$prediction>=res$par,1,0)

sum(electrons$predicted_label)/nrow(electrons)
```

```
## [1] 0.9
```

```
pions$predicted_label <- ifelse(pions$prediction>=res$par,1,0)

pions$misclassified_as_electron <- ifelse(pions$predicted_label==1,1,0)

sum(pions$misclassified_as_electron)/nrow(pions)
```

```
## [1] 0.5389702
```

```
model_1$final_pred <- ifelse(model_1$X1>=res$par,1,0)

require(caret)

caret::confusionMatrix(data=factor(model_1$final_pred),reference = factor(model_1$X2))
```

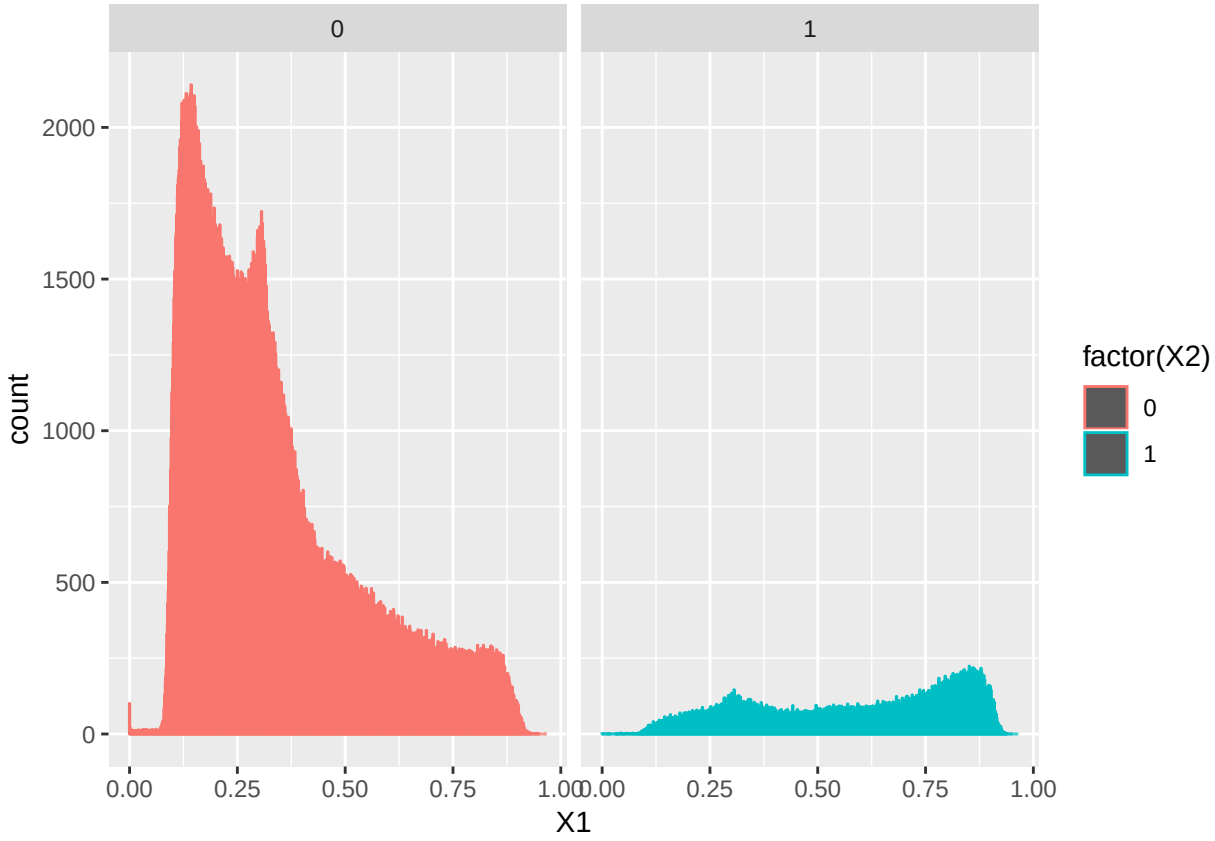
```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction     0     1
##          0 75430  8183
##          1 88182 73647
##
##                Accuracy : 0.6074
```

```
##                  95% CI : (0.6054, 0.6093)
##     No Information Rate : 0.6666
##     P-Value [Acc > NIR] : 1
##
##                   Kappa : 0.2901
##
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.4610
##             Specificity : 0.9000
##          Pos Pred Value : 0.9021
##          Neg Pred Value : 0.4551
##              Prevalence : 0.6666
##          Detection Rate : 0.3073
##    Detection Prevalence : 0.3407
##       Balanced Accuracy : 0.6805
##
##        'Positive' Class : 0
##
```

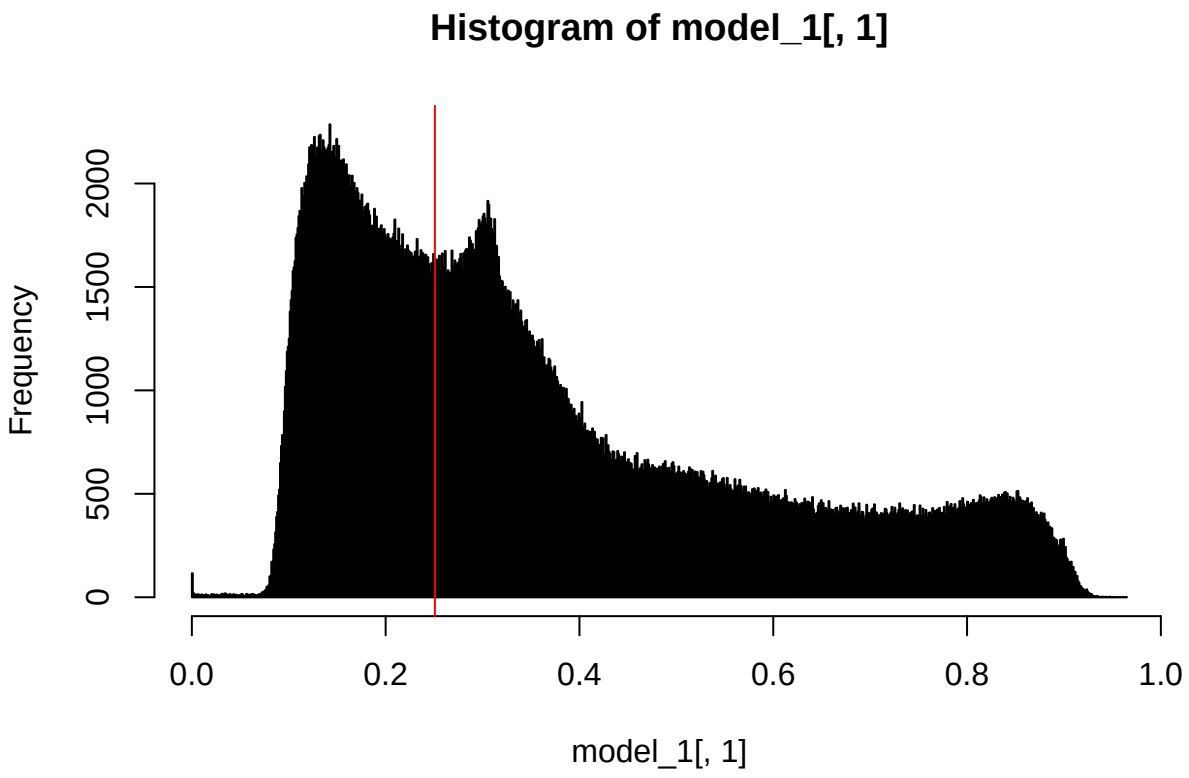
Model 3

```
rm(list=ls())

model_1.preds <- read.csv("C:/USers/gerhard/Documents/msc-hpc/round3/feedforward/local/round3_model2_re

model_1.labels <- read.csv("C:/USers/gerhard/Documents/msc-hpc/round3/feedforward/local/round3_model2_y

model_1 <- data.frame(cbind(model_1.preds[,2],model_1.labels[,2]))

model_1.electrons <- which(model_1[,2]==1)

electrons <- model_1[model_1.electrons,]

pions <- model_1[-as.numeric(model_1.electrons),]

electrons <- data.frame(electrons)

names(electrons) <- c("prediction","label")

pions <- data.frame(pions)

names(pions) <- c("prediction","label")


electron_efficiency <- function(electrons.,par){

  electrons <- electrons.

  electrons$electron_pred <- ifelse(electrons$prediction>=par[1],1,0)

  correct <- ifelse(electrons$electron_pred==electrons$label,1,0)
```

```
  error_metric <- sum(correct)/nrow(electrons)

  error_metric <- (error_metric-0.9)^2

  return(error_metric)

}

res <- optim(par=c(0),fn=electron_efficiency,lower = 0,upper = 1,electrons.=electrons,method="Brent")

require(ggplot2)

ggplot(model_1,aes(X1,colour=factor(X2)))+geom_histogram(bins = 1000)+facet_wrap(factor(model_1$X2))
```




```
hist(model_1[,1],breaks=1000)
abline(v=res$par,col="red")
```




```
electrons$predicted_label <- ifelse(electrons$prediction>=res$par,1,0)

sum(electrons$predicted_label)/nrow(electrons)
```

```
## [1] 0.9000012
```

```
pions$predicted_label <- ifelse(pions$prediction>=res$par,1,0)

pions$misclassified_as_electron <- ifelse(pions$predicted_label==1,1,0)

sum(pions$misclassified_as_electron)/nrow(pions)
```

```
## [1] 0.576587
```

```
model_1$final_pred <- ifelse(model_1$X1>=res$par,1,0)

require(caret)

caret::confusionMatrix(data=factor(model_1$final_pred),reference = factor(model_1$X2))
```

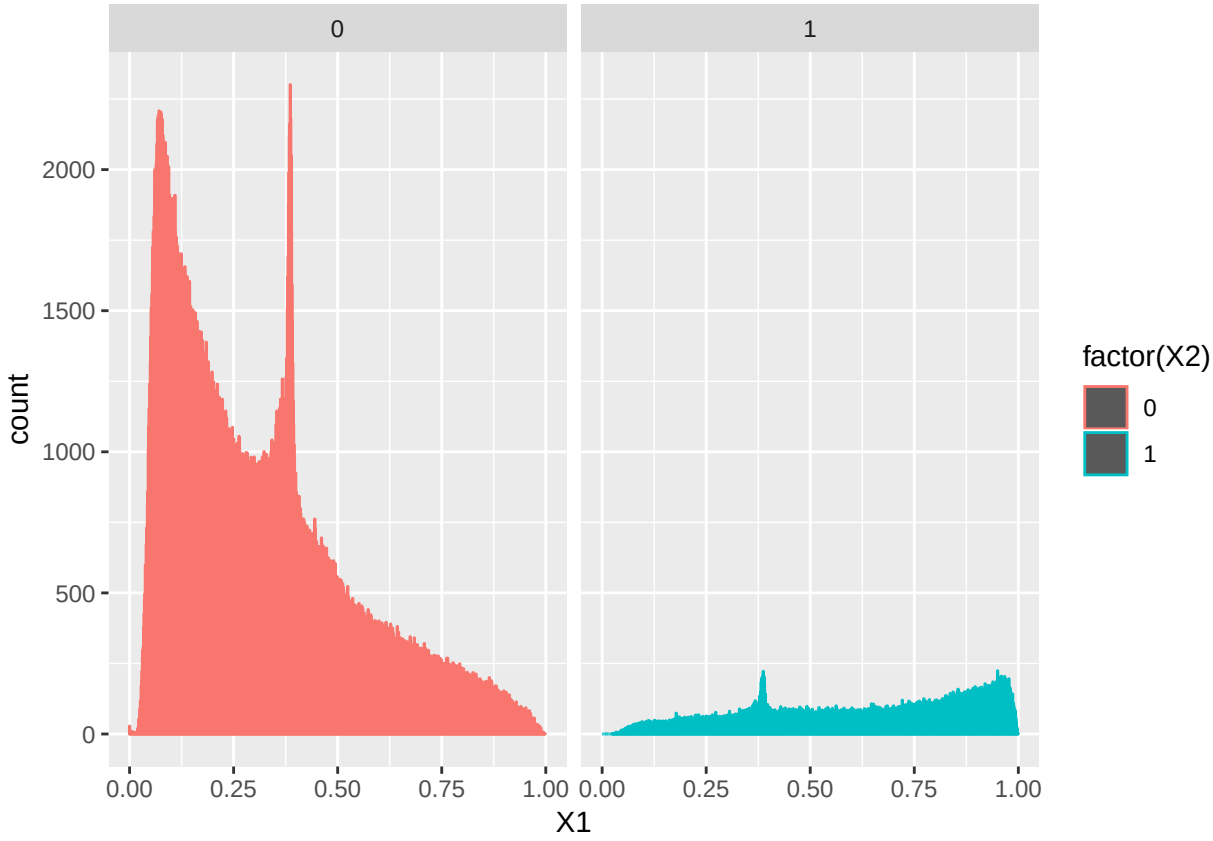
```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction      0      1
##          0 285316   8183
##          1 388532  73648
##
##                Accuracy : 0.475
```

```
##                  95% CI : (0.4739, 0.4761)
##     No Information Rate : 0.8917
##     P-Value [Acc > NIR] : 1
##
##                   Kappa : 0.1063
##
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.4234
##             Specificity : 0.9000
##          Pos Pred Value : 0.9721
##          Neg Pred Value : 0.1593
##              Prevalence : 0.8917
##          Detection Rate : 0.3776
##    Detection Prevalence : 0.3884
##       Balanced Accuracy : 0.6617
##
##        'Positive' Class : 0
##
```

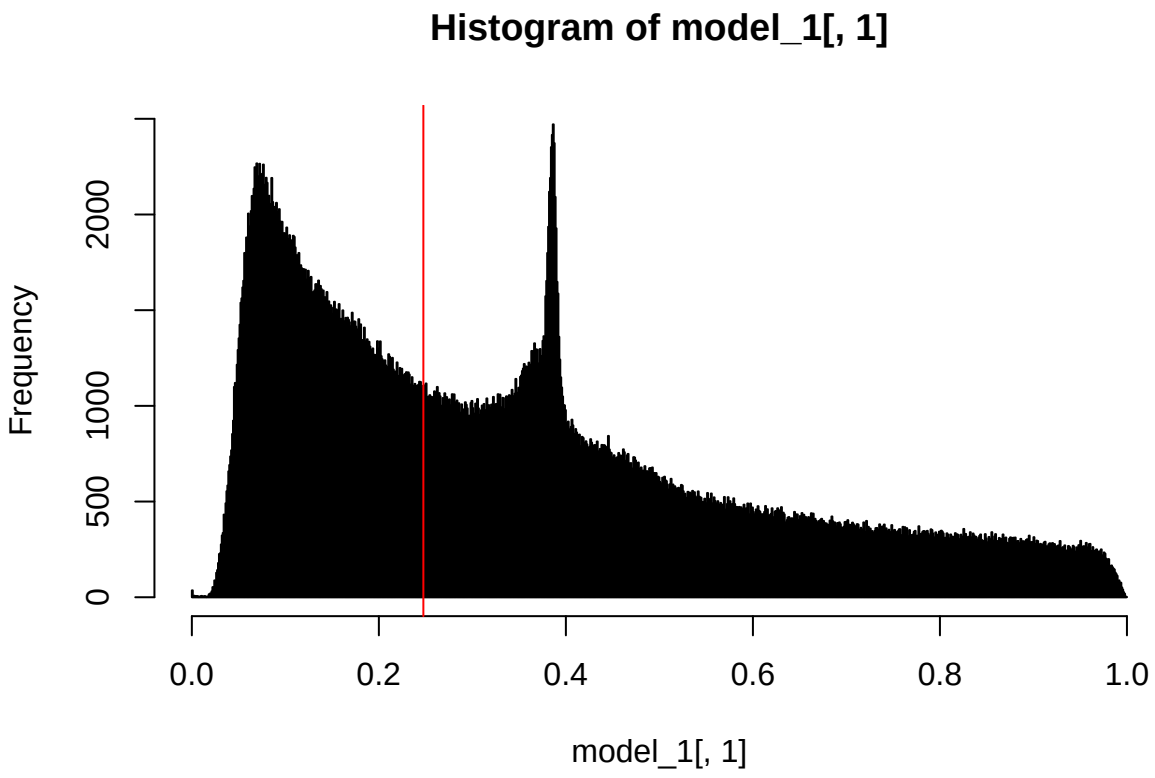
Model 4

```
rm(list=ls())

model_1.preds <- read.csv("C:/USers/gerhard/Documents/msc-hpc/round3/feedforward/local/round3_model3_re

model_1.labels <- read.csv("C:/USers/gerhard/Documents/msc-hpc/round3/feedforward/local/round3_model3_y

model_1 <- data.frame(cbind(model_1.preds[,2],model_1.labels[,2]))

model_1.electrons <- which(model_1[,2]==1)

electrons <- model_1[model_1.electrons,]

pions <- model_1[-as.numeric(model_1.electrons),]

electrons <- data.frame(electrons)

names(electrons) <- c("prediction","label")

pions <- data.frame(pions)

names(pions) <- c("prediction","label")


electron_efficiency <- function(electrons.,par){

  electrons <- electrons.

  electrons$electron_pred <- ifelse(electrons$prediction>=par[1],1,0)

  correct <- ifelse(electrons$electron_pred==electrons$label,1,0)
```

```
  error_metric <- sum(correct)/nrow(electrons)

  error_metric <- (error_metric-0.9)^2

  return(error_metric)

}

res <- optim(par=c(0),fn=electron_efficiency,lower = 0,upper = 1,electrons.=electrons,method="Brent")

require(ggplot2)

ggplot(model_1,aes(X1,colour=factor(X2)))+geom_histogram(bins = 1000)+facet_wrap(factor(model_1$X2))
```




```
hist(model_1[,1],breaks=1000)
abline(v=res$par,col="red")
```




```
electrons$predicted_label <- ifelse(electrons$prediction>=res$par,1,0)

sum(electrons$predicted_label)/nrow(electrons)
```

```
## [1] 0.8998668
```

```
pions$predicted_label <- ifelse(pions$prediction>=res$par,1,0)

pions$misclassified_as_electron <- ifelse(pions$predicted_label==1,1,0)

sum(pions$misclassified_as_electron)/nrow(pions)
```

```
## [1] 0.5304787
```

```
model_1$final_pred <- ifelse(model_1$X1>=res$par,1,0)

require(caret)

caret::confusionMatrix(data=factor(model_1$final_pred),reference = factor(model_1$X2))
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction      0      1
##          0 316386   8194
##          1 357462  73637
##
##                Accuracy : 0.5161
```

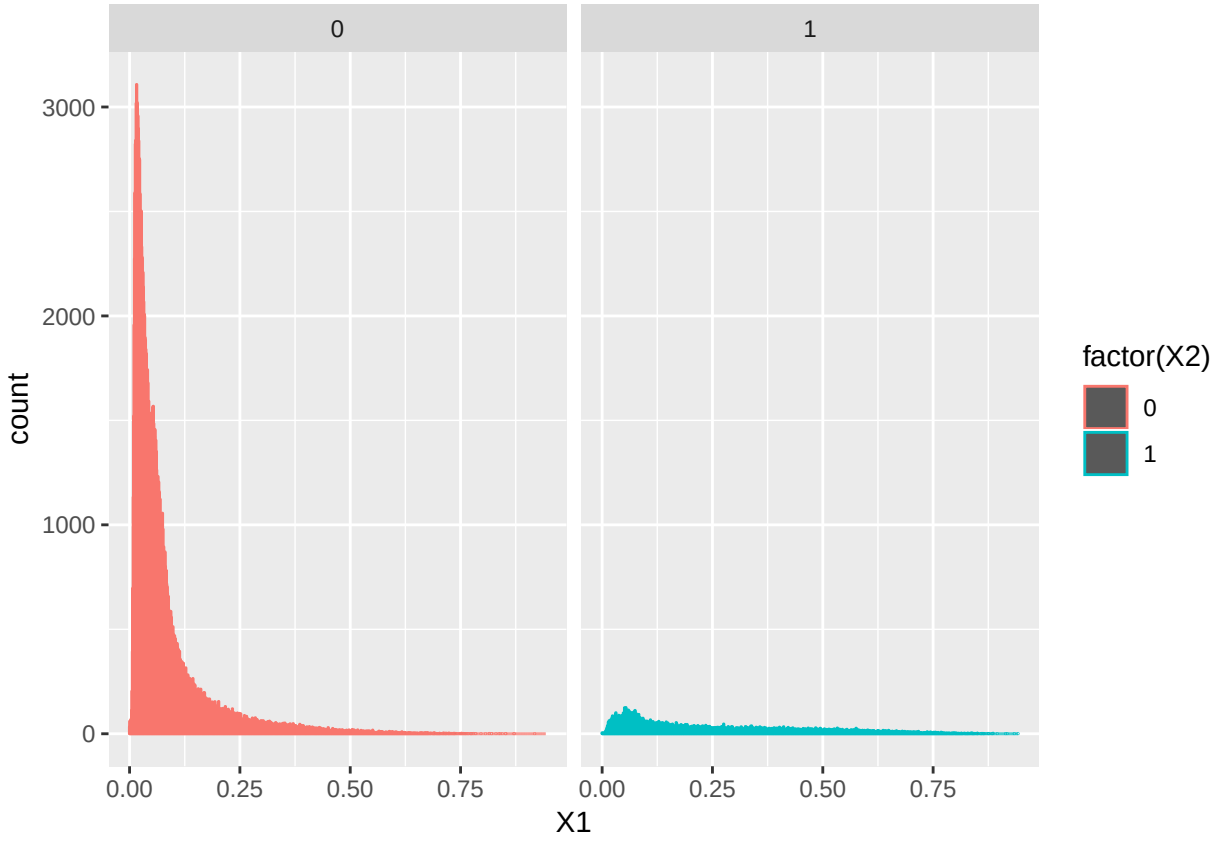
```
##                  95% CI : (0.515, 0.5172)
##     No Information Rate : 0.8917
##     P-Value [Acc > NIR] : 1
##
##                   Kappa : 0.1285
##
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.4695
##             Specificity : 0.8999
##          Pos Pred Value : 0.9748
##          Neg Pred Value : 0.1708
##              Prevalence : 0.8917
##          Detection Rate : 0.4187
##    Detection Prevalence : 0.4295
##       Balanced Accuracy : 0.6847
##
##        'Positive' Class : 0
##
```

Convolutional Neural Networks

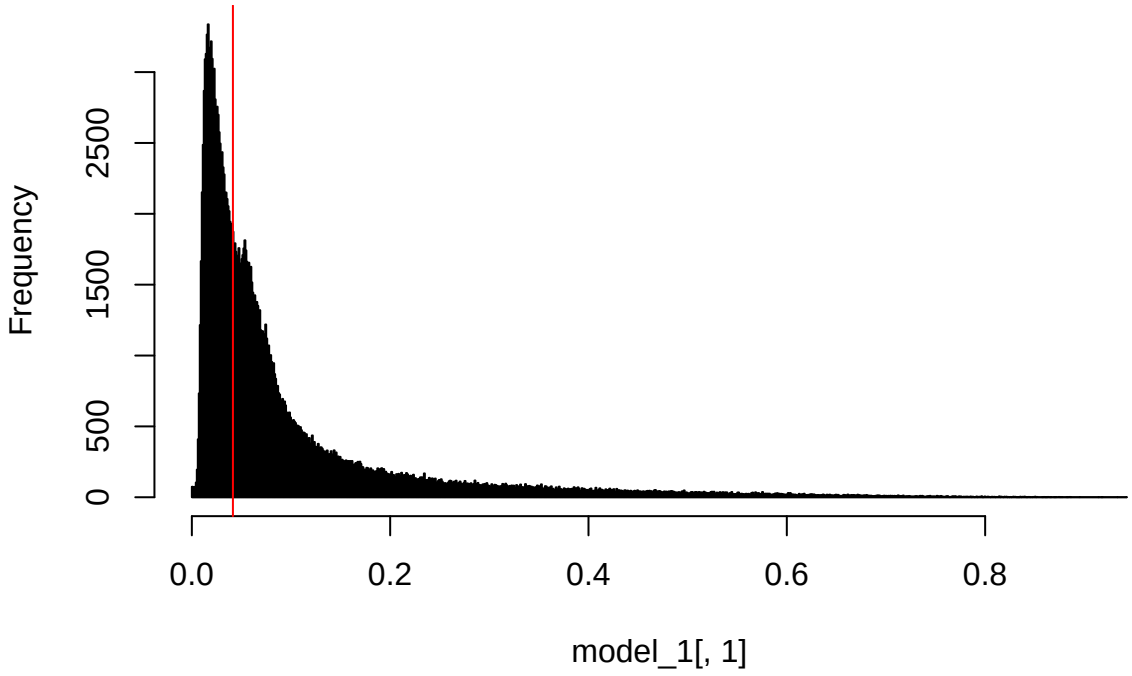
Model 1

```
rm(list=ls())

model_1.preds <- read.csv("C:/USers/gerhard/Documents/msc-hpc/round3/conv/round3_model1_results.csv",hea

model_1.labels <- read.csv("C:/USers/gerhard/Documents/msc-hpc/round3/conv/round3_model1_y_test.csv",hea

model_1 <- data.frame(cbind(model_1.preds[,2],model_1.labels[,2]))

model_1.electrons <- which(model_1[,2]==1)

electrons <- model_1[model_1.electrons,]

pions <- model_1[-as.numeric(model_1.electrons),]

electrons <- data.frame(electrons)

names(electrons) <- c("prediction","label")

pions <- data.frame(pions)

names(pions) <- c("prediction","label")


electron_efficiency <- function(electrons.,par){

  electrons <- electrons.

  electrons$electron_pred <- ifelse(electrons$prediction>=par[1],1,0)
```

```
  correct <- ifelse(electrons$electron_pred==electrons$label,1,0)

  error_metric <- sum(correct)/nrow(electrons)

  error_metric <- (error_metric-0.9)^2

  return(error_metric)
}

res <- optim(par=c(0),fn=electron_efficiency,lower = 0,upper = 1,electrons.=electrons,method="Brent")

require(ggplot2)

ggplot(model_1,aes(X1,colour=factor(X2)))+geom_histogram(bins = 1000)+facet_wrap(factor(model_1$X2))
```




```
hist(model_1[,1],breaks=1000)
abline(v=res$par,col="red")
```




```
electrons$predicted_label <- ifelse(electrons$prediction>=res$par,1,0)

sum(electrons$predicted_label)/nrow(electrons)
```

```
## [1] 0.9000086
```

```
pions$predicted_label <- ifelse(pions$prediction>=res$par,1,0)

pions$misclassified_as_electron <- ifelse(pions$predicted_label==1,1,0)

sum(pions$misclassified_as_electron)/nrow(pions)
```

```
## [1] 0.5614019
```

```
model_1$final_pred <- ifelse(model_1$X1>=res$par,1,0)

require(caret)

caret::confusionMatrix(data=factor(model_1$final_pred),reference = factor(model_1$X2))
```

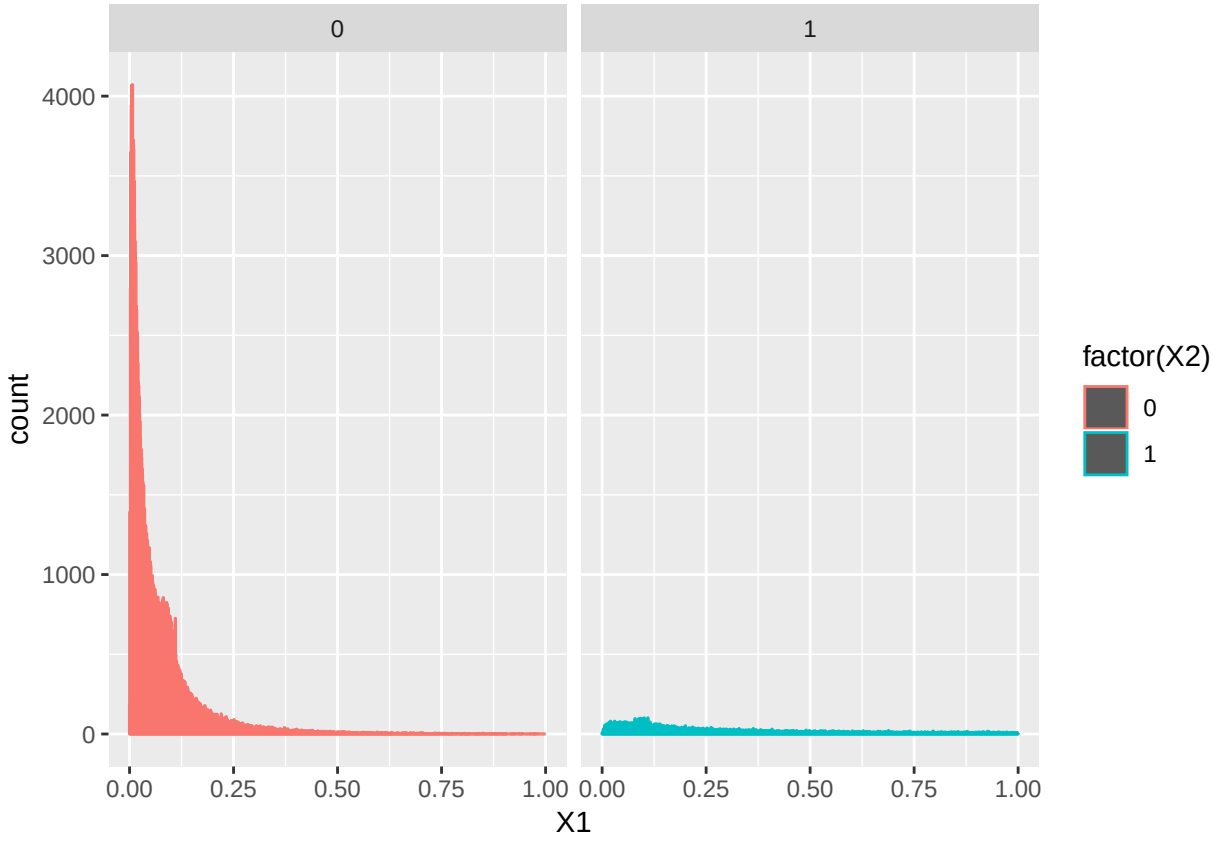
```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction      0      1
##          0  83724   2327
##          1 107166  20945
##
##                Accuracy : 0.4887
```

```
##                  95% CI : (0.4866, 0.4909)
##     No Information Rate : 0.8913
##     P-Value [Acc > NIR] : 1
##
##                   Kappa : 0.1137
##
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.4386
##             Specificity : 0.9000
##          Pos Pred Value : 0.9730
##          Neg Pred Value : 0.1635
##              Prevalence : 0.8913
##          Detection Rate : 0.3909
##    Detection Prevalence : 0.4018
##       Balanced Accuracy : 0.6693
##
##        'Positive' Class : 0
##
```

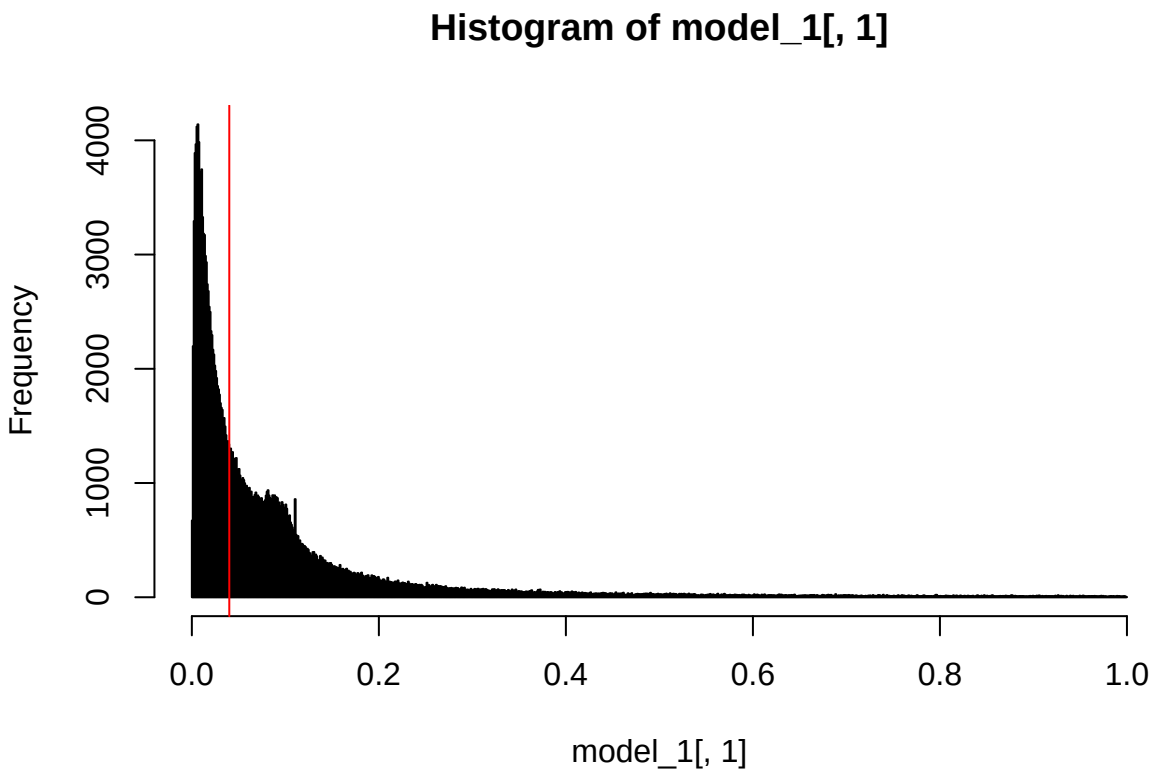
Model 2

```
rm(list=ls())

model_1.preds <- read.csv("C:/USers/gerhard/Documents/msc-hpc/round3/conv/round3_model2_results.csv",hea

model_1.labels <- read.csv("C:/USers/gerhard/Documents/msc-hpc/round3/conv/round3_model2_y_test.csv",hea

model_1 <- data.frame(cbind(model_1.preds[,2],model_1.labels[,2]))

model_1.electrons <- which(model_1[,2]==1)

electrons <- model_1[model_1.electrons,]

pions <- model_1[-as.numeric(model_1.electrons),]

electrons <- data.frame(electrons)

names(electrons) <- c("prediction","label")

pions <- data.frame(pions)

names(pions) <- c("prediction","label")


electron_efficiency <- function(electrons.,par){

  electrons <- electrons.

  electrons$electron_pred <- ifelse(electrons$prediction>=par[1],1,0)

  correct <- ifelse(electrons$electron_pred==electrons$label,1,0)
```

```
  error_metric <- sum(correct)/nrow(electrons)

  error_metric <- (error_metric-0.9)^2

  return(error_metric)

}

res <- optim(par=c(0),fn=electron_efficiency,lower = 0,upper = 1,electrons.=electrons,method="Brent")

require(ggplot2)

ggplot(model_1,aes(X1,colour=factor(X2)))+geom_histogram(bins = 1000)+facet_wrap(factor(model_1$X2))
```




```
hist(model_1[,1],breaks=1000)
abline(v=res$par,col="red")
```




```
electrons$predicted_label <- ifelse(electrons$prediction>=res$par,1,0)

sum(electrons$predicted_label)/nrow(electrons)
```

```
## [1] 0.8999227
```

```
pions$predicted_label <- ifelse(pions$prediction>=res$par,1,0)

pions$misclassified_as_electron <- ifelse(pions$predicted_label==1,1,0)

sum(pions$misclassified_as_electron)/nrow(pions)
```

```
## [1] 0.4937241
```

```
model_1$final_pred <- ifelse(model_1$X1>=res$par,1,0)

require(caret)

caret::confusionMatrix(data=factor(model_1$final_pred),reference = factor(model_1$X2))
```

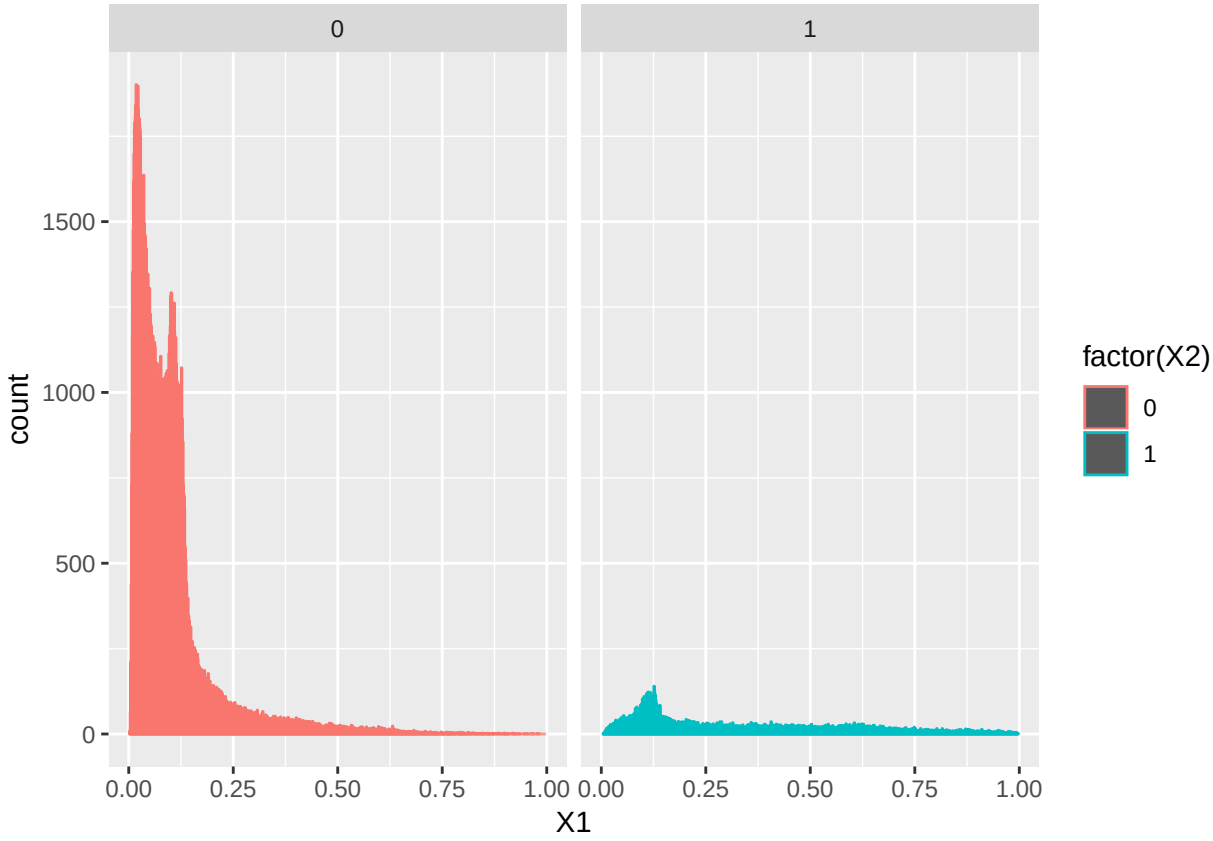
```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction     0     1
##          0 96643  2329
##          1 94247 20943
##
##                Accuracy : 0.5491
```

```
##                  95% CI : (0.5469, 0.5512)
##     No Information Rate : 0.8913
##     P-Value [Acc > NIR] : 1
##
##                   Kappa : 0.1486
##
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.5063
##             Specificity : 0.8999
##          Pos Pred Value : 0.9765
##          Neg Pred Value : 0.1818
##              Prevalence : 0.8913
##          Detection Rate : 0.4513
##    Detection Prevalence : 0.4621
##       Balanced Accuracy : 0.7031
##
##        'Positive' Class : 0
##
```

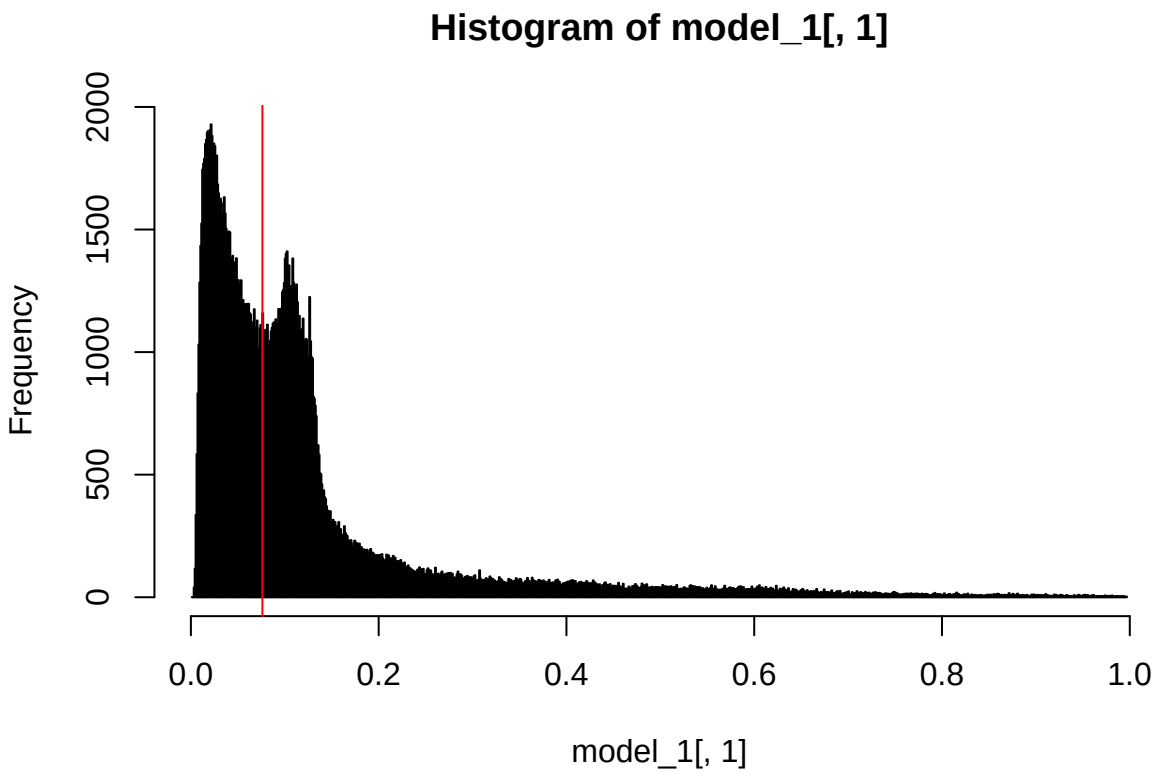
Model 3

```
rm(list=ls())

model_1.preds <- read.csv("C:/USers/gerhard/Documents/msc-hpc/round3/conv/round3_model3_results.csv",hea

model_1.labels <- read.csv("C:/USers/gerhard/Documents/msc-hpc/round3/conv/round3_model3_y_test.csv",hea

model_1 <- data.frame(cbind(model_1.preds[,2],model_1.labels[,2]))

model_1.electrons <- which(model_1[,2]==1)

electrons <- model_1[model_1.electrons,]

pions <- model_1[-as.numeric(model_1.electrons),]

electrons <- data.frame(electrons)

names(electrons) <- c("prediction","label")

pions <- data.frame(pions)

names(pions) <- c("prediction","label")


electron_efficiency <- function(electrons.,par){

  electrons <- electrons.

  electrons$electron_pred <- ifelse(electrons$prediction>=par[1],1,0)

  correct <- ifelse(electrons$electron_pred==electrons$label,1,0)
```

```
  error_metric <- sum(correct)/nrow(electrons)

  error_metric <- (error_metric-0.9)^2

  return(error_metric)

}

res <- optim(par=c(0),fn=electron_efficiency,lower = 0,upper = 1,electrons.=electrons,method="Brent")

require(ggplot2)

ggplot(model_1,aes(X1,colour=factor(X2)))+geom_histogram(bins = 1000)+facet_wrap(factor(model_1$X2))
```




```
hist(model_1[,1],breaks=1000)
abline(v=res$par,col="red")
```




```
electrons$predicted_label <- ifelse(electrons$prediction>=res$par,1,0)

sum(electrons$predicted_label)/nrow(electrons)
```

```
## [1] 0.9000945
```

```
pions$predicted_label <- ifelse(pions$prediction>=res$par,1,0)

pions$misclassified_as_electron <- ifelse(pions$predicted_label==1,1,0)

sum(pions$misclassified_as_electron)/nrow(pions)
```

```
## [1] 0.4912567
```

```
model_1$final_pred <- ifelse(model_1$X1>=res$par,1,0)

require(caret)

caret::confusionMatrix(data=factor(model_1$final_pred),reference = factor(model_1$X2))
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction     0     1
##          0 97114  2325
##          1 93776 20947
##
##                Accuracy : 0.5513
```

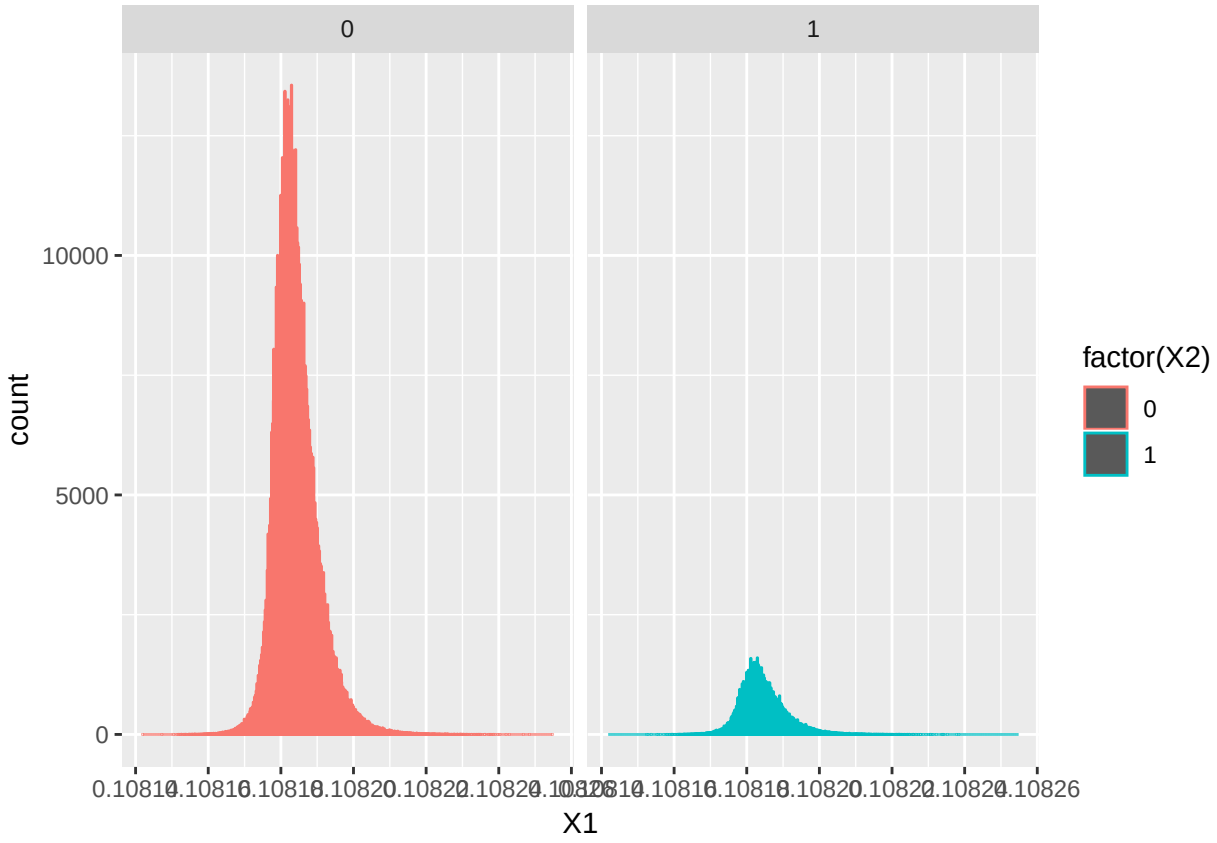
```
##                  95% CI : (0.5492, 0.5534)
##     No Information Rate : 0.8913
##     P-Value [Acc > NIR] : 1
##
##                   Kappa : 0.15
##
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.5087
##             Specificity : 0.9001
##          Pos Pred Value : 0.9766
##          Neg Pred Value : 0.1826
##              Prevalence : 0.8913
##          Detection Rate : 0.4535
##    Detection Prevalence : 0.4643
##       Balanced Accuracy : 0.7044
##
##        'Positive' Class : 0
##
```

Model 4

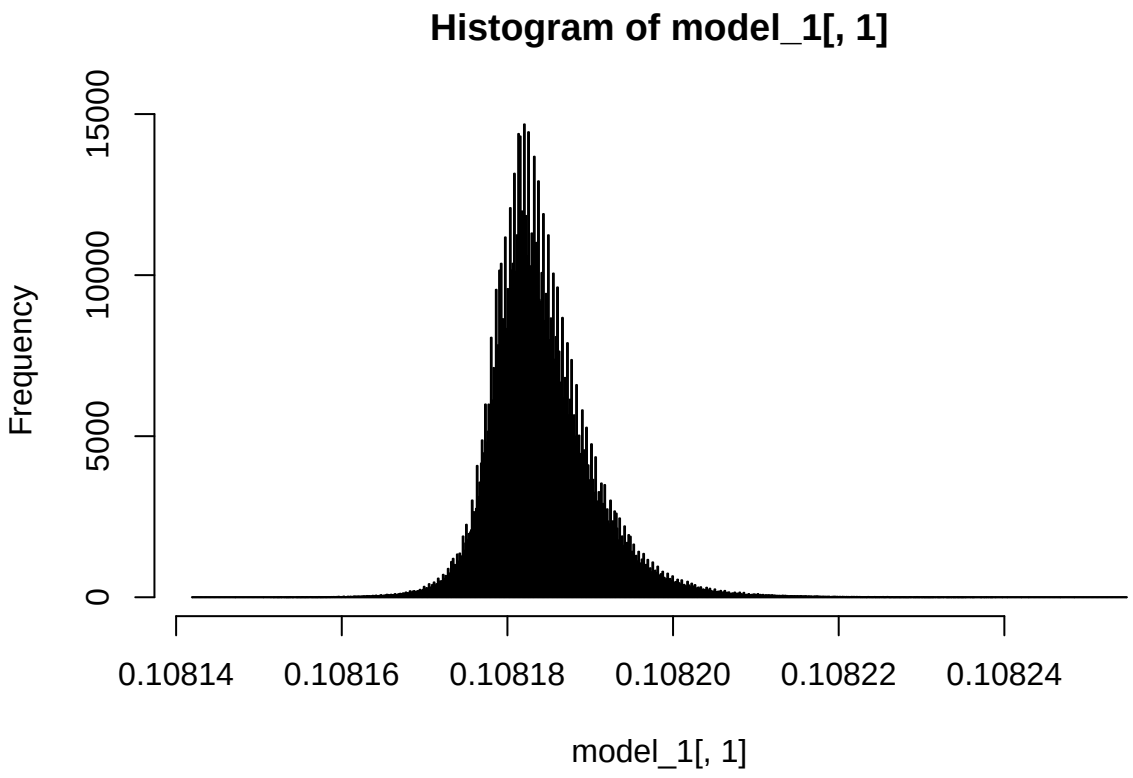
Model 5

```
rm(list=ls())

model_1.preds <- read.csv("C:/USers/gerhard/Documents/msc-hpc/round3/conv/round3_model5_results.csv",hea

model_1.labels <- read.csv("C:/USers/gerhard/Documents/msc-hpc/round3/conv/round3_model5_y_test.csv",hea

model_1 <- data.frame(cbind(model_1.preds[,2],model_1.labels[,2]))

model_1.electrons <- which(model_1[,2]==1)

electrons <- model_1[model_1.electrons,]

pions <- model_1[-as.numeric(model_1.electrons),]

electrons <- data.frame(electrons)

names(electrons) <- c("prediction","label")

pions <- data.frame(pions)

names(pions) <- c("prediction","label")


electron_efficiency <- function(electrons.,par){

  electrons <- electrons.

  electrons$electron_pred <- ifelse(electrons$prediction>=par[1],1,0)

  correct <- ifelse(electrons$electron_pred==electrons$label,1,0)
```

```
  error_metric <- sum(correct)/nrow(electrons)

  error_metric <- (error_metric-0.9)^2

  return(error_metric)

}

res <- optim(par=c(0),fn=electron_efficiency,lower = 0,upper = 1,electrons.=electrons,method="Brent")

require(ggplot2)

ggplot(model_1,aes(X1,colour=factor(X2)))+geom_histogram(bins = 1000)+facet_wrap(factor(model_1$X2))
```




```
hist(model_1[,1],breaks=1000)
abline(v=res$par,col="red")
```




```
electrons$predicted_label <- ifelse(electrons$prediction>=res$par,1,0)

sum(electrons$predicted_label)/nrow(electrons)
```

```
## [1] 0
```

```
pions$predicted_label <- ifelse(pions$prediction>=res$par,1,0)

pions$misclassified_as_electron <- ifelse(pions$predicted_label==1,1,0)

sum(pions$misclassified_as_electron)/nrow(pions)
```

```
## [1] 0
```

```
model_1$final_pred <- ifelse(model_1$X1>=res$par,1,0)

require(caret)

caret::confusionMatrix(data=factor(model_1$final_pred),reference = factor(model_1$X2))
```

```
## Warning in confusionMatrix.default(data = factor(model_1$final_pred),
## reference = factor(model_1$X2)): Levels are not in the same order for
## reference and data. Refactoring data to match.
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction       0       1
```

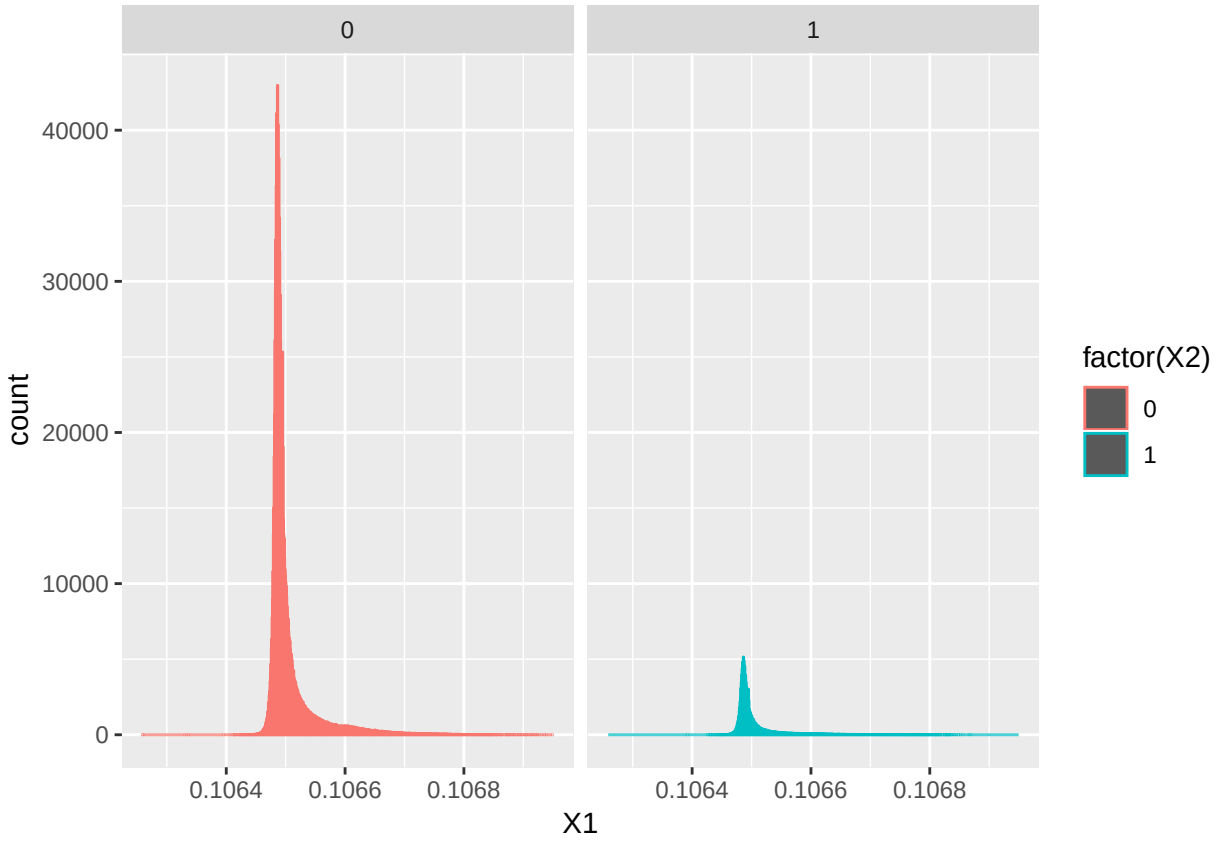
```
##          0 1182220  142775
##          1       0       0
##
##                Accuracy : 0.8922
##                  95% CI : (0.8917, 0.8928)
##     No Information Rate : 0.8922
##     P-Value [Acc > NIR] : 0.5007
##
##                   Kappa : 0
##
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 1.0000
##             Specificity : 0.0000
##          Pos Pred Value : 0.8922
##          Neg Pred Value :    NaN
##              Prevalence : 0.8922
##          Detection Rate : 0.8922
##    Detection Prevalence : 1.0000
##       Balanced Accuracy : 0.5000
##
##        'Positive' Class : 0
##
```

Recurrent Neural Networks

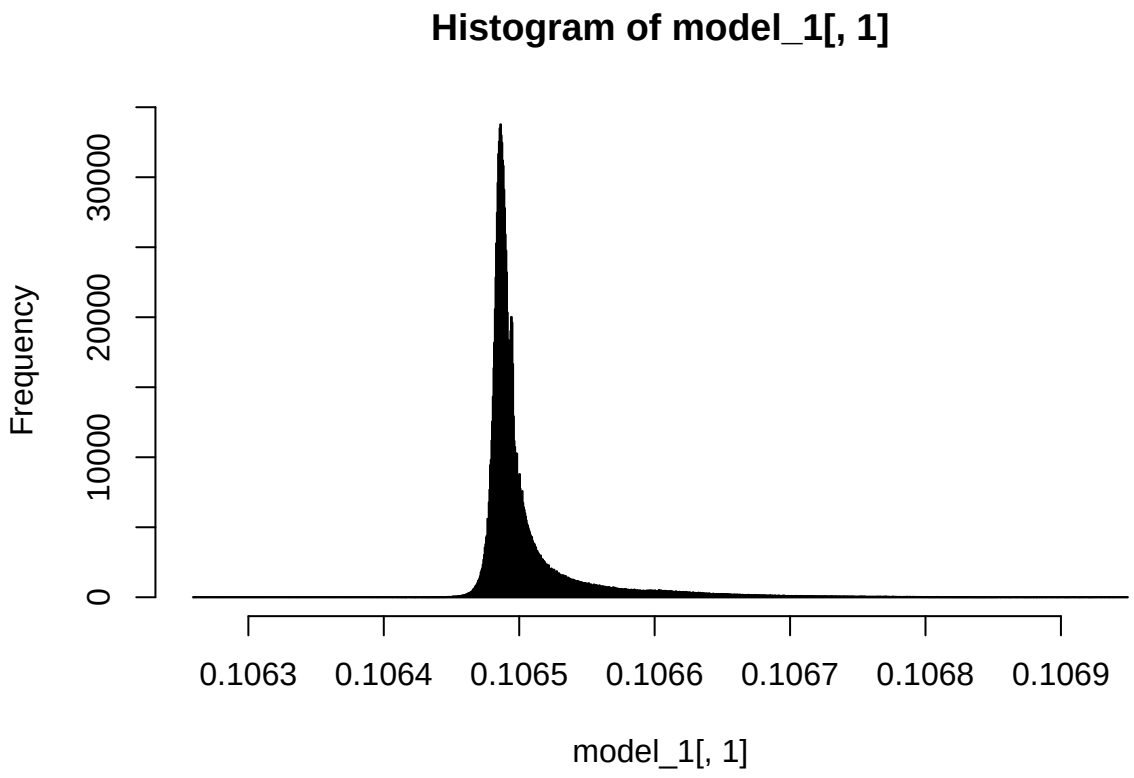
Model 1

```
rm(list=ls())

model_1.preds <- read.csv("C:/USers/gerhard/Documents/msc-hpc/round3/lstm/round3_model1_results.csv",hea

model_1.labels <- read.csv("C:/USers/gerhard/Documents/msc-hpc/round3/lstm/round3_model1_y_test.csv",hea

model_1 <- data.frame(cbind(model_1.preds[,2],model_1.labels[,2]))

model_1.electrons <- which(model_1[,2]==1)

electrons <- model_1[model_1.electrons,]

pions <- model_1[-as.numeric(model_1.electrons),]

electrons <- data.frame(electrons)

names(electrons) <- c("prediction","label")

pions <- data.frame(pions)

names(pions) <- c("prediction","label")


electron_efficiency <- function(electrons.,par){
```

```
  electrons <- electrons.

  electrons$electron_pred <- ifelse(electrons$prediction>=par[1],1,0)

  correct <- ifelse(electrons$electron_pred==electrons$label,1,0)

  error_metric <- sum(correct)/nrow(electrons)

  error_metric <- (error_metric-0.9)^2

  return(error_metric)

}

res <- optim(par=c(0),fn=electron_efficiency,lower = 0,upper = 1,electrons.=electrons,method="Brent")

require(ggplot2)

ggplot(model_1,aes(X1,colour=factor(X2)))+geom_histogram(bins = 1000)+facet_wrap(factor(model_1$X2))
```




```
hist(model_1[,1],breaks=1000)
abline(v=res$par,col="red")
```




```
electrons$predicted_label <- ifelse(electrons$prediction>=res$par,1,0)

sum(electrons$predicted_label)/nrow(electrons)
```

```
## [1] 0
```

```
pions$predicted_label <- ifelse(pions$prediction>=res$par,1,0)

pions$misclassified_as_electron <- ifelse(pions$predicted_label==1,1,0)

sum(pions$misclassified_as_electron)/nrow(pions)
```

```
## [1] 0
```

```
model_1$final_pred <- ifelse(model_1$X1>=res$par,1,0)

require(caret)

caret::confusionMatrix(data=factor(model_1$final_pred),reference = factor(model_1$X2))
```

```
## Warning in confusionMatrix.default(data = factor(model_1$final_pred),
## reference = factor(model_1$X2)): Levels are not in the same order for
## reference and data. Refactoring data to match.
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction       0       1
```

```
##          0 1182792  141008
##          1       0       0
##
##                Accuracy : 0.8935
##                  95% CI : (0.893, 0.894)
##     No Information Rate : 0.8935
##     P-Value [Acc > NIR] : 0.5007
##
##                   Kappa : 0
##
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 1.0000
##             Specificity : 0.0000
##          Pos Pred Value : 0.8935
##          Neg Pred Value :    NaN
##              Prevalence : 0.8935
##          Detection Rate : 0.8935
##    Detection Prevalence : 1.0000
##       Balanced Accuracy : 0.5000
##
##        'Positive' Class : 0
##
```

Model 2: Convolutional + Recurrent Neural Network

Gradient Boosting Machines

Model 1