naivebayes

Xingyu Wu

12/5/2021

```
data = read.csv("processed_counts.csv")
```

```
label = read.csv("annotation.csv")
```

```
label$Type[which(label$Type == "Normal")] <- 0
label$Type[which(label$Type != 0)] <- 1
```

```
library(sampling)
set.seed(6690)

train_id <- sample(label$ID, round(dim(label)[1]*0.75))

train_data <- data[data$ID %in% train_id, ]
test_data <- data[!(data$ID %in% train_id), ]

train_label <- label[data$ID %in% train_id, ]
test_label <- label[!(data$ID %in% train_id), ]
```

```
total_train = merge(train_data, train_label, by = "ID")
total_test = merge(test_data, test_label, by = "ID")
total_train = total_train[, -1]
total_test = total_test[, -1]
```

```
library(ggplot2)
library(lattice)
library(caret)
```

```
##
## Attaching package: 'caret'
```

```
## The following object is masked from 'package:sampling':
##
##     cluster
```

```
total_train$Type = factor(total_train$Type)
total_test$Type = factor(total_test$Type)
control <- trainControl(method = 'repeatedcv', number = 10, repeats = 2)
model <- train(Type~., total_train,
               method = 'naive_bayes',
               preProcess = c('center', 'scale'),
               trControl = control)
model
```

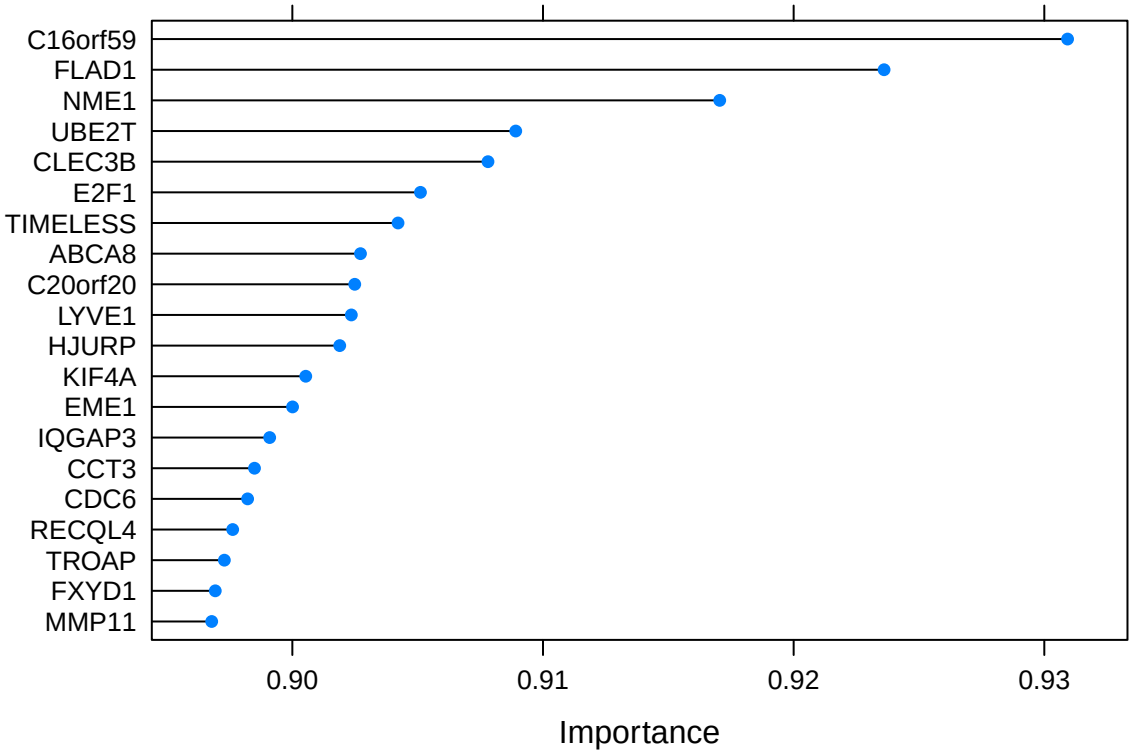
```
## Naive Bayes
##
```

```
## 5730 samples
## 2916 predictors
##    2 classes: '0', '1'
##
## Pre-processing: centered (2916), scaled (2916)
## Resampling: Cross-Validated (10 fold, repeated 2 times)
## Summary of sample sizes: 5157, 5156, 5157, 5157, 5158, 5157, ...
## Resampling results across tuning parameters:
##
##   usekernel  Accuracy   Kappa
##   FALSE      0.9137039  0.5859727
##    TRUE      0.9456399  0.6886613
##
## Tuning parameter 'laplace' was held constant at a value of 0
## Tuning
##  parameter 'adjust' was held constant at a value of 1
## Accuracy was used to select the optimal model using the largest value.
## The final values used for the model were laplace = 0, usekernel = TRUE
##  and adjust = 1.
```

```
truth <- total_test$Type
pred <- predict(model, newdata = total_test)
cm <- confusionMatrix(table(pred, truth))
cm
```

```
## Confusion Matrix and Statistics
##
##     truth
## pred    0    1
##    0  145  103
##    1   13 1649
##
##                Accuracy : 0.9393
##                  95% CI : (0.9276, 0.9496)
##     No Information Rate : 0.9173
##     P-Value [Acc > NIR] : 0.0001651
##
##                   Kappa : 0.6822
##
##  Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.91772
##             Specificity : 0.94121
##          Pos Pred Value : 0.58468
##          Neg Pred Value : 0.99218
##              Prevalence : 0.08272
##          Detection Rate : 0.07592
##    Detection Prevalence : 0.12984
##       Balanced Accuracy : 0.92947
##
##        'Positive' Class : 0
##
```

```
importance <- varImp(model, scale = FALSE)
plot(importance, top = 20)
```



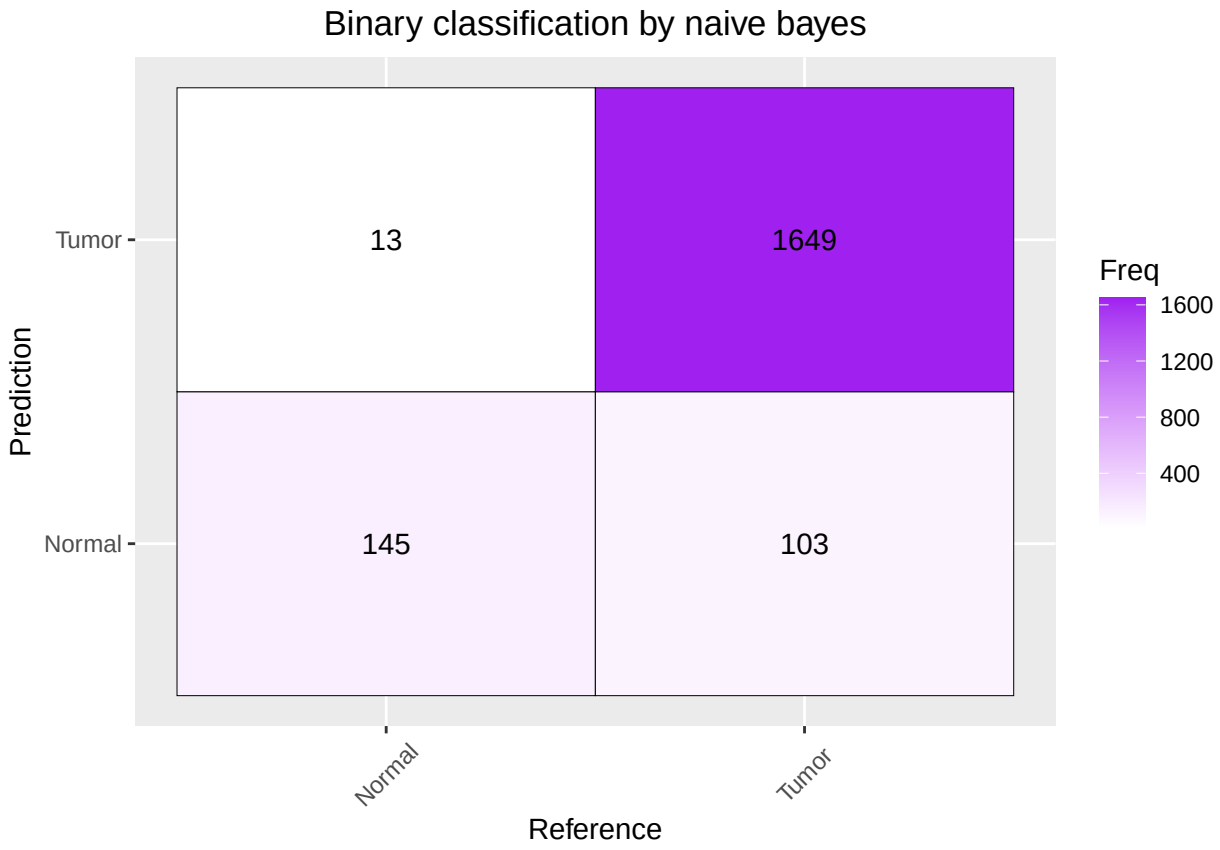

```
library(ggplot2)
library(dplyr)
```

```
##
## 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
```

```
table <- data.frame(cm$table)
q <- ggplot(table, aes(truth, pred, fill= Freq)) +
geom_tile(aes(fill = Freq), colour = "black") +
geom_text(aes(label=Freq)) +
scale_fill_gradient(low="white", high="purple") +
labs(x = "Reference",y = "Prediction") +
scale_x_discrete(labels = c("Normal", "Tumor")) +
scale_y_discrete(labels = c("Normal", "Tumor")) +
ggtitle("Binary classification by naive bayes") +
theme(plot.title = element_text(hjust = 0.5),
axis.text.x = element_text(angle = 45, vjust = 0.5, hjust = 0.5))
q
```




```
library(mltools)
mcc <- mcc(pred, truth)
mcc
```

```
## [1] 0.703903
```