Appendix B.3: Classification - Random forest

1.Preparation

loading library

```
library(randomForest)
```

```
## randomForest 4.6-12
```

```
## Type rfNews() to see new features/changes/bug fixes.
```

```
library(ggplot2)
```

```
##
## Attaching package: 'ggplot2'
```

```
## The following object is masked from 'package:randomForest':
##
##     margin
```

```
library(plotly)
```

```
##
## Attaching package: 'plotly'
```

```
## The following object is masked from 'package:ggplot2':
##
##     last_plot
```

```
## The following object is masked from 'package:stats':
##
##     filter
## The following object is masked from 'package:graphics':
## 
##     layout
```

```
library(caret)
```

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

read dataset

modify all column except for Age into Factor

```
for(i in 1:length(balanced_injury))
      balanced_injury[,i] <- as.factor(balanced_injury[,i])
```

Subset training and test datasets

```
smp_size <- floor(0.75 * nrow(balanced_injury))

set.seed(123)
train_ind <- sample(seq_len(nrow(balanced_injury)), size = smp_size, replace = FALSE)
train <- balanced_injury[train_ind, ]
test <- balanced_injury[-train_ind, ]
```

2.Build Model

Train the model

```
formular = CauseRecode39~  Education2003Revision +
                          Sex +
                          AgeRecode12 +
                          MaritalStatus +
                          DayOfWeekOfDeath +
                          RaceRecode5+
                          MonthOfDeath

set.seed(123)
RF <- randomForest(formular, data=train, ntree = 50, mtry = 7)
```

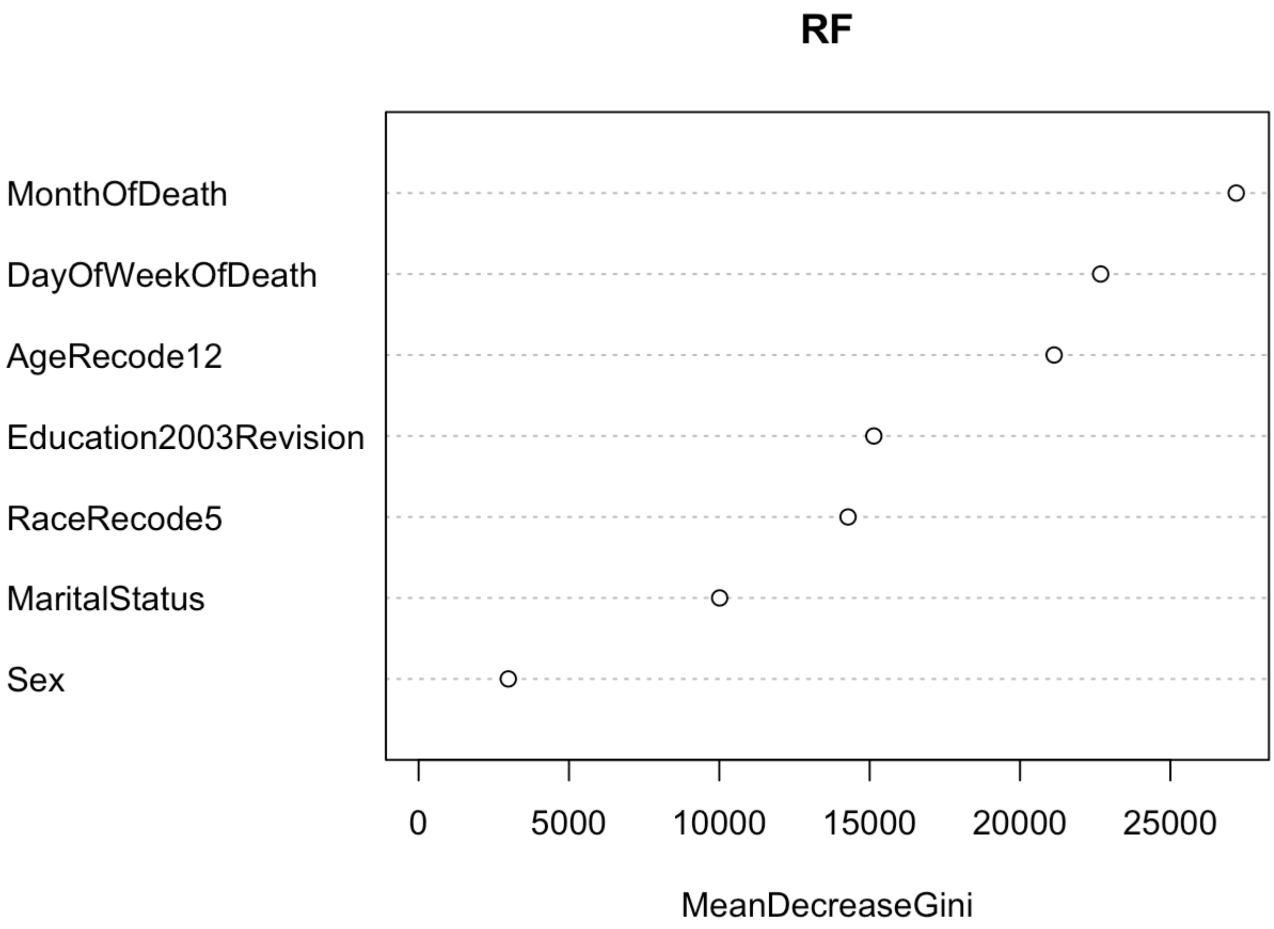
Visualization of model

```
print(RF)
```

```
##
## Call:
##  randomForest(formula = formular, data = train, ntree = 50, mtry = 7)
##                Type of random forest: classification
##                      Number of trees: 50
## No. of variables tried at each split: 7
##
##         OOB estimate of  error rate: 45.09%
## Confusion matrix:
##       38    39    40    41    42 class.error
## 38 19671  8481 12341  7720  5237   0.6319738
## 39 11006 31335 17115  7768  7333   0.5797175
## 40  9400  9445 31251  6709  7389   0.5131788
## 41  4235  2739  5393 42662  3911   0.2761792
## 42  1315  1185  2648  1845 37336   0.1577523
```

importance of variable

```
varImpPlot(RF)
```



```
RF$importance
```

```
##                       MeanDecreaseGini
## Education2003Revision        15138.921
## Sex                           2982.694
## AgeRecode12                  21134.057
## MaritalStatus                10011.588
## DayOfWeekOfDeath             22682.037
## RaceRecode5                  14281.200
## MonthOfDeath                 27190.747
```

3.Prediction

```
RFpred <- predict(RF, test)
```

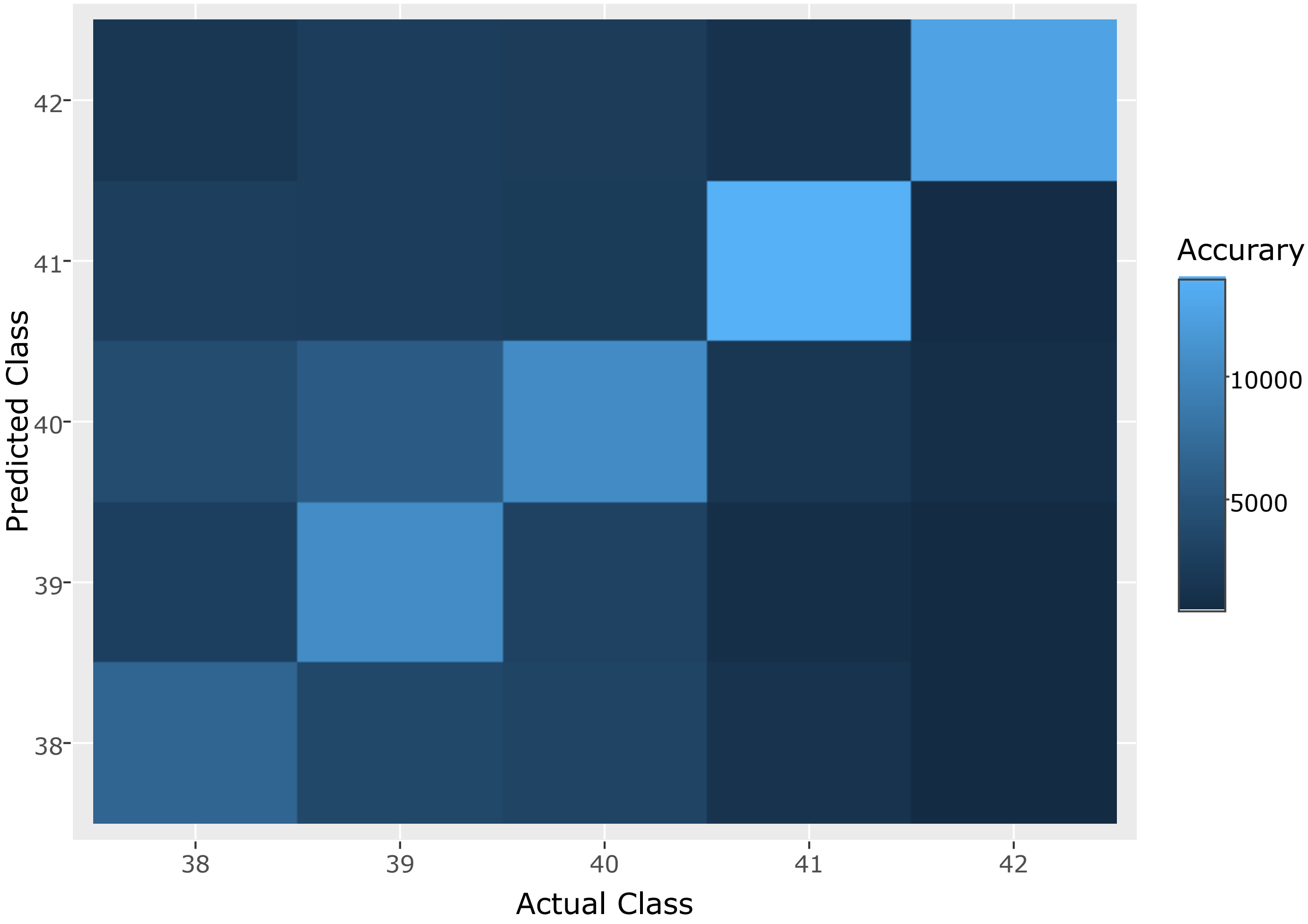
Prediction Accuracy

```
col_n <- grep('CauseRecode39', colnames(train))

confusion <- as.data.frame(table(test[ ,col_n], RFpred))
colnames(confusion) <- c('Actual','Predict', 'Freq')

plot <- ggplot(confusion) +
  geom_tile(aes(x=Actual, y=Predict, fill=Freq))+
  scale_x_discrete(name="Actual Class") +
  scale_y_discrete(name="Predicted Class") +
  labs(fill="Accurary")

ggplotly(plot)
```


Predicted Class
Actual Class
Accurary