ML Project

Suzana Iacob

21/11/2019

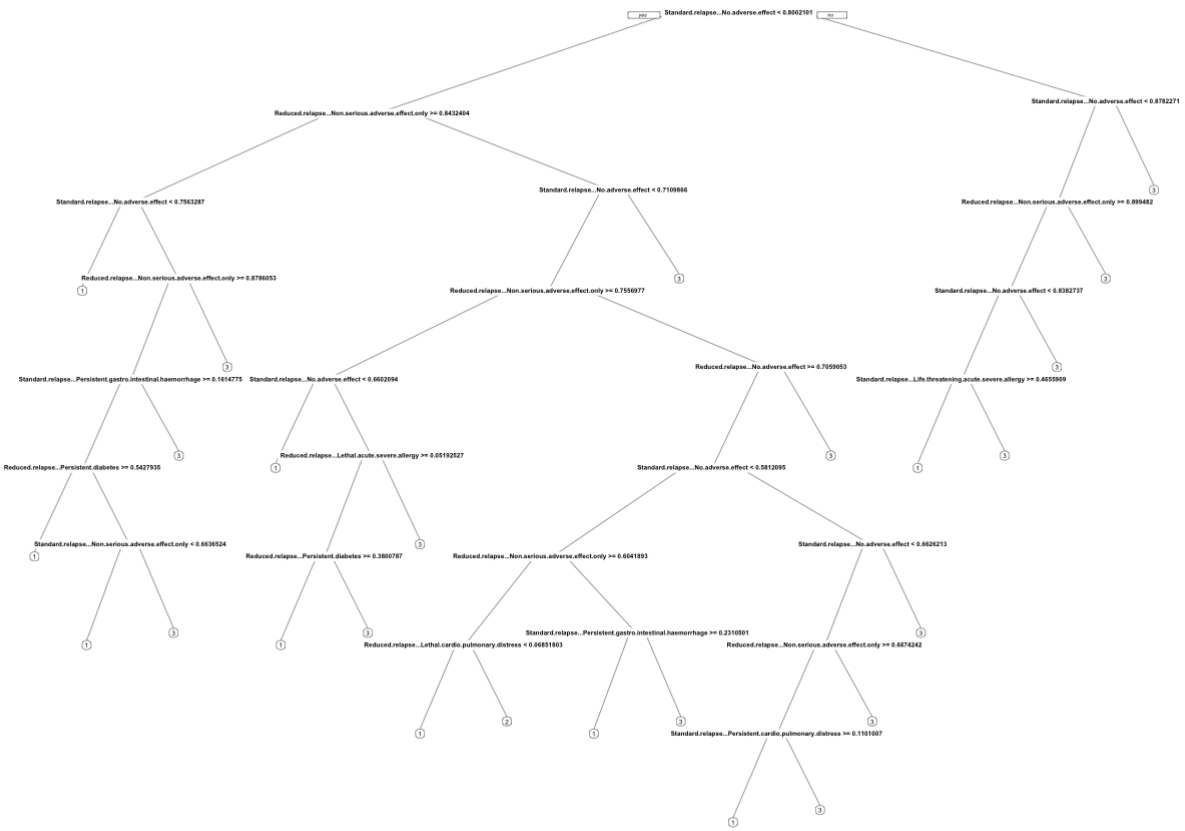
Import Processed Data

```
utilities_train <- read.csv("utilities_s1_d1_train.csv")
utilities_test <- read.csv("utilities_s1_d1_test.csv")
global_utilities <- rbind(utilities_train,utilities_test )
utilities_train$best_treatment = as.factor(utilities_train$best_treatment)
utilities_test$best_treatment = as.factor(utilities_test$best_treatment)
table(utilities_train$best_treatment)
```

```
##
##    1    2    3
## 1660  881 4459
```

Model

```
utilities_tree = rpart(best_treatment ~., data=utilities_train,cp=0.002, minbucket=10
, method="class")
prp(utilities_tree, digits = 0, varlen = 0, faclen = 0)
```



```
prediction = predict(utilities_tree, newdata = utilities_test, type="class")
matrix = table(utilities_test$best_treatment, prediction)
matrix
```

```
##    prediction
##        1    2    3
##   1  354    3  355
##   2  151    1  225
##   3  168    1 1742
```

Prediction accuracy

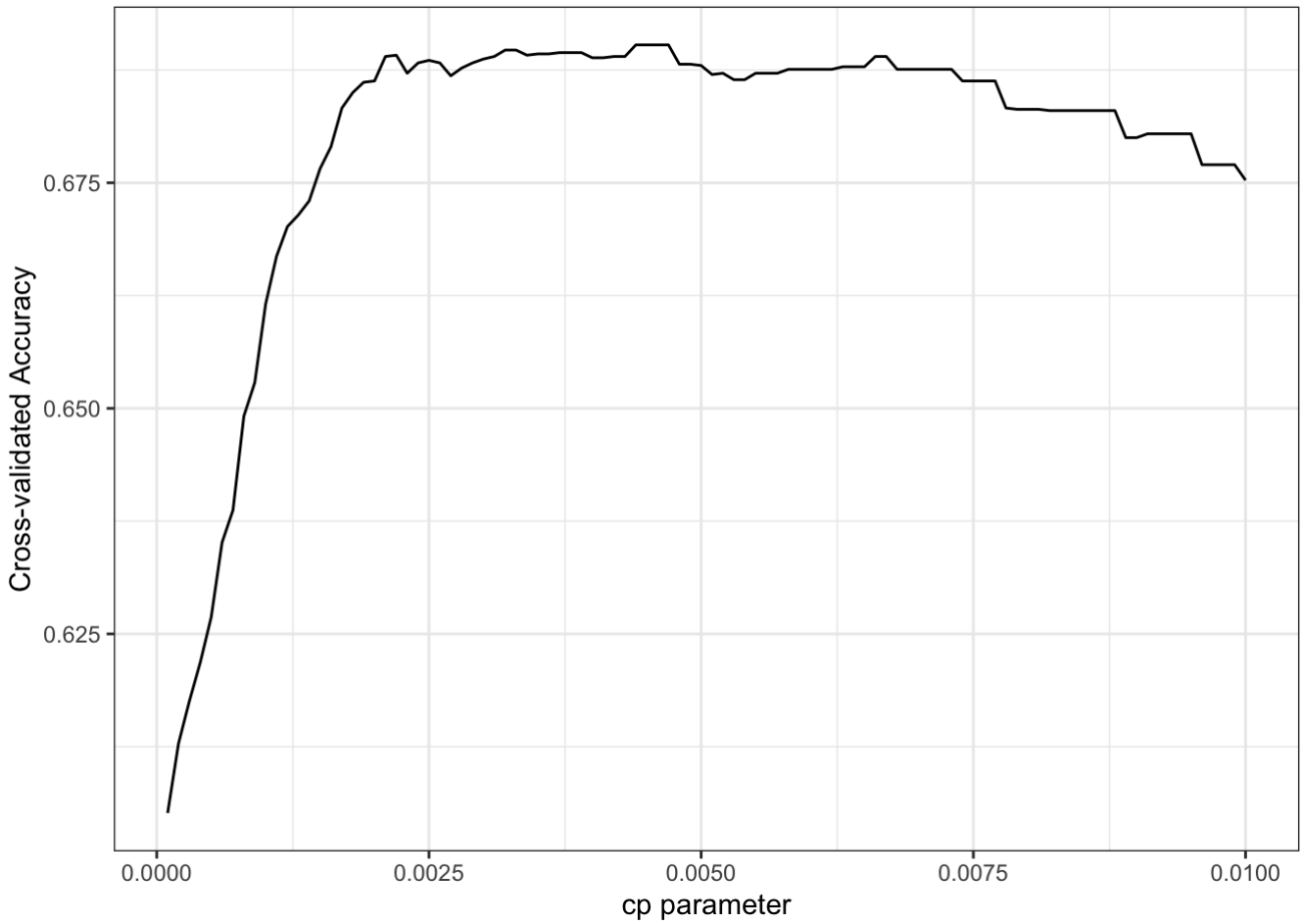
```
# First tree
print((matrix[1,1]+matrix[2,2]+matrix[3,3])/nrow(utilities_test))
```

```
## [1] 0.699
```

Cross Validation

```
PenaltyMatrix = matrix(c(0,1,1.5,2,0,2,1,2,0), byrow=TRUE, nrow=3)
cpVals <- data.frame(.cp = seq(.0001, .01, by=.0001))
set.seed(123)
cpCV = train(best_treatment~.,
             trControl=trainControl(method="cv",number=10), data=utilities_train,meth
od="rpart",minbucket=35,
             tuneGrid=cpVals, metric="Accuracy", maximize=TRUE,parms=list(loss=Penalt
yMatrix))
```

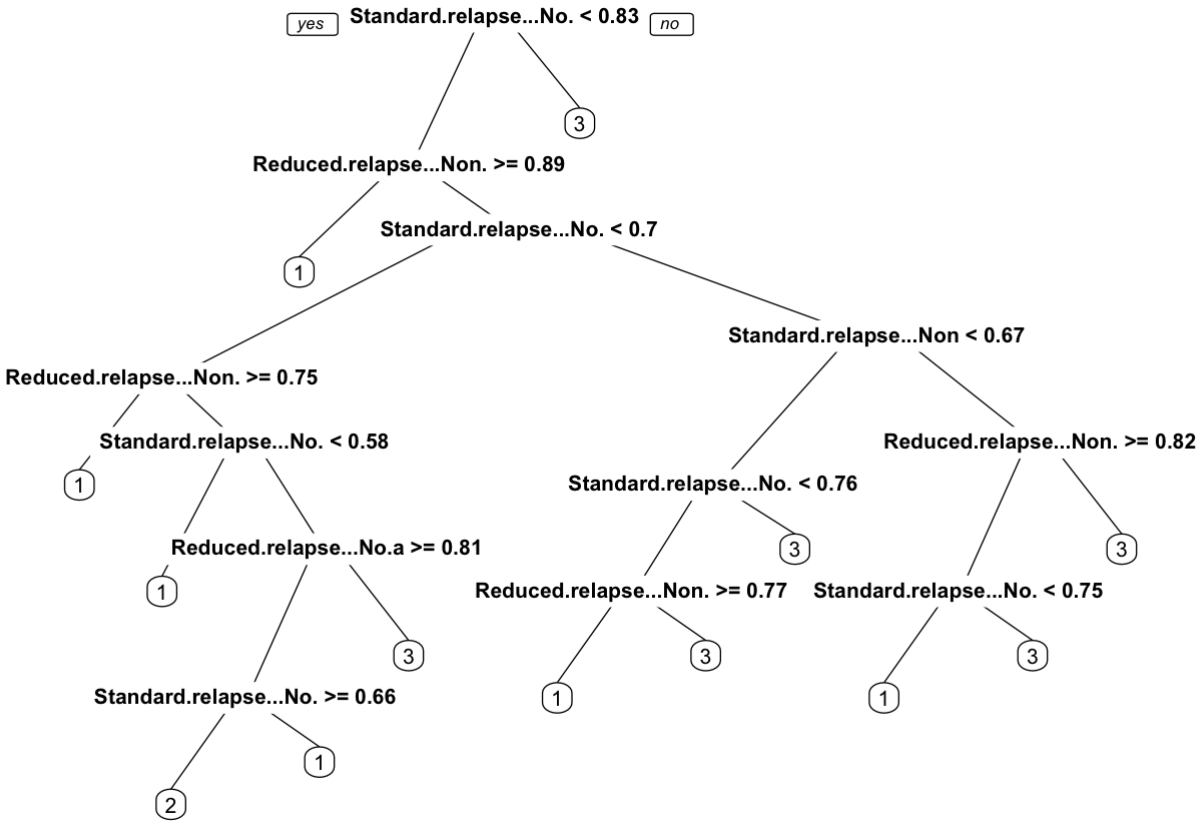
```
ggplot(cpCV$results, aes(x=cp, y=Accuracy)) +
  geom_line() +
  theme_bw() +
  xlab("cp parameter") +
  ylab("Cross-validated Accuracy")
```




```
best.cp = cpCV$bestTune
print(best.cp)
```

```
##        cp
## 47 0.0047
```

```
utilities_tree_cv <- rpart(best_treatment ~ ., data=utilities_test, minbucket = 35, c
p=0.0025, parms=list(loss=PenaltyMatrix))
prp(utilities_tree_cv)
```




```
prediction = predict(utilities_tree_cv, newdata = utilities_test, type="class")
matrix_cv = table(utilities_test$best_treatment, prediction)
matrix_cv
```

```
##    prediction
##        1    2    3
##   1  432   12  268
##   2  191   18  168
##   3  207   14 1690
```

```
# First tree
print((matrix_cv[1,1]+matrix_cv[2,2]+matrix_cv[3,3])/nrow(utilities_test))
```

```
## [1] 0.7133333
```