Assignment 5

NAME: Enmin Zhou
DUE DATE: April 13th, 11:59pm

Problem 1 (100 pts)

In the folder Assignment 3, you will find the data set called FF_wave6_2020v2.dta. This data set is from the Fragile Family Data Set, and it includes many different variables (socio-demographic, economics, and health status) of teenagers (15 years old) and their parents. The codebook (ff_wave6_codebook.txt) associated with the data set is on Canvas (folder Assignment 3). As done in assignment 3, Consider the variable *doctor diagnosed youth with depression/anxiety*. In the data set, the name of this variable is *p6b5*. Then consider in the data set these variables: *p6b10*, *p6b35*, *p6b55*, *p6b60*, *p6c21*, *p6f32*, *p6f35*, *p6h74*, *p6h102*, *p6i7*, *p6i8*, *p6i11*, *p6j37*, *k6b21a*, *k6b22a*, *k6c1*, *k6c4e*, *k6c28*, *k6d37*, *k6f63*, *ck6cbmi*, *k6d10*. Now, you have a data set with 4898 subjects and 23 variables. Clean the data in these three steps. 1- Each variable has a value with a number and a text (for example, a value for the variable *p6b5* is *2 No*). Remove the text from all the variables in the data set (hint: use the function sub for each column). 2- Transform each variable in numeric (hint: use the function as.numeric for each column). 3- Transform all the values less than 0 in NA and then remove all your NA values from the data set. Now call the variables with an appropriate name (for example *p6b5* can become *Depression*). This will be exactly the same process done in Assignment 3.

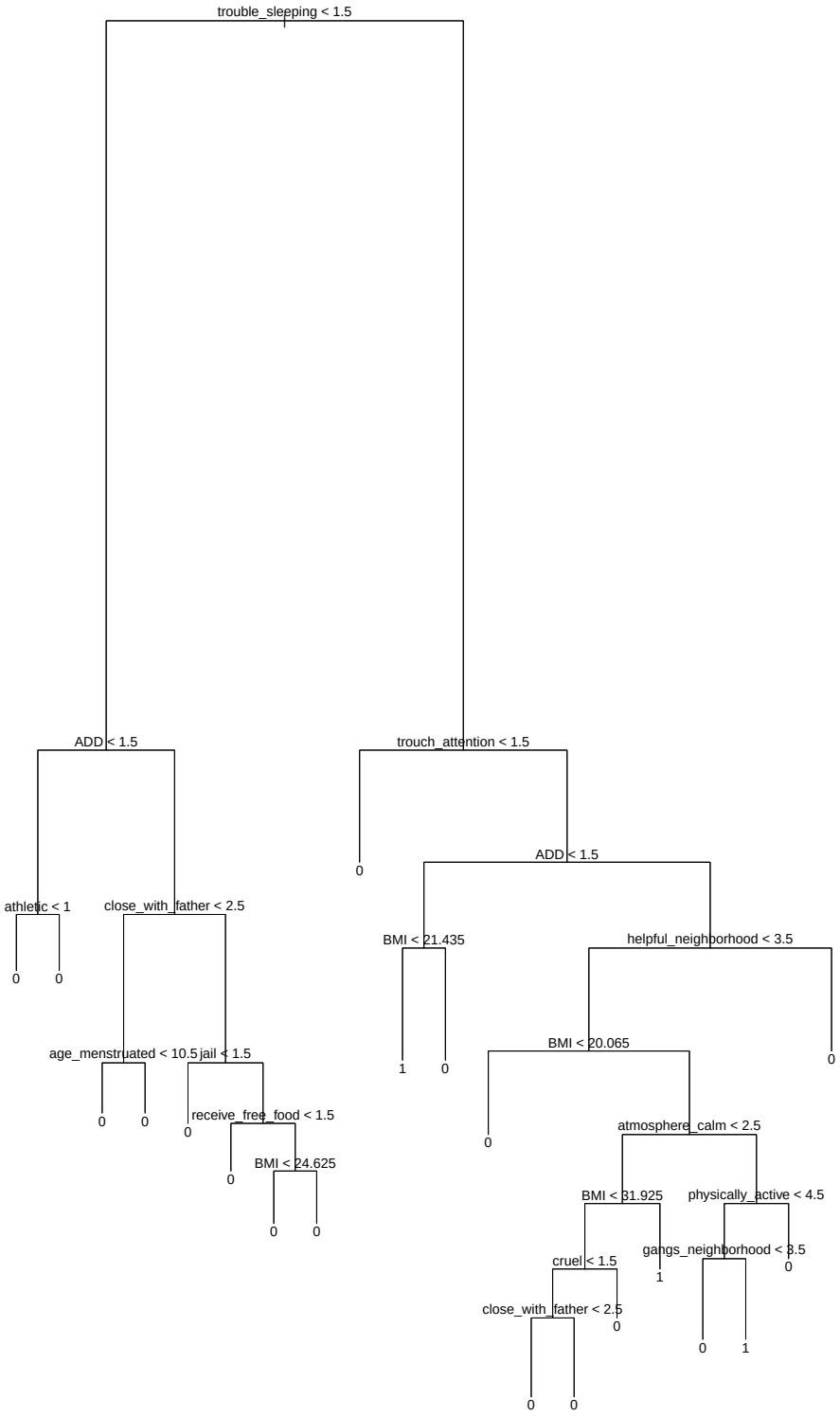
(a) (30 points) Perform the classification tree by using *Depression* as the outcome variable. Read the tree from plot() and reproduce it using ggparty (a function to plot the tree in ggplot), merge the useless branches at the same time (here is the tutorial: https://cran.r-project.org/web/packages/ggparty/vignettes/ggparty-graphic-partying.html).

```
vars <- c('p6b5','p6b10', 'p6b35', 'p6b55', 'p6b60', 'p6c21', 'p6f32', 'p6f35', 'p6h74', 'p6h102', 'p6i
'p6i11', 'p6j37', 'k6b21a', 'k6b22a', 'k6c1', 'k6c4e', 'k6c28', 'k6d37', 'k6f63', 'ck6cbmi', 'k6d10')
data = read_dta('/home/enminz/Graduate/Data-2020/Assignment5/FF_wave6_2020v2.dta', col_select = vars)
```

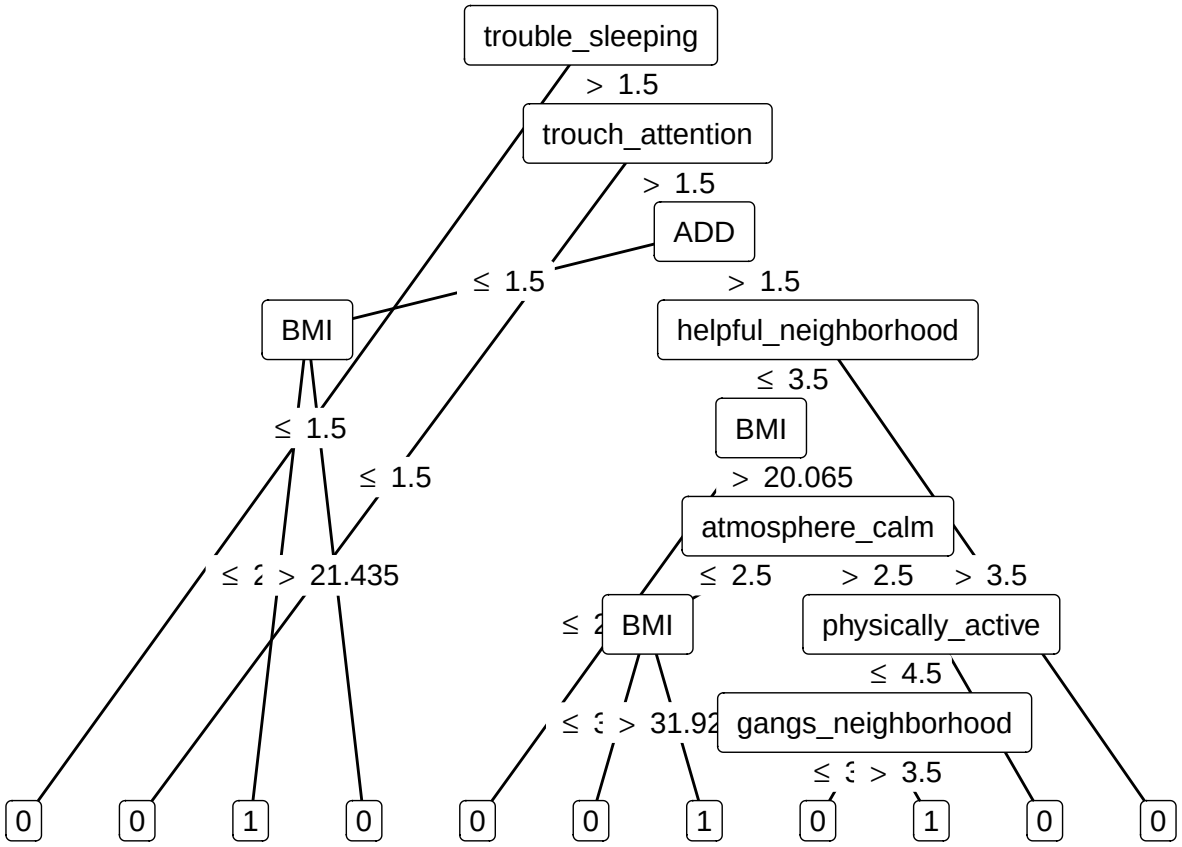
```
## Note: Using an external vector in selections is ambiguous.
## i Use `all_of(vars)` instead of `vars` to silence this message.
## i See <https://tidyselect.r-lib.org/reference/faq-external-vector.html>.
## This message is displayed once per session.
```

```
cols = c(1:23)
data[,cols] = apply(data[,cols], 2, function(x) (x));
data[,cols][data[,cols]<0] <- NA
df <- na.omit(data)
df <- df %>% rename(Depression=p6b5, ADD=p6b10,
                cruel=p6b35,  trouble_sleeping=p6b55,
                run_away=p6b60, suspend=p6c21,
                drug=p6f32, parent_jail=p6f35, smoke=p6h74, jail=p6h102, helpful_neighborhood=p6i7, clos
                gangs_neighborhood=p6i11, receive_free_food=p6j37, trouch_attention=k6b21a, athletic=k6
                atmosphere_calm=k6c4e, close_with_father=k6c28, age_menstruated=k6d10, physically_activ
                BMI=ck6cbmi
                )
df <- df %>% mutate(Depression = ifelse(Depression == 2, 0, 1), Depression = as.factor(Depression))
```

```r
tree_class <- tree(Depression~., df, method="class")
plot(tree_class)
text(tree_class)
```


trouble_sleeping < 1.5
ADD < 1.5
trouch_attention < 1.5
athletic < 1
close_with_father < 2.5
0
ADD < 1.5
0
0
BMI < 21.435
helpful_neighborhood < 3.5
age_menstruated < 10.5
jail < 1.5
1
0
BMI < 20.065
0
0
0
receive_free_food < 1.5
atmosphere_calm < 2.5
0
BMI < 24.625
BMI < 31.925
physically_active < 4.5
0
0
0
cruel < 1.5
gangs_neighborhood < 3.5
1
0
close_with_father < 2.5
0
0
1
0
0

```
sp_sleep <- partysplit(4L, breaks=1.5)
sp_trouch_attention <- partysplit(15L, breaks = 1.5)
sp_ADD <- partysplit(2L, breaks=1.5)
sp_BMI <- partysplit(23L, breaks=21.435)
sp_helpful_neighborhood <- partysplit(11L, breaks=3.5)
sp_BMI2 <- partysplit(23L, breaks=20.065)
sp_atmosphere_calm <- partysplit(18L, breaks=2.5)
sp_BMI_3 <- partysplit(23L, breaks=31.925)
sp_physically_active <- partysplit(21L, breaks=4.5)
sp_gangs_neighborhood <- partysplit(13L, breaks=3.5)
pn <- partynode(4L, split = sp_sleep, kids = list(
  partynode(4L, info="0"),
  partynode(15L, split=sp_trouch_attention, kids=list(
    partynode(15L, info="0"),
    partynode(2L, split=sp_ADD, kids=list(
      partynode(23L, split=sp_BMI, kids=list(
        partynode(23L, info="1"),
        partynode(23L, info="0")
      )),
      partynode(11L, split=sp_helpful_neighborhood, kids=list(
        partynode(23L, split=sp_BMI2, kids=list(
          partynode(23L, info="0"),
          partynode(18L, split=sp_atmosphere_calm, kids=list(
            partynode(23L, split=sp_BMI_3, kids=list(
              partynode(23L, info="0"),
              partynode(23L, info="1")
            )),
            partynode(21L, split=sp_physically_active, kids=list(
              partynode(13L, split=sp_gangs_neighborhood, kids=list(
                partynode(13L, info="0"),
                partynode(13L, info="1")
              )),
              partynode(21L, info="0")
            ))
          ))
        )),
        partynode(11L, info="0")
      ))
    ))
  ))
))
py <- party(pn, df)
ggparty(py)+geom_edge() +
  geom_edge_label() +
  geom_node_label(aes(label = splitvar), ids = "inner") +
  geom_node_splitvar() +
  geom_node_label(aes(label = info), ids = "terminal")
```



(b) (40 points) Split the data set in half, perform the tree with the training set with half of the sample. Then, with the remaining half perform the prediction and calculate the percentage of correct predictions. Show this percentage. Now run the logistic regression with the same training sample you have. Perform the prediction for the logistic regression in the test set. What is the percentage now for the logistic regression? Considering just the prediction accuracy, will you use the classification tree or the logistic regression?

Answer: The percetage for tree model is 0.852 and for logistic model is 0.877. I will choose logistic regression, since the accuracy of the predictions on the test set is higher for logistic regression.

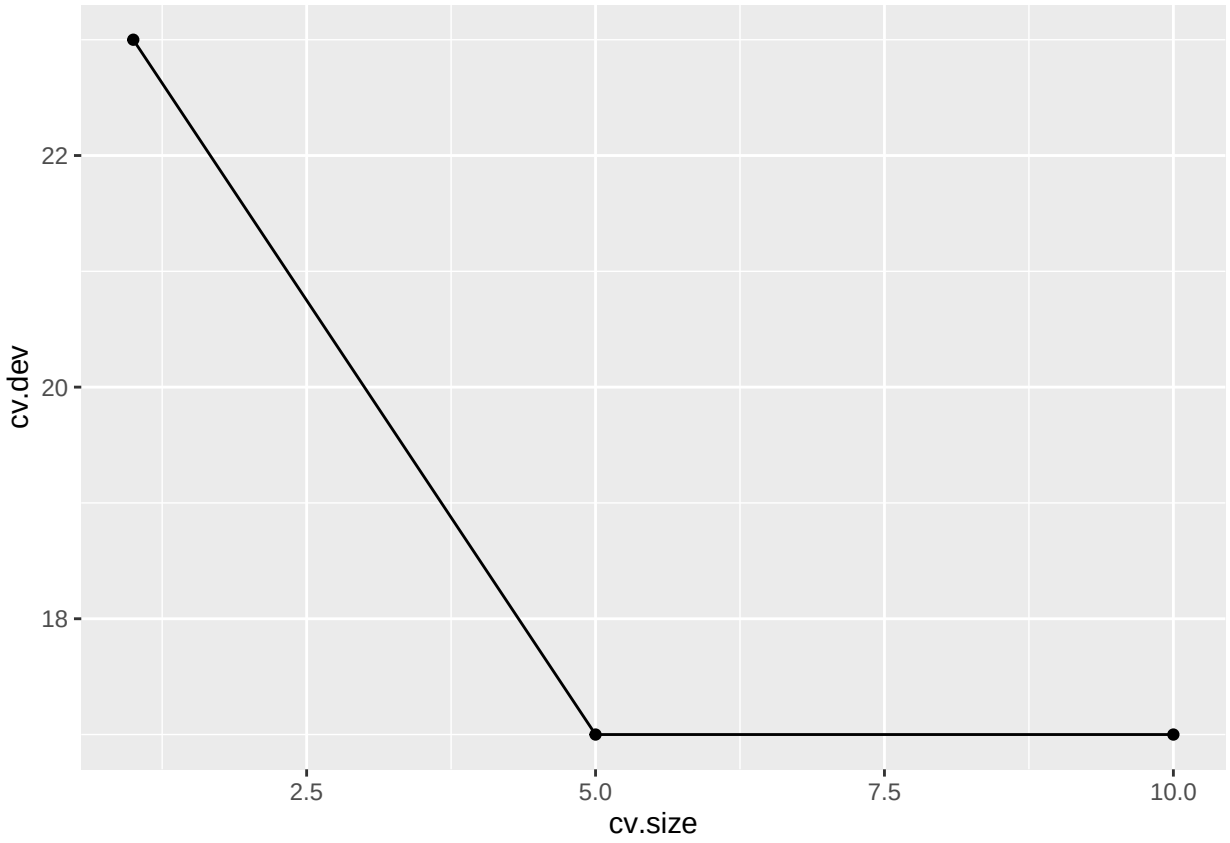
```
set.seed(2)
idx <- sample (1: nrow(df), nrow(df)/2)
train <- df[idx,]
test <- df[-idx,]
tree_model <- tree(Depression~., train)
tree_predictions <- predict(tree_model, test, type="class")
cm<-table(tree_predictions, test$Depression)
tree_accuracy <- (cm[1,1]+cm[2,2])/244

glm_model <- glm(Depression~., train, family = "binomial")
glm_predictions <- predict(glm_model, test)
glm_predictions <- ifelse(glm_predictions > 0.5 , 1, 0)
cm<-table(glm_predictions, test$Depression)
glm_accuracy <- (cm[1,1]+cm[2,2])/244
c(tree_accuracy, glm_accuracy)
```

```
## [1] 0.8606557 0.8606557
```

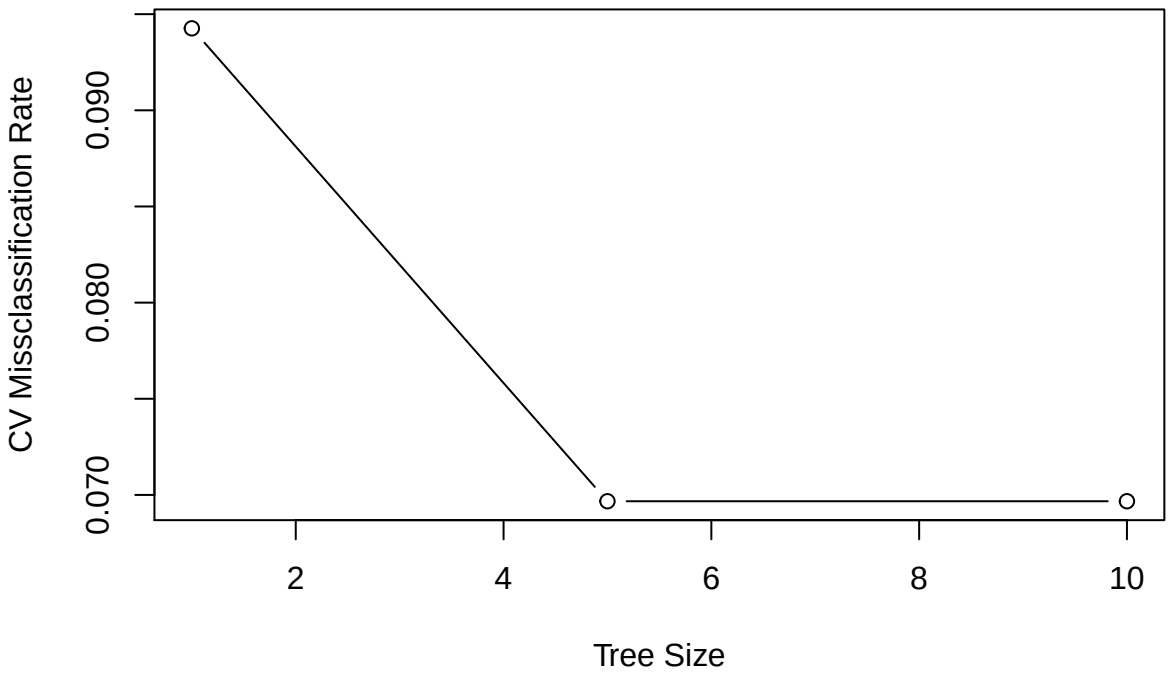
(c) (30 points) Prune the tree now. First, use the cross validation to know the best size to use for your tree (you will use this number in the function prune.misclass for best option). Plot the results by obtaining a plot with missclassification errors on the y axis. What is the best size you will choose? Now use the function prune.misclass to prune your tree with the best value selected. Calculate the percentage of the correct prediction with your pruned tree. Will it increase or decrease?

Answer: The best size is 5 according to cross validation. The imrpoved accuracy is 0.877 after pruning the tree.

```
cv <- cv.tree(tree_model ,FUN=prune.misclass)
df_cv <- data.frame(cv$size, cv$dev)
ggplot(data=df_cv, aes(x=cv.size, y=cv.dev)) + geom_point() + geom_line()
```




```
plot(cv$size, cv$dev / nrow(train), type="b", xlab="Tree Size", ylab= "CV Missclassification Rate")
```




```
pruned_tree <- prune.misclass(tree_model, best=cv$size[which.min(cv$dev[1:length(cv$dev)-1])])
pruned_tree_predictions <- predict(pruned_tree, test, type="class")
cm <- table(pruned_tree_predictions, test$Depression)
pruned_tree_accuracy <- (cm[1,1]+cm[2,2])/244 #
pruned_tree_accuracy
```

```
## [1] 0.8606557
```