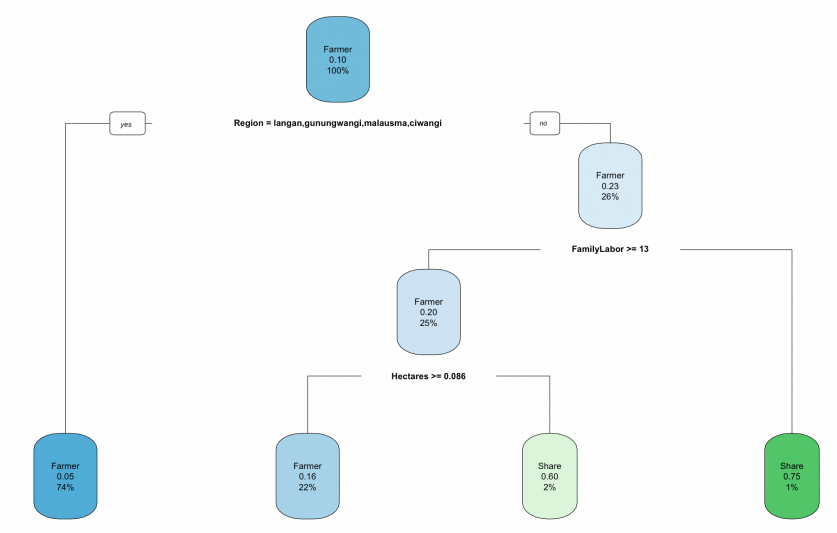
1. From R:
    1. All code from your R script (Code should be presented single-spaced in a fixed-width font. Adjust the font size so that no lines of code extend to the next line in the document)

```
# YUEH-TING WU
# MIS 545 Section 02
# Lab09WuY.R
# In this R programming, importing a csv file, and creating a
decision tree
# model to make a prediction.

# Instal the tidyverse and rpart.plot packages
# install.packages("tidyverse")
# install.packages("rpart.plot")

# Load the tidyverse, rpart, and rpart.plot libraries
library(tidyverse)
library(rpart)
library(rpart.plot)

# Set the working directory
setwd("~/MIS 545/Lab09")

# Read IndonesianRiceFarms.csv into a tibble called riceFarms
riceFarms <- read_csv(file = "IndonesianRiceFarms.csv",
                      col_types = "fniiinf",
                      col_names = TRUE)

# Display riceFarms in the console
print(riceFarms)

# Display the structure of riceFarms in the console
str(riceFarms)

# Display the summary of riceFarms in the console
summary(riceFarms)

# Randomly split the dataset into riceFarmsTraining (75% of
records) and
# riceFarmsTesting (25% of records) using 370 as the random seed
set.seed(370)
simpleSet <- sample(nrow(riceFarms),
                    round(nrow(riceFarms) * 0.75),
                    replace = FALSE)
riceFarmsTraining <- riceFarms[simpleSet, ]
```

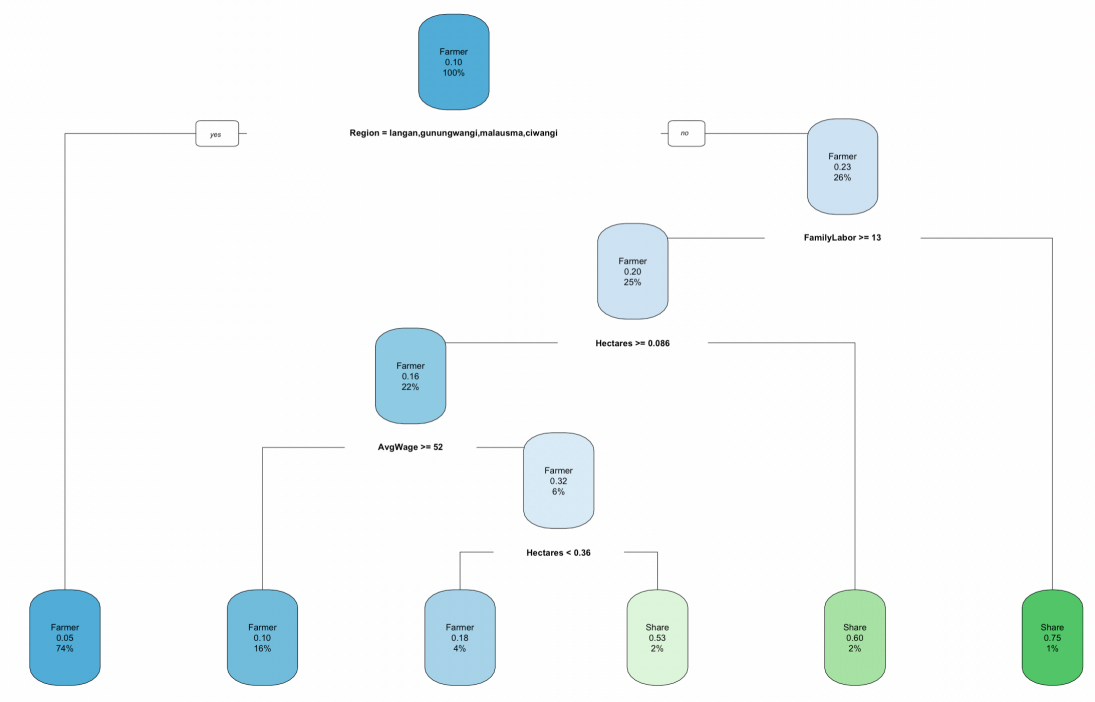
```
riceFarmsTesting <- riceFarms[-simpleSet, ]

# Generate the decision tree model to predict FarmOwnership
based on the other
# variables in the dataset. Use 0.01 as the complexity parameter
riceFarmsDecisionTreeModel <- rpart(formula = FarmOwnership ~ .,
                                    method = "class",
                                    cp = 0.01,
                                    data = riceFarmsTraining)

# Display the riceFarmsDecisionTreeModel in the console
rpart.plot(riceFarmsDecisionTreeModel)

# Predict the classes for each record in the testing dataset and
store them in
# riceFarmsPrediction
riceFarmsPrediction <- predict(riceFarmsDecisionTreeModel,
                               riceFarmsTesting,
                               type = "class")

# Display riceFarmsPrediction on the console
print(riceFarmsPrediction)

# Evaluate the model by forming a confusion matrix
riceFarmsConfusionMatrix <-
table(riceFarmsTesting$FarmOwnership,
                                  riceFarmsPrediction)

# Display the confusion matrix on the console
print(riceFarmsConfusionMatrix)

# Calculate the model predictive accuracy and store it into a
variable called
# predictiveAccuracy
predictiveAccuracy <- sum(diag(riceFarmsConfusionMatrix)) /
  nrow(riceFarmsTesting)

# Display the predictive accuracy on the console
print(predictiveAccuracy)

# Create a new decision tree model using 0.007 as the complexity
parameter

riceFarmsDecisionTreeModel <- rpart(formula = FarmOwnership ~ .,
                                    method = "class",
                                    cp = 0.007,
```

```
                                    data = riceFarmsTraining)

# Display the decision tree visualization
rpart.plot(riceFarmsDecisionTreeModel)

# Calculate its predictive accuracy
riceFarmsPrediction <- predict(riceFarmsDecisionTreeModel,
                               riceFarmsTesting,
                               type = "class")
print(riceFarmsPrediction)

# Evaluate the model by forming a confusion matrix
riceFarmsConfusionMatrix <-
table(riceFarmsTesting$FarmOwnership,
                                  riceFarmsPrediction)

# Display the confusion matrix on the console
print(riceFarmsConfusionMatrix)

# Calculate the model predictive accuracy and store it into a
variable called
# predictiveAccuracy
predictiveAccuracy <- sum(diag(riceFarmsConfusionMatrix)) /
  nrow(riceFarmsTesting)

# Display the predictive accuracy on the console
print(predictiveAccuracy)
```

2. An image of your decision tree visualization using cp = 0.01 (the default)



3. An image of your decision tree visualization using cp = 0.007



2. Answer the following question in a sentence: Did increasing the complexity of the decision tree improve the model's predictive accuracy? Why do you think this is the case?

Yes, it did, and I think that splitting the dataset into more small datasets is not helpful in this prediction