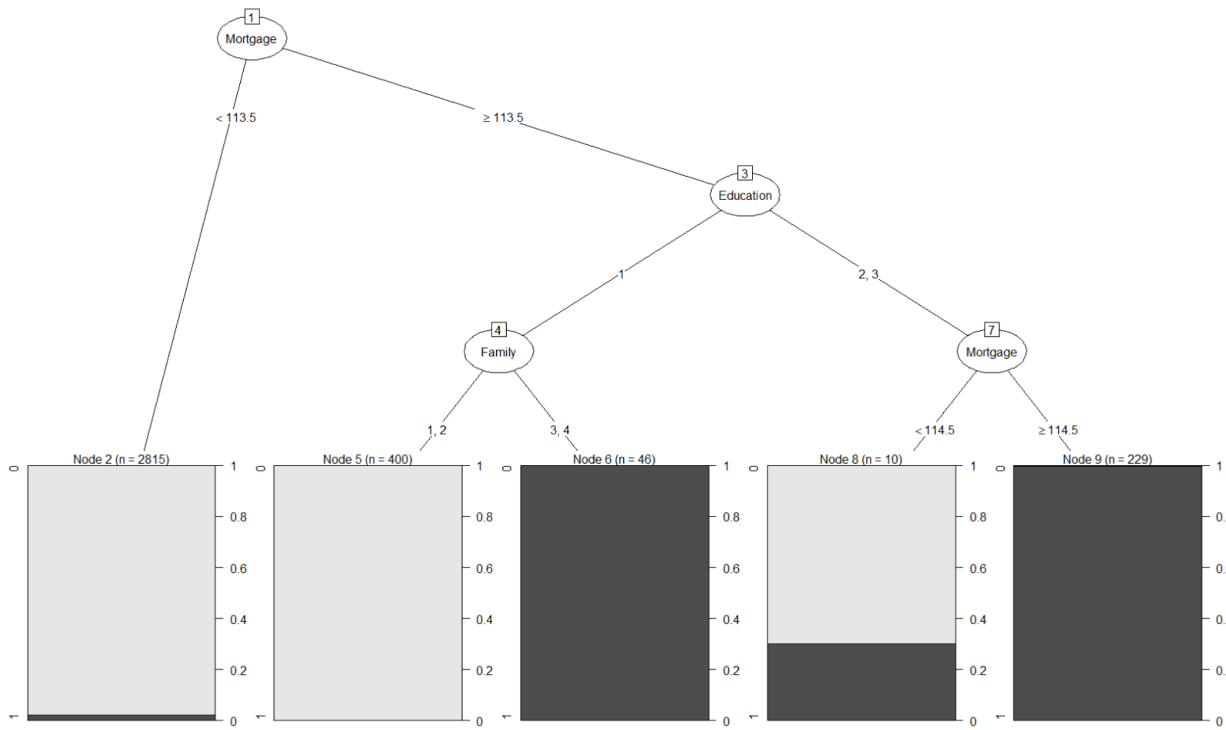
Assign3_Small

```r
#Libraries
library(caret)
library(ipred)
library(adabag)
library(randomForest)
library(rpart)
library(rpart.plot)
library(e1071)
library(ggplot2)
library(partykit)

df <- read.csv("/Users/Nick/Desktop/UniversalBank_Ensemble.csv", stringsAsFac
tors=FALSE)
Personal.Loan <- as.factor(df$Personal.Loan)
Education <- as.factor(df$Education)
Securities.Account <- as.factor(df$Securities.Account)
Family <- as.factor(df$Family)
CreditCard <- as.factor(df$CreditCard)
Age <- as.numeric(df$Age)
Experience <- as.numeric(df$Experience)
Income <- as.numeric(df$Income)
Mortgage <-as.numeric(df$Income)

df <- cbind(Personal.Loan, df[,-7])
df <- cbind(Education, df[,-6])
df <- cbind(Securities.Account, df[,-8])
df <- cbind(Family, df[,-7])
df <- cbind(CreditCard, df[,-9])
df <- cbind(Age, df[,-6])
df <- cbind(Experience, df[,-7])
df <- cbind(Income, df[,-8])
df <- cbind(Mortgage, df[,-9])
```

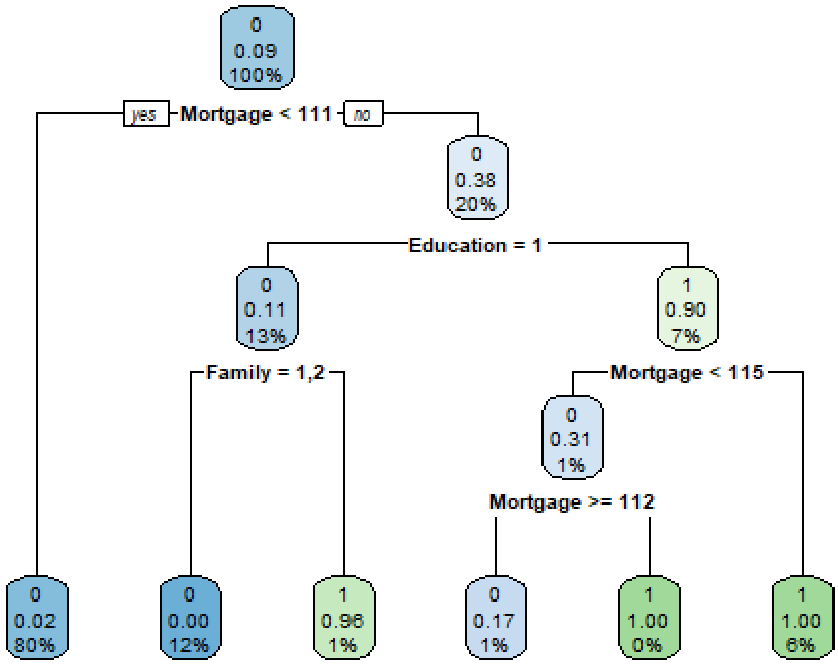
```r
#Training and splitting Data Set

set.seed(1234)
intrain <- createDataPartition(y=df$Personal.Loan, p=0.7, list=FALSE)
trainset <- df[intrain,]
testset <- df[-intrain,]
```

#Decision Tree

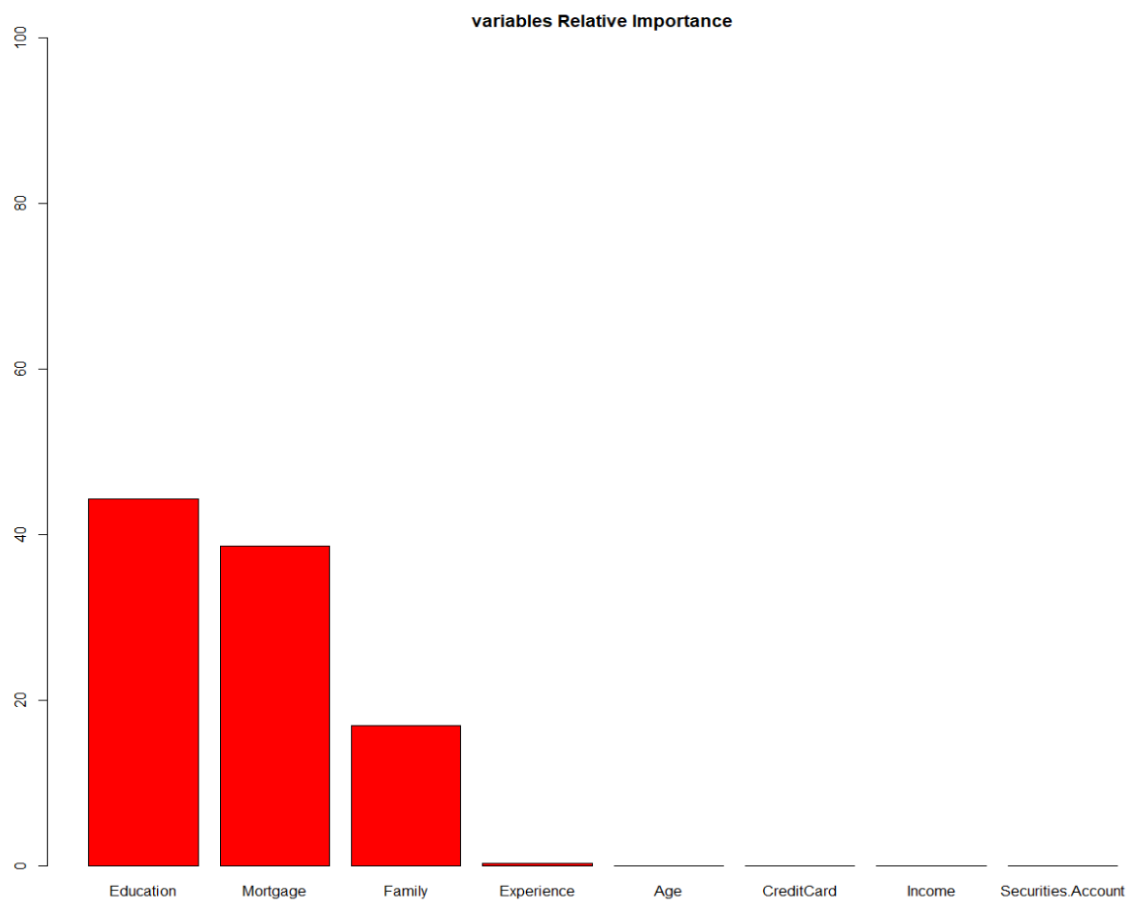
```r
tree <- rpart(Personal.Loan ~ ., data=trainset, method = "class")
tree <- as.party(tree)
plot(tree)
```




```r
#Bagging

df.single <- rpart(Personal.Loan~., data = trainset, method = 'class')

df.bagging <- bagging(Personal.Loan~., data = trainset, mfinal = 5, control = 
rpart.control(maxdepth=5,minsplit = 5))

rpart.plot(df.bagging$trees[[1]])
```




```
barplot(df.bagging$importance[order(df.bagging$importance, decreasing=TRUE)],
ylim=c(0,100), main="variables Relative Importance", col="red")
```



<u>The variables of importance for the bagging shows that Education, Mortgage, and Family are the most important variables. These are the factors that would determine whether a customer would choose to accept a personal loan. Using this bagging method we can also infer that the marketing team should target these 3 variables for a high chance of success.</u>

```
#Performance of Bagging

pred <- predict(df.single, testset, type = "class")
confusionMatrix(pred,testset$Personal.Loan)

## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0 1351   27
##          1    5  117
##
##                Accuracy : 0.9787
##                  95% CI : (0.97, 0.9854)
##     No Information Rate : 0.904
##     P-Value [Acc > NIR] : < 2.2e-16
```

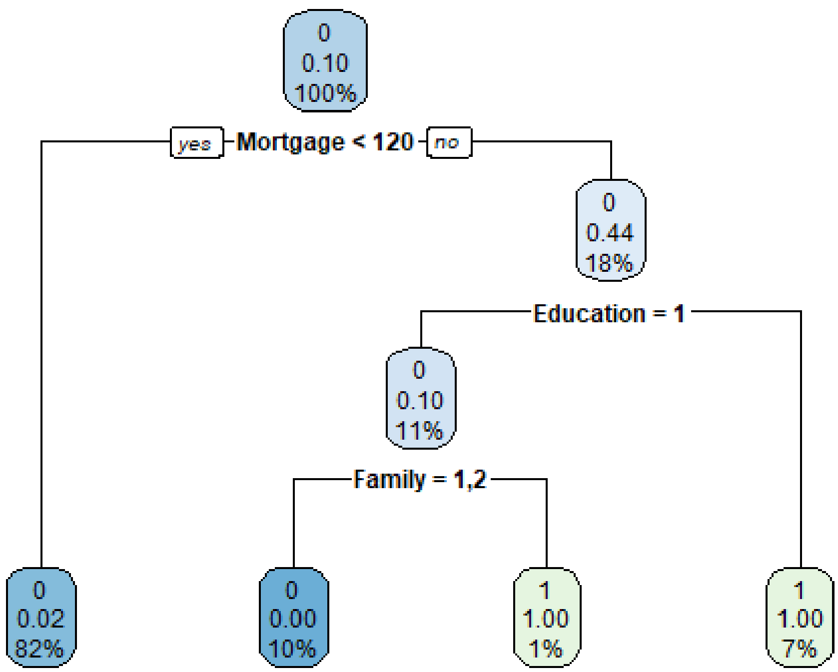
```
##
##                   Kappa : 0.8681
##
##  Mcnemar's Test P-Value : 0.0002054
##
##             Sensitivity : 0.9963
##             Specificity : 0.8125
##          Pos Pred Value : 0.9804
##          Neg Pred Value : 0.9590
##              Prevalence : 0.9040
##          Detection Rate : 0.9007
##    Detection Prevalence : 0.9187
##       Balanced Accuracy : 0.9044
##
##        'Positive' Class : 0
##
```

```r
pred2 <- predict(df.bagging, testset, type="class")

confusionMatrix(factor(pred2$class),testset$Personal.Loan)
```

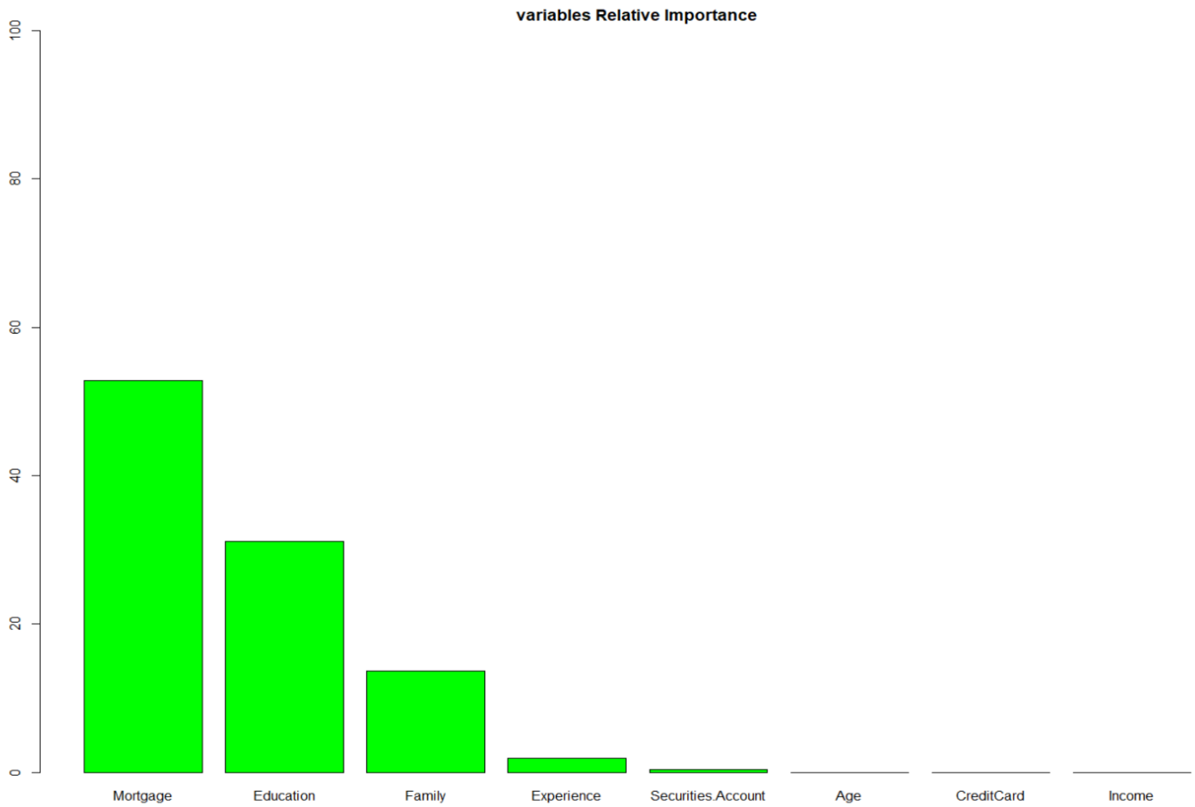
```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0 1351   27
##          1    5  117
##
##                Accuracy : 0.9787
##                  95% CI : (0.97, 0.9854)
##     No Information Rate : 0.904
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.8681
##
##  Mcnemar's Test P-Value : 0.0002054
##
##             Sensitivity : 0.9963
##             Specificity : 0.8125
##          Pos Pred Value : 0.9804
##          Neg Pred Value : 0.9590
##              Prevalence : 0.9040
##          Detection Rate : 0.9007
##    Detection Prevalence : 0.9187
##       Balanced Accuracy : 0.9044
##
##        'Positive' Class : 0
```

In Conclusion for Bagging, we observe that the accuracy of the model is 97.87 %. True Negative = 117, True Positive = 1351, False Negative = 5, and False Positive = 27. Comparing both pred and pred 2 we observe that nothing has changed in the accuracy.

```
#Boosting
set.seed(123)

df.boost <- boosting(Personal.Loan~., data = trainset, mfinal = 5, control =
rpart.control(maxdepth=5,minsplit = 5))

rpart.plot(df.boost$trees[[1]])
```




```
barplot(df.boost$importance[order(df.boost$importance, decreasing=TRUE)], ylim=c(0,100), main="variables Relative Importance", col="green")
```



We observe that Mortgage, Education, and Family are the most important variables in Boosting. These are the factors that would determine whether a customer would choose to accept a personal loan. Using this boosting method we can also infer that the marketing team should target these 3 variables for a high chance of success.

```
#Performance of Boosting

pred3 <- predict(df.single, testset, type = "class")
confusionMatrix(pred3,testset$Personal.Loan)

## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0 1351   27
##          1    5  117
##
##                Accuracy : 0.9787
##                  95% CI : (0.97, 0.9854)
##     No Information Rate : 0.904
##     P-Value [Acc > NIR] : < 2.2e-16
```

```
##
##                   Kappa : 0.8681
##
##  Mcnemar's Test P-Value : 0.0002054
##
##             Sensitivity : 0.9963
##             Specificity : 0.8125
##          Pos Pred Value : 0.9804
##          Neg Pred Value : 0.9590
##              Prevalence : 0.9040
##          Detection Rate : 0.9007
##    Detection Prevalence : 0.9187
##       Balanced Accuracy : 0.9044
##
##        'Positive' Class : 0
##
```

```
pred4 <- predict(df.boost, testset, type="class")

confusionMatrix(factor(pred4$class),testset$Personal.Loan)
```

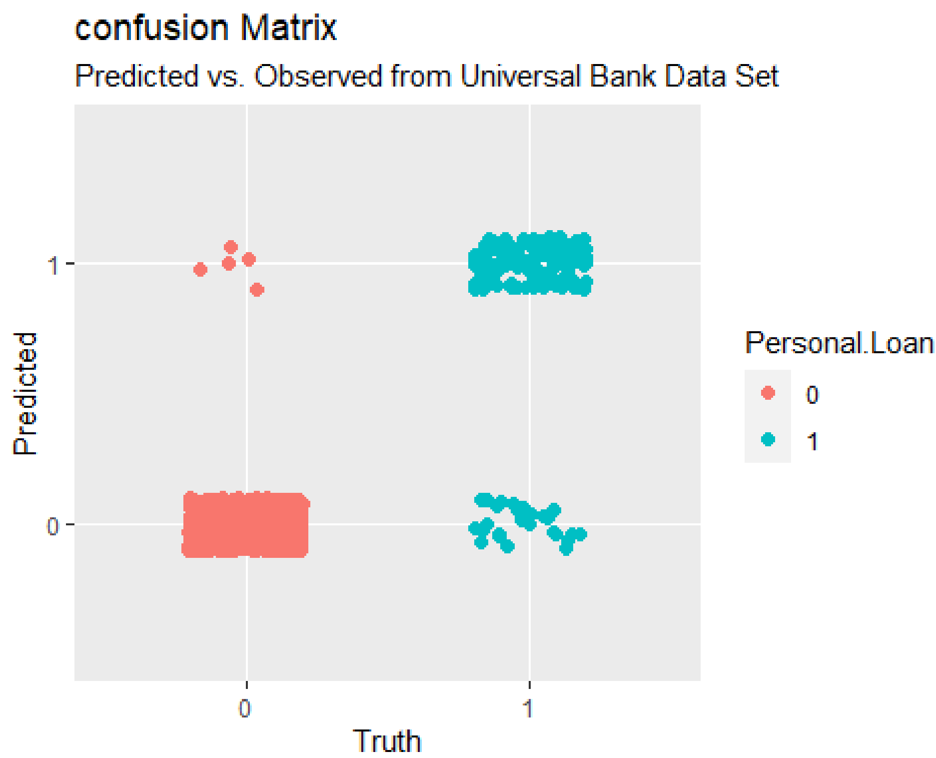
```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0 1338   23
##          1   18  121
##
##                Accuracy : 0.9727
##                  95% CI : (0.9631, 0.9803)
##     No Information Rate : 0.904
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.84
##
##  Mcnemar's Test P-Value : 0.5322
##
##             Sensitivity : 0.9867
##             Specificity : 0.8403
##          Pos Pred Value : 0.9831
##          Neg Pred Value : 0.8705
##              Prevalence : 0.9040
##          Detection Rate : 0.8920
##    Detection Prevalence : 0.9073
##       Balanced Accuracy : 0.9135
##
##        'Positive' Class : 0
```

In Conclusion for Boosting, we observe that the accuracy of the model is 97.27%. True Negative = 121, True Positive = 1338, False Negative = 18, and False Positive = 23. The Accuracy has decreased slightly, but not significantly.

```
#RandomForest

set.seed(400)
myForest <- randomForest(Personal.Loan~., nodesize = 3, mtry = 2, ntree = 15,
data=df)
myForest.pred <- predict(myForest, newdata = testset)

myForest

##
## Call:
##  randomForest(formula = Personal.Loan ~ ., data = df, nodesize = 3,      mtry = 2, ntree = 15)
##                Type of random forest: classification
##                      Number of trees: 15
## No. of variables tried at each split: 2
##
##         OOB estimate of  error rate: 2.02%
## Confusion matrix:
##      0   1 class.error
## 0 4497  17 0.003766061
## 1   84 395 0.175365344
```

Using the Random forest tree we observe that this model is 97.98% accurate. This is phenomenal. Below we can see that there is no significant deviation within the graph which is good. This proves that RandomForest is the best model from all three that should be used to determine how to increase the banks customers base that will accept more personal loans and which customers to target. In Random Forest we observe in the last chart below that Mortgage, Education, and Income are the factors that would increase a customers chance of accepting a personal loan and that is the criterias that should be targeted by the marketing team.

```
ggplot(testset, aes(Personal.Loan, pred, color = Personal.Loan)) +
  geom_jitter(width = 0.2, height = 0.1, size = 2) +
  labs(title="confusion Matrix",
       subtitle="Predicted vs. Observed from Universal Bank Data Set",
       y="Predicted", x="Truth")
```




```
varImpPlot(myForest)
```