MSC class demo on loan delinquent case study

Ipinnuoluwa

2024-11-13

Load Packages

```
library(ggplot2) # For graphs and visualizations
library(caTools) # Split Data into Test and Train Set
library(rpart) # Recursive Partitioning and Regression Trees (Decision Trees)
library(rattle) # To visualize decision tree
```

Import Data

```
setwd("/home/tolu/Desktop/dddd")
loandata = read.csv('Loan_Delinquent_Dataset.csv')
```

Exploratory Data Analysis

Sanity checks

```
# Look at the first and last few rows to ensure that the data is read in properly
head(loandata)
```

```
##   ID isDelinquent      term gender purpose home_ownership   age    FICO
## 1  1            1 36 months Female   House       Mortgage   >25 300-500
## 2  2            0 36 months Female   House           Rent 20-25    >500
## 3  3            1 36 months Female   House           Rent   >25 300-500
## 4  4            1 36 months Female     Car       Mortgage   >25 300-500
## 5  5            1 36 months Female   House           Rent   >25 300-500
## 6  6            1 36 months   Male     Car            Own   >25    >500
```

```
tail(loandata)
```

```
##          ID isDelinquent      term gender  purpose home_ownership   age    FICO
## 11543 11543            1 36 months   Male    House       Mortgage   >25    >500
## 11544 11544            0 60 months   Male    other       Mortgage   >25 300-500
## 11545 11545            1 36 months   Male    House           Rent 20-25 300-500
## 11546 11546            0 36 months Female Personal       Mortgage 20-25    >500
## 11547 11547            1 36 months Female    House           Rent 20-25 300-500
## 11548 11548            1 36 months   Male Personal       Mortgage 20-25 300-500
```

```
dim(loandata)
```

```
## [1] 11548     8
```

```
colnames(loandata)
```

```
## [1] "ID"             "isDelinquent"   "term"           "gender"
## [5] "purpose"        "home_ownership" "age"            "FICO"
```

Descriptive Statistics

```r
# Structure of data
str(loandata)
```

```
## 'data.frame':    11548 obs. of  8 variables:
##  $ ID            : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ isDelinquent  : int  1 0 1 1 1 1 0 0 1 1 ...
##  $ term          : chr  "36 months" "36 months" "36 months" "36 months" ...
##  $ gender        : chr  "Female" "Female" "Female" "Female" ...
##  $ purpose       : chr  "House" "House" "House" "Car" ...
##  $ home_ownership: chr  "Mortgage" "Rent" "Rent" "Mortgage" ...
##  $ age           : chr  ">25" "20-25" ">25" ">25" ...
##  $ FICO          : chr  "300-500" ">500" "300-500" "300-500" ...
```

- The dataset has 11548 rows and 8 columns of data
- ID column does not hold any statistical significance
- isDelinquent is the dependent variable - type integer.
- isDelinquent is a an integer variable and should be converted to factor for further analysis.
- All the dependent variables are of factor type.

```r
# Change vaiables with character datatype to factor
loandata$isDelinquent = as.factor(loandata$isDelinquent)
loandata$term = as.factor(loandata$term)
loandata$gender = as.factor(loandata$gender)
loandata$purpose = as.factor(loandata$purpose)
loandata$home_ownership = as.factor(loandata$home_ownership)
loandata$age = as.factor(loandata$age)
loandata$FICO = as.factor(loandata$FICO)

# Remove ID Column
loandata = loandata[, -1] # Dropping ID Column

# Summary of dataset
summary(loandata)
```

```
##  isDelinquent        term          gender          purpose      home_ownership
##  0:3827       36 months:10589   Female:4993   Car     :2080   Mortgage:5461
##  1:7721       60 months:  959   Male  :6555   House   :6892   Own     : 871
##                                               Medical : 266   Rent    :5216
##                                               other   :  82
##                                               Other   : 928
##                                               Personal: 892
##                                               Wedding : 408
##     age             FICO
##  >25  :5660   >500   :5178
##  20-25:5888   300-500:6370
##
##
##
##
##
```

Observations

- Most of the loans are for a 36 month term loan.
- Customers in the age group 20-25 are almost as many as those of age >25

- Most loan applications that we get are for house loans followed by car loans
- There are 2 levels named 'other' and 'Other' under purpose variable. Since we do not have any other information about these, we will merge these levels
- There are no missing vaues in out dataset
- Most customers have either mortgaged their houses or live on rent. Very few applicants <10% own their house

Data Cleaning

```
levels(loandata$purpose)
```

```
## [1] "Car"      "House"    "Medical"  "other"    "Other"    "Personal" "Wedding"
```

```
#Merge the purpose levels 'Other' and 'other'
levels(loandata$purpose) = c("Car","House","Medical","Other","Other","Personal","Wedding")
levels(loandata$purpose)
```

```
## [1] "Car"      "House"    "Medical"  "Other"    "Personal" "Wedding"
```

```
summary(loandata)
```

```
##  isDelinquent        term          gender         purpose      home_ownership
##  0:3827       36 months:10589   Female:4993   Car     :2080   Mortgage:5461
##  1:7721       60 months:  959   Male  :6555   House   :6892   Own     : 871
##                                               Medical : 266   Rent    :5216
##                                               Other   :1010
##                                               Personal: 892
##                                               Wedding : 408
##     age             FICO
##  >25  :5660   >500   :5178
##  20-25:5888   300-500:6370
##
##
##
##
```

Univariate and Bivariate analysis

```
#Distribution of the dependent variable
prop.table(table(loandata$isDelinquent))
```

```
##
##         0         1
## 0.3313994 0.6686006
```

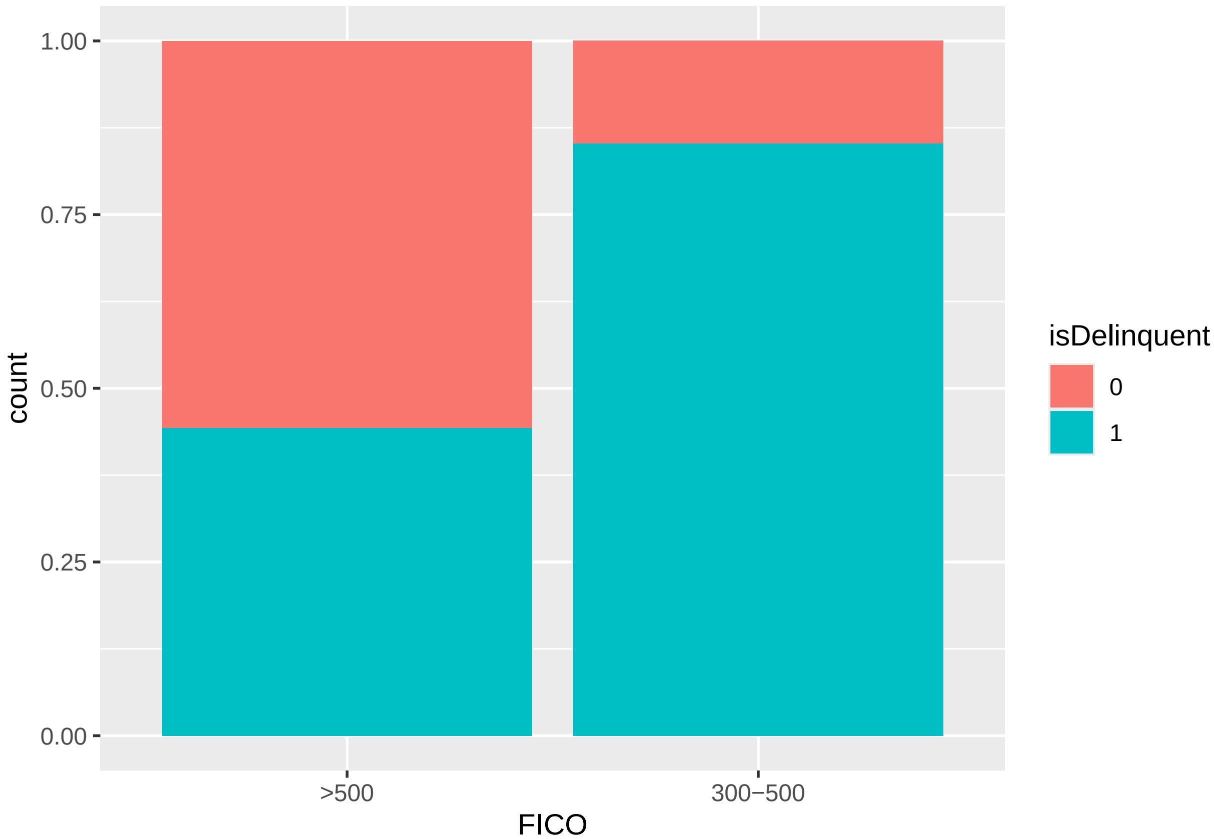
```
summary(loandata)
```

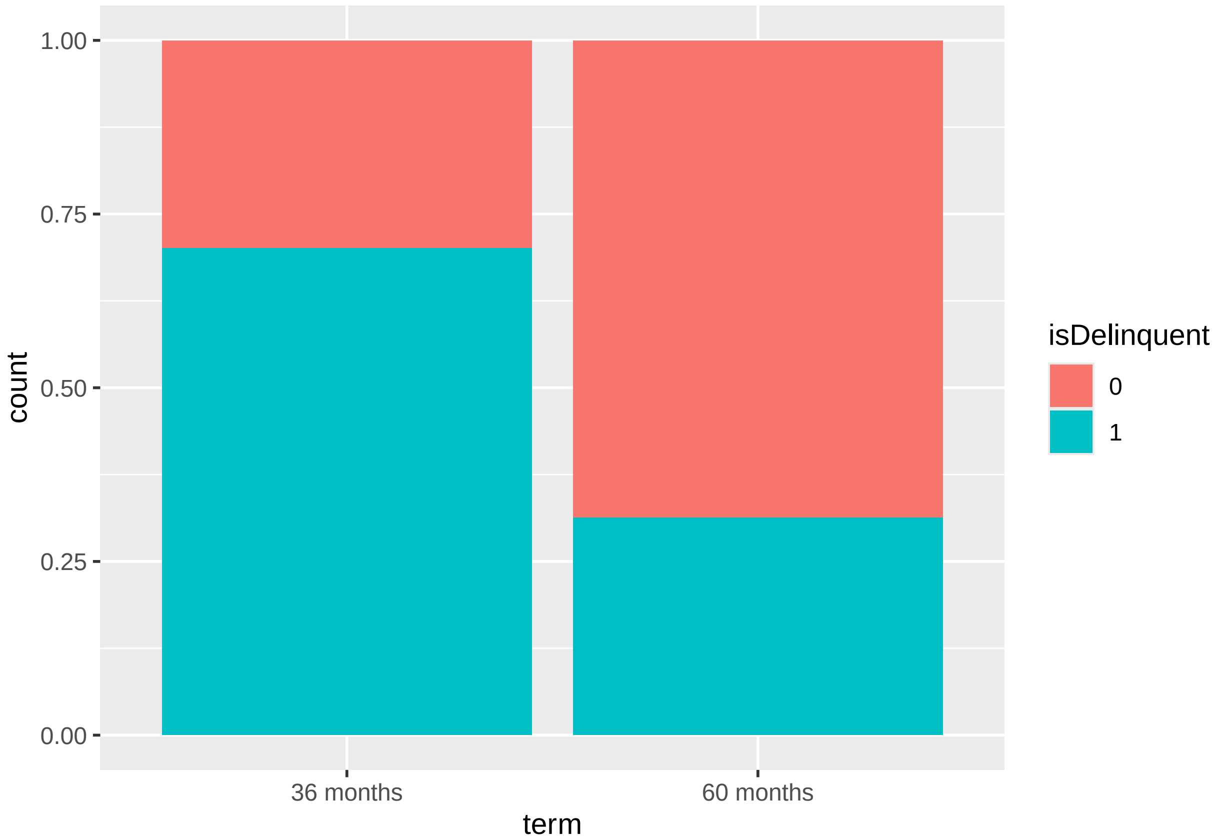
```
##  isDelinquent        term          gender         purpose      home_ownership
##  0:3827       36 months:10589   Female:4993   Car     :2080   Mortgage:5461
##  1:7721       60 months:  959   Male  :6555   House   :6892   Own     : 871
##                                               Medical : 266   Rent    :5216
##                                               Other   :1010
##                                               Personal: 892
##                                               Wedding : 408
##     age             FICO
##  >25  :5660   >500   :5178
```

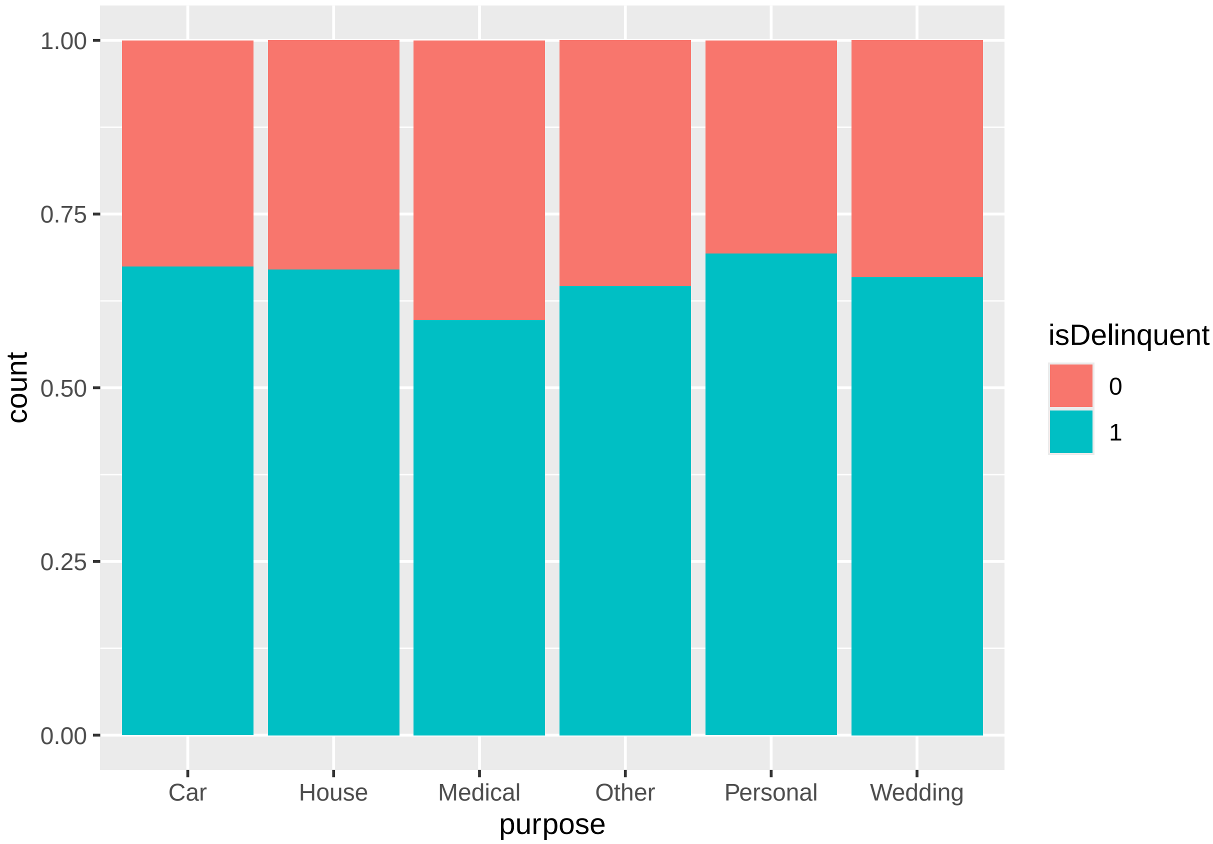
```
##   20-25:5888   300-500:6370
##
##
##
##
```

66% of the customers are delinquent.

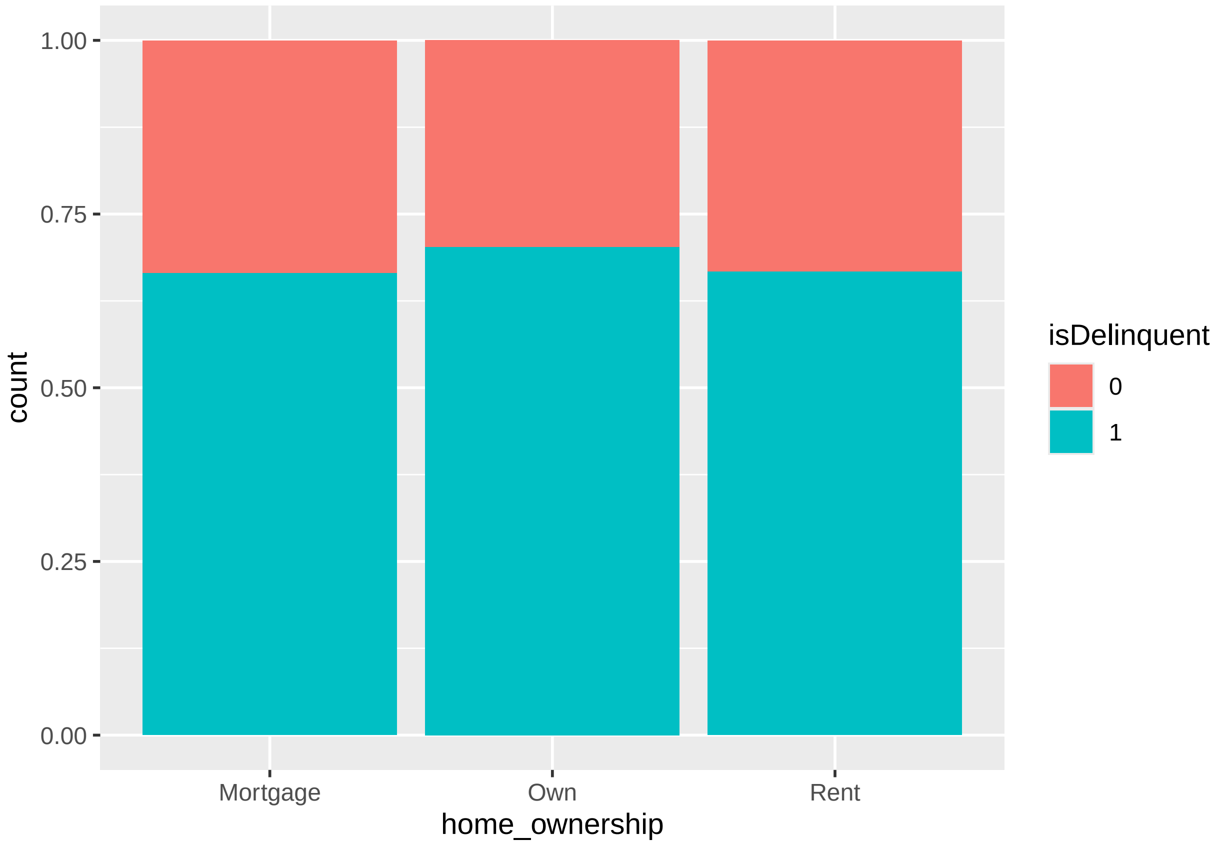
Let us plot percent stacked barchart to see the effect of independent variables on the probability of delinquency

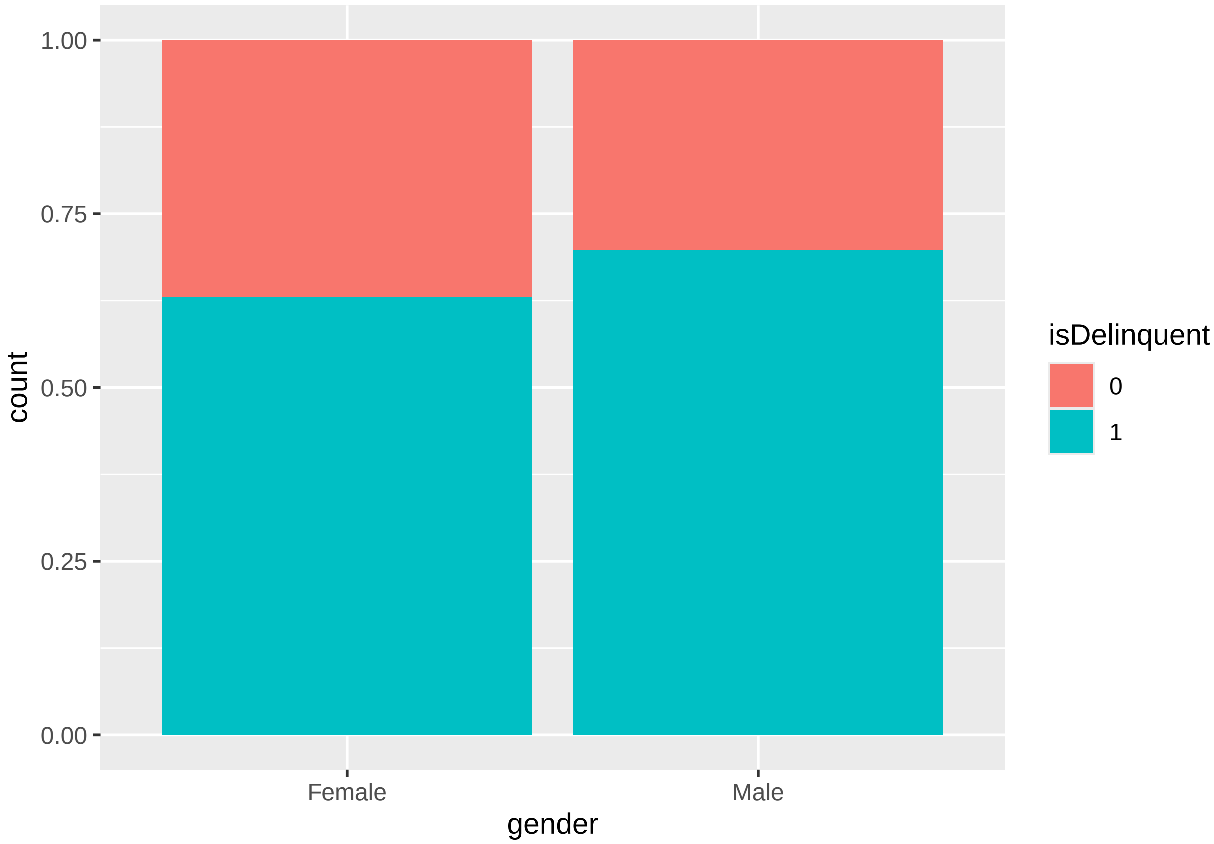



1.00
0.75
0.50
0.25
0.00
count
36 months
60 months
term
isDelinquent
0
1

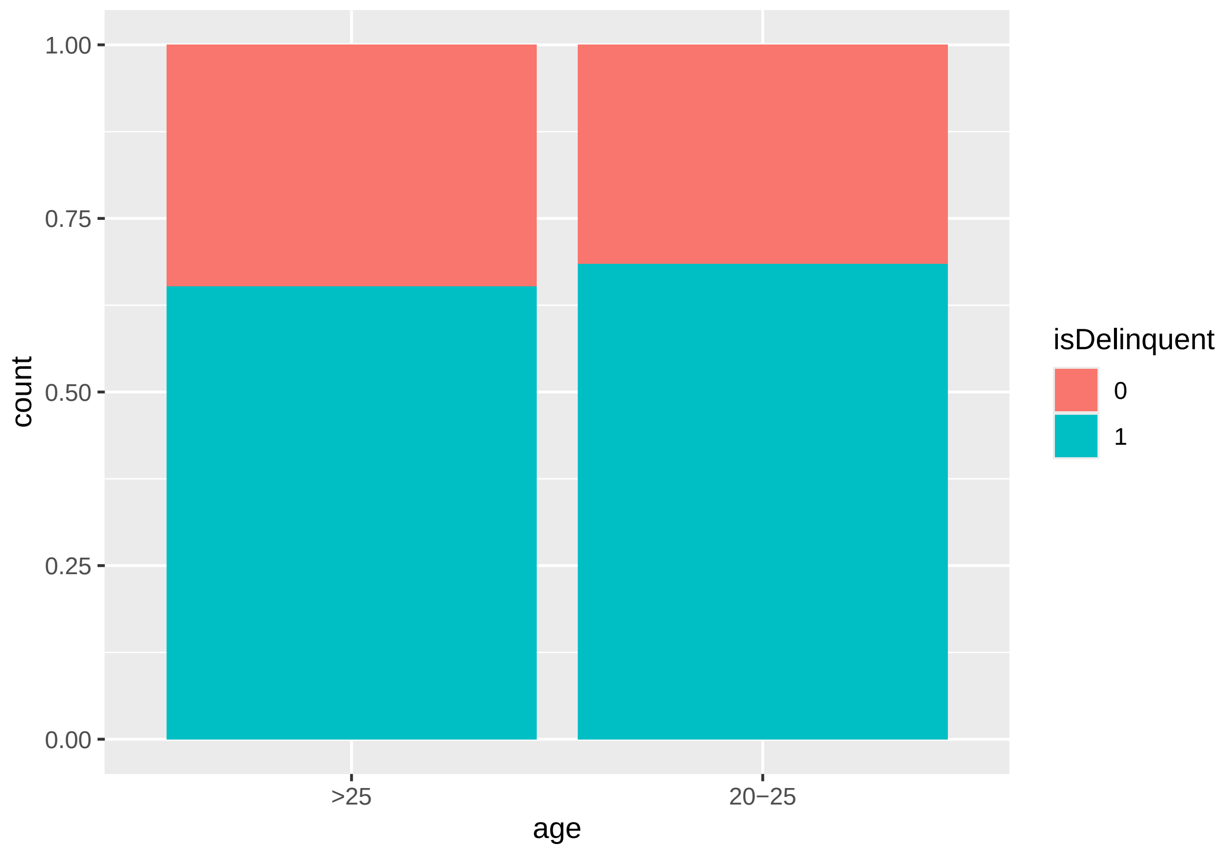


1.00
0.75
0.50
0.25
0.00
count
Car
House
Medical
Other
Personal
Wedding
purpose
isDelinquent
0
1


1.00
0.75
0.50
0.25
0.00
count
Mortgage
Own
Rent
home_ownership
isDelinquent
0
1



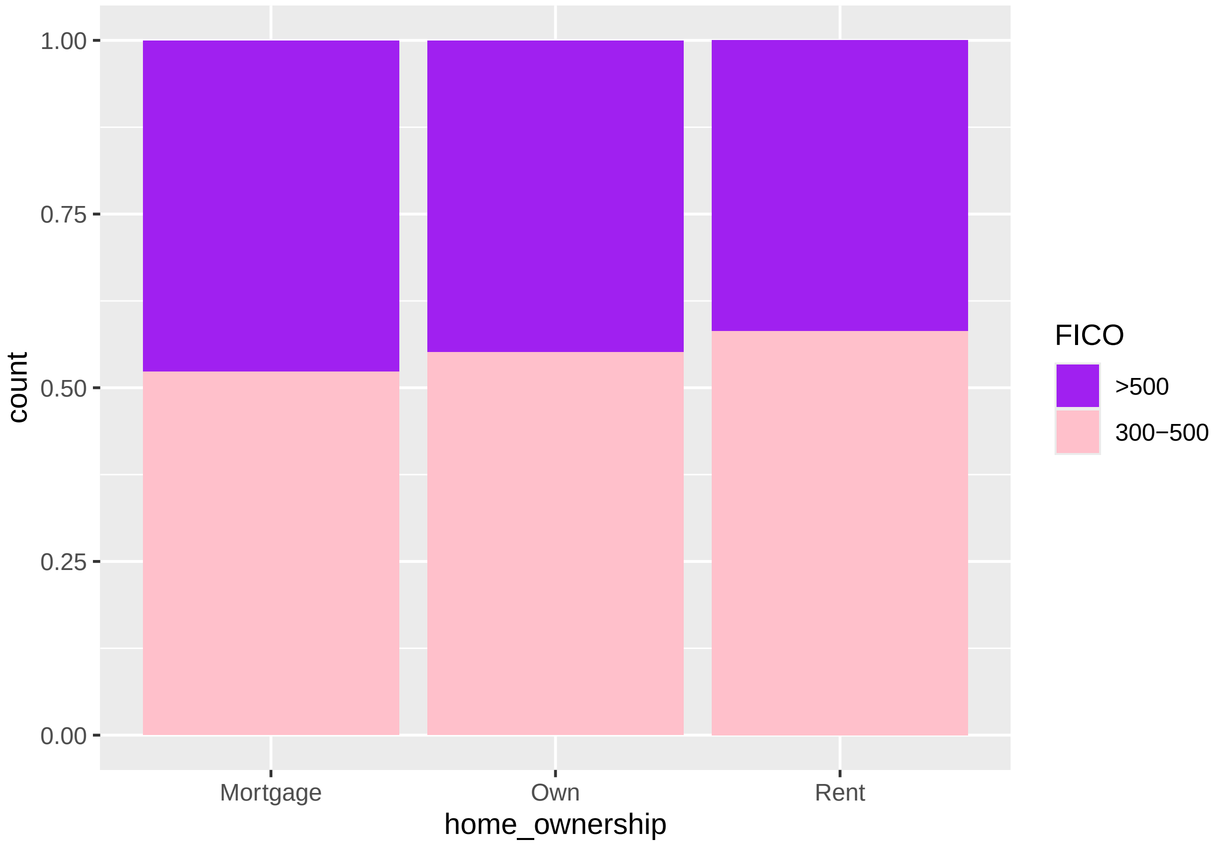
1.00
0.75
0.50
0.25
0.00
count
Female
Male
gender
isDelinquent
0
1

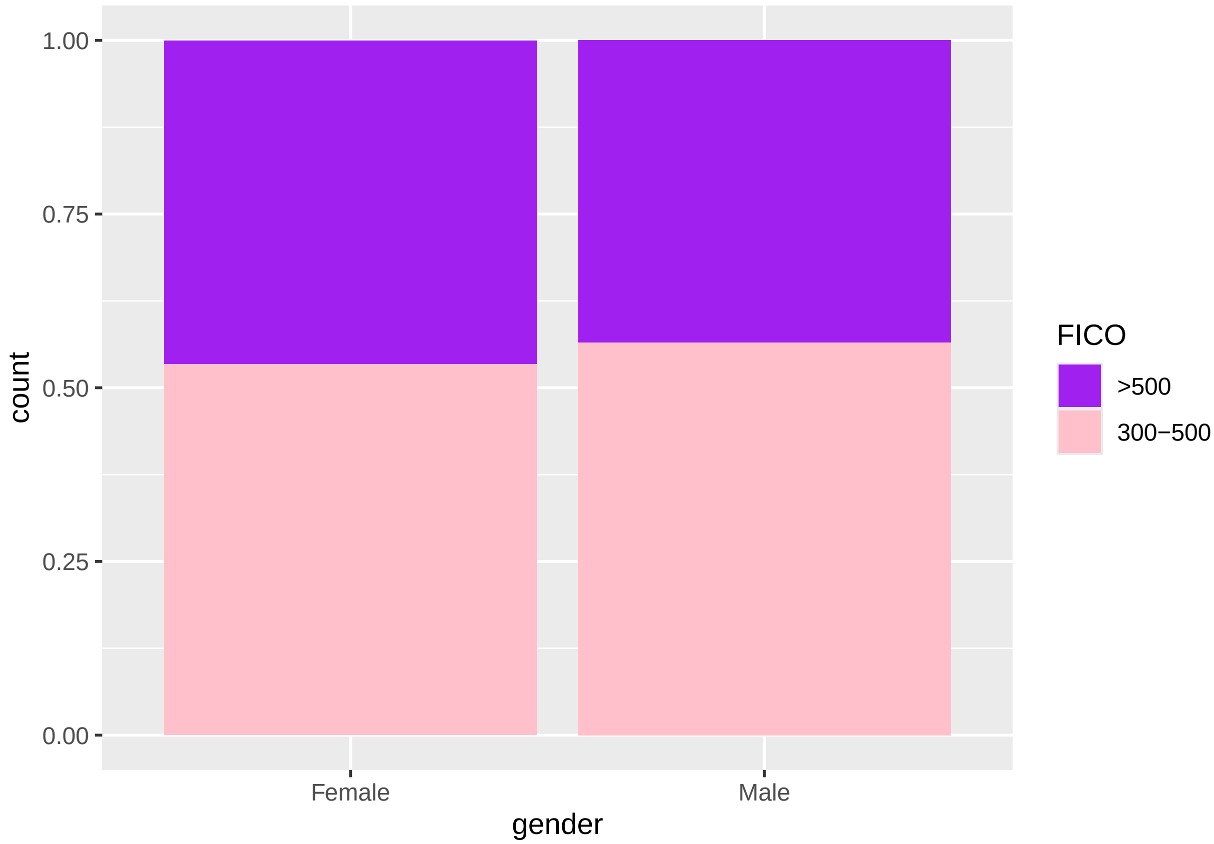


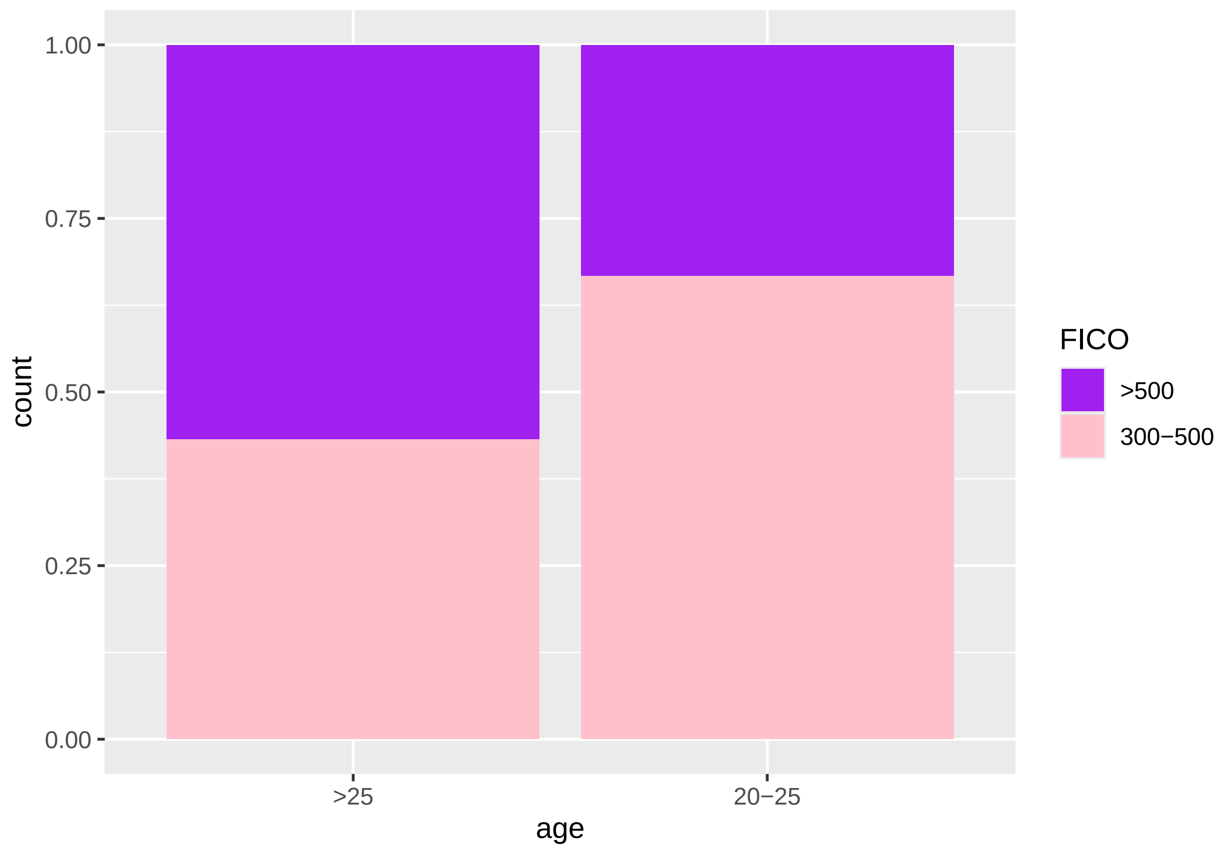
Observations

- FICO score and term of loan application appear to be very strong indicators of delinquency.
- If FICO score is >500 the chances of delinquency decrease quite a lot compared to when FICO score is between 300-500
- Similarly, loans for 60 month term loan seem to be prominently from the delinquent customers.
- Other factors appear to be not very good indicators of delinqency. (We can use chi square tests to determine statistical significance in the association between two categorical variables)

We observed that a high FICO score means that the chances of delinquency are lower, let us see if any of the other variables indicate higher FICO scores


1.00
0.75
0.50
0.25
0.00
count
Mortgage
Own
Rent
home_ownership
FICO
>500
300−500


1.00
0.75
0.50
0.25
0.00
count
Female
Male
gender
FICO
>500
300−500



1. Home ownership and gender seem to have slight impact on the FICO scores.
2. Age seems to have a much bigger impact on FICO scores.

Let us check which of these differences are statistically significant.

Similar to z-tests and t-tests that we have learned for numerical variables. Chi-Square test is a statistical method to determine if two categorical variables have a significant correlation between them. (https://www.tutorialspoint.com/r/r_chi_square_tests.htm)

Null Hypothesis - There is no correlation between the two categorical variables
Alternate Hypothesis - Variable A is correlated with variable B

```
chisq.test(loandata$FICO,loandata$home_ownership) #Chi-sq test between FICO and Home Ownership
```

```
##
##  Pearson's Chi-squared test
##
## data:  loandata$FICO and loandata$home_ownership
## X-squared = 36.682, df = 2, p-value = 1.083e-08
```

```
chisq.test(loandata$FICO,loandata$gender) #Chi-sq test between FICO and Gender
```

```
##
##  Pearson's Chi-squared test with Yates' continuity correction
##
## data:  loandata$FICO and loandata$gender
## X-squared = 11.222, df = 1, p-value = 0.0008085
```

```r
chisq.test(loandata$FICO,loandata$age) #Chi-sq test between FICO and Age
```

```
##
##  Pearson's Chi-squared test with Yates' continuity correction
##
## data:  loandata$FICO and loandata$age
## X-squared = 645.18, df = 1, p-value < 2.2e-16
```

p-values are « 0.01
All the differences that we see in the 3 plots are infact statistically significant.

1. There is a correlation between FICO Score and house_ownership. People who have mortgaged their houses have higher FICO scores than people who own the house (peculiar!).
2. There is a correlation between FICO Score and gender. More females have >500 FICO scores as compared to Males.
3. There is a correlation between FICO Score and age. People >25 years of age have higher FICO scores as compared to people of age 20-25.

Model Building - Approach

1. Partition the data into train and test set.
2. Built a CART model on the train data.
3. Tune the model and prune the tree, if required.
4. Test the data on test set.

Split into train and test

```r
set.seed(1000) #To ensure reproducibility

sample <- sample.split(loandata$isDelinquent,SplitRatio = 0.7)
train <- subset(loandata,sample == TRUE)
test <- subset(loandata,sample == FALSE)

nrow(train)
```

```
## [1] 8084
```

```r
nrow(test)
```

```
## [1] 3464
```

```r
# Check that the distribution of the dependent variable is similar in train and test sets
prop.table(table(loandata$isDelinquent))
```

```
##
##         0         1
## 0.3313994 0.6686006
```

```r
prop.table(table(train$isDelinquent))
```

```
##
##         0         1
## 0.3313953 0.6686047
```

```r
prop.table(table(test$isDelinquent))
```

```
##
##         0         1
```

```
## 0.3314088 0.6685912
```

Build a CART model on the train dataset

We will use the "rpart" and the "rattle" libraries to build decision trees.

```
# One of the benefits of decision tree training is that you can stop training
# based on several thresholds.

# Setting the control parameters (to control the growth of the tree)

# minsplit - min # of obs that must exist in a node in order for a split to be attempted
# minbucket - the minimum number of observations in any terminal leaf node
# cp - complexity parameter
# xval - number of cross-validations

# The initial minsplit and minbucket parameters are set using general thumb rules
# minsplit = 2-3% of data
# minbucket = minsplit/3

min_split = 0.2*nrow(train)
min_split
```

```
## [1] 1616.8
```

```
min_bucket = min_split/3
min_bucket
```

```
## [1] 538.9333
```

```
r.ctrl = rpart.control(minsplit = min_split, minbucket = min_bucket, cp = 0, xval = 10)

# Building the CART model

# formula - response variable~predictor variables
# data - dataset
# method - "class" - for classification, "anova" for regression
# control - tree control parameters

model1 <- rpart(formula = isDelinquent~., data = train, method = "class", control = r.ctrl)
model1
```

```
## n= 8084
##
## node), split, n, loss, yval, (yprob)
##       * denotes terminal node
##
##  1) root 8084 2679 1 (0.3313953 0.6686047)
##    2) FICO=>500 3633 1589 0 (0.5626204 0.4373796)
##      4) gender=Female 1687  393 0 (0.7670421 0.2329579) *
##      5) gender=Male 1946  750 1 (0.3854060 0.6145940)
##       10) age=20-25 678  332 0 (0.5103245 0.4896755) *
##       11) age=>25 1268  404 1 (0.3186120 0.6813880) *
##    3) FICO=300-500 4451  635 1 (0.1426646 0.8573354) *
```

- Let us understand the decision tree created

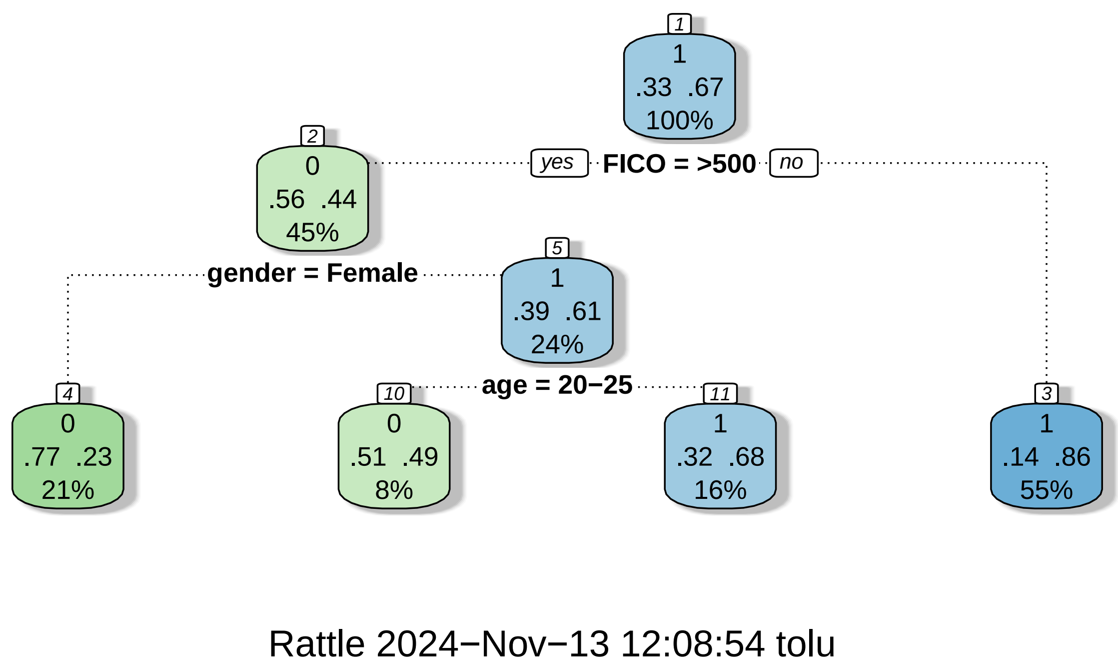
– node) - Node number

– split - split or test
– n - number of entities at that node
– loss - number of incorrectly classified entities at that node
– yval - default classification for that node
– (yprob) - the distribution of classes in that node (The distribution is ordered by the classes, and is the same order for all nodes)
– '*' - denotes terminal node (i.e., the tree is not split any further at that node)

- The first node of any tree is always the root node.

Visualise the decision tree

```
#Displaying the decision tree
fancyRpartPlot(model1)
```




```
model1$variable.importance
```

```
##       FICO     gender        age       term    purpose
## 705.560017 263.221569 146.280355  61.512278   9.574277
```

```
# Variable importance is generally computed based on the corresponding reduction of predictive accuracy
# when the predictor of interest is removed.
```

The major advantage of a decision tree model over other classification models is that it gives a highly interpretable output.

We can look at the tree and determine underlaying business rules in our data.

Let us understand this tree output

The right most node -
Condition -> FICO = >500 NO
Class -> isDelinquent = 1
Actual isDelinquent = 0 -> 14%
Actual isDelinquent = 1 -> 86%
Contains 55% of the training dataset

The left most node -
Condition -> FICO = >500 YES and gender = Female YES
Class -> isDelinquent = 0
Actual isDelinquent = 0 -> 77%
Actual isDelinquent = 1 -> 23%
Contains 21% of the training dataset

Model Validation

```
# Predicting on the train dataset
train_predict.class1 <- predict(model1, train, type="class") # Predicted Classes
train_predict.score1 <- predict(model1, train) # Predicted Probabilities

# Create confusion matrix for train data predictions
tab.train1 = table(train$isDelinquent, train_predict.class1)
tab.train1
```

```
##    train_predict.class1
##        0    1
##   0 1640 1039
##   1  725 4680
```

```
# Accuracy on train data
accuracy.train1 = sum(diag(tab.train1)) / sum(tab.train1)
accuracy.train1
```

```
## [1] 0.7817912
```

78% accuracy is a clear improvement on the baseline model.
The baseline model for this data would predict all borrowers as delinquent.
Baseline accuracy would be 66%.

Let us see if we can improve model performance furthur by tuning the model.

Model Tuning

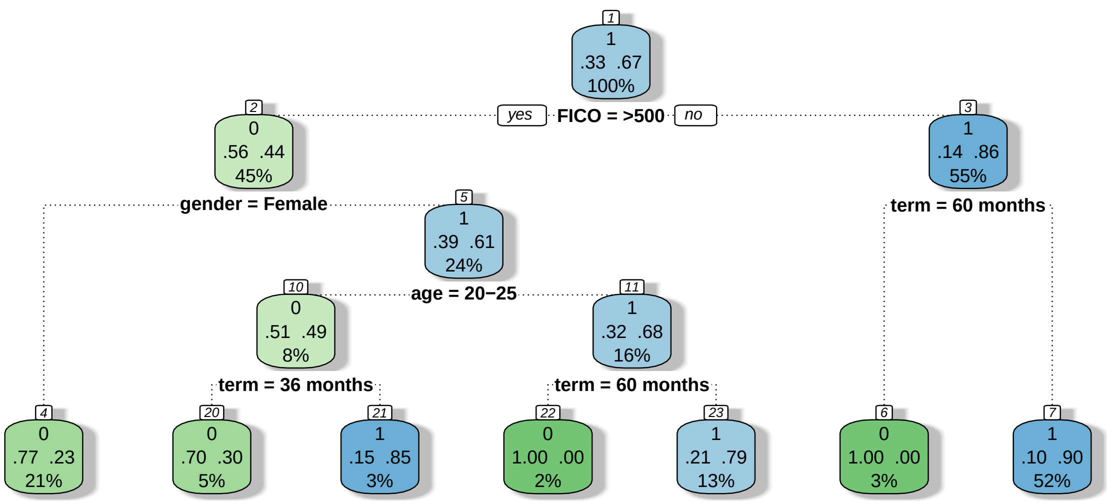
```
# Let us ease the control parameter restrictions to check if we can get a better better fit
r.ctrl2 = rpart.control(minsplit = 500, minbucket = 150, cp = 0, xval = 10)

# Building the CART model

model2 <- rpart(formula = isDelinquent~., data = train, method = "class", control = r.ctrl2)
model2
```

```
## n= 8084
##
## node), split, n, loss, yval, (yprob)
```

```
##       * denotes terminal node
##
##  1) root 8084 2679 1 (0.33139535 0.66860465)
##    2) FICO=>500 3633 1589 0 (0.56262042 0.43737958)
##      4) gender=Female 1687  393 0 (0.76704209 0.23295791) *
##      5) gender=Male 1946  750 1 (0.38540596 0.61459404)
##       10) age=20-25 678  332 0 (0.51032448 0.48967552)
##         20) term=36 months 444  134 0 (0.69819820 0.30180180) *
##         21) term=60 months 234   36 1 (0.15384615 0.84615385) *
##       11) age=>25 1268  404 1 (0.31861199 0.68138801)
##         22) term=60 months 178    0 0 (1.00000000 0.00000000) *
##         23) term=36 months 1090  226 1 (0.20733945 0.79266055) *
##    3) FICO=300-500 4451  635 1 (0.14266457 0.85733543)
##      6) term=60 months 232    0 0 (1.00000000 0.00000000) *
##      7) term=36 months 4219  403 1 (0.09552027 0.90447973) *
```

Visualise the decision tree

```
#Displaying the decision tree
fancyRpartPlot(model2)
```



Rattle 2024−Nov−13 12:08:55 tolu

Model Validation

```
# Predicting on the train dataset
train_predict.class2 <- predict(model2, train, type="class") # Predicted Classes
```

```r
train_predict.score2 <- predict(model2, train) # Predicted Probabilities

# Create confusion matrix for train data predictions
tab.train2 = table(train$isDelinquent, train_predict.class2)
tab.train2
```

```
##    train_predict.class2
##        0    1
##   0 2014  665
##   1  527 4878
```

```r
# Accuracy on train data
accuracy.train2 = sum(diag(tab.train2)) / sum(tab.train2)
accuracy.train2
```

```
## [1] 0.8525482
```

Model2 is better than Model1 with a 7% improvement in accuracy on the train data.

```r
model2$variable.importance
```

```
##      FICO      term    gender       age   purpose
## 705.56002 704.41182 263.22157 146.28035  15.00767
```

Let us see how both the models perform on the test data (? are we actually building a better model or leading to overfitting)

Model Evaluation

MODEL 1

```r
# Predicting on the test dataset using MODEL 1
test_predict.class1 <- predict(model1, test, type="class") # Predicted Classes
test_predict.score1 <- predict(model1, test) # Predicted Probabilities

# Create confusion matrix for test data predictions (using MODEL 1)
tab.test1 = table(test$isDelinquent, test_predict.class1)
tab.test1
```

```
##    test_predict.class1
##        0    1
##   0  671  477
##   1  313 2003
```

```r
# Accuracy on train data (MODEL 1 predictions)
accuracy.test1 = sum(diag(tab.test1)) / sum(tab.test1)
accuracy.test1
```

```
## [1] 0.77194
```

MODEL 2

```r
# Predicting on the test dataset using MODEL 2
test_predict.class2 <- predict(model2, test, type="class") # Predicted Classes
test_predict.score2 <- predict(model2, test) # Predicted Probabilities

# Create confusion matrix for test data predictions (using MODEL 2)
```

```
tab.test2 = table(test$isDelinquent, test_predict.class2)
tab.test2
```

```
##    test_predict.class2
##        0    1
##   0  830  318
##   1  211 2105
```

```
# Accuracy on train data (MODEL 2 predictions)
accuracy.test2 = sum(diag(tab.test2)) / sum(tab.test2)
accuracy.test2
```

```
## [1] 0.8472864
```

Comparing Models

```
Model_Name = c("Baseline", "Model1", "Model2")
Train_Accuracy_perc = c(66, accuracy.train1*100, accuracy.train2*100)
Test_Accuracy_perc = c(66, accuracy.test1*100, accuracy.test2*100)
output = data.frame(Model_Name,Train_Accuracy_perc,Test_Accuracy_perc)
output
```

```
##   Model_Name Train_Accuracy_perc Test_Accuracy_perc
## 1   Baseline            66.00000           66.00000
## 2     Model1            78.17912           77.19400
## 3     Model2            85.25482           84.72864
```

Model2 performs very well both on the test and train data. We will use model2 as the final model.

Conclusion

- Decision Tree Model has 19% more accuracy than baseline model
- Accuracy on the Training Data: 85%
- Accuracy on the Test Data: 84%
- Accuracy for test data is almost inline with training data.
- This tells that the model is neither underfit nor overfit.

Business Insights

- FICO, term and gender (in that order) are the most important variables in determining if a borrower will get into a delinquent stage
- No borrower shall be given a loan if they are applying for a 36 month term loan and have a FICO score in the range 300-500.
- Female borrowers with a FICO score greater than 500 should be our target customers.
- The decision tree model has significant improvement over baseline in terms of accuracy.
- The model is 19% more accurate in identifying delinquent and non-delinquent customers and shall help us set better business rules.

Things to try

- Try tuning the model using the diffrent control parameters, including cp to see if you can improve it.
- Explore if Accuracy is the best performance metric for such a model (considering the business objective).