Credit Risk Model

## 1. Problem Statement

To build a most optimal statistical model for creditworthiness based on the given dataset. To achieve this we would be following the below approach:

Data Loading => Data Exploration => Data Model Creation => Evaluation => Conclusion

## 2. Loading the Data

Loading relevant packages for the analysis

library(ggplot2) # For visualization  
library(gmodels) # For Cross tabulation function  
library(dplyr) # For case\_when function  
library(vcd) # For calculating kappa statistic  
library(randomForest) # For building the randomforest model

Reading the data into a data frame.

*Note: As the name of the first column of the data has the character "#" in it, hence we need to set the "comment.char" attribute of the read.table function as empty space instead of default #.*

crm <- read.table("ModelingData.txt", header = TRUE, comment.char = "")

## 3. Exploratory Data Analysis and Data Cleansing

**Step 1:** Verifying the loaded data to ensure whether the data types mapped are in line with that in the Modeling Code book.

str(crm)

## 'data.frame': 1000 obs. of 32 variables:  
## $ OBS. : int 1 2 3 4 5 6 7 8 9 10 ...  
## $ CHK\_ACCT : int 0 1 3 0 0 3 3 1 3 1 ...  
## $ DURATION : int 6 48 12 42 24 36 24 36 12 30 ...  
## $ HISTORY : int 4 2 4 2 3 2 2 2 2 4 ...  
## $ NEW\_CAR : int 0 0 0 0 1 0 0 0 0 1 ...  
## $ USED\_CAR : int 0 0 0 0 0 0 0 1 0 0 ...  
## $ FURNITURE : int 0 0 0 1 0 0 1 0 0 0 ...  
## $ RADIO.TV : int 1 1 0 0 0 0 0 0 1 0 ...  
## $ EDUCATION : int 0 0 1 0 0 1 0 0 0 0 ...  
## $ RETRAINING : int 0 0 0 0 0 0 0 0 0 0 ...  
## $ AMOUNT : int 1169 5951 2096 7882 4870 9055 2835 6948 3059 5234 ...  
## $ SAV\_ACCT : int 4 0 0 0 0 4 2 0 3 0 ...  
## $ EMPLOYMENT : int 4 2 3 3 2 2 4 2 3 0 ...  
## $ INSTALL\_RATE : int 4 2 2 2 3 2 3 2 2 4 ...  
## $ MALE\_DIV : int 0 0 0 0 0 0 0 0 1 0 ...  
## $ MALE\_SINGLE : int 1 0 1 1 1 1 1 1 0 0 ...  
## $ MALE\_MAR\_or\_WID : int 0 0 0 0 0 0 0 0 0 1 ...  
## $ CO.APPLICANT : int 0 0 0 0 0 0 0 0 0 0 ...  
## $ GUARANTOR : int 0 0 0 1 0 0 0 0 0 0 ...  
## $ PRESENT\_RESIDENT: int 4 2 3 4 4 4 4 2 4 2 ...  
## $ REAL\_ESTATE : int 1 1 1 0 0 0 0 0 1 0 ...  
## $ PROP\_UNKN\_NONE : int 0 0 0 0 1 1 0 0 0 0 ...  
## $ AGE : int 67 22 49 45 53 35 53 35 61 28 ...  
## $ OTHER\_INSTALL : int 0 0 0 0 0 0 0 0 0 0 ...  
## $ RENT : int 0 0 0 0 0 0 0 1 0 0 ...  
## $ OWN\_RES : int 1 1 1 0 0 0 1 0 1 1 ...  
## $ NUM\_CREDITS : int 2 1 1 1 2 1 1 1 1 2 ...  
## $ JOB : int 2 2 1 2 2 1 2 3 1 3 ...  
## $ NUM\_DEPENDENTS : int 1 1 2 2 2 2 1 1 1 1 ...  
## $ TELEPHONE : int 1 0 0 0 0 1 0 1 0 0 ...  
## $ FOREIGN : int 0 0 0 0 0 0 0 0 0 0 ...  
## $ RESPONSE : int 1 0 1 1 0 1 1 1 1 0 ...

**Step 2:** Fixing the loaded variable data types

The read.table function has loaded all the variables as class integer. Based on the codebook all but OBS#,DURATION,AMOUNT,INSTALL\_RATE,AGE,NUM\_CREDITS and NUM\_DEPENDENTS are categorical variables. So we need to convert all of these to factors.

crm[,-c(1,3,11,14,23,27,29)] <- as.data.frame(lapply(crm[,-c(1,3,11,14,23,27,29)],as.factor))

**Step 3:** Removing unique identifiers

The OBS# variable identifies each records uniquely and as such doesnt provide any useful information and hence can be excluded. Also the levels of the Response variable can be encoded as "Yes" for 1 and "No" for 0

crm <- crm[-1]  
levels(crm$RESPONSE) <- c("No","Yes")

**Step 4:** Exploratory data analysis

Lets look at some of the obvious suspects that may result in bad credit ratings.

prop.table(table(crm$CHK\_ACCT))\*100

##   
## 0 1 2 3   
## 27.4 26.9 6.3 39.4

Around 27% records have less than 0 DM in their checking account while around 39% records dont have a checking account.

prop.table(table(crm$HISTORY))\*100

##   
## 0 1 2 3 4   
## 4.0 4.9 53.0 8.8 29.3

There are around 29% records with critical accounts, almost 9% with delayed payments while 4% records havent taken any credits.

summary(crm$DURATION)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 4.0 12.0 18.0 20.9 24.0 72.0

The duration ranges from 4 months to 72 months with a median of 18 months

prop.table(table(crm$SAV\_ACCT))\*100

##   
## 0 1 2 3 4   
## 60.3 10.3 6.3 4.8 18.3

Around 60% records have less than 100 DM on an average in their savings account while around 18% records have unknown/no savings account

summary(crm$AMOUNT)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 250 1366 2320 3271 3972 18420

The loan amount ranges from 250 DM to 18420 DM with a median of 2320 DM

We can see that 30% of the records have bad credit rating

prop.table(table(crm$RESPONSE))\*100

##   
## No Yes   
## 30 70

It is interesting to note that the data has detailed relationship data about the male applicants but not for the female/other ones. We can combine all the relationship variable for males and create a new variable called **APPLICANT** having multiple levels based on the relationship status of the males and the remaining ones coded as OTHER.

crm$APPLICANT <-factor(case\_when(  
 crm$MALE\_DIV==1 ~ "MALE\_DIV",  
 crm$MALE\_MAR\_or\_WID==1 ~ "MALE\_MAR\_or\_WD",  
 crm$MALE\_SINGLE==1 ~ "MALE\_SINGLE",  
 TRUE ~ "OTHER"),labels = c("MD","MWD","MS","OTH")  
)

Based on this newly created variable, lets look at the credit worthiness of each of the levels.

addmargins(xtabs(~crm$APPLICANT+crm$RESPONSE))

## crm$RESPONSE  
## crm$APPLICANT No Yes Sum  
## MD 20 30 50  
## MWD 25 67 92  
## MS 146 402 548  
## OTH 109 201 310  
## Sum 300 700 1000

Similarly, lets create a combined variable **PURPOSE** for the purpose of the credit.

crm$PURPOSE <- factor(case\_when(  
 crm$NEW\_CAR==1 ~ "NEW\_CAR",  
 crm$USED\_CAR==1 ~ "USED\_CAR",  
 crm$FURNITURE==1 ~ "FURNITURE",  
 crm$RADIO.TV==1 ~ "RADIO.TV",  
 crm$EDUCATION==1 ~ "EDUCATION",  
 crm$RETRAINING==1 ~ "RETRAINING",  
 TRUE ~ "OTHER")  
)

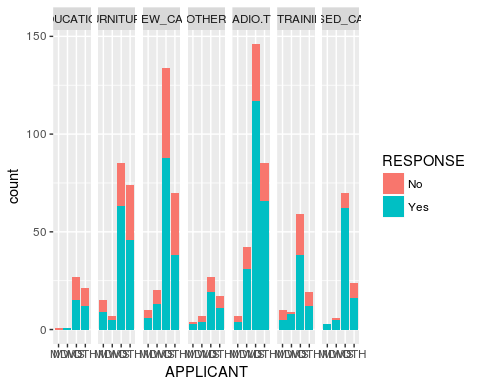
Based on this newly created variable, lets look at the credit worthiness of each of the levels.

addmargins(xtabs(~crm$PURPOSE+crm$RESPONSE))

## crm$RESPONSE  
## crm$PURPOSE No Yes Sum  
## EDUCATION 22 28 50  
## FURNITURE 58 123 181  
## NEW\_CAR 89 145 234  
## OTHER 18 37 55  
## RADIO.TV 62 218 280  
## RETRAINING 34 63 97  
## USED\_CAR 17 86 103  
## Sum 300 700 1000

We can plot these two newly created variable to visualise whether we see any relationship between purpose of credit, applicant type and their creditworthiness.

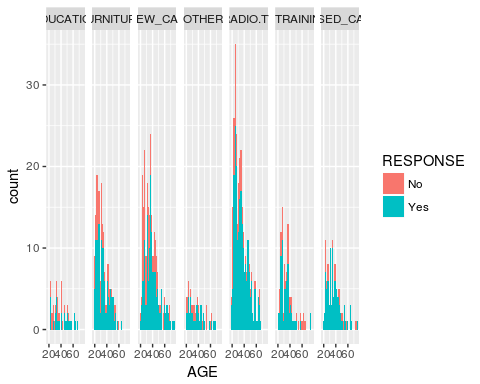
ggplot(crm[1:1000,], aes(APPLICANT, fill=RESPONSE))+geom\_bar()+facet\_grid(.~PURPOSE)



We can also plot the relationship between age, purpose of credit and their creditworthiness. We can observe that the older applicants have better creditworthiness.

ggplot(crm[1:1000,], aes(AGE, fill=RESPONSE))+geom\_histogram()+facet\_grid(.~PURPOSE)

## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.



Now with type of applicant. We can observe that the older married or widowed applicants have better creditworthiness.

**Step 5:** Creating training and test datasets

Lets create the training and test datasets. As the data is already randomized, we can directly create these datasets with a 90-10 percentage split. Also we will remove all the dependent variables due to creation of the new ones.

crm\_train <- crm[1:900,]  
crm\_train <- crm\_train[-c(4,5,6,7,8,9,14,15,16)]  
crm\_test <- crm[901:1000,]  
crm\_test <- crm\_test[-c(4,5,6,7,8,9,14,15,16)]

## 4. Creating the Data Model

**Step 1:** Selecting an appropriate data model

We are now ready to model the data. Because this is a supervised learning problem (i.e. the target variable is known) and a classification one at that (as we are predicting a categorical variable), we can use **decision trees algorithm** to build our model. Essentially, the algorithm would be splitting the data at each decision "node" until it reaches the final outcome, like a flowchart.

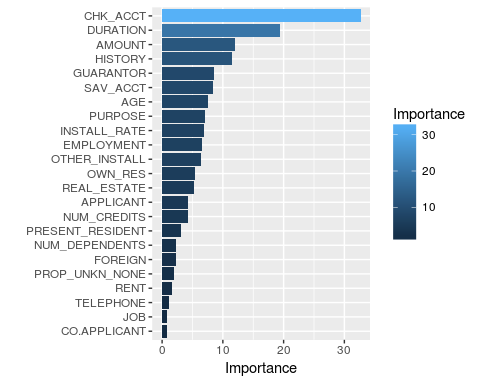
We also need to be careful about the kind of errors that are more "acceptable" than others. As we are predicting credit worthiness of an applicant, a **Type 2 error** (i.e. predicting that the applicant has a good credit rating when in fact s/he doesnt or "False Negative") is more costly than a **Type 1 error** (i.e. predicting that the applicant has a bad credit rating when in fact s/he doesnt or "False Positive"). To reduce Type 2 errors and to improve on normal decision tree performance, we will be using the **randomforest** function to built the data on the training data set. As the RESPONSE variable is also included in the training dataset, we need to exclude it while building the data model.

set.seed(123)  
crm\_model <- randomForest(crm\_train[-22],crm\_train$RESPONSE,importance = TRUE)

**Step 2:** Model visualization

We can visualize the relative importance of the contributing variables. Interestingly, the DURATION and AMOUNT variable have turned out to be more important than AGE and other newly created variables.

imp <- importance(crm\_model,type=1)  
fimp <- data.frame(Feature=row.names(imp), Importance=imp[,1])  
ggplot(fimp, aes(x=reorder(Feature, Importance), y=Importance, fill=Importance)) +  
 geom\_bar(stat="identity") +  
 coord\_flip()+  
 xlab("") +  
 ylab("Importance")



## 5. Evaluating Model Performance

Now we use the data model and test it on our test dataset.

crm\_predict <- predict(crm\_model,crm\_test)

To verify model performance, lets build the confusion matrix.

CrossTable(crm\_test$RESPONSE,crm\_predict, prop.chisq = FALSE, prop.r = FALSE, prop.c = FALSE,dnn = c("Actual","Predicted"))

##   
##   
## Cell Contents  
## |-------------------------|  
## | N |  
## | N / Table Total |  
## |-------------------------|  
##   
##   
## Total Observations in Table: 100   
##   
##   
## | Predicted   
## Actual | No | Yes | Row Total |   
## -------------|-----------|-----------|-----------|  
## No | 13 | 19 | 32 |   
## | 0.130 | 0.190 | |   
## -------------|-----------|-----------|-----------|  
## Yes | 5 | 63 | 68 |   
## | 0.050 | 0.630 | |   
## -------------|-----------|-----------|-----------|  
## Column Total | 18 | 82 | 100 |   
## -------------|-----------|-----------|-----------|  
##   
##

Kappa(table(crm\_test$RESPONSE,crm\_predict))

## value ASE z Pr(>|z|)  
## Unweighted 0.3763 0.09864 3.815 0.0001362  
## Weighted 0.3763 0.09864 3.815 0.0001362

## 6. Conclusion

With the accuracy at 76% (i.e. an error rate of 24%) and the unweighted kappa statistic 0.37, we can conclude that it indicates a fair agreement between the model's prediction and the true values.