CREDIT CARD FRAUD DETECTION

GROUP NO - 319

SHIVANI AGRAWAL | SARTHAK AGRAWAL

CONTENTS

BACKGROUND

DATA EXPLANATION

DATA PREPROCESSING

RESEARCH PROBLEMS

RESEARCH SOLUTIONS

RESULTS

CONCLUSION

BACKGROUND & INTRODUCTION

Credit Card is the most powerful and better way to use money to purchase things with a provided line by the credit card provider which requires a minimum monthly payment.

Best health for credit cards is measured accurately when we count the number of people who did not pay the bills rather than who did. Bad payments through credit cards

credit cards results in bad or lower credit reports and hence creates issues while getting loan disbursement or can even lead to the takeaway of vehicles or home..

In addition, this will help to make predictions for making such decisions for credit card providers to distinguish between fraud or legal transaction.

Fraud Detection Process

Terminal check

Correct Pin?
Sufficient Balance ?
Blocked Card?

Rejected

Predictive model

Fraud score

Investigators

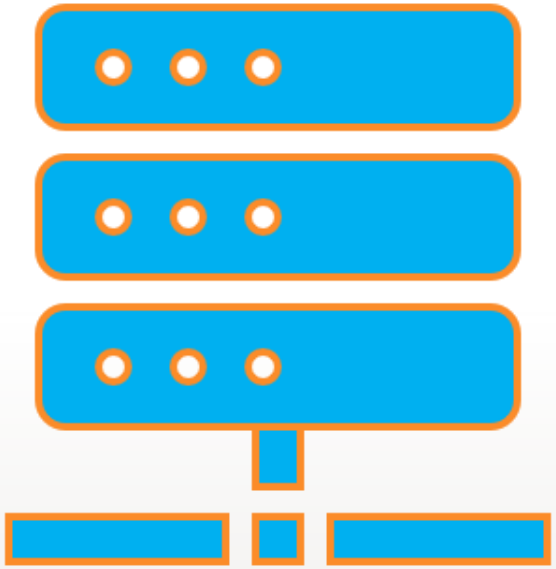
Feedback

RESEARCH PROBLEMS

Column	Description	Type
Time	Time in seconds taken between each transaction	Numeric
V1-V28	Unknown	Numeric
Amount	Money used for particular transaction	Numeric
Class	Fraud or Legal	Boolean

DATA EXPLANATION

- The dataset used here is taken from Kaggle having more than 300,000 rows with 31 different attributes mentioned in columns.
- This dataset has been taken as it contains only numerical input variables which are the result of a PCA transformation. Out of 31 columns 28 columns attribute is unknown for us which are obtained from PCA transformation whereas Time and Amount columns are not obtained from PCA transformation.



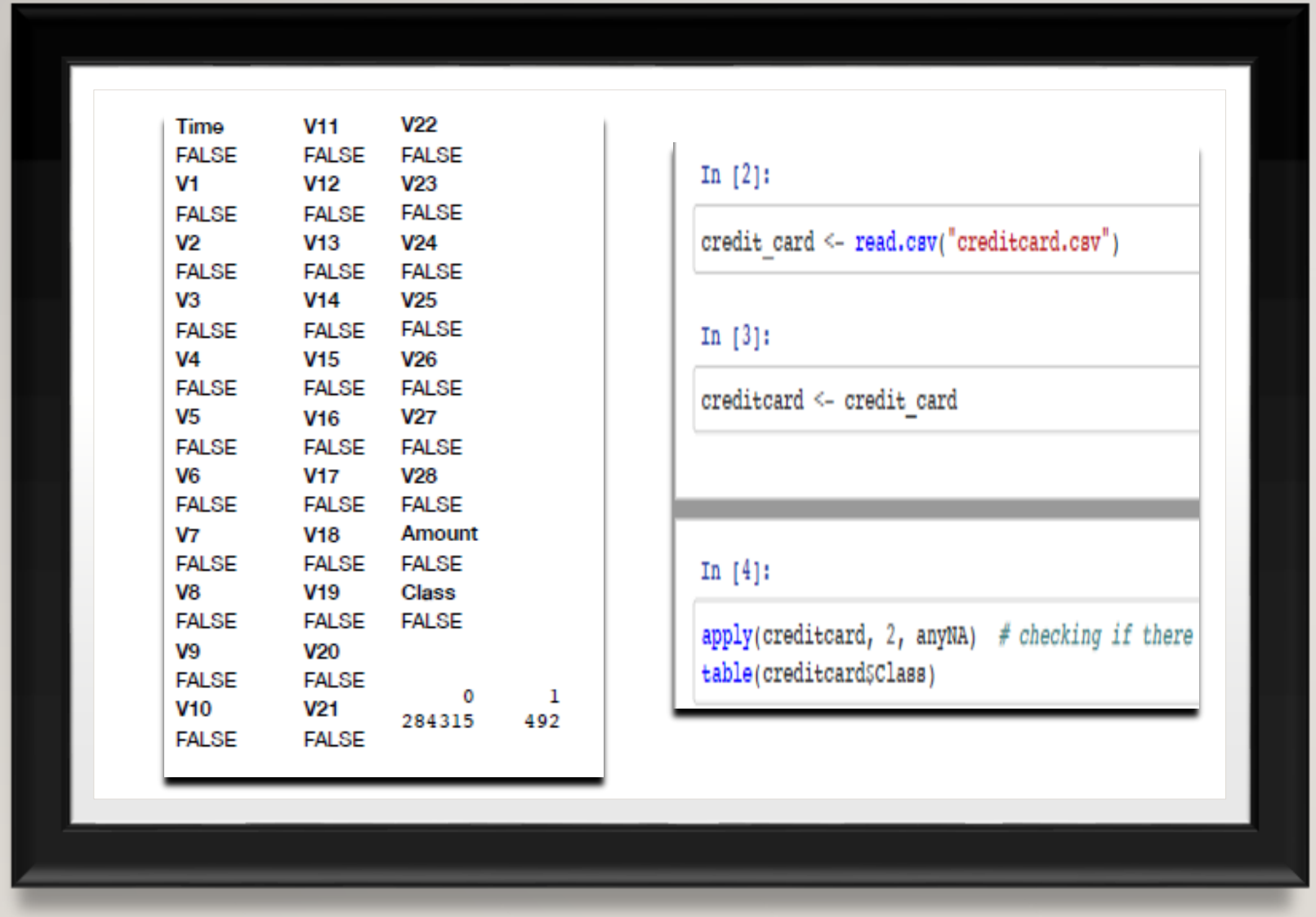
DATA PREPROCESSING

```
Type 'citation("pROC")' for a citation.

Attaching package: 'pROC'

The following objects are masked from 'package:stats':

    cov, smooth, var

Loaded gbm 2.1.8
randomForest 4.6-14
Type rfNews() to see new features/changes/bug fixes.
Loading required package: lattice
Loading required package: ggplot2
Registered S3 methods overwritten by 'ggplot2':
  method         from
  [.quosures     rlang
  c.quosures     rlang
  print.quosures rlang

Attaching package: 'ggplot2'

The following object is masked from 'package:randomForest':

    margin

Loading required package: rpart
Loading required package: survival

Attaching package: 'survival'

The following object is masked from 'package:caret':

    cluster

Loading required package: Formula

Attaching package: 'Hmisc'

The following objects are masked from 'package:plyr':

    is.discrete, summarize

The following objects are masked from 'package:base':

    format.pval, units
```

In [1]:

```
library(pROC)
library(gbm)
library(randomForest)
library(caret)
library(readr)
library(rpart.plot)
library(caTools)
library(rpart)
library(plyr)
library(Hmisc)
```

LOADING OF LIBRARIES



LOADING OF DATASET & CHECKING IF ANY MISSING VALUES

In [5]:

```
#---------setting the seed----------#
set.seed(4495)
creditcard$Time <- NULL ##### removing the time variable
creditcard[is.na(creditcard)] = -9999
```

In [6]:

```
#---------- removing NA values ---------------#
replaceNAWithMean <- function(data) {
  for(i in 1:ncol(data)){
    data[is.na(data[,i]), i] <- mean(data[,i], na.rm = TRUE)
  }
}
replaceNAWithMean(creditcard)
```

In [7]:

```
#---------- creating partition ---------------#
set.seed(4495)
t<-createDataPartition(p=0.5,y=creditcard$Class,list = F)
training<-creditcard[t,]
testing<-creditcard[-t,]
```

In [8]:

```
table(training$Class)
table(testing$Class)
```

```
     0      1
142149    255

     0      1
142166    237
```

- SETTING THE SEED VALUE
- REMOVING ALL THE NA VALUES
- SPLITTING THE DATA
- CREATING TRAINING
- TESTING DATASET

METHODS USED

- LOGISTIC REGRESSION
- TREE BAG
- RANDOM FOREST
- DECISION TREE MODEL

```
#-------------------- Visualization --------------#
library(ggplot2)
Y <- creditcard$Class
Y <- as.factor(Y)
ggplot(creditcard,aes(x = Y)) + geom_bar(aes(fill = Y)) + xlab('Class')
training$Class <- as.factor(training$Class)
ggplot(training,aes(x = training$Class)) + geom_bar(aes(fill = training$Class)
) + xlab('Class')
(table(Y)[1]/length(Y))*100
(table(Y)[2]/length(Y))*100
```

VISUALIZATION GRAPH

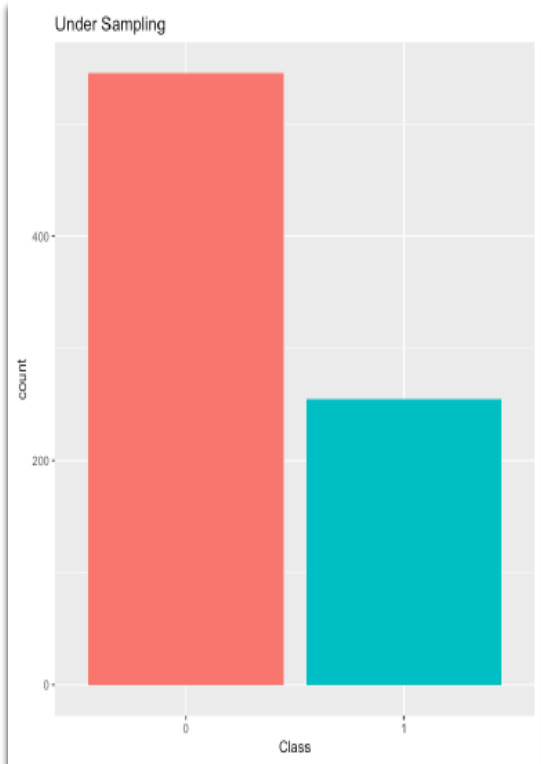
VISUALIZATION GRAPH

SYNTHETIC DATA CREATION

In [10]:

```
#---------------------- generating Synthetic data -------------------#
library(ROSE)
attach(training)
set.seed(4495)
training_Rose <- ROSE(Class~.,data=training,seed = 4495)$data
training_Rose$Class <- as.factor(training_Rose$Class)
ggplot(training_Rose,aes(x = Class)) + geom_bar(aes(fill = Class))
```

```
Loaded ROSE 0.0-3
```




```
In [11]:

## ------------------ Undersampling ----------------------
training <- na.omit(training)
attach(training)
training$Class <- as.factor(training$Class)
training_under <- ovun.sample(Class~.,data = training,method = "under",
                              N=800,seed=4495)$data
ggplot(training_under,aes(x = Class)) + geom_bar(aes(fill = Class))+ggtitle("U
nder Sampling")
```

```
The following objects are masked from training (pos = 3):

    Amount, Class, V1, V10, V11, V12, V13, V14, V15, V16, V17, V18
,
    V19, V2, V20, V21, V22, V23, V24, V25, V26, V27, V28, V3, V4,
V5,
    V6, V7, V8, V9
```

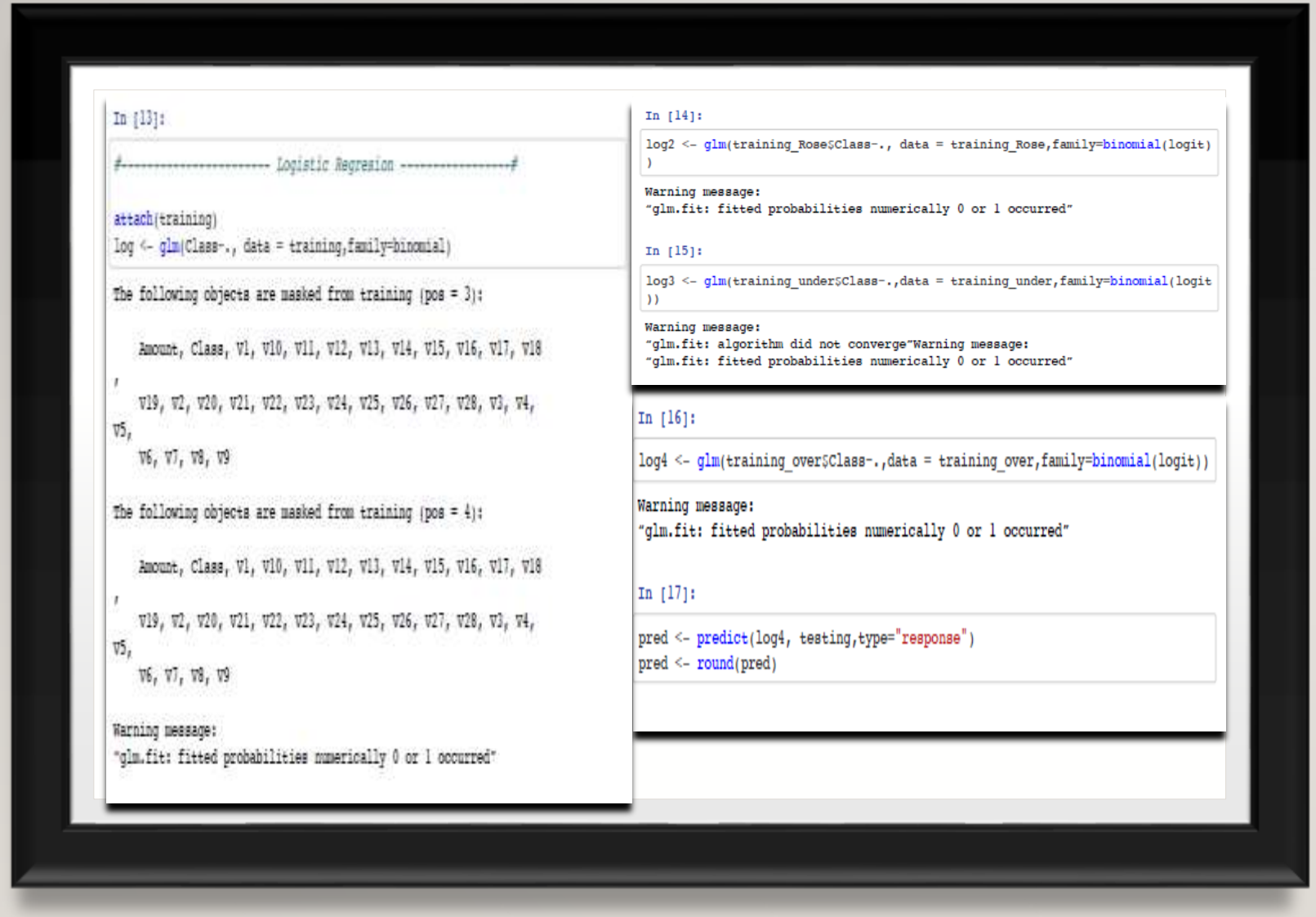
UNDERSAMPLING

In [12]:

```
## ------------------ oversampling ----------------------

training_over <- ovun.sample(Class~.,data = training,method = "over",
                             N=202404,seed=4495)$data
ggplot(training_over,aes(x = Class)) + geom_bar(aes(fill = Class))+ggtitle("Ov
er Sampling")
```

OVERSAMPLING

OVERSAMPLING



LOGISTIC REGRESSION AND ITS CONFUSION MATRIX WITH ROC

In [18]:

```
accuracy <- (1-mean(pred != testing$Class))*100
accuracy
```

99.073053236238

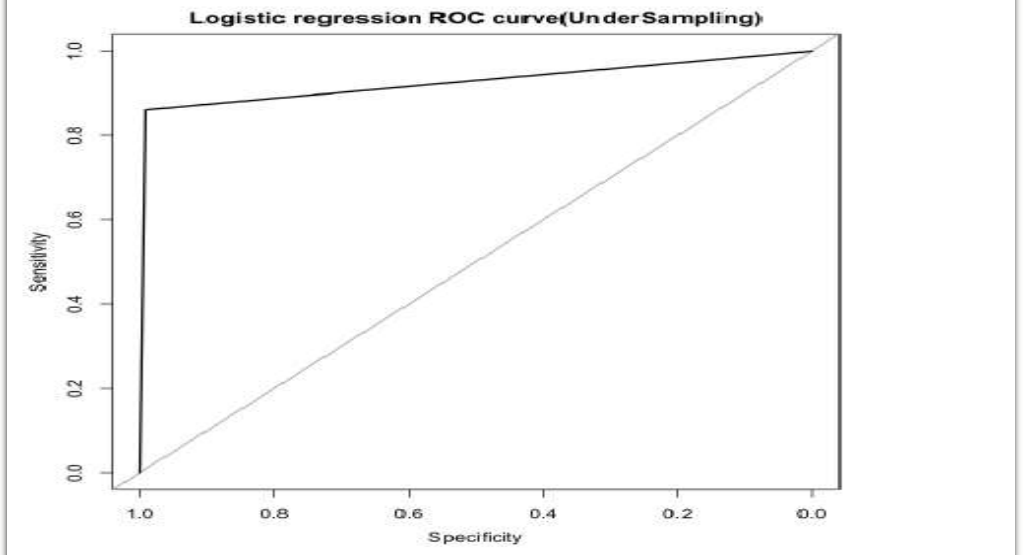
In [19]:

```
confusionMatrix(table(pred,testing$Class))
mat <- as.matrix(confusionMatrix(table(pred,testing$Class)))
```

```
Confusion Matrix and Statistics


pred      0      1
   0 140879     33
   1   1287    204

               Accuracy : 0.9907
                 95% CI : (0.9902, 0.9912)
    No Information Rate : 0.9983
    P-Value [Acc > NIR] : 1

                  Kappa : 0.2339

 Mcnemar's Test P-Value : <2e-16

            Sensitivity : 0.9909
            Specificity : 0.8608
         Pos Pred Value : 0.9998
         Neg Pred Value : 0.1368
             Prevalence : 0.9983
         Detection Rate : 0.9893
   Detection Prevalence : 0.9895
      Balanced Accuracy : 0.9259

       'Positive' Class : 0
```

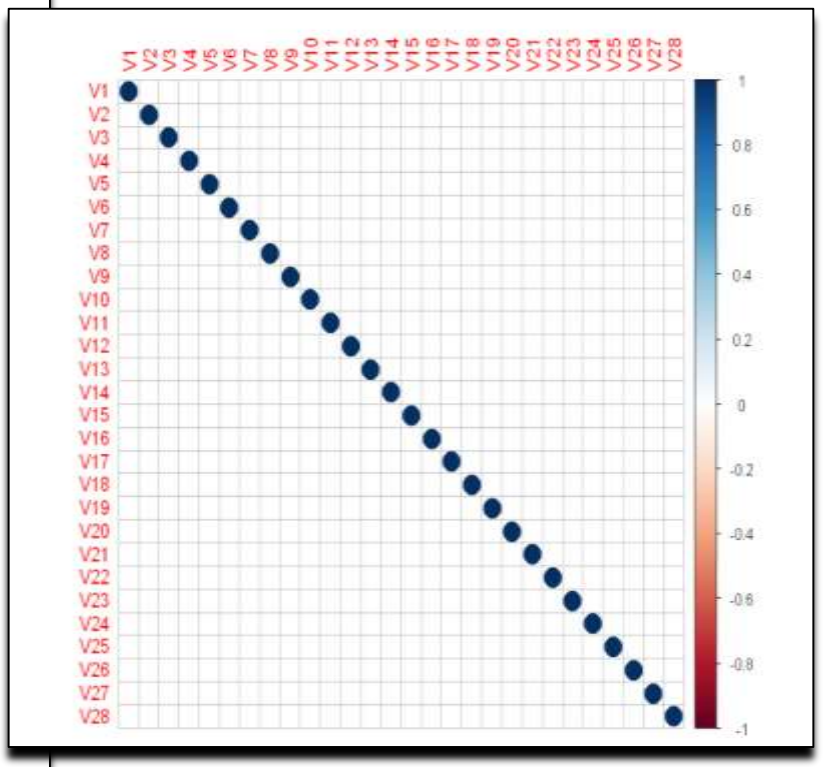
LOGISTIC REGRESSION AND ITS CONFUSION MATRIX WITH ROC

```
In [20]:

print(roc(testing$Class,pred))
plot(roc(testing$Class,pred),main = "Logistic regression ROC curve(UnderSampli
ng)")
Setting levels: control = 0, case = 1
Setting direction: controls < cases

Call:
roc.default(response = testing$Class, predictor = pred)

Data: pred in 142166 controls (testing$Class 0) < 237 cases (testi
ng$Class 1).
Area under the curve: 0.9259

Setting levels: control = 0, case = 1
Setting direction: controls < cases
```



LOGISTIC REGRESSION AND ITS CONFUSION MATRIX WITH ROC

	<dbl>		
V10	64393.638		
V11	55568.386		
V12	56527.482	V9	0.000
V14	67253.315	V13	0.000
V17	60349.214	V15	0.000
V20	1689.982	V16	0.000
V4	2176.235	V18	0.000
V1	0.000	V19	0.000
V2	0.000	V21	0.000
V3	0.000	V22	0.000
V5	0.000	V23	0.000
V6	0.000	V24	0.000
V7	0.000	V25	0.000
V8	0.000	V26	0.000
V9	0.000	V27	0.000
V13	0.000	V28	0.000
V15	0.000	Amount	0.000



CORRELATION MATRIX & VARIABLE IMPORTANCE

In [27]:

```
#------------------- RANDOM FOREST REGRESSION ---------------#
set.seed(4495)
library(e1071)
```

```
Attaching package: 'e1071'

The following object is masked from 'package:Hmisc':

    impute
```

In [28]:

```
library(randomForest)
```

RANDOM FOREST MODEL WITH CONFUSION MATRIX, ROC

In [29]:

```
set.seed(4495)
system.time(rand_model <- randomForest(Class~., data = training, ntree = 200))
```

```
   user  system elapsed
331.179   4.395 337.450
```

In [30]:

```
training$Class <- as.numeric(training$Class)
```

In [31]:

```
pred1 <- predict(rand_model,type = "prob")
```

In [32]:

```
library(ROCR)
perf <- prediction(pred1[,2], training$Class)
```

In [33]:

```
auc <- performance(perf, "auc")
```

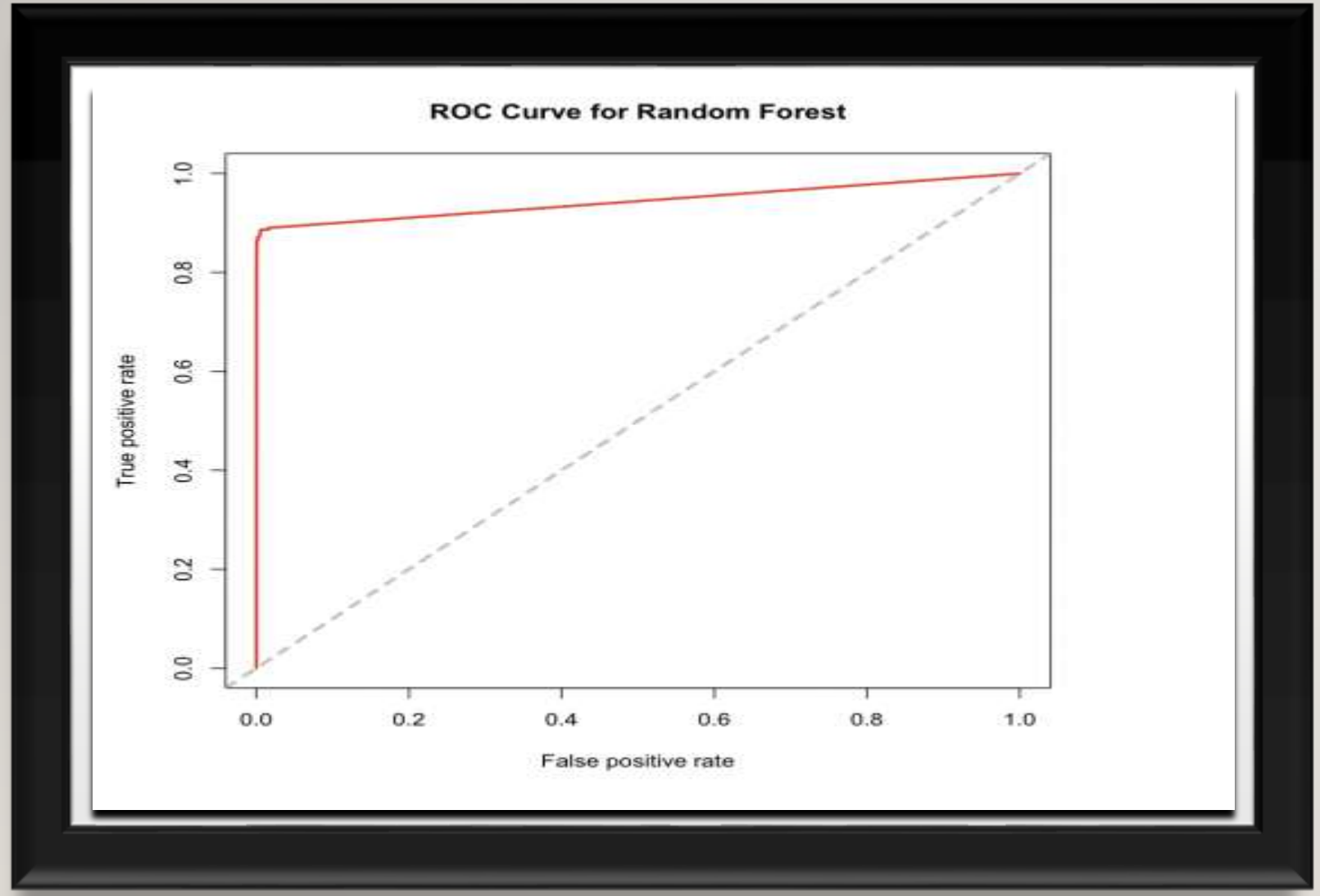
In [34]:

```
pred3 <- performance(perf, "tpr","fpr")
```

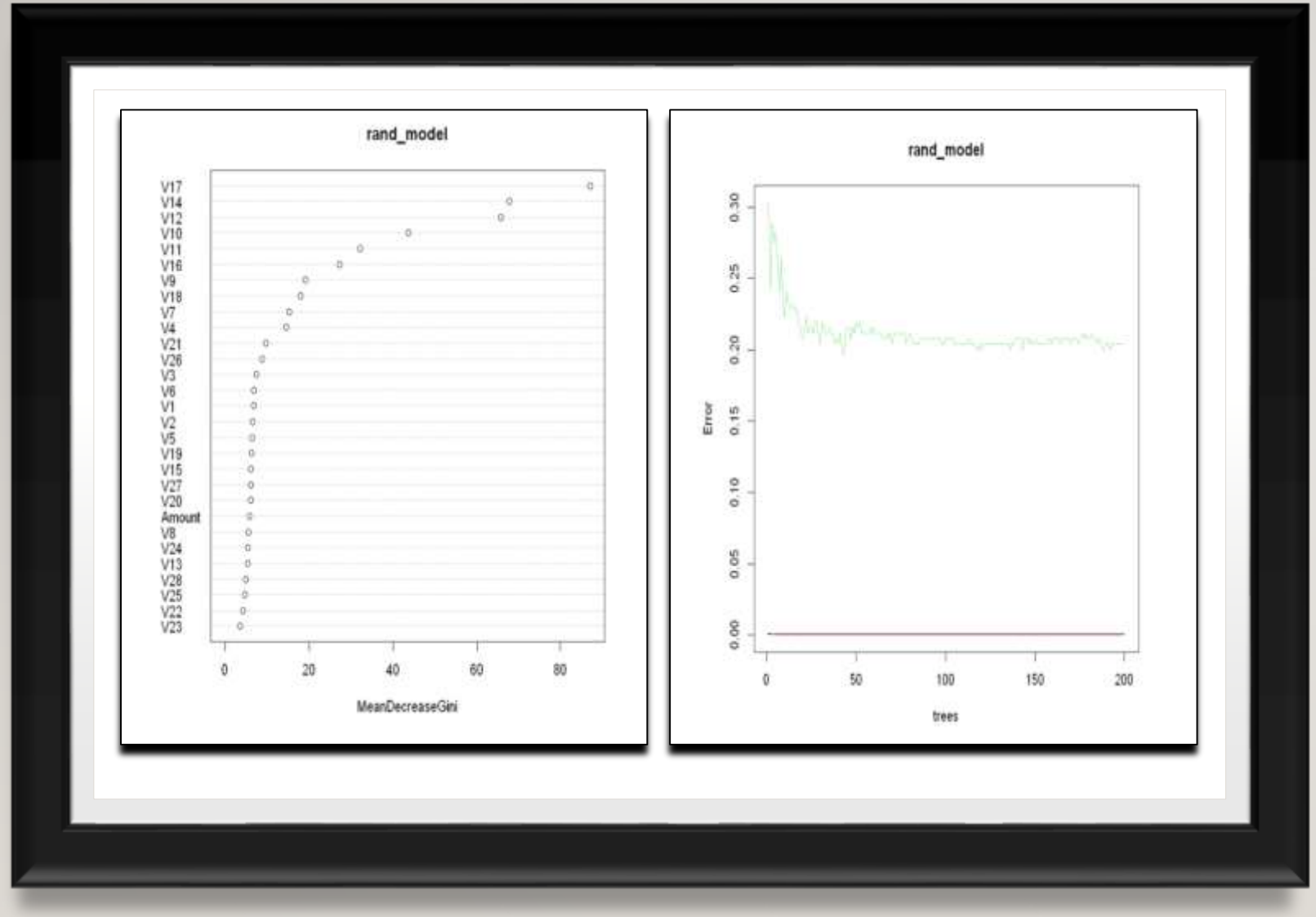
In [35]:

```
plot(pred3,main="ROC Curve for Random Forest",col=2,lwd=2)
abline(a=0,b=1,lwd=2,lty=2,col="gray")
```

RANDOM FOREST MODEL WITH CONFUSION MATRIX, ROC



RANDOM FOREST MODEL WITH CONFUSION MATRIX, ROC



VARIABLE IMPORTANCE & VISUALIZATION GRAPH

In [37]:

```
#--------------Decision tree model- ------------#
library(rpart)
set.seed(4495)
```

In [38]:

```
tree.model1 <- rpart(Class ~ ., data = training, method = "class", minbucket =
20)
```

In [39]:

```
tree.model2 <- rpart(training_Rose$Class ~ ., data = training, method = "class
", minbucket = 20)
```

In [40]:

```
tree.model3 <- rpart(training_under$Class ~ ., data = training_under, method =
"class", minbucket = 20)
```

DECISION TREE MODEL WITH CONFUSION MATRIX, ROC

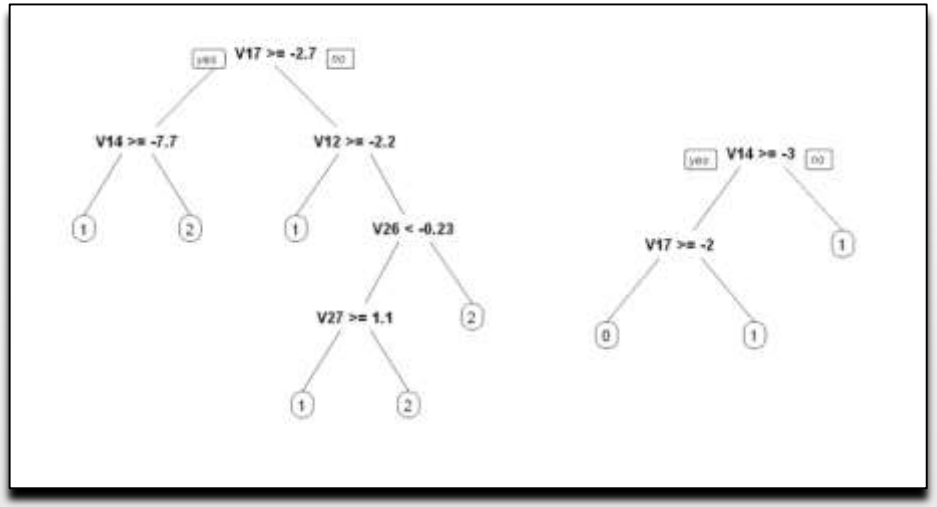
DECISION TREE MODEL WITH CONFUSION MATRIX, ROC

In [41]:

```
tree.model4 <- rpart(training_over$Class~.,data = training_over,method = "clas
s",minbucket = 20)
```

In [42]:

```
prp(tree.model)
prp(tree.model4)
```



In [43]:

```
tree.predict <- predict(tree.model4,testing,type = "class")
accuracy <-(1-mean(tree.predict != testing$Class))*100
```

In [44]:

```
accuracy
```

99.2212242719606

DECISION TREE MODEL WITH CONFUSION MATRIX, ROC

```
In [45]:
confusionMatrix(table(tree.predict,testing$Class))
mat <- as.matrix(confusionMatrix(table(tree.predict,testing$Class)))
Confusion Matrix and Statistics


tree.predict      0      1
           0 141097     40
           1   1069    197

               Accuracy : 0.9922
                 95% CI : (0.9917, 0.9927)
    No Information Rate : 0.9983
    P-Value [Acc > NIR] : 1

                  Kappa : 0.2601

 Mcnemar's Test P-Value : <2e-16

            Sensitivity : 0.9925
            Specificity : 0.8312
         Pos Pred Value : 0.9997
         Neg Pred Value : 0.1556
             Prevalence : 0.9983
         Detection Rate : 0.9908
   Detection Prevalence : 0.9911
      Balanced Accuracy : 0.9119

       'Positive' Class : 0
```

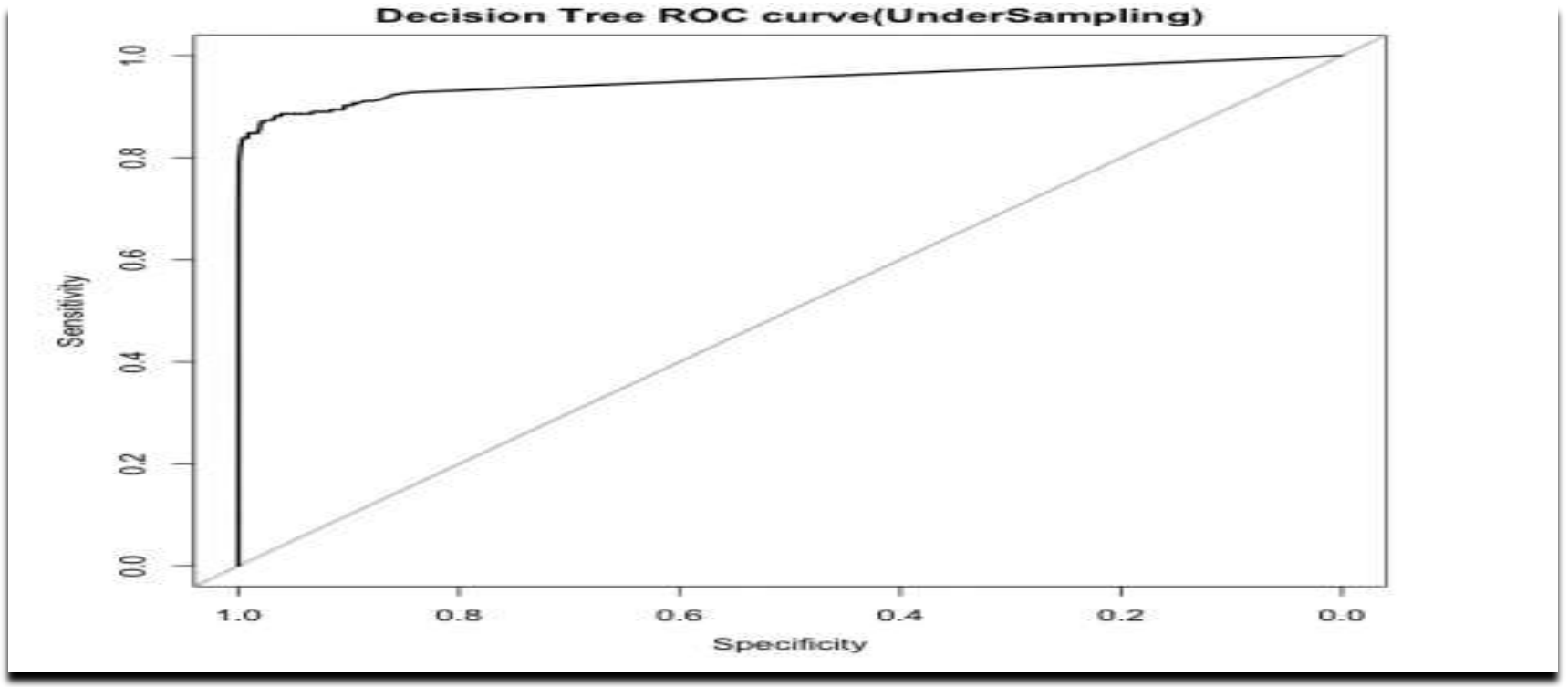
DECISION TREE MODEL WITH CONFUSION MATRIX, ROC

DECISION TREE MODEL WITH CONFUSION MATRIX, ROC

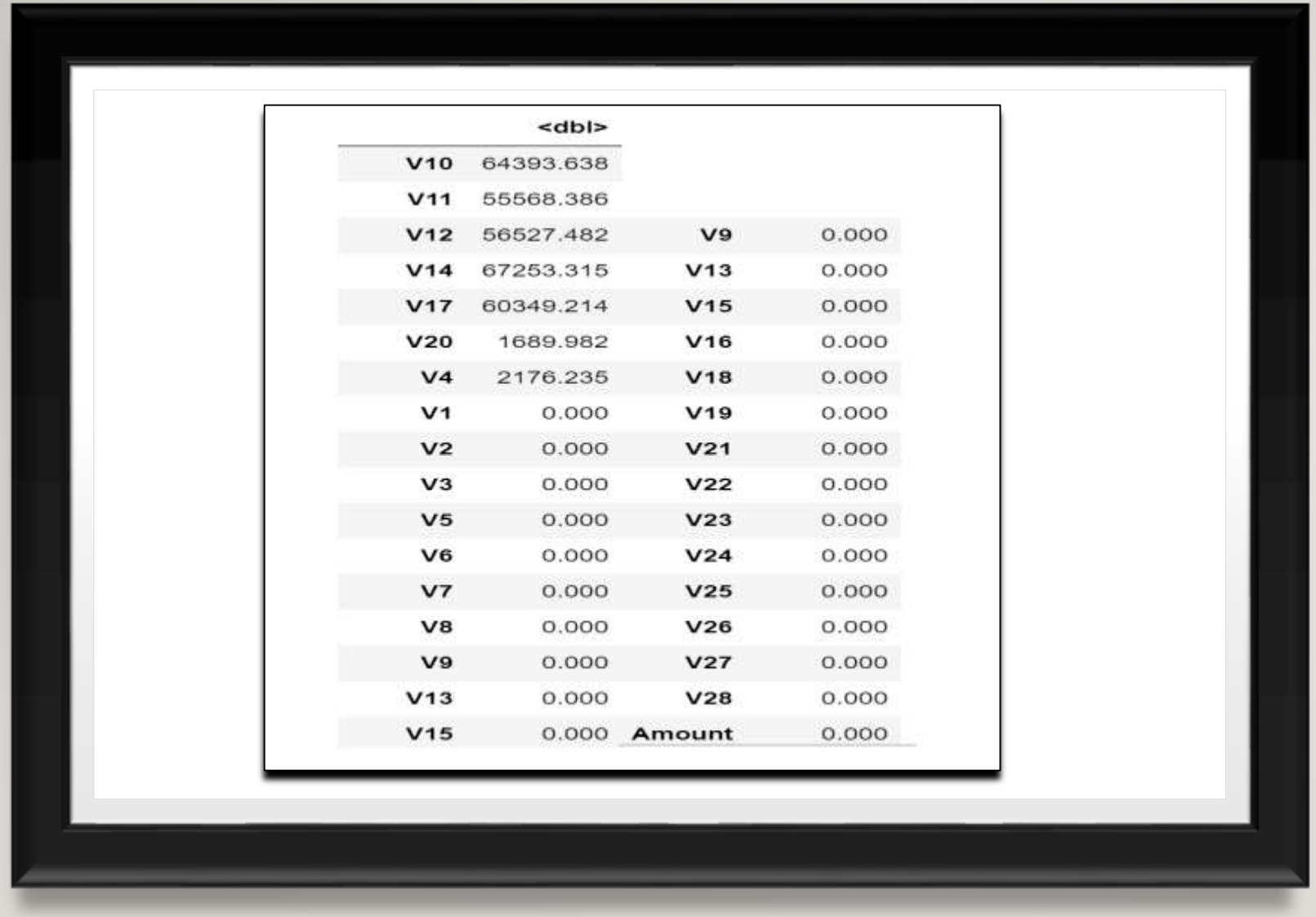
In [46]:

```
print(roc(testing$Class,pred))
plot(roc(testing$Class,pred),main = "Decision Tree ROC curve(UnderSampling)")
```

```
Setting levels: control = 0, case = 1
Setting direction: controls < cases

Call:
roc.default(response = testing$Class, predictor = pred)

Data: pred in 142166 controls (testing$Class 0) < 237 cases (testi
ng$Class 1).
Area under the curve: 0.9526

Setting levels: control = 0, case = 1
Setting direction: controls < cases
```

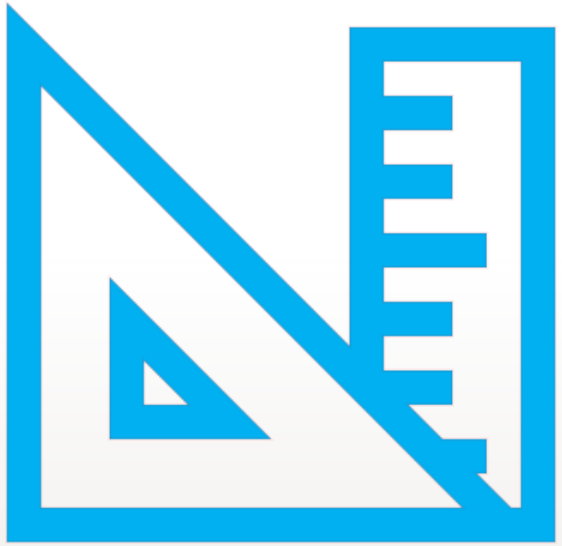


DECISION TREE MODEL WITH CONFUSION MATRIX, ROC



	<dbl>		
V10	64393.638		
V11	55568.386		
V12	56527.482	V9	0.000
V14	67253.315	V13	0.000
V17	60349.214	V15	0.000
V20	1689.982	V16	0.000
V4	2176.235	V18	0.000
V1	0.000	V19	0.000
V2	0.000	V21	0.000
V3	0.000	V22	0.000
V5	0.000	V23	0.000
V6	0.000	V24	0.000
V7	0.000	V25	0.000
V8	0.000	V26	0.000
V9	0.000	V27	0.000
V13	0.000	V28	0.000
V15	0.000	Amount	0.000

DECISION TREE VARIABLE IMPORTANCE



PREDICTIONS

AUC OF ALL THE MODELS

METHODS	ACCURACY	PRECITION	RECALL	AUROC
LOGISTIC REGRESSION	99.07	86.07	99.09	0.92
RANDOM FOREST	99.95	92.69	79.86	0.90
DECISION TREE	99.22	83.12	99.25	0.92

RESULTS AND CONCLUSION

With the above three models used, we can predict or say that the decision tree model has better area under graph and hence can be the most useful method to determine whether the transaction is fraud or legal with an accuracy of 99.22%.

FUTURE DEVELOPMENT

We have worked on four methods but for further development in the project we would like to work on few more models which are:

1. GBM (GRADIENT BASED ALGORITHM)
2. SVM (SUPPORT VECTOR MACHINE)
3. XGBoost
4. LIGHTGBM

THANK YOU..!!