Loan Approval Random Forest Analysis

Load Data

```
data<-read.csv('loan_data.csv')
data$Employment_Status<-as.factor(data$Employment_Status)
data$Approval<-as.factor(data$Approval)
str(data)
```

```
## ’data.frame’:    24000 obs. of  7 variables:
##  $ Text             : chr  "I need a loan to pay for an international vacation with my family." "I wa
##  $ Income           : int  26556 197392 44561 190363 61853 108236 110165 40656 38233 81024 ...
##  $ Credit_Score     : int  581 389 523 729 732 404 570 600 346 403 ...
##  $ Loan_Amount      : int  8314 111604 34118 118757 19210 50797 61217 21267 8467 19217 ...
##  $ DTI_Ratio        : num  79.3 22.1 45.4 10.2 44.1 ...
##  $ Employment_Status: Factor w/ 2 levels "employed","unemployed": 1 1 1 2 1 1 1 2 2 2 ...
##  $ Approval         : Factor w/ 2 levels "Approved","Rejected": 2 2 2 2 1 2 1 2 2 2 ...
```

Train-Test Split

```
set.seed(100)
splitIndex <- createDataPartition(data$Approval, p = 0.8, list = FALSE)
train_data <- data[splitIndex, ]
test_data  <- data[-splitIndex, ]
```

Train Random Forest

```
bag.res<-randomForest(Approval~Income+Credit_Score+Loan_Amount+DTI_Ratio+Employment_Status,
data=train_data,xtest=test_data[,2:6],ytest=test_data[,7],mtry=5,ntree=1000,importance=T)
rf.res<- randomForest(Approval~Income+Credit_Score+Loan_Amount+DTI_Ratio+Employment_Status,
data=train_data,xtest=test_data[,2:6],ytest=test_data[,7],mtry=2,ntree=1000,importance=T)
names(rf.res)
```

```
##  [1] "call"            "type"            "predicted"       "err.rate"
##  [5] "confusion"       "votes"           "oob.times"       "classes"
##  [9] "importance"      "importanceSD"    "localImportance" "proximity"
## [13] "ntree"           "mtry"            "forest"          "y"
## [17] "test"            "inbag"           "terms"
```

```r
names(rf.res$test)
```

```
## [1] "predicted" "err.rate"  "confusion" "votes"     "proximity"
```

Confusion Matrix

```r
conf.rf<-rf.res$test$confusion
conf.rf
```

```
##          Approved Rejected class.error
## Approved      776       10 0.012722646
## Rejected        7     4006 0.001744331
```

```r
# Overall Accuracy
sum(diag(conf.rf[,1:2])) / sum(conf.rf[,1:2])
```

```
## [1] 0.9964576
```

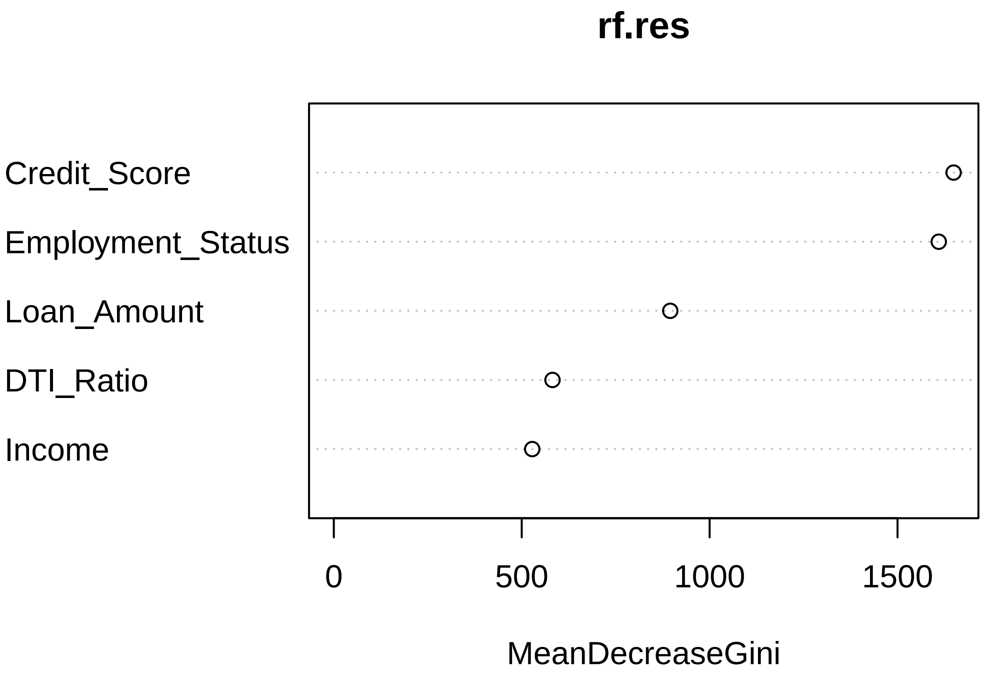
```r
# Sensitivity
conf.rf[1,1] / sum(conf.rf[1, ])
```

```
## [1] 0.9872614
```

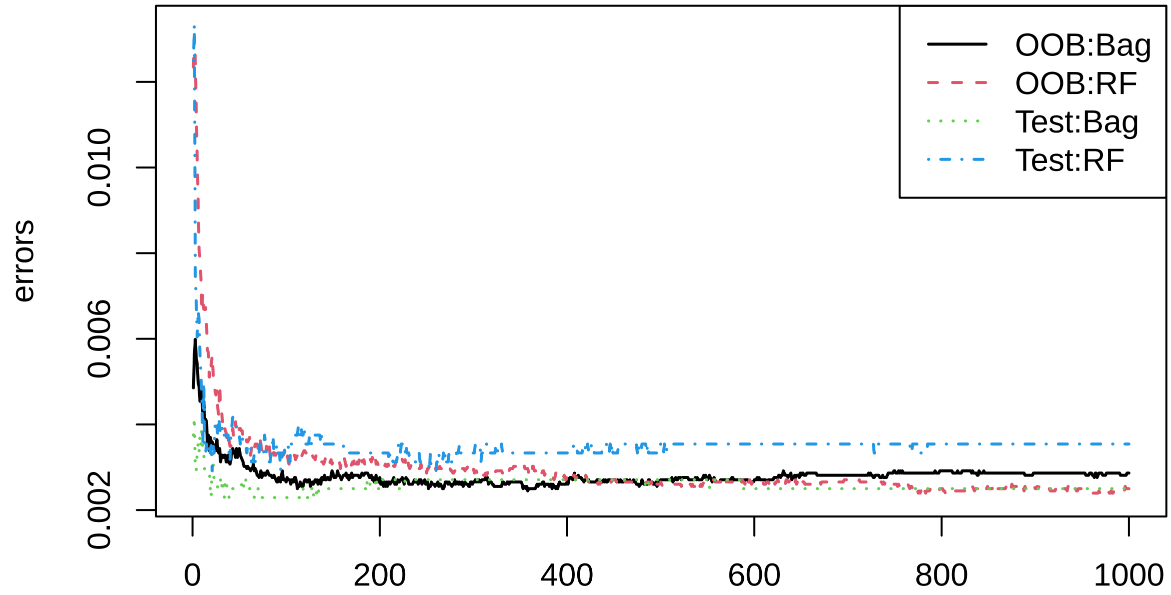
```r
# Specificity
conf.rf[2,2] / sum(conf.rf[2, ])
```

```
## [1] 0.9982552
```

Variable Importance Plot



Error Rate Plot



Conclusion

- Random Forest achieved high accuracy
- Most important variables: Credit Score, Employment Status
- Very low error rates