Predicting Systemic Crisis in Africa

Derrick Owusu Ofori

25/10/2020

Contents

Introduction

A systemic crisis or systemic risk is a domino effect in which financial troubles spread between institutions and markets until it affects the whole monetary and financial system with dire economic consequences.

With the COVID-19 pandemic upon us, many countries in Africa are faced with financial and economic instability. The ominous impact of this pandemic inspired me to undertake this project with the goals of developing a model that would predict the occurrence of a systemic crisis, as well as uncovering how some economic indicators significantly contribute to it.

To achieve this, I will utilize a dataset derived from Reinhart et. al's Global Financial Stability dataset which can be found online at https://www.hbs.edu/behavioral-finance-and-financial-stability/data/Pages/global.aspx.

The derived data set focuses on the Banking, Debt, Financial, Inflation, and Systemic Crises that occurred, from 1860 to 2014, in 13 African countries, namely: Algeria, Angola, Central African Republic, Ivory Coast, Egypt, Kenya, Mauritius, Morocco, Nigeria, South Africa, Tunisia, Zambia, and Zimbabwe.

The variable names and descriptions can be found below.

Variable Name	Description
case	A number which denotes a specific country
cc3	A three letter country code
country	The name of the country
year	The year of the observation
systemic_crisis	Systemic crisis (1) or no systemic crisis (0)
exch_usd	The exchange rate of the country vis-a-vis the USD
domestic_debt_in_default	Domestic debt default (1) or no domestic debt default (0)
sovereign_external_debt_default	Sovereign external debt (1) or no sovereign external debt (0)
gdp_weighted_default	The total debt in default vis-a-vis the GDP
inflation_annual_cpi	The annual CPI Inflation rate
independence	Independence (1) or no independence (0)
currency_crises	Currency crisis (1) or no currency crisis (0)
inflation_crises	Inflation (1) or no inflation (0)
banking_crisis	Banking Crisis (crisis) or no banking crisis (no_crisis)

Once I have the data imported, I will preprocess and clean it, perform some exploratory data analysis, correct class imbalances, determine the best model to make predictions, and find the most important variables associated with a systemic crisis.

Methods

Installing packages and Importing data

```r
#Load Packages and libraries
if (!require(tidyverse)) install.packages('tidyverse') #Data manipulation and visuals
library(tidyverse)
if (!require(caret)) install.packages('caret')#Classification and regression
library(caret)
if (!require(ROSE)) install.packages('ROSE')# Package for Binary Imbalance
library(ROSE)
```

```
if (!require(randomForest)) install.packages('randomForest')# Classification algorithm
library(randomForest)
if (!require(e1071)) install.packages('e1071')#SVM, Naive Bayes, Parameter Tuning
library(e1071)
if (!require(knitr)) install.packages('knitr')# Dynamic Report Generation
library(knitr)
if (!require(car)) install.packages('car') # Checking Multicollinearity
library(car)
if (!require(ROCR)) install.packages('ROCR') # Prediction: ROC Curve
library(ROCR)
if (!require(rpart)) install.packages('rpart') # Prediction: Decision Tree
library(rpart)
if (!require(rpart.plot)) install.packages('rpart.plot')# Plot Decision Tree
library(rpart.plot)
```

```
#Importing data
url<-'https://raw.githubusercontent.com/Derrick015/HarvardX/master/african_crises.csv'
african_crises<-read.csv(url)

#Structure
str(african_crises)
```

```
## 'data.frame':    1059 obs. of  14 variables:
##  $ case                           : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ cc3                            : chr  "DZA" "DZA" "DZA" "DZA" ...
##  $ country                        : chr  "Algeria" "Algeria" "Algeria" "Algeria" ...
##  $ year                           : int  1870 1871 1872 1873 1874 1875 1876 1877 1878 1879 ...
##  $ systemic_crisis                : int  1 0 0 0 0 0 0 0 0 0 ...
##  $ exch_usd                       : num  0.0523 0.0528 0.0523 0.0517 0.0513 ...
##  $ domestic_debt_in_default       : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ sovereign_external_debt_default: int  0 0 0 0 0 0 0 0 0 0 ...
##  $ gdp_weighted_default           : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ inflation_annual_cpi           : num  3.44 14.15 -3.72 11.2 -3.85 ...
##  $ independence                   : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ currency_crises                : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ inflation_crises               : int  0 0 0 0 0 0 0 1 0 0 ...
##  $ banking_crisis                 : chr  "crisis" "no_crisis" "no_crisis" "no_crisis" ...
```

Data preprocessing and cleaning

After evaluating the data structure I realize several categorical variables are in numeric and integer form. These will be converted to factors.

```
# Convert categorical variables from numeric and integers to factors
african_crises$systemic_crisis<-as.factor(african_crises$systemic_crisis)
african_crises$domestic_debt_in_default<-as.factor(african_crises$domestic_debt_in_default)
african_crises$sovereign_external_debt_default<-as.factor(african_crises$sovereign_external_debt_defaul
african_crises$independence<-as.factor(african_crises$independence)
african_crises$inflation_crises<-as.factor(african_crises$inflation_crises)
african_crises$banking_crisis<-as.factor(african_crises$banking_crisis)
african_crises$currency_crises<-as.factor(african_crises$currency_crises)
```

```r
str(african_crises)
```

```
## 'data.frame':    1059 obs. of  14 variables:
##  $ case                           : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ cc3                            : chr  "DZA" "DZA" "DZA" "DZA" ...
##  $ country                        : chr  "Algeria" "Algeria" "Algeria" "Algeria" ...
##  $ year                           : int  1870 1871 1872 1873 1874 1875 1876 1877 1878 1879 ...
##  $ systemic_crisis                : Factor w/ 2 levels "0","1": 2 1 1 1 1 1 1 1 1 1 ...
##  $ exch_usd                       : num  0.0523 0.0528 0.0523 0.0517 0.0513 ...
##  $ domestic_debt_in_default       : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ sovereign_external_debt_default: Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ gdp_weighted_default           : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ inflation_annual_cpi           : num  3.44 14.15 -3.72 11.2 -3.85 ...
##  $ independence                   : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ currency_crises                : Factor w/ 3 levels "0","1","2": 1 1 1 1 1 1 1 1 1 1 ...
##  $ inflation_crises               : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 2 1 1 ...
##  $ banking_crisis                 : Factor w/ 2 levels "crisis","no_crisis": 1 2 2 2 2 2 2 2 2 2 ...
```

The currency crisis variable has an extra level of 2 which is erroneous. This will be converted into level 1 implying the presence of a currency crisis while 0 will indicate no currency crisis.

```r
#Replacing erronous level 2 with 1 in currency crisis column
african_crises$currency_crises[african_crises$currency_crises==2]<-1
levels(african_crises$currency_crises)
```

```
## [1] "0" "1" "2"
```

```r
#Dropping unused level 2
african_crises$currency_crises<-factor(african_crises$currency_crises)
levels(african_crises$currency_crises)
```

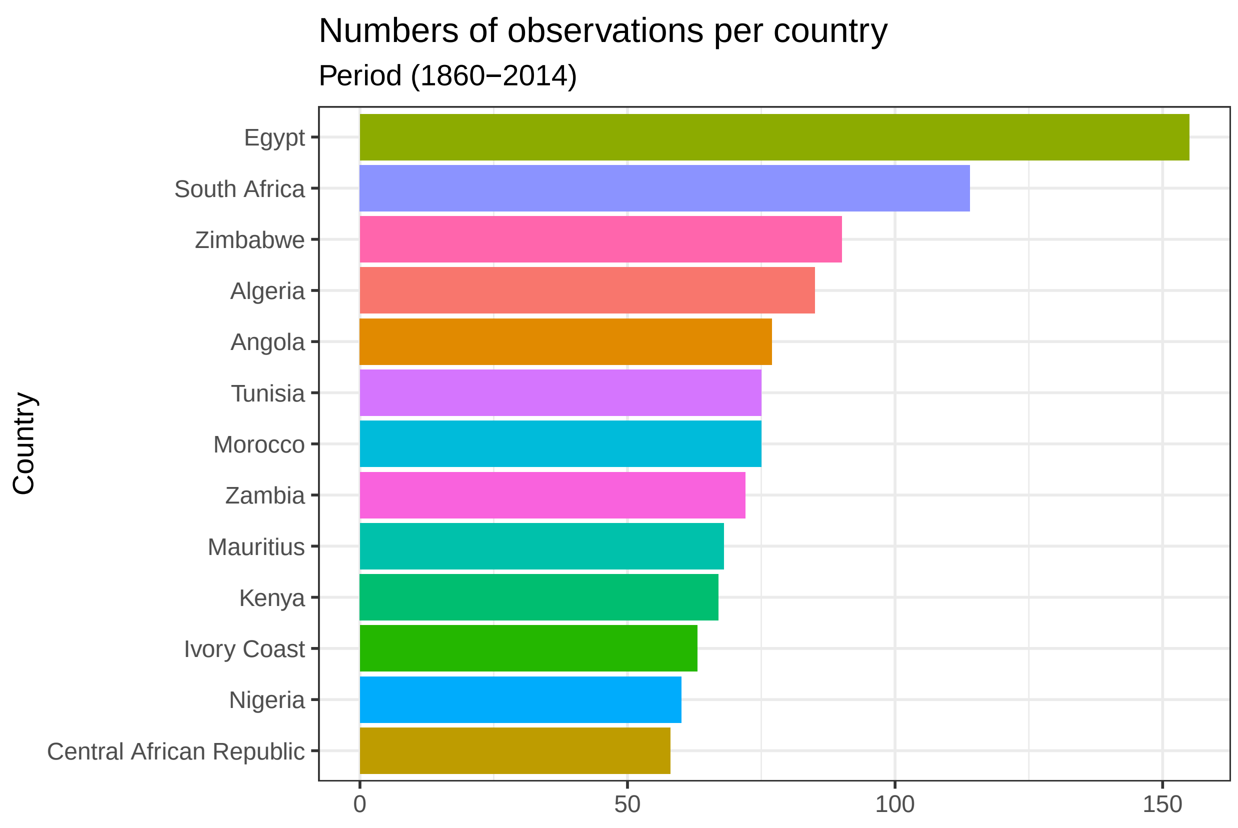
```
## [1] "0" "1"
```

Data exploration and visualization

Number of observations per country

Egypt has 155 observations available, South Africa has 114 observations, Zimbabwe has 90 observations and at the very end is the Central African Republic with 58 observations.

```r
#Number of total obervations per country
african_crises %>%
  group_by(country) %>%
  summarise(total=n()) %>%
  arrange(desc(total)) %>%
  ggplot(aes(x=reorder(country,+total),y=total, fill=country)) +
  geom_col() +
  theme_bw() + theme(legend.position="none") +
  labs(x = 'Country', y = '',
       title = 'Numbers of observations per country',
       subtitle = 'Period (1860-2014)') +
  coord_flip()
```



Systematic crisis per country

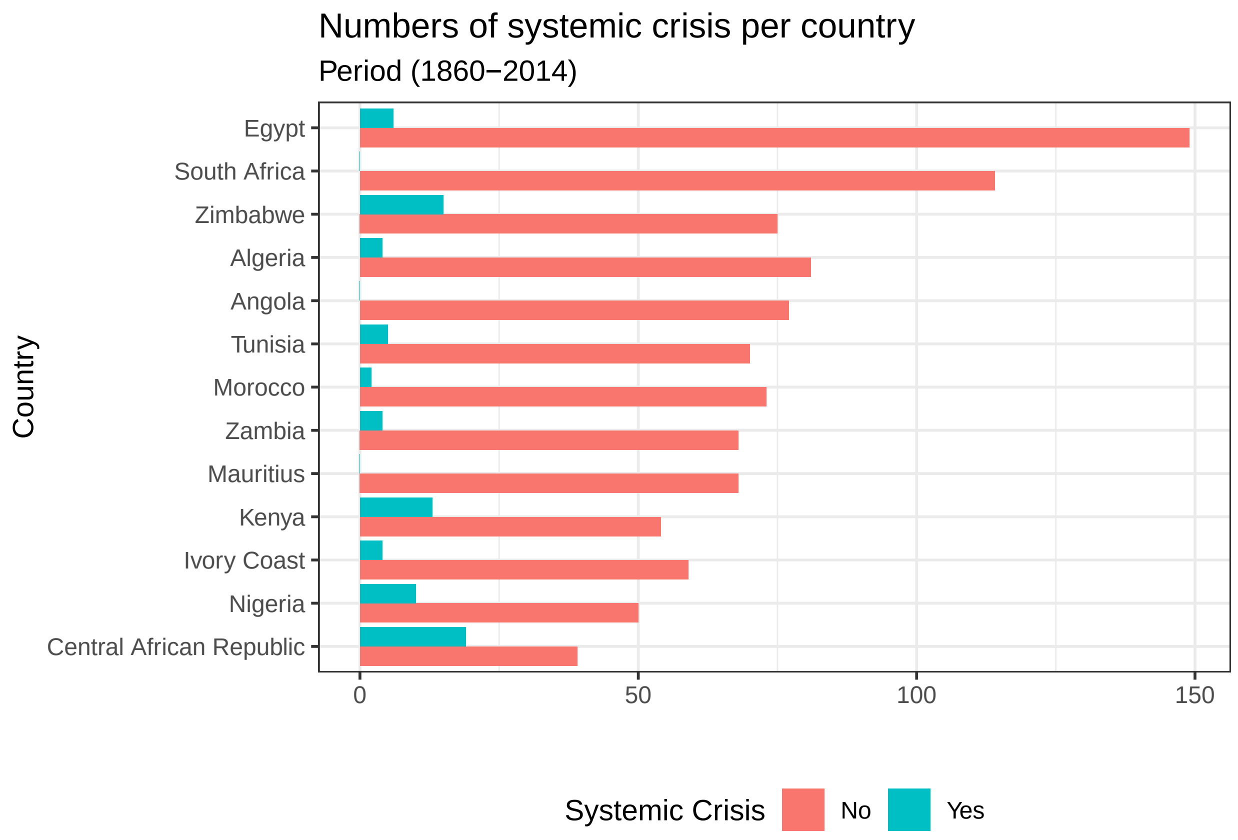
Central African Republic, Zimbabwe, Kenya, and Nigeria are among the countries with the highest number of systemic crises, with 19, 15, 13, and 10 crises, respectively. Angola, Mauritius, and South Africa on the other hand have never experienced a systemic crisis.

```
#Number of systematic crisis per country
african_crises %>%
  mutate(systemic_crisis = ifelse(systemic_crisis == 1, 'yes', 'no')) %>%
  group_by(country) %>%
  count(systemic_crisis) %>%
  spread(systemic_crisis, n) %>%
  mutate(yes = ifelse(is.na(yes), 0, yes)) %>%
  gather('no', 'yes', key = 'systemic_crisis', value = "cases") %>%
  arrange(desc(cases)) %>%
  kable()
```

country	systemic_crisis	cases
Egypt	no	149
South Africa	no	114
Algeria	no	81
Angola	no	77
Zimbabwe	no	75
Morocco	no	73

country	systemic_crisis	cases
Tunisia	no	70
Mauritius	no	68
Zambia	no	68
Ivory Coast	no	59
Kenya	no	54
Nigeria	no	50
Central African Republic	no	39
Central African Republic	yes	19
Zimbabwe	yes	15
Kenya	yes	13
Nigeria	yes	10
Egypt	yes	6
Tunisia	yes	5
Algeria	yes	4
Ivory Coast	yes	4
Zambia	yes	4
Morocco	yes	2
Angola	yes	0
Mauritius	yes	0
South Africa	yes	0

```r
african_crises %>%
  mutate(systemic_crisis = ifelse(systemic_crisis == 1, 'yes', 'no')) %>%
  group_by(country) %>%
  count(systemic_crisis) %>%
  spread(systemic_crisis, n) %>%
  mutate(yes = ifelse(is.na(yes), 0, yes)) %>%
  gather('no', 'yes', key = 'systemic_crisis', value = "cases") %>%
  data.frame() %>%
  ggplot(aes(x = reorder(country, +cases), y = cases,
           fill = systemic_crisis)) + geom_col(position = 'dodge') +
  theme_bw() +
  labs(x = 'Country', y = '',
       title = 'Numbers of systemic crisis per country',
       subtitle = 'Period (1860-2014)') +
  theme(legend.position="bottom") +
  scale_fill_discrete(name = "Systemic Crisis", labels = c('No', 'Yes')) +
  coord_flip()
```



Modeling approach

There are a host of algorithms for classification problems and for this project I will utilize three of them namely; Logistic Regression, Regression Trees, and Random Forest.

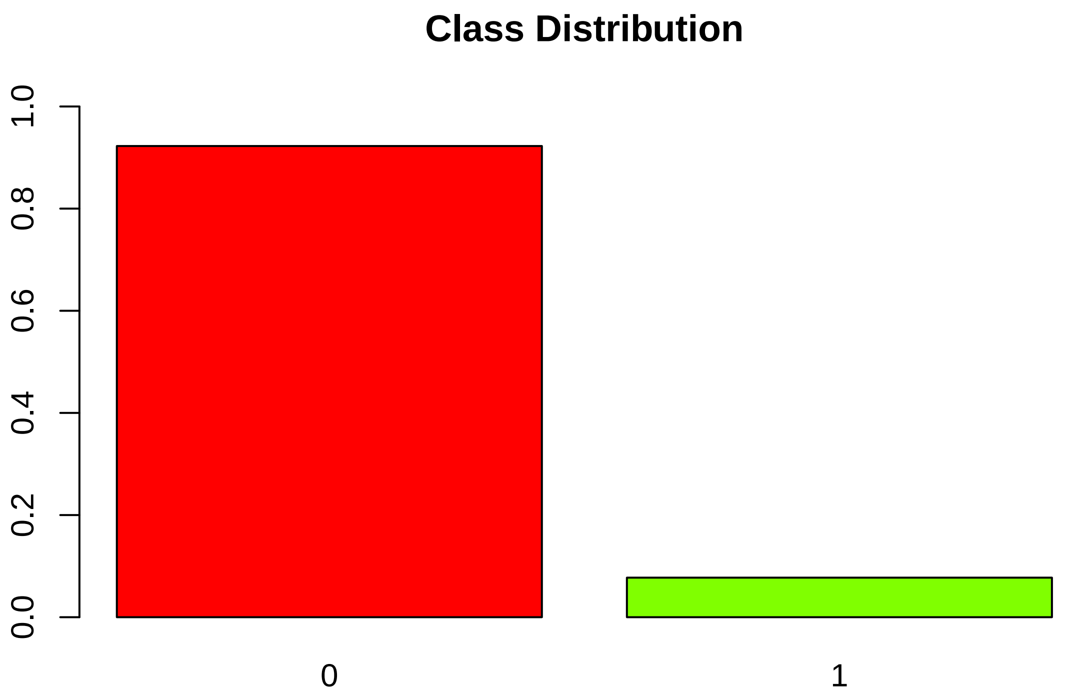
Class Imbalalance

However, before I proceed it would be wise to select relevant variables for these models and check for class imbalance as it would affect the accuracy of our models.

```
#select relevant variables for model
crisis<-african_crises %>%
  dplyr::select(-c(case,cc3,country,year))

# class imbalance problem detected
prop.table(table(crisis$systemic_crisis))
```

```
##
##          0          1
## 0.92256846 0.07743154
```

```
barplot(prop.table(table(crisis$systemic_crisis)),
        col = rainbow(4),
        ylim = c(0,1),
        main = 'Class Distribution')
```



As we can see systemic crisis has historically been detected in only 7.7% of our observations with no systemic crisis occupying about 92%.

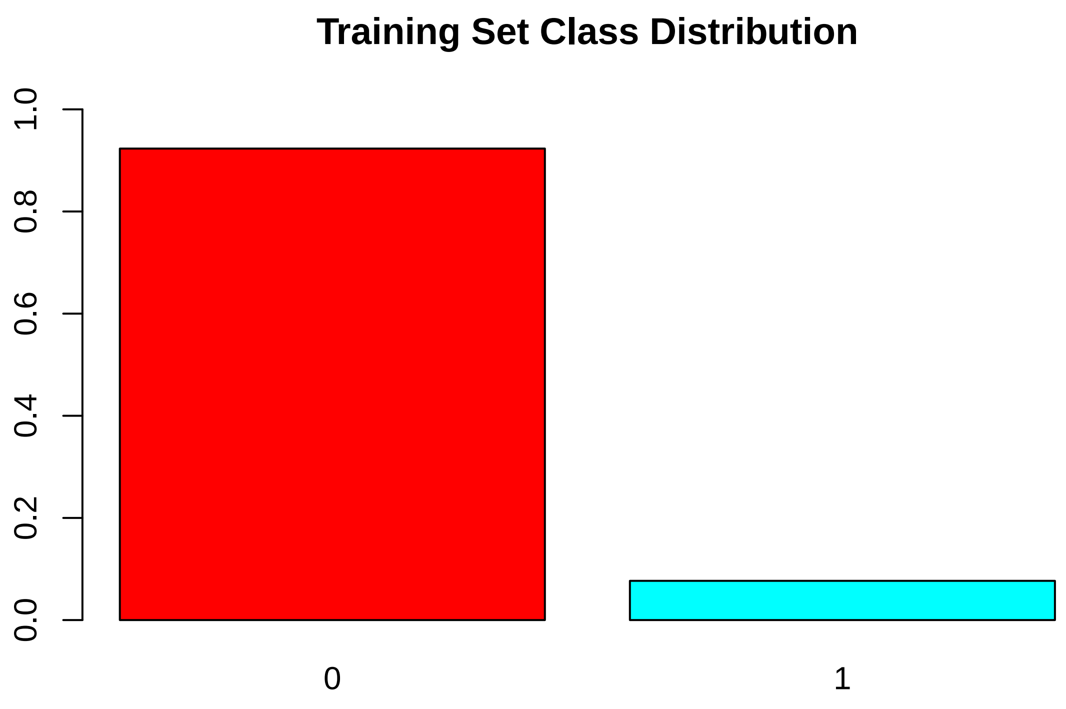
To remedy this we have to balance the class distribution. We will do this by utilizing Random oversampling which duplicates examples from the minority class to compensate for an imbalance that is present in the data, but first, we will split the data into a training and testing set and check for imbalance in the training set.

A split of 80% to the train set and 20% to the test set will be utilized to ensure more training set representativeness

```
#Data Partitioning Train: 80%, Test: 20%
set.seed(46, sample.kind = "Rounding")
y<-crisis$systemic_crisis
test_index <- createDataPartition(y, times = 1, p = 0.2, list = FALSE)
test_set <- crisis[test_index,]
train_set <- crisis[-test_index,]

#Imbalance in training set
table(train_set$systemic_crisis)
```

```
##
##   0   1
## 781  65
```

```r
barplot(prop.table(table(train_set$systemic_crisis)),
        col = rainbow(2),
        ylim = c(0,1),
        main = 'Training Set Class Distribution')
```



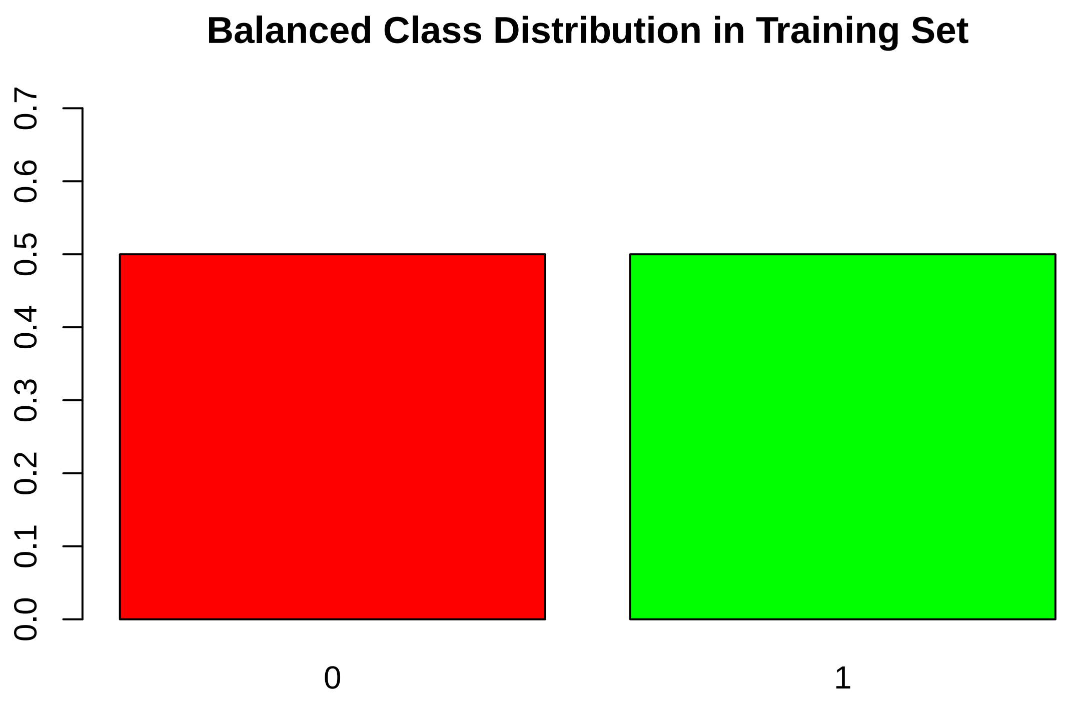
The imbalance in the training class mirrors that of the imbalance seen in the overall data. To correct this we will undergo random oversampling.

To achieve class balance I will select a sample of equal size in both the level 0 and level 1 class. I will therefore duplicate the number of samples in the 0 level to the 1 level (781 * 2 = 1562) and set N to 1562 creating an almost equal split between the two classes.

```r
# Over sampling to correct class imbalance
new_train_set<- ovun.sample(systemic_crisis~., data = train_set, 'over',
                  seed =222,#set seed to 222 for repeatability
                  N=1562)$data

#Imbalance corrected
table(new_train_set$systemic_crisis)
```

```
##
##   0   1
## 781 781
```

```r
barplot(prop.table(table(new_train_set$systemic_crisis)),
        col = rainbow(3),
        ylim = c(0,0.7),
        main = 'Balanced Class Distribution in Training Set')
```



There is now class balance in the training set. I will now proceed to use the new balanced training set in model development.

Logistic Regressoin

The first classification algorithm I would employ is the Logistic Regression, but first, we have to make sure the assumptions of logistic regression have been met, that is, variables should be independent of each other and residuals should not be autocorrelated.

```r
# correlation of numeric features
cor(crisis[,unlist(lapply(crisis,is.numeric))])
```

```
##                         exch_usd gdp_weighted_default inflation_annual_cpi
## exch_usd              1.00000000         -0.040725677         -0.011946587
## gdp_weighted_default -0.04072568          1.000000000         -0.004535404
## inflation_annual_cpi -0.01194659         -0.004535404          1.000000000
```

Two variables are strongly correlated if the correlation coefficient is either greater than 0.70 or less than -0.70. From the correlation matrix, none of the numeric features are strongly correlated. Therefore, the multicollinearity does not exist amongst the numeric variables.

Now we will check for multicollinearity in the factor variables.

```
# p-value of Chi Square tests to check for multicollinearity in factor variables
syd<-chisq.test(crisis$systemic_crisis,crisis$domestic_debt_in_default)$p.value
sys<-chisq.test(crisis$systemic_crisis,crisis$sovereign_external_debt_default)$p.value
syind<-chisq.test(crisis$systemic_crisis,crisis$independence)$p.value
syc<-chisq.test(crisis$systemic_crisis,crisis$currency_crises)$p.value
syi<-chisq.test(crisis$systemic_crisis,crisis$inflation_crises)$p.value
syb<-chisq.test(crisis$systemic_crisis,crisis$banking_crisis)$p.value

dsys<-chisq.test(crisis$domestic_debt_in_default,crisis$systemic_crisis)$p.value
ds<-chisq.test(crisis$domestic_debt_in_default,crisis$sovereign_external_debt_default)$p.value
dind<-chisq.test(crisis$domestic_debt_in_default,crisis$independence)$p.value
dc<-chisq.test(crisis$domestic_debt_in_default,crisis$currency_crises)$p.value
di<-chisq.test(crisis$domestic_debt_in_default,crisis$inflation_crises)$p.value
db<-chisq.test(crisis$domestic_debt_in_default,crisis$banking_crisis)$p.value

ssys<-chisq.test(crisis$sovereign_external_debt_default,crisis$systemic_crisis)$p.value
sd<-chisq.test(crisis$sovereign_external_debt_default,crisis$domestic_debt_in_default)$p.value
sind<-chisq.test(crisis$sovereign_external_debt_default,crisis$independence)$p.value
sc<-chisq.test(crisis$sovereign_external_debt_default,crisis$currency_crises)$p.value
si<-chisq.test(crisis$sovereign_external_debt_default,crisis$inflation_crises)$p.value
sb<-chisq.test(crisis$sovereign_external_debt_default,crisis$banking_crisis)$p.value

indsys<-chisq.test(crisis$independence,crisis$systemic_crisis)$p.value
indd<-chisq.test(crisis$independence,crisis$domestic_debt_in_default)$p.value
inds<-chisq.test(crisis$independence,crisis$sovereign_external_debt_default)$p.value
indc<-chisq.test(crisis$independence,crisis$currency_crises)$p.value
indi<-chisq.test(crisis$independence,crisis$inflation_crises)$p.value
indb<-chisq.test(crisis$independence,crisis$banking_crisis)$p.value

csys<-chisq.test(crisis$currency_crises,crisis$systemic_crisis)$p.value
cd<-chisq.test(crisis$currency_crises,crisis$domestic_debt_in_default)$p.value
cs<-chisq.test(crisis$currency_crises,crisis$sovereign_external_debt_default)$p.value
cind<-chisq.test(crisis$currency_crises,crisis$independence)$p.value
ci<-chisq.test(crisis$currency_crises,crisis$inflation_crises)$p.value
cb<-chisq.test(crisis$currency_crises,crisis$banking_crisis)$p.value

isys<-chisq.test(crisis$inflation_crises,crisis$systemic_crisis)$p.value
id<-chisq.test(crisis$inflation_crises,crisis$domestic_debt_in_default)$p.value
is<-chisq.test(crisis$inflation_crises,crisis$sovereign_external_debt_default)$p.value
iind<-chisq.test(crisis$inflation_crises,crisis$independence)$p.value
ic<-chisq.test(crisis$inflation_crises,crisis$currency_crises)$p.value
ib<-chisq.test(crisis$inflation_crises,crisis$banking_crisis)$p.value

bsys<-chisq.test(crisis$banking_crisis,crisis$systemic_crisis)$p.value
bd<-chisq.test(crisis$banking_crisis,crisis$domestic_debt_in_default)$p.value
bs<-chisq.test(crisis$banking_crisis,crisis$sovereign_external_debt_default)$p.value
bind<-chisq.test(crisis$banking_crisis,crisis$independence)$p.value
bc<-chisq.test(crisis$banking_crisis,crisis$currency_crises)$p.value
bi<-chisq.test(crisis$banking_crisis,crisis$inflation_crises)$p.value

cormatrix <- matrix(c(0,syd,sys,syind,syc,syi,syb,
                      dsys,0,ds,dind,dc,di,db,
                      ssys,sd,0,sind,sc,si,sb,
                      indsys,indd,inds,0,indc,indi,indb,
                      csys,cd,cs,cind,0,ci,cb,
                      isys,id,is,iind,ic,0,ib,
                      bsys,bd,bs,bind,bc,bi,0),
                    7,7,byrow = TRUE)

row.names(cormatrix) = colnames(cormatrix) = c('systemic_crisis','domestic_debt_in_default',
                                               'sovereign_external_debt_default',
                                              'independence','currency_crises',
                                              'inflation_crises','banking_crisis')
cormatrix
```

```
##                                 systemic_crisis domestic_debt_in_default
## systemic_crisis                    0.000000e+00             2.325439e-04
## domestic_debt_in_default           2.325439e-04             0.000000e+00
## sovereign_external_debt_default    1.574014e-15             3.042646e-50
## independence                       3.343650e-06             7.735109e-04
## currency_crises                    1.634895e-04             3.159207e-11
## inflation_crises                   5.201978e-08             1.562455e-12
## banking_crisis                    1.972036e-167             1.536810e-12
##                                 sovereign_external_debt_default independence
## systemic_crisis                                    1.574014e-15 3.343650e-06
## domestic_debt_in_default                           3.042646e-50 7.735109e-04
## sovereign_external_debt_default                    0.000000e+00 2.416552e-13
## independence                                       2.416552e-13 0.000000e+00
## currency_crises                                    1.482677e-09 8.526918e-03
## inflation_crises                                   2.118056e-09 5.326906e-01
## banking_crisis                                     3.151490e-17 4.088792e-07
##                                 currency_crises inflation_crises banking_crisis
## systemic_crisis                    1.634895e-04     5.201978e-08  1.972036e-167
## domestic_debt_in_default           3.159207e-11     1.562455e-12   1.536810e-12
## sovereign_external_debt_default    1.482677e-09     2.118056e-09   3.151490e-17
## independence                       8.526918e-03     5.326906e-01   4.088792e-07
## currency_crises                    0.000000e+00     1.012317e-37   1.122847e-07
## inflation_crises                   1.012317e-37     0.000000e+00   5.726851e-14
## banking_crisis                     1.122847e-07     5.726851e-14   0.000000e+00
```

I used the Chi-Square test to test the independence of the categorical variables. Since almost all the variables have p-values < 0.05, we reject each Ho: Two factors are independent at a 5% significance level.

This violates the independence assumption of variables, thus, we can conclude that multicollinearity does exist amongst the factor variables. I will correct this issue down the road, for now, I will go ahead and fit the logistic regression model.

```
# Fitting Logistic Regression on Training set
set.seed(46, sample.kind = "Rounding")
fit_glm0<- glm(formula = systemic_crisis ~.,data = new_train_set ,family = 'binomial')
summary(fit_glm0)
```

```
##
## Call:
## glm(formula = systemic_crisis ~ ., family = "binomial", data = new_train_set)
```

```
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -4.0033  -0.2234   0.0000   0.1122   1.5859
##
## Coefficients:
##                                   Estimate Std. Error z value Pr(>|z|)
## (Intercept)                     -1.100e+01  7.406e+02  -0.015   0.9881
## exch_usd                        -2.950e-03  1.265e-03  -2.332   0.0197 *
## domestic_debt_in_default1       -4.083e+00  7.172e-01  -5.693 1.25e-08 ***
## sovereign_external_debt_default1  3.582e+00  4.047e-01   8.849  < 2e-16 ***
## gdp_weighted_default            -1.718e+01  3.653e+00  -4.702 2.57e-06 ***
## inflation_annual_cpi             1.597e-05  5.481e-05   0.291   0.7707
## independence1                    1.607e+01  7.406e+02   0.022   0.9827
## currency_crises1                 2.955e+00  3.481e-01   8.489  < 2e-16 ***
## inflation_crises1               -3.268e+00  5.580e-01  -5.857 4.72e-09 ***
## banking_crisisno_crisis         -8.730e+00  6.210e-01 -14.058  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 2165.39  on 1561  degrees of freedom
## Residual deviance:  392.01  on 1552  degrees of freedom
## AIC: 412.01
##
## Number of Fisher Scoring iterations: 18
```

The variables **independence** and **inflation annual cpi** are not statistically significant at any reasonable level of significance. The standard error of the **independence variable** is quite large, a typical effect of multicollinearity.

Let's note that the more statistically significant a variable, the more asterisks(*) it has as we can see from the summary above.

Handling Multicollinearity

To make our model more robust. I will remove all statistically insignificant variables including the **independence variable** which exibited a high degree of multicollinearity.

```
# Fitting Logistic Regression on Training set without insignificant variables
set.seed(46, sample.kind = "Rounding")
fit_glm1<- glm(formula = systemic_crisis ~ domestic_debt_in_default+
                 sovereign_external_debt_default+
                 gdp_weighted_default+
                 currency_crises+
                 inflation_crises+
                 banking_crisis+
                 exch_usd
               ,data = new_train_set ,family = 'binomial')

summary(fit_glm1)
```

```
##
## Call:
```

```
## glm(formula = systemic_crisis ~ domestic_debt_in_default + sovereign_external_debt_default +
##     gdp_weighted_default + currency_crises + inflation_crises +
##     banking_crisis + exch_usd, family = "binomial", data = new_train_set)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -4.0621  -0.1944  -0.0103   0.1107   1.6367
##
## Coefficients:
##                                    Estimate Std. Error z value Pr(>|z|)
## (Intercept)                        5.119962   0.537429   9.527  < 2e-16 ***
## domestic_debt_in_default1         -4.258953   0.722617  -5.894 3.77e-09 ***
## sovereign_external_debt_default1   3.872820   0.403079   9.608  < 2e-16 ***
## gdp_weighted_default             -18.028625   3.697556  -4.876 1.08e-06 ***
## currency_crises1                   3.140056   0.344603   9.112  < 2e-16 ***
## inflation_crises1                 -3.409011   0.555373  -6.138 8.34e-10 ***
## banking_crisisno_crisis           -9.077947   0.628802 -14.437  < 2e-16 ***
## exch_usd                          -0.002983   0.001262  -2.363   0.0181 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 2165.39  on 1561  degrees of freedom
## Residual deviance:  402.85  on 1554  degrees of freedom
## AIC: 418.85
##
## Number of Fisher Scoring iterations: 8
```

We can now see all the variables are significant with small standard errors.

Checking for Autocorrelation

To check the degree of autocorrelation we will use the Durbin Watson test. This test is a measure of autocorrelation in the residuals, that is, it considers the correlation of the residuals. so if the data are ordered in some way, you'll get a significant DW test. Since our data set is ordered we will have to randomize it to get a valid result from the Dubin Watson Test.

The DW statistic always has a value between 0 and 4.0. A value of 2.0 means there is no autocorrelation detected in the sample. Values from zero to 2.0 indicate positive autocorrelation and values from 2.0 to 4.0 indicate negative autocorrelation.

```r
# Randomize training set order
set.seed(46, sample.kind = "Rounding")
new_train_set <- new_train_set[order(runif(nrow(new_train_set))), ]

#Fit model with radominzed data
fit_glm1<- glm(formula = systemic_crisis ~ domestic_debt_in_default+
                 sovereign_external_debt_default+
                 gdp_weighted_default+
                 currency_crises+
                 inflation_crises+
                 banking_crisis+
                 exch_usd
               ,data = new_train_set ,family = 'binomial')
```

```r
#DurbinWatsonTest
set.seed(46, sample.kind = "Rounding")
durbinWatsonTest(fit_glm1)
```

```
##  lag Autocorrelation D-W Statistic p-value
##    1      0.01024064      1.979331   0.648
##  Alternative hypothesis: rho != 0
```

The Durbin-Waston test now presents a D-W Statistic of 1.979 (close to 2.0) and a p-value of > 0.05, thus, we do not reject Ho: Residuals are not autocorrelated.

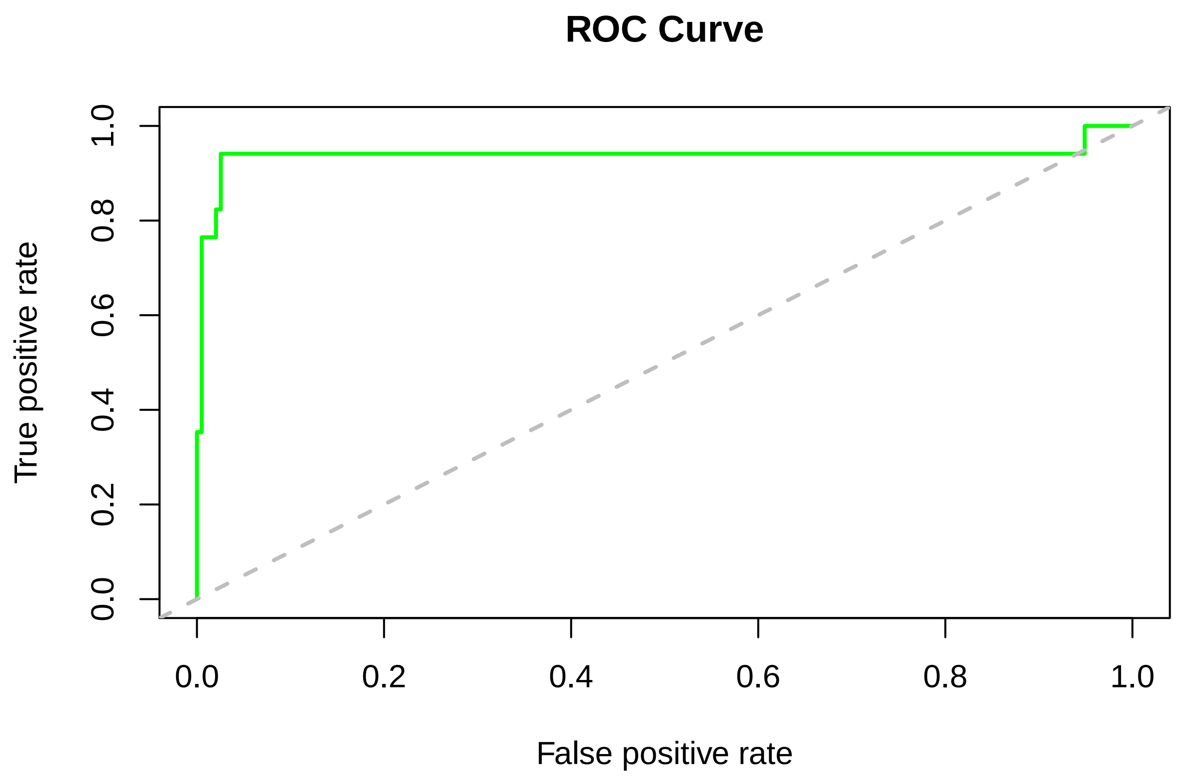
We can therefore conclude there is sufficient evidence to say residuals are not autocorrelated. The assumptions have been checked and passed.

Next we move on to prediciton.

```r
# Predicting the test set results with the glm model
pred_glm_1<- predict(fit_glm1,test_set, type = 'response')
y_hat_glm1<- ifelse(pred_glm_1>0.5,1,0)
confusionMatrix(as.factor(y_hat_glm1),test_set$systemic_crisis)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 190   1
##          1   6  16
##
##                Accuracy : 0.9671
##                  95% CI : (0.9335, 0.9867)
##     No Information Rate : 0.9202
##     P-Value [Acc > NIR] : 0.004173
##
##                   Kappa : 0.8028
##
##  Mcnemar's Test P-Value : 0.130570
##
##             Sensitivity : 0.9694
##             Specificity : 0.9412
##          Pos Pred Value : 0.9948
##          Neg Pred Value : 0.7273
##              Prevalence : 0.9202
##          Detection Rate : 0.8920
##    Detection Prevalence : 0.8967
##       Balanced Accuracy : 0.9553
##
##        'Positive' Class : 0
##
```

```r
# Use the predictions to build a ROC curve to assess the performance of our model
fitpred = prediction(pred_glm_1, test_set$systemic_crisis)
fitperf = performance(fitpred,"tpr","fpr")
plot(fitperf,col="green",lwd=2,main="ROC Curve")
abline(a=0,b=1,lwd=2,lty=2,col="gray")
```



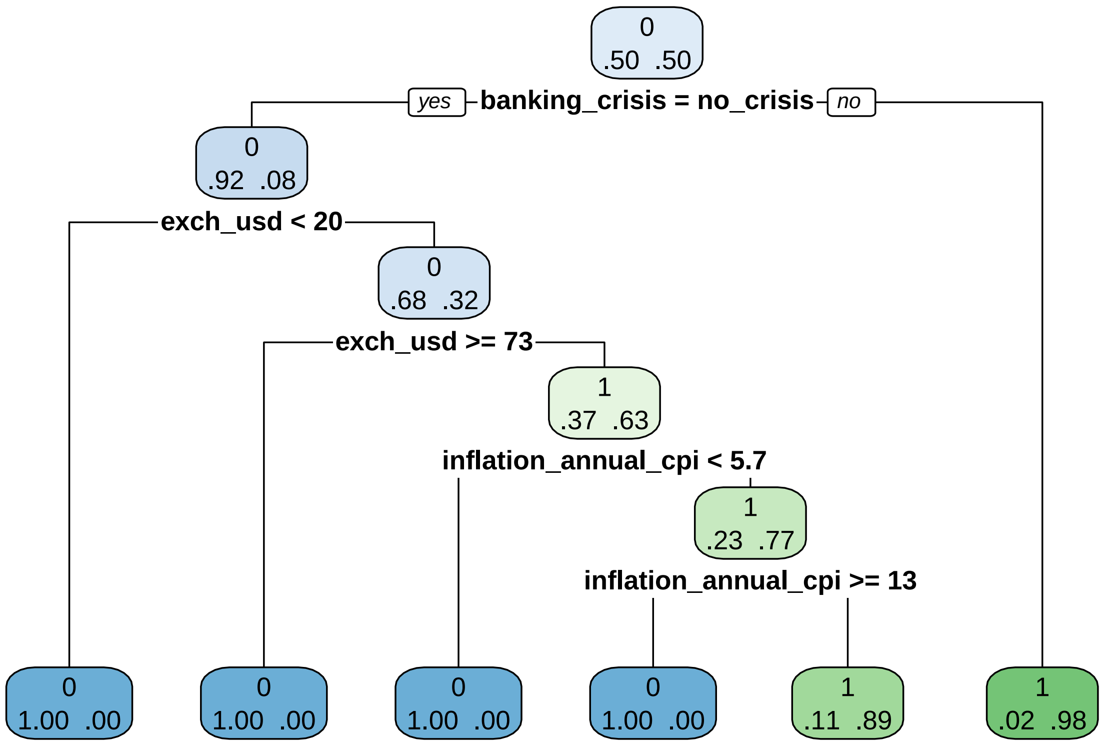
The ROC (Receiver Operating Characteristics) curve is a graphical representation of the performance of the logistic model and it shows the performance of our model rises well above the diagonal line. This indicates that our logistic regression model performs better than just a random guess with an accuracy of 0.9671, sensitivity of **0.969**, the specificity of **0.941**, and balanced accuracy of **0.955** as seen in the confusion matrix above.

Decision Trees

Next, we will utilize the Decision Tree algorithm. Decision trees operate by predicting an outcome variable by partitioning the predictor space. Although it utilizes a single tree unlike the random forest, its merit resides in its interpretability which the random forest algorithm lacks.

Now let's fit a decision tree to our model.

```
#Fitting decision tree model to the training set
set.seed(46, sample.kind = "Rounding")
fit_rpart<-rpart(systemic_crisis~., data = new_train_set,
                 method = 'class')
# Decision tree visualizaton
rpart.plot(fit_rpart, extra = 4)
```



The tree uses three variables **banking crisis, exch USD and inflation annual cpi.**

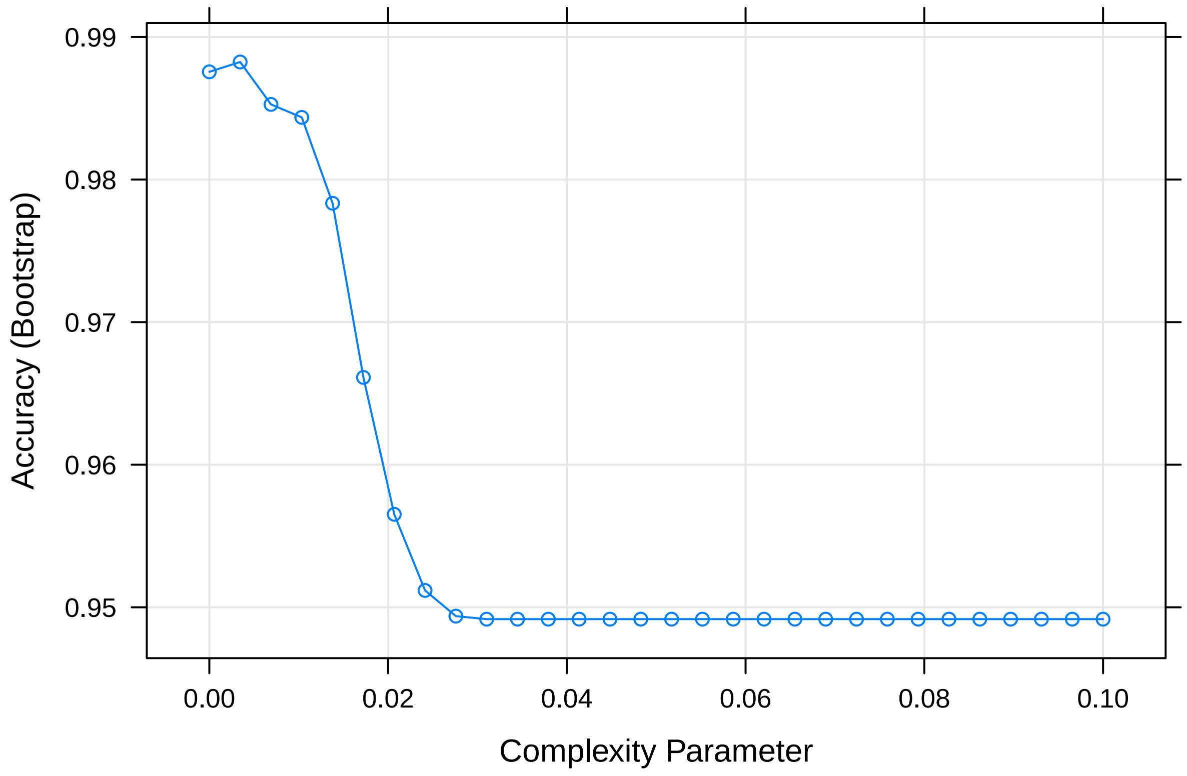
Let's see how well the model performs

```
#Predicting the test set results with the decision tree model
pred_rpart<- predict(fit_rpart,test_set,type = 'class')
confusionMatrix(pred_rpart,test_set$systemic_crisis)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 187   1
##          1   9  16
##
##                Accuracy : 0.9531
##                  95% CI : (0.9154, 0.9773)
##     No Information Rate : 0.9202
##     P-Value [Acc > NIR] : 0.04272
##
##                   Kappa : 0.7369
##
##  Mcnemar's Test P-Value : 0.02686
##
##             Sensitivity : 0.9541
##             Specificity : 0.9412
##          Pos Pred Value : 0.9947
##          Neg Pred Value : 0.6400
##              Prevalence : 0.9202
##          Detection Rate : 0.8779
##    Detection Prevalence : 0.8826
##       Balanced Accuracy : 0.9476
##
##        'Positive' Class : 0
##
```

I had an accuracy of **0.9531** however, let's see if we can improve the model by using **cross-validation to choose the best complexity parameter (cp)**.

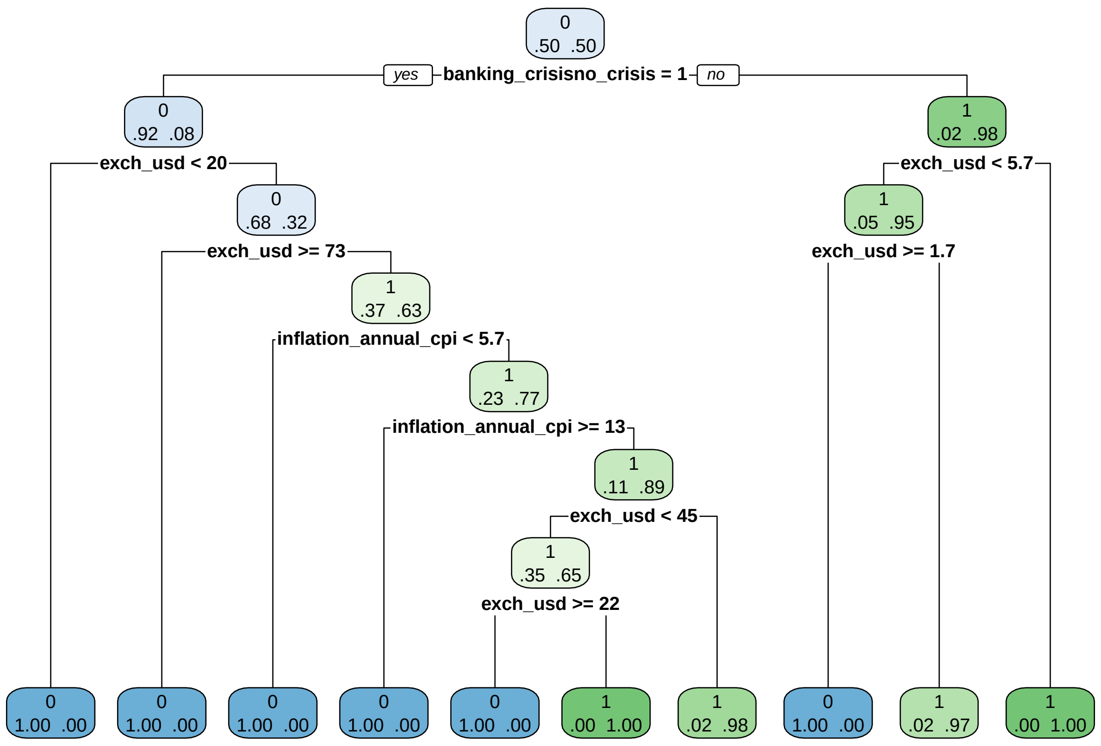
```
# Using cross validation to choose cp
set.seed(46, sample.kind = "Rounding")
fit_rpart_cv<- train(systemic_crisis~., data = new_train_set,
                     method = 'rpart',
                     tuneGrid = data.frame(cp = seq(0, 0.1, len = 30)))

# Best tune
plot(fit_rpart_cv)
```



```
fit_rpart_cv$bestTune
```

```
##            cp
## 2 0.003448276
```

```r
# Tree Visualization
rpart.plot(fit_rpart_cv$finalModel, extra = 4)
```



```r
#Prediction the test set results with cross validated decision tree model
pred_rpart_cv<- predict(fit_rpart_cv,test_set)
confusionMatrix(pred_rpart_cv,test_set$systemic_crisis)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 190   2
##          1   6  15
##
##                Accuracy : 0.9624
##                  95% CI : (0.9273, 0.9836)
##     No Information Rate : 0.9202
##     P-Value [Acc > NIR] : 0.01013
##
##                   Kappa : 0.7691
##
##  Mcnemar's Test P-Value : 0.28884
##
##             Sensitivity : 0.9694
##             Specificity : 0.8824
##          Pos Pred Value : 0.9896
##          Neg Pred Value : 0.7143
##              Prevalence : 0.9202
##          Detection Rate : 0.8920
##    Detection Prevalence : 0.9014
##       Balanced Accuracy : 0.9259
##
##        'Positive' Class : 0
##
```

I was able to improve the accuracy of the model from **0.9531 to 0.9624** by selecting the best complexity parameter of **0.003448276** through cross-validation.

However the same variables, that is, **banking crisis**, **exch USD**, and **inflation annual cpi** were maintained in the model.

Random Forest

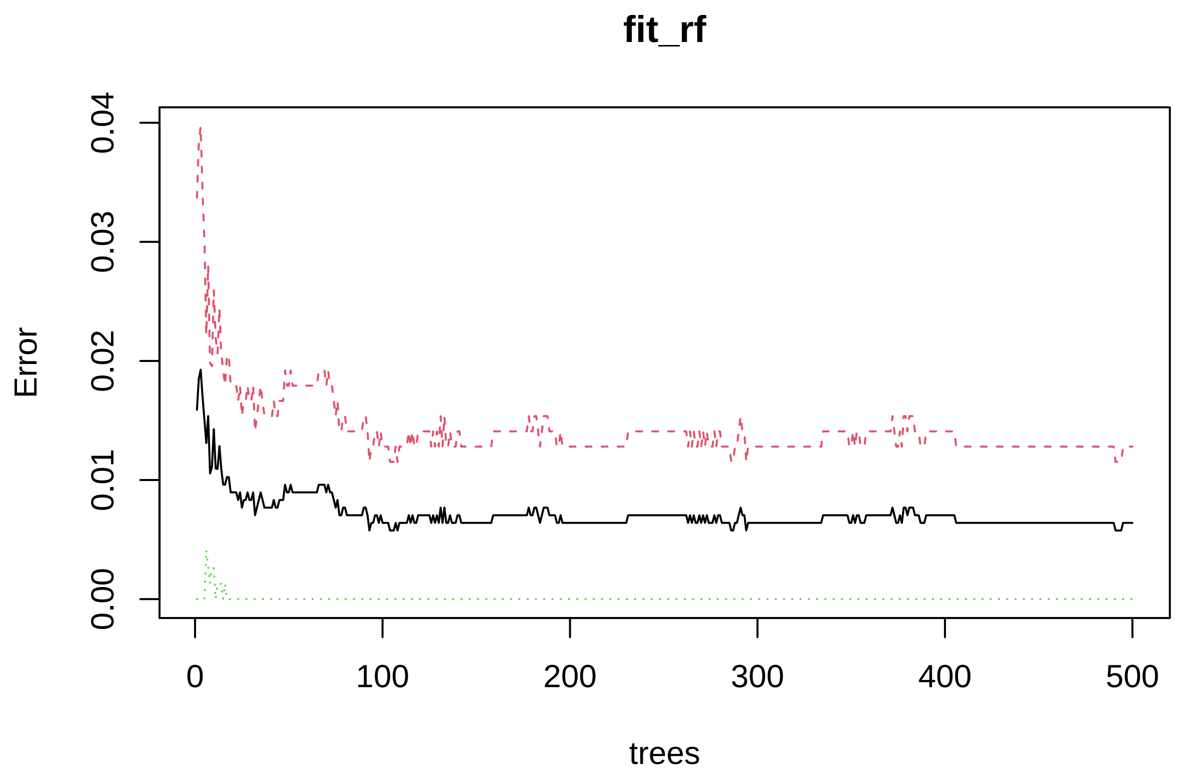
Random forests improve accuracy by generating a large number of bootstrapped trees. It reduces instability by averaging multiple decision trees, that is, a forest of trees constructed with randomness.

```
# Fitting Random Forest Classification to the Training set
set.seed(46, sample.kind = "Rounding")
fit_rf<-randomForest(systemic_crisis~.,
                data= new_train_set)

#Predicting the test set results with the random forest model
pred_rf<- predict(fit_rf,test_set)
confusionMatrix(pred_rf,test_set$systemic_crisis)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 191   1
##          1   5  16
##
##                Accuracy : 0.9718
##                  95% CI : (0.9397, 0.9896)
##     No Information Rate : 0.9202
##     P-Value [Acc > NIR] : 0.001506
##
##                   Kappa : 0.8268
##
##  Mcnemar's Test P-Value : 0.220671
##
##             Sensitivity : 0.9745
##             Specificity : 0.9412
##          Pos Pred Value : 0.9948
##          Neg Pred Value : 0.7619
##              Prevalence : 0.9202
##          Detection Rate : 0.8967
##    Detection Prevalence : 0.9014
##        Balanced Accuracy : 0.9578
##
##         'Positive' Class : 0
##
```

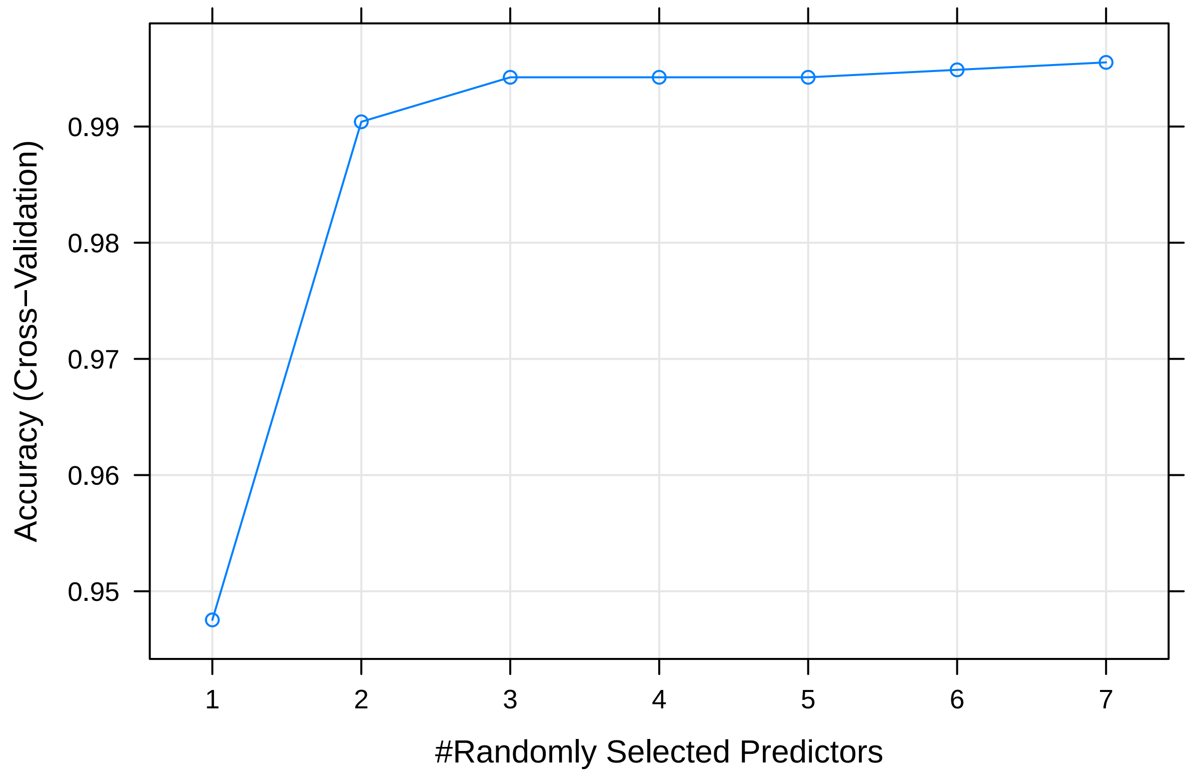
I had an accuracy of **0.9718** which is higher than that of the single decision tree algorithm of 0.9624. Let's try to improve this model by using cross-validation to choose the best parameters.

```
#Choosing the number of trees
plot(fit_rf)
```



The red, black, and green lines represent the error rate for no systemic crisis, overall, and a systemic crisis. The overall error rate converges to around 0.007. The overall error rate seems to be low around 100 trees so I will tune this model to that number. I would also use 10-fold cross-validation and test values of mtry (The tuning parameter for random forest) ranging from 1 to 7 for the best accuracy.

```
# Fitting to the training set and Using cross validation to choose the best parameter for the random fo
set.seed(46, sample.kind = "Rounding")
fit_rf_cv<-train(systemic_crisis~.,
              method = 'rf', data= new_train_set,
              tuneGrid = data.frame(mtry = seq(1, 7)),
              ntree = 100,
              trControl=trainControl(method = "cv", number = 10, p = .9))

# Best mtry value that maximizes accuracy
plot(fit_rf_cv)
```



```
fit_rf_cv$bestTune
```

```
##   mtry
## 7    7
```

```
#Checking the prediction accuracy
pred_rf_cv<- predict(fit_rf_cv,test_set)
confusionMatrix(pred_rf_cv,test_set$systemic_crisis)
```

```
## Confusion Matrix and Statistics
##
##             Reference
## Prediction   0   1
##          0 193   2
##          1   3  15
##
##                Accuracy : 0.9765
##                  95% CI : (0.9461, 0.9923)
##     No Information Rate : 0.9202
##     P-Value [Acc > NIR] : 0.000466
##
##                   Kappa : 0.8444
##
##  Mcnemar's Test P-Value : 1.000000
##
```

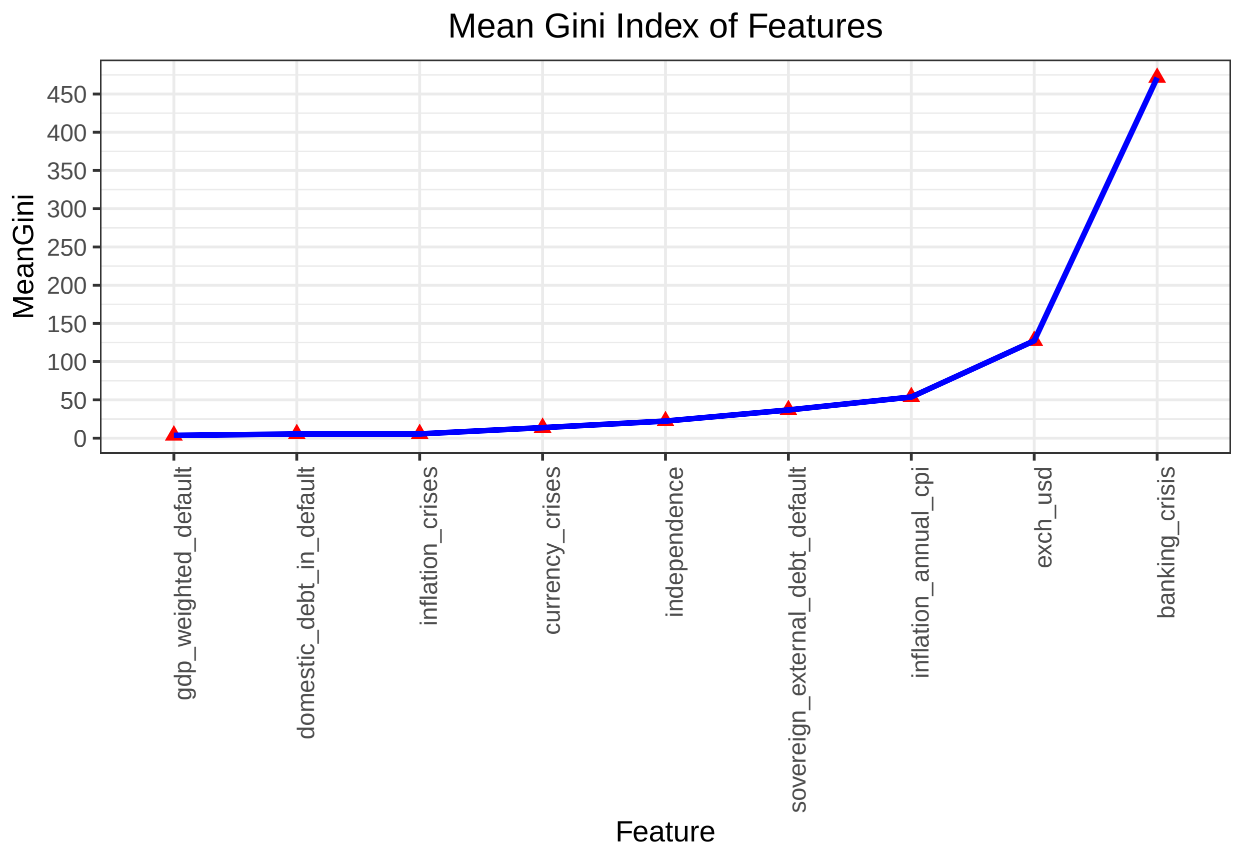
```
##             Sensitivity : 0.9847
##             Specificity : 0.8824
##          Pos Pred Value : 0.9897
##          Neg Pred Value : 0.8333
##              Prevalence : 0.9202
##          Detection Rate : 0.9061
##    Detection Prevalence : 0.9155
##       Balanced Accuracy : 0.9335
##
##        'Positive' Class : 0
##
```

With **10-fold cross-validation** and choosing the best parameter of **mtry = 7**, we were able to improve the accuracy from **0.9718** to **0.9765**.

The random forest algorithm is notorious for its lack of interpretability. It would be tough to visualize all 100 trees and identify the best variables for the model.

However, we can assess the **Feature Importance** using the **Gini index measure**.

```
# Feature Importance
gini = as.data.frame(importance(fit_rf))
gini = data.frame(Feature = row.names(gini),
                  MeanGini = round(gini[ ,'MeanDecreaseGini'], 2))
gini = gini[order(-gini[,"MeanGini"]),]

ggplot(gini,aes(reorder(Feature,MeanGini), MeanGini, group=1)) +
  geom_point(color='red',shape=17,size=2) +
  geom_line(color='blue',size=1) +
  scale_y_continuous(breaks=seq(0,500,50)) +
  xlab("Feature") +
  ggtitle("Mean Gini Index of Features") +
  theme_bw() +
  theme(plot.title = element_text(hjust = 0.5)) +
  theme(axis.text.x = element_text(angle = 90, hjust = 1))
```



The feature/variable **banking crisis** had the highest mean Gini index, thus the highest importance. **Exch USD** also had relatively high importance and this is followed by **inflation annual cpi**.

Results

Logistic Regression

- The assumptions of multicollinearity and autocorrelation were met.
- The best model produced an accuracy, sensitivity, specificity, and balanced accuracy of **0.9671, 0.9694, 0.9412, and 0.9553** respectively.
- Seven variables, **domestic__debt__in__default, sovereign external debt default, GDP weighted default, currency crises, inflation crises, banking crisis, exch USD** significantly contributed to the model in predicting a systemic crisis.

Decision Tree

- The accuracy of the model was improved from **0.9531 to 0.9624**. The final model also had a sensitivity, specificity, and balanced accuracy of **0.9694, 0.8824, and 0.9259** respectively.
- The model was improved by choosing a complexity parameter of **0.003448276** through cross-validation.
- The variables used for the model were **banking crisis, exch USD, and inflation annual cpi**.

Random Forest

- The accuracy of the model was improved from **0.9718 to 0.9765**. This is the **highest accuracy** of all the algorithms employed. The final model also had a sensitivity, specificity, and balanced accuracy of **0.9847,0.8824 and 0.9335** respectively.
- The model was improved by choosing **100 trees, a mtry of 7 and also utilizing 10-fold cross-validation.**
- The first five variables based on their importance (highest to lowest) are as follows: **banking crisis, exch USD, inflation annual cpi, sovereign external debt default, and independence.**

Conclusion

Summary

The goal of this project was to develop a model that would predict the occurrence of a systemic crisis in Africa and find the most important variables associated with it.

By correcting the class imbalance, performing cross-validation, and tunning the model for the best result, the **Random Forest Algorithm** was the best classification model with an accuracy of about **97.6%**. This was followed by the logistic regression and the decision tree algorithm with accuracies of **96.7% and 96.2%** respectively.

I also found out from the Random Forest model that the top 3 features/variables most associated with a systemic crisis were **banking crisis, an exchange rate (USD), and inflation (Annual consumer price index).**

Potential Impact

These findings can guide African governments to monitor and control the key variables most associated with a systemic crisis to avert its occurrence.

The models can also be employed in determining the trajectory of a country's economy in terms of an occurrence of a systemic crisis so that preventive and remedial measures can be quickly implemented.

Limitations

A major limitation of this project was the small sample of African countries in the data set. This data only covered 13 out of 54 African countries thereby depriving us of a more holistic analysis of the continent.

Another limitation was the limited number of variables to analyze. Information on the unemployment rate, bond yields, The stock market, and many others may have contributed to a more inclusive and encompassing model.

Future work

In this project, we mainly looked at macroeconomic variables which focuses more on decisions made by governments. For my future work I would like to consider the effects of microeconomics, that is, decisions made by individuals and businesses, and analyze how it may also affect the occurrence of a systemic crisis in an economy.

The limitations of this project also point towards work to be done in the future. I would therefore work towards including data from other African countries as well as improving the number of relevant macro and microeconomic variables to produce a more robust model.