BANKING CRISIS IN AFRICA

BETWEEN 1860-2014 AND THE LESSONS THEREFROM

Sruti Raman - 18BCE7102

Praveen Raj - 18BCE7088

1.0 Summary of the Previous Report

Our descriptive analytics study indicated that the crisis in African Countries have emerged in an area and can have cascading effect if not controlled with timely interventions. Additionally, the countries which had higher inflation, faced devaluation of their currencies, defaults in sovereign debt, followed by banking crisis and systemic risk. In a few cases, the policies of the Governments also have led to further crisis. Our analysis also indicated that Central African Republic (CAR) also suffered due to the early 1980 recession the world has witnessed between the start of 1980 and 1983. The world again witnessed early 1990s recession which describes the period of economic downturn affecting much of the western world in the early 1990s.

2.0 Abstract of our Present Study

Tools Used: Python and Excel

Section 3 details the Business Question

Section 4 gives a brief on data points collected and analyzed.

Section 5 gives definitions of terminology of a few economic indicators used in our study.

Section 6 extends the business analysis and data visualization.

Section 7 introduces the Prescriptive Analysis and Model used; the results derived therefrom.

Section 8 Validates the test results of the Model, with an independent analysis of facts.

Section 9 gives the Conclusion and the Summary of results derived from the Model.

Section 10 gives the screen shots of implementation of Model.

3.0 Business Questions

Comprehend frequency of banking crisis in African countries and understand whether there exists any correlation to other economic events, in the light of banking crisis in the region, on many occasions during the past one-and-half century.

A brief business analysis was done in our previous paper, by collecting a few data points available in the public domain, which is further analyzed with a regression model and other validation checks.

4.0 Datasets

The sample size of dataset comprises 13 African countries consisting of Algeria, Angola, Central African Republic, Ivory Coast, Egypt, Kenya, Mauritius, Morocco, Nigeria, South Africa, Tunisia, Zambia, and Zimbabwe and relates to the period 1860 to 2014. The data points primarily consist of economic, banking, systemic factors etc.

Link https://www.hbs.edu/behavioral-finance-and-financial-stability/data/Pages/global.aspx

5.0 Definitions

In order to appreciate the analysis, a few of the terminology used in the analysis are briefly explained below.

5.1 Systemic crisis:

In a systemic crisis financial troubles spread between institutions and markets until the whole monetary and financial system gets effected with global consequences. It may result in high inflationary trends, exchange rate fluctuations, domestic and external debt defaults etc with far reaching effects.

5.2 Sovereign Debt Default:

Sovereign Debt default is similar to default on debt by an individual or business, but in this case, default is by a Central or State Government when it fails to repay its interest or principal due.

Sovereign default results in Credit Rating Agencies downgrading the credit rating of the State Government or Central Government, making their future borrowings expensive (with high rates of interest) or non-availability of credit to them. Thus, such governments/countries are forced to practice more austerity measures before any further credit is available to them from international agencies.

5.3 Gross Domestic Product (GDP):

Gross Domestic Product (GDP) is the value of final goods and services produced in a country during a certain period, typically 1 Year. GDP is the most closely watched important economic indicator by both economists and investors, as it represents the economic growth rate and the prospect of range of returns expected on the investments made in the country.

5.4 Inflation:

Inflation reflects the increase in the level of prices of goods and services. The most common indicator of inflation is CPI, Consumer Price Index, which measures the percentage change in the prices of a basket of goods and services consumed by households.

6.0 Extension of Descriptive Analysis – Data Visualization

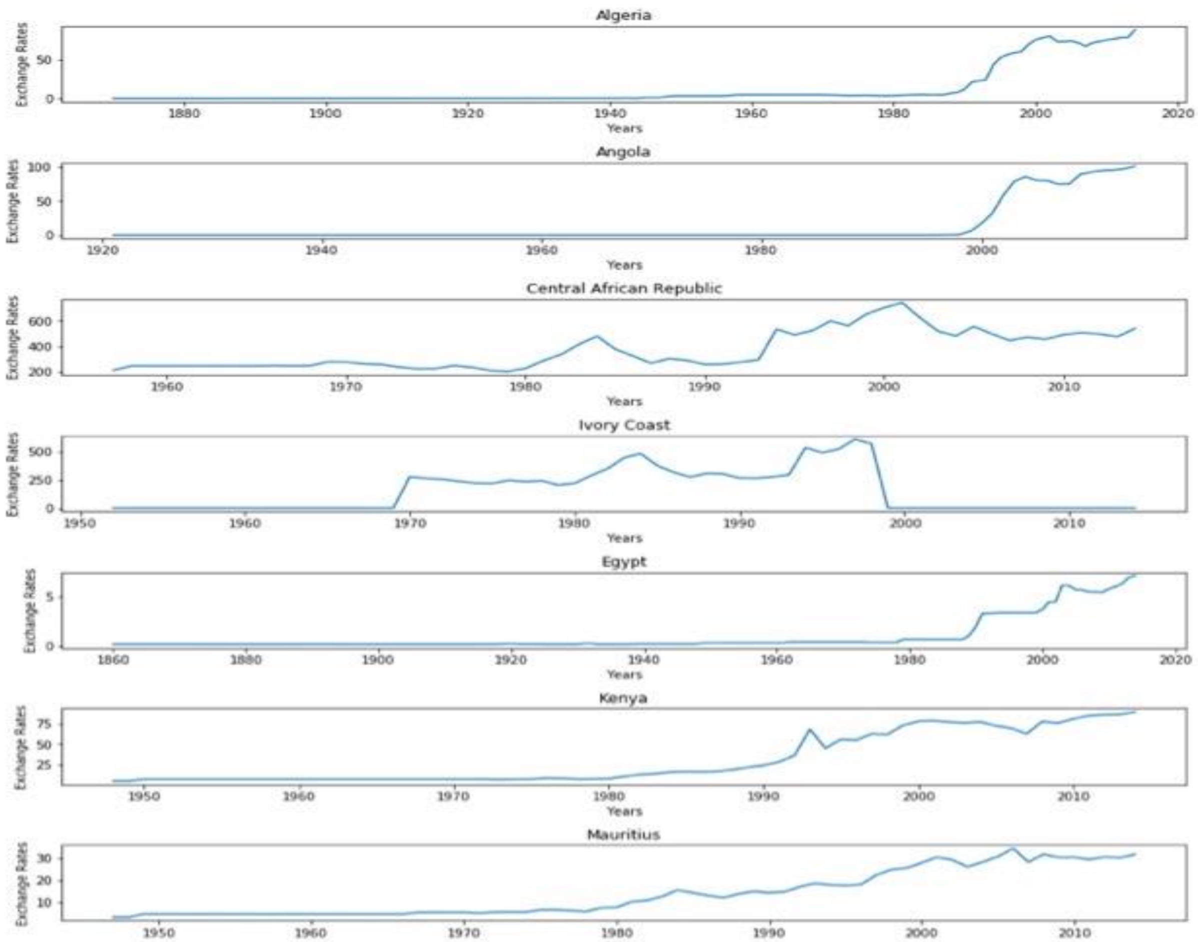
6.1 Country with high number of Systemic Crisis:

The logistic regression model has enabled us to identify which factors are most associated with the banking crisis in Africa.
The country that shows the highest count for systemic crisis is Central African Republic (CAR) followed by Zimbabwe and Kenya.
As systemic crisis normally accompanies with banking crisis, we have checked whether the countries shortlisted by us had a banking crisis at the same time.
6.2 Correlation of Banking Crisis with Systemic Crisis:

As economists normally feel that banking crisis precedes or follows the Systemic Crisis, we have also looked at the correlation between the Banking Crisis and Systemic Crisis.
We find that there is an overlap confirming that systemic crisis has an impact on banking crisis. Our independent validations given below also establish this fact.

6.3 Exchange Value and Exchange Crisis

The movement in the Exchange rate of the country vs USD is captured first by us. The currencies of a few countries such as Angola, Kenya and Mauritius are more or less stable for a long period, they are volatile in others as shown in the Graph-1 below. It is further analyzed by our independent analysis furnished in Section 8.3.



Graph 1

6.4 Sovereign Debt Defaults

6.4.1 Domestic Debt Defaults

Only Angola and Zimbabwe are seen defaulted on the sovereign debts when they faced banking crisis. Around late 1970's, Angola defaulted on domestic debt even though it did not have a banking crisis. After that, it was only around the beginning of 1990, the country faced a banking crisis, which lead the country to default again. Zimbabwe was already facing a banking crisis, which eventually led it to default on domestic debts.

6.4.2 External Debt Defaults

We also find that Central African Republic, Ivory Coast and Zimbabwe defaulting on External Debt.

We furnished our further independent analysis in Section 8.5 below, which also corelates the above results.

7.0 Prescriptive Analysis

Logistic Regression Model is used by for our business analysis and visualization. The reason for choosing Logistic regression model is because our dependent variable, in other words, target is categorical in nature.

7.1 Mechanism of the Model

Logistic regression uses an equation as the representation, very much like linear regression. Input values (x) are combined linearly using weights or coefficient values (referred to as the Greek capital letter Beta) to predict an output value (y). A key difference from linear regression is that the output value being modelled is a binary value (0 or 1) rather than a numeric value.

Below is an example logistic regression equation:

$$y = e^{(b0 + b1*x)} / (1 + e^{(b0 + b1*x)})$$

Where y is the predicted output, b0 is the bias or intercept term and b1 is the coefficient for the single input value (x). Each column in your input data has an associated b coefficient (a constant real value) that must be learned from your training data.

The actual representation of the model that you would store in memory or in a file are the coefficients in the equation (the beta value or b's).

7.2 Input variables:

The Input variables and predict variable for the model are as below.

- systemic_crisis: "0" means that no systemic crisis occurred in the year and "1" means that a systemic crisis occurred in the year.
- exch_usd: The exchange rate of the country vis-a-vis the USD.
- domestic_debt_in_default: "0" means that no sovereign domestic debt default occurred in the year and "1" means that a sovereign domestic debt default occurred in the year.
- sovereign_external_debt_default: "0" means that no sovereign external debt default occurred in the year and "1" means that a sovereign external debt default occurred in the year.
- gdp_weighted_default: The total debt in default vis-a-vis the GDP.
- inflation_annual_cpi: The annual CPI Inflation rate.
- independence: "0" means "no independence" and "1" means "independence".
- currency_crises: "0" means that no currency crisis occurred in the year and "1" means that a currency crisis occurred in the year.
- inflation_crises: "0" means that no inflation crisis occurred in the year and "1" means that an inflation crisis occurred in the year.

Predict variable (desired target):

banking_crisis: "no_crisis" means that no banking crisis occurred in the year and "crisis" means that a banking crisis occurred in the year.

7.3 Model validation and evaluation

On examining whether there is imbalance in classes we noticed that

% Of no crisis 91.123701605288
% Of crisis is 8.876298394711

Our classes are imbalanced, and the ratio of no crisis is 91%.

Since our classes are not balanced, SMOTE algorithm is used with Oversampling Technique.

With our training data created, we have up sampled the crisis using the SMOTE algorithm (Synthetic Minority Oversampling Technique) SMOTE: Works by creating synthetic samples from the minor class (crisis) instead of creating copies. Randomly choose one of the k-nearest-neighbours and using it to create similarly, but randomly tweaked new observations. On running the model, the accuracy achieved was 93% and the coefficients are represented below:

	0	1
0	systemic_crisis	[8.013062387097765]
1	exch_usd	[-0.00588172201344592]
2	gdp_weighted_default	[1.3559775722361]
3	Independence	[1.7282525484728413]

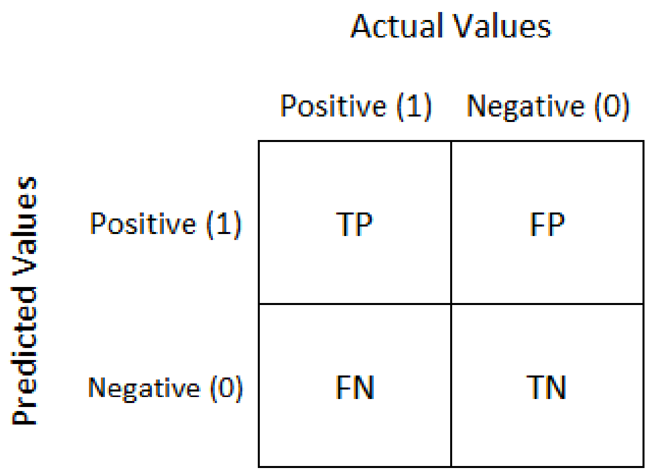
The variable coefficients indicate the change in the log (odds) for a unit change in the variable. The coefficient for the **systemic_crisis** variable is 8.01. This implies that, if the **systemic_crisis** variable increases by 1, the log(odds) will increase by 6.46 and, hence, the probability of the **emergence of banking crises** will change accordingly.

The preceding logistic regression model is built on the entire data. Now, the data was split into training and testing sets, building the model using the training set, and then checking the accuracy using the testing set. The purpose of this exercise is to see whether it improves the accuracy of the prediction or not:

The accuracy of the test classifier is 95%

7.3.1 Confusion Matrix:

Confusion Matrix is a performance measurement for machine learning classification problem where output can be two or more classes. It is a table with 4 different combinations of predicted and actual values.



< The above figure is general representation, however in our scenario positive is replaced with Banking crisis and Negative is replaced with No banking crisis>

[[205   0]
[21   182]]

7.3.2 Evaluation Metrics

7.4 Common terms

Sensitivity: the ability of the model to correctly identify the country with the crisis.
Specificity: the ability of the model to correctly identify a country without crisis
True positive: the country is facing a crisis and the model predicts it as positive.
True- negative: the country is not facing a crisis and the model predicts it as negative.
False-positive: the country is not facing a crisis, but the model predicts it as positive.
False-negative: The country is facing a crisis, but the output is negative.

Positive as in the country is facing a crisis.
Negative: the country is not facing a crisis.

Sensitivity: True positive / (True positive + False negative)
: 205 / (205+21)
: 0.90
This indicates that the model will 90% identify a country with crisis.

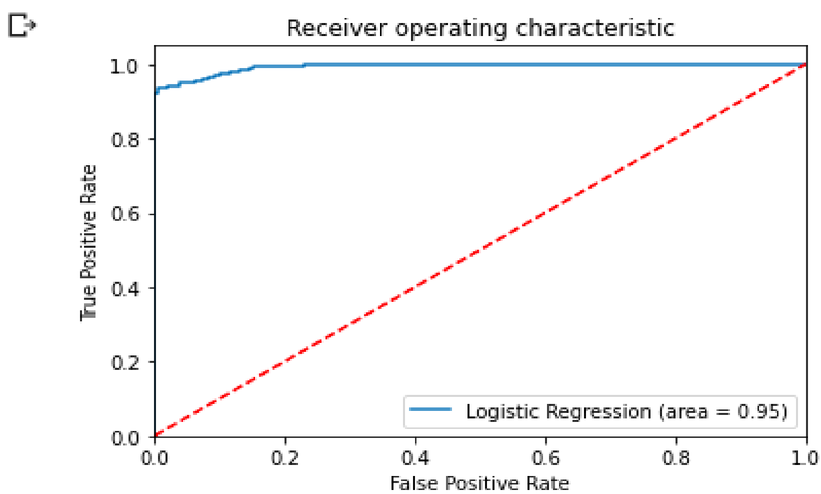
Specificity: True negative / (True negative + False positive)
: 182/ (182+0)
: 1
This indicates that the model will (100%) correctly determine countries without crisis.

7.5 ROC CURVE

ROC Curve summarizes the trade-off between the true positive rate and false-positive rate for our predictive model using different probability thresholds. ROC curves also give us the ability to assess the performance of the classifier over its entire operating range.

7.6 AUC CURVE

The area under the curve is 0.95 which is approximately close to 1, so the model is fairly good. However, AUC is used to compare more than 1 classifier.



8.0 Further Independent Validations by Analyzing Data Points Further

We have carried out an independent analysis outside the purview of the model to validate the test results emanated from the Model and our observations are under:

8.1 Country with high number of Systemic Crisis:

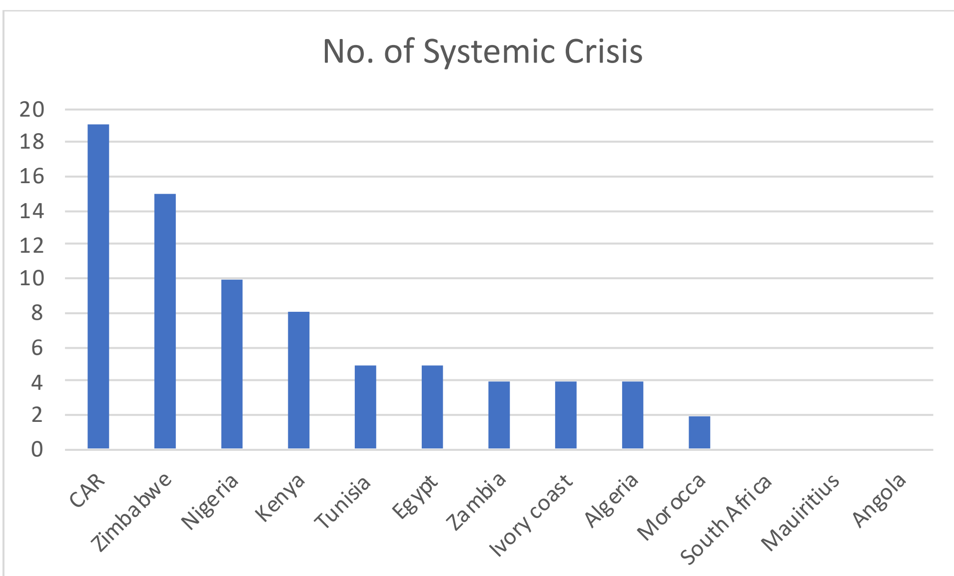
As mentioned in Section 6.3 above, the Model shows that the highest count for systemic crisis is Central African Republic (CAR) followed by Zimbabwe and Kenya.

Our independent analysis has also thrown up the same results. The country that shows the highest count for systemic crisis is Central African Republic followed by Zimbabwe, Nigeria, and Kenya.

Countries with number of Systemic Crisis

Country	No. of Systemic Crisis
CAR	19
Zimbabwe	15
Nigeria	10
Kenya	8
Tunisia	5
Egypt	5
Zambia	4
Ivory coast	4
Algeria	4
Morocco	2
South Africa	0
Mauritius	0
Angola	0

Table 1



Graph-2

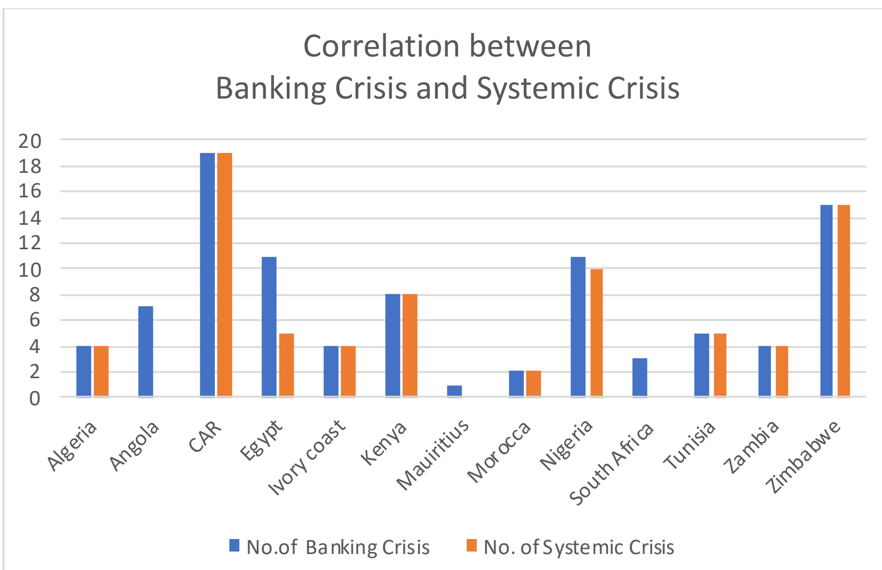
8.2 Correlation of Banking Crisis with Systemic Crisis:

It is established that in the Year of Banking Crisis, there exists the Systemic Crisis in almost all the countries under observation, thus establishing that a high correlation exists between the two events. This is in line with the test results emanated from the Model as mentioned in Section 6.4 above. The following table and the diagram capture the number of such occurrences during 1840-1914:

No. of occurrence of Banking and Systemic Crisis during 1840-2014

Country	No. of Banking Crisis	No. of Systemic Crisis
Algeria	4	4
Angola	7	0
CAR	19	19
Egypt	11	5
Ivory coast	4	4
Kenya	8	8
Mauritius	1	0
Morocco	2	2
Nigeria	11	10
South Africa	3	0
Tunisia	5	5
Zambia	4	4
Zimbabwe	15	15

Table-2



Graph-3

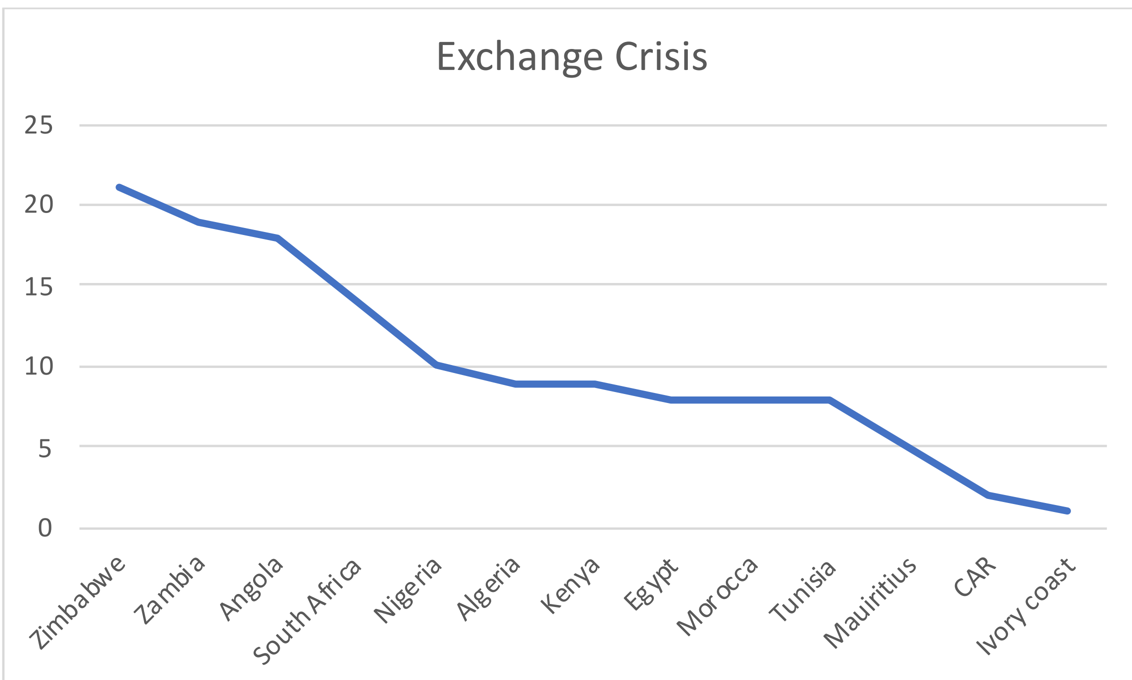
8.3 Our observations - Exchange Crisis:

In order to validate our results on exchange crisis as mentioned in Section 6.5 above, we have done an independent validation as below:

No. of occurrence of Exchange Crisis

Country	Exchange Crisis
Zimbabwe	21
Zambia	19
Angola	18
South Africa	14
Nigeria	10
Algeria	9
Kenya	9
Egypt	8
Morocco	8
Tunisia	8
Mauritius	5
CAR	2
Ivory coast	1

Table-4



Graph -4

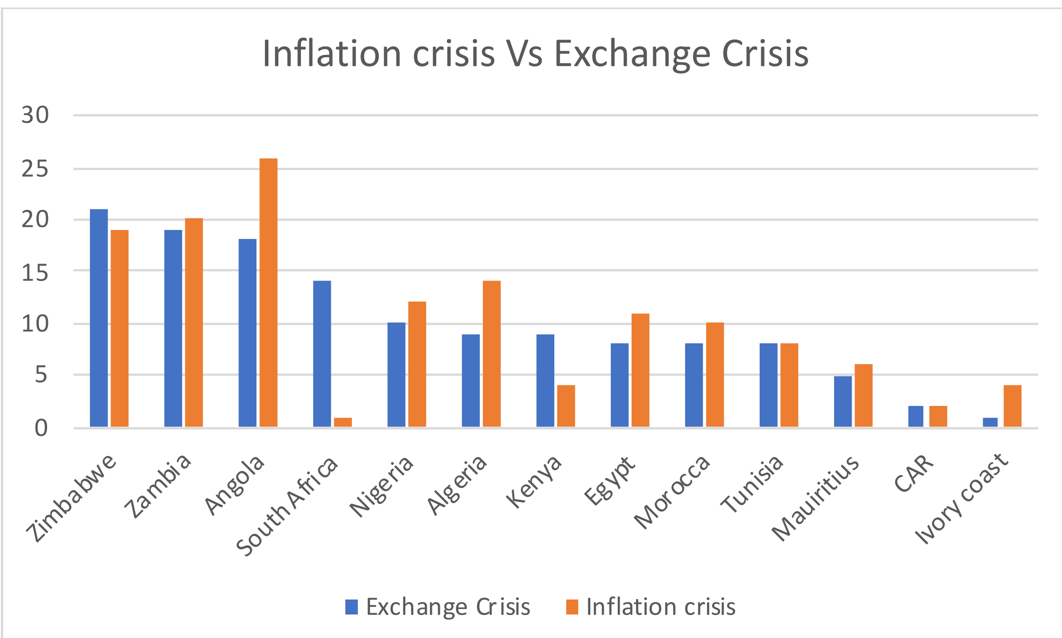
Zimbabwe faced highest number of exchange crisis, followed by Zambia, Angola etc. as shown in the Table-4 and Graph-4, during the period of our study.

8.4 Exchange Crisis Vs Inflation Crisis

We have further done an analysis to establish the correlation between the exchange crisis and inflation crisis which is depicted below:

Country	Exchange Crisis	Inflation Crisis
Zimbabwe	21	19
Zambia	19	20
Angola	18	26
South Africa	14	1
Nigeria	10	12
Algeria	9	14
Kenya	9	4
Egypt	8	11
Morocco	8	10
Tunisia	8	8
Mauritius	5	6
CAR	2	2
Ivory coast	1	4

Table-5



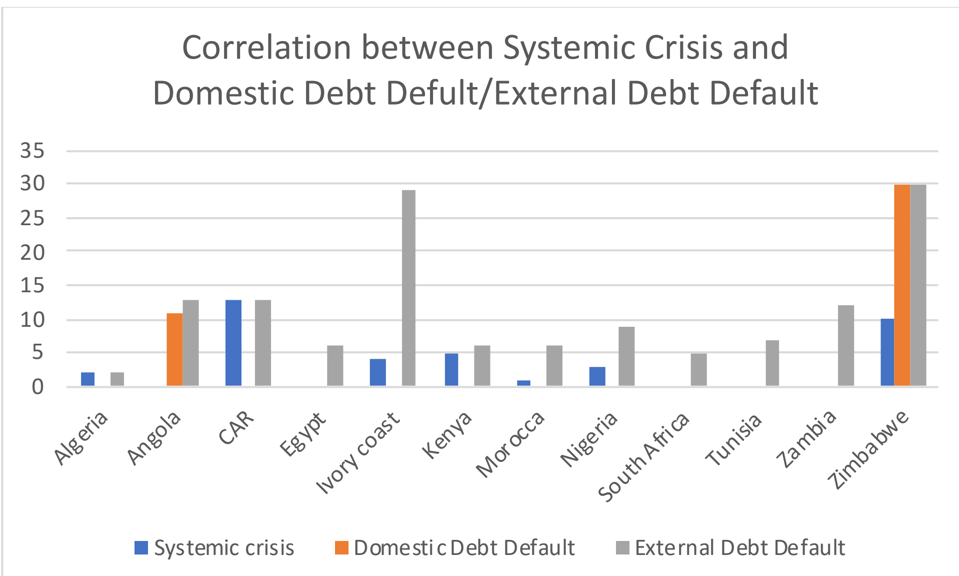
Graph-5

8.5 Sovereign Debt Defaults

In addition to the observations in Section 4.7, our independent analysis also did not establish a direct relationship between Systemic Crisis and Domestic/Eternal Debt Defaults. In case of Angola and Zimbabwe, the domestic and external debt defaults have occurred together, but not necessarily resulted in systemic crisis.

Country	Systemic crisis	Domestic Debt Default	External Debt Default
Algeria	2	0	2
Angola	0	11	13
CAR	13	0	13
Egypt	0	0	6
Ivory coast	4	0	29
Kenya	5	0	6
Morocco	1	0	6
Nigeria	3	0	9
South Africa	0	0	5
Tunisia	0	0	7
Zambia	0	0	12
Zimbabwe	10	30	30

Table-6



Graph - 6

9.0 Observation and Conclusions

The regression models helped us to analyze the events witnessed in the African countries and validate the test results.

Our analysis through establishes the initial findings submitted in our earlier paper that Zimbabwe faced the worst currency crisis, coupled with hyper-inflation, banking and sovereign defaults leading to Systemic crisis. The country's economic woes have continued beyond 2000 and continues to struggle. Even though Zimbabwe attempted a fixed exchange rate of USD 1 / ZWL 57 in June 2020, things did not improve.

Further information collected from the data available on the public domain and analyzed in Section 8 above also validates the results. It is said that the policies of the Governments have led to further financial crisis in countries such as Zimbabwe. In order to tide over the currency fluctuations due to it volatile foreign exchange rates, Zimbabwe banned all mobile money services, but consumer and business units have suffered as most of the transactions are done through such e-services, crippling the sagging economy. Currency devaluation also caused doubling of fuel prices.

Central African Republic (CAR) had wide variety of crisis. This is also validated by the fact that the world has witnessed recession in the early 1980s. During this period CAR has also witnessed high inflationary pressure.

In a nutshell, banking crises in Africa is significantly correlated with four macroeconomics factors:

- systemic_crisis
- gdp_weighted_default
- exch_usd
- independence

10.0 Model – Python File

sruti raman and Praveen raj 18bce7102 , 18bce7088

```
from google.colab import files


uploaded = files.upload()
```

Choose Files no files selected Upload widget is only available when the cell has been executed in the current browser session. Please rerun this cell to enable.

```
Saving african_crises.csv to african_crises.csv
```

```
[ ] import pandas as pd
    import io

    data = pd.read_csv(io.BytesIO(uploaded['african_crises.csv']))
    print(data)
```

```
      case  cc3   country  ...  currency_crises  inflation_crises  banking_crisis
0        1  DZA   Algeria  ...                0                 0          crisis
1        1  DZA   Algeria  ...                0                 0       no_crisis
2        1  DZA   Algeria  ...                0                 0       no_crisis
3        1  DZA   Algeria  ...                0                 0       no_crisis
4        1  DZA   Algeria  ...                0                 0       no_crisis
...    ...  ...       ...  ...              ...               ...             ...
1054    70  ZWE  Zimbabwe  ...                1                 0          crisis
1055    70  ZWE  Zimbabwe  ...                0                 0       no_crisis
1056    70  ZWE  Zimbabwe  ...                0                 0       no_crisis
1057    70  ZWE  Zimbabwe  ...                0                 0       no_crisis
1058    70  ZWE  Zimbabwe  ...                0                 0       no_crisis

[1059 rows x 14 columns]
```

```
[ ] data.shape
```

```
(1059, 14)
```

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1059 entries, 0 to 1058
Data columns (total 14 columns):
 #   Column                           Non-Null Count  Dtype  
---  ------                           --------------  -----  
 0   case                             1059 non-null   int64  
 1   cc3                              1059 non-null   object 
 2   country                          1059 non-null   object 
 3   year                             1059 non-null   int64  
 4   systemic_crisis                  1059 non-null   int64  
 5   exch_usd                         1059 non-null   float64
 6   domestic_debt_in_default         1059 non-null   int64  
 7   sovereign_external_debt_default  1059 non-null   int64  
 8   gdp_weighted_default             1059 non-null   float64
 9   inflation_annual_cpi             1059 non-null   float64
 10  independence                     1059 non-null   int64  
 11  currency_crises                  1059 non-null   int64  
 12  inflation_crises                 1059 non-null   int64  
 13  banking_crisis                   1059 non-null   object 
dtypes: float64(3), int64(8), object(3)
memory usage: 116.0+ KB
```

```
[ ] import pandas as pd
    import numpy as np
    import matplotlib.pyplot as plt
    from sklearn.linear_model import LogisticRegression
    from sklearn.model_selection import train_test_split
    import seaborn as sns

    %matplotlib inline
```

```
[ ] data.shape
```

```
(1059, 14)
```

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1059 entries, 0 to 1058
Data columns (total 14 columns):
 #   Column                            Non-Null Count  Dtype
---  ------                            --------------  -----
 0   case                              1059 non-null   int64
 1   cc3                               1059 non-null   object
 2   country                           1059 non-null   object
 3   year                              1059 non-null   int64
 4   systemic_crisis                   1059 non-null   int64
 5   exch_usd                          1059 non-null   float64
 6   domestic_debt_in_default          1059 non-null   int64
 7   sovereign_external_debt_default   1059 non-null   int64
 8   gdp_weighted_default              1059 non-null   float64
 9   inflation_annual_cpi              1059 non-null   float64
 10  independence                      1059 non-null   int64
 11  currency_crises                   1059 non-null   int64
 12  inflation_crises                  1059 non-null   int64
 13  banking_crisis                    1059 non-null   object
dtypes: float64(3), int64(8), object(3)
memory usage: 116.0+ KB
```

```
[ ] import pandas as pd
    import numpy as np
    import matplotlib.pyplot as plt
    from sklearn.linear_model import LogisticRegression
    from sklearn.model_selection import train_test_split
    import seaborn as sns

    %matplotlib inline
```

```
print("Proportion of no crisis data in oversampled data is ",len(os_data_Y[os_data_Y['Y']==0])/len(os_data_X))
print("Proportion of crisis data in oversampled data is ",len(os_data_Y[os_data_Y['Y']==1])/len(os_data_X))
```

```
length of oversampled data is  1360
Number of crisis in oversampled data 680
Number of no crisis 680
Proportion of no crisis data in oversampled data is  0.5
Proportion of crisis data in oversampled data is  0.5
/usr/local/lib/python3.7/dist-packages/sklearn/utils/validation.py:760: DataConversionWarning: A column-vector y was passed when a 1d array was expected. Ple
  y = column_or_1d(y, warn=True)
/usr/local/lib/python3.7/dist-packages/sklearn/utils/deprecation.py:87: FutureWarning: Function safe_indexing is deprecated; safe_indexing is deprecated in
  warnings.warn(msg, category=FutureWarning)
```

```
X=os_data_X
Y=os_data_Y
import statsmodels.api as sm
logit_model=sm.Logit(Y,X)
result=logit_model.fit()
print(result.summary2())
```

```
X=os_data_X
Y=os_data_Y
import statsmodels.api as sm
logit_model=sm.Logit(Y,X)
result=logit_model.fit()
print(result.summary2())
```

```
/usr/local/lib/python3.7/dist-packages/statsmodels/tools/_testing.py:19: FutureWarning: pandas.util.
  import pandas.util.testing as tm
Optimization terminated successfully.
         Current function value: inf
         Iterations 12
                                Results: Logit
================================================================================
Model:                    Logit                  Pseudo R-squared:      inf
Dependent Variable:       Y                      AIC:                   inf
Date:                     2021-06-11 15:43       BIC:                   inf
No. Observations:         1360                   Log-Likelihood:        -inf
Df Model:                 8                      LL-Null:               0.0000
Df Residuals:             1351                   LLR p-value:           1.0000
Converged:                1.0000                 Scale:                 1.0000
No. Iterations:           12.0000
--------------------------------------------------------------------------------
                                Coef.  Std.Err.    z     P>|z|   [0.025  0.975]
--------------------------------------------------------------------------------
systemic_crisis                17.8266   2.0249   8.8035 0.0000 13.8578 21.7954
exch_usd                       -0.0155   0.0041  -3.7812 0.0002 -0.0236 -0.0075
domestic_debt_in_default        4.0911   2.0614   1.9846 0.0472  0.0508  8.1315
sovereign_external_debt_default -4.4215  1.9431  -2.2755 0.0229 -8.2300 -0.6131
gdp_weighted_default           17.2062   5.5139   3.1205 0.0018  6.3992 28.0132
inflation_annual_cpi            0.0027   0.0049   0.5447 0.5860 -0.0069  0.0122
independence                   -1.9927   0.1550 -12.8548 0.0000 -2.2965 -1.6888
currency_crises                 0.1922   0.3250   0.5914 0.5543 -0.4448  0.8293
inflation_crises               -0.1774   0.3441  -0.5155 0.6062 -0.8518  0.4971
================================================================================
```

```
# Divide the data into “attributes”  X and “labels”  Y
X = data[['systemic_crisis', 'exch_usd', 'domestic_debt_in_default',
       'sovereign_external_debt_default', 'gdp_weighted_default',
       'inflation_annual_cpi', 'independence', 'currency_crises',
       'inflation_crises']]
# Define the Y variable
Y = data['banking_crisis']

## percentage of  crisis & no crisis

count_no_crisis = len(data[Y=='no_crisis'])
count_crisis = len(data[Y=='crisis'])
pct_of_no_crisis = count_no_crisis/(count_no_crisis+count_crisis)
print("percentage of no crisis is", pct_of_no_crisis*100)
pct_of_crisis = count_crisis/(count_no_crisis+count_crisis)
print("percentage of crisis", pct_of_crisis*100)
```

```
percentage of no crisis is 91.123701605288
percentage of crisis 8.876298394711993
```

```
Y= pd.get_dummies(Y)
Y = Y.drop(['no_crisis'], axis = 1)

# Over-sampling using SMOTE
from imblearn.over_sampling import SMOTE
os = SMOTE(random_state=0)
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.3, random_state=0)
columns = X_train.columns
os_data_X,os_data_Y=os.fit_sample(X_train, Y_train)
os_data_X = pd.DataFrame(data=os_data_X,columns=columns )
os_data_Y= pd.DataFrame(data=os_data_Y,columns=['Y'])
# we can Check the numbers of our data
print("length of oversampled data is ",len(os_data_X))
print("Number of crisis in oversampled data",len(os_data_Y[os_data_Y['Y']==1]))
print("Number of no crisis",len(os_data_Y[os_data_Y['Y']==0]))
print("Proportion of no crisis data in oversampled data is ",len(os_data_Y[os_data_Y['Y']==0])/len(os_data_X))
```

```
length of oversampled data is  1360
Number of crisis in oversampled data 680
Number of no crisis 680
Proportion of no crisis data in oversampled data is  0.5
Proportion of crisis data in oversampled data is  0.5
/usr/local/lib/python3.7/dist-packages/sklearn/utils/validation.py:760: DataConversionWarning:
  y = column_or_1d(y, warn=True)
/usr/local/lib/python3.7/dist-packages/sklearn/utils/deprecation.py:87: FutureWarning: Functio
  warnings.warn(msg, category=FutureWarning)
```

```
X=os_data_X
Y=os_data_Y
import statsmodels.api as sm
logit_model=sm.Logit(Y,X)
result=logit_model.fit()
print(result.summary2())
```

```
/usr/local/lib/python3.7/dist-packages/statsmodels/tools/_testing.py:19: FutureWarning: pandas.util.
  import pandas.util.testing as tm
Optimization terminated successfully.
         Current function value: inf
         Iterations 12
                                Results: Logit
===============================================================================
Model:                     Logit                  Pseudo R-squared:       inf
Dependent Variable:        Y                      AIC:                    inf
Date:                      2021-06-11 15:43       BIC:                    inf
No. Observations:          1360                   Log-Likelihood:         -inf
Df Model:                  8                      LL-Null:                0.0000
Df Residuals:              1351                   LLR p-value:            1.0000
Converged:                 1.0000                 Scale:                  1.0000
No. Iterations:            12.0000
-------------------------------------------------------------------------------
                                Coef.  Std.Err.    z     P>|z|   [0.025  0.975]
-------------------------------------------------------------------------------
systemic_crisis               17.8266   2.0249   8.8035 0.0000 13.8578 21.7954
exch_usd                      -0.0155   0.0041  -3.7812 0.0002 -0.0236 -0.0075
domestic_debt_in_default       4.0911   2.0614   1.9846 0.0472  0.0508  8.1315
sovereign_external_debt_default -4.4215  1.9431  -2.2755 0.0229 -8.2300 -0.6131
gdp_weighted_default          17.2062   5.5139   3.1205 0.0018  6.3992 28.0132
inflation_annual_cpi           0.0027   0.0049   0.5447 0.5860 -0.0069  0.0122
independence                  -1.9927   0.1550 -12.8548 0.0000 -2.2965 -1.6888
currency_crises                0.1922   0.3250   0.5914 0.5543 -0.4448  0.8293
inflation_crises              -0.1774   0.3441  -0.5155 0.6062 -0.8518  0.4971
===============================================================================
```

```
[ ] clf.score(X,Y)
```

```
0.9419117647058823
```

```
pd.DataFrame(zip(X.columns, np.transpose(clf.coef_)))
```

	0	1
0	systemic_crisis	[7.816041413743628]
1	exch_usd	[-0.004404696065513252]
2	domestic_debt_in_default	[-0.4835507575888725]
3	sovereign_external_debt_default	[-0.3186639823725079]
4	gdp_weighted_default	[1.1656656632786104]
5	inflation_annual_cpi	[0.005292633762368325]
6	independence	[2.497553497860037]
7	currency_crises	[0.27337052469871886]
8	inflation_crises	[0.8377153762950065]

```
[ ] from sklearn.linear_model import LogisticRegression
    from sklearn import metrics
    X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.3, random_state=0)
    logreg = LogisticRegression()
    logreg.fit(X_train, Y_train)
```

```
from sklearn import linear_model
clf = linear_model.LogisticRegression()
clf.fit(X, Y)
```

```
/usr/local/lib/python3.7/dist-packages/sklearn/utils/validation.py
  y = column_or_1d(y, warn=True)
/usr/local/lib/python3.7/dist-packages/sklearn/linear_model/_logis
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver opti
    https://scikit-learn.org/stable/modules/linear_model.html#logi
  extra_warning_msg=_LOGISTIC_SOLVER_CONVERGENCE_MSG)
LogisticRegression(C=1.0, class_weight=None, dual=False, fit_inter
                   intercept_scaling=1, l1_ratio=None, max_iter=10
                   multi_class='auto', n_jobs=None, penalty='l2',
                   random_state=None, solver='lbfgs', tol=0.0001,
                   warm_start=False)
```

```
[ ] clf.score(X,Y)
```

```
0.9419117647058823
```

```
[ ] Y_pred = logreg.predict(X_test)
    print('Accuracy of logistic regression classifier on test set: {:.2f}'.format(logreg.score(X_test,
```

```
Accuracy of logistic regression classifier on test set: 0.95
```

```
[ ] from sklearn.metrics import confusion_matrix
    confusion_matrix = confusion_matrix(Y_test, Y_pred)
    print(confusion_matrix)
```

```
[[205   0]
 [ 20 183]]
```

```
from sklearn.metrics import roc_auc_score
from sklearn.metrics import roc_curve
logit_roc_auc = roc_auc_score(Y_test, logreg.predict(X_test))
fpr, tpr, thresholds = roc_curve(Y_test, logreg.predict_proba(X_test)[:,1])
plt.figure()
plt.plot(fpr, tpr, label='Logistic Regression (area = %0.2f)' % logit_roc_auc)
plt.plot([0, 1], [0, 1],'r--')
plt.xlim([0.0, 1.0])
plt.ylim([0.0, 1.05])
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('Receiver operating characteristic')
plt.legend(loc="lower right")
plt.savefig('Log_ROC')
plt.show()
```