

ASSIGNING UGANDAN BUSINESSES A CREDIT SCORE PLUS PREDICTING DEFAULTS

```
In [41]: import matplotlib.pyplot as plt
%matplotlib inline
import pandas as pd
import numpy as np
import plotly.offline as pyo
pyo.init_notebook_mode(connected = True)
import plotly.express as px
from plotly.subplots import make_subplots
import plotly.graph_objects as go
import seaborn as sns
```

```
In [26]: df = pd.read_csv('responses2.csv')
```

```
In [441]: df.shape
```

```
Out[441]: (15, 32)
```

```
In [5]: pd.options.display.max_columns = None
display(df)
```

	NAME OF BUSINESS	INDUSTRY	NATURE OF SALES	LOCATION	OWNERSHIP	NUMBER OF PEOPLE EMPLOYED	HOW EXPERIENCED ARE YOUR EMPLOYEES	YEARS ENTREPRENEURS
0	Ruzenet finance	Financial Services	Retail	Mukono	Sole owner	3	Not so experienced	
1	BENIVEN SUPERMARKET	Supermarket	Wholesale	KIRA	Sole owner	6	Not so experienced	
2	Ruzenet electronics	Electronics	Retail	Wakiso, Kira mamerito road	Sole owner	1	Not so experienced	
3	Vossi Wines & Spirits	Wines and spirits	Retail	Kulambiro	Partnership (General or limited)	1	part time	
4	Twinex Gadgets	Electronics	Wholesale	Kampala	Family owned	1	part time	
5	Maxon Capital	Financial Services	Retail	Kampala	Sole owner	3	Experienced	
6	INNOVATION AFRICA LIMITED	Construction	other	Plot 758, Cape road, Munyonyo	Partnership (General or limited)	41	Experienced	
7	Bussi coffee and Bogoya farm	Agriculture	Wholesale	Bussi Island	Family owned	10	Experienced	
8	Delique industrial coating Ltd	Manufacturing	Retail	Kasangati	Partnership (General or limited)	18	Experienced	
9	BUSITEMA FILLING STATION	Retail	Retail	Busitema, Busia	Sole owner	4	part time	

	NAME OF BUSINESS	INDUSTRY	NATURE OF SALES	LOCATION	OWNERSHIP	NUMBER OF PEOPLE EMPLOYED	HOW EXPEREINCED ARE YOUR EMPLOYEES	YEARS ENTREPRENEURS
10	Melcorp Media	Media	other	Ntinda	Partnership (General or limited)	2	Experienced	
11	Elevate	Media	Retail	MOTIV , Old Port Bell Road	Partnership (General or limited)	15	Experienced	
12	CORNERSTONE HIGH SCHOOL- NANGABO	Education	other	KASANGATI	Family owned	29	Experienced	
13	Fine creations	clothings	Retail	Kampala	Sole owner	1	Not so experienced	
14	Helpware	digital	other	Kampala	Partnership (General or limited)	30	Experienced	

WE START BY LOADING UP THE DATASET WHICH COMPRISED OF 13 BUSINESSES ANSWERING A GOOGLE DOC TO ASCERTAIN THIS INFROMATION. THIS INFORMATION WAS EXTRACTED WITH THE HELP OF OWNERS OF THE BUSINESS

```
In [6]: df.dtypes
```

```
Out[6]: NAME OF BUSINESS          object
INDUSTRY          object
NATURE OF SALES   object
LOCATION           object
OWNERSHIP         object
NUMBER OF PEOPLE EMPLOYED    int64
HOW EXPERIENCED ARE YOUR EMPLOYEES    object
YEARS IN ENTREPRENEURSHIP    int64
LEVEL OF FORMAL EDUCATION OF BUSINESS DIRECTOR    object
Product Differentiation/ Unique selling point    object
How do you get people to know about your business/Products    object
METHOD OF FINANCE    object
HOW MUCH REVENUE DOES YOUR BUSINESS MAKE IN A YEAR(UGX)    object
Average Revenue(UGX)    int64
Total Debt exposure (if any)    object
Average debt(UGX)    int64
Profitability of business (operating profit/Revenue)    object
Average Profitability(UGX)    int64
Inventory Management    object
How are you able to get the product/service you sell    object
Team experience(years)    int64
Human resource policies    object
Registration & Licenses    object
Risk Management Scheme    object
What stage of Growth is your business    int64
Average weekly customers    int64
what is your supply chain like    object
Nature of skill required for employees    object
Employee monthly pay(UGX)    object
Average Monthly pay (UGX)    int64
Annual loss range    object
Average Annual loss(UGX)    int64
dtype: object
```

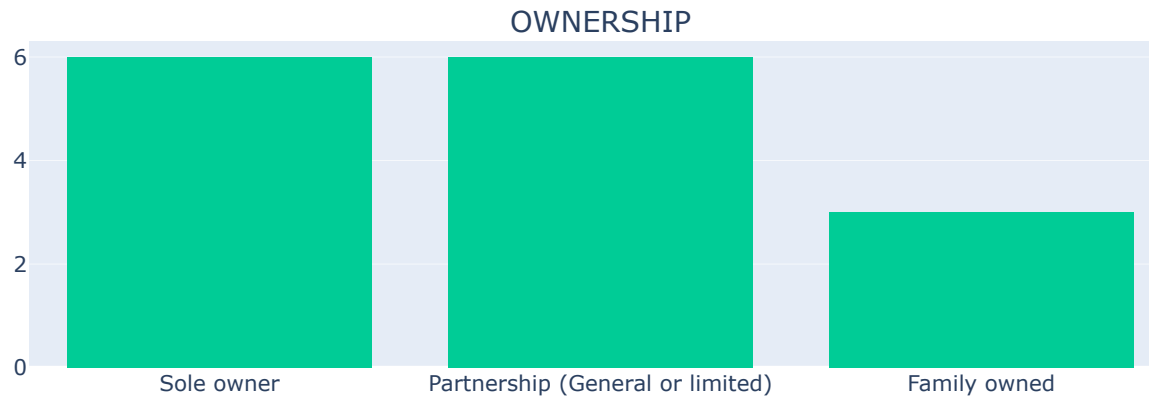
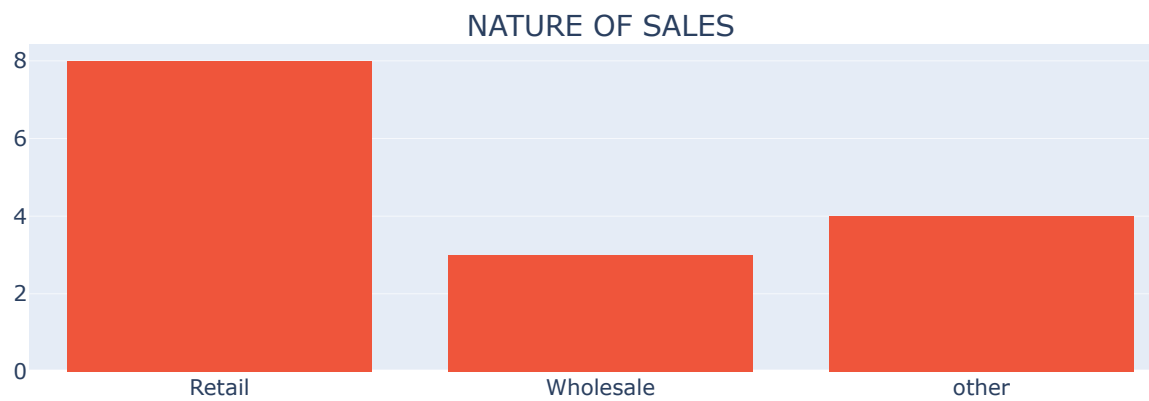
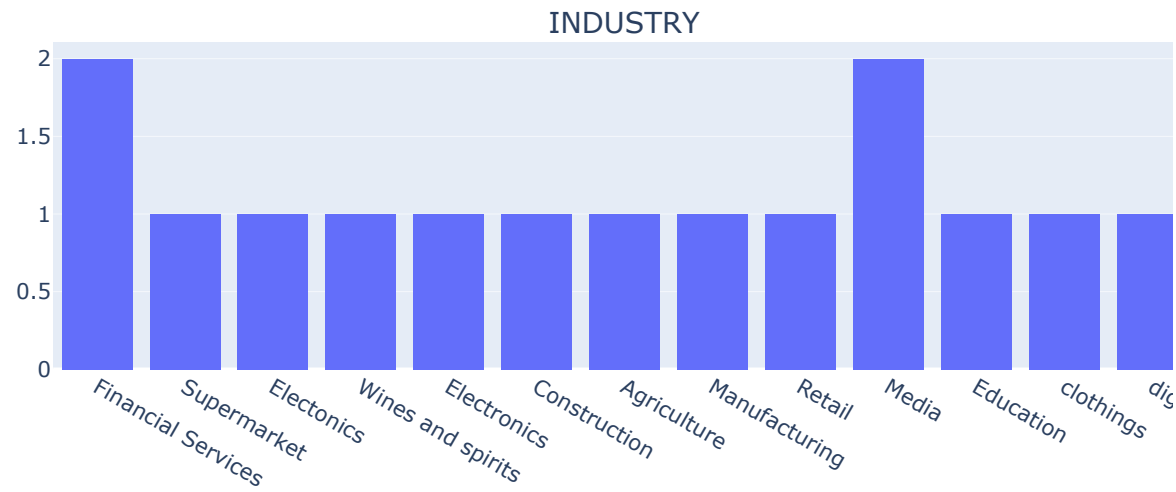
THE INFORMATION COLLECTED ALSO SEEMS TO PROVIDE INSIGHTS INTO COMMON UGANDAN BUSINESS TROPES DUE TO THE ECONOMY OF UGANDA WHICH WE EXPLORE BELOW

```
In [7]: important_cols = ['INDUSTRY', 'NATURE OF SALES', 'OWNERSHIP', 'NUMBER OF PEOPLE EMPLOYED',
'HOW EXPERIENCED ARE YOUR EMPLOYEES', 'YEARS IN ENTREPRENEURSHIP',
'LEVEL OF FORMAL EDUCATION OF BUSINESS DIRECTOR', 'Product Differentiation/ Unique selling point',
'How do you get people to know about your business/Products', 'METHOD OF FINANCE', 'Average Revenue(UGX)',
'Average debt(UGX)', 'Average Profitability(UGX)', 'Inventory Management',
'How are you able to get the product/service you sell', 'Team experience(years)',
'Human resource policies', 'Registration & Licenses', 'Risk Management Scheme',
'What stage of Growth is your business', 'Average weekly customers', 'what is your supply chain like',
'Nature of skill required for employees', 'Average Monthly pay (UGX)', 'Average Annual loss(UGX)']
```

```
In [36]: df2 = df[important_cols]
cols_to_plot1 = ['INDUSTRY', 'NATURE OF SALES', 'OWNERSHIP']
cols_to_plot2 = ['NUMBER OF PEOPLE EMPLOYED', 'YEARS IN ENTREPRENEURSHIP', 'LEVEL OF FORMAL EDUCATION OF BUSINESS DIRECTOR']
cols_to_plot3 = ['METHOD OF FINANCE', 'Average Revenue(UGX)', 'Average debt(UGX)', 'Average Profitability(UGX)']
num_cols = 1
fig = make_subplots(rows = num_rows, cols = num_cols, subplot_titles = cols_to_plot1)
for index, column in enumerate(cols_to_plot1):
    row = (index // num_cols)+1
    col = (index % num_cols)+1
    fig.add_trace(go.Histogram(x=df2[column], name = column),
                    row = row,
                    col = col)
fig.update_layout(height = 1000, width = 900, title_text = 'Business Tropes in Uganda')
fig.show()
```

```
'Distribution of Answers Concerning the Nature of the Business')  
fig.show()
```

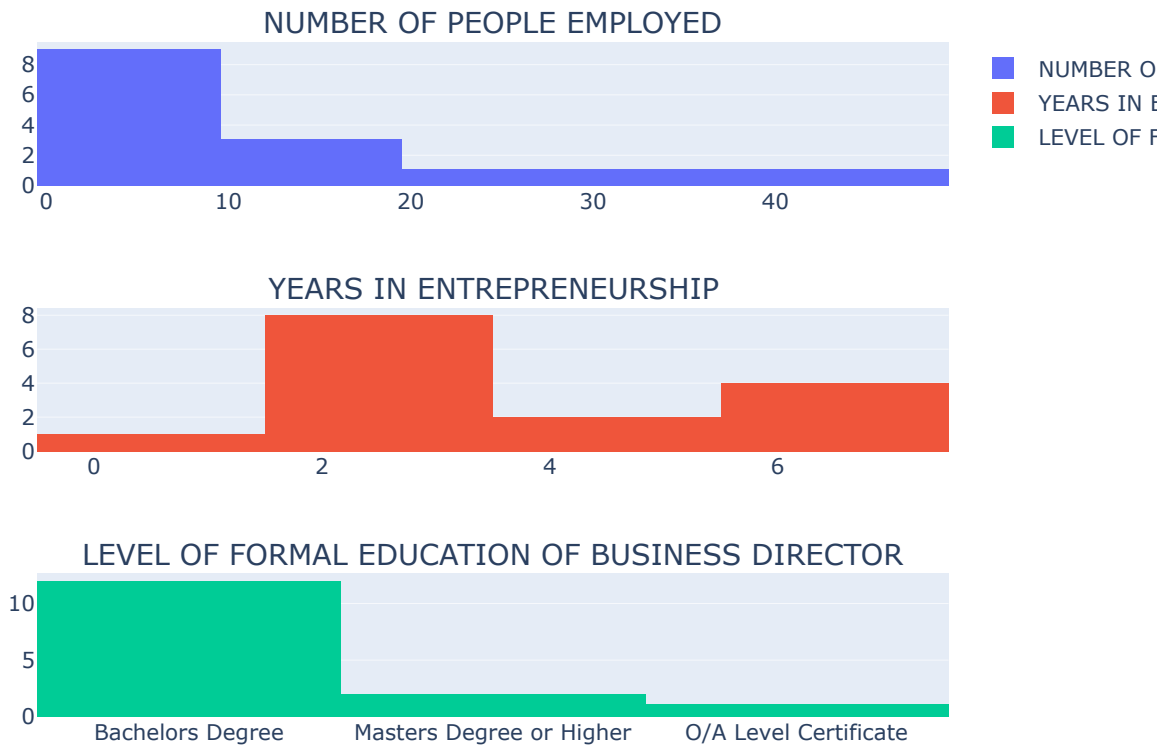
Distribution of Answers Concerning the Nature of the Business



MOST BUSINESS ARE ENGAGED IN RETAIL(LOWER COSTS TO SET UP) AND MAJORITY BUSINESSES ARE SOLE OWNED. ONE OF THE REASONS IS BECAUSE THEIR ARE LIMITED JOBS IN BOTH PUBLIC AND PRIVATE SECTOR TO ABSORB ALL THESE WORKERS SO MOST PEOPLE VENTURE INTO BUSINESS. NO WONDER UGANDA SHOWS UP ON LISTS AS EITHER THE MOST ENTREPRENUERIAL COUNTRY OR AMONG THE TOP 5-10 IN THE WORLD

```
In [446... num_cols = 1
fig = make_subplots(rows = num_rows, cols = num_cols, subplot_titles = cols_to_plot2)
for index, column in enumerate(cols_to_plot2):
    row = (index // num_cols)+1
    col = (index % num_cols)+1
    fig.add_trace(go.Histogram(x=df2[column], name = column),
                    row = row,
                    col = col)
fig.update_layout(height = 1000, width = 1000, title_text =
'Distribution of Answers Concerning the employess and directors of the Business')
fig.show()
```

Distribution of Answers Concerning the employess and directors of the Busin

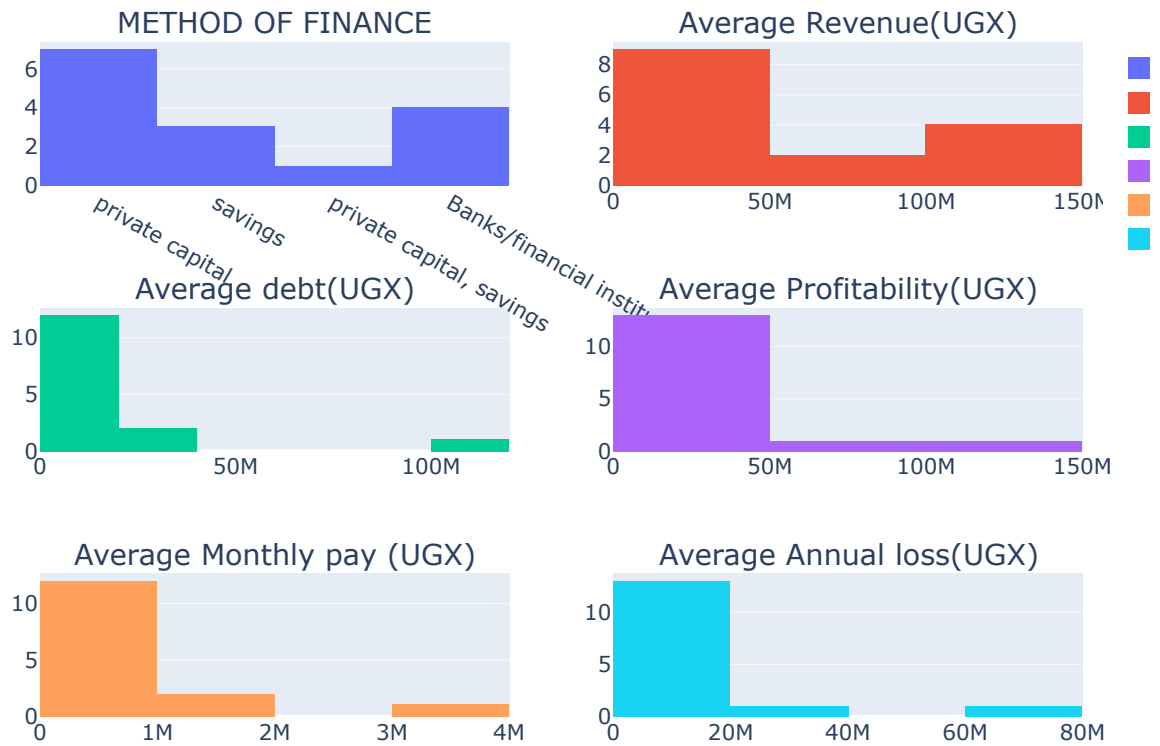


SO HERE WE CAN CONFIRM TWO/THREE THINGS THAT ARE COMMONLY DISCUSSED ABOUT BUSINESS IN UGANDA; 1) SINCE MOST PEOPLE ENGAGE IN RELATIVELY LOW COST RETAIL, THEY EMPLOY FEW PEOPLE AS THE FIRST HISTOGRAM SHOWS.

2) ITS CLEAR FROM THE REMAINING TWO HISTOGRAMS THAT SINCE THEIR ARE LACK OF JOBS, PEOPLE VENTURE OUT INTO BUSINESS AFTER GRADUATION TO MAKE ENDS MEET OWING TO THE HISTOGRM SHOWING MOST BUSINESS OWNERS HAVE A BACHELORS DEGREE WITH VERY FEW ATTAINING HIGHER QUALIFICATIONS BECAUSE THOSE ARE MOSTLY FOR CAREER ADVANCEMENT WHICH IS LOW DUE TO LIMITED JOBS/EMPLOYMENT.

```
In [39]: num_cols = 2
num_rows = 6
fig = make_subplots(rows = num_rows, cols = num_cols, subplot_titles = cols_to_plot3)
for index, column in enumerate(cols_to_plot3):
    row = (index // num_cols)+1
    col = (index % num_cols)+1
    fig.add_trace(go.Histogram(x=df2[column], name = column),
                    row = row,
                    col = col)
fig.update_layout(height = 1000, width = 900, title_text =
'Distribution of Answers Concerning the Nature of Finace of the Business')
fig.show()
```

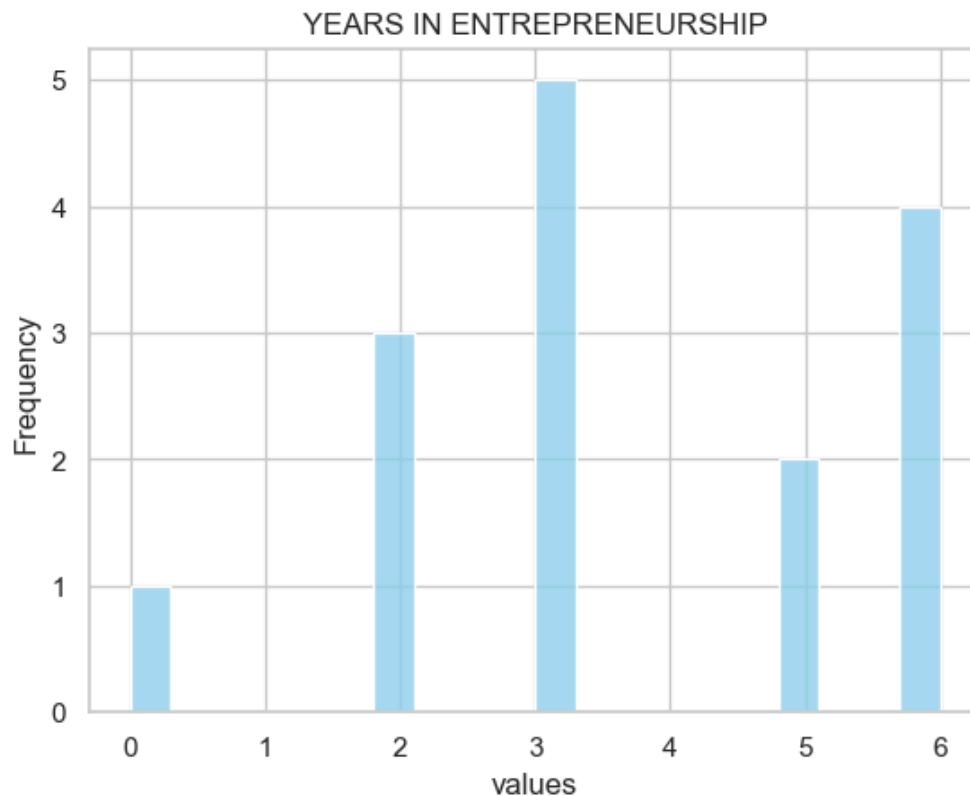

Distribution of Answers Concerning the Nature of Finace of the Business

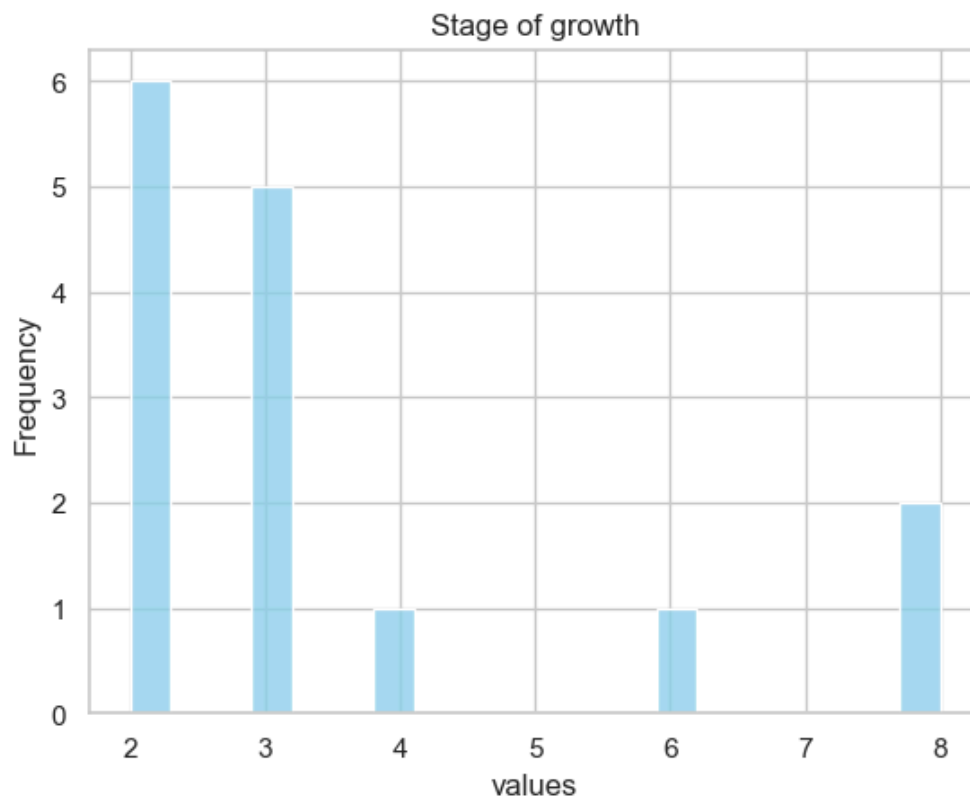


1) MOST BUSINESSES AS WE CAN SEE FROM THE DATASET PREFER PRIVATE CAPITAL/SAVINGS BECAUSE OF HIGH BANK INTEREST RATES(UP TO 15-16%) AS WELL AS BUSINESSES IN UGANDA LACKING FORMALIZATION AND PROPER DOCUMENTATION THAT BANKS REQUIRE.

2) THE 2ND HISTOGRAM SHOWS US THAT MOST BUSINESSES PULL IN REVENUES OF <50M UGX AND A FEW OUTLIERS WHO MAKE MORE THAN THAT, THIS IS DUE TO LOW VOLUMES OF TRANSACTIONS IN UGANDAS ECONOMY WHICH WE CAN INFER FROM GRAPH 5 THAT MOST BUSINESSES DON'T MAKE ENOUGH TO PAY WORKERS A WAGE TO ENCOURAGE AN INCREASE IN VOLUMES OF PURCHASES AND TRANSACTIONS.

```
In [43]: sns.histplot(df2['YEARS IN ENTREPRENEURSHIP'], kde = False, bins = 20, color = 'skyblue')
plt.title('YEARS IN ENTREPRENEURSHIP')
plt.xlabel('values')
plt.ylabel('Frequency')
plt.show()
sns.histplot(df2['What stage of Growth is your business'], kde = False, bins = 20, color = 'skyblue')
plt.title('Stage of growth')
plt.xlabel('values')
plt.ylabel('Frequency')
plt.show()
```





AS MENTIONED EARLIER, UGANDA IS ONE OF THE MOST ENTREPRENEURIAL COUNTRIES IN THE WORLD AS WE CAN SEE FROM THE FIRST GRAPH THAT MOST OF THE RESPONDERS HAVE BEEN IN BUSINESS FOR 2-5 YEARS BUT THE 2ND GRAPH SHOWS ANOTHER EXAMPLE OF THE TYPICAL UGANDAN BUSINESS THAT MOSTLY FAIL IN THE FIRST YEAR AND EACH SUBSEQUENT YEAR BECOMING HARDER AND HARDER TO SUSTAIN BUSINESS WITH MOST RESPONDERS HAVING THE BUSINESS THEY WERE QUESTIONED ON BEING AT 2-3 YEARS OF GROWTH. THIS IS USUALLY DOWN TO LOW VOLUMES OF PURCHASES AND TRANSACTIONS FROM AN ALREADY POORLY PAID POPULACE BUT AS WELL AS THE HIGH AND UNFAIR TAX REGIME IN UGANDA THAT DOESN'T HELP BUSINESSES GROW AND IS MOSTLY JUST EXTRACTIVE.

ASSIGNING WEIGHTS FOR EACH OF THE IMPORTANT QUESTONS TO COME UP WITH A CREDIT SCORE

THE WEIGHTING SYSTEM WILL BE ; O(HIGH RISK)-5(LOW RISK)

In [449...

```
weights1 = {'Sole owner':2.5, 'Family owned':3, 'Partnership (General or limited)':3.5}
df2['OWNERSHIP weights'] = df2['OWNERSHIP'].map(weights1)

def weights2(people_employed):
    if people_employed < 10:
        return 2.5
    elif people_employed < 20:
        return 3
    elif people_employed < 30:
        return 3.5
    elif people_employed < 40:
        return 4
    else:
        return 5
df2['Number of people employed weighted'] = df2['NUMBER OF PEOPLE EMPLOYED'].apply(weights2)
```

C:\Users\jbmad\AppData\Local\Temp\ipykernel_18936\10856210.py:2: SettingWithCopyWarning:

A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy

C:\Users\jbmad\AppData\Local\Temp\ipykernel_18936\10856210.py:15: SettingWithCopyWarning:

A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy

In [450...

```
weights3 = { 'Not so experienced':1.5, 'part time':1, 'Experienced':4}
df2['Employee experience weighted'] = df2['HOW EXPERIENCED ARE YOUR EMPLOYEES'].map(weights3)
df2['Employee experience weighted']
```

C:\Users\jbmad\AppData\Local\Temp\ipykernel_18936\4021022579.py:2: SettingWithCopyWarning:

A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy

Out[450]:

```
0    1.5
1    1.5
2    1.5
3    1.0
4    1.0
5    4.0
6    4.0
7    4.0
8    4.0
9    1.0
10   4.0
11   4.0
12   4.0
13   1.5
14   4.0
Name: Employee experience weighted, dtype: float64
```

In [451...

```
def weights4(Years_in_entrepreneurship):
    if Years_in_entrepreneurship < 1:
        return 1
    elif Years_in_entrepreneurship < 3:
        return 2
    elif Years_in_entrepreneurship < 6:
        return 3
    elif Years_in_entrepreneurship < 8:
        return 4
    else:
        return 5
df2['Entrepreneurship years weighted'] = df2['YEARS IN ENTREPRENEURSHIP'].apply(weights4)
```

C:\Users\jbmad\AppData\Local\Temp\ipykernel_18936\3236732646.py:12: SettingWithCopyWarning:

A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy

In [452...

```
weights5 = { 'Bachelors Degree':2, 'Masters Degree or Higher':3, 'O/A Level Certificate':1}
df2['Education weighted'] = df2['LEVEL OF FORMAL EDUCATION OF BUSINESS DIRECTOR'].map(weights5)
```

C:\Users\jbmad\AppData\Local\Temp\ipykernel_18936\2753066791.py:2: SettingWithCopyWarning:

A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy

In [453...

```
df2['Product Differentiation/Unique selling point weighted'] = [3.5, 2.0, 2.0, 2.0, 2.0, 3.5, 4.0,
```

C:\Users\jbmad\AppData\Local\Temp\ipykernel_18936\1236564112.py:1: SettingWithCopyWarning:

A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy

In [454...

```
weights6 = { 'private capital':2, 'private capital, savings':2, 'savings':2, 'Banks/financial inst'
df2['Method of finance weighted'] = df2['METHOD OF FINANCE'].map(weights6)
```

In [455...

```
def weights7(Average_Revenue):
    if Average_Revenue < 1000000:
        return 1.5
    elif Average_Revenue < 5000000:
        return 2
    elif Average_Revenue < 10000000:
        return 2.5
    elif Average_Revenue < 15000000:
        return 3.5
    elif Average_Revenue < 20000000:
        return 4
    elif Average_Revenue < 100000000:
        return 4.5
    else:
        return 5
df2['Average Revenue weighted'] = df2['Average Revenue(UGX)'].apply(weights7)
```

In [456...

```
def weights8(Average_debt):
    if Average_debt < 1000000:
        return 5
    elif Average_debt < 5000000:
        return 4
    elif Average_debt < 10000000:
        return 3
    elif Average_debt < 15000000:
        return 2
    elif Average_debt < 20000000:
        return 1.5
```

```

    elif Average_debt < 100000000:
        return 1
    else:
        return 0
df2['Average debt weighted'] = df2['Average debt(UGX)'].apply(weights8)

```

```

In [457... def weights9(Average_profitability):
    if Average_profitability <= 1000000:
        return 1
    elif Average_profitability <= 5000000:
        return 3
    elif Average_profitability <= 10000000:
        return 3.5
    elif Average_profitability <=15000000:
        return 4
    elif Average_profitability <= 20000000:
        return 4.5
    else:
        return 5
df2['Average profitability weighted'] = df2['Average Profitability(UGX)'].apply(weights9)

```

```

In [458... df2['Inventory Management weighted'] = [4, 4, 1, 1.5, 0, 0, 1.5, 0, 4, 4, 0, 0, 4, 1.5, 4]

```

```

In [459... def weights10 (team_experience):
    if team_experience < 1:
        return 1
    elif team_experience < 2:
        return 1.5
    elif team_experience < 3:
        return 2.5
    elif team_experience < 4:
        return 3
    elif team_experience < 5:
        return 3.5
    else:
        return 5
df2['Team experience weighted'] = df2['Team experience(years)'].apply(weights10)

```

```

In [461... df2['Human resource policies weighted'] = [0, 4, 0, 0, 0, 0, 1.5, 0, 4, 0, 0, 1.5, 4, 0, 4]

```

```

In [462... df2['Registration $ Licenses weighted'] = [4, 4, 4, 4, 4, 2, 4, 2, 4, 4, 2, 4, 4 ,4, 4]

```

```

In [463... df2['Risk management weighted'] = [5, 2, 1, 1, 1, 1, 1, 1, 1, 5, 1, 1, 2, 1, 5]

```

```

In [464... def weights11(growth_stage):
    if growth_stage < 2:
        return 1.5
    elif growth_stage < 4:
        return 2
    elif growth_stage < 6:
        return 4
    else:
        return 5
df2['Stage of growth weighted'] = df2['What stage of Growth is your business'].apply(weights11)

```

```

In [465... def weights12(Average_customers):
    if Average_customers < 5:
        return 1.5
    elif Average_customers < 10:
        return 2.5
    elif Average_customers < 15:
        return 3
    elif Average_customers < 20:
        return 3.5

```

```

    elif Average_customers < 30:
        return 4
    elif Average_customers < 50:
        return 4.5
    else:
        return 5
df2['Average customers weighted'] = df2['Average weekly customers'].apply(weights12)

```

```
In [466... df2['Supply chain weighted'] = [4, 3, 3, 2.5, 3, 4, 3, 2.5, 4, 2.5, 4, 4, 2.5, 2.5, 2]
```

```
In [467... df2['Nature of skills weighted'] = [2.5, 2, 2.5, 2, 2.5, 2.5, 2.5, 2, 2, 2, 4, 2.5, 4, 2.5, 2.5]
```

```
In [468... def weights13(Average_pay):
    if Average_pay < 150000:
        return 1
    elif Average_pay < 350000:
        return 1.5
    elif Average_pay < 650000:
        return 2.5
    elif Average_pay < 2000000:
        return 3
    elif Average_pay < 5000000:
        return 3.5
    elif Average_pay < 10000000:
        return 4
    else:
        return 5
df2['Average Monthly pay weighted'] = df2['Average Monthly pay (UGX)'].apply(weights13)

```

```
In [469... def weights14(Average_loss):
    if Average_loss < 150000:
        return 5
    elif Average_loss < 500000:
        return 4
    elif Average_loss < 2000000:
        return 3
    elif Average_loss < 5000000:
        return 1.5
    elif Average_loss < 10000000:
        return 1
    else:
        return 0
df2['Average Annual loss weighted'] = df2['Average Annual loss(UGX)'].apply(weights14)

```

```
In [470... df2.loc[:, ['OWNERSHIP', 'OWNERSHIP weights', 'NUMBER OF PEOPLE EMPLOYED', 'Number of people employed',
'Entrepreneurship years weighted', 'LEVEL OF FORMAL EDUCATION OF BUSINESS DIRECTOR', 'Education weighted',
'Product Differentiation/Unique selling point weighted', 'METHOD OF FINANCE', 'Method of finance weighted',
'Average Revenue(UGX)', 'Average Revenue weighted', 'Average debt(UGX)', 'Average debt weighted',
'Average Profitability(UGX)', 'Average profitability weighted', 'Inventory Management', 'Inventory Management weighted',
'Team experience(years)', 'Team experience weighted', 'Human resource policies', 'Human resource policies weighted',
'Registration & Licenses', 'Registration & Licenses weighted', 'Risk Management Scheme', 'Risk management weighted',
'What stage of Growth is your business', 'Stage of growth weighted', 'Average weekly customers', 'Average weekly customers weighted',
'What is your supply chain like', 'Supply chain weighted', 'Nature of skill required for employees', 'Nature of skill required for employees weighted',
'Average Monthly pay (UGX)', 'Average Monthly pay weighted', 'Average Annual loss(UGX)', 'Average Annual loss weighted']] = df2[['OWNERSHIP', 'OWNERSHIP weights', 'NUMBER OF PEOPLE EMPLOYED', 'Number of people employed',
'Entrepreneurship years weighted', 'LEVEL OF FORMAL EDUCATION OF BUSINESS DIRECTOR', 'Education weighted',
'Product Differentiation/Unique selling point weighted', 'METHOD OF FINANCE', 'Method of finance weighted',
'Average Revenue(UGX)', 'Average Revenue weighted', 'Average debt(UGX)', 'Average debt weighted',
'Average Profitability(UGX)', 'Average profitability weighted', 'Inventory Management', 'Inventory Management weighted',
'Team experience(years)', 'Team experience weighted', 'Human resource policies', 'Human resource policies weighted',
'Registration & Licenses', 'Registration & Licenses weighted', 'Risk Management Scheme', 'Risk management weighted',
'What stage of Growth is your business', 'Stage of growth weighted', 'Average weekly customers', 'Average weekly customers weighted',
'What is your supply chain like', 'Supply chain weighted', 'Nature of skill required for employees', 'Nature of skill required for employees weighted',
'Average Monthly pay (UGX)', 'Average Monthly pay weighted', 'Average Annual loss(UGX)', 'Average Annual loss weighted']]

```

Out[470]:

	OWNERSHIP	OWNERSHIP weights	NUMBER OF PEOPLE EMPLOYED	Number of people employed weighted	YEARS IN ENTREPRENEURSHIP	Entrepreneurship years weighted	LEVEL OF FORMAL EDUCATION OF BUSINESS DIRECTOR	Educational weight
0	Sole owner	2.5	3	2.5	3	3	Bachelors Degree	
1	Sole owner	2.5	6	2.5	6	4	Bachelors Degree	
2	Sole owner	2.5	1	2.5	3	3	Bachelors Degree	
3	Partnership (General or limited)	3.5	1	2.5	2	2	Masters Degree or Higher	
4	Family owned	3.0	1	2.5	0	1	O/A Level Certificate	
5	Sole owner	2.5	3	2.5	2	2	Bachelors Degree	
6	Partnership (General or limited)	3.5	41	5.0	6	4	Masters Degree or Higher	
7	Family owned	3.0	10	3.0	5	3	Bachelors Degree	
8	Partnership (General or limited)	3.5	18	3.0	6	4	Bachelors Degree	
9	Sole owner	2.5	4	2.5	6	4	Bachelors Degree	

	OWNERSHIP	OWNERSHIP weights	NUMBER OF PEOPLE EMPLOYED	Number of people employed weighted	YEARS IN ENTREPRENEURSHIP	Entrepreneurship years weighted	LEVEL OF FORMAL EDUCATION OF BUSINESS DIRECTOR	Educational weight
10	Partnership (General or limited)	3.5	2	2.5	2	2	Bachelors Degree	
11	Partnership (General or limited)	3.5	15	3.0	3	3	Bachelors Degree	
12	Family owned	3.0	29	3.5	3	3	Bachelors Degree	
13	Sole owner	2.5	1	2.5	3	3	Bachelors Degree	
14	Partnership (General or limited)	3.5	30	4.0	5	3	Bachelors Degree	

CHECKING TO SEE IF WE HAVE THE WEIGHTED COLUMNS

```
In [471... df2['Total Credit Score'] = [56, 56, 46.5, 48.5, 50, 44.5, 55.5, 46.5, 63.5, 54.5, 47, 58.5, 56.5,
```

CREATING A CREDIT SCORE COLUMN

```
In [472... df2.loc[:,['Total Credit Score','OWNERSHIP', 'OWNERSHIP weights','NUMBER OF PEOPLE EMPLOYED','Numb
'Entrepreneurship years weighted','LEVEL OF FORMAL EDUCATION OF BUSINESS DIRECTOR','Education weig
'Product Differentiation/Unique selling point weighted', 'METHOD OF FINANCE','Method of finance we
'Average Revenue(UGX)', 'Average Revenue weighted', 'Average debt(UGX)','Average debt weighted',
'Average Profitability(UGX)','Average profitability weighted','Inventory Management','Inventory Ma
'Team experience(years)', 'Team experience weighted','Human resource policies','Human resource pol
'Registration & Licenses','Registration $ Licenses weighted','Risk Management Scheme','Risk manage
'What stage of Growth is your business', 'Stage of growth weighted','Average weekly customers', 'A
'what is your supply chain like','Supply chain weighted', 'Nature of skill required for employees'
'Average Monthly pay (UGX)','Average Monthly pay weighted','Average Annual loss(UGX)', 'Average Ar
```

Out[472]:

	Total Credit Score	OWNERSHIP	OWNERSHIP weights	NUMBER OF PEOPLE EMPLOYED	Number of people employed weighted	YEARS IN ENTREPRENEURSHIP	Entrepreneurship years weighted	LEVEL OF FORMAL EDUCATION OF BUSINESS DIRECTOR
0	56.0	Sole owner	2.5	3	2.5	3	3	Bachelors Degree
1	56.0	Sole owner	2.5	6	2.5	6	4	Bachelors Degree
2	46.5	Sole owner	2.5	1	2.5	3	3	Bachelors Degree
3	48.5	Partnership (General or limited)	3.5	1	2.5	2	2	Masters Degree or Higher
4	50.0	Family owned	3.0	1	2.5	0	1	O/A Level Certificate
5	44.5	Sole owner	2.5	3	2.5	2	2	Bachelors Degree
6	55.5	Partnership (General or limited)	3.5	41	5.0	6	4	Masters Degree or Higher
7	46.5	Family owned	3.0	10	3.0	5	3	Bachelors Degree
8	63.5	Partnership (General or limited)	3.5	18	3.0	6	4	Bachelors Degree
9	54.5	Sole owner	2.5	4	2.5	6	4	Bachelors Degree

	Total Credit Score	OWNERSHIP	OWNERSHIP weights	NUMBER OF PEOPLE EMPLOYED	Number of people employed weighted	YEARS IN ENTREPRENEURSHIP	Entrepreneurship years weighted	LEVEL OF FORMAL EDUCATION OF BUSINESS DIRECTOR
10	47.0	Partnership (General or limited)	3.5	2	2.5	2	2	Bachelors Degree
11	58.5	Partnership (General or limited)	3.5	15	3.0	3	3	Bachelors Degree
12	56.5	Family owned	3.0	29	3.5	3	3	Bachelors Degree
13	46.5	Sole owner	2.5	1	2.5	3	3	Bachelors Degree
14	64.0	Partnership (General or limited)	3.5	30	4.0	5	3	Bachelors Degree

USING A RANDOM FOREST CLASSIFIER TO PREDICT LOAN DEFAULTS

```
In [473... from sklearn.model_selection import train_test_split, cross_val_score
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import classification_report, confusion_matrix
from sklearn.model_selection import StratifiedKFold
```

```
In [474... df2['Potential Default'] = df2['Total Credit Score'].apply(lambda x: 1 if x>=50 else 0)
df2.loc[:,['Total Credit Score', 'Potential Default']].head(15)
```

Out[474]:

	Total Credit Score	Potential Default
0	56.0	1
1	56.0	1
2	46.5	0
3	48.5	0
4	50.0	1
5	44.5	0
6	55.5	1
7	46.5	0
8	63.5	1
9	54.5	1
10	47.0	0
11	58.5	1
12	56.5	1
13	46.5	0
14	64.0	1

ANYONE WITH A CREDIT SCORE OF BELOW 50 ASSIGNED A 0 AND ABOVE OR EQUAL TO 50 A 1

TRAINING THE MODEL

In [418...

```
features = df2[['Total Credit Score','OWNERSHIP weights','Number of people employed weighted',
'Entrepreneurship years weighted','Education weighted','Product Differentiation/Unique selling poi
'Method of finance weighted','Average Revenue weighted','Average debt weighted',
'Average profitability weighted','Inventory Management weighted','Team experience weighted','Humar
'Registration $ Licenses weighted','Risk management weighted','Stage of growth weighted','Average
'Supply chain weighted','Nature of skills weighted','Average Monthly pay weighted','Average Annual
target = df2['Potential Default']
x = features
y= target

x_train, x_test, y_train, y_test = train_test_split(x,y, test_size = 0.2,random_state = 42)
rf_classifier = RandomForestClassifier(n_estimators = 10, max_depth = 5,min_samples_split=4
,min_samples_leaf = 2)
cv = StratifiedKFold(n_splits = 5 , shuffle = True, random_state = 42)
cross_val_scores = cross_val_score(rf_classifier, x, y, cv=cv, scoring = 'accuracy')
print(f'Cross validation scores: {cross_val_scores}')
print(f'mean cv accuracy: {np.mean(cross_val_scores)}')
```

```
Cross validation scores: [1.          1.          0.66666667 1.          1.          ]
mean cv accuracy: 0.9333333333333333
```

In [423...

```
RF_classifier = RandomForestClassifier()
RF_classifier.fit(x,y)
predicted_defaults = RF_classifier.predict(x)
number_of_defaults = sum(predicted_defaults)
print(f'model predicted {number_of_defaults} defaults.')
```

model predicted 9 defaults.

AS WE CAN SEE ALTHOUGH THE DATASET IS SMALL AND THE MODEL STILL NEEDS FINE TUNING WITH THE FEATURES PRESENTED, THE MODEL PREDICTED 9 DEFAULTS FROM OUR DATASET

In []: