Monty K Vasita

Setting up libraries, logging, and pandas display — no data insights yet, just environment setup

In [57]:

```
# Import Data Manipulation Libraries
import numpy as np
import pandas as pd

# Import Data Visualization Libraries
import matplotlib.pyplot as plt
import seaborn as sns

# Import Filter Warning Libraries
import warnings
warnings.filterwarnings('ignore')

# Import Logging  Files
import logging

logging.basicConfig(
    level=logging.INFO,
    filemode='w',
    filename='app.log',
    format='%(asctime)s - %(levelname)s - %(message)s')
```

In [58]:

```
pd.set_option("display.max_columns", None)
pd.set_option("display.max_rows", 100)
```

Loading Data Set

In [59]:

```
# DataSet
url="https://raw.githubusercontent.com/MontyVasita18/CodeB_Internship/refs/heads/main/dataset_phishing.csv"
df=pd.read_csv(url)
df.sample(frac=1) # To make the code execution faster
```

Out[59]:

	url	length_url	length_hostname	ip	nb_dots	nb_hyphens	nb_at	nb_qm	nb_and	nb_
4505	https://www.iconfinder.com/search/?q=parking	44	18	0	2	0	0	1	0	
7500	http://www.shadetreetechnology.com/V4/validati...	81	27	1	2	0	0	0	0	
8559	https://www.komikgue.com/	25	16	0	2	0	0	0	0	
8257	https://www.inopressa.ru/	25	16	0	2	0	0	0	0	
54	http://www.astroresidential.co.uk/application/...	51	26	0	3	0	0	0	0	
...	...	...	...	...	...	...	...	...	...	
7981	https://623112j4j3.codesandbox.io/index.html	44	25	0	3	0	0	0	0	
474	https://www.wunderground.com/us/ca/fontana	42	20	0	2	0	0	0	0	
5848	http://www.businessdictionary.com/definition/r...	74	26	0	3	3	0	0	0	
6358	https://www.therightnote.com.au/	32	23	0	3	0	0	0	0	
11013	https://www.onlineocr.net	25	17	0	2	0	0	0	0	

11430 rows × 89 columns

📊 Exploratory Data Analysis (EDA)

Checking shape, data types, and missing values. The dataset looks clean, with no major null-value issues. That means no need for imputation or heavy cleaning

In [60]:

```
df.shape
```

Out[60]:

```
(11430, 89)
```

In [61]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 11430 entries, 0 to 11429
Data columns (total 89 columns):
 #   Column                       Non-Null Count  Dtype  
---  ------                       --------------  -----  
 0   url                          11430 non-null  object 
 1   length_url                   11430 non-null  int64  
 2   length_hostname              11430 non-null  int64  
 3   ip                           11430 non-null  int64  
 4   nb_dots                      11430 non-null  int64  
 5   nb_hyphens                   11430 non-null  int64  
 6   nb_at                        11430 non-null  int64  
 7   nb_qm                        11430 non-null  int64  
 8   nb_and                       11430 non-null  int64  
 9   nb_or                        11430 non-null  int64  
 10  nb_eq                        11430 non-null  int64  
 11  nb_underscore                11430 non-null  int64  
 12  nb_tilde                     11430 non-null  int64  
 13  nb_percent                   11430 non-null  int64  
 14  nb_slash                     11430 non-null  int64  
 15  nb_star                      11430 non-null  int64  
 16  nb_colon                     11430 non-null  int64  
 17  nb_comma                     11430 non-null  int64  
 18  nb_semicolumn                11430 non-null  int64  
 19  nb_dollar                    11430 non-null  int64  
 20  nb_space                     11430 non-null  int64  
 21  nb_www                       11430 non-null  int64  
 22  nb_com                       11430 non-null  int64  
 23  nb_dslash                    11430 non-null  int64  
 24  http_in_path                 11430 non-null  int64  
 25  https_token                  11430 non-null  int64  
 26  ratio_digits_url             11430 non-null  float64
 27  ratio_digits_host            11430 non-null  float64
 28  punycode                     11430 non-null  int64  
 29  port                         11430 non-null  int64  
 30  tld_in_path                  11430 non-null  int64  
 31  tld_in_subdomain             11430 non-null  int64  
 32  abnormal_subdomain           11430 non-null  int64  
 33  nb_subdomains                11430 non-null  int64  
 34  prefix_suffix                11430 non-null  int64  
 35  random_domain                11430 non-null  int64  
 36  shortening_service           11430 non-null  int64  
 37  path_extension               11430 non-null  int64  
 38  nb_redirection               11430 non-null  int64  
 39  nb_external_redirection      11430 non-null  int64  
 40  length_words_raw             11430 non-null  int64  
 41  char_repeat                  11430 non-null  int64  
 42  shortest_words_raw           11430 non-null  int64  
 43  shortest_word_host           11430 non-null  int64  
 44  shortest_word_path           11430 non-null  int64  
 45  longest_words_raw            11430 non-null  int64  
 46  longest_word_host            11430 non-null  int64  
 47  longest_word_path            11430 non-null  int64  
 48  avg_words_raw                11430 non-null  float64
 49  avg_word_host                11430 non-null  float64
 50  avg_word_path                11430 non-null  float64
 51  phish_hints                  11430 non-null  int64  
 52  domain_in_brand              11430 non-null  int64  
 53  brand_in_subdomain           11430 non-null  int64  
 54  brand_in_path                11430 non-null  int64  
 55  suspecious_tld               11430 non-null  int64  
 56  statistical_report           11430 non-null  int64  
 57  nb_hyperlinks                11430 non-null  int64  
 58  ratio_intHyperlinks          11430 non-null  float64
 59  ratio_extHyperlinks          11430 non-null  float64
 60  ratio_nullHyperlinks         11430 non-null  int64  
 61  nb_extCSS                    11430 non-null  int64  
 62  ratio_intRedirection         11430 non-null  int64  
 63  ratio_extRedirection        11430 non-null  float64
 64  ratio_intErrors             11430 non-null  int64
 65  ratio_extErrors             11430 non-null  float64
 66  login_form                  11430 non-null  int64
 67  external_favicon            11430 non-null  int64
 68  links_in_tags               11430 non-null  float64
 69  submit_email                11430 non-null  int64
 70  ratio_intMedia              11430 non-null  float64
 71  ratio_extMedia              11430 non-null  float64
 72  sfh                         11430 non-null  int64
 73  iframe                      11430 non-null  int64
 74  popup_window                11430 non-null  int64
 75  safe_anchor                 11430 non-null  float64
 76  onmouseover                 11430 non-null  int64
 77  right_clic                  11430 non-null  int64
 78  empty_title                 11430 non-null  int64
 79  domain_in_title             11430 non-null  int64
 80  domain_with_copyright       11430 non-null  int64
 81  whois_registered_domain     11430 non-null  int64
 82  domain_registration_length  11430 non-null  int64
 83  domain_age                  11430 non-null  int64
 84  web_traffic                 11430 non-null  int64
 85  dns_record                  11430 non-null  int64
 86  google_index                11430 non-null  int64
 87  page_rank                   11430 non-null  int64
 88  status                      11430 non-null  object
dtypes: float64(13), int64(74), object(2)
memory usage: 7.8+ MB
```

In [62]:

```
# Checking Null Value in DataSet
df.isnull().sum()/len(df)*100
```

Out[62]:

```
url                           0.0
length_url                    0.0
length_hostname               0.0
ip                            0.0
nb_dots                       0.0
nb_hyphens                    0.0
nb_at                         0.0
nb_qm                         0.0
nb_and                        0.0
nb_or                         0.0
nb_eq                         0.0
nb_underscore                 0.0
nb_tilde                      0.0
nb_percent                    0.0
nb_slash                      0.0
nb_star                       0.0
nb_colon                      0.0
nb_comma                      0.0
nb_semicolumn                 0.0
nb_dollar                     0.0
nb_space                      0.0
nb_www                        0.0
nb_com                        0.0
nb_dslash                     0.0
http_in_path                  0.0
https_token                   0.0
ratio_digits_url              0.0
ratio_digits_host             0.0
punycode                      0.0
port                          0.0
tld_in_path                   0.0
tld_in_subdomain              0.0
abnormal_subdomain            0.0
nb_subdomains                 0.0
prefix_suffix                 0.0
random_domain                 0.0
shortening_service            0.0
path_extension                0.0
nb_redirection                0.0
nb_external_redirection       0.0
length_words_raw              0.0
char_repeat                   0.0
shortest_words_raw            0.0
shortest_word_host            0.0
shortest_word_path            0.0
longest_words_raw             0.0
longest_word_host             0.0
longest_word_path             0.0
avg_words_raw                 0.0
avg_word_host                 0.0
avg_word_path                 0.0
phish_hints                   0.0
phish_hints                   0.0
domain_in_brand               0.0
brand_in_subdomain            0.0
brand_in_path                 0.0
suspecious_tld                0.0
statistical_report            0.0
nb_hyperlinks                 0.0
ratio_intHyperlinks           0.0
ratio_extHyperlinks           0.0
ratio_nullHyperlinks          0.0
nb_extCSS                     0.0
ratio_intRedirection          0.0
ratio_extRedirection          0.0
ratio_intErrors               0.0
ratio_extErrors               0.0
login_form                    0.0
external_favicon              0.0
links_in_tags                 0.0
submit_email                  0.0
ratio_intMedia                0.0
ratio_extMedia                0.0
sfh                           0.0
iframe                        0.0
popup_window                  0.0
safe_anchor                   0.0
onmouseover                   0.0
right_clic                    0.0
empty_title                   0.0
domain_in_title               0.0
domain_with_copyright         0.0
whois_registered_domain       0.0
domain_registration_length    0.0
domain_age                    0.0
web_traffic                   0.0
dns_record                    0.0
google_index                  0.0
page_rank                     0.0
status                        0.0
dtype: float64
```

Summary statistics show how features are spread. You can spot skewed distributions and outliers (e.g., long URLs, high digit ratio).

In [63]:

```
df.describe()
```

Out[63]:

	length_url	length_hostname	ip	nb_dots	nb_hyphens	nb_at	nb_qm	nb_and	nb_or	
count	11430.000000	11430.000000	11430.000000	11430.000000	11430.000000	11430.000000	11430.000000	11430.000000	11430.0	114
mean	61.126684	21.090289	0.150569	2.480752	0.997550	0.022222	0.141207	0.162292	0.0	
std	55.297318	10.777171	0.357644	1.369686	2.087087	0.155500	0.364456	0.821337	0.0	
min	12.000000	4.000000	0.000000	1.000000	0.000000	0.000000	0.000000	0.000000	0.0	
25%	33.000000	15.000000	0.000000	2.000000	0.000000	0.000000	0.000000	0.000000	0.0	
50%	47.000000	19.000000	0.000000	2.000000	0.000000	0.000000	0.000000	0.000000	0.0	
75%	71.000000	24.000000	0.000000	3.000000	1.000000	0.000000	0.000000	0.000000	0.0	
max	1641.000000	214.000000	1.000000	24.000000	43.000000	4.000000	3.000000	19.000000	0.0	

Separating numerical and categorical columns. Then, for each numeric feature, you analyze spread, skewness, and outliers — very helpful for choosing scaling techniques or detecting which features might need transformation.

In [64]:

```
# Spliting data into Numerical Data and Catagorical Data
numerical_data=df.select_dtypes(exclude='object')
numerical_data

categorical_data=df.select_dtypes(include='object')
```

In [65]:

```
from collections import OrderedDict
stats=[]

for col in df.columns:
```

```
for col in df.columns:
    if df[col].dtype !='object':
        numerical_stats=OrderedDict({
            'Feature': col,
            'Minimum': df[col].min(),
            'Maximum': df[col].max(),
            'Mean': df[col].mean(),
            'Mode': df[col].mode()[0] if not df[col].mode().empty else None,
            '25%': df[col].quantile(0.25),
            '75%': df[col].quantile(0.75),
            'IQR': df[col].quantile(0.75) - df[col].quantile(0.25),
            'Standard Deviation': df[col].std(),
            'Skewness': df[col].skew(),
            'Kurtosis': df[col].kurt()


        })
        stats.append(numerical_stats)
report=pd.DataFrame(stats)
report
```

Out[65]:

	Feature	Minimum	Maximum	Mean	Mode	25%	75%	IQR	Standard Deviation
0	length_url	12.0	1.641000e+03	61.126684	26.0	33.000000	71.000000	38.000000	5.529732e+01
1	length_hostname	4.0	2.140000e+02	21.090289	16.0	15.000000	24.000000	9.000000	1.077717e+01
2	ip	0.0	1.000000e+00	0.150569	0.0	0.000000	0.000000	0.000000	3.576436e-01
3	nb_dots	1.0	2.400000e+01	2.480752	2.0	2.000000	3.000000	1.000000	1.369686e+00
4	nb_hyphens	0.0	4.300000e+01	0.997550	0.0	0.000000	1.000000	1.000000	2.087087e+00
5	nb_at	0.0	4.000000e+00	0.022222	0.0	0.000000	0.000000	0.000000	1.554999e-01
6	nb_qm	0.0	3.000000e+00	0.141207	0.0	0.000000	0.000000	0.000000	3.644558e-01
7	nb_and	0.0	1.900000e+01	0.162292	0.0	0.000000	0.000000	0.000000	8.213374e-01
8	nb_or	0.0	0.000000e+00	0.000000	0.0	0.000000	0.000000	0.000000	0.000000e+00
9	nb_eq	0.0	1.900000e+01	0.293176	0.0	0.000000	0.000000	0.000000	9.983172e-01
10	nb_underscore	0.0	1.800000e+01	0.322660	0.0	0.000000	0.000000	0.000000	1.093336e+00
11	nb_tilde	0.0	1.000000e+00	0.006649	0.0	0.000000	0.000000	0.000000	8.127444e-02
12	nb_percent	0.0	9.600000e+01	0.123097	0.0	0.000000	0.000000	0.000000	1.466450e+00
13	nb_slash	2.0	3.300000e+01	4.289589	3.0	3.000000	5.000000	2.000000	1.882251e+00
14	nb_star	0.0	1.000000e+00	0.000700	0.0	0.000000	0.000000	0.000000	2.644776e-02
15	nb_colon	1.0	7.000000e+00	1.027909	1.0	1.000000	1.000000	0.000000	2.403255e-01
16	nb_comma	0.0	4.000000e+00	0.004024	0.0	0.000000	0.000000	0.000000	1.032395e-01
17	nb_semicolumn	0.0	2.000000e+01	0.062292	0.0	0.000000	0.000000	0.000000	5.981896e-01
18	nb_dollar	0.0	6.000000e+00	0.001925	0.0	0.000000	0.000000	0.000000	7.711078e-02
19	nb_space	0.0	1.800000e+01	0.034821	0.0	0.000000	0.000000	0.000000	3.755757e-01
20	nb_www	0.0	2.000000e+00	0.448469	0.0	0.000000	1.000000	1.000000	5.019124e-01
21	nb_com	0.0	6.000000e+00	0.127997	0.0	0.000000	0.000000	0.000000	3.790079e-01
22	nb_dslash	0.0	1.000000e+00	0.006562	0.0	0.000000	0.000000	0.000000	8.074153e-02
23	http_in_path	0.0	4.000000e+00	0.016710	0.0	0.000000	0.000000	0.000000	1.693581e-01
24	https_token	0.0	1.000000e+00	0.610936	1.0	0.000000	1.000000	1.000000	4.875592e-01
25	ratio_digits_url	0.0	7.238806e-01	0.053137	0.0	0.000000	0.079365	0.079365	8.936273e-02
26	ratio_digits_host	0.0	8.000000e-01	0.025024	0.0	0.000000	0.000000	0.000000	9.342200e-02
27	punycode	0.0	1.000000e+00	0.000350	0.0	0.000000	0.000000	0.000000	1.870466e-02
28	port	0.0	1.000000e+00	0.002362	0.0	0.000000	0.000000	0.000000	4.854720e-02
29	tld_in_path	0.0	1.000000e+00	0.065617	0.0	0.000000	0.000000	0.000000	2.476219e-01
30	tld_in_subdomain	0.0	1.000000e+00	0.050131	0.0	0.000000	0.000000	0.000000	2.182252e-01
31	abnormal_subdomain	0.0	1.000000e+00	0.021610	0.0	0.000000	0.000000	0.000000	1.454121e-01
32	nb_subdomains	1.0	3.000000e+00	2.231671	2.0	2.000000	3.000000	1.000000	6.370688e-01
33	prefix_suffix	0.0	1.000000e+00	0.202450	0.0	0.000000	0.000000	0.000000	4.018432e-01

34	random_domain	0.0	1.000000e+00	0.083290	0.0	0.000000	0.000000	0.000000	2.763315e-01	3
35	shortening_service	0.0	1.000000e+00	0.123447	0.0	0.000000	0.000000	0.000000	3.289641e-01	2
36	path_extension	0.0	1.000000e+00	0.000175	0.0	0.000000	0.000000	0.000000	1.322735e-02	75
37	nb_redirection	0.0	6.000000e+00	0.498250	0.0	0.000000	1.000000	1.000000	6.919070e-01	1
38	nb_external_redirection	0.0	1.000000e+00	0.003150	0.0	0.000000	0.000000	0.000000	5.603535e-02	17
39	length_words_raw	1.0	1.060000e+02	6.232808	2.0	2.000000	8.000000	6.000000	5.572355e+00	5
40	char_repeat	0.0	1.460000e+02	2.927472	3.0	1.000000	4.000000	3.000000	4.768936e+00	15
41	shortest_words_raw	1.0	3.100000e+01	3.127297	3.0	2.000000	3.000000	1.000000	2.211571e+00	3
42	shortest_word_host	1.0	3.900000e+01	5.019773	3.0	3.000000	6.000000	3.000000	3.941580e+00	2
43	shortest_word_path	0.0	4.000000e+01	2.398950	0.0	0.000000	3.000000	3.000000	2.997809e+00	4
44	longest_words_raw	2.0	8.290000e+02	15.393876	9.0	9.000000	16.000000	7.000000	2.208364e+01	13
45	longest_word_host	1.0	6.200000e+01	10.467979	9.0	7.000000	13.000000	6.000000	4.932015e+00	1
46	longest_word_path	0.0	8.290000e+02	10.561505	0.0	0.000000	11.000000	11.000000	2.307788e+01	12
47	avg_words_raw	2.0	1.282500e+02	7.258882	6.0	5.250000	8.000000	2.750000	4.145827e+00	9
48	avg_word_host	1.0	3.900000e+01	7.678075	5.0	5.250000	9.000000	3.750000	3.578435e+00	1
49	avg_word_path	0.0	2.500000e+02	5.092425	0.0	0.000000	6.714286	6.714286	7.147050e+00	13
50	phish_hints	0.0	1.000000e+01	0.327734	0.0	0.000000	0.000000	0.000000	8.426004e-01	3
51	domain_in_brand	0.0	1.000000e+00	0.104199	0.0	0.000000	0.000000	0.000000	3.055325e-01	2
52	brand_in_subdomain	0.0	1.000000e+00	0.004112	0.0	0.000000	0.000000	0.000000	6.399559e-02	15
53	brand_in_path	0.0	1.000000e+00	0.004899	0.0	0.000000	0.000000	0.000000	6.982700e-02	14
54	suspecious_tld	0.0	1.000000e+00	0.017935	0.0	0.000000	0.000000	0.000000	1.327220e-01	7
55	statistical_report	0.0	2.000000e+00	0.059755	0.0	0.000000	0.000000	0.000000	3.312662e-01	5
56	nb_hyperlinks	0.0	4.659000e+03	87.189764	0.0	9.000000	101.000000	92.000000	1.667583e+02	7
57	ratio_intHyperlinks	0.0	1.000000e+00	0.602457	0.0	0.224991	0.944767	0.719776	3.764745e-01	-0
58	ratio_extHyperlinks	0.0	1.000000e+00	0.276720	0.0	0.000000	0.474840	0.474840	3.199583e-01	1
59	ratio_nullHyperlinks	0.0	0.000000e+00	0.000000	0.0	0.000000	0.000000	0.000000	0.000000e+00	0
60	nb_extCSS	0.0	1.240000e+02	0.784864	0.0	0.000000	1.000000	1.000000	2.758802e+00	23
61	ratio_intRedirection	0.0	0.000000e+00	0.000000	0.0	0.000000	0.000000	0.000000	0.000000e+00	0
62	ratio_extRedirection	0.0	2.000000e+00	0.158926	0.0	0.000000	0.230769	0.230769	2.664370e-01	2
63	ratio_intErrors	0.0	0.000000e+00	0.000000	0.0	0.000000	0.000000	0.000000	0.000000e+00	0
64	ratio_extErrors	0.0	1.000000e+00	0.062469	0.0	0.000000	0.034483	0.034483	1.562087e-01	3
65	login_form	0.0	1.000000e+00	0.063605	0.0	0.000000	0.000000	0.000000	2.440578e-01	3
66	external_favicon	0.0	1.000000e+00	0.442170	0.0	0.000000	1.000000	1.000000	4.966661e-01	0
67	links_in_tags	0.0	1.000000e+02	51.978211	0.0	0.000000	98.061004	98.061004	4.152314e+01	-0
68	submit_email	0.0	0.000000e+00	0.000000	0.0	0.000000	0.000000	0.000000	0.000000e+00	0
69	ratio_intMedia	0.0	1.000000e+02	42.870444	0.0	0.000000	100.000000	100.000000	4.624990e+01	0
70	ratio_extMedia	0.0	1.000000e+02	23.236293	0.0	0.000000	33.333333	33.333333	3.838658e+01	1
71	sfh	0.0	0.000000e+00	0.000000	0.0	0.000000	0.000000	0.000000	0.000000e+00	0
72	iframe	0.0	1.000000e+00	0.001312	0.0	0.000000	0.000000	0.000000	3.620398e-02	27
73	popup_window	0.0	1.000000e+00	0.006037	0.0	0.000000	0.000000	0.000000	7.746501e-02	12
74	safe_anchor	0.0	1.000000e+02	37.063922	0.0	0.000000	75.000000	75.000000	3.907339e+01	0
75	onmouseover	0.0	1.000000e+00	0.001137	0.0	0.000000	0.000000	0.000000	3.370703e-02	29
76	right_clic	0.0	1.000000e+00	0.001400	0.0	0.000000	0.000000	0.000000	3.738968e-02	26
77	empty_title	0.0	1.000000e+00	0.124759	0.0	0.000000	0.000000	0.000000	3.304604e-01	2
78	domain_in_title	0.0	1.000000e+00	0.775853	1.0	1.000000	1.000000	0.000000	4.170376e-01	-1
79	domain_with_copyright	0.0	1.000000e+00	0.439545	0.0	0.000000	1.000000	1.000000	4.963535e-01	0
80	whois_registered_domain	0.0	1.000000e+00	0.072878	0.0	0.000000	0.000000	0.000000	2.599482e-01	3
81	domain_registration_length	-1.0	2.982900e+04	492.532196	0.0	84.000000	449.000000	365.000000	8.147694e+02	9

82	domain_age	-12.0	1.287400e+04	4062.543745	-1.0	972.250000	7026.750000	6054.500000	3.107785e+03
83	web_traffic	0.0	1.076799e+07	856756.643307	0.0	0.000000	373845.500000	373845.500000	1.995606e+06
84	dns_record	0.0	1.000000e+00	0.020122	0.0	0.000000	0.000000	0.000000	1.404254e-01
85	google_index	0.0	1.000000e+00	0.533946	1.0	0.000000	1.000000	1.000000	4.988682e-01
86	page_rank	0.0	1.000000e+01	3.185739	0.0	1.000000	5.000000	4.000000	2.536955e+00

Several features showed significant skewness, suggesting non-normal distributions.

Wide ranges and high standard deviations in some columns (e.g., web_traffic, length_url) indicate the presence of outliers.

Features with high kurtosis are likely to have heavy tails or sharp peaks.

Checking frequency counts for categorical columns — this helps you see whether categories are balanced or dominated by one class (like the target label status).

In [66]:

```
# Frequency Distribution
for col in df.columns:
    if df[col].dtype=='object':
        print(f"Frequency Distribution Of {col}\n")
        print(df[col].value_counts)
```

```
Frequency Distribution Of url

<bound method IndexOpsMixin.value_counts of 0                    http://www.crestonwood.com/router.php
1        http://shadetreetechnology.com/V4/validation/a...
2        https://support-appleld.com.secureupdate.duila...
3                                       http://rgipt.ac.in
4        http://www.iracing.com/tracks/gateway-motorspo...
                               ...                        
11425        http://www.fontspace.com/category/blackletter
11426    http://www.budgetbots.com/server.php/Server%20...
11427    https://www.facebook.com/Interactive-Televisio...
11428                   http://www.mypublicdomainpictures.com/
11429    http://174.139.46.123/ap/signin?openid.pape.ma...
Name: url, Length: 11430, dtype: object>
Frequency Distribution Of status

<bound method IndexOpsMixin.value_counts of 0        legitimate
1          phishing
2          phishing
3        legitimate
4        legitimate
            ...    
11425    legitimate
11426      phishing
11427    legitimate
11428    legitimate
11429      phishing
Name: status, Length: 11430, dtype: object>
```

In [67]:

```
df['status'].value_counts()
```

Out[67]:

```
status
legitimate    5715
phishing      5715
Name: count, dtype: int64
```

The target column status is well-balanced, which is ideal for binary classification models and ensures fair learning across both classes.

In [68]:

```
df['status'].mode()
```

Out[68]:

```
0    legitimate
1      phishing
Name: status, dtype: object
```

Label encoding turns 'legitimate' and 'phishing' into 0 and 1 — readying the target for machine learning models.

In [69]:

```
# Encoding Target column
df['status']=df['status'].replace({'legitimate':0,'phishing':1})
df['status']
```

Out[69]:

```
0        0
1        1
2        1
3        0
4        0
        ..
11425    0
11426    1
11427    0
11428    0
11429    1
Name: status, Length: 11430, dtype: int64
```

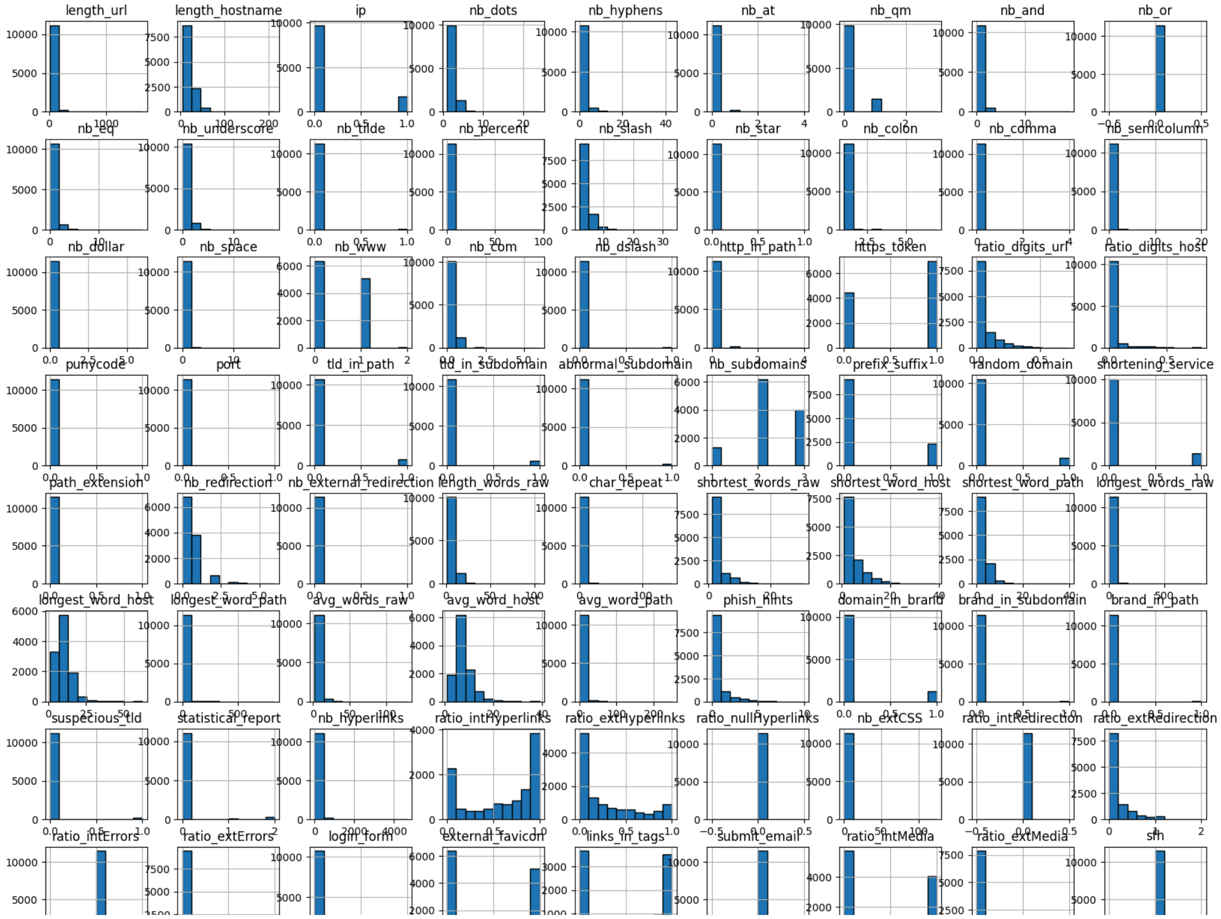
The target variable status was originally categorical, labeled as “phishing” and “legitimate.” It was converted into a binary format (1 and 0) for model compatibility.

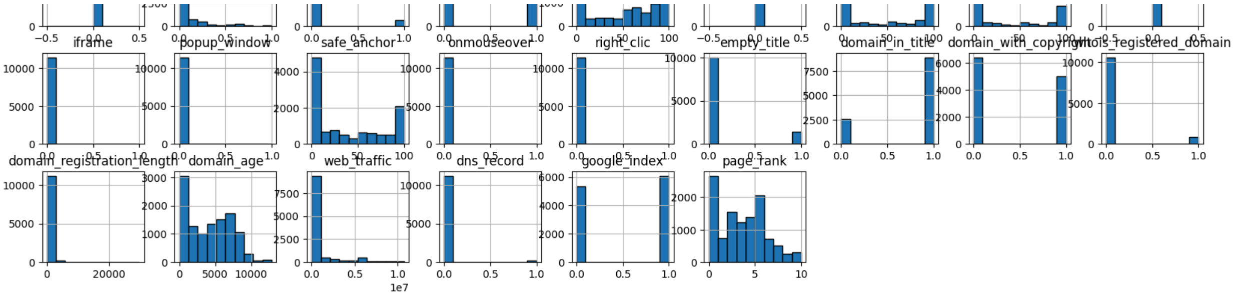
Histogram

Histograms Reveal skewed features and possible outliers. Some features like web_traffic or length_url may need scaling or normalization.

In [70]:

```
# Plotting Histogram
numerical_data.hist(figsize=(20,20),bins=10,edgecolor='black')
plt.title('Histogram example',y=1.02)
plt.xlabel('Value')
plt.ylabel('Frequency')
plt.show()
```



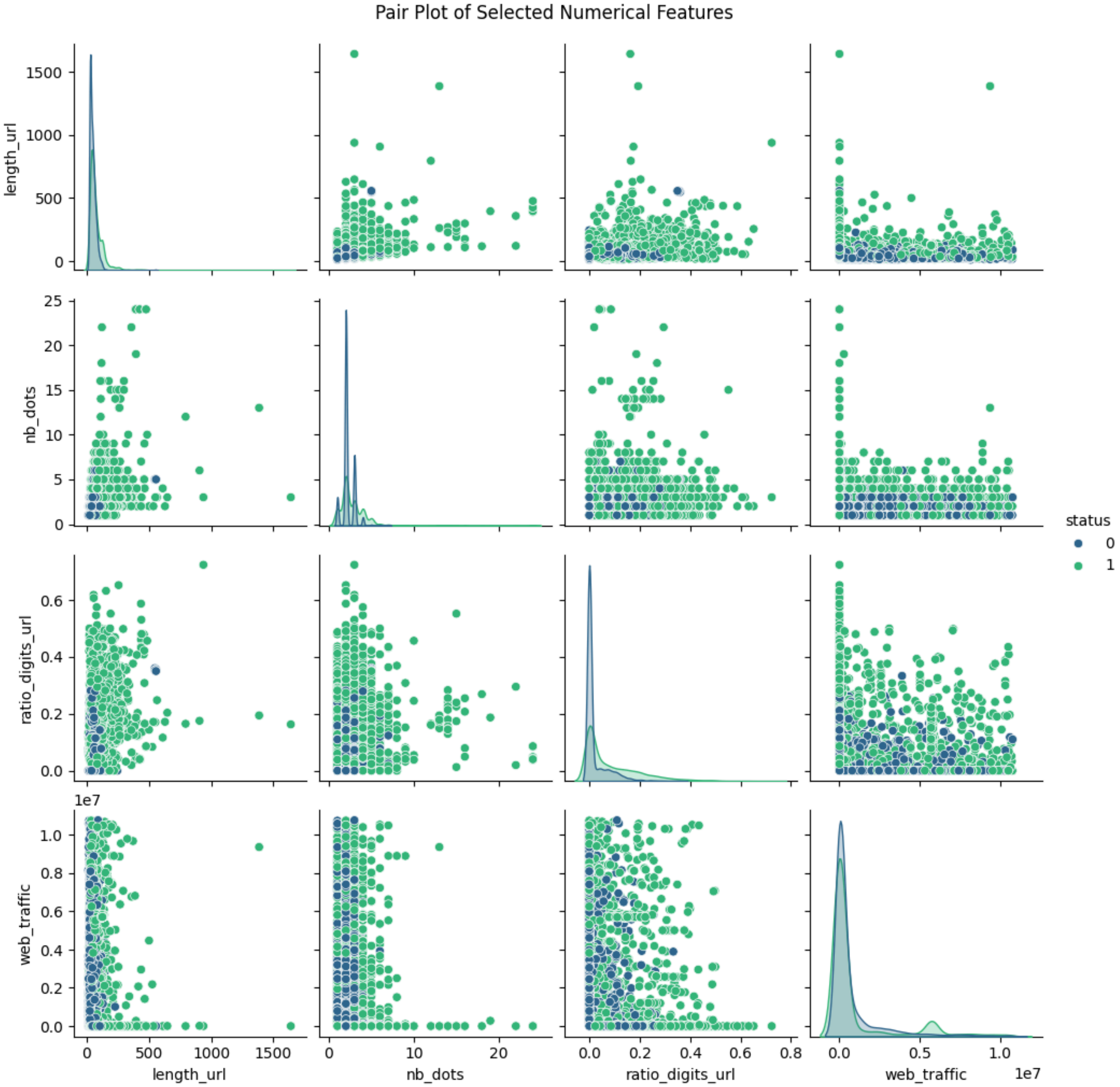


Many features are right-skewed, indicating potential preprocessing needs (e.g., log transformation). Distribution plots also highlighted high concentration of values in specific ranges for features like ratio_digits_url.

Pair Plot

In [71]:

```python
selected_features = ['length_url', 'nb_dots', 'ratio_digits_url', 'web_traffic', 'status']
# Plot pair plot
sns.pairplot(df[selected_features], hue='status', palette='viridis')
# Optional: Add title
plt.suptitle("Pair Plot of Selected Numerical Features", y=1.02)
plt.show()
```



The pairplot shows some visual separation between phishing and legitimate classes in selected features — especially in ratio_digits_url and web_traffic. That means these features might be strong indicators for classification.

✅ Insights & Recommendations

Key Findings:

Several numerical features display non-normal distributions and contain outliers, which could affect model performance if not addressed.

Features like ratio_digits_url and web_traffic show clear separation between classes and can act as strong indicators for phishing detection.

The target column status is well-balanced, which is ideal for binary classification models and ensures fair learning across both classes.

Recommended Actions:

Normalize or transform skewed numerical features (e.g., using log or power transforms) to reduce the effect of extreme values.

Scale features using standardization (e.g., MinMaxScaler or StandardScaler) to ensure uniform treatment by algorithms.

Use feature selection techniques (e.g., correlation thresholding, mutual information, or tree-based feature importance) to focus on the most predictive variables.

Check for multicollinearity using correlation matrices or VIF to avoid redundant features

Checking duplicates

In [72]:

```
duplicates=df.duplicated()
```

In [73]:

```
duplicates.value_counts()
```

Out[73]:

```
False    11430
Name: count, dtype: int64
```

Label Encoding was applied to the url column to convert categorical values into numeric form. One-Hot Encoding was avoided because it would have significantly increased the number of columns due to the high number of unique URLs. Label Encoding keeps the dataset compact and efficient without adding unnecessary dimensions.

In [74]:

```
# Label Encoding Url Column

from sklearn.preprocessing import LabelEncoder
LE=LabelEncoder()
df['url']=LE.fit_transform(df['url'])


df['url'].value_counts()
```

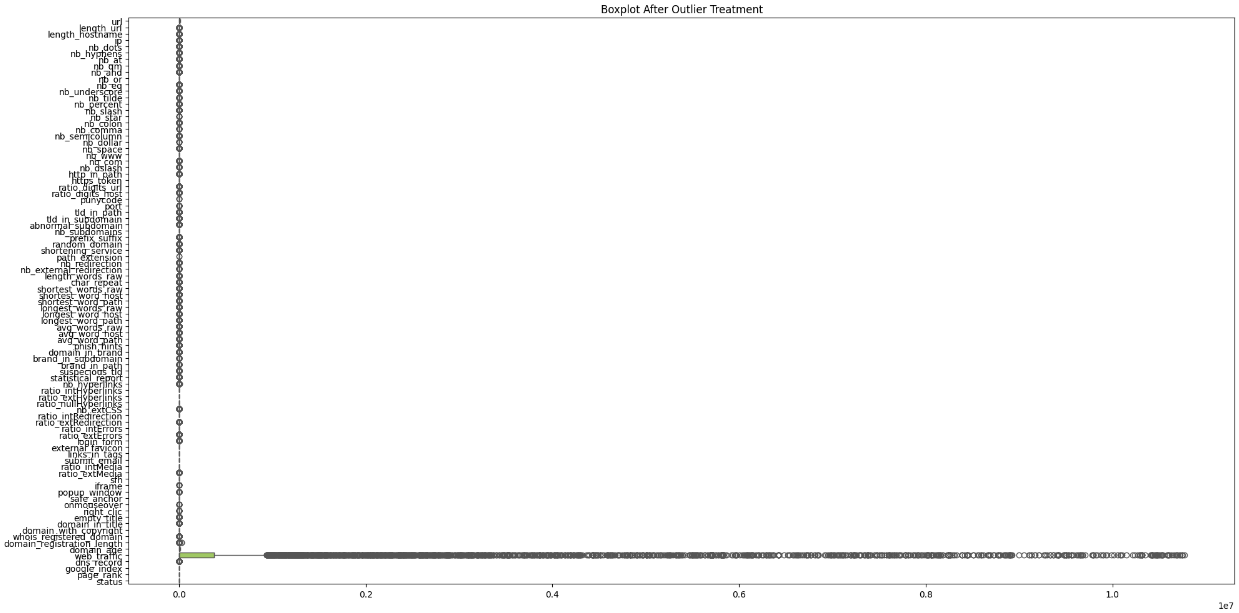
Out[74]:

```
url
1065     2
4501     1
10779    1
1315     1
9201     1
        ..
6539     1
819      1
9629     1
5956     1
62       1
```

```
Name: count, Length: 11429, dtype: int64
```

In [75]:

```
# Checking Outliers Using Boxplot
# Set figure size
plt.figure(figsize=(20, 10))

# Create boxplot for all numerical columns
sns.boxplot(data=df, orient='h', palette='Set2')

# Set title
plt.title('Boxplot After Outlier Treatment')
plt.tight_layout()
plt.show()
```



In [76]:

```
# Removing Outliers
# Replace Outliers with Median Statergy

for col in df.select_dtypes(include='number').columns:
    Q1 = df[col].quantile(0.25)
    Q3 = df[col].quantile(0.75)
    IQR = Q3 - Q1

    lower_bound = Q1 - 1.5 * IQR
    upper_bound = Q3 + 1.5 * IQR

    # Identify outliers
    outliers = (df[col] < lower_bound) | (df[col] > upper_bound)
    outlier_count = outliers.sum()

    if outlier_count > 0:
        replacement = df[col].median()
        df.loc[outliers, col] = replacement
        print(f"Replaced {outlier_count} outliers in '{col}' with median.")
    else:
        print(f"No outliers found in '{col}'.")
```

```
No outliers found in 'url'.
Replaced 620 outliers in 'length_url' with median.
Replaced 775 outliers in 'length_hostname' with median.
Replaced 1721 outliers in 'ip' with median.
Replaced 567 outliers in 'nb_dots' with median.
Replaced 1371 outliers in 'nb_hyphens' with median.
Replaced 245 outliers in 'nb_at' with median.
Replaced 1555 outliers in 'nb_qm' with median.
Replaced 761 outliers in 'nb_and' with median.
No outliers found in 'nb_or'.
Replaced 1564 outliers in 'nb_eq' with median.
Replaced 1695 outliers in 'nb_underscore' with median.
Replaced 76 outliers in 'nb_tilde' with median.
Replaced 355 outliers in 'nb_percent' with median.
Replaced 401 outliers in 'nb_slash' with median.
Replaced 8 outliers in 'nb_star' with median.
Replaced 197 outliers in 'nb_colon' with median.
Replaced 24 outliers in 'nb_comma' with median.
Replaced 248 outliers in 'nb_semicolumn' with median.
Replaced 11 outliers in 'nb_dollar' with median.
Replaced 210 outliers in 'nb_space' with median.
No outliers found in 'nb_www'.
Replaced 1327 outliers in 'nb_com' with median.
Replaced 75 outliers in 'nb_dslash' with median.
Replaced 150 outliers in 'http_in_path' with median.
No outliers found in 'https_token'.
Replaced 933 outliers in 'ratio_digits_url' with median.
Replaced 1503 outliers in 'ratio_digits_host' with median.
Replaced 4 outliers in 'punycode' with median.
Replaced 27 outliers in 'port' with median.
Replaced 750 outliers in 'tld_in_path' with median.
Replaced 573 outliers in 'tld_in_subdomain' with median.
Replaced 247 outliers in 'abnormal_subdomain' with median.
No outliers found in 'nb_subdomains'.
Replaced 2314 outliers in 'prefix_suffix' with median.
Replaced 952 outliers in 'random_domain' with median.
Replaced 1411 outliers in 'shortening_service' with median.
Replaced 2 outliers in 'path_extension' with median.
Replaced 166 outliers in 'nb_redirection' with median.
Replaced 36 outliers in 'nb_external_redirection' with median.
Replaced 264 outliers in 'length_words_raw' with median.
Replaced 310 outliers in 'char_repeat' with median.
Replaced 1435 outliers in 'shortest_words_raw' with median.
Replaced 1093 outliers in 'shortest_word_host' with median.
Replaced 428 outliers in 'shortest_word_path' with median.
Replaced 1035 outliers in 'longest_words_raw' with median.
Replaced 220 outliers in 'longest_word_host' with median.
Replaced 929 outliers in 'longest_word_path' with median.
Replaced 725 outliers in 'avg_words_raw' with median.
Replaced 568 outliers in 'avg_word_host' with median.
Replaced 282 outliers in 'avg_word_path' with median.
Replaced 2041 outliers in 'phish_hints' with median.
Replaced 1191 outliers in 'domain_in_brand' with median.
Replaced 47 outliers in 'brand_in_subdomain' with median.
Replaced 56 outliers in 'brand_in_path' with median.
Replaced 205 outliers in 'suspecious_tld' with median.
Replaced 377 outliers in 'statistical_report' with median.
Replaced 953 outliers in 'nb_hyperlinks' with median.
No outliers found in 'ratio_intHyperlinks'.
No outliers found in 'ratio_extHyperlinks'.
No outliers found in 'ratio_nullHyperlinks'.
Replaced 1019 outliers in 'nb_extCSS' with median.
No outliers found in 'ratio_intRedirection'.
Replaced 999 outliers in 'ratio_extRedirection' with median.
No outliers found in 'ratio_intErrors'.
Replaced 2149 outliers in 'ratio_extErrors' with median.
Replaced 727 outliers in 'login_form' with median.
No outliers found in 'external_favicon'.
No outliers found in 'links_in_tags'.
No outliers found in 'submit_email'.
No outliers found in 'ratio_intMedia'.
Replaced 2012 outliers in 'ratio_extMedia' with median.
No outliers found in 'sfh'.
Replaced 15 outliers in 'iframe' with median.
Replaced 69 outliers in 'popup_window' with median.
No outliers found in 'safe_anchor'.
Replaced 13 outliers in 'onmouseover' with median.
Replaced 16 outliers in 'right_clic' with median.
Replaced 1426 outliers in 'empty_title' with median.
Replaced 2562 outliers in 'domain_in_title' with median.
No outliers found in 'domain_with_copyright'.
Replaced 833 outliers in 'whois_registered_domain' with median.
Replaced 1529 outliers in 'domain_registration_length' with median.
No outliers found in 'domain_age'.
Replaced 2138 outliers in 'web_traffic' with median.
Replaced 230 outliers in 'dns_record' with median.
No outliers found in 'google_index'.
No outliers found in 'page_rank'.
No outliers found in 'status'.
```

In [77]:

```
# Checking Correlation
df.corr()['status']
```

Out[77]:

```
url                             -0.290971
length_url                       0.217898
length_hostname                  0.105306
ip                                    NaN
nb_dots                          0.109748
nb_dots                       0.109748
nb_hyphens                    0.158158
nb_at                              NaN
nb_qm                              NaN
nb_and                             NaN
nb_or                              NaN
nb_eq                              NaN
nb_underscore                      NaN
nb_tilde                           NaN
nb_percent                         NaN
nb_slash                      0.201618
nb_star                            NaN
nb_colon                           NaN
nb_comma                           NaN
nb_semicolumn                      NaN
nb_dollar                          NaN
nb_space                           NaN
nb_www                       -0.443468
nb_com                             NaN
nb_dslash                          NaN
http_in_path                       NaN
https_token                   0.114669
ratio_digits_url              0.222690
ratio_digits_host                  NaN
punycode                           NaN
port                               NaN
tld_in_path                        NaN
tld_in_subdomain                   NaN
abnormal_subdomain                 NaN
nb_subdomains                 0.112891
prefix_suffix                      NaN
random_domain                      NaN
shortening_service                 NaN
path_extension                     NaN
nb_redirection               -0.043685
nb_external_redirection            NaN
length_words_raw              0.195110
char_repeat                  -0.122545
shortest_words_raw           -0.167907
shortest_word_host            0.102649
shortest_word_path            0.094549
longest_words_raw             0.164277
longest_word_host             0.094016
longest_word_path             0.187312
avg_words_raw                 0.122929
avg_word_host                 0.140864
avg_word_path                 0.229877
phish_hints                        NaN
domain_in_brand                    NaN
brand_in_subdomain                 NaN
brand_in_path                      NaN
suspecious_tld                     NaN
statistical_report                 NaN
nb_hyperlinks                -0.442032
ratio_intHyperlinks          -0.243982
ratio_extHyperlinks           0.083357
ratio_nullHyperlinks               NaN
nb_extCSS                    -0.080464
ratio_intRedirection               NaN
ratio_extRedirection         -0.267394
ratio_intErrors                    NaN
ratio_extErrors              -0.179422
login_form                         NaN
external_favicon             -0.146565
links_in_tags                -0.184401
submit_email                       NaN
ratio_intMedia               -0.193333
ratio_extMedia               -0.098609
sfh                                NaN
iframe                             NaN
popup_window                       NaN
safe_anchor                  -0.173397
onmouseover                        NaN
right_clic                         NaN
empty_title                        NaN
domain_in_title                    NaN
domain_with_copyright        -0.173098
whois_registered_domain            NaN
domain_registration_length   -0.146138
domain_age                   -0.331889
web_traffic                  -0.203653
dns_record                         NaN
google_index                  0.731171
page_rank                    -0.511137
status                        1.000000
Name: status, dtype: float64
```

A Ranked list of features based on Variance Inflation Factor

(VIF)

In [78]:

```
from statsmodels.stats.outliers_influence import variance_inflation_factor
# Checking VIF:
def calculate_vif(dataset):
    vif = pd.DataFrame()
    vif['features'] = dataset.columns
    vif['VIF_Values'] = [variance_inflation_factor(dataset.values,i) for i in range(dataset.shape[1])]
    vif['VIF_Values'] = round(vif['VIF_Values'], 2)
    vif = vif.sort_values(by = 'VIF_Values', ascending=False)
    return (vif)

calculate_vif(df.drop('status',axis = 1))
```

Out[78]:

	features	VIF_Values
0	url	9.42
25	https_token	7.70
47	longest_word_path	5.86
33	nb_subdomains	5.36
46	longest_word_host	5.18
4	nb_dots	4.88
58	ratio_intHyperlinks	4.49
45	longest_words_raw	4.38
21	nb_www	4.21
50	avg_word_path	4.16
40	length_words_raw	4.08
1	length_url	3.81
49	avg_word_host	3.45
68	links_in_tags	3.04
48	avg_words_raw	2.49
2	length_hostname	2.36
44	shortest_word_path	2.35
67	external_favicon	2.33
59	ratio_extHyperlinks	2.32
41	char_repeat	2.25
87	page_rank	2.20
42	shortest_words_raw	1.88
86	google_index	1.88
14	nb_slash	1.80
43	shortest_word_host	1.76
83	domain_age	1.70
70	ratio_intMedia	1.65
57	nb_hyperlinks	1.54
61	nb_extCSS	1.49
75	safe_anchor	1.47
26	ratio_digits_url	1.28
38	nb_redirection	1.24
63	ratio_extRedirection	1.20
80	domain_with_copyright	1.18
82	domain_registration_length	1.18
5	nb_hyphens	1.15

71	ratio_extMedia	1.11
65	ratio_extErrors	1.08
84	web_traffic	1.07
16	nb_colon	0.00
79	domain_in_title	0.00
3	ip	NaN
6	nb_at	NaN
7	nb_qm	NaN
8	nb_and	NaN
9	nb_or	NaN
10	nb_eq	NaN
11	nb_underscore	NaN
12	nb_tilde	NaN
13	nb_percent	NaN
15	nb_star	NaN
17	nb_comma	NaN
18	nb_semicolumn	NaN
19	nb_dollar	NaN
20	nb_space	NaN
22	nb_com	NaN
23	nb_dslash	NaN
24	http_in_path	NaN
27	ratio_digits_host	NaN
28	punycode	NaN
29	port	NaN
30	tld_in_path	NaN
31	tld_in_subdomain	NaN
32	abnormal_subdomain	NaN
34	prefix_suffix	NaN
35	random_domain	NaN
36	shortening_service	NaN
37	path_extension	NaN
39	nb_external_redirection	NaN
51	phish_hints	NaN
52	domain_in_brand	NaN
53	brand_in_subdomain	NaN
54	brand_in_path	NaN
55	suspecious_tld	NaN
56	statistical_report	NaN
60	ratio_nullHyperlinks	NaN
62	ratio_intRedirection	NaN
64	ratio_intErrors	NaN
66	login_form	NaN
69	submit_email	NaN
72	sfh	NaN
73	iframe	NaN
74	popup_window	NaN
76	onmouseover	NaN

77	right_clic	NaN
78	empty_title	NaN
81	whois_registered_domain	NaN
85	dns_record	NaN

In [79]:

```
# Spliting Data into Independent And target Column
X=df.drop(columns='status')
y=df['status']
```

In [80]:

```
from sklearn.model_selection import train_test_split
X_train,X_test,y_train,y_test=train_test_split(X,y,train_size=0.70,random_state=42)
```

In [81]:

```
X_train_original = X_train.copy()
```

Scaling Technique:- Robust Scaler

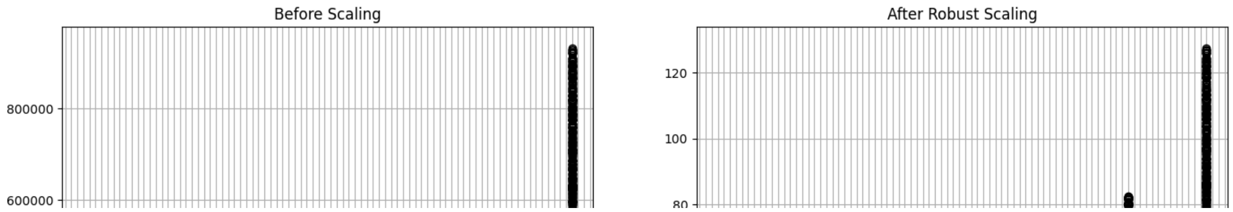
Robust Scaler was used to handle outliers effectively, as boxplots showed many extreme values in the numerical features. It scales data based on the median and IQR, making it less sensitive to outliers compared to StandardScaler or MinMaxScaler.

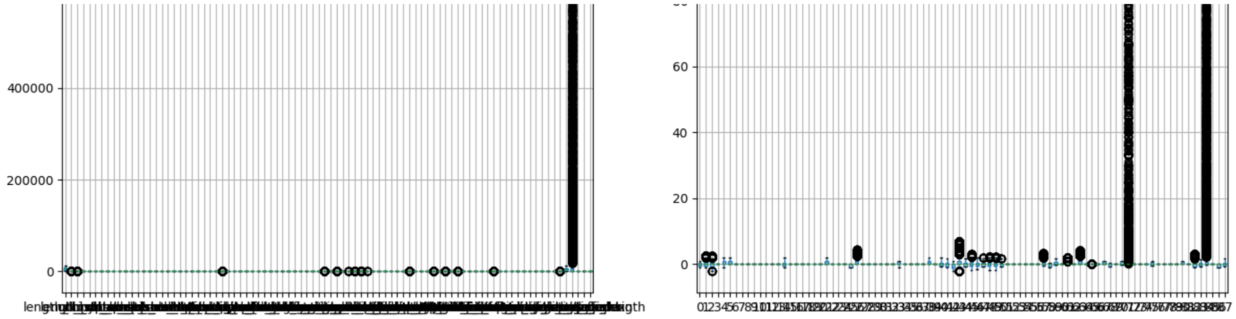
In [82]:

```
from sklearn.preprocessing import MinMaxScaler,StandardScaler,RobustScaler
scaler=RobustScaler()
X_train=scaler.fit_transform(X_train)
X_test=scaler.transform(X_test)
```

In [83]:

```
import pandas as pd
import matplotlib.pyplot as plt
X_train_scaled=X_train.copy()
# If X_train is a NumPy array, convert it to a DataFrame
X_train_df = pd.DataFrame(X_train_original)
X_train_scaled_df = pd.DataFrame(X_train_scaled)

# Plot before and after scaling side by side
plt.figure(figsize=(14, 6))

plt.subplot(1, 2, 1)
X_train_df.boxplot()
plt.title("Before Scaling")

plt.subplot(1, 2, 2)
X_train_scaled_df.boxplot()
plt.title("After Robust Scaling")

plt.tight_layout()
plt.show()
```





Appling PCA For Dimension Reduction

Displaying Variance Ratio

In [84]:

```python
from sklearn.preprocessing import RobustScaler
from sklearn.decomposition import PCA
# Using PCA Concept:

# Step 1: Standardize the data

scaler = RobustScaler()
X_scaled = scaler.fit_transform(df)

# Step 2: Determine number of components to retain 90% variance

for i in range(1, df.shape[1] + 1):
    pca = PCA(n_components=i)
    pca.fit(X_scaled)
    evr = np.cumsum(pca.explained_variance_ratio_)
    if evr[i - 1] >= 0.90:
        pcs = i
        break

print("Explained Variance Ratio:", evr)
print("Number of components selected:", pcs)

# Step 3: Apply PCA

pca = PCA(n_components=pcs)
pca_data = pca.fit_transform(X_scaled)

# Step 4: Create DataFrame

pca_columns = [f'PC{j+1}' for j in range(pcs)]
pca_df = pd.DataFrame(pca_data, columns=pca_columns)

# Step 5: Join Target Column with PCA:

pca_df = pca_df.join(df['status'], how = 'left')

pca_df
```

```
Explained Variance Ratio: [0.57046592 0.970943  ]
Number of components selected: 2
```

Out[84]:

	PC1	PC2	status
0	-7.873171	-5.366775	0
1	-6.311023	14.559213	1
2	-7.736537	-5.395951	1
3	5.910682	-2.885161	0
4	-6.772865	-5.403439	0
...	...	...	...
11425	-1.094859	73.295599	0
11426	-7.901367	-5.380939	1
11427	-7.912587	-5.342337	0
11428	-7.664649	-5.369196	0
11429	-7.893053	-5.346426	1

11429	7.893053	5.540420	1

11430 rows × 3 columns

Training Machine Learning Model

In [85]:

```
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score

X=pca_df.drop(columns='status')
y=pca_df['status']

X_train,X_test,y_train,y_test= train_test_split(X,y,train_size=0.70,random_state=42)
```

Logistic Regression

In [108…

```
from sklearn.linear_model import LogisticRegression
lr=LogisticRegression()
lr.fit(X_train,y_train)
y_pred_lr=lr.predict(X_test)

accuracy_lr=accuracy_score(y_test,y_pred_lr)
print("Accuracy",(accuracy_lr)*100,'%')
```

```
Accuracy 60.96209912536443 %
```

Random Forest Classifier

In [109…

```
from sklearn.ensemble import RandomForestClassifier
rfc=RandomForestClassifier()
rfc.fit(X_train,y_train)
y_pred_rfc=rfc.predict(X_test)

accuracy_rfc=accuracy_score(y_test,y_pred_rfc)
print("Accuracy:",(accuracy_rfc)*100,"%")
```

```
Accuracy: 84.3731778425656 %
```

Decission Tree Classifier

In [111…

```
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import accuracy_score  # Make sure this is the function

# Train the model
dfc = DecisionTreeClassifier()
dfc.fit(X_train, y_train)

# Predict
y_pred_dc = dfc.predict(X_test)

# Accuracy
accuracy = accuracy_score(y_test, y_pred_dc)
print("Accuracy:", (accuracy)*100,"%")
```

```
Accuracy: 80.29154518950438 %
```