In [1]:
```
# Importing Data Manipilation Libraries
import pandas as pd
import numpy as np

# Import Data Visualization Libraries

import matplotlib.pyplot as plt
import seaborn as sns
from scipy import stats

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

# Import Data Logging Libraries
import logging
logging.basicConfig(level = logging.INFO,
                        filename = 'model.log',
                        filemode = 'w',
                        format = '%(asctime)s - %(levelname)s - %(message)s')
```

In [2]:
```
pd.set_option('display.max_columns', None)
pd.set_option('display.max_rows', 100)
```

Loading Dataset

In [3]:
```
# Loading the dataset

url = 'https://raw.githubusercontent.com/mukeshmagar543/CODEB_Internship/refs/heads/main/dataset_phishi

df = pd.read_csv(url)

df.sample(frac = 1) # Data Shuffle
```

Out[3]:

	url	length_url	length_hostname	ip	nb_dots	nb_hyphens	r
10753	https://docs.google.com/document/d/1gy-xysaRMQ...	96	15	0	2	4	
7775	http://www.timebie.com/timezone/greenwichmeanp...	60	15	0	3	0	
704	http://www.apronus.com/music/flashpiano.htm	43	15	0	3	0	
5812	https://www.liesegang-partner.com	33	25	0	2	1	
4155	http://www.jinjitter.jp/	24	16	0	2	0	
...	...	...	...	...	...	...	
5495	http://blog.wanken.com/7644/typefaces-of-the-w...	58	15	0	2	4	
7730	https://data.gov.tw/	20	11	0	2	0	
1250	https://support-appleId.com.secureupdate.duila...	128	50	1	4	1	
5763	http://homologacao.xocovid19.com.br	35	28	0	3	0	
5355	https://google.com.hk/	22	13	0	2	0	

11430 rows × 89 columns

Getting Information about Dataset Like which

column is object and which column is numerical

In [4]:
```
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
```

Checking Null Values

- There is No Null Values are present in the given dataset.

In [5]:
```
df.isnull().sum()
```

Out[5]:
```
url                         0
length_url                  0
length_hostname             0
ip                          0
nb_dots                     0
nb_hyphens                  0
nb_at                       0
nb_qm                       0
nb_and                      0
nb_or                       0
nb_eq                       0
nb_underscore               0
nb_tilde                    0
nb_percent                  0
nb_slash                    0
nb_star                     0
nb_colon                    0
nb_comma                    0
nb_semicolumn               0
nb_dollar                   0
nb_space                    0
nb_www                      0
nb_com                      0
nb_dslash                   0
http_in_path                0
https_token                 0
ratio_digits_url            0
ratio_digits_host           0
punycode                    0
port                        0
tld_in_path                 0
tld_in_subdomain            0
abnormal_subdomain          0
nb_subdomains               0
prefix_suffix               0
```

```
prefix_suffix                0
random_domain                0
shortening_service           0
path_extension               0
nb_redirection               0
nb_external_redirection      0
length_words_raw             0
char_repeat                  0
shortest_words_raw           0
shortest_word_host           0
shortest_word_path           0
longest_words_raw            0
longest_word_host            0
longest_word_path            0
avg_words_raw                0
avg_word_host                0
avg_word_path                0
phish_hints                  0
domain_in_brand              0
brand_in_subdomain           0
brand_in_path                0
suspecious_tld               0
statistical_report           0
nb_hyperlinks                0
ratio_intHyperlinks          0
ratio_extHyperlinks          0
ratio_nullHyperlinks         0
nb_extCSS                    0
ratio_intRedirection         0
ratio_extRedirection         0
ratio_intErrors              0
ratio_extErrors              0
login_form                   0
external_favicon             0
links_in_tags                0
submit_email                 0
ratio_intMedia               0
ratio_extMedia               0
sfh                          0
iframe                       0
popup_window                 0
safe_anchor                  0
onmouseover                  0
right_clic                   0
empty_title                  0
domain_in_title              0
domain_with_copyright        0
whois_registered_domain      0
domain_registration_length   0
domain_age                   0
web_traffic                  0
dns_record                   0
google_index                 0
page_rank                    0
status                       0
dtype: int64
```

Descriptive Analysis

In [6]:
```
df.describe()
```

Out[6]:

	length_url	length_hostname	ip	nb_dots	nb_hyphens	nb_at	nb_qm	
count	11430.000000	11430.000000	11430.000000	11430.000000	11430.000000	11430.000000	11430.000000	114
mean	61.126684	21.090289	0.150569	2.480752	0.997550	0.022222	0.141207	
std	55.297318	10.777171	0.357644	1.369686	2.087087	0.155500	0.364456	
min	12.000000	4.000000	0.000000	1.000000	0.000000	0.000000	0.000000	
25%	33.000000	15.000000	0.000000	2.000000	0.000000	0.000000	0.000000	

25%	33.000000	15.000000	0.000000	2.000000	0.000000	0.000000	0.000000
50%	47.000000	19.000000	0.000000	2.000000	0.000000	0.000000	0.000000
75%	71.000000	24.000000	0.000000	3.000000	1.000000	0.000000	0.000000
max	1641.000000	214.000000	1.000000	24.000000	43.000000	4.000000	3.000000

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 [7]:
```
numerical_columns = df.select_dtypes(exclude= 'object')
numerical_columns
```

Out[7]:

	length_url	length_hostname	ip	nb_dots	nb_hyphens	nb_at	nb_qm	nb_and	nb_or	nb_eq	nb_undersc
0	37	19	0	3	0	0	0	0	0	0	
1	77	23	1	1	0	0	0	0	0	0	
2	126	50	1	4	1	0	1	2	0	3	
3	18	11	0	2	0	0	0	0	0	0	
4	55	15	0	2	2	0	0	0	0	0	
...	...	...	...	...	...	...	...	...	...	...	
11425	45	17	0	2	0	0	0	0	0	0	
11426	84	18	0	5	0	1	1	0	0	1	
11427	105	16	1	2	6	0	1	0	0	1	
11428	38	30	0	2	0	0	0	0	0	0	
11429	477	14	1	24	0	1	1	9	0	9	

11430 rows × 87 columns

In [8]:
```
# Descriptive statistics
from collections import OrderedDict

stats = []

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)

# Convert to DataFrame
report = pd.DataFrame(stats)

report
```

Out[8]:

	Feature	Minimum	Maximum	Mean	Mode	25%	75%	
0	length_url	12.0	1.641000e+03	61.126684	26.0	33.000000	71.000000	38.0
1	length_hostname	4.0	2.140000e+02	21.090289	16.0	15.000000	24.000000	9.0
2	ip	0.0	1.000000e+00	0.150569	0.0	0.000000	0.000000	0.0
3	nb_dots	1.0	2.400000e+01	2.480752	2.0	2.000000	3.000000	1.0
4	nb_hyphens	0.0	4.300000e+01	0.997550	0.0	0.000000	1.000000	1.0
5	nb_at	0.0	4.000000e+00	0.022222	0.0	0.000000	0.000000	0.0
6	nb_qm	0.0	3.000000e+00	0.141207	0.0	0.000000	0.000000	0.0
7	nb_and	0.0	1.900000e+01	0.162292	0.0	0.000000	0.000000	0.0
8	nb_or	0.0	0.000000e+00	0.000000	0.0	0.000000	0.000000	0.0
9	nb_eq	0.0	1.900000e+01	0.293176	0.0	0.000000	0.000000	0.0
10	nb_underscore	0.0	1.800000e+01	0.322660	0.0	0.000000	0.000000	0.0
11	nb_tilde	0.0	1.000000e+00	0.006649	0.0	0.000000	0.000000	0.0
12	nb_percent	0.0	9.600000e+01	0.123097	0.0	0.000000	0.000000	0.0
13	nb_slash	2.0	3.300000e+01	4.289589	3.0	3.000000	5.000000	2.0
14	nb_star	0.0	1.000000e+00	0.000700	0.0	0.000000	0.000000	0.0
15	nb_colon	1.0	7.000000e+00	1.027909	1.0	1.000000	1.000000	0.0
16	nb_comma	0.0	4.000000e+00	0.004024	0.0	0.000000	0.000000	0.0
17	nb_semicolumn	0.0	2.000000e+01	0.062292	0.0	0.000000	0.000000	0.0
18	nb_dollar	0.0	6.000000e+00	0.001925	0.0	0.000000	0.000000	0.0
19	nb_space	0.0	1.800000e+01	0.034821	0.0	0.000000	0.000000	0.0
20	nb_www	0.0	2.000000e+00	0.448469	0.0	0.000000	1.000000	1.0
21	nb_com	0.0	6.000000e+00	0.127997	0.0	0.000000	0.000000	0.0
22	nb_dslash	0.0	1.000000e+00	0.006562	0.0	0.000000	0.000000	0.0
23	http_in_path	0.0	4.000000e+00	0.016710	0.0	0.000000	0.000000	0.0
24	https_token	0.0	1.000000e+00	0.610936	1.0	0.000000	1.000000	1.0
25	ratio_digits_url	0.0	7.238806e-01	0.053137	0.0	0.000000	0.079365	0.0
26	ratio_digits_host	0.0	8.000000e-01	0.025024	0.0	0.000000	0.000000	0.0
27	punycode	0.0	1.000000e+00	0.000350	0.0	0.000000	0.000000	0.0
28	port	0.0	1.000000e+00	0.002362	0.0	0.000000	0.000000	0.0
29	tld_in_path	0.0	1.000000e+00	0.065617	0.0	0.000000	0.000000	0.0
30	tld_in_subdomain	0.0	1.000000e+00	0.050131	0.0	0.000000	0.000000	0.0
31	abnormal_subdomain	0.0	1.000000e+00	0.021610	0.0	0.000000	0.000000	0.0
32	nb_subdomains	1.0	3.000000e+00	2.231671	2.0	2.000000	3.000000	1.0
33	prefix_suffix	0.0	1.000000e+00	0.202450	0.0	0.000000	0.000000	0.0
34	random_domain	0.0	1.000000e+00	0.083290	0.0	0.000000	0.000000	0.0
35	shortening_service	0.0	1.000000e+00	0.123447	0.0	0.000000	0.000000	0.0
36	path_extension	0.0	1.000000e+00	0.000175	0.0	0.000000	0.000000	0.0
37	nb_redirection	0.0	6.000000e+00	0.498250	0.0	0.000000	1.000000	1.0

38	nb_external_redirection	0.0	1.000000e+00	0.003150	0.0	0.000000	0.000000	0.0
39	length_words_raw	1.0	1.060000e+02	6.232808	2.0	2.000000	8.000000	6.0
40	char_repeat	0.0	1.460000e+02	2.927472	3.0	1.000000	4.000000	3.0
41	shortest_words_raw	1.0	3.100000e+01	3.127297	3.0	2.000000	3.000000	1.0
42	shortest_word_host	1.0	3.900000e+01	5.019773	3.0	3.000000	6.000000	3.0
43	shortest_word_path	0.0	4.000000e+01	2.398950	0.0	0.000000	3.000000	3.0
44	longest_words_raw	2.0	8.290000e+02	15.393876	9.0	9.000000	16.000000	7.0
45	longest_word_host	1.0	6.200000e+01	10.467979	9.0	7.000000	13.000000	6.0
46	longest_word_path	0.0	8.290000e+02	10.561505	0.0	0.000000	11.000000	11.0
47	avg_words_raw	2.0	1.282500e+02	7.258882	6.0	5.250000	8.000000	2.7
48	avg_word_host	1.0	3.900000e+01	7.678075	5.0	5.250000	9.000000	3.7
49	avg_word_path	0.0	2.500000e+02	5.092425	0.0	0.000000	6.714286	6.7
50	phish_hints	0.0	1.000000e+01	0.327734	0.0	0.000000	0.000000	0.0
51	domain_in_brand	0.0	1.000000e+00	0.104199	0.0	0.000000	0.000000	0.0
52	brand_in_subdomain	0.0	1.000000e+00	0.004112	0.0	0.000000	0.000000	0.0
53	brand_in_path	0.0	1.000000e+00	0.004899	0.0	0.000000	0.000000	0.0
54	suspecious_tld	0.0	1.000000e+00	0.017935	0.0	0.000000	0.000000	0.0
55	statistical_report	0.0	2.000000e+00	0.059755	0.0	0.000000	0.000000	0.0
56	nb_hyperlinks	0.0	4.659000e+03	87.189764	0.0	9.000000	101.000000	92.0
57	ratio_intHyperlinks	0.0	1.000000e+00	0.602457	0.0	0.224991	0.944767	0.7
58	ratio_extHyperlinks	0.0	1.000000e+00	0.276720	0.0	0.000000	0.474840	0.4
59	ratio_nullHyperlinks	0.0	0.000000e+00	0.000000	0.0	0.000000	0.000000	0.0
60	nb_extCSS	0.0	1.240000e+02	0.784864	0.0	0.000000	1.000000	1.0
61	ratio_intRedirection	0.0	0.000000e+00	0.000000	0.0	0.000000	0.000000	0.0
62	ratio_extRedirection	0.0	2.000000e+00	0.158926	0.0	0.000000	0.230769	0.2
63	ratio_intErrors	0.0	0.000000e+00	0.000000	0.0	0.000000	0.000000	0.0
64	ratio_extErrors	0.0	1.000000e+00	0.062469	0.0	0.000000	0.034483	0.0
65	login_form	0.0	1.000000e+00	0.063605	0.0	0.000000	0.000000	0.0
66	external_favicon	0.0	1.000000e+00	0.442170	0.0	0.000000	1.000000	1.0
67	links_in_tags	0.0	1.000000e+02	51.978211	0.0	0.000000	98.061004	98.0
68	submit_email	0.0	0.000000e+00	0.000000	0.0	0.000000	0.000000	0.0
69	ratio_intMedia	0.0	1.000000e+02	42.870444	0.0	0.000000	100.000000	100.0
70	ratio_extMedia	0.0	1.000000e+02	23.236293	0.0	0.000000	33.333333	33.3
71	sfh	0.0	0.000000e+00	0.000000	0.0	0.000000	0.000000	0.0
72	iframe	0.0	1.000000e+00	0.001312	0.0	0.000000	0.000000	0.0
73	popup_window	0.0	1.000000e+00	0.006037	0.0	0.000000	0.000000	0.0
74	safe_anchor	0.0	1.000000e+02	37.063922	0.0	0.000000	75.000000	75.0
75	onmouseover	0.0	1.000000e+00	0.001137	0.0	0.000000	0.000000	0.0
76	right_clic	0.0	1.000000e+00	0.001400	0.0	0.000000	0.000000	0.0
77	empty_title	0.0	1.000000e+00	0.124759	0.0	0.000000	0.000000	0.0

78	domain_in_title	0.0	1.000000e+00	0.775853	1.0	1.000000	1.000000	0.0
79	domain_with_copyright	0.0	1.000000e+00	0.439545	0.0	0.000000	1.000000	1.0
80	whois_registered_domain	0.0	1.000000e+00	0.072878	0.0	0.000000	0.000000	0.0
81	domain_registration_length	-1.0	2.982900e+04	492.532196	0.0	84.000000	449.000000	365.0
82	domain_age	-12.0	1.287400e+04	4062.543745	-1.0	972.250000	7026.750000	6054.5
83	web_traffic	0.0	1.076799e+07	856756.643307	0.0	0.000000	373845.500000	373845.5
84	dns_record	0.0	1.000000e+00	0.020122	0.0	0.000000	0.000000	0.0
85	google_index	0.0	1.000000e+00	0.533946	1.0	0.000000	1.000000	1.0
86	page_rank	0.0	1.000000e+01	3.185739	0.0	1.000000	5.000000	4.0

Frequency distribution for categorical features

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 [9]:
```
# Frequency distribution for categorical features (if any)
for col in df.columns:
    if df[col].dtype == 'object':
        print(f"\nFrequency distribution for {col}:\n")
        print(df[col].value_counts())
```

```
Frequency distribution for url:

url
http://e710z0ear.du.r.appspot.com/c:/users/user/downlo
2
https://lt.mydplr.com/16672ac75448ecdb528e1c663c0df3a7-f10ed321df1a4fbc893c86fbb12f0913
1
http://appleid.apple.com-app.es/
1
http://174.139.46.123/ap/signin?openid.pape.max_auth_age=0&amp;openid.return_to=https%3A%2F%2Fwww.amazon.
co.jp%2F%3Fref_%3Dnav_em_hd_re_signin&amp;openid.identity=http%3A%2F%2Fspecs.openid.net%2Fauth%2F2.0%2Fid
entifier_select&amp;openid.assoc_handle=jpflex&amp;openid.mode=checkid_setup&amp;key=a@b.c&amp;openid.cla
imed_id=http%3A%2F%2Fspecs.openid.net%2Fauth%2F2.0%2Fidentifier_select&amp;openid.ns=http%3A%2F%2Fspecs.o
penid.net%2Fauth%2F2.0&amp;&amp;ref_=nav_em_hd_clc_signin    1
http://www.crestonwood.com/router.php
1

..
https://www.dissernet.org/
1
https://workprotocoles-com.webs.com/
1
http://www.vg247.com/2017/04/24/best-nintendo-switch-games/
1
https://www.facebook.com/Publictransporthub/
1
http://www.game.co.uk/en/games/nintendo-switch/nintendo-switch/
1
Name: count, Length: 11429, dtype: int64

Frequency distribution for status:
```

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

The target label is balanced — There is no need to use SMOTE techniques to Blanace the Target columm.

In [10]:
```
df['status'].mode()
```

Out[10]:
```
0    legitimate
1      phishing
Name: status, dtype: object
```

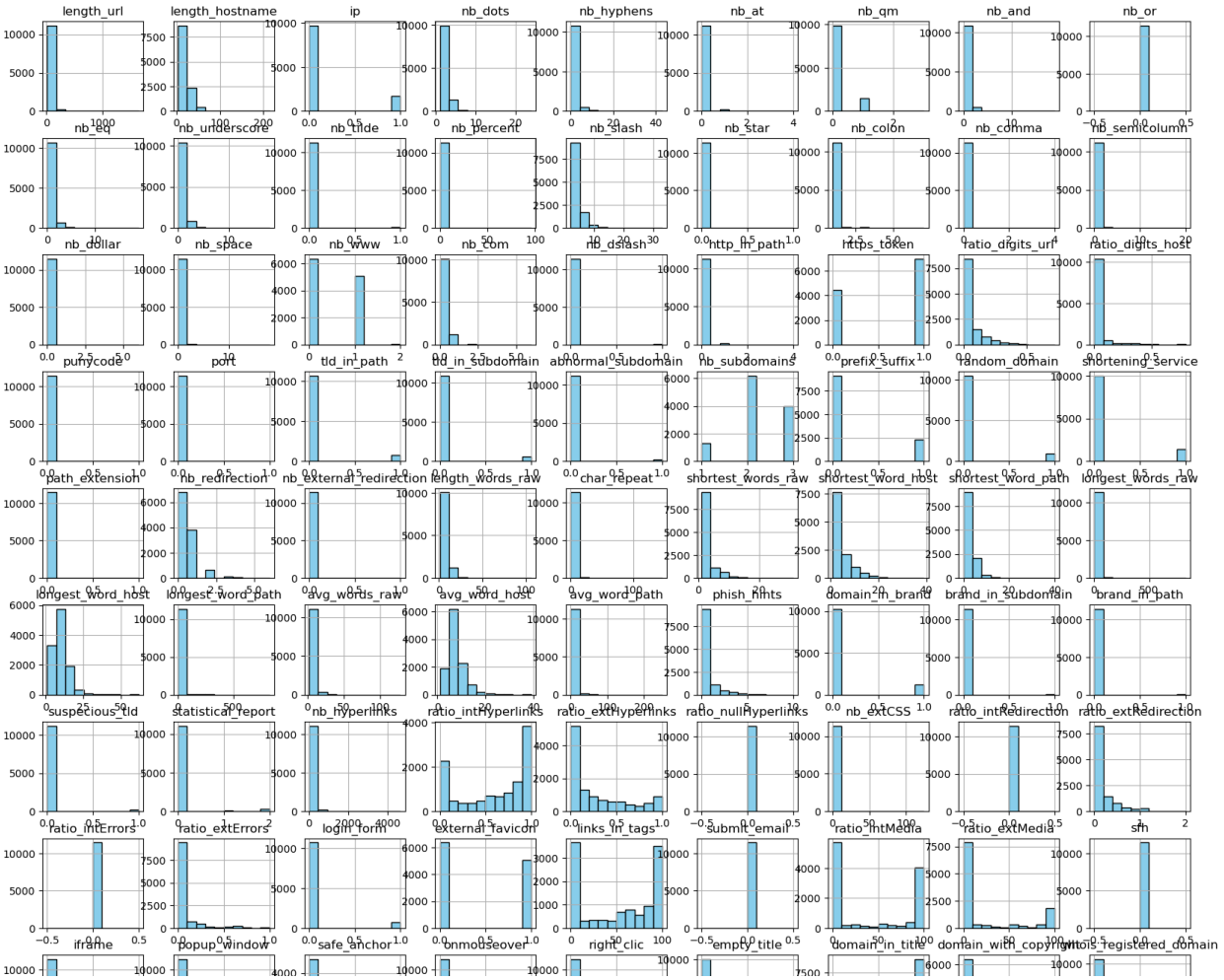
In [11]:
```
df['url'].mode()
```

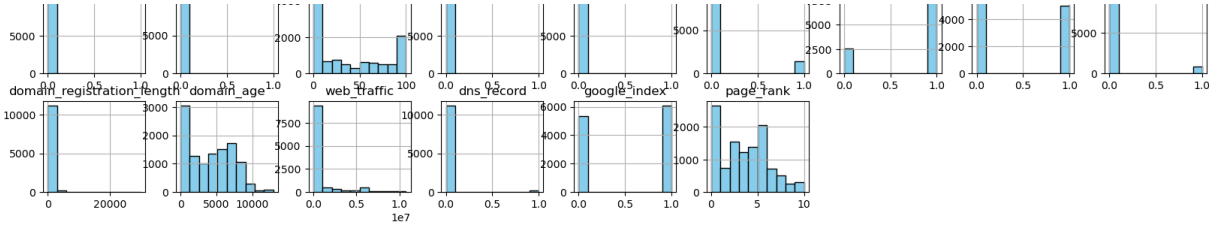
Out[11]:
```
0    http://e710z0ear.du.r.appspot.com/c:/users/use...
Name: url, dtype: object
```

Histogram

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

In [12]:
```
# Histograms for numerical features
numerical_columns.hist(figsize=(20, 20),bins= 10, color= 'skyblue', edgecolor= 'black')
plt.title("Histogram")
plt.xlabel("Value")
plt.ylabel("Frequency")
plt.show()
```

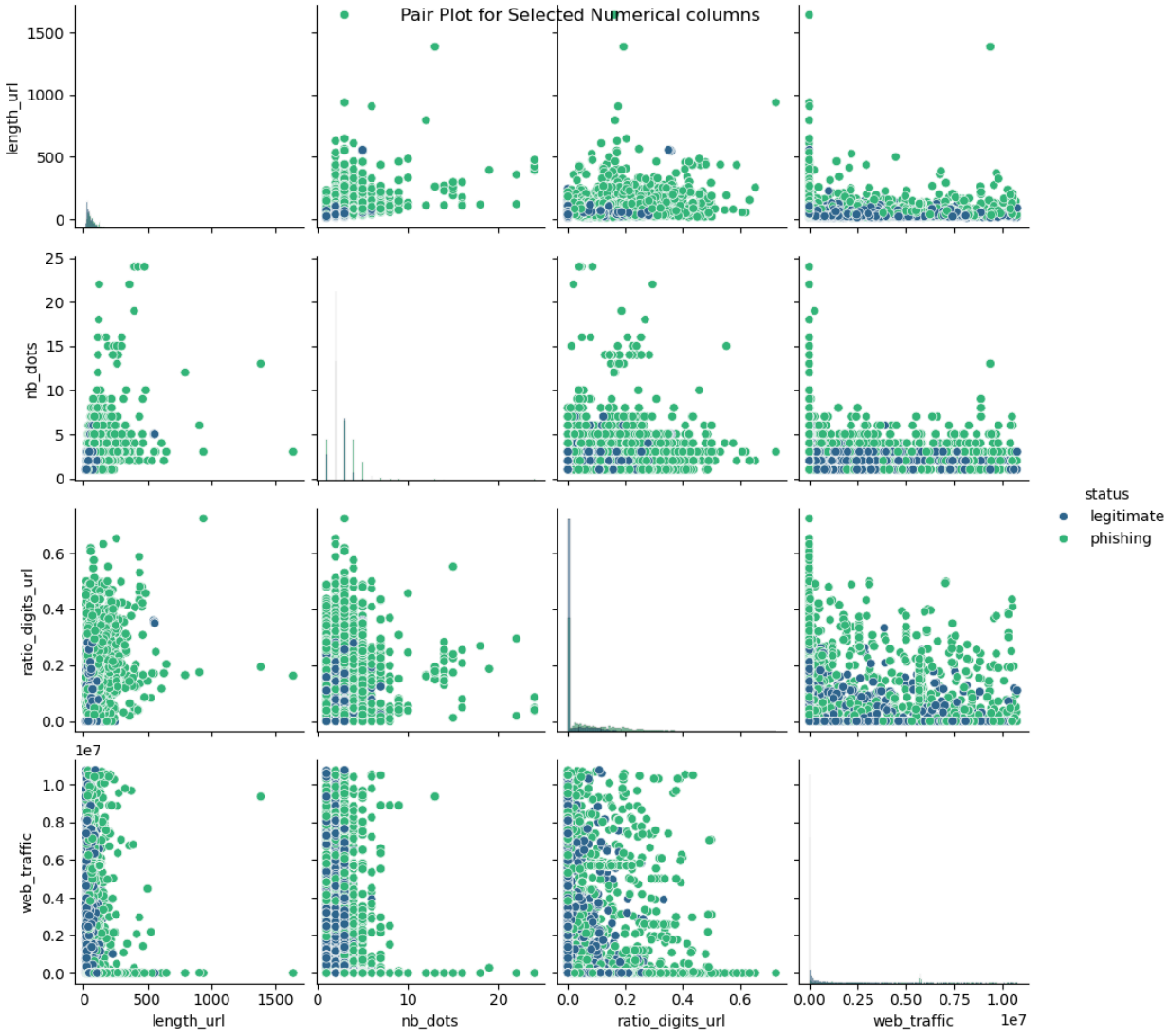




Pair Plot

- We have use only selected important features to create the Pair Plot
- 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.

In [13]:
```
selected_features = ['length_url', 'nb_dots', 'ratio_digits_url', 'web_traffic', 'status']
# plot pair plot
sns.pairplot(df[selected_features], hue='status', diag_kind='hist', palette= 'viridis')
plt.suptitle('Pair Plot for Selected Numerical columns')
plt.show()
```



Using Replace function to 'legitimate' and 'phishing' into 0 and 1 — readying the target for machine learning models.

In [14]:
```
df['status'] = df['status'].replace({'legitimate' : 0, 'phishing' : 1})
```

Label encoding to url column — to convert the categorical data into numerical

In [15]:
```
# Using Label Encoding in Url column
from sklearn.preprocessing import LabelEncoder
le = LabelEncoder()

df['url'] = le.fit_transform(df['url'])
df['url'].value_counts()
```

Out[15]:
```
url
1065    2
8258    1
363     1
62      1
4501    1
       ..
9799    1
9324    1
6684    1
9920    1
4919    1
Name: count, Length: 11429, dtype: int64
```

Insights and Recommendations

- Features like `web_traffic` , `SSLfinal_State` , and `page_rank` are crucial indicators.
- The Dataset has huge amount of Outliers.
- Outliers can be capped using the IQR method.
- Use `RobustScaler` to normalize numerical features.
- Remove redundant features with high multicollinearity.
- The target is balance hence, there is no need for SMOTE.
- We can use Feature Engineering.
- The Dataset have doesn't have any null values.

Checking Duplicates

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.

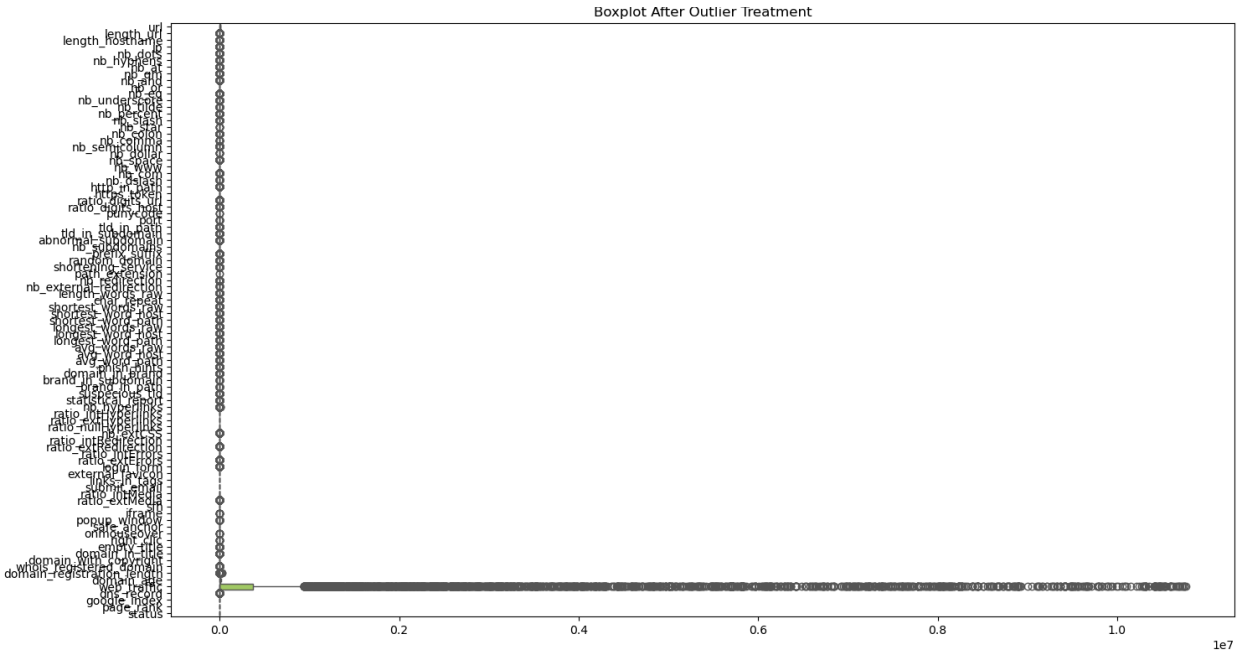
In [16]:
```
# Checking Duplicates
duplicates = df.duplicated()
duplicates.value_counts()
```

Out[16]:
```
False    11430
Name: count, dtype: int64
```

In [17]:
```
# Set figure size
plt.figure(figsize=(15, 8))

# 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 [19]:
```
# Spliting Data into Independent And target Column
X=df.drop(columns='status')
y=df['status']
```

In [20]:
```
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 [21]:
```
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 [22]:
```
from sklearn.preprocessing import MinMaxScaler,StandardScaler,RobustScaler
scaler=RobustScaler()
X_train=scaler.fit_transform(X_train)
X_test=scaler.transform(X_test)
```

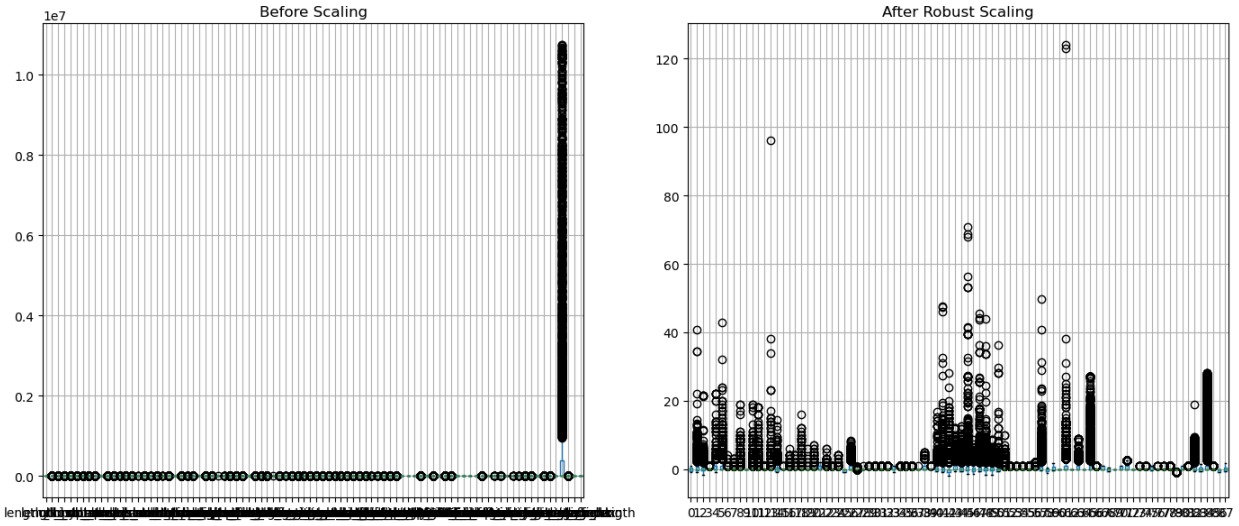
In [23]:
```
X_train_scaled=X_train.copy()
# If X_train is a NumPy array, convert it to a DataFrame
X_train_df = pd.DataFrame(X_train_original)
X_train_scaled_df = pd.DataFrame(X_train_scaled)

# 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()
```


Before Scaling
1e7
1.0
0.8
0.6
0.4
0.2
0.0
After Robust Scaling
120
100
80
60
40
20
0

In [ ]: