In [1]:

```
import warnings
warnings.filterwarnings('ignore')
import sys
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
import plotly.graph_objects as go
import plotly.express as px
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score,f1_score,recall_score,precision_score
from sklearn import tree
from sklearn.metrics import ConfusionMatrixDisplay
from sklearn.metrics import confusion_matrix,classification_report
from urllib.parse import urlparse
import re
import whois
import datetime
import requests
from wordcloud import WordCloud
```

In [2]:

```
# This is main refrences to extract features from URLs
#https://towardsdatascience.com/phishing-domain-detection-with-ml-5be9c99293e5
#https://arxiv.org/pdf/2205.05121.pdf
```

Data reading

- reading the CSV

In [3]:

```
df = pd.read_csv('malicious_phish.csv')
print('Shape of DataFrame:', df.shape)
print('Size of DataFrame:', df.size)
df_copy = df.copy()
```

```
Shape of DataFrame: (651191, 2)
Size of DataFrame: 1302382
```

- checking first five rows

In [4]:

```
df.head()
```

Out[4]:

	url	type
0	br-icloud.com.br	phishing
1	mp3raid.com/music/krizz_kaliko.html	benign
2	bopsecrets.org/rexroth/cr/1.htm	benign
3	http://www.garage-pirenne.be/index.php?option=...	defacement
4	http://adventure-nicaragua.net/index.php?optio...	defacement

- checking last five rows

In [5]:

```
df.tail()
```

Out[5]:

	url	type
651186	xbox360.ign.com/objects/850/850402.html	phishing
651187	games.teamxbox.com/xbox-360/1860/Dead-Space/	phishing
651188	www.gamespot.com/xbox360/action/deadspace/	phishing
651189	en.wikipedia.org/wiki/Dead_Space_(video_game)	phishing
651190	www.angelfire.com/goth/devilmaycrytonite/	phishing

concise summary of our dataset

In [6]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 651191 entries, 0 to 651190
Data columns (total 2 columns):
 #   Column  Non-Null Count   Dtype 
---  ------  --------------   ----- 
 0   url     651191 non-null  object
 1   type    651191 non-null  object
dtypes: object(2)
memory usage: 9.9+ MB
```

Describing the Data

In [7]:

```
df.describe(exclude='number').T
```

Out[7]:

	count	unique	top	freq
url	651191	641119	http://style.org.hc360.com/css/detail/mysite/s...	180
type	651191	4	benign	428103

Checking for null values

In [8]:

```
df.isna().sum()
```

Out[8]:

```
url     0
type    0
dtype: int64
```

Checking if there are duplicates

In [9]:

```
df.duplicated().sum()
```

Out[9]:

```
10066
```

In [10]:

```
print(df.shape)
df.drop_duplicates(inplace=True)
print(df.shape)
```

```
(651191, 2)
(641125, 2)
```

Data Sampling

In [11]:

```
malware_data = df[df['type']=='malware'].head(1000)
benign_data = df[df['type']=='benign'].sample(n=2500,random_state=391)
defacement_data = df[df['type']=='defacement'].head(1000)
phishing_data = df[df['type']=='phishing'].head(1000)
```

In [12]:

```
df = pd.concat([malware_data,benign_data,defacement_data,phishing_data]).reset_index()
```

In [13]:

```
df.drop('index',axis=1,inplace=True)
```

In [14]:

```
df.head()
```

Out[14]:

	url	type
0	http://www.824555.com/app/member/SportOption.p...	malware
1	http://9779.info/%E5%84%BF%E7%AB%A5%E7%AB%8B%E...	malware
2	http://9779.info/%E6%A0%91%E5%8F%B6%E7%B2%98%E...	malware
3	http://9779.info/%E5%8F%A4%E4%BB%A3%E4%BA%8C%E...	malware
4	http://chinacxyy.com/piccodejs-000.asp?lm2=191...	malware

Data ploting

In [15]:

```
df_copy.type.value_counts()
```

Out[15]:

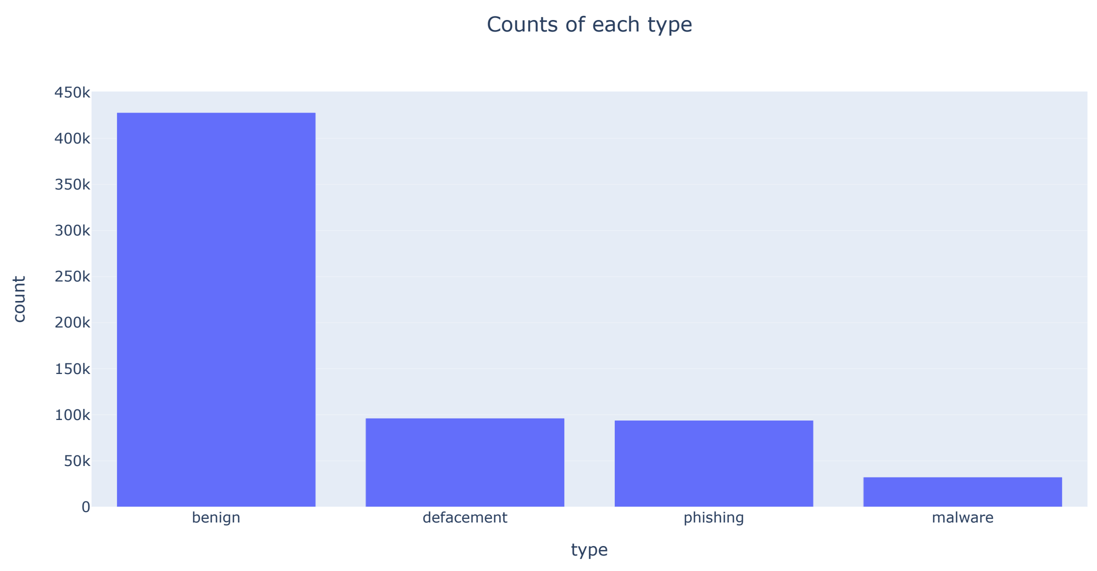
```
benign         428103
defacement      96457
phishing        94111
malware         32520
Name: type, dtype: int64
```

In [16]:

```
countTypes = pd.DataFrame({'type':['benign','defacement','phishing','malware'],
                      'count':[428103,96457,94111,32520]})
```

In [144]:

```
.bar(data_frame=countTypes, x='type',y='count',title='Counts of each type',color_discrete_map={'count':'#c47d7d'}).update_layou
()
```



insight

- benign is the most, and phishing and defacemet are close together.
- Malware is the least.

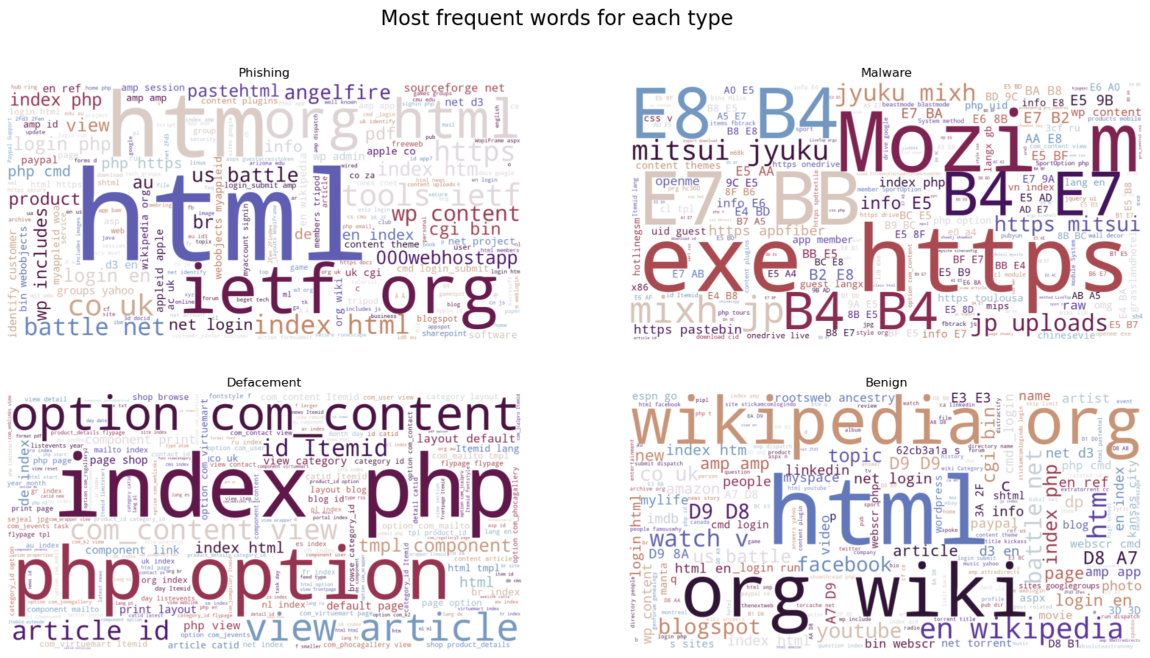
We found that the dataset was unblanced between the 4 types, so we balnced the data by taking sample of each type

In [18]:

```
df_phish = df_copy[df_copy.type=='phishing']
df_malware = df_copy[df_copy.type=='malware']
df_deface = df_copy[df_copy.type=='defacement']
df_benign = df_copy[df_copy.type=='benign']
```

In [19]:

```
plt.figure(figsize=[20,10])
plt.suptitle('Most frequent words for each type',fontsize = 20)
phish_url = " ".join(i for i in df_phish.url)
wordcloud1 = WordCloud(width=1600, height=800,colormap='twilight',background_color='white').generate(phish_url)
plt.subplot(2,2,1)
plt.title('Phishing')
plt.imshow(wordcloud1, interpolation='bilinear')
plt.axis('off')


malware_url = " ".join(i for i in df_malware.url)
wordcloud2 = WordCloud(width=1600, height=800,colormap='twilight',background_color='white').generate(malware_url)
plt.subplot(2,2,2)
plt.title('Malware')
plt.imshow(wordcloud2, interpolation='bilinear')
plt.axis('off')


deface_url = " ".join(i for i in df_deface.url)
wordcloud3 = WordCloud(width=1600,height=800,colormap='twilight',background_color='white').generate(deface_url)
plt.subplot(2,2,3)
plt.title('Defacement')
plt.imshow(wordcloud3, interpolation='bilinear')
plt.axis('off')


benign_url = " ".join(i for i in df_benign.url)
wordcloud4 = WordCloud(width=1600, height=800,colormap='twilight',background_color='white').generate(benign_url)
plt.subplot(2,2,4)
plt.title('Benign')
plt.imshow(wordcloud4, interpolation='bilinear')
plt.axis('off')


plt.show()
```



insight

- The most frequent words:
  - Phishing: https, org, html and tools.
  - Malware: Mozi, m, https and exe.
  - Defacement: index, php, option and com_content.
  - Benign: html , org , wiki and wikipedia.

Features Selection

In [20]:

```
def charCount(url, feature):
    return  url.count(feature)
```

In [21]:

```
feature = ['@','?','-','=','.','#','%','+','$','!','*',',','//']
for a in feature:
    df[a] = df['url'].apply(lambda i: charCount(i,a))
```

In [22]:

```
df
```

Out[22]:

	url	type	@	?	-	=	.	#	%	+	$	!	*	,	//
0	http://www.824555.com/app/member/SportOption.p...	malware	0	1	0	2	3	0	0	0	0	0	0	0	1
1	http://9779.info/%E5%84%BF%E7%AB%A5%E7%AB%8B%E...	malware	0	0	0	0	1	0	21	0	0	0	0	0	1
2	http://9779.info/%E6%A0%91%E5%8F%B6%E7%B2%98%E...	malware	0	0	0	0	1	0	15	0	0	0	0	0	1
3	http://9779.info/%E5%8F%A4%E4%BB%A3%E4%BA%8C%E...	malware	0	0	0	0	1	0	27	0	0	0	0	0	1
4	http://chinacxyy.com/piccodejs-000.asp?lm2=191...	malware	0	1	1	8	2	0	0	0	0	0	0	0	1
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
5495	wedrifastct.com	phishing	0	0	0	0	1	0	0	0	0	0	0	0	0
5496	paypal.com.it.webapps.mpp.home.holpbenk24.com	phishing	0	0	0	0	7	0	0	0	0	0	0	0	0
5497	delaraujo.com.br	phishing	0	0	0	0	2	0	0	0	0	0	0	0	0
5498	http://www.helderheidbokaal.nl//wp-content/plu...	phishing	0	0	3	0	3	0	0	0	0	0	0	0	2
5499	http://www.vighnahartainn.in/new/quote/	phishing	0	0	0	0	2	0	0	0	0	0	0	0	1

5500 rows × 15 columns

In [23]:

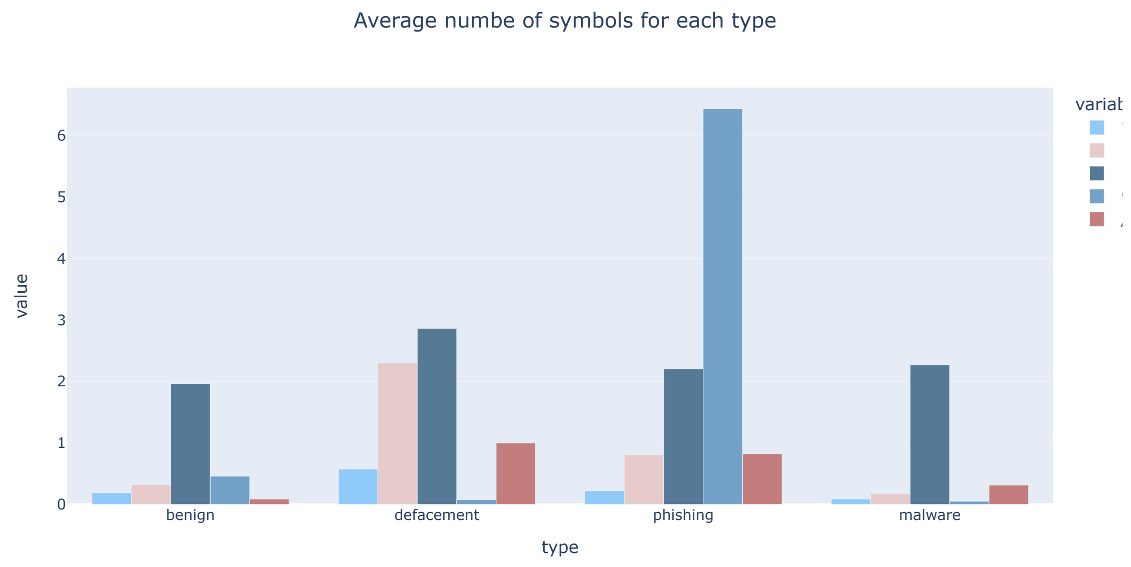
```
All = df.groupby('type').mean()
result = All[['@','?','-','=','.','#','%','+','$','!','*',',','//']]
result['type']=['benign','defacement','phishing','malware']
#"benign": 0, "defacement": 1, "phishing":2, "malware":3
result
```

Out[23]:

	@	?	-	=	.	#	%	+	$	!	*	,	//	type
type														
benign	0.0008	0.1904	1.722	0.3232	1.9652	0.0004	0.4584	0.098	0.0	0.000	0.0	0.000	0.088	benign
defacement	0.0000	0.5760	1.677	2.3010	2.8590	0.0000	0.0780	0.008	0.0	0.000	0.0	0.007	1.000	defacement
malware	0.0010	0.2230	1.087	0.8050	2.2060	0.0000	6.4360	0.063	0.0	0.004	0.0	0.000	0.826	phishing
phishing	0.0000	0.0870	0.541	0.1720	2.2700	0.0010	0.0490	0.000	0.0	0.000	0.0	0.000	0.314	malware

In [119]:

```
px.bar(data_frame=result,x='type',y=['?','=','.','%','//'],barmode='group',title='Average numbe of symbols for each type',colo

#color_discrete_map={'%':'#73a1c7','//':'#c47d7d'}
# blue,darkblue,royalblue,lightcyan
```



insights ¶

- Phishing URLs can have a lot of % symbol.

In [25]:

```
#https://dmitripavlutin.com/parse-url-javascript/
#https://docs.python.org/3/library/re.html
# re.search : Scan through string looking for the
# first location where the regular expression pattern produces a match,
# and return a corresponding match object. Return None if
# no position in the string matches the pattern; note that
# this is different from finding a zero-length match at some point in the string.
```

In [26]:

```
# check if the url has a hostname or not
def HasHostname(url):
    hostname = urlparse(url).hostname
    hostname = str(hostname)
    match = re.search(hostname, url)
    if match:
        return 1
    else:
        return 0
```

In [27]:

```
df['HasHostname'] = df['url'].apply(lambda i: HasHostname(i))
```

In [28]:

```
Host = pd.crosstab(df.type,df.HasHostname)
Host['type']=['benign','defacement','phishing','malware']
Host.rename(columns={0:'no_HostName',1:'has_HostName'},inplace=True)
```

In [29]:

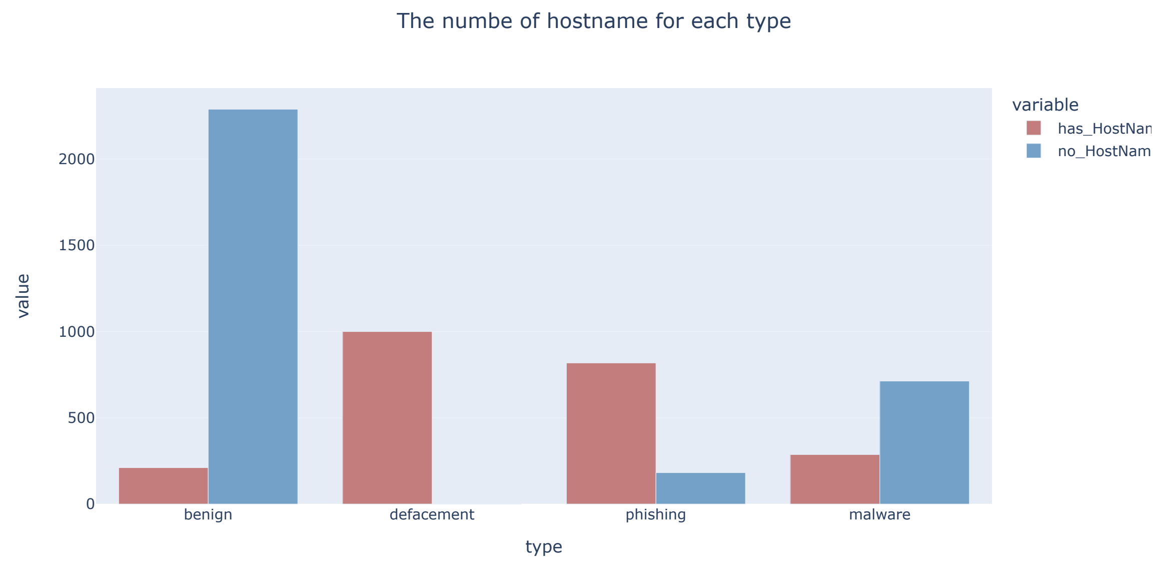
```
Host
```

Out[29]:

HasHostname	no_HostName	has_HostName	type
type			
benign	2289	211	benign
defacement	0	1000	defacement
malware	182	818	phishing
phishing	713	287	malware

In [121]:

```
px.bar(data_frame=Host,x=Host.type,y=['has_HostName','no_HostName'],barmode='group',title='The numbe of hostname for each type
```



insights

- Benign URL with no Hostname have the highest count.

In [31]:

```
#https://python.readthedocs.io/en/v2.7.2/library/urlparse.html
# scheme return either http or https or None
# IsHttps to check if the url is https
def IsHttps(url):
    htp = urlparse(url).scheme
    match = str(htp)
    if match=='https':
        return 1
    else:
        return 0
```

In [32]:

```
df['IsHttps'] = df['url'].apply(lambda i: IsHttps(i))
```

In [33]:
```
df
```
Out[33]:

	url	type	@	?	-	=	.	#	%	+	$	!	*	,	//	HasHostname	IsHttps
0	http://www.824555.com/app/member/SportOption.p...	malware	0	1	0	2	3	0	0	0	0	0	0	0	1	1	0
1	http://9779.info/%E5%84%BF%E7%AB%A5%E7%AB%8B%E...	malware	0	0	0	0	1	0	21	0	0	0	0	0	1	1	0
2	http://9779.info/%E6%A0%91%E5%8F%B6%E7%B2%98%E...	malware	0	0	0	0	1	0	15	0	0	0	0	0	1	1	0
3	http://9779.info/%E5%8F%A4%E4%BB%A3%E4%BA%8C%E...	malware	0	0	0	0	1	0	27	0	0	0	0	0	1	1	0
4	http://chinacxyy.com/piccodejs-000.asp?lm2=191...	malware	0	1	1	8	2	0	0	0	0	0	0	0	1	1	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
5495	wedrifastct.com	phishing	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
5496	paypal.com.it.webapps.mpp.home.holpbenk24.com	phishing	0	0	0	0	7	0	0	0	0	0	0	0	0	0	0
5497	delaraujo.com.br	phishing	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0
5498	http://www.helderheidbokaal.nl//wp-content/plu...	phishing	0	0	3	0	3	0	0	0	0	0	0	0	2	1	0
5499	http://www.vighnahartainn.in/new/quote/	phishing	0	0	0	0	2	0	0	0	0	0	0	0	1	1	0

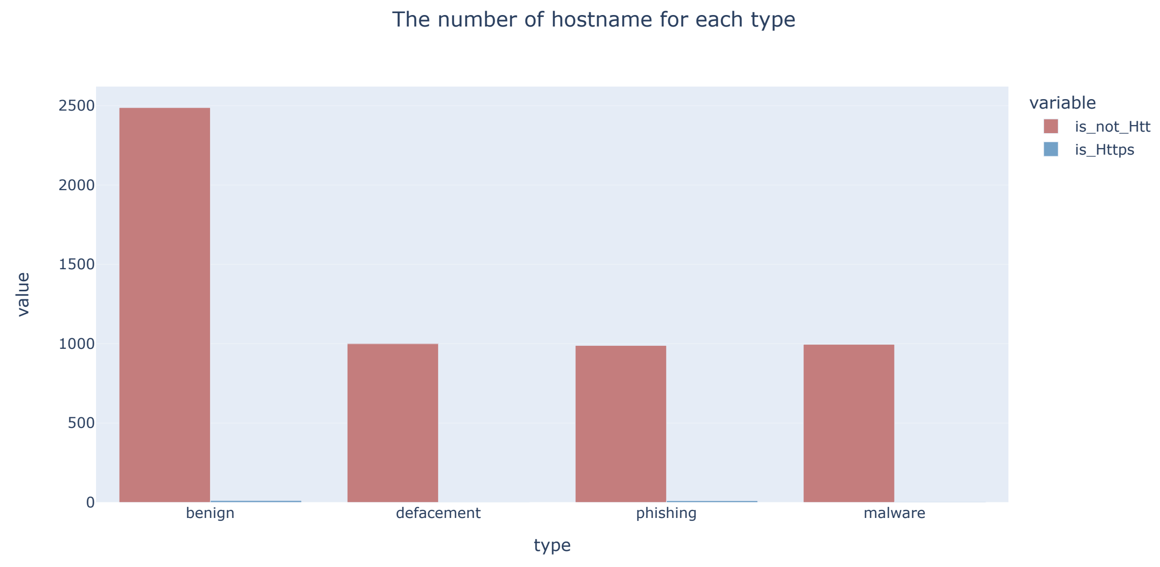
5500 rows × 17 columns

In [34]:
```
https = pd.crosstab(df.type,df.IsHttps)
https['type']=['benign','defacement','phishing','malware']
https.rename(columns={0:'is_not_Https',1:'is_Https'},inplace=True)
```

In [35]:
```
https
```
Out[35]:

IsHttps	**is_not_Https**	**is_Https**	**type**
type			
benign	2488	12	benign
defacement	1000	0	defacement
malware	989	11	phishing
phishing	996	4	malware

In [122]:
```
px.bar(data_frame=https,x=https.type,y=['is_not_Https','is_Https'],barmode='group',title='The number of hostname for each type
```



insights

- The protocol (https) can be used for phishing and malware

In [37]:

```
# Count the number of digits in url (how many numbers there?)
def numberCount(url):
    numbers = 0
    for i in url:
        if i.isnumeric():
            numbers = numbers + 1
    return numbers
```

In [38]:

```
df['numberCount']= df['url'].apply(lambda i: numberCount(i))
```

In [39]:

```
df
```

Out[39]:

	url	type	@	?	-	=	.	#	%	+	$	!	*	,	//	HasHostname	IsHttps	numberCo
0	http://www.824555.com/app/member/SportOption.p...	malware	0	1	0	2	3	0	0	0	0	0	0	0	1	1	0	
1	http://9779.info/%E5%84%BF%E7%AB%A5%E7%AB%8B%E...	malware	0	0	0	0	1	0	21	0	0	0	0	0	1	1	0	
2	http://9779.info/%E6%A0%91%E5%8F%B6%E7%B2%98%E...	malware	0	0	0	0	1	0	15	0	0	0	0	0	1	1	0	
3	http://9779.info/%E5%8F%A4%E4%BB%A3%E4%BA%8C%E...	malware	0	0	0	0	1	0	27	0	0	0	0	0	1	1	0	
4	http://chinacxyy.com/piccodejs-000.asp?lm2=191...	malware	0	1	1	8	2	0	0	0	0	0	0	0	1	1	0	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
5495	wedrifastct.com	phishing	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	
5496	paypal.com.it.webapps.mpp.home.holpbenk24.com	phishing	0	0	0	0	7	0	0	0	0	0	0	0	0	0	0	
5497	delaraujo.com.br	phishing	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	
5498	http://www.helderheidbokaal.nl//wp-content/plu...	phishing	0	0	3	0	3	0	0	0	0	0	0	0	2	1	0	
5499	http://www.vighnahartainn.in/new/quote/	phishing	0	0	0	0	2	0	0	0	0	0	0	0	1	1	0	

5500 rows × 18 columns

In [40]:

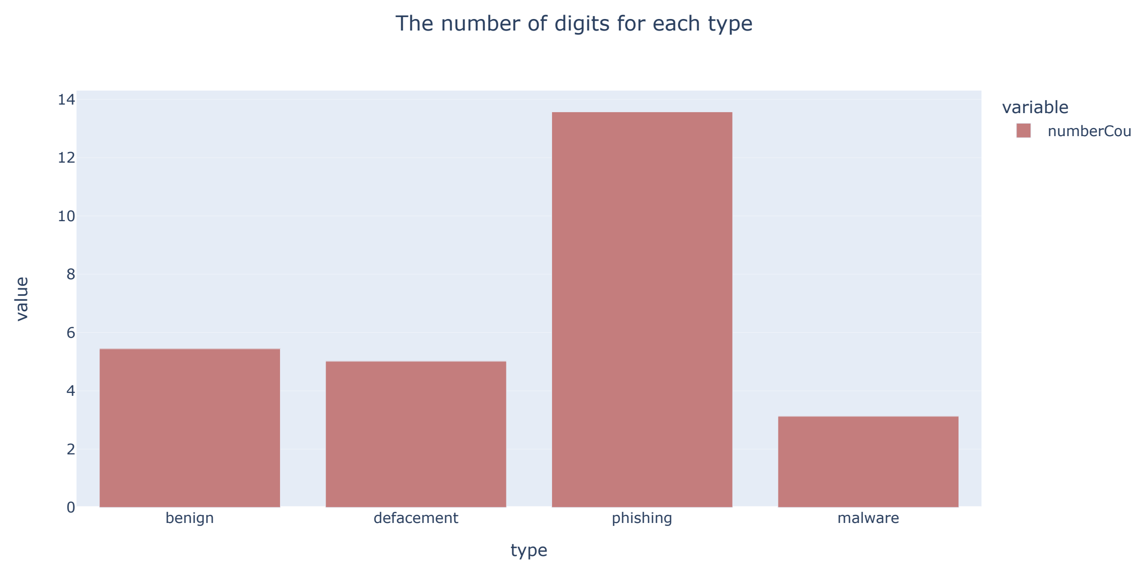
```
All_num = df[['type','numberCount']].groupby('type').mean()
number = pd.DataFrame()
number['numberCount'] = All_num['numberCount']
number['type']=['benign','defacement','phishing','malware']
#"benign": 0, "defacement": 1, "phishing":2, "malware":3
number
```

Out[40]:

	numberCount	type
type		
benign	5.4424	benign
defacement	5.0140	defacement
malware	13.5690	phishing
phishing	3.1270	malware

In [125]:

```
ar(data_frame=number,x=number.type,y=['numberCount'],barmode='group',title='The number of digits for each type',color_discrete_
```



insight

- The number of digits increas in Malware , Defacement and phishing
- The numbers that appear in benign may be due to the port number and username

In [42]:

```
# Count the number of alphabets in url (how many letter there?)
def alphabetCount(url):
    alphabets = 0
    for i in url:
        if i.isalpha():
            alphabets = alphabets + 1
    return alphabets
```

In [43]:

```
df['alphabetCount']= df['url'].apply(lambda i: alphabetCount(i))
```

In [44]:

```
df
```

Out[44]:

	url	type	@	?	-	=	.	#	%	+	$	!	*	,	//	HasHostname	IsHttps	numberCo
0	http://www.824555.com/app/member/SportOption.p...	malware	0	1	0	2	3	0	0	0	0	0	0	0	1	1	0	
1	http://9779.info/%E5%84%BF%E7%AB%A5%E7%AB%8B%E...	malware	0	0	0	0	1	0	21	0	0	0	0	0	1	1	0	
2	http://9779.info/%E6%A0%91%E5%8F%B6%E7%B2%98%E...	malware	0	0	0	0	1	0	15	0	0	0	0	0	1	1	0	
3	http://9779.info/%E5%8F%A4%E4%BB%A3%E4%BA%8C%E...	malware	0	0	0	0	1	0	27	0	0	0	0	0	1	1	0	
4	http://chinacxyy.com/piccodejs-000.asp?lm2=191...	malware	0	1	1	8	2	0	0	0	0	0	0	0	1	1	0	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
5495	wedrifastct.com	phishing	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	
5496	paypal.com.it.webapps.mpp.home.holpbenk24.com	phishing	0	0	0	0	7	0	0	0	0	0	0	0	0	0	0	
5497	delaraujo.com.br	phishing	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	
5498	http://www.helderheidbokaal.nl//wp-content/plu...	phishing	0	0	3	0	3	0	0	0	0	0	0	0	2	1	0	
5499	http://www.vighnahartainn.in/new/quote/	phishing	0	0	0	0	2	0	0	0	0	0	0	0	1	1	0	

5500 rows × 19 columns

In [45]:

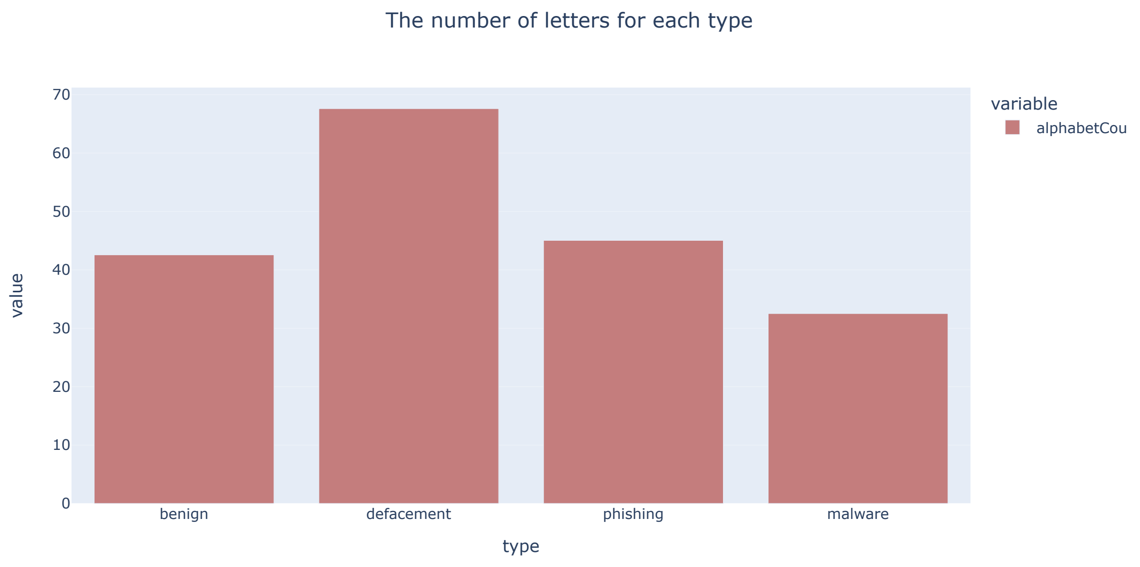
```
All_alph = df[['type','alphabetCount']].groupby('type').mean()
letters = pd.DataFrame()
letters['alphabetCount'] = All_alph['alphabetCount']
letters['type']=['benign','defacement','phishing','malware']

letters
```

Out[45]:

	alphabetCount	type
type		
benign	42.5712	benign
defacement	67.6090	defacement
malware	45.0460	phishing
phishing	32.5150	malware

In [124]:

```
px.bar(data_frame=letters,x=letters.type,y=['alphabetCount'],barmode='group',title='The number of letters for each type',color_
```



insight

- The possibility of the site to be malicious is greater if the number of characters is large

In [47]:

```
# Check if the url conatins the short url (tinyURL)

def shortUrl(url):
    match = re.search('bit\.ly|goo\.gl|shorte\.st|go2l\.ink|x\.co|ow\.ly|t\.co|tinyurl|tr\.im|is\.gd|cli\.gs|'
                      'yfrog\.com|migre\.me|ff\.im|tiny\.cc|url4\.eu|twit\.ac|su\.pr|twurl\.nl|snipurl\.com|'
                      'short\.to|BudURL\.com|ping\.fm|post\.ly|Just\.as|bkite\.com|snipr\.com|fic\.kr|loopt\.us|'
                      'doiop\.com|short\.ie|kl\.am|wp\.me|rubyurl\.com|om\.ly|to\.ly|bit\.do|t\.co|lnkd\.in|'
                      'db\.tt|qr\.ae|adf\.ly|goo\.gl|bitly\.com|cur\.lv|tinyurl\.com|ow\.ly|bit\.ly|ity\.im|'
                      'q\.gs|is\.gd|po\.st|bc\.vc|twitthis\.com|u\.to|j\.mp|buzurl\.com|cutt\.us|u\.bb|yourls\.org|'
                      'x\.co|prettylinkpro\.com|scrnch\.me|filoops\.info|vzturl\.com|qr\.net|1url\.com|tweez\.me|v\.gd|'
                      'tr\.im|link\.zip\.net',
                      url)
    if match:
        return 1
    else:
        return 0
```

In [48]:

```
df['shortUrl'] = df['url'].apply(lambda x: shortUrl(x))
```

In [49]:
```
df
```
Out[49]:

	url	type	@	?	-	=	.	#	%	+	$	!	*	,	//	HasHostname	IsHttps	numberC
0	http://www.824555.com/app/member/SportOption.p...	malware	0	1	0	2	3	0	0	0	0	0	0	0	1	1	0	
1	http://9779.info/%E5%84%BF%E7%AB%A5%E7%AB%8B%E...	malware	0	0	0	0	1	0	21	0	0	0	0	0	1	1	0	
2	http://9779.info/%E6%A0%91%E5%8F%B6%E7%B2%98%E...	malware	0	0	0	0	1	0	15	0	0	0	0	0	1	1	0	
3	http://9779.info/%E5%8F%A4%E4%BB%A3%E4%BA%8C%E...	malware	0	0	0	0	1	0	27	0	0	0	0	0	1	1	0	
4	http://chinacxyy.com/piccodejs-000.asp?lm2=191...	malware	0	1	1	8	2	0	0	0	0	0	0	0	1	1	0	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
5495	wedrifastct.com	phishing	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	
5496	paypal.com.it.webapps.mpp.home.holpbenk24.com	phishing	0	0	0	0	7	0	0	0	0	0	0	0	0	0	0	
5497	delaraujo.com.br	phishing	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	
5498	http://www.helderheidbokaal.nl//wp-content/plu...	phishing	0	0	3	0	3	0	0	0	0	0	0	0	2	1	0	
5499	http://www.vighnahartainn.in/new/quote/	phishing	0	0	0	0	2	0	0	0	0	0	0	0	1	1	0	

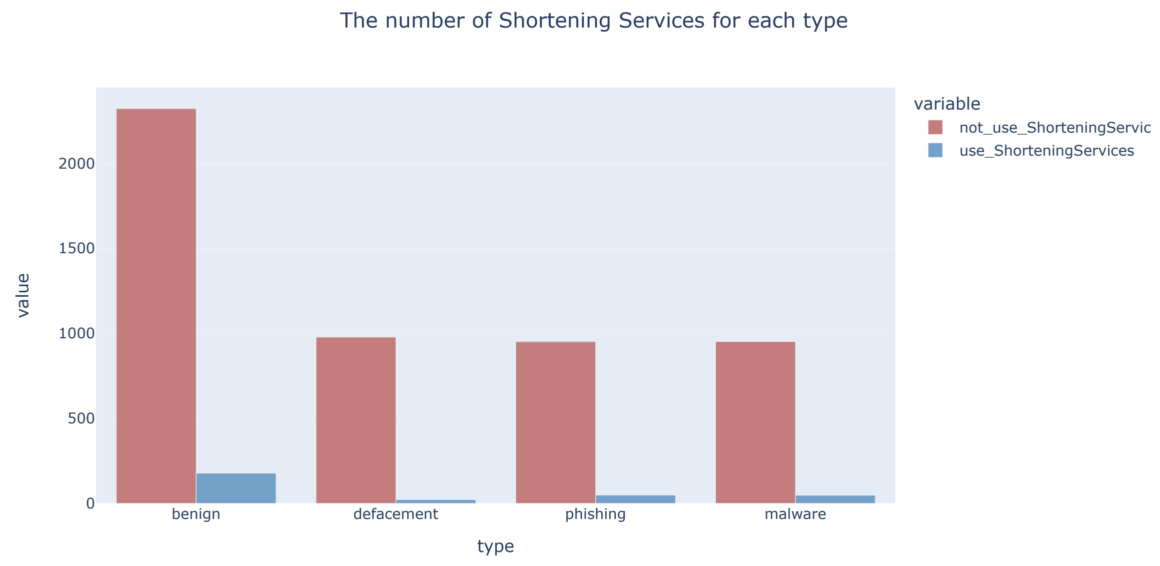
5500 rows × 20 columns

In [50]:
```
shortUrl = pd.crosstab(df.type,df.shortUrl)
shortUrl['type']=['benign','defacement','phishing','malware']
shortUrl.rename(columns={0:'not_use_ShorteningServices',1:'use_ShorteningServices'},inplace=True)
```

In [51]:
```
shortUrl
```
Out[51]:

shortUrl	not_use_ShorteningServices	use_ShorteningServices	type
type			
benign	2322	178	benign
defacement	978	22	defacement
malware	951	49	phishing
phishing	952	48	malware

In [127]:
```
px.bar(data_frame=shortUrl,x=shortUrl.type,y=['not_use_ShorteningServices','use_ShorteningServices'],barmode='group',title='The
```



insight

- Benign URLs does not use the (shorting URL).
- Not all URL shorting is malicious or dangerous.

In [53]:

```
# check if url contains IPv4 or IPv6
def ipAddress(url):
    match = re.search(
        '(([01]?\\d\\d?|2[0-4]\\d|25[0-5])\\.([01]?\\d\\d?|2[0-4]\\d|25[0-5])\\.([01]?\\d\\d?|2[0-4]\\d|25[0-5])\\.'
        '([01]?\\d\\d?|2[0-4]\\d|25[0-5])\\/)|'  # IPv4
        '(([01]?\\d\\d?|2[0-4]\\d|25[0-5])\\.([01]?\\d\\d?|2[0-4]\\d|25[0-5])\\.([01]?\\d\\d?|2[0-4]\\d|25[0-5])\\.'
        '([01]?\\d\\d?|2[0-4]\\d|25[0-5])\\/)|'  # IPv4 with port
        '((0x[0-9a-fA-F]{1,2})\\.(0x[0-9a-fA-F]{1,2})\\.(0x[0-9a-fA-F]{1,2})\\.(0x[0-9a-fA-F]{1,2})\\/)' # IPv4 in hexadecimal
        '(?:[a-fA-F0-9]{1,4}:){7}[a-fA-F0-9]{1,4}|'
        '([0-9]+(?:\.[0-9]+){3}:[0-9]+)|'
        '((?:(?:\d|[01]?\d\d|2[0-4]\d|25[0-5])\.){3}(?:25[0-5]|2[0-4]\d|[01]?\d\d|\d)(?:\/\d{1,2})?)', url)  # Ipv6
    if match:
        return 1
    else:
        return 0
```

In [54]:

```
df['ipAddress'] = df['url'].apply(lambda i: ipAddress(i))
```

In [55]:

```
df
```

Out[55]:

	url	type	@	?	-	=	.	#	%	+	...	!	*	,	//	HasHostname	IsHttps	numberCount	alphabetCo
0	http://www.824555.com/app/member/SportOption.p...	malware	0	1	0	2	3	0	0	0	...	0	0	0	1	1	0	6	
1	http://9779.info/%E5%84%BF%E7%AB%A5%E7%AB%8B%E...	malware	0	0	0	0	1	0	21	0	...	0	0	0	1	1	0	22	
2	http://9779.info/%E6%A0%91%E5%8F%B6%E7%B2%98%E...	malware	0	0	0	0	1	0	15	0	...	0	0	0	1	1	0	21	
3	http://9779.info/%E5%8F%A4%E4%BB%A3%E4%BA%8C%E...	malware	0	0	0	0	1	0	27	0	...	0	0	0	1	1	0	30	
4	http://chinacxyy.com/piccodejs-000.asp?lm2=191...	malware	0	1	1	8	2	0	0	0	...	0	0	0	1	1	0	17	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
5495	wedrifastct.com	phishing	0	0	0	0	1	0	0	0	...	0	0	0	0	0	0	0	
5496	paypal.com.it.webapps.mpp.home.holpbenk24.com	phishing	0	0	0	0	7	0	0	0	...	0	0	0	0	0	0	2	
5497	delaraujo.com.br	phishing	0	0	0	0	2	0	0	0	...	0	0	0	0	0	0	0	
5498	http://www.helderheidbokaal.nl//wp-content/plu...	phishing	0	0	3	0	3	0	0	0	...	0	0	0	2	1	0	1	

In [56]:

```
ipAddress = pd.crosstab(df.type,df.ipAddress)
ipAddress['type']=['benign','defacement','phishing','malware']
ipAddress.rename(columns={0:'not_use_ipAddress',1:'use_ipAddress'},inplace=True)
```

In [57]:

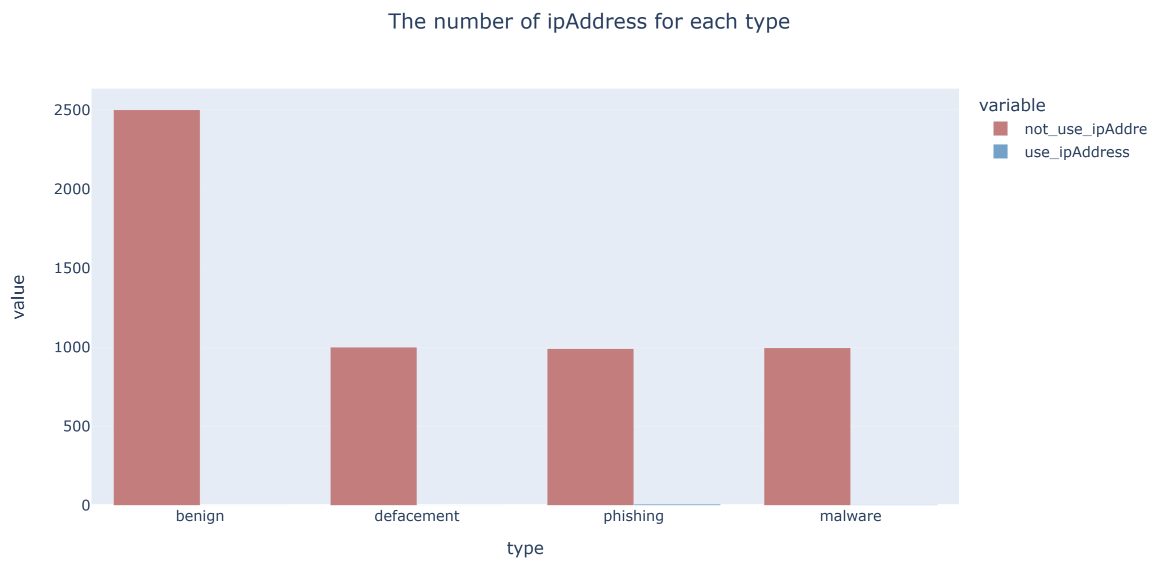
```
ipAddress
```

Out[57]:

ipAddress	**not_use_ipAddress**	**use_ipAddress**	**type**
type			
benign	2500	0	benign
defacement	1000	0	defacement
malware	992	8	phishing
phishing	996	4	malware

In [128]:

```
px.bar(data_frame=ipAddress,x=ipAddress.type,y=['not_use_ipAddress','use_ipAddress'],barmode='group',title='The number of ipAdd
```



insight

- The IP address does not appear in benign.

In [59]:

```
# https://pypi.org/project/python-whois/
# https://www.geeksforgeeks.org/how-to-convert-datetime-to-date-in-python/
'''
To get domain age:
1- get domain name using whois and urlparse
2- If url has domain name ==> extract the creation and expiration dates ==> check if the age is more than 12

if age > 12 less phishing possibility
else higher phishing possibility
'''
```

Out[59]:

```
'\nTo get domain age:\n1- get domain name using whois and urlparse\n2- If url has domain name ==> extract the cre
ation and expiration dates ==> check if the age is more than 12\n\nif age > 12 less phishing possibility  \nelse
higher phishing possibility \n'
```

In [60]:

```
def ageLess12Mon(url):
  try:
    domain_name = whois.whois(urlparse(url).netloc)
    creation_date = domain_name.creation_date
    expiration_date = domain_name.expiration_date
    if (isinstance(creation_date,str) or isinstance(expiration_date,str)):
      try:
        creation_date = datetime.strptime(creation_date,'%Y-%m-%d')
        expiration_date = datetime.strptime(expiration_date,"%Y-%m-%d")
      except:
        return 1
    if ((expiration_date is None) or (creation_date is None)):
        return 1
    else:
      ageofdomain = abs((expiration_date - creation_date).days)
      if ((ageofdomain/30) < 12):
        age = 1
      else:
        age = 0
  except:
      age = 1

  return age
```

In [61]:

```
df['ageLess12Mon'] = df['url'].apply(lambda i: ageLess12Mon(i))
```

In [62]:

```
df
```

Out[62]:

	url	type	@	?	-	=	.	#	%	+	...	*	,	//	HasHostname	IsHttps	numberCount
0	http://www.824555.com/app/member/SportOption.p...	malware	0	1	0	2	3	0	0	0	...	0	0	1	1	0	6
1	http://9779.info/%E5%84%BF%E7%AB%A5%E7%AB%8B%E...	malware	0	0	0	0	1	0	21	0	...	0	0	1	1	0	22
2	http://9779.info/%E6%A0%91%E5%8F%B6%E7%B2%98%E...	malware	0	0	0	0	1	0	15	0	...	0	0	1	1	0	21
3	http://9779.info/%E5%8F%A4%E4%BB%A3%E4%BA%8C%E...	malware	0	0	0	0	1	0	27	0	...	0	0	1	1	0	30
4	http://chinacxyy.com/piccodejs-000.asp?lm2=191...	malware	0	1	1	8	2	0	0	0	...	0	0	1	1	0	17
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
5495	wedrifastct.com	phishing	0	0	0	0	1	0	0	0	...	0	0	0	0	0	0
5496	paypal.com.it.webapps.mpp.home.holpbenk24.com	phishing	0	0	0	0	7	0	0	0	...	0	0	0	0	0	2
5497	delaraujo.com.br	phishing	0	0	0	0	2	0	0	0	...	0	0	0	0	0	0
5498	http://www.helderheidbokaal.nl//wp-content/plu...	phishing	0	0	3	0	3	0	0	0	...	0	0	2	1	0	1
5499	http://www.vighnahartainn.in/new/quote/	phishing	0	0	0	0	2	0	0	0	...	0	0	1	1	0	0

5500 rows × 22 columns

In [63]:

```
ageLess12Mon = pd.crosstab(df.type,df.ageLess12Mon)
ageLess12Mon['type']=['benign','defacement','phishing','malware']
ageLess12Mon.rename(columns={0:'ageMore12Mon',1:'ageLess12Mon'},inplace=True)
```

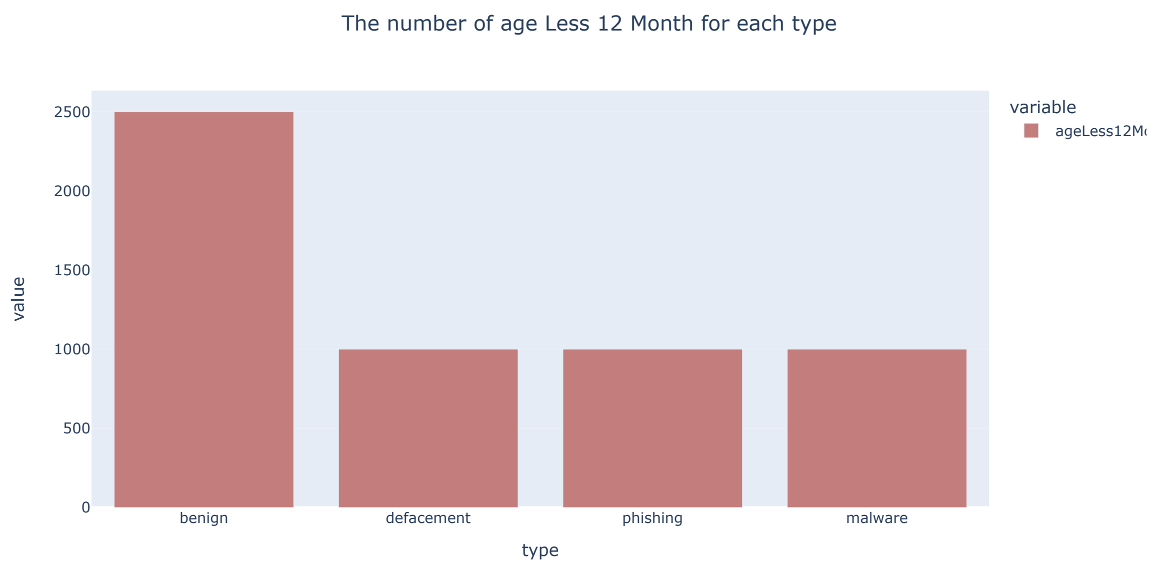
In [64]:

```
ageLess12Mon
```

Out[64]:

ageLess12Mon	ageLess12Mon	type
type		
benign	2500	benign
defacement	1000	defacement
malware	1000	phishing
phishing	1000	malware

In [129]:
```
px.bar(data_frame=ageLess12Mon,x=ageLess12Mon.type,y=['ageLess12Mon'],barmode='group',title='The number of age Less 12 Month fo
```



In [66]:
```
'''
To get domain end:
1- get domain name using whois and urlparse
2- If url has domain name ==> extract the expiration date ==> check if the end is less than 6

if end > 6 less phishing possibility
else higher phishing possibility
'''
```

Out[66]:
```
'\nTo get domain end:\n1- get domain name using whois and urlparse\n2- If url has domain name ==> extract the exp
iration date ==> check if the end is less than 6\n\nif end > 6 less phishing possibility  \nelse higher phishing
possibility \n'
```

In [67]:
```
def endLess6Mon(url):
  try:
    domain_name = whois.whois(urlparse(url).netloc)
    expiration_date = domain_name.expiration_date
    if isinstance(expiration_date,str):
      try:
        expiration_date = datetime.strptime(expiration_date,"%Y-%m-%d")
      except:
        return 1
    if (expiration_date is None):
        return 1
    else:
      today = datetime.now()
      end = abs((expiration_date - today).days)
      if ((end/30) < 6):
        end = 1
      else:
        end = 0
  except:
    end = 1

  return end
```

In [68]:
```
df['endLess6Mon'] = df['url'].apply(lambda i: endLess6Mon(i))
```

In [69]:
```
df
```
Out[69]:

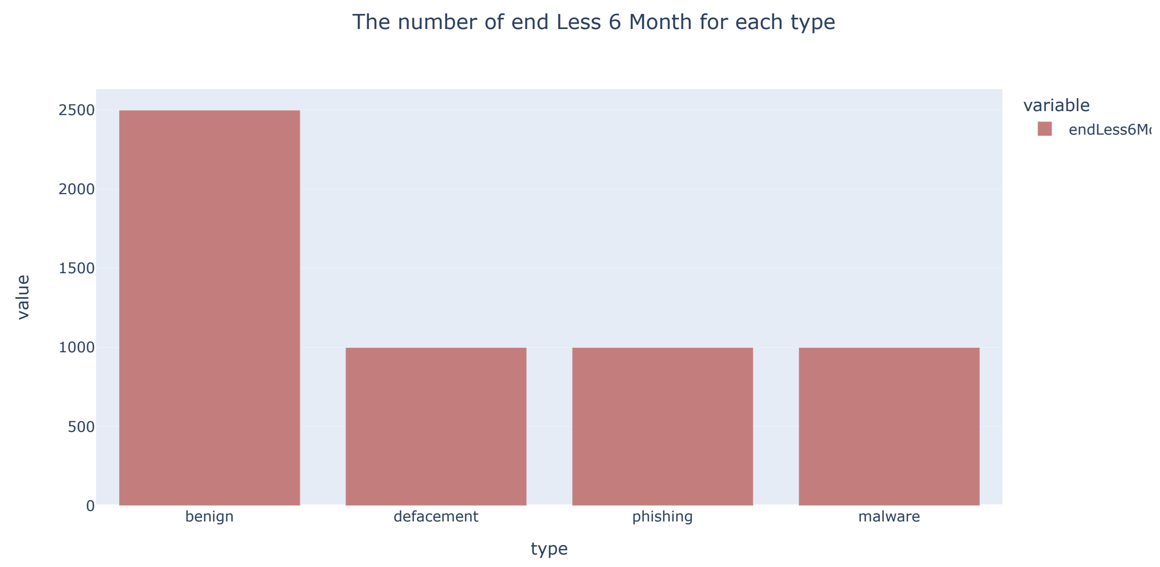
	url	type	@	?	-	=	.	#	%	+	...	,	//	HasHostname	IsHttps	numberCount	al
0	http://www.824555.com/app/member/SportOption.p...	malware	0	1	0	2	3	0	0	0	...	0	1	1	0	6	
1	http://9779.info/%E5%84%BF%E7%AB%A5%E7%AB%8B%E...	malware	0	0	0	0	1	0	21	0	...	0	1	1	0	22	
2	http://9779.info/%E6%A0%91%E5%8F%B6%E7%B2%98%E...	malware	0	0	0	0	1	0	15	0	...	0	1	1	0	21	
3	http://9779.info/%E5%8F%A4%E4%BB%A3%E4%BA%8C%E...	malware	0	0	0	0	1	0	27	0	...	0	1	1	0	30	
4	http://chinacxyy.com/piccodejs-000.asp?lm2=191...	malware	0	1	1	8	2	0	0	0	...	0	1	1	0	17	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
5495	wedrifastct.com	phishing	0	0	0	0	1	0	0	0	...	0	0	0	0	0	
5496	paypal.com.it.webapps.mpp.home.holpbenk24.com	phishing	0	0	0	0	7	0	0	0	...	0	0	0	0	2	
5497	delaraujo.com.br	phishing	0	0	0	0	2	0	0	0	...	0	0	0	0	0	
5498	http://www.helderheidbokaal.nl//wp-content/plu...	phishing	0	0	3	0	3	0	0	0	...	0	2	1	0	1	
5499	http://www.vighnahartainn.in/new/quote/	phishing	0	0	0	0	2	0	0	0	...	0	1	1	0	0	

5500 rows × 23 columns

In [70]:
```
endLess6Mon = pd.crosstab(df.type,df.endLess6Mon)
endLess6Mon['type']=['benign','defacement','phishing','malware']
endLess6Mon.rename(columns={0:'ageMore6Mon',1:'endLess6Mon'},inplace=True)
endLess6Mon
```
Out[70]:

endLess6Mon	endLess6Mon	type
type		
benign	2500	benign
defacement	1000	defacement
malware	1000	phishing
phishing	1000	malware

In [130]:
```
px.bar(data_frame=endLess6Mon,x=endLess6Mon.type,y=['endLess6Mon'],barmode='group',title='The number of end Less 6 Month for ea
```



In [72]:

```
# phishing sites use iframe tags to create invisible links that users maybe click it
def hasIfram(url):
    try:
        response = requests.get(url)
        if re.findall(r"[<iframe>|<frameBorder>]", response.text):
            return 0
        else:
            return 1
    except:
        return 1
```

In [73]:

```
df['hasIfram'] = df['url'].apply(lambda i: hasIfram(i))
```

In [74]:

```
df
```

Out[74]:

	url	type	@	?	-	=	.	#	%	+	...	//	HasHostname	IsHttps	numberCount	alpha
0	http://www.824555.com/app/member/SportOption.p...	malware	0	1	0	2	3	0	0	0	...	1	1	0	6	
1	http://9779.info/%E5%84%BF%E7%AB%A5%E7%AB%8B%E...	malware	0	0	0	0	1	0	21	0	...	1	1	0	22	
2	http://9779.info/%E6%A0%91%E5%8F%B6%E7%B2%98%E...	malware	0	0	0	0	1	0	15	0	...	1	1	0	21	
3	http://9779.info/%E5%8F%A4%E4%BB%A3%E4%BA%8C%E...	malware	0	0	0	0	1	0	27	0	...	1	1	0	30	
4	http://chinacxyy.com/piccodejs-000.asp?lm2=191...	malware	0	1	1	8	2	0	0	0	...	1	1	0	17	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
5495	wedrifastct.com	phishing	0	0	0	0	1	0	0	0	...	0	0	0	0	
5496	paypal.com.it.webapps.mpp.home.holpbenk24.com	phishing	0	0	0	0	7	0	0	0	...	0	0	0	2	
5497	delaraujo.com.br	phishing	0	0	0	0	2	0	0	0	...	0	0	0	0	
5498	http://www.helderheidbokaal.nl//wp-content/plu...	phishing	0	0	3	0	3	0	0	0	...	2	1	0	1	
5499	http://www.vighnahartainn.in/new/quote/	phishing	0	0	0	0	2	0	0	0	...	1	1	0	0	

5500 rows × 24 columns

In [75]:

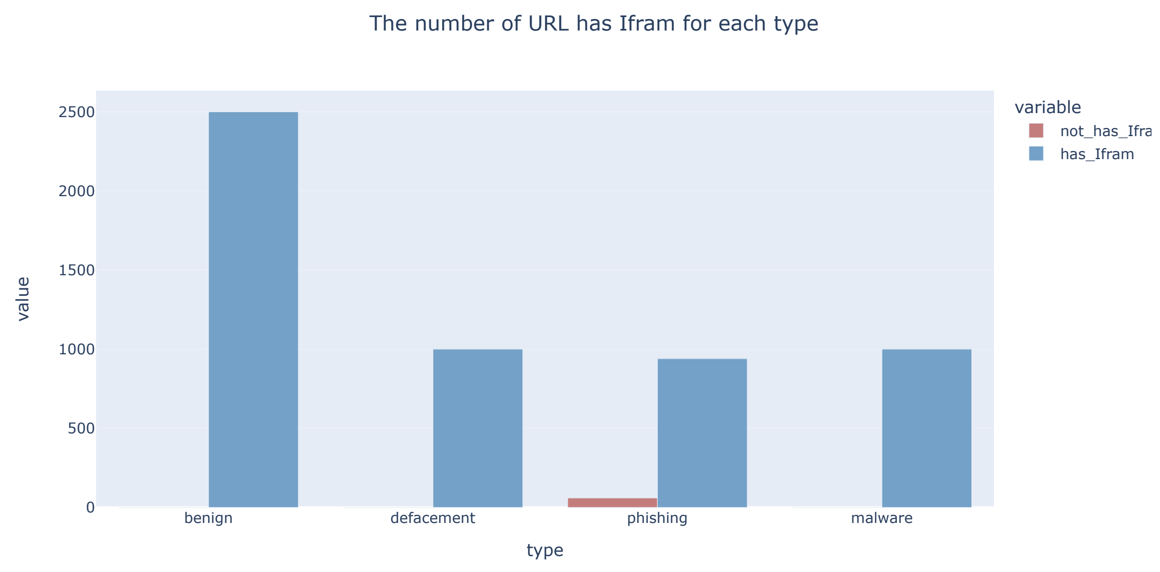
```
hasIfram = pd.crosstab(df.type,df.hasIfram)
hasIfram['type']=['benign','defacement','phishing','malware']
hasIfram.rename(columns={0:'not_has_Ifram',1:'has_Ifram'},inplace=True)
hasIfram
```

Out[75]:

hasIfram	not_has_Ifram	has_Ifram	type
type			
benign	0	2500	benign
defacement	0	1000	defacement
malware	60	940	phishing
phishing	0	1000	malware

In [131]:

```
px.bar(data_frame=hasIfram,x=hasIfram.type,y=['not_has_Ifram','has_Ifram'],barmode='group',title='The number of URL has Ifram i
```



insight

- Benign type must have ifram.

In [77]:

```
# https://www.google.com/url?sa=i&url=https%3A%2F%2Fsecurity.stackexchange.com%2Fquestions%2F41527%2Fis-the-web-browser-status-
# phishing sites use mouseover event from javascript to hide fake url
def hasMouseOver(url):
  try:
    response = requests.get(url)
    if re.findall("<script>.+onmouseover.+</script>", response.text):
      return 1
    else:
      return 0
  except:
        return 1
```

In [78]:

```
df['hasMouseOver'] = df['url'].apply(lambda i: hasMouseOver(i))
```

In [79]:

```
df
```

Out[79]:

	url	type	@	?	-	=	.	#	%	+	...	HasHostname	IsHttps	numberCount	alphabet(
0	http://www.824555.com/app/member/SportOption.p...	malware	0	1	0	2	3	0	0	0	...	1	0	6	
1	http://9779.info/%E5%84%BF%E7%AB%A5%E7%AB%8B%E...	malware	0	0	0	0	1	0	21	0	...	1	0	22	
2	http://9779.info/%E6%A0%91%E5%8F%B6%E7%B2%98%E...	malware	0	0	0	0	1	0	15	0	...	1	0	21	
3	http://9779.info/%E5%8F%A4%E4%BB%A3%E4%BA%8C%E...	malware	0	0	0	0	1	0	27	0	...	1	0	30	
4	http://chinacxyy.com/piccodejs-000.asp?lm2=191...	malware	0	1	1	8	2	0	0	0	...	1	0	17	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
5495	wedrifastct.com	phishing	0	0	0	0	1	0	0	0	...	0	0	0	
5496	paypal.com.it.webapps.mpp.home.holpbenk24.com	phishing	0	0	0	0	7	0	0	0	...	0	0	2	
5497	delaraujo.com.br	phishing	0	0	0	0	2	0	0	0	...	0	0	0	
5498	http://www.helderheidbokaal.nl//wp-content/plu...	phishing	0	0	3	0	3	0	0	0	...	1	0	1	
5499	http://www.vighnahartainn.in/new/quote/	phishing	0	0	0	0	2	0	0	0	...	1	0	0	

5500 rows × 25 columns

In [80]:

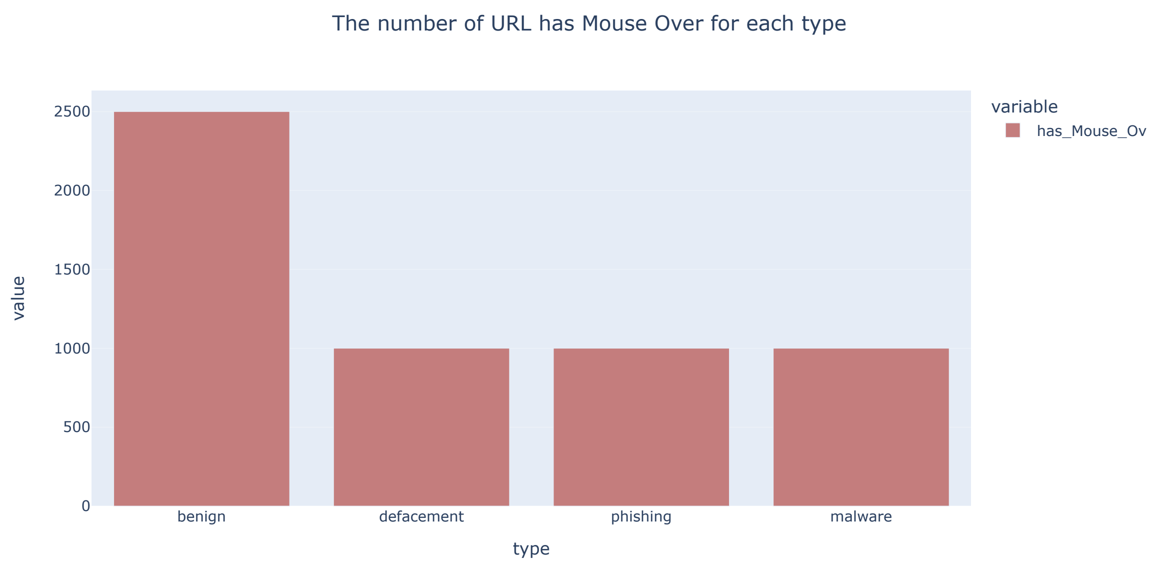
```
hasMouseOver = pd.crosstab(df.type,df.hasMouseOver)
hasMouseOver['type']=['benign','defacement','phishing','malware']
hasMouseOver.rename(columns={0:'not_has_Mouse_Over',1:'has_Mouse_Over'},inplace=True)
hasMouseOver
```

Out[80]:

hasMouseOver	has_Mouse_Over	type
type		
benign	2500	benign
defacement	1000	defacement
malware	1000	phishing
phishing	1000	malware

In [132]:

```
le='The number of URL has Mouse Over for each type',color_discrete_map={'has_Mouse_Over':'#c47d7d'}).update_layout(title_x=0.5)
```



In [82]:

```
# This part explained in the paper, disabled the right click option so the user cann't incpect the webpage
def disabledRightClick(url):
  try:
    response = requests.get(url)
    if re.findall(r"event.button ?== ?2", response.text):
      return 0
    else:
      return 1
  except:
        return 1
```

In [83]:

```
df['disabledRightClick'] = df['url'].apply(lambda i: disabledRightClick(i))
```

In [84]:

```
df
```

Out[84]:

	url	type	@	?	-	=	.	#	%	+	...	IsHttps	numberCount	alphabetCount	shortUrl
0	http://www.824555.com/app/member/SportOption.p...	malware	0	1	0	2	3	0	0	0	...	0	6	48	0
1	http://9779.info/%E5%84%BF%E7%AB%A5%E7%AB%8B%E...	malware	0	0	0	0	1	0	21	0	...	0	22	32	0
2	http://9779.info/%E6%A0%91%E5%8F%B6%E7%B2%98%E...	malware	0	0	0	0	1	0	15	0	...	0	21	21	0
3	http://9779.info/%E5%8F%A4%E4%BB%A3%E4%BA%8C%E...	malware	0	0	0	0	1	0	27	0	...	0	30	36	0
4	http://chinacxyy.com/piccodejs-000.asp?lm2=191...	malware	0	1	1	8	2	0	0	0	...	0	17	41	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
5495	wedrifastct.com	phishing	0	0	0	0	1	0	0	0	...	0	0	14	1
5496	paypal.com.it.webapps.mpp.home.holpbenk24.com	phishing	0	0	0	0	7	0	0	0	...	0	2	36	0
5497	delaraujo.com.br	phishing	0	0	0	0	2	0	0	0	...	0	0	14	0
5498	http://www.helderheidbokaal.nl//wp-content/plu...	phishing	0	0	3	0	3	0	0	0	...	0	1	66	0
5499	http://www.vighnahartainn.in/new/quote/	phishing	0	0	0	0	2	0	0	0	...	0	0	31	0

5500 rows × 26 columns

In [85]:

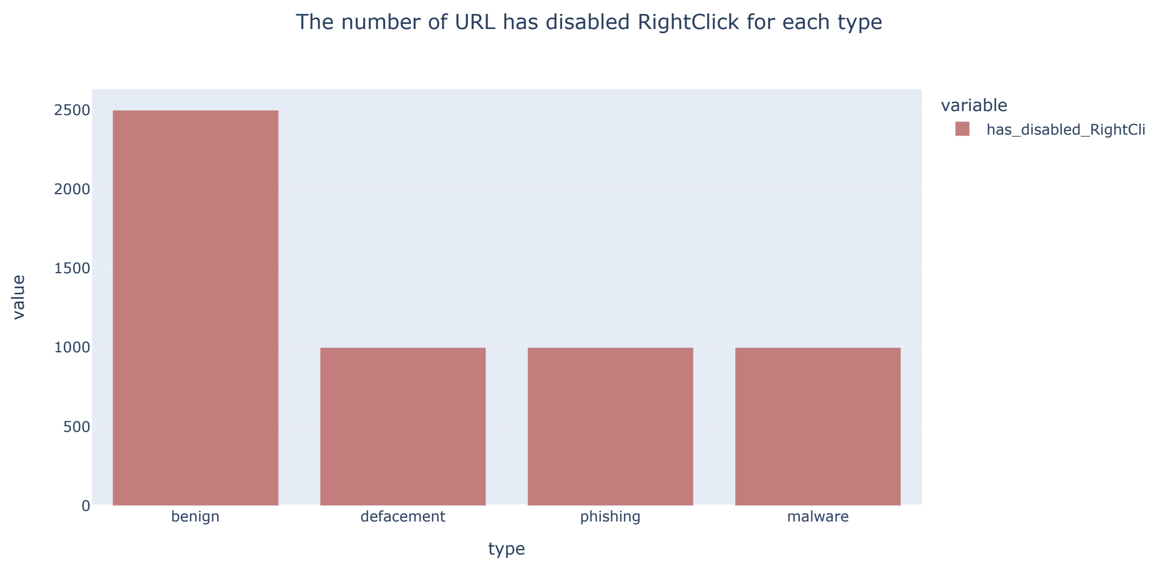
```
disabledRightClick = pd.crosstab(df.type,df.disabledRightClick)
disabledRightClick['type']=['benign','defacement','phishing','malware']
disabledRightClick.rename(columns={0:'not_has_disabled_RightClick',1:'has_disabled_RightClick'},inplace=True)
disabledRightClick
```

Out[85]:

disabledRightClick	has_disabled_RightClick	type
type		
benign	2500	benign
defacement	1000	defacement
malware	1000	phishing
phishing	1000	malware

In [133]:

```
URL has disabled RightClick for each type',color_discrete_map={'has_disabled_RightClick':'#c47d7d'}).update_layout(title_x=0.5)
```



In [87]:

```
# This part explained in the paper, multiple redirect webpages have a high possibility to be a phising websits
def isMultiDirected(url):
  try:
    response = requests.get(url)
    if len(response.history) <= 2:
      return 0
    else:
      return 1
  except:
        return 1
```

In [88]:

```
df['isMultiDirected'] = df['url'].apply(lambda i: isMultiDirected(i))
```

In [89]:

```
df
```

Out[89]:

	url	type	@	?	-	=	.	#	%	+	...	numberCount	alphabetCount	shortUrl	ipAddre
0	http://www.824555.com/app/member/SportOption.p...	malware	0	1	0	2	3	0	0	0	...	6	48	0	
1	http://9779.info/%E5%84%BF%E7%AB%A5%E7%AB%8B%E...	malware	0	0	0	0	1	0	21	0	...	22	32	0	
2	http://9779.info/%E6%A0%91%E5%8F%B6%E7%B2%98%E...	malware	0	0	0	0	1	0	15	0	...	21	21	0	
3	http://9779.info/%E5%8F%A4%E4%BB%A3%E4%BA%8C%E...	malware	0	0	0	0	1	0	27	0	...	30	36	0	
4	http://chinacxyy.com/piccodejs-000.asp?lm2=191...	malware	0	1	1	8	2	0	0	0	...	17	41	0	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
5495	wedrifastct.com	phishing	0	0	0	0	1	0	0	0	...	0	14	1	
5496	paypal.com.it.webapps.mpp.home.holpbenk24.com	phishing	0	0	0	0	7	0	0	0	...	2	36	0	
5497	delaraujo.com.br	phishing	0	0	0	0	2	0	0	0	...	0	14	0	
5498	http://www.helderheidbokaal.nl//wp-content/plu...	phishing	0	0	3	0	3	0	0	0	...	1	66	0	
5499	http://www.vighnahartainn.in/new/quote/	phishing	0	0	0	0	2	0	0	0	...	0	31	0	

5500 rows × 27 columns

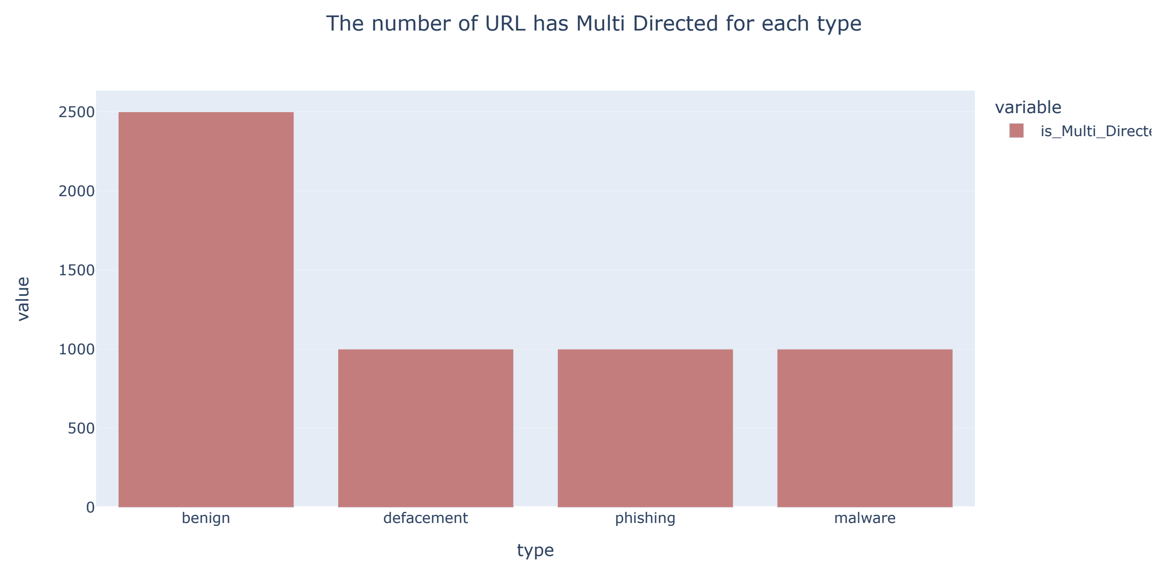
In [90]:

```
isMultiDirected = pd.crosstab(df.type,df.isMultiDirected)
isMultiDirected['type']=['benign','defacement','phishing','malware']
isMultiDirected.rename(columns={0:'is_not_Multi_Directed',1:'is_Multi_Directed'},inplace=True)
isMultiDirected
```

Out[90]:

isMultiDirected	**is_Multi_Directed**	**type**
type		
benign	2500	benign
defacement	1000	defacement
malware	1000	phishing
phishing	1000	malware

In [134]:
```
number of URL has Multi Directed for each type',color_discrete_map={'is_Multi_Directed':'#c47d7d'}).update_layout(title_x=0.5)
```



Models

In [105]:
```
modelData = df.copy()
```

In [106]:
```
modelData
```

Out[106]:

	url	type	@	?	-	=	.	#	%	+	...	numberCount	alphabetCount	shortUrl	ipAddress
0	http://www.824555.com/app/member/SportOption.p...	3	0	1	0	2	3	0	0	0	...	6	48	0	0
1	http://9779.info/%E5%84%BF%E7%AB%A5%E7%AB%8B%E...	3	0	0	0	0	1	0	21	0	...	22	32	0	0
2	http://9779.info/%E6%A0%91%E5%8F%B6%E7%B2%98%E...	3	0	0	0	0	1	0	15	0	...	21	21	0	0
3	http://9779.info/%E5%8F%A4%E4%BB%A3%E4%BA%8C%E...	3	0	0	0	0	1	0	27	0	...	30	36	0	0
4	http://chinacxyy.com/piccodejs-000.asp?lm2=191...	3	0	1	1	8	2	0	0	0	...	17	41	0	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
5495	wedrifastct.com	2	0	0	0	0	1	0	0	0	...	0	14	1	0
5496	paypal.com.it.webapps.mpp.home.holpbenk24.com	2	0	0	0	0	7	0	0	0	...	2	36	0	0
5497	delaraujo.com.br	2	0	0	0	0	2	0	0	0	...	0	14	0	0
5498	http://www.helderheidbokaal.nl//wp-content/plu...	2	0	0	3	0	3	0	0	0	...	1	66	0	0
5499	http://www.vighnahartainn.in/new/quote/	2	0	0	0	0	2	0	0	0	...	0	31	0	0

5500 rows × 27 columns

In [94]:
```
df['type'] = df['type'] .map({"benign": 0, "defacement": 1, "phishing":2, "malware":3})
```

In [95]:
```
df.head()
```
Out[95]:

	url	type	@	?	-	=	.	#	%	+	...	numberCount	alphabetCount	shortUrl	ipAddress	ageLe
0	http://www.824555.com/app/member/SportOption.p...	3	0	1	0	2	3	0	0	0	...	6	48	0	0	
1	http://9779.info/%E5%84%BF%E7%AB%A5%E7%AB%8B%E...	3	0	0	0	0	1	0	21	0	...	22	32	0	0	
2	http://9779.info/%E6%A0%91%E5%8F%B6%E7%B2%98%E...	3	0	0	0	0	1	0	15	0	...	21	21	0	0	
3	http://9779.info/%E5%8F%A4%E4%BB%A3%E4%BA%8C%E...	3	0	0	0	0	1	0	27	0	...	30	36	0	0	
4	http://chinacxyy.com/piccodejs-000.asp?lm2=191...	3	0	1	1	8	2	0	0	0	...	17	41	0	0	

5 rows × 27 columns

In [96]:
```
#df.to_csv('finalURL_version3.csv')
```

In [97]:
```
#['ageLess12Mon','endLess6Mon','hasMouseOver','disabledRightClick','isMultiDirected']
```

In [107]:
```
modelData['type'] = modelData['type'].map({0: 0, 1: 1, 2:1, 3:1})
```

In [108]:
```
modelData['type']
```
Out[108]:
```
0       1
1       1
2       1
3       1
4       1
       ..
5495    1
5496    1
5497    1
5498    1
5499    1
Name: type, Length: 5500, dtype: int64
```

In [109]:
```
X = modelData.drop(columns=['url','type'],axis=1)
y = modelData['type']
```

In [110]:
```
X.head()
```
Out[110]:

	@	?	-	=	.	#	%	+	$	!	...	numberCount	alphabetCount	shortUrl	ipAddress	ageLess12Mon	endLess6Mon	hasIfram	hasMouseOver	disabledI
0	0	1	0	2	3	0	0	0	0	0	...	6	48	0	0	1	1	0	1	
1	0	0	0	0	1	0	21	0	0	0	...	22	32	0	0	1	1	0	1	
2	0	0	0	0	1	0	15	0	0	0	...	21	21	0	0	1	1	0	1	
3	0	0	0	0	1	0	27	0	0	0	...	30	36	0	0	1	1	0	1	
4	0	1	1	8	2	0	0	0	0	0	...	17	41	0	0	1	1	0	1	

5 rows × 25 columns

In [ ]:
```

```

In [111]:
```
y.head()
```
Out[111]:
```
0    1
1    1
2    1
3    1
4    1
Name: type, dtype: int64
```

In [112]:

```
X.describe().T
```

Out[112]:

	count	mean	std	min	25%	50%	75%	max
@	5500.0	0.000545	0.023351	0.0	0.0	0.0	0.0	1.0
?	5500.0	0.247636	0.457847	0.0	0.0	0.0	0.0	4.0
-	5500.0	1.383636	2.651228	0.0	0.0	0.0	1.0	21.0
=	5500.0	0.742909	1.755488	0.0	0.0	0.0	0.0	19.0
.	5500.0	2.226909	1.590960	1.0	1.0	2.0	3.0	19.0
#	5500.0	0.000364	0.019068	0.0	0.0	0.0	0.0	1.0
%	5500.0	1.401636	6.169748	0.0	0.0	0.0	0.0	98.0
+	5500.0	0.057455	0.556312	0.0	0.0	0.0	0.0	21.0
$	5500.0	0.000000	0.000000	0.0	0.0	0.0	0.0	0.0
!	5500.0	0.000727	0.038135	0.0	0.0	0.0	0.0	2.0
*	5500.0	0.000000	0.000000	0.0	0.0	0.0	0.0	0.0
,	5500.0	0.001273	0.082018	0.0	0.0	0.0	0.0	6.0
//	5500.0	0.429091	0.510544	0.0	0.0	0.0	1.0	3.0
HasHostname	5500.0	0.421091	0.493779	0.0	0.0	0.0	1.0	1.0
IsHttps	5500.0	0.004909	0.069899	0.0	0.0	0.0	0.0	1.0
numberCount	5500.0	6.421091	11.339644	0.0	0.0	2.0	7.0	122.0
alphabetCount	5500.0	45.745091	30.935174	2.0	25.0	38.0	60.0	509.0
shortUrl	5500.0	0.054000	0.226038	0.0	0.0	0.0	0.0	1.0
ipAddress	5500.0	0.002182	0.046663	0.0	0.0	0.0	0.0	1.0
ageLess12Mon	5500.0	1.000000	0.000000	1.0	1.0	1.0	1.0	1.0
endLess6Mon	5500.0	1.000000	0.000000	1.0	1.0	1.0	1.0	1.0
hasIfram	5500.0	0.989091	0.103885	0.0	1.0	1.0	1.0	1.0
hasMouseOver	5500.0	1.000000	0.000000	1.0	1.0	1.0	1.0	1.0
disabledRightClick	5500.0	1.000000	0.000000	1.0	1.0	1.0	1.0	1.0
isMultiDirected	5500.0	1.000000	0.000000	1.0	1.0	1.0	1.0	1.0

In [115]:

```
sc = StandardScaler()
X = sc.fit_transform(X)
```

In [116]:

```
from sklearn import tree

X_train, X_test, y_train, y_test = train_test_split(X,y, test_size = 0.2, random_state = 20)
print("The shape of X_train is      ", X_train.shape)
print("The shape of X_test is       ",X_test.shape)
print("The shape of y_train is      ",y_train.shape)
print("The shape of y_test is       ",y_test.shape)
```

```
The shape of X_train is       (4400, 25)
The shape of X_test is        (1100, 25)
The shape of y_train is       (4400,)
The shape of y_test is        (1100,)
```

In [117]:

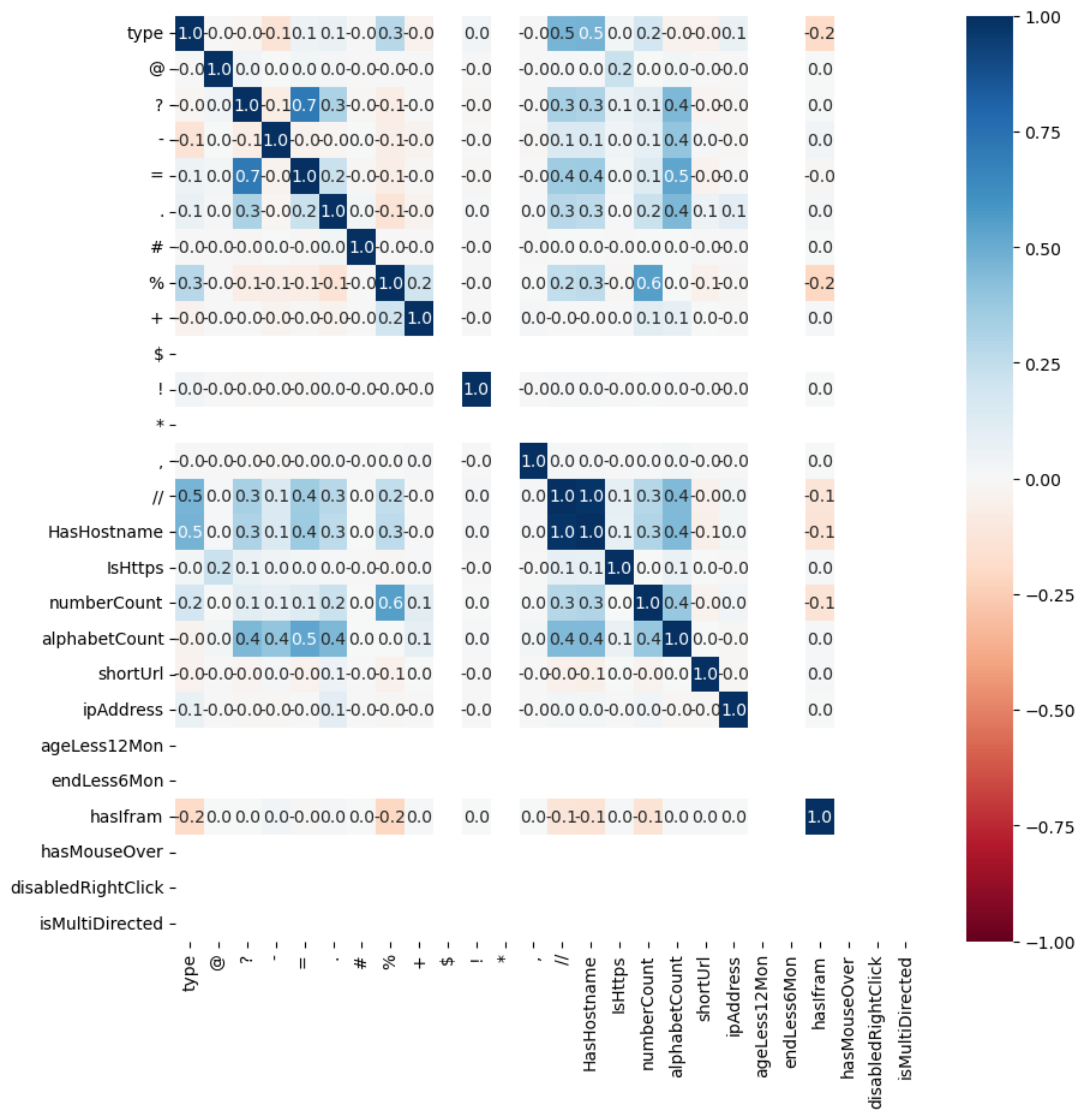
```
from sklearn.ensemble import RandomForestClassifier, AdaBoostClassifier, ExtraTreesClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn.linear_model import SGDClassifier
from sklearn.naive_bayes import GaussianNB
from sklearn.tree import DecisionTreeClassifier
```

In [25]:

```
#px.imshow(df.corr(),text_auto=True,aspect="auto",template="plotly_white").update_layout(title_text='Correla
plt.figure(figsize=(10,10))
sns.heatmap(df.corr(),cmap='RdBu',annot=True,fmt=".1f",vmin=-1, vmax=1)
```

Out[25]:

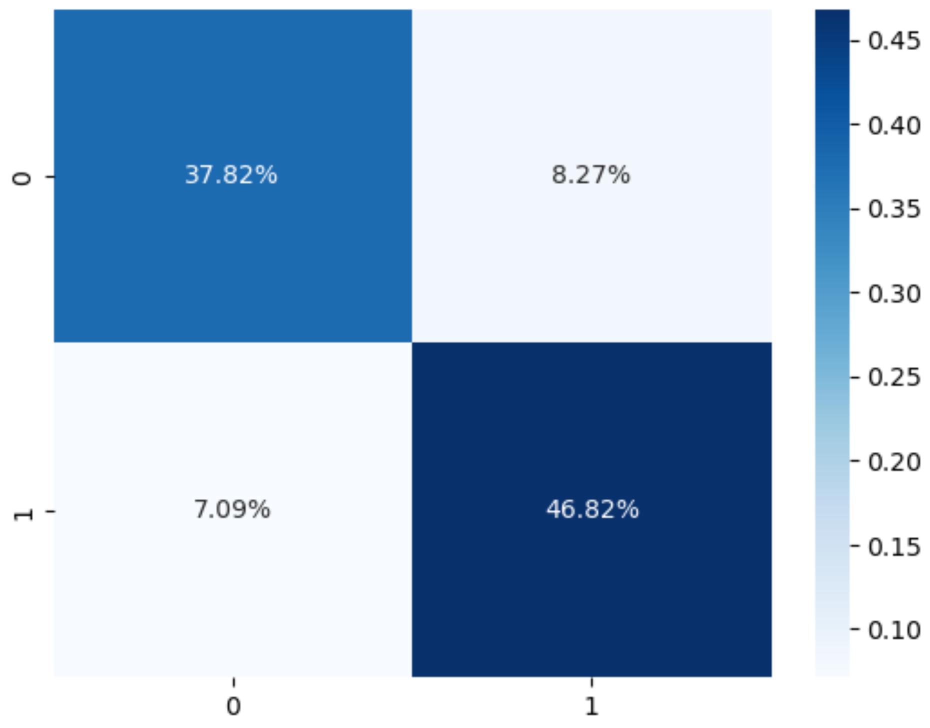
```
<AxesSubplot:>
```



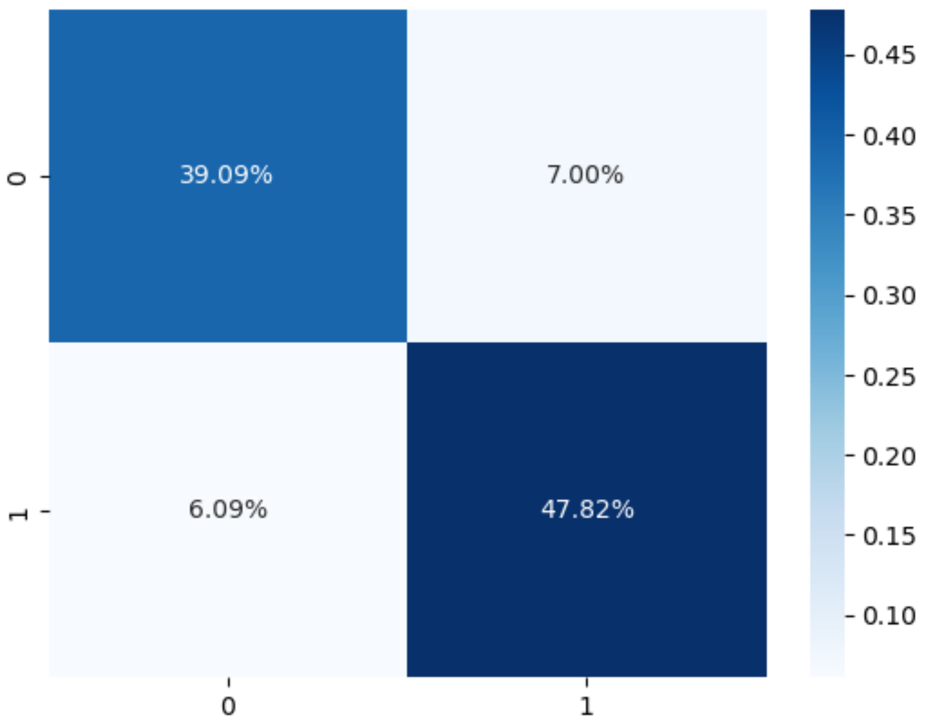
In [27]:

```
dels = [DecisionTreeClassifier,RandomForestClassifier,KNeighborsClassifier,LogisticRe
curacy_test=[]
meModels = ['DecisionTreeClassifier','RandomForestClassifier','KNeighborsClassifier',
Score = []
callScore = []
ecisionScore = []
r m in models:
 print("--------------------Start-----------------")
 print('-----Model =>\033[07m {} \033[0m'.format(m))
 model_ = m()
 model_.fit(X_train, y_train)
 pred = model_.predict(X_test)
 acc = accuracy_score(pred, y_test)
 f1 = f1_score(pred, y_test)
 recall = recall_score(pred, y_test)
 precision = precision_score(pred, y_test)


 train_yhat = model_.predict(X_train)
 train_acc = accuracy_score(y_train, train_yhat)


 accuracy_test.append(acc)
 f1Score.append(f1)
 recallScore.append(recall)
 precisionScore.append(precision)

 print('Test Accuracy :\033[32m \033[01m {:.2f}% \033[30m \033[0m'.format(acc*100))
 print('Train Accuracy :\033[32m \033[01m {:.2f}% \033[30m \033[0m'.format(train_acc
 print('\033[01m              Classification_report \033[0m')
 print(classification_report(y_test, pred))
 print('\033[01m             Confusion_matrix \033[0m')
 cf_matrix = confusion_matrix(y_test, pred)
 plot_ = sns.heatmap(cf_matrix/np.sum(cf_matrix), annot=True,fmt= '0.2%',cmap='Blues
 plt.show()
 print('\033[31m----------------- End --------------\033[0m')
```

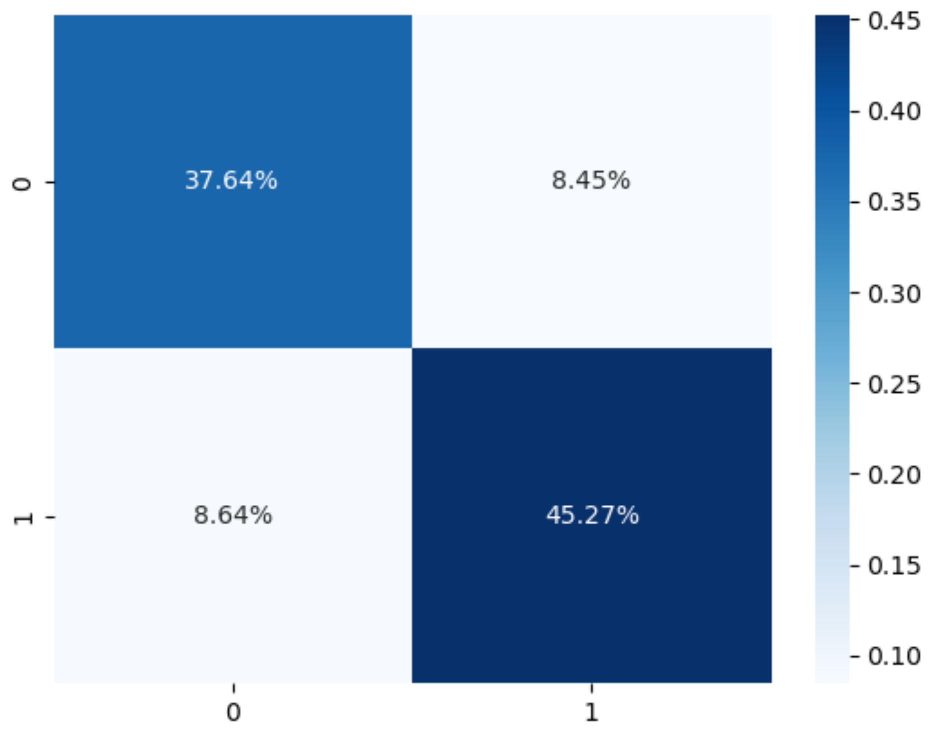
```
--------------------Start-----------------
-----Model => <class 'sklearn.tree._classes.DecisionTreeClassifier'>
Test Accuracy :  84.64%
Train Accuracy :  93.95%
              Classification_report
              precision    recall  f1-score   support

           0       0.84      0.82      0.83       507
           1       0.85      0.87      0.86       593

    accuracy                           0.85      1100
   macro avg       0.85      0.84      0.85      1100
weighted avg       0.85      0.85      0.85      1100

             Confusion_matrix
```



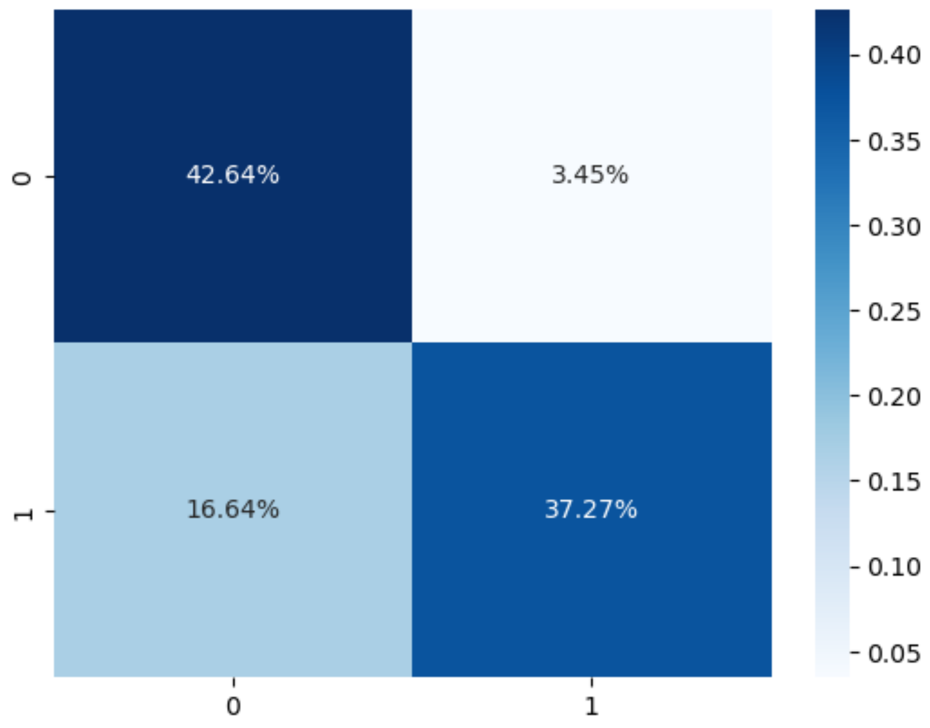
```
----------------- End --------------
--------------------Start----------------
-----Model => <class 'sklearn.ensemble._forest.RandomForestClassifie
r'>
Test Accuracy :  86.91%
Train Accuracy :  93.95%
              Classification_report
              precision    recall  f1-score   support

           0       0.87      0.85      0.86       507
           1       0.87      0.89      0.88       593

    accuracy                           0.87      1100
   macro avg       0.87      0.87      0.87      1100
weighted avg       0.87      0.87      0.87      1100

             Confusion_matrix
```



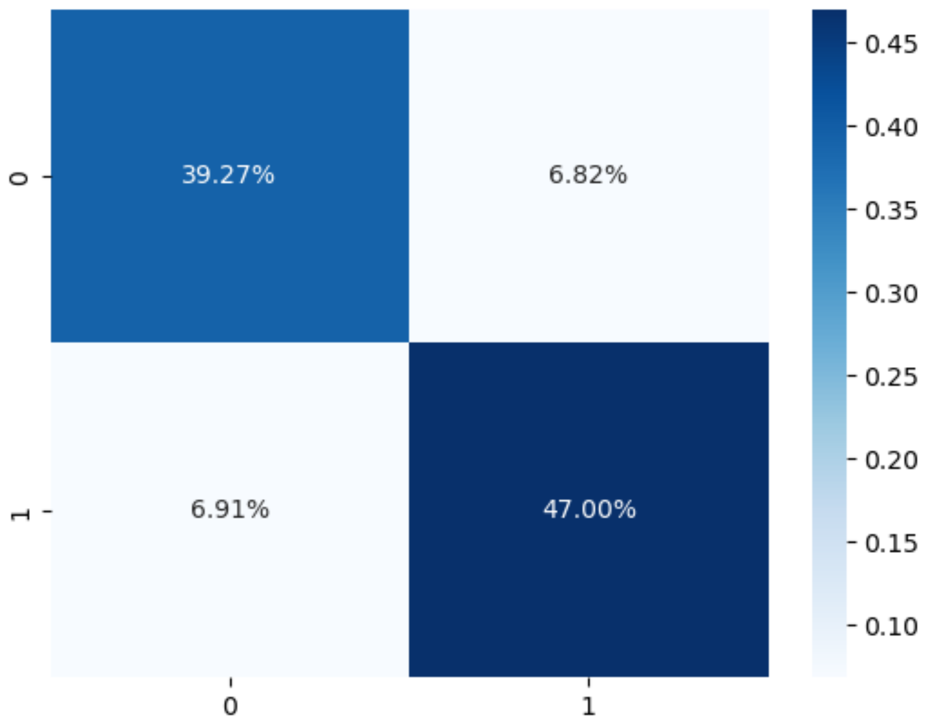
```
----------------- End --------------
--------------------Start----------------
-----Model => <class 'sklearn.neighbors._classification.KNeighborsClas
sifier'>
Test Accuracy :  82.91%
Train Accuracy :  89.14%
              Classification_report
              precision    recall  f1-score   support

           0       0.81      0.82      0.81       507
           1       0.84      0.84      0.84       593

    accuracy                           0.83      1100
   macro avg       0.83      0.83      0.83      1100
weighted avg       0.83      0.83      0.83      1100

             Confusion_matrix
```



```
----------------- End --------------
--------------------Start----------------
-----Model => <class 'sklearn.linear_model._logistic.LogisticRegressio
n'>
Test Accuracy :  79.91%
Train Accuracy :  80.18%
              Classification_report
              precision    recall  f1-score   support

           0       0.72      0.93      0.81       507
           1       0.92      0.69      0.79       593

    accuracy                           0.80      1100
   macro avg       0.82      0.81      0.80      1100
weighted avg       0.82      0.80      0.80      1100

             Confusion_matrix
```



```
------------------ End ---------------
--------------------Start-----------------
-----Model => <class 'sklearn.ensemble._gb.GradientBoostingClassifie
r'>
Test Accuracy :  86.27%
Train Accuracy :  89.02%
              Classification_report
              precision    recall  f1-score   support

           0       0.85      0.85      0.85       507
           1       0.87      0.87      0.87       593

    accuracy                           0.86      1100
   macro avg       0.86      0.86      0.86      1100
weighted avg       0.86      0.86      0.86      1100

             Confusion_matrix
```



```
----------------- End --------------
```

In [36]:

```
print('acc = {}'.format(accuracy_test))
print('F1 = {}'.format(f1Score))
print('recall = {}'.format(recallScore))
print('precision = {}'.format(precisionScore))
print('Models = {}'.format(nameModels))
```

```
acc = [0.8463636363636363, 0.8690909090909091, 0.8290909090909091, 0.7990909090909091, 0.8627272727272727]
F1 = [0.8590492076730608, 0.8795986622073578, 0.8412162162162162, 0.7877041306436118, 0.8725738396624473]
recall = [0.8498349834983498, 0.8723051409618574, 0.8426395939086294, 0.9151785714285714, 0.8733108108108109]
precision = [0.8684654300168634, 0.8870151770657673, 0.8397976391231029, 0.6913996627318718, 0.8718381112984823]
Models = ['DecisionTreeClassifier', 'RandomForestClassifier', 'KNeighborsClassifier', 'LogisticRegression', 'Grad
ientBoostingClassifier']
```

In [37]:

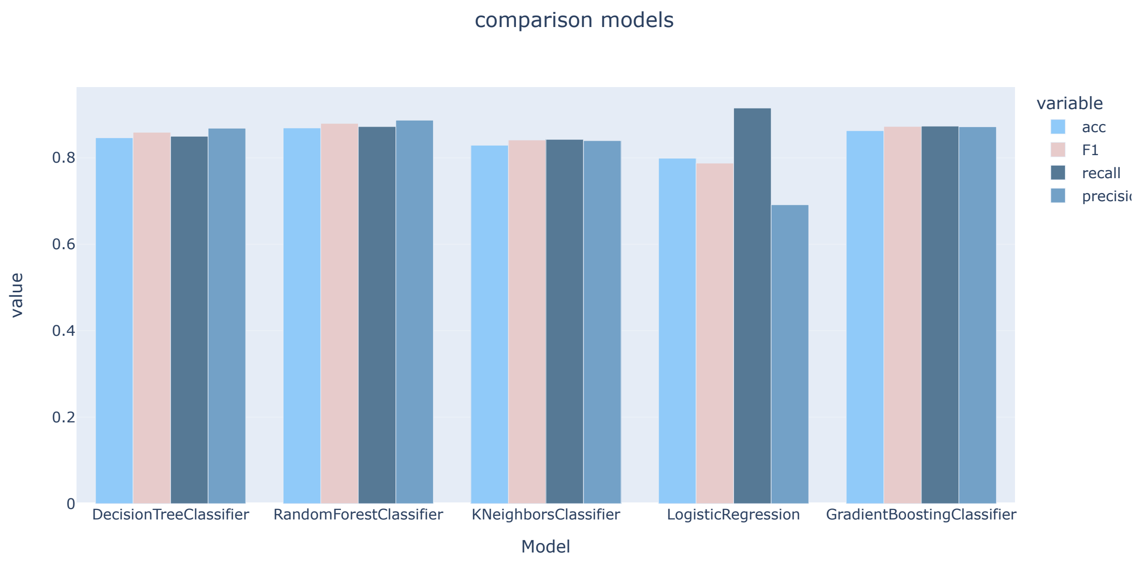
```
modelDetil = pd.DataFrame()
modelDetil['Model']=nameModels
modelDetil['acc']=accuracy_test
modelDetil['F1']=f1Score
modelDetil['recall']=recallScore
modelDetil['precision']=precisionScore
modelDetil
```

Out[37]:

	Model	acc	F1	recall	precision
0	DecisionTreeClassifier	0.846364	0.859049	0.849835	0.868465
1	RandomForestClassifier	0.869091	0.879599	0.872305	0.887015
2	KNeighborsClassifier	0.829091	0.841216	0.842640	0.839798
3	LogisticRegression	0.799091	0.787704	0.915179	0.691400
4	GradientBoostingClassifier	0.862727	0.872574	0.873311	0.871838

In [38]:

```
dels',color_discrete_map={'acc':'#90caf9','F1':'#e7cbcb','recall':'#567995', 'precision':'#73a1c7'}).update_layout(title_x=0.5)
```



In [ ]: