Chi2 independency test

May 22, 2021

Chi2 independency test is carried out to check whether a given class is depend on the features/attributes selected. We can verify the hypothesis that we defined for each feature, using this test.

Lets carry out tests for each feature.

0.1 Feature 1 - Links in Meta, Script, Link tag

[10]:
```
import pandas as pd #import pandas
import numpy as np #import numpy
import matplotlib.pyplot as plt #import matplotlib
from scipy.stats import chi2_contingency #import chi2_contigency for chi2␣␣
 ↪independency tet
from matplotlib import style
style.use('ggplot')
```

[43]:
```
df = pd.read_csv('dataset1.csv')

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

[43]:

	Links_in_tags	Abnormal_URL	Submitting_to_email	SFH	Iframe	popUpWidnow	\
0	73.913043	-1	0	0	0	0	
1	85.000000	-1	0	0	0	0	
2	97.000000	-1	0	0	0	0	
3	12.000000	-1	0	0	0	0	
4	55.555556	-1	0	0	0	0	

	on_mouseover	RightClick	Redirect	Result
0	0	0	0	1
1	0	0	0	1
2	0	0	1	1
3	0	0	0	-1
4	0	0	0	-1

Here, we have to check whether the website status is depend on feature 1 (links in meta tags)

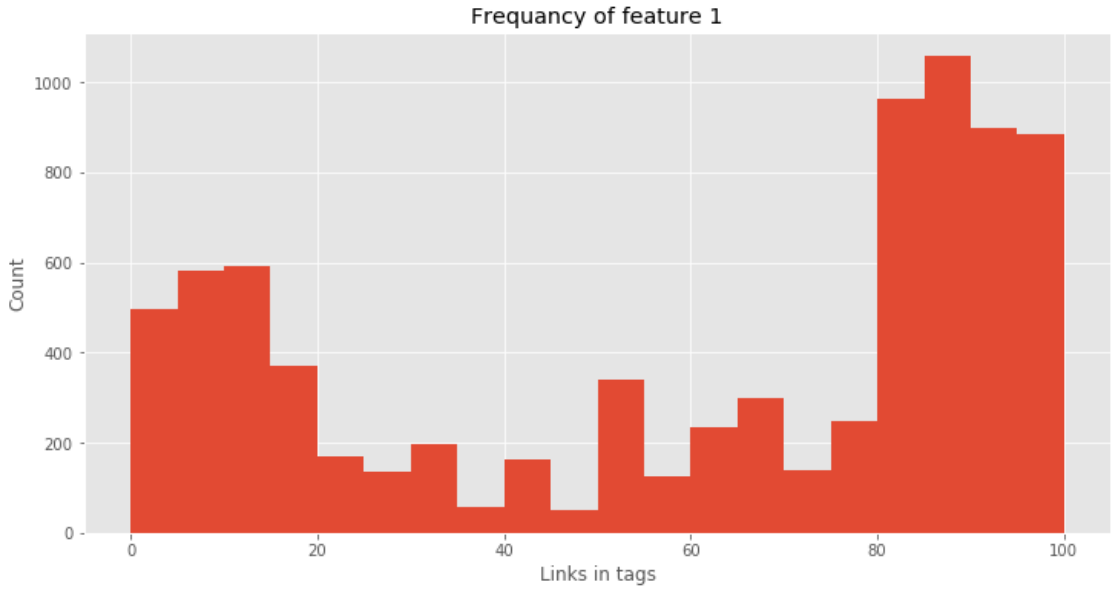
[45]:
```
df1 = df[['Links_in_tags','Result']]
df1.head()
```

[45]:

	Links_in_tags	Result
0	73.913043	1
1	85.000000	1
2	97.000000	1
3	12.000000	-1
4	55.555556	-1

[47]:
```
#Let's plot and check the frequency of links_in_tags feature

plt.figure(figsize=(12,6))
plt.hist(df1['Links_in_tags'], bins=20)
plt.title('Frequancy of feature 1')
plt.xlabel('Links in tags')
plt.ylabel('Count')
plt.show()
```



[48]:
```
#crosstab to check the website status and page rank.
# to get a better vissualization we will round off the values into nearest 10
df1_new = df1.copy()

df1_new.Links_in_tags = df1.Links_in_tags.round(-1) #round to nearest 10

cross_tab = pd.crosstab(df1_new['Links_in_tags'],df1_new['Result']).T
cross_tab
```

[48]:
```
Links_in_tags  0.0    10.0   20.0   30.0   40.0   50.0   60.0   70.0   80.0   \
Result
```

```
-1              576    983    532    178    166    261    293    326    404
 1               18     93     70     92     59    126     65    111    960

Links_in_tags  90.0   100.0
Result
-1              282      0
 1             1520    884
```

phishing websites indicated by 1

legitimate websites indicated by -1

statstical method to check whether the website's status dependent on feature 1 or not.

[49]:
```
#H0 : website status is independent of feature1
#H1 : website status depends on feature1
#Alpha : 0.05
alpha = 0.05

stats,p_value,degrees_of_freedom,expected = chi2_contingency(cross_tab)
if p_value > alpha:
  print(f' Accept Null Hypothesis\n P-Value is {p_value}\n Website status is␣
 ↪Independent of feature1')
else:
  print(f' Reject Null Hypothesis\n P-Value is {p_value}\n Website status␣
 ↪depends on feature1')
```

```
 Reject Null Hypothesis
 P-Value is 0.0
 Website status depends on feature1
```

Chi2 independency test tells us that the links in tags are not independent of status. We can also check it using

[51]:
```
phishing_df = df1[df1['Result']==1]  #store all the phishing websites in a␣
 ↪phishing
legitimate_df = df1[df1['Result']==-1]  #store all the legitimate websites in a␣
 ↪legitimate
```

[52]:
```
phishing_df.head()
```

[52]:

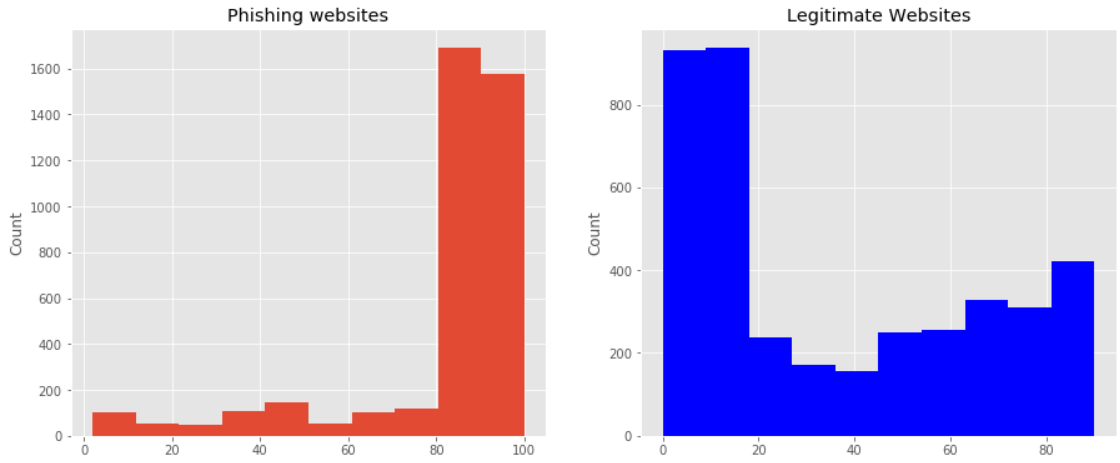
	Links_in_tags	Result
0	73.913043	1
1	85.000000	1
2	97.000000	1
5	81.000000	1
6	86.000000	1

[53]:
```
#plot different histograms for phishing and legitmate websites
fig,(ax1,ax2) = plt.subplots(1,2,figsize=(15,6))
```

```
phishing_df['Links_in_tags'].hist(ax=ax1)
ax1.set_title('Phishing websites')
ax1.set_ylabel('Count')

legitimate_df['Links_in_tags'].hist(ax=ax2,color='blue')
ax2.set_title('Legitimate Websites')
ax2.set_ylabel('Count')

plt.show()
```



[118]:
```
# According to the first graph, you can clearly see the links in tag percentage␣
 ↪greater than 85 are most likely to be phishing
# According to the second graph, you can clearly see the links in tag␣
 ↪percentage less than 20 are most likely to be legitimate
```

[ ]:

[ ]:

[ ]:

0.2 Feature 2 - Abnormal URL

[54]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from scipy.stats import chi2_contingency #import chi2_contigency for chi2␣␣
 ↪independency test
from matplotlib import style
style.use('ggplot')
```

```
df = pd.read_csv('dataset1.csv')

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

[54]:
```
   Links_in_tags  Abnormal_URL  Submitting_to_email  SFH  Iframe  popUpWidnow  \
0      73.913043            -1                    0    0       0            0
1      85.000000            -1                    0    0       0            0
2      97.000000            -1                    0    0       0            0
3      12.000000            -1                    0    0       0            0
4      55.555556            -1                    0    0       0            0

   on_mouseover  RightClick  Redirect  Result
0             0           0         0       1
1             0           0         0       1
2             0           0         1       1
3             0           0         0      -1
4             0           0         0      -1
```

[55]:
```
df2 = df[['Abnormal_URL','Result']]
df2.head()
```

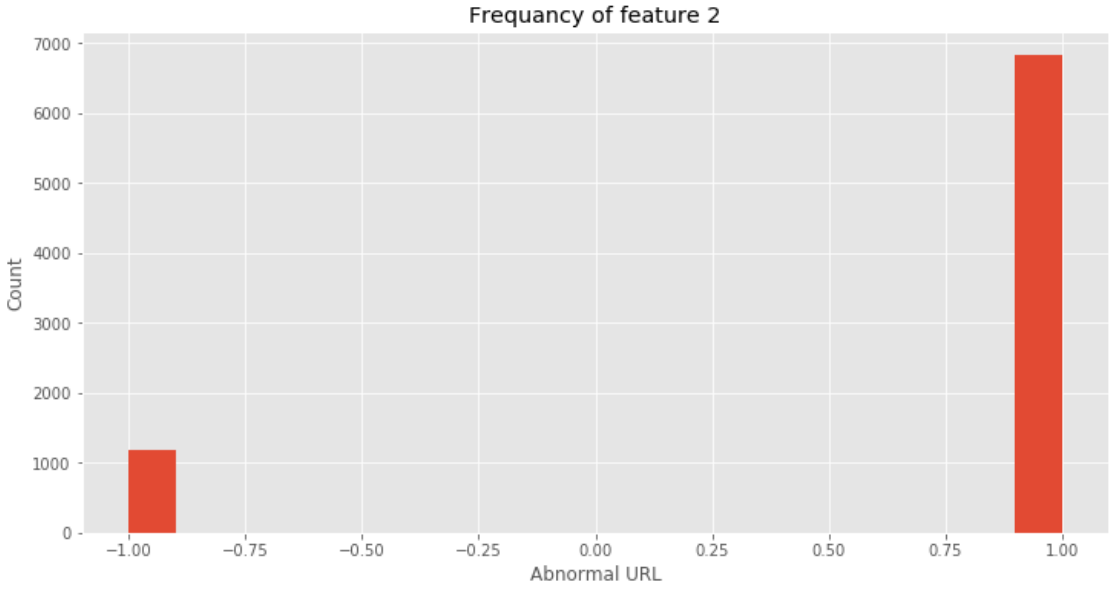
[55]:
```
   Abnormal_URL  Result
0            -1       1
1            -1       1
2            -1       1
3            -1      -1
4            -1      -1
```

[56]:
```
cross_tab = pd.crosstab(df2['Abnormal_URL'],df2['Result']).T
cross_tab
```

[56]:
```
Abnormal_URL   -1     1
Result
-1            741  3260
 1            437  3561
```

[57]:
```
#Let's plot and check the frequency of abnormal url feature

plt.figure(figsize=(12,6))
plt.hist(df2['Abnormal_URL'], bins=20)
plt.title('Frequancy of feature 2')
plt.xlabel('Abnormal URL')
plt.ylabel('Count')
plt.show()
```



[58]:
```
#H0 : website status is independent of feature2
#H1 : website status depends on feature2
#Alpha : 0.05
alpha = 0.05

stats,p_value,degrees_of_freedom,expected = chi2_contingency(cross_tab)
if p_value > alpha:
  print(f' Accept Null Hypothesis\n P-Value is {p_value}\n Website status is␣
 ↪Independent of feature2')
else:
  print(f' Reject Null Hypothesis\n P-Value is {p_value}\n Website status␣
 ↪depends on feature2')
```

```
Reject Null Hypothesis
P-Value is 1.3455226733682224e-21
Website status depends on feature2
```

Chi2 independency test tells us that the feature2 is not independent of status. We can also check it using

[59]:
```
phishing_df2 = df2[df2['Result']==1]  #store all the phishing websites in a␣
 ↪phishing
legitimate_df2 = df2[df2['Result']==-1]  #store all the legitimate websites in␣
 ↪a legitimate
phishing_df2.head()
```

[59]:

	Abnormal_URL	Result
0	-1	1

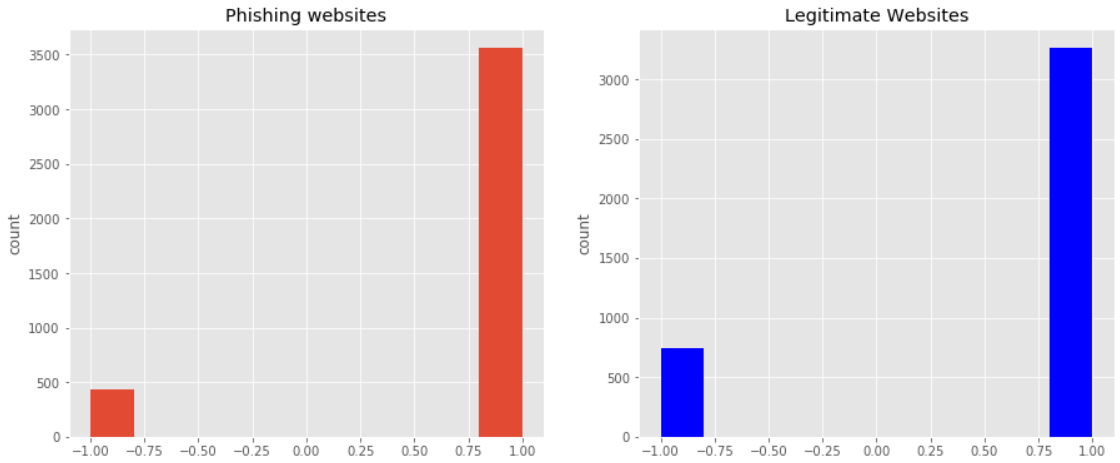
```
1             -1        1
2             -1        1
5             -1        1
6             -1        1
```

[60]:
```
#plot different histograms for phishing and legitmate websites
fig,(ax1,ax2) = plt.subplots(1,2,figsize=(15,6))

phishing_df2['Abnormal_URL'].hist(ax=ax1)
ax1.set_title('Phishing websites')
ax1.set_ylabel('count')

legitimate_df2['Abnormal_URL'].hist(ax=ax2,color='blue')
ax2.set_title('Legitimate Websites')
ax2.set_ylabel('count')

plt.show()
```



Above graphs shows that abnormal urls are higher in phsishing websites. So that we can conclude that URLs having abnormal URL feature is most likely to be phishing

[ ]:

[ ]:

[ ]:

[ ]:

0.3 Feature 3 - Website Forwarding

[61]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from scipy.stats import chi2_contingency #import chi2_contigency for chi2␣␣
 ↪independency test
from matplotlib import style
style.use('ggplot')

df = pd.read_csv('dataset1.csv')

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

[61]:

	Links_in_tags	Abnormal_URL	Submitting_to_email	SFH	Iframe	popUpWidnow	\
0	73.913043	-1	0	0	0	0	
1	85.000000	-1	0	0	0	0	
2	97.000000	-1	0	0	0	0	
3	12.000000	-1	0	0	0	0	
4	55.555556	-1	0	0	0	0	

	on_mouseover	RightClick	Redirect	Result
0	0	0	0	1
1	0	0	0	1
2	0	0	1	1
3	0	0	0	-1
4	0	0	0	-1

[63]:
```
df3 = df[['Redirect','Result']]
df3.head()
```

[63]:

	Redirect	Result
0	0	1
1	0	1
2	1	1
3	0	-1
4	0	-1

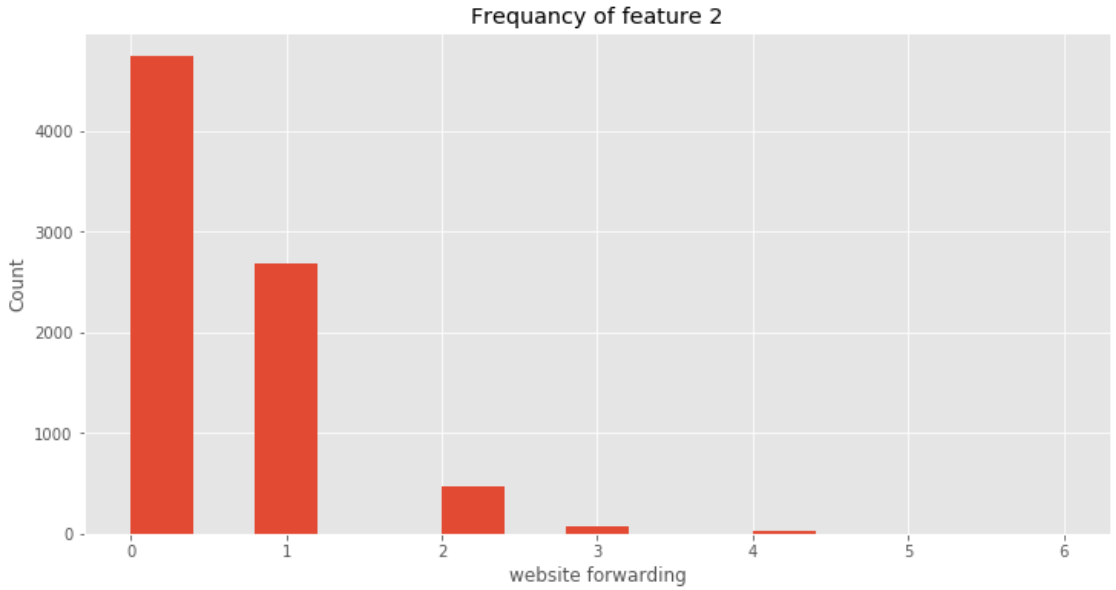
[64]:
```
cross_tab = pd.crosstab(df3['Redirect'],df3['Result']).T
cross_tab
```

[64]:

Redirect	0	1	2	3	4	5	6
Result							
-1	2311	1398	254	29	7	2	0
1	2433	1284	216	48	15	1	1

[70]:
```
#Let's plot and check the frequency of website forwarding feature

plt.figure(figsize=(12,6))
plt.hist(df3['Redirect'], bins=15)
plt.title('Frequancy of feature 3')
plt.xlabel('website forwarding')
plt.ylabel('Count')
plt.show()
```



[71]:
```
#H0 : website status is independent of feature3
#H1 : website status depends on feature3
#Alpha : 0.05
alpha = 0.05

stats,p_value,degrees_of_freedom,expected = chi2_contingency(cross_tab)
if p_value > alpha:
  print(f' Accept Null Hypothesis\n P-Value is {p_value}\n Website status is␣
 ↪Independent of feature3')
else:
  print(f' Reject Null Hypothesis\n P-Value is {p_value}\n Website status␣
 ↪depends on feature3')
```

```
Reject Null Hypothesis
P-Value is 0.002786439466466466
Website status depends on feature3
```

[ ]:

Chi2 independency test tells us that the feature3 is not independent of Result. We can also check it as follows

[72]:
```
phishing_df3 = df3[df3['Result']==1]  #store all the phishing websites in a␣
↪phishing
legitimate_df3 = df3[df3['Result']==-1]  #store all the legitimate websites in␣
↪a legitimate
phishing_df3.head()
```

[72]:

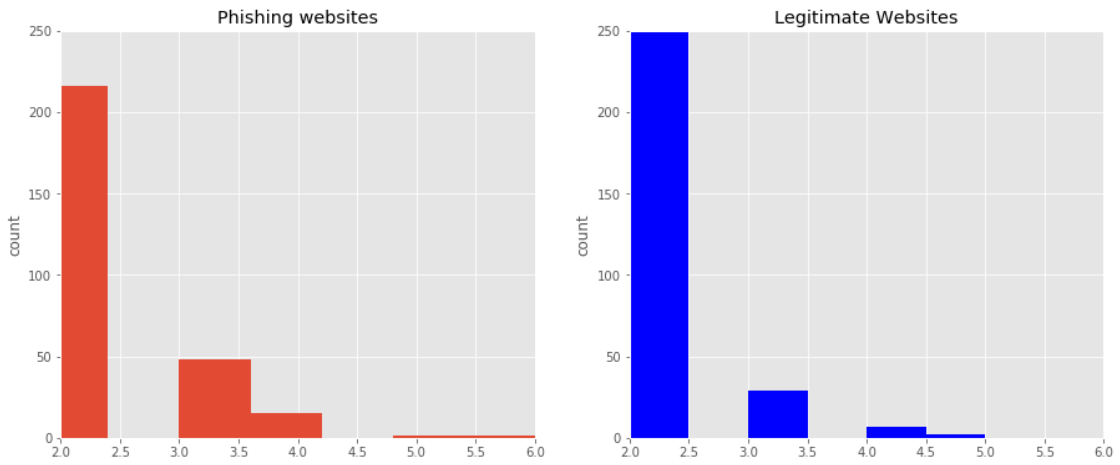
	Redirect	Result
0	0	1
1	0	1
2	1	1
5	0	1
6	0	1

[80]:
```
#plot different histograms for phishing and legitmate websites
fig,(ax1,ax2) = plt.subplots(1,2,figsize=(15,6))

phishing_df3['Redirect'].hist(ax=ax1)
ax1.set_title('Phishing websites')
ax1.set_ylabel('count')
ax1.set_xlim([2,6])
ax1.set_ylim([0,250])


legitimate_df3['Redirect'].hist(ax=ax2,color='blue')
ax2.set_title('Legitimate Websites')
ax2.set_ylabel('count')
ax2.set_xlim([2,6])
ax2.set_ylim([0,250])

plt.show()
```



According to the above graphs, we can clearly observe that websites with more than 3 redirections are most likely to be phishing

[ ]:

[ ]:

[ ]:

[ ]:

0.4 Feature 4 - Submitting Information to Email

[9]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from scipy.stats import chi2_contingency #import chi2_contigency for chi2␣␣
 ↪independency test
from matplotlib import style
style.use('ggplot')

df = pd.read_csv('dataset1.csv')

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

[9]:

	Links_in_tags	Abnormal_URL	Submitting_to_email	SFH	Iframe	popUpWidnow	\
0	73.913043	-1	1	0	0	0	
1	85.000000	-1	1	0	0	0	
2	97.000000	-1	1	0	0	0	
3	12.000000	-1	1	0	0	0	
4	55.555556	-1	1	0	0	0	

	on_mouseover	RightClick	Redirect	Result
0	0	0	0	1
1	0	0	0	1
2	0	0	1	1
3	0	0	0	-1
4	0	0	0	-1

[10]:
```
df4 = df[['Submitting_to_email','Result']]
df4.head()
```

[10]:

	Submitting_to_email	Result
0	1	1

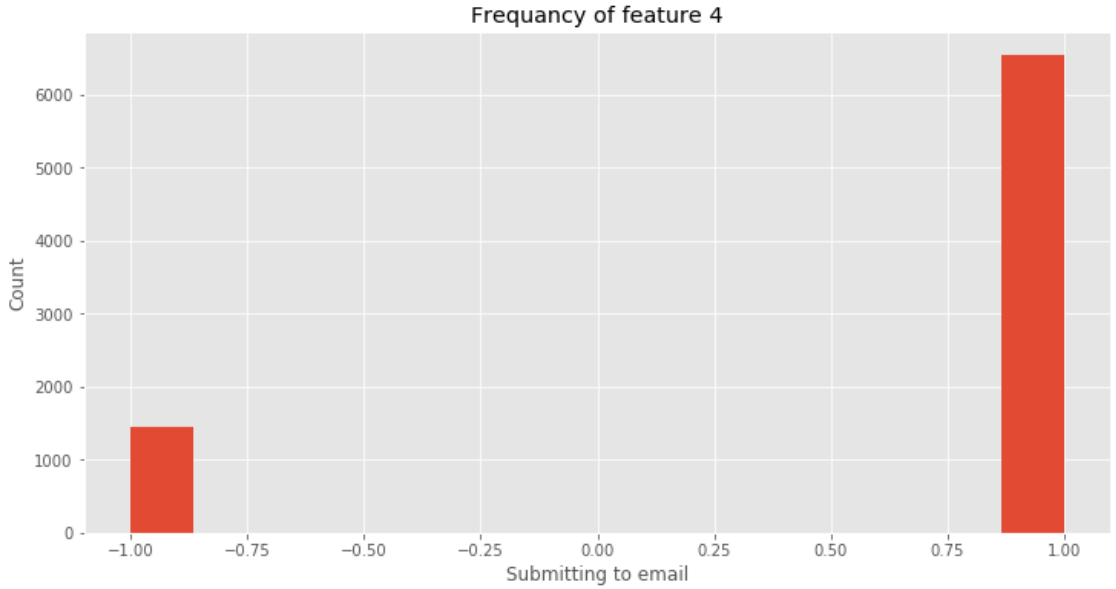
```
1                    1       1
2                    1       1
3                    1      -1
4                    1      -1
```

[12]:
```
cross_tab = pd.crosstab(df4['Submitting_to_email'],df4['Result']).T
cross_tab
```

[12]:
```
Submitting_to_email    -1     1
Result
-1                   1131  2870
 1                    327  3671
```

[13]:
```
#Let's plot and check the frequency of submit to email feature

plt.figure(figsize=(12,6))
plt.hist(df4['Submitting_to_email'], bins=15)
plt.title('Frequancy of feature 4')
plt.xlabel('Submitting to email')
plt.ylabel('Count')
plt.show()
```



[14]:
```
#H0 : website status is independent of feature4
#H1 : website status depends on feature4
#Alpha : 0.05
alpha = 0.05
```

```
stats,p_value,degrees_of_freedom,expected = chi2_contingency(cross_tab)
if p_value > alpha:
  print(f' Accept Null Hypothesis\n P-Value is {p_value}\n Website status is␣
 ↪Independent of feature4')
else:
  print(f' Reject Null Hypothesis\n P-Value is {p_value}\n Website status␣
 ↪depends on feature4')
```

```
Reject Null Hypothesis
P-Value is 1.7941687293486418e-119
Website status depends on feature4
```

Chi2 independency test tells us that the feature4 is not independent of Result. We can also check it as follows

[17]:
```
phishing_df4 = df4[df4['Result']==1]  #store all the phishing websites in a␣
 ↪phishing
legitimate_df4 = df4[df4['Result']==-1]  #store all the legitimate websites in␣
 ↪a legitimate
phishing_df4.head()
```

[17]:

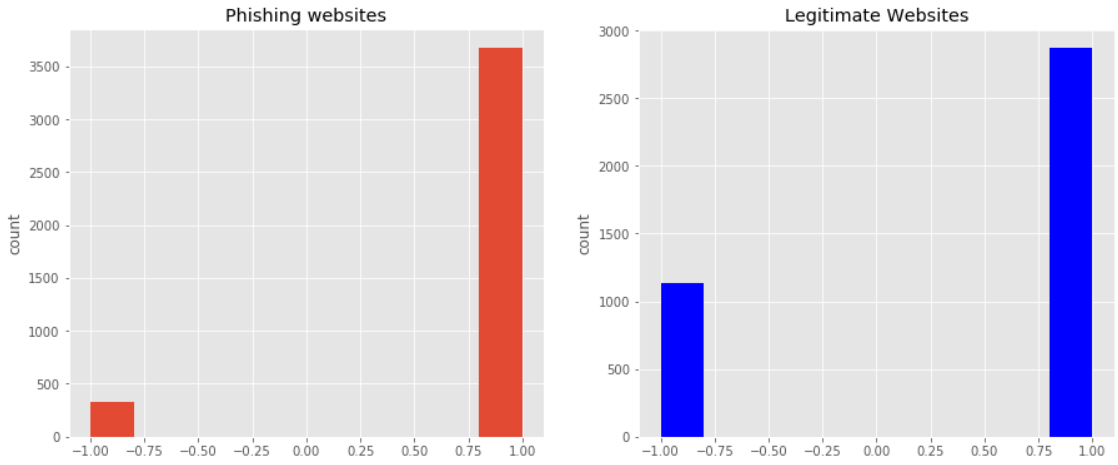
	Submitting_to_email	Result
0	1	1
1	1	1
2	1	1
5	1	1
6	1	1

[19]:
```
#plot different histograms for phishing and legitmate websites
fig,(ax1,ax2) = plt.subplots(1,2,figsize=(15,6))

phishing_df4['Submitting_to_email'].hist(ax=ax1)
ax1.set_title('Phishing websites')
ax1.set_ylabel('count')

legitimate_df4['Submitting_to_email'].hist(ax=ax2,color='blue')
ax2.set_title('Legitimate Websites')
ax2.set_ylabel('count')

plt.show()
```



According to the above graphs, we can clearly see that "submiting to email feature is available mostly in phishing websites than the legitimate websites

[ ]:

[ ]:

[ ]:

[ ]:

0.5 Feature 5 - IFrame Redirection

[26]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from scipy.stats import chi2_contingency #import chi2_contigency for chi2␣␣
 ↪independency test
from matplotlib import style
style.use('ggplot')

df = pd.read_csv('dataset1.csv')
df.drop('url',axis=1,inplace=True)
df.head()
```

[26]:

	Links_in_tags	Abnormal_URL	Submitting_to_email	SFH	Iframe	popUpWidnow	\
0	73.913043	-1	1	0	-1	0	
1	85.000000	-1	1	0	-1	0	
2	97.000000	-1	1	0	-1	0	
3	12.000000	-1	1	0	1	0	
4	55.555556	-1	1	0	1	0	

```
   on_mouseover  RightClick  Redirect  Result
0             0           0         0       1
1             0           0         0       1
2             0           0         1       1
3             0           0         0      -1
4             0           0         0      -1
```

[27]:
```
df5 = df[['Iframe','Result']]
df5.head()
```

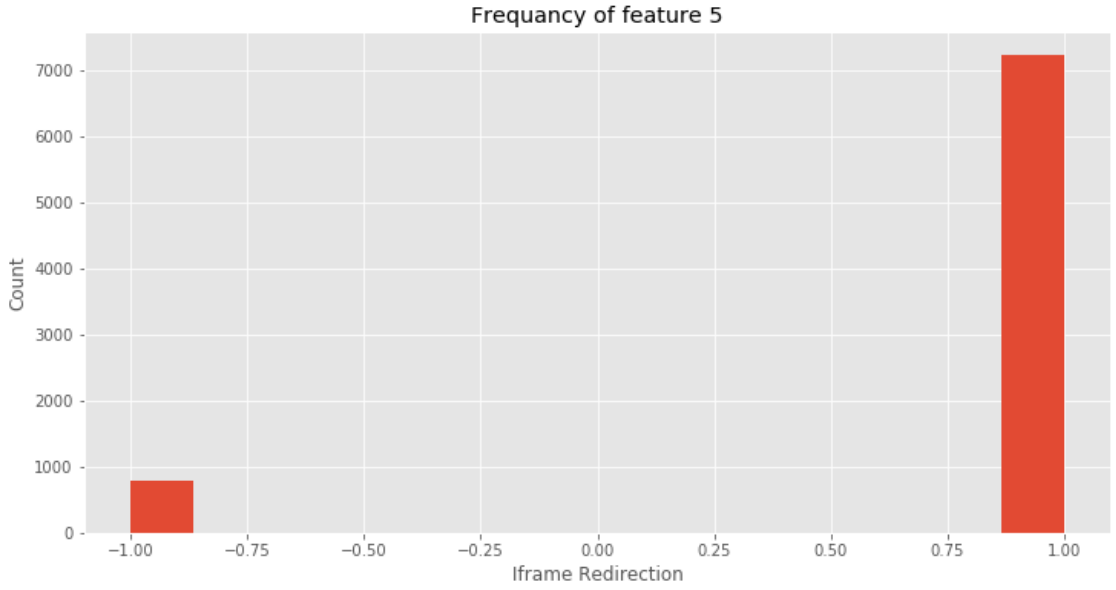
[27]:
```
   Iframe  Result
0      -1       1
1      -1       1
2      -1       1
3       1      -1
4       1      -1
```

[28]:
```
cross_tab = pd.crosstab(df5['Iframe'],df5['Result']).T
cross_tab
```

[28]:
```
Iframe   -1     1
Result
-1      773  3228
 1        7  3991
```

[29]:
```
#Let's plot and check the frequency of Iframe feature

plt.figure(figsize=(12,6))
plt.hist(df5['Iframe'], bins=15)
plt.title('Frequancy of feature 5')
plt.xlabel('Iframe Redirection')
plt.ylabel('Count')
plt.show()
```



[30]:
```
#H0 : website status is independent of feature5
#H1 : website status depends on feature5
#Alpha : 0.05
alpha = 0.05

stats,p_value,degrees_of_freedom,expected = chi2_contingency(cross_tab)
if p_value > alpha:
  print(f' Accept Null Hypothesis\n P-Value is {p_value}\n Website status is␣
 ↪Independent of feature5')
else:
  print(f' Reject Null Hypothesis\n P-Value is {p_value}\n Website status␣
 ↪depends on feature5')
```

```
Reject Null Hypothesis
P-Value is 1.130035218017294e-182
Website status depends on feature5
```

Chi2 independency test tells us that the feature5 is not independent of Result. We can also check it as follows

[31]:
```
phishing_df5 = df5[df5['Result']==1]  #store all the phishing websites in a␣
 ↪phishing
legitimate_df5 = df5[df5['Result']==-1]  #store all the legitimate websites in␣
 ↪a legitimate
phishing_df5.head()
```

[31]:

	Iframe	Result
0	-1	1

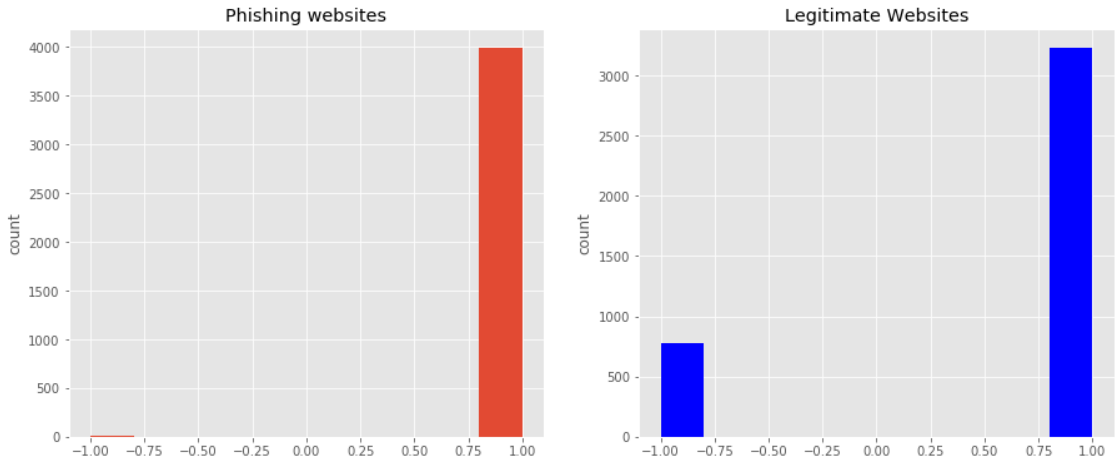
```
1       -1        1
2       -1        1
5       -1        1
6       -1        1
```

[32]:
```
#plot different histograms for phishing and legitmate websites
fig,(ax1,ax2) = plt.subplots(1,2,figsize=(15,6))

phishing_df5['Iframe'].hist(ax=ax1)
ax1.set_title('Phishing websites')
ax1.set_ylabel('count')

legitimate_df5['Iframe'].hist(ax=ax2,color='blue')
ax2.set_title('Legitimate Websites')
ax2.set_ylabel('count')

plt.show()
```



According to the above graphs, we can clearly see that “Iframe redirection” feature is available mostly in phishing websites than the legitimate websites

[ ]:

[ ]:

0.6 Feature 6 - Using Pop-up Window

[36]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

```
from scipy.stats import chi2_contingency #import chi2_contigency for chi2␣␣
 ↪independency test
from matplotlib import style
style.use('ggplot')

df = pd.read_csv('dataset1.csv')
df.drop('url',axis=1,inplace=True)
df.head()
```

[36]:

	Links_in_tags	Abnormal_URL	Submitting_to_email	SFH	Iframe	popUpWidnow	\
0	73.913043	-1	1	0	-1	1	
1	85.000000	-1	1	0	-1	1	
2	97.000000	-1	1	0	-1	1	
3	12.000000	-1	1	0	1	-1	
4	55.555556	-1	1	0	1	-1	

	on_mouseover	RightClick	Redirect	Result
0	0	0	0	1
1	0	0	0	1
2	0	0	1	1
3	0	0	0	-1
4	0	0	0	-1

[37]:
```
df6 = df[['popUpWidnow','Result']]
df6.head()
```

[37]:

	popUpWidnow	Result
0	1	1
1	1	1
2	1	1
3	-1	-1
4	-1	-1

[38]:
```
cross_tab = pd.crosstab(df6['popUpWidnow'],df6['Result']).T
cross_tab
```

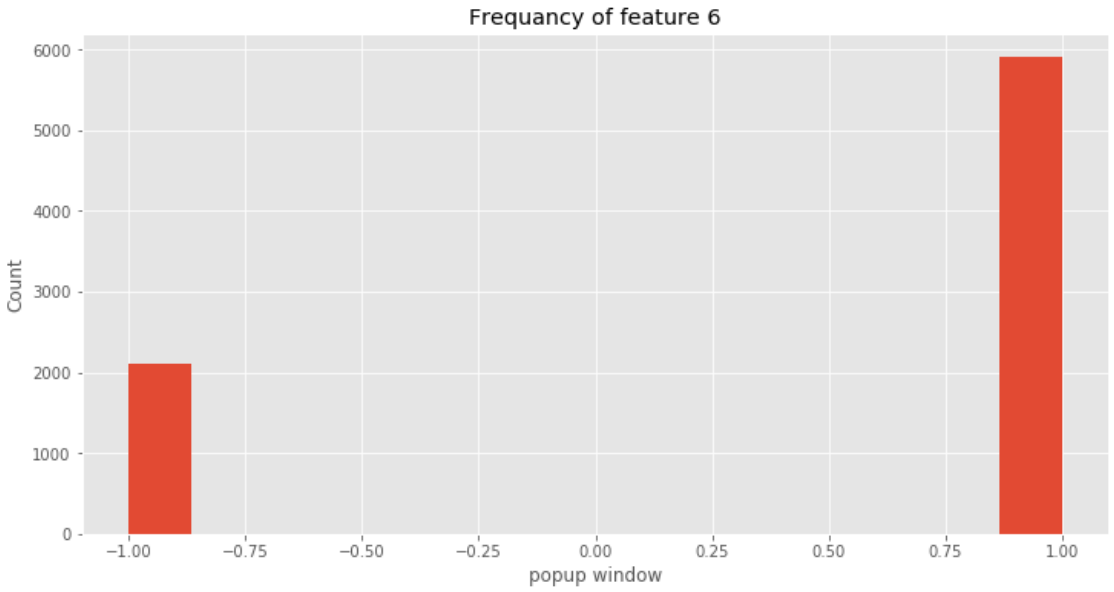
[38]:

popUpWidnow	-1	1
Result		
-1	1354	2647
1	743	3255

[39]:
```
#Let's plot and check the frequency of popup window feature

plt.figure(figsize=(12,6))
plt.hist(df6['popUpWidnow'], bins=15)
plt.title('Frequancy of feature 6')
plt.xlabel('popup window')
```

```
plt.ylabel('Count')
plt.show()
```



[40]:
```
#H0 : website status is independent of feature6
#H1 : website status depends on feature6
#Alpha : 0.05
alpha = 0.05

stats,p_value,degrees_of_freedom,expected = chi2_contingency(cross_tab)
if p_value > alpha:
  print(f' Accept Null Hypothesis\n P-Value is {p_value}\n Website status is␣
 ↪Independent of feature6')
else:
  print(f' Reject Null Hypothesis\n P-Value is {p_value}\n Website status␣
 ↪depends on feature6')
```

```
Reject Null Hypothesis
P-Value is 4.1966169165675175e-54
Website status depends on feature6
```

Chi2 independency test tells us that the feature6 is not independent of Result. We can also check it as follows

[41]:
```
phishing_df6 = df6[df6['Result']==1]  #store all the phishing websites in a␣
 ↪phishing
legitimate_df6 = df6[df6['Result']==-1]  #store all the legitimate websites in␣
 ↪a legitimate
phishing_df6.head()
```

[41]:

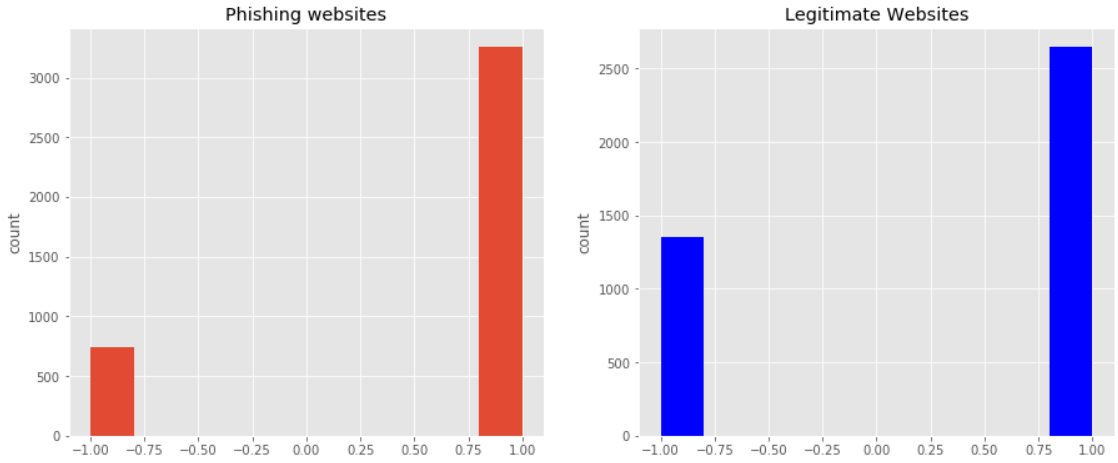
	popUpWidnow	Result
0	1	1
1	1	1
2	1	1
5	1	1
6	1	1

[56]:
```
#plot different histograms for phishing and legitmate websites
fig,(ax1,ax2) = plt.subplots(1,2,figsize=(15,6))

phishing_df6['popUpWidnow'].hist(ax=ax1)
ax1.set_title('Phishing websites')
ax1.set_ylabel('count')

legitimate_df6['popUpWidnow'].hist(ax=ax2,color='blue')
ax2.set_title('Legitimate Websites')
ax2.set_ylabel('count')

plt.show()
```



According to the above graphs, we can clearly see that “Popup window” feature is available mostly in phishing websites than the legitimate websites

[ ]:

[ ]:

[ ]:

[ ]:

[ ]:

0.7 Feature 7 - Disabling Right Click

[50]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from scipy.stats import chi2_contingency #import chi2_contigency for chi2␣␣
 ↪independency test
from matplotlib import style
style.use('ggplot')

df = pd.read_csv('dataset1.csv')
df.drop('url',axis=1,inplace=True)
df.head()
```

[50]:

	Links_in_tags	Abnormal_URL	Submitting_to_email	SFH	Iframe	popUpWidnow	\
0	73.913043	-1	1	0	-1	1	
1	85.000000	-1	1	0	-1	1	
2	97.000000	-1	1	0	-1	1	
3	12.000000	-1	1	0	1	-1	
4	55.555556	-1	1	0	1	-1	

	on_mouseover	RightClick	Redirect	Result
0	0	1	0	1
1	0	1	0	1
2	0	1	1	1
3	0	-1	0	-1
4	0	-1	0	-1

[51]:
```
df7 = df[['RightClick','Result']]
df7.head()
```

[51]:

	RightClick	Result
0	1	1
1	1	1
2	1	1
3	-1	-1
4	-1	-1

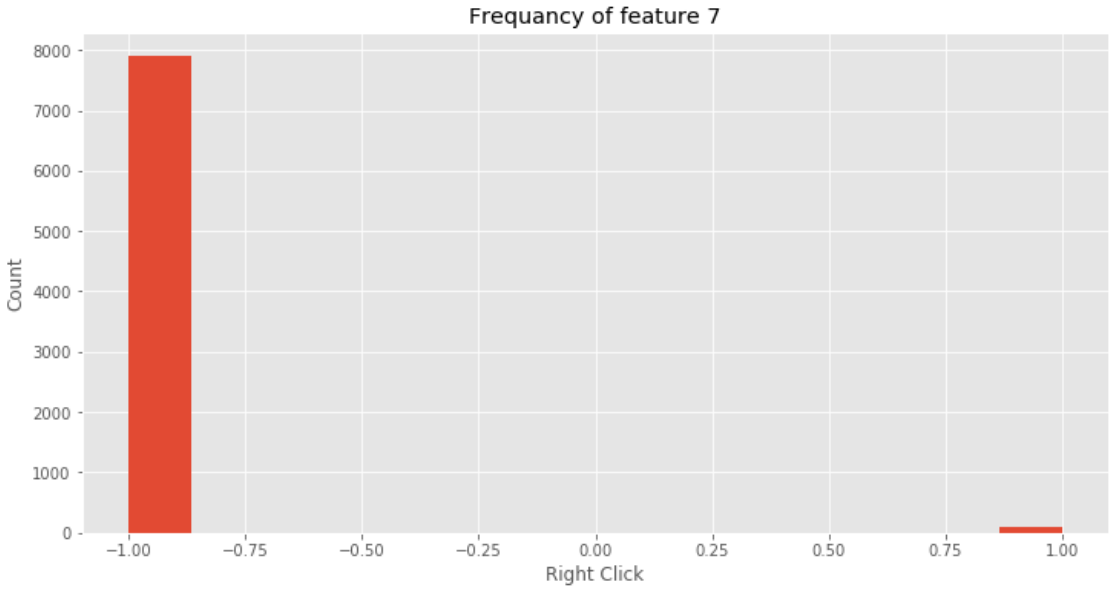
[52]:
```
cross_tab = pd.crosstab(df7['RightClick'],df7['Result']).T
cross_tab
```

[52]:

RightClick	-1	1
Result		
-1	3984	17
1	3915	83

[53]:
```
#Let's plot and check the frequency of right click feature
```

```
plt.figure(figsize=(12,6))
plt.hist(df7['RightClick'], bins=15)
plt.title('Frequancy of feature 7')
plt.xlabel('Right Click')
plt.ylabel('Count')
plt.show()
```



[54]:
```
#H0 : website status is independent of feature7
#H1 : website status depends on feature7
#Alpha : 0.05
alpha = 0.05

stats,p_value,degrees_of_freedom,expected = chi2_contingency(cross_tab)
if p_value > alpha:
  print(f' Accept Null Hypothesis\n P-Value is {p_value}\n Website status is␣
 ↪Independent of feature7')
else:
  print(f' Reject Null Hypothesis\n P-Value is {p_value}\n Website status␣
 ↪depends on feature7')
```

```
Reject Null Hypothesis
P-Value is 5.957918860957485e-11
Website status depends on feature7
```

Chi2 independency test tells us that the feature7 is not independent of Result. We can also check it as follows

[55]:
```
phishing_df7 = df7[df7['Result']==1]  #store all the phishing websites in a␣
 ↪phishing
legitimate_df7 = df7[df7['Result']==-1]  #store all the legitimate websites in␣
 ↪a legitimate
phishing_df7.head()
```

[55]:

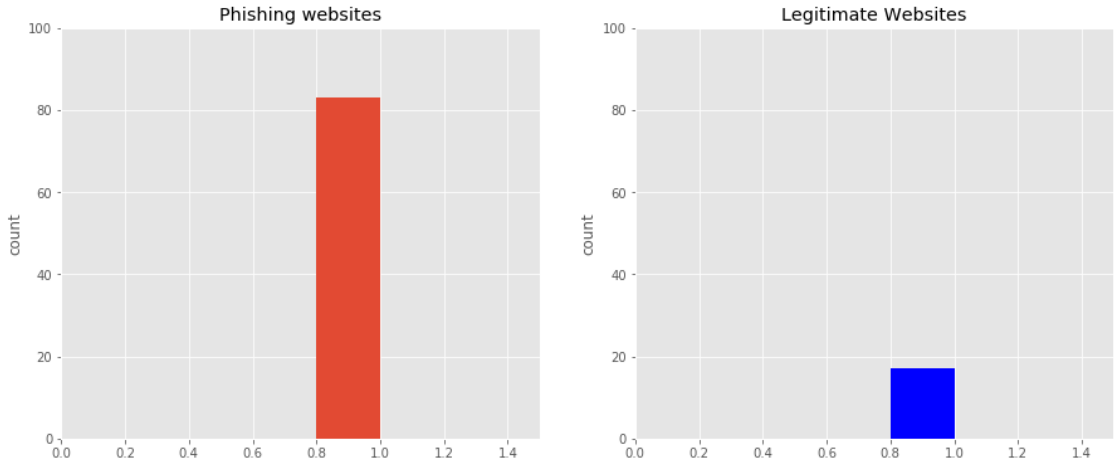
	RightClick	Result
0	1	1
1	1	1
2	1	1
5	1	1
6	1	1

[61]:
```
#plot different histograms for phishing and legitmate websites
fig,(ax1,ax2) = plt.subplots(1,2,figsize=(15,6))

phishing_df7['RightClick'].hist(ax=ax1)
ax1.set_title('Phishing websites')
ax1.set_ylabel('count')
ax1.set_xlim([0,1.5])
ax1.set_ylim([0,100])

legitimate_df7['RightClick'].hist(ax=ax2,color='blue')
ax2.set_title('Legitimate Websites')
ax2.set_ylabel('count')
ax2.set_xlim([0,1.5])
ax2.set_ylim([0,100])

plt.show()
```



According to the above graphs, we can clearly see that “disabling right click” feature is available

mostly in phishing websites than the legitimate websites

[ ]:

[ ]:

[ ]:

[ ]:

[ ]:

0.8 Feature 8 - Status Bar Customization

[65]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from scipy.stats import chi2_contingency #import chi2_contigency for chi2␣␣
 ↪independency test
from matplotlib import style
style.use('ggplot')

df = pd.read_csv('dataset1.csv')
df.drop('url',axis=1,inplace=True)
df.head()
```

[65]:

	Links_in_tags	Abnormal_URL	Submitting_to_email	SFH	Iframe	popUpWidnow	\
0	73.913043	-1	1	0	-1	1	
1	85.000000	-1	1	0	-1	1	
2	97.000000	-1	1	0	-1	1	
3	12.000000	-1	1	0	1	-1	
4	55.555556	-1	1	0	1	-1	

	on_mouseover	RightClick	Redirect	Result
0	1	1	0	1
1	1	1	0	1
2	1	1	1	1
3	-1	-1	0	-1
4	-1	-1	0	-1

[66]:
```
df8 = df[['on_mouseover','Result']]
df8.head()
```

[66]:

	on_mouseover	Result
0	1	1
1	1	1
2	1	1
3	-1	-1

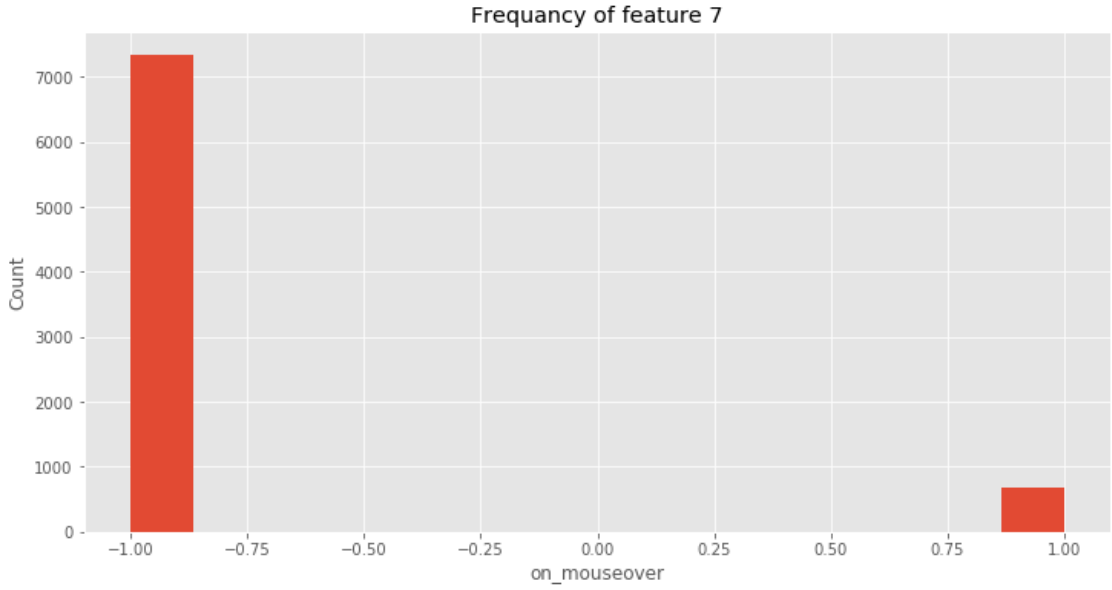
```
4             -1      -1
```

[67]:
```
cross_tab = pd.crosstab(df8['on_mouseover'],df8['Result']).T
cross_tab
```

[67]:
```
on_mouseover    -1    1
Result
-1            3993    8
 1            3338  660
```

[68]:
```
#Let's plot and check the frequency of feature 8

plt.figure(figsize=(12,6))
plt.hist(df8['on_mouseover'], bins=15)
plt.title('Frequancy of feature 7')
plt.xlabel('on_mouseover')
plt.ylabel('Count')
plt.show()
```



[69]:
```
#H0 : website status is independent of feature8
#H1 : website status depends on feature8
#Alpha : 0.05
alpha = 0.05

stats,p_value,degrees_of_freedom,expected = chi2_contingency(cross_tab)
if p_value > alpha:
```

```
  print(f' Accept Null Hypothesis\n P-Value is {p_value}\n Website status is␣
 ↪Independent of feature8')
else:
  print(f' Reject Null Hypothesis\n P-Value is {p_value}\n Website status␣
 ↪depends on feature8')
```

```
Reject Null Hypothesis
P-Value is 1.1138381147533346e-152
Website status depends on feature8
```

Chi2 independency test tells us that the feature8 is not independent of Result. We can also check it as follows

[70]:
```
phishing_df8 = df8[df8['Result']==1]  #store all the phishing websites in a␣
 ↪phishing
legitimate_df8 = df8[df8['Result']==-1]  #store all the legitimate websites in␣
 ↪a legitimate
phishing_df7.head()
```

[70]:

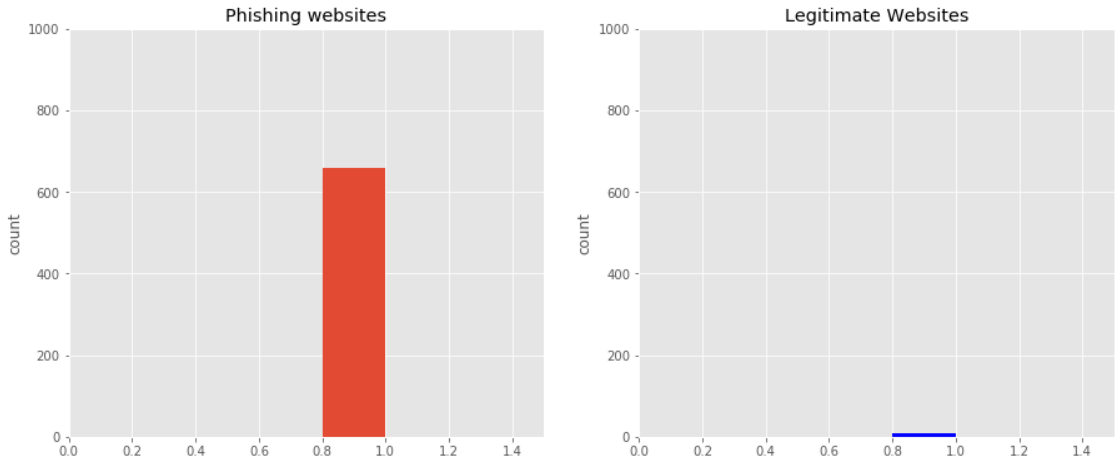
	RightClick	Result
0	1	1
1	1	1
2	1	1
5	1	1
6	1	1

[71]:
```
#plot different histograms for phishing and legitmate websites
fig,(ax1,ax2) = plt.subplots(1,2,figsize=(15,6))

phishing_df8['on_mouseover'].hist(ax=ax1)
ax1.set_title('Phishing websites')
ax1.set_ylabel('count')
ax1.set_xlim([0,1.5])
ax1.set_ylim([0,1000])

legitimate_df8['on_mouseover'].hist(ax=ax2,color='blue')
ax2.set_title('Legitimate Websites')
ax2.set_ylabel('count')
ax2.set_xlim([0,1.5])
ax2.set_ylim([0,1000])

plt.show()
```



According to the above graphs, we can clearly see that "status bar customization" feature is available mostly in phishing websites than the legitimate websites

0.9 Feature 9 - Server Form Handler

[7]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from scipy.stats import chi2_contingency #import chi2_contigency for chi2␣␣
 ↪independency test
from matplotlib import style
style.use('ggplot')

df = pd.read_csv('dataset1.csv')
df.drop('url',axis=1,inplace=True)
df.head()
```

[7]:

	Links_in_tags	Abnormal_URL	Submitting_to_email	SFH	Iframe	popUpWidnow	\
0	73.913043	-1	1	1	-1	1	
1	85.000000	-1	1	1	-1	1	
2	97.000000	-1	1	1	-1	1	
3	12.000000	-1	1	-1	1	-1	
4	55.555556	-1	1	-1	1	-1	

	on_mouseover	RightClick	Redirect	Result
0	1	1	0	1
1	1	1	0	1
2	1	1	1	1
3	-1	-1	0	-1
4	-1	-1	0	-1

[8]:
```
df9 = df[['SFH','Result']]
df9.head()
```

[8]:

	SFH	Result
0	1	1
1	1	1
2	1	1
3	-1	-1
4	-1	-1

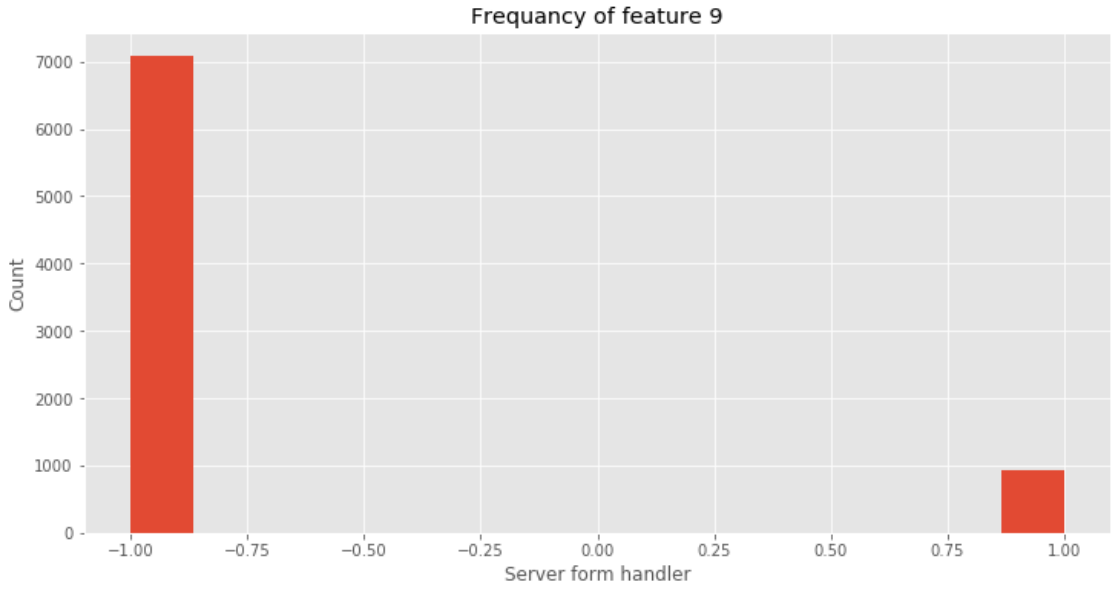
[9]:
```
cross_tab = pd.crosstab(df9['SFH'],df9['Result']).T
cross_tab
```

[9]:

SFH	-1	1
Result		
-1	3889	112
1	3191	807

[10]:
```
#Let's plot and check the frequency of SFH feature

plt.figure(figsize=(12,6))
plt.hist(df9['SFH'], bins=15)
plt.title('Frequancy of feature 9')
plt.xlabel('Server form handler')
plt.ylabel('Count')
plt.show()
```



[11]:
```
#H0 : website status is independent of feature9
#H1 : website status depends on feature9
#Alpha : 0.05
alpha = 0.05

stats,p_value,degrees_of_freedom,expected = chi2_contingency(cross_tab)
if p_value > alpha:
  print(f' Accept Null Hypothesis\n P-Value is {p_value}\n Website status is␣
 ↪Independent of feature9')
else:
  print(f' Reject Null Hypothesis\n P-Value is {p_value}\n Website status␣
 ↪depends on feature9')
```

```
 Reject Null Hypothesis
 P-Value is 6.470665866834179e-131
 Website status depends on feature9
```

Chi2 independency test tells us that the feature 9 is not independent of Result. We can also check it as follows

[12]:
```
phishing_df9 = df9[df9['Result']==1]  #store all the phishing websites in a␣
 ↪phishing
legitimate_df9 = df9[df9['Result']==-1]  #store all the legitimate websites in␣
 ↪a legitimate
phishing_df9.head()
```

[12]:

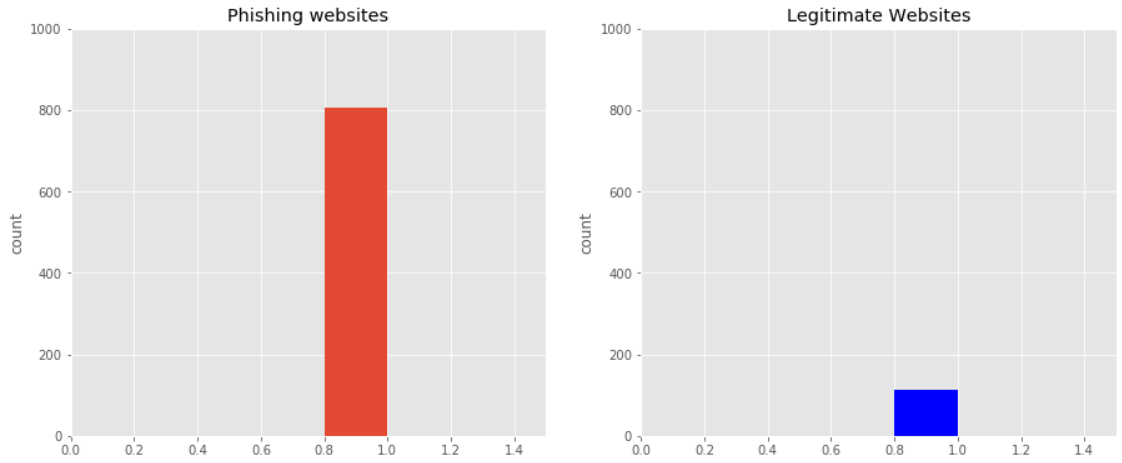
	SFH	Result
0	1	1
1	1	1
2	1	1
5	1	1
6	1	1

[14]:
```
#plot different histograms for phishing and legitmate websites
fig,(ax1,ax2) = plt.subplots(1,2,figsize=(15,6))

phishing_df9['SFH'].hist(ax=ax1)
ax1.set_title('Phishing websites')
ax1.set_ylabel('count')
ax1.set_xlim([0,1.5])
ax1.set_ylim([0,1000])

legitimate_df9['SFH'].hist(ax=ax2,color='blue')
ax2.set_title('Legitimate Websites')
ax2.set_ylabel('count')
ax2.set_xlim([0,1.5])
ax2.set_ylim([0,1000])
```

```
plt.show()
```



According to the above graphs, we can clearly see that ” server form handler” feature is available mostly in phishing websites than the legitimate websites

[ ]: