In [4]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns
import time

from sklearn.metrics import classification_report
from sklearn.metrics import confusion_matrix
from nltk.tokenize import RegexpTokenizer
from nltk.stem.snowball import SnowballStemmer
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.pipeline import make_pipeline

from PIL import Image

import pickle
```

In [6]:
```
df= pd.read_csv("C:/Shreya/AIURL/AIURL/Src/data/phishing_site_urls.csv")
df.head()
```

Out[6]:

	URL	Label
0	nobell.it/70ffb52d079109dca5664cce6f317373782/...	bad
1	www.dghjdgf.com/paypal.co.uk/cycgi-bin/webscrc...	bad
2	serviciosbys.com/paypal.cgi.bin.get-into.herf....	bad
3	mail.printakid.com/www.online.americanexpress....	bad
4	thewhiskeydregs.com/wp-content/themes/widescre...	bad

In [7]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 549346 entries, 0 to 549345
Data columns (total 2 columns):
 #   Column  Non-Null Count   Dtype 
---  ------  --------------   ----- 
 0   URL     549346 non-null  object
 1   Label   549346 non-null  object
dtypes: object(2)
memory usage: 8.4+ MB
```

In [8]:
```
df.shape
```

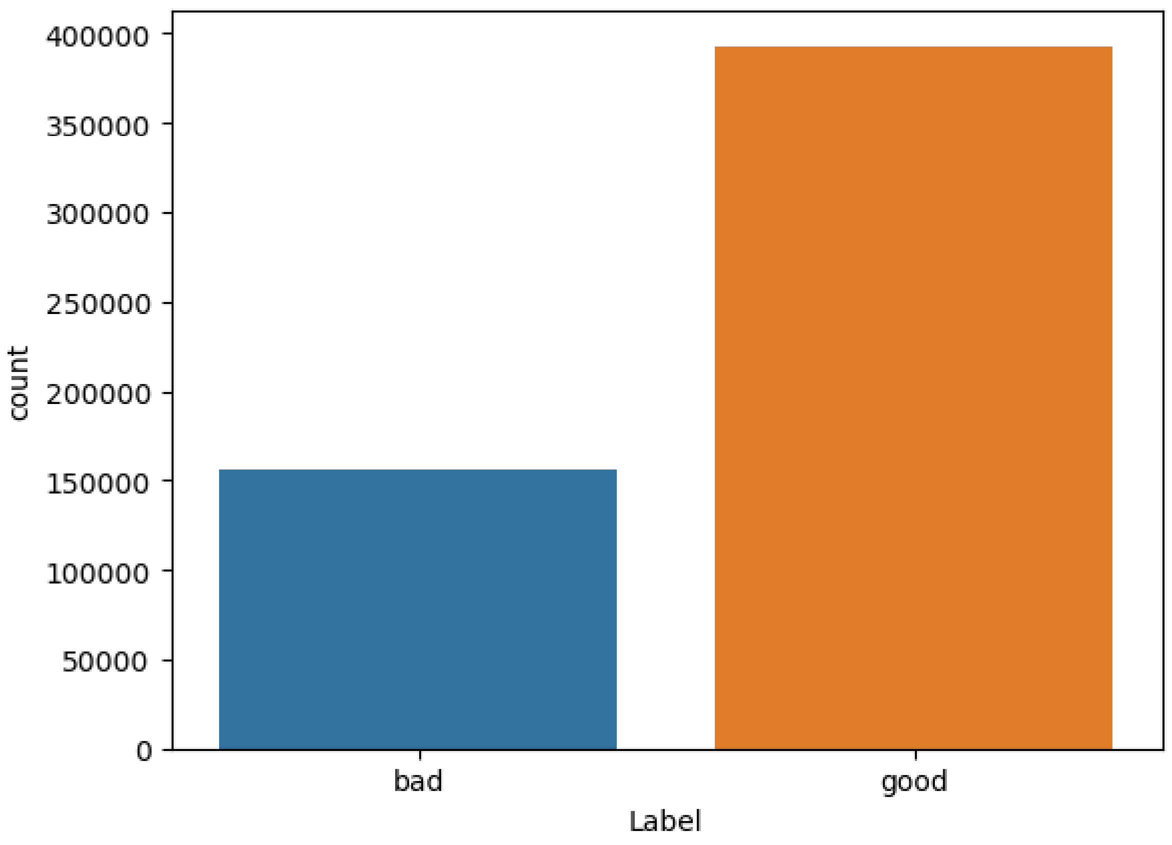
Out[8]:
```
(549346, 2)
```

In [9]:
```
df.isnull().sum()
```

Out[9]:
```
URL      0
Label    0
dtype: int64
```

In [10]:
```
sns.countplot(x="Label",data=df)
```

Out[10]:
```
<AxesSubplot:xlabel='Label', ylabel='count'>
```



In [11]:
```
tokenizer = RegexpTokenizer(r'[A-Za-z]+')
```

In [12]:
```
tokenizer.tokenize(df.URL[0])
```

```
Out[12]: ['nobell',
 'it',
 'ffb',
 'd',
 'dca',
 'cce',
 'f',
 'login',
 'SkyPe',
 'com',
 'en',
 'cgi',
 'bin',
 'verification',
 'login',
 'ffb',
 'd',
 'dca',
 'cce',
 'f',
 'index',
 'php',
 'cmd',
 'profile',
 'ach',
 'outdated',
 'page',
 'tmpl',
 'p',
 'gen',
 'failed',
 'to',
 'load',
 'nav',
 'login',
 'access']
```

```
In [20]: print('Getting words tokenized ...')
t0= time.perf_counter()
df['text_tokenized'] = df.URL.map(lambda t: tokenizer.tokenize(t))
t1 = time.perf_counter() - t0
print('Time taken',t1 ,'sec')
```

```
Getting words tokenized ...
Time taken 2.6101871999999844 sec
```

```
In [14]: df.sample(10)
```

Out[14]:

	URL	Label	text_tokenized
178680	en.wikipedia.org/wiki/NTV_(Newport_Television)	good	[en, wikipedia, org, wiki, NTV, Newport, Telev...
75819	www.tutorialized.com/tutorials/Java/1	good	[www, tutorialized, com, tutorials, Java]
271898	allisonkimball.com/simple_testimony/david-a-be...	good	[allisonkimball, com, simple, testimony, david...
406299	northatlanticbooks.com/category/martial/brucelee/	good	[northatlanticbooks, com, category, martial, b...
266644	abcpaydayloan.com/	good	[abcpaydayloan, com]
269246	acmepackingcompany.com/2011/4/19/2121317/the-2...	good	[acmepackingcompany, com, the, green, bay, pac...
317460	duke.edu/~tmc/motherpage/albums_prod/alb-phili...	good	[duke, edu, tmc, motherpage, albums, prod, alb...
297892	campbellsoup.com/	good	[campbellsoup, com]
92566	www.freewebs.com/keepersofultramar/	good	[www, freewebs, com, keepersofultramar]
50066	www.tommyvideo.com/catalog/customer/	good	[www, tommyvideo, com, catalog, customer]

In [18]:
```
stemmer = SnowballStemmer("english")
```

In [19]:
```
print('Getting words stemmed ...')
t0= time.perf_counter()
df['text_stemmed'] = df['text_tokenized'].map(lambda l: [stemmer.stem(word) for wor
t1= time.perf_counter() - t0
print('Time taken',t1 ,'sec')
```

```
Getting words stemmed ...
Time taken 51.8993911 sec
```

In [21]:
```
df.sample(10)
```

Out[21]:

	URL	Label	text_tokenized	tex
290323	bennetlaw.com/about-us/attorneys/robert-a-silv...	good	[bennetlaw, com, about, us, attorneys, robert,...	[bennetlaw, us, attorn
374475	lindenhills.coop/node/1118	good	[lindenhills, coop, node]	[lindenhil,
454721	ugo.com/girls/casey-mckinnon-1	good	[ugo, com, girls, casey, mckinnon]	[ugo, con
121013	constructionhugolafleur.com/file	bad	[constructionhugolafleur, com, file]	[construction
321097	elyrics.net/song/e/echo-hollow-lyrics.html	good	[elyrics, net, song, e, echo, hollow, lyrics, ...	[elyr, net, sc hollov
296866	businessweek.com/bschools/rankings/full_time_m...	good	[businessweek, com, bschools, rankings, full, ...	[busines bschool, rank
515314	91.239.24.168:6892	bad	[]	
278650	americanthinker.com/james_holmes/	good	[americanthinker, com, james, holmes]	[america
152736	boucherieabu.foodpages.ca/	good	[boucherieabu, foodpages, ca]	[boucherieat
6415	www.ctdi.cn/js/?us.battle.net/login/en/?ref=us...	bad	[www, ctdi, cn, js, us, battle, net, login, en...	[www, ct battl, ne

In [22]:
```
print('Get joiningwords ...')
t0= time.perf_counter()
df['text_sent'] = df['text_stemmed'].map(lambda l: ' '.join(l))
t1= time.perf_counter() - t0
print('Time taken',t1 ,'sec')
```

```
Get joiningwords ...
Time taken 0.2936014 sec
```

In [23]:
```
bad_sites = df[df.Label == 'bad']
good_sites = df[df.Label == 'good']
```

In [24]:
```
bad_sites.head()
```

Out[24]:

	URL	Label	text_tokenized	text_stemmed	
0	nobell.it/70ffb52d079109dca5664cce6f317373782/...	bad	[nobell, it, ffb, d, dca, cce, f, login, SkyPe...	[nobel, it, ffb, d, dca, cce, f, login, skype,...	no dca sky
1	www.dghjdgf.com/paypal.co.uk/cycgi-bin/webscrc...	bad	[www, dghjdgf, com, paypal, co, uk, cycgi, bin...	[www, dghjdgf, com, paypal, co, uk, cycgi, bin...	ww com ul
2	serviciosbys.com/paypal.cgi.bin.get-into.herf....	bad	[serviciosbys, com, paypal, cgi, bin, get, int...	[serviciosbi, com, paypal, cgi, bin, get, into...	servic pay get ir
3	mail.printakid.com/www.online.americanexpress....	bad	[mail, printakid, com, www, online, americanex...	[mail, printakid, com, www, onlin, americanexp...	mai com \ americ
4	thewhiskeydregs.com/wp-content/themes/widescre...	bad	[thewhiskeydregs, com, wp, content, themes, wi...	[thewhiskeydreg, com, wp, content, theme, wide...	thewh com w wi

```
In [25]: good_sites.head()
```

Out[25]:

	URL	Label	text_tokenized	text_stemmed	t
18231	esxcc.com/js/index.htm?us.battle.net/noghn/en/...	good	[esxcc, com, js, index, htm, us, battle, net, ...	[esxcc, com, js, index, htm, us, battl, net, n...	e r
18232	www□eira¯&nvinip¿ncH¯wVö%ÆåyDaHðû/ ÏyEùu□Ë\nÓ□6...	good	[www, eira, nvinip, ncH, wV, yDaH, yE, u, rT, ...	[www, eira, nvinip, nch, wv, ydah, ye, u, rt, ...	n wv u
18233	'www.institutocgr.coo/web/media/syqvem/dk-□óij...	good	[www, institutocgr, coo, web, media, syqvem, d...	[www, institutocgr, coo, web, media, syqvem, d...	ins sy
18234	□□Yìê□🡅koãÕ»Î§DéÎ□l½ñ¡ââqtò‚/à; Í	good	[Y, ko, D, l, qt]	[y, ko, d, l, qt]	y
18236	ruta89fm.com/images/AS@Vies/1i75cf7b16vc<F□d16...	good	[ruta, fm, com, images, AS, Vies, i, cf, b, vc...	[ruta, fm, com, imag, as, vie, i, cf, b, vc, f...	c as

```
In [26]: df.head()
```

Out[26]:

	URL	Label	text_tokenized	text_stemmed	
0	nobell.it/70ffb52d079109dca5664cce6f317373782/...	bad	[nobell, it, ffb, d, dca, cce, f, login, SkyPe...	[nobel, it, ffb, d, dca, cce, f, login, skype,...	no dca sky
1	www.dghjdgf.com/paypal.co.uk/cycgi-bin/webscrc...	bad	[www, dghjdgf, com, paypal, co, uk, cycgi, bin...	[www, dghjdgf, com, paypal, co, uk, cycgi, bin...	ww com uk
2	serviciosbys.com/paypal.cgi.bin.get-into.herf....	bad	[serviciosbys, com, paypal, cgi, bin, get, int...	[serviciosbi, com, paypal, cgi, bin, get, into...	servic pay get ir
3	mail.printakid.com/www.online.americanexpress....	bad	[mail, printakid, com, www, online, americanex...	[mail, printakid, com, www, onlin, americanexp...	mai com ameri
4	thewhiskeydregs.com/wp-content/themes/widescre...	bad	[thewhiskeydregs, com, wp, content, themes, wi...	[thewhiskeydreg, com, wp, content, theme, wide...	thewh com w wi

In [27]:
```python
cv = CountVectorizer()
feature = cv.fit_transform(df.text_sent)
feature[:5].toarray()
```

Out[27]:
```
array([[0, 0, 0, ..., 0, 0, 0],
       [0, 0, 0, ..., 0, 0, 0],
       [0, 0, 0, ..., 0, 0, 0],
       [0, 0, 0, ..., 0, 0, 0],
       [0, 0, 0, ..., 0, 0, 0]], dtype=int64)
```

In [28]:
```python
from sklearn.model_selection import train_test_split
trainX, testX, trainY, testY = train_test_split(feature, df.Label)
```

In [29]:
```python
from sklearn.linear_model import LogisticRegression
lr = LogisticRegression()
lr.fit(trainX,trainY)
```

```
C:\Users\shrey\anaconda3\lib\site-packages\sklearn\linear_model\_logistic.py:814:
ConvergenceWarning: lbfgs failed to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression
  n_iter_i = _check_optimize_result(
```

Out[29]:
```
LogisticRegression()
```

In [30]:
```python
lr.score(testX,testY)
```

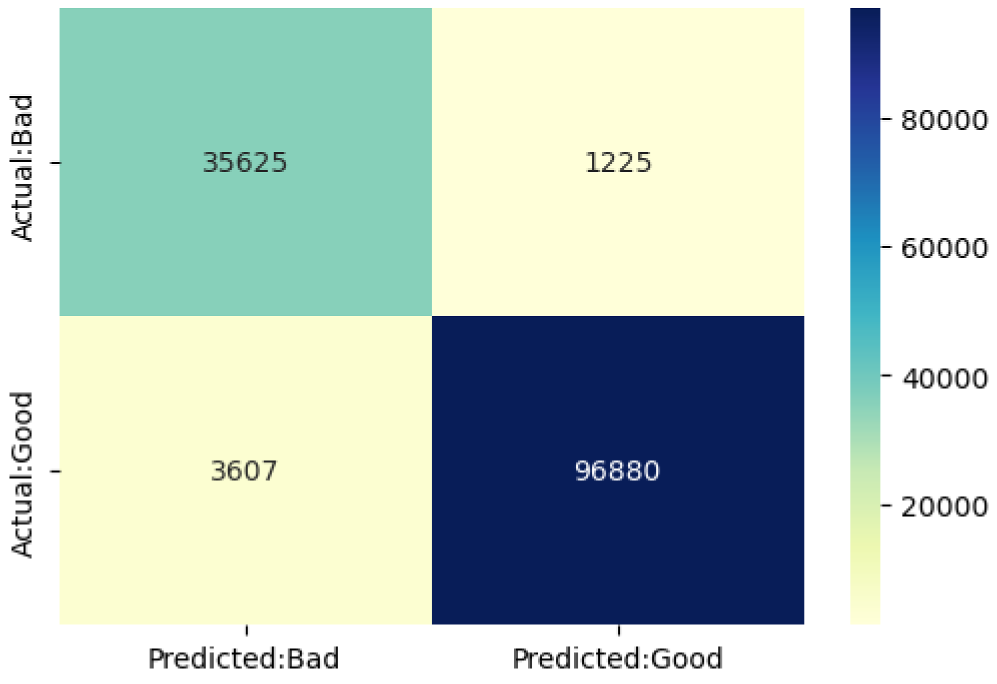
Out[30]:
```
0.9648164733465854
```

In [31]:
```python
Scores_ml = {}
Scores_ml['Logistic Regression'] = np.round(lr.score(testX,testY),2)
```

In [32]:
```
print('Training Accuracy :',lr.score(trainX,trainY))
print('Testing Accuracy :',lr.score(testX,testY))
con_mat = pd.DataFrame(confusion_matrix(lr.predict(testX), testY),
            columns = ['Predicted:Bad', 'Predicted:Good'],
            index = ['Actual:Bad', 'Actual:Good'])


print('\nCLASSIFICATION REPORT\n')
print(classification_report(lr.predict(testX), testY,
                            target_names =['Bad','Good']))

print('\nCONFUSION MATRIX')
plt.figure(figsize= (6,4))
sns.heatmap(con_mat, annot = True,fmt='d',cmap="YlGnBu")
```

```
Training Accuracy : 0.9786630874568274
Testing Accuracy : 0.9648164733465854

CLASSIFICATION REPORT

              precision    recall  f1-score   support

         Bad       0.91      0.97      0.94     36850
        Good       0.99      0.96      0.98    100487

    accuracy                           0.96    137337
   macro avg       0.95      0.97      0.96    137337
weighted avg       0.97      0.96      0.97    137337


CONFUSION MATRIX
```

Out[32]:
```
<AxesSubplot:>
```



In [33]:
```
from sklearn.naive_bayes import MultinomialNB
mnb = MultinomialNB()
mnb.fit(trainX,trainY)
```

Out[33]:
```
MultinomialNB()
```

In [34]:
```
mnb.score(testX,testY)
```

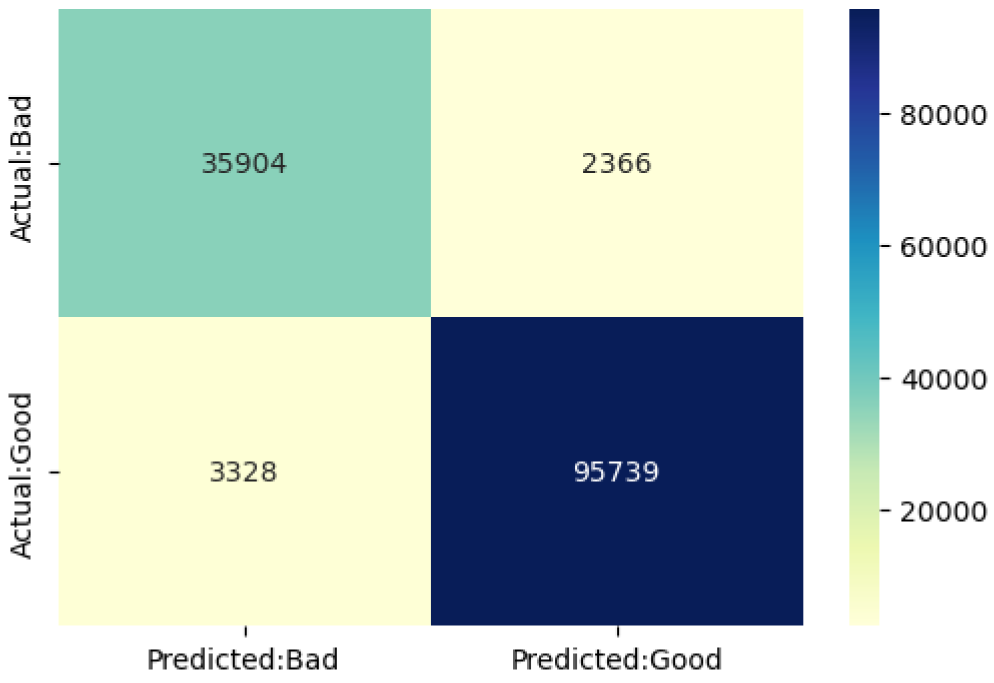
Out[34]:
```
0.9585399418947552
```

In [35]:
```
Scores_ml['MultinomialNB'] = np.round(mnb.score(testX,testY),2)
```

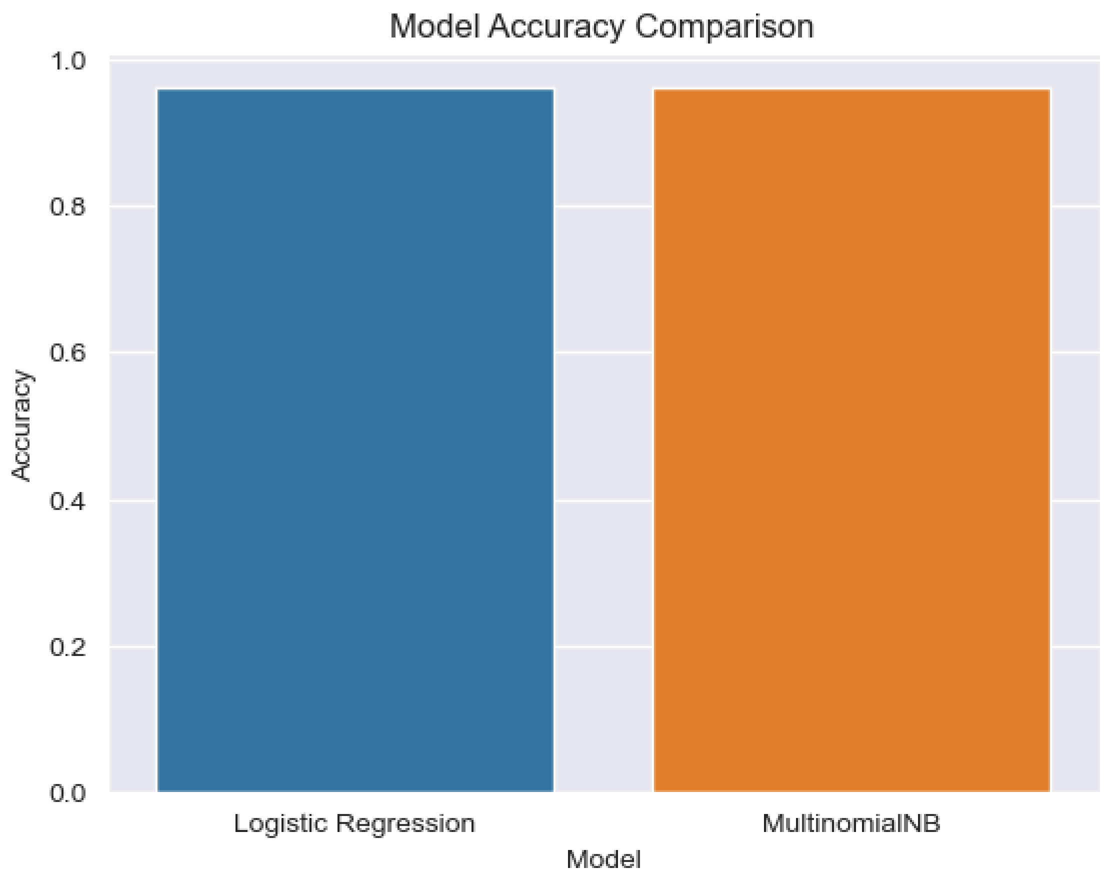
In [36]:
```
print('Training Accuracy :',mnb.score(trainX,trainY))
print('Testing Accuracy :',mnb.score(testX,testY))
con_mat = pd.DataFrame(confusion_matrix(mnb.predict(testX), testY),
            columns = ['Predicted:Bad', 'Predicted:Good'],
            index = ['Actual:Bad', 'Actual:Good'])


print('\nCLASSIFICATION REPORT\n')
print(classification_report(mnb.predict(testX), testY,
                            target_names =['Bad','Good']))

print('\nCONFUSION MATRIX')
plt.figure(figsize= (6,4))
sns.heatmap(con_mat, annot = True,fmt='d',cmap="YlGnBu")
```

```
Training Accuracy : 0.9741316330468509
Testing Accuracy : 0.9585399418947552

CLASSIFICATION REPORT

              precision    recall  f1-score   support

         Bad       0.92      0.94      0.93     38270
        Good       0.98      0.97      0.97     99067

    accuracy                           0.96    137337
   macro avg       0.95      0.95      0.95    137337
weighted avg       0.96      0.96      0.96    137337


CONFUSION MATRIX
```

Out[36]:
```
<AxesSubplot:>
```



In [37]:
```
acc = pd.DataFrame.from_dict(Scores_ml, orient='index', columns=['Accuracy'])

acc.reset_index(inplace=True)
acc.rename(columns={'index': 'Model'}, inplace=True)

sns.set_style('darkgrid')

sns.barplot(data=acc, x='Model', y='Accuracy')

plt.xlabel('Model')
plt.ylabel('Accuracy')
plt.title('Model Accuracy Comparison')


plt.show()
```



In [38]:
```
pipeline_ls = make_pipeline(CountVectorizer(tokenizer = RegexpTokenizer(r'[A-Za-z]
```

In [39]:
```
trainX, testX, trainY, testY = train_test_split(df.URL, df.Label)
pipeline_ls.fit(trainX,trainY)
```

```
C:\Users\shrey\anaconda3\lib\site-packages\sklearn\linear_model\_logistic.py:814:
ConvergenceWarning: lbfgs failed to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression
  n_iter_i = _check_optimize_result(
```

Out[39]:
```
Pipeline(steps=[('countvectorizer',
                 CountVectorizer(stop_words='english',
                                 tokenizer=<bound method RegexpTokenizer.tokenize
of RegexpTokenizer(pattern='[A-Za-z]+', gaps=False, discard_empty=True, flags=re.U
NICODE|re.MULTILINE|re.DOTALL)>)),
                ('logisticregression', LogisticRegression())])
```

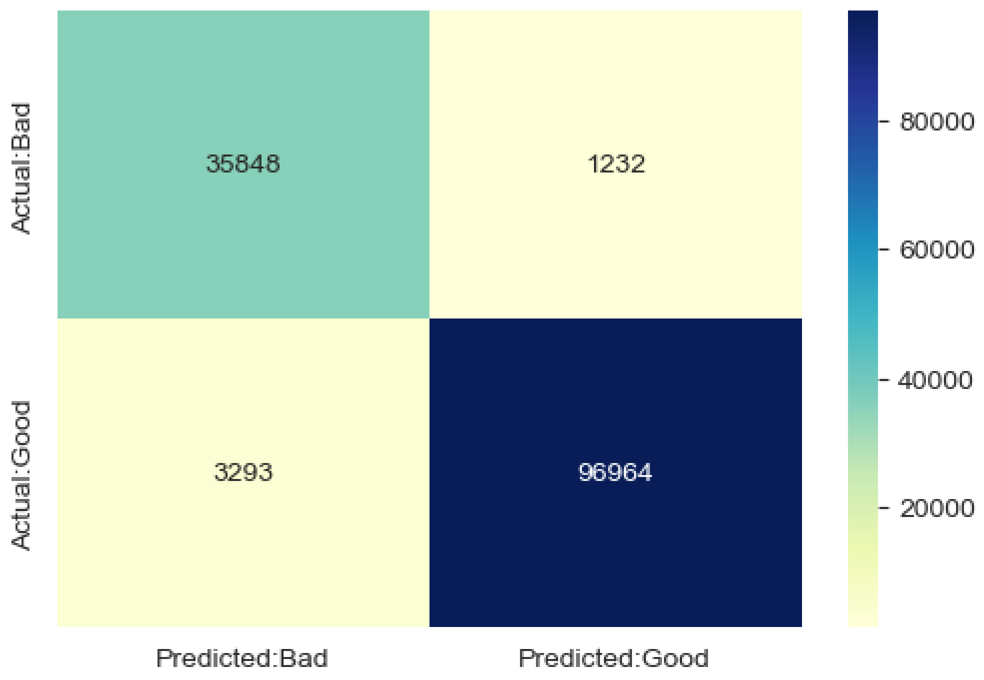
In [40]:
```
pipeline_ls.score(testX,testY)
```

Out[40]:
```
0.9670518505573881
```

In [41]:
```
print('Training Accuracy :',pipeline_ls.score(trainX,trainY))
print('Testing Accuracy :',pipeline_ls.score(testX,testY))
con_mat = pd.DataFrame(confusion_matrix(pipeline_ls.predict(testX), testY),
            columns = ['Predicted:Bad', 'Predicted:Good'],
            index = ['Actual:Bad', 'Actual:Good'])


print('\nCLASSIFICATION REPORT\n')
print(classification_report(pipeline_ls.predict(testX), testY,
                            target_names =['Bad','Good']))

print('\nCONFUSION MATRIX')
plt.figure(figsize= (6,4))
sns.heatmap(con_mat, annot = True,fmt='d',cmap="YlGnBu")
```

```
Training Accuracy : 0.9806096468766459
Testing Accuracy : 0.9670518505573881

CLASSIFICATION REPORT

              precision    recall  f1-score   support

         Bad       0.92      0.97      0.94     37080
        Good       0.99      0.97      0.98    100257

    accuracy                           0.97    137337
   macro avg       0.95      0.97      0.96    137337
weighted avg       0.97      0.97      0.97    137337


CONFUSION MATRIX
```

Out[41]:
```
<AxesSubplot:>
```



In [43]:
```
pickle.dump(pipeline_ls,open('phishing.pkl','wb'))
loaded_model = pickle.load(open('phishing.pkl', 'rb'))
result = loaded_model.score(testX,testY)
print(result)
```

```
0.9670518505573881
```

```
In [ ]:
```