In [1]:

```
import pandas as pd
```

```
/Users/harshil/anaconda3/lib/python3.10/site-packages/pandas/core/ar
rays/masked.py:60: UserWarning: Pandas requires version '1.3.6' or n
ewer of 'bottleneck' (version '1.3.5' currently installed).
  from pandas.core import (
/var/folders/4p/7vfgzzfn7q5dmq63pxmv3dlm0000gn/T/ipykernel_17219/408
0736814.py:1: DeprecationWarning:
Pyarrow will become a required dependency of pandas in the next majo
r release of pandas (pandas 3.0),
(to allow more performant data types, such as the Arrow string type,
and better interoperability with other libraries)
but was not found to be installed on your system.
If this would cause problems for you,
please provide us feedback at https://github.com/pandas-dev/pandas/i
ssues/54466 (https://github.com/pandas-dev/pandas/issues/54466)

  import pandas as pd
```

In [2]:

```
df = pd.read_csv('Cyber2_train.csv')
```

In [3]:

```
df.head()
```

Out[3]:

	url	category	label	ID
0	blackpast.org/?q=african-american-history-bibl...	good	1	196598
1	co8bo23vsd.mymazisocimowsed.com/nb9zatf4tk\nww...	bad	0	389728
2	lkis.or.id/845yfgh?riuoiuem=qwhxpkwlmho	bad	0	414140
3	51mct.com/js?ref=http://qszrysyus.battle.net/d3	bad	0	28193
4	beauty-plus.co.uk/tmp/https:/atendimento/chama...	bad	0	24091

In [4]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 269096 entries, 0 to 269095
Data columns (total 4 columns):
 #   Column    Non-Null Count   Dtype 
---  ------    --------------   ----- 
 0   url       269096 non-null  object
 1   category  269096 non-null  object
 2   label     269096 non-null  int64 
 3   ID        269096 non-null  int64 
dtypes: int64(2), object(2)
memory usage: 8.2+ MB
```

In [5]:
```
df['url']
```

Out[5]:
```
0         blackpast.org/?q=african-american-history-bibl...
1         co8bo23vsd.mymazisocimowsed.com/nb9zatf4tk\nww...
2                   lkis.or.id/845yfgh?riuoiuem=qwhxpkwlmho
3           51mct.com/js?ref=http://qszrysyus.battle.net/d3
4         beauty-plus.co.uk/tmp/https:/atendimento/chama...
                                ...
269091    baseballprospectus.com/player_search.php?searc...
269092     manta.com/c/mm31jpd/john-j-montefusco-associates
269093    articles.timesofindia.indiatimes.com/keyword/a...
269094      227-youtube-chili-nbc-tv-nba-news.blogspot.com/
269095    discogs.com/artist/Philippe+Wynne?anv=Philipp%...
Name: url, Length: 269096, dtype: object
```

In [6]:
```
X = df['url']
y=df['label']
```

In [7]:
```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X,y)
```

In [8]:
```
from sklearn.feature_extraction.text import CountVectorizer
v = CountVectorizer()
X_train_count = v.fit_transform(X_train.values)
X_train_count.toarray()[:2]
```

Out[8]:
```
array([[0, 0, 0, ..., 0, 0, 0],
       [0, 0, 0, ..., 0, 0, 0]])
```

In [9]:
```
# Used Naive Bayes Classifier

from sklearn.naive_bayes import MultinomialNB
model = MultinomialNB()
model.fit(X_train_count,y_train)
```

Out[9]:
```
▾ MultinomialNB
MultinomialNB()
```

In [10]:
```
X_test_count = v.transform(X_test)
```

In [11]:
```
model.score(X_test_count, y_test)
```

Out[11]:
```
0.9701221868775456
```

In [12]:
```
import pickle

# Save the models
with open('model.pkl', 'wb') as f:
    pickle.dump(model, f)

with open('vectorise.pkl', 'wb') as f:
    pickle.dump(v, f)
```

In [ ]:

```
```

In [ ]:

```
```

In [15]:

```
# Used Decision Tree Classifier

from sklearn.tree import DecisionTreeClassifier
model = DecisionTreeClassifier()
model.fit(X_train_count,y_train)
```

Out[15]:

```
▾ DecisionTreeClassifier
DecisionTreeClassifier()
```

In [16]:

```
X_test_count = v.transform(X_test)
model.score(X_test_count, y_test)
```

Out[16]:

```
0.9639831138329815
```

In [ ]:

```
```

In [ ]:

```
```

In [17]:

```
# Used K Nearest Neighbour Classifier

from sklearn.neighbors import KNeighborsClassifier
model = KNeighborsClassifier()
model.fit(X_train_count,y_train)
```

Out[17]:

```
▾ KNeighborsClassifier
KNeighborsClassifier()
```

In [18]:

```
X_test_count = v.transform(X_test)
model.score(X_test_count, y_test)
```

Out[18]:

```
0.9447632071825668
```

In [ ]:

```
```

In [ ]:

```
```

In [19]:

```
# Used XG Boost Classifier

from xgboost import XGBClassifier
model = XGBClassifier()
model.fit(X_train_count, y_train)
```

Out[19]:

```
▾                                    XGBClassifier

XGBClassifier(base_score=None, booster=None, callbacks=None,
              colsample_bylevel=None, colsample_bynode=None,
              colsample_bytree=None, device=None, early_stopping_rou
nds=None,
              enable_categorical=False, eval_metric=None, feature_ty
pes=None,
              gamma=None, grow_policy=None, importance_type=None,
              interaction_constraints=None, learning_rate=None, max_
bin=None,
              max_cat_threshold=None, max_cat_to_onehot=None,
```

In [20]:

```
X_test_count = v.transform(X_test)
model.score(X_test_count, y_test)
```

Out[20]:

```
0.9353836549038261
```

In [ ]:

In [ ]:

In [*]:

```
# Used SVM

from sklearn.svm import SVC
model = SVC()
model.fit(X_train_count, y_train)
```

In [*]:

```
X_test_count = v.transform(X_test)
model.score(X_test_count, y_test)
```

In [ ]: