Machine Learning Malicious URL Detection

In [1]:
```
# import necessary libraries
import pandas as pd
```

In [2]:
```
df = pd.read_csv('urldata.csv')
df.head()
```

Out[2]:

	url	label
0	clicks.careerbits.com/lt/click/8E04747359F18A2...	bad
1	adeccona.com	bad
2	links.email.informedamericantoday.com/u/click?...	bad
3	crew@email.informedamericantoday.com	bad
4	x9wystizllxtpcj3xnvd@d7vsnutmv9sfhfrrayqxm6m9w...	bad

In [3]:
```
# cleaning data removing unnecessary characters in the text data, punctuations,
# and repetitive words.
def makeTokens(f):
    tkns_BySlash = str(f.encode('utf-8')).split('/') # make tokens after splitting by slash
    total_Tokens = []

    for i in tkns_BySlash:
        tokens = str(i).split('-') # make tokens after splitting by dash
        tkns_ByDot = []

    for j in range(0,len(tokens)):
        temp_Tokens = str(tokens[j]).split('.') # make tokens after splitting by dot
        tkns_ByDot = tkns_ByDot + temp_Tokens
        total_Tokens = total_Tokens + tokens + tkns_ByDot
        total_Tokens = list(set(total_Tokens))  #remove redundant tokens

    if 'com' in total_Tokens:
        total_Tokens.remove('com') # removing .com since it occurs a lot of times and it should
    return total_Tokens
```

In [4]:
```
# add features and labels
df_list = df['url']
y = df['label']
```

In [5]:
```
# import sklearn libraries
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import TfidfVectorizer
```

In [6]:
```
# convert the text data into vectors of numbers
vectorizer = TfidfVectorizer(tokenizer=makeTokens)
```

In [7]:
```
X = vectorizer.fit_transform(df_list)
```

In [8]:
```
# Spliting the dataset into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

In [9]:
```
# Building Logistic Regression Model
logit = LogisticRegression(C=1.0, class_weight=None,
                           dual=False, fit_intercept=True, intercept_scaling=1,
                           max_iter=100, multi_class='ovr', n_jobs=1, penalty='l2',
                           random_state=None, solver='liblinear',
                           tol=0.0001, verbose=0, warm_start=False)

# fitting algorithm
logit.fit(X_train, y_train)
```

Out[9]:
```
LogisticRegression(multi_class='ovr', n_jobs=1, solver='liblinear')
```

In [10]:
```
# Get the model accuracy
print("Accuracy: ",logit.score(X_test, y_test))
```

```
Accuracy:  0.8975206611570248
```

In [11]:
```
# have an accuracy of ~90%

# First test of urls
X_predict = ["https://www.section.io/engineering-education/",
             "https://www.youtube.com/",
             "https://www.traversymedia.com/",
             "https://www.kleinehundezuhause.com",
             "http://ttps://www.mecymiafinance.com",
             "https://www.atlanticoceanicoilandgas.com",
             "www.supersecretphishingwebsite.golf"]
```

In [12]:
```
X_predict = vectorizer.transform(X_predict)
New_predict = logit.predict(X_predict)
```

In [13]:
```
print(New_predict)
```

```
['good' 'good' 'good' 'bad' 'bad' 'bad' 'bad']
```

In [14]:
```
# Second test of urls
X_predict1 = ["www.buyfakebillsonlinee.blogspot.com",
              "www.unitedairlineslogistics.com",
              "www.stonehousedelivery.com",
              "www.silkroadmeds-onlinepharmacy.com",
              "www.pendims.golf"]
```

In [15]:
```
X_predict1 = vectorizer.transform(X_predict1)
New_predict1 = logit.predict(X_predict1)
print(New_predict1)
```

```
['bad' 'bad' 'bad' 'bad' 'bad']
```