

In [1]:

```
#!/usr/bin/env python
```

```
import pandas as pd
import numpy as np
import random
from sklearn.feature_extraction
n.text import CountVectorizer
from sklearn.feature_extraction
n.text import TfidfVectorizer
from sklearn.linear_model impo
rt LogisticRegression
from sklearn.model_selection i
mport train_test_split
```

In [2]:

```
# Reading data from csv file
data = pd.read_csv("~/datascie
nce/data.csv")
data.head()
```

Out[2]:

	url	label
0	diaryofagameaddict.com	bad
1	espdesign.com.au	bad
2	iamagameaddict.com	bad

Out[2]:

	url	label
0	diaryofagameaddict.com	bad
1	espdesign.com.au	bad
2	iamagameaddict.com	bad
3	kalantzis.net	bad
4	slightlyoffcenter.net	bad

In [3]:

```
# Labels
y = data["label"]

# Features
url_list = data["url"]
```

In [4]:

```
# Using Tokenizer
vectorizer = TfidfVectorizer()

# Store vectors into X variable as Our XFeatures
X = vectorizer.fit_transform(url_list)
```

In [5]:

In [4]:

```
# Using Tokenizer
vectorizer = TfidfVectorizer()

# Store vectors into X variable as Our XFeatures
X = vectorizer.fit_transform(url_list)
```

In [5]:

```
# Split into training and testing dataset 80:20 ratio
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

In [6]:

```
# Model Building using logistic regression
logit = LogisticRegression()
logit.fit(X_train, y_train)
```

Out[6]:

```
LogisticRegression(C=1.0, class_weight=None, dual=False, fit_intercept=True,
                    intercept_scaling=1,
                    max_iter=100, multi_class='ov
r' n_jobs=1
```



```
st = train_test_split(X, y, test_size=0.2, random_state=42)
```

In [6]:

```
# Model Building using logistic regression  
logit = LogisticRegression()  
logit.fit(X_train, y_train)
```

Out[6]:

```
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)
```

In [7]:

```
# Accuracy of Our Model  
print("Accuracy of our model is: ", logit.score(X_test, y_test))
```

```
Accuracy of our model is: 0.964634392875
```