```
from flask import Flask, render_template
```

In [35]:

```
from flask import request
```

In [5]:

```
import numpy as np
import pandas as pd
from sklearn.linear_model import LogisticRegression
import pickle
import joblib
import inputScript
```

Reading Dataset

In [8]:

```
dataset = pd.read_csv("dataset.csv")
dataset = dataset.drop('index', 1)
dataset = dataset.dropna()
```

```
C:\Users\subhi\anaconda3\envs\summerproject\lib\site-
packages\ipykernel_launcher.py:2: FutureWarning: In a future version of
pandas all arguments of DataFrame.drop except for the argument 'labels'
will be keyword-only
```

In [9]:

```
x = dataset.iloc[ : , :-1].values
y = dataset.iloc[:, -1:].values

#spliting the dataset into training set and test set
from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test = train_test_split(x,y,test_size =
0.25, random_state =0 )
```

Developing Logistic Regression

In [10]:

```
#fitting logistic regression
classifier = LogisticRegression(random_state = 0)
classifier.fit(x_train, y_train)

#predicting the tests set result
y_pred = classifier.predict(x_test)
```

```
#confusion matrix
from sklearn.metrics import confusion_matrix
cm = confusion_matrix(y_test, y_pred)
print(cm)
```

```
[[1088  106]
 [ 100 1446]]
```

```
C:\Users\subhi\anaconda3\envs\summerproject\lib\site-
packages\sklearn\utils\validation.py:993: DataConversionWarning: A column-
vector y was passed when a 1d array was expected. Please change the shape
of y to (n_samples, ), for example using ravel().
  y = column_or_1d(y, warn=True)
```

In [11]:

```
print(y_pred)
```

```
[-1  1  1 ...  1 -1  1]
```

Saving Model

In [12]:

```
pickle.dump(classifier, open('model.pkl','wb'))
```

Integrating Flask with Model

In [48]:

```
app = Flask(__name__)
model = pickle.load(open('model.pkl','rb'))
@app.route('/')
def home():
    return render_template('home.html')

@app.route('/predict',methods=['POST'])
def predict():
    #For rendering results on HTML GUI
    int_features = request.form['url']
    print(int_features)
    checkprediction = inputScript.main(url)
    prediction = classifier.predict(checkprediction)
    print(prediction)
    result=""
    if(prediction==1):
        result="Url is not safe to enter"
    elif(prediction==-1):
        result="Url is safe to enter"
    return render_template('home.html', prediction_text= result)
```

```
if __name__ == '__main__':
    app.run()
```

```
 * Serving Flask app '__main__' (lazy loading)
 * Environment: production
   WARNING: This is a development server. Do not use it in a production deployment.
   Use a production WSGI server instead.
 * Debug mode: off

 * Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
127.0.0.1 - - [28/Oct/2022 19:31:29] "GET / HTTP/1.1" 200 -

https://www.binance-co.com/

127.0.0.1 - - [28/Oct/2022 19:31:49] "POST /predict HTTP/1.1" 200 -

module 'whois' has no attribute 'whois'
[[0, -1, 0, -1, 0, 1, -1, 0, 0, 0, 0, -1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]]
[1]
```

In [ ]:

In [ ]: