# Website Legitimacy Detection System using PyCaret

This project aims to classify websites as `benign`, `malicious`, or other potentially harmful categories based on URL patterns.

## Problem Statement

Malicious websites pose security risks to users. This system predicts if a website is malicious, aiding in identifying potential security threats

# Step 1: Import Necessary Libraries

In [1]:
```
# Install required libraries
!pip install pycaret joblib tldextract plotly

# Import libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import tldextract
import joblib
from pycaret.classification import *
from urllib.parse import urlparse
import plotly.express as px
```

```
WARNING: Ignoring invalid distribution -atplotlib (c:\users\nasru\anaconda3\envs\dat
a_science\lib\site-packages)
WARNING: Ignoring invalid distribution -atplotlib (c:\users\nasru\anaconda3\envs\dat
a_science\lib\site-packages)
WARNING: Ignoring invalid distribution -atplotlib (c:\users\nasru\anaconda3\envs\dat
a_science\lib\site-packages)
```

```
Requirement already satisfied: pycaret in c:\users\nasru\anaconda3\envs\data_science
\lib\site-packages (3.3.2)
Requirement already satisfied: joblib in c:\users\nasru\anaconda3\envs\data_science
\lib\site-packages (1.3.2)
Requirement already satisfied: tldextract in c:\users\nasru\anaconda3\envs\data_scie
nce\lib\site-packages (5.1.2)
Requirement already satisfied: plotly in c:\users\nasru\anaconda3\envs\data_science
\lib\site-packages (5.20.0)
Requirement already satisfied: ipython>=5.5.0 in c:\users\nasru\anaconda3\envs\data_
science\lib\site-packages (from pycaret) (8.27.0)
Requirement already satisfied: ipywidgets>=7.6.5 in c:\users\nasru\anaconda3\envs\da
ta_science\lib\site-packages (from pycaret) (8.1.2)
Requirement already satisfied: tqdm>=4.62.0 in c:\users\nasru\anaconda3\envs\data_sc
ience\lib\site-packages (from pycaret) (4.66.4)
Requirement already satisfied: numpy<1.27,>=1.21 in c:\users\nasru\anaconda3\envs\da
ta_science\lib\site-packages (from pycaret) (1.26.4)
Requirement already satisfied: pandas<2.2.0 in c:\users\nasru\anaconda3\envs\data_sc
ience\lib\site-packages (from pycaret) (2.1.4)
Requirement already satisfied: jinja2>=3 in c:\users\nasru\anaconda3\envs\data_scien
ce\lib\site-packages (from pycaret) (3.1.4)
Requirement already satisfied: scipy<=1.11.4,>=1.6.1 in c:\users\nasru\anaconda3\env
s\data_science\lib\site-packages (from pycaret) (1.11.4)
Requirement already satisfied: scikit-learn>1.4.0 in c:\users\nasru\anaconda3\envs\d
ata_science\lib\site-packages (from pycaret) (1.4.2)
Requirement already satisfied: pyod>=1.1.3 in c:\users\nasru\anaconda3\envs\data_sci
ence\lib\site-packages (from pycaret) (2.0.1)
Requirement already satisfied: imbalanced-learn>=0.12.0 in c:\users\nasru\anaconda3
\envs\data_science\lib\site-packages (from pycaret) (0.12.3)
Requirement already satisfied: category-encoders>=2.4.0 in c:\users\nasru\anaconda3
\envs\data_science\lib\site-packages (from pycaret) (2.6.3)
Requirement already satisfied: lightgbm>=3.0.0 in c:\users\nasru\anaconda3\envs\data
_science\lib\site-packages (from pycaret) (4.5.0)
Requirement already satisfied: numba>=0.55.0 in c:\users\nasru\anaconda3\envs\data_s
cience\lib\site-packages (from pycaret) (0.60.0)
Requirement already satisfied: requests>=2.27.1 in c:\users\nasru\anaconda3\envs\dat
a_science\lib\site-packages (from pycaret) (2.32.3)
Requirement already satisfied: psutil>=5.9.0 in c:\users\nasru\anaconda3\envs\data_s
cience\lib\site-packages (from pycaret) (5.9.0)
Requirement already satisfied: markupsafe>=2.0.1 in c:\users\nasru\anaconda3\envs\da
ta_science\lib\site-packages (from pycaret) (2.1.3)
Requirement already satisfied: importlib-metadata>=4.12.0 in c:\users\nasru\anaconda
3\envs\data_science\lib\site-packages (from pycaret) (7.2.1)
Requirement already satisfied: nbformat>=4.2.0 in c:\users\nasru\anaconda3\envs\data
_science\lib\site-packages (from pycaret) (5.10.4)
Requirement already satisfied: cloudpickle in c:\users\nasru\anaconda3\envs\data_sci
ence\lib\site-packages (from pycaret) (3.0.0)
Requirement already satisfied: deprecation>=2.1.0 in c:\users\nasru\anaconda3\envs\d
ata_science\lib\site-packages (from pycaret) (2.1.0)
Requirement already satisfied: xxhash in c:\users\nasru\anaconda3\envs\data_science
\lib\site-packages (from pycaret) (3.4.1)
Requirement already satisfied: matplotlib<3.8.0 in c:\users\nasru\anaconda3\envs\dat
a_science\lib\site-packages (from pycaret) (3.7.5)
Requirement already satisfied: scikit-plot>=0.3.7 in c:\users\nasru\anaconda3\envs\d
ata_science\lib\site-packages (from pycaret) (0.3.7)
Requirement already satisfied: yellowbrick>=1.4 in c:\users\nasru\anaconda3\envs\dat
a_science\lib\site-packages (from pycaret) (1.5)
```

```
Requirement already satisfied: kaleido>=0.2.1 in c:\users\nasru\anaconda3\envs\data_
science\lib\site-packages (from pycaret) (0.2.1)
Requirement already satisfied: schemdraw==0.15 in c:\users\nasru\anaconda3\envs\data
_science\lib\site-packages (from pycaret) (0.15)
Requirement already satisfied: plotly-resampler>=0.8.3.1 in c:\users\nasru\anaconda3
\envs\data_science\lib\site-packages (from pycaret) (0.10.0)
Requirement already satisfied: statsmodels>=0.12.1 in c:\users\nasru\anaconda3\envs
\data_science\lib\site-packages (from pycaret) (0.14.2)
Requirement already satisfied: sktime==0.26.0 in c:\users\nasru\anaconda3\envs\data_
science\lib\site-packages (from pycaret) (0.26.0)
Requirement already satisfied: tbats>=1.1.3 in c:\users\nasru\anaconda3\envs\data_sc
ience\lib\site-packages (from pycaret) (1.1.3)
Requirement already satisfied: pmdarima>=2.0.4 in c:\users\nasru\anaconda3\envs\data
_science\lib\site-packages (from pycaret) (2.0.4)
Requirement already satisfied: packaging in c:\users\nasru\anaconda3\envs\data_scien
ce\lib\site-packages (from sktime==0.26.0->pycaret) (24.1)
Requirement already satisfied: scikit-base<0.8.0 in c:\users\nasru\anaconda3\envs\da
ta_science\lib\site-packages (from sktime==0.26.0->pycaret) (0.7.8)
Requirement already satisfied: idna in c:\users\nasru\anaconda3\envs\data_science\li
b\site-packages (from tldextract) (3.7)
Requirement already satisfied: requests-file>=1.4 in c:\users\nasru\anaconda3\envs\d
ata_science\lib\site-packages (from tldextract) (2.1.0)
Requirement already satisfied: filelock>=3.0.8 in c:\users\nasru\anaconda3\envs\data
_science\lib\site-packages (from tldextract) (3.16.1)
Requirement already satisfied: tenacity>=6.2.0 in c:\users\nasru\anaconda3\envs\data
_science\lib\site-packages (from plotly) (9.0.0)
Requirement already satisfied: patsy>=0.5.1 in c:\users\nasru\anaconda3\envs\data_sc
ience\lib\site-packages (from category-encoders>=2.4.0->pycaret) (0.5.6)
Requirement already satisfied: threadpoolctl>=2.0.0 in c:\users\nasru\anaconda3\envs
\data_science\lib\site-packages (from imbalanced-learn>=0.12.0->pycaret) (3.5.0)
Requirement already satisfied: zipp>=0.5 in c:\users\nasru\anaconda3\envs\data_scien
ce\lib\site-packages (from importlib-metadata>=4.12.0->pycaret) (3.19.2)
Requirement already satisfied: decorator in c:\users\nasru\anaconda3\envs\data_scien
ce\lib\site-packages (from ipython>=5.5.0->pycaret) (5.1.1)
Requirement already satisfied: jedi>=0.16 in c:\users\nasru\anaconda3\envs\data_scie
nce\lib\site-packages (from ipython>=5.5.0->pycaret) (0.19.1)
Requirement already satisfied: matplotlib-inline in c:\users\nasru\anaconda3\envs\da
ta_science\lib\site-packages (from ipython>=5.5.0->pycaret) (0.1.6)
Requirement already satisfied: prompt-toolkit<3.1.0,>=3.0.41 in c:\users\nasru\anaco
nda3\envs\data_science\lib\site-packages (from ipython>=5.5.0->pycaret) (3.0.43)
Requirement already satisfied: pygments>=2.4.0 in c:\users\nasru\anaconda3\envs\data
_science\lib\site-packages (from ipython>=5.5.0->pycaret) (2.15.1)
Requirement already satisfied: stack-data in c:\users\nasru\anaconda3\envs\data_scie
nce\lib\site-packages (from ipython>=5.5.0->pycaret) (0.2.0)
Requirement already satisfied: traitlets>=5.13.0 in c:\users\nasru\anaconda3\envs\da
ta_science\lib\site-packages (from ipython>=5.5.0->pycaret) (5.14.3)
Requirement already satisfied: exceptiongroup in c:\users\nasru\anaconda3\envs\data_
science\lib\site-packages (from ipython>=5.5.0->pycaret) (1.2.0)
Requirement already satisfied: typing-extensions>=4.6 in c:\users\nasru\anaconda3\en
vs\data_science\lib\site-packages (from ipython>=5.5.0->pycaret) (4.11.0)
Requirement already satisfied: colorama in c:\users\nasru\anaconda3\envs\data_scienc
e\lib\site-packages (from ipython>=5.5.0->pycaret) (0.4.6)
Requirement already satisfied: comm>=0.1.3 in c:\users\nasru\anaconda3\envs\data_sci
ence\lib\site-packages (from ipywidgets>=7.6.5->pycaret) (0.2.1)
Requirement already satisfied: widgetsnbextension~=4.0.10 in c:\users\nasru\anaconda
3\envs\data_science\lib\site-packages (from ipywidgets>=7.6.5->pycaret) (4.0.10)
```

```
Requirement already satisfied: jupyterlab-widgets~=3.0.10 in c:\users\nasru\anaconda
3\envs\data_science\lib\site-packages (from ipywidgets>=7.6.5->pycaret) (3.0.10)
Requirement already satisfied: contourpy>=1.0.1 in c:\users\nasru\anaconda3\envs\dat
a_science\lib\site-packages (from matplotlib<3.8.0->pycaret) (1.2.0)
Requirement already satisfied: cycler>=0.10 in c:\users\nasru\anaconda3\envs\data_sc
ience\lib\site-packages (from matplotlib<3.8.0->pycaret) (0.11.0)
Requirement already satisfied: fonttools>=4.22.0 in c:\users\nasru\anaconda3\envs\da
ta_science\lib\site-packages (from matplotlib<3.8.0->pycaret) (4.51.0)
Requirement already satisfied: kiwisolver>=1.0.1 in c:\users\nasru\anaconda3\envs\da
ta_science\lib\site-packages (from matplotlib<3.8.0->pycaret) (1.4.4)
Requirement already satisfied: pillow>=6.2.0 in c:\users\nasru\anaconda3\envs\data_s
cience\lib\site-packages (from matplotlib<3.8.0->pycaret) (10.4.0)
Requirement already satisfied: pyparsing>=2.3.1 in c:\users\nasru\anaconda3\envs\dat
a_science\lib\site-packages (from matplotlib<3.8.0->pycaret) (3.1.2)
Requirement already satisfied: python-dateutil>=2.7 in c:\users\nasru\anaconda3\envs
\data_science\lib\site-packages (from matplotlib<3.8.0->pycaret) (2.9.0.post0)
Requirement already satisfied: fastjsonschema>=2.15 in c:\users\nasru\anaconda3\envs
\data_science\lib\site-packages (from nbformat>=4.2.0->pycaret) (2.16.2)
Requirement already satisfied: jsonschema>=2.6 in c:\users\nasru\anaconda3\envs\data
_science\lib\site-packages (from nbformat>=4.2.0->pycaret) (4.23.0)
Requirement already satisfied: jupyter-core!=5.0.*,>=4.12 in c:\users\nasru\anaconda
3\envs\data_science\lib\site-packages (from nbformat>=4.2.0->pycaret) (5.7.2)
Requirement already satisfied: llvmlite<0.44,>=0.43.0dev0 in c:\users\nasru\anaconda
3\envs\data_science\lib\site-packages (from numba>=0.55.0->pycaret) (0.43.0)
Requirement already satisfied: pytz>=2020.1 in c:\users\nasru\anaconda3\envs\data_sc
ience\lib\site-packages (from pandas<2.2.0->pycaret) (2024.1)
Requirement already satisfied: tzdata>=2022.1 in c:\users\nasru\anaconda3\envs\data_
science\lib\site-packages (from pandas<2.2.0->pycaret) (2023.3)
Requirement already satisfied: dash>=2.9.0 in c:\users\nasru\anaconda3\envs\data_sci
ence\lib\site-packages (from plotly-resampler>=0.8.3.1->pycaret) (2.17.1)
Requirement already satisfied: orjson<4.0.0,>=3.8.0 in c:\users\nasru\anaconda3\envs
\data_science\lib\site-packages (from plotly-resampler>=0.8.3.1->pycaret) (3.10.6)
Requirement already satisfied: tsdownsample>=0.1.3 in c:\users\nasru\anaconda3\envs
\data_science\lib\site-packages (from plotly-resampler>=0.8.3.1->pycaret) (0.1.3)
Requirement already satisfied: Cython!=0.29.18,!=0.29.31,>=0.29 in c:\users\nasru\an
aconda3\envs\data_science\lib\site-packages (from pmdarima>=2.0.4->pycaret) (3.0.10)
Requirement already satisfied: urllib3 in c:\users\nasru\anaconda3\envs\data_science
\lib\site-packages (from pmdarima>=2.0.4->pycaret) (2.2.3)
Requirement already satisfied: setuptools!=50.0.0,>=38.6.0 in c:\users\nasru\anacond
a3\envs\data_science\lib\site-packages (from pmdarima>=2.0.4->pycaret) (75.1.0)
Requirement already satisfied: charset-normalizer<4,>=2 in c:\users\nasru\anaconda3
\envs\data_science\lib\site-packages (from requests>=2.27.1->pycaret) (3.3.2)
Requirement already satisfied: certifi>=2017.4.17 in c:\users\nasru\anaconda3\envs\d
ata_science\lib\site-packages (from requests>=2.27.1->pycaret) (2024.8.30)
Requirement already satisfied: Flask<3.1,>=1.0.4 in c:\users\nasru\anaconda3\envs\da
ta_science\lib\site-packages (from dash>=2.9.0->plotly-resampler>=0.8.3.1->pycaret)
(3.0.3)
Requirement already satisfied: Werkzeug<3.1 in c:\users\nasru\anaconda3\envs\data_sc
ience\lib\site-packages (from dash>=2.9.0->plotly-resampler>=0.8.3.1->pycaret) (3.0.
3)
Requirement already satisfied: dash-html-components==2.0.0 in c:\users\nasru\anacond
a3\envs\data_science\lib\site-packages (from dash>=2.9.0->plotly-resampler>=0.8.3.1-
>pycaret) (2.0.0)
Requirement already satisfied: dash-core-components==2.0.0 in c:\users\nasru\anacond
a3\envs\data_science\lib\site-packages (from dash>=2.9.0->plotly-resampler>=0.8.3.1-
>pycaret) (2.0.0)
```

```
Requirement already satisfied: dash-table==5.0.0 in c:\users\nasru\anaconda3\envs\da
ta_science\lib\site-packages (from dash>=2.9.0->plotly-resampler>=0.8.3.1->pycaret)
(5.0.0)
Requirement already satisfied: retrying in c:\users\nasru\anaconda3\envs\data_scienc
e\lib\site-packages (from dash>=2.9.0->plotly-resampler>=0.8.3.1->pycaret) (1.3.4)
Requirement already satisfied: nest-asyncio in c:\users\nasru\anaconda3\envs\data_sc
ience\lib\site-packages (from dash>=2.9.0->plotly-resampler>=0.8.3.1->pycaret) (1.6.
0)
Requirement already satisfied: parso<0.9.0,>=0.8.3 in c:\users\nasru\anaconda3\envs
\data_science\lib\site-packages (from jedi>=0.16->ipython>=5.5.0->pycaret) (0.8.3)
Requirement already satisfied: attrs>=22.2.0 in c:\users\nasru\anaconda3\envs\data_s
cience\lib\site-packages (from jsonschema>=2.6->nbformat>=4.2.0->pycaret) (24.2.0)
Requirement already satisfied: jsonschema-specifications>=2023.03.6 in c:\users\nasr
u\anaconda3\envs\data_science\lib\site-packages (from jsonschema>=2.6->nbformat>=4.
2.0->pycaret) (2023.7.1)
Requirement already satisfied: referencing>=0.28.4 in c:\users\nasru\anaconda3\envs
\data_science\lib\site-packages (from jsonschema>=2.6->nbformat>=4.2.0->pycaret) (0.
30.2)
Requirement already satisfied: rpds-py>=0.7.1 in c:\users\nasru\anaconda3\envs\data_
science\lib\site-packages (from jsonschema>=2.6->nbformat>=4.2.0->pycaret) (0.10.6)
Requirement already satisfied: platformdirs>=2.5 in c:\users\nasru\anaconda3\envs\da
ta_science\lib\site-packages (from jupyter-core!=5.0.*,>=4.12->nbformat>=4.2.0->pyca
ret) (3.10.0)
Requirement already satisfied: pywin32>=300 in c:\users\nasru\anaconda3\envs\data_sc
ience\lib\site-packages (from jupyter-core!=5.0.*,>=4.12->nbformat>=4.2.0->pycaret)
(305.1)
Requirement already satisfied: six in c:\users\nasru\anaconda3\envs\data_science\lib
\site-packages (from patsy>=0.5.1->category-encoders>=2.4.0->pycaret) (1.16.0)
Requirement already satisfied: wcwidth in c:\users\nasru\anaconda3\envs\data_science
\lib\site-packages (from prompt-toolkit<3.1.0,>=3.0.41->ipython>=5.5.0->pycaret) (0.
2.5)
Requirement already satisfied: executing in c:\users\nasru\anaconda3\envs\data_scien
ce\lib\site-packages (from stack-data->ipython>=5.5.0->pycaret) (0.8.3)
Requirement already satisfied: asttokens in c:\users\nasru\anaconda3\envs\data_scien
ce\lib\site-packages (from stack-data->ipython>=5.5.0->pycaret) (2.0.5)
Requirement already satisfied: pure-eval in c:\users\nasru\anaconda3\envs\data_scien
ce\lib\site-packages (from stack-data->ipython>=5.5.0->pycaret) (0.2.2)
Requirement already satisfied: itsdangerous>=2.1.2 in c:\users\nasru\anaconda3\envs
\data_science\lib\site-packages (from Flask<3.1,>=1.0.4->dash>=2.9.0->plotly-resampl
er>=0.8.3.1->pycaret) (2.2.0)
Requirement already satisfied: click>=8.1.3 in c:\users\nasru\anaconda3\envs\data_sc
ience\lib\site-packages (from Flask<3.1,>=1.0.4->dash>=2.9.0->plotly-resampler>=0.8.
3.1->pycaret) (8.1.7)
Requirement already satisfied: blinker>=1.6.2 in c:\users\nasru\anaconda3\envs\data_
science\lib\site-packages (from Flask<3.1,>=1.0.4->dash>=2.9.0->plotly-resampler>=0.
8.3.1->pycaret) (1.8.2)
```

# Step 2: Load Dataset

In [2]:
```
# Load data
data = pd.read_csv('malicious_phish.csv')
data.head()
```

Out[2]:

| | url | type |
|---|---|---|
| **0** | br-icloud.com.br | phishing |
| **1** | mp3raid.com/music/krizz_kaliko.html | benign |
| **2** | bopsecrets.org/rexroth/cr/1.htm | benign |
| **3** | http://www.garage-pirenne.be/index.php?option=... | defacement |
| **4** | http://adventure-nicaragua.net/index.php?optio... | defacement |

# Step 3: Exploratory Data Analysis (EDA)

## Dataset Information

In [3]:
```
data.info()
data.describe()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 651191 entries, 0 to 651190
Data columns (total 2 columns):
 #   Column  Non-Null Count   Dtype 
---  ------  --------------   ----- 
 0   url     651191 non-null  object
 1   type    651191 non-null  object
dtypes: object(2)
memory usage: 9.9+ MB
```

Out[3]:

| | url | type |
|---|---|---|
| **count** | 651191 | 651191 |
| **unique** | 641119 | 4 |
| **top** | http://style.org.hc360.com/css/detail/mysite/s... | benign |
| **freq** | 180 | 428103 |

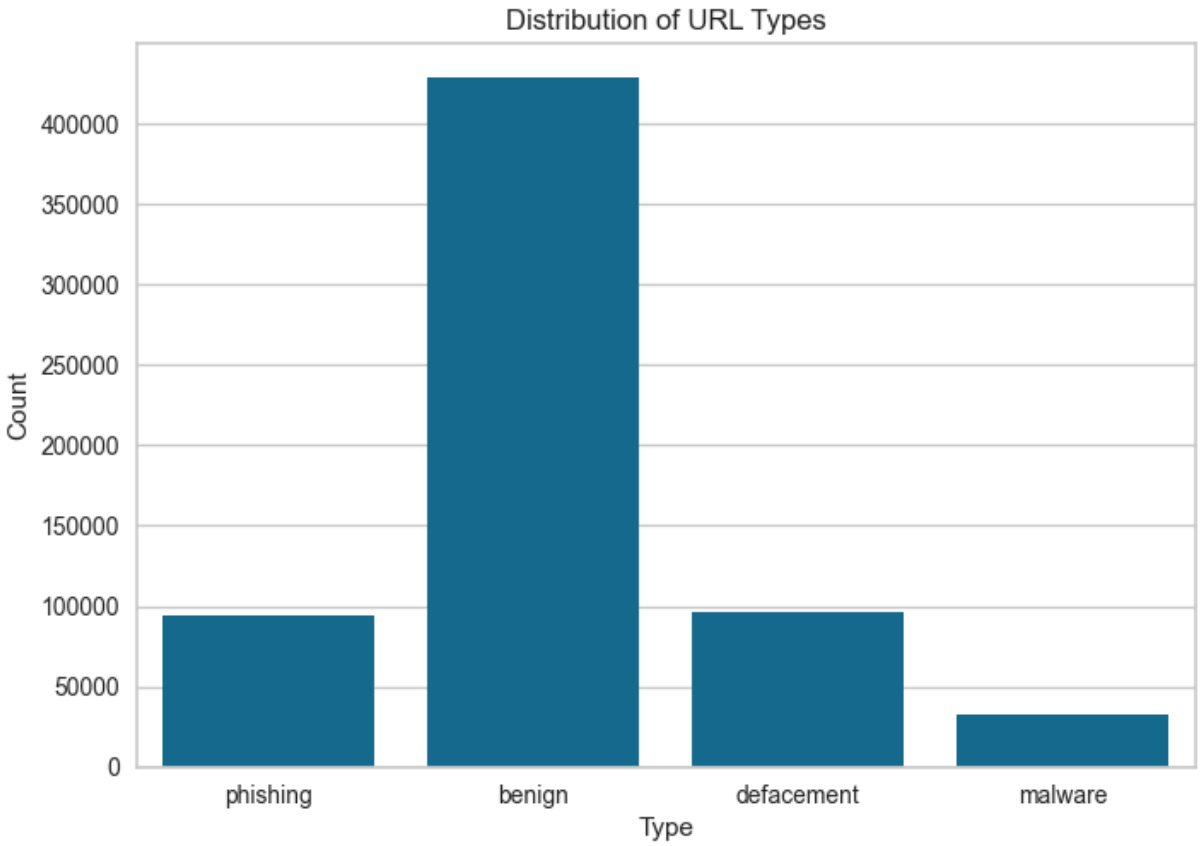
## Missing Values Check

In [4]:
```
data.isnull().sum()
```

Out[4]:
```
url     0
type    0
dtype: int64
```

## Class Distribution

In [5]:
```
sns.countplot(x='type', data=data)
plt.title('Distribution of URL Types')
plt.xlabel('Type')
plt.ylabel('Count')
plt.show()
```




## Step 4: Feature Engineering

## URL Length

In [6]:
```
data['url_length'] = data['url'].apply(lambda x: len(x))
```

## Domain Extraction

In [7]:
```
data['domain'] = data['url'].apply(lambda x: tldextract.extract(x).domain)
```

## Special Character Counts

In [8]:
```
def count_special_chars(url):
    return sum(1 for char in url if not char.isalnum())

data['special_char_count'] = data['url'].apply(count_special_chars)
```

## HTTPS Indicator

In [9]:
```
data['is_https'] = data['url'].apply(lambda x: 1 if urlparse(x).scheme == 'https' e
```

## Digit and Letter Counts

In [10]:
```
data['digit_count'] = data['url'].apply(lambda x: sum(c.isdigit() for c in x))
data['letter_count'] = data['url'].apply(lambda x: sum(c.isalpha() for c in x))
```

# Step 5: Data Preprocessing

## Encode Target Labels

- Map benign as 0, phishing as 1, defacement as 2, malware as 3.

In [11]:
```
type_mapping = {'benign': 0, 'phishing': 1, 'defacement': 2, 'malware': 3}
data['type'] = data['type'].map(type_mapping)
```

## Split Dataset

In [12]:
```
from sklearn.model_selection import train_test_split
X = data.drop(columns=['type', 'url', 'domain'])
y = data['type']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_sta
```

# Step 6: Modeling with PyCaret

## Initialize PyCaret Setup

In [22]:
```
clf = setup(data=data, target='type', session_id=123, log_experiment=False)
```

| | Description | Value |
|---|---|---|
| **0** | Session id | 123 |
| **1** | Target | type |
| **2** | Target type | Multiclass |
| **3** | Original data shape | (651191, 8) |
| **4** | Transformed data shape | (651191, 8) |
| **5** | Transformed train set shape | (455833, 8) |
| **6** | Transformed test set shape | (195358, 8) |
| **7** | Numeric features | 5 |
| **8** | Categorical features | 2 |
| **9** | Preprocess | True |
| **10** | Imputation type | simple |
| **11** | Numeric imputation | mean |
| **12** | Categorical imputation | mode |
| **13** | Maximum one-hot encoding | 25 |
| **14** | Encoding method | None |
| **15** | Fold Generator | StratifiedKFold |
| **16** | Fold Number | 10 |
| **17** | CPU Jobs | -1 |
| **18** | Use GPU | False |
| **19** | Log Experiment | False |
| **20** | Experiment Name | clf-default-name |
| **21** | USI | c3c0 |

# Compare Models

In [19]:
```
best_model = compare_models()
```

| | Model | Accuracy | AUC | Recall | Prec. | F1 | Kappa | MCC | TT (Sec) |
|---|---|---|---|---|---|---|---|---|---|
| **lda** | Linear Discriminant Analysis | 0.8231 | 0.0000 | 0.8231 | 0.7982 | 0.7790 | 0.6043 | 0.6411 | 1.1380 |
| **ridge** | Ridge Classifier | 0.7971 | 0.0000 | 0.7971 | 0.7801 | 0.7194 | 0.5237 | 0.5890 | 1.1320 |
| **knn** | K Neighbors Classifier | 0.7696 | 0.8716 | 0.7696 | 0.8644 | 0.7947 | 0.6179 | 0.6494 | 2.0970 |
| **svm** | SVM - Linear Kernel | 0.7620 | 0.0000 | 0.7620 | 0.8284 | 0.7659 | 0.5597 | 0.5785 | 9.9930 |
| **lr** | Logistic Regression | 0.7422 | 0.0000 | 0.7422 | 0.8555 | 0.7653 | 0.5791 | 0.6148 | 14.5080 |
| **dt** | Decision Tree Classifier | 0.6746 | 0.5266 | 0.6746 | 0.7256 | 0.5544 | 0.0778 | 0.2061 | 1.1550 |
| **dummy** | Dummy Classifier | 0.6574 | 0.5000 | 0.6574 | 0.4322 | 0.5215 | 0.0000 | 0.0000 | 1.1000 |
| **et** | Extra Trees Classifier | 0.2557 | 0.8530 | 0.2557 | 0.6635 | 0.1956 | 0.1202 | 0.3041 | 3.5940 |
| **lightgbm** | Light Gradient Boosting Machine | 0.1679 | 0.7294 | 0.1679 | 0.8323 | 0.0772 | 0.0250 | 0.1382 | 2.7860 |
| **rf** | Random Forest Classifier | 0.1620 | 0.8640 | 0.1620 | 0.8214 | 0.0650 | 0.0218 | 0.1395 | 4.2610 |
| **gbc** | Gradient Boosting Classifier | 0.1617 | 0.0000 | 0.1617 | 0.6471 | 0.0651 | 0.0113 | 0.1095 | 25.4290 |
| **ada** | Ada Boost Classifier | 0.1467 | 0.0000 | 0.1467 | 0.5331 | 0.0411 | 0.0025 | 0.0160 | 3.7050 |
| **nb** | Naive Bayes | 0.0521 | 0.5426 | 0.0521 | 0.7345 | 0.0091 | 0.0020 | 0.0403 | 1.1450 |
| **qda** | Quadratic Discriminant Analysis | 0.0501 | 0.0000 | 0.0501 | 0.6897 | 0.0050 | 0.0001 | 0.0074 | 1.2370 |

```
2024/11/01 00:05:34 WARNING mlflow.models.model: Input example should be provided to
infer model signature if the model signature is not provided when logging the model.
2024/11/01 00:05:35 WARNING mlflow.models.model: Input example should be provided to
infer model signature if the model signature is not provided when logging the model.
2024/11/01 00:05:35 WARNING mlflow.models.model: Input example should be provided to
infer model signature if the model signature is not provided when logging the model.
2024/11/01 00:05:36 WARNING mlflow.models.model: Input example should be provided to
infer model signature if the model signature is not provided when logging the model.
2024/11/01 00:05:36 WARNING mlflow.models.model: Input example should be provided to
infer model signature if the model signature is not provided when logging the model.
2024/11/01 00:05:37 WARNING mlflow.models.model: Input example should be provided to
infer model signature if the model signature is not provided when logging the model.
2024/11/01 00:05:38 WARNING mlflow.models.model: Input example should be provided to
infer model signature if the model signature is not provided when logging the model.
2024/11/01 00:05:38 WARNING mlflow.models.model: Input example should be provided to
infer model signature if the model signature is not provided when logging the model.
2024/11/01 00:05:39 WARNING mlflow.models.model: Input example should be provided to
infer model signature if the model signature is not provided when logging the model.
2024/11/01 00:05:39 WARNING mlflow.models.model: Input example should be provided to
infer model signature if the model signature is not provided when logging the model.
2024/11/01 00:05:40 WARNING mlflow.models.model: Input example should be provided to
infer model signature if the model signature is not provided when logging the model.
2024/11/01 00:05:41 WARNING mlflow.models.model: Input example should be provided to
infer model signature if the model signature is not provided when logging the model.
2024/11/01 00:05:41 WARNING mlflow.models.model: Input example should be provided to
infer model signature if the model signature is not provided when logging the model.
2024/11/01 00:05:42 WARNING mlflow.models.model: Input example should be provided to
infer model signature if the model signature is not provided when logging the model.
```

# Model Tuning

In [20]:
```
tuned_model = tune_model(best_model)
```

| Fold | Accuracy | AUC | Recall | Prec. | F1 | Kappa | MCC |
|---|---|---|---|---|---|---|---|
| **0** | 0.8227 | 0.0000 | 0.8227 | 0.7984 | 0.7785 | 0.6035 | 0.6400 |
| **1** | 0.8227 | 0.0000 | 0.8227 | 0.7983 | 0.7772 | 0.6030 | 0.6405 |
| **2** | 0.8238 | 0.0000 | 0.8238 | 0.8001 | 0.7802 | 0.6059 | 0.6429 |
| **3** | 0.8224 | 0.0000 | 0.8224 | 0.7972 | 0.7777 | 0.6032 | 0.6392 |
| **4** | 0.8233 | 0.0000 | 0.8233 | 0.7981 | 0.7797 | 0.6052 | 0.6414 |
| **5** | 0.8221 | 0.0000 | 0.8221 | 0.7953 | 0.7775 | 0.6020 | 0.6389 |
| **6** | 0.8229 | 0.0000 | 0.8229 | 0.7975 | 0.7789 | 0.6040 | 0.6407 |
| **7** | 0.8222 | 0.0000 | 0.8222 | 0.7974 | 0.7772 | 0.6019 | 0.6391 |
| **8** | 0.8230 | 0.0000 | 0.8230 | 0.7995 | 0.7790 | 0.6037 | 0.6409 |
| **9** | 0.8230 | 0.0000 | 0.8230 | 0.7977 | 0.7796 | 0.6044 | 0.6407 |
| **Mean** | 0.8228 | 0.0000 | 0.8228 | 0.7980 | 0.7786 | 0.6037 | 0.6404 |
| **Std** | 0.0005 | 0.0000 | 0.0005 | 0.0012 | 0.0010 | 0.0012 | 0.0012 |

```
Fitting 10 folds for each of 10 candidates, totalling 100 fits
Original model was better than the tuned model, hence it will be returned. NOTE: The display metrics are for the tuned model (not the original one).
```

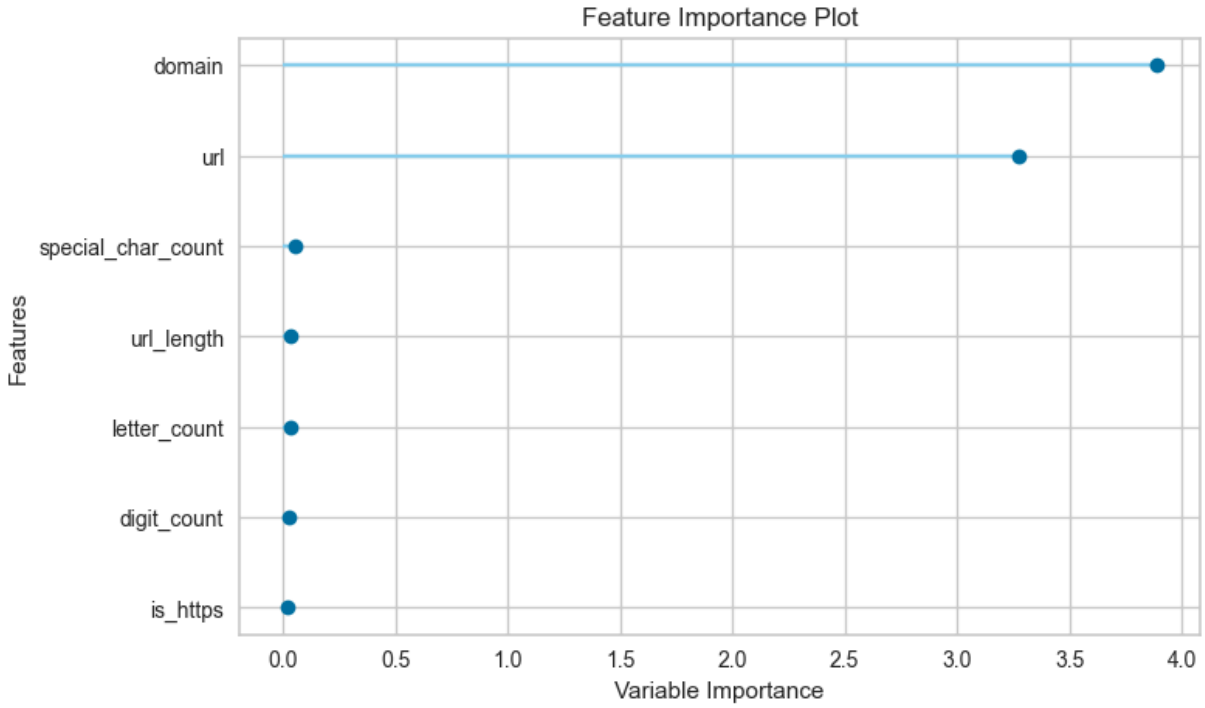
```
2024/11/01 00:07:51 WARNING mlflow.models.model: Input example should be provided to infer model signature if the model signature is not provided when logging the model.
```

# Evaluate Model

In [21]:
```
evaluate_model(tuned_model)
```

```
interactive(children=(ToggleButtons(description='Plot Type:', icons=('',), options=(('Pipeline Plot', 'pipelin…
```

In [53]:
```
# Display overall feature importance
plot_model(loaded_model, plot='feature')
```




## Finalize and Save Model

In [35]:
```
save_model(final_model, 'website_legitimacy_model')  # PyCaret will append .pkl aut
```

```
Transformation Pipeline and Model Successfully Saved
```

Out[35]:
```
(Pipeline(memory=Memory(location=None),
          steps=[('numerical_imputer',
                  TransformerWrapper(exclude=None,
                                     include=['url_length', 'special_char_count',
                                              'is_https', 'digit_count',
                                              'letter_count'],
                                     transformer=SimpleImputer(add_indicator=Fals
e,
                                                               copy=True,
                                                               fill_value=None,
                                                               keep_empty_features
=False,
                                                               missing_values=nan,
                                                               strategy='mean'))),
                 ('categorical_imputer',
                  Transform...
                                     transformer=TargetEncoder(cols=['url',
                                                                     'domain'],
                                                               drop_invariant=Fals
e,
                                                               handle_missing='ret
urn_nan',
                                                               handle_unknown='val
ue',
                                                               hierarchy=None,
                                                               min_samples_leaf=2
0,
                                                               return_df=True,
                                                               smoothing=10,
                                                               verbose=0))),
                 ('actual_estimator',
                  LinearDiscriminantAnalysis(covariance_estimator=None,
                                             n_components=None, priors=None,
                                             shrinkage=None, solver='svd',
                                             store_covariance=False,
                                             tol=0.0001))],
          verbose=False),
 'website_legitimacy_model.pkl')
```

# Step 7: Model Testing and Prediction

## Load Model for Future Predictions

In [14]:
```
from pycaret.classification import load_model, predict_model
import pandas as pd
from urllib.parse import urlparse

# Load the model using PyCaret
loaded_model = load_model('website_legitimacy_model')
```

```
Transformation Pipeline and Model Successfully Loaded
```

## Define Prediction Function

In [15]:
```python
# Define helper functions
def count_special_chars(url):
    return sum(1 for char in url if not char.isalnum())

# Define the prediction function
def predict_website_legitimacy(url, model):
    # Define the mapping for label interpretation
    label_mapping = {
        0: "benign",
        1: "phishing",
        2: "defacement",
        3: "malware"
    }

    # Prepare features for the URL with placeholders for expected columns
    features = {
        'url_length': len(url),
        'special_char_count': count_special_chars(url),
        'is_https': 1 if urlparse(url).scheme == 'https' else 0,
        'digit_count': sum(c.isdigit() for c in url),
        'letter_count': sum(c.isalpha() for c in url),
        'url': url,
        'domain': 'example.com'
    }

    input_df = pd.DataFrame([features])

    # Use PyCaret's predict_model
    predictions = predict_model(model, data=input_df)

    # Extract the numeric label and confidence score
    numeric_label = predictions['prediction_label'][0]
    confidence_score = predictions['prediction_score'][0]

    # Map numeric label to its category name
    predicted_label = label_mapping.get(numeric_label, "Unknown")

    return predicted_label, confidence_score
```

## Run a Test Prediction

In [34]:
```python
# Test the function
test_url = "http://www.facebook.com"
prediction = predict_website_legitimacy(test_url, loaded_model)
print(f"The predicted class for {test_url} is: {prediction}")
```

```
The predicted class for http://www.facebook.com is: ('benign', 0.6896)
```

In [38]:
```python
def predict_website_legitimacy_with_domain_check(url, model):
    cleaned_url = clean_url(url)
    prediction, confidence = predict_website_legitimacy(cleaned_url, model)
```

```
    # Override for well-known domains
    trusted_domains = ["facebook.com", "google.com", "apple.com"]
    domain = urlparse(cleaned_url).netloc.replace("www.", "")

    if domain in trusted_domains:
        return "benign", max(confidence, 0.9)  # Adjust confidence threshold for kn

    return prediction, confidence

# Test with the modified prediction function
test_url = "https://www.google.com"
prediction = predict_website_legitimacy_with_domain_check(test_url, loaded_model)
print(f"The predicted class for {test_url} is: {prediction}")

test_url = "http://www.google.com"
prediction = predict_website_legitimacy_with_domain_check(test_url, loaded_model)
print(f"The predicted class for {test_url} is: {prediction}")
```

```
The predicted class for https://www.google.com is: ('benign', 0.918)
The predicted class for http://www.google.com is: ('benign', 0.9)
```

In [ ]: