MALICIOUS URL DETECTION

The goal of this project is to develop a malicious URL detection system that identifies potentially harmful URLs to users.

```
!pip install pyspark
```

```
Collecting pyspark
  Downloading pyspark-3.5.1.tar.gz (317.0 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 317.0/317.0 MB 2.7 MB/s eta 0:00:00
  Preparing metadata (setup.py) ... done
Requirement already satisfied: py4j==0.10.9.7 in /usr/local/lib/python3.10/dist-packages (from pyspark) (0.10.9.7)
Building wheels for collected packages: pyspark
  Building wheel for pyspark (setup.py) ... done
  Created wheel for pyspark: filename=pyspark-3.5.1-py2.py3-none-any.whl size=317488491 sha256=2a68d0aa42bf056398434b8fd76621f2b6a5a3b75
  Stored in directory: /root/.cache/pip/wheels/80/1d/60/2c256ed38dddce2fdd93be545214a63e02fbd8d74fb0b7f3a6
Successfully built pyspark
Installing collected packages: pyspark
Successfully installed pyspark-3.5.1
```

```
pip install findspark
```

```
Collecting findspark
  Downloading findspark-2.0.1-py2.py3-none-any.whl (4.4 kB)
Installing collected packages: findspark
Successfully installed findspark-2.0.1
```

```
import findspark
findspark.init()
from pyspark.sql import SparkSession
spark = SparkSession.builder.master("local[*]").getOrCreate()
spark.conf.set("spark.sql.repl.eagerEval.enabled", True) # Property used to format output tables better
spark
```

SparkSession - in-memory

SparkContext

Spark UI

Version
v3.5.1
Master
local[*]
AppName
pyspark-shell

```
#importing datasets
md =spark.read.csv("/content/drive/MyDrive/malicious_url.csv",header=True,inferSchema=True)
```

```
md.show(10,False)
```

```
+--------------------------------+-----+----------------------+-------------+-------------+-------------+-------------+----------+--
|url                             |label|source                |url_has_login|url_has_client|url_has_server|url_has_admin|url_has_ip|ur
+--------------------------------+-----+----------------------+-------------+-------------+-------------+-------------+----------+--
|spa-security.de                 |0    |dmoz_harvard          |0            |0            |0            |0            |0         |0
|mallander.de                    |0    |majestic_million      |0            |0            |0            |0            |0         |0
|zzndb.com.cn                    |0    |alexatop1m            |0            |0            |0            |0            |0         |0
|enviroseal.com/                 |0    |data_clean_train_mendel|0            |0            |0            |0            |0         |0
|hunt1ngtonbank.3utilities.com   |1    |ALL-phishing-domains  |0            |0            |0            |0            |0         |0
|augenblickstudios.com           |0    |domcop                |0            |0            |0            |0            |0         |0
|semidiceviprima.ilcannocchiale.it|0    |dmoz_harvard          |0            |0            |0            |0            |0         |0
|nrgeology.blogspot.com.au       |0    |domcop                |0            |0            |0            |0            |0         |1
|eng.uwo.ca/research/fluidization/|0    |data_clean_train_mendel|0            |0            |0            |0            |0         |0
|geocities.com/crinoo/           |0    |data_clean_train_mendel|0            |0            |0            |0            |0         |0
+--------------------------------+-----+----------------------+-------------+-------------+-------------+-------------+----------+--
only showing top 10 rows
```

ACTIONS

```
#count the number of rows
md.count()
```

```
1682213
```

```
#take -retrieve the first n elements
md.take(3)
```

```
[Row(url='spa-security.de', label=0, source='dmoz_harvard', url_has_login=0, url_has_client=0, url_has_server=0, url_has_admin=0,
url_has_ip=0, url_isshorted=0, url_len=15, url_entropy=3.640223928941852, url_hamming_1=0.5083333333333333,
url_hamming_00=0.1416666666666666, url_hamming_10=0.2, url_hamming_01=0.2833333333333333, url_hamming_11=0.1583333333333333,
url_2bentropy=1.8003115692558134, url_3bentropy=2.3993425283550898, url_count_dot=1, url_count_https=0, url_count_http=0,
url_count_perc=0, url_count_hyphen=1, url_count_www=0, url_count_atrate=0, url_count_hash=0, url_count_semicolon=0,
url_count_underscore=0, url_count_ques=0, url_count_equal=0, url_count_amp=0, url_count_letter=13, url_count_digit=0,
url_count_sensitive_financial_words=0, url_count_sensitive_words=0, url_nunique_chars_ratio=0.8666666666666667, path_len=0,
path_count_no_of_dir=0, path_count_no_of_embed=0, path_count_zero=0, path_count_pertwent=0, path_has_any_sensitive_words=0,
path_count_lower=13, path_count_upper=0, path_count_nonascii=0, path_has_singlechardir=0, path_has_upperdir=0, query_len=0,
query_count_components=0, pdomain_len=0, pdomain_count_hyphen=0, pdomain_count_atrate=0, pdomain_count_non_alphanum=0,
pdomain_count_digit=0, tld_len=2, tld='de', tld_is_sus=0, pdomain_min_distance=7, subdomain_len=0, subdomain_count_dot=0),
 Row(url='mallander.de', label=0, source='majestic_million', url_has_login=0, url_has_client=0, url_has_server=0, url_has_admin=0,
url_has_ip=0, url_isshorted=0, url_len=12, url_entropy=2.9182958340544887, url_hamming_1=0.4791666666666667,
url_hamming_00=0.1458333333333333, url_hamming_10=0.2291666666666666, url_hamming_01=0.28125, url_hamming_11=0.1666666666666666,
url_2bentropy=1.837704275830528, url_3bentropy=2.3398850493643777, url_count_dot=1, url_count_https=0, url_count_http=0,
url_count_perc=0, url_count_hyphen=0, url_count_www=0, url_count_atrate=0, url_count_hash=0, url_count_semicolon=0,
url_count_underscore=0, url_count_ques=0, url_count_equal=0, url_count_amp=0, url_count_letter=11, url_count_digit=0,
url_count_sensitive_financial_words=0, url_count_sensitive_words=0, url_nunique_chars_ratio=0.6666666666666666, path_len=0,
path_count_no_of_dir=0, path_count_no_of_embed=0, path_count_zero=0, path_count_pertwent=0, path_has_any_sensitive_words=0,
path_count_lower=11, path_count_upper=0, path_count_nonascii=0, path_has_singlechardir=0, path_has_upperdir=0, query_len=0,
query_count_components=0, pdomain_len=0, pdomain_count_hyphen=0, pdomain_count_atrate=0, pdomain_count_non_alphanum=0,
pdomain_count_digit=0, tld_len=2, tld='de', tld_is_sus=0, pdomain_min_distance=3, subdomain_len=0, subdomain_count_dot=0),
 Row(url='zzndb.com.cn', label=0, source='alexatop1m', url_has_login=0, url_has_client=0, url_has_server=0, url_has_admin=0,
url_has_ip=0, url_isshorted=0, url_len=12, url_entropy=2.9182958340544887, url_hamming_1=0.5520833333333334,
url_hamming_00=0.0729166666666666, url_hamming_10=0.21875, url_hamming_01=0.2604166666666667, url_hamming_11=0.21875,
url_2bentropy=1.740229209352982, url_3bentropy=2.3451616264274184, url_count_dot=2, url_count_https=0, url_count_http=0,
url_count_perc=0, url_count_hyphen=0, url_count_www=0, url_count_atrate=0, url_count_hash=0, url_count_semicolon=0,
url_count_underscore=0, url_count_ques=0, url_count_equal=0, url_count_amp=0, url_count_letter=10, url_count_digit=0,
url_count_sensitive_financial_words=0, url_count_sensitive_words=0, url_nunique_chars_ratio=0.6666666666666666, path_len=0,
path_count_no_of_dir=0, path_count_no_of_embed=0, path_count_zero=0, path_count_pertwent=0, path_has_any_sensitive_words=0,
path_count_lower=10, path_count_upper=0, path_count_nonascii=0, path_has_singlechardir=0, path_has_upperdir=0, query_len=0,
query_count_components=0, pdomain_len=0, pdomain_count_hyphen=0, pdomain_count_atrate=0, pdomain_count_non_alphanum=0,
pdomain_count_digit=0, tld_len=6, tld='com.cn', tld_is_sus=0, pdomain_min_distance=3, subdomain_len=0, subdomain_count_dot=0)]
```

```
#first- retrieve the first row in the dataset
md.first()
```

```
Row(url='spa-security.de', label=0, source='dmoz_harvard', url_has_login=0, url_has_client=0, url_has_server=0, url_has_admin=0,
url_has_ip=0, url_isshorted=0, url_len=15, url_entropy=3.640223928941852, url_hamming_1=0.5083333333333333,
url_hamming_00=0.1416666666666666, url_hamming_10=0.2, url_hamming_01=0.2833333333333333, url_hamming_11=0.1583333333333333,
url_2bentropy=1.8003115692558134, url_3bentropy=2.3993425283550898, url_count_dot=1, url_count_https=0, url_count_http=0,
url_count_perc=0, url_count_hyphen=1, url_count_www=0, url_count_atrate=0, url_count_hash=0, url_count_semicolon=0,
url_count_underscore=0, url_count_ques=0, url_count_equal=0, url_count_amp=0, url_count_letter=13, url_count_digit=0,
url_count_sensitive_financial_words=0, url_count_sensitive_words=0, url_nunique_chars_ratio=0.8666666666666667, path_len=0,
path_count_no_of_dir=0, path_count_no_of_embed=0, path_count_zero=0, path_count_pertwent=0, path_has_any_sensitive_words=0,
path_count_lower=13, path_count_upper=0, path_count_nonascii=0, path_has_singlechardir=0, path_has_upperdir=0, query_len=0,
query_count_components=0, pdomain_len=0, pdomain_count_hyphen=0, pdomain_count_atrate=0, pdomain_count_non_alphanum=0,
pdomain_count_digit=0, tld_len=2, tld='de', tld_is_sus=0, pdomain_min_distance=7, subdomain_len=0, subdomain_count_dot=0)
```

```
#limit - limit the number of rows
md.limit(10)
```

url	label	source	url_has_login	url_has_client	url_has_server	url_has_admin	url_has_ip	url_isshorted	url_len	url_entropy
spa-security.de	0	dmoz_harvard	0	0	0	0	0	0	15	3.640223928941
mallander.de	0	majestic_million	0	0	0	0	0	0	12	2.918295834054
zzndb.com.cn	0	alexatop1m	0	0	0	0	0	0	12	2.918295834054
enviroseal.com/	0	data_clean_train_...	0	0	0	0	0	0	15	3.640223928941
hunt1ngtonbank.3u...	1	ALL-phishing-domains	0	0	0	0	0	0	29	3.935398667466
augenblickstudios...	0	domcop	0	0	0	0	0	0	21	3.916126946588
semidiceviprima.i...	0	dmoz_harvard	0	0	0	0	0	0	33	3.675871105457
nrgeology.blogspo...	0	domcop	0	0	0	0	0	1	25	3.719079570624
eng.uwo.ca/resear...	0	data_clean_train_...	0	0	0	0	0	0	33	4.104407755459
geocities.com/cri...	0	data_clean_train_...	0	0	0	0	0	0	21	3.368042422572

```
#retrieves the columns in the dataset
md.columns
```

```
['url',
 'label',
 'source',
 'url_has_login',
 'url_has_client',
 'url_has_server',
 'url_has_admin',
 'url_has_ip',
 'url_isshorted',
 'url_len',
 'url_entropy',
 'url_hamming_1',
 'url_hamming_00',
 'url_hamming_10',
 'url_hamming_01',
 'url_hamming_11',
 'url_2bentropy',
 'url_3bentropy',
 'url_count_dot',
 'url_count_https',
 'url_count_http',
 'url_count_perc',
 'url_count_hyphen',
 'url_count_www',
 'url_count_atrate',
 'url_count_hash',
 'url_count_semicolon',
 'url_count_underscore',
 'url_count_ques',
 'url_count_equal',
 'url_count_amp',
 'url_count_letter',
 'url_count_digit',
 'url_count_sensitive_financial_words',
 'url_count_sensitive_words',
 'url_nunique_chars_ratio',
 'path_len',
 'path_count_no_of_dir',
 'path_count_no_of_embed',
 'path_count_zero',
 'path_count_pertwent',
 'path_has_any_sensitive_words',
 'path_count_lower',
 'path_count_upper',
 'path_count_nonascii',
 'path_has_singlechardir',
 'path_has_upperdir',
 'query_len',
 'query_count_components',
 'pdomain_len',
 'pdomain_count_hyphen',
 'pdomain_count_atrate',
 'pdomain_count_non_alphanum',
 'pdomain_count_digit',
 'tld_len',
 'tld',
 'tld_is_sus',
 'pdomain_min_distance',
```

```
#returns the datatypes of every features in the dataset
md.dtypes
```

```
[('url', 'string'),
 ('label', 'int'),
 ('source', 'string'),
 ('url_has_login', 'int'),
 ('url_has_client', 'int'),
 ('url_has_server', 'int'),
 ('url_has_admin', 'int'),
 ('url_has_ip', 'int'),
 ('url_isshorted', 'int'),
 ('url_len', 'int'),
 ('url_entropy', 'double'),
 ('url_hamming_1', 'double'),
 ('url_hamming_00', 'double'),
 ('url_hamming_10', 'double'),
 ('url_hamming_01', 'double'),
 ('url_hamming_11', 'double'),
 ('url_2bentropy', 'double'),
 ('url_3bentropy', 'double'),
```

```
('url_count_dot', 'int'),
('url_count_https', 'int'),
('url_count_http', 'int'),
('url_count_perc', 'int'),
('url_count_hyphen', 'int'),
('url_count_www', 'int'),
('url_count_atrate', 'int'),
('url_count_hash', 'int'),
('url_count_semicolon', 'int'),
('url_count_underscore', 'int'),
('url_count_ques', 'int'),
('url_count_equal', 'int'),
('url_count_amp', 'int'),
('url_count_letter', 'int'),
('url_count_digit', 'int'),
('url_count_sensitive_financial_words', 'int'),
('url_count_sensitive_words', 'int'),
('url_nunique_chars_ratio', 'double'),
('path_len', 'int'),
('path_count_no_of_dir', 'int'),
('path_count_no_of_embed', 'int'),
('path_count_zero', 'int'),
('path_count_pertwent', 'int'),
('path_has_any_sensitive_words', 'int'),
('path_count_lower', 'int'),
('path_count_upper', 'int'),
('path_count_nonascii', 'int'),
('path_has_singlechardir', 'int'),
('path_has_upperdir', 'int'),
('query_len', 'int'),
('query_count_components', 'int'),
('pdomain_len', 'int'),
('pdomain_count_hyphen', 'int'),
('pdomain_count_atrate', 'int'),
('pdomain_count_non_alphanum', 'int'),
('pdomain_count_digit', 'int'),
('tld_len', 'int'),
('tld', 'string'),
('tld_is_sus', 'int'),
```

```
md.printSchema()
```

```
root
 |-- url: string (nullable = true)
 |-- label: integer (nullable = true)
 |-- source: string (nullable = true)
 |-- url_has_login: integer (nullable = true)
 |-- url_has_client: integer (nullable = true)
 |-- url_has_server: integer (nullable = true)
 |-- url_has_admin: integer (nullable = true)
 |-- url_has_ip: integer (nullable = true)
 |-- url_isshorted: integer (nullable = true)
 |-- url_len: integer (nullable = true)
 |-- url_entropy: double (nullable = true)
 |-- url_hamming_1: double (nullable = true)
 |-- url_hamming_00: double (nullable = true)
 |-- url_hamming_10: double (nullable = true)
 |-- url_hamming_01: double (nullable = true)
 |-- url_hamming_11: double (nullable = true)
 |-- url_2bentropy: double (nullable = true)
 |-- url_3bentropy: double (nullable = true)
 |-- url_count_dot: integer (nullable = true)
 |-- url_count_https: integer (nullable = true)
 |-- url_count_http: integer (nullable = true)
 |-- url_count_perc: integer (nullable = true)
 |-- url_count_hyphen: integer (nullable = true)
 |-- url_count_www: integer (nullable = true)
 |-- url_count_atrate: integer (nullable = true)
 |-- url_count_hash: integer (nullable = true)
 |-- url_count_semicolon: integer (nullable = true)
 |-- url_count_underscore: integer (nullable = true)
 |-- url_count_ques: integer (nullable = true)
 |-- url_count_equal: integer (nullable = true)
 |-- url_count_amp: integer (nullable = true)
 |-- url_count_letter: integer (nullable = true)
 |-- url_count_digit: integer (nullable = true)
 |-- url_count_sensitive_financial_words: integer (nullable = true)
 |-- url_count_sensitive_words: integer (nullable = true)
 |-- url_nunique_chars_ratio: double (nullable = true)
 |-- path_len: integer (nullable = true)
 |-- path_count_no_of_dir: integer (nullable = true)
 |-- path_count_no_of_embed: integer (nullable = true)
 |-- path_count_zero: integer (nullable = true)
```

```
 |-- path_count_pertwent: integer (nullable = true)
 |-- path_has_any_sensitive_words: integer (nullable = true)
 |-- path_count_lower: integer (nullable = true)
 |-- path_count_upper: integer (nullable = true)
 |-- path_count_nonascii: integer (nullable = true)
 |-- path_has_singlechardir: integer (nullable = true)
 |-- path_has_upperdir: integer (nullable = true)
 |-- query_len: integer (nullable = true)
 |-- query_count_components: integer (nullable = true)
 |-- pdomain_len: integer (nullable = true)
 |-- pdomain_count_hyphen: integer (nullable = true)
 |-- pdomain_count_atrate: integer (nullable = true)
 |-- pdomain_count_non_alphanum: integer (nullable = true)
 |-- pdomain_count_digit: integer (nullable = true)
 |-- tld_len: integer (nullable = true)
 |-- tld: string (nullable = true)
```

```
#calculate the number of unique values in the "source"
from pyspark.sql import functions as func
source_unique_count = md.select(func.size(func.collect_set("source"))).collect()[0][0]
print(f'Number of unique sources: {source_unique_count}')
```

```
Number of unique sources: 13
```

```
#corr- calculates the correlation coefficient between url_len and url_entropy
correlation =md.stat.corr("url_len", "url_entropy")
print(f'Correlation between url_len and url_entropy: {correlation}')
```

```
Correlation between url_len and url_entropy: 0.4880093630965763
```

```
#mean - calculates the mean value of the url-len
mean_url_len = md.select(func.mean("url_len")).collect()[0][0]
print(f'Mean url length: {mean_url_len}')
```

```
Mean url length: 23.78756614055414
```

```
#min - calculates the minimum value of the url_hamming
min_url_hamming_1 =md.select(func.min("url_hamming_1")).collect()[0][0]
print(f'Minimum url hamming_1: {min_url_hamming_1}')
```

```
Minimum url hamming_1: 0.3157894736842105
```

```
#max - calculates the maximum value of the url_entropy
max_url_entropy = md.select(func.max("url_entropy")).collect()[0][0]
print(f'Maximum url entropy: {max_url_entropy}')
```

```
Maximum url entropy: 5.354315429593166
```

```
#describe - returns the statistical summary of url_len and url_entropy
stats = md.describe(['url_len', 'url_entropy'])
stats
```

summary	url_len	url_entropy
count	1682213	1682213
mean	23.78756614055414	3.5270062993174243
stddev	34.23825991155175	0.46003807487031595
min	2	-0.0
max	8250	5.354315429593166

TRANSFORMATIONS

```
#calculates the average 'url_len' for each distinct value in the 'source'
grouped_data = md.groupBy('source').agg({'url_len': 'mean'})
grouped_data
```

source	avg(url_len)
data_clean_test_m...	25.793289799594685
openphish	58.6
dmoz_harvard	16.547755120749898
phishtank	26.46653075854653
manual	121.02671755725191
domcop	16.837430317894302
ALL-phishing-links	80.20846829047095
alexatop1m	14.320089947513923
aa419	18.71719641401793
majestic_million	15.149602155141654
data_clean_train_...	25.80477114414816
tranco_K2K4W	13.632867504450171
ALL-phishing-domains	25.722650704523783

```
#creates a new DataFrame 'md1' by adding a constant value of 10 to the values in the 'url_len' column of the original DataFrame 'md'.
from pyspark.sql.functions import col, lit
md1 = md.withColumn('new_column', col('url_len') + lit(10))
md1
```

url	label	source	url_has_login	url_has_client	url_has_server	url_has_admin	url_has_ip	url_isshorted	url_len	url_entropy
spa-security.de	0	dmoz_harvard	0	0	0	0	0	0	15	3.640223928941
mallander.de	0	majestic_million	0	0	0	0	0	0	12	2.918295834054
zzndb.com.cn	0	alexatop1m	0	0	0	0	0	0	12	2.918295834054
enviroseal.com/	0	data_clean_train_...	0	0	0	0	0	0	15	3.640223928941
hunt1ngtonbank.3u...	1	ALL-phishing-domains	0	0	0	0	0	0	29	3.935398667466
augenblickstudios...	0	domcop	0	0	0	0	0	0	21	3.916126946588
semidiceviprima.i...	0	dmoz_harvard	0	0	0	0	0	0	33	3.675871105457
nrgeology.blogspo...	0	domcop	0	0	0	0	0	1	25	3.719079570624
eng.uwo.ca/resear...	0	data_clean_train_...	0	0	0	0	0	0	33	4.104407755459
geocities.com/cri...	0	data_clean_train_...	0	0	0	0	0	0	21	3.368042422572
angelfire.com/rpg...	0	data_clean_train_...	0	0	0	0	0	0	35	3.668728288980
esspeedee.com	0	domcop	0	0	0	0	0	0	13	2.653544297030
aguasandinas.onli...	1	ALL-phishing-links	0	0	0	0	0	0	118	4.631289130086
tier-freizeitpark.de	0	dmoz_harvard	0	0	0	0	0	0	20	3.346439344671
hwa.net	0	dmoz_harvard	0	0	0	0	0	0	7	2.807354922057
selbstmordgedanke...	0	dmoz_harvard	0	0	0	0	0	0	23	3.849223912390
i.webring.com/hub...	0	data_clean_train_...	0	0	0	0	0	0	32	4.054229296672
opusdei.org	0	alexatop1m	0	0	0	0	0	0	11	3.277613436819
cjyqyxnssq.duckdn...	1	phishtank	0	0	0	0	0	0	23	3.795088586397
akinsalortaokul.m...	0	dmoz_harvard	0	0	0	0	0	0	26	3.844106544814

only showing top 20 rows

```
#Rename the source colum into datasource
renamed_data = md.withColumnRenamed('source', 'data_source')
renamed_data
```

url	label	data_source	url_has_login	url_has_client	url_has_server	url_has_
spa-security.de	0	dmoz_harvard	0	0	0	0
mallander.de	0	majestic_million	0	0	0	0
zzndb.com.cn	0	alexatop1m	0	0	0	0
enviroseal.com/	0	data_clean_train_...	0	0	0	0
hunt1ngtonbank.3u...	1	ALL-phishing-domains	0	0	0	0
augenblickstudios...	0	domcop	0	0	0	0
semidiceviprima.i...	0	dmoz_harvard	0	0	0	0
nrgeology.blogspo...	0	domcop	0	0	0	0
eng.uwo.ca/resear...	0	data_clean_train_...	0	0	0	0
geocities.com/cri...	0	data_clean_train_...	0	0	0	0
angelfire.com/rpg...	0	data_clean_train_...	0	0	0	0
esspeedee.com	0	domcop	0	0	0	0
aguasandinas.onli...	1	ALL-phishing-links	0	0	0	0
tier-freizeitpark.de	0	dmoz_harvard	0	0	0	0
hwa.net	0	dmoz_harvard	0	0	0	0
selbstmordgedanke...	0	dmoz_harvard	0	0	0	0
i.webring.com/hub...	0	data_clean_train_...	0	0	0	0
opusdei.org	0	alexatop1m	0	0	0	0
cjyqyxnssq.duckdn...	1	phishtank	0	0	0	0
akinsalortaokul.m...	0	dmoz_harvard	0	0	0	0

only showing top 20 rows

```
#sort the url_len value in ascending order
sorted_data = md.orderBy('url_len', ascending=False)
sorted_data
```

url	label	source	url_has_login	url_has_client	url_has_server	url_has_admin
hpnepgnmwrgdrrsds...	1	ALL-phishing-links	0	0	0	0
hpnepgnmwrgdrrsds...	1	ALL-phishing-links	0	0	0	0
sg-stamps.com/__m...	1	ALL-phishing-links	0	0	0	0
sg-stamps.com/__m...	1	ALL-phishing-links	0	0	0	0
sg-stamps.com/__m...	1	ALL-phishing-links	0	0	0	0
sg-stamps.com/__m...	1	ALL-phishing-links	0	0	0	0
sg-stamps.com/__m...	1	ALL-phishing-links	0	0	0	0
sg-stamps.com/__m...	1	ALL-phishing-links	0	0	0	0
hpnepgnmwrgdrrsds...	1	ALL-phishing-links	0	0	0	0
sg-stamps.com/__m...	1	ALL-phishing-links	0	0	0	0
sg-stamps.com/__m...	1	ALL-phishing-links	0	0	0	0
sg-stamps.com/__m...	1	ALL-phishing-links	0	0	0	0
sg-stamps.com/__m...	1	ALL-phishing-links	0	0	0	0
sg-stamps.com/__m...	1	ALL-phishing-links	0	0	0	0
sg-stamps.com/__m...	1	ALL-phishing-links	0	0	0	0
sg-stamps.com/__m...	1	ALL-phishing-links	0	0	0	0
		ALL-				

```
#creates a df 'filtered' by selecting rows from the original DataFrame 'md' where the URL length is greater than 50 characters and the count

filterd=md.filter((col("url_len") > 50) & (col("url_count_dot") >= 3))
filterd
```

url	label	source	url_has_login	url_has_client	url_has_server	url_has
162.241.69.15/f87...	1	ALL-phishing-links	0	0	0	0
rabo-aanmaning.in...	1	ALL-phishing-links	0	0	0	0
benthon-resonator...	1	ALL-phishing-links	1	0	0	0
itmanagedservices...	1	ALL-phishing-links	0	0	0	0
nocontent-mrmkyku...	1	ALL-phishing-links	0	0	0	0
httppostsfb-7oxem...	1	ALL-phishing-links	0	0	0	0
198.55.96.123/ind...	1	ALL-phishing-links	1	0	0	0
oliverconstructio...	1	ALL-phishing-links	0	0	0	0
fittbikese.hu/t/o...	1	ALL-phishing-links	0	0	0	0
kongfoojew.com/on...	1	ALL-phishing-links	1	0	0	0
gplcrown.com/http...	1	ALL-phishing-links	0	0	0	0
bafutcouncil.net/...	1	ALL-phishing-links	1	0	0	0
pochtarefund-xeq1...	1	ALL-phishing-links	0	0	0	0
refund-hmrc.uk-co...	1	ALL-phishing-links	1	0	0	0
yale.edu/ynhti/cu...	0	data_clean_test_m...	0	0	0	0
qloqqqzasmhcjidnt...	1	ALL-phishing-domains	0	0	0	0
quod.lib.umich.ed...	0	data_clean_train_...	0	0	0	0
info.clienti.sanp...	1	ALL-phishing-links	1	1	0	0
02fecb9e9ee6ebb5c...	1	ALL-phishing-domains	0	0	0	0
m.fb.com-xqdyvjgg...	1	ALL-phishing-links	0	0	0	0

only showing top 20 rows

```
#extracts key-value pairs for all columns except 'source' and 'url', and returns these pairs as a list of tuples
def extract_values(row):
    values = []
    for key, value in row.asDict().items():
        if key not in ['source', 'url']:
            values.append((f"{key}_value", value))
    return values

# Apply flatMap transformation
flat_mapped_rdd = md.rdd.flatMap(extract_values)

# Convert the RDD back to DataFrame for better visualization
flat_mapped_df = flat_mapped_rdd.toDF(['feature', 'value'])
flat_mapped_df.show()
```

```
+--------------------+-----+
|             feature|value|
+--------------------+-----+
|         label_value|    0|
| url_has_login_value|    0|
|url_has_client_value|    0|
|url_has_server_value|    0|
| url_has_admin_value|    0|
|    url_has_ip_value|    0|
| url_isshorted_value|    0|
|       url_len_value|   15|
|   url_entropy_value| NULL|
| url_hamming_1_value| NULL|
|url_hamming_00_value| NULL|
|url_hamming_10_value| NULL|
|url_hamming_01_value| NULL|
|url_hamming_11_value| NULL|
| url_2bentropy_value| NULL|
| url_3bentropy_value| NULL|
| url_count_dot_value|    1|
|url_count_https_v...|    0|
|url_count_http_value|    0|
|url_count_perc_value|    0|
+--------------------+-----+
only showing top 20 rows
```

```
#isNull(),filter(): to check null values in all the columns
from pyspark.sql.functions import col
for col_name in md.columns:
    null_count = md.filter(col(col_name).isNull()).count()
    print(f"Column '{col_name}' has {null_count} null values.")
```

```
Column 'url' has 0 null values.
Column 'label' has 0 null values.
Column 'source' has 0 null values.
Column 'url_has_login' has 0 null values.
Column 'url_has_client' has 0 null values.
Column 'url_has_server' has 0 null values.
Column 'url_has_admin' has 0 null values.
Column 'url_has_ip' has 0 null values.
Column 'url_isshorted' has 0 null values.
Column 'url_len' has 0 null values.
Column 'url_entropy' has 0 null values.
Column 'url_hamming_1' has 0 null values.
Column 'url_hamming_00' has 0 null values.
Column 'url_hamming_10' has 0 null values.
Column 'url_hamming_01' has 0 null values.
Column 'url_hamming_11' has 0 null values.
Column 'url_2bentropy' has 0 null values.
Column 'url_3bentropy' has 0 null values.
Column 'url_count_dot' has 0 null values.
Column 'url_count_https' has 0 null values.
Column 'url_count_http' has 0 null values.
Column 'url_count_perc' has 0 null values.
Column 'url_count_hyphen' has 0 null values.
Column 'url_count_www' has 0 null values.
Column 'url_count_atrate' has 0 null values.
Column 'url_count_hash' has 0 null values.
Column 'url_count_semicolon' has 0 null values.
Column 'url_count_underscore' has 0 null values.
Column 'url_count_ques' has 0 null values.
Column 'url_count_equal' has 0 null values.
Column 'url_count_amp' has 0 null values.
Column 'url_count_letter' has 0 null values.
Column 'url_count_digit' has 0 null values.
Column 'url_count_sensitive_financial_words' has 0 null values.
Column 'url_count_sensitive_words' has 0 null values.
Column 'url_nunique_chars_ratio' has 0 null values.
Column 'path_len' has 0 null values.
Column 'path_count_no_of_dir' has 0 null values.
Column 'path_count_no_of_embed' has 0 null values.
Column 'path_count_zero' has 0 null values.
Column 'path_count_pertwent' has 0 null values.
Column 'path_has_any_sensitive_words' has 0 null values.
Column 'path_count_lower' has 0 null values.
Column 'path_count_upper' has 0 null values.
Column 'path_count_nonascii' has 0 null values.
Column 'path_has_singlechardir' has 0 null values.
Column 'path_has_upperdir' has 0 null values.
Column 'query_len' has 0 null values.
Column 'query_count_components' has 0 null values.
Column 'pdomain_len' has 0 null values.
Column 'pdomain_count_hyphen' has 0 null values.
Column 'pdomain_count_atrate' has 0 null values.
Column 'pdomain_count_non_alphanum' has 0 null values.
Column 'pdomain_count_digit' has 0 null values.
Column 'tld_len' has 0 null values.
Column 'tld' has 0 null values.
Column 'tld_is_sus' has 0 null values.
Column 'pdomain_min_distance' has 0 null values.
```

SPARK SQL

```
#drops the duplicates in the dataset
dropdup = md.dropDuplicates()
dropdup.show(10)
```

```
+--------------------+-----+--------------------+-------------+--------------+--------------+-------------+----------+-------------+----
|                 url|label|              source|url_has_login|url_has_client|url_has_server|url_has_admin|url_has_ip|url_isshorted|url_
+--------------------+-----+--------------------+-------------+--------------+--------------+-------------+----------+-------------+----
|       .shamber.info|    0|        dmoz_harvard|            0|             0|             0|            0|         0|            0|
|     pedroamador.com|    0|          alexatop1m|            0|             0|             0|            0|         0|            0|
|pge152932id072447...|    1|  ALL-phishing-links|            0|             0|             0|            0|         0|            0|
|      sxlaoqiang.com|    0|          alexatop1m|            0|             0|             0|            0|         0|            0|
|    central-bank.net|    0|        dmoz_harvard|            0|             0|             0|            0|         0|            0|
|       mccamylaw.com|    0|        dmoz_harvard|            0|             0|             0|            0|         0|            0|
|   thisiswccc.co.uk/|    0|data_clean_train_...|            0|             0|             0|            0|         0|            0|
|       reisefotos.li|    0|        dmoz_harvard|            0|             0|             0|            0|         0|            0|
```

```
|       ereleases.com|    0|          alexatop1m|           0|           0|           0|           0|        0|           0|
|vedicpredictiveas...|    0|data_clean_test_m...|           0|           0|           0|           0|        0|           0|
+--------------------+-----+--------------------+------------+------------+------------+------------+---------+------------+----
only showing top 10 rows
```

```
#Show all entries in source column
select_md=md.select("source").show(10)
```

```
+--------------------+
|              source|
+--------------------+
|        dmoz_harvard|
|    majestic_million|
|          alexatop1m|
|data_clean_train_...|
|ALL-phishing-domains|
|              domcop|
|        dmoz_harvard|
|              domcop|
|data_clean_train_...|
|data_clean_train_...|
+--------------------+
only showing top 10 rows
```

```
#creates a new df 'when_sql' by adding a column that assigns a value of 1 when the 'source' column matches 'ALL-phishing-domains' and 0 othe
from pyspark.sql.functions import *
from pyspark.sql.types import *
when_sql=md.select("source", when(md.source == 'ALL-phishing-domains', 1).otherwise(0)).show()
```

```
+--------------------+-------------------------------------------------------------+
|              source|CASE WHEN (source = ALL-phishing-domains) THEN 1 ELSE 0 END|
+--------------------+-------------------------------------------------------------+
|        dmoz_harvard|                                                            0|
|    majestic_million|                                                            0|
|          alexatop1m|                                                            0|
|data_clean_train_...|                                                            0|
|ALL-phishing-domains|                                                            1|
|              domcop|                                                            0|
|        dmoz_harvard|                                                            0|
|              domcop|                                                            0|
|data_clean_train_...|                                                            0|
|data_clean_train_...|                                                            0|
|data_clean_train_...|                                                            0|
|              domcop|                                                            0|
|  ALL-phishing-links|                                                            0|
|        dmoz_harvard|                                                            0|
|        dmoz_harvard|                                                            0|
|        dmoz_harvard|                                                            0|
|data_clean_train_...|                                                            0|
|          alexatop1m|                                                            0|
|           phishtank|                                                            0|
|        dmoz_harvard|                                                            0|
+--------------------+-------------------------------------------------------------+
only showing top 20 rows
```

```
#isin - filters the dataset to include rows where the 'source' column values are either "dmoz_harvard" or "Edomcop"
isin_sql=md [md.source.isin("dmoz_harvard", "Edomcop")].show()
```

```
+--------------------+-----+------------+-------------+-------------+-------------+-------------+---------+-------------+-------+----
|                 url|label|      source|url_has_login|url_has_client|url_has_server|url_has_admin|url_has_ip|url_isshorted|url_len|
+--------------------+-----+------------+-------------+-------------+-------------+-------------+---------+-------------+-------+----
|     spa-security.de|    0|dmoz_harvard|            0|            0|            0|            0|        0|            0|     15| 3.6
|semidiceviprima.i...|    0|dmoz_harvard|            0|            0|            0|            0|        0|            0|     33|  3.
|tier-freizeitpark.de|    0|dmoz_harvard|            0|            0|            0|            0|        0|            0|     20|3.34
|             hwa.net|    0|dmoz_harvard|            0|            0|            0|            0|        0|            0|      7|2.80
|selbstmordgedanke...|    0|dmoz_harvard|            0|            0|            0|            0|        0|            0|     23|3.84
|akinsalortaokul.m...|    0|dmoz_harvard|            0|            0|            0|            0|        0|            0|     26|3.84
|foerderschwerpunk...|    0|dmoz_harvard|            0|            0|            0|            0|        0|            0|     21|3.68
|    bernsteinhexe.de|    0|dmoz_harvard|            0|            0|            0|            0|        0|            0|     16|3.14
|       .shamber.info|    0|dmoz_harvard|            0|            0|            0|            0|        0|            0|     13| 3.5
|  tb-reintegratie.nl|    0|dmoz_harvard|            0|            0|            0|            0|        0|            0|     18|3.30
|         crostel.com|    0|dmoz_harvard|            0|            0|            0|            0|        0|            0|     11|3.09
|ci.riverdale-park...|    0|dmoz_harvard|            0|            0|            0|            0|        0|            0|     23|3.76
|        .bory-var.hu|    0|dmoz_harvard|            0|            0|            0|            0|        0|            0|     12|3.25
```

```
| westlocktractor.com|    0|dmoz_harvard|           0|           0|           0|           0|        0|           0|     19| 3.3
|      lisademetz.com|    0|dmoz_harvard|           0|           0|           0|           0|        0|           0|     14| 3.5
|         oldrauma.fi|    0|dmoz_harvard|           0|           0|           0|           0|        0|           0|     11|3.27
|      bigarrowup.com|    0|dmoz_harvard|           0|           0|           0|           0|        0|           0|     14| 3.5
| viglianoedilizia.it|    0|dmoz_harvard|           0|           0|           0|           0|        0|           0|     19|3.22
|   fmercury.narod.ru|    0|dmoz_harvard|           0|           0|           0|           0|        0|           0|     17| 3.3
|     bouncyfun.co.uk|    0|dmoz_harvard|           0|           0|           0|           0|        0|           0|     15|3.05
+--------------------+-----+------------+------------+--------------+--------------+------------+---------+------------+-------+----
only showing top 20 rows
```

```
#like - checks if the 'source' column contains the substring 'd'
like_sql=md.select("url_len", "source", md.source.like("% d %")).show()
```

```
+-------+--------------------+-----------------+
|url_len|              source|source LIKE % d %|
+-------+--------------------+-----------------+
|     15|        dmoz_harvard|            false|
|     12|    majestic_million|            false|
|     12|          alexatop1m|            false|
|     15|data_clean_train_...|            false|
|     29|ALL-phishing-domains|            false|
|     21|              domcop|            false|
|     33|        dmoz_harvard|            false|
|     25|              domcop|            false|
|     33|data_clean_train_...|            false|
|     21|data_clean_train_...|            false|
|     35|data_clean_train_...|            false|
|     13|              domcop|            false|
|    118|  ALL-phishing-links|            false|
|     20|        dmoz_harvard|            false|
|      7|        dmoz_harvard|            false|
|     23|        dmoz_harvard|            false|
|     32|data_clean_train_...|            false|
|     11|          alexatop1m|            false|
|     23|           phishtank|            false|
|     26|        dmoz_harvard|            false|
+-------+--------------------+-----------------+
only showing top 20 rows
```

```
#startswith - check if the 'source' column starts with the string "ALL".
start=md.select("url_len", "source", md.source.startswith("ALL")).show(5)
```

```
+-------+--------------------+-----------------------+
|url_len|              source|startswith(source, ALL)|
+-------+--------------------+-----------------------+
|     15|        dmoz_harvard|                  false|
|     12|    majestic_million|                  false|
|     12|          alexatop1m|                  false|
|     15|data_clean_train_...|                  false|
|     29|ALL-phishing-domains|                   true|
+-------+--------------------+-----------------------+
only showing top 5 rows
```

```
#endswith- check if the 'source' column ends with the string "rd".
ends=md.select("url_len", "source", md.source.endswith("rd")).show()
```

```
+-------+--------------------+--------------------+
|url_len|              source|endswith(source, rd)|
+-------+--------------------+--------------------+
|     15|        dmoz_harvard|                true|
|     12|    majestic_million|               false|
|     12|          alexatop1m|               false|
|     15|data_clean_train_...|               false|
|     29|ALL-phishing-domains|               false|
|     21|              domcop|               false|
|     33|        dmoz_harvard|                true|
|     25|              domcop|               false|
|     33|data_clean_train_...|               false|
|     21|data_clean_train_...|               false|
|     35|data_clean_train_...|               false|
|     13|              domcop|               false|
|    118|  ALL-phishing-links|               false|
|     20|        dmoz_harvard|                true|
|      7|        dmoz_harvard|                true|
|     23|        dmoz_harvard|                true|
```

```
|     32|data_clean_train_...|              false|
|     11|          alexatop1m|              false|
|     23|           phishtank|              false|
|     26|        dmoz_harvard|               true|
+-------+--------------------+-------------------+
only showing top 20 rows
```

```
#substr- Extracts a substring starting from the first character and taking the next 10 characters
substr=md.select(md.url.substr(1, 10).alias("url")).show()
```

```
+----------+
|       url|
+----------+
|spa-securi|
|mallander.|
|zzndb.com.|
|enviroseal|
|hunt1ngton|
|augenblick|
|semidicevi|
|nrgeology.|
|eng.uwo.ca|
|geocities.|
|angelfire.|
|esspeedee.|
|aguasandin|
|tier-freiz|
|   hwa.net|
|selbstmord|
|i.webring.|
|opusdei.or|
|cjyqyxnssq|
|akinsalort|
+----------+
only showing top 20 rows
```

```
#calculates the count of each distinct value in the 'source' column
group=md.groupBy("source").count().show()
```

```
+--------------------+------+
|              source| count|
+--------------------+------+
|data_clean_test_m...| 71056|
|           openphish|   175|
|        dmoz_harvard|479959|
|           phishtank|133914|
|              manual|  1048|
|              domcop|159833|
|  ALL-phishing-links|123496|
|          alexatop1m|199672|
|               aa419|  2454|
|    majestic_million|112104|
+--------------------+------+
only showing top 10 rows
```

```
#show only rows where the 'source' column is equal to 'domcop'
filter_sql=md.filter(md["source"] == 'domcop').show(5)
```

```
+--------------------+-----+------+-------------+--------------+--------------+-------------+----------+-------------+-------+----------
|                 url|label|source|url_has_login|url_has_client|url_has_server|url_has_admin|url_has_ip|url_isshorted|url_len|       url
+--------------------+-----+------+-------------+--------------+--------------+-------------+----------+-------------+-------+----------
|augenblickstudios...|    0|domcop|            0|             0|             0|            0|         0|            0|     21| 3.9161269
|nrgeology.blogspo...|    0|domcop|            0|             0|             0|            0|         0|            1|     25|3.71907957
|       esspeedee.com|    0|domcop|            0|             0|             0|            0|         0|            0|     13|2.65354429
|  abookishescape.com|    0|domcop|            0|             0|             0|            0|         0|            0|     18|3.46132014
| theartofliving.info|    0|domcop|            0|             0|             0|            0|         0|            0|     19| 3.5766176
+--------------------+-----+------+-------------+--------------+--------------+-------------+----------+-------------+-------+----------
only showing top 5 rows
```

```
# Write & Save File in .json format
md.select("source", "url","url_len","url_has_login","url_has_server") \
        .write \
        .save("Url.json",format="json")
```

```
md_output = spark.read.format("json").load("Url.json")

# Show the output DataFrame
md_output.show()
```

```
+--------------------+--------------------+------------+--------------+-------+
|              source|                 url|url_has_login|url_has_server|url_len|
+--------------------+--------------------+------------+--------------+-------+
|          alexatop1m|         idegene.com|           0|             0|     11|
|        dmoz_harvard|irvinequeersatuci...|           0|             0|     30|
|              domcop|   resolution.org.uk|           0|             0|     17|
|data_clean_train_...|beartoothpc.50meg...|           0|             0|     23|
|  ALL-phishing-links|proserviceunit.co...|           0|             0|     40|
|           phishtank|aesosn.aoeesccone...|           0|             0|     29|
|          alexatop1m|   baixarcdstops.com|           0|             0|     17|
|        dmoz_harvard| iowaacupuncture.com|           0|             0|     19|
|  ALL-phishing-links|thebeauty-spa.com...|           0|             0|    189|
|        dmoz_harvard|skate-libchavy.wz.cz|           0|             0|     20|
|          alexatop1m|   novinhasporno.org|           0|             0|     17|
|              domcop|    the-french.co.uk|           0|             0|     16|
|          alexatop1m|theconversionpros...|           0|             0|     21|
|          alexatop1m|  thibaudpoirier.com|           0|             0|     18|
|        dmoz_harvard|         kneissl.com|           0|             0|     11|
|           phishtank|   fesxcn.clfmicl.cn|           0|             0|     17|
|    majestic_million|   madidashoes.co.kr|           0|             0|     17|
|        tranco_K2K4W|            co-co.nl|           0|             0|      8|
|        dmoz_harvard|alaskansfordonyou...|           0|             0|     23|
|data_clean_test_m...|epsrc.ac.uk/websi...|           0|             0|     69|
+--------------------+--------------------+------------+--------------+-------+
only showing top 20 rows
```

Importing libraries

```
from pyspark.ml.feature import VectorAssembler,StringIndexer
from pyspark.ml.classification import LogisticRegression,DecisionTreeClassifier


#Rearrange the dataset
data = md.select('url','source', 'url_has_login', 'url_has_client', 'url_has_server', 'url_has_admin', 'url_has_ip', 'url_isshorted', 'url_l


data.dtypes
```

```
[('url', 'string'),
 ('source', 'string'),
 ('url_has_login', 'int'),
 ('url_has_client', 'int'),
 ('url_has_server', 'int'),
 ('url_has_admin', 'int'),
 ('url_has_ip', 'int'),
 ('url_isshorted', 'int'),
 ('url_len', 'int'),
 ('url_entropy', 'double'),
 ('url_hamming_1', 'double'),
 ('url_hamming_00', 'double'),
 ('url_hamming_10', 'double'),
 ('url_hamming_01', 'double'),
 ('url_hamming_11', 'double'),
 ('url_2bentropy', 'double'),
 ('url_3bentropy', 'double'),
 ('url_count_dot', 'int'),
 ('url_count_https', 'int'),
 ('url_count_http', 'int'),
 ('url_count_perc', 'int'),
 ('url_count_hyphen', 'int'),
 ('url_count_www', 'int'),
 ('url_count_atrate', 'int'),
 ('url_count_hash', 'int'),
 ('url_count_semicolon', 'int'),
 ('url_count_underscore', 'int'),
 ('url_count_ques', 'int'),
 ('url_count_equal', 'int'),
 ('url_count_amp', 'int'),
 ('url_count_letter', 'int'),
 ('url_count_digit', 'int'),
 ('url_count_sensitive_financial_words', 'int'),
 ('url_count_sensitive_words', 'int'),
 ('url_nunique_chars_ratio', 'double'),
 ('path_len', 'int'),
 ('path_count_no_of_dir', 'int'),
 ('path_count_no_of_embed', 'int'),
```

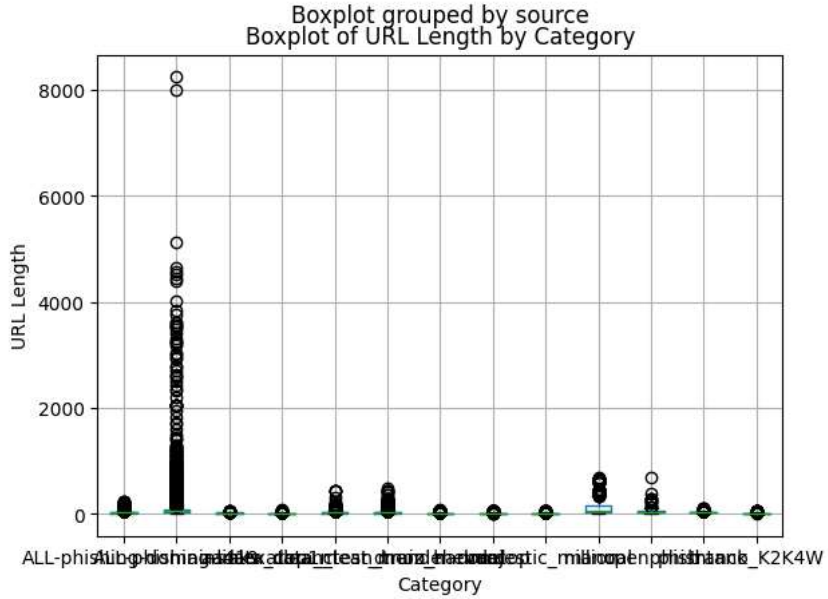
```
        ('path_count_zero', 'int'),
        ('path_count_pertwent', 'int'),
        ('path_has_any_sensitive_words', 'int'),
        ('path_count_lower', 'int'),
        ('path_count_upper', 'int'),
        ('path_count_nonascii', 'int'),
        ('path_has_singlechardir', 'int'),
        ('path_has_upperdir', 'int'),
        ('query_len', 'int'),
        ('query_count_components', 'int'),
        ('pdomain_len', 'int'),
        ('pdomain_count_hyphen', 'int'),
        ('pdomain_count_atrate', 'int'),
        ('pdomain_count_non_alphanum', 'int'),
        ('pdomain_count_digit', 'int'),
        ('tld_len', 'int'),
        ('tld', 'string'),
        ('tld_is_sus', 'int'),
        ('pdomain_min_distance', 'int'),
        ('subdomain_len', 'int'),
```

```
data.groupBy('label').count().show()
```

```
+-----+-------+
|label|  count|
+-----+-------+
|    1| 361419|
|    0|1320794|
+-----+-------+
```

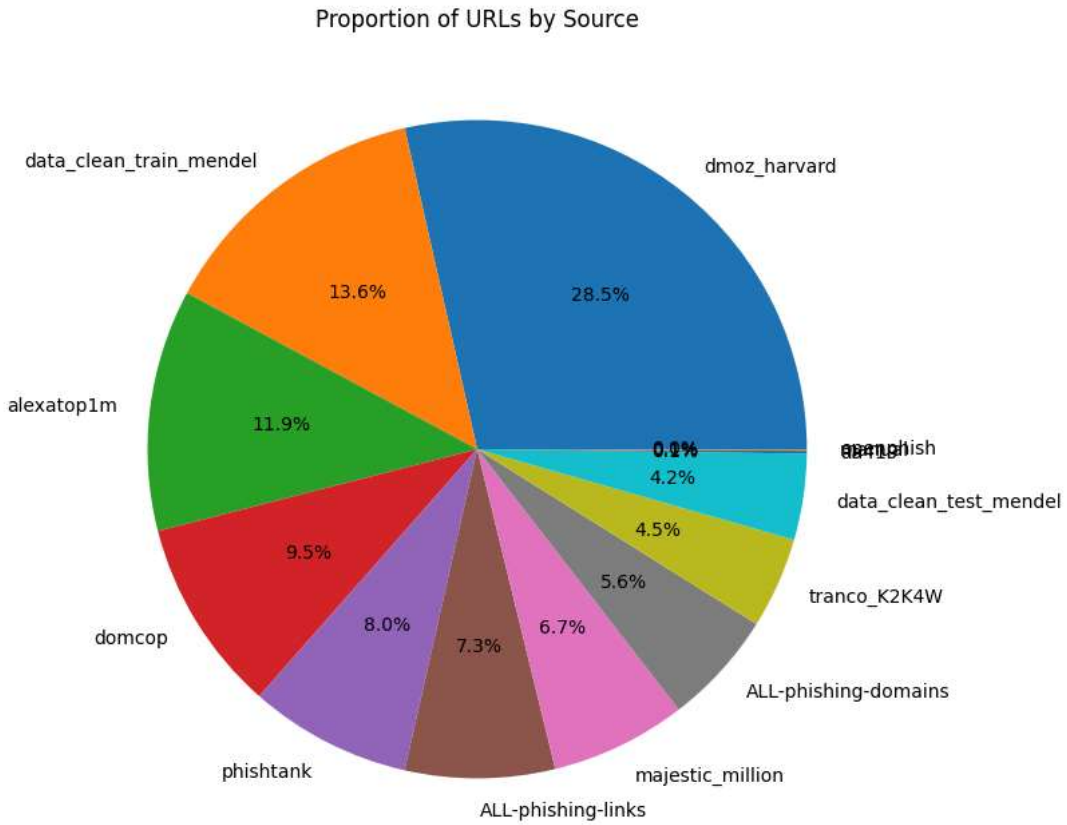
```
import matplotlib.pyplot as plt
#boxplot to compare the distribution of URL lengths between different categories (e.g., malicious vs. non-malicious).
plt.figure(figsize=(10, 6))
md.select('source', 'url_len').toPandas().boxplot(column='url_len', by='source')
plt.xlabel('Category')
plt.ylabel('URL Length')
plt.title('Boxplot of URL Length by Category')
plt.show()
```

```
<Figure size 1000x600 with 0 Axes>
```




```
#pie chart to visualize the proportion of URLs from different sources in the dataset.
sources_count = md.groupBy('source').count().orderBy('count', ascending=False).toPandas()

plt.figure(figsize=(8, 8))
plt.pie(sources_count['count'], labels=sources_count['source'], autopct='%1.1f%%')
plt.title('Proportion of URLs by Source')
plt.show()
```




```
#unique values for source
data.select("source").distinct().show()
```

```
+--------------------+
|              source|
+--------------------+
|data_clean_test_m...|
|           openphish|
|        dmoz_harvard|
|           phishtank|
|              manual|
|              domcop|
|  ALL-phishing-links|
|          alexatop1m|
|               aa419|
|    majestic_million|
|data_clean_train_...|
|        tranco_K2K4W|
|ALL-phishing-domains|
+--------------------+
```

```
#Converting the string into numerical
sourceEncode = StringIndexer(inputCol = "source",outputCol ="source_index").fit(data)
data = sourceEncode.transform(data)
```

```
#after encoding
data.select("source_index").distinct().show()
```

```
+------------+
|source_index|
+------------+
|         8.0|
|         0.0|
|         7.0|
|         1.0|
|         4.0|
|        11.0|
|         3.0|
|         2.0|
|        10.0|
|         6.0|
```

```
|         5.0|
|         9.0|
|        12.0|
+------------+
```

FEATURE SELECTION

```
df = data.select('source','source_index','url_has_login','url_has_client','url_has_server','url_len','label')

df.show()
```

```
+--------------------+------------+-------------+--------------+--------------+-------+-----+
|              source|source_index|url_has_login|url_has_client|url_has_server|url_len|label|
+--------------------+------------+-------------+--------------+--------------+-------+-----+
|        dmoz_harvard|         0.0|            0|             0|             0|     15|    0|
|    majestic_million|         6.0|            0|             0|             0|     12|    0|
|          alexatop1m|         2.0|            0|             0|             0|     12|    0|
|data_clean_train_...|         1.0|            0|             0|             0|     15|    0|
|ALL-phishing-domains|         7.0|            0|             0|             0|     29|    1|
|              domcop|         3.0|            0|             0|             0|     21|    0|
|        dmoz_harvard|         0.0|            0|             0|             0|     33|    0|
|              domcop|         3.0|            0|             0|             0|     25|    0|
|data_clean_train_...|         1.0|            0|             0|             0|     33|    0|
|data_clean_train_...|         1.0|            0|             0|             0|     21|    0|
|data_clean_train_...|         1.0|            0|             0|             0|     35|    0|
|              domcop|         3.0|            0|             0|             0|     13|    0|
|  ALL-phishing-links|         5.0|            0|             0|             0|    118|    1|
|        dmoz_harvard|         0.0|            0|             0|             0|     20|    0|
|        dmoz_harvard|         0.0|            0|             0|             0|      7|    0|
|        dmoz_harvard|         0.0|            0|             0|             0|     23|    0|
|data_clean_train_...|         1.0|            0|             0|             0|     32|    0|
|          alexatop1m|         2.0|            0|             0|             0|     11|    0|
|           phishtank|         4.0|            0|             0|             0|     23|    1|
|        dmoz_harvard|         0.0|            0|             0|             0|     26|    0|
+--------------------+------------+-------------+--------------+--------------+-------+-----+
only showing top 20 rows
```

```
#VectorAssembler
req_features =  ['source_index','url_has_login','url_has_client','url_has_server','url_len']

vec_assemb = VectorAssembler(inputCols=req_features,outputCol ="features")


vec_df = vec_assemb.transform(df)
vec_df.show()
```

```
+--------------------+------------+-------------+--------------+--------------+-------+-----+--------------------+
|              source|source_index|url_has_login|url_has_client|url_has_server|url_len|label|            features|
+--------------------+------------+-------------+--------------+--------------+-------+-----+--------------------+
|        dmoz_harvard|         0.0|            0|             0|             0|     15|    0|      (5,[4],[15.0])|
|    majestic_million|         6.0|            0|             0|             0|     12|    0|(5,[0,4],[6.0,12.0])|
|          alexatop1m|         2.0|            0|             0|             0|     12|    0|(5,[0,4],[2.0,12.0])|
|data_clean_train_...|         1.0|            0|             0|             0|     15|    0|(5,[0,4],[1.0,15.0])|
|ALL-phishing-domains|         7.0|            0|             0|             0|     29|    1|(5,[0,4],[7.0,29.0])|
|              domcop|         3.0|            0|             0|             0|     21|    0|(5,[0,4],[3.0,21.0])|
|        dmoz_harvard|         0.0|            0|             0|             0|     33|    0|      (5,[4],[33.0])|
|              domcop|         3.0|            0|             0|             0|     25|    0|(5,[0,4],[3.0,25.0])|
|data_clean_train_...|         1.0|            0|             0|             0|     33|    0|(5,[0,4],[1.0,33.0])|
|data_clean_train_...|         1.0|            0|             0|             0|     21|    0|(5,[0,4],[1.0,21.0])|
|data_clean_train_...|         1.0|            0|             0|             0|     35|    0|(5,[0,4],[1.0,35.0])|
|              domcop|         3.0|            0|             0|             0|     13|    0|(5,[0,4],[3.0,13.0])|
|  ALL-phishing-links|         5.0|            0|             0|             0|    118|    1|(5,[0,4],[5.0,118...|
|        dmoz_harvard|         0.0|            0|             0|             0|     20|    0|      (5,[4],[20.0])|
|        dmoz_harvard|         0.0|            0|             0|             0|      7|    0|       (5,[4],[7.0])|
|        dmoz_harvard|         0.0|            0|             0|             0|     23|    0|      (5,[4],[23.0])|
|data_clean_train_...|         1.0|            0|             0|             0|     32|    0|(5,[0,4],[1.0,32.0])|
|          alexatop1m|         2.0|            0|             0|             0|     11|    0|(5,[0,4],[2.0,11.0])|
|           phishtank|         4.0|            0|             0|             0|     23|    1|(5,[0,4],[4.0,23.0])|
|        dmoz_harvard|         0.0|            0|             0|             0|     26|    0|      (5,[4],[26.0])|
+--------------------+------------+-------------+--------------+--------------+-------+-----+--------------------+
only showing top 20 rows
```

MODEL BUILDING

```
#Train test split
train_df,test_df = vec_df.randomSplit([0.7,0.3])
```

```
train_df.count()
```

```
1177011
```

```
test_df.count()
```

```
505202
```

```
train_df.describe()
```

summary	source	source_index	url_has_login	url_has_client	url_has_server	url_len	label
count	1177011	1177011	1177011	1177011	1177011	1177011	1177011
mean	NULL	2.9006517356252406	0.01566255540517463	0.001932862139776094	0.002285450178460524	23.78241749652297	0.21507020750018
stddev	NULL	2.803622824233719	0.12416623076345035	0.04392183765202071	0.04775174167762791	34.441351140745354	0.41087121677359
min	ALL-phishing-domains	0.0	0	0	0	2	0

PCA

```
from pyspark.ml.feature import PCA
```

```
numericColsAll  = ['url_has_login','url_has_client','url_has_server','url_len']
label = 'label'
assembler = VectorAssembler(inputCols=numericColsAll , outputCol="Numfeatures")
df = assembler.transform(md)
```

```
    # Apply PCA
pca = PCA(k=2, inputCol="Numfeatures", outputCol="pca_features")
model = pca.fit(df)
result = model.transform(df)
result.select("Numfeatures", "pca_features").show(truncate=False)
```

```
+---------------+-----------------------------------------+
|Numfeatures    |pca_features                             |
+---------------+-----------------------------------------+
|(4,[3],[15.0]) |[-14.999992762772711,0.014593764420373079]|
|(4,[3],[12.0]) |[-11.99999421021817,0.011675011536298463] |
|(4,[3],[12.0]) |[-11.99999421021817,0.011675011536298463] |
|(4,[3],[15.0]) |[-14.999992762772711,0.014593764420373079]|
|(4,[3],[29.0]) |[-28.999986008027243,0.02821461121272128]|
|(4,[3],[21.0]) |[-20.999989867881794,0.02043127018852231] |
|(4,[3],[33.0]) |[-32.99998407809996,0.03210628172482077]  |
|(4,[3],[25.0]) |[-24.99998793795452,0.024322940700621798] |
|(4,[3],[33.0]) |[-32.99998407809996,0.03210628172482077]  |
|(4,[3],[21.0]) |[-20.999989867881794,0.02043127018852231] |
|(4,[3],[35.0]) |[-34.999983113136324,0.03405211698087052] |
|(4,[3],[13.0]) |[-12.99999372773635,0.012647929164323335] |
|(4,[3],[118.0])|[-117.99994306714532,0.11480428010693489] |
|(4,[3],[20.0]) |[-19.999990350363614,0.01945835256049744] |
|(4,[3],[7.0])  |[-6.999996622627265,0.006810423396174103] |
|(4,[3],[23.0]) |[-22.999988902918158,0.022377105444572054]|
|(4,[3],[32.0]) |[-31.999984560581783,0.0311333640967959]  |
|(4,[3],[11.0]) |[-10.999994692699987,0.010702093908273591]|
|(4,[3],[23.0]) |[-22.999988902918158,0.022377105444572054]|
|(4,[3],[26.0]) |[-25.9999874554727,0.02529585832864667]   |
+---------------+-----------------------------------------+
only showing top 20 rows
```

LOGISTIC REGRESSION

```
lr = LogisticRegression(featuresCol ="features",labelCol="label")

lr_model = lr.fit(train_df)

y_pred1 = lr_model.transform(test_df)

y_pred1.show()
```

```
+--------------------+------------+-------------+--------------+--------------+-------+-----+--------------------+--------------------+--
|              source|source_index|url_has_login|url_has_client|url_has_server|url_len|label|            features|       rawPrediction|
+--------------------+------------+-------------+--------------+--------------+-------+-----+--------------------+--------------------+--
|ALL-phishing-domains|         7.0|            0|             0|             0|      5|    1|(5,[0,4],[7.0,5.0])|[1.72823993077039...|[0
|ALL-phishing-domains|         7.0|            0|             0|             0|      5|    1|(5,[0,4],[7.0,5.0])|[1.72823993077039...|[0
|ALL-phishing-domains|         7.0|            0|             0|             0|      6|    1|(5,[0,4],[7.0,6.0])|[1.64632299637371...|[0
|ALL-phishing-domains|         7.0|            0|             0|             0|      6|    1|(5,[0,4],[7.0,6.0])|[1.64632299637371...|[0
|ALL-phishing-domains|         7.0|            0|             0|             0|      7|    1|(5,[0,4],[7.0,7.0])|[1.56440606197703...|[0
|ALL-phishing-domains|         7.0|            0|             0|             0|      7|    1|(5,[0,4],[7.0,7.0])|[1.56440606197703...|[0
|ALL-phishing-domains|         7.0|            0|             0|             0|      7|    1|(5,[0,4],[7.0,7.0])|[1.56440606197703...|[0
|ALL-phishing-domains|         7.0|            0|             0|             0|      7|    1|(5,[0,4],[7.0,7.0])|[1.56440606197703...|[0
|ALL-phishing-domains|         7.0|            0|             0|             0|      7|    1|(5,[0,4],[7.0,7.0])|[1.56440606197703...|[0
|ALL-phishing-domains|         7.0|            0|             0|             0|      7|    1|(5,[0,4],[7.0,7.0])|[1.56440606197703...|[0
|ALL-phishing-domains|         7.0|            0|             0|             0|      7|    1|(5,[0,4],[7.0,7.0])|[1.56440606197703...|[0
|ALL-phishing-domains|         7.0|            0|             0|             0|      7|    1|(5,[0,4],[7.0,7.0])|[1.56440606197703...|[0
|ALL-phishing-domains|         7.0|            0|             0|             0|      7|    1|(5,[0,4],[7.0,7.0])|[1.56440606197703...|[0
|ALL-phishing-domains|         7.0|            0|             0|             0|      7|    1|(5,[0,4],[7.0,7.0])|[1.56440606197703...|[0
|ALL-phishing-domains|         7.0|            0|             0|             0|      7|    1|(5,[0,4],[7.0,7.0])|[1.56440606197703...|[0
|ALL-phishing-domains|         7.0|            0|             0|             0|      7|    1|(5,[0,4],[7.0,7.0])|[1.56440606197703...|[0
|ALL-phishing-domains|         7.0|            0|             0|             0|      7|    1|(5,[0,4],[7.0,7.0])|[1.56440606197703...|[0
|ALL-phishing-domains|         7.0|            0|             0|             0|      7|    1|(5,[0,4],[7.0,7.0])|[1.56440606197703...|[0
|ALL-phishing-domains|         7.0|            0|             0|             0|      7|    1|(5,[0,4],[7.0,7.0])|[1.56440606197703...|[0
|ALL-phishing-domains|         7.0|            0|             0|             0|      7|    1|(5,[0,4],[7.0,7.0])|[1.56440606197703...|[0
+--------------------+------------+-------------+--------------+--------------+-------+-----+--------------------+--------------------+--
only showing top 20 rows
```

```
y_pred1.select('label','rawPrediction', 'probability', 'prediction').show()
```

```
+-----+--------------------+--------------------+----------+
|label|       rawPrediction|         probability|prediction|
+-----+--------------------+--------------------+----------+
|    1|[1.72823993077045...|[0.84918714962729...|       0.0|
|    1|[1.72823993077045...|[0.84918714962729...|       0.0|
|    1|[1.64632299637376...|[0.83839347341011...|       0.0|
|    1|[1.64632299637376...|[0.83839347341011...|       0.0|
|    1|[1.56440606197708...|[0.82698468535175...|       0.0|
|    1|[1.56440606197708...|[0.82698468535175...|       0.0|
|    1|[1.56440606197708...|[0.82698468535175...|       0.0|
|    1|[1.56440606197708...|[0.82698468535175...|       0.0|
|    1|[1.56440606197708...|[0.82698468535175...|       0.0|
|    1|[1.56440606197708...|[0.82698468535175...|       0.0|
|    1|[1.56440606197708...|[0.82698468535175...|       0.0|
|    1|[1.56440606197708...|[0.82698468535175...|       0.0|
|    1|[1.56440606197708...|[0.82698468535175...|       0.0|
|    1|[1.56440606197708...|[0.82698468535175...|       0.0|
|    1|[1.56440606197708...|[0.82698468535175...|       0.0|
|    1|[1.56440606197708...|[0.82698468535175...|       0.0|
|    1|[1.56440606197708...|[0.82698468535175...|       0.0|
|    1|[1.56440606197708...|[0.82698468535175...|       0.0|
|    1|[1.56440606197708...|[0.82698468535175...|       0.0|
|    1|[1.56440606197708...|[0.82698468535175...|       0.0|
+-----+--------------------+--------------------+----------+
only showing top 20 rows
```

```
from pyspark.ml.evaluation import MulticlassClassificationEvaluator
mult_eval =MulticlassClassificationEvaluator(labelCol="label",metricName="accuracy")


accuracy = mult_eval.evaluate(y_pred1)
print("Accuracy : " ,accuracy)
```

```
Accuracy :  0.8366811691165118
```

DECISIONTREE CLASSIFIER

```
#Decision tree
dc = DecisionTreeClassifier(featuresCol ="features",labelCol="label")
dc_model = dc.fit(train_df)
y_pred = dc_model.transform(test_df)

y_pred.show()
```

```
+--------------------+------------+-------------+--------------+--------------+-------+-----+--------------------+--------------+--------
|              source|source_index|url_has_login|url_has_client|url_has_server|url_len|label|            features| rawPrediction|probabil
+--------------------+------------+-------------+--------------+--------------+-------+-----+--------------------+--------------+--------
|ALL-phishing-domains|         7.0|            0|             0|             0|      5|    1|(5,[0,4],[7.0,5.0])|[0.0,246401.0]|  [0.0,1
|ALL-phishing-domains|         7.0|            0|             0|             0|      5|    1|(5,[0,4],[7.0,5.0])|[0.0,246401.0]|  [0.0,1
|ALL-phishing-domains|         7.0|            0|             0|             0|      6|    1|(5,[0,4],[7.0,6.0])|[0.0,246401.0]|  [0.0,1
|ALL-phishing-domains|         7.0|            0|             0|             0|      6|    1|(5,[0,4],[7.0,6.0])|[0.0,246401.0]|  [0.0,1
|ALL-phishing-domains|         7.0|            0|             0|             0|      7|    1|(5,[0,4],[7.0,7.0])|[0.0,246401.0]|  [0.0,1
|ALL-phishing-domains|         7.0|            0|             0|             0|      7|    1|(5,[0,4],[7.0,7.0])|[0.0,246401.0]|  [0.0,1
|ALL-phishing-domains|         7.0|            0|             0|             0|      7|    1|(5,[0,4],[7.0,7.0])|[0.0,246401.0]|  [0.0,1
|ALL-phishing-domains|         7.0|            0|             0|             0|      7|    1|(5,[0,4],[7.0,7.0])|[0.0,246401.0]|  [0.0,1
|ALL-phishing-domains|         7.0|            0|             0|             0|      7|    1|(5,[0,4],[7.0,7.0])|[0.0,246401.0]|  [0.0,1
|ALL-phishing-domains|         7.0|            0|             0|             0|      7|    1|(5,[0,4],[7.0,7.0])|[0.0,246401.0]|  [0.0,1
```