CYDEV INTELLIGENCE

Prototype 2.0

CONTENTS

PISHING URL DETECTION

Typical Pishing Example

IMPORTANT YOUR PASSWORD WILL EXPIRE WITHIN 24 HOURS!!

[redacted]

11:08 PM (1 minute ago)

to [redacted]

Dear Students;

This email is meant to inform you that our University network password will expire within 24 hours.
Log in with your old password and follow next steps to update your Email and Microsoft team's password.

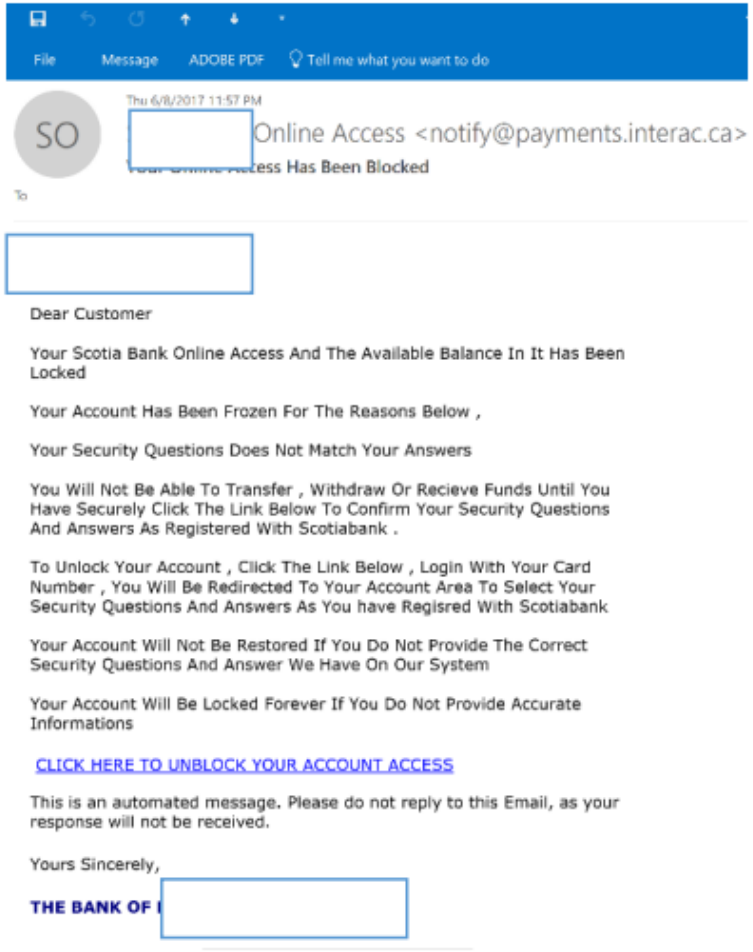
Please follow the link below.
https://www.[redacted]/admin/reset-password/

[redacted]

ज्ञानम् आत्म प्रदीपाय

Thank You;
System Administrator

Disclaimer:
For Educational Purposes Only



File Message ADOBE PDF Tell me what you want to do

SO

Thu 6/8/2017 11:57 PM

Online Access <notify@payments.interac.ca>

Has Been Blocked

To

Dear Customer

Your Scotia Bank Online Access And The Available Balance In It Has Been Locked

Your Account Has Been Frozen For The Reasons Below ,

Your Security Questions Does Not Match Your Answers

You Will Not Be Able To Transfer , Withdraw Or Recieve Funds Until You Have Securely Click The Link Below To Confirm Your Security Questions And Answers As Registered With Scotiabank .

To Unlock Your Account , Click The Link Below , Login With Your Card Number , You Will Be Redirected To Your Account Area To Select Your Security Questions And Answers As You have Regisred With Scotiabank

Your Account Will Not Be Restored If You Do Not Provide The Correct Security Questions And Answer We Have On Our System

Your Account Will Be Locked Forever If You Do Not Provide Accurate Informations

CLICK HERE TO UNBLOCK YOUR ACCOUNT ACCESS

This is an automated message. Please do not reply to this Email, as your response will not be received.

Yours Sincerely,

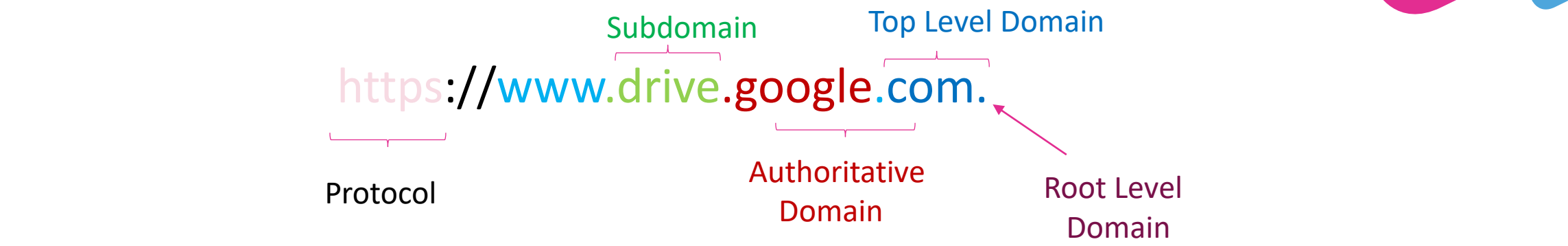
THE BANK OF

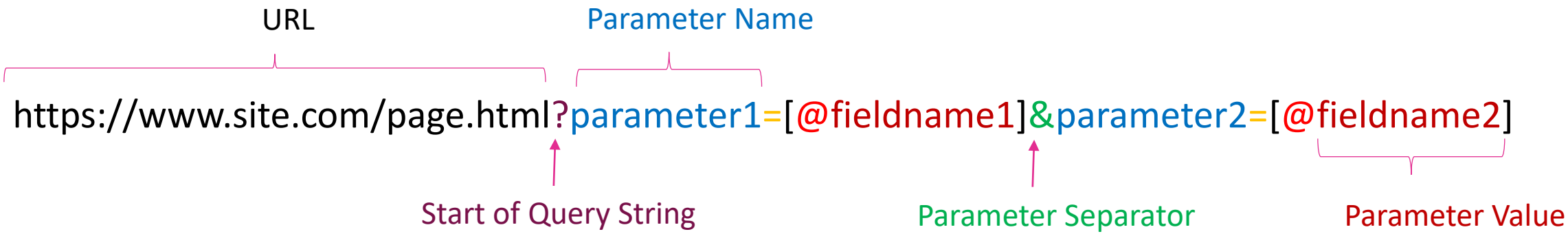
Your Account Will Be Locked Forever If You Do Not Provide Accurate Informations

CLICK HERE TO UNBLOCK YOUR ACCOUNT ACCESS

This is an automated message. Please do not reply to this Email, as your response will not be received.

Understanding URL structure





Visualize internal links of a website

In [70]:
```
list_urls = ['https://www.amazon.com/'] #here i take good site
links_with_text = []
```

In [71]:
```
for url in list_urls:
    drvr.get(url)
    soup = BeautifulSoup(drvr.page_source,"html.parser")
    for line in soup.find_all('a'):
        href = line.get('href')
        links_with_text.append([url, href])
```

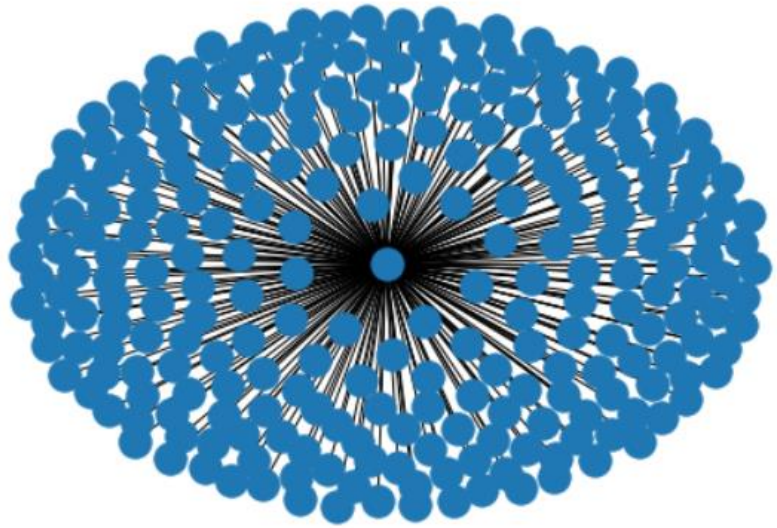
In [72]:
```
df = pd.DataFrame(links_with_text, columns=["from", "to"])
```

In [90]:
```
df.tail()
```

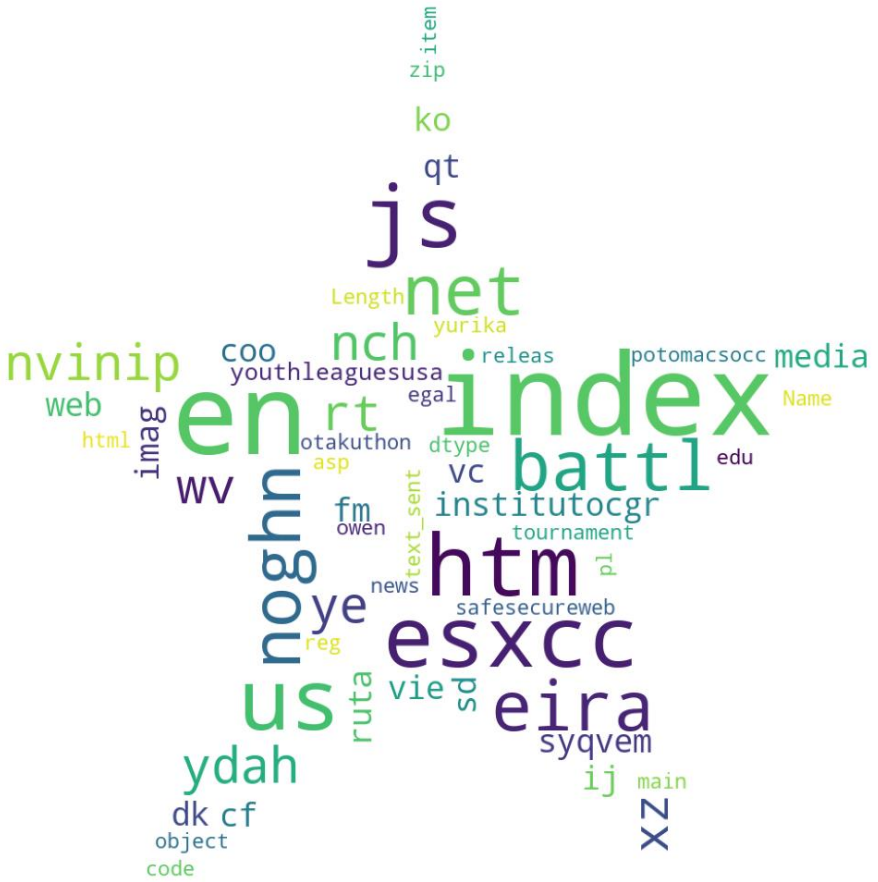
Out[90]:

	from	to
281	https://www.amazon.com/	/gp/browse.html?node=14498690011&ref_=amzn_nav...
282	https://www.amazon.com/	https://www.pillpack.com
283	https://www.amazon.com/	/gp/help/customer/display.html?nodeId=508088&r...
284	https://www.amazon.com/	/gp/help/customer/display.html?nodeId=468496&r...
285	https://www.amazon.com/	/interestbasedads/ref=footer_iba

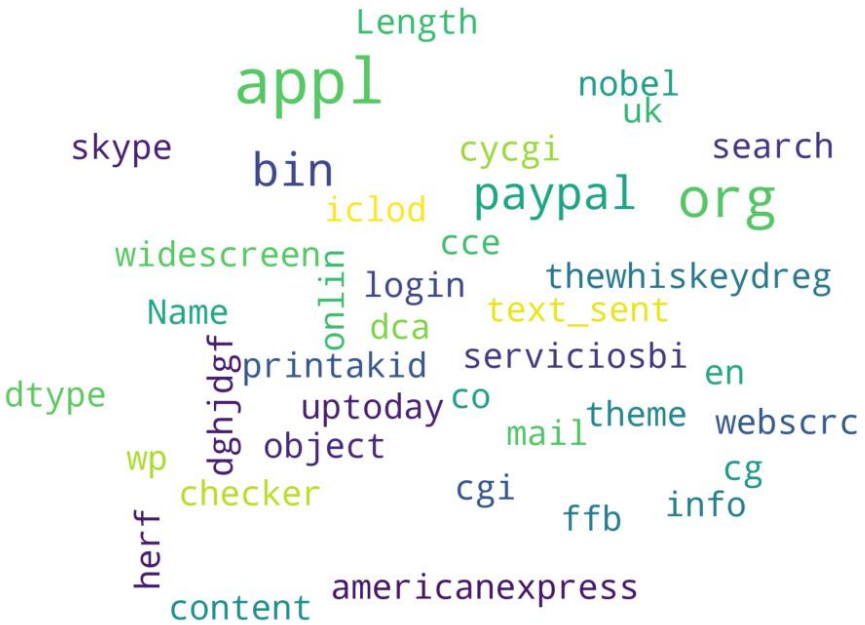
In [74]:
```
GA = networkx.from_pandas_edgelist(df, source="from", target="to")
networkx.draw(GA, with_labels=False)
```



Commonly used word in good URL



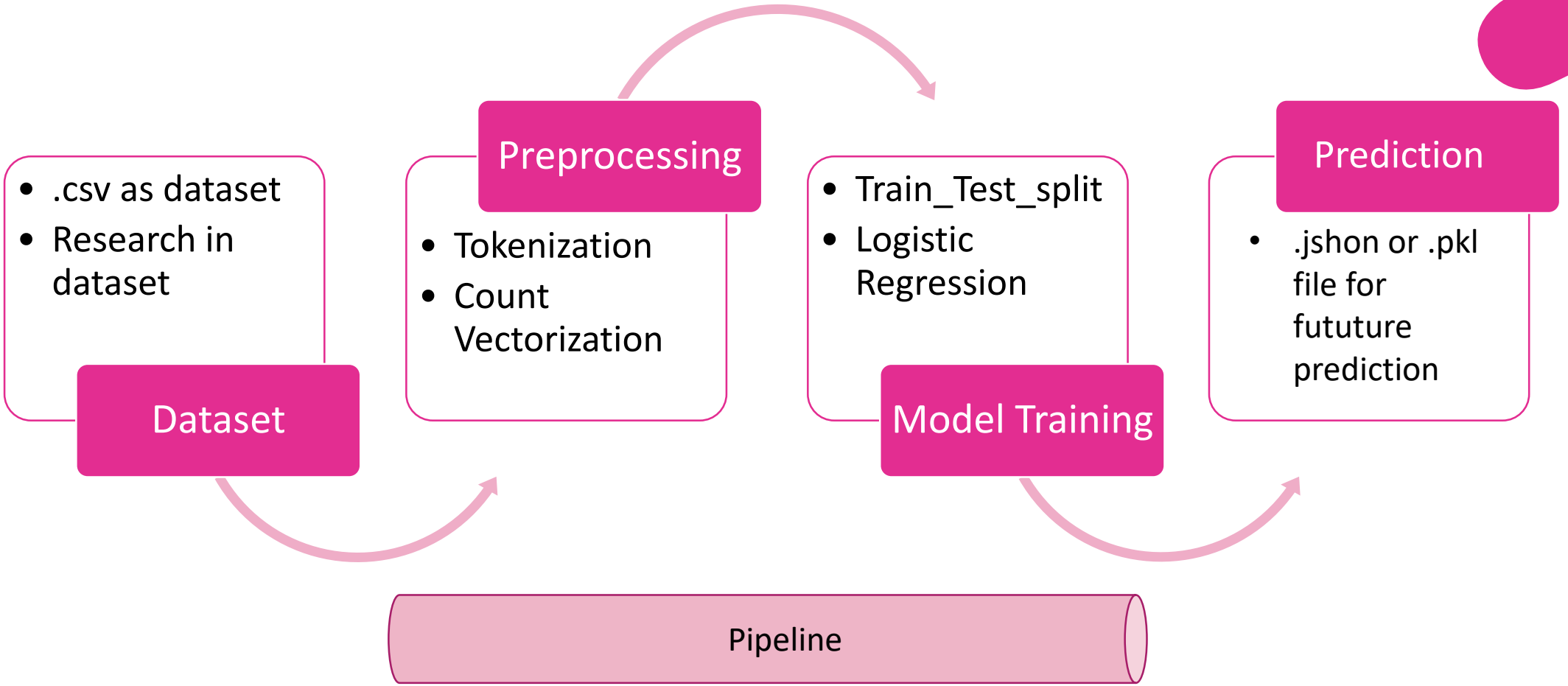
Commonly used words in Bad URL



Understanding the machine learning workflow

- Gathering data
- Data pre-processing
- Researching the model that will be best for the type of data
- Training and testing the model
- Evaluation

Dataflow Diagram:



Phising_site_urls.csv

- Data is containing 5,49,346 unique entries.
- There are two columns.
- Label column is prediction col which has 2 categories A. Good - which means the urls is not containing malicious stuff and **this site is not a Phishing Site.** B. Bad - which means the urls contains malicious stuffs and **this site are Phishing Sites.**

More about data set

In [2]:
```
phish_data = pd.read_csv('phishing_site_urls.csv')
phish_data.head()
```

Out[2]:

	URL	Label
0	nobell.it/70ffb52d079109dca5664cce6f317373782/...	bad
1	www.dghjdgf.com/paypal.co.uk/cycgi-bin/webscrc...	bad
2	serviciosbys.com/paypal.cgi.bin.get-into.herf....	bad
3	mail.printakid.com/www.online.americanexpress....	bad
4	thewhiskeydregs.com/wp-content/themes/widescre...	bad

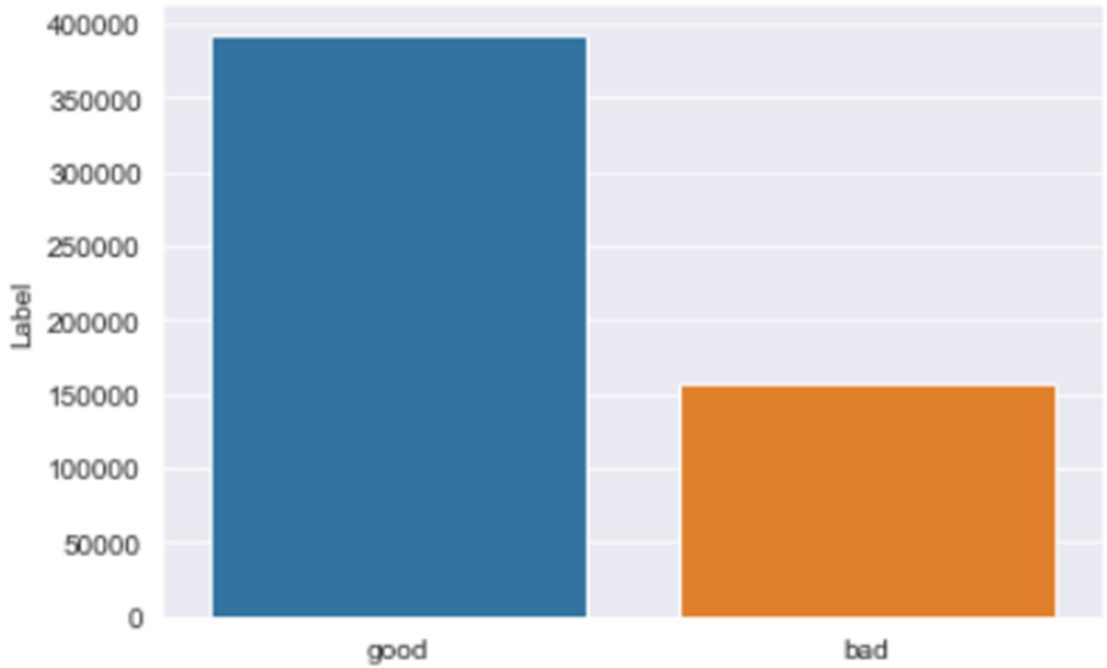
In [4]:
```
phish_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 549346 entries, 0 to 549345
Data columns (total 2 columns):
 #   Column  Non-Null Count   Dtype
---  ------  --------------   -----
 0   URL     549346 non-null  object
 1   Label   549346 non-null  object
dtypes: object(2)
memory usage: 8.4+ MB
```

Visualize Good and Bad URLs

```
In [7]: #visualizing target_coloumn
        sns.set_style('darkgrid')
        sns.barplot(label_counts.index,label_counts.Label)
```

```
Out[7]: <AxesSubplot:ylabel='Label'>
```



Preprocessing

- We have the data, we have to vectorize our URLs.
- I use RegexpTokenizer to tokenized words.
- And use CountVectorizer to transform a corpora of text to a vector of term / token counts.

Tokenization

- In order to use textual data for predictive modeling, the text must be parsed to remove certain words – this process is called **tokenization**.
- We can easily search for and customize text patterns that we wish to tokenize using RegexpTokenizer.
- We use RexepTokenizer from NLTK which is used for NLP
- RegexpTokenizer(r'[A-Za-z]+')
- Regular expressions (regex) are extremely useful in extracting characters from text by searching matches of a specific search pattern.

RexepTokenizer

```
: tokenizer = RegexpTokenizer(r'[A-Za-z]+').tokenize
```

```
: phish_data.URL[12]
```

```
: 'www.coincoele.com.br/Scripts/smiles/?pt-br/Paginas/default.aspx'
```

```
: # this will be pull letter which matches to expression
  tokenizer.tokenize(phish_data.URL[12])
```

```
: ['www',
   'coincoele',
   'com',
   'br',
   'Scripts',
   'smiles',
   'pt',
   'br',
   'Paginas',
   'default',
   'aspx']
```

CountVectorizer

Tokenized words need to be encoded as integers, or floating-point values, for use as inputs in machine learning algorithms. This process is called **vectorization**.

Data = ['The', 'quick', 'brown', 'fox', 'jumps', 'over', ' the', 'lazy', 'dog']

	The	quick	brown	fox	jumps	over	lazy	dog
Data	2	1	1	1	1	1	1	1

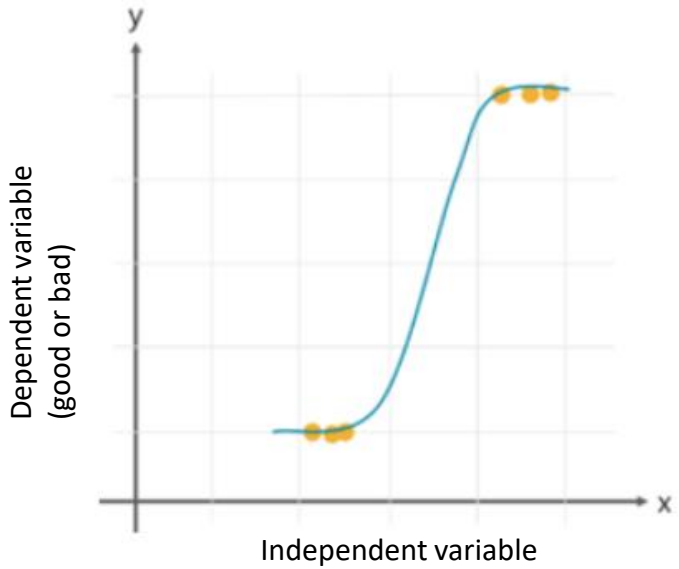
Model Training

- Before train our model we need to split our dataset into two subset Train and Test.

- Training dataset: it is used to train the algorithm and fit the machine learning model.
- Test dataset: Using the input element from the training data, the algorithms make predictions.

Logistic Regression

- Logistic Regression is a Machine Learning classification algorithm that is used to predict the probability of a categorical dependent variable. In logistic regression, the dependent variable is a binary variable that contains data coded as 1 (yes, success, etc.) or 0 (no, failure, etc.). In other words, the logistic regression model predicts P(Y=1) as a function of X.



Training And Testing Accuracy

```
# create lr object
lr = LogisticRegression()
```

```
##implementing Logistic regression for train the data
lr.fit(trainX,trainY)
```

```
LogisticRegression()
```

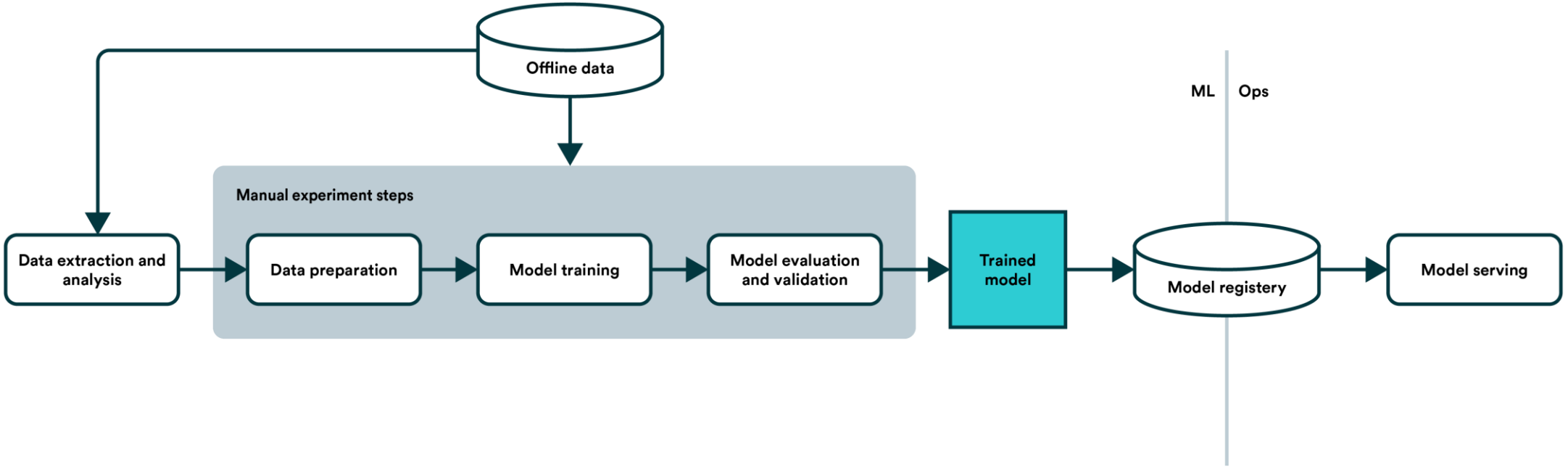
```
#calculting test score
lr.score(testX,testY)
```

```
0.9660324602983901
```

```
#calculating train score
lr.score(trainX,trainY)
```

```
0.9809106111759694
```

Pipeline


Offline data
ML
Ops
Manual experiment steps
Data extraction and analysis
Data preparation
Model training
Model evaluation and validation
Trained model
Model registery
Model serving

Classification Report And Confusion Matrix

```
Training Accuracy : 0.9809300282275387
Testing Accuracy : 0.9655227651688911

CLASSIFICATION REPORT

              precision    recall  f1-score   support

         Bad       0.91      0.97      0.94     36699
        Good       0.99      0.97      0.98    100638

    accuracy                           0.97    137337
   macro avg       0.95      0.97      0.96    137337
weighted avg       0.97      0.97      0.97    137337


CONFUSION MATRIX

<AxesSubplot:>
```

Prediction

Pickle lets the user to store data in byte format.
We store the trained data here and use it for predict urls are legitimate or phishing.

```
3]: pickle.dump(pipeline_ls,open('phishing.pkl','wb'))
```

```
4]: loaded_model = pickle.load(open('phishing.pkl', 'rb'))
result = loaded_model.score(testX,testY)
print(result)
```

```
0.966636813094796
```

```
5]: URLs_input1 = ['www.google.com','https://f461-106-196-1-48.in.ngrok.io']
URLs_input2 = ['www.controlledprojects.info/security/','https://www.facebook.com']
loaded_model = pickle.load(open('phishing.pkl', 'rb'))
result1 = loaded_model.predict(URLs_input1)
result2 = loaded_model.predict(URLs_input2)
print(result1)
print("*"*20)
print(result2)
```

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

References:

- BLACKHAT ASIA- https://youtu.be/s95CQNntunQ
- BLACKHAT EUROPE- https://youtu.be/0JhmR1Dpclo/
- MACHINE LEARNING TECHNIQUES- https://edureka.com/

FUTURE UPDATES

- Malicious Url Detection using Deep learning and neural network
- Extensions for Browser

THANK YOU