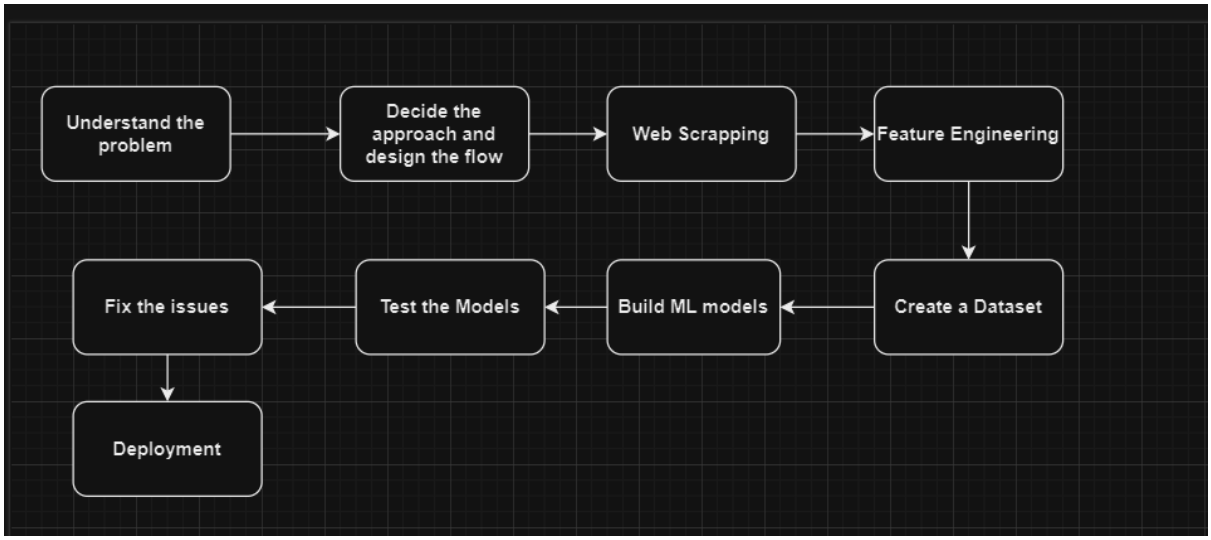
Phishing Detection Model

Flowchart



Understanding the problem statement

What is a Phishing Attack?

Phishing attack is a type of cyber threat and social engineering attack that is used to steal confidential data such as credentials, credit card info, sensitive personal data or money and it targets human vulnerabilities.

Problems:

- Phishing is a common problem
- It welcomes financial loss
- Targets all internet users
- Easy to perform do not require extra skills
- Social Engineering Techniques could use manuplation techniques

Solutions:

- User education
- Software Systems-Preventing attacks through software

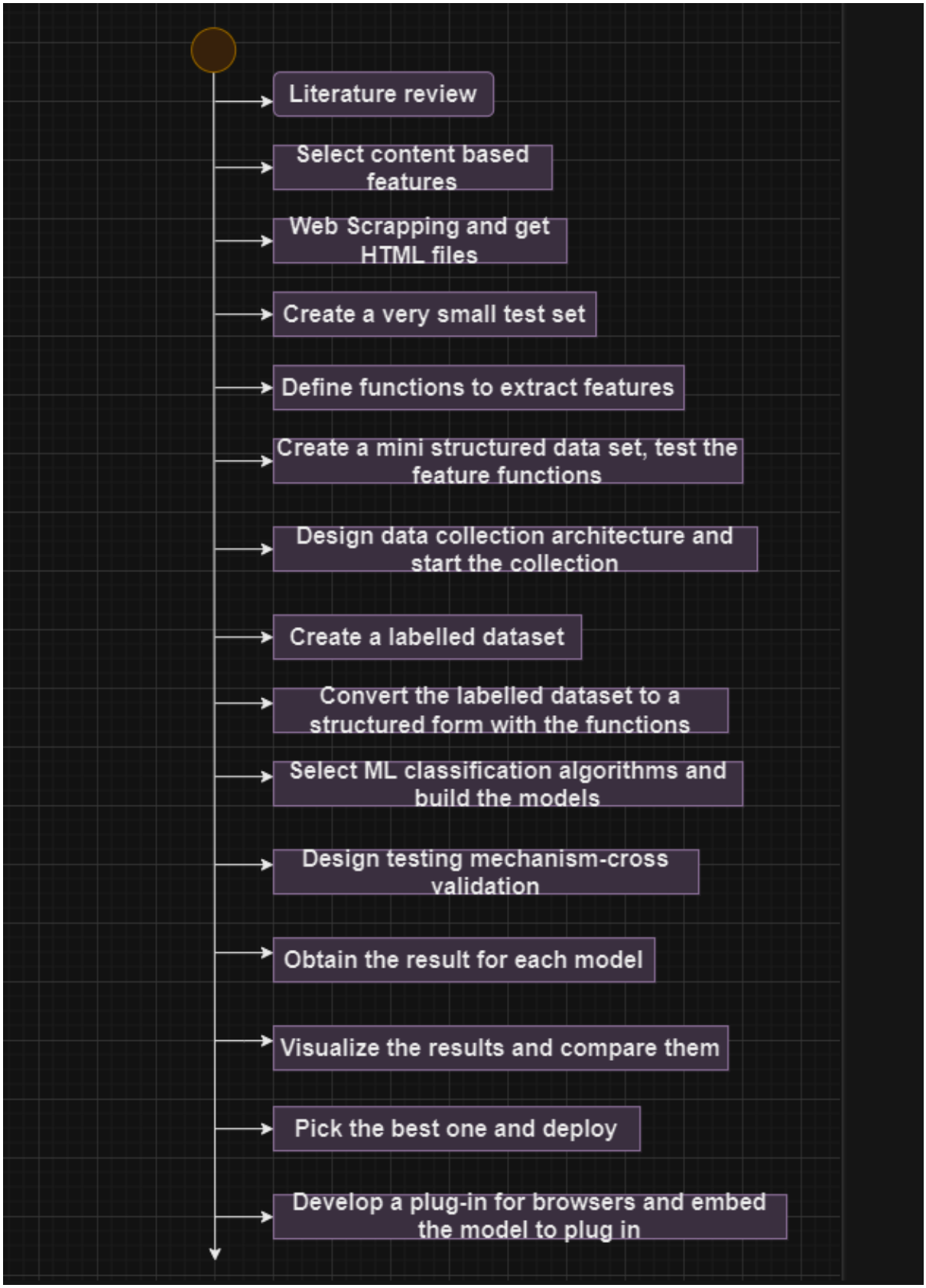
Could use

1. Blacklist/Whitelist: Whitelist defines approved entities that are permitted to access. Blacklist defines prohibited entities.
2. Rule-based-Uses a predefined set of rules or heuristics to identify and block such websites.
3. Heuristic-Examining urls for characteristics such as domain,primary domain, subdomain ,and path domain
4. Machine learning-using svm, neural network ,decision tree using these kind of models to detect attacks.

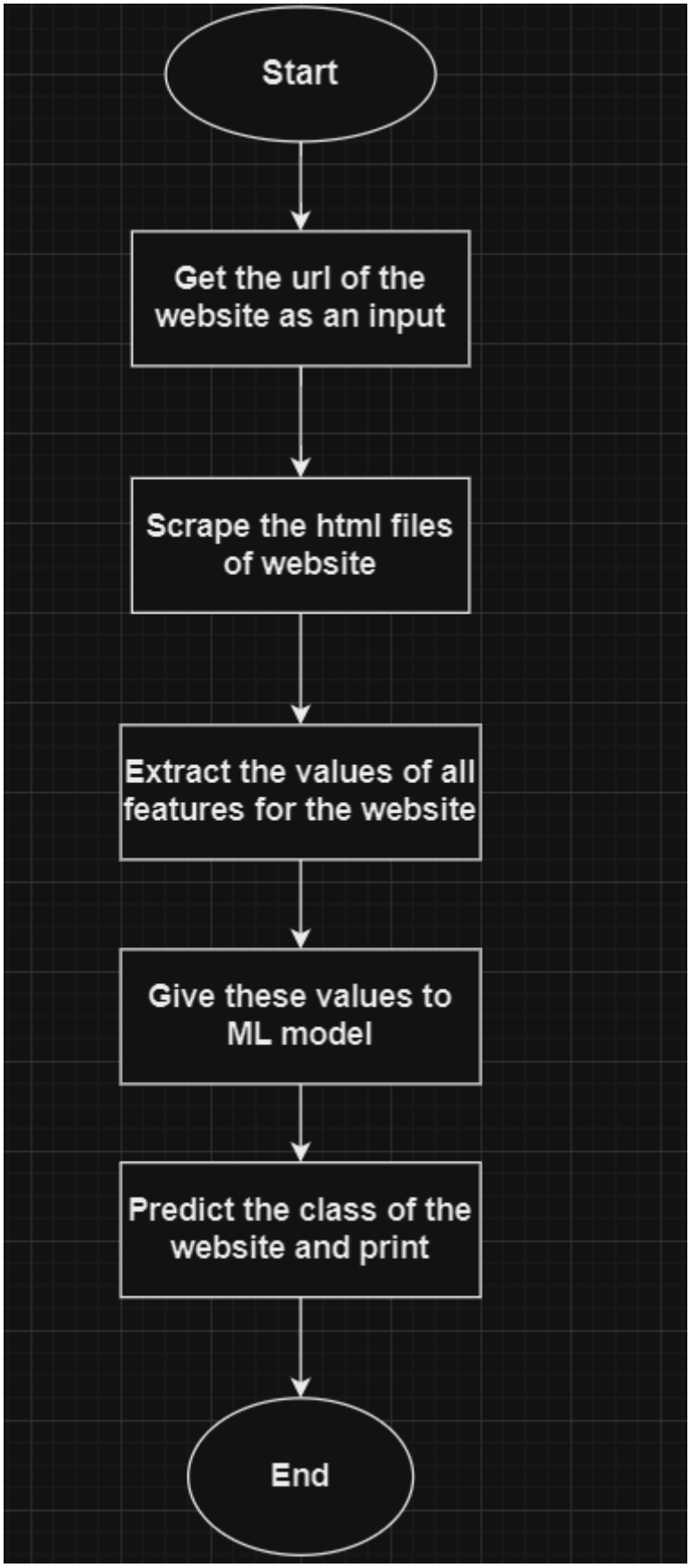
   Machine Learning classified as
   - Url based
   - **Content Based**
   - Visual Similarity
   - Hybrid

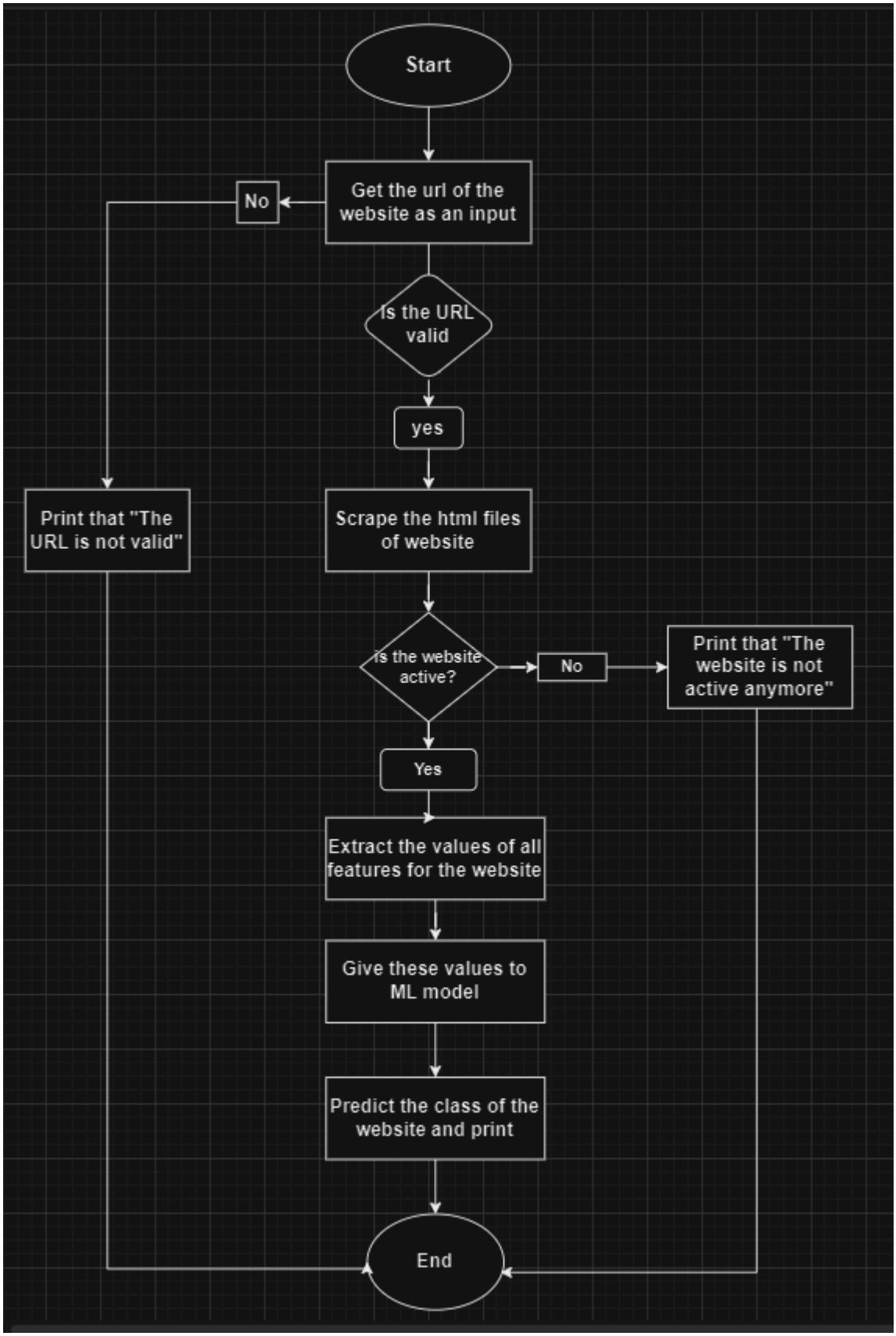
     Here we are gonna deal with content based phishing attacks inorder to put models and to show graphs this following type is taken.

Flow diagram



Web Scrapping




Start
Get the url of the website as an input
No
Is the URL valid
yes
Print that "The URL is not valid"
Scrape the html files of website
Is the website active?
No
Print that "The website is not active anymore"
Yes
Extract the values of all features for the website
Give these values to ML model
Predict the class of the website and print
End

Using dataset to scrape the models

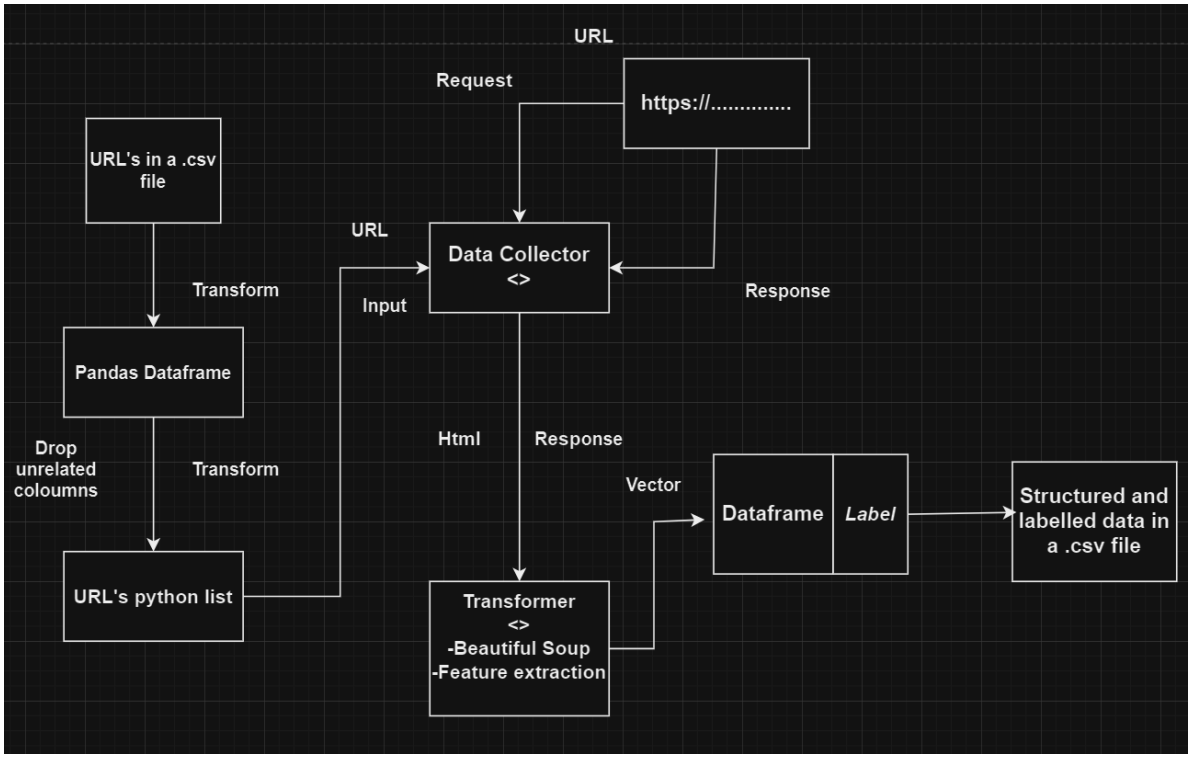
1. Create a folder that saves these then
2. define a function that scrapes and returns it
3. define a function to save files of the scraped webpage in a directory
4. define a url list variable
5. define a function which takes the url list and run step2 and 3 for each url

Feature Extraction

Binary	**Quantitative**
Has_title	Number_of_inputs
Has_input	Number_of_buttons
Has_button	Number_of_images
Has_ Video image	Number_of_option
Has_submit	Number_of_list
Has_link	Number_of_TH
Has_password	Number_of_TR
Has_email_input	Number_of_href
Has_hidden_elements	Number_of_paragraph
has_audio	Number_of_script
Has_video	Length_of_title

1. Define a function that opens a html file and returns the content.
2. Define a function that creates a BeautifulSoup object.
3. Define a function that creates a vector by running all feature functions for the soup object.
4. Run steps 1,2,3 for all html files and create a 2-D array.
5. Create a database by using 2-D array.

Creating a Dataset



Build ML models

Data collection Problems

Out of entire datasets 70-75% would remain legitimate and rest phished.

- 70-75% of legitimate websites would show these problems:
    1. HTTP connection problem
    2. Connection Timeout
    3. Exceeded maximum try

- 25-30% of phished websites have a shorter life span hence shows:
    1. Http connection is not successful
    2. Website is not active anymore

To build ML models used scikit-learn to build them
Steps to be followed:

1. Import libraries
2. Read the csv files and create pandas data frames

3. Combine legitimate and phishing data frames ,and shuffle
4. Remove url and remove duplicates then we can create X and Y for this models, supervised learning
5. Split data to train and test
6. Create a ml model using sklearn
7. Train the model
8. Make some predictions using test data
9. Create a confusion matrix and tn, tp, fn, fp(Here tn=true negative, tp=true positive, fn=false negative, fp=false positive)
10. Calculate accuracy, precision and recall scores

accuracy=(tp+tn)/ (tp+tn+fp+fn

precision=tp/ (tp+fp)

recall=tp/ (tp+fn)

Building ML models

K-Fold cross validation(k=5)

By using models like decision tree, random forest, Ada Boost, Support vector, Gaussian Naïve Bayees

The following accuracy, precision, recall can be calculated as:

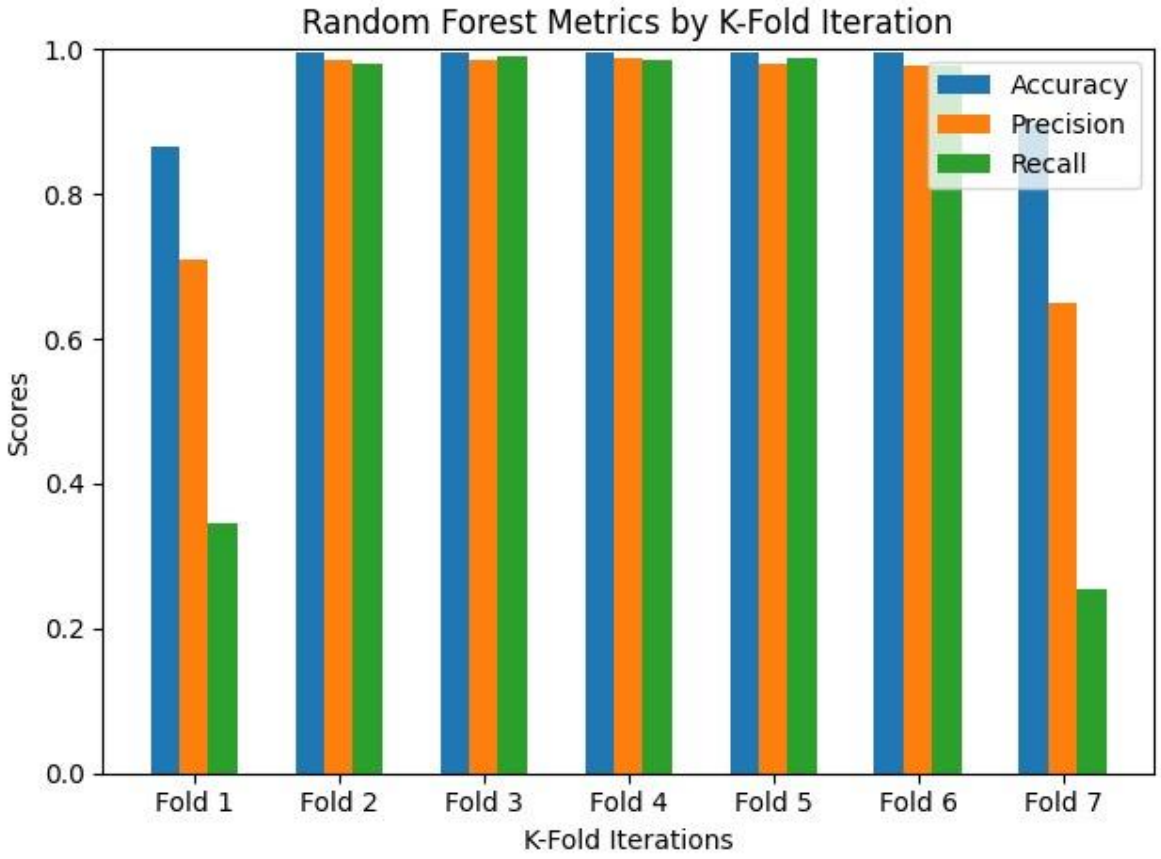
For example I take up Naivee Bayees (NB in short)

NB_accuracy=sum(nb_accuracy_list)/len(nb_accuracy_list)

NB_precision=sum(nb_precision_list)/ len(nb_precision_list)

NB_recall=sum(nb_recall_list)/ len(nb_recall_list)

The following result would look like this:



(The following graph is for Random Forest model. Shows accuracy, precision and recall)

Developing and Deploying

Steps to be followed**:**

1. Install Django and check the version
2. Create a Django app file and run
3. Design the app mockup using excalidraw
4. Start to develop import packages
5. Add the ML project details inside an expander
6. Add selection box to get user's choice for ML models
7. Get the URL from user and predict if its phished or legitimate
8. Refresh the page and fix minor/syntax errors
9. App is ready in the localhost and test the legitimate website
10. Add example phishing urls and test phished websites