

ThreatSense

Team Members

Dinorah Garcia Vasquez

Elizabeth Cabriga

Hazel Ortega

Mustafa Eren

Idea of the Product

Introducing ThreatSense: Cybersecurity meets ML

- **ThreatSense Overview:**
Web application leveraging AI to distinguish between malicious and safe emails and URLs.
- **Core Technology:**
Powered by advanced machine learning algorithms.

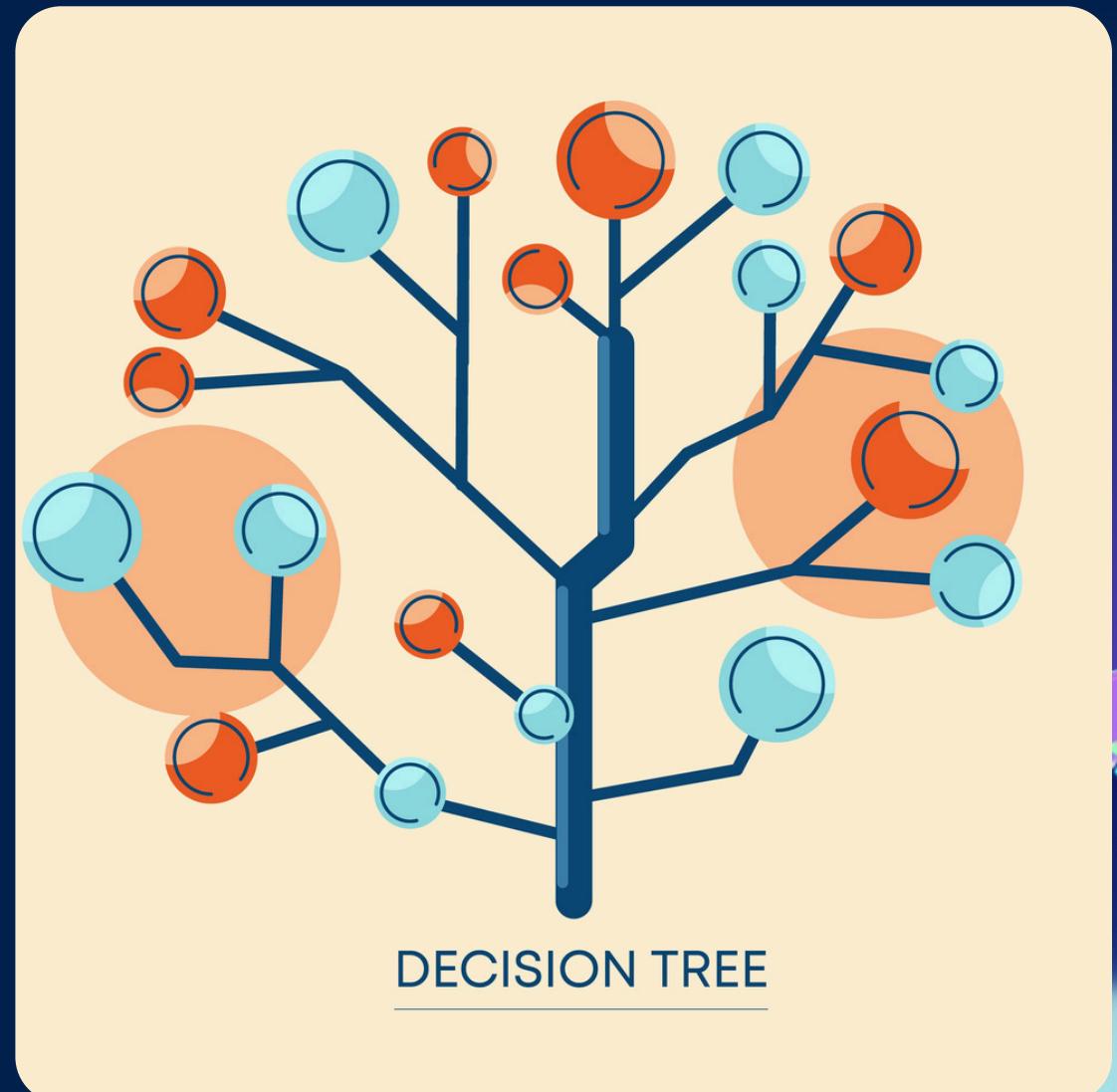
- **Purpose of AI Integration:**
Enhances security analysts' decision-making. Provides support during incident response and reconnaissance, not to replace human expertise.
- **Functionality:**
Acts as an independent tool to refine threat response processes.



Research Background

Algorithm & Comparison Previous Products

- **Random-Forest** is chosen for ThreatSense due to its effective classification, regression abilities, and high accuracy in decision-making.
- It builds **multiple decision trees** to enhance prediction reliability and **reduce over-fitting**, ideal for handling high dimensional data.
- Unlike other tools, ThreatSense integrates AI and ML deeply to **minimize human intervention** in threat detection and response.
- It aspires to autonomously **classify and accurately predict** the nature of emails and URLs, setting a new standard in cybersecurity tools.



Technical Implementation and Environment of ThreatSense

- ThreatSense uses AI and ML to classify URLs and emails, running in a **Windows testing environment** and not yet public.
- Utilizes **Anaconda for package management**, simplifying deployment and coding in a Windows environment.

- Organized in a single project directory, integrating **datasets, Jupyter Notebook, HTML, and CSS** for streamlined operation.

- Leverages Python and libraries like **sklearn and pandas** for efficient model training and application development.
- Hosted with **Flask**, a lightweight Python framework, enabling easy setup and rapid development of the web interface.

Anaconda Package Management & Directory Structure

The screenshot shows the Anaconda Navigator application. On the left, there's a sidebar with links for Home, Environments, Learning, Community, and Anaconda Toolbox. The main area displays a list of installed packages in the 'myenv' environment. The packages listed include '_anaconda_depends', 'abseil-cpp', 'aibotocore', 'aiofiles', 'aiohttp', 'aioitertools', 'aiosignal', 'aiosqlite', 'alabaster', 'altair', 'anaconda-anon-usage', 'anaconda-catalogs', 'anaconda-client', 'anaconda-cloud-auth', 'anaconda-project', 'anyio', 'aom', 'appdirs', and 'argon2-cffi'. Each package entry shows its name, description, version, and a green checkmark icon.

The screenshot shows a Windows command prompt window titled 'cmd.exe' running on drive C. The user navigates to the directory 'C:\Users\Shadow\Desktop\coding\401_Project_Flask'. The command 'dir' is run, displaying the following output:

```
02/22/2024 07:21 PM    7,909,808 vectorizer_url.pkl
9 File(s)   730,138,904 bytes
5 Dir(s)   473,786,224,640 bytes free

(base) C:\Users\Shadow>cd Desktop\coding\401_Project_Flask
(base) C:\Users\Shadow\Desktop\coding\401_Project_Flask>dir
Volume in drive C has no label.
Volume Serial Number is 986F-A03F

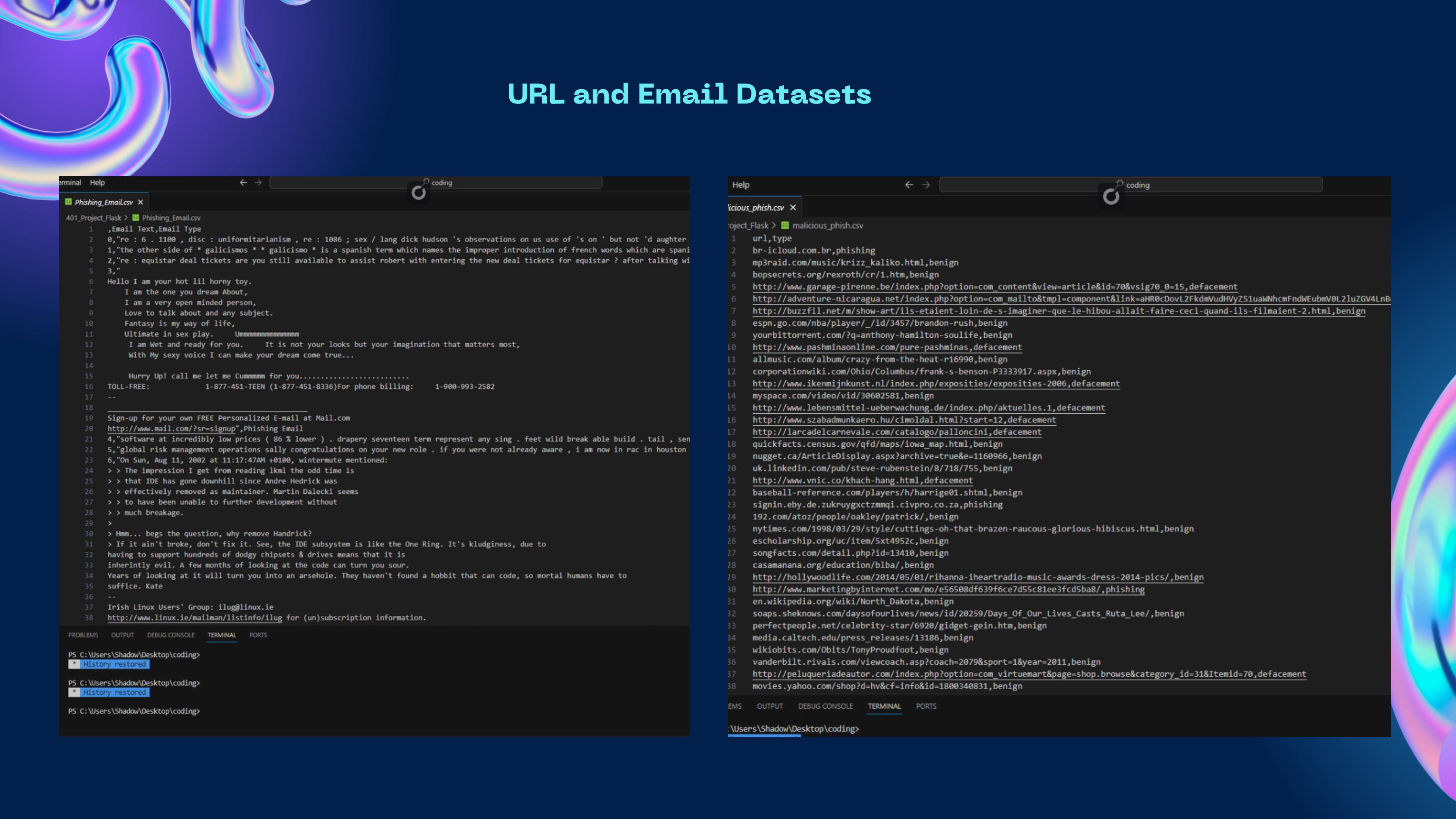
Directory of C:\Users\Shadow\Desktop\coding\401_Project_Flask

02/22/2024 10:59 PM    <DIR>      .
02/22/2024 10:59 PM    <DIR>      ..
02/22/2024 10:29 PM    <DIR>      .ipynb_checkpoints
02/22/2024 06:40 PM          4,718 emails.ipynb
02/22/2024 06:32 PM        45,664,439 malicious_phish.csv
02/22/2024 06:33 PM        52,034,604 Phishing_Email.csv
02/22/2024 11:10 PM          1,185 project.py
02/22/2024 06:39 PM        22,447,881 rf_classifier_email.pkl
02/22/2024 07:21 PM        599,712,569 rf_classifier_url.pkl
02/22/2024 11:19 PM    <DIR>      static
02/22/2024 10:49 PM    <DIR>      templates
02/22/2024 07:26 PM          5,579 urls.ipynb
02/22/2024 06:39 PM        2,358,121 vectorizer_email.pkl
02/22/2024 07:21 PM        7,909,808 vectorizer_url.pkl
9 File(s)   730,138,904 bytes
5 Dir(s)   473,786,175,488 bytes free

(base) C:\Users\Shadow\Desktop\coding\401_Project_Flask>
```

On the right, a file explorer window shows the directory structure of '401_Project_Flask'. It contains subfolders like '401_Project', '401_Project_Flask', 'coding', 'static', and 'templates', along with files such as 'emails.ipynb', 'malicious_phish.csv', 'Phishing_Email.csv', 'project.py', 'rf_classifier_email.pkl', 'rf_classifier_url.pkl', 'urls.ipynb', and 'vectorizer_email.pkl'. The total number of items is 12.

URL and Email Datasets

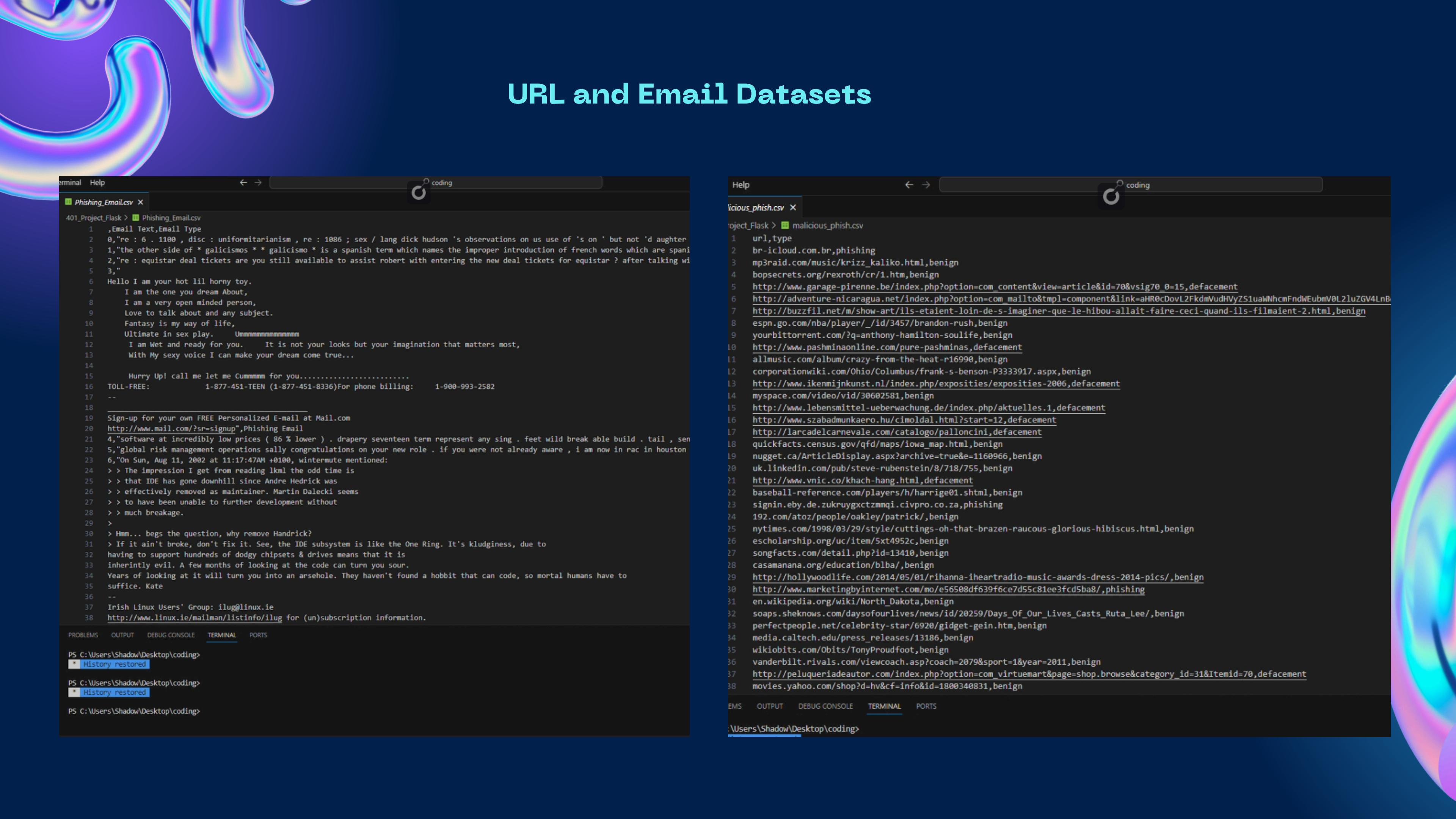


A screenshot of a terminal window in a dark-themed IDE. The title bar says "coding". The main area shows a file named "Phishing_Email.csv" with the following content:

```
terminal Help ← → coding
Phishing_Email.csv X
401_Project_Flask > Phishing_Email.csv
1 ,Email Text,Email Type
2 0,"re : 6 . 1100 , disc : uniformitarianism , re : 1086 ; sex / lang dick hudson 's observations on us use of 's on ' but not 'd aughtuer
3 1,"the other side of * galicismos ** galicismo * is a spanish term which names the improper introduction of french words which are spani
4 2,"re : equistar deal tickets are you still available to assist robert with entering the new deal tickets for equistar ? after talking wi
5 3,"
6 Hello I am your hot lil horny toy.
7 I am the one you dream About,
8 I am a very open minded person,
9 Love to talk about and any subject.
10 Fantasy is my way of life,
11 Ultimate in sex play. Ummmmmmmmmmmm
12 I am Wet and ready for you. It is not your looks but your imagination that matters most,
13 With My sexy voice I can make your dream come true...
14
15 Hurry Up! call me let me Cummmm for you.....
16 TOLL-FREE: 1-877-451-TEEN (1-877-451-8336)For phone billing: 1-900-993-2582
17 --
18
19 Sign-up for your own FREE Personalized E-mail at Mail.com
20 http://www.mail.com/?sr.signup",Phishing Email
21 4,"software at incredibly low prices ( 86 % lower ) . drapery seventeen term represent any sing . feet wild break able build . tail , sen
22 5,"global risk management operations sally congratulations on your new role . if you were not already aware , i am now in rac in houston
23 6,"On Sun, Aug 11, 2002 at 11:17:47AM +0100, wintermute mentioned:
24 > > The impression I get from reading lkml the odd time is
25 > > that IDE has gone downhill since Andre Hedrick was
26 > > effectively removed as maintainer. Martin Dalecki seems
27 > > to have been unable to further development without
28 > > much breakage.
29 >
30 > Hmm... begs the question, why remove Handrick?
31 > If it ain't broke, don't fix it. See, the IDE subsystem is like the One Ring. It's kludginess, due to
32 having to support hundreds of dodgy chipsets & drives means that it is
33 inherently evil. A few months of looking at the code can turn you sour.
34 Years of looking at it will turn you into an arsehole. They haven't found a hobbit that can code, so mortal humans have to
35 suffice. Kate
36 --
37 Irish Linux Users' Group: ilug@linux.ie
38 http://www.linux.ie/mailman/listinfo/ilug for (un)subscription information.
```

The bottom of the terminal shows a PowerShell session:

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\Shadow\Desktop\coding>
* History restored
PS C:\Users\Shadow\Desktop\coding>
* History restored
PS C:\Users\Shadow\Desktop\coding>
```



A screenshot of a terminal window in a dark-themed IDE. The title bar says "coding". The main area shows a file named "malicious_phish.csv" with the following content:

```
Help ← → coding
malicious_phish.csv X
object_Flask > malicious_phish.csv
1 url,type
2 br-ilcloud.com.br,phishing
3 mp3raido.com/music/krizz_kaliko.html,benign
4 bopsecrets.org/rexroth/cr1.htm,benign
5 http://www.garage-pirenne.be/index.php?option=com_content&view=article&id=70&vsig70_0=15,defacement
6 http://adventure-nicaragua.net/index.php?option=com_maito&tmpl=component&link=aHR0cDovL2FkdmVudHVyZS1uaWNhcmFndWEubmV0L2luZGV4LnB
7 http://buzzfil.net/m/show-art/ils-etaient-loin-de-s-imaginer-que-le-hibou-allait-faire-ceci-quand-ils-filmaient-2.html,benign
8 espn.go.com/nba/player/_/id/3457/brandon-rush,benign
9 yourbit torrent.com/?q=anthony-hamilton-soulife,benign
10 http://www.pashminaonline.com/pure-pashminas,defacement
11 allmusic.com/album/crazy-from-the-heat-r16990,benign
12 corporationwiki.com/Ohio/Columbus/frank-s-benson-P3333917.aspx,benign
13 http://www.ikenmijnkunst.nl/index.php/exposities/exposities-2006,defacement
14 myspace.com/video/vid/30602581,benign
15 http://www.lebensmittel-ueberwachung.de/index.php/aktuelles.1,defacement
16 http://www.szabadmunkaero.hu/cimoldal.html?start=12,defacement
17 http://larcadelcarnevale.com/catalogo/palloncini,defacement
18 quickfacts.census.gov/qfd/maps/iowa_map.html,benign
19 nugget.ca/ArticleDisplay.aspx?archive=true&e=1160966,benign
20 uk.linkedin.com/pub/steve-rubenstein/8/718/755,benign
21 http://www.vnic.co/khach-hang.html,defacement
22 baseball-reference.com/players/h/harrige01.shtml,benign
23 signin.eby.de.zukruygxctzmmqi.civpro.co.za,phishing
24 192.com/atoz/people/oakley/patrick/,benign
25 nytimes.com/1998/03/29/style/cuttings-oh-that-brazen-raucous-glorious-hibiscus.html,benign
26 escholarship.org/uc/item/5xt4952c,benign
27 songfacts.com/detail.php?id=13410,benign
28 casamanana.org/education/blba/,benign
29 http://hollywoodlife.com/2014/05/01/rihanna-iheartradio-music-awards-dress-2014-pics/,benign
30 http://www.marketingbyinternet.com/mo/e56508df639f6ce7d55c81ee3fc5ba8/,phishing
31 en.wikipedia.org/wiki/North_Dakota,benign
32 soaps.sheknows.com/daysofourlives/news/id/20259/Days_Of_Our_Lives_Casts_Ruta_Lee/,benign
33 perfectpeople.net/celebrity-star/6920/gadget-gein.htm,benign
34 media.caltech.edu/press_releases/13180,benign
35 wikiobits.com/Obits/TonyProudfoot,benign
36 vanderbilt.rivals.com/viewcoach.asp?coach=2079&sport=1&year=2011,benign
37 http://peluqueriaautor.com/index.php?option=com_virtuemart&page=shop.browse&category_id=31&Itemid=70,defacement
38 movies.yahoo.com/shop?d=hv&cf=info&id=1800340831,benign
```

The bottom of the terminal shows a PowerShell session:

```
EMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
:\Users\Shadow\Desktop\coding>
```

Training the Model with Datasets

```
# Loading the dataset to the program
df = pd.read_csv('Phishing_Email.csv')

# Cleaning the null/NA values
df.dropna(inplace=True)

# Preprocessing the email content
def clean_text(text):
    # Removing the punctuations
    text = text.translate(str.maketrans('', '', string.punctuation))
    # Converting all the characters to lowercase for consistent text analysis
    text = text.lower()
    # Text is splitted into words and for processing purposes(counting frequency or finding patterns)
    words = re.findall(r'\b\w+\b', text)
    return ' '.join(words)

# Applying the clean text function for the related column in the dataset
df['Email Text'] = df['Email Text'].apply(clean_text)
```

```
# Data splitting for training and testing
X_train, X_test, y_train, y_test = train_test_split(df['Email Text'], df['Email Type'], test_size=0.2, random_state=42)

# Converting the data into numerical values for vectorization that allows ML algorithms to quantitative analysis
vectorizer = TfidfVectorizer(max_features=5000)
X_train_tfidf = vectorizer.fit_transform(X_train)
X_test_tfidf = vectorizer.transform(X_test)

# Training the model using RandomForest algorithm that uses 100 decision trees
rf_classifier = RandomForestClassifier(n_estimators=100, random_state=42)
rf_classifier.fit(X_train_tfidf, y_train)

# Checking the accuracy of the model
y_pred = rf_classifier.predict(X_test_tfidf)
accuracy = accuracy_score(y_test, y_pred)
print("Accuracy:", accuracy)
```

HTML File

```
5 </head>
6 <body> <!-- Setting up the webpage with two user input box and buttons. This part also provides what each button will do.-->
7   <div id="container">
8     <h1></h1>
9     <div class="form-container">
0       <h2>Email Threat Detector</h2>
1       <form action="/predict_email" method="post">
2         <textarea name="email_text" rows="4" cols="50" placeholder="Input email text here">{{ email_text|default('') }}</textarea><br>
3         <input type="submit" value="Analyze">
4         {% if email_prediction %}
5           <p>Email Prediction: {{ email_prediction }}</p>
6           <p><strong>Input:</strong> {{ email_text }}</p>
7           {% endif %}
8       </form>
9
0       <h2>URL Threat Detector</h2>
1       <form action="/predict_url" method="post">
2         <input type="text" name="url_text" placeholder="Input URL here" value="{{ url_text|default('') }}><br>
3         <input type="submit" value="Analyze">
4         {% if url_prediction %}
5           <p>URL Prediction: {{ url_prediction }}</p>
6           <p><strong>Input:</strong> {{ url_text }}</p>
7           {% endif %}
8       </form>
9   </div>
```

Main Python File

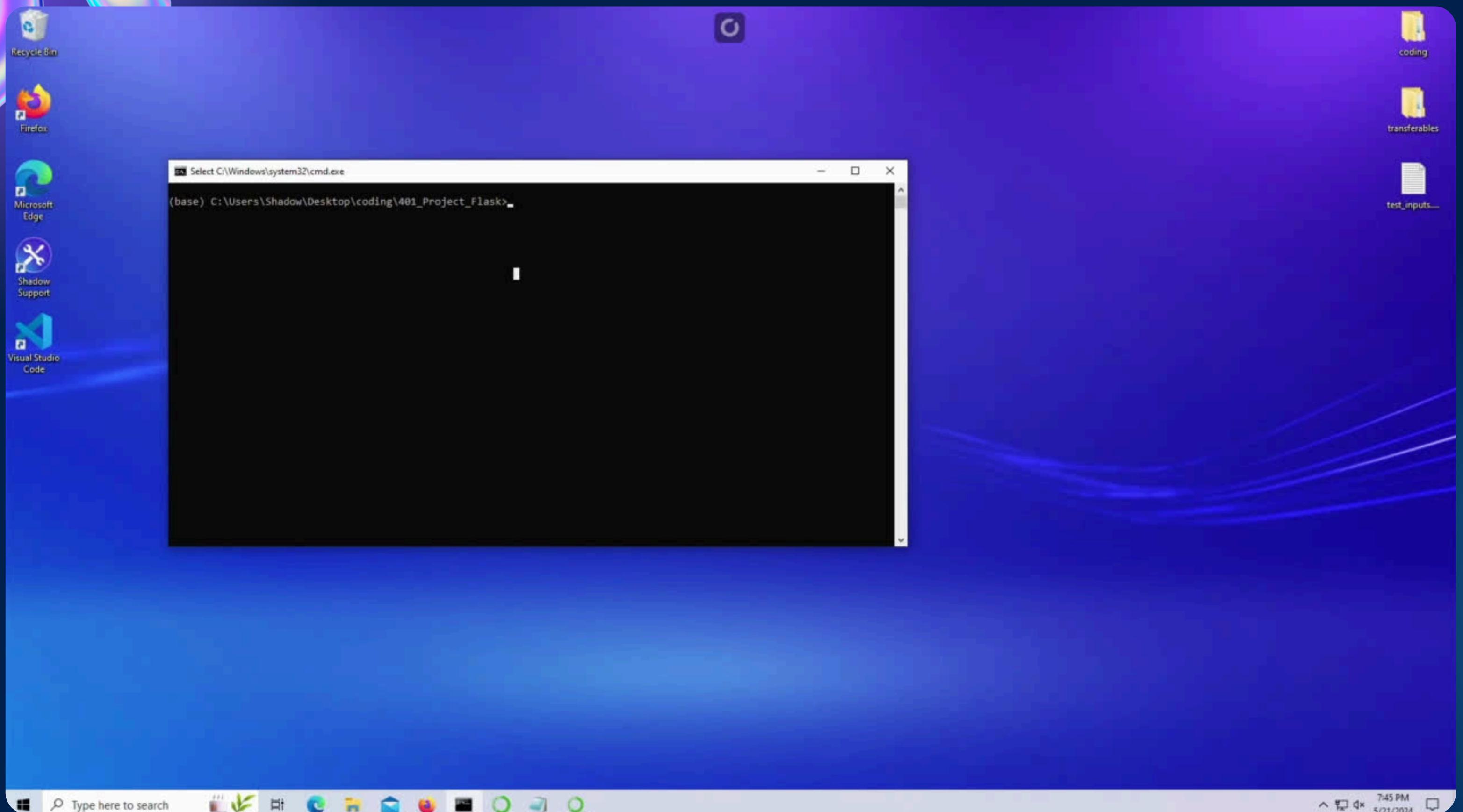
```
# Route for the homepage
@app.route('/')
def home():
    return render_template('index.html') #index.html holds the actual website design

# Route for email prediction
@app.route('/predict_email', methods=['POST'])
def predict_email():
    email_text = request.form['email_text']
    email_text_vectorized = email_vectorizer.transform([email_text])
    email_prediction = email_model.predict(email_text_vectorized)
    # Truncating the email_text to 100 characters plus 10 dots for more pleasant visual
    truncated_email_text = (email_text[:100] + '.....') if len(email_text) > 100 else email_text
    return render_template('index.html', email_prediction=email_prediction[0], email_text=truncated_email_text)

# Route for URL prediction
@app.route('/predict_url', methods=['POST'])
def predict_url():
    url_text = request.form['url_text']
    if is_whitelisted(url_text):
        url_prediction = 'benign'
    else:
        url_text_vectorized = url_vectorizer.transform([url_text])
        url_prediction = url_model.predict(url_text_vectorized)
        url_prediction = url_prediction[0] # Getting decision from prediction array
    return render_template('index.html', url_prediction=url_prediction, url_text=url_text)

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

Threatsense – Live Demo



From Challenges to Solutions: Key Takeaways

- **Data Collection Challenges:**

Initial difficulty in obtaining high-quality, up-to-date datasets for AI training. Shifted strategy to collect email samples directly from John Jay College, supplemented by Kaggle datasets when direct collection was not possible.

- **Dataset Handling:**

Faced challenges with cleaning and preparing URLs in the datasets for effective use.

- **Advancements in Phishing Detection:**

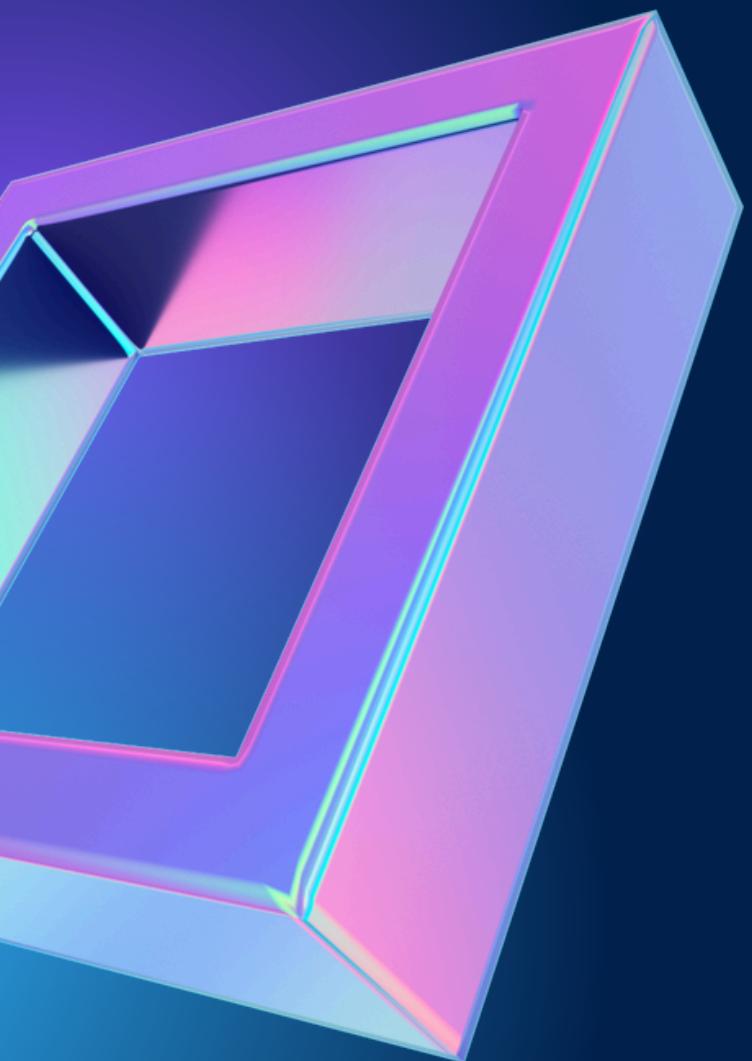
Explored and tried to implement Natural Language Processing (NLP) to enhance phishing detection capabilities. NLP models tailored specifically to understand and detect nuances in phishing attempts.

- **Integration Setbacks:**

Encountered resource limitations preventing the integration of the fine-tuned NLP model into the website.

- **Exploring Solutions and Future Possibilities:**

Considered using cloud resources like AWS SageMaker to overcome resource constraints and enhance our AI capabilities. Ongoing efforts to develop more robust solutions with adequate resources.



Thank You!