Business Problem Understanding Document

1. Business Problem Summary

Phishing websites are fraudulent sites that attempt to steal sensitive information such as usernames, passwords, and financial details by masquerading as legitimate websites. The ability to detect phishing websites accurately is crucial for cybersecurity, as phishing attacks lead to financial losses, identity theft, and reputational damage for individuals and organizations.

The goal of this project is to develop a **phishing website detection model** using data-driven techniques. By analyzing various website features, the model will help identify phishing attempts and enhance online security. This solution is particularly valuable for businesses, financial institutions, and security firms seeking to protect users from cyber threats.

2. Key Insights from Literature on Phishing

- **Common Characteristics of Phishing Websites:**
  - Use of misleading domain names (e.g., typosquatting, similar-looking URLs).
  - Excessive redirections and shortened URLs.
  - Presence of deceptive pop-ups and fake login forms.
  - Lack of HTTPS security and valid SSL certificates.
- **Detection Challenges:**
  - Phishing tactics evolve rapidly, making rule-based detection less effective.
  - Traditional blacklist-based approaches fail to detect new phishing sites.
  - Feature extraction from website content, URL structure, and HTML is complex.
- **Potential Solutions:**
  - **Machine Learning-Based Detection:** Utilizing classifiers trained on website attributes.
  - **Heuristic Analysis:** Identifying suspicious patterns in URLs, domain registration, and HTTPS usage.
  - **Real-Time Monitoring:** Combining AI with real-time scanning to detect zero-day phishing sites.

Dataset Exploration Report

1. Overview of the Dataset

The dataset contains various features extracted from website URLs, HTML structure, and security indicators. The primary objective is to use these features to classify websites as either **phishing (malicious) or legitimate (safe).**

- **Number of Features:** [To be determined from dataset analysis]
- **Types of Data:**
  - **Numerical Features:** Metrics such as domain age, URL length, number of subdomains.
  - **Categorical Features:** Presence of HTTPS, presence of login forms.
  - **Binary Features:** Indicators like "Has IP address in URL" (Yes/No).
- **Target Variable Distribution:** The dataset contains labeled data indicating whether a website is phishing (1) or legitimate (0). A distribution analysis will help identify any class imbalances that could affect model performance.

2. Description of Individual Features

Each feature contributes to phishing detection by identifying suspicious behaviors or security flaws. Key features include:

- **URL-Based Features:**
  - Length of URL, presence of special characters, number of subdomains.
  - Whether the domain uses an IP address instead of a hostname (common in phishing sites).
- **Domain and Hosting Information:**
  - Age of the domain (newly registered domains are often phishing sites).
  - WHOIS information consistency and SSL certificate validity.
- **Website Content & HTML Features:**
  - Presence of deceptive elements like fake login forms.
  - Frequency of redirects and presence of pop-ups.
- **Network and Security Indicators:**
  - Use of HTTPS (secure connection vs. insecure HTTP).
  - Presence in known phishing blacklists.

Conclusion

This dataset exploration helps establish a foundation for phishing detection by identifying important patterns in website attributes.

--- Internship project (phishing data set) ---

```
[22]: import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
import seaborn as sns
from scipy import stats
from sklearn.preprocessing import LabelEncoder,OneHotEncoder,StandardScaler,MinMaxScaler
df=pd.read_csv('dataset_phishing.csv')
```

```
[2]: df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 11430 entries, 0 to 11429
Data columns (total 89 columns):
 #   Column                     Non-Null Count  Dtype
---  ------                     --------------  -----
 0   url                        11430 non-null  object
 1   length_url                 11430 non-null  int64
 2   length_hostname            11430 non-null  int64
 3   ip                         11430 non-null  int64
 4   nb_dots                    11430 non-null  int64
 5   nb_hyphens                 11430 non-null  int64
 6   nb_at                      11430 non-null  int64
```

```
[3]: df.isnull()
```

	url	length_url	length_hostname	ip	nb_dots	nb_hyphens	nb_at	nb_qm	nb_and	nb_or	...	domain_in_title	domain_with_copyright	whois_registered_domain	domain_registration_length
0	False	False	False	False	False	False	False	False	False	False	...	False	False	False	False
1	False	False	False	False	False	False	False	False	False	False	...	False	False	False	False
2	False	False	False	False	False	False	False	False	False	False	...	False	False	False	False
3	False	False	False	False	False	False	False	False	False	False	...	False	False	False	False
4	False	False	False	False	False	False	False	False	False	False	...	False	False	False	False
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
11425	False	False	False	False	False	False	False	False	False	False	...	False	False	False	False
11426	False	False	False	False	False	False	False	False	False	False	...	False	False	False	False
11427	False	False	False	False	False	False	False	False	False	False	...	False	False	False	False
11428	False	False	False	False	False	False	False	False	False	False	...	False	False	False	False
11429	False	False	False	False	False	False	False	False	False	False	...	False	False	False	False

11430 rows × 89 columns

```
missing_values=df.isnull().sum()
print("\n missing_values:")
missing_values>0
```

```
 missing_values:
url                  False
length_url           False
length_hostname      False
ip                   False
nb_dots              False
                     ...
web_traffic          False
dns_record           False
google_index         False
page_rank            False
status               False
Length: 89, dtype: bool
```

```
# Check for duplicate entries
duplicates = df.duplicated().sum()
print(f"\nNumber of duplicate rows: {duplicates}")
```

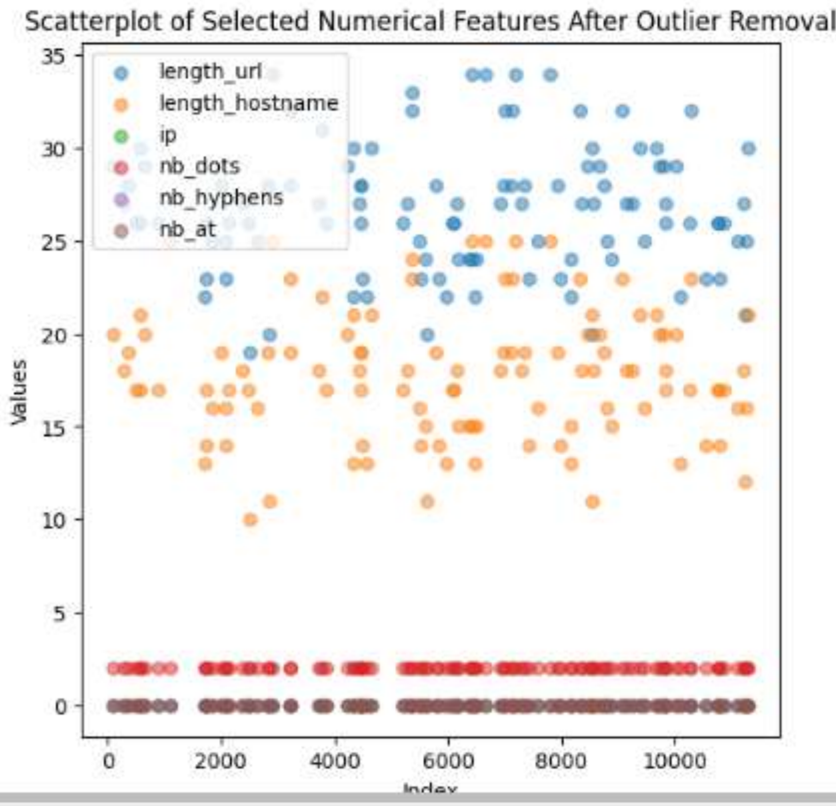
```
Number of duplicate rows: 0
```

```
# Select numerical columns for outlier detection
numerical_cols = df.select_dtypes(include=['int64', 'float64']).columns.tolist()
numerical_cols.remove('page_rank')
```

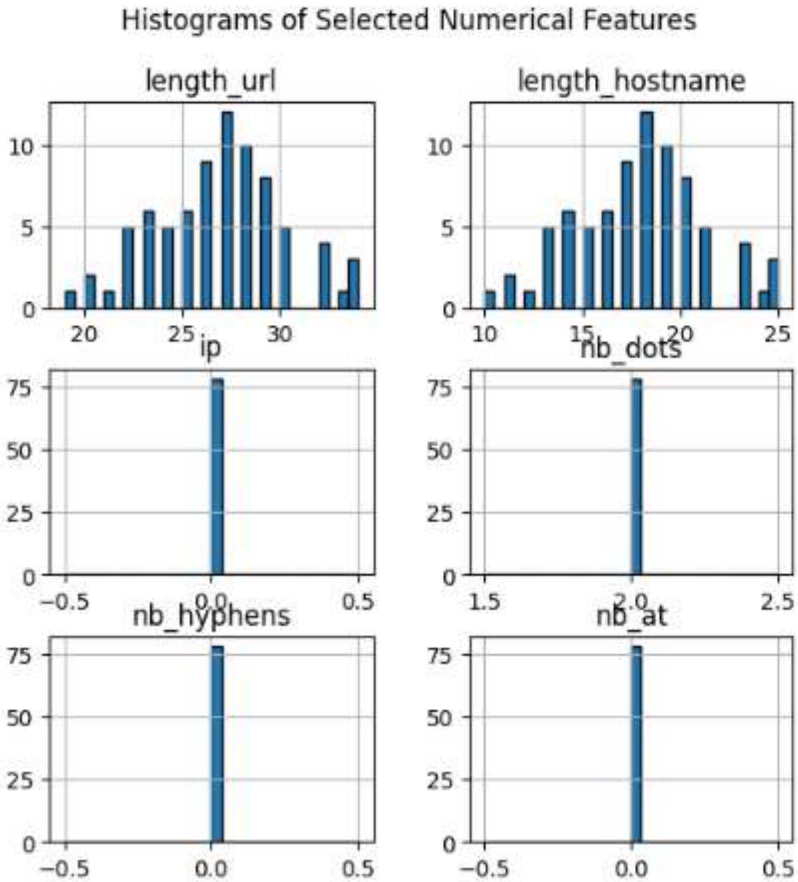
```
# Calculate IQR and remove outliers
for col in numerical_cols:
    Q1 = df[col].quantile(0.25)
    Q3 = df[col].quantile(0.75)
    IQR = Q3 - Q1
    lower_bound = Q1 - 1.5 * IQR
    upper_bound = Q3 + 1.5 * IQR
    df = df[(df[col] >= lower_bound) & (df[col] <= upper_bound)]

print("Outliers removed using IQR method.")
```

```
Outliers removed using IQR method.
```

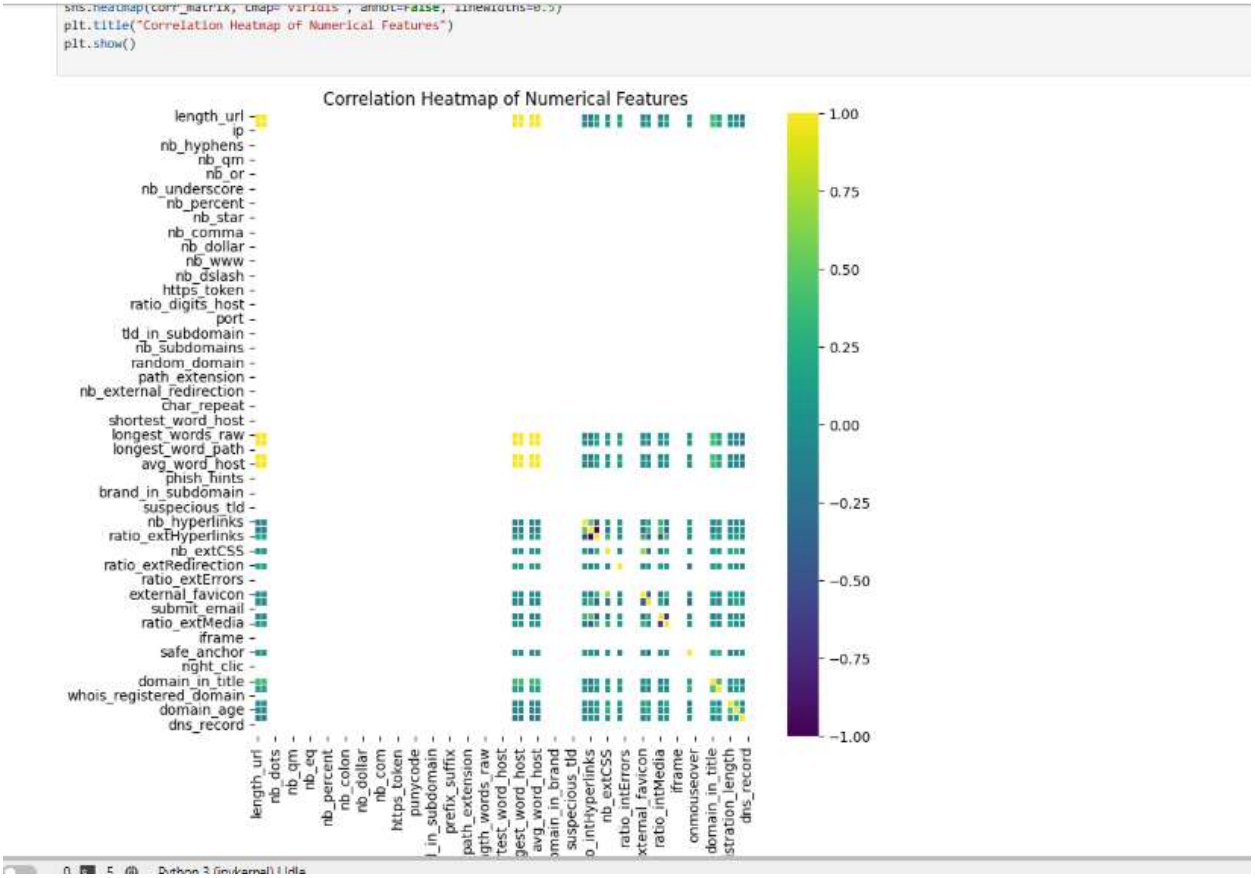
```python
# Scatterplot for outlier detection
plt.figure(figsize=(6, 6))
for col in numerical_cols[:6]:
    plt.scatter(df.index, df[col], label=col, alpha=0.5)
plt.legend()
plt.title("Scatterplot of Selected Numerical Features After Outlier Removal")
plt.ylabel("Values")
plt.xlabel("Index")
plt.show()
```



```
# Histograms for key numerical features
df[numerical_cols[:6]].hist(figsize=(6, 6), bins=30, edgecolor='black')
plt.suptitle("Histograms of Selected Numerical Features")
plt.show()
```



<u>Correlation heatmap:</u>



<u>Summary of the EDA Process in Theory</u>

This Python script performs **Exploratory Data Analysis (EDA)** on a phishing website dataset using **Pandas, Matplotlib, and Seaborn**. The main steps involved are:

1. **Loading the Dataset**
    - Reads the dataset from a CSV file into a Pandas DataFrame.
    - Displays basic information such as column names, data types, and the first few rows.
2. **Handling Missing Values**
    - Checks for missing values in each column.
    - If missing values are found, it visualizes them using a bar plot.
    - If no missing values exist, it prints a confirmation message.

3. **Outlier Detection and Removal Using IQR**
    - Identifies numerical columns (excluding `page_rank`).
    - Uses the **Interquartile Range (IQR) method** to detect and remove outliers:
        - Computes the 1st quartile (Q1) and 3rd quartile (Q3).
        - Calculates the **IQR** as `Q3 - Q1`.
        - Defines the lower and upper bounds as `Q1 - 1.5*IQR` and `Q3 + 1.5*IQR`, respectively.
        - Filters out values beyond these bounds to remove outliers.
4. **Visualization of Data Distributions**
    - **Boxplots**: Show the distribution of numerical features after outlier removal.
    - **Histograms**: Display the frequency distribution of selected numerical features.
5. **Correlation Analysis**
    - Computes the correlation matrix for numerical features.
    - Plots a **heatmap** using the **"viridis"** colormap to visualize relationships between features.
    - Helps identify highly correlated variables that may be redundant or important for classification.

Outcome of the EDA Process

- **Missing values**: Checked and handled if found.
- **Outliers**: Removed using IQR to improve data quality.
- **Data distribution**: Explored through histograms and boxplots.
- **Feature relationships**: Examined using a correlation heatmap.

This process ensures that the dataset is **cleaned, structured, and ready** for further analysis or machine learning modeling.