# INTERNSHIP PROJECT DOCUMENTATION

## Project Title

## Machine Learning–Based Spam & Phishing Detection System

---

**Internship Details**

- **Internship Domain:** Cybersecurity with Machine Learning
- **Project Type:** Individual Project
- **Technology Used:** Python, Machine Learning, NLP
- **Platform:** Google Colab
- **Dataset Source:** Public Spam Dataset (spam.csv)

---

### 1. Introduction

With the increasing reliance on digital communication, spam and phishing attacks have become a significant cybersecurity concern. These attacks often exploit human behavior through deceptive messages, leading to credential theft, financial loss, and data breaches.

Machine Learning (ML) combined with Natural Language Processing (NLP) provides effective techniques to automatically analyse and classify textual data, enabling early detection of malicious content.

This project focuses on developing a **machine learning–based spam and phishing detection system** capable of classifying messages as **legitimate or malicious** with high accuracy.

---

### 2. Problem Statement

Traditional rule-based spam filters are often ineffective against evolving phishing techniques and contextual attacks. There is a need for an intelligent system that can:

- Automatically analyze message content
- Identify spam or phishing attempts
- Reduce false positives for legitimate messages

The objective of this project is to design and implement a **machine learning model** that accurately classifies messages using text-based features.

---

**3. Objectives of the Project**

- To build a spam and phishing detection model using machine learning
- To apply NLP techniques for text preprocessing and feature extraction
- To evaluate the model using standard performance metrics
- To test the model using real-world message content

---

**4. Dataset Description**

The dataset used in this project consists of labeled text messages categorized as **spam** or **ham (legitimate)**.

- **File Name:** spam.csv
- **Columns Used:**
    - label → spam / ham
    - message → text content

Label Encoding:

- 0 → Legitimate (Ham)
- 1 → Spam / Phishing

---

Fig1. ***Dataset Preview Screenshot Here***

```
df = pd.read_csv("/content/spam.csv", encoding="latin-1")

df.head()
```

Out[5]:

| | v1 | v2 | Unnamed: 2 | Unnamed: 3 | Unnamed: 4 |
|---|---|---|---|---|---|
| 0 | ham | Go until jurong point, crazy.. Available only ... | NaN | NaN | NaN |
| 1 | ham | Ok lar... Joking wif u oni... | NaN | NaN | NaN |
| 2 | spam | Free entry in 2 a wkly comp to win FA Cup fina... | NaN | NaN | NaN |
| 3 | ham | U dun say so early hor... U c already then say... | NaN | NaN | NaN |
| 4 | ham | Nah I don't think he goes to usf, he lives aro... | NaN | NaN | NaN |

In [6]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5572 entries, 0 to 5571
Data columns (total 5 columns):
 #   Column      Non-Null Count  Dtype
---  ------      --------------  -----
 0   v1          5572 non-null   object
 1   v2          5572 non-null   object
 2   Unnamed: 2  50 non-null     object
 3   Unnamed: 3  12 non-null     object
 4   Unnamed: 4  6 non-null      object
dtypes: object(5)
memory usage: 217.8+ KB
```

---

## 5. Methodology

The project followed the steps outlined below:

### 5.1 Data Loading

The dataset was loaded into Google Colab using the Pandas library.

### 5.2 Data Cleaning

- Removed unnecessary columns
- Normalized label values
- Handled missing or inconsistent data

### 5.3 Feature Extraction

Text data was converted into numerical features using **TF-IDF (Term Frequency–Inverse Document Frequency)** vectorization.

### 5.4 Model Training

A **Naive Bayes classifier** was trained on the transformed text data due to its efficiency and strong performance in text classification tasks.

**5.5 Model Evaluation**

The model was evaluated using:

- Accuracy
- Confusion Matrix
- Precision, Recall, and F1-score

---

**6. Machine Learning Model Used**

**Naive Bayes Classifier**

Reason for selection:

- Efficient for text classification
- Performs well with high-dimensional sparse data
- Widely used in spam detection systems

---

**7. Results and Performance**

- **Accuracy Achieved:** ~97%
- The model successfully classified spam and legitimate messages
- Low false-positive rate for legitimate content

---

Fig 2. ***Accuracy Output Screenshot Here***

```
In [36]:
from sklearn.metrics import accuracy_score

accuracy = accuracy_score(y_test, y_pred)
print(f"Model Accuracy: {accuracy * 100:.2f}%")
```

```
Model Accuracy: 97.40%
```

---

Fig 3. ***Classification Report Screenshot Here***

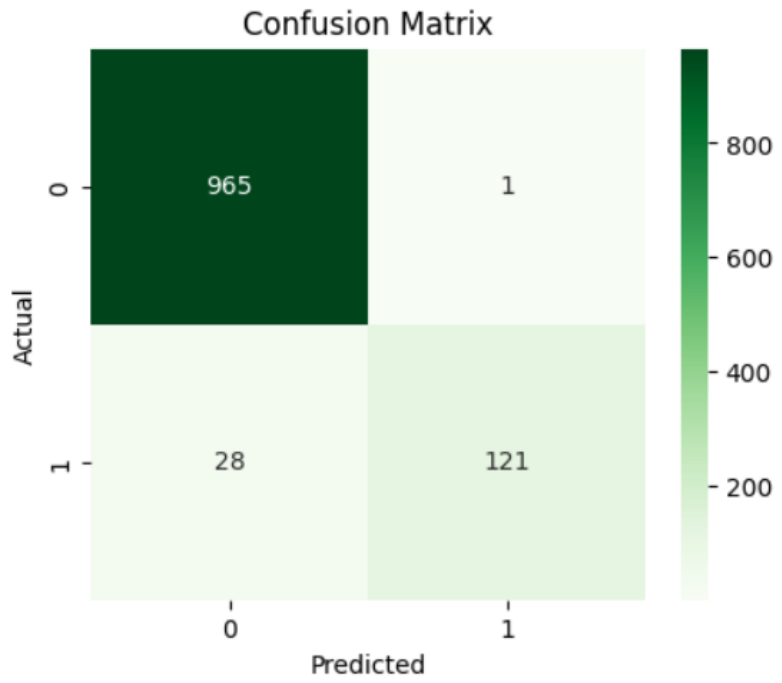
```python
from sklearn.metrics import classification_report

print("Classification Report:\n")
print(classification_report(y_test, y_pred))
```

```
Classification Report:

              precision    recall  f1-score   support

           0       0.97      1.00      0.99       966
           1       0.99      0.81      0.89       149

    accuracy                           0.97      1115
   macro avg       0.98      0.91      0.94      1115
weighted avg       0.97      0.97      0.97      1115
```

---

Fig 4. ***Confusion Matrix Screenshot Here***

```
cm = confusion_matrix(y_test, y_pred)

plt.figure(figsize=(5,4))
sns.heatmap(cm, annot=True, fmt='d', cmap='Greens')
plt.xlabel("Predicted")
plt.ylabel("Actual")
plt.title("Confusion Matrix")
plt.show()
```




**8. Real-World Testing**

To validate real-world applicability, the model was tested using actual cybersecurity-related content (e.g., Security+ discussion posts). The system correctly classified educational and informational content as legitimate, demonstrating logical decision-making rather than keyword-based filtering.

---

Fig 5. ***Real-World Input & Prediction Screenshot Here***

In [41]:

```
test_message = [
    """Test Your Security+ Knowledge: Day 27
    Question: A company wants to ensure that employees can access internal resources
    while working remotely, but also wants to enforce strong authentication and limit
    access based on device health and user role. Which of the following solutions BEST
    meets this requirement?"""
]
```

In [42]:

```
test_message_vec = vectorizer.transform(test_message)
```

In [43]:

```
prediction = model.predict(test_message_vec)
```

In [44]:

```
if prediction[0] == 1:
    print("⚠️ Spam / Phishing Content Detected")
else:
    print("✅ Legitimate / Informational Content")
```

```
✅ Legitimate / Informational Content
```

In [ ]:

---

### 9. Future Scope

The current project serves as a prototype. In the future, this system can be enhanced by:

- Developing a full-fledged application
- Integrating the model with **live Gmail or email gateways**
- Enabling **real-time spam and phishing detection**
- Applying deep learning models for improved contextual understanding

---

### 10. Conclusion

This project successfully demonstrates the application of **Machine Learning and NLP in cybersecurity**, specifically for spam and phishing detection. The model achieved high accuracy and performed effectively on real-world inputs.

The project strengthened practical understanding of how AI-driven systems can enhance email security and reduce risks associated with social engineering attacks.

---

**11. Tools & Technologies Used**

- Python
- Google Colab
- Pandas, NumPy
- Scikit-learn
- Matplotlib, Seaborn
- Natural Language Processing (TF-IDF)

---

**12. Declaration**

I Achyut Kumar Pandey, hereby declare that this project was completed by me as part of my internship and is an original work developed for learning and academic purposes.

Github repo link and Entire project Github repo link