

Abstract

This project demonstrates the development of a spam email classifier using Logistic Regression, a machine learning algorithm for binary classification problems. The system takes email content as input and classifies it as either 'Spam' or 'Ham (Not Spam)'. A Bag-of-Words model via CountVectorizer is used to convert text data into numerical features suitable for model training. The program features a user-friendly GUI built with Tkinter, allowing users to input emails, classify them with a single click, and view the result instantly. This project aims to provide a simple yet effective demonstration of text classification techniques and the practical application of machine learning in everyday digital communication.

Introduction

Email communication is widely used in both personal and professional contexts. However, spam emails pose a significant threat by wasting time, spreading malware, and promoting fraudulent activities. Detecting spam efficiently is crucial for enhancing cybersecurity and improving user experience. Logistic Regression is a widely used algorithm for binary classification tasks and is suitable for this purpose due to its simplicity, efficiency, and interpretability.

Dataset

A sample dataset of emails is created for demonstration purposes:

Email Content	Label
Win money now	1
Get free coupons	1
Hello friend how are you	0
Congratulations, you won a lottery	1
Let's catch up for lunch	0
Earn \$\$\$ from home	1
Are we still meeting tomorrow?	0
Buy cheap meds online	1
See you at the office	0

1 = Spam, 0 = Ham (Not Spam)

Methodology

1. **Text Preprocessing:** Emails are converted into numerical feature vectors using CountVectorizer (Bag-of-Words model).

2. **Model Training:** Logistic Regression is trained on the sample dataset to learn patterns of spam and ham emails.
3. **Prediction Function:** The GUI includes a function `classify_email()` that takes user input, vectorizes it, predicts spam/ham, and displays the result.
4. **GUI Implementation:** Tkinter library is used to create a user-friendly interface with a text box for email input, a 'Classify' button, and a result label.

Implementation

```
import tkinter as tk
from tkinter import scrolledtext, messagebox
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.linear_model import LogisticRegression

# Sample Dataset
emails = [
    "Win money now", "Get free coupons", "Hello friend how are you",
    "Congratulations, you won a lottery", "Let's catch up for lunch",
    "Earn $$$ from home", "Are we still meeting tomorrow?",
    "Buy cheap meds online", "See you at the office"
]
labels = [1, 1, 0, 1, 0, 1, 0, 1, 0]

# Train Model
vectorizer = CountVectorizer()
X = vectorizer.fit_transform(emails)
model = LogisticRegression()
model.fit(X, labels)

# Prediction Function
def classify_email():
    text = text_box.get("1.0", tk.END).strip()
    if not text:
        messagebox.showwarning("Input Error", "Please enter email content")
        return
    X_new = vectorizer.transform([text])
    prediction = model.predict(X_new)[0]
    result = "🔴 Spam" if prediction == 1 else "🟢 Ham (Not Spam)"
    result_label.config(text=f"Result: {result}", fg="red" if prediction == 1
                        else "green")

# GUI Setup
root = tk.Tk()
root.title("Spam Email Classifier - Logistic Regression")
root.geometry("500x400")

tk.Label(root, text="Enter Email Content:", font=("Arial", 12)).pack(pady=5)
text_box = scrolledtext.ScrolledText(root, wrap=tk.WORD, width=50, height=10,
font=("Arial", 10))
text_box.pack(pady=5)
```

```
classify_btn = tk.Button(root, text="Classify", font=("Arial", 12, "bold"),
command=classify_email)
classify_btn.pack(pady=10)
result_label = tk.Label(root, text="Result: ", font=("Arial", 14, "bold"))
result_label.pack(pady=10)
root.mainloop()
```

Conclusion

This project provides a practical example of email spam detection using Logistic Regression. It demonstrates how text data can be transformed into numerical features and classified efficiently using machine learning. The Tkinter GUI allows end-users to interact with the model easily, making it suitable as an educational tool or prototype for more advanced spam detection systems.