Tecnológico Nacional de México / Instituto Tecnológico de Culiacán.

Ingeniería en Sistemas Computacionales.

Inteligencia Artificial

Zuriel Dathan Mora Felix.

Modulo 2

Detección de Spam

Ontiveros Sánchez Jesús Daniel (22170750).

Ramírez Ruiz Santiago (22170783).

Horario de 9:00 a 10:00 AM.

Culiacán Sinaloa a 12 de octubre del 2025

Descripción General del código de Detección de Spam

El presente proyecto implementa un sistema automatizado de detección de correos electrónicos no deseados (spam) utilizando técnicas de procesamiento de lenguaje natural (NLP) y aprendizaje automático (Machine Learning).

El código fue desarrollado en Python, haciendo uso de bibliotecas como pandas, numpy, nltk y scikit-learn, con el objetivo de analizar el contenido textual de los correos y determinar si corresponden a mensajes legítimos (HAM) o potencialmente maliciosos (SPAM).

El sistema realiza un proceso completo que incluye:

- Lectura y limpieza de los datos provenientes de un conjunto de correos electrónicos.
- Normalización del texto eliminando caracteres especiales, acentos y palabras vacías.
- Conversión del texto en vectores numéricos mediante la técnica TF-IDF (Term Frequency – Inverse Document Frequency).
- Entrenamiento de un modelo de clasificación (Naive Bayes) para predecir la categoría del mensaje.
- Aplicación de reglas adicionales para reforzar la detección de correos sospechosos en función de su remitente, asunto, enlaces y adjuntos.

Este código permite identificar de manera eficiente mensajes potencialmente peligrosos, como intentos de phishing, propaganda no deseada o archivos adjuntos con riesgo de malware, proporcionando así una herramienta práctica para fortalecer la seguridad del correo electrónico.

Descripción general del conjunto de datos

El conjunto de datos utilizado para entrenar y evaluar el modelo de detección de spam estuvo compuesto por correos electrónicos etiquetados en dos categorías principales:

- **HAM (no deseados)**: correos legítimos o seguros.
- **SPAM (no deseados)**: mensajes con contenido sospechoso o malicioso.

Cada registro contenía información relevante del correo, como:

- **Remitente**: dirección o nombre del emisor del mensaje.
- **Asunto**: línea de título del correo.
- **Cuerpo del mensaje**: texto completo del contenido.
- **Enlaces**: URLs o direcciones externas incluidas.
- **Adjuntos**: nombres y extensiones de los archivos anexados.

En caso de no tener un archivo CSV con datos, el programa genera automáticamente un pequeño conjunto de ejemplos para realizar las pruebas.

Antes del análisis, los textos fueron normalizados aplicando técnicas de limpieza:

- Conversión a minúsculas.
- Eliminación de signos de puntuación, números y caracteres especiales.
- Eliminación de stopwords en español.
- Reducción de palabras mediante lematización.

El modelo utilizó un vectorizador TF-IDF para convertir el texto en vectores numéricos y un clasificador Naive Bayes Multinomial para determinar la probabilidad de que un correo sea spam.

Reglas creadas para la detección

El código no solo se apoya en el modelo de aprendizaje, sino también en un conjunto de reglas personalizadas que ayudan a mejorar la precisión del sistema. Estas reglas permiten detectar comportamientos o patrones típicos de correos peligrosos.

Nº	Regla	Descripción
1	**Limpieza del texto**	Se eliminan acentos, mayúsculas, signos y caracteres especiales para un análisis uniforme.
2	**Dominio del remitente**	Se analiza la parte del correo después del "@" (por ejemplo, gmail.com, empresa.org) para detectar dominios sospechosos o poco comunes.
3	**Detección de enlaces**	El sistema busca direcciones web (http:// o https://) y genera tokens como has_url, url_dom_, url_tld_ o url_sub_ para identificar su tipo.
4	**Análisis de archivos adjuntos**	Se revisan los nombres y extensiones de los archivos incluidos en el correo. Se crean etiquetas como att_ext_pdf, att_ext_cmd, att_ext_dangerous, etc.
5	**Extensiones peligrosas**	Si el correo contiene archivos con extensiones como .exe, .bat, .cmd, .js, .vbs, .scr, .jar o .apk, el sistema las marca como de alto riesgo.
6	**Presencia de adjuntos o enlaces**	Si el mensaje tiene adjuntos o enlaces, se agregan los indicadores has_attachment y has_url para reflejarlo.
7	**Refuerzo de peligro**	Si se detecta una extensión peligrosa, la regla aumenta su peso varias veces para que el modelo le dé mayor importancia al clasificar.
8	**Palabras clave en nombres de archivos**	El sistema extrae palabras del nombre de los adjuntos (como "factura", "cuenta", "verificar") para reforzar patrones comunes de phishing.

Estas reglas funcionan como filtros adicionales que enriquecen la información antes de enviarla al modelo matemático.

De esta forma, el sistema combina inteligencia estadística (aprendida del dataset) con detección lógica (definida por las reglas).

Resultados del sistema

Durante las pruebas realizadas con 10 correos electrónicos reales (tanto legítimos como no deseados), el sistema de detección mostró un desempeño sobresaliente.

El modelo clasificó correctamente los 10 correos, alcanzando una precisión global del 100 %, lo que significa que no se produjeron errores de clasificación.

Los resultados se resumen de la siguiente manera:

- Total de correos analizados: 10
- Correos SPAM reales: 6
- Correos HAM (legítimos): 4
- Aciertos totales: 10
- Precisión general (accuracy): 100 %
- Tasa de detección de spam (recall): 100 %
- Falsos positivos: 0 %
- Falsos negativos: 0 %

En cuanto a las probabilidades calculadas, se observó que:

- Los correos legítimos (HAM) tuvieron valores bajos (entre 0.05 y 0.58), indicando poca probabilidad de ser spam.
- Los correos no deseados (SPAM) presentaron valores altos (entre 54 % y 98 %), lo que refleja una detección confiable.

Conclusiones y posibles mejoras

El clasificador demostró ser efectivo al combinar técnicas automáticas de aprendizaje con reglas hechas a mano.

Gracias a esta mezcla, se logró un equilibrio entre precisión y flexibilidad, adaptándose tanto a correos reales como a casos simulados.

No obstante, hay varias maneras en las que el sistema podría mejorar:

- Agregar más palabras clave sospechosas
- Filtrar dominios falsos o poco comunes
- Agregar más ejemplos al dataset
- Detectar correos con emojis o símbolos extraños
- Darle prioridad a algúno de los campos ya sea adjuntos, enlaces, etc.

En general, el proyecto demuestra que es posible detectar correos spam de manera automática y confiable combinando reglas lógicas con aprendizaje automático, lo que ofrece una base sólida para futuras versiones más completas.