Para desarrollar un modelo de detección de amenazas en correos electrónicos mediante machine learning en Python, primero es útil identificar cómo queremos detectar estas amenazas. Podemos enfocarnos, por ejemplo, en detectar correos electrónicos de phishing o aquellos que contengan malware. Un modelo común en este ámbito es el de clasificación supervisada, donde se entrena con ejemplos de correos seguros y correos maliciosos para que aprenda a distinguir entre ellos.

Aquí te explico los pasos generales para desarrollar un modelo de este tipo utilizando técnicas de procesamiento de lenguaje natural (NLP) y un modelo de clasificación supervisada (como Random Forest o Naive Bayes) en Python. Luego, adaptaremos la estrategia de detección para amenazas en correos electrónicos y explicaré el código paso a paso.

1. Preparación de datos y preprocesamiento

Necesitamos un conjunto de datos de correos electrónicos etiquetados como seguros o maliciosos. Si tienes tus propios datos, asegúrate de que cada correo esté etiquetado de manera adecuada.

Pasos de preprocesamiento

1. Limpieza de datos: Eliminación de espacios, caracteres especiales y links en el texto.
2. Tokenización: Separar el texto en palabras o "tokens".
3. Vectorización: Convertir el texto en una representación numérica. Aquí usamos la técnica de TF-IDF (Term Frequency - Inverse Document Frequency).

2. Desarrollo del modelo

Vamos a utilizar el modelo Naive Bayes para clasificación. Este es un modelo de aprendizaje supervisado eficaz y comúnmente utilizado en detección de spam y phishing.