



**TECNOLÓGICO
NACIONAL DE MÉXICO**



Inteligencia Artificial

Tarea:

Ejemplos prácticos de los enfoques

Alumno:

Rementeria Medina Jesus Hector

```
import tkinter as tk
from tkinter import messagebox
import pandas as pd
import numpy as np
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.naive_bayes import MultinomialNB
from sklearn.model_selection import train_test_split
```

Cargamos las librerías necesarias para el funcionamiento del programa, Tkinter para la interfaz, pandas numpy y Sklearn para el funcionamiento de la ia

```
#Carga del Dataset
df = pd.read_csv(filepath_or_buffer='spam.csv', encoding='latin1')[['v1', 'v2']]
#Convertir columna 1 a label y columna 2 a texto
df = df.rename(columns={'v1': 'label', 'v2': 'text'})
#Convertir NoSpam a 0, Spam a 1
df['label'] = df['label'].map({'ham': 0, 'spam': 1})

# Separacion de datos
X_train, X_test, y_train, y_test = train_test_split(*arrays: df['text'], df['label'], test_size=0.2, random_state=42)

#Convertir el texto a vectores numericos
vectorizer = TfidfVectorizer()
#transformar vocabulario a valores numericos
X_train_tfidf = vectorizer.fit_transform(X_train)

# Modelo de valles
modelo = MultinomialNB()
#Entrenamiento de los datos
modelo.fit(X_train_tfidf, y_train)
```

Cargamos nuestro dataset que se encuentra en formato .csv, y convertimos las columnas de la tabla a formato label y texto, se utiliza el formato label para convertir los valores de ham y spam en 0 Y 1 para representar cuando un mensaje es spam o no, el texto simplemente se trata de un correo ejemplo que se usara como entrenamiento para la IA

Se seleccionan el 20% de los datos y se utilizan para llevar a cabo pruebas, luego se vectorizan los extos para que la IA pueda aprender el vocabulario y asi transforme el texto en valores numéricos, Creamos un modelo multinomial naive bayes y se entrena el modelo con los datos convertidos

```

#Presicion esperada del modelo
print(f"Precisión del modelo: {modelo.score(vectorizer.transform(X_test), y_test):.2f}")

#Ventana principal
root = tk.Tk()
root.title("Detector de SPAM")
root.geometry("650x550")
|

#Enviar Correo
def enviarCorreo(): 1 usage
    Correo = entry_correo.get()
    Asunto = entry_asunto.get()
    Mensaje = text_mensaje.get(index1: "1.0", tk.END)
    #Combina los 3 campos en una sola variable
    TextoCombinado = f"{Correo} {Asunto} {Mensaje}"

    # Vectorizar el mensaje
    input_vector = vectorizer.transform([TextoCombinado])
    pred_prob = modelo.predict_proba(input_vector)[0]
    #probabilidad de NoSpam
    P_noSpam = pred_prob[0]
    #Probabilidad de Spam
    p_spam = pred_prob[1]

```

Se mide la Precisión del modelo utilizando los datos de prueba y se muestra en la consola, Creamos el método enviarCorreo, el cual recibe de la interfaz el texto escrito en los cambios Correo, Asunto y Mensaje, se combina la información de los 3 campos en una sola variable la cual es vectorizada y se obtiene cuáles son las probabilidades de que el correo sea Spam o no lo sea

```

# Mostrar Probabilidades
resultado_prob.config(text=f"Probabilidad de SPAM: {p_spam:.2f}\nProbabilidad de NO SPAM: {p_ham:.2f}")

#Calcular si es Spam
if p_spam > P_noSpam:
    messagebox.showwarning(title: "SPAM", message: "El correo fue marcado como SPAM")
else:
    messagebox.showinfo(title: "Éxito", message: "Correo enviado correctamente")

```

Se muestran ambas probabilidades en un campo de la interfaz y se calcula si la probabilidad de spam es mayor que la de que no sea spam, si este es el caso se crea una ventana emergente avisándole al usuario que el correo fue marcado como spam, de lo contrario la ventana emergente notificara al usuario que el correo fue mandado de forma correcta

```

#Campo Correo
tk.Label(root, text="Correo:").pack(pady=5)
entry_correo = tk.Entry(root, width=60)
entry_correo.pack(pady=5)

#Campo Asunto
tk.Label(root, text="Asunto:").pack(pady=5)
entry_asunto = tk.Entry(root, width=60)
entry_asunto.pack(pady=5)

#Campo mensaje
tk.Label(root, text="Mensaje:").pack(pady=5)
text_mensaje = tk.Text(root, width=70, height=10)
text_mensaje.pack(pady=5)

#Boton Enviar
tk.Button(root, text="Enviar Correo", command=enviarCorreo).pack(pady=20)

# Area probabilidades
resultado_prob = tk.Label(root, text="", font=("Arial", 12))
resultado_prob.pack(pady=10)

root.mainloop()

```

El resto del código es parte de la interfaz principal donde el usuario simula el mandar un correo teniendo 3 campos, uno para el correo, otro para escribir el asunto y un último campo para escribir el contenido del correo junto con un botón para simular mandar dicho correo

Ejemplos

El modelo fue entrenado con un dataset que contenía correos en ingles así que las pruebas se realizaron simulando correos en ingles

Detector de SPAM con Naive Bayes y TF-IDF

Tu correo:

lottery@mega.com


Asunto:

Claim your reward NOW

Mensaje:

Congratulations! You have been selected for a \$5000 gift card. Hurry up and claim your reward before it expires.

SPAM

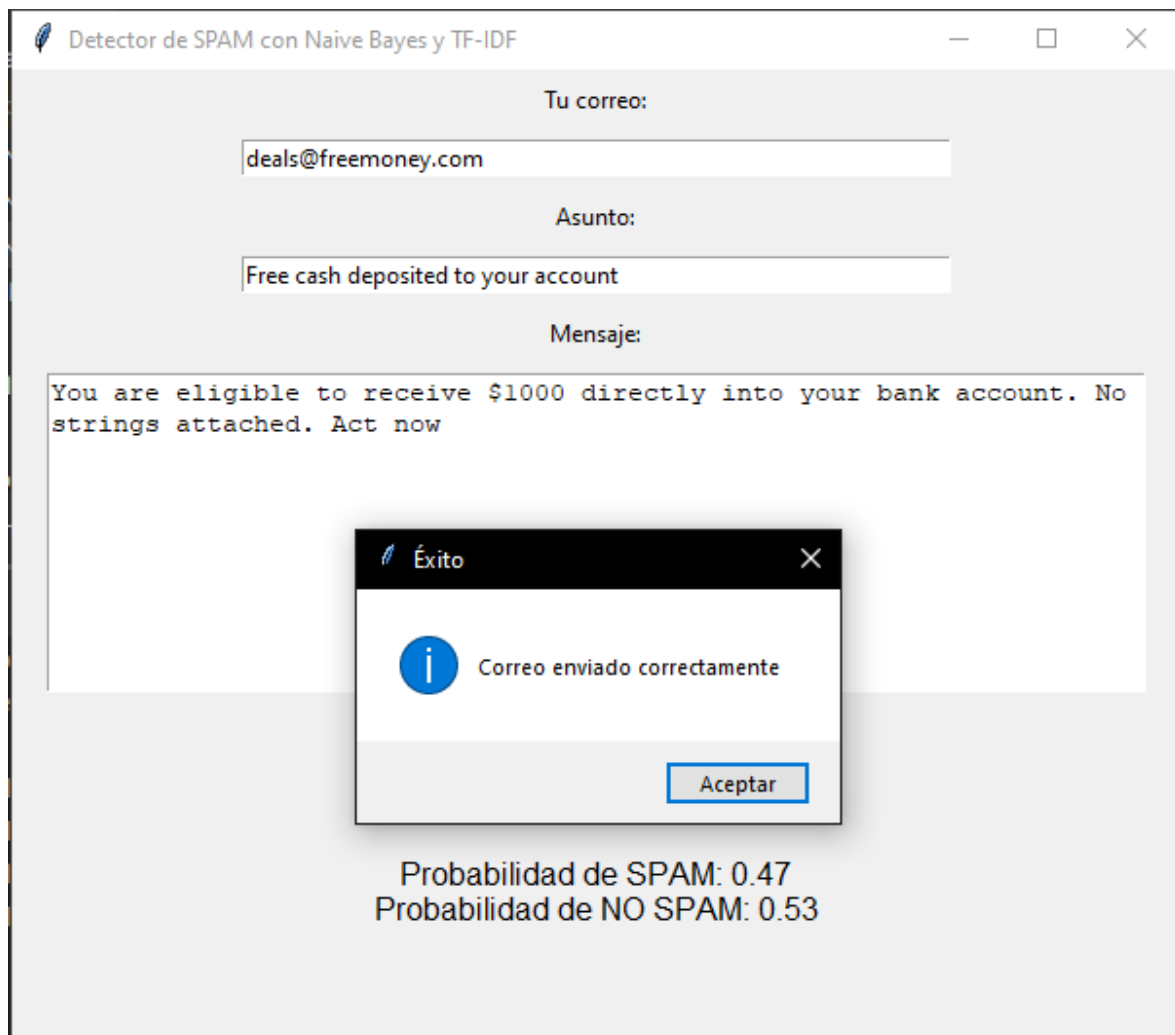
 El correo fue clasificado como SPAM

Aceptar


Probabilidad de SPAM: 0.73

Probabilidad de NO SPAM: 0.27

En este caso se marcó exitosamente el correo como spam con una probabilidad el 73% contra el 27%



En este ejemplo el contenido del correo si pertenece a un mensaje de spam, pero el modelo no logro conseguir el porcentaje de spam suficiente solamente marcando un 47% de probabilidad fallando en detectar el spam, mostrando que no siempre será capaz de detectar si algo es spam o no

 Detector de SPAM con Naive Bayes y TF-IDF

—

□

×

Tu correo:


john@example.com


Asunto:

Meeting Reminder

Mensaje:

Don't forget about our meeting tomorrow at 3 PM in the conference room.

 Éxito


 Correo enviado correctamente

Aceptar

Probabilidad de SPAM: 0.01

Probabilidad de NO SPAM: 0.99

En este ejemplo las probabilidades de Spam fueron mínimas al ser un mensaje que no contiene ninguna palabra o elemento que pueda hacer sospechar al modelo de Spam

 Detector de SPAM con Naive Bayes y TF-IDF

—

□

×

Tu correo:

team@company.com


Asunto:

Project Update

Mensaje:

Please find attached the latest project update. Let me know if you have any questions.

Éxito

 Correo enviado correctamente

Aceptar

Probabilidad de SPAM: 0.08

Probabilidad de NO SPAM: 0.92

En este igualmente las probabilidades son muy bajas