

|  |
| --- |
| MANUAL TÉCNICO  CLASIFICADOR DE CORREO SPAM |
| ANALISIS Y PROCESAMIENTO DE TEXTOS  UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO  FACULTAD DE INGENIERÍA |
| 27/MAYO/2020  ALUMNO:   * Orozco Magadán Brandon * Tavares Rosales Gerardo |

PROGRAMA CLASIFICADOR DE CORREOS SPAM

**Librerías, paquetes, módulos y kits de herramientas**

* Requerimos de una las siguientes librerías:
  + numpy 🡪 Librería principal para la informática científica.
  + pandas 🡪 Uso matricial para los dataframes.
  + nltk 🡪 Conjunto de bibliotecas y programas para el procesamiento del lenguaje natural y estadísticos.
  + re 🡪 Modulo para expresiones regulares
  + matplotlib 🡪 Biblioteca para la generación de gráficos.

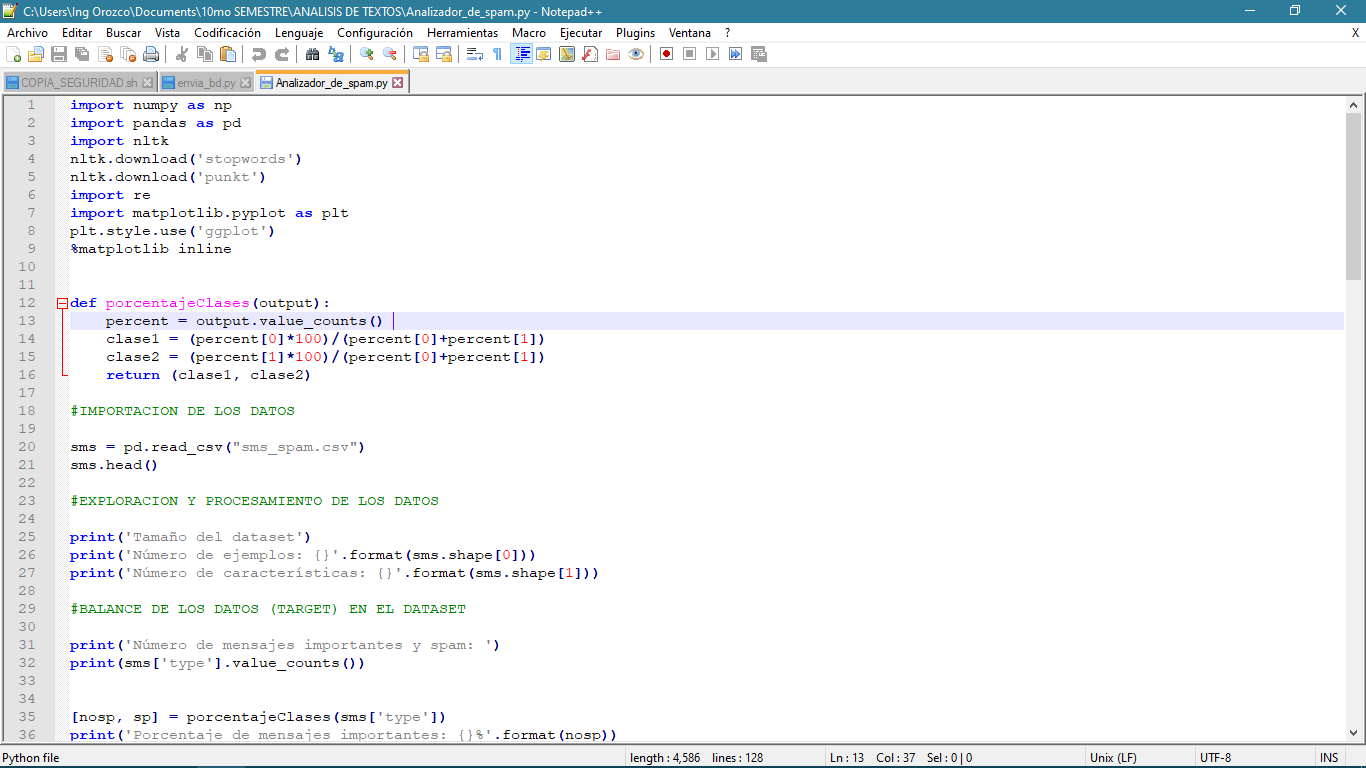


Imagen 1. Librerías, paquetes, módulos y kits de herramientas del clasificador de correos spam.

**Declaración y definición de las funciones del clasificador de correo spam**

* Función de obtención de porcentajes de las clases.

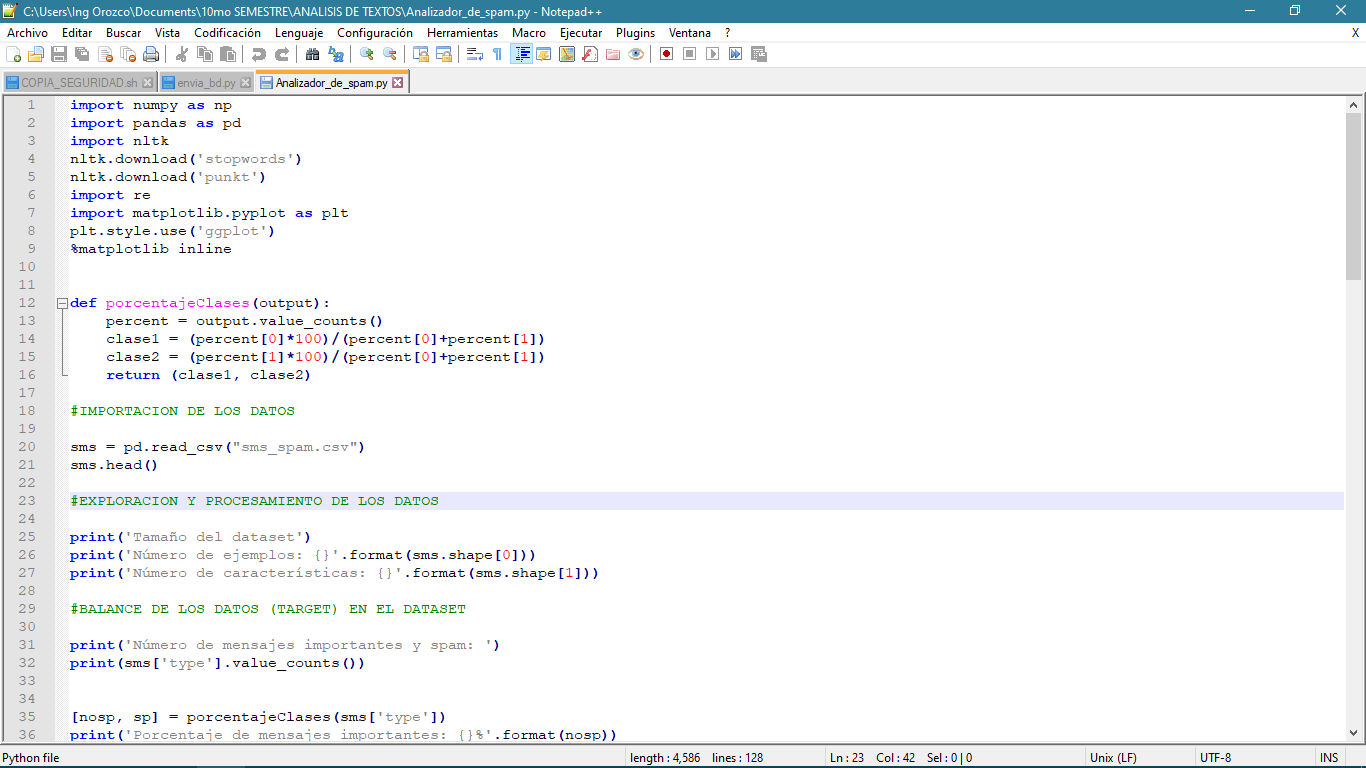


Imagen 2. Definición de la función porcentajeClases.

* Cargamos el Dataset de los datos de correos.

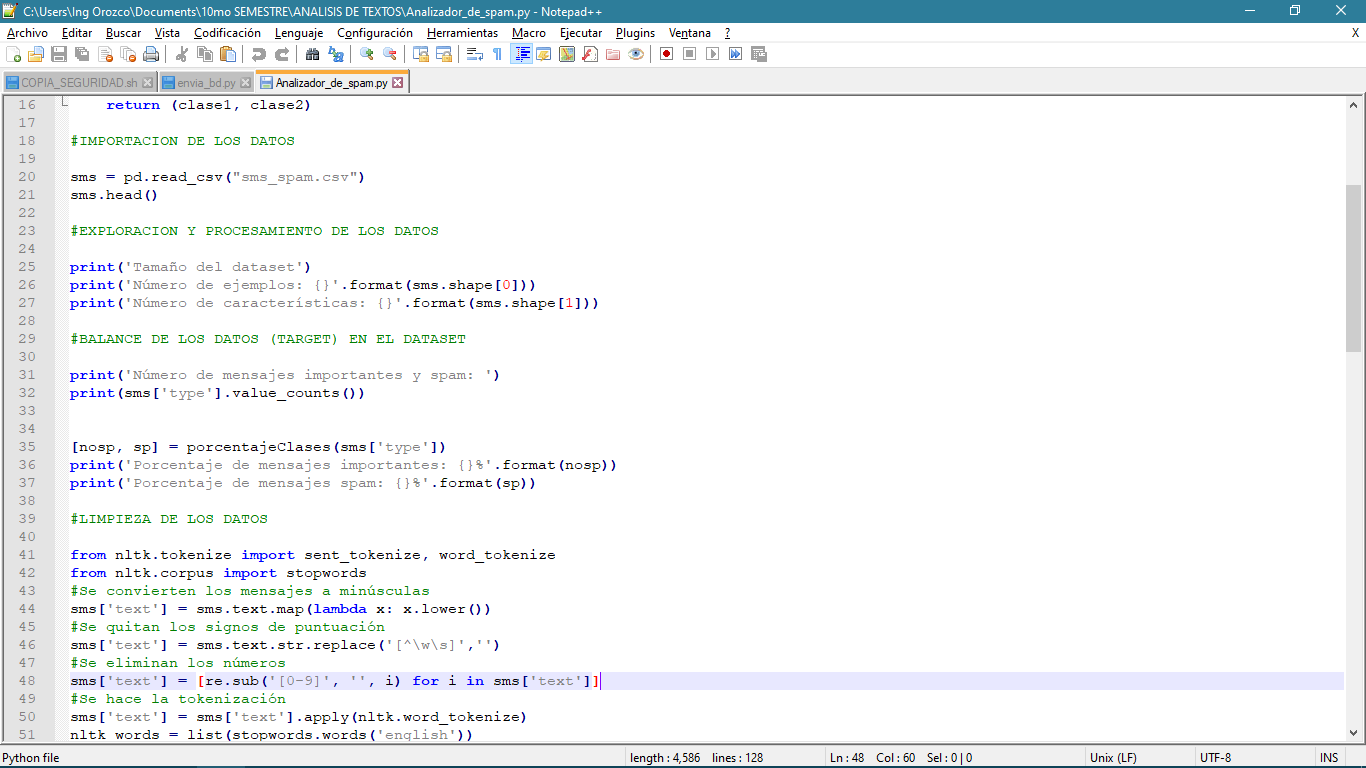


Imagen 3. Importación de los datos contenidos en el Dataset “sms\_spam.csv”.

* Imprimimos el número de registros y las características del Dataset para conocer los datos sobre los que se basa el programa para clasificar.

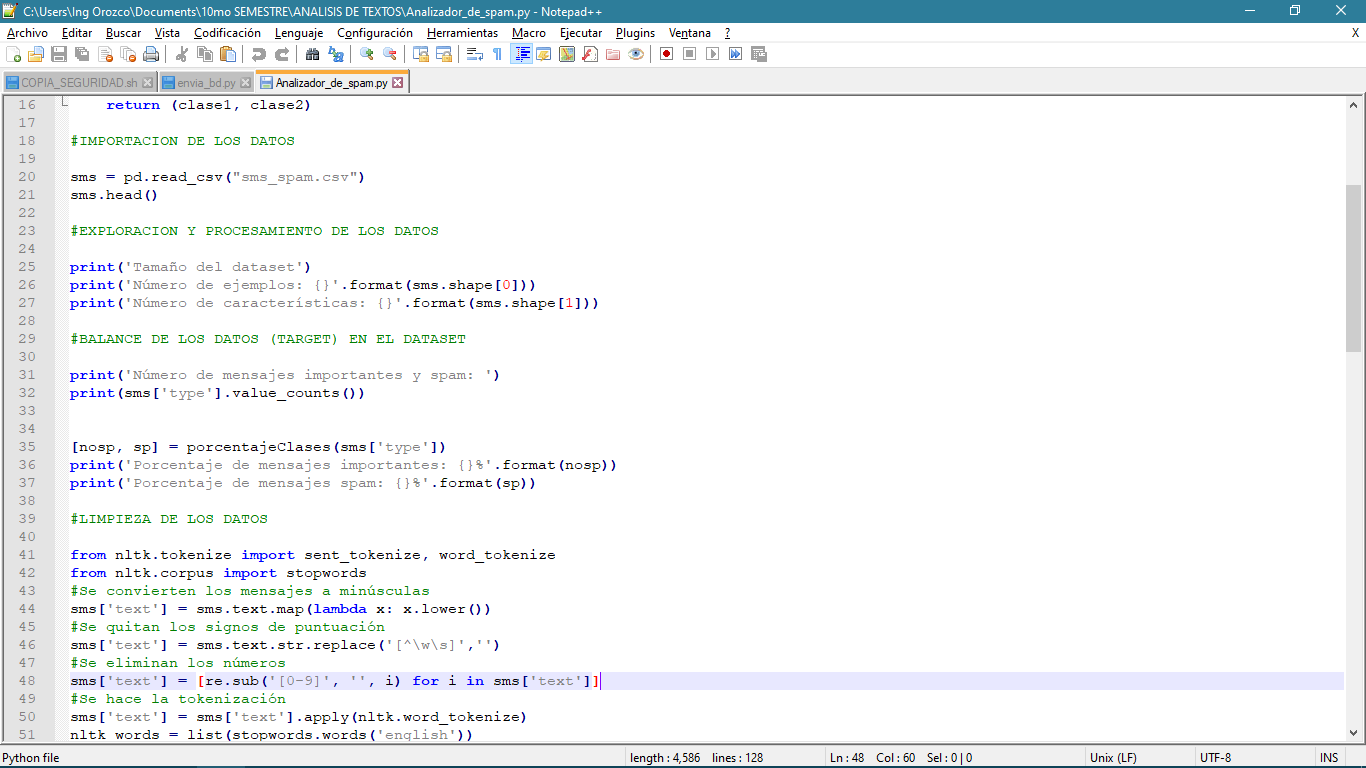


Imagen 4. Impresión de los datos y características del Dataset.

* Se genera un balance de los datos.

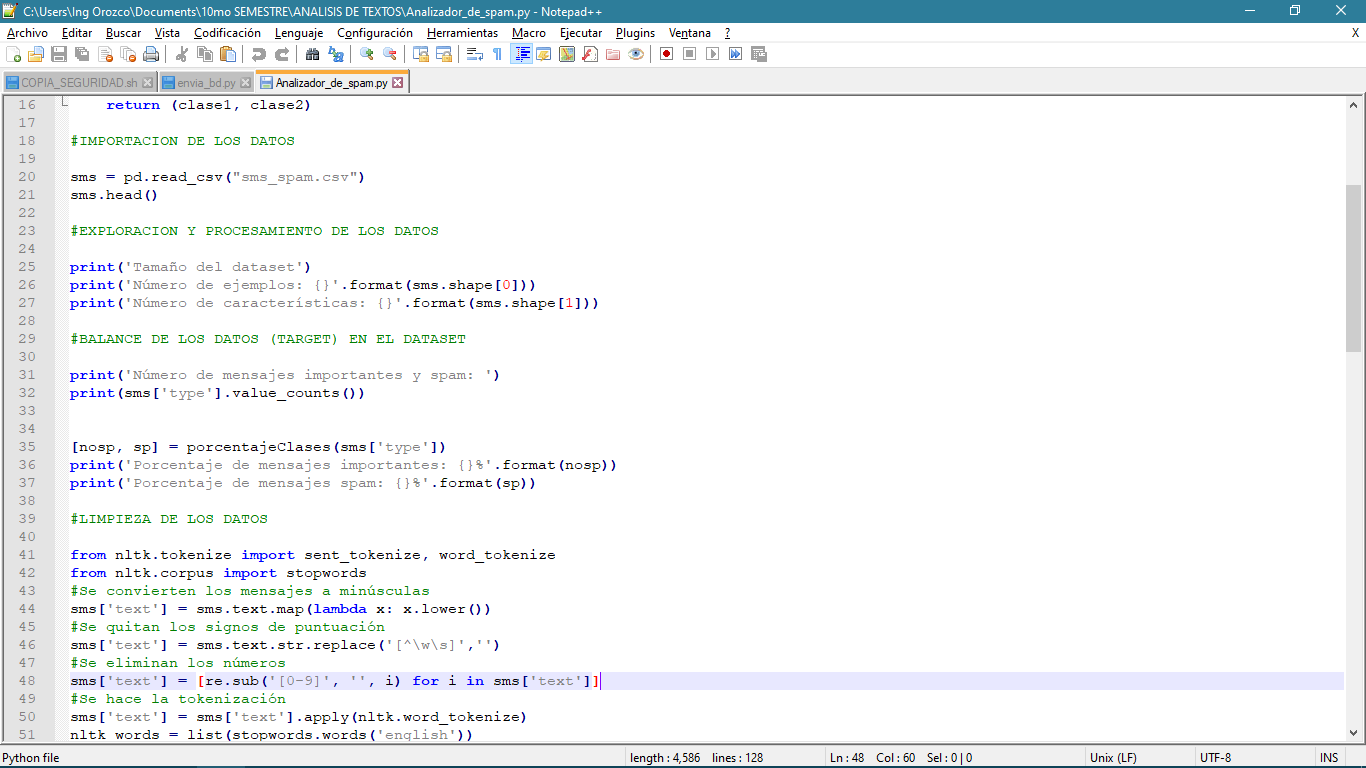


Imagen 5. Balance de los datos Target en el Dataset.

* Se genera una limpieza en distintos puntos precisos del Dataset para tratarlo de manera correcta y poder aplicarle los métodos de clasificación. Para ello, hacemos:

1. Convertimos los mensajes a minúsculas.
2. Se quitan los signos de puntuación.
3. Se eliminan los números.
4. Se hace la tokenización.
5. Generamos la lista de “*stopwords*” en inglés.
6. Obtenemos una lista de palabras en un *string* separado por espacios.

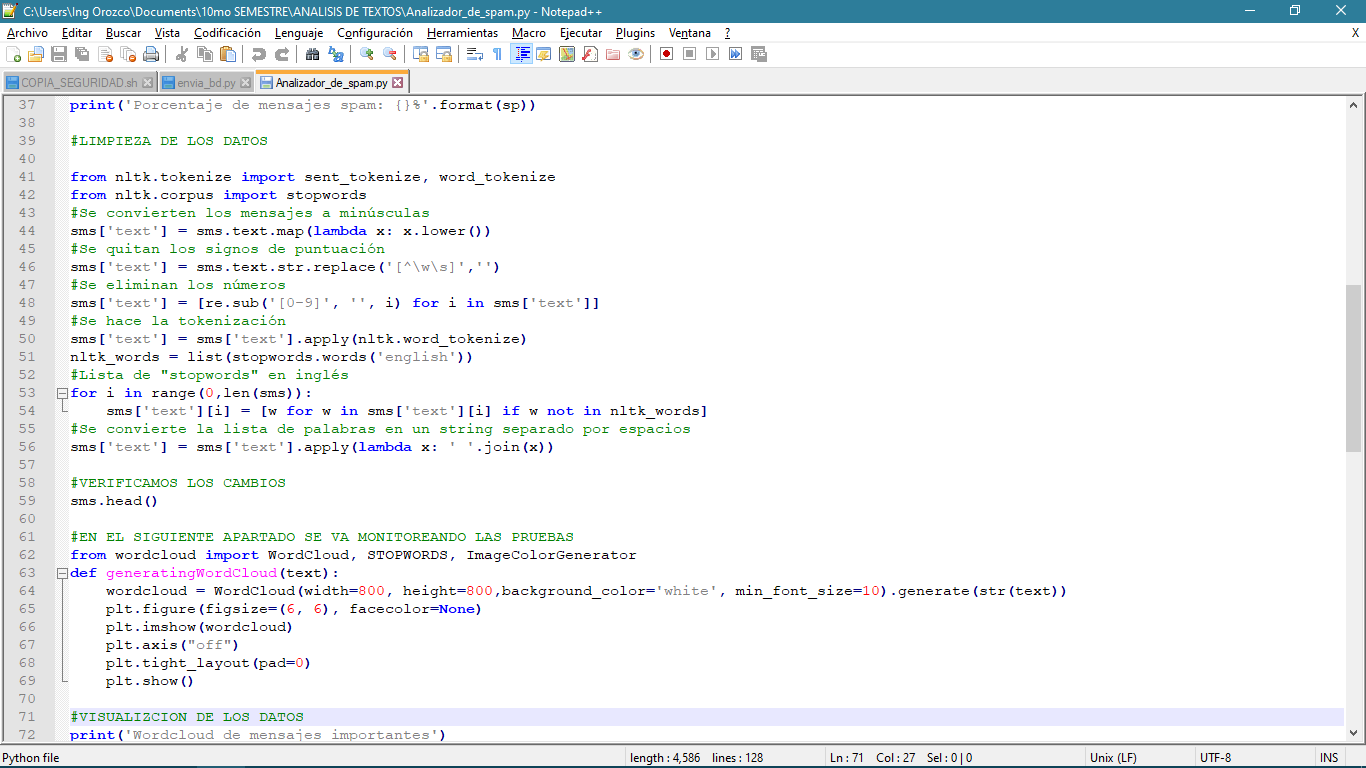


Imagen 6. Limpieza y trato de los datos.

* Se verifica con una impresión de los datos ya tratados.

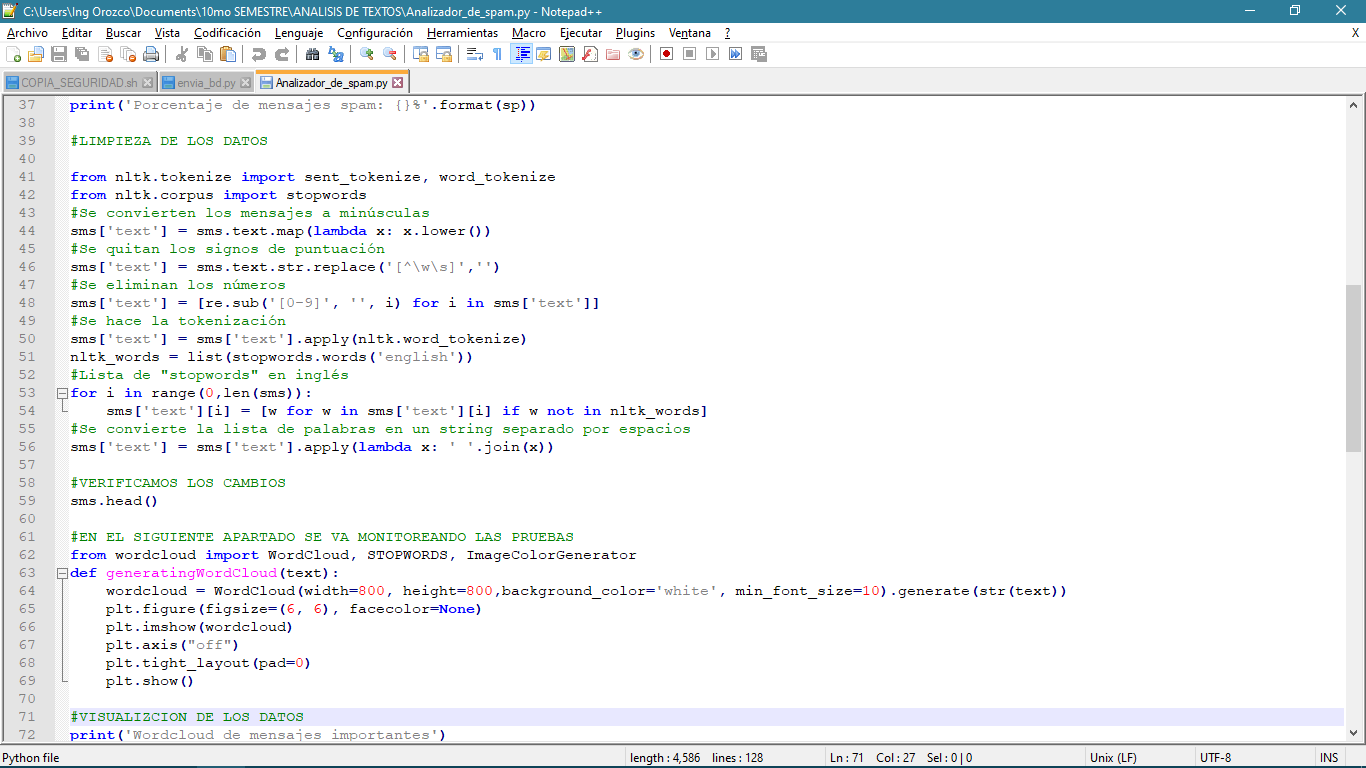


Imagen 7. Impresión de los datos.

* Se realiza una monitorización de los datos ploteando las gráficas de los datos.

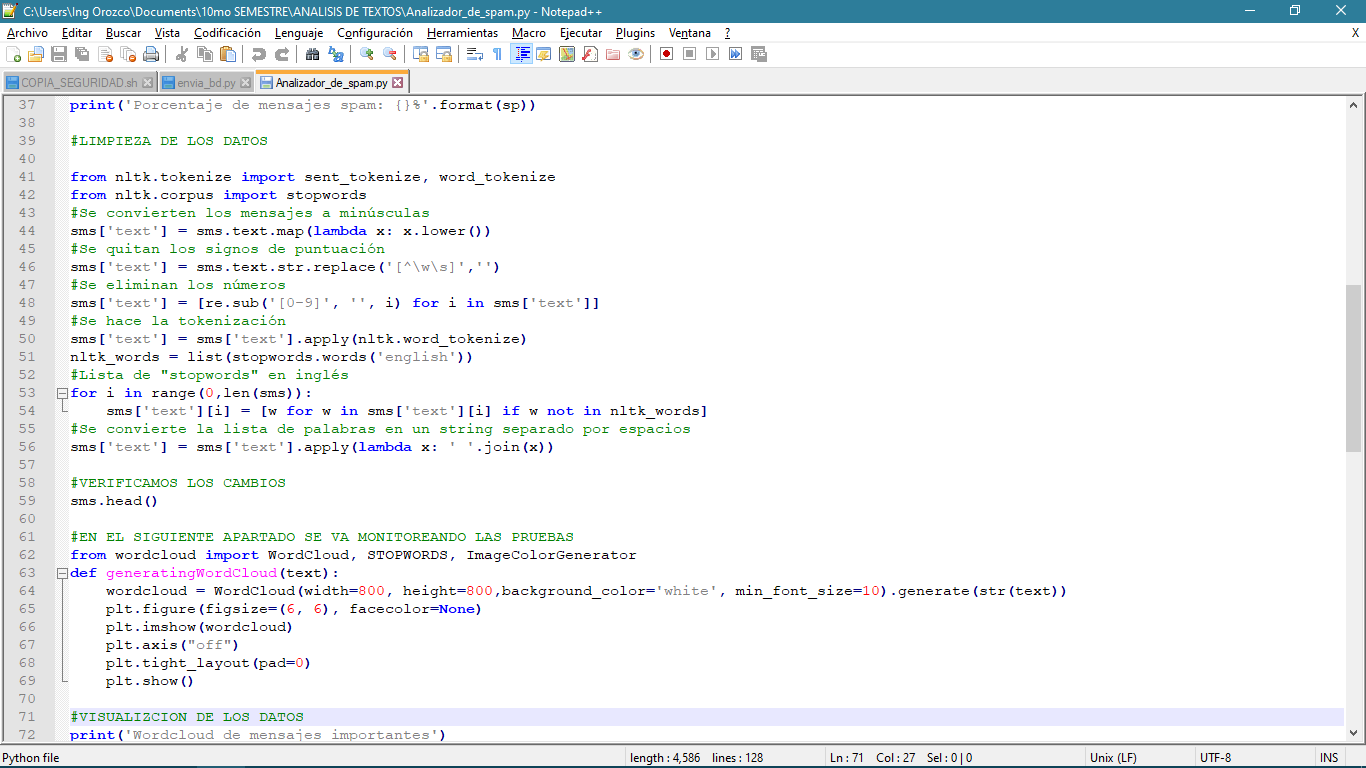


Imagen 8. Monitoreo de las pruebas.

* Visualizamos los datos “importantes” y los “spam”.

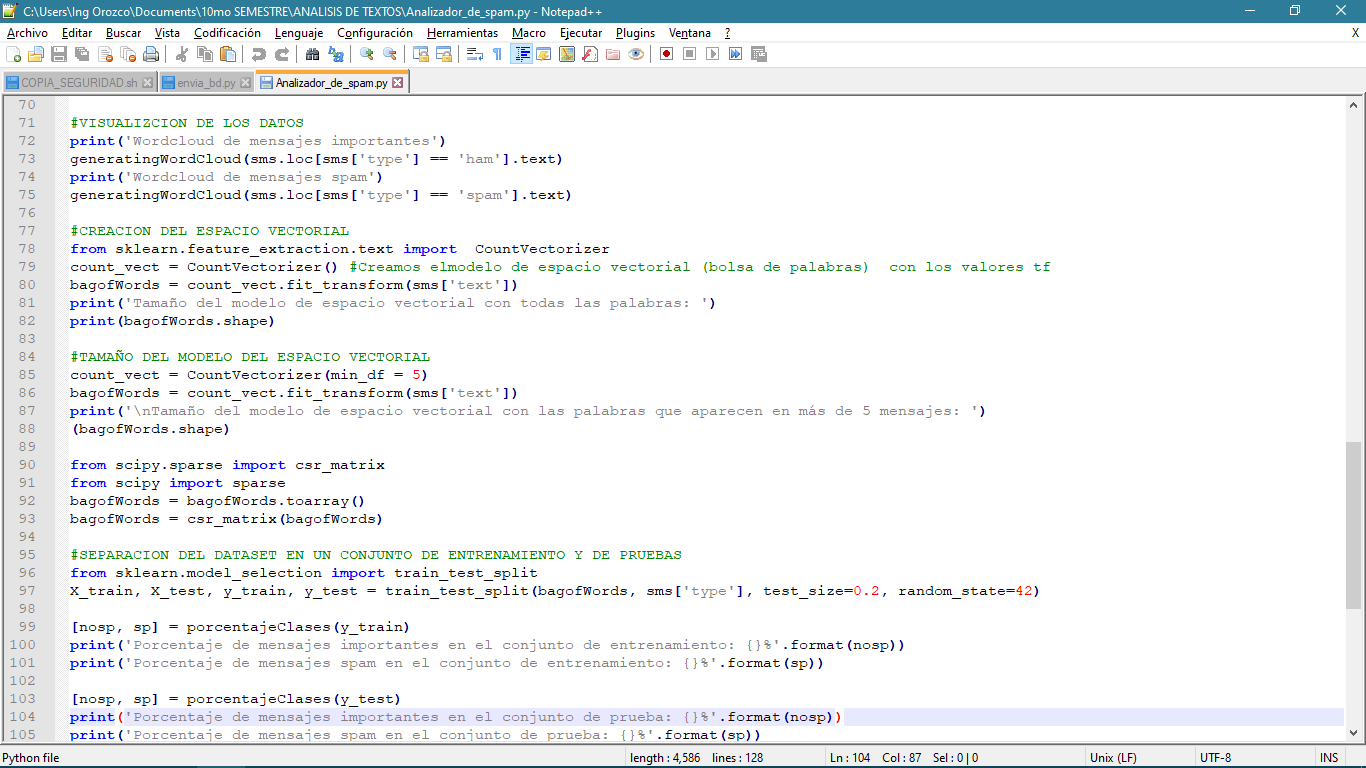


Imagen 9. Impresión de los distintos datos.

* Creación del espacio vectorial (bolsa de palabras) las que definirán cuanto usaremos del dataset.

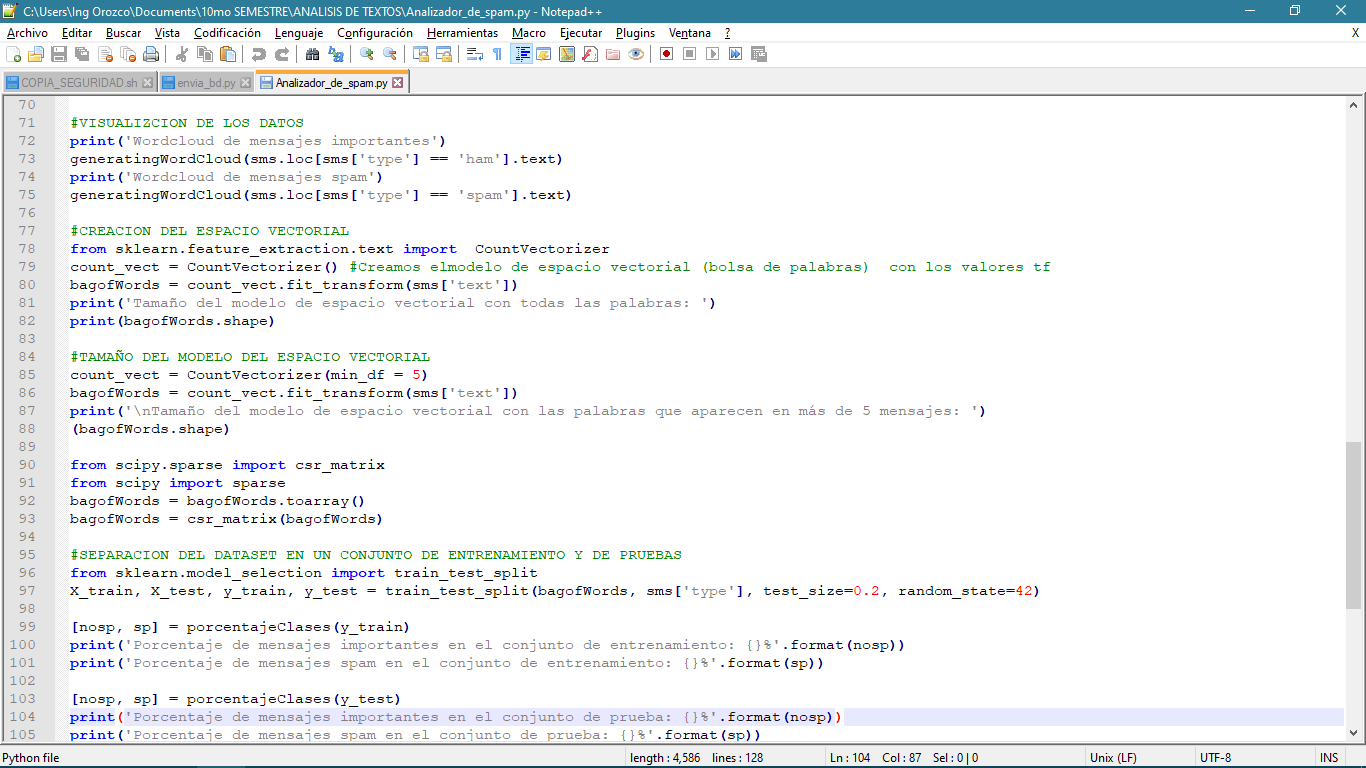


Imagen 10. Creación del espacio vectorial.

* Definimos el tamaño de palabras referido a cuantas veces aparecerán las palabras repetidas.

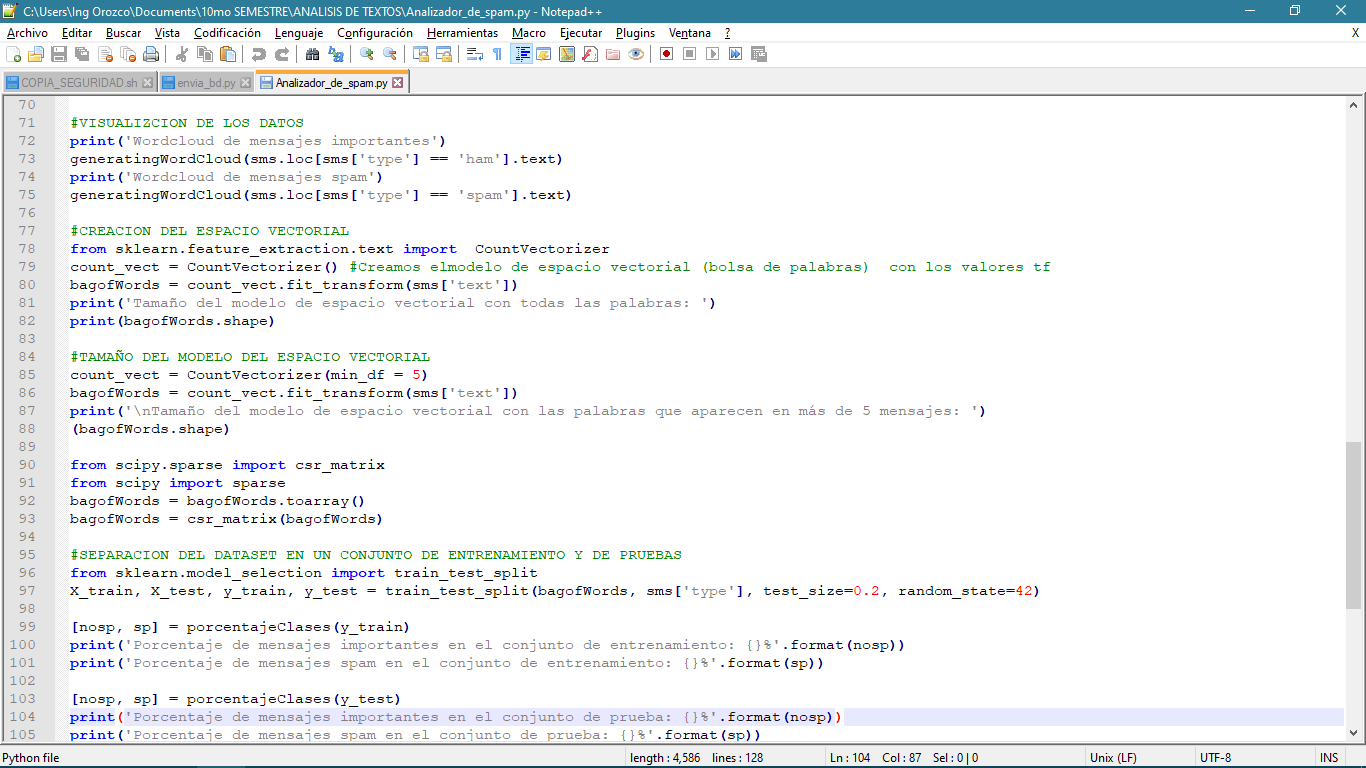


Imagen 11. Definimos el tamaño del espacio.

* Realización del set de entrenamiento y de prueba.

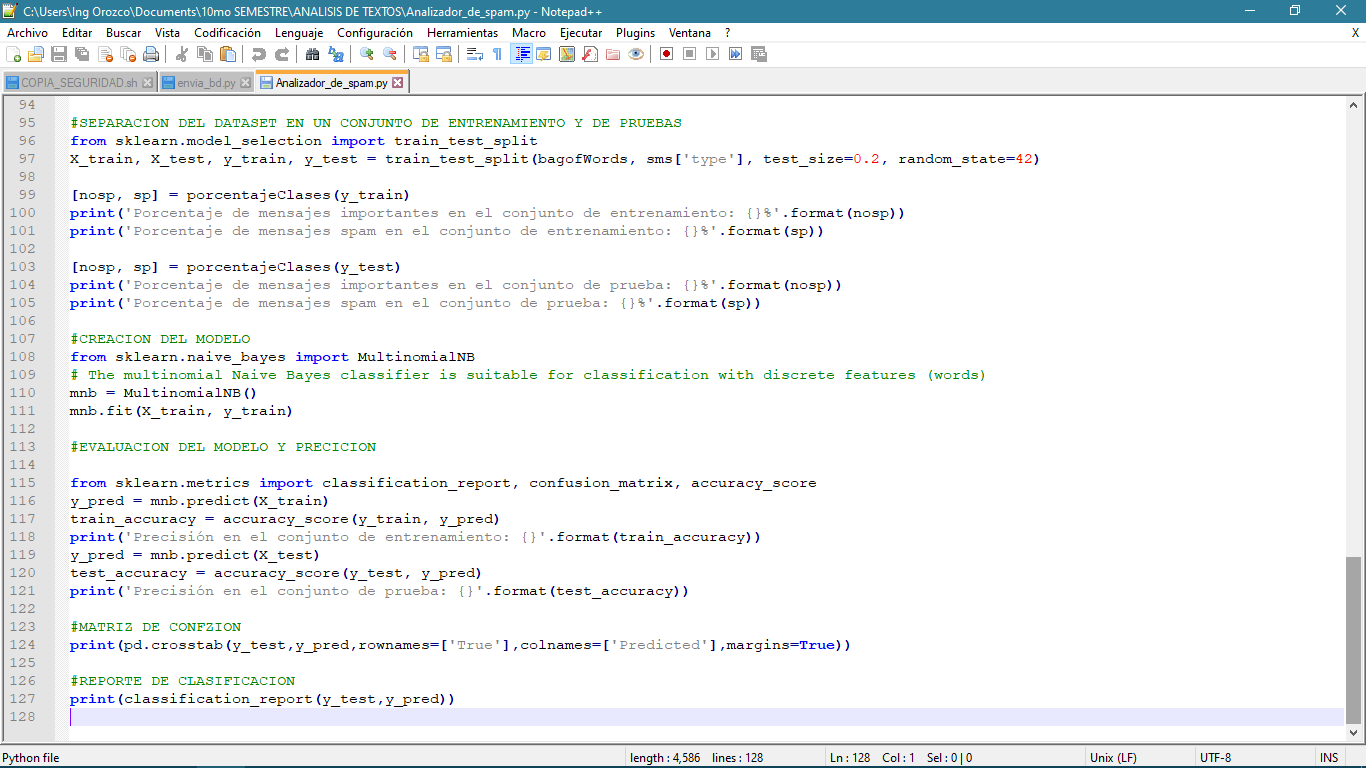


Imagen 12. Impresión del porcentaje de precisión para pruebas y entrenamiento.

* Realización del modelo de bayes.

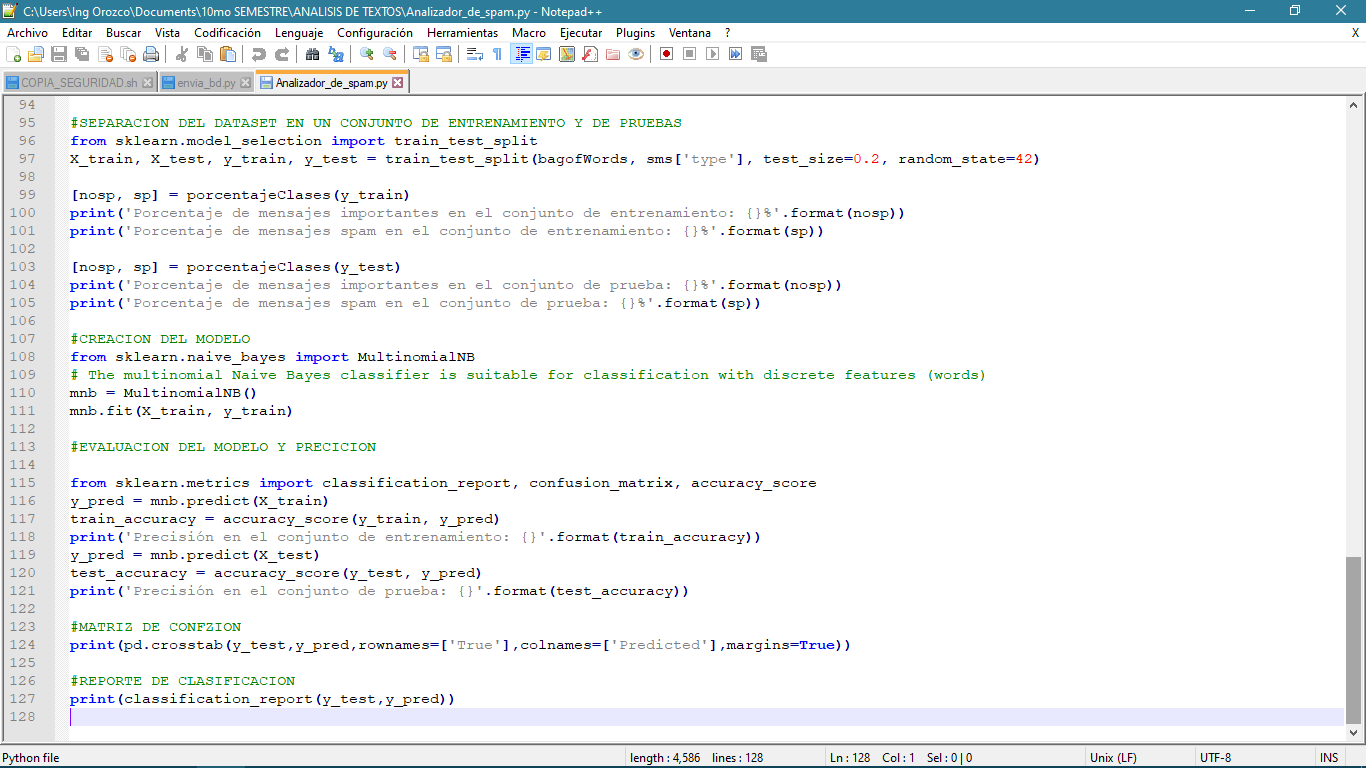


Imagen 13. Realizamos el modelo de Naive Bayes.

* Evaluación del modelo ajustando la precisión del mismo.

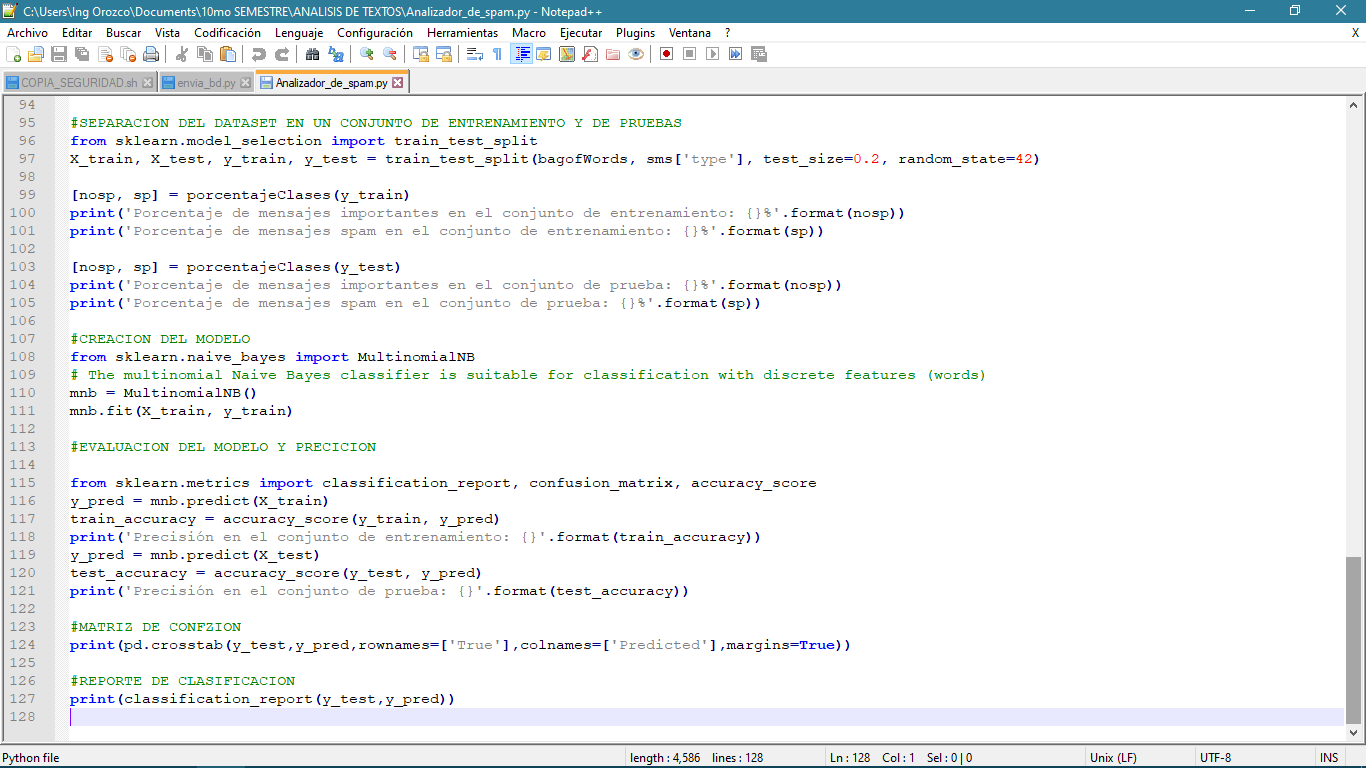


Imagen 14. Aplicación de la evaluación.

* Matriz de confusión y reporte final de la clasificación.

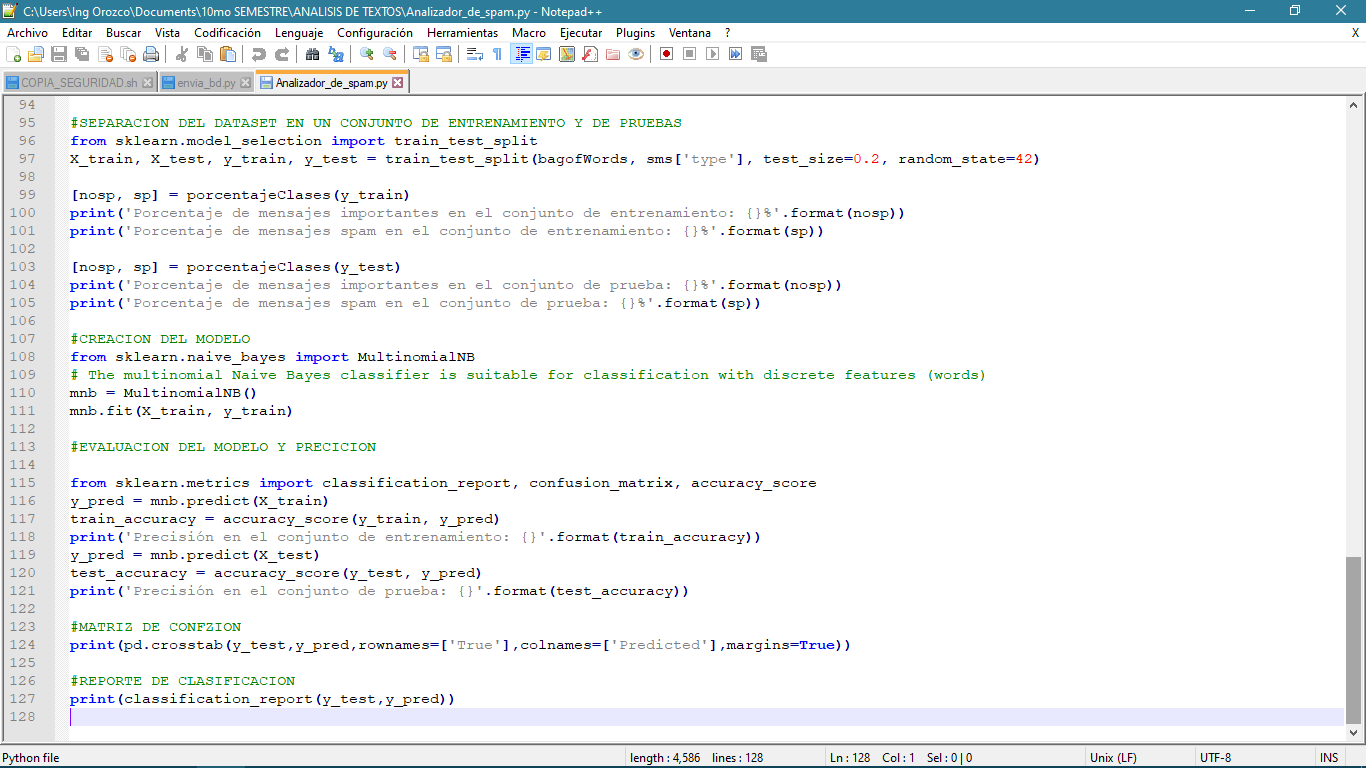
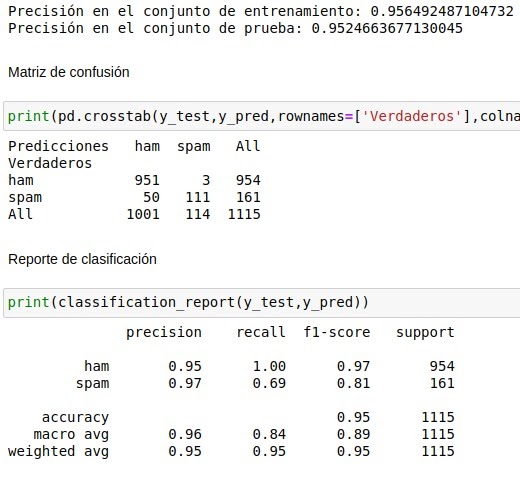


Imagen 15. Generación de la matriz y reporte de la clasificación de textos.

Pruebas

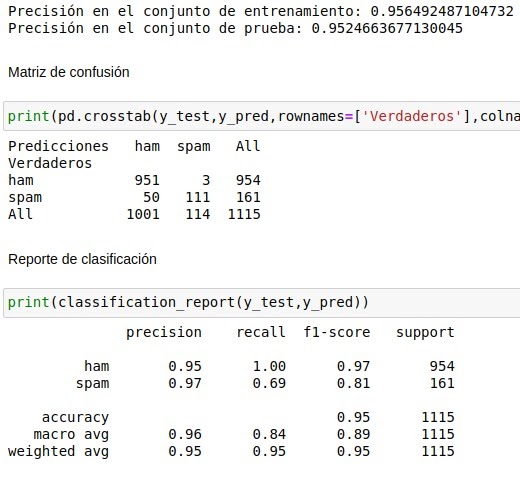
## **Prueba uno**

* Porcentaje de prueba de 0.2
* min\_df = 5



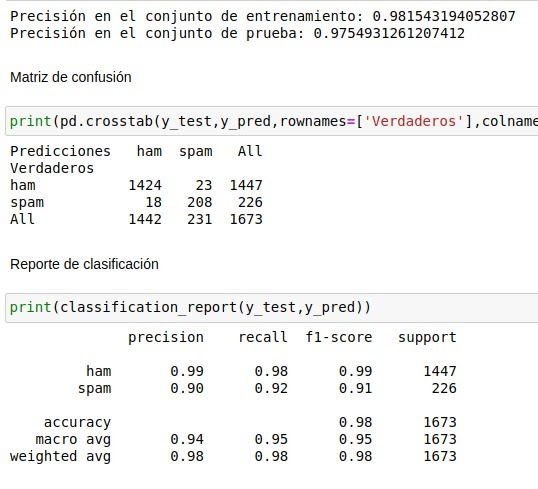
## **Prueba dos**

* Porcentaje de prueba de 0.3
* min\_df = 8



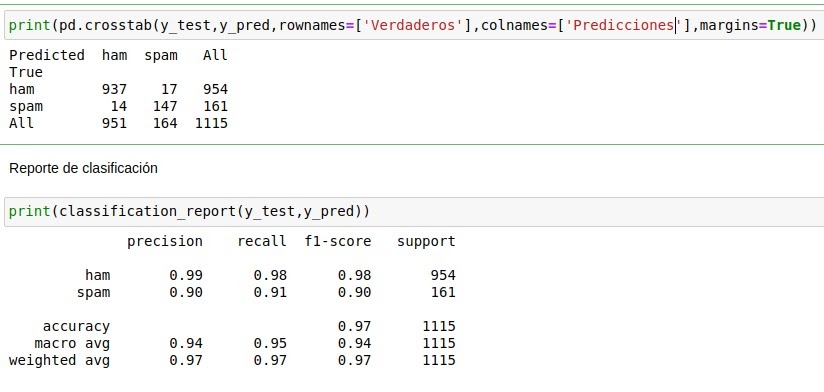
## **Prueba tres**

* Porcentaje de prueba de 0.3
* min\_df = 8



## **Prueba cuatro**

* Porcentaje de prueba de 0.2
* Se eliminaron los parámetros para la vectorización
* min\_df = 5



**Recursos y Referencias**

* Dasgupta Anijit (2018). Siliconindia. NAIVE BAYES: An Implementation Of Email Spam Filtering. Recuperado de: <https://www.kaggle.com/balaka18/email-spam-classification> el 27 de mayo del 2020.
* S/A (2019). Heliyon. Machine Learning for email spam filtering: review, approaches and open research problems. Recuperado de: <https://www.sciencedirect.com/science/article/pii/S2405844018353404> el 27 de mayo del 2020.
* S/A (S/A). qwewiki. Naive Bayes filtrado de spam - Naive Bayes spam filtering. Recuperado de: <https://es.qwe.wiki/wiki/Naive_Bayes_spam_filtering> el 27 de mayo del 2020.
* Milián Vladimir (2011). ResearchGate. Spam Detection. Recuperado de: <https://www.researchgate.net/publication/320161680_SPAM_DETECTION> el 27 de mayo del 2020.