

|  |
| --- |
| MANUAL TÉCNICO  CLASIFICADOR DE CORREO SPAM |
| ANALISIS Y PROCESAMIENTO DE TEXTOS  UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO  FACULTAD DE INGENIERÍA |
| 26/MAYO/2020  ALUMNO:   * Orozco Magadán Brandon * Tavares Rosales Gerardo |

REQUERIMIENTOS Y DATOS PRINCIPALES

Para el acceso al programa del clasificador de correos spam requerimos:

* El programa fue desarrollado en el lenguaje de programación Python por lo que requerimos contar con un intérprete de dicho lenguaje en una versión 3.x.x.
* El desarrollo fue mediante la herramienta de procesamiento de datos Jupyter Notebook.
* La información que necesita el programa para trabajar está basada en un Dataset recuperado de la página de Kaggle.

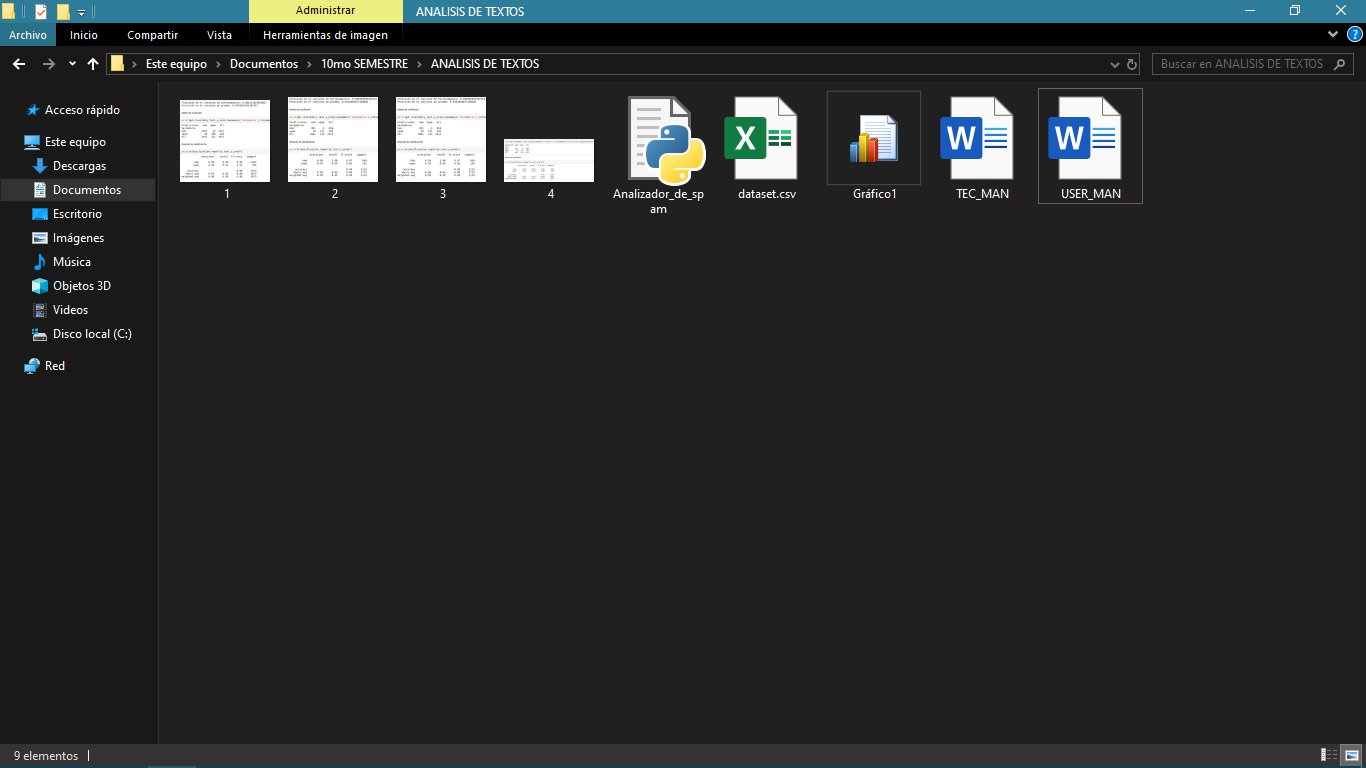
¿QUÉ NECESITAMOS SABER PARA EL CLASIFICADOR DE CORREOS SPAM?

* El funcionamiento se basa en Machine Learning, específicamente en el clasificador Naive Bayes.
* Este algoritmo se trata de una clase de algoritmos de aprendizaje automático basado en una técnica de clasificación estadística llamado “teorema de Bayes”.
* En este algoritmo se asume que las variables predictoras son independientes entre sí. En otras palabras, que la presencia de una cierta característica en un conjunto de datos no está en absoluto relacionada con la presencia de cualquier otra característica.
* El fundamento de este algoritmo se basa en el cálculo de la probabilidad ‘posterior’ de que ocurra un cierto evento Y, dadas algunas probabilidades de eventos ‘anteriores’.
* Algoritmo
  + Calcular la probabilidad previa para las etiquetas de clase dadas.
  + Determinar la probabilidad de probabilidad con cada atributo para cada clase.
  + Poner estos valores en el teorema de Bayes y calcular la probabilidad posterior.
  + Ver qué clase tiene una probabilidad más alta, dado que la variable de entrada pertenece a la clase de probabilidad más alta.

INTERACCIÓN Y ACCIÓN DEL PROGRAMA

**Archivos pertinentes**

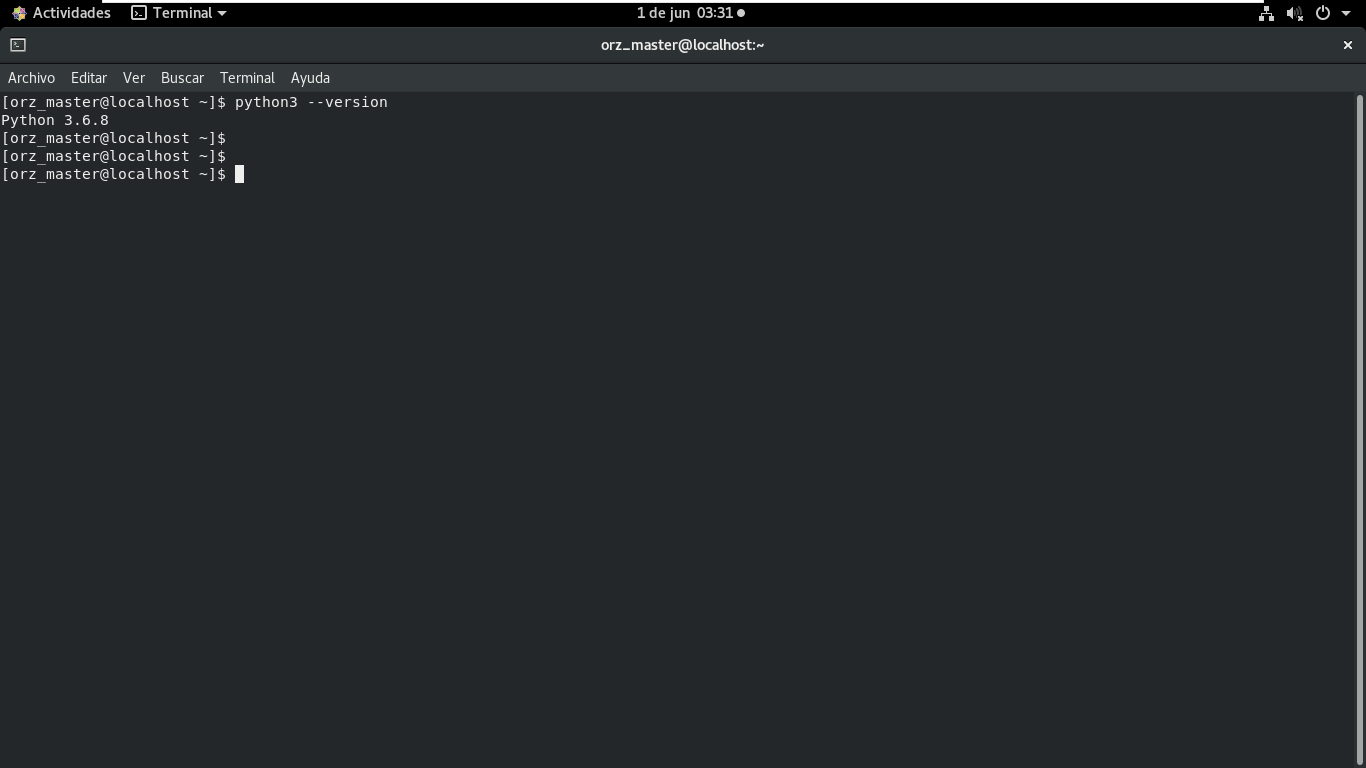
* Necesitamos del siguiente archivo que deberá estar contenido en la misma ruta en donde se vaya a ejecutar en caso de correrlo mediante un intérprete.
  + Analizador\_de\_spam.py
    - Este es el programa principal que contiene las librerías y toda la lógica de funcionamiento del algoritmo.



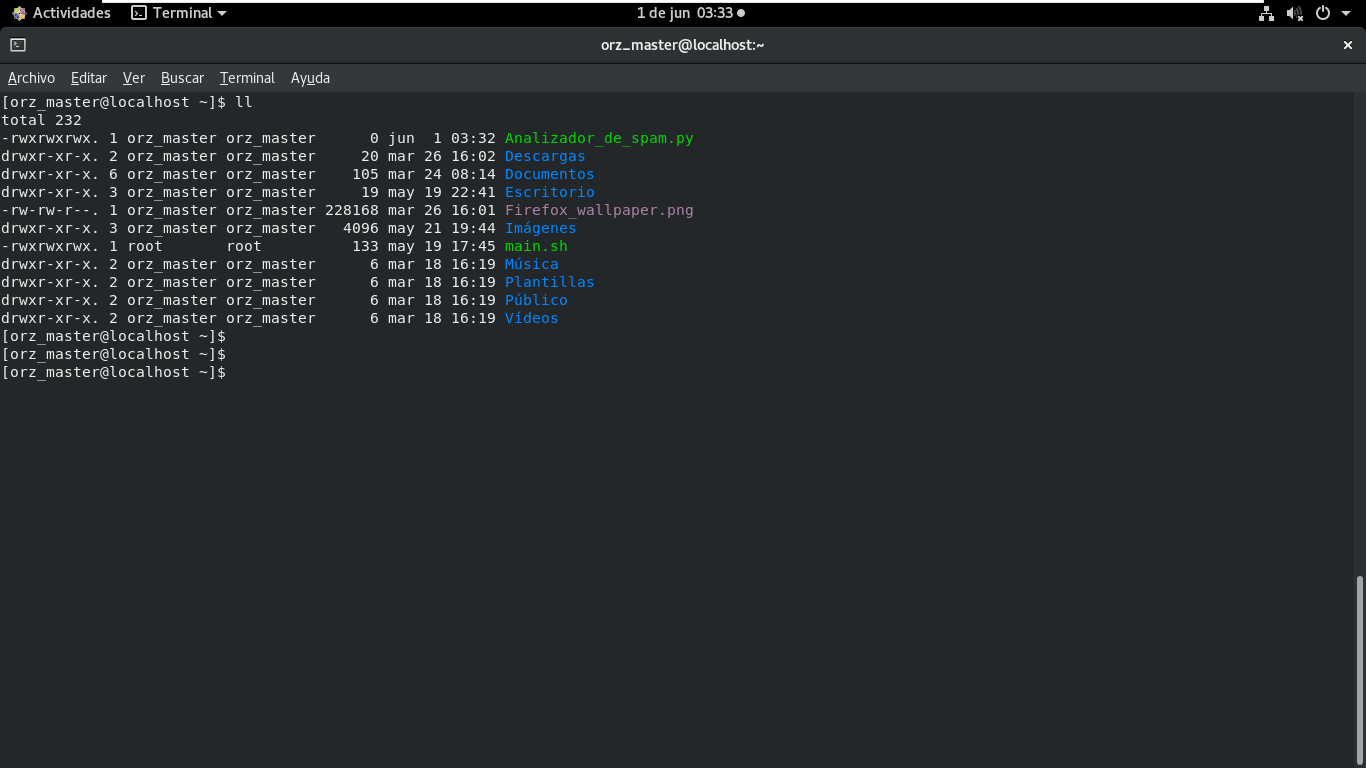
* Para hacer funcionar dicho programa requerimos de la siguiente secuencia de comandos en la terminal para el caso de Linux:
  + **Ejecución.**
    - Realizamos:
      * python3 analizador\_de\_spam.py

**Ejecución**

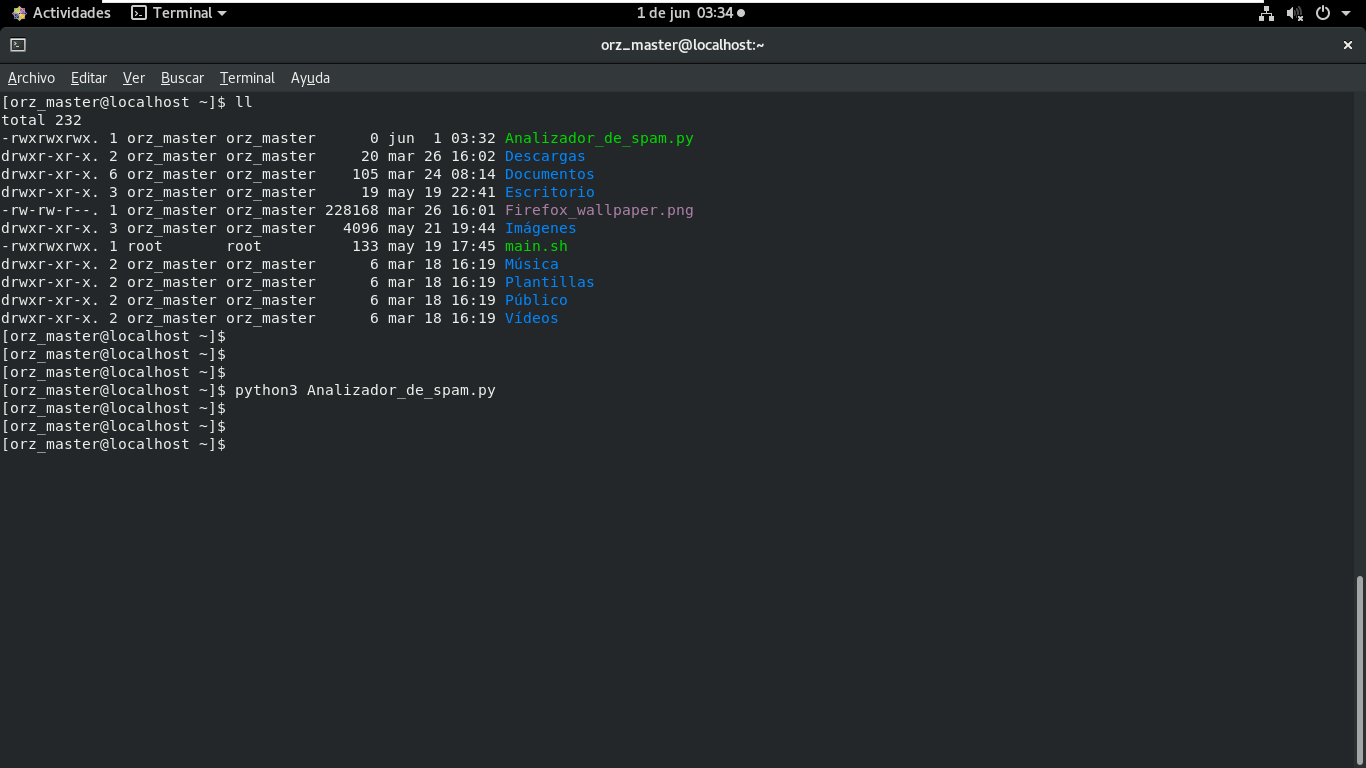
* Nuestro sistema operativo es CentOS 8 que pertenece a una distribución Debian.
* Dicho todo lo anterior, comenzamos paso a paso:
  + Corroborando que este el compilador “python3 *--version*”.



* Verificamos que tengamos el archivo Analizador\_de\_spam.py en el directorio donde haremos la ejecución:



* En nuestro caso usamos la herramienta de editor de texto *vim* para la programación de este algoritmo.
* Realizamos la ejecución:



Y podemos ver como enseguida comienza el funcionamiento del algoritmo.

**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.