

Universidad Nacional Experimental del Táchira

Vicerrectorado Académico

Decanato de Docencia

Departamento de Ingeniería Informática

**ANÁLISIS DEL CONTEXTO DE LAS UNIVERSIDADES VENEZOLANAS BASADO EN ANÁLISIS SENTIMENTAL EN REDES SOCIALES**

Autor: Angelica Landazabal

V- 20.626.929

angelica.landazabal@unet.edu.ve

San Cristóbal, Septiembre 2019

# DESARROLLO

Éste capítulo abarca la construcción del algoritmo comprendiendo la metodología SEMMA, siguiendo cada uno de los pasos correspondientes para obtener los resultados que ayudarán al proceso de la contextualización de la problemática presentada en los recintos universitarios de todo el país.

Esta información parte de una serie de datos obtenidos desde la red social de Twitter, el cuál difunde diferentes mensajes que explican la situación universitaria por el lapso de un año pertenecientes a las cuentas oficiales de las diversas casas de estudio, abarcando también las sedes de cada una de las instituciones.

A continuación se irá describiendo cada una de las fases de la metodología y herramientas que se van a utilizar para desarrollar al mismo tiempo el script necesario que mostrará más a detalle los resultados obteniendos y la descripción de su realización.

1. **Muestreo (Sample)**

En la primera etapa de éste proceso es donde se identifica el conjunto de datos con los que se va a trabajar. Para ésta investigación, se tomará cómo referencia todas las universidades adscritas al Ministerio del Poder Popular para la Educación Universitaria, Ciencia y Tecnología (MPPEUCT), para obtener el listado de las Universidades tanto públicas cómo privadas de Venezuela las cuales se almacenaran en un archivo de excel en dos columnas ( nombre de la institución y el nombre de usuario en Tweeter), y serán cargadas al script cómo un array que se utilizará cómo filtro en las búsquedas de los comentarios.

Se recolectó un total de 91 cuentas de Twitter para la investigación y están almacenadas en el archivo **CuentasUniversidades.xlsx.**  Entre éstas cuentas se encuentran todas las universidades sin importar su ideología politica ni su condición de autonomía.

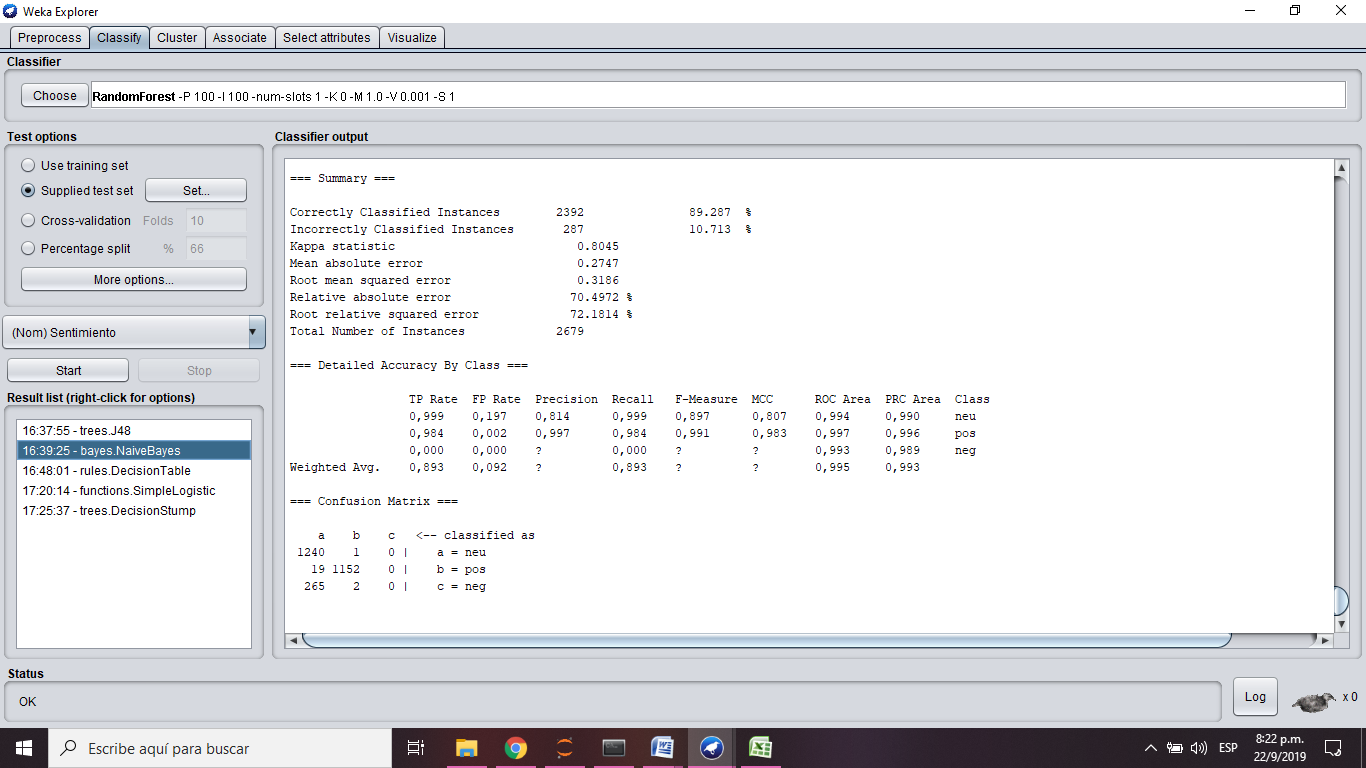
Los datos recaudados se extraen a través de la API propia de la red social de Twitter, dónde se obtienen las credenciales necesarias para establecer la conexión con la libreria de python “Tweetpy”; luego de la conexión se procede a capturar los datos de las cuentas de las universidades filtradas como una lista, y se evaluan consecuentemente con las fechas de inicio y fin del estudio que se quiere realizar. Los datos guardados pertenecen a las columnas de Fecha, Universidad, Tweet, cantidad de retweets y likes de esa publicación.

Para realizar el calculo correspondiente, se ejecuta el script recolectando la información de cada uno de los Tweets desde el 01/09/2018 hasta el 31/08/2019 y almacenados en un documento de excel cuyo nombre es dbUniversidades.xlsx. Este proceso se ejecutó en el software de código abierto de Jupyter en su versión 5.7.8.

En la recolección se obtiene una población de 19504 tweets, de los cuales 3900 fueron clasificados manualmente dependiendo del contexto del tweet, el restante 80% de los tweets fueron etiquetados a través de un script con el clasificador Naive Bayes. Para el etiquetado de cada uno de los tweets, se decide agregar una nueva columna al documento de excel con el nombre de “Sentimiento”, el cual contiene un valor del rango definido para la clasificación (pos: positivo, neg: negativo, y neu: neutral).

Del grupo que se clasificó manualmente, se utilizó el 75% para el entrenamiento del modelo dando un porcentaje de precisión del 75%, la precisión dada por el set de prueba fue del 58%. Se escogió utilizar el algoritmo de clasificación Naive Bayes debido al alto grado de porcentaje en la correcta clasificación de las instancias con un 89.28% (Ver figura 1) usando la herramienta de aprendizaje automático y minería de datos Weka versión 3.8.3.

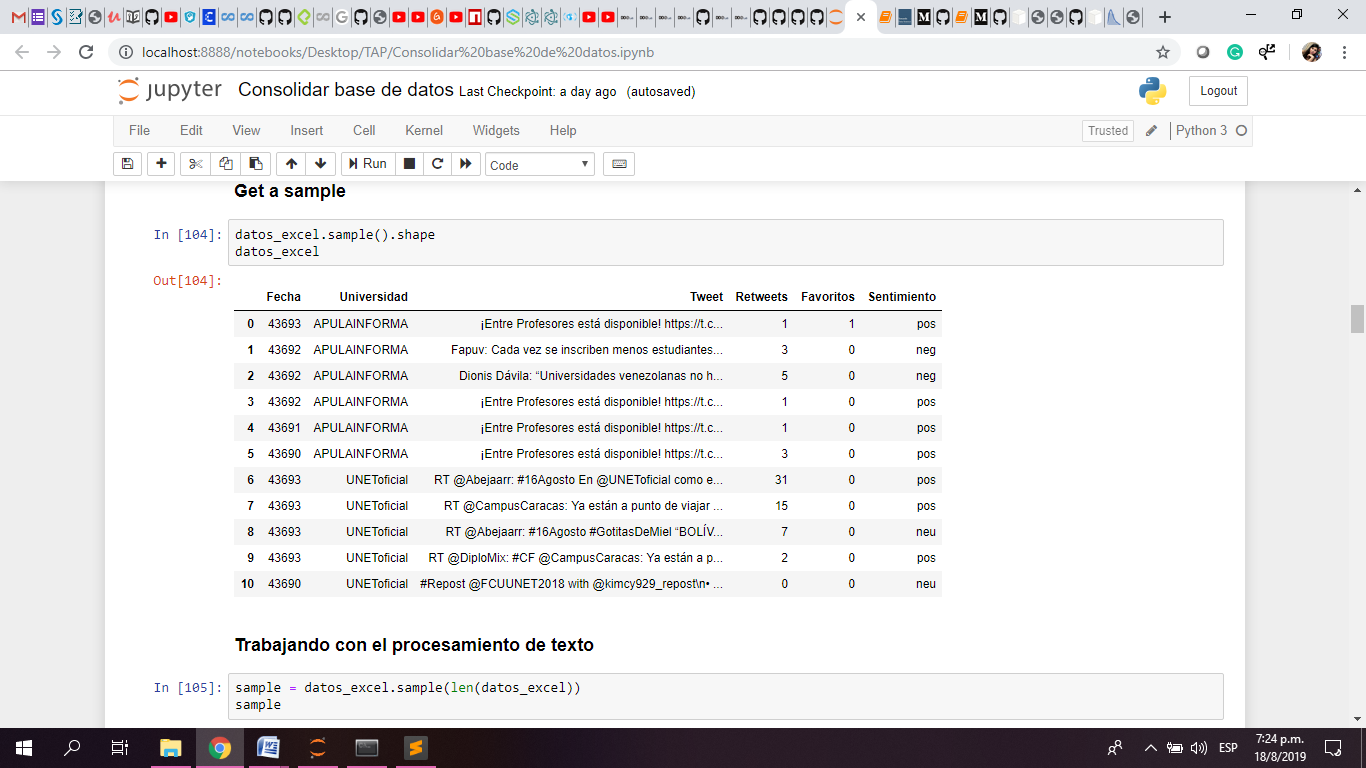
Otros modelos de clasificación utilizados fueron Regresión logistica con un 59.61% de instancias correctamente clasificadas, tablas de decisión con un 51.02%, arboles J48 con un 49.32% y arboles de decisión con un 47.29%.



**Figura 1.** Resumen de datos de la clasificación por Weka.

**Fuente:** Propia.

Como muestra del desarollo, se decide utilizar toda la población descrita anteriormente obteniendo un total de 19504 comentarios para las 91 cuentas utilizadas en el momento de la extracción ya clasificadas con los valores definidos.

****

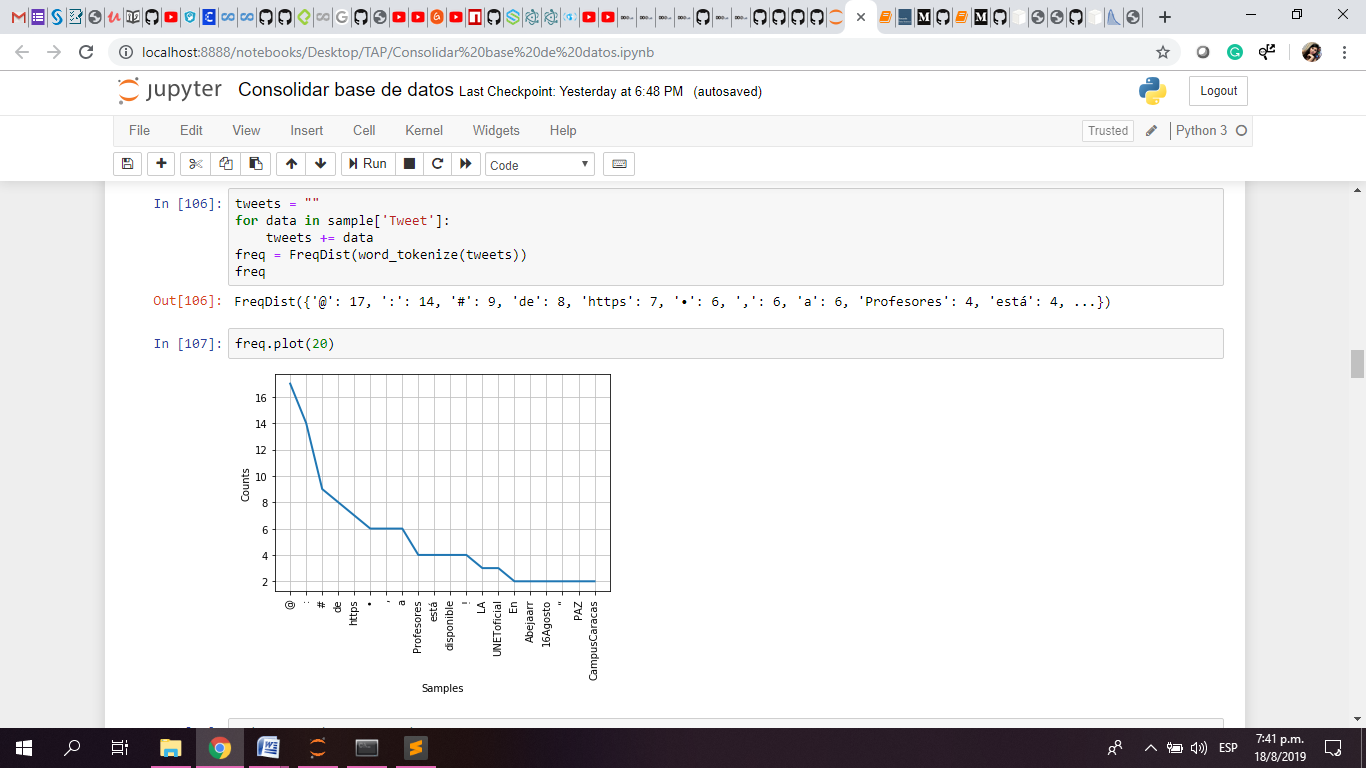
**Figura 2.** Dataset con los datos obtenidos desde la API de Tweeter.

**Fuente:** Propia.

En la tabla se puede detallar cada uno de los tweets recolectados junto con la información de la fecha, cuenta de la universidad a la que corresponde, cantidad de retweets y likes obtenidos.

1. **Exploración (Explore)**

Luego se exporta la librería Numpy para trabajar con las estructuras de datos con mayor facilidad, Matplotlib para la visualización de los datos, y NLTK para el tratamiento de las palabras. Todos los comentarios son concatenados en un mismo array donde luego es tokenizado para evaluar la frecuencia de cada una de las palabras. Consecuentemente, se gráfica la información obtenida por la función FreqDist para facilitar la ubicación de outliers y tendencias dentro del texto.



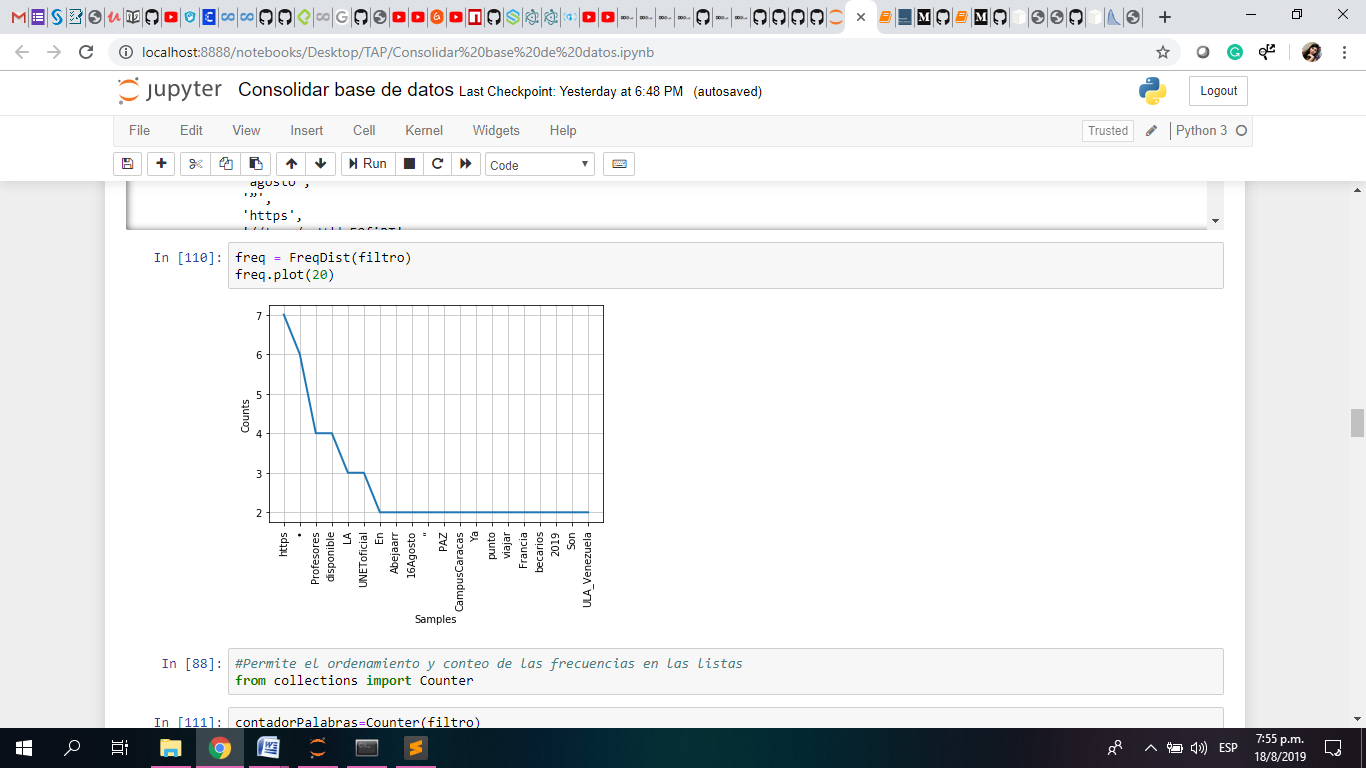
**Figura 3.** Comandos de Python para la obtención de la tokenización del array de los comentarios y el gráfico de frecuencia.

**Fuente:** Propia.

En el gráfico se puede observar las primeras 20 palabras con mayor frecuencia existente dentro del array de los comentarios de la muestra. Cabe destacar, que las palabras principales hacen referencia a artículos y algunos signos de puntuación que no dan valor alguno al análisis que se quiere realizar, por lo que se utilizará la técnica correspondiente para su eliminación.

1. **Modificación (Modify)**

Debido a que los datos gráficados en el paso anterior contienen palabras y signos de puntuación que aportan poco valor para el análisis correspondiente, se procederá a eliminarlos utilizando las funciones stopwords y punctuation de Python, para que solo quede en el corpus las palabras que puedan dar mayor significado al análisis deseado.



**Figura 4.** Comandos de Python para el calculo de la frecuencia y graficación del filtrado.

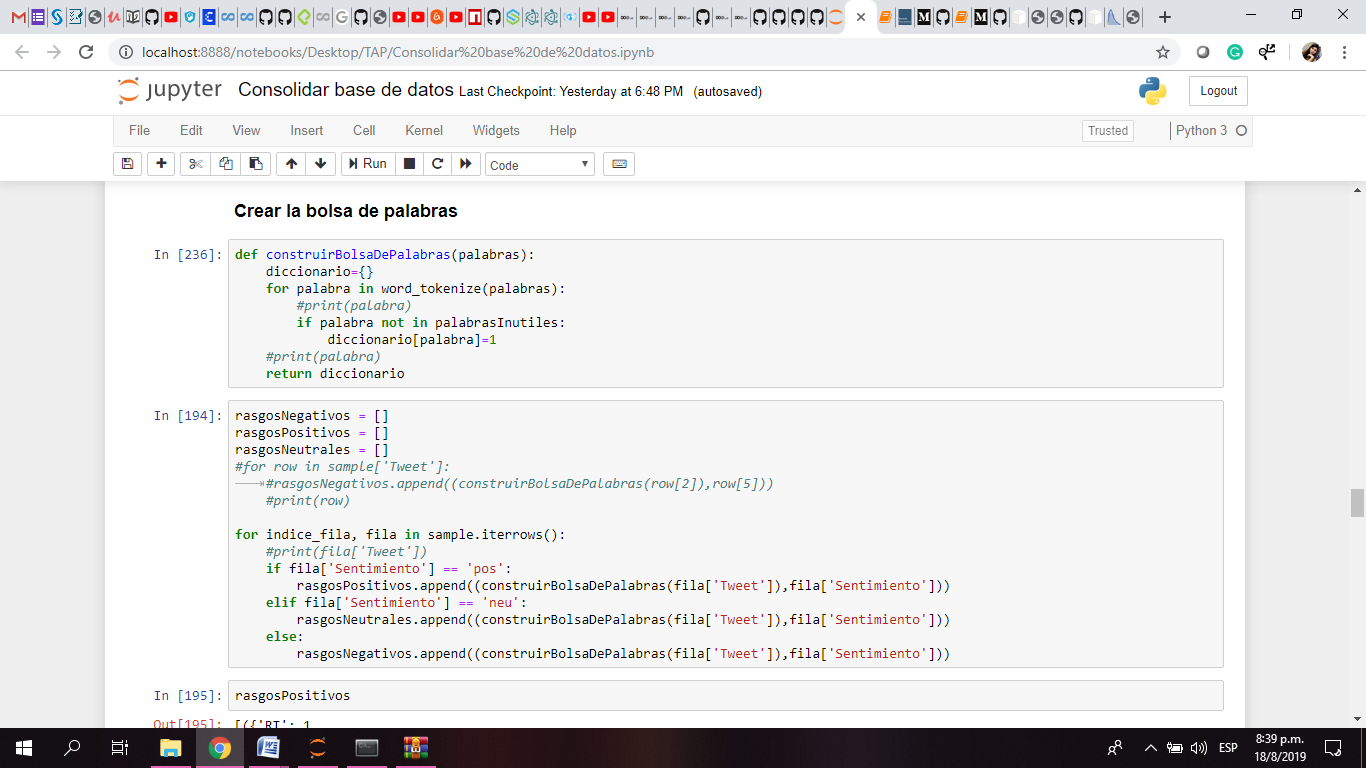
**Fuente:** Propia.

En el gráfico se puede apreciar, como se eliminarón aquellas palabras y signos de puntuación que le aportaban muy poco valor a la investigación para que no afectara al calculo del resultado.

Adicionalmente en ésta fase, se adicionó una nueva columna “Sentimiento” en el archivo “dbUniversidades.xlsx”, la cual se le asignó manualmente una etiqueta a cada comentario con el rango de pos(positivo), neg(negativo) y neu(neutro) para la clasificación del sentimiento. Esta será una de las variables con las que se trabajará al momento de hacer el entrenamiento del modelado junto con la columna del tweet.

1. **Modelado (Model)**

Para la ejecución de cada uno de los modelos conseguidos, se utilizará el 90% de los tweets como base de entrenamiento, y el 10% restante se utilizará como pruebas. Para conseguir cada uno de estos sets, se utilizará la técnica de la bolsa de palabras (BOW, en inglés Bag of Words), para filtrar dependiendo de la etiqueta del tweet, todas aquellas palabras que están contenidas en él.



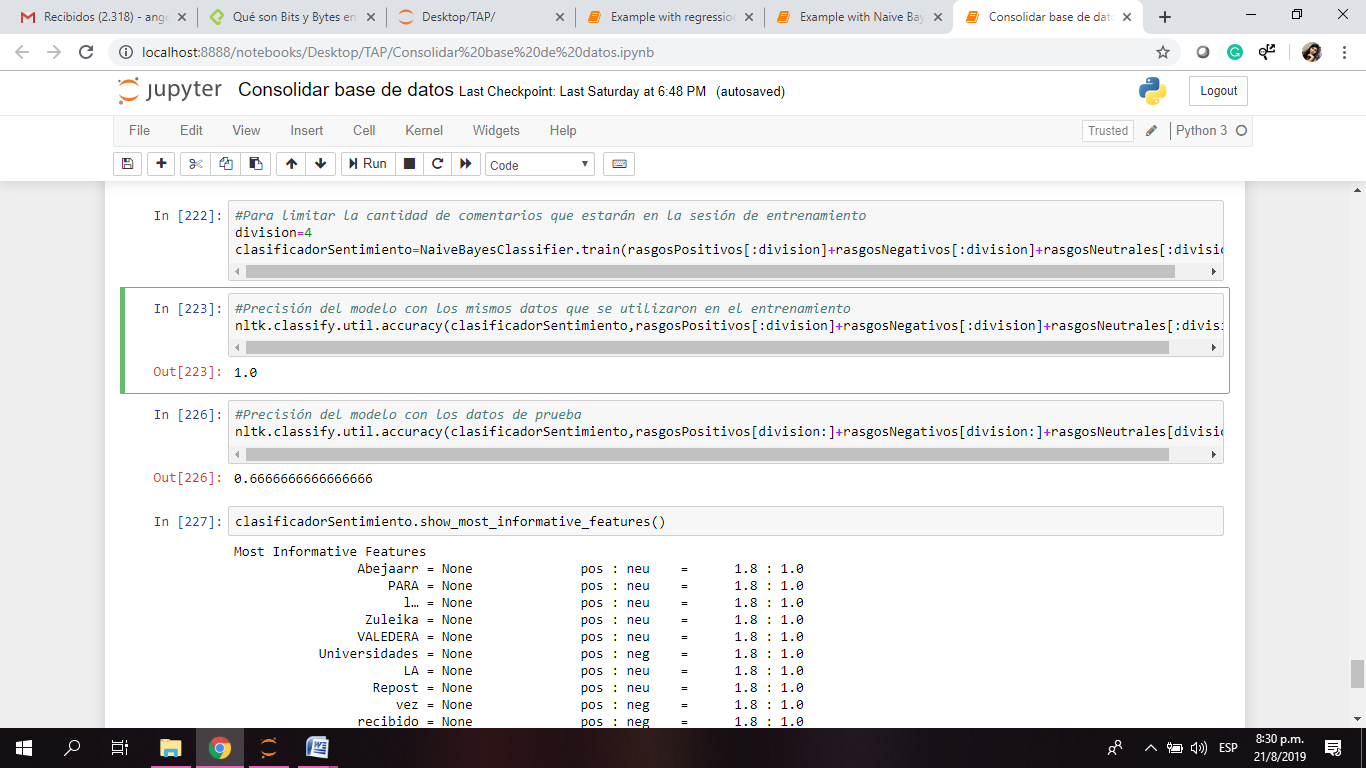
**Figura 4.** Comandos de Python para la creación de la bolsa de palabras de cada etiqueta.

**Fuente:** Propia.

En el script se puede apreciar la creación de 3 arrays (rasgosPositivos, rasgosNegativos y rasgosNeutrales) los cuales contendrán un conjunto de tuplas formados por un diccionario de palabras de cada tweet y la etiqueta del sentimiento correspondiente.

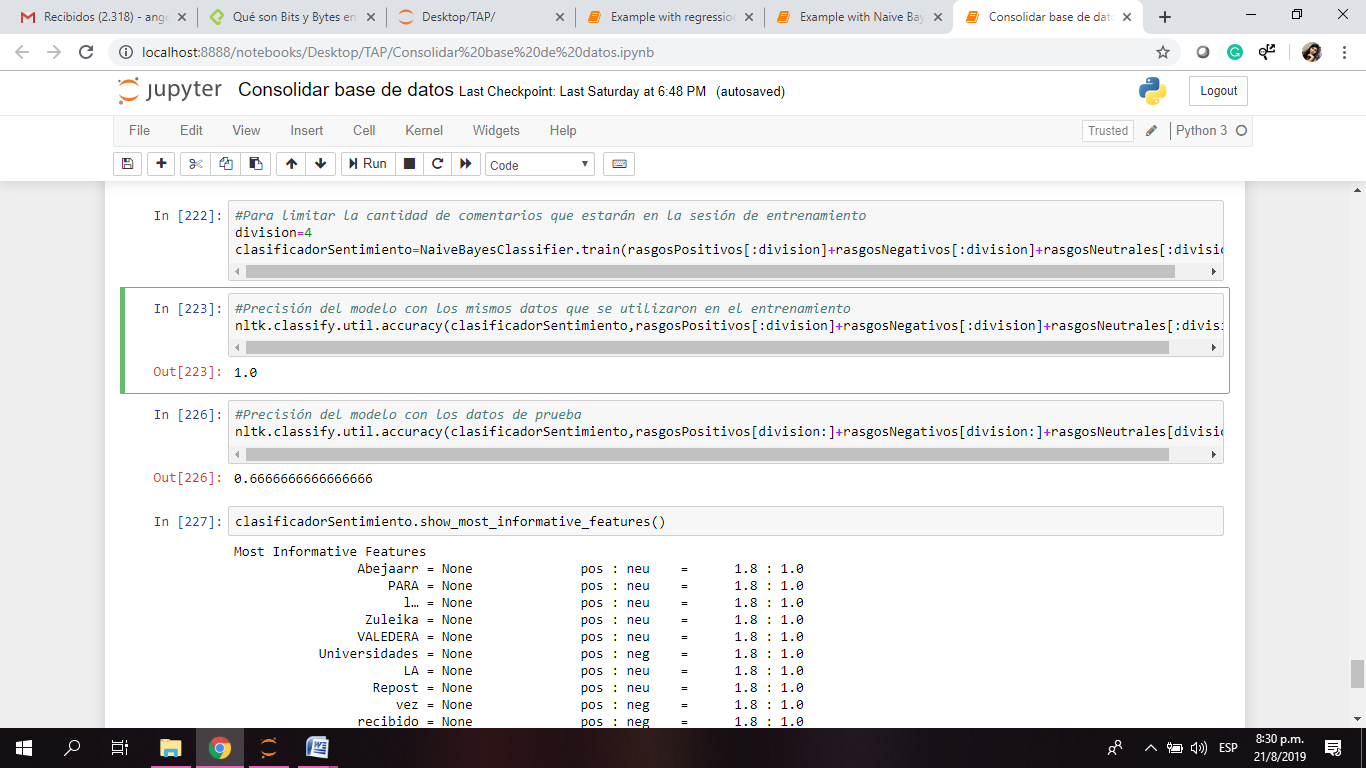
* + 1. **Naive Bayes**

Para éste modelo se utilizó el clasificador perteneciente a la librería NLTK de python, el cual contiene el método train para el entrenamiento del modelo con los grupos de datos ya separados manualmente en positivos, negativos y neutrales; y un método para el cálculo de la precisión del modelo para asegurar la validez en la clasificación de cada nuevo comentario introducido.



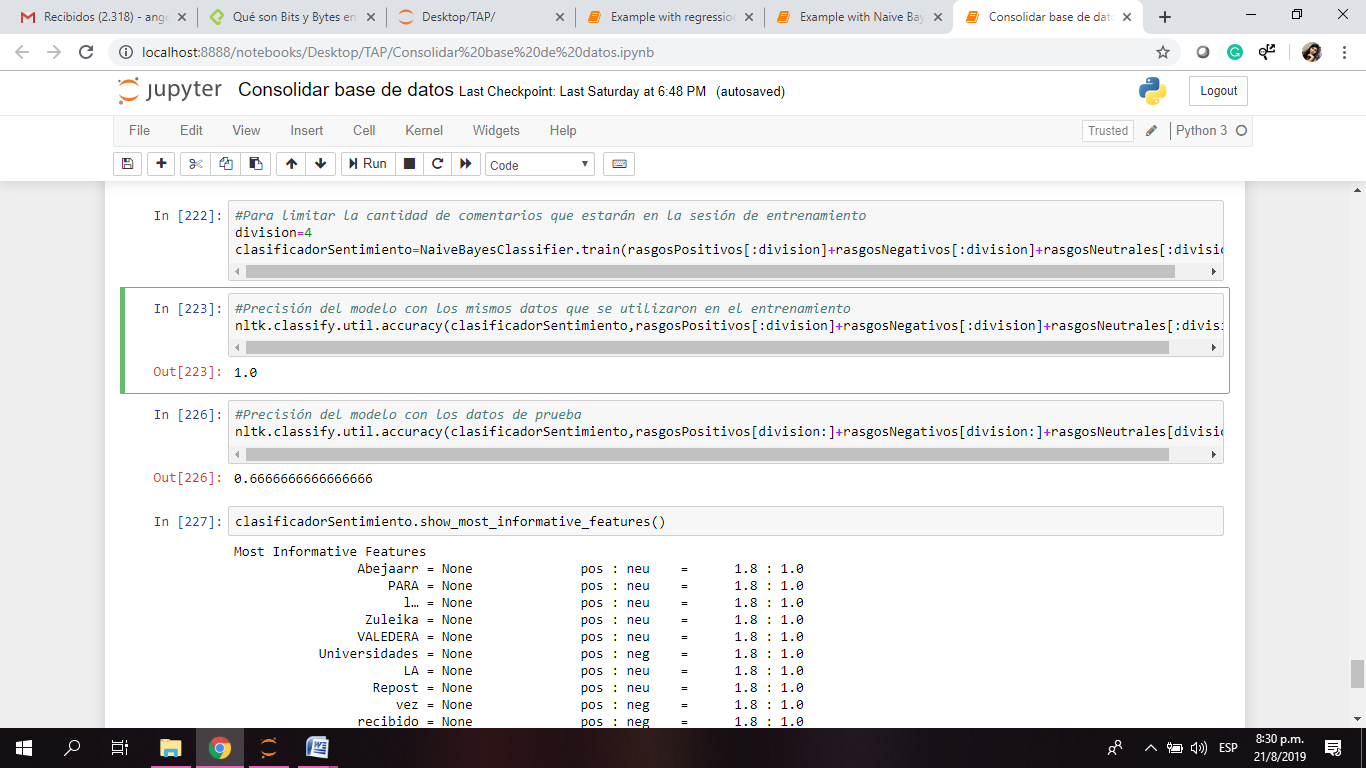
**Figura 5.** Comandos de Python para entrenar el clasificador del modelo Naive Bayes.

**Fuente:** Propia.



**Figura 6.** Comandos de Python para obtener la presición del clasificador con los mismos datos del entrenamiento.

**Fuente:** Propia.



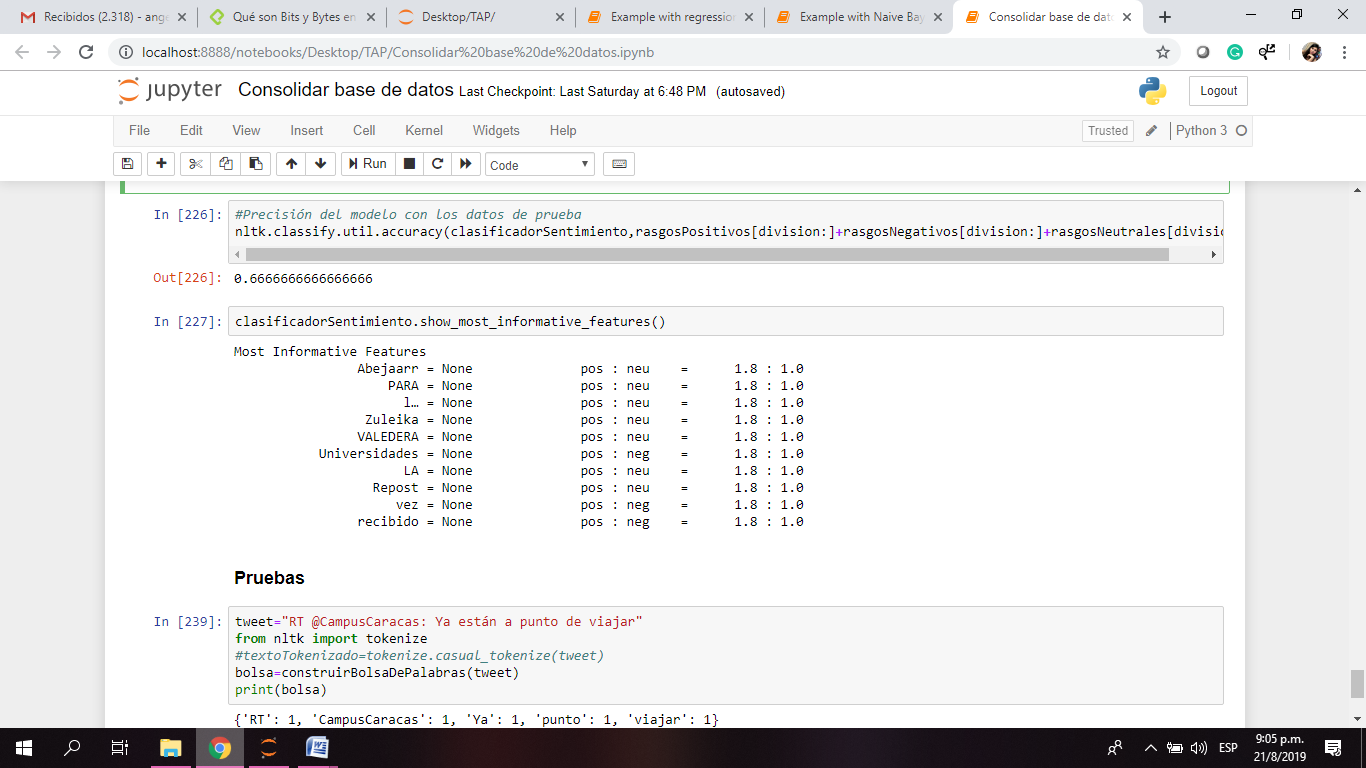
**Figura 7.** Comandos de Python para obtener la precisión de la clasificación de los datos de prueba.

**Fuente:** Propia.

Cómo se puede apreciar, se obtuvo un puntaje de un (1) punto de precisión para los valores introducidos del set de entrenamiento; y un puntaje del 0.67 de precisión para los datos del set de prueba.

1. **Valoración (Assess)**

Los resultados obtenidos de cada uno de los modelos, serán desplegados en ésta fase para realizar la comparación entre estos y contextualizar de la mejor manera el sentimiento expresado por la comunidad universitaria dado un rango de fechas determinado.

****

**Figura 8.** Comando de Python para obtener la clasificación de las palabras de la bolsa.

**Fuente:** Propia.

En la figura 8 se puede apreciar la clasificación y puntaje de cada una de las palabras con respecto a la etiqueta que fue asignada; en éste caso se tiene que la palabra “Universidades” tiene un puntaje de 1.8 para que se clasifique el comentario cómo positivo.