Machine learning para evaluar la calidad de resúmenes escritos por estudiantes con R y librerías de Python

Pedro Abraham Montoya Calzada

Introducción

Los lenguajes de programación: Python y R, son dos de los más utilizados en ciencia de datos y aprendizaje automático, cada lenguaje tiene sus ventajas y desventajas, por ejemplo: Python es excelente a la hora de crear modelos de aprendizaje automático, pues cuenta con librerías muy potentes como; TensorFlow, scikit-learn, pandas, etc. Que facilitan bastante esas tareas, sin embargo, el poder de R en el análisis estadístico y en le visualización de datos es sobresaliente.

Este proyecto lo hice con la intención de mostrar que se puede obtener lo mejor de ambos lenguajes, el objetivo de este trabajo es construir un modelo de regresión, que sea capas de dar una calificación de forma automática a resúmenes de texto creados por estudiantes. Utilizando en todo momento el lenguaje R pero combinándolo con dos de las más potentes librerías de Python: TensorFlow y scikit-learn.

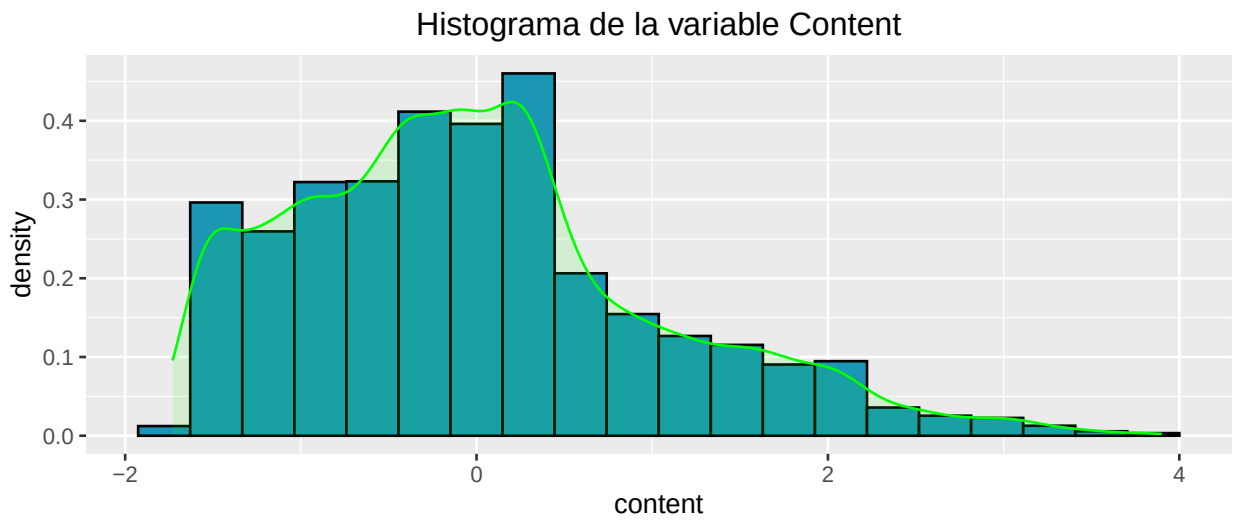
Los datos que se utilizaron fueron tomados estan disponibles **aquí.**

Los datos cuentan con 5 columnas, las dos primeras son id de identificación, la tercera es el texto escrito por los estudiantes, la cuarta es content: la puntuación de contenido para el resumen y la última columna es wording: la puntuación de redacción del resumen. La tercera columna será la entrada del algoritmo, y las ultimas dos columnas son el objetivo a predecir, sin embargo, en esta ocasión, solo se va a trabajar con la variable "content".

Análisis de datos y visualización

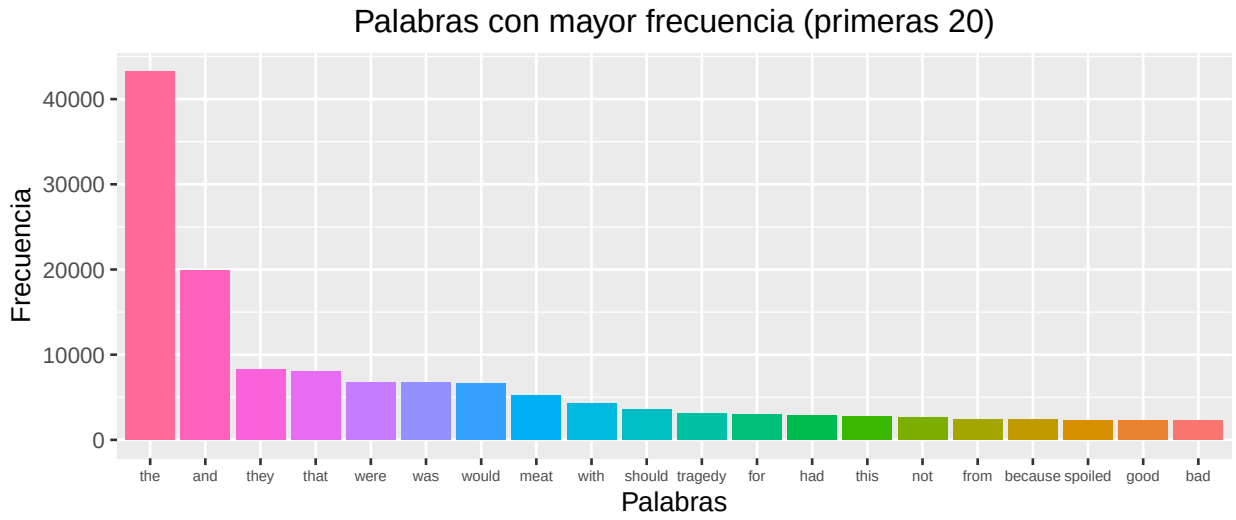
Estadísticas descriptivas de la puntuación del contenido

	Valor
Min.	-1.7298595
1st Qu.	-0.7995453
Median	-0.0938138
Mean	-0.0148530
3rd Qu.	0.4996599
Max.	3.9003261



Palabras más frecuentes

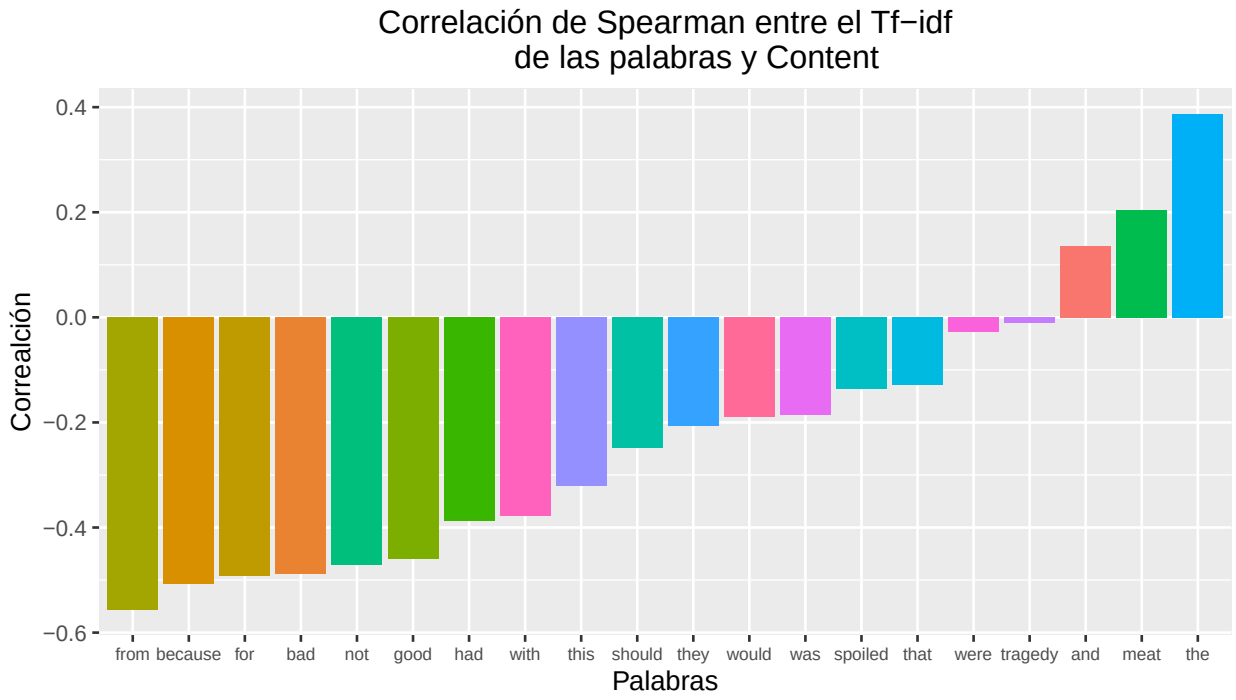
Las stop words (palabras vacías o palabras de paro en español) son palabras que se filtran o eliminan de un texto durante el procesamiento de lenguaje natural (NLP) porque se consideran comunes y poco informativas en términos de contenido semántico. Sin embargo, para este proyecto decidí no utilizarlas, porque si algún estudiante utiliza en exceso ese tipo de palabras, podría ser causa de penalización, y por lo tanto, recibir una calificación más baja.



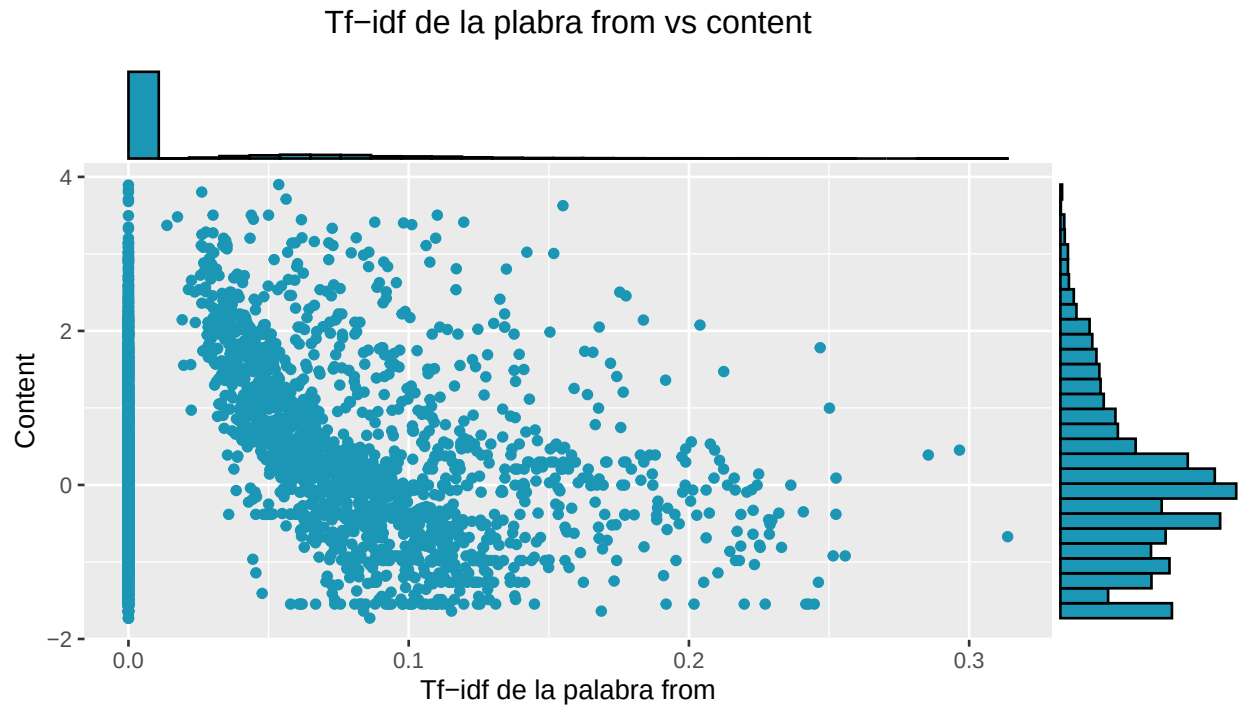
Relación entre las palabras y la calificación recibida

La transformación TF-IDF (Term Frequency-Inverse Document Frequency) es una técnica utilizada en el procesamiento de lenguaje natural (NLP) para evaluar la importancia relativa de una palabra en un documento dentro de un conjunto de documentos más amplio.

Ahora, vamos a ver como se relaciona el valor tf-idf de las 20 palabras más frecuentes con la variable respuesta: content.



La palabra "from" es la que presenta la relación más fuerte con la variable respuesta, vamos a ver el tipo de relación que hay con un diagrama de dispersión.



Creación y evaluación del modelo predictivo

El modelo que se construyo fue una red neuronal, se utilizo tensorflow y keras, a continuación, muestro la estructura de la red.

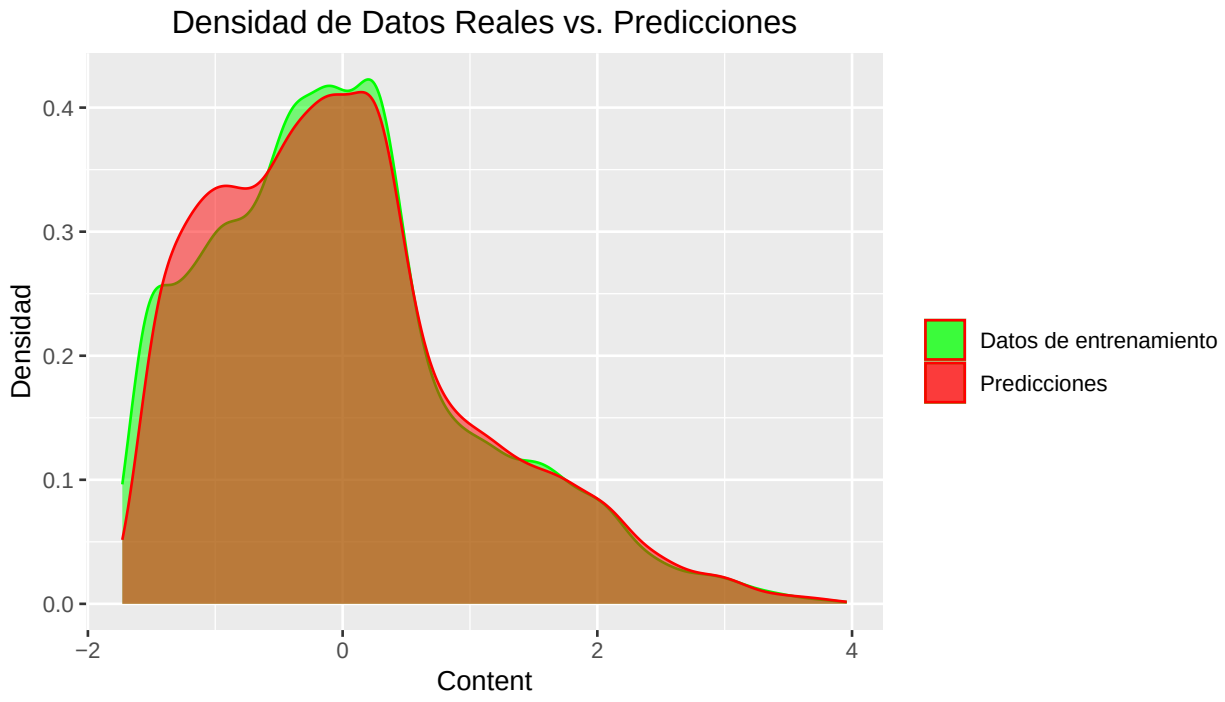
```
model <- keras_model_sequential(input_shape = shape) %>%
  layer_flatten() %>%
  layer_dense(5, activation = "relu") %>%
  layer_dense(5, activation = "relu") %>%
  layer_dense(1)

model %>% compile(
  optimizer = "adam",
  loss = "mse",
  metrics = "mse"
)
```

Para entrenar el modelo se tomo de manera aleatoria una muestra con el 80% de las observaciones, el otro 20% se utilizó para evaluar el desempeño del modelo.

Table 2: Evaluación del modelo con los datos de prueba

MSE	RMSE	MAE	R2
0.0193394	0.1390661	0.0860342	0.9820593



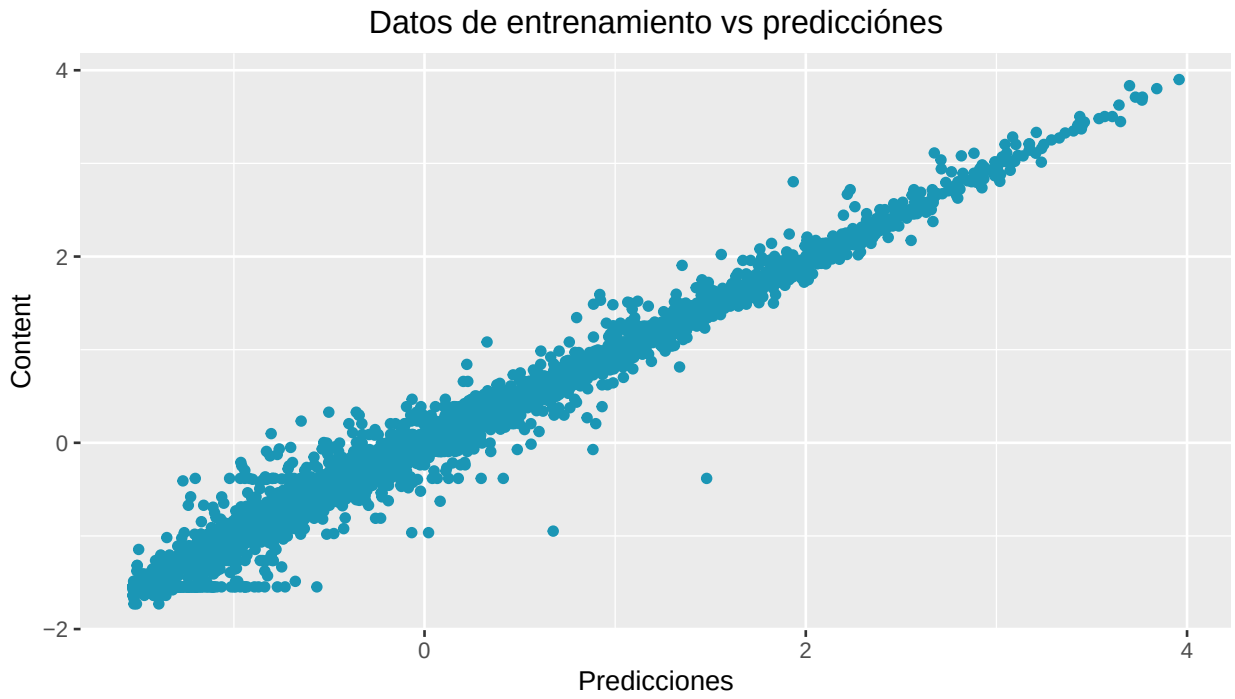
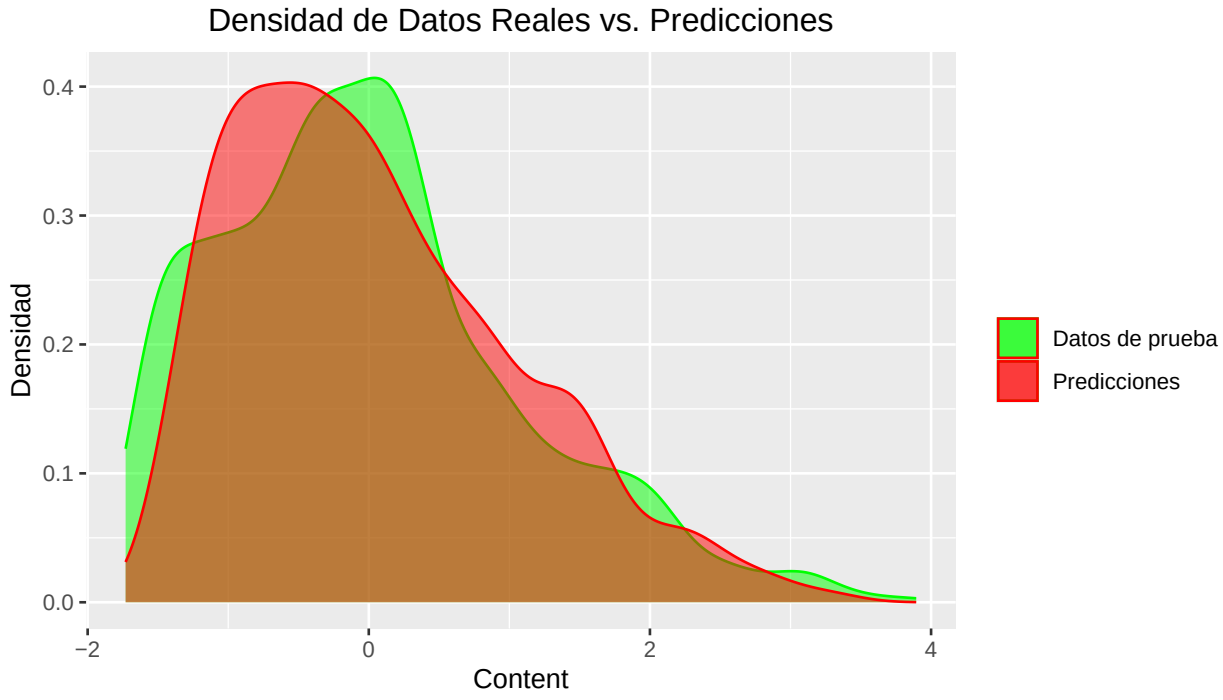
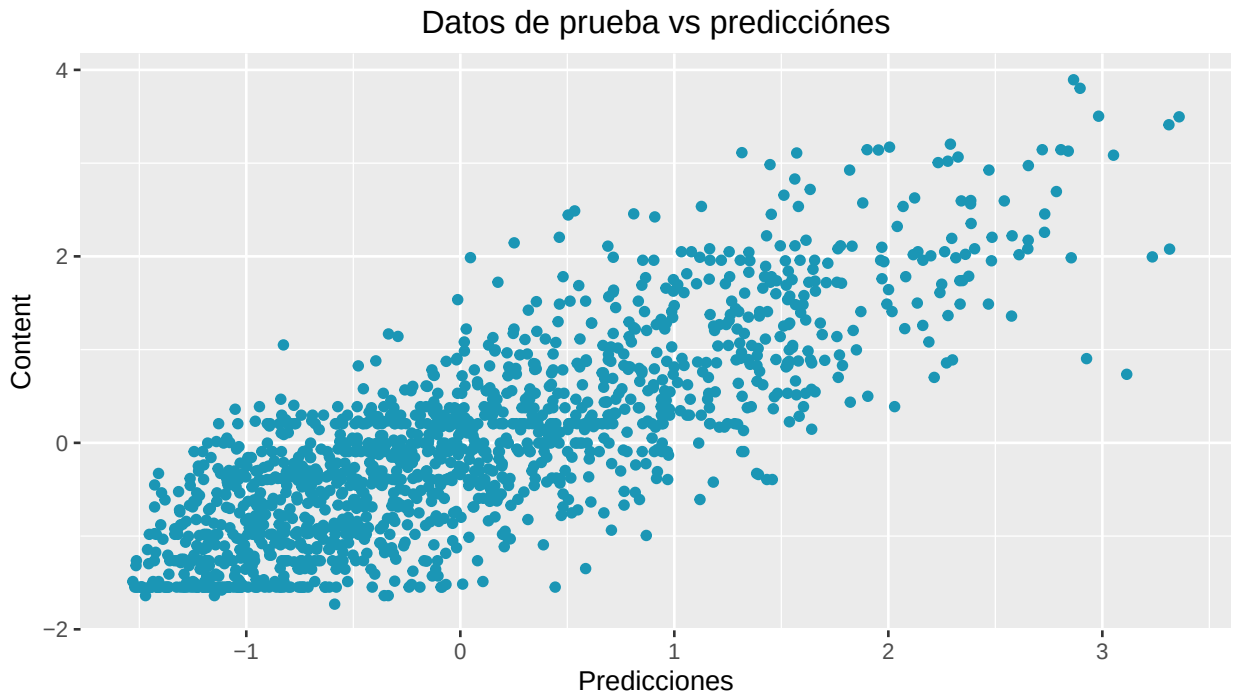


Table 3: Evaluación del modelo con los datos de prueba

MSE	RMSE	MAE	R2
0.4046988	0.6361594	0.5066237	0.6426693





Conclusiones

Se logro construir un modelo que puede predecir la calificación que un resumen de texto va a obtener, el modelo obtuvo un error cuadrático medio de 0.4046988 , lo cual es bastante bueno, sobre todo si se tiene en cuenta que es un modelo sencillo. Además, se pudo mostrar el potencial de ambos lenguajes de programación, pues, aunque todo se ejecuto en R, se utilizaron las que posiblemente sean las librerías más utilizadas en Python para ciencia de datos: tensorflow y sklearn. El modelo construido aquí se puede guardar y después cargar en un script de Python, y funcionará perfectamente en el entorno de Python, igual que como lo hizo en el entorno de R.