



DOCUMENTACIÓN DEL DESARROLLO

Python

Equipo anónimo:

Aldama Lagos Daniel

Morales Sánchez Yesica

Reynoso Sánchez Vivian Paola

Villarroel Hernandez Carlos

Zepeda Reyes Luis Daniel.

Creación del Back End

En la creación del back end se utilizó Python.

Se importaron librerías como numpy, panda o requests, por parte de nltk se importaron librerías de stopwords y Lemmatizer, de sklearn se usaron Count Vectorizer, train_test_split y el modelo naive bayes.

Naive bayes es una clase especial de algoritmos de clasificación de Aprendizaje Automatico, o Machine Learning.

El modelo sume 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.

Una vez instaladas nuestras librerías y escritas en nuestro archivo Python procedimos a cargar los archivos a modelar (test y train).

Obtenidos los datos cambiamos las locaciones 'nan' por 'none', enseguida configuramos las stopwords y posteriormente creamos una función para limpiar los tweets.

En la función de limpieza quitamos hipervínculos, HTML, números, menciones, puntuación, espacios y removimos las stopwords para al final aplicar dichos cambios al tweet original y guardarlo o retornarlo como un tweet limpio/nuevo.

Enseguida creamos una función de lematización y lo aplicamos al tweet limpio, esto da como resultado un tweet más pequeño.

Como tenemos 5 categorías o sentimientos que representar, asignamos un identificador o target a cada uno de ellos y creamos su columna.

De igual forma, creamos algunos filtros para obtener un menor número de lugares en donde se realizaron los tweets, en total se realizaron 24 filtros, entre ellos UK, Nueva Zelanda, Francia, Alemania, entre otros.

Creados los filtros creamos una nueva columna con los países de nuestros filtros y posteriormente cambiamos los nombres largos por un nombre de 3 dígitos.

Obtenida nuestros datos como queríamos, procedimos a aplicar el modelo naive bayes.

Para esto, transformamos el tweet limpio con vect.fit_transform a las variables X_train, X_test, y_train, y_test y usando como 'y' nuestro identificador de sentimiento. Separamos usando train_test_split y posteriormente aplicamos el modelo multinomia naive bayes para al finalizar mostrar los resultados.

En cuanto a las visualizaciones, nosotras utilizamos wordcloud y plotly principalmente. En la primera visualización se muestra la distribución de los tweets según su sentimiento, en la segunda muestra una nube de palabras expuestas en los tweets y en la tercera un mapa con los países donde se originaron los tweets y la cantidad de estos.

Un punto que no se pudo completar es la clasificación del tweet propuesto o ingresado por el usuario, se tenía la obtención, la sectorización, el array y la matriz de frecuencia, sin embargo, no se pudo continuar debido a que surgió un problema con la conexión del back end con el front end y tuvimos que realizar la conexión o no saldría nada. Solo hubiese faltado aplicar el modelo naive bayes, y sin embargo, nos surgió otra duda, ¿Cómo saber si es positivo, negativo, neutral, extremadamente negativo o extremadamente positivo?, el modelo nos da una precisión y otros parámetros, pero no una clasificación como tal, así que habíamos pensado en hacer una comparativa del resultado obtenido del tweet ingresado con los ya obtenidos de los csv.

Creación del Front End

Para la creación del front end se utilizó Flask en lugar de Django.

Se creo el archivo principal llamado index.html que contiene 3 secciones: Inicio, Acerca de y visualizaciones.

Posteriormente se crearon 3 templates:

1. about.html que contiene la información de quienes integran al equipo
2. home.html que contiene la caja de texto donde el usuario debe introducir el tweet, y
3. layout.html que contiene el menú para desplazarse por la página.

En cuanto a la conexión con las visualizaciones de la aplicación, se crearon 2 HTML desde la misma aplicación utilizando “output_file” y “pio.renderers.default”.

Estos HTML fueron incrustados posteriormente a la página.

Finalmente, se le dio estilos a la página mediante un css.

En el se especifica el background de la página, las etiquetas (h1, h4), la barra, botones, los contenedores y la zona de datos.

Algunas características especificadas son los colores, la orientación del texto, bordes, ancho, entre otros.