

ISIS3301 – Proyecto 1 Etapa 2 – El culto de Dijkstra

Proceso de automatización del proceso de preparación de datos, construcción del modelo, persistencia del modelo y acceso por medio de API:

- **Preparación de datos:**

Debido a que la base de datos de la que se parte es bastante sencilla (únicamente dos columnas, ambas de cadenas de caracteres), en principio no es tan necesaria una limpieza a fondo, pues solo basta con eliminar los nulos y verificar que la columna de 'class' (la que indica si el que hizo los comentarios se suicida o no) solo tuviera dos posibles valores: 'suicide' y 'non-suicide', cosa que se cumplió para el conjunto de entrenamiento. En cuanto a los datos de la columna 'text', que indica el comentario a evaluar, aparte de la eliminación de los comentarios nulos si fue necesario realizar un par de ajustes adicionales: eliminación de caracteres no ASCII (como emojis) y pasar todo a letras minúsculas. En cuanto al proceso de automatización, solo fue necesario automatizar el proceso de limpieza de valores relacionados a la columna 'text' pues es la base que se va a evaluar para verificar si el comentario realizado da indicios de que el usuario respectivo vaya a intentar quitarse la vida. La automatización de este proceso fue relativamente sencilla, pues se implementó un Tokenizer que, como parte del proceso de tokenización, se encarga también de eliminar los caracteres no ASCII del texto, y pasarlo todo a minúscula antes de generar los tokens.

- **Construcción del modelo:**

El modelo usado en esta etapa fue un modelo resultante de reentrenar el modelo de Random Forest presentado en la etapa anterior del proyecto, con 100 árboles y una profundidad media de 1761.47. Luego de esto, este modelo fue exportado como un archivo .joblib, para poder usarlo en la api.

- **Acceso por medio de API:**

Para la construcción de la API fue necesario usar Flask, puesto que FastAPI presentaba problemas. Para la construcción de esta API solamente se configuró para una petición POST en la que se ingresa un texto que corresponde a un comentario, el cual es luego evaluado por el modelo e imprime la predicción. Debido a problemas del peso de los archivos, el archivo .joblib no se pudo subir al repositorio, por lo que es necesario descargarlo desde la siguiente carpeta compartida: https://uniandes-my.sharepoint.com/personal/g_beltranl_uniandes_edu_co/_layouts/15/onedrive.aspx?id=%2Fpersonal%2Fg_beltranl_uniandes_edu_co%2FDocuments%2FEtapa2&ga=1. Para acceder a la api solo basta con descargar los archivos (y el .joblib y dejarlo dentro de la carpeta 'assets'), y ejecutar el comando 'python app.py' desde una terminal estando en el directorio principal del proyecto (el que se llama Etapa2).

- **Consideraciones:**

Durante este proceso de construcción del modelo, se encontró que puede ser bastante beneficioso para el proyecto el que se pueda integrar el nombre de usuario de quien haya hecho el comentario, para así poder evaluar previamente todos los comentarios

hechos por un usuario y buscar la manera de filtrar los comentarios que no den indicios de un intento de suicidio, en caso de que ese usuario lo haya intentado, para evitar falsos positivos. Además, el agregar esa columna podría abrir la puerta a ingresar una base de datos grande que contenga el comentario y el usuario, para así retornar la lista de usuarios propensos a intentar quitarse la vida de acuerdo a los comentarios. Otra adición que también puede ser interesante, aunque bastante complicado, integrar una columna en la que el usuario que hizo el comentario indique una calificación que le da a su salud mental (o de si está pensando en quitarse la vida) al momento de hacer el comentario, con el fin de tener un modelo más preciso.

Desarrollo de la aplicación y justificación

Como se mencionó en la etapa anterior del proyecto, existen bastantes partes que pueden estar interesadas en el uso del modelo: los administradores de las comunidades r/depression y r/suicide, los administradores/dueños de la red social Reddit, miembros de ONGs que buscan dar apoyo en salud mental, e incluso profesionales de la salud mental. Por este motivo, se podría considerar que todos ellos pueden ser usuarios de la aplicación.

En primer lugar, los administradores de las comunidades de r/depression y r/suicide pueden estar usando esta aplicación para ayudar a miembros que vean que puedan ser propensos a intentar quitarse la vida.

En segundo lugar, se tienen a los dueños/administradores de la red social, quienes puede que no se pongan en la tarea de evaluar usuarios, pero pueden considerar aprovechar la herramienta para distribuirla a los administradores de las diferentes comunidades presentes en la red social, para que así ellos la puedan usar para verificar si es probable que miembros de estas necesiten ayuda.

En tercer lugar, las ONGs que buscan dar apoyo en salud mental pueden usar esta herramienta para evaluar comentarios en diferentes comunidades de Reddit (o incluso de otras redes sociales ya que es bastante genérica) para así buscar las que presenten una alta cantidad de predicciones positivas y buscar brindar ayuda a modo de: ayuda directa al contactar a esos usuarios o administradores, publicidad dirigida específicamente a esas comunidades, o directamente con los administradores o dueños de la comunidad/red social para así buscar la forma de hacer alguna campaña para brindar ayuda.

Finalmente, se tienen los profesionales de salud mental, quienes pueden hacer uso de esta aplicación a modo de una herramienta adicional para apoyarse en el diagnóstico de sus pacientes y brindarles una mejor ayuda.

Resultados

El resultado de esta etapa fue una aplicación que por medio de una API permite al usuario ingresar un comentario a modo de texto y tras presionar un botón, esta imprime en pantalla la predicción de si la persona que hizo ese comentario presenta tendencias suicidas:

Input text to be predicted:

Text

This person has non-suicide tendencies

Predict!

Figura 1. Ejemplo de la aplicación

En cuanto a la precisión del modelo, se cuenta con un valor de f1 de 0.895, lo que indica que es bastante preciso, aunque puede mejorar. Además, se cuenta que para el conjunto de entrenamiento solo hubo 6 falsos negativos y 3 falsos positivos entre unos 160000 datos, un valor que aumenta considerablemente para el conjunto de prueba que de unos 40000 datos, hubo casi 2000 falsos positivos y 2000 falsos negativos, un 5%.

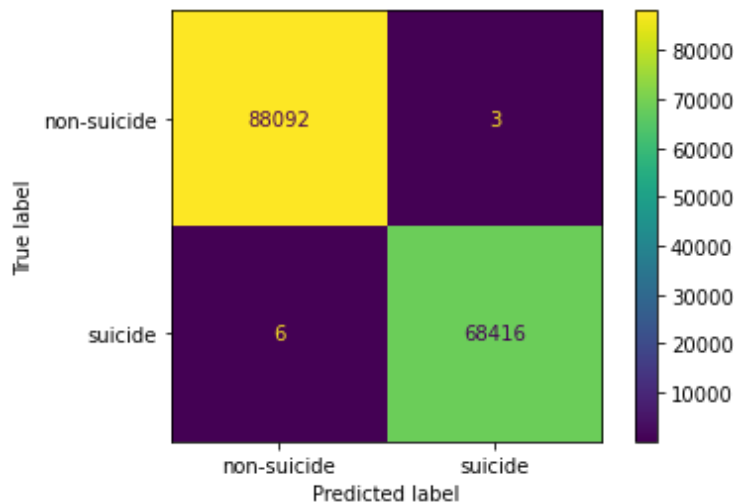


Figura 2. Matriz de confusión para el conjunto de entrenamiento.

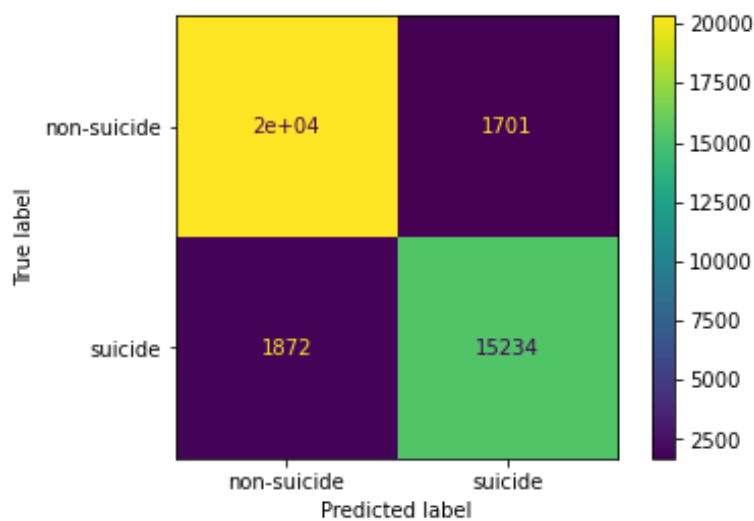


Figura 3. Matriz de confusión del conjunto de entrenamiento.

Trabajo en equipo

Para el desarrollo de esta etapa solo se hizo una reunión en la que se realizó un proceso de lectura y entendimiento del enunciado, y de avance en la construcción de la API. Luego de esto se distribuyó el trabajo a realizar y la principal comunicación fue por medio de mensajes de texto hasta el momento de la entrega.