



PROYECTO 1 – ETAPA 2

Apoyo a la detección de suicidios mediante aprendizaje automático con procesamiento del lenguaje natural



Grupo 7

Kevin Cohen Solano – 202011864

Juan Felipe Castro Vanegas – 201818130

Juan Carlos Marín Morales – 202013973

Tabla de contenido

Objetivos	3
Roles	3
Ingeniero de Datos.....	3
Ingeniero de software responsable de desarrollar la aplicación final:	4
Ingeniero de software responsable del diseño de la aplicación y resultados.....	5
Trabajo en equipo	6

Objetivos

- Automatizar un proceso replicable para aplicar la metodología de analítica de textos en la construcción de modelos analíticos
- Desarrollar una aplicación que utilice un modelo analítico basado en aprendizaje automático y sea de interés para un rol dentro de una organización

Roles

★ **Ingeniero de datos:** Es responsable de velar por la calidad del proceso de automatización relacionado con la construcción del modelo analítico.

★ Por: Kevin Cohen Solano

Automatización del proceso:

Preparación de los datos:

El proceso de preparación de los datos es similar al realizado en la etapa 1 del proyecto, en la cual se siguieron los siguientes pasos:

- **Transformar la columna 'class' a una variable numérica de 1s y 0s**, esto con tal de que los algoritmos propuestos sean capaces de asignar una clase en sus test.

- **Pasar todos los textos a minúsculas**, pues facilita la creación de vocabulario al no tener elementos diferentes que realmente sean los mismos.

Eliminar los números de los textos, para evitar que sucedan errores en la generación del vocabulario.

Tokenizar los textos: esta operación se encarga de generar un vocabulario de las diferentes palabras que se encuentran en los datos. Esta operación consta de asignar a cada palabra un token y separar cada texto en estos últimos en forma de una lista, teniendo en cuenta una expresión regular que busca eliminar aquellos elementos ajenos al lenguaje inglés (tales como emojis, caracteres no ASCII, etc). Después de esto se le aplica a cada token una función lemmatizer, la cual se encarga de unificar las diferentes conjugaciones y formas de las palabras en una sola para facilitar el procesamiento posterior de los algoritmos. Después de esto, las listas de listas se vuelven a unificar en un texto ya estando limpias.

Se eliminan los artículos, las conjunciones y las preposiciones, pues estas no aportan nada a los algoritmos. Algunas palabras se eliminan de manera manual, mientras que la mayoría de las StopWords se eliminan con ayuda del módulo que contiene la información de estas.

La diferencia con la primera parte es que no ejecutamos estas transformaciones directamente sobre el dataframe, sino que creamos unas funciones que harán parte de unas clases que luego serán utilizadas por el pipeline.

Construcción del modelo:

Para la construcción del modelo realizamos la creación de un pipeline, pues con la ayuda de las clases Texto_Basico y Texto_Definitivo condensamos todo el preprocesamiento de los datos (desde limpieza hasta tokenización) para que los utilicemos como pasos del pipeline, los cuales terminan con una vectorización mediante el modulo TfidfVectorizer y la aplicación del algoritmo de Naive Bayes, el cual funciona con la aplicación del teorema de Bayes y probabilidad condicional, según las diferentes características (palabras vectorizadas) y su influencia en la decisión de si son de una clase u otra.

Persistencia del modelo analítico:

Para la persistencia solo debimos exportar el pipeline con ayuda de la librería joblib y su función dump, guardándose en un archivo llamado "Proyecto1-Parte2Pro.joblib".

- ★ **Ingeniero de software responsable de desarrollar la aplicación final:**
Gestiona el proceso de construcción de la aplicación.
- ★ Juan Carlos Marín.

Aplicación final:

Creación de la aplicación a partir del modelo:

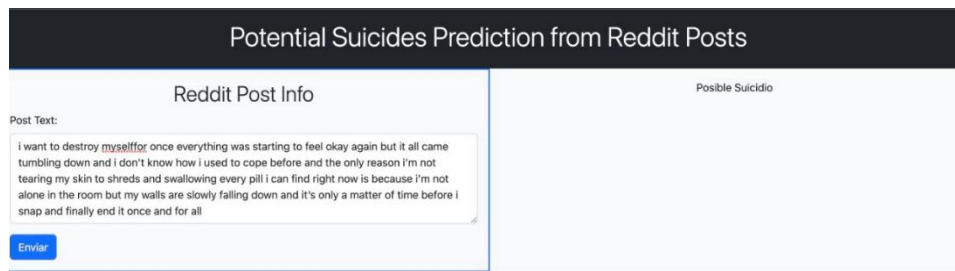
Para la creación de la aplicación, utilizamos el framework de FastAPI, donde creamos un API con servicios donde apartir de uno o más textos provenientes de publicaciones en reddit se obtenga una predicción sobre si es posible que exista un suicidio o no usando el pipeline creado para el modelo. Este servicio se ve como en la siguiente imagen.



Poner el modelo a disposición del usuario.

Para ponerlo a disposición del usuario final, decidimos crear una aplicación web, en la que el usuario pueda introducir un texto obtenido de publicaciones de reddit, y que la aplicación, a partir de una comunicación con el api, le diga al usuario si este puede llegar a ser un posible suicidio o no. Para la

construcción de esta aplicación usamos el lenguaje de JavaScript usando el Framework de React. Un ejemplo del uso de la aplicación es el siguiente.



También se puede encontrar un video del funcionamiento en el siguiente enlace: [Funcionamiento App Web](#)

- ★ **Ingeniero de software responsable del diseño de la aplicación y resultados:** Se encarga de liderar el diseño de la aplicación y de la generación del video con los resultados obtenidos.
- ★ Juan Felipe Castro Vanegas

Usuario final y relacionamiento con el cliente:

Descripción del usuario final de la aplicación, junto con la importancia de esta para el negocio y el usuario:

Nuestro modelo será utilizado principalmente por escuelas, las cuales les proveerán el servicio a los padres de los estudiantes, para que ellos puedan consumir nuestro producto mediante el API que construimos. Esto hará que los padres tengan control sobre el verdadero significado de lo que sus hijos están publicando en sus redes sociales. Para lograr esto se acordará una capacitación con la gente respectiva en cada colegio para la correcta utilización del modelo.

Justificación-Conexión entre la aplicación y el proceso de negocio que va a apoyar:

Nuestra aplicación está en la capacidad de ayudar a personas con tendencias suicidas, por tal motivo, podemos afirmar que se justifica la utilización de nuestro modelo para complementar el trabajo de los médicos expertos en estos temas, para poder desempeñar una mejor función como profesionales y así poder salvar muchas más vidas con antelación.

Trabajo en Equipo

Líder de proyecto: Kevin Cohen Solano

El proyecto fue realizado en un periodo intensivo de aproximadamente 19 horas, donde tuvimos ciertos desafíos con la creación del Pipeline y la ejecución de la aplicación Web, pues a la hora de cargar el pipeline, los transformadores personalizados generaban problemas que tuvimos que investigar para poder solucionarlos. Más allá de esto, fue más un proceso de entendimiento de nuevos elementos y aplicación de otros conocidos que con trabajo conjunto salieron adelante exitosamente.

A continuación, las reuniones que realizamos:

Fecha	Razón	Asistentes
Octubre 24	Reunión de lanzamiento y planeación	Kevin, Carlos, Felipe
Noviembre 4	Seguimiento general, solución de problemas y aclaración de dudas	Kevin, Carlos, Felipe
Noviembre 13	Consolidación final, detalles finales	Kevin, Carlos, Felipe

En una escala de 100 puntos proponemos la siguiente repartición:

- Juan Felipe Castro: 33.33
- Kevin Cohen: 33.33
- Juan Carlos Marín: 33.33