Proyecto 1: Analítica de Textos – Entrega 2

Gabriela Cagua Bolívar - 201812944

Juan Andrés Méndez Galvis - 20181580138

Juan Andrés Romero Colmenares - 202013449

1. Descripción de Roles

Para el desarrollo de este proyecto cada uno de los integrantes asumió un rol específico para que hubiera una buena distribución de trabajo. A continuación, se presentan los roles que desempeñó cada estudiante:

- Juan Andrés Romero:

Líder de proyecto: Encargado de la gestión del proyecto. Define fechas de reuniones, entregables del grupo, verificación de tareas.

- Ingeniero de software responsable de desarrollar la aplicación final: Encargado de gestionar el proceso de construcción de la aplicación

- Horas de trabajo: 8
- Retos afrontados: Construcción de la pipeline desde cero, uso de templates de FastAPI sin previo conocimiento sobre el tema.

- Juan Andrés Méndez:

Ingeniero de software responsable del diseño de la aplicación y resultados: Se encarga de liderar el diseño de la aplicación y de la coordinación del video con los resultados obtenidos

- Horas de trabajo: 8
- Retos afrontados:

- Gabriela Cagua

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

- Horas de trabajo: 8
- Retos afrontados: Hacer cuadrar bien los estilos y diseño de la página web, no hubo problemas con el proceso de automatización

Distribución de puntos entre integrantes del equipo:

Para esta entrega consideramos que todos hicimos un buen trabajo y fue distribuido de manera bastante equitativa, por lo tanto, pensamos que si se fueran a distribuir 100 puntos entre todos, cada uno recibiría 33.33 periódico, lo que equivale a una distribución igualitaria.

2. **Proceso de automatización del proceso de preparación de datos, construcción del modelo, persistencia del modelo y acceso por medio de API:**
   En cuanto a la implementación del API y el despliegue del código, se utilizó la el framework de FastAPI de Python como motor principal del proyecto y Tailwind CSS para algunos detalles estéticos de la página principal de la aplicación. En principio se desarrolló un único endpoint llamado /predict, el cual recibe una cadena de texto por medio del body del POST, la cual es analizada de la mano del modelo de Machine Learning implementado y devuelve la predicción de dicho modelo a quien realice la request. De igual manera, debido a la naturaleza del procesamiento de lenguaje natural y los resultados obtenidos en la parte 1 del proyecto, los cuales consideramos como satisfactorios, no vimos necesaria la modificación sobre los datos de entrada para el desarrollo de esta entrega. Es decir, no se realizó ningún tipo de integración de nuevas features ni limpieza adicional de datos para el despliegue del API del proyecto.

   Por otra parte, para mejorar la automatización y despliegue del modelo seleccionado, decidimos crear un pipeline que tomara los datos de entrada y los modificara de acuerdo a las necesidades del modelo. En principio, el pipeline implementado tiene pocos elementos, se basa en un TFIDF Vectorizer y el modelo de Regresión Logística desarrollado en la entrega pasada, en estos, el mejor modelo usó un tweet tokenizer y los siguientes hiperparámetros: C = 10, penalty = l2, solver=saga.

   Ya que no hubo mayor preprocesamiento de datos de entrada, no fue necesario incluir ningún otro tipo de paso anterior al vectorizer (el archivo con la construcción del pipeline se puede encontrar en logistic_regression_pipeline.py).

```
pipeline = Pipeline(
    [
        ('tfidf', TfidfVectorizer(tokenizer=tokenizer, stop_words = stop_words, lowercase = True)),
        ('model', LogisticRegression(random_state = 1, C=10, penalty='l2', solver='saga', max_iter=1000, n_jobs = -1))
    ]
)
```

Imagen 1. Construcción del pipeline

```
# Función para tokenizar los comentarios
def tokenizer(text):
    tt = TweetTokenizer()
    tokens = tt.tokenize(text)
    return tokens
```

Imagen 2. Tokenizer utilizado

Persistencia y serialización del modelo:
Para facilitar el proceso de predicción de la aplicación, se decidió persistir el pipeline anterior ya entrenado en forma de joblib, de esta manera no es necesario reentrenar la regresión y hace más fácil el trabajo de carga de datos para el API. Para obtener este archivo binario, se ejecutó el archivo de la regresión logística.

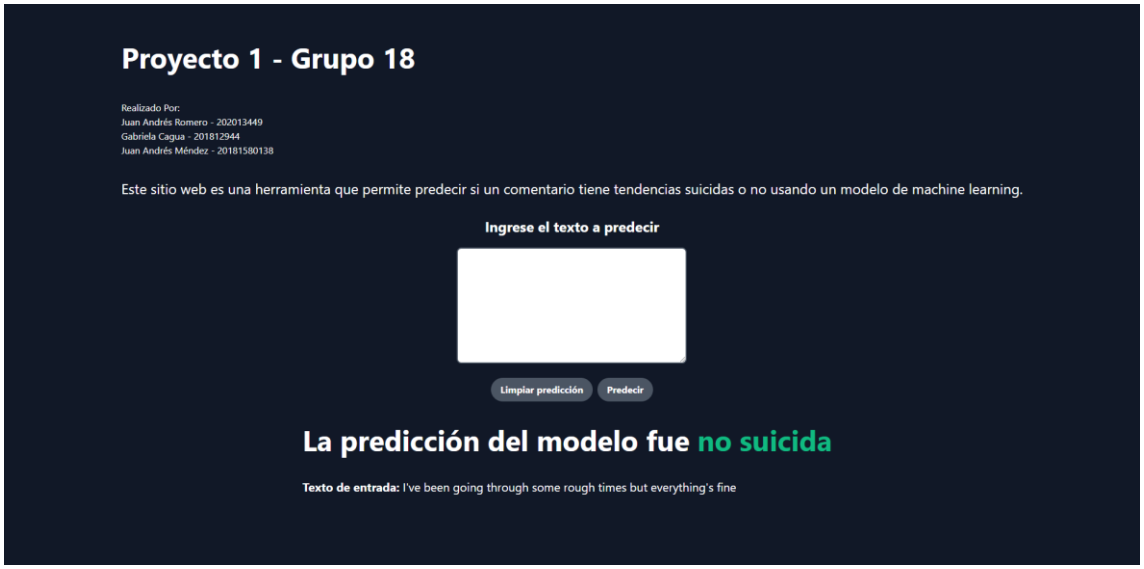
3. **Desarrollo de la aplicación y justificación**
   Sobre el negocio, se tiene pensado que hay dos tipos de organizaciones que pueden utilizar la aplicación y sacar provecho de ella:

   - Consultorios psicológicos: Diferentes tipos de psicólogos pueden utilizar esta herramienta de machine learning para predecir si una persona presenta ideación suicida y de esta manera ofrecerles tratamientos que se ajusten a la situación de cada paciente. Es importante el papel que juega la aplicación dentro del contexto de negocio, ya que, de la mano del conocimiento y la experiencia de los psicólogos, además de una base de datos robusta y clasificada con respecto a este tema, pueden predecir más precisamente la situación de cada paciente y llegar a salvar vidas que de otra forma no se podría hacer.
   - Universidades: Dado que reddit es una red social cuyos usuarios son, en mayor medida jóvenes, al igual que los estudiantes universitarios, el modelo predictivo de machine learning podría ser utilizado por psicólogos que prestan sus servicios en universidades para detectar el nivel de salud mental en los estudiantes y ofrecerles alternativas de terapia y tratamientos para mejorar la calidad de salud mental en cada paciente. Al igual que con los consultorios psicológicos, esta herramienta le sirve al negocio para detectar a temprana edad diferentes comportamientos suicidas y de esta manera poder tratarlos o remitirlos a sus tratamientos correspondientes.

Para cumplir con lo anterior, se desarrolló una pequeña página web que permite a un usuario interactuar con el modelo entrenado y de esta manera recibir predicciones sobre la entrada del mismo. No es necesario ningún tipo de registro y cualquiera puede usarla, basta con solo introducir el comentario a analizar en la caja de texto y hacer click en predecir para obtener resultados.

Ejemplo aplicación desarrollada:

Predicción no suicida



Predicción suicida