

PRE ENTREGA 2

- **Nombre:** Juan Ramognino
- **Curso:** IA
- **Nombre del proyecto:** Gestión del Burnout en Deportistas a través de Generación de Prompts

Introducción

- **Nombre del proyecto:** Gestión del Burnout en Deportistas
- **Presentación del problema a abordar:** El burnout en deportistas afecta tanto el rendimiento como la salud mental de los atletas. Es importante desarrollar una solución para abordar este problema y mejorar el bienestar de los deportistas.

Desarrollo de la propuesta de solución

La propuesta consiste en utilizar técnicas de generación de prompts para ayudar a los deportistas a gestionar el burnout. Los prompts serán diseñados para fomentar la reflexión y el autoanálisis, y se implementarán en un entorno de Jupyter Notebook.

Los prompts incluirán:

1. **Texto-texto:** Preguntas sobre experiencias personales con el burnout, como "¿Cuáles son los principales factores que contribuyen a tu estrés?" y "¿Cómo te sientes después de una competencia?".
2. **Texto-imagen:** Generación de imágenes que representen las metas y estrategias de afrontamiento de los deportistas, como gráficos que muestren el progreso y el bienestar emocional.

Justificación de la viabilidad del proyecto

El uso de la API de OpenAI para generar prompts es técnicamente viable y permitirá obtener respuestas personalizadas para cada deportista. El tiempo requerido para desarrollar la prueba de concepto se estima en 4 semanas, lo que permite ajustes y optimizaciones a lo largo del proceso.

Objetivos

- Demostrar la comprensión de los principios y técnicas detrás de Fast Prompting.
- Experimentar con diferentes configuraciones de prompts para optimizar la eficacia.
- Preparar una demostración efectiva en el Jupyter Notebook para mostrar el funcionamiento de la POC.

Metodología

1. **Investigación:** Recopilar información sobre el burnout en deportistas y las técnicas de Fast Prompting.
2. **Desarrollo de prompts:** Crear una serie de prompts efectivos y relevantes para abordar el problema.
3. **Implementación en Jupyter Notebook:** Utilizar Jupyter Notebook para mostrar cómo los prompts generan respuestas y ayudar a los deportistas a reflexionar sobre su situación.

Herramientas y tecnologías

- API de OpenAI para la generación de prompts.
- Jupyter Notebook para la implementación y demostración de la prueba de concepto.
- Herramientas de generación de imágenes como Nightcafe para visualizar las metas y estrategias de afrontamiento.

Implementación

La implementación del proyecto se llevará a cabo en varias etapas, utilizando Jupyter Notebook como entorno principal para la prueba de concepto (POC). A continuación, se detallan los pasos y el código necesario para llevar a cabo la generación de prompts y su aplicación en el manejo del burnout en deportistas.

1. Configuración del Entorno

Se instalarán las bibliotecas necesarias para interactuar con la API de OpenAI y para realizar análisis de datos y visualización.

```
pip install openai pandas matplotlib
```

2. Conexión a la API de OpenAI

Se establecerá la conexión con la API utilizando la clave de acceso. Esto permitirá la generación de prompts de forma dinámica.

```
import openai
```

```
# Configuración de la API  
openai.api_key = 'tu_api_key'
```

3. Desarrollo de Prompts de Texto

Se crearán prompts específicos para que los deportistas puedan reflexionar sobre su experiencia con el burnout. Por ejemplo, se pueden utilizar preguntas que inviten a la introspección y autoanálisis.

```
def generar_prompt_texto():
    prompt_texto = "Describe cómo te sientes después de una competencia y qué factores contribuyen a tu estrés."
    return prompt_texto

# Ejemplo de uso
prompt = generar_prompt_texto()
```

4. Generación de Respuestas con el Modelo de IA

Se desarrollará una función que enviará el prompt al modelo de OpenAI y retornará una respuesta.

```
def obtener_respuesta(prompt):
    response = openai.ChatCompletion.create(
        model="gpt-3.5-turbo",
        messages=[
            {"role": "system", "content": "Actúa como un entrenador personal que ayuda a los deportistas a manejar el burnout."},
            {"role": "user", "content": prompt}
        ]
    )
    return response.choices[0].message['content']

# Ejemplo de uso
respuesta = obtener_respuesta(prompt)
print(respuesta)
```

5. Visualización de Datos

Se incluirá un análisis de los datos obtenidos de las respuestas de los deportistas, lo que permitirá identificar patrones o áreas comunes de estrés. Utilizando bibliotecas como `matplotlib` o `pandas`, se pueden crear gráficos para visualizar la información.

```
import pandas as pd
import matplotlib.pyplot as plt

# Ejemplo de datos ficticios
datos = {
    'Deportista': ['Atleta 1', 'Atleta 2', 'Atleta 3'],
    'Nivel de Estrés': [5, 7, 4], # Escala del 1 al 10
}
```

```
'Síntomas': ['Ansiedad', 'Fatiga', 'Desmotivación']  
}
```

```
df = pd.DataFrame(datos)
```

```
# Gráfico de barras del nivel de estrés  
plt.bar(df['Deportista'], df['Nivel de Estrés'])  
plt.title('Niveles de Estrés en Deportistas')  
plt.xlabel('Deportista')  
plt.ylabel('Nivel de Estrés (1-10)')  
plt.show()
```

6. Optimización de Consultas a la API

Para garantizar la rentabilidad del proyecto, se analizará la cantidad de consultas que se realizan a la API. Se buscará reducir el número de consultas mediante la agrupación de prompts o el uso de preguntas abiertas que generen respuestas más completas. Se puede implementar un sistema de caché que almacene respuestas frecuentes, evitando así repetir consultas idénticas.

7. Documentación del Proceso

A lo largo del desarrollo, se mantendrá una documentación clara que describa cada parte del proceso. Esto facilitará futuras mejoras y permitirá a otros entender cómo interactuar con el sistema.

8. Ejemplo de Prompt para Visualización

Para la generación de imágenes que representen visualmente las metas del deportista, se utilizarán prompts adecuados para el modelo de generación de imágenes. Por ejemplo:

```
def generar_prompt_imagen():  
    prompt_imagen = "Crea una visualización de un deportista alcanzando su meta en un evento deportivo, reflejando la satisfacción y el bienestar emocional."  
    return prompt_imagen  
  
prompt_img = generar_prompt_imagen()
```