



MEMORIA INTELEGENCIA ARTIFICIAL

Desarrollo de un chatGPT con datos propios

[Siguiendo el siguiente paso en la inteligencia artificial](#)

Investigación y desarrollo de las nuevas tecnologías enfocadas a la inteligencia artificial y el sector público o privado.

Gemma Villarroel Fernández
Gvillf00@estudiantes.unileon.es

Índice

Introducción	1
Breve descripción de mi objetivo	2
Herramientas utilizadas	3
Desarrollos probados	5
1. Desarrollo con K-means	5
2. Tecnología Langchain y HugginFace	6
3. Tecnología Google Translator - Davinci	7
4. Motor gpt 3.5 turbo	7
5. Motor Davinci Chat Sencillo	8
6. Mi propio ChatBot a través de tecnología LangChain	8
7. Uso del modelo chatGPT4	9
8. Intento con Llama-Index.....	10
Proyecto definitivo con Chat GPT	13
ANÁLISIS DE RESULTADOS	20
DAFO.....	21
Lecciones aprendidas	22
REFERENCIAS:.....	23

Introducción

Esta asignatura nos ha servido para profundizar de lleno en los avances tecnológicos que se desarrollan en torno a la inteligencia artificial.

A través de múltiples y diversas lecturas, libros, artículos, vídeos e incluso mensajes de texto cortos (twit) hemos profundizado en el aprendizaje de los LLM (Grandes modelos de Lenguaje) y hemos visto su evolución a lo largo de cortos periodos de tiempo.

Al inicio de esta asignatura, como digo, la lectura ha sido mi principal aliada, empapándome de conceptos hasta ahora desconocidos para mi.

Dentro de la herramienta OSF pueden ver los artículos que he ido leyendo dentro de este aprendizaje no automático

Casi 40 entradas, para ser exactos, 38 entradas de conocimiento, opiniones y debate conmigo misma he mantenido en el OSF, con focos tan diversos y a la vez con un punto en común, la inteligencia artificial siempre como nexo de todo este aprendizaje.

Me he empapado de conceptos sobre el afinamiento de los modelos de aprendizaje, sobre las nuevas herramientas como LlamaIndex con los LLM, como entrenar con datos propios y entrenar a mi chatGPT con datos personalizados.

Uno de los temas que más me ha impresionado (por desconocimiento probablemente) tras la lectura de todo el material al que hemos tenido acceso (y al que no, porque la red se actualiza al minuto con nuevos artículos y nuevo contenido sobre estos temas) es el nivel de “alucinamiento” que tienen estos sistemas, y además lo he podido comprobar de primera mano (lo explicaré más adelante), desde luego no pensé, antes de profundizar en esta materia, que podría ser así, que los LLM no son para nada fiables, y gracias a eso hemos profundizado también en la tecnología RAG

También he tenido la oportunidad de conocer un entorno desconocido hasta el momento, las bases de datos de grafos y como interactuar con ellas en los desarrollos de inteligencia artificial.

No solo de este tema me he documentado, pues para poder desarrollar cualquier software de este tipo se hace necesario conocer el lenguaje de programación de python que es todo un mundo, sobre todo en lo que tiene que ver con sus versiones y la limitación de las mismas con según lo que quieras llevar a cabo.

Mi proyecto a tomado distintas derivas a lo largo de las semanas, en parte propiciadas por nuevos temas que iban surgiendo y que parecían interesantes, claro, que con respecto a la inteligencia artificial ¿Qué no es interesante? Lo es absolutamente todo.

En esta memoria intentaré recoger las vías de investigación y desarrollo que he ido realizando, solo una de ellas ha tenido un resultado tangible que mostrar, pero con todas y cada una he aprendido multitud de conceptos, de herramientas, y de productos que se pueden llevar a cabo.

Breve descripción de mi objetivo

Durante todo el proceso de aprendizaje he tenido claro mi objetivo, o dicho de otra manera, para qué podría ser útil realizar un desarrollo de estas características.

Sin duda, los ámbitos de aplicación son muchos y muy variados, por no decir, que son todos, pues en cualquier área tiene cabida la inteligencia artificial, de hecho en clase hemos visto ejemplos muy llamativos de aplicación y en el OSF pueden encontrar comentarios de artículos que he leído como por ejemplo como aplicar la inteligencia artificial al sector de la logística o de la educación.

Para mi, personalmente, creo que tiene mucho recorrido en el ámbito de la Administración pública, pues considero que tendría un doble beneficio, el de optimizar la gestión propiamente y que el beneficio sería para toda la población, pues a más efectividad en la gestión menor coste de la misma y por tanto menor recaudación de impuestos para sostenerla (dicho así, parece idílico ¿no crees?)

En definitiva, mi objetivo consiste en crear un chatGPT con datos propios, en este caso datos sobre población, haciendo hincapié en conocer la dirección y el barrio a la que pertenece, la edad de cada persona y el sexo, con el fin de buscar los mejores emplazamientos para instalaciones, realización de actividades, nuevas construcciones, etc... más bien un emplazamiento indiscutible basado en datos reales y no en opiniones o argumentos subjetivos, la mayor parte de veces insolidarios, o interesados.

Los datos, son datos, y son incuestionables e incontestables, acabando así con las especulaciones políticas sobre el ámbito que estoy comentando.

Herramientas utilizadas

Bases de datos

Lo primero que necesitamos son datos, y al tratarse de la Administración Pública, a lo primero que recurrí es a ella, en nuestro caso y por sentimiento de pertenencia busqué en el Ayuntamiento de León y la Diputación provincial de León, como era de esperar, no encontré absolutamente ninguna información en datos abiertos, entiendo que los censos están, pero son datos privados y en ningún caso se hace mención a ellos en su web.

El siguiente paso ha sido buscar, también por pertenencia, los datos en la Junta de Castilla y León, al contrario que las entidades locales, la junta si proporciona un portal de datos abiertos, con mucha información, y he de decir que información muy relevante e interesante para futuros proyectos, desgraciadamente para este que me ocupa no encontré los datos que necesitaba, pero hay muchos interesantes con respecto a la vivienda, a los movimientos migratorios, a la agricultura, y un sinfín de parámetros a los que volvería acudir para aplicar este proyecto en otros ámbitos.

Por último, y por quemar todos los cartuchos, busqué el tipo de datos que necesitaba en el INE, que he de decir que es bastante completo, pero nuevamente si encuentras una cosa que te pueda valer le falta algún campo que se hace imprescindible.

Finalmente me decidí a buscar en Kaggle, una grandiosa herramienta que sirve todo tipo de bases de datos en diferentes formatos, algo encontré como el censo en una importante capital sur americana, sin embargo, al descargarlo no fui capaz de abrir el archivo con ningún software ni máquina, el resultado siempre era “archivo dañado”.



Al final, con no mucha esperanza, me decidí a buscar en el portal de datos abiertos de Vitoria, ya que el País Vasco siempre es un referente nacional en los temas de datos abiertos, y efectivamente, lo es por algo.

En su portal encontré en formato csv un archivo sobre el censo por barrios con el sexo y edad de las personas, es cierto que está limitado a los barrios (por lo que no hay muchos registros) y para cumplir mi objetivo de manera más específica me venía mucho mejor contar con todas las calles y que uno de los campos de agrupamiento fuera el barrio, pero esta información me ha servido para trabajar sobre mi proyecto, aunque le he tenido que realizar varias modificaciones como veremos más adelante.

Para los desarrollos he utilizado Visual Studio Code y Python principalmente.



Sobre Python he tenido que hacer un curso express para, por lo menos, comprender el código.

He de decir que los problemas con Python y las APIs de openAI han sido mi desesperación, puesto que las versiones de Python tienen grandes incompatibilidades con los desarrollos de las API, y ese ha sido mi gran problema.

Desarrollos probados

1. Desarrollo con K-means

Encuentro por la red un programa sencillo se acerca mucho a lo que busco.

Es una herramienta que me permite según un array de barrios, el número de niños de cada barrio y la localización del barrio me da como resultado el centro de los klusters de los barrios con más niños, si esto consigo que funcione lo puedo extrapolar para personas mayores, jóvenes o cualquier franja de edad.

El sistema funciona, aún no he conseguido que funciones asociándole directamente un fichero, sí con los arrays, pero claro, utiliza herramientas de aprendizaje automático que no es el objetivo de esta asignatura, utilizamos el entrenamiento K-means.

Pongo un pantallazo del ejemplo que conseguí que funcionara:

```
# -*- coding: utf-8 -*-

import pandas as pd
from sklearn.cluster import KMeans
import matplotlib.pyplot as plt

# variables que ponemos por defecto para que no salgan advertencias
# super()._check_params_vs_input(default_n_init=10)

OMP_NUM_THREADS=1

##### df = pd.read_csv('D:/GEMMA/UNIVERSIDAD/PruebaSIST.csv')

data = {'barrio': ['A', 'B', 'C', 'D'], 'ninos': [100, 150, 50, 200], 'lat': [37.7749, 37.7750, 37.7751, 37.7752], 'lon': [-122.4194, -122.4195, -122.4196, -122.4197]}
df = pd.DataFrame(data)

# Usando K-Means para encontrar el centro de los clusters de barrios con más niños
##### kmeans = KMeans(n_clusters=1, random_state=0).fit(df[['NOMBRE_BARRIO']], sample_weight=df['< 12'])
kmeans = KMeans(n_clusters=1, random_state=0).fit(df[['lat', 'lon']], sample_weight=df['ninos'])

# Graficando los resultados
##### plt.scatter(df['NOMBRE_BARRIO'], s=df['< 12'])
plt.scatter(df['lat'], df['lon'], s=df['ninos'])
plt.scatter(kmeans.cluster_centers_[0], kmeans.cluster_centers_[1], c='red')
plt.show()
```

Sin embargo, con el fichero directamente no me ha funcionado.

Abandono esta línea porque como digo es un proyecto más de aprendizaje automático y mi objetivo es que el chatGPT se alimente de mi base de datos.

2. Tecnología Langchain y HuggingFace

hacemos correr con la sentencia run la ciudad que introduce como variable en la plantilla para que el llm responda como sería un día perfecto. Utiliza el modelo gpt-neo-2.7B

```

> UNIVERSIDAD > SISTEMAS DE INFORMACIÓN DE GESTION SIST > Mi practica > otra_prueba3.py > ...
1  from langchain.prompts import PromptTemplate
2  from langchain.llms import HuggingFace
3  from langchain.chains import LLMChain
4
5  prompt = PromptTemplate(
6      input_variables=["city"],
7      template="¿Cómo sería un día perfecto en {city}?",
8  )
9
10 llm = HuggingFace(
11     model_name="gpt-neo-2.7B",
12     temperature=0.9)
13
14 llmchain = LLMChain(llm=llm, prompt=prompt)
15 llmchain.run("París")

```

3. Tecnología Google Translator- Davinci

Con tecnología openai y los servicios web que pone a disposición Google translator, utilizando la máquina “davinci” y una vez introducida la API de openAi, se le introduce una pregunta a modo de ejemplo y el sistema te devuelve un máximo de 30 respuestas

```

F: > UNIVERSIDAD > SISTEMAS DE INFORMACIÓN DE GESTION SIST > Mi practica > otra_prueba.py > ...
1  import os
2  import openai
3  import googletrans
4  key = "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"
5  translator = googletrans.Translator()
6  sample_text_it = "¿Cuanto tiempo se necesita para aprender Aleman?"
7  translated = translator.translate(sample_text_it, src='es', dest='en')
8  openai.api_key = key
9  response = openai.Completion.create(
10     engine="davinci",
11     prompt=translated.text,
12     temperature=0.25,
13     max_tokens=30,
14     top_p=1
15 )
16 print(response["choices"])

```

4. Motor gpt 3.5 turbo

Esta versión no he conseguido hacerla funcionar, probablemente por la versión de Python que ahora tengo instalada, el problema es que cuando actualizo me dejan de funcionar otros procesos.

```
F: > UNIVERSIDAD > SISTEMAS DE INFORMACIÓN DE GESTION SIST > Mi practica > otra_prueba2_funciona.py > get_completion
1  import openai
2  import os
3
4  from dotenv import load_dotenv, find_dotenv
5  _ = load_dotenv(find_dotenv())
6
7  openai.api_key = os.getenv('OPENAI_API_KEY')
8
9  def get_completion(prompt, model="gpt-3.5-turbo"):
10     messages = [{"role": "user", "content": prompt}]
11     response = openai.ChatCompletion.create(
12         model=model,
13         messages=messages,
14         temperature=0, # this is the degree of randomness of the model's output
15     )
16     return response.choices[0].message["content"]
```

5. Motor Davinci Chat Sencillo

Con el motor davinci realicé otro ejemplo muy sencillo, como prompt se le introduce Hola y se muestra en pantalla la respuesta del chatbot.

```
F: > UNIVERSIDAD > SISTEMAS DE INFORMACIÓN DE GESTION SIST > Mi practica > prueba_chat.py > ...
1  import openai
2
3  # Establece las credenciales de OpenAI API
4  openai.api_key = "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"
5
6  # Envía el prompt(mensaje de usuario) y la entrada del usuario a la API
7  response = openai.Completion.create(
8      engine='davinci',
9      prompt="Hola.",
10     temperature=1.0,
11     max_tokens=100,
12     stop="\n"
13 )
14
15 # Obtiene la respuesta del chatbot desde la API de OpenAI
16 chatbot_response = response.choices[0].text.strip()
17
18 # Imprime la respuesta del chatbot
19 print(chatbot_response)
```

6. Mi propio ChatBot a través de tecnología LangChain

Investigando por la red encontré varios ejemplos de cómo llevar a cabo mi propio chatbot a través de la tecnología langchain, conseguí desarrollar un archivo de prueba en python que no he conseguido hacer funcionar, pues influyen mucho las versiones que utilices de python para este tipo de tecnologías, claro que, eso lo he descubierto hace poco.

Comencé con la última versión de python por lógica y tenía muchísimos problemas con todas las librerías que necesitan los LLM, todas me daban errores a la hora de importar, hasta que descubrí que debía utilizar una versión de python anterior, con la versión 3.10 he conseguido eliminar varios errores.

Otro de los problemas que me ha surgido es que muchos de los ejemplos se han implementado con código obsoleto (bueno de hace un mes) y que la api que da el servicio de chatGPT ha cambiado.

La cuestión es que encontré como poder hacer mi chat con la tecnología langchain, no conseguí que funcionara, adjunto pantallazo:

```
app.py 1 X
D: > GEMMA > UNIVERSIDAD > SISTEMAS DE INFORMACIÓN DE GESTIÓN SIST > Mi practica > ChatGPT_misdatos > docs > app.py > ...
1 from gpt_index import SimpleDirectoryReader, GPTListIndex, GPTSimpleVectorIndex, LLMPredictor, PromptHelper
2 from langchain import openai
3 import gradio as gr
4 import sys
5 import os
6
7 openai.api_key="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"
8 #os.environ["OPENAI_API_KEY"] = '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'
9
10 def construct_index(directory_path):
11     max_input_size = 4096
12     num_outputs = 512
13     max_chunk_overlap = 20
14     chunk_size_limit = 600
15
16     prompt_helper = PromptHelper(max_input_size, num_outputs, max_chunk_overlap, chunk_size_limit=chunk_size_limit)
17
18     llm_predictor = LLMPredictor(llm=openai(temperature=0.7, model_name="text-davinci-003", max_tokens=num_outputs))
19
20     documents = SimpleDirectoryReader(directory_path).load_data()
21
22     index = GPTSimpleVectorIndex(documents, llm_predictor=llm_predictor, prompt_helper=prompt_helper)
23
24     index.save_to_disk('index.json')
25
26     return index
27
28 def chatbot(input_text):
29     index = GPTSimpleVectorIndex.load_from_disk('index.json')
30     response = index.query(input_text, response_mode="compact")
31     return response.response
32
33 iface = gr.Interface(fn=chatbot,
34                       inputs=gr.inputs.Textbox(lines=7, label="Introduce tu texto"),
35                       outputs="text",
36                       title="Mi AI Chatbot")
```

abandoné esa vía para intentarlo con llama con el contenido que nos facilitó el profesor.

7. Uso del modelo chatGPT4

```

F: > UNIVERSIDAD > SISTEMAS DE INFORMACIÓN DE GESTIÓN SIST > Mi practica > api_chatgpt.py > ...
1
2 from openai import OpenAI
3
4 client = OpenAI(
5     api_key=os.environ['sk-jNajGbuBF2eHsiP6YhgdT3B1bkFJwZULz5yn60FSBAOrsLt6'], # this is also the default, it can be omitted
6 )
7
8 completion = openai.ChatCompletion.create(
9     model="gpt-4",
10    messages=[
11        {"roles": "system", "content": "Eres un asistente de ayuda"},
12        {"roles": "user", "content": "¿Puedes decirme algo que inspire a la audiencia del microsoft build 2023?"},
13    ]
14)
15 print(completion.choices[0].message.content)
16 response=completion
17 print(response["choices"][0]["message"]["content"])

```

Utilizando el modelo chatGPT 4

Junto con la API se introducen los prompts que queramos y muestra por pantalla los mensajes del chatbot.

Me ha dado error por la versión de Python, esto es muy común y por lo visto confronta muy a menudo con los sistemas y librerías de inteligencia artificial, muchos módulos solo están adaptados a una versión concreta de python y eso dificulta mucho los desarrollos.

8. Intento con Llama-Index

Siguiendo el siguiente tutorial:

<https://github.com/run-llama/LlamaIndexTS/tree/main/packages/create-llama>

```
npm run generate

Using chat engine: context

Installing dependencies:
- ai
- llamaindex
- next
- react
- react-dom

Installing devDependencies:
- @types/node
- @types/react
- @types/react-dom
- autoprefixer
- eslint
- eslint-config-next
- postcss
- tailwindcss
- typescript

npm WARN deprecated fs-promise@2.0.3: Use mz or fs-extra^3.0 with Promise Support

added 465 packages, and audited 466 packages in 3m

120 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
Created '.env.local' file containing OPENAI_API_KEY

Copying test data to C:\Users\Gemma\chat_gemma\data

Running npm run generate to generate the context data.

> chat_gemma@0.1.0 generate
> node app\api\chat\engine\generate.mjs

Generating storage context...
(node:12856) [DEP0040] DeprecationWarning: The `punycode` module is deprecated. Please use a userland alternative instead.
(Use `node --trace-deprecation ...` to show where the warning was created)
No valid data found at path: cache\doc_store.json starting new store.
No valid data found at path: cache\index_store.json starting new store.
No valid data found at path: ./cache/vector_store.json starting new store.
```

Siguiendo los pasos recién publicados por los creadores de llama index he sido capaz de llegar a instalar un chatbot en la unidad local, como se muestra en la imagen.

```
Storage context successfully generated in 105.664s.
Finished generating storage.

Success! Created chat_gemma at C:\Users\Gemma\chat_gemma
Now have a look at the README.md ( file:///chat_gemma/README.md ) and learn how to get started.

C:\Users\Gemma>
```

En la ruta local C:\Users\Gemma\chat_gemma me crea el directorio de funcionamiento de la aplicación, una de las carpetas es la carpeta data donde debemos meter los archivos de datos de los que queremos se nutra nuestro propio chatGPT

```

C:\Users\Gemma>npm run
C:\Users\Gemma>npm install
up to date, audited 32 packages in 799ms
1 package is looking for funding
  run `npm fund` for details
found 0 vulnerabilities
C:\Users\Gemma>node -v
v21.1.0
C:\Users\Gemma>cd prueba-3.chat-gemma
El sistema no puede encontrar la ruta especificada.
C:\Users\Gemma>cd prueba-3-chat-gemma
C:\Users\Gemma\prueba-3-chat-gemma>npm install
up to date, audited 580 packages in 1s
143 packages are looking for funding
  run `npm fund` for details
found 0 vulnerabilities
C:\Users\Gemma\prueba-3-chat-gemma>npm run dev
> prueba-3-chat-gemma@0.1.0 dev
> next dev

   ▲ Next.js 14.0.4
   - Local:      http://localhost:3000

Attention: Next.js now collects completely anonymous telemetry regarding usage.
This information is used to shape Next.js' roadmap and prioritize features.
You can learn more, including how to opt-out if you'd not like to participate in this anonymous program, by visiting the following URL:
https://nextjs.org/telemetry

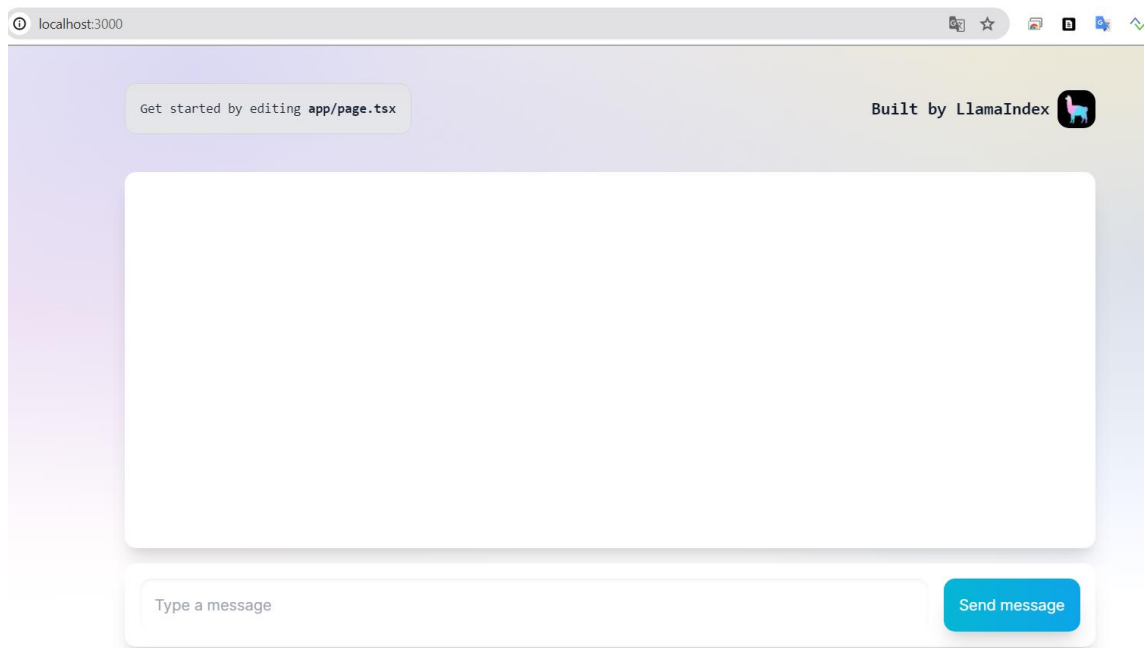
✓ Ready in 3.4s
(node:14800) [DEP0040] DeprecationWarning: The `punycode` module is deprecated. Please use a userland alternative instead.
(Use `node --trace-deprecation ...` to show where the warning was created)
o Compiling / ...
✓ Compiled / in 3.3s (538 modules)
✓ Compiled in 380ms (255 modules)

```

```

✓ Ready in 3.4s
(node:14800) [DEP0040] DeprecationWarning: The `punycode` module is deprecated. Please use a userland alternative instead.
(Use `node --trace-deprecation ...` to show where the warning was created)
o Compiling / ...
✓ Compiled / in 3.3s (538 modules)
✓ Compiled in 380ms (255 modules)
o Compiling /favicon.ico ...
✓ Compiled /favicon.ico in 1208ms (543 modules)
✓ Compiled /api/chat in 218ms (302 modules)
[llamaIndex] Error: Set OpenAI Key in OPENAI_API_KEY env variable
    at new OpenAISession (C:\Users\Gemma\prueba-3-chat-gemma\node_modules\llamaindex\dist\index.js:2363:13)
    at getOpenAISession (C:\Users\Gemma\prueba-3-chat-gemma\node_modules\llamaindex\dist\index.js:2379:15)
    at new OpenAI2 (C:\Users\Gemma\prueba-3-chat-gemma\node_modules\llamaindex\dist\index.js:2655:81)
    at POST (webpack-internal:///./rsc)/./app/api/chat/route.ts:49:21
    at process.processTicksAndRejections (node:internal/process/task_queues:95:5)
    at async C:\Users\Gemma\prueba-3-chat-gemma\node_modules\next\dist\compiled\next-server\app-route.runtime.dev.js:6:63251
✓ Compiled in 256ms (557 modules)
✓ Compiled /favicon.ico in 58ms (328 modules)
[llamaIndex] Error: Set OpenAI Key in OPENAI_API_KEY env variable
    at new OpenAISession (C:\Users\Gemma\prueba-3-chat-gemma\node_modules\llamaindex\dist\index.js:2363:13)
    at getOpenAISession (C:\Users\Gemma\prueba-3-chat-gemma\node_modules\llamaindex\dist\index.js:2379:15)
    at new OpenAI2 (C:\Users\Gemma\prueba-3-chat-gemma\node_modules\llamaindex\dist\index.js:2655:81)
    at POST (webpack-internal:///./rsc)/./app/api/chat/route.ts:49:21
    at process.processTicksAndRejections (node:internal/process/task_queues:95:5)
    at async C:\Users\Gemma\prueba-3-chat-gemma\node_modules\next\dist\compiled\next-server\app-route.runtime.dev.js:6:63251
✓ Compiled in 312ms (583 modules)

```



Lamentablemente no he conseguido la funcionalidad que quería, he llegado hasta aquí.

Proyecto definitivo con Chat GPT

<https://chat.openai.com/gpts/discovery>

Configuraciones, añadiendo esta configuración permito que chatGPT además de buscar información en las bases de datos o ficheros que le aporte utilice la red y realice acciones fuera de chatGPT.


Añadimos la clave para usar la API

ChatGPT

x +

← → ↻

chat.openai.com/gpts/editor/g-Qkx1tHGaz

< 

Nuevo GPT


● Borrador

<

api.openai.com

Deje que su GPT recupere información o realice acciones fuera de ChatGPT

Aprende más.



Esquema


```
{
  "openapi": "3.1.0",
  "info": {
    "title": "OpenAI Profile",
    "description": "Gets data about the current user's OpenAI profile",
    "version": "v1.0.0"
  },
  "servers": [
    {
      "url": "https://api.openai.com"
    }
  ],
  "paths": {
    "/v1/me": {
      "get": {
        "description": "Gets the user's profile",
        "operationId": "GetProfile",
        "parameters": [],
        "deprecated": false.
      }
    }
  }
}
```

Acciones disponibles

Nombre	Método	Camino
Obtener perfil	CONSEGUIR	/v1/yo

Completamos las opciones de configuración, las que se mostrarán al usuario y lo más importante, añadimos los archivos necesarios sobre los que queremos realizar la consulta, en mi caso, los datos de la ciudad de Vitoria, discriminado por barrios que contempla por rangos de edad los habitantes.

<

 **GPT Gemma**

● Borrador

Crear

Configurar



Nombre

GPT Gemma

Descripción

Experta en recomendación de infraestructura urbana

Instrucciones

Como GPT Gemma, mi función es recomendar la ubicación ideal para el mobiliario urbano y las instalaciones deportivas en las ciudades, basándome en la edad y el sexo de los habitantes de los diferentes barrios. Utilizo datos demográficos y estadísticas para proporcionar recomendaciones precisas y pertinentes, asegurándome de que las infraestructuras propuestas satisfagan las necesidades de los residentes de manera efectiva. Además, tengo la capacidad de analizar archivos de datos, realizar búsquedas en internet y generar imágenes para apoyar mis recomendaciones. Mi enfoque es claro y directo, evitando confusiones y asegurando que mis

Iniciadores de conversación

¿Qué instalaciones recomendarías para un barrio con mayoría de jóvenes?



¿Cómo influye el género en la elección de infraestructuras urbanas?



¿Puedes analizar datos demográficos para sugerir instalaciones?



Crear

Configurar

¿Qué tipo de parques son mejores para adultos mayores?



Conocimiento

Las conversaciones con su GPT pueden incluir contenidos de archivos. Los archivos se pueden descargar cuando el intérprete de código está habilitado.

Los siguientes archivos solo están disponibles para Code Interpreter:



PoblacionXBarrioYEdadY...

hoja de calculo

Subir archivos

Capacidades

- ☒ Buscando en la web
- ☒ Generación de imágenes DALL-E
- ☒ Intérprete de código ?

Comportamiento

api.openai.com



Crear nueva acción

> Ajustes adicionales

Avance



GPT Gemma

Experto en recomendación de infraestructura
urbana

¿Qué instalaciones recomendaría para un barrio con...

¿Puedes analizar datos demográficos para observar...

¿Cómo influye el género en la elección de infraestructu...

¿Qué tipo de parques son mejores para adultos may...



Mensaje GPT Gemma...



Sin embargo, al realizar pruebas el sistema no analiza bien el contenido del csv, según le vas diciendo que el resultado no es correcto, el sistema reconduce y ya sí te aporta una respuesta correcta

Preview



GPT Gemma

Tomando en cuenta los barrios de Zabalgana y Salburua, los datos muestran que Zabalgana tiene la mayor cantidad de niños menores de 12 años, con un total de 6,195 niños. Salburua también tiene una cantidad significativa de 4,176 niños en esta franja de edad. Estos barrios, junto con Arriaga-Lakua, serían las ubicaciones más recomendables para instalar parques para menores de 12 años basándonos en la cantidad de niños residentes.

Aquí está la actualización con estos barrios incluidos:

1. **Zabalgana** - 6,195 niños
2. **Arriaga-Lakua** - 4,289 niños
3. **Salburua** - 4,176 niños
4. **Coronación** - 1,157 niños
5. **Zaramaga** - 1,009 niños

Estos datos indican que Zabalgana sería la ubicación más conveniente, seguida por Arriaga-Lakua y Salburua. [↗]




Message GPT Gemma...



?

Código asociado a mi chatGPT

×

 Copiar código

Resultado

 Copiar código

Resultado

```
('CASCO VIEJO', 860.0)
```

ANÁLISIS DE RESULTADOS

Solo voy a comentar los resultados obtenidos con el sistema que he conseguido hacer funcionar.

Es una herramienta muy potente la que ha pretendido poner en marcha chatGPT, sin embargo, al subir los datos, en mi caso una Excel, algo sencillo, he detectado no sólo que el sistema alucina, ya que no me devuelve resultados coherentes, si no que he descubierto que ni siquiera consulta el archivo en su totalidad, y eso que no es muy grande ni tiene un gran volumen de datos.

La cuestión es que al realizar la consulta sobre el barrio idóneo para la instalación por ejemplo de un parque para menores de 12 años, me muestra un resultado sobre los primeros registros.

¿Cómo lo he solucionado? He tenido que modificar el archivo origen, añadiendo nuevas columnas que identifiquen los porcentajes de población en función de las edades y ahí ya sí la herramienta realiza un desarrollo perfecto, pongo una imagen de ejemplo:

Avance



Procedimiento

Los barrios con los porcentajes más altos de población mayor de 64 años son:

1. **Gazalbide** con un 42% de su población mayor de 64 años.
2. **El Pilar** con un 40%.
3. **Txagorritxu** con un 37%.
4. **Zaramaga** con un 36%.
5. **San Cristobal** con un 34%.

Basándome en estos datos, recomendaría considerar uno de estos barrios, preferentemente Gazalbide o El Pilar, para la instalación de un parque biosaludable destinado a personas mayores. Estas áreas tienen una alta concentración de residentes dentro del grupo de edad objetivo, lo que significa que el parque tendría el potencial de beneficiar a una gran parte de la comunidad local de la tercera edad.

Además, sería importante evaluar otros factores como la accesibilidad del área, la presencia de otras instalaciones para personas mayores en las cercanías y la facilidad para realizar actividades al aire libre, para asegurar que el parque sea lo más beneficioso posible. ` ` 【oacite:0】 ` `​

 Mensaje GPT Gemma... 

?

Como podemos ver se puede ampliar con mucha más información relevante esta herramienta

DAFO

Debilidades:

Los datos se hacen vulnerables en estos sistemas, además tenemos que considerar que son datos propios, de empresas, particulares o, como es mi caso, administraciones públicas. Es un tema delicado, y por tanto debilidad, mantener la privacidad y la propiedad de los datos sobre a los que a su vez se quiere interactuar en lenguaje natural.

La falta de recursos en tiempo, en 3 meses, 4 horas semanales más todas las dedicadas en casa no son suficientes para estudiar y analizar todo el conocimiento que hay entorno a esta temática.

Además, se requieren destrezas de todo tipo para conseguir un proyecto local en funcionamiento

Amenazas:

En mi opinión la más relevante puede ser el propio mercado y los constantes cambios que se realizan entorno a la inteligencia artificial y todo lo que conlleva, los artículos de investigación son ingentes, de la misma manera que los descubrimientos, cada minuto conocemos información nueva sobre los sistemas de inteligencia artificial.

Otra amenaza importante es la falta de regulación / legislación a este respecto, esta tecnología avanza y sin embargo la regulación no avanza al mismo ritmo.

También los cambios en las preferencias de los usuarios suponen una amenaza para este entorno en el que nos movemos

Fortalezas:

A pesar de tiempo insuficiente para el volumen de información y tecnología a estudiar, se ven unas grandes capacidades, en nuestro caso al final del proceso, de muchas pruebas no exitosas en varias tecnologías si se ve claro que el sistema puede funcionar en local con datos propios y ofrecer una experiencia de usuario satisfactoria en entornos cerrados como empresas, administraciones o incluso de uso casero si eres un amante de la organización en hojas de cálculo y quieres interactuar con lenguaje natural con la información de la que eres propietario.

Hoy en día, contamos con todas las herramientas para poder llevarlo a cabo

Oportunidades:

Se abre un horizonte inmenso de oportunidades en el ámbito empresarial, una posible comercialización de un producto que proporciona a la empresa la interacción de sus empleados con todos los formatos de datos procedentes de la empresa de manera natural, como si de un empleado más se tratase.

Además, los mantenimientos y el crecimiento pueden llevarse a cabo sin grandes trastornos

La empresa o ente que llegue a utilizar estos sistemas se mantienen como propietarios de sus datos como no podía ser de otra manera, y a su vez celosos de su privacidad.

Lecciones aprendidas

Desde el comienzo de esta asignatura lo primero que he aprendido es a ser constante en el aprendizaje.

El campo en el que hemos trabajado es un campo en constante evolución, totalmente cambiante, cada minuto prácticamente se publica nuevo contenido al respecto (exagerando un poco), por este motivo la constancia en el aprendizaje ha sido crucial, casi 80 entradas en el OSF lo demuestran y muchas de ellas ha sido imposible dejarlas reflejadas por escrito.

En segundo lugar, he aprendido a aceptar el fracaso como parte de este aprendizaje, muchos de los caminos que he tomado no han tenido éxito, la actitud en este caso es clave, tomarme esos fracasos como la oportunidad para aprender y mejorar, por supuesto sin dejar de analizar que o por qué no ha funcionado.

Quizás he echado en falta algo tan importante como es la colaboración y la comunicación, creo que en un proyecto tan novedoso como son estos en los que estamos embarcados sería de gran ayuda realizar solo un proyecto con la colaboración de todos los estudiantes, y por supuesto el profesor, con una buena tormenta de ideas se puede seleccionar un buen proyecto y gracias al conocimiento colectivo podríamos aprender unos de otros, pero es sólo una opinión.

También creo que he mejorado y aprendido en la gestión de tiempos y recursos, con todo el material que inunda la red sobre estas cuestiones ha sido clave organizar ese contenido, de gran ayuda ha sido el filtro proporcionado por el profesor, y aun así el contenido ha sido ingente, por eso ha sido clave aprender a gestionar de manera eficiente y eficaz el tiempo y los recursos.

Casi parece obvio, pero lo voy a nombrar, otra lección aprendida ha sido la adaptabilidad y flexibilidad que he necesitado durante el semestre pues como ya hemos comentado previamente cada día contábamos con nuevos avances tecnológicos y siempre nos movemos en terrenos muy cambiantes, incluso, se puede decir que he llegado a tener una idea clara que a las dos semanas ha sido reemplazada por otra

completamente diferente. Mucho me temo que en el terreno de la inteligencia artificial esto será el pan nuestro de cada día.

Finalmente creo que es y ha sido muy importante sentir curiosidad por este tema y por qué no, pasión por aprender sobre las nuevas técnicas que cambiarán nuestras vidas, que ya las están cambiando, continuaré con el deseo de seguir aprendiendo y explorando en esta temática.

REFERENCIAS:

Kaggle: <https://www.kaggle.com/datasets>

Phyton: <https://docs.python.org/es/3.12/>

OpenAI: <https://openai.com/blog/openai-api>

LlamaIndex: <https://docs.llamaindex.ai/en/latest/>

<https://docs.llamaindex.ai/en/stable/>

Sciencedirect: <https://www.sciencedirect.com/>

Arxiv: <https://arxiv.org/search/?query=LLM&searchtype=all&source=header>

Administraciones públicas:

<https://www.jcyl.es/sie/v2/padronv2irAmodulo.html>

<https://www.ine.es/>

<https://www.vitoria-gasteiz.org/j34-01w/catalogo/portada?idioma=es&>

GitHub: <https://github.com/openai/openai-python/discussions/742>

Mis accesos:

GitHub: <https://github.com/gvillf00/SIST>

OSF: <https://osf.io/dashboard>