TECNOLÓGICO NACIONAL DE MÉXICO

Documentación del Proyecto: Recomendador de Menú

Alumnos:

- Lara Beltrán Jorge Alberto
- Pacheco Pérez Diego

Docente:

Zuriel Dathan Mora Felix

Materia:

Inteligencia Artificial

Contenido

1. Descripción General 3
2. Instalación y Requisitos 5
3. Uso 6
4. Backend (FastAPI) 8
5. Motor de Recomendación (Red Bayesiana) 9
6. Frontend (Streamlit) 9
7. Guia de uso 9
8. Mejores prácticas y Consideraciones 11

1. Descripción General

Este proyecto consiste en una aplicación web que recomienda platos a los clientes basado en sus preferencias, restricciones dietéticas y disponibilidad de ingredientes.

- **Frontend:** Streamlit, para interacción rápida y visual.
- **Backend:** FastAPI, para procesar solicitudes de recomendación.
- **Motor de recomendación:** Red bayesiana utilizando pgmpy.
- **Entrada del usuario:** Preferencias (vegano, picante), alergias, disponibilidad de ingredientes.
- **Salida:** Plato recomendado, distribución de probabilidad y visualización con imágenes.

Modelo de razonamiento implementado

El proyecto utiliza un modelo de razonamiento probabilístico basado en redes bayesianas para recomendar platos a los usuarios según sus preferencias y restricciones.

Características del modelo:

- **Red Bayesiana (Bayesian Network)**: Representa relaciones de dependencia entre variables clave del sistema:
    - PreferenciaCliente → PlatoRecomendado
    - RestriccionDietetica → PlatoRecomendado
    - DisponibilidadIngredientes → PlatoRecomendado
- **Inferencia probabilística**:
  Se utiliza VariableElimination de pgmpy para calcular la distribución de probabilidad sobre los platos recomendados, dado un conjunto de evidencias (por ejemplo, si el cliente es vegano o le gusta el picante).
- **Salida del modelo**:
  Una **distribución de probabilidad** sobre todos los platos posibles, donde se selecciona el plato con mayor probabilidad como recomendación principal.

- **Ejemplo de evidencia y resultado**:

```
{
  "IsVeganCustomer": 0,
  "LikesSpicy": 1,
  "IngredientsAvailable": 1,
  "AllergicToNuts": 0
}
```


Resultado esperado: Probabilidades más altas para platos picantes y disponibles.

Representación del conocimiento

El conocimiento del sistema está representado mediante variables discretas y CPDs (Conditional Probability Distributions):

- **Variables del modelo**:
    1. PreferenciaCliente: Categorías como Saludable, Italiana, Picante.
    2. RestriccionDietetica: Ninguna, Vegetariana, Vegana.
    3. DisponibilidadIngredientes: Disponible, NoDisponible.
    4. PlatoRecomendado: Todos los platos posibles, por ejemplo: Ensalada César, Pasta Alfredo, Tacos Picantes, etc.
- **CPDs (TabularCPD)**:
  Cada variable tiene una tabla de probabilidad condicional que define cómo la probabilidad de cada estado de la variable depende de los estados de sus variables padre.
    - Ejemplo: La probabilidad de recomendar “Tacos Picantes” aumenta si PreferenciaCliente=Picante y DisponibilidadIngredientes=Disponible.
- **Mecanismo de razonamiento**:
    - Se transforma la entrada del usuario (checklists de Streamlit) en evidencia para la red.
    - La red bayesiana combina las probabilidades condicionales para calcular la probabilidad de cada plato.

    - Se selecciona el plato con la probabilidad más alta como recomendación.

- **Ventajas de esta representación**:
    - Permite capturar relaciones inciertas entre preferencias, restricciones y disponibilidad.
    - Es flexible para agregar nuevos platos, categorías o restricciones sin reescribir toda la lógica.
    - Facilita explicar el porqué de cada recomendación a partir de las probabilidades calculadas.

2. Instalación y Requisitos

1. **Clonar repositorio**

```
git clone <https://github.com/Jorgebelioo/Inteligencia-Artificial.git>

cd Unidad 2/menu-recommender
```

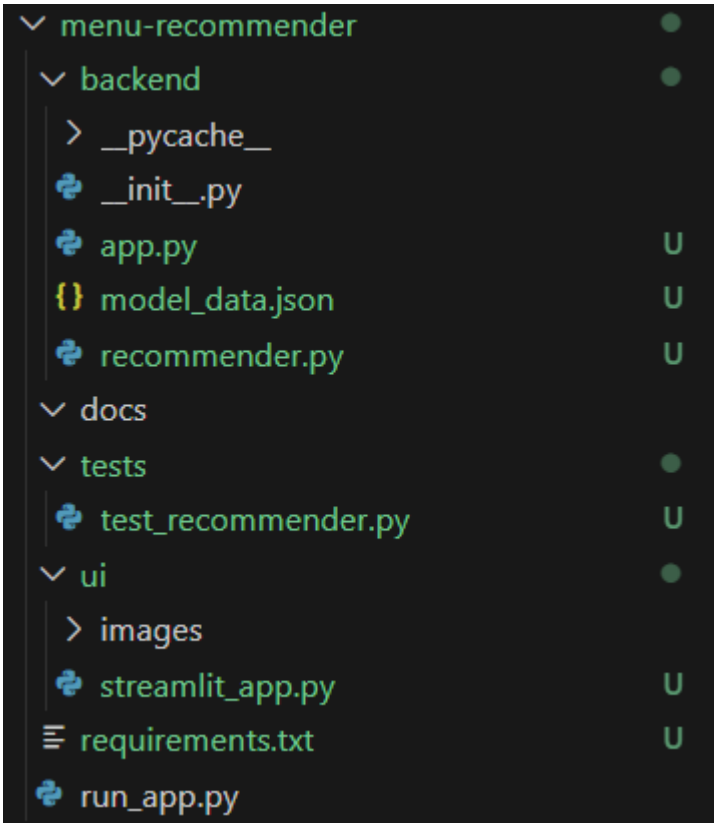
2. **Crear entorno virtual**

```
python -m venv venv

venv\Scripts\activate  # Windows

source venv/bin/activate  # Linux/Mac
```

3. **Instalar dependencias**

```
pip install -r requirements.txt
```

4. **Estructura de carpetas**



3. Uso

Dentro de la carpeta raíz del proyecto (menu-recommender) hacer lo siguiente

3.1 Ejecutar Backend (FastAPI)

python app.py

El backend correrá en http://localhost:8000.

3.2 Ejecutar Frontend (Streamlit)

streamlit run streamlit_app.py

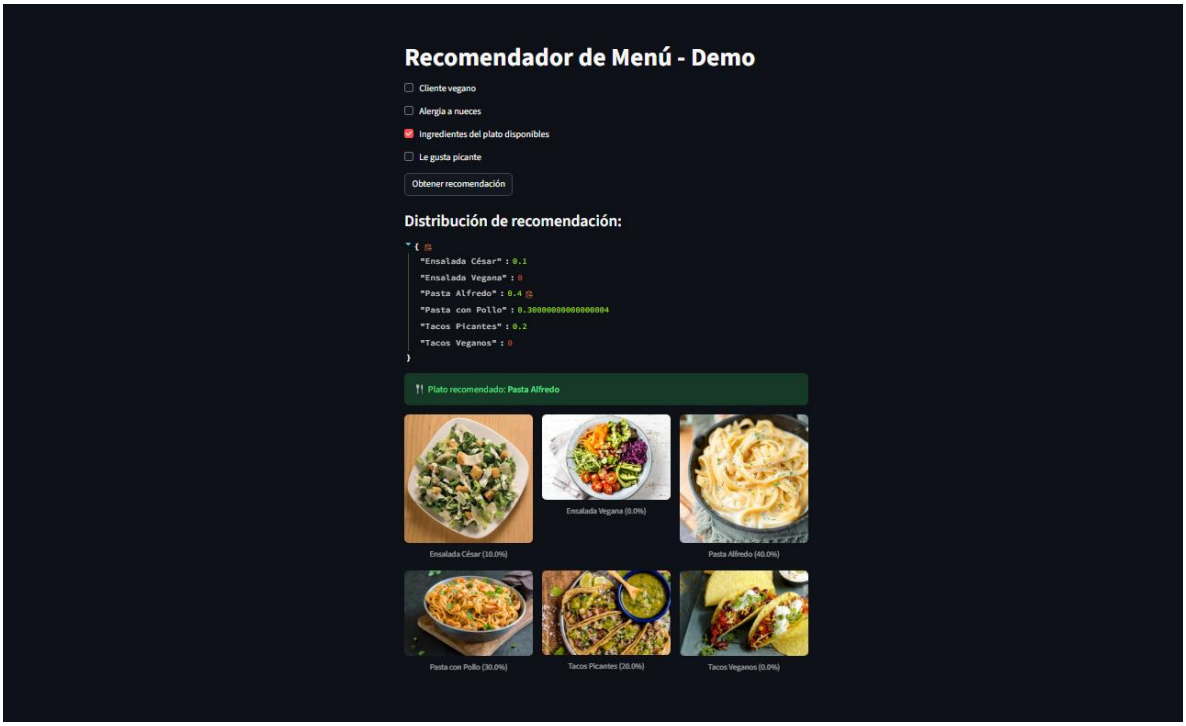
El frontend estará disponible en http://localhost:8501.

3.3 Ambas

python run_app.py

O en su defecto, debido a fallas iniciales del proyecto, hicimos ese .py para poder correr ambas al mismo tiempo sin necesidad de abrir 2 terminales

3.4 Interfaz de Usuario



- Checkboxes para seleccionar preferencias: vegano, picante, alergias, disponibilidad de ingredientes.
- Botón "Obtener recomendación" envía la solicitud al backend.
- Se muestra:
    - Plato recomendado destacado.
    - Distribución de probabilidad de todos los platos.
    - Imágenes de los platos disponibles en un grid compacto.

En el frontend se deben verificar (hacer click) sobre los checkboxes que mas se adecuen a las necesidades del cliente, ya sea si el cliente es vegano se debe seleccionar el checkbox correspondiente y si no hay otra necesidad, entonces se procede a clickear sobre el botón Obtener recomendación para que se ejecute el back y muestre lo dicho anteriormente.

4. Backend (FastAPI)

Endpoint principal:
POST /recommend

Payload de ejemplo:


```
{
  "IsVeganCustomer": 0,
  "AllergicToNuts": 0,
  "IngredientsAvailable": 1,
  "LikesSpicy": 1
}
```


Respuesta de ejemplo:


```
{
  "recommend_distribution": {
              "Ensalada César": 0.2,
              "Ensalada Vegana": 0.1,
              "Pasta Alfredo": 0.25,
              "Pasta con Pollo": 0.15,
              "Tacos Picantes": 0.2,
              "Tacos Veganos": 0.1
            }
}
```

5. Motor de Recomendación (Red Bayesiana)

- Variables:
    - PreferenciaCliente: Saludable, Italiana, Picante
    - RestriccionDietetica: Ninguna, Vegetariana, Vegana
    - DisponibilidadIngredientes: Disponible, NoDisponible
    - PlatoRecomendado: 6 platos posibles
- Se utiliza VariableElimination para calcular la distribución de probabilidad de cada plato.

6. Frontend (Streamlit)

- Presenta la información en un **grid de imágenes**, mostrando solo los platos con imagen.
- El plato recomendado se destaca con un texto y un ícono.
- Las imágenes se ajustan automáticamente al ancho del contenedor (use_container_width=True).

7. Guia de uso

Una vez explicado el proyecto, se debe ejecutar, para esto en la terminal estando en la carpeta:

C:\Inteligencia-Artificial

Se debe ejecutar en la terminal el siguiente comando para generar el entorno virtual:

python -m venv venv

Una vez generado instalamos las dependencias escribiendo el comando en la terminal:
pip install -r requirements.txt

Ya que todas se hayan instalado, procedemos a cambiar de directorio y a ejecutar el proyecto

cd C:\Inteligencia-Artificial\menu-recommender

Y dentro de esta dirección ejecutamos los siguientes comandos

python app.py

El backend correrá en http://localhost:8000.

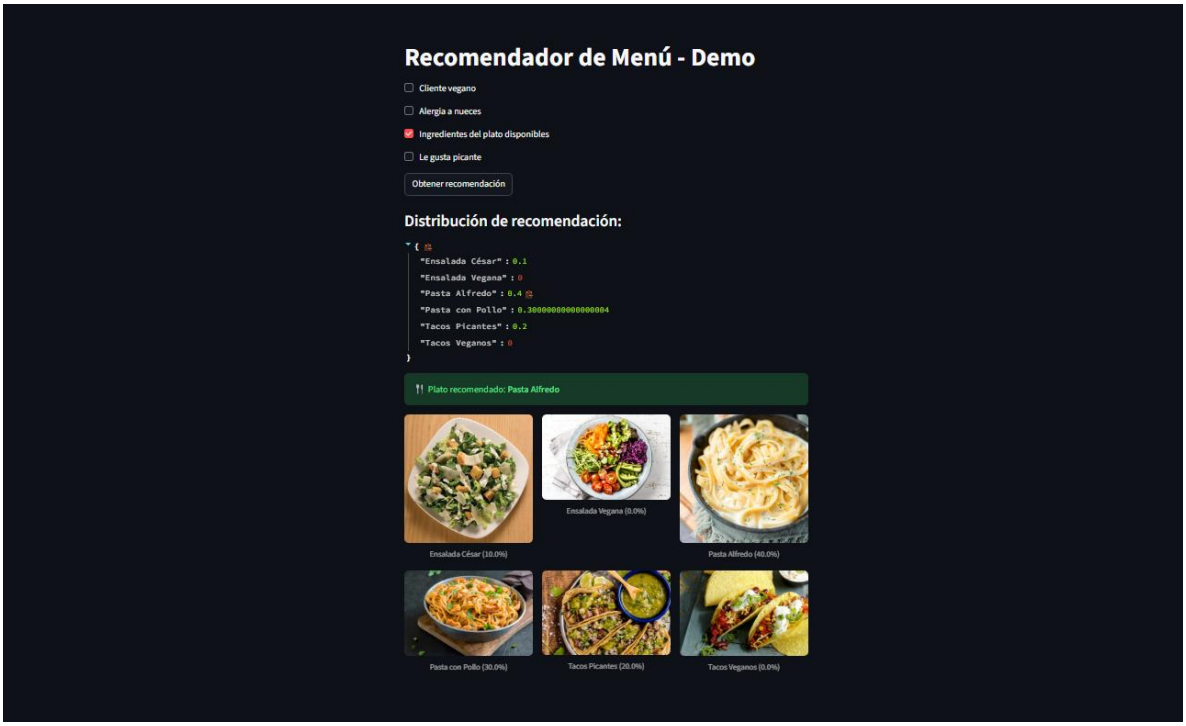
streamlit run streamlit_app.py

El frontend estará disponible en http://localhost:8501.

O si solo se desea usar una terminal, se ejecutaría solamente el siguiente comando

python run_app.py

Una vez que se hayan ejecutado cualquiera de las 2 opciones se abrirá el navegador y mostrará la siguiente página en la cual se deben verificar (hacer click) sobre los checkboxes que más se adecuen a las necesidades del cliente, ya sea si el cliente es vegano se debe seleccionar el checkbox correspondiente y si no hay otra necesidad, entonces se procede a clickear sobre el botón Obtener recomendación para que se ejecute el back y muestre lo dicho anteriormente.



8. Mejores prácticas y Consideraciones

- Normalizar nombres de archivos de imagen (minúsculas, guiones bajos, sin acentos) para que sean consistentes con los nombres de plato.
- Mantener actualizado cpd_plato si se agregan nuevos platos.
- Asegurarse que la carpeta ui/images tenga las imágenes correspondientes.
- Validar inputs en frontend antes de enviar al backend para evitar errores de JSON.