

# Sistema de Recomendação de Produtos

Universidade de Brasília – DCC  
Projeto de IA 2025/1

## Visão Geral

Este projeto conecta pequenos produtores a consumidores de uma região via recomendação colaborativa KNN, filtragem geográfica e preferências.

## Funcionalidades

- Recomendações baseadas no histórico de clientes similares.
- Interface web para seleção de clientes e exibição de recomendações.
- Visualização do histórico de compras.
- API REST para integração.

## Pré-requisitos

- Docker instalado.
- Arquivo `sells_data.csv` em `data/`.

## Docker

```
# Build da imagem
docker build -t recommendation-api .

# Executar container
docker run -p 5000:5000 \
  -v $(pwd)/data:/app/data \
  recommendation-api
```

Obs.: se a porta 5000 estiver ocupada, use `-p 5001:5000`.

## Como Executar

### Serviço principal (Flask + KNN)

```
python ai.py [-f path/to/best_params.json]
python app.py
```

Acesse em: <http://localhost:5000>

### Testbench

- Modo finito:

```
python testbench.py -n 50
```

Gera `testbench_50/logs/Exec_##/` e `testbench_50/metrics.json`.

- Modo indefinido:

```
python testbench.py -i
```

Atualiza continuamente o diretório `best/` até interrupção (Ctrl+C).

### Grid Search

```
python grid_search.py --out grid_results --reps 5
```

Gera:

- `grid_results/logs/combo_/run_/`
- `grid_results/best_params.json`
- `grid_results/grid_metrics.json`

### Parâmetros e Flags

Script	Flag	Descrição
<code>ai.py</code>	<code>-f</code> , <code>-best-params</code>	Carrega hiperparâmetros de <code>best_params.json</code> .
<code>testbench.py</code>	<code>-n</code> , <code>-n_runs</code>	Número de iterações finitas (padrão: 100).
<code>grid_search.py</code>	<code>-i</code> , <code>-indefinite</code>	Modo indefinido até Ctrl+C.
	<code>-o</code> , <code>-out</code>	Diretório base para logs/resultados (padrão: <code>grid_search/</code> ).
	<code>-r</code> , <code>-reps</code>	Número de runs por combinação (padrão: 10).

## Testes & Validação

- **Precision@K**: fração de itens recomendados que são relevantes.
- **Recall@K**: fração de itens relevantes que foram recomendados.
- Use `testbench.py` para medir estabilidade (ex.: média de retries).
- Use `grid_search.py` para otimizar hiperparâmetros (`K_VIZINHOS`, `K_RECS`, `ALPHA`, etc.).

## Métricas de Performance

O arquivo `grid_metrics.json` contém:

- **best**: melhores parâmetros e métricas.
- **combos**: resumo de cada combinação.
- **time\_metrics**:
  - `winner_run_time`: tempo do run vencedor.
  - `winner_combo_total_time`: tempo total do combo vencedor.
  - `winner_combo_pct_of_grid`: percentual do grid.
  - `combo_times`: tempo total e percentual de cada combo.
  - `total_grid_time`: duração completa do grid.

## Licença

Liberado sob uma licença de uso geral no contexto da UnB.  
Consulte `LICENSE` para obter mais detalhes.