Sistema de Recomendação de GPUs com TF-IDF e Tratamento dos Dados

Arthur Galvão Torres Venceslau

4/11/2025

Resumo

Este relatório descreve o desenvolvimento de um sistema de recomendação de placas de vídeo (GPUs) baseado em linha de comando. Utilizando técnicas de Processamento de Linguagem Natural (PLN), especificamente *TF-IDF* (*Term Frequency-Inverse Document Frequency*) e Similaridade de Cosseno, o sistema compara uma consulta de usuário em linguagem natural com um banco de dados de GPUs. O diferencial do projeto reside em uma robusta etapa de tratamento dos dados, onde dados estruturados (como preço e ano) e semiestruturados (como o nome do produto) são convertidos em um rico conjunto de descritores textuais para alimentar o modelo *TF-IDF*, aumentando significativamente a relevância das recomendações.

1 Introdução

Nos tempos atuais, a escolha de uma placa de vídeo (GPU) é difícil por conta da saturação do mercado e os preços. Isso se deve ao mercado saturado com uma vasta gama de produtos, onde a nomenclatura é notoriamente confusa, misturando séries, gerações e variantes (ex: ”RTX 4060 Ti”, ”RX 6600 XT”). Para um usuário, traduzir uma necessidade (”quero uma placa boa para jogos recentes sem gastar muito”) em uma decisão de compra informada é uma tarefa difícil.

O dataset de GPUs foi especificamente escolhido por representar um desafio de recomendação ideal para a técnica de *TF-IDF*. Os dados das placas de vídeo não são simples blocos de texto, mas sim uma mistura de:

- Dados semiestruturados (os nomes dos produtos, ex: ”GeForce GTX 1050 Ti”);
- Dados categóricos (o preço de lançamento, ex: "*Budget*", "*Mainstream*");
- Dados numéricos (o ano de lançamento, ex: ”2020”).

Este projeto propõe uma solução que aborda diretamente essa complexidade. O objetivo é criar um sistema de recomendação inteligente que utiliza o tratamento dos dados para converter essa mistura de informações em uma ”superdescrição” textual unificada. Em seguida, aplicando técnicas de Processamento de Linguagem Natural (*TF-IDF* e Similaridade de Cosseno), o sistema permite que o usuário encontre a GPU ideal usando linguagem natural, de forma intuitiva e eficaz.

2 Metodologia

A abordagem central do projeto baseia-se em extrair os dados do dataset, processá-los, adequá-los ao processo *TF-IDF* e, assim, retornar a recomendação ao usuário. Para isso, convertemos tanto as GPUs do nosso dataset quanto a consulta do usuário em vetores numéricos e calculamos a similaridade entre eles.

2.1 Fonte de Dados

A base de conhecimento do sistema é o *Graphics Card Database*, um catálogo detalhado de especificações de hardware mantido por Nathan Mayer. Originalmente um arquivo Excel (`Graphics Card Database.xlsx`), ele é estruturado em múltiplas planilhas.

Para este projeto, utilizamos duas categorias dessas planilhas:

- **Dados Principais (por Fabricante):** As planilhas `ATIAMD` e `NVIDIA` serviram como fonte primária, contendo as especificações de cada placa.
- **Metadados (Auxiliares):** A planilha `Pricing` foi fundamental, pois atua como uma ”legenda” que define as categorias de preço (ex: *Budget*, *Mainstream*), em vez de valores monetários brutos.

Este dataset é ideal para o projeto, pois já fornece dados pré-categorizados (preço) e estruturados (ano de lançamento) que são perfeitos para a etapa de tratamento dos dados. O primeiro passo da implementação (Etapa 1) é focado em extrair e consolidar apenas as colunas relevantes dessas planilhas em um único arquivo, `gpu_dataset_combinado.csv`.

2.2 Tratamento dos Dados

A etapa mais crítica do projeto é a criação de uma ”super-descrição” textual para cada GPU. Esta descrição combina os dados brutos de forma inteligente, criando um documento rico em palavras-chave que o *TF-IDF* pode processar.

2.2.1 Tratamento de Nomes de Produtos

Os nomes das GPUs (ex: ”GTX 1050 Ti”) são processados para extrair suas partes componentes. A lógica utiliza expressões regulares para identificar a série (ex: `'RTX'`, `'GTX'`, `'RADEON RX'`) e o número do modelo (ex: `'4070'`). Esse número é então quebrado em ”geração” (ex: `'generation 40'`) e ”identificador de modelo” (ex: `'model_id 70'`). Variantes como `'TI'` e `'XT'` também são extraídas como termos separados, permitindo buscas por famílias de produtos (ex: ”rtx geração 40”).

2.2.2 Tratamento de Datas (Ano de Lançamento)

O ano de lançamento (`Year of Release`) é utilizado de duas formas:

- **Como string exata:** O ano (ex: ”2023”) é incluído diretamente, permitindo buscas por ”placa de 2023”.

- **Como categoria:** O ano é classificado em categorias textuais para buscas mais semânticas, com base na seguinte lógica:
  - $\geq$ 2020: ”recente”
  - $\geq$ 2010 e $<$ 2020: ”antiga”
  - $<$ 2010: ”muito antigas”

2.2.3 Tratamento de Preço (Preço de Lançamento)

O preço de lançamento (`Price at Launch (MSRP)`), que já se encontra em formato categórico (ex: *Highend*, *Budget*), é mapeado de seus termos em inglês para descrições em português (ex: 'Alto Nível (Premium)', 'Entrada'). Isso facilita buscas em linguagem natural pelo usuário.

2.3 Modelo TF-IDF e Similaridade de Cosseno

Uma vez que cada GPU possui sua ”super-descrição” (coluna `combined_text`), utilizamos a classe `TfidfVectorizer` da biblioteca *Scikit-learn* com suas configurações padrão para:

1. Aprender o vocabulário de todas as descrições.
2. Criar uma matriz *TF-IDF*, onde cada linha é uma GPU e cada coluna é uma palavra do vocabulário, com seu respectivo score *TF-IDF*.

(Nota: A remoção de *stop words* não foi utilizada na implementação final para garantir que termos curtos como ”TI” ou ”XT” não fossem acidentalmente removidos.)

Quando um usuário insere uma consulta, ela é processada pelo *mesmo* vetorizador (usando o método `transform`) e convertida em um vetor. Em seguida, a `cosine_similarity` (Similaridade de Cosseno) é calculada entre o vetor da consulta e todos os vetores da matriz *TF-IDF*. O resultado é um score de similaridade entre 0 e 1 para cada GPU.

3 Implementação

O sistema foi implementado integralmente em Python. O processo foi dividido em duas etapas principais: 1) um script para consolidação dos dados brutos e 2) o script principal do sistema de recomendação.

3.1 Etapa 1: Consolidação e Limpeza dos Dados

O primeiro desafio foi extrair os dados relevantes do arquivo Excel `Graphics Card Database.xlsx`. Foi criado um script para automatizar esse processo, lendo apenas as planilhas dos fabricantes desejados (`ATIAMD` e `NVIDIA`) e selecionando um conjunto específico de colunas.

3.2 Etapa 2: Script Principal de Recomendação

O script principal é responsável por carregar o `gpu_dataset_combinado.csv` e aplicar a lógica de recomendação. As bibliotecas principais utilizadas foram *Pandas*, *Scikit-learn* e *re* (Regex).

3.2.1 Tratamento de Preço e Ano

Primeiramente, os dados de preço são traduzidos e os anos são categorizados usando funções de mapeamento.

3.2.2 Extração de Componentes do Nome

Uma função usando Regex (`re.search`) é definida para analisar o nome da GPU (`GPU_NAME`) e extrair seus componentes.

3.2.3 Criação da "Super-Descrição"

Todas as características criadas são concatenadas em uma única string de texto (a "super-descrição"), `combined_text`.

3.2.4 Vetorização e Loop de Recomendação

Finalmente, a matriz *TF-IDF* é criada e o sistema entra em um loop interativo, aguardando as consultas do usuário.

4 Resultados e Testes

Após executar o código, pôde-se constatar o seguinte: o script de recomendação foi executado com sucesso. A matriz *TF-IDF* final foi criada com as dimensões (468, 491), indicando que o sistema aprendeu um vocabulário de 491 termos únicos a partir das 468 GPUs do dataset consolidado (NVIDIA e ATIAMD).

Foram realizados testes práticos para validar a eficácia do tratamento dos dados:

Tabela 1: Exemplos de consultas e resultados do sistema.

Tipo de Busca	*Consulta (Query)*	*Resultado Principal (Top 1)*
Específica	`1050 ti`	*GPU*: GeForce GTX 1050 Ti | *Score*: 0.5118
Específica (c/ Marca)	`rx 6600`	*GPU*: Radeon RX 6600 | *Score*: 0.5990
Geração (AMD)	`rx 600`	*GPU*: Radeon RX 7600 | *Score*: 0.5753
Modelo (Geral)	`80`	*GPU*: Radeon RX 580 | *Score*: 0.3548
Série (NVIDIA)	`rtx`	*GPU*: Titan RTX | *Score*: 0.6314

4.1 Análise dos Resultados

Os resultados validam a abordagem de tratamento dos dados:

- **Buscas Específicas:** As buscas por ”1050 ti” e ”rx 6600” retornaram as placas exatas como melhor resultado, provando a eficácia do *TF-IDF*.
- **Buscas por Família:** As buscas por ”rx 600” e ”80” demonstram o sucesso da quebra dos nomes. O sistema identificou corretamente ”600” como parte de ”RX 7600” e ”RX 6600”, e ”80” como parte de ”RX 580” e ”RTX 2080” (que apareceu em 4º lugar).
- **Buscas por Série:** A busca por ”rtx” agrupou corretamente todas as placas da série ”RTX” no topo dos resultados, com a `Titan RTX` tendo o score mais alto.

5 Conclusão

Este projeto demonstrou a eficácia do modelo *TF-IDF* para sistemas de recomendação quando combinado com uma forte etapa de tratamento dos dados. Ao ”traduzir” dados estruturados (preço, ano) e semiestruturados (nome do produto) em um vocabulário textual rico, permitimos que o modelo compreenda nuances de preço, idade e famílias de produtos, entregando resultados mais relevantes para o usuário final. A implementação final, utilizando *Scikit-learn* e *Pandas*, provou ser robusta e eficiente para esta tarefa.