Relatório de Tabela Hash para Deduplicação de Dataset

integrantes : Felipe Rodrigues Ribeiro DRE:119052031

Igor Torres Torres da Costa DRE:119034669

Este estudo detalha a construção de uma estrutura de dados conhecida como Tabela Hash, ou Tabela de Dispersão, empregada para otimizar a solução de problemas de remoção de duplicidades em arquivos de dados, como os CSVs. A premissa central foi desenvolver uma tabela apta a inserir, pesquisar e excluir dados com base em uma identificação exclusiva, como o CPF, de um jeito ágil e prático. Para isso, foi criada uma classe denominada TabelaHash, que internamente emprega uma lista de listas (ou buckets) para guardar os dados. Essa opção de estrutura, chamada de encadeamento exterior, foi selecionada para lidar com as colisões que ocorrem quando diferentes identificações acabam criando o mesmo índice por meio da função de hash.

A função de hash padrão utilizada foi a função de divisão, que calcula o resto da divisão do valor numérico da identificação pelo tamanho da tabela. Contudo, a classe foi projetada de maneira adaptável para viabilizar que o programador defina outras funções de hash, caso queira explorar variações como funções baseadas em multiplicação ou métodos de dobra, dependendo do tipo de dados e da distribuição das identificações.

O propósito primordial dessa tabela foi aplicá-la em uma questão muito comum em ciência de dados: eliminar registros duplicados antes de análises ou uso em modelos de machine learning. A ideia é ler um arquivo CSV, escolher uma coluna como identificação (por exemplo, o CPF) e inserir cada linha na tabela hash. Quando a identificação já existe, sabemos que se trata de uma duplicata e simplesmente a ignoramos. Ao final, conseguimos obter todos os registros únicos e, se desejado, gerar um novo CSV já limpo e pronto para uso.

Durante o desenvolvimento, algumas complicações surgiram. Um dos maiores desafios foi certificar que a detecção de duplicatas fosse realmente precisa, principalmente quando se tratava de identificações numéricas em formato de texto. Muitas vezes, campos numéricos em CSV aparecem com zeros à esquerda ou com formatações diferentes, o que exigiu um cuidado extra ao comparar as identificações. Outra complicação foi implementar a lógica de colisões de forma eficaz, já que era necessário garantir que cada bucket guardasse os dados de forma organizada e que a busca dentro do bucket fosse rápida. Também foi necessário pensar bem no tamanho inicial da tabela hash, pois escolher um tamanho muito pequeno aumentava o número de colisões, enquanto um tamanho muito grande acabava desperdiçando espaço de memória.

Ademais, durante os testes, surgiram dificuldades para lidar com arquivos CSV muito grandes, pois era preciso garantir que a leitura fosse feita de forma eficaz e sem sobrecarregar a memória. Foi necessário utilizar a leitura linha a linha e tratar cada registro cuidadosamente.

Mesmo com algumas dificuldades, o uso da Tabela Hash se mostrou muito vantajoso. Ela possibilitou remover dados duplicados de forma eficiente, com uma única varredura dos dados, em tempo linear O(n). Isso é consideravelmente mais veloz do que as técnicas que dependem da ordenação, cujo custo é geralmente de O(n log n). Após a conclusão dos testes, os arquivos CSV resultantes continham apenas registros distintos, o que confirmou a eficácia da nossa abordagem.

Afora a sua rapidez, a estrutura da implementação é clara e bem organizada, o que facilita o seu entendimento e futuras modificações. Uma possibilidade seria testar diferentes abordagens para resolver conflitos, como o endereçamento aberto, ou ampliar o suporte para chaves combinadas (como, por exemplo, a junção do nome e da data de nascimento). Uma outra ideia interessante seria rastrear a frequência de cada chave, para analisar quantas vezes cada entrada repetida apareceu na base de dados original.

Em resumo, a Tabela Hash que criamos demonstrou ser uma ferramenta valiosa para o tratamento de informações, auxiliando na limpeza e organização dos dados de maneira ágil e segura. Essa solução é de grande utilidade na fase inicial de preparação dos dados em projetos de ciência de dados e inteligência artificial, já que um conjunto de dados limpo e livre de duplicatas é fundamental para assegurar resultados consistentes e análises mais exatas.