

Matéria: Tecnologia da Informação

Assunto: Programação e Análise de Dados

Resumo Teórico do Assunto

Para ter sucesso nas questões sobre "Programação e Análise de Dados", é fundamental compreender os conceitos de pré-processamento de dados, técnicas de análise e o funcionamento de algoritmos de busca.

Programação e Análise de Dados: Conceitos Essenciais

A área de Programação e Análise de Dados envolve a coleta, limpeza, transformação, modelagem e visualização de dados com o objetivo de descobrir informações úteis, apoiar a tomada de decisões e resolver problemas.

1. Pré-processamento e Manipulação de Dados

O **pré-processamento de dados** é uma etapa crucial na análise de dados e no aprendizado de máquina. Ele envolve a transformação dos dados brutos em um formato limpo e adequado para análise. Isso pode incluir:

- **Limpeza de Dados:** Tratamento de valores ausentes, remoção de ruídos e correção de inconsistências.
- **Transformação de Dados:** Normalização, agregação, discretização ou criação de novas variáveis (campos) a partir das existentes.
- **Redução de Dados:** Redução do volume de dados sem perder informações importantes.

NumPy é uma biblioteca fundamental em Python para computação numérica, especialmente para trabalhar com arrays multidimensionais e operações matemáticas de alto desempenho.

- **`np.where()`:** Esta é uma função poderosa do NumPy que permite aplicar uma condição a um array (ou coluna de um DataFrame) e atribuir valores diferentes com base no resultado dessa condição.

- * **Sintaxe:** `np.where(condição, valor_se_verdadeiro, valor_se_falso)`

- * **Funcionamento:** Para cada elemento do array, se a `condição` for verdadeira, o `valor_se_verdadeiro` é atribuído; caso contrário, o `valor_se_falso` é atribuído. É uma forma eficiente de realizar atribuições condicionais em massa.

2. Análise de Dados e Big Data

Big Data refere-se a conjuntos de dados tão grandes e complexos que os métodos

tradicionais de processamento de dados não são suficientes para lidar com eles. É caracterizado pelos "3 Vs":

- **Volume:** Grande quantidade de dados.
- **Velocidade:** Geração e processamento rápido dos dados.
- **Variedade:** Diversidade de formatos e tipos de dados (estruturados, semiestruturados, não estruturados).

A **Análise Exploratória de Dados (AED)** é uma abordagem inicial para analisar conjuntos de dados, resumindo suas principais características, muitas vezes com métodos visuais. Um dos objetivos da AED é identificar padrões, detectar anomalias e testar hipóteses.

• **Deteção de Outliers (Pontos Fora da Curva) ou Deteção de Desvios:** Um **outlier** é um ponto de dados que se desvia significativamente de outras observações. A deteção de outliers é o processo de identificar esses pontos incomuns que podem indicar erros na coleta de dados, eventos raros ou novas tendências. É crucial tratá-los (removê-los, corrigi-los ou analisá-los separadamente) pois podem distorcer os resultados de análises estatísticas e modelos de aprendizado de máquina.

3. Algoritmos de Busca

Algoritmos de Busca são procedimentos sistemáticos para encontrar um item específico dentro de uma coleção de dados.

• **Busca Binária (Binary Search):** É um algoritmo de busca eficiente para encontrar um item em uma lista **ordenada** de elementos.

* **Pré-requisito:** A lista (ou array) deve estar **ordenada** (ascendente ou descendente).

* **Princípio:** O algoritmo funciona dividindo repetidamente pela metade a parte da lista que pode conter o item, eliminando metade dos elementos restantes em cada passo.

* **Passos:**

1. Compara o valor-alvo com o elemento do meio da lista.
2. Se o valor-alvo for igual ao elemento do meio, a busca é concluída.
3. Se o valor-alvo for menor que o elemento do meio, a busca continua apenas na metade inferior da lista.
4. Se o valor-alvo for maior que o elemento do meio, a busca continua apenas na metade superior da lista.
5. Os passos 1-4 são repetidos até que o item seja encontrado ou o subconjunto da lista se torne vazio (indicando que o item não está presente).

* **Eficiência:** A busca binária é muito mais rápida que a busca linear para grandes conjuntos de dados, pois sua complexidade de tempo é logarítmica ($O(\log n)$).

Compreender esses conceitos e como eles se aplicam na prática é essencial para resolver questões relacionadas à programação e análise de dados em concursos.

Questões de Provas Anteriores

Fonte: escriturario_agente_de_tecnologia.pdf, Página: 18

pcimarkpci MjgwNDowMTRkOjE0YTU6OTI1ODozOGQ2OjNhMGM6NTM0MzplZmI1:U3VuLCAyNyBKdWwgMjAyNSAyMzo0NzozMSAtMDMwMA==
www.pciconcursos.com.br

18

BANCO DO BRASIL

AGENTE DE TECNOLOGIA - Microrregião 158 -TI GABARITO 1

61

Em uma base de dados (dataset) com avaliações sobre a qualidade de um determinado produto, o campo stars tem um valor de 1 a 5. Na fase de pré-processamento, decidiu-se criar o campo sentimento com valores bom ou ruim, e utilizando-se da biblioteca NumPy, escreveu-se o comando abaixo:
`dataset['sentimento'] = np.where(dataset['stars'] >= 4, 'bom', 'ruim')`
Quanto aos valores do campo sentimento, o referido comando atribuirá o valor

- (A) bom, para stars entre 2 e 5
- (B) bom, para stars 4 e 5
- (C) bom, para stars 3 e 4
- (D) ruim, para stars 4 e 5
- (E) ruim, para stars entre 1 e 4

62

Um profissional de TI está trabalhando com um grande banco de dados (Big Data), realizando uma análise prévia da base de dados, com o objetivo de identificar anomalias ou resultados raros, de forma a tratá-los ou descartá-los para utilização.

Esse profissional está realizando a seguinte tarefa:

- (A) Agrupamento dos dados
- (B) Análise de associações dos dados
- (C) Análise de segmentação dos dados
- (D) Análise de outliers (pontos fora da curva) ou detecção de desvios
- (E) Classificação dos dados e das anomalias

63

Para entender como o algoritmo de busca binária se comporta, um estudante de computação resolveu inserir um comando

`System.out.printf()` em um método chamado busca. Esse método, escrito em Java, realiza uma busca binária em

um array de números inteiros, ordenados de forma ascendente. O objetivo do `printf` é exibir, no console, o valor de cada

elemento do array visitado pelo algoritmo de busca binária.

Para testar o código que criou, o estudante escreveu o método `main` a seguir.

```
public class Main {
```

```
public static void main(String[] args) {  
    int lista[]={5,18,27,33,44,49,54,67,69,72,79,86,87,92};
```

```
// o array lista possui 14 elementos
```

```
    busca(78, lista);
```

```
}
```

```
public static int busca(int val,int lista[]) {
```

```
    // código relativo ao algoritmo de busca binária
```

```
}
```

```
}
```

O que será exibido no console quando o método main for executado?

(A) 54 79 69 72

(B) 49 72 86 79

(C) 54 86 69 72 79

(D) 67 86 72 79

(E) 54 79 67 69 72