

PROGRAMAÇÃO AVANÇADA EM BIG DATA

EXERCÍCIOS DE REVISÃO PARA N1

Exercício 1: Tratamento de Outliers e Medidas Descritivas (Unidade 1)

Cenário: Uma plataforma de streaming monitora o tempo médio diário de reprodução (em minutos) dos seus usuários. Durante uma verificação no pipeline de ingestão, a equipe identificou discrepâncias causadas por sessões ativas deixadas abertas por engano (bots ou telas esquecidas ligadas).

Problema: Escreva um script em Python para:

1. Receber uma lista amostral do tempo assistido em minutos: [45, 50, 52, 60, 58, 62, 51, 49, 530, 47].
2. Calcular e exibir o Primeiro Quartil (Q1), Terceiro Quartil (Q3) e o Intervalo Interquartil (IQR = Q3 - Q1).
3. Identificar o Limite Superior de corte do Boxplot (LS = Q3 + 1.5 × IQR) e filtrar os outliers do conjunto de dados.
4. Imprimir a média dos tempos antes e depois da remoção dos outliers para demonstrar o impacto dos valores discrepantes.

Exercício 2: Ajuste de Regressão Linear Simples via MQO (Unidade 2)

Cenário: O time de infraestrutura em nuvem deseja projetar o consumo diário de armazenamento (em Gigabytes) de um cluster de banco de dados com base na quantidade de transações de vendas processadas pelo sistema (em milhares).

Problema: Sem utilizar bibliotecas externas de Machine Learning (apenas Python puro/módulo math), desenvolva uma função que:

1. Receba os dados das últimas 5 semanas:
 - a) Transações em milhares (X): [10, 20, 30, 40, 50]
 - b) Armazenamento em GB (Y): [15, 25, 31, 45, 54]
2. Calcule os coeficientes angular (β_1) e linear (β_0) pelo Método dos Mínimos Quadrados Ordinários (MQO):

$$\beta_1 = \frac{\sum (X_i - \bar{X})(Y_i - \bar{Y})}{\sum (X_i - \bar{X})^2}, \quad \beta_0 = \bar{Y} - \beta_1 \bar{X}$$

3. Estime o armazenamento necessário (Y) caso a plataforma atinja um volume de 65 mil transações (X = 65).

Exercício 3: Matriz de Confusão e Métricas para Prevenção de Fraudes (Unidades 3 e 4)

Cenário: Uma fintech treinou um modelo para identificar transações fraudulentas de cartão de crédito ($Y=1$ para fraude, $Y=0$ para legítima). Para validar a implantação em produção, o modelo foi testado em uma amostra contendo as respostas reais e as respostas previstas.

Problema: Desenvolva um código em Python que:

1. Receba duas listas com 10 registros cada:
 - a) Rótulos Reais (Y): [1, 0, 0, 1, 0, 0, 1, 1, 0, 0]
 - b) Previsões ($\hat{Y}$): [1, 0, 0, 0, 0, 1, 1, 1, 0, 0]
2. Utilize um laço de repetição (for ou while) para percorrer as listas e contabilizar: Verdadeiros Positivos (VP), Falsos Positivos (FP), Verdadeiros Negativos (VN) e Falsos Negativos (FN).
3. Calcule e imprima as métricas de Acurácia, Precisão e Sensibilidade (Recall).

Exercício 4: Algoritmo de Agrupamento K-Means Nativo e Distância Euclidiana (Unidade 5)

Cenário: Um e-commerce deseja agrupar novos clientes sem rótulos prévios em 2 perfis de consumo ($K=2$) baseando-se na Idade e na Pontuação de Compras.

Problema: Escreva uma rotina em Python que:

1. Calcule a Distância Euclidiana bidimensional entre pontos: $d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$.
2. Considere os centroides iniciais fixos:
 - a) Centroide 0: (25, 80) (Perfil Jovem/Alto Gasto)
 - b) Centroide 1: (50, 20) (Perfil Sênior/Baixo Gasto)
3. Percorra a lista de novos clientes: [(22, 85), (48, 25), (28, 75), (55, 15), (30, 90)].
4. Classifique cada cliente atribuindo-o ao centroide mais próximo e exiba no terminal a lista dos clientes em cada um dos dois grupos resultantes.

Exercício 5: Higienização de Strings e Extrator Regex de Logs (Unidades 6 e 7)

Cenário: O pipeline de ingestão de Cybersecurity de um servidor precisa processar requisições web despadronizadas enviadas por múltiplos micros serviços.

Problema: Desenvolva um script em Python utilizando o módulo nativo re (Expressões Regulares) e métodos de manipulação de strings para:

1. Receber a lista bruta de logs de acesso:

```
logs_brutos = [  
    " 192.168.0.10 - [GET] /api/v1/usuarios - STATUS: 200 \n",  
    "10.0.0.1 - [POST] /api/v1/login - STATUS: 401\t",  
    " 172.16.254.3 - [GET] /admin/config - STATUS: 404 \n"  
]
```

2. Aplicar métodos de string (strip, upper) para higienizar espaços e quebras de linha.
3. Aplicar uma Expressão Regular para extrair três informações chave de cada log: o IP do cliente, o Método HTTP (GET/POST) e o Código de Status HTTP.
4. Exibir as informações extraídas em um formato limpo e padronizado.