

projeto_1

July 25, 2024

Olá, Angel!

Meu nome é Luiz. Fico feliz em revisar seu projeto. Ao longo do texto farei algumas observações sobre melhorias no código e também farei comentários sobre suas percepções sobre o assunto. Estarei aberto a feedbacks e discussões sobre o tema.

Peço que mantenha e não altere os comentários que eu fizer por aqui para que possamos nos localizar posteriormente, ok?

Mais uma coisa, vamos utilizar um código de cores para você entender o meu feedback no seu notebook. Funciona assim:

Comentário do revisor:

Sucesso. Tudo foi feito corretamente.

Comentário do revisor:

Alerta não crítico, mas que pode ser corrigido para melhoria geral no seu código/análise.

Comentário do revisor:

Erro que precisa ser arrumado, caso contrário seu projeto **não** será aceito.

Você pode interagir comigo através dessa célula:

Resposta do Aluno.

Comentário do revisor v1 Obrigado por enviar o seu projeto e pelo esforço para executar as etapas necessárias. Saber manipular corretamente as funções e operadores no python é muito relevante para uma carreira na área de dados.

No geral, o seu projeto está bom, você implementou corretamente a maioria das questões. Entretanto, deixei alguns comentários em questões que precisam de pequenos ajustes no código para avançarmos.

Se tiver alguma dúvida, pode contar comigo.

Até breve!

Comentário geral do revisor v2

Obrigado por enviar o seu projeto e fazer os ajustes sugeridos. Essa versão do seu trabalho ficou muito melhor! Espero que as sugestões sejam relevantes para projetos futuros.

Te desejo uma jornada de muito sucesso e aprendizado.

Qualquer dúvida, pode contar comigo.

Até breve!

Uma empresa de comércio eletrônico, Store 1, começou recentemente a coletar dados sobre seus clientes. O objetivo da Store 1 é entender melhor o comportamento dos clientes e tomar decisões baseadas em dados para melhorar experiência online deles.

Como parte da equipe analítica, sua primeira tarefa é avaliar a qualidade de uma amostra de dados coletados e preparar elas para análises futuras.

1 Quiz

A Store 1 visa garantir a consistência na coleta de dados. Como parte desse esforço, a qualidade dos dados coletados sobre os usuários precisa ser avaliada. Foi pedido que você revise os dados coletados e proponha alterações. Abaixo, você verá dados sobre um determinado usuário. Revise os dados e identifique possíveis problemas.

```
[ ]: user_id = '32415'
      user_name = ' mike_reed '
      user_age = 32.0
      fav_categories = ['ELECTRONICS', 'SPORT', 'BOOKS']
```

Opções:

1. O tipo de dados de `user_id` deve ser alterado de string para número inteiro (integer).
2. A variável `user_name` contém uma string com espaçamento desnecessário e um sublinhado entre o nome e o sobrenome.
3. O tipo de dados de `user_age` está incorreto.
4. A lista `fav_categories` contém strings em letras maiúsculas. Em vez disso, devemos converter os valores da lista para letras minúsculas.

Escreva na célula de marcação abaixo do número de opções que você identificou como problemas. Se você identificou vários problemas, separe o número por vírgulas. Por exemplo, se você acha que os números 1 e 3 estão corretos, escreva 1, 3.

Escreva sua resposta e explique seu raciocínio: 2, 3

Para a opção 2: como o Python reconhece um espaço como um caractere, ele interpretará que os espaços no início e no final da string de texto fazem parte do nome e isso está incorreto. então no banco de dados o registro: 'mike_reed' é diferente de ' mike_reed '

Para a opção 3: Atribuir um número do tipo (float: Q - R racional - real) a uma variável que representará a idade de um cliente associado à Store 1, permitindo assim a atribuição de valores decimais, complica a interpretação dos valores atribuídos às idades dos clientes como por exemplo: 5,5 irá gerar uma análise mais profunda nesse valor porque 5 pode representar 5 anos mas 0,5 anos é um valor que precisa ser convertido (meses e dias) por isso é melhor atribuir um número inteiro valor e em outro campo os meses e dias caso esta informação seja necessária

Comentário do revisor:

Boa análise, mas leve em consideração que essas características podem mudar a depender do problema em questão.

2 Tarefa 1

Vamos implementar as mudanças que identificamos. Primeiro, queremos corrigir os problemas com a variável `user_name`. Como verificamos, ela possui espaços desnecessários e um sublinhado como separador entre o nome e o sobrenome. Seu objetivo é remover os espaços e depois substituir o sublinhado por espaço.

```
[1]: user_name = ' mike_reed '  
user_name = user_name.strip() # remova os espaços na string original  
user_name = user_name.replace('_', ' ')# substitua o sublinhado por espaço  
  
print(user_name)
```

mike reed

Comentário do revisor:

Correto! O estudante usou corretamente os métodos `strip()` e `replace()`.

3 Tarefa 2

Em seguida, precisamos dividir o `user_name` atualizado em duas substrings para obter uma lista que contém dois valores: a string para o nome e a string para o sobrenome.

```
[2]: user_name = 'mike reed'  
name_split = user_name.split(' ') # divida a string user_name aqui  
  
print(name_split)
```

['mike', 'reed']

Comentário do revisor:

Correto! O método `split()` permite quebrar uma string em um delimitador. Dica: aqui você poderia usar `.split()` também (sem o ' ')

4 Tarefa 3

Ótimo! Agora queremos trabalhar com a variável `user_age`. Como mencionamos antes, ela possui um tipo de dados incorretos. Vamos corrigir esse problema transformando o tipo de dados e imprimindo o resultado final.

```
[3]: user_age = 32.0  
user_age = int(user_age) # altere o tipo de dados da idade de um usuário  
  
print(user_age)
```

32

Comentário do revisor:

Correto. O estudante converteu o tipo de dado do atributo “idade” para `int`.

5 Tarefa 4

Como sabemos, os dados nem sempre são perfeitos. Temos que considerar cenários em que o valor de `user_age` não pode ser convertido em um número inteiro. Para evitar que nosso sistema falhe, devemos tomar medidas com antecedência.

Escreva um código que tenta converter a variável `user_age` em um número inteiro e atribua o valor transformado a `user_age_int`. Se a tentativa falhar, vamos exibir uma mensagem solicitando que o usuário forneça sua idade como um valor numérico com a mensagem: `Forneça sua idade como um valor numérico`.

```
[5]: user_age = 'thirty two' # esta é a variável que armazena a idade como uma
    ↪string.
    try:
        user_age_int = int(user_age) # escreva um código que vai tentar transformar
    ↪user_age em um número inteiro e, se falhar, vai imprimir a mensagem
    ↪especificada
    except:
        print('Forneça sua idade como um valor numérico.')
```

Forneça sua idade como um valor numérico.

Comentário do revisor:

Aqui você poderia especificar qual tipo de exceção o seu código irá manipular. Adicione a exceção do tipo `ValueError` nesse caso.

Referência: https://www.w3schools.com/python/python_try_except.asp.

6 Tarefa 5

Por fim, observe que todas as categorias de favoritos são armazenadas em letras maiúsculas. Para preencher uma nova lista chamada `fav_categories_low` com as mesmas categorias, mas em letras minúsculas, repita os valores na lista `fav_categories`, os modifique e anexe os novos valores à lista `fav_categories_low`. Como sempre, imprima o resultado final.

```
[1]: fav_categories = ['ELECTRONICS', 'SPORT', 'BOOKS']
    fav_categories_low = []

    fav_categories_low = [aux.lower() for aux in fav_categories] # Aplicamos
    ↪Compreensão de lista para dentro do ciclo for aplicar à variável aux o
    ↪método lower e seja atribuído de uma vez à variável fav_categories_low

    # não remova a instrução de impressão abaixo
    print(fav_categories_low)
```

```
['electronics', 'sport', 'books']
```

Comentário do revisor: Aqui você precisa usar um loop para converter as strings de categorias para `lowercase` de forma automatizada.

Sugestão usando list comprehensions:

```
[cat.lower() for cat in categories]
```

Referência: https://w3schools.com/python/python_lists_comprehension.asp

Resposta do Aluno: Depois de ver a sugestão fui até o link e consegui entender a lógica das listas de compressão e resumindo linhas de código você pode percorrer a lista e aplicar o método ao mesmo tempo.

7 Tarefa 6

Conseguimos informações adicionais sobre os hábitos de consumo de nossos usuários, incluindo o valor gasto em cada uma de suas categorias favoritas. A administração está interessada nas seguintes métricas:

- Valor total gasto pelo usuário
- Valor mínimo gasto
- Valor máximo gasto

Vamos calcular e imprimir esses valores:

```
[1]: fav_categories_low = ['electronics', 'sport', 'books']
spendings_per_category = [894, 213, 173]

total_amount = sum(spendings_per_category) # Aqui calcula o valor total gasto
↳ pelo usuário
max_amount = max(spendings_per_category) # Aqui calcula o valor máximo gasto
↳ pelo usuário
min_amount = min(spendings_per_category) # Aqui calcula o valor mínimo gasto
↳ pelo usuário

# não remova a instrução de impressão abaixo
print(total_amount)
print(max_amount)
print(min_amount)
```

1280

894

173

Comentário do revisor:

Correto. O estudante calculou corretamente o valor total, mínimo e máximo.

8 Tarefa 7

A empresa quer oferecer descontos aos seus clientes fiéis. Clientes que fizerem compras totalizando mais de \$1.500 são considerados fiéis e vão receber um desconto.

Nosso objetivo é criar um ciclo `while` que verifique o valor total gasto e pare quando ele for atingido. Para simular novas compras, a variável `new_purchase` gera um número entre 30 e 80 em cada ciclo. Isso representa a quantidade de dinheiro gasto em uma nova compra, e é o que você precisa adicionar ao total.

Assim que o valor alvo for atingido e o ciclo `while` for encerrado, o valor final será impresso.

```
[6]: from random import randint

total_amount_spent = 1280
target_amount = 1500

while total_amount_spent < target_amount: # Comparando as despesas acumuladas
    ↳ com o valor da despesa condicional para o desconto
        new_purchase = randint(30, 80) # Atribuimos o valor aleatorio
    ↳ gastado pelo cliente: geramos um número aleatório de 30 a 80
        total_amount_spent += new_purchase # Atribuimos os gastos acumulados
    ↳ para cada ciclo (cada compra) à variável total_amount_spent

print(total_amount_spent)
```

1538

Comentário do revisor:

Correto, porém o seu código possui problemas de indentação (perceba que tem uma “->” no bloco de código). Considere apagar essa seta e usar o `tab` novamente para indentação.

Resposta do Aluno: Foi executado a indentação para estruturar corretamente o ciclo `while`

9 Tarefa 8

Agora temos todas as informações sobre um cliente da maneira que queremos. A administração de uma empresa nos pediu para encontrar uma maneira de resumir toda a informação sobre um usuário. Seu objetivo é criar uma string formatada que usa informações das variáveis `user_id`, `user_name` e `user_age`.

Aqui está a string final que queremos criar: Usuário 32415 chama-se mike e tem 32 anos.

```
[2]: user_id = '32415'
user_name = ['mike', 'reed']
user_age = 32
```

```

user_info = f'Usuário {user_id} chama-se {user_name[0]} e tem {user_age} anos.'
↪ # Aplicamos f-string para criar uma strings inserindo as variáveis nos
↪ espaços correspondentes

# não remova a instrução de impressão abaixo
print(user_info)

```

Usuário 32415 chama-se mike e tem 32 anos.

Comentário do revisor:

Correto. Bom trabalho usando f-strings!

Como você já deve saber, as empresas coletam e armazenam dados de uma maneira específica. A Store 1 deseja armazenar todas as informações sobre seus clientes em uma tabela.

user_id	user_name	user_age	purchase_category	spending_per_category
'32415'	'mike', 'reed'	32	'electronics', 'sport', 'books'	894, 213, 173
'31980'	'kate', 'morgan'	24	'clothes', 'shoes'	439, 390

Em termos técnicos, uma tabela é simplesmente uma lista aninhada que possui uma sublista para cada usuário.

A Store 1 criou essa tabela para seus usuários. Ela está armazenada na variável `users`. Cada sublista contém o ID do usuário, nome e sobrenome, idade, categorias favoritas e o valor gasto em cada categoria.

10 Tarefa 9

Para calcular a receita da empresa, siga estas etapas:

1. Use um ciclo `for` para iterar na lista `users`.
2. Extraia a lista de gastos de cada usuário e some os valores.
3. Atualize o valor da receita com o total de cada usuário.

Isso vai fornecer a receita total da empresa, que você vai imprimir no final.

```

[3]: users = [
    # este é o começo da primeira sublista
    ['32415', ['mike', 'reed'], 32, ['electronics', 'sport', 'books'],
     [894, 213, 173]
    ], # este é o fim da primeira sublista

    # este é o começo da segunda sublista
    ['31980', ['kate', 'morgan'], 24, ['clothes', 'shoes'],
     [439, 390]
    ] # este é o fim da segunda sublista
]

```

```

revenue = 0

for user in users:
    spendings_list = user[4]                # extraia a lista de gastos para
    ↪ cada usuário e some os valores
    total_spending = sum(spendings_list)    # some os gastos em todas as
    ↪ categorias para obter um total para um usuário específico
    revenue += total_spending               # atualize a receita

# não remova a instrução de impressão abaixo
print(revenue)

```

2109

Comentário do revisor:

Correto!

11 Tarefa 10

Use um ciclo for para percorrer a lista de usuários que fornecemos e imprima os nomes dos clientes com menos de 30 anos.

```

[2]: users = [
    ['32415', ['mike', 'reed'], 32, ['electronics', 'sport', 'books'],
    [894, 213, 173]],
    ['31980', ['kate', 'morgan'], 24, ['clothes', 'books'], [439,
    390]],
    ['32156', ['john', 'doe'], 37, ['electronics', 'home', 'food'],
    [459, 120, 99]],
    ['32761', ['samantha', 'smith'], 29, ['clothes', 'electronics',
    'beauty'], [299, 679, 85]],
    ['32984', ['david', 'white'], 41, ['books', 'home', 'sport'], [234,
    329, 243]],
    ['33001', ['emily', 'brown'], 26, ['beauty', 'home', 'food'], [213,
    659, 79]],
    ['33767', ['maria', 'garcia'], 33, ['clothes', 'food', 'beauty'],
    [499, 189, 63]],
    ['33912', ['jose', 'martinez'], 22, ['sport', 'electronics', 'home'
    ], [259, 549, 109]],
    ['34009', ['lisa', 'wilson'], 35, ['home', 'books', 'clothes'],
    [329, 189, 329]],
    ['34278', ['james', 'lee'], 28, ['beauty', 'clothes', 'electronics'
    ], [189, 299, 579]],
    ]

```



```

users_maior_30 = [] # Criamos uma variável vazia de tipo list para armazenar as
↳ listas filtradas

for user in users: # Criamos ciclo for para percorrer a lista
↳ principal users
    if user[2] < 30: # Comparamos o valor da posição 2 de cada
↳ sublista (idade) com o operador de comparação menor que (<).
        users_maior_30.append(user) # Todos os valores das sublists na posição
↳ 2 que atendem a condição aplicando o método append toda a sublista é
↳ armazenada na lista users_maior_30

for user in users_maior_30: # Criamos ciclo for para percorrer a nova
↳ lista filtrada
    user = ' '.join(user[1]) # Atribuímos à variável de ciclo user a
↳ aplicação do método join() para converter em strings
    print(user) # Aplicamos a função imprimir para
↳ mostrar os nomes dos clientes com menos de 30 anos.

```

```

kate morgan
samantha smith
emily brown
jose martinez
james lee

```

Comentário do revisor: Aqui você precisa imprimir somente o nome dos clientes. Sugestão: você pode acessar o nome dos clientes em `user[?]`, onde `?` deve ser o índice da lista acima. Você pode imprimir todos os índices para testar, mas o resultado acima nos mostra que o índice do nome é o segundo da lista (lembrando que índices no python iniciam em 0). Além disso, você pode concatenar o nome e sobrenome dos usuários em uma string única. Você consegue fazer isso, por exemplo, usando o método `.join()` ou `f-strings`.

Resposta do Aluno: Após a aplicação das sugestões recebidas, as linhas de código foram ajustadas dentro do ciclo correspondente.

Comentário do revisor v2:

Sugestão: aqui você poderia fazer tudo no mesmo ciclo/loop, pois você precisa iterar sobre a mesma estrutura de dados. Considere essa implementação para projetos futuros.

12 Tarefa 11

Vamos juntar as tarefas 9 e 10 e imprimir os nomes de usuários com menos de 30 anos com gastos totais acima de 1.000 dólares.

```

[3]: users = [
    ['32415', ['mike', 'reed'], 32, ['electronics', 'sport', 'books'],
    [894, 213, 173]],
    ['31980', ['kate', 'morgan'], 24, ['clothes', 'books'], [439,
    390]],

```

```

['32156', ['john', 'doe'], 37, ['electronics', 'home', 'food'],
[459, 120, 99]],
['32761', ['samantha', 'smith'], 29, ['clothes', 'electronics',
'beauty'], [299, 679, 85]],
['32984', ['david', 'white'], 41, ['books', 'home', 'sport'], [234,
329, 243]],
['33001', ['emily', 'brown'], 26, ['beauty', 'home', 'food'], [213,
659, 79]],
['33767', ['maria', 'garcia'], 33, ['clothes', 'food', 'beauty'],
[499, 189, 63]],
['33912', ['jose', 'martinez'], 22, ['sport', 'electronics', 'home'
], [259, 549, 109]],
['34009', ['lisa', 'wilson'], 35, ['home', 'books', 'clothes'],
[329, 189, 329]],
['34278', ['james', 'lee'], 28, ['beauty', 'clothes', 'electronics'
], [189, 299, 579]],
]

```

```

for user in users:                                # Criamos um ciclo for para
↳percorrer a lista principal
    if user[2] < 30:                                # Criamos um condicional para
↳filtrar na posição 2 os valores menores que 30
        gastos = user[4]                            # Atribuímos os valores da posição
↳4 de cada ciclo a uma variável chamada gastos do tipo lista.
        gastos_acum = sum(gastos)                   # Aplicamos a função sum() para
↳sumar todos os valores da posição 4 da sublista e obter um só valor acumulado
        user.pop()                                  # Da sublista dentro do ciclo
↳removemos o ultimo item (mesmo sendo de tipo lista com varios valores)
        user.append(gastos_acum)                    # Adicionamos na sublista do ciclo
↳o valor acumulado na variável gastos_acum aplicando o método
        if user[4]>1000:                             # Criamos outro condicional para
↳continuar filtrando dentro do mesmo ciclo a sublista já filtrada (para <30)
↳e em sua posição 4 filtamos os valores > 1000
            users_nome = ' '.join(user[1])          # Aplicamos o método join() para
↳concatenar os 2 itens para formar o nome
            print(users_nome)                        # Aplicamos a função print dentro
↳do ciclo para imprimir os valores já filtrados

# escreva seu código aqui

```

samantha smith
james lee

Comentário do revisor: Aqui também você precisa imprimir somente o nome dos clientes. Sugestão:

```
user_name = user[1][0]
surname = user[1][1]
print(f"{user_name} {surname}")
```

Resposta do aluno: Após a aplicação das sugestões recebidas, as linhas de código foram ajustadas dentro do ciclo correspondente.

13 Tarefa 12

Agora vamos imprimir o nome e a idade de todos os usuários que compraram roupas (clothes). Imprima o nome e a idade na mesma instrução de impressão.

```
[4]: users = [
    ['32415', ['mike', 'reed'], 32, ['electronics', 'sport', 'books'],
     [894, 213, 173]],
    ['31980', ['kate', 'morgan'], 24, ['clothes', 'books'], [439,
     390]],
    ['32156', ['john', 'doe'], 37, ['electronics', 'home', 'food'],
     [459, 120, 99]],
    ['32761', ['samantha', 'smith'], 29, ['clothes', 'electronics',
     'beauty'], [299, 679, 85]],
    ['32984', ['david', 'white'], 41, ['books', 'home', 'sport'], [234,
     329, 243]],
    ['33001', ['emily', 'brown'], 26, ['beauty', 'home', 'food'], [213,
     659, 79]],
    ['33767', ['maria', 'garcia'], 33, ['clothes', 'food', 'beauty'],
     [499, 189, 63]],
    ['33912', ['jose', 'martinez'], 22, ['sport', 'electronics', 'home'
     ], [259, 549, 109]],
    ['34009', ['lisa', 'wilson'], 35, ['home', 'books', 'clothes'],
     [329, 189, 329]],
    ['34278', ['james', 'lee'], 28, ['beauty', 'clothes', 'electronics'
     ], [189, 299, 579]],
    ]

for user in users:
    # Criamos o ciclo para percorrer a
    ↪ lista original users utilizando como sublista a variável user
    auxiliar = ','.join(user[3])          # Aplicamos o método join() só por o
    ↪ itens correspondente ao espaço 3 da sublista user para ser convertido de
    ↪ tipo lista ao string e atribuímos à variável auxiliar
    if 'clothes' in auxiliar:             # Criamos um condicional para
    ↪ verificar e filtrar se a string 'clothes' estava na strings
        nome_string = ' '.join(user[1])  # Caso sim, aplicamos o método join()
    ↪ para a posição 1 (nome) para concatenar o nome dos clientes
```

```
print(nome_string, user[2])      # Utilizamos a função print para
↳ imprimir o nome e a idade de todos os usuários que compraram roupas
↳ (clothes)
```

```
kate morgan 24
samantha smith 29
maria garcia 33
lisa wilson 35
james lee 28
```

Comentário do revisor:

Correto, mas aqui você poder concatenar o nome dos usuários em uma string.

Resposta do aluno: Foi concatenado os nomes de acordo com a sugestão recebida

Como o objetivo da equipe de análise da empresa Store 1 é entender o comportamento do cliente com base em dados, o primeiro ponto a analisar após a revisão do banco de dados é se a amostra estatística é verdadeiramente representativa da população ou do mercado de clientes. Diversas variáveis poderiam ser consideradas. , como se a empresa possui um mercado estratificado e se aquela amostra é de um único estrato ou, em última análise, saber quais métodos foram aplicados para obter a amostra selecionada ou determinar as causas, se houver (custos, logística, contratação de funcionarios, deficit orçamento, etc).