

Tópicos em Engenharia de Software: Relatório Técnico

Pedro Lucas Aires dos Santos

github.com/Pedrolaires/pratica-topicos-eng-software-python

Introdução

A atividade prática de Teste de Código em Python para Processamento de Dados com foco em Big Data, visa lidar, principalmente, os erros previsíveis que podem acontecer no processamento de um volume muito grande de dados.

1 Desenvolvimento de Testes Unitários

1.1 Função avgAgeCountry

- Arquivo JSON vazio: caso o arquivo JSON esteja vazio, o retorno será None.
- Valores de idade ausentes ou nulos: O teste da função espera que o processamento identifique e trate como 0 os dados que são encontrados neste estado para não corromper o cálculo.
- Campo 'country' ausente ou nulo. Assim como se os valores de idade forem ausentes, o teste espera que, caso o dado seja nulo, o processamento pule este dado.
- nome fornecido para pesquisa no processamento ausente: O retorno esperado é nulo.

1.2 Outras Funções de Processamento

- `most_common_name_by_country`: Função que busca o nome mais comum conforme o país, permitindo a busca por nome ou sobrenome.
- `age_pyramid`: Visa retornar uma pirâmide etária do país, contando a quantidade de pessoas que estão presentes em cada faixa etária, sendo ela definida por: 0-10 anos, 11-20 anos, etc...

1.3 Função de Transformação

A função de transformação de idade, é implementada a fim de gerar mais flexibilidade na tratativa dos dados. Por exemplo, no teste realizado, multiplicando o valor da idade por 12 obtivemos a conversão de anos para meses, de forma efetiva e rápida, reutilizando a função mas não perdendo o seu propósito.

2 Objetivos dos Testes

- Tipos de problemas que estavam sendo prevenidos: Dados nulos ou imprevistos foram tratados como "não existentes" durante o cálculo, permitindo que o software continue funcionando. Posteriormente poderia ser gerado um relatório com os casos que estavam inconsistentes
- Relevância do teste unitário e a liberdade para criar diferentes asserts: O teste Unitário é um componente fundamental para atestar a assertividade e qualidade de um software. Tendo em vista a tratativa de dados, os asserts são funções extremamente relevantes para indicar erros de cálculo ou mudanças não previstas.
- A intenção de cada teste: Cada teste tem um propósito específico, testando não somente o caso ótimo, onde os dados são perfeitos mas também os casos realistas, onde muitos deles estão "corrompidos" por falta de informações

3 Reflexão sobre Testes em Big Data

Testar grandes volumes de dados é desafiador, tendo em vista o processamento, tratativa de erros e o tempo de execução dos testes. Acredito que com o PySpark e seu processamento paralelo poderoso, os testes poderiam ter sido realizados em um volume de dados muito maior e em muito menos tempo. Avaliando a quantidade de dados do teste e na velocidade de cada um, poderia ter sido usado em comparação com as implementações do PySpark para atestar a eficiência do framework.

4 Conclusão

Tratar dados são de extrema importância, e isso se torna evidente ao se trabalhar com um volume muito grande. Realizei os testes antes mesmo de criar as novas funcionalidades e, com isto, absteri melhor o intuito do processamento que gostaria de realizar e as possibilidades utilizando as funções de transformação.