Entrevista para Engenheiro de Dados

Dataset

- biosales
- biosales-buyers
- biosales-sellers
- biosales-regions
- biosales-items
- biosales-platforms

Enunciado

A empresa Fakenexo possui uma plataforma online de compra e venda de produtos hospitalares, e ao final de todo mês é realizada uma extração de vendas do período para calcular o valor médio de todos os produtos comercializados por região, organizado em ordem descendente (do maior para o menor) em relação a este valor.
Esta extração é realizada por uma outra ferramenta que o time não possui acesso, que disponibiliza os dados na nuvem através dos links de download listados anteriormente. Estes arquivos precisam ser baixados e analizados por um arquivo python que possui um template idêntico ao disponibilizado em nosso repositório biosales.py.
Como a empresa Fakenexo acredita que seu dataset irá aumentar consideravelmente ao longo dos anos, seus desenvolvedores optaram por utilizar a ferramenta Apache Spark para processar estes dados. Foi definida a versão 2.3.0 como a base para os seus desenvolvimentos. Além disso, o código deve ser escrito utilizando a linguagem Python na versão 3.5 ou superior.
Os resultados finais devem ser gravados em 2 diretórios distintos, chamados `biosales-by-region-rdd.csv` e `biosales-by-region-sql.csv`, que deverão ser estruturados de acordo com algumas regras pré-definidas pelo time.

Em outras palavras, as 2 funções disponibilizadas no arquivo `biosales.py` (link aqui) devem realizar o download dos arquivos csv necessários que estão disponibilizados no bucket s3, processá-los utilizando o Apache Spark, e salvar o resultado final em um arquivo csv com a seguinte estrutura:

- 1a coluna: Nome da região;
- 2a coluna: Nome do produto;
- 3a coluna: Valor médio do produto;

Todos os campos devem ser separados por `;`. Este arquivo deve ser gerado via spark, ou seja, não deve ser escrito utilizando a function `open()` nativa do Python. Um exemplo de

como deve ser a estrutura final pode ser encontrada em nosso exemplo pelo link biosales-by-region. Note que o resultado final gerado é um diretório, já que o Spark irá escrever arquivos particionados dentro deste diretório (utilizando como prefixo a string `part-0000`).
Sinta-se a vontade para particionar os dados pela região, com o intuito de melhorar a perfomance das consultas que serão executadas pelas outras equipes (neste caso, o arquivo csv final conterá apenas o nome do produto e o valor médio).

Cada uma das funções do arquivo python devem ser preenchidas realizando o mesmo cálculo, porém com estratégias diferentes:

- Function `price_by_region_rdd`: deve utilizar SOMENTE o conceito RDD (Resilient Distributed Datasets) do Apache Spark (utilizando funções como `mapValues`, `reduceByKey`, `sortByKey`, etc.) para calcular o resultado final;
- Function `price_by_region_sql`: deve utilizar a função SQL do Apache Spark (utilizando a função `sql`, realizando as devidas transformações previamente necessárias para isso) para calcular o resultado final;

Ambas as funções devem baixar o arquivo s3 para o diretório local onde será executado o script python, o qual está em um bucket s3 público, como mencionado anteriormente. O resultado final também deve ser escrito no mesmo local, de acordo com o nome da respectiva função utilizada (`biosales-by-region-rdd.csv` ou `biosales-by-region-sql.csv`).

Todos os arquivos csv possuem cabeçalho referente ao que cada coluna representa. O arquivo principal, ou seja, o arquivo que possui relacionamento com todos os outros é o `biosales.csv`. Este arquivo possui 10 campos, sendo eles:

- id_grupo_comprador: Os hospitais, no intuito de encontrar um preço mais barato de um determinado produto, acabam formando grupos hospitalares. Este campo referencia o ID do grupo o qual o hospital faz parte;
- id_regiao_comprador: Cada hospital pertence a uma região do Brasil (Norte, Nordeste, Sul, Sudeste e Centro-Oeste). Este campo referencia o ID desta região;
- id_comprador: Cada hospital possui uma identificação única de acordo com o grupo de compradores a qual pertence, ou seja, a junção entre id_grupo_comprador e id_comprador identificam um único hospital. Sendo assim, este campo referencia o ID do hospital que realizou a compra dentro de um determinado grupo;
- id_vendedor: Este campo referencia o ID do fornecedor que fechou a compra com o hospital;
- id_pataforma_utilizada: Os hospitais podem realizar a compra por diversas plataformas (web, mobile, ERP, etc). Este campo referencia o ID desta plataforma;
- id_venda: Uma compra realizada pelo hospital pode ter, no mínimo, 1 item e, no máximo, infinitos itens. Este campo referencia o ID da venda realizada de um ou mais itens;
- id_item: Itens diversos ou do mesmo tipo podem ser comercializados em uma mesma venda com diferentes preços. Este campo referencia o ID do item comercializado em uma

venda;
- quantidade_item: Este campo referencia a quantidade de itens que foram comercializados em uma determinada venda;
- preco_unitario_item: Este campo referencia o preço unitário de cada item comercializado em uma determinada venda;
- data_compra: Este campo referencia a data em que a compra foi realizada;

Os outros arquivos possuem somente informações complementares referentes aos nomes dos compradores, fornecedores, itens, regiões e plataformas.

Como a empresa se preocupa com a qualidade de seus serviços, testes são indispensáveis para minimizar o risco de problemas no ambiente de produção. Caso você esteja opte por criar outras functions que auxiliam na resolução o problema, é necessário criar testes unitários para cada uma delasutilizando o módulo `pytest`.

Envio

É estritamente necessário que as 2 funções descritas anteriormente não tenham seu nome e seus argumentos modificados, para fins de avaliação por parte do time de desenvolvimento. Fique a vontade para criar outras functions para reaproveitar código, diminuir a complexidade, etc.

Quando o exercício for finalizado, o script python `biosales.py` juntamente com os arquivos de teste, devem ser enviados para a equipe de desenvolvedores da Bionexo pelo e-mail `analytics@bionexo.com` com o assunto "teste de seleção Bionexo". Por favor, não coloque os arquivos em um repositório público para que não haja chances de plágio, ok?

Aqui vão algumas dicas que serão úteis para o desenvolvimento do seu script:

- Alguns desenvolvedores alertaram o time sobre um possível bug onde alguns preços de alguns itens vieram com valor zero ou negativos. Estes devem ser eliminados do cálculo para não afetarem o valor final da média e do desvio padrão dos produtos. Além disso, alguns hospitais não foram cadastrados com a região a qual pertencem. Estes também devem ser eliminados do cálculo;
- Tenha paciência, leia atentamente tudo o que foi descrito anteriormente, pois detalhes podem facilmente passar despercebido. Não enxergue este exercício como Rocket Science, muito pelo contrário, ele provavelmente será mais simples do que você imagina;
- Qualquer dúvida que você tiver, não tenha medo, vergonha ou qualquer outra coisa! Pergunte! Ninguém nasceu sabendo tudo, e nós desenvolvedores da Fakenexo estamos sempre disponíveis para discutir sobre qualquer coisa! Afinal de contas, o que queremos é entregar um produto completo com a menor chance de erros possíveis, não é mesmo? Envie um e-mail para analytics@bionexo.com com a sua dúvida e estaremos sempre prontos para te ajudar!

Sucesso e boa sorte!