✨ História exclusiva para membros

Introdução à qualidade de dados com Apache Spark

Serge Smertin · Seguir
Publicado em MLOps como fazer
7 minutos de leitura · 4 de maio

Listen Share

No mundo **da engenharia de dados** , a equipe de dados implanta o **primeiro pipeline** até a produção e todos ficam felizes. Eles implantam o **segundo, terceiro,**

quinto e décimo pipelines para produção.

Mas então, eles começaram a pensar *, estamos monitorando os dados?* Nosso pipeline ETL é **saudável** e **robusto** o suficiente para uso em produção para que outras equipes confiem nos conjuntos de dados produzidos por esse pipeline?

"A qualidade dos dados requer um certo nível de sofisticação dentro da empresa até mesmo para entender que é um problema". Esta citação foi de Colleen Graham no artigo Performance Management Driving BI Spending de 2006, mas ainda se aplica aos dias de hoje.

Por que nos importamos?

A qualidade dos dados é um aspecto vital de qualquer sistema de processamento de dados. Com dados de alta qualidade, as empresas podem tomar decisões precisas e atingir seus objetivos. O Apache Spark é uma estrutura robusta de processamento de dados de código aberto que oferece várias ferramentas para garantir a qualidade dos dados.

Vamos testar a qualidade dos dados com o Apache Spark e como isso pode ser alcançado. As empresas precisam tomar decisões informadas. A qualidade dos dados pode levar a insights corretos, previsões precisas e decisões de negócios falhas.

Dados de baixa qualidade também podem afetar negativamente a satisfação do cliente e resultar em perda de receita. Portanto, é crucial garantir a qualidade dos dados para evitar essas armadilhas e alcançar os resultados desejados.

O Apache Spark é uma escolha popular para processamento de dados e oferece várias ferramentas para garantir a qualidade dos dados. Aqui estão algumas das maneiras pelas quais o Spark pode ser usado para melhorar a qualidade dos dados.

Criação de perfil de dados

A criação de perfis de dados é um aspecto crucial da qualidade dos dados e é essencial garantir que os dados usados para análise sejam precisos, completos e consistentes.

O Apache Spark é uma ferramenta ideal para criação de perfil de dados, pois fornece uma ampla variedade de funções de análise de dados e pode lidar com grandes conjuntos de dados em tempo real. Com o Apache Spark , a criação de

perfis de dados pode ser executada de forma rápida e eficiente, permitindo que os analistas identifiquem e corrijam problemas de qualidade de dados imediatamente.

O melhor da criação de perfil de dados com o Apache Spark é que a aceleração para os analistas de dados obter insights sobre os dados que estão trabalhando dentro das ordens de magnitude mais rapidamente. Os analistas podem identificar inconsistências ou erros explorando o esquema de dados e examinando os tipos de dados. Esse processo pode ajudá-los a entender as relações entre os vários elementos de dados e identificar quaisquer problemas de qualidade de dados que possam estar presentes.

Outro benefício importante da criação de perfil de dados com o Apache Spark é que ele permite que os analistas executem verificações de qualidade de dados com eficiência. Ao usar as funções integradas do Spark para identificar dados ausentes ou inválidos, os analistas de dados podem identificar rapidamente quaisquer problemas de qualidade de dados que precisam ser resolvidos. Esse processo pode melhorar a precisão dos dados e garantir que eles sejam adequados para uso em análises e tomadas de decisão.

O Spark oferece várias ferramentas para criação de perfis de dados, o que ajuda a entender a estrutura e a qualidade dos dados. Dados de criação de perfil podem identificar problemas de qualidade de dados, como valores ausentes, duplicados e inconsistências. A API DataFrame fornece uma variedade de funções para criação de perfil de dados, como `describe()` e `summary()`. Essas funções fornecem uma visão geral das propriedades estatísticas dos dados e ajudam a identificar exceções, anomalias e distribuição de dados.

Por exemplo, considere um conjunto de dados contendo informações do cliente, como nome, idade e endereço. Usando a `describe()` função, podemos obter propriedades estatísticas de cada coluna no conjunto de dados, como **contagem** , **média** , **desvio padrão** , valores **mínimo** e **máximo** . Podemos identificar valores ausentes ou discrepantes analisando essas propriedades estatísticas e tomando as medidas necessárias para melhorar a qualidade dos dados.

Limpeza de dados

A limpeza de dados é essencial para garantir a qualidade dos dados usados para análise. Envolve identificar e corrigir quaisquer imprecisões, inconsistências ou erros nos dados. Um dos principais benefícios da limpeza de dados com o Apache

Spark é que ele permite que os analistas de dados identifiquem e corrijam problemas de qualidade de dados rapidamente. Usando as funções integradas do Spark para identificar dados ausentes ou inválidos, os analistas de dados podem localizar rapidamente problemas de qualidade de dados e tomar medidas para corrigi-los. Esse processo pode ajudar a melhorar a precisão dos dados e garantir que eles sejam adequados para análise e tomada de decisão.

A limpeza de dados com o Apache Spark também pode ajudar a melhorar a eficiência geral do processamento de dados. Os analistas eliminam dados desnecessários ou redundantes e tempo de processamento significativo, tornando o processo de análise mais rápido e eficiente. Esse processo pode ajudar as organizações a usar seus recursos de dados e melhorar as práticas gerais de gerenciamento de dados.

A API DataFrame do Spark oferece várias funções para limpeza de dados, que podem ser usadas para limpar e transformar dados. A função `na` pode ser usada para lidar com valores ausentes e a `dropDuplicates()` função pode ser usada para remover registros duplicados. A `regexp_replace()` função pode substituir caracteres e `cast()` pode converter tipos de dados. Essas funções podem ser usadas para limpar dados e garantir a qualidade dos dados.

Por exemplo, considere um conjunto de dados contendo informações do produto, como `product_name`, `price` e `description`. A `product_name` coluna inclui alguns caracteres especiais que precisam ser removidos. Usando a `regexp_replace()` função, podemos remover esses caracteres e garantir a qualidade dos dados. Da mesma forma, a `price` coluna contém valores decimais que são armazenados como strings. Usando a `cast()` função, podemos converter o tipo de dados da `price` coluna em float e garantir a qualidade dos dados.

Data de validade

A validação de dados é essencial para garantir a qualidade dos dados usados para análise. Envolve verificar se os dados estão em conformidade com regras e padrões de negócios específicos e se são precisos e consistentes.

O benefício da validação de dados com o Apache Spark é que ele permite que os analistas de dados realizem a validação cruzada e identifiquem inconsistências em várias fontes de dados. Os analistas podem comparar dados de diferentes fontes usando as funções de integração de dados do Spark e identificar inconsistências ou

erros. Esse processo pode ajudar a garantir que os dados sejam consistentes e precisos em todas as fontes de dados, melhorando a confiabilidade dos dados e permitindo decisões mais bem informadas.

A API DataFrame do Spark fornece várias funções para validação de dados, que podem ser usadas para validar a exatidão dos dados. A `when()` função pode aplicar instruções condicionais aos dados, como filtragem ou transformação. Essas funções podem ser usadas para validar dados e garantir sua exatidão.

Padronização de dados

Embora ter um esquema de dados padrão seja vital para comparação e análise de dados, é importante observar que nem todos os dados seguirão necessariamente esse esquema. Isso é especialmente verdadeiro em setores com fontes e formatos de dados altamente variados. Algumas indústrias até têm esquemas de dados padronizados para segurança cibernética, como STIX 2.0 .

No entanto, mesmo nos casos em que os dados podem não estar em conformidade com um esquema padronizado, a API DataFrame do Apache Spark fornece uma variedade de funções que podem ser usadas para padronizar os dados e garantir a consistência.

Por exemplo, as funções `lower()` , `upper()` e `trim()` podem padronizar dados de texto convertendo-os em maiúsculas e minúsculas consistentes e removendo qualquer espaço em branco desnecessário. Isso pode ajudar a garantir que os dados sejam consistentes e fáceis de trabalhar, mesmo que não sigam um esquema estrito.

Além dos dados de texto, a API DataFrame do Spark também fornece funções para padronizar dados numéricos e dados de texto, além de dados de texto. Por exemplo, a função round() pode arredondar valores numéricos para um número especificado de casas decimais. Em contraste, a função cast() pode converter tipos de dados em um formato consistente. Essas funções podem ser particularmente úteis nos casos em que os dados podem ser fornecidos em formatos diferentes ou com níveis variados de precisão.

Por exemplo, considere um conjunto de dados contendo informações do cliente

ajudar a garantir a qualidade dos dados, facilitando a comparação e a análise dos dados.

Enriquecimento de dados

O enriquecimento de dados está aprimorando os dados existentes com informações ou contexto adicionais. Isso pode ser feito complementando os dados existentes com dados de fontes externas ou usando técnicas de manipulação de dados para extrair insights adicionais dos dados existentes. No contexto da qualidade dos dados, o enriquecimento de dados pode desempenhar um papel crítico na melhoria da qualidade geral dos dados. Ao adicionar informações adicionais aos dados existentes, os analistas podem obter uma compreensão mais profunda dos dados e tomar decisões mais informadas com base nos insights que obtêm.

A API DataFrame do Spark fornece várias funções para enriquecimento de dados, como `join()` e `union()` . Essas funções podem combinar dados de várias fontes e enriquecer os dados.

Por exemplo, considere um conjunto de dados contendo pedidos de clientes, como nome do produto, preço e quantidade. O conjunto de dados não inclui informações sobre a categoria do produto, o que pode ser útil na análise. Usando a `join()` função, podemos combinar o `order` conjunto de dados com um `product` conjunto de dados contendo informações de categoria de produto. Esse processo pode enriquecer os dados e melhorar a qualidade dos dados, fornecendo mais informações para análise.

Resumo

A qualidade dos dados é crucial para que as empresas tomem decisões informadas e alcancem seus objetivos. O Apache Spark fornece várias ferramentas para garantir a qualidade dos dados, como criação de perfil, limpeza de dados, validação, padronização e enriquecimento. Ao usar essas ferramentas, as empresas podem melhorar a qualidade dos dados e evitar as armadilhas dos dados de baixa qualidade.

Esta postagem do blog é apenas uma introdução à qualidade de dados com o Apache Spark . Há muitos outros tópicos a serem explorados, como **governança de dados** , **linhagem** e **segurança** .

Aqui nós apenas arranhamos a superfície. Posteriormente na série, aprofundaremos esses tópicos e exploraremos como o Apache Spark pode ser usado

para garantir a qualidade dos dados em diferentes domínios e setores. Na próxima postagem do blog, revisaremos os primeiros princípios de configuração da prática de qualidade de dados .

Veja também

- Qualidade de dados com ou sem Apache Spark e seu ecossistema (palestra)
- Privacidade de dados com Apache Spark (palestra)
- Estratégias para qualidade de dados com Apache Spark

Se você gostou de ler este artigo, há uma boa chance de que outros também gostem! Ao compartilhar este artigo nas mídias sociais usando os botões na parte superior da página, você está ajudando a espalhar informações valiosas para sua rede e, possivelmente, até mesmo além dela. Por favor, assine o feed RSS para ficar atualizado com o próximo conteúdo.

Todas as imagens do autor.

Engenharia de Dados ciência de dados Apache SparkGenericName

Engenharia de software

Follow

Escrito por Serge Smertin

127 Seguidores · Escritor paraMLOps como fazer

Programação extrema. Forense digital. Os pensamentos são meus.