Advanced Institute for Artificial Intelligence

DataOps

Advanced Institute for Artificial Intelligence – AI2

https://advancedinstitute.ai

Agenda

Agenda

- Desafios
- Processo
- ETL
- Datawarehouse
- Ferramentas

DataOps

Manifesto DataOps: https://www.dataopsmanifesto.org/

- Iniciativa para explorar técnicas de desenvolvimento contínuo ao campo de análise de dados
- Sob a perspectiva operacional do ciclo de vida dos dados para desenvolver aplicações diversas: relatórios, visualizações e modelos preditivos

DataOps

O processo de se obter valor de um conjunto de dados pode ser identificado por diferentes terminologias:

- Big Data
- Data Analytics
- Business inteligence

O DataOps se encaixa nessas iniciaticas como o conjunto de técnicas e metodologias que permite com que tais iniciativas ocorram de forma automatizada

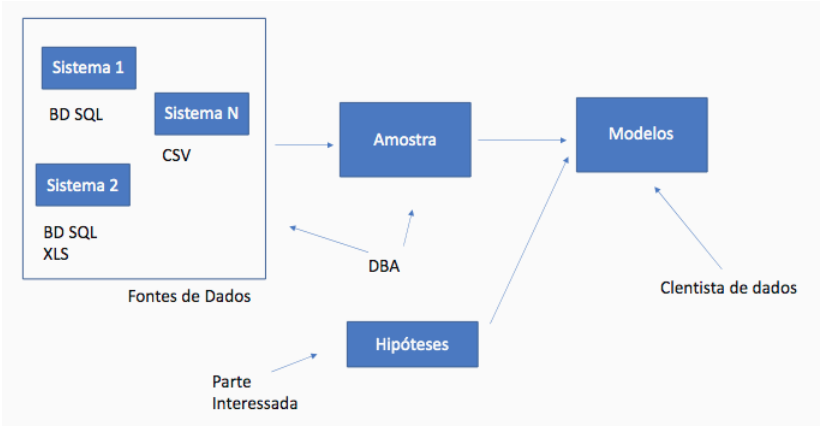
DataOps

Desafios

- Dados isolados
- Dados em diferentes formatos (SQL,csv, etc)
- Consumo de espaço e desempenho para criar filtros e consultas diversos
- Visão do dado combinando diferentes visões e agregações, para atender demandas de modelos e relatórios

Processo de Desenvolvimento

Trabalho colaborativo na construção de um modelo



Desenvolvimento Contínuo

ETL (Extract, Transform and Load) Extração Transformação e Carga é um processo para integração de dados de fontes distintas. A idéia é construir uma base de dados centralizada por meio de três passos:

- Extração dos dados de diferentes fontes
- Transformação dos dados para um formato que permita a análise conjunta dos dados
- Carga dos dados em um repositório com todas as informações em um único local

Desenvolvimento Contínuo

É comum utilizar ETL para diversos processos:

- Integrar dados de múltiplos sistemas, e obter uma visão unificada de um processo que passa por todos esses sistemas
- Integrar com dados de fontes externas
- Preparar os dados para uma análise específica
- Uniformizar formatos
  - Exemplo: usar sempre 0 e 1 para identificar masculino/feminino
- Usar ferramentas que permitam com que esse pré-processamento seja executado de modo eficiente

Desenvolvimento Contínuo

- Algoritmos de Aprendizagem de Máquina normalmente são treinados em um conjunto de dados preparado adequadamente
- É comum que a obtenção, preparação e gerenciamento dos dados seja feito por um profissional especializado nessa atividades (chamado engenheiro de dados), por ser um processo complexo e independente do desenvolvimento do modelo de aprendizagem de máquina
- Ferramentas que automatizam essa etapa são essenciais para permitir uma melhor integração entre o trabalho do engenheiro de dados e o cientista de dados
  - Essa integração é chamada de DataOps

Desenvolvimento Contínuo

O processo de desenvolvimento de modelo de aprendizagem de máquina de modo geral se baseia em um objetivo (normalmente definido em termos de uma predição) e um conjunto de dados

- A definição do objetivo e do conjunto de dados é desafiadora, normalmente é iniciada por um processo exploratório da base
  - A partir dai o objetivo pode ser refinado
  - Interações entre engenheiro de dados e cientista de dados por ser necessárias
- A exploração inicial dos dados é fundamental nesse processo de concepção de uma modelo, pois agrega entendimento do problema

Desenvolvimento Contínuo

DataOps

- uma metodologia automatizada com base no processo de desenvolvimento de algoritmos de aprendizagem de máquina
- Melhora a qualidade dos dados e reduz o tempo de análise de dados
- Basedo na metodologia ágil e no conceito de entrega contínua
- Diversas ferramentas permitem implementar esse processo
- É comum utilizarmos workflows para coordenadar a execução de rotinas para preparação de dados

Desenvolvimento Contínuo

Datawarehouse

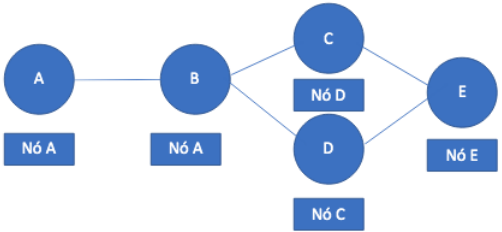
- Ferramentas para unificar diversas bases de dados em um único lugar
- Porque unificar bases em um único local?
  - Economia de desempenho e espaço na base da aplicação, que demanda desempenho
  - Otimizar o processo de fazer relatórios e busca em grandes conjuntos de dados
  - Linguagem padronizada para ferramentas de aprendizagem de máquina e relatório

Desenvolvimento Contínuo

Workflow (DAG - Directed Acyclic Graph)

- Um workflow é um técnica para representar uma aplicação como um processo composto por tarefas e uma ordem de execucao entre das tarefas
- Cada tarefa possui uma entrada e uma saída, permitindo assim controlar o processo de maneira mais ampla
  - Monitorar cada tarefa e recuperar em caso de problemas (não necessariamente a aplicação inteira)
  - Facilitar o uso de diversos recursos computacionais para executar um workflow (Escalonamento)
  - Executar partes de uma mesma aplicação simultaneamente

Desenvolvimento Contínuo



- Característica do Workflow de Exemplo:
- Diversos nós executam as tarefas do workflow
- Tarefa C e D podem ser executadas simultaneamente
- Dados de Saída das tarefas **C** e **D** precisam ser copiados para o recurso que executa a tarefa **E**
  - Nesse caso, a rede pode trazer impacto na transferência de dados
- Quando a aplicação é definida como um workflow é possível escalonar as tarefas com base em diferentes funções objetivo:
  - Melhorar desempenho, minimizar custo, facilitar reuso com outros workflows etc

Desenvolvimento Contínuo

Apache Airflow é uma ferramenta aberta de DataOps

- Permitir especificar workflows capazes de extrair informação de fontes de dados e aplicar transformações diversas para gerar um dataset adequado ao cientista de dados.
- As tarefas do workflow podem ser implementadas em diversas linguagens como o Python por exemplo
- Os workflows podem ser agendados para rodar em intervalos regulares
  - Dessa forma, é possível manter um dataset relativo a um processo que ocorre todo dia sempre atualizado
- A colaboração entre o engenheiro de dados e o cientista de dados pode entao ser completamente gerenciada pela ferramenta de DataOps Apache Airflow

Desenvolvimento Contínuo

A seguir a estrutura de um código que implementa um DAG em Airflow (Tarefas implementadas em Python)

```
1  from airflow import DAG
2  from airflow.operators.python_operator import PythonOperator
3
4  def prep_cliente():
5
6      print('task1')
7
8  def prep_cliente_perfil():
9
10     print('task2')
```

Desenvolvimento Contínuo

Criação do DAG

```
1
2  default_args = {
3      'owner': 'airflow',
4      'depends_on_past': False,
5      'email': ['airflow at example.com'],
6      'email_on_failure': False,
7      'email_on_retry': False,
8      'retries': 1
9  }
10
11 dag = DAG(
12     'prep_sicoob',
13     default_args=default_args,
14     description='DAG de preparacao de dados para Sicoob'
15 )
```

Desenvolvimento Contínuo

Referencia as tarefas

```
1
2 prep_cliente = PythonOperator(task_id='prep_cliente',
3                               python_callable=prep_cliente, dag=dag,)
4 prep_cliente_perfil = PythonOperator(task_id='prep_cliente_perfil',
5                               python_callable=prep_cliente_perfil, dag=
                                    dag,)
```

Sequência da execução das tarefas

```
1 prep_cliente >>  prep_cliente_perfil
```

Desenvolvimento Contínuo

- Ao terminar o desenvolvimento do DAG é necessário armazena-lo e executa-lo no terminal
- O DAG estará disponível para ser executado a partir da interface gráfico do Airflow ou a partir do terminal

HIVE

- Apache Hive é um data warehouse construído em cima do Apache Hadoop para fornecer consulta e análise de dados.
- Oferece uma interface semelhante a SQL HSQL(Hive SQL) para consultar dados armazenados em vários bancos de dados e sistemas de arquivos que se integram ao Hadoop.
- Dados são armazenados diretamente no Sistema de Arquivos Distribuídos Apache Hadoop (HDFS) ou outros sistemas de armazenamento de dados, como Apache HBase.

HIVE

- Funcionalidade: motor de consulta semelhante a SQL projetado para grandes volumes
- Tipo de processamento: processamento em lote usando Apache Tez ou Estruturas de computação MapReduce.
- Desempenho das consultas distribuídas pode ser superior a consultas em SGBDs
- Construído com base no modelo Map Reduce

Modelo de Dados Hive

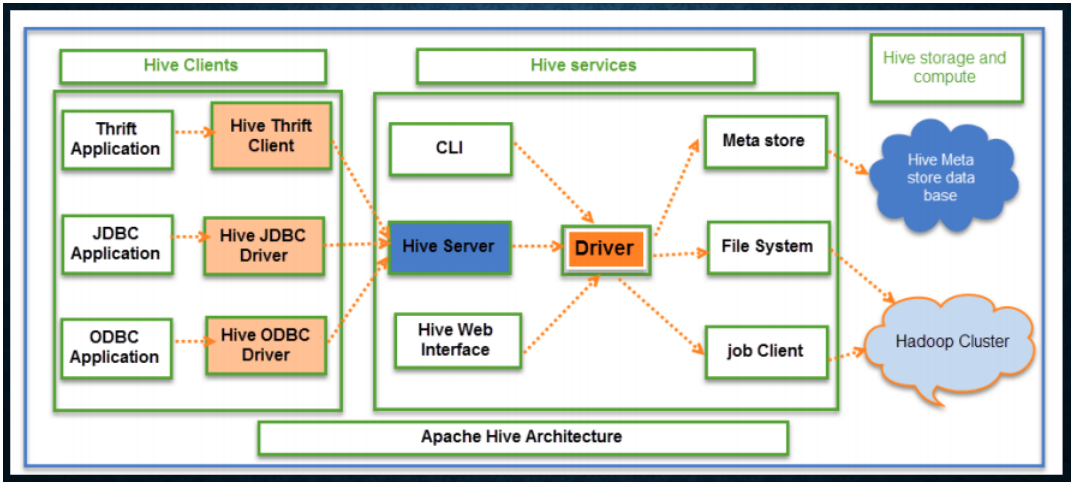
- Organizados em:
  - Tabelas
  - Partições
  - Buckets

HIVE

Análogo a tabelas relacionais

- Um banco de dados pode conter várias tabelas
- Cada banco/tabela possui um diretório correspondente em HDFS
- Uma tabela pode ser particionada
- Partições determinam a distribuição dos dados nos sub-diretórios
  - Exemplo: uma tabela de clientes pode possuir uma partição por cada estado

Hive


Hive Clients
Hive services
Hive storage and compute
Thrift Application
Hive Thrift Client
JDBC Application
Hive JDBC Driver
ODBC Application
Hive ODBC Driver
CLI
Hive Server
Hive Web Interface
Driver
Meta store
File System
job Client
Hive Meta store data base
Hadoop Cluster
Apache Hive Architecture

HIVEQL (HIVE QUERY LANGUAGE)

- Hive fornece uma CLI para escrever consultas Hive usando Hive Linguagem de consulta (HiveQL). A sintaxe HQL é semelhante à sintaxe SQL
- A linguagem inspirada em SQL do Hive separa o usuário do a complexidade da programação Map Reduce.
- Reutiliza conceitos familiares do mundo do banco de dados relacional, como tabelas, linhas, colunas e esquema