Turma: Ciência de Dados

Professor: PhD. Luciano Barbosa

Machine Learning Workflow e Monitoramento de Modelos

Professor convidado:
MSc. Arthur Caíque Bezerra Vieira

Data: 22/10/2020

Arthur Caíque

Contatos

E-mails: acbv@cesar.org.br / acbv2@cin.ufpe.br / arthurcaiquebv2@gmail.com

LinkedIn: https://www.linkedin.com/in/arthurcaique

Arthur Caíque

Formação Acadêmica

- Bacharel em Sistemas de Informação pela FACITEC/UPE.
- Mestre em Ciência da Computação pelo CIn/UFPE.
- Estudante de Doutorado em Ciência da Computação pelo CIn/UFPE.
- Áreas de interesse: Engenharia de Machine Learning, Processamento de Linguagem Natural e Recuperação de Informação.

Arthur Caíque

Experiência Profissional

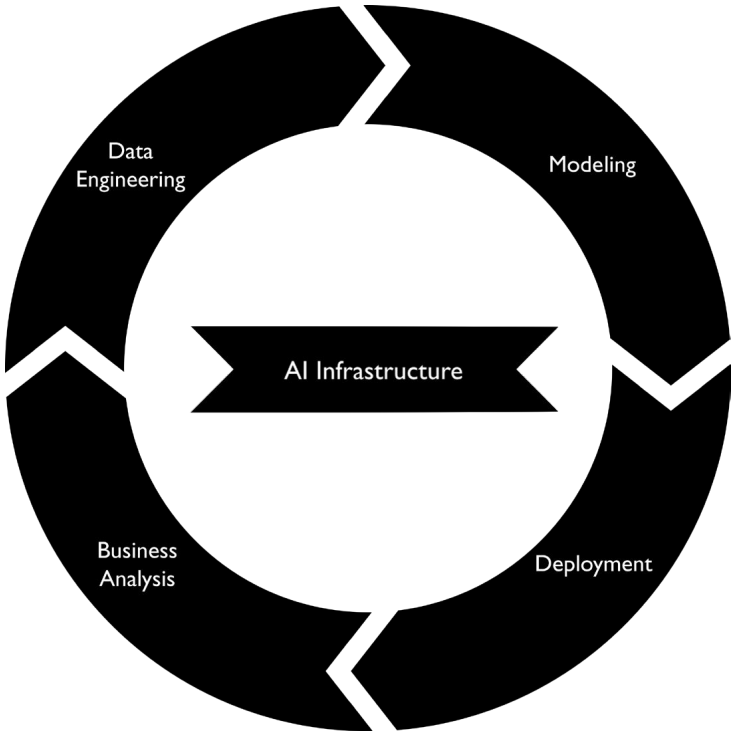
- Cientista de Dados/Engenheiro de Machine Learning há 3 anos.
- Empresas: Accenture e CESAR.
- Projetos: Nacionais e Internacionais.

Conteúdo

- Ciclo de Vida do Desenvolvimento de um Projeto de Inteligência Artificial
- Cientista de Dados vs Engenheiro de Machine Learning vs Engenheiro de Dados
- Machine Learning Workflow
- Ferramentas de Workflow de Machine Learning
- Demo DVC e MLflow

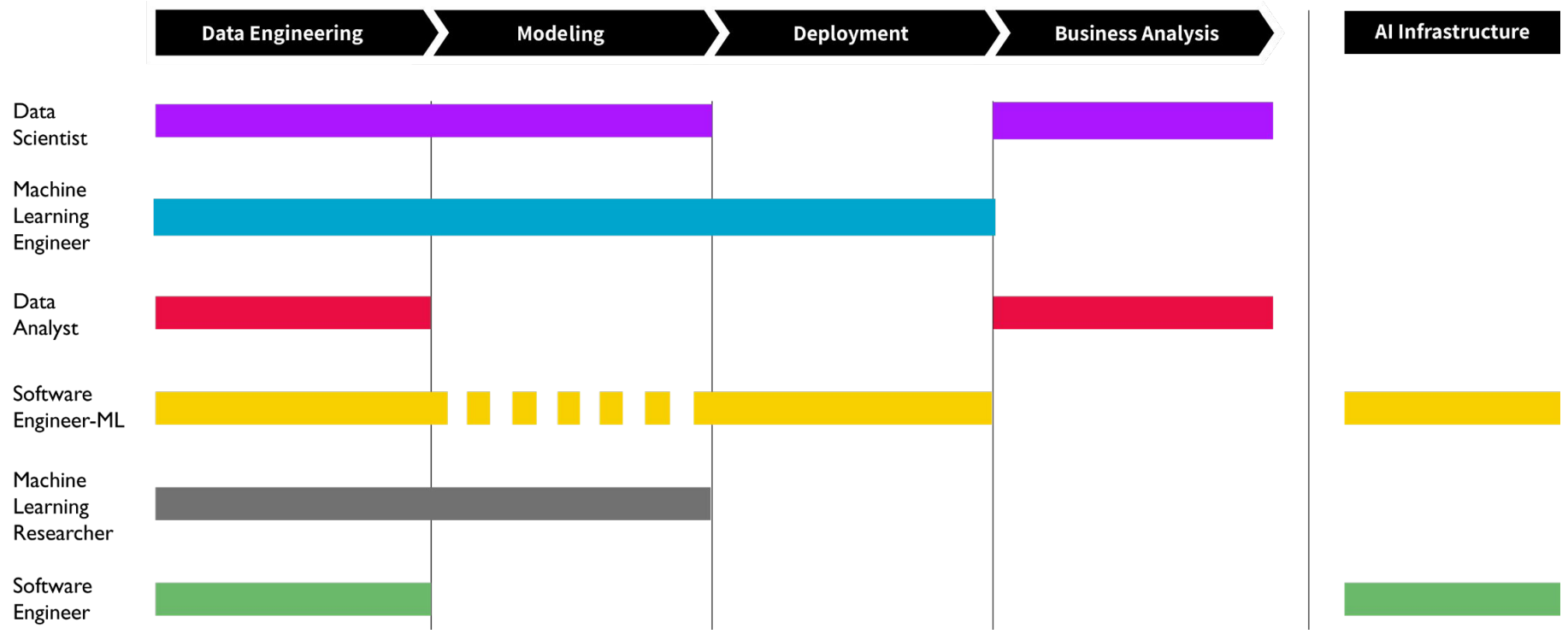
Ciclo de vida do desenvolvimento de um projeto de Inteligência Artificial

- **Processo iterativo** ao qual, primeiro, **dados são preparados** para a modelagem.
- Então, **modelos são treinados.**
- Para que, então, possam ser analisados o seu **valor para o negócio**.
- Enquanto isso, o time de **infra de IA** trabalha para tornar esse **ciclo mais eficiente**. (Workera, 2020)



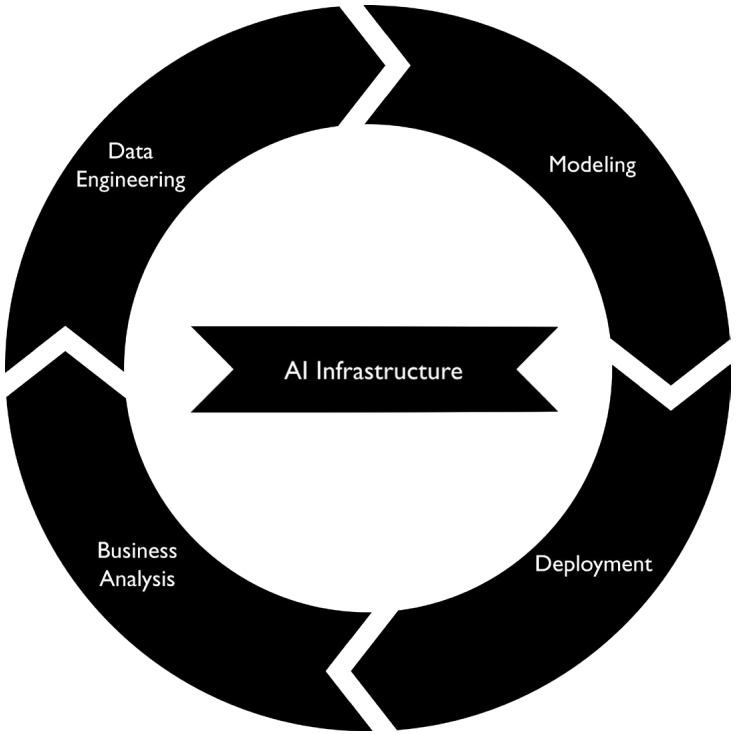
Ciclo de vida do desenvolvimento de um projeto de Inteligência Artificial

Workera, 2020



Ciclo de vida do desenvolvimento de um projeto de Inteligência Artificial

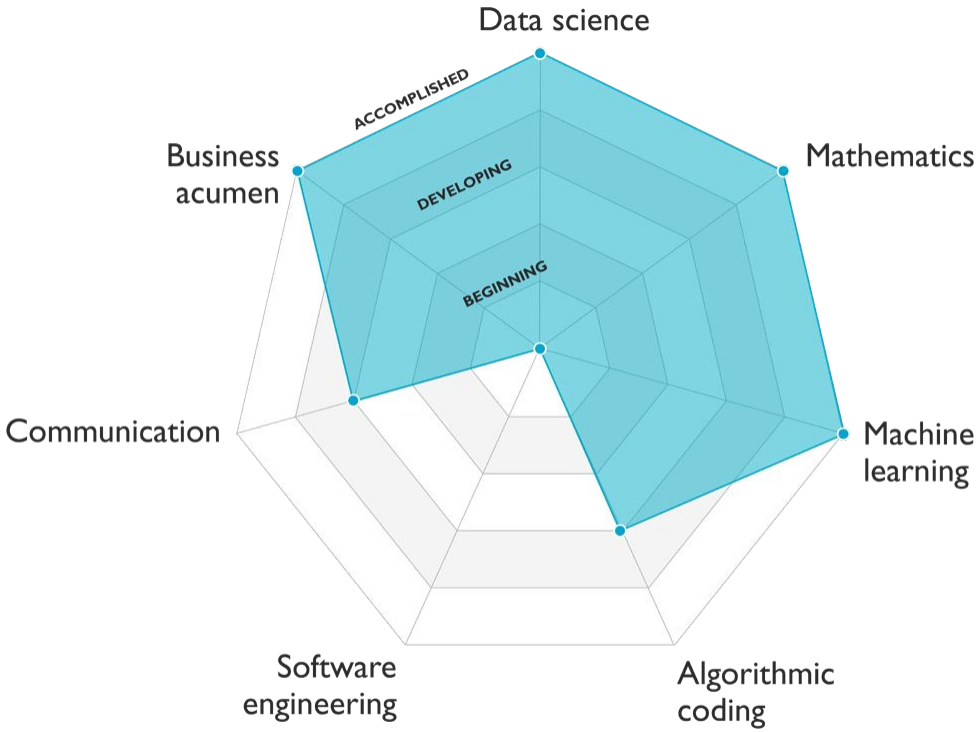
- Um **projeto de ML** começa com **dados sobre um serviço ou produto,** aos quais, **modelos são construídos** para serem **produtizados**.
- Esses modelos precisam ser **monitorados**, e o seu **desempenho avaliado**. (Workera, 2020).

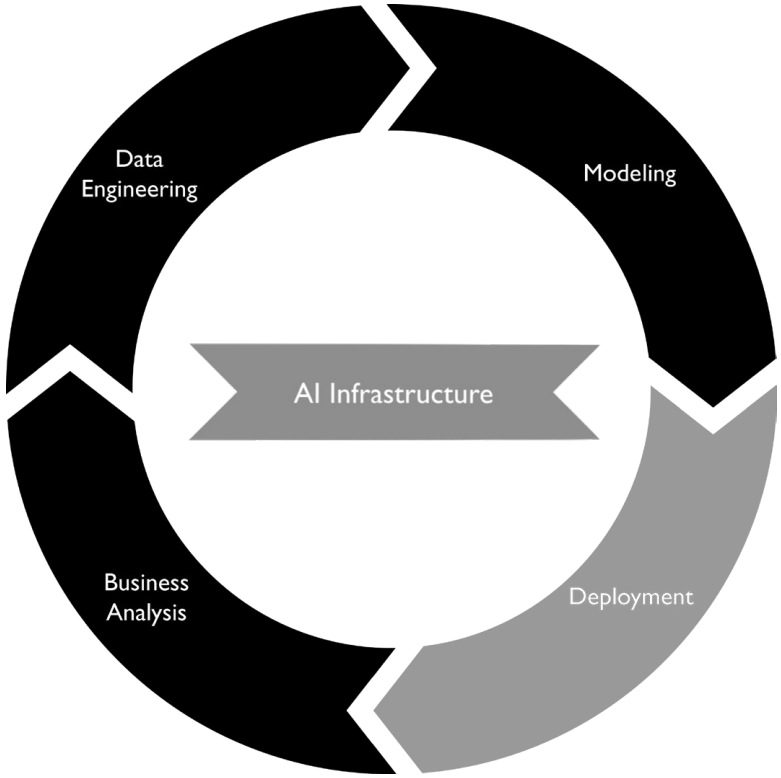


Ciclo de vida do desenvolvimento de um projeto de Inteligência Artificial

- Diversos **débitos técnicos** foram reportados na literatura e indústria sobre o ciclo de vida do desenvolvimento desses projetos (Sculley et al., 2015).
- Deve-se manter em mente que, em projetos dessa natureza, **bugs** podem ocorrer em 3 esferas: no **código**, no **dado**, no **modelo**.
- Portanto, **processos** e **ferramentas** vêm sendo desenvolvidos nos últimos anos para **mitigar a ocorrência de débitos técnicos** e **acelerar o desenvolvimento desses sistemas/projetos** de maneira mais **confiável**.

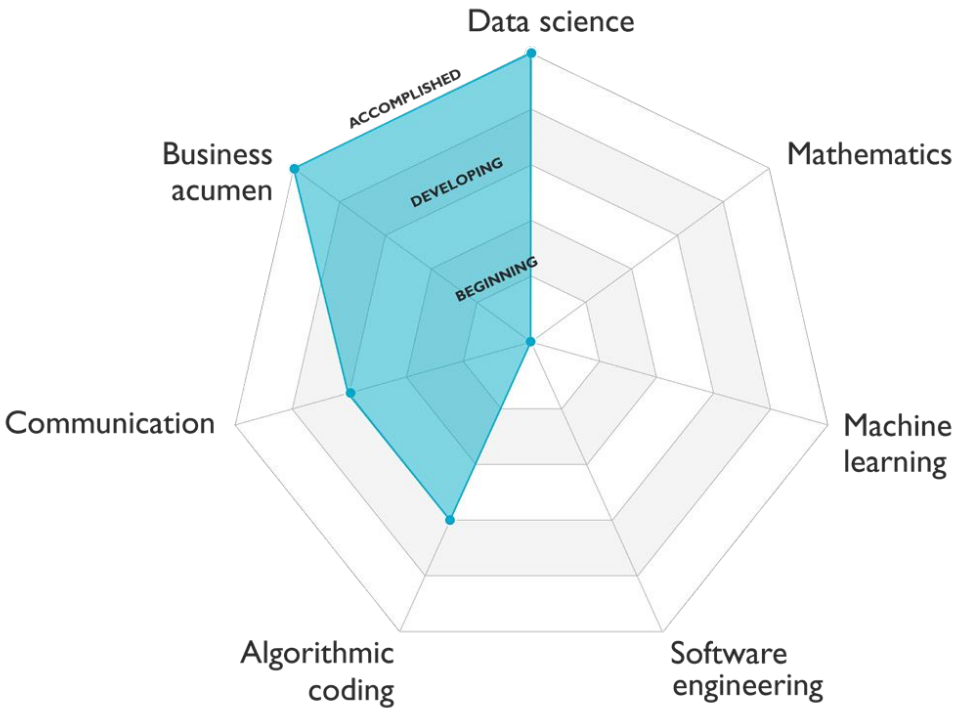
Papeis: Cientista de Dados

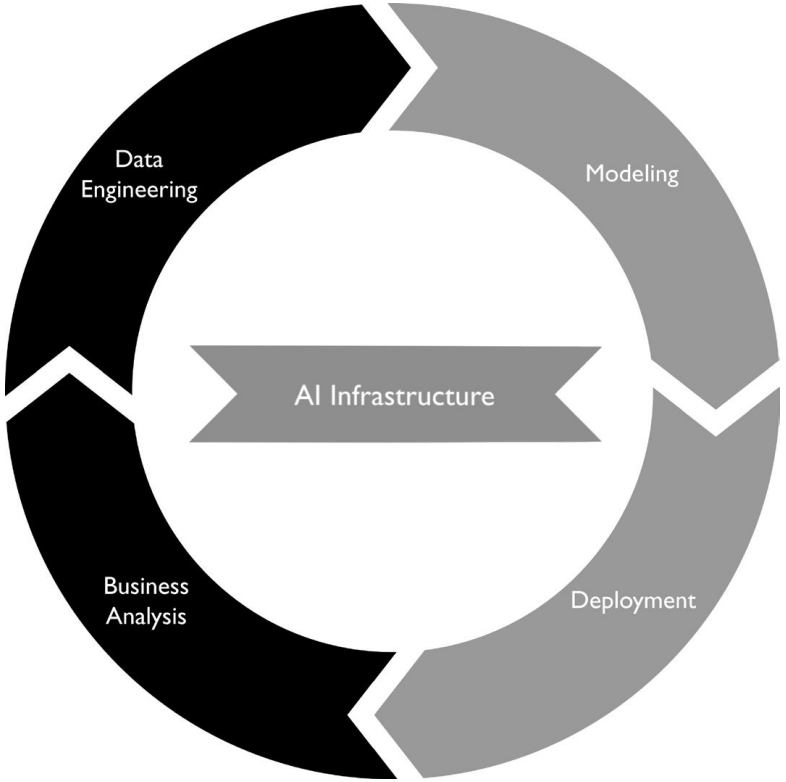




Workera, 2020

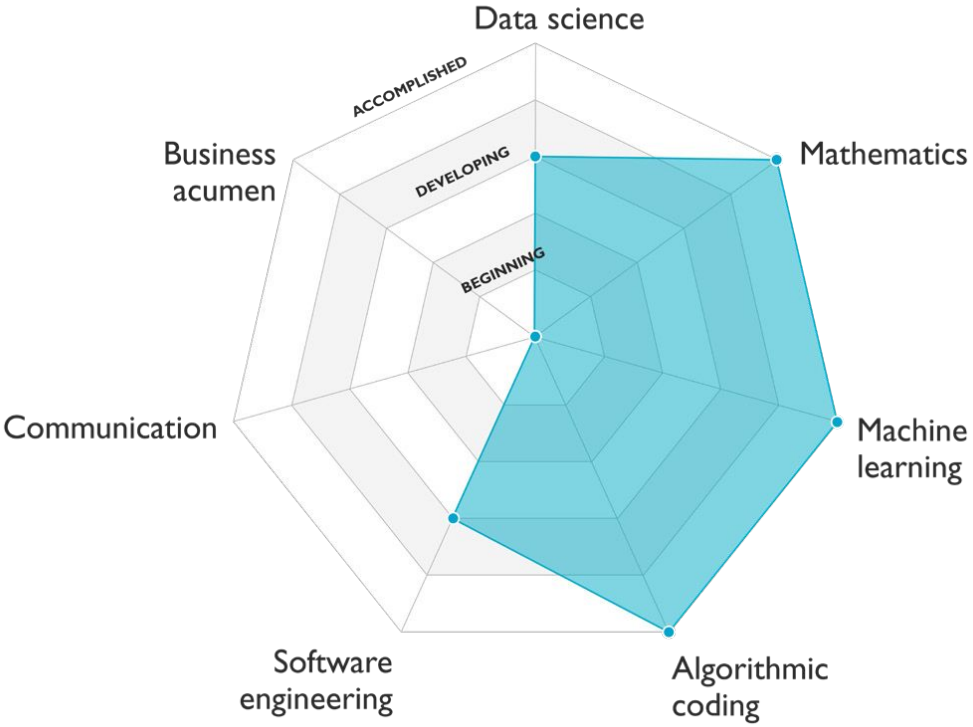
Papeis: Engenheiro de Dados

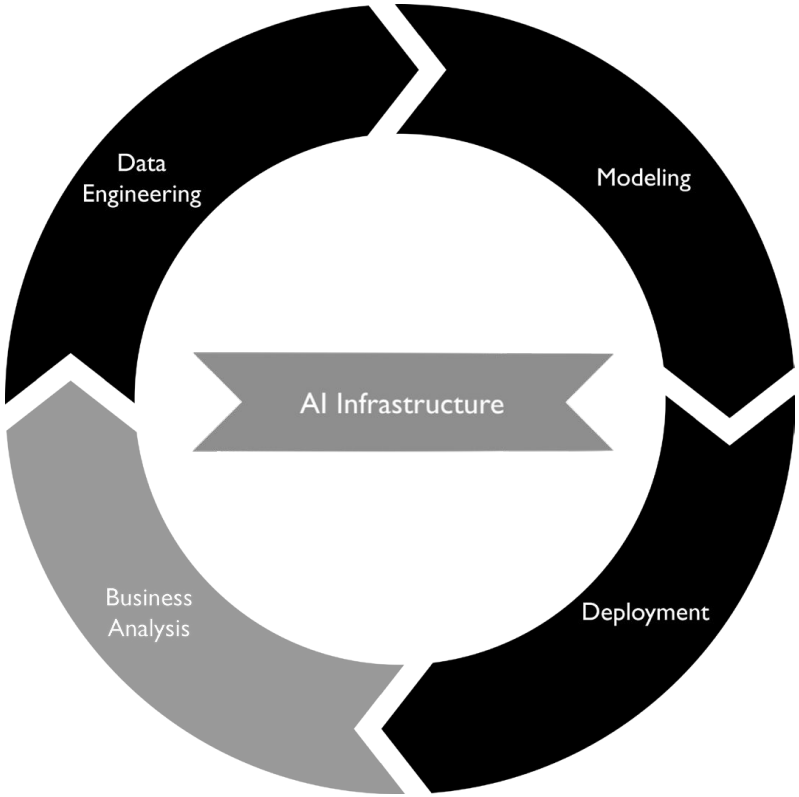




Workera, 2020

Papeis: Engenheiro de Machine Learning

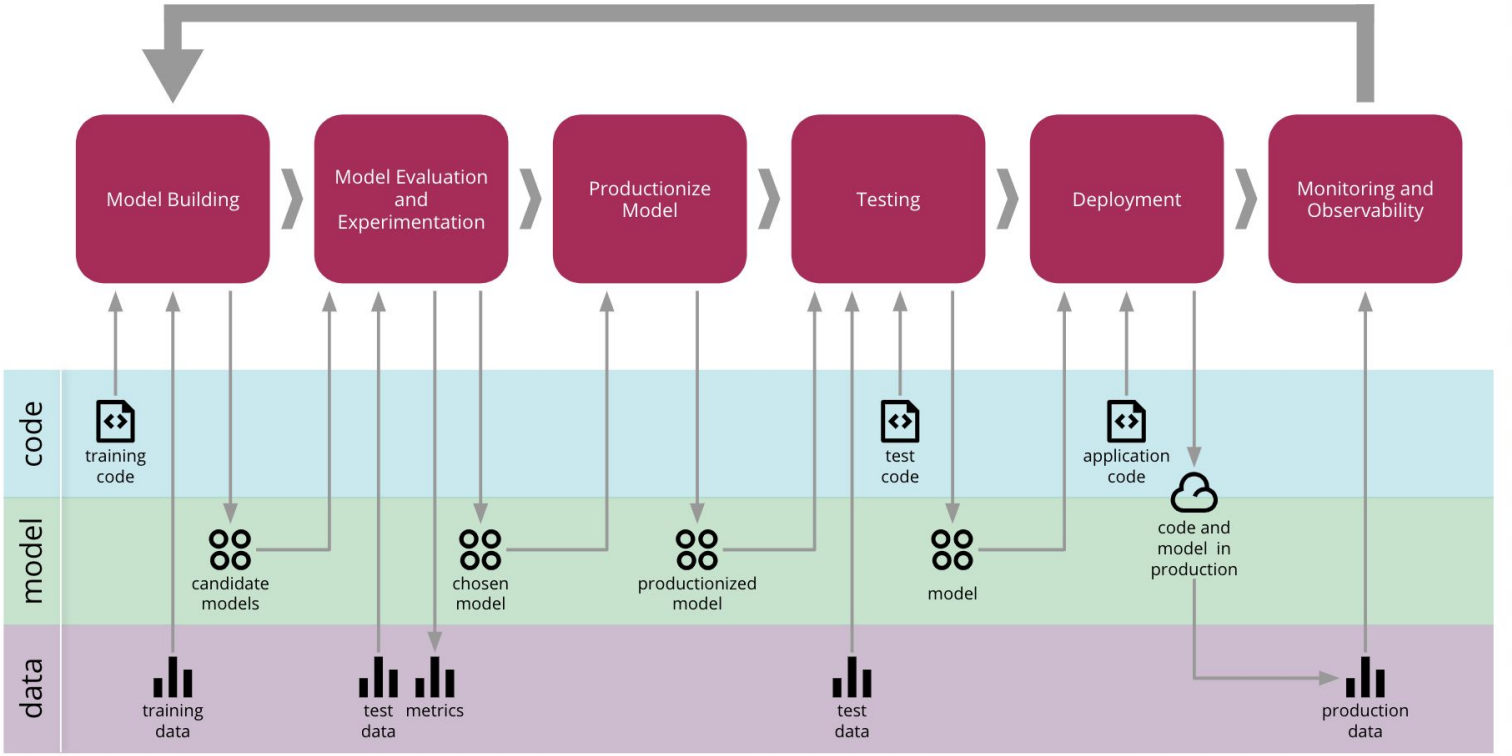




Workera, 2020

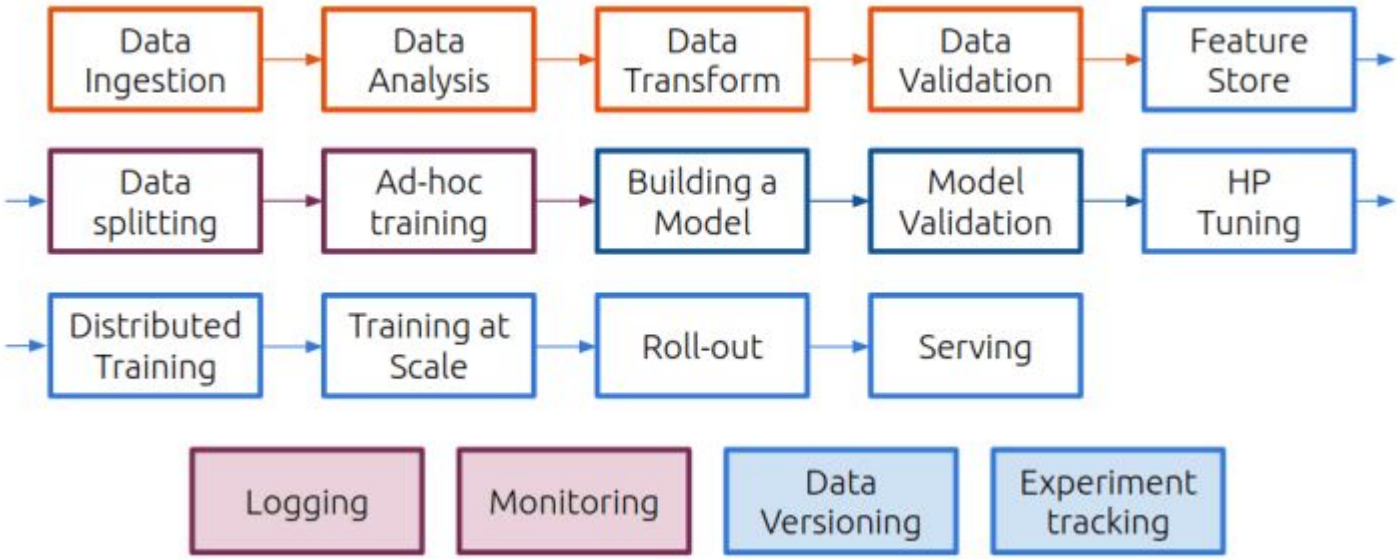
Machine Learning Workflow

- Continuous Integration of Machine Learning Systems (Fowler, 2019)



Machine Learning Workflow

- A Machine Learning Workflow (Rui Vasconcelos - Ubuntu, 2020).
- O que é importante observar é que a partir de cada uma dessas etapas, artefatos são gerados, e devem ser armazenados.



Directed Acyclic Graphs (DAG's)

- É uma coleção de todas as tarefas que devem ser executadas em uma pipeline, organizadas de uma forma que refletem as suas relações e dependências. (Airflow, 2020).
- A depender da ferramenta, uma DAG pode ser definida a partir de um script ou a partir de um arquivo yaml.

Directed Acyclic Graphs (DAG's)

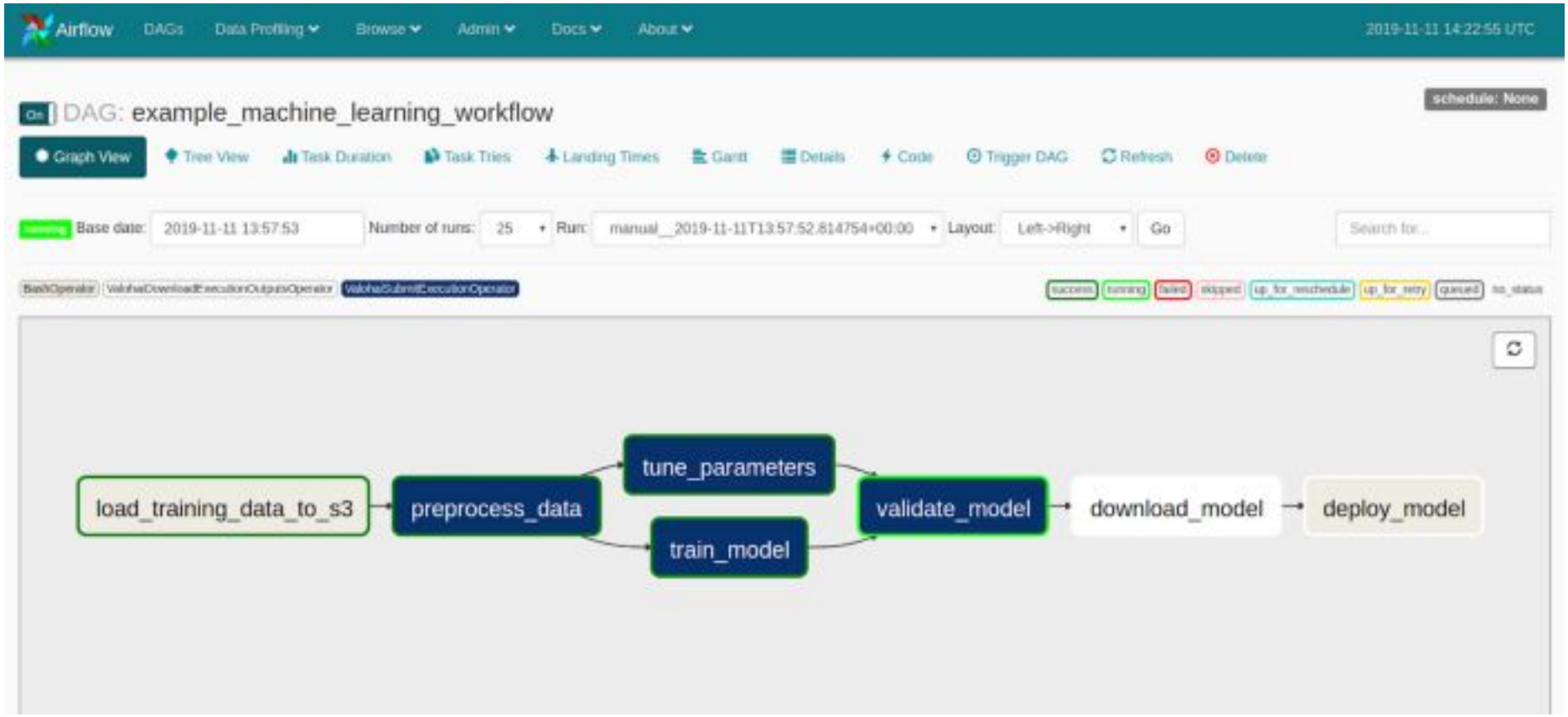


Imagem reproduzida do [Towards Data Science]

Kubeflow

- O projeto Kubeflow é dedicado a tornar simples, portável e escalável a produtização de workflows de Machine Learning no Kubernetes. O principal objetivo é fornecer uma maneira amigável de produtizar para diversas infra-estruturas. Onde quer que você esteja rodando o Kubernetes, você deveria ser capaz de executar o Kubeflow.

Kubeflow

- Possui serviços para criar e gerenciar jupyter notebooks interativos.
- Possui suporte nativo do Tensorflow para treinamento e deploy de modelos.
- **Kubeflow pipelines** para realizar o deploy e gerenciar workflows de ML. É possível agendar e comparar cada execução de experimentos.

Airflow

- Foi uma ferramenta criada para criar, agendar e monitorar workflows.
- **Escalável**: tem uma arquitetura modular e usa mensageria para orquestrar um número arbitrário de workers.
- **Dinâmico**: As pipelines são definidas em python, permitindo a geração dinâmica das mesmas.

MLflow

- Ferramenta *open source* para o ciclo de vida de *machine learning*
- Amplamente utilizada pela indústria.
- Fornece 4 componentes principais:
  - MLflow tracking
  - MLflow projects
  - MLflow models
  - Model registry

MLflow Tracking

- É uma API e UI para logar parâmetros, versões de código, métricas e arquivos de saída quando se executa um experimento de ML. E, posteriormente para visualização de resultados.

MLflow Projects

- Um MLflow Project é uma formato para empacotar código de data science de uma forma reproducível, baseado em convenções. Adicionalmente, esse componente inclui uma API e CLI para executar projetos, tornando possível encadeá-los em workflows.

MLflow models

- Um MLflow model é um formato padrão para empacotar modelos de machine learning que podem ser usados em ferramentas auxiliares, aplicações em tempo real ou inferência em batch pelo Apache Spark a partir de uma API REST. O formato define uma convenção que permite salvar um modelo em diferentes "sabores" que podem ser entendidos por diferentes ferramentas.

Model registry

- Esse componente é um repositório de modelos, conjunto de APIs, e UI, centralizados, para gerenciar colaborativamente o ciclo de vida de um MLflow Model. Ele fornece uma linhagem (qual experimento e execução produziu o modelo), versionamento de modelos, e transição de estados (de staging para produção), e anotações.

DVC

- Ferramenta *open-source* para **controle de versão** de projetos de *machine learning.*
- É uma ferramenta de gerenciamento de experimentos de *ml* que tira vantagem de um conjunto de ferramentas que já estamos familiarizados (Git, CI/CD, etc).
- DVC não veio para substituir o git, mas sim para ser usado junto com o mesmo.

DVC - Principais Funcionalidades

- Versionamento de dados
- Acesso a dados
- Pipeline de dados
- Experimentos
- *UI* em fase beta (https://viewer.iterative.ai/)

DVC - Principais Funcionalidades

- Versionamento de dados
  - Quão legal seria fazer o Git lidar com arquivos e diretório com uma quantidade msssiva de dados com o mesmo desempenho que com pequenos arquivos de código? Imagine fazer um `git clone` e ver seus dados e modelos de ML no seu workspace. Ou mudar para uma versão diferente de um arquivo de 100Gb em menos de 1 segundo com o `git checkout`?

DVC - Principais Funcionalidades

- Pipeline de dados
  - DVC torna fácil iterar on seu projeto usando git commits, tags ou branches. Você pode tentar diferentes ideias rapidamente, tunando parâmetros, comparando o desempenho com métricas e visualizá-los com plots.

Kedro

- É um framework open-source em python que aplica as melhores práticas de engenharia de software para pipelines de dados e machine learning. Você pode usá-lo, por exemplo, para otimizar o processo de levar um modelo de machine learning para o ambiente de produção. Você pode usar o Kedro para organizar um projeto de apenas um usuário em um ambiente local, ou colaborar dentro de um time em um projeto na indústria.

Kedro

Feature	What is this?
Project Template	A standard, modifiable and easy-to-use project template based on Cookiecutter Data Science.
Data Catalog	A series of lightweight data connectors used for saving and loading data across many different file formats and file systems including local and network file systems, cloud object stores, and HDFS. The Data Catalog also includes data and model versioning for file-based systems. Used with a Python or YAML API.
Pipeline Abstraction	Automatic resolution of dependencies between pure Python functions and data pipeline visualisation using Kedro-Viz.
The Journal	An ability to reproduce pipeline runs with saved pipeline run results.
Coding Standards	Test-driven development using `pytest`, produce well-documented code using Sphinx, create linted code with support for `flake8`, `isort` and `black` and make use of the standard Python logging library.
Flexible Deployment	Deployment strategies that include the use of Docker with Kedro-Docker, conversion of Kedro pipelines into Airflow DAGs with Kedro-Airflow, leveraging a REST API endpoint with Kedro-Server *(coming soon)* and serving Kedro pipelines as a Python package. Kedro can be deployed locally, on-premise and cloud (AWS, Azure and Google Cloud Platform) servers, or clusters (EMR, EC2, Azure HDinsight and Databricks).

ease.ml/ci

- Premissa: Modelos de *ml* são artefatos de *software*.
- Questão: Podemos testar continuamente modelos de *ml* da mesma maneira que testamos *software*?
- É um mecanismo de integração contínua desenvolvido para *ml*. Dado um novo modelo *commitado* no sistema, um conjunto de condições especificadas pelo usuário e uma base de teste, a ease.ml/ci testa se o modelo satisfaz todas as condições.

ease.ml/ci

- Um desafio técnico é *overfitting* --- após cada rodada de teste, o conjunto de teste perderá um pouco do seu poder estatístico. Se não tivermos cuidado, o modelo irá sobreajustar-se ao conjunto de teste fornecido, e a ease.ml/ci potencialmente poderia retornar uma resposta errada.
- O core da ease.ml/ci é uma coleção de técnicas para medir a "information leakage" que cada instância de teste pode trazer, e informar ao usuário quando um novo conjunto de teste é necessário.

Infra

- Docker / Kubernetes
  - Sugestão de disciplina do CIn (Microsserviços - Vinicius Cardoso: vcg@cin.ufpe.br)
- Cloud
  - Azure
  - GCP
  - AWS

Cursos de Engenharia de Machine Learning

- Full Stack Deep Learning Bootcamp, 2019
- Udacity: AWS - Machine Learning Engineer Nanodegree

Workera: Teste as suas skills

- https://workera.ai/?utm_source=deeplearning_ai&utm_medium=deeplearning_ai_website&utm_campaign=deeplearning_ai_nav

Perguntas?

Hora da Demo

Referências

- Workera: Career Pathways - Put Yourself in the Right Track
- Paper: Hidden Technical Debt in Machine Learning Systems - Sculley, 2015
- Código da Demo
- Continuous Delivery for Machine Learning - Fowler, 2019
- DAGs - Airflow, 2020
- Demystifying Kubeflow pipelines: Data science workflows on Kubernetes – Part 1 - Rui Vasconcelos, Ubuntu, 2020