



*Inatel*

# Esteira de implementação

## “Produtização”

Prof. Eng. Ranyeri do Lago Rocha  
e-mail [ranyeri.rocha@inatel.br](mailto:ranyeri.rocha@inatel.br)

***Inatel***

# Esteira de implementação Produtização

Todo o conteúdo deste documento está relacionado a direito autoral e é de circulação restrita, porquanto de propriedade exclusiva da Fundação Instituto Nacional de Telecomunicações (CNPJ 24.492.886/0001-04), protegido por força das disposições da Lei n.º 9.610/1998. A utilização deste material sem prévia e expressa autorização da proprietária constituirá infração à lei, com repercussões tanto na esfera civil quanto criminal.

# Esteira de implementação

- Introdução à Implantação de Modelos de IA
  - Pipelines de Aprendizado de Máquina
  - Empacotamento do modelo
  - Escolhendo a Infraestrutura de Deploy
  - Utilização de APIs com modelos de IA
  - Gerenciamento de Recursos
  - Garantindo a reprodutibilidade de um modelo utilizando Container
  - Monitoramento e Logging de Modelos

### Introdução à Implantação de Modelos de IA

Apesar de tudo que a IA, principalmente *Machine Learning*, se tornou nos anos recentes, pouca discussão é focada no chamado *Production Machine Learning*.

Trazer ML para produtos e aplicações! A área cobre **TUDO** o que vai além de simplesmente treinar um modelo de Aprendizado de Máquina.

É comum (muito comum) associar todos os conceitos de Aprendizado de Máquina no contexto acadêmico ou de pesquisa, onde tipicamente temos:

1. **Dataset**, geralmente já definido e organizado
2. **Treinamento e avaliação** dos resultados  
*Resultado*: modelo que faz uma boa predição

### Introdução à Implantação de Modelos de IA

Em *Production Machine Learning* somente o modelo não é suficiente.

Sobre “Produção”, as aplicações serão:

1. Implementadas
2. Mantidas
3. Melhoradas

Há diferenças consideráveis entre ML em ambiente não produção e ML em ambiente produção. Vamos aos detalhes...

## Introdução à Implantação de Modelos de IA

ML em ambiente não produção	ML em ambiente produção
Tipicamente, usa <i>dataset</i> estático	Dados reais, dinâmicos e com “ <i>shifting</i> ”
<u>Objetivo</u> : melhor acurácia sobre o <i>dataset</i>	<u>Objetivo</u> : <b>prioridades</b> . Latência, <i>fairness</i> , boa interpretabilidade, acurácia aceitável, custo...
Ajuste e treinamento para atingir um resultado ótimo	Monitoramento contínuo, assessment e novos treinamentos
Interpretabilidade e fairness	Interpretabilidade e fairness, reforçado!
<u>Desafio</u> : encontrar o modelo com melhor acuraria	<u>Desafio</u> : encontrar o modelo com melhor acurácia considerando todos os requisitos de sistema para operar o modelo em produção

### Introdução à Implantação de Modelos de IA

... considerando todos os requisitos de sistema para operar o modelo em produção:

1. Métodos de Pré-Processamento de dados
2. Setups para treinamento do modelo de forma paralelizada
3. Análise do modelo possível de repetição
4. Implementação do modelo de forma escalável

**Ao final:** atingir máximo desempenho ao custo mínimo.

- Monitoramento contínuo do desempenho do modelo, inserção de novos dados, novos treinamentos se necessário e novas implementações para manter ou melhorar o desempenho.

# Introdução à Implantação de Modelos de IA

## Pipelines de Aprendizado de Máquina

O modelo já está em produção...

...novos dados está disponíveis para treinamento.

validação dos dados > pré-processamento > treinamento > análise > implementação

Importante:

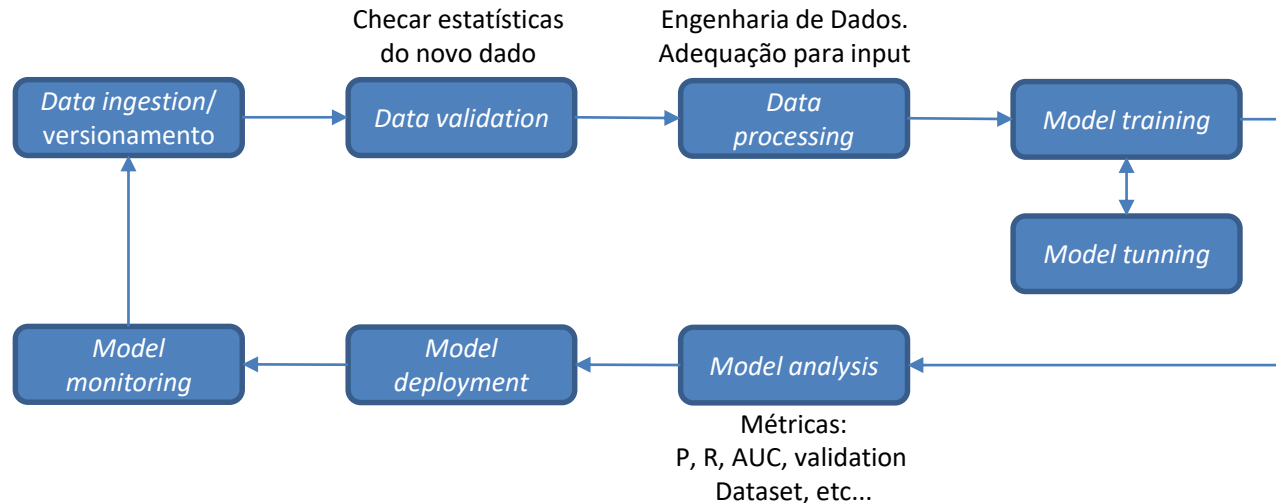
- Foco em novos modelos **vs** manter modelos existentes
- Se um modelo está em produção e em uso, um ML Pipeline é geralmente necessário.



## Introdução à Implantação de Modelos de IA

### Pipelines de Aprendizado de Máquina

#### Passos em um ML Pipeline



## Introdução à Implantação de Modelos de IA

### Pipelines de Aprendizado de Máquina

Passos em um ML Pipeline

*“Data changes often”*

- Mudanças graduais
  - Mudança nos dados ou mudanças no mundo que afetam os dados. Tendências, sazonalidade, importância relativa de uma *feature*.
- Mudanças repentinas
  - Problemas de coleta de dados e problemas de sistema (conexão, atualizações, etc)

Data shift vs Concept drift

# Introdução à Implantação de Modelos de IA

## Pipelines de Aprendizado de Máquina

Passos em um ML Pipeline

O processo de rotulagem: dados precisam ser rotulados!

- Rotulagem direta
  - Um dataset pode ser continuamente criado
- Rotulagem humana
  - Avaliadores examinam os dados e determinam os rótulos. A qualidade pode ser afetada!
  - Especialização ou expertise

## Introdução à Implantação de Modelos de IA

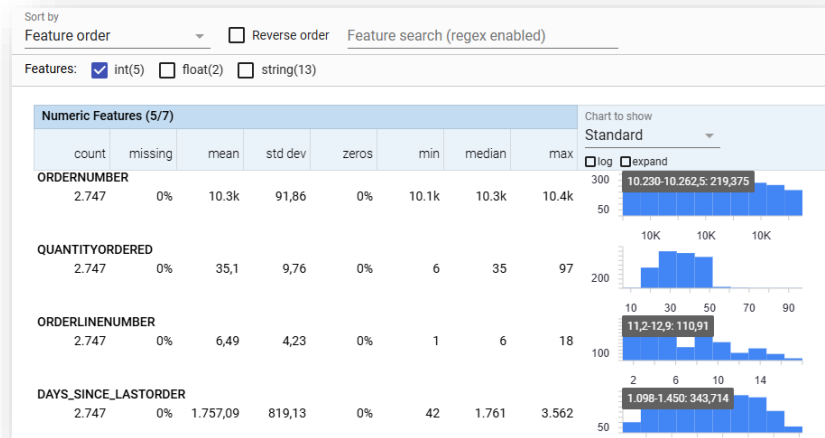
### Pipelines de Aprendizado de Máquina

#### Passos em um ML Pipeline

#### Processo de validação dos dados

- Validar grande quantidade de dados
- Manutenção da “saúde” dos pipes de ML em produção
- Entender o dado e problemas relacionados a ele

Uma ferramentas muito útil: **TFDV – Tensor Flow Data Validation**



## Introdução à Implantação de Modelos de IA

### Pipelines de Aprendizado de Máquina

#### Passos em um ML Pipeline

Na etapa de Data Processing em ML, a engenharia de dados e seleção de feature são essenciais, especialmente para treinamento do modelo.

**Importante:** o pré-processamento feito durante o treinamento precisa ser feito durante a inferência do modelo.

São operações:

1. Manipulação e limpeza de dados
2. Normalização
3. Codificação
4. Redução de dimensionalidade
5. Transformações em Imagem

## Introdução à Implantação de Modelos de IA

### Pipelines de Aprendizado de Máquina

#### Passos em um ML Pipeline

Na etapa de Data Processing em ML, a engenharia de dados e seleção de *feature* são essenciais, especialmente para treinamento do modelo.

**Texto:** uma classe de dados com uma quantidade grande de transformações para pré-processamento

- De forma geral, dados de texto devem ser convertidos em dados numéricos

Se dados categóricos, técnicas como **one-hot encoding**.

Se muitas categorias ou cada valor de texto é único, um **vocabulário com índice e frequência** pode ser usado

Se textos em NLP, técnicas como *embedding space*, *stemming and lemmatization*, TF-IDF e n-grams

# Introdução à Implantação de Modelos de IA

## Pipelines de Aprendizado de Máquina

### Passos em um ML Pipeline

Na etapa de Data Processing em ML, a engenharia de dados e seleção de *feature* são essenciais, especialmente para treinamento do modelo.

**Imagens:** Similar ao texto, com transformações importantes

Rotações, inversões, redução ou aumento de escala, redimensionamento, corte de uma área específica, borramentos, detecções de borda, ou outras distorções fotométricas.

# Introdução à Implantação de Modelos de IA

## Pipelines de Aprendizado de Máquina

### Passos em um ML Pipeline

Na etapa de Data Processing em ML, a engenharia de dados e seleção de feature são essenciais, especialmente para treinamento do modelo.

Em *Feature Selection*, o processo é feito para melhorar a qualidade do dado. Como? Determinando quais *features* do dado são relevantes para o aprendizado do modelo. Aumentar a qualidade do dataset!

São comuns: correlação, *Wrapper Method*, *Embedded Method* (regularização), *tokenization*...



## Introdução à Implantação de Modelos de IA

### Pipelines de Aprendizado de Máquina

#### TRAIN



#### INFERENCE



# Introdução à Implantação de Modelos de IA

## Empacotamento do modelo

Um modelo antes de ir para produção precisa ser exportado. Diferentes formatos de exportação impactam diretamente na **velocidade de inferência**, compatibilidade com **infraestrutura** e uso de hardware especializado, como **GPU** ou **TPU**.

São modelos comuns:

- ONNX – Open Neural Network Exchange (PyTorch, TensorFlow e Scikit-Learn)
- TorchScript (PyTorch)
- TensorRT (Otimizada para GPUs NVIDIA)

# Introdução à Implantação de Modelos de IA

## Empacotamento do modelo

### ONNX

- Interoperabilidade entre frameworks. Ampla compatibilidade.
- Suporta aceleração e Compatível com hardware especializado

### TorchScript

- Facilita inferência em dispositivos embarcados (mobile, IoT)
- Melhor desempenho que a inferência padrão do PyTorch

### Tensor RT

- Reduz latência e otimiza o desempenho para execução com GPU Nvidia

# Introdução à Implantação de Modelos de IA

## Escolhendo a Infraestrutura de Deploy

### *On-Premise vs Cloud*

A infraestrutura utilizada para implementação do modelo em produção envolve os recursos computacionais e a arquitetura usada para disponibilizar o modelo para consumo, por exemplo, usando API.

Para implementação *On-Premise*, ou Local, o modelo roda em servidores locais

- Data center privados
- Máquinas dentro da empresa

## Introdução à Implantação de Modelos de IA

### Escolhendo a Infraestrutura de Deploy

#### *On-Premise vs Cloud*

##### Vantagens *On-Premise*

- Controle total sobre hardware e segurança. São gerenciados e mantidos na empresa.
- Menor custo recorrente após a compra do hardware
- Baixa latência. Útil para aplicações em tempo real.
- Personalização, como ajuste fino de GPU, TPU, Rede, etc.

##### Desvantagem *On-Premise*

- Alto custo inicial para aquisição dos dispositivos
- Escalabilidade limitada
- Manutenção complexa e requer mão de obra especializada

# Introdução à Implantação de Modelos de IA

## Escolhendo a Infraestrutura de Deploy

### *On-Premise vs Cloud*

Mesmo com a implementação local, o acesso ao modelo para inferência se dá por meio de API.

#### Exemplo:

1. Um servidor com processamento adequado à aplicação (CPU ou GPU)
2. Modelo hospedado com *TensorFlow Serving* ou *Triton Inference Server*
3. API criada com ferramentas como *FastAPI* ou *Flask*
4. Monitoramento com Prometheus e Grafana.

Todos os dispositivos da empresa que necessitam de inferência acessam o modelo via API interna.

# Introdução à Implantação de Modelos de IA

## Escolhendo a Infraestrutura de Deploy

### *On-Premise vs Cloud*

A infraestrutura utilizada para implementação do modelo em produção envolve os recursos computacionais e a arquitetura usada para disponibilizar o modelo para consumo, por exemplo, usando API.

Para implementação *Cloud*, o modelo é hospedado (implementado) em um provedor de serviço de nuvem, como AWS, Google Cloud, Azure, etc). Pela definição do modelo *Cloud*, não há outra alternativa.

## Introdução à Implantação de Modelos de IA

### Escolhendo a Infraestrutura de Deploy

#### *On-Premise vs Cloud*

##### Vantagens *Cloud*

- Alta escalabilidade, sob demanda. Recursos dinâmicos.
- Menor complexidade de manutenção. Não há a necessidade de um time especializado na empresa. O provedor gerencia a infra
- Disponibilidade global com fácil acesso a partir de qualquer lugar

##### Desvantagem *Cloud*

- Custo recorrente. O uso pelos serviços de computação e tráfego é pago
- Latência variável com dependência com a localização do usuário
- Segurança e privacidade como um ponto de atenção



# Introdução à Implantação de Modelos de IA

## Escolhendo a Infraestrutura de Deploy

### *On-Premise vs Cloud*

A implementação do modelo na nuvem, pode assumir algumas formas:

1. Máquinas virtuais (IaaS) – Compute Engine, AWS EC2, Azure VMs...
  - Similar ao On-Premise.
2. Servidores de Inferência (Paas) – Google AI Platform, AWS Sagemaker, Azure ML
3. Serverless AI (FaaS) – Google Cloud Run, AWS Lambda, Azure Functions
  - Código executado sob demanda sem gerenciamento de servidores

# Introdução à Implantação de Modelos de IA

## Escolhendo a Infraestrutura de Deploy

### *On-Premise vs Cloud*

Em projetos reais, a necessidade de cada empresa/projeto define a abordagem.

Projetos com inferência sensíveis são executadas localmente.

Projetos com inferências escaláveis são executadas na nuvem.

# Introdução à Implantação de Modelos de IA

## Escolhendo a Infraestrutura de Deploy

A decisão: CPU, GPU ou TPU?

Aceleração de Hardware é o termo utilizado para se referir ao uso de hardware de computação especialmente feito para executar um conjunto de funções, como aceleração de I/O ou aceleração de cálculos matemáticos com ponto flutuante.

Um GPU ou TPU são desenvolvidos para acelerar operações matemáticas matriciais.

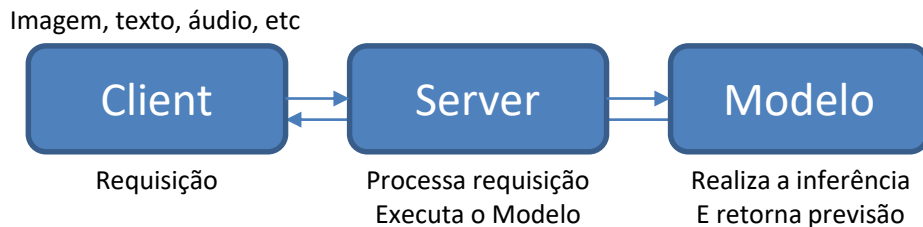
Uso destes hardwares define o **treinamento** e a **inferência** de um modelo muito mais rápido quando executado em um CPU de propósito geral.

## Introdução à Implantação de Modelos de IA

### Utilização de APIs com modelos de IA

APIs são utilizadas para disponibilizar um modelo de IA para consumo externo (ao ambiente local). São úteis para que diferentes sistemas interajam com o modelo.

No contexto da IA, uma API é usada para expor o modelo treinado para que outros sistemas possam utilizar (inferência) em tempo real.



# Introdução à Implantação de Modelos de IA

## Utilização de APIs com modelos de IA

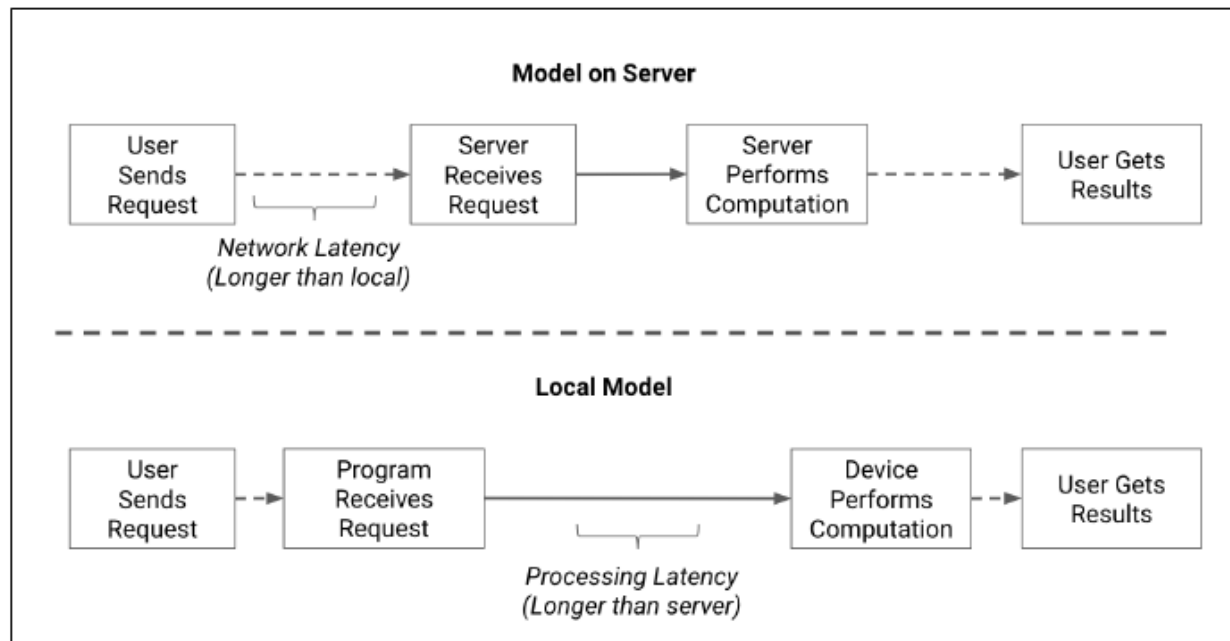
APIs são utilizadas para disponibilizar um modelo de IA para consumo externo (ao ambiente local). São úteis para que diferentes sistemas interajam com o modelo.

No contexto da IA, uma API é usada para expor o modelo treinado para que outros sistemas possam utilizar (inferência) em tempo real.

- Modelo treinado **implementado de forma centralizada**, acessado remotamente
- Modelo treinado **implementado em instâncias distribuídas** do seu modelo nos dispositivos próximos ao usuários final (mobile, *edge*, sistema embarcado)

## Introdução à Implantação de Modelos de IA

### Utilização de APIs com modelos de IA



## Introdução à Implantação de Modelos de IA

### Utilização de APIs com modelos de IA

#### Model Servers

Usuários do modelo necessitam fazer solicitações. Algumas ferramentas:

- Flask
- Django
- FastAPI
- TorchServer (PyTorch)
- TensorFlow Serving (TensorFlow)
- Kserve (Kubernetes)

# Introdução à Implantação de Modelos de IA

## Utilização de APIs com modelos de IA

### Model Serving

As formas de um “servir” a um modelo treinado e implementado são: batch ou real-time

Em aplicações onde atrasos são toleráveis, um modelo pode ser usado para fornecer previsões em “lote”.

Aplicações:

- Recomendação de produtos
- Análise de sentimento
- Previsão de demanda



## Introdução à Implantação de Modelos de IA

### Utilização de APIs com modelos de IA

#### Model Serving

As formas de um “servir” a um modelo treinado e implementado são: batch ou real-time

Em aplicações onde atrasos não são toleráveis, a solução é mais desafiadora.

Aplicar um modelo em tempo real implica em “espera” pelo cliente. Altos volumes de requisição e recursos de computação limitados agravam o cenário.

- *Target marketing*
- *Bidding for ads*
- *Food delivery times*
- *Autonomous driving systems*

# Introdução à Implantação de Modelos de IA

## Gerenciamento de Recursos

Os recursos requeridos para rodar seu modelo irá determinar quanto vai custar para colocar e manter seu modelo em produção.

Três áreas de interesse neste tópico:

1. Redução de dimensionalidade
2. Quantização de parâmetros de modelos e poda de modelos
3. Destilação de conhecimento para capturar conhecimento contido em *large models*

# Introdução à Implantação de Modelos de IA

## Gerenciamento de Recursos

### Redução de Dimensionalidade

Hoje, armazenar dados se tornou mais rápido, fácil e a custo menor. Resultado: os *datasets* armazenados são, geralmente, com alta dimensão. Lembre-se que dimensão está relacionado com a quantidade de *features* nos dados.

O acontece em uma rede treinada com um dataset que contém muitas features que são irrelevantes?

- Do ponto de vista dos pesos (parâmetros)
- E a consequência prática na utilização dos recursos

# Introdução à Implantação de Modelos de IA

## Gerenciamento de Recursos

### Redução de Dimensionalidade

Dados coletados em sistema de métricas de um veículo, com 50 sensores.

- Um dado 50-dimensional, de alta dimensão

Imagens 50x50 pixels

- Um dado que envolve 2500 dimensões se *grayscale*, ou 7500 se RGB.

# Aprendizado de Máquina é excelente para análise de alta dimensão! Muito superior à capacidade humana.

# Aumentar a dimensionalidade demanda mais recursos: mais dimensões, mais poder computacional e mais dados para treinamento são necessários.

## Introdução à Implantação de Modelos de IA

### Gerenciamento de Recursos

#### Redução de Dimensionalidade

**A pergunta de milhões:** Quantas *features* são ideais para um problema de ML?

A resposta: não existe!

O valor ótimo depende de vários fatores, como volume de dados de treinamento, variabilidade do dado, modelo específico que está sendo usado.

**Essencialmente 1:** dados suficientes, com as melhores *features*, variedade suficiente nos valores destas *features*, e informação preditiva suficiente nestas *features*.

**Essencialmente 2:** maximizar o desempenho do modelo enquanto o simplifica o máximo possível

# Introdução à Implantação de Modelos de IA

## Gerenciamento de Recursos

### Redução de Dimensionalidade

As abordagens possíveis são:

1. Seleção de *feature* manual
2. Algoritmo de seleção de *feature*
3. Algoritmo de redução de dimensionalidade
  1. O principal é o PCA – *Principal Component Analysis*

Não são  
mutuamente  
exclusivas!

## Introdução à Implantação de Modelos de IA

### Gerenciamento de Recursos

#### Quantização e Poda

**Objetivo:** criar modelos que sejam os mais eficientes e acurados possível visando atingir o melhor desempenho ao menor custo.

Quantização significa ter uma representação equivalente funcional do modelo, usando parâmetros e cálculos com uma precisão menor.

- Redução da precisão, significa utilizar menos bits. **Aumenta** velocidade de execução e eficiência, mas **reduz** acurácia geral do modelo.
- De floating point para inteiro, principalmente relacionado aos parâmetros do modelo!

# Introdução à Implantação de Modelos de IA

## Gerenciamento de Recursos

### Quantização e Poda

**Objetivo:** criar modelos que sejam os mais eficientes e acurados possível visando atingir o melhor desempenho ao menor custo.

A quantização pode ser feita durante o treinamento (*quantization-aware*)

A quantização pode ser feita no pós treinamento (*post-training quantization*).



## Introdução à Implantação de Modelos de IA

### Gerenciamento de Recursos

### Quantização e Poda

#### *Post-training*

Technique	Benefits
Dynamic range quantization	4x smaller, 2x–3x speedup
Full integer quantization	4x smaller, 3x+ speedup
Float16 quantization	2x smaller, GPU acceleration

Model	Top-1 accuracy (original)	Top-1 accuracy (post-training quantized)	Top-1 accuracy (quantization-aware training)
Mobilenet-v1-1-224	0.709	0.657	0.70
Mobilenet-v2-1-224	0.719	0.637	0.709
Inception_v3	0.78	0.772	0.775
Resnet_v2_101	0.770	0.768	N/A

Model	Latency (original) (ms)	Latency (post-training quantized) (ms)	Latency (quantization-aware training) (ms)
Mobilenet-v1-1-224	124	112	64
Mobilenet-v2-1-224	89	98	54
Inception_v3	1130	845	543
Resnet_v2_101	3973	2868	N/A

Model	Size (original) (MB)	Size (optimized) (MB)
Mobilenet-v1-1-224	16.9	4.3
Mobilenet-v2-1-224	14	3.6
Inception_v3	95.7	23.9
Resnet_v2_101	178.3	44.9

# Introdução à Implantação de Modelos de IA

## Gerenciamento de Recursos

### Quantização e Poda

O processo de poda de um modelo visa aumentar sua eficiência retirando partes que não contribuem substancialmente com os resultados.

*Biologicamente, nosso cérebro poda conexões redundantes ou irrelevantes constantemente. A quantidade de conexões neuronais e neurônios que temos hoje é drasticamente menor do que quando nascemos.*

**Objetivo 1:** menos parâmetros com menos conexões. Inferência mais rápida.

**Objetivo 2:** modelos mais complexos são mais passíveis de *overfitting*

## Introdução à Implantação de Modelos de IA

### Diferenças entre os ambientes: Desenvolvimento x Produção

Desenvolvimento	Produção
Treinamento e teste em notebooks	Modelos servindo aplicações reais
Dados estáticos e/ou limitados	Dados dinâmicos e fluxo contínuo
Experimentação sem preocupação com latência e escalabilidade	Escalabilidade, latência, disponibilidade e segurança!
Diferença nos ambientes (versão das bibliotecas, hardware, etc)	
Monitoramento para detectar problemas em tempo real	

# Introdução à Implantação de Modelos de IA

## Garantindo a reprodutibilidade de um modelo utilizando Container

Um modelo construído, treinado e testado, quando implementado, encontra diferentes hardwares e softwares. A padronização da implementação é difícil de ser garantida.

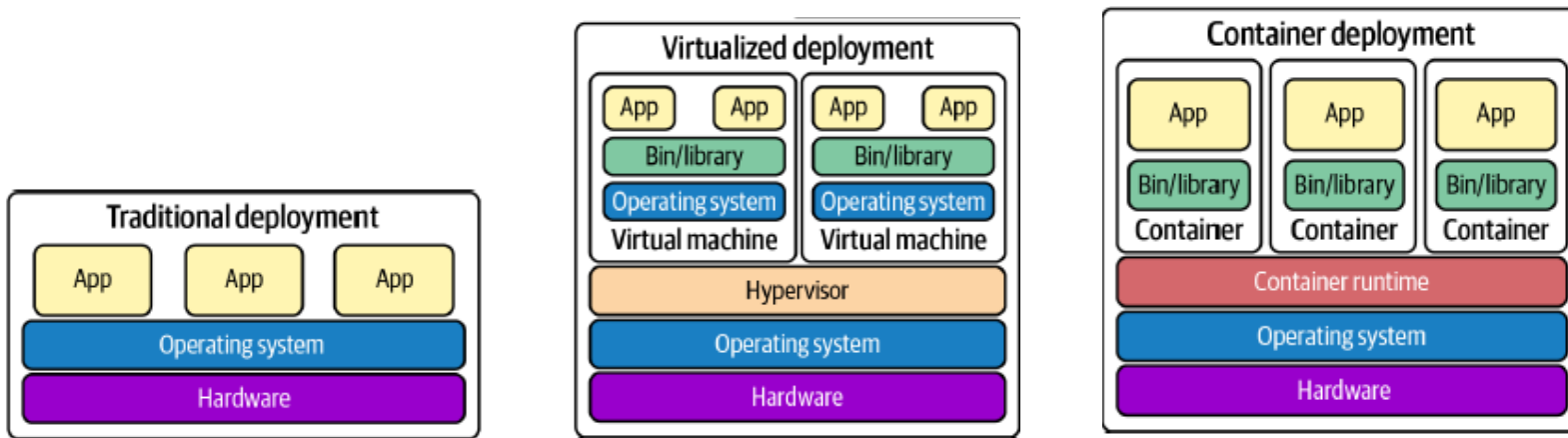
O modelo pode ser salvo e implementado utilizando um *container image* nos dispositivos *edge*, mesmo que com diferentes configurações.

Algumas vantagens:

- Compartilhamento de SO
- Não há diferenças, do ponto de vista da aplicação, estar em uma VM ou Container
- O *framework* Docker é largamente utilizado.

## Introdução à Implantação de Modelos de IA

**Garantindo a reprodutibilidade de um modelo utilizando Container**



# Introdução à Implantação de Modelos de IA

## **Garantindo a reprodutibilidade de um modelo utilizando Container**

A utilização de container evita problemas com compatibilidade e facilita a implementação em servidores e em ambiente cloud.

Os passos comuns para utilização do Docker em uma inferência pode ser sumarizado como:

1. Criar um Dockerfile com definições do ambiente (python version, bibliotecas, diretórios, API)
2. Construir uma imagem Docker
3. Rodar o container e fazer inferências

# Introdução à Implantação de Modelos de IA

## **Garantindo a reprodutibilidade de um modelo utilizando Container**

Assim como já visto alguns slides atrás, a implementação pode ser local ou na nuvem.

Para cenários local, a implementação do container pode ser feita através da ferramenta *Docker Compose*, geralmente distribuída junto à solução *Docker Desktop*.

Em cenários nuvem, a implementação do container pode ser feita em Google Cloud Run ou AWS Lambda, também citados para inferência PaaS.

# Introdução à Implantação de Modelos de IA

## Monitoramento e *Logging* de Modelos

Por que monitorar um modelo de IA?

1. Ao longo do uso, o desempenho do modelo pode se deteriorar devido a dinâmica dos dados
2. Novos padrões nos dados tornam o modelo obsoleto
3. Falhas no sistema podem levar à indisponibilidade da API
4. O tempo de inferência pode aumentar, impactando a experiência do usuário

O monitoramento contínuo é necessário para modelos em produção. Detecção de mudanças no desempenho garante a correta execução dos modelos ao longo do tempo.



# Introdução à Implantação de Modelos de IA

## Monitoramento e *Logging* de Modelos

### Tipos de Monitoramento de Modelos

1. Monitoramento de Desempenho
2. Monitoramento de Qualidade do Modelo
3. Monitoramento de Negócio

O que se pode analisar, de antemão, a partir dos tipos de monitoramento de modelos?  
Qual dos três tipos é o mais importante?

## Introdução à Implantação de Modelos de IA

### Monitoramento e *Logging* de Modelos

#### Monitoramento de Desempenho

**Latência ou velocidade de inferência:** tempo para processar uma solicitação

**Uso de CPU/GPU/RAM:** monitorar recursos computacionais utilizados

**Taxa de operações:** Apesar de fixo para o seu modelo construído é uma métrica importante

**Taxa de Erros:** identificar e quantizar falhas no sistema, como falhas de requisição

# definição: FLOPS – Floating-Point Operations Per Second

É usada para determinar o desempenho de um computador ou quantidade de operações que um modelo realiza!

## Introdução à Implantação de Modelos de IA

### Monitoramento e *Logging* de Modelos

#### Monitoramento de Desempenho

Nvidia T4  
65 TFLOPS

GeForce RTX 4090  
Até 191 TFLOPS

Model	size (pixels)	mAP <sup>val</sup> 50-95	Speed CPU ONNX (ms)	Speed T4 TensorRT10 (ms)	params (M)	FLOPs (B)
YOLO11n	640	39.5	56.1 ± 0.8	1.5 ± 0.0	2.6	6.5
YOLO11s	640	47.0	90.0 ± 1.2	2.5 ± 0.0	9.4	21.5
YOLO11m	640	51.5	183.2 ± 2.0	4.7 ± 0.1	20.1	68.0
YOLO11l	640	53.4	238.6 ± 1.4	6.2 ± 0.1	25.3	86.9
YOLO11x	640	54.7	462.8 ± 6.7	11.3 ± 0.2	56.9	194.9

# Introdução à Implantação de Modelos de IA

## Monitoramento e *Logging* de Modelos

### Monitoramento de Qualidade

Estas métricas estão alinhadas com o que foi visto no material “Modelos com Imagens”.

**Acurácia / Precisão / Recall:** monitoramento contínuo e identificação de quedas no desempenho

**Distribuição de dados de Entrada vs Treinamento:** este monitoramento indica *Drift* relacionado ao dado

**Distribuição das previsões:** apesar do dado de entrada não ter sofrido *Drift*, as inferências podem sofrer, indicando um *drift* do modelo

## Introdução à Implantação de Modelos de IA





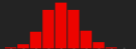



### Monitoramento e *Logging* de Modelos

#### Monitoramento de Qualidade

Ferramenta:

*Drift* – Evidently AI

Drift is detected for 100.0% of columns (4 out of 4).

Column	Type	Reference Distribution	Current Distribution	Data Drift	Stat Test	Drift Score
> target	cat			Detected	Z-test p_value	0.000002
> feature_numerica2	num			Detected	K-S p_value	0
> feature_numerica1	num			Detected	K-S p_value	0
> feature_categorica	cat			Detected	chi-square p_value	0

Rows per page: 4 rows |< < 1-4 of 4 > >|

## Introdução à Implantação de Modelos de IA

### Monitoramento e *Logging* de Modelos

#### Monitoramento de Negócio

Estas métricas não estão relacionadas “diretamente” com o modelo ou a implementação do mesmo. É uma análise de negócio, suportada pelo modelo.

#### **Impacto nas métricas de negócio**

- houve aumento de vendas devido a recomendações personalizadas?
- houve diminuição de atendimentos no call center, devido ao sistema de reconhecimento embarcado?
- houve...

## Introdução à Implantação de Modelos de IA

### Monitoramento e *Logging* de Modelos

Os logs detalhados de uma execução podem ser gerados/coletados utilizando algumas ferramentas específicas. A ideia é acompanhar os dados de desempenho durante a execução em produção. As métricas são coletadas em tempo real.

#### Prometheus + Grafana

São ferramentas open-source usadas para coletar, armazenar e consultar métricas em *endpoints HTTP*. Estas métricas são apresentadas em dashboards interativos (personalizados).

**As métricas comuns:** latência, uso de GPU, tempo de inferência

**Visualizações:** gráficos interativos

# Introdução à Implantação de Modelos de IA

## Atualização do modelo em produção

Caso o modelo tenha se degradado, no desempenho, novos dados mais recentes podem ser utilizados para um novo processo de treinamento. Uma nova versão do modelo é obtida e então utilizada em produção.

1. Monitoramento
2. Coleta de novos dados
3. Treinamento
4. Teste do novo modelo
5. Implantação

A etapa de implantação pode ser feita como um teste A/B, comparando os dois modelos em produção



*Inatel*



inatel



inateloficial



ascominatel



inatel.tecnologias



company/inatel

***Inatel***

Inatel - Instituto Nacional de Telecomunicações  
Campus em Santa Rita do Sapucaí - MG - Brasil  
Av. João de Camargo, 510 - Centro - 37540-000  
+55 (35) 3471 9200

Escritório em São Paulo - SP - Brasil  
WTC Tower, 18º andar - Conjunto 1811/1812  
Av. das Nações Unidas, 12.551 - Brooklin Novo - 04578-903  
+55 (11) 3043 6015