

Trecho	Código	Categoria
Bom. A gente trabalha na <<compahia>>, a gente trabalha com equipes bem interdisciplinares, né? E a gente trabalha - pelo menos no laboratório do Rio, hoje - com problemas muito aplicados a diferentes indústrias.	Equipes multidisciplinares	
	Problemas de diferentes indústrias.	
Então principalmente nessa parte, quando você tá trabalhando com ciência aplicada, você tá trabalhando em cima de um domínio - no geral que você não conhece necessariamente esse domínio.	Em geral desenvolvedores tem pouca experiência no domínio	Desafio [Geral]
Por exemplo: você pode estar trabalhando no domínio de meteorologia, no domínio de bioinformática, no domínio de petróleo e gás - que gente acaba necessitando muito do especialista do domínio.	Dependência do desenvolvedor com o especialista do domínio	Desafio [Geral]
	Presença do especialista do domínio	Ator [Geral]
	Presença do desenvolvedor.	Ator [Geral]
Então, por exemplo, a parte de Model Requirements, acaba sendo uma decisão colegiada né? Assim, juntam-se todos os especialistas em machine learning, especialistas de domínio, e no geral a gente vai fazer um modelo requerido pelos cientistas de domínio, então provavelmente os requisitos dos problemas serão trazidos por eles. Mas enfim, de quem partiu o problema vai ser levantado isso [os requisitos], e isso acaba tornando - essa parte vermelha [Model Requirement] - uma decisão mais colegiada.	Todos os envolvidos elicitam os requisitos do modelo em uma decisão colegiada	Atividade [Requisitos do modelo]
	Requisitos coletados de quem originou o problema.	Método [Requisitos do modelo]
Essa parte de Data Collection, Cleaning e Labelling, geralmente fica [com] o especialista de domínio. Porque enfim, ele que conhece o domínio do dado, ele que geralmente traz o dado, então geralmente fica com o cientista de domínio.	Processamento de dados executado pelo especialista do domínio	Atividade [Processamento de dados]
A parte de Feature Engineering, ela é meio complicado hoje, né? Porque a gente tá trabalhando, grande parte a gente pode trabalhar com... A gente pode fazer propriamente a Feature Engineering do dado, extrair os dados ou a gente pode aprender representações em cima desses dados né? Que entra nesse contexto de Deep Learning	Atributos podem ser coletados diretamente dos dados	Método [Engenharia de atributos]
	Atributos podem ter representações aprendidas nos dados.	Método [Engenharia de atributos]
No geral os modelos mais novos fazem a parte de Feature Engineering junto com Model Training, a gente monta lá o nosso modelo, a arquitetura do nosso modelo e faz tudo junto. Então seria uma coisa mais do machine learning engineer ou do data scientist, não sei qual o termo melhor para você usar, mas aqui ficaria mais o machine learning engineer.	Modelos mais novos fazem a engenharia de atributos e o treino de forma única	Método [Engenharia de atributos]
	Engenharia de atributos de modelos mais novos é feita pelo desenvolvedor.	Atividade [Engenharia de atributos]
No contexto de métodos tradicionais de machine learning, onde a gente tem que pegar feature mesmo, fazer a feature, aí eu acho que já é uma questão de colaboração. Ou essa feature vem inteira junto com o dado ou é uma feature combinada, do tipo, o cientista de domínio ele me aponta o que é importante e eu com os conhecimentos de machine learning vou obter aquela informação, seja ela uma imagem, fazendo, mudando, extraindo características da imagem, seja lá o que for. Então tem essas duas abordagens.	Modelos mais tradicionais não fazem a engenharia de atributos	Método [Engenharia de atributos]
Se for uma abordagem mais de apreender a representação do dado, então fica só para mim. Se for uma questão de ter que escolher uma feature fica com os dois, tanto o machine learning engineer quanto o especialista do domínio.	Utilizando modelos novos a engenharia de atributos é feita pelo desenvolvedor.	Atividade [Engenharia de atributos]
	Utilizando modelos tradicionais a engenharia de atributos é feita pelo desenvolvedor e o especialista de domínio.	Atividade [Engenharia de atributos]
	Treino e avaliação do modelo é feita na maioria dos casos pelo desenvolvedor	Atividade [Treino do modelo]

A parte do [Model] Training e Evaluation ela é senão totalmente, em grande parte feita

A parte do [Model] Training e Evaluation ela é senão totalmente, em grande parte feita só pelo machine learning engineer. Pode acontecer em situações específicas por a gente tá trabalhando com workflows complexos, workflows no sentido do problemas, né? Problemas complexos, que a gente possa necessitar do especialista do domínio para entender um certo comportamento que esteja acontecendo no [Model] Training. Ou tentar visualizar uma, uma, uma, uma circunstância que está acontecendo no [Model] Evaluation e para isso a gente pode recorrer a um especialista domínio, mas no geral essa é uma parte que fica para o machine learning engineer.	Especialista do domínio pode participar do Treino e Avaliação do modelo para opinar sobre algum comportamento que está ocorrendo.	Atividade [Treino do modelo]
Modelos mais tradicionais não fazem a engenharia de atributos	Deployment do modelos em soluções mais complexas é feita por equipes de infraestrutura.	Atividade [Implantação do modelo]
Mas também essa parte de deployment, dependendo da sua infraestrutura, se a gente pegar por exemplo o Watson Studio, ela é invisível, porque aí o machine learning engineer vai desenvolver a aplicação dele, o notebook dele, e vai carregar para o Watson Studio e o Watson Studio se encarrega do Deployment. Então se esse Deployment foi feito através de uma ferramenta como essa, fica só o machine learning engineer.	Deployment em plataformas para machine learnig é feita pelo desenvolvedor.	Atividade [Implementação do modelo]
Agora se você vai fazer um Deployment no cliente, em um cluster, em uma coisa mais com o arcabouço de produto, aí você vai precisar de um cara que entenda mais dessa questão de Kubernetes, Docker, etc. Que é esse pessoal de Devops.	Presença da equipe de infraestrutura.	Ator [Implantação do modelo]
	Equipes de infraestrutura precisam dominar tecnologias específicas.	Limitação [Implantação do modelo]
	Equipe de infraestrutura para implantação do modelo.	Atividade [Implantação do modelo]
Olha acaba sendo em duas vias, porque depende muito do problema que a gente tá trabalhando. Se eu tô trabalhando em problemas com ML, que são problemas que enfim, a gente não tá interessado em resolver "um problema real", a gente tá interessado em fazer pesquisa em desenvolver um novo método, novo modelo, a gente acaba trabalhando em todo processo.	Único desenvolvedor em todas as etapas para soluções cinéticas	Limitação [Geral]
Entretanto, [para soluções reais] por exemplo, a parte vermelha [Model Requirements] e a parte de dados [Data Collection, Cleaning and Labelling] já são dadas. Você tá trabalhando em um problema científico, onde já tenho dataset pronto, já tem o problema muito claro, você só tem que ir lá e fazer o modelo. Existem algumas questões que você precisa mudar um pouquinho, mudar alguma coisa no dataset, ou fazer uma variação nesse dataset, daí você acaba pintando um pouquinho no amarelo [Data Collection, Cleaning and Labelling]. No geral fica concentrado no azul [Feature Engineering], verde [Model Training] e roxo [Model Evaluation]. E na prática, se a gente tá trabalhando em um grupo plural como é esse da <<compahia>>, aí de fato a gente majoritariamente trabalha mais na, no azul [Feature Engineering], verde [Model Training] e roxo [Model Evaluation].	Desenvolvedor atua apenas na construção do modelo em casos de aplicações reais.	Limitação [Geral]
Por que as outras duas partes terão outros cientistas responsáveis por isso.	Para de aplicações reais etapas de dados e implantação são executadas por outras equipes.	Limitação [Geral]
O Model Training. Acaba sendo muito aquilo que eu falei, né? Se a gente tá trabalhando com aprendizagem de representação, o Model Training é o design do nosso modelo e acaba entrando nessa nessa caixinha. Então acho que é mais ou menos isso.	Entrevistado considera ter mais experiência no treino do modelo.	Limitação [Treino do modelo]
a gente desenvolve o código do modelo e utiliza a própria parte de treinamento dos frameworks o .train do TensorFlow, PyTorch. E aí você faz um cenário pequeno na sua máquina, você coloca só um teste, né? Acaba fazendo assim só para ver se tá tudo funcionando e depois você transfere provavelmente esse modelo para um ambiente capaz de rodar isso, pra um cluster de GPU e etc. Aí você configura os hiperparâmetros propriamente para o seu modelo, e você segue o mesmo processo lá de executar o comando de treinar e deixar ele rodando lá até treinar o modelo	Uso de frameworks para a criação de modelos: TensorFlow, PyTorch.	Método [Treino do modelo]
	Cenários menores para teste do treino	Método [Treino do modelo]
	Transferência do código de treino para infraestruturas maiores, capazes de executar.	Método [Treino do modelo]

Exatamente através da etapa seguinte, do Model Evaluation [...]. Aí você avalia com base em uma métrica, uma acurácia, uma precisão, um recall, depende do que foi definido para o problema.	Avaliação dá indícios da correteude do modelo	Limitação [Avaliação do modelo]
	Métricas clássicas utilizadas para avaliar o modelo.	Atividade [Avaliação do modelo]
	Uso de métricas clássicas.	Método [Avaliação do modelo]
Caso seja um problema complexo, novamente, um problema do mundo real, às vezes uma métrica, um precision, um recall não é suficiente. Você também vai precisar pegar esse resultado, compilar esse resultado e mostrar para cientista de domínio, para ele avaliar aquele resultado. Às vezes a precisão e o recall estão bons, mas tem outro aspecto do dado que o produto não foi interessante e só quem vai saber te dizer isso é o especialista de domínio.	Métricas clássicas podem ser insuficientes para avaliar aplicações reais	Limitação [Avaliação do modelo]
	Definir métricas adequadas para avaliar o problema.	Desafio [Avaliação do modelo]
	Algumas características do modelo podem só ser observadas pelo especialista.	Limitação [Avaliação do modelo]
	Especialista de domínio avalia o modelo.	Atividade [Avaliação do modelo]
é a parametrização desse modelo, a parametrização eu acho que é mais difícil depois do design da arquitetura, né?	Dentro a definição de hiperparâmetros é a etapa mais desafiadora.	Desafio [Treino do modelo]
Porque enfim, é extremamente empírico, você não tem não tem como medir. [Você pensa] "Ah eu vou botar aqui 0.5, porque ou é o que todo mundo usa ou é porque a sua experiência te disse que é assim". Mas no fim você acaba tendo que fazer uma variação de determinado conjunto de parâmetros, para achar o modelo [leia-se valores de hiperparâmetros] que melhor se adequa ao seu dado e você ter um produto melhor, claro, tomando cuidado para não ter overfitting e esse tipo de coisa.	A definição de hiperparâmetros é um processo empírico.	Atividade [Treino do modelo]
depende do experimento que você está executando, né? No geral você faz um cross-validation, um team cross-validation, alguma coisa assim, se você tá fazendo um experimento mais aprofundado, onde você quer resultados mais confiáveis.	Cross-validation para resultados mais confiáveis.	Método [Avaliação do modelo]
Ou você faz só um split aleatório dos seus dados, divide lá em 60% treino, 20% teste, 20% avaliação. Você faz alguma métrica [leia-se divisão] simples dessa, mas sempre aleatório.	Separação da base de treino, teste e validação em proporções tradicionais para soluções gerais.	Método [Avaliação do modelo]
	Separar os dados em treino, teste e validação.	Atividade [Treino do modelo]
essa [divisão em] 60%, 20% e 20% é uma das mais clássicas, que todo mundo usa. [Sendo que] a seleção aleatória é fundamental, a seleção inicial aleatória no conjunto total de quais serão os 60% tem que ser aleatória, entendeu? Quem serão os 60%? Quem serão os 20? E quem serão os outros 20%? [Entretanto] Essa proporção varia muito do dado, a clássica é 60%, 20% e 20%.	Seleção aleatória de treino, teste e validação é fundamental.	Método [Avaliação do modelo]
[Essas abordagens] ficam para soluções mais científicas, se a gente tá em um teste inicial, no primeiro momento você faz a mais simples, mas depois "terminamos, vamos escrever o paper", aí você faz um cross validation.	Cross-validation apenas ao fim para dar mais confiabilidade.	Limitação [Avaliação do modelo]
geralmente o dado que você usa para treino, ele vêm do cenário real, né? Ou algo muito próximo do cenário real. O que você pode fazer no processo de Data collection é fazer avaliações estatísticas em cima do seu dado, né? Você analisa a distribuição dos dados e etc. No treino do modelo você já não tem muito mais o que fazer, o dado já tá pronto, você só vai dividir ele.	Dados de treino oriundos do mundo real	Método [Processamento de dados]
	Estatística usada para verificar a representatividade do dado.	Método [Engenharia de dados]
	Uso de estatística na engenharia de atributos.	Atividade [Engenharia de atributos]

<p>Você tem que assumir que o dado já segue uma distribuição uniforme, porque você vai fazer uma seleção aleatória, agora se você sabe de antemão que o seu dado não segue uma distribuição aleatória que, você tá no momento de treino e ainda tem que fazer alguma coisa então, você faz um data augmentation, um sampling do seu dado, pra você tentar mitigar [a possibilidade de] que o seu modelo gere algum bias em relação ao mundo real, entendeu? Então você vai deixar o seu dado o mais uniforme possível e você assume que ele vai refletir o mundo real.</p>	<p>No treino se assume que o dado é representativo para o mundo real</p>	<p>Limitação [Treino do modelo]</p>
<p>então, o algoritmo eu acho que ..., eu acho que ambos são muito na intuição, né? Você tem um problema, você estuda o seu problema, você vê o que a maioria das pessoas estão desenvolvendo e você desenvolve em cima de um algoritmo similar ou alguma ideia próxima. Se ficar no contexto de pesquisa, beleza, é o que a imaginação permitir.</p>	<p>Uniformizar o dado de treino de modo que se assuma que ele reflete o mundo real.</p>	<p>Método [Engenharia de dados]</p>
<p>A questão de hiperparâmetros, ela é muito como eu falei antes, né? Ela é muito experimental, não tem muito como você definir. Você sabe um conjunto básico de parâmetros, você sabe que uma determinada distribuição de dados [leia-se parâmetros] funciona por exemplo, no contexto de embeddings [onde] a gente usa uma distribuição de dados um pouquinho diferente da uniforme, porque alguém fez um teste um dia e viu que ela é melhor do que criar todos os vetores aleatoriamente. A gente utiliza essas coisas porque enfim, acelera o processo. Mas é extremamente experimental, pode acontecer de amanhã eu achar uma outra distribuição inicial que enfim, ajuda a treinar mais rápido meu modelo, que ajuda ele a convergir mais rápido e que foi descoberta a partir de pura experimentação.</p>	<p>Literatura define pontos de partida para algoritmos e hiperparâmetros</p>	<p>Limitação [Treino do modelo]</p>
<p>A questão de hiperparâmetros, ela é muito como eu falei antes, né? Ela é muito experimental, não tem muito como você definir. Você sabe um conjunto básico de parâmetros, você sabe que uma determinada distribuição de dados [leia-se parâmetros] funciona por exemplo, no contexto de embeddings [onde] a gente usa uma distribuição de dados um pouquinho diferente da uniforme, porque alguém fez um teste um dia e viu que ela é melhor do que criar todos os vetores aleatoriamente. A gente utiliza essas coisas porque enfim, acelera o processo. Mas é extremamente experimental, pode acontecer de amanhã eu achar uma outra distribuição inicial que enfim, ajuda a treinar mais rápido meu modelo, que ajuda ele a convergir mais rápido e que foi descoberta a partir de pura experimentação.</p>	<p>A definição de hiperparâmetros é um processo empírico.</p>	<p>Limitação [Treino do modelo]</p>
<p>sim, a gente sempre compara com baseline sim. Aí no contexto de ciência tem os baselines típicos, né? Que são os estados da arte, se você está construindo um novo modelo, você compara com esses baselines, tentando bater o estado da arte.</p>	<p>A literatura define os baselines para soluções científicas.</p>	<p>Limitação [Treino do modelo]</p>
<p>Em um domínio de mundo real já é mais complicado, porque não necessariamente alguém já trabalhou naquele problema. Se você tá na indústria no geral, ninguém fez aquela solução e aí você tem os próprios baselines, você pega soluções que você sabe que são soluções tradicionais para aquele problema, implementa, toma isso como base e aí desenvolve novas abordagens, comparando com esses baselines.</p>	<p>Aplicações reais possuem próprios baselines que devem ser buscados para comparação.</p>	<p>Método [Treino do modelo]</p>
<p>já surgiu oportunidade mas nunca usei não, porque demora muito.</p>	<p>Otimizadores de hiperparâmetros consomem muito tempo.</p>	<p>Desafio [Treino do modelo]</p>
<p>assim, não necessariamente manualmente, porque a gente faz a experimentação, por exemplo, define lá um range específico. Mas aí o range quem define é a gente, então essa parte é manual. Varia em cima de tudo aquilo [range] mas é manual.</p>	<p>Variação de hiperparâmetros é feita dentro de uma extensão de valores.</p>	<p>Limitação [Treino do modelo]</p>

enfim, eu nunca trabalhei com problemas que eu tive isso como resultado, então eu nunca resolvi um problema desse. Mas no geral quando você tem problemas que vê que são da natureza do dado, você tem que mudar o seu conjunto de dados.	Problemas da natureza dos dados são resolvidos reexecutando o processamento de dados para obter um novo conjunto	Método [Processamento de dados]
[Supondo que] Você fez um modelo para uma série temporal e essa série temporal muda de tempos em tempos, por exemplo: nenhum modelo temporal vai funcionar agora durante a época do coronavírus, porque enfim, não se teve quarentena desde a gripe espanhola. Em um caso desses, ou você sabe que o seu modelo tem essa natureza, que é um modelo temporal, então você vai tomar abordagens de aprendizagem online, aprendizagem por reforço, etc.	Soluções com características temporais requerem tipos específicos de aprendizagem.	Limitação [Treino do modelo]
[...] Ou você assume que seu modelo - que aí já aconteceu comigo na prática em um problema de indústria - é temporal mas pode ser estacionário no sentido de que, eu posso atualizar esse modelo anualmente e não vai ferir nada. Então [...] é interessante você treinar o seu modelo em batches, de tempos em tempos. Você treina o modelo, e faz o deployment do novo modelo, que é uma solução um pouco mais custosa.	Modelos estacionários e temporais devem ser retreinados com certa frequência.	Restrições [Treino do modelo]
Você pode utilizar uma metodologia mais técnica. "Ah eu sei que meu modelo gira em torno de uma determinada [...] métrica", e ele consistentemente foge [dos valores esperados], aí você pode fazer uma distribuição normal [...] se você vê que a pontuação do seu modelo naquela métrica muda, isso pode indicar [...] uma necessidade de retreinar.	Métricas podem ser utilizadas para monitorar um modelo implantado	Método [Implantação do modelo]
Mas no geral, na prática você tá muito orientado ao domínio, você tem um domínio onde ele claramente te permite marcar um batch, ou seja, "eu vou fazer isso de seis em seis meses, eu vou fazer isso anualmente". [Isso pode ser causado] até mesmo por necessidade [leia-se restrições] no deployment. Se pessoas tiram férias em dezembro e o cliente precisa do modelo funcionando sempre, não tem como eu ficar fazendo deployment. Então eu chego lá com o papai noel, treino o modelo e faço o deployment. Ele [o cliente] não vai sentir nada. Aí você tem essas decisões de negócio.	O período para retreinar o modelo está associado ao domínio.	Restrições [Treino do modelo]
via experimentação, você compara os seus algoritmos com múltiplos algoritmos que você esteja desenvolvendo e também com as baselines, para ver se ele melhora as baselines, se o algoritmo é de fato melhor.	Corretude de algoritmos é verificada através da comparação com baselines	Método [Avaliação do modelo]
	Comparar algoritmos com baselines	Atividade [Avaliação do modelo]
A parametrização é a comparação dele com ele mesmo com diversos tipos de parâmetros, né? Então você roda em diferentes parametrizações, o que tiver a melhor métrica, você adota.	A corretude de hiperparâmetros é verificada através da variação de hiperparâmetros e comparação de resultados de avaliação	Método [Avaliação do modelo]
depende muito do contexto, no geral você vai utilizar um framework, um Tensorflow, um Sklearn, um ..., qualquer coisa assim. Aí se você vai usar frameworks, que são desenvolvidos em cima desses frameworks, como é o caso de .... Sei lá, você está trabalhando no domínio de aprendizagem de representação em grafos de conhecimento, tem algumas bibliotecas específicas, algumas implementações que o	Uso de frameworks para a criação de modelos: TensorFlow, Scikit-learn, PyTorch.	Método [Treino do modelo]

<p>peçoal já desenvolve em cima, aí você adota uma dessas que já facilite o seu processo, e você treina em cima delas. Mas no geral, se você tá fazendo algo muito novo, você vai desenvolver a sua metodologia, você que vai codar [desenvolver o código para] o treino e a validação.</p>	<p>Soluções novas requerem desenvolvimento do zero.</p>	<p>Limitação [Treino do modelo]</p>
<p>: assim, você testa. Você pega um TensorFlow e um PyTorch, é o que a comunidade usa, você assume que eles estão certos, enfim. Você assume que está certo e é isso. Dificilmente você vai fazer casos de teste para a biblioteca, para ver se ela tá realmente bem implementada, ou implementada de modo correto. O máximo que você pode fazer na prática é olhar lá o fórum de discussão, olhar as issues e vê onde tem problema que a comunidade já detectou. E você tá desenvolvendo, detectou um problema, vai lá e reporta.</p>	<p>Confiança na correteude da implementação de frameworks.</p>	<p>Limitação [Treino do modelo]</p>
<p>você olha pela métrica que você definiu, não tem ... É quase impossível você fazer um debug em um modelo que treina, porque enfim, a inicialização desses modelos por vezes é aleatória, então você não tem nenhuma garantia de que você vai conseguir enxergar o mesmo cenário todas as vezes, é tudo estocástico, então você não tem como debugar.</p>	<p>Característica aleatória do modelo dificulta a verificação.</p>	<p>Desafio [Treino do modelo]</p>
<p>O que você faz é, você roda o modelo e vê o resultado dele, se o resultado estiver bom, você provavelmente acertou, se não estiver, ou seja, ele para de funcionar em alguma parte, você evidentemente sabe que tem um erro. Mas senão se ele estiver ruim, ou você acha que ele tem um erro ou ele só é ruim mesmo. É muito do feeling do desenvolvedor, não tem forma de fazer isso.</p>	<p>A indicação de correteude de um modelo é observada na avaliação.</p>	<p>Limitação [Avaliação do modelo]</p>
<p>executo através dos frameworks de machine learning, não é muito diferente disso não.</p>	<p>Uso de framework para avaliação do modelo.</p>	<p>Método [Avaliação do modelo]</p>
<p>você pode aplicar em uma outra parte do dado, se você tiver uma parte não observada do dado, você pode fazer uma anotação em cima desse dado, e executar o seu modelo em cima desse dado. Aí você garante empiricamente que o seu modelo funciona.</p>	<p>Uso da base de validação para verificação final do modelo.</p>	<p>Atividade [Avaliação do modelo]</p>
<p>Ou você faz análises estatísticas, você tira diferentes métricas em cima do seu modelo, faz uma análise sobre a questão de assertividade do seu modelo com base em determinada característica ou determinada label do dado, etc. Você tem essas duas formas.</p>	<p>Uso de métodos estatísticos para avaliação do modelo.</p>	<p>Atividade [Avaliação do modelo]</p>
<p>dependendo do seu domínio, se você tá trabalhando em um modelo científico, quem vai definir essa [medida] são os baselines. Você tem os baselines lá e eles vão te dizer qual é o [valor] suficiente.</p>	<p>Literatura define alvos de métricas para modelos científicos.</p>	<p>Limitação [Avaliação do modelo]</p>
<p>Quando você tá trabalhando em um cenário real, você não tem esse [valor] dado. Você vai ter que ..., por exemplo, você vai ter uma acurácia de 70% às vezes, e vai estar imaginando que só 99% que serve. Mas os cientistas de domínio [...] vão olhar aquele produto e vão ver que aquilo é bom, muito melhor do que existia antes. Às</p>	<p>Soluções reais não possuem alvos para as métricas</p>	<p>Desafio [Avaliação do modelo]</p>

<p>aquele precisa e vai ter que aquele é bom, mais nem sei se que aquela aquela... vezes eles tinham que fazer na mão, passavam dias, e ele aceita perder esses 30% de acurácia e não ter que fazer nada, talvez fazer um fine tuning manual depois. Então tem essas duas óticas no geral. No prática você vai precisar do dono do dado, o cientista do domínio, e no caso científico você tem as baselines lá.</p>	<p>A avaliação do especialista do domínio pode ser mais importante do que os valores de métricas.</p>	<p>Limitação [Avaliação do modelo]</p>
<p>fazer justamente isso que você perguntou. Garantir que ele é um bom modelo para um resultado prático. Garantir que ele não tem nenhum bias e etc.</p>	<p>Garantir que o modelo é bom para fins práticos é o maior desafio da avaliação do modelo.</p>	<p>Desafio [Avaliação do modelo]</p>
<p>ele é muito auxiliado, quer dizer, ele sempre vai ser auxiliado dos cientistas de domínio, entendeu? Pelo menos de alguma pessoa que entenda de domínio.</p>	<p>Especialista de domínio auxilia a engenharia de atributos.</p>	<p>Atividade [Engenharia de atributos]</p>
<p>No geral eu sigo a primeiro modo as técnicas clássicas de fazer Feature engineering. Por exemplo, trabalhando com imagens eu vou tentar usar a primeira coisa que é um shift, se não foi suficiente, eu pego essa mesma imagem eu combino com outras características que eu extraio dela. E aí você vai compondo lá o seu vetor de features, né? Da forma que mais lhe agrade e que melhor se adequa o problema.</p>	<p>Extração de atributos se inicia de acordo com técnicas clássicas</p>	<p>Método [Engenharia de atributos]</p>
<p>Não tem..., é muito na experimentação e no que já existe sobre o tema. Ou [então] o cientista de domínio ele fala, "Ah não, para esse dado aqui só interessa você olhar a intensidade da cor vermelha", beleza. Você faz lá o filtro em cima da imagem, que extrai a cor vermelha e pronto, acabou. Então tem essas duas abordagens.</p>	<p>Desenvolvedor executa a engenharia de atributos.</p>	<p>Atividade [Engenharia de atributos]</p>
	<p>Desenvolvedor não conhecimento específico sobre o dado.</p>	<p>Desafio [Engenharia de atributos]</p>
	<p>Especialista de domínio auxilia a engenharia de atributos.</p>	<p>Atividade [Engenharia de atributos]</p>
<p>quando é avaliado. Você só vai saber se o seu input é suficiente, se o seu modelo ele consegue aprender em cima daquele dado e você só sabe se o seu modelo aprendeu algo sobre aquele dado, quando ele te dá uma métrica.</p>	<p>Incerteza sobre a correteude da engenharia de atributos</p>	<p>Desafio [Engenharia de atributos]</p>
	<p>A avaliação da indício sobre a correteude da engenharia de atributos.</p>	<p>Limitação [Engenharia de atributos]</p>
<p>Feature engineering é um processo bem complicado, primeiro que você tem que entender o domínio e depois de você entender o domínio, você tem que identificar quais são os pontos que farão o modelo estatístico - que no caso é nosso modelo de machine learning - ter melhor eficiência para entender a representação daquele dado.</p>	<p>Necessidade de compreender o domínio</p>	<p>Desafio [Engenharia de atributos]</p>
<p>A Feature Engineer faz justamente o processo de representar o dado. A gente tá transformando uma imagem em um vetor de características, um vetor de pontos flutuantes, e o seu modelo vai ter que entender que aqueles pontos flutuantes refletem determinado símbolo. Você fazer esse mapeamento manualmente é extremamente complexo, você tem que conhecer o domínio, você tem que conhecer técnicas para fazer isso. Em geral é isso, essa é maior dificuldade.</p>	<p>Dificuldade em garantir que as features são relevantes para o modelo aprender a representação desejada.</p>	<p>Desafio [Engenharia de atributos]</p>