

# Inteligência Artificial - Problemas

Produção e Fundamentos da Indústria 4.0

Carlos Diego Rodrigues

Federação das Indústrias do Estado do Ceará - FIEC

2 de junho de 2022

# Agenda

## Problemas de otimização

- Problemas industriais clássicos

- Problemas de alocação de recursos

- Problema logísticos

## Problemas de aprendizado de máquina

- Conceitos iniciais

- Problemas

- Validação

# Pesquisa operacional

A pesquisa operacional (PO), ou investigação operacional (IO), é um ramo interdisciplinar da matemática aplicada que faz uso de modelos matemáticos, estatísticos e de algoritmos na ajuda à tomada de decisão. É usada sobretudo para analisar sistemas complexos do mundo real, tipicamente com o objetivo de melhorar ou otimizar a performance.

- Programação linear.
- Programação não-linear.
- Programação inteira.
- Otimização em redes.
- Programação dinâmica.
- Simulação discreta.
- Teoria dos Jogos

# Três principais etapas da pesquisa operacional

1. **Modelagem:** o problema é abstraído da realidade para um conjunto de objetos matemáticos que permitem (e ao mesmo tempo limitam) sua precisa definição.
2. **Resolução:** um método adequado para o modelo e para o usuário é selecionado para o processo de resolução. A depender da forma como o problema foi modelado, o usuário tem acesso a uma seleção diferente de métodos.
3. **Interpretação:** Uma vez resolvido o problema, é preciso retornar à aplicação prática, interpretando-se a solução uma vez abstraída de volta à realidade.

# Problema em pesquisa operacional

- Temos uma função, a qual queremos otimizar, maximizar ou minimizar, chamada função objetivo.
- A análise é feita através de um modelo matemático, em geral de equações, inequações, embora algumas áreas trabalhem com outras metodologias de análise - tabelas, listas, etc.
- Os dados são divididos em parâmetros (dados de entrada) e variáveis (dados a serem calculados para fornecer a saída).
- É realizada uma caracterização do universo de soluções, em geral, através de um conjunto de relações entre as variáveis chamado de conjunto de restrições.

# Problema da dieta



- Um dos primeiros problemas estudados na programação linear.
- O objetivo é construir uma dieta de custo mínimo com base numa série de alimentos para os quais são conhecidos seus custos por unidade e valores nutricionais.
- A dieta também deve obedecer os requisitos mínimo e máximo de cada componente nutricional.

## Um exemplo de dieta

	Bread	Milk	Cheese	Potato	Fish	Yogurt
Cost	2.0	3.5	8.0	1.5	11.0	1.0
Protein (grams)	4.0	8.0	7.0	1.3	8.0	9.2
Fat (grams)	1.0	5.0	9.0	0.1	7.0	1.0
Carbohydrates (grams)	15.0	11.7	0.4	22.6	0.0	17.0
Calories	90.0	120.0	106.0	97.0	130.0	180.0

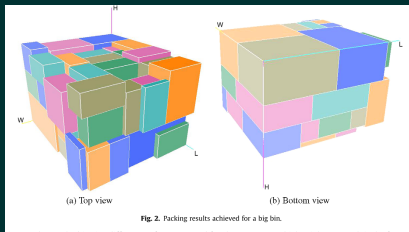
- A dieta deve conter pelo menos 300 calorias.
- Um mínimo de 10g de proteínas, 10g de carboidratos e 8g de gordura.
- É interessante comer pelo menos meia porção de peixe e tomar uma medida de leite.
- Qual o menor custo possível para essa dieta? Quais alimentos fazem parte dela?

# Modelando o problema da dieta

- A quantidade de cada alimento  $j$  é uma variável denominada  $x_j$ .
- Queremos minimizar a soma dos custos  $c_j$  de cada alimento  $j$ :  
$$\sum_j c_j x_j$$
- Além disso para cada nutriente  $i$  a quantidade daquele nutriente no alimento  $j$  é uma entrada na tabela  $a_{ij}$ . Deve-se obedecer  $\sum_j a_{ij} x_j \geq b_i$ , onde  $b_i$  seria a quantidade mínima do nutriente  $i$ .



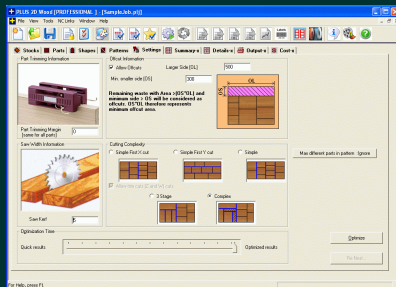
# Problemas de corte e empacotamento



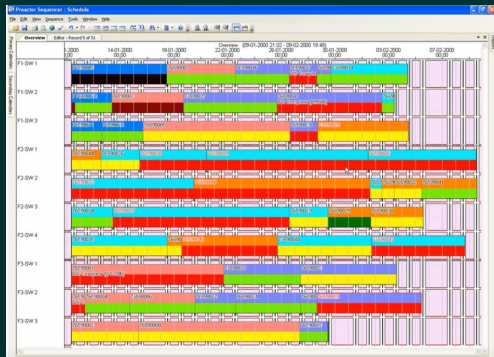
- Problemas de corte e empacotamento são importantes problemas que ocorrem na prática.
- Diversas indústrias podem ser atravessadas por problemas dessa natureza, tais como construção civil, indústrias de vidro, madeira, aço, tecido, etc. Ou qualquer indústria que comprometida com um setor logístico.
- Corte de peças, carregamento de paletes, carregamento de caminhões.

# Dimensões de corte e empacotamento

- Unidimensional: problemas de vigas de metal na construção civil.
- Bidimensional: problemas da indústria de vidro ou indústria moveleira.
- Tridimensional: carregamento de paletes, empilhamento de contêineres ou carregamento de caminhões.

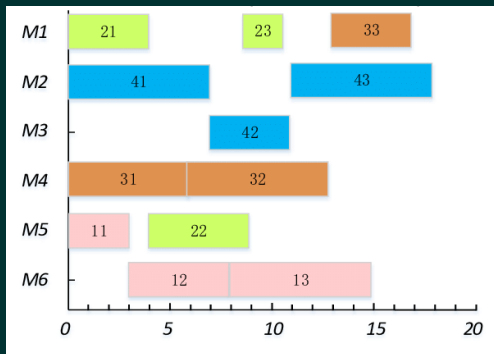


# Problemas de escalonamento de produção



- Uma série de trabalhos precisam ser realizados em diversas máquinas (operadores).
- Qual o melhor sequenciamento das tarefas de forma a minimizar o tempo máximo (*makespan*)?
- Alternativamente: considerar datas de entrega.

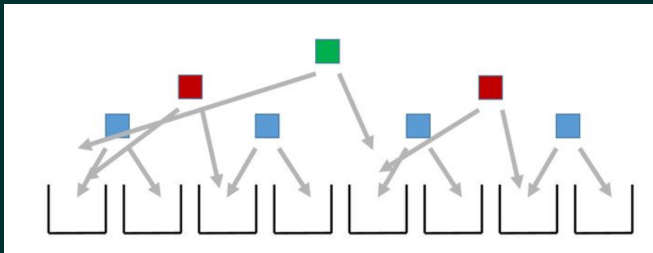
# Problemas de escalonamento de produção



- Organização de um projeto com restrições de precedência.
- Número limitado de colaboradores ou problemas de formatação da equipe.

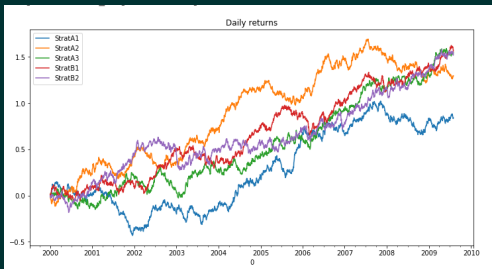
# Problema do equilíbrio de contas

- Uma série de contas a pagar precisam ser organizadas em datas de vencimento adequadas.
- Qual a melhor distribuição desses pagamentos de forma a causar o menor impacto no orçamento?



# Carteira de investimentos para um projeto

- Como organizar as ações de um projeto (custos) ao longo do tempo alinhadas a uma carteira de investimentos que paga pelo projeto?
- Qual o investimento de capital mínimo a cada período de tempo para garantir a execução do projeto dentro do prazo?
- Quais são as melhores opções de investimento na execução do projeto?



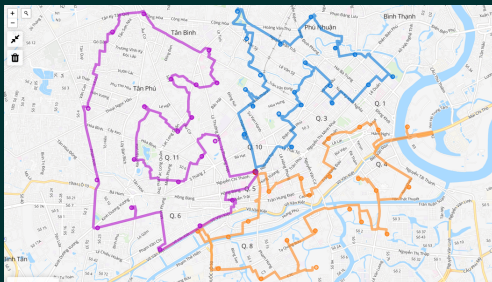
# Problema de roteamento de veículos

- Encontrar a melhor rota para entrega e/ou recolhimento de materiais ou prestação de serviços.
- Além dos problemas clássicos no mapa, problemas internos na planta com roteamento de drones.



# Problema de roteamento de veículos

- Mais restrições: tamanho da frota, janelas de tempo, precedência, tempo de serviço, etc.
- Online: novos itens ou tarefas são adicionados na rota.





# Agenda

## Problemas de otimização

- Problemas industriais clássicos

- Problemas de alocação de recursos

- Problema logísticos

## Problemas de aprendizado de máquina

- Conceitos iniciais

- Problemas

- Validação

# Componentes da entrada

- Conceito: algo a ser aprendido. Nosso objetivo é construir um conceito operacional e inteligível, uma regra ou cálculo capaz de fornecer valores para as instâncias.
- Instâncias: Exemplos individuais de um conceito a ser aprendido.
- Atributos: medidas sobre uma instância.

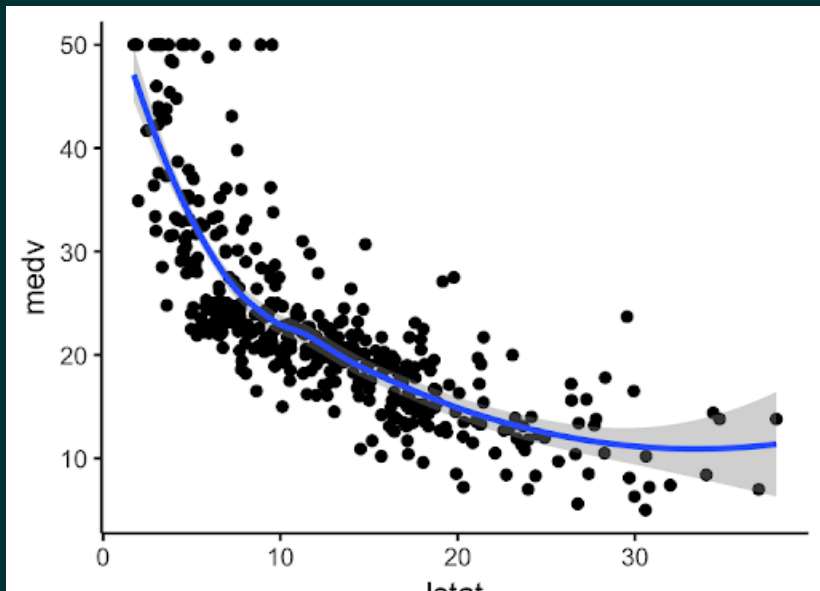
# O que é um conceito

- Comportamento da variável alvo a partir dos atributos apresentados nas instâncias.
- É a saída de um esquema de aprendizado.
- Tipos de aprendizado:
  - Regressão: predição de um valor numérico (contínuo).
  - Classificação: predição de uma classe discreta.
  - Associação: detecção de relações entre as características.
  - Agrupamento: reunião de instâncias em grupos por similaridade.

# Regressão

- Tipo mais comum de mineração de dados.
- Busca-se uma função numérica capaz de descrever um conjunto de pontos em um espaço multidimensional.
- Regressão é aprendizado *supervisionado*.
- Medidas de distância (ex.: teste  $\chi^2$ ) calculam a taxa de sucesso de uma regressão.

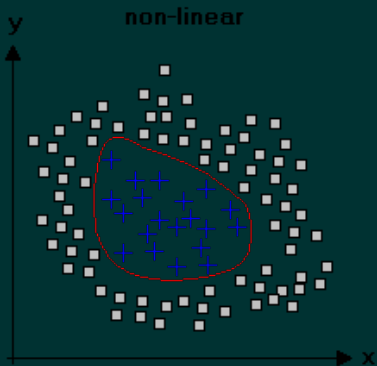
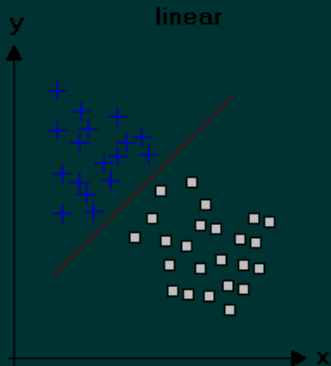
## Exemplo regressão



# Classificação

- Exemplos: previsão do tempo, lentes de contato, classificação de espécies, negociações (empréstimos).
- Classificação é aprendizado *supervisionado*.
- Resultado é uma *classe* de um exemplo.
- A medida do sucesso é a proporção de instâncias corretamente classificadas em um conjunto de teste.
- Na prática a medida é subjetiva...

## Exemplo classificação



# Associação

- Pode ser aplicado quando não há classes ou não há estruturas consideradas interessantes.
- Associação é aprendizado *não supervisionado*.
- Diferença para classificação:
  - Pode prever valores de qualquer atributo, não apenas a classe.
  - Em geral vai produzir um conjunto muito maior de regras.
  - Limites são necessários: cobertura, acurácia são medidas de sucesso impostas para o método.



## Exemplo associação

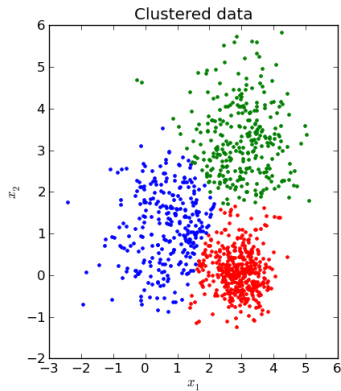
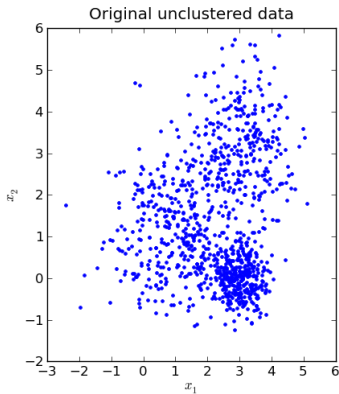


<u>Rule</u>	<u>Support</u>	<u>Confidence</u>
$A \Rightarrow D$	2/5	2/3
$C \Rightarrow A$	2/5	2/4
$A \Rightarrow C$	2/5	2/3
$B \ \& \ C \Rightarrow D$	1/5	1/3

# Agrupamento

- Encontrar grupos de itens que são similares em seus atributos.
- Agrupamento é aprendizado *não supervisionado*.
- Sucesso é medido subjetivamente.

# Exemplo agrupamento



# Validação: avaliando o que foi aprendido

- Questões: treinar, testar, melhorar.
- Predizendo a performance: limites de confiança.
- Separar, validação cruzada, *bootstrap*

# Avaliação, a chave do sucesso

- Quão bom é o modelo que foi aprendido?
- Erro nos dados de treinamento não são bons indicativos.
- Solução simples: separar os conjuntos de treinamento e teste.
- Algumas vezes os dados pré-classificados são limitados.

# Problemas na avaliação

- Confiança estatística
- Escolha da medida de desempenho
  - Número de classificações corretas
  - Acurácia das estimativas probabilísticas
  - Erro nas previsões numéricas
- Custos associados aos erros.

# Treinamento e testes

- Medida natural para problemas de classificação: taxa de erro.
  - Sucesso: classe predita corretamente.
  - Erro: classe predita incorretamente.
  - Taxa de erro: proporção de erros sobre todo o conjunto de instâncias.
- Erro de resubstituição: taxa de erro quando o modelo é aplicado ao conjunto de treinamento.
- Erro de resubstituição é otimista!

# Treinamento e testes

- Conjunto de teste: conjunto de instâncias independentes que não foram utilizadas na elaboração do classificador.
- Treinamento e teste podem ser de natureza diferente.
  - Exemplo: dados conhecidos sobre os funcionários de uma empresa A podem ser utilizados para fazer inferências sobre uma empresa B.
  - Idealmente porém eles são retirados de dados que possuem uma distribuição comum.
- Em nenhuma das etapas da elaboração do classificador pode haver o uso do conjunto de testes!
- Alguns classificadores vão separar os dados em três: conjunto de treinamento, validação e testes.



## Retirando o máximo dos dados

- Uma vez o processo de avaliação esteja completo, todos os dados podem ser usados para construir o classificador final (para usos práticos).
- Em geral, quanto mais dados, melhor o classificador (mas o retorno diminui).
- Quanto maior o conjunto de teste mais precisa é a estimativa do erro.
- Esse procedimento de divisão do conjunto é chamado *holdout*.

# Predizendo a performance

- Dada uma taxa de erro obtida de um processo experimental, quão perto ela está da verdadeira taxa de erro?
- Podemos construir uma distribuição amostral da proporção de erros para estimar um intervalo de confiança para o erro.
- Em um processo similar àqueles utilizados na inferência estatística para obter intervalos de confiança para proporções de uma população.

## Intervalo de confiança para a taxa de sucesso

- Seja  $f$  a taxa de sucesso que obtivemos.
- Seja  $p$  a taxa real de sucesso na população.
- Seja  $c$  o índice de confiança desejado.

$$P(-z_c \leq X \leq z_c) = c$$

## Transformação para a normal padrão

- Transformando para a normal padrão:

$$P \left( -z_c \leq \frac{f - p}{\sqrt{\frac{p(1-p)}{N}}} \leq z_c \right) = c$$

- Resolvendo para  $p$  temos para o limite de confiança  $c$ :

$$p = \frac{\left( f + \frac{z_c^2}{2N} + z_c \sqrt{\frac{f}{N} - \frac{f^2}{N} + \frac{z_c^2}{4N^2}} \right)}{1 + \frac{z_c^2}{N}}$$

# Exemplos

- $f = 75\%$ ,  $N = 1000$ ,  $c = 80\%$  (ou seja,  $z_c = 1.28$ ):  
[0.691, 0.801]
- $f = 75\%$ ,  $N = 100$ ,  $c = 80\%$  (ou seja,  $z_c = 1.28$ ):  
[0.549, 0.881]

# Como dividir os conjuntos

- Se tivermos apenas um conjunto, como dividir?
- Uma porcentagem para teste e o restante para treinamento.
- Problema: os exemplos podem não ser representativos no treinamento ou teste.
- Solução: estratificação.
- *Holdout* repetido: várias seleções de amostra para teste e treinamento são feitas sobre o mesmo conjunto para efeitos de estabilidade.

# Validação cruzada

- Uma validação cruzada de  $K$  partes evita a sobreposição.
  - Primeiro passo: dividir o conjunto em  $K$  partes iguais.
  - Segundo passo: usar uma das partes para teste e as demais para treinamento.
  - Algoritmo é aplicado em  $K$  conjuntos de treinamento diferentes.
- Subconjunto estratificados.
- Erro é a média das  $K$  execuções.
- Outros métodos de estimação desse erro.
- Método padrão  $K = 10$  estratificado.

# Validação cruzada com um-de-fora

- Um-de-fora: técnica particular onde todos os exemplos são usados para treino e apenas um é usado para teste.
- Melhor uso dos dados.
- Não há aleatoriedade.
- Computacionalmente caro pois envolve uma execução para cada instância.
- Não permite estratificação.



# Bootstrap

- Validação cruzada não permite amostras com o mesmo elemento mais de uma vez.
- *Bootstrap* permite reposição com elementos do conjunto de treino.
  - Amostragem com reposição em um conjunto com  $n$  instâncias para formar um novo conjunto de dados com  $n$  instâncias.
  - Use este novo conjunto como treinamento.
  - Use as instâncias do conjunto original que não aparecem nesse conjunto como teste.
- *Bootstrap* 0.632