

Seleção de Features

Diversificados

Quem Somos?



Tatyana Zabanova



Estevão Uyra

Agenda

1. Métodos "embutidos"
2. Wrappers
3. Seleção de Features no processo de modelagem
4. Mão na massa

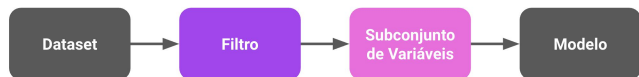
Revisão: Visão Geral

Tipos de seleção de features

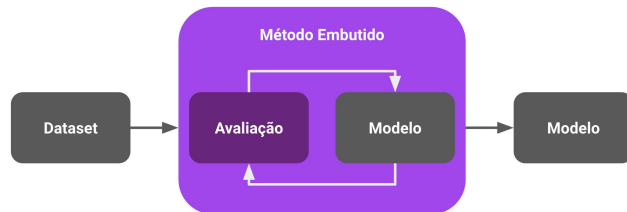
Três principais grupos de abordagens de seleção de features:

- Baseados em **filtros**: a ideia é pensar em algum critério e descartar (filtrar) todas as variáveis que não satisfazem esse critério
- Métodos "**embutidos**": alguns algoritmos já incluem seleção de feature na sua lógica
- **Wrappers**: seleção de features como problema de busca, testando várias combinações de features

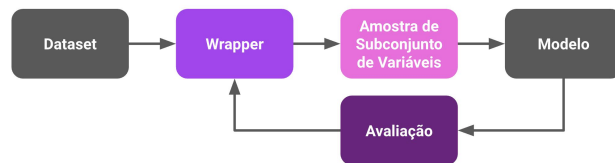
Filtro



Seleção embutida



Wrapper



Seleção de Features:

Métodos com SF embutida

Processo

Um método com seleção de features embutida funciona assim:

- Treina um modelo de machine learning
- Deriva a importância das variáveis no modelo (o peso de cada variável para prever o resultado)
- Remove as variáveis de pouco impacto usando a importância

Vamos ver melhor como isso funciona usando como exemplo Regressão Lasso e Random Forest.

Regressão Linear

É um modelo linear, isto é, a nossa predição é uma combinação linear das variáveis:

$$\hat{y} = w_0x_0 + w_1x_1 + \dots + w_nx_n + b$$

Acima, temos n variáveis. Se $n=1$, temos a Regressão Linear Simples com inclinação w_0 e intercepto b .

Mínimos Quadrados

Para encontrar os parâmetros da regressão, minimizamos o custo (famigerados Mínimos Quadrados):

$$custo = \sum_{i=0}^m (y_i - \hat{y})^2 = \sum_{i=0}^m \left(y_i - b - \sum_{j=0}^n (w_j x_j) \right)^2$$

Lasso

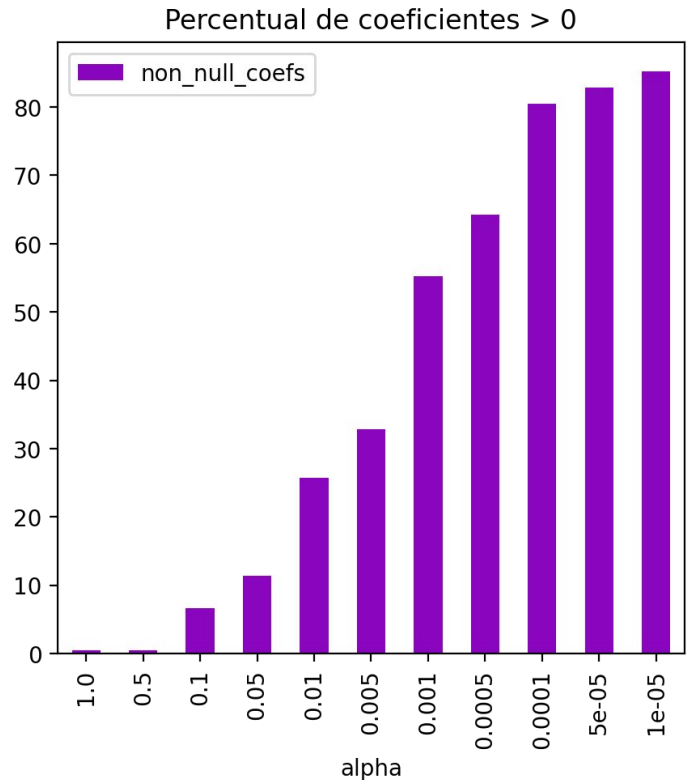
A ideia do Lasso é penalizar não só o erro, mas também a magnitude dos parâmetros.

$$custo = \sum_{i=0}^m \left(y_i - b - \sum_{j=0}^n (w_j x_j) \right)^2 + \alpha \left(|b| + \sum_{j=0}^n |w_j| \right)$$

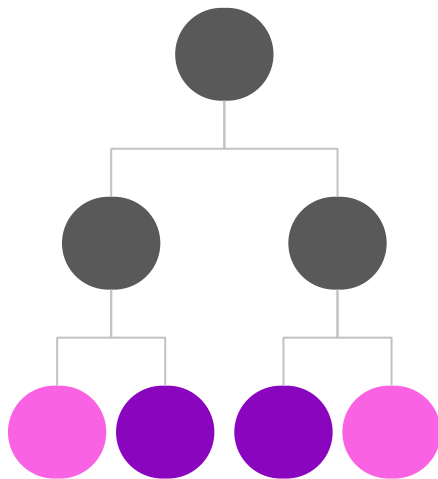
Lasso

Mas o que isso tem a ver com seleção de variáveis?

- O Lasso força os coeficientes das variáveis que contribuem pouco para explicar a resposta a serem nulos
- Podemos ajustar um Lasso, pegar as variáveis com coeficiente 0 e jogar fora



Floresta Aleatória e Importâncias



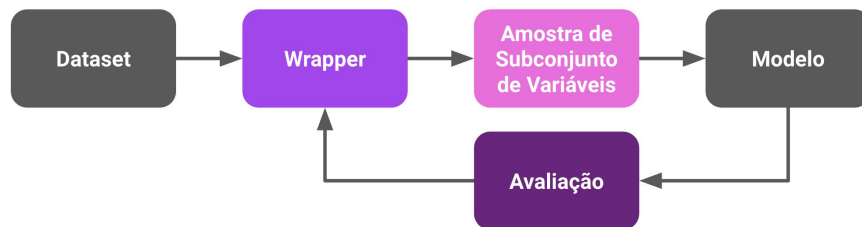
Duas medidas de importância simples: **diminuição média de impurezas** e **diminuição média de precisão**.

- Floresta Aleatória: grupo de árvores de decisão
- Cada árvore "vê" somente parte das variáveis e das observações
- Cada nó é uma condição em cima de uma variável, separando em 2 conjuntos
- Importância: medida de pureza destes conjuntos
- Podemos jogar fora as variáveis pouco importantes

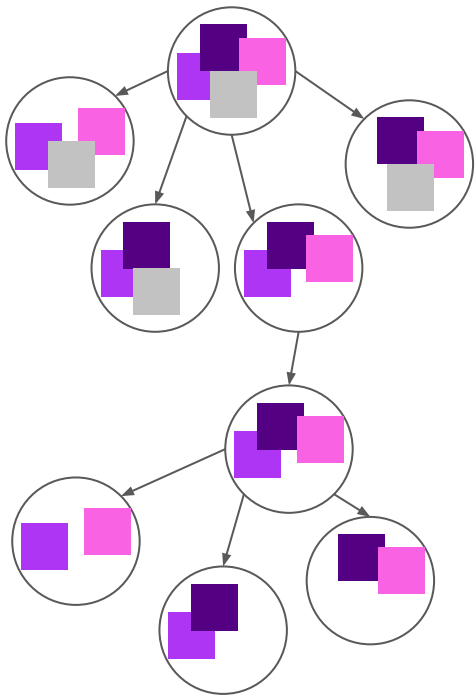
Seleção de Features: Wrappers

Wrappers

- Computacionalmente mais exigentes, pois necessitam de múltiplas iterações de treinamento e avaliação.
- Conseguem se valer dos potenciais e limitações de cada modelo.



A força bruta (Não usar!)

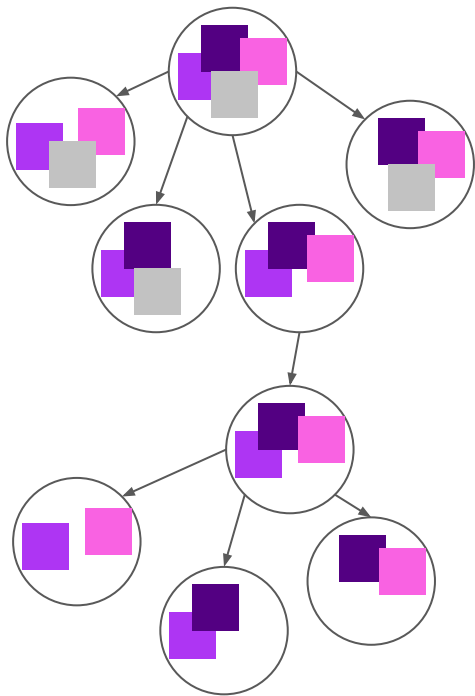


- Teste todas as combinações com -1 variável
- Mantenha a melhor das combinações
- Repita

Dessa forma, teoricamente seria possível encontrar o subconjunto de features com melhor desempenho.

Como vocês acham que isso pode dar errado?

A força bruta (Não usar!)

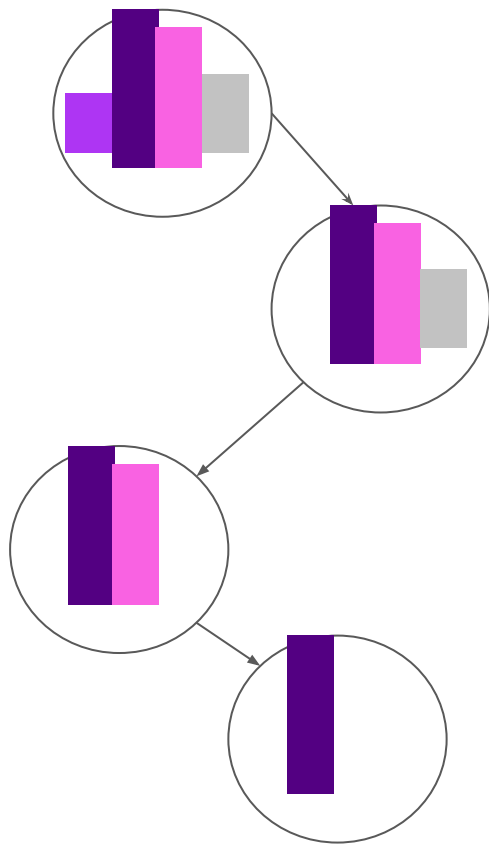


Porém:

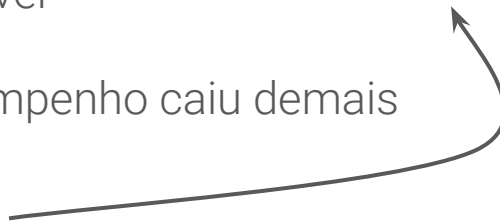
- Na maioria dos casos é impossível testar todas as combinações. Com 10 features, são 3628800
- Seleção de features também pode sobreajustar

Precisamos pensar em alguma estratégia mais eficiente de prosseguir com isso.

Sequencial



- Treine um modelo com todas as variáveis
- Remova uma variável
- Observe se o desempenho caiu demais
- Se não, repita



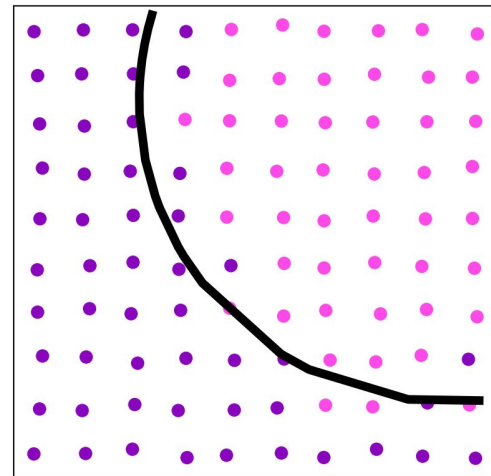
**Isso funciona em qualquer ordem?
Como você ordenaria as variáveis?**

Importância

Intuição: remover primeiro as variáveis que pouco contribuem para a predição.

Importância: uma variável é importante se contribui muito para a predição.

- Regressão Linear com variáveis normalizadas: coeficientes mais próximos de 0 têm importância menor
- Modelos de árvores:
 - Cada nó é uma condição em cima de uma variável, separando em 2 conjuntos
 - Importância: medida de pureza destes conjuntos
 - Se os conjuntos são muito puros, significa que a nossa condição separa muito bem, logo a variável que usamos é muito importante para fazer a predição



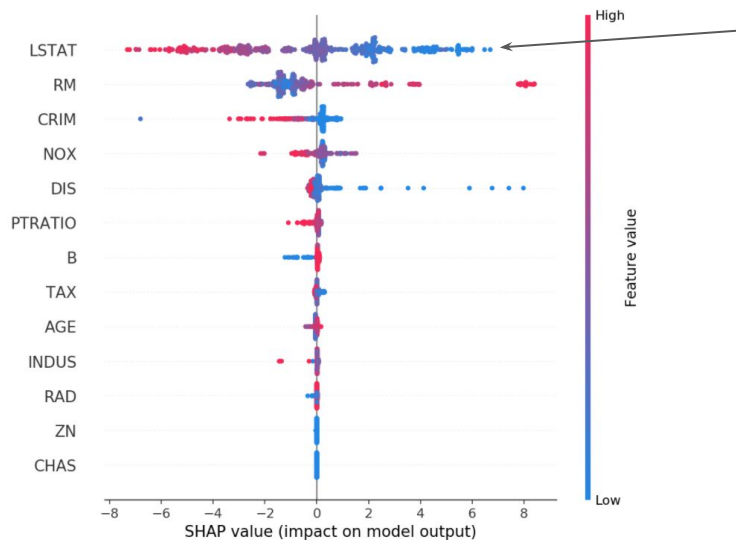
Importância

Algumas medidas comuns (modelos de boosting):

- Split importance: número de vezes que usamos a variável nas nossas árvores
- Average Gain: ganho médio quando a variável é usada
- Shap values: ~~magia negra~~ impacto diferencial a partir de teoria dos jogos

Indo um pouco mais fundo em Shap values

Eles são definidos para cada observação. São o **impacto da variável para aquele exemplo**.

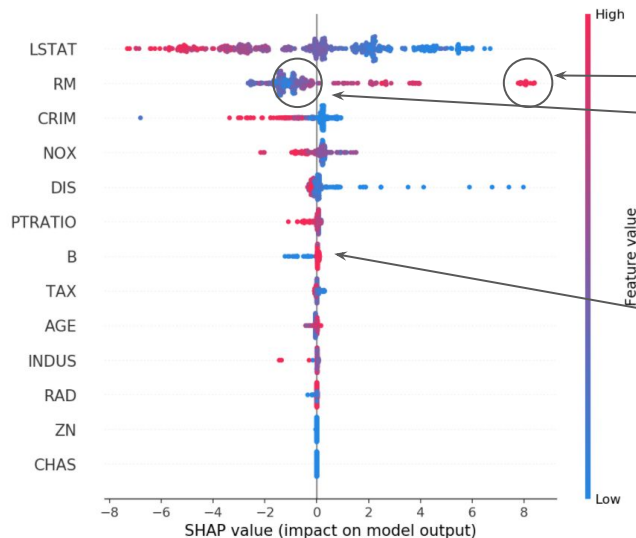


Cada pontinho é um exemplo

Como podemos definir a importância da variável, se para cada exemplo ela é diferente?

Indo um pouco mais fundo em Shap values

Eles são definidos para cada linha. São o *impacto da feature para aquele exemplo*.

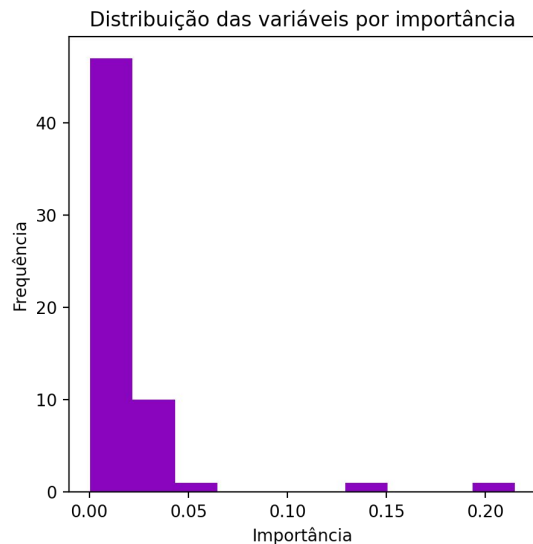


Muito importante para algumas pessoas
Média importância para muitas

Como descrever o impacto dessa variável no modelo?

Solução mais comum:
média dos valores absolutos

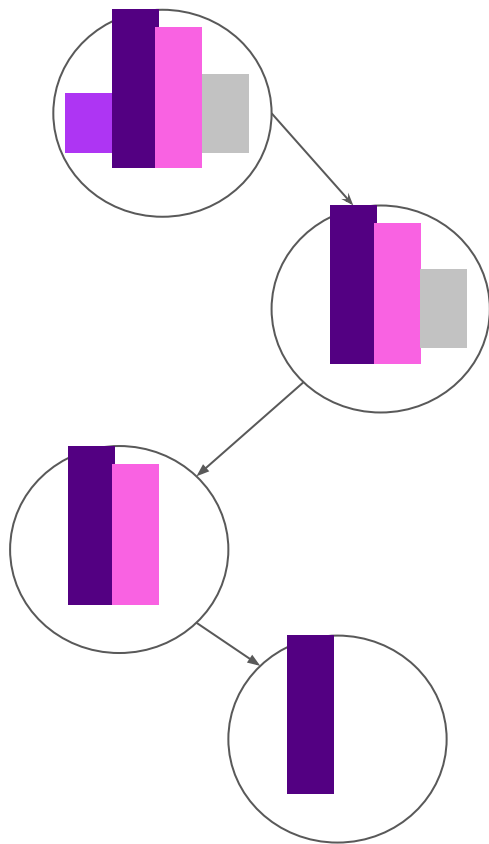
Armadilhas: importância baixa



O que é **importância baixa**?

Será que se colocarmos uma variável aleatória no modelo, a importância dela vai ser 0?

Seleção de Variáveis Backward



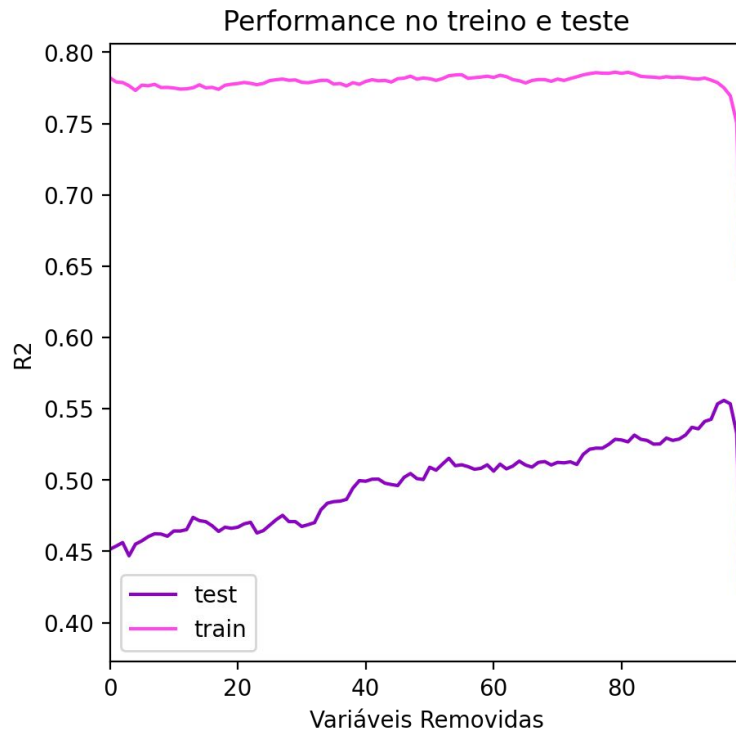
- Treine o modelo
- Calcule a importância das variáveis
- Remova uma variável
- Observe se o **desempenho** caiu demais
- Se não, repita

O que acontece se a gente avaliar o desempenho na própria base?

Desempenho?

Será que medir o desempenho na própria base de treino funciona? Não vamos ter aquele problema de overfit?

Solução: avaliar performance em um conjunto de teste.



Para agilizar o processo

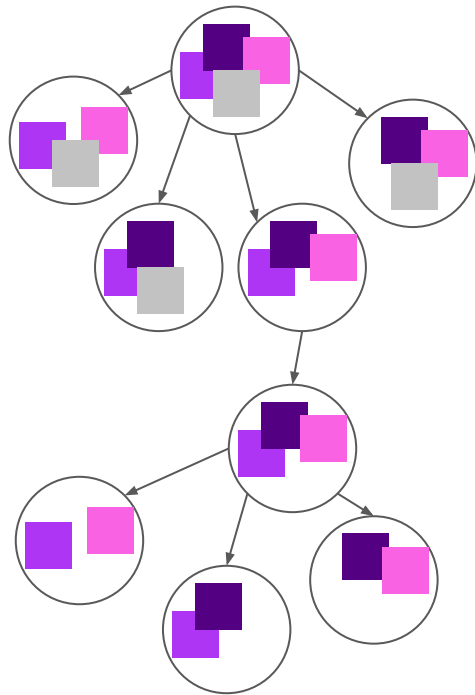
Para bases maiores, esse procedimento pode demorar um pouco (e mais que um pouco).

Para agilizar:

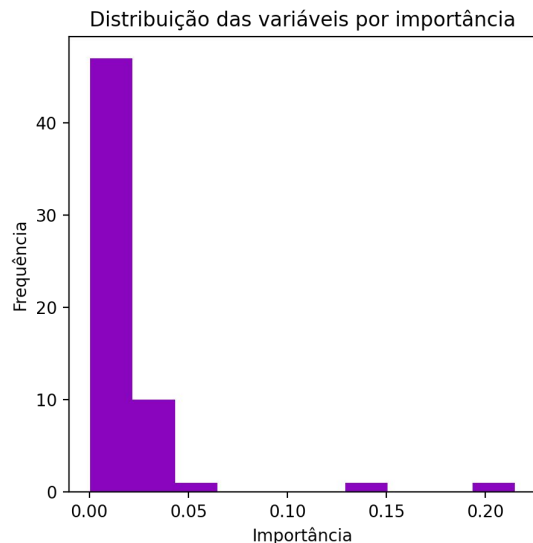
- Use filtros primeiro
- Em vez de remover variáveis uma a uma, remova em blocos (digamos, 5 em 5)
- Calcular a importância pode ser computacionalmente intensivo - não é certo mas dá para calcular só na primeira iteração

Variações

- **Forward**: a mesma coisa, mas ao contrário: começamos com a variável mais importante e vamos adicionando outras
- **Stepwise**: a cada passo, avaliamos adicionar e remover variáveis, até chegar num conjunto de variáveis ótimo (ineficiente, duh)



Armadilhas: correlação

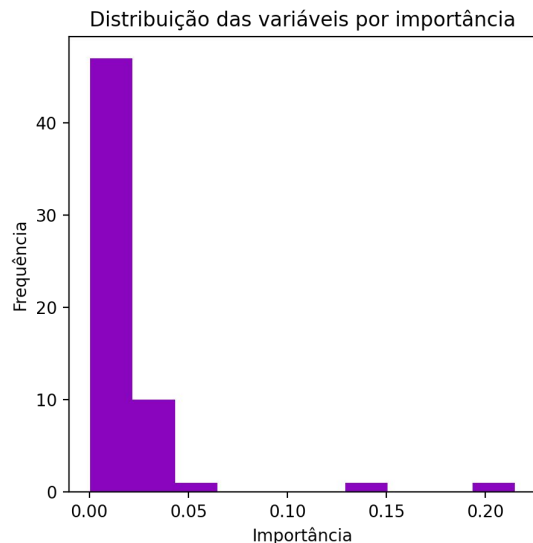


Importance splitting:

- Imagine que colocamos uma variável duplicada no modelo (X e Y, digamos)

O que acha que pode acontecer?

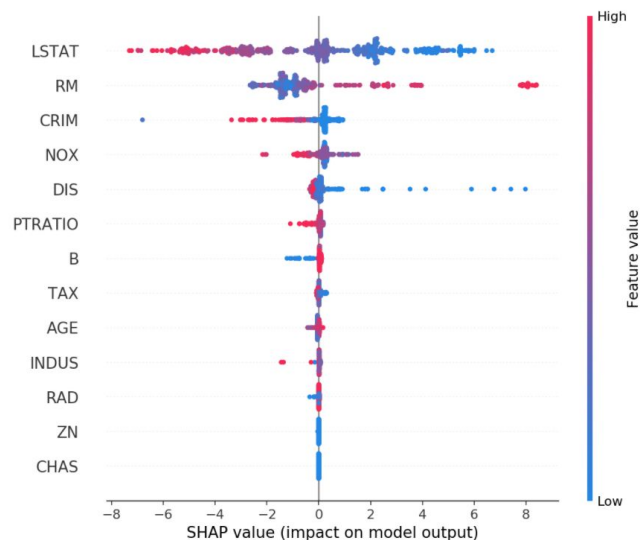
Armadilhas: correlação



Importance splitting:

- Imagine que colocamos uma variável duplicada no modelo (X e Y, digamos)
- A árvore vai usar X metade das vezes, e Y a outra metade das vezes
- A importância vai ser "dividida" entre as duas features, e as duas vão parecer menos importantes

Armadilhas: importância baixa



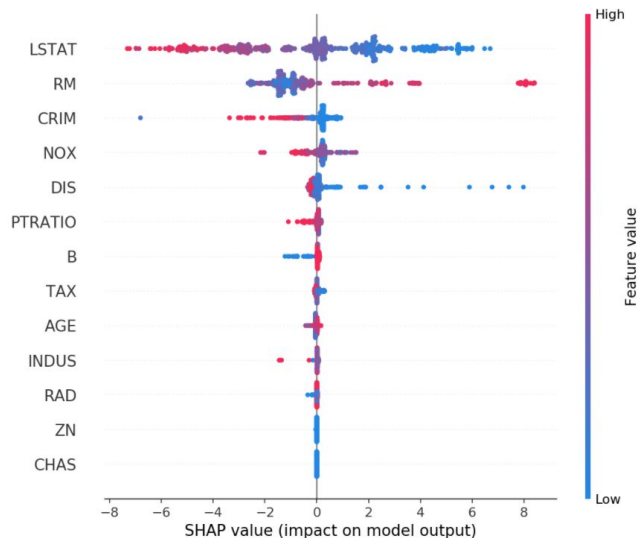
O que é **importância baixa**? 0.1? 0.05?

Depende do problema?

A importância normalmente é uma média. O que acontece se uma variável é muito importante para 1% da população?

O que acham?

Armadilhas: importância baixa



Será que se colocarmos uma variável aleatória no modelo, a importância dela vai ser 0? Não, não vai.

Podemos inserir **variáveis aleatórias** para usar a importância delas como linha de corte.

Para resolver o problema da média, podemos olhar para percentis ou quantis em vez da média.

Seleção de Features: Processo de Modelagem

Como normalmente é feito

- Fazemos seleção de variáveis
- Treinamos vários modelos diferentes
- Escolhemos um e tunamos hiperparâmetros

Ou

- Treinamos vários modelos diferentes
- Escolhemos um
- Fazemos seleção de variáveis e tunamos hiperparâmetros



O jeito certo (mas muito demorado)

- Para cada modelo, fazemos:
 - Seleção de variáveis
 - Hiperparâmetros
 - PS: idealmente, as duas coisas devem ser feitas simultaneamente (a cada iteração da seleção, tunamos os hiperparâmetros também)
- Escolhemos o melhor modelo

Dá pra fazer isso na prática?

O jeito factível

Alternativa factível:



- Hiperparâmetros razoáveis
- Fazer uma seleção de variáveis mais bruta no pré-processamento
- Com um subconjunto de variáveis em mãos, para cada modelo, fazer feature selection com otimização de hiperparâmetros
- Comparar os modelos

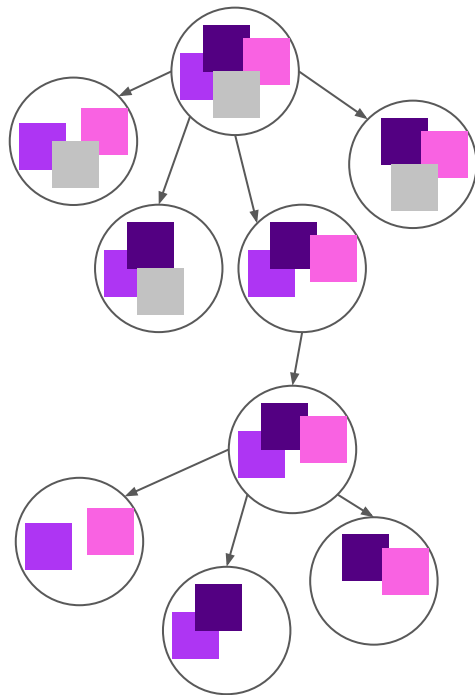
Seleção de Features:

“Consistência” e cuidados práticos

Existe “chance” na seleção das features

- Se repetir a análise, nem sempre os conjuntos selecionados são iguais.
- Para medir a consistência, podemos usar a Intersecção sobre União (*Intersection over Union*, **IoU**)

IoU() = 



Minimize os problemas da inconsistência

- Sempre escolha uma random seed
- Busque ver se há ao menos algumas variáveis em comum nos diferentes conjuntos

Dúvidas?

Colinha

SHAP

Shap é uma ferramenta bem legal para estimar a importância das variáveis em modelos de Machine Learning.

- Github: <https://github.com/slundberg/shap>
- Tutorial 1:
<https://datarisk.io/como-interpretar-modelos-de-machine-learning-complexos/>
- Tutorial 2:
<https://www.kaggle.com/frankepeixoto/shap-values-mini-estudo-para-interpretar-seus>

CHAPTER

**x•DATA
SCIENCE•x**

