
Tera

Módulo 4:
Estatística e Modelagem de Dados

Aula 16: Cross Validation

Instrutora

Cristiane Rodrigues

- **Bacharel em Matemática – UNESP Rio Claro.**
- **Mestre em Estatística – USP Piracicaba**
- Experiências Profissionais:
  - Modelagem de Credito para PF e PJ – Banco Bradesco
  - Experiência com Segmentação e Análise de Series temporais – Atento
  - Consultora Analítica no SAS Institute Brasil
  - Professora do curso SAS Academy for Data Science

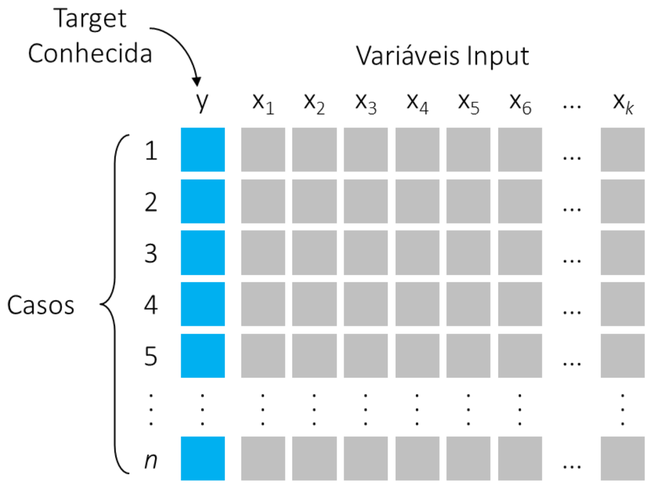
crisrodrigues_27@hotmail.com

Índice

- Motivação
- Princípio do otimismo
- Divisão da base
- Overfitting/Underfitting
- Cross Validation : K-Folds Cross Validation
- Cross Validation: Leave One Out (LOOCV)
- Cross Validation: Outros métodos

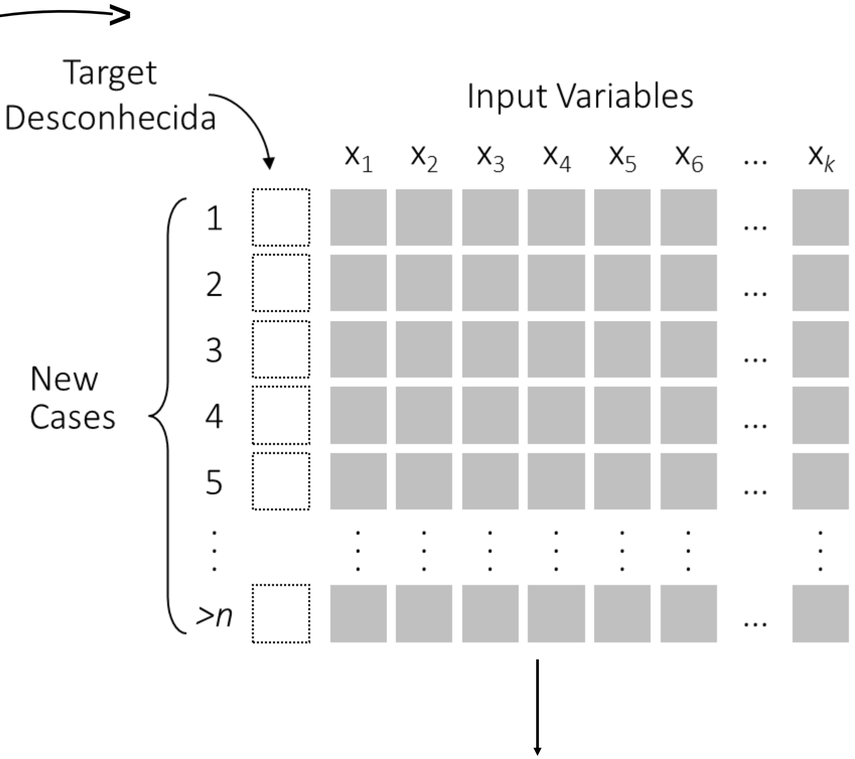
Motivação

- O objetivo dos modelos preditivos é a generalização, ou seja, a habilidade de prever a variável resposta para novos casos



Desenvolvimento de um modelo estatístico

Obtenção da equação que descreve a relação entre a variável Target e as variáveis Input

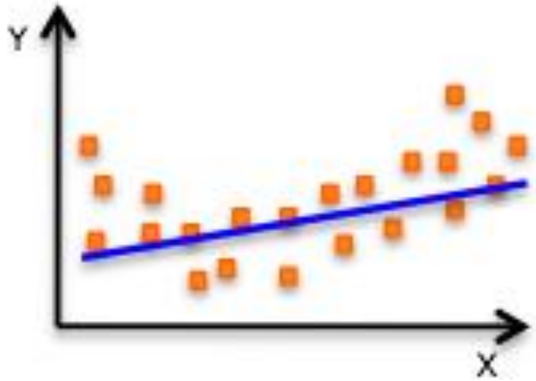


Generalização: Obtenção da variável resposta para novos casos

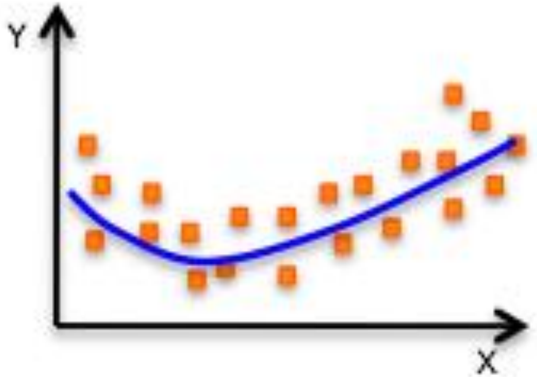
Motivação

- Não basta apenas ajustar um modelo, precisamos verificar se este modelo está bem ajustado, ou seja, se a generalização está ocorrendo de forma efetiva.

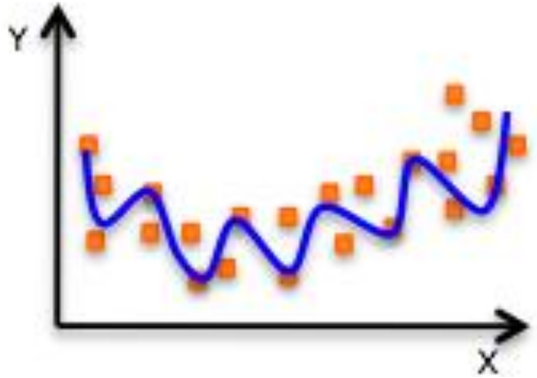
- Como podemos fazer isto?
- Como verificar se o modelo obtido
  - é Simples demais
  - é Complexo demais
  - está OK



Simples Demais
Underfitting

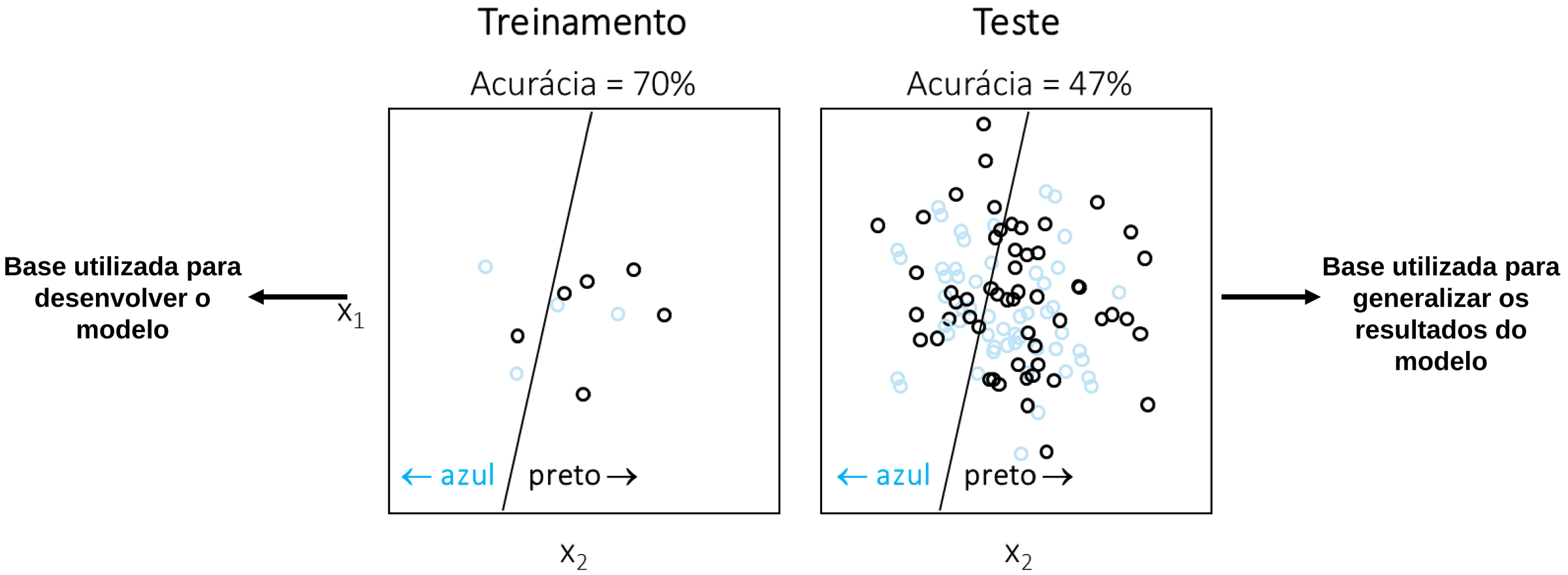


OK
Just Right



Complexo Demais
Overfitting

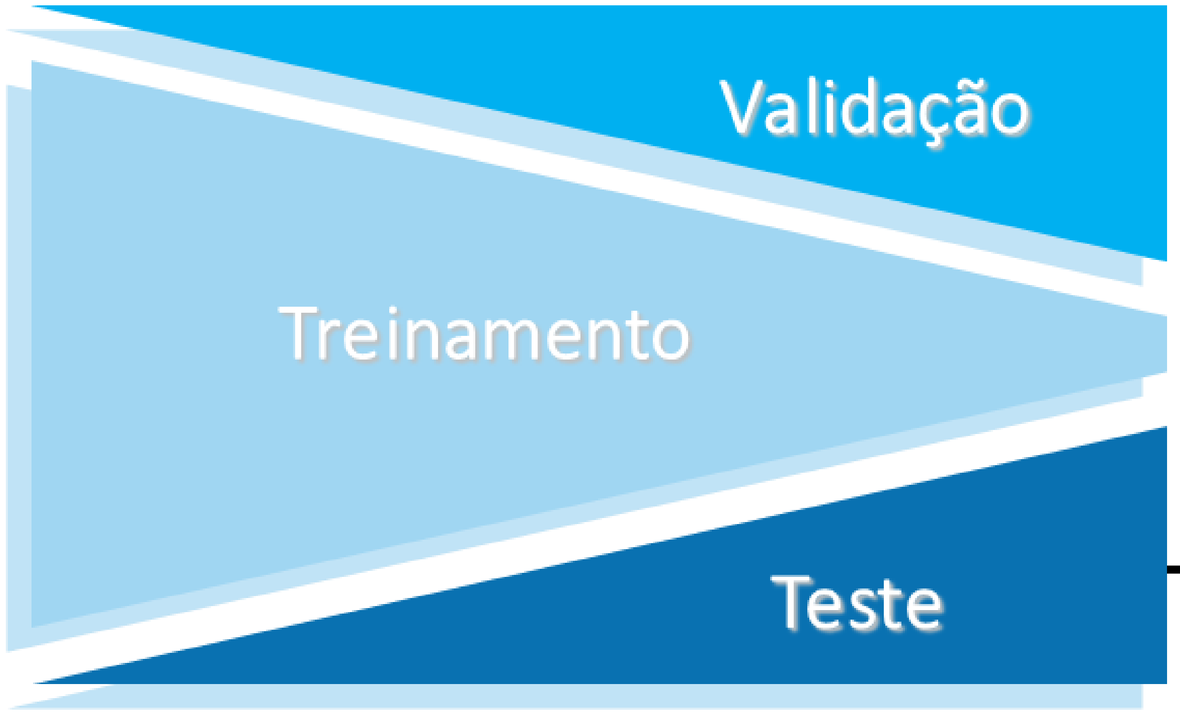
Princípio do otimismo



- Utilizar a mesma base para ajustar um modelo e verificar a qualidade do ajuste pode levá-lo ao princípio do otimismo. No caso acima parecia que o modelo iria generalizar bem, mas ao utilizá-lo em uma base teste contamos que isto não ocorreu. Esta prática pode levar ao overfitting ou underfitting

Divisão da Base de Dados

- Para não ter problemas com o princípio do otimismo uma boa prática na modelagem é dividir a base de dados em treinamento, validação e teste.



Base utilizada ajustar vários modelos

Verifica a qualidade do ajuste para cada um dos modelos ajustados na base de treinamento e escolhe o melhor modelo: Comparação, Seleção e Ajuste

Valida o modelo escolhido, pela base de validação.

- É comum utilizar apenas as base de Treinamento e validação. Muitas vezes por falta de dados. Nestes casos é usual usar as seguintes proporções: 80/20, 70/30

O que é Overfitting/Underfitting em um Modelo?

- Quando ajustamos um modelo nos dados de treinamento, uma das duas coisas pode acontecer:
  - superajustar o modelo
  - subajustar o modelo.

- Não queremos que essas coisas aconteçam, pois elas afetam a previsibilidade do modelo. A ideia é encontrar um modelo que se comporta de forma parecida na base de treinamento e de validação.

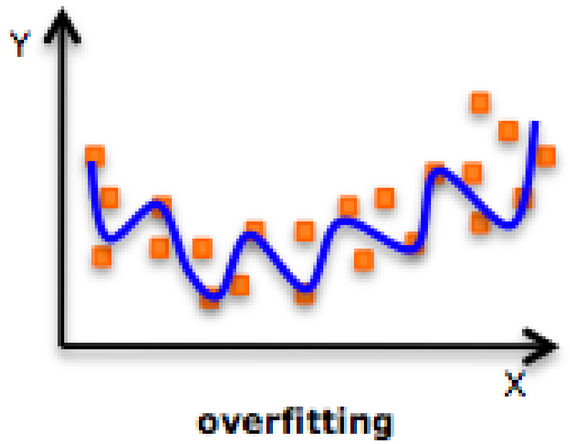
Overfitting

- O Overfitting significa que o modelo que treinamos treinou "muito bem", ou seja, se ajusta muito bem ao conjunto de dados de treinamento.

- Por que o overfitting ocorre?

  - Geralmente porque o modelo ajustado é muito complexo, ou seja, utiliza muitas características/variáveis em relação ao número de observações. Este modelo é muito preciso nos dados de treinamento, mas provavelmente não será preciso em dados novos.

  - Porque o modelo aprende ou descreve o "ruído" nos dados de treinamento em vez das relações reais entre as variáveis nos dados. Esse ruído, obviamente, não faz parte de nenhum novo conjunto de dados, e não pode ser aplicado a ele.



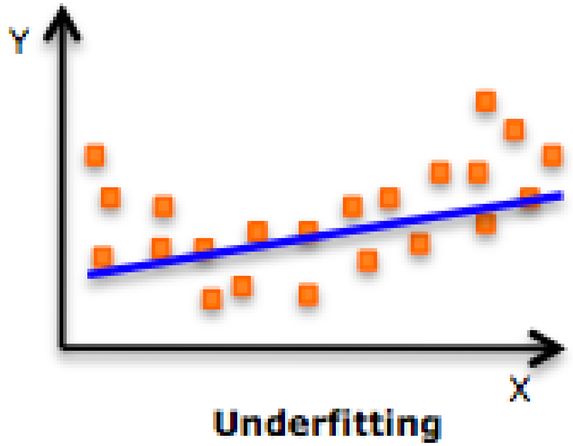
Underfitting

- O Underfitting significa que o modelo ajustado não corresponde aos dados de treinamento e, portanto, perde as tendências nos dados e não pode ser generalizado para novos dados.

- Por que o underfitting ocorre?

  - Geralmente é o resultado de um modelo muito simples, o qual não tem variáveis preditoras/independentes suficientes.

  - Pode ser resultado do uso de um modelo linear para dados que não são lineares. O que acarretará em uma capacidade de previsão pobre em dados de treinamento e não pode ser generalizado para outros dados.



Estudo de Caso
Cross Validation no Python

Fonte da dados: 

Link: http://scikit-learn.org/stable/modules/generated/sklearn.datasets.load_diabetes.html
Resumo: 442 pacientes com diabetes foram medidos para 10 variáveis (idade, sexo, bmi (índice de massa corporal), map (pressão arterial média), seis medidas de soro sanguíneo)
Objetivo: Ajustar um modelo de previsão, em uma base de treinamento, para a medida da progressão da doença um ano após o ponto de partida (variável resposta), fazer a previsão desta resposta e avaliar a qualidade de ajuste do modelo em uma base de teste.

Estudo de Caso
Cross Validation no Python

Parte_1: Dividindo a base em treinamento e teste
Parte_2: Ajustando um modelo linear a base de treinamento

Divisão da Base de Dados – Problemas?

- Parece bom, certo?

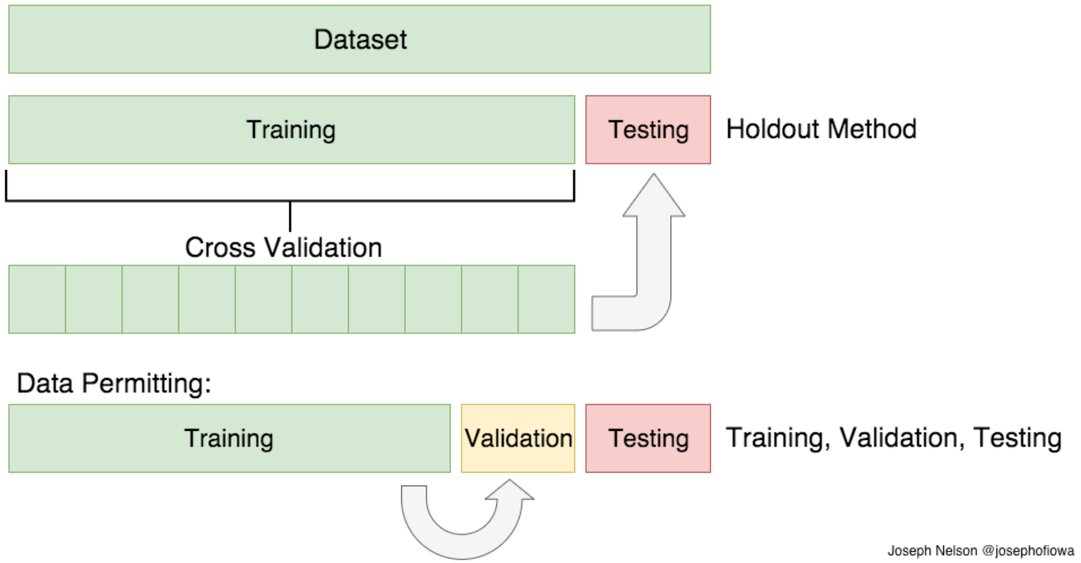
- Mas a divisão em treinamento/teste tem seus perigos:
  - E se a divisão não for aleatória?
  - E se um subconjunto dos dados tiver apenas pessoas de um determinado estado ou funcionários com um certo nível de renda, mas não outros níveis de renda, apenas mulheres ou apenas pessoas com certa idade?

  Isso pode resultar em overfitting, mesmo que tentemos evitá-lo!

- É aqui que entra a validação cruzada.

Cross Validation

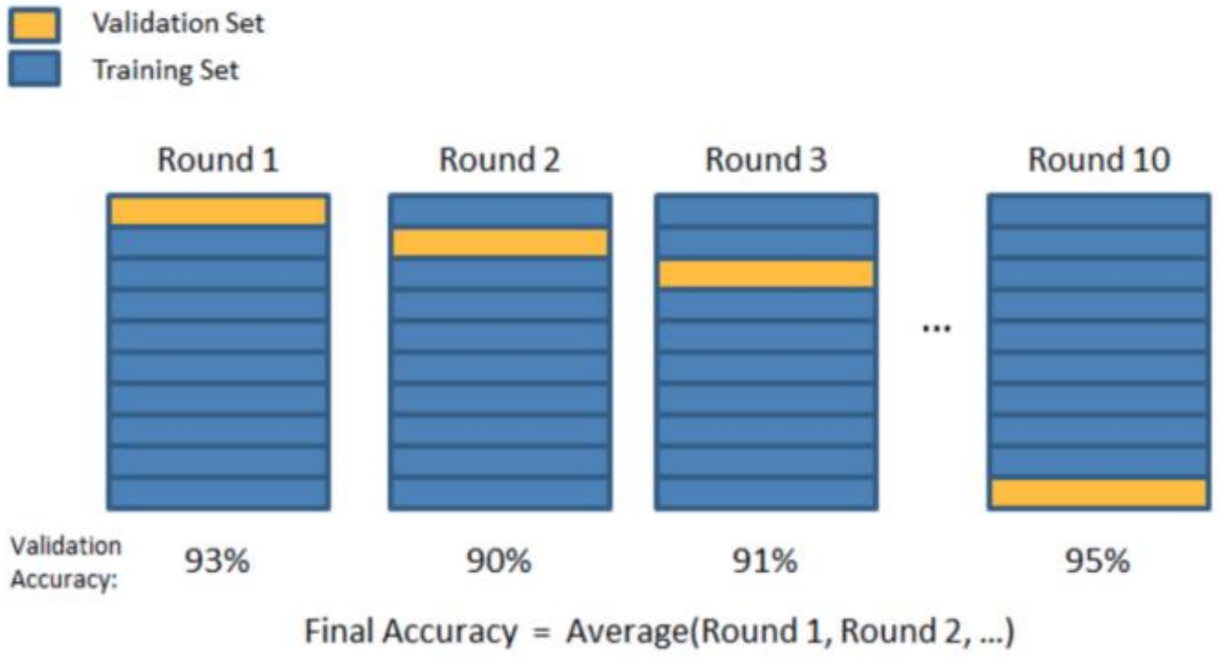
- É muito parecida com a divisão de treinamento/teste, mas é aplicado a mais subconjuntos.
  Dividimos os dados em k subconjuntos e treinamos em k-1 um desses subconjuntos.
  Sempre mantendo o último subconjunto para teste.
  A ideia é ir alternando até que todos os subconjuntos tenham sido usados para teste.



- Há vários métodos de validação cruzada, vamos passar por dois deles
  - K-Folds Cross Validation
  - Leave One Out Cross Validation (LOOCV)

K-Folds Cross Validation

1. Dividir os dados em k subconjuntos diferentes (ou dobras).
2. Usar k-1 subconjuntos para treinar os dados e deixar o último subconjunto (ou a última dobra) como teste.
3. Medir o modelo contra cada uma das dobras e finalizá-lo.
4. Testar o modelo na base de teste.
5. A acurácia final é obtida pela acurácia média de cada um dos modelos ajustado.

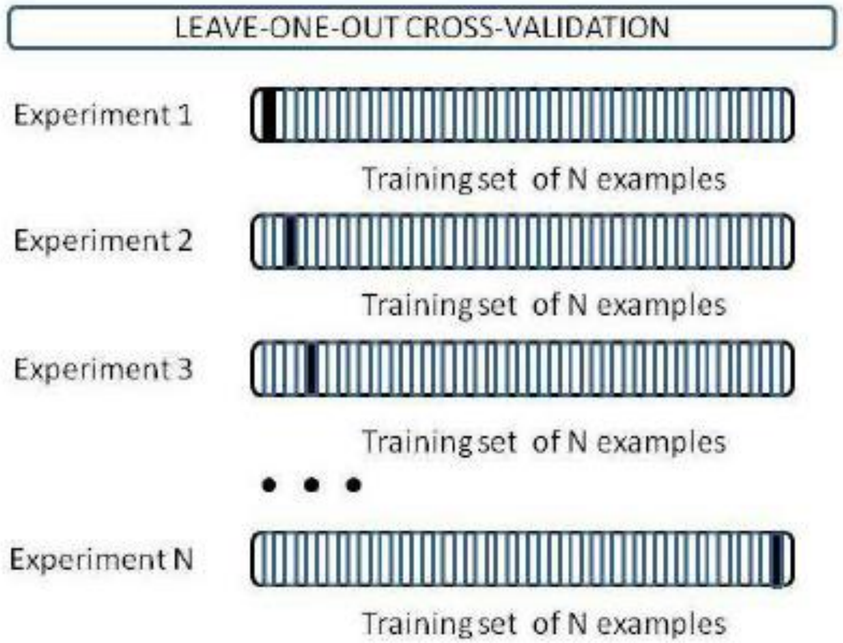


Estudo de Caso
Cross Validation no Python

Parte_3: Performando K-Folds Cross Validation

Leave One Out Cross Validation (LOO)

- A ideia de validação cruzada é igual a do método do K-fold, a única diferença é o número de folds. Neste caso o número de folds é igual ao número de observações que temos na base de dados.
- Como geralmente um grande número de conjuntos de treinamento é obtido, esse método é muito caro e deve ser usado apenas em pequenos conjuntos de dados.
  - Se o conjunto de dados for grande, provavelmente será melhor usar um método diferente, como o K-fold.



Estudo de Caso
Cross Validation no Python

Parte_4: Performando Leave One Out Cross Validation

Cross Validation – Outros métodos

- K-Fold estratificada
- Leave P Out (LPO)
- Leave One Label Out (LOLO)
- Leave P Label Out (LPLO)

Desafio Alunos
Cross Validation no Python

1. Dividir os dados em train e validate
2. Use a base de treinamento para ajustar um modelo (ou mais modelos)
3. Avalie o(s) modelo(s) usando a base de validação
4. Utilize um método de validação cruzada para verificar a qualidade de ajuste no modelo

DÚVIDAS?!

Referências

1. https://medium.com/towards-data-science/train-test-split-and-cross-validation-in-python-80b61beca4b6

2. http://scikit-learn.org/stable/modules/cross_validation.html

3. http://ogrisel.github.io/scikit-learn.org/sklearn-tutorial/modules/cross_validation.html

4. http://scikit-learn.org/stable/auto_examples/model_selection/plot_learning_curve.html

Curiosidades

1. Bibliotecas para Data Science no Python
https://medium.com/activewizards-machine-learning-company/top-15-python-libraries-for-data-science-in-in-2017-ab61b4f9b4a7

Obrigada

Cristiane Rodrigues

crisrodrigues_27@hotmail.com