Aprendizado de Máquinas (Machine Learning)

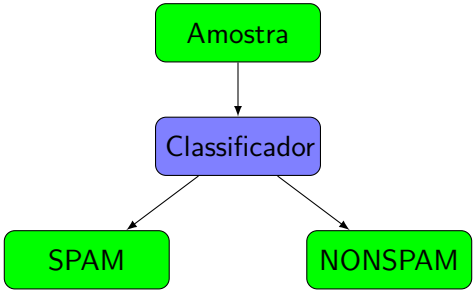
Douglas Rodrigues

Universidade Federal Fluminense

Aulas Anteriores

- Exemplo de aplicação de aprendizado de máquinas (classificação).

- Exemplo de aplicação de aprendizado de máquinas (classificação).

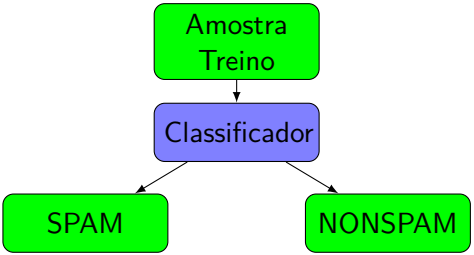


Tipos de Erro

- Erro amostral (*In sample error*): erro observado na amostra treino.

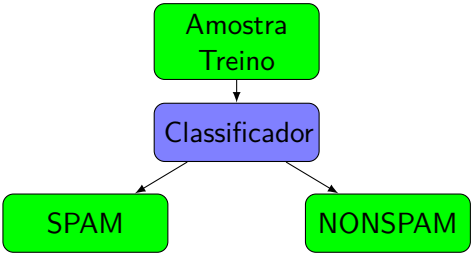
Tipos de Erro

- Erro amostral (*In sample error*): erro observado na amostra treino.



Tipos de Erro

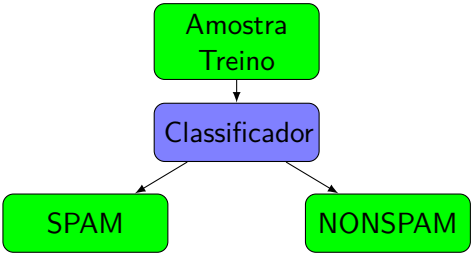
- Erro amostral (*In sample error*): erro observado na amostra treino.



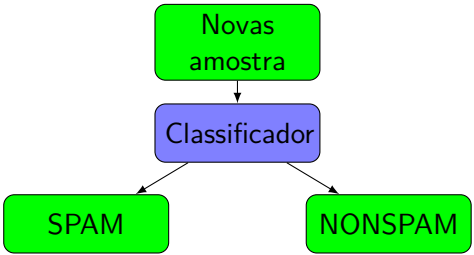
- Erro fora da amostra (*Out of sample error*): Erro nas novas amostras.

Tipos de Erro

- Erro amostral (*In sample error*): erro observado na amostra treino.



- Erro fora da amostra (*Out of sample error*): Erro nas novas amostras.



O que é o Aprendizado de Máquina?

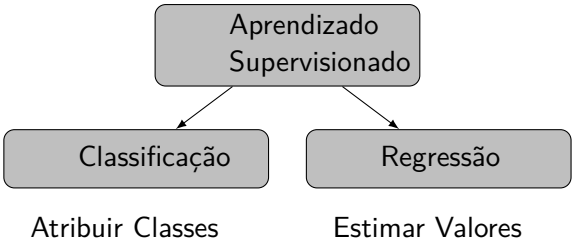
- Em 1959, Arthur Samuel definiu o aprendizado de máquina como o "campo de estudo que dá aos computadores a habilidade de aprender sem serem explicitamente programados".
- Ou seja, é um método de análise de dados que automatiza a construção de modelos analíticos.
- É baseado na ideia de que sistemas podem aprender com dados, identificar padrões e tomar decisões com o mínimo de intervenção humana.

Tipos de Aprendizado

- **Supervisionado:** temos acesso a exemplos com rótulos definidos por especialistas, e procuramos um estimador que possa rotular novas amostras não-rotuladas com maior precisão possível.

Tipos de Aprendizado

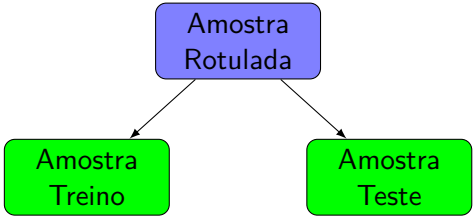
- **Supervisionado:** temos acesso a exemplos com rótulos definidos por especialistas, e procuramos um estimador que possa rotular novas amostras não-rotuladas com maior precisão possível.



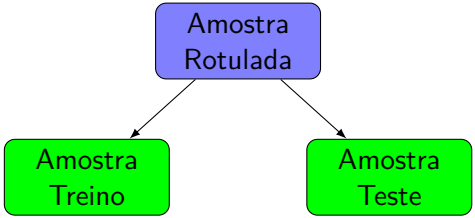
Tipos de Aprendizado

- **Não-Supervisionado:** não há exemplos rotulados, buscamos classificar os elementos baseado nas similaridades entre suas características. Exemplo: clusterização *k-means*.
- **Aprendizado por reforço:** é uma técnica que tem como objetivo treinar um agente a interagir em um ambiente por meio de ações para atingir um determinado objetivo. Exemplo: treinar um carro para correr em uma pista virtual, ensinar destreza a um robô.

Aprendizado Supervisionado: Ideia Central



Aprendizado Supervisionado: Ideia Central



Utilizamos essa amostra para treinar o classificador/regressor

Utilizamos essa amostra para estimar o **erro fora da amostra**

Design de Predição

1. Definimos taxa de erro aceitável (*benchmark*)

Design de Predição

1. Definimos taxa de erro aceitável (*benchmark*)
2. Separamos nossa amostra rotulada em 2 grupos: TREINO E TESTE.

Design de Predição

1. Definimos taxa de erro aceitável (*benchmark*)
2. Separamos nossa amostra rotulada em 2 grupos: TREINO E TESTE.
3. Realizamos o pré-processamento dos dados: definimos quais variáveis serão utilizadas para estimação dos parâmetros da função preditora, podemos realizar padronização dos dados, aplicar PCA, etc.

Design de Predição

1. Definimos taxa de erro aceitável (*benchmark*)
2. Separamos nossa amostra rotulada em 2 grupos: TREINO E TESTE.
3. Realizamos o pré-processamento dos dados: definimos quais variáveis serão utilizadas para estimação dos parâmetros da função preditora, podemos realizar padronização dos dados, aplicar PCA, etc.
4. Definimos o método que será utilizado para construção da função preditora.

Design de Predição

1. Definimos taxa de erro aceitável (*benchmark*)
2. Separamos nossa amostra rotulada em 2 grupos: TREINO E TESTE.
3. Realizamos o pré-processamento dos dados: definimos quais variáveis serão utilizadas para estimação dos parâmetros da função preditora, podemos realizar padronização dos dados, aplicar PCA, etc.
4. Definimos o método que será utilizado para construção da função preditora.
5. Utilizando a amostra TREINO, definimos os parâmetros da função preditora, obtendo o melhor modelo possível (treinamos o modelo).

Design de Predição

1. Definimos taxa de erro aceitável (*benchmark*)
2. Separamos nossa amostra rotulada em 2 grupos: TREINO E TESTE.
3. Realizamos o pré-processamento dos dados: definimos quais variáveis serão utilizadas para estimação dos parâmetros da função preditora, podemos realizar padronização dos dados, aplicar PCA, etc.
4. Definimos o método que será utilizado para construção da função preditora.
5. Utilizando a amostra TREINO, definimos os parâmetros da função preditora, obtendo o melhor modelo possível (treinamos o modelo).
6. Agora, aplicamos o melhor modelo obtido na amostra TESTE **uma única vez**, para estimar o **erro fora da amostra** da função preditora

Observações

- Por que aplicamos o melhor apenas uma vez na amostra TESTE?
  A amostra teste serve para estimar qual será o erro da nossa função preditora quando utilizarmos amostras novas, sem rótulos. Se aplicamos diversas vezes para tentar melhorar a função preditora, estaremos utilizando a amostra TESTE como TREINO.

Observações

- Por que aplicamos o melhor apenas uma vez na amostra TESTE?
  A amostra teste serve para estimar qual será o erro da nossa função preditora quando utilizarmos amostras novas, sem rótulos. Se aplicamos diversas vezes para tentar melhorar a função preditora, estaremos utilizando a amostra TESTE como TREINO.

  AMOSTRA TREINO: Utilizamos para treinar o modelo (estimar os parâmetros do classificador)
  AMOSTRA TESTE: Serve para estimar o erro do classificador quando utilizamos amostras não rotuladas.

Observações

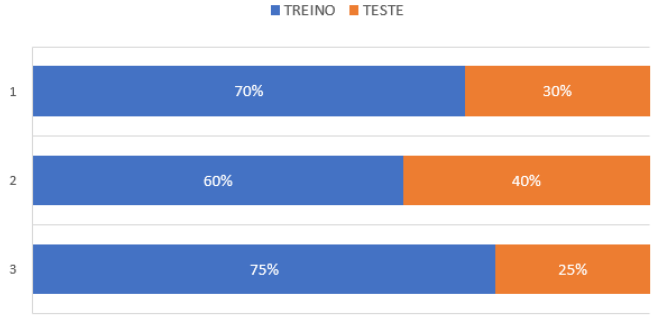
- Como comparar dois classificadores distintos, para definir o melhor?

  Há diversas métricas que devem ser levadas em consideração, que envolvem desde a acurácia até o custo computacional do modelo.
  Veremos posteriormente os métodos para comparação de modelos.

Observações

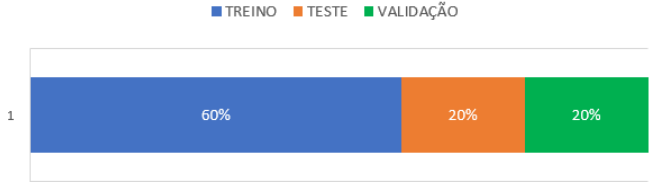
- Qual tamanho ideal para as amostras TREINO/TESTE?

  Não existe nenhuma regra quanto a isso. Na prática, quando há uma quantidade razoável de dados, normalmente dividimos os dados em 70% para treino e 30% para teste, mas não é incomum encontrar 60% e 40% ou 75% e 25%.

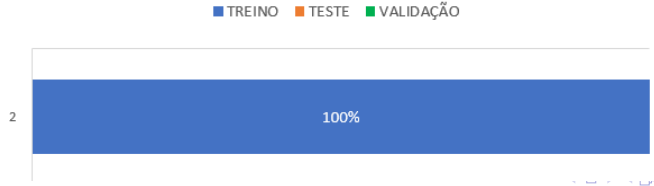


Observações

Quando há uma quantidade muito grande, é possível dividir os dados em 3 conjuntos: treino, teste e validação, normalmente na proporção 60/20/20. Abordaremos esse tema posteriormente.



Para amostras muito pequenas, utilizamos validação cruzada (*cross validation*), e reportamos a advertência de banco de dados pequeno.



Exemplo: Classificação

```
> library(caret)
> library(kernlab)
> data(spam)
#y=classe dos dados, para manter a mesma proporção nos conjuntos TREINO/TESTE
#p=porcentagem p/ grupo de TREINO
#Retorna numero da linha a ser selecionada
inTrain <- createDataPartition(y=spam$type,p=0.70,list=F)
```

Separação Treino/Teste

```
#Separamos linhas para amostra treino
 training <- spam[inTrain,]

#Separamos linhas para amostra teste
testing <- spam[-inTrain,]
```

Separação Treino/Teste

#Separamos linhas para amostra treino

```
training <- spam[inTrain,]
```

#Separamos linhas para amostra teste

```
testing <- spam[-inTrain,]
```

#O comando `createDataPartition()` deve ser utilizando quando os dados são amostras independentes.

#Quando há evolução temporal dos dados, deve-se utilizar `createTimeSlices()`.

createDataPartition()

#Vamos verificar que as proporções entre spam e não-spam são as mesmas entre as amostras

```
> library(epiDisplay)

> tab1(spam$type)

> tab1(training$type)

> tab1(training$type)
```