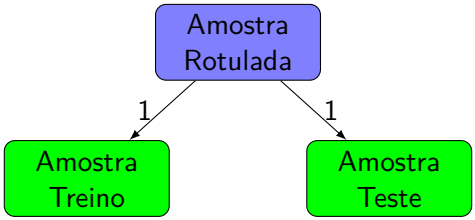
Aprendizado de Máquinas (Machine Learning)

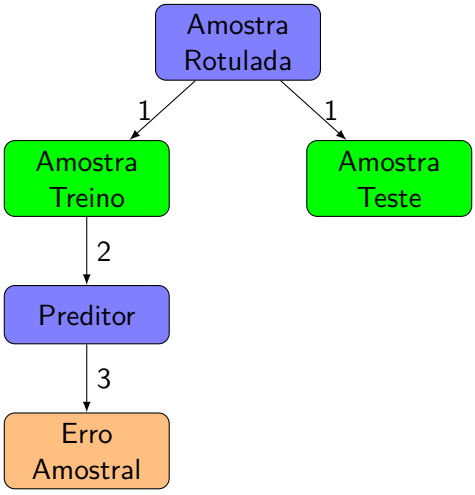
Douglas Rodrigues

Universidade Federal Fluminense

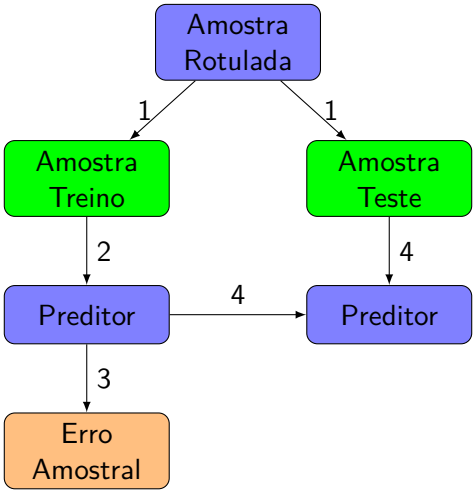
Última Aula



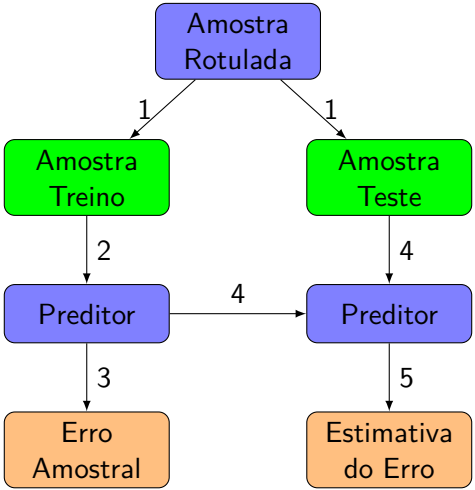
Última Aula


Amostra Rotulada
1
1
Amostra Treino
Amostra Teste
2
Preditor
3
Erro Amostral

Última Aula


Amostra Rotulada
1
1
Amostra Treino
Amostra Teste
2
4
Preditor
4
Preditor
3
Erro Amostral

Última Aula


Amostra Rotulada
1
1
Amostra Treino
Amostra Teste
2
4
Preditor
4
Preditor
3
5
Erro Amostral
Estimativa do Erro

Realizando o Treinamento: `train()`

- Vamos utilizar o pacote `caret` (***C**lassification **A**nd **RE**gression **T**raining*), criado por Max Kuhn.
- O primeiro passo para TREINAR um modelo é realizar a separação das amostras TREINO/TESTE, e escolher qual modelo será implantado.
- Para conhecer todos os modelos implementados no `caret`, basta digitar o seguinte comando:

```
> names(getModelInfo())
```

Exemplo: Classificação

```
> library(caret)
> library(kernlab)
> data(spam)

> inTrain <- createDataPartition(y=spam$type,p=0.70,list=F)

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

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

Exemplo - Classificação

Vamos aplicar o modelo **glm** (Modelos Lineares Generalizados) para tentar classificar os emails em **spam** ou **nonspam**.

```
#Criamos o modelo, utilizando a amostra TREINO.
> modelFit <- train(type ~ .  , data=training, method="glm")
```

Exemplo - Classificação

Vamos aplicar o modelo **glm** (Modelos Lineares Generalizados) para tentar classificar os emails em **spam** ou **nonspam**.

#Criamos o modelo, utilizando a amostra TREINO.

```
> modelFit <- train(type ~ .  , data=training, method="glm")
```

#Vamos verificar o erro amostral.

```
> modelFit
```

Exemplo - Classificação

```
Generalized Linear Model

3451 samples
  57 predictor
   2 classes:  'nonspam', 'spam'

No pre-processing Resampling:  Bootstrapped (25 reps) Summary of
sample sizes:  3451, 3451, 3451, 3451, 3451, 3451, ...
Resampling results:

  Accuracy   Kappa
  0.9176549  0.828082
```

Exemplo - Classificação

#Aplicamos o modelo criado na amostra TESTE, para estimarmos a precisão do classificador.

```
> prediction <- predict(modelFit,newdata=testing)
```

Exemplo - Classificação

#Aplicamos o modelo criado na amostra TESTE, para estimarmos a precisão do classificador.

```
> prediction <- predict(modelFit,newdata=testing)
```

#Realizamos a avaliação do modelo, comparando os resultados da amostra TESTE.

```
> confusionMatrix(prediction,testing$type)
```

Matriz de Confusão

```
Confusion Matrix and Statistics

          Reference
Prediction nonspam spam
     nonspam    659   48
     spam        38  405

Accuracy :  0.9252
95% CI : (0.9085, 0.9398)

Sensitivity :     0.9455
Specificity :     0.8940
Pos Pred Value :  0.9321
Neg Pred Value :  0.9142

'Positive' Class :  nonspam
```

Matriz de Confusão

```
            Reference
Prediction nonspam spam
     nonspam    659   48
     spam        38  405

'Positive' Class :  nonspam
```

Matriz de Confusão

```
           Reference
Prediction nonspam spam
    nonspam    659   48
    spam        38  405
```

```
'Positive' Class :  nonspam
```

TRADUÇÃO

```
                          Reference
Prediction          nonspam          spam
    nonspam    True Positive   False Positive
    spam       False Negative   True Negative
```

Matriz de Confusão

```
           Reference
Prediction nonspam spam
    nonspam    659   48
    spam        38  405
```

```
'Positive' Class :  nonspam
```

TRADUÇÃO

```
           Reference
Prediction nonspam spam
    nonspam    TP   FP
    spam       FN   TN
```

Avaliando Classificadores

- **Accuracy (Precisão):** taxa de acerto do classificador.

$$\frac{\text{True Positive} + \text{True Negative}}{\text{n}}$$

Avaliando Classificadores

- **Accuracy (Precisão):** taxa de acerto do classificador.

$$\frac{\text{True Positive} + \text{True Negative}}{\text{n}}$$

- **Sensitivity (Sensibilidade):** taxa de acertar os casos que são positivos.

$$\frac{\text{True Positive}}{\text{True Positive} + \text{False Negative}}$$

Avaliando Classificadores

- **Accuracy (Precisão):** taxa de acerto do classificador.

$$\frac{\text{True Positive} + \text{True Negative}}{\text{n}}$$

- **Sensitivity (Sensibilidade):** taxa de acertar os casos que são positivos.

$$\frac{\text{True Positive}}{\text{True Positive} + \text{False Negative}}$$

- **Specificity (Especificidade):** taxa de acertar os casos que são negativos.

$$\frac{\text{True Negative}}{\text{True Negative} + \text{False Positive}}$$

ATENÇÃO

- Imagine a seguinte situação: temos uma amostra com dados de diversos pacientes, onde 95% são saudáveis e 5% são portadores de uma doença.
- Queremos criar um CLASSIFICADOR para tentar identificar se a pessoa é saudável ou não.

ATENÇÃO

- Imagine a seguinte situação: temos uma amostra com dados de diversos pacientes, onde 95% são saudáveis e 5% são portadores de uma doença.
- Queremos criar um CLASSIFICADOR para tentar identificar se a pessoa é saudável ou não.
- Olhando apenas para a métrica **ACURÁCIA**, basta eu classificar TODOS pacientes como saudáveis, que terei 95% de acerto.

ATENÇÃO

- Imagine a seguinte situação: temos uma amostra com dados de diversos pacientes, onde 95% são saudáveis e 5% são portadores de uma doença.
- Queremos criar um CLASSIFICADOR para tentar identificar se a pessoa é saudável ou não.
- Olhando apenas para a métrica **ACURÁCIA**, basta eu classificar TODOS pacientes como saudáveis, que terei 95% de acerto.
- No entanto, se observar a métrica **ESPECIFICIDADE**, teremos 0%. Eis a importância de não observar apenas acurácia.

PRÓXIMA AULA

- Treinando REGRESSORES.
- Como avaliar regressores.