THAÍS GAUDENCIO
GAUDENCIOTHAIS@GMAIL.COM

Avaliação de Modelos Preditivos


Hoje é o dia mais importante da tua vida

Métricas de Erro

Análise do desempenho do preditor gerado por ele na rotulação de novos objetos, não apresentados previamente em seu treinamento.

Métricas para Classificação

$$err(\hat{f}) = \frac{1}{n}\sum_{i=1}^{n} I(y_i \neq \hat{f}(x_i))$$

- TAXA DE ERRO OU DE CLASSIFICAÇÕES INCORRETAS.

- I(A) = 1, SE A É VERDADEIRO E 0, EM CASO CONTRÁRIO.

Tamanho (P)	Largura (P)	Tamanho (S)	Largura (S)	Espécie
5,1	*3,5*	*1,4*	*0,2*	*Setosa*
4,9	*3,0*	*1,4*	*0,2*	*Setosa*
7,0	*3,2*	*4,7*	*1,4*	*Versicolor*
6,4	*3,2*	*4,5*	*1,5*	*Versicolor*

Métricas para Classificação

- A TAXA DE ERRO VARIA ENTRE 0 E 1, E VALORES PRÓXIMOS AO EXTREMO 0 SÃO MELHORES.

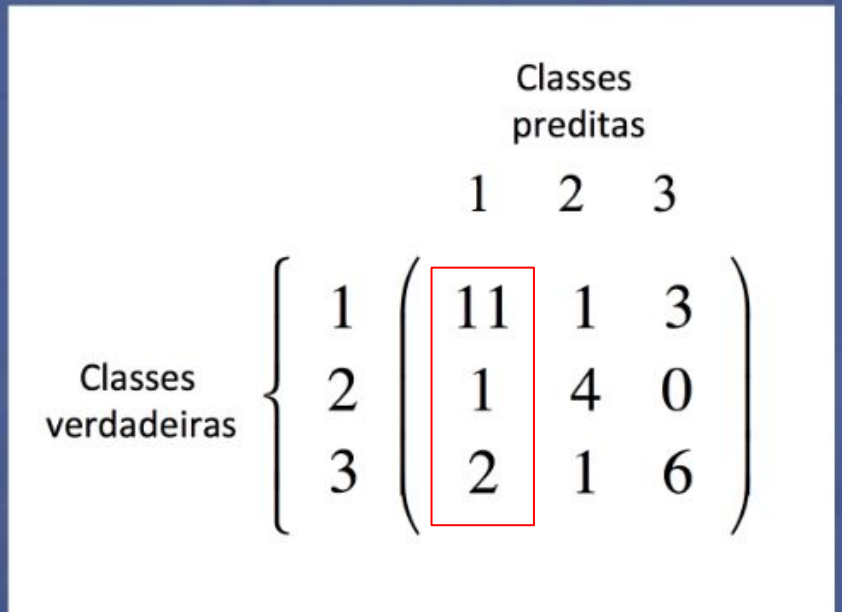
- O COMPLEMENTO DESSA TAXA CORRESPONDE A TAXA DE ACERTO OU ACURÁCIA DO CLASSIFICADOR.

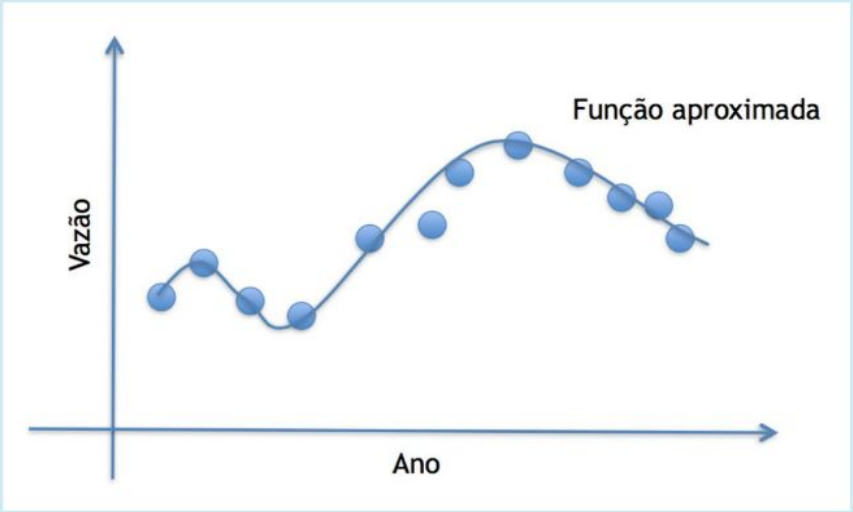
$$ac(\hat{f}) = 1 - err(\hat{f})$$

- NESSE CASO, VALORES PRÓXIMOS DE 1 SÃO CONSIDERADOS MELHORES.

Matriz de confusão

- MATRIZ QUE ILUSTRA O NÚMERO DE PREDIÇÕES CORRETAS E INCORRETAS EM CADA CLASSE.





Métricas para Regressão

• O erro da hipótese f pode ser calculado pela distância entre o valor yi conhecido e aquele predito pelo modelo, ou seja, f(xi).

• As medidas de erro mais conhecidas e usadas nesse caso são o erro quadrático médio (MSE) - mean squared error e a distância absoluta média (MAD)– mea absolute distance.

$$MSE(\hat{f}) = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{f}(x_i))^2$$

$$MAD(\hat{f}) = \frac{1}{n}\sum_{i=1}^{n}\left|y_i - \hat{f}(x_i)\right|$$

Métricas para Regressão

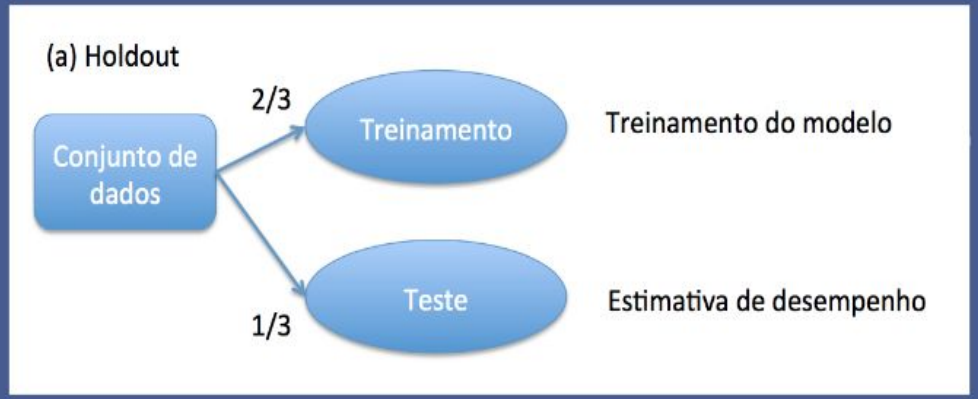
O MSE e MAD são sempre não negativos. Para ambas as medidas, valores mais baixos correspondem a melhores modelos, ou seja, melhores aproximações dos rótulos verdadeiros dos objetos.

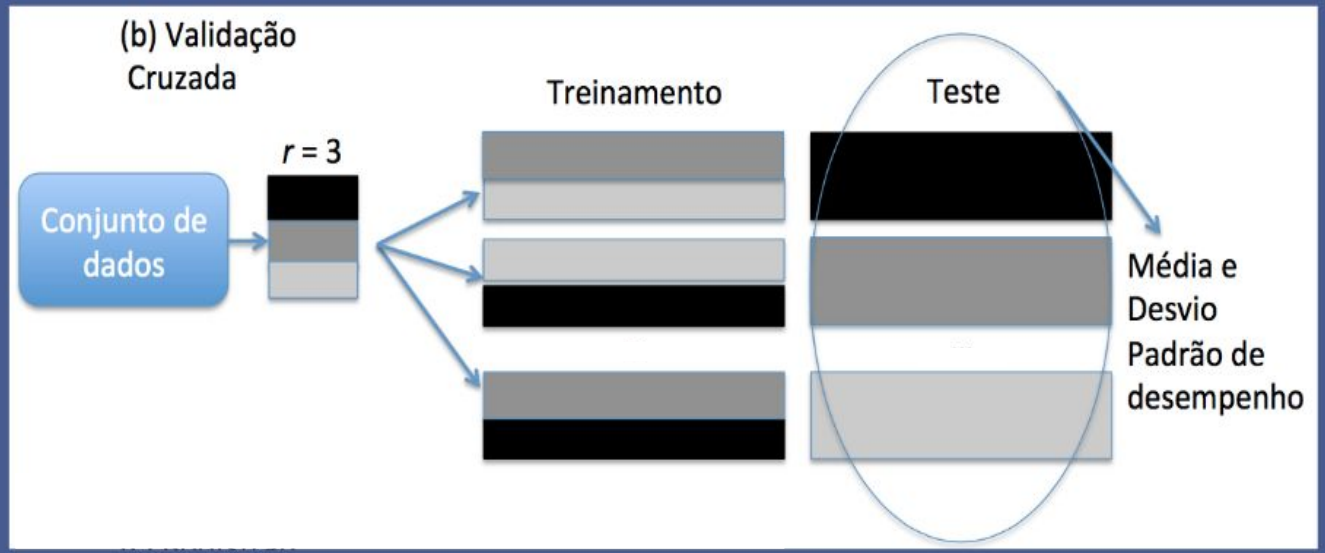
Amostragem

Definem-se subconjuntos de treinamento e teste para obtenção de estimativas de predição.

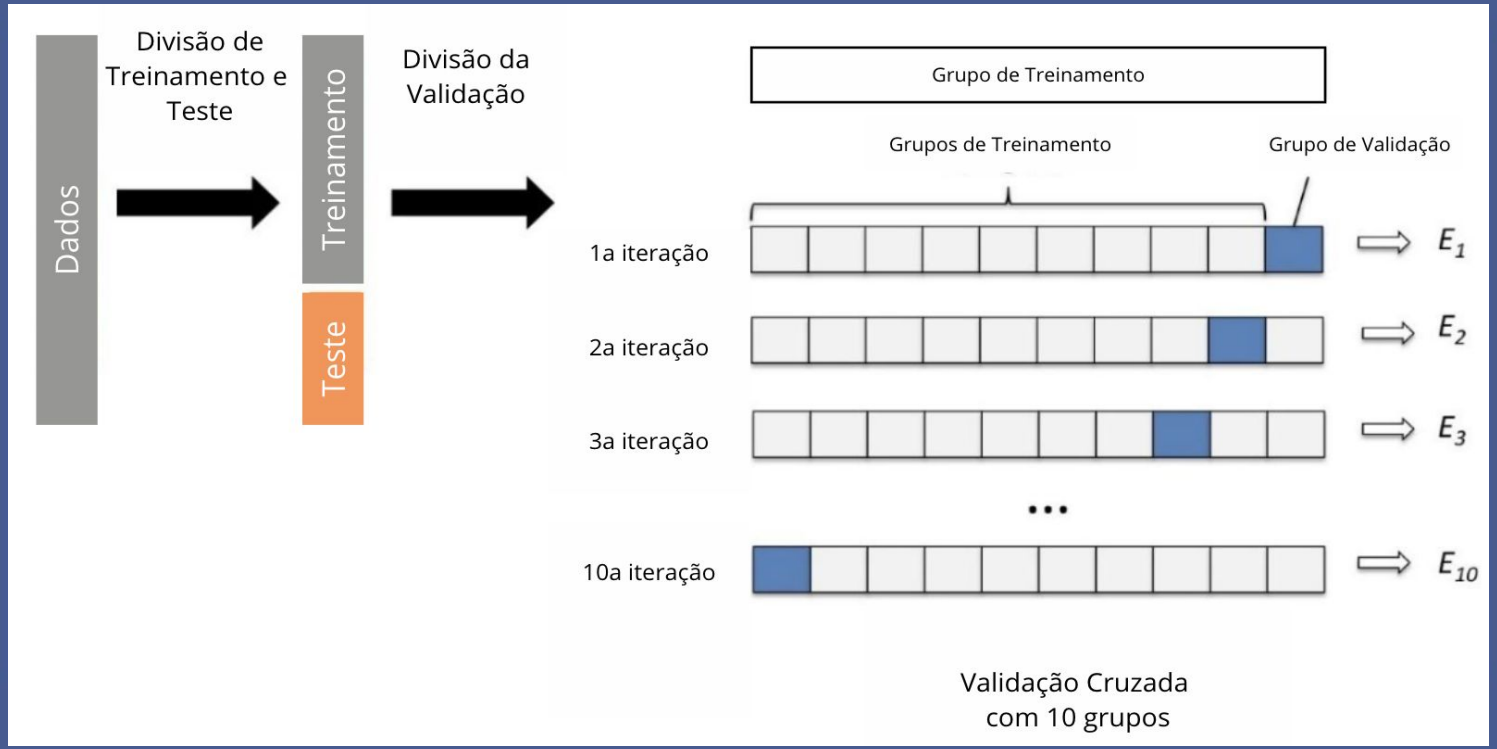
Esses subconjuntos são disjuntos para assegurar que as medidas de desempenho sejam obtidas a partir de um conjunto de exemplos diferente daquele usado no aprendizado.

Amostragem


(a) Holdout
2/3
Conjunto de dados
Treinamento
Treinamento do modelo
Teste
Estimativa de desempenho
1/3


(b) Validação Cruzada
Treinamento
Teste
r = 3
Conjunto de dados
Média e Desvio Padrão de desempenho

Amostragem


Dados
Divisão de Treinamento e Teste
Treinamento
Teste
Divisão da Validação
Grupo de Treinamento
Grupos de Treinamento
Grupo de Validação
1a iteração
2a iteração
3a iteração
10a iteração
E_1
E_2
E_3
E_{10}
Validação Cruzada com 10 grupos

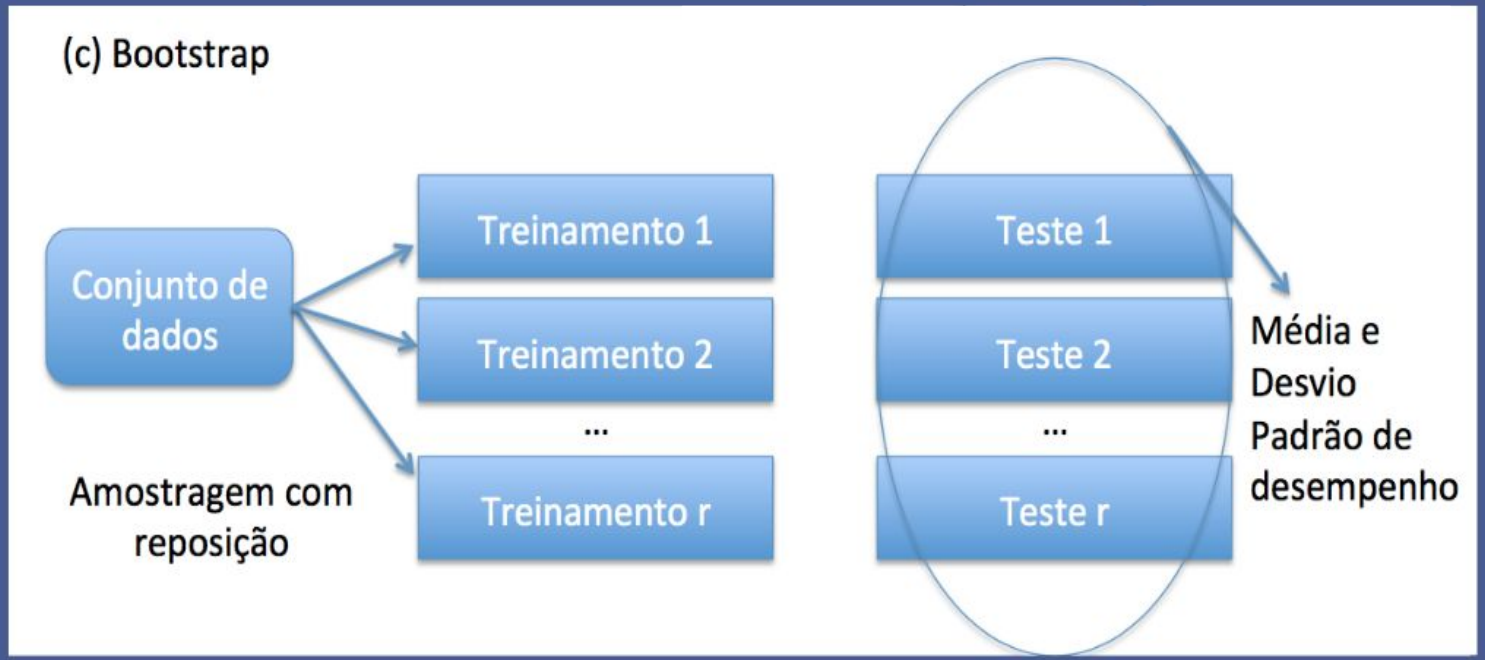
LEMBRANDO QUE O NORMAL É 70/30

Amostragem

```
trainSample = data.sample(frac=0.8, random_state = 1)
```

```
from sklearn.model_selection import KFold

scores = []
cv = KFold(n_splits=10, shuffle=True)

for train_index, test_index in cv.split(X):
    X_train, X_test, y_train, y_test = X[train_index], X[test_index], y[train_index], y[test_index]

    reg.fit(X_train, y_train)

    scores.append(reg.score(X_test, y_test))
```

Amostragem


(c) Bootstrap
Conjunto de dados
Amostragem com reposição
Treinamento 1
Treinamento 2
...
Treinamento r
Teste 1
Teste 2
...
Teste r
Média e Desvio Padrão de desempenho

Holdout

Amostragem

- UMA PROPORÇÃO DE P PARA TREINAMENTO E (1-P) PARA TESTE;
- NORMALMENTE, EMPREGA-SE P=2/3;
- GERALMENTE USADA QUANDO UM CONJUNTO DE DADOS É GRANDE O SUFICIENTE;
- NÃO PERMITE AVALIAR O QUANTO O DESEMPENHO EM RELAÇÃO A CONSTRUÇÃO DE CONJUNTOS COM DIFERENTES COMBINAÇÕES DE OBJETOS.
- SOLUÇÃO: RANDOM SUBSAMPLING.

Validação Cruzada

Amostragem

No Leave-One-Out, o modelo final para produção é treinado com todos os dados disponíveis, após a validação feita com as múltiplas divisões de treino e teste.

- O CONJUNTO DE EXEMPLOS É DIVIDIDO EM R SUBCONJUNTOS DE TAMANHO APROXIMADAMENTE IGUAL;

- R-FOLD CROSS VALIDATION ESTRATIFICADO: MANTÉM EM CADA PARTIÇÃO A PROPORÇÃO DE EXEMPLOS DE CADA CLASSE SEMELHANTE A PROPORÇÃO CONTIDA NO CONJUNTO DE DADOS TOTAL;

- LEAVE-ONE-OUT: A CADA CICLO EXATAMENTE UM EXEMPLO É SEPARADO PARA TESTE, ENQUANTO N-1 EXEMPLOS RESTANTES SÃO UTILIZADOS NO TREINAMENTO DO PREDITOR.

Boostrap

Amostragem

- R SUBCONJUNTOS SÃO GERADOS A PARTIR DO CONJUNTO DE EXEMPLOS ORIGINAL;
- OS EXEMPLOS SÃO AMOSTRADOS ALEATORIAMENTE DESSE CONJUNTO, COM REPOSIÇÃO;
- NORMALMENTE, ADOTA-SE $R \geq 100$;
- GERALMENTE APLICADO EM AMOSTRAS DE DADOS PEQUENAS.

Conjunto validação

O conjunto validação fornece uma avaliação do ajuste do modelo ao conjunto treinamento enquanto é feito o ajuste dos hiperparâmetros, por exemplo, o número de camadas escondidas em uma rede neural.

Problema de Duas Classes
n=VP+VN+FP+FN

- **VP–VERDADEIROS POSITIVOS**

  objetos da classe positiva classificados corretamente

- **VN–VERDADEIROS NEGATIVOS**

  objetos da classe negativa classificados corretamente

- **FP–FALSOS POSITIVOS**

  classe verdadeira é negativa, mas que foram classificados incorretamente como da classe positiva

- **FN–FALSOS NEGATIVOS**

  classe verdadeira é positiva, mas que foram classificados incorretamente como da classe negativa

Matriz de Confusão

CLASSES PREDITAS

CLASSES VERDADEIRAS

	+	-
+	VP	FN
-	FP	VN

Medidas de Desempenho

- **ACURÁCIA TOTAL**

$$ac(\hat{f}) = \frac{VP + VN}{n}$$

- **PRECISÃO**

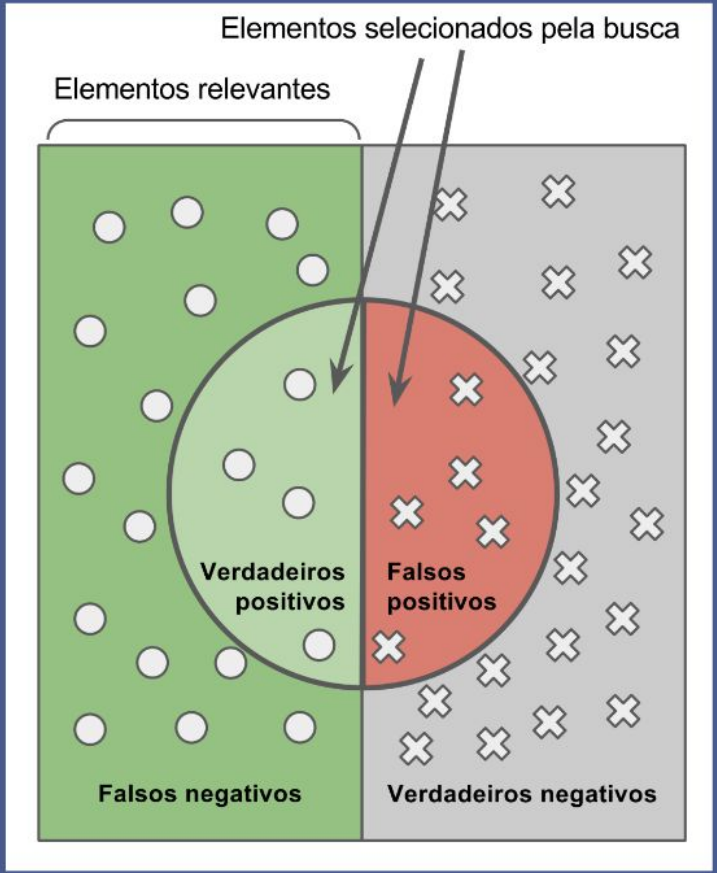
$$prec(\hat{f}) = \frac{VP}{VP + FP}$$

- **SENSIBILIDADE**

$$sens(\hat{f}) = rev(\hat{f}) = TVP(\hat{f}) = \frac{VP}{VP + FN}$$

- **ESPECIFICIDADE**

$$esp(\hat{f}) = \frac{VN}{VN + FP} = 1 - TFP(\hat{f})$$



Medidas de Desempenho

Precisão = $\frac{\text{verdadeiros positivos}}{\text{elementos selecionados}}$

"Quantos elementos selecionados são relevantes?"

Revocação = $\frac{\text{verdadeiros positivos}}{\text{elementos relevantes}}$

"Quantos elementos relevantes foram selecionados?"

```
print("\nK-NN")
print("Acurácia: %0.2f" % (metrics.accuracy_score(testTarget, resultKNN)))
print("Medida F1: %0.2f" % (metrics.f1_score(testTarget, resultKNN)))


matrizConfusao = metrics.confusion_matrix(testTarget, resultKNN)
print("Matriz de Confusão:\n",matrizConfusao)
print("Sensibilidade: %0.2f" % (matrizConfusao[0][0]/(matrizConfusao[0][0] + matrizConfusao[1][1])))
print("Especificidade: %0.2f" % (matrizConfusao[1][0]/(matrizConfusao[1][0] + matrizConfusao[0][1])))
```

```
K-NN
Acurácia: 0.78
Medida F1: 0.76
Matriz de Confusão:
 [[41 11]
 [ 9 31]]
Sensibilidade: 0.57
Especificidade: 0.45
```

Medidas de Desempenho

Medidas de Desempenho

- A precisão pode ser vista como uma medida de exatidão do modelo;
- A revocação é uma medida de completude;
- Geralmente a precisão e a revocação são combinadas em uma única medida, como a medida-F, que é a média harmônica ponderada da precisão e a revocação:

Lembrando que Revocação = Sensibilidade

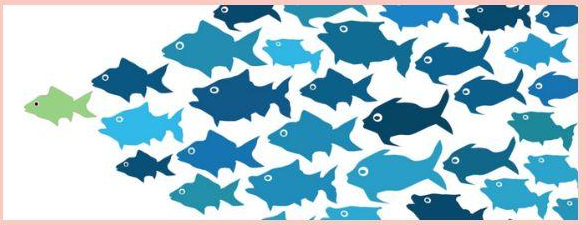
$$F = 2 \cdot \frac{\text{precis} \cdot \text{revoc}}{\text{precis} + \text{revoc}}$$

O número 2 na fórmula do F1-score aparece porque ele é a média harmônica entre a precisão (precision) e o recall (revocação)

Medidas de Desempenho

LEMBREM!!!!!

Principalmente em problemas de classes desbalanceadas, utilizar só a acurácia não é interessante, uma vez que, caso o modelo tenda a prever as instâncias, inclusive da classe minoritária como sendo da classe majoritária, essa métrica permanecerá alta, o que não demonstra a qualidade do modelo!!!



Coeficiente de Matthews

A

Actual \ Predicted	Control	Disease
Control	TN	FP
Disease	FN	TP

B

$$MCC = \frac{(TP \times TN) - (FP \times FN)}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}}$$

Diferentemente de métricas como a acurácia, o MCC avalia o desempenho do modelo de forma mais completa e justa. Ele varia de -1 a 1, onde 1 indica uma classificação perfeita, 0 uma classificação aleatória e -1 uma classificação totalmente incorreta.

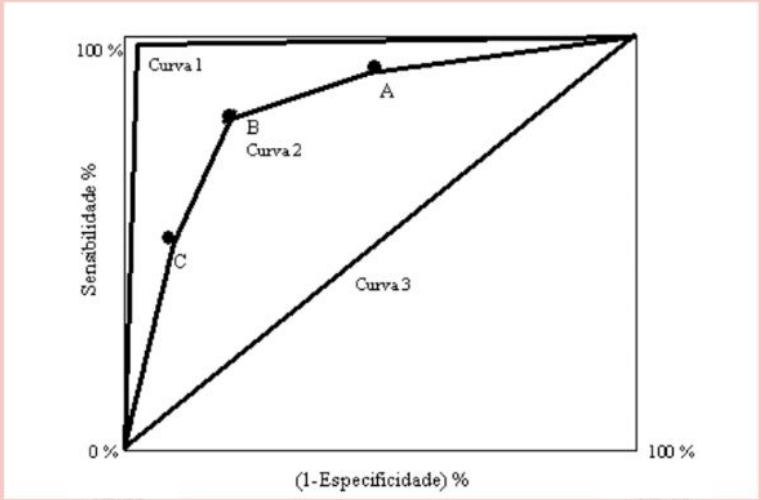
Medidas de Desempenho

Modelos muito complexas - chance alta de overfitting

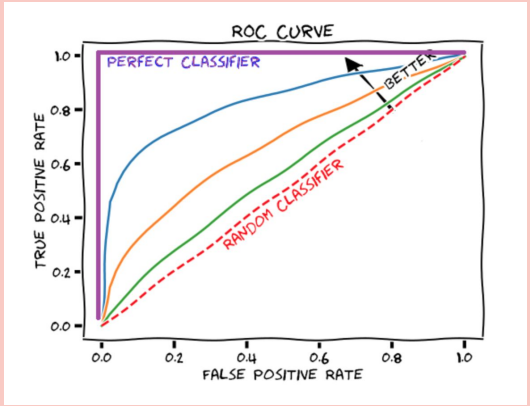
Modelos muito generalistas - chance alta deunderfitting

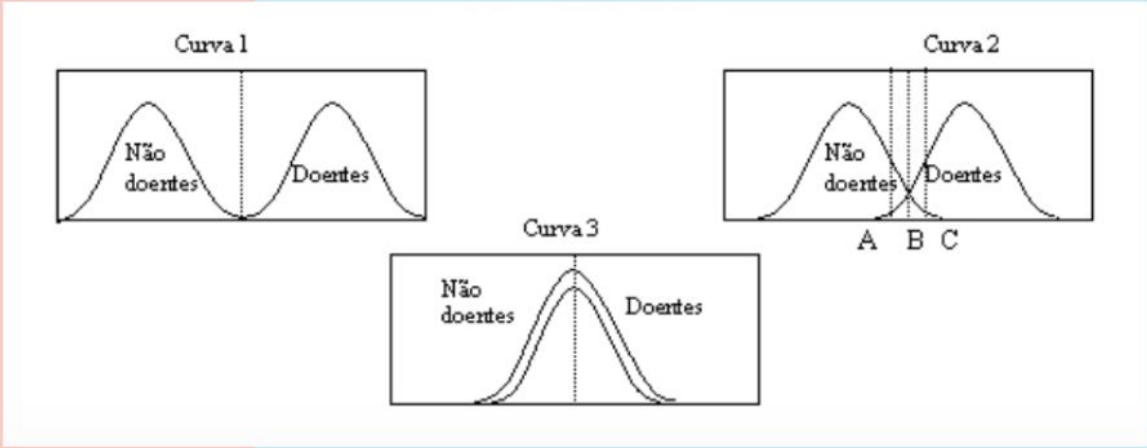
Métricas de desempenho ajudam a observar esses efeitos!!!

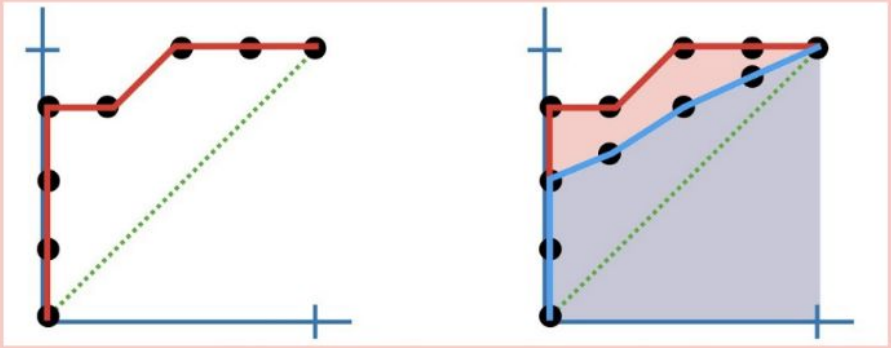
Cada ponto da curva ROC corresponde a um valor de corte diferente do modelo



CURVA ROC



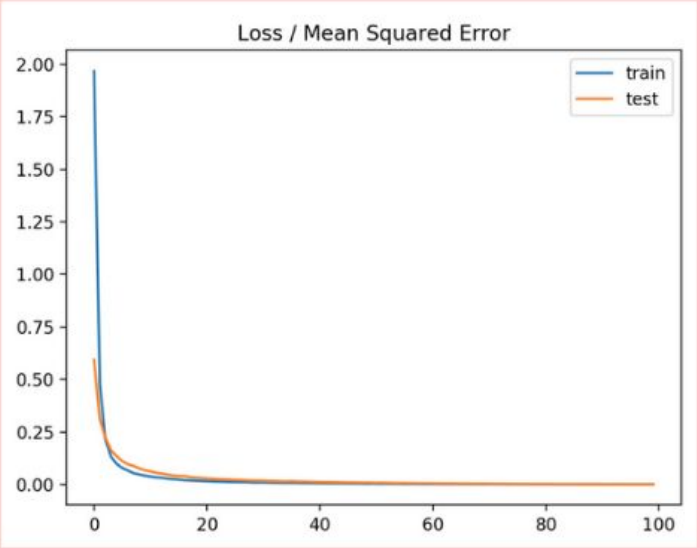




ÁREA SOB A CURVA ROC - AUC

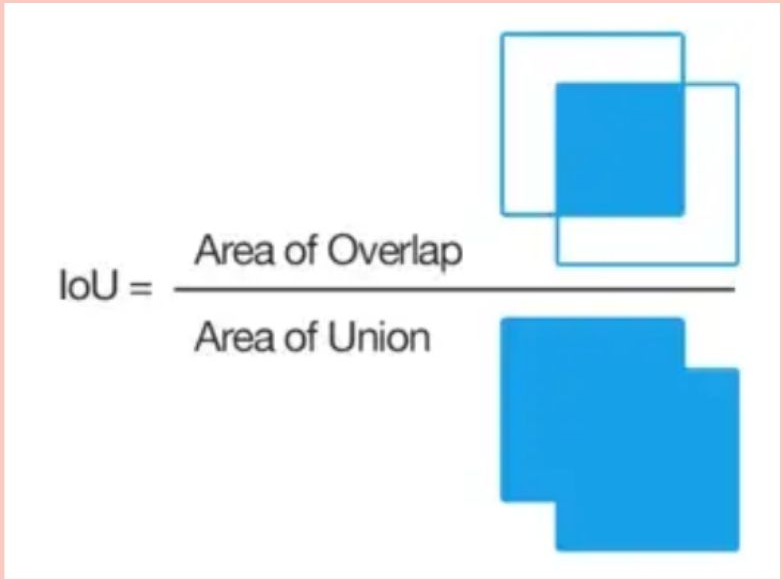
```
#AUC Curve
y_pred_probability = clf.predict_proba(X_test)[::,1]
fpr, tpr, _ = metrics.roc_curve(y_test,  y_pred_probability)
auc = metrics.roc_auc_score(y_test, y_pred_probability)
plt.plot(fpr,tpr,label="data 1, auc="+str(auc))
plt.legend(loc=4)
plt.show()
```

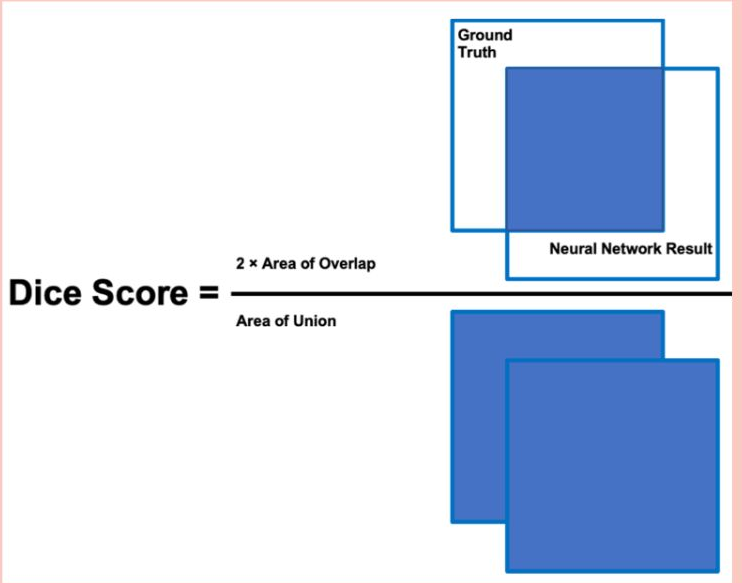
Função de Perda



Uma função de perda é uma função que compara os valores de saída previstos com os esperados

Métricas de avaliação - Imagens





Métricas de avaliação - Imagens

Aspecto	Dice	IoU (Jaccard)
Sensibilidade	Mais sensível a verdadeiros positivos	Mais rigoroso, penaliza mais a falta de sobreposição
Uso comum	Segmentação médica, visão computacional	Detecção de objetos, benchmarks padrão
Penalização	Penaliza menos falsos negativos	Penaliza mais falsos positivos e negativos
Aplicação ideal	Quando se quer maximizar similaridade geral	Quando a precisão da localização é crítica