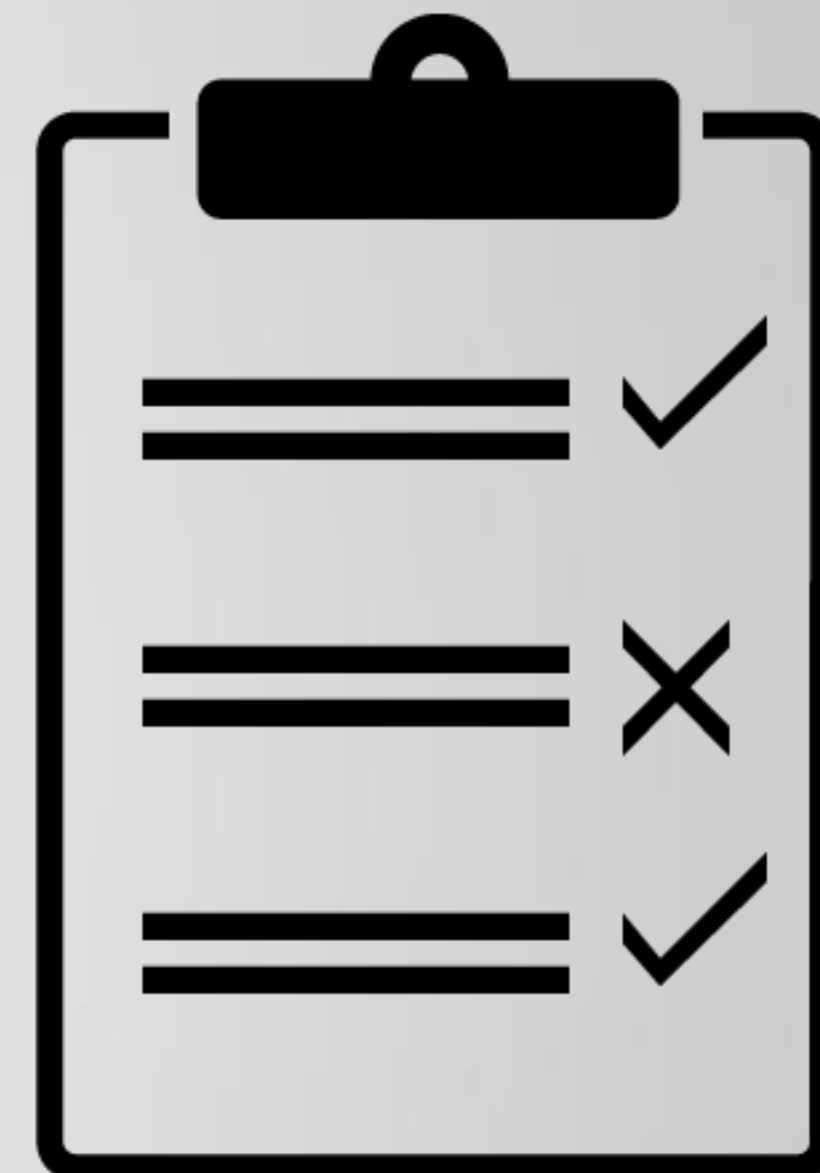


Sdco



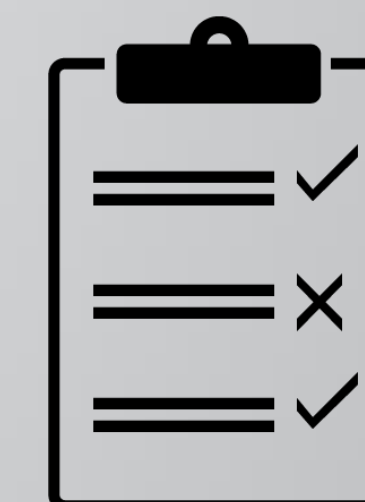
AVALIAÇÃO DE DESEMPENHO PARA CLASSIFICAÇÃO



AVALIAÇÃO DE DESEMPENHO



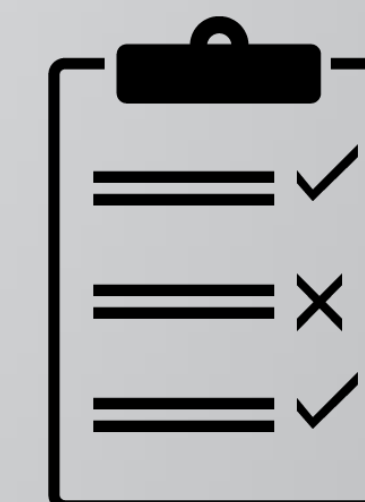
UM ALGORITMO DE CLASSIFICAÇÃO É
BOM SE PRODUZ HIPÓTESES QUE FAZEM
UM BOM TRABALHO DE PREVISÃO DE
CLASSIFICAÇÃO DE **EXEMPLOS NÃO-
VISTOS**



AValiação DE DESEMPENHO



MACHINE LEARNING UMA FERRAMENTA
PODEROSA PARA AQUISIÇÃO AUTOMÁTICA
DE CONHECIMENTO, ENTRETANTO, **NÃO**
EXISTE UM ÚNICO ALGORITMO DE
CLASSIFICAÇÃO QUE APRESENTE MELHOR
DESEMPENHO PARA TODOS PROBLEMAS.



COMO

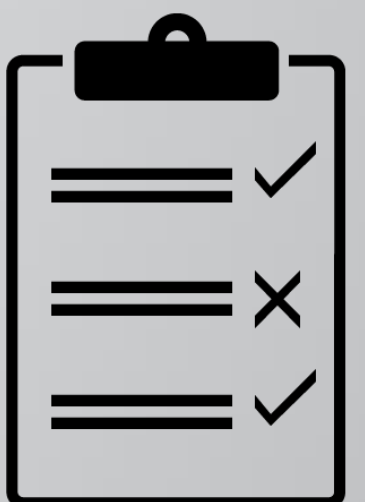
Sdco

TESTAR UM ALGORITMO DE CLASSIFICAÇÃO



PASSOS PARA **TESTE DESEMPENHO** Sd ∞

- 1 – COLETAR UM GRANDE CONJUNTO DE EXEMPLOS;
- 2 – DIVIDI-LO EM DOIS CONJUNTOS (TREINO/TESTE);
- 3 – TREINAR O ALGORITMO COM O CONJUNTO DE TREINO;
- 4 – TESTAR A HIPÓTESE COM O CONJUNTO DE TESTE;
- 5 – MEDIR OS ACERTOS DA HIPÓTESE.



ERRO EM CLASSIFICAÇÃO

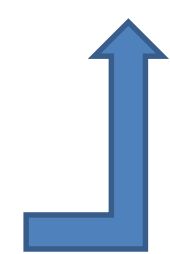


ERRO NA CLASSIFICAÇÃO

EQUAÇÃO

$$err(h) = \frac{1}{n} \sum_{i=1}^n \parallel y_i \neq h(x_i) \parallel$$

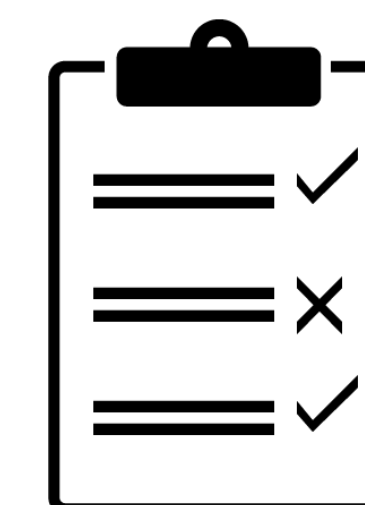
Somatório dos objetos
Classificados erroneamente



EXEMPLO:

Em um conjunto com 100 objetos de teste 95 foram classificados corretamente e 5 não.

Então o erro será de
 $5/100 = 0.05$ ou 5%



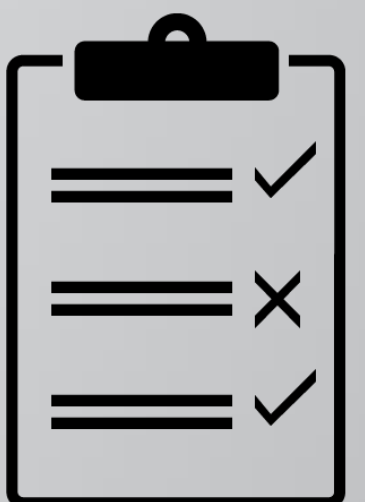
ACURÁCIA



ACURÁCIA



ENTRETANTO EM CLASSIFICAÇÃO,
NÃO ENFATIZAMOS OS ERROS DO
ALGORITMOS MAS SIM A TAXA DE
ACERTOS COMO A ACURÁCIA.

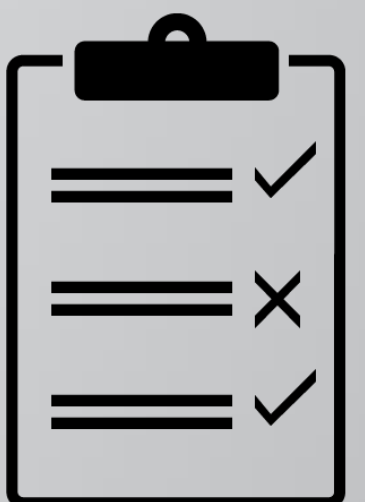


ACURÁCIA



É OBTIDA PELA **MATRIZ DE CONFUSÃO**
QUE VERMOS MAIS A FRENTE

E TAMBÉM VEREMOS ONDE A ACURÁCIA
PODE SER PROBLEMÁTICA.



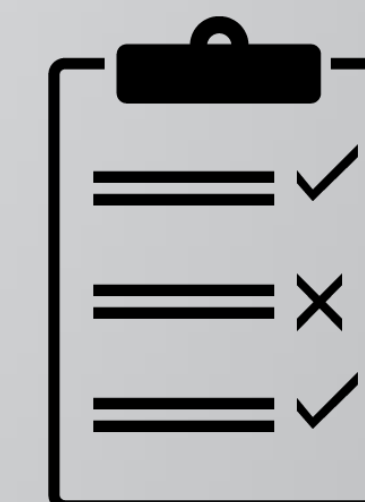
MAS,

DE

FORMA BEM

SIMPLES

Sdco



EQUAÇÃO

$$acc(h) = 1 - err(h)$$

EQUAÇÃO

$$acc(h) = 1 - err(h)$$

EXEMPLO:

Em um conjunto com 100 objetos de teste 95 foram classificados corretamente e 5 não.

Então o erro será de $5/100 = 0.05$

e a acurácia de $1 - 0.05 = 0.95$

ou 95%

SCIKIT LEARN

PARA OBTER A ACURÁCIA DO MODELO USE
O MÉTODO SCORE(x_test, y_test)

```
from sklearn.neighbors import KNeighborsClassifier  
knn = KNeighborsClassifier(n_neighbors=3)  
knn.score(X_test, y_test)
```

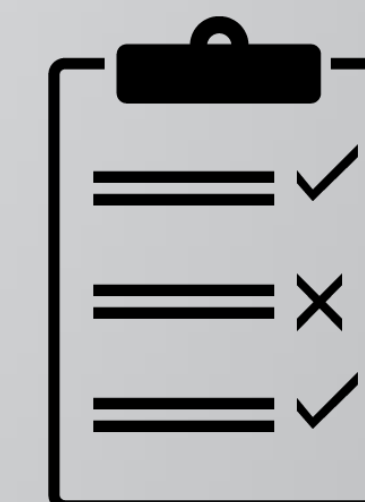
X_test, y_test -> são obtidos na validação cruzada

VALIDAÇÃO CRUZADA

VALIDAÇÃO CRUZADA



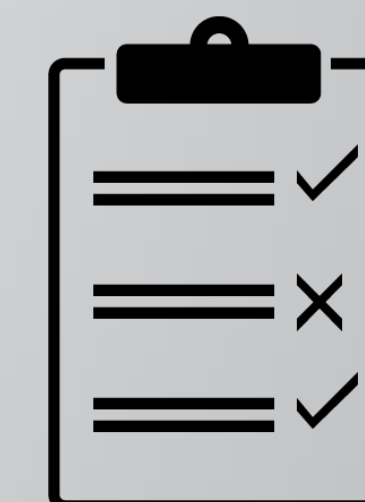
QUANDO DIVIDIMOS O CONJUNTO DE DADOS EM DOIS CONJUNTOS, PODEMOS REDUZIR DRASTICAMENTE O NÚMERO DE AMOSTRAS PARA O TREINAMENTO.



VALIDAÇÃO CRUZADA



O CONCEITO CENTRAL DAS TÉCNICAS DE
VALIDAÇÃO CRUZADA É O
PARTICIONAMENTO DO CONJUNTO DE
DADOS EM SUBCONJUNTOS
MUTUALMENTE EXCLUSIVOS.



VALIDAÇÃO CRUZADA



BASICAMENTE 3 TIPOS:

HOLDOUT;

K-FOLD;

LEAVE-ONE-OUT.



VALIDAÇÃO CRUZADA



HOLDOUT

Dividir o conjunto de dados em dois.

Muito comum é considerar $\frac{2}{3}$ dos dados para treinamento e o $\frac{1}{3}$ restante para teste.



SCIKIT LEARN

IMPORTAR **TRAIN_TEST_SPLIT** PARA DIVIDIR
O CONJUNTO DE DADOS EM TREINO E TESTE

```
from sklearn.model_selection import train_test_split
```

SCIKIT LEARN

EXECUTAR A DIVISÃO DOS CONJUNTOS

```
X_train, X_test, y_train, y_test=train_test_split(X,y,test_size=0.4)
```

X = Conjunto original das Features

Y = Conjunto original das targets

test_size = porcentagem do conjunto de teste

SCIKIT LEARN

EXECUTAR A DIVISÃO DOS CONJUNTOS

```
X_train, X_test, y_train, y_test=train_test_split(X,y,test_size=0.4)
```

X_train = Conjunto de treino das Features

y_train = Conjunto de treino das targets

X_test = Conjunto de teste das Features

y_test = Conjunto de teste das targets

SCIKIT LEARN

Executando KNN

```
from sklearn.neighbors import KNeighborsClassifier
```

```
knn = KNeighborsClassifier(n_neighbors=3)
```

```
knn.fit(X_train, y_train)
```

```
knn.score(X_test, y_test)
```


SCIKIT LEARN

Executando com árvores de decisão

```
from sklearn import tree

tree = tree.DecisionTreeClassifier()

tree.fit(X_train, y_train)

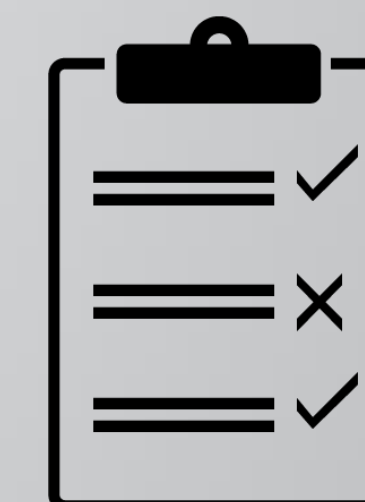
tree.score(X_test, y_test)
```

UMA

MELHOR DIVISÃO

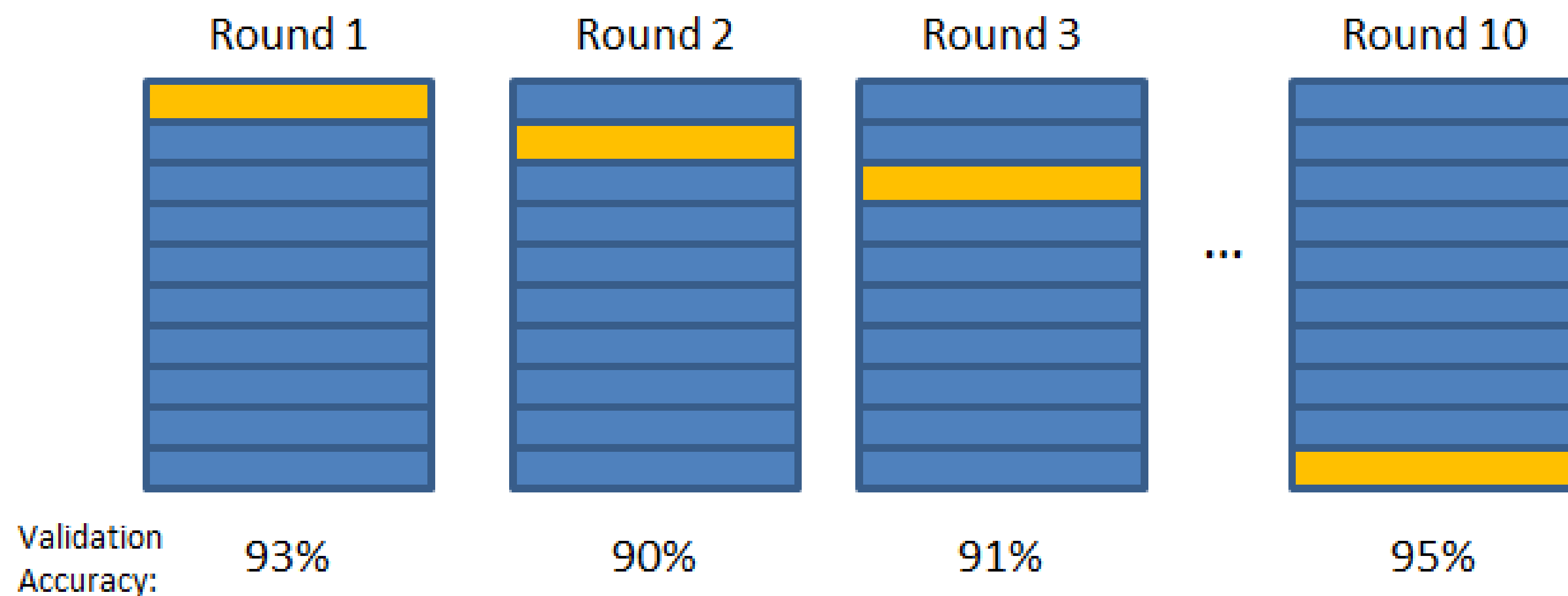


DIVIDIR O CONJUNTO TOTAL DE DADOS
EM **K** SUBCONJUNTOS MUTUAMENTE
EXCLUSIVOS DO MESMO TAMANHO.



K-FOLD

Validation Set
Training Set



Final Accuracy = Average(Round 1, Round 2, ...)

Partir disto, um subconjunto é utilizado para teste e os $k-1$ para treino. Calcula-se a acurácia do modelo. Este processo é realizado k vezes alternando de forma circular o subconjunto de teste.

```
from sklearn.neighbors import KNeighborsClassifier
from sklearn import model_selection

kfold = model_selection.KFold(n_splits=10, random_state=42)
model = KNeighborsClassifier(n_neighbors=3)

results = model_selection.cross_val_score(model, X, y, cv=10, scoring= 'accuracy')

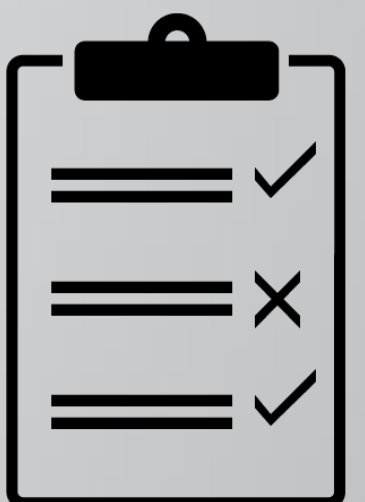
print(results) ## acurácia de cada fold

print(results.mean()) ## acurácia total
```

LEAVE-ONE-OUT



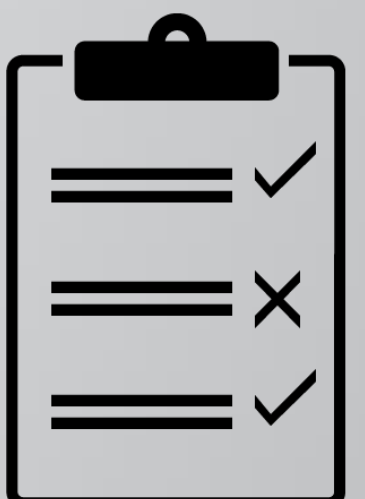
É UM CASO ESPECÍFICO DO **K-FOLD**,
COM **K** IGUAL AO NÚMERO TOTAL DE
DADOS **N**. NESTA ABORDAGEM SÃO
REALIZADOS **N** CÁLCULOS DE ERRO, UM
PARA CADA DADO.



LEAVE-ONE-OUT



APESAR DA INVESTIGAÇÃO COMPLETA
DO MODELO EM RELAÇÃO AOS DADOS
UTILIZADOS, POSSUI ALTO CUSTO
COMPUTACIONAL.



```
from sklearn.neighbors import KNeighborsClassifier  
from sklearn import model_selection
```

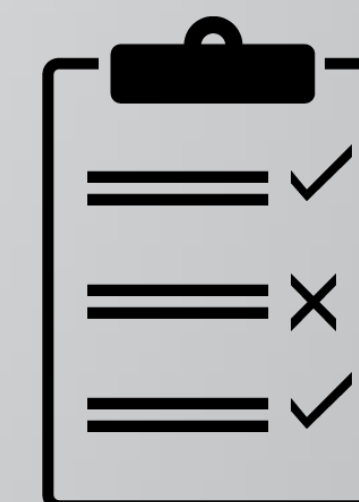
```
N = X.shape[0] <-Leave-one-out
```

```
kfold = model_selection.KFold(n_splits=N, random_state=42)
```

```
model = KNeighborsClassifier(n_neighbors=3)
```

```
results = model_selection.cross_val_score(model, X, y, cv=kfold, scoring= 'accuracy')
```


MATRIZ DE CONFUSÃO

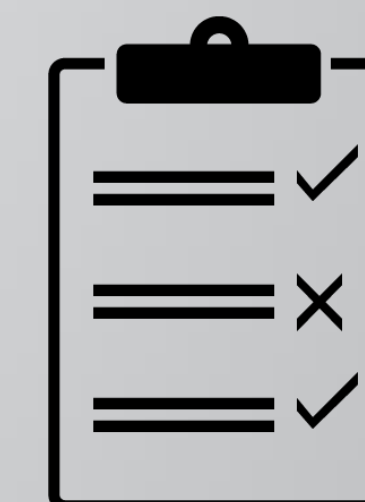


MATRIZ DE CONFUSÃO



CUIDADO!!!

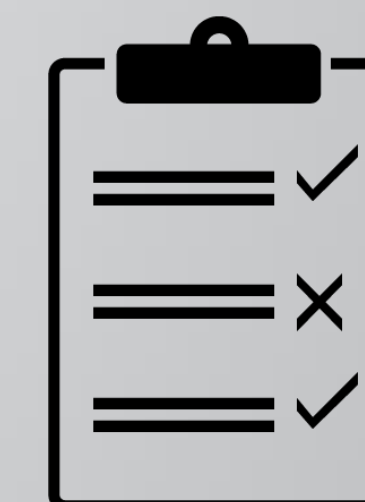
A ACURÁCIA PODE SER ENGANOSA.



MATRIZ DE CONFUSÃO

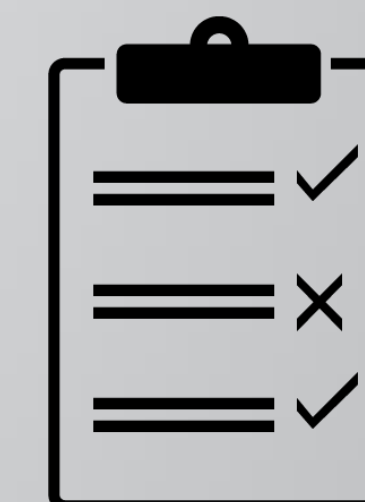
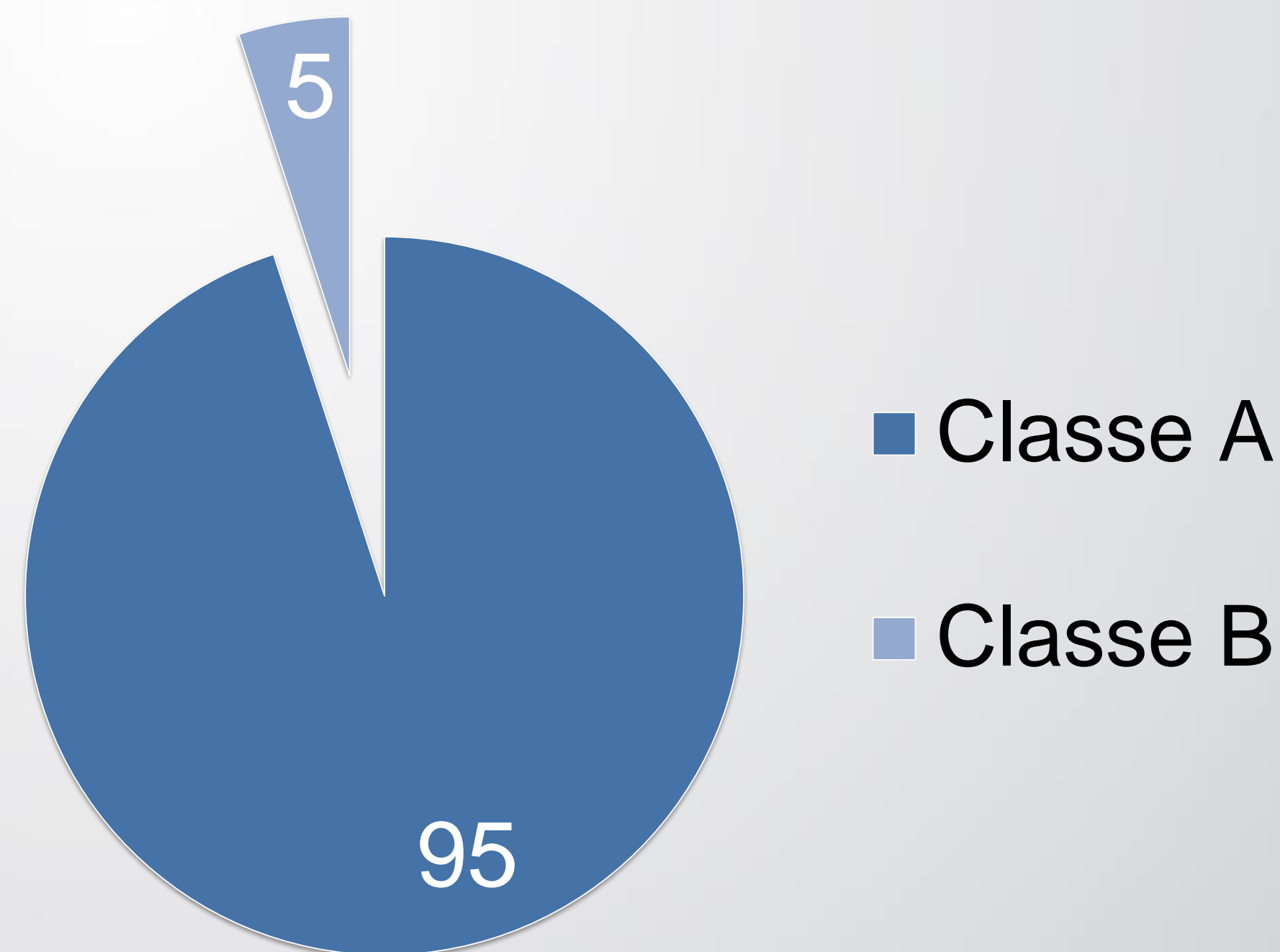


A ACURÁCIA PODE PRODUZIR
RESULTADOS ENGANADORES SE O
CONJUNTO DE DADOS ESTIVER
DESEQUILIBRADO.



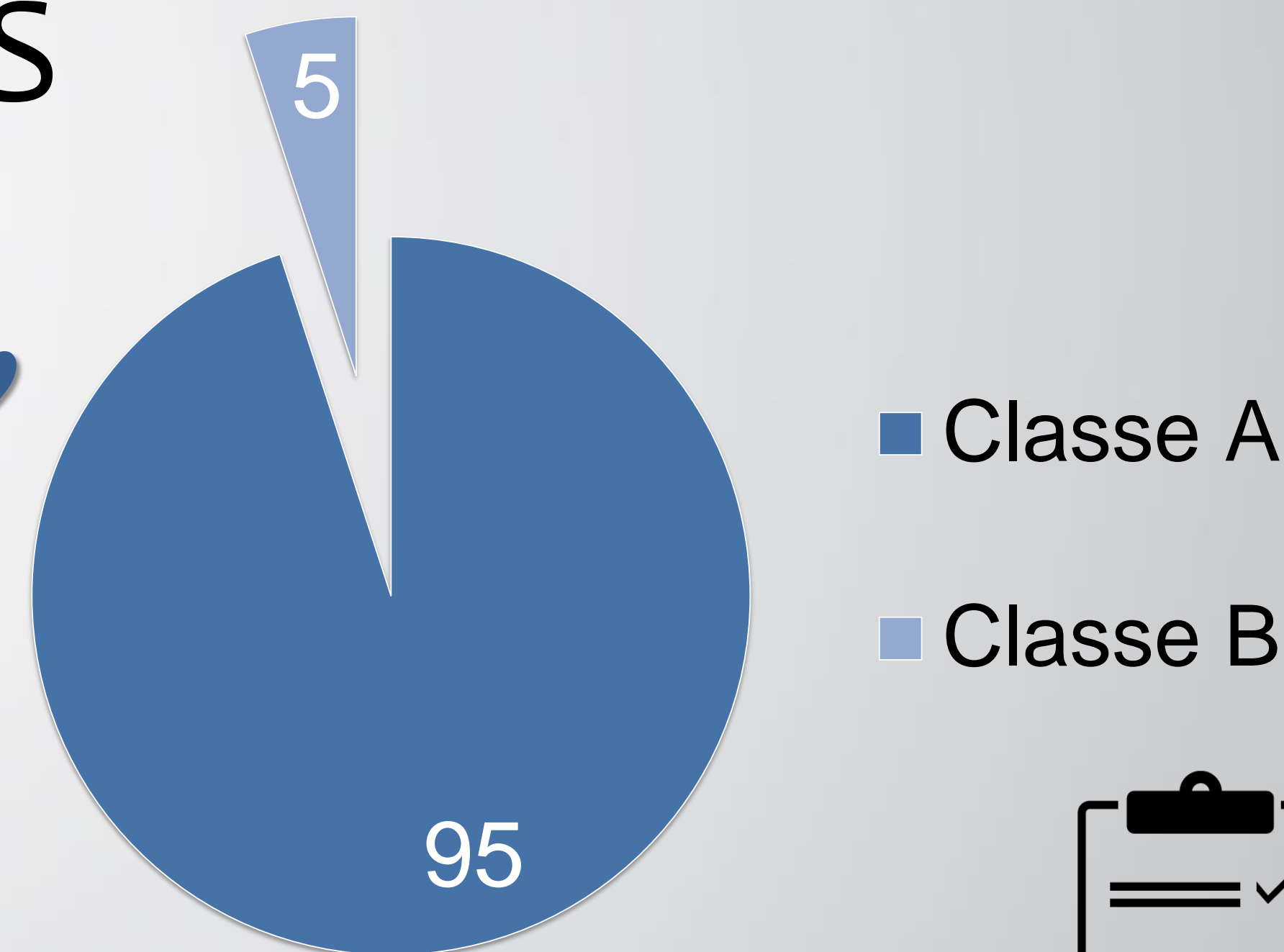
MATRIZ DE CONFUSÃO

POR EXEMPLO, SE HOUVESSE 95
EXEMPLOS DE UMA **CLASSE A** E APENAS
5 DE OUTRA **CLASSE B**



MATRIZ DE CONFUSÃO

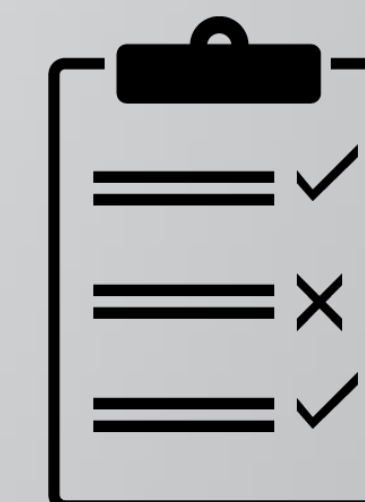
UM CLASSIFICADOR PARTICULAR
PODERIA CLASSIFICAR TODAS AS
OBSERVAÇÕES COMO **CLASSE A**,
DADO O DESEQUILÍBRIO



MATRIZ DE CONFUSÃO



DADO UM EXPERIMENTO
DE INSTÂNCIAS **P** POSITIVAS E **N**
INSTÂNCIAS NEGATIVAS A MATRIZ DE
CONFUSÃO SERÁ:



MATRIZ DE CONFUSÃO

		VALOR PREDITO NA CLASSIFICAÇÃO	
		POSITIVO	NEGATIVO
		VP	FN
		FP	VN

Digamos que temos duas classes A e B

Se chamarmos a classe A de positivo, então a classe B será negativo

Por exemplo:

Imagens com pneumonia e sem pneumonia

POSITIVO - Pneumonia

NEGATIVO - Sem pneumonia

É Cachorro ou Gato?

POSITIVO - Cachorro

NEGATIVO - Gato

É email span?

POSITIVO - É span

NEGATIVO - Não Span

MATRIZ DE CONFUSÃO

		VALOR PREDITO NA CLASSIFICAÇÃO	
		POSITIVO	NEGATIVO
	POSITIVO	VP	FN
	NEGATIVO	FP	VN

VERDADEIRO POSITIVO

Quantidade de valores que o Classificador acertou para a Classe A (positivo).

O classificador disse que era A...

...e realmente era A.

MATRIZ DE CONFUSÃO

		VALOR PREDITO NA CLASSIFICAÇÃO	
		POSITIVO	NEGATIVO
		VP	FN
		FP	VN

FALSO POSITIVO

Quantidade de valores que o Classificador errou para a Classe A.

O classificador disse que era A...

...mas era B.

MATRIZ DE CONFUSÃO

		VALOR PREDITO NA CLASSIFICAÇÃO	
		POSITIVO	NEGATIVO
		VP	FN
		FP	VN

VERDADEIRO NEGATIVO

Quantidade de valores que o Classificador acertou para a Classe B (positivo).

O classificador disse que era B...

...e realmente era B.

MATRIZ DE CONFUSÃO

		VALOR PREDITO NA CLASSIFICAÇÃO	
		POSITIVO	NEGATIVO
		VP	FN
		FP	VN

FALSO NEGATIVO

Quantidade de valores que o Classificador errou para a Classe B.

O classificador disse que era B...

...mas era A.

MATRIZ DE CONFUSÃO

		VALOR PREDITO NA CLASSIFICAÇÃO	
		CACHORRO	GATO
	CACHORRO	50	5
	GATO	2	10
		52 CACHORROS	15 GATOS

EXEMPLO:

NOSSA BASE TEM

52 CACHORROS

15 GATOS

ACURÁCIA

$$(50+10)/67 = 89\%$$

Classifica melhor cachorro do que gato.

MATRIZ DE CONFUSÃO + K-FOLD



SCIKIT LEARN

```
from sklearn.model_selection import cross_val_predict
```

```
from sklearn.metrics import confusion_matrix
```

```
kfold = model_selection.KFold(n_splits=10, random_state=42)
```

```
y_pred = cross_val_predict(model,X,y,cv=kfold)
```

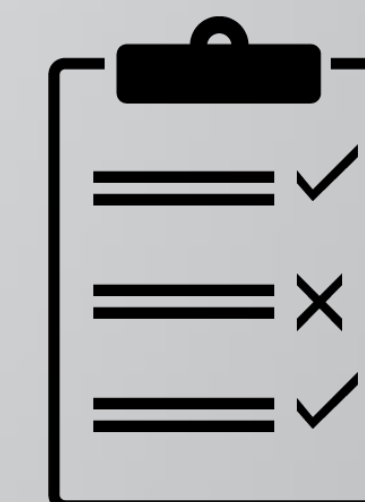
```
matriz = confusion_matrix(y,y_pred)
```

MATRIZ DE CONFUSÃO



MÉTRICAS GERADAS

A PARTIR DA MATRIZ DE CONFUSÃO



MATRIZ DE CONFUSÃO



ACURÁCIA

A PROPORÇÃO DE PREDIÇÕES CORRETAS, SEM LEVAR EM CONSIDERAÇÃO O QUE É POSITIVO E O QUE É NEGATIVO. ESTA MEDIDA É ALTAMENTE SUSCETÍVEL A DESBALANCEAMENTOS DO CONJUNTO DE DADOS E PODE FACILMENTE INDUZIR A UMA CONCLUSÃO ERRADA SOBRE O DESEMPENHO DO SISTEMA.

		VALOR PREDITO NA CLASSIFICAÇÃO	
		CACHORRO	GATO
		VP	FN
		FP	VN

ACURACIA

TOTAL DE ACERTOS / N DE EXEMPLOS
 $(VP + VN) / (P + N)$

MATRIZ DE CONFUSÃO



SENSIBILIDADE ou REVOCAÇÃO

A PROPORÇÃO DE VERDADEIROS POSITIVOS:

A CAPACIDADE DO SISTEMA EM PREDIZER

CORRETAMENTE A CONDIÇÃO PARA CASOS QUE

REALMENTE A TÊM.

SENSIBILIDADE ou REVOCAÇÃO

ACERTOS POSITIVOS / TOTAL DE POSITIVOS

$$VP / (VP + FN)$$

		VALOR PREDITO NA CLASSIFICAÇÃO	
		CACHORRO	GATO
	CACHORRO	VP	FN
	GATO	FP	VN

MATRIZ DE CONFUSÃO



ESPECIFICIDADE

A PROPORÇÃO DE VERDADEIROS NEGATIVOS.

A CAPACIDADE DO SISTEMA EM PREDIZER
CORRETAMENTE A AUSÊNCIA DA CONDIÇÃO
PARA CASOS QUE REALMENTE NÃO A TÊM.

ESPECIFICIDADE

ACERTOS NEGATIVOS / TOTAL DE NEGATIVOS

$$VN / (VN + FP)$$

		VALOR PREDITO NA CLASSIFICAÇÃO	
		CACHORRO	GATO
	CACHORRO	VP	FN
	GATO	FP	VN

MATRIZ DE CONFUSÃO



EFICIÊNCIA

A MÉDIA ARITMÉTICA DA SENSIBILIDADE E ESPECIFICIDADE. NA PRÁTICA, A SENSIBILIDADE E A ESPECIFICIDADE VARIAM EM DIREÇÕES OPOSTAS. ISTO É, GERALMENTE, QUANDO UM MÉTODO É MUITO SENSÍVEL A POSITIVOS, TENDE A GERAR MUITOS FALSO-POSITIVOS, E VICE-VERSA. ASSIM, UM MÉTODO DE DECISÃO PERFEITO (100 % DE SENSIBILIDADE E 100% ESPECIFICIDADE) RARAMENTE É ALCANÇADO.

EFICIÊNCIA

$$(SENS + ESPEC) / 2$$

VALOR PREDITO NA CLASSIFICAÇÃO			
		CACHORRO	GATO
		VP	FN
		FP	VN

MATRIZ DE CONFUSÃO



VALOR PREDITIVO POSITIVO ou PRECISÃO

A PROPORÇÃO DE VERDADEIROS POSITIVOS EM
RELAÇÃO A TODAS AS PREDIÇÕES POSITIVAS.

ESTA MEDIDA É ALTAMENTE SUSCETÍVEL A
DESBALANCEAMENTOS DO CONJUNTO DE DADOS E
PODE FACILMENTE INDUZIR A UMA CONCLUSÃO
ERRADA SOBRE O DESEMPENHO DO SISTEMA.

		VALOR PREDITO NA CLASSIFICAÇÃO	
		CACHORRO	GATO
		VP	FN
		FP	VN

VALOR PREDITIVO POSITIVO ou PRECISÃO

ACERTOS POSITIVOS / TOTAL DE PREDIÇÕES POSITIVAS
 $VP / (VP + FP)$

MATRIZ DE CONFUSÃO



VALOR PREDITIVO NEGATIVO

A PROPORÇÃO DE VERDADEIROS NEGATIVAS EM
RELAÇÃO A TODAS AS PREDIÇÕES NEGATIVAS.

ESTA MEDIDA É ALTAMENTE SUSCETÍVEL A
DESBALANCEAMENTOS DO CONJUNTO DE DADOS E
PODE FACILMENTE INDUZIR A UMA CONCLUSÃO
ERRADA SOBRE O DESEMPENHO DO SISTEMA.

		VALOR PREDITO NA CLASSIFICAÇÃO	
		CACHORRO	GATO
		VP	FN
		FP	VN

VALOR PREDITIVO NEGATIVO

ACERTOS NEGATIVOS / TOTAL DE PREDIÇÕES NEGATIVAS

$$VN / (VN + FN)$$

MATRIZ DE CONFUSÃO



COEFICIENTE DE CORRELAÇÃO DE MATTHEWS – COEFICIENTE (PHI)

O coeficiente de correlação de Matthews é uma medida de qualidade de duas classificações binárias que pode ser usada mesmo se as classes possuem tamanhos bastante diferentes. Retorna um valor entre (-1) e (+1), em que um coeficiente de (+1) representa uma predição perfeita, (0) representa uma predição aleatória media, e (-1) uma predição inversa. Esta estatística é equivalente ao coeficiente phi, e tenta, assim como a eficiência, resumir a qualidade da tabela de contingência em um único valor numérico passível de ser comparado.

VALOR PREDITO NA CLASSIFICAÇÃO			
		CACHORRO	GATO
		VP	FN
		FP	VN

VALOR PREDITIVO POSITIVO

$$(VP * VN - FP * FN) / \sqrt{((VP + FP) * (VP + FN) * (VN + FP) * (VN + FN))}$$

MATRIZ DE CONFUSÃO



F-MEASURE

A MEDIDA QUE COMBINA PRECISÃO E REVOCAÇÃO É A MÉDIA HARMÓNICA DE PRECISÃO E REVOCAÇÃO, A TRADICIONAL F-MEASURE OU F-SCORE BALANCEADA

		VALOR PREDITO NA CLASSIFICAÇÃO	
		CACHORRO	GATO
	CACHORRO	VP	FN
	GATO	FP	VN

F-MEASURE

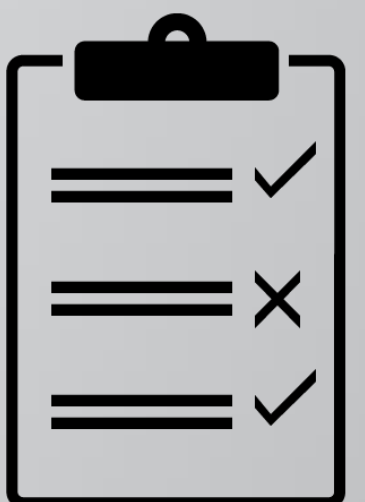
$$2 * ((Precisao * revocação) / (Precisao + revocação))$$

UMA

METRICA VISUAL



A CURVA ROC É UMA FERRAMENTA
PODEROSA PARA MEDIR E ESPECIFICAR
PROBLEMAS DE DESEMPENHO DOS
CLASSIFICADORES.

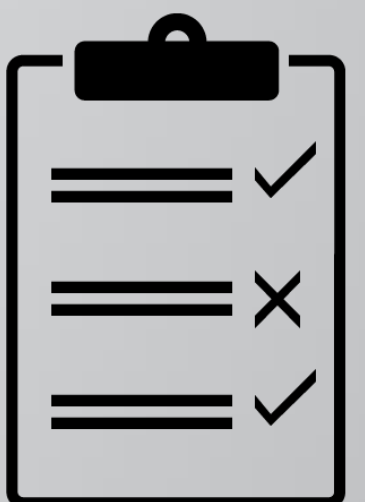


CURVA ROC

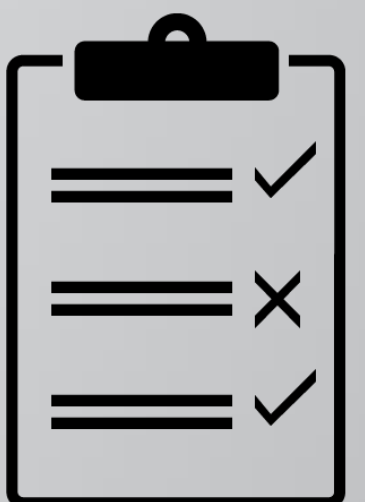


É BASEADA EM MÉTRICAS DA MATRIZ
DE CONFUSÃO.

SENSIBILIDADE e ESPECIFICIDADE.



QUANTO MAIOR A SENSIBILIDADE DO
ALGORITMO DE CLASSIFICAÇÃO AO
PROBLEMA, MELHOR ELE SERÁ.



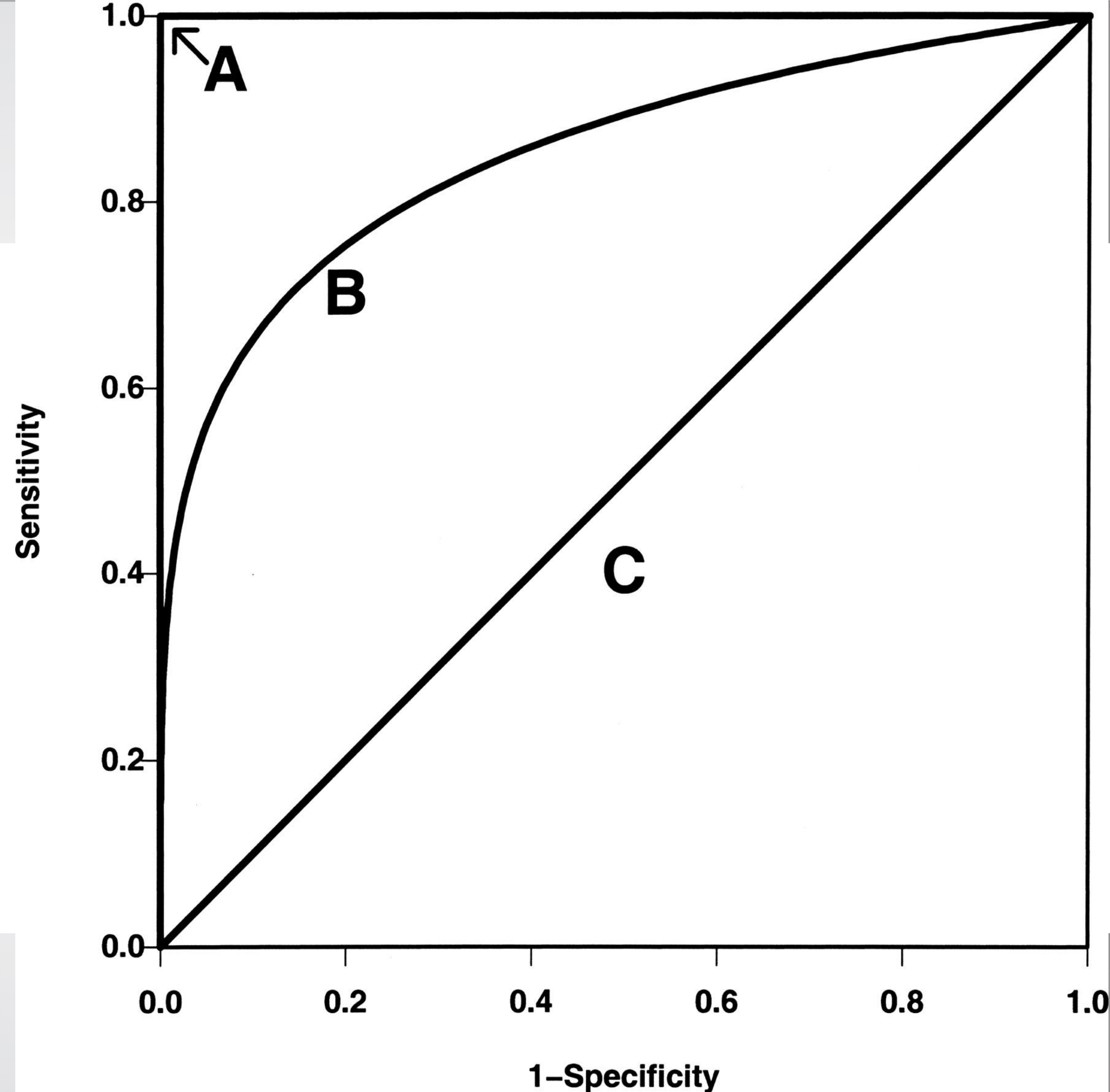
CURVA ROC

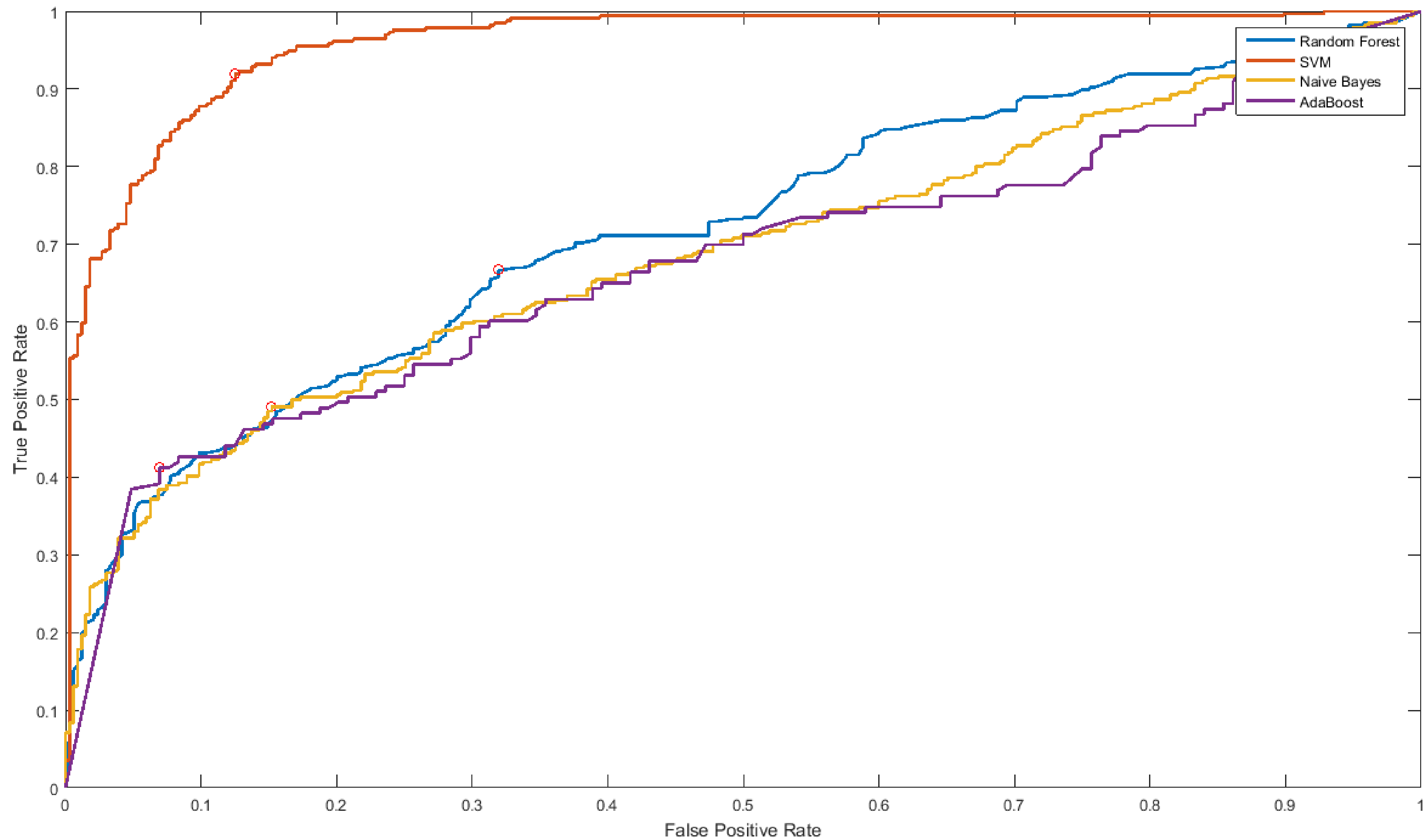
EXEMPLO

A – PONTO IDEAL

B – BOA CLASSIFICAÇÃO

C – RUIM CLASSIFICAÇÃO

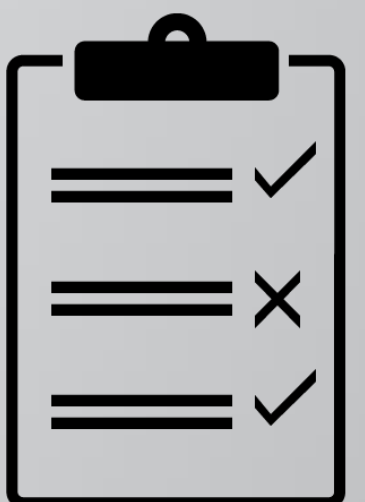




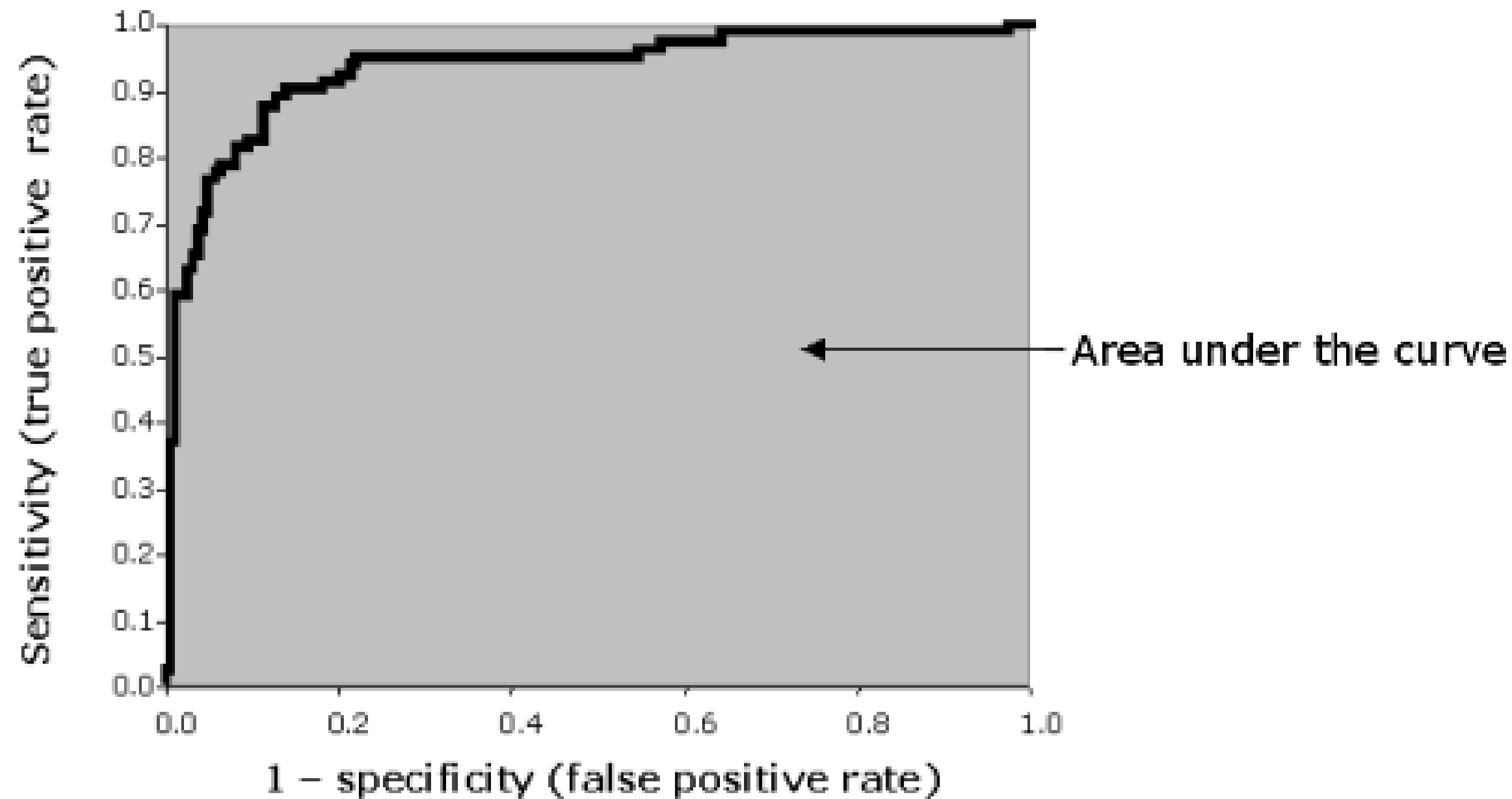
ÁREA SOB CURVA ROC



ÁREA SOB CURVA ROC (OU AUC ABREVIADA) É UMA MÉTRICA DE DESEMPENHO PARA PROBLEMAS DE CLASSIFICAÇÃO BINÁRIA.



ÁREA SOB A CURVA ROC



ÁREA SOB A CURVA ROC



```
from sklearn.neighbors import KNeighborsClassifier
from sklearn import model_selection

kfold = model_selection.KFold(n_splits=10, random_state=42)
model = KNeighborsClassifier(n_neighbors=3)

results = model_selection.cross_val_score(model, X, y, cv=kfold, scoring= 'roc_auc')
```

MATRIZ DE CONFUSÃO



K-FOLD

MATRIZ CONFUSÃO + K-FOLD



SCIKIT LEARN

```
from sklearn.model_selection import cross_val_predict  
  
y_pred = model_selection.cross_val_predict(model, X, y, cv=kfold)  
  
matriz = confusion_matrix(y, y_pred)
```