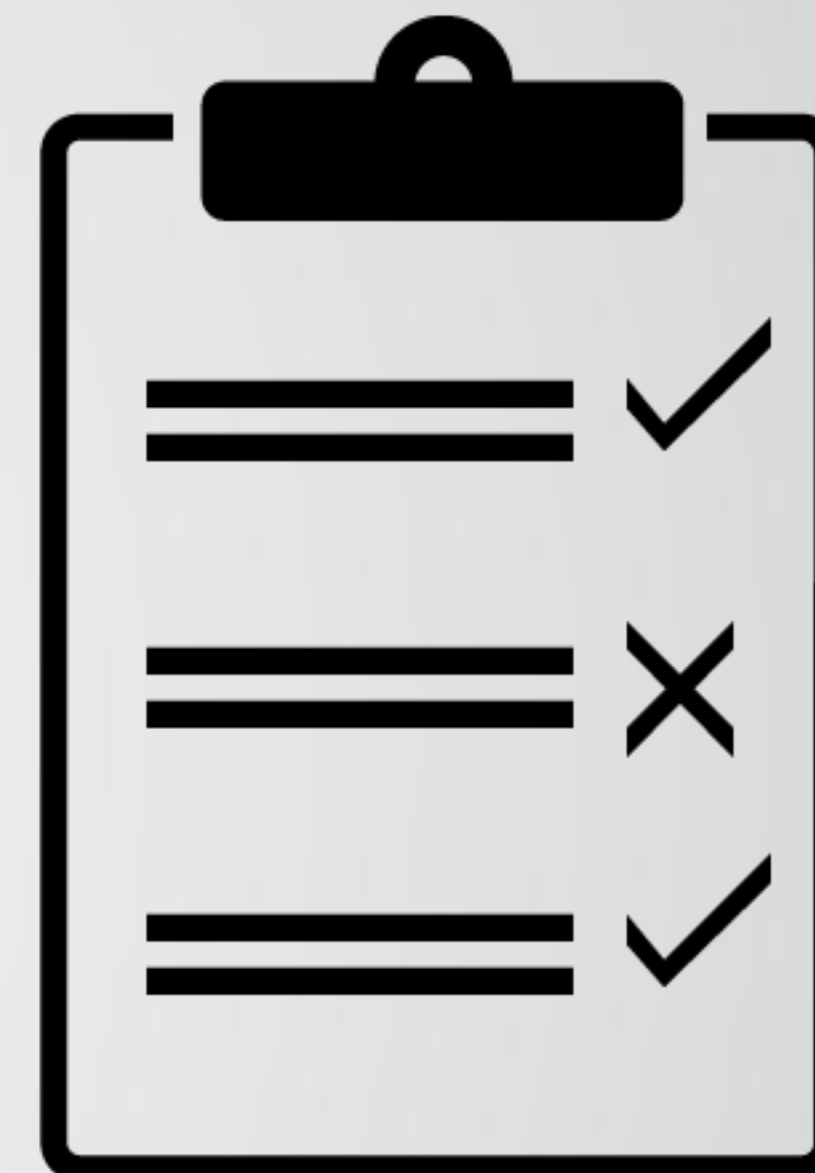


Sdco



/ CANALSANDECO

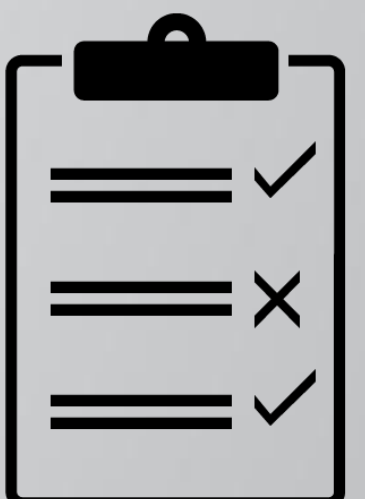
MEDIDAS DE DESEMPENHO



MEDIDAS DE DESEMPENHO



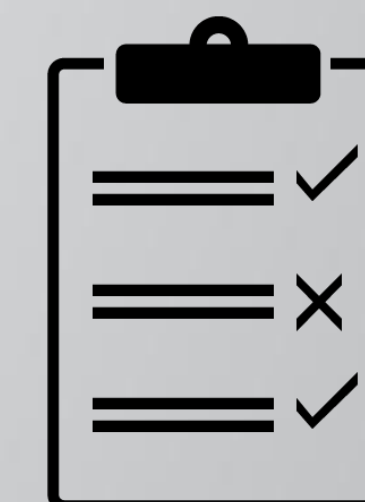
UM ALGORITMO É BOM SE PRODUZ
HIPÓTESES QUE FAZEM UM BOM
TRABALHO DE PREVISÃO DE
CLASSIFICAÇÃO DE **EXEMPLOS NÃO-
VISTOS**



MEDIDAS DE DESEMPENHO



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

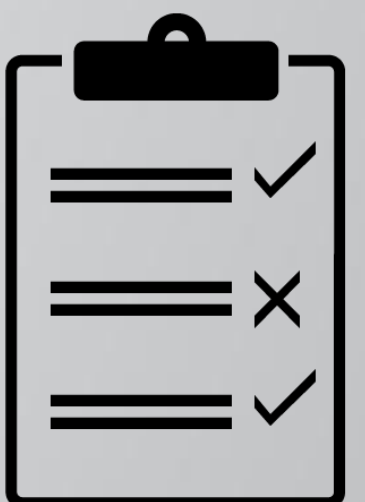


COMO TESTAR UM ALGORITMO

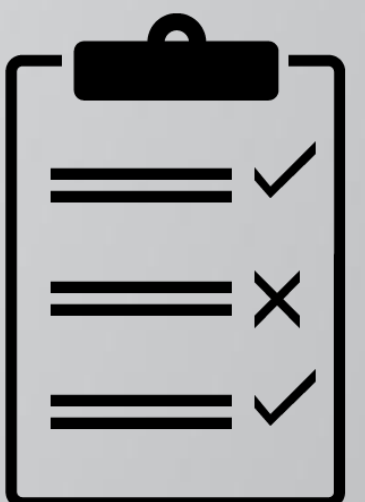


PASSOS PARA **TESTE DESEMPENHO** Sd ∞

- 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



EQUAÇÃO

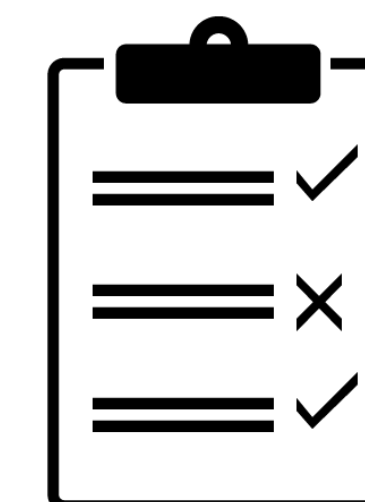
$$err(h) = \frac{1}{n} \sum_{i=1}^n \mathbb{1}_{y_i \neq h(x_i)}$$

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%



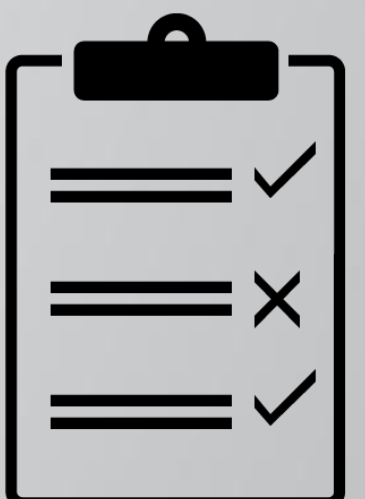
ACURÁCIA



ACURÁCIA



ENTRETANTO EM MACHINE LEARNING
NÃO ENFATIZAMOS OS ERROS DO
ALGORITMOS E SIM A TAXA DE ACERTOS
COMO A ACURÁCIA.

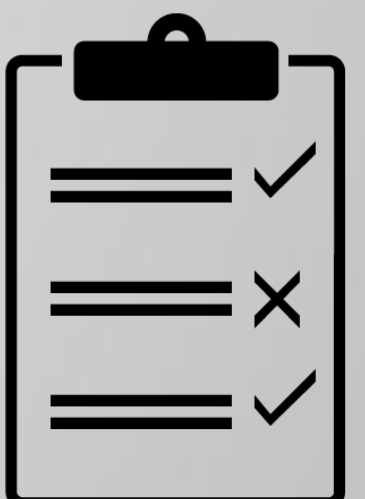


ACURÁCIA



DEFINIÇÃO:

PROXIMIDADE ENTRE O VALOR OBTIDO
EXPERIMENTALMENTE E O VALOR VERDADEIRO
NA MEDIÇÃO DE UMA GRANDEZA FÍSICA.



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

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_train = Conjunto de treino das Features

X_test = Conjunto de teste das Features

y_train = Conjunto de treino das targets

y_test = Conjunto de teste das targets

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

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)
```