



Analisar o Modelo

Após a estratificação, treinamento e teste
O resultado muda a cada execução.
Otimizar o modelo (SEED – Semente) lab_ml_03

`X = df[['inicial', 'palestras', 'patrocinio', 'contato']]` - dados

`y = df['comprou']` - rótulos

Inicial

Palestras

Patrocínio

Contato

Comprar Ingresso

Bem-vindo ao Evento

head()

1	inicial	palestras	contato	comprou	patrocinio
2	1	1	0	0	0
3	1	1	0	0	0
4	1	1	0	0	0
5	1	1	0	0	0
6	1	1	0	0	0
7	1	0	1	1	0
8	1	1	0	0	0

Padrões : a cara que tá
Clicando em duas tá comprando

Comprou?

9	1	0	1	1	0
10	1	1	0	0	0
11	1	0	1	1	0
12	1	1	0	1	0
13	1	0	1	1	0
14	0	0	1	0	0

Machine Learning (Aprendizado de Máquina) Supervisionado

X – dados

Y – rótulos (labels) – classes (1,0)

Classe Comprou – 1

Classe Não Comprou - 0

X

Dados de Treino

$X = df[['inicial', 'palestras', 'patrocinio', 'contato']]$



Y

Rótulos de Treino

$y = df['comprou']$ - rótulos





df.info()

99 linhas e 5 colunas



df.info()

99 linhas e 5 colunas

Treinar



df.info()

99 linhas e 5 colunas

Avaliar



Treino 75%

Teste 25%

treino, classe_treino, teste, classe_teste
= train_test_split(x, y)



`X = df[['inicial', 'palestras', 'patrocinio', 'contato']]`

`y = df['comprou'] - rótulos`

Treino 75%

Teste 25%

treino_x, treino_y, teste_x, teste_y

= train_test_split(x, y)



X = df[['inicial', 'palestras', 'patrocinio', 'contato']]

y = df['comprou'] - rótulos

Rótulo = Classe = y

Classificar:

Comprou = 1

Não comprou = 0

Próximos Passos

- Escolher o Algoritmo:
 - **LinearSVC?**
 - RandomForest ?
 - Regressão Logística?
- **Treinar (Fit)**
- Previsão
- **Acurácia**
 - **Gabarito: 91 a 95.23 %**
- Reavaliar

```
import pandas as pd
dados= pd.read_csv('dataset/tracking.csv')
dados.head(5)
```



```
comprou=dados.comprou  
dados.drop(columns='comprou',inplace=True)  
comprou.head()
```

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

```
treino, teste, classe_treino, classe_teste = train_test_split(dados,  
comprou, random_state=42, test_size=0.3)
```

```
from sklearn.svm import LinearSVC
```

```
modelo= LinearSVC()
```

```
modelo.fit(treino,classe_treino)
```

```
from sklearn.metrics import accuracy_score
```

```
taxa_acerto= accuracy_score(classe_teste,previsoes)*100  
print(f'Acuracia: {taxa_acerto}')
```