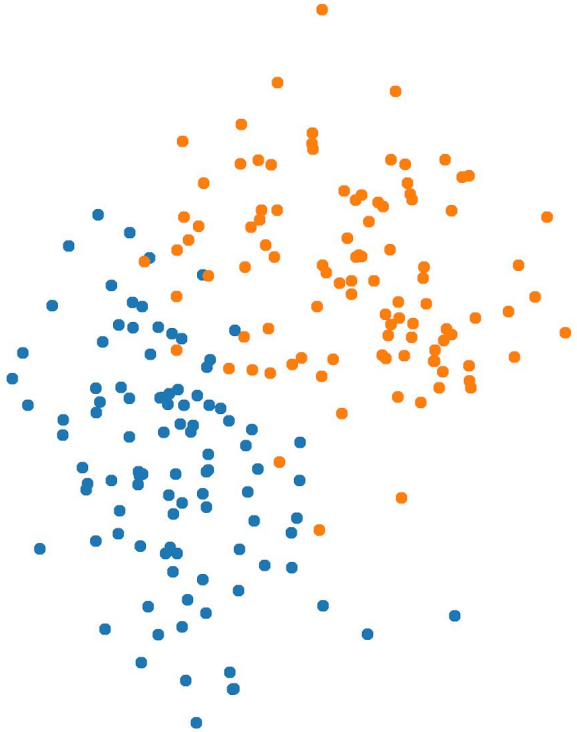
Classificação: KNN

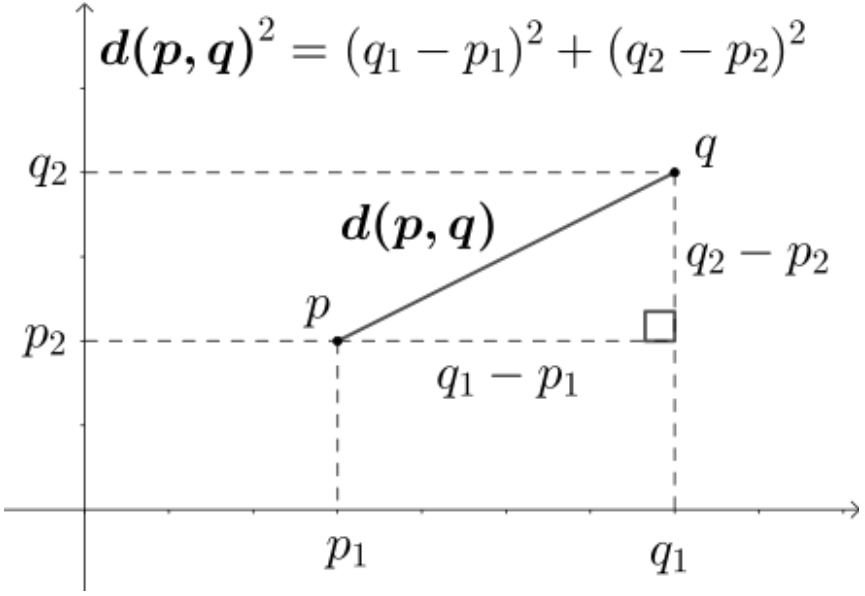
ML: Classificação - KNN (vizinhos próximos)



a) A labeled dataset has three distict groups

Class 1
Class 2

ML: Classificação - KNN (vizinhos próximos)



- E como calculamos distância?

KNN: Train Test Split

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(df
X_train.head(3)
```

	age	is_male	engine_size	months_last_claim
18	44	0	200	6
14	58	0	70	12
36	44	1	70	6

KNN: is sensitive to features scale

```
from sklearn.preprocessing import MinMaxScaler
scaler = MinMaxScaler()
data_scaled = scaler.fit_transform(df)
df_scaled = pd.DataFrame(data_scaled, columns = df.columns)
df_scaled
```

	age	is_male	engine_size	months_last_claim	fraud
0	0.743243	0.0	0.384615	1.000000	1.0
1	0.324324	0.0	0.000000	0.333333	1.0
2	0.837838	0.0	1.000000	0.333333	0.0

KNN: fit, score, predict

```
from sklearn.neighbors import KNeighborsClassifier
knn = KNeighborsClassifier(n_neighbors=3)
knn.fit(X_train,y_train)
train_accuracy= knn.score(X_train,y_train)*100
print(train_accuracy)
knn.predict(X_test)
```

```
82.5
array([0, 0, 0, 0, 0, 0, 1, 1, 1, 0, 0, 0, 1, 0, 0, 0, 1, 1, 1, 0])
```

KNN: “hyperparameter tuning”

```
for k in range(3,20,1):
  knn = KNeighborsClassifier(n_neighbors=k)
  knn.fit(X_train,y_train)
  train_accuracy= knn.score(X_train,y_train)*100
  print(k,train_accuracy)
```

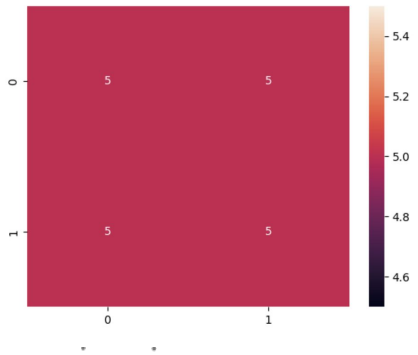
```
3 82.5
4 72.5
5 70.0
6 72.5
7 65.0
8 62.5
9 65.0
10 66.25
11 65.0
12 62.5
13 66.25
14 62.5
15 65.0
16 62.5
17 65.0
18 60.0
19 60.0
```

KNN: “hyperparameter tuning”

```
preds = knn.predict(X_test)
```

```
from sklearn.metrics import confusion_matrix
print(accuracy_score(y_test,preds)*100)
print(confusion_matrix(y_test,preds))
print(classification_report(y_test,preds))
```

	precision	recall	f1-score	support
0	0.50	0.50	0.50	10
1	0.50	0.50	0.50	10
accuracy			0.50	20
macro avg	0.50	0.50	0.50	20
weighted avg	0.50	0.50	0.50	20



Exercício: kNN

- Utilizando o algoritmo do kNN, treine seu modelo para reconhecer os clientes com bom perfil de crédito e os que não.
    - Aplique seu modelo treinado ao um conjunto novo de potenciais clientes.
    - Avalie o nível de qualidade do seu modelo, calculando as seguintes métricas
        - Accuracy
        - Precision
        - Recall

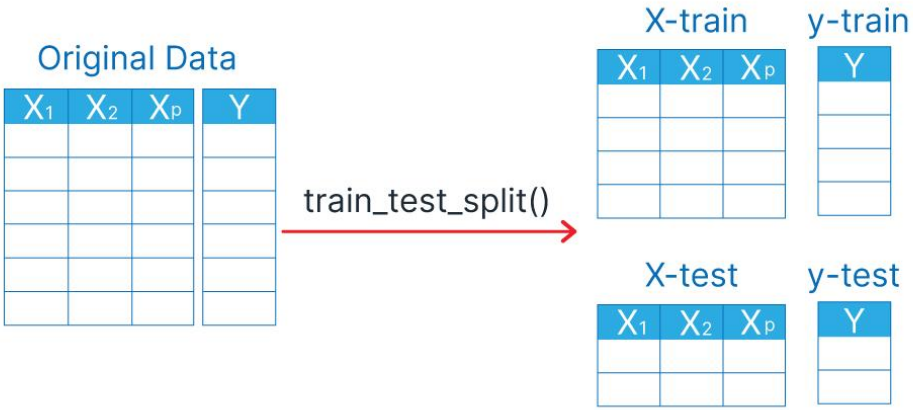
treino: https://raw.githubusercontent.com/lcbjrrr/data/main/RiscoCredito%20-%20okk.csv
teste: https://raw.githubusercontent.com/lcbjrrr/data/main/RiscoCredito%20-%20prever2.csv


MAD MEN
Intuición · Creatividad · Appeal
MATH MEN
Analytics · Resultados · Lógica
Google

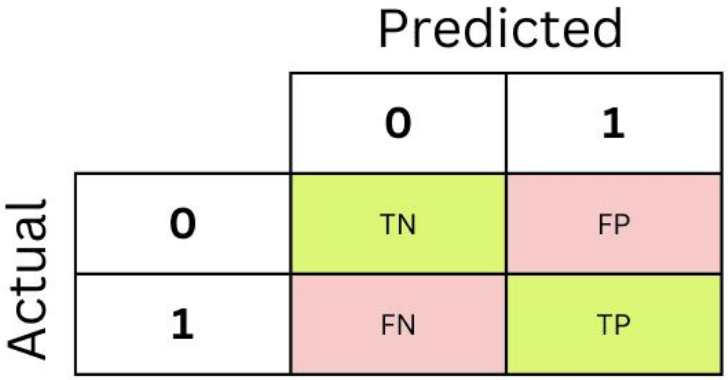
ATIVIDADE: KNN

- Escolha uma base de dados no https://www.kaggle.com/datasets, e se familiarize com sua base
- Procure realizar a previsão (inferência) de uma variável categórica através de um kNN. Se certifique de medir seus níveis de assertividade. Esteja a vontade a realizar mais um hiperparâmetro (número de vizinhos) de um e compará-los
- Não esqueça de junto com seus códigos realizar suas análises/conclusões (use o botão de +Texto).

Separação Treino Teste



Matriz de Confusão



$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

Matriz de Confusão