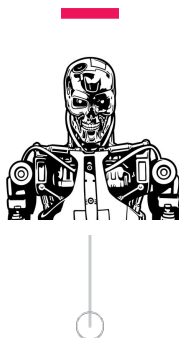
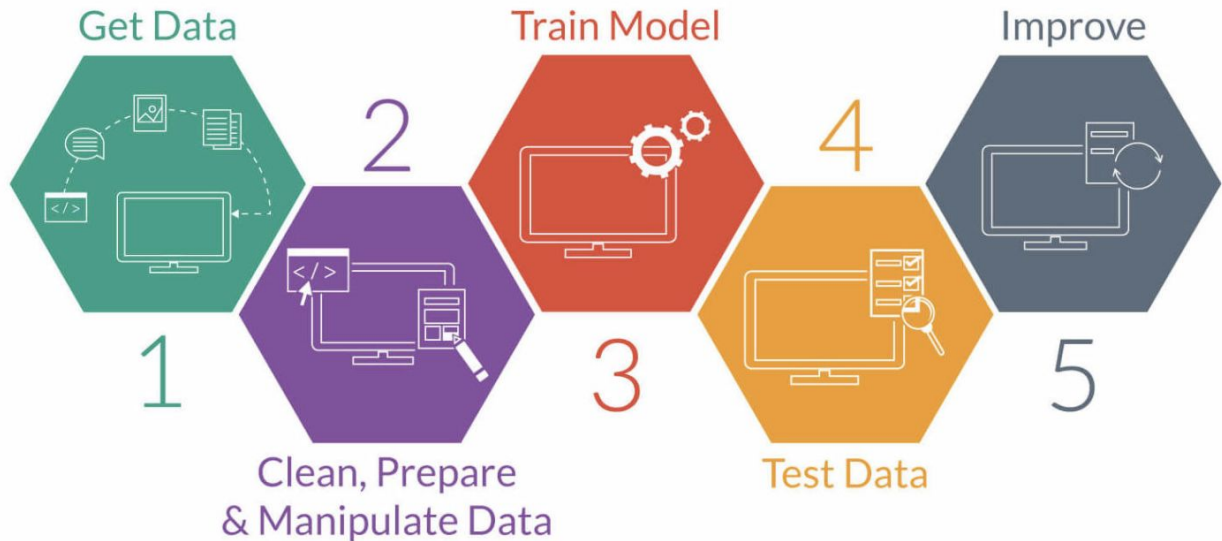


# Processo de IA



# Machine Learning (ML) - Processo



# Machine Learning (ML) - Processo

dados treino  
70-80%:

The diagram shows a table with 5 columns and 5 rows. A bracket above the first four columns is labeled 'Features', and a bracket above the last column is labeled 'Label'. A bracket to the left of all rows is labeled 'Rows', and a bracket below all columns is labeled 'Columns'.

Size	Beds	Baths	Zip	Price
1100	1	1	64576	1.29
1900	3	1.5	78321	2.14
2800	3	3	98712	3.10
3400	4	3.5	25721	3.75

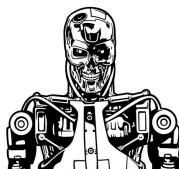
dados teste  
20-30%:

1900	3	1.5	78321
2800	3	3	98712

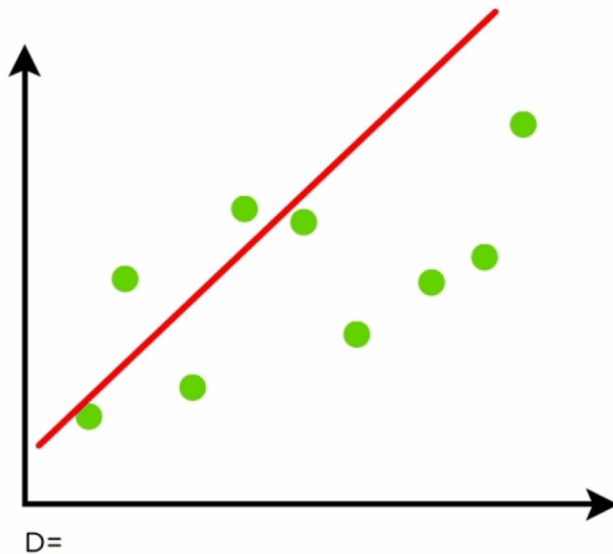
## Machine Learning (ML)

	<b><i>Supervisionado</i></b>	<b><i>Não Supervisionado</i></b>
<b>Projeção</b>	Regressão Linear	X
<b>Categorização</b>	<u>Classificação</u> - KNN - Naive Bayes - Arv Decisão / Forest - Sup Vector (SVM) - Reg Logística  <u>Rede Neural (NN)</u> - Deep NN	<u>Agrup. / Cluster</u> - KMeans - HClust  <u>Associação</u> - Apriori  <u>Redução Dimensional</u> - PCA

# Projeções: Regressão Linear



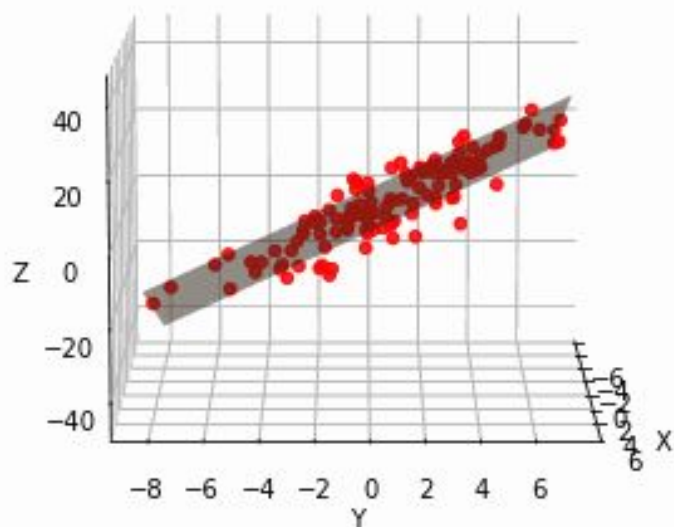
# Regressão Linear



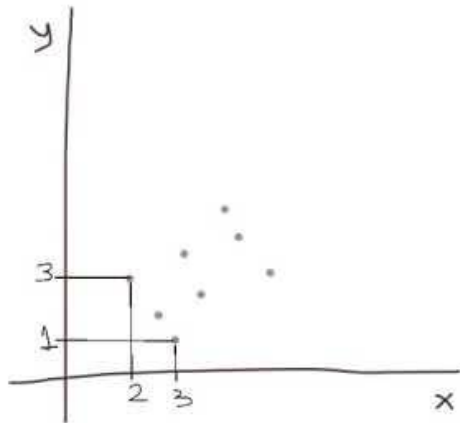
E qual a equação da reta?

$$y = ax + b$$

# Regressão Linear



# Regressão Linear



E qual a equação da reta?

$$y = ax + b$$

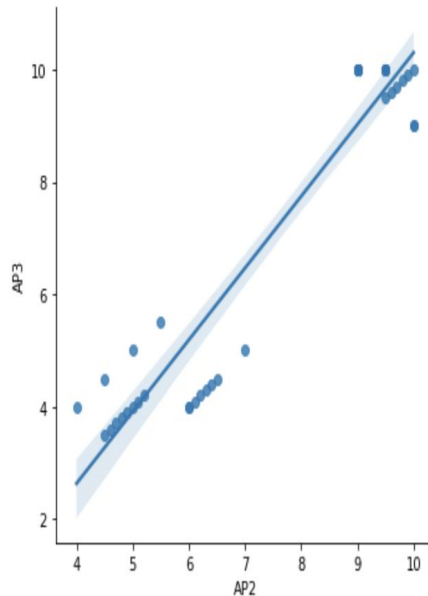


## Regressão Linear

- Use **lmplot()** para plotar o gráfico de dispersão da biblioteca **seaborn**.

```
sns.lmplot('AP2', 'AP3', data=nts)
```

<seaborn.axisgrid.FacetGrid at 0x7f80b3a5dd30>

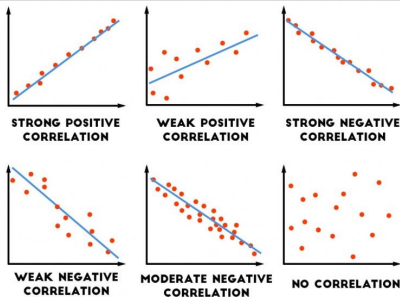


## Regressão Linear

- Análise a correlação com o método **corr()**.

```
nts.corr()
```

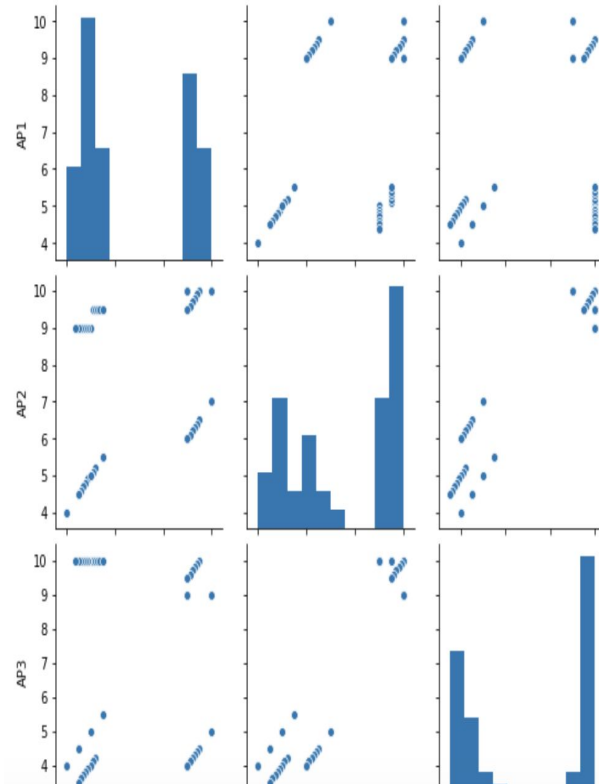
	AP1	AP2	AP3	Final
AP1	1.000000	0.278632	0.003717	0.630113
AP2	0.278632	1.000000	0.952691	0.920175
AP3	0.003717	0.952691	1.000000	0.776387
Final	0.630113	0.920175	0.776387	1.000000



- e visualize com **pairplot()**

```
sns.pairplot(nts)
```

```
<seaborn.axisgrid.PairGrid at 0x7f5020cf4828>
```



## Regressão Linear



- Uma das opções mais comuns, é utilizar o objeto do **LinearRegression** da biblioteca **sklearn**.
- Separar o **label/rótulo** (no ex, AP3) das **variáveis** (no ex, AP1 e AP2)

```
from sklearn.linear_model import LinearRegression
reglin = LinearRegression()
reglin.fit(nts[['AP1', 'AP2']], nts['AP3'])
```

- Uma vez treinado, temos definido no modelo os coeficientes e o intercepto

```
print(reglin.coef_, reglin.intercept_)
```

Assim te  $[-0.36608176 \quad 1.38338992] -0.8257714929014783 * -0,36 + AP2 * 1,38 -0.82$

## Regressão Linear

....Com isso poderíamos criar nossa uma funcao propria

```
def prever_ap3(ap1, ap2):
    return ap1 * -0.36 + ap2 * 1.38 - 0.82

print(prever_ap3(8,8))
print(prever_ap3(2,4))
print(prever_ap3(3.5,8))
```

7.34

3.98

8.959999999999999

- Ou simplesmente chamar a função **predict()**.

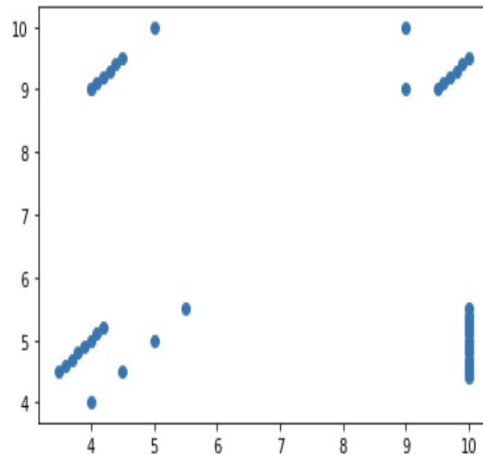
```
nts_futuras = pd.DataFrame({'AP1':[8,2,3.5],
                           'AP2':[8,4,8]})
reglin.predict(nts_futuras)
```

array([7.31269382, 3.97562468, 8.96006172])

## Regressão Linear

- Para plotar o Gráfico de Dispersão (Scaterplot), use **scatter()** da **matplotlib**.

```
nts = pd.read_csv('/content/drive/My Drive/IBMEC/notas - ok.csv')
plt.scatter(nts['AP3'], nts['AP1'])
```



## Regressão Linear

- Para medir a qualidade do modelo, o `LinearRegression()`, oferece basicamente a função **score()** que calcula o R2. Esta função recebe como parâmetros as variáveis preditoras, no exemplo AP1 e AP2, juntamente com os valores previstos esperados, no exemplo AP3.

```
r2=reglin.score(nts[['AP1','AP2']],nts['AP3'])
print('R2: ',r2)
```

R2: 0.9818896713333346

- Outras metricas podem ser obtidas pelo objeto **metrics**, da mesma biblioteca, que fornece funções como **R2**, **MSE** e **RMSE**. Para utiliza-las devem ser passados como parâmetros a variável prevista e seus valores esperados.

```
ap3_prev = reglin.predict(nts[['AP1','AP2']])

from sklearn import metrics
print('R2:', metrics.r2_score(nts['AP3'], ap3_prev))
print('MSE:', metrics.mean_squared_error(nts['AP3'], ap3_prev))
print('RSME:', np.sqrt(metrics.mean_squared_error(nts['AP3'], ap3_prev)))
```

R2: 0.9818896713333346

MSE: 0.14550924670519017

RSME: 0.38145674290172166

$$R^2 = 1 - \frac{\sum_{i=1}^n (\hat{y}_i - y_i)^2}{\sum_{i=1}^n (y_i - \bar{y})^2}$$

## Regressão Linear

- Caso se queira ter acesso a número maior de métricas sobre o modelo, recomenda-se o uso da biblioteca **statsmodel**. Em particular com sua função, OLS (mínimos quadrados ordinários). A única diferença no seu uso é que o dataframe de entrada precisa ter um coluna inicial com a constante 1.0

```
import statsmodels.api as sm

ap1_ap2 = sm.add_constant(nts[['AP1', 'AP2']])
rglin = sm.OLS(nts['AP3'], ap1_ap2).fit()
```

Obs: Veja

```
ap1_ap2.head()
```

	const	AP1	AP2
0	1.0	10.0	10.0
1	1.0	9.0	10.0
2	1.0	9.5	10.0
3	1.0	9.4	9.9
4	1.0	9.3	9.8

## Regressão Linear

- Para ver as métricas do modelo basta chamar o método, **summary()**

```
rglin.summary()
```

```
/usr/local/lib/python3.6/dist-packages/numpy/core/f:  
return ptp(axis=axis, out=out, **kwargs)
```

OLS Regression Results

Dep. Variable:	AP3	R-squared:	0.982
Model:	OLS	Adj. R-squared:	0.981
Method:	Least Squares	F-statistic:	1003.
Date:	Thu, 13 Feb 2020	Prob (F-statistic):	5.91e-33
Time:	19:56:05	Log-Likelihood:	-18.207
No. Observations:	40	AIC:	42.41
Df Residuals:	37	BIC:	47.48
Df Model:	2		

Covariance Type: nonrobust

	coef	std err	t	P> t	[0.025	0.975]
const	-0.3258	0.265	-3.111	0.004	-1.364	-0.288
AP1	-0.3661	0.030	-12.318	0.000	-0.426	-0.306
AP2	1.3834	0.031	44.788	0.000	1.321	1.446
Omnibus:	4.027	Durbin-Watson:	0.680			
Prob(Omnibus):	0.134	Jarque-Bera (JB):	3.534			
Skew:	0.642	Prob(JB):	0.171			
Kurtosis:	2.314	Cond. No.	43.9			

- E para o RMSE

```
np.sqrt(rglin.mse_resid)
```

```
0.39661983527723466
```



## Regressão Linear: Exercício

- Será que existe alguma relação entre a cotação do dólar e o índice da Bovespa (IBOV)?
- Essa relação é diretamente proporcional (positiva) ou negativa?
- Qual o racional econômico que explica essa relação?
- Crie então sua máquina para prever o futuro do dólar base na cotação do índice IBOV
- A Bolsa hoje está em quanto? E o dólar? Se o IBOV chegar em 100 mil pontos, qual o valor do dólar previsto?

dados: <https://raw.githubusercontent.com/lcbjrrr/data/main/usdbrl%20-%202024.csv>



# MAD MEN

Intuición · Creatividad · Appeal



# MATH MEN

Analytics · Resultados · Lógica



## ATIVIDADE: RegLin

- 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 numérica através de uma regressão linear. Se certifique que existe correlação entre as features escolhidas e o label a ser previsto,
- Se certifique em separar seu dados entre treino e teste, não esquecendo em realizar as medições de assertividade, ambas etapas.
- Não esqueça de junto com seus códigos realizar suas análises/conclusões (use o botão de +Texto).