



# APRENDIZAGEM DE MÁQUINA

PROF. JOSENALDE OLIVEIRA

[josenalde.oliveira@ufrn.br](mailto:josenalde.oliveira@ufrn.br)

<https://github.com/josenalde/machinelearning>

ANÁLISE E DESENVOLVIMENTO DE SISTEMAS - UFRN

# TESTES DE HIPÓTESES PARA AVALIAR MODELO

Um modelo de regressão pressupõe hipóteses em relação aos **resíduos**, que precisam ser TESTADOS para confiabilidade na significância dos parâmetros obtidos

- Hipóteses são AFIRMAÇÕES que queremos verificar como verdadeiras ou não (alegações sobre uma propriedade da população), considerando certo nível de **significância**

Resíduo (erro):  $y - \hat{y}$ , ou, seja, a diferença entre o valor observado e o valor predito pelo modelo

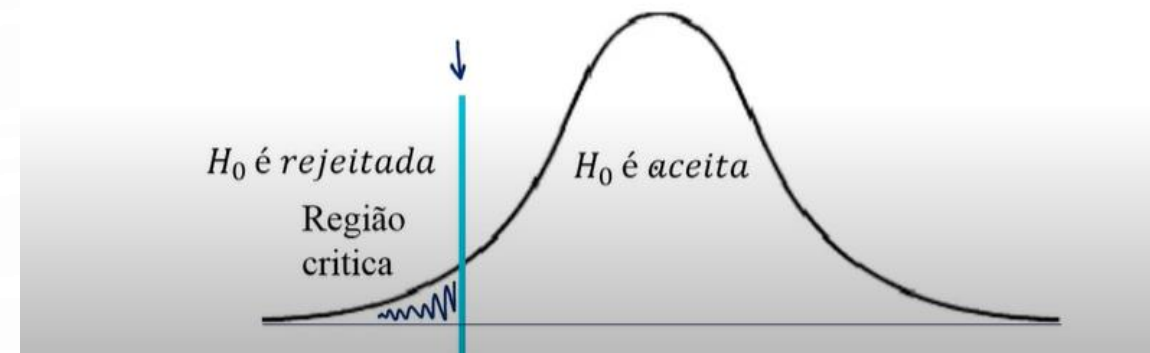
Significância:  $\alpha$ . Por exemplo, para confiança de 95% temos 5% de significância

- Cada propriedade possui um teste específico (normal padrão, t-student, qui-quadrado etc.)
- É necessário definir a **hipótese nula** (padrão,  $H_0$ ) e a **hipótese alternativa** ( $H_1$ ). Se a  $H_0$  falhar, é porque pode ser rejeitada, assumindo-se como verdade a hipótese alternativa.
- Em machine learning é usual considerar o teste unilateral à esquerda

$$H_0: \mu \geq \mu_0$$

$$H_1: \mu < \mu_0$$

**$\mu$  : média**



# TESTES DE HIPÓTESES PARA AVALIAR MODELO

- Em nossas aplicações, iremos focar na medida **VALOR-P** (p-value) e não no valor crítico associado ao teste em uso (consulta de tabela etc.). O valor-p indica a probabilidade do resultado ser extremo (laterais)

$\text{p-value} < \alpha \implies \text{rejeitar } H_0$

$\text{p-value} > \alpha \implies \text{aceitar } H_0$

- Exemplo: no teste de normalidade do scipy (normaltest) a hipótese nula é de que os dados são aderentes
- Estes testes exigem número mínimo de amostras para confiabilidade. Por exemplo, [Jarque-Bera](#) > 2000

```
from scipy import stats
res = stats.normaltest(residuos)

alpha = 5e-2 #5%
print("p-value = {:.g}".format(res.pvalue))

print('TEST: residuos aderentes à distribuição normal:')
if res.pvalue < alpha: # null hypothesis: x comes from a normal distribution
    print(f"{bcolors.FAIL}FAIL") #H0...FAIL
else:
    print(f"{bcolors.OKGREEN}PASSED") #H1...PASSED
```

- [Shapiro-wilk](#)

# TESTES DE HIPÓTESES PARA AVALIAR MODELO

- Teste de INDEPENDÊNCIA dos resíduos, ou seja, não estão correlacionados
- Exemplo: no teste de INDEPENDÊNCIA Durbin-Watson de statsmodels, a hipótese nula é de são independentes

```
from statsmodels.stats.stattools import durbin_watson

#perform Durbin-Watson test
dw = durbin_watson(resíduos)
print('TEST: resíduos independentes?:')
if (dw < 1.5 or dw > 2.5): # null hypothesis: x are independent
    print(f"{bcolors.FAIL}FAIL") #H0...FAIL
else:
    print(f"{bcolors.OKGREEN}PASSED") #H1...PASSED
```

- Neste teste, é retornado o valor da estatística em si, variando entre 0 e 4.
- Entre 1.5 e 2.5 é considerado normal. Próximo de 2 ideal, sem correlação. Próximo de 0, correlação positiva e próximo de 4 correlação negativa:  $2 * (1-r)$

# TESTES DE HIPÓTESES PARA AVALIAR MODELO

- Teste de HETEROCIDASTICIDADE dos resíduos
- Exemplo: no teste de HETEROCIDASTICIDADE White de statsmodels, a hipótese nula é de que são homocedasticos

```
import statsmodels.formula.api as smf
```

```
#fit regression model
```

```
fit = smf.ols('tempo ~ distancia', data=df). fit ()
```

```
#view model summary
```

```
print (fit.summary())
```

```
from statsmodels.stats.diagnostic import het_white
```

```
from statsmodels.compat import lzip
```

```
keys = ['Lagrange Multiplier statistic:', 'LM test\'s p-value:', 'F-statistic:', 'F value:']
```

```
results = het_white(fit.resid, fit.model.exog)
```

