Universidade Federal do Pará
Programa de Pós-graduação em Engenharia de Processos (PPGEP)
Disciplina: Tópicos Especiais em Engenharia de Processos
Prof. Dr. Alan de Souza
Aula 3

Setembro/2025

Na aula anterior...

1. Tipos de aprendizado;
2. Tipos de problemas;
3. Metodologia CRISP-DM;
4. Regressão linear e por A.D.;
5. Projeto prático envolvendo regressão linear.

Ainda sobre árvore de decisão

- Como as árvores de decisão decidem qual variável é melhor para ser o nó da vez para "fatiar" os dados de forma eficiente? Qual o critério?
- Índice Gini:

$$G = \frac{\sum_{i=1}^{n} \sum_{j=1}^{n} \|x_i - x_j\|}{2n^2 \bar{x}}$$

Ainda sobre árvore de decisão

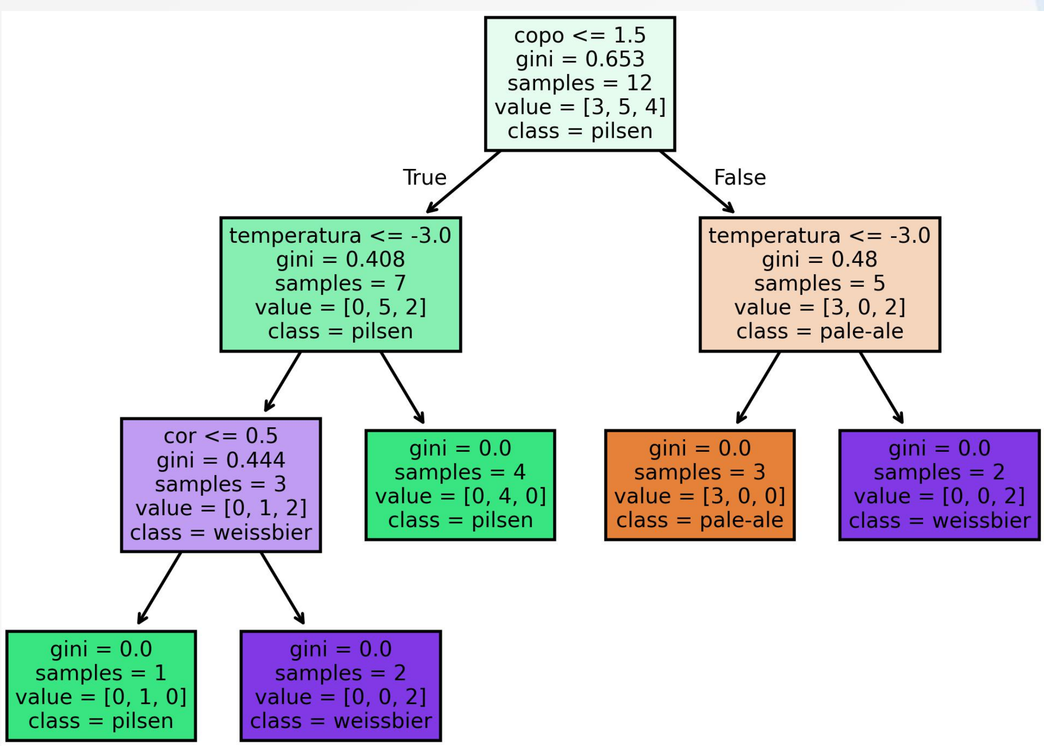
- Entropia:

$$H = -[p \log_2(p) + (1-p) \log_2(1-p)]$$

$$H = -\sum_{i=1}^{c} p_i \log_2(p_i)$$

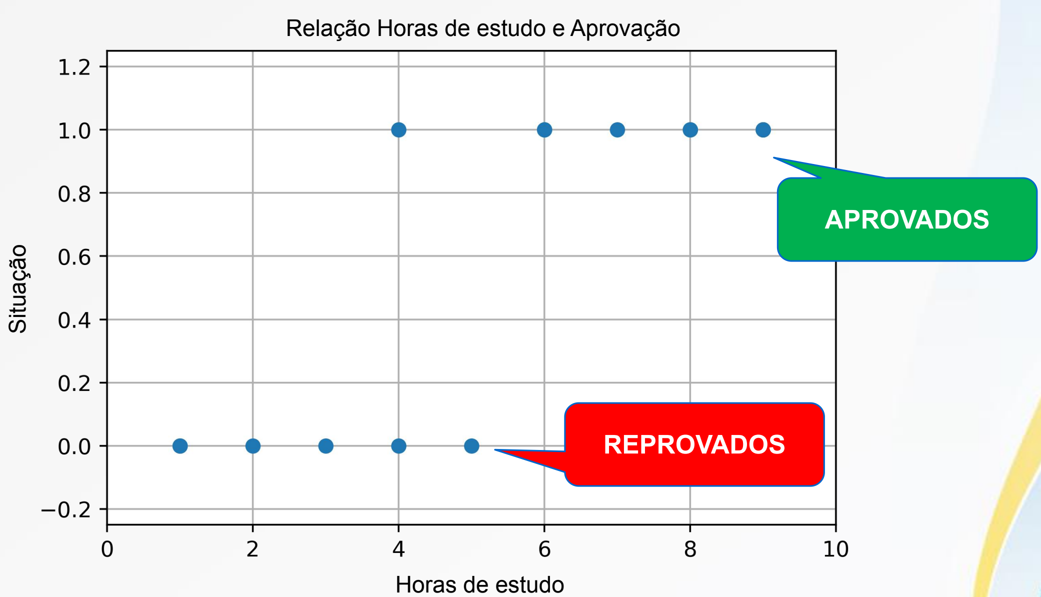
Ainda sobre árvore de decisão

- Quanto mais puro o nó, ou seja, quanto mais homogêneo o nó, menor o índice Gini/entropia;
- Portanto o algoritmo de AD trabalha para **minimizar** o Gini e a entropia;
- Documentação de AD Python:
- https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html

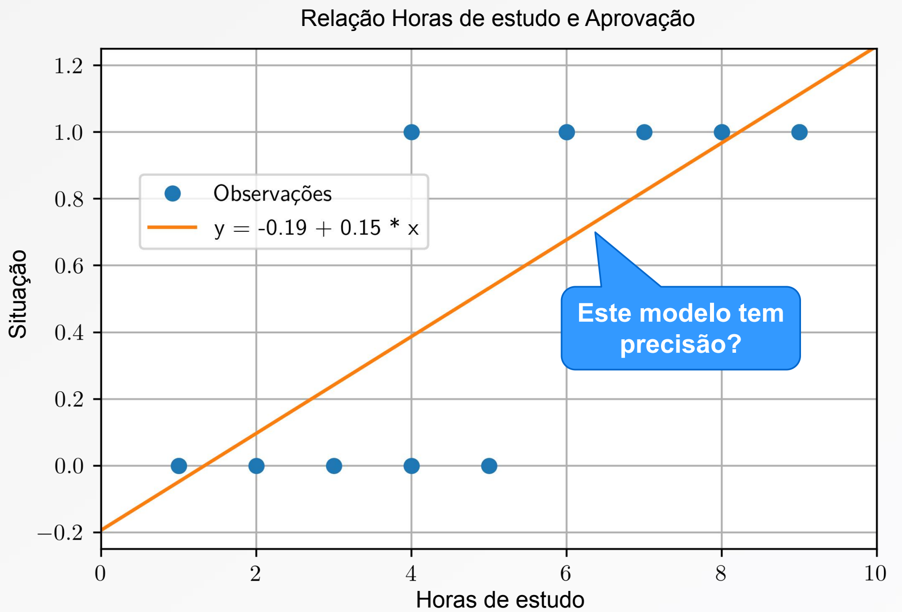


DÚVIDAS?

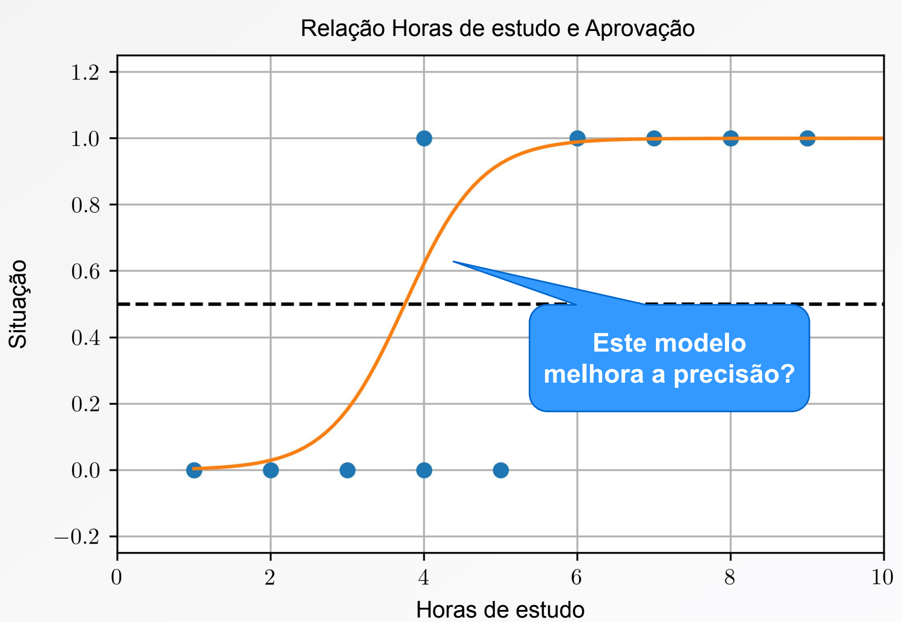
Problema de classificação usando regressão logística



Problema de classificação usando regressão logística



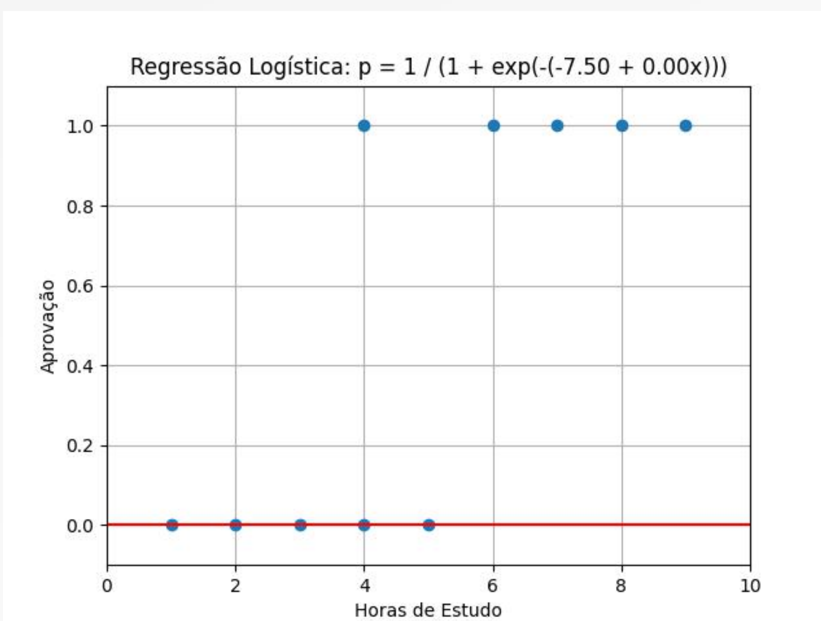
Problema de classificação usando regressão logística



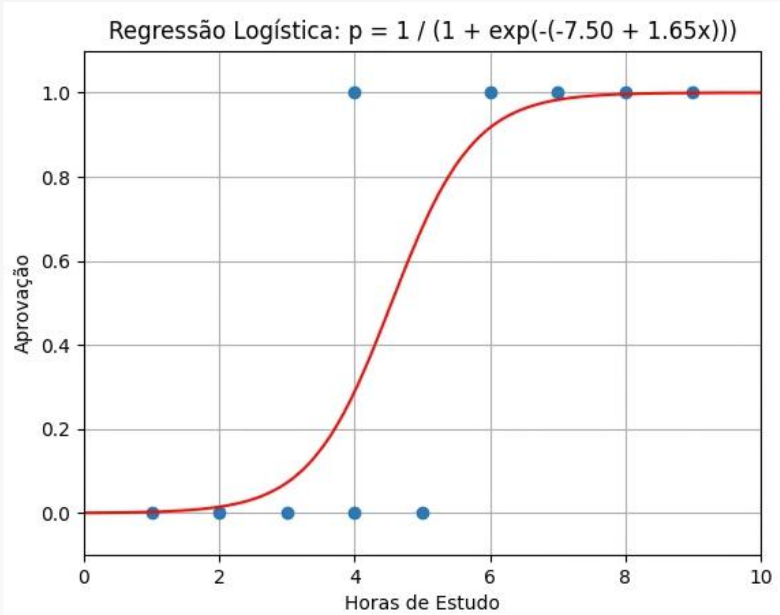
Problema de classificação usando regressão logística

$$y = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x_1)}}$$

Problema de classificação usando regressão logística



Problema de classificação usando regressão logística



Problema de classificação usando regressão logística

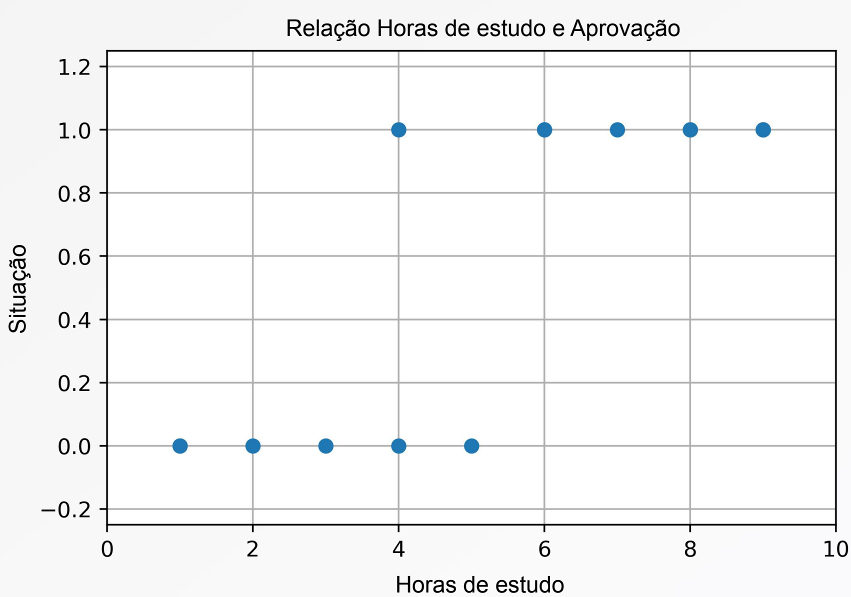
- Notação matemática:
- Log loss:

$$L_{\log}(y, p) = -(y\log(p) + (1-y)\log(1-p))$$

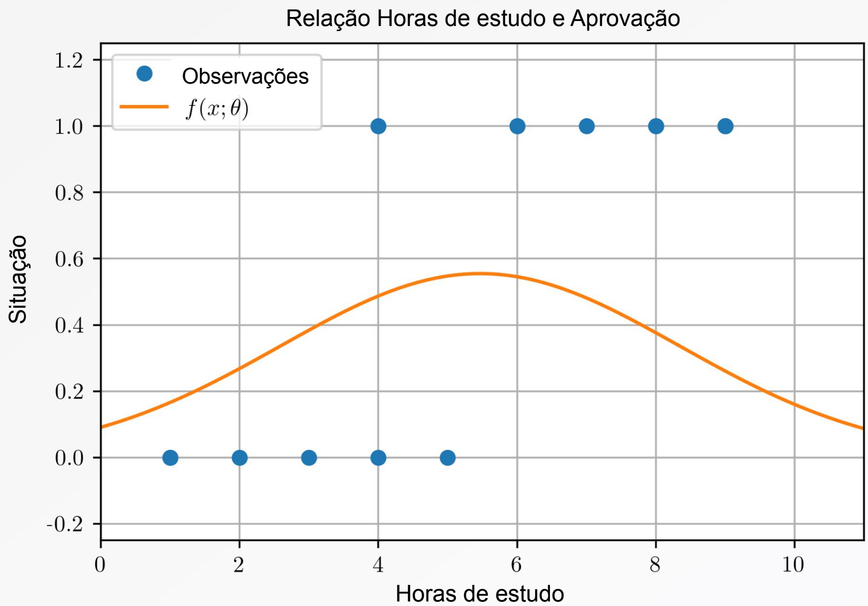
$$\sum_{i=1}^{n} L_{\log}(y_i, p_i) = -\sum_{i=1}^{n} (y_i\log(p_i) + (1-y_i)\log(1-p_i))$$

DÚVIDAS?

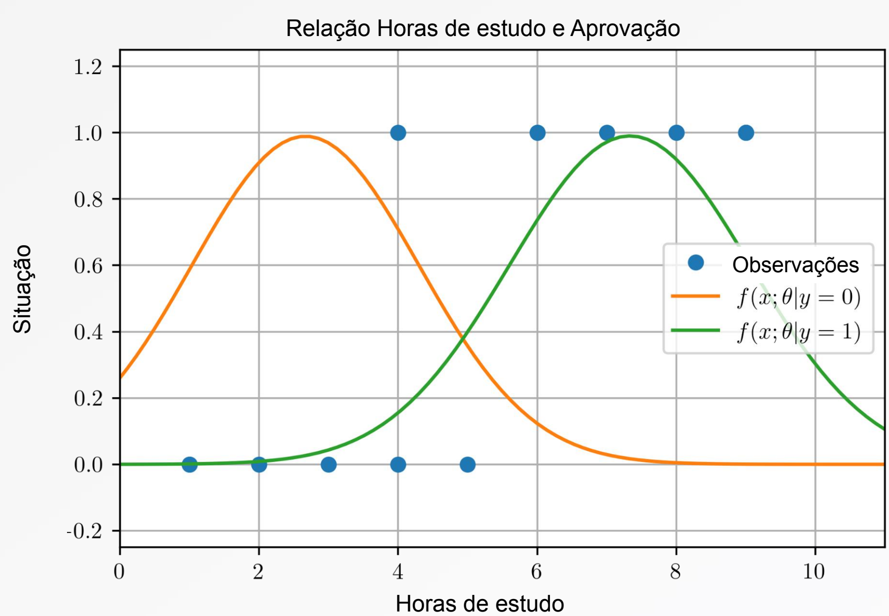
Naive Bayes



Naive Bayes



Naive Bayes



Naive Bayes

- Qual a probabilidade de ser diabético, dado algumas informações?

$$
\begin{aligned}
y &= \text{Diabetes}(1 = \text{Sim}; 0 = \text{Não}) \\
x_1 &= \text{Histórico Familiar}(1 = \text{Sim}; 0 = \text{Não}) \\
x_2 &= \text{Acima do peso}(1 = \text{Sim}; 0 = \text{Não}) \\
x_3 &= \text{Atividade Física}(1 = \text{Sim}; 0 = \text{Não})
\end{aligned}
$$

Naive Bayes

- Usando o Teorema de Bayes, temos:

$$P(y|X) = \frac{P(y)P(X|y)}{P(X)}$$

$$P(\underbrace{y}_{\substack{\text{Var. de saída} \\ \text{(sim, não)}}}|\underbrace{x_1, x_2, x_3}_{\substack{\text{Vars. de entrada} \\ \text{(hist, peso, ativ.)}}}) = \frac{P(y)P(x_1, x_2, x_3|y)}{P(x_1, x_2, x_3)}$$

Naive Bayes

Diabetes	Histórico Familiar	Acima do Peso	Atividade Física
1	1	1	1
1	1	1	0
1	0	1	0
1	1	0	0
0	1	0	1
0	0	1	1
0	0	0	0

Qual a probabilidade de ser diabético?

$$P(y = 1|X)$$

- Têm histórico familiar
- Não está acima do peso
- Não pratica atividade física

ou seja:

$x_1 = 1$; $x_2 = 0$; $x_3 = 0$

$$P(y = 1|x_1 = 1, x_2 = 0, x_3 = 0) = \frac{P(y = 1)P(x_1 = 1, x_2 = 0, x_3 = 0|y = 1)}{P(x_1 = 1, x_2 = 0, x_3 = 0)}$$

Diabetes	Histórico Familiar	Acima do Peso	Atividade Física
1	1	1	1
1	1	1	0
1	0	1	0
1	1	0	0
0	1	0	1
0	0	1	1
0	0	0	0

$$\frac{P(y = 1)P(x_1 = 1|y = 1)P(x_2 = 0|y = 1)P(x_3 = 0|y = 1)}{P(x_1 = 1, x_2 = 0, x_3 = 0)}$$

$$\left.\begin{aligned} P(y = 1) &= 4/7 \\ P(x_1 = 1|y = 1) &= 3/4 \\ P(x_2 = 0|y = 1) &= 1/4 \\ P(x_3 = 0|y = 1) &= 3/4 \end{aligned}\right\} = 9/112 = 0,08035$$

$$P(y=0|x_1=1,x_2=0,x_3=0)=\frac{P(y=0)P(x_1=1,x_2=0,x_3=0|y=0)}{P(x_1=1,x_2=0,x_3=0)}$$

Diabetes	Histórico Familiar	Acima do Peso	Atividade Física
1	1	1	1
1	1	1	0
1	0	1	0
1	1	0	0
0	1	0	1
0	0	1	1
0	0	0	0

$$=\frac{P(y=0)P(x_1=1|y=0)P(x_2=0|y=0)P(x_3=0|y=0)}{P(x_1=1,x_2=0,x_3=0)}$$

$$\left.\begin{array}{l}P(y=0)=3/7\\P(x_1=1|y=0)=1/3\\P(x_2=0|y=0)=1/3\\P(x_3=0|y=0)=2/3\end{array}\right\}\begin{array}{l}=2/63\\=0,03174\end{array}$$

Naive Bayes

- Conclusão:
  - Como P_{sim} > $P_{não}$ (0,08 > 0,03), então a classe estimada é que a pessoa tem diabetes.

DÚVIDAS?

Projeto prático - Situação x Horas de estudo

1) Baixe o arquivo dados_estudo_nota_situacao.xlsx;
2) Faça a importação no projeto Python;
3) Se necessário, converta textos para números;
4) Crie três modelos de classificação baseados nos algoritmos: árvore de decisão (AD), regressão logística (RL), naive bayes (NB).
5) Use o modelo criado para fazer predições.
6) Crie o gráfico de comparação entre dados reais *versus* dados estimados pelo modelo.
   – O que podemos concluir comparando os modelos?

Projeto prático - Situação x Horas de estudo

```
# Importação de dados
import pandas as pd
df = pd.read_excel('dados_estudo_nota_situacao.xlsx')
df
```

Projeto prático - Situação x Horas de estudo

```
# Convertendo texto para número (coluna situação)
df['situacao'] = df['situacao'].replace({
    "reprovado": 0,
    "aprovado": 1
})
df
```

Projeto prático - Situação x Horas de estudo

```
# Criando os modelos:
from sklearn import linear_model
from sklearn import tree
from sklearn import naive_bayes
modelo_rl = linear_model.LogisticRegression(fit_intercept=True,
                                            penalty=None)
modelo_ad = tree.DecisionTreeClassifier(random_state=42,
                                        max_depth=3)
modelo_nb = naive_bayes.GaussianNB()
```

Projeto prático - Situação x Horas de estudo

```
# treinamento dos modelos:
modelo_rl.fit(X, y)
modelo_ad.fit(X, y)
modelo_nb.fit(X, y)
```

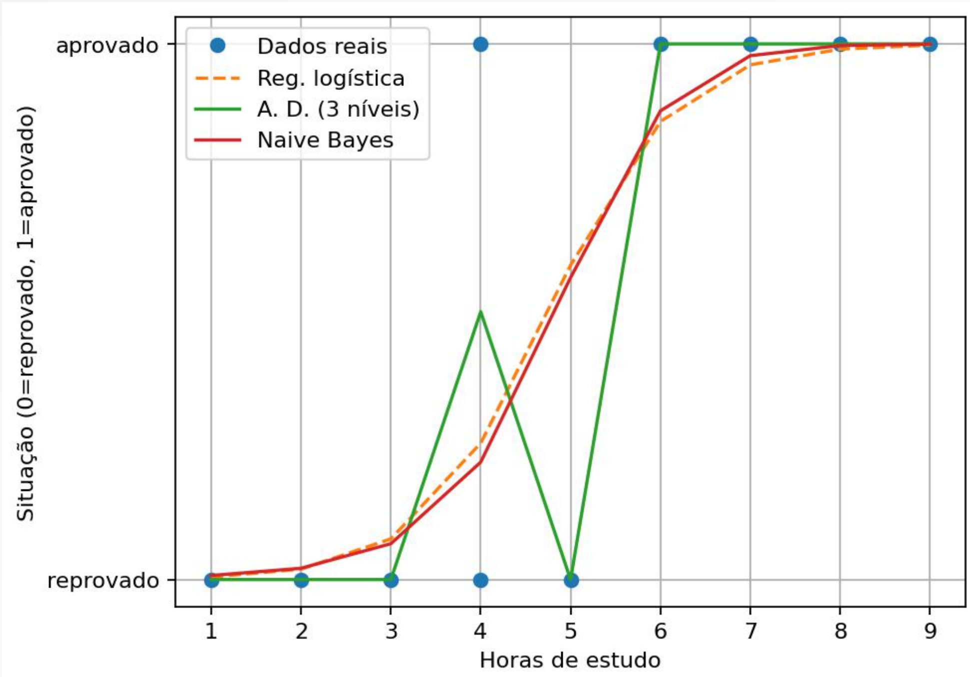
Projeto prático - Situação x Horas de estudo

```
# Usando o modelo para fazer previsões:
previsao_rl = modelo_rl.predict_proba(X)[:,1]
previsao_ad = modelo_ad.predict_proba(X)[:,1]
previsao_nb = modelo_nb.predict_proba(X)[:,1]
```

Projeto prático - Situação x Horas de estudo

```
# gráfico compartivo entre dados reais vs estimados pelos modelos:
import matplotlib.pyplot as plt
plt.figure(dpi=160)
plt.plot(df['horas_estudo'], df['situacao'], 'o', label='Dados reais')
plt.plot(df['horas_estudo'], previsao_rl, '--', label='Reg. logística')
plt.plot(df['horas_estudo'], previsao_ad, label='A. D. (3 níveis)')
plt.plot(df['horas_estudo'], previsao_nb, label='Naive Bayes')
plt.grid(True)
plt.xlabel('Horas de estudo')
plt.ylabel('Situação (0=reprovado, 1=aprovado)')
plt.legend()
plt.show()
```

Resultado



DÚVIDAS?