

## Análise de Sentimento com Naive-Bayes e N-grams

<b>Entrega</b>	Arquivo .ipynb com células já executadas
<b>Modalidade</b>	Individual

### Objetivo

Construir e avaliar um modelo de Machine Learning (**Naive-Bayes Multinomial**) para classificar reviews em três categorias de sentimento: **Negativo** (0), **Neutro** (1), e **Positivo** (2).

### Dataset

- **Arquivo:** B2W-Reviews01.csv

### Roteiro do Notebook (9 Etapas)

Seu notebook deve ser claro, bem documentado (usando Markdown) e rodar sequencialmente.

#	Seção	Instruções e Requisitos
1	<b>Preparação do Ambiente</b>	Importar as bibliotecas necessárias (pandas, nltk, sklearn, seaborn, matplotlib). Baixar recursos NLTK (punkt, stopwords).
2	<b>Carregamento e Limpeza</b>	Carregar o DataFrame (review_text, overall_rating). <b>Remover linhas com valores nulos</b> (dropna()). Exibir o total de reviews restantes.
3	<b>Definição das Classes</b>	Criar a função de mapeamento de notas para 3 classes: <b>1 → 0 (Negativo)</b> , <b>2, 3, 4 → 1 (Neutro)</b> , <b>5 → 2 (Positivo)</b> . Aplicar e exibir a distribuição das novas classes.
4	<b>Pré-processamento (NLP)</b>	Criar e aplicar a função preprocessText para: remover pontuação, converter para minúsculas, tokenizar, remover <i>stopwords</i> (Português) e dígitos. Gerar a coluna texto_pre.
5	<b>Vetorização N-grams</b>	Instanciar <b>CountVectorizer</b> com <b>ngram_range=(1,3)</b> . <b>Ajustar/Transformar</b> texto_pre para criar a matriz de recursos <b>X</b> . Imprimir o X.shape.

6	<b>Divisão dos Dados</b>	Usar <code>train_test_split</code> (20% para teste, <code>random_state=42</code> ) para separar os dados. Imprimir os <i>shapes</i> dos conjuntos de treino e teste.
7	<b>Treinamento Naive-Bayes</b>	Instanciar e treinar (fit) o modelo <b>MultinomialNB()</b> . Imprimir a acurácia de treino e teste.
8	<b>Avaliação Completa</b>	Gerar o <b>classification_report</b> . Plotar o <b>confusion_matrix</b> como um <b>heatmap</b> (seaborn) para visualização da performance por classe.
9	<b>Teste Manual (Demo)</b>	Criar e rodar uma função interativa ( <code>input()</code> ) que: processa o texto do usuário, vetoriza, <b>prevê a classe</b> e exibe o resultado final com o nível de <b>confiança</b> ( <code>predict_proba</code> ).

#### Foco na Avaliação (Criatividade e Documentação)

- **Documentação:** Cada seção deve ter um breve cabeçalho em **Markdown** explicando o propósito da etapa.
- **Análise:** Inclua comentários (em Markdown ou código) sobre o volume de dados perdido (Etapa 2), o balanceamento das classes (Etapa 3), e a interpretação dos resultados do `classification_report` (Etapa 8).
- **Teste Criativo:** Crie 5 exemplos de frases distintas para demonstrar o funcionamento e os limites do modelo na Etapa 9.