

Naive Bayes, Classificação de Textos e Sentimentos

Introdução à Classificação

A classificação é uma tarefa central tanto na inteligência humana quanto na artificial. Envolve decidir a qual categoria uma determinada informação pertence. No contexto de **Processamento de Linguagem Natural (PLN)**, essa tarefa é aplicada em diversas atividades, como:

- **Análise de sentimentos:** identificar se um texto expressa uma opinião positiva, negativa ou neutra.
- **Deteção de spam:** classificar e-mails como "spam" ou "não-spam".
- **Identificação de idioma:** detectar automaticamente o idioma de um texto.
- **Atribuição de autoria:** estimar quem escreveu um determinado texto.
- **Classificação temática:** rotular textos por assunto, como artigos jornalísticos ou científicos.

Essas tarefas geralmente usam **aprendizado supervisionado**, em que um modelo aprende a partir de exemplos rotulados para classificar novos dados.

O Classificador Naive Bayes

O **Naive Bayes** é um algoritmo probabilístico baseado no Teorema de Bayes. Ele é chamado de "ingênuo" (naive) porque assume que as características (como palavras em um texto) são **condicionalmente independentes** umas das outras, dado a classe. Essa suposição é raramente verdadeira, mas funciona surpreendentemente bem em muitas aplicações práticas de PLN.

Características do Naive Bayes:

- **Modelo generativo:** tenta modelar como os dados são gerados para cada classe.
- **Simples e eficiente:** funciona bem mesmo com conjuntos de dados pequenos.
- **Boa performance em textos:** especialmente em tarefas como análise de sentimentos e filtragem de spam.

Como o Naive Bayes funciona na prática

1. Representação dos textos

Os textos são transformados em vetores de características usando o modelo **Bag-of-Words (BoW)**, que ignora a ordem das palavras e considera apenas a presença ou frequência de cada uma.

2. Treinamento do modelo

Durante o treinamento, o modelo aprende:

- A **probabilidade de cada classe** (positiva, negativa etc.) com base na frequência nos dados.
- A **probabilidade de cada palavra dentro de uma classe**, estimada pela frequência da palavra em documentos dessa classe.

3. Classificação de novos textos

Para um novo documento, o modelo calcula, para cada classe, a probabilidade de o texto pertencer àquela classe com base nas palavras presentes. A classe com maior probabilidade é atribuída ao documento.

Suavização de Laplace

Um problema comum ocorre quando uma palavra do texto de teste nunca apareceu nos dados de treino para uma classe: isso gera uma probabilidade zero, anulando toda a multiplicação. A solução é a **suavização add-one (ou Laplace)**, que adiciona 1 a todas as contagens, evitando esse problema.

Otimizações comuns

1. **Binarização**: em vez de contar quantas vezes uma palavra aparece, considera-se apenas se ela aparece ou não. Isso pode melhorar a robustez do modelo em certos casos.
2. **Tratamento de negação**: palavras como “não” ou “nunca” podem inverter o significado. Uma técnica útil é modificar o token seguinte (ex.: “não gostei” → “NOT_gostei”).
3. **Uso de léxicos sentimentais**: listas pré-definidas de palavras positivas e negativas podem ser usadas para enriquecer a classificação ou como uma abordagem alternativa simples.

Exemplo prático resumido

Cenário:

- **Frases de treino**: 5 frases, sendo 3 negativas e 2 positivas.
- **Frase de teste**: "predictable with no fun".

Etapas:

- Calcula-se a probabilidade de cada classe com base nos dados de treino.
- Estima-se a probabilidade de cada palavra dada a classe.
- Ignora-se palavras fora do vocabulário de treino (como “with”).

- O modelo combina essas probabilidades para cada classe e escolhe a mais provável.

Métricas de Avaliação

Para avaliar o desempenho do classificador, usamos métricas como:

- **Acurácia:** proporção total de acertos.
- **Precisão:** entre os documentos classificados como positivos, quantos realmente são positivos.
- **Recall** (revocação): entre os documentos realmente positivos, quantos foram identificados corretamente.
- **F1-Score:** média harmônica entre precisão e recall, útil quando há desbalanceamento entre classes.

Essas métricas são baseadas na **matriz de confusão**, que mostra os acertos e erros do modelo em cada classe.

Considerações finais

Apesar de ser um algoritmo simples e baseado em suposições fortes, o Naive Bayes continua sendo amplamente usado em tarefas de classificação de texto por sua:

- Eficiência computacional
- Facilidade de implementação
- Bom desempenho em muitos domínios práticos

Ele é particularmente útil como ponto de partida em projetos de NLP, além de servir como baseline comparativo para algoritmos mais sofisticados.