

Aprendizado com Classes Desbalanceadas

Cesar de Oliveira F. Silva

Setembro / 2026

Roteiro

- 1 Introdução
- 2 Acurácia e Matriz de Confusão
- 3 Espaço ROC e AUC
- 4 Técnicas de Reamostragem
- 5 Encerramento

O que são classes desbalanceadas?

- Dizemos que um problema tem **classes desbalanceadas** quando há grande diferença na quantidade de exemplos de cada classe.
- Ex.: 0,1% de transações fraudulentas vs. 99,9% legítimas.
- Intuição: é como procurar **agulha no palheiro** – muitos exemplos da classe comum e pouquíssimos da classe rara.
- Consequência: classificadores tendem a “ignorar” a classe minoritária e acertar quase sempre a classe majoritária.

Quando isso é (ou não é) um problema?

- Nem todo conjunto desbalanceado é difícil:
 - Se as classes são muito bem separadas, o modelo pode ter bom desempenho mesmo com poucos exemplos da minoria.
- Mas, em geral, o desbalanceamento:
 - dificulta o aprendizado da classe de interesse;
 - pode enganar as métricas tradicionais (como acurácia);
 - aumenta a chance de **erros graves** na classe minoritária.
- Foco típico: a classe menos frequente costuma ser a mais importante (doenças raras, fraudes, desastres naturais).

Desafios principais

- **Aprender com poucos exemplos** positivos (classe minoritária).
- **Evitar vieses** a favor da classe majoritária.
- **Medir corretamente** o desempenho do classificador.
- Em termos de analogia:
 - É como treinar um guarda para reconhecer um animal raro vendo poucas fotos dele e milhares de outros animais.

Taxa de acerto (acurácia)

- Acurácia é a métrica mais usada, mas pode ser **enganosa**.
- Definição para duas classes:

$$\text{Acurácia} = \frac{\text{nº de acertos}}{\text{nº total de exemplos}} = \frac{TP + TN}{TP + TN + FP + FN}$$

onde:

- TP = verdadeiros positivos,
 - TN = verdadeiros negativos,
 - FP = falsos positivos,
 - FN = falsos negativos.
- Pergunta: um classificador com 99,9% de acerto é sempre bom?

Exemplo extremo de acurácia

- Problema médico:
 - C1 = pacientes com câncer (4 pacientes);
 - C2 = pacientes saudáveis (500 pacientes).
- Classificador trivial: sempre predizer **saudável**.
 - Acerta os 500 saudáveis $\Rightarrow$ 500 acertos.
 - Erra todos os 4 pacientes com câncer.

- Acurácia:

$$\text{Acurácia} = \frac{500}{504} \approx 99,2\%.$$

- Intuição: para o hospital, este é um classificador **inútil**, embora a acurácia seja alta.

Matriz de confusão

- Ferramenta central para analisar erros por classe.

	Predito Positivo	Predito Negativo
Real Positivo	TP	FN
Real Negativo	FP	TN

- Permite definir métricas focadas na classe positiva:

$$\text{Recall} = \frac{TP}{TP + FN}, \quad \text{Precisão} = \frac{TP}{TP + FP}$$

- Em problemas desbalanceados, **recall da classe positiva** costuma ser mais importante que a acurácia global.

Acurácia pode confundir

- Imagine dois classificadores, ambos com 99,9% de acerto.
- Exemplo simplificado (valores em % dos dados):

Classificador A

	PPos	PNeg
Pos	0	0,1
Neg	0	99,9

Nenhum positivo é detectado.

Classificador B

	PPos	PNeg
Pos	0,09	0,01
Neg	0,09	99,81

Aqui alguns positivos são encontrados.

- Mesma acurácia, mas comportamento **muito diferente** na classe que importa.

Classe menos frequente (positiva)

- Em muitos problemas, a classe minoritária é a de maior interesse:
 - doente vs. saudável;
 - geada vs. clima normal;
 - transação fraudulenta vs. legítima.
- Um classificador que “esquece” a classe positiva é como um detector de fumaça que nunca dispara: **parece silencioso, mas não serve para nada.**

Proporção de exemplos pode mudar

- A prevalência das classes não é fixa:
 - epidemias, mudanças climáticas, alterações de uso do solo.
- A acurácia depende diretamente dessa proporção.

Acurácia = $P(\text{acerto em positivo})P(\text{positivo}) + P(\text{acerto em negativo})P(\text{negativo})$

- Se a proporção muda, a acurácia muda **mesmo que o modelo não tenha “esquecido” nada.**
- Precisamos de métricas menos sensíveis à prevalência.

Espaço ROC

- ROC = *Receiver Operating Characteristic*.
- Nasceu em detecção de sinais (radar): trade-off entre **detectar alvos** e **falsos alarmes**.
- No contexto de classificação binária:
 - Eixo x: **Taxa de Falsos Positivos** (FPR)

$$\text{FPR} = \frac{FP}{FP + TN}$$

- Eixo y: **Taxa de Verdadeiros Positivos** (TPR ou Recall)

$$\text{TPR} = \frac{TP}{TP + FN}$$

- Cada classificador $\Rightarrow$ um ponto no espaço ROC.

Classificador aleatório e diagonal

- Um classificador puramente aleatório produz pontos próximos à diagonal:

$$\text{TPR} \approx \text{FPR}$$

- Regra de ouro:
 - Acima** da diagonal: melhor que o acaso.
 - Na** diagonal: desempenho aleatório.
 - Abaixo** da diagonal: pior que aleatório (basta inverter a decisão para melhorar).
- Visualmente, queremos pontos rumo ao canto superior esquerdo (0, 1): **poucos falsos positivos, muitos verdadeiros positivos.**

Fecho convexo (Convex Hull)

- Dado um conjunto de pontos ROC, o **fecho convexo** é o menor polígono que contém todos os pontos.
- Intuição geométrica:
 - Classificadores “dentro” do fecho convexo nunca são melhores que algum ponto na borda.
- Logo, ao comparar modelos, focamos nos pontos da borda convexa:
 - são os que podem ser ótimos para alguma combinação de custos e prevalências.

Linhas de isodesempenho

- No espaço ROC, linhas com mesma inclinação representam **mesmo custo esperado** para um dado cenário de:
 - custos de erro (C_{FN} , C_{FP});
 - prevalências $P(\text{Pos})$, $P(\text{Neg})$.
- Para uma combinação fixa de custos e prevalências, o melhor classificador é o ponto do fecho convexo que toca a linha de isodesempenho mais alta.
- Dois modelos podem ter o mesmo custo global, mas errar de forma diferente nas duas classes.

Gerando uma curva ROC

- Suponha que o classificador produza uma **pontuação** $s(x)$ (probabilidade ou score) para cada instância.
- Para cada limiar τ :
 - prever Positivo se $s(x) \geq \tau$;
 - computar $(FPR(\tau), TPR(\tau))$.
- Variando τ de 1 até 0, obtemos uma curva ROC:

$$ROC = \{(FPR(\tau), TPR(\tau)) : 0 \leq \tau \leq 1\}$$

- Classificadores como redes neurais, regressão logística e modelos bayesianos fornecem naturalmente esses scores.

Comparando curvas ROC e AUC

- Se duas curvas ROC não se cruzam:
 - aquela mais próxima do canto (0, 1) é melhor.
- Resumo em um número: **Área sob a curva** (AUC).
 - Probabilidade de um exemplo positivo receber score maior que um negativo.
 - Calculada numericamente, por exemplo com a regra do trapézio.
- Interpretação:
 - $AUC = 0,5 \Rightarrow$ desempenho aleatório;
 - $AUC = 1 \Rightarrow$ separação perfeita.

Por que reamostrar as classes?

- Ideia geral: **ajustar a distribuição das classes** no **conjunto de treino** para:
 - dar mais “voz” à classe minoritária;
 - reduzir o viés do algoritmo;
 - facilitar o uso de métricas padrão como acurácia.
- Importante:

Regra de ouro

Técnicas de balanceamento devem ser aplicadas **apenas** no conjunto de **treino**, nunca no conjunto de teste.

Random Under- e Over-sampling

- **Random Under-sampling:**

- Remove aleatoriamente exemplos da classe majoritária.
- Prós: reduz o tamanho do conjunto; treino mais rápido.
- Contras: pode jogar fora exemplos importantes.

- **Random Over-sampling:**

- Replica aleatoriamente exemplos da classe minoritária.
- Prós: simples, preserva toda a informação da minoria.
- Contras: aumenta risco de **overfitting** (o modelo “decora” exemplos repetidos).

Limitações e extensões

- Under/over-sampling puros ignoram a estrutura dos dados.
- Para melhorar, surgem métodos que usam **heurísticas** ou modelos auxiliares:
 - Métodos de **limpeza** de dados (under-sampling guiado).
 - Métodos de **geração sintética** de exemplos.
- Muitos deles usam o algoritmo dos **k-vizinhos mais próximos** (k-NN) para identificar vizinhanças relevantes.

Edited Nearest Neighbor (ENN)

- ENN é um método de **limpeza** (data cleaning).
- Ideia:
 - 1 Para cada exemplo E_i , encontrar seus 3 vizinhos mais próximos.
 - 2 Se pelo menos 2 desses vizinhos têm classe diferente de E_i , então E_i é removido.
- Efeito:
 - remove exemplos “suspeitos” ou ruidosos;
 - bordas de decisão ficam mais suaves;
 - pode servir como under-sampling inteligente.

Neighborhood Cleaning Rule (NCL)

- Extensão do ENN, voltada a under-sampling da majoritária.
- Para cada exemplo E_i :
 - Se E_i é da classe **majoritária** e é contradito pelos 3 vizinhos $\Rightarrow$ remove E_i .
 - Se E_i é da classe **minoritária** e é contradito pelos vizinhos $\Rightarrow$ remove os **vizinhos majoritários** que o contradizem.
- Intuição:
 - protege exemplos da minoria;
 - elimina majoritários “invasores” próximos à fronteira.

SMOTE: Synthetic Minority Over-sampling Technique

- SMOTE é um método de **over-sampling inteligente**.
- Em vez de **replicar** exemplos, ele **gera novos** exemplos sintéticos na classe minoritária.
- Passos simplificados:
 - 1 Para cada exemplo minoritário x , encontrar seus k vizinhos minoritários.
 - 2 Escolher um vizinho x_{NN} aleatoriamente.
 - 3 Criar um novo exemplo:

$$x_{\text{novo}} = x + \lambda(x_{NN} - x), \quad \lambda \sim U(0, 1)$$

- Assim, os exemplos sintéticos ficam na “linha” entre dois casos reais, explorando melhor o espaço da minoria.

SMOTE: vantagens e cuidados

- Vantagens:
 - reduz overfitting em relação ao over-sampling simples;
 - preenche “buracos” na região da classe minoritária.
- Cuidados:
 - pode gerar exemplos em regiões ambíguas (perto da fronteira);
 - pode ser combinado com técnicas de limpeza (ex.: SMOTE + ENN).
- Sempre lembrar:

Treino vs. Teste

SMOTE e outros métodos de reamostragem são aplicados **apenas** no conjunto de treino. O teste deve refletir a distribuição real.

Resumo

- Em problemas com classes desbalanceadas:
 - acurácia sozinha é **insuficiente**;
 - devemos olhar matriz de confusão, recall, precisão, F1, AUC.
- O espaço ROC e a AUC fornecem visão global do desempenho sob diferentes limiares.
- Técnicas de reamostragem (under/over-sampling, ENN, NCL, SMOTE) ajudam a treinar modelos mais sensíveis à classe minoritária.
- O objetivo final é simples de enunciar, mas difícil de alcançar:

Objetivo

Detectar **bem** os poucos casos importantes sem explodir a quantidade de falsos alarmes.