Uso de estratégias de reamostragem para correção de desbalanceamento entre classes em modelos de classificação

DEMIAN B. O. GRAMS
Orientador: Dr. João Henrique F. Flores

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

22 de agosto de 2024

Introdução

O trabalho tem 3 ingredientes principais:

- Dados (categóricos) desbalanceados
  - Variável resposta binária
- Estratégias de reamostragem
  - Sobreamostragem
  - Subamostragem
  - Mista
- Modelos de classificação
  - Regressão logística
  - *Support vector classifier* (SVC)

Introdução

Após o treinamento dos modelos é preciso avaliar seu desempenho.

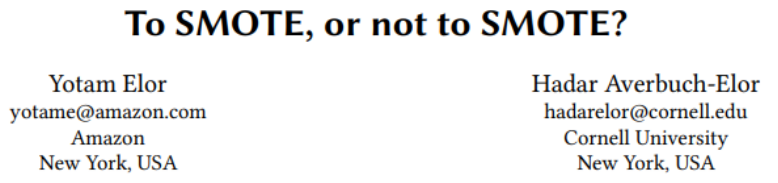
- F_1 *score*
- Área sob a curva característica de operação (ROC AUC)
- *Score* de Brier
- Gráfico de **calibração**

Introdução

Exemplos do "problema" de classes desbalanceadas:

- Classificar transação em fraudulenta ou legítima
- Detectar presença de doença rara
- Classificar *e-mail* como spam ou não

Falta de consenso na área de *imbalanced learning* quanto à reamostragem...



Dados desbalanceados

- Caso **binário** ou multiclasse
- Desbalanceamento **relativo** ou absoluto (n pequeno)
- A prevalência da classe positiva é baixa

$$\frac{\#\text{classe positiva}}{\#\text{observações}} < 10\%$$

- O classificador trivial tem acurácia alta
- A classe rara usualmente é a mais importante (falso positivo vs falso negativo)

Modelos de classificação

Um **classificador** leva um padrão $\boldsymbol{x}$ a uma classe $y \in \{0, 1\}$. Os dados são observações iid da forma $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^{n}$, usados para aproximar a verdadeira função de classificação $h(\cdot)$.

$$\begin{aligned} h : \mathcal{X} &\longrightarrow \mathcal{Y} \\ \boldsymbol{x} &\longmapsto y = h(\boldsymbol{x}) \end{aligned}$$

Modelos de classificação

Um **classificador** leva um padrão $\boldsymbol{x}$ a uma classe $y \in \{0, 1\}$. Os dados são observações iid da forma $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$, usados para aproximar a verdadeira função de classificação $h(\cdot)$.

$$h : \mathcal{X} \longrightarrow \mathcal{Y}$$
$$\boldsymbol{x} \longmapsto y = h(\boldsymbol{x})$$

Possivelmente o *output* é um *score* e não uma classificação.

$$h : \mathcal{X} \longrightarrow \mathbb{R}$$

Modelos de classificação

Um **classificador** leva um padrão $\boldsymbol{x}$ a uma classe $y \in \{0, 1\}$. Os dados são observações iid da forma $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$, usados para aproximar a verdadeira função de classificação $h(\cdot)$.

$$\begin{aligned} h : \mathcal{X} &\longrightarrow \mathcal{Y} \\ \boldsymbol{x} &\longmapsto y = h(\boldsymbol{x}) \end{aligned}$$

Possivelmente o *output* é um *score* e não uma classificação.

$$h : \mathcal{X} \longrightarrow \mathbb{R}$$

Ao definirmos uma regra de decisão obtemos uma classificação:

$$y_{\text{pred}} = \begin{cases} 1, & \text{se } h(\boldsymbol{x}) > \lambda, \\ 0, & \text{se } h(\boldsymbol{x}) \leq \lambda. \end{cases}$$

Modelos de classificação

Seja $y \in \{0, 1\}$ a resposta e $\boldsymbol{x} \in \mathbb{R}^K$, o **modelo de regressão logística** é dado por:

$$P(Y_i = 1|\boldsymbol{x}_i) = \frac{\exp\{\beta_0 + \sum_{k=1}^{K} \beta_k x_{ik}\}}{1 + \exp\{\beta_0 + \sum_{k=1}^{K} \beta_k x_{ik}\}} \tag{1}$$

O problema de otimização envolve maximizar a verossimilhança com respeito aos parâmetros $\boldsymbol{\beta}$

$$\ell(y; \boldsymbol{x}, \boldsymbol{\beta}) = \prod_{i=1}^{n} \left[P(Y_i = 1|\boldsymbol{x}_i, \boldsymbol{\beta})\right]^{y_i} \left[1 - P(Y_i = 1|\boldsymbol{x}_i, \boldsymbol{\beta})\right]^{1-y_i} \tag{2}$$

Modelos de classificação

Sejam $y \in \{-1, +1\}$, $\lambda \in \mathbb{R}$. Um classificador baseado em **support vector machines** (SVMs) pode ser definido por

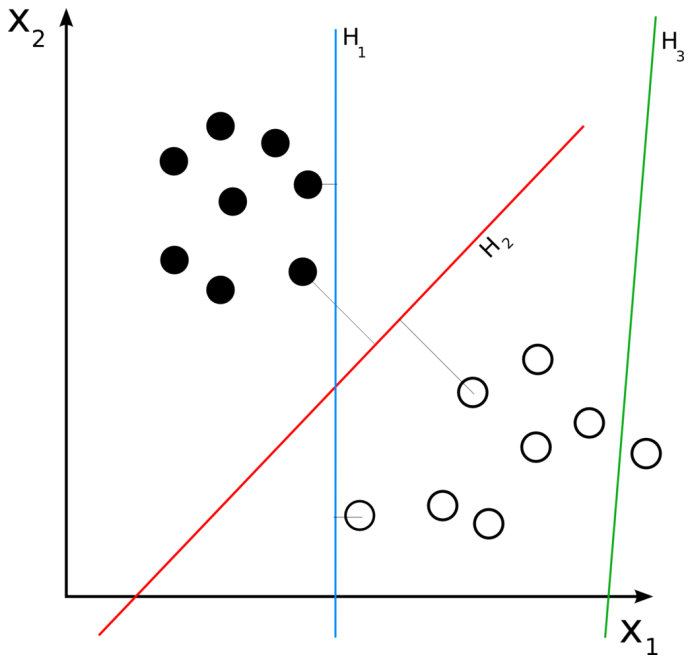
$$h(\boldsymbol{X}) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_K X_K \tag{3}$$

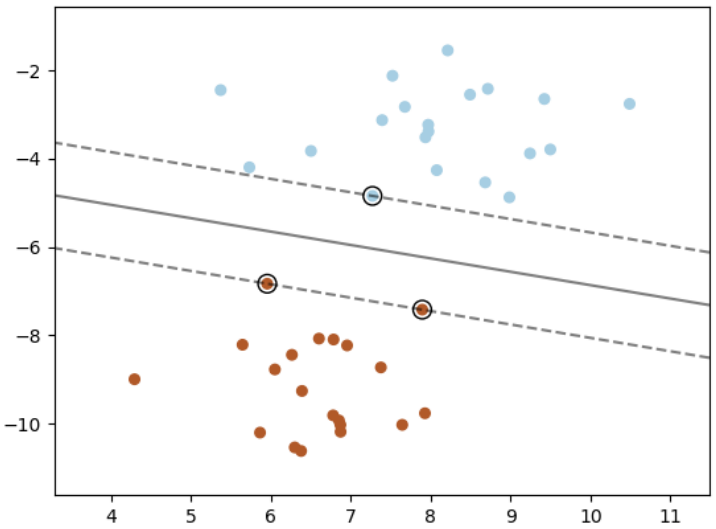
$$\underset{\beta_0, \beta_1, \ldots, \beta_K}{\text{minimize}} \quad \sum_{i=1}^{n} \max\{0, 1 - y_i h(\boldsymbol{x}_i)\} + \lambda \sum_{j=1}^{K} \beta_j^2 \tag{4}$$

- O termo da esquerda na expressão 4 é chamado *hinge-loss*
- O termo da direita é uma penalização *ridge*
- λ maior permite mais observações do lado errado da **margem**

Modelos de classificação

- Ambos os modelos envolvem a noção de **separabilidade linear**
- SVC procura o hiperplano de margem máxima
- Algumas instâncias podem ficar do lado errado (*soft margin*)





Métodos de reamostragem

São uma solução *data-level*, ou seja, obtemos diferentes conjuntos de dados a partir do original.

Estratégias de sobreamostragem:

- *Random oversampling* (ROS)
- *Synthetic minority oversampling technique* (SMOTE)
- *Borderline*-SMOTE1
- *Adaptive synthetic sampling approach* (ADASYN)

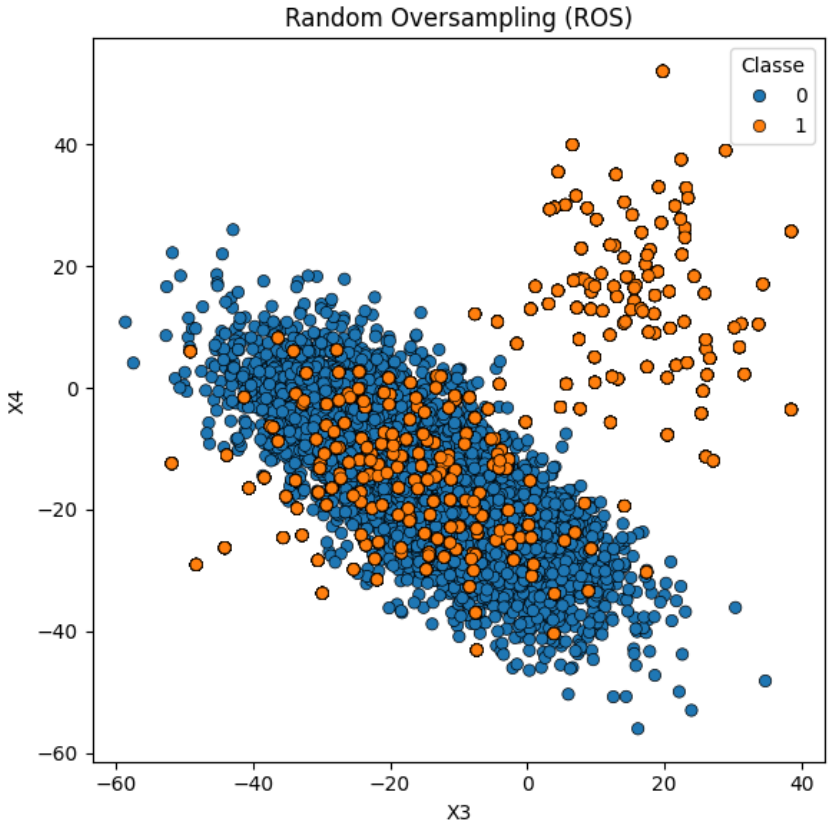
Estratégias de subamostragem:

- *Random undersampling* (RUS)
- *Edited Nearest Neighbors* (ENN)
- *Cluster Centroids*

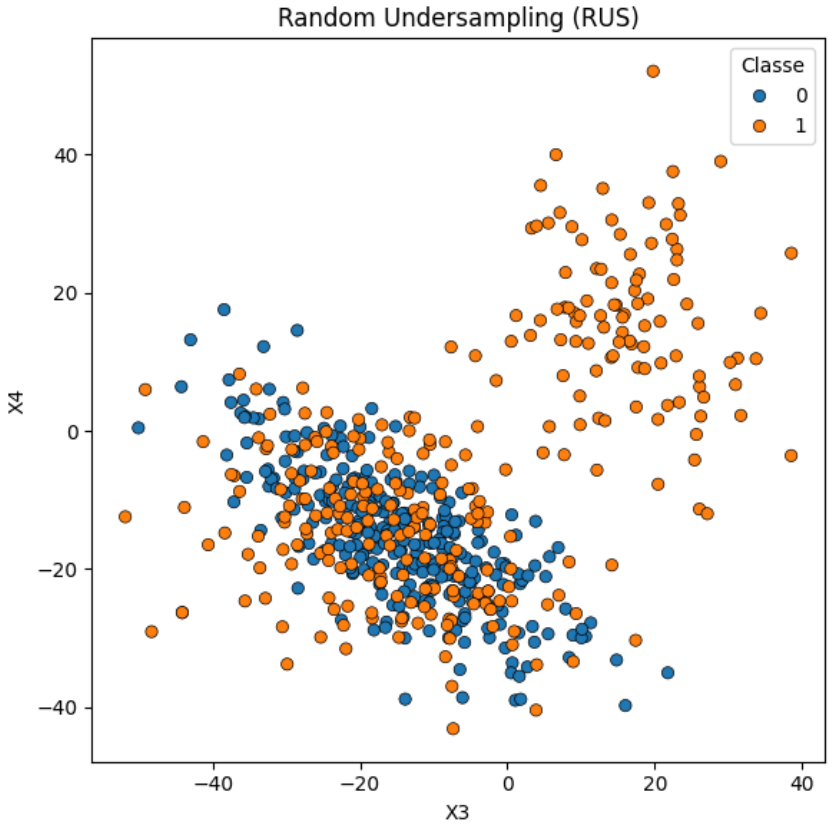
Estratégia mista utilizando SMOTE e ENN conjuntamente.

Oversampling e *undersampling*

- **ROS**: Aleatoriamente toma amostras com reposição da classe minoritária



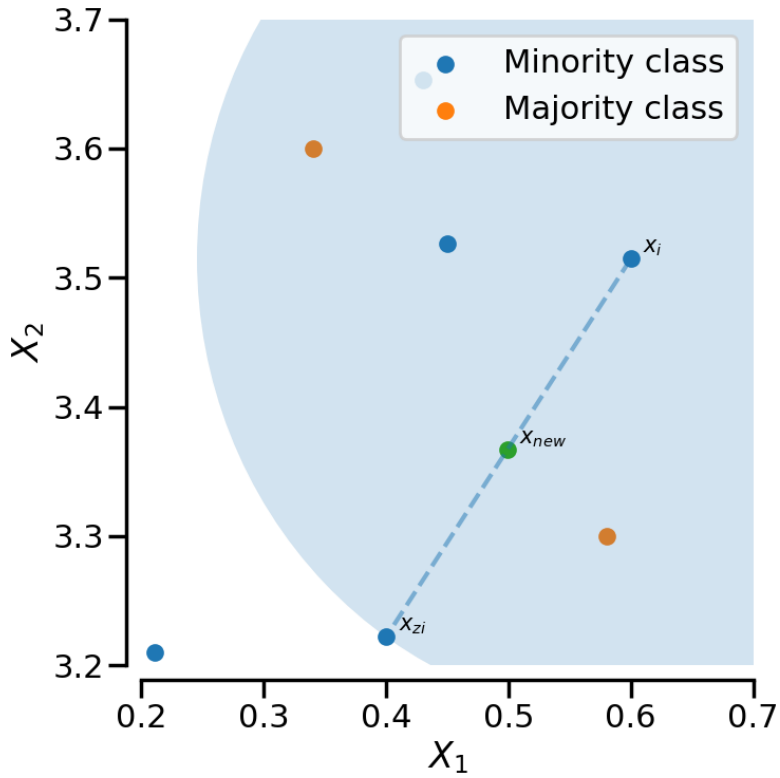
- **RUS**: Aleatoriamente remove observações da classe majoritária



SMOTE

Método de sobreamostragem baseado em k vizinhos mais próximos (kNN) que gera observações sintéticas.

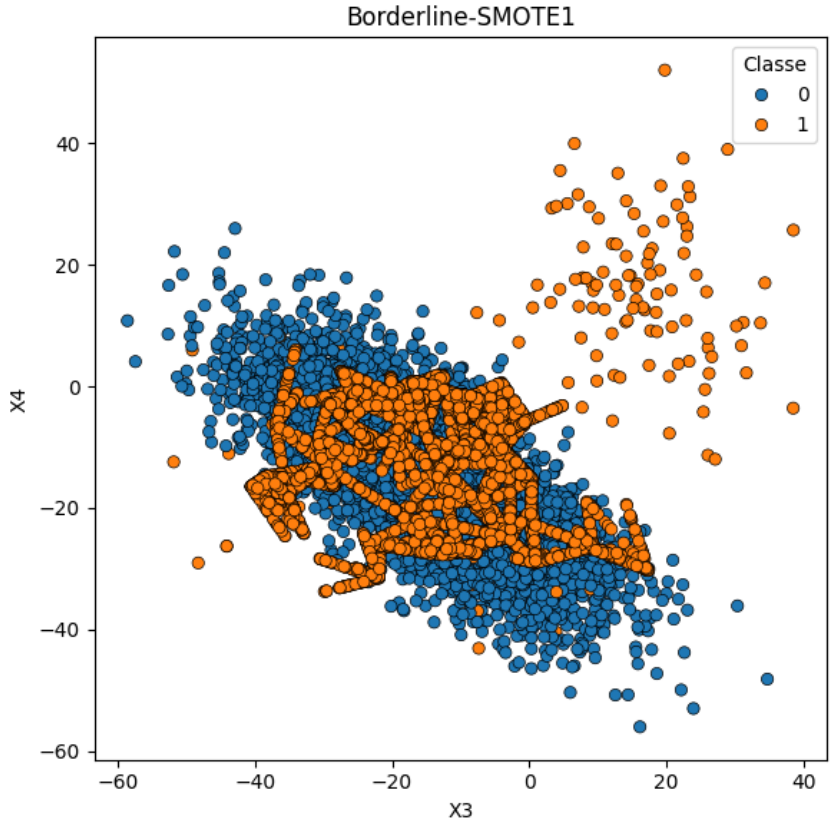
- Para cada instância minoritária $\boldsymbol{x}_i$ encontra-se os k vizinhos mais próximos
- Sorteia-se um deles ao acaso
- Gera-se uma nova obs. na reta que liga a instância original e o vizinho sorteado



Borderline-SMOTE1

SMOTE que prioriza instâncias da classe minoritária próximas ao limiar entre as classes e ignora instâncias ruído.

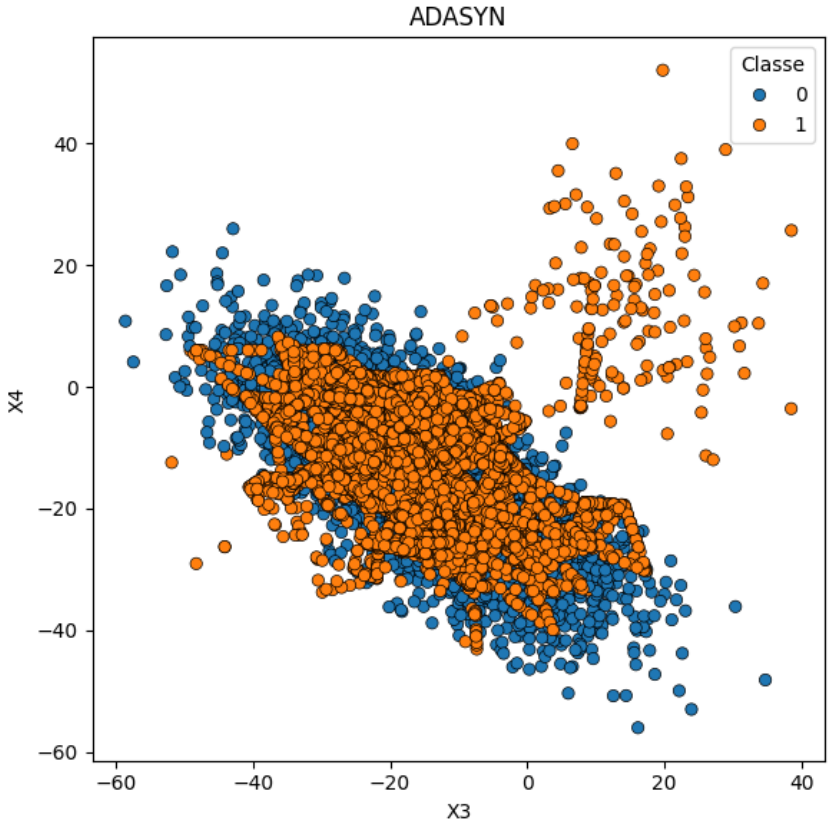
- Considera $\boldsymbol{x}_i$ ruído se todos k' vizinhos forem da classe oposta
- Considera $\boldsymbol{x}_i \in$ DANGER se 50% ou mais dos vizinhos for da classe oposta
- Reamostragem as instâncias do conjunto DANGER usando SMOTE



ADASYN

Mesma ideia do SMOTE e *Borderline*-SMOTE1, porém prioriza instâncias proporcionalmente à quantidade de vizinhos da classe oposta.

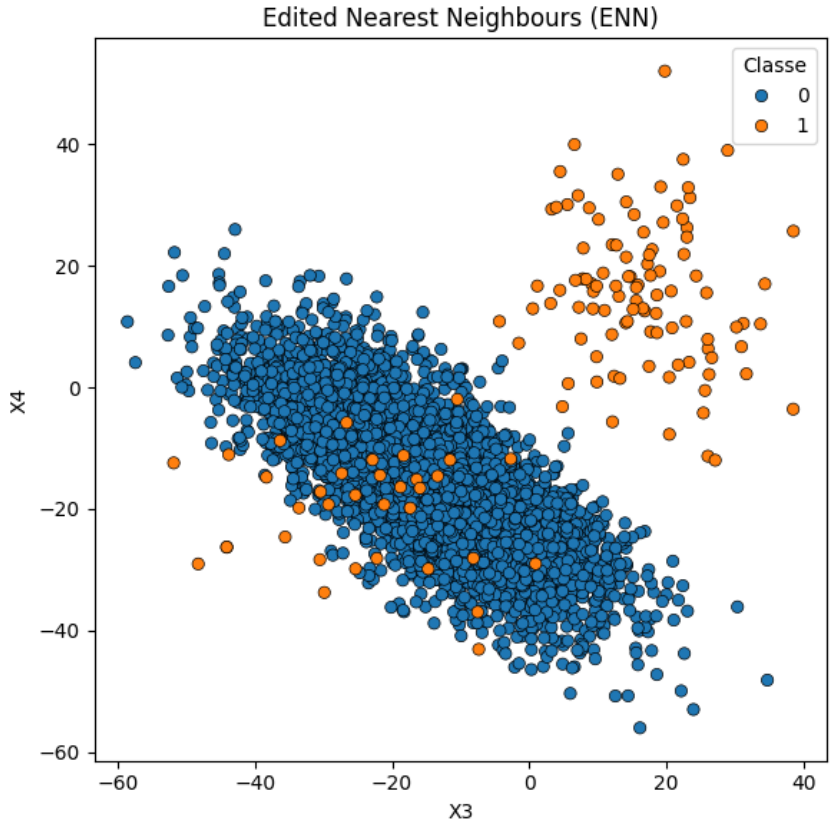
- Um ponto cuja vizinhança é da classe oposta é suposto mais importante de se aprender
- Para cada instância $\boldsymbol{x}_i$ encontra-se os k vizinhos e a proporção r_i destes que é da classe oposta
- Reamostragem por SMOTE proporcionalmente a r_i



ENN

Um ADASYN que ao invés de adicionar remove. A ideia é limpar o *dataset* deletando instâncias cuja vizinhança é da classe oposta.

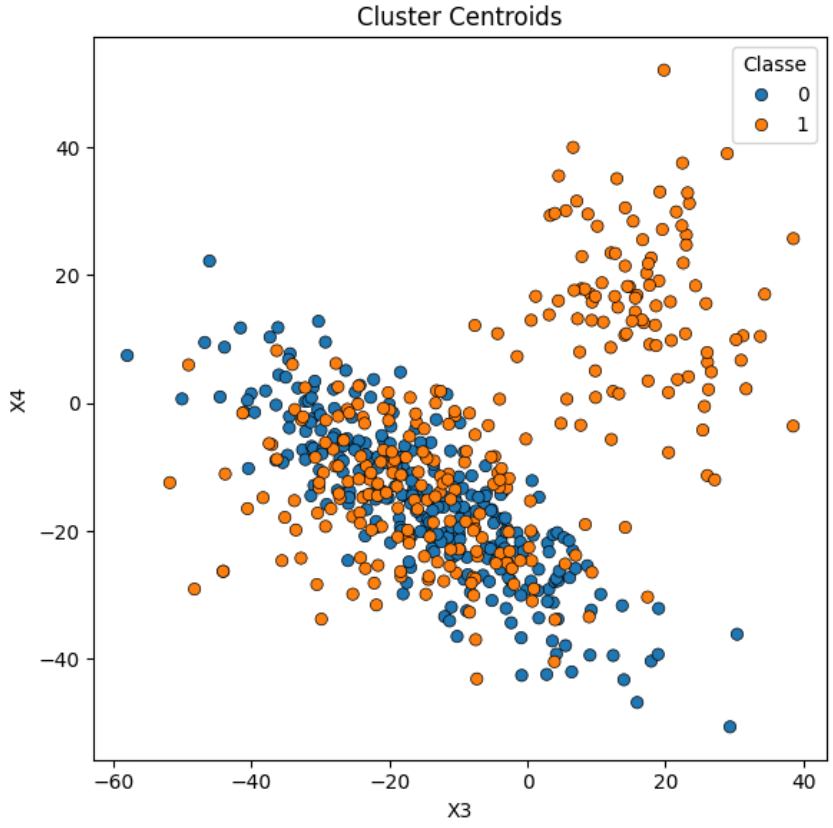
- Especifica-se quais classes considerar
- Especifica-se o percentual de vizinhos que precisa ser da classe oposta para que se remova a instância
- Para cada instância $\boldsymbol{x}_i$ na classe (ou classes) definida encontra-se os k vizinhos
- Deleta-se a instância de acordo com as especificações



Cluster Centroids

Baseado em *k-means clustering*. Substitui instâncias da classe majoritária pelo centroide do seu *cluster* obtido por *k-means*.

- Inicialmente cria-se os centroides aleatoriamente
- Para cada instância obtém-se o conglomerado com centroide mais próximo e considera a instância como pertencente a esse conglomerado



Avaliação de desempenho

A partir da matriz de confusão é possível obter várias métricas.

	Predito +1	**Predito -1**
Observado +1	TP	FN
Observado -1	FP	TN

- A precisão é $\frac{TP}{\text{TP}+\text{FP}}$
- A taxa de verdadeiros positivos (TPR) ou *recall* é $\frac{\text{TP}}{\text{TP}+\text{FN}}$
- A taxa de falsos positivos (FPR) é $\frac{\text{FP}}{\text{FP}+\text{TN}}$
- A curva ROC é um gráfico de FPR vs TPR

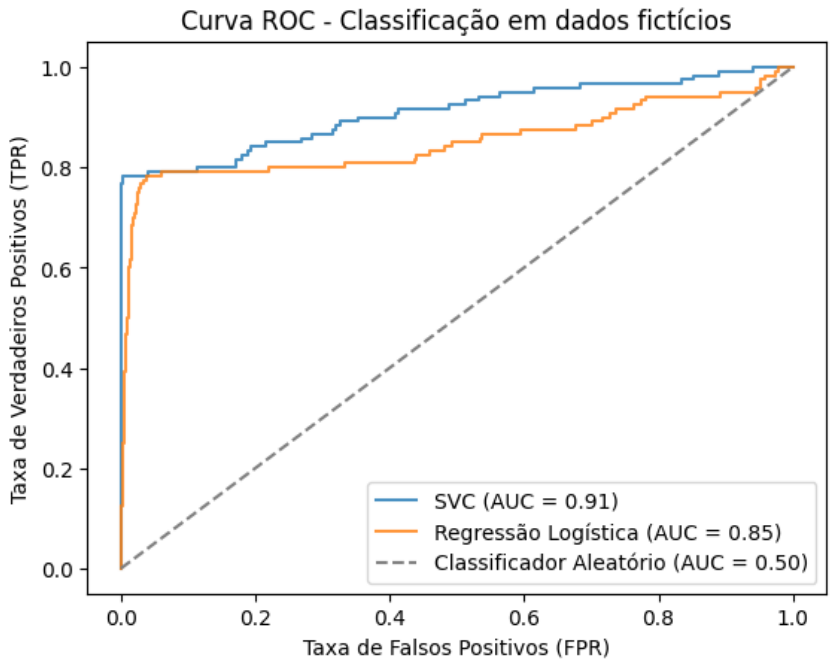
O *score* F_1 é uma média harmônica da precisão e do recall:

$$F_1 = 2\frac{PR}{P+R}$$

Avaliação de desempenho

Seja $h(\cdot)$ um classificador tal que $y_{\text{pred}} = \begin{cases} 1 & \text{se } h(\boldsymbol{x}) \geq \lambda, \\ 0 & \text{se } h(\boldsymbol{x}) < \lambda. \end{cases}$

- A curva ROC é definida por $\lambda \mapsto (\text{FPR}(\lambda), \text{TPR}(\lambda))$, ao variarmos o limiar λ no intervalo $(-\infty, +\infty)$
- ROC AUC nada mais é que a área sob essa curva
- $AUC = \mathbb{P}(h(\boldsymbol{x}^1) \geq h(\boldsymbol{x}^0))$



Avaliação de desempenho

Calibração é uma medida de concordância das probabilidades estimadas com as frequências observadas. Matematicamente, um classificador $h : \mathcal{X} \longrightarrow [0, 1]$ é dito calibrado se, para qualquer $p \in [0, 1]$, vale que

$$P(Y = 1 \mid h(\boldsymbol{X}) = p) = p$$

- Um meteorologista é calibrado se chover em 30% das vezes que a previsão de chuva for 30%
- Um classificador é calibrado se quantifica corretamente a incerteza nas estimativas

Score de Brier

É uma medida que avalia a discriminação e calibração de um modelo. É um erro quadrático médio, podendo ser escrito como

$$B(\boldsymbol{y}, \hat{\boldsymbol{p}}) = n^{-1} \sum_{i=1}^{n} (y_i - \hat{p}_i)^2$$

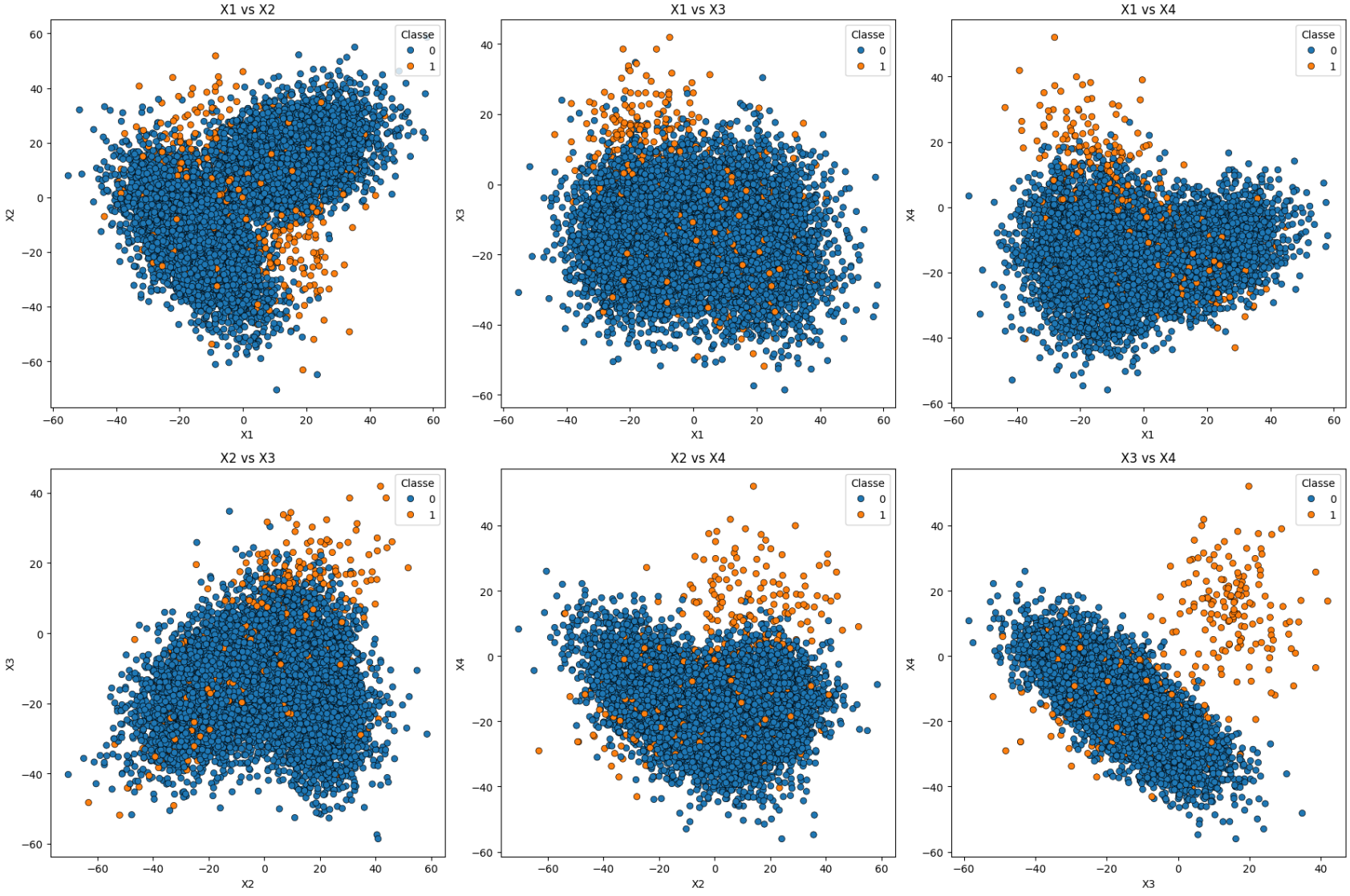
onde $\boldsymbol{y}$ é um vetor de realizações de variáveis aleatórias $Y_i \sim Ber(\pi_i)$, e $\hat{\boldsymbol{p}}$ um vetor de probabilidades estimadas.

Dados simulados

- Os dados foram gerados utilizando a biblioteca *scikit-learn*
- 10k observações no total
- 414 observações da classe positiva

```
X, y = make_classification(n_samples=10000,
    n_features=4,
    n_informative=4,
    n_redundant=0,
    n_repeated=0,
    weights=(0.97, 0.03), # 3% classe minoritaria
    flip_y=0.02, # 2% ruido inversao de classe
    class_sep=1.5,
    scale=10,
    random_state=SEED) # SEED = 42
```

Dados simulados



Distribuição após reamostragem

- Conjunto de treinamento com 7000 observações
- Nem todos métodos buscam balanceamento exato

Reamostragem	**#Negativa**	**#Positiva**	**Proporção**
Nenhuma	6710	290	4.14%
ROS	6710	6710	50%
SMOTE	6710	6710	50%
Borderline-SMOTE1	6710	6710	50%
ADASYN	6710	6768	50.22%
RUS	290	290	50%
Cluster Centroids	290	290	50%
ENN	6371	135	2.08%
SMOTEENN	5516	6236	46.94%

Desempenho nos dados de treinamento

Estimativas usando 5-*fold* CV com estratificação:

	Regressão Logística		
Estratégia	F1 *Score*	AUC	Brier
Sem reamostragem	0.437 (0.037)	0.641 (0.014)	0.028 (0.001)
ROS	0.175 (0.007)	0.704 (0.014)	0.177 (0.001)
SMOTE	0.180 (0.007)	0.712 (0.013)	0.176 (0.002)
Borderline-SMOTE1	0.168 (0.016)	0.686 (0.031)	0.175 (0.004)
ADASYN	0.127 (0.009)	0.645 (0.026)	0.231 (0.002)
RUS	0.178 (0.009)	0.700 (0.014)	0.175 (0.002)
Cluster Centroids	0.173 (0.010)	0.687 (0.017)	0.174 (0.0009)
ENN	0.437 (0.037)	0.641 (0.014)	0.029 (0.001)
SMOTEENN	0.177 (0.006)	0.726 (0.013)	0.186 (0.002)

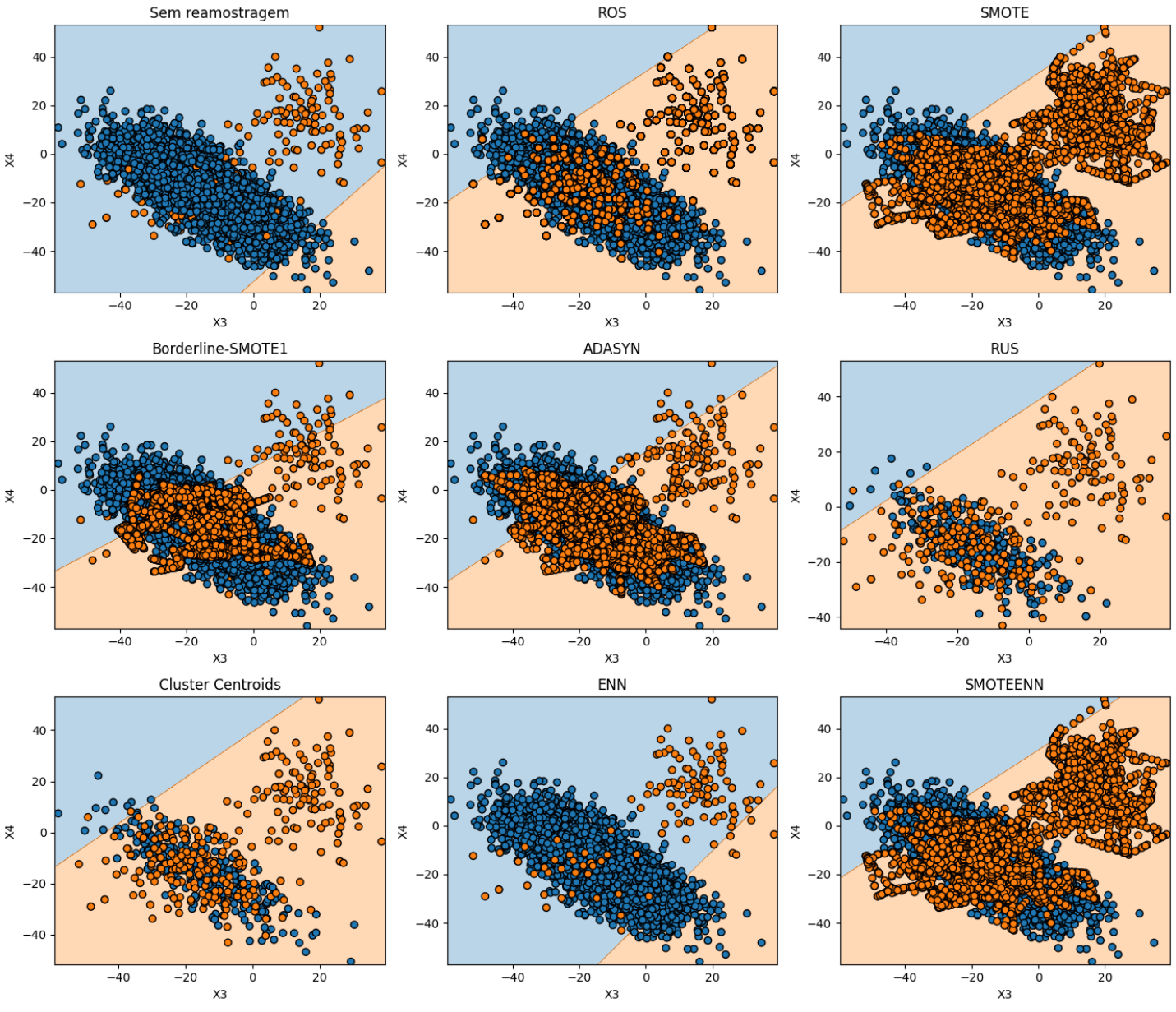
- Piora no *score* F_1 (exceto ENN)
- Pequena melhora na AUC
- Piora no *score* de Brier

Desempenho nos dados de treinamento

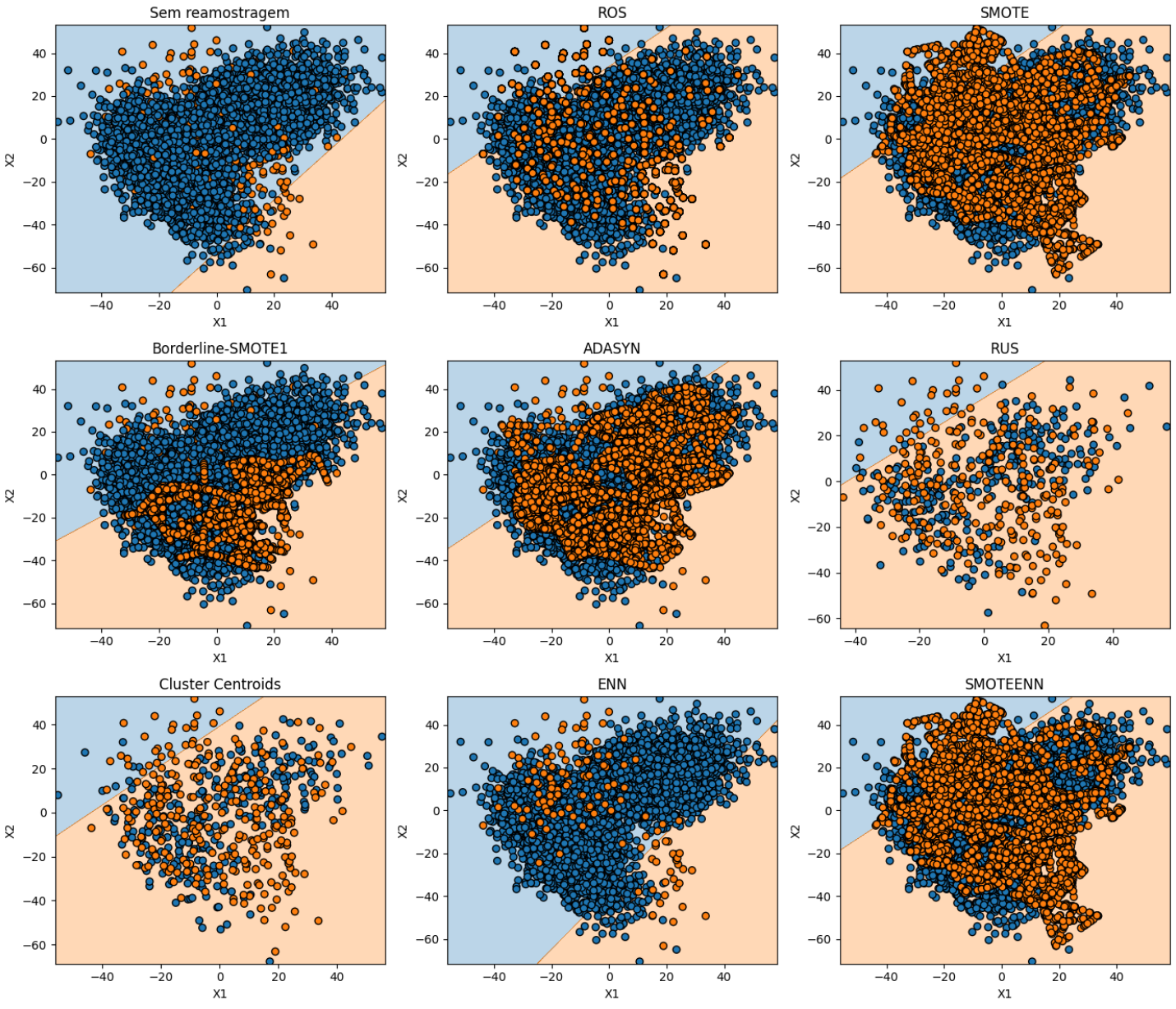
	Support Vector Classifier		
Estratégia	F1 *Score*	AUC	Brier
Sem reamostragem	0.599 (0.043)	0.717 (0.021)	0.020 (0.001)
ROS	0.530 (0.014)	0.813 (0.009)	0.088 (0.0006)
SMOTE	0.527 (0.022)	0.809 (0.010)	0.088 (0.0008)
Borderline-SMOTE1	0.416 (0.018)	0.796 (0.009)	0.070 (0.003)
ADASYN	0.227 (0.009)	0.759 (0.005)	0.171 (0.003)
RUS	0.484 (0.026)	0.810 (0.009)	0.108 (0.003)
Cluster Centroids	0.492 (0.015)	0.809 (0.010)	0.106 (0.002)
ENN	0.610 (0.036)	0.722 (0.018)	0.020 (0.001)
SMOTEENN	0.430 (0.027)	0.801 (0.010)	0.100 (0.003)

- Pouca mudança no *score* F_1
- Melhora na AUC
- Piora no *score* de Brier

Impacto da reamostragem na região de decisão



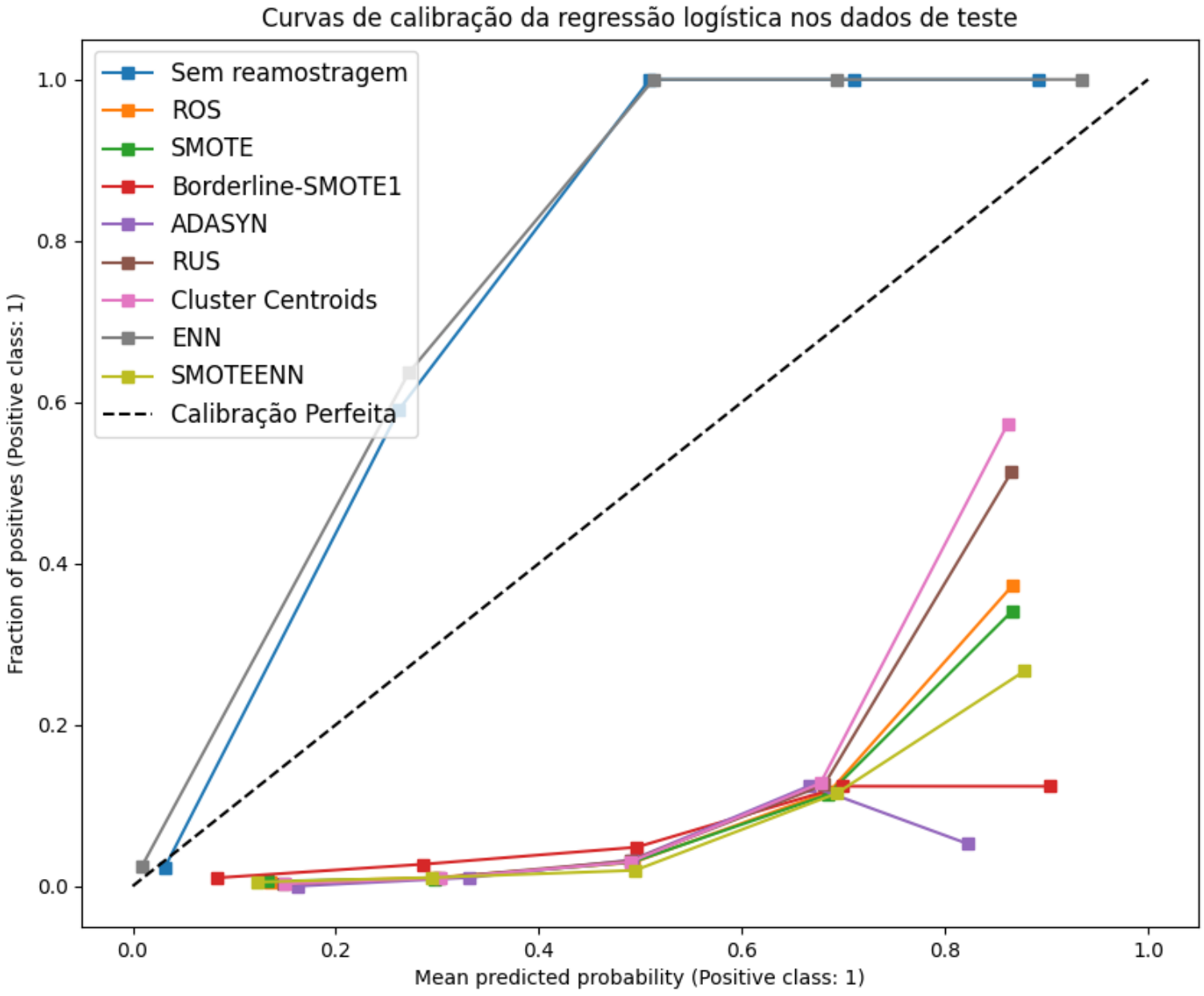
Impacto da reamostragem na região de decisão



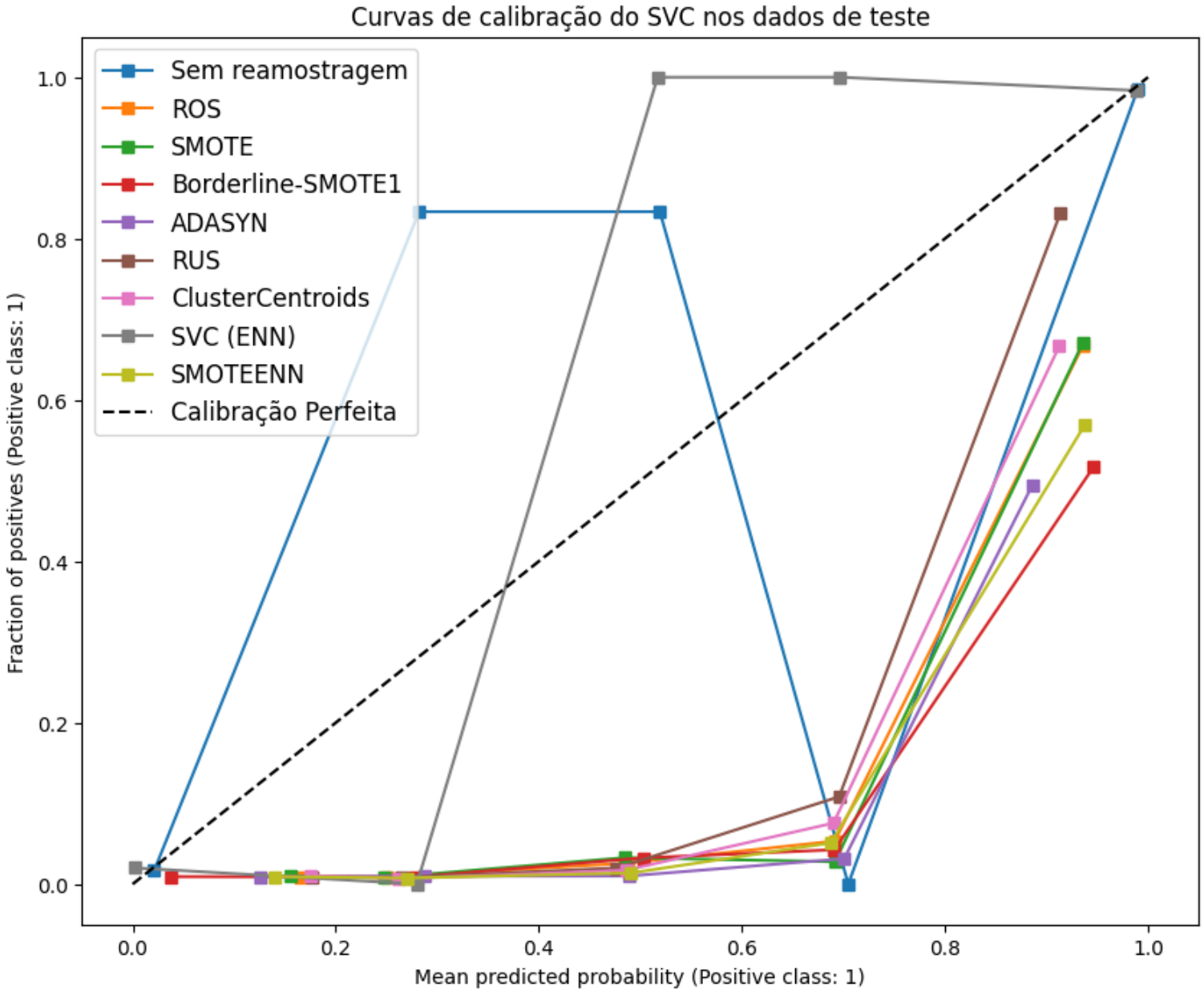
Desempenho nos dados de teste

Modelo	**Estratégia**	**Métricas**				
		F_1 *Score*	AUC	Brier	Precisão	*Recall*
Regressão Logística	Sem reamostragem	0.469	0.814	0.027	1.000	0.306
	ROS	0.213	0.857	0.176	0.122	0.831
	SMOTE	0.212	0.857	0.176	0.121	0.831
	Borderline-SMOTE1	0.180	0.776	0.176	0.104	0.669
	ADASYN	0.139	0.744	0.235	0.077	0.750
	RUS	0.220	0.854	0.174	0.127	0.815
	Cluster Centroids	0.214	0.854	0.175	0.124	0.790
	ENN	0.469	0.817	0.027	1.000	0.306
	SMOTEENN	0.204	0.856	0.186	0.116	0.847
SVC	Sem reamostragem	0.649	0.861	0.018	0.984	0.484
	ROS	0.595	0.893	0.083	0.480	0.782
	SMOTE	0.615	0.886	0.083	0.511	0.774
	Borderline-SMOTE1	0.454	0.889	0.065	0.320	0.782
	ADASYN	**0.265**	0.881	0.173	0.159	**0.815**
	RUS	0.574	0.893	0.097	0.453	0.782
	Cluster Centroids	0.574	**0.900**	0.102	0.448	0.798
	ENN	0.656	0.861	0.017	0.984	0.492
	SMOTEENN	0.544	0.886	0.089	0.419	0.774

Calibração da regressão logística



Calibração do SVC



Conclusão

- Alinhado com Goorbergh et al. (2022), há uma piora significativa na calibração dos modelos sob reamostragem
- As probabilidades estimadas são excessivamente altas
- F_1 *score* piorou, por mais que o *recall* tenha melhorado, a precisão piorou muito
- ROC AUC ligeiramente melhor
- Reamostragem modifica a região de decisão

Conclusão

- O contexto importa
- Definir a métrica a ser otimizada
- Considerar a otimização do limiar de decisão
- Considerar algoritmos *cost-sensitive*

Limitações

- "Graus de liberdade do pesquisador": hiperparâmetros, métricas, distribuição dos dados etc
- Não utilização de teste estatístico para verificar calibração (ex. *Spiegelhalter Z-statistic*)
- Não foram considerados métodos de *ensemble* (*Bagging*, *Boosting* etc)

Referências

- EuroSciPy 2023 - Get the best from your scikit-learn classifier
- He, H. e Ma, Y. (2013), Imbalanced Learning: Foundations, Algorithms, and Applications.
- Izbicki, R. e Santos, T. M. dos. (2020), Aprendizado de máquina: uma abordagem estatística.
- van den Goorbergh, R. et al. (2022), The harm of class imbalance corrections for risk prediction models: illustration and simulation using logistic regression.
- Filho, T. S. et al. (2023), Classifer calibration: a survey on how to assess and improve predicted class probabilities.
- Tarawneh, A. S. et al. (2022), Stop Oversampling for Class Imbalance Learning: A Review.