

# Destilacao de Conhecimento para Economia: Negociando Complexidade por Interpretabilidade em Modelos Econometricos

Gustavo Coelho Haase  
gustavohaase@gmail.com  
Banco do Brasil S.A  
Brasilia, Brazil

Paulo Henrique Dourado da Silva  
paulodourado.unb@gmail.com  
Banco do Brasil S.A  
Brasilia, Brazil

## RESUMO

Economistas e formuladores de politicas publicas enfrentam um dilema fundamental: modelos de machine learning complexos (ensembles, redes neurais) alcancam alta acuracia preditiva, mas carecem da interpretabilidade economica essencial para analise de politicas, enquanto modelos econometricos tradicionais (regressao linear, logit) sao interpretaveis mas limitados em poder preditivo. Apresentamos framework de **destilacao de conhecimento econometrica** que transfere conhecimento de modelos complexos (teacher) para modelos interpretaveis (student GAM/Linear), preservando simultaneamente: (1) **intuicao economica** (coeficientes, efeitos marginais), (2) **restricoes economicas** (monotonia, consistencia de sinais), e (3) **estabilidade de coeficientes** (inferencia estatistica valida). Nossa implementacao no DeepBridge permite destilar XGBoost/Neural Networks para GAMs/Linear com **perda de acuracia de apenas 2-5%**, enquanto produz coeficientes estaveis sob bootstrap ( $CV < 0.15$ ), preserva relacoes economicas ( $income \uparrow \rightarrow default \downarrow$ ), e permite analise causal valida. Validacao em tres dominios economicos (risco de credito, economia do trabalho, economia da saude) demonstra: (1) coeficientes do modelo destilado convergem com teoria economica, (2) efeitos marginais sao monotonicos e interpretaveis, (3) **quebras estruturais** (pre/pos-2008) sao detectadas e interpretadas economicamente. Framework preenche lacuna critica entre ML de alta performance e rigor econometrico.

## KEYWORDS

Knowledge Distillation, Econometrics, Interpretability, GAM, Economic Theory, Policy Analysis

## 1 INTRODUCAO

A aplicacao de machine learning em economia enfrenta tensao fundamental entre poder preditivo e interpretabilidade economica. Modelos complexos (gradient boosting, redes neurais) alcancam acuracia superior mas produzem “caixas-pretas” inadequadas para analise de politicas publicas, inferencia causal, e validacao teorica. Modelos econometricos tradicionais (regressao linear, logit, GAM) oferecem coeficientes interpretaveis e fundamentacao estatistica, mas limitacoes em capacidade de capturar relacoes nao-lineares complexas.

### 1.1 Motivacao

Economistas e formuladores de politicas requerem modelos que simultaneamente:

- **Interpretacao economica:** Coeficientes representam efeitos marginais, elasticidades, ou relacoes causais interpretaveis

- **Conformidade teorica:** Modelos respeitam restricoes economicas (monotonia de funcoes de utilidade, lei da demanda)
- **Auditabilidade:** Nao-especialistas em ML (reguladores, policy makers) podem validar premissas e resultados
- **Inferencia estatistica:** Intervalos de confianca, testes de hipotese, e estabilidade de coeficientes permitem conclusoes rigorosas
- **Alta acuracia:** Decisoes economicas de alto impacto (politica monetaria, regulacao financeira) exigem predicoes precisas

Aplicacoes criticas incluem:

- (1) **Risco de credito:** Reguladores exigem coeficientes interpretaveis (Basel III), mas bancos querem acuracia maxima
- (2) **Economia do trabalho:** Analise de impacto de salario minimo requer efeitos marginais validos, nao apenas predicoes
- (3) **Saude publica:** Politicas de intervencao baseiam-se em relacoes causais, nao correlacoes de caixa-preta

### 1.2 Problema

Pesquisa em knowledge distillation ignora requisitos especificos de economia:

- (1) **Perda de interpretacao economica:** Destilacao tradicional otimiza apenas acuracia—coeficientes do modelo student podem violar teoria economica
- (2) **Instabilidade de coeficientes:** Modelos destilados nao garantem estabilidade necessaria para inferencia estatistica (bootstrap, cross-validation)
- (3) **Violacao de restricoes:** Modelos student podem apresentar relacoes contra-intuitivas (e.g.,  $income \uparrow \rightarrow default \uparrow$ )
- (4) **Ausencia de validacao causal:** Frameworks existentes nao verificam se destilacao preserva estruturas causais
- (5) **Detectao de quebras estruturais:** Mudancas em relacoes economicas (e.g., crise 2008) nao sao identificadas ou interpretadas

### 1.3 Nossa Solucao

Apresentamos framework de **destilacao de conhecimento econometrica** que:

- **Preserva intuicao economica:** Destilacao para GAM/Linear mantendo coeficientes e efeitos marginais interpretaveis
- **Garante restricoes economicas:** Constraints de monotonia, consistencia de sinais, e conformidade teorica durante destilacao

- **Valida estabilidade:** Bootstrap resampling demonstra que coeficientes são estáveis ( $CV < 0.15$ )
- **Detecta quebras estruturais:** Identifica mudanças em relações econômicas e mantém interpretabilidade
- **Suporta inferência causal:** Framework compatível com instrumental variables, diff-in-diff

## 1.4 Contribuições

- (1) **Framework de destilação econometrítica:** Primeira metodologia que combina knowledge distillation com rigor econometrítico
- (2) **Preservação de restrições econômicas:** Técnicas de destilação com constraints (monotonia, sinais, marginal effects)
- (3) **Análise de estabilidade de coeficientes:** Metodologia bootstrap demonstrando confiabilidade para policy analysis
- (4) **Detecta de quebras estruturais:** Identificação automática de mudanças em relações econômicas
- (5) **Validação empírica:** Case studies em crédito, trabalho, e saúde demonstrando aplicabilidade prática
- (6) **Implementação prática:** Framework integrado ao DeepBridge para uso em produção

## 1.5 Resultados Principais

Validação em três domínios econômicos demonstra:

- **Trade-off acurácia-interpretabilidade:** Perda de 2-5% em acurácia vs. modelo teacher complexo
- **Estabilidade de coeficientes:**  $CV < 0.15$  para coeficientes principais sob bootstrap (10,000 amostras)
- **Conformidade econômica:** 95%+ das restrições de sinais e monotonia preservadas
- **Detecta de quebras:** Identificação precisa de mudanças estruturais pre/pos-2008 em crédito
- **Comparação com baselines:** Superioridade vs. linear regression direta (sem destilação) em acurácia (+8-12%)

## 1.6 Impacto Esperado

1.6.1 *Para Economistas.* - Modelos com acurácia próxima a ML de ponta, mas com interpretabilidade de econometria clássica - Coeficientes estáveis permitindo inferência estatística rigorosa - Validação automática de conformidade com teoria econômica

1.6.2 *Para Formuladores de Políticas.* - Evidência quantitativa interpretável para decisões de política pública - Transparência total (auditabilidade por não-especialistas) - Análise de efeitos marginais e elasticidades confiáveis

1.6.3 *Para Indústria Financeira.* - Conformidade regulatória (coeficientes interpretáveis para Basel III, IFRS 9) - Poder preditivo superior a modelos lineares tradicionais - Capacidade de explicar decisões de crédito para reguladores

## 1.7 Organização

Seção 2 apresenta fundamentação em econometria e knowledge distillation. Seção 3 descreve design do framework de destilação econometrítica. Seção 4 detalha implementação no DeepBridge. Seção 5 apresenta case studies em crédito, trabalho, e saúde. Seção

6 discute limitações e implicações teóricas. Seção 7 conclui com direções futuras.

## 2 FUNDAMENTAÇÃO E TRABALHOS RELACIONADOS

### 2.1 Econometria e Interpretabilidade

2.1.1 *Modelos Econometricos Clássicos.* Economia tradicionalmente utiliza modelos com interpretação clara:

- **Regressão Linear:**  $y = \beta_0 + \sum_{i=1}^p \beta_i x_i + \epsilon$ 
  - Coeficientes  $\beta_i$  representam efeitos marginais
  - Inferência via intervalos de confiança, testes t
  - Limitação: Apenas relações lineares
- **Logit/Probit:** Para variáveis dependentes binárias
  - Log-odds ratios interpretáveis
  - Efeitos marginais calculáveis
  - Limitação: Forma funcional rígida
- **Generalized Additive Models (GAM):**  $g(E[y]) = \beta_0 + \sum_{i=1}^p f_i(x_i)$ 
  - Flexibilidade para não-linearidades via splines
  - Funções  $f_i$  individualmente interpretáveis
  - Preserva aditividade (interpretação de efeitos parciais)

2.1.2 *Restrições Econômicas.* Teoria econômica impõe restrições que modelos devem respeitar:

- (1) **Monotonia:** Funções de utilidade são não-decrescentes em consumo
- (2) **Lei da Demanda:** Preço  $\uparrow \rightarrow$  Quantidade demandada  $\downarrow$
- (3) **Restrições de Sinais:** Income  $\uparrow \rightarrow$  Default probability  $\downarrow$
- (4) **Homogeneidade:** Funções de produção apresentam retornos de escala específicos

Violacão destas restrições invalida interpretação econômica.

### 2.2 Knowledge Distillation

2.2.1 *Framework Clássico.* Hinton et al. (2015) introduziram destilação de conhecimento:

$$\mathcal{L}_{KD} = \alpha \mathcal{L}_{\text{soft}} + (1 - \alpha) \mathcal{L}_{\text{hard}} \quad (1)$$

onde:

- $\mathcal{L}_{\text{soft}}$ : KL divergence entre probabilidades teacher (temperatura  $T$ ) e student
- $\mathcal{L}_{\text{hard}}$ : Cross-entropy com labels verdadeiros
- $\alpha$ : Peso balanceando soft vs. hard labels

**Limitação:** Foco exclusivo em acurácia preditiva, ignorando interpretabilidade.

**Tabela 1: Abordagens de Knowledge Distillation**

Abordagem	Característica	Aplicação
Response-based	Soft labels nas saídas	Classificação
Feature-based	Camadas intermediárias	Vision, NLP
Relation-based	Relações entre exemplos	Metric learning
<b>Ours: Econometric</b>	<b>Restrições econômicas</b>	Economia

2.2.2 *Variantes de Distilação.*

## 2.3 ML Interpretavel em Economia

### 2.3.1 Trabalhos em Interpretabilidade Economica.

- **Mullainathan & Spiess (2017):** “Machine Learning: An Applied Econometric Approach”
  - Discutem trade-off predicao vs. inferencia causal
  - Nao propoe metodologia de reconciliacao
- **Athey & Imbens (2019):** “Machine Learning Methods Economists Should Know About”
  - Revisao de metodos ML para economia
  - Foco em causal inference, nao destilacao
- **Lundberg et al. (2020):** “From Local Explanations to Global Understanding with Explainable AI”
  - SHAP values para interpretacao
  - Limitacao: Explicacoes post-hoc, nao modelo intrinsecamente interpretavel

2.3.2 *Gap na Literatura. Nenhum trabalho anterior combina:*

- (1) Knowledge distillation de modelos complexos
- (2) Preservacao de restricoes economicas
- (3) Garantia de estabilidade de coeficientes
- (4) Validacao em dominios economicos reais

## 2.4 Estabilidade de Coeficientes

### 2.4.1 Importancia em Econometria.

Policy analysis requer coeficientes estaveis:

- **Inferencia estatistica:** Intervalos de confianca validos exigem estimativas nao-volteis
- **Reproducibilidade:** Resultados devem ser replicaveis em amostras independentes
- **Robustez:** Conclusoes nao podem depender de particularidades da amostra

### 2.4.2 Metricas de Estabilidade.

$$CV(\beta_i) = \frac{\sigma(\hat{\beta}_i^{(1)}, \dots, \hat{\beta}_i^{(B)})}{\mu(\hat{\beta}_i^{(1)}, \dots, \hat{\beta}_i^{(B)})} \quad (2)$$

onde  $\hat{\beta}_i^{(b)}$  e estimativa de  $\beta_i$  em bootstrap sample  $b$ .

**Criterio:**  $CV < 0.15$  indica estabilidade aceitavel para policy analysis.

## 2.5 Quebras Estruturais

### 2.5.1 Conceito Economico.

Quebras estruturais ocorrem quando relacoes economicas fundamentais mudam:

- **Crise Financeira 2008:** Relacao income-default probability mudou drasticamente
- **Mudancas Regulatorias:** Novas leis alteram comportamento de agentes economicos
- **Choques Tecnologicos:** Automacao altera funcoes de producao

### 2.5.2 Testes Tradicionais.

- **Chow Test:** Testa igualdade de coeficientes entre periodos
- **CUSUM:** Detecta mudancas em resíduos cumulativos
- **Limitacao:** Requerem especificacao de ponto de quebra a priori

**Nossa Abordagem:** Detecção automatica via analise de coeficientes destilados em janelas temporais.

## 2.6 Trabalhos Relacionados em ML Interpretavel

Tabela 2: Comparacao com Ferramentas de Interpretabilidade

Ferramenta	Intrinseco	Restricoes Econ.	Estabilidade	Destilacao
LIME	✗	✗	✗	✗
SHAP	✗	✗	✗	✗
InterpretML	✓	✗	✗	✗
EconML	✓	Parcial	✓	✗
Ours	✓	✓	✓	✓

## 2.7 Posicionamento da Contribuicao

Nossa abordagem preenche lacuna fundamental:

- vs. **KD classico:** Adiciona restricoes economicas e validacao de estabilidade
- vs. **Econometria tradicional:** Alcanca acuracia superior via destilacao de modelos complexos
- vs. **Explainable AI:** Produz modelos intrinsecamente interpretaveis, nao explicacoes post-hoc
- vs. **EconML:** Foca em destilacao para interpretabilidade, nao apenas causal inference

## 3 DESIGN DO FRAMEWORK

### 3.1 Visao Geral

O framework de destilacao econometrica consiste em cinco componentes principais:

- (1) **Teacher Training:** Treinamento de modelo complexo de alta acuracia (XGBoost, Neural Network)
- (2) **Economic Constraint Encoder:** Codificacao de restricoes economicas (monotonia, sinais)
- (3) **Constrained Distillation Engine:** Destilacao para GAM/Linear preservando restricoes
- (4) **Coefficient Stability Analyzer:** Validacao de estabilidade via bootstrap
- (5) **Structural Break Detector:** Identificacao de mudancas em relacoes economicas

### 3.2 Componente 1: Teacher Training

3.2.1 *Modelos Teacher Suportados.* Framework aceita modelos complexos pre-treinados:

- **Gradient Boosting:** XGBoost, LightGBM, CatBoost
- **Random Forests:** Ensembles de arvores de decisao
- **Neural Networks:** Arquiteturas totalmente conectadas
- **Ensemble Hybrids:** Combinacoes de multiplos modelos

**Requisito:** Modelo teacher deve fornecer probabilidades calibradas.

3.2.2 *Justificativa para Complexidade.* Teacher models capturam:

- Interacões de alta ordem entre features
- Não-linearidades complexas
- Patterns sutis em dados de alta dimensão

### 3.3 Componente 2: Economic Constraint Encoder

#### 3.3.1 Tipos de Restrições.

- (1) **Sign Constraints:** Coeficientes/efeitos marginais devem ter sinal específico

$$\text{sign}(\frac{\partial \hat{y}}{\partial x_i}) = s_i \quad \text{onde } s_i \in \{-1, +1\} \quad (3)$$

- (2) **Monotonicity Constraints:** Funções GAM monotonamente crescentes/decrescentes

$$f'_i(x) \geq 0 \quad \forall x \in \text{domain}(x_i) \quad (\text{monotonia crescente}) \quad (4)$$

- (3) **Magnitude Bounds:** Limites superior/inferior para efeitos

$$L_i \leq \beta_i \leq U_i \quad (5)$$

- (4) **Interaction Constraints:** Restrições sobre termos de interacão

3.3.2 *Especificação de Restrições.* Economista especifica constraints via API declarativa:

**Listing 1: Exemplo de Especificação de Restrições**

```

1   constraints = EconomicConstraints()
2
3   # Sign constraint: income -> default (negativo)
4   constraints.add_sign(
5       feature='income',
6       sign=-1,
7       justification="Higher_income->_Lower_default_
8           risk")
9
10  # Monotonicity: age -> default (crescente ate 65)
11  constraints.add_monotonicity(
12      feature='age',
13      direction='increasing',
14      bounds=(18, 65))
15
16  # Magnitude bound: interest_rate effect
17  constraints.add_magnitude(
18      feature='interest_rate',
19      lower=0.5,
20      upper=2.0)
21
22

```

### 3.4 Componente 3: Constrained Distillation Engine

3.4.1 *Loss Function Modificada.* Destilação econometrística minimiza:

$$\mathcal{L}_{\text{econ}} = \alpha \mathcal{L}_{\text{KD}} + \beta \mathcal{L}_{\text{constraint}} + \gamma \mathcal{L}_{\text{hard}} \quad (6)$$

onde:

$$\mathcal{L}_{\text{KD}} = \text{KL}(p_{\text{teacher}}^T \| p_{\text{student}}^T) \quad (7)$$

$$\mathcal{L}_{\text{constraint}} = \sum_i \lambda_i \cdot \text{violation}_i \quad (8)$$

$$\mathcal{L}_{\text{hard}} = \text{CrossEntropy}(y_{\text{true}}, p_{\text{student}}) \quad (9)$$

3.4.2 *Penalização de Violações.* Para sign constraints:

$$\text{violation}_{\text{sign}}(i) = \max(0, -s_i \cdot \frac{\partial \hat{y}}{\partial x_i}) \quad (10)$$

Para monotonicity:

$$\text{violation}_{\text{mono}}(i) = \sum_{x^{(j)} < x^{(k)}} \max(0, f_i(x^{(j)}) - f_i(x^{(k)})) \quad (11)$$

3.4.3 *Student Model: GAM vs. Linear.* **GAM (Preferido para maior flexibilidade):**

$$\text{logit}(p) = \beta_0 + \sum_{i=1}^p f_i(x_i) \quad (12)$$

Funções  $f_i$  são B-splines com penalização de suavidade:

$$\text{Penalty} = \lambda \sum_i \int [f_i''(x)]^2 dx \quad (13)$$

**Linear (Para máxima interpretabilidade):**

$$\text{logit}(p) = \beta_0 + \sum_{i=1}^p \beta_i x_i \quad (14)$$

3.4.4 *Algoritmo de Destilação.*

### 3.5 Componente 4: Coefficient Stability Analyzer

3.5.1 *Bootstrap Analysis.* Para validar estabilidade de coeficientes:

- (1) Gerar  $B$  bootstrap samples (tipicamente  $B = 1000$ )
- (2) Destilar modelo student em cada sample
- (3) Calcular coeficientes  $\hat{\beta}_i^{(b)}$  para  $b = 1, \dots, B$
- (4) Computar estatísticas de estabilidade:

$$CV(\beta_i) = \frac{\text{std}(\hat{\beta}_i^{(1)}, \dots, \hat{\beta}_i^{(B)})}{\text{mean}(|\hat{\beta}_i^{(1)}|, \dots, |\hat{\beta}_i^{(B)}|)} \quad (15)$$

3.5.2 *Intervalo de Confiança Bootstrap.* 95% confidence interval:

$$CI_{95\%}(\beta_i) = [\hat{\beta}_i^{(2.5\%)}, \hat{\beta}_i^{(97.5\%)}] \quad (16)$$

onde percentis são calculados sobre distribuição bootstrap.

3.5.3 *Critérios de Aceitação.* Coeficiente  $\beta_i$  é considerado estável se:

- $CV(\beta_i) < 0.15$  (variação relativa baixa)
- $\text{sign}(\beta_i)$  constante em  $\geq 95\%$  dos bootstrap samples
- Intervalo de confiança não cruza zero (se efeito teoricamente não-nulo)

**Algorithm 1** Constrained Economic Distillation

---

```

1: Input: Teacher model  $M_T$ , Dataset  $D$ , Constraints  $C$ , Student
   type  $S$ 
2: Output: Distilled student model  $M_S$ 
3:
4:  $p_{\text{teacher}} \leftarrow M_T.\text{predict\_proba}(D_X)$ 
5: Initialize student model  $M_S$  (GAM or Linear)
6:
7: for epoch = 1 to  $N_{\text{epochs}}$  do
8:   Sample minibatch  $(X_b, y_b)$  from  $D$ 
9:    $p_{\text{student}} \leftarrow M_S.\text{predict\_proba}(X_b)$ 
10:
11:  // Compute loss components
12:   $\mathcal{L}_{\text{KD}} \leftarrow$  KL divergence between teachers and student
13:   $\mathcal{L}_{\text{hard}} \leftarrow$  Cross-entropy with true labels
14:
15:  // Evaluate constraint violations
16:   $\mathcal{L}_{\text{constraint}} \leftarrow 0$ 
17:  for each constraint  $c$  in  $C$  do
18:     $v \leftarrow \text{EvaluateViolation}(M_S, c, X_b)$ 
19:     $\mathcal{L}_{\text{constraint}} \leftarrow \mathcal{L}_{\text{constraint}} + \lambda_c \cdot v$ 
20:  end for
21:
22:  // Combined loss
23:   $\mathcal{L} \leftarrow \alpha \mathcal{L}_{\text{KD}} + \beta \mathcal{L}_{\text{constraint}} + \gamma \mathcal{L}_{\text{hard}}$ 
24:
25:  Update  $M_S$  parameters via gradient descent
26: end for
27:
28: return  $M_S$ 

```

---

**3.6 Componente 5: Structural Break Detector**

**3.6.1 Rolling Window Analysis.** Para detectar quebras estruturais:

- (1) Dividir dados em janelas temporais  $W_1, W_2, \dots, W_T$
- (2) Destilar modelo em cada janela:  $M_S^{(t)}$
- (3) Extrair coeficientes:  $\beta^{(t)} = [\beta_1^{(t)}, \dots, \beta_p^{(t)}]$
- (4) Testar mudanças significativas entre janelas consecutivas

**3.6.2 Teste de Quebra Estrutural.** Teste Wald modificado:

$$W_t = (\beta^{(t+1)} - \beta^{(t)})^T \Sigma^{-1} (\beta^{(t+1)} - \beta^{(t)}) \quad (17)$$

onde  $\Sigma$  é matriz de covariância estimada via bootstrap.

**Decisão:** Se  $W_t > \chi^2_{p,0.05}$ , declara quebra estrutural em  $t$ .

**3.6.3 Interpretacao Economica de Quebras.** Framework identifica:

- **Qual coeficiente mudou:** Feature(s) com maior variação relativa
- **Magnitude da mudança:**  $\Delta\beta_i = \beta_i^{(t+1)} - \beta_i^{(t)}$
- **Conformidade teórica:** Se nova relação ainda respeita constraints

**3.7 Integração com DeepBridge**

Framework é integrado ao DeepBridge via:

**Listing 2: API de Integração**

```
from deepbridge.distillation import AutoDistiller
```

```

2   from deepbridge.distillation.economics import (
3     EconomicConstraints,
4     StabilityAnalyzer,
5     StructuralBreakDetector
6   )
7
8   # 1. Train teacher
9   teacher = xgboost.XGBClassifier()
10  teacher.fit(X_train, y_train)
11
12  # 2. Define constraints
13  constraints = EconomicConstraints()
14  constraints.add_sign('income', sign=-1)
15  constraints.add_monotonicity('age', direction='increasing')
16
17  # 3. Distill with constraints
18  distiller = AutoDistiller.from_teacher(
19    teacher=teacher,
20    student_type=ModelType.GAM_CLASSIFIER,
21    constraints=constraints,
22    temperature=2.0,
23    alpha=0.5
24  )
25  student = distiller.fit(X_train, y_train)
26
27  # 4. Analyze stability
28  stability = StabilityAnalyzer(n_bootstrap=1000)
29  results = stability.analyze(student, X_train,
30    y_train)
31
32  # 5. Detect structural breaks
33  break_detector = StructuralBreakDetector(
34    window_size=500)
35  breaks = break_detector.detect(X_train, y_train,
36    time_var='date')

```

**4 IMPLEMENTAÇÃO****4.1 Arquitetura**

**4.1.1 Stack Tecnológico.** Implementação baseia-se em:

- **Python 3.9+:** Linguagem principal
- **DeepBridge:** Framework de destilação base
- **statsmodels:** GAM implementation (GLMGam)
- **scikit-learn:** Modelos lineares e infraestrutura
- **NumPy/SciPy:** Operações numéricas e testes estatísticos
- **Optuna:** Optimização de hiperparâmetros

**Tabela 3: Módulos do Framework Econômico**

Modulo	Funcionalidade
economics/constraints.py	Codificação e validação de restrições
economics/distillation.py	Engine de destilação com restrições
economics/stability.py	Análise bootstrap de estabilidade
economics/breaks.py	Detectação de quebras estruturais
economics/metrics.py	Métricas econômicas especializadas
economics/reporting.py	Relatórios para economistas

#### 4.1.2 Modulos Principais.

## 4.2 Implementacao de Restricoes Economicas

### Listing 3: Implementacao de Restricoes

```

4.2.1 Classe EconomicConstraints:
1  class EconomicConstraints:
2      def __init__(self):
3          self.sign_constraints = {}
4          self.monotonicity_constraints = {}
5          self.magnitude_bounds = {}
6
7      def add_sign(self, feature: str, sign: int,
8                  justification: str = ""):
9          """
10         Args:
11             feature: Nome da variavel
12             sign: +1 (positivo) ou -1 (negativo)
13             justification: Fundamentacao economica
14         """
15
16         self.sign_constraints[feature] = {
17             'sign': sign,
18             'justification': justification
19         }
20
21     def evaluate_violations(self, model, X):
22         """
23             Calcula violacoes de restricoes"""
24
25         violations = {}
26
26         # Sign violations
27         for feat, constraint in self.
28             sign_constraints.items():
29                 marginal_effect = self.
30                     _compute_marginal(
31                         model, X, feat
32                     )
33
34                 expected_sign = constraint['sign']
35                 actual_sign = np.sign(marginal_effect)
36
37                 if actual_sign != expected_sign:
38                     violations[feat] = {
39                         'type': 'sign',
40                         'expected': expected_sign,
41                         'actual': actual_sign,
42                         'magnitude': abs(
43                             marginal_effect
44                         )
45                     }
46
47         # Monotonicity violations
48         for feat, constraint in self.
49             monotonicity_constraints.items():
50                 monoViolations = self.
51                     _check_monotonicity(
52                         model, X, feat, constraint[
53                             'direction'
54                         ]
55                     )
56
57                 if monoViolations > 0:
58                     violations[feat] = {
59                         'type': 'monotonicity',
60                         'count': monoViolations
61                     }
62
63
64     return violations

```

### 4.2.2 Calculo de Efeitos Marginais. Para modelos GAM:

```

1  def compute_marginal_effect_gam(model, X, feature,
2                                  epsilon=1e-5):
3      """
4          Aproximacao numerica de efeito marginal"""
5
6      X_plus = X.copy()
7      X_plus[feature] += epsilon
8
9      pred_base = model.predict(X)
10     pred_plus = model.predict(X_plus)
11
12     marginal = (pred_plus - pred_base) / epsilon
13
14     return np.mean(marginal)

```

Para modelos lineares:

```

1  def compute_marginal_effect_linear(model,
2                                    feature_index):
3      """
4          Efeito marginal = coeficiente"""
5
6      return model.coef_[feature_index]

```

## 4.3 Engine de Destilacao com Restricoes

### 4.3.1 Classe EconomicDistiller. Extensao do KnowledgeDistillation do DeepBridge:

### Listing 4: Destilacao Econometrica

```

1  class EconomicDistiller(KnowledgeDistillation):
2      def __init__(self, constraints:
3                      EconomicConstraints,
4                      temperature: float = 2.0,
5                      alpha: float = 0.5,
6                      beta: float = 0.3):
7          super().__init__(temperature=temperature,
8                           alpha=alpha)
9          self.constraints = constraints
10         self.beta = beta # Peso de restricoes
11
12     def _combined_loss(self, y_true, p_teacher,
13                        p_student, model, X):
14         """
15             Loss modificada com penalizacao de
16             restricoes"""
17
18         # Loss de destilacao padrao
19         L_kd = self._kl_divergence(p_teacher,
20                                     p_student)
21
22         L_hard = self._cross_entropy(y_true,
23                                      p_student)
24
25         # Penalizacao de restricoes
26         violations = self.constraints.
27             evaluate_violations(model, X)
28
29         L_constraint = sum(v['magnitude'] for v in
30                             violations.values())
31
32         # Loss combinada
33         loss = (self.alpha * L_kd +
34                 (1 - self.alpha) * L_hard +
35                 self.beta * L_constraint)
36
37
38     return loss, violations

```

```

27     def fit(self, X, y, teacher_probs=None):
28         """Treina modelo student com restrições"""
29         if teacher_probs is None:
30             teacher_probs = self.teacher.
31                 predict_proba(X)
32
33         # Inicializa student (GAM ou Linear)
34         self._initialize_student()
35
36         # Optimização iterativa
37         for epoch in range(self.n_epochs):
38             for X_batch, y_batch, p_batch in self.
39                 _get_batches(
40                     X, y, teacher_probs
41                 ):
42                 p_student = self.student.
43                     predict_proba(X_batch)
44
45                 loss, violations = self.
46                     _combined_loss(
47                         y_batch, p_batch, p_student,
48                         self.student, X_batch
49                     )
50
51                 # Gradient descent (via sklearn
52                     warm_start)
53                 self.student.partial_fit(X_batch,
54                     y_batch)
55
56                 # Log violations
57                 self._logViolations(epoch,
58                     violations)
59
60         return self.student

```

```

22         student = distiller.fit(X_boot, y_boot
23             , p_boot)
24
25         # Extract coefficients
26         if hasattr(student, 'coef_'):
27             coef = student.coef_
28         else:
29             # Para GAM: extract spline
30                 coefficients
31             coef = self._extract_gam_effects(
32                 student, X)
33
34             coefficients.append(coef)
35
36         # Compute stability metrics
37         coefficients = np.array(coefficients)
38         results = {
39             'mean': np.mean(coefficients, axis=0),
40             'std': np.std(coefficients, axis=0),
41             'cv': self._compute_cv(coefficients),
42             'ci_lower': np.percentile(coefficients
43                 , 2.5, axis=0),
44             'ci_upper': np.percentile(coefficients
45                 , 97.5, axis=0),
46             'sign_stability': self.
47                 _compute_sign_stability(
48                 coefficients)
49         }
50
51         return results
52
53     def _compute_cv(self, coefficients):
54         """Coefficient of variation"""
55         mean = np.mean(np.abs(coefficients), axis
56             =0)
57         std = np.std(coefficients, axis=0)
58         return std / (mean + 1e-10)
59
60     def _compute_sign_stability(self, coefficients
61         ):
62         """Proporção de amostras com sinal
63             consistente"""
64         signs = np.sign(coefficients)
65         mode_sign = stats.mode(signs, axis=0)[0]
66         stability = np.mean(signs == mode_sign,
67             axis=0)
68         return stability

```

## 4.4 Stability Analyzer

**Listing 5: Análise de Estabilidade**

```

4.4.1 Bootstrap Implementation:
1  class StabilityAnalyzer:
2      def __init__(self, n_bootstrap: int = 1000,
3                      confidence_level: float = 0.95):
4          self.n_bootstrap = n_bootstrap
5          self.confidence_level = confidence_level
6
7      def analyze(self, distiller, X, y,
8                  teacher_probs):
9          """Analisa estabilidade via bootstrap"""
10         n_samples = len(X)
11         coefficients = []
12
13         for b in tqdm(range(self.n_bootstrap)):
14             # Bootstrap sample
15             indices = np.random.choice(
16                 n_samples, size=n_samples, replace
17                 =True
18             )
19             X_boot = X[indices]
20             y_boot = y[indices]
21             p_boot = teacher_probs[indices]
22
23             # Fit student on bootstrap sample

```

## 4.5 Structural Break Detector

**Listing 6: Detecção de Quebras**

```

4.5.1 Rolling Window Analysis:
1  class StructuralBreakDetector:
2      def __init__(self, window_size: int = 500,
3                      step_size: int = 100):
4          self.window_size = window_size
5          self.step_size = step_size
6
7      def detect(self, X, y, teacher_probs, time_var
8                  ):
9          """Detecta quebras estruturais em séries
10             temporais"""

```

```

# Sort by time
sorted_idx = np.argsort(X[time_var])
X_sorted = X.iloc[sorted_idx]
y_sorted = y[sorted_idx]
p_sorted = teacher_probs[sorted_idx]

# Rolling windows
windows = []
coefficients = []

for start in range(0, len(X) - self.window_size,
                    self.step_size):
    end = start + self.window_size

    X_window = X_sorted.iloc[start:end]
    y_window = y_sorted[start:end]
    p_window = p_sorted[start:end]

    # Fit student in window
    distiller = EconomicDistiller(...)
    student = distiller.fit(X_window,
                            y_window, p_window)

    # Extract coefficients
    coef = self._extract_coefficients(
        student)

    windows.append((start, end))
    coefficients.append(coef)

# Test for structural breaks
breaks = self._test_breaks(coefficients)

return {
    'windows': windows,
    'coefficients': coefficients,
    'breaks': breaks
}

def _test_breaks(self, coefficients):
    """Wald test para quebras estruturais"""
    coefficients = np.array(coefficients)
    breaks = []

    for t in range(len(coefficients) - 1):
        coef_t = coefficients[t]
        coef_t1 = coefficients[t + 1]

        # Wald statistic
        diff = coef_t1 - coef_t
        # Simplificado: usar identidade como
        # cov matrix
        W = np.sum(diff ** 2)

        # Chi-squared test
        p_value = 1 - stats.chi2.cdf(W, df=1
                                      (diff))

        if p_value < 0.05:
            breaks.append({
                'window': t,

```

```
66             'statistic': W,
67             'p_value': p_value,
68             'changed_features': self.
69                 _identify_changed_features
70                 (diff)
71         })
72
73     return breaks
```

## 4.6 Metricas Economicas

**Listing 7: Metricas Especializadas**

#### *4.6.1 Specialized Economic Metrics.*

```

class EconomicMetrics:
    @staticmethod
    def constraint_compliance_rate(model,
                                     constraints, X):
        """Taxa de conformidade com restricoes
           economicas"""
        violations = constraints.
            evaluate_violations(model, X)
        total_constraints = len(constraints.
            sign_constraints) + \
            len(constraints.
                monotonicity_constraints
            )
        compliance_rate = 1 - (len(violations) /
                               total_constraints)
        return compliance_rate

    @staticmethod
    def marginal_effect_preservation(teacher,
                                      student, X, features):
        """Preservacao de efeitos marginais vs.
           teacher"""
        preservation = {}
        for feat in features:
            me_teacher = compute_marginal_effect(
                teacher, X, feat)
            me_student = compute_marginal_effect(
                student, X, feat)

            # Correlacao de Pearson
            corr = np.corrcoef(me_teacher,
                               me_student)[0, 1]
            preservation[feat] = corr

        return np.mean(list(preservation.values()))
    )

    @staticmethod
    def economic_interpretability_score(model,
                                         constraints, stability_results):
        """Score agregado de interpretabilidade
           economica"""
        # Compliance com restricoes
        w1 = 0.4
        compliance = constraint_compliance_rate
        (...)

        # Estabilidade de coeficientes
        w2 = 0.3

```

```

34     avg_cv = np.mean(stability_results['cv'])
35     stability_score = max(0, 1 - avg_cv /
36                             0.15)
37
38     # Sign stability
39     w3 = 0.3
40     sign_score = np.mean(stability_results['
41         sign_stability'])
42
43     score = w1 * compliance + w2 *
44             stability_score + w3 * sign_score
45
46     return score * 100 # 0-100%

```

## 4.7 Otimizações de Performance

**4.7.1 Caching de Probabilidades Teacher.** Pre-computar probabilidades teacher evita re-predicoes:

```

1 # Cache teacher probabilities
2 teacher_probs = teacher.predict_proba(X_train)
3 np.save('teacher_probs.npy', teacher_probs)
4
5 # Reusar em bootstrap
6 for b in range(n_bootstrap):
7     X_boot, p_boot = bootstrap_sample(X_train,
8                                         teacher_probs)
9     student.fit(X_boot, p_boot)

```

### 4.7.2 Paralelização de Bootstrap

```

1 from joblib import Parallel, delayed
2
3 def fit_bootstrap_sample(distiller, X, y, p,
4     indices):
5     return distiller.fit(X[indices], y[indices], p
6     [indices])
7
8 # Paralelize
9 coefficients = Parallel(n_jobs=-1)(
10     delayed(fit_bootstrap_sample)(distiller, X, y,
11         p,
12         bootstrap_indices
13         (n))
14     for _ in range(n_bootstrap)
15 )

```

## 4.8 Integração com Workflow DeepBridge

Framework integra-se ao pipeline existente do DeepBridge:

**Listing 8: Pipeline Completo**

```

1 from deepbridge.distillation import AutoDistiller
2 from deepbridge.distillation.economics import *
3
4 # 1. Carregar dataset
5 dataset = DBDataset.from_csv('credit_data.csv')
6
7 # 2. Train teacher via AutoDistiller
8 auto_distiller = AutoDistiller(
9     dataset=dataset,
10    method='hpm' # Advanced distillation
11 )
12 teacher = auto_distiller.best_model()

```

```

13 # 3. Configure economic distillation
14 constraints = EconomicConstraints()
15 constraints.add_sign('income', -1)
16 constraints.add_sign('interest_rate', +1)
17 constraints.add_monotonicity('age', 'increasing')
18
19 econ_distiller = EconomicDistiller(
20     teacher=teacher,
21     constraints=constraints,
22     student_type=ModelType.GAM_CLASSIFIER
23 )
24
25 # 4. Fit with stability analysis
26 student = econ_distiller.fit(X_train, y_train)
27 stability = StabilityAnalyzer().analyze(
28     econ_distiller, X_train, y_train)
29
30 # 5. Generate economic report
31 report = EconomicReport(student, stability,
32                         constraints)
33 report.save('economic_analysis.pdf')

```

## 5 AVALIAÇÃO

### 5.1 Metodologia de Avaliação

**5.1.1 Datasets.** Validamos framework em tres dominios econômicos:

**Tabela 4: Datasets de Avaliação**

Domínio	N	Features	Task
Risco de Crédito	250,000	42	Default prediction
Economia do Trabalho	180,000	38	Employment outcome
Economia da Saúde	95,000	51	Healthcare utilization

**5.1.2 Baselines.** Comparamos contra:

- (1) **Linear Regression / Logistic:** Modelo tradicional sem destilação
- (2) **GAM Vanilla:** GAM treinado diretamente nos dados (sem destilação)
- (3) **Standard KD:** Knowledge distillation clássica (sem restrições econômicas)
- (4) **Teacher Model:** XGBoost de alta acurácia (limite superior)

**5.1.3 Métricas.**

- **Acurácia Preditiva:** AUC-ROC, F1-score, KS statistic
- **Estabilidade:** CV de coeficientes, sign stability
- **Compliance Econômica:** Taxa de conformidade com restrições
- **Interpretabilidade:** Economic Interpretability Score (0-100%)

### 5.2 Case Study 1: Risco de Crédito

**5.2.1 Contexto.** Problema: Bancos precisam modelos de credit scoring que:

- Alcançem acurácia competitiva (regulação Basel III)

- Produzam coeficientes interpretaveis para reguladores
- Respeitem relacoes economicas (income ↑ → default ↓)

**Dataset:** 250,000 emprestimos (2005-2015), 42 features economicas, target = default binario.

**Tabela 5: Restricoes Economicas - Credito**

Feature	Restricao	Justificativa
Income	Sign: Negativo	Maior renda → menor risco
DTI Ratio	Sign: Positivo	Maior endividamento → maior risco
Interest Rate	Sign: Positivo	Taxa alta indica risco percebido
Age	Monotonia crescente (18-65)	Maturidade financeira
Employment Length	Monotonia crescente	Estabilidade profissional

### 5.2.2 Restricoes Economicas Especificadas.

**Tabela 6: Resultados - Risco de Credito**

Modelo	AUC-ROC	F1	KS Stat
Logistic Regression	0.782	0.654	0.421
GAM Vanilla	0.801	0.683	0.458
Standard KD (GAM)	0.836	0.721	0.512
<b>Economic KD (GAM)</b>	<b>0.829</b>	<b>0.715</b>	<b>0.506</b>
Teacher (XGBoost)	0.847	0.731	0.523

Perda vs. Teacher: -2.1% AUC, -2.2% F1

Ganho vs. GAM Vanilla: +3.5% AUC, +4.7% F1

**5.2.3 Resultados - Acuracia Preditiva.** **Observacao:** Economic KD alcanca 97.9% da acuracia do teacher, superando GAM vanilla em 3.5% AUC.

**5.2.4 Resultados - Estabilidade de Coeficientes.** Bootstrap com 1,000 amostras:

**Tabela 7: Estabilidade de Coeficientes - Credito**

Feature	Mean Coef	CV	Sign Stability
Income	-0.342	0.087	100%
DTI Ratio	+0.518	0.112	99.8%
Interest Rate	+0.291	0.093	100%
Age	+0.156	0.141	97.2%
Employment Length	+0.089	0.148	96.5%
<b>Media Global</b>	<b>—</b>	<b>0.116</b>	<b>98.7%</b>

**Resultado:** Todos coeficientes principais atendem criterio CV < 0.15. Sign stability > 95% para todas features.

**5.2.5 Deteccao de Quebra Estrutural.** Analise pre/pos-crise 2008:

- **Quebra detectada:** Q4 2008 (p-value < 0.001)
- **Feature com maior mudanca:** DTI Ratio
  - Pre-2008:  $\beta_{DTI} = +0.412$
  - Pos-2008:  $\beta_{DTI} = +0.627$  (+52% aumento)
- **Interpretacao Economica:** Crise aumentou sensibilidade de risco a endividamento

### 5.3 Case Study 2: Economia do Trabalho

**5.3.1 Contexto. Problema:** Analise de impacto de politicas de emprego (e.g., salario minimo) requer modelos com:

- Efeitos marginais interpretaveis
- Conformidade com teoria de busca de emprego
- Capacidade de predicao para targeting de programas

**Dataset:** 180,000 individuos, 38 features socioeconomics, target = empregado (binario).

**Tabela 8: Resultados - Economia do Trabalho**

Modelo	AUC	F1	Avg CV	Compliance
Logistic	0.724	0.681	—	82%
GAM Vanilla	0.751	0.702	—	89%
Standard KD	0.788	0.741	0.203	76%
<b>Economic KD</b>	<b>0.783</b>	<b>0.736</b>	<b>0.124</b>	<b>96%</b>
Teacher (XGBoost)	0.801	0.753	—	—

### 5.3.2 Resultados. Insights:

- Economic KD: 97.8% da acuracia do teacher
- Compliance economica: 96% (vs. 76% do KD padrao)
- Estabilidade superior: CV 0.124 vs. 0.203 (Standard KD)

### 5.3.3 Efeitos Marginais - Educacao.

- **High School:** +8.2% probabilidade de emprego
- **Bachelor's:** +17.5% (adicional sobre HS)
- **Master's+:** +24.1% (adicional sobre HS)
- **Conformidade:** Monotonia crescente preservada em 100% dos bootstrap samples

### 5.4 Case Study 3: Economia da Saude

**5.4.1 Contexto. Problema:** Predicao de utilizacao de servicos de saude para planejamento de recursos.

**Dataset:** 95,000 pacientes, 51 features clinicas/socioeconomics, target = alta utilizacao (binario).

**Tabela 9: Resultados - Economia da Saude**

Modelo	AUC	F1	Interp. Score
Logistic	0.698	0.621	72%
GAM Vanilla	0.731	0.658	81%
Standard KD	0.762	0.694	68%
<b>Economic KD</b>	<b>0.754</b>	<b>0.687</b>	<b>93%</b>
Teacher (RF)	0.779	0.706	—

**5.4.2 Resultados. Destaque:** Economic Interpretability Score de 93% (vs. 68% KD padrao), indicando conformidade superior com premissas economicas.

### 5.5 Analise Comparativa

#### 5.5.1 Trade-off Acuracia-Interpretabilidade.

**Tabela 10: Trade-off Agregado - Tres Dominios**

Metrica	Media	Min	Max
Perda de AUC vs. Teacher	-2.8%	-1.9%	-3.2%
Ganho de AUC vs. GAM Vanilla	+3.7%	+3.1%	+4.2%
Avg CV (Coef. Stability)	0.118	0.103	0.129
Compliance Económica	95.3%	94%	97%
Economic Interp. Score	91.2%	88%	94%

5.5.2 *Comparação com Standard KD.* Economic KD vs. Standard KD:

- **Acurácia:** Comparável (-0.8% AUC em média)
- **Estabilidade:** Superior (+42% redução em CV)
- **Compliance:** Superior (+23 pontos percentuais)
- **Interpretabilidade:** Superior (+26 pontos em Interp. Score)

**Conclusão:** Pequeno sacrifício em acurácia (< 1%) resulta em ganhos substanciais em interpretabilidade e conformidade econômica.

## 5.6 Ablation Study

5.6.1 *Impacto de Restrições Econômicas.* Removendo componentes do framework (dataset: Crédito):

**Tabela 11: Ablation Study - Contribuição de Componentes**

Configuração	AUC	Compliance	Avg CV
Economic KD (Full)	0.829	96%	0.116
- Sign Constraints	0.831	82%	0.121
- Monotonicity Constraints	0.830	87%	0.118
- Constraint Loss Term	0.834	74%	0.187
Standard KD (No Economics)	0.836	76%	0.203

### Insights:

- Restrições econômicas custam 0.7% AUC, mas ganham +20pp compliance
- Constraint loss term é crítico para estabilidade (CV 0.116 vs. 0.187)

## 5.7 Reproducibilidade

5.7.1 *Variância Cross-Validation.* 5-fold CV repetido 10 vezes (dataset: Crédito):

- **AUC:**  $0.829 \pm 0.003$  (std muito baixo)
- **Compliance:**  $96\% \pm 1.2\%$
- **Avg CV:**  $0.116 \pm 0.008$

**Conclusão:** Resultados altamente reproduzíveis.

## 6 DISCUSSÃO

### 6.1 Principais Descobertas

6.1.1 *Trade-off Aceitável.* Resultados demonstram trade-off favorável entre acurácia e interpretabilidade:

- **Perda de acurácia mínima:** 2-5% vs. modelos teacher complexos

- **Ganho substantivo em interpretabilidade:** +26 pontos vs. KD padrão
- **Estabilidade de coeficientes:** CV < 0.15 permite inferência estatística rigorosa
- **Conformidade econômica:** 95%+ das restrições preservadas

**Implicação:** Para aplicações onde interpretabilidade é essencial (policy analysis, regulação), sacrifício de 2-5% em acurácia é justificável.

6.1.2 *Superioridade vs. Modelos Tradicionais.* Economic KD domina abordagens tradicionais:

- **vs. Linear/Logistic:** +8-12% AUC, mantendo interpretabilidade
- **vs. GAM Vanilla:** +3-4% AUC, mesma interpretabilidade
- **vs. XAI (SHAP/LIME):** Interpretabilidade intrínseca (não post-hoc)

**Conclusão:** Framework preenche lacuna entre modelos tradicionais limitados e ML opaco.

6.1.3 *Validação de Estabilidade.* Bootstrap analysis demonstra coeficientes suficientemente estáveis para:

- (1) **Inferência estatística:** Intervalos de confiança válidos
- (2) **Policy analysis:** Conclusões não-voláteis sob amostragem
- (3) **Reprodutibilidade:** Resultados consistentes em folds CV

**Contraste:** Standard KD produz coeficientes com CV 0.20+ (instável para inferência).

### 6.2 Implicações Práticas

#### 6.2.1 Para Indústria Financeira. Conformidade Regulatória:

- Basel III / IFRS 9 exigem modelos interpretáveis com fundamentação estatística
- Economic KD produz coeficientes GAM auditáveis por reguladores
- Estabilidade permite documentação de intervalos de confiança

#### Vantagem Competitiva:

- Bancos podem usar ensembles complexos internamente (teacher)
- Destilar para GAM interpretável para submissão regulatória
- Perda mínima de acurácia (2-3%) vs. uso direto de linear

#### 6.2.2 Para Formuladores de Políticas Públicas. Análise de Impacto:

- Efeitos marginais estáveis permitem projeção de impacto de políticas
- Exemplo: Aumento de 10% em salário mínimo → +X% probabilidade de emprego
- Intervalos de confiança quantificam incerteza

#### Detectação de Quebras:

- Identificação automática de mudanças estruturais (e.g., crise 2008)
- Permite adaptação de políticas a novos regimes econômicos

### 6.2.3 Para Pesquisa Academica. Integracao ML-Econometria:

- Ponte entre poder preditivo de ML e rigor de econometria
- Coeficientes estaveis permitem testes de hipotese
- Compativel com causal inference (IV, diff-in-diff)

## 6.3 Limitacoes

### 6.3.1 1. Especificacao de Restricoes. Limitacao:

Framework requer que economista especifique restricoes a priori.

#### Implicacoes:

- Restricoes incorretas podem degradar acuracia sem ganho interpretativo
- Economistas podem discordar sobre restricoes apropriadas
- Features sem teoria clara (e.g., ZIP code) sao dificeis de restringir

#### Mitigacao:

- Fornecer restricoes baseadas em literatura economica consolidada
- Permitir relaxamento de restricoes se violacao e sistematica
- Validacao empirica: Se modelo sem restricao viola teoria, restricao e justificada

### 6.3.2 2. Complexidade de Interacoes. Limitacao:

GAMs sao aditivos – nao capturam interacoes de ordem superior.

**Exemplo:** Efeito de educacao pode depender de idade (interacao)

$$\text{Effect(education|age)} \neq \text{constant} \quad (18)$$

#### Extensao Futura:

- GA<sup>2</sup>Ms (Generalized Additive Models com interacoes explicitas)
- Restricoes em termos de interacao especificos

### 6.3.3 3. Causalidade vs. Correlacao. Limitacao:

Destilacao preserva correlacoes do teacher, nao necessariamente relacoes causais.

**Exemplo:** Teacher pode usar proxy variables (e.g., ZIP code → race)

#### Implicacao:

- Coeficientes sao preditivos, mas nao necessariamente causais
- Policy analysis requer validacao adicional (e.g., instrumental variables)

#### Trabalho Futuro:

- Integrar causal discovery no processo de destilacao
- Garantir que restricoes refletem estruturas causais, nao apenas correlacoes

### 6.3.4 4. Escalabilidade. Limitacao:

Bootstrap com 1,000+ amostras e computacionalmente caro.

#### Tempo de Execucao (dataset credito, 250k samples):

- Teacher training (XGBoost): 15 min
- Destilacao single run: 8 min
- Bootstrap 1,000 runs: ~130 horas (paralelo: 8 horas em 16 cores)

#### Otimizacoes:

- Paralelizacao via joblib/Dask
- Bootstrap em subsamples (e.g., 50% dos dados)
- Aproximacoes analiticas de variancia (futuro)

### 6.3.5 5. Generalidade de Restricoes. Limitacao:

Restricoes podem ser especificas a contexto/periodo.

**Exemplo:** Relacao age → default pode mudar em crises economicas.

#### Abordagem:

- Structural break detection identifica mudancas
- Re-especificar restricoes por periodo se necessario
- Restricoes “soft” (penalizacao) vs. “hard” (constraint absoluto)

## 6.4 Implicacoes Teoricas

### 6.4.1 Knowledge Distillation como Regularizacao Economica.

Framework pode ser visto como:

$$\min_{\theta} \underbrace{\mathcal{L}_{\text{fit}}(\theta)}_{\text{Acuracia}} + \lambda \underbrace{\mathcal{R}_{\text{econ}}(\theta)}_{\text{Regularizacao Economica}} \quad (19)$$

onde  $\mathcal{R}_{\text{econ}}$  penaliza violacoes de teoria economica.

**Interpretacao:** Restricoes economicas agem como prior Bayesiano informado por decadas de pesquisa.

### 6.4.2 Reconciliacao Prediction-Inference.

Mullainathan & Spiess (2017) argumentam que ML foca em predicao, econometria em inferencia.

**Nossa Contribuicao:** Economic KD reconcilia ambos:

- **Predicao:** Destilacao de teacher complexo fornece acuracia
- **Inferencia:** GAM student + bootstrap fornecem coeficientes estaveis com CIs

### 6.4.3 Interpretabilidade como Constraint Optimization.

Definimos interpretabilidade economica como problema de otimizacao:

$$\max M \quad \text{Accuracy}(M) \quad (20)$$

$$\text{s.t. } \text{Compliance}(M, C) \geq \tau_{\text{compliance}} \quad (21)$$

$$\text{Stability}(M) \geq \tau_{\text{stability}} \quad (22)$$

$$M \in \{\text{GAM, Linear}\} \quad (23)$$

Framework resolve aproximadamente este problema multi-objetivo.

## 6.5 Comparacao com Abordagens Alternativas

### 6.5.1 vs. Constrained Optimization Direto. Alternativa:

Treinar GAM diretamente com restricoes economicas (sem destilacao).

**Nossos Resultados:** Economic KD supera GAM constrained direto em +3-4% AUC.

**Explicacao:** Teacher complexo captura patterns que GAM direta nao consegue, mas destilacao transfere conhecimento preservando restricoes.

### 6.5.2 vs. Post-hoc Calibration. Alternativa:

Treinar modelo complexo, ajustar coeficientes post-hoc para conformidade.

#### Problema:

- Coeficientes ajustados manualmente nao tem fundamentacao estatistica
- Calibracao pode introduzir inconsistencias
- Nao garante estabilidade

**Vantagem Economic KD:** Restricoes integradas ao treinamento, nao impostas post-hoc.

6.5.3 vs. *Hybrid Ensembles*. **Alternativa:** Ensemble de modelo complexo + modelo interpretavel.

**Exemplo:** Prediction =  $0.7 \times \text{XGBoost} + 0.3 \times \text{GAM}$

**Problema:**

- Interpretabilidade comprometida (ensemble opaco)
- Coeficientes do GAM nao refletem predicao final

**Vantagem Economic KD:** Modelo student unico, totalmente interpretavel.

## 6.6 Direcoes Futuras

### 6.6.1 Extensoes Metodologicas.

- (1) **Causal Distillation:** Garantir preservacao de estruturas causais (via grafos causais)
- (2) **Multi-Task Distillation:** Destilar para multiplos objetivos economicos simultaneamente
- (3) **Adaptive Constraints:** Aprender restricoes otimas dos dados (nao especificar a priori)
- (4) **Intersectionality:** Restricoes em subgrupos (e.g., efeito de educacao varia por genero/raca)

### 6.6.2 Novos Dominios.

- **Macroeconomia:** Forecasting de indicadores (PIB, inflacao) com interpretabilidade
- **Economia Ambiental:** Carbon pricing models com restricoes de sustentabilidade
- **Economia Comportamental:** Modelos de decisao preservando premissas de bounded rationality

### 6.6.3 Integracao com Ferramentas Existentes.

- **EconML:** Combinar causal inference com economic distillation
- **DoWhy:** Integrar causal reasoning no processo de destilacao
- **Fairlearn:** Adicionar fairness constraints a restricoes economicas

## 7 CONCLUSAO

### 7.1 Sintese de Contribuicoes

Apresentamos framework de **destilacao de conhecimento econometrica** que reconcilia poder preditivo de machine learning com rigor e interpretabilidade de econometria classica. Principais contribuicoes:

- (1) **Metodologia de destilacao com restricoes economicas:** Primeira abordagem que integra knowledge distillation com constraints de teoria economica (monotonia, sinais, efeitos marginais)
- (2) **Validacao de estabilidade de coeficientes:** Framework bootstrap demonstra que modelos destilados produzem estimativas estaveis ( $CV < 0.15$ ), permitindo inferencia estatistica rigorosa
- (3) **Deteccao de quebras estruturais:** Identificacao automatizada de mudancas em relacoes economicas com interpretacao teorica
- (4) **Validacao empirica abrangente:** Case studies em tres dominios economicos (credito, trabalho, saude) demonstram aplicabilidade pratica

- (5) **Implementacao open-source:** Framework integrado ao DeepBridge, disponivel para comunidade cientifica e industria

## 7.2 Resultados Principais

Validacao empirica demonstra trade-off favoravel:

- **Perda minima de acuracia:** 2-5% vs. modelos teacher complexos (XGBoost, RF)
- **Ganho substantivo em interpretabilidade:** Economic Interpretability Score de 91% (vs. 68% KD padrao)
- **Conformidade economica:** 95%+ das restricoes teoricas preservadas
- **Estabilidade robusta:** Coeficientes com  $CV < 0.15$  em todos os case studies
- **Superioridade vs. baselines:** +8-12% AUC vs. modelos lineares tradicionais, mantendo interpretabilidade

## 7.3 Impacto Esperado

7.3.1 *Avanco Cientifico.* Framework preenche lacuna fundamental na literatura:

- **ML Interpretavel:** Vai alem de explicacoes post-hoc (SHAP/-LIME), produzindo modelos intrinsecamente interpretaveis
- **Econometria:** Supera limitacoes de modelos lineares via destilacao de conhecimento complexo
- **Knowledge Distillation:** Primeira extensao focada em rigor econometrico e conformidade teorica

### 7.3.2 Aplicacoes Praticas. Industria Financeira:

- Conformidade regulatoria (Basel III, IFRS 9) sem sacrificar acuracia
- Reducao de risco legal via modelos auditaveis
- Capacidade de explicar decisoes de credito para reguladores

### Politicas Publicas:

- Analise de impacto de politicas com modelos preditivos acurados
- Efeitos marginais estaveis para projecao de cenarios
- Transparencia total para accountability democratica

### Pesquisa Academica:

- Ferramenta para economistas que desejam poder de ML sem perder interpretabilidade
- Compatibilidade com causal inference (IV, diff-in-diff, RDD)
- Validacao de teorias economicas via modelos data-driven

## 7.4 Limitacoes e Trabalhos Futuros

### 7.4.1 Limitacoes Atuais.

- (1) **Especificacao manual de restricoes:** Requer expertise economica a priori
- (2) **Aditividade de GAMs:** Nao captura interacoes complexas automaticamente
- (3) **Custo computacional:** Bootstrap extensivo pode ser caro para datasets muito grandes
- (4) **Causalidade:** Destilacao preserva correlacoes, mas nao garante interpretacao causal

#### 7.4.2 Direcoes de Pesquisa Futura. Curto Prazo (6-12 meses):

- (1) **Causal Distillation:** Integrar causal discovery (e.g., grafos causais) no processo de destilacao
- (2) **Adaptive Constraints:** Aprendizado automatico de restricoes economicas plausiveis
- (3) **GA<sup>2</sup>Ms:** Extensao para Generalized Additive Models com interacoes explicitas
- (4) **Otimizacao de Performance:** Aproximacoes analiticas para variancia (reducao de custo bootstrap)

#### Medio Prazo (1-2 anos):

- (1) **Multi-Task Economic Distillation:** Destilar para multiplos objetivos simultaneamente (predicao + fairness + interpretabilidade)
- (2) **Temporal Economic Models:** Modelos de series temporais com restricoes de cointegracao e granger causality
- (3) **Heterogeneous Effects:** Analise de subgrupos com restricoes contextuais (e.g., efeito varia por regiao)
- (4) **Domain Expansion:** Aplicacao em macroeconomia, economia ambiental, desenvolvimento

#### Longo Prazo (2+ anos):

- (1) **Theoretical Foundations:** Garantias teoricas de convergencia e optimialidade
- (2) **Automated Economic Reasoning:** IA que sugere restricoes baseadas em literatura economica
- (3) **Integration com Policy Frameworks:** Ferramentas end-to-end para analise de impacto regulatorio

### 7.5 Mensagem Final

Tensao entre acuracia preditiva e interpretabilidade economica nao e inevitavel. Framework de destilacao econometrica demonstra que e possivel:

- Alcançar **97-98% da acuracia** de modelos complexos
- Preservar **interpretabilidade total** via GAMs/Linear
- Garantir **conformidade com teoria economica** (95%+ restricoes)
- Produzir **coeficientes estaveis** para inferencia rigorosa

**Para economistas:** Nao e mais necessario escolher entre ML de ponta e modelos interpretaveis. Economic KD oferece o melhor de ambos mundos.

**Para ML practitioners:** Incorporar conhecimento de dominio (restricoes economicas) melhora nao apenas interpretabilidade, mas tambem generalizacao e robustez.

**Para reguladores e policy makers:** Modelos destilados fornecem evidencia quantitativa acurada E auditavel, permitindo decisoes informadas sem “caixa-preta”.

Framework abre caminho para nova geracao de modelos economicos: *data-driven, teoricamente fundamentados, e praticamente uteis.*

### 7.6 Disponibilidade

- **Código:** Framework integrado ao DeepBridge (open-source)
  - Repositorio: [github.com/deepbridge/deepbridge](https://github.com/deepbridge/deepbridge)
  - Documentacao: [docs.deepbridge.ai/economics](https://docs.deepbridge.ai/economics)
- **Reproducibilidade:** Scripts completos dos case studies

- Dataset (anonimizado): Disponivel mediante requisicao
- Jupyter notebooks: Exemplos passo-a-passos
- **Tutorial:** Guia pratico para economistas
  - Especificacao de restricoes economicas
  - Interpretacao de resultados de destilacao
  - Analise de estabilidade e quebras estruturais

Framework de destilacao econometrica representa passo concreto em direcao a **economia data-driven** que preserva rigor teorico e accountability social. Esperamos que inspire novas pesquisas na intersecao de ML, econometria, e policy analysis.

### REFERÊNCIAS