# PROJETO DE CIÊNCIA DE DADOS

André Noronha
Igor Felippe Muzel

# OBJETIVO GERAL

- Analisar dados relacionados a planos de saúde e pacientes
- Explorar padrões e relações entre variáveis
- Construir modelos preditivos para classificação e regressão

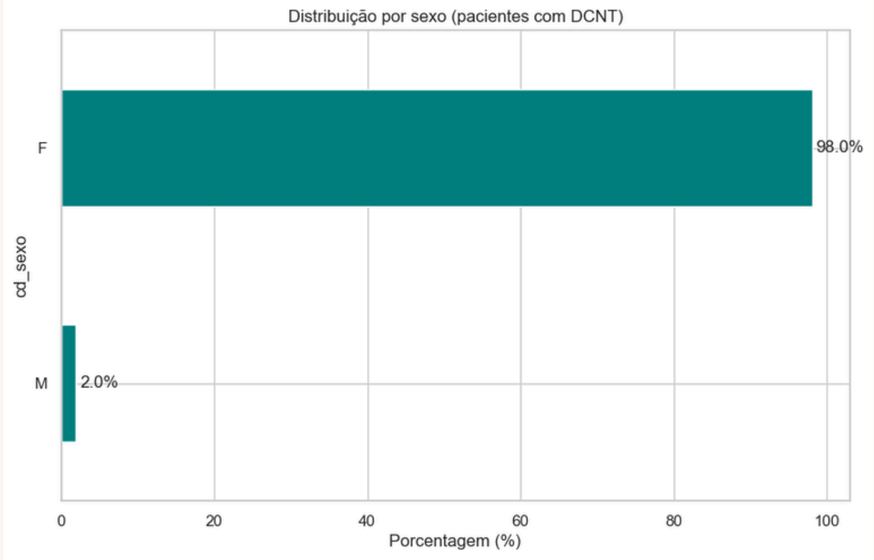
# METODOLOGIA GERAL

- Pré-processamento e unificação dos dados
- Análise exploratória com visualizações gráficas
- Modelagem preditiva com classificação (LogReg, RF, LightGBM)
- Modelagem de regressão (ElasticNet, CatBoost, XGBoost)
- Avaliação com métricas: Accuracy, F1, $R^2$, MAE, RMSE

# ANÁLISE INICIAL

- Dataset com mais de 12 milhões de linhas
- Variáveis: IMC, faixa etária, plano, sexo, linha de cuidado...
- Foco em estruturação e filtragem dos dados
- Primeira Observação: Grande quantidade de gestantes.

Figura 1: Distribuição por sexo (pacientes com DCNT)

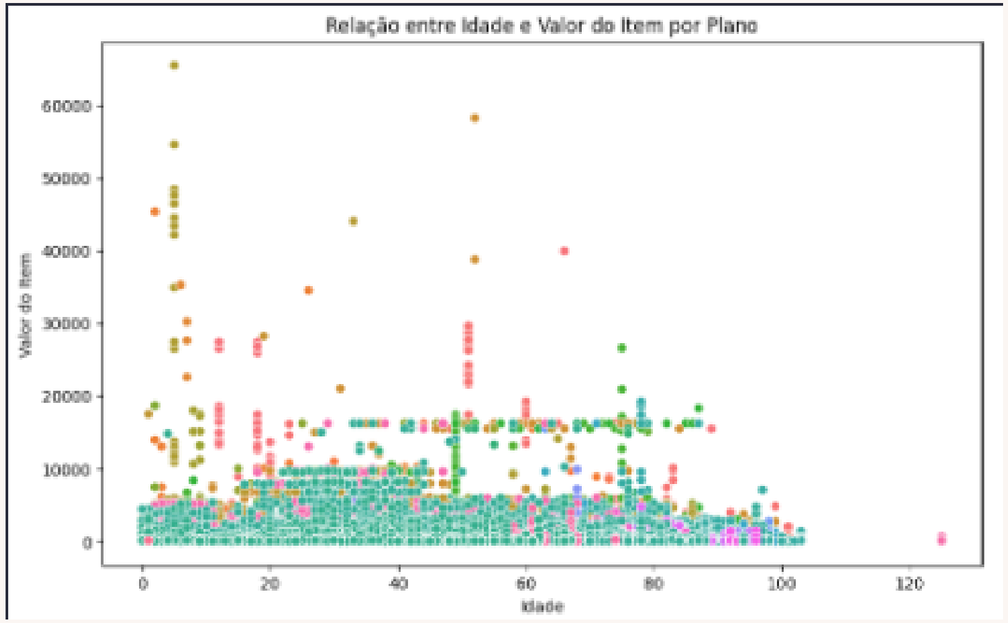



Fonte: Autoria Própria (2025)

# ANÁLISE EXPLORATÓRIA

- Tratamento de outliers (boxplot)
- Correlação de variáveis
- Observação: Concentração nos planos Individuais familiares e UNIFAMILIA Empresarial

Figura 2: Relação entre Idade e Valor do Item por Plano




Fonte: Autoria Própria (2025)

# DCNT

- Prever DCNT
- Criação da variável alvo: doenca_cronica
- Palavras chave na coluna ds_cid
- Verifica a coluna faixa_imc
- doenca_cronica recebe o valor 1(crônico) e 0 (não crônico)

Figura 3: Criação da variável target

```
# Criar a variável alvo 'doenca_cronica'
cronico_por_cid = dcnt['ds_cid'].str.contains('diabetes|hipertens|obesidade', case=False, na=False)
cronico_por_imc = dcnt['faixa_imc'].str.contains('Obesidade', case=False, na=False)
dcnt['doenca_cronica'] = (cronico_por_cid | cronico_por_imc).astype(int)
```

Fonte: Autoria Própria (2025)

# PRÉ-PROCESSAMENTO(DCNT)

- Prepara os dados para que o algoritmo possam “entendê-los”
- Separação de X e Y
- ColumnTransformer
- train_test_split

Figura 4: Features e variável alvo

```
# Definir Features (X) e Alvo (y)
features = ['qt_altura_cm_x', 'faixa_etaria', 'plano_agrupado', 'sexo_legivel', 'ds_linha_cuidado']
X = dcnt[features]
y = dcnt['doenca_cronica']
```

Fonte: Autoria Própria (2025)

Figura 5: Pré-processamento

```
# Definir o pré-processador

faixa_etaria_cats = sorted(list(X['faixa_etaria'].unique()))
# faixa_imc_cats não é mais necessário aqui
ordinal_features = ['faixa_etaria'] # Apenas faixa_etaria agora é ordinal
nominal_features = ['plano_agrupado', 'sexo_legivel', 'ds_linha_cuidado']

preprocessor = ColumnTransformer(
    transformers=[
        ('ord', OrdinalEncoder(categories=[faixa_etaria_cats]), ordinal_features),
        ('cat', OneHotEncoder(handle_unknown='ignore'), nominal_features)
    ],
    remainder='passthrough'
)

# Dividir os dados com estratificação
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42, stratify=y)
[105]
```

Fonte: Autoria Própria (2025)

# REGRESSÃO LOGÍSTICA

- Motivo
- Recall: 0.51
- Precision: 0.60
- F1-Score: 0.55
- Accuracy: 0.59

Figura 6: Relatório de Classificação Regressão Logística

Relatório de Classificação Regressão:

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.58 | 0.67 | 0.62 | 63993 |
| 1 | 0.60 | 0.51 | 0.55 | 62675 |
| accuracy | | | 0.59 | 126668 |
| macro avg | 0.59 | 0.59 | 0.59 | 126668 |
| weighted avg | 0.59 | 0.59 | 0.59 | 126668 |

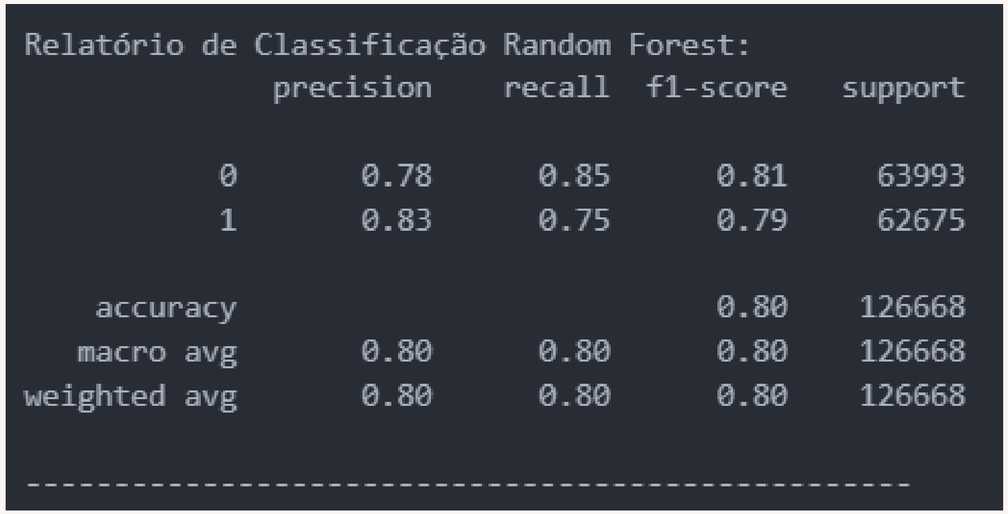
---

Fonte: Autoria Própria (2025)

# RANDOM FOREST

- Motivo
- Recall: 0.75
- Precision: 0.83
- F1-Score: 0.79
- Accuracy: 0.80

Figura 7: Relatório de Classificação Random Forest



Relatório de Classificação Random Forest:

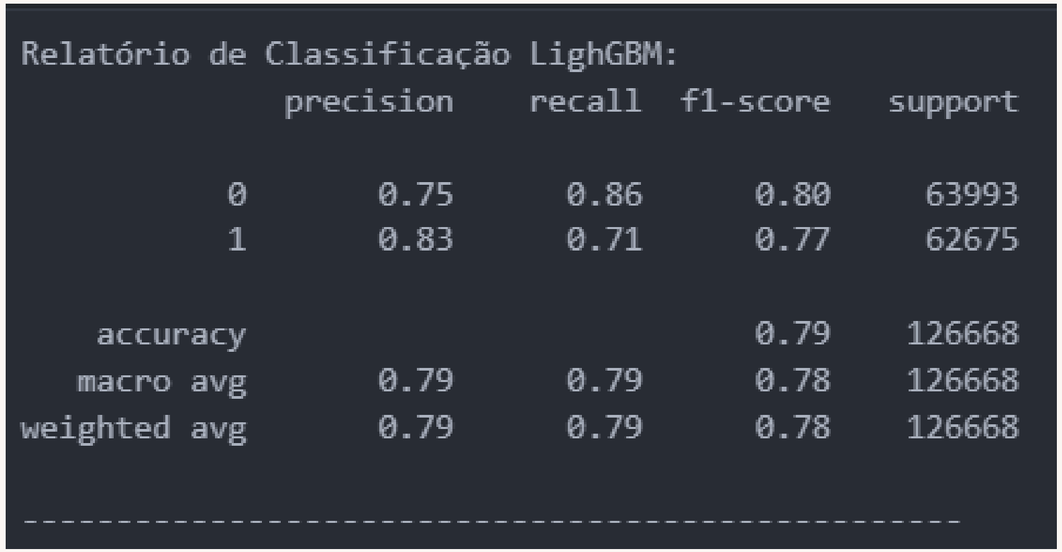
| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.78 | 0.85 | 0.81 | 63993 |
| 1 | 0.83 | 0.75 | 0.79 | 62675 |
| accuracy | | | 0.80 | 126668 |
| macro avg | 0.80 | 0.80 | 0.80 | 126668 |
| weighted avg | 0.80 | 0.80 | 0.80 | 126668 |

Fonte: Autoria Própria (2025)

# LIGHTGBM

- Motivo
- Recall: 0.71
- Precision: 0.83
- F1-Score: 0.77
- Accuracy: 0.79

Figura 8: Relatório de Classificação LightGBM



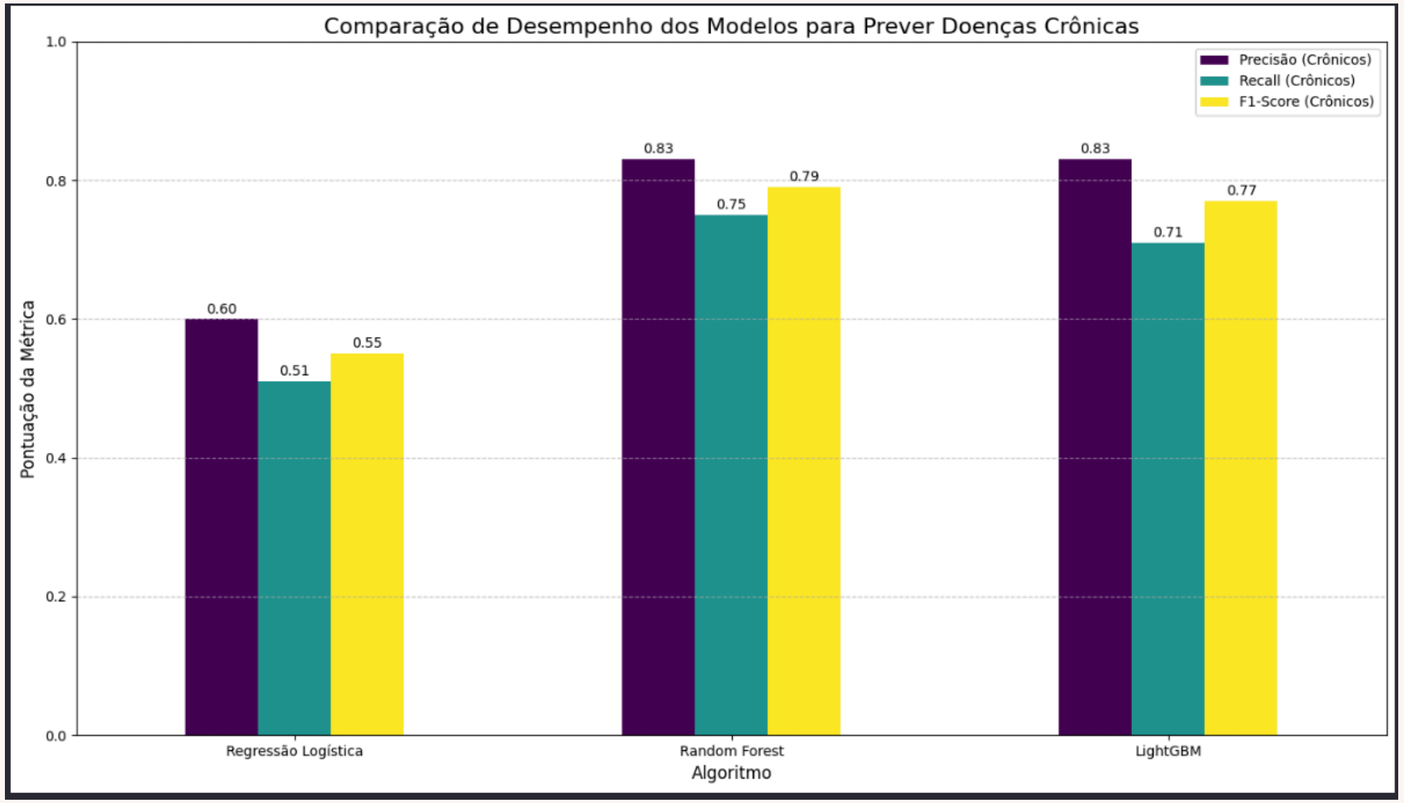
Relatório de Classificação LighGBM:

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.75 | 0.86 | 0.80 | 63993 |
| 1 | 0.83 | 0.71 | 0.77 | 62675 |
| accuracy | | | 0.79 | 126668 |
| macro avg | 0.79 | 0.79 | 0.78 | 126668 |
| weighted avg | 0.79 | 0.79 | 0.78 | 126668 |

Fonte: Autoria Própria (2025)

# RESULTADOS

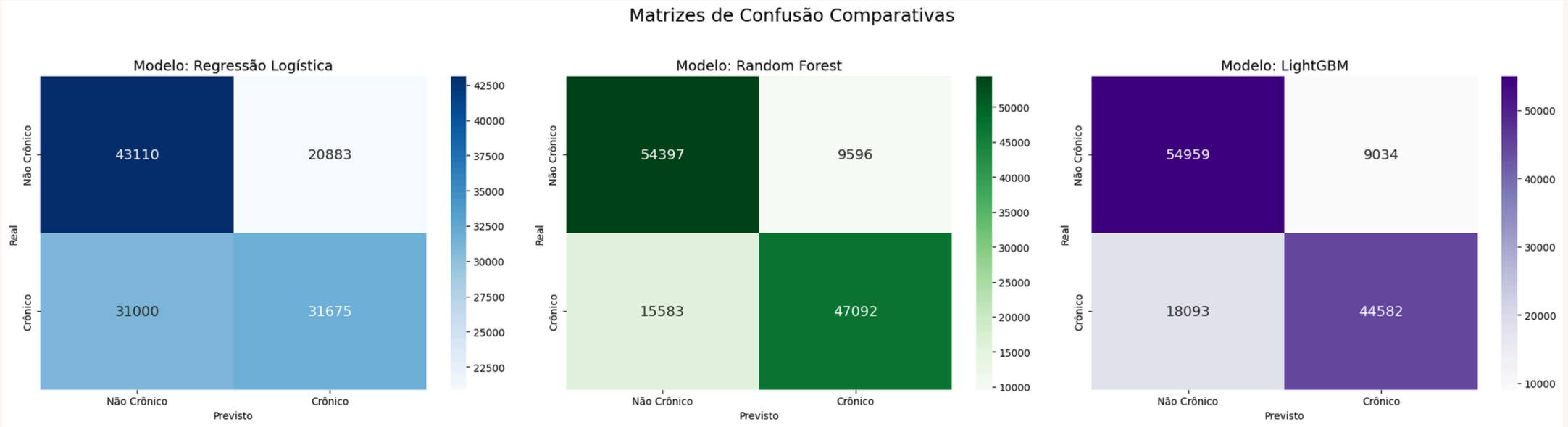
Figura 9: Modelo para Prever as DCNT’s




Fonte: Autoria Própria (2025)

# CONCLUSÃO

Figura 10: Matriz de confusão




Fonte: Autoria Própria (2025)

# MODELAGEM REGRESSIVA – CUSTO

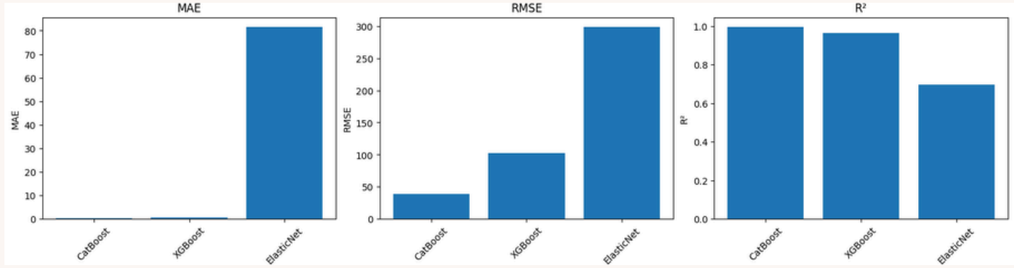
- Criação da variável target “custo_evitado”
- Divisão treino e teste 80/20
- CatBoost - MAE = 0.42 | RMSE = 38 | $R^2$ = 0.99
- XGBoost - MAE = 0.64 | RMSE = 102 | $R^2$ = 0.96
- ElasticNet - MAE = 81 | RMSE = 298 | $R^2$ = 0.69
- Melhor modelo: CatBoost com ajuste quase perfeito

Figura 11: Criação da variável target

```python
# Calcular custo real a partir de 'faixa_gasto'
custo["custo_real"] = custo["faixa_gasto"].apply(midpoint).astype(float)

# Calcular valor médio por item e custo projetado
custo["vl_medio_item"] = custo["faixa_vl_item"].apply(midpoint).astype(float)
custo["custo_projetado"] = custo["vl_medio_item"] * custo["qt_item"]

# Gerar coluna 'custo_evitado'
custo["custo_evitado"] = custo["custo_projetado"] - custo["custo_real"]

# Evitar valores negativos
custo["custo_evitado"] = custo["custo_evitado"].clip(lower=0)
```

Fonte: Autoria Própria (2025)

Figura 12: Comparação de Métricas

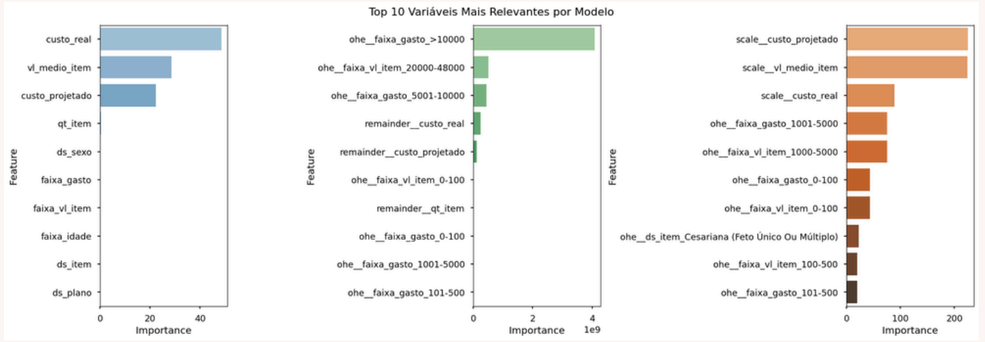



Fonte: Autoria Própria (2025)

# MODELAGEM REGRESSIVA – CUSTO

- Criação da variável target “custo_evitado”
- CatBoost - Principais variáveis: custo_real, vl_medio_item e custo_projetado
- XGBoost - Principais variáveis: ohe__faixa_gasto_>10000 e faixa_vl_item_20000-48000]
- ElasticNet - Principais variáveis: scale__custo_projetado, scale__vl_medio_item, scale__custo_real
- Melhor modelo: CatBoost com ajuste quase perfeito

Figura 13: Feature Importance



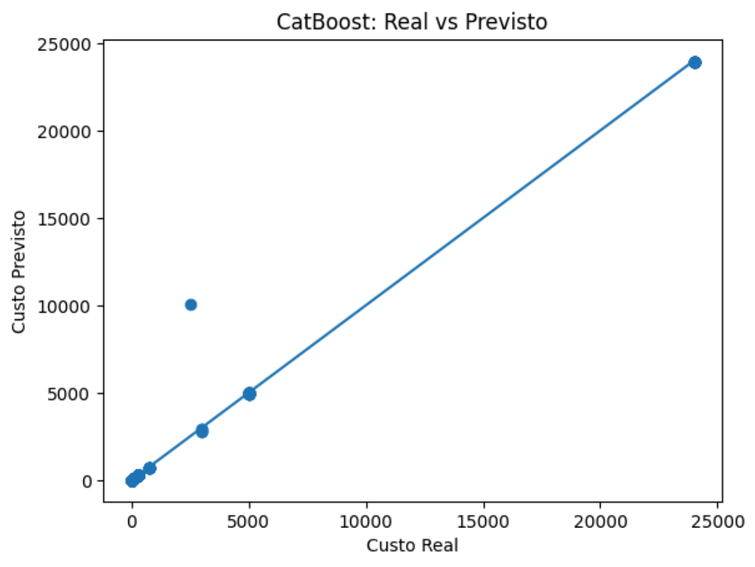

Fonte: Autoria Própria (2025)

# MODELAGEM REGRESSIVA – CUSTO

Desempenho da predição do CatBoost:

- A maioria dos pontos está bem próxima da linha ideal.
- O modelo teve bom desempenho geral, com exceção de um ou outro desvio pontual.

Figura 14: CatBoost: Real vs Previsto



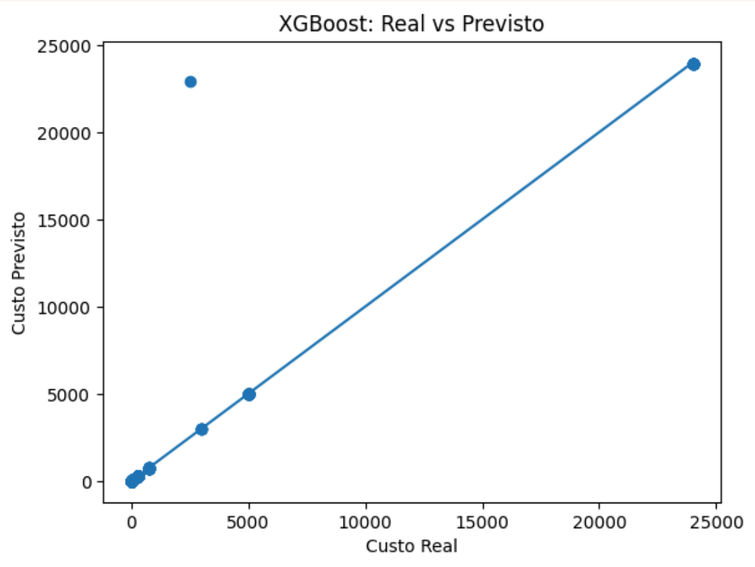

Fonte: Autoria Própria (2025)

# MODELAGEM REGRESSIVA – CUSTO

Desempenho da predição do XGBoost:

- Bom desempenho, com os pontos em geral próximos da linha ideal.
- Bom desempenho geral, mas com um erro grave em pelo menos uma amostra, indicando maior sensibilidade a outliers ou overfitting.

Figura 15: XGBoost: Real vs Previsto



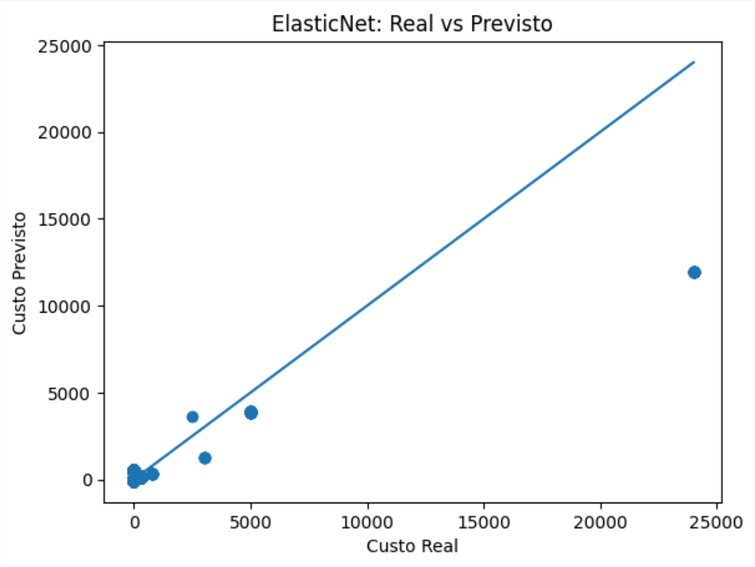

Fonte: Autoria Própria (2025)

# MODELAGEM REGRESSIVA – CUSTO

Desempenho da predição do ElasticNet:

- Muitos pontos estão abaixo da linha ideal, ou seja, subestimando o custo real.
- O modelo tem dificuldade clara em prever valores altos
- ElasticNet está com desempenho inferior, especialmente em casos com custo elevado. Pode ser uma limitação do modelo linear em capturar relações mais complexas nos dados.

Figura 16: ElasticNet: Real vs Previsto

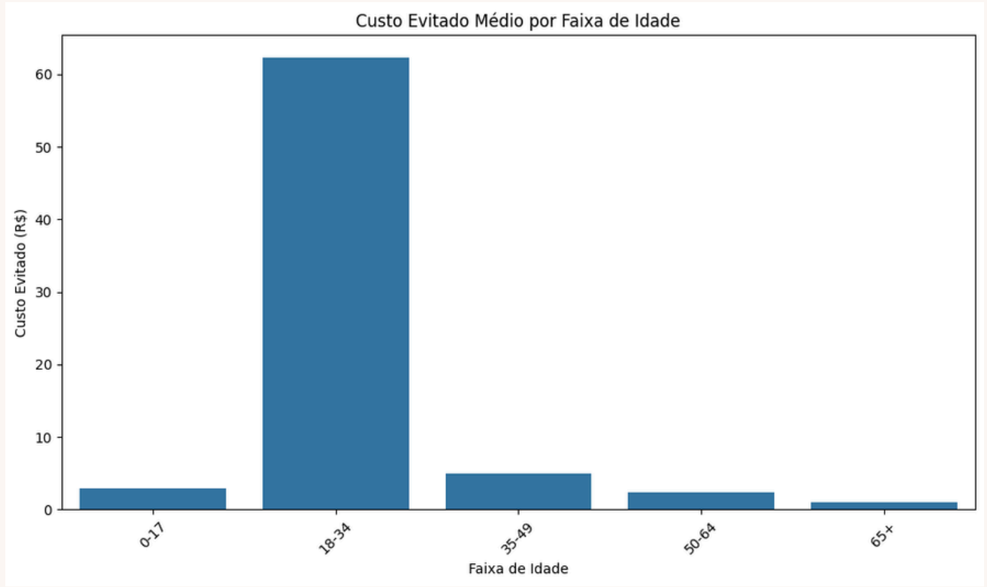



Fonte: Autoria Própria (2025)

# CONCLUSÕES – CUSTO EVITADO

- Faixa 18-34 anos apresentou o maior custo evitado médio (~R$ 62), destacando-se fortemente entre os grupos.
- O grupo de jovens adultos se beneficia mais de intervenções que evitam custos, talvez por aderirem mais a programas de prevenção ou por terem maior risco de doenças evitáveis nessa fase.
- Idosos (65+) têm o menor custo evitado, o que pode indicar menor efetividade preventiva nessa população ou dificuldade de acesso/adesão.

Figura 17: Custo Evitado Médio por Faixa de Idade

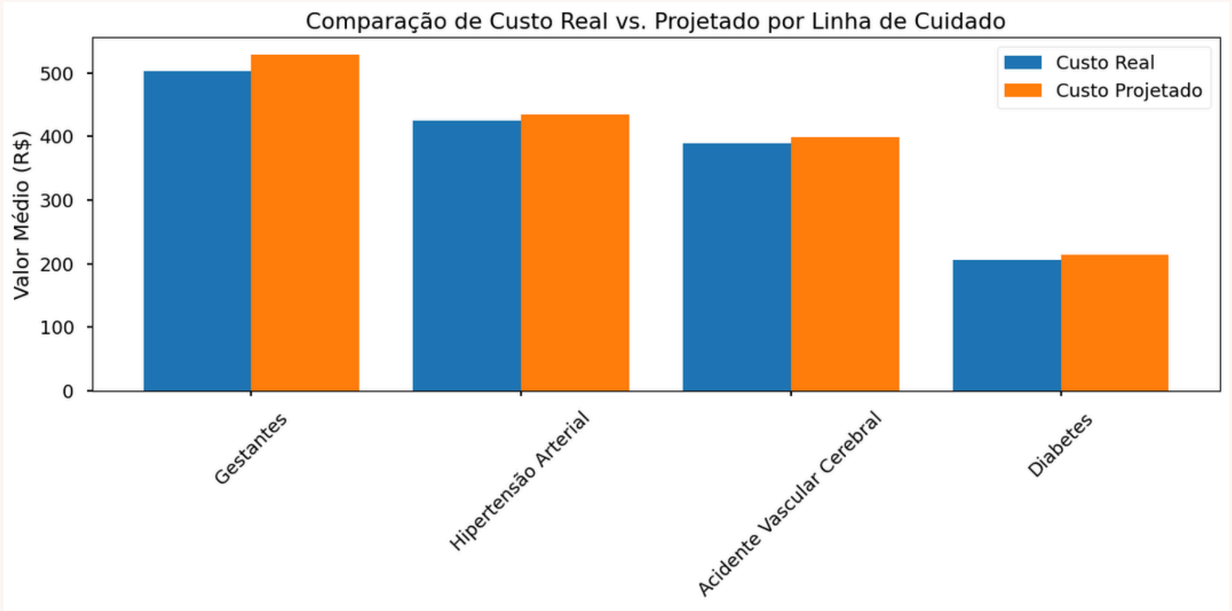



Fonte: Autoria Própria (2025)

# CONCLUSÕES – CUSTO EVITADO

- Em todas as linhas, o custo projetado é maior do que o custo real.
- O impacto mais significativo é observado na linha das Gestantes.
- A menor diferença é em Diabetes, indicando ganho mais modesto.

Figura 18: Custo Evitado por LC

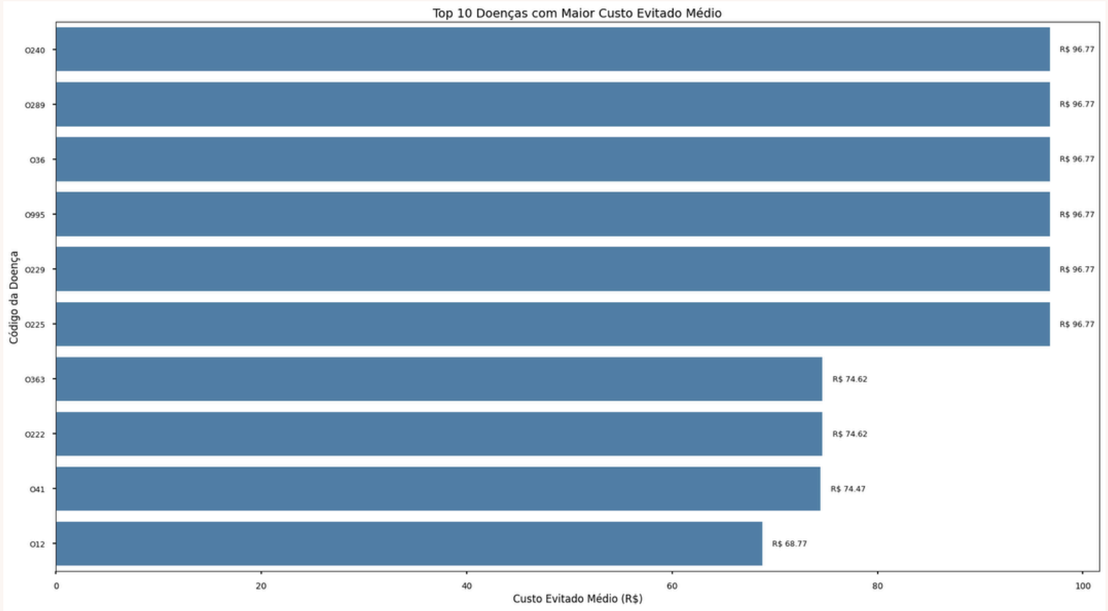



Fonte: Autoria Própria (2025)

# CONCLUSÕES – CUSTO EVITADO

- A maioria dos CIDs com maior custo evitado parece estar relacionada a gestação/parto.
- Isso confirma os achados anteriores: intervenções na linha de cuidado Gestantes são altamente efetivas em termos de economia.

Figura 19: Top 10 doenças com maior custo evitado




Fonte: Autoria Própria (2025)

# CONCLUSÃO FINAL – CUSTO EVITADO

| Área | Insight principal |
| --- | --- |
| Linha de Cuidado | Gestantes lideram em custo evitado médio e impacto real vs. previsto |
| Faixa Etária | Adultos jovens (18-34) têm maior benefício médio das ações. |
| Comparação de custos | Em todas as linhas há economia, mas varia em intensidade. |
| Doenças com maior economia | Doenças obstétricas dominam o top 10 em custo evitado médio. |

# Referências


COSTA, Felipe. **Prevendo números: entendendo métricas de regressão.** Medium, 28 set. 2019. Disponível em: https://medium.com/data-hackers/prevendo-n%C3%BAmeros-entendendo-m%C3%A9tricas-de-regress%C3%A3o-35545e011e70. Acesso em: 2 jul. 2025.

DATACAMP. **CatBoost Tutorial: A Machine Learning Library for Categorical Data**. Disponível em: https://www.datacamp.com/tutorial/catboost. Acesso em: 2 jul. 2025.

DATACAMP. **XGBoost in Python: A Practical Guide**. Disponível em: https://www.datacamp.com/tutorial/xgboost-in-python. Acesso em: 2 jul. 2025.

RISWANTO, Ujang. **Step-by-Step Guide to Implementing Elastic Net Regression in Python**. Medium, 2023. Disponível em: https://ujangriswanto08.medium.com/step-by-step-guide-to-implementing-elastic-net-regression-in-python-eff1757aad0a. Acesso em: 2 jul. 2025.