**Lucas Meller**

Predição de Doença Cardíaca

**Data de Apresentação**

# Introdução do Problema

Com o aumento do sedentarismo e consumo de alimentos não saudáveis, ocorrências de doenças cardíacas estão em ascensão prejudicando a saúde e longevidade daqueles que praticam tais hábitos. Com isso, a equipe decidiu desenvolver um sistema classificatório que fizesse a predição de quais pacientes iriam ter doenças cardíacas por meio de métricas de avaliação de saúde.

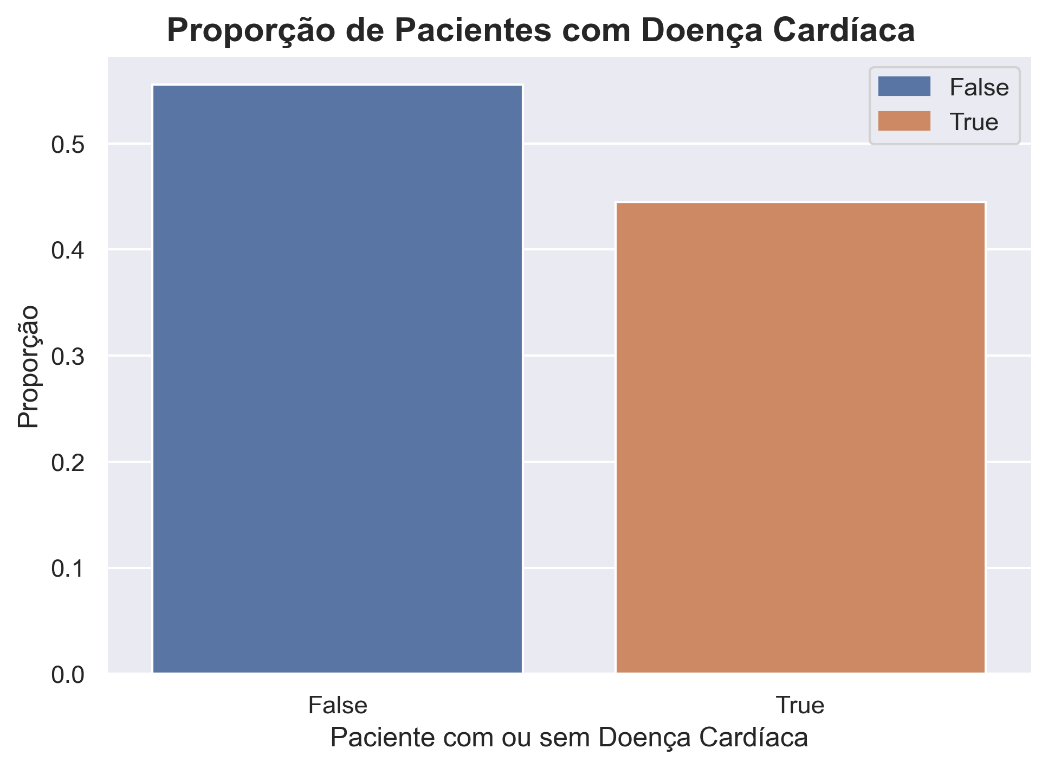
Para tal, foi escolhido um dataset da *University of California Irvine’s ML Repository* com as seguintes informações:

|  |  |
| --- | --- |
| * Idade * Sexo * Tipo de Dor no Peito * Pressão Arterial * Colesterol * Glicose em Jejum acima de 120 * Resultados EKG | * Frequência Cardíaca Máxima * Angina Induzida por Exercício * Depressão ST * Inclinação de ST * Número de Vasos Fluoroscópicos * Talio * Doença Cardíaca (target) |

Para avaliação de performance dos modelos, a métrica *recall* foi a escolhida já que busca minimizar os falsos negativos e, desse forma, evitar que alguém com indícios de problemas cardíacos seja diagnosticado como saudável e mitigar a chance de descoberta de última hora ?

# Análise do Dataset

Foram desenvolvidos gráficos para análise de quais variáveis possuíam correlação com o target e, por conseguinte, avaliar quais modelos seriam mais adequados aos dados. No conjunto de dados escolhido, os dados já estavam tratados e com o target balanceado.



*Figura 1. Proporção variável target*

Para avaliação de distribuição das variáveis contínuas com relação ao target, foram plotadas as seguintes análises:

Gráfico, Gráfico de linhas

Descrição gerada automaticamente

Figura . Relação Variáveis Independentes Contínuas vs Target

Nota-se que a distribuição do target com relação as variáveis Idade, Frequência Cardíaca Máxima e Depressão ST diferem entre pacientes com ou sem doença, significando dessa forma uma possível capacidade de diferenciação de futuros clientes usando tais colunas.

Além disso, foram criados gráficos para análise das relações entre target e variáveis discretas para as colunas Sexo, Tipo de Dor no Peito, Glicose em Jejum Acima de 120, Resultados EKG, Angina Induzida por Exercício, Inclinação ST, Número de Vasos Fluoroscópicos e Talio.

Gráfico, Gráfico de barras

Descrição gerada automaticamente

Figura . Relação Variáveis Independentes Discretas vs Target

De acordo com o gráfico anterior, nos gráficos relacionados ao Talio, Tipo de Dor no Peito e Número de Vasos Fluoroscópicos há uma separação entre clientes com e sem doença cardíaca e, consequentemente, tornando possível a classificação de novos clientes usando essas colunas como será visto posteriormente no modelo RandomForestClassifier.

Para verificação das relações entre todas as variáveis presentes no dataset foi utilizada a matriz de correlação para verificação de tais métricas.

Gráfico, Gráfico de mapa de árvore

Descrição gerada automaticamente

Figura . Matriz de Correlação

Por fim, decidiu-se usar a separação Train-Test-Val no qual o subgrupo de teste foi usado para calibração de hiper parâmetros e o de validação para verificação de performance.

# Seleção de features e feature Engineering

Para seleção de feature, decidiu-se usar todas as colunas já que não havia dados nulos e que iriam atrapalhar para criação dos modelos. Já para feature engineering, foi usado pipelines com polynomial features para criação de novos colunas e standard scaler para padronização de média e desvio padrão, evitando que variáveis interferissem outras.

Texto

Descrição gerada automaticamente

Figura . Colunas antes do StandardScaler

Interface gráfica do usuário, Texto, Aplicativo

Descrição gerada automaticamente

Figura . Colunas após StandardScaler

# Treinamento e Fine Tuning (Grid Search obrigatório)

Para o treinamento, foi usado um pipeline com PolynomialFeatures, StandardScaler, PCA e modelo para padronização de procedimentos.

Texto

Descrição gerada automaticamente

Figura . Exemplo de treinamento de modelo

Tela de computador com texto preto sobre fundo branco

Descrição gerada automaticamente

Figura . Parâmetros utilizados nos modelos

Como visto acima, foram testados 5 modelos (LogisticRegression, SVC, RandomForest, KNN e CatBoost) com intervalos de parâmetros específicos a cada um. Para cada intervalor, o GridSearch testa valores e retorna o melhor modelo dada a métrica escolhida, sendo nesse caso o recall. Optou-se pelo RandomizedSearchCV já que usa amostragens representativas dos parâmetros mesmo sem testar todos do intervalo, aumentando de forma significativa o tempo de treinamento.

De forma inicial, todos os modelos foram testados apenas com o StandardScaler como pré-processamento dos dados para definição de uma baseline de performance de cada modelo.

Gráfico, Gráfico de barras

Descrição gerada automaticamente

Figura . Performance dos Modelos apenas com StandardScaler

Após isso, aplicou-se PolynomialFeatures de grau 3 em cada pipeline, resultando na seguinte performance:

Gráfico, Gráfico de barras

Descrição gerada automaticamente

Figura . Performance dos Modelos com StandardScaler e PolynomialFeatures

Por fim, foi aplicado PCA em cada pipeline com 30 componentes, resultando na seguinte performance:

Gráfico, Gráfico de barras

Descrição gerada automaticamente

Figura . Performance dos Modelos com PolynomialFeatures, PCA e StandardScaler

Após os testes citados, chegou-se à conclusão de que o melhor modelo treinado foi o RandomForestClassifier com PolynomialFeatures e StandardScaler (foi mantido por padronização de pipelines, mas para modelos de árvores classificatórios a padronização da distribuição dos dados não faz diferença).

Para melhor compreensão de quais variáveis foram importantes para o modelo escolhido, foi plotado gráficos com e sem PolynomialFeatures para compreensão das escolhas tomadas pelo modelo para redução da entropia presente.

Gráfico, Gráfico de barras

Descrição gerada automaticamente

Figura . Importância das Features para modelo - Sem PolynomialFeatures

# Gráfico, Gráfico de barras Descrição gerada automaticamente

Figura . Importância das Features para o modelo - Com PolynomialFeatures

# Avaliação

Como foi feita a avaliação do modelo, qual o resultado final. Gráfico e explicativo.

# Modelo em Produção

Como foi feito o uso do modelo. Mostrando ele em funcionamento. Pode ser uma API, pode ser um app, pode ser uma aplicação terminal. Não pode ser um jupyter notebook

# Conclusão

Aqui vai a conclusão do relatório.