Predição de Ataques Cardíacos usando Aprendizado de Máquina

Pedro de Almeida Marim
Thiago Peres Casagrande

12 de julho de 2024

1 Introdução

A aplicação de modelos de aprendizado de máquina (AM) em cenários reais e de fundamental importância à vida humana tem mostrado resultados extremamente promissores, incluindo a área da saúde. Assim, este projeto buscou utilizar, sob a premissa de resolver um problema de classificação, técnicas de aprendizado de máquina para a elaboração de modelos sobre um conjunto de dados que trata de ataques cardíacos.

Além disso, as funções loss foram explicitadas, e um teste de acurácia foi realizado sobre cada modelo para possibilitar a comparação de modelos distintos.

2 O Conjunto de Dados

O conjunto de dados, obtido na plataforma Kaggle [1], trata do risco de ataque cardíaco de um indivíduo a partir de um vetor de variáveis. São elas: idade, gênero, presença de angina, número de vasos principais (1-3), tipo de dor no peito, pressão sanguínea, nível de colesterol, glicose no sangue, resultados de eletrocardiograma e máximo de batimentos cardíacos obtidos. Ainda, sua variável **target** é binária e é denominada risco de ataque cardíaco: se assume valor 1, o risco de ataque cardíaco é alto; se assume valor 0, o risco é baixo.

3 Preparação do Conjunto de Dados

A primeira medida realizada antes da criação e treinamento de cada modelo foi a normalização de cada variável. Tal medida é importante para que as features mantenham-se na mesma escala, evitando assim que um valor de uma feature em escala diferente das demais altere o treinamento de algum modelo. Além disso, como será elaborado posteriormente, a função kernel, utilizada na máquina de

[1] https://www.kaggle.com/datasets/rashikrahmanpritom/heart-attack-analysis-prediction-dataset

suporte de vetores, pode ser sensível a diferentes escalas, por poder utilizar a métrica euclidiana; a normalização garante que a função kernel mais adequada seja utilizada.

Além disso, o conjunto de dados é dividido em subconjuntos de treino e teste, de forma que o conjunto de teste representa 30% do conjunto original.

4 Resultados obtidos

A implementação do projeto foi feita em Python e o notebook com os códigos foi enviado em conjunto com esse pdf.

Para resolver esse problema de classificação binária forem implementos os seguintes modelos de aprendizado: Regressão Logística, Rede Neural, Árvore de Decisão e Maquina de Vetores de Suporte.

A seguir será mostrado os resultados e métricas obtidos com cada um desses modelos.

4.1 Regressão Logística

A regressão logística é uma técnica estatística usada para modelar a probabilidade de uma variável dependente binária, ou seja, um resultado que pode ter apenas duas categorias (como "sim" ou "não", "verdadeiro" ou "falso"). Ao contrário da regressão linear, que prevê valores contínuos, a regressão logística aplica a função logística (ou sigmoide) para garantir que as previsões resultem em probabilidades entre 0 e 1.

Com esse simples modelo obtivemos uma acurácia de 90%.

4.2 Rede Neural

Redes neurais são modelos computacionais inspirados no funcionamento do cérebro humano, utilizados para resolver uma ampla gama de problemas de aprendizado de máquina. Compostas por camadas de neurônios artificiais, as redes neurais podem aprender padrões complexos a partir dos dados. Cada neurônio recebe múltiplas entradas, aplica uma função de ativação e passa o resultado para os neurônios da camada seguinte. Redes neurais profundas (DNNs), que possuem várias camadas ocultas, são particularmente eficazes em tarefas como reconhecimento de imagem, processamento de linguagem natural e previsão de séries temporais. A capacidade de aprender representações hierárquicas dos dados e a flexibilidade para ajustar-se a diferentes tipos de problemas fazem das redes neurais uma ferramenta poderosa na inteligência artificial e no aprendizado profundo.

Nesse projeto foi implementada uma rede neural com as seguintes propriedades: 64 neurônios na primeira camada oculta com função de ativação Relu, 32 neurônios na segunda camada oculta com função de ativação Relu e 1 neurônio na camada de saída (já que estamos num problema de classificação binária) com função de ativação sigmoid.

Além disso, o processo de treinamento foi feito com dados de validação para acompanhar o ocorrimento de overfitting. O gráfico das métricas da função de custo obtidas ao longo das épocas ilustra bem o que foi visto em aula.

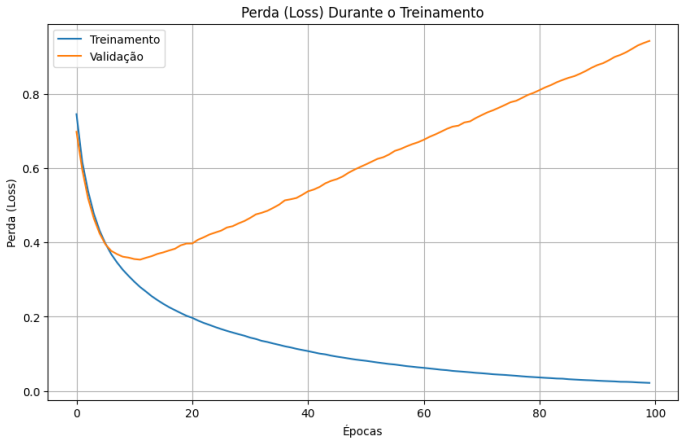

Figure 1: Overfitting durante o treinamento

Da figura, percebemos que a partir da 15º época comece a ocorrer o overfitting, é nessa época que a rede está com os pesos otimizados para generalizar o desempenho. No gráfico a seguir acompanha-se o comportamento das métricas de acurácia, e percebe-se que com os pesos ótimos obtem-se uma acurácia de 90%.

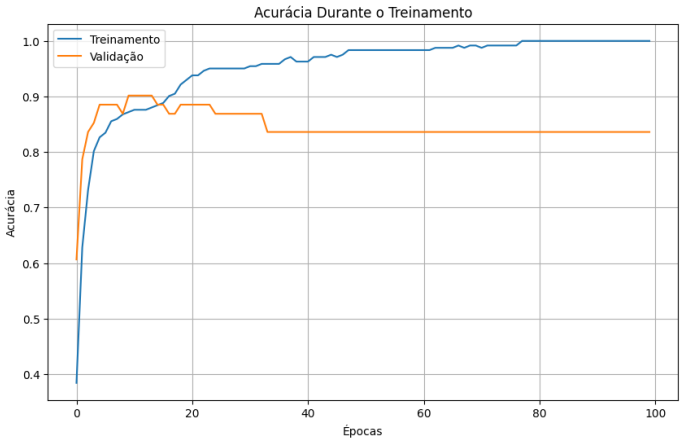

Figure 2: Acurácia durante o treinamento

4.3 Máquina de Suporte de Vetores

A Máquina de Suporte de Vetores (SVM) é um modelo que visa separar duas classes do nosso conjunto de dados; neste caso, pacientes com alto e baixo risco de ataque cardíaco. Para fazê-lo, este busca encontrar um hiperplano de forma a maximizar a margem entre as classes.

Para sua implementação, foi necessário primeiramente avaliar se haveria a necessidade de utilização de uma função kernel. Uma função kernel é uma função, normalmente não linear, utilizada quando nossos dados não são linearmente separáveis. A função é aplicada conjunto de dados, tornando-o, por conseguinte, linearmente separável.

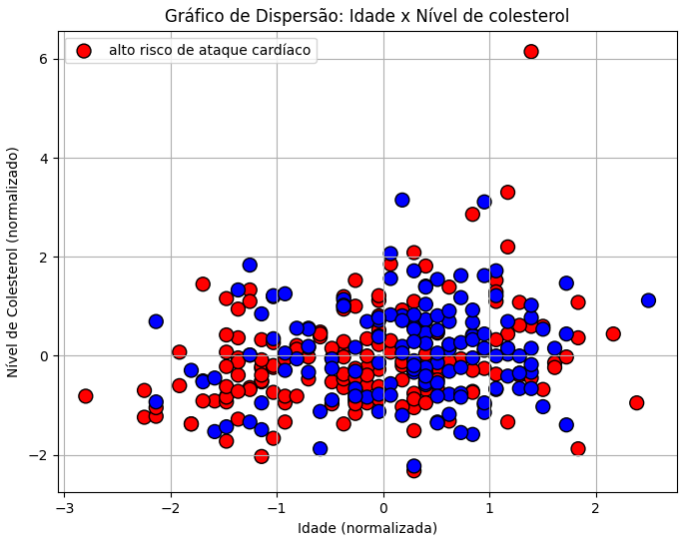

Figure 3: Relação entre variáveis "Idade" e "Nível de Colesterol", ambas normalizadas.

Tão somente para as duas variáveis na imagem acima, nota-se que nosso conjunto de dados não é linearmente separável.

Verificado que nosso conjunto de dados não é linearmente separável, urge a necessidade de utilização de uma função kernel. Para escolher qual seria a mais adequada, foi efetuada uma busca em grade entre este e outros parâmetros, criando uma série de SVMs distintos. Em seguida, eles foram comparados por validação cruzada, e foi escolhido o melhor. Dentre os parâmetros, havia:

- constante de regularização C, que determina o quão ampla será a margem entre o vetor de suporte e o hiperplano;
- tipo de função kernel, variando entre função sigmoid, linear, polinomial e função de base radial;

- e grau do polinômio, para o caso de uma função kernel polinomial.

e grau do polinômio, para o caso de uma função kernel polinomial.

Curiosamente, foi encontrado que o o melhor SVM (de melhor acurácia), muito embora os dados não sejam linearmente separáveis, utilizou uma função kernel linear. Este modelo obteve acurácia de 84%, um valor razoável para um modelo simples.

4.4 Árvore de Decisão

Uma árvore de decisão é um modelo preditivo estruturado em grafo, especificamente em uma árvore. Cada nó interno representa um teste sobre variável (se é maior ou menor que um valor fixo, por exemplo); cada aresta representa o resultado deste teste; por fim, os nós folhas representam uma classe. Neste caso, os nós folhas são alto e baixo risco de ataque cardíaco.

Com este modelo, uma árvore de decisão de 9 nós, foi obtida uma acurácia de 80%.

5 Conclusão

Com este trabalho, foi avaliada a performance de diferentes modelos sobre o mesmo conjunto de dados. Dentre os melhores modelos, isto é, aqueles que obtiveram melhor acurácia, temos a regressão logística e a rede neural. Outros modelos, como SVM e a árvore de decisão não perfomaram tão bem quanto os anteriores.

Vale pontuar, acerca dos modelos com maior acurácia, que alta complexidade do modelo evidentemente não implica em mais acurácia, haja vista que o modelo de regressão logística performou tão bem quanto a rede neural; um exemplo que ilustra o gráfico de curvas erro in e erro out, vista em classe.