Uso de Algoritmos de Classificação para Predição de Novos Casos de Diabetes Mellitus

Pedro Jorge de Souza Colombrino
Matheus Ferreira Amaral Madeira
Guilherme Vieira Rodrigues

31 de Outubro, 2024

Abstract

Este estudo apresenta a aplicação de algoritmos de aprendizado de máquina, com foco em *RandomForestClassifier*, para a previsão de novos casos de diabetes em mulheres. O objetivo principal é demonstrar como treinar, avaliar e implementar esses modelos na predição de uma variável dependente dicotômica (*diabético*) com base em dados clínicos e demográficos. Utilizando validação cruzada (*K-Fold*), identificamos o modelo mais eficaz e avaliamos sua aplicabilidade em cenários reais de saúde pública.

1 Introdução

A capacidade de prever novos casos de diabetes com base em dados clínicos é uma ferramenta valiosa para a saúde pública e medicina personalizada. Diabetes Mellitus é uma doença crônica associada a sérios riscos de complicações, sendo prevalente em mulheres devido a fatores como diabetes gestacional e alterações hormonais.

Com a crescente disponibilidade de dados clínicos, algoritmos de aprendizado de máquina destacam-se pela capacidade de identificar padrões complexos e fazer previsões precisas. Este trabalho explora a aplicação do algoritmo *RandomForestClassifier* para prever diabetes com base em características clínicas e demográficas, demonstrando como os modelos podem ser aplicados na prática clínica.

2 Metodologia

2.1 Conjunto de Dados

Os dados utilizados foram obtidos do Kaggle, com 768 registros de pacientes do sexo feminino, contendo 8 variáveis explicativas e uma variável alvo (*Diabético*) que indica a presença ou ausência de diabetes.

Table 1: Descrição das Variáveis do Conjunto de Dados

Variável	**Descrição**	**Tipo de Dado**
Gravidez	Número de gestações do paciente	Inteiro
Glicose	Concentração de glicose no plasma após teste oral	Inteiro
PressaoSanguinea	Pressão arterial diastólica (mmHg)	Inteiro
EspessuraDaPele	Espessura da dobra cutânea tricipital (mm)	Inteiro
Insulina	Nível sérico de insulina de 2h (mu U/ml)	Inteiro
IMC	Índice de Massa Corporal (peso em kg/(altura em m)^2)	Decimal
DiabetesPedigree	Histórico genético de diabetes	Decimal
Idade	Idade do paciente (anos)	Inteiro
Diabético	Presença de diabetes (1: Sim, 0: Não)	Inteiro

2.2 Modelo: RandomForestClassifier

RandomForestClassifier é uma abordagem baseada em árvore de decisão que usa um conjunto de árvores para melhorar a robustez e a precisão. É adequado para detectar interações não lineares entre variáveis.

2.3 Validação Cruzada com K-Fold

K-Fold A validação cruzada foi usada para avaliar o desempenho do modelo. O conjunto de dados foi dividido em 5 subconjuntos (*folds*), garantindo que todos os dados fossem utilizados para treinamento e teste em diferentes iterações.

- Reduz a possibilidade de *overfitting*.
- fornece métricas de desempenho mais confiáveis.
- garante uma avaliação mais abrangente do modelo.

Para as features da ferramenta *K-Fold*, utilizamos de ténicas heurísticas para uma aproximação prática, onde:

$$n\_splits = \min(5, \lfloor \frac{n}{10} \rfloor) \tag{1}$$

n é o número total de amostras.

Também utilizamos a feature *Shuffle*, que aceita entradas booleanas, definido como *True* e *random_state* definido em 42, garantindo uma grande representatividade dos dados e uma boa reprodutibilidade dos resultados.

2.4 Tratamento de Outliers com a Técnica IQR

Para assegurar a qualidade dos dados utilizados no treinamento do modelo, empregamos a técnica do *Interquartile Range* (IQR) para a identificação e remoção de outliers. Essa técnica baseia-se na dispersão dos dados, calculando o intervalo entre o primeiro quartil ($Q1$) e o terceiro quartil ($Q3$), conhecido como IQR. Valores considerados outliers são aqueles que estão abaixo de $Q1 - k \times IQR$ ou acima de $Q3 + k \times IQR$, onde k é um fator ajustável (neste trabalho, utilizamos $k = 0.9$, ajustado empiricamente para o conjunto de dados).

Ao aplicar essa técnica ao conjunto de dados, identificamos e removemos registros extremos, que representavam cerca de 100 ocorrências antes da filtragem. Após a remoção, apenas dois valores considerados atípicos persistiram, ambos na variável "Idade". Essa redução significativa reforça a eficácia do IQR em criar um conjunto de dados mais limpo e adequado para modelagem.

Os dados finais, sem outliers, foram armazenados no arquivo `diabetes_input.csv`, que serviu como entrada para o modelo.

3 Treinamento e Avaliação

3.1 Hiperparâmetros do RandomForestClassifier

Os hiperparâmetros de *RandomForestClassifier* controlam o comportamento e o desempenho do modelo. Aqui, adotamos uma configuração baseada em avaliação heurística, que é uma aproximação prática do problema real e é particularmente útil quando não há dados suficientes para uma busca exaustiva de hiperparâmetros.

- **Número de árvores (`n_estimators`):** Define o número de árvores na floresta. Um número maior pode melhorar a estabilidade do modelo, mas por outro lado aumenta o custo computacional. A fórmula heurística utilizada é:

$$n\_estimators = 10 \times \sqrt{n}, \tag{2}$$

  onde n é o número total de amostras. Este cálculo fornece um ponto de partida para um equilíbrio entre desempenho e eficiência computacional.

- **Profundidade Máxima (`max_depht`):** Este hiperparâmetro controla a profundidade em que cada árvore pode crescer. Árvores mais profundas tendem a capturar mais detalhes, mas podem levar ao *overfitting*. A profundidade é limitada pela seguinte fórmula:

$$max\_depth = \log_2(n), \tag{3}$$

  onde n é o número total de amostras. Isso permite capturar padrões importantes sem complicar demais o modelo.

- **Número mínimo de amostras para divisão (`min_samples_split`):** Define o número mínimo de amostras necessárias para dividir um nó. Para evitar partições muito pequenas e garantir robustez usamos:

$$min\_samples\_split = \max(2, \frac{n}{100}), \tag{4}$$

  onde n é o número total de amostras.

- **Amostras mínimas por folha (`min_samples_leaf`):** Determina o número mínimo de amostras permitido em uma folha terminal. A heurística utilizada é:

$$min\_samples\_leaf = \max(1, \frac{n}{1000}), \tag{5}$$

  garantindo que cada folha tenha um número mínimo de amostras para conclusões confiáveis.

- **Número máximo de atributos por Divisão (`max_features`):** Define quantos atributos são considerados para encontrar a melhor partição. Para problemas de classificação, as escolhas comuns são:

$$max\_features = \sqrt{m}, \tag{6}$$

  onde m é o número total de atributos no conjunto de dados.

Essas heurísticas fornecem uma base sólida para configurar o modelo antes de realizar ajustes mais avançados, como otimização de hiperparâmetros via *Grid Search* ou *Random Search*. O uso destas avaliações iniciais é amplamente aceito na prática porque proporciona um equilíbrio entre simplicidade e desempenho.

3.2 Métricas de Avaliação

O desempenho foi avaliado com a métrica de *acurácia*, calculada como:

$$\text{Acurácia} = \frac{\text{Número de Previsões Corretas}}{\text{Número Total de Previsões}}$$

4 Resultados

Os resultados da validação cruzada mostram que a precisão média por fold é a seguinte:

- Precisão do modelo da 1ª fold: 0,74
- Precisão do modelo da 2ª fold: 0,74
- Precisão do modelo na 3ª fold: 0,78
- Precisão do modelo de 4ª fold: 0,79
- Precisão do modelo de 5ª fold: 0,74

A melhor precisão do modelo salvo foi de 0,79. Estes resultados demonstram que o modelo *RandomForestClassifier* apresenta desempenho consistente e robusto com precisão média geral satisfatória. A validação cruzada confirmou a estabilidade do modelo, com 79% de acurácia.

5 Aplicação Prática

Após treinar o modelo, use a biblioteca `pickle` para salvar o modelo e depois carregá-lo para prever novos casos. Aqui está um exemplo:

```
import pickle

nova_linha = [[5, 176, 72, 17, 24.6, 0.387, 34]]
# Carregar o modelo salvo
with open("melhor_modelo_random_forest.pkl", "rb") as f:
    modelo = pickle.load(f)
# Previsão
predicao = modelo.predict(nova_linha)
```

O resultado indica se o paciente é diabético (`1`) ou não (`0`).

O uso da biblioteca `pickle` para salvar o modelo treinado permite que ele seja carregado em qualquer máquina sem a necessidade de reprocessar os dados ou reler o CSV original. Isso facilita a implementação do modelo em ambientes de produção, onde previsões precisam ser feitas rapidamente e com eficiência. Basta carregar o arquivo do modelo salvo e utilizá-lo para prever novos casos, garantindo que o processo seja ágil e sem a sobrecarga computacional de treinar o modelo novamente.

6 Conclusão

O *RandomForestClassifier* provou ser uma ferramenta poderosa para classificação, devido à sua robustez e capacidade de manejar dados complexos com múltiplas variáveis explicativas. Sua abordagem baseada em múltiplas árvores decisórias permite capturar interações não lineares entre os atributos, tornando-o uma escolha ideal para problemas de classificação com estruturas de dados variadas.

Neste trabalho, o *RandomForestClassifier* demonstrou eficiência ao alcançar uma acurácia média consistente durante a validação cruzada, evidenciando sua estabilidade e aplicabilidade em contextos reais. A capacidade de ajustar hiperparâmetros como o número de árvores (`n_estimators`) e a profundidade máxima das árvores (`max_depth`) permite ao modelo um alto grau de personalização para diferentes conjuntos de dados, sem perder sua característica de generalização.

Embora o conjunto de dados de diabetes tenha servido como um estudo de caso, a flexibilidade do *RandomForestClassifier* torna-o aplicável a diversos domínios, como detecção de fraudes, análise de crédito e diagnósticos médicos, entre outros. Seu desempenho robusto, aliado à facilidade de implementação e ao suporte nativo para salvar modelos com bibliotecas como `pickle`, faz dele uma solução prática para aplicações em produção.

Futuras extensões deste trabalho podem incluir o uso de técnicas avançadas de otimização de hiperparâmetros, como *Grid Search* e *Bayesian Optimization*, e a comparação com outros algoritmos baseados em árvores, como *Gradient Boosting* e *XGBoost*, para aprofundar a análise de desempenho. Dessa forma, este estudo reforça a importância do *RandomForestClassifier* como uma ferramenta versátil e eficaz em projetos de aprendizado de máquina voltados à classificação.