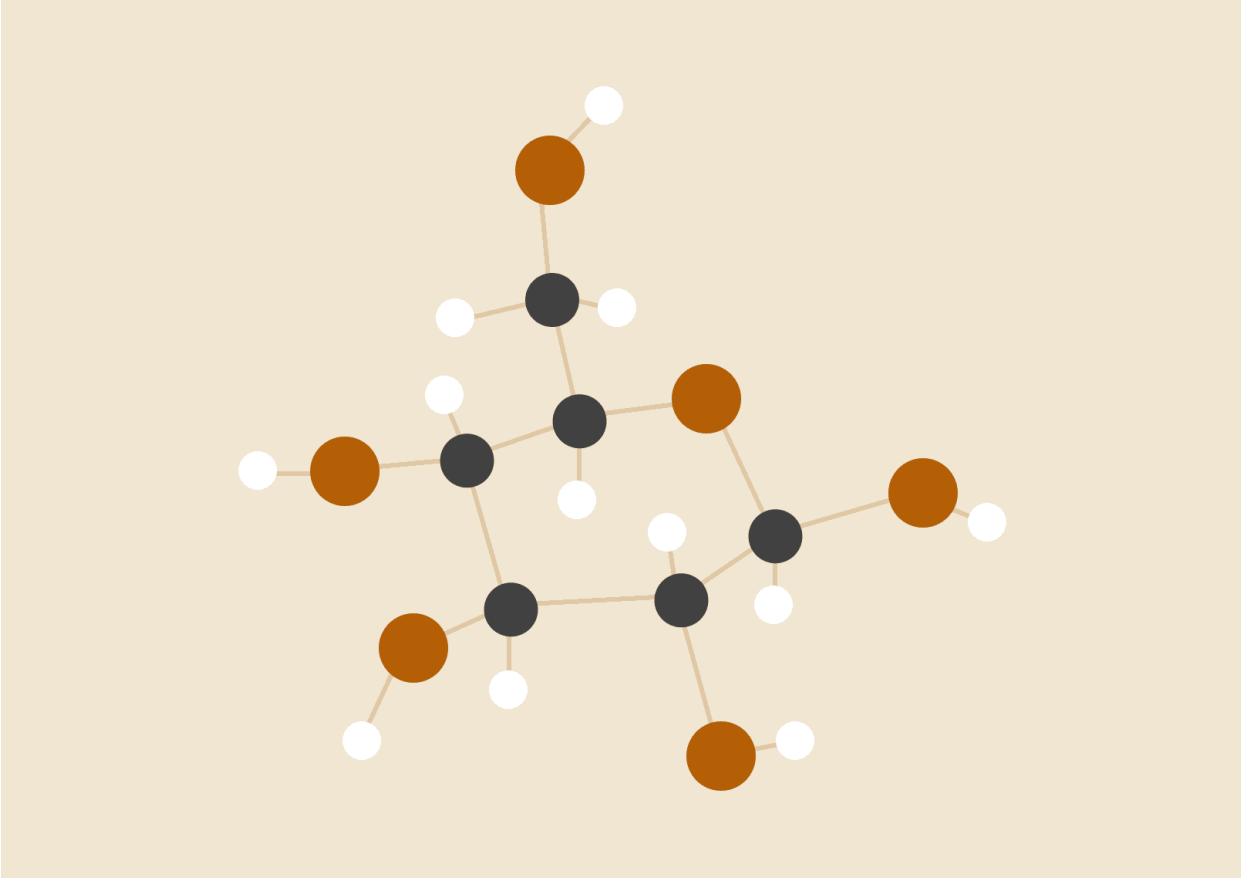
Modelagem com Support Vector Machine para previsão de Acidente Vascular Cerebral

Um exemplo de aprendizado de máquinas com dados de saúde



Guilherme Ceacero Rodrigues Maia

30/05/2024

INTRODUÇÃO

De acordo com a Organização Mundial da Saúde (OMS) o AVC é a 2ª maior causa de morte no mundo, responsável por aproximadamente 11% do total de mortes.

O conjunto de dados selecionado é usado para prever se um paciente provavelmente sofrerá AVC nos próximos 10 anos com base nos parâmetros de entrada, como sexo, idade, doenças preexistentes e tabagismo.

Diante da previsão do modelo, a ideia é aconselhar mudanças no estilo de vida do paciente para diminuir o seu risco.

MATERIAIS

- Pacotes R Utilizados:
    - readr: Para carregamento rápido da base de dados.
    - caret: Pelo seu ferramental poderoso para aprendizado de máquinas, incluindo suas funções de pré-processamento*, bootstrap e* ajuste de modelos.
    - dplyr: para manipulação do conjunto de dados eficiente.
    - mlr: pela sua ferramenta de alta eficiência de treinar modelos de imputação de dados faltantes.
    - rpart: pacote escolhido para usar *Random Forest* na imputação de dados faltantes de variáveis numéricas.
    - e1071: pacote escolhido para usar *Naive Bayes* na imputação de dados faltantes de variáveis numéricas.

PROCEDIMENTO

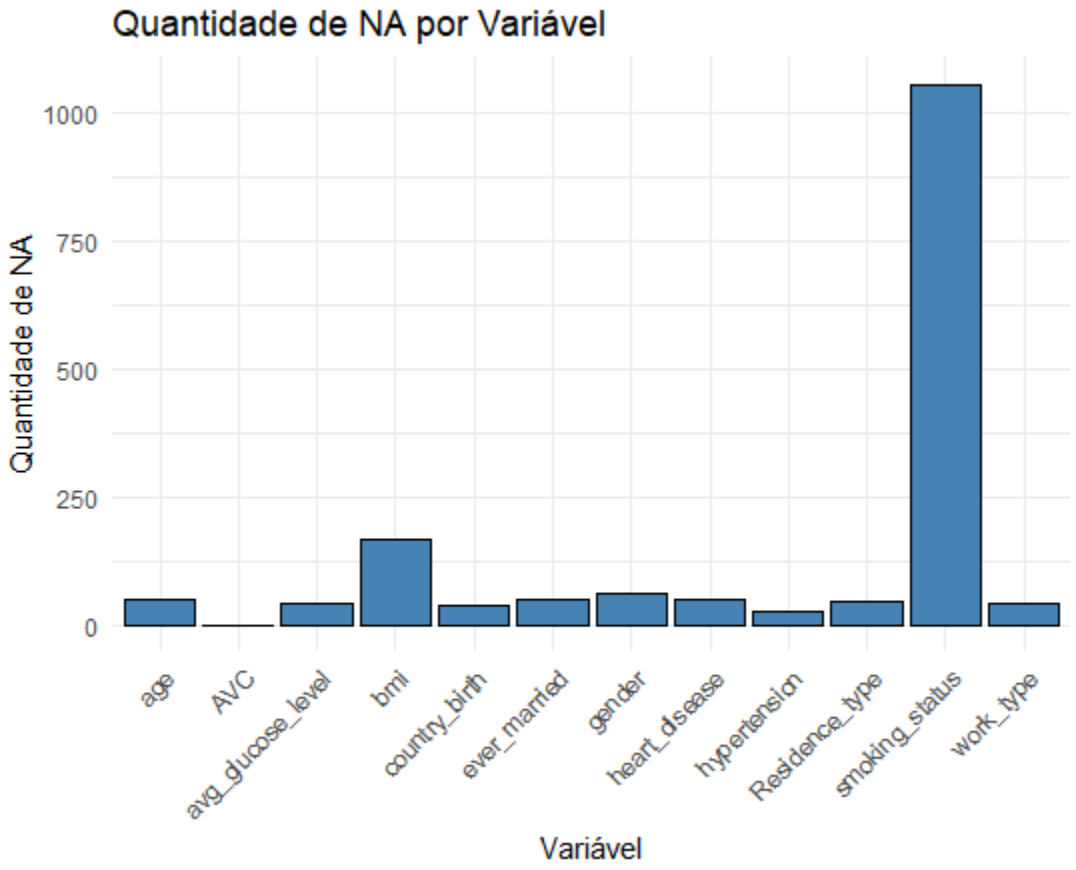
Vamos começar falando sobre a separação da base de dados em base de treino e base de teste. Assim como em todo trabalho de Aprendizado de Máquinas, particionaremos a base de dados que nos é fornecida em duas: uma de treino, a qual vamos usar para fazer todo nosso estudo e ajuste de modelo, e uma de teste, a qual vamos usar para avaliar a qualidade do nosso modelo final, treinado usando a base de treino.

A função dessa separação é saber se o nosso modelo final tem realmente valor preditivo, diante de novas informações que chegarão a ele no futuro.

Inicialmente, eu estava usando uma proporção de 90% da base de dados na base de treino. Os modelos alcançaram números melhores nessa proporção, entretanto, eram mais variados - cada vez que eu rodava o modelo, conseguia valores muito diferentes da acurácia e especificidade. Essa incerteza na qualidade do meu ajuste me fez optar pela proporção conservadora de 80%, que fornecia resultados mais consistentes nas medidas de qualidade de ajuste quando aplicava o modelo na base de teste, o que me deu mais segurança para entregar o trabalho final.

Agora, sobre o modelo de pré-processamento; começamos pelo tratamento básico e essencial que precisamos fazer nas variáveis de qualquer conjunto inicial de dados. Mudar o tipo de variável de todas as colunas para tipos que possamos usar no treinamento do modelo (numeric e factor). Depois, fazer uma análise de *outliers*. Nessa análise, eu descobri que a variável IMC foi guardada com um erro muito comum - a falta de vírgula. Haviam pessoas com IMC até 600. Consertei isso dividindo os valores dessa variável por 10, e ficando com os dados corretos. Nesse mesmo contexto, transformar as variáveis categóricas em *dummies* também é recomendado.

Depois, realizando uma análise de dados faltantes, percebe-se que existem muitos NA's na base de dados, como pode ser visto no gráfico a seguir.

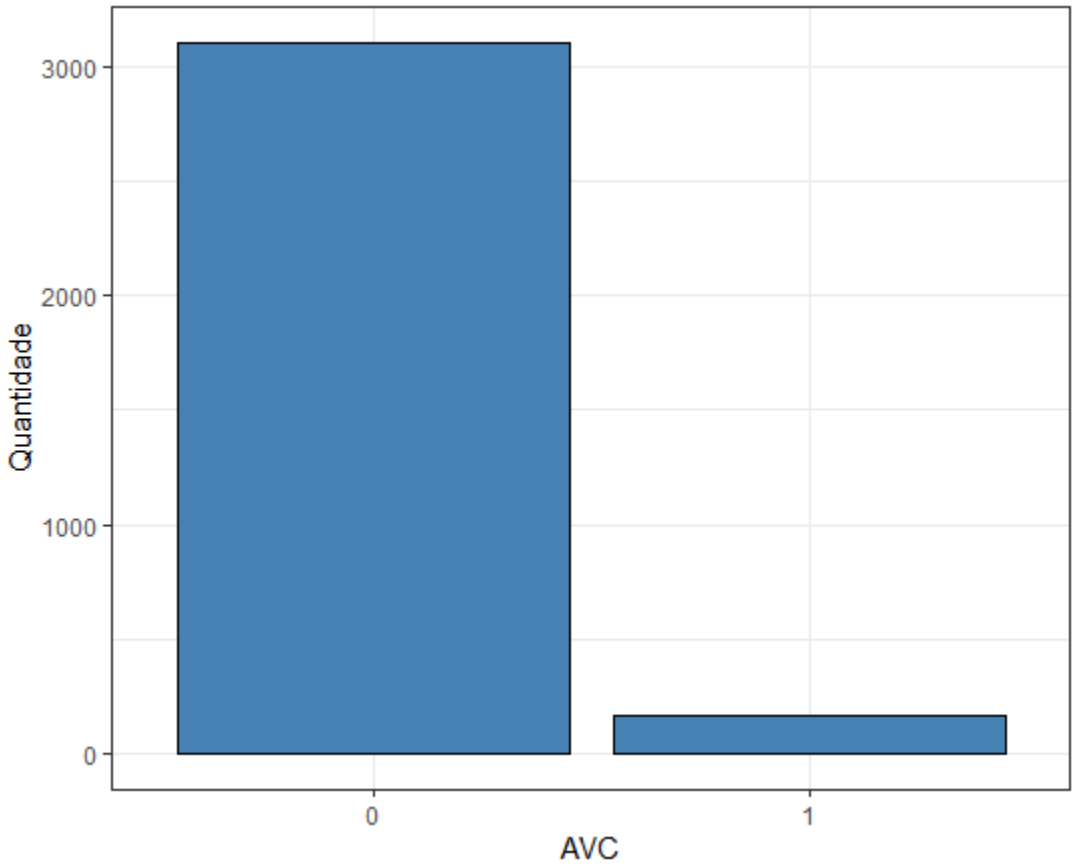


E então, de forma muito sofisticada, optei por usar um modelo de impute para os dados faltantes, onde escolhi:

- Random Forest para variáveis numéricas.
- Naive Bayes para variáveis categóricas.

Depois, eu renomeei algumas colunas que tinham símbolos "-" para "_", a fim de evitar erros; e por fim, selecionei todas as variáveis exceto a "id" para trabalhar em cima.

Sobre o rebalanceamento; quando em um contexto de variável resposta binária, para um bom ajuste final é necessário sempre prestar atenção na proporção dessa variável resposta no seu conjunto de dados. Se a maioria do conjunto de dados que possuímos é constituído de “0”s e poucos “1”s, isso pode fazer nosso modelo dizer “bem, ninguém nunca terá AVC, e com isso eu vou ter uma taxa de erro de apenas 1%”. Isso seria péssimo, porque queremos que nosso modelo identifique com boa eficiência pessoas em risco de terem AVC.



Bem cedo nesse trabalho eu vi que a variável resposta é desbalanceada. Temos muito mais pacientes que não sofreram AVC do que pacientes que de fato sofreram AVC.

Para métodos de rebalanceamento, eu testei usar o UpSample, DownSample e ROSE. Para o ROSE em específico, testei com diferentes possíveis proporções. No final, obtive os melhores resultados para esse estudo utilizando o UpSample, o qual eu adotei até o final para melhorar a especificidade do modelo.

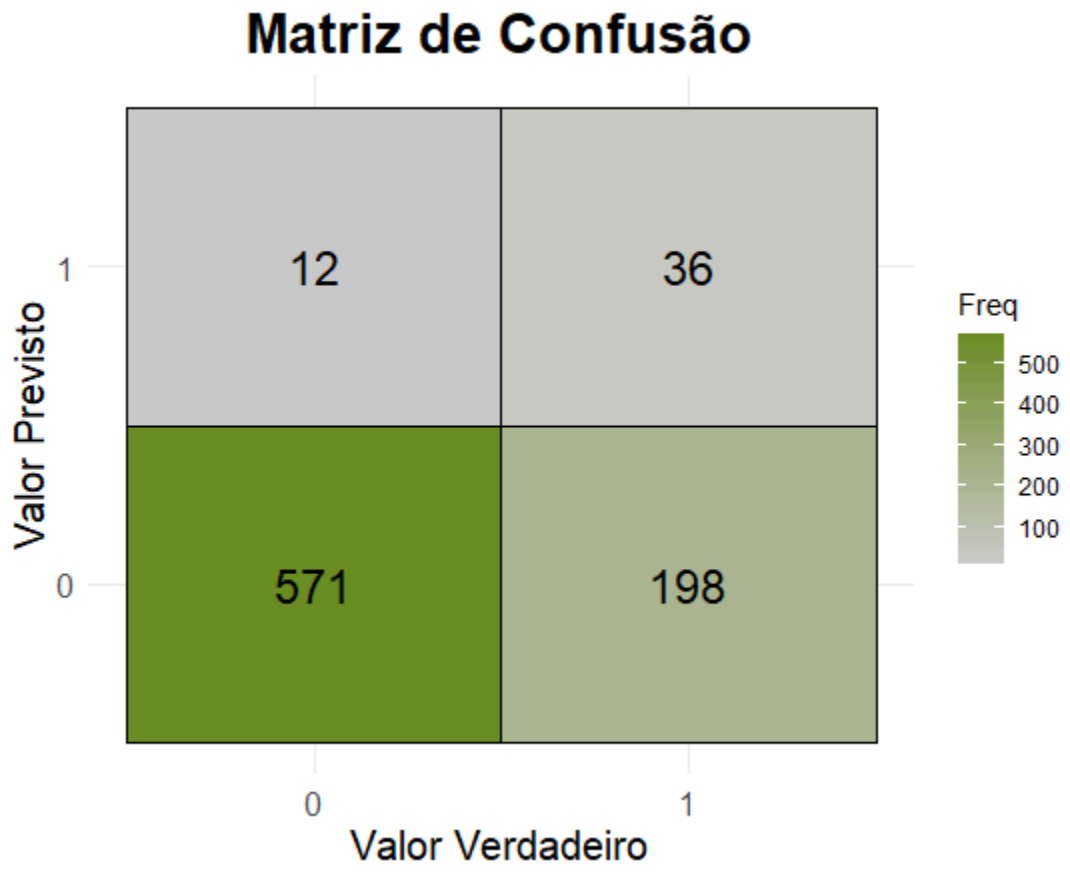
Sobre a seleção de hiperparâmetros para o modelo final, foi realizado um procedimento de *bootstrap* com *cross-validation* repetido, com 3 folds e 10 repetições. Isso envolve, de maneira simplificada, separar a nossa base de treino em 3, ajustar o modelo de Support Vector Machine com kernel linear em cada um desses conjuntos, e usar os outros conjuntos como base de teste para avaliar a qualidade de cada um dos 3 modelos, e repetir este procedimento 10 vezes. Dessa forma, podemos testar como diferentes valores para os hiperparâmetros do modelo impactam a qualidade do ajuste.

E então, no ajuste final de modelo, eu adotei o modelo Support Vector Machine com kernel Linear, que foi o de melhor desempenho dado todo o tratamento realizado. Na função de pré-processamento dentro da função train() do pacote *caret,* eu usei "*BoxCox*", já que todas as variáveis da base de dados são positivas (fato presente muitas vezes quando tratamos de dados de saúde). Veja um resumo da base de dados, que havia um total de 4088 pacientes:

Variável	Tipo	Valores assumidos na base de dados
Gênero	Categórica	Feminino ou Masculino
Idade	Numérica	0 a 82
Hipertensão	Categórica	Sim ou Não
Doença do Coração	Categórica	Sim ou Não
Casado	Categórica	Sim ou Não
Tipo de trabalho	Categórica	Privado, Governamental, com Crianças, Autônomo ou Nunca Trabalhou
Tipo de residência	Categórica	Rural ou Urbano
País de nascimento	Categórica	EUA ou Outro
Nível de glucose médio	Numérica	50 a 27.000
IMC	Numérica	9.2 a 66.8
Status de fumo	Categórica	Fuma, Já fumou ou Nunca fumou
AVC	Categórica	Sim ou Não

RESULTADOS

Após o ajuste do modelo final com a base de treino, podemos então testá-lo na base de teste, com a partição inicial que fizemos do conjunto de dados original. Com isso, podemos construir a matriz de confusão.



As métricas do nosso modelo foram:

Acurácia: 75%. Ou seja, nosso modelo final acertou 75% das previsões de que as pessoas teriam AVC ou não, dadas as informações disponíveis.

Sensibilidade: 74%. Ou seja, quando um paciente que vai ter AVC é avaliado pelo modelo, o modelo avisa o paciente dessa possibilidade 74% das vezes.

Especificidade: 76%. Ou seja, quando um paciente que não vai ter AVC é avaliado pelo modelo, o modelo acerta que ele de fato não terá AVC 76% das vezes.

Na matriz de confusão, podemos ver como essas métricas se comportam: quando o paciente não havia tido AVC, o modelo acertou 571 vezes e errou 198 vezes.

Já quando o paciente havia tido AVC, o modelo acertou 36 vezes e errou 12.

CONCLUSÃO

Ao final do estudo, temos um modelo equilibrado. Ele acusa risco de AVC em pessoas doentes com consistência boa, e também não erra o diagnóstico negativo de risco de AVC em pacientes saudáveis a níveis subótimos.

Caso fosse do interesse, o modelo poderia ser ajustado para aumentar ou diminuir a sensibilidade em detrimento da especificidade, ou vice versa, alterando o procedimento de rebalanceamento ou de *bootstrap.* Talvez seja mais importante ter um modelo com 99% de sensibilidade (que quase nunca erra ao avisar um paciente que está em risco) em detrimento da especificidade; mas essa decisão fica a interesse do médico ou pesquisador.

O modelo poderia ser melhorado também com um maior conjunto de dados - o utilizado havia informações de apenas 4088 indivíduos. Quanto maior o conjunto de dados, mais podemos treinar o modelo e mais eficiente ele se torna.