Trabalho 3 - INF-0615

Kaleb Roncatti de Souza e Nelson Gomes Brasil Júnior

25 de setembro de 2022

1 Introdução

Nos últimos três anos, vivenciamos uma situação que se tornou calamidade pública ao redor do mundo devido à pandemia causada pela COVID-19. Devido à sua letalidade e capacidade de contágio rápido, O vírus afetou e modificou completamente a maneira da qual vivemos. Logo com o início da epidemia, gerou-se um enorme desafio ao redor do planeta para a descoberta de testes que pudessem detectar a presença ou ausência do vírus de maneira rápida e eficiente.

Trazendo para o contexto da disciplina, modelos de aprendizado de máquina se mostram como excelentes candidatos para a previsão da existência do vírus no nosso organismo, utilizando-se informações do paciente tais como pressão arterial, quantidade de leucócitos, plaquetas, dentre outras informações que podem ser extraídas de exames sanguíneos. No trabalho vigente, utilizamos uma base de dados para a previsão de tal fenômeno utilizando árvores de decisão e árvores aleatórias.

2 Tarefas

Nesta seção, segmentaremos as respostas apresentando de maneira incremental os resultados obtidos via código.

2.1 Separação dos conjuntos e inspeção dos dados

Para a separação do conjunto utilizamos 80% do conjunto para treinamento e 20% para validação, removendo as duplicatas antes da separação. A separação resultou em 2700 amostras para o conjunto de treinamento e 676 amostras para o conjunto de validação.

Logo após a separação, estudamos a questão do balanceamento do `dataset` no conjunto de treinamento, observando a quantidade de amostras cujo `Resultado` foi `POSITIVO` (544 amostras) e a quantidade de amostras cujo `Resultado` foi `NEGATIVO` (2156 amostras). Percebemos então um desbalanceamento de classes no que diz respeito aos casos positivos do vírus, dado que temos aproximadamente 4 vezes de casos negativos do que temos de positivo. Para o tratamento, aplicamos a técnica de `oversampling`, fazendo com que o conjunto com ambos os `Resultados` `POSITIVO` e `NEGATIVO` ficasse com o mesmo número de amostras (2156 amostras).

Observação: Iremos apresentar em todas as seções abaixo a matriz de confusão relativa, que estará no seguinte formato

$$\begin{pmatrix} \texttt{TN} & \texttt{FP} \\ \texttt{FN} & \texttt{TP} \end{pmatrix} \tag{1}$$

2.2 Modelo baseline

No que diz respeito ao modelo considerado como Baseline, utilizamos todas as `features` fornecidas pelo conjunto e treinamos um modelo com `minsplit = 4` e usamos uma validação cruzada com 10 partições (`xval = 10`). Computamos a matriz de confusão relativa para os três conjuntos fornecidos (treino, validação e teste), **as quais demonstramos via código**, e obtivemos como acurácia balanceada:

- Treino: `Acurácia Balanceada = 0.9930`

$$\begin{pmatrix} 0.99 & 0.01 \\ 0.00 & 1.00 \end{pmatrix}$$

- Validação: `Acurácia Balanceada = 0.6353`

$$\begin{pmatrix} 0.84 & 0.16 \\ 0.57 & 0.43 \end{pmatrix}$$

- Teste: `Acurácia Balanceada = 0.6372`

$$\begin{pmatrix} 0.84 & 0.16 \\ 0.57 & 0.43 \end{pmatrix}$$

A ser utilizado para os próximos itens, decidimos encontrar a importância de cada um dos atributos, para posteriormente escolhermos os mais influentes para treinamento de modelos mais específicos.

2.3 Modelo com variação de profundidade

Para a variação em profundidade, utilizamos os mesmo parâmetros selecionados para `minsplit` e `xval`, variando-se o parâmetro `maxdepth` entre 1 e 29, observamos a variação da acurácia balanceada, e para o valor de `depth = 3` obtivemos o melhor resultado de acurácia balanceada para o conjunto de validação como mostrado na Figura 1.

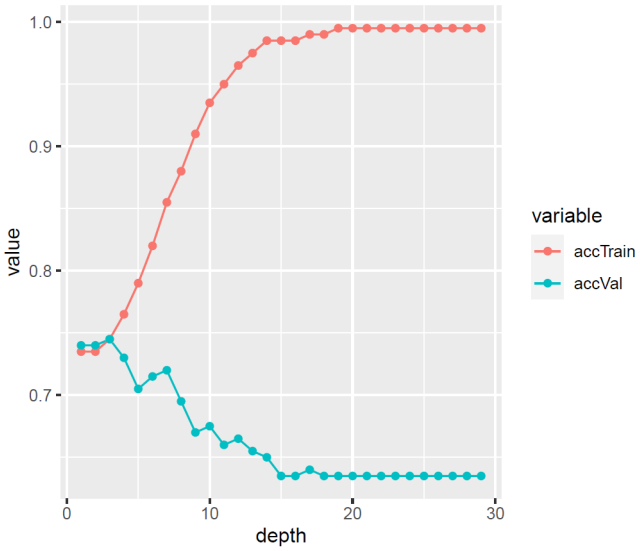

Figura 1: Acurácia balanceada para conjuntos de treino e validação variando-se a profundidade das árvores de decisão.

Para tal modelo, as matrizes de confusão para os cunjuntos de validação e testes **estão explicitadas no código**, e obtivemos:

- Validação: `Acurácia Balanceada = 0.7437`

$$\begin{pmatrix} 0.81 & 0.19 \\ 0.32 & 0.68 \end{pmatrix}$$

- Teste: `Acurácia Balanceada = 0.7146`

$$\begin{pmatrix} 0.83 & 0.17 \\ 0.40 & 0.60 \end{pmatrix}$$

Observe que, no conjunto de validação obtivemos uma acurácia balanceada significantemente relevante acima do modelo `baseline` para alguns valores de profundidade. Conseguimos observar:

- O fenômeno de `underfitting` entre os valores 1 e 2 de profundidade, dado que a acurácia tanto do conjunto de treino quanto do conjunto de validação seguem aumentando (o erro diminui) de forma relevante.
- Para o valor 3 em profundidade, observamos o valor ótimo do modelo a nível profundidade.
- O fenômeno de `overfitting` começa a ser observado para profundidades a partir de 4, mas novamente, entre 5 e 7, observamos uma pequena demonstração do que poderia ser considerado `underfitting`.
- Finalmente, a partir de profundidade igual a 8 observamos o fenômeno de `overfitting` até a estabilização do valor de acurácia balanceada para o conjunto de validação num valor extremamente reduzido, enquanto a acurácia balanceada do conjunto de treinamento se mostra muito próxima de 1, mostrando que o modelo se "viciou" nos dados de treino.

2.4 Modelo com subconjuntos de `features`

Como explicitado anteriormente no final da Seção 2.2, utilizamos as `features` com maior importância segundo o modelo `baseline` de árvores de decisão, e escolhemos dois modelos:

- O primeiro deles com 14 dos atributos mais importantes, utilizando-se `minsplit = 4`, `xval = 10` e `maxdepth = 3`.
- O segundo deles com apenas 8 dos atributos mais influentes, utilizando-se `minsplit = 4`, `xval = 10` e `maxdepth = 4`.

Obtivemos os seguintes valores de acurácia balanceada:

- Validação (14 `features`): `Acurácia Balanceada = 0.7409`
- Validação (8 `features`): `Acurácia Balanceada = 0.7631`

$$\begin{pmatrix} 0.84 & 0.16 \\ 0.32 & 0.68 \end{pmatrix}$$

Mostrando que o modelo com 8 atributos performou melhor do que todos os modelos até então, e para o conjunto de testes obtivemos:

- Teste (8 `features`): `Acurácia Balanceada = 0.694`

$$\begin{pmatrix} 0.83 & 0.17 \\ 0.45 & 0.55 \end{pmatrix}$$

2.5 Modelo com florestas aleatórias

No que diz respeito ao modelo de `random forests`, variamos o número de árvores entre 1 e 1000, utilizando-se os mesmos dois modelos de fórmulas (14 e 8 atributos) do item anterior. Os resultados são mostrados respectivamente na Figura 2 a seguir.

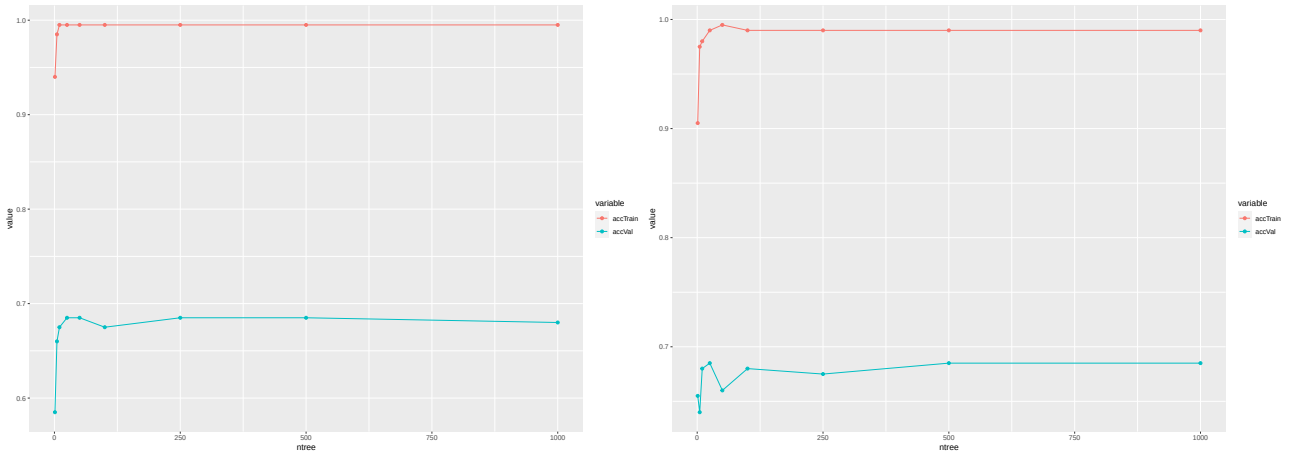

Figura 2: Acurácia balanceada para conjuntos de treino e validação variando-se a quantidade de florestas aleatórias respectivamente para 14 e 8 atributos.

Através das Figuras e dos valores de acurácia balanceada variando-se as florestas aleatórias, conseguimos perceber que o modelo mais performático se mostrou como o modelo com as 8 `features` mais relevantes e `ntree = 25`.

Para este caso, obtivemos:

- Validação: `Acurácia Balanceada = 0.6862`

$$\begin{pmatrix} 0.91 & 0.09 \\ 0.53 & 0.47 \end{pmatrix}$$

- Teste: `Acurácia Balanceada = 0.6536`

$$\begin{pmatrix} 0.89 & 0.11 \\ 0.59 & 0.41 \end{pmatrix}$$

Observamos que para 8 features, a partir de 25 árvores já encontramos uma região de overfitting. Além disto, mesmo na região ótima a escolha de features e parâmetros para a floresta aleatória obteve um resultado inferior aos modelos apresentados anteriormente, seja no conjunto de validação ou no conjunto de teste. Uma análise mais detalhadas nos hiperparâmetros ou da escolha de features poderia levar, talvez, a um resultado mais interessante.

2.6 Conclusão

Através dos modelos estudados, podemos perceber que o modelo da Seção 2.3, o modelo com profundidade `depth = 3` se mostrou como mais performático no conjunto de testes, obtendo `Acurácia Balanceada = 0.7146`. Olhando para a matriz de confusão percebemos que, se fôssemos usar este modelo em produção, ele classificaria de maneira mais efetiva casos negativos do que positivos (TFP = 0.6033 e `TFN = 0.826`). Poderíamos tentar melhorar estes resultado fazendo uso, por exemplo, de *ensembles* ou de uma melhor análise de features ou então de uma escolha do hiperparâmetro `minsplit`.