Trabalho 3

INF-0615 – Aprendizado de Máquina Supervisionado I

Grupo:
Evandro Santos Rocha
Laíssa Pacheco de Oliveira
Rafael Dantas de Moura

O objetivo deste trabalho é treinar árvores de decisão e florestas aleatórias utilizando uma base de dados com informações sobre pacientes com COVID-19 e inferir o seu estado: em tratamento, falecido ou recuperado.

1. Inspecionem os dados de treinamento. Quantos exemplos há de cada classe? O dataset está desbalanceado? Se sim, como vocês lidarão com o desbalanceamento ?

Com o comando `dim` aplicado ao conjunto fornecido, observamos que existiam **36.421** exemplos no total e que não havia qualquer dado com **NA**. Após a remoção dos repetidos, o conjunto final ficou com **30.351** exemplos.

Como temos somente um conjunto de dados, fizemos a divisão do mesmo em dois conjuntos: 80% para treinamento e 20% para validação. Através do comando `sample`, foram selecionados aleatoriamente 80% dos exemplos (dos 30.351) como sendo o conjunto de treinamento e o complemento (20%) como o conjunto de validação. Como os 30.351 não tinha repetição, esse processo de separação garante que não há repetição entre os conjuntos de treinamento e validação.

Conjunto	Número de exemplos (linhas)	Número de colunas
Treino	24.280	15
Validação	6.071	15

De posse do conjunto de treinamento, podemos avaliar as classes (comando `table`):

dead	onTreatment	recovered
1.412 (5,8%)	17.206 (70,9%)	5.662 (23,3%)

Notadamente, há um desbalanceamento entre as classes. Dentre as técnicas para fazer balanceamento, decidimos por **Undersampling**, onde, depois de alguns testes, resolvemos que os exemplos de classes com **onTreatment** seria 2,1 vezes maior que a **dead**, e os da classe **recovered** seria 1,4 vezes maior que a **dead**.

No final, o conjunto de treinamento ficou assim:

dead	onTreatment	recovered
1.412 (22,2%)	2.965 (46,7%)	1.976 (31,1%)

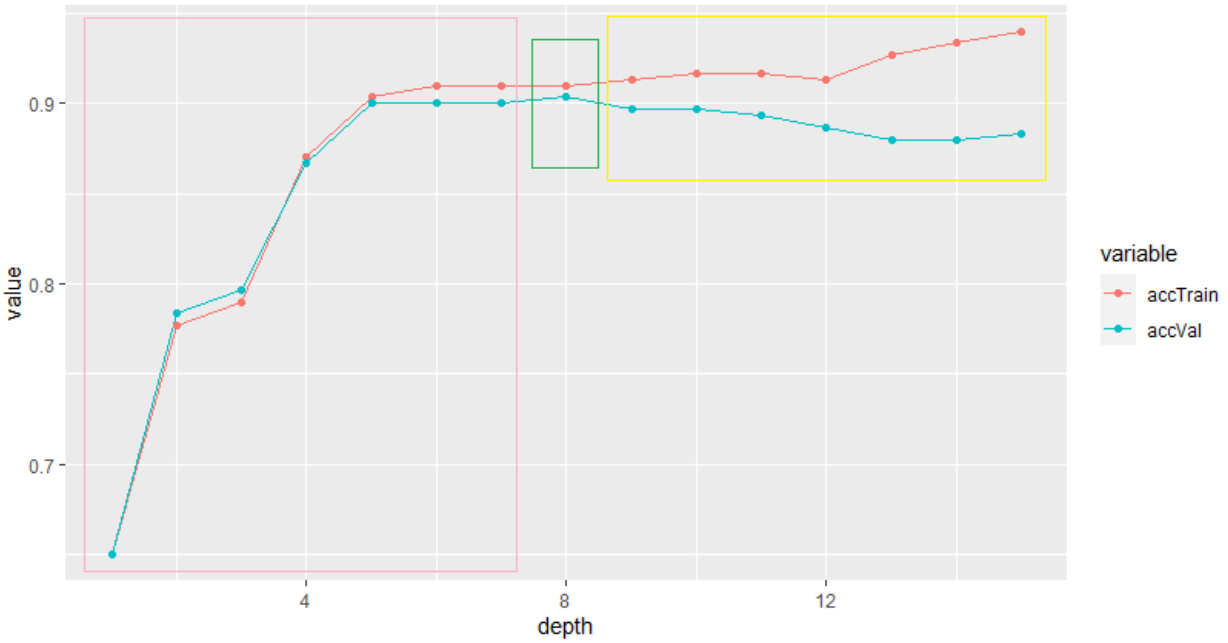
2. Treinem uma árvore de decisão como baseline e reportem a matriz de confusão relativa e a acurácia balanceada nos conjuntos de treinamento, validação e teste.

Treinamos uma Árvore de Decisão com todos os atributos com a qual obtivemos as seguintes matrizes de confusão relativa e acurácias balanceadas:

		Treinamento			Validação			Teste		
		dead	onTreatment	recovered	dead	onTreatment	recovered	dead	onTreatment	recovered
Reference	dead	1.00	0.00	0.00	0.98	0.01	0.00	0.99	0.01	0.00
	onTreatment	0.00	0.98	0.02	0.00	0.84	0.15	0.00	0.83	0.16
	recovered	0.00	0.02	0.98	0.00	0.22	0.78	0.01	0.23	0.77
Acc Bal		0.9866667			0.8666667			0.8633333		

3. Treinem outras árvores de decisão variando o tamanho das árvores geradas. Plotem a acurácia balanceada no conjunto de treinamento e validação pela profundidade da árvore de decisão. Identifiquem as regiões de underfitting, ponto ótimo e overfitting. Tomem a árvore com tamanho ótimo e reportem também a matriz de confusão relativa e a acurácia balanceada no conjunto de teste.

Variando o tamanhos das árvores, obtivemos os seguintes valores de acurácia balanceada:



Considerando o gráfico acima, o **ponto ótimo** (destaque em verde) foi obtido para a árvore de profundidade **8**. Podemos também dizer que o modelo apresentou **underfitting** (destaque em rosa) para as profundidades **entre 1 e 7**, e **overfitting a partir de 9** (destaque em dourado).

Considerando a árvore ótima, com profundidade máxima igual a 8, com o conjunto de testes obtivemos:

Matriz de confusão	**Prediction**				Acurácia balanceada
	dead	onTreatment	recovered		0.8966667
dead	0.99	0.01	0.00		
onTreatment	0.00	0.72	0.28		
recovered	0.01	0.01	0.98		

4. Explorem pelo menos 2 possíveis subconjuntos de features (feature selection) para treinar uma árvore de decisão. Tomem o melhor modelo e reportem a matriz de confusão relativa e a acurácia balanceada do no conjunto de teste.

Inicialmente, montamos uma árvore com todos os atributos já considerando o ponto ótimo da questão anterior (profundidade máxima igual a 8).

Com esta árvore, consultamos as suas variáveis mais importantes e, a partir delas geramos um primeiro subconjunto com as variáveis (atributos) que tinham mais do 10% de importância relativa: date_death_or_discharge (21% de importância relativa), date_admission_hospital (17,2%), country (13,7%), travel_history_dates (13,2%), longitude (13,2%) e lives_in_Wuhan (11,9%).

Em seguida, montamos um segundo subconjunto com as variáveis que, juntas, somavam mais que 66% de importância: date_death_or_discharge (21% de importância relativa), date_admission_hospital (17,2%), country (13,7%), travel_history_dates (13,2%).

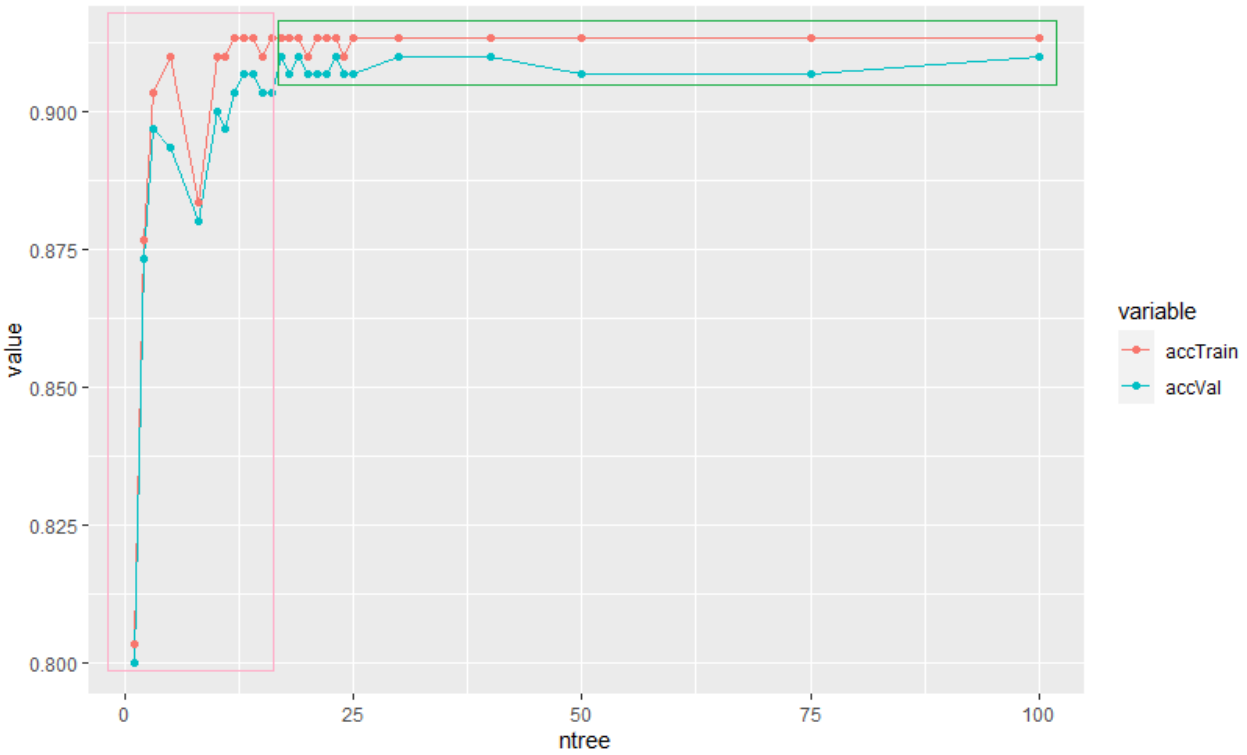
Finalmente, montamos um subconjunto que dizem respeito ao indivíduo, sintomas e viagem: age, sex, chronic_disease_binary, date_onset_symptoms, date_admission_hospital, date_confirmation, lives_in_Wuhan, travel_history_dates, travel_history_location e travel_history_binary.

Avaliando esses modelos com o conjunto de validação, obtivemos, respectivamente as seguintes acurácias balanceadas: **0.8033333, 0.8 e 0.9.** Assim, o melhor subconjunto de atributos foi o terceiro, onde utilizamos atributos que dizem respeito ao indivíduo, sintomas e viagem. Aplicando esse modelo para o conjunto de testes, obtivemos:

Matriz de confusão	**Prediction**				Acurácia balanceada
	dead	onTreatment	recovered		0.9
dead	1.00	0.00	0.00		
onTreatment	0.00	0.71	0.29		
recovered	0.00	0.01	0.99		

5. Treinem várias florestas aleatórias variando o número de árvores. Plotem a acurácia balanceada no conjunto de treinamento e validação variando o número de árvores geradas. Identifiquem as regiões de underfitting, ponto ótimo e overfitting. Reportem também a matriz de confusão relativa e a acurácia balanceada no teste para a floresta com o melhor número de árvores.

Variando o número de árvores, obtivemos os seguintes valores de acurácia balanceada:



Considerando o gráfico acima, a **região ótima** (destaque em verde) foi obtida a partir de **17 árvores até 100**. Fizemos outros experimentos com mais árvores e o modelo ficou variando entre 0.903 e 0.91. Podemos também dizer que o modelo apresentou **underfitting** (destaque em rosa) para quantidade de árvores entre **1 e 16**. Como o modelo a partir de 17 árvores fica variando entre 0.91 de acurácia e um pouco menos que esse valor, não foi identificada uma região de **overfitting**. Como a região ótima possui vários pontos, podemos considerar como **ponto ótimo** a de menor complexidade, ou seja, número de árvores **igual a 17**. Aplicando o melhor modelo para o conjunto de testes, obtivemos:

Matriz de confusão	**Prediction**				Acurácia balanceada
	dead	onTreatment	recovered		0.9
dead	1.00	0.00	0.00		
onTreatment	0.00	0.72	0.28		
recovered	0.00	0.02	0.98		

6. Escreva um relatório de no máximo 5 páginas reportando:
(a) A diferença de desempenho entre o baseline e os outros modelos mais complexos gerados.

Alguns detalhes estão melhor explicados nas questões acima.

Inicialmente, no exercício 2, foi criada uma Árvore de Decisão, chamada de baseline, com os dados de treinamento. Com essa baseline, foi realizada uma predição onde acurácia balanceada foi de **0.8666667** no conjunto de validação.

Em seguida, no exercício 3, variamos o tamanho das árvores geradas, de 1 a 15, e verificamos como se comporta a acurácia nos conjuntos de treinamento e validação. Analisando o gráfico de viés e variância, foi identificado que o melhor tamanho limite de árvore é 8 com acurácia de **0.9033333** no conjunto de validação.

Já na questão 4, variamos o conjunto de atributos da árvore de decisão. Inicialmente, construímos uma árvore, parametrizada com tamanho limite de 8, e variamos o conjunto de atributos do modelo. Montamos 3 subconjuntos, onde o melhor deu uma acurácia de **0.9**. Isso mostra a importância da seleção dos atributos, pois nos devolve acurácias diferentes.

Na questão 5, partimos para modelos com florestas aleatórias, onde o algoritmo se encarrega de criar árvores variadas e combiná-las. Neste exercício, deixamos fixada a quantidade de atributos selecionados (aleatoriamente) como 4 (mtry=4), que é a raiz quadrada da quantidade de atributos, aproximada para cima, e variamos a quantidade de árvores criadas, entre 1 e 100. Para a quantidade de 17 árvores, o modelo atingiu uma acurácia de **0.91**. Acima desta quantidade de árvores, o modelo ficou variando entre este máximo de 0.91 de acurácia e um valor um pouco abaixo disso, indicando que, a partir de 17, é uma região ótima.

(b) Houve overfitting? Houve underfitting? Analisem as curvas viés/variância geradas ao longo do trabalho.

As respostas estão melhor detalhadas nas questões acima.

Na questão 3, onde variamos o tamanho da árvore gerada em uma árvore de decisão, encontramos as três regiões: underfitting, região ótima e overfitting. Pelo gráfico de viés e variância gerado, as regiões ficaram muito claras, pois, após atingir a acurácia máxima para o conjunto de validação no ponto ótimo (igual a 8), a acurácia no conjunto de validação começou a cair enquanto a acurácia no conjunto de treinamento aumentava, indicando overfitting.

Já na questão 5, onde variamos o número de árvores em uma floresta aleatória, não houve overfitting. A partir do ponto ótimo (igual a 17), o modelo ficou variando a acurácia no conjunto de validação, demonstrando que a partir desse ponto tudo é uma região ótima.

(c) uma Seção de conclusão do relatório explicando a diferença entre os modelos e o porquê que estas diferenças levaram a resultados piores ou melhores.

Como conclusão, ao utilizar árvores de decisão para solucionar problemas de classificação, conseguimos melhores resultados ao variar o tamanho da árvore e o subconjunto de atributos utilizados. Além disso, ao tentar selecionar o melhor subconjunto de atributos, caímos em uma situação de conhecimento do negócio/problema para montar a melhor árvore. A ideia de florestas aleatórias resolve esse problema ao montar árvores com combinação de atributos.

Também é essencial plotar os gráficos de “viés e variância” e de acurácia balanceada para, ao observar o comportamento das perdas e acurácia dos modelos, identificar as regiões ótima, de underfitting e overfitting.

Opcional 1 - Implementem manualmente o protocolo Random Forest de forma que cada árvore na floresta tenha as mesmas quantidades de exemplos das três classes. Note que, para cada modelo, vocês devem selecionar com repetição um subconjunto de exemplos de cada uma das classe para treiná-lo.

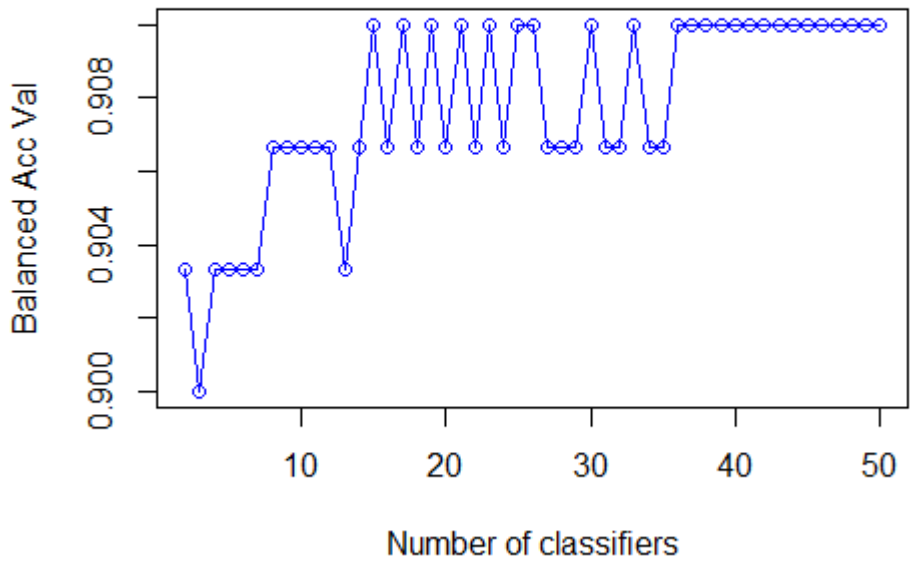
Foram montadas 50 árvores, onde cada uma foi treinada com um número igual de exemplos de cada classe.

Os exemplos foram escolhidos aleatoriamente e com repetição. ,

Após fazer a predição para todas as árvores, foi realizada a votação utilizando a moda.

Em seguida, foram calculadas a matriz de confusão, onde a acurácia balanceada foi de **0.91**.

Ao analisar a variação de classificadores, pudemos ver como varia a acurácia balanceada no conjunto de validação (gráfico abaixo). Este não foi um gráfico de viés e variância por não ter a comparação com a acurácia no conjunto de treinamento, mas podemos dizer que a acurácia balanceada atingiu o ponto máximo em **0,91** no ponto 14. Após esse ponto, ficou em uma variação até se estabilizar nesse máximo a partir do ponto 35.

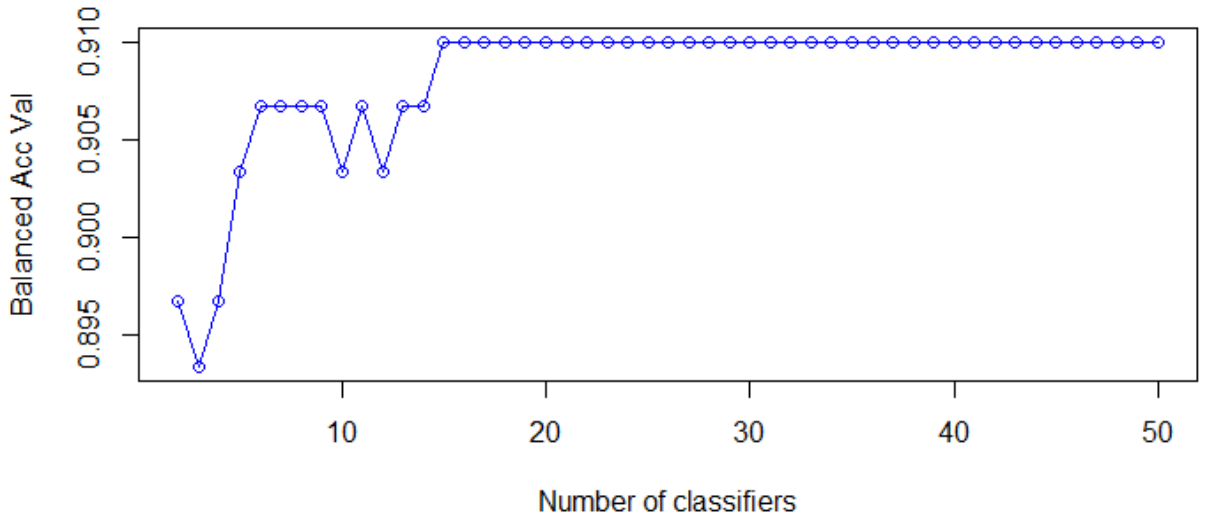


Opcional 2 - Variem o número de features consideradas no treinamento. Utilizando sqrt(m), m/2 e 3m/4 atributos, em que m é o número total de atributos que vocês têm disponível.

Antes fizemos uma função getHypothesis que devolve a fórmula baseda nas features fornecidas. Depois, fizemos uma maneira de sortear a quantidade de features e as próprias features.

Após essa parte de desenvolvimento, montamos novamente 50 árvores, onde novamente cada uma foi treinada com um número igual de exemplos de cada classe, mas agora sorteando a quantidade de features e as próprias features.

O processo foi como no exercício anterior, e as acurácias, por coincidências foram as mesmas, **0.91** e a acurácia máxima também foi atingida no ponto 14. A diferença foi que, após atingir o máximo, a acurácia permaneceu a mesma para os pontos seguintes, ou seja, não houve variação.



Opcional 3 - Reportem seus resultados e suas conclusões no relatório. Esses resultados foram melhores que os modelos treinados realizando o balanceamento a priori?

Os detalhes foram respondidos nas questões acima.

Em termos de resultado, a acurácia balanceada foi a mesma. A diferença, ao menos nos testes efetuados, parece estar no desenho final da variação da acurácia. Em especial o último gráfico, além de ter atingido a acurácia máxima em um ponto antes dos exercícios anteriores, obtemos uma estabilidade da mesma.