INF-0615 – Aprendizado de Máquina Supervisionado I

Trabalho 3 - Árvores de Decisão e Florestas Aleatórias

I IDENTIFICAÇÃO

- Daniele Montenegro da Silva Barros
- Rodrigo Dantas da Silva
- Thiago Bruschi Martins

QUESTÃO 1: BALANCEAMENTO

No banco de dados de treino e validação existem ao todo 36421 registros. A distribuição das classes é dada pela Tabela 1. Podemos observar que a classe OnTreatment é muito mais abundante que as outras, e a classe Dead possui bem menos observações do que as outras, o que caracteriza um desbalanceamento entre as classes.

Tabela 1. Distribuição das classes no conjunto de treino

Classe	**Observações**	**Proporção (%)**
onTreatment	25840	71
Recovered	8435	23
Dead	2146	6

Para balancear os dados, nós testamos três técnicas de balanceamento sobre um modelo de Árvore de Decisão: Oversampling (Recovered), Undersampling (onTreatment) e a técnica que utiliza pesos diferentes para cada classe nos dados de treino. Nós não testamos balancear a classe Dead pois os resultados na matriz de confusão relativa indicaram um acerto de pelo menos 99% para esta classe, além de não haver nenhum falso positivo para esta classe. Testamos então três combinações possíveis destas técnicas, além de treinarmos um modelo sem balanceamento, tomando a acurácia balanceada no conjunto de validação como métrica de comparação, e obtivemos o seguinte resultado:

Tabela 2.Acurácia Balanceada no conjunto de validação para diferentes estratégias de balanceamento

Modelo	**Acurácia Balanceada**
Undersampling + Weights	0.870
Oversampling + Undersampling + Weights	0.860
Oversampling + Weights	0.847
Sem Balanceamento	0.833

QUESTÃO 2: BASELINE

Através da Tabela 2 escolhemos o balanceamento com Undersampling + Weights para treinar nosso baseline. Sendo assim, geramos a matriz de confusão relativa, bem como a acurácia balanceada, do nosso baseline para os 3 conjuntos de dados: treino, validação e teste. Os resultados são apresentados nas Tabelas 3, 4 e 5 respectivamente.

Tabela 3: Matriz de Confusão Relativa do Baseline no conjunto de Treino

• Modelo: Baseline • Dataset: Treino • Acurácia Balanceada: 0.980		Predição		
		Dead	OnTreatment	Recovered
Referência	Dead	1.00	0.00	0.00
	OnTreatment	0.00	0.94	0.06
	Recovered	0.00	0.00	1.00

Tabela 4: Matriz de Confusão Relativa do Baseline no conjunto de Validação

• Modelo: Baseline • Dataset: Validação • Acurácia Balanceada: 0.870		Predição		
		Dead	OnTreatment	Recovered
Referência	Dead	0.99	0.01	0.00
	OnTreatment	0.00	0.84	0.16
	Recovered	0.00	0.22	0.78

Tabela 5: Matriz de Confusão Relativa do Baseline no conjunto de Teste

• Modelo: Baseline • Dataset: Teste • Acurácia Balanceada: 0.853		Predição		
		Dead	OnTreatment	Recovered
Referência	Dead	0.99	0.01	0.00
	OnTreatment	0.00	0.82	0.18
	Recovered	0.00	0.25	0.75

QUESTÃO 3: AUMENTANDO A PROFUNDIDADE

Após selecionarmos e testarmos nosso baseline, variamos o parâmetro de profundidade da árvore sobre ele e obtivemos o seguinte gráfico de diagnóstico de Viés e Variância.

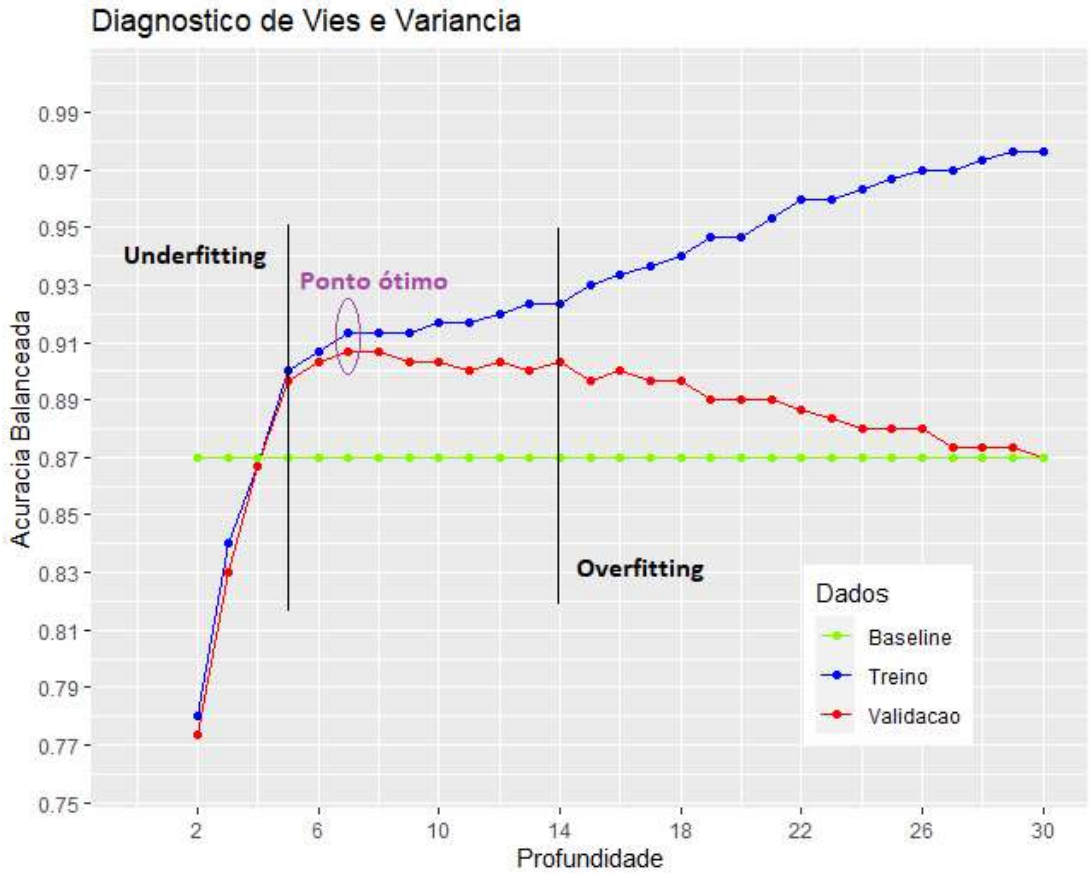

Figura 1. Diagnóstico de Viés e Variância variando o parâmetro maxdepth para o modelo com Undersampling + Weights.

As regiões de Underfitting, ponto ótimo e Overfitting estão indicadas na Figura 1. A região de Underfitting está bem evidente no gráfico, pois há uma grande diferença na acurácia balanceada dos pontos de profundidade inferior a 5. Já na região de Overfitting nós escolhemos o ponto em que começa uma descida mais acentuada na acurácia balanceada do conjunto de validação. Embora esta acurácia tenha começado a descer a partir do ponto de profundidade igual a 9, ela fica mais evidente depois da profundidade igual a 14. Enquanto isso, a acurácia balanceada no conjunto de treino continua crescendo até próximo dos 98%.

O ponto ótimo escolhido foi o de profundidade (max_depth) igual a 7, pois é o ponto que possui maior acurácia balanceada para o conjunto de validação, junto com o ponto de profundidade igual a 8. Optamos pela profundidade igual a 7 pois é um modelo com complexidade mais baixa e resultado semelhante ao modelo de complexidade mais alta. Após esta análise, testamos o modelo escolhido no conjunto de teste.

Tabela 6: Matriz de Confusão Relativa do ponto ótimo no conjunto de Teste

• Modelo: MaxDepth = 7 • Dataset: Teste • Acurácia Balanceada: 0.900		Predição		
		Dead	OnTreatment	Recovered
Referência	Dead	0.99	0.01	0.00
	OnTreatment	0.00	0.73	0.27
	Recovered	0.00	0.02	0.98

Desta forma, conseguimos uma melhora significativa em relação ao nosso baseline. A acurácia balanceada no conjunto de teste subiu de 0.853 para 0.900.

QUESTÃO 4: SELEÇÃO DE FEATURES

Para realizar a seleção de features, geramos uma tabela com o atributo $*variable.importance* que é criado após o treinamento de cada modelo *rpart*. Ao invés de utilizarmos os valores absolutos, utilizamos os valores relativos para que fosse mais fácil comparar os valores entre as variáveis. A tabela a seguir foi gerada a partir do melhor modelo da seção anterior (MaxDepth=7) sobre o conjunto de treino balanceado e utilizada para a seleção de features.

Tabela 7. Importância relativa de cada uma das features gerada pelo nosso melhor modelo de árvore de decisão

Posição	Feature	Importância Relativa (%)
1	date_death_or_discharge	0.1862759898
2	date_admission_hospital	0.1669837382
3	country	0.1490444974
4	travel_history_dates	0.1440062400
5	longitude	0.1430208221
6	lives_in_Wuhan	0.1348552750
7	age	0.0370385244
8	sex	0.0323346751
9	travel_history_location	0.0042645207
10	date_confirmation	0.0006821653
11	date_onset_symptoms	0.0005856634
12	travel_history_binary	0.0005457322
13	latitude	0.0003621563

Nesta tabela podemos observar em ordem decrescente as features consideradas as mais importantes para o nosso modelo. Os valores de importância são calculados levando em consideração quantas vezes cada feature foi selecionada para fazer parte de um split da árvore, tanto como *primary variable* quanto como *surrogate variable*. Além disso, é levado em consideração o ganho que aquela variável trouxe em cada split, isto é, o quanto aquele split diminuiu diminuiu o erro do nosso modelo.

Com base na Tabela 7 realizamos uma seleção de features variando de 2 até 13, isto é, treinamos nosso melhor modelo (max_depth = 7) variando a quantidade de features de 2 até 13, seguindo a ordem decrescente da Tabela 7. Com os resultados de treinamento e validação construímos o gráfico da Figura 2.

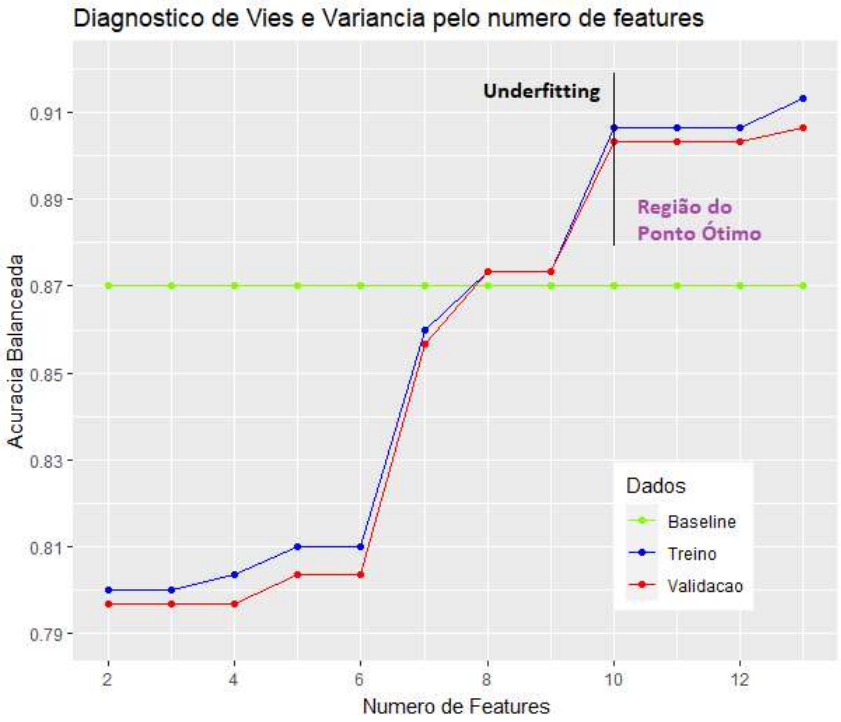

Figura 2. Diagnóstico de Viés e Variância pelo número de features

Analisando este gráfico juntamente com os dados da Tabela 6, podemos observar que algumas features que não estão nas primeiras posições de importância causaram um aumento da acurácia balanceada bem mais significativo do que algumas das features que estão nas primeiras posições. Comparando o eixo X do nosso gráfico com a primeira coluna da Tabela 6, verificamos que as duas features que tiveram um efeito mais positivo na acurácia balanceada, depois das duas primeiras, foram a 7ª e a 10ª feature: *age* e *date_confirmation*. Acreditamos que esta divergência entre a tabela e o gráfico deu-se porque provavelmente estas duas features combinadas com as features que foram adicionadas antes delas devem trazer um ganho muito maior do que elas sozinhas. E provavelmente a tabela gerada através da *$variable.importance* provavelmente analisou cada feature individualmente nos splits.

Verificamos também que o desempenho do nosso modelo foi melhor utilizando a maior parte das features, tendo uma pequena melhora quando a última feature (latitude) foi inserida. Sendo assim, não houve uma região de Overfitting no gráfico e o nosso melhor modelo continua sendo o mesmo (que utiliza todas as features). Portanto, nossa matriz de confusão relativa também não muda. De qualquer forma colocamos a matriz de confusão do nosso melhor modelo aqui novamente.

Tabela 8: Matriz de Confusão Relativa sobre o conjunto de Teste do melhor modelo após a seleção de features

• Modelo: MaxDepth = 7 • Dataset: Teste • Acurácia Balanceada: 0.900		Predição		
		Dead	OnTreatment	Recovered
Referência	Dead	0.99	0.01	0.00
	OnTreatment	0.00	0.73	0.27
	Recovered	0.00	0.02	0.98

QUESTÃO 5: VARIANDO O NÚMERO DE ÁRVORES

Como última análise, variamos o número de árvores de decisão em uma Random Forest. A dificuldade aqui foi encontrar um número de features adequado para que o diagnóstico de viés e variância ficasse bom. O parâmetro que controla o número de features do modelo *randomForest* é o *mtry*. Quando tentamos variar o número de árvores sem ajustar esse parâmetro, o gráfico ficou muito confuso. Então tentamos variar o número de features até encontrar um gráfico razoável. Após várias tentativas, o melhor resultado foi obtido com 30% das features iniciais. O gráfico gerado pode ser observado na figura a seguir.

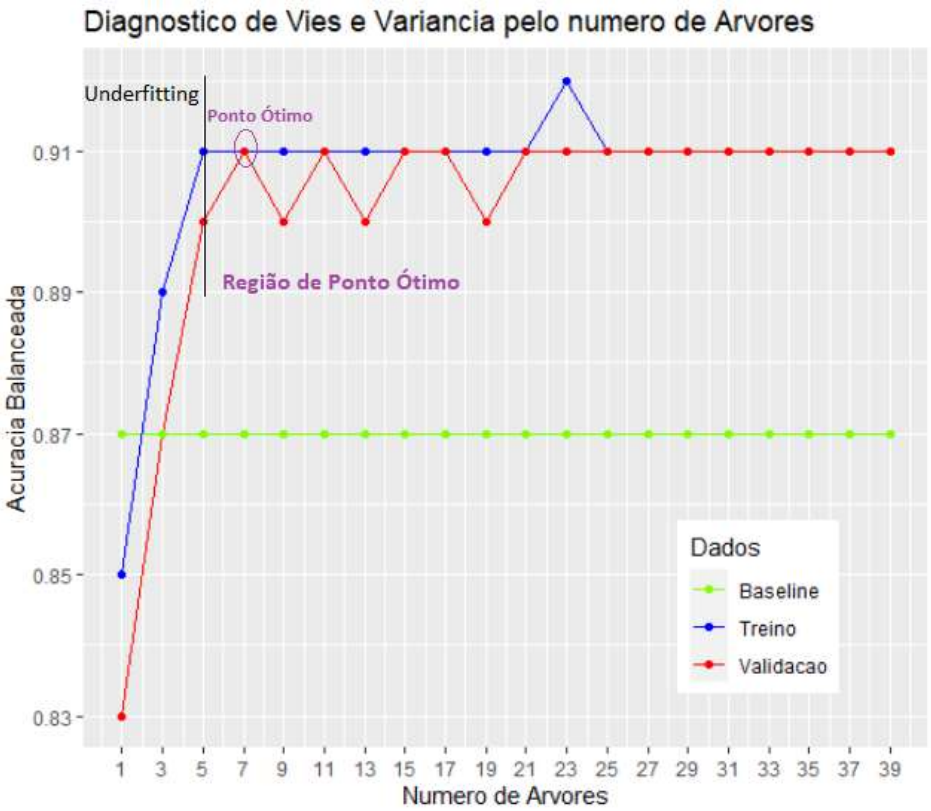

Figura 3. Diagnóstico de Viés e Variância do modelo Random Forest pelo número de Árvores

As regiões de Underfitting e Região do Ponto ótimo estão indicadas no gráfico. Não houve região de Overfitting nem para os modelos com valores mais altos de árvores. O ponto ótimo escolhido foi o ponto onde o número de árvores é igual a 7, pois ele possui o desempenho mais alto junto com outros modelos, porém é o que possui grau de complexidade mais baixo entre eles. Sendo assim, testamos o modelo escolhido no conjunto de teste. O resultado pode ser observado na tabela a seguir. Através da acurácia balanceada observamos que o desempenho foi muito semelhante ao que havíamos obtido com uma Árvore de Decisão de profundidade 7. Houve uma pequena diferença na matriz de confusão relativa, mas ela foi mínima.

Tabela 9: Matriz de Confusão Relativa sobre o conjunto de Teste do melhor modelo de Random Forest

• Modelo: Número de Árvores = 7 • Dataset: Teste • Acurácia Balanceada: 0.900		Predição		
		Dead	OnTreatment	Recovered
Referência	Dead	1.00	0.01	0.00
	OnTreatment	0.00	0.74	0.26
	Recovered	0.00	0.04	0.96

CONCLUSÃO

Inicialmente observamos que nosso modelo de baseline obteve um resultado no conjunto de teste bastante inferior à acurácia balanceada no conjunto de treino, o que pode ser um indício de overfitting. Esta suposição ficou evidente quando plotamos o gráfico da Figura 1 e lembramos que o baseline não possuía parâmetro de limite de profundidade para a árvore de decisão. Sendo assim, ao escolhermos um modelo menos complexo e na região do ponto ótimo, já era esperado um desempenho melhor no conjunto de teste. Resultado que se concretizou e subiu nossa acurácia balanceada de 0.853 para 0.900. Ao realizarmos uma seleção do conjunto de features, observamos que o melhor resultado continuou sendo do modelo que utilizava todas as features. Acreditamos que isso tenha acontecido pois ao reduzirmos o número de features, reduzimos a complexidade do modelo, e tendo em vista que nosso modelo era o modelo da região do ponto ótimo com complexidade mais baixa do gráfico da Figura 1, ao reduzirmos o número de features, e portanto a sua complexidade, o modelo acabou entrando numa região de Underfitting. Por último, variamos o número de árvores do modelo de Random Forest. O fato do desempenho da random forest ter sido muito parecido com o do modelo de uma árvore só acreditamos que tenha acontecido pela baixa complexidade dos dados, o que pode ser observado por exemplo, pela pequena quantidade de features. Além disso, a região de ponto ótimo da Random Forest começou já com 5 árvores. Sendo assim, um modelo mais complexo como a Random Forest poderia, neste caso, ser substituído por uma Árvore de Decisão de profundidade 7. Caso tivéssemos dados com um maior grau de complexidade, como por exemplo, um maior número de features, aí provavelmente a Random Forest utilizaria mais árvores e teria um resultado melhor. Mas lembrando sempre que não é possível garantir isso sem antes ser feita uma verdadeira análise sobre o problema.