INF0615 – Aprendizado de Máquina Supervisionado

Trabalho 3 - Regressão Linear

Nicole Nogueira Silva Rodolfo Dalla Costa

Introdução

Durante o ano de 2020 e 2021, a humanidade foi afetada pela pandemia do vírus COVID-19 que ceifou milhares de vidas e gerou impactos em diversas áreas sociais. Milhões de pessoas de diferentes países foram contaminadas apresentando diferentes quadros clínicos de reação ao vírus. Dessa forma, o objetivo desse trabalho é inferir o possível estado do paciente diagnosticado com o vírus COVID-19 dentre três possíveis classes: em tratamento, recuperado ou falecido.

Banco de dados

O banco de dados provem originalmente de um conjunto de dados reunidos por diversos países no mundo e contém informações como data de internação, se o paciente tem ou não doenças crônicas, entre outras variáveis. O banco de dados fornecido para treino e validação possui 15 variáveis, sendo 14 atributos e a outra o nosso target, ou seja, o valor que queremos predizer, totalizando 36421 registros no total.

Além disso, verificou-se que não existem dados faltantes na base. Caso esse fato não fosse verdadeiro, poderíamos atuar com a remoção dos exemplos que tivessem alguma featura nula, desde que a ausência de informações não estivesse correlacionada com algum tipo de comportamento específico que pudesse viesar os resultados do modelo.

Na base identifica-se que as features tratavam-se de variáveis contínuas (age, latitude, longitude, date_onset_symptoms, date_admission_hospital, date_confirmation e date_death_or_discharge) e categóricas (sex, country, lives_in_Wuhan, travel_history_location, chronic_disease_binary, travel_history_binary e label). Neste problema, como a variável resposta é categória, utilizaremos um modelo de classificação a partir de árvores de decisão e florestas aleatórias.

É importante ressaltar que nossos labels não são balanceados, isto é, a target de mortos representa apenas x dos dados, a target de tratamento x e em tratamento y. Estes casos podem ser tratados com técnicas de oversampling ou SMOTE, entre outras. Além disso, foram encontrados valores duplicados no cojunto de dados, que foram excluídos para a realização da análise.

Análise descritiva

Para entender as relações entre as variáveis e o estado do paciente (nosso Target), vamos analisar a distribuição da correlação entre as features numéricas. A Figura 2 apresenta um panorama geral da correlação. Nota-se que a variável date_admission_hospital e travel_history_dates possem correlação de com 79% e as variáveis travel_history_dates e longitude, representando -86%.

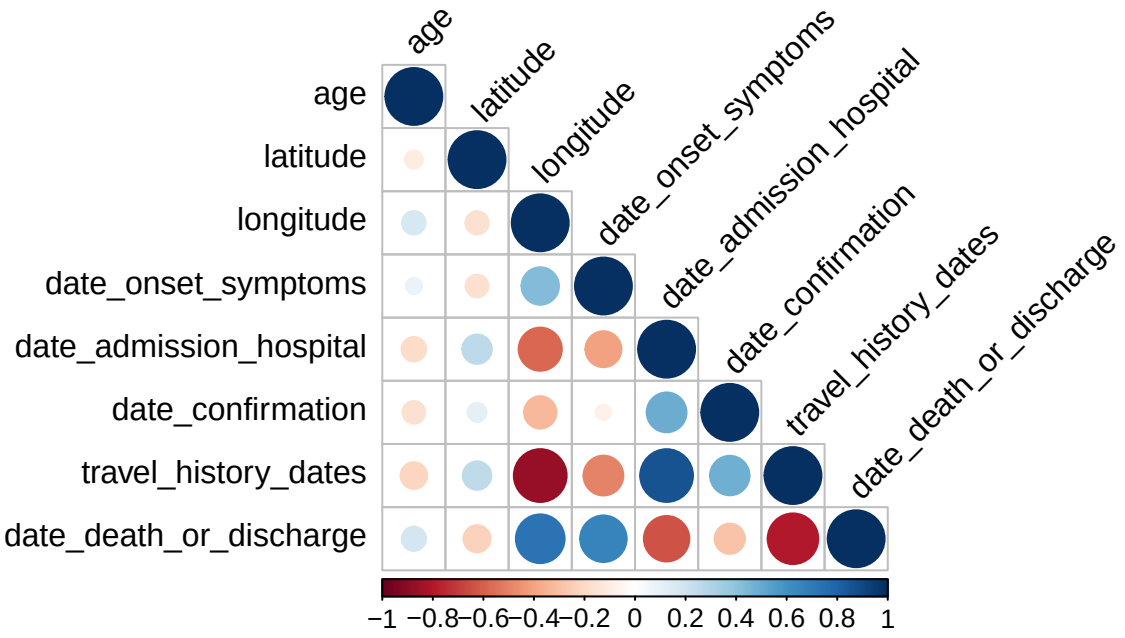

Figura 1: Correlações 2 a 2 das variáveis.

Metodologia

Para inferir o estado do paciente diagnosticado com COVID-19, é valido testar um modelo de árvore de decisão com todas as features e sem poda, esse será o baseline. Além disso, para verificar o desempenho do modelo, consideramos um conjunto de validação retirando uma amostra aleatória de 20% do dataset de treinamento. Desta forma, agora o conjunto de treinamento possui 24282 observações, enquanto validação possui 6069.

Tabela 1: Matriz de confusão para Treino

	dead	onTreatment	recovered
dead	1	0.00	0.00
onTreatment	0	0.97	0.03
recovered	0	0.12	0.88

Tabela 2: Matriz de confusão para Validação

	dead	onTreatment	recovered
dead	0.98	0.01	0.00
onTreatment	0.00	0.88	0.12
recovered	0.00	0.33	0.67

Tabela 3: Matriz de confusão para Teste

	dead	onTreatment	recovered
dead	0.99	0.01	0.00
onTreatment	0.00	0.87	0.13
recovered	0.00	0.36	0.64

Baseline

Inicialmente, consideramos o modelo baseline sem quaisquer transformações nos dados ou técnicas de reamostragem para verificar como o algoritmo se adapta as informações que temos em mãos. A Tabela 3 reportam a matriz de confusão para os conjuntos de treino e validação, respectivamente. É possível observar uma boa performance do modelo, onde os maiores erros estão nas classificações entre "onTreatment" e "recovered". A acurácia balanceada para o treino foi de 95%, para a validação 84% enquanto no Teste a acurácia balanceada foi 83% . É interessante observar como a performance da árvore de decisão foi alta nos conjuntos de treino e validação mesmo sem nenhuma técnica associada, porém não houve o mesmo comportamento no dataset de teste, onde a acurácia diminuiu significativamente, principalmente nas classes desbalanceadas. Associamos este tipo de desempenho ao overfittig.

Tamanho das árvores

Quanto maior o valor de profundidade da árvore, mais nós ela terá e mais segmentada ela será, fazendo classificações ainda mais assertivas. Porém, podemos observar a partir da Figura 2 que a acurácia balanceada atinge um platô muito rápido. Desta forma, a profundidade da árvore não precisa ser muito alta, o melhor resultado, a partir do conceito de parcimônia, foi para maxdepth = 8.

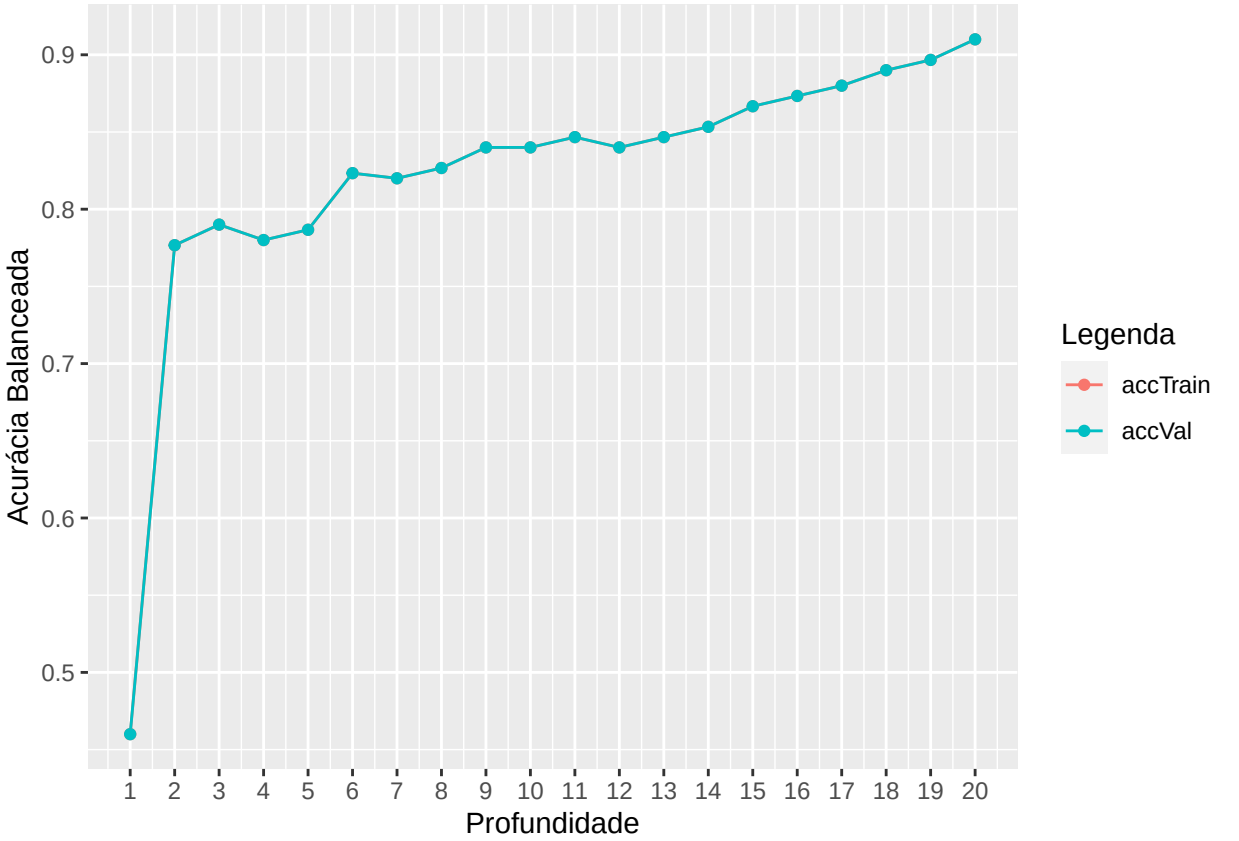

Figura 2: Acurácia balanceada do treino e validação.

Utilizando o modelo baseline com profundidade da árvore igual a 8, os resultados foram bastante satisfatórios, conforme representado na Tabela 4. Comparado ao resultado anterior, é possível identificar que aumentando a profundidade o modelo passou a errar menos na classificação "dead", desta forma, tivemos uma acurácia ainda maior no conjunto de teste, alcançando 84%, matendo a performance dos modelos de validação e teste. Entretanto, o modelo ainda não consegue diferenciar corretamente a diferente entre as classes "onTreatmente" e "recoverd".

Tabela 4: Matriz de confusão para Teste

	dead	onTreatment	recovered
dead	0.99	0.01	0.00
onTreatment	0.00	0.92	0.08
recovered	0.00	0.37	0.63

Seleção das features

Para melhorar os resultados do modelo, podemos selecionar as features que mais contribuem para a predição. Ao aplicar a árvore de decisão, podemos verificar quais atributos foram mais importantes para a classificação preditiva. Desta forma, as features mais relevantes para cada modelo em ordem crescente (baseline e max_depth=8) foram:

- Feature selection 1: date_death_or_discharge, longitude, date_admission_hospital, country, tra-

vel_history_dates, lives_in_Wuhan, date_confirmation, age, latitude, sex, travel_history_location, data_onset_symptoms e travel_history_binary.

- Feature selection 2: date_death_or_discharge, date_admission_hospital, country, longitude, travel_history_dates, lives_in_Wuhan,age, date_confirmation, sex, latitude, travel_history_location e date_onset_symptoms e travel_history_binary.

Desta forma, para o primeiro conjunto desconsideramos as variáveis sex, travel_history_location, data_onset_symptoms e travel_history_binary. Já no segundo as features sex, latitude, travel_history_location, data_onset_symptoms e travel_history_binary foram descartadas por apresentarem importância relativa menor do que 0,03.

Tabela 5: Matriz de confusão para Teste do modelo com a combinação de features 2.

	dead	onTreatment	recovered
dead	0.99	0.01	0.00
onTreatment	0.00	0.99	0.01
recovered	0.00	0.48	0.52

A partir do primeiro conjunto, podemos observar que o modelo alcançou acurácia balanceada de 83%, enquanto a acurácia balanceada no segundo conjunto foi de 84% nos dados de validação. Desta forma, será considerado para o teste o conjunto e parâmetros que tiveram melhores performance, isto é, a combinação de features 2.

Os resultados para o conjunto de teste estão apresentados na Tabela 5, garantindo 83,3% de acurácia no modelo. Novamente, ainda existem muitos erros na diferenciação entre "onTreatment" e "recovered", enquanto a classe "dead" se mostra muito bem segmentada.

Floresta Aleatória

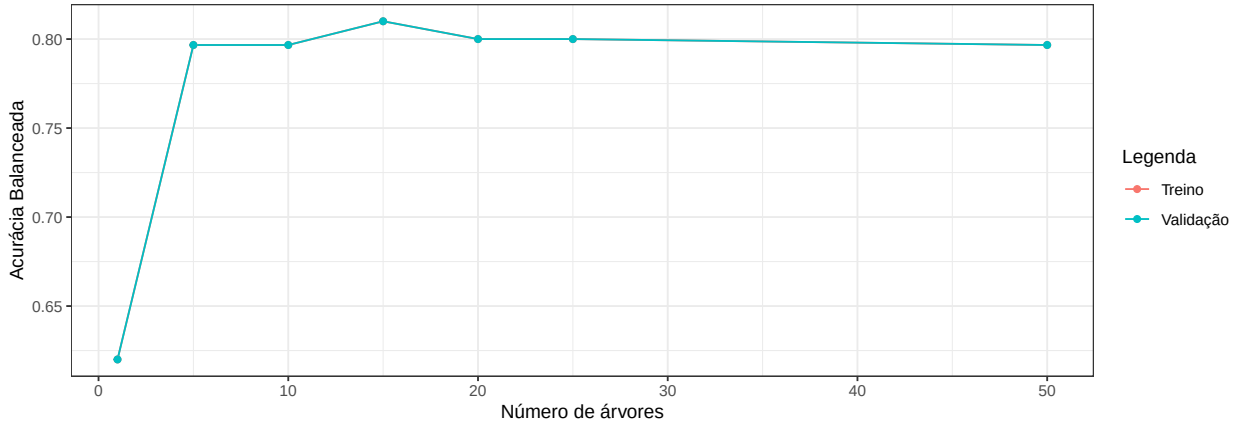

Figura 3: Acurácia balanceada para treino e validação - Floresta Aleatória.

Conclusão