Preparação dos dados

Os conjuntos de dados utilizados foram os relativos à admissão dos pacientes (`B1_Admission.csv`), dos exames (`B1_Exam.csv`) e das prescrições médicas (`B1_Prescription.csv`).

Devido ao grande volume dos dados e à limitação de memória do computador utilizado, decidiu-se por usar uma porção de 10% de cada dataset original, o que ainda resultou em cerca de 197.000 registros no dataset resultante, após o término do processo de preparação.

Foi dada uma atenção especial à variável `Discharge_Motive`, que indica o motivo pelo qual o paciente deixou a unidade de UTI. Os registros estavam pouco padronizados, e para resolver isso decidiu-se fazer um mapping dos valores, resultando em apenas 3 possíveis: `ALTA`, `OBITO` e `TRANSFERENCIA`.

Os registros dos 3 datasets originais foram agregados considerando os valores das variáveis `Hospital_ID` e `Patient_ID`, que posteriormente foram retiradas do dataset resultante, por serem irrelevantes para as tarefas de mineração.

Além disso, utilizou-se a técnica *one-hot encode* para tornar algumas variáveis textuais, como `Sex` e `Exam_Name`, em booleanos.

Ainda, descartou-se as variáveis `Weight`, `Height`, por serem em sua maioria dos registros vazias, `Skin_Color`, `Pharmacy_Assessment` e `Exam_Date` por serem consideradas não relevantes para as análises.

Finalmente, os registros com campos nulos foram descartados também, por considerar-se a quantidade de dados restantes suficiente para as tarefas.

Análise exploratória

Para a fase de análise exploratória, inicialmente verificou-se a distribuição dos motivos de alta, obtendo as quantidades:

Motivo	Quantidade
ALTA	185392
OBITO	11407
TRANSFERENCIA	206
DESISTENCIA	2

Com isso, relatou-se que nesse sentido o dataset encontrava-se desbalanceado, o que afetou como as análises posteriores foram realizadas.

Após isso, através de matrizes de correlação, explorou-se possíveis influencias entre as variáveis, onde as mais relevantes encontradas foram:

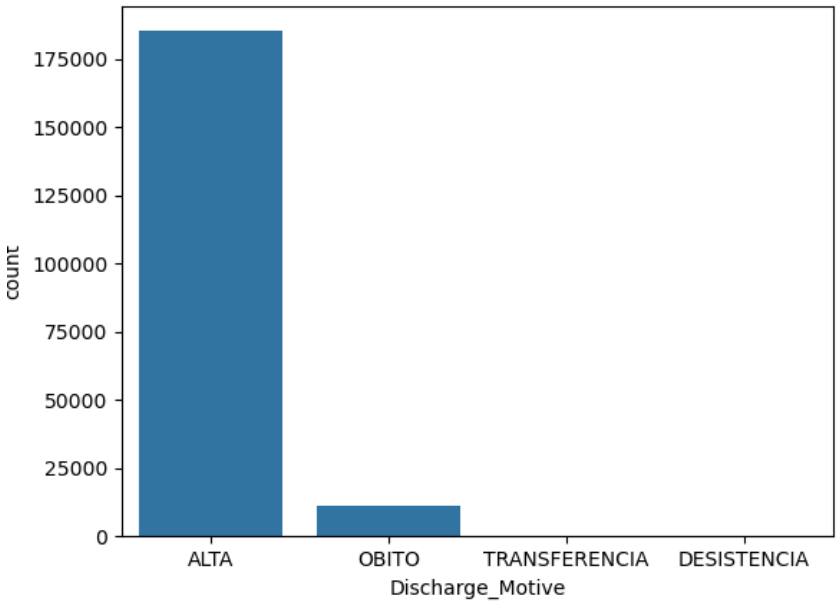

Figure 1: Motivos de alta

x_1	x_2	Coeficiente
Controlled	High_Alert	0.51
Public	IC	-0.38
Public	Emergency	0.37
Public	Surgical	0.40
Different_Drugs > 1	High_Alert	0.39
Different_Drugs > 1	Controlled	0.41
Obito	IC	0.2

Como pode-se constatar, não foi possível, inicialmente estabelecer uma relação direta entre uma variável e o óbito do paciente, entretanto foi possível constatar algumas relações, como:

- Pacientes que utilizam medicações controladas tendem a serem considerados de alto alerta.
- Pacientes que são internados em UTIs de hospitais públicos tendem a serem casos de emergência ou cirúrgicos.
- Por outro lado, pacientes que são internados em UTIs públicas não se encaixam em IC (cuidados intensivos), o que pode mostrar deficiência de infraestrutura nesses hospitais.
- Pacientes que usam mais de um tipo de medicamento tendem a ser de alto alerta e utilizarem drogas controladas.

A partir daqui, decidiu-se realizar uma tarefa de classificação para treinar um modelo que consiga prever quais pacientes tendem a vir a óbito, considerando algumas das variáveis disponíveis.

Classificação: previsão de óbito em pacientes

Objetivo

O objetivo deste processo foi desenvolver um modelo de classificação que prediga o desfecho de óbito em pacientes, com base em diversas variáveis médicas. A variável alvo foi definida como Obito, sendo 1 para óbito e 0 para os outros desfechos (alta ou transferência).

Metodologia

Os dados do dataset criado no processo de preparação foram carregados de um arquivo `.csv`.

Em seguida, criou-se uma variável nova no conjunto de dados chamada de `Obito`, que tem o valor 1 quando a `Discharge_Motive == OBITO` e 0 em qualquer outro caso.

Selecionou-se como features relevantes para o modelo as variáveis `Age`, `Complications`, `Interventions`, `High_Alert`, `Controlled`,`Different_Drugs`, `Stay_Length`, `Public`, `Surgical`, `Emergency` e `IC`.

E definiu-se como target a variável recém criada `Obito`.

Divisão e balanceamento dos dados

A divisão escolhida para treinamento e teste seguiu a proproção 80/20. Essa divisão foi feita utilizando o método `train_test_split` da biblioteca *scikit-learn*.

Como pode-se constatar na exploração inicial, os dados encontravam-se desbalanceados em relação ao número de altas e óbitos. Assim, decidiu-se utilizar o *SMOTE* para realizar o *Oversampling* dos casos de óbito e balancear os dados. O *SMOTE* cria exemplos sintéticos da classe minoritária (óbito) para garantir que o modelo seja capaz de aprender as características dessa classe.

O modelo de *XGBoost (XGBClassifier)* foi treinado utilizando os dados balanceados. A configuração do modelo foi ajustada para lidar com o desbalanceamento de classes, por meio do parâmetro `scale_pos_weight` (definido como 0.6). Esse parâmetro ajusta o peso das instâncias de óbito para que o modelo não se torne enviesado em favor das altas.

O modelo foi treinado com os dados balanceados, e a previsão foi realizada no conjunto de teste.

Avaliação do modelo

Após o treinamento, realizamos a avaliação do modelo usando as métricas de precisão, recall, f1-score e AUC-ROC. A precisão e o recall são especialmente importantes devido ao desbalanceamento das classes, já que queremos garantir que o modelo seja eficaz em identificar casos de óbito, sem aumentar muito os falsos positivos.

Os resultados foram:

Class	Precision	Recall	F1-Score	Support
Não-Óbito	1.00	0.99	0.99	37125
Óbito	0.88	0.97	0.92	2277

O modelo teve uma precisão de 0.88 para a classe óbito (classe 1), o que indica que, quando o modelo classifica um paciente como óbito, ele está correto em 88% dos casos. Já para as altas (classe 0), a precisão foi de 1.00, ou seja, todas as altas foram corretamente identificadas.

A AUC-ROC foi de 1.00, o que indica que o modelo tem um excelente desempenho na discriminação entre as classes (óbito e alta).

As métricas no treino mostraram um alto desempenho, com precisão de 0.97 para a classe não-óbito e 0.99 para óbito:

Class	Precision	Recall	F1-Score	Support
Não-Óbito	0.97	0.99	0.98	148475
Óbito	0.99	0.97	0.98	148475

Matriz de confusão

Foi gerada uma matriz de confusão objetivando mostrar o número de predições corretas e incorretas para cada classe. A matriz gerada foi visualizada com a ajuda de uma *heatmap*, onde pode-se observar que o modelo é capaz de distinguir bem os casos de óbito, embora tenha um número de falsos positivos (classificando altas como óbitos) em menor número, devido ao balanceamento de classes realizado.

	$Alta_{pred}$	$Obito_{pred}$
$Alta_{real}$	36813	312
$Obito_{real}$	73	2204

Validação cruzada

A validação cruzada foi realizada para avaliar a estabilidade do modelo. Utilizou-se a métrica precisão da classe 1 (óbito) como critério para avaliar o modelo em cada fold.

Com 8 folds, os resultados foram bastante consistentes, apresentando uma precisão média de 0.9907, e resultados individuais iguais a:

F_n	Precisão
1	0.99139396
2	0.99144831
3	0.98833644
4	0.99216352
5	0.9904502
6	0.99138499
7	0.99062037
8	0.98986301

Isso indica que o modelo não está apenas se ajustando aos dados de treino, mas também se comporta bem em dados diferentes.

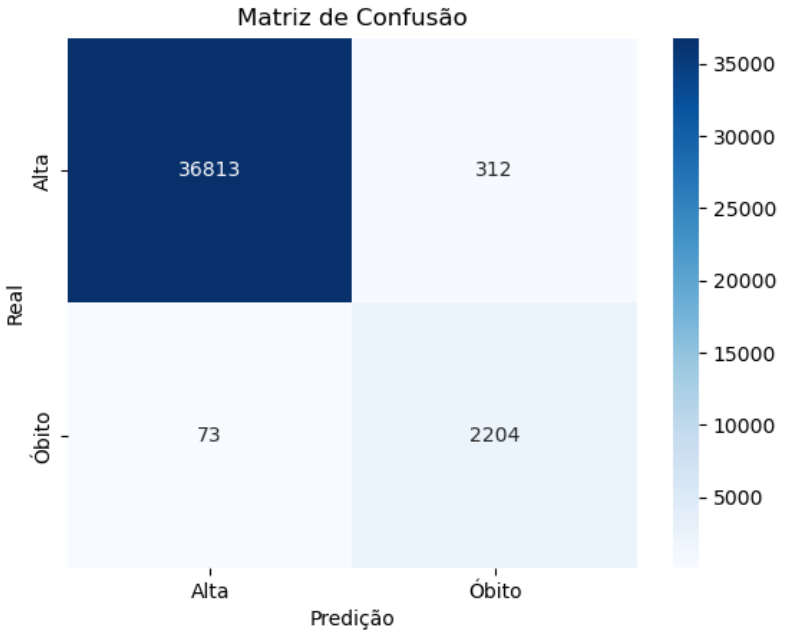

Figure 2: Matriz de confusão

Conclusão

O modelo *XGBoost* demonstrou excelente performance para prever o desfecho de óbito nos pacientes, com alta precisão e recall na classe minoritária (óbito). A utilização de *SMOTE* para balanceamento de classes foi essencial para melhorar a detecção de óbitos. Além disso, a validação cruzada confirmou a robustez do modelo, com precisão consistente em múltiplos folds.