PONTIFÍCIA UNIVERSIDADE CATÓLICA DE MINAS GERAIS
NÚCLEO DE EDUCAÇÃO A DISTÂNCIA
Pós-graduação *Lato Sensu* em Inteligência Artificial e Aprendizado de Máquina

Fellipe Augusto Oliveira Santos Lopes

APLICAÇÃO DE TÉCNICAS DE APRENDIZADO DE MÁQUINA PARA CLASSIFICAÇÃO INTERNACIONAL DE DOENÇAS

Belo Horizonte - MG
2020

Fellipe Augusto Oliveira Santos Lopes

APLICAÇÃO DE TÉCNICAS DE APRENDIZADO DE MÁQUINA PARA CLASSIFICAÇÃO INTERNACIONAL DE DOENÇAS

Trabalho de Conclusão de Curso apresentado ao Curso de Especialização em Inteligência Artificial e Aprendizado de Máquina como requisito parcial à obtenção do título de especialista.

Belo Horizonte - MG

2020

SUMÁRIO

1. Introdução

O Hospital das Clínicas de Marília (HC-FAMEMA) é uma autarquia do estado de São Paulo formado basicamente pelo Hospital das Clínicas unidade I (HC I), Hospital da Mulher (HC II), Hospital São Francisco (HC III) e o ambulatório de especialidades Mário Covas.

Além disso, integra a Rede Regional de Atenção a Saúde (RRAS – 10) do Departamento Regional de Saúde de Marília, estado de São Paulo, englobando 5 microrregiões de saúde: Marília, Assis, Ourinhos, Tupã e Adamantina.

1.1. Contextualização

O Pronto Atendimento (PA) do HC-FAMEMA recebe pacientes que precisam de cuidados imediatos. Quando o paciente chega, é realizado o processo de triagem que consiste em uma avaliação geral e a partir dessa encaminhar para a especialidade de acordo com as queixas e sintomas relatados e observados.

Dentro desse contexto, dependo do caso, o médico poderá solicitar exames com o intuito de auxiliar o diagnóstico e determinar a Classificação Internacional de Doenças (CID).

Por outro lado, esses profissionais contam com o prontuário eletrônico, desenvolvido pelo Departamento de Tecnologia da Informação (DTI) do próprio HC-FAMEMA. Através desse, é possível gerir as principais informações do paciente, inclusive solicitar e visualizar resultados de exames e informar o CID do atendimento.

1.2. O problema proposto

Primeiramente, o cenário atual parte do principio que o paciente já passou pelo processo de triagem e dessa maneira sabe-se qual especialidade vai tomar conta do seu caso. Alguns exemplos de especialidades médicas são: cardiologista, clínico geral, ortopedista dentre outros.

Baseado em conhecimento de domínio, o médico determina o CID principal e ocasionalmente os CIDs secundários a partir, mas não somente, do resultado do exame de sangue.

Acontece que, em certos momentos, como no caso de uma epidemia de dengue, o PA fica sobrecarregado o que acarreta em escassez de profissionais e aumenta o tempo de atendimento aos novos pacientes entrantes.

A princípio, seria interessante aplicar técnicas de aprendizado supervisionado para verificar a possibilidade de classificar apenas o CID principal utilizando os dados contidos no resultado do exame de sangue, nesse caso, o hemograma completo.

Em virtude disso, espera-se que o processo de diagnóstico de doenças seja acelerado e direcionado. Dessa forma, o médico poderá iniciar mais rapidamente o tratamento e os procedimentos necessários ao paciente.

Para tanto, foram fornecidos os dados dos últimos cincos anos referentes a resultados de hemogramas de pacientes atendidos no PA e com a devida classificação de CIDs.

2. Coleta de Dados

Antes de tudo, é importante frisar que a coleta dos dados foi autorizada pela coordenação do DTI, os dados foram extraídos do banco de dados principal do prontuário eletrônico e disponibilizados no formato.

Os dados foram entregues separadamente, seguindo as três separações lógicas em relação ao domínio do problema, sendo essas: atendimento, resultado do exame e CID, descritos nas próximas tabelas.

Tabela 1 – Estrutura dos dados do atendimento

Nome da coluna/campo	Descrição	Tipo
SEQ_REQUISICAO	Identificador: número sequencial da requisição de exame no atendimento.	Int64
SEQ_REQUISICAO_ITEM	Identificador: número sequencial do item da requisição de exame.	Int64
DAT_NASCIMENTO	Data de nascimento do paciente.	Object
FLG_SEXO	Sexo do paciente poderá assumir os valores: M, F e I.	Object
FLG_GESTANTE	Paciente em período gestacional. Possíveis valores S ou N. Exclusivo do sexo feminino.	Object
SEQ_CID	Identificador: número sequencial do CID.	Int64

Fonte: elaborado pelo autor.

Tabela 2 - Estrutura dos dados do resultado de exame

Nome da coluna/campo	Descrição	Tipo
SEQ_REQUISICAO	Identificador: número sequencial da requisição de exame.	Int64
SEQ_REQUISICAO_ITEM	Identificador: número sequencial do item da requisição de exame.	Int64
SEQ_EXAME_DETALHE	Identificador: número sequencial do nome do campo do detalhe do exame.	Int64
NOM_EXAME_DETALHE	Nome do campo do detalhe do exame.	Int64
DES_UNIDADE_MEDIDA	Unidade de medida associado ao valor do detalhe do exame.	Object
VLR_RESULTADO	Valor do detalhe do resultado do exame.	Object

Fonte: elaborado pelo autor.

Tabela 3 – Estrutura dos dados do CID

Nome da coluna/campo	Descrição	Tipo
SEQ_CID	Identificador: sequencial do CID.	Int64
COD_CID	Código do CID.	Object
NOM_CID	Nome do CID.	Object
DES_CAPITULO	Descrição do capítulo que o CID pertence.	Object
COD_ABRAGENCIA	Código de abrangência que o CID pertence.	Object

Fonte: elaborado pelo autor.

3. Processamento/Tratamento de Dados

Primeiramente, é importante salientar que os dados fornecidos tem origem de um banco de dados relacional e de natureza transacional, por consequência são considerados brutos e necessita-se de tratamento e pré-processamento antes da aplicação das técnicas de aprendizado supervisionado.

A tabela 4 demonstra a quantidade de registros e os respectivos arquivos para cada entidade.

Tabela 4 - Entidades

Entidade	Total de registros	Arquivos
Atendimento	60.647	atendimentos_5_anos.csv
Resultado de exame	9.076.191	resultados_exames_detalhes_5_anos_I.csv resultados_exames_detalhes_5_anos_II.csv resultados_exames_detalhes_5_anos_III.csv
CID	12.367	cids.csv

Fonte: elaborado pelo autor.

A relação entre atendimento e CID é um para um, dessa maneira suas colunas foram unidas, através da operação de join, em um dataframe que foi exportado com o nome de arquivo atendimentos.csv. Além disso, criou-se a coluna GRUPO_CID derivada de COD_CID, para obter um nível maior de granularidade.

Cada linha em resultado de exame representa somente uma característica do hemograma, além disso, esse possui 24 características que o descreve e isso justifica o número alto de quantidade de registros.

Nesse cenário, sabe-se que cada atendimento possui apenas um hemograma e assim, para efeito de consistência, foi realizado uma operação de inner join. Dessa forma, garante-se que realmente um atendimento tem um exame e vice-versa.

A coluna que descreve o nome de determinada característica do hemograma é a NOM_EXAME_DETALHE, retirou-se de seus valores diversos pontos que serviam somente para alinhamento de impressão.

Essa coluna tinha os seguintes valores distintos: Eritrócitos, Hemoglobina, Hematócrito, VCM, CHBCM, HBCM, comentários, Leucócitos, Mielócitos, Metamielócitos, Bastonetes, Segmentados, Eosinófilos, Linfócitos típicos, Linfócitos atípicos, Monócitos, total, OBS 1, OBS 2, OBS 3 e Plaquetas.

De acordo com a imagem 1, percebeu-se que o valor informado de cada característica encontrava-se na mesma linha só que na coluna VLR_RESULTADO.

Imagem 1 – Exames detalhes

```
In [31]: resultados_df[['NOM_EXAME_DETALHE','VLR_RESULTADO']].head()
```

Out[31]:

SEQ_REQUISICAO_ITEM	NOM_EXAME_DETALHE	VLR_RESULTADO
6112349	Eritrocitos	2,26
6112349	Hemoglobina	7,2
6112349	Hematocrito	21,40
6112349	VCM	94,70
6112349	CHbCM	33,80

Fonte: elaborado pelo autor.

Esses dados foram transpostos em um novo dataframe em que cada característica virou uma coluna com seus respectivos valores, imagem 2.

Imagem 2 – Dados transpostos

Eritrocitos	Hemoglobina	Hematocrito	VCM	CHbCM	HbCM	Comentarios	LEUCOCITOS	Mielocitos	Metamielocitos	Bastonetes	S
2,26	7,2	21,40	94,70	33,80	32,50	Normocitica, normocromica	7980	0	0	0	
5,45	16,40	46,00	84,40	35,70	30,10	Normocitica, normocromica	7640	0	0	0	
4,48	14,20	41,00	91,52	34,63	31,7	NEUTROFILOS: Ausencia de alteracoes degenerati...	8.750	00	00	02	
5,03	15,90	42,90	85,29	37,06	31,61	NEUTROFILOS: Ausencia de alteracoes degenerati...	10.850	00	00	00	
3,58	10,20	31,20	87,15	32,69	28,49	Leve microcitose.	8.590	00	00	00	

Fonte: elaborado pelo autor.

Todas as colunas textuais (comentários, OBS 1, OBS 2 e OBS3) foram deletadas, pois não se aplicou técnicas de linguagem natural, em contrapartida, manteve-se as que continham informação numérica em sua maioria.

A imagem 3 mostra a quantidade de valores ausentes encontrados no novo conjunto de dados.

Imagem 3 – Dados ausentes

Eritrocitos	6
Hemoglobina	6
Hematocrito	6
VCM	5
CHbCM	5
HbCM	5
LEUCOCITOS	8
Mielocitos	1142
Metamielocitos	1108
Bastonetes	836
Segmentados	7
Eosinofilos	140
Basofilos	808
Linfocitos Tipicos	11
Linfocitos Atipicos	1085
Monocitos	10
Total	209
Plaquetas	10

Fonte: elaborado pelo autor.

Por tratar-se de informações sobre os componentes do sangue de pacientes e que estão de alguma forma associadas ao seu estado clínico, optou-se por apagar os dados ausentes ao invés de alimentá-los por meio de uma estimativa. Além disso, não havia correlações significativas entre essas variáveis explicativas para usar a abordagem de preenchimento por regressão linear.

Depois disso, removeu-se diversos caracteres não numéricos que faziam parte de seus valores, assim como as vírgulas substituídas por pontos, entretanto, na coluna LEUCOCITOS tanto as vírgulas quanto os pontos foram retirados.

Após todo esse tratamento executou-se a conversão dessas colunas de object para float e esses dados foram salvos no arquivo detalhes_resultados_pre_processados.csv.

4. Análise e Exploração dos Dados

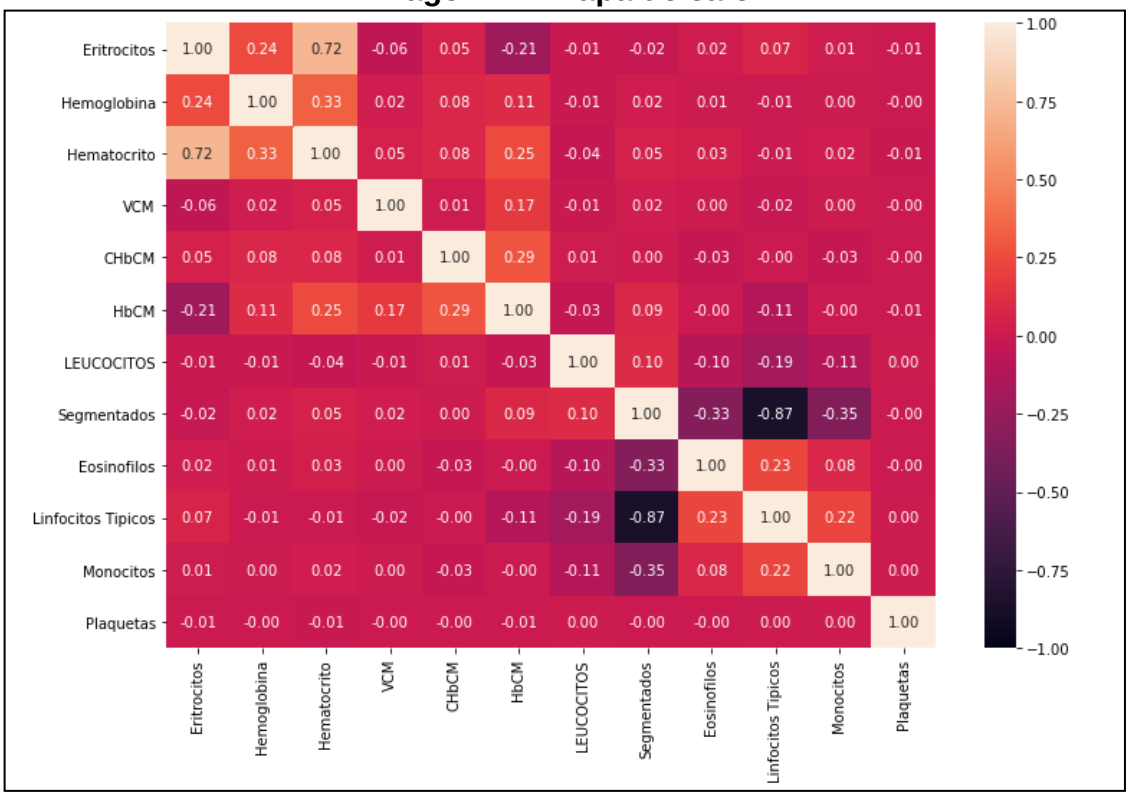
Em primeiro lugar, empregou-se uma análise descritiva sobre as variáveis explicativas, que representam as características do resultado do hemograma.

Por consequência, observou-se a alta ocorrência de zeros, em aproximadamente 75% dos Mielócitos, Metamielócitos e Linfócitos Atípicos; 50% dos Bastonetes e Basófilos. Além disso, verificou-se que cerca de 75% da variável total estava preenchida com valores 100.

Por causa dessa alta concentração de observações em regiões específicas dessas variáveis, conforme citado anteriormente, decidiu-se por eliminá-las do conjunto de dados, pois apresentaram baixa variabilidade.

Em outro aspecto, a análise de correlação de Pearson revelou que a maioria das variáveis são independentes, sendo que, apenas o Eritrócitos versus Hematócrito apresentaram uma correlação forte positiva e Linfócitos Típicos versus Segmentados uma correlação forte negativa, conforme destacado no mapa de calor abaixo.

Imagem 4 – Mapa de calor



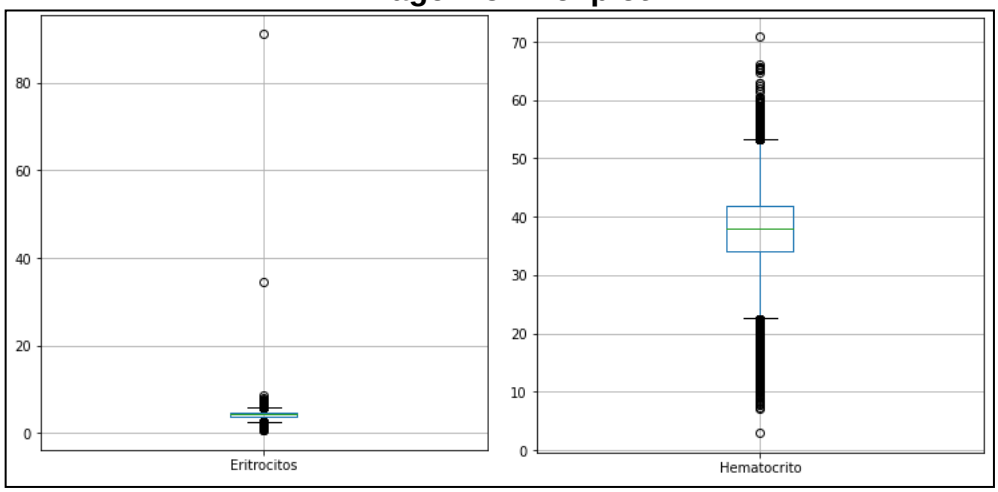
Fonte: elaborado pelo autor.

Apesar desses dois pares de variáveis retratarem alta dependência entre si, esses foram mantidos, pois mais pra frente aplicou-se uma técnica de seleção de features com o intuito de indicar as melhores variáveis explicativas para treinamento de modelos de aprendizado supervisionado.

Em relação aos valores que se diferenciam drasticamente dos outros, os outliers, todas as variáveis demonstraram valores com essa característica. Isso significa que alterações na saúde do paciente tende a refletir em grandes alterações de seus componentes sanguíneos tanto para mais quanto para menos.

O boxplot a seguir, expressa a presença de outliers nos Eritrócitos e Hematócritos.

Imagem 5 - Boxplot



Fonte: elaborado pelo autor.

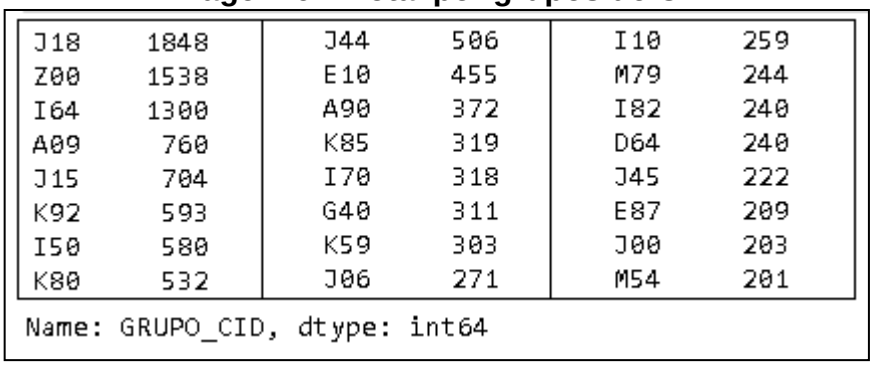
É importante salientar que removeu-se apenas os outliers isolados, em outras palavras, no caso dos Eritrócitos retirou-se os pontos 91 e 34,5, essa abordagem foi adotada em todas as variáveis. Dessa maneira, manteve-se a variabilidade dos componentes sanguíneos quando houve comprometimento da saúde do paciente de modo que também valores exorbitantes foram excluídos.

Depois disso, constatou-se que no conjunto de dados havia códigos de abrangências de CIDs fora do escopo de doenças, sendo assim, filtrou-se apenas as abrangências: A00-B99, D50-D89, E00-E90, G00-G99, J00-J99, K00-K93, M00-M99, I00-I99.

Além disso, houve outra consistência realizada, verificou-se que todos os atendimentos tinham resultados de hemograma e vice-versa.

Preferiu-se agrupar os dados de acordo com o GRUPO_CID, dessa vez para averiguar a quantidade de registros em cada um, a imagem 6 destaca os códigos de grupos de CIDs que obtiveram quantidade superior a 200 observações.

Imagem 6 – Total por grupos de CID



J18	1848	J44	506	I10	259
Z00	1538	E10	455	M79	244
I64	1300	A90	372	I82	240
A09	760	K85	319	D64	240
J15	704	I70	318	J45	222
K92	593	G40	311	E87	209
I50	580	K59	303	J00	203
K80	532	J06	271	M54	201

Name: GRUPO_CID, dtype: int64

Fonte: elaborado pelo autor.

De acordo com a investigação nos dados, descobriu-se que o código do grupo Z00 tem 1538 registros, esse refere-se ao exame geral e investigação de pessoas sem queixas ou diagnóstico relatado. Dessa maneira, a seguinte hipótese surgiu: seria possível determinar qualquer outro grupo de CID, que classifica uma doença, somente com observações desse grupo em conjunto com Z00?

Para confirmar essa hipótese o Z00 serviu de apoio para determinar que não houvesse algo de errado com o paciente. E a partir disso, cada grupo de CID foi preparado para ser confrontado com Z00, para tanto criou-se uma coluna para cada um com valores entre 0 e 1 para indicar respectivamente inexistência e existência desses no atendimento.

Na sequência, algumas colunas irrelevantes ao problema foram descartadas, sendo essas: SEQ_REQUISICAO, FLG_SEXO, DAT_NASCIMENTO, FLG_GESTANTE, SEQ_CID, COD_ABRANGENCIA, DES_CAPITULO, COD_CID, NOM_CID, SEQ_REQUISICAO_ITEM. Para finalizar, gerou-se outro arquivo, o input.csv, para entrada de dados nos modelos de aprendizado de máquina da próxima seção.

5. Criação de Modelos de Machine Learning

Antes de tudo é necessário esclarecer em alto nível que o seguinte fluxo foi adotado: filtragem das amostras, separação de dados em treino e teste, seleção de features, normalização de dados, criação do conjunto de validação nos dados de treino, reamostragem para equilíbrio das classes de treinamento, busca otimizada de hiperparâmetros, escolha dos melhores modelos e por último os cálculos de métricas para avaliação nos dados de treino e teste.

Cada amostragem foi diferenciada em relação a cada grupo de CID, sendo que cada uma recebeu também as observações pertencentes ao grupo de CID de exame de rotina (Z00), conforme evidenciado na imagem 7.

Imagem 7 – Separação de amostras

```
for grupo_cid in grupos_cids:

    df_amostragem = df[(df[nome_coluna_classes] == grupo_cid) | (df[nome_coluna_classes] == cid_exame_rotina)]\
                            .drop([nome_coluna_classes],axis=1)

    y = df_amostragem[grupo_cid]

    x = df_amostragem.drop(grupos_cids,axis=1)\
                     .drop(cid_exame_rotina,axis=1)
```

Fonte: elaborado pelo autor.

De acordo com a imagem 8, 70% dos dados foram destinados aleatoriamente ao conjunto de treino enquanto que os 30% restante para teste.

Imagem 8 – Separação dados de treino e teste

```
X_train, X_test, y_train, y_test = train_test_split(x, y, test_size=.3, random_state=42)
```

Fonte: elaborado pelo autor.

Depois disso, aplicou-se o modelo de árvore de decisão em conjunto com a busca de hiperparâmetros com a finalidade de selecionar variáveis que melhor explicam a ocorrência de determinado grupo de CID. O modelo foi instanciado com o hiperparâmetro de balanceamento de classes, pois todas as amostras de grupos de CID apresentaram o problema de desbalanceamento.

Para seleção de variáveis, buscou-se por árvores com profundidade de 1 até 10, que tivessem de 30 até 180 observações em cada nó de decisão e com critério de separação gini ou entropia.

Ao final, a melhor árvore de decisão, juntamente com os hiperparâmetros descobertos, foi treinada, testada e avaliada por métricas que foram salvas para posterior comparação em relação a outras técnicas de aprendizagem supervisionada.

A figura 9 demonstra os passos até então descritos para o algoritmo de árvore de decisão.

Imagem 9 – Árvore de decisão

```
max_depth = np.array([1,2,3,4,5,6,7,8,9,10])
criterio = np.array(['gini','entropy'])
min_samples_leaf = np.array([30,60,90,120,150,180])
valores_grid = {'max_depth':max_depth,'criterion': criterio,'min_samples_leaf': min_samples_leaf}

modelo_simples = DecisionTreeClassifier(class_weight='balanced')
grid_decision_tree = GridSearchCV(estimator = modelo_simples,param_grid=valores_grid,cv=3,scoring='f1',n_jobs=-1)
grid_decision_tree.fit(X_train,y_train)
melhor_modelo = grid_decision_tree.best_estimator_

select_model = SelectFromModel(melhor_modelo, prefit=True)
features_selecionadas = list(x.columns[select_model.get_support()])

X_train_sel = X_train[features_selecionadas]
X_test_sel = X_test[features_selecionadas]

(f1_train,precision_train,
 recall_train,f1_test,
 precision_test,
 recall_test) = executar_treinamento_teste(melhor_modelo,
                                           X_train_sel,
                                           y_train,
                                           X_test_sel,
                                           y_test)
```

Fonte: elaborado pelo autor.

Como citado outrora, já que existiu o problema de desbalanceamento de classes é importante destacar as métricas calculadas e apropriadas para avaliação dos modelos, sendo essas: precisão, recuperação e pontuação F1.

Ainda nesse âmbito, vale à ressalva que os modelos de Naive Bayes, KNN e redes neurais foram treinados e avaliados com o objetivo de comparar suas pontuações F1, incluído o modelo seleção de variáveis, a árvore de decisão.

Antes disso, precisou-se normalizar os dados utilizando a técnica de escala em relação os valores mínimos e máximos. É importante frisar que não optou-se pela padronização dos valores, pois nem todas as variáveis tinham distribuição normal. Ainda sobre esse assunto, isso foi necessário, senão os modelos sofreriam com a diferença de grandezas dos valores entre as variáveis.

Em outro aspecto, para não ocasionar o problema de vazamento de dados e assim impactando o desempenho da validação cruzada realizada dentro da busca por hiperparâmetros, houve a personalização das partições do conjunto de dados de treinamento.

Conforme ilustrado na imagem 10, primeiramente subdividiu-se 20% dos dados de treinamento para validação e posteriormente aplicou-se a técnica de reamostragem Smote apenas nos 80% restantes de treinamento, para aumentar o número de exemplos do grupo do CID minoritário. Dessa forma, garantiu-se o treinamento em dados balanceados à medida que a validação fosse realizada com grupos distribuídos semelhantemente ao mundo real.

Imagem 10 – Validação cruzada pré-definida

```
X_train_fold, X_val_fold, y_train_fold, y_val_fold = train_test_split(X_train_norm,
                                                      y_train,
                                                      train_size = 0.8,
                                                      stratify = y_train,
                                                      random_state = 42)

manipulador_amostragem = SMOTE(sampling_strategy=1,random_state=42)

X_train_fold,y_train_fold = manipulador_amostragem.fit_resample(X_train_fold, y_train_fold)

X_train_grid = np.concatenate((X_train_fold, X_val_fold))
y_train_grid = np.concatenate((y_train_fold, y_val_fold))

split_index_train = [-1 for _ in X_train_fold]
split_index_val =   [1 for _ in X_val_fold]

split_index = split_index_train + split_index_val

pds = PredefinedSplit(test_fold = split_index)

valores_K = np.array([2,3,4,5,6,7,8,9,10])
calculo_distancia = ['minkowski','chebyshev']
valores_p = np.array([1,2,3,4])
valores_grid = {'n_neighbors': valores_K,'metric':calculo_distancia,'p':valores_p}
modelo = KNeighborsClassifier()

grid_knn = GridSearchCV(estimator=modelo,param_grid = valores_grid,cv=pds,n_jobs=-1,scoring='f1',refit=False)
```

Fonte: elaborado pelo autor.

A figura acima deixou claro também a procura dos hiperparâmetros do KNN, sendo o número de vizinhos mais próximos entre 2 até 10, utilizando às formulas de distâncias Minkowski ou Chebyshev. Além disso, o parâmetro p diz a respeito ao grau de elevação da distância na formula de Minkowski, especificamente quando p = 1 equivale à distância de Manhattan e p = 2 refere-se à distância Euclidiana.

O modelo de Naive Bayes, nesse caso o GaussianNB, foi treinado sem a busca otimizada de hiperparâmetros, conforme ilustrado na imagem 11.

Imagem 11 – Naive Bayes

```
manipulador_amostragem = SMOTE(sampling_strategy=1,random_state=42)
X_train_sample,y_train_sample = manipulador_amostragem.fit_resample(X_train_norm, y_train)

melhor_modelo = GaussianNB()

(f1_train,
 precision_train,
 recall_train,
 f1_test,precision_test,
 recall_test) = executar_treinamento_teste(melhor_modelo,X_train_sample,y_train_sample,X_test_norm,y_test)
```

Fonte: elaborado pelo autor.

A imagem 12, mostra que a rede neural, MPLClassifier, teve a busca de hiperparâmetros configurada entre 1 até 5 camadas escondidas de 10 neurônios cada e com as funções de ativação, logistic, relu e tanh.

Imagem 12 – Redes neurais

```
valores_grid={'hidden_layer_sizes': [(10,),(10,10), (10,10,10),(10,10,10,10),(10,10,10,10,10)],
              'activation': ["logistic", "relu", "Tanh"]}

grid_nn = GridSearchCV(estimator=MLPClassifier(),param_grid = valores_grid,cv=pds,n_jobs=-1,scoring='f1',refit=False)

grid_nn.fit(X_train_grid,y_train_grid)

melhor_modelo = MLPClassifier(**grid_nn.best_params_)

(f1_train,
 precision_train,
 recall_train,
 f1_test,precision_test,
 recall_test) = executar_treinamento_teste(melhor_modelo, X_train_fold,y_train_fold,X_test_norm,y_test)
```

Fonte: elaborado pelo autor.

Depois dos cálculos das métricas de avaliação de cada modelo para cada grupo de CID, essas foram salvas em um dicionário de scores, imagem 13.

Imagem 13 – Dicionário de scores

```
scores['ALGORITMO'].append('NN')
scores[nome_coluna_classes].append(grupo_cid)
scores['F1_TREINO'].append(f1_train)
scores['PRECISION_TREINO'].append(precision_train)
scores['RECALL_TREINO'].append(recall_train)
scores['F1_TESTE'].append(f1_test)
scores['PRECISION_TESTE'].append(precision_test)
scores['RECALL_TESTE'].append(recall_test)
scores['FEATURES_SELECIONADAS'].append(features_selecionadas)
```

Fonte: elaborado pelo autor.

6. Apresentação dos Resultados

Após terem sido computadas todas as métricas de avaliação para cada grupo de CID versus Z00 para os melhores modelos dos quatros algoritmos de aprendizado supervisionado, apenas três ficaram com a pontuação F1 acima de 0,70 nos dados de teste. A imagem 14 mostra a classificação das pontuações F1 nos dados de teste.

Imagem 14 – Pontuações F1

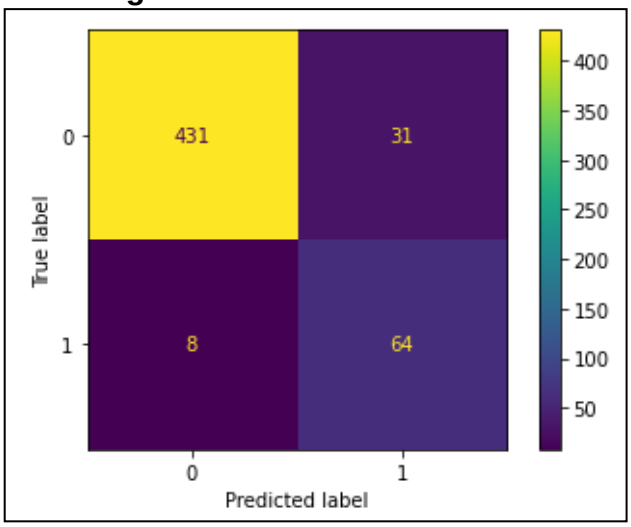
GRUPO_CID	ALGORITMO	F1_TREINO	F1_TESTE	FEATURES_SELECIONADAS
D64	ARVORE DE DECISÃO	0.764543	0.766467	[Hemoglobina]
J18	NN	0.729116	0.739195	[Eritrocitos, CHbCM, LEUCOCITOS, Eosinofilos, Linfocitos Tipicos, Monocitos]
A90	NAIVE BAYES	0.819312	0.707547	[Eritrocitos, VCM, LEUCOCITOS, Segmentados, Plaquetas]
K92	NN	0.704799	0.622517	[Eritrocitos]
J15	ARVORE DE DECISÃO	0.670213	0.611111	[Eritrocitos, Hemoglobina, Hematocrito, VCM, CHbCM, Segmentados, Eosinofilos, Linfocitos Tipicos, Monocitos, Plaquetas]
A09	NN	0.712230	0.600390	[Eritrocitos, VCM, CHbCM, HbCM, Segmentados, Eosinofilos, Linfocitos Tipicos, Monocitos, Plaquetas]
I64	NN	0.642151	0.588506	[Eritrocitos, Hemoglobina, Hematocrito, CHbCM, LEUCOCITOS, Segmentados, Eosinofilos, Linfocitos Tipicos, Plaquetas]
J44	NN	0.728485	0.562500	[Eritrocitos, Hemoglobina, Hematocrito, LEUCOCITOS, Segmentados, Eosinofilos, Linfocitos Tipicos, Monocitos]
I50	ARVORE DE DECISÃO	0.626219	0.524038	[Eritrocitos, Hemoglobina, Hematocrito, VCM, CHbCM, HbCM, LEUCOCITOS, Eosinofilos, Linfocitos Tipicos, Monocitos, Plaquetas]
K85	NAIVE BAYES	0.676129	0.488889	[Hemoglobina, VCM, CHbCM, Linfocitos Tipicos]

Fonte: elaborado pelo autor.

É importante salientar que 5 dos 10 primeiros modelos são de redes neurais (NN), entretanto apenas em J18 - Pneumonia por micro-organismo não especificada, que obteve-se pontuação maior que 0,70.

Apesar da primeira colocação, a árvore de decisão com D64 – Outras anemias, classificou que 8 pacientes estavam saudáveis sendo que tinham anemia, por outro lado, classificou 31 pacientes com anemia sendo que estavam saudáveis, conforme ilustrado na matriz de confusão na imagem 15.

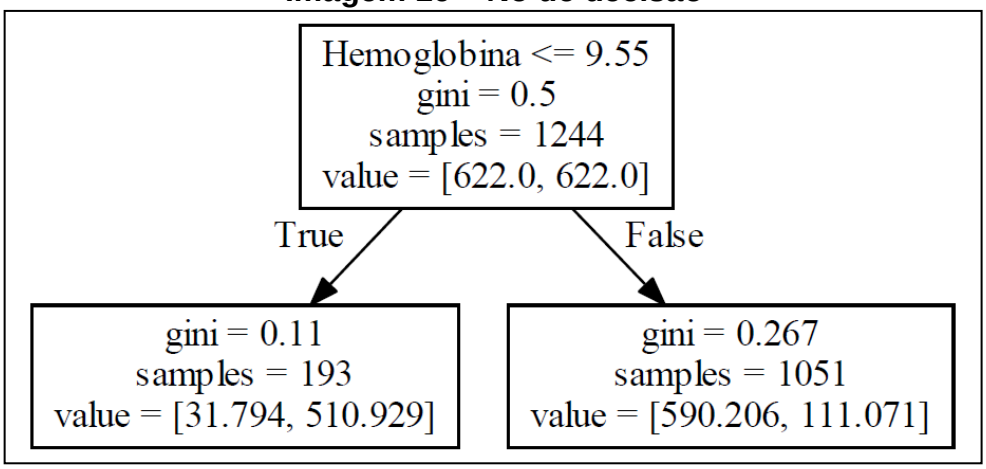
Imagem 15 – Matriz de confusão



Fonte: elaborado pelo autor.

De acordo com a imagem 16, o modelo aprendeu que se o paciente estivesse com hemoglobina menor ou igual a 9,55 g/dl seria classificado com anemia, ao contrário disso, estaria saudável.

Imagem 16 – Nó de decisão



Fonte: elaborado pelo autor.

Todavia o desempenho dos modelos treinados não foi aceitável, pois houve o risco de alguns pacientes não receberem o tratamento adequado quando deveriam e vice-versa.

Para contornar essa situação, ao invés de tentar prever diretamente a doença, seria melhor apresentar a probabilidade de o paciente pertencer a determinado grupo de CID. Dessa forma, a imagem 17, deixa claro que se a Hemoglobina for menor ou igual ao limiar esperado, maior será a probabilidade de possuir anemia.

Imagem 17 – Probabilidade de anemia

	Hemoglobina	Probabilidade Saudável	Probabilidade Anemia
0	4.5	0.058582	0.941418
1	5.7	0.058582	0.941418
2	6.9	0.058582	0.941418
3	11.6	0.841616	0.158384
4	13.9	0.841616	0.158384

Fonte: elaborado pelo autor.

De acordo com os resultados na imagem 18, percebeu-se que o modelo de rede neural utilizado para pneumonia é mais complexo e difícil de interpretar.

Imagem 18 – Probabilidade de pneumonia

	Eritrocitos	CHbCM	LEUCOCITOS	Eosinofilos	Linfocitos Tipicos	Monocitos	Probabilidade Saudável	Probabilidade Pneumonia
0	6.18	30.90	10070.0	0.0	3.0	2.0	0.111111	0.888889
1	4.28	31.90	9450.0	0.1	14.7	12.9	0.333333	0.666667
2	4.39	33.50	10800.0	0.0	12.0	6.0	0.000000	1.000000
3	4.29	34.33	11100.0	1.0	16.0	6.0	0.222222	0.777778
4	5.14	33.70	7500.0	3.0	39.0	5.0	0.888889	0.111111

Fonte: elaborado pelo autor.

A imagem 19 demonstra as probabilidades dos pacientes estarem com dengue, relacionado ao código de grupo de CID A90.

Imagem 19 – Probabilidade de dengue

	Eritrocitos	VCM	LEUCOCITOS	Segmentados	Plaquetas	Probabilidade Saudável	Probabilidade Dengue
568	4.79	77.24	4300.0	30.0	35000.0	0.001030	0.998970
569	4.29	78.00	5800.0	58.0	242000.0	0.654389	0.345611
570	3.72	91.67	8600.0	63.0	194000.0	0.969668	0.030332
571	3.74	92.00	5100.0	40.0	69000.0	0.136195	0.863805
572	4.21	90.02	18400.0	78.0	168000.0	0.999857	0.000143

Fonte: elaborado pelo autor.

Como resultado, esse relatório técnico demonstrou que não seria possível prever diretamente, com desempenho aceitável, o grupo de CID com base somente no resultado do hemograma, sendo assim, rejeitou-se a hipótese levantada na seção 4.

Entretanto, contribuiu-se com a entrega de três modelos capazes de informar quais são as probabilidades de o paciente estar com anemia, pneumonia ou dengue baseado nos dados históricos fornecidos.

Para trabalhos futuros espera-se aumentar o desempenho desses modelos coletando mais exemplos de casos dessas doenças e aprofundar mais a busca de hiperparâmetros. Além disso, analisar e explorar textos sobre os sintomas e históricos de pacientes através do processamento de linguagem natural.

7. Links

A tabela 5 contêm os links dos repositórios com os notebooks e conjuntos de dados utilizados e gerados por esse trabalho.

Tabela 5 - Links

Repositório	Descrição
Dados fornecidos pelo DTI do HC-FAMEMA.	www.kaggle.com/dataset/f59f3dc3440b5df0ed24bc81b2af42678c1b973e714c2efac2a26faeff415441
Jupyters notebooks em Python.	https://github.com/faosl/previsao_doencas

Fonte: elaborado pelo autor.