FAPESP COVID-19 Data Sharing BR - Uma análise de dados

Leonardo Thurler[1], **Rodrigo Veloso**[1]

[1]Universidade Federal Fluminense

Abstract. *The coronavirus disease 2019, a new respiratory disease spread exponentially all around the world. In this work data mining is done over the COVID-19 DataSharing BR repository, a specific database put together by hospitals from São Paulo in order to support research related to coronavirus. The main goal of this paper is to apply methodologies learned in the data mining course aiming for knowledge discover in the coronavirus topic. Association rules with high lift were mined and also an interesting rule associating a irregular exam result to patient hospitalization, specially when the patient has coronavirus. Clusters were obtained by k-Means and DBSCAN algorithms, the k-Means results were better than the ones of DBSCAN. Finally, 3 different classifiers were applied, and also the autolearn function of sckit learn, to solve a classification problem where one tries to predict the treatment time.*

Resumo. *A doença do coronavírus de 2019, um nova doença de natureza respiratória se espalhou de forma exponencial pelo mundo todo. Neste trabalho é realizada uma mineração de dados utilizando o repositório COVID-19 Data Sharing BR, uma base de dados disponibilizada por hospitais de São Paulo com objetivo de apoiar pesquisas relacionadas à doença do coronavírus. O objetivo principal deste trabalho é aplicar as metodologias aprendidas no curso de mineração de dados de forma a colocar em prática essas metodologias e tentar expandir o conhecimento em relação ao coronavírus. Foram extraídas regras de associação com alto lift e uma regra interessante relacionando o resultado anormal de um exame específico com a internação de pacientes com coronavírus. Foram obtidos clusters através dos algoritmos k-Means e DBSCAN, os resultados encontrados pelo k-Means se mostraram superiores ao DBSCAN. Por último, 3 classificadores diferentes foram aplicados, assim como a função de autolearn do scikit learn, para resolver um problema de classificação onde deseja-se prever a duração do atendimento de um paciente.*

1. Introdução

A doença do coronavírus de 2019 (COVID-19), uma nova doença, de natureza respiratória e com alto poder de contágio, foi primeiramente descrita e encontrada em Wuhan, China, em dezembro de 2019 [Yang 2020]. Coincidentemente, o período inicial da propagação dessa doença foi o mesmo período do *chunyun*, quando ocorre uma migração massiva devido ao festival de verão anual. O COVID-19 se espalhou de uma forma exponencial, não só na china onde se originou, mas em todo o mundo, levando a uma pandemia quase que sem precedentes na história moderna.

O Brasil é o país mais afetado pelo coronavírus na América Latina [Dana et al. 2020]. Em 26 de fevereiro de 2020 foi confirmado o primeiro caso de

COVID-19 no Brasil, um homem que havia viajado para Itália, um dos focos da pandemia. Em apenas um mês já haviam sido confirmados 2915 casos com 77 óbitos. O pico da pandemia, no Brasil, ocorreu em 29 de julho, onde foram contabilizadas 1595 mortes em um único dia e 69074 novos casos [Sanar 2020]. Esses números podem ser explicados como uma união entre a falta de infraestrutura adequada para garantir um tratamento eficiente aos infectados, falta de conhecimento prévio sobre a doença e a negligência por parte do governo federal, que não se preocupou em desenvolver políticas públicas para combater a pandemia [Sanar 2020].

Dada a gravidade da pandemia, qualquer possível contribuição pode ser fundamental para que vidas sejam salvas. A proposta desse trabalho é fazer uma mineração de dados utilizando repositório COVID-19 Data Sharing BR (CDSBR) [FAPESP 2020] para tentar extrair: regras de associação, modelos preditivos e identificar possíveis grupos de interesse, como uma forma de se obter conhecimento importante para o combate ao coronavírus. Esse trabalho também é importante como forma de colocar em prática os conhecimentos obtidos durante o curso de mineração de dados.

2. COVID-19 Data Sharing/BR

O repositório COVID-19 Data Sharing/BR é uma iniciativa da FAPESP em cooperação com a Universidade de São Paulo, e participação do Instituto Fleury, Hospital Sírio-Libanês e Hospital Israelita Albert Einstein, com o objetivo de disponibilizar dados relacionadas à COVID-19 que possam contribuir para pesquisas desta temática.

Nessa base de dados há informação sobre dados pessoais, exames clínicos realizados e desfechos de pacientes (quando disponível) dos hospitais do Instituto Fleury, Hospital Sírio-Libanês (HSL) e Hospital Israelita Albert Einstein. No total existem dados sobre 177,208 pacientes, 4,735,041 exames clínicos e 9,634 desfechos. O número de desfechos é notoriamente menor pois dentre as instituições a única que forneceu dados sobre os desfechos foi a HSL [Mello et al. 2020].

A análise feita neste trabalho será feita apenas nos dados relativos ao HSL. A justificativa dessa escolha se deve à maior riqueza dos dados fornecidos pelo hospital, uma vez que foi o único que forneceu informação sobre os desfechos e também como uma forma de reduzir o número de tuplas da análise.

Os dados do HSL estão distribuídos em 3 datasets principais. O primeiro relativo à dados demográficos dos pacientes, o segundo relativo aos exames que os pacientes realizaram e o terceiro relativo aos desfechos dos atendimentos. Cada um desses datasets serão descritos a seguir.

2.1. Pacientes

O dataset relativo aos pacientes contem informações demográficas sobre os pacientes. A base conta com 4273 pacientes diferentes, ou seja, contem 4273 tuplas. Cada tupla tem 7 atributos. Para melhor exemplificar a base, as 5 primeiras tuplas se encontram na figura 1.

O atributo 'ID_PACIENTE' é um identificador único para cada paciente ou tupla, 'IC_SEXO' é o sexo do paciente, é uma atributo categórico que pode ser 'M' para pacientes do sexo masculino e 'F' para pacientes do sexo feminino. Aproximadamente, 51% dos pacientes são do sexo feminino e 49% são do sexo masculino.

	ID_PACIENTE	IC_SEXO	AA_NASCIMENTO	CD_PAIS	CD_UF	CD_MUNICIPIO	CD_CEPREDUZIDO
0	3487791F44C34B421C932DC8616A8437	M	1963	BR	SP	MMMM	CCCC
1	0AD1FFA4419472256666A3445414F1F9	M	1969	BR	SP	SAO PAULO	CCCC
2	BE35D08CF3EF6F114E9935F6D72C49FA	M	1964	BR	SP	SAO PAULO	CCCC
3	962F4020D456AAB602B356E57238EF42	F	1959	BR	SP	SAO PAULO	CCCC
4	F983089DAC62E42124227C856AE3444C	M	1947	BR	SP	MMMM	CCCC

Figura 1. As 5 primeiras tuplas da base de dados relativa aos pacientes.

O atributo 'AA_NASCIMENTO' corresponde ao ano de nascimento do paciente. É mais intuitivo trabalhar com a idade do paciente do que com o ano de seu nascimento, por isso, uma transformação de ano para idade foi feita. A distribuição da idade dos pacientes se encontra na figura 2, a média de idade é de 44 anos com desvio padrão de 5 anos.

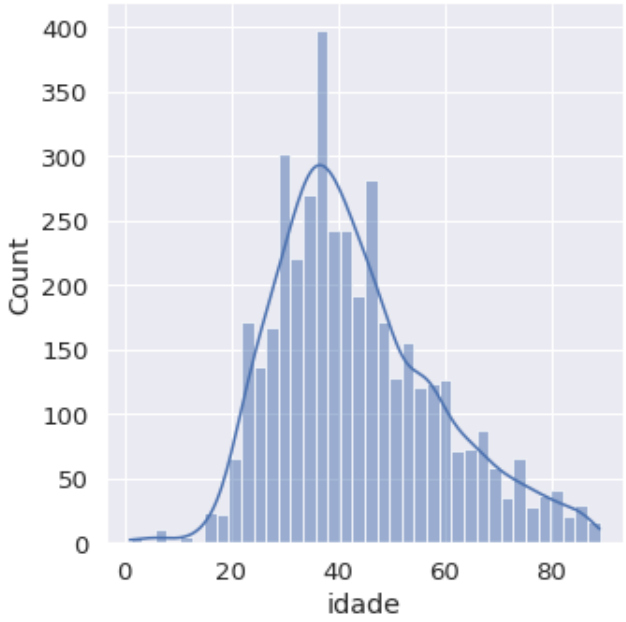

Figura 2. Distribuição da idade dos pacientes.

O atributo 'CD_PAIS' é relativo ao país de nascimento do paciente, é um atributo categórico que pode ser 'BR' para brasileiros que nasceram no Brasil ou 'XX' para pacientes que nasceram no exterior. O atributo 'CD_UF' corresponde ao estado de origem do paciente, o atributo 'CD_MUNICIPIO' ao município de origem e o atributo 'CD_CEPREDUZIDO' ao CEP da residência do paciente. Como esses atributos descritos nesse parágrafo não impactam a qualidade do atendimento, a gravidade do quadro clínico, ou possuem qualquer relação com características físicas ou biológicas dos pacientes, foram descartados das análises.

2.2. Desfechos

A base de dados com os desfechos contêm informações sobre o atendimento dos pacientes de uma forma geral. A base de dados possui 16957 tuplas, ou seja, 16957 desfechos de

atendimentos, com 8 atributos. O número de tuplas dessa base é maior que o número de tuplas de pacientes pois um paciente pode ser atendido mais de uma vez em dias diferentes. Para melhor exemplificar a base, as 5 primeiras tuplas da base se encontram na figura 3.

id_paciente	id_atendimento	dt_atendimento	de_tipo_atendimento	id_clinica	de_clinica	dt_desfecho	de_desfecho
3487791F44C34B421C932DC8616A8437	33277D918111011E48FABCD6FC09012B	2020-07-08	Pronto Atendimento	6	Ortopedia	2020-07-08	Alta médica melhorado
3487791F44C34B421C932DC8616A8437	2DF164AD7E51B0A2F52B6DB58F904A22	2020-05-08	Externo	20	Procedimentos	2020-05-08	Alta Administrativa
3487791F44C34B421C932DC8616A8437	11676CFEEF6A4CDD49B0A74D3F3A85EC	2020-05-25	Externo	20	Procedimentos	2020-05-25	Alta Administrativa
0AD1FFA4419472256666A3445414F1F9	70A834FEFF3562FCBA3888D592DA2609	2020-03-16	Externo	20	Procedimentos	2020-03-16	Alta Administrativa
3487791F44C34B421C932DC8616A8437	A3F017D035922D728E06B744511A6797	2020-05-11	Pronto Atendimento	42	CL Médica Síndromes Virais	2020-05-11	Alta médica melhorado

Figura 3. As 5 primeiras tuplas da base de dados relativa aos desfechos.

O atributo 'id_paciente' é o mesmo que o atributo 'ID_PACIENTE' na base de dados dos pacientes. O atributo 'id_atendimento' é um identificador do atendimento e portanto é único para cada tupla.

O atributo 'dt_atendimento' é relativo à data em que o atendimento teve início, 'dt_desfecho' é relativo à data do fim do atendimento, essa data foi transformada do formato mês-ano-dia para dias do ano de 2020. Por exemplo, um atendimento que se iniciou em 2020-1-10, se iniciou no 10^o dia de 2020.

O atributo 'id_clinica' contém a identificação e o atributo 'de_clinica' contém o tipo de clínica em que o atendimento foi feito. 'de_tipo_atendimento' é uma descrição do tipo atendimento, nesta base de dados existem 4 tipos de atendimento, são eles: pronto atendimento, externo, internado e ambulatorial.

'de_desfecho' é um atributo que descreve o tipo do desfecho do atendimento, existem 13 tipos de desfechos diferentes. Neste trabalho temos como foco apenas pacientes que vieram a óbito ou que tiveram alta médica ou administrativa, os outros tipos de desfecho eram inconclusivos, como 'transferência hospitalar', ou que não trazem informações importantes no contexto desse trabalho, como 'desistência do atendimento'. Após a remoção dos desfechos não interessantes, unimos os diferentes tipos de óbitos em um mesmo tipo 'O' e os diferentes tipos de alta em um mesmo tipo 'A', transformando esse atributo em um atributo categórico binário.

É notório que os atributos 'id_clinica', 'id_atendimento' e 'de_clinica' não são relevantes no contexto de uma análise de dados e foram removidos.

Para melhor conduzir a análise, seria interessante transformar a base de dados de modo a uma tupla ser relativa à um paciente e não à um atendimento, assim como na base dos pacientes. Com isso, a tarefa de combinar informações contidas em ambas as bases se torna mais simples. Para isso, as informações relativas à um mesmo paciente foram unidas, como ilustrado na figura 4. Foi incluído o atributo 'n_atendimentos', que indica o número de atendimentos diferentes realizados por um por paciente.

Pacientes com mais de um tipo de desfecho, ou seja, que foram atendidos mais de uma vez, receberam como desfecho único 'A' se não foram à óbito em algum atendimento, e desfecho único 'O' caso contrário. Uma estratégia semelhante foi feita com pacientes

id_paciente	dt_atendimento	de_tipo_atendimento	dt_desfecho	de_desfecho	n_atendimentos
BA74194B979B086198794D1470107709	194	Internado	204	O	1
8D265EED52D2877EC2F75B60FEEC4065	90	Internado	99	O	1
7B03A278AD0026D4E34D0D84453A3548	103,70	Internado,Externo	131,70	O,A	2
DE791245DF4FD631AD71B2B307F7835D	121,121	Internado,Pronto Atendimento	132,121	O,A	2
44DA670729D76F0264E6066E2E42CF45	80,80	Internado,Pronto Atendimento	142,82	O,A	2

Figura 4. As 5 primeiras tuplas relativas a base de desfechos transformada.

com mais de um tipo de atendimento, pacientes receberam como tipo de atendimento único o mais crítico entre todos os seus atendimentos.

Com os atributos 'dt_atendimento' e 'dt_desfecho' é possível calcular o tempo que durou o atendimento, informação mais rica do que as datas. Dessa forma, foi incluído na base de dados um atributo referente ao tempo de atendimento e os atributos que continham as datas foram removidos. Ao invés de manter o tempo de cada atendimento, os tempos de atendimento foram somados, resultando em um tempo total de atendimento único para cada paciente. A distribuição do tempo de atendimento se encontra na figura 5, a média foi de dias com desvio padrão de dias.

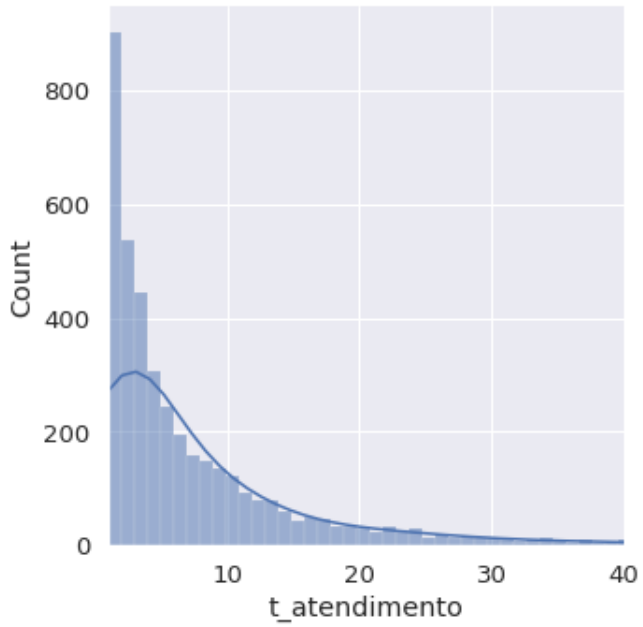

Figura 5. Distribuição do tempo de atendimento dos pacientes.

A forma final do dataset dos desfechos se encontra na figura 6.

	id_paciente	de_tipo_atendimento	de_desfecho	n_atendimentos	t_atendimento
400	BA74194B979B086198794D1470107709	Internado	O	1	11
511	8D265EED52D2877EC2F75B60FEEC4065	Internado	O	1	10
1070	7B03A278AD0026D4E34D0D84453A3548	Internado	O	2	30
1373	DE791245DF4FD631AD71B2B307F7835D	Internado	O	2	13
1803	44DA670729D76F0264E6066E2E42CF45	Internado	O	2	66

Figura 6. As 5 primeiras tuplas da forma final da base de desfechos transformada.

2.3. Exames

A base de dados de exames contêm informações sobre todos os exames clínicos realizados pelos pacientes durante os atendimentos. A base de dados possui 631352 tuplas, ou seja, 631352 exames clínicos, com 9 atributos.

Os atributos 'ID_PACIENTE' e 'ID_ATENDIMENTO' são os mesmos que os atributos 'id_paciente' e 'id_atendimento' da base de desfechos. Como o 'ID_ATENDIMENTO' não é útil na mineração de dados, foi removido da base.

'DT_COLETA' é um atributo que informa a data em que o exame foi realizado. 'DE_ORIGEM' informa onde o exame foi realizado. O 'CD_UNIDADE' é relativo a unidade de medida utilizada na metodologia para realizar o exame. Todo os atributos descritos nesse parágrafo são pouco úteis para as análises que serão realizadas e portanto foram removidos. Uma ilustração da base de dados após a remoção desses atributos se encontra na figura 7.

	ID_PACIENTE	DE_EXAME	DE_ANALITO	DE_RESULTADO	DE_VALOR_REFERENCIA
0	3487791F44C34B421C932DC8616A8437	Fosfatase Alcalina	Fosfatase Alcalina	106	40 - 129
1	3487791F44C34B421C932DC8616A8437	Gama Gt	Gama-GT	33	12 a 73
2	3487791F44C34B421C932DC8616A8437	Tgp	ALT (TGP)	51	Até 41
3	3487791F44C34B421C932DC8616A8437	Desidrogenase Lática (DHL/L)	DHL	530	240 a 480
4	3487791F44C34B421C932DC8616A8437	Proteína C Reativa, plasma	Proteína C-Reativa	1,84	Ver resultado tradicional

Figura 7. As 5 primeiras tuplas da base de exames.

'DE_EXAME' fornece a descrição do exame realizado. 'DE_ANALITO' fornece a descrição do analito, ou seja, o alvo da análise laboratorial. 'DE_RESULTADO' é o resultado do exame e 'DE_VALOR_REFERENCIA' é o valor de referência do exame.

Os atributos 'DE_RESULTADO' e 'DE_VALOR_REFERENCIA' são importantes mas podem ser substituídos por um atributo 'RESULTADO' que indica se o resultado do exame é irregular ou não. Para isso basta comparar o resultado com o valor de referência. Em algumas tuplas, os resultados ou valores de referência não foram preenchidos ou apresentam informações incoerentes, estas tuplas foram removidas. Tuplas que apresentavam valor de referência de difícil interpretação também foram removidas como a quinta tupla da figura 7, que possui como valor de referência 'Ver resultado tradicional'.

Alguns tipos de exames da base são similares, ou seja, possuem o mesmo analito. Estes exames foram renomeados com o mesmo nome com o objetivo de simplificar a base de dados reduzindo os tipos de exames diferentes. Após esse tratamento, restaram 308 tipos de exames diferentes.

Assim como feito na base de desfechos, os dados foram transformados para que uma tupla seja relativa à um paciente e não à um exame, assim como na base dos pacientes. Desse forma todos os exames realizados por um paciente e seus resultados ficam armazenados na forma de uma lista e na mesma tupla.

Com o atributo que informa a regularidade do exame, é possível também listar os exames problemáticos dos pacientes, ou seja, os exames que estão irregulares, esta lista de problemas foi inserida como atributo na base de dados. Foram inseridos também atributos numéricos para contabilizar o número total de exames feitos e o número total de problemas identificados. Com esses novos atributos e considerando o foco do trabalho, os atributos 'DE_ANALITO' e 'RESULTADO' foram removidos. A base de dados final se encontra na figura 8.

	ID_PACIENTE	exames	problemas	total_exames	total_problemas
0	3487791F44C34B421C932DC8616A8437	Fosfatase,Gama Gt,Tgp,Desidrogenase Lática (DH...	Tgp,Desidrogenase Lática (DHL/L),Fibrinogênio ...	109	38
149	0AD1FFA4419472256666A3445414F1F9	COVID	COVID	1	1
151	B8A474D172DDB5BAD11F088BDA20BE96	Interleucina 6 soro,Urina,Urina,Urina,Urina,Ur...	Interleucina 6 soro,Urina,Urina,Urina,Urina,Ur...	155	70
348	FD6F18D3E9DA3ADD5179E6B3EA21ABB2	COVID	COVID	1	1
350	3B9836D0E92704BFAB0B2E691D0A3BAE	Antígeno Influenza Teste Rápido (AGINFLU),COVI...	COVID	4	1

Figura 8. As 5 primeiras tuplas da versão final da base de exames.

3. Associação

A mineração de padrões frequentes pode levar a descoberta de associações e correlações entre itens em bases de dados relacionais. Com a grande quantidade de dados disponíveis e que continuamente são coletados, há um grande aumento de interesse por parte da maioria das industrias na mineração desse tipo de padrão em suas bases de dados. A indústria médica não é uma exceção, é um fato notório que este tipo de análise contribui para tomada de decisão médica e gerenciamento de pacientes [Calvo-Flores et al. 2001].

Existem dois aspectos principais que evidenciam a necessidade da mineração de padrões em bases de dados da indústria médica: dar suporte à atividades de resolução de problemas baseadas em conhecimentos prévios e descoberta de novo conhecimento [Calvo-Flores et al. 2001].

São buscadas regras do tipo $A \Rightarrow B$, ou seja, a doença A leva à doença B, ou, os fatos de A levam aos fatos de B. O lado esquerdo da seta se chama antecedente e o lado direito consequente. Uma definição formal de um regra de associação pode ser encontrada em [Gonçalves 2005]. Para a mineração das regras, foi utilizada uma implementação do algoritmo apriori [Agrawal and Srikant 1994]. O apriori se baseia no modelo suporte/confiança. O suporte de um conjunto de itens Z, $Sup(Z)$, representa a porcentagem de transações da base de dados que contêm os itens de Z. A confiança de uma regra, $Conf(A \Rightarrow B)$, representa, dentre as transações que contêm A, a porcentagem de transações que também contêm B.

A implementação do apriori utilizada primeiro gera todos os conjuntos que possuem suporte maior que um suporte mínimo (Sup_{min}), chamados conjuntos frequentes. Neste trabalho foi adotado $Sup_{min} = 0.03$, este valor foi escolhido de forma a tentar minerar o maior número de regras possíveis dentro de um tempo aceitável. Após a geração dos conjuntos frequentes se extraem regras de associação desses conjuntos que possuem confiança maior que uma confiança mínima $(Conf_{min})$. Foi adotada $Conf_{min} = 0.2$,

com o mesmo intuído de gerar o maior número de regras possíveis. Foram mineradas ao todo 16431672 regras de associação, envolvendo exames que tiveram resultado anormal, alta, óbito e internação.

Regras derivadas levando em consideração somente o modelo suporte/confiança, podem ser redundantes, ilusórias ou até mesmo contraditórias, e a chance de se obter esse tipo de regra aumenta quando o suporte de um determinado elemento da mesma é muito elevado. É preciso então levar em conta diferentes medidas de interesse.

A medida de interesse lift, é uma das mais utilizadas para avaliar dependências. Dada uma regra de associação $A \Rightarrow B$, esta medida indica o quanto mais frequente torna-se B quando A ocorre. Quando o lift é maior que 1, podemos dizer que há uma dependência positiva entre A e B, ou seja, A aumenta a chance de B. Quando o lift é menor que 1, podemos dizer que há uma dependência negativa entre A e B, ou seja, A diminui a chance de B. Quando lift é próximo de 1, A e B são independentes. Em uma extração de regras de associação é fundamental considerar o lift se queremos encontrar regras relevantes.

Neste trabalho busca-se por regras de associação extraídas da CDSBR, com o foco de reforçar conhecimentos de senso comum e obtenção de novos conhecimentos relacionado à COVID-19.

3.1. Regras de senso comum

Ao focar em regras com o maior lift possível, provavelmente serão encontradas relações já conhecidas pela comunidade médica, mas essa evidência adicional pode ajudar a reforçar ainda mais certas posições e decisões. São regras que indicam alta dependência positiva entre o antecedente e o consequente. Foram também buscadas regras com lift < 0.9, mas nenhuma regra foi encontrada.

Para facilitar a interpretação das regras e identificar correlações de forma mais evidente foram consideradas apenas regras com menos de 3 elementos no antecedente e consequente. As regras mineradas nessa seção são sempre relativas a exames com resultados anormais.

A busca foi iniciado com regras com lift > 16. Foram encontradas 4 regras {Fósforo, Uréia}$\Rightarrow$ {Gasometria Venosa, Gasometria Arterial}, {Gasometria Venosa, Gasometria Arterial}$\Rightarrow$ {Fósforo, Uréia}, {Lactato, Uréia}$\Rightarrow$ {Sódio, Gasometria Arterial}, {Sódio, Gasometria Arterial}$\Rightarrow$ {Lactato, Uréia}. Geralmente a gasometria é pedida quando há um problema respiratório que implique alterações na troca oxigênio-gás carbônico ou a possibilidade de um desequilíbrio acidobásico. A função renal é fator essencial no equilíbrio ácido-base. A ureia também avalia a função renal, logo problemas no exame de ureia de fato estão relacionados à problemas no exame de gasometria, indicando que o paciente tem sua função renal comprometida. O lactato indica redução na capitação ou oferta de oxigênio, estando também relacionado ao exame de gasometria.

Existem muitas outras regras com lift alto, por exemplo, existem 2748 regras com lift$>$10. Uma dessas regras é {Cálcio}$\Rightarrow$ {Fósforo}, que possui lift > 12, indicando que quem tem o cálcio alterado também tem o fósforo. Esse tipo de regra envolvendo diferentes minerais é um tipo que aparece com frequência nas regras com lift alto, indicando que de uma forma geral a deficiência em um certo tipo de mineral implica na deficiência

de outro, provavelmente associados à uma mesma causa.

3.2. Regras a partir de um alvo

3.2.1. Regras com COVID-19

Inicialmente buscavam-se regras envolvendo COVID-19, a grande dificuldade encontrada em minerar esse tipo de regra é que 99% dos pacientes possuem alteração no exame de COVID-19. O suporte alto de COVID-19 na base de dados faz com que seja fácil se extrair regras envolvendo COVID-19, mas difícil de se extrair regras que de fato são significantes.

Foram mineradas 52426 regras com COVID-19 no consequente, a média do lift dessas regras foi 0.989, o lift mínimo foi 0.961 e o máximo foi 1.007. Foram mineradas 9 regras com COVID-19 no antecedente, a média do lift dessas regras foi 0.994, o lift mínimo foi 0.991, e o máximo foi 1.01. Todas as regras mineradas possuem lift muito próximo à 1, portanto não há dependência entre o antecedente e o consequente dessas regras, ou seja, não são regras relevantes.

Como não foi possível a extração de regras relevantes envolvendo COVID-19, o foco foi mudado para regras com desfechos ou tipos de tratamento específicos.

3.2.2. Regras com foco em desfecho

Como não foi obtido sucesso na busca de regras com COVID-19 como alvo, focamos então em regras com Óbito no consequente, ou seja, buscam-se problemas encontrados em exames que levaram o paciente à óbito. Essa busca também não foi bem sucedida. O número de paciente que vieram à óbito é 44, isso corresponde à 1.06% da base de dados, sendo necessário então um suporte mínio de 0.0106 para a geração de regras envolvendo óbito. Esse valor de suporte mínimo é proibitivo pois levaria a um tempo de execução que não se tinha disponível. Em uma análise futura esse tipo de regra pode ser considerado.

O alvo foi mudado mais uma vez, o foco se tornou então regras com Internado no consequente, ou seja, buscam-se problemas encontrados em exames que levaram o paciente à internação. Esse tipo de regra é de grande interesse pois a internação reduz o número de leitos disponíveis, fator fundamental no combate a pandemia, dos pacientes que vieram a óbito, 43 deles foram internados e a internação envolve grande custo ao paciente e ao hospital. 21% dos pacientes da base de dados foram internados, sendo então possível a mineração de regras envolvendo internação, considerando o suporte mínimo de 0.03.

Foram mineradas no total 42501 regras com Internado no consequente, para filtrar as regras relevantes, considerou-se apenas regras com lift > 4.5, apenas 12 regras foram encontradas. Duas regras chamaram mais atenção, $\{$Interleucina 6 $\} \Rightarrow \{$Internado$\}$ e $\{$Interleucina 6, COVID-19 $\} \Rightarrow \{$Internado$\}$. Dados sugerem que a via da Interleucina-6 pode desempenhar um papel importante na resposta inflamatória exacerbada de pacientes com COVID-19. Em um grupo de 21 pacientes com COVID-19 tiveram a febre reduzida e 75% deles reduziram a necessidade de oxigênio suplementar poucos dias após o recebimento de outro anticorpo receptor de IL-6. Com base nesses resultados, a China atualizou

suas diretrizes de tratamento do COVID-19 e aprovou o uso desse inibidor de IL-6 para tratar pacientes com estados avançados da doença [Zhou et al. 2020].

As regras mineradas com foco na internação dão suporte as evidências encontradas por pesquisadores chineses.

4. Clusterização

A clusterização é a tarefa de identificar um conjunto finito de categorias que contêm objetos similares. É um processo de mineração não supervisionado, pois os elementos que compõem a base de entrada não tem seu grupo (ou cluster) previamente definido.

Algoritmos de clusterização organizam a base de dados em k clusters de tuplas semelhantes. Os clusters devem conter tuplas semelhantes entre si mas também dissimilares em relação à tuplas de outros clusters. Neste trabalho serão aplicados os algoritmos k-Means e DBSCAN à base CDSBR. Os algoritmos não serão explicados em detalhes, para mais informações sobre esses algoritmos ver [Han et al. 2011].

No contexto de combate a pandemia, vem sendo utilizadas análises envolvendo clusterização. Em Maharashtra na Índia, pesquisadores aplicaram técnicas de clusterização hierárquicas para identificar distritos mais afetados, e assim apoiar o governo e instituições médicas na criação de políticas e tratamentos com focos específicos [Kumar 2020].

Identificar grupos de interesse pode ser uma forma de direcionar os esforços no combate à COVID-19.

4.1. Preparação da base de dados

Para realizar a tarefa de clusterização primeiramente deve-se remover atributos que podem ser usados como rótulos. Foram então removidos da base de dados os atributos 'de_desfecho', 'de_tipo_atendimento', 'ID_PACIENTE'.

A clusterização é um algoritmo que trabalha com atributos numéricos, atributos categóricos devem ser transformados. Como a lista de exames e de problemas de cada paciente são atributos categóricos e já foram explorados na análise de regras de associação, foram removidos da base de dados. Seria possível criar atributos binários (0 ou 1) para representar se um paciente tem ou não uma determinada doença, essa abordagem tornaria a base com mais de 300 atributos diferentes e por isso também não foi considerada.

Em testes preliminares ficou evidente que o atributo referente ao sexo do paciente estava interferindo na clusterização, fazendo com que os clusters se dividissem em pacientes do sexo masculino e feminino. Considerou-se mais interessante também excluir esse atributo da análise.

Entre os 44 pacientes que vieram à óbito, 36% não tinha o atributo idade preenchido. Apesar de serem um grupo importante a se considerar, os pacientes que não apresentaram idades preenchidas foram removidos da base de dados. Os primeiros elementos da base de dados após a seleção de atributos podem ser vistos na figura 9.

Como os atributos se encontram em escalas diferentes, é necessário antes da aplicação dos algoritmos normalizar a base de dados. A normalização impede que atributos diferentes contribuam de formas de diferentes no cálculo da distância. A normalização foi feita através da estratégia min-max.

	n_atendimentos	t_atendimento	idade	total_exames	total_problemas
0	2	30	84.0	1123.0	40
1	2	13	78.0	318.0	21
2	2	20	87.0	552.0	35
3	15	171	79.0	2130.0	40
4	1	27	81.0	2481.0	37

Figura 9. As 5 primeiras tuplas da base de dados para clusterização após a seleção de atributos.

4.2. k-Means

O k-Means é um algoritmo iterativo que tenta encontrar k clusters não vazios tal que a soma das distâncias quadráticas entre as tuplas e os centroides de seus respectivos clusters seja minimizada. O algoritmo k-Means tem como parâmetro de entrada o k, que indica o número de clusters que devem ser formados.

Para identificar o número de clusters ótimo, é preciso identificar o k tal que minimize a distância entre os elementos de um cluster e seu centroide como também maximizar o coeficiente de silhueta. Um estudo preliminar para identificar o número de clusters adequado foi feito e se encontra na figura 10.

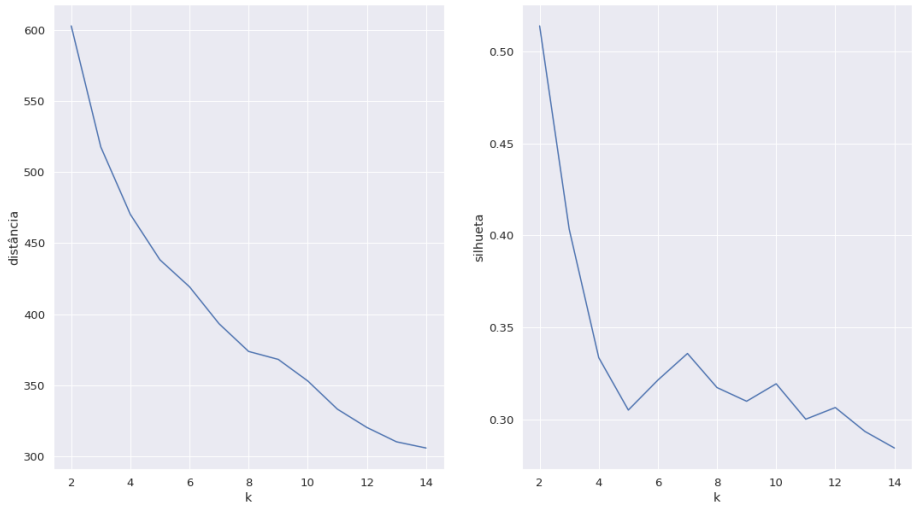

Figura 10. Soma das distâncias entre os centroides dos clusters e seus elementos (esquerda) e coeficiente de silhueta da clusterização (direita) em função do número de clusters.

A soma das distâncias entre os centroides dos clusters e seus elementos diminuí progressivamente com o número de clusters, o que é desejável, assim como coeficiente de silhueta da clusterização, o que é indesejável. É preciso então escolher o k de forma a balancear esses dois fatores, foram escolhidos $k = 2$ e $k = 3$, evitando um coeficiente de silhueta muito baixo. Outro motivo dessa escolha de k é interpretabilidade dos resultados.

Os centroides de ambas as clusterizações escolhidas, seus desvios padrões e o número de elementos de cada cluster se encontram nas tabelas 1 e 2. Onde n_atend é número de atendimentos feitos, t_atend é a duração total dos atendimentos em dias, total_exames é número total de exames feitos, total_prob é o número total de exames anormais e n é o número de elementos do cluster.

Tabela 1. Resultados do k-Means para $k = 2$.

cluster	n_atend	t_atend	idade	total_exames	total_prob	n
0	3.84 ± 3.60	7.39 ± 12.9	37.3 ± 9.64	26.7 ± 52.5	2.94 ± 3.26	3146
1	4.23 ± 6.51	17.4 ± 24.7	65.3 ± 10.7	381 ± 576	15.2 ± 11.3	869

Tabela 2. Resultados do k-Means para $k = 3$.

cluster	n_atend	t_atend	idade	total_exames	total_prob	n
0	3.95 ± 3.68	7.83 ± 13.8	33.4 ± 7.18	24.2 ± 48.8	2.78 ± 3.15	2419
1	3.38 ± 3.44	6.90 ± 9.15	55.4 ± 8.94	70.6 ± 92.5	5.55 ± 4.93	1266
2	5.81 ± 9.32	32.5 ± 33.4	69.6 ± 12.4	810 ± 748	26.5 ± 9.62	330

Analisando os centroides, parece que o k-Means separou os clusters com foco maior na idade, é o atributo com menor desvio padrão relativo. Separar os pacientes por idades é uma boa estratégia, note que os pacientes pertencentes à grupos com médias de idade maiores fazem mais exames, tem mais problemas nos exames, são atendidos mais vezes e por um tempo mais longo. Analisando o desvio padrão é notório que mesmo entre os elementos do mesmo cluster há uma grande diferença em todos os atributos menos o atributo idade.

Um fato curioso é que todos os pacientes que vieram à óbito pertencem ao cluster 2, na clusterização com $k = 3$ e ao cluster 1, na clusterização com $k = 2$.

4.3. DBSCAN

O DBSCAN é um algoritmo baseado em densidade, e possui dois parâmetros: raio (r) e densidade (d). O raio é a distância máxima entre um ponto e outro para que estes estejam contidos no mesmo cluster, um cluster só será formado se possuir no mínimo d elementos. É um algoritmo que não tem como parâmetro o número de clusters, bom para base com ruídos, e adequado para identificar clusters de formatos arbitrários. Na prática, a grande desvantagem do método DBSCAN está na calibração de seus parâmetros, que pode ser uma tarefa complexa, principalmente quando aplicado à base de dados com um grande número de atributos, onde a maldição da dimensionalidade se torna um fator importante.

Foram considerados apenas clusters com mais de 28 tuplas, ou seja, $d = 28$. Essa escolha se deve ao fato do grupo de pessoas que vieram à óbito é composto por 28 pacientes, assim havendo a possibilidade do algoritmo de encontrar esse grupo específico. Como a base de dados contem no total 4015 tuplas, grupos com menos de 28 pacientes representam menos de 0.7% da base, sendo pouco representativos.

Uma análise preliminar foi feita para identificar bons valores de r. O algoritmo DBSCAN foi executado variando o parâmetro r entre 1.0 e 0.001. Para esses valores foram identificados os números de clusters obtidos e o número de pontos considerados como ruído. Esses resultados se encontram na figura 11.

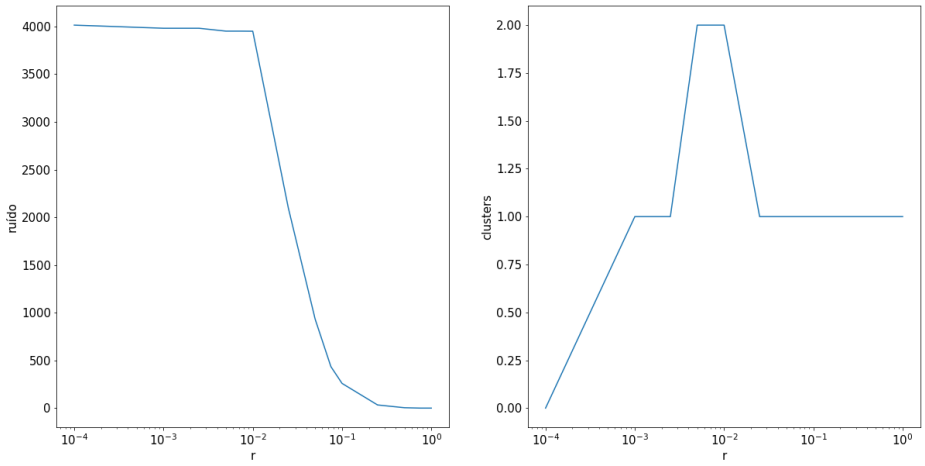

Figura 11. Número de pontos considerados como ruído (esquerda) e número de clusters identificados (direita) em função do parâmetro r, $d = 28$.

Repare que para r maiores valores de r todos os pontos são considerados como parte de um único cluster e não existe ruído. A diminuição do valor de r aumenta o número de pontos considerados como ruído, até eventualmente todos os pontos serem considerados como ruído. Apenas para valores de r entre 0.01 e 0.005 foi possível a obtenção de mais de um cluster, sendo assim foram analisadas somente essas clusterizações. Os resultados dessas clusterizações se encontram nas tabelas 3 e 4.

Tabela 3. Resultados do DBSCAN para $r = 0.01$ e $d = 28$.

cluster	n_atend	t_atend	idade	total_exames	total_prob	n
0	1.00 ± 0.00	1.13 ± 0.54	37.0 ± 0.00	1.33 ± 0.63	1.00 ± 0.00	36
1	1.00 ± 0.00	1.03 ± 0.18	38.0 ± 0.00	1.39 ± 0.49	1.00 ± 0.00	28

Tabela 4. Resultados do DBSCAN para $r = 0.005$ e $d = 28$.

cluster	n_atend	t_atend	idade	total_exames	total_prob	n
0	1.00 ± 0.00	1.05 ± 0.23	37.0 ± 0.00	1.39 ± 0.49	1.00 ± 0.00	36
1	1.00 ± 0.00	1.03 ± 0.18	38.0 ± 0.00	1.39 ± 0.49	1.00 ± 0.00	28

Para a clusterização com r = 0.01 foram obtidos 3951 tuplas consideradas como ruído, enquanto que para r = 0.005 foram consideradas 3952. Os elementos dos clusters encontrados possuem baixo desvio padrão mas elementos pertencentes à diferentes clusters são muito próximos entre si, podendo provavelmente fazer parte de um mesmo grupo. Além disso, os clusters encontrados são de grupos de pacientes muito específicos e que não revelam nenhuma informação importante.

Uma calibração de parâmetros mais atenciosa poderia levar à melhores resultados da clusterização. Alguns outros valores de d foram testados mas todos sem sucessos na obtenção de clusters mais representativos, os gráficos para $d = 50$ e $d = 100$ estão nas

figuras 12 e 13. Essa busca por parâmetros é uma das grandes desvantagens do DBSCAN e essa análise ilustra bem isso.

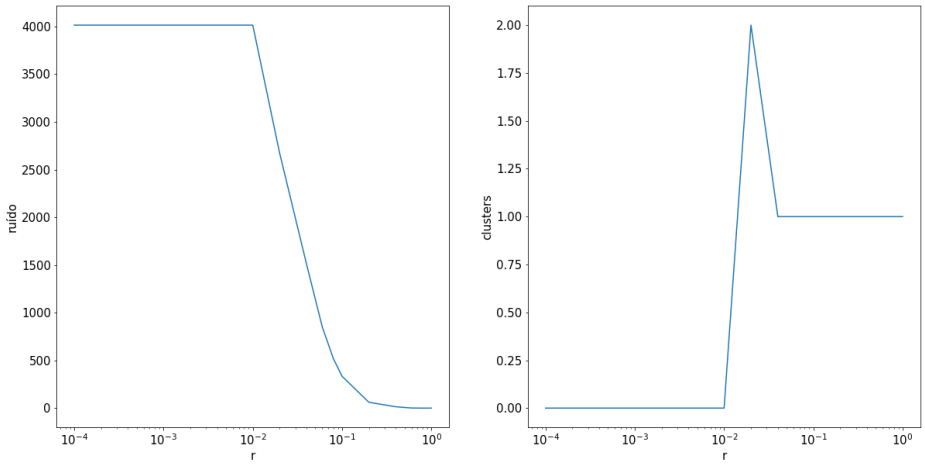

Figura 12. Número de pontos considerados como ruído (esquerda) e número de clusters identificados (direita) em função do parâmetro r, $d = 50$.

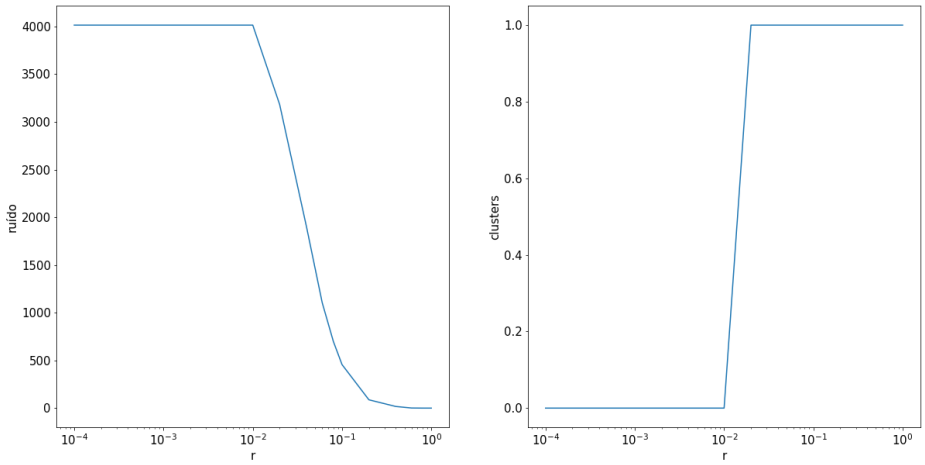

Figura 13. Número de pontos considerados como ruído (esquerda) e número de clusters identificados (direita) em função do parâmetro r, $d = 100$.

5. Classificação

A classificação é uma forma de mineração de dados na qual são extraídos modelos a partir de bases de dados. Esse modelos são chamados classificadores e são capazes de prever a classe (rótulo) de uma instância que tenha os mesmos atributos da base em que os modelos foram extraídos.

Neste trabalho o objetivo é tentar prever o tempo de atendimento de um paciente através de um modelo de classificação extraído da base de dados CDSBR. O tempo

de atendimento é um atributo numérico e contínuo, podendo ser previsto utilizando um processo de regressão. No entanto, é possível transformar esse atributo em categórico, pacientes que tiveram tempo de atendimento menor do que 25% dos pacientes da base recebem rótulo 0, pacientes entre 25% e 50% recebem rótulo 1, pacientes entre 50% e 75% recebem rótulo 2 e por último, pacientes que possuem tempo de atendimento maior que 75% da base recebem rótulo 3. Com isso transformamos um problema típico de regressão em um problema de classificação com 4 possíveis rótulos.

5.1. Preparação da base de dados

O primeiro passo é rotular a base adotando a estratégia explicada na seção anterior. O atributo 'ID_PACIENTE' foi removido pois não acrescenta informação ao problema.

Como a lista de exames e de problemas de cada paciente foram removidos da base de dados pelo mesmo argumento utilizado na seção sobre clusterização.

Inicialmente considerou-se utilizar o atributo de_desfecho como rótulo. Esse atributo apesar de interessante não foi escolhido pois o número de óbitos era muito menor que o número de altas, correspondendo apenas à menos de 1% da base de dados. Esse atributo foi descartado da análise.

Como a ferramenta utilizada no trabalho foi o scikit learn, foi necessário transformar todo os atributos categóricos em numéricos. A figura 14 ilustra a base de dados final, pronta para o processo de classificação. Após isso a base foi separada em atributos independentes (X) e rótulo (y).

	de_tipo_atendimento	n_atendimentos	t_atendimento	IC_SEXO	idade	total_exames	total_problemas
0	3	2	3	1	84.0	1123.0	40
1	3	2	3	1	78.0	318.0	21
2	3	2	3	1	87.0	552.0	35
3	3	15	3	1	79.0	2130.0	40
4	3	1	3	1	81.0	2481.0	37

Figura 14. As 5 primeiras tuplas da base de dados para classificação após a seleção de atributos.

Todo processo de classificação necessita de 2 passos para serem executados: treinamento que é o momento em que os algoritmos possuem seus parâmetros calibrados e geram modelos quando necessário, e teste que é a etapa onde é analisado o desempenho do algoritmo na base de dados.

Para que seja possível a execução desses 2 passos, se faz necessário realizar uma divisão na base de dados original. Essa divisão faz com que parte dos elementos da base sejam utilizados na etapa de treinamento, chamamos essa parte de base de treinamento, e a outra parte seja utilizada na etapa de teste, essa segunda é chamada de base de teste.

Uma estratégia bastante utilizada na separação da base de dados é a validação cruzada estratificada. Essa estratégia busca dividir a base em K partições, após isso são realizadas K iterações onde em cada iteração uma partição é utilizada para a base de teste e as outras são utilizadas para a base de treino. A divisão estratificada, indica que cada uma das partições geradas, buscará manter a proporção de elementos de cada classe

encontrada na base original. Uma exemplificação desse processo se encontra na figura 15.

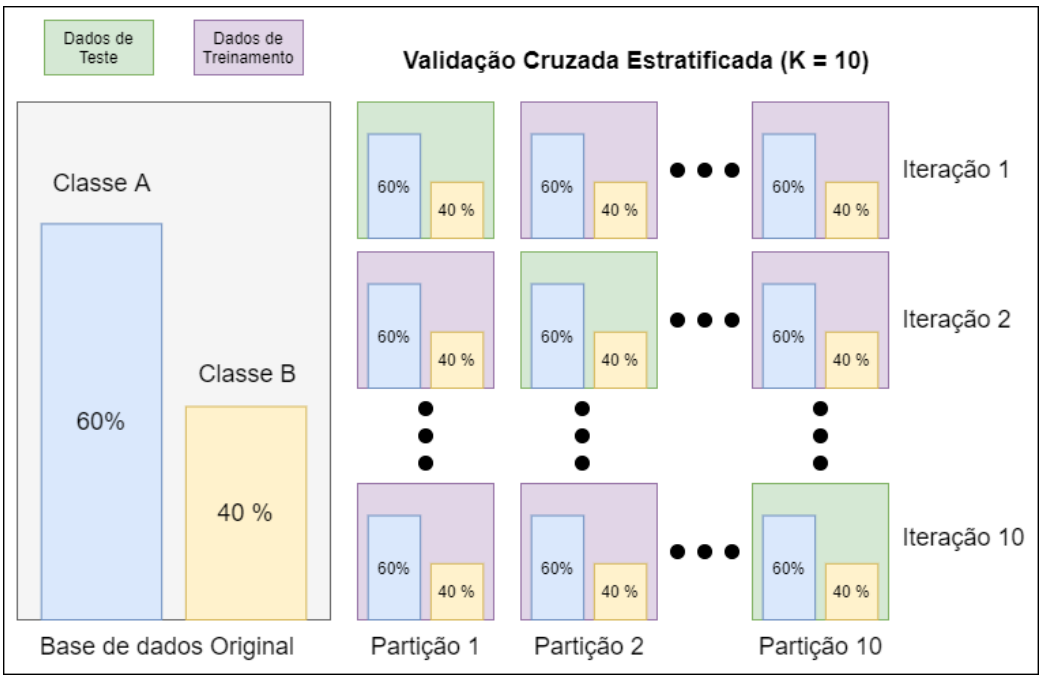

Figura 15. Exemplificação do processo de validação cruzada estratificada para o K = 10.

Neste trabalho 70% dos dados foram separados para a base de treinamento e 30% para a base de teste. Para calibrar os parâmetros do classificador antes de treinar e classificar a base de testes, foi utilizada uma validação cruzada estratificada na base de treino variando os parâmetros em busca do melhor modelo para a classificação.

Para realizar o processo de classificação foram utilizados os classificadores Árvore de decisão, Naive Bayes e kNN. As próximas seções do trabalho possuem um breve comentário sobre cada um deles, esses algoritmos não serão explicados em detalhes neste trabalho, para mais informações sobre eles ver [Han et al. 2011].

5.2. Métrica de avaliação dos classificadores

Outra característica importante para o processo de classificação é a análise dos resultados dos classificadores. Dentre as várias medidas existentes, neste trabalhos foram utilizadas 2 métricas muito recomendadas nesse tipo de mineração, que são: matriz de confusão e acurácia.

A matriz de confusão é representada através de uma tabela que fornece informações que indicam como o um classificador performou, ao classificar cada classe existente na base de dados. Cada linha dessa tabela representa a classe real da instância, as colunas representam a classe prevista pelo classificador. Cada célula M_{ij} contêm o número de instâncias da classe C_i que foram classificadas como C_j. A figura 16 apresenta um exemplo de matriz de confusão para problemas multiclasse.

A acurácia tem como objetivo indicar o quão eficiente o classificador está sendo em conseguir classificar uma nova instância. Ela representa a porcentagem de acerto do

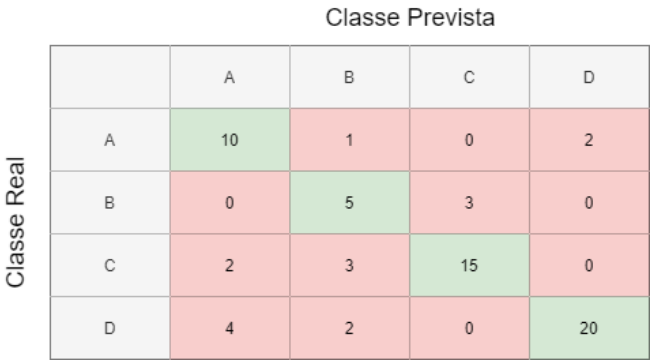

Figura 16. Exemplo de matriz de confusão para problemas multiclasse.

classificador e pode ser calculada através da seguinte equação:

$$A = \frac{N^o Acertos}{N^o Instâncias\ da\ Base\ de\ Teste}. \tag{1}$$

5.3. Árvore de decisão

A Árvore de decisão é um algoritmo que busca utilizar um conjunto de atributos de uma instância, para predizer o valor da sua classe (rótulo). Na etapa de treino, a árvore de decisão busca identificar e mapear os atributos que melhor separam as classes. Esse mapeamento é armazenado em uma estrutura de dados semelhante a uma árvore. Onde cada folha representa uma classe, os nós intermediários representam atributos que foram considerados na divisão e as arestas possuem os valores desses atributos. A etapa de teste consiste em verificar os atributos da instância de teste e percorrer a árvore da raiz até chegar a uma folha para classificar a tupla. Um exemplo de modelo de árvore de decisão se encontra na figura 17.

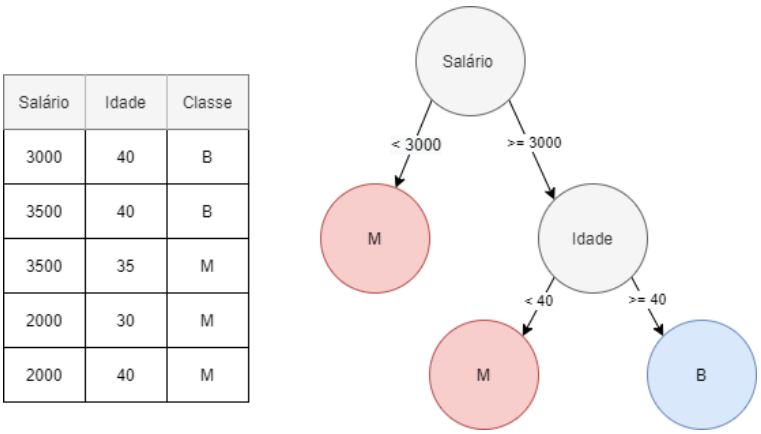

Figura 17. Exemplo de árvore de decisão.

Na figura 18, é apresentado um gráfico com o resultado do processo de estimativa de parâmetros utilizando a validação cruzada na base de teste. O gráfico a esquerda, apresenta a variação da acurácia em relação ao parâmetro max_depth. O da direita, representa o tempo de execução do algoritmo, também relacionado ao parâmetro max_depth.

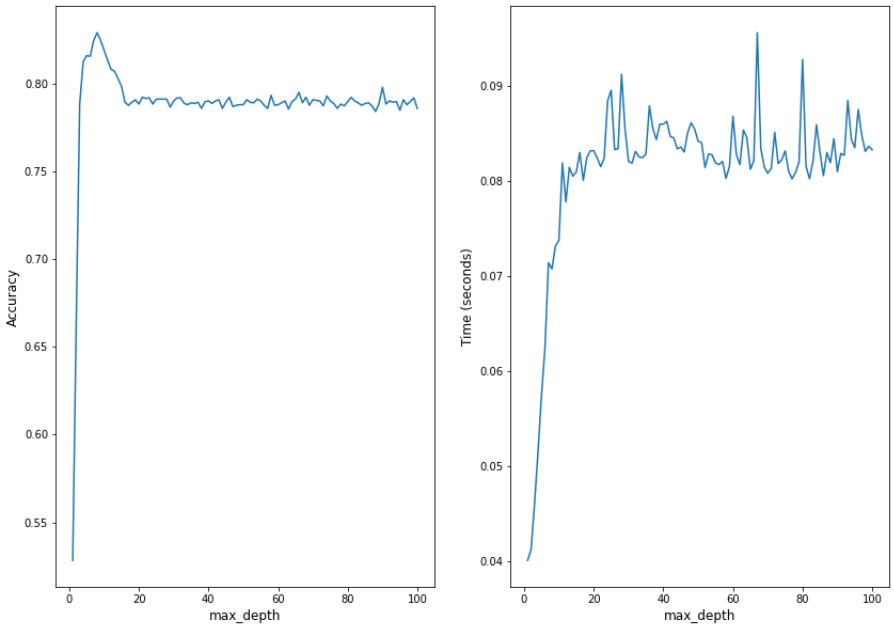

Figura 18. Gráfico de acurácia e tempo da árvore de decisão utilizando validação cruzada na base de teste.

Após esse processo foi verificado que a configuração que obteve o melhor resultado em relação a acurácia foi quando $\text{max\_depth} = 8$. Com essa configuração o classificador obteve uma acurácia de 0.8306, com 0.0171 de desvio padrão.

Esse parâmetro foi utilizado para treinar um modelo com toda a base de treino e classificar a base de teste. Através desse teste, foi gerado a matriz de confusão apresentada na figura 19. Através da matriz é possível analisar que o classificador, comete menos erros ao estimar que um paciente irá demandar menos tempo de atendimento em relação aos erros cometido no sentido oposto. Isso é uma característica importante para o problema abordado, pois é melhor que o tempo de atendimento seja superestimado do que subestimado.

	até 2 dias	3 ou 4 dias	5 a 10 dias	acima de 10 dias
até 2 dias	389	8	4	7
3 ou 4 dias	27	165	19	8
5 a 10 dias	3	31	221	37
acima de 10 dias	5	9	69	207

Figura 19. Matriz de confusão da árvore de decisão.

A acurácia obtida através desse teste foi de 0.8122.

5.4. Naïve Bayes

O classificador Naïve Bayes (NB) é um classificador estatístico. Como todo classificador Bayesiano, é baseado no teorema de Bayes [Han et al. 2011]. NB assumi que os valores de um atributo da base de dados são independentes dos valores de outros atributos da base. Essa hipótese é feita para simplificar os cálculos necessários, por isso o algoritmo é considerado Naïve (ingênuo).

A ideia principal é usar o teorema de Bayes para calcular a probabilidade ($P(y_i|X)$) de uma tupla X da base de dados pertencer a cada uma das classes y_i. O classificador decide que X é da classe y_i se $P(y_i|X) > P(y_j|X)$ para todo j diferente de i, ou seja, escolhe a classe com a maior probabilidade de X pertencer. Calcular essas probabilidades de maneira exata não é uma tarefa simples, o NB não calcula as probabilidades de forma exata, mas de forma precisa o suficiente para ser capaz de rankear as probabilidades e selecionar a maior.

Estudos apontam que classificadores do tipo NB se comparam com árvore de decisão em termos de performance e algumas arquiteturas de redes neurais, apresentando também alta acurácia e baixo tempo de processamento quando aplicados à grandes bases de dados [Han et al. 2011]. Sendo assim é um ótimo candidato para ser aplicado à base de dados CDSBR, pois é comparável à arvore de decisão e ao mesmo tempo é de um paradigma diferente.

Na figura 20, é apresentado um gráfico com o resultado do processo de estimativa de parâmetros utilizando a validação cruzada na base de teste. O gráfico a esquerda, apresenta a variação da acurácia em relação ao parâmetro $var\_smoothing$. O da direita, representa o tempo de execução do algoritmo, também relacionado ao parâmetro $var\_smoothing$.

Após o processo constatou-se que a configuração que obteve o melhor resultado em relação a acurácia foi quando var_smoothing $= 0.0000006876$. Com essa configuração o classificador obteve uma acurácia de 0.7868, com 0.0263 de desvio padrão.

Esse parâmetro foi utilizado para treinar um modelo com toda a base de treino e classificar a base de teste. Através desse teste, foi gerado a matriz de confusão apresentada na figura 21. Os erros cometidos pelo NB, podem ser interpretados da mesma forma que os da Árvore de Decisão.

	até 2 dias	3 ou 4 dias	5 a 10 dias	acima de 10 dias
até 2 dias	396	0	8	4
3 ou 4 dias	32	155	29	3
5 a 10 dias	8	45	220	19
acima de 10 dias	18	15	91	166

Figura 21. Matriz de confusão do Naive Bayes.

A acurácia obtida através desse teste foi de 0.7750.

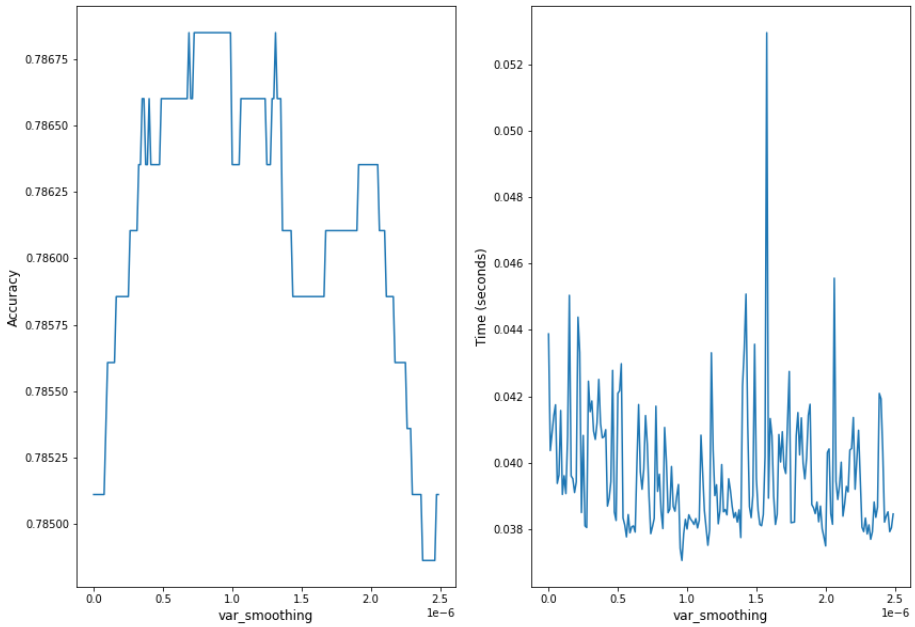

Figura 20. Gráfico de acurácia e tempo do Naive Bayes utilizando validação cruzada na base de teste.

5.5. kNN

Os métodos discutidos anteriormente são considerados classificadores do tipo eager. Classificadores eager geram um modelo a partir da base de dados antes de classificar novas tuplas.

O kNN é um método lazy, ou seja, ele só executa uma generalização com base nos dados de treino quando recebe um nova tupla para avaliar. Dessa forma o kNN realiza pouco ou nenhum trabalho na fase de treino, a fase mais custosa é a de predição. Classificar muitas tuplas pode ser uma tarefa computacionalmente cara [Han et al. 2011].

É um algoritmo de um paradigma diferente dos anteriores, se baseando no aprendizado por analogia. O kNN classifica um tupla X considerando uma votação simples entre os k vizinhos mais próximos de X que pertencem à base de treino. A noção de proximidade se dá utilizando uma métrica de distância, sendo a principal métrica utilizada a distância Euclidiana.

Na figura 22, é apresentado um gráfico com o resultado do processo de estimativa de parâmetros utilizando a validação cruzada na base de teste. O gráfico a esquerda, apresenta a variação da acurácia em relação ao parâmetro k. O da direita, representa o tempo de execução do algoritmo também relacionado ao parâmetro k.

Após o processo verificou-se que a configuração que obteve o melhor resultado em relação a acurácia foi quando k $= 9$. Com essa configuração o classificador obteve uma acurácia de 0.7114, com 0.0307 de desvio padrão.

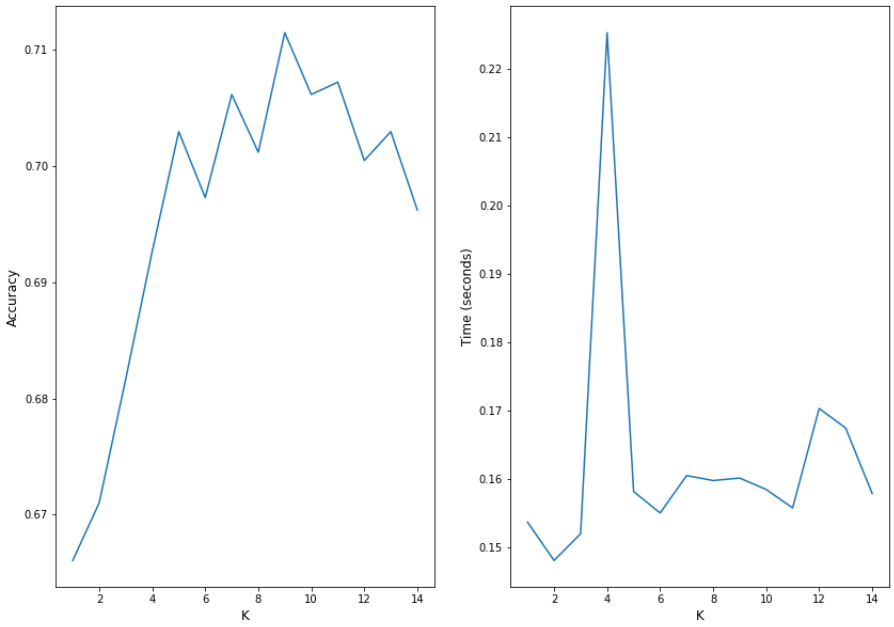

Figura 22. Gráfico de acurácia e tempo do kNN utilizando validação cruzada na base de teste.

Esse parâmetro foi utilizado para classificar a base de teste. Através desse teste, foi gerado a matriz de confusão apresentada na figura 23. Através da matriz foi analisado que, assim como o Naive Bayes, os erros cometidos pelo classificador kNN, podem ser interpretados da mesma forma que os da Árvore de Decisão.

	até 2 dias	3 ou 4 dias	5 a 10 dias	acima de 10 dias
até 2 dias	369	33	2	4
3 ou 4 dias	74	117	25	3
5 a 10 dias	24	65	171	32
acima de 10 dias	20	14	74	182

Figura 23. Matriz de confusão do kNN.

A acurácia obtida através desse teste foi de 0.6939.

5.6. AutoLearn

Algoritmos de classificação possuem performance diferente se aplicados à diferentes bases de dados. Não exite uma metodologia que seja melhor que as outras em todos os casos. Sendo assim o processo de seleção de classificadores é uma parte importante de toda tarefa de classificação.

A ferramenta scikit learn conta com uma função de auto learning [Feurer et al. 2015]. Essa aplicação faz de maneira automática os processos de seleção de classificadores e calibração de parâmetros, buscando encontrar os melhores modelos de classificação para a base de dados. O desempenho desse processo, depende do tempo configurado para encerra-lo, é esperado que quanto mais tempo este processo for executado, melhor seja o resultado obtido por ele. Ao final do processo de auto learning, é gerado um Ensemble contendo os classificadores que obtiveram os melhores resultados.

Ao executar o processo de auto learning por 2 horas na base de dados, foi observado que ele analisou um total de 408 configurações de classificadores. A figura 24 apresenta os 16 classificadores que foram selecionados para serem utilizados no Ensemble que foi gerado ao final do processo. A coluna acurácia dessa mesma figura, apresenta a acurácia de cada classificador no processo de avaliação interna do auto learning.

	Acurácia
gradient_boosting	0.8549
gradient_boosting	0.8539
gradient_boosting	0.8528
gradient_boosting	0.8496
random_forest	0.8474
random_forest	0.8464
extra_trees	0.8453
gradient_boosting	0.8442
gradient_boosting	0.8431
extra_trees	0.8421
random_forest	0.8421
gradient_boosting	0.841
random_forest	0.841
random_forest	0.841
gradient_boosting	0.841
random_forest	0.8399

Figura 24. Classificadores utilizados no Ensemble gerado através do auto learn do scikit.

Após este processo, o Ensemble gerado foi treinado com toda a base de treinamento e então avaliado através da base de teste.A matriz de confusão apresentada na figura 25, foi obtida através dos dados desse teste. Analisando a matriz foi verificado que, assim como os outros classificadores, os erros cometidos pelo Ensemble podem ser interpretados da mesma forma que os da Árvore de Decisão.

	até 2 dias	3 ou 4 dias	5 a 10 dias	acima de 10 dias
até 2 dias	399	3	3	3
3 ou 4 dias	28	166	18	7
5 a 10 dias	4	30	223	35
acima de 10 dias	15	10	42	223

Figura 25. Matriz de confusão do Ensemble gerado através do auto learn.

A acurácia obtida através desse teste foi de 0.8362.

6. Conclusões

A pandemia de COVID-19 é um fenômeno quase sem precedentes na história moderna e deve ser combatido utilizando o maior número de metodologias possíveis. Apesar da pandemia ter se iniciado em dezembro de 2019, parece longe de seu fim e há ainda muito a se entender e descobrir sobre ela, como vacinas, tratamentos eficientes e políticas públicas adequadas.

Abordagens baseada em mineração de dados se mostram eficazes no processo de obtenção de conhecimento, podendo então servir como base para a implementação de possíveis soluções para o combate à pandemia.

Utilizando a base de dados CDSBR, uma base de dados com informações sobre exames e tratamentos realizados por pacientes que em sua maioria possuíam COVID-19, foi possível aplicar diferentes metodologias de mineração com objetivo principal de gerar modelos preditivos, regras de associação relevantes e identificar grupos de interesse.

Através de transformações dos dados da base, foi possível a extração de regras de associação de senso comum, validando a análise e reforçando os conhecimentos já adquiridos na área médica. A mineração de regras relevantes envolvendo COVID-19 não foi possível, como 99% dos pacientes analisados estavam infectados, só foram extraídas regras com lift próximo à 1. Quando o foco passou para regras com internação como consequente, foi possível identificar as regras $\{\text{Interleucina 6, COVID-19}\} \Rightarrow \{\text{Internado}\}$ e $\{\text{Interleucina 6}\} \Rightarrow \{\text{Internado}\}$, com lift > 4.5. Essas regras revelam uma correlação importante entre a Interleucina 6 e a piora no quadro de COVID-19, dando suporte maior ao tratamento com inibidores de Interleucina 6.

Os algoritmos k-Means e DBSCAN foram aplicados à base de dados para a extração de possíveis grupos de interesse. Identificar grupos é uma forma de otimizar o atendimento e tratamento de pacientes, direcionando estratégias específicas para grupos específicos. Os resultados apresentados pelo k-Means revelam que o algoritmo separou os pacientes em grupos baseados em suas idades. A separação por idade é uma estratégia inteligente, principalmente no contexto da COVID-19 onde se sabe que o vírus tem maior letalidade em pessoas com idade mais avançada. O DBSCAN não conseguiu identificar grupos relevantes. Na maioria dos casos conseguiu identificar um único grupo ou grupos pequenos e pouco representativos, considerando como ruído a maior parte das tuplas. Uma calibração mais minuciosa de seus parâmetros poderia levar à resultados melhores, no entanto, esse tipo de calibração é custosa.

Através da técnica de classificação, foi possível criar modelos preditivos que apre-

sentaram um bom desempenho preditivo. Com base nos resultados, foi observado que os algoritmos de auto learn e a árvore de decisão geraram os modelos mais eficiente para o problema abordado. As análises realizadas através das informações existente na matriz de confusão, reforçam que a maior parte dos erros dos classificadores gerados, está relacionado a superestimar o tempo de atendimento ao invés de subestimar, essa característica de erro é a ideal para a proposta de classificação realizada neste trabalho. Desse modo, os modelos se mostraram eficientes e poderiam ser utilizados para ajudar a prever o tempo de atendimento que um novo paciente irá demandar.

Referências

Agrawal, R. and Srikant, R. (1994). Fast algorithms for mining association rules. In *In Proc. 1994 Int. Conf. Very Large Data Bases (VLDB'94)*, pages 487–499.

Calvo-Flores, M., Sánchez, D., Martin-Bautista, M., and Vila, M. (2001). Mining association rules with improved semantics in medical databases. *Artificial intelligence in medicine*, 21:241–5.

Dana, S., Simas, A. B., Bruno A Filardi, R. N. R., da Costa Valiengo, L. L., and Neto, J. G. (2020). Brazilian modeling of covid-19 (bram-cod): a bayesian monte carlo approach for covid-19 spread in a limited data set context. *medRXiv*.

FAPESP (2020). FAPESP COVID-19 Data Sharing/BR. `https://repositoriodatasharingfapesp.uspdigital.usp.br`.

Feurer, M., Klein, A., Eggensperger, K., Springenberg, J., Blum, M., and Hutter, F. (2015). Efficient and robust automated machine learning. In Cortes, C., Lawrence, N. D., Lee, D. D., Sugiyama, M., and Garnett, R., editors, *Advances in Neural Information Processing Systems 28*, pages 2962–2970. Curran Associates, Inc.

Gonçalves, E. (2005). Regras de associação e suas medidas de interesse objetivas e subjetivas. *INFOCOMP*.

Han, J., Kamber, M., and Pei, J. (2011). *Data Mining Concepts and Techniques*. The Morgan Kaufmann Series in Data Management Systems, 3rd edition.

Kumar, S. (2020). Use of cluster analysis to monitor novel coronavirus-19 infections in maharashtra, india. In *Indian Journal of Medical Sciences*, pages 44–48.

Mello, L. E., Suman, A., Medeiros, C. B., Prado, C. A., Rizzatti, E. G., Nunes, F. L. S., Barnabé, G. F., Ferreira, J. E., Sá, J., Reis, L. F. L., Rizzo, L. V., Sarno, L., de Lamonica, R., Maciel, R. M. d. B., Cesar-Jr, R. M., and Carvalho, R. (2020). Opening Brazilian COVID-19 patient data to support world research on pandemics.

Sanar (2020). Sanar — linha do tempo do coronavírus no brasil. Disponível em: `https://www.sanarmed.com/linha-do-tempo-do-coronavirus-no-brasil`. Acesso em: 20 novembro 2020.

Yang, Z. e. a. (2020). Modified seir and ai prediction of the epidemics trend of covid-19 in china under public health interventions. *Journal of thoracic disease vol. 12,3 (2020)*, pages 165–174.

Zhou, F., Yu, T., Du, R., Fan, G., Liu, Y., Liu, Z., Xiang, J., Wang, Y., Song, B., Gu, X., Guan, L., Wei, Y., Wu, H. L. X., Xu, J., Tu, S., Zhang, Y., Chen, H., and Cao, B.

(2020). Clinical course and risk factors for mortality of adult inpatients with covid-19 in wuhan, china: a retrospective cohort study. In *Lancet*, pages 2962–2970. doi: https://doi.org/10.1016/S0140-6736(20)30566-3.