Aprendizado de máquina para análise da frequência cardíaca e detecção de epilepsia

Ana Paula da Rocha, Alexandre Soli Soares, Vinicius Cin

(Grupo: 2)

Disciplina: Introdução a Informática Médica

Professor: Cesar Ramos Rodrigues

Introdução

A epilepsia é uma alteração temporária e reversível do funcionamento do cérebro, que não tenha sido causada por febre, drogas ou distúrbios metabólicos e se expressa por crises epilépticas repetidas [3]. Mesmo sendo caracterizada como um distúrbio no cérebro, foi descoberto pelos pesquisadores da Case Western University a partir dos resultados do estudo que há uma variabilidade da frequência cardíaca na epilepsia e aumento da atividade do sistema nervoso parassimpático durante o sono [2].

Dado esse cenário, um modelo de aprendizado de máquina que consiga detectar, a partir de dados de frequência cardíaca, quando está ocorrendo uma epilepsia pode ser relevante para o sistema de saúde como uma ferramenta que auxilie os médicos. Assim, é possível intervir o mais rápido possível quando alguma alteração seja detectada em um paciente resultando em um melhor atendimento.

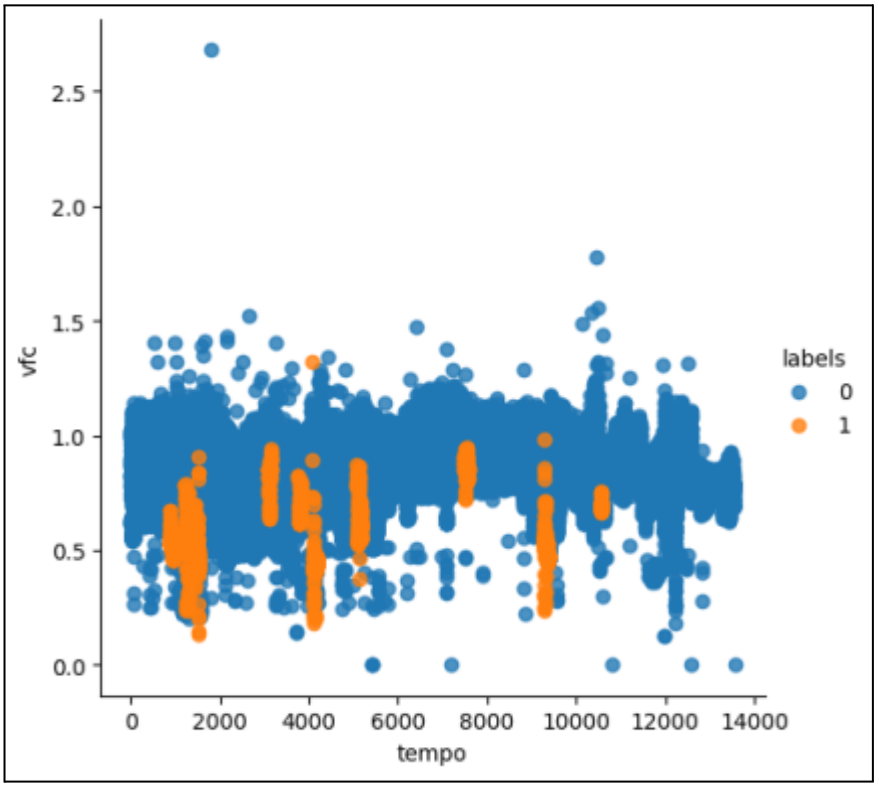
O desenvolvimento do trabalho foi baseado no processo de mineração de dados, que consiste em entender o problema, realizar a análise exploratória do conjunto e realizar transformações nos dados como tratamento de dados nulos. Para então, aplicar os modelos de aprendizado de máquina e validar os resultados obtidos através das métricas de classificação.

Neste trabalho serão aplicados três modelos de aprendizado de máquina para problemas de classificação, visto que o objetivo é identificar se o paciente está ou não com epilepsia a partir de dados já classificados. Escolheu-se comparar o modelo de regressão logística com os modelos de *Random Forest* e *KNeighbors*, sendo utilizadas as métricas de precisão, *Recall*, Score F1 e a matriz de confusão para avaliação dos desempenhos.

Análise Exploratória

O conjunto de dados é formado por 7 arquivos contendo informações sobre oscilações da frequência cardíaca pós-ictal em um grupo heterogêneo de pacientes com e sem epilepsia parcial. Após o processo de extração dos dados há um total de 73919 amostras, sendo 72507 normais e 1412 com epilepsia como mostrado na Figura 1. O fato das classes serem desbalanceadas deve ser considerado posteriormente na análise das métricas dos modelos.

Figura 1: Quantidade de amostras normais e com epilepsia no conjunto de dados.



Tratamentos dos Dados

Classificação das amostras

De forma a classificar cada amostra como sendo normal (0) ou convulsão (1), o arquivo times.seize foi utilizado. Este arquivo identifica os horários de início e término de uma convulsão nas amostras e através de uma conversão pode-se obter os índices das amostras positivas, desta forma podemos criar uma uma coluna chamada labels. Vale ressaltar que, para que não haja confusão entre os indexes, este processo é realizado individualmente para cada data frame, após este processo gera-se um único arquivo com todas as amostras rotuladas.

Extração de features

Para obter mais recursos dos dados foi realizado um reagrupamento, de tal forma que a cada N quantidade de amostras, eram extraídas as *features* espectrais e contínuas. A definição de qual *target* (normal ou com epilepsia) estaria associada a esse conjunto de amostras dependia de qual aparecia mais vezes. Inicialmente, considerou-se uma janela de 140 amostras, sendo esse valor obtido pela média da quantidade de amostras com epilepsia em cada arquivo. Contudo, como os dados são desbalanceados, a quantidade de amostras com epilepsia contabilizou apenas 10 no total, pois em todas as janelas geradas apenas 10 possuíam a maioria de labels convulsivas. Assim, duas possibilidades foram definidas:

- A quantidade N de amostras ser o mínimo disponível entre os arquivos, que no caso é 35 amostras ou
- utilizar 17 amostras, pois teria uma maior garantia de que todas as *targets* com epilepsia seriam usadas.

Ressalta-se que antes de realizar esse reagrupamento, o *dataframe* foi ordenado pela classe das *targets*, com isso é garantido que apenas uma vez será necessário verificar qual classe é dominante. Com a biblioteca foram obtidas ao todo 22 *features*.

Dados nulos

Como as classes são desbalanceadas, para a remoção dos dados nulos optou-se por diminuir a quantidade de *features* em vez da quantidade de amostras do conjunto de dados. Assim, para ambas as janelas de 17 e 35 amostras as colunas removidas foram: 'mean_hr', 'max_hr', 'std_hr', 'lf_hf_ratio', 'lfnu' e 'hfnu'. A tabela 1 apresenta a quantidade de nulos e valores infinitos em cada coluna e para cada quantidade de amostras adotada no trabalho.

Features Qtd. Amostras	'hfnu'	'lfnu'	'mean_hr'	'max_hr'	'std_hr'	'lf_hf_ratio'
17 amostras	1315	1315	7	7	7	1315
35 amostras	3	3	7	7	7	3

Tabela 1: Quantidade de valores nulos e infinitos para as *features*

Normalização

A fim de que tenhamos certeza de que uma feature não tenha prevalência sobre outra pelos fato de estar num *range* numérico maior, é necessário uma espécie de normalização dos dados. Duas técnicas populares são a Normalização (*Min Max Normalization*) e a Estandardização (*Z-Score*). A primeira acaba sendo sensível a *outliers* pelo fato de que ela limita os valores de 0 a 1, o que pode gerar vários valores aproximadamente nulos comparados com um único outlier que agora está limitado com o valor unitário. A segunda alternativa é mais viável pois não limita os dados, mas redistribui eles para uma distribuição com variância unitária. Essa segunda opção acaba sendo muito boa para o nosso dataset pois como foi visto, temos presença de vários valores infinitos, que mesmo após serem tratados, são indício de possíveis valores muito altos ao seu redor.

Modelos

Três modelos diferentes foram usados para treinamento, são eles, *Regressão Logística*, *Random Forest*, e *KNeighbors*. Para os dois últimos, foi implementado um *random search,* uma busca aleatória para otimização do modelo, no qual é verificado aleatoriamente quais conjuntos de hiperparâmetros geram melhores resultados. Em ambos também foi utilizado a técnica de validação cruzada *KFold,* que pode ser usada para melhorar a generalização do modelo quando se tem uma quantidade pequena de dados, e/ou quando há um certo desbalanceamento no mesmo.

Para uma comparação justa, os três modelos foram treinados com as mesmas features, as quais foram obtidas por meio de uma “janela temporal”. Essa janela é na verdade um parâmetro que indica a quantidade de amostras necessárias para a criação dos atributos, como valores de frequência cardíaca, min, max, e diversos outros.

Influência da Janela

De forma a definir um tamanho para a janela de amostragem (i.e quantas amostras utilizar para obter as *features*) inicialmente pensou-se em utilizar o valor médio do número do grupo de amostras da classe convulsão, o que corresponde a utilizar o tempo médio de convulsão, porém, ao utilizar uma janela de 140, obtemos apenas 518 amostras negativas e 10 amostras positivas, o que se mostrou insuficiente para o aprendizado dos modelos, desta forma, de modo a explorar a influência do tamanho da janela, realizou-se uma comparação com os seguintes valores: 35 (negativas: 2072, positivas: 40) e 17 (negativas: 4266, positivas: 83), uma vez que geram mais amostras positivas.

Deve-se mencionar que, de forma a realizar uma comparação justa e poder tirar conclusões da variação do tamanho da janela nos resultados, a comparação é realizada somente para valores de janelas onde se manteve as mesmas features, uma vez que ao variar a janela, algumas features apresentaram valores totalmente nulos ou infinitos, estas *features* foram removidas da análise. São consideradas *features* válidas as que apresentarem um score de correlação não nulo com as labels. Nas figuras abaixo, podemos ver as *features* e seu *score* de correlação para cada valor de janela.

Figura 2: Features para janelas de 17 amostras x score.

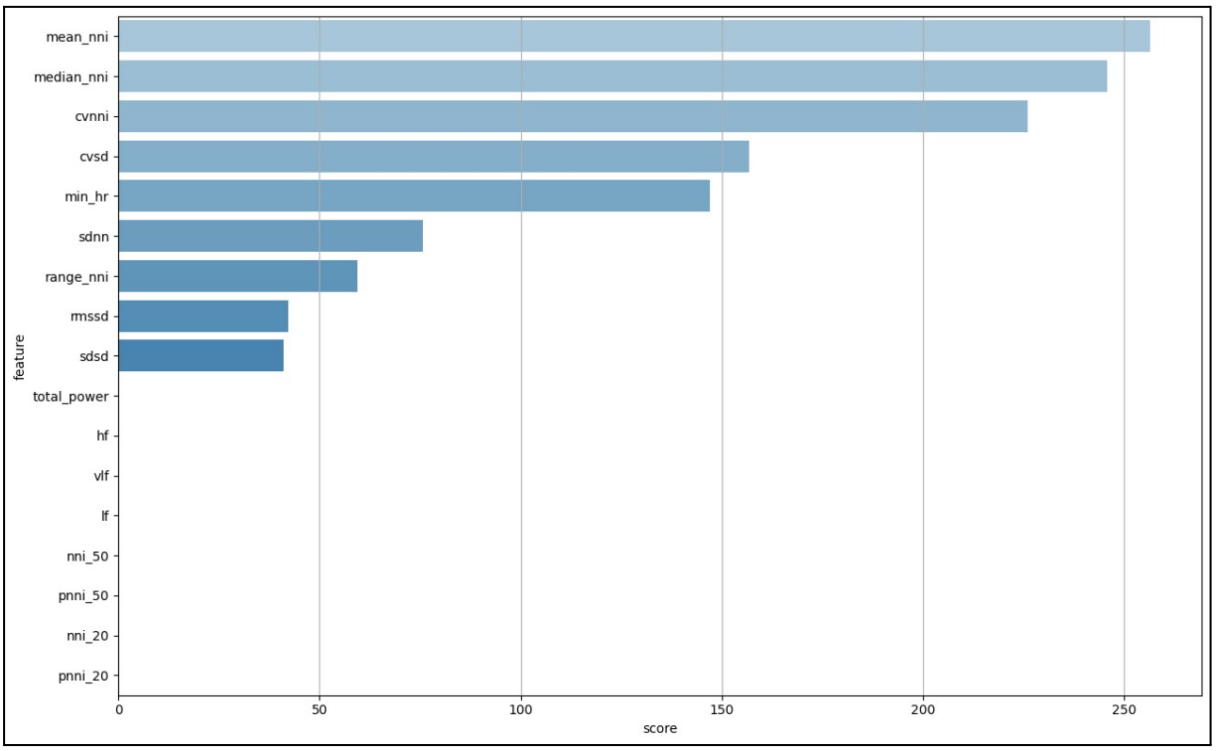
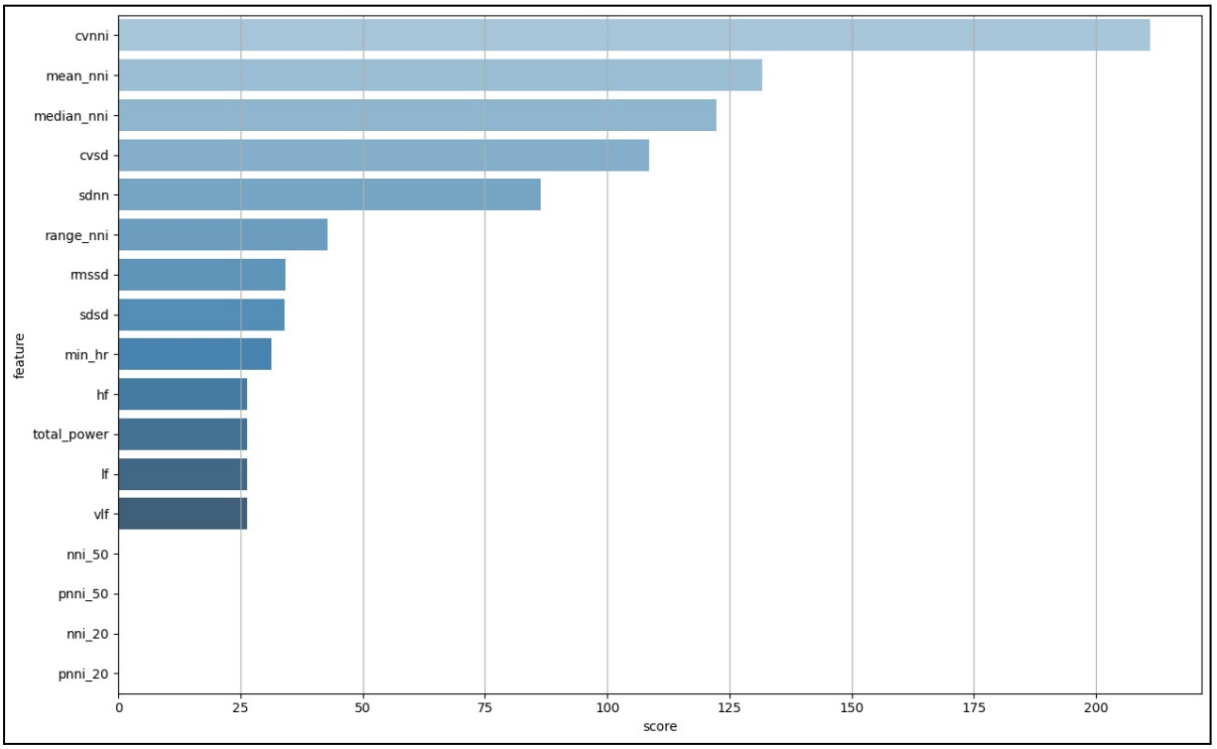


Figura 3: Features para janelas de 35 amostras x score.



Resultados

Métricas

A partir das métricas é possível verificar quão ideal o modelo treinado é, sendo que cada métrica avalia o resultado de uma maneira diferente. A métrica que teve maior peso na análise dos resultados foi o *Recall*, que segundo [1] é a razão entre os verdadeiros positivos com a soma dos verdadeiros positivos e os falsos negativos. Ainda segundo [1], a precisão indica quantas amostras classificadas como positivas são realmente positivas, assim sua ênfase é maior para os erros por falso positivo. Enquanto o Score F1, também de acordo com [1], é definido como uma média harmônica entre a precisão e o *Recall*, logo um modelo com Score F1 alto é capaz de acertar as predições e recuperar os exemplos da classe de interesse. No caso da classificação entre epilepsia e normalidade, é melhor o modelo identificar mais anomalias do que realmente há, visto que caso não identifique o problema pode causar a morte de uma pessoa.

Nas tabelas abaixo, podemos notar a tendência de todos os modelos em obter um bom desempenho na classe dominante (Normal), isso ocorre devido ao desbalanceamento dos dados. Ao trabalhar com um dataset desbalanceado, deve-se utilizar métricas que não sejam tendenciosas a favorecer classes com maior número de amostra. Uma possível solução, que não necessita da inserção de mais dados é dar pesos para cada classe, na tabela 4, podemos ver os resultados para os modelos onde foi possível adicionar um peso maior a classe positiva. Na figura 5, vemos uma comparação

entre os modelos e os métodos de balanceamento das amostras para a classe positiva. O balanceamento foi realizado utilizando *Oversampling* isto é, adicionando amostras da menor classe, para tal foram utilizados os métodos Naive Random, SMOTE e ADASYN.

Uma vez que os melhores resultados foram alcançados para a janela de 35, as análises são realizadas considerando esta janela.

Modelo	Classe	Precisão	Recall	Score F1
Random Forest	0	0.9960	0.9379	0.9660
	1	0.2413	**0.8400**	0.375
Regressão Logística	0	0.9823	0.9952	0.9887
	1	**0.5454**	0.2400	0.3333
KNeighbors	0	0.9787	0.9952	0.9869
	1	0.2857	0.080	0.1250

Tabela 2: Métricas obtidas para os modelos com janela de 17 amostras

Modelo	Classe	Precisão	Recall	Score F1
Random Forest	0	0.9979	0.9461	0.9713
	1	0.2000	**0.8750**	0.3255
Regressão Logística	0	0.9904	0.9942	0.9923
	1	0.5	0.375	0.4285
KNeighbors	0	0.9885	0.9980	0.9933
	1	**0.6666**	0.25	0.3636

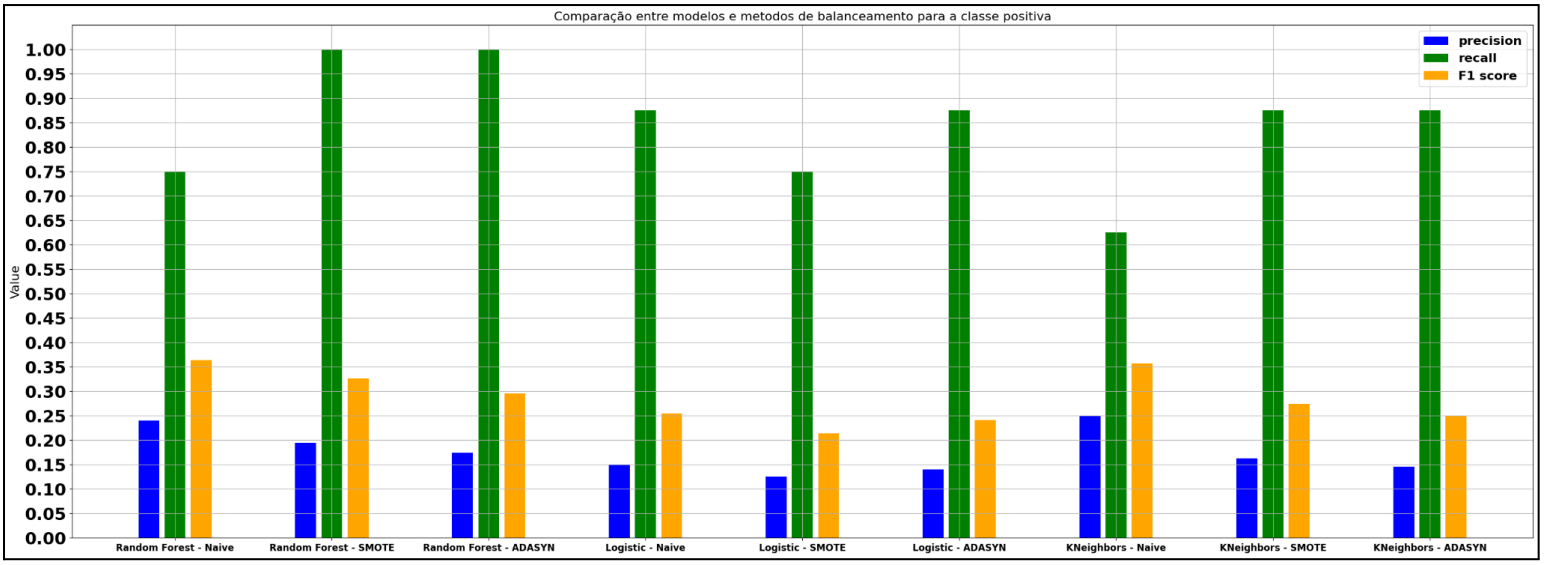
Tabela 3: Métricas obtidas para os modelos com janela de 35 amostras

Modelo	Classe	Precisão	Recall	Score F1
Random Forest (pesos 4 para 1)	0	0.9922	0.9903	0.9913
	1	**0.4444**	0.5	0.4705
Regressão Logística (pesos 90 para 1)	0	1.0	0.9019	0.9484
	1	0.1355	1.0	0.2388
Regressão Logística (pesos 80 para 1)	0	0.9978	0.9076	0.9506
	1	0.1272	**0.875**	0.2222

Tabela 4: Métricas obtidas para os modelos com janela de 35 amostras e balanceamento através de pesos

Obs: O Sklearn não proporciona adição de pesos para o método *KNeighbors*

Figura 4: Comparação entre os modelos e métodos de balanceamento de dados.



Matriz de confusão

Uma forma de se avaliar os resultados de um modelo é olhando para a matriz de confusão, que indica quantas amostras positivas foram classificadas corretamente (Verdadeiro positivo Verdadeiro negativo) e quantas foram classificadas erroneamente (Falso positivo e Falso negativo)

- TP: A classificação foi positiva e a label é positiva
- TN: A classificação foi negativa e a label é negativa
- FP: A classificação foi positiva e a label é negativa
- FN: A classificação foi negativa e a label é positiva

Figura 5: Entendendo a matriz de confusão.

		Valor Predito	
		Não	**Sim**
Real	**Não**	Verdadeiro Negativo (TN)	Falso Positivo (FP)
	Sim	Falso Negativo (FN)	Verdadeiro Positivo (TP)

Figura 6: Matrizes de confusão para cada modelo (janela de 17 amostras)

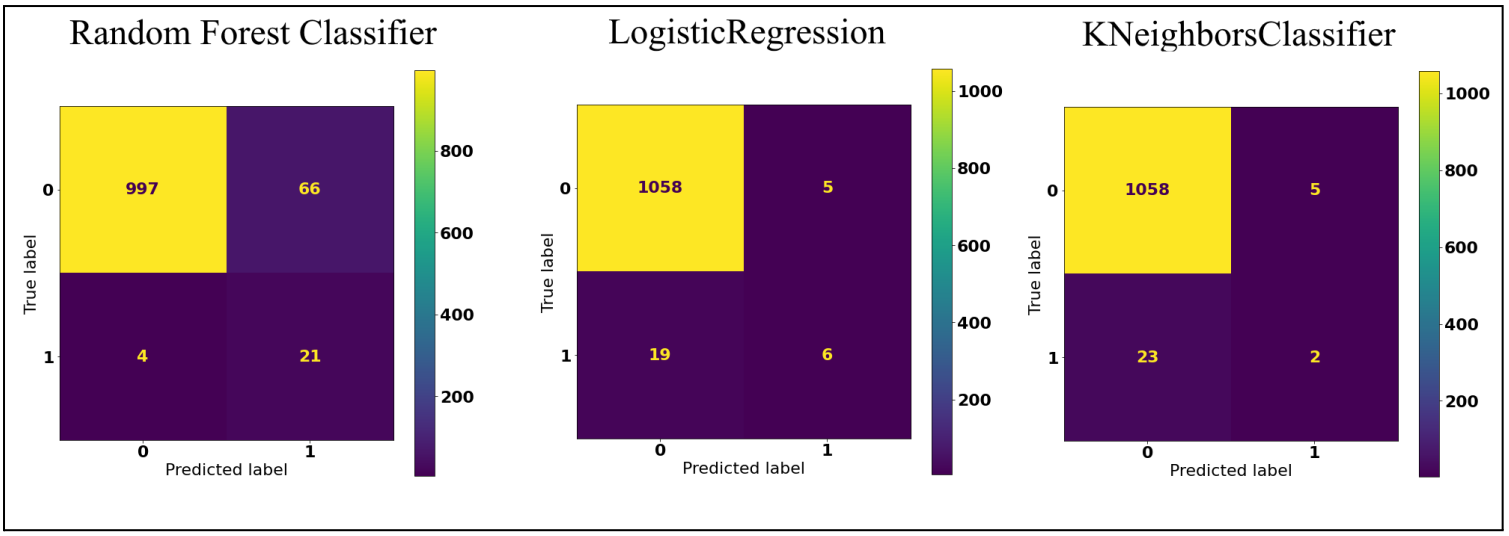


Figura 7: Matrizes de confusão para cada modelo (janela de 35 amostras)

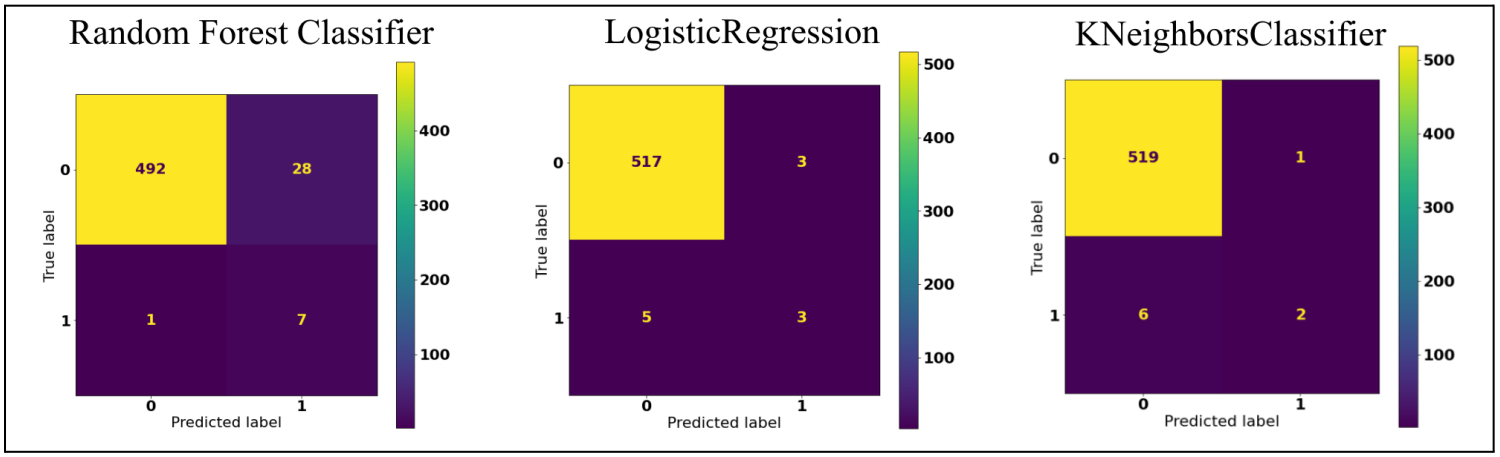
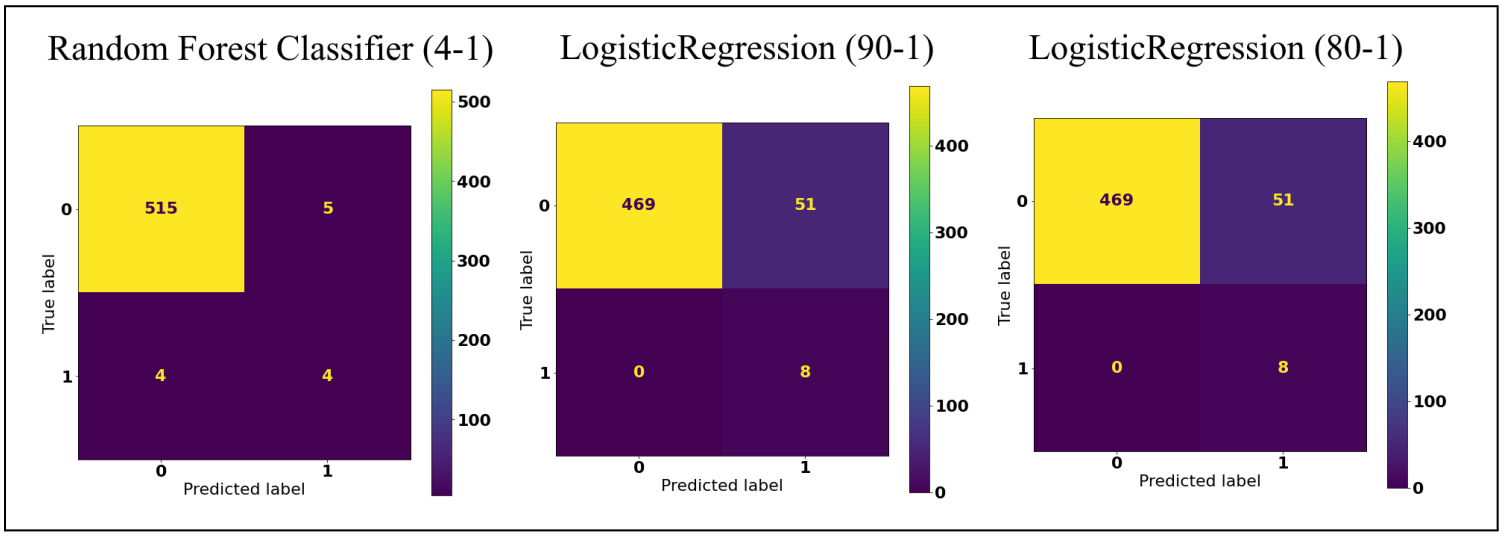


Figura 8: Matrizes de confusão para cada modelo (janela de 35 amostras com pesos)



Conclusão

Analisando as matrizes de confusão e a tabela 2, percebemos que, em termos de recall, o Random Forest obteve a melhor classificação, conseguindo prever 84% das ocorrências de epilepsia com uma precisão de 24.13%, a baixa precisão explica o alto valor de recall, uma vez que tende-se a haver uma relação inversa entre precisão e recall. Neste conjunto de testes, temos 1088 amostras com features extraídas que, para uma janela de 17, corresponde a 18496 amostras de variação da frequência cardíaca, para uma frequência de amostragem de 200 amostras/segundos, temos um tempo total de monitoramento de aproximadamente 1 minuto e 33 segundos, uma vez que 87 amostras com features foram previstas como sendo positivas, temos uma equivalência de 7.395 segundos onde o paciente seria avisado a permanecer em atenção, caso o algoritmo pudesse prever com antecedência o ataque epilético. Com esta pequena análise, podemos ver que o fato de ter uma baixa precisão não

necessariamente significa que o modelo não possa ser utilizado, uma vez que é importante que o recall seja o mais alto possível para evitar que o paciente seja pego de surpresa, para este modelo o paciente ficaria em estado de atenção somente ~8% do tempo obtendo um recall de 84%. Idealmente, um modelo a ser utilizado em produção precisaria obter um recall próximo a 100%, e uma razão entre tempo de monitoramento e alerta baixa, de forma com que o uso do aparelho não seja incômodo e que seja de fato benéfico para o paciente, mesmo errando uma porção significativa do tempo de alerta. O método de regressão logística obteve a maior precisão para a classe de interesse, 54.54%, porém seu recall é de apenas 24%. Pelos motivos explicados acima, para a janela de 17 amostras, o Random Forest apresentou os melhores resultados. Analisando de forma semelhante os resultados para a janela de 35 amostras, vemos que excluindo-se o método KNeighbour, um aumento do tamanho da janela proporcionou um aumento na taxa de recall e uma diminuição na precisão para a classe de interesse, para o KNeighbour ambas as métricas aumentaram, desta forma, pode-se dizer que o aumento da janela contribuiu para uma melhora no modelo (aumento no recall), esta melhora vem pelo fato das features serem mais representativas, uma vez que com o aumento da janela, as métricas são retiradas em cima de um conjunto maior de amostras. Novamente o método Random Forest alcançou os melhores resultados, com uma precisão de 20% e recall de 87,5%, o que corresponde a uma taxa de monitoramento sobre tempo de alerta de 1.51% no conjunto de testes. Na tabela 4 vemos que adicionar um peso maior para a classe positiva tem um efeito de aumentar a precisão no Random Forest e de aumento de recall na Regressão logística, com isso tem-se também a possibilidade do uso da regressão logística como modelo, porém, configurando-se os pesos para obter 87.5% de recall (mesmo valor para o melhor modelo até então), obtemos uma precisão de 12.72%, desta forma o modelo de Random Forest ainda apresenta melhores resultados por apresentar uma precisão maior (20%). Em relação às experimentações realizadas com balanceamento de dados, pela figura 4 podemos concluir que o método Naive proporcionou uma melhor precisão para todos os casos e um score F1 maior, porém, na média foi o que resultou no menor recall. Levando em consideração a necessidade de alto recall, os melhores resultados em geral foram alcançados com o método ADASYN, porém, o melhor modelo encontrado neste trabalho foi o Random Forest com uso de SMOTE para balanceamento dos dados, alcançando uma precisão de aproximadamente 19,51% com um recall de 100% e F1 score de 32,65%, este modelo é considerado o melhor por possuir uma melhor ponderação precisão recall, onde o recall possui um peso maior.

Um grande dificuldade encontrada no treinamento dos modelos é o desbalanceamento dos dados e a falta de amostras positivas, com o decorrer das experimentações, foi possível concluir que a extração de features deve ser realizada em amostras da mesma classe, de forma a aproveitar ao máximo as amostras positivas e não confundir o modelo com a mistura de amostras. Para a obtenção de melhores resultados, sugere-se obter mais amostras do conjunto positivo de forma a balancear o dataset de forma natural, o que deve proporcionar um melhor desempenho nos modelos.

Bibliografia

[3] Epilepsia | Biblioteca Virtual em Saúde MS (saude.gov.br)

[2] Diagnóstico da Epilepsia: Atenção ao Coração | Clinica Regenerati

[1] Métricas de Avaliação em Machine Learning: Classificação | by Kunumi | Kunumi Blog | Medium