1 2 90

UNIVERSIDADE D COIMBRA

PROJETO COVID-19

Machine Learning

Ana Carolina Morais Nº2021222056
Fernanda Fernandes Nº2021212260

Introdução

O principal objetivo deste projeto é desenvolver um modelo de *Machine Learning* que ajude a decidir se um paciente com suspeita de COVID deve permanecer hospitalizado para exames adicionais ou pode ser enviado para casa.

Para alcançar este objetivo, é utilizado as seguintes informações:

1. Uma regra pré-definida, que indica a relação entre a dificuldade respiratória e a temperatura corporal.
2. Um *dataset* com variáveis numéricas e informações categóricas de 600 pacientes, cobrindo dados como idade, estado vacinal e sinais vitais.
3. Eletrocardiogramas (*ECGs*) em formato binário, representadas como matrizes de 21 por 21, que podem conter padrões visuais úteis para a análise clínica.

Ao combinar esses elementos, o projeto explora diferentes técnicas de *Machine Learning*, como *clustering*, árvores de decisão, redes neuronais e deep learning (e.g, *CNNs*), para criar um sistema eficiente e confiável. A abordagem será avaliada com base na capacidade de integrar essas fontes de informação para fornecer decisões clínicas precisas.

Etapas

A aplicação dos modelos enunciados, anteriormente, exige etapas sistemáticas que antecedem a construção dos mesmos:

1. Pré-processamento de dados tabulares e imagens.
2. Seleção de *features* para reduzir dimensionalidade e melhorar o desempenho.
3. Construção e treino do modelo.

É, de especial atenção, que no caso do *Clustering* não existe propriamente o treino do modelo nem a seleção de *features,* mas isso irá ser abordado posteriormente.

• Pré-processamento

O pré-processamento dos dados tabulares incluiu diversas etapas fundamentais. Primeiramente, foi adicionada uma nova variável (*feature*) binária baseada numa regra explícita: pacientes com dificuldade respiratória igual ou superior a 2 e temperatura corporal acima de 37,8 graus foram classificados como casos críticos.
Em seguida, lidou-se com valores ausentes, substituindo-os pela média ou moda, dependendo do tipo de dado, e realizamos a verificação se existe dados duplicados e, no caso afirmativo, são então removidos, para evitar viés no treino. Para as variáveis contínuas, como idade, frequência cardíaca e pressão arterial, aplicou-se uma normalização para o intervalo [0, 1], utilizando o método de escalonamento Min-Max.

Os dados de imagens de ECGs passaram por um pré-processamento distinto. Cada matriz 21x21 foi achatada, convertendo as imagens em vetores unidimensionais de 441 elementos. Essa transformação simplificou o processamento posterior, mantendo, porém, toda a informação binária contida nas imagens originais.

• Seleção de features

Após o pré-processamento, a seleção de *features* desempenhou um papel crucial na preparação dos dados tabulares. Três abordagens principais foram utilizadas para identificar as variáveis mais relevantes. A primeira envolveu o uso de *Random Forest* para calcular a importância relativa das features, com base na contribuição de cada uma para o desempenho do modelo (Fig. 1). Os resultados foram visualizados em gráficos de barras, destacando as variáveis mais significativas. Em seguida, aplicou-se o teste *ANOVA*, que mede a relevância estatística das variáveis relativamente à variável alvo. Essa abordagem permitiu priorizar *features* com maior significância estatística(Fig. 2). Por último, utilizou-se a métrica de informação mútua para avaliar a dependência entre as variáveis e o alvo, identificando padrões adicionais de relevância(Fig. 3).

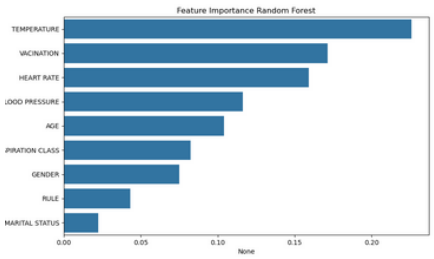

Fig. 1 - Importance Random Forest

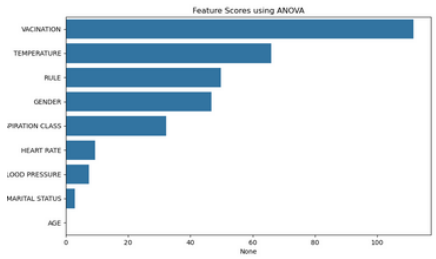

Fig. 2 - Score ANOVA

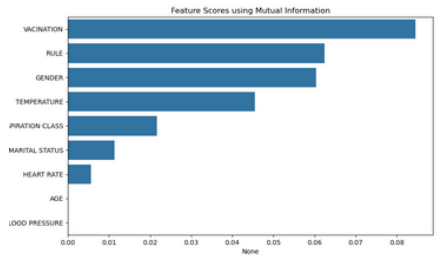

Fig. 3 - Score Mutual Information

Embora os gráficos obtidos (Fig. 2 e Fig. 3) indiquem uma forte importância da *feature* "*rule*", esta não será incluída. Isso deve-se ao fato de a sua relação com a variável "*target*" poder introduzir enviesamento. A "*rule*" afirma que, se for verdadeira, o utente permanece no hospital, o que não está totalmente alinhado com a definição do "*target*", conforme ilustrado na figura abaixo.

```
Number of coincidences between RULE and TARGET: 427 out of 600 cases
Percentage of coincidences: 71.17%
```

Combinando os resultados dessas três técnicas, foi elaborado um '*ranking*' global ponderado para selecionar as variáveis mais importantes. Essa abordagem garantiu que as informações mais úteis fossem priorizadas, evitando o "*overfitting*" e reduzindo a complexidade do modelo. No final, as sete *features* mais relevantes foram mantidas para compor o conjunto final de dados tabulares a ser utilizado no treinamento.

```
Selected features:
['AGE', 'HEART RATE', 'SYSTOLIC BLOOD PRESSURE', 'TEMPERATURE', 'VACINATION', 'RESPIRATION CLASS', 'GENDER']
```

• Integração dos dados tabulares e das ECGs

A integração dos dados tabulares e das imagens de ECG foi feita através da concatenação das features selecionadas com os vetores derivados das imagens. A ECG foi utilizada apenas nas Redes Neurais Convolucionais (CNNs), pois estas são adequadas para explorar padrões espaciais nas imagens. Nas redes neuronais convencionais e nas árvores de decisão, a ECG não foi utilizada, pois essas abordagens não lidam bem com dados espaciais, sendo mais eficazes em dados tabulares.

Clustering

Embora o objetivo principal deste projeto seja a classificação dos pacientes em dois grupos (retornar para casa ou permanecer hospitalizado), o *clustering* foi utilizado como uma etapa essencial para a análise exploratória dos dados e o entendimento da sua distribuição. Os algoritmos aplicados incluem K-Means, DBSCAN, Agglomerative Clustering e Subtractive Clustering, para estes foram realizados vários testes para otimizar os parâmetros, sendo, posteriormente, avaliados quanto ao *SSE* e *Silhouette Score.* O resultado encontra-se abaixo.

```
K-Means Silhouette Score: 0.36761574500899946
DBSCAN Silhouette Score: 0.5359905714415578
Agglomerative Clustering Silhouette Score: 0.36761574500899946
Subtractive Clustering Silhouette Score: 0.47763735073067837
K-Means SSE: 779.9162343917938
Subtractive Clustering SSE: 122.49195956807385
```

Entre os métodos, o *DBSCAN* apresentou o melhor desempenho em termos de *Silhouette Score* (0.5359), indicando maior coesão e separação entre os '*clusters*' formados, foram formados 48 '*clusters*'. Além disso, foi realizada uma análise detalhada, que incluiu a correlação das variáveis de domínio com os '*clusters*' e a comparação com a variável-alvo "*target*".

A análise de correlação mostrou que variáveis como "*Respiration Class*", "*Temperature*" e "*Age*" estavam associadas a '*clusters*' indicativos de maior gravidade clínica, como pacientes mais velhos e com maior dificuldade respiratória. A comparação com a variável-alvo "*target*" demonstrou que o *DBSCAN* é eficaz em identificar padrões úteis para a classificação. A avaliação, que considerou tanto métricas globais quanto variáveis clínicas, reforçou o *DBSCAN* como uma ferramenta valiosa para análise exploratória, auxiliando na escolha de features para modelos supervisionados. Neste caso, foi possível confirmar a seleção de *features* que foi utilizada nos algoritmos seguintes.

Decision Tree

Como referido anteriormente, começamos por fazer:

- Pré-processamento de dados tabulares.
- Seleção de *features* para reduzir dimensionalidade e melhorar o desempenho.

Em seguida, no processo de treinamento do modelo, abordamos diferentes modelos de árvores de decisão utilizando critérios como *gini* e *entropy*, bem como parâmetros variados de profundidade máxima, para explorar a capacidade preditiva dos dados. Além disso, foram implementamos modelos baseados em *ID3*, *CART* e uma variação aproximada de *Gain Ratio*.

Após o treinamento, os modelos foram avaliados utilizando o conjunto de teste, com métricas como a *accuracy*, relatórios de classificação, que incluem métricas como *precision*, *recall*, *f1-score* e *support*, e matrizes de confusão. Esta análise foi importante para comparar modelos.

Com base na análise dos modelos apresentados, o *Gini Depth* com profundidade 3 mostrou-se o melhor modelo com uma *accuracy* de 81.67% e um f1-score macro médio de 0.77. Este oferece uma boa combinação entre precisão e *recall*, sendo adequado para balancear decisões corretas de alta. No entanto, o *recall* para a classe 1.0 (internação) foi moderado (0.55), o que pode ser uma limitação em situações críticas, como na hospitalização de pacientes com sintomas graves.

O modelo *ID3* ocupa o segundo lugar, apresentando uma *accuracy* ligeiramente inferior (80.83%) e o melhor *recall* para a classe 1.0 (0.71), o que o torna ideal para priorizar decisões de internação. O equilíbrio entre as métricas sugere que pode ser mais confiável num ambiente clínico onde falsos negativos precisam de ser minimizados. Já o *Gini Depth* com profundidade 5, com *accuracy* de 80.00%, oferece um desempenho razoável, mas menos robusto em comparação ao *ID3*.

O recall para a classe 1.0 foi de 0.66, indicando que é menos eficaz na detecção de casos críticos. O *Gain Ratio* com uma *accuracy* de 77.50% apresentou resultados consistentes, mas inferiores em termos de *recall* e *f1-score* para a classe minoritária.
Os modelos *Gini Depth* com profundidade 10, *Gini Depth* sem profundidade e o *CART* ficaram empatados no desempenho, com *accuracy* de 75.83% e menor *recall* para a classe 1.0 (0.58). Estes modelos são menos adequados, especialmente em contextos onde é essencial evitar a alta de pacientes em estado grave.

Redes Neuronais & CNNs

- ## Redes Neuronais (MLNN)

Relativamente às redes neuronais, uma parte essencial do processo foi a otimização dos hiperparâmetros. Para isso, utilizámos o método *GridSearch*, uma abordagem sistemática e eficiente para encontrar as melhores combinações de parâmetros que maximizam o desempenho do modelo.
O *GridSearch* consiste em explorar exaustivamente um espaço de parâmetros pré-definidos, avaliando cada combinação por meio de validação cruzada. No caso das redes neuronais (MLNN) utilizadas neste projeto, diversos parâmetros foram ajustados, incluindo o número de neurónios nas camadas ocultas, funções de ativação, otimizadores, taxas de regularização, e estratégias de aprendizado.

Assim, os melhores parâmetros encontrados foram: 200 e 100 neurónios em duas camadas ocultas, utilizando a função de ativação tangente hiperbólica (*tanh*), o otimizador *Adam*, uma taxa de regularização de 0.0001 e aprendizado adaptativo com *batches* de 32.

Essa configuração foi então utilizada para treinar o modelo final. Cada um destes parâmetros desempenha um papel fundamental no desempenho da rede. O número de neurónios nas camadas ocultas determina a capacidade do modelo de capturar padrões complexos; mais neurónios aumentam a expressividade da rede, mas também podem levar ao *overfitting* se não forem adequadamente controlados. A função de ativação *tanh* foi escolhida pela sua capacidade de modelar relações não lineares eficientemente, especialmente com dados normalizados. O otimizador *Adam* combina eficiência e estabilidade, ajustando automaticamente a taxa de aprendizado ao longo do treinamento, enquanto a regularização ajuda a evitar *overfitting* penalizando pesos excessivamente grandes. Por fim, o aprendizado adaptativo ajusta dinamicamente a taxa de aprendizado, melhorando a convergência do modelo.

Além disso, métricas como *sensitivity*, *specificity*, *f1-score*, *accuracy* e curva *ROC* foram analisadas para garantir que o modelo fosse robusto e eficiente.

Os resultados obtidos demonstraram um desempenho sólido e equilibrado. Durante a validação cruzada, o modelo apresentou um *f1-score* médio de 0.7277 e uma *accuracy* média de 78.13%, evidenciando consistência na capacidade de generalização. No teste final, o modelo alcançou uma *sensitivity* de 71.05%, uma *specificity* de 92.68%, um *F1-Score* de 0.7606 e uma *accuracy* de 85.83%.

A matriz de confusão revelou 76 verdadeiros negativos e 27 verdadeiros positivos, destacando a capacidade do modelo de diferenciar corretamente as classes, com apenas 11 falsos negativos e 6 falsos positivos. Esta análise confirma que o modelo é adequado para o problema proposto e pode servir como uma ferramenta de suporte eficiente, particularmente em cenários onde minimizar erros em ambos os lados do espectro é crítico. A robustez demonstrada valida a abordagem de otimização utilizada e destaca a eficácia da configuração final no aprendizado dos padrões subjacentes aos dados.

- **CNNs**

Na aplicação do modelo de rede neural convolucional (*CNN*) combinado com dados tabulares para classificação binária, o processo começa mais uma vez com a preparação dos dados, onde as imagens são ajustadas para uma forma específica exigida pelo modelo *CNN*, enquanto os dados tabulares são mantidos na sua estrutura original. Estes dados foram então divididos em conjuntos de treinamento e teste, utilizando 80% para o treino e 20% para o teste tal como nos outros modelos.
O modelo foi definido em duas partes: a *CNN* processa os dados de imagem com camadas convolucionais e de pooling para extrair características relevantes, enquanto um submodelo separado com camadas densas lida com os dados tabulares. As saídas dos dois modelos são combinadas numa camada de fusão, seguidas por camadas densas adicionais e uma camada de ativação *sigmoid*, que prevê as probabilidades para a classificação binária.

Este foi treinado utilizando o *EarlyStopping*, que interrompe o treinamento automaticamente quando o desempenho no conjunto de validação não melhora após algumas épocas consecutivas, evitando o *overfitting*. A *CNN* utiliza filtros em camadas convolucionais para extrair padrões espaciais das imagens, seguidos por camadas de *pooling* que reduzem a dimensionalidade, mantendo as características mais relevantes. Paralelamente, os dados tabulares passam por camadas densas que abstraem as características, combinados numa camada de concatenação. Esta abordagem de fusão explora sinergicamente as informações tabulares e visuais para criar representações enriquecidas.

O modelo foi compilado com a função de perda *binary_crossentropy*, o otimizador Adam e a métrica de *accuracy*, e treinado em múltiplas épocas com validação em parte do conjunto de treino Foram feitos vários testes de ajuste da rede neuronal convulocional, para obter a *CNN* mais adequada ao problema. Após o treinamento, o modelo foi avaliado utilizando o conjunto de teste. Ele gera previsões em forma de probabilidades, convertidas em classificações binárias. As métricas de avaliação incluem a matriz de confusão, *sensitivity*, *specificity*, *f1-score* e a *accuracy.* A matriz de confusão é usada para analisar classificações corretas e incorretas para ambas as classes, enquanto a curva *ROC* (*Receiver Operating Characteristic*) é apresentada para ilustrar o equilíbrio entre a taxa de verdadeiros positivos e falsos positivos. Este pipeline aproveita eficazmente dados tabulares e visuais, integrando as forças de ambos os tipos de dados num modelo robusto de aprendizagem profunda, com especial atenção à estabilidade do treinamento e à eficiência no aprendizado das características críticas.

Após a realização de vários testes para identificar os melhores parâmetros para o modelo, considerando os seguintes atributos: *epochs*, *batch_size*, *dropout_rate*, l*earning_rate*, *conv_filters* e *dense_units*. Obtemos a melhor configuração de *epochs*=40, *batch_size*=64, *dropout_rate*=0.5, *learning_rate*=1e-3, *conv_filters*=[32, 64, 128] e *dense_units*=128, com uma *accuracy* de 87,5%, um *f1-score* de 0,8, sensibilidade da Classe 0 de 91%, especificidade da Classe 0 de 79%, sensibilidade da Classe 1 de 79% e especificidade da Classe 1 de 91%. A matriz de confusão revelou uma boa distribuição de acertos, indicando 75 verdadeiros negativos e 30 verdadeiros positivos, destacando a capacidade do modelo de diferenciar corretamente as classes, com apenas 8 falsos negativos e 7 falsos positivos.

Conclusão

Neste projeto, a integração de diferentes técnicas de *Machine Learning* demonstrou ser essencial para abordar o problema da decisão clínica em pacientes com suspeita de *COVID-19*. Entre os métodos analisados, as Redes Neuronais Convolucionais (*CNNs*) mostraram-se superiores, destacando-se como a abordagem mais robusta e eficiente. A capacidade exclusiva das *CNNs* de processar imagens em formato de matrizes binárias, como os eletrocardiogramas (*ECGs*), permite explorar padrões espaciais que outros métodos, como árvores de decisão e Redes Neuronais (*MLNN*), não conseguem capturar. Esta característica torna as *CNNs* indispensáveis em cenários onde dados visuais desempenham um papel crítico na análise clínica.

No entanto, árvores de decisão e Redes Neuronais (*MLNN*) desempenharam papéis complementares, sendo eficazes na análise de dados tabulares. Enquanto as árvores de decisão apresentaram modelos com boa interpretabilidade, essenciais em contextos clínicos, as *MLNN* destacaram-se pela sua flexibilidade em capturar relações não lineares nos dados.

A análise exploratória com *clustering*, especialmente o *DBSCAN*, provou ser fundamental para compreender a distribuição dos dados e comprovar a escolha das *features* mais relevantes. Este método identificou padrões significativos que sustentaram as etapas subsequentes de modelagem, garantindo maior precisão e eficiência nos modelos supervisionados.

A escolha de uma abordagem híbrida, que combina diferentes métodos de *Machine Learning*, mostrou-se essencial para lidar com a complexidade dos dados utilizados. Esta abordagem foi preferida em detrimento de métodos mais simples, como o *perceptron*, cujas limitações em capturar relações não lineares e integrar dados heterogéneos seriam ineficazes neste cenário.

Os resultados alcançados não apenas validam a eficácia dos modelos desenvolvidos, mas também destacam o potencial de uma abordagem multidisciplinar para resolver problemas clínicos reais. Este trabalho reforça a importância da integração de técnicas complementares e demonstra como o uso estratégico de dados tabulares e visuais pode resultar em sistemas de decisão clínica robustos e confiáveis.