# PROJETO 3 - APLICAÇÃO DO ALGORITMO DE *RANDOM FOREST*

*Fernanda Macedo de Sousa - 17/0010058*
*Mariana Alencar do Vale - 16/0014522*

**Resumo:** Este relatório apresenta a implementação e análise do resultado do algoritmo *Random Forest* aplicado em um conjunto de dados relativos a dados clínicos de pacientes com suspeita de COVID-19, como terceiro projeto da disciplina Introdução à Inteligência Artificial.

**Palavras-chave**: AI; Machine Learning; Random Forest; COVID-19; Python;

## 1 Introdução

No contexto pandêmico de COVID-19 em 2020, o Hospital Israelita Albert Einstein, São Paulo (Brasil), coletou, organizou e disponibilizou dados de 5644 casos de COVID-19. Estes dados clínicos foram disponibilizados anonimamente, contendo vários exames reais de saúde relacionados aos pacientes.

O terceiro projeto da disciplina consiste em desenvolver e aplicar um código de "Random Forest" nesses dados, e predizer/responder algumas questões relacionadas à capacidade do modelo implementado de prever diagnósticos de pacientes e recomendações da forma de acompanhamento médico.

## 2 Materiais e métodos

O projeto foi realizado através do Google Colab com a linguagem de programação Python, de forma que os métodos não foram separados tão rigorosamente, mas fora seguido o processo comumente usado na implementação de um modelo de *Machine Learning* através do uso de um *notebook*. O arquivo "Como abrir o trabalho pelo Google Colab.pdf" apresenta instruções para a execução do *notebook* por essa plataforma.

Primeiro, os dados coletados no site kaggle são armazenados manualmente na aba de arquivos do Google Colab e a biblioteca pandas foi importada para trabalhar com leitura e manipulação de dados. Com funções da biblioteca pandas foi possível realizar a leitura do arquivo dataset.xlsx, além de realizar a substituição de todos os campos do *dataframe* contendo o valor NaN por zero, como sugerido no fórum da disciplina.

Uma variável foi utilizada para conter a coluna da tabela com os resultados da primeira questão ('diagnostic'), colunas contendo valores do tipo *string* e colunas relacionadas ao resultado ("Patient addmited to") foram removidas para então atribuir as colunas restantes da planilha à variável de predição.

A função *train_test_split* da biblioteca sklearn foi utilizada para dividir os dados em 30% teste e 70% para treino do modelo.

O modelo classificador criado para a Random Forest, é composto por 100 árvores. Após a inicialização do modelo, o mesmo foi treinado (método *fit*) a partir dos dados de treino (70% do conjunto inicial).

A partir do modelo treinado, foi verificada a relevância de todas as colunas de preditores do modelo, utilizando o atributo do modelo feature_importances_. Esse conjunto de atributos mais relevantes foram colocados em ordem decrescente, conforme apresenta a **Tabela 1**. Logo após, a acurácia do diagnóstico para os dados que foram testados (30% do conjunto de dados original) foi verificada e analisada, conforme mostra a **Tabela 2**.

A fim de obter as informações pertinentes à segunda questão proposta, uma nova coluna, denominada como "addimited unit", foi adicionada à tabela dos dados. A mesma, armazena um valor de 0 a 4, onde 0 representa que o paciente não foi internado, 1 representa que o paciente foi internado na enfermaria, 2 representa que paciente foi internado em unidade semi-intensiva e 3 representa que o paciente foi internado em unidade intensiva, resultados obtidos a partir da análise das colunas "Patient addmited to", onde o valor 0 representa pacientes que receberam o valor 0 nas três colunas.

Após a adição da nova coluna utilizada para o resultado, os passos para a utilização do modelo de Random Forest é análogo ao anterior. Entretanto, desta vez utilizando-se da coluna recém-criada como diagnóstico e removendo-a da variável de predição. O resultado final da capacidade de predição para determinar onde o paciente deve ser tratado foi analisado, além de obtidas também quais colunas foram relevantes.

Mais comentários a respeito do processo de implementação podem ser lidos nos textos atrelados ao próprio arquivo "Trabalho3_IIA.ipynb".

## 3 Resultados quadro, gráficos e figuras

| | feature | importance |
|---|---|---|
| 0 | Patient age quantile | 0.191283 |
| 8 | Leukocytes | 0.071378 |
| 13 | Monocytes | 0.068363 |
| 3 | Platelets | 0.057373 |
| 19 | Proteina C reativa mg/dL | 0.039146 |
| 11 | Eosinophils | 0.032021 |
| 1 | Hematocrit | 0.025452 |
| 10 | Mean corpuscular hemoglobin (MCH) | 0.025262 |
| 5 | Red blood Cells | 0.025170 |
| 2 | Hemoglobin | 0.024588 |

**Tabela 1 –** *Dataframe* gerado com as dez colunas mais relevantes para o diagnóstico de COVID-19

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| negative | 0.91 | 1.00 | 0.95 | 1529 |
| positive | 1.00 | 0.03 | 0.06 | 165 |
| accuracy | | | 0.91 | 1694 |
| macro avg | 0.95 | 0.52 | 0.50 | 1694 |
| weighted avg | 0.91 | 0.91 | 0.86 | 1694 |

**Tabela 2 –** Precisão para diagnóstico de COVID-19

| SARS-Cov-2 exam result | |
|---|---|
| negative | 5086 |
| positive | 558 |
| dtype: int64 | |

**Tabela 3 –** Contagem do número de casos positivos e negativos utilizados para treinamento do modelo

| | feature | importance |
|---|---|---|
| 19 | Proteina C reativa mg/dL | 0.078876 |
| 3 | Platelets | 0.040879 |
| 11 | Eosinophils | 0.037662 |
| 0 | Patient age quantile | 0.036031 |
| 35 | pO2 (venous blood gas analysis) | 0.035912 |

**Tabela 4 –** Métricas mais relevantes para determinar a ala do paciente

| addimited unit | |
|---|---|
| 0 | 5474 |
| 1 | 79 |
| 2 | 50 |
| 3 | 41 |
| dtype: int64 | |

**Tabela 5 –** Contagem do número de exemplos no *dataframe* usado para treinar o modelo, onde o paciente foi tratado em cada caso

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.98 | 1.00 | 0.99 | 1643 |
| 1 | 0.00 | 0.00 | 0.00 | 23 |
| 2 | 0.12 | 0.06 | 0.08 | 16 |
| 3 | 1.00 | 0.17 | 0.29 | 12 |
| accuracy | | | 0.97 | 1694 |
| macro avg | 0.53 | 0.31 | 0.34 | 1694 |
| weighted avg | 0.96 | 0.97 | 0.96 | 1694 |

**Tabela 6 –** Precisão resultante a respeito da capacidade de previsão do tipo de acompanhamento em que o paciente deve ser tratado

## 4 Análise de Resultados

Para a amostra de dados fornecida, baseando-se nos dados de laboratório (sem PCR), podemos perceber que o modelo implementado com o algoritmo de *Random Forest* consegue obter com 91% de precisão, os casos em que o diagnóstico é negativo para COVID-19, conforme apresenta a **Tabela 2**. Além disso, obteve 100% de precisão para prever os casos positivos de COVID-19. Entretanto, a métrica *recall* para o diagnóstico positivo está baixa, o que indica que os dados utilizados para o treinamento do modelo estão desbalanceados

(existem muito mais diagnósticos negativos do que positivos no *dataframe* utilizado). Isso pode ser verificado ao realizar a contagem dos resultados dos diagnósticos (PCR), conforme mostra a **Tabela 3.** Ou seja, esse modelo está mais preparado para previsão de diagnósticos negativos do que positivos para a doença.

Para verificar as 10 colunas (testes/variáveis) mais relevantes para o diagnóstico final de COVID-19, criamos um *dataframe* com duas colunas (o nome da feature e a respectiva importância no modelo). Depois, ordenamos em ordem da maior para a menor. Esse *dataframe* é apresentado na **Tabela 1.**

A **Tabela 6** apresenta a precisão resultante a respeito da capacidade de previsão do tipo de acompanhamento em que o paciente deve ser tratado no caso de diagnóstico positivo para COVID-19 (0 - acompanhados em casa, 1 - internados em enfermaria, 2 - internados em unidade semi-intensiva, 3 - internados em unidade intensiva), baseando-se nos dados de laboratório. Podemos observar que podemos predizer com **98% de precisão** os casos em que o paciente tem que ser **acompanhado em casa**. Para os demais casos, obtemos as seguintes precisões:

1. 0% de precisão para pacientes que devem ser internados em enfermaria
2. 12% de precisão para pacientes que devem ser internados em unidade semi-intensiva
3. 100% de precisão para pacientes que devem ser internados em unidade intensiva

Entretanto, a métrica *recall* para a previsão do tipo de acompanhamento indicado está baixa para o casos 1, 2 e 3. Isso indica que os dados estão desbalanceados, visto que a Floresta Randômica (*Random Forest*) foi treinada com muito mais exemplos em que o paciente teria de ser acompanhado em casa, de acordo com a contagem apresentada na **Tabela 5**.

Dessa forma, os únicos caso que podemos prever de forma segura quando o diagnóstico é positivo é o caso em que o paciente tem que ser acompanhado em casa. Os demais apresentam níveis de acurácia muito abaixo do ideal. Considerando o contexto médico em que o algoritmo é utilizado, o ideal é que a acurácia fosse mais confiável para diagnosticar corretamente como o paciente deve ser acompanhado.

## 5 Considerações Finais/Conclusões

A amostra de dados fornecida possui diagnósticos desbalanceados (como mostra a **Tabela 3** e a **Tabela 5**), logo, o algoritmo terá mais precisão com resultados negativos para COVID-19 do que para os positivos. O mesmo vale para a questão das alas de internação, uma vez que o número de pacientes acompanhados em casa era muito maior do que o dos internados.

Pode-se interferir que se a amostra tivesse resultados mais próximos um do outro, o algoritmo teria uma melhor acurácia.

**Referências Bibliográficas**


Base de dados utilizada:
https://www.kaggle.com/dataset/e626783d4672f182e7870b1bbe75fae66bdfb232289da0a61f08c2ceb01cab01

O que é o Colaboratory? https://colab.research.google.com/notebooks/intro.ipynb#

Removendo strings de um dataframe:
https://stackoverflow.com/questions/42335385/delete-every-column-that-contains-a-string-in-dataframe

Modelo do RandomForest do Scikit-learn:
https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html

Leitura de arquivos Xlsx com pandas:
https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.read_excel.html

Verificando relevância das colunas:
https://towardsdatascience.com/an-implementation-and-explanation-of-the-random-forest-in-python-77bf308a9b76

https://scikit-learn.org/stable/developers/develop.html#:~:text=The%20fit()%20method%20takes,reference%20to%20X%20and%20y.

https://scikit-learn.org/stable/developers/develop.html#:~:text=The%20fit()%20method%20takes,reference%20to%20X%20and%20y.

https://scikit-learn.org/stable/modules/generated/sklearn.metrics.classification_report.html
https://www.kaggle.com/dataset/e626783d4672f182e7870b1bbe75fae66bdfb232289da0a61f08c2ceb01cab01
https://stackoverflow.com/questions/26886653/pandas-create-new-column-based-on-values-from-other-columns-apply-a-function-o