# Relatório do Projeto de Ciência de Dados para a Disciplina de Proficiência

Este arquivo contém informações sobre o projeto que está sendo executado (nesse caso, traçar o perfil epidemiológico de indivíduos afetados pela Síndrome Respiratória Aguda, utilizando machine learning). Ele é organizado de acordo com o Processo da Team Data Science (TDSP) [Data Science Lifecycle](https://github.com/Azure/Microsoft-TDSP/blob/master/Docs/lifecycle-detail.md).

# ## 1. Entendimento do Problema

## ### Definição do Problema

O objetivo do projeto é verificar se é possível traçar o perfil epidemiológico para identificar o quadro geral de saúde da população afetada pela Síndrome Respiratória Aguda (SAR) utilizando ferramentas de machine learning.

O dataset para este projeto se encontra no site do Ministério da Saúde [link](http://plataforma.saude.gov.br/coronavirus/dados-abertos/). Foram utilizados dados dos anos de 2019 e 2020 (atualizados até 04/05/2020) referentes a Síndrome Respiratória Aguda de aproximadamente 145 mil indivíduos. Baseado nas XXX e XXX features, o objetivo da utilização de machine learning é predizer XXXXXXXXX.

[comment]: # (Verificar os números do notebook)

Mais informações sobre os dados baixados e tratados podem ser encontrados no repositório deste projeto no [GitHub](https://github.com/tiagoacc/PosDS-sars)

## ### Escopo

\* Algoritmos baseados em árvores fornecem modelos preditivos confiáveis, estáveis e de fácil interpretação. Eles conseguem mapear bem as relações não-lineares e podem ser adaptados para resolver problema de classificação que será o nosso caso.

\* Utilizamos da ferramenta Pentaho com objetivo de tratar os dados, realizar a normalização das features.

[comment]: # (Reescrever a utilização do pentaho, mais detalhes)

\* Utilizamos da linguagem de programação Python juntamente com as bibliotecas Pandas para exploração dos dados e a biblioteca scikit-learn para teste e validação dos modelos de machine learning.

[comment]: # (Expandir este tópico)

## ### Planejamento

Seguimos os estágios do ciclo de vida do TDSP, a documentação sobre o trabalho e as descobertas de cada um dos estágios do ciclo de vida está incluída abaixo.

## ### Equipe

Este projeto foi executado pelos alunos:

DAVID COSTA TABOSA - POS19100198

IGOR KONOVALOFF LACERDA - POS19100100

TIAGO SOUSA DE ALENCAR - POS19100438

ALEXANDRE AUGUSTO DESTRO DE CAMPOS - POS19100459

## ### Métricas

O desempenho dos modelos de aprendizado de máquina será avaliado realizando a validação cruzada para estimar a habilidade de predição de cada modelo, serão avaliados os modelos Árvore de Decisão, Gradient Boosting e Random Forest.

O modelo que apresentar o melhor resultado será considerado aceitável e adequado para implantação.

# ## 2. Aquisição e Entendimento dos Dados

## ### Dados Brutos

Para informações mais detalhadas sobre os dados utilizados, favor acessar o[Dicionário de Dados](http://plataforma.saude.gov.br/coronavirus/dados-abertos/sivep-gripe/dicionario-dados-SRAG-hospitalizado-Sivepgripe.pdf), providenciado pelo Ministério da Saúde na sua plataforma de dados abertos.

Existe um total de 148.085 registros, antes que qualquer filtro.

TARGET: ???

[comment]: # (Confirmar)

FEATURES: Sinais e Sintomas, Fatores de Risco, Vacinação, Uso de Antiviral, Agente Etiológico, etc.

[comment]: # (Verificar necessidade de informar outras features)

## ### Exploração de dados utilizando Pentaho e Python

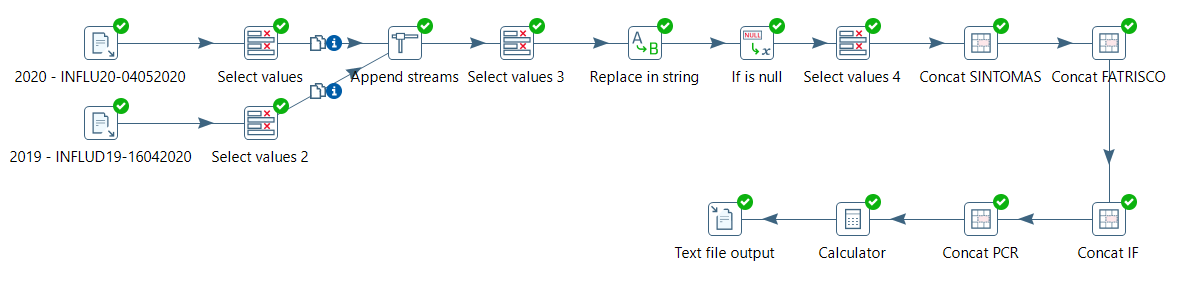
[comment]: # (Apenas informar as ferramentas usadas, colocar link para o arquivo de transformações do pentaho e o jupyter notebook)

# ## 3. Modeling

## ### Feature Engineering

### #### Limpeza dos dados

Utilizando o pentaho realizamos a normalização dos dados e criação das features utilizadas no modelo, conforme a figura abaixo e explicação das etapas aplicadas.



Na primeira etapa realizamos a leitura dos arquivos CSV dos anos de 2019 e 2020 e por terem um pequena diferença nas colunas, fizemos a leitura separadamente para somente após aplicação da segunda etapa de seleção das colunas escolhidas, juntarmos para formar o dataset na terceira etapa.

Foi realizado a etapa Replace in String e If is Null para substituir os valores originais do dataset para o padrão que escolhemos de 0 (não) e 1 (sim).

Em seguida aplicamos o concatenar para agrupar os Sintomas, Fatores de Risco (comorbidades), resultados dos exames IF (Imunofluorescência) e exames PCR, mas sem excluir as colunas originais. O objetivo foi testar se utilizando uma coluna com todos os sintomas agrupados em 0 e 1 (One Hot Encoding) seria melhor.

Por fim, geramos um novo arquivo csv, com os dados de 2019 e 2020, que foi levado para o Jupyter Notebook para uma nova análise e ajustes no dataset.

Após aplicação da análise exploratória dos dados, decidimos por excluir algumas colunas, remover os registros SRAG com situação de outro agente Etiológico ('CLASSI\_FIN'), remover os registros com valor 'Ignorado' para Sexo ('CS\_SEXO') e remoção dos registros com valor 'Ignorado' para Evolução ('EVOLUCAO').

### #### Padronização das features numéricas

[comment]: # (A Fazer)

### #### Saving processed data sets for modeling input

Os conjuntos de dados de treinamento e teste foram selecionados e salvos como arquivos .pkl para entrada na modelagem (dados de treinamento) e avaliação ou implantação do modelo (dados de teste).

[comment]: # (ver onde foram salvo sos arquivo, não está no github)

## ### Treinamento do Modelo

Após o tratamento dos dados, a base foi separada em conjuntos de treino e teste na proporção de 80/20, sendo 80% para treino (56254) e 20% para validação (14064).

Balanceamento das classes

Separação dos atributos (features) da classe

??? (neste tópico ou inicio do outro? ) Em seguida, foi executada a validação cruzada para estimar a habilidade de predição de cada modelo, foram avaliados os modelos Árvore de Decisão, Gradient Boosting e Random Forest.

## ### Avaliação do Modelo

Foi executada a validação cruzada para estimar a habilidade de predição de cada modelo, foram avaliados os modelos Árvore de Decisão, Gradient Boosting e Random Forest.

O modelo que obteve o melhor score dos dados de validação foi o Random Forest.

[comment]: # (Tirar print da avaliação, ou deixar apenas texto mono mesmo, vamos colocar a validação dos outros modelos testados?)

```bash

--------------------------------

SCORE DOS DADOS DE VALIDAÇÃO , MODELO : Random Forest Classifier PASSO: 0

--------------------------------

acurácia : 0.7591723549488054

roc : 0.7196570809732359

f1-score : 0.8311649469119187

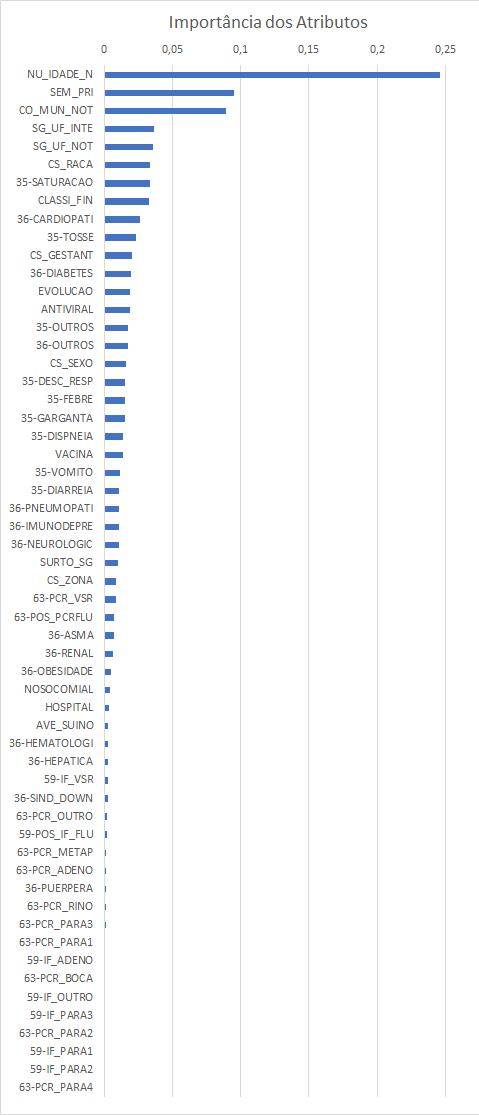
recall score : 0.8024061597690086

precision score : 0.862061834350119

```

[comment]: # (Justificativa porque da escolha do Random Forest - informar números)

A importância dos atributos do modelo Random Forest é mostrada a seguir:



# ## 4. Deployment

[comment]: # (A fazer)

## ### Architecture, Environments, Code Execution

### #### Development

[comment]: # (A fazer)

# ## Referências

Project Repository in GitHub

TDSP project template for Azure Machine Learning