UNIVERSIDADE FEDERAL DE SANTA CATARINA
CENTRO TECNOLÓGICO
DEPARTAMENTO DE INFORMÁTICA E ESTATÍSTICA

PREDIÇÃO DE NOVOS CASOS DE COVID-19

Sadi Júnior Domingos Jacinto

Professor orientador: Jônata Tyska Carvalho

Florianópolis
2021

Conteúdo

Lista de Tabelas

Lista de Figuras

1 Entendimento do Negócio

Há mais de um ano o mundo todo sofreu uma brusca mudança de paradigma, com conceitos como isolamento e distanciamento social agora fazendo parte intrínseca do dia-a-dia da população. O presente trabalho busca analisar os dados advindos de um *dataset* mantido pela organização *Our World in Data*, constituindo uma visão bem completa com diversas variáveis. O resultado será respostas que podem auxiliar a compreensão das características que contribuíram para uma das mais marcantes pandemias dos últimos 100 anos, comparável à gripe asiática.

1.1 Objetivos do Negócio

- Predizer o número de novos casos com base nos dados já existentes.
- Avaliar a predita curva de evolução do vírus para os países do *dataset*.

1.1.1 Critérios de Sucesso do Negócio

- Informações que ajudem o observador a perceber características que contribuem para a taxa de infecção do vírus, possibilitando inversamente determinar fatores que a reduzem.
- Um esquema que dê uma ideia do avanço do vírus durante o tempo, e dê uma projeção para determinar o estado do mundo no futuro.

1.2 Avaliar Situação

1.2.1 Inventário de Recursos

- Recursos Humanos: equipe inicialmente composta de 4 estudantes de *Data Mining*, atualmente reduzida para apenas 1 estudante.
- Recursos de Dados: *dataset* mantido pela organização *Our World in Data* (Owid), acessível através do *link* https://covid.ourworldindata.org/data/owid-covid-data.csv.

1.2.2 Requisitos, Premissas e Restrições

- O projeto terá duração de 12 semanas, iniciando dia 29/06/2021 e com data de entrega final dia 14/09/2021. Será realizada uma entrega intermediária dia 23/07/2021, referente às primeiras 3 etapas da CRISP-DM (entendimento do negócio, entendimento dos dados e preparação dos dados);
- Os dados deverão estar disponíveis e acessíveis durante a realização do projeto;
- As atividades serão realizadas na linguagem *Python* e será utilizado *Jupyter Notebook*, juntamento com o uso das bibliotecas *pandas*, *matplotlib*, *numpy*, *seaborn*, entre outras ferramentas que serão identificadas no arquivo *requiriments.txt* em anexo a esse trabalho.

1.2.3 Riscos e Contingências

	Descrição	**Probabilidade**	**Impacto**	**Fator de Exposição**	**Ações de Prevenção**	**Plano de Contingência**
R1	Dados não adequados ou insuficientes para treinamento	Improvável	Crítico	Médio (3)	Separar alguns *subsets* do *dataset* principal para uso em testes	Buscar por amostras diferentes da fonte, em último caso pensar em um modelo diferente
R2	Desistência de um ou mais integrantes do grupo da matéria ou trabalho	Muito Provável	Crítico	Alto (5)	Promover um ambiente de trabalho relativamente confortável	Tentar compensar com mais esforço pelos integrantes restantes
R3	Perda dos documentos e códigos desenvolvidos	Improvável	Catastrófico	Médio(4)	Utilizar repositórios e armazenamento em nuvem, fazer cópias ou compartilhamento dos artefatos desenvolvidos	Acesso aos *backups* ou versionamentos e recuperação dos arquivos perdidos
R4	Integrantes sem disponibilidade ou conhecimento das ferramentas suficiente para realizar entregas no prazo estipulado	Provável	Crítico	Médio(4)	Organizar cronograma fixo, separar tarefas de acordo com as habilidades individuais, rever aulas e materiais externos	Nivelamento de conhecimento através do compartilhamento de conhecimento dos outros integrantes
R5	Modelo formulado não atende os objetivos e especificações do projeto	Provável	Crítico	Médio(3)	Fazer um bom foco na etapa de entendimento dos dados para garantir que a decisão e formulação de modelo encaixe com os dados e objetivos	Pensar em um modelo diferente

Tabela 1: Tabela de Riscos de Contigências do Projeto

Impacto	Probabilidade			
		Muito Provável(2)	Provável (1)	Improvável (0)
	Catastrófico (4)	Alto (6)	Alto (5)	Médio (4)
	Crítico (3)	Alto (5)	Médio (4)	Médio (3)
	Marginal (2)	Médio (4)	Médio (3)	Baixo (2)
	Negligenciável (1)	Médio (3)	Baixo (2)	Baixo (1)

Tabela 2: Tabela de Cálculo do Fator de Exposição

1.2.4 Custos e Benefícios

- Custos:

  1. Tempo e mão-de-obra que será despendido pelos participantes no decorrer do projeto.

- Benefícios:

  1. Aplicação prático do que foi visto em sala de aula.
  2. Um estudo completo em uma base de dados extensa sobre Covid, que é um tema atual e presente nos tempos atuais.
  3. Possibilidade de utilizar o presente trabalho como base para trabalhos futuros.
  4. Incrementar o portfólio dos estudantes envolvidos.

1.3 Determinar Objetivos do Projeto

1.3.1 Objetivos do Projeto

- Ter uma visualização compreensiva do progresso (positivo e negativo) da reação dos países à pandemia.
- Predizer o número de novos casos de Covid-19.
- Entender as características mais impactantes no surgimento de novos casos de contágio por Covid-19.

1.3.2 Critérios de Sucesso do Projeto

- Predizer novas instâncias com:
  - $R_a^2 \geq 0.6$
  - $MSE \leq 50$
  - $RMSE \leq 5$
  - $MAE \leq 5$

1.4 Formalização do Projeto

1.4.1 Identificação do Projeto

- **Nome do Projeto:** PREDIÇÃO DE NOVOS CASOS DE COVID-19
- **Data de Início:** 29/06/2021
- **Data de Término:** 14/09/2021

1.4.2 Modelo CRISP-DM

Figura 1: Fluxo usado no CRISP-DM

Neste trabalho foi seguido o modelo CRISP-DM, que apresenta uma visão geral do ciclo do projeto de mineração de dados, incluindo fases e tarefas relacionadas ao projeto. O ciclo de vida do método consiste em seis fases que não precisam ser seguidas à risca e adota uma forma contínua, interativa e dinâmica de fornecer *feedback* às fases de descoberta de novas informações ou melhorias de processos.

2 Relatório da Coleta de Dados iniciais

Os dados utilizados neste estudo são coletados do *COVID-19 Dataset by Our World in Data*, que por sua vez é formado por dados referentes a Covid de diversas fontes confiáveis de vários países do mundo, utilizando fontes governamentais de ministério e veículos de comunicação respeitáveis. O conjunto de dados completo era composto inicialmente de 116103 registros, distribuídos em 60 colunas, existentes em um arquivo CSV chamado *owid-covid-data.csv*. Cada linha do *dataset* corresponde a um dia de registro, sendo esta a menor granularidade a ser avaliada.

O processamento dos dados foi realizado utilizando o *Jupyter Notebook*, com o uso das bibliotecas:

- pandas;
- numpy;
- matplot;
- seaborn;
- pycountry_convert;
- sklearn;
- iso3166;
- datetime e
- missingno.

2.1 Relatório da Descrição dos Dados

Coluna	Descrição
iso_code	Código do país (e.g. BRA)
continent	Continente
location	Nome do país (e.g Brasil)
date	Data
total_cases	Total de casos
new_cases	Novos casos
new_cases_smoothed	Novos casos com achatamento
total_deaths	Total de mortes
new_deaths	Novas mortes
new_deaths_smoothed	Novas mortes com achatamento
total_cases_per_million	Total de casos por milhão
new_cases_per_million	Novos casos por milhão
new_cases_smoothed_per_million	Novos casos por milhão com achatamento
total_deaths_per_million	Total de mortes por milhão
new_deaths_per_million	Novas mortes por milhão
new_deaths_smoothed_per_million	Novas mortes por milhão com achatamento
reproduction_rate	Taxa de reprodução do vírus

icu_patients	Quantidade de pacientes na UTI
icu_patients_per_million	Quantidade de pacientes na UTI por milhão
hosp_patients	Pacientes em hospital
hosp_patients_per_million	Pacientes em hospital por milhão
weekly_icu_admissions	Admissões semanais nas UTIs
weekly_icu_admissions_per_million	Admissões semanais nas UTIs por milhão
weekly_hosp_admissions	Admissões semanais em hospital
weekly_hosp_admissions_per_million	Admissões semanais em hospital por milhão
total_tests	Total de testes
new_tests	Novos testes
total_tests_per_thousand	Total de testes por milhares
new_tests_per_thousand	Novos testes por milhares
new_tests_smoothed	Novos testes com achatamento
new_tests_smoothed_per_thousand	Novos testes com achatamento por milhares
positive_rate	Taxa de resultados positivos
tests_per_case	Testes por caso
tests_units	Unidades de teste
total_vaccinations	Total de vacinações
people_vaccinated	Pessoas vacinadas
people_fully_vaccinated	Pessoas completamente vacinadas
new_vaccinations	Novas vacinações
new_vaccinations_smoothed	Novas vacinações com achatamento
total_vaccinations_per_hundred	Total de vacinações por centenas
people_vaccinated_per_hundred	Pessoas vacinadas por centenas
people_fully_vaccinated_per_hundred	Pessoas completamente vacinadas por centenas
new_vaccinations_smoothed_per_million	Novas vacinações com achatamento por milhão
stringency_index	Severidade da reação dos países (0-100)
population	População
population_density	Densidade populacional
median_age	Idade mediana
aged_65_older	Cidadãos com idade superior a 65
aged_70_older	Cidadãos com idade superior a 70
gdp_per_capita	PIB per capita
extreme_poverty	Parcela da população em Pobreza extrema
cardiovasc_death_rate	Taxa de morte cardiovascular
diabetes_prevalence	Prevalência de Diabetes
female_smokers	Mulheres fumantes
male_smokers	Homens fumantes
handwashing_facilities	Estabelecimentos com capacidade de lavar mão
hospital_beds_per_thousand	Leitos hospitalares por milhares
life_expectancy	Expectativa de vida
human_development_index	IDH
excess_mortality	Excesso de mortalidade

Tabela 3: Atributos do *dataset*

```
data.describe()
```

	total_cases	new_cases	new_cases_smoothed	total_deaths	new_deaths	new_deaths_smoothed	total_cases_per_million	new_cases_per_million
count	1.104690e+05	110466.000000	109451.000000	9.983700e+04	99992.000000	109451.000000	109886.000000	109883.000000
mean	1.342936e+06	6415.981306	6426.560982	3.456749e+04	146.096708	132.677122	16549.697928	81.224753
std	9.029340e+06	39314.225505	38848.838328	2.061684e+05	796.926341	744.054864	28716.511639	189.209107
min	1.000000e+00	-74347.000000	-6223.000000	1.000000e+00	-1918.000000	-232.143000	0.001000	-3125.829000
25%	1.807000e+03	3.000000	8.714000	6.400000e+01	0.000000	0.000000	319.281250	0.271000
50%	1.842900e+04	88.000000	109.143000	5.550000e+02	2.000000	1.571000	2458.610000	10.140000
75%	1.968120e+05	900.000000	948.642500	4.805000e+03	18.000000	15.429000	19155.119500	78.685500
max	2.242699e+08	905932.000000	826340.429000	4.624864e+06	17976.000000	14722.714000	208199.373000	8620.690000

Figura 2: Descrição dos dados com Pandas

2.2 Relatório da Exploração e Qualidade dos Dados

O gráfico a seguir demonstra a quantidade de valores faltantes:

```
import missingno as msno
msno.matrix(data, figsize=(15, 5))
```

```
<AxesSubplot:>
```

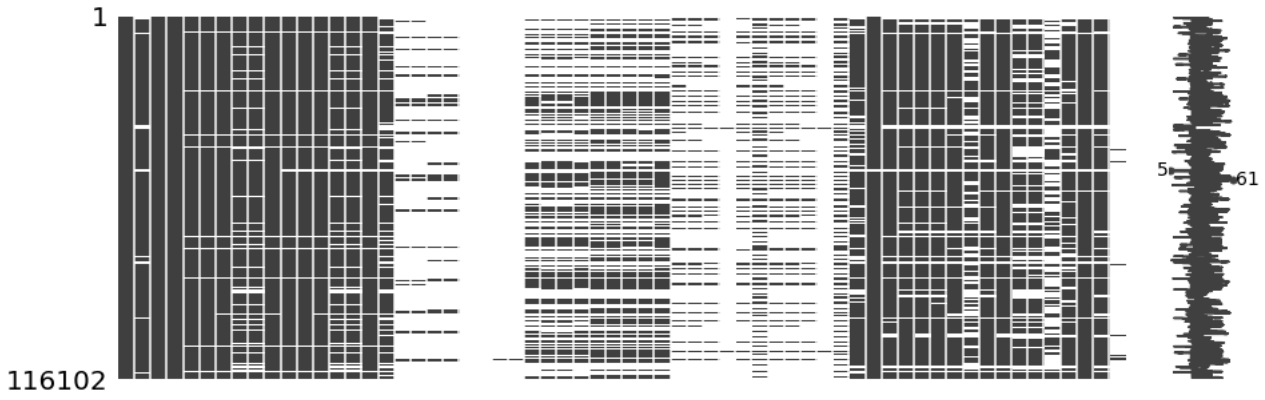

Figura 3: Matriz de valores faltantes

Onde, quanto mais clara a célula, mais valores faltantes existem. Porém, por haverem muitos atributos e linhas no *dataset*, esse tipo de informação pode ser mais facilmente visualizada com a imagem abaixo:

```
data.isna().sum()
```

```
iso_code                                0
continent                            5317
location                                0
date                                    0
total_cases                          5633
new_cases                            5636
new_cases_smoothed                   6651
total_deaths                        16265
new_deaths                          16110
new_deaths_smoothed                  6651
total_cases_per_million              6216
new_cases_per_million                6219
new_cases_smoothed_per_million       7229
total_deaths_per_million            16835
new_deaths_per_million              16680
new_deaths_smoothed_per_million      7229
reproduction_rate                   22778
icu_patients                       103482
icu_patients_per_million           103482
hosp_patients                      101016
hosp_patients_per_million          101016
weekly_icu_admissions              114960
weekly_icu_admissions_per_million  114960
weekly_hosp_admissions             114076
weekly_hosp_admissions_per_million 114076
new_tests                           66059
total_tests                         66296
total_tests_per_thousand            66296
new_tests_per_thousand              66059
new_tests_smoothed                  57064
new_tests_smoothed_per_thousand     57064
positive_rate                       60298
tests_per_case                      60916
tests_units                         55203
total_vaccinations                  91087
people_vaccinated                   92129
people_fully_vaccinated             95085
total_boosters                     114381
new_vaccinations                    95395
new_vaccinations_smoothed           71485
total_vaccinations_per_hundred      91087
people_vaccinated_per_hundred       92129
people_fully_vaccinated_per_hundred 95085
total_boosters_per_hundred         114381
new_vaccinations_smoothed_per_million 71485
stringency_index                    20550
population                            797
population_density                   8733
median_age                          13419
aged_65_older                       14563
aged_70_older                       13983
gdp_per_capita                      12856
extreme_poverty                     46732
cardiovasc_death_rate               13110
diabetes_prevalence                 10027
female_smokers                      35667
male_smokers                        36837
handwashing_facilities              64366
hospital_beds_per_thousand          22434
life_expectancy                      5970
human_development_index             13019
excess_mortality                   112099
dtype: int64
```

Figura 4: Quantidade de valores faltantes por coluna

Em seguida, foi verificada a quantidade de valores duplicados:

```
data.duplicated().sum()
```

```
0
```

Figura 5: Quantidade total de valores duplicados

E, felizmente, não existem registros duplicados. Em seguida, foi verificado quais eram os tipos de dados, através da função “dtypes” aliada à uma verificação manual do arquivo CSV e com o uso do conhecimento humano dos membros da equipe. O que gerou as conclusões:

```
data.dtypes
```

iso_code	object
continent	object
location	object
date	object
total_cases	float64
new_cases	float64
new_cases_smoothed	float64
total_deaths	float64
new_deaths	float64
new_deaths_smoothed	float64
total_cases_per_million	float64
new_cases_per_million	float64
new_cases_smoothed_per_million	float64
total_deaths_per_million	float64
new_deaths_per_million	float64
new_deaths_smoothed_per_million	float64
reproduction_rate	float64
icu_patients	float64
icu_patients_per_million	float64
hosp_patients	float64
hosp_patients_per_million	float64
weekly_icu_admissions	float64
weekly_icu_admissions_per_million	float64
weekly_hosp_admissions	float64
weekly_hosp_admissions_per_million	float64
new_tests	float64
total_tests	float64
total_tests_per_thousand	float64
new_tests_per_thousand	float64
new_tests_smoothed	float64
new_tests_smoothed_per_thousand	float64
positive_rate	float64
tests_per_case	float64
tests_units	object
total_vaccinations	float64
people_vaccinated	float64
people_fully_vaccinated	float64
total_boosters	float64
new_vaccinations	float64
new_vaccinations_smoothed	float64
total_vaccinations_per_hundred	float64
people_vaccinated_per_hundred	float64
people_fully_vaccinated_per_hundred	float64
total_boosters_per_hundred	float64
new_vaccinations_smoothed_per_million	float64
stringency_index	float64
population	float64
population_density	float64
median_age	float64
aged_65_older	float64
aged_70_older	float64
gdp_per_capita	float64
extreme_poverty	float64
cardiovasc_death_rate	float64
diabetes_prevalence	float64
female_smokers	float64
male_smokers	float64
handwashing_facilities	float64
hospital_beds_per_thousand	float64
life_expectancy	float64
human_development_index	float64
excess_mortality	float64

dtype: object

Figura 6: Tipos dos dados

1. O atributo *date*, é uma data no formato *yyyy-mm-dd*, mas está sendo interpretado pelo *pandas* como um *object* ou, mais precisamente, uma *str*. Logo de cara encontramos uma necessidade de transformação de tipos.

2. Existe uma quantidade muito grande de valores *NaN*, que devem ser tratados, seja por inferência ou remoção.

3. Existem diversos valores numéricos, como *population*, que deveriam ser inteiros mas estão sendo interpretados como *float*.

Finalmente, foram realizadas algumas análises estatísticas sobre os dados, como as contidas nas figuras abaixo, mas, por serem de difícil inclusão nesse relatório, as mesmas foram omitidas, e uma descrição mais detalhada das mesmas foi adicionada no *notebook*.

```
plt.figure(figsize=(12, 9))
sns.heatmap(data.corr(), annot=False)
```

```
<AxesSubplot:>
```

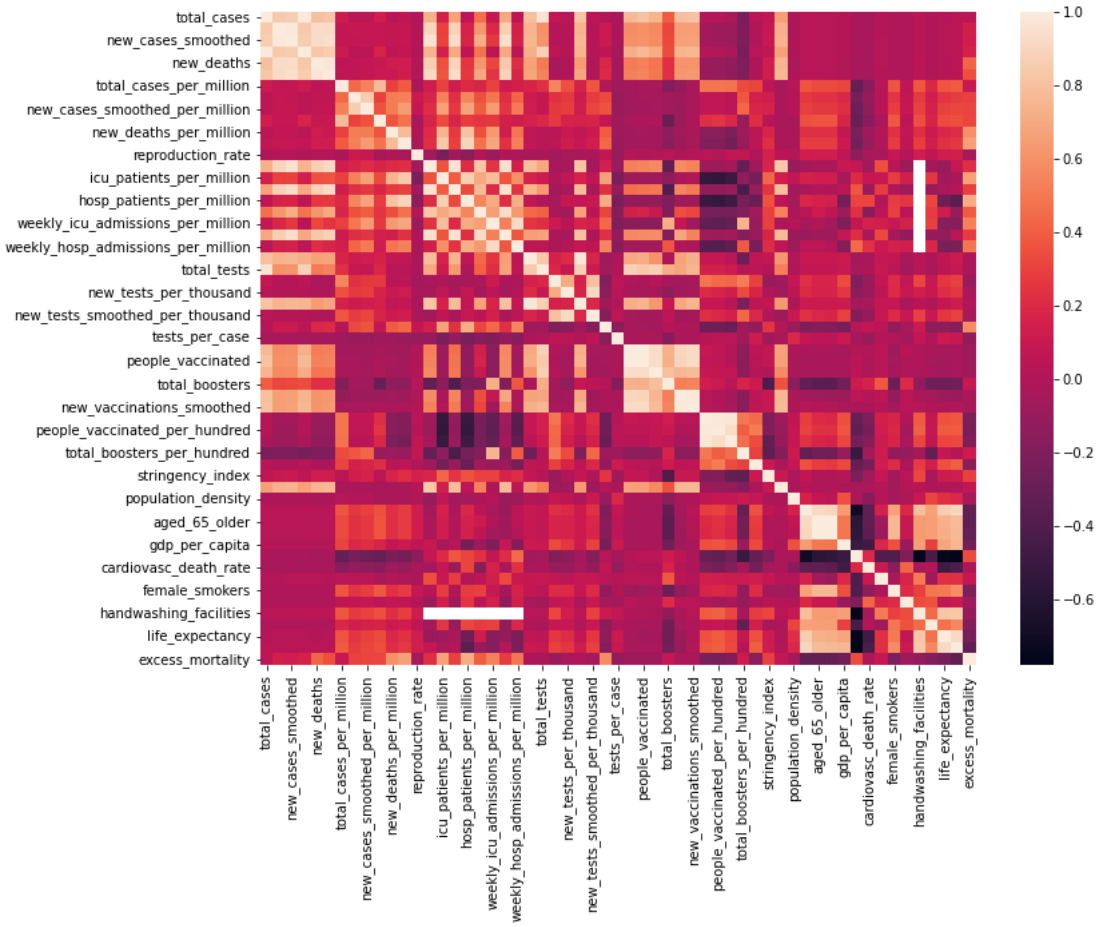

Figura 7: Correlação entre os atributos

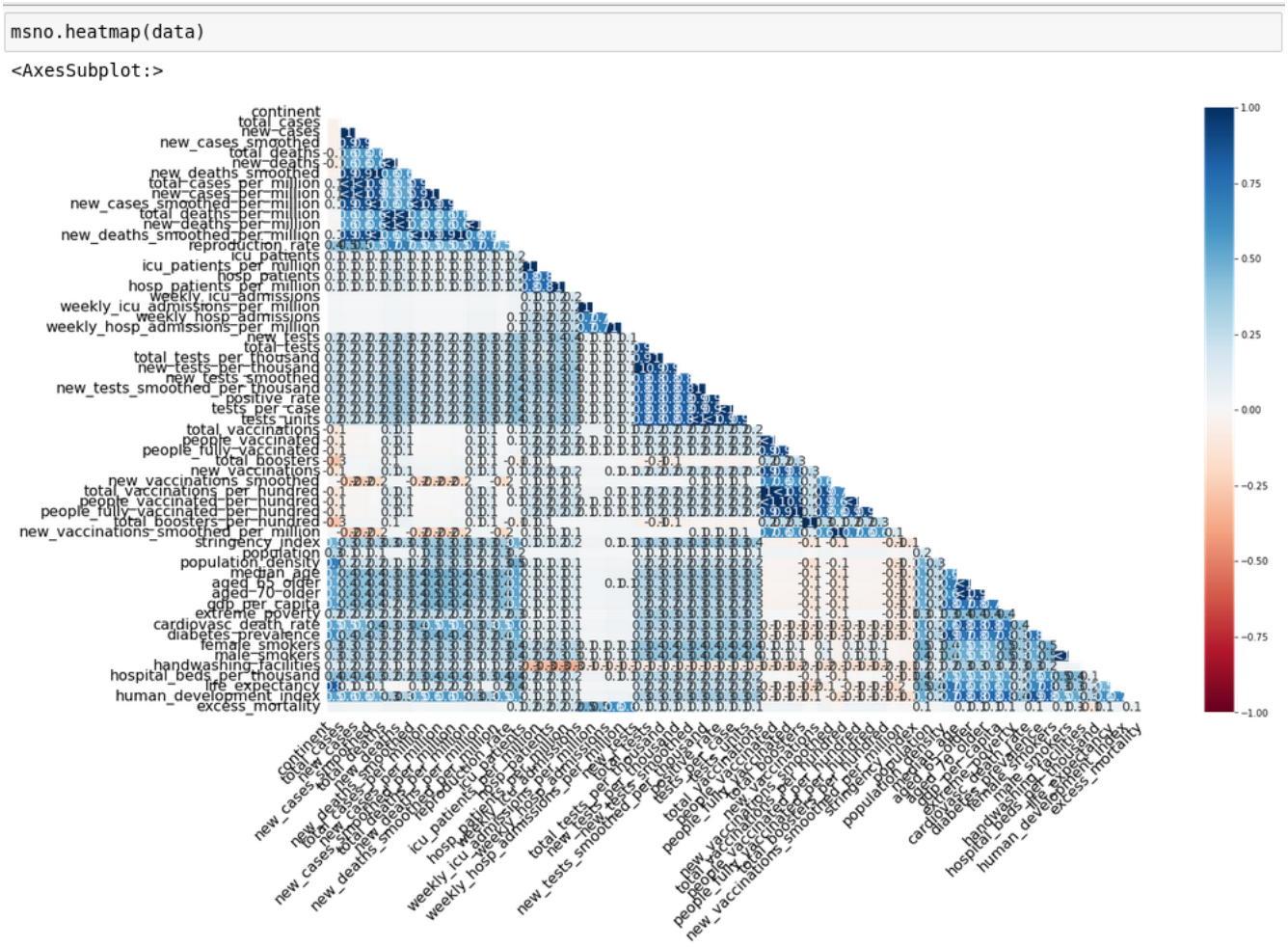

Figura 8: Correlação de atributos nulos

3 Preparação dos Dados

3.1 Lista de inclusões/exclusões

Foram inicialmente removidos os atributos:

- cardiovasc_death_rate;
- diabetes_prevalence;
- excess_mortality;
- icu_patients;
- icu_patients_per_million;
- population_density;
- stringency_index;
- tests_per_case;
- tests_units;
- weekly_hosp_admissions;
- weekly_host_admissions_per_million;
- weekly_icu_admissions e
- weekly_icu_admissions_per_million.

Esta remoção foi feita principalmente por não haver necessidade no uso dos mesmos. Além disso, uma grande porção dos valores destas colunas se encontravam nulos o que dificulta o processo de inferência. Dessa forma, restaram 47 dos 60 atributos originais.

3.2 Relatório de Limpeza dos Dados

Foram feitos os seguintes procedimentos:

- Verificação, correção e, se necessário, remoção de inconsistências entre os atributos *iso_code*, *continent* e *location*. Os dados que possuíam valores inconsistentes (como país sendo Oceano Atlântico), foram removidos.
- Conversão inicial do atributo *date* para *datetime*.
- Remoção de linhas que possuíam todos os dados nulos.
- Remoção de linhas que possuíam 39 ou mais atributos nulos.
- Tratamento de valores nulos, melhor descrito na seção 3.2.2.
- Remoção dos registros que não foram possíveis de serem tratados.
- Discretização das variáveis categóricas utilizando *LabelEncoder*[1]

[1] O uso de *OneHotEncode* foi cogitado e testado, mas o *dataset* ficou muito grande, resultando no processamento acabar "travando".

- Remoção de atributos que possuíam um valor de correlação inferior à *0.3* em relação a variável-alvo *new_ cases*.
- Remoção de linhas que possuíam algum valor numérico negativo.

3.2.1 Atributos Derivados

De acordo com a análise inicial, não serão utilizados, a princípio, atributos derivados, em grande parte pela abundância de linhas e atributos já existentes no *dataset*.

3.2.2 Registros Gerados

O preenchimento de valores nulos (*NaN*) no *dataset* consistiu na aplicação do seguinte método:

1. Encontra-se um atributo nulo.
2. É pego qual é o país e data (date) desse registro.
3. Calcula-se o dia inicial e final do mês e ano da data anteriormente selecionada.
4. Calcula-se a média daquele atributo nulo, no continente e mês anteriormente selecionados.
5. Caso a ela seja diferente de nulo ou maior que zero, a mesma é utilizada como inferência para o atributo e o processo passa para o próximo atributo nulo.
6. Caso a média tenha valor nulo ou igual ou inferior à zero, a mesma é recalculada usando a média do continente inteiro, sem limitação de data, para aquele atributo, e volta-se para o passo 5.
7. Caso a média continue nula ou igual ou inferior à zero, o valor daquele atributo é considerado como vazio, ou *NaN* (*not a number*), e limpo posteriormente.

3.3 Junção de Dados de Diversas Fontes

Não houve a necessidade de recorrer a dados de outras fontes.

3.4 Dados Formatados

Após a preparação dos dados, foi criado o arquivo *process.csv*, com os dados prontos para as próximas etapas do CRISP-DM. Vale ressaltar que esse *dataset* com os dados processados ainda não foi dividido em dois outros *datasets* (treino e teste).

A quantidade de registros ao final do processo foi de 13506 registros, cada um deles com 17 atributos, nos quais não existem dados nulos ou duplicados.

Maiores dúvidas sobre como os dados foram processados e o por quê, podem ser encontrados no *Notebook* utilizado, o qual contém várias células do tipo *Markdown* que descrevem, da forma mais detalhada possível, o que foi feito.

4 Modelagem

Dados os fatos que a variável-alvo (*new_ cases*) consiste em um valor contínuo, e a proposta desse projeto é de a prever, os modelos usados serão:

- Regressão Linear:
  Aprendizado de máquina supervisionado, usado para predizer valores contínuos, através de modelos lineares: $y = ax + b$, onde y é chamado de condicional ou dependente, enquanto x é chamado de independente. Consiste em encontrar os parâmetros de uma reta que melhor se adequa ao conjunto de dados.

- *Random Forest*:
  Algoritmo de aprendizado supervisionado, funciona construindo várias árvores de decisão durante o treinamento. Para tarefas de classificação, o *output* é a classe selecionada pela maioria das árvores. Para tarefas de regressão, que é o que interessa para o presente trabalho, é calculado a média de todas as árvores. Muito útil para corrigir o *overfitting* comum às árvores de decisão.

- KNN:
  Algoritmo que pode ser utilizado para regressão e classificação. Busca classificar/predizer novas instâncias baseado na similaridade dessa nova instância com os dados já existentes. É calculada a distância entre as instâncias, geralmente usando a distância euclidiana:

  $$d(x_i, x_j) \equiv \sqrt{\sum_{r=1}^{n} (a_r(x_i) - a_r(x_j))^2}$$

  Se houverem muitos atributos, o processo de classificação se torna lento. Além disso, atributos irrelevantes podem alterar o resultado.

Além disso, alguns hiperparâmetros, como tamanho do conjunto de testes, foram atribuídos de forma dinâmica em um *loop*, assim como foram testados três tipos de *datasets*:

1. *Dataset* gerado pelas três primeiras etapas do CRISP-DM.
2. *Dataset* com os atributos redimensionados utilizando *StandardScaler*.
3. *Dataset* com os atributos redimensionados utilizando *MinMaxScaler*.

feito o dimensionamento dos atributos utilizando

5 Avaliação

Foram utilizadas quatro métricas de avaliação, a saber:

- R_a^2 ou Coeficiente de Determinação Ajustado:
  Medida estatística de quão próximos os dados estão da linha de regressão ajustada. Varia entre 0 e 1, por vezes sendo expresso em termos percentuais. Nesse caso, expressa a quantidade da variância dos dados que é explicada pelo modelo linear. Assim, quanto maior o R_a^2, mais explicativo é o modelo linear, ou seja, melhor ele se ajusta à amostra.

  Por exemplo, um $R_a^2 = 0{,}8234$ significa que o modelo linear explica 82,34% da variância da variável dependente a partir das variáveis independentes incluídas naquele modelo linear.

  Diferencia da métrica R^2 pela fato de penalizar (reduzir) o valor caso uma *feature*presente não contribua significativamente para o modelo.

$$R_a^2 = 1 - \frac{(1 - R^2)(N - 1)}{N - p - 1}$$

- Erro Quadrático Médio (MSE):
  Consiste na média do erro das previsões ao quadrado. Em outras palavras, pega-se a diferença entre o valor predito pelo modelo e o valor real, eleva-se o resultado ao quadrado, faz-se a mesma coisa com todos os outros pontos, soma-os, e dividi-se pelo número de elementos preditos. Quanto maior esse número, pior o modelo.

  Essa métrica apresenta valor mínimo 0, sem valor máximo, e, uma vez que essa métrica eleva o erro ao quadrado, predições muito distantes do real aumentam o valor da medida muito facilmente, o que a torna uma métrica de avaliação excelente para problemas nos quais grandes erros não são tolerados.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(\widehat{y}_i - y_i)^2$$

- Raiz do erro quadrático médio (RMSE):
  Igual a do MSE, mas utiliza a raiz para melhorar a interpretabilidade da métrica.

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\widehat{y}_i - y_i)^2}$$

- Erro Absoluto Médio (MAE):
  Média das distâncias entre valores preditos e reais. Diferentemente do MSE e do RMSE, essa métrica não "pune" tão severamente os *outliers* do modelo. Essa medida apresenta valor mínimo 0 e não apresenta valor máximo.

  Pelo fato de não elevar as diferenças ao quadrado, essa medida torna-se uma opção não tão ideal para lidar com problemas delicados. Contudo, é uma métrica sólida para modelos que devem prever muitos dados ou dados sazonais, como em previsões de números de casos de doenças, nas quais prever a tendência e sazonalidade dos números é mais importante do que os valores absolutos de cada dia.

$$MAE = \frac{1}{n}\sum_{i=1}^{n} \mid \widehat{y}_i - y_i \mid$$

Todos os resultados obtidos se encontram no arquivo em anexo, chamado *model.pdf*, que foi gerado utilizando a própria ferramenta *Jupyter* para converter o *notebook* em um PDF. Como os modelos foram testados com várias alterações nos hiperparâmetros, o resultado final ficou bastante extenso e, por isso, não estará presente nesse relatório.

Segue exemplo dos melhores resultados obtidos na Regressão Linear como exemplo:

Random State	***Test Size***	***Dataset***	R_a^2	**MAE**	**MSE**	**RMSE**	**Tempo de Execução**
0	0.05	Original	0.9755769659354914	1308.1977440828402	13797996.112094117	3714.565400163809	0:00:32.080545
50	0.5	Standard Scaler	0.956971537907585	0.05815647416128945	0.04429631176335151	0.21046688994554824	0:00:00.015268
35	0.35	MinMax Scaler	0.9013948728493465	0.004154408606323387	0.0003432549550400703	0.01852714103794944	0:00:56.306996

Tabela 4: Regressão Linear

De forma geral, o algoritmo KNN obteve o pior desempenho dos 3, sendo que o, na maioria das vezes, o algoritmo de regressão linear obteve um desempenho superior ao *Random Forest*.

Finalmente, utilizando um *notebook* com 8 CPUs e tendo 8 Gibabytes dedicados ao *Jupyter Server*, a execução do pré-processamento e treinamento e validação dos modelos levou cerca de 3 horas.

6 Considerações Finais

Felizmente foi possível atingir os objetivos do projeto. Além disso, foi um trabalho interessante, que acrescenta muito ao conhecimento dos envolvidos.

Finalmente, uma das implementações que deveria ser feita, caso houvesse mais tempo, seria uma atualização dinâmica dos hiperparâmetros baseada na taxa de erro, além de armazenar os melhores hiperparâmetros para cada *dataset* e modelo.

Referências

[1] 3.3. metrics and scoring: quantifying the quality of predictions. Accessed: 2021-09-12.

[2] A beginner's guide to linear regression in python with scikit-learn. Accessed: 2021-09-12.

[3] Numpy documentation. Accessed: 2021-07-25.

[4] Pandas documentation. Accessed: 2021-07-25.

[5] User guide and tutorial - seaborn 0.11.2. Accessed: 2021-07-25.

[6] User's guide - matplotlib 3.4.3. Accessed: 2021-07-25.

[7] Crisp-dm, Aug 2021. Accessed: 2021-07-25.

[8] Azank, F. Como avaliar seu modelo de regressão, Aug 2020. Accessed: 2021-09-12.

[9] Azevedo, A., and Santos, M. Kdd, semma and crisp-dm: A parallel overview. pp. 182–185.

[10] Chapman, P., Clinton, J., Kerber, R., Khabaza, T., Reinartz, T., Shearer, C., and Wirth, R. Crisp-dm 1.0: Step-by-step data mining guide.

[11] Contributor, D. Deloitte brandvoice: For millennials and gen zs, social issues are top of mind-here's how organizations can drive meaningful change, Jul 2021. Accessed: 2021-07-25.

[12] jaydeemourgjaydeemourg. How to plot predicted values vs the true value?, Feb 1968. Accessed: 2021-09-12.

[13] Koehrsen, W. Random forest in python, Jan 2018. Accessed: 2021-09-12.

[14] Owid. owid/covid-19-data: Data on covid-19 (coronavirus) cases, deaths, hospitalizations, tests - all countries - updated daily by our world in data. Accessed: 2021-07-25.

[15] Ritchie, H., Mathieu, E., Rodés-Guirao, L., Appel, C., Giattino, C., Ortiz-Ospina, E., Hasell, J., Macdonald, B., Beltekian, D., Roser, M., and et al. Coronavirus pandemic (covid-19) - statistics and research, Mar 2020. Accessed: 2021-07-25.

[16] Robinson, S. Linear regression in python with scikit-learn, Jun 2021. Accessed: 2021-09-12.

[17] Zaidi, J. Project: Analyzing suicide clusters using exploratory data analysis & machine learning, Jan 2021. Accessed: 2021-09-12.