



CENTRO UNIVERSITÁRIO CARIOCA

CIÊNCIA DA COMPUTAÇÃO

PEDRO IGOR GRILO DE OLIVEIRA CARVALHO

**MACHINE LEARNING: MODELO DE PREDIÇÃO DE NOVOS CASOS DE
TUBERCULOSE NO BRASIL**

Rio de Janeiro

2020

PEDRO IGOR GRILO DE OLIVEIRA CARVALHO

MACHINE LEARNING: Modelo de Predição de Novos Casos de Tuberculose no Brasil

Trabalho de Conclusão de Curso apresentado ao curso de Ciência da Computação do Centro Universitário Carioca, como requisito parcial para a obtenção do título de Bacharel em Ciência da Computação.

Orientador: Prof. Alberto Tavares da Silva

Rio de Janeiro

2020

C257m Carvalho, Pedro Igor Grilo de Oliveira

Machine learning : modelo de predição de novos casos de tuberculose no Brasil / Pedro Igor Grilo de Oliveira Carvalho.– Rio de Janeiro, 2020.
60 f.

Orientador: Alberto Tavares da Silva
Trabalho de Conclusão de Curso (Graduação em Ciência da Computação) – Centro Universitário UniCarioca, Rio de Janeiro, 2020.

1. Machine Learning. 2. Data Science. 3. Tuberculose.
4. Gestão de saúde. 5. Facebook Prophet. 6. Extração de dados.
7. Visualização de dados. I. Silva, Alberto Tavares da, prof.
II. Título.

CDD 005.1

PEDRO IGOR GRILO DE OLIVEIRA CARVALHO

TEMA: MACHINE LEARNING: Modelo de Predição de Novos Casos de Tuberculose no Brasil

Banca Examinadora

Prof. Alberto Tavares da Silva - Orientador
Centro Universitário Carioca

Prof. André Luiz Avelino Sobral – Coordenador
Centro Universitário Carioca

Prof. Sérgio Assunção Monteiro – Professor Convidado
Centro Universitário Carioca

AGRADECIMENTOS

Agradeço primeiramente minha mãe, Silbene Grilo, e meu irmão Matheus Grilo pelo incentivo de voltar a estudar depois de alguns anos parado e por todo o apoio ao longo dos anos da graduação e na construção deste trabalho. O apoio deles foi fundamental para tomar as melhores decisões e conseguir conciliar a rotina de estudos com a rotina de trabalho e vida pessoal. Agradeço minha noiva Beatriz Silva e minha filha Marcella Silva por me acompanharem ao longo da construção do material e compreenderem os dias e fins de semana dedicados a ele. Para a construção deste trabalho não posso deixar de agradecer aos amigos Gabriel Daiha e Luiz Henrique Santos que contribuíram com seus conhecimentos técnicos e me passaram bastante aprendizado. Agradeço aos professores pela dedicação e conhecimento passado ao longo de todo o curso e especialmente ao professor e orientador Alberto Tavares, pela disposição, atenção e colaboração ao longo de toda a construção do material, desde escolha do tema até o fechamento.

DEDICATÓRIA

Dedico este trabalho a minha falecida esposa, Cristiane Santoro, que infelizmente nos deixou em 2016, no meu primeiro semestre na faculdade, mas que sempre acreditou em mim e tenho certeza que estaria bastante orgulhosa ao acompanhar este trabalho. As palavras dela sempre serviram de motivação para eu conseguir chegar onde cheguei.

RESUMO

Ferramentas de Ciência de Dados e Machine Learning estão sendo amplamente utilizadas na área da saúde para auxiliar os profissionais de diversas formas. Neste trabalho são apresentadas diversas ferramentas que podem ajudar na gestão da saúde e diversas outras áreas. Os processos e ferramentas de Machine Learning são utilizados para prever o número de novos casos de Tuberculose no Brasil, por Estado, com o objetivo de auxiliar a gestão de saúde em todo o país e para que medidas de prevenção possam ser adotadas. Ao longo do trabalho é possível visualizar todas as etapas de um projeto de Machine Learning, desde a extração dos dados, até o tratamento dos dados, a modelagem utilizando a ferramenta Facebook Prophet, a visualização das curvas e dos números, os resultados das previsões e a etapa final de interação com os dados através da ferramenta Microsoft Power BI para que o usuário final possa navegar, compreender os dados e tomar as melhores decisões. O número de casos de Tuberculose no Brasil vem crescendo desde 2016, com um elevado aumento em 2018 e uma tendência de alta para o futuro. É uma doença que segue um padrão de sazonalidade anual em cada Estado, onde é possível visualizar as curvas e entender quais são os meses mais críticos. Existe um possível problema de atraso e subnotificação dos dados por parte de algumas secretarias estaduais de saúde que podem gerar maiores erros pelo modelo, mas no geral, nos principais Estados do Brasil foi possível observar números satisfatórios quando comparados com os dados reais. No modelo foram utilizados dados de 2010 à 2018 e depois os dados previstos de 2019 foram comparados com os dados reais divulgados pelo Ministério da Saúde, através da página do DataSUS. O erro total, em 2019, ficou em 1,42% e ao longo do trabalho será apresentado todos os detalhes.

Palavras-chave: Machine Learning, Data Science, Tuberculose, Gestão de Saúde, Modelo, Facebook Prophet, Previsão, Extração de Dados, Visualização de Dados

ABSTRACT

Data Science and Machine Learning tools are being considered used in the health field to assist professionals in different ways. In this work there are several tools that can help in health management and different areas. Processes and Machine Learning tools are used to predict the number of new cases of Tuberculosis in Brazil, by State, with the aim of assisting in health management across the country and for preventive measures to be adopted. Throughout the work it is possible to visualize all the stages of a Machine Learning project, from data extraction, to data treatment, a modeling using Facebook Prophet tool, a visualization of curves and numbers, the result of forecasts and the final step of interacting with the data through the Microsoft Power BI tool so that the end user can browse, understand the data and make the best decisions. The number of cases of Tuberculosis in Brazil has been growing since 2016, with a high increase in 2018 and an upward trend for the future. It is a disease that follows an annual seasonality pattern in each state, where it is possible to view the curves and understand which are the most critical months. There is a possible problem of delay and underreporting of data by some state health departments that may generate greater errors by the model, but in general, in the main states of Brazil it was possible to observe satisfactory numbers when compared to the actual data. In the model, data were used from 2010 in 2018 and then the data provided in 2019 were compared with the actual data released by the Brazilian Ministry of Health, through the DataSUS page. The total error, in 2019, was 1.42% and throughout this work display all the details.

SUMÁRIO

| | |
|---|-----------|
| 1 INTRODUÇÃO | 13 |
| 1.1 Tuberculose | 13 |
| 1.2 Objetivo do Projeto | 14 |
| 1.3 Justificativa da Escolha do Tema | 14 |
| 1.4 Estrutura do Trabalho | 14 |
| 2 CONCEITOS DE DATA SCIENCE E MACHINE LEARNING | 16 |
| 2.1 Data Science | 16 |
| 2.2 Machine Learning | 16 |
| 2.2.1 Machine Learning na Área da Saúde | 17 |
| 2.3 Big Data | 18 |
| 2.3.1 Hadoop e NoSQL | 19 |
| 2.4 Mineração de Dados | 20 |
| 2.5 Storytelling – Visualização dos Dados | 21 |
| 2.6 Kaggle | 22 |
| 3 FERRAMENTAS APLICADAS AO DATA SCIENCE | 24 |
| 3.1 Python e R | 24 |
| 3.2 Plataforma Anaconda | 24 |
| 3.2.1 Jupyter Notebook | 25 |
| 3.2.2 Google Colab | 26 |
| 3.3 NumPy | 27 |
| 3.4 Pandas | 27 |
| 3.5 Matplotlib e Seaborn – Visualização de dados | 29 |
| 3.6 Scikit-learn | 31 |
| 3.6.1 Regressão Linear | 31 |
| 3.6.2 Regressão Logística | 32 |
| 3.7 Facebook Prophet | 32 |

| | |
|---|-----------|
| 3.8 Paralelização dos Processos | 35 |
| 3.9 Web Scraping | 36 |
| 3.9.1 Beautiful Soup | 36 |
| 3.9.2 Selenium | 36 |
| 3.10 Power BI | 37 |
| 4 ESTUDO DE CASO | 38 |
| 4.1 Diagramas de Casos de Uso e de Classes | 39 |
| 4.2 DataSUS | 41 |
| 4.2.1 Coleta dos Dados | 41 |
| 4.3 Tratamento dos dados brutos | 42 |
| 4.4 Criação do Modelo de Previsão com o Prophet | 44 |
| 4.4.1 Resultados das previsões com diferentes períodos de entrada | 47 |
| 4.4.2 Regressão Linear para validação dos resultados | 51 |
| 4.5 Casos por 100mil Habitantes | 52 |
| 4.6 Visualização dos Dados | 52 |
| 5 CONCLUSÃO | 56 |
| REFERÊNCIAS | 57 |

LISTA DE FIGURAS

| | |
|--|----|
| Figura 1– Previsão do Número de Mortes por Covid-19 no Brasil em Maio de 2020 | 17 |
| Figura 2 - Plataforma Analítica de Big Data da Oracle | 18 |
| Figura 3 - Vantagens do Data Lake | 20 |
| Figura 4 - Os desafios da Mineração de Dados | 21 |
| Figura 5 - Chamados ao longo do ano (exemplo de gráfico confuso) | 22 |
| Figura 6 – Chamados ao longo do ano (exemplo de gráfico auto explicativo) | 22 |
| Figura 7 - Plataforma Anaconda e principais ferramentas | 25 |
| Figura 8 - Jupyter Notebook | 26 |
| Figura 9 – Google Colab | 27 |
| Figura 10 – Dataframe com informações sobre o COVID-19 no Brasil | 28 |
| Figura 11 – Acesso ao banco de dados Oracle e criação de um Dataframe | 29 |
| Figura 12 – Número de casos, mortes e curados pelo COVID-19 (Matplotlib) | 30 |
| Figura 13 - Gráficos de dispersão para relações conjuntas e histogramas para distribuições univariadas | 30 |
| Figura 14 – Exemplo de uma reta gerada por uma regressão linear | 31 |
| Figura 15 – Resultado do Prophet para visualizar os acessos ao perfil do jogador Peyton Manning no Wikipedia | 34 |
| Figura 16 – Valores de predição gerados pelo Prophet | 35 |
| Figura 17 – Microsoft Power BI | 37 |
| Figura 18 - Diagrama de Atividades | 38 |
| Figura 19 - Diagrama de Casos de Uso | 39 |
| Figura 20 - Diagrama de Classes | 40 |
| Figura 21 – Dados da Tuberculose no Brasil (Sistema DataSUS) | 41 |
| Figura 22 – Dataframe com números de novos casos de Tuberculose por Estado | 42 |
| Figura 23 – Tratamento do dataframe com número de casos de Tuberculose por Estado | 43 |
| Figura 24 – Dataframe em tratamento para melhor manipulação dos dados | 44 |
| Figura 25 – Avanço do número de novos casos da tuberculose no Brasil ao longo dos anos | 44 |
| Figura 26 – Dataframe pronto para rodar o Prophet | 45 |
| Figura 27 – Dataframe gerado após aplicação do Prophet | 46 |
| Figura 28 – Forte redução do número de casos no final de 2019 no AP | 48 |
| Figura 29 – Forte redução do número de casos em 2019 no RN | 48 |
| Figura 30 – Casos Reais vs. Casos Previstos em 2018 | 49 |

| | |
|---|----|
| Figura 31 – Casos Reais vs. Casos Previstos em 2019 | 49 |
| Figura 32 – Casos Reais vs Casos Previstos de 2010 a 2018 | 49 |
| Figura 33 – Curvas de Tendência e Sazonalidade gerados pelo Prophet no período de 2010 a 2018 no Brasil | 50 |
| Figura 34 – Changepoints (Mudanças de tendência detectadas pelo Prophet) | 51 |
| Figura 35 – Regressão Linear (Casos Previstos vs Casos Reais) | 52 |
| Figura 36 – Dashboard Número de casos absolutos | 53 |
| Figura 37 – Dashboard Número de casos previstos absolutos | 54 |
| Figura 38 – Dashboard com o número de casos por 100mil habitantes | 54 |
| Figura 39 – Dashboard com número previsto de casos por 100mil Habitantes | 55 |

1 INTRODUÇÃO

Gestão de saúde sempre foi um tema complicado, principalmente em países em desenvolvimento que precisam lidar com diversos desafios sociais, econômicos e políticos. Entender a evolução das doenças torna-se um desafio primordial para definir orçamentos, número de leitos, compra de material, contratação de médicos e outros profissionais da saúde, estruturar hospitais, aplicar vacinas, entre outras coisas.

O uso da tecnologia cada vez mais se faz necessário para auxiliar a tomada de decisão no setor da saúde. O projeto Watson da IBM, de inteligência artificial, já está presente em alguns hospitais e laboratórios, ajudando os médicos a dar diagnósticos precisos, utilizando uma grande quantidade de dados (Big Data) disponíveis para chegar em resultados que os seres humanos sozinhos talvez nunca chegassem (IBM, 2020). A inteligência artificial não substitui a experiência humana e o trabalho dos profissionais da saúde como um todo, mas auxilia a tomada de precisão e o melhor diagnóstico e decisão de tratamentos aos pacientes.

Ferramentas de Machine Learning ajudam a prever números de casos, número de mortes, melhores opções de tratamento e dar diagnósticos. Em 2020 o mundo passou por uma pandemia do novo coronavírus (COVID-19) e modelos de Machine Learning foram amplamente utilizados para prever novos casos e mortes, isso auxiliou aos governos a direcionar os investimentos, construir hospitais de campanha e como adotar medidas de isolamento social. Sem essas ferramentas, seria difícil entender o avanço da doença e tomar medidas precisas. Além disso, diagnósticos são feitos utilizando raio-x de pacientes, com base em exames de pacientes confirmados, utilizando modelos de Machine Learning (SILVA et al, 2020).

1.1 Tuberculose

A Tuberculose é uma doença presente em todo o mundo e muitas vezes contagiosa, estimasse que, aproximadamente, nove milhões de casos surgem todos os anos, no Brasil aproximadamente 70 mil novos casos surgem ao longo dos anos (MACHADO et al, 2011). É uma doença perigosa, que é possível se prevenir, tem cura e ainda assim assombra a população a séculos. Entender o avanço da doença, a concentração de novos casos, sintomas e comorbidades associadas é de extrema importância para que o governo possa se preparar e realizar campanhas de prevenção da doença.

Utilizando modelos de Machine Learning é possível prever novos casos da doença, por mês e por localidade, direcionar os melhores tratamentos e entender maiores riscos de morte devidos a comorbidades. Pessoas com HIV, por exemplo, chegam a ter até 25x mais chances

de contrair a Tuberculose, assim como moradores de rua chegam a ter 56x mais chances. Entendendo melhor a doença e preparando modelos precisos de Machine Learning é possível direcionar estudos e campanhas de prevenção (BRASIL, 2020a).

1.2 Objetivo do Projeto

Este trabalho tem como objetivo apresentar ferramentas de Data Science e Machine Learning na gestão de saúde. Foi escolhido a Tuberculose como exemplo para gerar a previsão de novos casos no Brasil nos próximos anos. Como relatado no tópico 1.1, a Tuberculose é uma doença presente no país a séculos e prever o número de casos por estado é importante para previsão de orçamento e planejamento em hospitais e campanhas de prevenção. Utilizando diversas ferramentas de Machine Learning, como o Facebook Prophet, e dados disponíveis pelo ministério da saúde do Brasil, é possível entender as curvas de sazonalidade e tendência da Tuberculose e com isso prever o número de casos. Toda metodologia empregada pode ser utilizada para outras doenças, assim auxiliando a gestão de saúde em todo país, seja na iniciativa privada ou na gestão pública dos estados, municípios e do governo federal.

1.3 Justificativa da Escolha do Tema

A gestão da saúde num país como o Brasil, sofre bastante com baixos investimentos e problemas sociais e o uso de tecnologia pode nos levar a um caminho melhor no futuro. A tecnologia ajuda a otimizar gastos e direcionar investimentos em locais com maior necessidade. Entender o avanço de doenças, melhores alternativas de tratamento e dar diagnósticos precisos, utilizando modelos de Machine Learning, é um excelente caminho para o nosso sistema de saúde no futuro. Essas ferramentas são essenciais para avançarmos na saúde com os recursos disponíveis pelo governo.

A Tuberculose é uma doença bastante conhecida e com muita informação disponível, sendo possível gerar modelos de previsão e identificar os períodos e locais de maior incidência da doença. Assim, este trabalho visa apresentar um exemplo de modelo e apresentar previsões baseadas nestes modelos para que seja possível guiar as políticas públicas de prevenção de doença e direcionamento de recursos.

1.4 Estrutura do Trabalho

O trabalho apresenta em sequência todas as etapas de ciência de dados, suas principais ferramentas e o projeto passando por todas essas etapas. No capítulo 2 é apresentado conceitos de Data Science e Machine Learning, mostrando a coleta de dados, armazenagem, mineração de dados, análise de dados e por último a visualização da informação gerada. Neste capítulo

ainda é apresentado conceitos de Big Data, principais aplicações e exemplos de plataformas utilizadas para projetos de ciência de dados, como o Kaggle.

No capítulo 3 é apresentado as principais ferramentas utilizadas em cada uma das etapas apresentada no capítulo 2, entre elas a linguagem de programação Python, a plataforma Anaconda, as plataformas de programação Jupyter e Google Colab, as principais bibliotecas utilizadas para tratamento de dados como o NumPy e o Pandas, visualização de gráficos como o Matplotlib e o Seaborn, geração de modelos utilizando o Scikit-learn e o Facebook Prophet, extração Web utilizando BeautifulSoup e Selenium e visualização de dados utilizando o Microsoft Power BI.

No capítulo 4 as principais ferramentas apresentadas são utilizadas para geração de um modelo de previsão de número de novos casos de tuberculose no Brasil por mês e por estado, utilizando o Facebook Prophet, que utiliza fatores de sazonalidade e tendência para as previsões. O projeto é desenvolvido em Python e os dados coletados direto do DataSUS (plataforma de divulgação de dados do ministério da saúde do Brasil). Para coleta de dados é utilizado ferramentas de web scraping, após a coleta é realizado o tratamento dos dados para que seja rodado o modelo, utilizando programação distribuída para otimização de tempo e boas práticas de programação. No final do capítulo é apresentada a possível visualização dos números históricos e previstos num dashboard interativo desenvolvido no Power BI, de forma que o usuário possa ter acesso aos dados tratados e com isso possa tomar as melhores decisões.

2 CONCEITOS DE DATA SCIENCE E MACHINE LEARNING

2.1 Data Science

A quantidade de informação disponível e gerada todos os dias vem mudando a forma como as empresas e governos definem suas estratégias. O crescimento da internet no século XXI e a enorme oferta de *smartphones* permitiu que boa parte da população mundial esteja conectada e compartilhando dados entre si e com empresas de todo o mundo. A informação ficou conhecida como o “petróleo do século XXI”.

Com toda essa informação disponível era necessário ter profissionais capacitados para manipular e analisar esses dados, de forma que as empresas pudessem seguir seus rumos com base nessas análises, tornando-se empresas *data driven*. A profissão de Cientista de Dados ficou conhecida na última década como a profissão mais *sexy* do mundo, segundo Davenport (2012). Profissionais das áreas de Ciências da Computação, Estatística e cursos de Engenharia são os mais procurados para embarcar nessa área. Há uma piada que diz que um cientista de dados é alguém que sabe mais sobre estatística do que um cientista da computação e mais sobre ciência da computação do que um estatístico (GRUS, 2016).

Segundo Passos (2016), Ciência de Dados pode ser definida como um conjunto de técnicas utilizadas no processamento e análise de dados, com intuito de fornecer informações para decisões inteligentes. Para tanto, mescla-se diversas áreas do conhecimento, desde conceitos simples de estatística até complexos algoritmos.

2.2 Machine Learning

A evolução dos computadores e a quantidade de dados disponíveis, foi possível criar o conceito de aprendizagem de máquina para que ela possa tomar decisões e realizar previsões com base nas entradas fornecidas. Temos dois principais métodos de aprendizagem, o supervisionado e o não-supervisionado.

No método supervisionado o modelo recebe dados, sendo treinado para dar uma resposta com base na experiência adquirida com informação pré-definida. Neste trabalho foi utilizado este tipo de modelo com base em uma série histórica para realizar uma previsão. No método não-supervisionado, espera-se que o modelo detecte padrões e encontre relações, sem que o modelo possua um treinamento prévio sobre aqueles dados (MATOS, 2019).

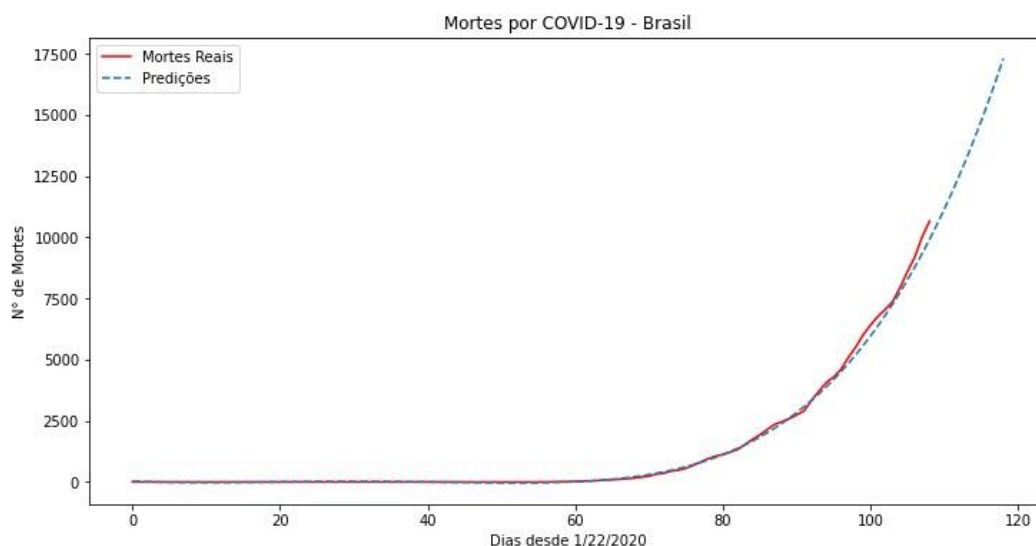
Um bom e conhecido exemplo de uso de ferramentas de Machine Learning na atualidade é a personalização de conteúdo utilizada pela empresa Netflix para fornecer ao usuário da

plataforma de filmes e séries o conteúdo que tem mais proximidade com o perfil daquele usuário. Assim os assinantes, ao acessarem o aplicativo recebem recomendações que satisfaçam seus interesses.

2.2.1 Machine Learning na Área da Saúde

Ferramentas de Machine Learning estão sendo amplamente usadas em todo o mundo para prever número de casos de uma determinada doença, probabilidade de morte de um paciente, opções de tratamento, desenvolvimento de vacinas, etc; Em 2020, com a pandemia do Covid-19 as famosas curvas de número de casos e de mortes estavam sendo divulgadas a todo momento e sendo utilizadas como base para construção de novos leitos de UTI, planejamento de retomada do isolamento social pelos governos e empresas e identificação dos locais mais críticos.

Figura 1– Previsão do Número de Mortes por Covid-19 no Brasil em Maio de 2020



Fonte: Elaborado pelo autor

Na Figura 1 é possível visualizar uma curva gerada por um modelo de Machine Learning para identificar a previsão do número de mortes por Covid-19 no Brasil entre os dias 8 e 17 de Maio de 2020. A linha vermelha mostra o número de casos reais informados pelo Ministério da Saúde do Brasil e a linha tracejada azul mostra a previsão de novas mortes para dez dias após a última divulgação. Este tipo de informação auxilia, por exemplo, o governo a planejar melhor o sistema de saúde e funerário, bem como entender o avanço da doença.

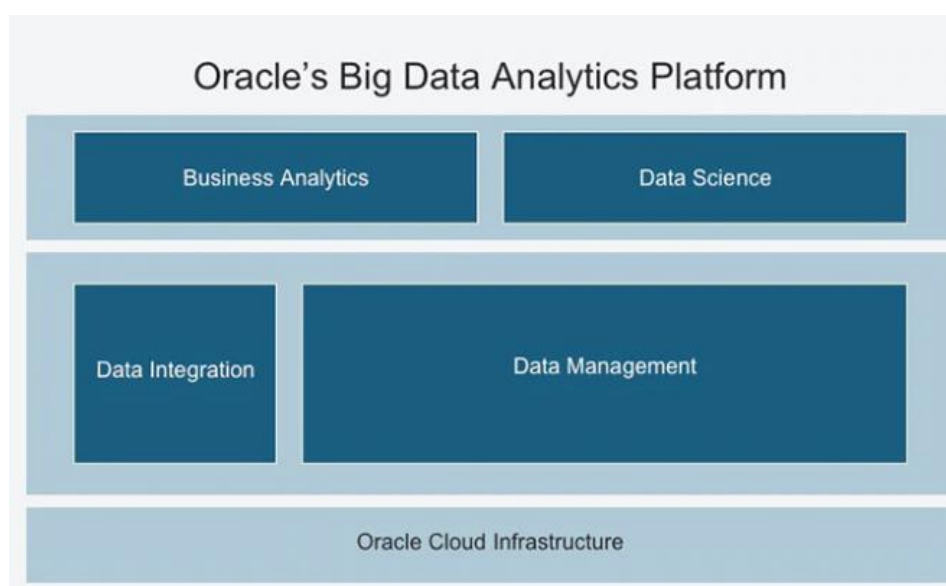
Entre alguns estudos brasileiros na área da saúde, utilizando ferramentas de Machine Learning, Oliveira et. al. (2017) desenvolveu modelos preditivos de diabetes não diagnosticadas. Foram utilizados cinco algoritmos diferentes: regressão logística, redes neurais,

K-nearest neighbor, *random forest* e *naive bayes*. O modelo com melhores resultados foi o de regressão logística e entre 403 pessoas entrevistadas, 274 foram identificadas como casos positivos de diabetes. Com base no modelo os autores implementaram um aplicativo que registra um *score* de risco para estimar possíveis casos de diabetes não diagnosticadas. A regressão logística é uma técnica de classificação muito usada em estudos médicos e epidemiológicos.

2.3 Big Data

A enorme disponibilidade de dados nos últimos anos colocou em alta o conceito de Big Data. “Big Data é um conjunto de dados maior e mais complexo, especialmente de novas fontes de dados. Esses conjuntos de dados são tão volumosos que o software tradicional de processamento de dados simplesmente não consegue gerenciá-los.” (Oracle, 2020a). Manipular, armazenar e analisar toda essa informação é um grande desafio.

Figura 2 - Plataforma Analítica de Big Data da Oracle



Fonte: (ORACLE, 2020b)

Em meados da década de 2000, com o surgimento de grandes redes sociais, como Facebook e YouTube ficou evidente a grande quantidade de dados gerados e coletados por essas plataformas com os seus usuários. A maior disponibilidade de dados também significa mais informação para modelos de Machine Learning e Inteligência Artificial para melhorar a interação com o usuário desses serviços.

Para termos uma noção do volume de dados coletados em alguns projetos diariamente, a NASA gera cerca de um *terabyte* de dados por dia. Uma missão da agência espacial norte-

americana chega a gerar 192 *terabytes* para gerar uma imagem de alta definição da lua. (ENTWISTLE, 2011)

Grandes e médias empresas utilizam diversas ferramentas para lidar com seus projetos de Big Data e cada vez mais precisam de profissionais capazes de trabalhar com essas ferramentas e investir em novas tecnologias para suportar o volume diário de informação gerada. Nos últimos anos, houve um aumento maciço nas startups de Big Data, todas tentando lidar com o volume de informação e ajudando as organizações a entender o Big Data.

2.3.1 Hadoop e NoSQL

Quando falamos de Big Data não podemos deixar de citar duas ferramentas muito importantes para manipular uma enorme quantidade de dados: Hadoop e NoSQL. “A biblioteca de software Apache Hadoop é uma estrutura que permite o processamento distribuído de grandes conjuntos de dados entre clusters de computadores usando modelos de programação simples.” (HADOOP, 2020).

NoSQL (Not Only SQL) trata-se de um banco de dados não relacional. Manipular e armazenar grande quantidade de dados usando bancos relacionais seria uma tarefa bastante complicada e não escalável, então grandes empresas como Facebook e Google começaram a utilizar banco de dados não relacionais para usar nos seus projetos com Big Data.

2.3.1.1 Data Lake

Data lake é o nome dado ao repositório de dados, de forma que a empresa possa armazenar todos os seus dados brutos para futuramente manipulá-los. Um projeto de data lake bem realizado ajuda toda a empresa a lidar com o Big Data. A ideia do data lake é ter disponível uma grande quantidade de dados brutos para que os cientistas de dados possam explorar os dados e realizar suas análises, conforme apresenta a figura 3.

Figura 3 - Vantagens do Data Lake



Fonte: (SADI, 2019)

2.3.1.2 Data Warehouse

Assim como o data lake, um data warehouse (ou armazém de dados) tem o objetivo de armazenar dados para que possam ser explorados por cientistas de dados. A diferença é que um data lake lida com dados brutos, não filtrados, para serem utilizadas em algum momento com uma finalidade específica, já no data warehouse, os dados já estão prontos para análises, pois foram filtrados e contextualizados para que possa ser gerada uma informação. Muitas empresas utilizam o data warehouse na nuvem, ganhando escalabilidade e diminuindo custos operacionais. (ORACLE, 2020c)

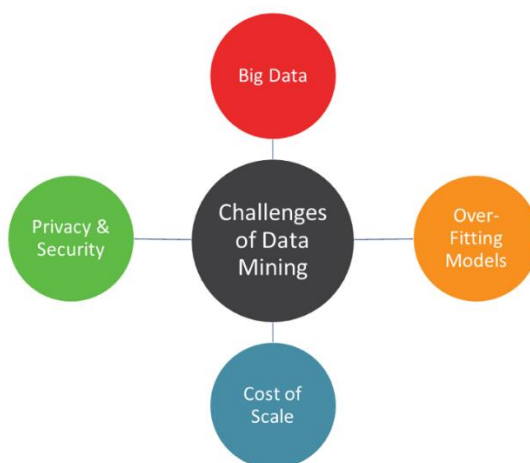
2.4 Mineração de Dados

Com grande volume de dados disponíveis para análise, é preciso saber tratar esses dados e capturar o que de fato é relevante para criação de modelos, insights e tomada de decisão. "Mineração de Dados é a análise de grandes conjuntos de dados a fim de encontrar relacionamentos inesperados e de resumir os dados de uma forma que eles sejam tanto úteis quanto compreensíveis ao dono dos dados". (HAND et al.,2001) Já de acordo com Fayyad et al(1996), mineração de dados é um passo no processo de descoberta de conhecimento, que consiste na realização da análise dos dados e na aplicação de algoritmos de descoberta que, sob certas limitações computacionais, produzem um conjunto de padrões de certos dados.

O processo de mineração de dados é de extrema importância para que os dados façam algum sentido e possam ser utilizados pelas companhias e governos, sem este processo teríamos apenas dados brutos que seriam passados despercebidos.

Há muitos desafios na mineração de dados, entre eles a própria coleta de dados (Big Data), o custo elevado para armazenamento e tratamento dos dados, utilizar as melhores ferramentas, privacidade e segurança dos dados e cuidados ao treinar os modelos para não gerar informações com sobreajuste (*overfitting* – quando o modelo se ajusta muito bem aos dados informados mas não consegue boas previsões). (MICROSTRATEGY, 2020). A figura 4 mostra os principais desafios.

Figura 4 - Os desafios da Mineração de Dados



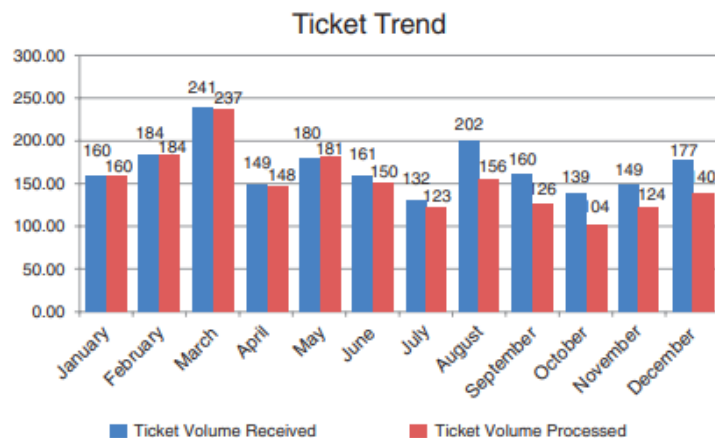
Fonte: (MICROSTRATEGY, 2020)

2.5 Storytelling – Visualização dos Dados

Após a mineração e análise dos dados existe uma etapa fundamental da Ciência de Dados que é como apresentar os dados para o público. Muitas vezes a equipe de data science tem as informações em mãos, sabem que aquele dado é relevante para o projeto, mas não conseguem uma forma de mostrar a sua análise.

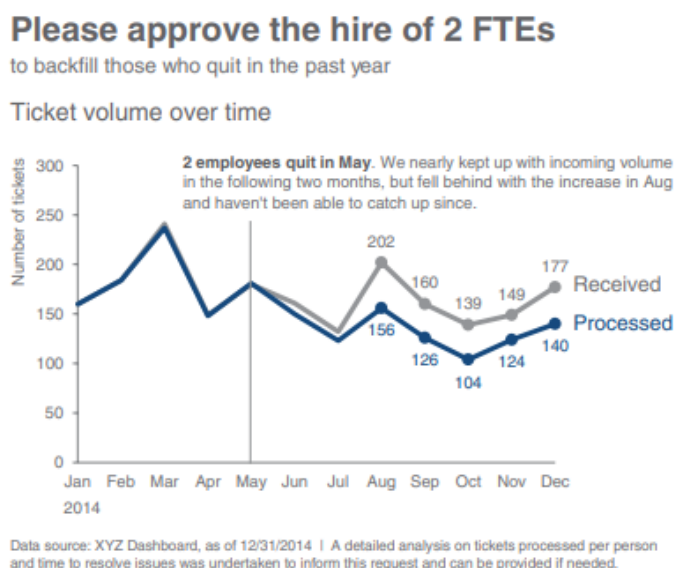
É comum a utilização de gráficos para apresentar os números e a diversidade de gráficos existentes nas principais ferramentas é enorme, então para isso é preciso saber qual gráfico escolher e até o melhor conjunto de cores, para destacar o objetivo da análise. Na figura 5 é possível visualizar um exemplo de gráfico ruim com redundância no eixo y e as cores não deixam claro a mensagem. Na figura 6, a mesma informação é apresentada de forma mais clara, num gráfico de linhas, onde claramente é possível identificar o número de tickets recebidos e processados (NUSSBAUMER, 2012).

Figura 5 - Chamados ao longo do ano (exemplo de gráfico confuso)



Fonte: (NUSSBAUMER, 2012)

Figura 6 – Chamados ao longo do ano (exemplo de gráfico auto explicativo)



Fonte: (NUSSBAUMER, 2012)

Há diversas técnicas de apresentação de dados para que o público alvo tenha fácil entendimento do que está sendo apresentado. Para isso existem ferramentas que auxiliam a montagem do Storytelling, como o Microsoft Powerpoint ou o Microsoft PowerBI, que permite que o usuário tenha uma interação com os dados e consiga explorar da melhor forma.

2.6 Kaggle

O site Kaggle.com reúne diversos desafios de Ciência de Dados e Machine Learning. Empresas e governos disponibilizam os desafios para que os usuários possam resolvê-los da forma mais precisa possível. Muitos desafios têm recompensas financeiras para os mais bem

colocados. Recentemente com a pandemia do Covid-19, diversas instituições colocaram desafios com o objetivo de prever o número de novos casos e mortes da doença, assim como desafios para diagnosticar casos de Covid-19 através do raio-x do paciente, usando ferramentas de Machine Learning, com base em exames de pacientes com casos confirmados em diversos países. Há desafio também para utilizar Machine Learning para reunir e classificar o maior número possível de artigos científicos sobre a doença, para facilitar os pesquisadores em seus estudos.

Um projeto desenvolvido por Eloá Guedes e Júlio Guedes realizado na plataforma Kaggle, mostra o avanço diário do Covid-19 no Brasil, mostrando diversos *insights* e informações relevantes para o entendimento da doença no país, utilizando ferramentas de Ciência de Dados e Machine Learning. No projeto é possível ver a letalidade ao longo do tempo da doença por estado, a distribuição geográfica dos casos confirmados, relação do avanço da doença com indicadores socioeconômicos, quantidade de leitos disponíveis por estado, previsão do número de novos casos (GUEDES E., GUEDES J., 2020).

A plataforma é muito utilizada por quem está começando na área para visualizar como os mais experientes desenvolvem seus projetos e muito utilizados por pessoas experientes na área para resolver desafios, seja buscando recompensas ou não. Na área da saúde tem ajudado diversos projetos e pesquisadores a procurar novos tratamentos e diagnósticos, por exemplo.

3 FERRAMENTAS APLICADAS AO DATA SCIENCE

Para realizar todas as etapas de coleta, mineração, modelagem, análise e visualização dos dados existem diversas ferramentas que auxiliam os profissionais de Ciência de Dados para que tenham todos os processos automatizados, de forma escalável e performáticos.

Entre as principais ferramentas utilizadas temos a linguagem de programação Python ou R, a plataforma *open source* Anaconda que conta com diversos pacotes para uso de Ciência de Dados, como o Jupyter Notebook, as bibliotecas Pandas (*data frames*), NumPy (cálculos matemáticos com arrays), Scikit-learn (Machine Learning), Matplotlib (visualização de dados), entre outras. O Facebook fornece uma biblioteca *open source* chamada Prophet para predição de dados através de séries temporais e por fim algumas ferramentas de visualização como o Microsoft Power BI.

3.1 Python e R

As duas principais linguagens de programação utilizadas por cientistas de dados são o Python e o R. O R é muito conhecido no meio acadêmico, principalmente nos cursos de engenharia e estatística e permite realizar cálculos matemáticos, gráficos e simulações com maior simplicidade. É a principal linguagem de programação utilizada no meio estatístico e muito utilizada por cientistas de dados. (EQUIPE DSA, 2018a)

O Python, mais conhecido entre os estudantes e profissionais de TI é amplamente utilizada por cientistas de dados e vem ganhando bastante espaço no desenvolvimento de aplicações para IoT (Internet of Things) e automação industrial. Assim como o R, permite realizar cálculos matemáticos, modelos de Machine Learning, simulações e visualização de gráficos de forma simples e prática.

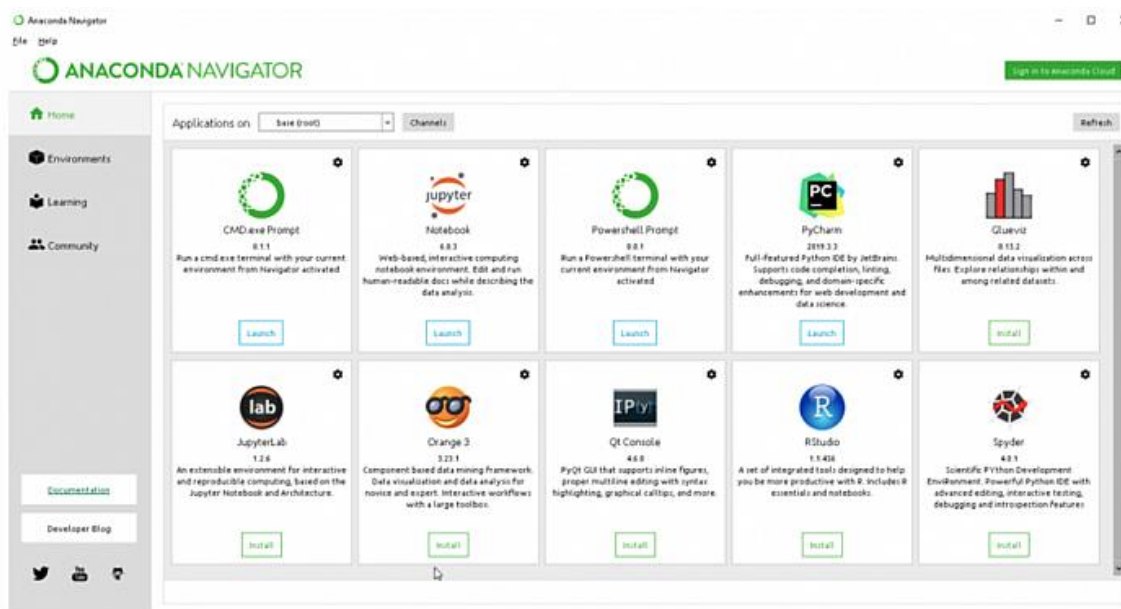
Diversas bibliotecas para Ciência de Dados estão disponíveis para R e Python, mas o Python possui uma maior popularidade e diversidade de ferramentas que fazem com que seja a principal linguagem de programação utilizada nesta área. (EQUIPE DSA, 2020)

3.2 Plataforma Anaconda

Uma das principais plataformas para aplicação de ciência de dados é a Anaconda, uma plataforma *open source*, de código aberto, disponível para Python e R e com diversas ferramentas já disponíveis no momento da instalação, como o Jupyter, NumPy, Pandas, Scikit-learn, Matplotlib, TensorFlow, entre outros (Figura 7). Com essa plataforma e todas essas

ferramentas disponíveis é possível programar em Python, coletar, manipular e analisar os dados, além de gerar e rodar modelos de Machine Learning (ANACONDA-INC, 2020).

Figura 7 - Plataforma Anaconda e principais ferramentas



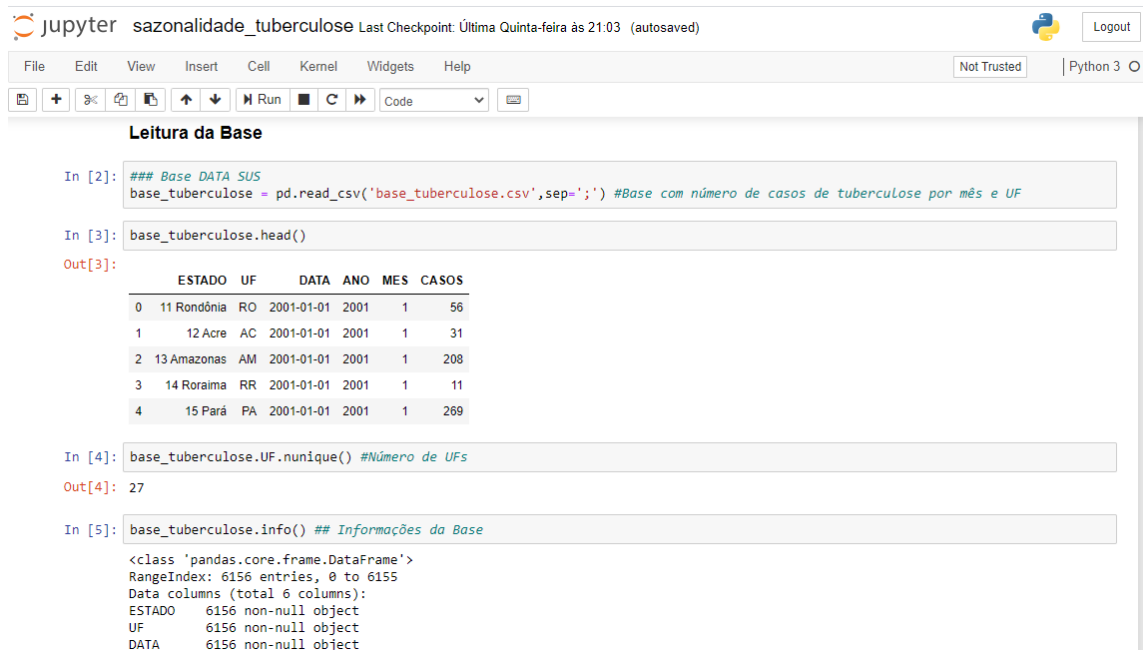
Fonte: Elaborado pelo autor

3.2.1 Jupyter Notebook

O projeto Jupyter, também *open source*, criado em 2014, sem fins lucrativos, foi criado para facilitar a programação voltada a Ciência de Dados e à computação científica. Entre os principais apoiadores e patrocinadores do projeto estão empresas como Microsoft, Netflix e Google e entre as instituições que utilizam esta ferramenta estão a IBM, Nasa e diversas Universidades no mundo todo. Nesta plataforma é possível programar em Python, carregar as principais bibliotecas e APIs disponíveis para Ciência de Dados e com isso realizar todas as etapas de um modelo de Machine Learning. (PROJECT-JUPYTER, 2020)

O que chama a atenção nesta plataforma é a simplicidade de aprendizado e facilidade para gerar modelos e cálculos matemáticos. O Jupyter Notebook é utilizado num navegador Web, como o Google Chrome, por exemplo, conforme é mostrado na figura 8. É um dos principais motivos para o Python ter se tornado a principal linguagem de programação no meio de Data Science.

Figura 8 - Jupyter Notebook



The screenshot shows a Jupyter Notebook titled 'sazonalidade_tuberculose'. The interface includes a top bar with the Jupyter logo, the notebook title, and a 'Last Checkpoint' timestamp. Below this is a menu bar (File, Edit, View, Insert, Cell, Kernel, Widgets, Help) and a toolbar with icons for file operations and execution. The notebook content is divided into cells. The first cell, titled 'Leitura da Base', contains two input cells. The first input cell (In [2]:) contains a comment '### Base DATA SUS' and a line of code: `base_tuberculose = pd.read_csv('base_tuberculose.csv', sep=';') #Base com número de casos de tuberculose por mês e UF`. The second input cell (In [3]:) contains `base_tuberculose.head()`. The output of the second cell (Out[3]:) is a table with 6 columns: ESTADO, UF, DATA, ANO, MES, and CASOS. The table shows the first 5 rows of data. The third input cell (In [4]:) contains `base_tuberculose.UF.nunique()` with a comment '#Número de UFs'. The output (Out[4]:) is the number 27. The fourth input cell (In [5]:) contains `base_tuberculose.info()` with a comment '## Informações da Base'. The output shows the data type and statistics for each column.

```
### Base DATA SUS
base_tuberculose = pd.read_csv('base_tuberculose.csv', sep=';') #Base com número de casos de tuberculose por mês e UF

base_tuberculose.head()

Out[3]:
   ESTADO UF  DATA ANO  MES  CASOS
0   11 Rondônia RO  2001-01-01  2001    1    56
1    12 Acre AC   2001-01-01  2001    1    31
2   13 Amazonas AM  2001-01-01  2001    1   208
3   14 Roraima RR  2001-01-01  2001    1    11
4   15 Pará PA   2001-01-01  2001    1   269

In [4]: base_tuberculose.UF.nunique() #Número de UFs
Out[4]: 27

In [5]: base_tuberculose.info() ## Informações da Base
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 6156 entries, 0 to 6155
Data columns (total 6 columns):
ESTADO    6156 non-null object
UF        6156 non-null object
DATA      6156 non-null object
```

Fonte: Elaborado pelo autor

3.2.2 Google Colab

O Google criou a ferramenta Google Colab, similar ao Jupyter Notebook, mas disponível na nuvem. Assim, qualquer usuário pode rodar os seus códigos em Python na nuvem, sem usar processamento local. É uma ferramenta muito interessante para estudantes e usuários comuns, pois não é necessário ter um bom hardware para rodar os modelos, já que o próprio Google disponibiliza GPU na nuvem. Assim como o Jupyter, o Google Colab é utilizado por meio de um navegador Web, preferencialmente o Google Chrome (figura 9) (GOOGLE, 2020).

Para utilizar a plataforma basta ter uma conta Google. Os arquivos de entrada e saída podem estar disponíveis no Google Drive, assim como o código. Todas as bibliotecas estão disponíveis e permitem rodar um modelo de Machine Learning e plotar gráficos, por exemplo.

Figura 9 – Google Colab



```
[ ] from selenium import webdriver
from selenium.webdriver import ActionChains, ChromeOptions
from selenium.webdriver.support.ui import Select
from selenium.webdriver.common import keys
from bs4 import BeautifulSoup
import time
import pandas as pd

[ ] def safe_click(xpath, driver, max_retry = 3):
    try:
        to_click = driver.find_element_by_xpath(xpath)
        to_click.click()
    except:
        print("ERRO NO CLICK")
        if max_retry != 0:
            time.sleep(10)
            safe_click(xpath, driver, max_retry-1)
        else:
            print('NUMERO DE TENTATIVAS EXCEDIDA')
            get_current_data()
            raise

def safe_action(xpath, action = 'right_click', max_retry = 3):
    if action == 'right_click':
        try:
            to_act = driver.find_element_by_xpath(xpath)
            actions = ActionChains(driver)
```

Fonte: Elaborado pelo autor

3.3 NumPy

O NumPy é uma das principais bibliotecas disponíveis para Python com o objetivo de realizar cálculos matemáticos com arrays multidimensionais. Ferramenta essencial para criação de modelos de Machine Learning, permitindo realizar cálculos complexos em arrays de forma prática e de simples escrita. Possui funções de álgebra linear, muito utilizadas na computação gráfica, por exemplo (OLIPHANT, 2006).

3.4 Pandas

Pandas talvez seja a biblioteca mais importante e utilizada no meio de Data Science. Trata-se de um projeto *open source*, patrocinado pela empresa NumFOCUS. Desde 2012 vem sendo amplamente utilizada na análise de dados e entre os principais objetivos estão a criação de um objeto dataframe (tabela), que permite a manipulação dos dados. Permite que dados sejam carregados de diversas maneiras e formatos, por exemplo, é possível utilizar arquivos no formato XLS (Excel) ou CSV ou carregar direto de um banco de dados SQL. Com o Pandas é possível manipular tabelas de forma similar aos comandos de SQL, podendo selecionar itens de acordo com condições, juntar tabelas, excluir linhas, criar colunas, inserir dados, entre outras coisas (PANDAS, 2020).

A figura 10 mostra o exemplo de um dataframe criado pelo Pandas, rodando no Jupyter Notebook, com informações sobre o COVID-19 no Brasil. No dataframe é possível ver a data do registro, o país, a data da última atualização, o número de casos confirmados até aquele momento, número de mortes e pacientes recuperados pela doença.

Figura 10 – Dataframe com informações sobre o COVID-19 no Brasil

```
df_brasil[['ObservationDate', 'Country/Region', 'Last Update', 'Confirmed', 'Deaths', 'Recovered']].tail(15)
```

| | ObservationDate | Country/Region | Last Update | Confirmed | Deaths | Recovered |
|-------|-----------------|----------------|---------------------|-----------|---------|-----------|
| 20273 | 05/01/2020 | Brazil | 2020-05-02 02:32:27 | 92202.0 | 6412.0 | 38039.0 |
| 20596 | 05/02/2020 | Brazil | 2020-05-03 02:32:28 | 97100.0 | 6761.0 | 40937.0 |
| 20919 | 05/03/2020 | Brazil | 2020-05-04 02:32:28 | 101826.0 | 7051.0 | 42991.0 |
| 21242 | 05/04/2020 | Brazil | 2020-05-05 02:32:34 | 108620.0 | 7367.0 | 45815.0 |
| 21565 | 05/05/2020 | Brazil | 2020-05-06 02:32:31 | 115455.0 | 7938.0 | 48221.0 |
| 21888 | 05/06/2020 | Brazil | 2020-05-07 02:32:28 | 126611.0 | 8588.0 | 51370.0 |
| 22211 | 05/07/2020 | Brazil | 2020-05-08 02:32:32 | 135773.0 | 9190.0 | 55350.0 |
| 22534 | 05/08/2020 | Brazil | 2020-05-09 02:32:35 | 146894.0 | 10017.0 | 59297.0 |
| 22857 | 05/09/2020 | Brazil | 2020-05-10 02:32:30 | 156061.0 | 10656.0 | 61685.0 |
| 23180 | 05/10/2020 | Brazil | 2020-05-11 02:32:30 | 162699.0 | 11123.0 | 64957.0 |
| 23503 | 05/11/2020 | Brazil | 2020-05-12 03:32:27 | 169594.0 | 11653.0 | 67384.0 |
| 23826 | 05/12/2020 | Brazil | 2020-05-13 03:32:26 | 178214.0 | 12461.0 | 72597.0 |
| 24149 | 05/13/2020 | Brazil | 2020-05-14 03:32:28 | 190137.0 | 13240.0 | 78424.0 |
| 24473 | 05/14/2020 | Brazil | 2020-05-15 02:33:02 | 203165.0 | 13999.0 | 79479.0 |
| 24850 | 05/15/2020 | Brazil | 2020-05-16 02:32:19 | 220291.0 | 14962.0 | 84970.0 |

Fonte: Elaborado pelo autor

A utilização do Pandas é fundamental na etapa de coleta e organização dos dados. Antes de inserir os dados em um modelo é preciso analisar o dataframe, verificar valores faltantes, analisar colunas, verificar correlações, criar colunas. Ferramentas como o Microsoft Excel são muito populares para visualização e manipulação de tabelas, mas o Pandas agregado ao Python e outras bibliotecas, permite a automatização de processos de forma simples e rápida, sendo possível, por exemplo, carregar uma tabela com milhões de linhas e milhares de colunas, o que não seria possível no Excel.

A etapa de processamento de dados é uma das mais importantes para o bom resultado de um modelo. O mau tratamento de dos dados é um dos grandes responsáveis por modelos com baixa precisão de assertividade. Por isso o Pandas pode ser considerado a ferramenta mais importante para um cientista de dados. Através das ferramentas Jupyter e Pandas é possível ler dados direto de uma tabela de um banco de dados. Diversos bancos de dados possuem interfaces para comunicação com o Python, entre eles o MySQL, Oracle e SQL Server.

Para realizar a leitura de um banco de dados Oracle, por exemplo, é necessário instalar a biblioteca cx_Oracle e importá-la no código. Para conectar-se ao banco é preciso inserir as credenciais e uma query e então, através do Pandas, realizar a leitura da tabela e chamar no código, conforme mostra a figura 11.

Figura 11 – Acesso ao banco de dados Oracle e criação de um Dataframe

Acesso Banco de Dados Oracle

```
import cx_Oracle #Biblioteca Oracle
conn = acessobd() #Função com credencias do banco de dados
#Query para acessar uma tabela e obter os dados
query = f"select * from tabela where no_am={data_ref} or no_am={data_ant}"
```

```
#Criação do dataframe com os dados da tabela
df = pd.read_sql(query,conn)
```

df|

| | DATA | SG_UF | TIPO |
|--------|--------|-------|--------------|
| 0 | 202002 | MG | Usinas/dest. |
| 1 | 202003 | MG | Usinas/dest. |
| 2 | 202002 | MG | Usinas/dest. |
| 3 | 202003 | MG | Usinas/dest. |
| 4 | 202002 | MG | Usinas/dest. |
| ... | ... | ... | ... |
| 302172 | 202003 | RS | Indústria |
| 302173 | 202002 | MG | Indústria |
| 302174 | 202003 | MG | Indústria |
| 302175 | 202002 | RJ | Outros |
| 302176 | 202003 | RJ | Outros |

302177 rows × 3 columns

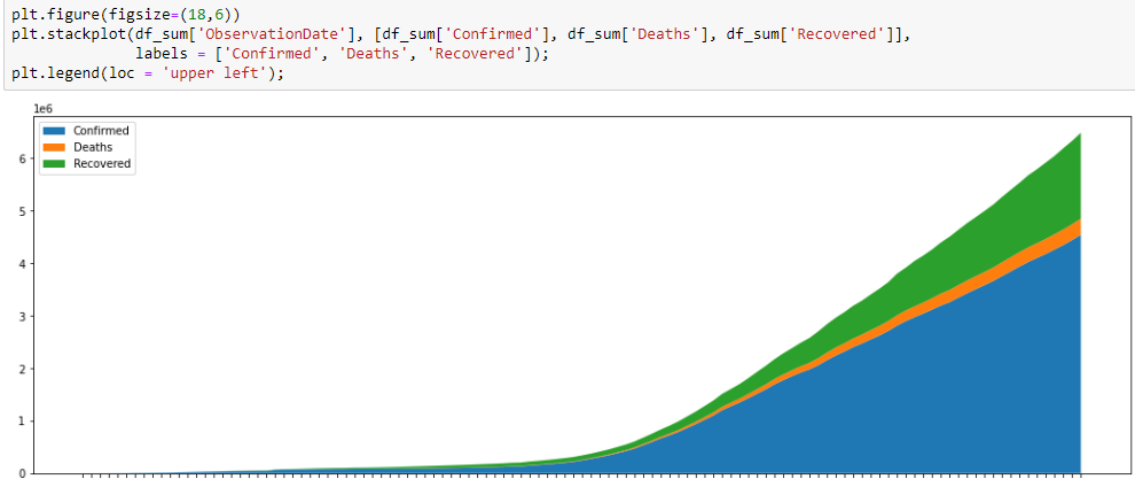
Fonte: Elaborado pelo autor

3.5 Matplotlib e Seaborn – Visualização de dados

Duas bibliotecas bastante utilizadas para visualização de gráficos, interpretação de dados, visualização de correlação, entre outras coisas, são o Matplotlib e o Seaborn.

O Matplotlib permite gerar gráficos interativos no Python, de forma que o programador possa analisá-los de uma forma visual, comparando dados de diversas formas. É possível gerar gráficos em 3D e mapas, por exemplo. Assim como o NumPy, também é um projeto patrocinado pela NumFOCUS e de código aberto (HUNTER, 2007). Na figura 12 é possível ver um gráfico gerado através do Matplotlib no Jupyter Notebook, que mostra as curvas de casos, mortes e curados pelo COVID-19, até Maio de 2020.

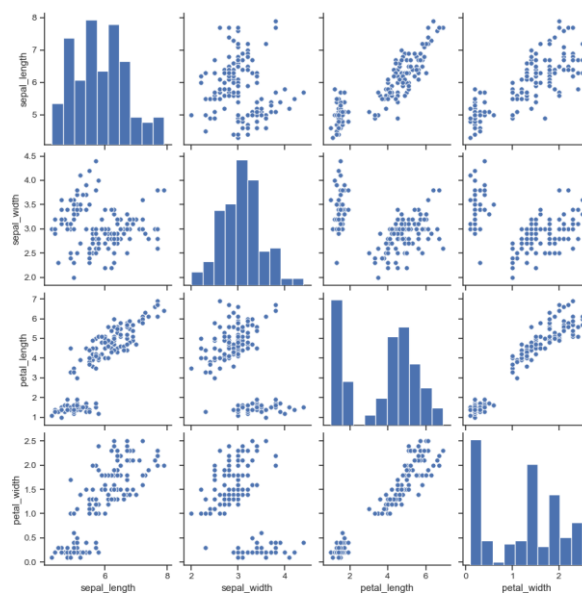
Figura 12 – Número de casos, mortes e curados pelo COVID-19 (Matplotlib)



Fonte: Elaborado pelo autor

O Seaborn é uma biblioteca de visualização de dados de mais alto nível, baseado no Matplotlib. É possível gerar gráficos estatísticos a partir de dataframes do Pandas. O Seaborn permite ao cientista de dados criar visualizações mais atraentes para o usuário e para análise dos dados (WASKOM, 2020a). É uma das ferramentas imprescindíveis no processamento de dados antes de inseri-los nos modelos. Na figura 13 é possível ver um gráfico do Seaborn, usando o comando pairplot, que vai mostrar ao programador a correlação entre os dados das colunas de um dataframe (WASKOM, 2020b).

Figura 13 - Gráficos de dispersão para relações conjuntas e histogramas para distribuições univariadas



Fonte: (WASKOM, 2020b).

3.6 Scikit-learn

O Scikit-learn é uma biblioteca do Python com diversos algoritmos de Machine Learning para problemas supervisionados e não supervisionados. (Pedregosa et al, 2011) É o principal pacote utilizado para rodar modelos de Machine Learning, amplamente utilizado por cientistas de dados.

Permite utilizar as ferramentas NumPy e Scipy para cálculos numéricos. Entre os principais algoritmos estão os de classificação, regressão e clusterização. Espera-se que ao rodar os algoritmos do Scikit-learn os dados já estejam tratados e processados para que seja possível treinar os modelos da melhor forma.

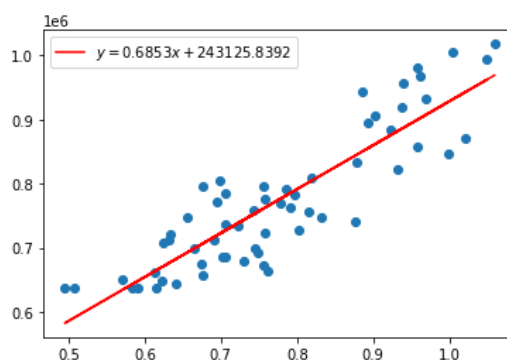
Entre os principais algoritmos dentro do Scikit-learn estão:

- Regressão Linear
- Regressão Logística
- Regressão Polinomial
- Nearest Neighbors
- Árvore de Decisão

3.6.1 Regressão Linear

Segundo Stevenson (1986), “A correlação mede a força, ou grau, de relacionamento entre duas variáveis; a regressão dá uma equação que descreve o relacionamento em termos matemáticos.” A regressão permite analisar dados históricos e encontrar correlações entre variáveis, uma dependente e outra independente de forma que seja possível prever dados futuros. O resultado da regressão linear é uma reta, onde é possível visualizar num gráfico a dispersão e a correlação entre os dados, conforme apresentado na figura 14 (Sell, 2005).

Figura 14 – Exemplo de uma reta gerada por uma regressão linear



Fonte: Elaborado pelo autor

A regressão linear possui a equação $Y = \beta_0 + \beta_1 X + \varepsilon$, onde Y é a variável dependente, β_0 é o termo constante, ou intercepto de Y , $\beta_1 X$ é a variável independente e representa a inclinação da reta e ε representa o erro aleatório (resíduo) (RAMOS, 2020).

Ao gerar o resultado de uma regressão linear é possível visualizar o coeficiente R^2 que mostra o quão próximos estão os dados da reta. O R^2 é igual a variação explicada dividido pela variação total, onde quanto mais próximo de 1 melhor o resultado e quanto mais próximo de zero, pior o resultado. Não necessariamente um R^2 baixo seja ruim, já que em alguns projetos é esperado esse resultado devido a dispersão dos dados.

3.6.2 Regressão Logística

A regressão logística é um modelo linear, utilizado principalmente quando temos variáveis categóricas e o resultado de sua equação é binário, 0 ou 1, ou sim ou não. Um exemplo famoso de uso de regressão logística é o desafio do *Titanic*, disponível no site Kaggle <<https://www.kaggle.com/c/titanic>>, para descobrir a probabilidade de uma pessoa sobreviver ao naufrágio do famoso navio. Para chegar na resposta final é verificado no modelo diversas variáveis categóricas, como o sexo da pessoa, faixa etária, onde foi comprado o bilhete e a classe que a pessoa se hospedou. Ao final do desafio, utilizando a regressão logística, é possível verificar, por exemplo, que mulheres, crianças e pessoas da primeira classe tiveram mais chances de sobreviver.

3.7 Facebook Prophet

O Facebook Prophet é uma ferramenta *open source*, criada pelo Facebook e disponível para as linguagens Python e R. O objetivo desta ferramenta é gerar previsões a partir de séries temporais, considerando fatores de sazonalidade e tendência. Segundo Taylor e Letham (2017a), o Facebook criou o Prophet com o objetivo de “tornar mais fácil para especialistas e não especialistas fazer previsões de alta qualidade que atendam à demanda.” A ferramenta vem sendo amplamente utilizada pelo Facebook e por diversos cientistas de dados para realizar previsões de forma simples e com resultados precisos.

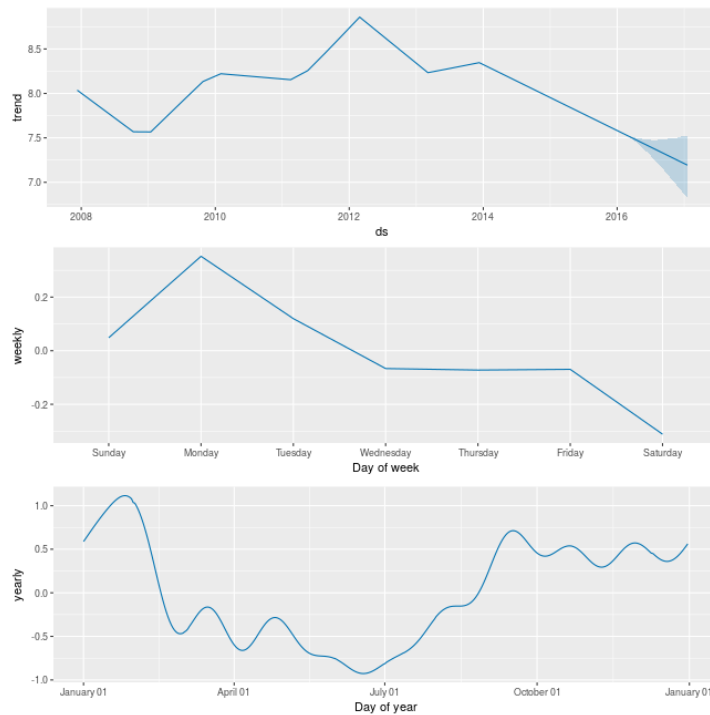
Para gerar previsões com séries temporais o Prophet possui vantagens como a possibilidade de gerar modelos a partir de dados diários, semanais ou mensais, considerar sazonalidades em “escala humana”, considerar importantes feriados no país, que podem afetar a predição dos dados e observar mudanças de tendência ao longo dos anos. O pacote Prophet inclui diferentes técnicas estatísticas de previsão, como ARIMA e suavização exponencial. (TAYLOR, LETHAM, 2017a).

As previsões do Prophet são personalizáveis, de maneira intuitiva, de forma que até iniciantes em modelos de predição possam utilizar a ferramenta de forma simples e eficaz. Segundo Taylor e Letham (2017a), nas configurações de ajustes do Prophet, “existem parâmetros de suavização para a sazonalidade que permitem ajustar a adequação dos ciclos históricos, bem como parâmetros de suavização para tendências que permitem ajustar com que agressividade devem seguir as alterações históricas das tendências.”

O Prophet é um modelo de regressão aditivo com quatro principais componentes: Uma tendência linear ou logística por partes das curvas de crescimento, onde ele é capaz de detectar automaticamente mudanças de tendência, um componente sazonal anual modelado usando a série de Fourier, um componente sazonal semanal usando variáveis fictícias e uma lista de feriados fornecida pelo usuário. (TAYLOR, LETHAM, 2017a).

Um exemplo sobre resultados gerados pelo Prophet é um teste feito com base na quantidade de visualizações na página do site Wikipedia no perfil do jogador de futebol americano Peyton Manning <https://en.wikipedia.org/wiki/Peyton_Manning>. Por se tratar de um jogador da liga de futebol americano dos EUA (NFL), é possível ver o componente sazonal forte das visualizações de seu perfil em períodos de temporada, ficando mais forte em semanas de finais e em anos em que ele chegou ao *Super Bowl* (final da NFL). A figura 15 mostra os componentes sazonais anuais, semanais e a curva de tendência deste teste realizado com a página do perfil do jogador. No primeiro gráfico é possível visualizar a tendência, onde nos períodos em que o atleta estava em alta é possível ver os picos da curva, enquanto quanto mais próximo ao final da carreira é possível visualizar uma queda no número de visualizações. A segunda curva mostra a sazonalidade semanal e pode-se verificar que nos domingos e segundas é onde há o maior registro de visitas, já que os jogos da NFL normalmente acontecem aos domingos. Na terceira curva, de sazonalidade anual, fica claro que o número de visitas ao perfil do jogador aumenta nos períodos de temporada e pós-temporada da NFL, que acontece entre o início de Setembro e o início de Fevereiro (TAYLOR, LETHAM, 2017a).

Figura 15 – Resultado do Prophet para visualizar os acessos ao perfil do jogador Peyton Manning no Wikipedia



Fonte: (TAYLOR, LETHAM, 2017a).

O Prophet utiliza um modelo de série temporal com três principais variáveis, tendência, sazonalidade e feriados. É possível verificar esses fatores na seguinte fórmula: $y(t) = g(t) + s(t) + h(t) + \varepsilon(t)$, onde $g(t)$ é a função de tendência, $s(t)$ representa alterações periódicas como sazonalidade semanal e anual, $h(t)$ representa os efeitos relacionados a feriados e $\varepsilon(t)$ representa o resíduo, não explicado pelos outros fatores. Para calcular a tendência ($g(t)$), o Prophet pode implementar dois modelos, um linear de uso mais simples, quando se tem uma taxa de crescimento constante e um de crescimento não linear e saturado, representado pela equação logística:

$$g(t) = \frac{C}{1 + \exp(-k(t - m))}$$

onde C é a capacidade de carga (valor máximo da curva), k é a taxa de crescimento (inclinação da curva) e m é um parâmetro de deslocamento. A sazonalidade ($s(t)$), fornece um modelo flexível de mudanças periódicas devido a sazonalidade semanal e mensal. O modelo anual de sazonalidade do Prophet se baseia na série de Fourier. Os feriados ($h(t)$), é uma lista inserida pelo usuário, onde o Prophet ao gerar as previsões poderá levar em consideração essas datas

específicas que afetam os outros fatores. Tudo o que não pode ser explicado por sazonalidade, tendência e feriados é o fator residual ($\epsilon(t)$) (TAYLOR, LETHAM, 2017b).

Para rodar um modelo de previsão utilizando o Prophet no Python, é necessário importar a biblioteca do Prophet e passar os principais parâmetros, como o modo de cálculo de sazonalidade (multiplicativo ou aditivo), o tipo de sazonalidade que vai ser analisado (diário, semanal ou anual), o número de pontos de mudança de tendência e sazonalidade (*changepoints*), a lista de feriados, entre outros parâmetros. Ao rodar o modelo de previsão, é gerado um *dataframe* onde, entre as principais informações, estão a data (*ds*), o valor da predição (*yhat*), a tendência (*trend*) e o coeficiente de sazonalidade anual (*yearly*), conforme mostra a figura 16.

Figura 16 – Valores de predição gerados pelo Prophet

```
coefs[['ds', 'trend', 'yhat', 'yearly']]
```

| | ds | trend | yhat | yearly |
|---|------------|---------------|---------------|-----------|
| 0 | 2015-01-02 | 804040.388487 | 749129.464758 | -0.068294 |
| 1 | 2015-02-02 | 796936.910805 | 702195.699950 | -0.118882 |
| 2 | 2015-03-02 | 790520.866448 | 769758.124933 | -0.026265 |
| 3 | 2015-04-02 | 783417.388679 | 749440.547500 | -0.043370 |
| 4 | 2015-05-02 | 776543.055355 | 779500.763681 | 0.003809 |
| 5 | 2015-06-02 | 769439.577369 | 738470.425200 | -0.040249 |

Fonte: Elaborado pelo autor

3.8 Paralelização dos Processos

Para rodar modelos de Machine Learning é necessário que a máquina realize muitos cálculos complexos, que se fossem feitos por um humano poderiam levar dias, ou até anos. Até mesmo para uma máquina, realizar esses cálculos é uma tarefa pesada e dependendo do tipo de modelo e da quantidade de dados inseridos, é um processo demorado. Supercomputadores são utilizados para rodar modelos complexos, principalmente para aplicações na área genética ou meteorológica.

A biblioteca Joblib do Python, permite paralelizar os modelos, de forma que todos os núcleos do processador da máquina sejam utilizados, para cálculos de matrizes ou iterações, por exemplo. Isso reduz bastante o tempo de geração das predições e deixa o processo mais otimizado. Para modelos simples a diferença é pequena, mas em grandes modelos, esta prática de programação distribuída é essencial para aproveitar o máximo do hardware e ganhar tempo.

3.9 Web Scraping

Um dos grandes desafios da Ciência de Dados é coletar os dados e muitas vezes os dados estão espalhados, de forma não organizada e seria quase impossível um ser humano de forma manual coletar toda essa informação, principalmente quando o assunto é big data. Segundo a definição do Data Science Academy, “web scraping é o ato de baixar automaticamente os dados de uma página web e extrair informações muito específicas dela. As informações extraídas podem ser armazenadas praticamente em qualquer lugar”. (DSA, 2018b) Web scraping vem sendo utilizado de forma ampla nos tempos atuais para diversas aplicações, entre elas legais e ilegais, o que acaba gerando uma “má fama” para este tipo de aplicação. Mas trata-se de uma ferramenta muito importante, desde que utilizada de forma legal e consciente, para os cientistas de dados conseguirem ter acesso aos dados para iniciar as etapas de mineração e análise.

Muitas empresas hoje em dia, criam robôs para coletar dados de redes sociais, por exemplo, e com isso entender melhor os usuários da sua marca, aplicando esses dados em ferramentas de Machine Learning. É possível também utilizar este tipo de ferramenta para encontrar artigos e pesquisas sobre uma determinada doença e depois coletar dados necessários para rodar modelos e chegar a novos tratamentos para esta doença. No Python existe duas bibliotecas bastante utilizadas para realizar extração de dados Web, o BeautifulSoup e o Selenium.

3.9.1 BeautifulSoup

A biblioteca BeautifulSoup, para Python, permite extrair dados de arquivos HTML e XML. Com esta ferramenta é possível acessar uma página Web e acessar o código em HTML para realizar uma extração de dados da página. Diversas opções permitem acessar cada parte do código HTML, realizar downloads, acessar links, entre outras coisas. O BeautifulSoup permite, por exemplo, monitorar uma página e saber quando que ela foi atualizada e assim baixar os dados mais recentes. Sempre que utilizado este tipo de ferramenta é importante tomar cuidado com os tempos de acesso à página Web, já que um robô pode realizar vários acessos num período de um segundo, o que pode derrubar a página, então é importante dar tempos seguros para novos acessos, sem prejudicar a página que está sendo visitada.

3.9.2 Selenium

Selenium é uma biblioteca *open source* disponível para diversas linguagens de programação, inclusive o Python, com o objetivo principal de realizar testes automatizados em páginas Web, simulando o usuário acessando o navegador e interagindo com a página. Para

funcionar, precisa de um *driver* do navegador (preferencialmente Google Chrome ou Mozilla Firefox) para realizar o acesso automatizado.

Selenium também pode ser utilizado para realizar a extração de dados de forma automatizada, simulando o acesso de um usuário comum, realizando marcações, cliques e downloads. Assim como o Beautiful Soup é importante tomar cuidados ao realizar acessos automatizados em páginas Web, respeitando tempos seguros de acesso, sem causar prejuízos aos donos da página. A ferramenta permite extrair dados de *dashboards* ou sites como o DataSUS, do Ministério da Saúde, para ter acesso à informação sobre uma determinada doença, por exemplo.

3.10 Power BI

Uma vez que passamos da etapa de análise dos dados é importante que o cientista de dados tenha uma boa forma de apresentar esta informação de forma clara para o expectador e para isso existe ferramentas como o Power BI, da Microsoft, que permite realizar a criação de *dashboards* interativos de forma que o usuário possa explorar os dados. É possível utilizar a ferramenta de forma gratuita, mas para uso profissional a Microsoft oferece planos de assinatura, muito utilizado por empresas.

O Power BI permite realizar a leitura de dados de um arquivo no formato CSV, ou um arquivo do Microsoft Excel, ou até mesmo ler dados direto de um banco de dados. As informações podem ser atualizadas de forma dinâmica e em tempo real, de forma que os interessados tenham sempre a informação mais recente e objetiva e pode ser acessado através de um painel Web ou por um aplicativo no *smartphone*, conforme mostra a figura 17.

Figura 17 – Microsoft Power BI



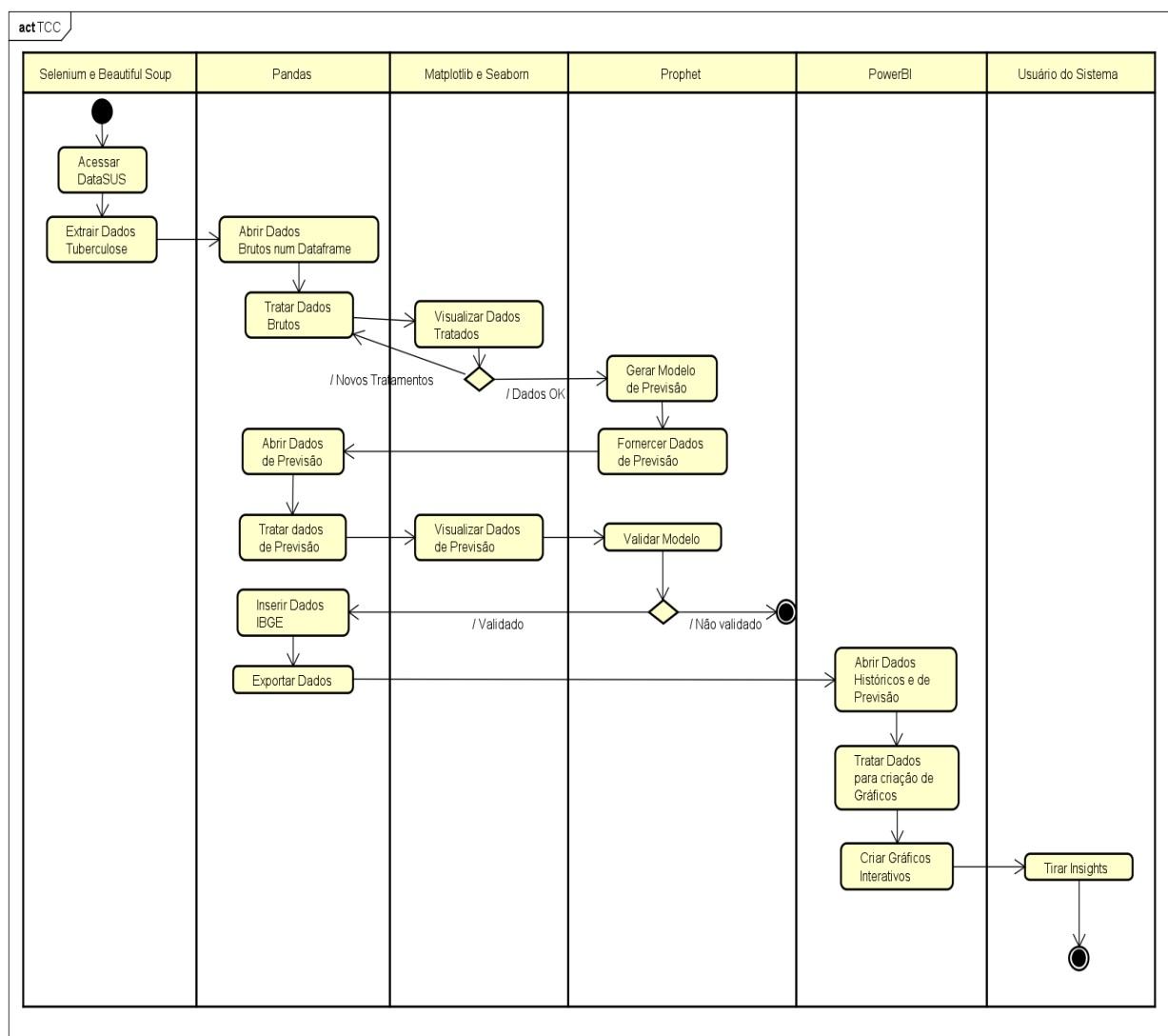
Fonte: (MICROSOFT, 2020)

4 ESTUDO DE CASO

Para prever o número de casos de tuberculose no Brasil, primeiramente foi necessário coletar dados do site do Ministério da Saúde, processar e preparar os dados para que pudesse ser gerado os modelos de previsão, através do Prophet, análise dos resultados e por fim a criação de um dashboard interativo no Power BI para que seja possível uma fácil leitura dos números encontrados. Todo conteúdo deste estudo de casos está disponível no GitHub, através do endereço < <https://github.com/pedroigorgrilo/tcc> >.

O diagrama de atividades abaixo (figura 18) apresenta todo o processo que será abordado neste capítulo, que tem o objetivo de extrair os dados, realizar as previsões e divulgar os resultados no dashboard interativo.

Figura 18 - Diagrama de Atividades



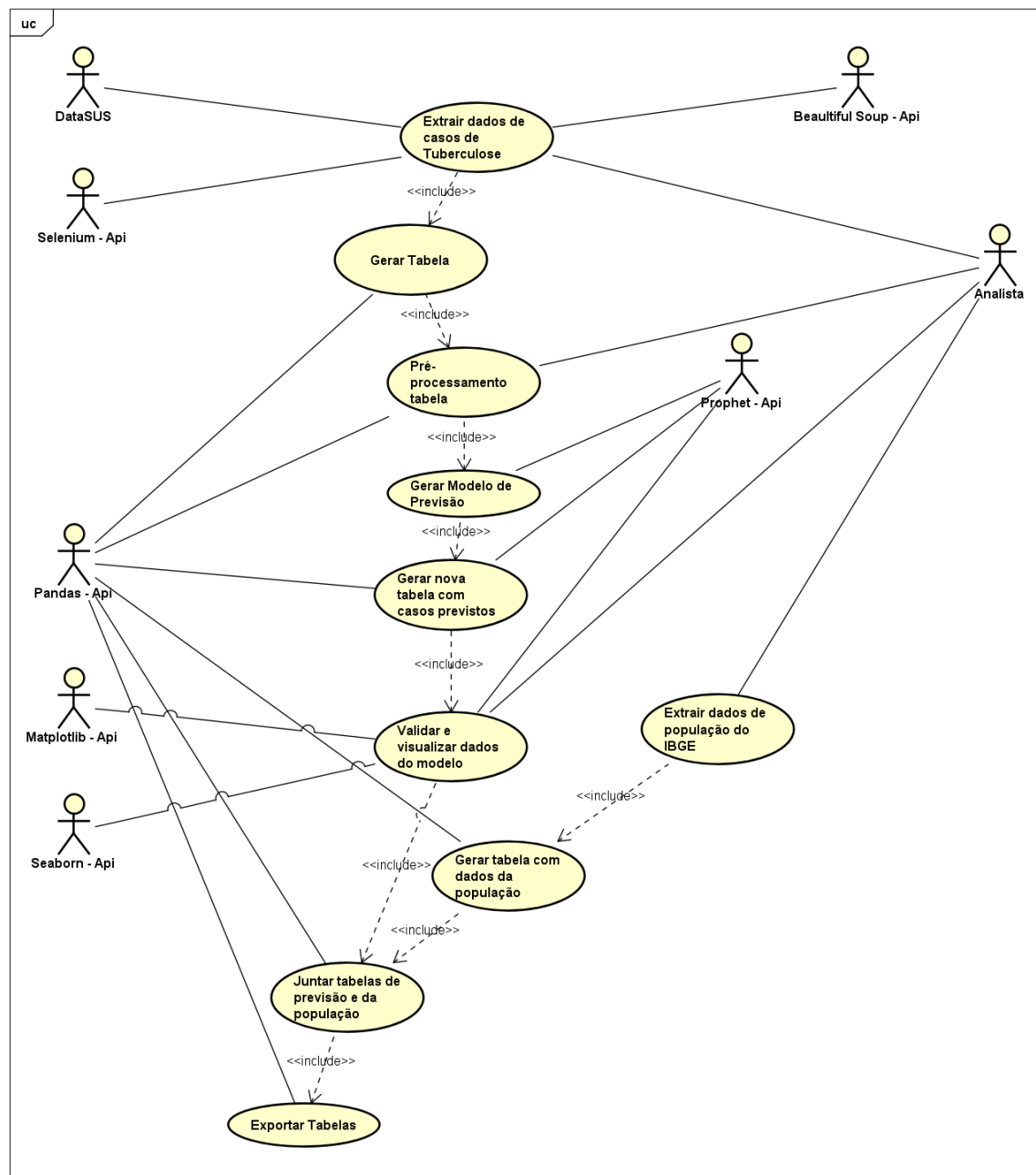
Fonte: Elaborado pelo autor

4.1 Diagramas de Casos de Uso e de Classes

Para descrever os programas utilizados neste projeto, foi elaborado um diagrama de Casos de Uso e um Diagrama de Classes.

A figura 19 apresenta o diagrama de Casos de Uso, onde todos os atores são ligados às suas respectivas tarefas no código de extração de dados, processamento e geração do modelo de previsão.

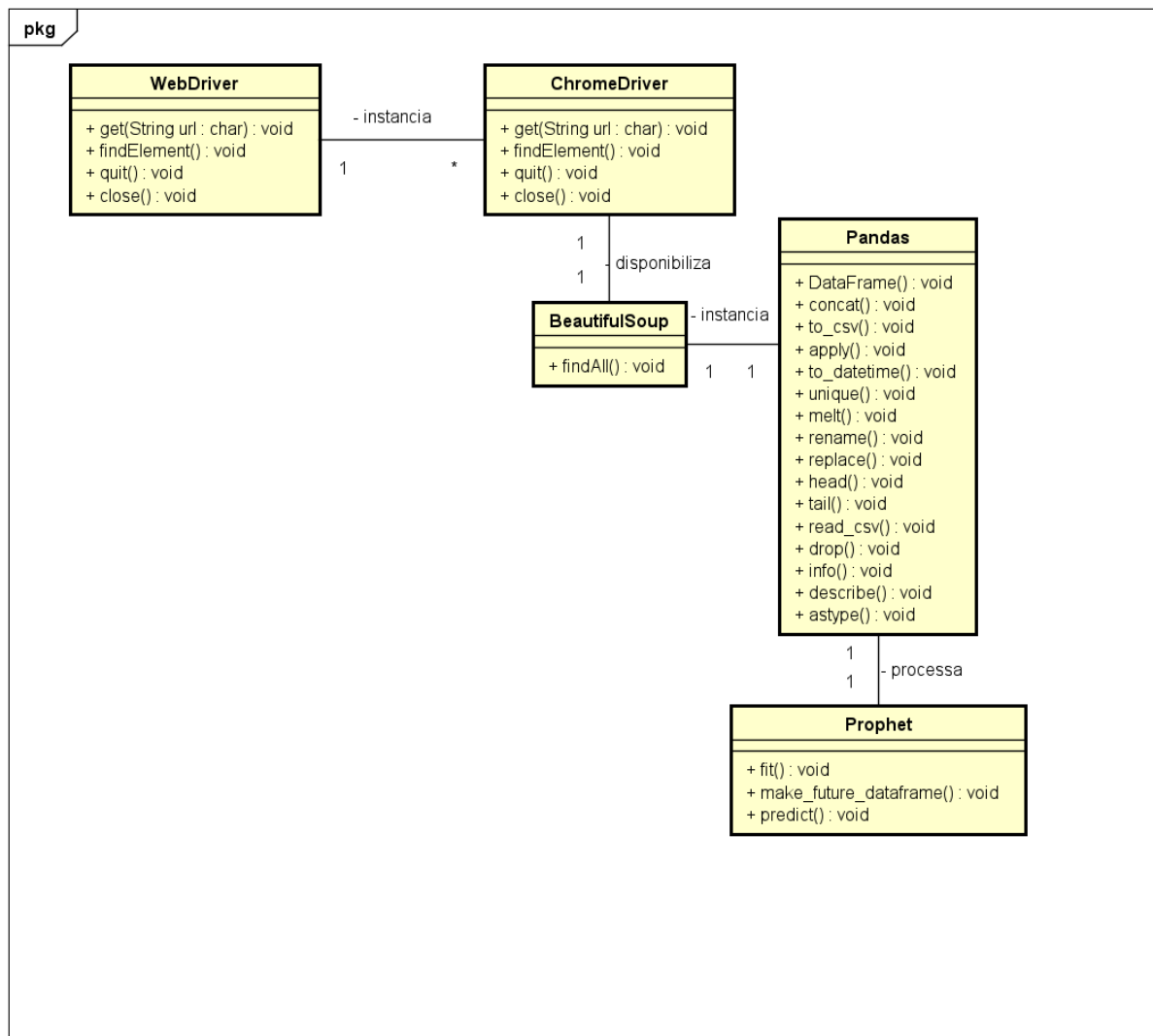
Figura 19 - Diagrama de Casos de Uso



Fonte: Elaborado pelo autor

A figura 20 apresenta o diagrama de Classes onde a classe WebDriver do Selenium é responsável por chamar a classe ChromeDriver (driver do Google Chrome para o Selenium utilizado neste projeto) e que no final deixará os dados disponíveis para a classe BeautifulSoup do Python, que realiza o acesso via html, através da função “*findall()*” e armazena em um dataframe do Pandas, onde será possível realizar toda a manipulação dos dados coletados. No Pandas diversas funções são utilizadas para o pré-processamento e depois o dataframe é utilizado no Prophet, onde será possível utilizar funções para rodar o modelo e gerar as previsões.

Figura 20 - Diagrama de Classes



Fonte: Elaborado pelo autor

4.2 DataSUS

O DataSUS é um sistema do Ministério da Saúde do Brasil que disponibiliza dados históricos de diversas doenças no país. Através deste site é possível coletar toda a informação disponível pelo governo sobre a tuberculose, conforme ilustrado na figura 21 <<http://tabnet.datasus.gov.br/cgi/tabcgi.exe?sinannet/cnv/tubercbr.def>>, desde 2001, entre elas, número de casos por estado, número de mortes, tipo do tratamento, comorbidade dos pacientes, tipo da doença, entre outros. Trata-se de uma fonte segura dos dados para que as previsões sejam as melhores possíveis.

Figura 21 – Dados da Tuberculose no Brasil (Sistema DataSUS)

Fonte: Elaborado pelo autor

Como citado acima, os dados estão disponíveis desde 2001 até 2019, mas anos passados continuam tendo números atualizados de acordo com os registros feitos por cada secretaria de saúde dos estados, então números de 2016 a 2019 (principalmente) ainda podem sofrer alterações ao longo dos próximos anos, enquanto dados até 2015 não devem mais ser atualizados. Para o modelo de previsão foi utilizado dados de 2001 a 2018 com o objetivo de prever o número de casos em 2019 e comparar com os dados disponibilizados pelo DataSUS.

4.2.1 Coleta dos Dados

A primeira parte de um processo de previsão é coletar os dados, neste caso na fonte de dados do DataSUS. O sistema permite realizar filtros e baixar as informações necessárias num arquivo formato CSV (valores separados por vírgula), porém não permite baixar de uma só vez os dados de números de casos por estado, mês e ano, sendo necessário pegar no mínimo doze arquivos diferentes, um para cada mês, contendo o número de casos por ano (naquele mês) para cada estado. Além disso, os dados são atualizados dinamicamente no sistema, então toda vez que fosse preciso atualizar os dados de previsão seria necessário buscar manualmente os dados

novamente. Para isso foi criado um código de Web scraping (extração de dados), utilizando as bibliotecas Selenium e BeautifulSoup no Python. Para escrever o código foi utilizado o Jupyter Notebook, disponível no pacote Anaconda e realizado a instalação de todas as bibliotecas necessárias para extrair os dados e criação do dataframe.

O Selenium permite realizar os filtros disponíveis no site dentro do código e, simulando uma interação humana no sistema, consegue baixar os dados automaticamente e a cada interação, utilizando a biblioteca Pandas os dados são salvos em um dataframe, para que possam ser tratados posteriormente. Com este robô é possível inserir novos filtros, caso seja necessário no futuro e até mesmo buscar dados de outras doenças disponíveis no sistema.

Além de selecionar os estados, anos e meses disponíveis foi necessário aplicar um filtro de “Tipo de Entrada”, selecionando apenas as opções “CASO NOVO”, “NÃO SABE” e “PÓS ÓBITO”, conforme é aconselhado pelo próprio Ministério da Saúde no site: “Para cálculo da incidência selecione o local de residência e as seguintes categorias da variável “Tipo de entrada”: caso novo, não sabe e pós óbito.” (BRASIL, 2020b). O sistema também faz outras considerações, como o aumento do número de casos de 2013 a 2016 pode ser atribuído ao processo de implementação e adesão ao SITE-TB (sistema com as informações sobre a tuberculose), que ocorreu em 2013, não sendo necessariamente um aumento real na incidência (BRASIL, 2020b). Este tipo de informação é importante para geração do modelo, principalmente porque o Prophet utiliza a série histórica e curvas de tendência para gerar as previsões. Ao final do processo de extração de dados é possível visualizar os dados brutos através do Pandas, conforme mostra a figura 22.

Figura 22 – Dataframe com números de novos casos de Tuberculose por Estado

| | "UF de notificação" | "2001" | "2002" | "2003" | "2004" | "2005" | "2006" | "2007" | "2008" | "2009" | "2010" | "2011" | "2012" | "2013" | "2014" | "2015" | "2016" | "2017" | "2018" |
|---|------------------------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|
| 0 | "11 Rondônia" | 56 | 52 | 48 | 29 | 33 | 27 | 35 | 45 | 37 | 44 | 30 | 56 | 45 | 56 | 43 | 50 | 47 | 42 |
| 1 | "12 Acre" | 31 | 40 | 32 | 29 | 34 | 37 | 17 | 18 | 37 | 22 | 29 | 40 | 27 | 54 | 22 | 34 | 31 | 23 |
| 2 | "13 Amazonas" | 208 | 184 | 190 | 166 | 162 | 153 | 179 | 179 | 184 | 204 | 167 | 173 | 247 | 229 | 229 | 219 | 247 | 255 |
| 3 | "14 Roraima" | 11 | 7 | 12 | 13 | 8 | 8 | 9 | 12 | 16 | 7 | 7 | 10 | 9 | 12 | 19 | 13 | 16 | 16 |
| 4 | "15 Pará" | 269 | 270 | 280 | 312 | 287 | 261 | 296 | 288 | 288 | 301 | 297 | 279 | 323 | 312 | 281 | 296 | 301 | 300 |

Fonte: Elaborado pelo autor

4.3 Tratamento dos dados brutos

A figura 22 mostrou os dados brutos e nele é possível visualizar que as colunas e linhas estão com aspas, então o passo após a coleta de dados é tratar o dataframe e conhecer melhor os dados disponíveis, para que seja possível aplicar as melhores configurações e rodar o modelo

de predição. Para retirar as aspas das colunas e do nome dos estados nas linhas, é possível utilizar uma função com o comando “*strip()*”, disponível no Pandas e assim deixar os nomes necessários para trabalhar com o dataframe. Colunas são renomadas para seguir o mesmo padrão. Um outro problema no dataframe mostrado na figura 22 é os anos nas colunas e, para rodar o modelo de previsão, é necessário que haja uma coluna única para o ano e uma outra para os meses, assim, em cada linha teremos uma UF, um ano, um mês e o número de casos naquele período. Para realizar essa transformação na tabela é utilizado a função “*melt()*” disponível no Pandas, que leva as colunas para linhas, tendo assim uma tabela mais organizada e preparada para o modelo, conforme apresentado na figura 23.

Figura 23 – Tratamento do dataframe com número de casos de Tuberculose por Estado

| | ESTADO | MES | ANO | CASOS |
|---|-------------|---------|------|-------|
| 0 | 11 Rondônia | Janeiro | 2001 | 56 |
| 1 | 12 Acre | Janeiro | 2001 | 31 |
| 2 | 13 Amazonas | Janeiro | 2001 | 208 |
| 3 | 14 Roraima | Janeiro | 2001 | 11 |
| 4 | 15 Pará | Janeiro | 2001 | 269 |

Fonte: Elaborado pelo autor

Ainda é necessário um ajuste no dataframe para que o Prophet possa ler os dados, na tabela da figura 23, ano e mês estão em colunas separadas e o mês está por extenso, então é necessário a criação de uma coluna “DATA”, no formato de *datetime* reconhecido pelo Python. Para isso, primeiro é criado um dicionário para transformar o nome do mês no número do mês e depois as colunas de ano e mês são concatenadas, criando uma coluna no formato “YYYYMM” (formato de ano mês, exemplo: 202001). Após a criação desta coluna é possível, com a função “*to_datetime()*” do Pandas, gerar a coluna data no formato de *datetime* reconhecido pelo Prophet “YYYY-MM-DD” (exemplo: 2020-01-01). Nas linhas da tabela é possível observar um código numérico ao lado do nome dos estados, que são identificações que o Ministério da Saúde utiliza para cada UF. Então, para melhorar a leitura dos dados é criado um dicionário dentro de uma função para que seja criado uma coluna com a abreviação da UF. Assim é possível visualizar um dataframe mais limpo e de fácil visualização e manipulação, como mostra a figura 24.

Figura 24 – Dataframe em tratamento para melhor manipulação dos dados

| | ESTADO | MES | ANO | CASOS | NO_AM | DATA | UF |
|---|-------------|-----|------|-------|--------|------------|----|
| 0 | 11 Rondônia | 1 | 2001 | 56 | 200101 | 2001-01-01 | RO |
| 1 | 12 Acre | 1 | 2001 | 31 | 200101 | 2001-01-01 | AC |
| 2 | 13 Amazonas | 1 | 2001 | 208 | 200101 | 2001-01-01 | AM |
| 3 | 14 Roraima | 1 | 2001 | 11 | 200101 | 2001-01-01 | RR |
| 4 | 15 Pará | 1 | 2001 | 269 | 200101 | 2001-01-01 | PA |

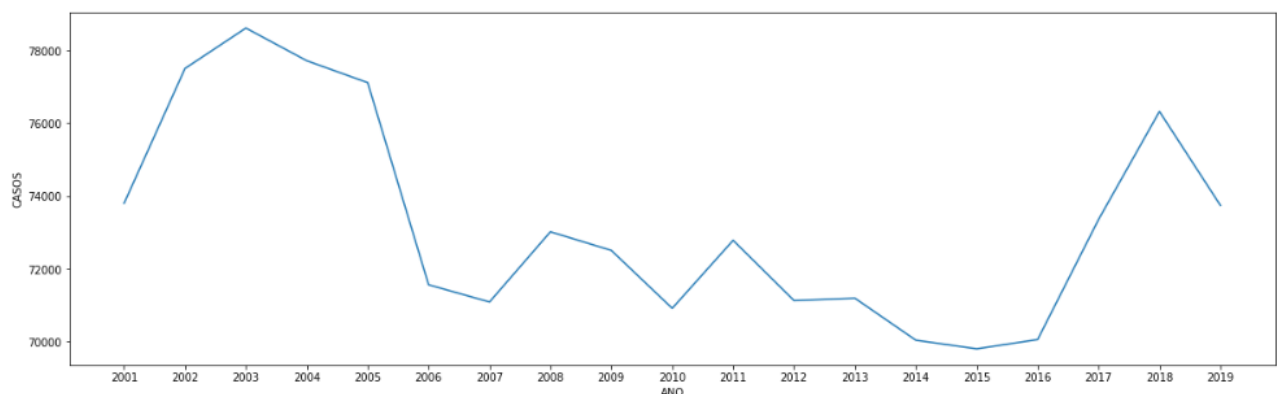
Fonte: Elaborado pelo autor

4.4 Criação do Modelo de Previsão com o Prophet

Com os dados do número de novos casos já tratados é necessário entendê-los para que sejam realizados os melhores filtros ao inseri-los no modelo. Conforme informado no tópico 4.1.1, o Ministério da Saúde informa que houve um crescimento no número de casos entre 2013 e 2016 devido a adesão ao sistema e melhor notificação dos casos, diferente do que ocorria entre 2001 e 2012. Porém, ao analisar o gráfico com o número de novos casos no Brasil ao longo dos anos é possível visualizar uma queda entre 2013 e 2016, assim como houve um aumento significativo entre 2001 e 2003, conforme mostra a figura 25, gerada através das ferramentas Matplotlib e Seaborn do Python.

Figura 25 – Avanço do número de novos casos da tuberculose no Brasil ao longo dos anos

```
plt.figure(figsize=(20,6))
sns.lineplot(data = base_agrupada,x='ANO',y='CASOS');
```



Fonte: Elaborado pelo autor

Como o objetivo inicial do modelo de predição é obter os números de 2019 é importante realizar diversos testes para entender qual a melhor entrada para o modelo. Entre os testes estão os dados de entrada separados entre 2001-2018, 2010-2018 e 2014-2018. Os dados de 2019 não

serão utilizados nos modelos, já que o objetivo é comparar a assertividade entre os dados previstos e os dados reais, apesar de existir uma defasagem entre a informação disponível no momento e atualizações que ainda serão feitas nos próximos anos, principalmente para 2019, conforme informa o próprio Ministério da Saúde. O modelo irá prever o número de casos por estado e por mês, de forma que os dados possam auxiliar as secretarias de saúde de cada estado a entender o crescimento ou diminuição do número de novos casos.

O Prophet reconhece apenas dataframes com data (coluna chamada “ds”) e uma coluna chama de “y” que contém os valores reais que serão utilizados como entrada para o modelo. Então é preciso fazer o último ajuste no dataframe para realizar a leitura dos dados, conforme apresenta a figura 26. A coluna UF é mantida apenas para realizar a interação por estado.

Figura 26 – Dataframe pronto para rodar o Prophet

| | UF | ds | y |
|------|----|------------|-----|
| 2916 | RO | 2010-01-01 | 44 |
| 2917 | AC | 2010-01-01 | 22 |
| 2918 | AM | 2010-01-01 | 204 |
| 2919 | RR | 2010-01-01 | 7 |
| 2920 | PA | 2010-01-01 | 301 |

Fonte: Elaborado pelo autor

Para otimizar o modelo e deixar o processamento dos dados, o modelo é colocado numa função chamada “*prophet_uf*” e o processo é realizado utilizando programação distribuída com a ferramenta Jobs para Python, assim o modelo para cada UF roda em paralelo utilizando todos os núcleos disponíveis do processador. No final os dados são concatenados em dataframe do Pandas. Num notebook com quatro núcleos e um processador Intel Core I5 vPro 8º geração, o processo chegou a rodar 5x mais rápido do que rodando em sequência utilizando apenas um núcleo. Aplicar modelos de predição utilizando programação distribuída é fundamental para otimizar os processos e obter o melhor desempenho com o hardware disponível.

Nas configurações do Prophet foi utilizado o modo de sazonalidade aditivo, para obter um melhor resultado e desativado a sazonalidade diária e semanal, já que os dados disponíveis são mensais, então o principal objetivo é calcular a sazonalidade anual. Para variações de tendência existe o parâmetro “*changepoint_prior_scale*” que é de extrema importância para detecção deste tipo de variação ao longo do tempo. Para escolher o melhor valor para este

parâmetro, foi utilizado a ferramenta de “*cross validation*” do Prophet para verificar, através do RMSE a melhor configuração para cada UF, ao invés de utilizar apenas um valor para todas as UFs, assim entendendo as particularidades de cada série. É criado também um dataframe de previsão utilizando a ferramenta “*make_future_dataframe*” do Prophet de forma que teremos um dataframe com os valores reais e previsões com os dados de entrada e valores de previsões para datas futuras. Na tabela com valores reais até 2018 é criado também uma coluna chamada “resíduo” que contém os valores residuais que não são explicados por fatores de sazonalidade e tendência. Na configuração de tipo de crescimento (*growth*) foi utilizado o método logístico e criado as colunas “*cap*” e “*floor*” no dataframe, para definir valores máximos e mínimos, respectivamente, a fim de evitar valores negativos de número de casos. Foi definido um valor 0 para mínimo e 3000 para máximo, por mês.

Ao gerar uma predição utilizando o Prophet, é gerado um dataframe que pode ser visualizado pelo Pandas, contendo as informações de data (*ds*), do valor real (*y*), o valor previsto (*yhat*), o coeficiente de sazonalidade anual (*yearly*), a tendência (*trend*) e outras informações como o intervalo dos valores previstos (*yhat_lower* e *yhat_upper*) e os valores de intervalo da tendência (*trend_lower* e *trend_upper*). O intervalo entre o *yhat_lower* e o *yhat_upper*, possui 80% de confiança, levando em consideração os padrões de sazonalidade e tendência da série histórica. Na base com valores previstos e sem dados de entrada, não vai existir a coluna “*y*”, já que os dados reais não são conhecidos. A figura 27 mostra um dataframe criado após aplicação do modelo Prophet.

Figura 27 – Dataframe gerado após aplicação do Prophet

| | ds | uf | y | yhat | yhat_lower | yhat_upper | trend | trend_lower | trend_upper | yearly | residuo |
|-----|------------|-----|-----|-----------|------------|------------|-----------|-------------|-------------|-----------|-----------|
| 0 | 2014-01-01 | RO | 56 | 53.315840 | 48.833817 | 57.722895 | 53.583912 | 53.583912 | 53.583912 | -0.268072 | 2.684160 |
| 1 | 2014-02-01 | RO | 47 | 47.968022 | 43.546321 | 52.538947 | 43.604592 | 43.604592 | 43.604592 | 4.363430 | -0.968022 |
| 2 | 2014-03-01 | RO | 38 | 41.786605 | 37.245260 | 46.275606 | 34.591013 | 34.591013 | 34.591013 | 7.195592 | -3.786605 |
| 3 | 2014-04-01 | RO | 51 | 46.778009 | 42.181455 | 51.197614 | 36.750530 | 36.750530 | 36.750530 | 10.027479 | 4.221991 |
| 4 | 2014-05-01 | RO | 46 | 45.013314 | 40.908581 | 49.557755 | 38.840386 | 38.840386 | 38.840386 | 6.172929 | 0.986686 |
| ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| 55 | 2018-08-01 | DF | 38 | 37.599911 | 35.055420 | 40.190320 | 28.672608 | 28.672608 | 28.672608 | 8.927303 | 0.400089 |
| 56 | 2018-09-01 | DF | 29 | 34.550669 | 31.824712 | 37.441580 | 27.692047 | 27.692047 | 27.692047 | 6.858623 | -5.550669 |
| 57 | 2018-10-01 | DF | 37 | 33.832087 | 31.035803 | 36.571241 | 26.743116 | 26.743116 | 26.743116 | 7.088971 | 3.167913 |
| 58 | 2018-11-01 | DF | 35 | 33.466342 | 30.804272 | 36.234945 | 25.762555 | 25.762555 | 25.762555 | 7.703788 | 1.533658 |
| 59 | 2018-12-01 | DF | 24 | 25.688207 | 22.789630 | 28.261819 | 24.813624 | 24.813624 | 24.813624 | 0.874583 | -1.688207 |

Fonte: Elaborado pelo autor

4.4.1 Resultados das previsões com diferentes períodos de entrada

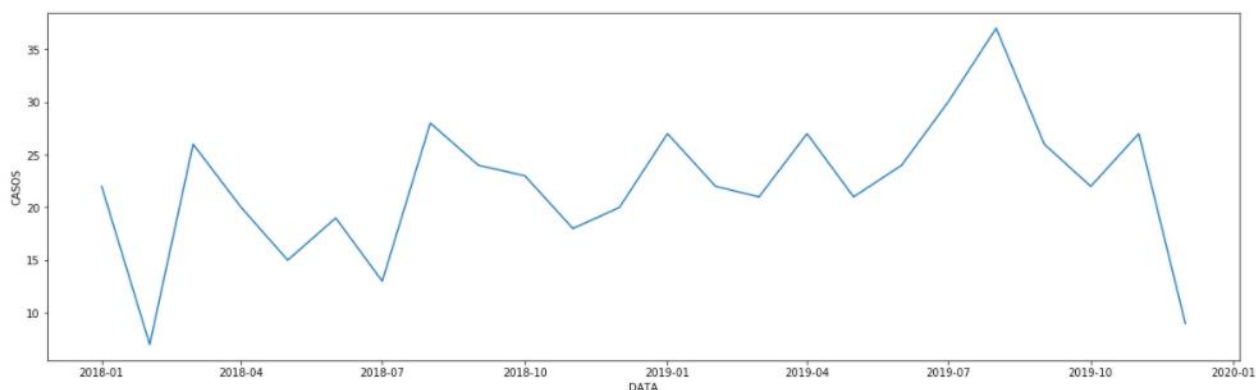
No primeiro teste foi utilizado dados de 2001 a 2018, que apresentam bastante variações ao longo dos anos, conforme mostrado na figura 25. Entre a previsão de 2018 com os dados reais de 2018 é possível encontrar um erro de 3,16%, em 2019 o erro é de 2,29% e utilizando a ferramenta MAPE (*mean absolute percentage error*) ou média percentual absoluta do erro, do Prophet é apresentado um valor de 12,5%. Na previsão para 2019, os maiores erros se encontram nos estados de Amapá (38,94%) e Acre (18,31%)

No segundo teste foi utilizado dados de 2010 a 2018, desconsiderando assim o elevado número de casos existentes no início dos anos 2000. Neste cenário, o erro em 2018 foi de 1,05% no total e 1,42% em 2019. No MAPE o erro foi de 10,65%, enquanto as UFs com maiores erros em 2019 foram o Rio Grande do Norte (22,7%) e o Amapá (17,03%).

No terceiro teste, com dados de 2014 a 2018, o erro em 2018 de 0,68% e em 2019 de 2,36%. No MAPE o erro foi de 9,51% e os estados com maiores erros em 2019 foram Rio Grande do Norte (23,29%) e Paraná (22,43%). Em São Paulo e Rio Janeiro, estados com os maiores número de casos históricos de tuberculose, foi encontrado melhores percentuais de erro, 0,33% e 4,37%, respectivamente. Santa Catarina também apresentou um ótimo resultado, com erro de 0,88%. Mostrando que essas localidades possuem uma sazonalidade e tendência mais bem definida ao longo do tempo e um maior registro de número de casos ajuda um modelo de previsão que utiliza séries temporais como principal parâmetro.

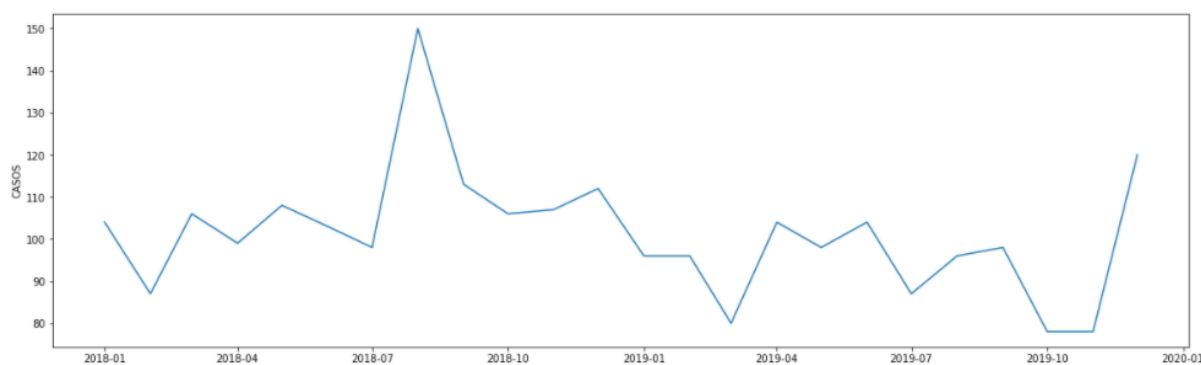
O segundo teste (2010 à 2018) foi o mais bem-sucedido, mas em todos os resultados foi possível verificar um erro maior nos estados do Amapá e Rio Grande do Norte. Porém, observando os gráficos históricos das figuras 28 e 29, e na figura 23 (Brasil), é possível visualizar que há uma forte redução no número de casos em 2019, principalmente nos últimos meses do ano, indicando que ainda há uma possível subnotificação dos casos em algumas UFs, como AP e RN. Além disso, Estados com menor número absoluto de casos, mesmo uma variação pequena acaba trazendo um impacto maior no erro percentual.

Figura 28 – Forte redução do número de casos no final de 2019 no AP



Fonte: Elaborado pelo autor

Figura 29 – Forte redução do número de casos em 2019 no RN

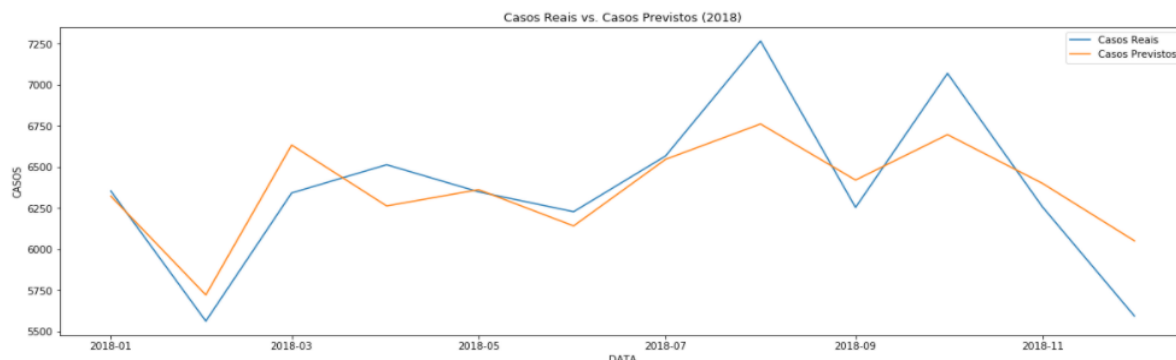


Fonte: Elaborado pelo autor

Então, olhando os erros no ano de 2018, que possui uma notificação mais próxima da realidade e os dados foram inseridos no modelo, os maiores erros (ou resíduos) encontrados foram no Mato Grosso do Sul (8,63%) e Rondônia (7,07%), enquanto os estados do Amapá (4,22%) e Rio Grande Norte (3,81%), apresentaram erros muito menores se comparado a previsão de 2019. Rio de Janeiro (0,76%) e São Paulo (0,43%), estados com maior número de casos, também tiveram valores precisos, com erro de seis a trinta casos em valores absolutos de 11mil a 18mil casos que os dois estados tiveram em 2018, respectivamente. Analisando o MAPE por estado, nas previsões de 2010 a 2018, os maiores erros se encontram nos estados do Roraima (26,27%) e Amapá (20,85%), enquanto os melhores resultados foram encontrados nos estados em Minas Gerais (5,02%), São Paulo (4,28%). Na figura 30 é possível observar a curva real de casos em 2018 (em azul) e a curva de previsão (em laranja), neste caso dá para observar um maior erro nos meses de Agosto e Setembro. Na figura 31 é possível observar a mesma

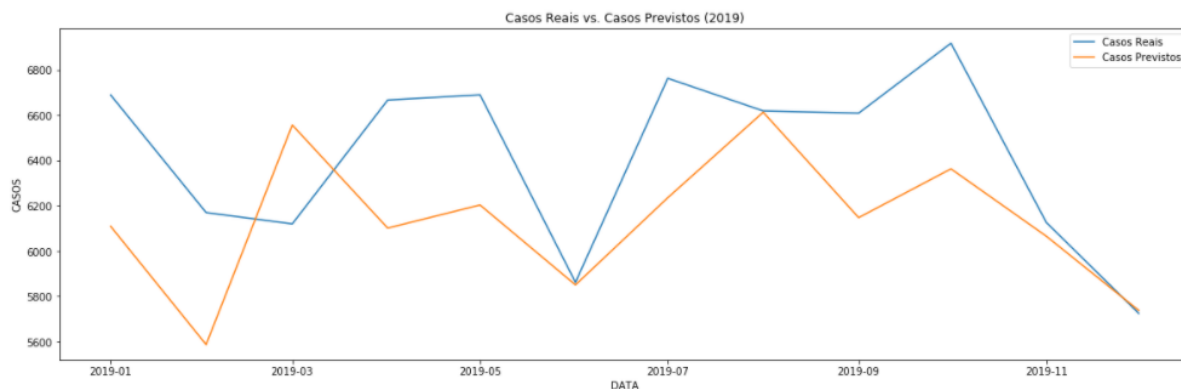
comparação, porém para 2019. Na figura 32 é plotado os valores reais e valores previstos de 2010 a 2018, período utilizado para previsão do número de novos casos.

Figura 30 – Casos Reais vs. Casos Previstos em 2018



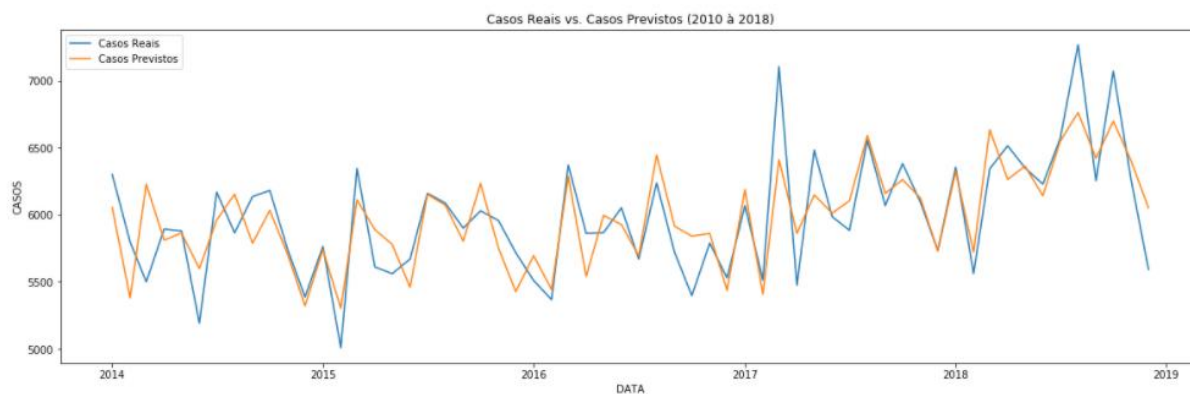
Fonte: Elaborado pelo autor

Figura 31 – Casos Reais vs. Casos Previstos em 2019



Fonte: Elaborado pelo autor

Figura 32 – Casos Reais vs Casos Previstos de 2010 a 2018

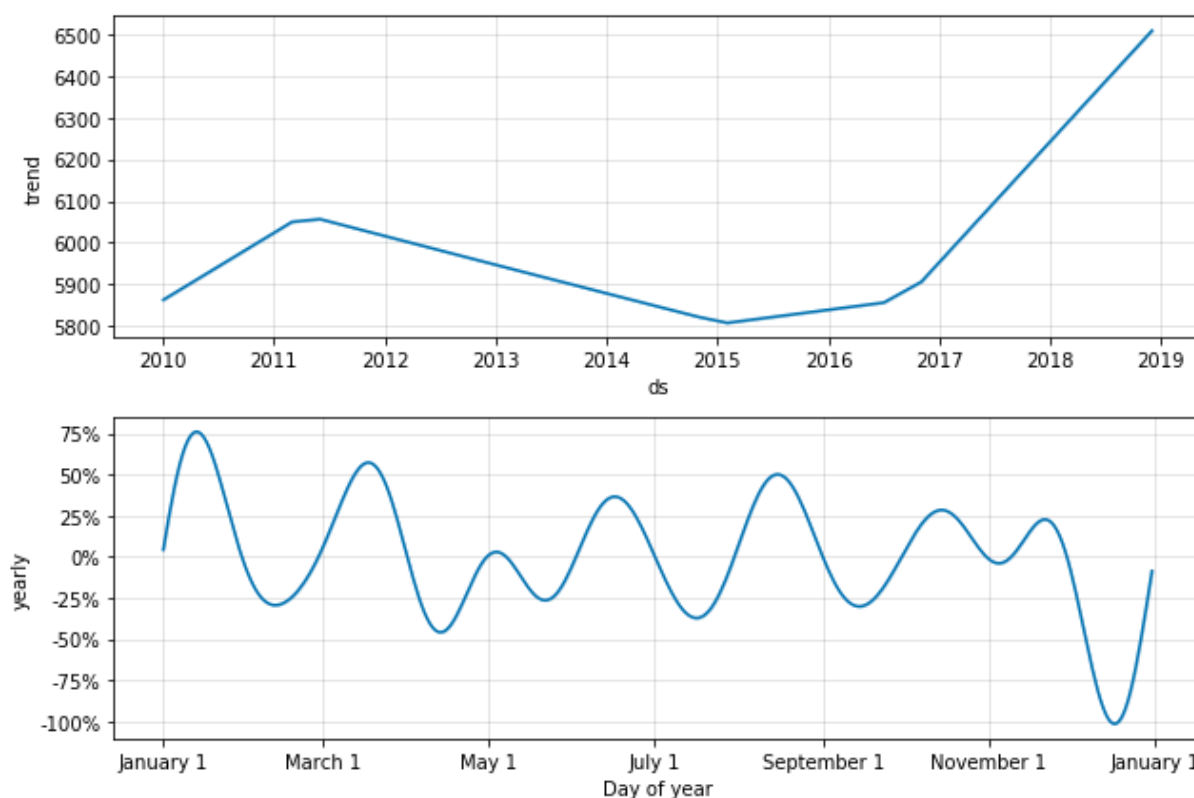


Fonte: Elaborado pelo autor

Para visualização dos dados, foi gerado também uma previsão para os anos de 2020, 2021 e 2022, utilizando os dados de 2010 a 2018. Porém, não foi considerado nessas previsões a pandemia do COVID-19, que pode influenciar no número de novos casos (diminuir) de tuberculose, devido aos isolamentos sociais e questões de higiene adotada pela população em geral.

O Prophet também possui uma ferramenta para mostrar as curvas de sazonalidade e tendência (figura 33) dos dados ao longo do período que foi usado como entrada para o modelo. Essas informações são muito importantes para entender como se comportam os dados ao longo dos meses do ano (sazonalidade) e ao longo dos anos (tendência). Na curva de tendência (*trend*), no Brasil, é observado um crescimento do número de casos em 2016, crescendo bastante até o final de 2018. Já na curva de sazonalidade é observado uma queda de notificações nos meses de Fevereiro, Abril e Julho, enquanto é visualizado um aumento de notificações nos meses de Janeiro, Março, Junho e Agosto.

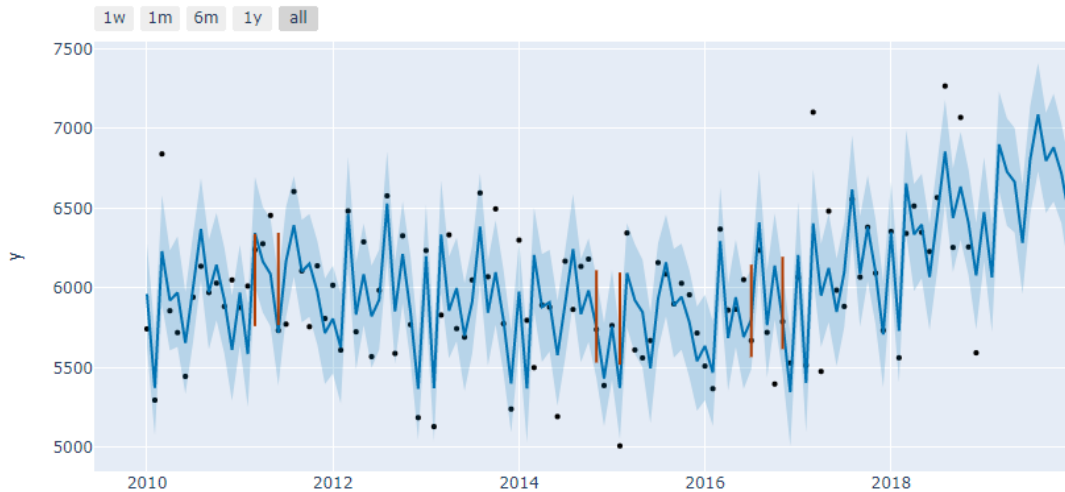
Figura 33 – Curvas de Tendência e Sazonalidade gerados pelo Prophet no período de 2010 a 2018 no Brasil



Fonte: Elaborado pelo autor

No gráfico da figura 34 é observado todas as mudanças de tendência (*changepoints*), ao longo do período inserido na entrada, que o Prophet enxergou, com os parâmetros configurados. Foram 6 mudanças detectadas ao todo.

Figura 34 – Changepoints (Mudanças de tendência detectadas pelo Prophet)

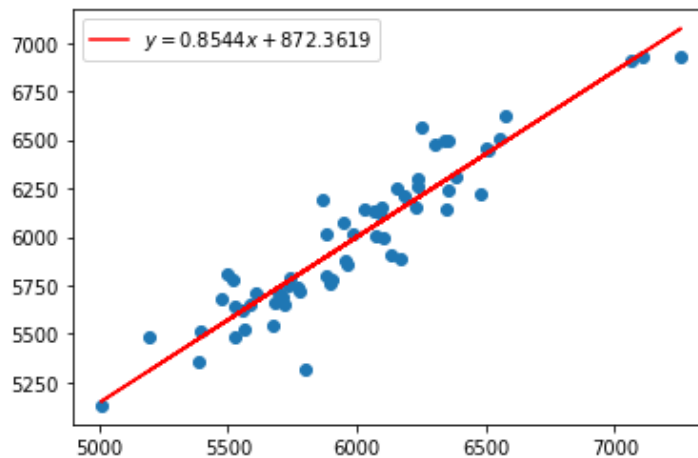


Fonte: Elaborado pelo autor

4.4.2 Regressão Linear para validação dos resultados

Para validar os resultados, já foi utilizado o cálculo do MAPE, que é a média percentual do erro absoluto, mas também é possível verificar o R^2 . Conforme explicado no capítulo 3, o R^2 é igual a variação explicada dividido pela variação total, onde quanto mais próximo de 1 melhor o resultado e quanto mais próximo de zero, pior o resultado. Utilizando a regressão linear é possível ver graficamente como se comporta a dispersão de casos reais e os casos previstos e verificar o R^2 . A figura 35 mostra o gráfico com uma regressão linear dos casos previstos (reta) e os casos reais (dispersão). O R^2 encontrado foi de 0,86, o que é um bom resultado.

Figura 35 – Regressão Linear (Casos Previstos vs Casos Reais)



Fonte: Elaborado pelo autor

4.5 Casos por 100mil Habitantes

Olhar somente o número de casos por estado pode gerar a falsa impressão de que estados como São Paulo tem um número muito alto e estados como Amazonas tem um número muito baixo. Então a melhor forma de visualizar os números é comparando com a população de cada região. Para isso foi solicitado ao IBGE, através da plataforma de transparência do Governo Federal (*e-sic*), uma base de dados com as projeções da população por estado e por mês, de 2001 até 2030. Para ler esses dados foi criado um código, em Python, para preparar o dataframe e juntar com o dataframe gerado com o de previsões e de casos reais. Assim, será possível, na etapa de visualização dos dados, interpretar melhor a situação de cada estado. Foi escolhido a opção de 100mil habitantes para que os números fiquem mais compreensíveis, já que existe uma grande diferença de população entre os estados da Região Sudeste e Região Norte, por exemplo.

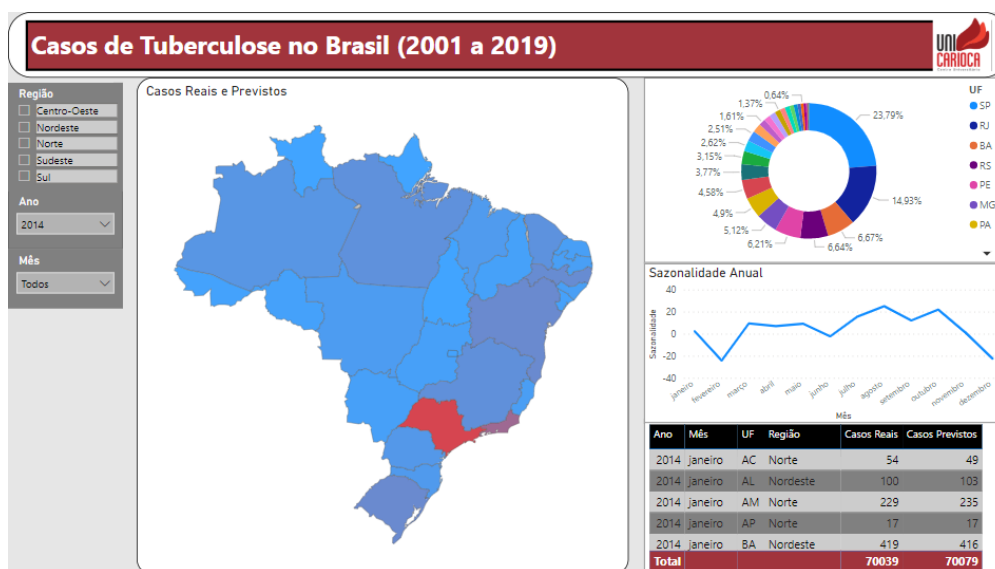
4.6 Visualização dos Dados

Para visualização dos dados gerados pelo modelo foi utilizado a ferramenta Microsoft PowerBI e criado um dashboard para apresentar a informação de forma interativa e de simples compreensão. O PowerBI é uma ótima ferramenta para criação de dashboards e possibilita que o usuário possa selecionar os filtros e visualizar o que tem interesse, interagir com os dados, diferente de uma apresentação no Power Point, por exemplo.

Na primeira aba de visualização do PowerBI é possível visualizar os dados reais e previstos pelo Prophet entre 2001 e 2019. Entre 2001 e 2013 será apresentado apenas os dados reais, já que esse período não foi inserido no modelo, porém de 2010 a 2019 será possível

realizar a comparação de casos reais com os dados previstos. Em 2019, especificamente, foi feito uma junção dos dataframes com dados reais e o gerado pelo modelo, já que o ano de 2019 não foi inserido no modelo. Nesta primeira visualização, como mostra a figura 36, é apresentado um mapa do Brasil, dividido por estados e colorido de acordo com a quantidade de casos naquele estado, naquele período selecionado. No canto superior esquerdo existem três filtros, onde é possível selecionar uma região específica do país, o ano e um mês de interesse. No canto superior direito é apresentado um gráfico de rosca mostrando a proporção dos estados de acordo com o filtro selecionado. Abaixo, o gráfico de sazonalidade anual, onde é possível ver como se comporta cada estado, região ou o país inteiro, ao longo do período selecionado, mês a mês. No canto inferior direito, uma tabela com os dados apresentados em detalhe.

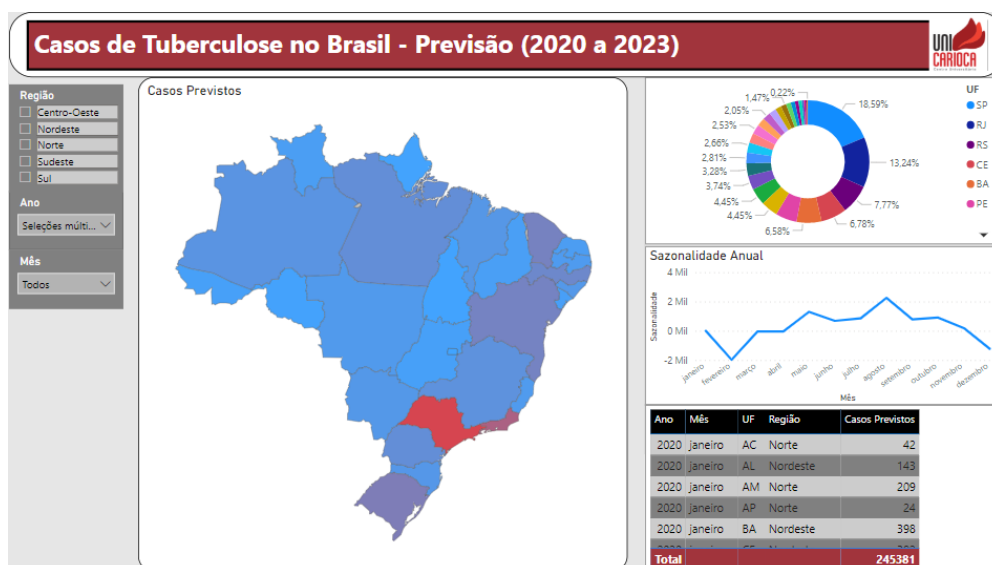
Figura 36 – Dashboard Número de casos absolutos



Fonte: Elaborado pelo autor

Na segunda aba de visualização é apresentado os mesmos gráficos, mas apenas com as previsões futuras, entre 2020 e 2022, onde não conhecemos o número de casos reais, conforme mostra a figura 37. Para as previsões de 2020 a 2022, o modelo utilizou os dados até 2018, como no modelo gerado para prever os casos até 2019.

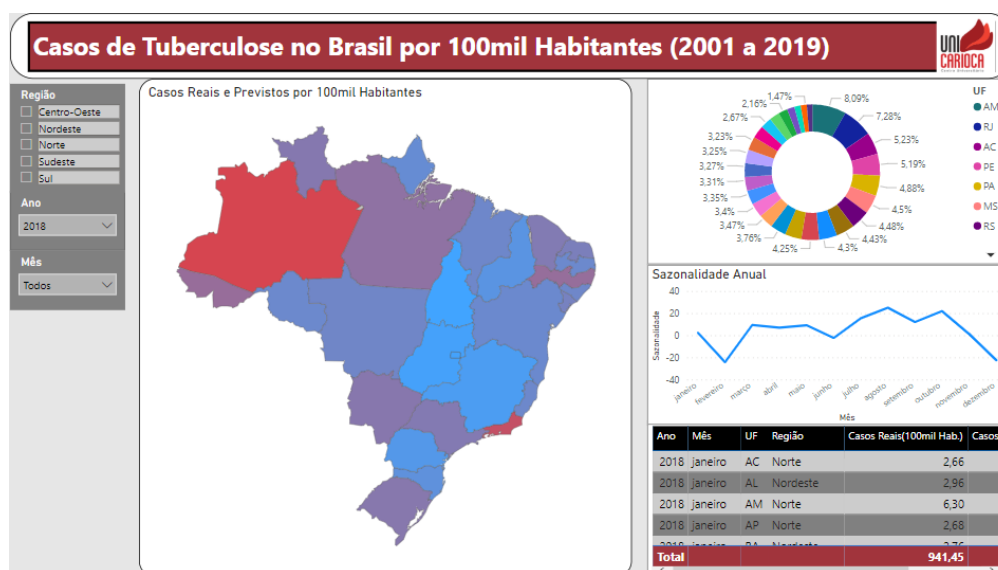
Figura 37 – Dashboard Número de casos previstos absolutos



Fonte: Elaborado pelo autor

Na terceira aba é possível visualizar o número de casos por 100mil habitantes, como explicado no capítulo 4.4 esta é a melhor forma de interpretar os dados. Na figura 36, por exemplo, São Paulo aparece em vermelho por ter o maior número de casos absolutos, porém quando temos a visão comparada com a população de cada região, os estados do Amazonas e Rio de Janeiro, aparecem em destaque como os estados com maior número de casos por 100mil habitantes, conforme mostra a figura 38.

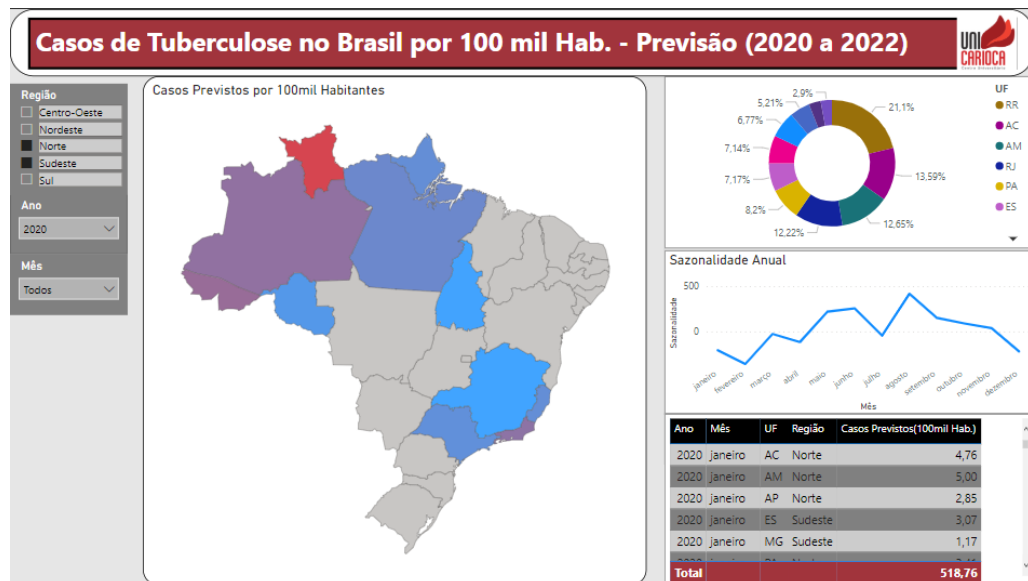
Figura 38 – Dashboard com o número de casos por 100mil habitantes



Fonte: Elaborado pelo autor

Por último é apresentado o número de casos previstos por 100mil habitantes, onde também é possível visualizar outros estados com maior impacto da Tuberculose ao longo dos próximos anos. A figura 39 mostra o gráfico de previsão de casos por 100mil habitantes filtrado apenas para o ano de 2020 nas regiões Norte e Sudeste, onde pode-se visualizar os estados com maior número de casos por 100mil habitantes.

Figura 39 – Dashboard com número previsto de casos por 100mil Habitantes



Fonte: Elaborado pelo autor

5 CONCLUSÃO

O objetivo deste trabalho foi apresentar todas as etapas de um processo de Ciência de Dados e Machine Learning aplicado a previsão do número de casos de tuberculose no Brasil, utilizando as principais ferramentas disponíveis no mercado. O projeto passou pela extração dos dados no site do Ministério da Saúde através de ferramentas de web scraping, utilizou a linguagem de programação Python e diversas bibliotecas de pré-processamento e visualização dos dados, utilizou Machine Learning através da ferramenta Prophet que gera previsões através de dados históricos e séries temporais, considerando efeitos de sazonalidade anual e tendência, e no final uma apresentação dos resultados num dashboard no Power BI para que o usuário possa interagir e compreender as informações.

A tuberculose no Brasil é uma doença bastante sazonal, onde fica claro em cada região os períodos com maiores índices de números de casos. Ainda é possível observar uma subnotificação por parte de algumas secretarias de saúde estaduais, dificultando as previsões nestes Estados. Mas, no geral é possível observar uma tendência elevada de alta no número de casos a partir de 2017, cenário que pode mudar com a pandemia do COVID-19 em 2020, conforme apresentado no capítulo 4.

São Paulo possui um maior número de casos absolutos, porém ao analisar o número de casos por 100mil habitantes é possível observar uma maior taxa nos Estados do Amazonas e Rio de Janeiro. Prever o número de casos e a possibilidade de visualizar os dados em um dashboard iterativo, facilita a tomada de decisões para que as pessoas e órgãos responsáveis possam tomar as melhores medidas de prevenção e realizar um melhor planejamento. Este trabalho utilizou a Tuberculose como referência, mas ferramentas de Ciência de Dados e Machine Learning podem e devem auxiliar na área da saúde em todo o mundo.

REFERÊNCIAS

- ANACONDA-INC, 2020. Disponível em <<https://www.anaconda.com/products/individual>>. Acesso em 21 de Junho de 2020.
- APACHE. Apache TM Hadoop®. **Apache**, 2020. Disponível em <<http://hadoop.apache.org/>>. Acesso em 20 de Junho de 2020.
- BRASIL. Tuberculose: o que é, causas, sintomas, tratamento, diagnóstico e prevenção. **Ministério da Saúde do Brasil**, 2020a. Disponível em <<http://saude.gov.br/saude-de-a-z/tuberculose>>. Acesso em 11 de Julho de 2020.
- BRASIL. Ministério da Saúde/SVS - Sistema de Informação de Agravos de Notificação - Sinan Net. **Ministério da Saúde do Brasil**, 2020b. Disponível em <<http://tabnet.datasus.gov.br/cgi/tabcgi.exe?sinanet/cnv/tubercbr.def>>. Acesso em 11 de Julho de 2020.
- DAVENPORT, T.H. Data Scientist: The Sexiest Job of the 21st Century. **Harvard Business Review**, 2012. Disponível em <<https://hbr.org/2012/10/data-scientist-the-sexiest-job-of-the-21st-century>>. Acesso em 20 de Junho de 2020.
- ENTWISTLE, J. LRO Releases Final Set of Exploration Data. **NASA**, 2011. Disponível em <https://blogs.nasa.gov/NES_Teachers_Corner/2011/05/16/post_1305578409053/>. Acesso em 20 de Junho de 2020.
- EQUIPE DSA. Linguagem R – Por que é hora de aprender?. **Data Science Academy**, 2018a. Disponível em <<http://datascienceacademy.com.br/blog/linguagem-r-por-que-e-hora-de-aprender/>>. Acesso em 21 de Junho de 2020.
- EQUIPE DSA. WEB Scraping e WEB Crawling são legais ou ilegais?. **Data Science Academy**, 2018b. Disponível em <http://datascienceacademy.com.br/blog/web-scraping-e-web-crawling-sao-legais-ou-ilegais/>. Acesso em 04/07/2020.
- EQUIPE DSA. Por que a linguagem Python é tão popular em Machine Learning e Inteligência Artificial?. **Data Science Academy**, 2020 Disponível em <<http://datascienceacademy.com.br/blog/por-que-a-linguagem-python-e-tao-popular-em-machine-learning-e-inteligencia-artificial/>>. Acesso em 21 de Junho de 2020.

FAYYAD, U; PIATETSKY-SHAPIRO, G; SMYTH, P. From Data Mining to Knowledge Discovery in Databases. American Association for Artificial Intelligence. **AI Magazine** v.17(3), 1996.

GOOGLE, 2020. Disponível em <<https://colab.research.google.com/notebooks/intro.ipynb>>. Acesso em 21 de Junho de 2020.

GOULARTE JUNIOR, S.J. O que é um Data Lake?. **Scurra**, 2019 Disponível em <<http://www.scurra.com.br/blog/o-que-e-um-data-lake/>>. Acesso em 20 de Junho de 2020.

GRUS, J. **Data Science do Zero**. 1º ed. Alta Books, 2016. p.26

GUEDES, E., GUEDES, J. Panorama do COVID-19 no Brasil. **Plataforma Kaggle**, 2020. Disponível em <<https://www.kaggle.com/elloaguedes/panorama-do-covid-19-no-brasil>>. Acesso em 04 de Julho de 2020.

HAND, D; MANNILA, H; SMYTH, P. **Principles of Data Mining**. 1º ed. MIT Press, 2001.

HUNTER, J.D. Matplotlib: A 2D Graphics Environment, **Computing in Science & Engineering**, 2007, vol. 9 (3), pp. 90-95.

IBM. Artificial intelligence in healthcare. **IBM**, 2020. Disponível em <<https://www.ibm.com/watson-health/learn/artificial-intelligence-healthcare>>. Acesso em 06 de Julho de 2020.

MACHADO, A., et al. Fatores associados ao atraso no diagnóstico da tuberculose pulmonar no estado do Rio de Janeiro. **Jornal Brasileiro de Pneumologia**, 2011. 37 (4), p512-520

MATOS, D. Conceitos Fundamentais de Machine Learning. **Ciência e Dados**, 2019. Disponível em <<http://www.cienciaedados.com/conceitos-fundamentais-de-machine-learning/>> Acesso em 20 de Junho de 2020.

MICROSOFT. O que é o Power BI?. **Microsoft**, 2020. Disponível em <<https://powerbi.microsoft.com/pt-br/what-is-power-bi/>>. Acesso em 04 de Julho de 2020.

MICROSTRATEGY. O que é mineração de dados?. **Microstrategy**, 2020. Disponível em <<https://www.microstrategy.com/br/resources/introductory-guides/data-mining-explained>>. Acesso em 20 de Junho de 2020.

NUSSBAUMER, C. Drawing attention with data labels. **Storytelling with Data**, 2012.

Disponível em <<https://www.storytellingwithdata.com/blog/2012/06/drawing-attention-with-data-labels>>. Acesso em 21 de Junho de 2020..

OLIPHANT T. E. **Guide to NumPy**. 1º ed. Createspace Independent Publishing Platform, 2006. P13-16.

OLIVEIRA, A.R. et al. Comparison of machine-learning algorithms to build a predictive model for detecting undiagnosed diabetes. ELSA-Brasil: accuracy study. **São Paulo Medical Journal**, São Paulo, 2017. v.135, n.3, p.234-246

ORACLE. O que é Big Data?. **Oracle**, 2020a. Disponível em <<https://www.oracle.com/br/big-data/what-is-big-data.html>>. Acesso em 20 de Junho de 2020.

ORACLE. Por que o Big Data da Oracle?. **Oracle**, 2020b. Disponível em <<https://www.oracle.com/br/big-data/>>. Acesso em 20 de Junho de 2020.

ORACLE. O que É um Data Warehouse?. **Oracle**, 2020c. Disponível em <<https://www.oracle.com/br/database/what-is-a-data-warehouse/>>. Acesso em 04 de Julho de 2020.

PANDAS, 2020. Disponível em <<https://pandas.pydata.org/about/>>. Acesso em 21 de Junho de 2020.

PASSOS, D.S. Big Data, Data Science e seus contributos para o avanço no uso da open source intelligence. **Sistemas & Gestão**, 2016. Disponível em <<https://revistasg.uff.br/sg/article/view/1026/524>>. Acesso em 20 de Junho de 2020.

PEDEGROSA, F, et al. **Scikit-learn: Machine Learning in Python**. JMLR 12, 2011 pp. 2825-2830.

PROJETO JUPYTER, 2020. Disponível em <<https://jupyter.org/about>>. Acesso em 21 de Junho de 2020.

RAMOS, R. Regressão Linear Simples: O Que é? Para Que Serve? Como Funciona?. **O estatístico**, 2020. Disponível em <<https://oestatistico.com.br/regressao-linear-simples/>>. Acesso em 27 de Junho de 2020.

SELL, I. Utilização da regressão linear como ferramenta de decisão na gestão de custos. In: IX Congresso Internacional de Custos, 2005. Santa Catarina. p.4-6

SILVA, L., et al. Aplicação de Deep Learning no pré-diagnóstico da COVID-19 através de imagens de raio-x. **UNIFESSPA**. 2020. Disponível em <<https://acoescovid19.unifesspa.edu.br/2-uncategorised/100-aplica%C3%A7%C3%A3o-de-deep-learning-no-pr%C3%A9-diagn%C3%B3stico-da-covid-19-atrav%C3%A9s-de-imagens-de-raio-x.html>>. Acesso em 06 de Julho de 2020.

STEVENSON, W. J. **Estatística aplicada à administração**. 1^o ed, Harbra, São Paulo, 1986, p. 341.

TAYLOR, S.J., LETHAM, B. Prophet: forecasting at scale. **PeerJ PrePrints**, 2017a. Disponível em <<https://research.fb.com/blog/2017/02/prophet-forecasting-at-scale/>>. Acesso em 27 de Junho de 2020.

TAYLOR, S.J., LETHAM, B. Forecasting at Scale. **PeerJ PrePrints**, 2017b. Disponível em <<https://research.fb.com/blog/2017/02/prophet-forecasting-at-scale/>>. Acesso em 27 de Junho de 2020.

WASKOM, M. An Introduction to Seaborn. **Seaborn**, 2020a. Disponível em <<https://seaborn.pydata.org/introduction.html>>. Acesso em 21 de Junho de 2020.

WASKOM, M. Seaborn.pairplot. **Seaborn**, 2020b. Disponível em <<https://seaborn.pydata.org/generated/seaborn.pairplot.html?highlight=pairplot#seaborn.pairplot>>. Acesso em 21 de Junho de 2020.