Relatório do Projeto

O intuito desse projeto, como parte do Sprint 1 da equipe PrimaType, é, principalmente, aprender as várias etapas fundamentais da ciência de dados, envolvendo coleta, limpeza/tratamento, e visualização dos dados. O objetivo é realizar uma análise detalhada dos dados populacionais de diferentes espécies de primatas utilizando Python e bibliotecas essenciais para o ramo.

No processo de análise de dados, precisamos seguir alguns passos pra assegurar a precisão e concisão da análise. Dentre esses passos, destaca-se o EDA (Exploratory Data Analysis, ou Análise Exploratória de Dados). O objetivo do EDA é compreender melhor os dados, suas características, e garantir sua integridade antes de aplicar técnicas avançadas de análise ou modelagem.

Essa etapa, que envolve limpeza, organização, e transformação dos dados, é a etapa seguinte da coleta dos dados que serão extraídas as informações úteis pra tomadas de decisão. Fora Python, outras ferramentas podem ser úteis como Excel e técnicas mais avançadas como uso de redes neurais (inteligência artificial). A ideia é identificar padrões e tendências à fim de solucionar algum problema específico.[^1]

I. - Coleta de Dados

A princípio, coletamos os dados que serão analisados. No escopo desse projeto, é o primates_dataset.csv, um arquivo CSV, mas poderiam também ser extraídos de um banco de dados, APIs, dentre outras fontes.

Com o intuito de facilitar a visualização dos dados, foi criado uma classe de utilidade chamada **Query**. Essa classe é responsável por tratar o CSV como um banco de dados relacional, trazendo suporte à consultas em formato similar à ORMs como TypeORM, SQLAlchemy, Eloquent, dentre outros. Pra se parecer com a sintaxe de ORMs, a classe tem métodos encadeáveis que visam a padronização do SQL padrão ANSI.[^2]

II. - Análise das variáveis

Em segundo passo, conhecemos as variáveis. Entender o que cada uma delas representa, suas unidades de medida. Essa etapa pode envolver tanto análise do próprio cientista de dados quanto de leitura da documentação dos dados, se disponível, à fim de obter mais contexto sobre como os dados foram coletados, e suas limitações.

II.1 - Quais são as variáveis disponíveis?

O CSV contém as colunas:

- species_id -> Identificador de cada linha de 1 a 150;
- species_name -> Nome da espécie de primata. São elas:
  - Gorilla;
  - Chimpanzee;
  - Orangutan;
  - Gibbon;
  - Bonobo;
  - Lemur;
  - Tarsier;
  - Howler Monkey;
  - Spider Monkey;
  - Macaque.

Todos os dados a seguir são referentes à devida espécie correspondente (na mesma linha).

- population -> Quantidade de espécimes vivos.
- year -> referente à qual ano é a informação dessa linha. Os anos vão de 2020 até 2006.
- habitat_region -> Região em que habita a espécie. São elas:
  - Central Africa;
  - West Africa;
  - Madagascar;
  - Southeast Asia;
  - East Asia;
  - South America.
- diet -> A dieta da espécie de primata. Pode ser:
  - Herbivore;
  - Omnivore;
  - Frugivore;
  - Insectivore.
- avg_lifespan -> Um valor que representa quantos anos é o tempo de vida médio de um espécime.
- social_behavior -> O comportamento social das espécies. Pode ser:
  - Group;
  - Solitary;
  - Pair.
- genetic_variation -> A variação genética da espécie.
- health_status -> Classificação que indica o quão ameaçada de extinção está a espécie. São possíveis valores:
  - Healthy;
  - Near Threatened.

  - Vulnerable;
  - Endangered;
  - Critically Endangered;
- latitude -> A coordenada latitudinal da região em que habita a espécie.
- longitude -> A coordenada longitudinal da região em que habita a espécie.

II.2 - Questionamentos referentes às variáveis

Dos dados reunidos acima, é possível inferir sobre cada um deles seus tipos primitivos. No entanto, dois levantam o questionamento: genetic_variation e health_status.

II.2.1 - genetic_variation

A variável genetic_variation implica variação genética da espécie, mas variação à oque? De acordo com [algumas informações](https://humanorigins.si.edu/evidence/genetics#:~:text=The%20DNA%20difference%20with%20gorillas,Asian%20great%20ape%2C%20the%20orangutan.), os valores batem com a variação genética do gene da espécie referente o gene humano. Mas nas mesmas fontes, outros valores são providenciados, e discutir a veracidade do fornecido pelo CSV é plausível.[^3]

II.2.2 - health_status

Podemos inferir que esses valores se referem à classificação determinada pela [União Internacional pela Conservação da Natureza](https://www.iucn.org/)União Internacional pela Conservação da Natureza. Na amostragem dos dados II.1, os dados referentes à variável "health_status" já estão em ordem (de cima pra baixo) pra mais próxima da extinção.[^4]

II.2.3 - latitude & longitude

Em primeira instância uma pessoa poderia se perguntar aonde se referem as coordenadas fornecidas em cada linha. Elas levam para a localização indicada pela variável "habitat_region". Cabe à etapa de limpeza dos dados verificar se isso é verossímil pra todas as linhas.

II.3 - Transformação das variáveis

É uma etapa importante da análise pós-coleta de dados que verifiquemos quais variáveis possuem mais correlação com o que desejamos. Para o escopo do projeto atual, percebe-se que as variáveis estão satisfatórias com sua exibição, talvez por exceção de "genetic_variation", que, ao invés de um float entre 0.00 à 0.10, poderia ser uma porcentagem. Depende de cada cientista. Além disso, a separação de dois campos "latitude" e "longitude" pode desagradar quem preferisse um único campo "coordinates".

II.4 - Variáveis mais Importantes

Em alguns projetos, é necessária uma investigação sobre quais das variáveis possuem mais correlação com o objetivo. Como o intuito desse projeto é analisar a variação da população dessas espécies, identificar e visualizar as tendências populacionais ao longo do tempo, fica evidente que as nossas variáveis que receberão mais destaque serão "population" e "year".

III. - Limpeza dos Dados

Uma das etapas mais importantes da Análise Exploratória de Dados é a limpeza dos dados. As informações que temos, não necessariamente, podem estar 100% corretas. É nosso trabalho identificar possíveis discrepâncias, como anomalias, dados errôneos, valores ausentes e inconsistências como duplicatas.

III.1 - Tratamento de Valores Ausentes

Começaremos a limpeza dos dados identificando e tratando valores ausentes. Podemos fazer isso removendo linhas ou colunas com muitos valores ausentes, ou adicionando nossos próprios valores como indicação de

valores nulos (adicionando 0, por exemplo) ou informações previstas de acordo com os outros dados.

Para fazer isso, foi criado a classe **DataConsistencyValidator** e seu método **verify_all_empty_entries**. Esse é responsável por percorrer, linha a linha, os dados fornecidos, e ao final, indica quais linhas tiveram valores vazios, e em qual coluna exatamente. Graças à ele, foi possível tirar essa informação:

- Vazio encontrado na linha 4, coluna 'habitat_region'
- Vazio encontrado na linha 13, coluna 'population'
- Vazio encontrado na linha 54, coluna 'habitat_region'
- Vazio encontrado na linha 63, coluna 'population'

Os valores perdidos em "habitat_region" são fáceis de prever: Pegaremos das outras linhas essa informação e preencheremos na mão mesmo. Será preenchido na mão pois o CSV é pequeno, no entanto, em CSVs maiores, seria necessária a criação de outro método na classe **DataConsistencyValidator** ou criar uma classe apropriada pra essa etapa da limpeza.

Na linha 4, a espécie "Gibbon" está sem habitat_region. Vemos em outras linhas da espécie "Gibbon" que essa informação é "Southeast Asia". Na linha 54 vemos o mesmo caso, também com a espécie "Gibbon", então também será preenchido com o dado "Southeast Asia".

III.1.1 - O que fazer para tratar valores ausentes

Pro caso das linhas 13 e 63, o valor da população da espécie "Orangutan" está faltando. Para isso, podemos ou prever qual era a população do momento pra substituí-lo, ou deletar a informação.

Se a quantidade de dados ausentes é pequena e não compromete significativamente a integridade da análise, optamos pela remoção das linhas com valores ausentes. Isso é totalmente válido, se houver um número suficiente de outras observações para realizar análises sem esses registros.[^5]

Entretanto, se a remoção das linhas não é desejável devido à perda de dados, é plausível considerarmos estimar o valor ausente com base em uma média ou outra medida central dos valores disponíveis.

A escolha entre essas abordagens depende do contexto específico do estudo e das características dos dados. Enquanto a remoção garante que a análise seja feita apenas com dados originais e completos, ela reduz a quantidade total de dados disponíveis. Já a estimação do valor ausente permite manter mais dados para a análise, mas introduz uma estimativa que pode afetar a precisão dos resultados.

Cabe ao cientista/analista de dados decidir baseado em seu contexto: revisar as diretrizes/normas aplicáveis à sua análise, garantindo que as escolhas sejam adequadas e bem fundamentadas, e também deve-se considerar a consulta de especialistas ou colegas do projeto.[^6]

III.1.2 - Manter a transparência

Na preparação de um relatório ou visualização dos dados em que houve imputações ou estimativas de valores ausentes, é importante fornecer transparência sobre como esses dados foram tratados. Possíveis práticas:

- Legenda ou Nota de Rodapé no Gráfico/Relatório
- Marcadores Visuais
- Discussão no Texto

Mostrar transparência aumenta a credibilidade do trabalho, demonstrando a consideração atenta quanto ao tratamento de dados ausentes, além de ajudar os leitores a interpretarem corretamente os resultados, entendendo o impacto que valores ausentes tratados podem ter no resultado. Ademais, possibilita os envolvidos com o projeto de compreenderem melhor as decisões aplicadas no processamento, fortalecendo a confiança.

III.1.3 - O que foi feito

Usando o aprendizado anterior, decidiu-se estimar os valores ausentes ante excluí-los, visando manter mais dados para análise.

Fazendo uso da nossa **Query** para capturar todas as linhas referentes à orangutangos, tivemos que os valores faltantes estão precedidos de uma população de 700 (anos 2013 e 2018) e seguidos de uma população de 750 (anos 2015 e 2020). Como a população cresceu 50 em 2 anos, podemos inferir pela média que houve um crescimento de 25 por ano. Conclui-se que um valor possível a se considerar pra preencher os valores ausentes seria 25.

Em futuras visualizações, como gráficos, haverá a transparência de indicar o preenchimento artificial desses dados, como neste mesmo documento.

III.2 - Correção de Inconsistências

Antes de termos falado sobre os valores vazios na coluna "population", houve valores vazios na coluna "habitat_region" que precisaram ser preenchidos. É importante que certos valores qualitativos ou quantitativos arbitrários permaneçam consistentes entre si. As colunas "habitat_region", "diet", "social_behavior", "genetic_variation", "health_status", "latitude" e "longitude" não fazem sentido mudarem no contexto do projeto. Pra esses valores mudarem, muitos anos precisam se passar, muito mais do que a faixa 2006-2020 que temos no nosso banco atual.

A coluna mais provável de se mudar nessa pequena faixa de tempo seria "health_status", já que isso é um dado qualitativo determinado pelo órgão internacional falado anteriormente. No entanto, pro nosso projeto, não houve quaisquer mudanças. No mesmo ponto em que isso foi resolvido também foi falado sobre confirmar que as coordenadas (latitude e longitude) se adequam com a fornecida pela "habitat_region".

Pra cada espécie, os valores que são, sim, esperados terem mudanças hora ou outra, são os das colunas "population", "year", e "avg_lifespan".

Para confirmar tudo isso, foi-se criado o método **verify_column_consistency** na classe **DataConsistencyValidator**. Esse método verifica, dado uma base de dados e as devidas colunas, se todos os dados se encaixam com o primeiro valor fornecido pra cada espécie.

Utilizando esse método, foi possível inferir que todos os dados estão consistentes após o preenchimento dos valores ausentes, possibilitando que sigamos em frente no tratamento.

III.3 - Remoção de Outliers

"Outliers" são valores que se desviam significativamente dos outros dados em um conjunto. Podem ser causados por erros de medição, entrada incorreta de dados, ou até mesmo valores legítimos, representando variabilidade real no conjunto de dados em questão. Eles podem distorcer estatísticas descritivas como média e desvio padrão, afetar

modelos estatísticos de Machine Learning e influenciar negativamente na visualização de dados. Em resumo, são dados anômalos que podem distorcer a análise.

A identificação e remoção dos outliers é uma etapa importante na análise de dados, mas deve ser realizada com cuidado para garantir que os resultados finais sejam precisos. Possíveis formas de identificar outliers são por meio de métodos estatísticos e pela visualização direta dos dados.[^7]

III.2.1 - Método Estatístico Z-Score

O Z-Score é uma pontuação que mede quantos desvios padrão um valor está distante da média. Valores de Z-Score acima de 3 ou abaixo de -3 (ou, simplesmente, o valor absoluto do Z-Score acima de 3) são geralmente considerados outliers. Pontuações positivas indicam que o valor está acima da média, enquanto que a pontuação negativa indica que está abaixo dessa média. Serve pra determinar a volatilidade dos dados do conjunto.

O desvio padrão é essencialmente um reflexo da quantidade de variabilidade dentro desse conjunto. Segundo estudos decorrentes desde a década de 1960[^8], 99,7% dos valores prestativos contam dentro desse intervalo de -3 a 3. No entanto, ele é tão preciso quanto os dados inseridos nele, logo, não é imune à dados falsos ou inseridos errôneamente. É por isso que a etapa de remoção de anomalias é uma das últimas na limpeza.[^9]

1. Calculamos a média dos valores da coluna de interesse;
2. Da mesma forma, calculamos o desvio padrão;
3. Calculamos o Z-Score pra cada valor na coluna subtraindo o valor pela média e dividindo pelo desvio padrão;
4. Identificar todos os valores que, absolutos, são maiores que 3.

Pra atingir isso, foi-se criado o método **z_score** na classe **Outliers**. Essa classe servirá pra comportar os métodos de rastreamento de anomalias, sendo o método z_score, a métrica apresentada agora.

Com esse método, foi possível notar que nossos dados populacionais e de média de vida não estão voláteis.

III.2.2 - Método Estatístico Interquartile Range

O IQR é a diferença entre o terceiro quartil (Q3) e o primeiro quartil (Q1) do conjunto de dados. Valores que estão abaixo de Q1 - 1.5 _ IQR ou acima de Q3 + 1.5 _ IQR são, normalmente, considerados outliers.[^10]

1. Encontramos o valor do primeiro quartil;
2. Encontramos o valor do terceiro quartil;
3. Calculamos o IQR subtraindo o terceiro quartil do primeiro;
4. Identificamos todos os valores menores que Q1 - 1.5IQR e maiores que Q3 + 1.5IQR.

Pra cumprir com esse fim, foi criado o método **interquartile_range**, também na classe **Outliers**. Com esse método, foi possível notar que os dados de média de vida não estão muito discrepantes. No entanto, há registro de dados discrepantes quanto à população, especificamente, de chimpanzés.

No entanto, deve-se ter ciência do contexto do domínio dos dados e análise da sensibilidade para interpretar corretamente os outliers. Nem todos os valores extremos são, necessariamente, erros ou dados a serem removidos. Ao analisar, visualmente, o CSV, vemos que a população das espécies é aproximadamente a mesma, com pouca variação, exceto para os chimpanzés, que têm, realmente, uma população mais alta do que das demais. No entanto, isso não é um erro: é simplesmente uma observação de que há um grupo com população maior.[^11]

Sendo assim, conclui-se que nenhuma remoção ou alteração desses valores será necessária.

IV. - Transformação dos Dados

Tendo os conjuntos de dados sido devidamente agregados e separados em suas estruturas mais apropriadas, uma importante etapa é a normalização / padronização destes. Essa etapa está, principalmente, ligada à Machine Learning (aprendizado de máquina), redes neurais e modelos de linguagem. Portanto, é muito comum explorarmos a biblioteca scikit-learn.

Scikit-learn é uma biblioteca de Python desenvolvida especificamente para a aplicação prática do machine learning. Dispõe de ferramentas simples e eficientes para análise preditiva de dados, é reutilizável, código aberto e acessível, principalmente por ter sido construída em cima de outras bibliotecas muito bem conhecidas e consolidadas: NumPy, SciPy e matplotlib.[^12]

Suas principais aplicações envolvem pré-processamento de dados, classificação, regressão, clusterização, redução de dimensionalidade, ajuste de parâmetros, dentre outras funcionalidades. Não serão vistas devido ao escopo desse projeto, mas serão exploradas.[^12]

IV.1 - Codificação de Dados Categóricos

Dados categóricos (qualitativos) têm esse nome por serem divididos, separados em categorias. Exemplos incluem cores, marcas, etc. Os modelos de Machine Learning esperam receber dados numéricos, na grande maioria dos casos não é possível usar variáveis categóricas nesses modelos. É necessário converter eles em variáveis numéricas, de uma forma que mantenha a informação e a relação entre os dados.[^13]

IV.1.1 - Codificação de Rótulo (Label Encoding)

A Codificação de Rótulo é uma técnica em que cada categoria única é atribuída a um número inteiro único. Por exemplo, "vermelho" e "azul" em uma coluna "Cores" poderiam ser 0 e 1.[^13]

Vale notar que a Codificação de Rótulo cria uma suposição implícita de que as categorias têm uma ordem, ou hierarquia, que pode ou não existir. Nesse sentido, é melhor utilizá-la para variáveis categóricas ordinais, isto é, seguem uma ordem intrínseca. Um exemplo seria "pequeno", "médio", "grande". Em caso de variáveis categóricas nominais, outras técnicas seriam mais apropriadas.[^13]

IV.1.2 - Codificação One-Hot

Um dos métodos mais comuns, essa técnica transforma cada categoria em uma nova coluna binária (0 ou 1). A ausência dessa categoria indica 0 e a presença, 1. Por exemplo, uma coluna "Gênero" com valores "Macho" e "Fêmea" se transformariam em duas colunas: "Gênero_Macho" e "Gênero_Fêmea". Os antigos dados que teriam "Macho" preenchido na coluna "Gênero" agora teriam 1 na coluna "Gênero_Macho" e 0 na coluna "Gênero_Fêmea".[^13]

Apesar de ser uma das codificações mais utilizadas e eficazes, ela aumenta a dimensionalidade do conjunto de dados se existirem muitas categorias únicas. Isso aumenta o tempo de computação e memória, além de reduzir o desempenho.[^13]

Existem várias outras métricas, cada uma abrange um escopo diferente. A escolha de cada uma vai do tipo de dado analisado e da prefêrencia da equipe analisando.

V. - Normalização dos Dados

A normalização dos dados é uma técnica de pré-processamento de dados pra ajustar os valores de diferentes variáveis em uma escala comum, sem distorcer as diferenças nos intervalos de valores. Ela é importante pra garantir que todas contribuam igualmente para

a análise ou modelo, evitando que seu estudo fique enviesado para as variáveis com maior ordem de grandeza.[^14]

As técnicas mais utilizadas de normalização em Aprendizado de Máquina são: Min-max, Z-Score, e Log Scaling.[^15] Falaremos somente sobre a Min-max, haja vista que a Z-Score já foi comentada.

V.1 - Min-Max Scaling

A técnica Min-max reescala valores à um alcance entre 0 e 1, ou -1 e 1.

Um valor x está normalizado nessa métrica, ao subtraí-lo pelo valor mínimo encontrado na coluna e dividido pela subtração do máximo pelo mínimo.

Por exemplo, em 5 números: 14, 9, 24, 39 e 60. O menor valor é 9 e o maior é 60. Portanto, o valor normalizado de 60 seria 1 e o de 9 seria 0, e dos intermediários:

14: (14 - 9) / (60 - 9) = 5 / 51 = 0.098

24: (24 -9) / (60 - 9) =15 / 51 = 0.29

39: (39 - 9) / (60 - 9) = 30 / 51 = 0.58

A biblioteca sklearn.preprocessing.MinMaxScaler implementa isso pra gente. Utilizando os métodos fit() e depois transform() com nossos dados como parâmetros, recebemos esses dados normalizados.[^15]

Nesse projeto, não usaremos normalização de dados.

VI. - Agregação de Dados

Saindo da etapa de limpeza, remoção e tratamento dos dados, temos o processo de resumir os conjuntos de dados. Coletar e agrupar os dado em um formato compacto permite compreender mais facilmente e representa melhor visões estatísticas. Dados agrupados facilitam o processo de tomada de decisão.[^16]

Como até agora todo o código tem sido escrito em orientação a objetos, faz muito sentido separarmos cada dado de primata em uma classe própria **Primate**. Essa classe separa todas as informações necessárias em seus devidos tipos primitivos, ou não: alguns dados qualitativos e outras informações mais específicas tiveram tipagens próprias também, definidas na utilidade **Types**. A classe Primate também reforça a tipagem lançando exceções (erros) caso seja inserido um dado inválido, pois, em se tratando da análise de dados, é plausível querer sensibilidade em relação ao que estamos trabalhando. Toda informação importa e deve ser exatamente o que esperamos (ou precisamos) que seja.

No entanto, como queremos que nossa classe Primate apenas comporte as informações referentes aos primatas que estamos estudando, manter a complexidade de código de validação de entrada dos dados, bem como assegurar a estrutura de cada uma distorce o propósito da classe. Por conta disso, foi decidido criar um arquivo **PrimateFactory**, que, como em Java, é responsável pela complexidade extra na criação da classe que desejamos. Assim, cada arquivo bate exatamente com a expectativa de que teria dentro.

VII. - Visualização dos Dados

A etapa de visualização dos dados consiste em observá-los em forma de gráficos e outras informações visuais.

Bibliografia

[^1]: ENTENDA o que é análise de dados, quais os processos envolvidos e como implementar na sua empresa. Cinnecta. Disponível em: https://cinnecta.com/conteudos/analise-de-dados/ . Acesso em: 2 de jul. de 2024.

[^2]: ANSI SQL O idioma para sistemas de gerenciamento de banco de dados relacional. FasterCapital. Disponível em: https://fastercapital.com/pt/contente/ANSI-SQL--O-idioma-para-sistemas-de-gerenciamento-de-banco-de-dados-relacional.html#:~:text=A%20ANSI%20SQL%2C%20ou%20American,padr%C3%A3o%20atual%20ANSI%20SQL%3A%202016. . Acesso em: 2 de jul. de 2024.

[^3]: GENETIC Evidence. The Smithsonian National Museum of Natural History. Disponível em: https://humanorigins.si.edu/evidence/genetics . Acesso em: 29 de jun. de 2024.

[^4]: IUCN, IUCN. Página inicial. Disponível em: https://www.iucn.org/ . Acesso em: 29 de jun. de 2024.

[^5]: JÚNIOR, Clébio de Oliveira. Feature Engineering: Técnicas para lidar com dados faltantes em um projeto de ciência de dados. Medium. Disponível em: https://medium.com/data-hackers/feature-engineering-t%C3%A9cnicas-para-lidar-com-dados-faltantes-em-um-projeto-de-ci%C3%AAncia-de-dados-debdd57eb662 . Acesso em: 2 de jul. de 2024.

[^6]: MACHINE Learning: Preenchimento de zeros - Manipulação de dados faltantes. Awari. Disponível em: https://awari.com.br/machine-learning-preenchimento-de-zeros-manipulacao-de-dados-faltantes-2/?utm_source=blog&utm_campaign=projeto+blog&utm_medium=Machine%20Learning:%20Preenchimento%20de%20zeros%20-%20Manipula%C3%A7%C3%A3o%20de%20dados%20faltantes . Acesso em: 2 de jul. de 2024.

[^7]: MEDEIROS, Ricardo. Tratando Valores Outliers em um DataFrame usando Python. dio. Disponível em: https://www.dio.me/articles/tratando-valores-outliers-em-um-dataframe-usando-python . Acesso em: 2 de jul. de 2024.

[^8]: Z-SCORE: saiba o que é e como funciona. Mais Retorno, 2022. Disponível em: https://maisretorno.com/portal/termos/z/z-score . Acesso em: 30 de jun. de 2024.

[^9]: Z-SCORE. Oracle Help Center. Disponível em: https://docs.oracle.com/cloud/help/pt_BR/pbcs_common/PFUSU/insights_metrics_Z-Score.htm#PFUSU-GUID-640CEBD1-33A2-4B3C-BD81-EB283F82D879 . Acesso em: 30 de jun. de 2024.

[^10]: BHANDARI, Pritha. How to Find Interquartile Range (IQR) | Calculator & Examples. Scribbr. Disponível em: https://www.scribbr.com/statistics/interquartile-range/ . Acesso em: 2 de jul. de 2024.

[^11]: MACIEL, Prof. Fernanda. Excluir Outliers? Usar média ou mediana? | Prof. Fernanda Maciel. YouTube, 23 de ago. de 2021. 3m31s. Disponível em: https://www.youtube.com/watch?v=o3uTAZyROI8 . Acesso em: 2 de jul. de 2024.

[^12]: A Biblioteca scikit-learn - Python: o que é, para que serve. Didática Tech. Disponível em: https://medium.com/@pedrorp/guia-de-codificadores-de-atributos-categ%C3%B3ricos-em-machine-learning-60a9f22c9a3b . Acesso em: 3 de jul. de 2024.

[^13]: PASSOS, Pedro César Ribeiro. Guia de Codificadores de Atributos Categóricos em Machine Learning. Disponível em: https://medium.com/@pedrorp/guia-de-codificadores-de-atributos-categ%C3%B3ricos-em-machine-learning-60a9f22c9a3b . Acesso em: 3 de jul. de 2024.

[^14]: VAZ, Arthur Lamblet. Normalizar ou padronizar as variáveis?. Medium. Disponível em: https://medium.com/data-hackers/normalizar-ou-padronizar-as-vari%C3%A1veis-3b619876ccc9 . Acesso em: 6 de jul. de 2024.

[^15]: DATA Normalization With Python Scikit-Learn: Tips & Tricks for Data Science. Turing. Disponível em: https://www.turing.com/kb/data-normalization-with-python-scikit-learn-tips-tricks-for-data-science . Acesso em: 6 de jul. de 2024.

[^16]: SPASOJEVIC, Anastasia. O que é agregação de dados?. phoenixNAP Global IT Services. Disponível em: https://www.phoenixnap.pt/gloss%C3%A1rio/Agrega%C3%A7%C3%A3o-de-dados . Acesso em: 5 de jul. de 2024.