Universidade de Brasília
Instituto de Ciências Exatas
Departamento de Ciência da Computação

Mineração de dados socioeconômicos do World Bank

Luthiery Costa Cavalcante
Fernando Ferreira Cordeiro

Monografia apresentada como requisito parcial
para conclusão do Bacharelado em Ciência da Computação

Orientador
Prof. Dr. Jan Mendonça Correa

Brasília
2025

Universidade de Brasília
Instituto de Ciências Exatas
Departamento de Ciência da Computação

Mineração de dados socioeconômicos do World Bank

Luthiery Costa Cavalcante
Fernando Ferreira Cordeiro

Monografia apresentada como requisito parcial
para conclusão do Bacharelado em Ciência da Computação

Prof. Dr. Jan Mendonça Correa (Orientador)
CIC/UnB

Profa. Dra. Roberta Barbosa
Universidade de Brasília

Prof. Dr. Wilson Veneziano
Universidade de Brasília

Prof. Dr. Marcelo Grandi Mandelli
Coordenador do Bacharelado em Ciência da Computação

Brasília, 04 de Agosto de 2025

Dedicatória

Dedico esse trabalho aos meus pais por sempre me apoiarem no meu trajeto acadêmico, e que em nenhum momento deixaram de me incentivar a seguir os meus sonhos.

- Fernando Ferreira Cordeiro

Dedicatória

Dedico este trabalho à Luzia e ao Luiz, meus pais. Ao Lindberg e ao Lucas, meus irmãos. E também a meus sobrinhos que tenho com tanto carinho: Allana, Lianna, Agatha, Arthur e Yasmin. Todos vocês, tendo noção disso ou não, me inspiram e me impedem de desistir.

- Luthiery Costa Cavalcante

Agradecimentos

Sou grato aos meus pais, Walb Lenard e Dácia Ferreira, pelo apoio incondicional e pelo incentivo constante ao longo da minha graduação.

Agradeço também à minha prima Maria Leonor, que sempre esteve presente e preocupada com minha trajetória acadêmica, incentivando-me e oferecendo suporte em todos os momentos.

Um agradecimento especial ao professor Jan Corrêa, pela orientação fundamental durante todo o processo do trabalho de graduação, e ao meu amigo Luthiery, que, junto ao professor Jan, se manteve ao meu lado durante todo o período do projeto.

Por fim, sou grato aos amigos que fiz na universidade, pelas risadas e pelas longas madrugadas de estudos. Em especial, quero agradecer ao professor e coordenador Marcelo Mandelli, por sua disposição em ajudar e contribuir durante toda a minha trajetória acadêmica.

- Fernando Ferreira Cordeiro

Agradecimentos

Agradeço aos meus pais por me apoiarem tanto em todos os aspectos ao longo da graduação.

Agradeço ao Sistema de Seleção Unificada (SiSU) e à lei de cotas a oportunidade de entrar na Universidade de Brasília (UnB). Agradeço a esta universidade em especial ao Departamento de Ciência da Computação (CIC) pelo ensino que agregou muito à minha trajetória profissional, acadêmica e ética.

Agradeço ao professor Jan Corrêa por nos sugerir o tema e objetivo do trabalho, assim como por fornecer de antemão trabalhos anteriores de antigos orientandos, poupando tempo precioso na pesquisa bibliográfica. Agradeço, em conjunto, o professor Jan e meu amigo Fernando, pela parceria e por ajudar a manter o projeto vivo apesar dos percalços.

Agradeço também, em especial, a professores com quem tive mais contato ao longo da graduação: Marcos Caetano e Marcus Vinicius Lamar, dos quais fui monitor. Essas experiências me ajudaram a pensar fora da caixa e exercitar o ensino. Agradeço também ao Luis Paulo Garcia, cuja disciplina de IA me despertou o interesse na área. E, claro, ao Marcelo Mandelli, nosso professor e coordenador extremamente prestativo que foi de vital ajuda logo no início e depois na reta final do curso.

Por fim, agradeço aos amigos que formei na universidade - uma lista tão grande que não cabe em um nem dois parágrafos. Vocês ajudaram na minha formação (acadêmica e como pessoa) e fizeram essa jornada valer a pena.

- Luthiery Costa Cavalcante

Resumo

A análise de indicadores socioeconômicos, como o PIB, é fundamental para entender as dinâmicas econômicas e sociais de países e regiões ao longo do tempo. No entanto, a complexidade e a grande quantidade de dados disponíveis, muitas vezes com milhares de atributos, exigem métodos eficientes tanto para extração de informações quanto para previsão de padrões e tendências. Este trabalho utilizou técnicas de mineração de dados e Aprendizado de Máquina, com o apoio do ecossistema *Python* e ferramentas como *Pandas* e *Scikit-Learn*, para abordar esse desafio. A partir de uma base de dados do Banco Mundial, que reúne indicadores socioeconômicos de diversos países, regiões e territórios coletados ao longo das últimas décadas, foi desenvolvido um modelo preditivo para estimar o crescimento do PIB em porcentagem. Além de avaliar a acurácia do modelo final, foram abordadas as etapas de pré-processamento e seleção de atributos, fundamentais para lidar com a alta dimensionalidade dos dados. A modelagem realizada permitiu construir com sucesso um modelo de regressão capaz de estimar a variável desejada, utilizando-se do modelo de seleção de atributos para melhorar sua performance.

Palavras-chave: Python, Pandas, Scikit-Learn, Aprendizado de Máquina, Mineração de Dados, World Bank, Banco Mundial

Abstract

The analysis of socioeconomic indicators, such as GDP, is essential for understanding the economic and social dynamics of countries and regions over time. However, the complexity and the vast amount of available data, often comprising thousands of attributes, require efficient methods for both information extraction and the prediction of patterns and trends. This project employs data mining and Machine Learning techniques, supported by the *Python* ecosystem and tools such as *Pandas* and *Scikit-Learn*, to address this challenge. Using a dataset from the World Bank, which includes socioeconomic indicators from various countries, regions, and territories collected over the past decades, the project develops a predictive model to estimate GDP growth in percentage. In addition to evaluating the accuracy of the final model, the analysis also examines the preprocessing and feature selection steps, which are crucial for handling the high dimensionality of the data. The modeling performed allowed the successful construction of a regression model capable of estimating the desired variable, using the attribute selection model to improve its performance.

Keywords: Python, Pandas, Scikit-Learn, Machine Learning, Data Mining, World Bank

Sumário

Lista de Figuras

Lista de Tabelas

Lista de Abreviaturas e Siglas

AM Aprendizado de Máquina.

API *Application Programming Interface.*

BI *Business Inteligence.*

CAPES Coordenação de Aperfeiçoamento de Pessoal de Nível Superior.

CART *Classification and Regression Trees.*

CIC Departamento de Ciência da Computação.

CRISP-DM *Cross-Industry Standard Process for Data Mining.*

CSV Valores Separados por Vírgula.

DSA *Data Staging Area.*

DW *Data Warehouse.*

ETL *Extract, Transform, Load.*

FN Falso Negativo.

FP Falso Positivo.

IA Inteligência Artificial.

IDH Índice de Desenvolvimento Humano.

IEEE Instituto de Engenheiros Eletricistas e Eletrônicos.

KDD *Knowledge Discovery in Databases.*

KNN *K-Nearest Neighbors.*

ML *Machine Learning*.

PCA Análise de Componentes Principais.

PIB Produto Interno Bruto.

PLN Processamento de Linguagem Natural.

RNA Rede Neural Artificial.

SBC Sociedade Brasileira de Computação.

SGBD Sistema Gerenciador de Banco de Dados.

SI Sistema de Informação.

TI Tecnologia de Informação.

UnB Universidade de Brasília.

USD Dólar americano.

VN Verdadeiro Negativo.

VP Verdadeiro Positivo.

WDI Indicadores de Desenvolvimento Mundial.

Capítulo 1

Introdução

A introdução consiste em apresentar de forma geral o presente trabalho. São contextualizados o problema, objeto e justificativa da pesquisa. Além disso, são definidos de forma clara os objetivos gerais, objetivos específicos e as hipóteses do trabalho. Nas seções finais, são expostas a metodologia do estudo e a estrutura dos demais Capítulos.

1.1 Problema

No ambiente acadêmico, há pesquisas em áreas como Economia, Ciências Sociais e Políticas Públicas, assim como na Ciência de Dados em particular, voltados à identificação de padrões sobre a tendência geral de desenvolvimento econômico de um determinado país ou região. A compreensão dos fatores que influenciam o crescimento ou declínio da riqueza de um Estado ou território é um desafio complexo e de grande relevância, como mencionado por Acemoglu e Robison [1]. No entanto, com milhares de indicadores socioeconômicos disponíveis e aferidos sobre esses Estados ao longo do tempo, torna-se desafiador estabelecer uma resposta definitiva para essa questão. A complexidade desses indicadores, usualmente subestimada, exige abordagens mais rigorosas, uma vez que sua interpretação depende diretamente dos objetivos da pesquisa e do contexto de análise [2]. Para isso, pode ser útil o uso de análise e mineração de dados, pois podem ser usadas para descobrir padrões e relações diversas a partir de dados armazenados em diversos formatos [3].

1.2 Objeto de Estudo

O *World Bank*, ou Banco Mundial [4], é uma instituição financeira internacional que oferece empréstimos e assistência técnica a países em desenvolvimento. Fundado em 1944, o Banco Mundial tem como objetivo reduzir a pobreza e promover o desenvolvimento

sustentável em todo o mundo. Ele trabalha em parceria com governos, organizações não governamentais e setor privado para implementar projetos nas áreas de saúde, educação, infraestrutura, meio ambiente e economia.

O Banco Mundial disponibiliza dados de desenvolvimento global por várias interfaces dentro de sua plataforma: o **DataCatalog** (usada neste projeto), onde é possível baixar bases de dados inteiras em lote, em diversos formatos [5]; as **APIs de Dados** [6] e o **DataBank** [7]. Todos esses métodos, e o motivo para a escolha do primeiro, são mais elaborados na Seção 4.2.1.

Entre muitas bases de dados disponibilizadas pelo Banco Mundial, será usada a de **Indicadores de Desenvolvimento Mundial** ou *World Development Indicators* (WDI) [8]. Trata-se de uma fonte abrangente de informações socioeconômicas e demográficas de países em todo o mundo. As bases de dados do Banco Mundial são amplamente utilizadas no apoio à decisão por pesquisadores, formuladores de políticas e profissionais da área econômica [9] [10].

1.3 Objetivos Gerais

Esta seção apresenta os objetivos gerais e específicos deste trabalho. Os objetivos gerais consistem em investigar padrões úteis sobre o crescimento ou decrescimento do indicador econômico do Produto Interno Bruto (PIB) de diversos países e territórios utilizando a linguagem de programação *Python*; demonstrar o poder do *Python* em conjunto com as bibliotecas *Scikit-Learn* e *Pandas* para a análise de dados descritiva, modelagem com Aprendizado de Máquina e seleção de atributos, em outras palavras, demonstrar sua adequação para as mais diversas tarefas de mineração de dados econômicos, incluindo particularmente o caso de tarefas de mineração sobre bases de dados com muitos registros e características.

1.3.1 Objetivos específicos

Usar as bibliotecas *Scikit-Learn* e *Pandas* e a linguagem *Python* para:

- Manipular a estrutura dos dados do Banco Mundial e realizar tratamento de valores vazios nos mesmos, através da exclusão e inferência de valores;
- Reduzir a quantidade de atributos da base, selecionando os mais importantes para a modelagem;
- Por fim, construir um modelo capaz de aproximar, ou predizer, o indicador (atributo) de "Crescimento anual do PIB (em porcentagem)" a partir do conjunto de treinamento com os indicadores remanescentes.

1.4 Hipóteses

As seguintes hipóteses estão relacionadas ao trabalho com a proposta de expandir o entendimento sobre a capacidade de modelagem das ferramentas, assim como expandir o conhecimento já obtido pelos estudos relacionados:

Hipótese 1: Os modelos gerados pelos algoritmos da biblioteca *Scikit-Learn* possibilitam: uma valoração numérica da sua qualidade; a derivação de uma grande quantidade de informação sobre a base de dados; o reconhecimento de padrões a partir da visualização do modelo gerado — em particular, padrões preditivos sobre a evolução do indicador do PIB; e a seleção e visualização dos atributos mais relevantes para explicar tais padrões, sendo este o foco da presente investigação.

Hipótese 2: A utilização da biblioteca *Scikit-Learn* para selecionar (filtrar) atributos de forma automática, em uma base de dados com uma quantidade suficientemente grande de atributos, é mais eficiente para o analista e gera um modelo melhor do que aquele gerado com uma seleção manual de atributos.

1.5 Justificativa

O Produto Interno Bruto (PIB) é um dos principais indicadores econômicos utilizados para medir a atividade produtiva de um país, refletindo o valor de bens e serviços produzidos em um determinado período. Como destacado por Nordhaus e Samuelson [11], o PIB é amplamente utilizado para avaliar o desempenho econômico de nações e embasar políticas públicas. No entanto, o crescimento ou decrescimento do PIB é influenciado por uma complexa rede de fatores socioeconômicos, o que torna desafiador identificar padrões claros e prever tendências futuras.

Nesse sentido, surge a necessidade de estudos que analisem grandes volumes de dados socioeconômicos, como os disponibilizados pelo Banco Mundial, a fim de extrair informações úteis e *insights* que contribuam para a compreensão do desenvolvimento econômico de diferentes países. Este trabalho propõe a utilização de um conjunto específico de ferramentas (ver Seção 4.1), combinando técnicas de mineração de dados e aprendizado de máquina para identificar padrões e relações complexas entre os diversos indicadores disponíveis, a partir da mesma base de dados. A fundamentação teórica e a relevância do tema são discutidas no Capítulo 3, no qual são apresentadas as referências que fundamentam esta pesquisa.

1.6 Metodologia

A primeira etapa consiste na coleta manual dos dados contendo países e seus indicadores a partir da aplicação *web* do Banco Mundial.

Foi realizada uma revisão bibliográfica abrangente sobre conceitos fundamentais de mineração de dados, aprendizado de máquina e indicadores socioeconômicos, a fim de embasar teoricamente as etapas e decisões metodológicas adotadas neste estudo. Estes conceitos são detalhados no Capítulo 2. Dentro da revisão bibliográfica, também foi aplicada uma metodologia de pesquisa para encontrar trabalhos relacionados a este, detalhada no Capítulo 3.

Por meio das ferramentas abordadas na Seção 4.1, os *scripts* construídos são hospedados em um repositório público no *GitHub*[1], juntamente com os dados extraídos, o que possibilita a colaboração mútua, cópia, execução local e *feedback* da comunidade. Tais *scripts* devem ser capazes de:

- tratar os dados brutos originalmente extraídos e os exportar para um novo arquivo tratado;
- realizar o pré-processamento de dados;
- selecionar os atributos mais relevantes;
- construir um modelo de Aprendizado de Máquina (AM) com fim de realizar predições conforme os objetivos do trabalho;
- gerar gráficos para a visualização da base de dados e do modelo gerado.

Os *scripts* possuem parâmetros definidos nas suas linhas iniciais, que impactam a etapa de pré-processamento e a geração do modelo. O modelo pode ser gerado algumas vezes, com parâmetros diferentes, a fim de comparação de resultados.

Os resultados gerados devem ser facilmente reproduzíveis em qualquer ambiente ou sistema operacional em que o projeto esteja sendo executado, o que é garantido pela estrutura do mesmo, detalhada na Seção 4.

Os gráficos gerados, assim como a avaliação numérica (*score*) da qualidade dos modelos gerados, serão utilizados para avaliar o resultado do modelo e fazer conclusões sobre as hipóteses levantadas.

1.7 Estrutura do Trabalho

Os próximos Capítulos deste trabalho estão estruturados da seguinte forma:

[1]Disponível em: `https://github.com/luthierycosta/tg2`. Acesso em: 25/07/2025.

Capítulo 2 - Revisão teórica: Abrange definições úteis para o entendimento do trabalho, com base em pesquisa bibliográfica.

Capítulo 3 - Trabalhos relacionados: Discorre sobre trabalhos similares encontrados na bibliografia que se relacionam com o presente trabalho em seus objetivos e objetos de análise abordados.

Capítulo 4 - Mineração de dados de Indicadores de Desenvolvimento Mundial: Apresenta o desenvolvimento metodológico, desde a coleta de dados até a aferição dos resultados das modelagens.

Capítulo 5 - Resultados: A análise e avaliação sobre o conhecimento obtido.

Capítulo 6 - Conclusões: Nessa Seção, é analisada a qualidade do próprio estudo, e as hipóteses são retomadas para concluir se as mesmas foram observadas.

Ao fim, nos apêndices, estão inclusos todos os *scripts* confeccionados, assim como informação adicional sobre a base de dados objeto.

Capítulo 2

Revisão teórica

Diversos conceitos relacionados à mineração de dados possuem grau de importância para o entendimento do projeto e precisam ser contextualizados para o entendimento deste trabalho, em uma ordem lógica. Na Seção 2.1, é abordada a própria definição de dado e como obter conhecimento a partir de dados. As seções 2.1.2 e 2.2 discorrem sobre estruturas de armazenamento e agrupamento de dados, na forma de Bancos de Dados e *Data Warehouses*. A Seção 2.3 define o conceito de Mineração de Dados, seus processos e algumas de suas aplicações mais comuns. A Seção 2.4 aborda a teoria inicial sobre Inteligência Artificial, que serve de base para a formulação do Aprendizado de Máquina (AM), descrito na Seção 2.5. Nessa mesma Seção, são contextualizados e diferenciados o aprendizado supervisionado do não-supervisionado, exemplos de algoritmos em cada categoria - em particular os que serão utilizados na presente análise - assim como o conceito de redução de dimensionalidade.

2.1 Dado, Informação e Conhecimento

Castro e Ferrari [12] fornecem definições sucintas para esses conceitos: **dados** são símbolos ou sinais não estruturados, sem significado inerente; a **informação** é obtida nas descrições ou manipulações sobre esses dados, agregando significado e utilidade; já o **conhecimento** é algo que permite uma tomada de decisão para agregar valor de negócio ao que está sendo trabalhado, a partir da interpretação das informações obtidas.

Dados são sequências de fatos ainda não analisados - armazenados em um formato arbitrário, seja uma tabela em um banco de dados, uma planilha, um caderno físico etc. - representativos de eventos que ocorrem em um certo contexto organizacional, antes de terem sido organizados e dispostos de forma que as pessoas possam entendê-los e usá-los. Informação é um conjunto de dados que já foi organizado e modelado em um formato significativo útil para as pessoas. A transformação de dado em informação pode ser feita

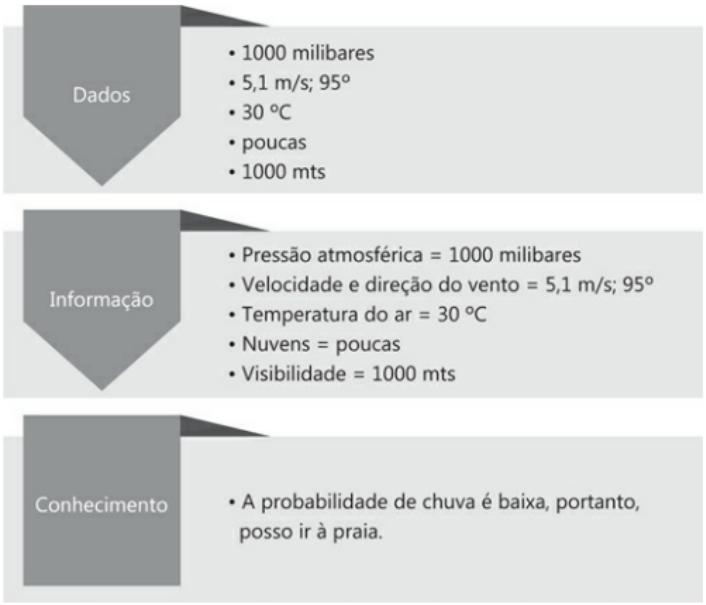

Figura 2.1: Exemplo abstrato de transformação de dado em conhecimento (Fonte: [12]).

através de um Sistema de Informação (SI) [13]. A geração de conhecimento é a etapa (geralmente) humana de analisar e interpretar as informações de forma que seja útil para as decisões de negócio de uma organização ou, em um contexto de análise experimental, para a descoberta de padrões na base de dados, por exemplo.

Elmasri e Navathe [14] fornecem definições complementares: a de conhecimento *dedutivo*, que deduz novas informações com base na aplicação de regras lógicas previamente especificadas sobre os dados; e a de conhecimento *indutivo*, que descobre novas regras e padrões a partir dos dados.

2.1.1 Sistemas de Informação

Em um contexto organizacional, um Sistema de Informação (SI) pode ser definido como um conjunto de componentes inter-relacionados que coletam, processam, armazenam e distribuem informações destinadas a apoiar a tomada de decisões, a coordenação e o controle em uma organização. Além disso, os sistemas de informação também auxiliam os gerentes e trabalhadores a analisar problemas, visualizar assuntos complexos e criar novos produtos. Um SI também contém informações sobre pessoas, locais e itens significativos para a organização ou ambiente que os cerca [13].

Há três atividades em um SI responsáveis por gerar as informações necessárias às decisões, operações, análise de problemas e o desenvolvimento de novas soluções [13]. São elas:

entrada: que coleta ou recupera dados brutos dentro de uma organização ou de seu ambiente externo;

processamento: que converte, manipula e mapeia esses dados em uma forma mais significativa (informação);

saída: que transfere as informações processadas às pessoas que as utilizarão ou às atividades nas quais serão empregadas.

O termo Sistema de Informação (SI) por vezes é confundido com o de Tecnologia de Informação (TI). Embora os SI informatizados utilizem a tecnologia de computadores para processar dados brutos e transformá-los em informações inteligíveis, existe uma diferença entre hardware e software, de um lado, e um SI, de outro. Os computadores são os equipamentos que armazenam e processam as informações. Os programas de computador ou software são conjuntos de instruções operacionais que dirigem e controlam o processamento do computador. Eles são apenas parte de um Sistema de Informação. Além do emprego de Tecnologia de Informação, os SI compreendem também uma natureza organizacional e humana [13].

2.1.2 Bancos de Dados

Elmasri e Navathe [14] definem um **banco de dados** (tradução de *database*, do inglês) como uma coleção de dados relacionados logicamente, estruturados com algum significado inerente, o qual é projetado, construído e populado para uma finalidade específica. Um banco de dados tem alguma fonte da qual se derivam os dados presentes, um grau de interação fidedigna com eventos do mundo real (esses eventos compõem o que são chamados de *minimundo* ou *universo de discurso* do banco de dados) e um público ativamente interessado em seu conteúdo, que pode consultá-lo ou fazer alterações no mesmo.

Bancos de Dados podem ser gerados e mantidos manualmente (em uma agenda, por exemplo, onde o processo de alteração ou inclusão de dados é lento) ou de forma computadorizada [14]. No mundo digital em que vivemos, o termo *banco de dados* quase sempre se referem àqueles controlados de forma computadorizada através de um Sistema Gerenciador de Banco de Dados (SGBD) - eles permitem com que os bancos de dados cresçam exponencialmente em volume de dados armazenados sem que isso impossibilite o trabalho de consulta e alteração dos mesmos.

Um Sistema Gerenciador de Banco de Dados é definido como uma coleção de programas que facilita os processos de:

Definição do banco de dados - especificar os tipos, estruturas, domínios e restrições dos dados a serem armazenados, especificação que também é armazenada no SGBD com o nome de *metadados*;

Construção do banco de dados - especificar o armazenamento dos dados em um meio controlado pelo SGBD;

Manipulação do banco de dados - inclui funcionalidades de consulta para recuperar, atualizar (por meio de transações) ou gerar relatórios relativos a esses dados;

Compartilhamento do banco de dados entre diversos usuários, programas e sistemas da *web* de forma simultânea;

Proteção do sistema de banco de dados contra falhas de hardware (geralmente por meio de redundância) assim como proteção de segurança (por meio de controle de acesso);

Manutenção do sistema, permitindo que o banco de dados tenha um ciclo de vida longo e evolua conforme seus requisitos de negócio mudam.

Um outro termo que vemos ser usado por autores como Castro e Ferrari [12] é o de **base de dados**, definida pelos mesmos como: "coleção organizada de dados, ou seja, valores quantitativos ou qualitativos referentes a um conjunto de itens, que permite uma recuperação eficiente dos dados" - definição bastante similar à do primeiro parágrafo desta Seção. Ainda segundo os autores, "conceitualmente, os dados podem ser entendidos como o nível mais básico de abstração a partir do qual informação e, depois, conhecimento, podem ser extraídos", processo ilustrado na Figura 2.1.

Na bibliografia de bancos de dados, autores como Elmasri e Navathe [14] e Silberschatz et al. [15] não se preocupam em fazer distinção entre os termos *banco de dados* e *base de dados* - de fato usando apenas o primeiro. Nessa concepção, base de dados seria um banco de dados que, todavia, não está arquitetado sob nenhum SGBD, tendo uma estrutura arbitrária (mais flexível) [1] . Para os que fazem a distinção entre os termos, um banco de dados estaria intrinsecamente ligado ao uso de SGBD, sendo necessário usar outro termo para coleções lógicas digitais de dados que não estão estruturados sob eles [16].

O termo base de dados costuma ser mais utilizado em contextos de análise exploratória de dados e artigos relacionados aos conceitos de *Big Data*, conforme Cavique [17], e

[1] A confusão na terminologia é agravada já que *base de dados* também é uma tradução aceitável para *database*. No contexto deste trabalho, onde não são usados SGBDs, a diferença conceitual entre os termos acaba não sendo relevante. Portanto, eles serão usados de forma sinônima, podendo ainda, no lugar, ser usado o termo *conjunto de dados*.

Inteligência Artificial (IA), nos quais o conhecimento a ser extraído desses dados - e como extrair - é a parte crucial, independente da estrutura em que os mesmos estão implementados. Alguns exemplos de artigos nesse escopo são os trabalhos do Capítulo 3.

2.2 *Data Warehouse*

Um armazém de dados - ou *Data Warehouse* (DW) - oferece armazenamento, funcionalidade e responsividade a consultas além das capacidades de um banco de dados tradicional. Surgiu como consequência do crescente poder de processamento e a sofisticação das ferramentas e técnicas analíticas [14], motivados pela tecnologia de armazenamento dos SGBDs, o advento das aplicações online e a "bagunça" que essa inundação de dados criou nas organizações [18].

Existe uma distinção clara entre um banco de dados tradicional e um *Data Warehouse*: ambos são coleções logicamente coerentes de informações, acompanhadas por softwares de suporte e gerenciamento. Contudo, os bancos de dados são **transacionais** - sejam eles com implementação relacional, orientada a objetos, em rede ou hierárquica - enquanto os DW servem principalmente para aplicações de **apoio à decisão**, sendo otimizados para recuperação de dados, em vez de processamento de transações de rotina [14].

Enquanto SGBDs relacionais são otimizados para processar consultas que envolvem uma parte pequena do banco de dados e transações que lidam com inserções e atualizações em uma relação (tabela) por vez, *Data Warehouse*s são projetados para dar suporte à extração, processamento e apresentação eficientes para fins analíticos e de tomada de decisão. Eles contêm quantidades muito grandes de dados de várias fontes, que podem incluir até bancos de dados de diferentes modelos, assim como arquivos adquiridos de sistemas e plataformas independentes [14].

Dessa forma, Elmasri e Navathe [14] caracterizam os *Data Warehouse* como uma coleção de tecnologias de apoio à decisão, visando a habilitar o trabalhador do conhecimento (por exemplo, executivo, gerente, analista) a tomar decisões melhores e mais rápidas. Porém, não existe uma única definição canônica para esse termo. Outros autores, como Inmon et al. [18] definem um *Data Warehouse* como uma coleção de dados para processamento de informações, a qual, além de cumprir a função de apoio à decisão, deve estritamente possuir as seguintes propriedades:

Integrado: os dados são reunidos no DW a partir de diversas fontes, de naturezas e formatos arbitrários, agrupados a fim de gerar uma consistência coerente;

Orientado a assunto: esse agrupamento traz informações sobre um domínio específico e determinado;

Não-volátil: não deve haver mutabilidade nos dados. Dessa forma, são adicionados e agregados novos dados sem alterar os registros anteriores, funcionando como um registro temporal para a organização;

Variável ao longo do tempo: Os dados são identificados a cada período determinado de tempo.

2.2.1 *Business Inteligence*

Vários autores, como Kimball et al. [19], colocam os *Data Warehouse*s como parte fundamental e inseparável da Inteligência de Negócio, ou *Business Inteligence* (BI).

De acordo com Santana e Carvalho [20], o BI abrange conjuntos de ferramentas e técnicas que se dedicam à obtenção, análise, organização, compartilhamento e monitoramento de dados dentro dos bancos de dados, oferecendo suporte à decisão e à gestão de negócios. Essas tecnologias são capazes de suportar uma enorme quantidade de dados, mesmo desestruturados.

Segundo Kimball et al. [19], o termo BI teria começado a ser usado nos anos 1990, referindo-se à construção de relatórios e à análise dos dados guardados nos armazéns - no início, existiam inúmeras organizações que usavam os *warehouses* como meros depósitos de registros brutos, sem consideração ao processo de entregar dados e conhecimentos organizados aos usuários do negócio de forma útil, levando à cunhagem desse novo termo para se referir à entrega de valor por meio dos DWs.

Na indústria, alguns tratam o BI como uma parte (ou etapa) do *data warehousing*, que seria o processo como um todo, enquanto há outros que fazem o contrário: consideram os DWs a camada central de dados e processos relacionados, contidos dentro de uma estratégia maior de *Business Inteligence*. Por isso, Kimball et al. [19] propõem o termo conjunto DW/BI para se referir a todo o conjunto, reforçando sua interdependência; todos os dados recuperáveis no sistema DW/BI estão no *Data Warehouse* do negócio, enquanto as ferramentas de análise e geração de valor são *aplicações de BI*.

2.2.2 ETL

Uma das bases que estruturam o *data warehousing*, ou DW/BI, é o *Extract, Transform, Load* (ETL) - inglês para Extração, Transformação e Carga. Em uma analogia, Kimball et al. [19] definem o ETL como a "cozinha" que mantém o "restaurante" do *BI* em funcionamento. Dados brutos são extraídos das fontes de dados organizacionais e levados à "cozinha", onde eles são transformados em informação útil ao negócio. Tal espaço precisa ser arquitetado bem antes de haver dados para trafegar; ele é projetado para entregar a maior vazão de dados possível; também é desenhado para entregar um produto final com

a maior qualidade possível; os insumos que chegam a ele devem ser checados quanto à sua integridade e qualidade; e os próprios critérios de qualidade são frequentemente revisados.

Santana e Carvalho [20] descrevem os processos de extração, transformação e carga, da seguinte forma:

Extração: Rotinas de extração são executadas sobre as diversas fontes de dados, podendo ser sistemas de bancos de dados, sistemas operacionais ou até planilhas e arquivos de texto. Procura-se a capacidade de ler e extrair dados em diversos formatos, além de integrá-los, sem causar perda de informações em relação às fontes originais.

Transformação: Os dados extraídos são propagados para a chamada *Data Staging Area* (DSA), onde os dados serão manipulados sem a necessidade de consultas às bases originais. A transformação consiste em adequar os dados às necessidades e restrições do modelo DW em questão, de forma a garantir a qualidade, consistência e limpeza dos dados.

Carga: Rotinas de carga são responsáveis por entregar os dados já integrados ao *Data Warehouse* em si, respeitando restrições de integridade e criando uma visão concreta unificada dos dados extraídos.

É necessário notar que não serão usadas aplicações de DW/BI no presente trabalho, optando-se por uma estratégia de armazenamento de dados mais direta e simples usando arquivos em formato de Valores Separados por Vírgula (CSV), e realização de manipulação sobre essa massa usando scripts na linguagem Python. Contudo, no tratamento inicial da massa de dados objeto são aplicadas técnicas análogas a um ETL (processo detalhado na Seção 4.2.1), de forma que se fez útil a definição de todos os conceitos relacionados dentro desta Seção.

2.3 Mineração de Dados

Mineração de Dados é o processo que tem como objetivo descobrir padrões e tendências em grandes conjuntos de dados [21] por meio de algoritmos e outros sistemas de informação. O termo é uma alusão ao conceito "original" de mineração, que é a extração de minerais valiosos, como ouro e pedras preciosas, a partir de uma fonte - uma mina. No contexto computacional, é um campo vasto e multidisciplinar que envolve áreas como bancos de dados, estatística, Aprendizado de Máquina (AM), engenharia, Sistemas de Informação (SI), Redes Neurais Artificiais (RNA), processamento de sinais e visualização de dados [12], entre outras.

No contexto histórico mundial atual, há uma quantidade enorme e exponencialmente crescente de informação sendo coletada por sistemas corporativos, governamentais, entre muitos outros. Entretanto, segundo Larose e Larose [21], essa informação não necessariamente se converte inteiramente em conhecimento, devido à relativa escassez de analistas de dados no mercado. A demanda por profissionais especializados em dados foi criada, segundo os mesmos autores, por essa oferta gigante de dados coletados, a evolução da tecnologia em sistemas de bancos de dados, o aumento no volume de acesso a esses sistemas por vários usuários simultaneamente e a própria evolução do poder de processamento e de armazenamento dos computadores atuais.

2.3.1 KDD

A mineração de dados está englobada no processo de Descoberta de Conhecimento em Bancos de Dados (ou *Knowledge Discovery in Databases* (KDD)), como uma de suas etapas. O processo de KDD sempre recebe um banco de dados e, a partir de regras de negócio e objetivos específicos dos analistas, submete essa base a uma série de processos para obter um determinado conjunto de conhecimentos ao final. Sejam as seis etapas do KDD, a seguir, descritas por Elmasri e Navathe [14]:

Seleção de dados: filtragem da amostra de registros (linhas) [2] e/ou atributos (colunas) [3] específicos;

Limpeza de dados: onde ocorre o tratamento de atributos inválidos e/ou exclusão de registros com dados incorretos (segundo as regras de negócio determinadas);

Enriquecimento: melhora a base de dados, por exemplo, inserindo atributos processados (derivados) a partir de outros atributos e fontes de dados externas ao banco;

Transformação de dados: pode ser usado para reduzir a quantidade de dados, por meio do mapeamento (ou codificação) de um domínio extenso de valores de um atributo em um domínio menor;

Mineração de dados: etapa em que os dados são submetidos a técnicas que processam esses dados com o objetivo de descobrir padrões e obter conhecimento (algumas técnicas serão detalhadas na Seção 2.3.3);

Exibição: etapa onde o conhecimento obtido é organizado, condensado e exposto em um formato logicamente agradável ao público interessado nos conhecimentos obtidos.

[2] Os termos “registro”, “instância”, “tupla”, “dado”, “objeto” e “linha” são usados de forma sinônima e intercambiável na bibliografia.

[3] Os termos“atributo”,“característica”,“variável” e“coluna” são usados de forma sinônima e intercambiável na bibliografia.

2.3.2 CRISP-DM

O *Cross-Industry Standard Process for Data Mining* (CRISP-DM) é um padrão multi-industrial, independente de ferramentas e aplicações, desenvolvido por analistas para representar as etapas da mineração de dados. Segundo Larose e Larose [21], esse padrão serve o objetivo de resolver problemas dentro de uma organização (ou projeto de pesquisa) através do uso de mineração de dados, sendo um padrão não-proprietário e gratuito.

Um projeto de mineração de dados qualquer, segundo o CRISP-DM, tem um ciclo de vida iterativo e adaptativo baseado em seis etapas. Isso significa que a decisão de qual será a próxima etapa por vezes depende do resultado da etapa anterior - por vezes, os projetos precisam voltar a etapas anteriores para serem refinados antes de prosseguir para os próximos passos.

Assim seguem as etapas do CRISP-DM para projetos de mineração de dados [21]:

Entendimento do negócio (ou pesquisa): Enunciar os objetivos, escopo e requisitos do projeto em termos de decisões negociais ou dos pesquisadores. A partir disso, traduzir tais requisitos para uma definição de um problema de mineração de dados e preparar uma estratégia preliminar para o cumprimento dos objetivos.

Entendimento dos dados: Coletar os dados em si. Em seguida, usar análise exploratória de dados para se familiarizar com os mesmos. Por fim, avaliar a qualidade dos dados e, se for o caso, detectar subconjuntos de dados que sejam mais úteis aos objetivos negociais.

Preparação dos dados: Aqui entram todos os aspectos de preparação da base de dados final, a partir da coleta bruta inicial dos dados. É necessário selecionar casos, variáveis e subconjuntos que são mais apropriados para a análise; realizar transformações em certas variáveis (atributos) para tornar mais fácil a manipulação; e filtrar o conjunto de dados para maximizar a utilidade ferramentas de modelagem.

Modelagem: Escolher, aplicar e calibrar as técnicas de modelagem que serão usadas no problema, de forma a otimizar os resultados. Por vezes, várias técnicas podem ser aplicadas em um mesmo processo de mineração (aqui entra a natureza adaptativa dos projetos - a base de dados pode precisar voltar à fase de preparação para se adaptar a cada modelo particular).

Avaliação: Os *modelos* gerados na etapa anterior precisam ser avaliados segundo sua qualidade, acurácia e efetividade. É necessário também certificar-se que o modelo atende os requisitos definidos na 1ª etapa. Caso não os atenda, pode ser necessário voltar e escolher outra técnica de modelagem.

Implantação: O modelo gerado não é a resposta do problema, mas sim o uso desse modelo para gerar conhecimento - essa é a etapa final chamada implantação. Um exemplo muito comum é gerar e expor relatórios sobre os resultados obtidos. Outro exemplo seria usar esse modelo como insumo para outros projetos relacionados de mineração de dados.

2.3.3 Tarefas em Mineração de Dados

As tarefas - ou funcionalidades - da mineração de dados especificam os tipos de informações que podem ser obtidas pela mineração. Segundo Castro e Ferrari [12], as tarefas podem ser separadas em duas categorias: descritivas e preditivas. As descritivas dispõem sobre propriedades gerais da base de dados e dos seus atributos; as preditivas fazem inferência a partir dos dados almejando fazer predições de valores. Frequentemente, esse segunda categoria de tarefas está associada ao uso de algoritmos de Aprendizado de Máquina (exemplificados na Seção 2.5). Em muitos casos, é incerto ao analista a escolha de qual funcionalidade aplicar ao problema de mineração, tornando importante a capacidade dessas ferramentas de se adaptar e encontrar conhecimento por vários caminhos.

Castro e Ferrari [12] e Han et al. [3] descrevem as principais tarefas de mineração de dados:

Análise descritiva de dados

É uma funcionalidade que não requer nível elevado de complexidade algorítmica. Consiste de ferramentas capazes de medir e descrever características intrínsecas aos dados. Particularmente, obtém-se medidas estatísticas como a distribuição de frequência, medidas de centro (média, mediana e moda), variância, medidas de posição relativa e associação dos dados, além de técnicas elementares para visualização de dados como a *plotagem*[4] de gráficos.

As análises descritivas permitem sumarizar e compreender os objetos presentes na base de dados e seus atributos. É possível, a partir de um exemplo (ou registro) da base, determinar o quão semelhante ele é da média de todos os registros - e outras deduções estatísticas. Também é possível detectar padrões a partir da visualização dos registros em gráficos de diversos tipos, como histogramas (distribuições de frequência).

[4]Termo técnico que, na Ciência de Dados e outras áreas relacionadas à Ciência da Computação, se refere ao ato de gerar gráficos de forma algorítmica e parametrizada a partir de uma fonte de dados.

Predição - classificação e regressão

Predição é a terminologia usada para se referir à construção de modelos preditivos - seja para determinar o rótulo (ou classe[5]) de um objeto (registro) não-rotulado, no qual a tarefa é chamada de classificação, ou para estimar o valor de um ou mais atributos de um objeto, no qual a tarefa é denominada regressão ou estimação. Temos nessas duas tarefas os principais tipos de problemas de predição, sendo que a classificação é usada para predizer valores discretos (não necessariamente numéricos), enquanto a regressão é usada para predizer valores contínuos (comumente numéricos).

Um exemplo simples usado por Castro e Ferrari [12] para distinguir os dois tipos de tarefas é o de crédito pessoal: um cliente se dirige a uma instituição bancária com objetivo de obter um financiamento. O cliente, dessa forma, é um objeto (ou registro) da base de dados da instituição; usa-se a classificação para determinar se ele deve ou não ter acesso a crédito (particularmente, trata-se de uma classificação *binária*, com apenas dois valores possíveis); além de um modelo de regressão para determinar qual o valor, em dinheiro, que tal cliente deve obter.

O uso de mineração de dados neste projeto, que visa responder perguntas sobre a base de dados do Banco Mundial, consiste em um problema de regressão. Na Seção 2.5, veremos que os modelos criados a partir de Aprendizado de Máquina para resolver problemas de classificação e regressão se encaixam dentro da categoria chamada de aprendizado *supervisionado*.

Agrupamento

O agrupamento (em inglês *clustering*) é o nome dado à tarefa de separar, particionar ou segmentar um conjunto de objetos em grupos (ou *clusters*), que são montados com base nas semelhanças entre os atributos dos objetos. Um *cluster* pode ser definido como um subconjunto de objetos similares uns aos outros e diferentes dos objetos pertencentes a outros *clusters*. O objetivo é determinar classes de objetos similares, separados em subconjuntos.

Na Seção 2.5, veremos que os modelos feitos com Aprendizado de Máquina para executar tarefas de agrupamento se encaixam dentro da categoria chamada de aprendizado *não-supervisionado*.

[5]Ambos os termos "rótulo", "classe", "alvo" e "classe-alvo" são usados de forma sinônima e intercambiável na bibliografia

Associação

Em tarefas preditivas e de agrupamento, o objetivo é encontrar relações - sejam de similaridade, classes ou estimativas - entre objetos de uma base. Entretanto, em tarefas de associação, o objetivo é determinar relações entre os próprios atributos (variáveis) presentes na base, ao invés de seus registros (tuplas).

A análise por associação, também chamada de mineração de regras de associação, se encarrega da descoberta de regras de associação que apresentam valores de atributos que ocorrem concomitantemente em um conjunto de dados, ou seja, atributos que podem ser determinados por (ou variam em função de) outros atributos.

Há dois aspectos centrais para a construção dessas regras: a proposição (construção) eficiente das regras e a quantificação da significância das mesmas. Um bom algoritmo de mineração de regras de associação precisa ser capaz de propor associações que sejam estatisticamente relevantes dentro do universo representado pela base de dados.

Detecção de anomalias

Objetos ou tuplas que não seguem um comportamento ou característica comum entre os dados da base, ou que possuem valores de atributos muito díspares da média, são chamados de anomalias, exceções, ruído, valores discrepantes ou *outliers*. Como eles geralmente não conseguem ser representados pelo modelo gerado, a maioria das ferramentas de mineração os descarta - caso contrário, a capacidade de predição, associação ou descrição do modelo pode ser impactada negativamente. Há outras tarefas, no entanto, que têm o objetivo de ressaltar (e não deletar) as anomalias encontradas, como em algoritmos de detecção de fraude bancária.

As anomalias podem ser detectadas de várias formas, incluindo métodos estatísticos que obtêm uma distribuição ou modelo de probabilidade dos dados, ou medidas de distância por meio das quais objetos muito distantes dos demais são considerados anomalias.

2.3.4 Pré-processamento de Dados

O pré-processamento de dados, ou tratamento de dados, abrange tarefas que se enquadram dentro das fases de Entendimento dos Dados e de Preparação dos Dados dentro do padrão CRISP-DM [21].

Muitos dos dados brutos coletados nos bancos de dados são incompletos e contêm informação ruidosa que não será aproveitada. Por exemplo, essas bases podem conter: campos que são obsoletos e redundantes; valores faltantes; *outliers*; dados em formato não apropriado ou compatível com os modelos de modelagem de dados; e valores inconsistentes com o observado na realidade ou fora de conformidade com as regras de negócio [21]. O

pré-processamento de dados é necessário para resolver tais problemas e atribuir **qualidade** aos dados. A qualidade de um banco de dados consiste de propriedades como a acurácia, completude, consistência, temporalidade, credibilidade e interpretabilidade [3].

A seguir veremos técnicas como a remoção ou inferência de valores vazios, redução do ruído nos dados, identificação de *outliers* e correção de inconsistência, que são definidas como técnicas de **limpeza de dados** [12].

Valores vazios

Um valor ausente numa base de dados constitui-se em um valor que, por algum motivo, foi ignorado ou não foi observado no momento de coleta. Ele costuma ser representado por um código de ausência, que pode ser um valor específico, como um caractere específico (por exemplo, "`?`"), o número 0, um dado de tipo `NaN`[6] ou um valor em branco [12].

É necessário à maioria dos algoritmos de modelagem uma grande massa de dados para seu correto funcionamento. Ao mesmo tempo, entretanto, é frequente encontrar uma grande proporção de valores ausentes - particularmente em grandes bancos de dados, com muitos dados e atributos [21].

Muitos dos algoritmos de modelagem não funcionam corretamente ao se deparar com valores ausentes. Nesses casos, é necessário algum tratamento - os mais frequentes são a remoção de dados (seja de registros ou de atributos) e também a inferência ou imputação de dados, que consiste em preencher os valores ausentes seguindo alguma fórmula consistente, a fim de manter o banco de dados populado com valores significativos [22] [12].

A imputação de valores ausentes assume que a ausência de valores implica em perda de informação relevante para a eficácia da mineração. Assim, os valores que serão inferidos não devem somar nem subtrair informação da base - isto é, não deve enviesá-la [12]. Já a remoção de tuplas ou de atributos com valores ausentes, segundo Han et al. [3], é uma técnica trivial, não muito benéfica visando a eficácia do algoritmo, visto que pode acabar sacrificando dados válidos úteis consigo. Ela pode ser útil, no entanto, para descartar atributos que sabidamente não interferirão no resultado da mineração.

São estratégias comuns para a inferência ou imputação de dados [12] [3]:

- Preencher valores ausentes manualmente, segundo o conhecimento do próprio analista - tarefa inviável em cenários de grandes bases de dados;

[6] Abreviação de "Not a Number", conceito computacional para a representação de números não definidos ou inválidos.

- Preencher todos os valores ausentes com um único valor constante, como o número 0: método simples, mas que pode prejudicar o resultado da mineração ao analisar tal valor como tendo um significado;
- Preencher valores com alguma medida de tendência central (como a média ou mediana) correspondente a cada atributo;
- Preencher valores usando medidas de tendência central dos registros que possuem a mesma classe do registro em questão;
- Preencher valores usando como base os valores válidos dos objetos mais similares - a similaridade pode ser medida usando uma comparação categórica ou medidas de distância entre objetos (método chamado de imputação *hot-deck*);
- Preencher valores a partir da última observação - atribui os valores válidos logo anteriores encontrados na base (parte da premissa que a base está ordenada sob algum critério lógico);
- Preencher valores ausentes usando o valor mais *provável* - para isso, são empregadas diversas técnicas preditivas como regressão, inferência estatística a partir da teoria Bayesiana ou indução com árvores de decisão. Esse é o método mais popular, por usar uma maior quantidade de informação sobre a própria base para inferir valores.

Dados ruidosos

Problemas na coleta de dados podem também gerar dados com erros. Frequentemente, esses erros se tornam parte indissociável dos dados e não podem ser removidos facilmente. O acúmulo desses erros e distorções, além da presença de *outliers*, que são dados fiéis ao observado, porém discrepantes dos padrões, é chamado de ruído [12] [3].

Existem métodos para "suavizar" os dados na base e reduzir o ruído, apesar de não haver um padrão consistente que permita a identificação dos mesmos, o que ocasiona um mínimo irredutível de ruído [12]. Alguns deles são:

- Encaixotamento *(binning)*: distribuir os valores de um atributo em "caixas", de forma que elas tenham o mesmo tamanho, ou abranjam um mesmo intervalo;
- Agrupamento *(clustering)*: encontrar grupos de objetos similares e passar a referir-se aos objetos pelo seu grupo, ou por um objeto específico do mesmo;
- Aproximação: aproximar os dados por alguma função ou modelo paramétrico.

Dados inconsistentes

A falta da propriedade de consistência, em uma base de dados, influencia na validade, utilidade e integridade de uma aplicação de mineração de dados. Um exemplo é o uso de nomes diferentes para se referir a uma mesma característica, que, aos olhos do algoritmo de mineração, seriam tratados como rótulos diferentes. Outro exemplo ocorre quando valores apresentados não condizem com o domínio dos atributos dados. Soluções frequentes envolvem rotinas manuais desenvolvidas pelos analistas responsáveis [12].

Além das técnicas de limpeza de dados vistas acima, são parte do pré-processamento as tarefas de **integração de dados** (que consistem em unir dados oriundos de diversas fontes e assegurar a integridade e consistência dessa junção, observando a remoção de redundâncias, duplicidades e conflitos), assim como as de **redução de dados** e **redução de dimensionalidade** [12] [3], que serão mais detalhadas na Seção 2.5.4.

Normalização de dados

A normalização (ou escalonamento) de dados [3] [23] consiste em aplicar uma escala de grandeza comum para todos os valores analisados, de forma proporcional, para atenuar a possível influência dos atributos que têm uma grandeza e amplitude maior sobre o modelo como um todo.

Uma técnica conhecida de normalização de um atributo qualquer é a subtração pela média e subsequente divisão pelo desvio padrão (ou variância) do conjunto de valores - chamada de *valor z* [3] ou *padronização* [23]. Outra técnica usada, chamada *escalonamento min-max* consiste em comprimir a escala dos valores, mantendo as proporções, de forma que o valor mínimo observado sempre seja igual a 0 e o valor máximo seja sempre igual a 1 [3] [23].

2.4 Inteligência Artificial

A Inteligência Artificial (IA) é a área da Ciência da Computação que se propõe, não apenas a compreender a **inteligência** do ser humano, mas a construir entidades que consigam emular o processo humano do pensamento e, junto dele, toda sua capacidade de perceber, compreender, prever e manipular o mundo ao seu redor [24].

Nota-se que, ao longo da história da IA, existe uma divergência entre acadêmicos a respeito da definição dessa área do conhecimento: se deve ser o estudo dos sistemas que pensam (ou agem) *como seres humanos*; ou que pensam (ou agem) *racionalmente*. A diferença é sutil - pensar racionalmente (isto é, de forma ideal, ótima ou perfeita) pode por vezes ser antônima ao comportamento humano, que é imperfeito. Uma abor-

dagem centrada nos seres humanos deve ser em parte uma ciência empírica e cognitiva, com aspectos biológicos e neuropsicológicos, hipóteses e confirmação experimental; uma abordagem racionalista, por sua vez, envolve uma combinação de lógica, matemática e engenharia [24].

Um dos trabalhos pioneiros da IA, elaborado entre os anos 1940 e 1950, foi o *Teste de Turing*, construído por Alan Turing, no qual ele formula uma definição objetiva para *inteligência*. O teste consiste em submeter um programa computacional a uma conversa por texto com uma pessoa - o interrogador - que lhe faz perguntas. Tal programa passa no teste e, portanto, é classificado inteligente, se o interrogador não conseguir distinguir se teve uma conversa com uma pessoa ou com um computador [24].

Para passar no Teste de Turing, um programa (isto é, máquina), precisaria possuir algumas capacidades. É possível traçar uma correspondência entre essas capacidades e as principais sub-áreas de conhecimento (ou disciplinas) dentro da Inteligência Artificial atualmente [24]:

Processamento de Linguagem Natural: o programa precisa se comunicar com sucesso em um idioma natural, logo precisa saber tanto perceber quanto emitir mensagens;

Representação de conhecimento: para possibilitar o armazenamento das informações percebidas;

Raciocínio automatizado: para usar informações com a finalidade de responder a perguntas e tirar novas conclusões; e

Aprendizado de Máquina: que dá ao programa a habilidade de se adaptar a novas circunstâncias, assim como detectar e extrapolar padrões.

Trabalhos posteriores ao de Turing evoluíram o teste para considerar ainda a dimensão física de uma máquina que se passa por humano [24]. Além de saber entender e se comunicar por texto, no chamado *Teste de Turing Total* o programa precisa possuir capacidades extras de:

Visão Computacional: que confere à maquina a capacidade de perceber objetos físicos; e da

Robótica: que a possibilita se movimentar e manipular objetos conforme solicitado.

Tendo em mente o escopo da aplicação da IA no presente projeto, será trabalhado mais a fundo particularmente o conceito do Aprendizado de Máquina (AM), também conhecido em inglês como *Machine Learning (ML)*, na Seção 2.5.

2.5 Apredizado de Máquina

De acordo com Russell e Norvig [24], diz-se que um agente (de qualquer natureza) **aprende** quando se torna capaz de melhorar seu desempenho em tarefas de predição de padrões e comportamentos após fazer observações sobre o mundo, construindo um conhecimento interno através do processamento de suas próprias percepções anteriores.

No contexto da computação e da inteligência artificial, os agentes inteligentes podem ser empregados em problemas complexos de predição de comportamentos (por exemplo, tarefas de mineração de dados como descritas na Seção 2.3.3, graças aos seus poderes de processamento, armazenando conhecimento para utilizá-los em suas diferentes aplicações.

Em um problema de mineração de dados, o mapeamento explícito entre todas as entradas e todas as saídas possíveis é inviável ou mesmo impossível, para muitas aplicações na vida real. Isso pode ocorrer por vários motivos: os conjuntos de entradas e/ou saídas podem ser infinitos ou suficientemente grandes a ponto de inviabilizar uma representação direta; mesmo que seja possível representar todo o domínio e contradomínio do problema, frequentemente o projetista não conhece a função que o resolve, pois determiná-la foge da capacidade humana em tempo viável; além disso, projetar um programa para um conjunto exaustivo de cenários removeria sua capacidade de aprendizado e adaptação [24].

Géron [23], em raciocínio parecido, sumariza problemas e aplicações ideais para o uso de Aprendizado de Máquina: resolver problemas onde as soluções existentes exigem muita configuração manual, listas extensas de regras de inferência e códigos longos; problemas complexos para os quais não existe uma boa solução conhecida com a abordagem tradicional; problemas em ambientes flutuantes que constantemente tem os dados alterados; e problemas com um enorme volume de dados. Muitas vezes, os problemas apresentam várias dessas características ao mesmo tempo.

Esses algoritmos devem ser capazes de inferir conhecimento sobre o conjunto de entradas para que aprendam a resolver o problema da forma esperada independentemente da entrada que lhe for dada ao longo do tempo, delegando-lhes a tarefa de aprender a resolver problemas complexos, ao invés de descrever explicitamente as regras da solução [24] [23].

Posto isso, Géron [23] retoma duas definições amplamente utilizadas para o Aprendizado de Máquina (AM):

> [Aprendizado de Máquina é o] campo de estudo que dá aos computadores a habilidade de aprender sem ser explicitamente programado. [25]

> Diz-se que um programa de computador aprende pela experiência E em relação a algum tipo de tarefa T e alguma medida de desempenho P se o seu desempenho em T, conforme medido por P, melhora com a experiência E. [26]

Uma das primeiras soluções mundialmente aplicadas de Aprendizado de Máquina foram os filtros de *spam* [23], capazes de ler o conteúdo de um e-mail dentro da caixa de entrada de um usuário e determinar se ele deve ser marcado ou não como *spam*. Nesse exemplo simples, aplicando a definição de Mitchell [26], a tarefa **T** seria atribuir ou não o rótulo de *spam*; a experiência **E** seria o dado de treinamento (o conjunto de textos de e-mail que ele armazenou anteriormente, ou que foi entregue a ele para processar); e a medida de desempenho **P** poderia ser, por exemplo, a proporção entre e-mails marcados corretamente (dentro da categoria *spam* ou fora dela) e o total de e-mails processados. Essa medida específica é chamada de *acurácia* [23] e será aprofundada, junto com outras métricas, na Seção 2.5.5.

Os algoritmos de Aprendizado de Máquina podem ser classificados de acordo com o tipo de supervisão que recebem durante o treinamento. Existem quatro categorias principais de aprendizado: supervisionado, não supervisionado, semi-supervisionado e por reforço [23]. Dependendo de sua categorização, os algoritmos podem ser considerados mais adequados a tarefas específicas de mineração de dados (Seção 2.3.3).

2.5.1 Aprendizado Supervisionado e Não Supervisionado

Sistemas de Aprendizado de Máquina precisam de conjuntos de dados de treinamento para adquirir experiência - é a partir deles que o sistema adquire conhecimento para fazer predições com novos dados. A diferença entre as duas abordagens consiste nas informações presentes nesse conjunto de treinamento.

Aprendizado supervisionado

No aprendizado supervisionado, esses dados de treinamento fornecidos incluem as soluções desejadas - ou seja, cada tupla (registro) tem seu rótulo (classe) conhecido(a) [23].

Retomando a Seção 2.3.3, é sabido que as tarefas de classificação e de regressão são típicos problemas de aprendizado supervisionado. O filtro de *spam* é um bom exemplo disso [23]: ele é treinado com muitos exemplos de e-mails onde a informação de que se trata de *spam* ou não já é sabida; a partir desse conhecimento obtido, ele deve ser capaz de classificar corretamente novos e-mails. Nessa mesma Seção, o problema da concessão de crédito pessoal por uma instituição bancária é um problema de regressão onde o algoritmo precisa ser treinado com rótulos de clientes anteriores (no caso, o valor do crédito concedido), para então ser capaz de atribuir um valor de crédito a clientes futuros.

Alguns dos algoritmos mais importantes do *estado da arte* da literatura de aprendizado supervisionado são citados por Castro e Ferrari [12], categorizados por sua estrutura:

- os baseados em conhecimento (regras lógicas);
- os baseados em distância, como o *k-Nearest-Neighbors (KNN)*;
- os conexionistas, tendo como grande exemplo a Rede Neural Artificial (RNA) e suas diversas variações e evoluções;
- os baseados em funções (parametrizadas);
- os probabilísticos, como o *Naive Bayes*; e
- os baseados em árvores, como as Árvores de Decisão e as Florestas Aleatórias (*Random Forests*).

Essa categorização se aplica tanto aos problemas de classificação quanto aos de regressão. Existem problemas de regressão implementados com algoritmos conexionistas, com algoritmos estatísticos paramétricos, com modelos baseados em árvores, entre outros. A modelagem do problema deste trabalho é feita por uma Floresta Aleatória. Sendo assim, detalharemos os dois algoritmos baseados em árvores supracitados nas seções 2.5.2 e 2.5.3, respectivamente.

Aprendizado não-supervisionado

No aprendizado não-supervisionado, os dados de treinamento não possuem rótulo (classe). Logo, o algoritmo não deve prever rótulos a partir de rótulos conhecidos e, em vez disso, tenta obter conhecimento sobre a base de dados de forma autônoma - geralmente, descobrindo semelhanças e correlações dentro desses dados.

Segundo Géron [23], as tarefas de mineração mais comumente realizadas com esses algoritmos são as de agrupamento ou *clustering* (com os algoritmos *k-Means* e *Clustering Hierárquico*, por exemplo), mineração de regras de associação (com algoritmos como *Apriori* e *Eclat*) e a de redução de dimensionalidade, tendo como principal expoente o algoritmo Análise de Componentes Principais (PCA), descrito mais detalhadamente na Seção 2.5.4.

2.5.2 Árvores de Decisão

As árvores de decisão são algoritmos versáteis de Aprendizado de Máquina supervisionado [23] [3] capazes de executar tarefas de classificação, regressão ou mesmo tarefas *multioutput*[7]. Uma árvore de decisão é uma estrutura de fluxograma em formato de árvore, que

[7]Em alguns casos, pode ser necessário que o classificador atribua várias classes para uma mesma instância. Essas tarefas denominam-se tarefas *multilabel*, caso essas classes sejam binárias, ou *multioutput*, no caso generalizado [23].

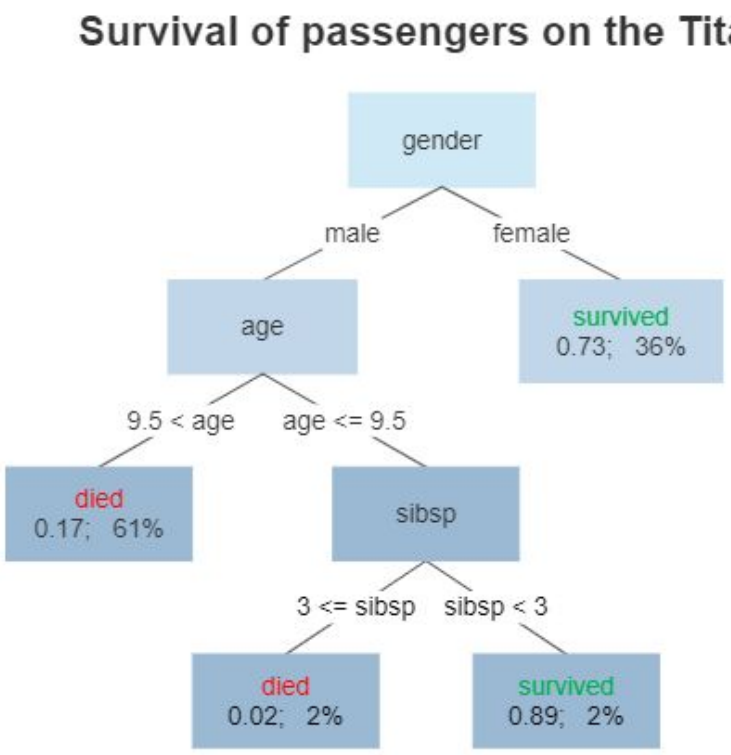

Figura 2.2: Exemplo de uma árvore de decisão aplicada na base de dados do *Titanic*, com a tarefa de determinar se uma pessoa passageira sobreviveu a partir dos seus dados pessoais (em inglês) (Fonte: [27] [28]).

possui nós e ramificações, representando o fluxo da classificação. Cada nó interno (que não é folha, ou seja, que possui ramificações saindo do mesmo) denota um teste condicional sobre um conjunto de atributos; cada ramificação representa uma saída (ou caminho) possível a partir da resposta dos testes; e cada nó-folha (ou terminal) representa a atribuição de um rótulo.

Para cada instância de uma base de dados, os valores devem ser testados na árvore a partir do nó-raiz (na profundidade 0, marcando o início do algoritmo). Então, tal instância deve ser confrontada com a árvore de decisão, passando sucessivamente por todos os nós (testes condicionais) até que chegue em um nó-folha de maior profundidade e receba um rótulo. Cada nó pode testar apenas um atributo, como no caso da Figura 2.2, ou vários deles [3]. É demonstrado que instâncias com valores suficientemente similares seguirão o mesmo caminho pela árvore, sendo classificadas com o mesmo rótulo, e vice-versa.

Existem cenários onde, em uma mesma base de treinamento, tuplas que possuem o mesmo rótulo podem seguir caminhos diferentes na árvore. A árvore é construída através

do exemplo dos dados da base, por isso tal base deve ser rica e variada, abrangendo todos os cenários. Se dentro desse banco de dados houverem "buracos", a construção da árvore e predições serão menos eficientes [21]. Pode, ainda, haver ruído na base que impacta o formato da árvore. Para resolver isso, podem ser realizadas as técnicas de "poda" das árvores (*pruning*) [3].

Com respeito à construção das árvores, destacam-se os algoritmos `ID3`, `C4.5` e *Classification and Regression Trees* (CART). Tais algoritmos adotam uma estratégia gulosa (*greedy*) por divisão e conquista, construindo árvores de forma indutiva baseados na entropia ou *ganho de informação* dos atributos. Esses algoritmos (particularmente o CART), assim como os critérios usados para a criação dos nós das árvores, são explicados em detalhe por Han et al. [3] e Larose e Larose [21]. O CART é o algoritmo utilizado pelo *Scikit-Learn* para treinar suas árvores de decisão [23].

Algumas vantagens dos algoritmos de árvores de decisão [29] incluem: a facilidade de ilustração gráfica, a habilidade de lidar tanto com atributos categóricos quanto numéricos, o custo computacional da execução dos modelos ser logarítmico (em relação ao tamanho do conjunto de treinamento) e a flexibilidade para treinar com dados sem muito pré-processamento - isto é, funciona com valores nulos e não necessita de normalização ou codificação.

Já entre as desvantagens [29], é notável a tendência para a criação de árvores complexas que não são capazes de generalizar bem as predições, isto é, acarretando no fenômeno do *overfitting* (abordado na Seção 2.5.5). Com isso, as árvores tendem a variar muito em função dos dados específicos usados no treinamento.

2.5.3 Florestas Aleatórias

Floresta Aleatória (ou *Random Forest*) é um tipo de método de agregação (*ensemble*) [3]. Os *ensembles* consistem de uma combinação de k modelos de AM (chamados de classificadores-base) $M_1, M_2, ..., M_k$. Cada M_k é treinado com um subconjunto D_k **distinto** da base de dados D. O modelo resultante M' executa a previsão para uma nova tupla com base em "votos" dos k modelos - cada modelo processa e atribui uma classe para a tupla e, após isso, seguindo algum critério — por exemplo, o voto da maioria —, se obtém a classe resultante.

Usar um conjunto de classificadores/estimadores geralmente resulta em modelos mais eficazes do que a utilização de um modelo individual - não por acaso, as florestas aleatórias são alguns dos algoritmos mais poderosos de AM disponíveis atualmente [23]. Métodos *ensemble* performam melhor quando existe uma diversidade significativa nos dados - isto é, com pouca correlação entre os modelos-base [3] [23].

Uma Floresta Aleatória, como o nome pode sugerir, é um conjunto de árvores de decisão [3]. Cada árvore M_i recebe uma base D_i derivada da base de dados original D. Os conjuntos de tuplas D_i são geralmente feitos com o método de *bagging*, que consiste em criar amostras aleatórias com substituição[8] do mesmo tamanho de D. Cada árvore, em cada nó, seleciona aleatoriamente uma quantia bem inferior de atributos em relação aos atributos originais, para determinar as decisões e ramificações. Ás árvores são construídas com o algoritmo CART. O desempenho de uma floresta aleatória depende da eficácia das árvores individuais, em conjunto com a medida de correlação (dependência) entre os mesmos.

Em caso de problemas de classificação, um critério comum para se definir a classe resultante de uma tupla, após ser classificada por todas as árvores individuais, é a votação simples majoritária. Já em problemas de regressão, é comum utilizar uma média simples dos valores retornados por cada árvore [3].

Uma grande qualidade das florestas aleatórias é facilitar a medição da importância relativa de cada atributo. É possível, na ferramenta *Scikit-Learn*, medir a importância de uma característica analisando o quanto os nós de árvores que a utilizam reduzem a impureza, em média [23].

Uma vantagem do uso das florestas aleatórias [30], assim como outros métodos de agregação *ensemble* é a maior capacidade de generalização e robustez, assim reduzindo o *overfitting* acarretado pelo uso de um único classificador. Além disso, elas reduzem a variância do treinamento ao introduzir aleatoriedade pelo uso de amostragem por *bagging*. Assim, o modelo de floresta aleatória ajuda a resolver a maior limitação das árvores de decisão individuais.

2.5.4 Redução de Dimensionalidade

Problemas de Aprendizado de Máquina podem envolver milhares (como no caso da base de dados deste projeto) ou até milhões de características para todo dado usado no treinamento. À primeira vista, parece um ponto positivo por agregar mais informações à base de dados. No entanto, isso torna o treinamento progressivamente mais lento e, além disso, piora as soluções encontradas. Isso é um problema conhecido na bibliografia como a **maldição da dimensionalidade** [23] [31]. Para contornar o problema, é possível transformar um problema insolúvel (em tempo viável) em um problema tratável por algoritmos de AM ao reduzir consideravelmente o número de características abordadas [23].

[8] A técnica de *bagging* (*bootstrap aggregation*) constrói amostras com substituição, o que significa que uma tupla, após incluída na amostra, tem a mesma chance de ser selecionada novamente. Isso pode ocasionar a duplicação de tuplas nas amostras, e a omissão de outras. Tais amostras são chamadas de amostras *bootstrap* [3].

O aumento do número de objetos e, principalmente, de dimensões, podem fazer com que os dados fiquem esparsos e as medidas matemáticas e estatísticas usadas na análise se tornem instáveis. Além disso, como esperado, uma base muito grande em tuplas e atributos pode tornar os algoritmos de aprendizado, assim como os modelos gerados pelos mesmos, muito complexos [12].

Em contextos como o de *Big Data*, entre outros, onde o volume de dados gerado é grande, dinâmico e não-estruturado, a resposta para contornar a base e reduzir a complexidade dos algoritmos pode estar na redução de dimensionalidade dos dados [17].

Castro e Ferrari [12] destacam alguns métodos de redução de dimensionalidade de dados, dos quais o primeiro é particularmente relevante nesse projeto:

Seleção de Atributos (ou *Feature Selection*): técnica que remove atributos pouco relevantes ou redundantes.

Compressão de atributos: emprega técnicas de codificação (ou transformação) de dados, resultando em uma nova base com atributos mais relevantes, em vez de seleção.

Redução de dados: os dados (tanto tuplas quanto atributos) podem ser removidos, estimados ou substituídos por representações menores, como modelos paramétricos (que armazenam apenas parâmetros em vez dos dados em si) e os não-paramétricos, como o agrupamento, a amostragem e o uso de histogramas.

Discretização: os valores de atributos são substituídos por intervalos ou níveis conceituais de abstração mais elevados, reduzindo a quantidade final e o domínio dos atributos.

Ao reduzir a dimensionalidade, perdemos informação (por exemplo, em uma compressão de arquivo de imagem). Embora acelere o treinamento, também pode fazer com que o modelo, o aprendizado e as predições funcionem de forma pior [23]. É importante que os métodos de redução de dimensionalidade de dados preservem a integridade e características dos dados originais, de forma a manter a eficácia da tarefa de mineração, produzindo modelos e resultados igualmente confiáveis com menos esforço computacional [12].

Seleção de Atributos

A Seleção de Atributos ou *Feature Selection* é uma técnica que reduz o tamanho do conjunto de dados ao remover atributos redundantes ou irrelevantes [3]. Seu objetivo é encontrar um conjunto mínimo de atributos tal que a distribuição de probabilidade dos rótulos seja a mais próxima possível da distribuição de probabilidade original com todos os atributos. A seleção de atributos desempenha papel importante na construção de modelos

mais eficientes, pois são processados menos dados, porém com o objetivo de preservar a tendência dos dados originais [3] [32].

Para n atributos, existem 2^n subconjuntos de atributos possíveis, o que tornaria inviável uma busca exaustiva pelo melhor subconjunto quando n é muito grande. Assim, a abordagem mais utilizada envolve métodos heurísticos que trabalham em uma parte restrita dos dados: de forma *forward*, onde o conjunto de atributos inicia vazio e iterativamente o "melhor" atributo fora do conjunto é incluído no mesmo; de forma *backward*, onde os "piores" atributos são removidos sucessivamente do conjunto original de atributos; ou uma junção de ambos os métodos. O critério de parada desses algoritmos pode se basear no tamanho do subconjunto (quando o usuário quiser selecionar um número k de atributos) ou quando a métrica usada atingir um certo limiar [3].

Mais importante que o funcionamento do algoritmo de seleção, entretanto, é saber a definição de "melhor" e de "pior". Segundo Han et al. [3] essas métricas envolvem o uso de testes estatísticos (como de variância e correlação) ou cálculos de ganho de informação (entropia).

As métricas providas pela biblioteca *Scikit-Learn*, assim como a decisão estratégica de qual número k de atributos selecionar na etapa de *feature selection* desse projeto, são abordadas na Seção 4.7.

Análise de Componentes Principais

A Análise de Componentes Principais (PCA) é um dos métodos mais populares de compressão de atributos encontrados no estado-da-arte do Aprendizado de Máquina. É um procedimento estatístico que converte um conjunto de objetos com presença de alguma correlação em um outro conjunto de objetos com variáveis linearmente descorrelacionados, os *componentes principais* [12].

O número de componentes principais é sempre menor ou igual ao número de atributos originais; a transformação é feita de forma que os componentes resultantes tenham a maior variância possível, realizando um mapeamento linear dos dados (chamado de projeção) em um espaço de dimensão menor [12]. Se os dados originais puderem ser reconstruídos com a aplicação inversa do algoritmo, sem perda de informação, houve uma compressão "sem perda" *(lossless)*; caso contrário, a compressão foi "com perda" *(lossy)* [3].

Essa projeção, que essencialmente transforma a base em uma nova base com novos atributos, difere da técnica de seleção de atributos, à medida em que não preserva os valores originais. Em um experimento, Traskas [33] se propõe a ilustrar a diferença entre os algoritmos PCA e *feature selection* na performance de modelos de predição - aplicando ambos na etapa de pré-processamento dos dados. O estudo se baseia nas respectivas implementações dos algoritmos na biblioteca *Scikit-Learn*.

2.5.5 Avaliação de Desempenho

A avaliação do desempenho de um algoritmo de Aprendizado de Máquina corresponde à aferição da qualidade da modelagem. Determina se o modelo aproxima a solução ideal ou, em outras palavras, se desenvolveu aprendizado. É a última etapa do processo de construção e aplicação de um modelo preditivo , pertencendo à etapa de Avaliação dentro do processo CRISP-DM (vide a Seção 2.3.2). As medidas de desempenho se propõem a responder o quão bem o modelo generalizará para dados fora da base de treinamento. No caso do aprendizado supervisionado, elas são baseadas em cálculos de acerto e erro entre a saída fornecida pelo modelo e a saída desejada [12].

Desempenho de classificadores: a Matriz de Confusão

O desempenho de um algoritmo de classificação depende de sua flexibilidade (*bias*) e da qualidade do treinamento (variância). A forma mais comum de avaliar é simplesmente calcular o percentual de classificação correta, mais conhecida como **acurácia**, assim como seu complemento, o **erro**. Por padrão, a acurácia não considera o custo de uma predição incorreta - qualquer erro, para qualquer classe, possui o mesmo peso [12]. A Acurácia, apesar de ser uma medida específica, também se refere genericamente à qualidade de predição de um modelo [3].

Os problemas binários (com dois valores possíveis para a classe-alvo) são um caso particular de grande interesse dentro dos problemas de classificação. Vários problemas reais podem ser mapeados em classificação binária, como a concessão de crédito ou o classificador de spam [12]. O alvo recebe o nome de classe positiva para o rótulo `Verdadeiro`, e de classe negativa para o rótulo `Falso`. A partir dessas definições, são definidos alguns conceitos os quais são atribuídos a tuplas, e que constroem a matriz de confusão de um classificador binário [12], ilustrada no Quadro 2.1:

- Verdadeiro Positivo (VP): objeto de classe positiva classificado como `Verdadeiro`;
- Verdadeiro Negativo (VN): objeto de classe negativa classificado como `Falso`;
- Falso Positivo (FP): objeto de classe negativa classificado como `Verdadeiro` - conhecido como “alarme falso” ou Erro do Tipo 1;
- Falso Negativo (FN): objeto de classe positiva classificado como `Falso` - conhecido como Erro do Tipo 2;

A partir desses conceitos, definem-se as principais métricas de avaliação: Acurácia; Erro; Precisão; *Recall* ou Revocação ou Sensitividade; Especificidade; e *F-score* ou Medida F [12] [3].

		Classe predita	
		Positiva	**Negativa**
Classe original	**Positiva**	VP	FN
	Negativa	FP	VN

Quadro 2.1: Matriz de confusão de um classificador binário.

$$Acc = \frac{VP + VN}{VP + FP + VN + FN} \tag{2.1}$$

(Acurácia)

$$E = 1 - Acc = \frac{FP + FN}{VP + FP + VN + FN} \tag{2.2}$$

(Erro)

$$Pr = \frac{VP}{VP + FP} \tag{2.3}$$

(Precisão)

$$Re = \frac{VP}{VP + FN} \tag{2.4}$$

(*Recall* ou Revocação ou Sensitividade)

$$Esp = \frac{VN}{VN + FP} \tag{2.5}$$

(Especificidade)

$$F_1 = \frac{2 * Pr * Re}{Pr + Re} \tag{2.6}$$

(*F-score* ou Medida F)

Essas métricas têm propósitos distintos e complementares [12] [3]:

- **Acurácia:** mede a proporção de acertos em relação ao total de predições, sendo útil para problemas com classes balanceadas;
- **Erro:** representa a taxa de classificações incorretas, complementar da acurácia;
- **Precisão:** indica quantas das predições positivas feitas pelo modelo são realmente corretas. É importante quando o custo de falsos positivos é elevado;
- ***Recall* (ou Revocação, ou Sensibilidade):** mede a capacidade do modelo de identificar corretamente os casos positivos reais, sendo relevante quando é crítico reduzir os falsos negativos;

- **Especificidade:** avalia a capacidade de identificar corretamente os casos negativos;
- **F-Score:** combina precisão e *recall* em uma única métrica, sendo ideal para bases desbalanceadas.

Validação cruzada

Na aprendizagem supervisionada, os modelos precisam atingir um equilíbrio (conhecido como o *Dilema bias-variância*) [12], de forma a serem flexíveis para se aproximarem ao máximo da solução com os dados de treinamento disponíveis, de forma a evitar o *underfitting*[9] e o *overfitting*[10].

A validação cruzada é o método sistemático mais comum usado para atingir esse equilíbrio. Ela consiste em executar o modelo sobre um subconjunto de dados de teste não usados no treinamento. Com o passar das iterações, se o erro nas predições do conjunto de teste começar a aumentar de forma consistente, é o momento mais indicado para parar o treinamento. Outra finalidade da validação cruzada é a de aferir o desempenho dos algoritmos [12].

Um algoritmo comum é a *validação cruzada em k-pastas*, que consiste em dividir a base de dados em k subconjuntos, de forma que, em k iterações, o subconjunto i ($i \in 1...k$) será usado para teste, enquanto os outros $k-1$ são usados para treinamento. O método da separação de pastas influencia o resultado final, por isso é comum que o algoritmo acima seja executado k vezes, cada vez com uma estratégia diferente de separação [12].

Desempenho de modelos de regressão

Tarefas de classificação podem ser vistas como casos particulares de regressão, nas quais a saída é discreta (ou categórica). Assim, as métricas vistas e o conceito de validação cruzada são válidos em ambos os tipos de modelo. A saída de um estimador é um valor numérico contínuo o qual se deseja ser o mais próximo possível do valor observado. A diferença entre os dois valores fornece medidas de erro de estimação do algoritmo [12].

Para cada tupla (registro) j em uma base de dados com n tuplas, seja d_j o valor real da variável desejada, y_j o valor estimado pelo modelo, e $e_j = d_j - y_j$ o erro observado. A partir disso, definem-se várias métricas de desempenho de natureza contínua [12] [21]. De maior relevância ao projeto, temos as métricas de: Soma dos erros quadráticos; Erro quadrático médio; Erro absoluto médio; Raiz do erro quadrático médio ou erro padrão da estimativa; e Coeficiente de determinação.

[9] Diz respeito à incapacidade e inflexibilidade do modelo de se adaptar aos dados de treinamento. É também conhecido como erro de representação [12].

[10] Ocorre quando o modelo é treinado em excesso e absorve ruídos da base de treinamento, tornando-se ineficaz para a predição de novos dados. Chamado também de erro de generalização [12].

$$SSE = \sum_{j=1}^{n} (e_j)^2 \tag{2.7}$$

(Soma dos erros quadráticos)

$$MSE = \frac{1}{n} * \sum_{j=1}^{n} (e_j)^2 \tag{2.8}$$

(Erro quadrático médio)

$$MAE = \frac{1}{n} * \sum_{j=1}^{n} |e_j| \tag{2.9}$$

(Erro absoluto médio)

$$s = \sqrt{\frac{1}{n} * \sum_{j=1}^{n} (e_j)^2} \tag{2.10}$$

(Raiz do erro quadrático médio ou erro padrão da estimativa)

$$R^2 = 1 - \frac{\sum_{j=1}^{n} (e_j)^2}{\sum_{j=1}^{n} (d_j - \overline{d})^2} \tag{2.11}$$

(Coeficiente de determinação)

Essas métricas têm diferentes objetivos na avaliação de modelos de regressão [12] [21]:

- **SSE (Soma dos Erros Quadráticos)**: mede o erro total acumulado ao quadrado. Quanto menor, melhor o ajuste geral do modelo.
- **MSE (Erro Quadrático Médio)**: representa a média dos erros quadráticos, servindo para avaliar o desempenho médio do modelo em relação ao conjunto de dados.
- **MAE (Erro Absoluto Médio)**: calcula a média dos erros absolutos, sendo menos sensível a outliers que o MSE.
- **RMSE (Raiz do Erro Quadrático Médio)**: expressa o erro médio em mesma unidade da variável-alvo, destacando maiores desvios.
- **R^2 (Coeficiente de Determinação)**: mede a proporção da variabilidade da variável-alvo explicada pelo modelo. Varia de 0 a 1, sendo 1 um ajuste perfeito.

Ao fim deste Capítulo, foram apresentados os conceitos básicos que são explorados neste trabalho a partir da bibliografia. O próximo Capítulo, mesmo sendo também parte da revisão bibliográfica, aborda a metodologia sistêmica de pesquisa que possibilitou a descoberta e revisão de artigos relacionados ao tema e, além disso, fornece um resumo das referências pesquisadas.

Capítulo 3

Trabalhos relacionados

Nos últimos tempos, um grande número de pesquisas foi desenvolvido com o intuito de aplicar a mineração de dados nos mais variados universos de dados. Neste Capítulo, são listadas as referências acadêmicas que foram analisadas e revisadas para refinar e embasar tema, escopo, hipótese e objetivo deste projeto.

Dentre as fontes bibliográficas primárias procuradas, foram usados trabalhos de graduação do Departamento de Ciência da Computação (CIC) da UnB, assim como artigos obtidos no acervo de periódicos da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) e no portal *IEEE Xplore*, mantido pelo Instituto de Engenheiros Eletricistas e Eletrônicos (IEEE). Foi explorada também a ferramenta da Sociedade Brasileira de Computação (SBC) [34], mas não foram encontrados na data da pesquisa nenhum artigo com os termos (*"World Bank"* e *"data mining"*) ou (*"World Bank"* e *"Python"*) que faça uma análise semelhante à que foi feita aqui.

Todas os trabalhos obtidos pelas fontes citadas compartilham o objeto - bases de dados do Banco Mundial - em comum com este projeto.

Com respeito aos artigos publicados no portal da *IEEE*, há o artigo de Arif e Ali [35], que propuseram o emprego da mineração de dados sobre estatísticas do Banco Mundial utilizando o método da matriz de correlação para a tarefa de encontrar correlações e padrões, utilizando dados num intervalo de seis anos entre 2000 e 2005. É relevante também citar o trabalho de Gamberger et al. [36], que se debruçam a aplicar mineração de dados, tendo escopo voltado especificamente a dados sobre crises financeiras enfrentadas pelo mundo entre 1976 e 2011, para construir agrupamentos (*clusterização*) de países. Com isso, se propõem a obter conhecimento sobre as origens e características distintas de tais crises. Por último, o trabalho de Ahammad et al. [37] aplica diversos algoritmos de Aprendizado de Máquina (AM) como o KNN, Florestas Aleatórias, Árvores de Decisão etc. para a tarefa de predição do indicador do PIB em Bangladesh, a partir de indicadores do Banco Mundial - utilizando-se de uma métrica de desempenho em todos os modelos

de aprendizado empregados para comparar a qualidade das predições.

Henrique et al. [38] utilizam dados do Banco Mundial, mas também dados de outras fontes como a Organização das Nações Unidas e a Organização Mundial da Saúde, para a mineração de dados. Tal projeto tinha o objetivo de encontrar padrões sobre o impacto da quantidade de casos e mortes ocasionados pela pandemia de COVID-19, entre 2020 e 2022, nos indicadores de qualidade de vida como Produto Interno Bruto (PIB), Índice de Desenvolvimento Humano (IDH), PIB *per capita* e a proporção de pessoas vivendo em situação de extrema miséria no mundo. A metodologia da mineração consistia em aplicar algoritmos de regressão múltipla *stepwise*, entre outros algoritmos de correlação. Utilizou-se a linguagem de programação *R*, comumente utilizada em análises de dados [32].

O trabalho de Santana e Carvalho [20] se debruça a estudar o Banco Mundial e a base de dados Indicadores de Desenvolvimento Mundial (WDI), servindo de inspiração para este projeto. No entanto, este projeto tem objetivos diferentes. A tarefa desse trabalho se propôs a obter conhecimento e desenhar padrões sobre a base através do agrupamento de países em *clusters* de acordo com a similaridade dos seus indicadores, tarefa que se encaixa como aprendizado não-supervisionado, ao passo que também empregaram tarefas de regressão usando árvores de decisão para predizer diversos indicadores, como o crescimento do Produto Interno Bruto (PIB), inflação, PIB *per capita*, entre outros. Um ponto notável no estudo supracitado é a estratégia de filtragem dos melhores indicadores. Foi realizada, após uma análise exploratória manual, uma filtragem de indicadores com base na quantidade de valores ausentes, reduzindo-os a 125, assim como uma filtragem significativa no número de países, reduzindo bastante a quantidade de registros na base. Esse filtro de países foi implementado de forma manual e subjetiva, mantendo os de mais "importância" continental e que possuíam mais valores presentes. Essa filtragem rígida de dados tanto de países quanto de indicadores, somada ao pequeno intervalo temporal abrangido (os 10 anos mais recentes à época), reduziam consideravelmente a base de dados. Foram utilizadas por eles ferramentas como *Weka*, *Pentaho* e *Orange*, assim como o *MySQL* para a persistência de dados, sem fazer uso da linguagem *Python*.

Em contraste, este projeto se propõe a melhorar a metodologia de Santana e Carvalho [20] aplicando uma filtragem algorítmica de indicadores (parte importante do objetivo) e removendo apenas os países com mais dados ausentes. Com essa estratégia, a quantidade superlativa de indicadores não é um impeditivo à eficiência da modelagem ou à análise dos autores. Também é proposto um passo adicional de inferência dos valores ausentes que restarem, também usando modelos preditivos - detalhado na Seção 4.5.2.

A partir da revisão bibliográfica e dos resultados observados nos estudos mencionados, foi possível verificar a possibilidade de fazer um trabalho similar, detalhado no próximo

Capítulo.

Capítulo 4

Mineração de dados de Indicadores de Desenvolvimento Mundial

Este Capítulo descreve o processo de coleta e preparação dos dados utilizados na análise. Inicialmente, os dados contendo países e seus indicadores são coletados de forma manual a partir da aplicação *web* do Banco Mundial. Em seguida, são aplicadas técnicas de mineração de dados para analisar esses indicadores e identificar padrões relacionados ao desenvolvimento mundial.

O fluxograma do projeto, ilustrando as transformações aplicadas nos dados e a relação entre os dois *scripts*, se encontra na Figura 4.1. Ela descreve a relação lógica entre os *scripts* e a sequência temporal na qual são executados os mesmos. Os três *scripts* são executados separadamente e se comunicam por meio da escrita e leitura nos arquivos de Valores Separados por Vírgula (CSV) gerados no processo: `WDITransformada.csv` e `WDIPreProcessada.csv`. Cada artefato gerado como resultado de uma etapa será utilizado pela próxima etapa.

O projeto, que consiste desses *scripts* e, também, dos arquivos de dados brutos extraídos da *web*, assim como ilustrações, gráficos e tabelas gerados, está hospedado em um repositório público através da ferramenta de controle de versionamento *Git*. Com isso, foi possível a colaboração mútua dos autores para a elaboração dos *scripts*. Também por esse motivo, o projeto pode ser copiado e executado localmente em qualquer sistema operacional que tenha instaladas as ferramentas da Seção 4.1.

Ao longo das seções, são mostrados alguns trechos do *script* da modelagem, mostrados nos Códigos 4.1, 4.2, 4.3 e 4.4. Todos eles fazem parte do arquivo mostrado integralmente nos Apêndices B e C.

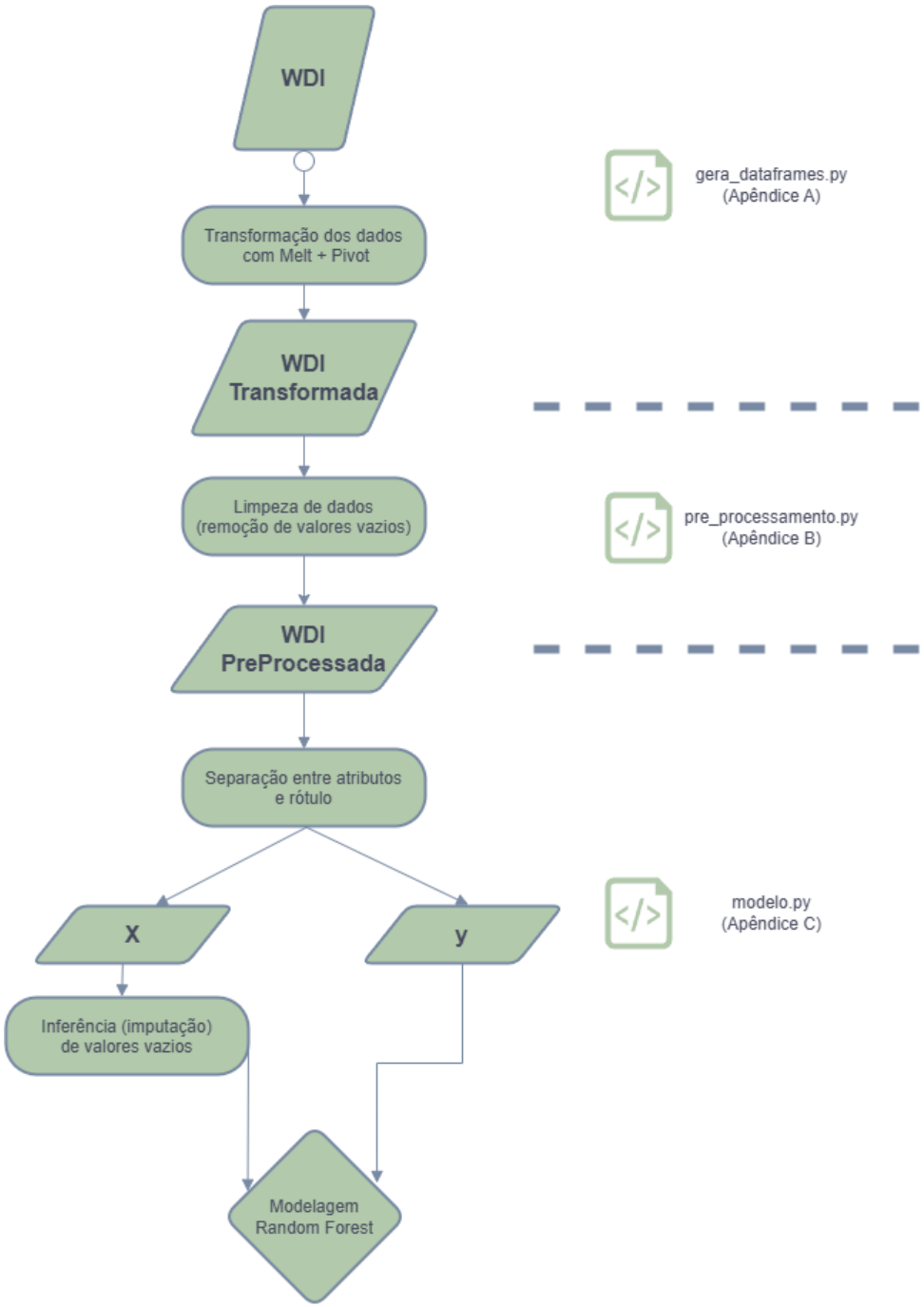

Figura 4.1: Fluxograma de todas as transformações aplicadas nos dados brutos até a construção do modelo.

4.1 Ferramentas

As ferramentas utilizadas neste trabalho foram escolhidas com base na sua relevância e eficácia para tarefas de mineração de dados e aprendizado de máquina. Além de oferecerem ampla documentação e comunidade ativa, essas ferramentas garantem compatibilidade entre si e possibilitam a replicação dos experimentos em diferentes ambientes operacionais. A Figura 4.2 apresenta uma visão geral das principais ferramentas utilizadas ao longo das etapas de desenvolvimento, desde a coleta até a modelagem e análise dos dados.

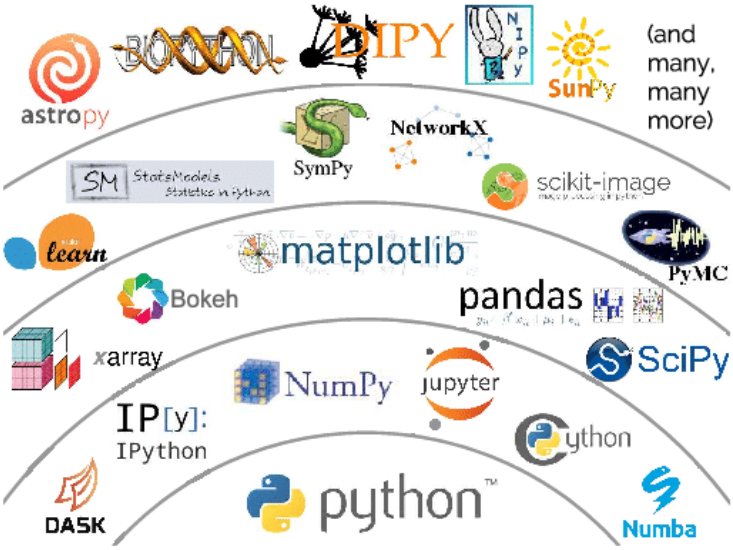

Figura 4.2: Ilustração esquemática do ecossistema de software científico da linguagem *Python*, de onde se obtém algumas das ferramentas utilizadas neste trabalho (Fonte: [39]).

Git

O *Git* [40] é um sistema de controle de versão amplamente utilizado para rastrear alterações em arquivos e facilitar o desenvolvimento colaborativo de projetos. Ele permite a organização eficiente do código, a documentação das mudanças feitas ao longo do tempo e a colaboração entre membros da equipe e orientadores. Além disso, possibilita a reprodutibilidade dos resultados e oferece um meio para a revisão por pares.

Python

Python [41] é uma linguagem de programação amplamente utilizada em ciência de dados e mineração de dados. Sua sintaxe simples e legibilidade tornam-na uma escolha popular

para análise e modelagem de dados. Além disso, a vasta comunidade de desenvolvedores e a disponibilidade de bibliotecas especializadas facilitam a implementação de algoritmos de aprendizado de máquina e a manipulação de dados . A versão usada no projeto é a **3.11.5**.

Spyder

O *Spyder* [42] é um Ambiente de Desenvolvimento Integrado (IDE, na sigla em inglês) específico para ciência de dados e análise numérica em *Python*. Ele oferece recursos como edição de código, depuração, visualização de variáveis e integração com bibliotecas populares, incluindo *Pandas* e *Scikit-Learn*. O *Scikit-Learn* é uma escolha conveniente para desenvolvedores que desejam trabalhar com eficiência em projetos de mineração de dados. A versão usada no projeto é a **5.4.3** e esta instalação foi responsável pela instalação e gerenciamento das versões das outras ferramentas.

Pandas

O *Pandas* [43] é uma biblioteca *Python* para análise de dados que oferece estruturas de dados flexíveis, como *dataframes* e séries. Ele permite a leitura, manipulação e limpeza eficiente de dados tabulares. Com o Pandas, é possível realizar operações como filtragem, agregação, transformação e visualização de dados de maneira eficaz. A versão usada no projeto é a **2.0.3**.

Scikit-Learn

O *Scikit-Learn* (também abreviada como *sklearn*) [44] é uma biblioteca de Aprendizado de Máquina (AM) em *Python*. Ele fornece uma ampla variedade de algoritmos de classificação, regressão, agrupamento e pré-processamento de dados. O *Scikit-Learn* é amplamente utilizado para construir, treinar e avaliar modelos de aprendizado de máquina, tornando-o essencial para projetos de mineração de dados. Essa biblioteca fornece todos os algoritmos de AM utilizados na Seção 4.5.2 em diante. A versão usada no projeto é a **1.3.0**.

O desenvolvimento dos *scripts*, assim como todas as imagens de visualização de *dataframes* e visualização através do console, como a da Figura 4.7, foi obtido através desta ferramenta.

Matplotlib

O *Matplotlib* [45] é uma biblioteca poderosa e amplamente utilizada em *Python* para a criação de visualizações gráficas de dados. Em contextos acadêmicos, sua utilidade reside na capacidade de transformar dados complexos em gráficos claros e interpretáveis,

facilitando a análise e a apresentação dos resultados de pesquisas. O *Matplotlib* suporta uma ampla gama de tipos de gráficos, incluindo linhas, barras, dispersão, histogramas e gráficos de pizza, entre outros, permitindo uma versatilidade significativa na representação visual de dados acadêmicos. A versão usada no projeto é a **3.7.2**.

Todos os gráficos do projeto, como a Figura 4.8 e a Figura 5.8, foram gerados pelo *Matplotlib* e recuperados para o disco local através do console do *Spyder*.

Hardware

Para a execução e coleta dos resultados, o projeto utilizou todas as ferramentas acima mencionadas, sendo instaladas localmente em um computador pessoal de sistema operacional *Microsoft Windows 11 Pro* (versão *24H2*), cujas especificações de *hardware* incluíam: Processador *AMD* modelo *Ryzen 5 5500* com 6 núcleos e 12 *threads*; Memória RAM de 32 *gigabytes*; Memória secundária em *Solid-State Drive* de 1 *terabyte*; Placa de vídeo *MSI* modelo *Radeon RX 6750 XT* com 12 *gigabytes* de memória de vídeo.

4.2 A Base de Dados do Banco Mundial

Na base de dados dos Indicadores de Desenvolvimento Mundial (WDI) [8] estão contidos os dados de indicadores de todos os países cadastrados no mundo, assim como dados agregados por regiões geoeconômicas.

Para cada país ou região, dispostos como linhas, constam todos os indicadores socioeconômicos, também dispostos em linhas. Já as colunas representam os anos de aferição de dados, ou seja, a progressão anual de cada um desses indicadores, desde 1960 até 2023 (período de 64 anos). Em outras palavras, cada objeto nessa base de dados é composto por: país/região e sua abreviação; nome e código do indicador; valor do indicador para esse país/região em 1960; valor do indicador para esse país/região em 1961; e assim sucessivamente.

A base de dados WDI, portanto, é composta de **396.872 linhas** (produto dos 266 países/regiões pelos 1492 indicadores) e **68 colunas** (sendo 4 para a descrição dos países/regiões e indicadores; e mais 64 correspondentes aos valores anuais).

Todos os nomes de países, regiões e indicadores, assim como todos os metadados, se encontram em língua inglesa.

4.2.1 Fonte dos Dados

A base de dados dos Indicadores de Desenvolvimento Mundial (WDI) está distribuída em arquivos Valores Separados por Vírgula (CSV), que foram extraídos de um arquivo

zipado, obtido manualmente a partir da página *web* do *DataCatalog* (Catálogo de Dados) [46].

A base em si é apenas um dos 6 arquivos CSV presentes no arquivo zipado - os outros fornecem metadados diversos, incluindo sobre os países/regiões e sobre os indicadores. A base principal possui tamanho aproximado de 190 *megabytes*; já o conjunto dos arquivos tem um tamanho somado de aproximadamente 265 *megabytes*.

Alternativamente, o Banco Mundial oferece mais possibilidades de recuperação dos dados: uma é de forma computadorizada, por uma *Application Programming Interface* (API). Tal opção não foi explorada neste trabalho devido à dificuldade para aprender o funcionamento e contrato da API para obter os dados nos parâmetros desejados - exigindo um esforço significativamente maior que o *download* da base inteira diretamente do *DataCatalog*.

Outra opção, que inicialmente foi utilizada antes do *DataCatalog*, é o *DataBank*, ferramenta de visualização que contém coleções de dados diversas além da WDI. O *DataBank* oferece uma flexibilidade muito maior para, por exemplo, filtrar indicadores, países e períodos temporais e decidir o formato mais adequado para exportar. No entanto, uma limitação que impossibilitou seu uso foi o fato de não ser possível baixar a base WDI inteira sem que a aplicação "quebrasse": teria que ser feito o *download* manual uma vez para cada ano disponível. Inevitavelmente, obter a base pelo *DataBank* resultaria numa fonte de dados menos populada e rica.

4.2.2 Indicadores Socioeconômicos

Os registros da base de dados WDI são caracterizados por 1.492 indicadores socioeconômicos.

Entre os indicadores mais importantes, a base inclui dados sobre o Produto Interno Bruto (PIB), que, como destacado por Nordhaus e Samuelson [11], é amplamente utilizado para medir a atividade produtiva de um país e embasar políticas públicas. Além do PIB, a base também contém informações sobre expectativa de vida, taxa de alfabetização, desigualdade, pobreza, entre outros. Esses dados são coletados de fontes oficiais e permitem análises comparativas e tendências globais.

Todos os 1.492 dentro das bases de dados são referidos por um código - uma abreviatura de até 25 caracteres, assim como por seu nome por extenso, todos em inglês.

Os indicadores estão divididos em alguns grupos temáticos [8], descritos a seguir. Dentre cada categoria, eles podem aparecer replicados com diferentes clivagens, mantendo relativa semelhança ou dependência entre si - por exemplo, um indicador para cada gênero; valor absoluto e em porcentagem anual; medições em diferentes unidades; entre outros.

Pobreza e desigualdade: Pobreza, prosperidade, consumo, distribuição de renda, entre outros;

População: Dinâmicas de população, educação, trabalho, saúde e gênero, entre outros;

Ambiente: Agricultura, mudanças climáticas, energia, biodiversidade, saneamento, entre outros;

Economia: Crescimento econômico, estrutura econômica, renda, poupanças, comércio, entre outros;

Mercados: Negócios, mercados de ações, comunicações, transporte, tecnologia, indústria militar, entre outros;

Relações internacionais: Dívida externa, comércio externo, ajuda humanitária, turismo, migração, entre outros;

A tarefa da modelagem desse projeto é predizer os valores para um indicador específico dentro do tema da economia - o indicador **Crescimento Anual do PIB (em porcentagem anual)** (*GDP growth (annual %)*), de código `NY.GDP.MKTP.KD.ZG`. Essa predição será feita usando todos os indicadores de todos os grupos temáticos como atributos para o treinamento do modelo.

Existem ao todo 86 indicadores socioeconômicos na base que fazem referência ao PIB - geralmente um indicador de um tema diverso expresso em pontos percentuais em razão do PIB. Portanto, pode-se dizer que tais indicadores são fortemente dependentes, ou são determinados pela taxa **Crescimento Anual do PIB (em porcentagem anual)**. Eles são incluídos na categoria de *indicadores triviais* - sozinhos, esses atributos seriam capazes de treinar um modelo gerando previsões com alta acurácia, porém enviesadas. Alguns deles são exibidos na Figura 4.3.

No Apêndice D, há uma lista exaustiva de todos os indicadores da base WDI que obedeceram aos filtros aplicados na Seção 4.5.1, com seu código e seu nome em inglês[1].

4.2.3 Metadados

Junto à base dos Indicadores de Desenvolvimento Mundial (WDI), foram extraídos vários arquivos de metadados, ou seja, bases de dados que fornecem contexto em relação à base principal:

[1]Nomes de indicadores com mais de 50 caracteres foram abreviados no apêndice a fim de prevenir o excesso de páginas.

Series Code	Topic	Indicator Name
NE.TRD.GNFS.ZS	Economic Policy & Debt: National accounts: Shares of GDP & other	Trade (% of GDP)
NV.AGR.TOTL.ZS	Economic Policy & Debt: National accounts: Shares of GDP & other	Agriculture, forestry, and fishing, value added (% of GDP)
NV.IND.MANF.ZS	Economic Policy & Debt: National accounts: Shares of GDP & other	Manufacturing, value added (% of GDP)
NV.IND.TOTL.ZS	Economic Policy & Debt: National accounts: Shares of GDP & other	Industry (including construction), value added (% of GDP)
NV.SRV.TOTL.ZS	Economic Policy & Debt: National accounts: Shares of GDP & other	Services, value added (% of GDP)
NY.GDP.COAL.RT.ZS	Environment: Natural resources contribution to GDP	Coal rents (% of GDP)
NY.GDP.DEFL.KD.ZG	Financial Sector: Exchange rates & prices	Inflation, GDP deflator (annual %)
NY.GDP.DEFL.KD.ZG.AD	Financial Sector: Exchange rates & prices	Inflation, GDP deflator: linked series (annual %)
NY.GDP.DEFL.ZS	Financial Sector: Exchange rates & prices	GDP deflator (base year varies by country)
NY.GDP.DEFL.ZS.AD	Financial Sector: Exchange rates & prices	GDP deflator: linked series (base year varies by country)
NY.GDP.DISC.CN	Economic Policy & Debt: National accounts: Local currency at cur…	Discrepancy in expenditure estimate of GDP (current LCU)
NY.GDP.DISC.KN	Economic Policy & Debt: National accounts: Local currency at con…	Discrepancy in expenditure estimate of GDP (constant LCU)
NY.GDP.FRST.RT.ZS	Environment: Natural resources contribution to GDP	Forest rents (% of GDP)
NY.GDP.MINR.RT.ZS	Environment: Natural resources contribution to GDP	Mineral rents (% of GDP)
NY.GDP.MKTP.CD	Economic Policy & Debt: National accounts: US$ at current prices…	GDP (current US$)
NY.GDP.MKTP.CN	Economic Policy & Debt: National accounts: Local currency at cur…	GDP (current LCU)
NY.GDP.MKTP.CN.AD	Economic Policy & Debt: National accounts: Local currency at cur…	GDP: linked series (current LCU)
NY.GDP.MKTP.KD	Economic Policy & Debt: National accounts: US$ at constant 2015 …	GDP (constant 2015 US$)
NY.GDP.MKTP.KD.ZG	Economic Policy & Debt: National accounts: Growth rates	GDP growth (annual %)
NY.GDP.MKTP.KN	Economic Policy & Debt: National accounts: Local currency at con…	GDP (constant LCU)
NY.GDP.MKTP.PP.CD	Economic Policy & Debt: Purchasing power parity	GDP, PPP (current international $)
NY.GDP.MKTP.PP.KD	Economic Policy & Debt: Purchasing power parity	GDP, PPP (constant 2021 international $)
NY.GDP.NGAS.RT.ZS	Environment: Natural resources contribution to GDP	Natural gas rents (% of GDP)
NY.GDP.PCAP.CD	Economic Policy & Debt: National accounts: US$ at current prices…	GDP per capita (current US$)
NY.GDP.PCAP.CN	Economic Policy & Debt: National accounts: Local currency at cur…	GDP per capita (current LCU)
NY.GDP.PCAP.KD	Economic Policy & Debt: National accounts: US$ at constant 2015 …	GDP per capita (constant 2015 US$)

Figura 4.3: *Dataframe* contendo os indicadores diretamente dependentes de *GDP growth (annual %)* (visualizado pela interface do Spyder).

WDICountry: Lista os países, regiões e informações contextuais intrínsecas aos mesmos como, por exemplo, a região a que pertence, nome da moeda corrente, ano-base para o cálculo de indicadores e o último ano de realização de censos demográficos;

WDISeries: Lista todos os indicadores socioeconômicos com abreviaturas, nomes completos, textos de descrição, unidades com os quais são medidos, metodologias para a aferição, entre vários outros;

WDIcountry-series: Mapeia relacionamentos entre países e indicadores, denotando as fontes dos valores de cada indicador para cada país;

WDIseries-time: Contém relacionamentos entre alguns países e alguns anos, adicionando informação contextual relevante;

WDIfootnote: Notas de rodapé com observações relevantes para algumas aferições de alguns países, como, por exemplo, o grau de incerteza.

4.3 Processamento e Modelagem

Os artefatos construídos são *scripts* na linguagem de programação **Python**. Em ordem de execução, o primeiro *script* é responsável por processar o arquivo de Valores Separados por Vírgula (CSV), transformar os dados para um formato mais apropriado à modelagem e exportar esses dados resultantes para um novo arquivo CSV. O segundo é responsável pelo pré-processamento desses dados, tornando-os adequados para o terceiro e último, o mais extenso deles, responsável pela construção de um modelo de Aprendizado de Máquina (AM) capaz de minerar dados e obter conhecimento sobre essa base.

O segundo e terceiro *scripts* também são responsáveis por renderizar gráficos para uma visualização analítica descrevendo características a respeito da base de dados, assim como a visualização do resultado do modelo - dessa forma, o algoritmo constrói um modelo que é reproduzível a partir dos *scripts* e exibido a partir dos gráficos gerados.

Dentro dos *scripts*, o modelo será treinado a partir de um conjunto de treinamento e avaliado a partir de um conjunto de teste. Essa modelagem depende diretamente da etapa de pré-processamento dos dados, que dispõe de alguns parâmetros. Ao fim da modelagem, será possível determinar sua qualidade de predição com base nos resultados da sua aplicação sobre o conjunto de teste. Será possível, também, visualizar quais atributos são considerados os mais determinantes pelo modelo para realizar as predições.

4.4 Transformação e Carga dos Dados

A base de dados WDI dispõe os anos como colunas, ordenadas em ordem crescente, contendo os valores dos indicadores (que, por sua vez, são dispostos como linhas, representando cada país ou região).

Esse formato não é interessante para a aplicação nessa tarefa de regressão, que tem o objetivo de descobrir padrões sobre indicadores. Um outro tipo de mineração que poderia se beneficiar do formato original dessa base de dados seria uma tarefa de regressão para prever valores específicos para um ano que não consta na base de dados (por exemplo, no futuro).

É necessária, então, uma transformação sobre a base de dados que viabilize o objetivo deste trabalho. Essa transformação busca representar os anos como linhas e os indicadores como colunas. Dessa forma, tanto a descoberta de padrões sobre indicadores quanto a seleção de quais indicadores mais impactam esses padrões são facilitadas.

Tal transformação foi feita com o Pandas submetendo o *dataframe* original (mostrado na Figura 4.4), aos métodos `melt` e `pivot` [47], sucessivamente (ver Apêndice A). As

figuras Figura 4.5 e Figura 4.6 mostram o estado do *dataframe* após a aplicação de `melt` e `pivot`, respectivamente.

- O `melt` converte as colunas de anos em linhas, criando um novo atributo para armazenar os valores observados em cada ano. Para cada registro original, são geradas 64 linhas (uma para cada ano). Atributos como "Nome do País" e "Nome do Indicador" permanecem inalterados. Essa operação é útil para reorganizar os dados em um formato mais adequado para análises temporais.
- O `pivot` transforma os valores de um ou mais atributos em novas colunas. Neste caso, o atributo "Código do Indicador" foi utilizado para criar colunas com os valores observados. Durante esse processo, o atributo "Nome do Indicador" é perdido, mas essa informação pode ser recuperada posteriormente a partir da base de metadados, utilizando o código do indicador.

Country Name	Country Code	Indicator Name	Indicator Code	1960	1961	1962	1963
Albania	ALB	Population ages 65-69, male (% of male population)	SP.POP.6569.MA.5Y	1.45798	1.4766	1.52173	1.57066
Albania	ALB	Population ages 70-74, female (% of female population)	SP.POP.7074.FE.5Y	1.78195	1.7001	1.61665	1.54833
Albania	ALB	Population ages 70-74, male (% of male population)	SP.POP.7074.MA.5Y	1.25739	1.1799	1.10039	1.03302
Albania	ALB	Population ages 75-79, female (% of female population)	SP.POP.7579.FE.5Y	1.2925	1.29623	1.29487	1.28154
Albania	ALB	Population ages 75-79, male (% of male population)	SP.POP.7579.MA.5Y	0.846119	0.84596	0.843368	0.833143
Albania	ALB	Population ages 80 and above, female (% of female population)	SP.POP.80UP.FE.5Y	1.48591	1.47107	1.45318	1.4331
Albania	ALB	Population ages 80 and above, male (% of male population)	SP.POP.80UP.MA.5Y	0.868032	0.854566	0.840647	0.827879
Albania	ALB	Population density (people per sq. km of land area)	EN.POP.DNST	nan	60.5766	62.4569	64.3292
Albania	ALB	Population growth (annual %)	SP.POP.GROW	nan	3.12086	3.05673	2.95375
Albania	ALB	Population in largest city	EN.URB.LCTY	134761	137714	139561	141434
Albania	ALB	Population in the largest city (% of urban population)	EN.URB.LCTY.UR.ZS	27.2805	26.8139	26.2943	25.8125
Albania	ALB	Population in urban agglomerations of more than 1 million	EN.URB.MCTY	nan	nan	nan	nan
Albania	ALB	Population in urban agglomerations of more than 1 million (% …	EN.URB.MCTY.TL.ZS	nan	nan	nan	nan
Albania	ALB	Population living in areas where elevation is below 5 meters …	EN.POP.EL5M.ZS	nan	nan	nan	nan
Albania	ALB	Population living in slums (% of urban population)	EN.POP.SLUM.UR.ZS	nan	nan	nan	nan

Figura 4.4: Trecho do *Dataframe* original, obtido a partir da base WDI original (visualizado pela interface do Spyder).

A nova base de dados transformada é então carregada (armazenada) no mesmo formato CSV em um arquivo com nome `WDItransformada`. Essa nova base de dados é composta de **17.024 linhas** (produto dos 266 países/regiões por 64 anos) e **1495 colunas** (sendo 3 para a descrição dos países/regiões e indicadores; e mais 1492 indicadores).

4.5 Pré-processamento de Dados

A base de dados `WDItransformada` possui uma quantidade enorme de valores nulos observados ao longo dos anos para todos os indicadores. Ao todo, são 16.572.594 aferições vazias de indicadores, resultando em aproximadamente 66% da quantidade total (Figura 4.7).

Índice	Country Name	untry Co	Indicator Code	Year	Value
75652	Albania	ALB	SP.POP.5054.FE.5Y	1960	3.60264
75653	Albania	ALB	SP.POP.5054.MA.5Y	1960	3.1755
75654	Albania	ALB	SP.POP.5559.FE.5Y	1960	2.7632
75655	Albania	ALB	SP.POP.5559.MA.5Y	1960	2.36419
75656	Albania	ALB	SP.POP.6064.FE.5Y	1960	2.86571
75657	Albania	ALB	SP.POP.6064.MA.5Y	1960	2.20846
75658	Albania	ALB	SP.POP.65UP.TO.ZS	1960	5.48908
75659	Albania	ALB	SP.POP.65UP.FE.IN	1960	51820
75660	Albania	ALB	SP.POP.65UP.FE.ZS	1960	6.60087
75661	Albania	ALB	SP.POP.65UP.MA.IN	1960	36488
75662	Albania	ALB	SP.POP.65UP.MA.ZS	1960	4.42952
75663	Albania	ALB	SP.POP.65UP.TO	1960	88308
75664	Albania	ALB	SP.POP.6569.FE.5Y	1960	2.04051
75665	Albania	ALB	SP.POP.6569.MA.5Y	1960	1.45798
75666	Albania	ALB	SP.POP.7074.FE.5Y	1960	1.78195
75667	Albania	ALB	SP.POP.7074.MA.5Y	1960	1.25739
75668	Albania	ALB	SP.POP.7579.FE.5Y	1960	1.2925
75669	Albania	ALB	SP.POP.7579.MA.5Y	1960	0.846119
75670	Albania	ALB	SP.POP.80UP.FE.5Y	1960	1.48591
75671	Albania	ALB	SP.POP.80UP.MA.5Y	1960	0.868032
75672	Albania	ALB	EN.POP.DNST	1960	nan
75673	Albania	ALB	SP.POP.GROW	1960	nan

Figura 4.5: Trecho do *Dataframe* original após a aplicação do método melt (visualizado pela interface do Spyder).

Essa proporção torna consideravelmente difícil construir qualquer modelo preditivo com uma boa qualidade [48].

Sendo assim, é indispensável, na etapa de pré-processamento dos dados, buscar reduzir esse "buraco" na base. Isso será feito com dois métodos diferentes, que são explicados nas subseções seguintes.

Ao longo dessa Seção, a base de dados será ilustrada através de figuras com análise da base de dados a partir do *Spyder* e do *Matplotlib*, representando a base antes e depois do pré-processamento.

Country Name	Country Code	Year	AG.AGR.TRAC.NO	AG.CON.FERT.PT.ZS	AG.CON.FERT.ZS	AG.LND.AGRI.K2	AG.LND.AGRI.ZS	AG.LND.ARBL.HA
Arab World	ARB	2018	nan	14.584	55.9942	5.23625e+06	39.9697	nan
Arab World	ARB	2019	nan	14.0102	55.5936	5.22804e+06	39.907	nan
Arab World	ARB	2020	nan	13.0246	54.6098	5.23672e+06	39.9733	nan
Arab World	ARB	2021	nan	13.0369	55.0926	5.23639e+06	39.9707	nan
Arab World	ARB	2022	nan	nan	nan	nan	nan	nan
Arab World	ARB	2023	nan	nan	nan	nan	nan	nan
Argentina	ARG	1960	nan	nan	nan	nan	nan	nan
Argentina	ARG	1961	120000	523.903	0.873313	1.37829e+06	50.3634	1.8597e+07
Argentina	ARG	1962	130000	211.85	0.662722	1.36434e+06	49.8537	1.918e+07
Argentina	ARG	1963	140000	421.6	1.07551	1.34875e+06	49.284	1.96e+07
Argentina	ARG	1964	150000	590	1.475	1.33297e+06	48.7074	2e+07
Argentina	ARG	1965	155000	810	1.97938	1.3178e+06	48.1531	2.0461e+07
Argentina	ARG	1966	160000	1413.88	2.4328	1.30948e+06	47.849	2.1608e+07
Argentina	ARG	1967	165000	604.183	3.06432	1.29703e+06	47.3941	2.2341e+07

Figura 4.6: Trecho do *Dataframe* tratado com a aplicação das funções melt e pivot (visualizado pela interface do Spyder).

```
In [3]: total_nan = wdi.isna().sum().sum()

In [4]: total_nan
Out[4]: 16752594

In [5]: total_values = total_countries * total_indicators * total_years

In [6]: total_values
Out[6]: 25304320

In [7]: total_nan / total_values
Out[7]: 0.6620448208053012
```

Figura 4.7: Cálculo da quantidade de valores nulos no *Dataframe* original, antes do processo de redução de dados (visualizado pelo console do Spyder).

4.5.1 Remoção de Valores Vazios

A redução de dados será feita com diferentes estratégias, visando reduzir a quantidade de valores vazios, o que aumenta a precisão das informações fornecidas à modelagem. Ao mesmo tempo, é preciso lembrar que essa "poda" muitas vezes tem como efeito colateral a perda de dados reais, úteis, não nulos. Sendo assim, é necessário encontrar um certo equilíbrio de forma a manter a qualidade da base de dados.

Há trabalhos que propõem redução de dados por meio de análise de dependências semânticas, como o de Zaidi et al. [49].

A primeira etapa de redução de dados é remover todos os registros em que o valor da variável-alvo **Crescimento Anual do PIB** é vazio. Como se trata de um algoritmo de

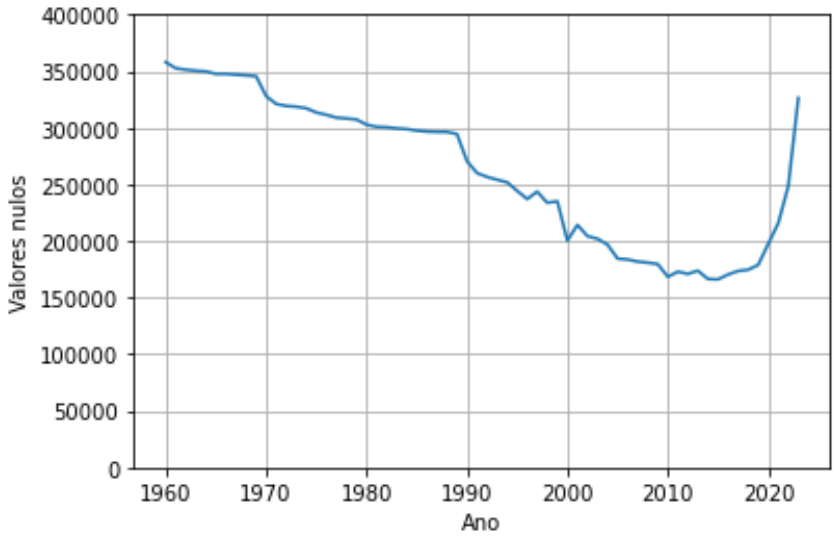

Figura 4.8: Quantidade de valores nulos observados em cada ano dentro da base original, considerando todos os países/regiões e indicadores.

aprendizado supervisionado, todos os dados de treinamento precisam de rótulos informados (neste caso, numéricos). Logo, não faz sentido manter na base as tuplas onde o rótulo desejado é desconhecido.

Esse é o caso de 3.173 registros (18,6% do total); após esse tratamento, a base de dados é reduzida de 17.024 para 13.851 registros.

Em seguida, serão aplicadas três etapas de redução de dados: redução de anos, redução de países e redução de indicadores, todas com o objetivo de eliminar aqueles com maior prevalência de valores vazios. As etapas são explicadas nos próximos parágrafos.

Serão retirados da base todos os registros pertencentes a anos com menos dados válidos. É possível visualizar a quantidade de valores nulos agrupados por cada ano na Figura 4.8, na qual nota-se que os dados mais antigos são os mais propensos a estarem ausentes, assim como o ano mais recente (2023). A quantidade de anos a remover da base é determinada pelo parâmetro `YEARS TO DROP`.

Em seguida, remover-se-ão do *Dataframe* todos os registros dos países mais "vazios", isto é, com a maior proporção de valores ausentes. A lista desses países é vista na Figura 4.9, onde é possível perceber a prevalência de pequenas ilhas não soberanas e países de menor território. A quantidade de países a remover é determinada pelo parâmetro `COUNTRIES TO DROP`.

Por último, outro jeito de reduzir os dados é "podar" os indicadores com dados mais vazios, efetivamente reduzindo a quantidade de atributos a serem considerados no treinamento - diferente das reduções anteriores em que a base perdia registros ao invés de atributos.

Country Code	Country Name	NaN values
INX	Not classified	95488
MAF	St. Martin (French part)	90013
SXM	Sint Maarten (Dutch part)	87659
IMN	Isle of Man	87572
MNP	Northern Mariana Islands	87519
CHI	Channel Islands	87251
CUW	Curacao	85609
GIB	Gibraltar	85536
ASM	American Samoa	85355
MCO	Monaco	85070
VGB	British Virgin Islands	84634
LIE	Liechtenstein	84264
TCA	Turks and Caicos Islands	84198
VIR	Virgin Islands (U.S.)	83824
GUM	Guam	83667
FRO	Faroe Islands	83512
CYM	Cayman Islands	83076
GRL	Greenland	82438
SMR	San Marino	81964
AND	Andorra	81523

Figura 4.9: Lista de países e territórios com maior ausência de valores na base original (visualizado pela interface do Spyder).

Isso pode ser feito de duas formas: remover uma quantidade parametrizada de indicadores com mais dados ausentes; ou estabelecer um limiar no qual todos os indicadores que tiverem uma quantidade de dados válidos menor que esse valor são eliminados da base. Neste projeto foi adotado o segundo caminho. O limiar de valores não-nulos a ser respeitado pelos indicadores para se manterem na base filtrada é determinado pelo parâmetro `INDICATORS_NOT_NAN_THRESHOLD`.

Na figura Figura 4.10 são mostrados os indicadores mais vazios e, na Figura 4.11, um histograma mostrando a frequência de valores vazios, onde nota-se que a grande maioria dos indicadores ultrapassa 8.500 valores nulos, ou seja, mais da metade do total de 17.024 registros, aproximadamente.

Ao longo de todas as etapas de redução de dados, os três parâmetros acima são de suma importância e podem alterar drasticamente a base de dados resultante, em qualidade e quantidade de dados (existem ainda outros parâmetros para a modelagem, a serem usados em outras partes do trabalho, como nas seções 4.6 e 4.7. É executado o pré-processamento

Name	NaN values	Percentage
Net official flows from UN agencies, UNIDIR (current US$)	17024	100
Net official flows from UN agencies, UNCTAD (current US$)	17004	99.88
Disaster risk reduction progress score (1-5 scale; 5=best)	16941	99.51
Female genital mutilation prevalence (%)	16933	99.47
Present value of external debt (% of exports of goods, services and income)	16931	99.45
Present value of external debt (% of GNI)	16929	99.44
Children in employment, self-employed, female (% of female children in employment, a…	16926	99.42
Children in employment, self-employed, male (% of male children in employment, ages …	16926	99.42
Children in employment, self-employed (% of children in employment, ages 7-14)	16926	99.42
Proportion of women subjected to physical and/or sexual violence in the last 12 mont…	16918	99.38
Public private partnerships investment in ICT (current US$)	16914	99.35
Multidimensional poverty headcount ratio (UNDP) (% of population)	16914	99.35
Adequacy of unemployment benefits and ALMP (% of total welfare of beneficiary househ…	16911	99.34
Benefit incidence of unemployment benefits and ALMP to poorest quintile (% of total …	16911	99.34
Present value of external debt (current US$)	16910	99.33
Women making their own informed decisions regarding sexual relations, contraceptive …	16910	99.33
Net official flows from UN agencies, UNCDF (current US$)	16904	99.3
Net official flows from UN agencies, UNWTO (current US$)	16901	99.28
Average working hours of children, working only, female, ages 7-14 (hours per week)	16900	99.27
Annualized average growth rate in per capita real survey mean consumption or income,…	16899	99.27
Annualized average growth rate in per capita real survey mean consumption or income,…	16899	99.27
Net official flows from UN agencies, UNEP (current US$)	16898	99.26
Average working hours of children, working only, male, ages 7-14 (hours per week)	16898	99.26
Average working hours of children, working only, ages 7-14 (hours per week)	16898	99.26

Figura 4.10: Lista de indicadores com mais valores nulos na base original (visualizado pela interface do Spyder).

com os seguintes valores para os parâmetros, conforme o Código 4.1:

```
YEARS_TO_DROP = 16                    # 1/4 do total de anos
COUNTRIES_TO_DROP = 28                # aprox. 10% dos 266 países e regiões
INDICATORS_NOT_NAN_THRESHOLD = 0.6    # exclui todo indicador com + de 40% valores nulos
```

Código 4.1: Declaração das variáveis de parâmetros no início do script de modelagem.

Ao fim da redução de dados com os parâmetros do Código 4.1, a base agora contém 10.653 registros e 551 colunas (portanto 549 indicadores). A proporção de valores nulos na base de dados diminuiu de aproximadamente **66%** para aproximadamente **20%** (Figura 4.12) - uma melhora significativa.

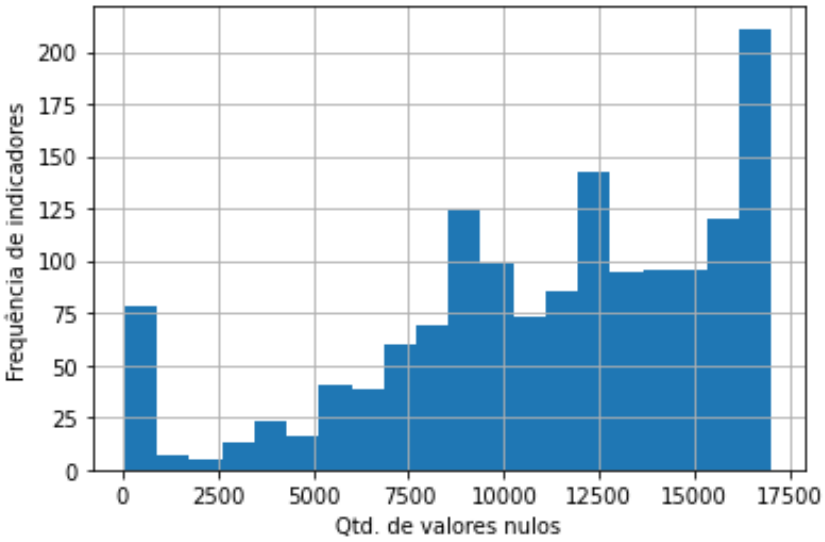

Figura 4.11: Histograma mostrando a frequência de valores nulos aferidos em cada indicador.

A base de dados resultante é armazenada em formato CSV em arquivo de nome `WDIPreProcessada` [2].

4.5.2 Inferência de Valores Vazios

A inferência de valores vazios tem como objetivo eliminar a ocorrência de valores nulos em toda a base, preenchendo esses valores seguindo alguma estratégia. Existem diversos métodos para a inferência de vazios, explicados na Seção 2.3.4.

Essa etapa não é necessária para a construção do modelo de predição para o indicador do PIB (detalhado na Seção 4.8), mas é necessária para o modelo de seleção de indicadores (detalhado na Seção 4.7), cujo resultado será utilizado posteriormente no treinamento do modelo principal.

Além das estratégias triviais, como atribuir o valor 0 ou uma constante qualquer a todos os valores nulos, não foram utilizadas medidas estatísticas como a média, moda e mediana dos atributos, já que os valores observados de um indicador em todos os países do mundo não podem ser generalizados para inferir a evolução do mesmo em um país particular.

Pode-se inferir que um valor ausente de um certo indicador em um certo país, região ou território siga a tendência dos valores observados mais "próximos", isto é: valores válidos de anos próximos do mesmo indicador e mesmo país, ou valores de indicadores

[2]Nas seções subsequentes do estudo, incluindo a modelagem propriamente dita, a base usada é a `WDIPreProcessada`, a qual poderá ser referida simplesmente como `WDI`, de forma sinônima.

```
In [5]: total_nan = wdi.isna().sum().sum()

In [6]: total_nan
Out[6]: 1242059

In [7]: total_values = total_countries * total_indicators * total_years

In [8]: total_values
Out[8]: 6234048

In [9]: total_nan / total_values
Out[9]: 0.1992379590275853
```

Figura 4.12: Cálculo da quantidade de valores nulos no *Dataframe* após as três etapas de redução de dados (visualizado pelo console do Spyder).

correlacionados, ou valores de países com valores semelhantes de indicadores. Tendo isso em mente, foi adotada a estratégia preditiva de utilizar o algoritmo de Aprendizado de Máquina dos **k vizinhos mais próximos** ou *K-Nearest Neighbors* (KNN) para inferir os valores ausentes, cuja implementação no Scikit-Learn é chamada `KNNImputer` [50]. A aplicação de KNN com esse fim também é explorada por Murti et al. [51].

Dessa forma, é aplicado um modelo de Aprendizado de Máquina de forma a auxiliar um outro modelo (o último sendo a finalidade da tarefa de mineração) a ser mais eficiente. Nesta aplicação de regressão com KNN, os dados de treinamento são todos os valores presentes da base de dados. Assim, para cada valor de indicador a ser inferido, tal indicador será o rótulo desejado e os valores ausentes serão aproximados de acordo com os registros mais "próximos" (semelhantes).

Ao final do treinamento e estimação do KNN, com o parâmetro listado no Código 4.2, obtém-se uma base de dados sem nenhum valor ausente restante. Esse parâmetro é capaz de influenciar as aproximações obtidas e, portanto, influenciar as decisões do modelo final na Seção 4.8.

```
1 KNN_IMPUTER_NEIGHBOURS = 10
```

Código 4.2: Parâmetro da quantidade de vizinhos que o KNN considera.

4.6 Conjuntos de Testes e Treinamento

Após o processo de limpeza dos dados, é feita a separação entre o conjunto de teste e o conjunto de treinamento.

Diversas estratégias foram cogitadas, como, por exemplo, realizar uma separação temporal - anos anteriores a um ano x seriam parte do conjunto de treinamento e os anos

restantes, mais recentes, seriam o conjunto de teste. Foi analisado também utilizar uma separação heurística por países - países específicos em cada continente, nesse caso, comporiam o conjunto de teste.

No entanto, tais estratégias consistiriam em critérios de seleção definidos diretamente pelos autores, sujeitos aos seus vieses individuais, o que poderia influenciar o resultado final sem uma análise mais aprofundada e heurística de seu impacto. Sendo assim, foi implementada uma separação entre teste e treinamento totalmente aleatória, com proporção correspondente ao parâmetro declarado no Código 4.3 - onde o valor declarado é a proporção do tamanho do conjunto de teste em relação ao todo. A separação aleatória é implementada com o método `train_test_split` [52] do Scikit-Learn.

```
1 TEST_SET_RATIO = 0.25
```

Código 4.3: Parâmetro do tamanho do conjunto de treinamento.

Como explicado na Seção 4.2.2, é realizada nessa etapa a separação entre a classe, de código `NY.GDP.MKTP.KD.ZG` e agora denotada como `y`, e o conjunto restante dos atributos, denotados como `X`. Essa fragmentação é necessária para a separação entre conjuntos de teste e treinamento, pois o `train_test_split` requer ambos `X` e `y` como parâmetros. Logo, existem agora 4 subconjuntos distintos do dataframe WDI: `X_train`, `X_test`, `Y_train` e `Y_test`.

É importante salientar que, além da classe `NY.GDP.MKTP.KD.ZG`, também foram excluídos do conjunto de atributos `X` os chamados *indicadores triviais* (alguns deles mostrados na Figura 4.3). Vários deles possuem em seu título, por exemplo, o sufixo `"(% of GDP)"`. Isso visa construir um modelo de predição do crescimento do PIB mais generalizado e sem o viés dos atributos que dependem fortemente da classe, assim como aumentar a possibilidade de detectar padrões não triviais na seleção de atributos.

4.7 Seleção de Atributos

Em seguida, será feita a seleção de atributos, realizada de forma algorítmica pelo *Scikit-Learn*. Dentre as muitas metodologias possíveis, como as de limiares de variância ou a eliminação recursiva de atributos [53], foi escolhido o algoritmo `SelectKBest`, considerado um método *univariável*. Ele consiste em manter apenas um número pré-determinado (pelo usuário) dos melhores atributos no *dataframe*.

O critério para quantificar o que é "melhor" consiste em cálculos estatísticos sobre as variáveis de interesse em relação à variável-alvo, a fim de medir a correlação entre elas. O cálculo específico a ser usado também é determinado pelo usuário; neste caso, foi

escolhido o `r_regression`, que calcula o coeficiente de determinação (R^2) (vide equação Coeficiente de determinação) entre variáveis em problemas de regressão, que lidam com o domínio dos números reais.

Escolhido o critério de filtragem dos atributos, o algoritmo será executado usando como argumentos os conjuntos `X_train` (características) e `Y_train` (rótulo desejado). A quantidade de atributos a serem preservados é determinada pelo parâmetro declarado no Código 4.4. A partir do resultado da seleção, esse filtro precisa ser aplicado não apenas sobre `X_train` mas também sobre `X_test`.

```
FEATURES_TO_SELECT = 32
```

Código 4.4: Parâmetro da quantidade de melhores atributos a serem selecionados.

Analisar quais atributos foram escolhidos pelo modelo `SelectKBest` é um dos objetivos centrais do trabalho, por isso essa informação será detalhada na Seção 5.1.

4.8 Modelo de *Random Forest*

Com o término da seleção de atributos pelo `SelectKBest`, é iniciada a construção do modelo de *Random Forest*. Como explicado na Seção 2.5.3, o *Random Forest* é um algoritmo de Aprendizado de Máquina que consiste em utilizar vários subconjuntos de dados de treinamento para construir uma série de árvores de decisão. Na biblioteca *Scikit-Learn*, ele é implementado como `RandomForestRegressor` [54].

Como visto nas seções 2.5.2 e 2.5.3, as florestas aleatórias combinam vantagens das árvores de decisão, como a flexibilidade, já que os dados não precisam passar por normalização nem inferência de valores (explicados na 2.3.4), com a capacidade de generalização melhorada através do uso de um conjunto grande de árvores e de amostragem aleatória sobre o conjunto de treinamento. Esses motivos pesaram na escolha desse algoritmo na execução da tarefa deste trabalho.

O modelo é treinado utilizando o subconjunto da seleção aplicada aos atributos de características (`X_train_selected`) e à variável-alvo (`Y_train`) no código 4.5.

```
random_forest = RandomForestRegressor(random_state=0)
random_forest.fit(X_train_selected, y_train)
```

Código 4.5: Treinamento do modelo de Random Forest

O único parâmetro explicitamente declarado para o algoritmo é o `random_state`, que garante uma execução aleatória inédita cada vez que o *script* for executado. No entanto,

o modelo possui diversos parâmetros que, por não estarem explicitados, são definidos de acordo com o padrão do *Scikit-Learn*, conforme ilustrado na Figura 4.13. Entre eles, os de maior relevância incluem: `n_estimators`, que determina a quantidade de árvores de decisão individuais usadas dentro da floresta (no caso, 100 árvores); `max_depth` que, caso presente, determina um limite para a profundidade de cada árvore; e `max_features`, que determina o limite máximo de atributos analisados em cada nó de decisão.

```
In [22]: random_forest.get_params()
Out[22]:
{'bootstrap': True,
 'ccp_alpha': 0.0,
 'criterion': 'squared_error',
 'max_depth': None,
 'max_features': 1.0,
 'max_leaf_nodes': None,
 'max_samples': None,
 'min_impurity_decrease': 0.0,
 'min_samples_leaf': 1,
 'min_samples_split': 2,
 'min_weight_fraction_leaf': 0.0,
 'n_estimators': 100,
 'n_jobs': None,
 'oob_score': False,
 'random_state': 0,
 'verbose': 0,
 'warm_start': False}
```

Figura 4.13: Parâmetros do modelo Random Forest.

Após a aplicação do modelo, seu desempenho pode ser avaliado pelo método `score()` do `RandomForestRegressor`, que calcula o coeficiente de determinação (R^2) do modelo [54] (visto na Equação Coeficiente de determinação na Seção 2.5.5) ao executá-lo no conjunto de teste. Um valor próximo de 1 indica que o modelo consegue explicar a maior parte da variabilidade dos dados, enquanto valores próximos de 0 indicam que o modelo não está capturando bem a relação entre os atributos e a variável-alvo. Este método é chamado de acordo com o código 4.6 e o resultado aferido é analisado na Seção 5.3.

```
score = random_forest.score(X_test_selected, y_test)
```

Código 4.6: Método para avaliação do desempenho do modelo de *Random Forest*

Após todos os passos do desenvolvimento, descritos no Capítulo 4, os *scripts* foram executados localmente e, os resultados dessa execução, coletados para serem apresentados e analisados no Capítulo 5.

Capítulo 5

Resultados dos modelos

A ordem da apresentação dos resultados neste Capítulo corresponde à ordem do desenvolvimento do trabalho como ilustrado na Figura 4.1. Inicialmente, na Seção 5.1, são analisados os resultados da seleção de atributos (desenvolvida na Seção 4.7). Na Seção 5.2, é exposto o modelo de regressão gerado na Seção 4.8. Na Seção 5.3, é analisado o desempenho desse modelo na sua tarefa de predição do PIB. Por fim, todos os resultados são sumarizados em uma visualização panorâmica no Quadro 5.1.

5.1 Resultado da Seleção de Atributos

O algoritmo `SelectKBest`, mencionado na Seção 4.7 e executado com o parâmetro informado `FEATURES_TO_SELECT`, foi executado sobre o conjunto de características de treinamento (`X_train`) e a variável de interesse respectiva, também de treinamento, (`Y_train`). Os indicadores restantes no *dataframe* `X_train`, após o procedimento, estão listados na Figura 5.1 e na Figura 5.2. Esse mesmo filtro de indicadores foi, então, aplicado ao *dataframe* do conjunto de teste `X_test`.

Com a obtenção dos indicadores mais correlacionados à análise do crescimento do PIB, a última afirmação da **Hipótese 1** é corroborada - ela dispunha sobre a capacidade do modelo de selecionar os atributos conforme sua relevância ou *score* comparado com o indicador desejado.

Os padrões de correlação detectados pela filtragem de atributos listada na Figura 5.1 e Figura 5.2 revelam, acima de tudo, o domínio de indicadores econômicos expressos em termos de *"annual % growth"* (crescimento percentual anual). Por estarem representados na mesma grandeza do **Crescimento Anual do PIB (em porcentagem anual)**, que é a de pontos percentuais, eles possuem mais correlação entre si.

Series Code	Topic	Indicator Name	Score
NV.IND.TOTL.KD.ZG	Economic Policy & Debt: National accou…	Industry (including construction), value added (annual % growth)	0.604597
NV.SRV.TOTL.KD.ZG	Economic Policy & Debt: National accou…	Services, value added (annual % growth)	0.494535
NE.CON.PRVT.PC.KD.ZG	Economic Policy & Debt: National accou…	Household final consumption expenditure per capita growth (annual %)	0.377761
NE.CON.TOTL.KD.ZG	Economic Policy & Debt: National accou…	Final consumption expenditure (annual % growth)	0.316495
NE.CON.PRVT.KD.ZG	Economic Policy & Debt: National accou…	Household and NPISHs Final consumption expenditure (annual % growth)	0.306099
NE.IMP.GNFS.KD.ZG	Economic Policy & Debt: National accou…	Imports of goods and services (annual % growth)	0.303785
NE.EXP.GNFS.KD.ZG	Economic Policy & Debt: National accou…	Exports of goods and services (annual % growth)	0.26345
NE.GDI.FTOT.KD.ZG	Economic Policy & Debt: National accou…	Gross fixed capital formation (annual % growth)	0.25822
NV.IND.MANF.KD.ZG	Economic Policy & Debt: National accou…	Manufacturing, value added (annual % growth)	0.220648
NV.AGR.TOTL.KD.ZG	Economic Policy & Debt: National accou…	Agriculture, forestry, and fishing, value added (annual % growth)	0.215103
NE.GDI.TOTL.KD.ZG	Economic Policy & Debt: National accou…	Gross capital formation (annual % growth)	0.197489
SP.POP.GROW	Health: Population: Dynamics	Population growth (annual %)	0.185867
SP.URB.GROW	Environment: Density & urbanization	Urban population growth (annual %)	0.171718
NY.GNS.ICTR.GN.ZS	Economic Policy & Debt: National accou…	Gross savings (% of GNI)	0.128277
NY.ADJ.NNAT.GN.ZS	Economic Policy & Debt: National accou…	Adjusted savings: net national savings (% of GNI)	0.126959
NE.CON.GOVT.KD.ZG	Economic Policy & Debt: National accou…	General government final consumption expenditure (annual % growth)	0.12561

Figura 5.1: Lista de indicadores selecionados pelo *SelectKBest* ordenada pelo seu *score* - parte 1 (visualizada pela interface do Spyder).

É notável que esses mesmos indicadores expressos em grandezas absolutas (como valor em Dólar americano (USD)) não foram selecionados dentre esses 32 atributos por não apresentarem elevada correlação com a classe de aumento do PIB.

Outros padrões notáveis nessa lista de atributos, especialmente na Figura 5.2, incluem a presença de indicadores a respeito da população dos países - nota-se o *score* maior dos indicadores de proporção de habitantes entre 20 e 24 anos, assim como a presença consistente de vários indicadores de população do sexo feminino com *score* mais altos que os respectivos indicadores para o sexo masculino.

5.2 Estrutura das Árvores de Decisão

O modelo de `RandomForestRegressor` [54] foi construído com 100 árvores de decisão conforme visto na Figura 4.13. A Figura 5.3, a Figura 5.4 e a Figura 5.5 ilustram a estrutura de uma árvore particular dentro do modelo, a "primeira"(acessada no índice 0). Todas as árvores de decisão resultantes do modelo podem ser obtidas através do *array* `estimators_`, um atributo do objeto `RandomForestRegressor`.

As árvores foram criadas sem um limite máximo de profundidade, o que, pela grande quantidade de atributos empregados nas predições (apesar de reduzida com a seleção de atributos) resultou em árvores com potencialmente milhares de nós de decisão (como ilustrado na Figura 5.3, até chegar nos nós-folhas que finalmente atribuem um valor à classe de interesse. Devido a isso, a ilustração completa de uma árvore como na Figura 5.4 é, na prática, ilegível.

SP.POP.2024.FE.5Y	Health: Population: Structure	Population ages 20-24, female (% of female population)	0.116124
SP.POP.2024.MA.5Y	Health: Population: Structure	Population ages 20-24, male (% of male population)	0.112928
NY.ADJ.ICTR.GN.ZS	Economic Policy & Debt: National accou…	Adjusted savings: gross savings (% of GNI)	0.108532
SP.POP.1519.FE.5Y	Health: Population: Structure	Population ages 15-19, female (% of female population)	0.100599
SP.POP.1519.MA.5Y	Health: Population: Structure	Population ages 15-19, male (% of male population)	0.0934546
SP.POP.2529.FE.5Y	Health: Population: Structure	Population ages 25-29, female (% of female population)	0.0774535
SL.FAM.WORK.FE.ZS	Social Protection & Labor: Economic ac…	Contributing family workers, female (% of female employment) (modeled ILO estimate)	0.0765242
SP.POP.1014.FE.5Y	Health: Population: Structure	Population ages 10-14, female (% of female population)	0.0740012
SL.EMP.SELF.MA.ZS	Social Protection & Labor: Economic ac…	Self-employed, male (% of male employment) (modeled ILO estimate)	0.0729717
SL.EMP.VULN.MA.ZS	Social Protection & Labor: Economic ac…	Vulnerable employment, male (% of male employment) (modeled ILO estimate)	0.0712803
SL.EMP.SELF.ZS	Social Protection & Labor: Economic ac…	Self-employed, total (% of total employment) (modeled ILO estimate)	0.0702327
SL.EMP.VULN.ZS	Social Protection & Labor: Economic ac…	Vulnerable employment, total (% of total employment) (modeled ILO estimate)	0.0701591
SL.EMP.VULN.FE.ZS	Social Protection & Labor: Economic ac…	Vulnerable employment, female (% of female employment) (modeled ILO estimate)	0.0699123
NY.ADJ.DRES.GN.ZS	Economic Policy & Debt: National accou…	Adjusted savings: natural resources depletion (% of GNI)	0.0694794
SL.EMP.SELF.FE.ZS	Social Protection & Labor: Economic ac…	Self-employed, female (% of female employment) (modeled ILO estimate)	0.0673386
SP.POP.1014.MA.5Y	Health: Population: Structure	Population ages 10-14, male (% of male population)	0.0657167

Figura 5.2: Lista de indicadores selecionados pelo *SelectKBest* ordenada pelo seu *score* - parte 2 (visualizada pela interface do Spyder).

```
In [10]: random_forest.estimators_[0].tree_.max_depth
Out[10]: 45

In [11]: random_forest.estimators_[0].tree_.node_count
Out[11]: 9965
```

Figura 5.3: Exibição de atributos da árvore de decisão de índice 0 - respectivamente, profundidade máxima e quantidade total de nós da árvore (visualizados pelo console do Spyder).

A Figura 5.5 contém a árvore exibida na Figura 5.4, porém limitada aos 2 primeiros níveis de profundidade. Essa visualização foi gerada para facilitar a legibilidade sobre os nós, pois, como visto na Figura 5.3, essa árvore em particular possui no total quase 10.000 nós, podendo um caminho de decisão arbitrário passar por até 45 nós de profundidade.

As árvores de decisão concentram os atributos mais relevantes no seu topo, onde estão as decisões com maior ganho de informação (vide Seção 2.5.2). Na Figura 5.5, em particular, é possível notar os padrões decorrentes dessa distribuição dos nós de decisão.

No topo da árvore está o `NV.IND.TOTL.KD.ZG` (*Crescimento do valor agregado da indústria [em porcentagem anual]*), onde o limiar de decisão é o valor $-2,4[\%]$, aproximadamente. No nível seguinte de profundidade, ambos os nós-filhos testam o mesmo indicador, o `NV.SRV.TOTL.KD.ZG` (*Crescimento do valor agregado de serviços [em porcentagem anual]*), porém com valores bem distintos para determinar a decisão.

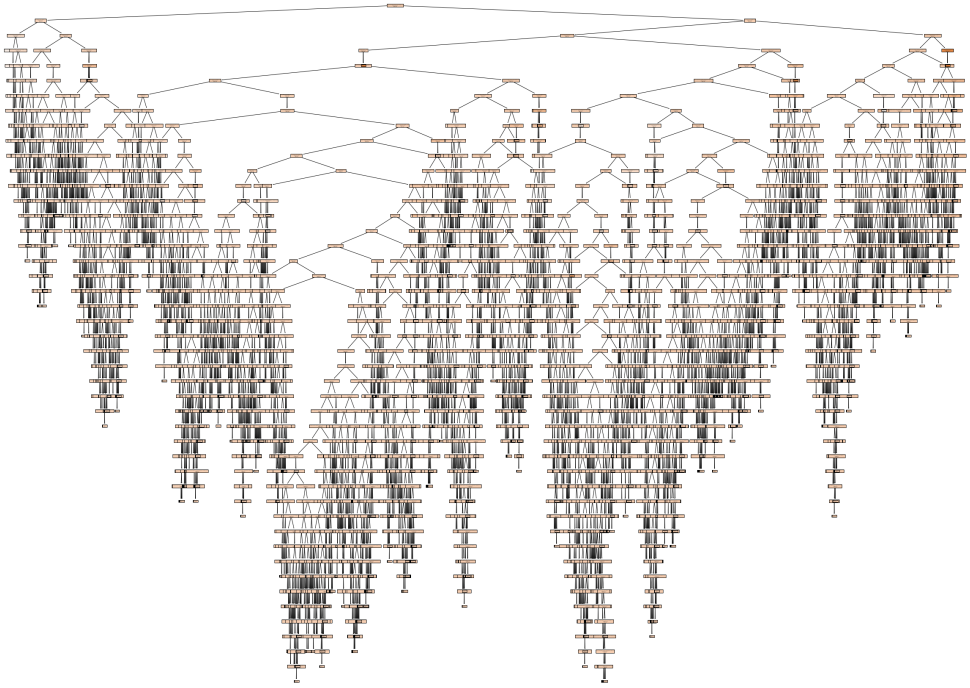

Figura 5.4: Ilustração integral da árvore de decisão de índice 0.

A repetição de atributos em nós é uma característica comum nas árvores construídas com esse algoritmo, visto que o indicador *Crescimento do valor agregado da indústria [em porcentagem anual]*, do nó-raiz, aparece novamente no terceiro nível de decisão, dessa vez com um valor bem diferente ($97,1$[%], aproximadamente). Outro indicador relevante ilustrado nesse nível, por exemplo, é o `NV.AGR.TOTL.KD.ZG` (*Crescimento do valor agregado da agricultura, silvicultura e pescaria [em porcentagem anual]*).

O peso desses atributos na árvore da Figura 5.5 corrobora com a lista exibida na Figura 5.1, onde os mesmos figuram como alguns dos atributos mais determinantes (de *score* mais alto segundo o `SelectKBest`) para o modelo.

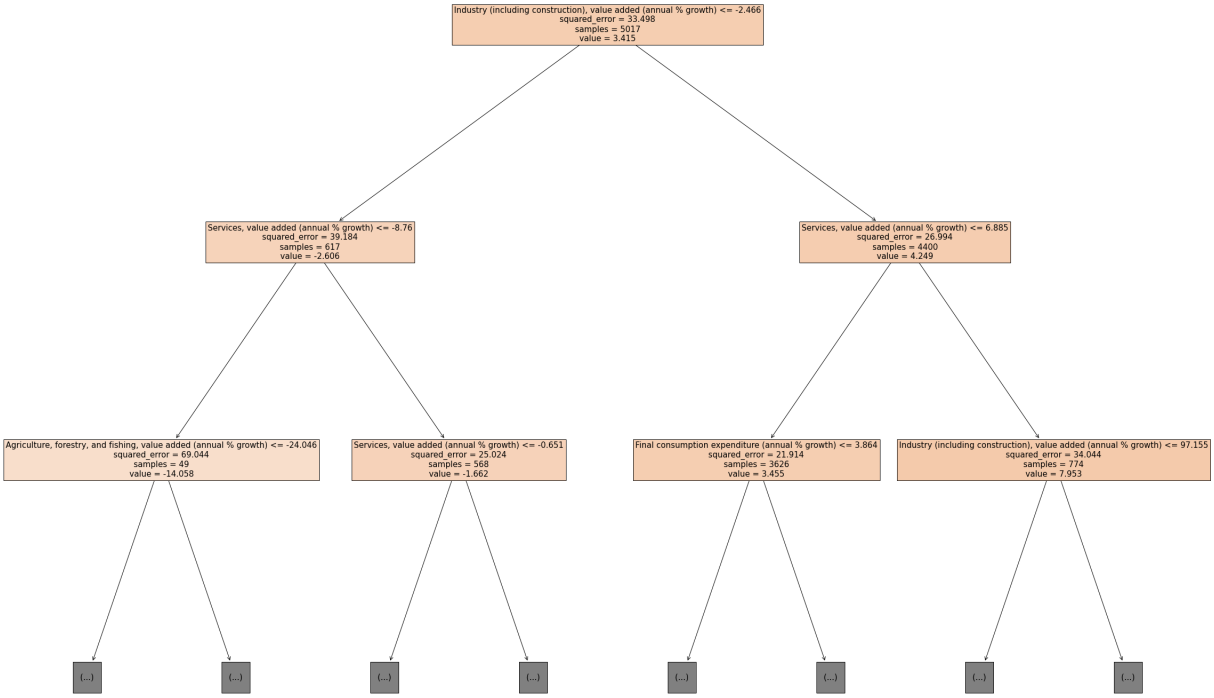

Figura 5.5: Ilustração da árvore de decisão de índice 0 - exibindo os 2 primeiros níveis de profundidade da árvore da Figura 5.4.

5.3 Desempenho da Predição do Indicador de Crescimento do PIB

```
In [16]: score
Out[16]: 0.4587067123411206
```

Figura 5.6: Valor do *score* de predição do indicador Crescimento Anual do PIB (em porcentagem anual) executado sobre o conjunto de teste (visualizado pelo console do Spyder).

Para avaliar o desempenho do modelo, foi utilizado o método `score()`, que calcula o coeficiente de determinação (R^2) (Equação Coeficiente de determinação, conforme também abordado na Seção 4.8).

A Figura 5.6 apresenta uma avaliação quantitativa do desempenho do modelo preditivo `RandomForestRegressor` utilizado no trabalho, exibindo um *score* (R^2) de aproximadamente 46% no conjunto de teste.

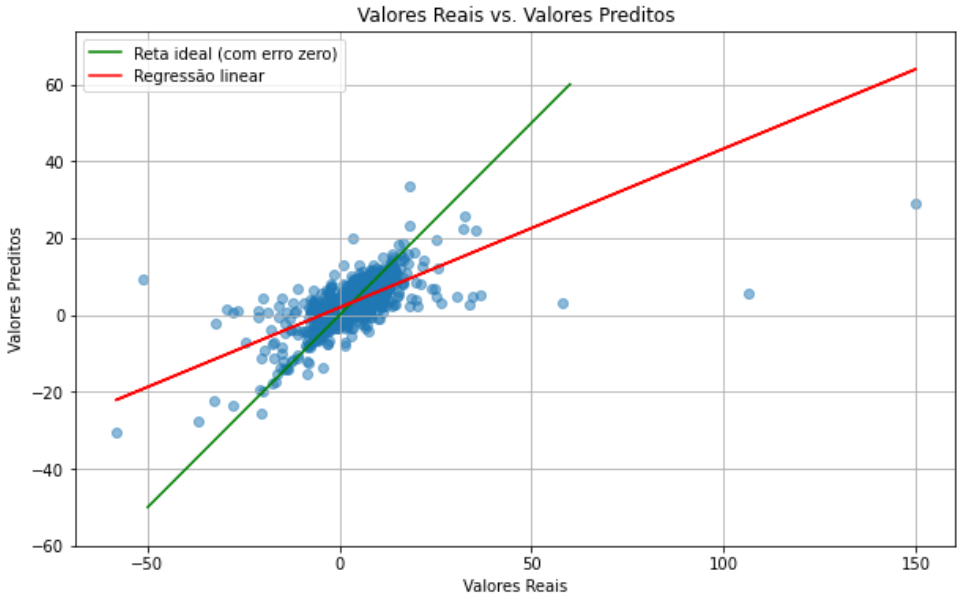

Figura 5.7: Gráfico de dispersão para análise de desempenho do modelo para o indicador Crescimento Anual do PIB (em porcentagem anual), comparando os valores reais do indicador (eixo X) com os valores preditos pelo modelo (eixo Y).

A Figura 5.7 fornece um gráfico de dispersão, com o eixo `X` representando os valores reais e o eixo `Y` as predições do modelo `RandomForestRegressor` para a variável **Crescimento Anual do PIB (em porcentagem anual)**. Com ele, é possível notar a diferença entre a regressão linear obtida com as previsões do modelo e a reta "ideal"(onde o valor predito seria sempre igual ao valor real e, portanto, o erro absoluto seria zero).

Em seguida, foi analisado os resíduos do modelo na Figura 5.8. Os resíduos são gerados a partir da diferença entre os valores reais e o valores preditos. O gráfico se diferencia do gráfico de dispersão pois ajuda a identificar padrões nos erros que podem indicar problemas no conjunto de variáveis. Ele serve para analisar o aumento ou diminuição do erro absoluto do modelo em função dos valores reais observados. Com esse gráfico, é possível notar que o erro absoluto (ou resíduo) se aproxima de uma curva normal na qual os maiores resíduos, em média, estão concentrados nas predições mais perto da mediana.

Tanto na Figura 5.7 quanto na Figura 5.8, é possível notar a presença de *outliers*, ou seja, valores que destoam muito da média tanto dos valores reais quanto dos valores preditos. Esses registros específicos são onde estão concentrados os maiores erros absolutos, o que pode ter impactado o *score* e outras métricas negativamente.

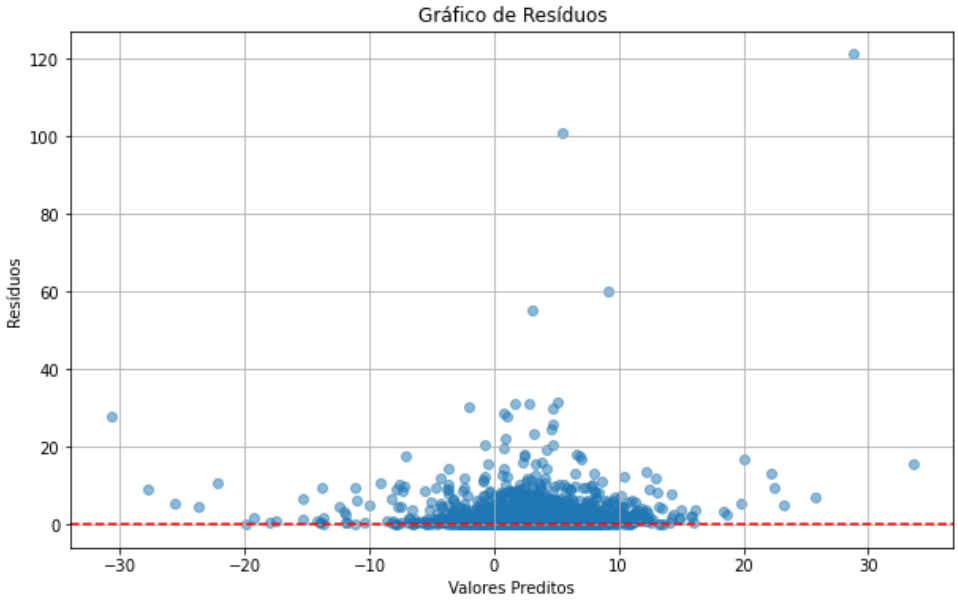

Figura 5.8: Gráfico de resíduos para análise de desempenho do modelo para o indicador Crescimento Anual do PIB (em porcentagem anual), comparando cada valores predito (eixo X) com o respectivo erro ou resíduo (eixo Y).

Portanto, os gráficos da Figura 5.7 e da Figura 5.8 fornecem uma visualização mais aprofundada do desempenho aparente do modelo. Ambas as figuras corroboram com a primeira afirmação da **Hipótese 1** apresentada no Capítulo 1, reforçando a eficácia e a precisão do modelo.

Country Name	Country Code	Region	Year	Real	Predicted	Absolute Error
Equatorial Guinea	GNQ	Sub-Saharan Africa	1997	149.973	28.8084	121.165
Liberia	LBR	Sub-Saharan Africa	1997	106.28	5.44464	100.835
Liberia	LBR	Sub-Saharan Africa	1990	-51.0309	9.11807	60.1489
Timor-Leste	TLS	East Asia & Pacific	2000	58.0781	3.0522	55.0259
Bosnia and Herzegovina	BIH	Europe & Central Asia	1997	36.6411	5.04025	31.6009
Kuwait	KWT	Middle East & North Africa	1993	33.9905	2.78144	31.209
Georgia	GEO	Europe & Central Asia	1993	-29.3002	1.61535	30.9155
Latvia	LVA	Europe & Central Asia	1992	-32.1186	-2.00844	30.1101
Equatorial Guinea	GNQ	Sub-Saharan Africa	1992	34.7453	4.73316	30.0122
Albania	ALB	Europe & Central Asia	1991	-28.0021	0.745369	28.7475
Kiribati	KIR	East Asia & Pacific	1976	-26.7682	1.05889	27.8271
Libya	LBY	Middle East & North Africa	2020	-58.3182	-30.7137	27.6045
Bahrain	BHR	Middle East & North Africa	1976	30.4763	4.7363	25.74
Brunei Darussalam	BRN	East Asia & Pacific	1981	-19.8267	4.55714	24.3839
Lebanon	LBN	Middle East & North Africa	1990	26.5332	3.19749	23.3357
Croatia	HRV	Europe & Central Asia	1991	-21.0887	0.890989	21.9797
Macao SAR, China	MAC	East Asia & Pacific	2010	25.1226	4.64785	20.4747
Estonia	EST	Europe & Central Asia	1992	-21.1687	-0.718842	20.4498

Figura 5.9: *Dataframe* dos resultados com valores reais e preditos para o indicador Crescimento Anual do PIB (em porcentagem anual), ordenado pelo erro absoluto em ordem decrescente (visualizados pela interface do Spyder).

A figura Figura 5.9 mostra explicitamente no conjunto de teste quais são os tais valores *outliers* com os maiores erros absolutos - em qual país e ano foram observados. Analogamente, a Figura 5.10 exibe os menores erros absolutos aferidos pelo modelo no conjunto de teste.

O Erro absoluto médio e o Erro quadrático médio (definidos na Seção 2.5.5) são medidas sobre o erro absoluto de um modelo. Aplicado a este conjunto de teste, a primeira medida fornece um valor menos sensível a valores longe da média (*outliers*), enquanto a segunda é mais sensível a eles. Podemos ver o quanto a presença dos *outliers* elevou o Erro quadrático médio na Figura 5.11. O *Scikit-Learn* fornece funções prontas para o cálculo dessas métricas usando os valores reais e valores preditos como argumento.

Country Name	Country Code	Region	Year	Real	Predicted	Absolute Error
Austria	AUT	Europe & Central Asia	2014	0.661273	0.66134	6.73151e-05
Hong Kong SAR, China	HKG	East Asia & Pacific	1992	6.23487	6.2358	0.000922942
Sub-Saharan Africa	SSF	nan	2007	6.16224	6.16121	0.00102741
Peru	PER	Latin America & Caribbean	2022	2.7255	2.72442	0.00107849
Sub-Saharan Africa	SSF	nan	1991	0.349682	0.351359	0.00167683
Argentina	ARG	Latin America & Caribbean	1999	-3.38546	-3.38729	0.00183082
Bolivia	BOL	Latin America & Caribbean	1990	4.63579	4.63358	0.00220324
Sub-Saharan Africa (IDA &…	TSS	nan	2022	3.66056	3.66303	0.00246819
Europe & Central Asia	ECS	nan	2018	2.13444	2.13183	0.00261486
North America	NAC	nan	2014	2.54945	2.55244	0.00298899
Mauritius	MUS	Sub-Saharan Africa	1992	6.5127	6.51596	0.00326609
Nigeria	NGA	Sub-Saharan Africa	1998	2.58125	2.58453	0.003272
Heavily indebted poor cou…	HPC	nan	1992	-1.12709	-1.12378	0.0033104
Benin	BEN	Sub-Saharan Africa	2022	6.25324	6.24947	0.00377526
Latvia	LVA	Europe & Central Asia	2001	6.32351	6.3194	0.00411082
Iran, Islamic Rep.	IRN	Middle East & North Africa	1990	13.5949	13.5991	0.00417061
Guatemala	GTM	Latin America & Caribbean	2009	0.476898	0.47267	0.00422808
Ghana	GHA	Sub-Saharan Africa	1995	4.11242	4.10809	0.00433042

Figura 5.10: *Dataframe* dos resultados com valores reais e preditos para o indicador Crescimento Anual do PIB (em porcentagem anual), ordenado pelo erro absoluto em ordem crescente (visualizados pela interface do Spyder).

Uma outra funcionalidade analítica fornecida por padrão pelo *Scikit-Learn* e pelo *Pandas* é o método `describe()` dos *dataframes*. Aplicando-o ao *dataframe* dos resultados do modelo sobre o conjunto de teste Figura 5.12, é possível ver que, apesar dos valores mínimos e máximos destoarem muito da média e dos quartis, o desvio padrão dos valores reais do indicador **Crescimento Anual do PIB (em porcentagem anual)**, assim como os erros absolutos, se mantém num valor perto da média, assim como os quartis.

```
In [39]: mae = mean_absolute_error(y_test, y_pred)

In [40]: mae
Out[40]: 1.9072024228086957

In [41]: mse = mean_squared_error(y_test, y_pred)

In [42]: mse
Out[42]: 24.20140384353007
```

Figura 5.11: Cálculo de erro médio absoluto (acima) e do erro médio quadrático (abaixo) aplicados ao conjunto de teste (visualizados pelo console do Spyder).

```
In [27]: results.describe()
Out[27]:
              Year          Real     Predicted  Absolute Error
count  2664.000000  2664.000000  2664.000000     2664.000000
mean   1999.561186     3.492010     3.377752        1.907202
std      13.495358     6.687834     4.052979        4.535604
min    1975.000000   -58.318230   -30.713687        0.000067
25%    1988.000000     1.411606     1.719641        0.261144
50%    2000.000000     3.707275     3.534474        0.769320
75%    2011.000000     5.911942     5.299884        2.071734
max    2022.000000   149.972963    33.613956      121.164609
```

Figura 5.12: Medidas estatísticas dos resultados do teste do modelo, exibindo, para cada uma das colunas: quantidade de registros, média, desvio padrão, valor mínimo, primeiro quartil (ou 25º percentil), mediana (ou 50º percentil), terceiro quartil (ou 75º percentil) e valor máximo, respectivamente. Visualização pelo console do Spyder.

Country Name	Absolute Error
Liberia	17.6641
Equatorial Guinea	16.5314
Timor-Leste	11.382
United Arab Emirates	8.1432
Kuwait	7.6426
Bosnia and Herzegovina	6.92897
Libya	6.73707
Brunei Darussalam	5.438
Kiribati	5.22207
Croatia	5.05981
Lebanon	5.03443
Maldives	4.96438
Syrian Arab Republic	4.95291
Macao SAR, China	4.88484
Turkmenistan	4.77291
Latvia	4.68666
Malta	4.63947
Georgia	4.62174
Mozambique	4.46158
Moldova	4.2403
Vanuatu	4.03786

Figura 5.13: Erros absolutos médios do conjunto de teste agrupados por país ou região, ordenados de forma decrescente (visualizado pela interface do Spyder).

Aprofundando-se na análise do resultado em busca da descoberta de padrões, o conjunto de teste foi agrupado por país ou região para identificar em quais deles os erros absolutos foram maiores e em quais foram menores. Os países com maiores erros absolutos médios (mostrados na Figura 5.13) são, intuitivamente, aqueles cujos valores reais do

indicador **Crescimento Anual do PIB (em porcentagem anual)** destoam mais da média, aumentando o erro da predição.

Já na Figura 5.14, os líderes entre as predições mais precisas são regiões como África Subsaariana, partes da Ásia (especialmente o Sul da Ásia) e América Latina. Essas regiões na base `WDI` já são agrupamentos dos valores médios dos indicadores dos países que fazem parte dessas regiões - tornando os dados mais abundantes e mais próximos da média, o que pode explicar os erros baixos.

Country Name	Absolute Error
Sub-Saharan Africa (IDA & IBRD countries)	0.12504
South Asia (IDA & IBRD)	0.133526
IDA & IBRD total	0.137431
Sub-Saharan Africa	0.145006
South Asia	0.152002
East Asia & Pacific (IDA & IBRD countries)	0.168474
Sub-Saharan Africa (excluding high income)	0.17061
Low & middle income	0.170732
Middle income	0.180638
Europe & Central Asia	0.182229
Latin America & Caribbean	0.188077
Latin America & the Caribbean (IDA & IBRD countries)	0.214478
Finland	0.231411
Indonesia	0.243472
European Union	0.248018
Netherlands	0.25648
Euro area	0.260388
Latin America & Caribbean (excluding high income)	0.265063
Post-demographic dividend	0.266966
France	0.268345
East Asia & Pacific (excluding high income)	0.276037
Germany	0.306829
Austria	0.308352
Upper middle income	0.332602
Denmark	0.36072

Figura 5.14: Erros absolutos médios do conjunto de teste agrupados por país ou região, ordenados de forma crescente (visualizado pela interface do Spyder).

Foi *plotado* também um gráfico mostrando a progressão dos erros absolutos médios das predições agrupados por ano, na Figura 5.15. A curva da progressão dos erros ao longo dos anos de certa forma ressoa com a Figura 4.8, revelando um padrão onde, quanto mais valores ausentes existem para um certo ano, na base de dados, maior será o erro médio das predições.

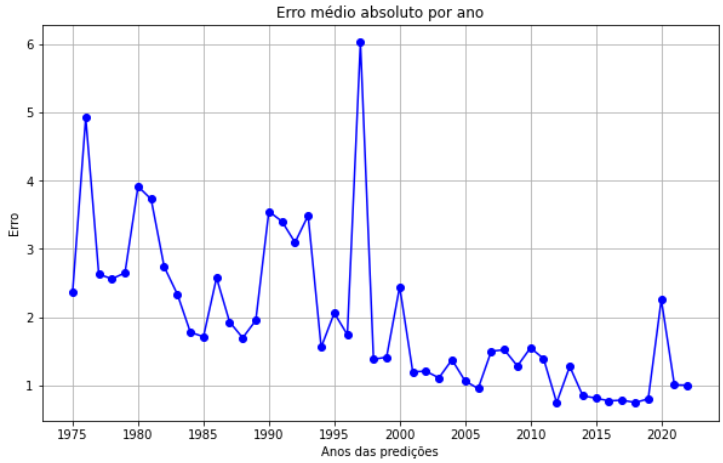

Figura 5.15: Erros absolutos médios do conjunto de teste (eixo Y) agrupados por ano (eixo X).

Por último, é possível analisar o resultado olhando apenas para países específicos, em busca de detectar padrões e obter conhecimento. Na Figura 5.16, são mostrados todos os registros do conjunto de teste cujo país é o Brasil, tendo o valor real e o predito para o **Crescimento Anual do PIB (em porcentagem anual)** brasileiro ao longo das décadas. É possível notar que o crescimento do PIB nacional segue um ritmo constante, sem quedas ou crescimentos muito bruscos, o que contribui para uma precisão maior do modelo preditivo para esse país específico.

Country Name	Country Code	Region	Year ▲	Real	Predicted	Absolute Error
Brazil	BRA	Latin America & Caribbean	1978	4.9699	4.24834	0.721554
Brazil	BRA	Latin America & Caribbean	1982	0.83	0.609325	0.220675
Brazil	BRA	Latin America & Caribbean	1985	7.85	7.15722	0.692784
Brazil	BRA	Latin America & Caribbean	2000	4.38795	3.91911	0.468843
Brazil	BRA	Latin America & Caribbean	2002	3.05346	2.52366	0.5298
Brazil	BRA	Latin America & Caribbean	2006	3.96199	3.84039	0.121594
Brazil	BRA	Latin America & Caribbean	2010	7.52823	7.01011	0.518116
Brazil	BRA	Latin America & Caribbean	2013	3.00482	2.55511	0.449717
Brazil	BRA	Latin America & Caribbean	2020	-3.27676	-3.46803	0.191276
Brazil	BRA	Latin America & Caribbean	2021	4.7626	4.50155	0.261058

Figura 5.16: *Dataframe* dos resultados para o conjunto de teste filtrado para exibir apenas os registros do Brasil (visualizados pela interface do Spyder).

Aspecto Analisado	Resultado Principal
Seleção de Atributos	O algoritmo *SelectKBest* identificou os 32 indicadores mais correlacionados com o crescimento do PIB. Destacam-se os indicadores expressos em crescimento percentual anual e dados demográficos, como a população feminina entre 20 e 24 anos.
Estrutura das Árvores de Decisão	As árvores da *Random Forest* revelaram estruturas profundas, com predominância de indicadores econômicos nos níveis iniciais e repetição de atributos ao longo dos nós.
Desempenho do Modelo (R^2)	O modelo de regressão obteve um coeficiente de determinação de aproximadamente **46%** no conjunto de teste, indicando uma capacidade moderada de explicação da variabilidade dos dados.
Erros Médios por Região	Regiões como América Latina, Sul da Ásia e África Subsaariana apresentaram os menores erros absolutos médios. Países com comportamento econômico volátil apresentaram os maiores erros.
Erros Médios por Ano	Anos com maior incidência de dados ausentes na base apresentaram maior erro médio nas predições, indicando impacto direto da completude dos dados sobre a acurácia. Esse padrão foi recorrente ao longo dos anos, reforçando a influência dos dados ausentes na performance do modelo.
Resultados para o Brasil	As predições para o Brasil mostraram alta precisão, com o modelo conseguindo capturar bem a evolução relativamente estável do PIB nacional.
Impacto de Outliers	A presença de outliers elevou o erro quadrático médio, concentrando os maiores resíduos em registros distantes da média. Alguns registros específicos com erro elevado foram identificados como países em anos com eventos econômicos atípicos.
Estatísticas Descritivas dos Resultados	A análise estatística com `describe()` mostrou que a maioria dos registros possui erros próximos da média e mediana. Isso indica que o modelo é robusto para casos típicos, mesmo com a existência de outliers extremos.

Quadro 5.1: Recapitulação dos Resultados Obtidos.

Os resultados coletados a respeito dos modelos de seleção de atributos e de predição de crescimento do PIB foram expostos neste Capítulo através das várias figuras e gráficos obtidos pela ferramenta *Spyder* e resumidos numa visão única no Quadro 5.1. A partir desses resultados, foram traçadas diversas conclusões a respeito da eficácia dos modelos aplicados. Essas considerações, juntamente com as limitações do estudo e possíveis trabalhos futuros, são abordadas no Capítulo 6.

Capítulo 6

Conclusões

Neste trabalho, foi realizada uma modelagem a partir de algoritmos de regressão com uso de Aprendizado de Máquina (AM), um tipo de tarefa de mineração de dados amplamente empregada em artigos científicos, como apresentado no Capítulo 3. O objetivo geral do trabalho era descobrir padrões úteis sobre o crescimento do Produto Interno Bruto (PIB) a partir da mineração de dados sobre os Indicadores de Desenvolvimento Mundial (WDI) do Banco Mundial.

O algoritmo de Aprendizado de Máquina em particular utilizado foi o de Floresta Aleatória (*Random Forest*). Visando melhorar a qualidade desse modelo, foram empregadas diversas etapas de pré-processamento de dados, como a remoção de dados vazios, a inferência de valores vazios - usando o algoritmo *K-Nearest Neighbors* (KNN) - e, por fim, a seleção de atributos utilizando o algoritmo *SelectKBest*.

Através de gráficos e tabelas, no Capítulo 5 foi possível visualizar os resultados de predições do modelo aplicado sobre dados de teste. Os resultados possibilitaram determinar padrões úteis e relevantes ao problema proposto, como determinar previsões do *Crescimento Anual do PIB (em porcentagem anual)*, representado pelo código `NY.GDP.MKTP.KD.ZG` na base WDI, para diversos países e regiões, com score de aproximadamente 46%.

Também foi possível determinar quais indicadores socioeconômicos são mais relevantes (ou com mais peso) para a atribuição dos valores preditivos numéricos do mesmo indicador.

Este trabalho exemplificou o uso da linguagem *Python* e ferramentas *Pandas*, *Scikit-Learn*, *Spyder* e *Matplotlib* para a mineração de dados. Todos os procedimentos, desde a coleta dos dados à construção do modelo, foram detalhados no Capítulo 4.

A exemplo de alguns dos trabalhos relacionados ao tema, constatou-se que é possível aplicar um algoritmo de mineração de dados em uma base de dados obtida da *web* diretamente em um *script*, sem necessidade de implantação de persistência de dados ou da estruturação de dados em formato relacional para garantir a reprodutibilidade da modelagem.

As diversas classes e métodos presentes no *Scikit-Learn* proveram funcionalidades para a mineração de dados que permitiram realizar todos os objetivos específicos e testar todas as hipóteses, listadas no Capítulo 1.

6.1 Limitações

Uma limitação técnica e semântica sobre esse trabalho é que ele é incapaz de analisar a progressão temporal dos indicadores para cada país e região ao longo dos anos - todos os registros são analisados de forma isolada dos demais, sem o emprego de séries temporais ou nenhuma análise considerando o tempo imediatamente anterior ou imediatamente posterior.

Caso essa abordagem mencionada acima fosse empregada , seria necessário a construção de um modelo de regressão separado para cada um dos países, causando um grande obstáculo com respeito à complexidade e ao tempo de execução dos algoritmos.

A falta da noção de continuidade temporal potencialmente atrapalhou o processo de inferência de valores vazios, pois é sensato pensar que um valor qualquer, uma vez ausente, deveria acompanhar, na média, os valores válidos dos anos mais próximos.

Outro obstáculo técnico importante encontrado no projeto foi a de como utilizar a ferramenta *Matplotlib* para a construção dos gráficos usados no projeto. Em muitos casos, em particular após a instanciação do modelo da Floresta Aleatória, há muitas informações disponíveis sobre o modelo em forma de atributos de classe, mas nenhuma forma predefinida de exibir a configuração da árvore ou da disposição da floresta em forma de gráfico, sendo necessário construí-los de forma manual, como visto na Figura 5.4 e Figura 5.5.

Este projeto está hospedado em um repositório público, facilitando a reprodutibilidade dos resultados, o teste, o compartilhamento e o aprimoramento das técnicas de modelagem empregadas, facilitando futuras consultas e contribuições acadêmicas.

6.2 Trabalhos Futuros

Para trabalhos futuros que explorem o mesmo objeto de estudo, a primeira sugestão é buscar aplicar um modelo que se aproveite de metadados como informações geográficas sobre os países (por exemplo, a região a qual pertencem e sua área) como atributos categóricos, em busca do descobrimento de padrões dentro das regiões e entre países de regiões distintas.

Ainda abordando padrões entre países, seria interessante executar um algoritmo de agrupamento, que figura no ramo dos modelos não-supervisionados, para agrupar países baseados nas suas semelhanças de valores aferidos nos indicadores, e comparar esse agru-

pamento com as fronteiras geográficas dos países, de forma a testar a influência de fatores geopolíticos de países e regiões sobre a evolução dos seus indicadores, em particular o de crescimento do PIB.

Neste trabalho, foram removidos os indicadores diretamente dependentes do PIB (como aqueles que têm como sufixo o `"(% of GDP)"`). Tais indicadores, por definição, são dependentes do valor do Produto Interno Bruto (PIB) e sua taxa de crescimento. Os indicadores foram filtrados da base por possuírem o termo *GDP* no nome da coluna. Uma sugestão para trabalhos futuros relacionados ao tema é aplicar um filtro de indicadores triviais de forma menos manual, utilizando-se de padrões e informações previamente obtidas da própria base de dados e seus valores, o que potencialmente melhoraria os resultados vistos na Seção 5.1.

Uma outra proposta de trabalho futuro é utilizar, para a predição dos indicadores, um outro tipo de algoritmo de regressão, o *Histogram-Based Gradient Boosting Regressor* [55], em complemento ou substituição ao algoritmo usado neste trabalho, o de Florestas Aleatórias. Dessa forma, é possível comparar a performance entre os diversos algoritmos disponíveis. O *Scikit-Learn*, em seu *site*, possui uma página [56] dedicada à comparação direta entre os modelos mencionados acima.

Referências

[1] Acemoglu, Daron e James A. Robison: *Why Nations Fail: The Origins of Power, Prosperity, and Poverty*. Crown Publishing Group, 2012. 1

[2] Paiva, Carlos A. N. e Paulo de Martino Jannuzzi: *Indicadores socioeconômicos e análise regional: fundamentos da centralidade do quociente locacional*. Informe GEPEC, 26(3):378–399, 2022. `https://e-revista.unioeste.br/index.php/gepec/article/view/29569`. 1

[3] Han, Jiawei, Micheline Kamber e Jian Pei: *Data mining: concepts and techniques*. Elsevier, 3ª edição, 2012. 1, 15, 18, 19, 20, 24, 25, 26, 27, 28, 29, 30, 31

[4] World Bank Group. Disponível em: `https://www.worldbank.org/en/home`. Acesso em: 13/07/2024. 1

[5] World Bank Group: Data Catalog. Disponível em: `https://datacatalog.worldbank.org/home`. Acesso em: 10/10/2024. 2

[6] World Bank Group: Sobre as APIs do World Bank. Disponível em: `https://datahelpdesk.worldbank.org/knowledgebase/articles/889386-developer-information-overview`. Acesso em: 10/10/2024. 2

[7] World Bank Group: Data Bank. Disponível em: `https://databank.worldbank.org/home.aspx`. Acesso em: 10/10/2024. 2

[8] World Bank Group: *World Development Indicators*. Disponível em: `https://datatopics.worldbank.org/world-development-indicators/`. Acesso em: 13/07/2024. 2, 41, 42

[9] Acemoglu, Daron, Simon Johnson e James A. Robinson: *The colonial origins of comparative development: An empirical investigation*. American Economic Review, 91(5):1369–1401, 2001. 2

[10] Martin, Xavier Sala-i: *The world distribution of income: Falling poverty and... convergence, period*. Quarterly Journal of Economics, 121(2):351–397, 2006. 2

[11] Samuelson, Paul A. e William D. Nordhaus: *Economics*. McGraw-Hill/Irwin, New York, 18ª edição, 2005. 3, 42

[12] Castro, Leandro N. e Daniel G. Ferrari: *Introdução à mineração de dados: conceitos básicos, algoritmos e aplicações*. Saraiva, 1ª edição, 2016. 6, 7, 9, 12, 15, 16, 18, 19, 20, 23, 28, 29, 30, 31, 32, 33

[13] Laudon, Kenneth C. e Jane P. Laudon: *Sistemas de informação gerenciais*. Pearson Education do Brasil, 11ª edição, 2014. 7, 8

[14] Elmasri, Ramez e Shamkant B. Navathe: *Sistemas de banco de dados*. Pearson Education do Brasil, 6ª edição, 2011. 7, 8, 9, 10, 13

[15] Silberschatz, Abraham, Henry F. Korth e S. Sudarshan: *Database system concepts*. McGraw-Hill, 7ª edição, 2020. 9

[16] United States Geological Survey (USGS): *What are the differences between data, a dataset, and a database?* Disponível em: `https://www.usgs.gov/faqs/what-are-differences-between-data-dataset-and-database`. Acesso em: 19/06/2024. 9

[17] Cavique, Luís: *Big data e data science*. Boletim da APDIO - Associação Portuguesa de Investigação Operacional, 51:11–14, 2014. 9, 28

[18] Inmon, William H., Derek Strauss e Genia Neushloss: *DW 2.0: the Architecture for the Next Generation of Data Warehousing*. Elsevier, 2008. 10

[19] Kimball, Ralph, Margy Ross, Warren Thornthwaite, Joy Mundy e Bob Becker: *The data warehouse lifecycle toolkit*. Wiley, 2ª edição, 2008. 11

[20] Santana, Matheus S. e Ytalo A. S. Carvalho: *Mineração de dados aplicados aos dados públicos do banco mundial*. Monografia (Graduação) de Bacharelado em Ciência da Computação. Universidade de Brasília (UnB), Brasília, Brasil, 2017. 11, 12, 35

[21] Larose, Daniel T. e Chantal D. Larose: *Discovering knowledge in data: an introduction to data mining*. Wiley, 2ª edição, 2014. 12, 13, 14, 17, 18, 26, 32, 33

[22] Dubey, Aditya e Akhtar Rasool: *Data mining based handling missing data*. Em *2019 Third International conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC)*, páginas 483–489, 2019. 18

[23] Geron, Aurélien: *Mãos à Obra: Aprendizado de Máquina com Scikit-Learn & TensorFlow*. Alta Books, 2019. 20, 22, 23, 24, 26, 27, 28

[24] Russell, Stuart e Peter Norvig: *Inteligência Artificial - tradução da 3ª edição*. Elsevier, 2013. 20, 21, 22

[25] Samuel, Arthur L.: *Some studies in machine learning using the game of checkers*. IBM Journal of Research and Development, 3(3):210–229, 1959. 22

[26] Mitchell, Tom M.: *Machine Learning*. McGraw-Hill, 1997. 22, 23

[27] Gilgoldm, CC BY-SA 4.0 <https://creativecommons.org/licenses/by-sa/4.0>, via Wikimedia Commons. *File:File:Decision Tree.jpg*. Disponível em: `https://commons.wikimedia.org/wiki/File:Decision_Tree.jpg`. Acesso em: 27/08/2024. 25

[28] *datasets/titanic.csv*. Disponível em: `https://github.com/datasciencedojo/datasets/blob/master/titanic.csv`. Acesso em: 27/08/2024. 25

[29] Scikit-Learn: Aprendizado supervisionado - Árvores de Decisão. Disponível em: https://scikit-learn.org/stable/modules/tree.html. Acesso em: 20/06/2024. 26

[30] Scikit-Learn: Aprendizado supervisionado - Agregação - Aumento de gradiente, florestas aleatórias, ensacamento, votação, empilhamento. Disponível em: https://scikit-learn.org/stable/modules/ensemble.html#random-forests. Acesso em: 20/06/2024. 27

[31] Bellman, Richard: *Dynamic Programming*. Princeton University Press, 1961. 27

[32] Cichosz, Pawel: *Data Mining Algorithms: Explained Using R*, páginas 558–601. Wiley, 2015. 29, 35

[33] Traskas, Georgios: *Influence of feature selection and pca on a small dataset.* Disponível em: https://gtraskas.github.io/post/titanic_prediction/. Acesso em: 19/06/2024. 29

[34] Biblioteca Digital da Sociedade Brasileira de Computação. Disponível em: https://sol.sbc.org.br/index.php/indice. Acesso em: 28/08/2024. 34

[35] Arif, Hera e Jauhar Ali: *Using data mining techniques on world bank statistics*. Em *2018 International Conference on Advances in Computing and Communication Engineering (ICACCE)*, páginas 439–444, 2018. 34

[36] Gamberger, Dragan, Dražen Lučanin e Tomislav Šmuc: *Analysis of world bank indicators for countries with banking crises by subgroup discovery induction*. Em *2013 36th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, páginas 1138–1142, 2013. 34

[37] Ahammad, Md. Saymon, Sadia Akter Sinthia, Mahjabeen Hossain, Md. Mustak Ahmed, Md. Nurul Afsar Ikram e Nur A All Asif: *Machine learning for gdp forecasting: Enhancing economic projections in bangladesh*. Em *2024 15th International Conference on Computing Communication and Networking Technologies (ICCCNT)*, páginas 1–5, 2024. 34

[38] Henrique, Daniel Christian, Ivan Aune de Aguiar Filho, João Carlos Prats Ramos e Gabriel Dudena de Faria: *Pobreza extrema e a covid-19 no mundo: Um estudo com abordagens de inteligência artificial*. Miscellaneous, 20(11):156–179, 2023. 35

[39] Rodrigues, Eduardo. The Scikit-HEP Project - Scientific Figure on ResearchGate. Disponível em: https://www.researchgate.net/figure/Schematic-view-of-the-Python-scientific-software-ecosystem-Figure-taken-from-Jake_fig1_332799309. Acesso em: 27/08/2024]. 39

[40] Git. Disponível em: https://git-scm.com/. Acesso em: 20/02/2025. 39

[41] Python. Disponível em: https://docs.python.org/3.11/. Acesso em: 28/08/2024. 39

[42] Spyder. Disponível em: https://docs.spyder-ide.org/current/index.html. Acesso em: 28/08/2024. 40

[43] Pandas. Disponível em: https://pandas.pydata.org/docs/index.html. Acesso em: 28/08/2024. 40

[44] Scikit-Learn. Disponível em: https://scikit-learn.org/stable/index.html. Acesso em: 28/08/2024. 40

[45] Matplotlib. Disponível em: https://matplotlib.org/stable/index.html. Acesso em: 09/09/2024. 40

[46] World Bank Group: *Base de dados World Development Indicators*. Versão 28/06/2024. Documento eletrônico disponível em: https://datacatalogfiles.worldbank.org/ddh-published/0037712/DR0045575/WDI_CSV_2024_06_28.zip?versionId=2024-07-01T13:30:38.4396512Z. Acesso em: 13/07/2024. 42

[47] Pandas: Reshaping and pivot tables. Disponível em: https://pandas.pydata.org/docs/user_guide/reshaping.html. Acesso em: 16/10/2024. 45

[48] Juszczuk, Przemysław, Jan Kozak, Grzegorz Dziczkowski, Szymon Głowania, Tomasz Jach e Barbara Probierz: *Real-world data difficulty estimation with the use of entropy*. Entropy, 23(12), 2021, ISSN 1099-4300. https://www.mdpi.com/1099-4300/23/12/1621. 47

[49] Zaidi, Houda, Faouzi Boufarès e Yann Pollet: *Improve data quality by processing null values and semantic dependencies*. Journal of Computer and Communications, 04:78–85, janeiro 2016. 48

[50] Scikit-Learn: Nearest neighbors imputation. Disponível em: https://scikit-learn.org/stable/modules/impute.html#nearest-neighbors-imputation. Acesso em: 07/08/2024. 53

[51] Murti, Della Murbarani Prawidya, Utomo Pujianto, Aji Prasetya Wibawa e Muhammad Iqbal Akbar: *K-nearest neighbor (k-nn) based missing data imputation*. Em *2019 5th International Conference on Science in Information Technology (ICSITech)*, páginas 83–88, 2019. 53

[52] Scikit-Learn: Train-Test Split. Disponível em: https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.train_test_split.html. Acesso em: 07/08/2024. 54

[53] Scikit-Learn: Feature selection. Disponível em: https://scikit-learn.org/stable/modules/feature_selection.html. Acesso em: 07/08/2024. 54

[54] Scikit-Learn: Random Forest Regressor. Disponível em: https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestRegressor.html. Acesso em: 08/08/2024. 55, 56, 58

[55] Scikit-Learn: Histogram-Based Gradient Boosting Regressor. Disponível em: https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.HistGradientBoostingRegressor.html. Acesso em: 14/02/2025. 72

[56] Scikit-Learn: Comparing Random Forests and Histogram Gradient Boosting models. Disponível em: https://scikit-learn.org/stable/auto_examples/ensemble/plot_forest_hist_grad_boosting_comparison.html#sphx-glr-auto-examples-ensemble-plot-forest-hist-grad-boosting-comparison-py. Acesso em: 08/08/2024. 72

Apêndice A

Script para obter e tratar a base original

```
""" Passo 1: extrai a base de dados de sua pasta original e transforma
    sua estrutura.
"""
import pandas as pd

EXTRACTS_PATH = '../Data/WDI_CSV_2024_06_28/'
RAW_FILENAME = 'WDICSV.csv'

DATAFRAMES_PATH = './dataframes/'
PREPROCESSED_DF_PATH = DATAFRAMES_PATH + 'WDItransformada.csv'

def get_wdi_dataframe():
    """
    Lê os dados extraídos em csv e o transforma em um dataframe Pandas,
    além de realizar o tratamento com melt+pivot para
    mover os anos para representação em linhas
    e os atributos (indicadores) para representação em colunas.
    """
    return pd \
        .read_csv(
            EXTRACTS_PATH + RAW_FILENAME,
            usecols= lambda col: col!='Indicator Name') \
        .melt(
            id_vars=['Country Name', 'Country Code', 'Indicator Code'],
            var_name='Year',
            value_name = 'Value') \
        .pivot(
            index=['Country Name', 'Country Code', 'Year'],
            columns='Indicator Code',
```

```
29              values='Value') \
30          .reset_index()
31
32
33 ### Salva dataframe resultante em arquivo csv
34
35 data = get_wdi_dataframe()
36 data.to_csv(PREPROCESSED_DF_PATH, index=False)
```

gera_dataframes.py

Apêndice B

Script para o pré-processamento dos dados

```
""" Passo 2: usa o framework Pandas para o pré-processamento dos dados.
"""
import pandas as pd

### Constantes de ambiente

DATAFRAMES_PATH = './dataframes/'
TABLES_PATH = './material_overleaf/tabelas/'

TRANSFORMED_DF_PATH = DATAFRAMES_PATH + 'WDItransformada.csv'
PREPROCESSED_DF_PATH = DATAFRAMES_PATH + 'WDIPreProcessada.csv'
RAW_DF_PATH = DATAFRAMES_PATH + 'WDICSV.csv'
COUNTRIES_PATH = DATAFRAMES_PATH + 'WDICountry.csv'
INDICATORS_PATH = DATAFRAMES_PATH + 'WDISeries.csv'
INDICATORS_FULLINFO_PATH = TABLES_PATH + 'indicadoresfull.csv'


### Parâmetros

YEARS_TO_DROP = 16                  # 1/4 do total de anos
COUNTRIES_TO_DROP = 28              # aprox. 10% dos 266 países e regiões
INDICATORS_NOT_NAN_THRESHOLD = 0.6  # exclui todo indicador com + de 40% valores nulos


### Extração dos dados

wdi = pd.read_csv(TRANSFORMED_DF_PATH)
```

```
raw_wdi = pd.read_csv(RAW_DF_PATH)
countries = pd.read_csv(COUNTRIES_PATH, index_col='Country Code')
indicators = pd.read_csv(INDICATORS_PATH, index_col='Series Code')

# Cria tabela de todos os indicadores
indicators['Indicator Name'] = indicators['Indicator Name'].map(lambda x
    : x if len(x) <= 49 else x[:49] + '...')
indicators.sort_values(['Topic']).to_csv(
    TABLES_PATH + 'indicadoresfull.csv',
    columns = ['Indicator Name'])


### Variáveis para análise sobre o dataset inicial

total_indicators = len(wdi.columns) - 3
total_countries = len(wdi.groupby('Country Code'))
total_years = len(wdi.groupby('Year'))
total_nan = wdi.isna().sum().sum()
total_values = total_countries * total_indicators * total_years

# Dataframe que mostra a qtd. de valores vazios para cada indicador
nan_per_indicator = wdi.isna().sum()[3:] \
    .to_frame().rename(columns={0: 'NaN values'})
nan_per_indicator.insert(0, 'Name', indicators['Indicator Name'])
nan_per_indicator.insert(2, 'Percentage', (nan_per_indicator['NaN values
    '] / len(wdi) * 100).round(2))

# Série que mostra a qtd. de valores vazios por ano, somando todos os pa
    íses e indicadores
nan_per_year = raw_wdi.isna().sum()[4:]

# Série que mostra a qtd. de valores vazios para cada país, somando
    todos os anos
nan_per_country = wdi.groupby(['Country Code', 'Country Name']) \
    .count() \
    .drop(columns='Year') \
    .sum(axis=1) \
    .apply(lambda x: total_indicators * total_years - x) \
    .to_frame('NaN values') \
    .reset_index()


### Pré-processamento

emptiest_indicators = nan_per_indicator.nlargest(50, 'NaN values')
```

```
emptiest_years = nan_per_year.nlargest(YEARS_TO_DROP)
emptiest_countries = nan_per_country.nlargest(COUNTRIES_TO_DROP, 'NaN values')


# Exlui registros que possuem a variável "crescimento do PIB" (o alvo do modelo) vazia
[gdp_growth_code] = indicators.query("`Indicator Name` == 'GDP growth (annual %)'").index
wdi = wdi.dropna(subset=[gdp_growth_code])

# Remove os anos que possuem mais valores vazios, conforme parâmetro
wdi = wdi[~wdi['Year'].isin(emptiest_years.index.astype(int))]

# Remove os países que possuem mais valores vazios, conforme parâmetro
wdi = wdi[~wdi['Country Code'].isin(emptiest_countries['Country Code'])]

# Mantém apenas indicadores que possuem uma porcentagem de valores não-nulos, conforme parâmetro
wdi = wdi.dropna(axis=1, thresh=INDICATORS_NOT_NAN_THRESHOLD*len(wdi))

# Cria tabela dos indicadores que passaram no filtro acima (apêndice D)
filtered_indicators = indicators[indicators.index.isin(wdi.columns)]
filtered_indicators.sort_values(['Series Code']).to_csv(
    TABLES_PATH + 'indicadoresFiltro.csv',
    columns = ['Indicator Name'])


### Salva dataframe resultante do pré-processamento em arquivo csv

wdi.to_csv(PREPROCESSED_DF_PATH, index=False)


### Criação de gráficos e tabelas para análise sobre o dataset inicial
# obs: executar um plot por vez

nan_per_indicator['NaN values'].plot.hist(
    xlabel='Qtd. de valores nulos',
    ylabel='Frequência de indicadores',
    bins=20,
    grid=True)

nan_per_year.plot(
    xlabel='Ano',
    ylabel='Valores nulos',
```

```
110     ylim=(0, 400000),
111     grid=True)
```

pre_processamento.py

Apêndice C

Script da construção do modelo com *Scikit-Learn*

```
""" Passo 3: realiza a modelagem com Scikit-Learn a partir da base de
    dados tratada nos passos anteriores.
"""
import pandas as pd
import matplotlib.pyplot as plt

from sklearn.feature_selection import SelectKBest, r_regression
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestRegressor
from sklearn.impute import KNNImputer
from sklearn.tree import plot_tree
from sklearn.metrics import mean_absolute_error, mean_squared_error
from scipy.stats import linregress


### Constantes de ambiente

DATAFRAMES_PATH = './dataframes/'
TABLES_PATH = './material_overleaf/tabelas/'

TRANSFORMED_DF_PATH = DATAFRAMES_PATH + 'WDItransformada.csv'
PREPROCESSED_DF_PATH = DATAFRAMES_PATH + 'WDIPreProcessada.csv'
RAW_DF_PATH = DATAFRAMES_PATH + 'WDICSV.csv'
COUNTRIES_PATH = DATAFRAMES_PATH + 'WDICountry.csv'
INDICATORS_PATH = DATAFRAMES_PATH + 'WDISeries.csv'


### Parâmetros

```

```
KNN_IMPUTER_NEIGHBOURS = 10
TEST_SET_RATIO = 0.25
FEATURES_TO_SELECT = 32


### Extração dos dados

wdi = pd.read_csv(PREPROCESSED_DF_PATH)
transformed_wdi = pd.read_csv(TRANSFORMED_DF_PATH)
raw_wdi = pd.read_csv(RAW_DF_PATH)
countries = pd.read_csv(COUNTRIES_PATH, index_col='Country Code')
indicators = pd.read_csv(INDICATORS_PATH, index_col='Series Code')


### Separa as variáveis de entrada (X) e variável alvo (y)

[gdp_growth_code] = indicators.query("`Indicator Name` == 'GDP growth (
    annual %)'").index
wdi = wdi.set_index(['Country Name', 'Country Code', 'Year'])
X = wdi.drop(columns=[gdp_growth_code])
y = wdi[gdp_growth_code]

### Normaliza o conjunto de entrada
# scaler = StandardScaler()
# X_scaled = scaler.fit_transform(X, y)


### Preenche os valores vazios no conjunto de entrada por inferência

imputer = KNNImputer(n_neighbors=KNN_IMPUTER_NEIGHBOURS, weights='
    uniform')
X_imputed = imputer.fit_transform(X)
X_imputed = pd.DataFrame(X_imputed, columns=X.columns, index=X.index)


### Remove indicadores triviais

# Usa critério: indicadores que contém "growth" ("crescimento") no nome
# trivial_indicators = indicators[indicators['Indicator Name'].str.
    contains('growth')]

# Usando critério: indicadores que contém "GDP" ("PIB") no código
# trivial_indicators = indicators[indicators['Indicator Name'].str.
    contains('GDP')]

```

```
trivial_indicators = indicators[
#    indicators['Indicator Name'].str.contains('growth') |
    indicators['Indicator Name'].str.contains('GDP')]

X_minus_trivials = X_imputed.drop(
    columns=[c for c in trivial_indicators.index if c in X_imputed.
    columns])


### Separa em conjuntos de teste e treinamento

X_train, X_test, y_train, y_test = train_test_split(
    X_minus_trivials, y, test_size=TEST_SET_RATIO, random_state=200)


### Seleciona os melhores indicadores, conforme parâmetro


feature_selector = SelectKBest(r_regression, k=FEATURES_TO_SELECT)
feature_selector.fit(X_train, y_train)

X_train_selected = pd.DataFrame(
    feature_selector.transform(X_train),
    columns = X_train.columns[feature_selector.get_support()],
    index = X_train.index)

X_test_selected = pd.DataFrame(
    feature_selector.transform(X_test),
    columns = X_test.columns[feature_selector.get_support()],
    index = X_test.index)




# Cria tabela dos melhores indicadores selecionados
selector_scores = pd.DataFrame(zip(X_train.columns, feature_selector.
    scores_)).set_index(0)
indicators['Score'] = selector_scores

selected_indicators = indicators[indicators.index.isin(X_train_selected.
    columns)][[
    'Topic', 'Indicator Name', 'Score']]


### APLICAÇÃO DOS MODELOS
```

```
112
113 random_forest = RandomForestRegressor(random_state=0)
114 random_forest.fit(X_train_selected, y_train)
115
116
117 ### AFERIÇÃO DO DESEMPENHO
118
119 y_pred = random_forest.predict(X_test_selected)
120 score = random_forest.score(X_test_selected, y_test)
121 mae = mean_absolute_error(y_test, y_pred)
122 mse = mean_squared_error(y_test, y_pred)
123
124
125
126 ### Criação do dataframe para análise sobre o resultado
127
128 results = y_test.reset_index()
129 results = results.rename(columns={gdp_growth_code: 'Real'})
130 results['Predicted'] = y_pred
131 results['Absolute Error'] = abs(results['Real'] - results['Predicted'])
132
133 results = results.join(countries[['Region']], on='Country Code')
134 results.insert(2, 'Region', results.pop('Region'))
135
136
137
138
139 ### Criação de gráficos para análise sobre o resultado
140
141 ## Tabela de indicadores triviais removidos
142
143 trivial_indicators[['Topic', 'Indicator Name']].to_csv(
144     TABLES_PATH + 'indicadoresRetirados.csv')
145
146 ## Tabelas ilustrando dos erros absolutos
147
148 results_per_country = results.groupby('Country Name')['Absolute Error'].
    mean()
149
150 # results_per_region = results.groupby('Region')['Absolute Error'].mean
    ()
151 results_per_region = results[results['Region'].isna()].groupby('Country
    Name')['Absolute Error'].mean()
152
153 results_brazil = results[results['Country Code'] =='BRA']
```

```
154
155 results_per_year = results.groupby('Year')['Absolute Error'].mean()
156 plt.figure(figsize=(10, 6))
157 plt.xlabel('Anos das predições')
158 plt.ylabel('Erro')
159 plt.title('Erro médio absoluto por ano')
160 plt.grid(True)
161 plt.plot(
162     results_per_year.index,
163     results_per_year.values,
164     marker='o',
165     linestyle='-',
166     color='b')
167 plt.xticks([y for y in results_per_year.index if y%5==0])
168 plt.show()
169
170
171
172 ## Calcula a previsão sobre os dados de teste
173
174
175
176 # Cria um gráfico de disperção
177
178 linear_regression = linregress(y_test, y_pred)
179 plt.scatter(y_test, y_pred, alpha=0.5)
180 plt.axis('equal')
181 plt.plot(
182     [-50, 60],
183     [-50, 60],
184     color='g',
185     label='Reta ideal (com erro zero)')
186 plt.plot(
187     y_test,
188     linear_regression.intercept + linear_regression.slope*y_test,
189     color='r',
190     label='Regressão linear')
191 plt.legend()
192 plt.xlabel('Valores Reais')
193 plt.ylabel('Valores Preditos')
194 plt.title('Valores Reais vs. Valores Preditos')
195 plt.grid(True)
196 plt.show()
197
```

```
198 ## Calcula a diferença entre os valores reais x valores preditos (
    residuos)
199
200 residuals = abs(y_test - y_pred)
201
202 # Cria um gráfico de resíduos
203 plt.figure(figsize=(10, 6))
204 plt.scatter(y_pred, residuals, alpha=0.5)
205 plt.axhline(y=0, color='r', linestyle='--')
206 plt.xlabel('Valores Preditos')
207 plt.ylabel('Resíduos')
208 plt.title('Gráfico de Resíduos')
209 plt.grid(True)
210 plt.show()
211
212 ## Extrai uma árvore de decisão individual do modelo
213
214 tree0 = random_forest.estimators_[0]
215
216 ## Plota árvore no tamanho original (centenas de nós, ilegível)
217 plt.figure(figsize=(40,30))
218 plot_tree(
219     tree0,
220     feature_names=[indicators['Indicator Name'][i] for i in
    X_train_selected.columns],
221     filled=True)
222 plt.show()
223
224 ## Plota árvore limitando a profundidade para legibilidade
225 plt.figure(figsize=(40,30))
226 plot_tree(
227     tree0,
228     max_depth=2,
229     feature_names=[indicators['Indicator Name'][i] for i in
    X_train_selected.columns],
230     filled=True,
231     fontsize=15)
232 plt.show()
```

modelo.py

Apêndice D

Lista de indicadores socioeconômicos da base de dados WDI (em inglês)

Series Code	Indicator Name
AG.CON.FERT.ZS	Fertilizer consumption (kilograms per hectare of ...
AG.LND.AGRI.K2	Agricultural land (sq. km)
AG.LND.AGRI.ZS	Agricultural land (% of land area)
AG.LND.ARBL.HA	Arable land (hectares)
AG.LND.ARBL.HA.PC	Arable land (hectares per person)
AG.LND.ARBL.ZS	Arable land (% of land area)
AG.LND.CREL.HA	Land under cereal production (hectares)
AG.LND.CROP.ZS	Permanent cropland (% of land area)
AG.LND.FRST.K2	Forest area (sq. km)
AG.LND.FRST.ZS	Forest area (% of land area)
AG.LND.PRCP.MM	Average precipitation in depth (mm per year)
AG.LND.TOTL.K2	Land area (sq. km)
AG.PRD.CREL.MT	Cereal production (metric tons)
AG.PRD.CROP.XD	Crop production index (2014-2016 = 100)

AG.PRD.FOOD.XD	Food production index (2014-2016 = 100)
AG.PRD.LVSK.XD	Livestock production index (2014-2016 = 100)
AG.SRF.TOTL.K2	Surface area (sq. km)
AG.YLD.CREL.KG	Cereal yield (kg per hectare)
BG.GSR.NFSV.GD.ZS	Trade in services (% of GDP)
BM.GSR.CMCP.ZS	Communications, computer, etc. (% of service impo...
BM.GSR.FCTY.CD	Primary income payments (BoP, current US$)
BM.GSR.GNFS.CD	Imports of goods and services (BoP, current US$)
BM.GSR.INSF.ZS	Insurance and financial services (% of service im...
BM.GSR.MRCH.CD	Goods imports (BoP, current US$)
BM.GSR.NFSV.CD	Service imports (BoP, current US$)
BM.GSR.ROYL.CD	Charges for the use of intellectual property, pay...
BM.GSR.TOTL.CD	Imports of goods, services and primary income (Bo...
BM.GSR.TRAN.ZS	Transport services (% of service imports, BoP)
BM.GSR.TRVL.ZS	Travel services (% of service imports, BoP)
BM.KLT.DINV.CD.WD	Foreign direct investment, net outflows (BoP, cur...
BM.KLT.DINV.WD.GD.ZS	Foreign direct investment, net outflows (% of GDP...
BM.TRF.PRVT.CD	Secondary income, other sectors, payments (BoP, c...
BM.TRF.PWKR.CD.DT	Personal remittances, paid (current US$)
BN.CAB.XOKA.CD	Current account balance (BoP, current US$)
BN.CAB.XOKA.GD.ZS	Current account balance (% of GDP)
BN.FIN.TOTL.CD	Net financial account (BoP, current US$)
BN.GSR.FCTY.CD	Net primary income (BoP, current US$)
BN.GSR.GNFS.CD	Net trade in goods and services (BoP, current US$...

BN.GSR.MRCH.CD	Net trade in goods (BoP, current US$)
BN.KAC.EOMS.CD	Net errors and omissions (BoP, current US$)
BN.KLT.DINV.CD	Foreign direct investment, net (BoP, current US$)
BN.RES.INCL.CD	Reserves and related items (BoP, current US$)
BN.TRF.CURR.CD	Net secondary income (BoP, current US$)
BX.GRT.EXTA.CD.WD	Grants, excluding technical cooperation (BoP, cur...
BX.GRT.TECH.CD.WD	Technical cooperation grants (BoP, current US$)
BX.GSR.CMCP.ZS	Communications, computer, etc. (% of service expo...
BX.GSR.FCTY.CD	Primary income receipts (BoP, current US$)
BX.GSR.GNFS.CD	Exports of goods and services (BoP, current US$)
BX.GSR.INSF.ZS	Insurance and financial services (% of service ex...
BX.GSR.MRCH.CD	Goods exports (BoP, current US$)
BX.GSR.NFSV.CD	Service exports (BoP, current US$)
BX.GSR.TOTL.CD	Exports of goods, services and primary income (Bo...
BX.GSR.TRAN.ZS	Transport services (% of service exports, BoP)
BX.GSR.TRVL.ZS	Travel services (% of service exports, BoP)
BX.KLT.DINV.CD.WD	Foreign direct investment, net inflows (BoP, curr...
BX.KLT.DINV.WD.GD.ZS	Foreign direct investment, net inflows (% of GDP)
BX.PEF.TOTL.CD.WD	Portfolio equity, net inflows (BoP, current US$)
BX.TRF.CURR.CD	Secondary income receipts (BoP, current US$)
BX.TRF.PWKR.CD.DT	Personal remittances, received (current US$)
BX.TRF.PWKR.DT.GD.ZS	Personal remittances, received (% of GDP)
DC.DAC.AUTL.CD	Net bilateral aid flows from DAC donors, Austria ...
DC.DAC.BELL.CD	Net bilateral aid flows from DAC donors, Belgium ...

DC.DAC.CANL.CD	Net bilateral aid flows from DAC donors, Canada (...
DC.DAC.CECL.CD	Net bilateral aid flows from DAC donors, European...
DC.DAC.CHEL.CD	Net bilateral aid flows from DAC donors, Switzerl...
DC.DAC.DEUL.CD	Net bilateral aid flows from DAC donors, Germany ...
DC.DAC.FRAL.CD	Net bilateral aid flows from DAC donors, France (...
DC.DAC.GBRL.CD	Net bilateral aid flows from DAC donors, United K...
DC.DAC.ITAL.CD	Net bilateral aid flows from DAC donors, Italy (c...
DC.DAC.JPNL.CD	Net bilateral aid flows from DAC donors, Japan (c...
DC.DAC.NLDL.CD	Net bilateral aid flows from DAC donors, Netherla...
DC.DAC.NORL.CD	Net bilateral aid flows from DAC donors, Norway (...
DC.DAC.SWEL.CD	Net bilateral aid flows from DAC donors, Sweden (...
DC.DAC.TOTL.CD	Net bilateral aid flows from DAC donors, Total (c...
DC.DAC.USAL.CD	Net bilateral aid flows from DAC donors, United S...
DT.NFL.NIFC.CD	IFC, private nonguaranteed (NFL, current US$)
DT.NFL.UNCF.CD	Net official flows from UN agencies, UNICEF (curr...
DT.NFL.UNDP.CD	Net official flows from UN agencies, UNDP (curren...
DT.NFL.UNFP.CD	Net official flows from UN agencies, UNFPA (curre...
DT.ODA.ALLD.CD	Net official development assistance and official ...
DT.ODA.ALLD.KD	Net official development assistance and official ...
DT.ODA.ODAT.CD	Net official development assistance received (cur...
DT.ODA.ODAT.GI.ZS	Net ODA received (% of gross capital formation)
DT.ODA.ODAT.GN.ZS	Net ODA received (% of GNI)
DT.ODA.ODAT.KD	Net official development assistance received (con...
DT.ODA.ODAT.MP.ZS	Net ODA received (% of imports of goods, services...

DT.ODA.ODAT.PC.ZS	Net ODA received per capita (current US$)
EG.ELC.ACCS.RU.ZS	Access to electricity, rural (% of rural populati...
EG.ELC.ACCS.UR.ZS	Access to electricity, urban (% of urban populati...
EG.ELC.ACCS.ZS	Access to electricity (% of population)
EG.ELC.COAL.ZS	Electricity production from coal sources (% of to...
EG.ELC.FOSL.ZS	Electricity production from oil, gas and coal sou...
EG.ELC.HYRO.ZS	Electricity production from hydroelectric sources...
EG.ELC.LOSS.ZS	Electric power transmission and distribution loss...
EG.ELC.NGAS.ZS	Electricity production from natural gas sources (...
EG.ELC.NUCL.ZS	Electricity production from nuclear sources (% of...
EG.ELC.PETR.ZS	Electricity production from oil sources (% of tot...
EG.ELC.RNWX.KH	Electricity production from renewable sources, ex...
EG.ELC.RNWX.ZS	Electricity production from renewable sources, ex...
EG.FEC.RNEW.ZS	Renewable energy consumption (% of total final en...
EG.IMP.CONS.ZS	Energy imports, net (% of energy use)
EG.USE.COMM.CL.ZS	Alternative and nuclear energy (% of total energy...
EG.USE.COMM.FO.ZS	Fossil fuel energy consumption (% of total)
EG.USE.CRNW.ZS	Combustible renewables and waste (% of total ener...
EG.USE.ELEC.KH.PC	Electric power consumption (kWh per capita)
EG.USE.PCAP.KG.OE	Energy use (kg of oil equivalent per capita)
EN.ATM.CO2E.EG.ZS	CO2 intensity (kg per kg of oil equivalent energy...
EN.ATM.CO2E.GF.KT	CO2 emissions from gaseous fuel consumption (kt)
EN.ATM.CO2E.GF.ZS	CO2 emissions from gaseous fuel consumption (% of...
EN.ATM.CO2E.KD.GD	CO2 emissions (kg per 2015 US$ of GDP)

EN.ATM.CO2E.KT	CO2 emissions (kt)
EN.ATM.CO2E.LF.KT	CO2 emissions from liquid fuel consumption (kt)
EN.ATM.CO2E.LF.ZS	CO2 emissions from liquid fuel consumption (% of ...
EN.ATM.CO2E.PC	CO2 emissions (metric tons per capita)
EN.ATM.CO2E.PP.GD	CO2 emissions (kg per PPP $ of GDP)
EN.ATM.CO2E.PP.GD.KD	CO2 emissions (kg per 2021 PPP $ of GDP)
EN.ATM.CO2E.SF.KT	CO2 emissions from solid fuel consumption (kt)
EN.ATM.CO2E.SF.ZS	CO2 emissions from solid fuel consumption (% of t...
EN.ATM.GHGO.KT.CE	Other greenhouse gas emissions, HFC, PFC and SF6 ...
EN.ATM.GHGT.KT.CE	Total greenhouse gas emissions (kt of CO2 equival...
EN.ATM.METH.AG.KT.CE	Agricultural methane emissions (thousand metric t...
EN.ATM.METH.AG.ZS	Agricultural methane emissions (% of total)
EN.ATM.METH.EG.KT.CE	Methane emissions in energy sector (thousand metr...
EN.ATM.METH.EG.ZS	Energy related methane emissions (% of total)
EN.ATM.METH.KT.CE	Methane emissions (kt of CO2 equivalent)
EN.ATM.NOXE.AG.KT.CE	Agricultural nitrous oxide emissions (thousand me...
EN.ATM.NOXE.AG.ZS	Agricultural nitrous oxide emissions (% of total)
EN.ATM.NOXE.EG.KT.CE	Nitrous oxide emissions in energy sector (thousan...
EN.ATM.NOXE.EG.ZS	Nitrous oxide emissions in energy sector (% of to...
EN.ATM.NOXE.KT.CE	Nitrous oxide emissions (thousand metric tons of ...
EN.CO2.BLDG.ZS	CO2 emissions from residential buildings and comm...
EN.CO2.ETOT.ZS	CO2 emissions from electricity and heat productio...
EN.CO2.MANF.ZS	CO2 emissions from manufacturing industries and c...
EN.CO2.OTHX.ZS	CO2 emissions from other sectors, excluding resid...

EN.CO2.TRAN.ZS	CO2 emissions from transport (% of total fuel com...
EN.POP.DNST	Population density (people per sq. km of land are...
EN.URB.LCTY	Population in largest city
EN.URB.LCTY.UR.ZS	Population in the largest city (% of urban popula...
EN.URB.MCTY.TL.ZS	Population in urban agglomerations of more than 1...
ER.FSH.AQUA.MT	Aquaculture production (metric tons)
ER.FSH.CAPT.MT	Capture fisheries production (metric tons)
ER.FSH.PROD.MT	Total fisheries production (metric tons)
ER.GDP.FWTL.M3.KD	Water productivity, total (constant 2015 US$ GDP ...
ER.H2O.INTR.K3	Renewable internal freshwater resources, total (b...
ER.H2O.INTR.PC	Renewable internal freshwater resources per capit...
FD.AST.PRVT.GD.ZS	Domestic credit to private sector by banks (% of ...
FI.RES.TOTL.CD	Total reserves (includes gold, current US$)
FI.RES.TOTL.MO	Total reserves in months of imports
FI.RES.XGLD.CD	Total reserves minus gold (current US$)
FM.AST.CGOV.ZG.M3	Claims on central government (annual growth as % ...
FM.AST.DOMS.CN	Net domestic credit (current LCU)
FM.AST.NFRG.CN	Net foreign assets (current LCU)
FM.AST.PRVT.GD.ZS	Monetary Sector credit to private sector (% GDP)
FM.AST.PRVT.ZG.M3	Claims on private sector (annual growth as % of b...
FM.LBL.BMNY.CN	Broad money (current LCU)
FM.LBL.BMNY.GD.ZS	Broad money (% of GDP)
FM.LBL.BMNY.ZG	Broad money growth (annual %)
FP.CPI.TOTL	Consumer price index (2010 = 100)

FP.CPI.TOTL.ZG	Inflation, consumer prices (annual %)
FS.AST.CGOV.GD.ZS	Claims on central government, etc. (% GDP)
FS.AST.PRVT.GD.ZS	Domestic credit to private sector (% of GDP)
IS.AIR.DPRT	Air transport, registered carrier departures worl...
IS.AIR.GOOD.MT.K1	Air transport, freight (million ton-km)
IS.AIR.PSGR	Air transport, passengers carried
IT.CEL.SETS	Mobile cellular subscriptions
IT.CEL.SETS.P2	Mobile cellular subscriptions (per 100 people)
IT.MLT.MAIN	Fixed telephone subscriptions
IT.MLT.MAIN.P2	Fixed telephone subscriptions (per 100 people)
IT.NET.USER.ZS	Individuals using the Internet (% of population)
MS.MIL.MPRT.KD	Arms imports (SIPRI trend indicator values)
MS.MIL.TOTL.P1	Armed forces personnel, total
MS.MIL.XPND.CD	Military expenditure (current USD)
MS.MIL.XPND.CN	Military expenditure (current LCU)
MS.MIL.XPND.GD.ZS	Military expenditure (% of GDP)
NE.CON.GOVT.CD	General government final consumption expenditure ...
NE.CON.GOVT.CN	General government final consumption expenditure ...
NE.CON.GOVT.KD	General government final consumption expenditure ...
NE.CON.GOVT.KD.ZG	General government final consumption expenditure ...
NE.CON.GOVT.ZS	General government final consumption expenditure ...
NE.CON.PRVT.CD	Household and NPISHs Final consumption expenditur...
NE.CON.PRVT.CN	Household and NPISHs Final consumption expenditur...
NE.CON.PRVT.KD	Households and NPISHs Final consumption expenditu...

NE.CON.PRVT.KD.ZG	Household and NPISHs Final consumption expenditur...
NE.CON.PRVT.PC.KD	Households and NPISHs final consumption expenditu...
NE.CON.PRVT.PC.KD.ZG	Household final consumption expenditure per capit...
NE.CON.PRVT.ZS	Households and NPISHs final consumption expenditu...
NE.CON.TOTL.CD	Final consumption expenditure (current US$)
NE.CON.TOTL.CN	Final consumption expenditure (current LCU)
NE.CON.TOTL.KD	Final consumption expenditure (constant 2015 US$)
NE.CON.TOTL.KD.ZG	Final consumption expenditure (annual % growth)
NE.CON.TOTL.ZS	Final consumption expenditure (% of GDP)
NE.DAB.TOTL.CD	Gross national expenditure (current US$)
NE.DAB.TOTL.CN	Gross national expenditure (current LCU)
NE.DAB.TOTL.KD	Gross national expenditure (constant 2015 US$)
NE.DAB.TOTL.ZS	Gross national expenditure (% of GDP)
NE.EXP.GNFS.CD	Exports of goods and services (current US$)
NE.EXP.GNFS.CN	Exports of goods and services (current LCU)
NE.EXP.GNFS.KD	Exports of goods and services (constant 2015 US$)
NE.EXP.GNFS.KD.ZG	Exports of goods and services (annual % growth)
NE.EXP.GNFS.ZS	Exports of goods and services (% of GDP)
NE.GDI.FTOT.CD	Gross fixed capital formation (current US$)
NE.GDI.FTOT.CN	Gross fixed capital formation (current LCU)
NE.GDI.FTOT.KD	Gross fixed capital formation (constant 2015 US$)
NE.GDI.FTOT.KD.ZG	Gross fixed capital formation (annual % growth)
NE.GDI.FTOT.ZS	Gross fixed capital formation (% of GDP)
NE.GDI.STKB.CD	Changes in inventories (current US$)

NE.GDI.STKB.CN	Changes in inventories (current LCU)
NE.GDI.TOTL.CD	Gross capital formation (current US$)
NE.GDI.TOTL.CN	Gross capital formation (current LCU)
NE.GDI.TOTL.KD	Gross capital formation (constant 2015 US$)
NE.GDI.TOTL.KD.ZG	Gross capital formation (annual % growth)
NE.GDI.TOTL.ZS	Gross capital formation (% of GDP)
NE.IMP.GNFS.CD	Imports of goods and services (current US$)
NE.IMP.GNFS.CN	Imports of goods and services (current LCU)
NE.IMP.GNFS.KD	Imports of goods and services (constant 2015 US$)
NE.IMP.GNFS.KD.ZG	Imports of goods and services (annual % growth)
NE.IMP.GNFS.ZS	Imports of goods and services (% of GDP)
NE.RSB.GNFS.CD	External balance on goods and services (current U...
NE.RSB.GNFS.CN	External balance on goods and services (current L...
NE.RSB.GNFS.ZS	External balance on goods and services (% of GDP)
NE.TRD.GNFS.ZS	Trade (% of GDP)
NV.AGR.EMPL.KD	Agriculture, forestry, and fishing, value added p...
NV.AGR.TOTL.CD	Agriculture, forestry, and fishing, value added (...
NV.AGR.TOTL.CN	Agriculture, forestry, and fishing, value added (...
NV.AGR.TOTL.KD	Agriculture, forestry, and fishing, value added (...
NV.AGR.TOTL.KD.ZG	Agriculture, forestry, and fishing, value added (...
NV.AGR.TOTL.KN	Agriculture, forestry, and fishing, value added (...
NV.AGR.TOTL.ZS	Agriculture, forestry, and fishing, value added (...
NV.IND.EMPL.KD	Industry (including construction), value added pe...
NV.IND.MANF.CD	Manufacturing, value added (current US$)

NV.IND.MANF.CN	Manufacturing, value added (current LCU)
NV.IND.MANF.KD	Manufacturing, value added (constant 2015 US$)
NV.IND.MANF.KD.ZG	Manufacturing, value added (annual % growth)
NV.IND.MANF.ZS	Manufacturing, value added (% of GDP)
NV.IND.TOTL.CD	Industry (including construction), value added (c...
NV.IND.TOTL.CN	Industry (including construction), value added (c...
NV.IND.TOTL.KD	Industry (including construction), value added (c...
NV.IND.TOTL.KD.ZG	Industry (including construction), value added (a...
NV.IND.TOTL.KN	Industry (including construction), value added (c...
NV.IND.TOTL.ZS	Industry (including construction), value added (%...
NV.SRV.EMPL.KD	Services, value added per worker (constant 2015 U...
NV.SRV.TOTL.CD	Services, value added (current US$)
NV.SRV.TOTL.CN	Services, value added (current LCU)
NV.SRV.TOTL.KD	Services, value added (constant 2015 US$)
NV.SRV.TOTL.KD.ZG	Services, value added (annual % growth)
NV.SRV.TOTL.KN	Services, value added (constant LCU)
NV.SRV.TOTL.ZS	Services, value added (% of GDP)
NY.ADJ.AEDU.CD	Adjusted savings: education expenditure (current ...
NY.ADJ.AEDU.GN.ZS	Adjusted savings: education expenditure (% of GNI...
NY.ADJ.DCO2.GN.ZS	Adjusted savings: carbon dioxide damage (% of GNI...
NY.ADJ.DFOR.CD	Adjusted savings: net forest depletion (current U...
NY.ADJ.DFOR.GN.ZS	Adjusted savings: net forest depletion (% of GNI)
NY.ADJ.DKAP.CD	Adjusted savings: consumption of fixed capital (c...
NY.ADJ.DKAP.GN.ZS	Adjusted savings: consumption of fixed capital (%...

NY.ADJ.DMIN.CD	Adjusted savings: mineral depletion (current US$)
NY.ADJ.DMIN.GN.ZS	Adjusted savings: mineral depletion (% of GNI)
NY.ADJ.DNGY.CD	Adjusted savings: energy depletion (current US$)
NY.ADJ.DNGY.GN.ZS	Adjusted savings: energy depletion (% of GNI)
NY.ADJ.DPEM.GN.ZS	Adjusted savings: particulate emission damage (% ...
NY.ADJ.DRES.GN.ZS	Adjusted savings: natural resources depletion (% ...
NY.ADJ.ICTR.GN.ZS	Adjusted savings: gross savings (% of GNI)
NY.ADJ.NNAT.GN.ZS	Adjusted savings: net national savings (% of GNI)
NY.ADJ.NNTY.CD	Adjusted net national income (current US$)
NY.ADJ.NNTY.PC.CD	Adjusted net national income per capita (current ...
NY.GDP.COAL.RT.ZS	Coal rents (% of GDP)
NY.GDP.DEFL.KD.ZG	Inflation, GDP deflator (annual %)
NY.GDP.DEFL.ZS	GDP deflator (base year varies by country)
NY.GDP.DISC.CN	Discrepancy in expenditure estimate of GDP (curre...
NY.GDP.FCST.CD	Gross value added at basic prices (GVA) (current ...
NY.GDP.FCST.CN	Gross value added at basic prices (GVA) (current ...
NY.GDP.FCST.KD	Gross value added at basic prices (GVA) (constant...
NY.GDP.FCST.KN	Gross value added at basic prices (GVA) (constant...
NY.GDP.FRST.RT.ZS	Forest rents (% of GDP)
NY.GDP.MINR.RT.ZS	Mineral rents (% of GDP)
NY.GDP.MKTP.CD	GDP (current US$)
NY.GDP.MKTP.CN	GDP (current LCU)
NY.GDP.MKTP.KD	GDP (constant 2015 US$)
NY.GDP.MKTP.KD.ZG	GDP growth (annual %)

NY.GDP.MKTP.KN	GDP (constant LCU)
NY.GDP.MKTP.PP.CD	GDP, PPP (current international $)
NY.GDP.MKTP.PP.KD	GDP, PPP (constant 2021 international $)
NY.GDP.NGAS.RT.ZS	Natural gas rents (% of GDP)
NY.GDP.PCAP.CD	GDP per capita (current US$)
NY.GDP.PCAP.CN	GDP per capita (current LCU)
NY.GDP.PCAP.KD	GDP per capita (constant 2015 US$)
NY.GDP.PCAP.KD.ZG	GDP per capita growth (annual %)
NY.GDP.PCAP.KN	GDP per capita (constant LCU)
NY.GDP.PCAP.PP.CD	GDP per capita, PPP (current international $)
NY.GDP.PCAP.PP.KD	GDP per capita, PPP (constant 2021 international ...
NY.GDP.PETR.RT.ZS	Oil rents (% of GDP)
NY.GDP.TOTL.RT.ZS	Total natural resources rents (% of GDP)
NY.GDS.TOTL.CD	Gross domestic savings (current US$)
NY.GDS.TOTL.CN	Gross domestic savings (current LCU)
NY.GDS.TOTL.ZS	Gross domestic savings (% of GDP)
NY.GNP.ATLS.CD	GNI, Atlas method (current US$)
NY.GNP.MKTP.CD	GNI (current US$)
NY.GNP.MKTP.CN	GNI (current LCU)
NY.GNP.MKTP.PP.CD	GNI, PPP (current international $)
NY.GNP.PCAP.CD	GNI per capita, Atlas method (current US$)
NY.GNP.PCAP.CN	GNI per capita (current LCU)
NY.GNP.PCAP.PP.CD	GNI per capita, PPP (current international $)
NY.GNS.ICTR.GN.ZS	Gross savings (% of GNI)

NY.GNS.ICTR.ZS	Gross savings (% of GDP)
NY.GSR.NFCY.CD	Net primary income (Net income from abroad) (curr...
NY.GSR.NFCY.CN	Net primary income (Net income from abroad) (curr...
NY.TAX.NIND.CD	Taxes less subsidies on products (current US$)
NY.TAX.NIND.CN	Taxes less subsidies on products (current LCU)
NY.TRF.NCTR.CD	Net secondary income (Net current transfers from ...
NY.TRF.NCTR.CN	Net secondary income (Net current transfers from ...
PA.NUS.ATLS	DEC alternative conversion factor (LCU per US$)
PA.NUS.FCRF	Official exchange rate (LCU per US$, period avera...
SE.ENR.PRIM.FM.ZS	School enrollment, primary (gross), gender parity...
SE.ENR.PRSC.FM.ZS	School enrollment, primary and secondary (gross),...
SE.ENR.SECO.FM.ZS	School enrollment, secondary (gross), gender pari...
SE.ENR.TERT.FM.ZS	School enrollment, tertiary (gross), gender parit...
SE.PRE.ENRR	School enrollment, preprimary (% gross)
SE.PRE.ENRR.FE	School enrollment, preprimary, female (% gross)
SE.PRE.ENRR.MA	School enrollment, preprimary, male (% gross)
SE.PRM.AGES	Primary school starting age (years)
SE.PRM.CMPT.FE.ZS	Primary completion rate, female (% of relevant ag...
SE.PRM.CMPT.MA.ZS	Primary completion rate, male (% of relevant age ...
SE.PRM.CMPT.ZS	Primary completion rate, total (% of relevant age...
SE.PRM.DURS	Primary education, duration (years)
SE.PRM.ENRL	Primary education, pupils
SE.PRM.ENRL.FE.ZS	Primary education, pupils (% female)
SE.PRM.ENRL.TC.ZS	Pupil-teacher ratio, primary

SE.PRM.ENRR	School enrollment, primary (% gross)
SE.PRM.ENRR.FE	School enrollment, primary, female (% gross)
SE.PRM.ENRR.MA	School enrollment, primary, male (% gross)
SE.PRM.GINT.ZS	Gross intake ratio in first grade of primary educ...
SE.PRM.PRIV.ZS	School enrollment, primary, private (% of total p...
SE.PRM.TCHR	Primary education, teachers
SE.PRM.TCHR.FE.ZS	Primary education, teachers (% female)
SE.SEC.AGES	Lower secondary school starting age (years)
SE.SEC.DURS	Secondary education, duration (years)
SE.SEC.ENRL	Secondary education, pupils
SE.SEC.ENRL.FE.ZS	Secondary education, pupils (% female)
SE.SEC.ENRL.GC	Secondary education, general pupils
SE.SEC.ENRL.GC.FE.ZS	Secondary education, general pupils (% female)
SE.SEC.ENRL.VO	Secondary education, vocational pupils
SE.SEC.ENRR	School enrollment, secondary (% gross)
SE.SEC.ENRR.FE	School enrollment, secondary, female (% gross)
SE.SEC.ENRR.MA	School enrollment, secondary, male (% gross)
SE.SEC.TCHR	Secondary education, teachers
SE.TER.ENRR	School enrollment, tertiary (% gross)
SE.TER.ENRR.FE	School enrollment, tertiary, female (% gross)
SE.TER.ENRR.MA	School enrollment, tertiary, male (% gross)
SG.LAW.INDX	Women Business and the Law Index Score (scale 1-1...
SH.DTH.0509	Number of deaths ages 5-9 years
SH.DTH.1014	Number of deaths ages 10-14 years

SH.DTH.1519	Number of deaths ages 15-19 years
SH.DTH.2024	Number of deaths ages 20-24 years
SH.DTH.IMRT	Number of infant deaths
SH.DTH.MORT	Number of under-five deaths
SH.DTH.NMRT	Number of neonatal deaths
SH.DYN.0509	Probability of dying among children ages 5-9 year...
SH.DYN.1014	Probability of dying among adolescents ages 10-14...
SH.DYN.1519	Probability of dying among adolescents ages 15-19...
SH.DYN.2024	Probability of dying among youth ages 20-24 years...
SH.DYN.MORT	Mortality rate, under-5 (per 1,000 live births)
SH.DYN.MORT.FE	Mortality rate, under-5, female (per 1,000 live b...
SH.DYN.MORT.MA	Mortality rate, under-5, male (per 1,000 live bir...
SH.DYN.NMRT	Mortality rate, neonatal (per 1,000 live births)
SH.IMM.IDPT	Immunization, DPT (% of children ages 12-23 month...
SH.IMM.MEAS	Immunization, measles (% of children ages 12-23 m...
SL.AGR.EMPL.FE.ZS	Employment in agriculture, female (% of female em...
SL.AGR.EMPL.MA.ZS	Employment in agriculture, male (% of male employ...
SL.AGR.EMPL.ZS	Employment in agriculture (% of total employment)...
SL.EMP.1524.SP.FE.ZS	Employment to population ratio, ages 15-24, femal...
SL.EMP.1524.SP.MA.ZS	Employment to population ratio, ages 15-24, male ...
SL.EMP.1524.SP.ZS	Employment to population ratio, ages 15-24, total...
SL.EMP.MPYR.FE.ZS	Employers, female (% of female employment) (model...
SL.EMP.MPYR.MA.ZS	Employers, male (% of male employment) (modeled I...
SL.EMP.MPYR.ZS	Employers, total (% of total employment) (modeled...

SL.EMP.SELF.FE.ZS	Self-employed, female (% of female employment) (m...
SL.EMP.SELF.MA.ZS	Self-employed, male (% of male employment) (model...
SL.EMP.SELF.ZS	Self-employed, total (% of total employment) (mod...
SL.EMP.TOTL.SP.FE.ZS	Employment to population ratio, 15+, female (%) (...
SL.EMP.TOTL.SP.MA.ZS	Employment to population ratio, 15+, male (%) (mo...
SL.EMP.TOTL.SP.ZS	Employment to population ratio, 15+, total (%) (m...
SL.EMP.VULN.FE.ZS	Vulnerable employment, female (% of female employ...
SL.EMP.VULN.MA.ZS	Vulnerable employment, male (% of male employment...
SL.EMP.VULN.ZS	Vulnerable employment, total (% of total employme...
SL.EMP.WORK.FE.ZS	Wage and salaried workers, female (% of female em...
SL.EMP.WORK.MA.ZS	Wage and salaried workers, male (% of male employ...
SL.EMP.WORK.ZS	Wage and salaried workers, total (% of total empl...
SL.FAM.WORK.FE.ZS	Contributing family workers, female (% of female ...
SL.FAM.WORK.MA.ZS	Contributing family workers, male (% of male empl...
SL.FAM.WORK.ZS	Contributing family workers, total (% of total em...
SL.GDP.PCAP.EM.KD	GDP per person employed (constant 2021 PPP $)
SL.IND.EMPL.FE.ZS	Employment in industry, female (% of female emplo...
SL.IND.EMPL.MA.ZS	Employment in industry, male (% of male employmen...
SL.IND.EMPL.ZS	Employment in industry (% of total employment) (m...
SL.SRV.EMPL.FE.ZS	Employment in services, female (% of female emplo...
SL.SRV.EMPL.MA.ZS	Employment in services, male (% of male employmen...
SL.SRV.EMPL.ZS	Employment in services (% of total employment) (m...
SL.TLF.ACTI.1524.FE.ZS	Labor force participation rate for ages 15-24, fe...
SL.TLF.ACTI.1524.MA.ZS	Labor force participation rate for ages 15-24, ma...

SL.TLF.ACTI.1524.ZS	Labor force participation rate for ages 15-24, to...
SL.TLF.ACTI.FE.ZS	Labor force participation rate, female (% of fema...
SL.TLF.ACTI.MA.ZS	Labor force participation rate, male (% of male p...
SL.TLF.ACTI.ZS	Labor force participation rate, total (% of total...
SL.TLF.CACT.FE.ZS	Labor force participation rate, female (% of fema...
SL.TLF.CACT.FM.ZS	Ratio of female to male labor force participation...
SL.TLF.CACT.MA.ZS	Labor force participation rate, male (% of male p...
SL.TLF.CACT.ZS	Labor force participation rate, total (% of total...
SL.TLF.TOTL.FE.ZS	Labor force, female (% of total labor force)
SL.TLF.TOTL.IN	Labor force, total
SL.UEM.1524.FE.ZS	Unemployment, youth female (% of female labor for...
SL.UEM.1524.MA.ZS	Unemployment, youth male (% of male labor force a...
SL.UEM.1524.ZS	Unemployment, youth total (% of total labor force...
SL.UEM.TOTL.FE.ZS	Unemployment, female (% of female labor force) (m...
SL.UEM.TOTL.MA.ZS	Unemployment, male (% of male labor force) (model...
SL.UEM.TOTL.ZS	Unemployment, total (% of total labor force) (mod...
SM.POP.NETM	Net migration
SM.POP.REFG	Refugee population by country or territory of asy...
SM.POP.REFG.OR	Refugee population by country or territory of ori...
SP.ADO.TFRT	Adolescent fertility rate (births per 1,000 women...
SP.DYN.AMRT.FE	Mortality rate, adult, female (per 1,000 female a...
SP.DYN.AMRT.MA	Mortality rate, adult, male (per 1,000 male adult...
SP.DYN.CBRT.IN	Birth rate, crude (per 1,000 people)
SP.DYN.CDRT.IN	Death rate, crude (per 1,000 people)

SP.DYN.IMRT.FE.IN	Mortality rate, infant, female (per 1,000 live bi...
SP.DYN.IMRT.IN	Mortality rate, infant (per 1,000 live births)
SP.DYN.IMRT.MA.IN	Mortality rate, infant, male (per 1,000 live birt...
SP.DYN.LE00.FE.IN	Life expectancy at birth, female (years)
SP.DYN.LE00.IN	Life expectancy at birth, total (years)
SP.DYN.LE00.MA.IN	Life expectancy at birth, male (years)
SP.DYN.TFRT.IN	Fertility rate, total (births per woman)
SP.DYN.TO65.FE.ZS	Survival to age 65, female (% of cohort)
SP.DYN.TO65.MA.ZS	Survival to age 65, male (% of cohort)
SP.POP.0004.FE.5Y	Population ages 00-04, female (% of female popula...
SP.POP.0004.MA.5Y	Population ages 00-04, male (% of male population...
SP.POP.0014.FE.IN	Population ages 0-14, female
SP.POP.0014.FE.ZS	Population ages 0-14, female (% of female populat...
SP.POP.0014.MA.IN	Population ages 0-14, male
SP.POP.0014.MA.ZS	Population ages 0-14, male (% of male population)
SP.POP.0014.TO	Population ages 0-14, total
SP.POP.0014.TO.ZS	Population ages 0-14 (% of total population)
SP.POP.0509.FE.5Y	Population ages 05-09, female (% of female popula...
SP.POP.0509.MA.5Y	Population ages 05-09, male (% of male population...
SP.POP.1014.FE.5Y	Population ages 10-14, female (% of female popula...
SP.POP.1014.MA.5Y	Population ages 10-14, male (% of male population...
SP.POP.1519.FE.5Y	Population ages 15-19, female (% of female popula...
SP.POP.1519.MA.5Y	Population ages 15-19, male (% of male population...
SP.POP.1564.FE.IN	Population ages 15-64, female

SP.POP.1564.FE.ZS	Population ages 15-64, female (% of female popula...
SP.POP.1564.MA.IN	Population ages 15-64, male
SP.POP.1564.MA.ZS	Population ages 15-64, male (% of male population...
SP.POP.1564.TO	Population ages 15-64, total
SP.POP.1564.TO.ZS	Population ages 15-64 (% of total population)
SP.POP.2024.FE.5Y	Population ages 20-24, female (% of female popula...
SP.POP.2024.MA.5Y	Population ages 20-24, male (% of male population...
SP.POP.2529.FE.5Y	Population ages 25-29, female (% of female popula...
SP.POP.2529.MA.5Y	Population ages 25-29, male (% of male population...
SP.POP.3034.FE.5Y	Population ages 30-34, female (% of female popula...
SP.POP.3034.MA.5Y	Population ages 30-34, male (% of male population...
SP.POP.3539.FE.5Y	Population ages 35-39, female (% of female popula...
SP.POP.3539.MA.5Y	Population ages 35-39, male (% of male population...
SP.POP.4044.FE.5Y	Population ages 40-44, female (% of female popula...
SP.POP.4044.MA.5Y	Population ages 40-44, male (% of male population...
SP.POP.4549.FE.5Y	Population ages 45-49, female (% of female popula...
SP.POP.4549.MA.5Y	Population ages 45-49, male (% of male population...
SP.POP.5054.FE.5Y	Population ages 50-54, female (% of female popula...
SP.POP.5054.MA.5Y	Population ages 50-54, male (% of male population...
SP.POP.5559.FE.5Y	Population ages 55-59, female (% of female popula...
SP.POP.5559.MA.5Y	Population ages 55-59, male (% of male population...
SP.POP.6064.FE.5Y	Population ages 60-64, female (% of female popula...
SP.POP.6064.MA.5Y	Population ages 60-64, male (% of male population...
SP.POP.6569.FE.5Y	Population ages 65-69, female (% of female popula...

SP.POP.6569.MA.5Y	Population ages 65-69, male (% of male population...
SP.POP.65UP.FE.IN	Population ages 65 and above, female
SP.POP.65UP.FE.ZS	Population ages 65 and above, female (% of female...
SP.POP.65UP.MA.IN	Population ages 65 and above, male
SP.POP.65UP.MA.ZS	Population ages 65 and above, male (% of male pop...
SP.POP.65UP.TO	Population ages 65 and above, total
SP.POP.65UP.TO.ZS	Population ages 65 and above (% of total populati...
SP.POP.7074.FE.5Y	Population ages 70-74, female (% of female popula...
SP.POP.7074.MA.5Y	Population ages 70-74, male (% of male population...
SP.POP.7579.FE.5Y	Population ages 75-79, female (% of female popula...
SP.POP.7579.MA.5Y	Population ages 75-79, male (% of male population...
SP.POP.80UP.FE.5Y	Population ages 80 and above, female (% of female...
SP.POP.80UP.MA.5Y	Population ages 80 and above, male (% of male pop...
SP.POP.BRTH.MF	Sex ratio at birth (male births per female births...
SP.POP.DPND	Age dependency ratio (% of working-age population...
SP.POP.DPND.OL	Age dependency ratio, old (% of working-age popul...
SP.POP.DPND.YG	Age dependency ratio, young (% of working-age pop...
SP.POP.GROW	Population growth (annual %)
SP.POP.TOTL	Population, total
SP.POP.TOTL.FE.IN	Population, female
SP.POP.TOTL.FE.ZS	Population, female (% of total population)
SP.POP.TOTL.MA.IN	Population, male
SP.POP.TOTL.MA.ZS	Population, male (% of total population)
SP.RUR.TOTL	Rural population

SP.RUR.TOTL.ZG	Rural population growth (annual %)
SP.RUR.TOTL.ZS	Rural population (% of total population)
SP.URB.GROW	Urban population growth (annual %)
SP.URB.TOTL	Urban population
SP.URB.TOTL.IN.ZS	Urban population (% of total population)
TG.VAL.TOTL.GD.ZS	Merchandise trade (% of GDP)
TM.VAL.AGRI.ZS.UN	Agricultural raw materials imports (% of merchand...
TM.VAL.FOOD.ZS.UN	Food imports (% of merchandise imports)
TM.VAL.FUEL.ZS.UN	Fuel imports (% of merchandise imports)
TM.VAL.INSF.ZS.WT	Insurance and financial services (% of commercial...
TM.VAL.MANF.ZS.UN	Manufactures imports (% of merchandise imports)
TM.VAL.MMTL.ZS.UN	Ores and metals imports (% of merchandise imports...
TM.VAL.MRCH.AL.ZS	Merchandise imports from economies in the Arab Wo...
TM.VAL.MRCH.CD.WT	Merchandise imports (current US$)
TM.VAL.MRCH.HI.ZS	Merchandise imports from high-income economies (%...
TM.VAL.MRCH.OR.ZS	Merchandise imports from low- and middle-income e...
TM.VAL.MRCH.R1.ZS	Merchandise imports from low- and middle-income e...
TM.VAL.MRCH.R2.ZS	Merchandise imports from low- and middle-income e...
TM.VAL.MRCH.R3.ZS	Merchandise imports from low- and middle-income e...
TM.VAL.MRCH.R4.ZS	Merchandise imports from low- and middle-income e...
TM.VAL.MRCH.R5.ZS	Merchandise imports from low- and middle-income e...
TM.VAL.MRCH.R6.ZS	Merchandise imports from low- and middle-income e...
TM.VAL.MRCH.RS.ZS	Merchandise imports by the reporting economy, res...
TM.VAL.MRCH.WL.CD	Merchandise imports by the reporting economy (cur...

TM.VAL.MRCH.WR.ZS	Merchandise imports from low- and middle-income e...
TM.VAL.MRCH.XD.WD	Import value index (2000 = 100)
TM.VAL.OTHR.ZS.WT	Computer, communications and other services (% of...
TM.VAL.SERV.CD.WT	Commercial service imports (current US$)
TM.VAL.TRAN.ZS.WT	Transport services (% of commercial service impor...
TM.VAL.TRVL.ZS.WT	Travel services (% of commercial service imports)
TX.VAL.AGRI.ZS.UN	Agricultural raw materials exports (% of merchand...
TX.VAL.FOOD.ZS.UN	Food exports (% of merchandise exports)
TX.VAL.FUEL.ZS.UN	Fuel exports (% of merchandise exports)
TX.VAL.INSF.ZS.WT	Insurance and financial services (% of commercial...
TX.VAL.MANF.ZS.UN	Manufactures exports (% of merchandise exports)
TX.VAL.MMTL.ZS.UN	Ores and metals exports (% of merchandise exports...
TX.VAL.MRCH.AL.ZS	Merchandise exports to economies in the Arab Worl...
TX.VAL.MRCH.CD.WT	Merchandise exports (current US$)
TX.VAL.MRCH.HI.ZS	Merchandise exports to high-income economies (% o...
TX.VAL.MRCH.OR.ZS	Merchandise exports to low- and middle-income eco...
TX.VAL.MRCH.R1.ZS	Merchandise exports to low- and middle-income eco...
TX.VAL.MRCH.R2.ZS	Merchandise exports to low- and middle-income eco...
TX.VAL.MRCH.R3.ZS	Merchandise exports to low- and middle-income eco...
TX.VAL.MRCH.R4.ZS	Merchandise exports to low- and middle-income eco...
TX.VAL.MRCH.R5.ZS	Merchandise exports to low- and middle-income eco...
TX.VAL.MRCH.R6.ZS	Merchandise exports to low- and middle-income eco...
TX.VAL.MRCH.RS.ZS	Merchandise exports by the reporting economy, res...
TX.VAL.MRCH.WL.CD	Merchandise exports by the reporting economy (cur...

TX.VAL.MRCH.WR.ZS	Merchandise exports to low- and middle-income eco...
TX.VAL.MRCH.XD.WD	Export value index (2000 = 100)
TX.VAL.OTHR.ZS.WT	Computer, communications and other services (% of...
TX.VAL.SERV.CD.WT	Commercial service exports (current US$)
TX.VAL.TRAN.ZS.WT	Transport services (% of commercial service expor...
TX.VAL.TRVL.ZS.WT	Travel services (% of commercial service exports)