

**PONTIFÍCIA UNIVERSIDADE CATÓLICA DE MINAS GERAIS**NÚCLEO DE EDUCAÇÃO A DISTÂNCIA

Pós-graduação *Lato Sensu* em Ciência de Dados e Big Data

**RELATÓRIO TÉCNICO**

UM MODELO DE APRENDIZADO DE MÁQUINA SUPERVISIONADO PARA PREVISÃO DE QUANTIDADE DE PÚBLICO NOS JOGOS DO CAMPEONATO BRASILEIRO DE FUTEBOL

Salomão Fernandes de Freitas Júnior

Belo Horizonte

2024

**SUMÁRIO**

[1. Introdução 3](#_Toc162890461)

[2. Entendimento do Domínio do Problema 4](#_Toc162890462)

[2.1. Contexto e Requisitos do Problema 4](#_Toc162890463)

[2.2. Definição da Estratégia a ser utilizada 6](#_Toc162890464)

[2.3. Escolha e Obtenção das Bases de Dados 7](#_Toc162890465)

[3. Preparação da Base de Dados 8](#_Toc162890466)

[3.1. Ambiente Tecnológico 9](#_Toc162890467)

[3.2. Análise Exploratória dos Dados 9](#_Toc162890468)

[3.3. Seleção de Atributos (Redução de Dimensionalidade) 13](#_Toc162890469)

[3.4. Ajustes de Valores e Tipos de Dados 14](#_Toc162890470)

[3.5. Análise de Valores Vazios 15](#_Toc162890471)

[3.5.1 Tratamento de Valores Vazios no *dataframe* df\_brasileirao\_comp 16](#_Toc162890472)

[3.5.2 Tratamento de Valores Vazios no *dataframe* df\_brasileirao 17](#_Toc162890473)

[3.6. Análise de Correlação 26](#_Toc162890474)

[3.6. Análise de Outliers 26](#_Toc162890475)

[3.7. Engenharia de Atributos (feature enginering) 26](#_Toc162890476)

[(Criação de Novos Atributos) 26](#_Toc162890477)

[3.8 Transformação de variáveis numéricas em categóricas 27](#_Toc162890478)

[3.9 Agrupamento de categorias (variáveis categóricas) 27](#_Toc162890479)

[3.10 Nova Seleção de Atributos 27](#_Toc162890480)

[3.11. Padronização/Normalização de Variáveis 27](#_Toc162890481)

[3.12. Codificação de Variáveis Categóricas 27](#_Toc162890482)

[4. Escolha do Modelo 27](#_Toc162890483)

[5. Implantação da Solução em Ambiente de Produção 28](#_Toc162890484)

[6. Links 28](#_Toc162890485)

[REFERÊNCIAS 29](#_Toc162890486)

# 1. Introdução

Atualmente muitas das técnicas de Análise de Dados, *Business Intelligence* (BI) e Ciência de Dados, vêm sendo aplicadas para aprimorar processos decisórios em todos os aspectos dos esportes, uma área em franca ebulição denominada análise de dados esportivos (*sports analytics*) [1. Sharda, Ramesh].

Em nosso recente Trabalho de Conclusão da Pós-Graduação em *Analytics e Business Intelligence* [referência módulo A] [referência módulo B] [referência módulo C], concluído em novembro de 2022, pela PUC-Minas, já abordamos esse tema. O referido trabalho se constituiu de três módulos, os quais abordaram respectivamente, os seguintes objetos:

- Modelagem e Construção de um *Data Warehouse* do Campeonato Brasileiro da era de pontos corridos (2003 a 2021);

- Desenvolvimento da camada de apresentação da solução de BI, com um *Dashboard*, com três níveis de perspectivas interligadas entre si: Estratégico, Tático e Operacional, exibindo informações a partir dos dados do *Data Warehouse*;

- Construção de um modelo simples de aprendizado de máquina (*machine learning*) supervisionado, utilizando a técnica de Classificação, através de um algoritmo de Árvore de Decisão, com o objetivo de predizer o resultado de jogos (VITÓRIA, EMPATE, DERROTA), sob a perspectiva de um clube específico.

Neste atual projeto acadêmico, seguimos na mesma temática do nosso trabalho prévio, citado anteriormente. Durante o decorrer das disciplinas da Pós-graduação em Ciência de Dados e Big Data, foi possível obter um aprofundamento do aprendizado das técnicas envolvidas em todas as etapas de um projeto de ciência de dados, bem como dos principais algoritmos de aprendizado de máquina e dos aspectos envolvidos na sua utilização.

Dessa forma, apresentamos neste trabalho um modelo de aprendizado máquina supervisionado, utilizando a técnica de Regressão, para predição de quantidade de público nos jogos do campeonato brasileiro de futebol.

O objetivo principal deste trabalho, foi a consolidação de conceitos, técnicas e abordagens tratados durante as disciplinas do curso de Pós-graduação *Lato Sensu* em Ciência de Dados e Big Data, com o desenvolvimento de um projeto completo de ciência de dados, em todas as suas etapas.

A figura 1, a seguir, ilustra as etapas de um projeto de Ciência de Dados, as quais detalhamos nos próximos tópicos, dentro do contexto do projeto desenvolvido neste trabalho.

FIGURA 1-22 DO LIVRO Pratical Machine Learnign withi Python (pág. 46)

OU 1-23 DO MESMO LIVRO OU 1-24

FIGURA: ETAPAS DE UM PROJETO DE CIÊNCIA DE DADOS (VER REFERÊNCIAS)

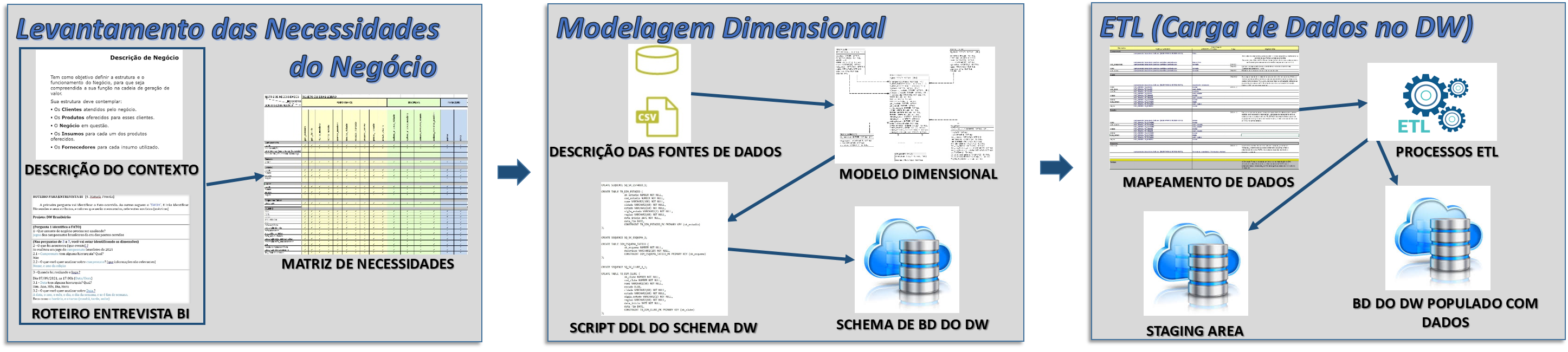


FIG. 1 – Etapas de Projeto de BI Abordadas no Módulo A do Relatório Técnico

# 2. Entendimento do Domínio do Problema

(Ver Disciplina Preparação de Dados para Machine Learnig)

(Ver citações nas referências)

Conforme descrito em [9.Zárate, Luis], nesta primeira etapa do projeto, o cientista de dados deve, juntamente com o especialista do negócio, entender e caracterizar o domínio do problema.

Devem ser bem entendidos o contexto e os requisitos do problema, identificadas as bases de dados disponíveis na organização ou mesmo externamente, identificando-se as características (atributos) relevantes que possam ser obtidos diretamente nas bases de dados disponíveis, ou que possam ser derivados destes.

Nos próximos tópicos descrevemos a etapa de Entendimento do Domínio do Problema no âmbito do nosso projeto.

## 2.1. Contexto e Requisitos do Problema

No nosso projeto, desejamos realizar a predição de quantidade de público nos jogos do campeonato brasileiro de futebol. Essa predição deverá ser realizada baseada em determinadas características que podem ser relevantes na determinação do interesse do público em um determinado jogo.

A predição do público esperado para um jogo, pode ser útil para determinação da logística envolvida na realização do mesmo, como confecção e venda de ingressos, quantidade de alimentos e outros itens a serem disponibilizados para venda no estádio, quantidade de força de segurança a ser disponibilizada dentro e fora do estádio, quantidade de efetivo e horário de funcionamento de transporte público, dentre outros aspectos. Também, o organizador do jogo poderá reforçar as estratégias de marketing, visando o maior incentivo da torcida naqueles jogos onde, pelas suas características, espera-se um baixo comparecimento de público.

Nosso escopo envolve somente o campeonato brasileiro de futebol, que caracteriza-se por ser um campeonato de “pontos corridos”, onde não existem jogos eliminatórios. O desempenho de um clube se define pela quantidade de pontos conquistados durante todo o campeonato.

Dessa forma, tentamos entender as características envolvidas em um jogo, que podem ter influência no interesse de uma torcida em comparecer ao mesmo, para prestigiar o time, podendo citar as seguintes:

- **Etapa do campeonato**: espera-se que as pessoas tenham um interesse menor no jogos iniciais, quando não há ainda uma definição clara de classificação;

- **Desempenho Geral de seu clube**: após passados os jogos iniciais, já se tem uma tendência de desempenho dos clubes, e um clube com bom desempenho naquele momento do campeonato, pode se mostrar postulante a ser campeão, ou a obter boas colocações, o que espera-se que gere incentivo em sua torcida a comparecer nos seus jogos locais;

- **Importância do clube adversário**: esta pode ser uma característica que motive uma torcida a comparecer a um jogo de seu clube na sua cidade. Clubes com maior tradição ou relevância, em visita a uma cidade, podem influenciar no interesse de uma torcida local a comparecer ao estádio;

- **Desempenho recente do seu clube**: também, mesmo que um clube não esteja em uma boa colocação no campeonato, espera-se que caso apresente um bom desempenho nos últimos jogos, isso gere um inventivo a sua torcida, pela tendência de crescimento. Da mesma forma, um clube bem colocado no campeonato, mas com desempenho ruim nos jogos mais recentes, pode causar desmotivação a sua torcida;

- **Dia da semana ou horário do jogo**: o dia de realização de um jogo (fim de semana ou dia útil) e o horário do mesmo (manhã, tarde, noite), podem interferir no interesse ou possibilidade de comparecimento de torcida;

- **Período do ano**: alguns meses podem apresentar tendência de maior ou menor comparecimento de torcida.

Além dos atributos citados, outras características mais julgamentais, como por exemplo o perfil de torcida, ou mesmo a condição climática, ou até o acontecimento de outros eventos na cidade, podem influenciar no comparecimento de público em um jogo.

**(VER EXPLICAÇÃO DO TERMO JULGAMENTAIS – DISCIPLINA DO CURSO)**

## 2.2. Definição da Estratégia a ser utilizada

Dado o entendimento inicial do problema e de seus requisitos, descritos no tópico anterior, foi possível definir a estratégia de solução do mesmo.

Assim, optamos por utilizar uma técnica de aprendizado de máquina supervisionado, já que queremos predizer o valor de um atributo (Quantidade de Público), levando em consideração dados históricos presentes em um conjunto de dados (*Dataset*).

Algoritmos de aprendizado supervisionado, basicamente, tentam modelar relacionamentos entre valores de atributos de entrada e suas saídas (atributos-alvo) correspondentes, a partir de dados de treinamento. Assim, tornam-se capazes de predizer valores de saída para novos dados de entrada. [2.Sarkar, Dipanjan]

Existem dois principais tipos de métodos de aprendizado supervisionado: **Classificação**, no qual o objetivo é predizer valores de saída para um atributo categórico (conhecidos como classes); e **Regressão**, que por outro lado, objetivam a estimação de valores de saída para atributos numéricos contínuos, conforme descrito em [2.Sarkar, Dipanjan] e [3.Guido, Sarah].

No nosso projeto, utilizamos a técnica de **Regressão**, uma vez que nosso atributo-alvo (Quantidade de Público) é um valor numérico, que embora seja representado por um número inteiro, devido à grandeza e variabilidade do mesmo, podemos tratá-lo como contínuo. Nossos atributos de entrada serão, a princípio, aqueles citados no tópico 2.1, conforme disponibilidade dos mesmos no conjunto de dados históricos (*Dataset*) que teremos disponível.

Nosso modelo deverá “aprender” com os dados históricos (os quais devem possuir registros que contenham tanto os atributos de entrada quanto o atributo-alvo com valores preenchidos), e deverá ser capaz de predizer o valor do atributo-alvo para novos dados de entrada.

## 2.3. Escolha e Obtenção das Bases de Dados

Após o entendimento geral do domínio do problema, já pudemos estabelecer uma estratégia para a solução. No entanto, precisamos verificar a disponibilidade das informações (atributos) definidas como requisitos da solução. Essas informações podem estar disponíveis nas bases de dados da própria organização para quem se desenvolve a solução, ou poderão ser obtidas externamente.

Para o nosso projeto, realizamos uma pesquisa geral na internet, bem como em sites mais específicos, conhecidos da comunidade de ciência de dados, destacando-se o *Kaggle* [<https://www.kaggle.com/datasets>] e o *Base Dos Dados* [<https://basedosdados.org/>].

Dentre os *datasets* encontrados, selecionamos aquele disponibilizado em

[<https://basedosdados.org/dataset/c861330e-bca2-474d-9073-bc70744a1b23?table=18835b0d-233e-4857-b454-1fa34a81b4fa>].

O *dataset* **brasileirao\_serie\_a.csv**, selecionado, possui a maioria dos atributos que definimos como requisito da nossa solução, e inclusive o nosso atributo-alvo (Quantidade de Público), com dados históricos do campeonato brasileiro de futebol da era de pontos corridos. O *dataset* disponibilizado de forma gratuita, contém dados de 29/03/2003 até 26/10/2023.

Para preenchimento dos dados faltantes (de 27/10/2023 a 7/12/2023), de modo a contemplar todos os jogos realizados até o momento da escrita deste trabalho, utilizamos as informações contidas no site [[https://www.srgoool.com.br](https://www.srgoool.com.br/)], tendo sido possível completar quase todas as informações de interesse, as quais concentramos em um segundo *dataset*, **brasileirao\_serie\_a\_dados\_complementares\_2023.csv**.

Dessa forma, para o nosso projeto, obtivemos dois *datasets*, os quais abrigam dados de todos os jogos do campeonato brasileiro de futebol da era de pontos corridos, desde de seu início (2003), até o último realizado (2023). Os *datasets* estão disponíveis no repositório do projeto, em [link]

A estrutura completa dos *datasets* é composta dos seguintes campos, auto-explicativos: *ano\_campeonato, data, rodada, estadio, arbitro, publico, publico\_max, time\_mandante, time\_visitante, tecnico\_mandante, tecnico\_visitante, colocacao\_mandante, colocacao\_visitante, valor\_equipe\_titular\_mandante, valor\_equipe\_titular\_visitante, idade\_media\_titular\_mandante, idade\_media\_titular\_visitante, gols\_mandante, gols\_visitante, gols\_1\_tempo\_mandante, gols\_1\_tempo\_visitante, escanteios\_mandante, escanteios\_visitante, faltas\_mandante, faltas\_visitante, chutes\_bola\_parada\_mandante, chutes\_bola\_parada\_visitante, defesas\_mandante, defesas\_visitante, impedimentos\_mandante, impedimentos\_visitante, chutes\_mandante, chutes\_visitante, chutes\_fora\_mandante, chutes\_fora\_visitante.*

# 3. Preparação da Base de Dados

Um dos principais desafios da ciência de dados é lidar com dados de baixa qualidade, pois isso tem um forte efeito negativo no desempenho obtido nas etapas seguintes. Para minimizar esse problema, é importante avaliar logo no início a qualidade do dados e tratar ou corrigir as imperfeições [X.Carvalho, André].

Dentre os problemas mais comuns que podem ser encontrados em bases de dados, podemos citar: atributos com valores ausentes, redundantes, inconsistentes, com ruídos, com valores atípicos (*outliers*) ou enviesados [X.Carvalho, André].

A preparação dos dados, é a etapa de maior investimento de tempo e esforço dentro de um projeto de ciência de dados. Devemos deixar a base de dados livre de problemas, para que em seguida, a mesma possa ser submetida a algoritmos de *machine learning.*

Pode-se notar que nem todos os atributos do *dataset* (listados no tópico 2.3) serão de nosso interesse, e ainda, que poderemos ter a necessidade criar algum atributo de interesse, que venha a enriquecer a nossa base, e consequentemente, as nossas análises.

Nessa etapa, iniciamos com um entendimento geral da base de dados, para em seguida realizar a limpeza e transformações necessárias nos dados.

# 3.1. Ambiente Tecnológico

Desenvolvemos nosso projeto, utilizando a plataforma de nuvem Google Colab [<https://colab.research.google.com/>], com a utilização da linguagem Python, fazendo uso de bibliotecas próprias para análise/preparação de dados e *machine learning*, como *sweetviz, pandas, numpy, matplotlib, sklearning, seaborn, tensorflow e keras*.

## 3.2. Análise Exploratória dos Dados

O objetivo da Análise Exploratória é estudar a base de dados que se está trabalhando e entender padrões e pontos de atenção que podem impactar em análises posteriores. [refer. curso python para data science]

Aqui nessa etapa, exploramos livremente o nosso *dataset*, para obter um maior entendimento do mesmo, e assim “ganhar intimidade” com os dados. Aqui já começamos a entender as estratégias a serem adotadas para a efetiva preparação da base de dados, necessária antes de se aplicar qualquer algoritmo de *machine learning* sobre a base de dados.

Dentre as principais atividades realizadas na análise exploratória estão:

* Verificação da estrutura do *dataset* (atributos disponíveis);
* Verificação do tipo de dados dos atributos;
* Exploração visual dos dados;
* Obtenção da estatística descritiva dos atributos numéricos;
* Obtenção das tabelas de frequência dos atributos categóricos;
* Detecção de dados nulos;
* Detecção de outliers.

Realizamos nossa análise exploratória, através do notebook Python ***01 – Análise Exploratória***, acessível em [link notebook], no qual estudamos de forma geral a estrutura e conteúdo da nossa base de dados.

Iniciamos realizando a leitura do *dataset*, gerando um *Dataframe* para a visualização de todas as colunas disponíveis. Em seguida, já selecionamos somente as colunas que, a princípio, seriam de nosso interesse. Na figura XX, é exibido, no ambiente *google colab*, um trecho do código python, onde selecionamos as colunas desejadas, e exibimos algumas linhas do *Dataframe* para análise inicial. Cada linha do *Dataframe* se refere a um jogo realizado.

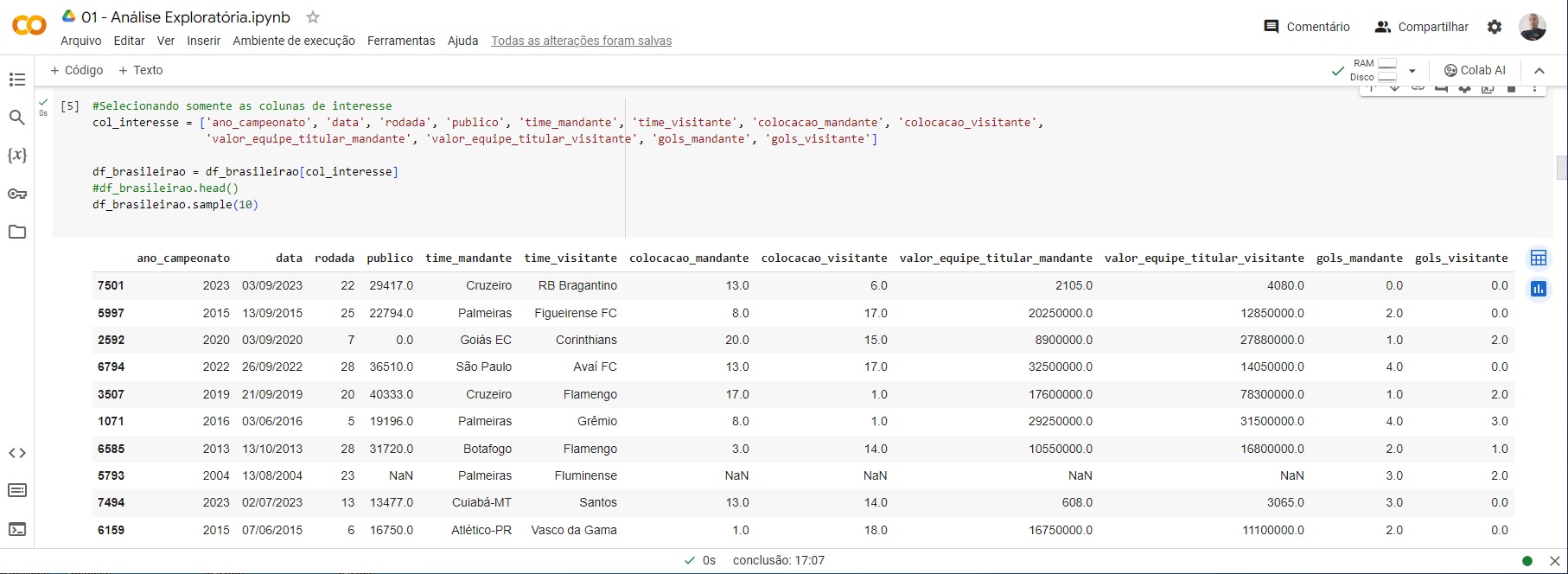


FIG. XX – Seleção das colunas de interesse e exibição do *Dataframe*

Após o reconhecimento da estrutura da base de dados, utilizamos a biblioteca ***sweetviz***, a qual nos fornece uma análise exploratória geral, de forma automatizada. Parte dos resultados é exibida na figura XX.

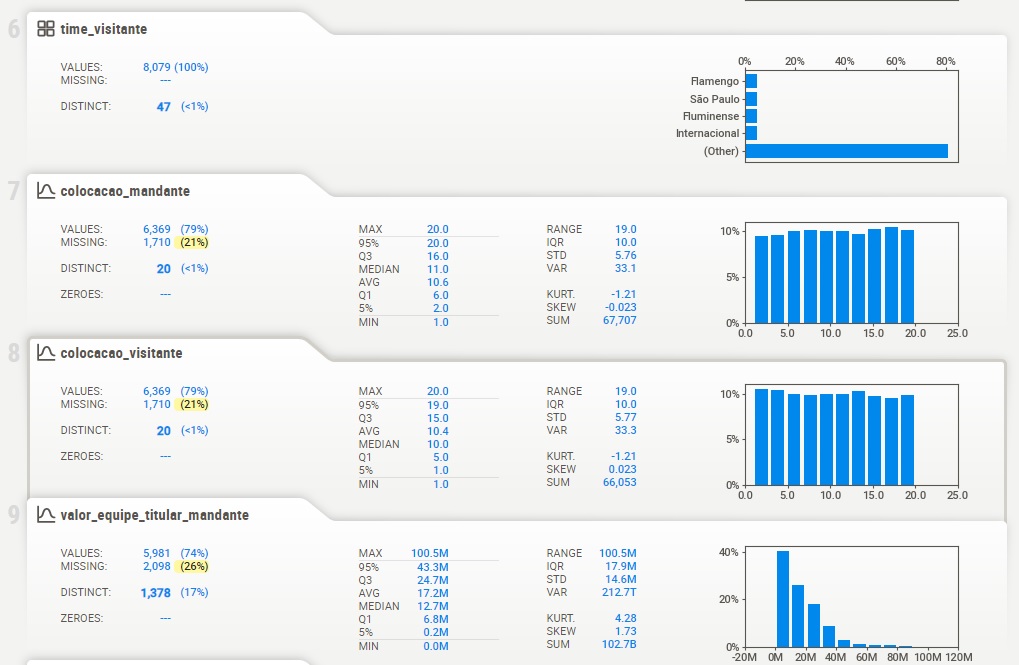


FIG. XX – Análise Exploratória geral com a biblioteca Sweetviz

Com a análise exploratória acima, já podemos observar que alguns atributos apresentam dados faltantes (nulos). Neste momento, estamos apenas realizando um conhecimento dos dados disponíveis. No entanto, mais à frente no projeto, quando efetivamente realizarmos a preparação da base de dados, vamos analisar individualmente cada atributo que apresenta dados vazios (nulos), para decidir a estratégia de preenchimento dos mesmos, ou ainda pela exclusão das linhas que apresentam atributos com dados nulos.

Outra percepção que obtivemos nesta etapa de análise exploratória, foi com relação aos atributos categóricos ***time\_mandante*** e ***time\_visitante***, onde foi possível observar em suas tabelas de frequência, a ocorrência duplicada dos seguintes times, os quais aparecem com nomes diferentes em linhas diversas do *Dataframe*:

* Athletico-PR / Atlético-PR
* Goiás / Goiás EC
* Santos / Santos FC

Assim, ao realizar a preparação da base de dados, teremos que padronizar esses nomes.

Para finalizar esta etapa de análise exploratória dos dados, observamos também o *dataset* **brasileirao\_serie\_a\_dados\_complementares\_2023.csv**, o qual foi preenchido por nós, com dados complementares (jogos no período de 27/10/2023 a 7/12/2023), os quais não estão disponibilizados no *dataset* **brasileirao\_serie\_a.csv**, conforme explicado no item 2.3 (Escolha e Obtenção das Bases de Dados) deste trabalho. Dessa forma, já sabíamos que os campos ***valor\_equipe\_titular\_mandante*** e ***valor\_equipe\_titular\_visitante***, estavam nulos, pois não tínhamos essa informação disponível para preenchimento. Ao realizarmos a preparação da base de dados, na próxima etapa do projeto, iremos decidir por uma estratégia de preenchimento dos mesmos.

A figura XX, a seguir ilustra um trecho do código, onde é realizada a leitura do *dataset* complementar, e exibidas algumas linhas do mesmo em um *dataframe*, onde pode-se observar os dados vazios dos atributos citados.

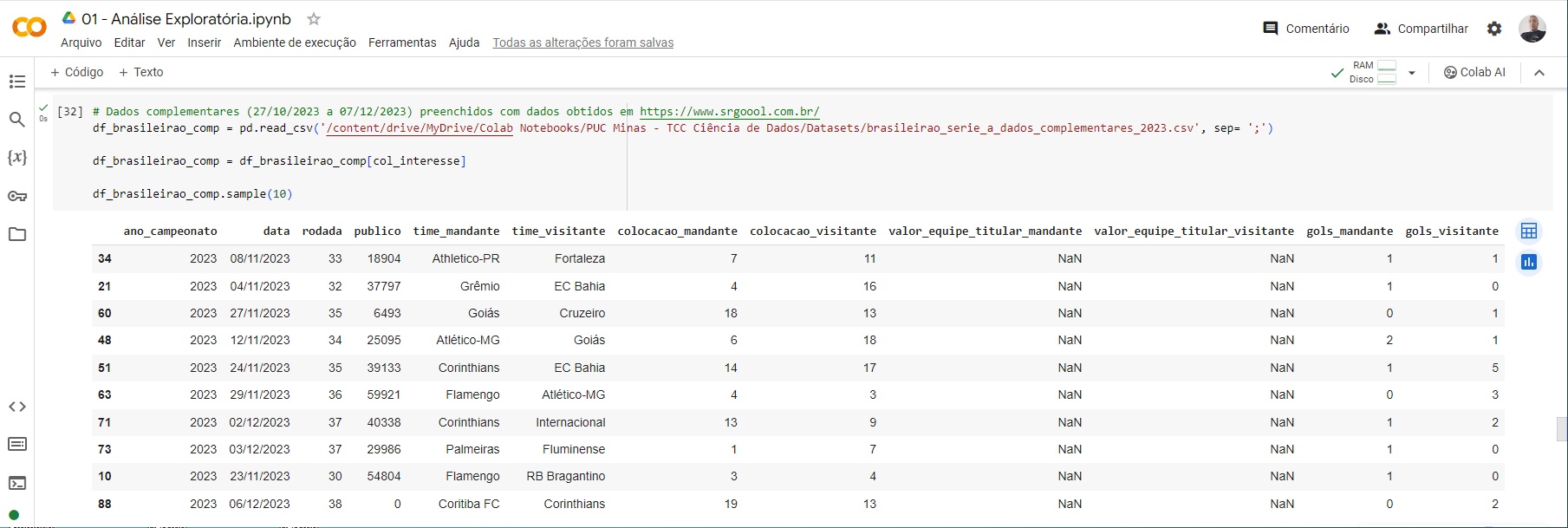


FIG. XX – *Dataframe* com os dados complementares

O Notebook Python completo com a Análise Exploratória tratada neste tópico pode ser acessado no repositório do projeto, em [link notebook].

Nos próximos tópicos, iremos avançar na preparação da base de dados, a partir das observações já realizadas até aqui, e realizaremos efetivamente os ajustes necessários na base de dados, deixando-a pronta para aplicação dos algoritmos de *machine learning*. Para fins de melhor organização, iniciaremos um novo Notebook Python onde concentraremos a preparação dos dados, sendo o mesmo acessível em [link].

Na figura XX, a seguir, mostramos um trecho do código do nosso notebook python ***02 – Preparação de Dados.ipynb***, onde é realizada a leitura e exibição dos nossos dois *datasets*, agora contidos nos *dataframes* **df\_brasileirao** e **df\_brasilerao\_comp**.

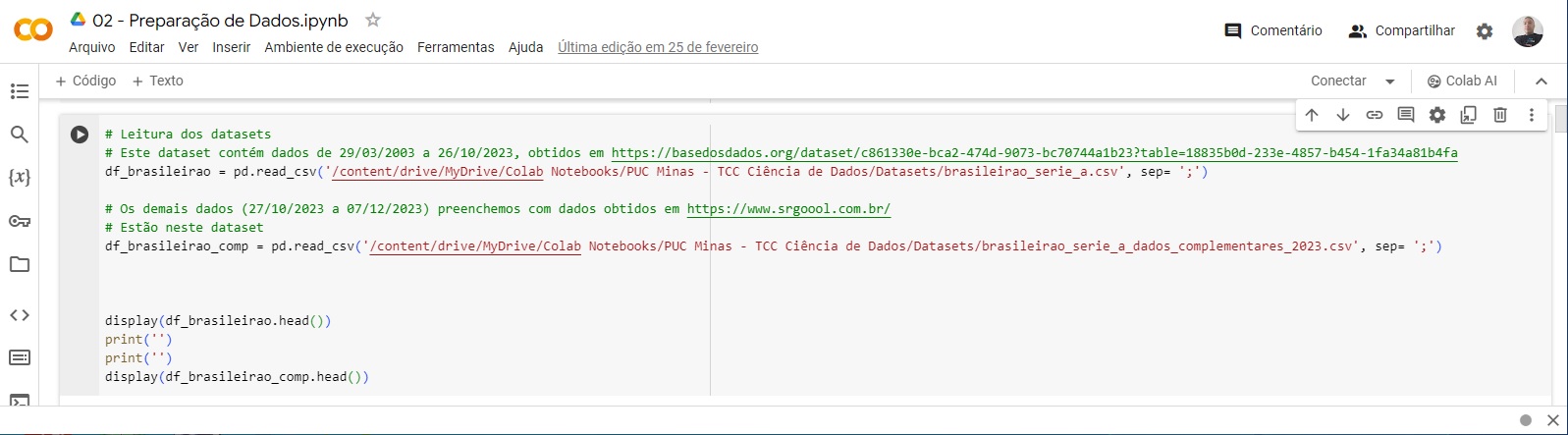


FIG. XX – Leitura dos *datasets* para a etapa de preparação de dados

## 3.3. Seleção de Atributos (Redução de Dimensionalidade)

Em nossa análise exploratória prévia, descrita no tópico 3.2, já pudemos identificar nos *datasets*, os atributos de interesse do nosso projeto, conforme requisitos de negócio descritos no item **2.1. Contexto e Requisitos do Problema.**

Assim, reduzimos a dimensionalidade da nossa base, descartando os atributos irrelevantes para o nosso problema. Fizemos isso no nosso código python, no ambiente *google colab*, onde selecionamos somente os atributos de nosso interesse, no total de 12, e em seguida exibimos os *dataframes* para visualização, conforme ilustrado na figura XX, a seguir.

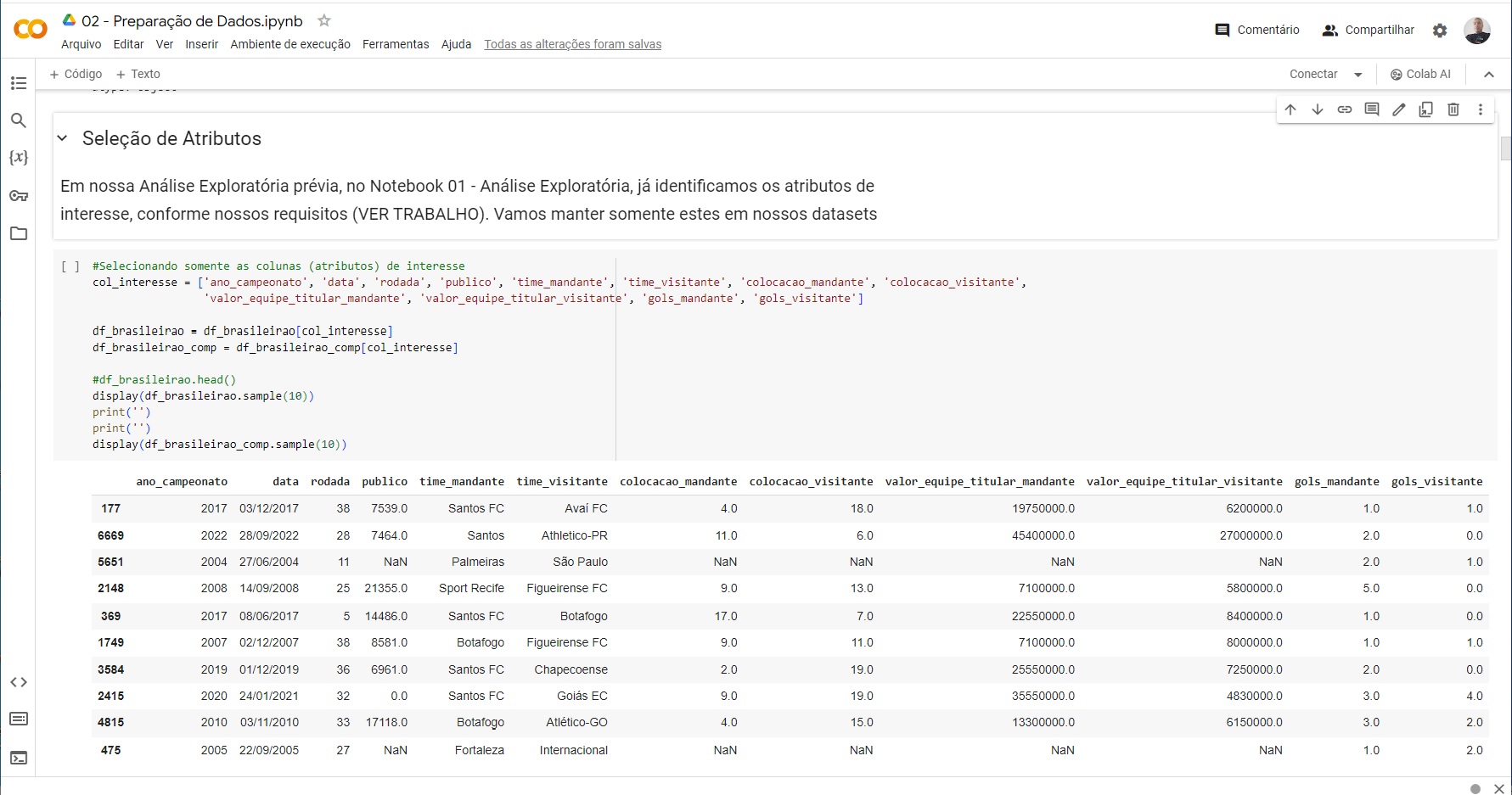


FIG. XX – Redução da Dimensionalidade por seleção de atributos

## 3.4. Ajustes de Valores e Tipos de Dados

Analisando as tabelas de frequência dos atributos **time\_mandante** e **time\_visitante**, percebemos ainda na análise exploratória, conforme descrito no item 3.2, que alguns times aparecem registrados com 2 nomes diferentes nas suas diversas ocorrências na base de dados.

Dessa forma, procedemos com a padronização desses nomes conforme código exibido na figura XX. Destaca-se que o procedimento foi necessário somente no *dataframe* **df\_brasileirao**, já que o *dataframe* **df\_brasileirao\_comp** possuem apenas os dados complementares, originalmente preenchidos por nós já com os nomes padronizados, conforme já explicado.



FIG. XX – Padronização dos valores de nomes dos times

Após o ajustes dos valores, verificamos se os tipos de dados dos atributos foram reconhecidos adequadamente pelo python (pandas), através do comando *dtypes*, conforme mostrado na figura XX.

Observa-se que apenas o atributo **data** não teve o seu tipo de dados corretamente reconhecido. Transformamos então para o tipo *datetime*, o que nos permitirá realizar operações de data com este atributo nas etapas seguintes.

|  |  |
| --- | --- |
|  |  |

FIG. XX – Visualização dos tipos de dados reconhecidos e transformação do tipo data

## 3.5. Análise de Valores Vazios

Um problema muito frequente em conjuntos de dados é a ausência de valores para alguns de seus atributos. A ausência de valores também é conhecida como dados faltantes, dados faltosos ou dados incompletos [X.Carvalho, André].

No entanto, grande parte dos algoritmos de *machine learning* foram projetados assumindo-se dados completos nos conjuntos de dados a serem trabalhados, conforme citado em [X.Carvalho, André] e [9.Zárate, Luis].

No nosso projeto, em nossa análise exploratória prévia, já foi possível perceber a existência de valores vazios (nulos) em atributos em algumas linhas dos nossos *datasets*. Chegou o momento de realizar uma análise mais detalhada e o tratamento necessário para cada caso, optando por alguma estratégia para eliminação dos valores nulos dos nossos *datasets*, sendo as principais citadas na literatura especializada, as seguintes:

- preenchimento dos valores ausentes com alguma estimativa estatística (média, moda, mínimo, máximo, etc);

- exclusão das linhas que apresentam algum atributo com valor nulo;

- exclusão do próprio atributo (coluna) do *dataset*.

Vamos tratar separadamente nossos dois *datasets*, e em seguida iremos unir ambos em um só *dataframe*, antes de prosseguir com as próximas sub-etapas da preparação de dados.

## 3.5.1 Tratamento de Valores Vazios no *dataframe* df\_brasileirao\_comp

Nosso *dataset* menor, está contido no *dataframe* **df\_brasileirao\_comp**. Conforme já explicado anteriormente, este *dataset* foi preenchido por nós, com o objetivo de complementar os dados não disponibilizados no *dataset* principal (obtido em [<https://basedosdados.org/dataset/c861330e-bca2-474d-9073-bc70744a1b23?table=18835b0d-233e-4857-b454-1fa34a81b4fa>]).

Dessa forma, já temos conhecimento da ausência de valores nos atributos **valor\_equipe\_titular\_mandante** e **valor\_equipe\_titular\_visitante**, pois estes dados não estavam disponíveis para preenchimento. Na figura xx, a seguir, mostramos o código onde visualizamos, para confirmação, a quantidade de dados nulos em cada atributo do nosso *dataset* menor, e em seguida exibimos as primeiras linhas do *dataframe*. Percebe-se que nosso *dataset* possui 91 linhas e 12 (colunas) atributos, todas com os atributos citados nulos.

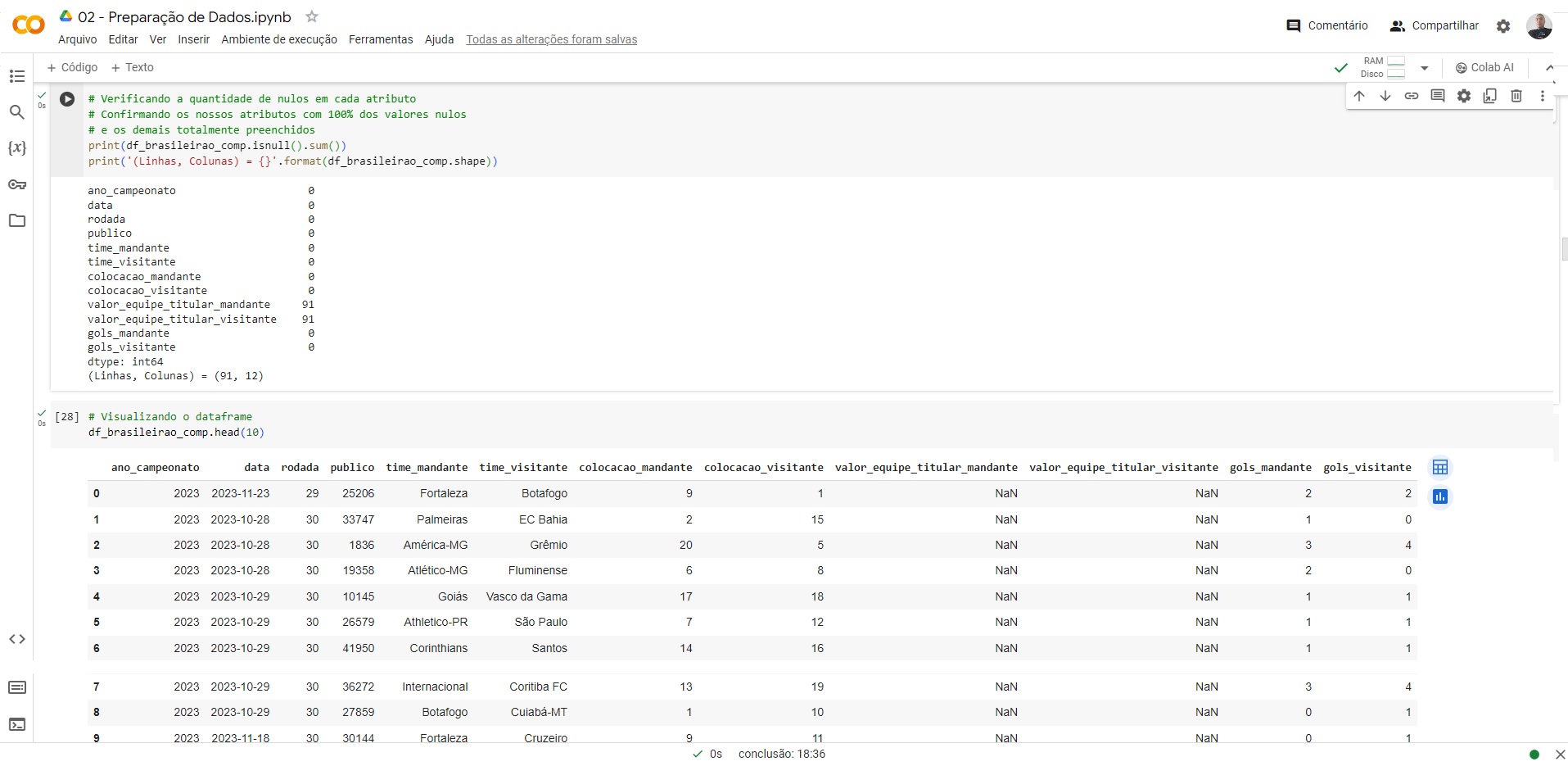


FIG. XX – Visualização dos dados nulos do *dataset* menor

A estratégia que entendemos adequada para preenchimento desses valores após realização de testes (disponíveis no notebook completo **02- Preparação de Dados** – COLOCAR LINK), foi preenchê-los com a média de cada clube especificamente, calculando essa média a partir dos valores do ano de 2023 disponíveis no *dataset* maior, já que os valores a serem preenchidos no *dataset* menor, correspondem somente aos jogos finais do ano de 2023. No nosso código python, fazemos isso em um laço ***for***, onde iteramos no *dataset* menor, calculando a média dos times mandante e visitante, e preenchendo os valores nulos. Em seguida visualizamos o *dataframe* com os valores já preenchidos, conforme mostrado na figura XX, a seguir.

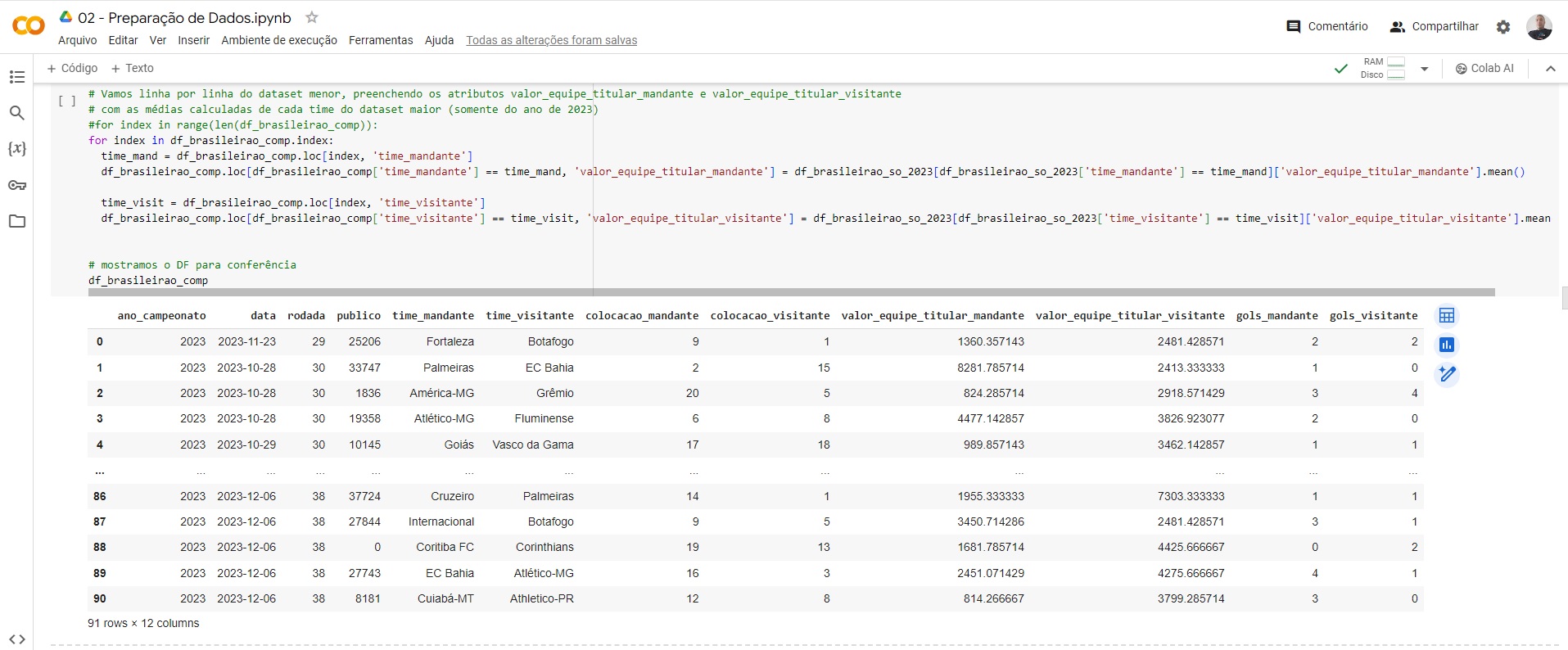


FIG. XX – Preenchimento dos dados nulos do *dataset* menor

## 3.5.2 Tratamento de Valores Vazios no *dataframe* df\_brasileirao

Nosso *dataset* maior contém os jogos do campeonato brasileiro de futebol de 29/03/2003 a 26/10/2023, obtidos em [<https://basedosdados.org/dataset/c861330e-bca2-474d-9073-bc70744a1b23?table=18835b0d-233e-4857-b454-1fa34a81b4fa>].

Este *dataset* necessita de um maior esforço para identificação e tratamento dos valores nulos. Iniciamos visualizando as quantidades de valores nulos em cada atributo, bem como exibindo algumas linhas do *dataframe*, para observação, conforme pode ser visto na figura XX. Observa-se que o dataframe possui 8079 linhas e 12 colunas (atributos), e apresenta uma quantidade considerável de valores nulos em alguns atributos.

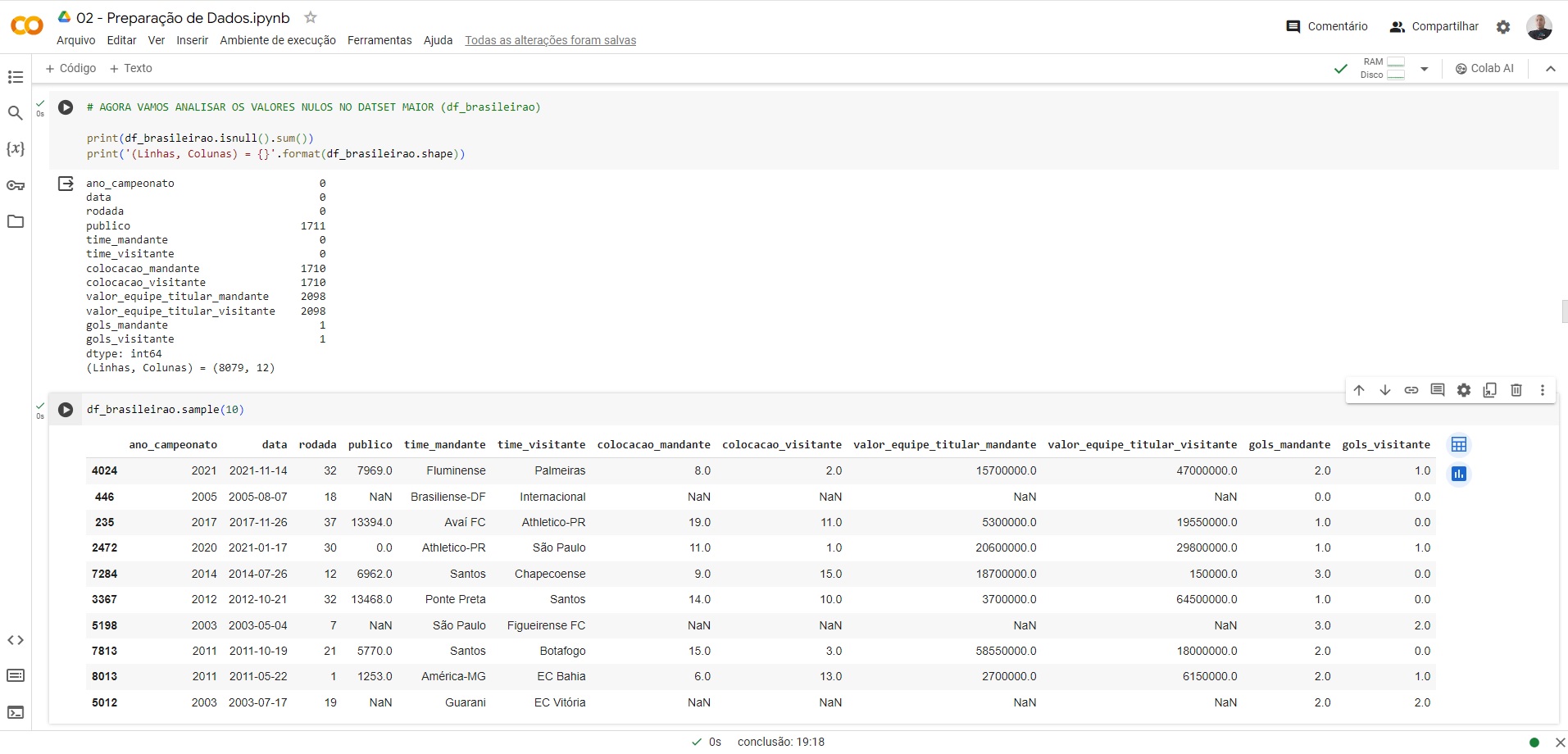


FIG. XX – Visualização dos dados nulos do *dataset* maior

Com a utilização da biblioteca **mathplotlib**, construímos um gráfico de barras empilhadas, para uma melhor noção da quantidade de valores nulos, inclusive em termos percentuais, em cada atributo, como vemos na figura XX, a seguir.

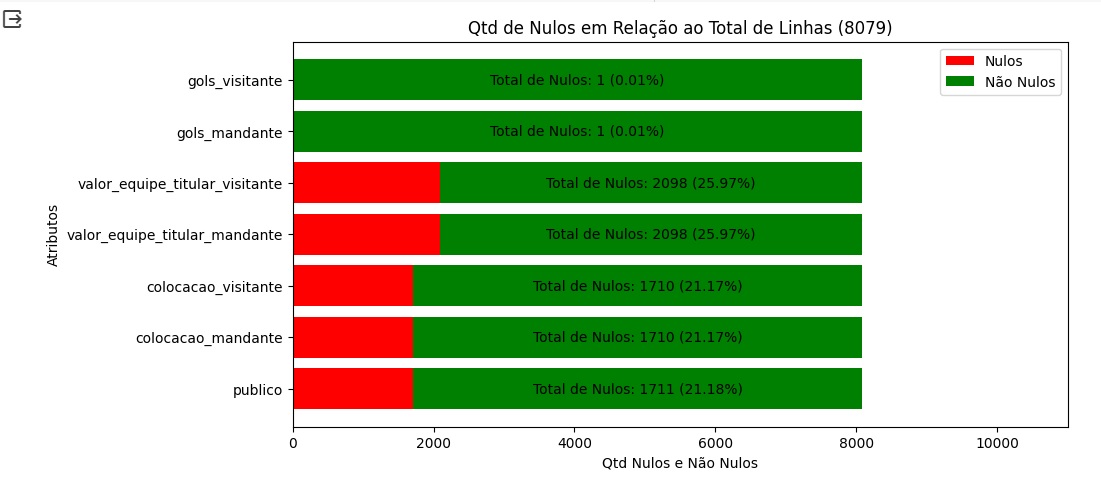


FIG. XX – Visualização gráfica das quantidades e percentuais de valores nulos por atributo

Então vamos analisar os valores nulos em cada atributo, definindo a estratégia a ser usada para cada caso.

**Valores nulos nos atributos “gols\_mandante” e “gols\_visitante”**

Observa-se no gráfico da figura XX, que esses atributos possuem apenas uma linha onde os valores são nulos. Exibimos essa linha para análise, e verificamos que se trata de um jogo da última rodada do campeonato brasileiro 2016, entre Chapecoense e Atlético-MG. Este jogo não aconteceu, em virtude do trágico acidente aéreo ocorrido com o time da Chapecoense. Então não resta outra estratégia a ser utilizada senão a remoção desta linha do nosso *dataset*. Fazemos isso com o comando **dropna()**, da classe ***Dataframe***, conforme mostrado na figura XX.

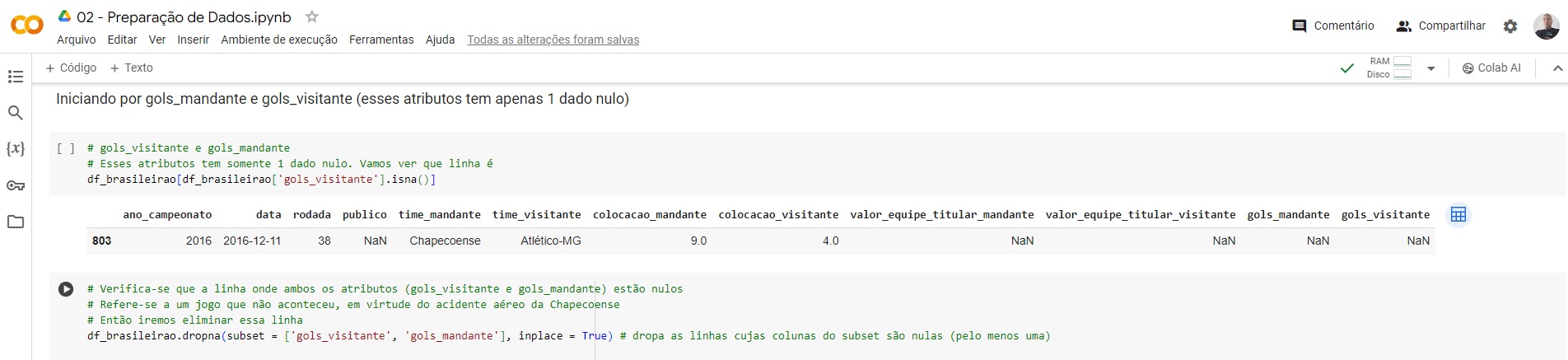


FIG. XX – Análise e remoção de linha do *dataframe*

**Valores nulos nos atributos ‘valor\_equipe\_titular\_mandante’ e ‘valor\_equipe\_titular\_visitante’**

Olhando para esses atributos na figura xx, verifica-se que cada um encontra-se com valor ausente em 2098 linhas, correspondendo a quase 26% dos dados de nosso *dataset*. No entanto, como já excluímos uma linha no passo anterior (figura xx), a qual também possuía estes atributos nulos, então ficamos agora com 2097 linhas com esses valores ausentes, como vemos na figura xx. Por meio da consulta exibida na figura xx, constatamos ainda que estas linhas são coincidentes para os dois atributos, ou seja, sempre que um é nulo o outro também é. Para realização dessa consulta, utilizamos o operador lógico **or** (símbolo **|**)

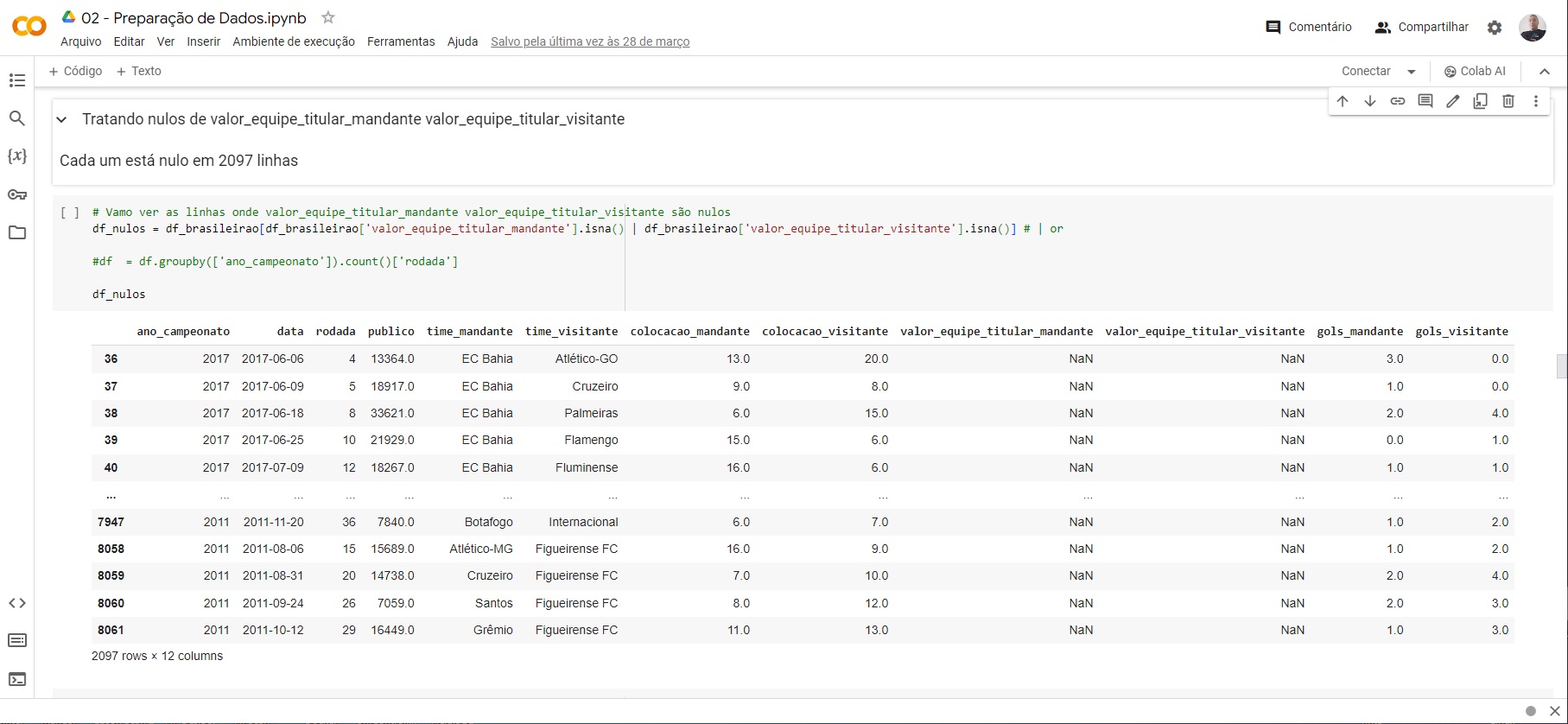


FIG. XX – Exibindo as linhas com dados nulos nos atributos **valor\_equipe\_titular\_mandante** e **valor\_equipe\_titular\_visitante**

Continuando com nossa análise, vamos verificar a quantidade de nulos por ano, utilizando o comando **groupby** da classe *Dataframe*. Conforme mostrado na figura xx, organizamos os dados da consulta em um *dataframe* (df\_count\_nulos), para facilitar a observação.

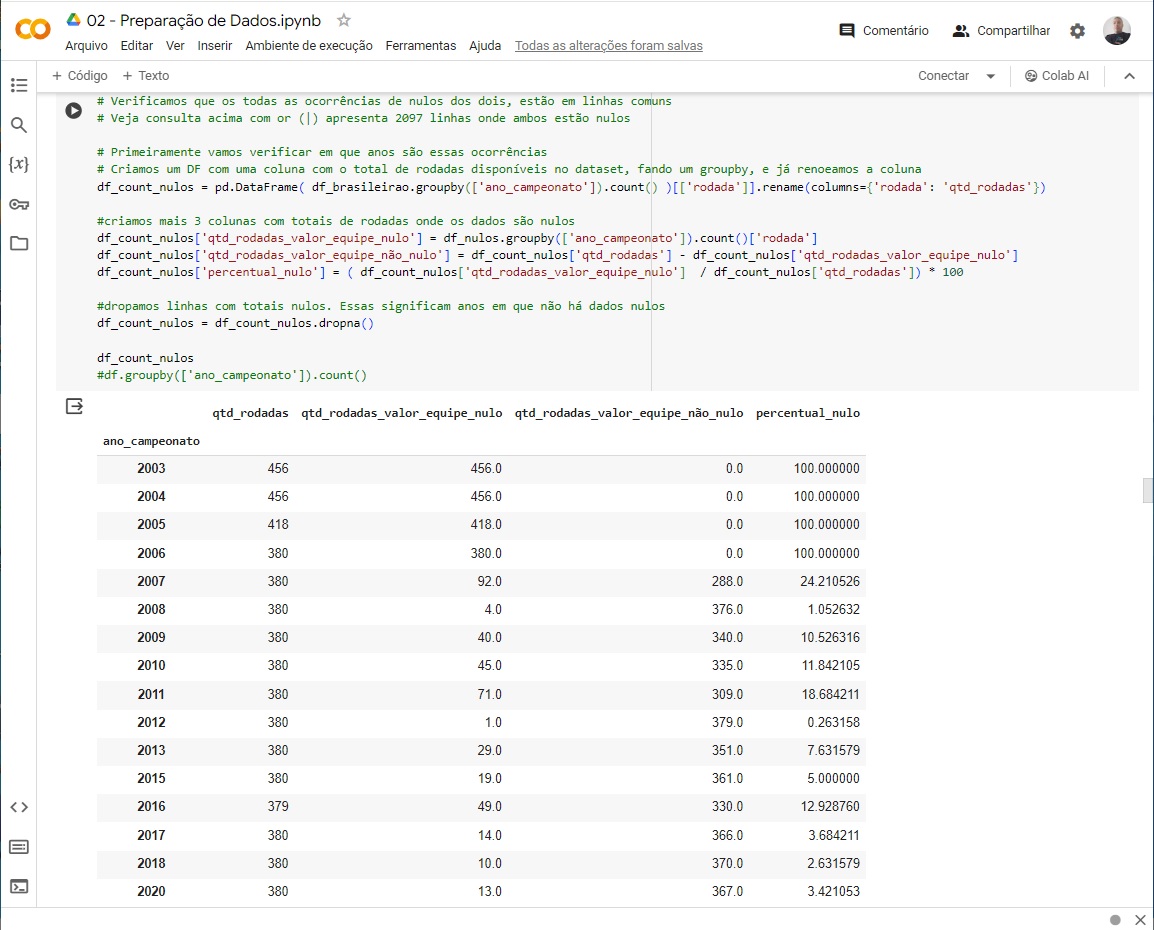


FIG. XX – Quantidade de linhas com valor nulo nos atributos dados nulos nos atributos **valor\_equipe\_titular\_mandante** e **valor\_equipe\_titular\_visitante**, por ano

Observamos que os anos 2003, 2004, 2005 e 2006 possuem 100% das linhas com dados nulos nos atributos considerados. Assim, o preenchimento desses dados deverá ser um pouco mais difícil, pois não temos outros valores de referência para realizar uma estimativa, podendo ter que se decidir até mesmo pela exclusão dessas linhas do nosso *dataset*.

Dessa forma, vamos iniciar pelo preenchimento dos demais anos, onde procederemos de forma análoga ao que já foi feito para o preenchimento desses mesmos atributos no *dataset* menor (*dataframe* df\_brasileirao\_comp), onde decidimos pela estratégia de se preencher esses valores para cada time, pela sua média no mesmo ano. Fazemos isso, com a utilização de dois laços do comando **for**aninhados, onde primeiramente iteramos em cada ano e em seguida, em cada time com jogos aquele ano. A cada iteração do **for** mais interno, nós calculamos a média dos valores e o preenchemos nas linhas onde ele está nulo. Mostramos o código pyhton correspondente, na figura xx, a seguir.

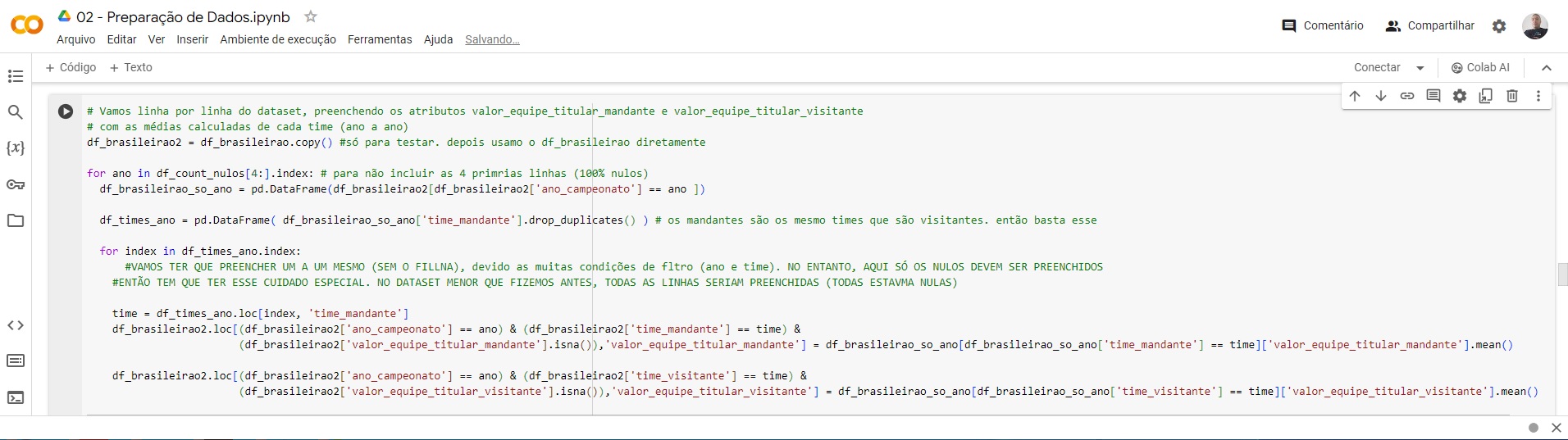


FIG. XX – Preenchimento de valores nulos nos atributos **valor\_equipe\_titular\_mandante** e **valor\_equipe\_titular\_visitante**

Após a execução do código mostrado na figura xx, repetimos a consulta agrupada por ano, para observação dos resultados, onde notamos que nos anos de 2007 e 2011 ainda restaram 38 linhas com dados nulos nesses atributos, conforme mostrado na figura xx, a seguir.

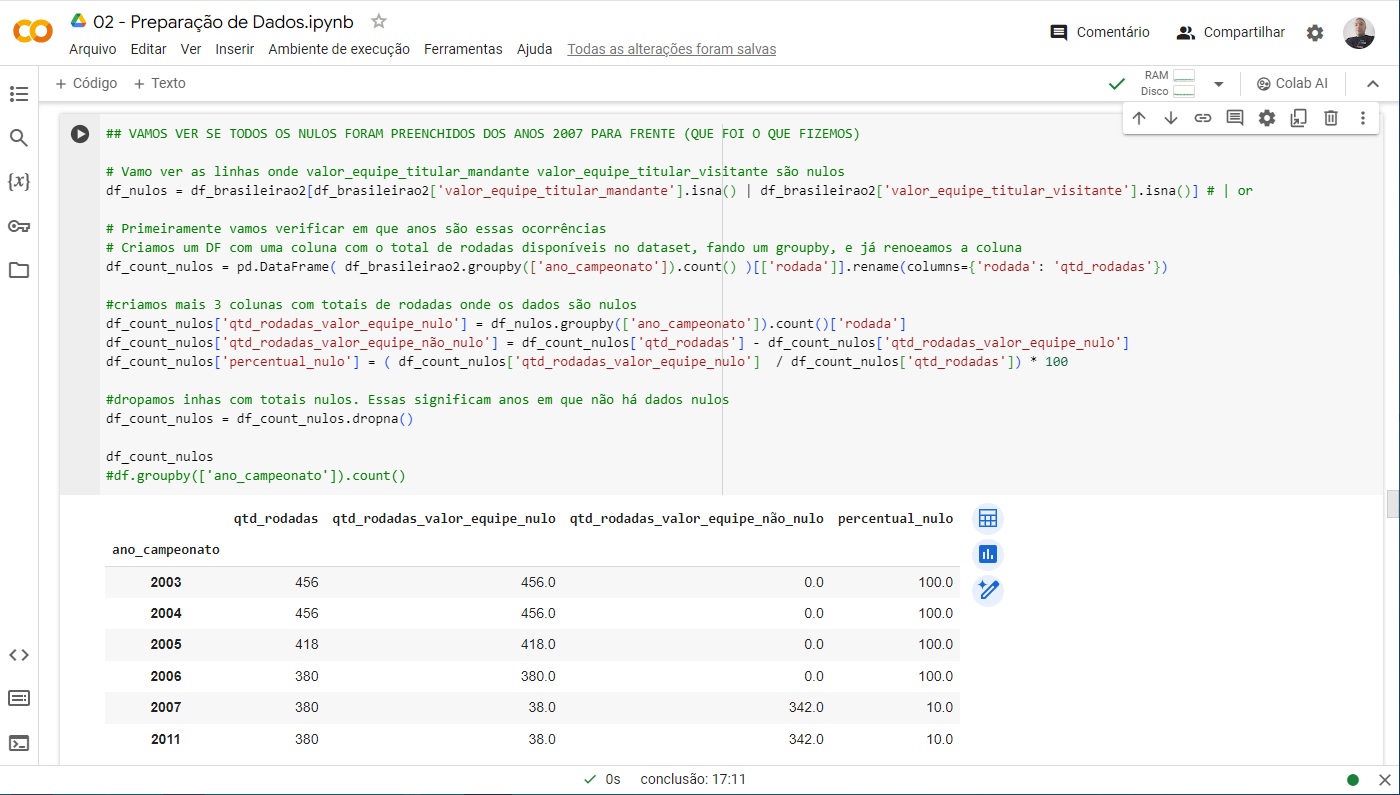


FIG. XX – Verificação do resultado do preenchimento dos valores nulos

Analisando os anos de 2007 e 2011, para verificar o motivo de não ter sido realizado o preenchimentos das 38 linhas restantes em cada ano, verificamos que nos dois casos, se trata dos jogos envolvendo os times do América-RN (em 2007) e do Figueirense FC (em 2011). Os valores não foram preenchidos devido ao fato desses times terem 100% de dados nulos nesses atributos, o que impossibilitou o cálculo da média para preenchimento.

Assim, após análise de cada caso, e consulta à tabela desses campeonatos no site [[https://www.srgoool.com.br](https://www.srgoool.com.br/)], optamos pelo preenchimento dos valores, utilizando estratégias um pouco diferentes para cada ano, como segue:

**- Ano de 2007**: o time América-RN foi o último colocado no campeonato, e também intuímos subjetivamente, que o time possui valor pequeno (provavelmente o menor dentre todos os times daquele campeonato). Assim, entendemos como uma boa estratégia, o preenchimento desses dados com o menor valor encontrado dentre os times daquele ano. Na figura xx, mostramos o código utilizado para o preenchimento desses valores, pelo mínimo valor encontrado.

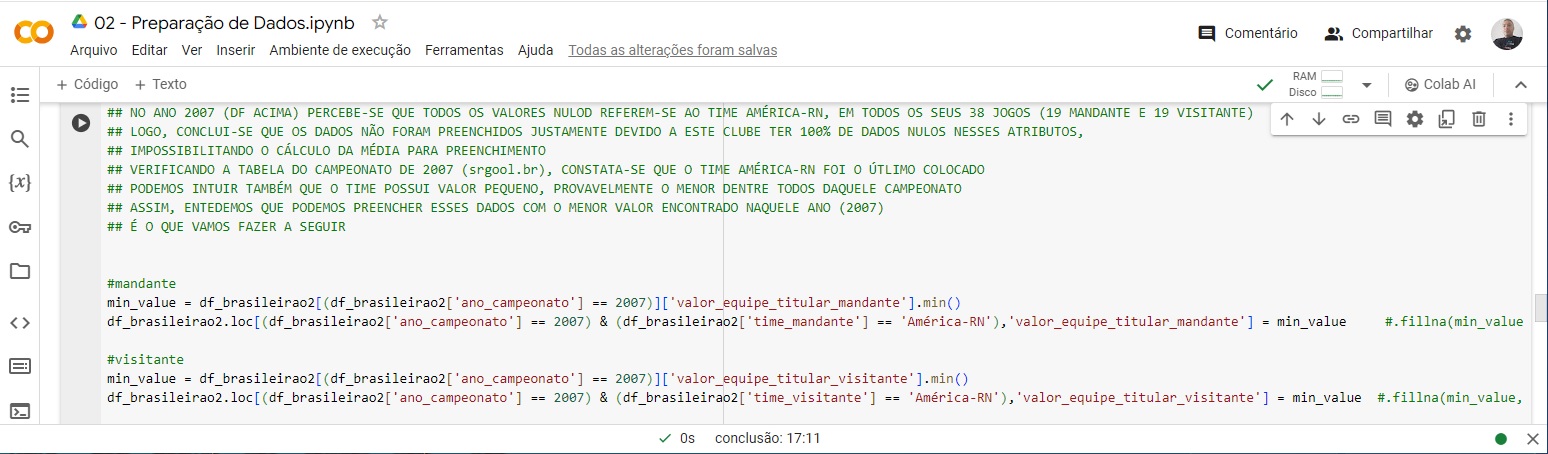


FIG. XX – Preenchimento dos valores nulos restantes do ano de 2007

**- Ano de 2011**: o time Figueirense FC foi o sétimo colocado no campeonato. Observando as médias de valor de cada time do campeonato, percebe-se que os times considerados grandes do futebol brasileiro, possuem de fato os maiores valores, enquanto que aqueles times não considerados grandes, concentram os 6 menores valores daquele ano, conforme mostramos na figura xx, com o resultado de consultas das médias desses atributos para cada time, ordenadas de forma crescente.

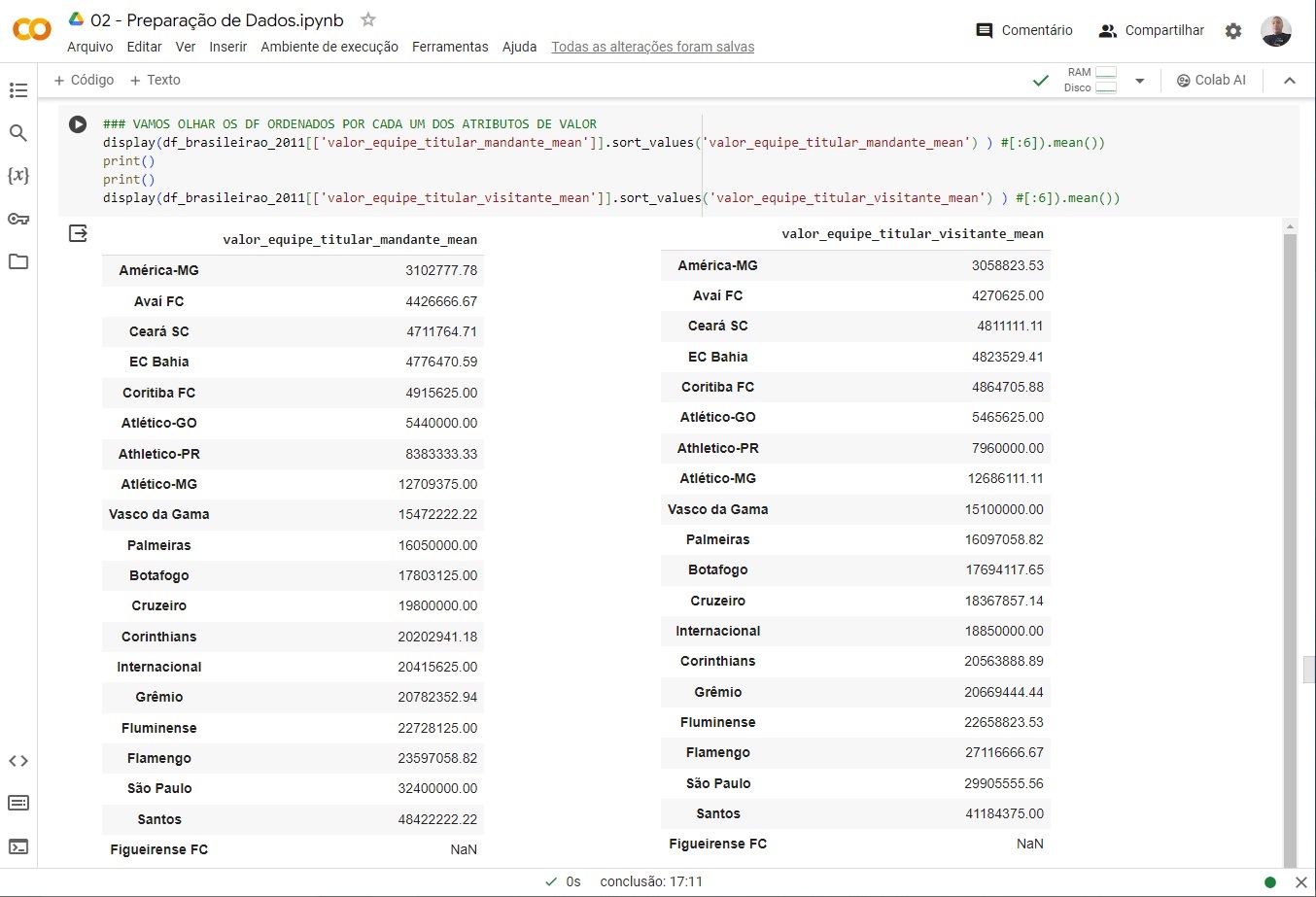


FIG. XX – Exibição das médias de valores de cada time do ano de 2011

Nesse caso, consideramos como uma boa estratégia de se estimar o valor para o preenchimento desses dados, com uma média dentre os times não considerados grandes daquele ano, que representam as 6 primeiras linhas dos resultados das nossas consultas mostradas na figura xx. O nosso código para o preenchimento dos valores é mostrado na figura xx, a seguir.



FIG. XX – Preenchimento dos valores nulos restantes do ano de 2011

Após a execução dos códigos mostrados nas figura xx e xx, repetimos novamente a consulta agrupada por ano, para observação dos resultados, onde observamos que os únicos valores nulos que restaram nesses atributos se referem aos anos de 2003, 2004, 2005 e 2006, conforme mostrado na figura xx, a seguir.

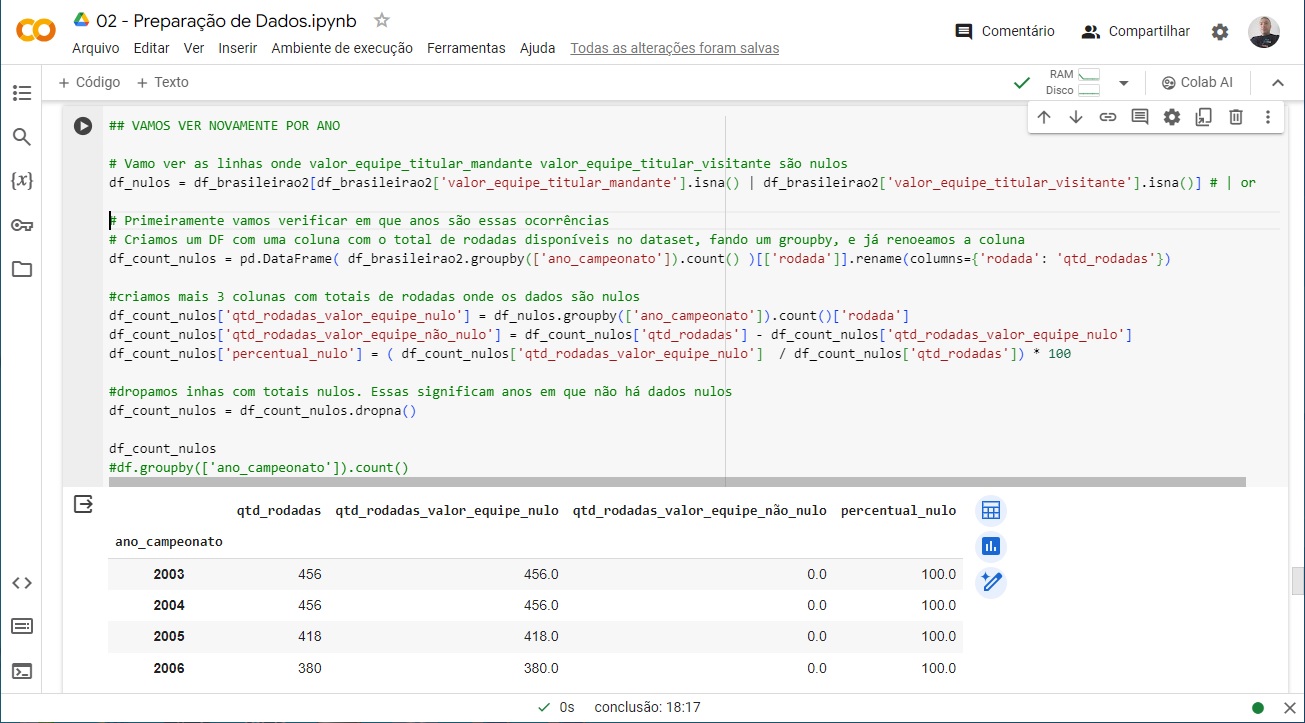


FIG. XX – Verificação do resultado do preenchimento dos valores nulos restantes

Já observamos antes que em relação a estes anos (2003, 2004, 2005 e 2006), 100% dos valores dos atributos *valor\_equipe\_titular\_mandante* e *valor\_equipe\_titular\_visitante* são nulos, o que nos deixa sem nenhuma referência aceitável para cálculo de estimativa de valor para preenchimento. Na figura xx apresentamos um gráfico de barras empilhadas, para uma melhor noção da quantidade de valores nulos em relação ao total de dados no nosso *dataset*, inclusive em termos percentuais, em cada atributo.

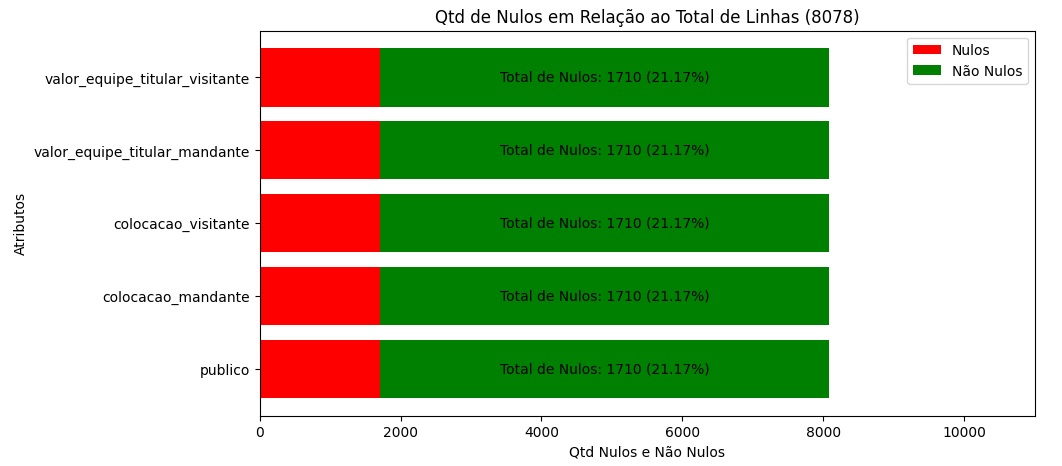


FIG. XX – Verificação do resultado do preenchimento dos valores nulos

O gráfico nos mostra que as 1710 linhas com valores nulos nos atributos de valor da equipe, representam 21,17% das linhas do nosso *dataset*, o que não é um valor desprezível. No entanto, como observado antes, não temos uma referência aceitável para calcular um valor estimado para preenchimento dos valores. Dessa forma ficamos sem alternativa senão remover essas linhas do nosso *dataset*.

Então excluímos as linhas que apresentam valores nulos nos atributos *valor\_equipe\_titular\_mandante* e *valor\_equipe\_titular\_visitante*, e em seguida verificamos novamente as quantidades de valores nulos restantes em todos os atributos, conforme mostrado na figura XX.

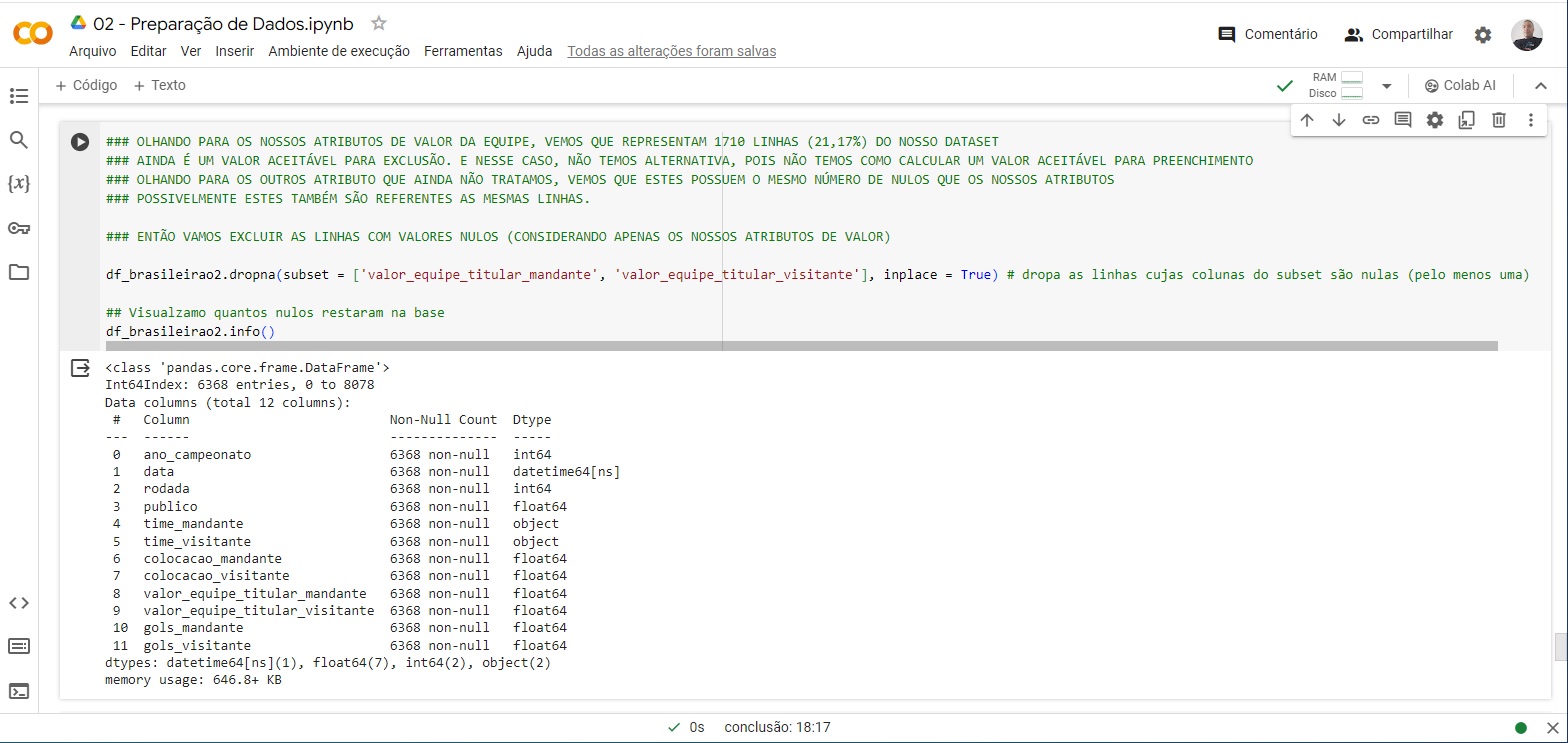


FIG. XX – Verificação do resultado do preenchimento dos valores nulos

Verifica-se que, após a exclusão, o *dataset* restou com 6386 linhas, e nenhum valor nulo em nenhum atributo. Isso significa que os atributos que ainda não tínhamos tratado, possuíam ocorrência de nulos nas mesmas linhas que excluímos.

Dessa forma, concluímos o tratamento de dados nulos nos nossos 2 *dataframes* (df\_brasileirao e df\_brasileirao\_comp). Então concatenamos os dois em um único *dataframe*, e em seguida exibimos algumas linhas do mesmo, para verificação, conforme mostrado na figura xx.

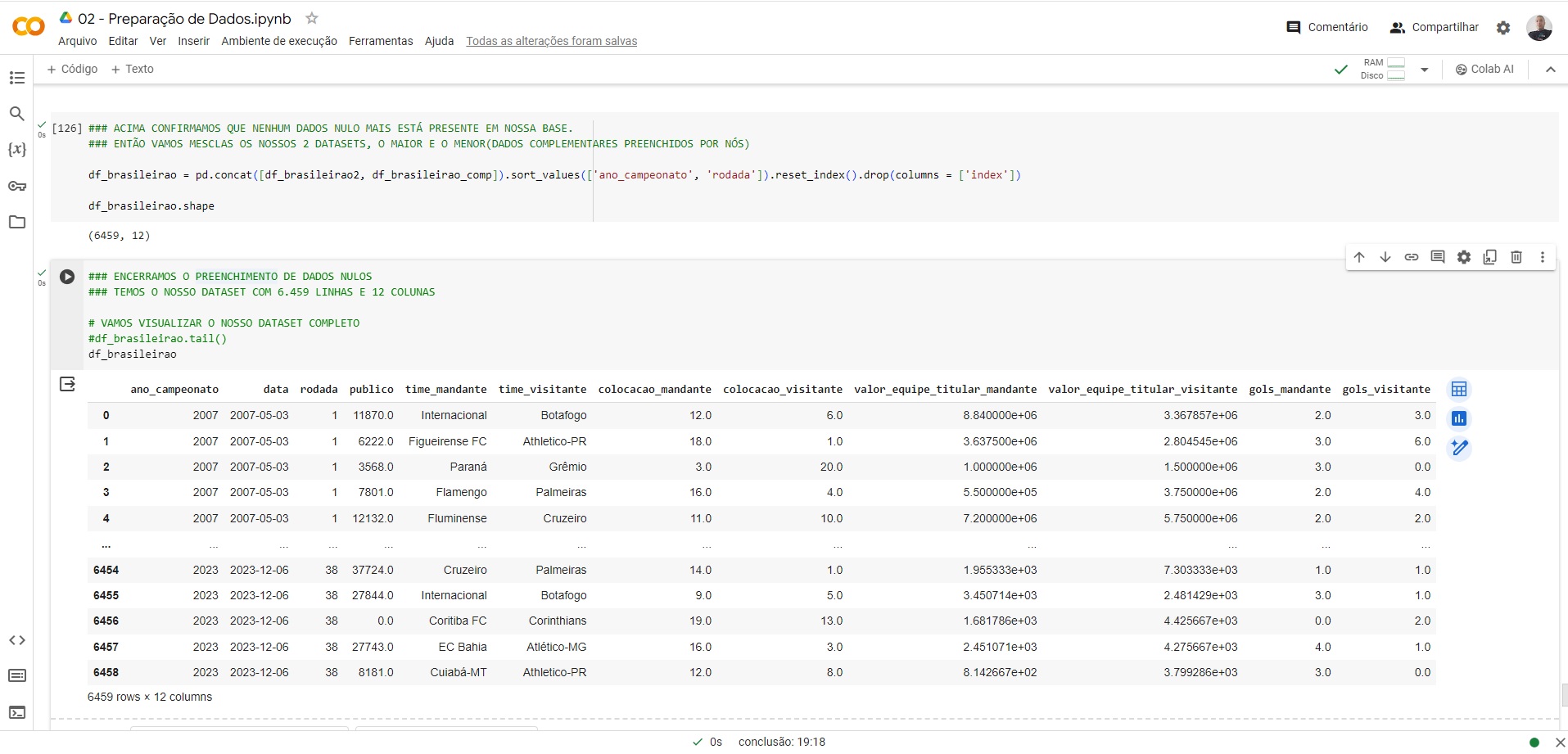


FIG. XX – Concatenação dos *dataframes* e exibição do resultado

Observa-se que nosso *dataframe* unificado restou com 6459 linhas e 12 colunas.

## 3.6. Análise de Correlação

A correlação entre dois atributos ilustra a força da relação linear entre eles [X.Carvalho, André]. O valor da correlação entre dois atributos é um conceito da estatística, e pode ser numericamente calculado, variando de -1 a 1. Valores próximos de 0 (zero) indicam correlação fraca, e valores próximos do extremos (-1 e 1) indicam correlação forte (negativa ou positiva).

Um par de atributos onde se observa uma correlação forte entre seus valores, pode indicar uma dependência ou mesmo redundância entre eles. E uma boa prática para se simplificar e melhorar a qualidade do *dataset* (e consequentemente do futuro modelo de *machine learning*), é a exclusão de um dos atributos do par fortemente correlacionado.

No nosso projeto, obtivemos a matriz de correlação entre as variáveis numéricas do nosso conjunto de dados por meio do método *corr()* da classe *DataFrame* da biblioteca *Pandas*. Além disso, para uma melhor visualização, utilizamos as bibliotecas *Seaborn* e *Matplotlib,* para exibir esses valores graficamente, através de um mapa de calor, conforme ilustrado na figura xx, a seguir.

Pelo gráfico, foi possível constatar que não existe nenhuma dupla de variáveis com correlação forte (próxima de 1 ou -1). Caso existisse, provavelmente eliminaríamos uma das variáveis da dupla (após análise cuidadosa), o que seria uma forma de redução da dimensionalidade do conjunto de dados.

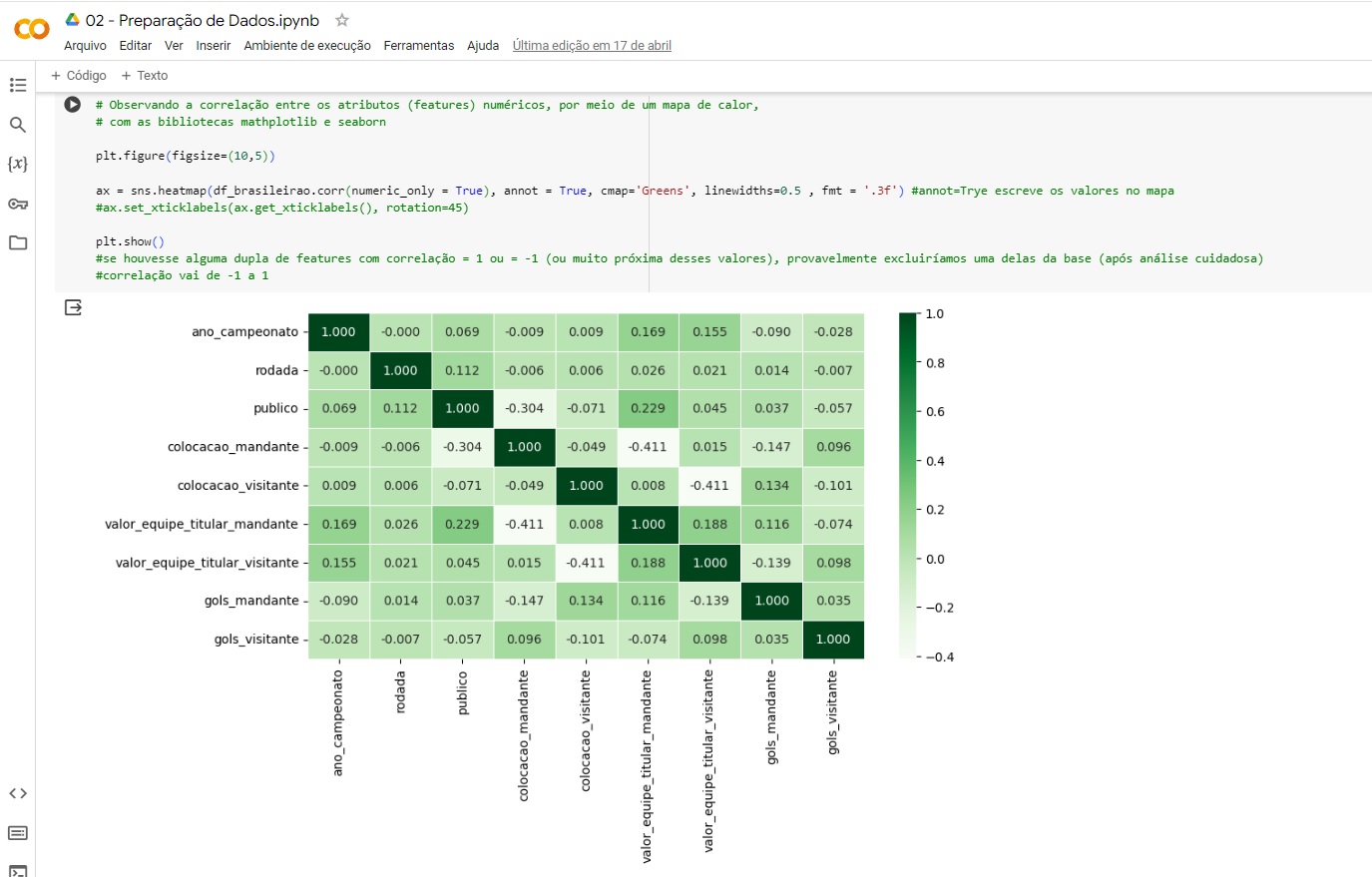


FIG. XX – Matriz de correlação dos atributos numéricos, representada em um mapa de calor

## 3.6. Análise de *Outliers*

Outra tarefa importante para a melhoria da qualidade do conjunto de dados é a identificação de valores atípicos, anomalias ou *outliers*, que destoam em comparação com a maioria dos valores de um atributo [X.Carvalho, André].

*Outliers*, então, são valores que estão além dos limites aceitáveis, ou são muito diferentes dos demais valores observados para o mesmo atributo [2.Sarkar, Dipanjan].

A ocorrência de *outliers*, não necessariamente significa um problema. Por isso deve ser feita uma análise cuidadosa ao se identificar um *outlier*, o qual pode ser um ruído (um dado mal preenchido), mas pode também ser um valor real. No caso de se constatar que um *outlier* representa mesmo um valor incorreto, alguns tratamentos podem ser realizados, como por exemplo:

- Remoção da linha com o *outlier* do conjunto de dados;

- Imputação de um novo valor (por exemplo média ou mediana)

Conforme citado em [7.Fujii, Juliene], a identificação de *outliers* pode ser realizada através do cálculo do limite inferior e superior, baseado na amplitude da amostra (distância entre os quartis 1 e 3). E um gráfico bem adequado para visualização é o *boxplot*.

No nosso projeto, para a identificação e análise de *outliers*, definimos funções para cálculo dos limites inferior e superior; e para exclusão de linhas com *outliers*, conforme mostrado na figura xx.

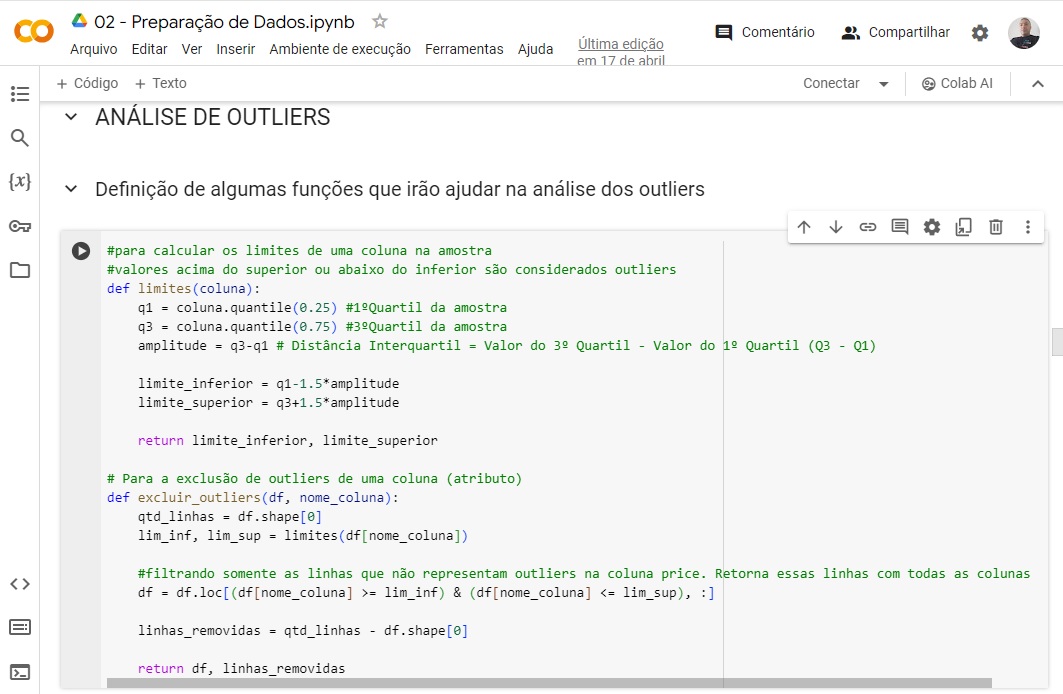


FIG. XX – Código python de definição de funções úteis para análise de *outliers*

Em seguida, para cada atributo numérico do nosso *dataset*, plotamos um *boxplot* para propiciar análise. O resultado das plotagens, é exibido na figura xx.

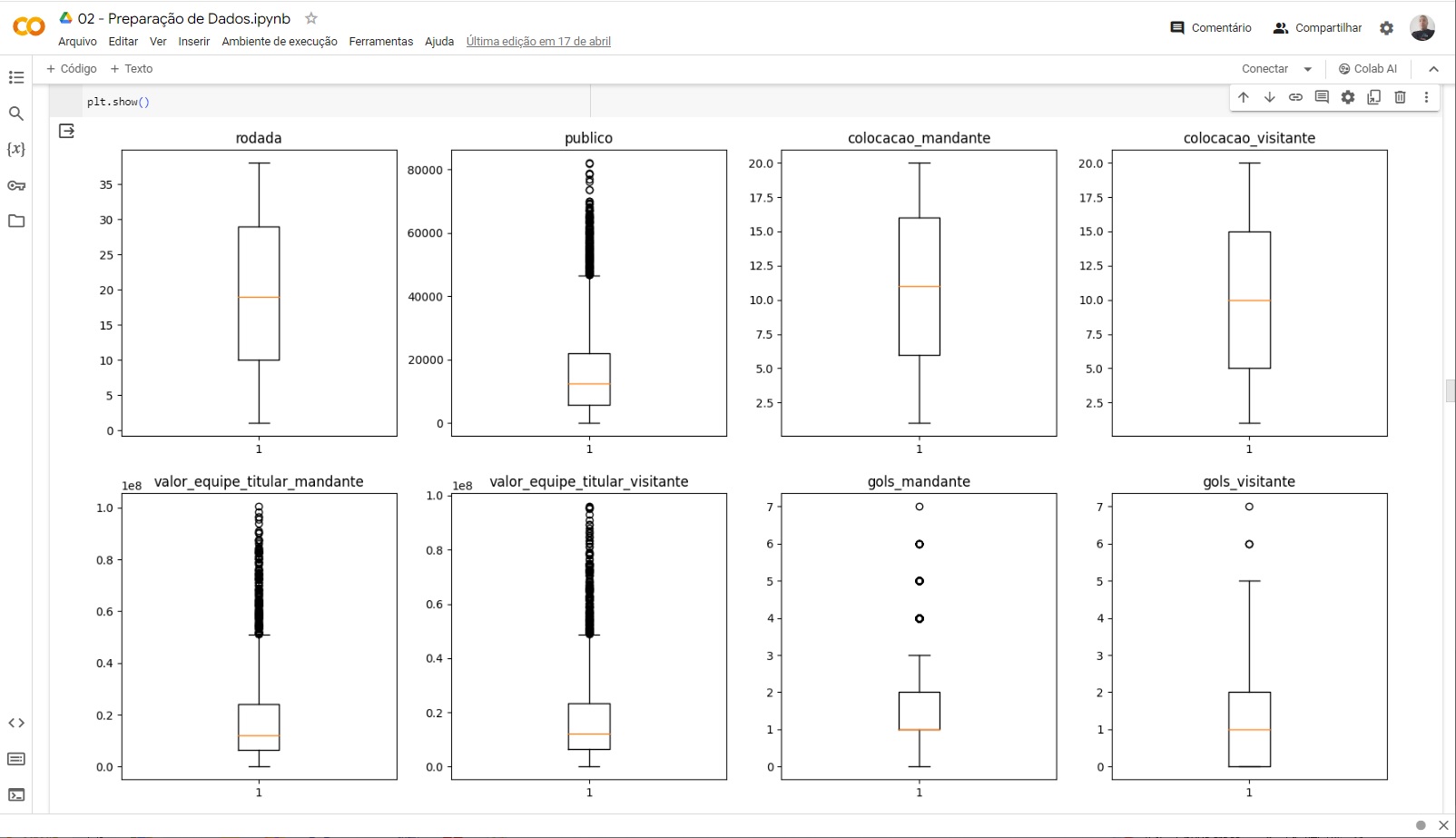


FIG. XX – Gráficos de *boxplot* para análise de outliers dos atributos numéricos

Pelos *boxplots*, já visualizamos a presença de *outliers* nos atributos **publico, valor\_equipe\_titular\_mandante, valor\_equipe\_titular\_visitante, gols\_mandante** e **gols\_visitante**, os quais apresentam valores fora dos limites.

Para uma visão mais detalhada das quantidades e percentuais de outliers de cada atributos, plotamos um gráfico de barras, mostrado na figura xx.

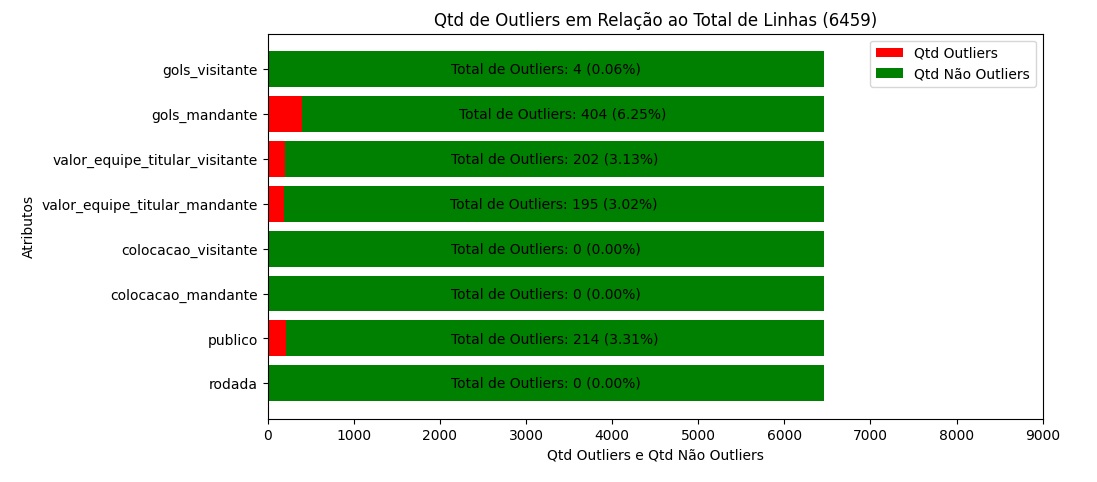


FIG. XX – Quantidade e percentual de outliers por atributo numérico

Em seguida, analisamos com mais detalhes cada atributo individualmente, conforme a seguir:

- **gols\_mandante** e **gols\_visitante**: percebe-se que os *outliers* representam casos de grandes goleadas, de 4 a 7 gols, ocorridos com o certa raridade. No entanto representam dados reais;

- **valor\_equipe\_titular\_mandante** e **valor\_equipe\_titular\_visitante**: para esses atributos, tivemos que fazer uma análise ano a ano, uma vez que como já citamos antes, existem diferenças nas escalas de valores em determinados anos. Também aqui concluímos que os *outliers* representam valores reais, devido a diferenças grandes de investimentos de alguns clubes;

- **publico**: nesse atributo, percebemos que pelos cálculos dos limites da amostra, os valores acima de 45.000 foram considerados *outliers*, no entanto aqui também são valores reais que representam jogos com grandes públicos, que de fato ocorreram.

Dessa forma, após as análises, concluímos pela não exclusão de nenhum *outlier*, pois todos se tratam de valores corretos. No entanto, em relação ao atributo publico (que srá o nosso atributo alvo, que deverá ser predito para dados novos), percebemos a ocorrência de valores zerados, que provavelmente devem ser jogos que ocorreram sem a presença de público, por punição ou outros fatores (como a pandemia do coronavírus, por exemplo). Estes jogos, deveremos excluir do dataset, pois não temos como estimá-lo de forma confiável. Entretanto, iremos realizar esta exclusão somente após a etapa de engenharia de atributos, onde precisaremos dos placares desses jogos para a criação de novos atributos, como veremos a seguir.

Importante também destacar que, em nossa análise utilizamos de outros gráficos e recursos mais detalhados, que podem ser consultados no notebook python da preparação de dados, **02 - Preparação de Dados.ipynb**, que pode ser acessado em [xxxxxxxxxx].

## 3.7. Engenharia de Atributos (feature enginering)

## (Criação de Novos Atributos)

Esse processo de preparação e ajustes dos dados, e de seleção dos atributos (*features*) para a análise, denomina-se ***feature selection*** ou ***feature engineering***, devendo se seguir alguns princípios na sua realização, conforme descrito em [6.De Paula, Hugo].

## 3.8 Categorização de variáveis numéricas

## Transformação de variáveis numéricas em categóricas

## 3.9 Agrupamento de categorias (variáveis categóricas)

## 3.10 Nova Seleção de Atributos

## 3.11. Padronização/Normalização de Variáveis

**(ajuste de escalas)**

## 3.12. Codificação de Variáveis Categóricas

# 4. Escolha do Modelo

(BOTAR GRÁFICO DA RETA DE REGRESSÃO DO MODELO ESCOLHIDO, PARA DADOS DE TREINAMENTO E DE TESTES – UM PRA CADA)

Xxxxxxxx

Em CD, o tempo gasto é muito maior na preparação dos dados do que na modelagem em si,

# pois os modelos/algorítmos estão todos prontos nas bibliotecas, tendo somente que ter a poposta definida

# para aplicação. Mas não vamos gastar tempo escrevendo algorítmos.

# Mas temos também que selecionar bem os atributos que serão parâmetros de entrada

The most important arguments in RandomizedSearchCV are n\_iter, which controls the number of different combinations to try, and cv which is the number of folds to use for cross validation (we use 100 and 3 respectively). More iterations will cover a wider search space and more cv folds reduces the chances of overfitting, but raising each will increase the run time. Machine learning is a field of trade-offs, and performance vs time is one of the most fundamental.

# 5. Implantação da Solução em Ambiente de Produção

# 6. Links

Os artefatos produzidos no desenvolvimento deste trabalho, bem como os *scripts*, *Datasets* e demais arquivos relevantes, estão disponíveis para acesso no repositório do projeto, no link [Relatório Técnico – Módulo A](https://github.com/salomaofreitasjr/TCC-PROJETO-INTEGRADO---PUC-MINAS---ANALYTICS-E-BI/tree/main/Projeto%20Integrado%20-%20Produ%C3%A7%C3%A3o/M%C3%B3dulo%20A).

# REFERÊNCIAS

[1.Sharda, Ramesh] SHARDA, Ramesh; DELEN, Dursun; TURBAN, Efraim. **Business Intelligence e Análise de Dados para a Gestão do Negócio, 4ª Edição.** Porto Alegre: Bookman, 2019.

[2.Sarkar, Dipanjan] DIPANJAN, Sarkar; BALI, Raghav; SHARMA, Tushar. **Practical Machine Learning with Python.** Bangalore, Karnataka, India: Apress, 2018.

[3.Guido, Sarah] GUIDO, Sarah; MÜLLER, Andreas. **Introduction to Machine Learning with Python.** Sebastopol, CA, USA: O’Reilly, 2017.

[4.Faceli, Katti] FACELI, Katti; LORENA, Ana Carolina; GAMA, João; CARVALHO, André Carlos. **Inteligência Artificial, Uma Abordagem de Aprendizado de Máquina.** Rio de Janeiro: LTC, 2011.

[X.Carvalho, André] CARVALHO, André; MENEZES, Angelo; BONIDIA, Robson. **Ciência de Dados - Fundamentos e Aplicações.** Rio de Janeiro: LTC, 2024.

[X.Feltrin, Fernando] FELTRIN, Fernando. **Ciência de Dados e Aprendizado de Máquina.** 2019.

[5.Gomes, Rodrigo] GOMES, Rodrigo. **Notas de aula da Disciplina Introdução à Linguagem Python, do Curso de Pós-Graduação em Analytics e Business Intelligence da PUC Minas.** Belo Horizonte, 2021.

[6.Gomes, Rodrigo] GOMES, Rodrigo. **Notas de aula da Disciplina Programação para Ciência de Dados, do Curso de Pós-Graduação em Analytics e Business Intelligence da PUC Minas.** Belo Horizonte, 2021.

[7.Fujii, Juliene] FUJII, Juliene Borges. **Notas de aula da Disciplina Estatística Geral – Teoria e Aplicações, do Curso de Pós-Graduação em Analytics e Business Intelligence da PUC Minas.** Belo Horizonte, 2021.

[8.De Paula, Hugo] DE PAULA, Hugo. **Notas de aula da Disciplina Machine Learning, do Curso de Pós-Graduação em Analytics e Business Intelligence da PUC Minas.** Belo Horizonte, 2021.

[9.Zárate, Luis] ZÁRATE, Luis E. **Notas de aula da Disciplina Modelagem e Preparação de Dados para Aprendizado de Máquina, do Curso de Pós-Graduação em Ciência de Dados e Big Data da PUC Minas.** Belo Horizonte, 2023.

[10.Patrocínio, Zenilton] PATROCÍNIO, Zenilton K. G. **Notas de aula da Disciplina Redes Neurais e Aprendizagem Profunda, do Curso de Pós-Graduação em Ciência de Dados e Big Data da PUC Minas.** Belo Horizonte, 2023.

[11.Alves, Pedro] ALVES, Pedro. **Apostila do Curso Python para Data Science e Analytics – Módulo Avançado, Aula 3 – Árvore de Decisão [https://python-para-datascience-analytics.club.hotmart.com/].** PA Analytics [https://paanalytics.net/], 2022.

[curso flai]

[curso pyhton impressionador]

<https://medium.com/data-hackers/engenharia-de-features-transformando-dados-categ%C3%B3ricos-em-dados-num%C3%A9ricos-e5d3991df715>

https://towardsdatascience.com/hyperparameter-tuning-the-random-forest-in-python-using-scikit-learn-28d2aa77dd74

https://scikit-learn.org/stable/modules/generated/sklearn.model\_selection.GridSearchCV.html

<https://www.kaggle.com/code/juanmah/tactic-03-hyperparameter-optimization-xtra-trees>

<https://www.analyticsvidhya.com/blog/2016/02/complete-guide-parameter-tuning-gradient-boosting-gbm-python/>

<https://www.kaggle.com/code/juanmah/tactic-03-hyperparameter-optimization-bagging>

https://www.kaggle.com/code/prashant111/bagging-vs-boosting

https://www.geeksforgeeks.org/voting-regressor/

Documentção da biblioteca scikit-learn

<https://scikit-learn.org/stable/modules/classes.html>

(SUBSTITUIR PELAS FONTES DOS DATASETS DESTE PROJETO)

[16.Duque, Adão]. DUQUE, Adão. **Fontes de Dados do Campeonato Brasileiro de Futebol.** <https://www.kaggle.com/datasets/adaoduque/campeonato-brasileiro-de-futebol/metadata>. <https://github.com/adaoduque/Brasileirao_Dataset>. Acessado em 04/07/2022.