



**CURSO DE BACHARELADO EM SISTEMAS DE INFORMAÇÃO**  
**TRABALHO DE CONCLUSÃO DE CURSO - TCC**


Edmilson Vitorino Scovino da Silva

**FOLHA DE APROVAÇÃO**

**IMPACTO DE PRÉ-PROCESSAMENTO EM CONJUNTO DE DADOS**

Após a exposição dos discentes **Edmilson Vitorino Scovino da Silva**, sobre a realização do artigo, a Banca Examinadora, composta pelos professores abaixo identificados, reuniu-se e aprovou o presente artigo que, por atender aos requisitos estabelecidos, pelo curso de Bacharelado em Sistemas de Informação da Faculdade Metodista Granbery, para obtenção do título de Bacharel em Sistemas de Informação.

Juiz de Fora, 06 de Dezembro de 2022.

  
Ricardo Silva Campos – Orientador  
Faculdade Metodista Granbery

  
Marco Antônio Pereira de Araújo – Examinador  
Faculdade Metodista Granbery

**JUIZ DE FORA**  
**2022**

# Impacto de pré-processamento em conjunto de dados

## Impact of preprocessing on dataset

Edmilson Vitorino Scovino da Silva<sup>1</sup>

Ricardo Silva Campos<sup>2</sup>

### Resumo

Os dados cada vez mais têm se tornado importantes para sociedade moderna. Com os dados podemos realizar diferentes estudos sobre os dados de vendas, saúde entre outros. Para possuir esses dados é necessário coletá-los, mas os dados acabam sempre sendo recebidos com inconsistências ou possuindo ausência, isso pode afetar de forma considerável nossas análises e modelos de aprendizagem de máquina. Podemos resolver esses problemas utilizando tratamento de dados, com ele cada dado será tratado de maneira correta e além de dados ausentes deixaram de existir utilizando maneiras diferentes para chegar ao resultado final.

Palavras-chave: Python, Dados, Pandas, Limpeza de dados.

### Abstract

Data has become increasingly important to modern society. With the data we can carry out different studies on sales data, health, among others. In order to have this data, it is necessary to collect it, but the data always end up being received with inconsistencies or absence, this can considerably affect our analyzes and machine learning models. We can solve these problems using data treatment, with it each data will be treated correctly and in addition to missing data no longer exist using different ways to reach the final result.

Keywords: Python, Data, Pandas, Data Cleansing.

<sup>1</sup> Graduando em Sistemas de Informação – Faculdade Metodista Granbery.

<sup>2</sup> Professor Orientador

## 1 INTRODUÇÃO

Para tratar do assunto devemos compreender que a informação e os dados estão ligados entre si. Mas são bem diferentes. Os dados por si só não transmitem informação, são objetos que capturamos do mundo à nossa volta para podermos organizar em categorias distintas como números, palavras, imagens. Com eles agrupamos, ordenamos, dimensionamos e classificamos. Essas diversas opções geram a informação para compreensão humana. Assim podemos compreender fatos do mundo que não compreendemos (CARDOSO, 2003).

Os dados estão presentes na humanidade há muitos anos e eram utilizados para compreender o tempo, terrenos, plantações, animais, doenças, entre outras possibilidades. Com o passar dos anos, foi observado que podemos estudar as informações, dessa maneira observamos que ao realizar a organização das informações podem classificar em diferentes conjuntos, formas e objetos também serem armazenados para reutilização futura. Sendo cada vez mais necessário armazenar a informação, os dados tornaram-se de grande importância para poder estudar, criar, desenvolver e inovar o mundo. Atualmente possuímos os dados gerando informações na palma de nossa mão.

A utilização dos dados tornou-se de grande importância. Diferentes organizações públicas, privadas e pessoas têm utilizado os dados para realizar estudos de diferentes tipos. Como exemplo, pode-se utilizar os dados para se entender a necessidade de consumidores, melhoria de produtos e até realizar análises de logísticas para tomar melhores decisões. Esse processo inicia com a coleta de dados, após essa etapa chega para os desenvolvedores como dados brutos. Esses dados devem ser pré-processados de maneira que tragam sua melhor qualidade e atendam aos requisitos até chegarem aos modelos de aprendizagem de máquina e nas análises. Essa etapa de pré-processamento tem o foco em limpar os dados por inconsistências ocorridas na coleta, têm grande necessidade de ser realizada a limpeza onde depende de maior tempo trabalho e atenção (WES, 2022). Realmente devemos dar atenção nesse ponto para quando realizarmos os testes em modelos de aprendizagem de máquina não tenhamos necessidade de voltarmos nessa etapa e não prejudicar as execuções, para não haver necessidade de realizar uma nova limpeza de dados.

A limpeza de dados é uma etapa do pré-processamento em que os dados são corrigidos: os valores ausentes são imputados ou apagados, os ruídos são

suavizados, os valores discrepantes são identificados e removidos e as inconsistências são corrigidas. Sem este tratamento, a utilização de algoritmos de aprendizagem de máquina não é possível (SILVA, PERES e BOSCARIOLI, 2016).

Este trabalho tem como objetivo testar o efeito do pré-processamento dos dados na acurácia de algoritmos de aprendizagem de máquina.

Será realizada uma análise exploratória dos dados, para auxiliar na tomada de decisão no tratamento dos dados. Então, os dados ausentes, discrepantes e inconsistentes serão corrigidos com os métodos adequados. Então serão aplicados alguns métodos de aprendizagem de máquina, especificamente algoritmos de classificação supervisionados, e serão mensurados os impactos do tratamento dos dados na acurácia destes algoritmos.

## **2 REFERENCIAL TEÓRICO**

Pois a necessidade de pré-processamento de dados varia dependendo do projeto específico. No entanto, algumas necessidades comuns que podem exigir o pré-processamento de dados incluem: limpeza e formatação de dados; remoção de entradas duplicadas; identificação de outliers; e agrupamento de dados por determinados critérios. Esses pontos abordados podem trazer problemas futuros deixando de serem pré-processados. Afetando a execução de algoritmos, diminuindo a precisão de algoritmos e prejudicando até análise de dados.

Durante o curso de análise e modelagem de dados, um tempo é gasto na preparação de dados: carregamento, limpeza, transformando e organizando. Tais tarefas são frequentemente relatadas como ocupando 80% ou mais do tempo de um analista. Às vezes, a maneira como os dados são armazenados em arquivos ou bancos de dados não está no formato correto para a tarefa específica. muitos pesquisadores optam por fazer o processamento ad hoc de dados de um formulário para outro usando uma linguagem de programação de uso geral, como Python, Perl, R, ou Java, ou ferramentas de processamento de texto Unix como sed ou awk. (WES, 2022, p. 287)

Ao realizar a análises dos dados brutos observamos como os dados podem trazer informações de anomalias como duplicidade, ausência de dados, erros de digitação além de outros. De acordo com Oliveira, Rodrigues e Henriques (2004), o perito nesse domínio de limpeza requer uma atenção essencial para que as correções sejam realizadas de maneira adequada.

A limpeza de dados visa detectar e remover anomalias dos dados com o objetivo de aumentar/melhorar a sua qualidade. Tipicamente o processo de limpeza de dados não pode ser executado sem o envolvimento de um perito do domínio, uma vez que a detecção e correção de anomalias requer conhecimento especializado. (OLIVEIRA, RODRIGUES e HENRIQUES, 2004, p. 5)

A grande demanda de dados hoje tem sido cada vez mais presente em sistemas de informação. Mas realizar o tratamento de dados com qualidade não é apenas substituir dados e criar gráficos. Existe a necessidade de utilização de técnicas de Data Mining para realizar com qualidade a exploração dos dados. Todo esse processo é necessário para que os dados tragam realmente a informação que precisamos. Realizar mineração dos dados não é apenas extrair dados, mas também explorá-los e tratá-los SANTOS (2022).

## **2.1 Recurso utilizados no desenvolvimento**

Atualmente existem linguagens para podermos trabalhar com dados como Julia, R , SQL e Python. Linguagens mundialmente usadas e bastante utilizadas na comunidade de ciência de dados, para o foco desse projeto será utilizada a linguagem Python lançada em 1991, por Guido van Rossum. Python possui diversas bibliotecas para trabalhar com manipulação, visualização e aprendizagem de máquina.

Foi empregada neste projeto a biblioteca Pandas, utilizada para manipular indexação integrada em conjuntos de dados, com opções de leitura e persistência de dados como opções de CSV, SQL, JSON e outras opções. Podendo manipular os dados por rótulos, indexação e conjuntos. Capaz de inserir, excluir, atualizar, dados por colunas e linhas.

Pandas fornece estruturas de dados de alto nível e funções projetadas para tornar trabalhando com dados estruturados ou tabulares intuitivos e flexíveis. Desde seu surgimento em 2010, ajudou a permitir que o Python fosse uma ferramenta poderosa e ambiente produtivo de análise de dados ( WES, 2002, p 20).

Ao adicionar a biblioteca Numpy aumentamos o potencial do projeto. Com matrizes multidimensionais, velozes para vetorização e indexação, possuindo

diversas funções matemáticas para trabalhar com ampla variedade de plataformas e sua sintaxe de fácil compreensão.

*NumPy*, abreviação de *Numerical Python*, tem sido uma pedra angular da computação numérica em *Python*. Ele fornece as estruturas de dados, algoritmos e cola de biblioteca necessária para a maioria das aplicações científicas envolvendo dados em *Python* (WES, 2002, p 19).

*Para realizar observações como os dados estão se comportando dentro das colunas ou no conjunto como um todo. Foi utilizado as bibliotecas Matplotlib e Plotly para visualizar como as colunas estão se dispersando dentro do conjunto, para observação de métricas dos algoritmos de aprendizagem de máquina foi utilizado a biblioteca YellowBrick, a ausência dos dados está presente dentro conjunto sendo observado com biblioteca missingno, essas bibliotecas de visualização possuem diferentes recursos, que podemos usufruir de maneira que possa facilitar os problemas encontrados no conjunto.*

*Matplotlib é a biblioteca Python mais popular para produzir gráficos e outras visualizações de dados bidimensionais. Foi originalmente criado por John D. Hunter e agora é mantido por uma grande equipe de desenvolvedores. Ele é projetado para a criação de parcelas adequadas para publicação. Enquanto existem outras bibliotecas de visualização disponíveis para programadores Python, matplotlib ainda é amplamente utilizado e se integra razoavelmente bem com o resto do ecossistema (WES, 2002, p 21).*

*A etapa de normalização de textos foi utilizado a biblioteca Unidecode, muitas entradas de texto podem chegar no conjunto de maneira despadronizada, como exemplo tendo um valor que corresponda “Evasaõ” em uma linha e na outra tenha “Evasão”, estarei possuindo dois valores diferentes para o mesmo. Com Unidecode posso retirar caracteres especiais e trazer o real valor que necessito para coluna ou conjunto que estou trabalhando, deixando de possuir uma redundância desnecessária.*

*Em nossos experimentos, o sistema de linha de base foi Unidecode, uma Biblioteca Python que fornece um mapeamento independente de idioma de um caractere Unicode para uma string ASCII fixa. Embora esta*

*seja uma linha de base ingênua, para idiomas que não têm muitos dados para treinar sistemas de transliteração, esta pode ser uma das poucas opções de transliteração disponíveis (Winston e Yarowsky, 2018, p 21).*

*Para realizar o impacto do pré-processamento em algoritmos de aprendizagem de máquina, foram escolhidos 3 algoritmos de aprendizagem de máquina do tipo de classificação. Sendo Random Forest, Logistic Regression e Gradient Booster.*

*Random Forest (RF) é um método de ensemble, que é baseado em uma árvore de decisão. A RF reduz o grau de overfitting combinando vários avaliadores de overfit (ou seja, árvores de decisão) para formar um algoritmo de aprendizado definido. Cada árvore de decisão pode obter o resultado da decisão de classificação correspondente (WANG, 2021, p 5).*

### **3 Metodologia**

Quando iniciamos o pré-processamento, existem diferentes etapas como organização, limpeza e estruturação dos dados. Nessas etapas possuem maior tempo do desenvolvedor, exige bastante atenção para quando finalizar essa etapa não traga uma para desnecessária do treinamento dos algoritmos ou atrapalhe a execução de alguma ferramenta de visualização. Para isso devemos utilizar técnicas adequadas e ferramentas para trazer a melhor qualidade dos dados. Nesse projeto está sendo tratado sobre a etapa limpeza dos dados dentro do pré-processamento, como no mundo real que limpamos atritos que estão incomodando, dentro conjunto pode existir inconsistências desnecessárias e ausência de dados.

#### **3.1 Base de dados**

Como base de dados foi utilizado o conjunto de dados da Universidade Federal de Juiz de Fora disponibilizados através de seu Portal da Transparência (ANDRADE et al, 2019). Esse conjunto foi concatenado com três conjuntos de diferentes tamanhos, conjunto graduação presencial, graduação doutorado e graduação a distância. De maneira artificial foi aumentando a incidência de inconsistências e ausência para medir o impacto do da limpeza realizada.

Abaixo pode ser visualizado como o conjunto está dividido, a primeira coluna está o nome das colunas do conjunto, a segunda encontra-se a sua descrição e por último o tipo da coluna.

Tabela 1 – Nome e tipos do conjunto de dados da Universidade Federal de Juiz de Fora

<b>Nome</b>	<b>Descrição</b>	<b>Tipo</b>
ANO_INGRESSO	Ano em que o aluno ingressou	INT
SEMESTRE_INGRESSO	Semestre em que o aluno início	INT
TIPO_INGRESSO	Entrou por vestibular	OBJECT
COTA	Grupos de classificação de cota	OBJECT
CURSO_NOME	Nome do curso que foi escolhido na Universidade	OBJECT
AREA	Área em que o curso está classificado	OBJECT
SITUACAO	Situação do aluno se ocorreu Conclusão.	OBJECT
MOTIVO_SAIDA	Motivo que escolheu para ser evadido	OBJECT
CAMPUS	Localização da cidade onde se encontra o campus	OBJECT
TURNIO	Horário que encontra-se o curso	OBJECT
ETNIA	Grupo culturalmente	OBJECT
SEXO	Gênero sexual	OBJECT
TIPOCURSO	Tipo do curso	OBJECT
LNG	Longitude	FLOAT
LAT	Latitude	FLOAT
LOCAL	Localização do aluno	OBJECT
LNG_ORGM	Longitude orgm	FLOAT
LAT_ORGM	Latitude orgm	FLOAT
LOCAL_ORGM	Localização do aluno orgm	OBJECT

Fonte: Produzido pelo autor do artigo.



### **3.2 Limpeza dos Dados**

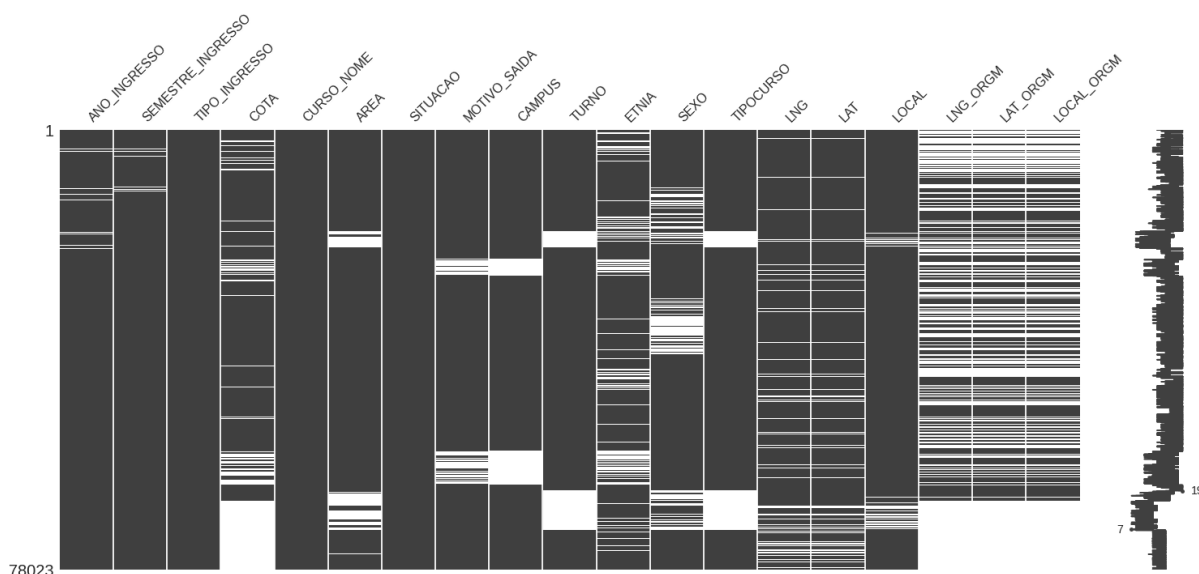
Quando realizamos a limpeza de dados deve ser desenvolvida uma estratégia utilizando meios para visualização para trazer melhor compreensão sobre a necessidade de limpeza do conjunto de dados. Nem sempre o conjunto trará informações adequadas para determinar um objetivo. Essa etapa trará visualizações sobre erros, ausência de dados, duplicidade, outliers e inconsistência. Utilizando a ferramenta Pandas possuímos um arsenal de manipulação de dados, para podermos resolver essas inconsistências. Essa etapa deve ser tratada como um dossiê dos dados, documentar cada ponto encontrado no conjunto será bastante útil quando efetuamos a limpeza dos dados. Abaixo podemos ver as inconsistências encontradas no conjunto da Universidade Federal de Juiz de Fora.

### **3.3 Ausência de Dados**

Os dados ausentes têm enorme impacto em nosso conjunto de dados. Podemos corrigir a ausência de dados de diferentes maneiras, mas devemos também lembrar que os dados ausentes pode ser também uma informação que conjunto está trazendo não realizado na coleta de dados. Ao visualizar onde se encontra a ausência dos dados, entenderemos onde pode ter ocorrido erros de entrada causando essa ausência.

Utilizando a biblioteca missingno podemos observar como os dados ausentes estão dissipando dentro do conjunto na figura 1.

Figura 1: Visualização de dados ausentes



Fonte: Produzido pelo autor do artigo.

Nós podemos excluir esses valores ausentes, mas vamos perder muitos dados. Uma maneira simples é observar o tipo da coluna que possui o valor ausente. Como exemplo a coluna “ETNIA”. De maneira simples substituir o valor ausente pela moda, mas realizando essa substituição estaremos atribuindo o valor ausente sendo 17.05% dos registros, junto com valor “BRANCA”, que possui 42.58 % uma vez que este possui a maior frequência, ficando total de 59.65 %.

Podemos substituir esse valor ausente, por um novo valor que identifique essa ausência como "valor ausente" ou “-”, deve sempre se atentar ao tipo da coluna, para que não criemos uma inconsistência desnecessária, assim temos um novo dado mostrando que antes de possuímos os dados, possui uma falha ao coletar os dados, gerando essa ausência. No futuro pode-se estudar o que houve essa ausência, realizando na próxima coleta dos dados maior eficiência.

### 3.4 Inconsistência de tipo de coluna

Com erros de digitação, erros de padronização, erros de conversões podem afetar o tipo de dado em que se encontra uma coluna. Observando a tabela abaixo do livro do autor LUBANOVIC (2020). Pode-se observar os tipos que a linguagem Python possui.

Tabela 2 – Tipos de dados da linguagem Python

Nome	Type	Mutável ?	Exemplos
Boolean	bool	no	True,False
Integer	int	no	47, 25000, 25_000
Floating	float	no	3.14, 2.7e5
Complex	complex	no	3j, 5 + 9j
Text string	str	no	'alas', "alack", "" a verse attack""
List	list	yes	['Winken', 'Blinken', 'Nod']
Tuple	tuple	no	(2,4,8)
Bytes	bytes	no	b'ab\xff'
ByteArray	bytearray	yes	bytearray(...)
Set	set	yes	set([3, 5, 7])
Frozen set	frozenset	no	frozenset(['Elsa', 'Otto'])
Dictionary	dict	yes	{'game': 'bingo', 'dog': 'dingo', 'drummer': 'Ringo'}

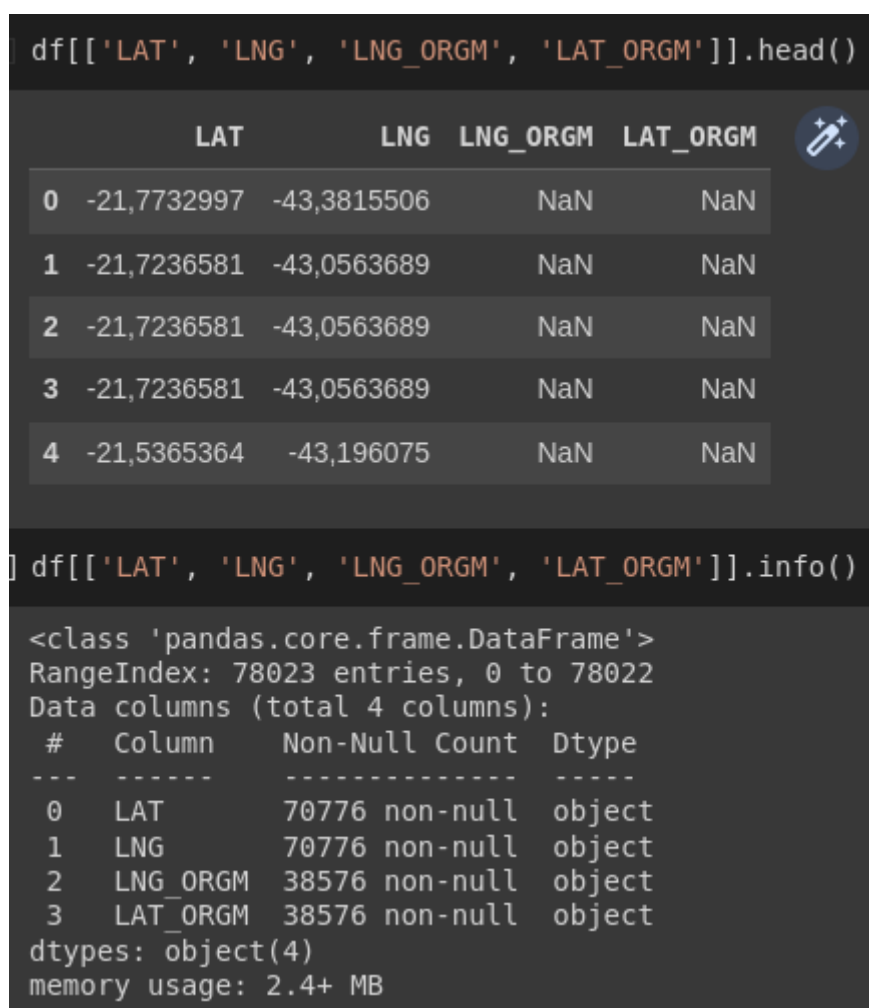
Fonte: LUBANOVIC (2020).

Utilizando a ferramenta Pandas observar os tipos das colunas, em muitos casos por conta de erros digitação ou padronização podem afetar o tipo da coluna.

Dessa maneira estamos perdendo uma coluna que pode ser utilizada para o processamento e afetando algumas visualizações.

Em nosso conjunto de dados foi observado que as colunas latitude, longitude, longitude orgm e latitude orgm estão sendo afetadas por conta desse erro de digitação, de maneira não padronizada. Utilizando um dos valores da coluna latitude observamos que está recebendo o valor “-21,7531292” por conta da vírgula a ferramenta Pandas interpreta como um valor tipo objeto, para a ferramenta conseguir interpretar esse valor como numérico deve ser substituído a vírgula por ponto. Na imagem abaixo podemos observar na figura XX como isso afeta transformando em valor ‘object’ sendo que deveria ser do tipo ‘floating’.

Figura 2: Visualização de dados inconsistentes



```
df[['LAT', 'LNG', 'LNG_ORGM', 'LAT_ORGM']].head()
```

	LAT	LNG	LNG_ORGM	LAT_ORGM
0	-21,7732997	-43,3815506	NaN	NaN
1	-21,7236581	-43,0563689	NaN	NaN
2	-21,7236581	-43,0563689	NaN	NaN
3	-21,7236581	-43,0563689	NaN	NaN
4	-21,5365364	-43,196075	NaN	NaN

```
df[['LAT', 'LNG', 'LNG_ORGM', 'LAT_ORGM']].info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 78023 entries, 0 to 78022
Data columns (total 4 columns):
#   Column      Non-Null Count  Dtype
---  -
0   LAT         70776 non-null  object
1   LNG         70776 non-null  object
2   LNG_ORGM    38576 non-null  object
3   LAT_ORGM    38576 non-null  object
dtypes: object(4)
memory usage: 2.4+ MB
```

Fonte: Produzido pelo autor do artigo.

### 3.5 Normalização de Textos

Formatar os textos traz uma atenção rigorosa para nossos valores dentro das colunas, como exemplo abaixo podemos observar que erros de digitação afetam a identificação dos valores, a linguagem python reconhece diferentes tipos da palavra “Evadido”, como exemplo na Figura 3, possuímos redundância desnecessária para o mesmo valor, quando usamos Pandas para contar esses valores dentro da coluna observamos como existem duplicatas desnecessárias criando perda de dados.

Figura 3: Diferença de texto

```
1 df['SITUACAO'].value_counts()

Ativo      24081
Concluido  23112
Evadido     15321
evadido     4781
ConcluIDo   4584
AtiVO       3710
EVaDIDO     2434
Name: SITUACAO, dtype: int64

1 if 'Evadido' != 'evadido':
2 |     print('True')

True
```

Fonte: Produzido pelo autor do artigo.

## 4 Resultados

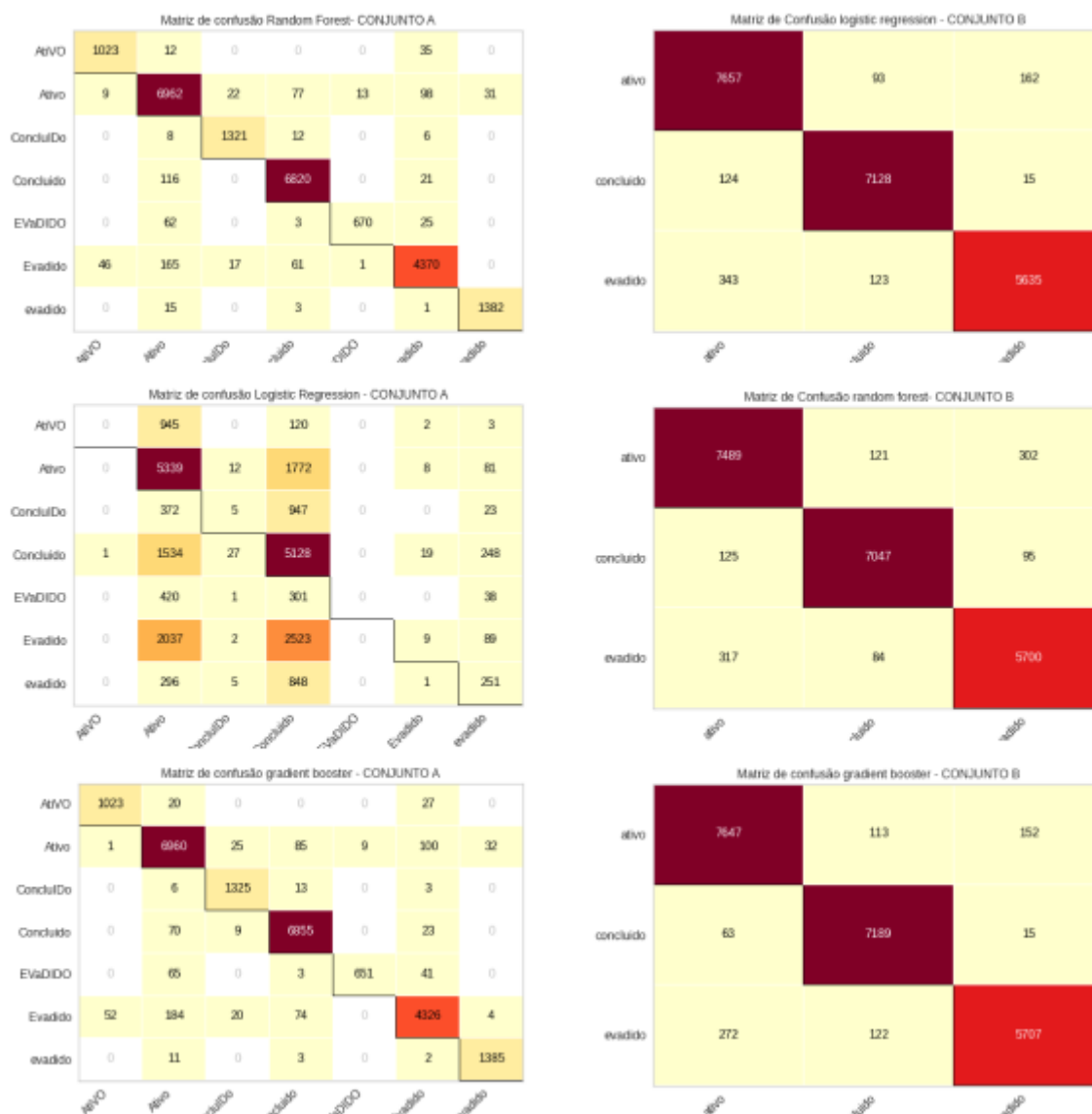
Por fim, para observar a qualidade dos dados após o pré-processamento. Foi separado em dois conjuntos, o conjunto A sendo os dados sem o pré-processamento e o conjunto B com os dados pré-processados. Nessa etapa de aprendizagem de máquina, foram utilizados os seguintes algoritmos, Random Forest, Gradient Booster e regressão logística.

Dando início ao treinamento do algoritmo no primeiro momento de execução com conjunto A foi observado uma exceção ("could not convert string to float: 'VESTIBULAR'"), essa exceção ocorre por conta do conjunto não ser convertido em dados numéricos. No conjunto A existem algumas colunas numéricas, quando ocorre a execução essas colunas numéricas não vão estar sendo executadas separadamente no conjunto, existe a necessidade de converter as colunas categóricas para numéricas também para não apresentar a exceção novamente. Diferente do conjunto B pode seguir a execução sem exceções.

Após ter solucionado a exceção anterior dando continuidade com treinamento, novamente o conjunto A apresentou novamente uma exceção, ("Input contains NaN, infinity or a value too large for dtype 'float32' "), essa exceção ocorreu pela grande ausência de dados dentro desse conjunto, para solucionar foi substituído os dados ausentes para poder prosseguir. O conjunto B não ocorreu exceção, ao realizar o tratamento dos valores ausentes na etapa anterior, apresentou eficiência, não prejudicando a execução do algoritmo.

Novamente após realizar correção da exceção apresentada pelo conjunto A, os dois grupos finalizaram o treinamento. Foram obtidos os seguintes resultados na figura 5.

Figura 4: Visualização de matriz de confusão



Fonte: Produzido pelo autor do artigo.

Pode ser observado na figura 4, que o conjunto A por não ter passado no pré-processamento a redundância dos textos para mesmo valor ficou disperso. No conjunto B os dados não trazem essa redundância agrupando os valores da coluna situação de maneira mais uniforme.

Realizando uma análise de acurácia dos algoritmos foram obtidos os seguintes resultados. Observando acurácia score existem resultados bastante similares, no conjunto A não realizando a limpeza observamos que traz erros de execução, tendo que solucionar os problemas dessa maneira chegando próximo ao mesmo nível de limpeza no conjunto B os algoritmos obtiveram escore próximos. Comparando os resultados da coluna *accuracy score* como mostrado na tabela 3, o

resultado não ficou claro se houve melhoria, para uma segunda tentativa de observação de melhora do conjunto A para o conjunto B, foi utilizado da biblioteca *sklearn* com sua função de validação cruzada passando como parâmetro 10 de *K-fold*, para o conjunto A e o conjunto B sendo divididos em conjuntos menores e treinados novamente, após isso foi selecionado a média de escore dos dois conjuntos. Abaixo na tabela 3 podemos observar o conjunto B que obteve melhor resultado para cada algoritmo utilizado neste projeto.

Tabela 3: Métricas do conjunto A e conjunto B

model	accuracy_score	cross_val_score_mean
Random Forest Conjunto A	0.9633015764514888	0.7342520014206538
Random Forest Conjunto B	0.9618421052631579	0.8697181006254215
Logistic Regression Conjunto A	0.45849532191224845	0.45327648582172975
Logistic Regression Conjunto B	0.9595864661654135	0.9017366578174354
Gradient Booster Conjunto A	0.9623189644123553	0.7785854272982052
Gradient Booster Conjunto B	0.9653665413533835	0.904371000988081

Fonte: Produzido pelo autor do artigo.

Também observamos a qualidade que foi obtida no pré-processamento para realização de análise gráfica.

Ao realizar uma análise de gráfica de um mapa para visualizar os dados de longitude e latitude, não realizar o pré-processamento trouxe falhas para a execução. Na figura 5, podemos observar o erro ao executar a função para visualizar os dados em um gráfico de mapa, isso ocorre pela falta de padronização da entrada dos dados afetando o tipo da coluna que deveria ser numérica. Ao gerar um gráfico não pode ser executado.



Figura 5: Exceção da visualização de Latitude e Longitude

```
-----  
TypeError                                Traceback (most recent call last)  
<ipython-input-321-7c7446cee2e2> in <module>  
----> 1 geometry = [Point(xy) for xy in zip(df['LNG'], df['LAT'])]  
      2 gdf = GeoDataFrame(df_pos, geometry=geometry)  
  
-----  
      2 frames -----  
/usr/local/lib/python3.7/dist-packages/shapely/geometry/point.py in geos_point_from_py(ob, update_geom, update_ndim)  
    260     coords = ob  
    261     n = len(coords)  
--> 262     dx = c_double(coords[0])  
    263     dy = c_double(coords[1])  
    264     dz = None  
  
TypeError: must be real number, not str
```

Fonte: Produzido pelo autor do artigo.

Na figura 6, observamos que ao realizar o pré-processamento trouxe melhoria para execução do gráfico de mapa, após as colunas de longitude e latitude receberem o pré-processamento podemos obter o resultado no gráfico, sem ocorrer exceções como na figura anterior trazendo melhoria.

Figura 6: Visualização de Latitude e Longitude



Fonte: Produzido pelo autor do artigo.

## 5 CONCLUSÃO

Após ter realizado o pré-processamento, o conjunto obteve melhoria nos resultados de algoritmos de aprendizagem de máquina. A limpeza dos dados evitou a ocorrência de exceções no treinamento, não gerando uma parada desnecessária no prosseguimento do projeto. No conjunto A que não teve o pré-processamento trouxe exceções, como exemplo a exceção pela ausência dos dados o algoritmo não pode prosseguir, dessa maneira mesmo não tendo feito uma limpeza com mais qualidade, para dar continuidade no treinamento foi realizado uma limpeza mais simples no conjunto, retirando a ausência dos dados. Mas não evitou a redundância de textos criados pela entrada de dados. Para o desenvolvimento de análises, algumas técnicas foram afetadas pela coluna não possuir o valor real. Pode-se concluir que para obter a melhor qualidade de dados deve se atentar a cada situação possível de inconsistência. Buscar as melhores técnicas para solucionarmos diferentes inconsistências que podem estar contidas nos conjuntos de dados. Utilizar ferramentas de visualização como matplotlib e plotly traz auxílio na etapa de limpeza de dados.

## REFERÊNCIAS

- ANDRADE, T. S. D.; CAMPOS, R. S.; AMORIM, C. C.; CONDÉ, E. A. S. (2019). **Acesso à informação digital: uma proposta para disponibilização de informações públicas no portal da transparência da UFJF**
- BILL, Lubanovic. **Introducing Python Modern Computing in Simple Packages** Published by O'Reilly Media, Inc., 1005 Gravenstein Highway North, Sebastopol, CA 95472. (20 setembro 2022)
- CARDOSO, Leonor; DUARTE Adelino Gomes; REBELO Teresa. **Gestão do conhecimento: dos dados à informação e ao conhecimento** Comportamento organizacional e Gestão 9 (2003): 55-84.
- OLIVEIRA, P.; RODRIGUES, Fátima; HENRIQUES P. **Limpeza de dados-uma visão geral** Data Gadgets (2004): 39-51.
- SANTOS, Douglas Brandão **Visualização de dados estruturados e não estruturados da área da saúde**. Universidade Estadual Paulista (Unesp), 2022. Disponível em: <<http://hdl.handle.net/11449/216221>>.
- SILVA, Leandro Augusto, **Introdução à mineração de dados: com aplicações em R** / Leandro Augusto da Silva, Sarajane Marques Peres, Clodis Boscarioli. – 1. ed. – Rio de Janeiro: Elsevier, 2016. il. ; 27 cm.
- WANG, Xuchun, et al. **Exploratory study on classification of diabetes mellitus through a combined Random Forest Classifier** BMC medical informatics and decision making 21.1 (2021): 1-14.
- WES, McKinney . **Python for Data Analysis: Data Wrangling with Pandas, Numpy, and Jupyter**. O'Reilly Media; 3rd edição, 20 setembro 2022
- WU, Winston; YAROWSKY David. **A comparative study of extremely low-resource transliteration of the world's languages** Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018). 2018.