Análise Preditiva da Saúde Cardíaca de Cidadãos Brasileiros

Bernardo Vieira
Instituto de Ciências Exatas e Informática
Sabará, Minas Gerais, Brasil
baavieira@sga.pucminas.br

Gabriel Jota Lizardo
Instituto de Ciências Exatas e Informática
Nova Lima, Minas Gerais, Brasil
gabriel.jota@sga.pucminas.br

Guilherme Oliveira de Rodrigues
Instituto de Ciências Exatas e Informática
Nova Lima, Minas Gerais, Brasil
guilherme.rodrigues.1449815@sga.pucminas.br

Henrique Oliveira da C. Franco
Instituto de Ciências Exatas e Informática
Belo Horizonte, Minas Gerais, Brasil
henrique.franco@sga.pucminas.br

Larissa Mariella
Instituto de Ciências Exatas e Informática
Matozinhos, Minas Gerais, Brasil
larissa.mariella@sga.pucminas.br

Victor Hugo
Instituto de Ciências Exatas e Informática
Belo Horizonte, Minas Gerais, Brasil
vhbraz@sga.pucminas.br

ABSTRACT

Esse projeto visa alcançar um entendimento melhor acerca de atividades, hábitos ou características que podem influenciar na probabilidade de uma pessoa ter (ou não) doenças cardíacas. Inicialmente foi aplicado um pré-processamento na base de dados, como seleção de colunas relevantes, balanceamento de dados e tratamento de dados ausentes. A partir disso foi aplicado uma árvore de decisão como modelo de previsão. Os resultados geraram um relatório que busca entender melhor quais circunstâncias estão mais relacionadas com problemas cardíacos.

KEYWORDS

complicações cardíacas, saúde, análise de dados, inteligência artificial, árvore de decisão, pesquisa nacional de saúde, medicina preventiva, predição

ACM Reference Format:
Bernardo Vieira, Gabriel Jota Lizardo, Guilherme Oliveira de Rodrigues, Henrique Oliveira da C. Franco, Larissa Mariella, and Victor Hugo. 2024. Análise Preditiva da Saúde Cardíaca de Cidadãos Brasileiros. In *Proceedings of ACM Conference (Conference'17).* ACM, New York, NY, USA, 14 pages. https://doi.org/10.1145/1122445.1122456

Conference'17, July 2017, Washington, DC, USA

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/1122445.1122456

1 INTRODUÇÃO

A saúde do coração é um dos pilares fundamentais para o bem-estar e a longevidade humana. Doenças cardiovasculares são uma das principais causas de morte no mundo, representando um grave problema de saúde pública. Assim, prever condições cardíacas antes que se tornem críticas é uma abordagem preventiva de grande valor.

Neste contexto, a utilização de inteligência artificial (IA) surge como uma ferramenta promissora na área médica. Com o objetivo de contribuir para a detecção precoce de problemas cardíacos, este projeto explora a criação de um modelo preditivo utilizando dados da Pesquisa Nacional de Saúde (PNS). A PNS é uma base de dados abrangente que coleta informações sobre diversos aspectos da saúde dos brasileiros, incluindo fatores de risco para doenças cardíacas, como hábitos de vida, condições de saúde preexistentes e características demográficas [1].

O projeto busca desenvolver uma IA capaz de analisar esses dados e identificar padrões associados à presença de problemas cardíacos. Utilizando técnicas de aprendizado de máquina, a IA será treinada para reconhecer sinais sutis que podem escapar à observação humana, oferecendo uma ferramenta potencialmente poderosa para apoiar diagnósticos clínicos.

A importância deste trabalho reside não apenas no avanço tecnológico, mas também na possibilidade de oferecer uma solução preventiva de baixo custo e alta precisão. A previsão de condições cardíacas com antecedência pode reduzir significativamente a mortalidade, melhorar a qualidade de vida dos pacientes e aliviar a pressão sobre o sistema de saúde. Ao final, este estudo propõe uma aplicação prática de IA que pode revolucionar o campo da medicina preventiva, colocando em evidência o papel da ciência de dados na saúde pública.

2 MATERIAIS E MÉTODOS

2.1 Descrição da base de dados

A figura 1 ilustra os grupos de atributos que são relevantes e consequentemente foram utilizados para a classificação das entidades com base na classe escolhida.

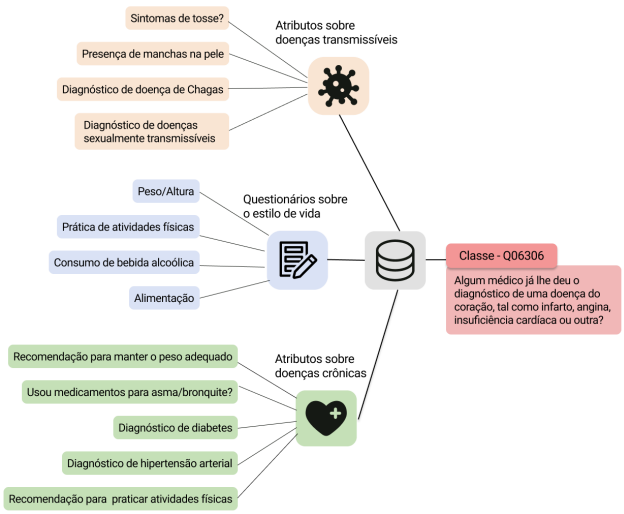

Figure 1: Grupo de atributos da Base de dados

2.2 Pré-processamento

2.2.1 Filtro de instâncias. Primeiramente foi aplicado um filtro na base de dados, mantendo somente as instâncias onde a variável alvo, chamada na base de "Q06306", é diferente de nula. Já diminuindo o tamanho da base em cerca de 61%, partindo de 293727 para 90847 instâncias. Logo, assim ficou o número de instâncias por classe:

Q06306	Count
2.0	86114
1.0	2356

Table 1: 2.0 = Não e 1.0 = Sim

Após isso, foram retiradas as colunas que possuíam mais de 70% de valores nulos, sendo estas inúteis para o trabalho, como segue o código:

```
initial_rows, initial_columns = df.shape
print(f"Numero Inicial de Instancias: {initial_rows}")
print(f"Numero Inicial de Colunas {initial_columns}")

missing_percentage = df.isnull().mean() * 100

columns_to_drop =
missing_percentage[missing_percentage > 70].index
df_cleaned = df.drop(columns=columns_to_drop)

after_column_removal_rows,
after_column_removal_columns
= df_cleaned.shape
print(f"Numero de Colunas Restantes
apos remover colunas com >70% de Null:
{after_column_removal_columns}")

threshold = len(df_cleaned.columns) * 0.5
df_cleaned = df_cleaned.dropna(thresh=threshold)

final_rows, final_columns = df_cleaned.shape
print(f"Numero de Instancias apos
retirar instancias com mais
de 50% de campos nulos: {final_rows}")

df = df_cleaned
```

2.2.2 Codificação numérica-simbólica. Como a maior parte das colunas já estava em uma codificação com variáveis de tipo correto e padronizado (por exemplo, 0=não, 1=sim), foi necessário o uso do "Label Encoder" somente para a variável alvo, transformando os valores de 2.0 = não e 1.0 = sim, para 0 = não e 1 = sim. Como demonstrado no código abaixo:

```
from sklearn.preprocessing import LabelEncoder

# Instanciando o LabelEncoder
le = LabelEncoder()

# Aplicar a transformação na coluna 'Q06306'
df['Q06306'] = le.fit_transform(df['Q06306'])

print(df['Q06306'].value_counts())
```

2.2.3 Balanceamento de dados. Após todo esse processo, foi percebido que havia um grande desbalanceamento entre a classe, sendo 86114 "nãos" e 2356 "sims", sendo necessário um balanceamento. Para a etapa de "Undersampling", onde são retirados dados da classe majoritária para balancear com a classe minoritária, aplicou-se o modelo "RandomUnderSampler", mantendo o conjunto majoritário 20% maior que o conjunto majoritário. Sendo necessário lembrar também que as instâncias retiradas foram introduzidas no conjunto de teste. Código abaixo:

```
from sklearn.model_selection import train_test_split

# Dividir os dados em conjunto de treino e teste

X_train, X_test, y_train, y_test =
train_test_split(df.drop(columns='Q06306'),
df['Q06306'], test_size=0.3, random_state=42)
```

```
from imblearn.under_sampling import RandomUnderSampler

import numpy as np

rus = RandomUnderSampler(sampling_strategy=0.8,
random_state=42)
X_resampled, y_resampled =
rus.fit_resample(X_train, y_train)
print(y_resampled.value_counts())

# Obter os índices que foram mantidos e removidos
mask_kept = rus.sample_indices_

# Índices das instâncias mantidas no conjunto de treino
mask_removed = np.setdiff1d(np.arange(len(X_train)),
mask_kept)

# Índices das instâncias removidas

# Capturar as instâncias removidas
X_removed = X_train.iloc[mask_removed]
y_removed = y_train.iloc[mask_removed]

# Verificar quantas instâncias foram removidas
print(f"Número de instâncias removidas:
{len(mask_removed)}")

X_test = pd.concat([X_test, X_removed],
ignore_index=True)
y_test = pd.concat([y_test, y_removed],
ignore_index=True)
```

Finalmente, após esses ajustes, o número de instâncias ficou como a seguir:

Q06306	Count
1:Não	2356
0:Sim	2356

Table 2: Número de instâncias por classe após Undersampling.

2.2.4 Tratamento de dados ausentes. Por conta do tamanho da base e o grande número de instâncias foi utilizado um "Imputer", mais especificamente o "KNNImputer"[2] ou seja, um modelo para imputar valores nos dados ausentes. Foi percebido uma demora muito grande para imputar todos os valores, então foi utilizada uma função que separa os conjuntos em blocos, imputa cada bloco, e os concatena. A função está mais detalhada na seção 3.5. O chamar da função está detalhado no código abaixo:

```
# Aplicando o KNNImputer com barra de progresso
X_resampled = impute_with_progress(X_resampled, imputer)

# Aplicando o KNNImputer com barra de progresso
X_test = impute_with_progress(X_test, imputer)
```

2.2.5 Ambiente de Desenvolvimento. Todo o código foi desenvolvido na versão 3.9 do Python, utilizando o ambiente `Jupyter Notebook` para facilitar a execução e visualização dos resultados. As principais bibliotecas utilizadas foram:

- `pandas` para manipulação de dados.
- `scikit-learn` para a construção e avaliação do modelo.
- `imblearn` para o balanceamento das classes.
- `matplotlib` e `seaborn` para a visualização de gráficos e resultados.
- `tqdm` para exibir o progresso da imputação de valores nulos.

2.2.6 Etapas de pré-processamento. As etapas de pré-processamento foram essenciais para a preparação da base de dados para o algoritmo de aprendizado de máquina. A Figura 2 apresenta o fluxograma com as principais etapas executadas:

- **Remoção de colunas desnecessárias ou redundantes**: Havia grande quantidade de colunas desnecessárias ou redundantes. Os atributos remanescentes estão na tabela da seção 2.5
  - `P00102` - O(A) Sr(a) sabe seu peso? – Redundante com as colunas P00103 e P00104, pois o peso final (P00104) já contêm essa informação.
  - `P00103` - Peso - Informado (em kg) – Redundante com as colunas P00102 e P00104, pois o peso final (P00104) já é utilizado.
  - `P00402` - O(A) Sr(a) sabe sua altura? (mesmo que seja valor aproximado) – Redundante com as colunas P00403 e P00404, pois a altura final (P00404) já contêm essa informação.
  - `P00403` - Altura - Informada (em cm) –Redundante com as colunas P00402 e P00404, pois a altura final (P00404) já é utilizada.
  - `UPA_PNS` - UPA. Desnecessário para o propósito do trabalho, já que a UPA não influência no diagnostico.
  - `VDDATA` - Data de geração do arquivo de microdados. Desnecessário para o propósito do trabalho, já que a a data de geração do arquivo não influência no diagnostico.
  - `V0006_PNS` - Número de ordem do domicílio na PNS foram removidas, já que são desnecessários para o propósito do trabalho.
- **Remoção de colunas com muitos valores nulos**: O número inicial de colunas foi diminuído de 181 para

103, removendo 78 colunas que possuíam mais de 70% de valores nulos.

- **Imputação de valores nulos**: Usamos o algoritmo `KNNImputer` com `n_neighbors=6` para preencher os valores ausentes. O valor '6' foi escolhido após testes e demonstrou a melhor performance.
- **Balanceamento de classes**: Inicialmente aplicamos `RandomUnderSampler` para reduzir o número de instâncias da classe majoritária e, em seguida, utilizamos o `SMOTE` para balancear a base de dados.

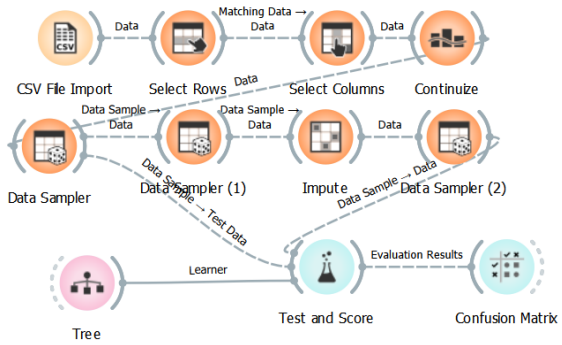

Figure 2: Fluxo de Pré-processamento dos Dados

2.3 Métricas de avaliação de qualidade

Para avaliar os resultados foram utilizadas como métricas de avaliação a precisão, o "recall", ou sensibilidade, e o "f1-score", que estão disponiveis na seção 3 - Resultados e Discussões.

Para a realização da avaliação de qualidade, foi utilizado o seguinte código:

```
# Importar bibliotecas necessárias
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier, plot_tree
from sklearn.metrics import accuracy_score, classification_report, confusion_matrix
import matplotlib.pyplot as plt

# Criar o classificador de árvore de decisão
clf = DecisionTreeClassifier(random_state=42, class_weight='balanced', max_depth=3)
# Treinar o modelo com os dados de treino
clf.fit(X_resampled, y_resampled)
# Fazer previsões nos dados de teste
y_pred = clf.predict(X_test)

# Avaliar o desempenho do modelo
print("Acurácia:", accuracy_score(y_test, y_pred))
print("\nRelatório de classificação:")
print(classification_report(y_test, y_pred))
# Matriz de confusão
print("\nMatriz de confusão:")
cm = confusion_matrix(y_test,y_pred)
# Plotar a árvore de decisão
plt.figure(figsize=(20, 10))  # Tamanho da figura
X_resampled = pd.DataFrame(X_resampled, columns=X_train.colu

# Certificando-se de que os nomes das características e classe
feature_names = X_resampled.columns.astype(str).tolist()
class_names = clf.classes_.astype(str).tolist()
plot_tree(clf, filled=True, feature_names=feature_names, cla
plt.show()
```

2.4 Codificação

Os atributos possuem diversos valores de resposta. Além de poder serem nulos, alguns podem ser sim, não ou ignorado – Alguns são ordinais, indicando, por exemplo, quantidade de horas, ou um determinado número de objetos, e alguns são nominais continuos – Como a frequência em que alguém consome algo, ou discretos, como qual exercício a pessoa pratica.

2.5 Atributos

A tabela seguinte possui uma lista de todos os atributos utilizados e seus respectivos valores. Vale lembrar que, para a arvore de decisão, a entropia de todos os atributos é calculada e apenas os atributos mais relevantes são mantidos.

Código	Descrição	Valores	Observações
P00104	Peso - Final (em kg) (3 inteiros e 1 casa decimal)	1-599: Kg. 999: Ignorado	
P00404	Altura - Final (em cm) (3 inteiros)	1-299: Cm, 999: Ignorado	
P005	A Sra está grávida no momento?	1: Sim, 2: Não, 3: Não sei	
P00601	Ontem o(a) Sr(a) comeu arroz, macarrão, polenta, cuscuz ou milho verde.	1: Sim, 2: Não, 9: Ignorado	
P00602	Batata comum, mandioca/aipim/macaxeira, cará ou inhame.	1: Sim, 2: Não, 9: Ignorado	
P00603	Feijão, ervilha, lentilha ou grão de bico.	1: Sim, 2: Não, 9: Ignorado	
P00604	Carne de boi, porco, frango, peixe	1: Sim, 2: Não, 9: Ignorado	
P00605	Ovo (frito, cozido ou mexido).	1: Sim, 2: Não, 9: Ignorado	
P00607	Alface, couve, brócolis, agrião ou espinfre.	1: Sim, 2: Não, 9: Ignorado	
P00608	Abóbora, cenoura, batata doce ou quiabo/caruru.	1: Sim, 2: Não, 9: Ignorado	
P00609	Tomate, pepino, abobrinha, berinjela, chuchu ou beterraba.	1: Sim, 2: Não, 9: Ignorado	
P00610	Mamão, manga, melão amarelo ou pequi.	1: Sim, 2: Não, 9: Ignorado	
P00611	Laranja, banana, maçã, abacaxi.	1: Sim, 2: Não, 9: Ignorado	
P00612	Leite	1: Sim, 2: Não, 9: Ignorado	
P00613	Amendoim, castanha de caju ou castanha do Brasil/Pará	1: Sim, 2: Não, 9: Ignorado	
P00614	ONTEM o(a) Sr(a) tomou ou comeu:Refrigerante	1: Sim, 2: Não, 9: Ignorado	
P00615	Suco de fruta em caixinha ou lata ou refresco em pó.	1: Sim, 2: Não, 9: Ignorado	
P00616	Bebida achocolatada ou iogurte com sabor.	1: Sim, 2: Não, 9: Ignorado	
P00617	Salgadinho de pacote ou biscoito/bolacha salgado.	1: Sim, 2: Não, 9: Ignorado	
P00618	Biscoito/bolacha doce ou recheado ou bolo de pacote.	1: Sim, 2: Não, 9: Ignorado	
P00619	Sorvete, chocolate, gelatina, flan ou outra sobremesa industrializada.	1: Sim, 2: Não, 9: Ignorado	
P00620	Salsicha, linguiça, mortadela ou presunto.	1: Sim, 2: Não, 9: Ignorado	
P00621	Pão de forma, de cachorro-quente ou de hambúrguer.	1: Sim, 2: Não, 9: Ignorado	
P00622	Margarina, maionese, ketchup ou outros molhos industrializados.	1: Sim, 2: Não, 9: Ignorado	
P00623	Macarrão instantâneo, sopa de pacote, lasanha congelada ou outro prato congelado comprado pronto industrilizado.	1: Sim, 2: Não, 9: Ignorado	
P006	Em quantos dias da semana o(a) Sr(a) costuma comer feijão?	1-7: Dias, 0: Nunca ou Menos de Uma Vez. 9: Ignorado	
P00901	Em quantos dias da semana, o(a) Sr(a) costuma comer pelo menos um tipo de verdura ou legume (sem contar batata, mandioca, cará ou inhame) como alface, tomate, couve, cenoura, chuchu, berinjela, abobrinha?	1-7: Dias, 0: Nunca ou Menos de Uma Vez. 9: Ignorado	
P01001	Em geral, o(a) Sr(a) costuma comer esse tipo de verdura ou legume:	1: - Uma vez por dia. 2: - Duas vezes por dia. 3: - Três vezes por dia. 9: Ignorado	
P01101	Em quantos dias da semana o(a) Sr(a) costuma comer carne vermelha (boi, porco, cabrito, bode, ovelha etc.)?	1-7: Dias, 0: Nunca ou Menos de Uma Vez. 9: Ignorado	
P013	Em quantos dias da semana o(a) Sr(a) costuma comer frango/galinha?	1-7: Dias, 0: Nunca ou Menos de Uma Vez. 9: Ignorado	
P015	Em quantos dias da semana o(a) Sr(a) costuma comer peixe?	1-7: Dias, 0: Nunca ou Menos de Uma Vez. 9: Ignorado	
P02001	Em quantos dias da semana o(a) Sr(a) costuma tomar suco de caixinha/lata ou refresco em pó ?	1-7: Dias, 0: Nunca ou Menos de Uma Vez. 9: Ignorado	

Código	Descrição	Valores	Observações
P02101	Que tipo de suco de caixinha/lata ou refresco em pó o(a) Sr(a) costuma tomar? (Ler as opções de resposta)	1: Diet/Light/Zero, 2: Normal, 3: Ambos, 9: Ignorado	
P01601	Em quantos dias da semana o(a) Sr(a) costuma tomar suco de fruta natural (incluída a polpa de fruta congelada)?	1-7: Dias, 0: Nunca ou Menos de Uma Vez. 9: Ignorado	
P018	Em quantos dias da semana o(a) Sr(a) costuma comer frutas?	1-7: Dias, 0: Nunca ou Menos de Uma Vez. 9: Ignorado	
P019	Em geral, quantas vezes por dia o(a) Sr(a) come frutas?	1: - Uma vez por dia. 2: - Duas vezes por dia. 3: - Três vezes por dia. 9: Ignorado	
P02002	Em quantos dias da semana o(a) Sr(a) costuma tomar refrigerante?	1-7: Dias, 0: Nunca ou Menos de Uma Vez. 9: Ignorado	
P02102	Que tipo de refrigerante o(a) Sr(a) costuma tomar?	1: Diet/Light/Zero, 2: Normal, 3: Ambos, 9: Ignorado	
P02401	Que tipo de leite o(a) Sr(a) costuma tomar?	1: Desnatado ou Semi-Desnatado, 2: Integral, 3: Os dois tipos. 9: Ignorado	
P02501	Em quantos dias da semana o(a) Sr(a) costuma comer alimentos doces como biscoito/bolacha recheado, chocolate, gelatina, balas e outros?	1-7: Dias, 0: Nunca ou Menos de Uma Vez. 9: Ignorado	
P02601	Considerando a comida preparada na hora e os alimentos industrializados, o(a) Sr(a) acha que o seu consumo de sal é:	1: Muito Alto, 2: Alto, 3: Adequado, 4: Baixo, 5: Muito Baixo, 9: Ignorado	
P027	Com que frequência o(a) Sr(a) costuma consumir alguma bebida alcoólica?	1: Não bebo nunca, 2: Menos de uma vez por mês, 3: Uma vez ou mais por mês, 9: Ignorado	
P029	Em geral, no dia que o(a) Sr(a) bebe, quantas doses de bebida alcoólica o(a) Sr(a) consome?	1-98: Doses, 99: Ignorado	
P035	Quantos dias por semana o(a) Sr(a) costuma (costumava)praticar exercício físico ou esporte?	1-7: Dias, 0: Nunca ou Menos de Uma Vez. 9: Ignorado	
P03701	Em geral, no dia que o(a) Sr(a) pratica exercício ou esporte, quanto tempo em horas dura essa atividade? Horas	0-24: Horas, 99: Ignorado	
P036	Qual o exercício físico ou esporte que o(a) Sr(a) pratica (praticava) com mais frequência? (Anotar apenas o primeiro citado)	1: Caminhada (Não vale para o trabalho), 2: Caminhada em esteira, 3: Corrida/-cooper, 4: Corrida em esteira, 5: Musculação, 6: Ginástica Aeróbica, 7: Hidroginástica, 8: Ginástica/Localizada, 9: Natação, 10: Artes Marciais, 11: Bicicleta, 12: Futebol, 13: Basquetebol, 14: Voliebol, 15: Tênis, 16: Dança, 17: Outro, 99: Ignorado	
P038	No seu trabalho, o(a) Sr(a) anda bastante a pé?	1: Sim, 2: Não, 9: Ignorado	
P039	No seu trabalho, o(a) Sr(a) faz faxina pesada, carrega peso ou faz outra atividade pesada que requer esforço físico intenso?	1: Sim, 2: Não, 9: Ignorado	

Código	Descrição	Valores	Observações
P03904	Em uma semana normal, em quantos dias, o(a) Sr(a) anda bastante a pé ou faz essas atividades pesadas ou que requerem esforço físico no seu trabalho?	1-7: Dias, 9: Ignorado	
P03905	Em um dia normal, quanto tempo o(a) Sr(a) passa andando bastante a pé ou realizando essas atividades pesadas ou que requerem esforço físico no seu trabalho?Horas	0-24: Horas, 99: Ignorado	
P040	Para ir ou voltar do trabalho, o(a) Sr(a) faz algum trajeto a pé ou de bicicleta?	1: Sim, para todo o trajeto, 2: Sim, para parte do trajeto, 3: Não, 9: Ignorado	
P04001	Quantos dias por semana o(a) Sr(a) faz algum trajeto a pé ou bicicleta?	1-7: Dias, 0: Nunca ou menos de uma vez por semana, 9: Ignorado	
P04101	Quanto tempo o(a) Sr(a) gasta, por dia, para percorrer este trajeto a pé ou de bicicleta, considerando a ida e a volta do trabalho?Horas	0-24: Horas, 99: Ignorado	
P042	Nas suas atividades habituais (tais como ir a algum curso, escola ou clube ou levar alguém a algum curso, escola ou clube), quantos dias por semana o(a) Sr(a) faz alguma atividade que envolva deslocamento a pé ou bicicleta? (Exceto o trabalho)	1-7: Dias, 0: Nunca ou menos de uma vez por semana, 9: Ignorado	
P04301	No dia em que o(a) Sr(a) faz essa(s) atividade(s), quanto tempo o(a) Sr(a) gasta no deslocamento a pé ou de bicicleta, considerando Ida e Volta?Horas	0-24: Horas, 99: Ignorado	
P044	Nas suas atividades domésticas, o(a) Sr(a) faz faxina pesada, carrega peso ou faz outra atividade pesada que requer esforço físico intenso? (não considerar atividade doméstica remunerada)	1: Sim, 2: Não, 9: Ignorado	
P04401	Em uma semana normal, nas suas atividades domésticas, em quantos dias o(a) Sr(a) faz faxina pesada ou realiza atividades que requerem esforço físico intenso? (não considerar atividade doméstica remunerada)	1-7: Dias, 9: Ignorado	
P04405	Quanto tempo gasta, por dia, realizando essas atividades domésticas pesadas ou que requerem esforço físico intenso? (não considerar atividade doméstica remunerada) Horas	0-24: Horas, 99: Ignorado	
P04501	Em média, quantas horas por dia o(a) Sr(a) costuma ficar assistindo televisão?	1: Menos de uma Hora, 2: De uma hora a menos de duas horas, 3: De duas horas a menos de três, 4: De três a menos de seis horas. 5: Seis horas ou mais. 6: Não vejo TV. 9: Ignorado	
P04502	Em um dia, quantas horas do seu tempo livre (excluindo o trabalho), o(a) Sr(a) costuma usar computador, tablet ou celular para lazer, tais como: utilizar redes sociais, para ver notícias, vídeos, jogar etc?	1: Menos de uma Hora, 2: De uma hora a menos de duas horas, 3: De duas horas a menos de três, 4: De três a menos de seis horas. 5: Seis horas ou mais. 6: Não costumo usar. 9: Ignorado	

Código	Descrição	Valores	Observações
P046	Perto do seu domicílio, existe algum lugar público (praça, parque, rua fechada, praia) para fazer caminhada, realizar exercício ou praticar esporte?	1: Sim, 2: Não, 9: Ignorado	
P04701	O(A) Sr(a) conhece algum programa público de estímulo à prática de atividade física no seu município?	1: Sim, 2: Não, 9: Ignorado	
P04801	O(A) Sr(a) participa desse programa público de estímulo à prática de atividade física no seu município?	1: Sim, 2: Não, 9: Ignorado	
P050	Atualmente, o(a) Sr(a) fuma algum produto do tabaco?	1: Sim, Diariamente, 2: Sim, menos que diariamente, 3: Não, nuna fumei, 9: Ignorado	
P052	E no passado, o(a) Sr(a) fumou algum produto do tabaco diariamente?	1: Sim, Diariamente, 2: Sim, menos que diariamente, 3: Não, nuna fumei, 9: Ignorado	
P053	Que idade o(a) Sr(a) tinha quando começou a fumar produto de tabaco diariamente?	1-98: Anos, 99: Ignorado	
P055	Quanto tempo depois de acordar o(a) Sr(a) normalmente fuma pela primeira vez?	1: Até cinco minutos. 2: De seis a 30 minutos 3: De 31 a 60 mins. 4: Mais de 60 mins. 9: Ignorado	
P058	Em média, quantos cigarros industrializados o(a) Sr(a) fumava por dia ou por semana?	1-98: Cigarros. 99: Ignorado	
P05901	Número de anos que parou de fumar	1-98: Anos. 99: Ignorado	
P067	ATUALMENTE, o (a) Sr (a) masca fumo, usa rapé ou algum outro produto do tabaco que não faz fumaça?	1: Sim, 2: Não, 9: Ignorado	
P06701	O(a) Sr(a) usa aparelhos eletrônicos com nicotina líquida ou folha de tabaco picado (cigarro eletrônico, narguilé eletrônico, cigarro aquecido ou outro dispositivo eletrônico para fumar ou vaporizar)?	1: Sim, 2: Não, 9: Ignorado	
P068	Com que frequência alguém fuma dentro do seu domicílio?	1: Diariamente, 2: Semanalmente, 3: Mensalmnete, 4: Menos que Mensalmente, 5: Nunca, 9: Ignorado	
Q00101	Quando foi a última vez que o (a) Sr(a) teve sua pressão arterial medida?	1: Menos de 6 Meses, 2: De 6 Meses a 1 Ano, 3: De 1 Ano a menos de 2 Anos, 4: De 2 anos a menos de 3 anos, 5: 3 anos ou mais, 6: Nunca fez, 9: Ignorado	
Q00201	Algum médico já lhe deu o diagnóstico de hipertensão arterial (pressão alta)?	1: Sim, 2: Não, 9: Ignorado	
Q00202	Essa hipertensão arterial (pressão alta) ocorreu apenas durante algum período de gravidez?	1: Sim, 2: Não, 9: Ignorado	
Q003	Que idade o(a) Sr(a) tinha no primeiro diagnóstico de hipertensão arterial (pressão alta)?	0: Menos de 1 ano, 1-99: Anos, 99: Ignorado	
Q00401	O(A) Sr(a) vai ao médico/serviço de saúde regularmente para acompanhamento da hipertensão arterial (pressão alta) ?	1: Sim, regularmente, 2: Não, só quando tem problema. 3: Nunca vai. 9: Ignorado	
Q00503	Algum médico já lhe receitou algum medicamento para a hipertensão arterial (pressão alta)?	1: Sim, 2: Não, 9: Ignorado	

Código	Descrição	Valores	Observações
Q00601	Nas duas últimas semanas, o(a) Sr(a) tomou os medicamentos para controlar a hipertensão arterial (pressão alta)?	1: Sim, 2: Não, 9: Ignorado	
Q01101	Quando foi a última vez que o(a) Sr(a) recebeu atendimento médico por causa da hipertensão arterial?	1: Menos de 6 Meses, 2: De 6 Meses a 1 Ano, 3: De 1 Ano a menos de 2 Anos, 4: De 2 anos a menos de 3 anos, 5: 3 anos ou mais, 6: Nunca fez, 9: Ignorado	
Q018010	Orientações para manter uma alimentação saudável	1: Sim, 2: Não, 9: Ignorado	
Q018011	Manter o peso adequado	1: Sim, 2: Não, 9: Ignorado	
Q018012	Ingerir menos sal	1: Sim, 2: Não, 9: Ignorado	
Q018013	Praticar atividade física regular	1: Sim, 2: Não, 9: Ignorado	
Q018014	Não fumar	1: Sim, 2: Não, 9: Ignorado	
Q018015	Não beber em excesso	1: Sim, 2: Não, 9: Ignorado	
Q018016	Fazer o acompanhamento regular com profissional de saúde	1: Sim, 2: Não, 9: Ignorado	
Q018017	Fazer uso de acupuntura, plantas medicinais e fitoterapia, homeopatia, meditação, yoga, tai chi chuan, liang gong ou alguma outra prática integrativa e complementar	1: Sim, 2: Não, 9: Ignorado	
Q01910	Foi pedido exame de sangue?	1: Sim, 2: Não, 9: Ignorado	
Q01911	Foi pedido exame de urina?	1: Sim, 2: Não, 9: Ignorado	
Q01912	Foi pedido eletrocardiograma?	1: Sim, 2: Não, 9: Ignorado	
Q01913	Foi pedido teste de esforço?	1: Sim, 2: Não, 9: Ignorado	
Q022	Em algum dos atendimentos para hipertensão arterial, houve encaminhamento para alguma consulta com médico especialista, tais como cardiologista ou nefrologista?	1: Sim, 2: Não, 9: Ignorado	
Q028	Em geral, em que grau a hipertensão ou alguma complicação da hipertensão limita as suas atividades habituais (como trabalhar, estudar, realizar afazeres domésticos etc.)?	1: Não Limita, 2: Um Pouco, 3: Moderadamente, 4: Intensamente, 5: Muito Intensamente, 9: Ignorado	
Q03001	Algum médico já lhe deu o diagnóstico de diabetes?	1: Sim, 2: Não, 9: Ignorado	
Q060	Algum médico já lhe deu o diagnóstico de colesterol alto?	1: Sim, 2: Não, 9: Ignorado	
Q061	Que idade o(a) Sr(a) tinha no primeiro diagnóstico de colesterol alto?	0: Menos de 1 ano, 1-99: Anos, 99: Ignorado	
Q06306	Algum médico já lhe deu o diagnóstico de uma doença do coração, tal como infarto, angina, insuficiência cardíaca ou outra?	1: Sim, 2: Não, 9: Ignorado	Atributo chave. É ele que tentamos prever.
Q068	Algum médico já lhe deu o diagnóstico de AVC (Acidente Vascular Cerebral) ou derrame?	1: Sim, 2: Não, 9: Ignorado	
Q074	Algum médico já lhe deu o diagnóstico de asma (ou bronquite asmática)?	1: Sim, 2: Não, 9: Ignorado	
Q087	Em geral, em que grau o problema na coluna limita as suas atividades habituais (tais como trabalhar, realizar afazeres domésticos, etc.)?	1: Não Limita, 2: Um Pouco, 3: Moderadamente, 4: Intensamente, 5: Muito Intensamente, 9: Ignorado	

Código	Descrição	Valores	Observações
Q092	Algum médico ou profissional de saúde mental (como psiquiatra ou psicólogo) já lhe deu o diagnóstico de depressão?	1: Sim, 2: Não, 9: Ignorado	
Q11604	Algum médico já lhe deu o diagnóstico de alguma outra doença crônica no pulmão, tais como enfisema pulmonar, bronquite crônica ou DPOC (Doença Pulmonar Obstrutiva Crônica)?	1: Sim, 2: Não, 9: Ignorado	
Q120	Algum médico já lhe deu diagnóstico de câncer?	1: Sim, 2: Não, 9: Ignorado	
Q124	Algum médico já lhe deu o diagnóstico de insuficiência renal crônica?	1: Sim, 2: Não, 9: Ignorado	
Q128	Algum médico já lhe deu algum diagnóstico de outra doença crônica (física ou mental) ou doença de longa duração (de mais de 6 meses de duração)?	1: Sim, 2: Não, 9: Ignorado	
Q132	Nas últimas duas semanas, o(a) Sr(a) fez uso de algum medicamento para dormir?	1: Sim, 2: Não, 9: Ignorado	
T001	O (a) Sr. (a) está com tosse há três semanas ou mais?	1: Sim, 2: Não	
T002	O (a) Sr (a) tem mancha com dormência ou parte da pele com dormência?	1: Sim, 2: Não	
T003	Algum médico já lhe deu o diagnóstico de doença de Chagas?	1: Sim, 2: Não	
T004	Nos últimos 12 meses, algum médico lhe deu diagnóstico de doença/infecção sexualmente transmissível?	1: Sim, 2: Não	

3 METODOLOGIA

3.1 Descrição dos Métodos Utilizados

Para a etapa de aprendizado, utilizamos uma Árvore de Decisão (*Decision Tree Classifier*), implementada por meio da biblioteca `scikit-learn`. O algoritmo foi configurado com os seguintes hiperparâmetros:

- **random_state**: 42
- **class_weight**: `balanced`, para lidar com o desbalanceamento inicial de classes.
- **max_depth**: 3, limitando a profundidade máxima da árvore para evitar overfitting.

Os hiperparâmetros foram ajustados com base na performance do modelo sobre os dados de treino, priorizando a melhor acurácia possível sem perder a interpretabilidade da árvore. A profundidade máxima de 3 foi escolhida para manter a simplicidade do modelo, facilitando a análise visual e evitando overfitting nos dados.

Além disso, utilizamos técnicas de balanceamento de classes com `SMOTE` (*Synthetic Minority Over-sampling Technique*) para criar instâncias sintéticas da classe minoritária. Isso garantiu um balanceamento adequado após a subamostragem (`RandomUnderSampler` com *sampling_strategy*=0.8) para evitar vieses no modelo.

3.2 Árvore de Decisão

A árvore de decisão é um algoritmo de aprendizado supervisionado usado para classificação e regressão. Ela organiza os dados em uma estrutura hierárquica, onde os nós internos representam atributos, os ramos indicam condições de decisão e as folhas correspondem às classes ou valores preditos.

O modelo é construído dividindo recursivamente os dados com base no atributo que maximiza um critério de separação, como o índice de Gini ou a entropia. O processo continua até que as partições atinjam pureza ideal ou outros critérios de parada sejam satisfeitos (ex.: profundidade máxima).

Aplicada a bases de dados, a árvore de decisão mapeia entradas para saídas em um formato interpretável, útil em casos onde a explicabilidade é essencial. Por exemplo, em uma base de pacientes com atributos como idade e sintomas, pode-se determinar regras para diagnosticar doenças.

3.3 Random Forest

O *Random Forest* é um algoritmo de aprendizado supervisionado baseado em um conjunto de árvores de decisão. Ele constrói múltiplas árvores em subconjuntos aleatórios dos dados e combina suas previsões por votação (para classificação) ou média (para regressão).

Cada árvore é gerada a partir de amostras com reposição (*bootstrap*) e com seleção aleatória de atributos em cada divisão, promovendo diversidade entre os modelos. Isso reduz problemas como *overfitting*, garantindo maior robustez e precisão.

Quando aplicado a bases de dados, o *Random Forest* é eficaz para problemas complexos, mantendo boa performance mesmo com dados ruidosos ou grandes conjuntos de atributos. Por exemplo, pode ser usado em uma base de clientes para prever a probabilidade de churn, combinando informações de comportamento e demografia.

3.4 K-Means

O algoritmo *K-Means* é utilizado para realizar *clustering*, particionando os dados em k grupos com base em características semelhantes. Ele funciona iterativamente em quatro etapas principais: (1) Inicialização, onde são definidos k centróides iniciais; (2) Atribuição, associando cada ponto ao centróide mais próximo com base na distância euclidiana; (3) Atualização, recalculando os centróides como a média dos pontos atribuídos a cada cluster; e (4) Repetição, até que os centróides se estabilizem ou o critério de parada seja atingido.

Quando aplicado a bases de dados, *K-Means* trabalha sobre representações vetoriais dos atributos, excluindo rótulos ou informações supervisionadas. Após sua execução, cada ponto recebe um rótulo indicando o cluster ao qual pertence. Os resultados auxiliam na identificação de padrões ou grupos naturais nos dados.

3.5 Utilização do ChatGPT

Devido ao número elevado de instâncias e tamanho da base, foi utilizada a ferramenta ChatGPT no contexto de geração do código referente ao algoritmo do `KNNImputer` da biblioteca `scikit-learn`. A partir disso, foi elaborada uma função que permitiu que a imputação fosse realizada em blocos, com o objetivo de otimizar o processo mencionado ao decorrer da base.

```
from sklearn.impute import KNNImputer
from tqdm import tqdm
import numpy as np

# Definindo o KNNImputer
imputer = KNNImputer(n_neighbors=6)
# Função personalizada para fazer a imputação
em blocos e acompanhar o progresso
def impute_with_progress(df, imputer,
    chunk_size=1000):
    # Calcula o número de blocos
    num_chunks = int(np.ceil(
    (df.shape[0] / chunk_size)
    )

    # Lista para armazenar blocos processados
    imputed_chunks = []
```

```
num_cols = df.shape[1]
# Número de colunas esperado

# Barra de progresso usando tqdm
for i in tqdm(range(num_chunks),
      desc="Imputando valores"):
    # Determina o índice de início
      e fim do bloco
    start = i * chunk_size
  end = min((i + 1) * chunk_size, df.shape[0])

    # Seleciona o bloco
    df_chunk = df.iloc[start:end]

    # Aplica o imputador no bloco
    imputed_chunk =
    imputer.fit_transform(df_chunk)

 # Verifica se o número de colunas está correto
    if imputed_chunk.shape[1] == num_cols:
        imputed_chunks.append(imputed_chunk)
    else:
      print(f"Erro: O bloco {i} tem um número
        diferente de colunas
        ({imputed_chunk.shape[1]})
        do que o esperado ({num_cols})")
        return None

# Junta todos os blocos imputados
return np.vstack(imputed_chunks)
```

4 RESULTADOS E DISCUSSÕES

4.1 Árvore de Decisão

O algoritmo encontrou que os melhores hiperparâmetros são: max-depth:5, min-samples-leaf:5, min-samples-split:2

Os resultados não foram os ideais, o modelo apresentou uma grande dificuldade em identificar instâncias com o valor "sim" (0), tendo uma precisão muito baixa nessa classe, enquanto isso, o modelo apresentou uma precisão altíssima na classe "não" (1), demonstrando talvez um "Overfitting". Abaixo se a árvore de decisão, na figura 3

```
Acurácia com melhores hiperparâmetros: 0.6950206833154589

Relatório de classificação:
              precision    recall  f1-score   support

           0       0.06      0.66      0.11       928
           1       0.99      0.70      0.82     31707

    accuracy                           0.70     32635
   macro avg       0.52      0.68      0.46     32635
```

Figure 3: Relatório de classificação

```
Regras da Árvore de Decisão:
|--- Q00201 <= 1.08
|   |--- Q028 <= 1.92
|   |   |--- Q01912 <= 1.25
|   |   |   |--- Q060 <= 1.58
|   |   |   |   |--- Q061 <= 70.50
|   |   |   |   |   |--- class: 0
|   |   |   |   |--- Q061 >  70.50
|   |   |   |   |   |--- class: 0
|   |   |   |--- Q060 >  1.58
|   |   |   |   |--- Q022 <= 1.08
|   |   |   |   |   |--- class: 0
|   |   |   |   |--- Q022 >  1.08
|   |   |   |   |   |--- class: 1
|   |   |--- Q01912 >  1.25
|   |   |   |--- Q061 <= 52.92
|   |   |   |   |--- Q00503 <= 1.08
|   |   |   |   |   |--- class: 1
|   |   |   |   |--- Q00503 >  1.08
|   |   |   |   |   |--- class: 1
|   |   |   |--- Q061 >  52.92
|   |   |   |   |--- P04401 <= 3.33
|   |   |   |   |   |--- class: 0
|   |   |   |   |--- P04401 >  3.33
|   |   |   |   |   |--- class: 1
|   |--- Q028 >  1.92
|   |   |--- Q01912 <= 1.08
|   |   |   |--- Q028 <= 3.50
|   |   |   |   |--- Q01913 <= 1.25
|   |   |   |   |   |--- class: 0
|   |   |   |   |--- Q01913 >  1.25
|   |   |   |   |   |--- class: 0
|   |   |   |--- Q028 >  3.50
|   |   |   |   |--- P039 <= 1.75
|   |   |   |   |   |--- class: 0
```

```
|   |   |   |   |--- P039 >  1.75
|   |   |   |   |   |--- class: 0
|   |   |--- Q01912 >  1.08
|   |   |   |--- P015 <= 2.50
|   |   |   |   |--- P058 <= 2.25
|   |   |   |   |   |--- class: 0
|   |   |   |   |--- P058 >  2.25
|   |   |   |   |   |--- class: 1
|   |   |   |--- P015 >  2.50
|   |   |   |   |--- P018 <= 3.50
|   |   |   |   |   |--- class: 1
|   |   |   |   |--- P018 >  3.50
|   |   |   |   |   |--- class: 0
|--- Q00201 >  1.08
|   |--- Q060 <= 1.17
|   |   |--- P053 <= 15.25
|   |   |   |--- P050 <= 1.50
|   |   |   |   |--- P029 <= 6.33
|   |   |   |   |   |--- class: 0
|   |   |   |   |--- P029 >  6.33
|   |   |   |   |   |--- class: 1
|   |   |   |--- P050 >  1.50
|   |   |   |   |--- P029 <= 7.33
|   |   |   |   |   |--- class: 0
|   |   |   |   |--- P029 >  7.33
|   |   |   |   |   |--- class: 1
|   |   |--- P053 >  15.25
|   |   |   |--- P058 <= 1.25
|   |   |   |   |--- P04502 <= 5.50
|   |   |   |   |   |--- class: 1
|   |   |   |   |--- P04502 >  5.50
|   |   |   |   |   |--- class: 0
|   |   |   |--- P058 >  1.25
|   |   |   |   |--- Q061 <= 27.50
|   |   |   |   |   |--- class: 1
|   |   |   |   |--- Q061 >  27.50
|   |   |   |   |   |--- class: 1
|   |--- Q060 >  1.17
|   |   |--- Q11604 <= 1.50
|   |   |   |--- Q003 <= 48.50
|   |   |   |   |--- P04801 <= 1.75
|   |   |   |   |   |--- class: 0
|   |   |   |   |--- P04801 >  1.75
|   |   |   |   |   |--- class: 0
|   |   |   |--- Q003 >  48.50
|   |   |   |   |--- class: 1
|   |   |--- Q11604 >  1.50
|   |   |   |--- Q092 <= 1.50
|   |   |   |   |--- P02002 <= 1.50
|   |   |   |   |   |--- class: 1
|   |   |   |   |--- P02002 >  1.50
|   |   |   |   |   |--- class: 1
|   |   |   |--- Q092 >  1.50
|   |   |   |   |--- Q132 <= 1.50
|   |   |   |   |   |--- class: 1
|   |   |   |   |--- Q132 >  1.50
|   |   |   |   |   |--- class: 1
```

4.2 Random Forest

O algoritmo determinou que os melhores hiperparâmetros foram uma profundidade máxima de 20 e um número de estimators de 20.

Os resultados foram muito parecidos com o da árvore de decisão simples – com resultados não ideais e com dificuldade de identificar as instâncias 'Sim', que são o número '0'. Também houve overfitting nas instâncias 'Não'.

```
Acurácia: 0.6720392216944998
Relatório de classificação:
              precision    recall  f1-score   support

           0       0.06      0.75      0.11       928
           1       0.99      0.67      0.80     31707

    accuracy                           0.67     32635
   macro avg       0.53      0.71      0.46     32635
weighted avg       0.96      0.67      0.78     32635
```

Figure 4: Relatório Random Forest

4.3 K-Neighbors

O algoritmo para esse modelo determinou que os melhores hiperparâmetros são n-neighbors:10 e p:1.

Quanto aos resultados do K-means, é possivel ver pelo relatório da Figura 8 que eles também não foram satisfatórios, com os mesmos problemas de precisão e recall para a instância '0'.

```
Acurácia: 0.5282059138961238
Relatório de classificação:
              precision    recall  f1-score   support

           0       0.04      0.71      0.08       928
           1       0.98      0.52      0.68     31707

    accuracy                           0.53     32635
   macro avg       0.51      0.62      0.38     32635
weighted avg       0.96      0.53      0.67     32635
```

Figure 5: Relatório K-Neighbors

4.4 K-Means

A figura 6 ilustra os centroides encontrados.

Os eixos "Componente Principal 1" e "Componente Principal 2" correspondem às direções principais (ou componentes principais) encontradas durante a aplicação da técnica de Análise de Componentes Principais (PCA).

Componente Principal 1: Representa a direção que explica a maior variabilidade possível nos dados originais. É a combinação linear das features originais que maximiza a variância ao longo desse eixo.

Componente Principal 2: Representa a segunda direção ortogonal ao Componente Principal 1, que explica a próxima maior parcela de variabilidade nos dados.

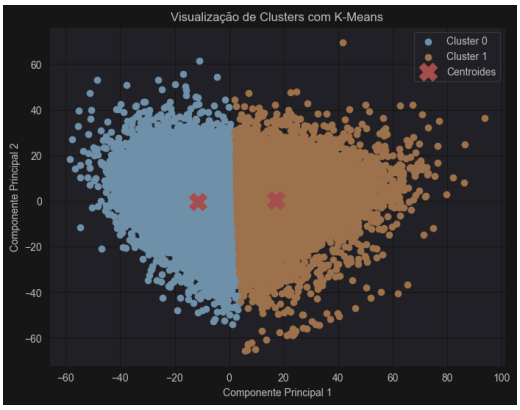

Figure 6: Centroides do Método K-Means

Também há a silhouetta

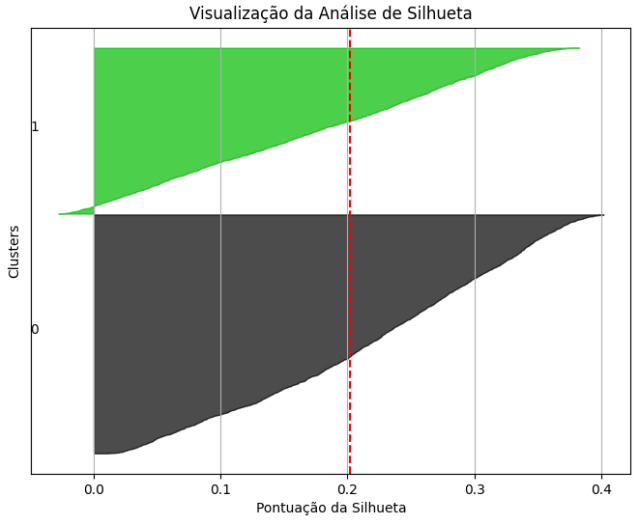

Figure 7: Silhouetta do K-NMeans

Há também as regras geradas por uma árvore de decisão para interpretar os grupos gerados pelo K-means, sendo que os rótulos dos clusters serão a variável alvo.

```
    Acurácia da árvore no conjunto de teste: 0.91
        |--- P00104 <= 75.08
|   |--- P00104 <= 71.42
|   |   |--- Q003 <= 27.75
|   |   |   |--- class: 0
|   |   |--- Q003 >  27.75
|   |   |   |--- class: 0
|   |--- P00104 >  71.42
|   |   |--- Q061 <= 41.92
|   |   |   |--- class: 1
|   |   |--- Q061 >  41.92
|   |   |   |--- class: 0
|--- P00104 >  75.08
|   |--- P00104 <= 81.08
|   |   |--- Q003 <= 49.58
|   |   |   |--- class: 1
|   |   |--- Q003 >  49.58
|   |   |   |--- class: 0
|   |--- P00104 >  81.08
|   |   |--- Q003 <= 56.58
|   |   |   |--- class: 1
|   |   |--- Q003 >  56.58
|   |   |   |--- class: 1
```

5 CÓDIGO DESENVOLVIDO

Os códigos referentes aos algoritmos implementados, juntamente do arquivo referente à base de dados estudada, se encontram no seguinte link do notebook do Deepnote: Código_desenvolvido

REFERENCES

[1] PNS. 2019. *BASES DE DADOS.* https://www.pns.icict.fiocruz.br/bases-de-dados/ Acessado em: 19 de setembro de 2024.

[2] scikit-learn developers. 2007 - 2024. *KNNImputer.* https://scikit-learn.org/stable/modules/generated/sklearn.impute.KNNImputer.html Acessado em: 24 de setembro de 2024.