

PONTIFÍCIA UNIVERSIDADE CATÓLICA DE MINAS GERAIS
NÚCLEO DE EDUCAÇÃO A DISTÂNCIA
Pós-graduação *Lato Sensu* em Ciência de Dados e Big Data

Eduardo Saldanha Bragato

**MACHINE LEARNING NA PREVISÃO DE ACIDENTES VASCULARES
CEREBRAIS**

Belo Horizonte

2024

Eduardo Saldanha Bragato

**MACHINE LEARNING NA PREVISÃO DE ACIDENTES VASCULARES
CEREBRAIS**

Trabalho de Conclusão de Curso apresentado
ao Curso de Especialização em Ciência de
Dados e Big Data como requisito parcial à
obtenção do título de especialista.

Belo Horizonte

2024

SUMÁRIO

1. Introdução.....	4
2. Coleta de dados	6
3. Processamento/Tratamento de Dados	9
4. Análise e Exploração dos Dados	13
5. Criação de Modelos de Machine Learning	35
6. Interpretação dos Resultados	53
7. Links e referências	553

1. Introdução

1.1. Contextualização

O Acidente Vascular Cerebral (AVC) ocorre quando vasos que levam sangue ao cérebro se entopem (AVC isquêmico) ou se rompem (AVC hemorrágico), provocando paralisia da área cerebral que ficou sem circulação sanguínea.¹ O AVC foi a segunda maior causa de morte entre os anos 2000 e 2019, responsável por cerca de 11% de todas as mortes no mundo². Com o surgimento da pandemia de COVID-19, a morte por AVC acabou caindo no ranking em 2021, ocupando a terceira posição no Brasil e quarta posição nos Estados Unidos da América³. Apesar disso, entre os meses de janeiro a julho de 2024 o infarto segue na segunda posição com cerca de 10,16% das causas de morte no mundo⁴;

O AVC também se mostrou a terceira principal causa no mundo quando se trata de perda de anos de vida, sendo sua prevenção primordial em vistas de aumentar a expectativa de vida das pessoas⁵. Para isso, o presente trabalho irá utilizar algoritmos de aprendizado de máquina para compreender as causas do AVC e prever se ele vai ou não acontecer. Dessa forma será possível gerar recomendações para que as pessoas sofram cada vez menos com essa adversidade.

Serão analisadas duas bases de dados, uma sobre norte-americanos e outra com dados criados artificialmente para compreender as causas do AVC e, posteriormente, será realizada uma análise transversal utilizando modelos de classificação para prever se cada indivíduo sofreu ou não com tal enfermidade. Com esses dados será possível utilizar o modelo em pacientes futuros e tratar suas causas principais de risco antes que o revés aconteça.

1.2. O problema proposto

O presente projeto irá tratar dados disponíveis no kaggle, acessados pelos seguintes domínios eletrônicos: <https://www.kaggle.com/datasets/fedesoriano/stroke-prediction-dataset>, uma fonte de dados confidenciais sobre pacientes nos Estados Unidos da América

com 5110 entradas, e <https://www.kaggle.com/datasets/teamincirbo/stroke-prediction>, uma fonte de dados criada artificialmente com 15000 entradas.

1.3. Objetivos

O problema proposto tem como objetivo gerar um modelo de aprendizado de máquina simples e capaz de prever a presença ou não de um Acidente Vascular Cerebral em um paciente a partir do uso de poucas informações. A ideia não é gerar um diagnóstico definitivo, pois ainda será dependente da clínica médica para decretar a presença dessa enfermidade, mas sim criar um modelo simples, porém robusto capaz de auxiliar os profissionais de saúde nesse diagnóstico e gerar recomendações aos pacientes e população de maior risco.

2. Coleta de Dados

O primeiro dataset consiste em uma fonte de dados confidenciais sobre pacientes nos Estados Unidos da América com 5110 entradas, pode ser obtido em arquivo .csv no link: <https://www.kaggle.com/datasets/fedesoriano/stroke-prediction-dataset>, com o formato original segundo a tabela abaixo:

Nome da coluna/campo	Descrição	Tipo
Id	Índice	int64
Gender	Gênero do paciente	object
Age	Idade do paciente	float64
Hypertension	Paciente é hipertenso	int64
Heart_disease	Paciente possui histórico de doenças cardíacas	int64
Ever_married	Paciente foi casado	object
Work_type	Tipo de emprego do paciente	object
Residence_type	Tipo de residência do paciente (urbano/rural)	object
Avg_glucose_level	Média glicêmica durante 24h	float64
Bmi	Índice de Massa Corporal - IMC	float64

Smoking_status	Condição de fumante	object
Stroke	Paciente sofreu AVC	int64

O segundo dataset consiste em uma fonte de dados criada artificialmente pela Team Incirbo com a intenção de auxiliar a criarem modelos de Machine Learning na prevenção e detecção de AVC. Possui 15000 entradas e pode ser obtido o arquivo .csv no link:

<https://www.kaggle.com/datasets/teamincirbo/stroke-prediction>, com o formato original da tabela abaixo:

Nome da coluna/campo	Descrição	Tipo
Patient ID	Índice	int64
Patient Name	Nome do paciente	object
Age	Idade do paciente	int64
Gender	Gênero do Paciente	object
Hypertension	Paciente é hipertenso	int64
Heart Disease	Paciente possui histórico de doenças cardíacas	int64
Marital Status	Condição civil do paciente	object
Work Type	Tipo de emprego do paciente	object

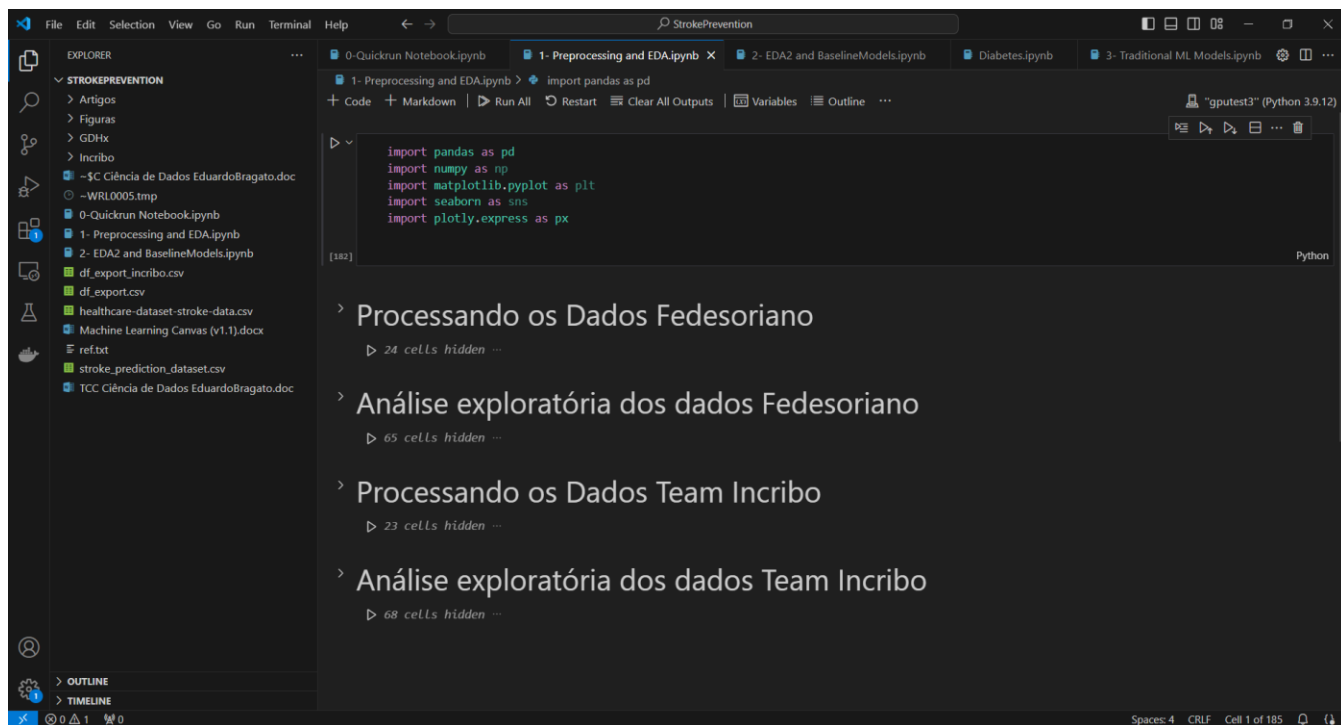
Residence Type	Tipo de residência do paciente	object
Average Glucose Level	Média glicêmica durante 24h	float64
Body Mass Index (BMI)	Índice de Massa Corporal - IMC	float64
Smoking Status	Condição de fumante	object
Alcohol Intake	Frequência de ingestão de álcool	object
Physical Activity	Intensidade de atividade física	object
Stroke History	Paciente tem histórico de AVC	int64
Family History of Stroke	Histórico familiar de AVC	object
Dietary Habits	Tipo de dieta que o paciente segue	object
Stress Levels	Nível de estresse do paciente	float64
Blood Pressure Levels	Pressão sanguínea (sistólica/diastólica)	object
Cholesterol Levels	Níveis de colesterol (HDL/LDL)	object
Symptoms	Sintomas do paciente	object
Diagnosis	Diagnóstico AVC ou não	object

3. Processamento/Tratamento de Dados

3.1 Ferramentas utilizadas

Como ferramentas para desenvolvimento dos *notebooks* em *Python* (formato. *ipynb*), foi escolhido o *Microsoft Visual Studio Code*, na sua versão 1.91.1, executando o *Python* na versão 3.9.12 (Figura 1).

Figura 1 Captura de tela da interface de trabalho do primeiro notebook



Fonte: Autor

3.2 Processamento e tratamento inicial dos dados

Nesse tópico será abordada a metodologia de processamento e tratamento das duas fontes de dados, ambas seguiram o mesmo fluxo de trabalho, respeitando as peculiaridades e diferenças entre elas.

3.2.1 Dados Fedesoriano

O primeiro passo da análise foi importar os dados, originalmente em formato *.csv*. Para isso, foi utilizada a biblioteca *Pandas*, a mais utilizada para importação e manipulação de *Dataframes* em *Python*.

Figura 2 - Importação e *preview* dos dados

```

Importação dos dados

df=pd.read_csv("healthcare-dataset-stroke-data.csv") #importa o arquivo .csv para um dataframe
[183] Python

df.head()
[184] Python
...

```

	id	gender	age	hypertension	heart_disease	ever_married	work_type	Residence_type	avg_glucose_level	bmi	smoking_status	stroke
0	9046	Male	67.0	0	1	Yes	Private	Urban	228.69	36.6	formerly smoked	1
1	51676	Female	61.0	0	0	Yes	Self-employed	Rural	202.21	NaN	never smoked	1
2	31112	Male	80.0	0	1	Yes	Private	Rural	105.92	32.5	never smoked	1
3	60182	Female	49.0	0	0	Yes	Private	Urban	171.23	34.4	smokes	1
4	1665	Female	79.0	1	0	Yes	Self-employed	Rural	174.12	24.0	never smoked	1

Fonte: Autor

O significado dos campos já foi abordado no tópico 2. Coleta de dados. Por se tratar de uma fonte de dados de pacientes reais, é comum que existam inconsistências nos dados, primeiramente foram tratados os dados ausentes e posteriormente cada coluna e seus dados foram analisadas individualmente. Para isso, a função `.isnull()` em conjunto com a função `.sum()`, nativas do objeto *dataframe*, realizaram o somatório das colunas preenchidas com *null*.

Figura 3 - Somatório de valores *null*

```

df.isnull().sum() #análise dos números faltantes dentro de cada coluna
[187] Python
...
id          0
gender      0
age         0
hypertension 0
heart_disease 0
ever_married 0
work_type   0
Residence_type 0
avg_glucose_level 0
bmi         201
smoking_status 0
stroke      0
dtype: int64

```

Fonte: Autor

Foram observadas 201 entradas com valor de IMC = *null*, representam cerca de 4% do total de entradas do conjunto de dados, por isso, foi realizado um *slice* dos dados eliminando as linhas em que a coluna "bmi" não possuía valores válidos: `df = df[df['bmi'].notna()]`;

3.2.1.1 Atributos categóricos

Em seguida, foram analisados os atributos categóricos, que possuem valores discretos como opção: *gender*, *hypertension*, *heart_disease*, *ever_married*, *work_type*, *Residence_type*, *smoking_status*, *stroke* (atributo alvo).

Figura 4 - Atributos categóricos

```

1 Female      2897
2 Male        2011
3 Other         1
4 Name: gender, dtype: int64
5 -----
6 0          4458
7 1           451
8 Name: hypertension, dtype: int64
9 -----
10 0          4666
11 1           243
12 Name: heart_disease, dtype: int64
13 -----
14 Yes        3204
15 No         1705
16 Name: ever_married, dtype: int64
17 -----
18 Private      2811
19 Self-employed  775
20 children     671
21 Govt_job     630
22 Never_worked  22
23 Name: work_type, dtype: int64
24 -----
25 Urban        2490
26 Rural        2419
27 Name: Residence_type, dtype: int64
28 -----
29 never smoked  1852
30 Unknown      1483
31 formerly smoked  837
32 smokes        737
33 Name: smoking_status, dtype: int64
34 -----
35 0          4700
36 1           209
37 Name: stroke, dtype: int64
38 -----
39 1

```

Fonte: Autor

Nessa parte da análise, será demonstrado como e porque as decisões foram tomadas, variável a variável.

Gênero (*gender*): Valores divididos entre Feminino e Masculino, com uma única entrada com valor de 'Outro' gênero. Não existem dados suficientes para treinar um modelo suficientemente capaz de usar esse valor para previsões consistentes, por isso, a tupla com esse valor foi eliminada do conjunto de dados.

Hipertensão (*hypertension*): Dados consistentes com cerca de 10% de hipertensos na amostra, não há nada a ser feito nessa etapa.

Doenças cardíacas (*heart_disease*): Dados consistentes com cerca de 5% dos pacientes possuindo doenças cardíacas, não há nada a ser feito nessa etapa.

Já foi casado (*ever_married*): Dados consistentes e equilibrados indicando que cerca de 65% dos pacientes já foram casados, incluindo-se nesse bloco os Casados e os Divorciados. Não há nada a ser feito nessa etapa.

Tipo de Emprego (*work_type*): Atributo possui cinco categorias diferentes: *Private* (trabalha em empresa privada), *Self-employed* (Autônomo/empreendedor), *Children* (criança), *Govt_job* (Trabalha para o governo / funcionário público) e *Never_worked* (nunca traba-

lhou). Aqui são observadas inconsistências tanto pela interseção entre os valores 'children' e 'Never_worked', mas também resumem os pacientes como pessoas que possuem apenas um emprego, sendo impossível nessa base de dados alguém trabalhar para o governo e para alguma empresa privada ou ser autônomo ao mesmo tempo, fato esse que pode sim existir na vida real. Nota-se que existem poucos dados de pessoas que nunca trabalharam, apenas 22 entradas, mas, de momento, nada será feito com isso.

Tipo de residência (Residence_type): Divide as moradias entre urbano e rural, conjunto de dados equilibrado com quase 50% para cada. Não há nada a ser feito nessa etapa.

Condição de fumante (smoking_status): Divide o conjunto de dados em quatro categorias diferentes: *Never Smoked* (nunca fumou), *Unknown* (condição desconhecida), *Formerly Smoked* (passado de fumante) e *smokes* (fuma).

AVC (stroke): Diagnóstico de AVC ou não. Cerca de 4% da amostra tem diagnóstico positivo, esse desbalanço será um problema para ser tratado durante a etapa de construção de modelo de aprendizado de máquina. De momento, não há nada a ser feito sobre esse atributo, que é o atributo alvo.

3.2.1.2 Atributos numéricos ou contínuos

A etapa seguinte consistiu em realizar uma análise parecida para as variáveis numéricas.

Figura 5 - Atributos numéricos

	id	age	hypertension	heart_disease	avg_glucose_level	bmi	stroke
count	4909.000000	4909.000000	4909.000000	4909.000000	4909.000000	4909.000000	4909.000000
mean	37064.313506	42.865374	0.091872	0.049501	105.305150	28.893237	0.042575
std	20995.098457	22.555115	0.288875	0.216934	44.424341	7.854067	0.201917
min	77.000000	0.080000	0.000000	0.000000	55.120000	10.300000	0.000000
25%	18605.000000	25.000000	0.000000	0.000000	77.070000	23.500000	0.000000
50%	37608.000000	44.000000	0.000000	0.000000	91.680000	28.100000	0.000000
75%	55220.000000	60.000000	0.000000	0.000000	113.570000	33.100000	0.000000
max	72940.000000	82.000000	1.000000	1.000000	271.740000	97.600000	1.000000

Fonte: Autor

Idade (age): Foram encontrados valores relativamente inesperados como idades em números decimais, representando bebês com meses de vida e outras crianças com menos de 5 anos de idade. Embora não sejam dados comuns de se falar quando o assunto é AVC, não são inconsistentes, logo não há nada a ser feito nessa etapa.

Média glicêmica(*avg_glucose_level*): Foram encontrados dados variando entre 55 até 271mg/dl. São valores válidos para média glicêmica sem nenhum indicativo de inconsistência, logo, não há nada a ser feito nessa etapa.

Índice de Massa Corporal – IMC (*bmi*): Aqui foram encontrados dados variando de 10,3 até 97,6. De início, podem parecer absurdos os valores de IMC próximos a 100, entretanto, tais valores não são impossíveis, visto que o maior IMC já registrado pertenceu a Eman Ahmed Abd El Aty, uma egípcia que chegou a atingir 251.1 de IMC. Para fins de comparação, uma pessoa de 1,75m de altura precisaria pesar cerca de 275kg para ter um IMC de 90, mas não há indicativos que os valores elevados de IMC sejam inconsistência de dados. Sobre os valores mínimos, valores próximos de 10 são extremamente raros para adultos, sendo atingidos basicamente apenas sobre condições extremas. Entretanto, ao analisar o atributo ‘idade’, foi percebido que existem crianças nesse conjunto de dados, e esses valores podem sim ser plausíveis para crianças. Dessa forma, foi necessária uma análise extra dos dados mínimos:

Figura 6 - IMC abaixo de 13



```
df.loc[df['bmi'] <= 13].sort_values('bmi')
```

	id	gender	age	hypertension	heart_disease	ever_married	work_type	Residence_type	avg_glucose_level	bmi	smoking_status	stroke
1609	38043	Female	1.24	0	0	No	children	Rural	122.04	10.3	Unknown	0
3307	3205	Female	79.00	0	0	Yes	Self-employed	Urban	79.03	11.3	Unknown	0
2187	59993	Male	40.00	0	0	Yes	Private	Rural	60.96	11.5	never smoked	0
657	20364	Female	4.00	0	0	No	children	Urban	107.25	12.0	Unknown	0
922	45893	Female	8.00	0	0	No	children	Urban	106.51	12.3	Unknown	0
3319	53924	Female	1.08	0	0	No	children	Urban	159.39	12.8	Unknown	0
3968	41500	Male	0.16	0	0	No	children	Rural	69.79	13.0	Unknown	0

Fonte: Autor

Dos sete menores valores de IMC, pode-se perceber que cinco deles são dados sobre crianças e outros dois representam adultos, que realmente podem ter algum motivo para valores tão extremos. Portanto, não existem indícios concretos que esses dados são inconsistentes.

3.2.2 Dados Team Inciribo

Existe uma grande diferença entre ambos conjuntos de dados, o primeiro se trata de dados de pacientes reais, enquanto este foi criado artificialmente. Portanto antes de pensar em agrupar esses dados, deve ser analisada a verossimilhança entre eles e identificar tendências em comum.

O primeiro passo da análise foi importar os dados, originalmente em formato .csv. Para isso, foi utilizada a biblioteca *Pandas*, tal como foi feito para o outro conjunto de dados.

Figura 7 - Importação dos dados Team Incribo

Importação dos dados

```
df2=pd.read_csv("stroke_prediction_dataset.csv") #importa o arquivo .csv para um dataframe
```

[52] ✓ 0.1s Python

```
df2.head()
```

[53] ✓ 0.0s Python

...

	Patient ID	Patient Name	Age	Gender	Hypertension	Heart Disease	Marital Status	Work Type	Residence Type	Average Glucose Level	...	Alcohol Intake	Physical Activity	Stroke History	Family History of Stroke	Di H
0	18153	Mamooty Khurana	56	Male	0	1	Married	Self-employed	Rural	130.91	...	Social Drinker	Moderate	0	Yes	V
1	62749	Kaira Subramaniam	80	Male	0	0	Single	Self-employed	Urban	183.73	...	Never	Low	0	No	
2	32145	Dhanush Balan	26	Male	1	1	Married	Never Worked	Rural	189.00	...	Rarely	High	0	Yes	
3	6154	Ivana Baral	73	Male	0	0	Married	Never Worked	Urban	185.29	...	Frequent Drinker	Moderate	0	No	
4	48973	Darshit Jayaraman	51	Male	1	1	Divorced	Self-employed	Urban	177.34	...	Rarely	Low	0	Yes	Pescat

Spaces: 8 CRLF Cell 87 of 185

Fonte: Autor

Nesse ponto, foram identificadas todas as colunas desse conjunto de dados e, em seguida, buscou-se deixar as colunas similares entre os dois *datasets*, para que seja possível comparar os mesmos atributos e, posteriormente, unificar esses dados e utilizar um modelo de aprendizado de máquina em comum entre ambos.

Figura 8 - Atributos totais do dataset

```

result2 = df2.dtypes

print("Output:")
print(result2)

```

[54] ✓ 0.0s Python

... Output:

Patient ID	int64
Patient Name	object
Age	int64
Gender	object
Hypertension	int64
Heart Disease	int64
Marital Status	object
Work Type	object
Residence Type	object
Average Glucose Level	float64
Body Mass Index (BMI)	float64
Smoking Status	object
Alcohol Intake	object
Physical Activity	object
Stroke History	int64
Family History of Stroke	object
Dietary Habits	object
Stress Levels	float64
Blood Pressure Levels	object
Cholesterol Levels	object
Symptoms	object
Diagnosis	object
dtype:	object

Fonte: Autor

O significado desses atributos já foi descrito no Capítulo 2- Coleta de dados. A partir disso foram eliminadas as colunas inexistentes no outro conjunto de dados e renomeadas as colunas equivalentes para que sigam o mesmo padrão. Entre as colunas eliminadas, citam-se:

Patient Name: Nome do paciente, atributo absolutamente irrelevante para a análise.

Alcohol Intake: Uso de álcool, atributo que seria relevante para análise de fatores de riscos segundo algumas organizações de saúde.

Physical Activity: Nível de intensidade de atividade física, atributo correlato com IMC, obesidade e outros fatores.

Stroke History: Histórico pessoal de AVC, poderia ser um atributo interessante para ser analisado.

Family History of Stroke: Histórico familiar de AVC, outro atributo com alto potencial de relevância.

Dietary Habits: Atributo que diferencia a alimentação do paciente com base em arquétipos dietéticos, como dietas onívoras, cetogênica, vegetariana, vegana, paleolítica e outras. Caso se mostrasse um atributo relevante por si só, talvez merecesse um estudo particular para buscar as relações e consequências desses padrões dietéticos com a saúde dos pacientes.

Stress Levels: Atributo que representa numericamente o nível de estresse do paciente, uma análise razoavelmente esquisita, pois geralmente estresse é medido de maneira qualitativa (alto, médio ou baixo), e caso fosse necessário realizar uma quantificação, talvez deveriam ser medidos os níveis de cortisol do paciente.

Blood Pressure Levels: Atributo correlato com hipertensão, potencialmente descartável, embora traga uma análise numérica para um atributo que acabou sendo apenas categórico.

Cholesterol Levels: Atributo capaz de identificar dislipidemia dos pacientes, é um dado que fez falta no primeiro conjunto de dados e infelizmente será descartado aqui.

Symptoms: Coluna que apresenta os sintomas dos pacientes ao serem atendidos, potencial valor para clínica médica, mas altamente subjetiva para análise de dados, nesse caso os pacientes nem existem, pois são dados sintéticos, logo essa coluna é facilmente eliminada.

Figura 9 - Eliminando atributos extras e renomeando colunas remanescentes

```

[58] ✓ 0.0s Python
reduced_df2 = df2.drop(columns=['Patient Name', 'Alcohol Intake', 'Physical Activity', 'Stroke History',
.....:                        'Family History of Stroke', 'Dietary Habits', 'Stress Levels',
.....:                        'Blood Pressure Levels', 'Cholesterol Levels', 'Symptoms'], axis=1, inplace= False) #elimina columnas extras do dataset si

[59] ✓ 0.0s Python
reduced_df2.columns

... Index(['Patient ID', 'Age', 'Gender', 'Hypertension', 'Heart Disease',
.....:      'Marital Status', 'Work Type', 'Residence Type',
.....:      'Average Glucose Level', 'Body Mass Index (BMI)', 'Smoking Status',
.....:      'Diagnosis'],
.....:      dtype='object')

[60] ✓ 0.0s Python
renamed_df2 = reduced_df2.rename(columns={'Patient ID': 'id', 'Age': 'age', 'Gender': 'gender', 'Hypertension': 'hypertension', 'Heart Disease':
.....:      'Marital Status': 'ever_married', 'Work Type': 'work_type', 'Residence Type': 'Residence_type',
.....:      'Average Glucose Level': 'avg_glucose_level', 'Body Mass Index (BMI)': 'bmi', 'Smoking Status': 'smoking_status',
.....:      'Diagnosis': 'stroke'})

```

Fonte: Autor

3.2.2.1 Atributos categóricos

Em seguida, foram analisados os atributos categóricos, que possuem valores discretos como opção: *gender*, *hypertension*, *heart_disease*, *ever_married*, *work_type*, *Residence_type*, *smoking_status*, *stroke* (atributo alvo).

Nessa parte da análise, será demonstrado como e porque as decisões foram tomadas, variável a variável.

Gênero (*gender*): Valores divididos entre Feminino e Masculino, bem equilibrado com cerca de 50% para cada, não há nada a ser feito nessa etapa.

Hipertensão (*hypertension*): Dados consistentes com cerca de 25% de hipertensos na amostra, não há nada a ser feito nessa etapa.

Doenças cardíacas (*heart_disease*): Dados equilibrados com cerca de 50% dos pacientes possuindo doenças cardíacas, de imediato, nota-se a alta prevalência de doenças cardíacas nesse conjunto de dados destoando dos 5% presente nos dados reais. Entretanto, nada foi feito nessa etapa.

Já foi casado (*ever_married*): Dados consistentes e equilibrados indicando que cerca de 66% dos pacientes já foram casados, incluindo-se nesse bloco os Casados e os Divorciados. Não há nada a ser feito nessa etapa.

Tipo de Emprego (*work_type*): Atributo possui quatro categorias diferentes: *Private* (trabalha em empresa privada), *Self-employed* (Autônomo/empreendedor), *Government Job* (Trabalha para o governo / funcionário público) e *Never Worked* (nunca trabalhou). Base de dados muito equilibrada com quase 25% para cada uma das quatro categorias. Não há nada a ser feito nessa etapa.

Tipo de residência (*Residence_type*): Divide as moradias entre urbano e rural, conjunto de dados equilibrado com quase 50% para cada. Não há nada a ser feito nessa etapa.

Condição de fumante (*smoking_status*): Divide o conjunto de dados em três categorias diferentes: *non-Smoker* (não fumante), *Formerly Smoked* (passado de fumante) e *Currently Smokes* (fuma). Muito equilibrado, com cerca de 33% para cada categoria. Não há nada a ser feito nessa etapa.

AVC (*stroke*): Diagnóstico de AVC ou não. Cerca de 50% da amostra tem diagnóstico positivo. De momento, não há nada a ser feito sobre esse atributo, que é o atributo alvo.

Nesse momento, antes de realizar a análise, cabe a suspeita e o estranhamento sobre o nível do balanceamento da amostra, destoando muito dos dados reais ou de quaisquer tipos de dados retirados do mundo real, que sempre envolve desbalanceamento e inconsistências, que não são encontrados nesse conjunto de dados artificiais.

3.2.2.2 Atributos numéricos ou contínuos

A etapa seguinte consistiu em realizar uma análise parecida para as variáveis numéricas.

Figura 10 - Atributos numéricos



```
renamed_df2.drop(columns=['id', 'hypertension', 'heart_disease']).describe()
```

	age	avg_glucose_level	bmi
count	15000.000000	15000.000000	15000.000000
mean	54.035667	129.445209	27.474302
std	21.063111	40.487792	7.230201
min	18.000000	60.000000	15.010000
25%	36.000000	94.517500	21.160000
50%	54.000000	128.900000	27.420000
75%	72.000000	164.592500	33.720000
max	90.000000	200.000000	40.000000

Fonte: Autor

Idade (*age*): Foram encontrados valores entre 18 a 90 anos, observa-se que os interquartis estão equidistantes entre si e isso, até para uma base de dados artificiais, é digno de algum questionamento sobre a metodologia que foi utilizada para gerar tais dados. Nota-se que não existem crianças, apenas adultos acima de 18 anos, diferente do primeiro conjunto de dados.

Média glicêmica (*avg_glucose_level*): Foram encontrados dados variando entre 60 até 200mg/dl. São valores válidos para média glicêmica sem nenhum indicativo de inconsistência, mas era esperado uma amplitude maior de valores para um conjunto de 15000 entradas.

Índice de Massa Corporal – IMC (*bmi*): Aqui foram encontrados dados variando de 15 até 40. Novamente, seria esperado encontrar valores mais extremos, especialmente acima do IMC 40 para um conjunto de 15000 entradas.

Embora dignos de suspeição, não há que se falar em inconsistência de dados nessa etapa, visto que os dados se apresentam de forma íntegra, sem valores faltantes ou valores inválidos.

4. Análise e Exploração dos Dados

Nessa seção serão abordados os procedimentos de análise e exploração de dados e as possíveis conclusões tiradas a partir disso. Antes de partir para a análise, foi realizada a adição de uma nova coluna com um conceito que foi popularizado durante a pandemia de COVID-19. O atributo “comorbidade” foi adicionado ao conjunto de dados, representando um somatório dos fatores de risco que cada paciente tem em relação ao AVC. Para isso, foram levantados quais seriam os fatores de risco segundo a AHA – American Heart Association⁶: hipertensão, dislipidemia, uso de tabaco, diabetes, obesidade, relação idade/gênero e presença de doenças cardíacas.

No que se refere a hipertensão, o conjunto de dados original já é claro o suficiente, com uma coluna significando exatamente isso. Ao prosseguir para a dislipidemia, encontramos um impasse, pois não existem dados de colesterol nessa base de dados, portanto esse fator de risco infelizmente não foi considerado. O uso de tabaco é bem delimitado no conjunto de dados com dados de pessoas que nunca fumaram, fumaram no passado, fumam ou dados desconhecidos, para o atributo comorbidade foi considerado apenas os fumantes ativos.

Sobre a presença ou não de diabetes, o conjunto de dados não é específico no que se refere a esse fator de risco, mas possui uma coluna capaz de determinar a existência desse fator de risco. O atributo “média glicêmica” (*avg_glucose_level*), quando maior que 126mg/dl pode ser considerado indicador de diabetes⁷, pontuando o atributo ‘comorbidade’.

Analogamente, o conjunto de dados também não possui indicador direto de obesidade, mas possui dados de IMC (*bmi*), pacientes com IMC acima de 30 foram considerados obesos e, portanto, portadores de fator de risco para AVC⁸.

A relação idade/gênero considerada como fator de risco foi: Se homem, acima de 55 anos e, se mulher, acima de 65 anos. A presença de doenças cardíacas é um conceito muito abrangente podendo significar diferentes tipos de disfunções e anomalias do coração, que por sua vez podem ou não significar fator de risco para AVC⁶. Para esse conjunto de dados, foi considerado que a presença de doença cardíaca (*heart_disease*) é fator de risco para AVC.

Dessa forma, o atributo “Comorbidade” foi criado da seguinte maneira:

Figura 11 - Implementação do novo atributo

Variáveis categóricas

Antes de analisarmos os dados, será criada uma coluna referente às comorbidades do AVC, definidas pela OMS:

- Hipertensão;
- Dislipidemia(Infelizmente, não está presente nesses dados);
- Uso de tabaco;
- Diabéticos > 126 mg/dL glicemia;
- Obesidade IMC > 30;

Além disso, também serão consideradas comorbidades:

- Idade > 65 anos para mulheres;
- Idade > 55 anos para homens;
- Doenças cardíacas (heart_disease);

```
df_copy['Comorbidades'] = 0
df_copy['Comorbidades'].loc[df_copy['hypertension']== 1] += 1
df_copy['Comorbidades'].loc[df_copy['smoking_status']=='smokes'] += 1
df_copy['Comorbidades'].loc[df_copy['avg_glucose_level']>126] += 1
df_copy['Comorbidades'].loc[df_copy['bmi']>30] += 1
df_copy['Comorbidades'].loc[df_copy['heart_disease']==1] += 1
df_copy['Comorbidades'].loc[(df_copy['age']>=55) & (df_copy['gender']=='Male')] += 1
df_copy['Comorbidades'].loc[(df_copy['age']>=65) & (df_copy['gender']=='Female')] += 1
```

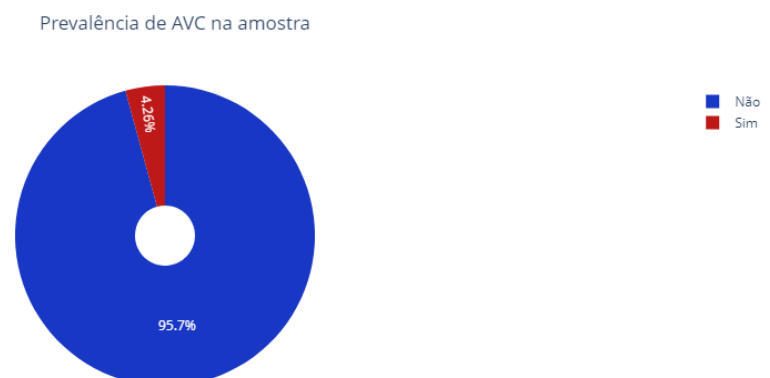
Fonte: Autor

4.1 Dados Fedesoriano

Nessa seção serão analisados em sequência cada atributo e as conclusões de cada análise.

AVC (variável alvo): Fara fins de contexto geral dos outros atributos, primeiramente foi analisada a variável alvo para identificar a prevalência de AVC ou não-AVC na amostra. Apenas 4,26% dos pacientes desse conjunto de dados têm esse diagnóstico positivo.

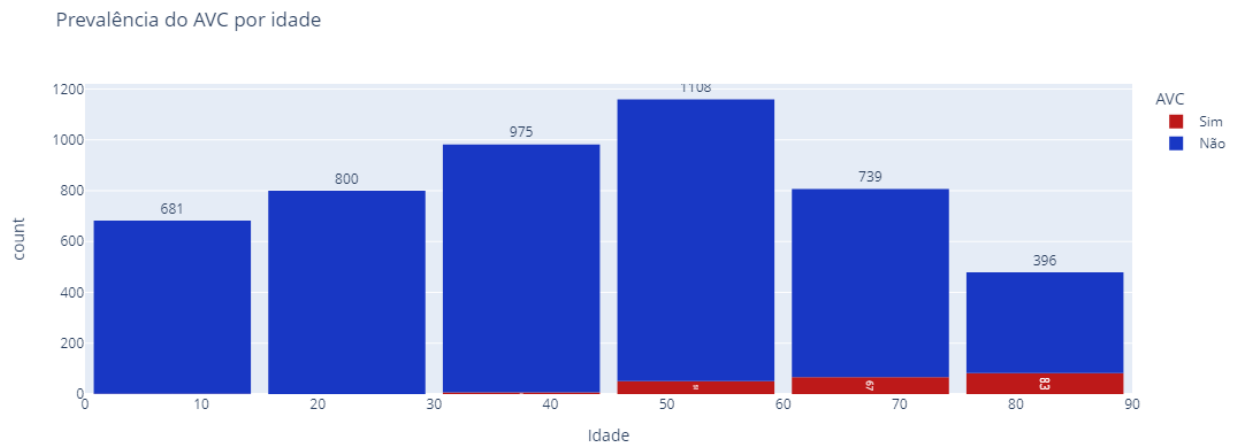
Figura 12 - Prevalência de AVC na amostra



Fonte: Autor

Idade: aqui buscou-se identificar a prevalência da variável alvo estratificada por idades, por isso, lançou-se mão de um histograma (figura 13).

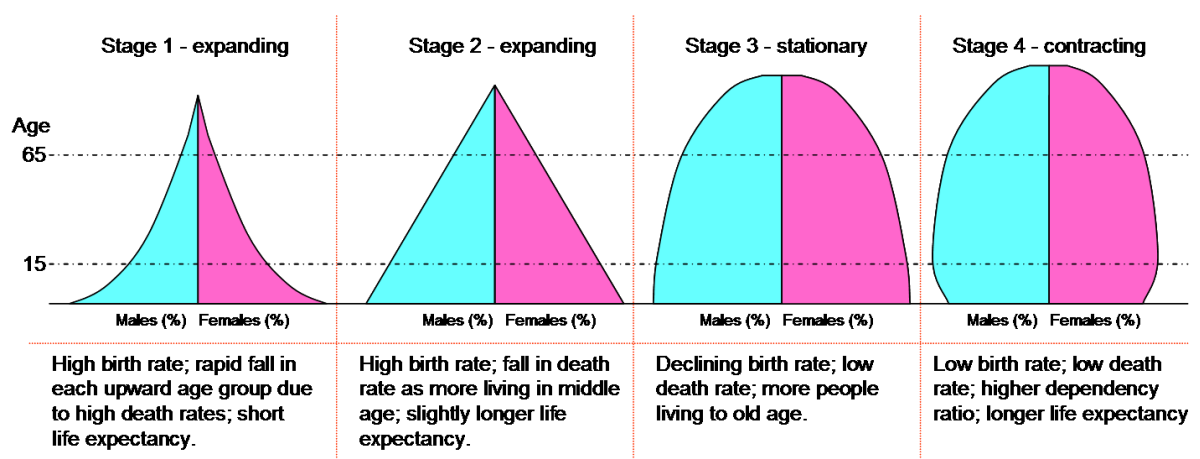
Figura 13 - Prevalência do AVC por idade



Fonte: Autor

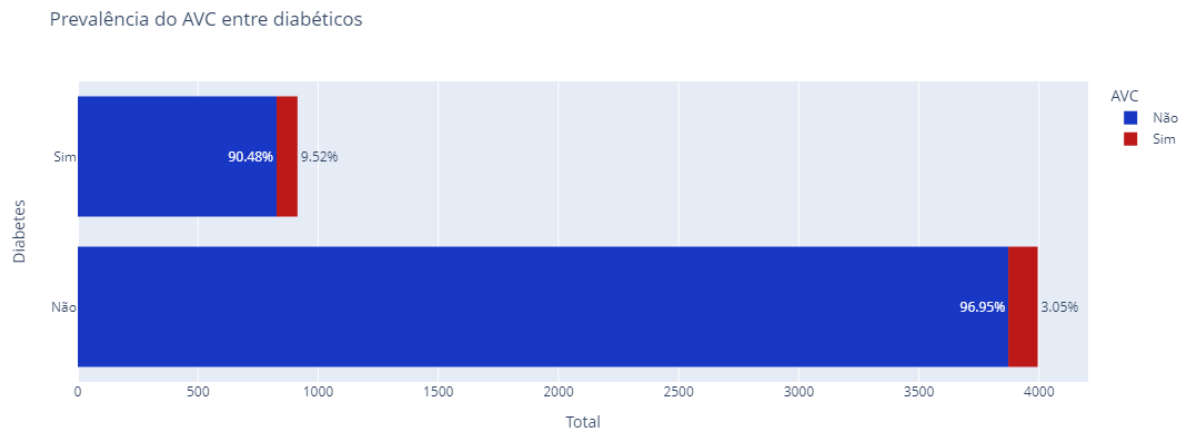
Os dados confirmam a característica de uma pirâmide etária de um país desenvolvido, como são os Estados Unidos da América, indicando uma 'Pirâmide envelhecida' ou estágio 4 (figura 14), mas confirmam também a expectativa da maior prevalência do AVC na parcela mais idosa da população. Nesse conjunto de dados, existe um único caso de AVC dos 0 a 14 anos, nenhum dos 15 a 29 anos, estando todo o nos 30 anos ou mais, com destaque para as últimas duas categorias: 60 a 74 e 75 a 90 anos de idade.

Figura 14 - Pirâmides etárias



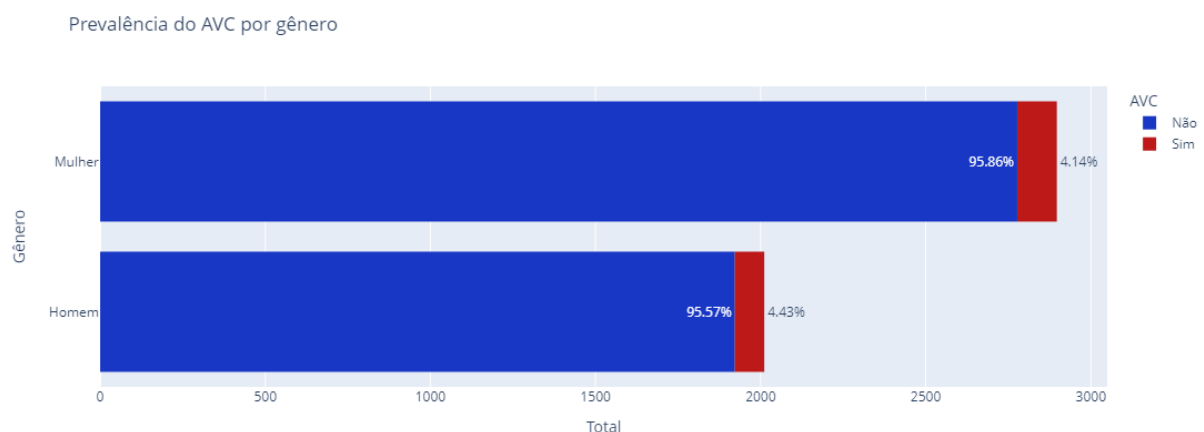
Fonte: Wikipédia⁹

Diabetes: Buscou-se comparar a prevalência do AVC entre os diabéticos e não diabéticos.

Figura 15 - Correlação entre diabetes e AVC**Fonte: Autor**

A prevalência de derrames em diabéticos é maior no grupo de pessoas que tem diabetes (considerada aqui glicemia acima de 126mg/dL). Cerca de 9,5% dos diabéticos tiveram acidentes vascular cerebral enquanto esse valor para não diabéticos foi de 3.1%. Esse achado corrobora com a ideia de a diabetes ser fator de risco para tal enfermidade.

Gênero: buscou-se comparar a prevalência do AVC estratificada por gênero.

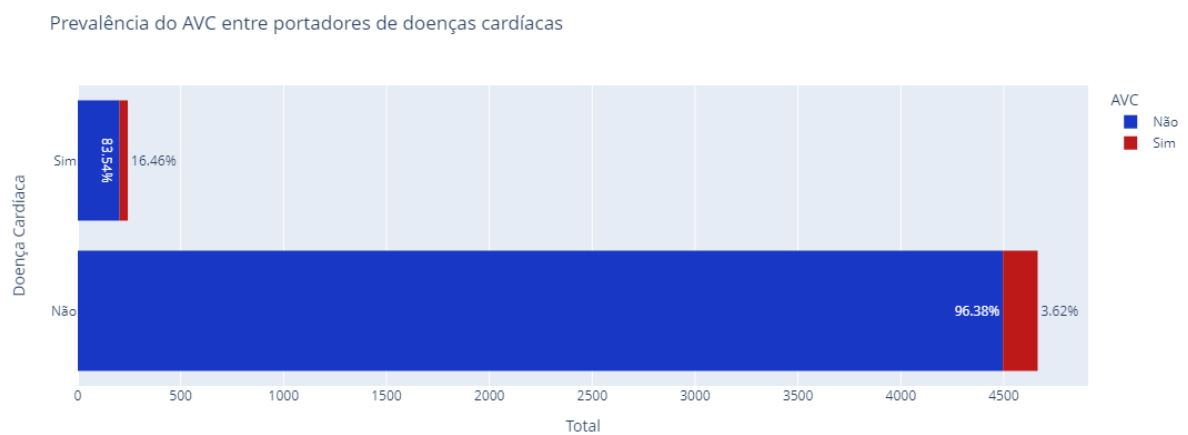
Figura 16 - Prevalência AVC por Gênero**Fonte: Autor**

Pode-se perceber que o banco de dados é formado majoritariamente por mulheres e que a prevalência do AVC é maior dentro do público masculino (4,43%) contra 4,14% das mulhe-

res, indicando uma leve diferença entre homens e mulheres nesse quesito, tal diferença já é abordada suavemente no atributo criado “Comorbidades”, pois na combinação de gênero e idade, foi considerada uma idade menor para homens já serem contabilizados como fator de risco.

Doenças cardíacas: buscou-se perceber as diferenças entre os grupos de pessoas que possuem e que não possuem doenças cardíacas.

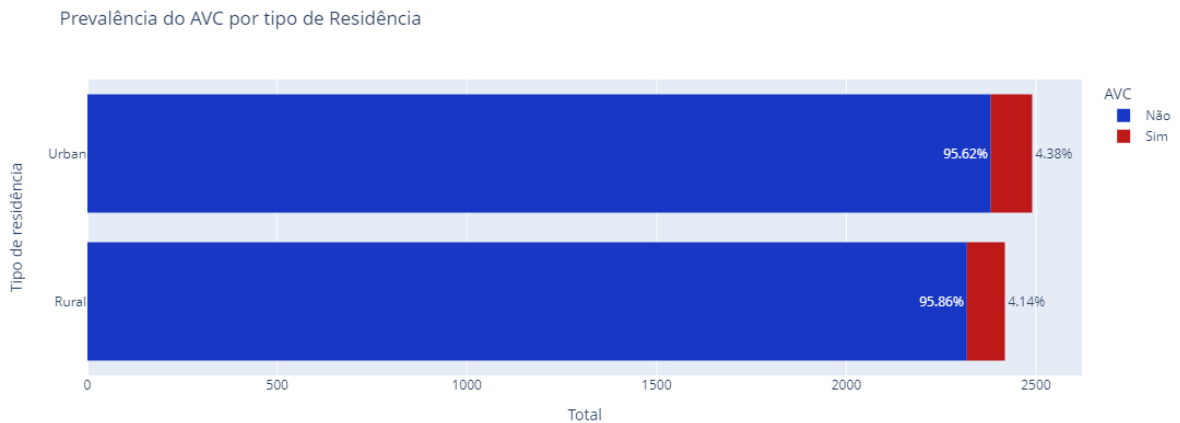
Figura 17 - Prevalência entre AVC e Doenças cardíacas



Fonte: Autor

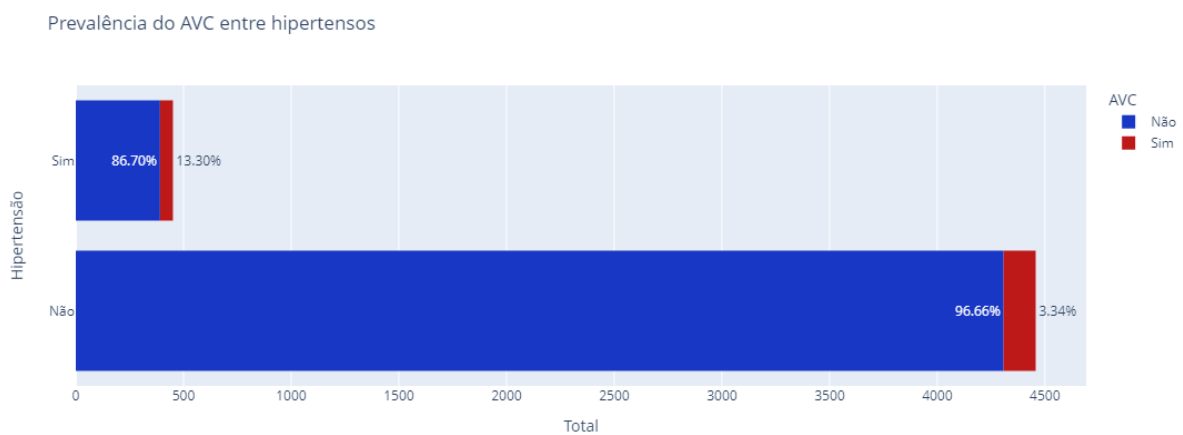
Embora possam representar diferentes anomalias do coração, o atributo “Doenças cardíacas” nessa amostra representou risco cerca de 4,5 vezes maior para o paciente sofrer um AVC.

Tipo de residência: Analisa as diferenças da prevalência do AVC para ambiente urbano e rural.

Figura 18 - Prevalência AVC por residência**Fonte: Autor**

Embora exista uma leve desvantagem na prevalência de AVC para os moradores da cidade, esse atributo não se mostrou indispensável para a análise e o modelo de aprendizado de máquina foi treinado e testado sem ele.

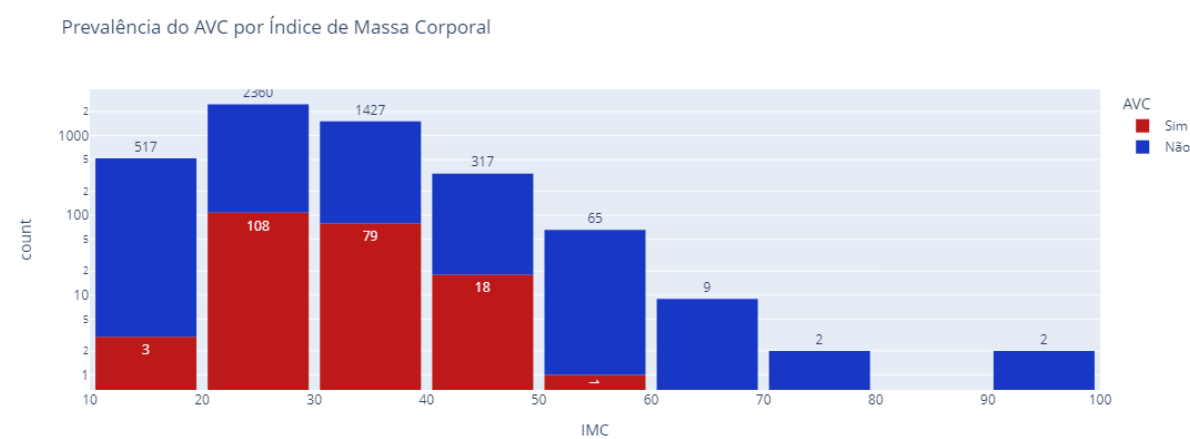
Hipertensão: Análise da prevalência do AVC entre pacientes hipertensos e não-hipertensos.

Figura 19 - Prevalência do AVC em hipertensos**Fonte: Autor**

Nesse conjunto de dados, a hipertensão representa um risco cerca de quatro vezes maior para o paciente sofrer um AVC, validando a hipótese de a hipertensão ser um fator de risco.

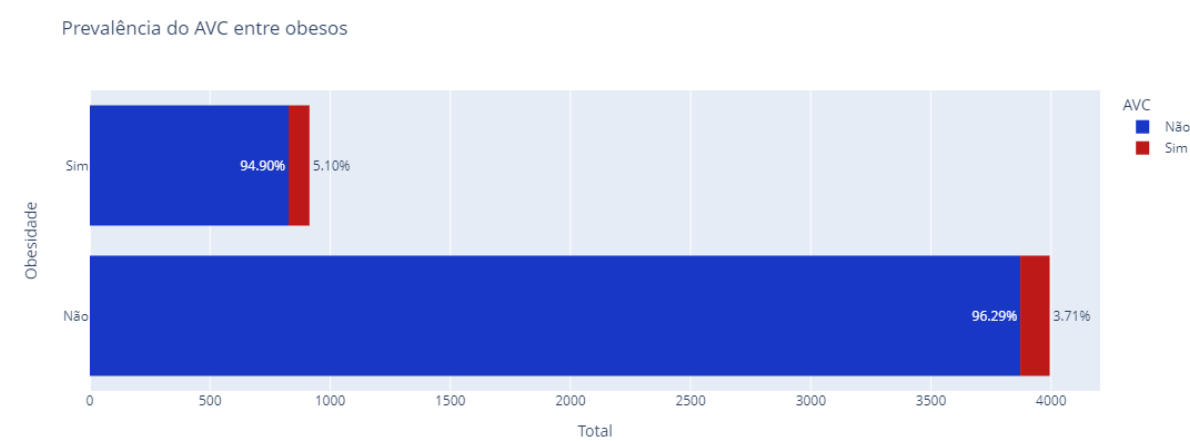
IMC e Obesidade: Foi analisada a prevalência do AVC estratificada por IMC (figura 20), esse gráfico possui eixo das ordenadas em escala logarítmica para facilitar visualização. Em seguida (figura 21), a população foi dividida entre obesos (IMC>30) e não-obesos (IMC<30), para avaliar a relevância desse fator de risco.

Figura 20 - AVC por IMC



Fonte: Autor

Figura 21 - AVC entre obesos e não-obesos

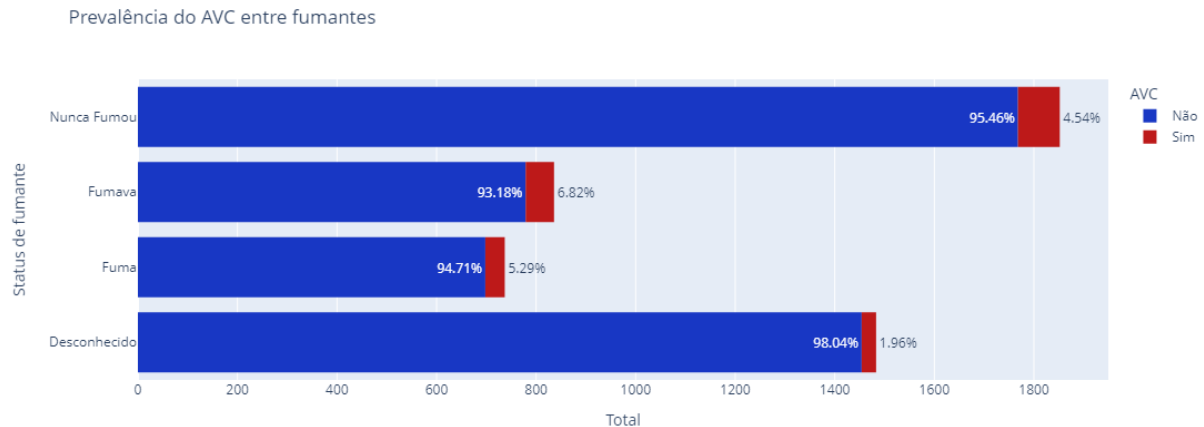


Fonte: Autor

Embora tenha demonstrado prevalência de AVC levemente maior entre obesos, essa diferença não se mostrou tão grande quanto nos demais fatores de risco, mas é relevante suficiente para se manter na análise.

Condição de fumante: Sobre esse atributo buscou-se tanto procurar a diferença entre eles (figura 22) e reduzir as opções de resposta para um campo de apenas duas opções (figura 24).

Figura 22 - Prevalência de AVC estratificado por uso de tabaco



Fonte: Autor

Os grupos de pessoas que fumam ou que já tiveram o hábito de fumar se mostraram fatores de risco para a presença de AVC. Além disso, é necessário tratar os valores “desconhecido” sobre os status de fumante, para isso, primeiramente deve-se pensar no motivo que existe essa opção para esse atributo. Por ser algo extremamente simples de ser perguntado e registrado, sem necessitar de nenhum tipo de exame complementar ou memória que o paciente não tenha na hora, é plausível concluir que se o valor resultante é “Desconhecido”, ou a pergunta não foi feita, ou o paciente não quis responder tal pergunta. Conhecendo mais profundamente o conjunto de dados, deve-se lembrar que existem pessoas de todas as idades, incluindo recém nascidos e crianças, dessa forma é possível imaginar que tal pergunta não foi feita para esse grupo de pessoas, hipótese essa comprovada pelo resultado da Figura 23.

Figura 23 - Relação fumante "desconhecido" com menores de idade

```
[106] df_copy['smoking_status'].value_counts()
Python
...
Nunca Fumou      1852
Desconhecido      1483
Fumava            836
Fuma              737
Name: smoking_status, dtype: int64

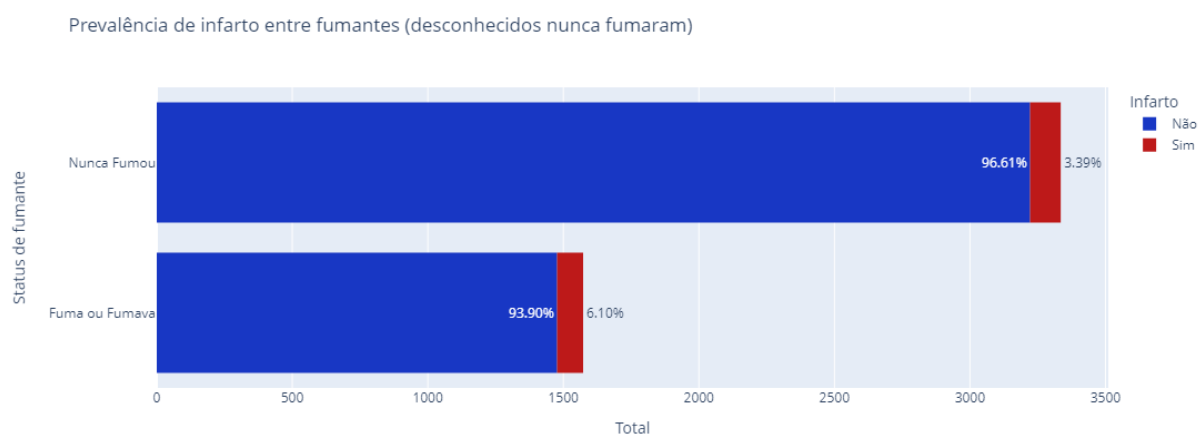
[112] below18 = df_copy[df_copy['age']<19]
Python

[113] below18['smoking_status'].value_counts()
Python
...
Desconhecido      693
Nunca Fumou       163
Fumava            26
Fuma              13
Name: smoking_status, dtype: int64
```

Fonte: Autor

Pode-se perceber que entre o grupo de pessoas menores de idade, a prevalência da resposta “desconhecido” subiu de cerca de 30% para 75%. Além disso, as pessoas com status desconhecido sobre fumantes aparentaram o menor grau de risco relacionado ao AVC, portanto é possível afirmar que esse grupo de pessoas provavelmente nunca fumou, habilitando reduzir o número de respostas possíveis para essa variável: “Nunca fumou” que engloba os originais “nunca fumou” + “desconhecido” e “Fuma ou fumava” que engloba os originais “fuma” + “fumava”.

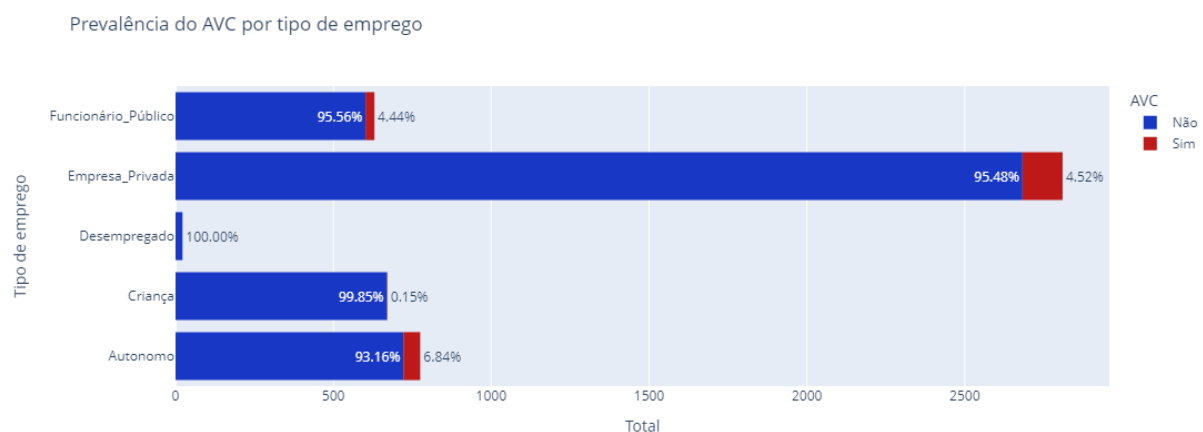
Figura 24 - Prevalência de AVC entre os dois grupos resultantes sobre tabaco



Fonte: Autor

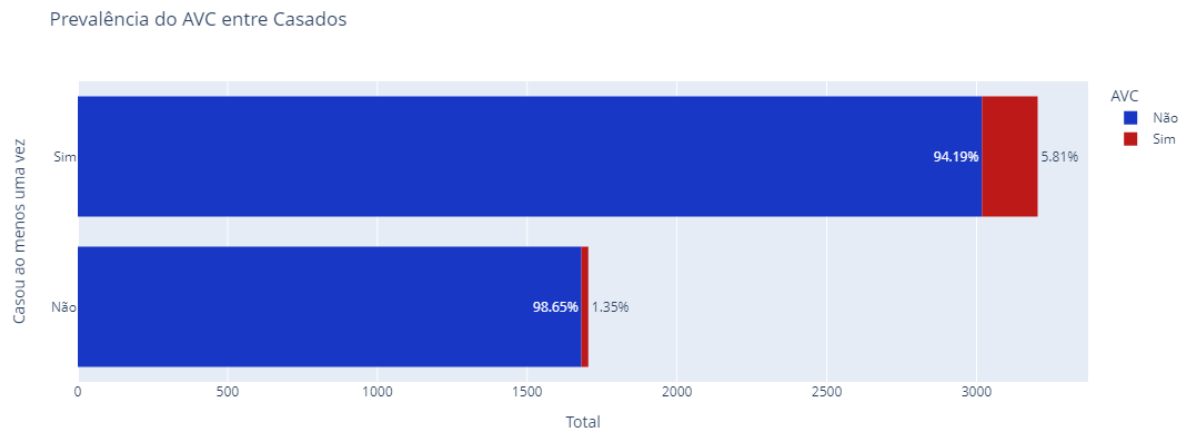
Dessa forma, esse atributo se tornou mais indicativo de fator de risco do que a forma em que ele se apresentava anteriormente, indicando quase o dobro de risco entre aqueles que tiveram o hábito de fumar em algum momento da vida.

Tipo de emprego: Buscou-se entender e interpretar as tendências entre tipo de emprego e AVC.



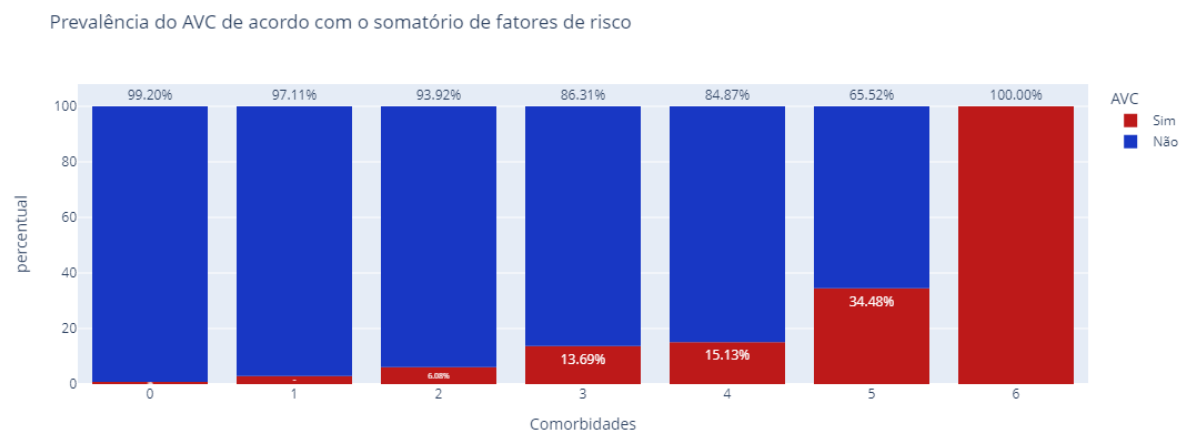
Pode-se perceber que existe pouca diferença entre o grupo de pessoas que trabalha para empresas privadas e o grupo de funcionários públicos, o grupo de “desempregados” tem poucas amostras, o grupo de “crianças” é mais correlato com idade do que com tipo de trabalho em si e existe alguma tendência maior dos ‘Autônomos’ sofrerem com AVC, talvez relacionado ao estresse extra que essa categoria sofre diariamente. A partir disso, poderiam ter sido tomadas duas decisões distintas: a primeira seria agrupar os dados em dois grandes grupos, “Autônomos” e “Não-Autônomos” e a segunda, utilizada nesse trabalho, que foi não utilizar essa coluna para realizar previsões, uma vez que vale lembrar que a categoria “Autônomo” pode também significar “Empreendedor” e pessoas com empregos fixos também podem ser empreendedores, dessa forma tais pessoas participariam de ambos os grupos, trazendo inconsistências aos dados.

Estado civil: Nessa parte da análise, são comparados dados de pessoas que são ou já foram casadas em algum momento e pessoas que nunca se casaram.

Figura 25 - Prevalência entre AVC e estado civil**Fonte: Autor**

De primeira vista, pode parecer que ser solteiro para sempre é a cura para o AVC, entretanto, deve-se lembrar que esse conjunto de dados possui muitos jovens, que são pessoas que não tiveram nem tempo de se casar, nem tempo de sofrer com essa enfermidade, portanto a baixa prevalência de AVC entre os não-casados é altamente influenciada por esse fator que já está escondido nas outras variáveis. Por isso, esse atributo não será utilizado na análise.

Comorbidades: Por fim a análise da coluna criada que realiza o somatório dos fatores de risco.

Figura 26 - Prevalência de AVC por somatório de fatores de risco**Fonte: Autor**

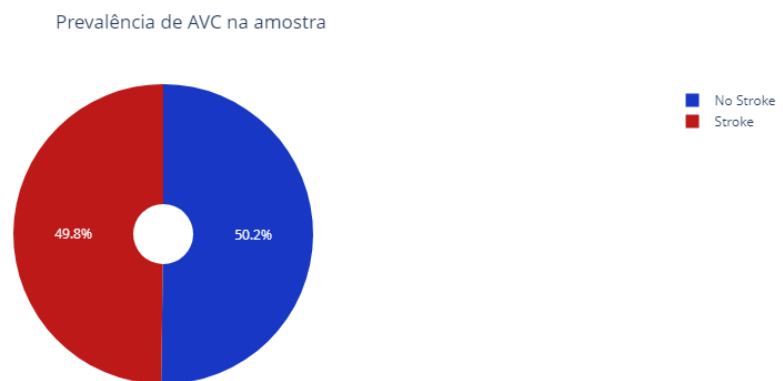
Esse atributo demonstrou exatamente o que se esperava dele, seguindo as orientações da American Heart Association sobre fatores de risco. Nesse gráfico percebe-se que quanto mais fatores de risco, maior a chance de a pessoa sofrer um AVC, e essa tendência se mostra de maneira exponencial, mas seriam necessários mais valores para confirmar tal hipótese e gerar uma curva com bom grau de confiabilidade. De momento, pode-se afirmar que é um atributo chave para a predição da variável alvo e levanta a hipótese de que é possível que o núcleo da prevenção dessa enfermidade esteja em impedir o surgimento de novos fatores de risco entre as pessoas que já tem alguns desses fatores, impedindo o acúmulo de comorbidades e evitando a parte mais vertical da curva exponencial.

4.2 Dados Team Inciribo

Nessa seção serão analisados em sequência cada atributo e as conclusões de cada análise do conjunto de dados sintéticos criados pela “Team Inciribo”, atributos já eliminados anteriormente não serão novamente analisados.

AVC (variável alvo): Fara fins de contexto geral dos outros atributos, primeiramente foi analisada a variável alvo para identificar a prevalência de AVC ou não-AVC na amostra (Figura 27).

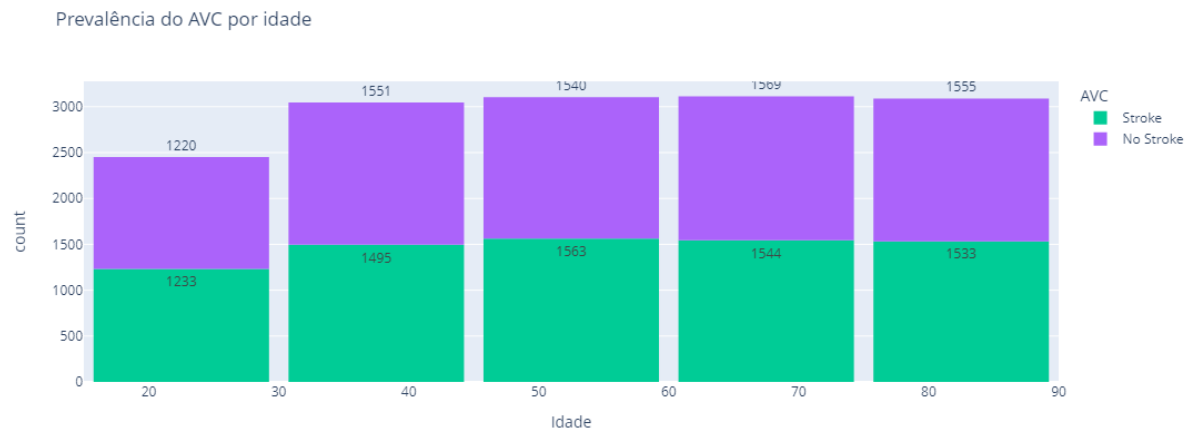
Figura 27 - Prevalência de AVC na amostra



Fonte: Autor

Idade: aqui buscou-se identificar a prevalência da variável alvo estratificada por idades, por isso, lançou-se mão de um histograma (Figura 28).

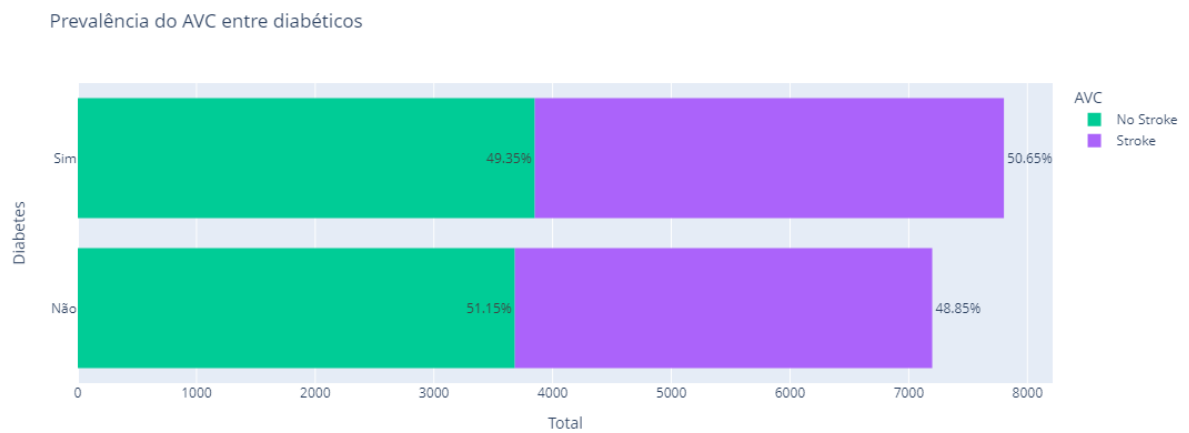
Figura 28 - Histograma idade x AVC



Fonte: Autor

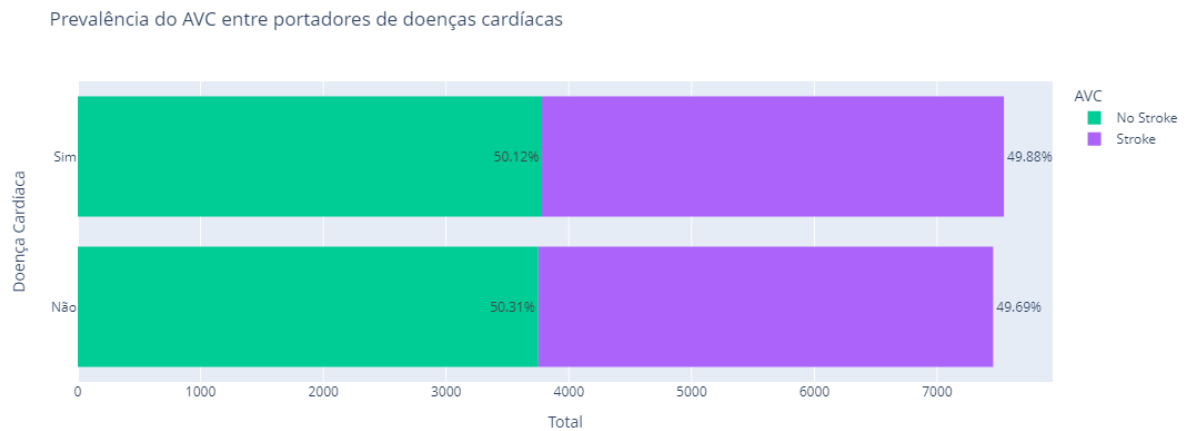
Diabetes: Buscou-se comparar a prevalência do AVC entre os diabéticos e não diabéticos (Figura 29).

Figura 29 - Relação Diabetes e AVC

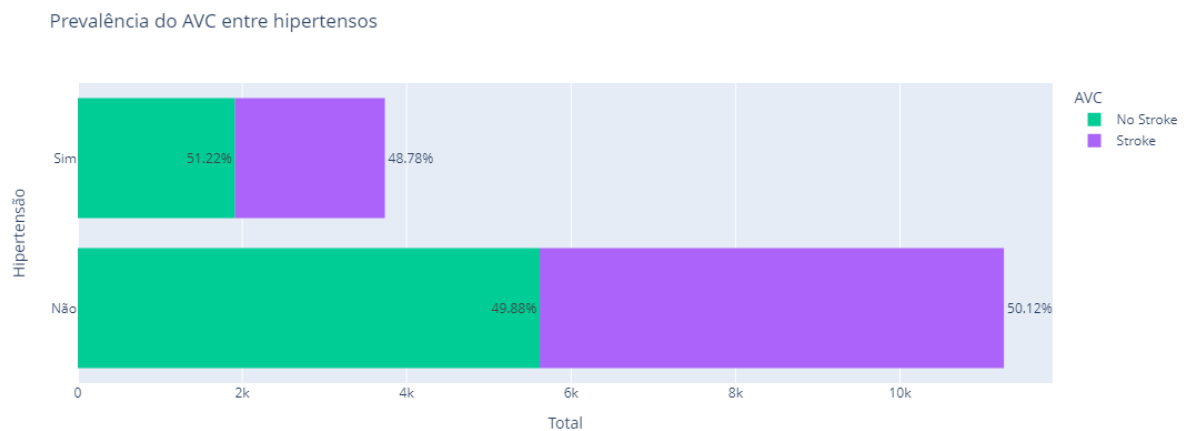


Fonte: Autor

Doenças cardíacas: buscou-se perceber as diferenças entre os grupos de pessoas que possuem e que não possuem doenças cardíacas (Figura 30).

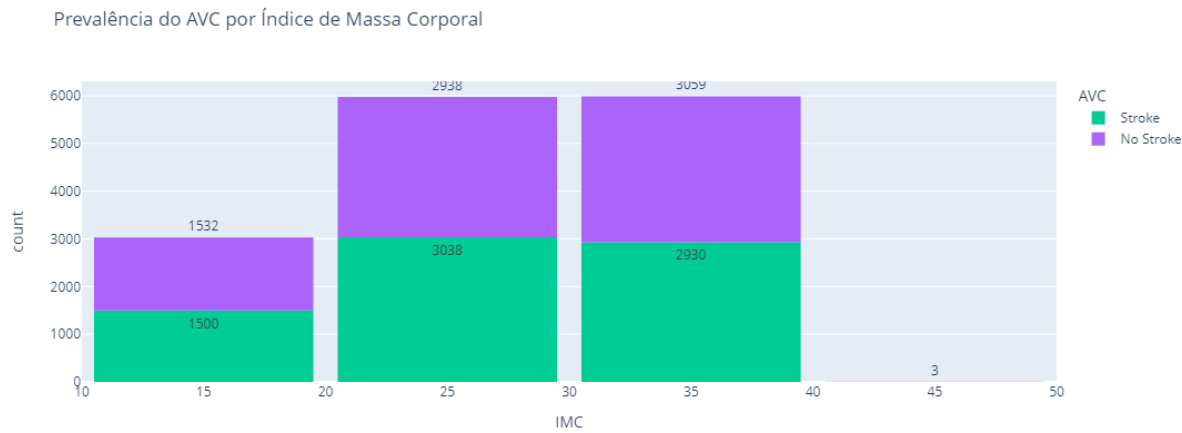
Figura 30 - Relação AVC e Doença Cardíaca**Fonte: Autor**

Hipertensão: Análise da prevalência do AVC entre pacientes hipertensos e não-hipertensos (Figura 31).

Figura 31 - Relação AVC e Hipertensão**Fonte: Autor**

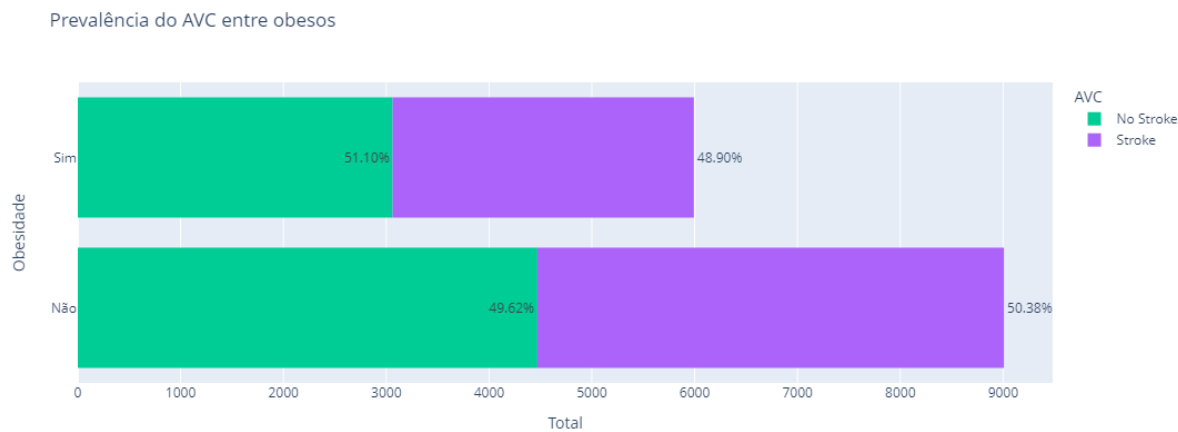
IMC e Obesidade: Foi analisada a prevalência do AVC estratificada por IMC (figura 32). Em seguida (figura 33), a população foi dividida entre obesos (IMC>30) e não-obesos (IMC<30), para avaliar a relevância desse fator de risco.

Figura 32 - Histograma IMC x AVC



Fonte: Autor

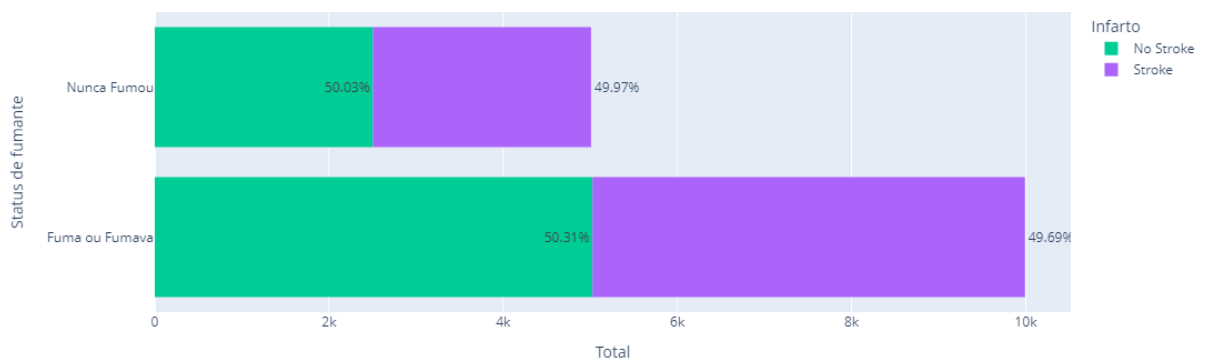
Figura 33 - Relação Obesidade x AVC



Fonte: Autor

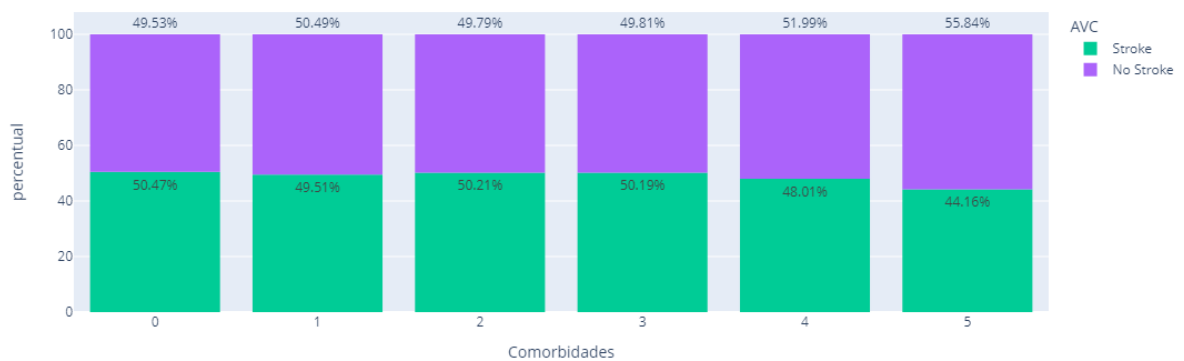
Condição de fumante: Sobre esse atributo buscou-se realizar a mesma análise feita no conjunto de dados anterior, resumindo em dois grandes grupos (figura 24).

Prevalência de infarto entre fumantes (desconhecidos nunca fumaram)



Comorbidades: Por fim a análise da coluna criada que realiza o somatório dos fatores de risco.

Prevalência do AVC de acordo com o somatório de comorbidades



A conclusão sobre os dados criados artificialmente é que foram criados dados balanceados sobre tudo, mas que parecem absolutamente aleatórios, não foi possível observar nenhuma diferença significativa entre a prevalência de AVC nos atributos chave do problema. Além disso, o resultado final do atributo “Comorbidades” criado anteriormente demonstrou resultado inverso ao esperado, esse conjunto de achados irreais na análise exploratória dos dados torna esse *dataset* extremamente divergente do dataset original, que se aproxima muito mais do mundo real, dos pacientes e das instruções retiradas dos órgãos oficiais.

5. Criação de Modelos de Machine Learning

Originalmente existem dois conjuntos de dados, um deles retirados de pacientes reais, dos Estados Unidos da América e o outro é um conjunto de dados artificiais que se demonstraram divergentes da realidade na análise exploratória de dados. Poderiam ser utilizadas as seguintes abordagens:

- Unificar os conjuntos de dados em um grande conjunto e depois particionar esse grande conjunto em grupos de treino e teste;
- Utilizar os conjuntos de dados separadamente, treinar e testar com o grupo de dados artificiais e validar com o grupo de dados reais.
- Utilizar os conjuntos de dados separadamente, treinar e testar com o grupo de dados reais e validar com o grupo de dados artificiais.

Entretanto, pela gritante diferença entre os dois conjuntos de dados, é praticamente impossível que o mesmo modelo funcione para os dois conjuntos. Dessa forma, unificar os dados e gerar um algoritmo capaz de lidar com esse grande conjunto pode parecer, a primeiro instante, a melhor abordagem. No entanto, deve-se lembrar que o objetivo do problema não é simplesmente prever com alguma precisão o maior conjunto de dados possível assumindo que o modelo será mais robusto por possuir mais dados. O objetivo, voltando ao Tópico 1.3, é criar um algoritmo capaz de auxiliar na resolução de problemas do mundo real, e os dados que se assemelham ao mundo real nesse caso são os dados retirados dos pacientes reais e, que na análise exploratória de dados, convergiram a resultados plausíveis que fazem sentido ao se tratar da espécie humana.

Dessa forma a abordagem escolhida foi treinar, testar e realizar validação cruzada com o conjunto de dados reais, e eventualmente o modelo foi utilizado também para tentar prever o grupo de dados artificiais.

A primeira etapa para a construção do modelo é selecionar e preparar as colunas para que o modelo seja treinável, para isso, variáveis categóricas com 2 opções serão convertidas para valores de 1 ou 0, e caso existam variáveis categóricas com mais de duas opções, deverão ser criadas colunas extras para cada uma dessas opções.

Dessa forma, os atributos finais escolhidos e suas opções foram:

- **Idade (*age*):** Numérico;

- **Hipertensão (*hypertension*)**: Categórico booleano, convertido para 1 ou 0;
- **Doenças Cardíacas (*heart_disease*)**: Categórico booleano, convertido para 1 ou 0;
- **IMC (*bmi*)**: Numérico;
- **Obesidade**: Categórico booleano, convertido para 1 ou 0;
- **Diabetes**: Categórico booleano, convertido para 1 ou 0;
- **Média Glicêmica (*avg_glucose_level*)**: Numérico;
- **Condição de fumante (*smoking_status*)**: Categórico com duas opções, “fuma ou fumava” convertido para 1, e “nunca fumou” convertido para 0;
- **Comorbidades**: Numérico;

Além disso, para o treinamento de um modelo de classificação é interessante que a variável alvo esteja representada de maneira similares para o desfecho positivo ou negativo, por isso foi utilizado o recurso de reamostragem (*upsampling*) para aumentar o número de dados em que ocorre AVC. Foram realizados os procedimentos demonstrados a seguir (Figura 34). Primeiro foram separados os dados AVC positivo e AVC falso em dois conjuntos de dados diferentes, depois o conjunto de AVC positivo foi acrescido de novos dados até ficar do mesmo tamanho que o outro conjunto, em seguida ambos foram concatenados e as colunas renomeadas novamente.

Figura 34 - Reamostragem (*upsampling*)

Reamostragem

```
data_0 = df_copy[df_copy['stroke']==0] #dados onde stroke = 0
data_1 = df_copy[df_copy['stroke']==1] #dados onde stroke = 1

df['stroke'].value_counts()
```

[12] ✓ 0.0s Python

```
0    4677
1     209
Name: stroke, dtype: int64
```

▷

```
data_1 = resample(data_1,replace=True , n_samples=data_0.shape[0] , random_state=123 )
data_1
```

[15] ✓ 0.0s Python

	id	age	hypertension	heart_disease	avg_glucose_level	bmi	smoking_status	stroke	Comorbidades	diabetes	obesidade
109	16817	78.0	1	0	130.54	20.1	0	1	3	1	0
126	46703	68.0	0	1	223.83	31.9	1	1	4	1	1
66	26727	79.0	0	0	88.92	22.9	0	1	1	0	0
98	7547	74.0	0	0	72.96	31.3	1	1	3	0	1
17	13861	52.0	1	0	233.29	48.9	0	1	3	1	1
...
93	48405	80.0	0	1	68.53	24.2	1	1	3	0	0
123	62861	78.0	0	0	67.29	24.6	0	1	1	0	0
125	58978	70.0	0	1	239.07	26.1	0	1	3	1	0
89	12062	54.0	0	0	191.82	40.4	1	1	3	1	1
139	44993	79.0	1	0	98.02	22.3	1	1	2	0	0

4677 rows × 11 columns

```
data_1.columns
```

[16] ✓ 0.0s Python

```
Index(['id', 'age', 'hypertension', 'heart_disease', 'avg_glucose_level',
      'bmi', 'smoking_status', 'stroke', 'Comorbidades', 'diabetes',
      'obesidade'],
      dtype='object')
```

▷

```
df_resampled = np.concatenate((data_0,data_1))
df_resampled = pd.DataFrame(df_resampled)
df_resampled
```

[17] ✓ 0.0s Python

	0	1	2	3	4	5	6	7	8	9	10
0	30669.0	3.0	0.0	0.0	95.12	18.0	0.0	0.0	0.0	0.0	0.0
1	30468.0	58.0	1.0	0.0	87.96	39.2	0.0	0.0	3.0	0.0	1.0
2	16523.0	8.0	0.0	0.0	110.89	17.6	0.0	0.0	0.0	0.0	0.0
3	56543.0	70.0	0.0	0.0	69.04	35.9	1.0	0.0	2.0	0.0	1.0
4	32257.0	47.0	0.0	0.0	210.95	50.1	0.0	0.0	2.0	1.0	1.0
...
9349	48405.0	80.0	0.0	1.0	68.53	24.2	1.0	1.0	3.0	0.0	0.0
9350	62861.0	78.0	0.0	0.0	67.29	24.6	0.0	1.0	1.0	0.0	0.0
9351	58978.0	70.0	0.0	1.0	239.07	26.1	0.0	1.0	3.0	1.0	0.0
9352	12062.0	54.0	0.0	0.0	191.82	40.4	1.0	1.0	3.0	1.0	1.0
9353	44993.0	79.0	1.0	0.0	98.02	22.3	1.0	1.0	2.0	0.0	0.0

9354 rows × 11 columns

```
df_resampled.columns = ['id', 'age', 'hypertension', 'heart_disease', 'avg_glucose_level',
                        'bmi', 'smoking_status', 'stroke', 'comorbidades', 'diabetes',
                        'obesidade']
```

[19] ✓ 0.0s Python

Fonte: Autor

Dessa forma, o novo conjunto de dados se apresenta da seguinte maneira (Figura 35):

Figura 35 - Conjunto de dados a ser treinado

df_resampled

	id	age	hypertension	heart_disease	avg_glucose_level	bmi	smoking_status	stroke	Comorbidades	diabetes	obesidade
0	30669.0	3.0	0.0	0.0	95.12	18.0	0.0	0.0	0.0	0.0	0.0
1	30468.0	58.0	1.0	0.0	87.96	39.2	0.0	0.0	3.0	0.0	1.0
2	16523.0	8.0	0.0	0.0	110.89	17.6	0.0	0.0	0.0	0.0	0.0
3	56543.0	70.0	0.0	0.0	69.04	35.9	1.0	0.0	2.0	0.0	1.0
4	32257.0	47.0	0.0	0.0	210.95	50.1	0.0	0.0	2.0	1.0	1.0
...
9349	48405.0	80.0	0.0	1.0	68.53	24.2	1.0	1.0	3.0	0.0	0.0
9350	62861.0	78.0	0.0	0.0	67.29	24.6	0.0	1.0	1.0	0.0	0.0
9351	58978.0	70.0	0.0	1.0	239.07	26.1	0.0	1.0	3.0	1.0	0.0
9352	12062.0	54.0	0.0	0.0	191.82	40.4	1.0	1.0	3.0	1.0	1.0
9353	44993.0	79.0	1.0	0.0	98.02	22.3	1.0	1.0	2.0	0.0	0.0

9354 rows x 11 columns

df_resampled['stroke'].value_counts()

```
0.0    4677
1.0    4677
Name: stroke, dtype: int64
```

Fonte: Autor

Em seguida, foram divididos os grupos de treino e teste (Figura 36) e verificado o balanceamento da variável alvo em ambos casos (Figura 37).

Figura 36 - Divisão em grupos de treino e teste

Grupos de Treino e Teste

```
x = df_resampled.drop(['stroke', 'id'], axis = 1)
y = df_resampled['stroke']
```

x.columns

```
Index(['age', 'hypertension', 'heart_disease', 'avg_glucose_level', 'bmi',
       'smoking_status', 'Comorbidades', 'diabetes', 'obesidade'],
      dtype='object')
```

Colunas=

```
['age', 'hypertension', 'heart_disease', 'avg_glucose_level', 'bmi',
 'smoking_status', 'Comorbidades', 'diabetes', 'obesidade']
```

```
scaler = StandardScaler()

x_scaled = scaler.fit_transform(x)
```

```
x_train, x_test, y_train, y_test= train_test_split(X_scaled,y,test_size = .20)
```

Fonte: Autor

Figura 37 - Verificação do balanceamento nos grupos de treino e teste

```
[27] x_train, x_test, y_train, y_test= train_test_split(X_scaled,y,test_size = .20)
✓ 0.0s Python
```

```
[28] y_train.value_counts()
✓ 0.0s Python
... 0.0 3765
     1.0 3718
     Name: stroke, dtype: int64
```

```
[29] y_test.value_counts()
✓ 0.0s Python
... 1.0 959
     0.0 912
     Name: stroke, dtype: int64
```

Os valores não são exatamente iguais para infarto verdadeiro ou falso nos grupos de treino e teste entretanto, os dados ainda estão balanceados em relação à variável alvo.

Fonte: Autor

Em seguida foram testados diferentes modelos de classificação, otimizados pelo modelo de busca *grid_search*, que realiza o treinamento de diferentes modelos com diferentes hiperparâmetros buscando encontrar aqueles de melhor performance nos grupos de teste. Seguem os resultados de cada modelo com os parâmetros selecionados para o *grid_search* e os resultados encontrados:

Regressão Logística

Figura 38 - Implementação e resultados da Regressão logística

```

param_grid = {
    'C': [0.001, 0.01, 0.1, 1, 10, 100],
    'penalty': ['l1', 'l2'],
    'solver': ['liblinear', 'saga']
}

# Create a Logistic Regression classifier
logistic_classifier = LogisticRegression()

# Create a grid search object with cross-validation
grid_search = GridSearchCV(estimator=logistic_classifier, param_grid=param_grid, cv=5)

# Fit the grid search to the training data
grid_search.fit(x_train, y_train)

# Get the best parameters and the best estimator
best_params_lg = grid_search.best_params_
best_logistic_classifier = grid_search.best_estimator_

# Evaluate the model on the test set
y_pred_lg = best_logistic_classifier.predict(x_test)

# Calculate accuracy and print the classification report
accuracy = accuracy_score(y_test, y_pred_lg)
print("Melhores Parâmetros:", best_params_lg)
print("Precisão no grupo de teste:", accuracy)
print("\nRelatório de Classificação:\n", classification_report(y_test, y_pred_lg))

#Crossvalidation
scores_lr = cross_val_score(best_logistic_classifier, X=x_train, y=y_train, cv=5, n_jobs=1)
print('Precisão de cada validação cruzada: %s' % scores_lr)
print('Precisão Média validação cruzada: %.3f +/- %.3f' % (np.mean(scores_lr), np.std(scores_lr)))

#ROC curve
roc_auc = roc_auc_score(y_test, best_logistic_classifier.predict_proba(x_test)[:, 1])
fpr, tpr, _ = roc_curve(y_test, best_logistic_classifier.predict_proba(x_test)[:, 1])

plt.figure(figsize=(8, 6))
plt.plot(fpr, tpr, color='darkorange', lw=2, label='Curva ROC (area = {:.2f})'.format(roc_auc))
plt.plot([0, 1], [0, 1], color='navy', lw=2, linestyle='--')
plt.xlim([0.0, 1.0])
plt.ylim([0.0, 1.05])
plt.xlabel('Taxa de Falsos Positivos')
plt.ylabel('Taxa de Verdadeiros Positivos')
plt.title('Curva ROC - Logistic Regression')
plt.legend(loc='lower right')
plt.show()

# Confusion Matrix
conf_matrix_lg = confusion_matrix(y_test, y_pred_lg)
plt.figure(figsize=(8, 6))
sns.heatmap(conf_matrix_lg, annot=True, fmt="d", cmap='Blues', linewidths=2.5, cbar=False)
plt.xlabel('Previsão')
plt.ylabel('Real')
plt.title('Matriz de Confusão - Logistic Regression')
plt.xticks([0.5, 1.5], ['Previsto não-AVC', 'Previsto AVC'])
plt.yticks([0.5, 1.5], ['Não-AVC real', 'AVC real'])
plt.show()

# Plot the feature importances
fig, ax = plt.subplots()
viz = FeatureImportances(best_logistic_classifier)
viz.fit(x_train, y_train)
indices = viz.features_
y_ordered = list()
for i in indices:

```



```

for i in indices:
    y_ordered.append(colunas[i])

ax.set_yticklabels(y_ordered)

plt.tick_params(labelsize=13)
plt.title('Feature Importance - Logistic Regression')

plt.show()

```

✓ 3.2s Python

Melhores Parâmetros: {'C': 0.001, 'penalty': 'l2', 'solver': 'saga'}

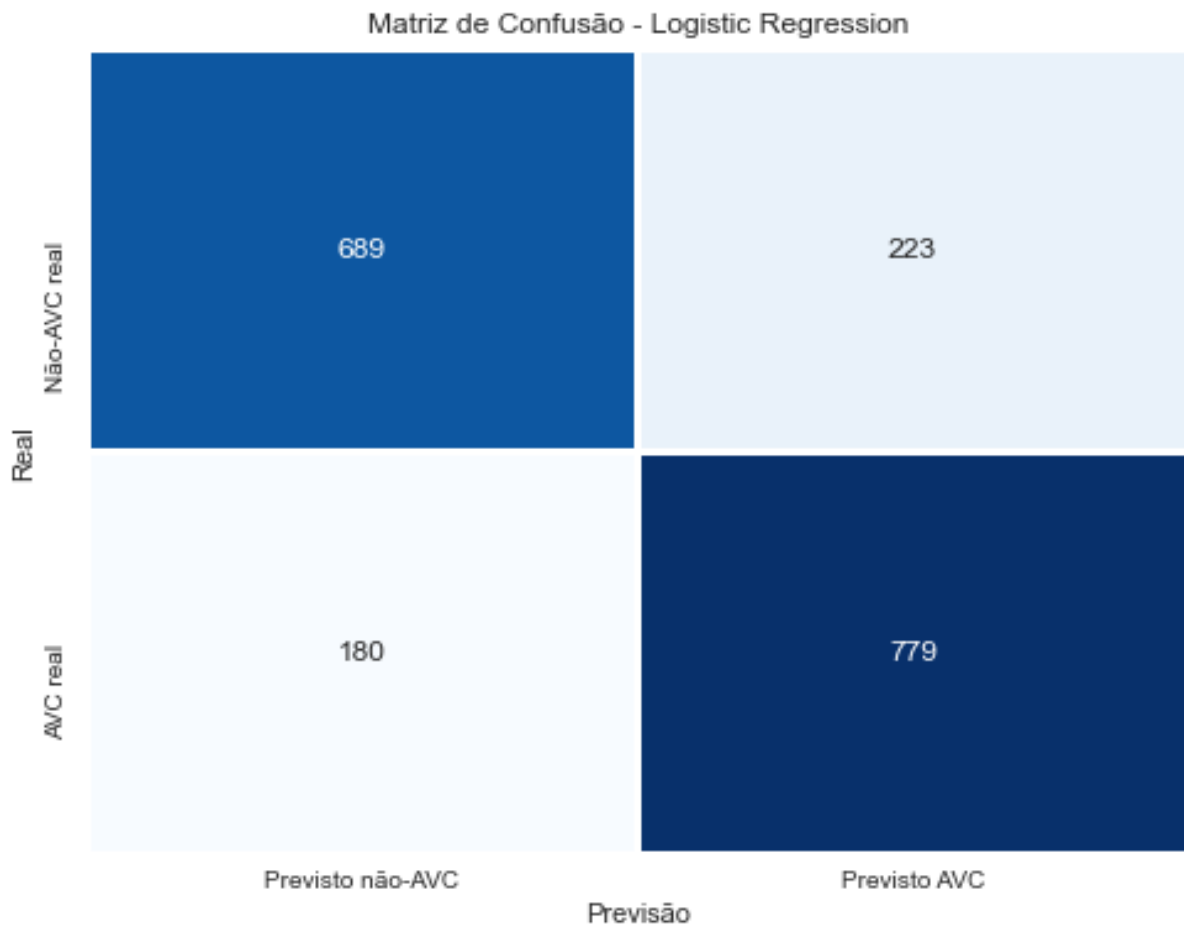
Precisão no grupo de teste: 0.7846071619454837

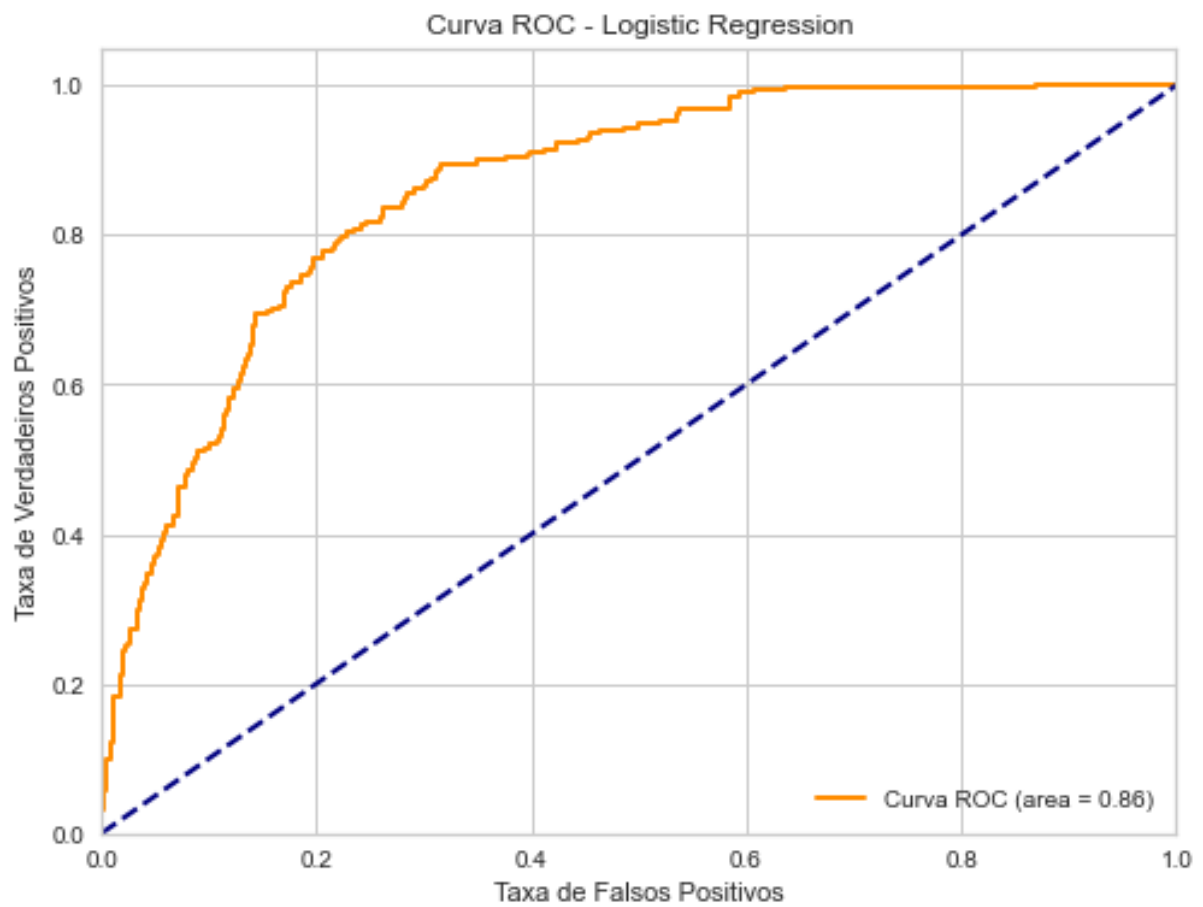
Relatório de Classificação:

	precision	recall	f1-score	support
0.0	0.79	0.76	0.77	912
1.0	0.78	0.81	0.79	959
accuracy			0.78	1871
macro avg	0.79	0.78	0.78	1871
weighted avg	0.78	0.78	0.78	1871

Precisão de cada validação cruzada: [0.77755511 0.76419506 0.78423514 0.7473262 0.77874332]

Precisão Média validação cruzada: 0.770 +/- 0.013





Fonte: Autor

A regressão logística apresentou como melhor resultado precisão média de validação cruzada em 5 partes = 0,77 e área da curva ROC = 0,86.

Decision Tree

Figura 39 - Implementação e resultados do Decision Tree

Decision Tree

```

param_grid = {
    'criterion': ['gini', 'entropy'],
    'max_depth': [None, 10, 20, 30, 40, 50],
    'min_samples_split': [2, 5, 10],
    'min_samples_leaf': [1, 2, 4]
}

# Create a Decision Tree classifier
dt_classifier = DecisionTreeClassifier()

# Create a grid search object with cross-validation
grid_search = GridSearchCV(estimator=dt_classifier, param_grid=param_grid, cv=5)

# Fit the grid search to the training data
grid_search.fit(x_train, y_train)

# Get the best parameters and the best estimator
best_params_dt = grid_search.best_params_
best_dt_classifier = grid_search.best_estimator_

# Evaluate the model on the test set
y_pred_dt = best_dt_classifier.predict(x_test)

# Calculate accuracy and print the classification report
accuracy = accuracy_score(y_test, y_pred_dt)
print("Melhores Parâmetros:", best_params_dt)
print("Precisão no grupo de teste:", accuracy)
print("\nRelatório de Classificação:\n", classification_report(y_test, y_pred_dt))

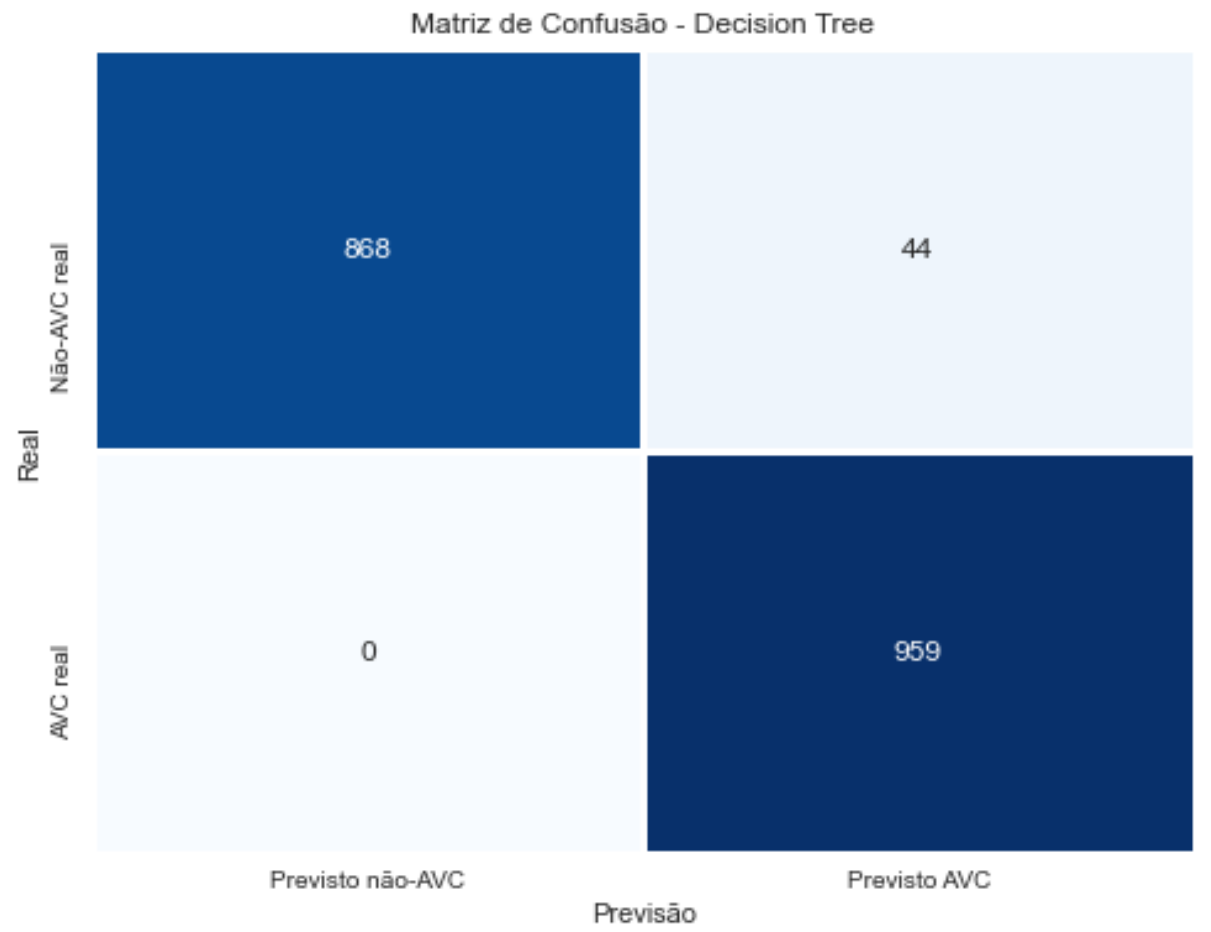
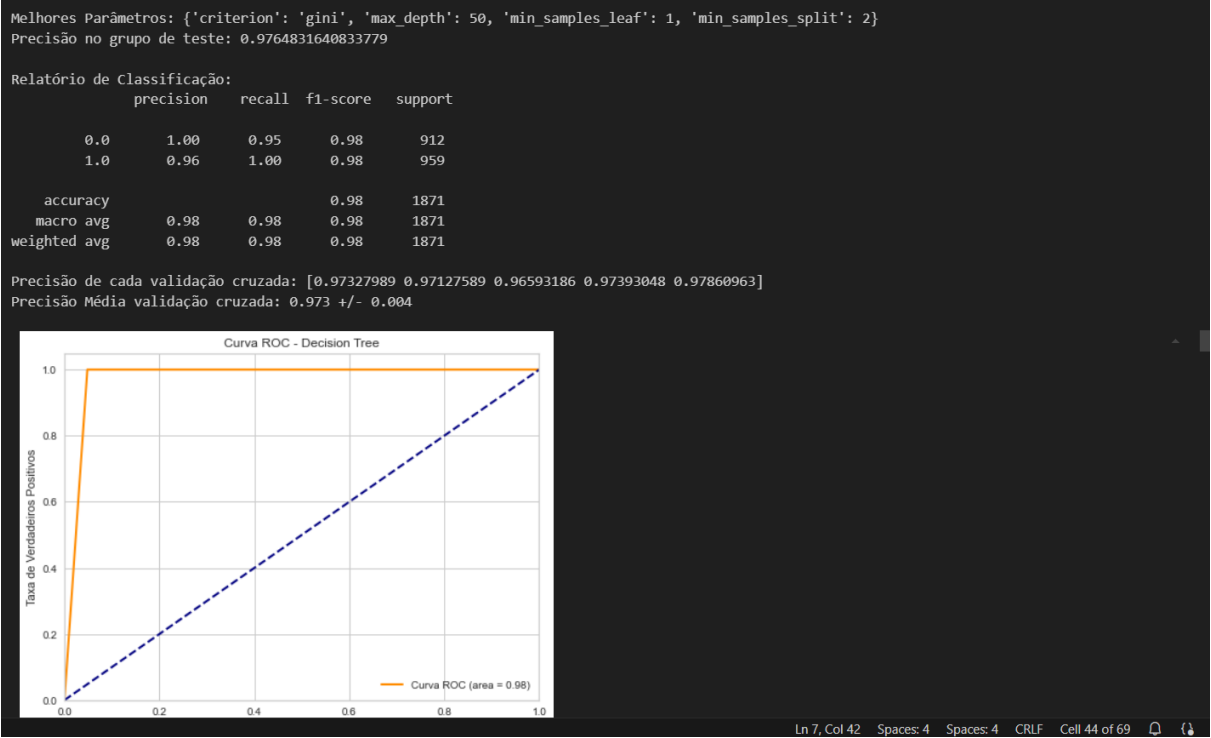
#Crossvalidation
scores_dt = cross_val_score(best_dt_classifier, x=x_train, y=y_train, cv=5, n_jobs=1)
print("Precisão de cada validação cruzada: %s" % scores_dt)
print("Precisão Média validação cruzada: %.3f +/- %.3f" % (np.mean(scores_dt), np.std(scores_dt)))

#ROC curve
roc_auc = roc_auc_score(y_test, best_dt_classifier.predict_proba(x_test)[:, 1])
fpr, tpr, _ = roc_curve(y_test, best_dt_classifier.predict_proba(x_test)[:, 1])

plt.figure(figsize=(8, 6))
plt.plot(fpr, tpr, color='darkorange', lw=2, label='Curva ROC (area = {:.2f})'.format(roc_auc))
plt.plot([0, 1], [0, 1], color='navy', lw=2, linestyle='--')
plt.xlim([0.0, 1.0])
plt.ylim([0.0, 1.05])
plt.xlabel('Taxa de Falsos Positivos')
plt.ylabel('Taxa de Verdadeiros Positivos')
plt.title('Curva ROC - Decision Tree')
plt.legend(loc='lower right')
plt.show()

# Confusion Matrix
conf_matrix_dt = confusion_matrix(y_test, y_pred_dt)
plt.figure(figsize=(8, 6))
sns.heatmap(conf_matrix_dt, annot=True, fmt="d", cmap='Blues', linewidths=2.5, cbar=False)
plt.xlabel('Previsão')
plt.ylabel('Real')
plt.title('Matriz de Confusão - Decision Tree')
plt.xticks([0.5, 1.5], ['Previsto não-AVC', 'Previsto AVC'])
plt.yticks([0.5, 1.5], ['Não-AVC real', 'AVC real'])
plt.show()

```



Fonte: Autor

O modelo de Decision tree apresentou como melhor resultado precisão média de validação cruzada em 5 partes = 0,973 e área da curva ROC = 0,98.

K-Nearest-Neighbors

Figura 40 - Implementação e Resultados do modelo KNN

```
param_grid = {
    'n_neighbors': [2, 3, 5, 7, 9],
    'weights': ['uniform', 'distance'],
    'metric': ['euclidean', 'manhattan', 'chebyshev']
}

# Create a K-Nearest Neighbors classifier
knn_classifier = KNN()

# Create a grid search object with cross-validation
grid_search = GridSearchCV(estimator=knn_classifier, param_grid=param_grid, cv=5)

# Fit the grid search to the training data
grid_search.fit(x_train, y_train)

# Get the best parameters and the best estimator
best_params_knn = grid_search.best_params_
best_knn_classifier = grid_search.best_estimator_

# Evaluate the model on the test set
y_pred_knn = best_knn_classifier.predict(x_test)

# Calculate accuracy and print the classification report
accuracy = accuracy_score(y_test, y_pred_knn)
print("Melhores Parâmetros:", best_params_knn)
print("Precisão no grupo de teste:", accuracy)
print("\nRelatório de Classificação:\n", classification_report(y_test, y_pred_knn))

#Crossvalidation
scores_knn = cross_val_score(best_knn_classifier, X=x_train, y=y_train, cv=5, n_jobs=1)
print('Precisão de cada validação cruzada: %s' % scores_knn)
print('Precisão Média validação cruzada: %.3f +/- %.3f' % (np.mean(scores_knn), np.std(scores_knn)))

#ROC curve
roc_auc = roc_auc_score(y_test, best_knn_classifier.predict_proba(x_test)[: , 1])
fpr, tpr, _ = roc_curve(y_test, best_knn_classifier.predict_proba(x_test)[: , 1])

plt.figure(figsize=(8, 6))
plt.plot(fpr, tpr, color='darkorange', lw=2, label='Curva ROC (area = {:.2f})'.format(roc_auc))
plt.plot([0, 1], [0, 1], color='navy', lw=2, linestyle='--')
plt.xlim([0.0, 1.0])
plt.ylim([0.0, 1.05])
plt.xlabel('Taxa de Falsos Positivos')
plt.ylabel('Taxa de Verdadeiros Positivos')
plt.title('Curva ROC - KNN')
plt.legend(loc='lower right')
plt.show()

# Confusion Matrix
conf_matrix_knn = confusion_matrix(y_test, y_pred_knn)
plt.figure(figsize=(8, 6))
sns.heatmap(conf_matrix_knn, annot=True, fmt="d", cmap='Blues', linewidths=2.5, cbar=False)
plt.xlabel('Previsão')
plt.ylabel('Real')
plt.title('Matriz de Confusão - KNN')
plt.xticks([0.5, 1.5], ['Previsto não-AVC', 'Previsto AVC'])
plt.yticks([0.5, 1.5], ['Não-AVC real', 'AVC real'])
plt.show()

# Não existe Feature Importance de KNN
```

```

Melhores Parâmetros: {'metric': 'euclidean', 'n_neighbors': 2, 'weights': 'uniform'}
Precisão no grupo de teste: 0.9823623730625334

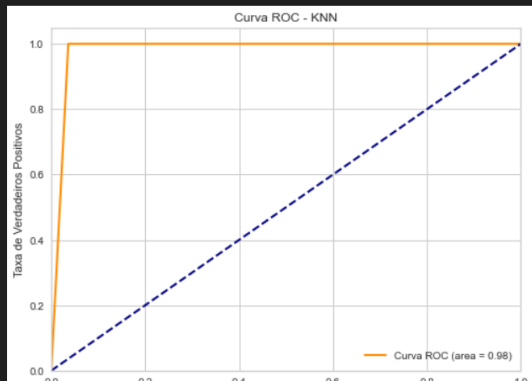
Relatório de Classificação:
      precision    recall  f1-score   support

    0.0         1.00     0.96     0.98         912
    1.0         0.97     1.00     0.98         959

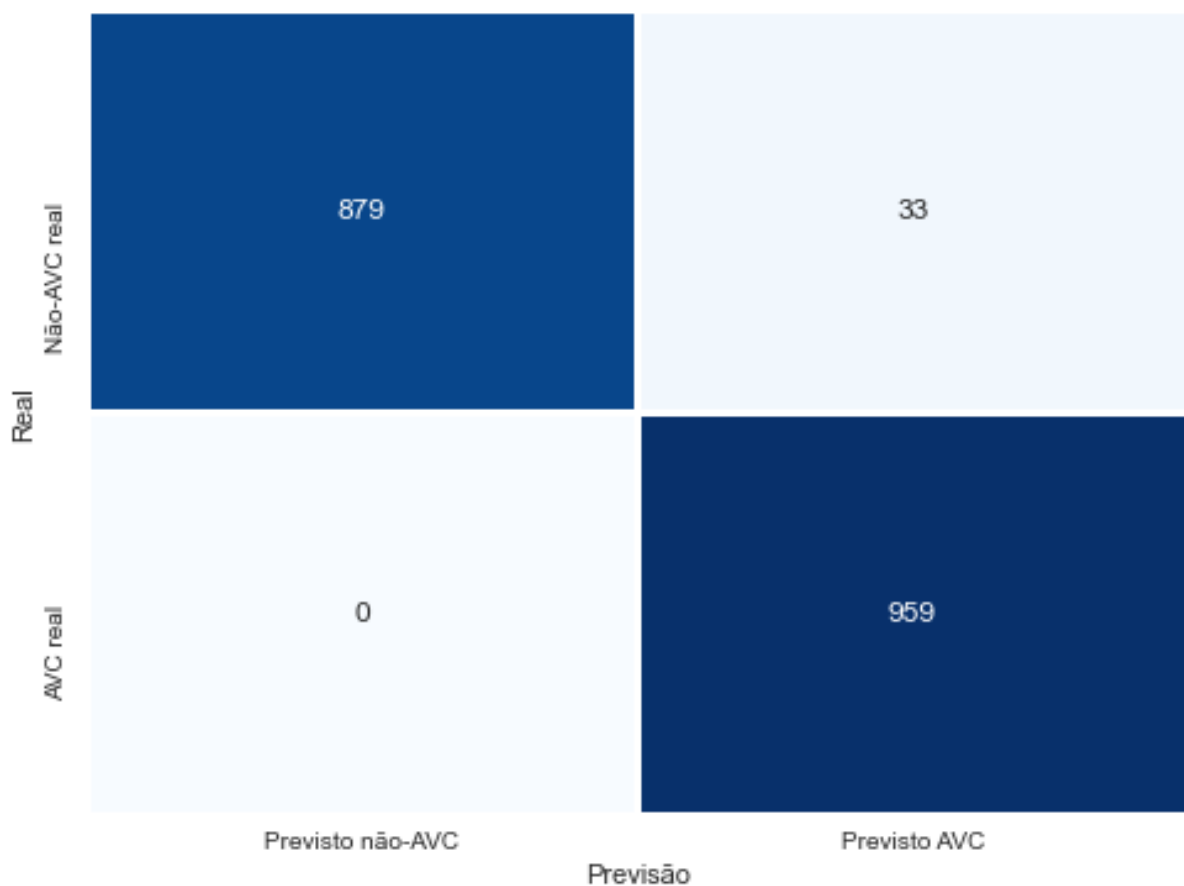
 accuracy         0.98
  macro avg         0.98
 weighted avg         0.98

Precisão de cada validação cruzada: [0.97327989 0.96593186 0.95991984 0.97593583 0.97727273]
Precisão Média validação cruzada: 0.970 +/- 0.007

```



Matriz de Confusão - KNN



O modelo de KNN apresentou como melhor resultado precisão média de validação cruzada em 5 partes = 0,970 e área da curva ROC = 0,98.

Random Forest

Figura 41 - Implementação e resultado dos modelo de Random Forest

```

param_grid_rf = {
    'n_estimators': [50, 100, 200, 400, 800],
    'max_depth': [None, 10, 20],
    'min_samples_split': [2, 3, 5, 7, 10],
    'min_samples_leaf': [1, 2, 3],
}

rf_classifier = RandomForestClassifier(random_state=0)

grid_search_rf = GridSearchCV(
    estimator=rf_classifier,
    param_grid=param_grid_rf,
    scoring='accuracy',
    cv=5,
    n_jobs=-1,
)

grid_search_rf.fit(x_train, y_train)

best_params_rf = grid_search_rf.best_params_

best_rf_classifier = RandomForestClassifier(random_state=42, **best_params_rf)
best_rf_classifier.fit(x_train, y_train)

y_pred_rf = best_rf_classifier.predict(x_test)

# Calculate accuracy and print the classification report
accuracy = accuracy_score(y_test, y_pred_rf)
print("Melhores Parâmetros:", best_params_rf)
print("Precisão no grupo de teste:", accuracy)
print("\nRelatório de Classificação:\n", classification_report(y_test, y_pred_rf))

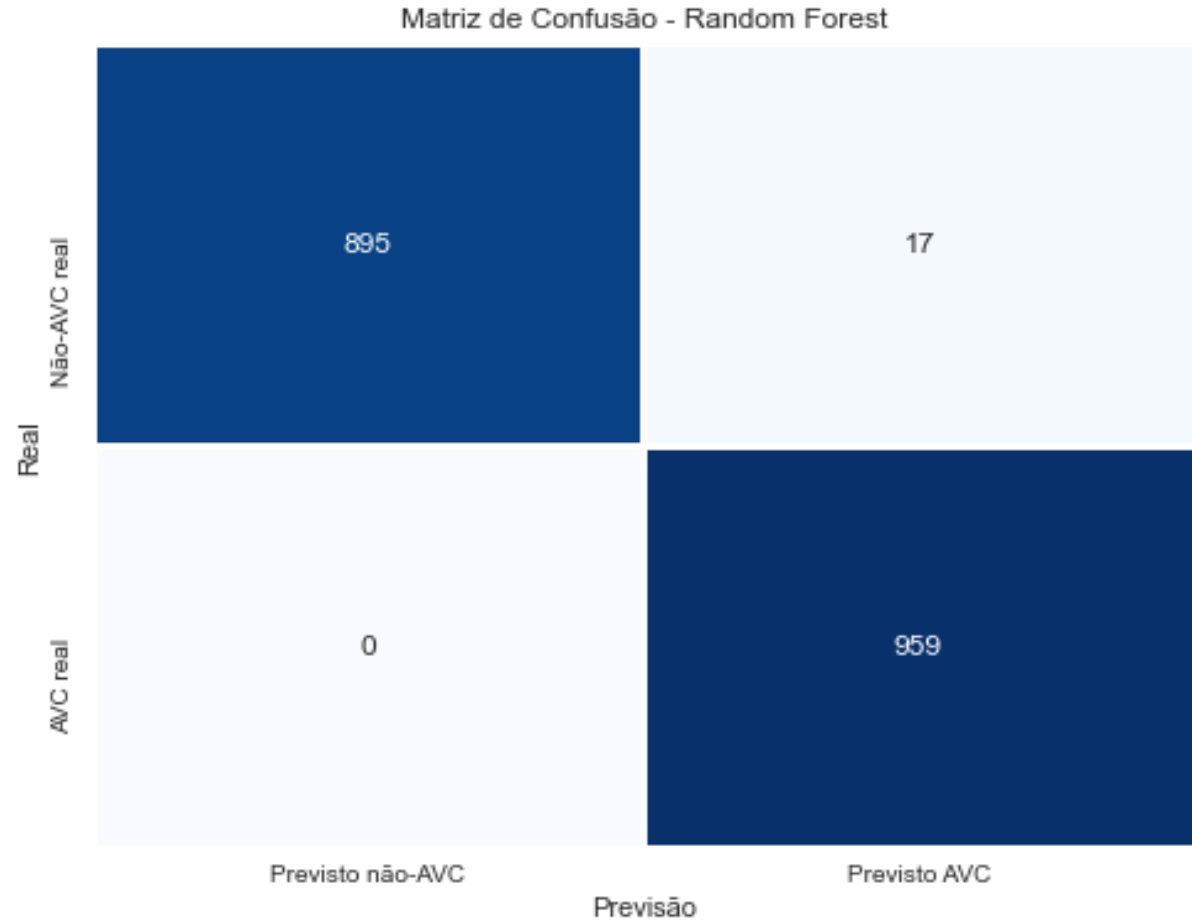
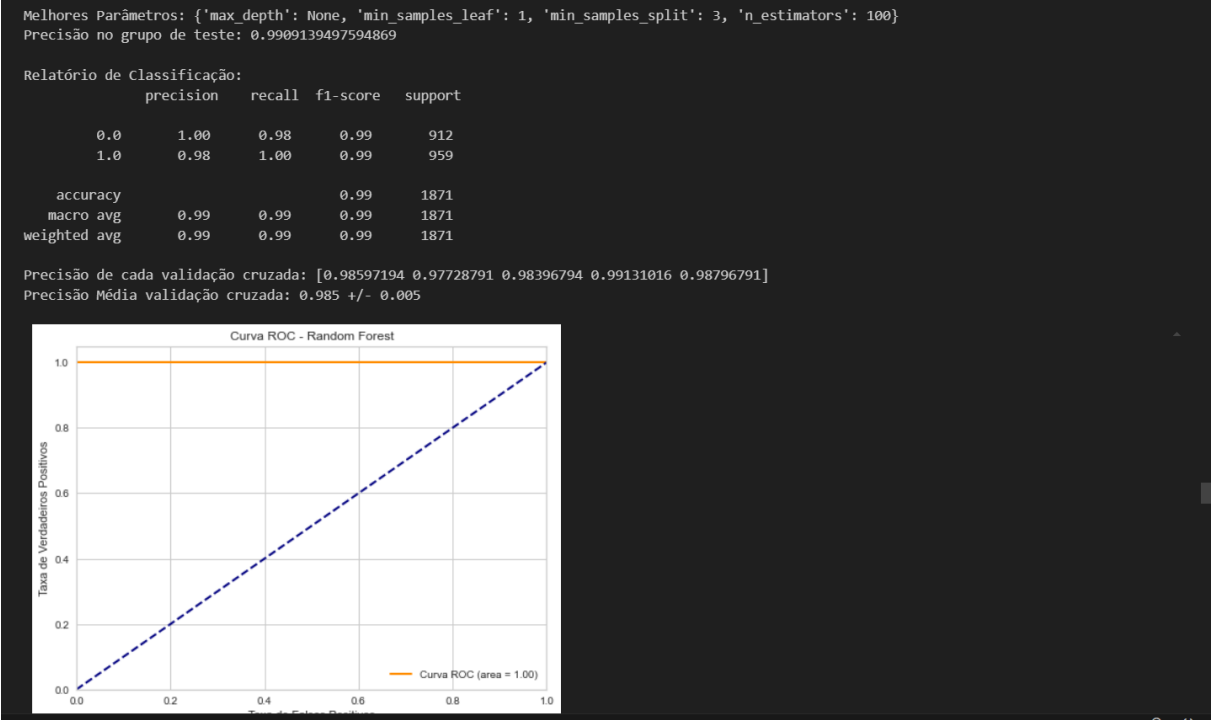
#Crossvalidation
scores_rf = cross_val_score(best_rf_classifier, X=x_train, y=y_train, cv=5, n_jobs=1)
print('Precisão de cada validação cruzada: %s' % scores_rf)
print('Precisão Média validação cruzada: %.3f +/- %.3f' % (np.mean(scores_rf), np.std(scores_rf)))

#ROC curve
roc_auc = roc_auc_score(y_test, best_rf_classifier.predict_proba(x_test)[:, 1])
fpr, tpr, _ = roc_curve(y_test, best_rf_classifier.predict_proba(x_test)[:, 1])

plt.figure(figsize=(8, 6))
plt.plot(fpr, tpr, color='darkorange', lw=2, label='Curva ROC (area = {:.2f})'.format(roc_auc))
plt.plot([0, 1], [0, 1], color='navy', lw=2, linestyle='--')
plt.xlim([0.0, 1.0])
plt.ylim([0.0, 1.05])
plt.xlabel('Taxa de Falsos Positivos')
plt.ylabel('Taxa de Verdadeiros Positivos')
plt.title('Curva ROC - Random Forest')
plt.legend(loc='lower right')
plt.show()

# Confusion Matrix
conf_matrix_rf = confusion_matrix(y_test, y_pred_rf)
plt.figure(figsize=(8, 6))
sns.heatmap(conf_matrix_rf, annot=True, fmt="d", cmap='Blues', linewidths=2.5, cbar=False)
plt.xlabel('Previsão')
plt.ylabel('Real')
plt.title('Matriz de Confusão - Random Forest')
plt.xticks([0.5, 1.5], ['Previsto não-AVC', 'Previsto AVC'])
plt.yticks([0.5, 1.5], ['Não-AVC real', 'AVC real'])
plt.show()

```



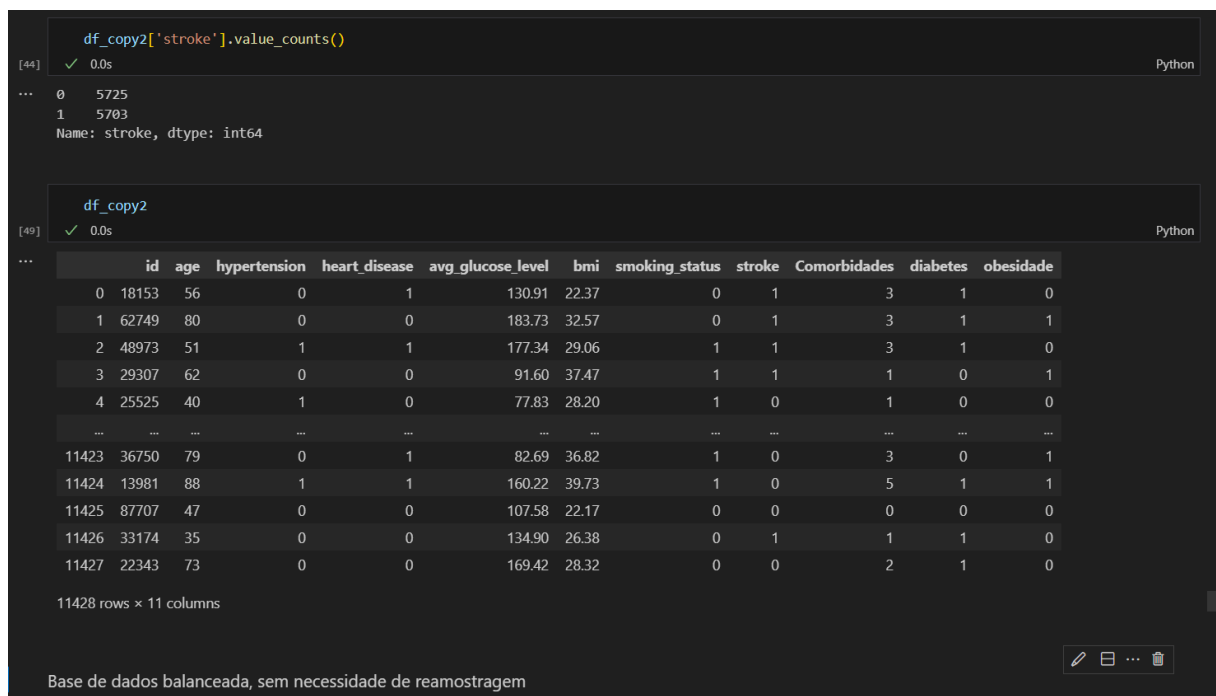
Fonte: Autor

O modelo de Random Forest apresentou como melhor resultado precisão média de validação cruzada em 5 partes = 0,985 e área da curva ROC = 1,00.

Teste do modelo com os dados artificiais (Team Incrigo)

Primeiro os dados foram devidamente preparados seguindo a metodologia usada no conjunto de dados anterior, chegando no seguinte *dataset* final (Figura 42), que foi posteriormente dividido da mesma maneira em grupos de treino e teste.

Figura 42 - Conjunto de dados a ser analisado



The screenshot shows two Jupyter Notebook cells. The first cell, labeled [44], contains the code `df_copy2['stroke'].value_counts()` and displays the following output:

```
0    5725
1    5703
Name: stroke, dtype: int64
```

The second cell, labeled [49], contains the code `df_copy2` and displays a preview of the dataset with 11428 rows and 11 columns. The columns are: id, age, hypertension, heart_disease, avg_glucose_level, bmi, smoking_status, stroke, Comorbidades, diabetes, and obesidade. The first few rows of data are shown below:

	id	age	hypertension	heart_disease	avg_glucose_level	bmi	smoking_status	stroke	Comorbidades	diabetes	obesidade
0	18153	56	0	1	130.91	22.37	0	1	3	1	0
1	62749	80	0	0	183.73	32.57	0	1	3	1	1
2	48973	51	1	1	177.34	29.06	1	1	3	1	0
3	29307	62	0	0	91.60	37.47	1	1	1	0	1
4	25525	40	1	0	77.83	28.20	1	0	1	0	0

At the bottom of the notebook interface, it states: "Base de dados balanceada, sem necessidade de reamostragem".

Fonte: Autor

Para esses dados não foi treinado um novo modelo, pelos motivos citados no início deste Capítulo, tampouco era esperado uma boa performance do modelo quando aplicado a estes dados, a implementação e os resultados seguem a seguir (Figura 43).

Figura 43 - Implementação e resultados do teste do modelo com dados sintéticos

```

y_pred_rf2 = best_rf_classifier.predict(x_test2)

# Calculate accuracy and print the classification report
accuracy2 = accuracy_score(y_test2, y_pred_rf2)
print("Melhores Parâmetros:", best_params_rf)
print("Precisão no grupo de teste:", accuracy2)
print("\nRelatório de Classificação:\n", classification_report(y_test2, y_pred_rf2))

#Crossvalidation
scores_rf2 = cross_val_score(best_rf_classifier, X=x_train2, y=y_train2, cv=5, n_jobs=1)
print('Precisão de cada validação cruzada: %s' % scores_rf2)
print('Precisão Média validação cruzada: %.3f +/- %.3f' % (np.mean(scores_rf2), np.std(scores_rf2)))

#ROC curve
roc_auc2 = roc_auc_score(y_test2, best_rf_classifier.predict_proba(x_test2)[:, 1])
fpr2, tpr2, _ = roc_curve(y_test2, best_rf_classifier.predict_proba(x_test2)[:, 1])

plt.figure(figsize=(8, 6))
plt.plot(fpr2, tpr2, color='darkorange', lw=2, label='Curva ROC (area = {:.2f})'.format(roc_auc2))
plt.plot([0, 1], [0, 1], color='navy', lw=2, linestyle='--')
plt.xlim([0.0, 1.0])
plt.ylim([0.0, 1.05])
plt.xlabel('Taxa de Falsos Positivos')
plt.ylabel('Taxa de Verdadeiros Positivos')
plt.title('Curva ROC - Random Forest')
plt.legend(loc='lower right')
plt.show()

# Confusion Matrix
conf_matrix_rf2 = confusion_matrix(y_test2, y_pred_rf2)
plt.figure(figsize=(8, 6))
sns.heatmap(conf_matrix_rf2, annot=True, fmt="d", cmap='Blues', linewidths=2.5, cbar=False)
plt.xlabel('Previsão')
plt.ylabel('Real')
plt.title('Matriz de Confusão - Random Forest')
plt.xticks([0.5, 1.5], ['Previsto não-AVC', 'Previsto AVC'])
plt.yticks([0.5, 1.5], ['Não-AVC real', 'AVC real'])
plt.show()

# Plot the feature importances

fig2, ax2 = plt.subplots()
viz_incribo = FeatureImportances(best_rf_classifier)
viz_incribo.fit(x_train2, y_train2)
indices_incribo = viz_incribo.features_
y_ordered_incribo = list()
for i in indices_incribo:
    y_ordered_incribo.append(colunas[i])

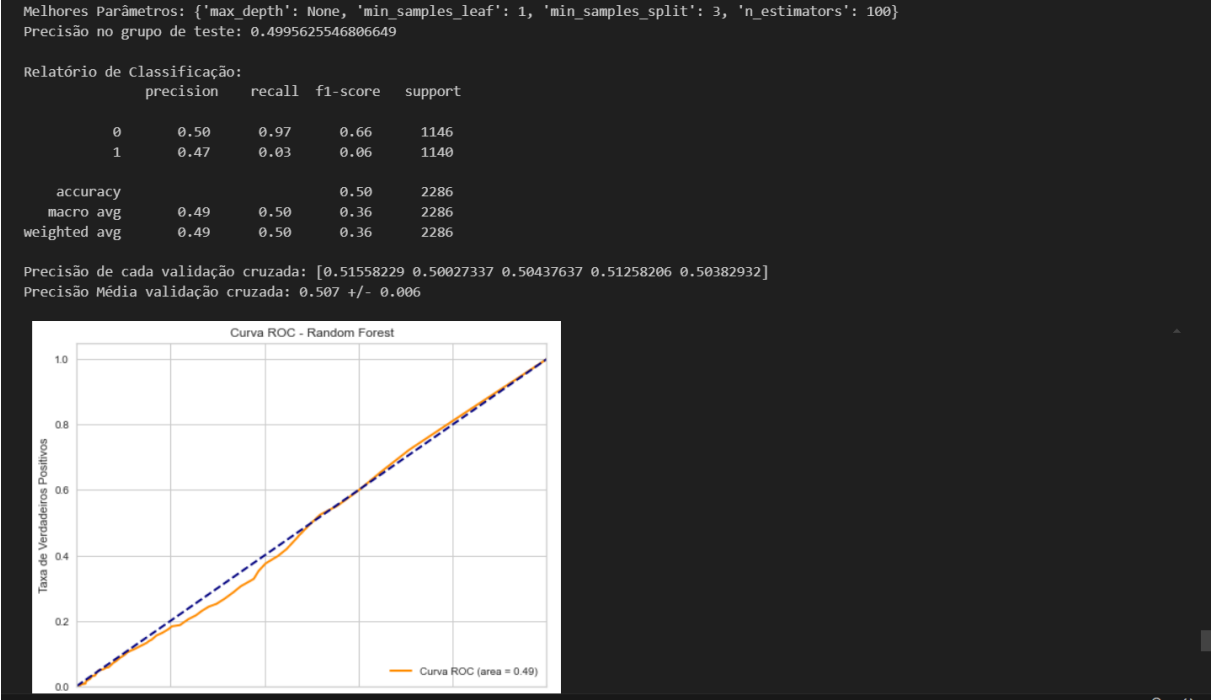
ax2.set_yticklabels(y_ordered_incribo)

plt.tick_params(labelsize=13)
plt.title('Feature Importance - Random Forest')

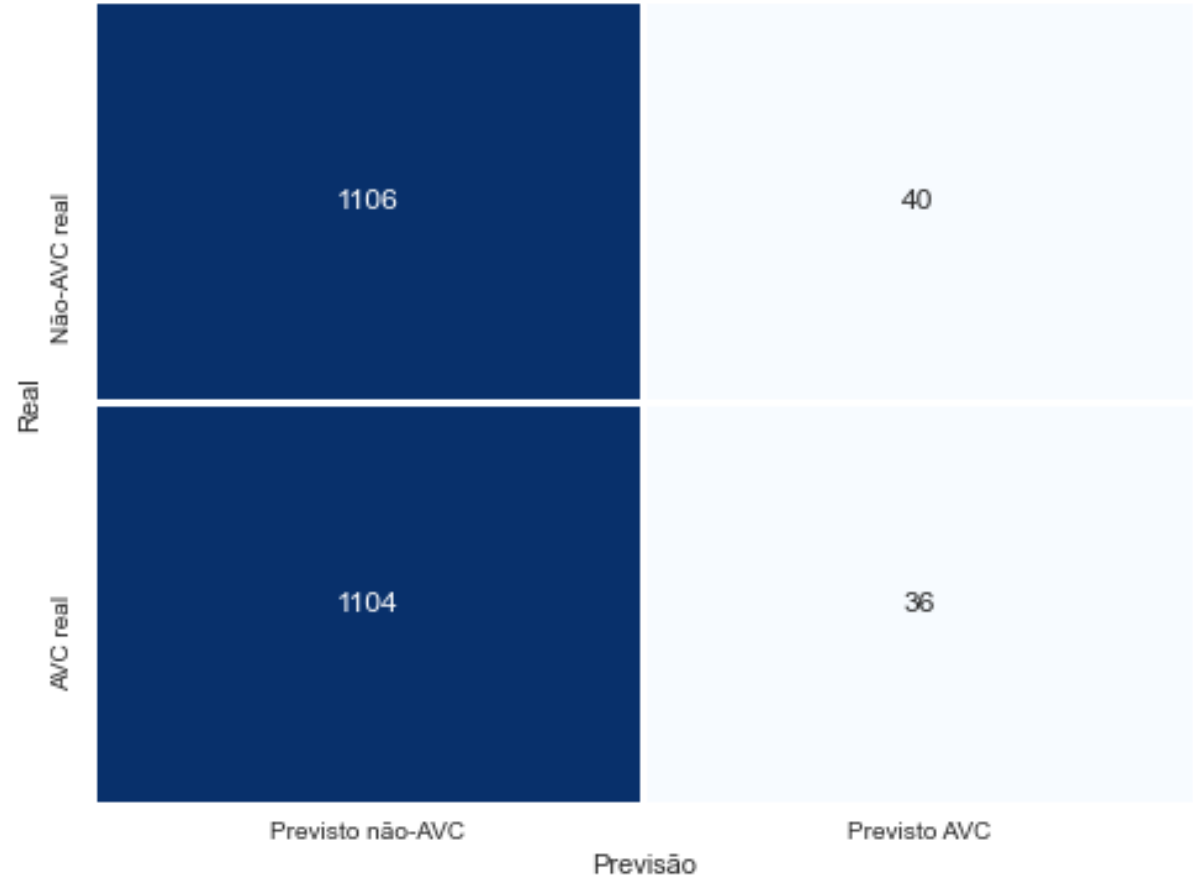
plt.show()

```

✓ 11.9s Python



Matriz de Confusão - Random Forest



Fonte: Autor

O modelo treinado com dados reais não foi capaz de prever o conjunto de dados artificiais, fato esse que era esperado devido à potencial falta de verossimilhança dos dados artificiais com a realidade.

O desempenho de precisão de validação cruzada 0,507 numa base de dados balanceada significa que não foram identificados padrões em comum entre os dados em que o modelo foi treinado e os dados em que ele foi testado que, nesse caso, eram de outro *dataset*. Essa hipótese é confirmada pela Matriz de confusão que demonstra que foi previsto “não-AVC” para quase toda a amostra, justificando resultado próximo de 0,5.

6. Interpretação dos Resultados

Os resultados obtidos no modelo de dados reais se mostram promissores, especialmente no tocante ao novo atributo criado, chamado de “Comorbidades”. Além disso, há de se confirmar a veracidade do Estudo com base em novos dados, também de pacientes reais e, de preferência, do mesmo país de onde foram retirados os dados originais (Estados Unidos da América).

Outra análise importante a ser feita é sobre a temporalidade do estudo. Esses dados são de natureza única no tempo, são apenas uma foto momentânea das condições de vida desses pacientes. A Medicina Baseada em Evidências (MBE) preconiza o uso de estudos longitudinais em detrimento dos estudos transversais, portanto, caso existam dados de séries temporais de cada paciente pode-se realizar um estudo com maior consistência de resultados e potencialmente maior impacto na saúde das pessoas.

Estudos longitudinais tendem a ser mais caros, portanto, é de suma importância que não sejam analisadas muitas variáveis e que os pacientes sigam nesse estudo pela maior quantidade de tempo possível. Um conjunto de atributos pequeno capaz de prever a variável alvo é imprescindível para tornar tal análise válida. O presente trabalho foi capaz de prever a variável alvo AVC com o uso de 9 atributos e um simples modelo de aprendizado de máquina, dentro desses 9 atributos, os dados que devem ser adquiridos do paciente são:

Idade: Através de um questionário simples, de custo basicamente nulo;

Hipertensão: Através da medição de Pressão Arterial Sistêmica (PAS) do paciente, comum em exames de rotina e idealmente mais de uma medição para confirmação de diagnóstico;

Doenças cardíacas: Devem ser rastreadas em um momento diferente ao desse estudo, para a análise desses dados seria aplicado apenas um questionário sobre doenças pregressas;

IMC e Obesidade: Definidos através das medições de altura e massa corporal;

Diabetes e Média Glicêmica: Através da medição de glicemia em jejum, também comum em exames de rotina e idealmente com mais de uma medição para confirmação de diagnóstico;

Condição de fumante: Definido através de um questionário;

Comorbidades: Definido através dos atributos anteriores, sem ter que perguntar nenhuma nova informação ao paciente.

Dessa forma, seria necessário aplicar apenas um questionário e as medições de PAS, glicemia em jejum, altura e peso do paciente, dados esses que estão plenamente disponíveis em exames de rotina das pessoas.

Além desses dados, seria interessante incluir mais dois atributos com outros exames de rotina:

HDL e Colesterol Total: Esses atributos não estiveram presentes nesse estudo mas podem ser de grande valia ao analisar a população real.

7. Links e Referências

Link do repositório github:

Link do vídeo do youtube:

Link dos datasets da kaggle: <https://www.kaggle.com/datasets/fedesoriano/stroke-prediction-dataset>

Link dos datasets da kaggle: <https://www.kaggle.com/datasets/teamincrito/stroke-prediction>

- 1- <https://www.gov.br/saude/pt-br/assuntos/saude-de-a-a-z/a/avc>
- 2- <https://www.who.int/news-room/fact-sheets/detail/the-top-10-causes-of-death>
- 3- <https://www.who.int/data/gho/data/themes/mortality-and-global-health-estimates/ghe-leading-causes-of-death>
- 4- <https://deathmeters.info/>
- 5- Collaborators, D., Burden, G., & Study, D. (2024). *Supplementary appendix 2 Appendix 2 : supplementary results appendix to “ Global burden of 288 causes of death and life expectancy decomposition in 204 countries and territories and 811 subnational locations , 1990 – 2021 : a systematic analysis for the G. 6736(24), 1990–2021.*
- 6- Boehme, A. K., Esenwa, C., & Elkind, M. S. V. (2017). Stroke Risk Factors, Genetics, and Prevention. *Circulation Research*, 120(3), 472–495.
<https://doi.org/10.1161/CIRCRESAHA.116.308398>
- 7- Caderno de Atenção Básica N. 16, Ministério da Saúde, 2006
- 8- https://bvsmis.saude.gov.br/bvs/dicas/215_obesidade.html
- 9- https://pt.wikipedia.org/wiki/Pirâmide_etária#/media/Ficheiro:Dtm_pyramids.png