RECA
REVISTA ELETRÔNICA DE COMPUTAÇÃO APLICADA
VOL. 3 NRO. 2 2022

MODELO PREDITIVO PARA IDENTIFICAÇÃO DE DOENÇAS CARDÍACAS

Guilherme Caravieri Alves
Graduando em Ciência da Computação – Uni-FACEF
caravierigui14@gmail.com

Jaqueline Brigladori Pugliesi
Doutora em Ciências – USP São Carlos
jbpugliesi@gmail.com

Resumo

Considerada a principal causa de mortes em nosso país, as doenças cardíacas, (DCV), relacionadas a todo o sistema circulatório e ao coração, atingem mais de 1.100 pessoas por dia, de acordo com dados da Sociedade Brasileira de Cardiologia. Mediante este cenário preocupante, o presente estudo tem por objetivo o desenvolvimento de um modelo de análise preditiva a fim de identificar futuros casos de doenças coronárias e, assim, buscar um algoritmo de classificação, visando um diagnóstico de futuros casos de DCV. Desta forma, para aprofundar o tema e alcançar o objetivo acadêmico, foi necessário a utilização de informações de pacientes de uma base de dados. O estudo envolveu homens e mulheres de diferentes idades, e os dados coletados para fins de pesquisa foram o banco de dados em Cleveland, o *dataset* do repositório UCI, e a análise do projeto exploratório feita com a linguagem Python. Atualmente, vários são os fatores que influenciam no surgimento de tais doenças, como sedentarismo, tabagismo, uso exagerado de álcool e drogas, obesidade, além de outros fatores comportamentais que comprometem a vida de milhares de indivíduos no mundo todo.

Palavras-chave: Doenças cardíacas. Aprendizado de máquina. Análise de dados. Modelo preditivo.

Abstract

Considered the main cause of death in our country, heart diseases (CVD), related to the entire circulatory system and the heart, affect more than 1,100 people a day, according to data from the Brazilian Society of Cardiology. Given this worrying scenario, the present study aims to develop a predictive analysis model in order to identify future cases of coronary heart disease and, thus, seek a classification algorithm, aiming at a diagnosis of future cases of CVD. Thus, to deepen the topic and achieve the academic objective, it was necessary to use patient information from a database. The study involved men and women of different ages, and the data collected for research purposes were the database in Cleveland, the UCI repository dataset, and the exploratory project analysis done with the Python language. Currently, there are several factors that influence the emergence of such diseases, such as sedentary lifestyle, smoking, excessive use of alcohol and drugs, obesity, in addition to other behavioral factors that compromise the lives of thousands of individuals worldwide.

Keywords: *Heart diseases. Machine learning. Data analysis. Predictive model.*

1 Introdução

Doença cardiovascular é uma das doenças que mais mata no mundo. De acordo com a World Heart Federation, 18,6 milhões de pessoas são atingidas por diferentes tipos de doenças cardiovasculares, onde 85% morrem dessas doenças coronárias (WORLD HEART FEDERATION, *sd*).

O termo doença cardiovascular faz referência a toda e qualquer alteração que possa acontecer no coração, assim como nos vasos sanguíneos, sendo conhecida também pelo nome de cardiopatia. Tais doenças podem ser causadas ainda por fatores relacionados à genética ou ao comportamento do indivíduo, levando-o a desenvolver diferentes variações agravantes ou não.

A metodologia de pesquisa para o desenvolvimento do artigo tem caráter exploratório, no qual busca analisar os dados do *dataset*, ao abordar doenças cardiovasculares que afetam o coração e os vasos sanguíneos.

A análise e a preparação da base de dados foram feitas a partir da linguagem Python e suas bibliotecas pandas, numpy e matplotlib. Para tanto, foi usado o Google Colab, no qual foram feitas as configurações para o ambiente de análise de dados.

2 Referencial Teórico

Esta seção aborda conceitos importantes e necessários para a compreensão do tema do trabalho, sendo eles doenças cardiovasculares e *Machine Learning*.

2.1 Doenças Cardiovasculares

De acordo com a World Heart Federation, as doenças cardiovasculares (DCV) são uma classe de doenças que afetam o coração ou os vasos sanguíneos. As DCV são divididas entre doença arterial coronariana, insuficiência cardíaca etc.

Os fatores que causam doenças cardíacas são o tabagismo, a obesidade, o sedentarismo, os altos níveis de gordura na corrente sanguínea e até o uso nocivo do álcool. “Ataques cardíacos e acidentes vasculares cerebrais geralmente são eventos agudos causados principalmente por um bloqueio que impede que o sangue flua para o coração ou para o cérebro.” (OPAS/OMS, *sd*, *online*).

Essas doenças são consideradas as que mais matam em todo o mundo, 85% das mortes por ano são vítimas de doenças coronárias ou cerebrovasculares. Segundo a Sociedade Brasileira de Cardiologia, a doença provoca mais de 1.100 mortes por dia, sendo o equivalente a 46 pessoas morrendo por hora (OPAS/OMS, *sd*, *online*).

> A doença cardíaca permanece a principal causa de morte em todo o mundo nos últimos 20 anos. No entanto, agora está matando mais pessoas do que

> nunca. O número de mortes por doenças cardíacas aumentou em mais de 2 milhões desde o ano 2000 para quase 9 milhões em 2019 (NAÇÕES UNIDAS, 2020, *online*).

A principal DCV seria a arterial coronariana, que é classificada como uma doença isquêmica, que ocorre com a obstrução dos vasos que levam o sangue para que o músculo cardíaco se contraia. A causa dessa obstrução é o acúmulo de placas de gordura, fazendo com que a passagem do sangue até o coração seja limitada (CARDIO, sd, online).

Um de seus sintomas é um desconforto ou dor no peito, que é chamado de angina. Esse desconforto pode se espalhar por outras partes do corpo. Essa sensação ocorre quando o coração passa por uma sobrecarga, em situações de atividade física ou *stress*. Assim, quando o fluxo de sangue para o músculo cardíaco estiver obstruído, o paciente pode passar por complicações como infarto (OPAS/OMS, *sd*, *online*).

.

A insuficiência cardíaca ocorre quando o coração perde a capacidade de bombear corretamente o sangue, a causa principal seria as doenças isquêmicas, pois a falta de fluxo sanguíneo faz com que o coração não consiga bombear sangue para o resto do corpo (OPAS/OMS, *sd*, *online*).

> Em geral, a insuficiência cardíaca é uma consequência de doenças que, ao longo do tempo, agridem o coração. Cerca de 70% dos casos estão relacionados às doenças isquêmicas, como infarto, além de hipertensão, problemas valvares e miocardite. Esses problemas acabam afetando o músculo cardíaco com o passar do tempo, reduzindo sua capacidade de bombear o sangue. (CARDIO, *sd*, *online*).

Isso mostra que as doenças coronárias devem ser identificadas o quanto antes para o tratamento.

2.2 *Machine Learning*

Segundo Borges (2020; apud MOHRI, ROSTAMIZADEH e TALWALKAR, 2018, *online*):

> *Machine Learning* pode ser definida como um método computacional para predição de dados, ou seja, a partir de uma base de dados, como por exemplo, uma série temporal que apresenta os dados ao longo do tempo, a "máquina" aprende sobre os padrões de comportamento destes dados através de algoritmos com o objetivo de predizer um resultado futuro.

Portanto, pode-se dividir esses algoritmos em dois principais tipos: o não supervisionado e o supervisionado. Para o artigo em questão foi utilizado o método supervisionado.

> O aprendizado supervisionado é baseado no treinamento de uma amostra de dados com a classificação correta já atribuída, enquanto o não supervisionado se refere à capacidade de aprender e organizar informações sem a atribuição da classificação correta (PAIXÃO *et al*., 2022, *online*).

O algoritmo de aprendizado supervisionado tem como método a comparação dos resultados do modelo e a classificação dos dados. Desse modo o algoritmo de treinamento é repetido até o erro ser mínimo (PAIXÃO et al., 2022).

A análise preditiva corresponde ao uso de algoritmos para entender os dados e criar regras de predição, sendo um de seus usos o de estimar os desfechos de uma situação (PAIXÃO et al., 2019). Além disso, considera-se uma ciência que busca usar os dados para a tomada de decisão (BARI, CHAOUCHI, JUNG, 2017).

3 Materiais e Métodos

Nesta seção são apresentados os materiais e métodos utilizados ao longo do desenvolvimento do presente artigo, cujo objetivo é analisar os dados e fazer a predição de diagnósticos de doenças cardíacas.

3.1 *Google Colab*

A empresa multinacional Google criou uma ferramenta para Ciência de Dados e Aprendizado de Máquina, o Google Colab ou Google *Colaboratory*. A ferramenta permite executar a linguagem Python no navegador, utilizando o Jupyter, sendo um serviço sem fins lucrativos, usando os servidores em nuvem do Google, sendo necessário somente sua conta do Gmail (COLAB, *sd*).

Com o Google Colab é possível fazer a importação de bibliotecas do Python, como o *numpy* e o *matplotlib*, para uso de algoritmos de treinamentos e desenvolvimentos de redes neurais.

3.2 Linguagem Python e suas bibliotecas

Criada pelo programador Guido Van Rossum, na década de 90, Python é uma das linguagens de programação mais usadas no mundo, segundo o ranking realizado em 23 de agosto de 2022, feito pelo Instituto de Engenheiros Elétricos e Eletrônicos (CANALTECH, 2022). sendo considerada uma linguagem de programação de alto nível, mesmo tendo uma sintaxe de fácil compreensão (PYTHON, sd).

Segundo documentação da bibliotaca, é uma linguagem poderosa e rápida, utilizada para diferentes ramos da tecnologia, como o desenvolvimento, análise de dados e Inteligência Artificial. Atualmente, o Python está sendo a linguagem mais escolhida para se trabalhar com dados (PYTHON, 2022).

De acordo com o site da biblioteca Pandas, “panda é uma ferramenta de análise e manipulação de dados de código aberto rápida, poderosa, flexível e fácil de usar” (PANDAS, sd, online) construído sobre a linguagem de programação Python. A biblioteca permite a fácil manipulação e análise dos dados.

A biblioteca Seaborn é utilizada para visualização dos dados, e tem seu fundamento na linguagem Python, sendo usada para criação de gráficos em alto nível e informativos. Uma das bibliotecas também utilizadas foi a matplotlib para a

visualização dos dados e a criação de gráficos, no site da própria biblioteca matplotlib.org, ela é descrita como “uma biblioteca abrangente para criar visualizações estáticas, animadas e interativas em Python” (SEABORN, sd, online)

Sklearn é uma biblioteca do Python para treino e modelos de classificação (SCIKIT-LEARN,sd), e para essa pesquisa foram utilizados os algoritmos: *LogisticRegression*, *Naive Bayes*, *KNeighborsClassifier* e *Support Vector Machines* (SVM).

4 Resultado do Processo de Análise de Dados

O objetivo da análise de dados, com o auxílio do aprendizado de máquina para a análise preditiva, tem o intuito de identificar futuros casos de doenças cardíacas em pacientes.

4.1 Análise do *DataSet*

O *dataset* analisado é intitulado *heart.csv*, que possui dados de doenças cardíacas, sendo constituído por 303 instâncias e 14 atributos, que são: *age* (idade)*, sex* (sexo) *, cp* (tipo de dor)*, trestbps (pressão arterial em repouso), chol (colesterol), fbs* (glicemia em jejum), *restecg* (resultados eletrocardiográficos)*, thalach* (frequência cardíaca máxima) *, exang* (angina induzida)*, oldpeak* (depressão do segmento)*, slope* (inclinação do segmento)*, ca* (número de vasos)*, thal* (tipos)*, target* (diagnóstico de doença cardíaca). O último atributo que é o atributo meta, foco da análise e do modelo preditivo.

A Figura 1 ilustra os detalhes da importação do *dataset e* das bibliotecas. No primeiro momento é feita a conexão do Colab com o Google Drive, onde o *dataset* está armazenado. Em seguida é realizada a importação das bibliotecas, que foram apresentadas na subseção 3.2, sendo feito o uso de uma função da biblioteca Pandas, o *read_csv*, que é responsável por fazer a leitura do *dataset*, e o armazenamento do *dataset* na variável *heart*.

RECA
REVISTA ELETRÔNICA DE COMPUTAÇÃO APLICADA
VOL. 3 NRO. 2 2022

Figura 1 – Importação do *dataset* e das bibliotecas

```
from google.colab import drive
drive.mount('/content/drive')

from os import sep
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn import model_selection
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.neighbors import KNeighborsClassifier
from sklearn.naive_bayes import GaussianNB
from sklearn.metrics import accuracy_score
from sklearn.svm import SVC



heart = pd.read_csv ('/content/drive/My Drive/Faculdade/TCC/DataSet/heart.csv', sep= ',')
```

Fonte: os autores

Foi utilizada a função *head* para criação de um *data frame* com os primeiros 5 registros (Figura 2), onde cada instância representa um paciente e seus atributos.

Figura 2 – Primeiros registros do dataset

```
heart.head()
```

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
0	63	1	3	145	233	1	0	150	0	2.3	0	0	1	1
1	37	1	2	130	250	0	1	187	0	3.5	0	0	2	1
2	41	0	1	130	204	0	0	172	0	1.4	2	0	2	1
3	56	1	1	120	236	0	1	178	0	0.8	2	0	2	1
4	57	0	0	120	354	0	1	163	1	0.6	2	0	2	1

Fonte: os autores

Com a função *info* (Figura 3), pode-se visualizar as informações do *dataset*, o qual contém o nome das 14 colunas e o seu *data type* (tipo de dado). Nota-se que a maioria é do tipo *int* (inteiro) e *float* (valor fracionário), e possui 303 registros de pacientes.

RECA
REVISTA ELETRÔNICA DE COMPUTAÇÃO APLICADA
VOL. 3 NRO. 2 2022

Figura 3 – Informações do *data frame*

```
heart.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 303 entries, 0 to 302
Data columns (total 14 columns):
 #   Column    Non-Null Count  Dtype
---  ------    --------------  -----
 0   age       303 non-null    int64
 1   sex       303 non-null    int64
 2   cp        303 non-null    int64
 3   trestbps  303 non-null    int64
 4   chol      303 non-null    int64
 5   fbs       303 non-null    int64
 6   restecg   303 non-null    int64
 7   thalach   303 non-null    int64
 8   exang     303 non-null    int64
 9   oldpeak   303 non-null    float64
 10  slope     303 non-null    int64
 11  ca        303 non-null    int64
 12  thal      303 non-null    int64
 13  target    303 non-null    int64
dtypes: float64(1), int64(13)
memory usage: 33.3 KB
```

Fonte: os autores

Para verificar detalhes dos dados, utiliza-se a função *describe*, como mostrado na Figura 4. Sua função é fazer uma estatística descritiva dos dados, e mostrar a quantidade de dados, o valor máximo, o valor mínimo e a mediana dos dados de cada coluna.

Figura 4 – Estatística do *dataset*

```
heart.describe()
```

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
count	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000
mean	54.366337	0.683168	0.966997	131.623762	246.264026	0.148515	0.528053	149.646865	0.326733	1.039604	1.399340	0.729373	2.313531	0.544554
std	9.082101	0.466011	1.032052	17.538143	51.830751	0.356198	0.525860	22.905161	0.469794	1.161075	0.616226	1.022606	0.612277	0.498835
min	29.000000	0.000000	0.000000	94.000000	126.000000	0.000000	0.000000	71.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
25%	47.500000	0.000000	0.000000	120.000000	211.000000	0.000000	0.000000	133.500000	0.000000	0.000000	1.000000	0.000000	2.000000	0.000000
50%	55.000000	1.000000	1.000000	130.000000	240.000000	0.000000	1.000000	153.000000	0.000000	0.800000	1.000000	0.000000	2.000000	1.000000
75%	61.000000	1.000000	2.000000	140.000000	274.500000	0.000000	1.000000	166.000000	1.000000	1.600000	2.000000	1.000000	3.000000	1.000000
max	77.000000	1.000000	3.000000	200.000000	564.000000	1.000000	2.000000	202.000000	1.000000	6.200000	2.000000	4.000000	3.000000	1.000000

Fonte: os autores

Uma verificação para identificar se o *dataset heart* possui elementos nulos (*null*) é realizada, pois podem comprometer o resultado da análise e o modelo preditivo. Para isso,foi feito o uso de duas funções sendo elas o *isnull*, que verifica se o *dataset* possui esses elementos nulos e a *value_counts* que faz a contagem dos valores nulos. Como pode ser observado no *dataframe* (Figura 5), os elementos retornados são do tipo *false*, mostrando que o *dataset* não possui valores do tipo *null.*

RECA
REVISTA ELETRÔNICA DE COMPUTAÇÃO APLICADA
VOL. 3 NRO. 2 2022

Figura 5 – Verificação de valores nulos

```
heart.isnull().value_counts()

age    sex    cp     trestbps  chol   fbs    restecg  thalach  exang  oldpeak  slope  ca     thal   target
False  False  False  False     False  False  False    False    False  False    False  False  False  False     303
dtype: int64
```

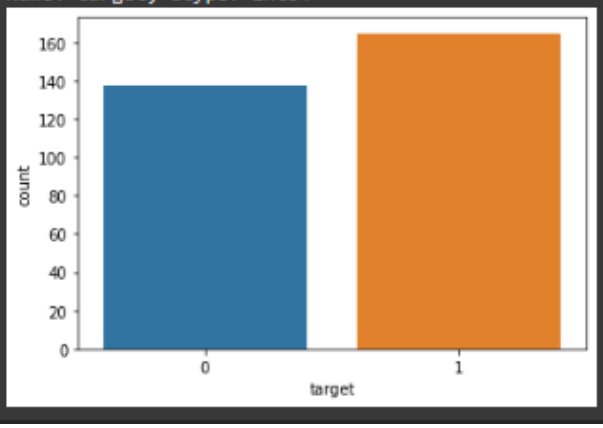
Fonte: os autores

A Figura 6 apresenta a distribuição dos valores da coluna *target*, que define se um paciente possui ou não algum tipo de doença cardíaca. Usa-se um *display* para apresentar os valores e passa como parâmetro o *dataset*, mas somente a coluna *target*. Com a função *value_counts*, faz-se a contagem dos valores da coluna. Por fim, utiliza-se da função *print* para mostrar um texto de descrição das variáveis dentro da coluna analisada. Além disso, é feita a construção e a plotagem de um gráfico de barras, com a função *countplot*, para apresentar a frequência da classe.

Figura 6 – Distribuição da variável *target*

```
print('1-Significa que possui algum tipo de doença cardiaca')
print('0-Significa que não possui doença cardiaca\n')

display(heart['target'].value_counts())

sns.countplot(x="target", data=heart)

plt.show()
```

```
1-Significa que possui algum tipo de doença cardiaca
0-Significa que não possui doença cardiaca

1    165
0    138
Name: target, dtype: int64
```



Fonte: os autores

4.2 Construção do Modelo Preditivo

Para a etapa de construção do Modelo Preditivo (Figura 7), foi feita a separação do *dataset* e a criação de uma variável x para incluir o *dataset* sem a variável *target*, e uma variável y para que possa ser indexado a coluna *target*. Foi utilizado a função *drop* para tirar a coluna *target*, desta forma é separado o *dataset*

para que possa ser feito a análise. Por fim, na Figura 8 e Figura 9, usa-se a função *print* para visualizar a separação.

Figura 7 – Separação do *dataset*

```
x = heart.drop(columns = 'target', axis = 1)
y = heart['target']
```

Fonte: os autores

Figura 8 – *Dataset* sem a coluna *target*

```
print(x)
```

```
      age  sex  cp  trestbps  chol  fbs  restecg  thalach  exang  oldpeak  \
0      63    1   3       145   233    1        0      150      0      2.3
1      37    1   2       130   250    0        1      187      0      3.5
2      41    0   1       130   204    0        0      172      0      1.4
3      56    1   1       120   236    0        1      178      0      0.8
4      57    0   0       120   354    0        1      163      1      0.6
..    ...  ...  ..       ...   ...  ...      ...      ...    ...      ...
298    57    0   0       140   241    0        1      123      1      0.2
299    45    1   3       110   264    0        1      132      0      1.2
300    68    1   0       144   193    1        1      141      0      3.4
301    57    1   0       130   131    0        1      115      1      1.2
302    57    0   1       130   236    0        0      174      0      0.0

     slope  ca  thal
0        0   0     1
1        0   0     2
2        2   0     2
3        2   0     2
4        2   0     2
..     ...  ..   ...
298      1   0     3
299      1   0     3
300      1   2     3
301      1   1     3
302      1   1     2

[303 rows x 13 columns]
```

Fonte: os autores

Figura 9 – *Dataset* somente com a coluna *target*

```
print(y)

0      1
1      1
2      1
3      1
4      1
      ..
298    0
299    0
300    0
301    0
302    0
Name: target, Length: 303, dtype: int64
```

Fonte: os autores

Com a divisão do *dataset* concluída é feita a separação do conjunto de treinamento e de teste (Figura 10). Cria-se as variáveis *x_train* e *y_train,* que corresponde ao treinamento, enquanto as variáveis *x_test* e *y_test que* é a parte de teste. É usada a função *train_test_split* para fazer a divisão de teste e treinamento, aleatoriamente, que possui como parâmetro as variáveis x e y, da separação anterior do *dataset*. Também foi utilizada a função *test_size* para que possa ser dividido a porcentagem de teste, que no caso foi de 30%.

Figura 10 – Separado o conjunto de dados de treinamento e de teste

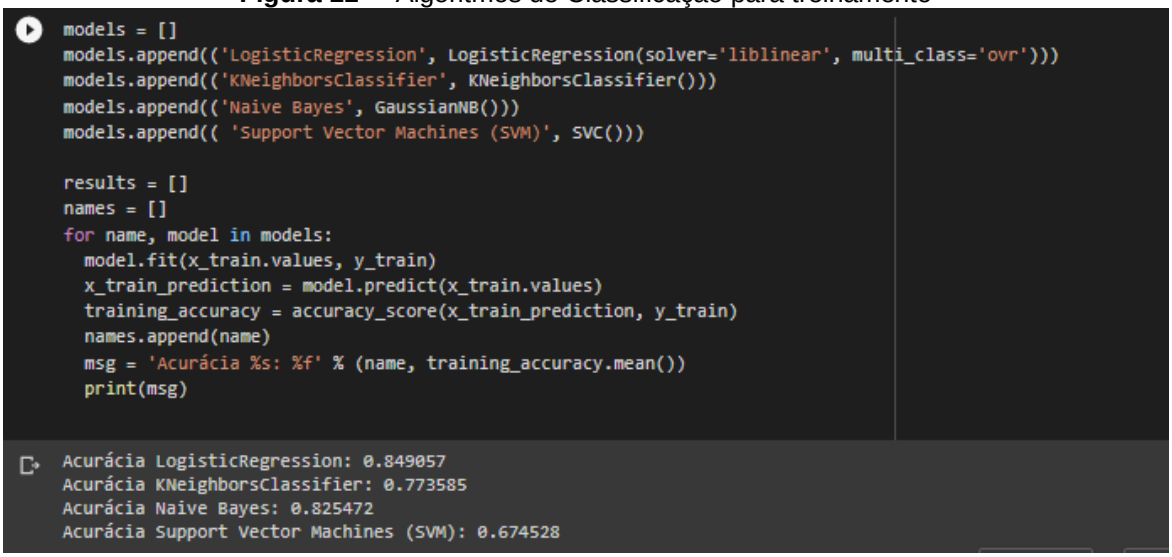
```
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.3, stratify=y, random_state=2)
print(x.shape, x_train.shape, x_test.shape)

(303, 13) (212, 13) (91, 13)
```

Fonte: os autores

4.3 Algoritmos de Classificação e Treinamento

O processo de treinamento é baseado no modelo de classificação escolhido. Para fazer a escolha é preciso levar em consideração para o que serve cada tipo de modelo. Foram utilizados os modelos *LogisticRegression*, *KNeighborsClassifier*, *Naive Bayes* e *Support Vector Machines*, que na literatura são muito ultilizados, com o objetivo de identificar qual é o maior nível de acurácia de cada modelo. Para isso, foi realiza a importação das bibliotecas e a declaração de cada um dos modelos de classificação. Na análise preditiva deve se ter em mente que o modelo preditivo não deve ter 100% de acurácia, pois assim é visto que ele não aprendeu a analisar os padrões, e sim, decorou os resultados. Em seguida, foram executados todos os quatro algoritmos a fim de avaliar o valor da acurácia de cada um dos modelos no conjunto de treinamento (Figura 11).

Figura 11 – Algoritmos de Classificação para treinamento



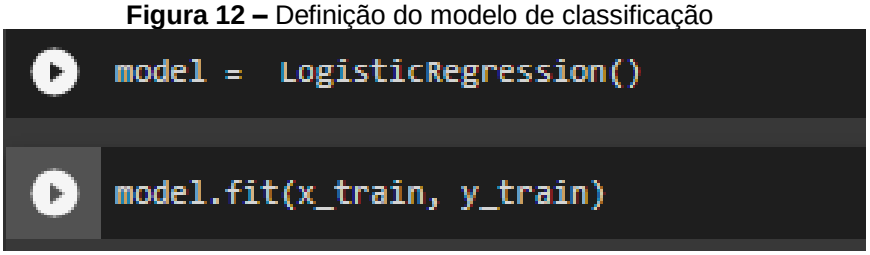
```
models = []
models.append(('LogisticRegression', LogisticRegression(solver='liblinear', multi_class='ovr')))
models.append(('KNeighborsClassifier', KNeighborsClassifier()))
models.append(('Naive Bayes', GaussianNB()))
models.append(( 'Support Vector Machines (SVM)', SVC()))

results = []
names = []
for name, model in models:
  model.fit(x_train.values, y_train)
  x_train_prediction = model.predict(x_train.values)
  training_accuracy = accuracy_score(x_train_prediction, y_train)
  names.append(name)
  msg = 'Acurácia %s: %f' % (name, training_accuracy.mean())
  print(msg)
```

```
Acurácia LogisticRegression: 0.849057
Acurácia KNeighborsClassifier: 0.773585
Acurácia Naive Bayes: 0.825472
Acurácia Support Vector Machines (SVM): 0.674528
```

Fonte: os autores

Na parte final da Figura 11 pode-se observar que o melhor modelo de classificação foi o *LogisticRegression*, com 85% de acurácia, sendo então o modelo escolhido. Na Figura 12 é declarado o modelo de classificação para variável model, e utilizada a função *fit* para ajustar o modelo, sendo passado como parâmetro as variáveis de treinamento.

Figura 12 – Definição do modelo de classificação



```
model =  LogisticRegression()
```

```
model.fit(x_train, y_train)
```

Fonte: os autores

Após toda a análise e definição do treinamento, é feito o uso do algoritmo de predição para casos futuros de doenças cárdicas. Para isso, é criada uma variável *input_data*, onde são passados os dados do *dataset* sem a coluna *target*, para que possa ser realizada a análise preditiva. O valor passado é transformado em *array* e, por fim, é utilizada a função *predict* para retornar a previsão da análise (Figura 13).

Figura 13 – Análise Preditiva

```
input_data = (41,1,1,120,157,0,1,182,0,0,2,0,2)

input_data_as_numpy_array= np.asarray(input_data)

input_data_reshaped = input_data_as_numpy_array.reshape(1,-1)

prediction = model.predict(input_data_reshaped)
print(prediction)


if (prediction[0] == 0):
  print('A pessoa não tem Doença cardíaca')
else:
  print('A pessoa tem Doença cardíaca')
```

```
[1]
A pessoa tem Doença cardíaca
```

Fonte: os autores

5 Resultados

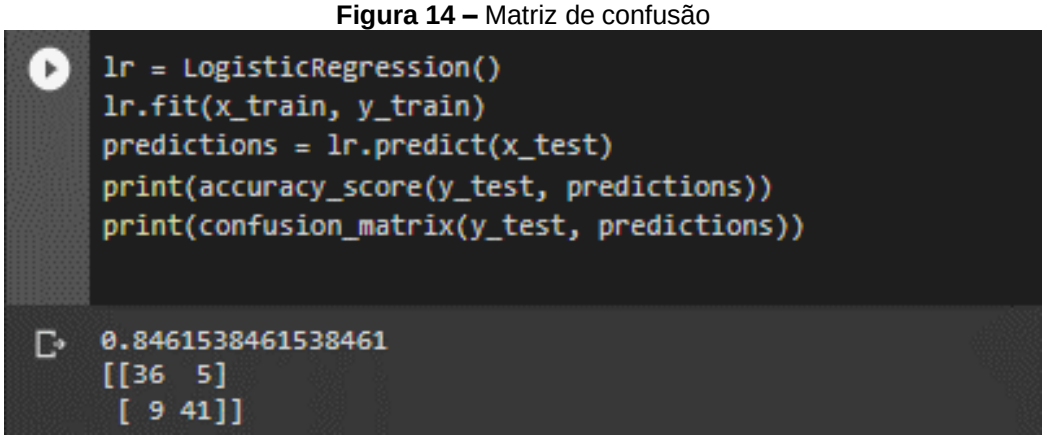
Nesta seção será apresentado o resultado dos modelos de classificação utilizados.

5.1 Resultado dos modelos preditivos

Na Figura 11 foram apresentados os quatro modelos de classificação utilizados, bem com a acurácia de cada um deles. Apenas o *LogisticRegression,* com 84,9% de acurácia, e o *Naive Bayes*, com 82,5% de acurácia, conseguiram alcançar um valor superior a 80%, considerado o limite, sendo que os outros dois ficaram abaixo: *KNeighborsClassifier,* com 77,3% de acurácia, e o *Support Vector Machines*, com 67,4% de acurácia.

O modelo de classificação escolhido foi o *LogisticRegression* por ter a maior acurácia. Para verificar a eficácia foi criado a matriz de confusão (Figura 14), que “exibe a distribuição dos registros em termos de suas classes atuais e de suas classes previstas. Isso indica a qualidade do modelo atual. Um modelo pode conter duas ou mais classes previstas.” (IBM, 2021, *online*).

Figura 14 – Matriz de confusão



Fonte: os autores

A matriz de confusão mostra que a acurácia do modelo de classificação foi de 84,6%, sendo que o conjunto de treinamento possuía 91 pacientes. O modelo fez a classificação correta de 77 pacientes, sendo 36 pacientes classificados sem problemas de DCV e eles não possuíam nenhuma, e 41 pacientes classificados com algum problema de DCV e eles realmente as possuíam. Além disso, 14 pacientes foram classificados de maneira incorreta, onde 5 pacientes que foram classificados sem nenhum tipo de DCV, mas as possuíam, e 9 pacientes sem DCV foram classificados com a doença.

6 Conclusão

A preocupação com as questões relacionadas a saúde, independentemente dos fatores genéticos ou não, sempre serão pautas de debates no mundo todo, e as doenças cardiovasculares não são uma exceção.

Esse assunto não merece apenas importância, mas aprofundamento por meio das condições fornecidas pela tecnologia, que avança ano após ano, fornecendo dados que podem mudar vidas, ou no mínimo, proporcionar uma solução que seja viável para melhorar a qualidade de vida das pessoas.

Os resultados aqui apresentados revelam padrões que podem ser previstos com as informações do indivíduo portador da doença. Assim, a utilização desse e outros dados reais, junto com os métodos apresentados, podem trazer um diagnóstico que antecipa as complicações futuras que, infelizmente, podem terminar em morte.

Para isso e para todo entendimento do estudo, foi analisado uma análise preditiva e constatado a acurácia maior que 80%, sendo os modelos de classificação, *LogisticRegression* com 84,9% e o *Naive Bayes* com 82,5%. Desta forma, conclui-se que a inovação que a análise e predição de dados trazem para área da saúde é de extrema importância para auxiliar no diagnóstico médico e qualidade de vida do paciente. Na análise preditiva deve se ter em mente que o modelo preditivo não deve ter 100% de acurácia, pois assim é visto que o mesmo não sabe analisar padrões, e sim o modelo decorou os resultados.

Assim, o artigo apresenta uma maneira de prever novos casos de doenças cardíacas. Além disso, como trabalhos futuros, propõe-se utilizar esses

métodos de análise para identificar outros tipos de doença visando auxiliar na melhora da saúde das pessoas.

Referências

BARI, Anasse; CHAOUCHI, Mohamed; JUNG, Tommy. **Predictive Analytics for Dummies**. 2. ed. Hoboken: John Wiley & Sons, Inc, 2017.

CANALTECH. **Ranking IEEE | Veja as linguagens de programação mais populares de 2022**. Disponível em: <https://canaltech.com.br/software/ranking-ieee-veja-as-linguagens-de-programacao-mais-populares-de-2022-224274/>. Acesso em: 15 out. 2022.

CARDIO, E. S. **Insuficiência Cardíaca: causas e tratamento**. sd. Disponível em: <https://seucardio.com.br/insuficiencia-cardiaca/>. Acesso em: 15 out. 2022.

CURA, Medicina Diagnóstica. **Doenças Cardiovasculares: Causa Número 1 de Mortes no Mundo.** sd. Disponível em: <https://cura.com.br/doencas-cardiovasculares-causa-numero-1-de-mortes-no-mundo/>. Acesso em: 03 ago. 2022.

CENTRO DE CARDIOLOGIA. **O que são doenças isquêmicas do coração?** 10/03/2019. Disponível em: <https://cardiologiahmt.com.br/o-que-sao-doencas-isquemicas-do-coracao/>. Acesso em: 03 ago. 2022.

COLABORATORY, Google. **Conheça o Colab.** 2021. Disponível em: <https://colab.research.google.com/>. Acesso em: 03 set. 2022.

EINSTEIN. Hospital Israelita Albert Einstein. **Doença Arterial Coronariana (DAC).** sd. Disponível em: <https://www.einstein.br/guia-doencas-sintomas/doenca-arterial-coronariana#:~:text=A%20doen%C3%A7a%20arterial%20coronariana%20(DAC)>. Acesso em: 01 ago. 2022.

EINSTEIN. Hospital Israelita Albert Einstein. **Insuficiência Cardíaca** sd. Disponível em: <https://www.einstein.br/especialidades/cardiologia/doencas-sintomas/insuficiencia-cardiaca>. Acesso em: 03 ago. 2022.

IBM. **Visualização da Matriz de Confusão.** 01/03/2021. Disponível em: <https://www.ibm.com/docs/pt-br/db2/10.5?topic=visualizer-confusion-matrix-view>. Acesso em: 5 out. 2022.

KAGGLE. **Heart Disease Prediction**. Disponível em: <https://www.kaggle.com/datasets/rishidamarla/heart-disease-prediction>. Acesso em: 20 jul. 2022.

MOHRI, Mehryar; ROSTAMIZADEH, Afshin; TALWALKAR, Ameet. **Foundations of machine learning.** MIT press, 2018.

NAÇÕES UNIDAS . **OMS revela principais causas de morte e incapacidade em todo o mundo entre 2000 e 2019**. 10/12/2020 Disponível em: <https://brasil.un.org/pt-

br/104646-oms-revela-principais-causas-de-morte-e-incapacidade-em-todo-o-mundo-entre-2000-e-2019#:~:text=O%20n%C3%BAmero%20de%20mortes%20por>. Acesso em: 7 out. 2022.

NUMPY**. NumPy.** sd. Disponível em: <https://numpy.org/>. Acesso em: 14 set. 2022.

OPAS/OMS, Organização Pan-Americana da Saúde**. Doenças Cardiovasculares.** Disponível em: <https://www.paho.org/pt/topicos/doencas-cardiovasculares>. Acesso em: 03 ago. 2022.

OPAS/OMS, Organização Pan-Americana da Saúde. **Doenças Cardiovasculares Continuam Sendo Principal Causa de Morte nas Américas. 29/09/** Disponível em: <https://www.paho.org/pt/noticias/29-9-2021-doencas-cardiovasculares-continuam-sendo-principal-causa-morte-nas-americas>. Acesso em: 03 ago. 2022.

PAIXÃO, G.M.M.; SANTOS, B.C.; ARAUJO, R.M.; RIBEIRO, M.H.; MORAES, J.L.; RIBEIRO, A.L. **Machine Learning in Medicine: Review and Applicability.** 2022

PANDAS**. Python Data Analysis Library — pandas: Python Data Analysis Library.** sd. Disponível em: <https://pandas.pydata.org/>. Acesso em: 14 set. 2022.

PANDAS. **Getting started** sd. Disponível em: <https://pandas.pydata.org/docs/getting_started/index.html#getting-started>. Acesso em: 14 set. 2022.

PYTHON. **Welcome to Python.org**. sd. Disponível em: <https://www.python.org/>. Acesso em: 10 set. 2022.

SCIKIT-LEARN. **scikit-learn: Machine learning in Python**. sd. Disponível em: <https://scikit-learn.org/stable/>. Acesso em: 14 set. 2022.

SEABORN. **seaborn: statistical data visualization**. sd. Disponível em: <https://seaborn.pydata.org/>. Acesso em: 15 set. 2022.

UCI, Machine Learning Repository. **Heart Disease Data Set.** sd. Disponível em: <https://archive.ics.uci.edu/ml/datasets/Heart+Disease>. Acesso em: 20 jul. 2022.

WORLD HEART FEDERATION. **What is CVD?** sd. Disponível em: <https://world-heart-federation.org/world-heart-day/cvd-causes-conditions/what-is-cvd/>. Acesso 29 ago. 2022