

**UNIVERSIDADE FEDERAL DE PERNAMBUCO
CENTRO DE TECNOLOGIA E GEOCIÊNCIAS
DEPARTAMENTO DE ENGENHARIA BIOMÉDICA**

Gabriel Del Cesare Barros

**Trabalho de Conclusão de Curso de Graduação
UMA APLICAÇÃO DE INTELIGÊNCIA
COMPUTACIONAL EM PHYTON PARA
APOIO AO DIAGNÓSTICO DE CÂNCER DE
MAMA**

Recife,

2020

Gabriel Del Cesare Barros

**UMA APLICAÇÃO DE INTELIGÊNCIA
COMPUTACIONAL EM PHYTON PARA APOIO AO
DIAGNÓSTICO DE CÂNCER DE MAMA**

Recife,
2020

Agradecimientos

AGRADECIMENTOS

AGRADECIMENTOS

AGRADECIMENTOS

AGRADECIMENTOS

Resumo

RESUMO

Palavras-chave: Inteligência computacional. Apoio ao diagnóstico. Script. Python. Câncer de mama.

Abstract

ABSTRACT

Key-words: Computational Intelligence. Diagnostic Support. Script. Phyton. Breast Cancer.

Lista de ilustrações

Figura 1 – Incidência mundial de câncer	11
Figura 2 – K Vizinhos Mais Próximos	14
Figura 3 – Árvore de Decisão	15
Figura 4 – Floresta Aleatória	15
Figura 5 – Máquina de Vetor de Suporte	16
Figura 6 – Naive Bayes	17
Figura 7 – Artificial Neural Network	17
Figura 8 – Validação Cruzada	18

Lista de Códigos

Código 1	Executar Script.py	18
Código 2	Sintaxe Simples	19
Código 3	Sintaxe com Parâmetro	19
Código 4	Código Final	24

Lista de tabelas

Tabela 1 – Resultado Final	19
--------------------------------------	----

Lista de abreviaturas e siglas

UFPE	Universidade Federal de Pernambuco
EUA	Estados Unidos da América
IC	Inteligência Computacional
IBM	International Business Machines Corporation
OMS	Organização Mundial de Saúde

Sumário

1	INTRODUÇÃO	10
2	O CÂNCER NO MUNDO	11
2.1	Etapas do diagnóstico de câncer	12
2.2	Importância do diagnóstico precoce	12
3	INTELIGÊNCIA COMPUTACIONAL E SAÚDE	13
3.1	Phyton	13
3.2	A base de dados	13
4	OS MÉTODOS DE ANÁLISE	14
4.1	K Vizinhos Mais Próximos	14
4.2	Árvore de Decisão	14
4.3	Floresta Aleatória	14
4.4	Máquina de Vetor de Suporte	14
4.5	Naive bayes	15
4.6	Artificial Neural Network	15
5	O SCRIPT DO CÂNCER DE MAMA	18
5.1	Validação Cruzada	18
5.2	Sintaxe	18
6	CONCLUSÃO	20
7	TRABALHOS FUTUROS	21
	REFERÊNCIAS	22
	APÊNDICE A – O CÓDIGO	24

1 Introdução

TEXTO ([Orgnização Pan-Americana de Saúde - OPAS Brasil, 2018](#)).

TEXTO ([Luiz Carlos Lobo, 2018](#)).

TEXTO ([DUA; GRAFF, 2017](#)),

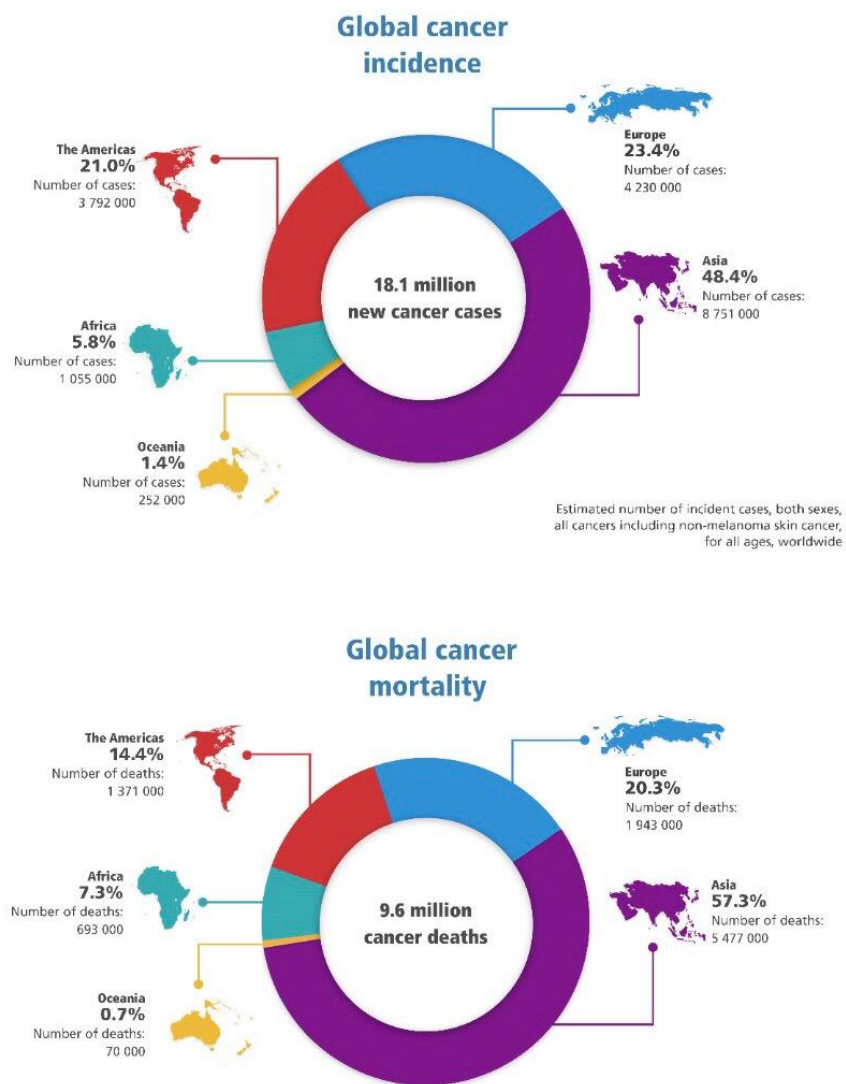
2 O Câncer no Mundo

(CÂNCER, 2018).

(BRASIL, 2018)

Referência figura 1.

Figura 1 – Incidência mundial de câncer



Fonte: <https://www.iarc.fr/wp-content/uploads/2018/09/Globocan_01.jpg> (ORGANIZATION, 2018)

2.1 Etapas do diagnóstico de câncer

(BRASIL, 2018).

(CÂNCER, 2017).

2.2 Importância do diagnóstico precoce

(BRASIL, 2020).

3 Inteligência Computacional e Saúde

3.1 **Phyton**

([FOUNDATION](#), 2020).

- Pandas ([MANAGEMENT](#), 2020).
- Scikit-learn ([COURNEPEAU](#), 2020).

3.2 **A base de dados**

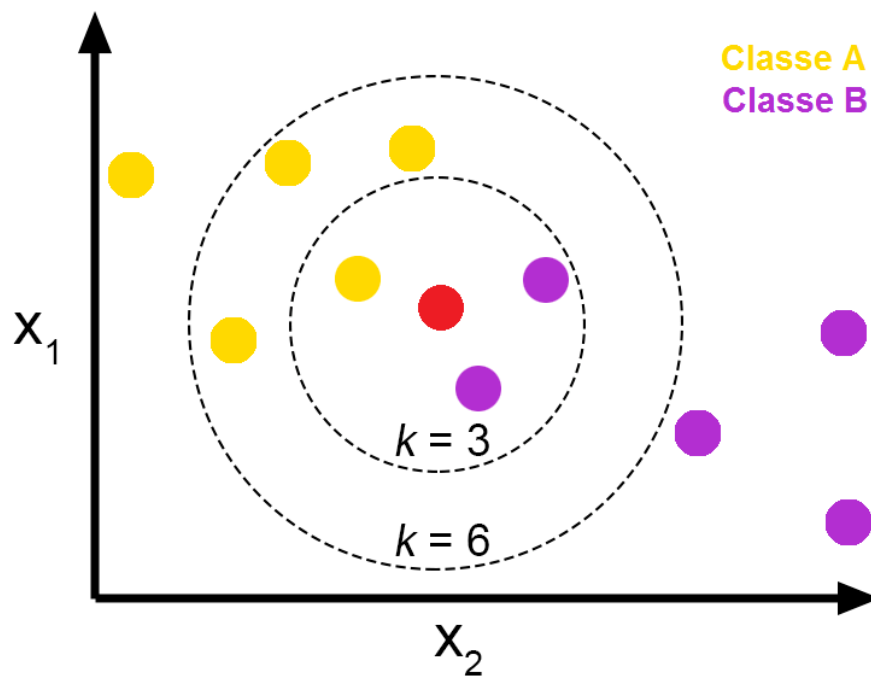
“Breast Cancer Wisconsin” (<<https://www.kaggle.com/uciml/breast-cancer-wisconsin-data>>)
([GRAFF](#), 2017).

4 Os Métodos de Análise

4.1 K Vizinhos Mais Próximos

2

Figura 2 – K Vizinhos Mais Próximos



Fonte: <https://miro.medium.com/max/1506/0*jqxx3-dJqFjXD6FA> (JOSÉ, 2018)

4.2 Árvore de Decisão

3

4.3 Floresta Aleatória

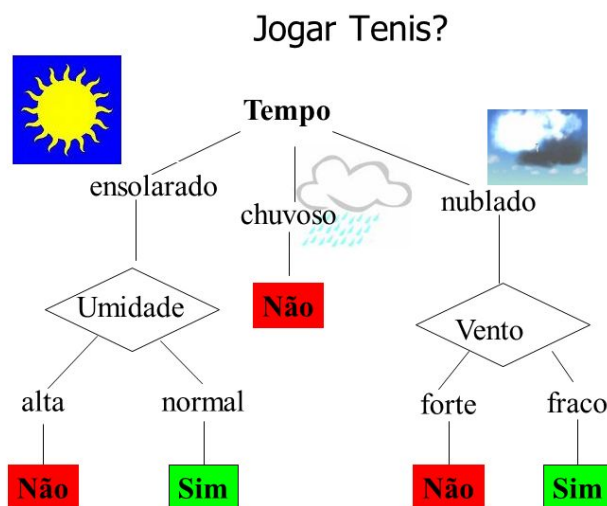
4.

4.4 Máquina de Vetor de Suporte

5

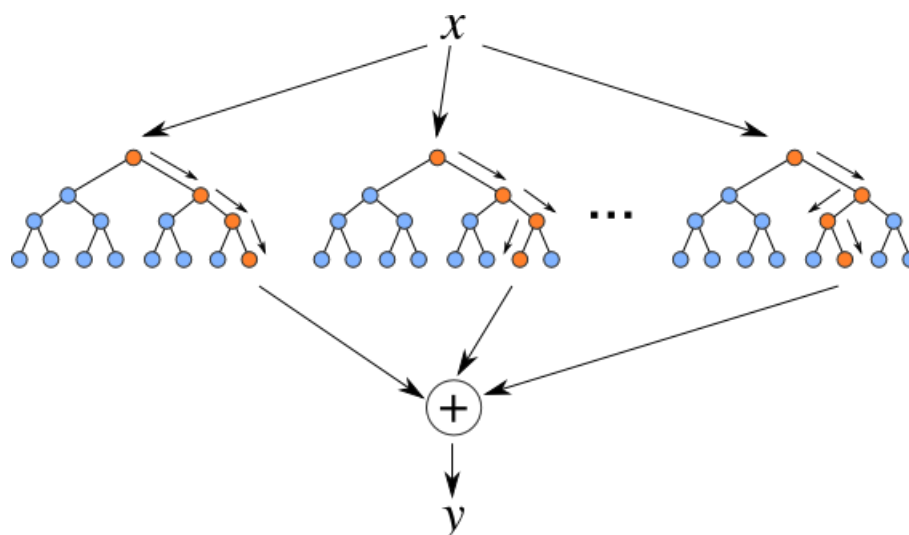
Figura 3 – Árvore de Decisão

Árvores de Decisão - Exemplo



Fonte: <<https://slideplayer.com.br/slide/358847/2/images/5/%C3%81rvores+de+Decis%C3%A3o+-+Exemplo.jpg>> (FARINHA, 2014)

Figura 4 – Floresta Aleatória

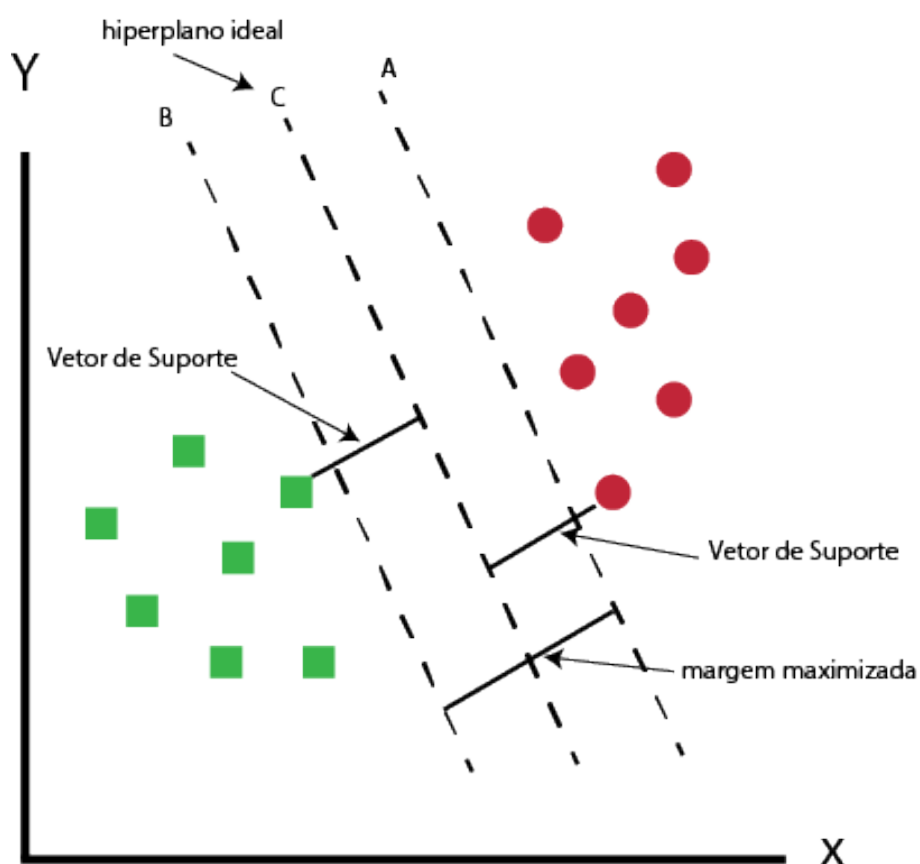


Fonte: <<https://www.paradigmigital.com/techbiz/machine-learning-dummies/>> (ZAFORAS, 2017)

4.5 Naive bayes

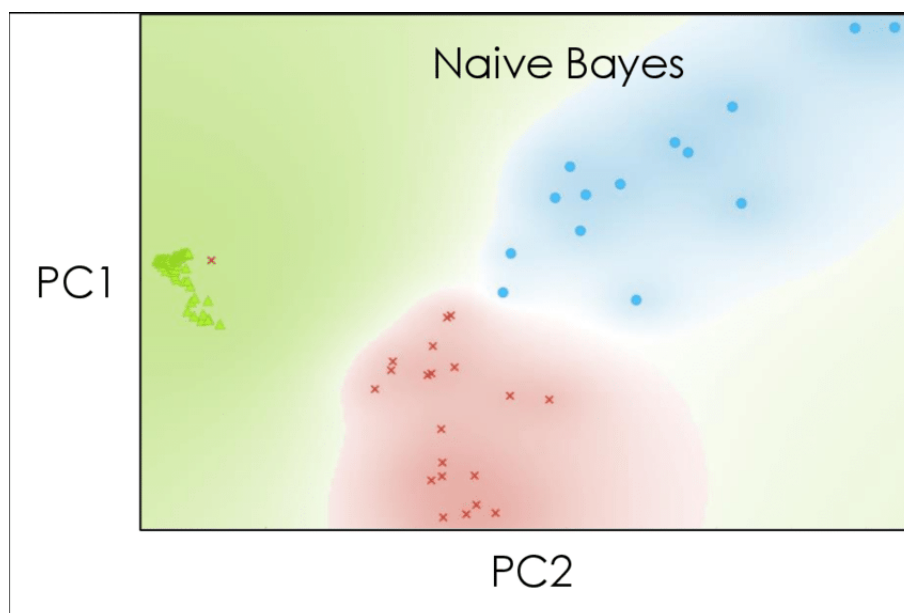
4.6 Artificial Neural Network

Figura 5 – Máquina de Vetor de Suporte



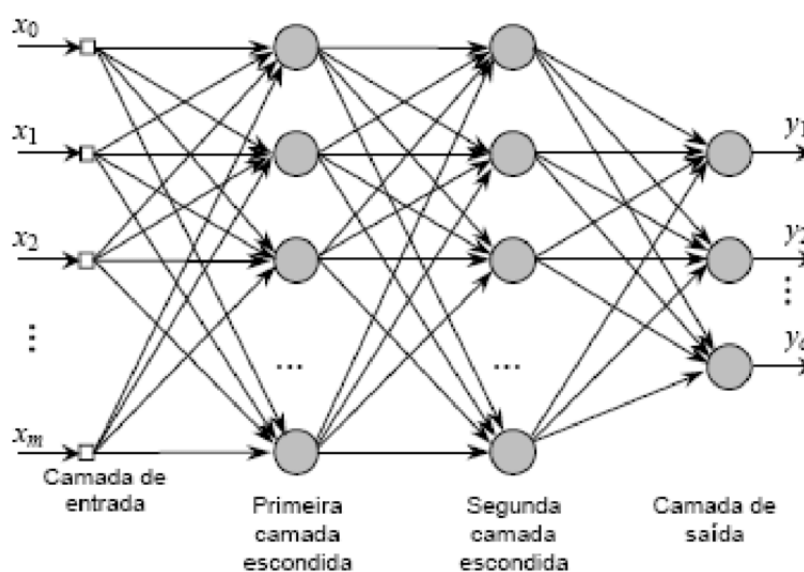
Fonte: <<https://www.codigofuente.com.br/wp-content/uploads/2019/06/SVM04.png>>
(CAVALCANTI, 2019)

Figura 6 – Naive Bayes



Fonte: https://www.researchgate.net/profile/Paolo_Dellaversana/publication/328020065/figure/fig5/AS:677213301121033@1538471641906/Naive-Bayes-classification-of-three-different-rock-types-based-on-nine-mineralogical.png (DELL' AVERSANA, 2020)

Figura 7 – Artificial Neural Network



Fonte: https://www.researchgate.net/profile/Anderson_Oliveira6/publication/240772105/figure/fig2/AS:667857415319554@1536241024122/Figura-1-Rede-Neural-Artificial-Multicamadas.png (OLIVEIRA et al., 2010)

5 O Script do Câncer de Mama

No apêndice A, seção 4, página 24. Na seção 5.2.

(FOUNDATION, 2020) A

(<https://www.kaggle.com/uciml/breast-cancer-wisconsin-data/download>) (DUA; GRAFF, 2017).

executa-se:

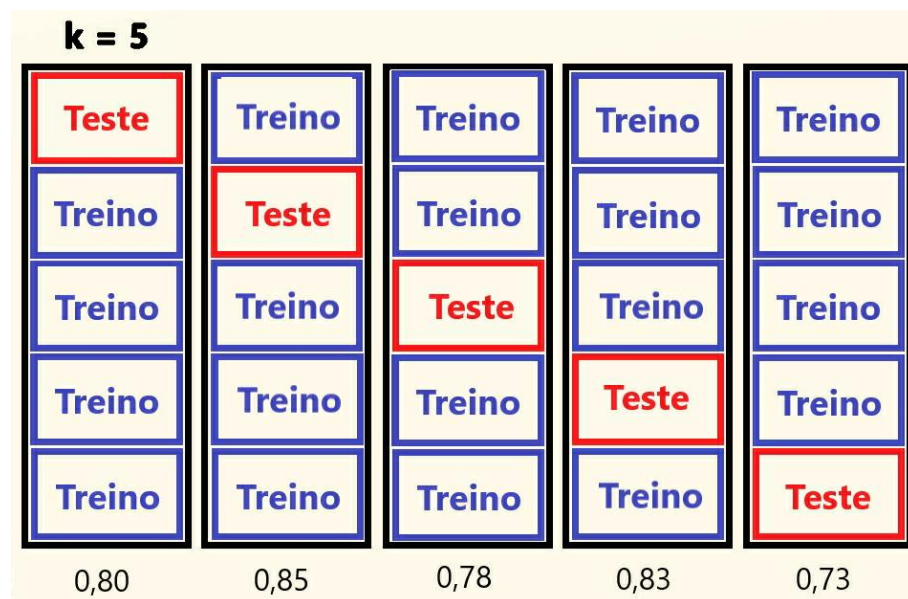
```
1 python script.py
```

Código 1 – Executar Script.py

5.1 Validação Cruzada

Na figura 8

Figura 8 – Validação Cruzada



Fonte: https://didatica.tech/wp-content/uploads/2019/10/Kfold_Resultados.png (TECH, 2020)

5.2 Sintaxe

É chamada na primeira linha do código 2.

```

1 svm = SVC(kernel='poly',degree=1)
2 scores = cross_val_score(svm, X, y, cv=10, scoring='accuracy')
3 function_print = 'SuppotVectorMachine:\t' + str(scores.mean())
4 print(function_print)
5 if scores.mean() > best_score:
6     best_score = scores.mean()
7     best_function=function_print

```

Código 2 – Sintaxe Simples

3,

```

1 max_score = 0
2 for n in range(1,10):
3     tree = DecisionTreeClassifier(max_depth=n, random_state=0)
4     scores = cross_val_score(tree, X, y, cv=10, scoring='accuracy')
5     if scores.mean() > max_score:
6         max_score = scores.mean()
7         max_n = n
8 function_print = 'DecisionTreeClassifier:\t' + str(max_score) + '\t(
    max_depth=' + str(max_n) + ')'
9 print(function_print)
10 if max_score > best_score:
11     best_score = max_score
12     best_function=function_print

```

Código 3 – Sintaxe com Parâmetro

Tabela 1.

Função	Acurácia	Parâmetro
KneighborsClassifier	0.9297619047619048	n_neighbors = 8
DecisionTreeClassifier	0.9280701754385964	max_depth = 5
RandomForestClassifier	0.9649122807017543	max_depth = 80
SuppotVectorMachine	0.9051065162907269	
GaussianNB	0.9367794486215537	
MLPClassifier	0.8963032581453634	
Melhor Função		
RandomForestClassifier:	0.9649122807017543	max_depth = 80

Tabela 1 – Resultado Final

6 Conclusão

CONCLUSÃO

7 Trabalhos Futuros

([GOOGLE, 2020](#))

([MOZILLA, 2020](#))

Referências

BRASIL, M. da S. *Atlas do Câncer Relacionado ao Trabalho no Brasil*. 1. ed. Brasília - DF, 2018. Citado na página 12.

BRASIL, M. da S. *Diagnóstico*. 2020. Disponível em: <<http://www.saude.gov.br/atencao-especializada-e-hospitalar/especialidades/oncologia/diagnostico>>. Acesso em: 21/09/2019. Citado na página 12.

BRASIL, M. de O. C. *Publicado novo relatório sobre dados mundiais de incidência e mortalidade por câncer*. 2018. Disponível em: <<https://mocbrasil.com/blog/noticias/publicado-novo-relatorio-sobre-dados-mundiais-de-incidencia-e-mortalidade-por-cancer/>>. Acesso em: 20/09/2019. Citado na página 11.

CAVALCANTI, T. *Support Vector Machine ou máquina de vetores de suporte*. 2019. Disponível em: <<https://www.codigofuente.com.br/wp-content/uploads/2019/06/SVM04.png>>. Acesso em: 30/04/2020. Citado na página 16.

COURNEPEAU, D. *Machine Learning in Python*. 2020. Disponível em: <<https://scikit-learn.org/stable/>>. Acesso em: 26/09/2019. Citado na página 13.

CÂNCER, I. N. D. *O que é câncer?* 2018. Disponível em: <<https://www.inca.gov.br/o-que-e-cancer/>>. Acesso em: 10/09/2019. Citado na página 11.

CÂNCER, I. V. o. *É comum a imunidade cair durante o tratamento oncológico?* 2017. Disponível em: <<https://www.vencerocancer.org.br/dia-a-dia-do-paciente/efeitos-colaterais/e-comum-a-imunidade-cair-durante-o-tratamento-oncologico/>>. Acesso em: 16/09/2019. Citado na página 12.

DELL'AVERSANA, P. *Naïve Bayes classification of three different rock types based on nine mineralogical*. 2020. Disponível em: <https://www.researchgate.net/profile/Paolo_Dellaversana/publication/328020065/figure/fig5/AS:677213301121033@1538471641906/Naive-Bayes-classification-of-three-different-rock-types-based-on-nine-mineralogical.png>. Acesso em: 30/04/2020. Citado na página 17.

DUA, D.; GRAFF, C. *UCI Machine Learning Repository*. 2017. Disponível em: <<http://archive.ics.uci.edu/ml>>. Acesso em: 25/09/2019. Citado 3 vezes nas páginas 10, 13 e 18.

FARINHA, R. *Inteligência Artificial*. 2014. Disponível em: <<https://slideplayer.com.br/slide/358847/>>. Acesso em: 30/09/2019. Citado na página 15.

FOUNDATION, P. S. *Python*. 2020. Disponível em: <<https://www.python.org/>>. Acesso em: 25/09/2019. Citado 2 vezes nas páginas 13 e 18.

GOOGLE. *Go*. 2020. Disponível em: <<https://golang.org/>>. Acesso em: 04/10/2019. Citado na página 21.

JOSÉ, I. *KNN (K-Nearest Neighbors)*. 2018. Disponível em: <<https://medium.com/brasil-ai/knn-k-nearest-neighbors-1-e140c82e9c4e>>. Acesso em: 17/09/2019. Citado na página 14.

- Luiz Carlos Lobo. *Inteligência Artificial e Medicina*. 2018. Disponível em: <http://www.scielo.br/scielo.php?script=sci_arttext&pid=S0100-55022017000200185&lng=pt&tlng=pt>. Acesso em: 20/08/2019. Citado na página 10.
- MANAGEMENT, A. C. *Python Data Analysis Library*. 2020. Disponível em: <<https://pandas.pydata.org/>>. Acesso em: 14/05/2020. Citado na página 13.
- MOZILLA. *JavaScript*. 2020. Disponível em: <<https://developer.mozilla.org/pt-BR/docs/Glossario/JavaScript>>. Acesso em: 04/10/2019. Citado na página 21.
- OLIVEIRA, A. et al. Aplicação de redes neurais artificiais na previsão da produção de álcool. *Ciencia E Agrotecnologia - CIENC AGROTEC*, v. 34, 04 2010. Citado na página 17.
- ORGANIZATION, W. H. *Globocan 2018 Latest global cancer data*. 2018. Disponível em: <<https://www.iarc.fr/infographics/globocan-2018-latest-global-cancer-data/>>. Acesso em: 08/10/2019. Citado na página 11.
- Organização Pan-Americana de Saúde - OPAS Brasil. *Organização Mundial da Saúde divulga novas estatísticas mundiais de saúde*. 2018. Disponível em: <https://www.paho.org/bra/index.php?option=com_content&view=article&id=5676:organizacao-mundial-da-saude-divulga-novas-estatisticas-mundiais-de-saude&Itemid=843>. Acesso em: 08/08/2019. Citado na página 10.
- TECH, D. *O pacote Caret – linguagem R*. 2020. Disponível em: <<https://didatica.tech/o-pacote-caret-linguagem-r/>>. Acesso em: 08/05/2019. Citado na página 18.
- ZAFORAS, M. *Machine Learning para dummies*. 2017. Disponível em: <<https://www.paradigmadigital.com/techbiz/machine-learning-dummies/>>. Acesso em: 30/04/2020. Citado na página 15.

APÊNDICE A – O Código

```
#!/bin/python
# Importar Bibliotecas
import pandas as pd
from sklearn.neighbors import KNeighborsClassifier
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import cross_val_score
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.naive_bayes import GaussianNB
from sklearn.neural_network import MLPClassifier
from sklearn.svm import SVC
import warnings
warnings.filterwarnings('ignore')

# Banco de Dados
cancer = pd.read_csv('./data.csv', index_col=0)

# Configurar Variaveis
diag = {'M':0, 'B':1}
cancer.diagnosis = [diag[item] for item in cancer.diagnosis]
X = cancer[cancer.columns[1:31]].to_numpy()
y = cancer[['diagnosis']].to_numpy()

# Funcoes
best_score = 0
max_score = 0
for n in range(1,10):
    knn = KNeighborsClassifier(n_neighbors=n, weights='uniform')
    scores = cross_val_score(knn, X, y, cv=10, scoring='accuracy')
    if scores.mean() > max_score:
        max_score = scores.mean()
        max_n = n
function_print = 'KneighborsClassifier:\t' + str(max_score) + '\t(
    n_neighbors=' + str(max_n) + ')'
print(function_print)
if max_score > best_score:
    best_score = max_score
    best_function=function_print

max_score = 0
for n in range(1,10):
    tree = DecisionTreeClassifier(max_depth=n, random_state=0)
    scores = cross_val_score(tree, X, y, cv=10, scoring='accuracy')
```

```

    if scores.mean() > max_score:
        max_score = scores.mean()
        max_n = n
function_print = 'DecisionTreeClassifier:\t' + str(max_score) + '\t(
    max_depth=' + str(max_n) + ' )'
print(function_print)
if max_score > best_score:
    best_score = max_score
    best_function=function_print

max_score = 0
for n in range(1,10):
    forest = RandomForestClassifier(n_estimators= n*10, random_state=0)
    scores = cross_val_score(forest, X, y, cv=10, scoring='accuracy')
    if scores.mean() > max_score:
        max_score = scores.mean()
        max_n = n*10
function_print = 'RandomForestClassifier:\t' + str(max_score) + '\t(
    max_depth='+ str(max_n) + ' )'
print(function_print)
if max_score > best_score:
    best_score = max_score
    best_function=function_print

svm = SVC(kernel='poly',degree=1)
scores = cross_val_score(svm, X, y, cv=10, scoring='accuracy')
function_print = 'SuppotVectorMachine:\t' + str(scores.mean())
print(function_print)
if scores.mean() > best_score:
    best_score = scores.mean()
    best_function=function_print

gnb = GaussianNB()
scores = cross_val_score(gnb, X, y, cv=10, scoring='accuracy')
function_print = 'GaussianNB:\t\t' + str(scores.mean())
print(function_print)
if scores.mean() > best_score:
    best_score = scores.mean()
    best_function=function_print

mlp = MLPClassifier(solver='adam', alpha=0.0001, hidden_layer_sizes
    =(10,20,40),
    random_state=42, learning_rate='constant', learning_rate_init=0.01,
    max_iter=100,
    activation='logistic', momentum=0.9, tol=0.0001)
scores = cross_val_score(mlp, X, y, cv=10, scoring='accuracy')
function_print = 'MLPClassifier:\t\t' + str(scores.mean())

```

```
print(function_print)
if scores.mean() > best_score:
    best_score = scores.mean()
    best_function=function_print

# Resultados
print("\nMelhor Funcao:")
print(best_function)
```

Código 4 – Código Final