

**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

Agradecimentos

Em primeiro lugar agradeço aos meus pais, Soraya e André por todo cuidado e suporte dados ao longo da minha vida.

Agradeço à Ingridy Souza, por seu apoio e companheirismo.

À Vinicius Galvão pelo auxílio com seu conhecimento na área.

Aos professores do departamento de Engenharia Biomédica da Universidade Federal de Pernambuco, em especial Ascendino Silva, por toda sua atenção e disponibilidade e Wellington Pinheiro, por sua orientação nesta monografia.

Resumo

Tendo em vista que a necessidade de um diagnóstico precoce de câncer é primordial para um tratamento efetivo, e consequentemente a cura da neoplasia, pesquisa-se sobre o desenvolvimento de uma Inteligência Computacional escrita na linguagem Python, com o intuito de criar um mecanismo de apoio ao diagnóstico médico. Para tanto, é necessário o desenvolvimento de um script com funções de aprendizado de máquina, aplicação do banco de dados “Diagnostic Dataset” da Breast Cancer Wisconsin (1992). Além da utilização de algoritmos computacionais, que possibilitarão a análise por meio da ferramenta, para que assim, ela informe se determinado conjunto de atributos tumorais tem alta probabilidade de ser maligno. Diante disso, após sua execução, verifica-se que o script apresenta, uma acurácia de 96%, o que impõe a constatação de que é possível o desenvolvimento de um script que auxilie o diagnóstico médico com uma taxa de exatidão relevante.

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

Abstract

In view of the necessity for early diagnosis of cancer is primordial for effective treatment and, consequently, the cure of cancer, research on the development of a Computational Intelligence written in Phyton language, in order to create a support mechanism to medical diagnosis. For this, it's necessary to develop a script with machine learning functions, application of Breast Cancer Wisconsi's "Diagnostic Dataset" (1992). In addition to the use of computational algorithms, which will enable analysis through the tool, so that it can inform if a certain set of tumor attributes is likely to be malignant. Given this, after its execution, it appears that the script has an accuracy of 96%, what imposes the realization that it is possible to develop a script that helps the medical diagnosis with a rate of relevant accuracy.

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

Lista de ilustrações

Figura 1 – Incidência mundial de câncer	13
Figura 2 – K Vizinhos Mais Próximos	18
Figura 3 – Árvore de Decisão	19
Figura 4 – Floresta Aleatória	20
Figura 5 – Máquina de Vetor de Suporte	21
Figura 6 – Naive Bayes	22
Figura 7 – Artificial Neural Network	22
Figura 8 – Validação Cruzada	24

Lista de Códigos

Código 1	Executar Script.py	23
Código 2	Sintaxe Simples	24
Código 3	Sintaxe com Parâmetro	25
Código 4	Código Final	30

Lista de tabelas

Tabela 1 – Resultado Final	25
--------------------------------------	----

Lista de abreviaturas e siglas

UFPE	Universidade Federal de Pernambuco
EUA	Estados Unidos da América
IACR	International Association of cancer Registries
IC	Inteligência Computacional
IBM	International Business Machines Corporation
IHQ	Imunohistoquímicos
INCA	Instituto Nacional de Câncer
OMS	Organização Mundial de Saúde
PAAF	Punção Aspirativa por Agulha Fina
PET-CT	<i>Positron Emission Tomography - Computed Tomography</i>

Sumário

1	INTRODUÇÃO	10
2	O CÂNCER NO MUNDO	12
2.1	Etapas do diagnóstico de câncer	13
2.2	Importância do diagnóstico precoce	14
3	INTELIGÊNCIA COMPUTACIONAL E SAÚDE	15
3.1	Python	15
3.2	A base de dados	16
4	OS MÉTODOS DE ANÁLISE	18
4.1	K Vizinhos Mais Próximos	18
4.2	Árvore de Decisão	18
4.3	Floresta Aleatória	19
4.4	Máquina de Vetor de Suporte	19
4.5	Naive bayes	20
4.6	Artificial Neural Network	20
5	O SCRIPT DO CÂNCER DE MAMA	23
5.1	Validação Cruzada	23
5.2	Sintaxe	24
6	CONCLUSÃO	26
7	TRABALHOS FUTUROS	27
	REFERÊNCIAS	28
	APÊNDICE A – O CÓDIGO	30

1 Introdução

As neoplasias malignas, também conhecidas como “Câncer” é um conjunto de patologias que de acordo com a OMS, é a segunda maior causa de óbitos no mundo, responsável por 9,6 milhões de mortes, apenas no ano de 2018 ([Organização Pan-Americana de Saúde - OPAS Brasil, 2018](#)).

Tal problema a nível global atrai a atenção para a importância de um diagnóstico precoce, o que aumenta a probabilidade de um tratamento eficaz, pois sabe-se que os pacientes que são diagnosticados em estádios tardios, podem não responder aos tratamentos curativos.

Para isso, a tecnologia vem sendo utilizada a fim de aprimorar e auxiliar as equipes multidisciplinares de saúde, principalmente nos diagnósticos médicos.

É é nesta realidade em que uma ferramenta tecnológica vem sendo cada vez mais estudada e aplicada: A Inteligência Computacional.

Esse ramo da ciência da computação aplicado na área de saúde consegue analisar e definir as variáveis de diversas doenças. De acordo com Lobo, “A Inteligência Artificial processa esses dados por meio de algoritmos, que tendem a se aperfeiçoar pelo seu próprio funcionamento, definindo pelo termo em inglês Self-learning e a propor hipóteses diagnósticas cada vez mais precisas.” ([Luiz Carlos Lobo, 2018](#)).

Assim, tendo conhecimento de tais inventos, viu-se a oportunidade da realização de um mecanismo, que facilitasse a detecção do câncer, auxiliando os profissionais médicos: Uma Inteligência computacional, idealizada em forma de Script, escalável, escrita na linguagem Python que, a partir de um banco de dados, aprenda como diferenciar se um tumor é maligno ou benigno, motivando assim a realização desse trabalho, que visa atingir, principalmente, os profissionais de saúde, iniciantes na área de aprendizagem de máquina.

Como base, para ensinar a inteligência computacional a analisar os atributos que determinam se um tumor é maligno ou benigno, será utilizada a base de dados da “Breast Cancer Wisconsin” intitulada como “Diagnostic Dataset” ([DUA; GRAFF, 2017](#)), uma base de dados numérica, o que significa que os atributos já foram extraídos das imagens, portanto não será necessário pré-processamento, o que facilita o aprendizado. As informações contidas na base, servirão de referência para, quando for suposto um caso de tumor ao Script ele consiga calcular a probabilidade, do mesmo ser de caráter maligno.

Essa análise será feita utilizando-se de seis algoritmos de aprendizagem de máquina, que são: “K Vizinhos Mais Próximos; Árvore de Decisão; Floresta Aleatória; Máquina de Vetor de Suporte; Naive Bayes e Rede Neural Artificial”, eles auxiliarão a inteligência

computacional na tomada de decisão do caso proposto.

O trabalho estrutura-se de forma a definir as estatísticas do câncer no mundo e seus impactos na saúde mundial, para retratar a necessidade de tratamentos e diagnósticos eficazes.

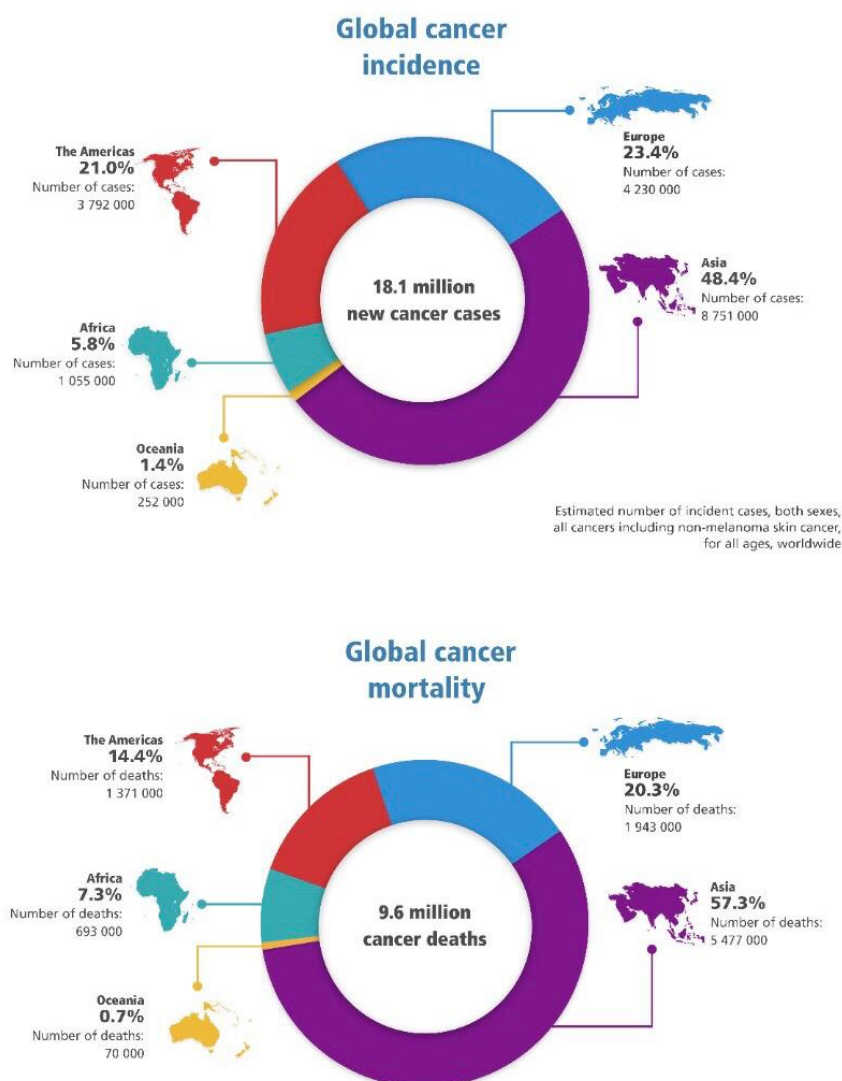
Posteriormente, explica-se sobre os avanços das aplicações da IC na área, com seus respectivos métodos de análise; finalizando com a exibição do código desenvolvido e suas conclusões.

2 O Câncer no Mundo

Denominado, também, como neoplasia ou tumor maligno, o câncer é um conjunto de mais de 100 doenças, em que há um crescimento exacerbado de células em um determinado tecido. Essas células, por sua vez, podem se espalhar a tal ponto, que chegam a atingir outros órgãos e tecidos, fenômeno denominado de “metástase”, demonstrando o poder agressivo dessa classe de patologias ([CÂNCER, 2018](#)).

Segundo dados divulgados pela IACR, era estimado para o ano de 2018 uma incidência de 18,1 milhões de novos casos de câncer no mundo. Também foi datado, no mesmo ano, 9,6 milhões de óbitos ([BRASIL, 2018](#)), como mostrado na 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

O diagnóstico tem como definição ser o processo de identificação a partir de exames médicos, sintomas e sinais além das amostras do tecido atingido (BRASIL, 2018).

Assim, até que o Patologista, especialista médico responsável pelo laudo anatomo-patológico, faça sua resolução, há alguns processos que auxiliam na análise do problema. Dentre os exames mais solicitados, está a retirada de um fragmento do tecido suspeito, para que possa ser feita uma análise em laboratório no microscópio afim de verificar aspecto dos núcleos e a morfologia das células presentes naquela amostra; exames de sangue, da medula óssea; radiografia; ultrassonografia, ressonância magnética; tomografia computadorizada

por emissão de pósitrons (PET-TC), são exemplos de exames convencionais (CÂNCER, 2017).

É importante destacar que a solicitação desses procedimentos irá variar de acordo com a suspeita do tumor, além da necessidade de uma avaliação mais abrangente, pois há situações em que os exames convencionais não expõem a informação necessária para um diagnóstico definitivo. E em alguns casos é preciso a realização de exames Imunohistoquímicos (IHQ), com marcações especiais.

Todo esse processo é de extrema importância, pois, apenas a partir da análise desses laudos, pelo médico responsável, que ele poderá, confirmar ou descartar a hipótese da neoplasia, e em caso de confirmação, iniciar um plano de tratamento específico para o problema.

2.2 Importância do diagnóstico precoce

O processo de identificação, exame e avaliação do câncer requer vários fatores, e, portanto, requer tempo.

Porém, profissionais e órgãos da área de saúde alertam para a importância de um diagnóstico precoce, pois esse, pode elevar as chances de um prognóstico favorável ao paciente.

O Ministério da Saúde do Brasil relata a importância de um diagnóstico em estágios iniciais da seguinte maneira:

“O objetivo do diagnóstico precoce é identificar pessoas com sinais e sintomas iniciais da doença, primando pela qualidade e pela garantia da assistência em todas as etapas da linha de cuidado da doença. O diagnóstico precoce, portanto, é uma estratégia que possibilita terapias mais simples e efetivas, ao contribuir para a redução do estágio de apresentação do câncer. Assim, é importante que a população em geral e os profissionais de saúde reconheçam os sinais de alerta dos cânceres mais comuns, passíveis de melhor prognóstico se descobertos no início. A maioria dos cânceres é passível de diagnóstico precoce mediante avaliação e encaminhamento após os primeiros sinais e sintomas.” (BRASIL, 2020).

3 Inteligência Computacional e Saúde

A inteligência Computacional é uma ferramenta de computação que automatiza processos complexos, como por exemplo a identificação e categorização de imagens. E já tem muito espaço na saúde, uma área de extrema relevância à sociedade, que requer atenção e agilidade, pois cuida das diversas aplicações neste ramo, estão:

- Tratamento de doenças: Se refere a capacidade dos computadores de auxiliar no tratamento das patologias. Por exemplo o “Watson”, algoritmo desenvolvido pela IBM, que informa os vários tratamentos possíveis para cada caso, informando efeitos colaterais e o grau de risco de cada uma.
- Acurácia no resultado de exames: Existem algoritmos computacionais, atualmente, que conseguem ser mais precisos em comparação à alguns exames médicos, por exemplo um algoritmo desenvolvido na Alemanha, EUA e China, que superou especialistas em retina na identificação de diagnósticos no exame de tomografia óptica
- Correlação de sintomas: O TensorFlow, por exemplo, uma biblioteca desenvolvida pela Google, consegue identificar retinopatia diabética através de semelhança de imagens, e consegue fazer associação de sintomas com taxa de acerto semelhante aos especialistas da área.
- Recuperação de dados: A partir da memória da IC, facilita-se a recuperação dos arquivos com os dados e ficha médica dos pacientes, quando houver necessidade.
- Alertas de emergência: Pode-se criar alertas sobre mudanças no quadro dos pacientes, os enviando ao profissional responsável, sendo de extrema importância principalmente em emergências.

Para o desenvolvimento da Inteligência Computacional, por meio do Script, que auxiliará na avaliação médica, foram utilizadas as seguintes ferramentas:

3.1 Phyton

Python é uma linguagem de programação, no ranking das mais utilizadas do mundo, ela foi criada por Guido Van Rossum, com o objetivo de proporcionar produtividade e legibilidade na realização de projetos computacionais ([FOUNDATION, 2020](#)).

A sua sintaxe é simples, sendo comparada até a um pseudocódigo executável. Usa a chamada “Identação”, que é um recuo no código, em relação a sua margem, para marcar blocos. Também é uma linguagem de baixo uso de caracteres especiais e palavras chaves.

Apesar de ser uma linguagem comumente utilizada para IC, ainda não é simples de encontrar artigos e documentos que instruem para a realização de um programa básico, a fim de iniciar os primeiros passos na área

Assim, será utilizado no Script de Câncer de mama as seguintes ferramentas básicas e de fácil acesso:

- Pandas - é uma biblioteca do Python, que fornece ferramentas de análise e estruturas de dados de alta performance e de simples utilização (MANAGEMENT, 2020).
- Scikit-learn - é outra biblioteca do Python que inclui alguns algoritmos de classificação, regressão e agrupamento (COURNEPEAU, 2020).

3.2 A base de dados

Para comprovar a correlação do diagnóstico de câncer a partir dos parâmetros dos exames, e ensinar a inteligência computacional a tentar descobrir o provável diagnóstico, foi-se utilizado uma base de dados de autoria da “Breast Cancer Wisconsin” chamada por eles como “Diagnostic Dataset”, presente no site: (<<https://www.kaggle.com/uciml/breast-cancer-wisconsin-data>>)(DUA; GRAFF, 2017).

É um conjunto de registros de resultados de exames utilizados para detecção do Câncer de mama, comumente utilizada para aprendizado e aplicações reais. Ela foi criada em 1995 por Dr. William H. Wolberg, W. Nick Street e Olvi L. Mangasarian na universidade de Wisconsin.

Possui 569 instâncias, que estão correlacionadas com as pessoas que participaram da coleta. E 32 tipos de atributos que são os dados coletados de cada análise. Entre os atributos está o diagnóstico final dado pelo médico informando se o tumor é maligno ou benigno.

Os dados dessa base foram obtidos através de diversos procedimentos de diagnóstico médico. Dentre eles, há a imagem digitalizada de um aspirado por agulha fina (PAAF) de um tecido mamário que tem como função determinar os aspectos dos núcleos celulares presentes no tecido.

Essa base é importante para o desenvolvimento do Script do Câncer de mama pois ela permite analisar os atributos coletados para ensinar a inteligência computacional a tentar descobrir o provável diagnóstico.

Os atributos coletados na base de dados determinam as características do tumor maligno, como:

Raio, textura, perímetro, área, suavidade, compacidade, concavidade e os pontos côncavos presentes; simetria, além da dimensão fractal. Para cada um desses parâmetros foram analisados a média, erro padrão e o “pior” ou mais largo (média dos 3 maiores valores).

4 Os Métodos de Análise

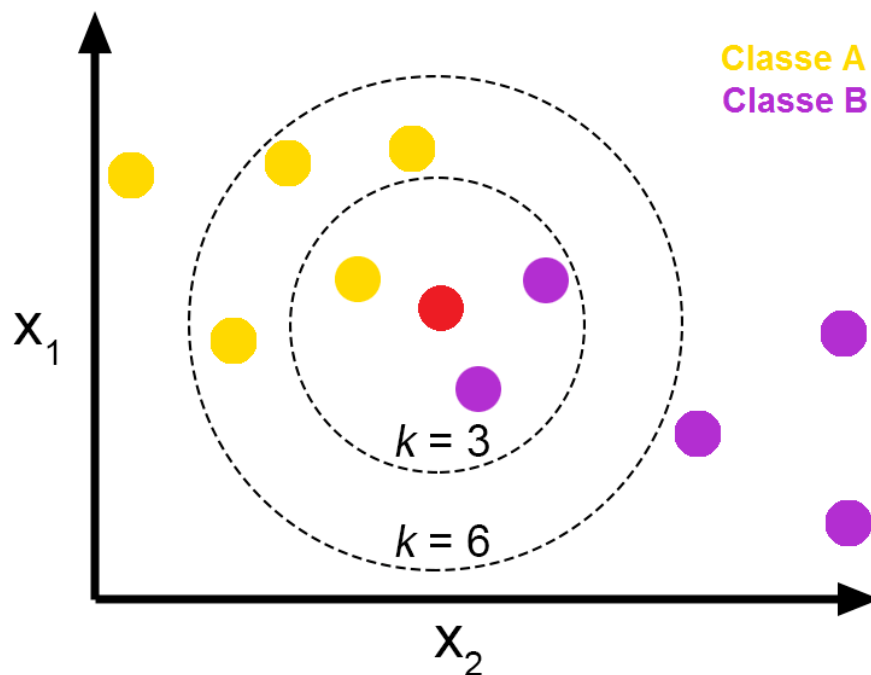
4.1 K Vizinhos Mais Próximos

Existem vários métodos de análise de algoritmos, um dos mais simples e conhecidos é o “K Vizinhos Mais Próximos”, utilizado no Script, que funciona da seguinte forma:

Observando a figura 2, observa-se um gráfico com um conjunto de bolas amarelas e roxas; também é possível perceber uma bola desconhecida, destacada na cor vermelha. Para calcular se ela é roxa ou amarela, o algoritmo do KNN determinará a sua cor, de acordo com as N bolas mais próximas.

Por exemplo, caso N seja igual a 3, a bola desconhecida será classificada como roxa e no caso de N igual à 6 será classificada como amarela.

Figura 2 – K Vizinhos Mais Próximos



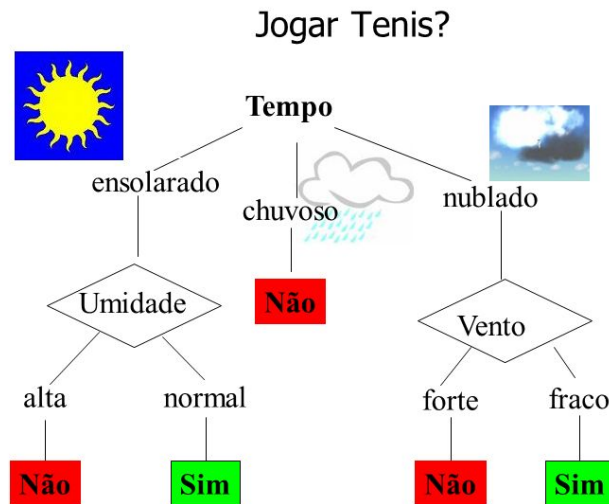
Fonte: <https://miro.medium.com/max/1506/0*jqxx3-dJqFjXD6FA> (JOSÉ, 2018)

4.2 Árvore de Decisão

Define um novo elemento com base em rotinas pré-estabelecidas. Por exemplo, se há um banco de dados de pessoas que foram jogar tênis com base na situação do clima, pode-se chegar a seguinte árvore de decisão da figura 3.

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)

E então quando alguém perguntar se haverá jogo de tênis no dia, ela responderá de acordo com o clima, seguindo a árvore de decisão.

4.3 Floresta Aleatória

Algoritmo que cria várias árvores de decisões, definido anteriormente, e as combinam para obter uma predição com maior acurácia e mais estável, assim como na figura 4.

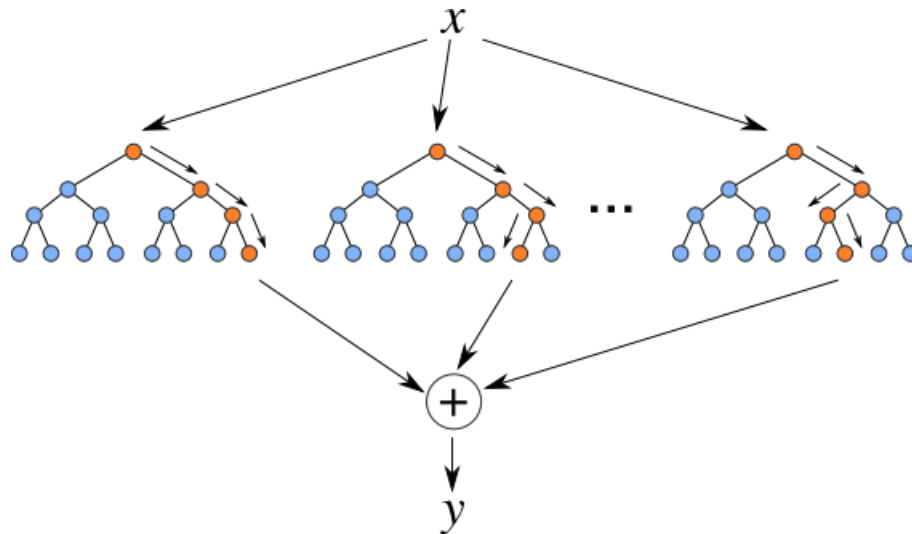
4.4 Máquina de Vetor de Suporte

O classificador categoriza, com base em uma divisão dos dados.

Por exemplo, na figura 5, há quadrados e círculos de cores vermelhas e azuis, respectivamente. Pode-se dividi-las com qualquer uma das retas do quadrante esquerdo. Porém, como mostrado no quadrante direito, há uma única reta em que a sua distância em relação a figura azul mais próxima é igual a distância da figura vermelha mais próxima, logo se encontra a melhor reta para separá-las.

Em casos mais complexos utilizam-se hiperplanos.

Figura 4 – Floresta Aleatória



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

4.5 Naive bayes

O classificador Naive Bayes é baseado no Teorema de Bayes. Ele desconsidera a correlação entre as variáveis, e por isso é chamado de “ingênuo”.

Um exemplo de sua utilização seria para considerar se um fruto qualquer é uma maçã, assim, analisa-se se ele é vermelho, redondo e de aproximadamente 3 polegadas (Características da fruta colocada em exemplo). De acordo com o Teorema de Bayes, cada um desses fatores contribui independentemente para a probabilidade de que este fruto seja realmente uma maçã.

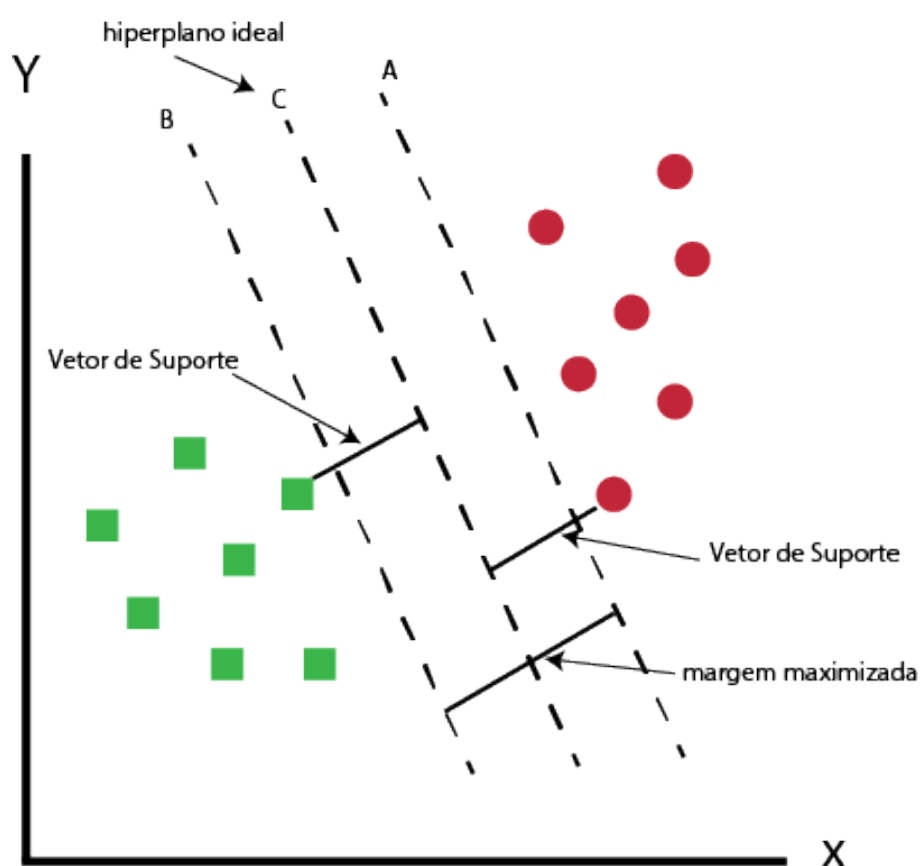
Quando se lida com muitos dados, sendo eles todos independentes, assume-se que a distribuição ocorre de acordo com uma gaussiana simples, por isso usa-se o classificador GauissanNB para calculá-lo.

4.6 Artificial Neural Network

A chamada, “Rede neural artificial” é um classificador que se baseia no sistema de redes neurais que utiliza o mecanismo de aprendizagem de pesos sinápticos de tal forma que a unidade de saída produza as respostas corretas para cada exemplo, realizando atualizações de forma dinâmica e interativa até chegar aos pesos corretos.

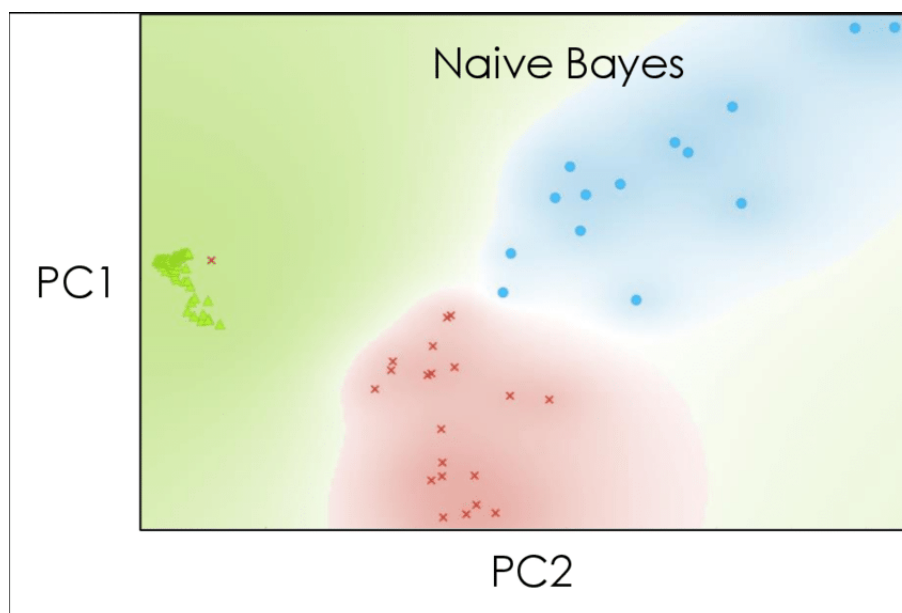
É a mais parecida, com o sistema que ocorre no cérebro.

Figura 5 – Máquina de Vetor de Suporte



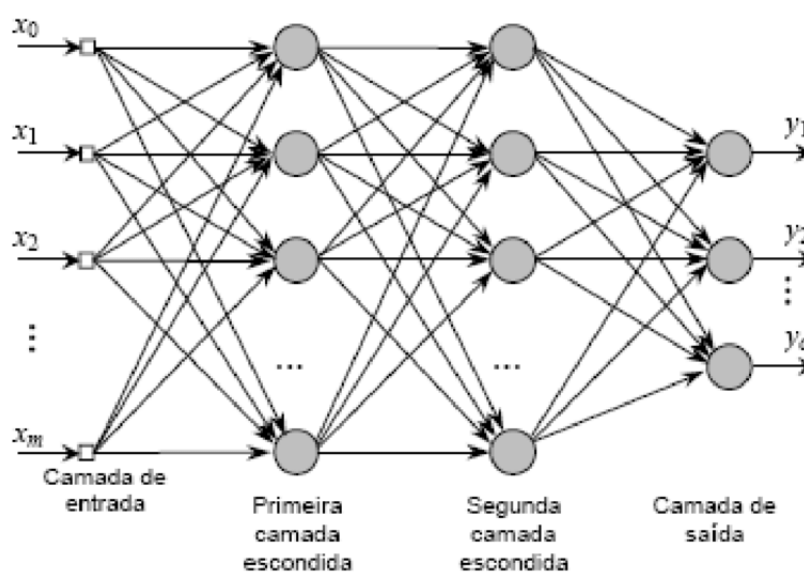
Fonte: <<https://www.codigofuente.com.br/wp-content/uploads/2019/06/SVM04.png>>
(CAVALCANTI, 2019)

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

Não foi exposto aqui o código final inteiro, por ser bastante extenso, encontra-se ao final do trabalho no apêndice A, seção 4, página 30, para quem se interessar, ler melhor.

As funções que se referem ao aprendizado de máquina são a parte principal do código e onde pode haver dúvidas quanto a sintaxe. Portanto será explicado na seção 5.2.

Para executar o script, precisa ter o python (FOUNDATION, 2020) instalado, e então salvar o conteúdo do código no apêndice A em um arquivo, por exemplo “script.py”, e colocar na mesma pasta do arquivo com o banco de dados (<<https://www.kaggle.com/uciml/breast-cancer-wisconsin-data/download>>) (DUA; GRAFF, 2017). Em seguida, no local onde estão os arquivos executa-se:

```
1 python script.py
```

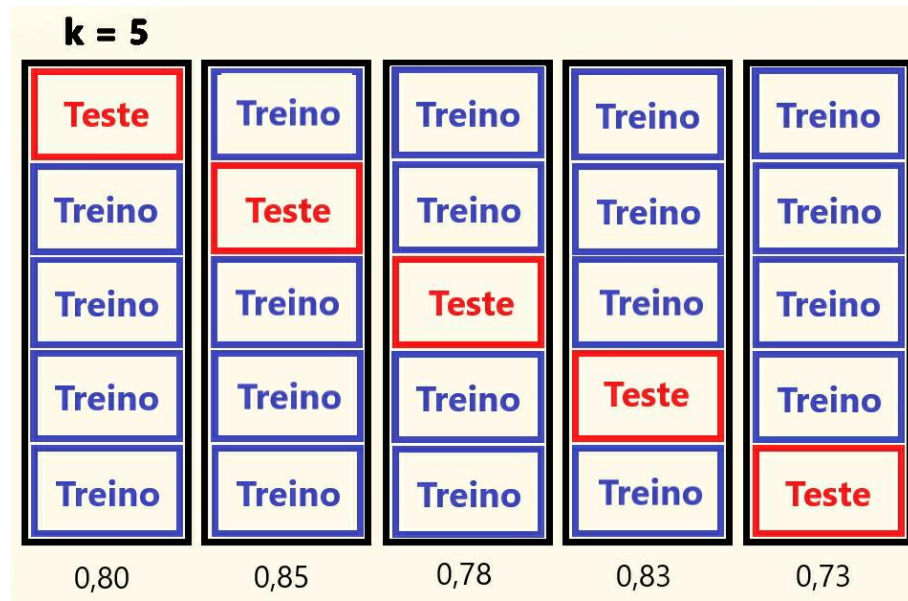
Código 1 – Executar Script.py

5.1 Validação Cruzada

Antes de ir diretamente ao código, é necessário a compreensão da validação cruzada. A validação cruzada é um método de aprendizado de máquina em que utiliza-se uma porcentagem para teste e outra para treinamento, comumente 10% para testes e 90% para treinamento. Desta forma o script treina de diferentes formas melhorando sua confiabilidade.

Por exemplo, na figura 8, a base de dados foi dividida em 6, e em cada iteração usa-se uma parte diferente como teste, e o restante para o treinamento.

Figura 8 – Validação Cruzada



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

5.2 Sintaxe

Primeiramente será explicada a função “Máquina de Vetor de Suporte”, ela é chamada na primeira linha do código 2.

Na segunda linha é chamada a validação cruzada *cross_val_score* que salva na variável *scores* um vetor com a acurácia das 10 iterações. Na terceira e quarta linha, informa na tela a média dos valores das acurácias. A quinta, sexta e sétima servem para ao final do script, informar qual foi a função com o melhor desempenho.

```

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

Por último será mostrada a função “Árvore de decisão” no código 3, pois ela possui um parâmetro *max_depth* = *n* que pode variar, e deseja-se saber para qual *n* ela consegue a melhor acurácia.

Para isso o *n* será variado de 1 a 9, como mostra as linhas 1 e 2. As linhas 5, 6

e 7 salvam o n com a maior acurácia juntamente com o resultado. O restante é igual ao código 2.

```

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('
9                 max_depth=' + str(max_n) + '),'
10 print(function_print)
11 if max_score > best_score:
12     best_score = max_score
13     best_function=function_print

```

Código 3 – Sintaxe com Parâmetro

Assim o script consegue definir a melhor função e parâmetro a partir da acurácia, em determinar se o tumor é maligno ou benigno.

E sua execução resultou nos dados encontrados na 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

Observa-se pelos resultados na tabela que com esse código simples, consegue-se uma acurácia de 96% utilizando o classificador “Floresta Aleatória” com $max_depth = 80$.

6 Conclusão

O desenvolvimento do presente trabalho possibilitou uma análise através do realização de um script simples de Inteligência computacional na área da saúde, que visa permitir um apoio na tomada de decisão médica através de atributos morfológicos de tumores de mama, contribuindo para uma reflexão da necessidade de criação de atalhos que permitam um rápido diagnóstico, diante das estatísticas alarmantes de mortalidade de câncer no mundo.

Com a execução da versão final do código, foi possível atingir o objetivo principal, onde, a partir das seis funções aplicadas no código, que permitiram o aprendizado da Inteligência artificial, por meio da base de dados das neoplasias, o script coletava as características inseridas para análise e lançava, em porcentagem, a probabilidade de malignidade, chegando ao resultado de 96% de acurácia utilizando o classificador “Floresta Aleatória” com *max_depth* de 80.

Apesar de existir outros fatores relevantes além da acurácia, percebe-se que a intenção de realizar uma ferramenta com um valor probabilístico razoável para servir de apoio na detecção de câncer foi alcançada, considerando que o diagnóstico final é, necessariamente de responsabilidade médica, pois sabe-se que a análise tumoral envolve outros aspectos, não tangíveis no script.

Tendo em vista a importância do tema abordado, é de extrema relevância a continuidade do desenvolvimento de projetos tecnológicos, que colaborem para uma rápida avaliação dos casos.

Nesse sentido, a utilização de recursos tecnológicos simples, como o script do câncer de mama permitem um bom auxílio na detecção de câncer, contribuindo para os profissionais médicos na busca pelo fornecimento de um prognóstico favorável, diminuindo assim, os dados alarmantes de mortes por neoplasias à nível mundial.

7 Trabalhos Futuros

Para os desenvolvimentos de novos trabalhos na área, sugere-se, o desenvolvimento de um reprodutor gráfico para demonstrar os resultados do Script de forma mais prática, e de fácil acesso, por exemplo uma interface web.

Em relação ao código desenvolvido nesse trabalho, um outro tipo de abordagem seria sua aplicação para outras doenças, como Hipertensão, Diabetes, que tem como base para seu diagnóstico exames que características padrões.

Em relação às linguagens de programação, a Golang ([GOOGLE, 2020](#)) apresenta uma maior velocidade, ao ser confrontado com outros programas já desenvolvidos, pois possui características estruturais únicas que lhe proporcionam essa vantagem. Também poderia ser utilizado o Javascript ([MOZILLA, 2020](#)), levando em consideração que ele é conhecido a nível mundial, de acordo com estatísticas pois possui características únicas como frontend web, o que em sua maioria, atrai mais o público.

Portanto, há muito o que se desenvolver no campo de Inteligência Computacional voltado à Engenharia Biomédica, auxílio ao diagnóstico de outras doenças e análise de exames.

Referências

BRASIL, M. da S. *Atlas do Câncer Relacionado ao Trabalho no Brasil*. 1. ed. Brasília - DF, 2018. Citado na página 13.

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 14.

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 12.

CAVALCANTI, T. *Support Vector Machine ou máquina de vetores de suporte*. 2019. Disponível em: <<https://www.codigofluente.com.br/wp-content/uploads/2019/06/SVM04.png>>. Acesso em: 30/04/2020. Citado na página 21.

COURNEPEAU, D. *Machine Learning in Python*. 2020. Disponível em: <<https://scikit-learn.org/stable/>>. Acesso em: 26/09/2019. Citado na página 16.

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 12.

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 14.

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 22.

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, 16 e 23.

FARINHA, R. *Inteligência Artificial*. 2014. Disponível em: <<https://slideplayer.com.br/slide/358847/>>. Acesso em: 30/09/2019. Citado na página 19.

FOUNDATION, P. S. *Python*. 2020. Disponível em: <<https://www.python.org/>>. Acesso em: 25/09/2019. Citado 2 vezes nas páginas 15 e 23.

GOOGLE. *Go*. 2020. Disponível em: <<https://golang.org/>>. Acesso em: 04/10/2019. Citado na página 27.

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 18.

- 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 16.
- MOZILLA. *JavaScript*. 2020. Disponível em: <<https://developer.mozilla.org/pt-BR/docs/Glossario/JavaScript>>. Acesso em: 04/10/2019. Citado na página 27.
- 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 22.
- 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 13.
- Orgnização 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 24.
- 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 20.

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