



UNIVERSIDADE  
CATÓLICA  
DE SANTOS

IPECI - INSTITUTO DE PESQUISAS  
CIENTÍFICAS E TECNOLÓGICAS

## COMITÊ INSTITUCIONAL DE INICIAÇÃO CIENTÍFICA – COIC/IPECI CICLO 2021/2022

### Relatório Final

O **relatório final** de atividades tem por objetivo demonstrar, de forma clara, o desenvolvimento e desempenho dos bolsistas de Iniciação Científica/Tecnológica. Ao preenchê-lo, é importante ter em mente que devem ser apresentados os avanços, os resultados positivos ou negativos e conclusões alcançadas.

Leia atentamente a descrição de cada item do relatório (Exemplo: INTRODUÇÃO, METODOLOGIA, etc.) e preencha conforme as instruções.

**Data de entrega:** Até 29 de agosto de 2022 – na plataforma Moodle da Universidade Católica de Santos – comunidade “IPECI – SALA IC\_IT\_2021-2022” **em formato Word, no TÓPICO [DISCENTES] RELATÓRIO FINAL**

Assunto e título do arquivo: "RF nome do aluno" Exemplo: RF Manuela Souza

### Avaliação Discente do Programa 2021-2022

Preenchimento **obrigatório** do **Questionário de avaliação do Programa de Iniciação Científica** na comunidade “IPECI – SALA IC\_IT\_2021-2022”. O formulário estará disponível no Moodle a partir do dia 15 de agosto de 2022.

### XVI JORNADA DE INICIAÇÃO CIENTÍFICA E TECNOLÓGICA

Pedimos também o envio do **resumo expandido** até **05 de setembro de 2022** em formulário eletrônico próprio, cujo *link* será disponibilizado em agosto. A submissão do resumo expandido é parte da participação obrigatória na **XVI Jornada de Iniciação Científica e Iniciação em Desenvolvimento Tecnológico e Inovação** a ser realizada em **19 e 20 de outubro de 2022**.

## RELATÓRIO FINAL DE ATIVIDADES DE PESQUISA

Instituto de Pesquisas Científicas e Tecnológicas – IPECI

Campus Dom Idílio José Soares - Av. Conselheiro Nébias, 300 - 11015-002 - Vila Mathias, Santos, SP - Tel.: (13) 3205-5555  
[www.unisantos.br](http://www.unisantos.br)



UNIVERSIDADE  
CATÓLICA  
DE SANTOS

IPECI - INSTITUTO DE PESQUISAS  
CIENTÍFICAS E TECNOLÓGICAS

(2021/2022)

**\*OBSERVAÇÃO: este relatório deve ser redigido pelo discente com supervisão do orientador**

<b>1. DADOS DO SUBPROJETO DE INICIAÇÃO CIENTÍFICA OU TECNOLÓGICA</b>
<b>1.4 TÍTULO DO SUBPROJETO: Estudo e Aplicações de Máquinas de Suporte Vetorial</b>
<b>1.5 ORIENTADOR: José Fontebasso Neto</b>
<b>1.6 ALUNO: Levy Souto Sousa</b>
<b>1.7 BOLSISTA: [ ] PIBIC / [ ] PIBITI / [ ] PROIN / [ ] PROITI / [ X ] VOLUNTÁRIO</b>
<b>2. DIVULGAÇÃO DA PESQUISA (Apresentação do trabalho em congressos/resumos publicados ou aprovados para publicação em anais de eventos; artigos publicados ou aprovados em periódicos; prêmios recebidos)</b>
<b>3. RESUMO DO SUBPROJETO (descrever de forma direta e concisa um resumo contendo 250 palavras, compreendendo os tópicos: problematização, objetivos, metodologia e resultados da pesquisa)</b>
<p>O estudo tem como propósito utilizar algoritmos de <i>Machine Learning</i> para realizar processamento de imagens de tomografia e ressonâncias magnéticas em três dimensões do depósito MRI, como cérebros, pulmões, fígado, entre outros, algumas delas contendo tumores ou edemas. Baseando o estudo em modelos de Classificação, foram usadas Máquinas de Suporte Vetorial (SVM)</p> <p>Com o processamento das diversas imagens, será possível separá-las em treino, <i>labels</i>(etiquetas) e teste, assim o algoritmo deverá ser capaz de prever em quais imagens de teste há um tumor ou não, e o nível de acurácia baseado nas <i>labels</i>.</p> <p>Observou-se que em outros algoritmos de classificação os resultados foram obtidos de forma rápida, em apenas alguns segundos, porém o algoritmo SVM se mostrou desvantajosa em relação aos demais algoritmos para trabalhar com o <i>Dataset</i> escolhido, devido ao grande tempo e esforço computacional.</p>

**Instituto de Pesquisas Científicas e Tecnológicas – IPECI**

Campus Dom Idílio José Soares - Av. Conselheiro Nébias, 300 - 11015-002 - Vila Mathias, Santos, SP - Tel.: (13) 3205-5555  
[www.unisantos.br](http://www.unisantos.br)



UNIVERSIDADE  
CATÓLICA  
DE SANTOS

IPECI - INSTITUTO DE PESQUISAS  
CIENTÍFICAS E TECNOLÓGICAS

#### 4. REFERENCIAL TEÓRICO (revisão da literatura que embasa o desenvolvimento da pesquisa)

Máquina de Suporte Vetorial (SVM) é um algoritmo supervisionado de Machine Learning utilizado para classificação e regressão de dados, separando-os em classes separadas linearmente ou não por um hiperplano. É dividido em módulos SVC para classificação e SVR para regressão. Na Figura 1.0 há um exemplo de classificação para dados rotulados utilizando SVC.

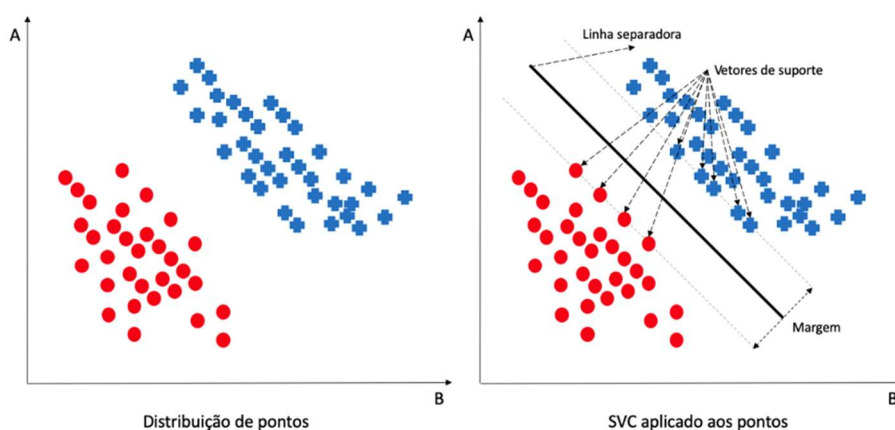


Figura 1.0

Fonte: Digital House

O SVM recebe dois parâmetros: *gamma* e *C* (*Scikit-Learn Documentation*); o parâmetro *gamma* é responsável por definir a influência de uma amostra no treinamento, quanto menor seu valor, pontos mais distantes são considerados no cálculo do hiperplano; Já o parâmetro *C* controla erros de margem na classificação e aplica penalidades afetando o hiperplano, quanto mais alto o valor do parâmetro *C*, mais clara será a separação entre classes, porém custa mais tempo de treinamento pela complexidade da formação de margens.

A Figura 1.1 mostra a aplicação do algoritmo com diferentes kernels, alterando a margem de classificação.

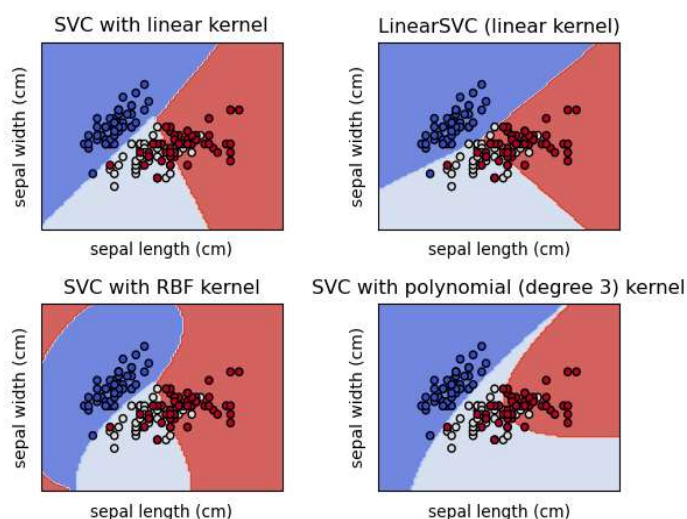


Figura 1.1  
Fonte: Scikit-Learn

## 5. METODOLOGIA (Descrever detalhadamente como a pesquisa foi desenvolvida, explicitando os procedimentos e os instrumentos utilizados, empregando uma redação científica, com uma linguagem impessoal)

A principal ferramenta utilizada foi a linguagem de programação Python, devido à sua gama de funcionalidades para manipulação de dados. Dentro desse escopo foram manuseadas as seguintes ferramentas: *Scikit-Learn*, *framework open-source* para suporte à algoritmos de Inteligência Artificial; Matplotlib, biblioteca para criação gráfica; Numpy para funções matemáticas, arranjos e matrizes; Nibabel, para manipular Neuro-Imagens.nii.gz; Nilearn, para trabalhar com Machine Learning em Neuro-Imagens, e Pickle, que permite guardar variáveis em arquivos e utilizá-las posteriormente no programa. As interfaces de desenvolvimento para interpretar o código-fonte foram Google Collab e Jupyter Notebook.

Para exemplificar, foram utilizadas tomografias de cérebros com tumores do *Dataset* MRI, como a representação abaixo, onde são exploradas camadas para verificar existência de tumores (PAL, Arghya; RATHI, Yogesh). As imagens com extensão.nii.gz podem ser manipuladas pela biblioteca Nibabel, são divididas em 4



UNIVERSIDADE  
CATÓLICA  
DE SANTOS

IPECI - INSTITUTO DE PESQUISAS  
CIENTÍFICAS E TECNOLÓGICAS

capturas, cada uma com 3 dimensões: altura, largura e profundidade(240X240X155).

O *Dataset* contém 484 imagens para treino do algoritmo, assim como suas respectivas *labels*, além de mais 250 para testes de predição.

Inicialmente é feito o carregamento do arquivo pelo Google Drive, a imagem é processada, selecionada e transformada em vetor numpy, reduz-se a dimensionalidade para 3D, a partir disso podemos criar as visualizações por camadas e eixos com Matplotlib, em algumas é possível notar o tumor.

Após estes processos, a biblioteca Nilearn é empregada com o intuito de aplicar a Máquina de Suporte Vetorial nos três eixos da imagem, a função *view\_img* permite interagir com o hiperplano e explorar as camadas, separando a visualização do tumor do resto do cérebro.

Em seguida são carregadas aproximadamente apenas 5 imagens de treino, *labels* e 2 para teste devido à limitação da Memória RAM, não é possível transferir todos os arquivos. A biblioteca Pickle permitiu guardar as variáveis em arquivos e carregá-las posteriormente com propósito de reutilizá-las sem grande esforço computacional.

Nas imagens de treino e testes, foram aplicadas máscaras booleanas para retirar o plano de fundo, e outras informações que não fossem o cérebro em si, após este processo, as variáveis de treino, *labels* e teste foram redimensionadas para respectivamente 2 dimensões (amostras, características), 1 dimensão(amostras) e 2 dimensões (amostras, características).

Finalmente é possível a aplicação do SVM a partir da documentação do *Scikit-Learn* em parte do *Dataset* para predição e classificação, o algoritmo deverá ser capaz de identificar se há tumores nas imagens de teste ou não, e revelar a precisão dos acertos (*Scikit-Learn Documentation*).

## 6. RESULTADOS E DISCUSSÃO (Esse é o item mais importante do relatório, no qual

Instituto de Pesquisas Científicas e Tecnológicas – IPECI

Campus Dom Idílio José Soares - Av. Conselheiro Nébias, 300 - 11015-002 - Vila Mathias, Santos, SP - Tel.: (13) 3205-5555  
[www.unisantos.br](http://www.unisantos.br)



UNIVERSIDADE  
CATÓLICA  
DE SANTOS

IPECI - INSTITUTO DE PESQUISAS  
CIENTÍFICAS E TECNOLÓGICAS

**devem constar os resultados obtidos, com uma descrição analítica, que podem ser expressos na forma de tabelas, gráficos ou figuras, quando for aplicável. Além disso, demonstrar neste item, a partir do levantamento bibliográfico, o confronto dos dados da literatura científica com os resultados obtidos)**

A imagem tem sua estrutura formada por *voxe/s* (grade regular tridimensional, equivalente a pixels em figuras 2d), podemos contar quantos *voxe/s* há em cada serie. temporal por cada imagem na Figura 2.0.

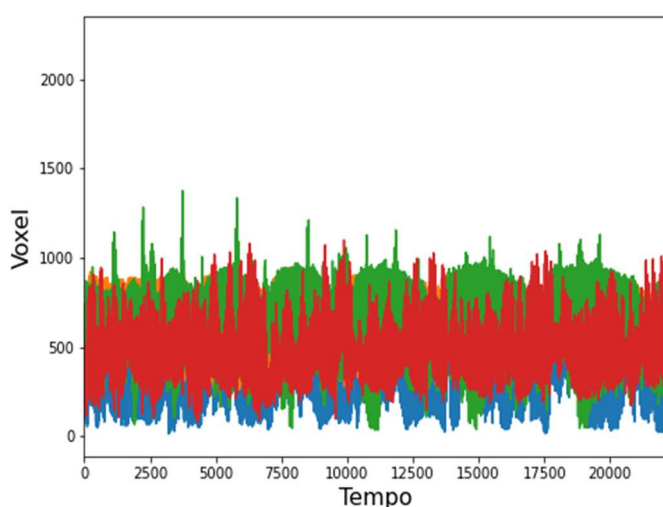
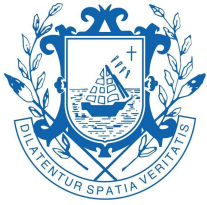


Figura 2.0

Fonte: autor

A Figura 2.1 é o código para criar a visualização da Figura 2.2 com a biblioteca Matplotlib, a função `para_3d` reduz a dimensionalidade da série temporal para 3d e retorna em vetor numpy. Na Figura 2.2 é possível observar o tumor.



UNIVERSIDADE  
CATÓLICA  
DE SANTOS

IPECI - INSTITUTO DE PESQUISAS  
CIENTÍFICAS E TECNOLÓGICAS

```
def para_3d(img_4d): #recebe imagem 4d e  
    devolve array numpy 3d  
    imagem_data = img_4d.get_fdata()#dado da  
    imagem em si  
    img_3d = imagem_data[...,3] #imagem 3d em  
    numpy array usando elipsis, 3 imagem em 4  
    return img_3d  
  
img_3d=para_3d(img)  
plt.imshow(img_3d[:, :, 77],cmap='inferno')  
plt.colorbar()
```

Figura 2.1

Fonte: autor

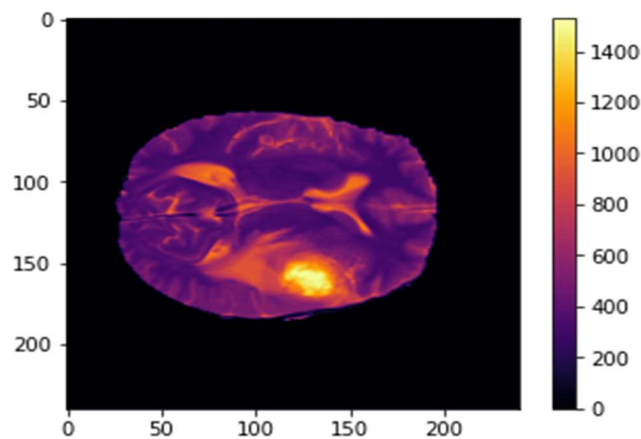
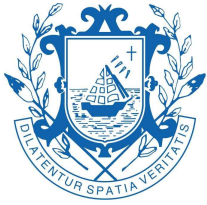


Figura 2.2

Fonte: autor

A Figura 2.3 apresenta o código para a exibição das camadas no eixo Z(cima), através de um índice de repetição que seleciona a partir das 60, dez camadas até a 110.





```
fig, axes = plt.subplots(nrows=1,ncols=6)

indices =[60,70,80,90,100,110]
for i in range(6):

    cerebro = img_3d[:, :, indices[i]]
    axes[i].imshow(cerebro, cmap='hsv')
    axes[i].axis('off')

plt.title('visualização de camadas no eixo Z')
plt.show()
```

Figura 2.3

Fonte: autor

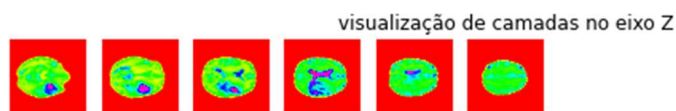


Figura 2.4

Fonte: autor

As Figuras 2.5 e 2.6 apresentam respectivamente o código para aplicação do SVM na imagem, é uma função interativa, sendo possível arrastar o hiperplano em qualquer eixo, o qual exibe a área a qual o tumor abrange.

```
from nilearn.plotting import view_img

view_img(lista_3d[3],
bg_img=lista_3d[3],title="SVM hiperplano
interativo",
dim=-1,colorbar=True,cbar_tick_format= "%i")
```

Figura 2.5

Fonte: autor



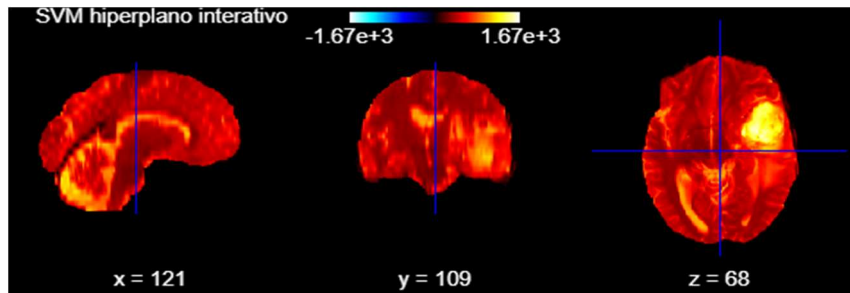
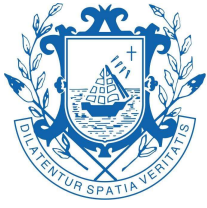


Figura 2.6

Fonte: autor

Agora, são separados treino, teste e *labels*, aplicando máscaras para remover o plano de fundo e guardando seus valores em listas, seguindo o mesmo processo da Figura 2.7.

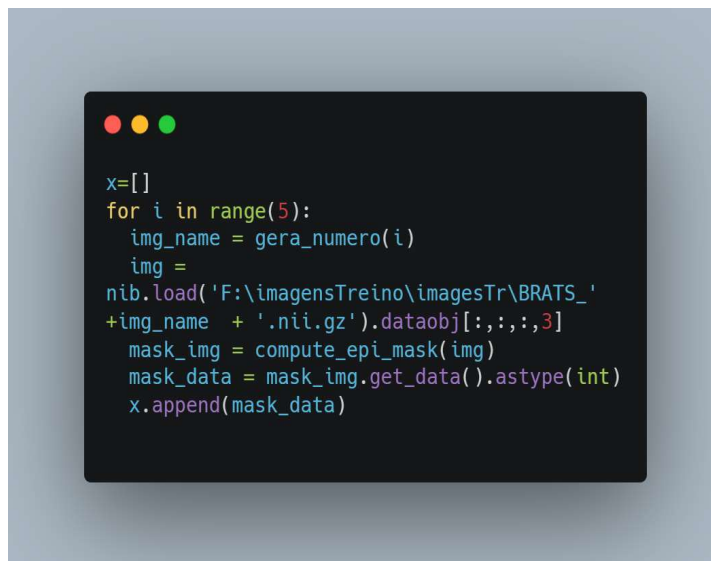


Figura 2.7

Fonte:autor

Após os dados lidos, são transformados em *arrays* numpy e passados para uma dimensão. Como mostra a Figura 2.8



UNIVERSIDADE  
CATÓLICA  
DE SANTOS

IPECI - INSTITUTO DE PESQUISAS  
CIENTÍFICAS E TECNOLÓGICAS

```
x = np.array(x)
y=np.array(y)
t= np.array(t)
x,t,y =
x.reshape(5,240*240*155),t.reshape(2,240*240*1
55),y.reshape(5,240*240*155)#dados com 2
dimensões
y = y.reshape(5*240*240*155)
x=x.reshape(5*240*240*155)#normalizando x e y
para uma dimensão
x = x.reshape(-1,1)
```

Figura 2.8

Fonte: Autor

Finalmente, é importado o módulo SVC do *Scikit-Learn*, os dados são mais uma vez separados com *train\_test\_split*, é realizado o fit e calculado a taxa de acerto. Como mostra a Figura 2.9.

```
from sklearn.svm import SVC
clf = svm.SVC(kernel = 'rbf', gamma = 'auto')
from sklearn.model_selection import
train_test_split

clf =
OneVsRestClassifier(BaggingClassifier(SVC(kern
el='rbf', probability=True,
class_weight='balanced'),max_samples=1.0/n_est
imators,n_estimators=n_estimators) )

clf.fit(x_train,y_train)
clf.predict(t)

acerto = clf.score(x_test,y_test) * 100
print("taxa de acerto %d%%" %acerto)
```

Figura 2.9

Fonte: Autor

**Instituto de Pesquisas Científicas e Tecnológicas – IPECI**

**Campus Dom Idílio José Soares** - Av. Conselheiro Nébias, 300 - 11015-002 - Vila Mathias, Santos, SP - Tel.: (13) 3205-5555  
**www.unisantos.br**



A Figura 3.0 mostra o código *fit* em execução, depois de algumas horas processando devido ao cálculo quadrático do SVM temos o resultado abaixo.

```
clf.fit(x, y)

... "Label %s is present in all training examples." % str(classes[c])
/usr/local/lib/python3.7/dist-packages/sklearn/multiclass.py:80: UserW
"Label %s is present in all training examples." % str(classes[c])
/usr/local/lib/python3.7/dist-packages/sklearn/multiclass.py:80: UserW
"Label %s is present in all training examples." % str(classes[c])
/usr/local/lib/python3.7/dist-packages/sklearn/multiclass.py:80: UserW
"Label %s is present in all training examples." % str(classes[c])
```

Figura 3.0

Fonte: Autor

Por fim, a Figura 3.1 mostra a taxa de acerto algoritmo, e apesar de alta, não se pode dizer que é satisfatória, devido ao baixo número de dados de treino, em sistemas computacionais mais avançados, há a possibilidade de treinar o algoritmo com mais dados e mais *labels*.

```
score = clf.score(x,y)*100
print("taxa de acerto %d%%" %score)

taxa de acerto 98%
```

Figura 3.1

Fonte: Autor

## 7. CONSIDERAÇÕES FINAIS/CONCLUSÃO (Relacionar os objetivos e problematização com os resultados)

A partir do que foi exposto, conclui-se que a Máquina de Suporte Vetorial é eficiente quando se trata de previsões por classificação, tanto em circunstâncias mais simples como venda de automóveis distintos por ano, quanto em circunstâncias complexas como a detecção de tumores cerebrais. Embora não tenha sido possível treinar e testar com o *Dataset* completo devido à limitação de hardware e esforço computacional, obtém-se uma visão positiva da capacidade do algoritmo em ambientes com maior potencial de processamento.

Diante disso, podemos inferir que a Máquina de Suporte Vetorial é eficiente e pode ser aplicada em ambientes comerciais e até mesmo hospitalares, supervisionada e treinada por um profissional, pode ser de grande utilidade para o meio a qual esta tecnologia estiver inserida.



**8. REFERÊNCIAS (Indicar as obras, documentos, artigos científicos e demais materiais bibliográficos/eletrônicos utilizados nesta primeira etapa de acordo com as Normas da ABNT)**

Osuna E., Freund R., and Girosi F., "Support Vector Machines: Training and Applications", A.I. Memo No. 1602, Artificial Intelligence Laboratory, MIT, 1997.

MORETTIN, Pedro A.; SINGER, Julio M. **Introdução à ciência de dados fundamentos e aplicações**. São Paulo-SP: Departamento de Estatística da Universidade de São Paulo, IMEUSP, 2020. 354 p.

EVGENIU, Theodorus; PONTIL, Massimiliano. **Support Vector Machines: Theory and Applications** DOI: [10.1007/3-540-44673-7\\_12](https://doi.org/10.1007/3-540-44673-7_12). Disponível em: [https://www.researchgate.net/publication/221621494\\_Support\\_Vector\\_Machines\\_Theory\\_and\\_Applications](https://www.researchgate.net/publication/221621494_Support_Vector_Machines_Theory_and_Applications), janeiro, 2001.

DOBILAS, Saul. Classificador SVM e kernel RBF.

Disponível em:

<https://ichi.pro/pt/classificador-svm-e-kernel-rbf-como-fazer-melhoresmodelos-em-python-127248222170971>.

Michael Joseph, Jerrold Jeyachandra, and Erin Dickie (eds):

Data Carpentry: **Introduction to MRI Data Analysis**." Version 2019.11, November 2019,

Disponível em: <https://github.com/carpentries-incubator/SDC-BIDS-IntroMRI>

Bailey, Sthephen, "**Exploring 3D images with matplotlib**". Dezembro de 2017.

Disponível em: <https://www.youtube.com/watch?v=5jQVQE6yfio>

Scarpance, L. et al. Radiology Data from The Cancer Genome Atlas Glioblastoma Multiforme [TCGA-GBM] collection [Data set]. The Cancer Imaging Archive, 2016. Disponível em: <https://doi.org/10.7937/K9/TCIA.2016.RNYFUYE9>.



UNIVERSIDADE  
CATÓLICA  
DE SANTOS

IPECI - INSTITUTO DE PESQUISAS  
CIENTÍFICAS E TECNOLÓGICAS

**SVM RBF parameters**, Scikit-Learn documentation disponível em: [https://scikit-learn.org/stable/auto\\_examples/svm/plot\\_rbf\\_parameters.html#rbf-svm-parameters](https://scikit-learn.org/stable/auto_examples/svm/plot_rbf_parameters.html#rbf-svm-parameters).

Acesso em setembro de 2021.

PAL, Arghya; RATHI, Yogesh, **A review of deep learning methods for MRI reconstruction.**

PDF Disponível em: <https://arxiv.org/abs/2109.08618>. Cornwell University. Acesso em set 2021

**9. ACEITE ELETRÔNICO DO ORIENTADOR (O aluno deverá postar o relatório final no moodle depois da autorização do professor orientador)**

O aceite do orientador deve ser preenchido pelo(a) orientador(a) em **questionário** próprio **[IC-IT] Avaliação Relatório Final Orientador 2021-2022** IPECI (disponível na comunidade moodle **IPECI - SALA IC\_IT\_2020-2021**)

**Sem o aceite do orientador, o relatório final não será aceito.**

**Instituto de Pesquisas Científicas e Tecnológicas – IPECI**

**Campus Dom Idílio José Soares** - Av. Conselheiro Nébias, 300 - 11015-002 - Vila Mathias, Santos, SP - Tel.: (13) 3205-5555  
**www.unisantos.br**