

Nome: João Geiger Piza

Repositório no Github: <https://github.com/joaogpizza/Lipai>

Códigos das Videoaulas:

introducao.py:

```
# matplotlib

import matplotlib.pyplot as plt
import numpy as np

t = np.linspace(0, 2*(np.pi), 500)
y = np.cos(4*t)
y1 = np.sin(4*t)

plt.figure('Cosseno', figsize=(7,5))
plt.plot(t, y)
plt.title('Gráfico do Cosseno')
plt.xlabel('Tempo (s)')
plt.ylabel('Amplitude')

plt.figure('Seno')
plt.plot(t, y1)
plt.title('Gráfico do Seno')
plt.xlabel('Tempo (s)')
plt.ylabel('Amplitude')
plt.show()
```

aula01.py:

```
import matplotlib.pyplot as plt
import numpy as np

t = np.linspace(0, 2*(np.pi), 100)
y = np.cos(t)
y1 = np.sin(t)

plt.figure('Gráfico', figsize= (6, 4))
plt.plot(t, y)
plt.plot(t, y1)
plt.title('Gráficos do Seno e Cosseno')
plt.xlabel('Eixo de Tempo')
plt.ylabel('Eixo da Amplitude')
plt.grid()
plt.show()
```

aula02.py:

```
import matplotlib.pyplot as plt
import numpy as np

x = np.linspace(0, 2*(np.pi), 500)
c = np.cos(x)
s = np.sin(x)

plt.figure('Gráficos cosenoidais', figsize=(8, 4))
plt.subplots_adjust(
    left=0.152,
    right=0.943,
    top=0.9,
    bottom=0.14,
    wspace=0.438,
    hspace=0.4
)

ax1 = plt.subplot(1, 2, 1)
plt.plot(x, c)
ax1.set_title('Gráfico do Cosseno')
ax1.set_xlabel('Eixo de Tempo')
ax1.set_ylabel('Eixo da Amplitude')
ax1.grid()

ax2 = plt.subplot(1, 2, 2)
plt.plot(x, s)
ax2.set_title('Gráfico do Seno')
ax2.set_xlabel('Eixo de Tempo')
ax2.set_ylabel('Eixo da Amplitude')
ax2.grid()

plt.show()
```

aula03.py:

```
import matplotlib.pyplot as plt
import numpy as np

x = np.arange(0, 5, 0.1)
y1 = x**2
y2 = x**5
```

```

# subplots
fig, axes = plt.subplots(nrows=2, ncols=2, figsize=(8,4))
plt.suptitle('Gráficos com subplots')
axes[0, 0].plot(x, y1)
axes[0, 0].set_title('Função do Quinto Grau  $x^5$ ')
axes[0, 0].set_xlabel('Tempo')
axes[0, 0].set_ylabel('Amplitude')
axes[0, 0].grid = True

axes[0, 1].plot(x, y1)
axes[0, 1].set_title('Função do Quinto Grau  $x^5$ ')
axes[0, 1].set_xlabel('Tempo')
axes[0, 1].set_ylabel('Amplitude')
axes[0, 1].grid = True

axes[1, 0].plot(x, y2)
axes[1, 0].set_title('Função do Quinto Grau  $x^5$ ')
axes[1, 0].set_xlabel('Tempo')
axes[1, 0].set_ylabel('Amplitude')
axes[1, 0].grid = True

axes[1, 1].plot(x, y2)
axes[1, 1].set_title('Função do Quinto Grau  $x^5$ ')
axes[1, 1].set_xlabel('Tempo')
axes[1, 1].set_ylabel('Amplitude')
axes[1, 1].grid = True

plt.show()

```

aula04.py:

```

from matplotlib import pyplot as plt
import numpy as np

x = np.linspace(0, 2*(np.pi), 70)
y = np.cos(4*x)

plt.figure(figsize=(8, 4))
plt.plot(x, y, color='#0f0f0f80', lw=0.5, marker='o',
         linestyle='dashdot')
plt.grid(True)
plt.title('Gráfico do Cosseno')

```

```
plt.xlabel('Eixo do Tempo')
plt.ylabel('Eixo da Amplitude')
plt.show()
```

aula05.py:

```
from matplotlib import pyplot as plt
import numpy as np

plt.style.use('ggplot')

x = np.linspace(0, 2*(np.pi), 300)
y = np.cos(3*x)

fig, axe = plt.subplots(figsize=(7, 4))
axe.plot(x,y)

axe.set_title('Gráfico do Cosseno', fontsize=16)
axe.set_xlabel('Eixo X', fontsize=14)
axe.set_ylabel('Eixo Y', fontsize=14)

SEPX = 0.4
SEPY = 0.3

plt.xticks(np.arange(0, 2*(np.pi)+SEPX, SEPX))
plt.yticks(np.arange(-1, 1+SEPY, SEPY))

plt.show()
```

aula06.py:

```
from matplotlib import pyplot as plt
import numpy as np

plt.style.use('ggplot')

x = np.linspace(1, 5, 500)
y = np.log10(x)

fig, axe = plt.subplots(figsize=(7, 4))
axe.plot(x, y, lw=1.2)

axe.text(2.6, 0.35, 'P(2,5; 0,4)')
axe.text(3, 0.42, 'Logarítmo  $y = \log_{10}x$ ',
```

```

        fontsize=10,

        bbox=dict(facecolor='red', alpha=0.5))
axe.annotate('P(2,5; 0,4)', xy=(2.5, 0.4),
            fontsize=14, xytext=(0.5, 0.5),
            arrowprops=dict(facecolor='red'),
            color='r')

axe.plot([0, 2.5], [0.4, 0.4],
        color='gray', linestyle='--', lw=0.8)
axe.plot([2.5, 2.5], [0.0, 0.4],
        color='gray', linestyle='--', lw=0.8)
axe.plot(2.5, 0.4, marker='o', color='gray')
axe.set_title('Gráfico Logarítmico')
axe.set_xlabel('Eixo X')
axe.set_ylabel('Eixo Y')
axe.set_xticks(np.arange(0, 5.5, 0.5))
plt.show()

```

Exercícios:

ex01.py:

```

""" Ex 01 S6A3 """

import os
from matplotlib import pyplot as plt
import numpy as np
import pandas as pd

caminho = os.path.join(os.path.dirname(__file__),
"classification_results_trial_0001.csv")

df = pd.read_csv(caminho)
# Item 1:
benign_malign = df.groupby('real_class').count()['image_path']
benign = int(benign_malign.get('benign', 0))
malign = int(benign_malign.get('malign', 0))
salto = 5

plt.bar(np.array([f'Benigno\n{benign}', f'Maligno\n{malign}']),
        np.array([benign, malign]),
        width=0.3)

plt.title('Contagem por real_class')
plt.ylabel('real_class')
plt.yticks(np.arange(0, 100+salto, salto))

```

```

plt.show()

# Item 2:
predicted_benign_malign =
df.groupby('predicted_class').count()['image_path']
predicted_benign = int(predicted_benign_malign.get('benign', 0))
predicted_malign = int(predicted_benign_malign.get('malign', 0))
salto = 5

plt.bar(np.array([f'Benigno\n{predicted_benign}',
f'Maligno\n{predicted_malign}']),
        np.array([predicted_benign, predicted_malign]),
        width=0.3)

plt.title('Contagem por predicted_class')
plt.ylabel('predicted_class')
plt.yticks(np.arange(0, 100+salto, salto))
plt.show()

# Item 3
probs_benign = np.asarray(df['prob_benign'])
qtd_intervalos = 10
salto = (probs_benign.max()-probs_benign.min()) / qtd_intervalos
plt.hist(probs_benign, bins=qtd_intervalos, edgecolor='black', lw=0.6,
alpha=0.9)
plt.xlabel('prob_benign')
plt.xticks(np.arange(probs_benign.min(), probs_benign.max() + salto,
salto))
plt.ylabel('contagem')
plt.title('Histograma de prob_benign')
plt.show()

# Item 4
probs_malign = np.asarray(df['prob_benign'])
qtd_intervalos = 20
salto = (probs_malign.max()-probs_malign.min()) / qtd_intervalos
plt.hist(probs_malign, bins=qtd_intervalos, edgecolor='black', lw=0.6,
alpha=0.9)
plt.xlabel('prob_malign')
plt.xticks(np.arange(probs_malign.min(), probs_malign.max() + salto,
salto))
plt.ylabel('contagem')
plt.title('Histograma de prob_malign')
plt.show()

```

```

# Item 5
x = np.asarray(df['prob_benign'])
y = np.asarray(df['prob_malign'])
real = np.asarray(df['real_class'])
pred = np.asarray(df['predicted_class'])
mask_ok = (real == pred)
mask_err = ~mask_ok
plt.scatter(x[mask_ok], y[mask_ok], s=30, alpha=0.7, label='acerto')
plt.scatter(x[mask_err], y[mask_err], s=30, alpha=0.8, marker='x',
label='erro')
plt.xlabel('prob_benign')
plt.ylabel('prob_malign')
plt.title('Scatter plot: prob_benign vs prob_malign')
plt.legend()
plt.grid(alpha=0.3)
plt.show()

```

```

# Item 6
# O mais comum é o falso negativo, com 9 ocorrências, enquanto
# o falso positivo tem apenas 3 ocorrências
# Código para criar o gráfico:
positivo = 'malign'
negativo = 'benign'
FP = ((df['real_class'] == negativo) & (df['predicted_class']
==positivo)).sum()
FN = ((df['real_class'] ==positivo) & (df['predicted_class'] ==
negativo)).sum()
salto = 1
plt.bar(np.array([f'FP\n{FP}', f'FN\n{FN}']),
np.array([FP, FN]))
plt.title('FP vs FN')
plt.ylabel('Quantidade')
plt.yticks(np.arange(0, 20+salto, salto))
plt.show()

```

```

# Item 7
# No contexto médico, ambos são preocupantes. Porém, no caso
# de um falso positivo, o paciente somente passará por um
# tratamento desnecessário, que pode ou não lhe prejudicar.
# Já no caso de um falso negativo, o paciente não saberá que
# algo de errado está ocorrendo com ele, permitindo uma possível

```

```
# evolução dos sintomas. Logo, deduzo que o falso negativo seja  
# mais preocupante dentre os dois.
```

ex02.py:

```
""" Ex02 S6A3 """  
  
import os  
from matplotlib import pyplot as plt  
import numpy as np  
import pandas as pd  
  
caminho = os.path.join(os.path.dirname(__file__), "metrics.csv")  
  
df = pd.read_csv(caminho)  
  
total = df.count()['train_loss']  
epochs = np.arange(1, total + 1)  
precisao_train = np.asarray(df['train_acc'])  
precisao_val = np.asarray(df['val_acc'])  
perda_train = np.asarray(df['train_loss'])  
perda_val = np.asarray(df['val_loss'])  
  
fig, eixos = plt.subplots(nrows=1, ncols=2, figsize=(12,4))  
plt.suptitle('Gráficos de treinamento')  
  
eixos[0].plot(epochs, precisao_train, label='train')  
eixos[0].plot(epochs, precisao_val, label='valid')  
eixos[0].set_title('model accuracy')  
eixos[0].set_xlabel('epoch')  
eixos[0].set_ylabel('accuracy')  
eixos[0].legend()  
  
eixos[1].plot(epochs, perda_train, label='train')  
eixos[1].plot(epochs, perda_val, label='valid')  
eixos[1].set_title('model loss')  
eixos[1].set_xlabel('epoch')  
eixos[1].set_ylabel('loss')  
eixos[1].legend()  
  
plt.show()
```