

SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I
INFORMACIJSKIH TEHNOLOGIJA OSIJEK

Sveučilišni diplomski studij računarstva

STATISTIČKA ANALIZA ISHODA PACIJENTA PREMA
ZNAČAJKAMA IZ MEDICINSKIH SLIKA

Obrada slike i računalni vid

Karlo Kokanović

Osijek, 2023.

Sadržaj

1.	Uvod	2
1.1.	Zadatak rada	2
2.	Pregled problematike	3
2.1.	Korištena tehnologija	3
3.	Programsko rješenje	4
4.	Zaključak	12
Prilog.	14

1. UVOD

Sarkomi mekog tkiva su skupina rijetkih karcinoma koji zahvaćaju tkiva koja povezuju, podupiru i okružuju druge tjelesne strukture i organe. Tkiva koja mogu zahvatiti sarkomi mekog tkiva uključuju masno tkivo, mišiće, krvne žile, duboko tkivo kože, kosti, tetive i ligamente. Sarkomi mekog tkiva mogu se razviti u gotovo bilo kojem dijelu tijela, uključujući noge, ruke i trbuh (abdomen). [1] Postoje brojne stvari za koje se zna da povećavaju rizik oboljenja, kao primjerice dob, genetika, prijašnje izlaganje radioterapiji i/ili određenim kemikalijama. Ukoliko liječnik sumnja na sarkom, vjerojatno će Vam trebati potpuni pregled i pretrage, uključujući:

- Uzorak stanica tumora, koji se naziva biopsija
- Slikovni testovi, kao što su CT skeniranje, ultrazvuk ili magnetska rezonanca, koji vam pomažu vidjeti unutar vašeg tijela
- Skeniranje kostiju, ako možda imate osteosarkom

Proces liječenja će ovisiti o tome koju vrstu sarkoma imate, gdje se u vašem tijelu nalazi, koliko je razvijen i je li se proširio na druge dijelove vašeg tijela ili nije metastazirao. Metode tretiranja uključuju operaciju, zračenje (pogotovo ako operacija nije opcija), kemoterapija i ciljane terapije.

1.1. Zadatak rada

Na temelju PET/CT slika ćemo vidjeti kakav je utjecaj značajki sarkoma (npr. veličina) na ishod pacijenta, odnosno hoće li pacijent imati metastaze ili ne. Metastaziranje znači da se se karcinom proširio dalje po tijelu. Intuitivno (i prema dokazima) očekujemo da što je rak veći i što je prokrvljeniji to je veća vjerojatnost za metastaziranje. Zadatak ovog rada je provjeriti tu tvrdnju.

2. PREGLED PROBLEMATIKE

Za izradu rješenja korišten je skup podataka dostupan na: [2]. Preuzeta je zadnje ažurirana verzija. Skup podataka se sastoji od tri datoteke: *lab_petct_vox_5.00mm.h5*, *patient_images_lowres.h5* te *study_list.csv*. U prvoj i ujedno najvažnijoj datoteci se nalaze tri grupe slika, a to su ct, pet i maske (provjereno definiranjem funkcije za ispis sadržaja pojedine .h5 datoteke). Slike su dostupne za ukupno sedam pacijenata. Druga datoteka sa slikama neće biti uzeta u obzir za izradu ovoga rada. U .csv tablici se nalaze podatci o svakome od pacijenata, ona sadrži podatke o 18 pacijenata, dok ćemo mi uzeti u obzir samo za 7 dostupnih. Tablica nam nudi informacije kao što su dob, spol, broj godina, tip sarkoma i sl. Nama je ključan podatak *Outcome* (is-hod), koji sadrži vrijednosti za 'Mets' (metastaze), 'Recurrence' (rak se vratio na isto mjesto) i prazno (rak se nije vratio). Za potrebe izrade, prve dvije vrijednosti ćemo podijeliti u novu klasu "Povratak raka", a prazno kao "Nije se vratio".

2.1. Korištena tehnologija

Cijelo programsko rješenje je pisano u programskome jeziku Python. Python nam nudi nekoliko biblioteka za lakšu obradu slike, te kasnije statističku obradu. *NumPy* je Python biblioteka koja se koristi za rad s nizovima, matricama, općenito domena linearne algebre. *Pandas* pak koristimo za analizu podataka. *Matplotlib.pyplot* i *seaborn* nam služe za iscrtavanje grafova, *os* za rad s operacijskim sustavom, *h5py* rad sa .h5 datotekama (koje sadrže slike), a preostale navedene za obradu i manipulaciju samih slika (slika 2.1).

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import os
import h5py
import cv2
from PIL import Image, ImageChops
from skimage import measure, img_as_ubyte, io
import seaborn as sns
from skimage.filters import threshold_otsu
from skimage.color import rgb2gray
```

Slika. 2.1: Korištene biblioteke

3. PROGRAMSKO RJEŠENJE

Prvi korak nam je izvući slike iz .h5 datoteke, kako bi si vizualizirali problematiku. Otvorimo samu datoteku, spremimo vrijednosti za svaku od tri grupe unutar nje, pa definiramo sliku dimenzija 7x4, gdje ćemo za svakog pacijenta u redu imati ct sliku, pet sliku, masku i rezultatnu sliku redom (slika 3.1). Rezultatnu sliku dobijemo množenjem maske i pet slike. Množenje ide element po element svakog od odgovarajućih piksela u paru ulaznih slika i vraća umnožak tih množenja u odgovarajućem pikselu u izlaznoj slici. Slika sa svim navedenim izgleda ovako: slika 3.4.

```
with h5py.File(os.path.join('Lab_petct_vox_5.00mm.h5'), 'r') as p_data:
    ct_images = p_data['ct_data'].items()
    pet_images = p_data['pet_data'].values()
    lab_images = p_data['label_data'].values()
    fig, sb_mat = plt.subplots(7, 4, figsize=(10,25))
    (ax1s, ax2s, ax3s, ax4s) = sb_mat.T
    for c_ax1, c_ax2, c_ax3, c_ax4, (p_id, ct_img), pet_img, lab_img in zip(ax1s, ax2s, ax3s, ax4s, ct_images, pet_images, lab_images):

        ct_image = np.mean(ct_img, 1)[::-1]
        c_ax1.imshow(ct_image, cmap = 'bone')
        c_ax1.axis('off')

        pet_image = np.sqrt(np.max(pet_img, 1).squeeze()[::-1,:])
        c_ax2.imshow(pet_image, cmap = 'bone')
        c_ax2.axis('off')

        label = np.mean(lab_img, 1)[::-1]
        c_ax3.imshow(label, cmap='bone')
        c_ax3.axis('off')

        result = cv2.multiply(pet_image, label)
        c_ax4.imshow(result, cmap='bone')
        c_ax4.axis('off')
```

Slika. 3.1: Funkcija za isrcetavanje slika

```
for x in range(sb_mat.shape[0]):
    for y in range(sb_mat.shape[1]):
        extent = sb_mat[x, y].get_window_extent().transformed(fig.dpi_scale_trans.inverted())
        fig.savefig('images/' + str(x) + str(y) + '.png', bbox_inches=extent.expanded(1.1, 1.2))
```

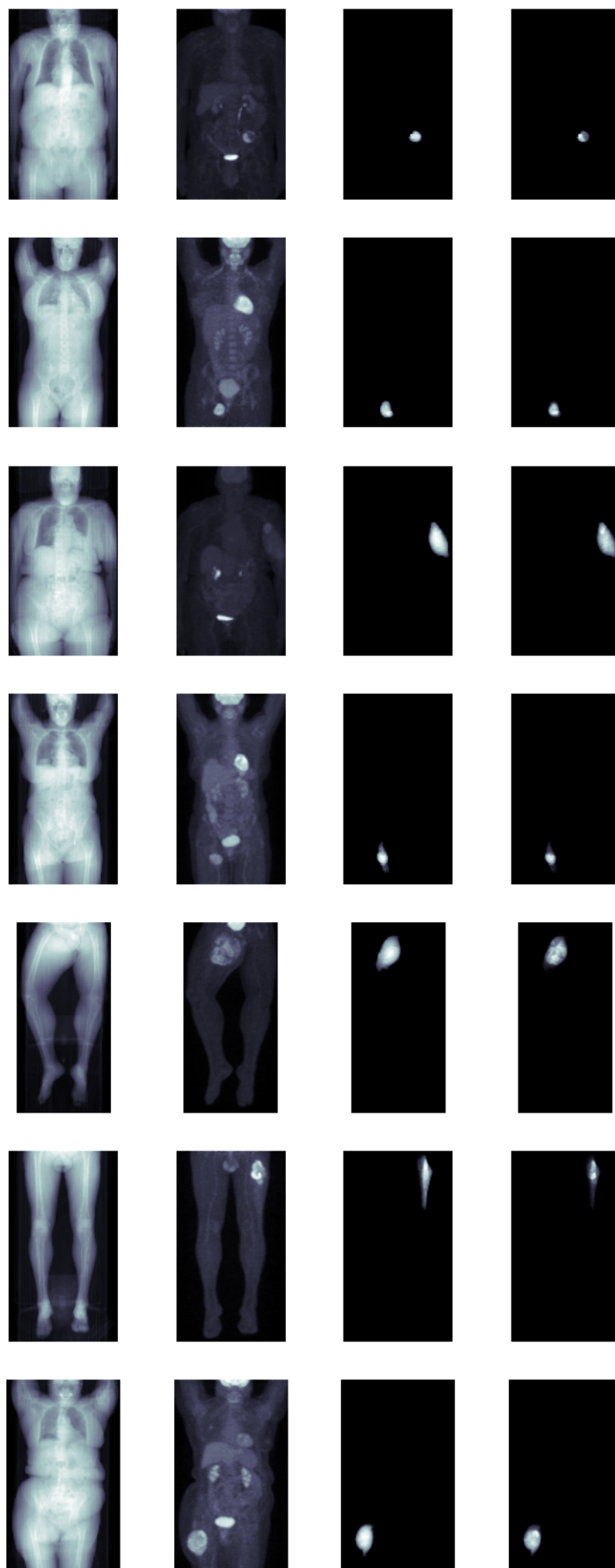
Slika. 3.2: Spremanje rezultata

```
def trim(im):
    bg = Image.new(im.mode, im.size, im.getpixel((0,0)))
    diff = ImageChops.difference(im, bg)
    diff = ImageChops.add(diff, diff, 2.0, -100)
    bbox = diff.getbbox()
    if bbox:
        return im.crop(bbox)
    else:
        return trim(im.convert('RGB'))
for i in range(0,7):
    im = Image.open("images/" + str(i) + ".png")
    im = trim(im)
    im = im.save("images/Final" + str(i) + ".png")
```

Slika. 3.3: Odbacivanje pozadine

Na osnovu iscrtanog, želimo spremiti samo najdasniju sliku, odnosno produkt množenja. To postizemo iteriranjem red po red, dok ne dođemo do četvrtog elementa i samo njega spremaemo (slika 3.2). Rezultat spremanja zbog proširivanja obuhvati i dio bijele pozadine, što nam može kasnije stvarati dodatne probleme, te iz tog razloga sa slike odrežemo neželjeno (slika 3.3).

Kada smo izvukli dio prekriven sarkomom, možemo započeti analizirati to područje. Najprije moramo obuhvatiti isključivo regiju sarkoma. Prvi korak je učitati rezultatnu sliku kao



Slika. 3.4: *Konačna slika*

sliku u sivim tonovima (grayscale). Drugim riječima, grayscale je ona slika u kojoj je vrijednost svakog piksela je jedan uzorak koji predstavlja samo količinu svjetlosti, odnosno nosi samo informaciju o intenzitetu [3]. Pretvorimo zatim sliku u format bajtova bez predznaka, s vrijednostima u $[0, 255]$. Pozivamo Otsu thresholding. Metoda obrađuje histogram slike, segmentirajući objekte minimiziranjem varijance na svakoj od klasa. Označimo značajke u niz, te mjerimo svojstva označenih područja slike (slika 3.5). Promatrati ćemo sljedeće parametre:

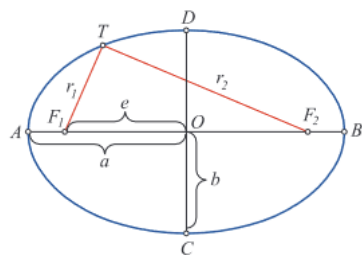
- 'area' - volumen sarkoma,
- 'eccentricity' - ekscentričnost je omjer žarišne udaljenosti (udaljenosti između žarišnih točaka) preko duljine glavne osi. Vrijednost je u intervalu $[0, 1)$. Kada je 0, elipsa postaje krug,
- 'axis_major_length' - duljina velike osi elipse koja ima iste normalizirane sekundarne središnje momente kao i područje,
- 'axis_minor_length' - duljina male osi elipse koja ima iste normalizirane sekundarne središnje momente kao i područje,
- 'intensity_mean' - vrijednost s prosječnim intenzitetom u regiji,
- 'intensity_max' - vrijednost najvećeg intenziteta u regiji,
- 'orientation' - kut između nulte osi i glavne osi elipse koji ima iste sekundarne momente kao i područje, u rasponu od $-\frac{\pi}{2}$ do $\frac{\pi}{2}$ u smjeru suprotnom od kazaljke na satu.

```
areas = []
eccentricity = []
axis_maj_l = []
axis_min_l = []
inten_mean = []
orient = []
inten_max = []
for j in range(0,7):
    test = img_as_ubyte(rgb2gray(io.imread("images/Final" + str(j) + ".png")))
    threshold = threshold_otsu(test)
    label_image = measure.label(test > threshold, connectivity = test.ndim)
    props = measure.regionprops_table(label_image, test, properties=['area', 'eccentricity', 'axis_major_length', 'axis_minor_length',
                                                                    'intensity_mean', 'intensity_max', 'orientation'])

    areas.append(props['area'][0])
    eccentricity.append(props['eccentricity'][0])
    axis_maj_l.append(props['axis_major_length'][0])
    axis_min_l.append(props['axis_minor_length'][0])
    inten_mean.append(props['intensity_mean'][0])
    inten_max.append(props['intensity_max'][0])
    orient.append(props['orientation'][0])
```

Slika. 3.5: Odabir regije i svojstava

Elipsa je skup svih točaka ravnine kojima je zbroj udaljenosti od dviju zadanih točaka uvijek isti (slika 3.6). Neka su zadane dvije točke, F_1 i F_2 . Njihovu udaljenost označimo s $2e$: $|F_1F_2| = 2e$. Udaljenost točke T u ravnini do točaka F_1 i F_2 označimo s r_1 i r_2 : $r_1 = |F_1T|$, $r_2 = |F_2T|$. Neka je a bilo koji broj veći od e . Točke F_1 i F_2 nazivaju se žarišta elipse. Polovica udaljenosti između žarišta je broj e koji nazivamo linearni ekscentricitet elipse. Vektore $\vec{F_1T}$ i $\vec{F_2T}$ nazivamo radijvektorima elipse. r_1 i r_2 su, dakle, duljine radijvektora elipse. Polovište O dužine $\overline{F_1F_2}$ zovemo središte. Pravac kroz žarišta siječe elipsu u točkama A i B . Dužina \overline{AB} naziva se velika os elipse, pa su dužine \overline{OA} i \overline{OB} velike poluosi elipse. Pravac koji prolazi središtem okomito na veliku os siječe elipsu u točkama C i D . Dužinu \overline{CD} nazivamo mala os, pa su \overline{OC} i \overline{OD} male poluosi elipse [4].



Slika. 3.6: *Primjer elipse*

Sada kada imamo bolji uvid u podatke, računamo ih i spremimo u novu .csv datoteku (slika 3.7).

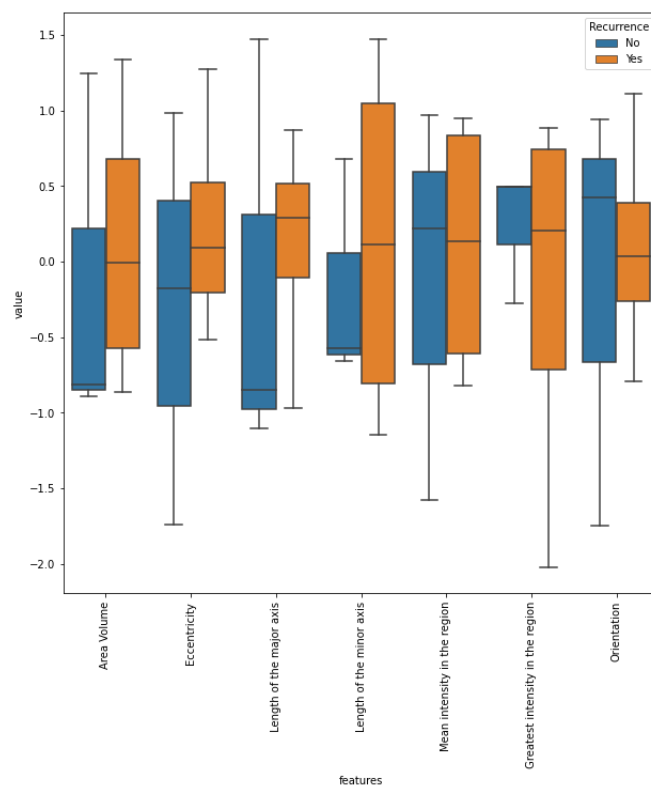
```
prop_indices = [0, 1, 2, -1, 3, -1, -1, 4, -1, 5, -1, -1, 6, -1, -1, -1, -1]
prop_names = ['Area Volume', 'Eccentricity', 'Length of the major axis', 'Length of the minor axis',
              'Mean intensity in the region', 'Greatest intensity in the region', 'Orientation']
prop_values = [areas, eccentricity, axis_maj_l, axis_min_l, inten_mean, inten_max, orient]

all_patients = ['STS_002', 'STS_003', 'STS_005', 'STS_011', 'STS_012', 'STS_015', 'STS_020', 'STS_021', 'STS_022', 'STS_023',
               'STS_024', 'STS_029', 'STS_031', 'STS_034', 'STS_037', 'STS_039', 'STS_041', 'STS_048']
df = pd.DataFrame()
for (name, values) in zip(prop_names, prop_values):
    df_row = []
    for idx in prop_indices:
        if idx == -1:
            df_row.append(float("nan"))
        else:
            df_row.append(values[idx])
    df[name] = df_row
# 0,0,0,0,1,0,1,1,1,1,0,0,1,1,1,1,0,0
df['Recurrence'] = ['No', 'No', 'No', 'No', 'Yes', 'No', 'Yes', 'Yes', 'Yes', 'Yes', 'No', 'No', 'Yes', 'Yes', 'Yes', 'Yes', 'No', 'No']
df.to_csv('Patient_analysis.csv')
```

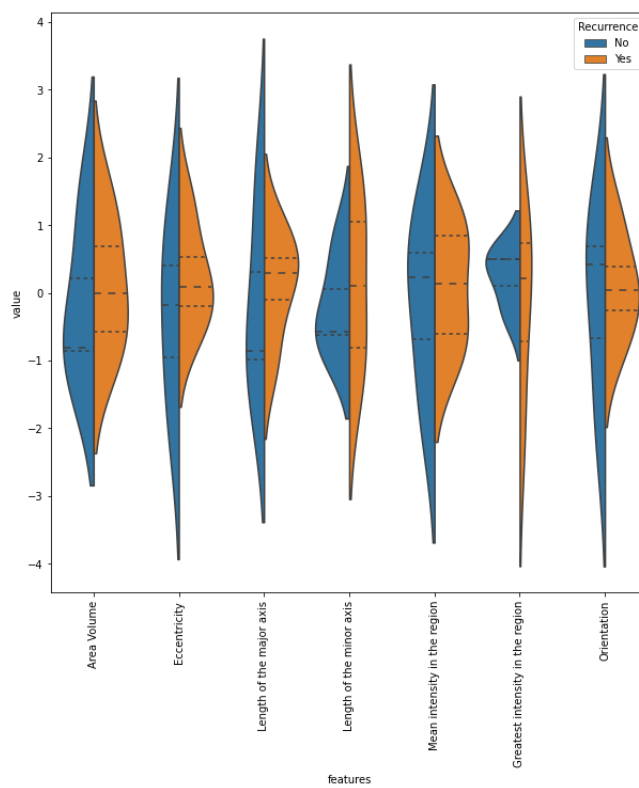
Slika. 3.7: *Spremanje rezultata u excel tablicu*

Na osnovu spremljenoga, možemo započeti s analizom. Vizualizirati ćemo koristeći se box plot i violin plotom. Box plot prikazuje distribuciju kvantitativnih podataka na način koji olakšava usporedbe između varijabli ili između razina kategoričke varijable. Okvir prikazuje kvartile skupa podataka dok se whisker proteže kako bi prikazali ostatak distribucije, osim točaka za koje je utvrđeno da su izuzeci pomoću metode koja je funkcija interkvartilnog raspona. Za razliku od boxplota, u kojem sve komponente dijagrama odgovaraju stvarnim podatkovnim točkama, violinplot sadrži procjenu gustoće jezgre osnovne distribucije. On prikazuje distribuciju kvantitativnih podataka na nekoliko razina jedne (ili više) kategoričkih varijabli tako da se te distribucije mogu usporediti. Dolje se nalazi i primjer histograma, on nam daje jednostavan prikaz frekvencija podataka u uzorku, što može biti vrlo korisno za naslućivanje distribucije uzorka (slika 3.10).

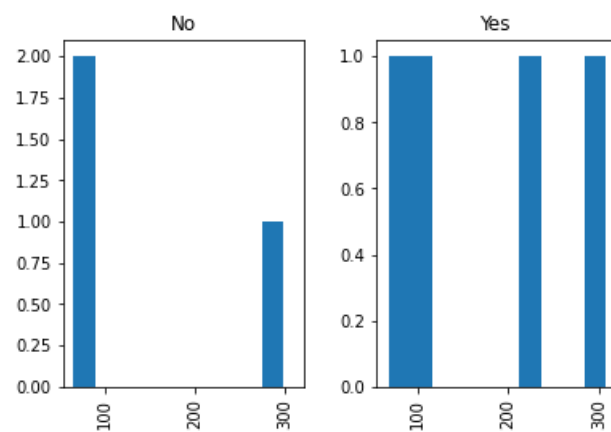
Narančastom bojom su označeni uzorci gdje se rak vratio, plavo gdje nije. Podatci su normalizirani, radi lakše usporedbe.



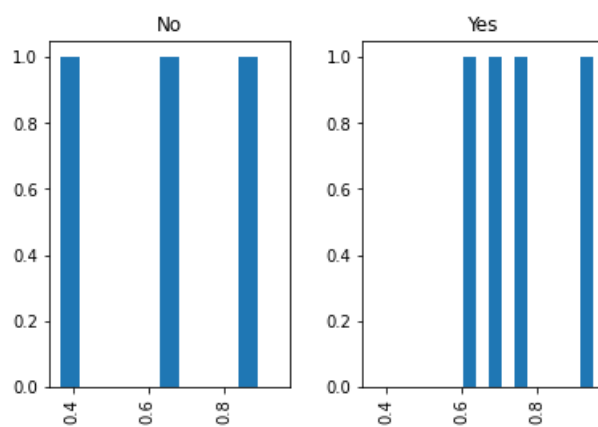
Slika. 3.8: *Boxplot ispis*



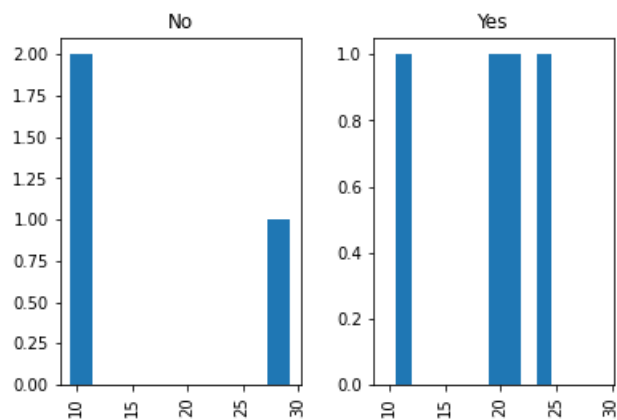
Slika. 3.9: *Violinplot ispis*



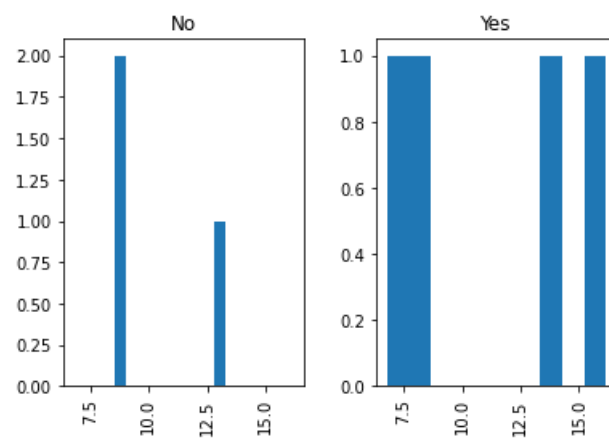
Slika. 3.10: Histogram za parametar volumena regije



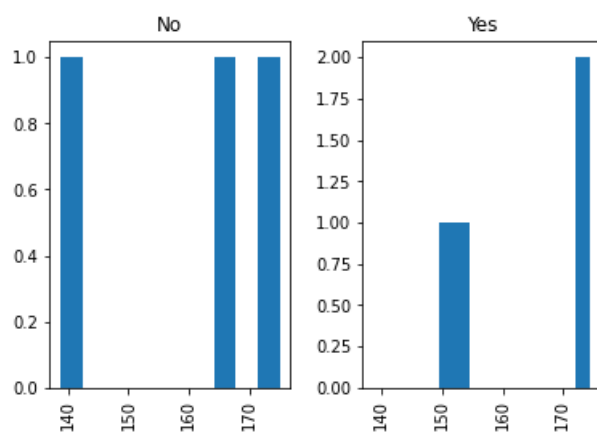
Slika. 3.11: Histogram za parametar ekscentričnosti regije



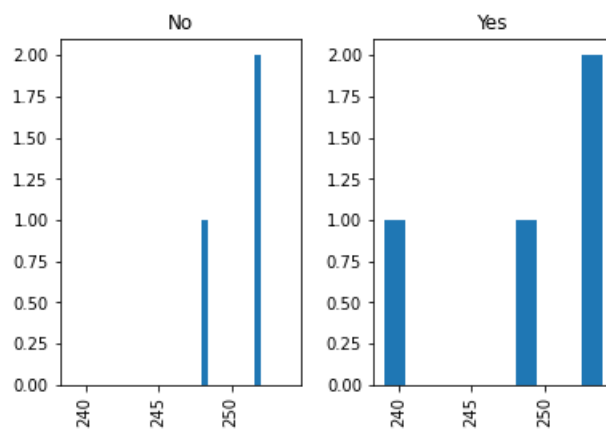
Slika. 3.12: Histogram za parametar duljina velike osi



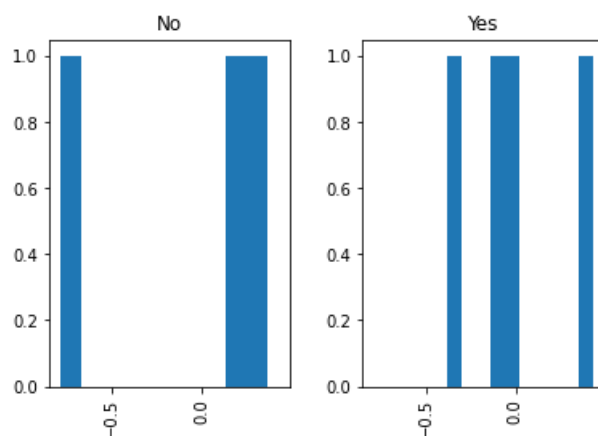
Slika. 3.13: Histogram za parametar duljina male osi



Slika. 3.14: Histogram za parametar prosječnog intenziteta



Slika. 3.15: Histogram za parametar najvećeg intenziteta



Slika. 3.16: *Histogram za parametar orijentacije*

Pobližom analizom grafova na dostupnome uzorku pacijenata, nasljućujemo da volumen raka, ekscentricitet, duljina velike i male osi regije potencijalno utječu na povrat raka. Naime, uočavamo da se veće vrijednosti tih varijabli pojavljuju kod pacijenata kojima se vratio rak. Njihove medijan vrijednosti su također veće u odnosu na pacijente kojima se nije vratio rak. Preostale varijable (prosječni, najveći intenzitet i orijentacija) imaju veće medijane za pacijente kojima se nije vratio rak. Što nije u suglasju za znanošću, koja tvrdi da je rak kako metastazira veći i višljeg intenziteta. Napomena da se varijable promatraju na malome uzorku, što nije optimalno.

4. ZAKLJUČAK

U ovome radu je cilj bio opisati problem analize sarkoma. Najprije smo morali izvući relevantnu regiju sa pet/ct slika. Mjerenja smo pomno radili s ciljem da analiziramo kako bi koja varijabla mogla utjecati na konačan ishod pacijenta (rak se vratio ili nije). Zaključujemo da potencijalno varijable volumena raka, ekscentriteta, te duljine velike i male osi regije utječu na ishod. Varijable su ispitane na samo sedam pacijenata, vjerujem kako bi zaključci bili relevantniji i precizniji na većem uzorku. S time smo ispitili početnu tvrdnju i sam zadatak projekta. Ovaj rad je autoru služio za bolje učenje i razumijevanje obrade slike i statističke analize.

LITERATURA

- [1] Definicija sarkoma. <https://www.nhs.uk/conditions/soft-tissue-sarcoma/>. Pristup: 14.2.2023.
- [2] Skup podataka. <https://www.kaggle.com/datasets/4quant/soft-tissue-sarcoma/>. Pristup: 14.2.2023.
- [3] Definicija grayscale. <https://en.wikipedia.org/wiki/Grayscale>. Pristup: 14.2.2023.
- [4] Sanja Varošaneć. *Matematika 3*. 2020.

PRILOG

Kompletno programsko rješenje zajedno s prezentacijom i dokumentacijom, te svim slikama je dostupno na sljedećoj poveznici: https://github.com/kokanovick/obrada_slike. Ispod se nalazi ispis napisanog Python koda:

```
1  import numpy as np
2  import pandas as pd
3  import matplotlib.pyplot as plt
4  import os
5  import h5py
6  import cv2
7  from PIL import Image, ImageChops
8  from skimage import measure, img_as_ubyte, io
9  import seaborn as sns
10 from skimage.filters import threshold_otsu
11 from skimage.color import rgb2gray
12
13 study_df =pd.read_csv(os.path.join('study_list.csv'))
14
15 def traverse_datasets(hdf_file):
16     def h5py_dataset_iterator(g, prefix=''):
17         for key in g.keys():
18             item =g[key]
19             path =f'{prefix}/{key}'
20             if isinstance(item, h5py.Dataset):
21                 yield (path, item)
22             elif isinstance(item, h5py.Group):
23                 yield from h5py_dataset_iterator(item, path)
24     for path, _ in h5py_dataset_iterator(hdf_file):
25         yield path
26 with h5py.File('lab_petct_vox_5.00mm.h5', 'r') as f:
27     for dset in traverse_datasets(f):
28         print('Path:', dset)
29         print('Shape:', f[dset].shape)
30         print('Data type:', f[dset].dtype)
31
32 with h5py.File(os.path.join('lab_petct_vox_5.00mm.h5'), 'r') as p_data:
33     ct_images =p_data['ct_data'].items()
34     pet_images =p_data['pet_data'].values()
35     lab_images =p_data['label_data'].values()
36     fig, sb_mat =plt.subplots(7, 4, figsize=(10,25))
```

```

37 (axis, ax2s, ax3s, ax4s) = sb_mat.T
38 for c_ax1, c_ax2, c_ax3, c_ax4, (p_id, ct_img), pet_img, lab_img in zip(ax1s, ax2s,
                                ↪ ax3s, ax4s, ct_images, pet_images,
                                ↪ lab_images):
39
40     ct_image = np.mean(ct_img, 1)[::-1]
41     c_ax1.imshow(ct_image, cmap = 'bone')
42     c_ax1.axis('off')
43
44     pet_image = np.sqrt(np.max(pet_img, 1).squeeze()[::-1,:])
45     c_ax2.imshow(pet_image, cmap = 'bone')
46     c_ax2.axis('off')
47
48     label = np.mean(lab_img, 1)[::-1]
49     c_ax3.imshow(label, cmap = 'bone')
50     c_ax3.axis('off')
51
52     result = cv2.multiply(pet_image, label)
53     c_ax4.imshow(result, cmap = 'bone')
54     c_ax4.axis('off')
55
56 for x in range(sb_mat.shape[0]):
57     for y in range(sb_mat.shape[1]):
58         extent = sb_mat[x, y].get_window_extent().transformed(fig.dpi_scale_trans.
                                ↪ inverted())
59         fig.savefig('images/' + str(x) + str(y) + '.png', bbox_inches=extent.expanded(1.
                                ↪ 1, 1.2))
60
61 def trim(im):
62     bg = Image.new(im.mode, im.size, im.getpixel((0,0)))
63     diff = ImageChops.difference(im, bg)
64     diff = ImageChops.add(diff, diff, 2.0, -100)
65     bbox = diff.getbbox()
66     if bbox:
67         return im.crop(bbox)
68     else:
69         return trim(im.convert('RGB'))
70 for i in range(0,7):
71     im = Image.open("images/" + str(i) + "3.png")
72     im = trim(im)

```



```

73     im =im.save("images/Final" +str(i) +".png")
74
75     areas =[]
76     eccentricity =[]
77     axis_maj_l =[]
78     axis_min_l =[]
79     inten_mean =[]
80     orient =[]
81     inten_max =[]
82     for j in range(0,7):
83         test =img_as_ubyte(rgb2gray(io.imread("images/Final" +str(j) +".png")))
84         threshold =threshold_otsu(test)
85         label_image =measure.label(test >threshold, connectivity =test.ndim)
86         props =measure.regionprops_table(label_image, test, properties=['area', '
                                ↪ eccentricity', 'axis_major_length',
                                ↪ 'axis_minor_length', '
                                ↪ intensity_mean', 'intensity_max', '
                                ↪ orientation'])
87         areas.append(props['area'][0])
88         eccentricity.append(props['eccentricity'][0])
89         axis_maj_l.append(props['axis_major_length'][0])
90         axis_min_l.append(props['axis_minor_length'][0])
91         inten_mean.append(props['intensity_mean'][0])
92         inten_max.append(props['intensity_max'][0])
93         orient.append(props['orientation'][0])
94
95     prop_indices =[0, 1, 2, -1, 3, -1, -1, 4, -1, 5, -1, -1, 6, -1, -1, -1, -1]
96     prop_names =['Area Volume', 'Eccentricity', ' Length of the major axis', 'Length of the
                                ↪ minor axis', 'Mean intensity in the
                                ↪ region', ' Greatest intensity in the
                                ↪ region', 'Orientation']
97     prop_values =[areas, eccentricity, axis_maj_l, axis_min_l, inten_mean, inten_max, orient
                                ↪ ]
98     all_patients =['STS_002', 'STS_003', 'STS_005', 'STS_011', 'STS_012', 'STS_015', 'STS_020
                                ↪ ', 'STS_021', 'STS_022', 'STS_023', '
                                ↪ STS_024', 'STS_029', 'STS_031', '
                                ↪ STS_034', 'STS_037', 'STS_039', '
                                ↪ STS_041', 'STS_048']
99     df =pd.DataFrame()
100     for (name, values) in zip(prop_names, prop_values):

```

```

101     df_row = []
102     for idx in prop_indices:
103         if idx == -1:
104             df_row.append(float("nan"))
105         else:
106             df_row.append(values[idx])
107     df[name] = df_row
108     # 0,0,0,0,1,0,1,1,1,1,0,0,1,1,1,1,0,0
109     df['Recurrence'] = ['No', 'No', 'No', 'No', 'Yes', 'No', 'Yes', 'Yes', 'Yes', 'Yes', 'No', 'No', 'Yes', 'Yes', 'Yes', 'Yes', 'No', 'No']
110     df.to_csv('Patient_analysis.csv')
111     y = df.Recurrence
112     x = df.drop('Recurrence', axis=1)
113     data_dia = y
114     data = x
115     data_n_2 = (data - data.mean()) / (data.std())
116     data = pd.concat([y, data_n_2.iloc[:, 0:18]], axis=1)
117     data = pd.melt(data, id_vars="Recurrence", var_name="features", value_name='value')
118     plt.figure(figsize=(10,10))
119     sns.violinplot(x="features", y="value", hue="Recurrence", data=data, split=True, inner="
120                                     ↪ quart")
121     plt.xticks(rotation=90)
122     plt.figure(figsize=(10,10))
123     sns.boxplot(x="features", y="value", hue="Recurrence", data=data)
124     plt.xticks(rotation=90)
125     for name in prop_names:
126         df[name].hist(by = df['Recurrence'], sharex = True)

```