

Assignment1-Part3

September 28, 2021

1 CSC 732 Pattern Recognition and Neural Networks

Instructor: Dr. Natacha Gueorguieva

Contributors:

Robert Kigobe, Aayushi Chirag Thakkar, Nikitha Pulluri

Date: 09-september-2021

QN: Part 3

Seaborn is a library mostly used for statistical plotting in Python. It is built on top of Matplotlib and provides beautiful default styles and color palettes to make statistical plots more attractive.

```
[1]: # increase width of jupyter notebook cells
from IPython.core.display import display, HTML
display(HTML("<style>.container { width:100% !important; }</style>"))

<IPython.core.display.HTML object>
```

2 Load Data

sns not able to load a non demo dataset so we shall use pandas to load the seeds dataset as before

```
[2]: # Load Libraries
import seaborn as sns
import numpy as np
from pandas import read_csv
from pandas import set_option
import warnings
#warnings.simplefilter(action='ignore', category=FutureWarning)

warnings.filterwarnings("ignore")
```

```
[3]: #Import the dataset
dataset = read_csv ('seeds_dataset.csv', header=None)
#Setup the column names

names= ['area','perimeter','compactness', 'kernel length', 'kernel width', '
↳ 'asymmetry coefficient', 'kernel groove Length']
```

```
dataset.columns = names
dataset
```

```
[3]:
```

	area	perimeter	compactness	kernel	length	kernel	width	\
0	15.26	14.84	0.8710		5.763		3.312	
1	14.88	14.57	0.8811		5.554		3.333	
2	14.29	14.09	0.9050		5.291		3.337	
3	13.84	13.94	0.8955		5.324		3.379	
4	16.14	14.99	0.9034		5.658		3.562	
..
205	12.19	13.20	0.8783		5.137		2.981	
206	11.23	12.88	0.8511		5.140		2.795	
207	13.20	13.66	0.8883		5.236		3.232	
208	11.84	13.21	0.8521		5.175		2.836	
209	12.30	13.34	0.8684		5.243		2.974	

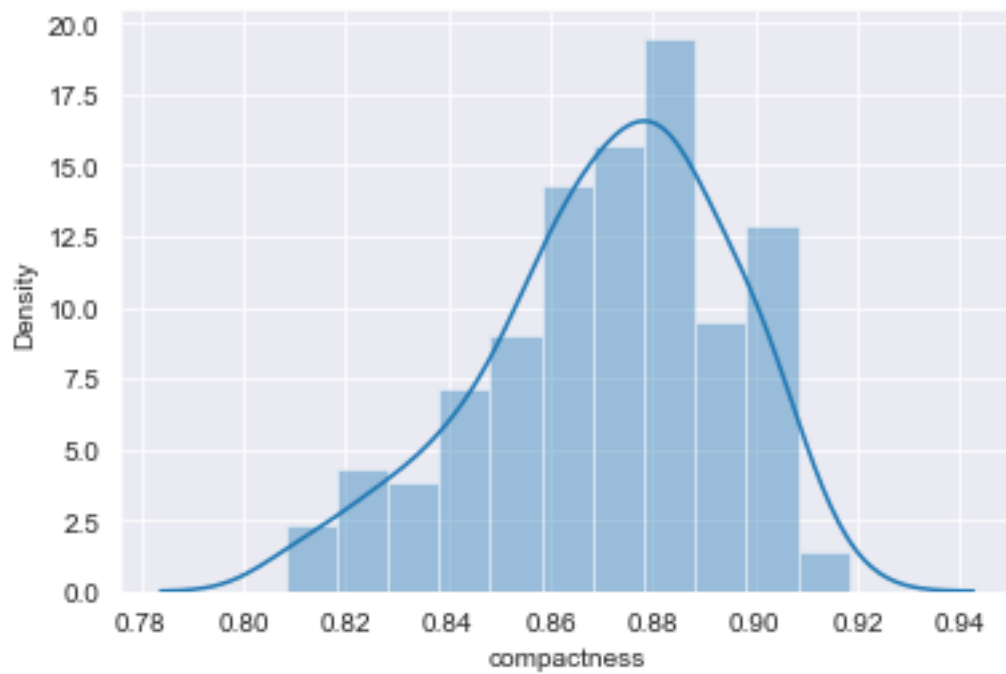
	asymmetry	coefficient	kernel	groove	Length
0		2.221			5.220
1		1.018			4.956
2		2.699			4.825
3		2.259			4.805
4		1.355			5.175
..	
205		3.631			4.870
206		4.325			5.003
207		8.315			5.056
208		3.598			5.044
209		5.637			5.063

[210 rows x 7 columns]

3 Desity Plots

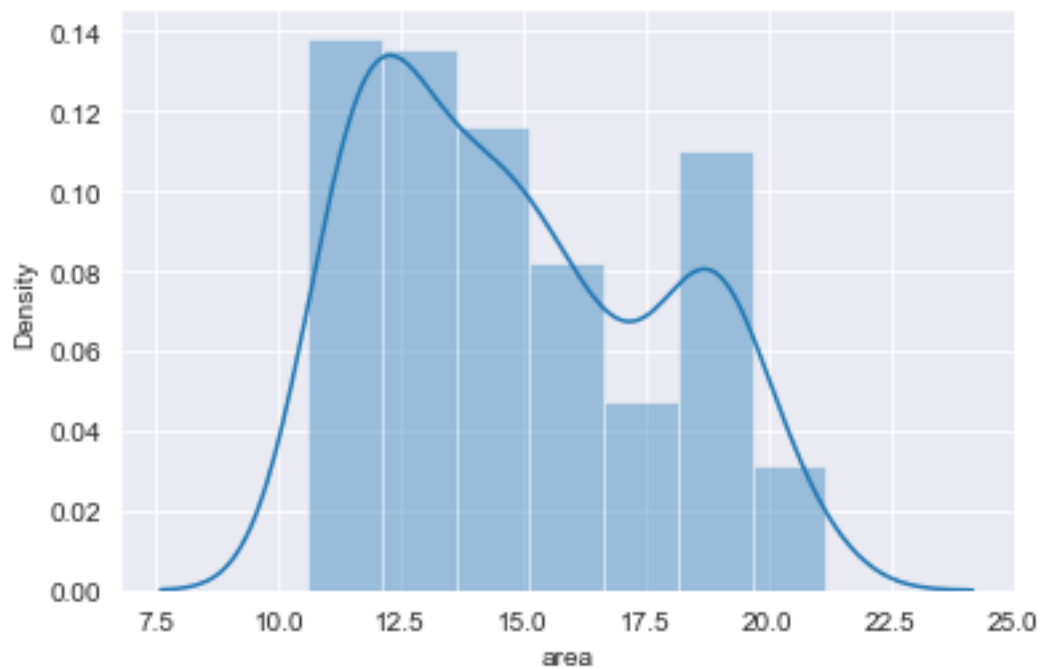
```
[4]: #plot the distribution of the DataFrame "Profit" column
#set the style we wish to use for our plots
sns.set_style("darkgrid")
sns.distplot(dataset['compactness'])
```

```
[4]: <AxesSubplot:xlabel='compactness', ylabel='Density'>
```

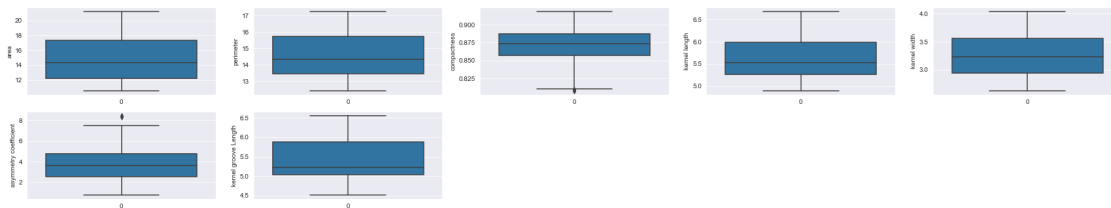


```
[5]: sns.distplot(dataset['area'])
```

```
[5]: <AxesSubplot:xlabel='area', ylabel='Density'>
```



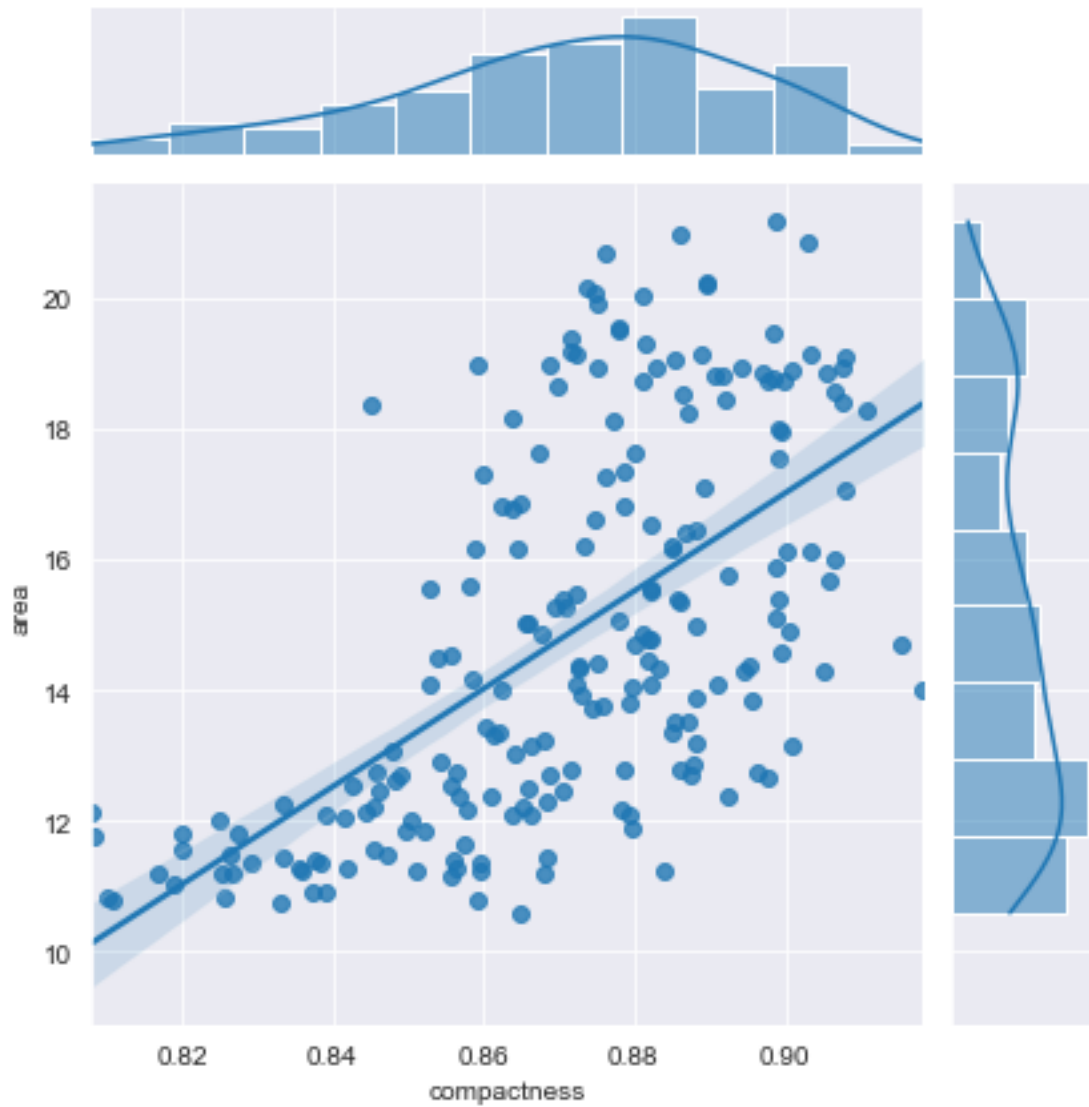
```
[7]: # univariate box and whisker plots done using seaborn
import matplotlib.pyplot as plt
plt.figure(figsize=(30,20))
for i,column in enumerate(dataset.columns):
    try:
        plt.subplot(7,5,1+i)
        sns.boxplot(data=dataset[column], orient='v')
        plt.ylabel(column)
    except:
        pass
```



4 Joint plots

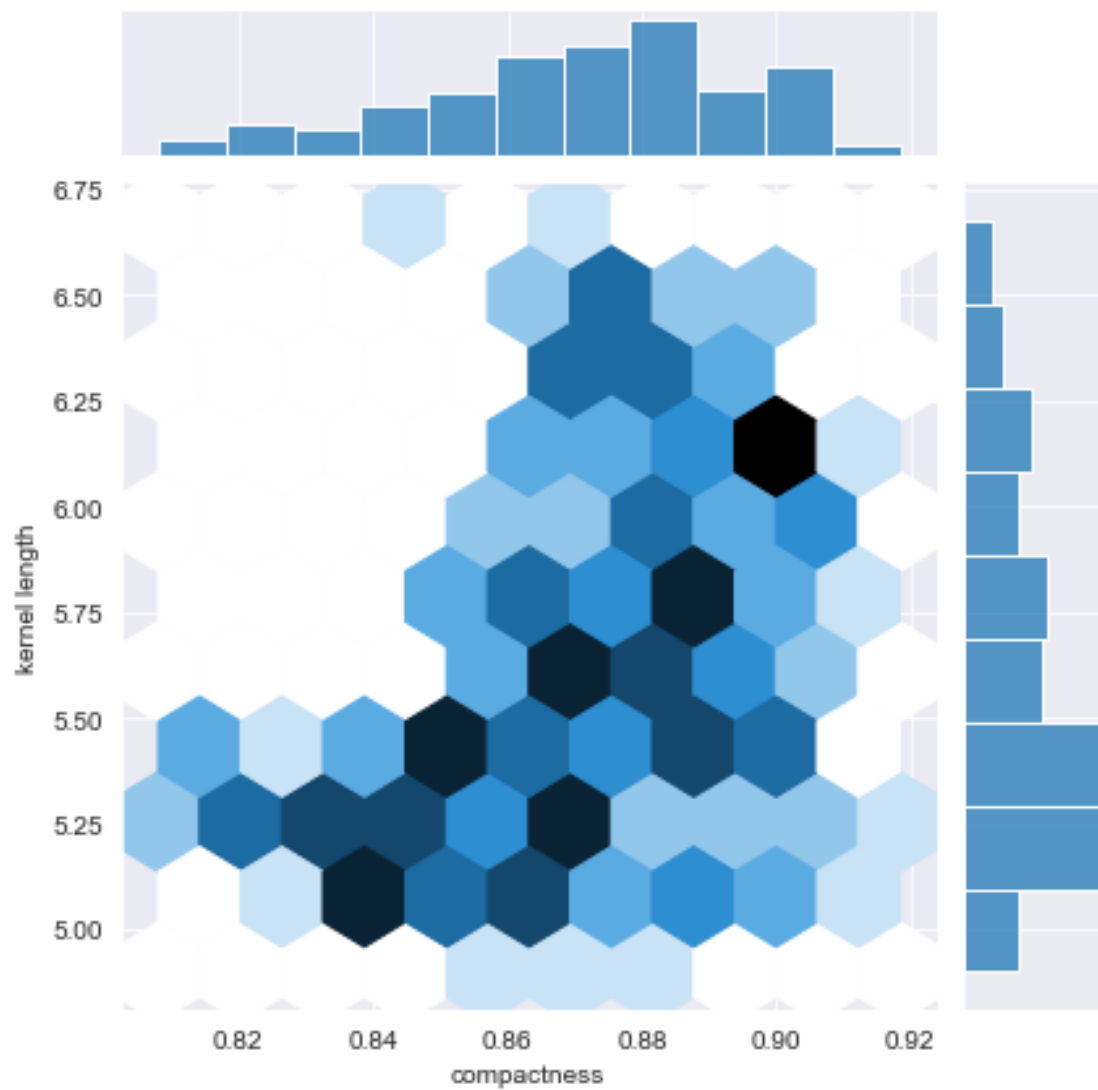
```
[8]: sns.jointplot(x='compactness',y='area',data=dataset, kind='reg')
```

```
[8]: <seaborn.axisgrid.JointGrid at 0x7fea01f16910>
```



```
[9]: sns.jointplot(x='compactness',y='kernel length',data=dataset, kind='hex')
```

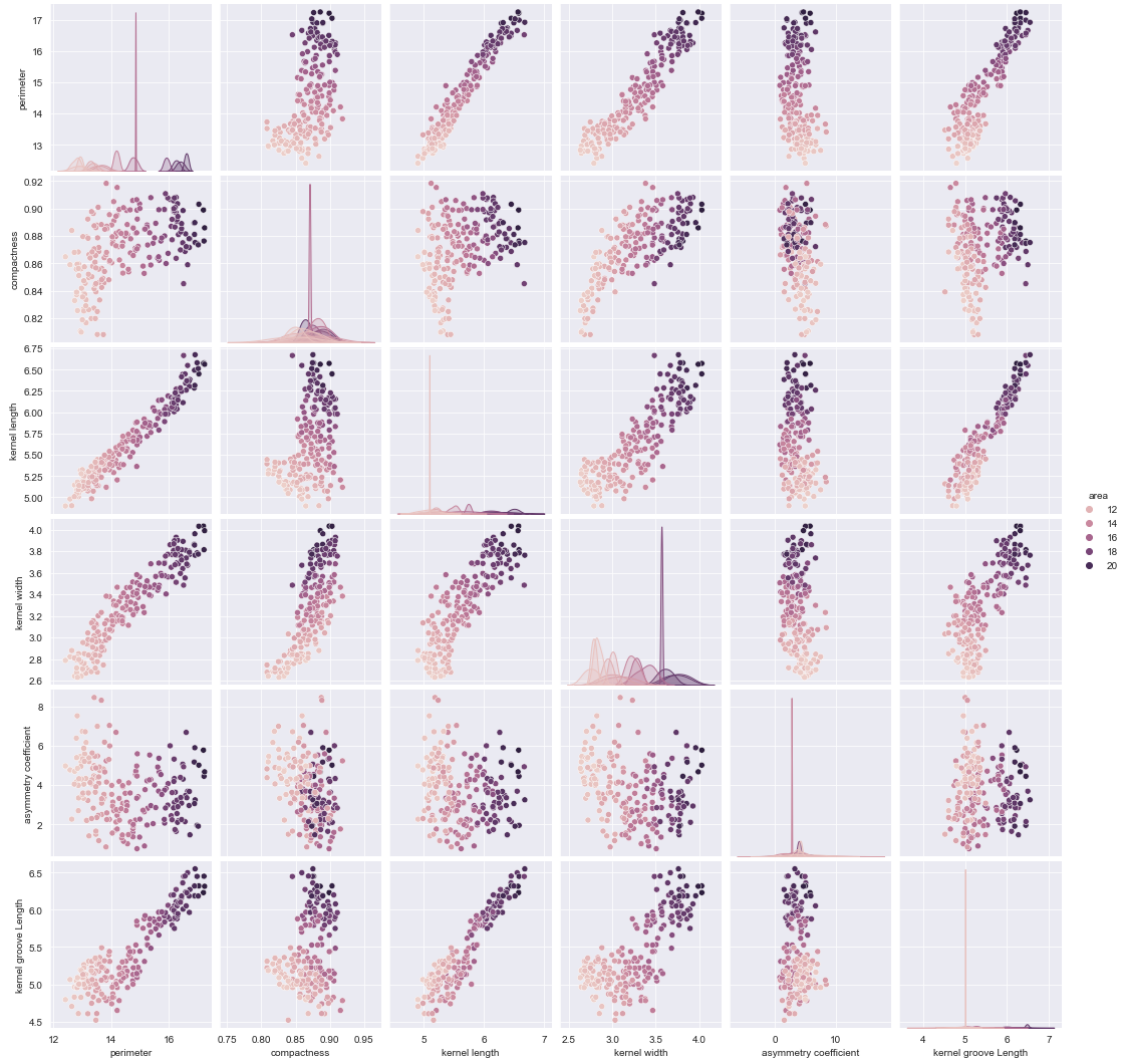
```
[9]: <seaborn.axisgrid.JointGrid at 0x7fea03074820>
```



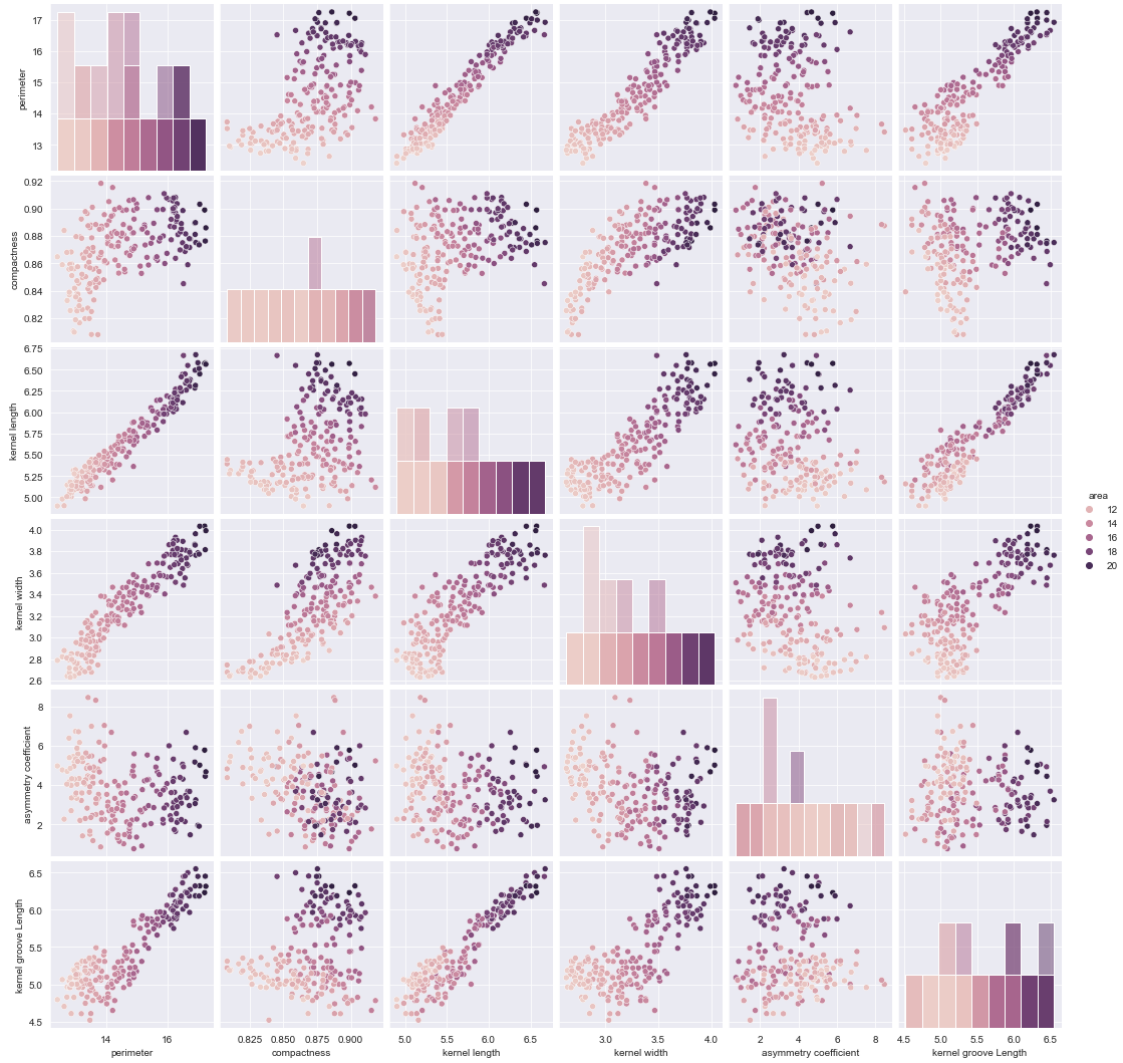
5 Pairplot

```
[10]: sns.pairplot(dataset, hue='area')
```

```
[10]: <seaborn.axisgrid.PairGrid at 0x7fea030663d0>
```

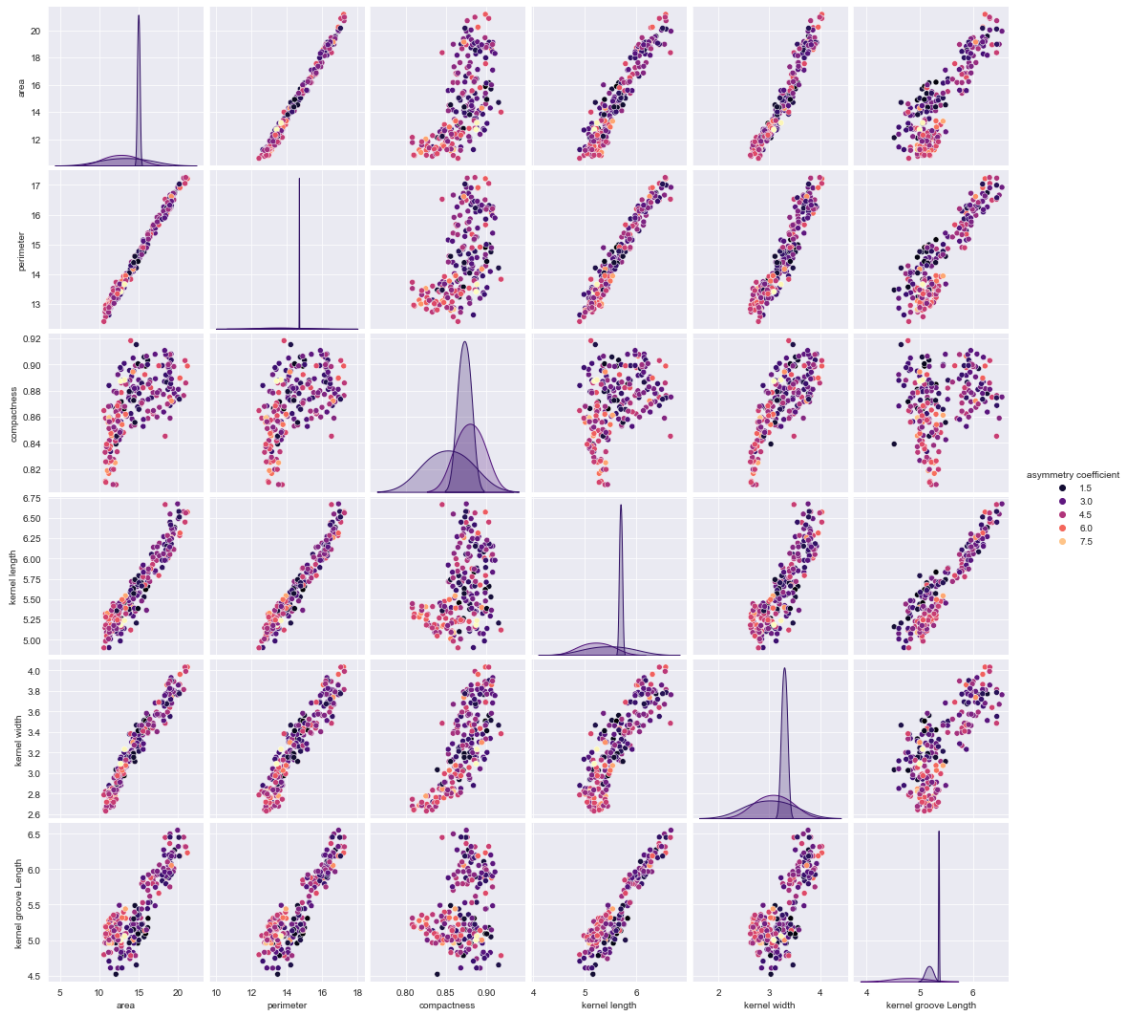


```
[11]: # scatter plot matrix
sns.pairplot(dataset, hue='area', diag_kind='hist')
plt.show()
```

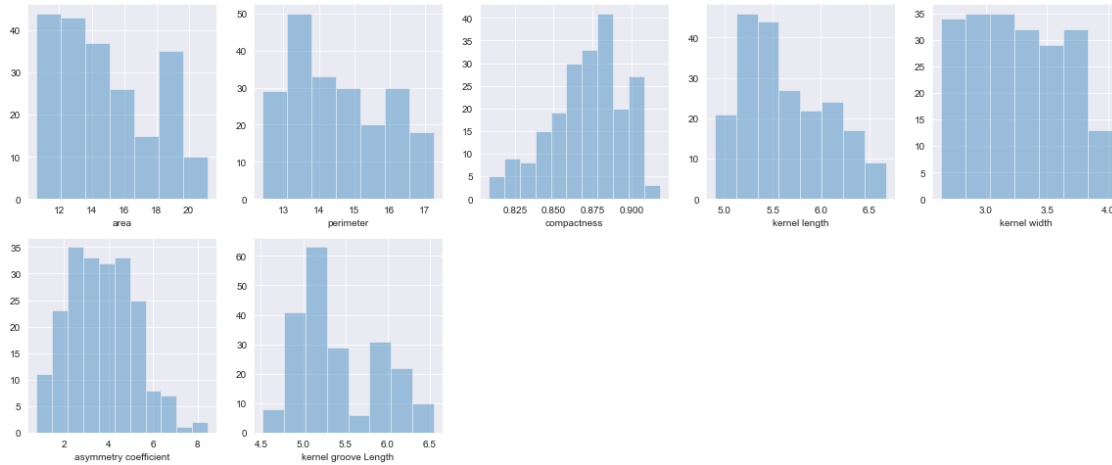


```
[12]: sns.pairplot(dataset,hue='asymmetry coefficient',palette='magma')
```

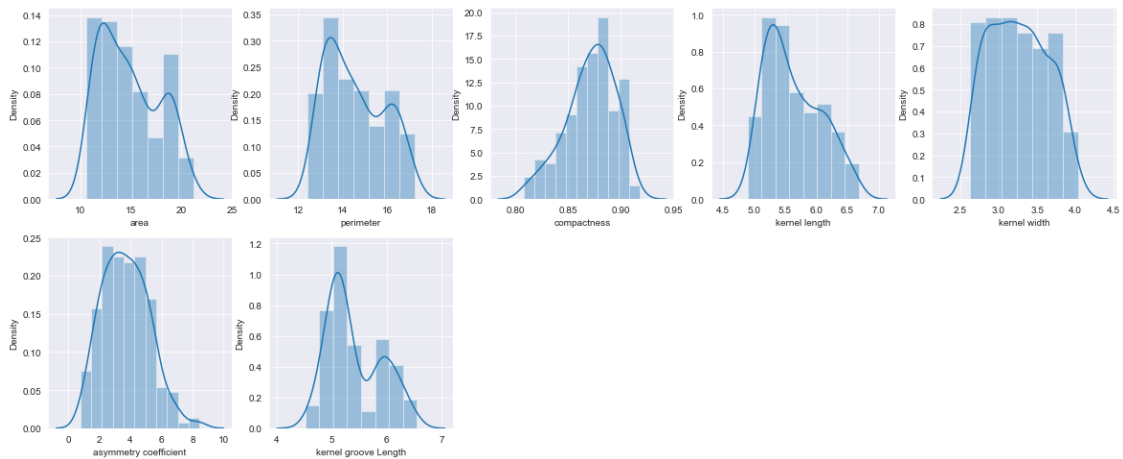
```
[12]: <seaborn.axisgrid.PairGrid at 0x7fea025b9dc0>
```

```
[13]: # histograms
plt.figure(figsize=(20,30))
for i,column in enumerate(dataset.columns):
    try:
        plt.subplot(7,5,1+i)
        sns.distplot(dataset[column], kde=False)
    except:
        pass
```

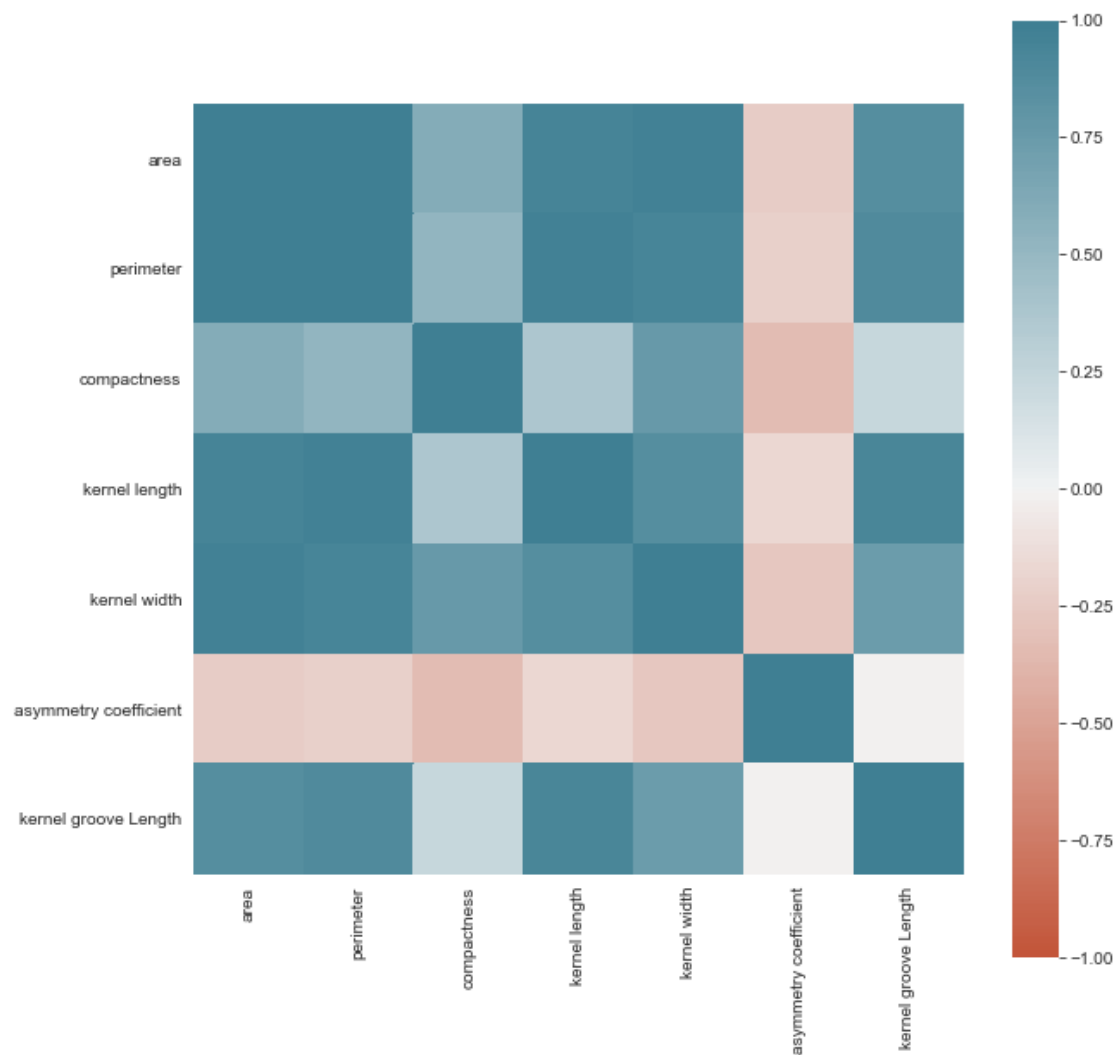


```
[14]: # plot Univariate KDE plots
plt.figure(figsize=(20,30))
for i,column in enumerate(dataset.columns):
    try:
        plt.subplot(7,5,1+i)
        sns.distplot(dataset[column])
    except:
        pass
```



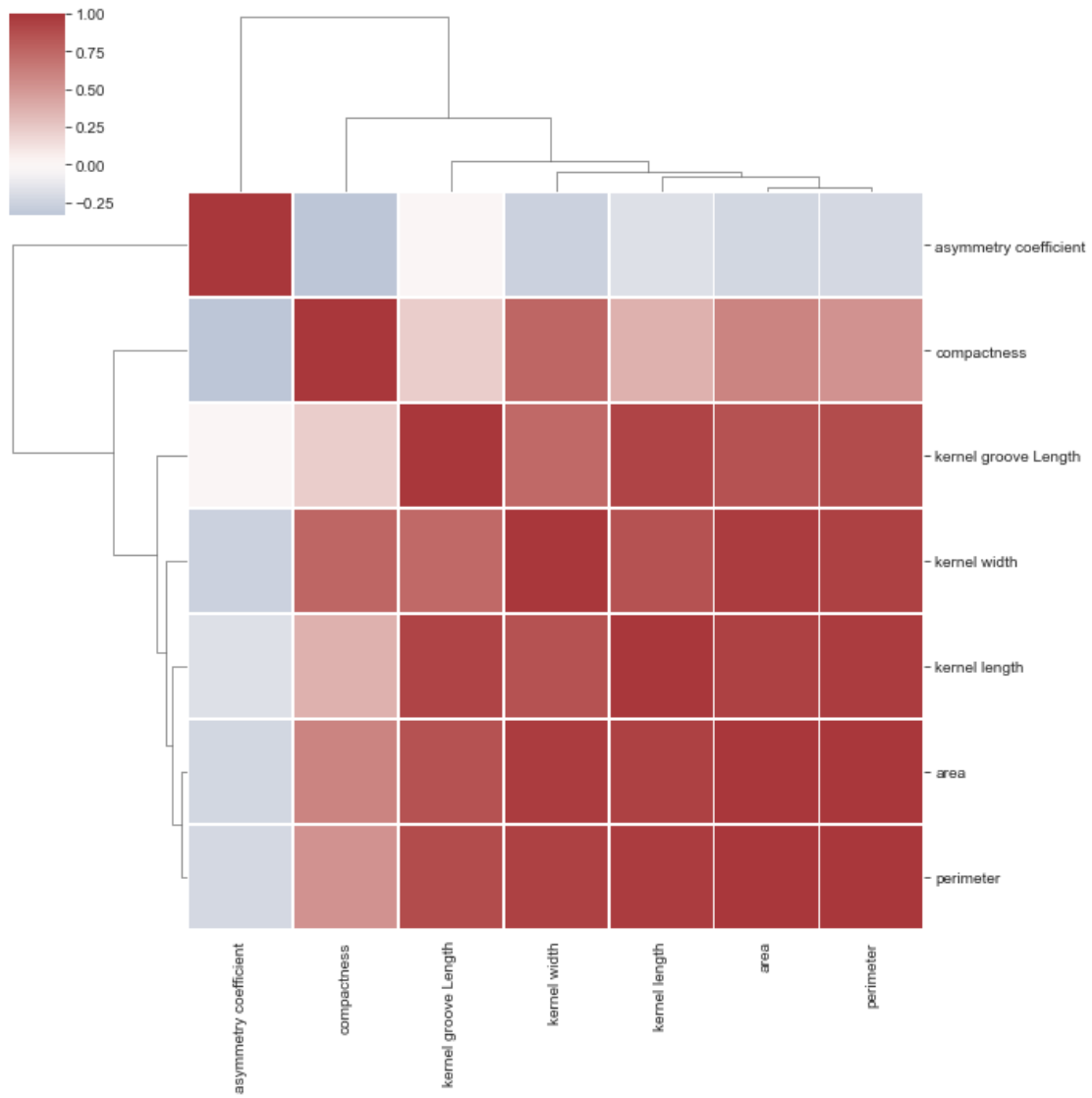
```
[16]: # seaborn code for part 3
plt.figure(figsize=(10,10))
corr = dataset.corr(method='pearson')
ax = sns.heatmap(
    corr,
    vmin=-1, vmax=1, center=0,
```

```
cmap=sns.diverging_palette(20, 220, n=200),
square=True)
```



```
[17]: # plot of hierarchically-clustered heatmap
# better correlation matrix
sns.clustermap(corr, center=0, cmap="vlag",
               linewidths=.75, figsize=(10, 10))
```

```
[17]: <seaborn.matrix.ClusterGrid at 0x7fea024aab20>
```



[]: