

```
In [1]: import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
get_ipython().run_line_magic('matplotlib', 'inline')
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.preprocessing import LabelEncoder
le=LabelEncoder()
```

```
In [2]: df = pd.read_csv("forestfires.csv")
```

```
In [3]: df1 = df.copy()
df1
```

```
Out[3]:
```

	month	day	FFMC	DMC	DC	ISI	temp	RH	wind	rain	...	monthfeb	monthjan	monthjul	monthjun	monthmar	monthmay	mor
0	mar	fri	86.2	26.2	94.3	5.1	8.2	51	6.7	0.0	...	0	0	0	0	1	0	
1	oct	tue	90.6	35.4	669.1	6.7	18.0	33	0.9	0.0	...	0	0	0	0	0	0	
2	oct	sat	90.6	43.7	686.9	6.7	14.6	33	1.3	0.0	...	0	0	0	0	0	0	
3	mar	fri	91.7	33.3	77.5	9.0	8.3	97	4.0	0.2	...	0	0	0	0	1	0	
4	mar	sun	89.3	51.3	102.2	9.6	11.4	99	1.8	0.0	...	0	0	0	0	1	0	
...	
512	aug	sun	81.6	56.7	665.6	1.9	27.8	32	2.7	0.0	...	0	0	0	0	0	0	
513	aug	sun	81.6	56.7	665.6	1.9	21.9	71	5.8	0.0	...	0	0	0	0	0	0	
514	aug	sun	81.6	56.7	665.6	1.9	21.2	70	6.7	0.0	...	0	0	0	0	0	0	
515	aug	sat	94.4	146.0	614.7	11.3	25.6	42	4.0	0.0	...	0	0	0	0	0	0	
516	nov	tue	79.5	3.0	106.7	1.1	11.8	31	4.5	0.0	...	0	0	0	0	0	0	

517 rows × 31 columns

```
In [4]: df1 = df1.drop(["month","day"], axis = 1)
df1.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 517 entries, 0 to 516
Data columns (total 29 columns):
#   Column              Non-Null Count  Dtype
---  -
0   FFMC                517 non-null    float64
1   DMC                 517 non-null    float64
2   DC                  517 non-null    float64
3   ISI                 517 non-null    float64
4   temp                517 non-null    float64
5   RH                  517 non-null    int64
6   wind                517 non-null    float64
7   rain                517 non-null    float64
8   area                517 non-null    float64
9   dayfri              517 non-null    int64
10  daymon              517 non-null    int64
11  daysat              517 non-null    int64
12  daysun              517 non-null    int64
13  daythu              517 non-null    int64
14  daytue              517 non-null    int64
15  daywed              517 non-null    int64
16  monthapr            517 non-null    int64
17  monthaug            517 non-null    int64
18  monthdec            517 non-null    int64
19  monthfeb            517 non-null    int64
20  monthjan            517 non-null    int64
21  monthjul            517 non-null    int64
22  monthjun            517 non-null    int64
23  monthmar            517 non-null    int64
24  monthmay            517 non-null    int64
25  monthnov            517 non-null    int64
26  monthoct            517 non-null    int64
27  monthsep            517 non-null    int64
28  size_category       517 non-null    object
dtypes: float64(8), int64(20), object(1)
memory usage: 117.3+ KB
```

```
In [5]: df.info()
```

```

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 517 entries, 0 to 516
Data columns (total 31 columns):
#   Column                Non-Null Count  Dtype
---  -
0   month                 517 non-null   object
1   day                   517 non-null   object
2   FFMC                  517 non-null   float64
3   DMC                   517 non-null   float64
4   DC                    517 non-null   float64
5   ISI                   517 non-null   float64
6   temp                  517 non-null   float64
7   RH                    517 non-null   int64
8   wind                  517 non-null   float64
9   rain                  517 non-null   float64
10  area                  517 non-null   float64
11  dayfri                517 non-null   int64
12  daymon                517 non-null   int64
13  daysat               517 non-null   int64
14  daysun               517 non-null   int64
15  daythu               517 non-null   int64
16  daytue               517 non-null   int64
17  daywed               517 non-null   int64
18  monthapr             517 non-null   int64
19  monthaug             517 non-null   int64
20  monthdec             517 non-null   int64
21  monthfeb             517 non-null   int64
22  monthjan             517 non-null   int64
23  monthjul             517 non-null   int64
24  monthjun             517 non-null   int64
25  monthmar             517 non-null   int64
26  monthmay             517 non-null   int64
27  monthnov             517 non-null   int64
28  monthoct             517 non-null   int64
29  monthsep             517 non-null   int64
30  size_category        517 non-null   object
dtypes: float64(8), int64(20), object(3)
memory usage: 125.3+ KB

```

```
In [6]: df1.isnull().sum()
```

```

Out[6]: FFMC      0
        DMC       0
        DC        0
        ISI       0
        temp      0
        RH        0
        wind      0
        rain      0
        area      0
        dayfri    0
        daymon    0
        daysat    0
        daysun    0
        daythu    0
        daytue    0
        daywed    0
        monthapr  0
        monthaug  0
        monthdec  0
        monthfeb  0
        monthjan  0
        monthjul  0
        monthjun  0
        monthmar  0
        monthmay  0
        monthnov  0
        monthoct  0
        monthsep  0
        size_category  0
dtype: int64

```

```
In [7]: df1.describe()
```

	FFMC	DMC	DC	ISI	temp	RH	wind	rain	area	dayfri	...	month
count	517.000000	517.000000	517.000000	517.000000	517.000000	517.000000	517.000000	517.000000	517.000000	517.000000	...	517.000
mean	90.644681	110.872340	547.940039	9.021663	18.889168	44.288201	4.017602	0.021663	12.847292	0.164410	...	0.017
std	5.520111	64.046482	248.066192	4.559477	5.806625	16.317469	1.791653	0.295959	63.655818	0.371006	...	0.130
min	18.700000	1.100000	7.900000	0.000000	2.200000	15.000000	0.400000	0.000000	0.000000	0.000000	...	0.000
25%	90.200000	68.600000	437.700000	6.500000	15.500000	33.000000	2.700000	0.000000	0.000000	0.000000	...	0.000
50%	91.600000	108.300000	664.200000	8.400000	19.300000	42.000000	4.000000	0.000000	0.520000	0.000000	...	0.000
75%	92.900000	142.400000	713.900000	10.800000	22.800000	53.000000	4.900000	0.000000	6.570000	0.000000	...	0.000
max	96.200000	291.300000	860.600000	56.100000	33.300000	100.000000	9.400000	6.400000	1090.840000	1.000000	...	1.000

8 rows × 28 columns

```
In [8]: df1["size_category"] = le.fit_transform(df1["size_category"])
```

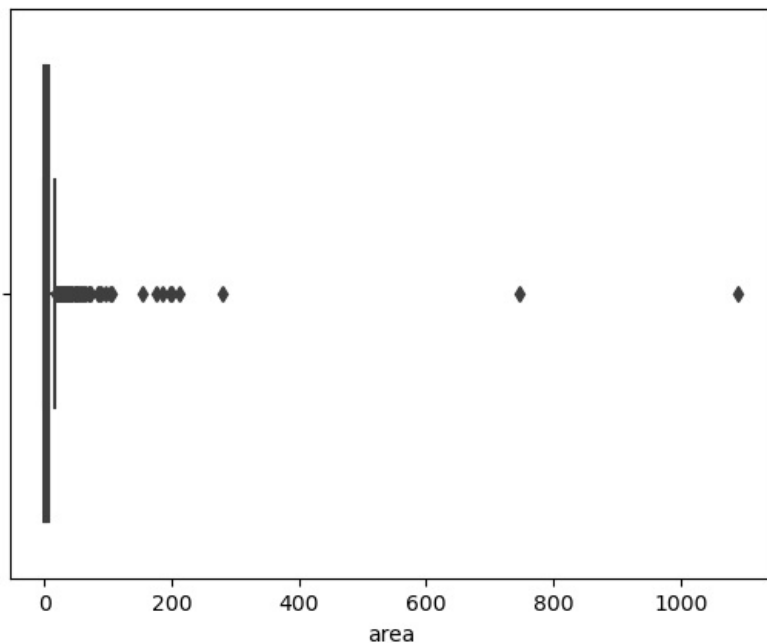
```
In [9]: df1.size_category.value_counts()
```

```
Out[9]: 1    378
        0    139
        Name: size_category, dtype: int64
```

```
In [10]: ax = sns.boxplot(df1['area'])
```

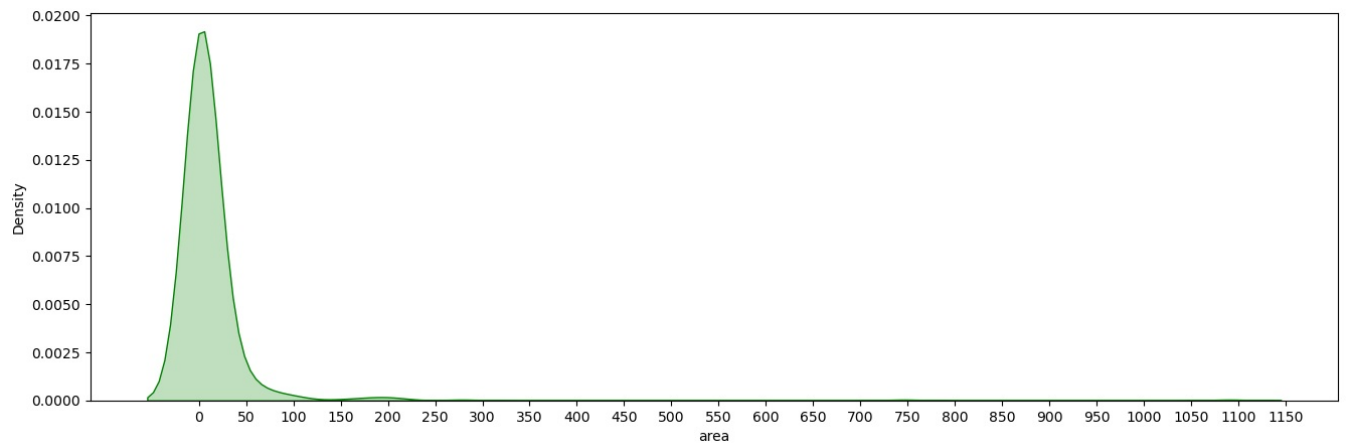
C:\Users\ROHIT\anaconda3\lib\site-packages\seaborn_decorators.py:36: FutureWarning: Pass the following variable as a keyword arg: x. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.

```
warnings.warn(
```

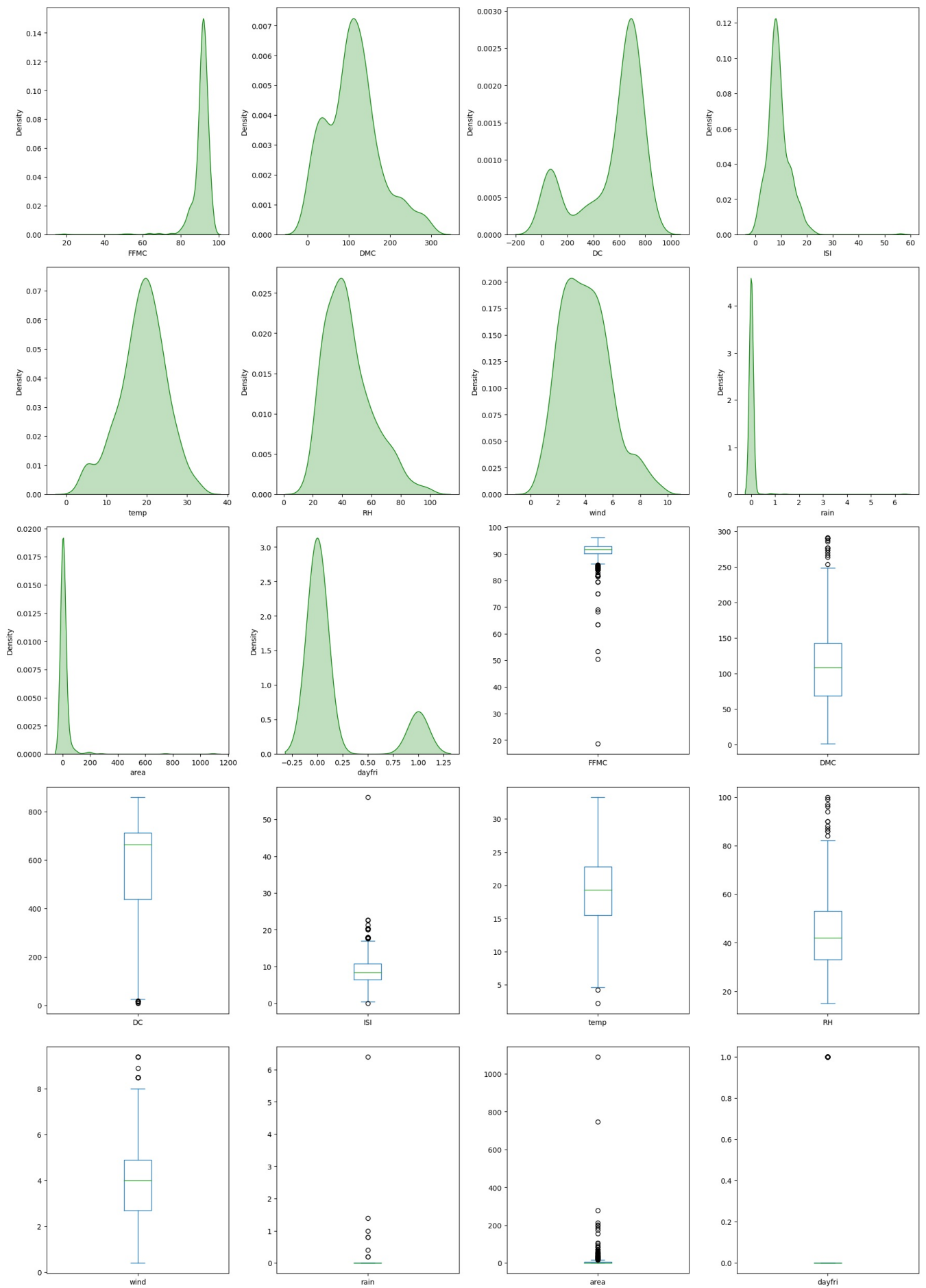


```
In [11]: plt.rcParams["figure.figsize"] = 9,4
plt.figure(figsize=(16,5))
print("Skew: {}".format(df1['area'].skew()))
print("Kurtosis: {}".format(df1['area'].kurtosis()))
ax = sns.kdeplot(df1['area'],shade=True,color='g')
plt.xticks([i for i in range(0,1200,50)])
plt.show()
```

Skew: 12.846933533934868
Kurtosis: 194.1407210942299



```
In [12]: dfa = df1[df1.columns[0:10]]
month_col = dfa.select_dtypes(include='object').columns.tolist()
num_columns = dfa.select_dtypes(exclude='object').columns.tolist()
plt.figure(figsize=(18,40))
for i,col in enumerate(num_columns,1):
    plt.subplot(8,4,i)
    sns.kdeplot(df[col],color='g',shade=True)
    plt.subplot(8,4,i+10)
    df[col].plot.box()
plt.tight_layout()
plt.show()
num_data = df[num_columns]
pd.DataFrame(data=[num_data.skew(),num_data.kurtosis()],index=['skewness','kurtosis'])
```



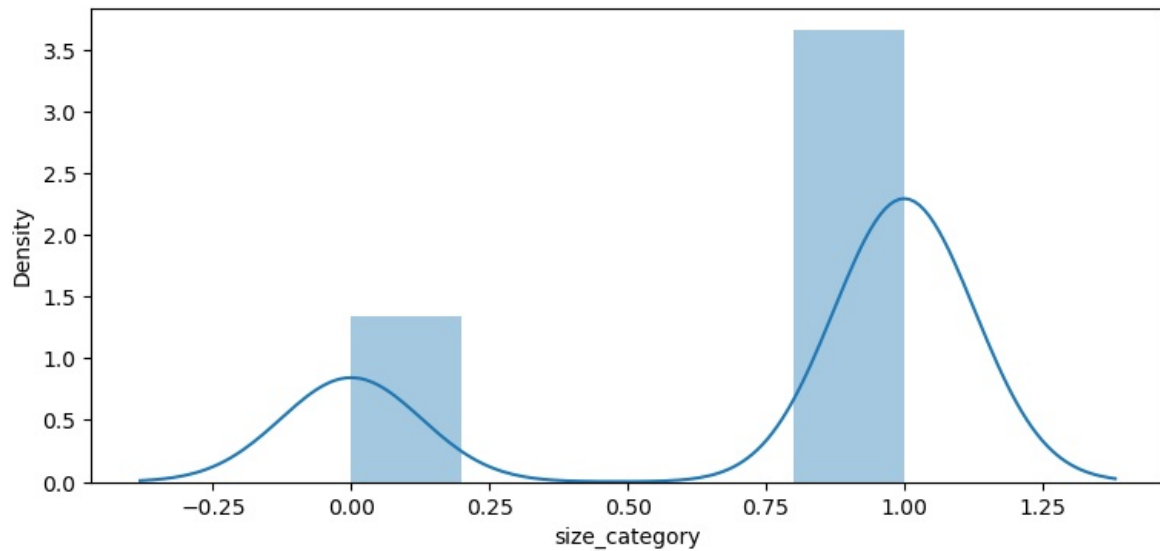
Out[12]:

	FFMC	DMC	DC	ISI	temp	RH	wind	rain	area	dayfri
skewness	-6.575606	0.547498	-1.100445	2.536325	-0.331172	0.862904	0.571001	19.816344	12.846934	1.816105
kurtosis	67.066041	0.204822	-0.245244	21.458037	0.136166	0.438183	0.054324	421.295964	194.140721	1.303265

In [13]: sns.distplot(df1.size_category)

C:\Users\R0HIT\anaconda3\lib\site-packages\seaborn\distributions.py:2619: FutureWarning: `distplot` is a deprecated function and will be removed in a future version. Please adapt your code to use either `displot` (a figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).
 warnings.warn(msg, FutureWarning)

Out[13]: <AxesSubplot:xlabel='size_category', ylabel='Density'>



In [14]: x = df1.iloc[:,0:28]
 y = df1.iloc[:,28]

In [15]: x

Out[15]:

	FFMC	DMC	DC	ISI	temp	RH	wind	rain	area	dayfri	...	monthdec	monthfeb	monthjan	monthjul	monthjun	monthmar	mo
0	86.2	26.2	94.3	5.1	8.2	51	6.7	0.0	0.00	1	...	0	0	0	0	0	1	
1	90.6	35.4	669.1	6.7	18.0	33	0.9	0.0	0.00	0	...	0	0	0	0	0	0	
2	90.6	43.7	686.9	6.7	14.6	33	1.3	0.0	0.00	0	...	0	0	0	0	0	0	
3	91.7	33.3	77.5	9.0	8.3	97	4.0	0.2	0.00	1	...	0	0	0	0	0	1	
4	89.3	51.3	102.2	9.6	11.4	99	1.8	0.0	0.00	0	...	0	0	0	0	0	1	
...	
512	81.6	56.7	665.6	1.9	27.8	32	2.7	0.0	6.44	0	...	0	0	0	0	0	0	
513	81.6	56.7	665.6	1.9	21.9	71	5.8	0.0	54.29	0	...	0	0	0	0	0	0	
514	81.6	56.7	665.6	1.9	21.2	70	6.7	0.0	11.16	0	...	0	0	0	0	0	0	
515	94.4	146.0	614.7	11.3	25.6	42	4.0	0.0	0.00	0	...	0	0	0	0	0	0	
516	79.5	3.0	106.7	1.1	11.8	31	4.5	0.0	0.00	0	...	0	0	0	0	0	0	

517 rows × 28 columns

In [16]: y

Out[16]:

0	1
1	1
2	1
3	1
4	1
...	...
512	0
513	0
514	0
515	1
516	1

Name: size_category, Length: 517, dtype: int32

In [17]: x_train, x_test, y_train, y_test = train_test_split(x, y, test_size = 0.3, stratify = y)
 scaled_values = StandardScaler()
 scaled_values.fit(x_train)

Out[17]: StandardScaler()

In [18]: x_train = scaled_values.transform(x_train)
 x_test = scaled_values.transform(x_test)

In [19]: from sklearn.neural_network import MLPClassifier

In [20]: mlp = MLPClassifier(hidden_layer_sizes = (10,10))

