MACHINE LEARNING

(Introduction to Data Visualisation & EDA)

Submitted by,

- Name:Jagannath v v

- Reg.No.-21122024

- class:1st MSc DataScience

LAB OVERVIEW

OBJECTIVES

Questions: Part A) Export the final DataFrame that you have created using pd.to_csv method to a file whose name is regno.csv. Load that DataFrame to another a new notebook named Lab2.ipynb.

or

Create a dataset with 12 features and 3 classes using make_classification. Name the features F1, F2, ... upto F12. F1: Can hold two values (True/False) F2: Can hold two values (Type 1 or Type 2) F3: Can hold three values (A or B or C) F4: Can hold three values (HIGH / MEDIUM / LOW) Features F5 to F12 will have numeric values generated during the creation process.

Part B) Use either of the above datasets to show the usage of distplot, jointplot, pairplot, rugplot, catplot, barplot, countplot, violinplot, striplot, swarmplot and facetgrid plots.

Part C) Load the Iris Dataset, and explore - scatterplot, scatter_3d, heatmap, boxplot, kdeplot etc. In both the cases, write your observations on the plot outputs and how it is relevant.

PROBLEM DEFINITION

1.we need to create a dataset using sklearn make_classification and visualize the data in it using different plots

2.load and visualize the iris data

APPROACH

Here i reffered internet to find some more about the libraries and visualisation tools/functions

PART A

(Q)Export the final DataFrame that you have created using pd.to_csv method to a file whose name is regno.csv. Load that DataFrame to another a new notebook named Lab2.ipynb.

or

(Q)Create a dataset with 12 features and 3 classes using make_classification. Name the features F1, F2, ... upto F12. F1: Can hold two values (True/False) F2: Can hold two values (Type 1 or Type 2) F3: Can hold three values (A or B or C) F4: Can hold three values (HIGH / MEDIUM / LOW) Features F5 to F12 will have numeric values generated during the creation process.

Importing some LIbraries

In [91]:
```
import sklearn as sk
import pandas as pd
from IPython.display import display
import numpy as np
```

In [95]:
```
from sklearn.datasets import make_classification
X, y = make_classification(n_samples=150, n_classes=3,
                           n_features=12, n_informative=3,
                           random_state=0)
```

In [120…
```
df=pd.DataFrame(X,columns=["F1","F2","F3","F4","F5","F6","F7","F8","F9","F10","F11",
df
```

Out[120…

	F1	F2	F3	F4	F5	F6	F7	F8	F9
0	0.566637	0.608894	-0.149934	-1.759827	2.821206	0.508474	0.235334	0.040168	-1.255969
1	-0.758437	-0.185108	0.261174	0.423278	-0.959230	-0.420025	-1.784105	0.754800	-0.086484
2	0.447970	-1.875646	2.266469	0.568155	-1.879125	-1.376408	0.666579	-0.774138	0.388498
3	-0.991936	-1.209257	1.216870	-0.046880	-1.596464	-0.298444	-0.014444	-0.070094	-0.743219
4	0.034821	-0.883901	0.822239	0.000440	-0.434399	-1.089371	0.750761	-1.017047	0.161205
...	...	...	...	...	...	...	...	...	...
145	-0.097108	-1.669714	1.660110	0.193479	-0.553132	-0.517311	0.055667	-0.311303	0.576733
146	2.600879	0.511898	-0.476120	-0.304736	-0.950497	-1.226141	-1.151466	-0.143223	-0.583135
147	-0.861744	-1.837153	2.521959	-0.984859	-1.799240	-0.687565	0.875093	-0.346702	-0.418724
148	-1.589788	0.393243	-0.021997	0.167865	1.464969	0.296061	0.116047	1.861252	-0.926909
149	0.666418	-2.289063	2.741261	-0.191984	-1.990926	-0.736162	0.280526	1.848671	0.423368

150 rows × 12 columns

In [121…
```
df['F1']=np.where(df['F1']>0,"TRUE",'FALSE')
df['F2']=np.where(df['F2']>0,"TYPE1","TYPE2")
```

```
df['F3']=np.where(df['F3']<0,"A",np.where(df['F3']>0,"B","C"))
df['F4']=np.where(df['F4']<0,"LOW",np.where(df['F4']>10,"MEDIUM","HIGH"))
```

In [122…

```
df
```

Out[122…

	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	
0	TRUE	TYPE1	A	LOW	2.821206	0.508474	0.235334	0.040168	-1.255969	0.999328	-1.89
1	FALSE	TYPE2	B	HIGH	-0.959230	-0.420025	-1.784105	0.754800	-0.086484	-1.396005	1.14
2	TRUE	TYPE2	B	HIGH	-1.879125	-1.376408	0.666579	-0.774138	0.388498	0.035599	1.96
3	FALSE	TYPE2	B	LOW	-1.596464	-0.298444	-0.014444	-0.070094	-0.743219	-1.805269	1.19
4	TRUE	TYPE2	B	HIGH	-0.434399	-1.089371	0.750761	-1.017047	0.161205	-1.398043	-0.09
…	…	…	…	…	…	…	…	…	…	…	…
145	FALSE	TYPE2	B	HIGH	-0.553132	-0.517311	0.055667	-0.311303	0.576733	-0.945930	-0.24
146	TRUE	TYPE1	A	LOW	-0.950497	-1.226141	-1.151466	-0.143223	-0.583135	-0.379565	1.40
147	FALSE	TYPE2	B	LOW	-1.799240	-0.687565	0.875093	-0.346702	-0.418724	0.711366	2.50
148	FALSE	TYPE1	A	HIGH	1.464969	0.296061	0.116047	1.861252	-0.926909	-0.876158	-0.67
149	TRUE	TYPE2	B	LOW	-1.990926	-0.736162	0.280526	1.848671	0.423368	1.177682	2.00

150 rows × 12 columns

PART-B

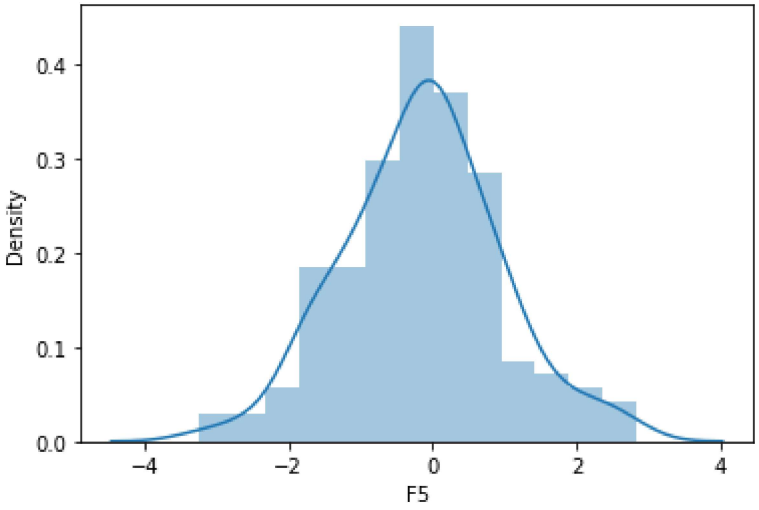
(Q)Use either of the above datasets to show the usage of distplot, jointplot, pairplot, rugplot, catplot, barplot, countplot, violinplot, striplot, swarmplot and facetgrid plots.

(1)Distplot

The distplot represents the univariate distribution of data i.e. data distribution of a variable against the density distribution. The seaborn. distplot() function accepts the data variable as an argument and returns the plot with the density distribution.

In [125…

```
ax = sn.distplot(df.F5)
```



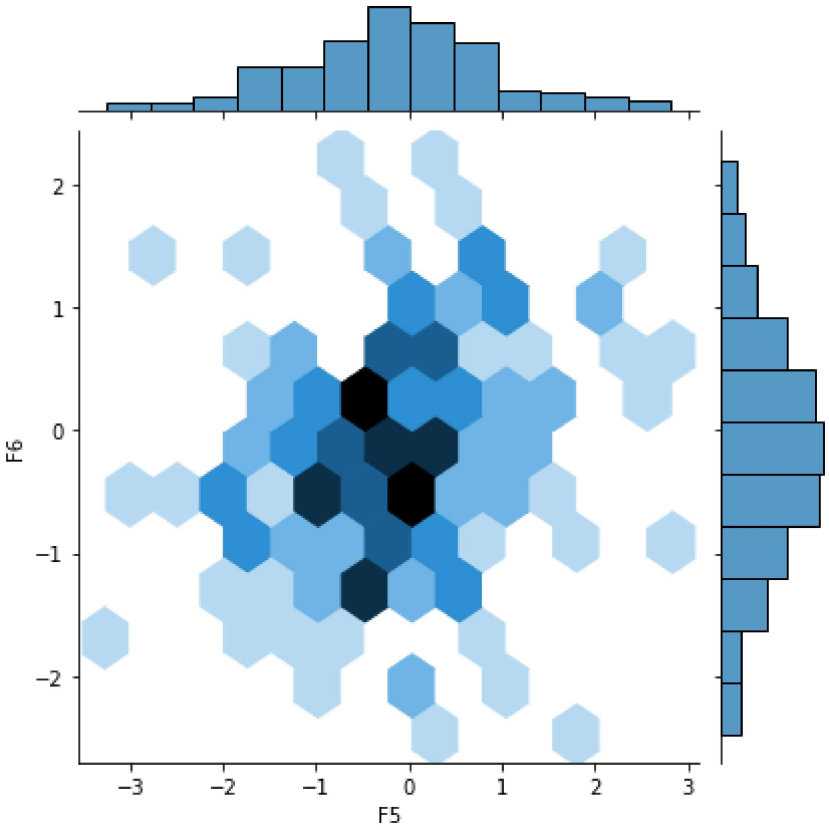
(2)Jointplot

A Jointplot comprises three plots. Out of the three, one plot displays a bivariate graph which shows how the dependent variable(Y) varies with the independent variable(X).

Jointplot is seaborn library specific and can be used to quickly visualize and analyze the relationship between two variables and describe their individual distributions on the same plot.

In [127…

```
sn.jointplot(x = "F5", y = "F6",
             kind = "hex", data = df)
# show the plot
plt.show()
```



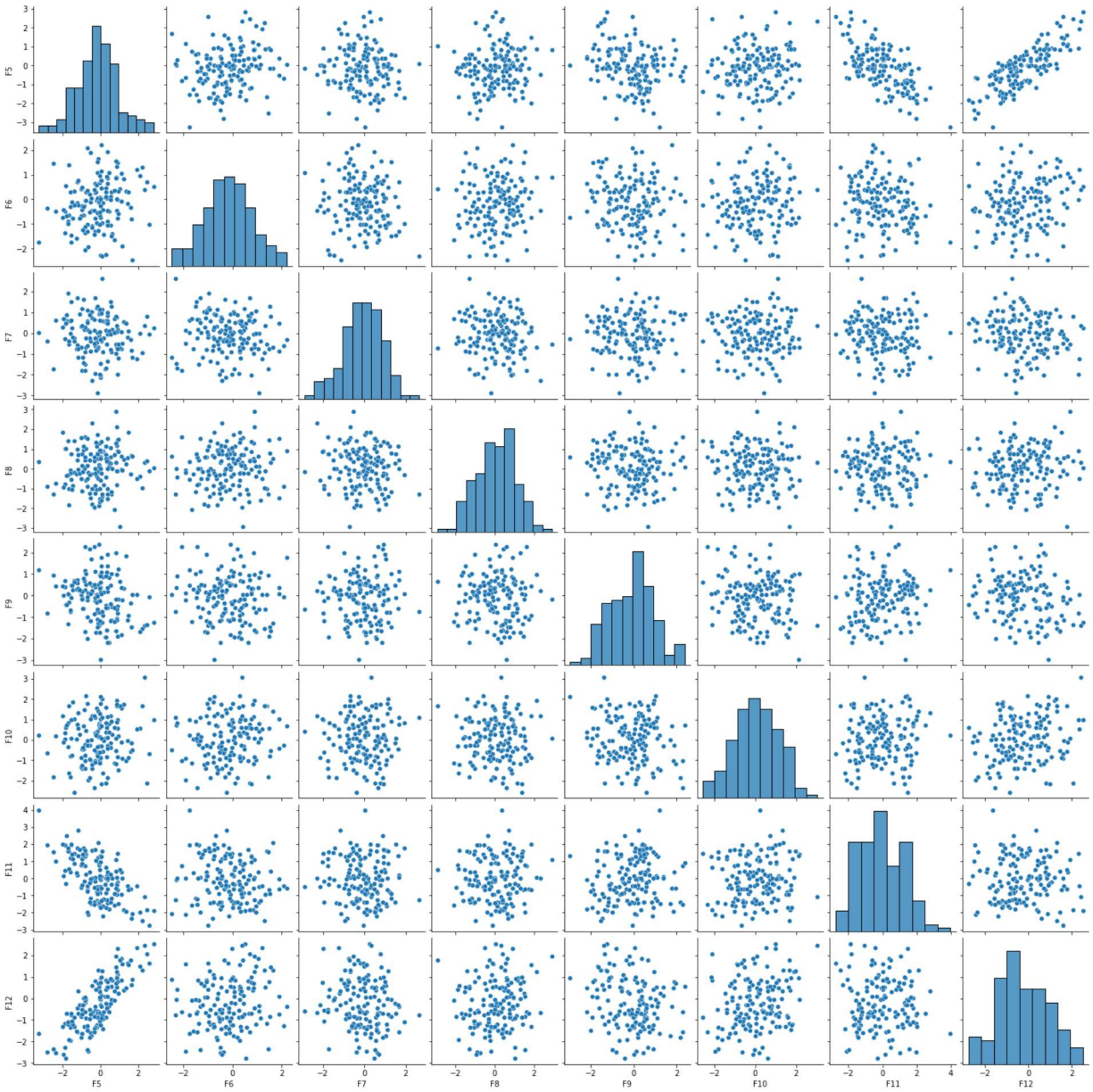
(3)Pairplot

The pairs plot is a grid of scatterplots. showing the bivariate relationships between all

pairs of variables in a multivariate dataset.

In [129…
```
sn.pairplot(df)
```

Out[129… <seaborn.axisgrid.PairGrid at 0x2cb12769250>

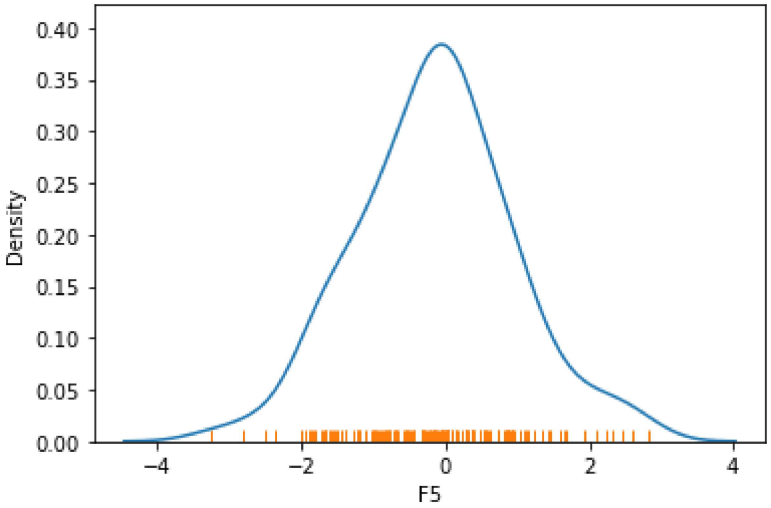


(4)rugplot

A rugplot is a graph that places a dash horizontally with each occurrence of an item in a dataset.

In [135…
```
sn.kdeplot(data=df, x="F5")
sn.rugplot(data=df, x="F5")
```

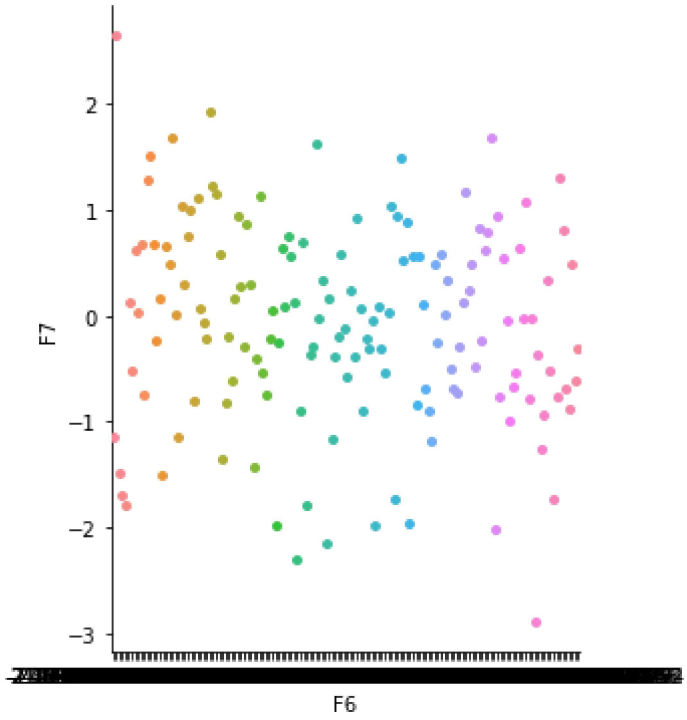
Out[135… <AxesSubplot:xlabel='F5', ylabel='Density'>



(5)catplot

catplot shows frequencies of the categories of one, two or three categorical variables.

In [137…
```
g = sn.catplot(x="F6", y="F7", data=df)
```

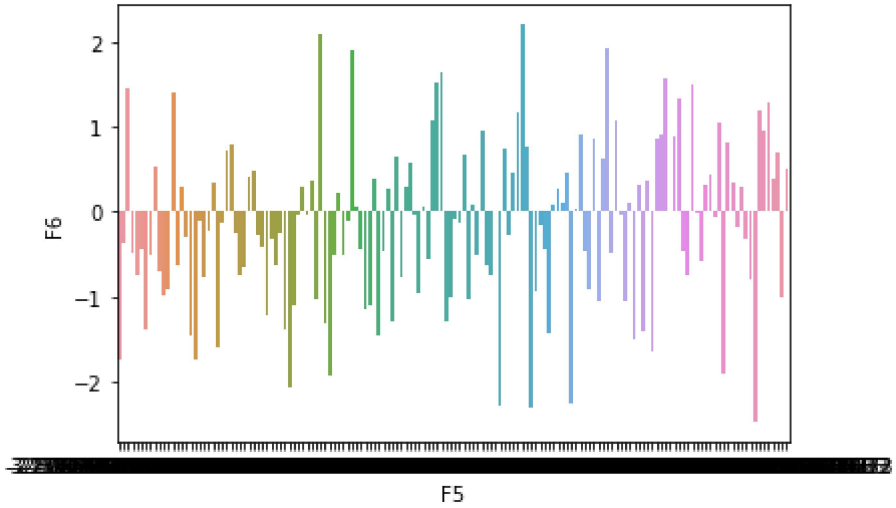


(6)Barplot

A barplot (or barchart) is one of the most common types of graphic. It shows the relationship between a numeric and a categoric variable.

In [166…
```
sn.barplot(x="F5",y="F6",data=df)
```

Out[166… `<AxesSubplot:xlabel='F5', ylabel='F6'>`

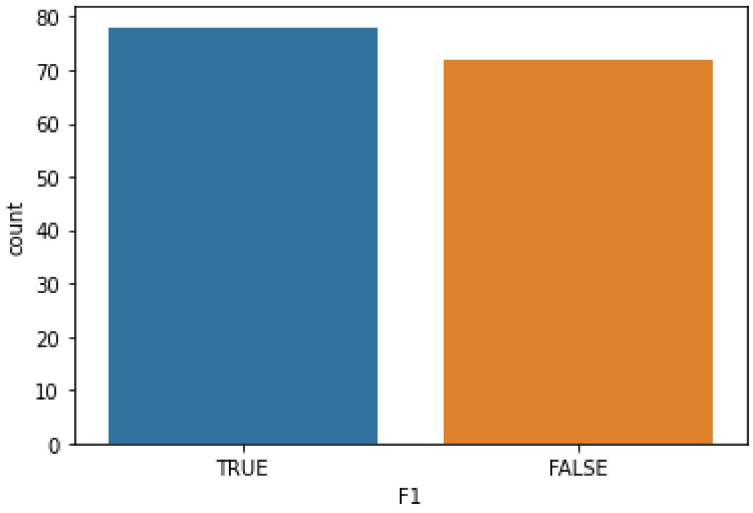


(7)countplot

A countplot is kind of likea histogram or a bar graph for some categorical area

In [150…

```
ax = sn.countplot(x="F1", data=df)
```

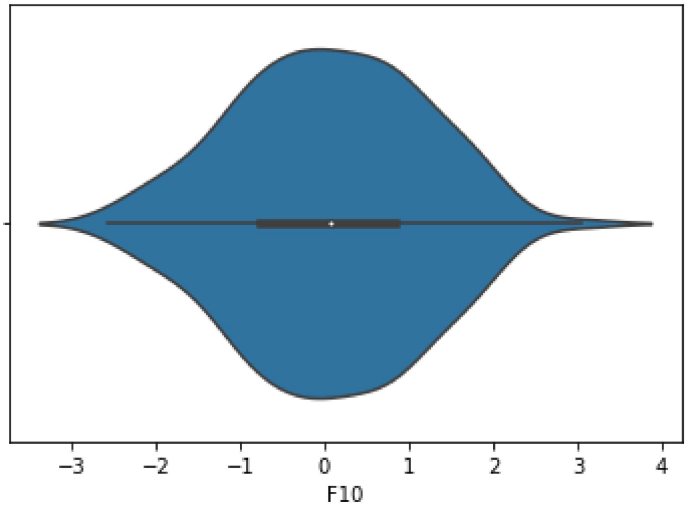


(8)violinplot

Violin plots are used when you want to observe the distribution of numeric data, and are especially useful when you want to make a comparison of distributions between multiple groups

In [152…

```
ax = sn.violinplot(x=df["F10"])
```

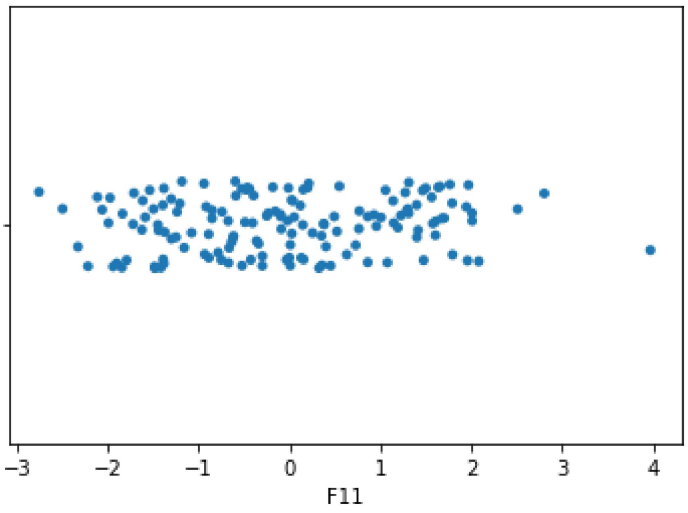


(9)Strip plot

strip plot is a graphical data anlysis technique for summarizing a univariate data set

In [153…

```
ax = sn.stripplot(x=df["F11"])
```

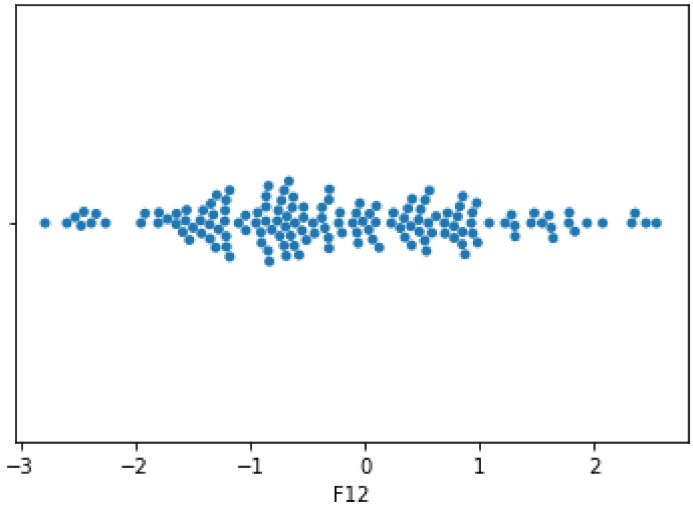


(10)swarmplot

A swarm plot is a type of scatter plot that is used for representing categorical values. It is very similar to the strip plot, but it avoids the overlapping of points

In [159…

```
ax = sn.swarmplot(x=df["F12"])
```

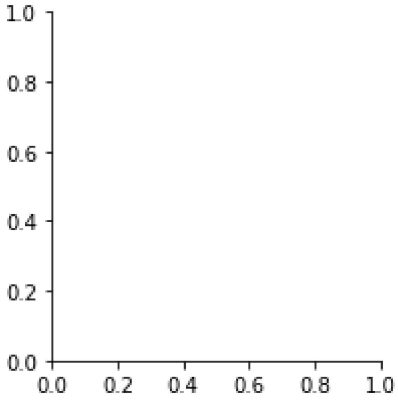


(11)FacetGrid

FacetGrid class helps in visualizing distribution of one variable as well as the relationship between multiple variables separately within subsets of your dataset using multiple panels.

In [157…
```
sn.FacetGrid(df)
```

Out[157… <seaborn.axisgrid.FacetGrid at 0x2cb177bf640>



PART-C

(Q)Load the Iris Dataset, and explore - scatterplot, scatter_3d, heatmap, boxplot, kdeplot etc.In both the cases, write your observations on the plot outputs and how it is relevant.

In [164…
```
#IMPORTING SOME LIBRARY FOR VISUALIZATION
import pandas as pd
from sklearn import datasets
import matplotlib.pyplot as plt
import seaborn as sn
```

LOADING DATASET

In [160…
```
data = pd.read_csv(r"C:\Users\jagan\Downloads\Iris.csv")
```

In [161…
```
data
```

Out[161…

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
0	1	5.1	3.5	1.4	0.2	Iris-setosa
1	2	4.9	3.0	1.4	0.2	Iris-setosa
2	3	4.7	3.2	1.3	0.2	Iris-setosa
3	4	4.6	3.1	1.5	0.2	Iris-setosa
4	5	5.0	3.6	1.4	0.2	Iris-setosa
…	…	…	…	…	…	…
145	146	6.7	3.0	5.2	2.3	Iris-virginica
146	147	6.3	2.5	5.0	1.9	Iris-virginica
147	148	6.5	3.0	5.2	2.0	Iris-virginica
148	149	6.2	3.4	5.4	2.3	Iris-virginica
149	150	5.9	3.0	5.1	1.8	Iris-virginica

150 rows × 6 columns

In [4]:
```
print (data.head(10))
```

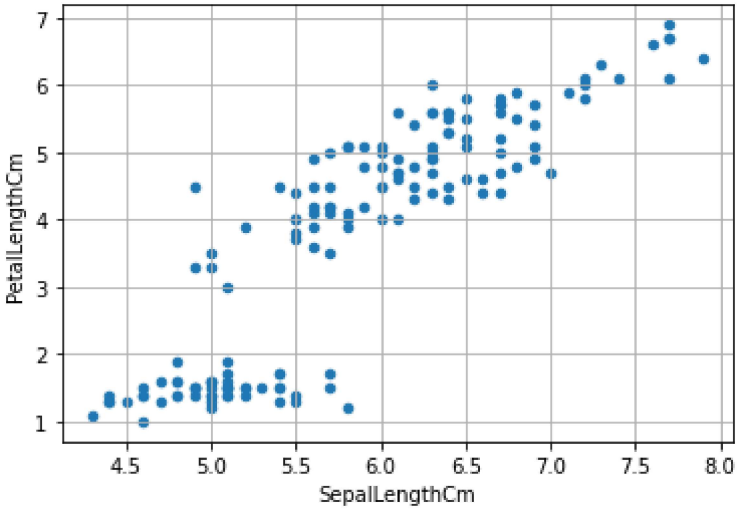
```
   Id  SepalLengthCm  SepalWidthCm  PetalLengthCm  PetalWidthCm      Species
0   1            5.1           3.5            1.4           0.2  Iris-setosa
1   2            4.9           3.0            1.4           0.2  Iris-setosa
2   3            4.7           3.2            1.3           0.2  Iris-setosa
3   4            4.6           3.1            1.5           0.2  Iris-setosa
4   5            5.0           3.6            1.4           0.2  Iris-setosa
5   6            5.4           3.9            1.7           0.4  Iris-setosa
6   7            4.6           3.4            1.4           0.3  Iris-setosa
7   8            5.0           3.4            1.5           0.2  Iris-setosa
8   9            4.4           2.9            1.4           0.2  Iris-setosa
9  10            4.9           3.1            1.5           0.1  Iris-setosa
```

Visualization

1.scatterplot

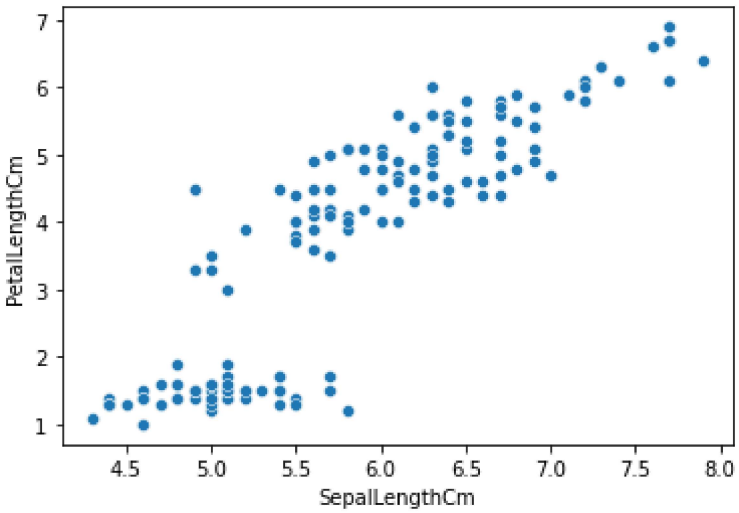
A scatter plot (also called a scatterplot, scatter graph, scatter chart, scattergram, or scatter diagram) is a type of plot or mathematical diagram using Cartesian coordinates to display values for typically two variables for a set of data.

In [165…
```
data.plot(kind ="scatter", x ='SepalLengthCm',y ='PetalLengthCm')
plt.grid()
```



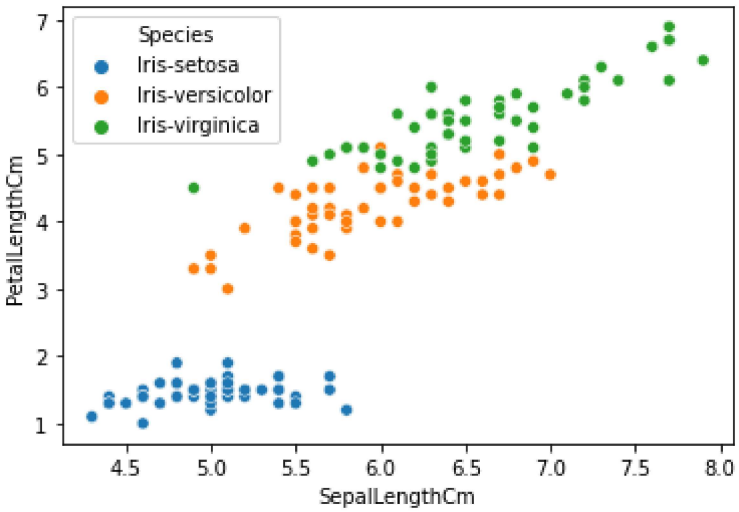
In [7]:
```
sn.scatterplot(data=data, x="SepalLengthCm", y="PetalLengthCm")
```

Out[7]: <AxesSubplot:xlabel='SepalLengthCm', ylabel='PetalLengthCm'>



In [8]:
```
sn.scatterplot(data=data, x="SepalLengthCm", y="PetalLengthCm",hue="Species")
#Assigning a variable to hue will map its levels to the color of the points:
```

Out[8]: <AxesSubplot:xlabel='SepalLengthCm', ylabel='PetalLengthCm'>



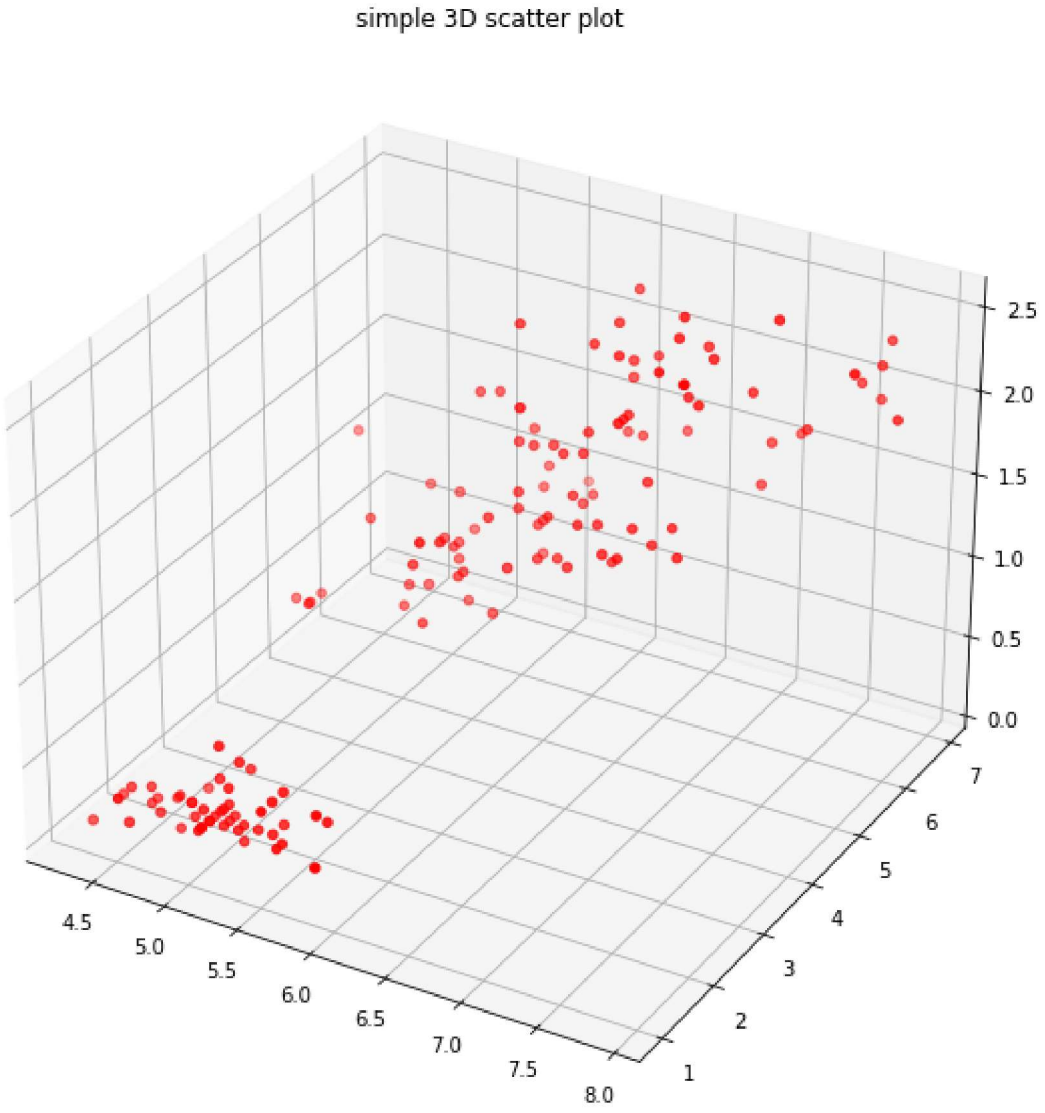
2.Scatter3d

In [9]:
```
# Import libraries
from mpl_toolkits import mplot3d
import matplotlib.pyplot as plt
```

In [10]:
```
# Creating figure
fig = plt.figure(figsize = (20, 10))
ax = plt.axes(projection ="3d")

# Creating plot
ax.scatter3D(data.SepalLengthCm,data.PetalLengthCm,data.PetalWidthCm, color = "red")
plt.title("simple 3D scatter plot")

# show plot
plt.show()
```

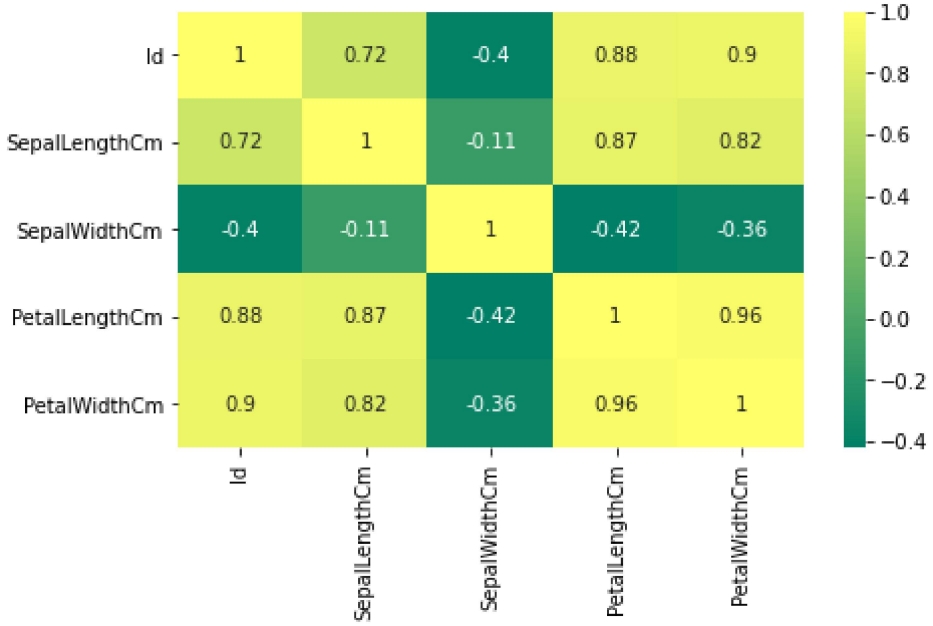


3.Heat Map

A heat map is a data visualization technique that shows magnitude of a phenomenon as color in two dimensions. The variation in color may be by hue or intensity, giving obvious visual cues to the reader about how the phenomenon is clustered or varies over space.

In [15]:
```
plt.figure(figsize=(7,4))
sn.heatmap(data.corr(),annot=True,cmap="summer")
```

Out[15]: <AxesSubplot:>

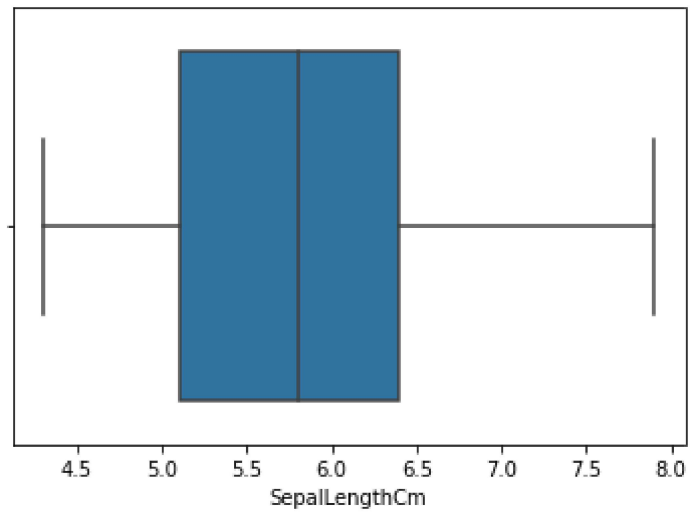


(4)Boxplot

shows the distribution of quantitative data in a way that facilitates comparisons between variables or across levels of a categorical variable.

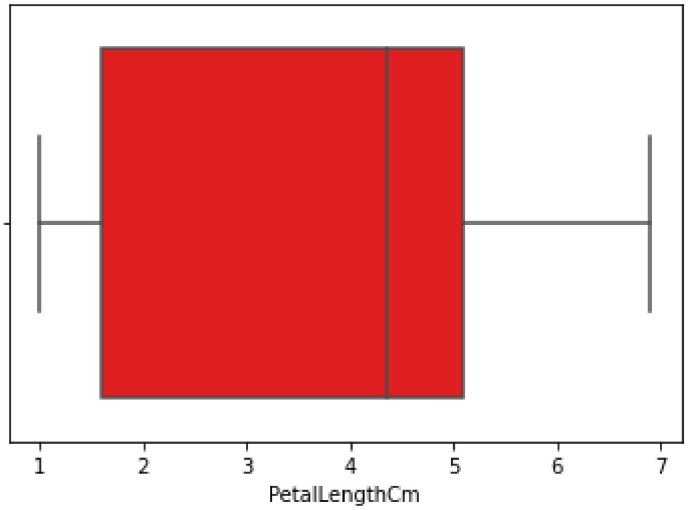
In [169…

```
x = sn.boxplot(x=data["SepalLengthCm"])
```

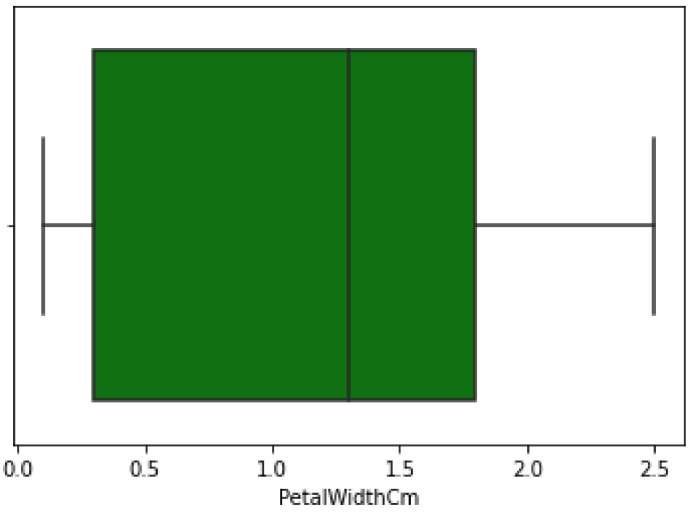


In [21]:

```
y = sn.boxplot(x=data["PetalLengthCm"],color="red")
```



In [20]:
```
z=sn.boxplot(x=data["PetalWidthCm"],color="green")
```

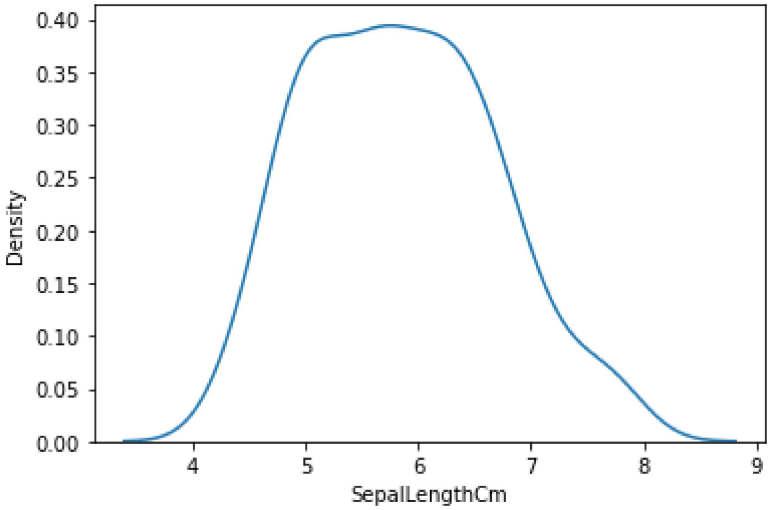


(5)kdeplot

Image result for kdeplot Kdeplot is a Kernel Distribution Estimation Plot which depicts the probability density function of the continuous or non-parametric data variables i.e. we can plot for the univariate or multiple variables altogether.

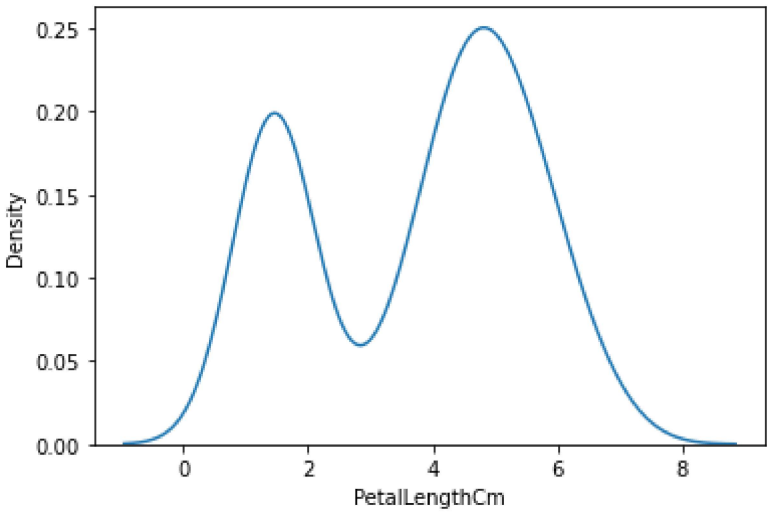
In [23]:
```
sn.kdeplot(data=data, x="SepalLengthCm")
```

Out[23]:
```
<AxesSubplot:xlabel='SepalLengthCm', ylabel='Density'>
```



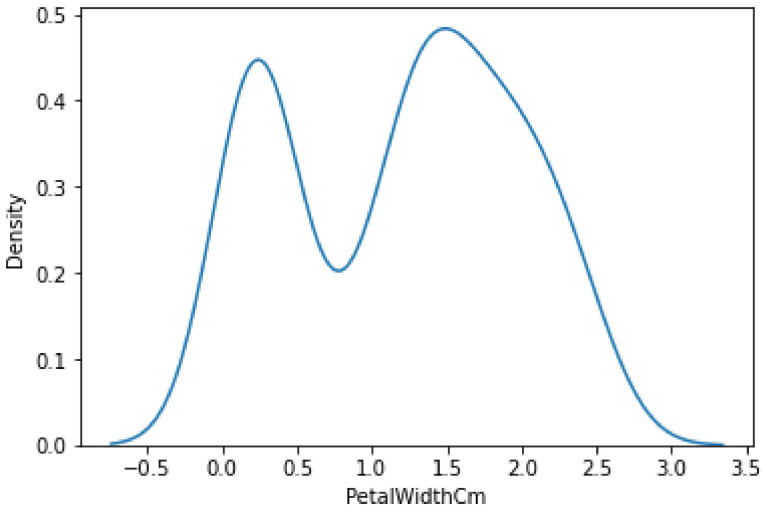
In [25]:
```
sn.kdeplot(data=data,x="PetalLengthCm")
```

Out[25]: <AxesSubplot:xlabel='PetalLengthCm', ylabel='Density'>



In [26]:
```
sn.kdeplot(data=data,x="PetalWidthCm")
```

Out[26]: <AxesSubplot:xlabel='PetalWidthCm', ylabel='Density'>



CONCLUSION

Here i understand some more concepts of machine learning like i tried to explore more about the library and explore more on the visualzation tools. mainly distplot, jointplot, pairplot, rugplot, catplot, barplot, countplot, violinplot, striplot, swarmplot and facetgrid plots,scatterplot, scatter_3d, heatmap, boxplot, kdeplot

REFERENCE

- https://seaborn.pydata.org/
- https://www.geeksforgeeks.org/matplotlib-tutorial/
- https://scikit-learn.org/stable/
-

In [ ]: