lets Grow More

Data science internship

Task 1

Name : Manisha thokale

Importing libraries

In [1]:

```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
from sklearn.neighbors import KNeighborsClassifier
%matplotlib inline
```

Read Data

In [2]:

```
df = pd.read_csv("c:\\Users\\Asus\\Downloads\\Iris dataset.csv")
```

In [3]:

```
df.describe()
```

Out[3]:

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm
count	150.000000	150.000000	150.000000	150.000000	150.000000
mean	75.500000	5.843333	3.054000	3.758667	1.198667
std	43.445368	0.828066	0.433594	1.764420	0.763161
min	1.000000	4.300000	2.000000	1.000000	0.100000
25%	38.250000	5.100000	2.800000	1.600000	0.300000
50%	75.500000	5.800000	3.000000	4.350000	1.300000
75%	112.750000	6.400000	3.300000	5.100000	1.800000
max	150.000000	7.900000	4.400000	6.900000	2.500000

In [4]:

```
df.head()
```

Out[4]:

	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

In [5]:

```
df.shape
```

Out[5]:

```
(150, 6)
```

In [6]:

```
df.shape
```

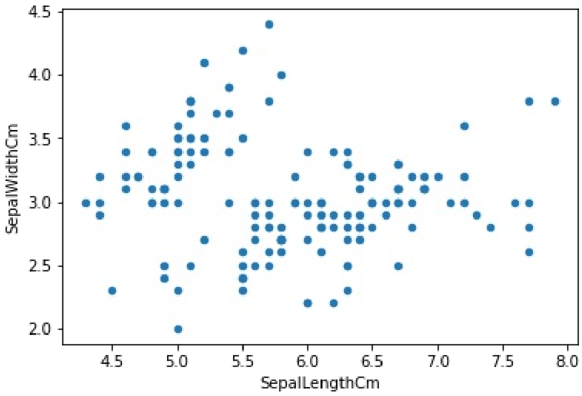
Out[6]:
```
(150, 6)
```

In [7]:
```
df.isnull().sum()
```

Out[7]:
```
Id               0
SepalLengthCm    0
SepalWidthCm     0
PetalLengthCm    0
PetalWidthCm     0
Species          0
dtype: int64
```

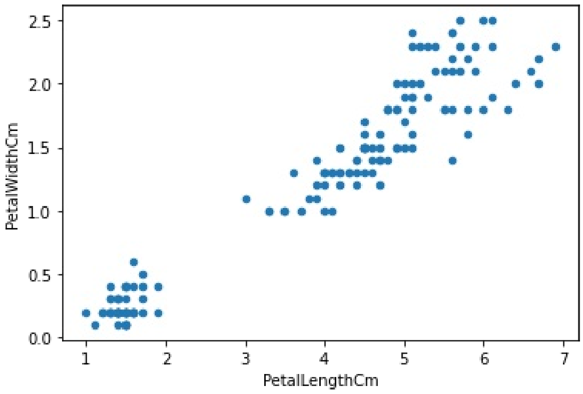
In [8]:
```
## ScatterPlot

df.plot(kind="scatter", x="SepalLengthCm", y="SepalWidthCm")
```

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



In [9]:
```
df.plot(kind="scatter" , x="PetalLengthCm", y="PetalWidthCm")
```

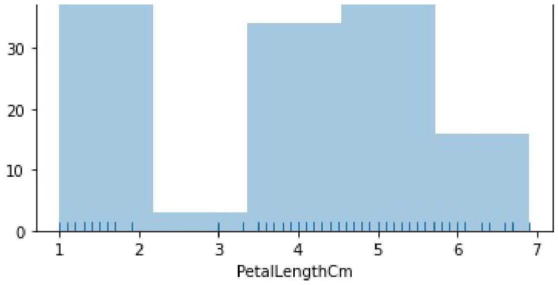
Out[9]:
```
<AxesSubplot:xlabel='PetalLengthCm', ylabel='PetalWidthCm'>
```



In [10]:
```
## Histogram

sns.distplot(df['PetalLengthCm'], kde=False, rug=True)
```

```
C:\Users\Asus\anaconda3\lib\site-packages\seaborn\distributions.py:2619: FutureWarning: `distplot` is a deprecate
d function and will be removed in a future version. Please adapt your code to use either `displot` (a figure-leve
l function with similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
C:\Users\Asus\anaconda3\lib\site-packages\seaborn\distributions.py:2103: FutureWarning: The `axis` variable is no
longer used and will be removed. Instead, assign variables directly to `x` or `y`.
  warnings.warn(msg, FutureWarning)
```

Out[10]:
```
<AxesSubplot:xlabel='PetalLengthCm'>
```



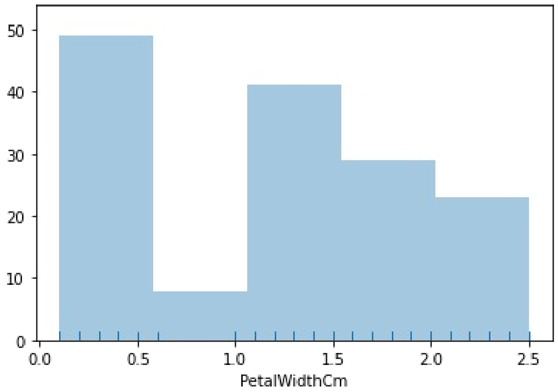
In [11]:

```
sns.distplot(df['PetalWidthCm'], kde=False, rug=True)
```

```
C:\Users\Asus\anaconda3\lib\site-packages\seaborn\distributions.py:2619: FutureWarning: `distplot` is a deprecate
d function and will be removed in a future version. Please adapt your code to use either `displot` (a figure-leve
l function with similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
C:\Users\Asus\anaconda3\lib\site-packages\seaborn\distributions.py:2103: FutureWarning: The `axis` variable is no
longer used and will be removed. Instead, assign variables directly to `x` or `y`.
  warnings.warn(msg, FutureWarning)
```

Out[11]:

```
<AxesSubplot:xlabel='PetalWidthCm'>
```

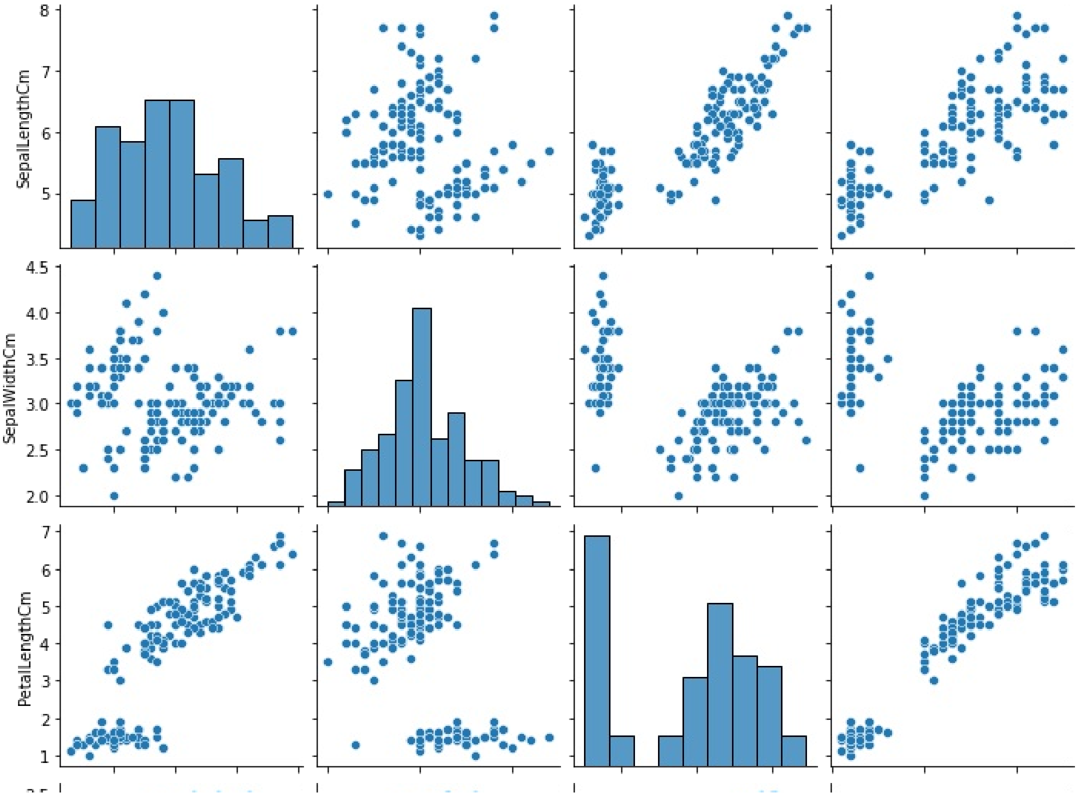


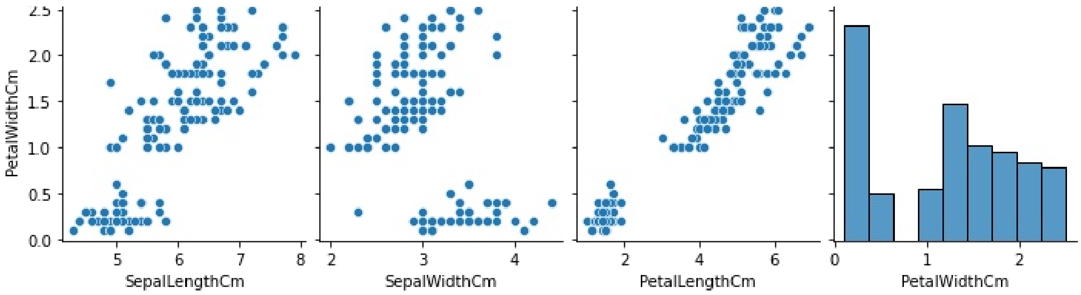
In [12]:

```
# Pairplot

sns.pairplot(df[['SepalLengthCm', 'SepalWidthCm', 'PetalLengthCm', 'PetalWidthCm']])
```

Out[12]:

```
<seaborn.axisgrid.PairGrid at 0x1ba7187f880>
```



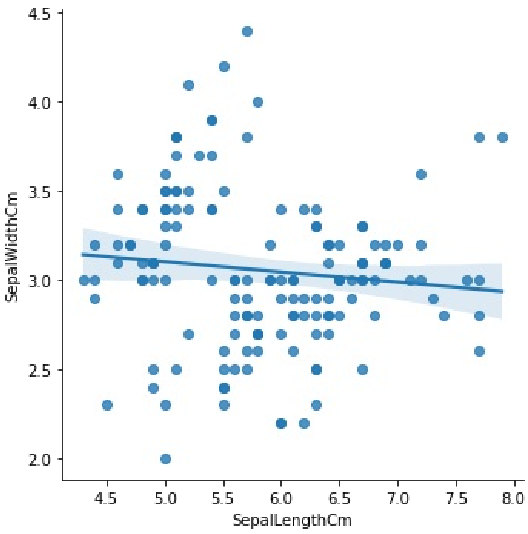


In [13]:

```
sns.lmplot(x="SepalLengthCm", y="SepalWidthCm", data=df)
```

Out[13]:

```
<seaborn.axisgrid.FacetGrid at 0x1ba721b8460>
```

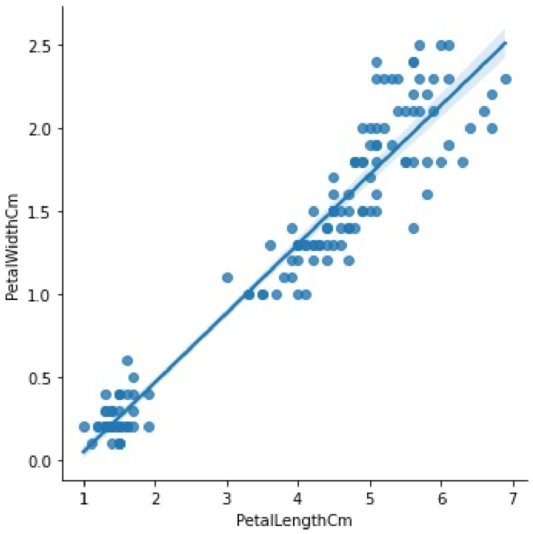


In [16]:

```
sns.lmplot(x="PetalLengthCm", y="PetalWidthCm", data=df)
```

Out[16]:

```
<seaborn.axisgrid.FacetGrid at 0x1ba72104880>
```



KNneighbor classifier

In [17]:

```
Numerical_data = df.drop('Species',axis=1)
Species_data = df['Species']
x_train, x_test, y_train, y_test = train_test_split(Numerical_data, Species_data, test_size=0.25)

kmeans=KNeighborsClassifier(n_neighbors=3)
kmeans.fit(x_train, y_train)
```

```
KNeighborsClassifier(n_neighbors=3)
```

Out[17]:

```
KNeighborsClassifier(n_neighbors=3)
```

In [18]:

```
print("Accuracy: ",kmeans.score(x_train,y_train)*100)
```

```
Accuracy:  100.0
```

In [ ]:

```

```

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js