EDA

March 23, 2023

1 Exploratory Data Analysis

1.1 Import all necessary Libraries

[1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

1.2 Import iris dataset

[2]:
```
df=sns.load_dataset("iris")
```

[3]:
```
df
```

[3]:

	sepal_length	sepal_width	petal_length	petal_width	species
0	5.1	3.5	1.4	0.2	setosa
1	4.9	3.0	1.4	0.2	setosa
2	4.7	3.2	1.3	0.2	setosa
3	4.6	3.1	1.5	0.2	setosa
4	5.0	3.6	1.4	0.2	setosa
..	…	…	…	…	…
145	6.7	3.0	5.2	2.3	virginica
146	6.3	2.5	5.0	1.9	virginica
147	6.5	3.0	5.2	2.0	virginica
148	6.2	3.4	5.4	2.3	virginica
149	5.9	3.0	5.1	1.8	virginica

[150 rows x 5 columns]

2 Summary of the Data

[4]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 150 entries, 0 to 149
Data columns (total 5 columns):
```

```
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
 0   sepal_length  150 non-null    float64
 1   sepal_width   150 non-null    float64
 2   petal_length  150 non-null    float64
 3   petal_width   150 non-null    float64
 4   species       150 non-null    object
dtypes: float64(4), object(1)
memory usage: 6.0+ KB
```

2.1 Statistical summary of Data

[5]:
```
df.describe()
```

[5]:

	sepal_length	sepal_width	petal_length	petal_width
count	150.000000	150.000000	150.000000	150.000000
mean	5.843333	3.057333	3.758000	1.199333
std	0.828066	0.435866	1.765298	0.762238
min	4.300000	2.000000	1.000000	0.100000
25%	5.100000	2.800000	1.600000	0.300000
50%	5.800000	3.000000	4.350000	1.300000
75%	6.400000	3.300000	5.100000	1.800000
max	7.900000	4.400000	6.900000	2.500000

3 See all the columns

[7]:
```
df.columns
```

[7]:
```
Index(['sepal_length', 'sepal_width', 'petal_length', 'petal_width',
       'species'],
      dtype='object')
```

3.1 Check for any null values

[8]:
```
df.isnull().sum()
```

[8]:
```
sepal_length    0
sepal_width     0
petal_length    0
petal_width     0
species         0
dtype: int64
```

3.2 Check the covariance of the Data

[9]:
```
df.cov()
```

```
/tmp/ipykernel_110/1545644723.py:1: FutureWarning: The default value of
numeric_only in DataFrame.cov is deprecated. In a future version, it will
default to False. Select only valid columns or specify the value of numeric_only
to silence this warning.
  df.cov()
```

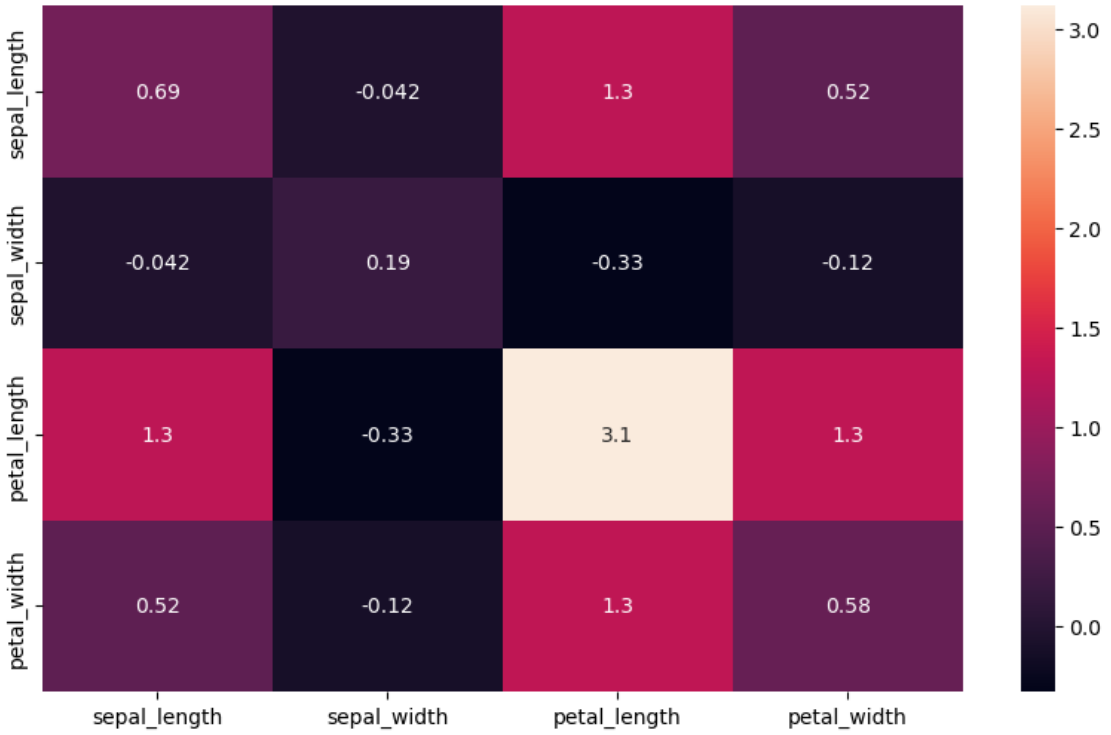
[9]:

	sepal_length	sepal_width	petal_length	petal_width
sepal_length	0.685694	-0.042434	1.274315	0.516271
sepal_width	-0.042434	0.189979	-0.329656	-0.121639
petal_length	1.274315	-0.329656	3.116278	1.295609
petal_width	0.516271	-0.121639	1.295609	0.581006

[11]:
```
plt.figure(figsize=(10,6))
sns.heatmap(df.cov(),annot=True)
```

```
/tmp/ipykernel_110/3498915755.py:2: FutureWarning: The default value of
numeric_only in DataFrame.cov is deprecated. In a future version, it will
default to False. Select only valid columns or specify the value of numeric_only
to silence this warning.
  sns.heatmap(df.cov(),annot=True)
```

[11]: <AxesSubplot: >



3.3 Check no of unique species

[13]:
```
df.species.unique()
```

[13]:
```
array(['setosa', 'versicolor', 'virginica'], dtype=object)
```

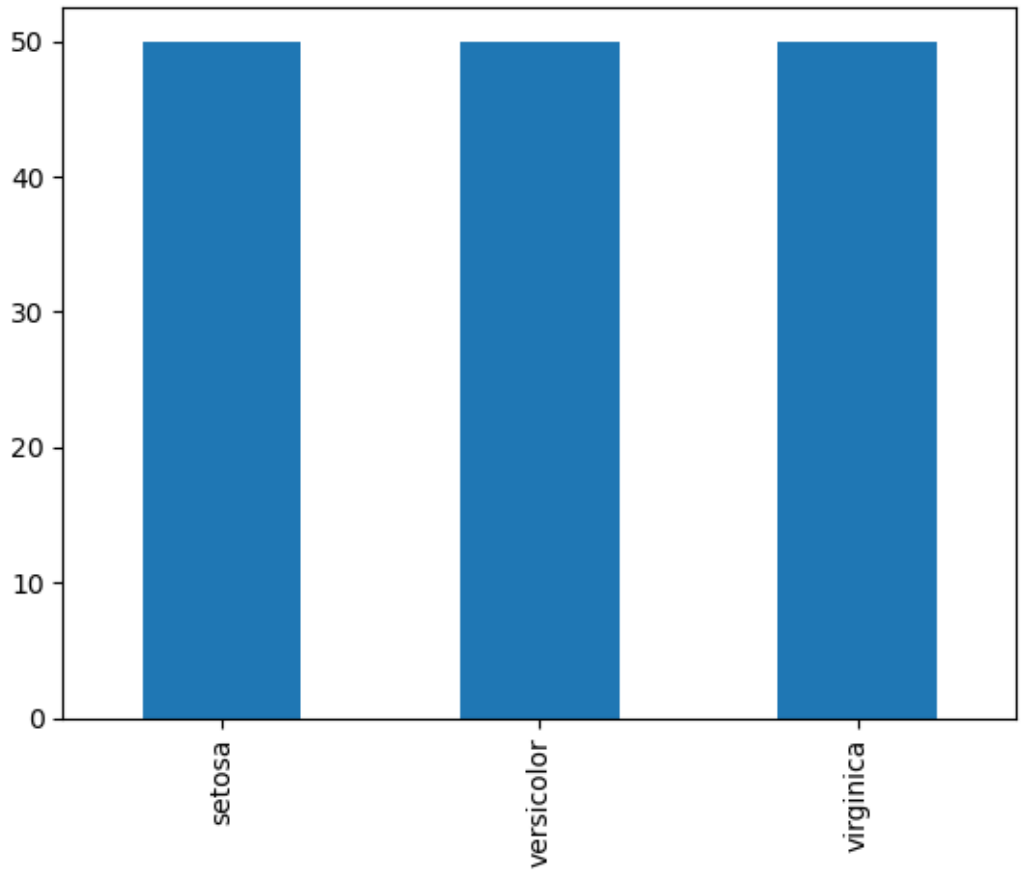
3.4 See no of species for each particular species

[14]:
```
df.species.value_counts()
```

[14]:
```
setosa        50
versicolor    50
virginica     50
Name: species, dtype: int64
```

[17]:
```
df.species.value_counts().plot(kind="bar")
```

[17]:
```
<AxesSubplot: >
```



3.5 Check for Duplicates

[24]:
```
df.duplicated()
```

[24]:
```
0      False
1      False
2      False
3      False
4      False
       …
145    False
146    False
147    False
148    False
149    False
Length: 150, dtype: bool
```

[25]:
```
df.shape
```

[25]:
```
(150, 5)
```

[26]:
```
df[df.duplicated()]
```

[26]:

	sepal_length	sepal_width	petal_length	petal_width	species
142	5.8	2.7	5.1	1.9	virginica

3.6 Drop the Duplicates

[29]:
```
df.drop_duplicates(inplace=True)
```

[30]:
```
df
```

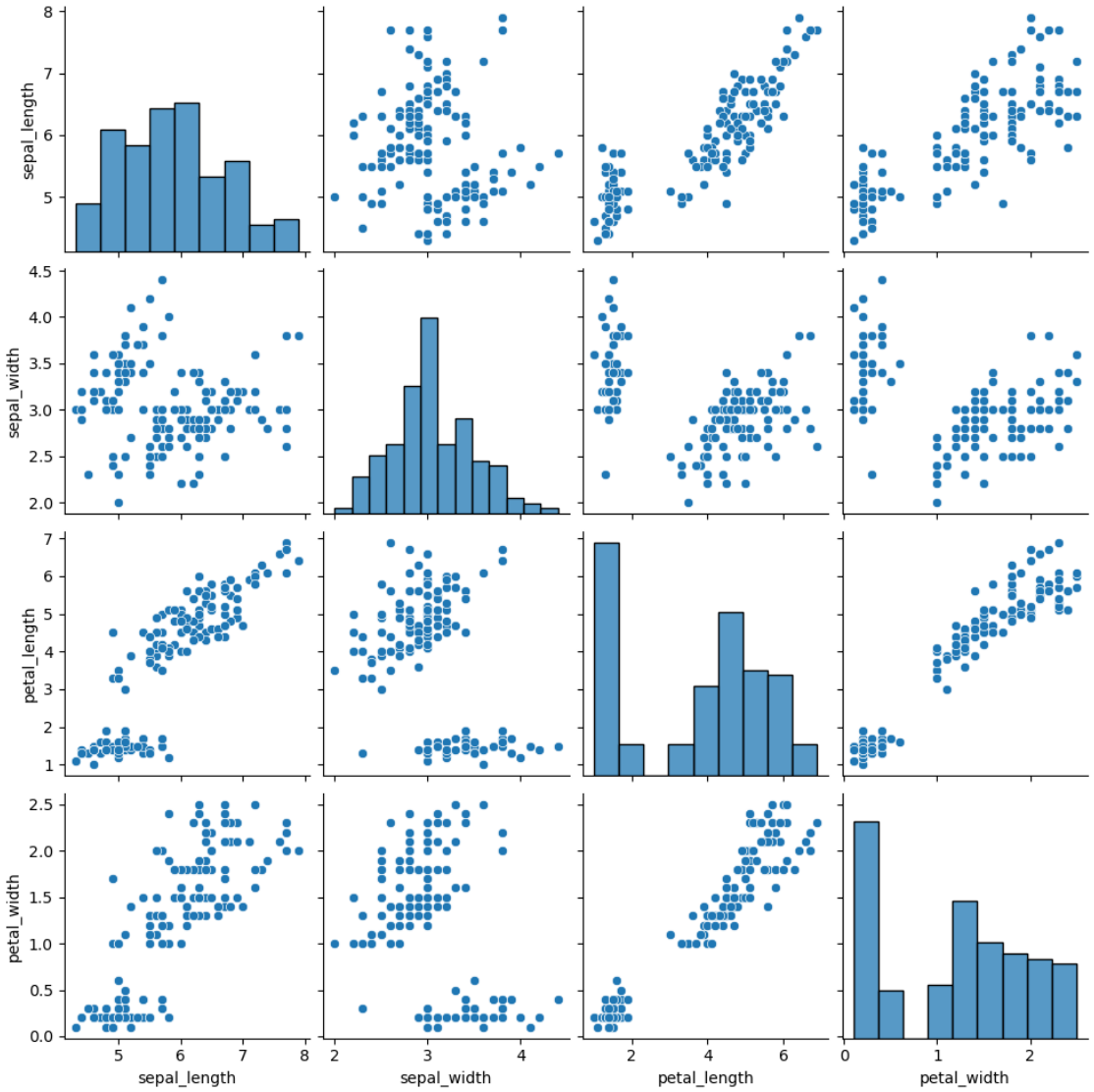
[30]:

	sepal_length	sepal_width	petal_length	petal_width	species
0	5.1	3.5	1.4	0.2	setosa
1	4.9	3.0	1.4	0.2	setosa
2	4.7	3.2	1.3	0.2	setosa
3	4.6	3.1	1.5	0.2	setosa
4	5.0	3.6	1.4	0.2	setosa
..	…	…	…	…	…
145	6.7	3.0	5.2	2.3	virginica
146	6.3	2.5	5.0	1.9	virginica
147	6.5	3.0	5.2	2.0	virginica
148	6.2	3.4	5.4	2.3	virginica
149	5.9	3.0	5.1	1.8	virginica

```
[149 rows x 5 columns]
```

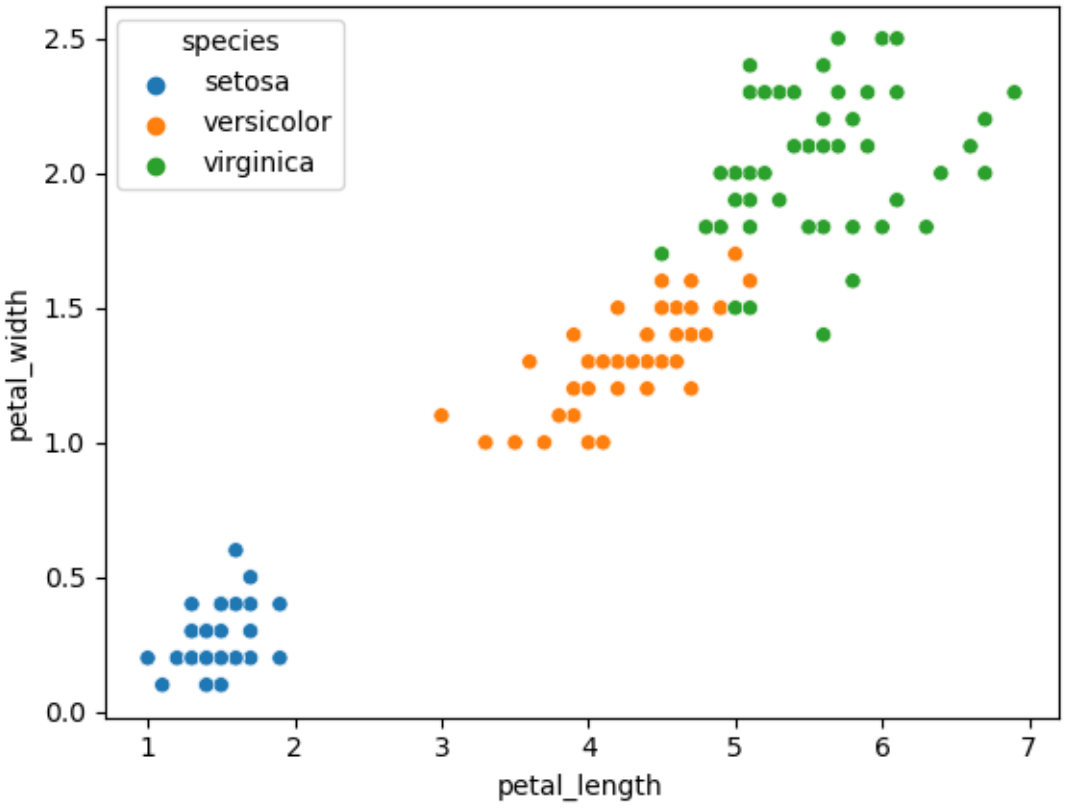
[31]:
```
sns.pairplot(df)
```

[31]:
```
<seaborn.axisgrid.PairGrid at 0x7f5199ede8c0>
```



[37]:
```
sns.scatterplot(x=df.petal_length,y=df.petal_width,hue=df["species"])
```

[37]:
```
<AxesSubplot: xlabel='petal_length', ylabel='petal_width'>
```


species
setosa
versicolor
virginica
petal_width
petal_length
0.0
0.5
1.0
1.5
2.0
2.5
1
2
3
4
5
6
7

[ ]: