Aadarsh Goyal

MIS: 111915001

In [1]:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

In [2]:

```
df = pd.read_csv("iris.csv")
df.head()
```

Out[2]:

	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 [3]:

```
df.shape
```

Out[3]:

```
(150, 6)
```

In [4]:

```
df.describe()
```

Out[4]:

	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

Dropping unwanted column

In [5]:

```
df.drop(columns="Id",inplace=True)
```

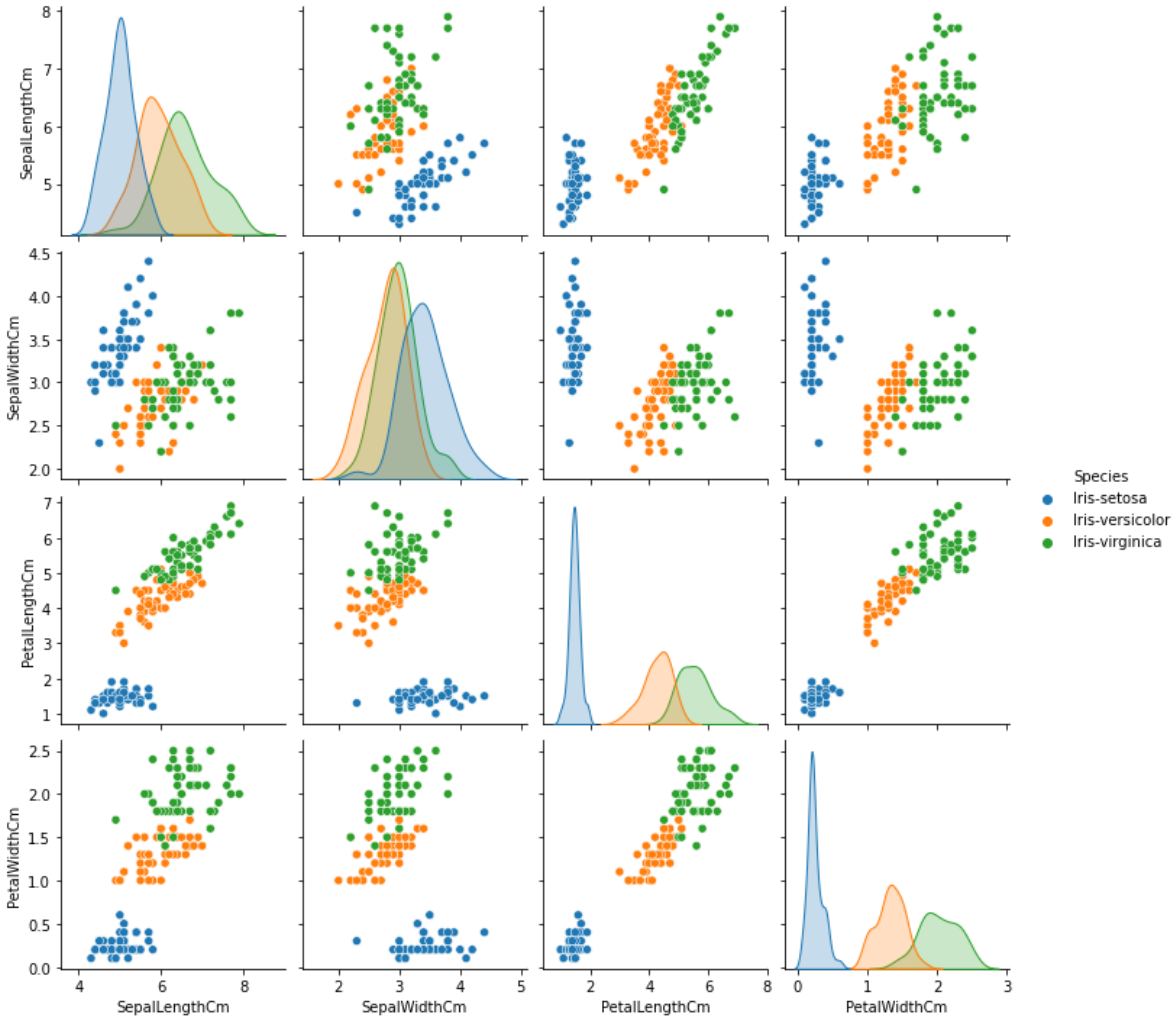
Visualization

In [6]:

```
sns.pairplot(df, hue='Species')
```

Out[6]:

```
<seaborn.axisgrid.PairGrid at 0x1bd96c426c8>
```



In [7]:

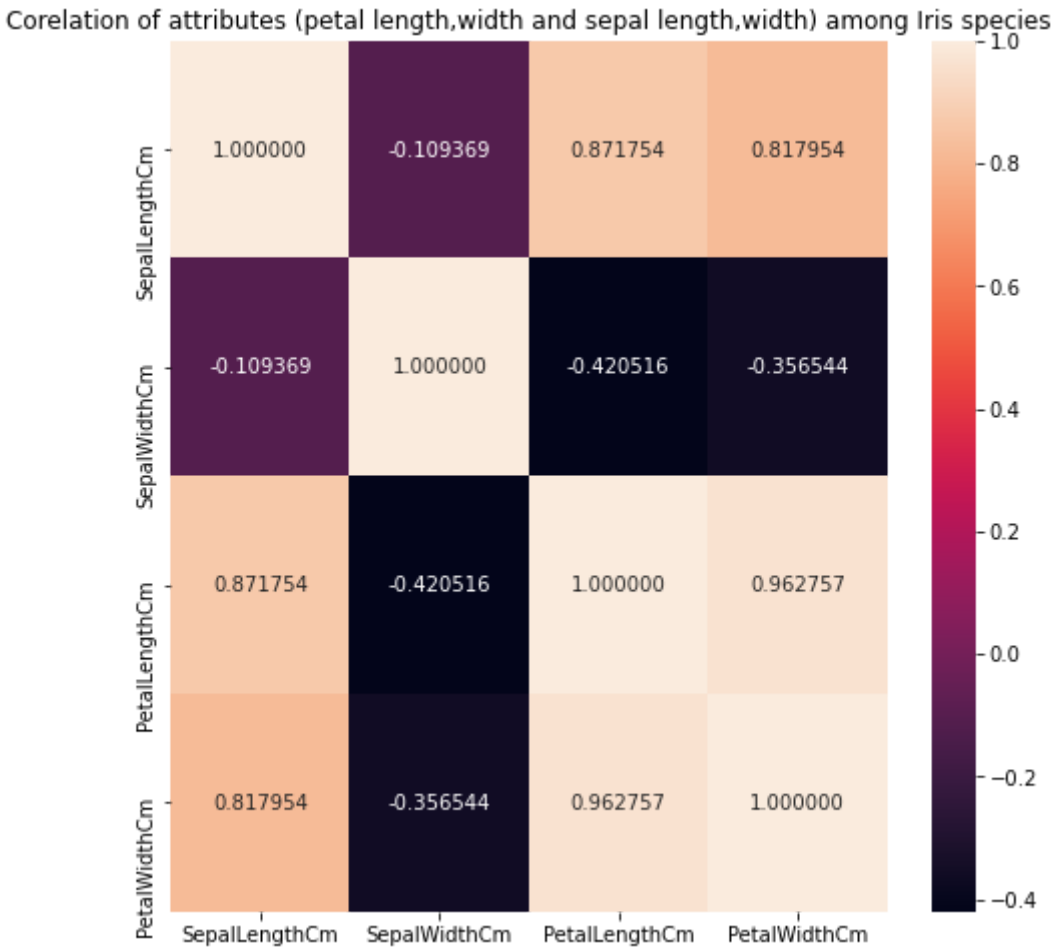
```
df.corr()
```

Out[7]:

	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm
SepalLengthCm	1.000000	-0.109369	0.871754	0.817954
SepalWidthCm	-0.109369	1.000000	-0.420516	-0.356544
PetalLengthCm	0.871754	-0.420516	1.000000	0.962757
PetalWidthCm	0.817954	-0.356544	0.962757	1.000000

In [8]:

```
plt.subplots(figsize = (8,8))
sns.heatmap(df.corr(),annot=True,fmt="f").set_title("Corelation of attributes (petal length
plt.show()
```



The Heatmap Shows that petal-Length and petal-Width are highly corelated

In [9]:

```
x = df.iloc[:,0:-1]
y = df.iloc[:,-1]
```

Apply K-means

In [10]:

```
from sklearn.cluster import KMeans

km = KMeans(n_clusters=3)
prediction = km.fit_predict(df[['PetalLengthCm','PetalWidthCm']])
```

In [11]:

```
dfp = df.iloc[:,0:-1]
dfp['Species'] = prediction
```

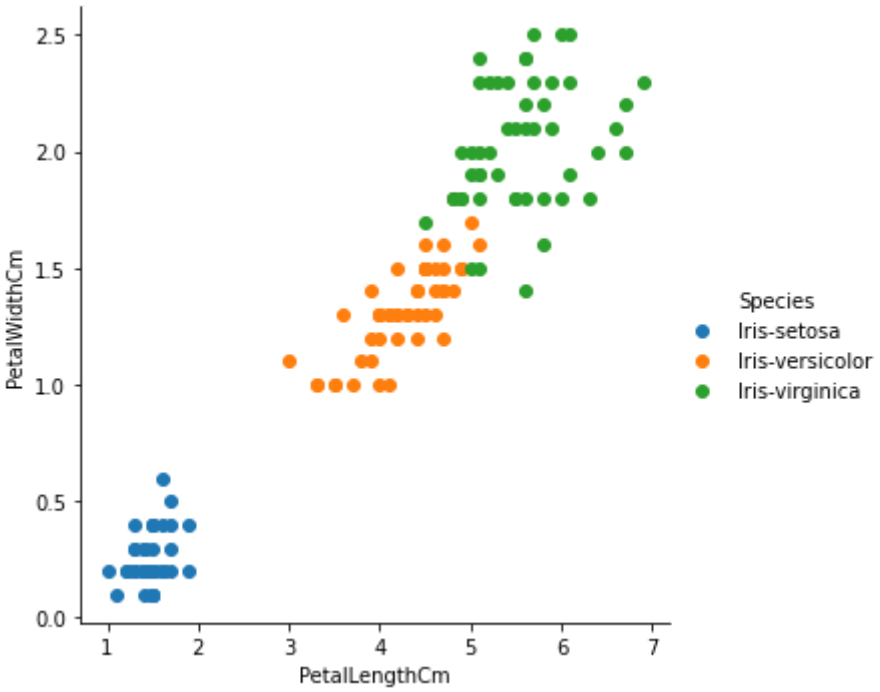
Scatter plot for original data

In [12]:

```
sns.FacetGrid(df, hue ="Species", height = 5).map(plt.scatter, 'PetalLengthCm', 'PetalWidth
```

Out[12]:

```
<seaborn.axisgrid.FacetGrid at 0x1bd96c1b888>
```



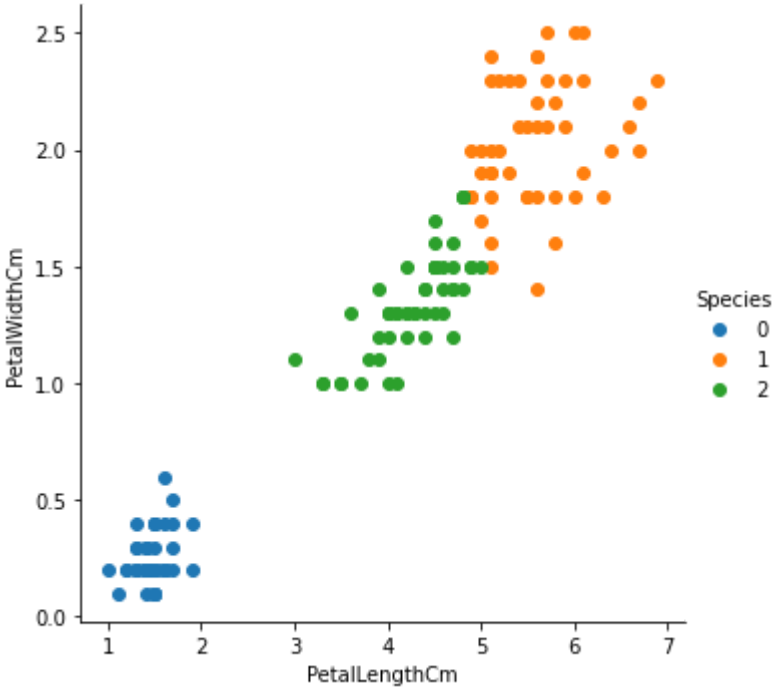
Scatter plot for Predicted data

In [13]:

```
sns.FacetGrid(dfp, hue ="Species", height = 5).map(plt.scatter, 'PetalLengthCm', 'PetalWidt
```

Out[13]:

```
<seaborn.axisgrid.FacetGrid at 0x1bd96c1b648>
```



Lets see the Sum of square error for dirrerent values of k

In [14]:

```
sse = []
for k in range(1,10):
    km = KMeans(n_clusters=k)
    km.fit(x)
    sse.append(km.inertia_)
```

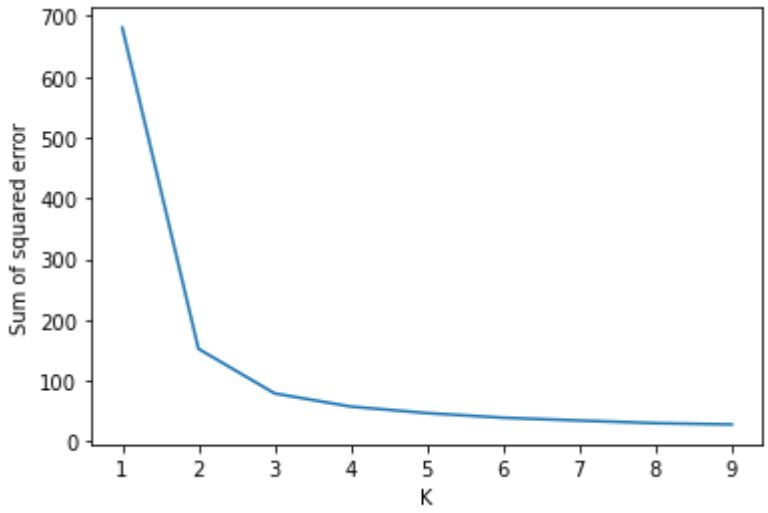
```
D:\Softwares\Anaconda 3\lib\site-packages\sklearn\cluster\_kmeans.py:882: Us
erWarning: KMeans is known to have a memory leak on Windows with MKL, when t
here are less chunks than available threads. You can avoid it by setting the
environment variable OMP_NUM_THREADS=1.
  f"KMeans is known to have a memory leak on Windows "
```

In [15]:

```
plt.xlabel('K')
plt.ylabel('Sum of squared error')
plt.plot(range(1,10),sse)
```

Out[15]:

```
[<matplotlib.lines.Line2D at 0x1bd9a570848>]
```



Now, DBSCAN Clustering

In [16]:

```
from sklearn.cluster import DBSCAN

db = DBSCAN(eps=0.8,min_samples=14)
db.fit(dfp)
```

Out[16]:

```
DBSCAN(eps=0.8, min_samples=14)
```

In [17]:

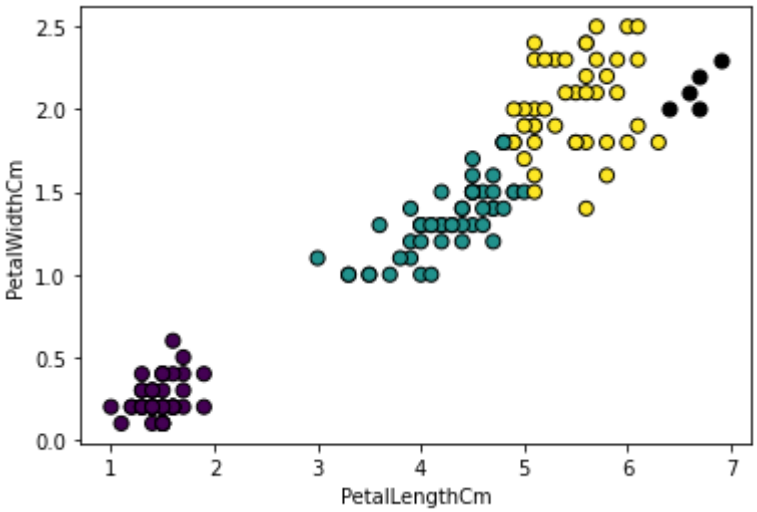
```
cluster=df[db.labels_!=-1]
outlier=df[db.labels_==-1]
colors= db.labels_
ccluster=colors[colors!=-1]
ocluster='black'
```

In [18]:

```
plt.scatter(cluster['PetalLengthCm'],cluster['PetalWidthCm'],c=ccluster,edgecolors='black',
plt.scatter(outlier['PetalLengthCm'],outlier['PetalWidthCm'],c=ocluster,edgecolors='black',
plt.xlabel('PetalLengthCm')
plt.ylabel('PetalWidthCm')
```

Out[18]:

```
Text(0, 0.5, 'PetalWidthCm')
```



In [ ]:

```

```