In [1]:

```
import pandas as pd
import numpy as np
import plotly
import plotly.express as px
import plotly.offline as pyo
import cufflinks as cf
from plotly.offline import init_notebook_mode,plot,iplot

import matplotlib.pyplot as plt
%matplotlib inline
from sklearn.metrics import accuracy_score
import os
```

In [54]:

```
df=pd.read_csv("D:/CHROME DOWNLOADS/iris.csv")
df
```

Out[54]:

	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 [55]:

```
df.sample(4)
```

Out[55]:

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
88	89	5.6	3.0	4.1	1.3	Iris-versicolor
4	5	5.0	3.6	1.4	0.2	Iris-setosa
53	54	5.5	2.3	4.0	1.3	Iris-versicolor
60	61	5.0	2.0	3.5	1.0	Iris-versicolor

In [56]:

```
df.tail(3)
```

Out[56]:

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
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

In [57]:

```
df.drop("Id",axis=1,inplace=True)
```

In [58]:

```
df.rename(columns={'SepalLengthCm':'SL','SepalWidthCm':'SW','PetalLengthCm':'PL','PetalWidt
df
```

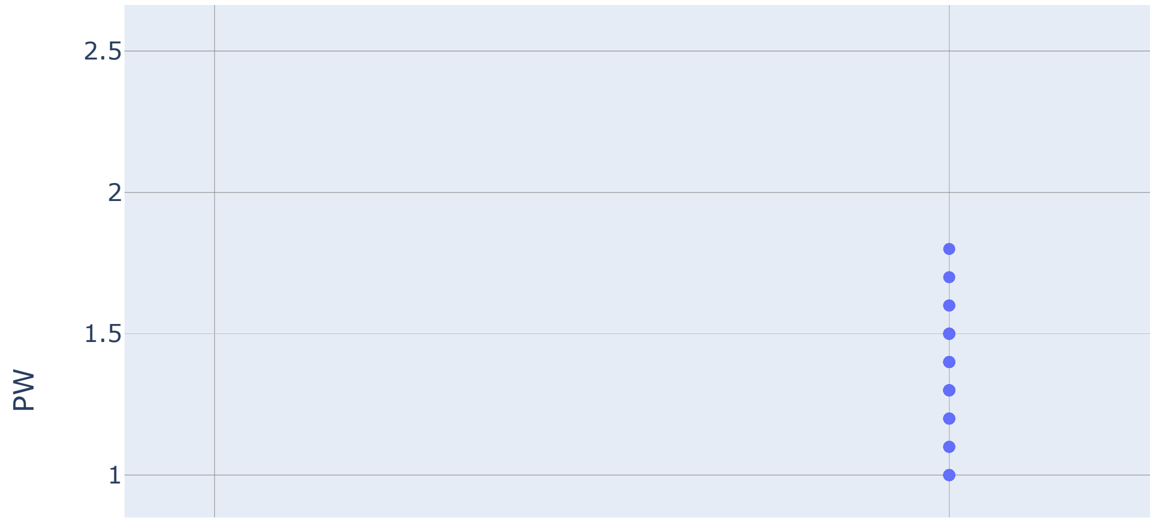
Out[58]:

	SL	SW	PL	PW	Species
0	5.1	3.5	1.4	0.2	Iris-setosa
1	4.9	3.0	1.4	0.2	Iris-setosa
2	4.7	3.2	1.3	0.2	Iris-setosa
3	4.6	3.1	1.5	0.2	Iris-setosa
4	5.0	3.6	1.4	0.2	Iris-setosa
...	...	...	...	...	...
145	6.7	3.0	5.2	2.3	Iris-virginica
146	6.3	2.5	5.0	1.9	Iris-virginica
147	6.5	3.0	5.2	2.0	Iris-virginica
148	6.2	3.4	5.4	2.3	Iris-virginica
149	5.9	3.0	5.1	1.8	Iris-virginica

150 rows × 5 columns

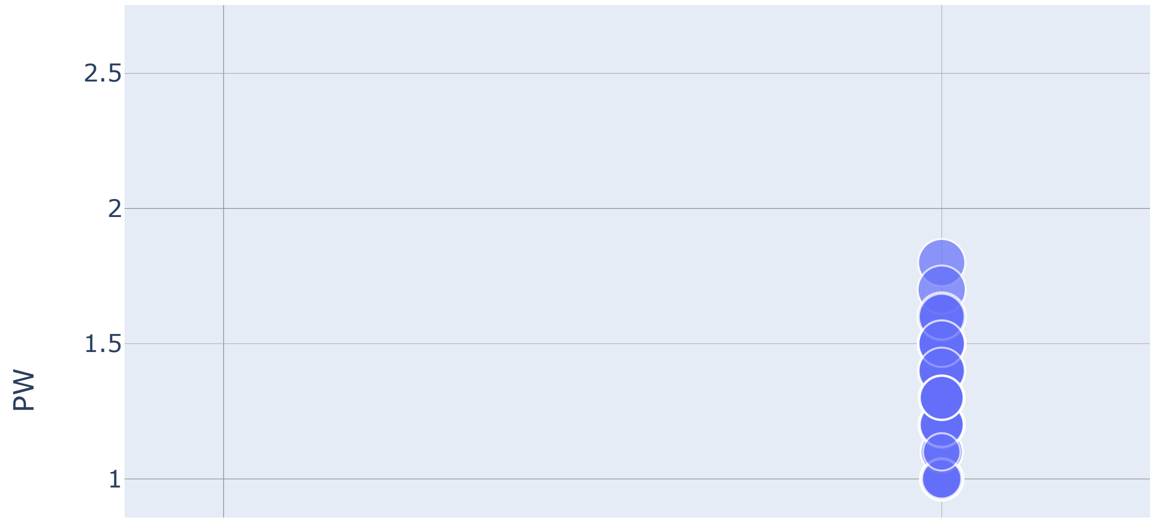
In [59]:

```
px.scatter(df,x="Species",y="PW")
```



In [60]:

```
px.scatter(df,x="Species",y="PW",size="PL")
```

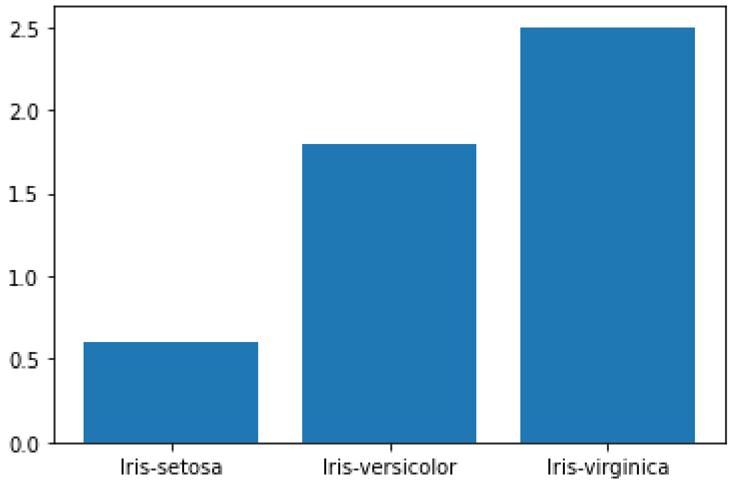


In [61]:

```
plt.bar(df['Species'],df['PW'])
```

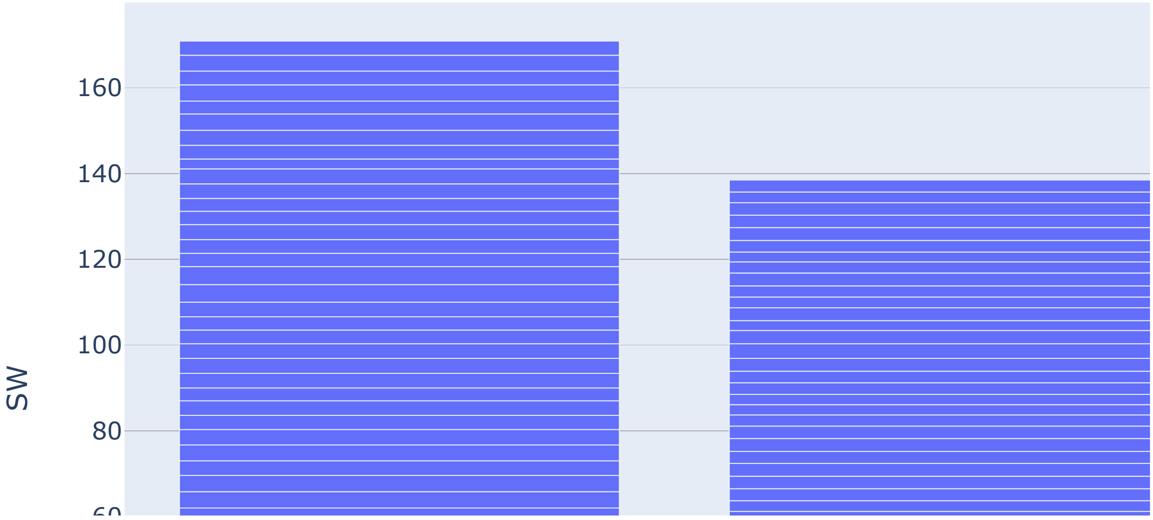
Out[61]:

```
<BarContainer object of 150 artists>
```



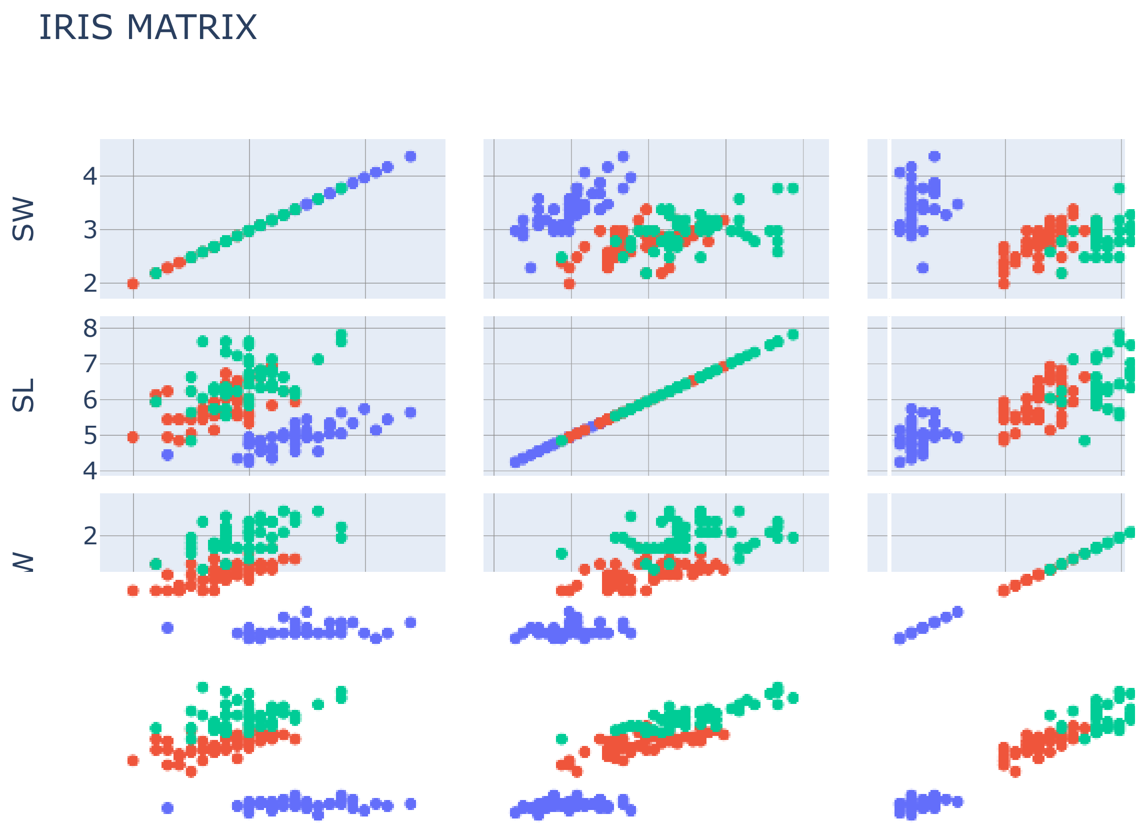
In [62]:

```
px.bar(df,x="Species",y="SW")
```



In [63]:

```
px.scatter_matrix(df,color='Species',title='IRIS MATRIX',dimensions=['SW','SL','PW','PL'])
```



In [64]:

```
y=df['Species']
from sklearn.preprocessing import LabelEncoder
l=LabelEncoder()
y=l.fit_transform(y)
y
```

Out[64]:

```
array([0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
       2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
       2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2])
```

In [65]:

```
x=df.drop('Species',axis=1)
x
```

Out[65]:

	SL	SW	PL	PW
0	5.1	3.5	1.4	0.2
1	4.9	3.0	1.4	0.2
2	4.7	3.2	1.3	0.2
3	4.6	3.1	1.5	0.2
4	5.0	3.6	1.4	0.2
...	...	...	...	...
145	6.7	3.0	5.2	2.3
146	6.3	2.5	5.0	1.9
147	6.5	3.0	5.2	2.0
148	6.2	3.4	5.4	2.3
149	5.9	3.0	5.1	1.8

150 rows × 4 columns

In [66]:

```
x=np.array(x)
x
```

Out[66]:

```
array([[5.1, 3.5, 1.4, 0.2],
       [4.9, 3. , 1.4, 0.2],
       [4.7, 3.2, 1.3, 0.2],
       [4.6, 3.1, 1.5, 0.2],
       [5. , 3.6, 1.4, 0.2],
       [5.4, 3.9, 1.7, 0.4],
       [4.6, 3.4, 1.4, 0.3],
       [5. , 3.4, 1.5, 0.2],
       [4.4, 2.9, 1.4, 0.2],
       [4.9, 3.1, 1.5, 0.1],
       [5.4, 3.7, 1.5, 0.2],
       [4.8, 3.4, 1.6, 0.2],
       [4.8, 3. , 1.4, 0.1],
       [4.3, 3. , 1.1, 0.1],
       [5.8, 4. , 1.2, 0.2],
       [5.7, 4.4, 1.5, 0.4],
       [5.4, 3.9, 1.3, 0.4],
       [5.1, 3.5, 1.4, 0.3],
```

In [67]:

```
from sklearn.model_selection import train_test_split

x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.3,random_state=2)
```

In [68]:

```
x_test
```

Out[68]:

```
array([[4.6, 3.4, 1.4, 0.3],
       [4.6, 3.1, 1.5, 0.2],
       [5.7, 2.5, 5. , 2. ],
       [4.8, 3. , 1.4, 0.1],
       [4.8, 3.4, 1.9, 0.2],
       [7.2, 3. , 5.8, 1.6],
       [5. , 3. , 1.6, 0.2],
       [6.7, 2.5, 5.8, 1.8],
       [6.4, 2.8, 5.6, 2.1],
       [4.8, 3. , 1.4, 0.3],
       [5.3, 3.7, 1.5, 0.2],
       [4.4, 3.2, 1.3, 0.2],
       [5. , 3.2, 1.2, 0.2],
       [5.4, 3.9, 1.7, 0.4],
       [6. , 3.4, 4.5, 1.6],
       [6.5, 2.8, 4.6, 1.5],
       [4.5, 2.3, 1.3, 0.3],
       [5.7, 2.9, 4.2, 1.3],
       [6.7, 3.3, 5.7, 2.5],
       [5.5, 2.5, 4. , 1.3],
       [6.7, 3. , 5. , 1.7],
       [6.4, 2.9, 4.3, 1.3],
       [6.4, 3.2, 5.3, 2.3],
       [5.6, 2.7, 4.2, 1.3],
       [6.3, 2.3, 4.4, 1.3],
       [4.7, 3.2, 1.6, 0.2],
       [4.7, 3.2, 1.3, 0.2],
       [6.1, 3. , 4.9, 1.8],
       [5.1, 3.8, 1.9, 0.4],
       [7.2, 3.2, 6. , 1.8],
       [6.2, 2.8, 4.8, 1.8],
       [5.1, 3.3, 1.7, 0.5],
       [5.6, 2.9, 3.6, 1.3],
       [7.7, 3.8, 6.7, 2.2],
       [5.4, 3. , 4.5, 1.5],
       [5.8, 4. , 1.2, 0.2],
       [6.4, 2.8, 5.6, 2.2],
       [6.1, 3. , 4.6, 1.4],
       [5.5, 2.3, 4. , 1.3],
       [6.9, 3.1, 5.1, 2.3],
       [6. , 2.9, 4.5, 1.5],
       [6.2, 2.9, 4.3, 1.3],
       [6.8, 3.2, 5.9, 2.3],
       [5. , 2.3, 3.3, 1. ],
       [4.8, 3.4, 1.6, 0.2]])
```

In [69]:

```
y_test
```

Out[69]:

```
array([0, 0, 2, 0, 0, 2, 0, 2, 2, 0, 0, 0, 0, 0, 1, 1, 0, 1, 2, 1, 1, 1,
       2, 1, 1, 0, 0, 2, 0, 2, 2, 0, 1, 2, 1, 0, 2, 1, 1, 2, 1, 1, 2, 1,
       0])
```

BY USING DT:

In [70]:

```
from sklearn import tree

d=tree.DecisionTreeClassifier()
d.fit(x_train,y_train)
```

Out[70]:

```
DecisionTreeClassifier()
```

ACCURACY OF THE GIVEN DATA SET:

In [73]:

```
pre_test=d.predict(x_test)
acc_test=accuracy_score(y_test,pre_test)*100
acc_test
```

Out[73]:

```
95.55555555555556
```

In [74]:

```
pre_test
```

Out[74]:

```
array([0, 0, 2, 0, 0, 1, 0, 2, 2, 0, 0, 0, 0, 0, 1, 1, 0, 1, 2, 1, 2, 1,
       2, 1, 1, 0, 0, 2, 0, 2, 2, 0, 1, 2, 1, 0, 2, 1, 1, 2, 1, 1, 2, 1,
       0])
```

In [75]:

```
y_test
```

Out[75]:

```
array([0, 0, 2, 0, 0, 2, 0, 2, 2, 0, 0, 0, 0, 0, 1, 1, 0, 1, 2, 1, 1, 1,
       2, 1, 1, 0, 0, 2, 0, 2, 2, 0, 1, 2, 1, 0, 2, 1, 1, 2, 1, 1, 2, 1,
       0])
```

In [ ]:

```

```

In [76]:

```
df.head()
```

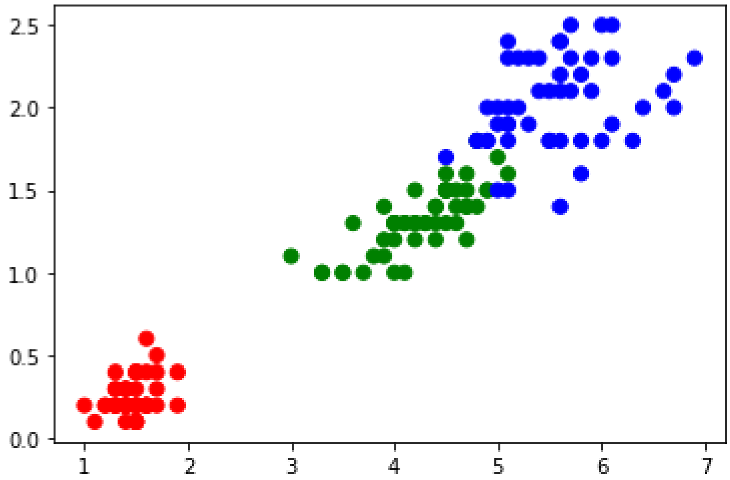
Out[76]:

	SL	SW	PL	PW	Species
0	5.1	3.5	1.4	0.2	Iris-setosa
1	4.9	3.0	1.4	0.2	Iris-setosa
2	4.7	3.2	1.3	0.2	Iris-setosa
3	4.6	3.1	1.5	0.2	Iris-setosa
4	5.0	3.6	1.4	0.2	Iris-setosa

K MEANS CLUSTERING:

In [77]:

```
colormap=np.array(['Red','green','blue'])
fig=plt.scatter(df['PL'],df['PW'],c=colormap[y],s=50)
```



In [78]:

```
x
```

```
       [4.6, 3.2, 1.4, 0.2],
       [5.3, 3.7, 1.5, 0.2],
       [5. , 3.3, 1.4, 0.2],
       [7. , 3.2, 4.7, 1.4],
       [6.4, 3.2, 4.5, 1.5],
       [6.9, 3.1, 4.9, 1.5],
       [5.5, 2.3, 4. , 1.3],
       [6.5, 2.8, 4.6, 1.5],
       [5.7, 2.8, 4.5, 1.3],
       [6.3, 3.3, 4.7, 1.6],
       [4.9, 2.4, 3.3, 1. ],
       [6.6, 2.9, 4.6, 1.3],
       [5.2, 2.7, 3.9, 1.4],
       [5. , 2. , 3.5, 1. ],
       [5.9, 3. , 4.2, 1.5],
       [6. , 2.2, 4. , 1. ],
       [6.1, 2.9, 4.7, 1.4],
       [5.6, 2.9, 3.6, 1.3],
       [6.7, 3.1, 4.4, 1.4],
```

In [79]:

```
y
```

Out[79]:

```
array([0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
       2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
       2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2])
```

In [80]:

```
from sklearn.cluster import KMeans
km=KMeans(n_clusters=3,random_state=2,n_jobs=4)
km.fit(x)
```

```
C:\Users\Bhanu\anaconda3\lib\site-packages\sklearn\cluster\_kmeans.py:938: F
utureWarning:

'n_jobs' was deprecated in version 0.23 and will be removed in 0.25.
```

Out[80]:

```
KMeans(n_clusters=3, n_jobs=4, random_state=2)
```

In [81]:

```
centers=km.cluster_centers_
print(centers)
```

```
[[5.9016129  2.7483871  4.39354839 1.43387097]
 [5.006      3.418      1.464      0.244     ]
 [6.85       3.07368421 5.74210526 2.07105263]]
```

In [82]:

```
km.labels_
```

Out[82]:

```
array([1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 0, 0, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 2, 0, 2, 2, 2, 2, 0, 2, 2, 2,
       2, 2, 2, 0, 0, 2, 2, 2, 2, 0, 2, 0, 2, 0, 2, 2, 0, 0, 2, 2, 2, 2,
       2, 0, 2, 2, 2, 2, 0, 2, 2, 2, 0, 2, 2, 2, 0, 2, 2, 0])
```

In [83]:

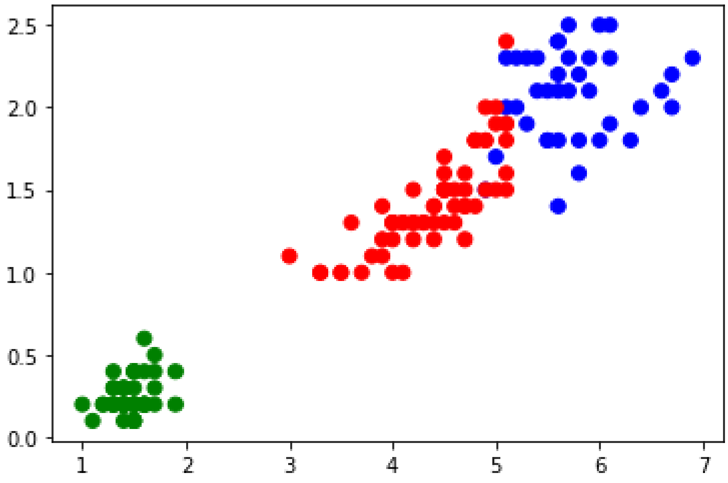
```
Catagory_kmeans=['Iris-Versicolor', 'Iris-Setosa', 'Iris-Virginica']
Catagory_kmeans
```

Out[83]:

```
['Iris-Versicolor', 'Iris-Setosa', 'Iris-Virginica']
```

In [84]:

```
colormap=np.array(['Red','green','blue'])
fig=plt.scatter(df['PL'],df['PW'],c=colormap[km.labels_],s=50)
```

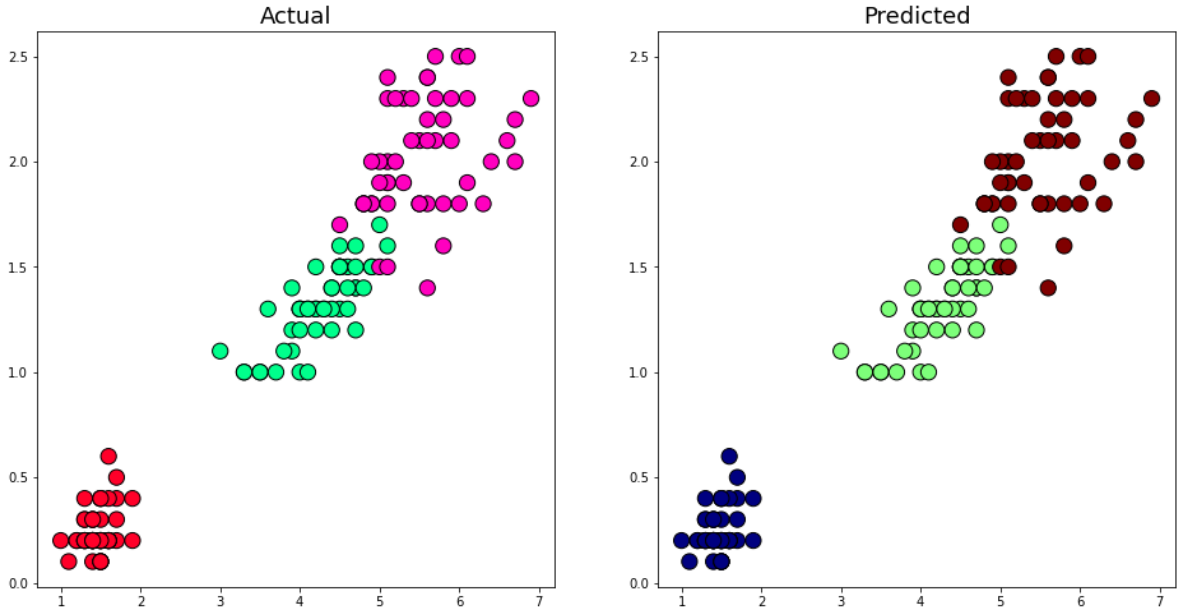


In [85]:

```
new_labels=km.labels_
fig,axes=plt.subplots(1,2,figsize=(16,8))
axes[0].scatter(x[:,2],x[:,3],c=y,cmap='gist_rainbow',edgecolor='k',s=150)
axes[1].scatter(x[:,2],x[:,3],c=y,cmap='jet',edgecolor='k',s=150)
axes[0].set_title('Actual',fontsize=18)
axes[1].set_title('Predicted',fontsize=18)
```

Out[85]:

```
Text(0.5, 1.0, 'Predicted')
```



In [86]:

```
x_km=np.array([[4 ,4, 3, 5]])
x_km_prediction=km.predict(x_km)
x_km_prediction[0]
print(Catagory_kmeans[int(x_km_prediction[0])])
```

```
Iris-Versicolor
```

In [ ]: